System Specification

Smart Acknowledge

V 1.7

San Ramon, CA, USA
Original Release 1.0, 2010

Executive Summary

This document is owned by the Technical Working Group (TWG) of the EnOcean Alliance. It is maintained and will be progressed within the authority of the chairman of the TWG.

Changes to this document will have to be proposed to the TWG for decision. A team of senior engineers will then act upon request by the TWG.

REVISION HISTORY

Ver.	Editor	Change	Date
1.1	MH	Changed the candidate priority evaluation.	
1.2	MH	Changed shortcuts for telegrams	
1.3	MH	Added criteria within candidate evaluation	
1.4	ASt	Exported document from system spec	15.10.2010
1.5	AP	ERP1 and ERP2 specific changes	19.08.2013
1.6	BE	Editorial review.	31.3.2014
1.7	MH	Editorial review.	05.11.2018

DISCLAIMER

This information within this document is the property of the EnOcean Alliance and its use and disclosure are restricted. Elements of the EnOcean Alliance specifications may also be subject to third party intellectual property rights, including without limitation, patent, copyright or trademark rights (such a third party may or may not be a member of the EnOcean Alliance.) The EnOcean Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. This document and the information contained herein are provided on an “as is” basis and the EnOcean Alliance disclaims all warranties express or implied, including but not limited to

(1) any warranty that the use of the information herein will not infringe any rights of third parties (including any intellectual property rights, patent, copyright or trademark rights, or

(2) any implied warranties of merchantability, fitness for a particular purpose, title or noninfringement.

In no event will the EnOcean Alliance be liable for any loss of profits, loss of business, loss of use of data, interruption of business, or for any other direct, indirect, special or exemplary, incidental, punitive or consequential damages of any kind, in contract or in tort, in connection with this document or the information contained herein, even if advised of the possibility of 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 Smart Acknowledge 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.
2603 Camino Ramon, Suite 200

San Ramon, CA 94583
USA
Graham Martin
Chairman & CEO EnOcean Alliance

Table of Contents

1. Introduction

SMART ACK enables a fast response bidirectional communication between an energy autarkic sensor and a powered line device (Controller). The communication is managed by a Controller that responds to the Sensors telegrams with acknowledges. A scenario with the HVAC as a Controller is shown in the figure below.

Figure 1 SMART ACK scenario

1.1 Functional description

SMART ACK is a bidirectional communication protocol between at least two actors. At least one actor must be an energy autarkic Sensor, and at least one must be a line powered Controller. The communication protocol and data payload in the Sensor to Controller direction is not in the scope of this document. SMART Acknowledge describes the communication in direction from the Controller to the autarkic Sensor. This leads us to the main challenge of SMART Acknowledge design:

- The radio receive mode consumes a large amount of energy

To keep the receive mode on the sensor as short as possible we use message synchronization (Message flow is performed in predefined time intervals). A sensor sends its data after a defined period then a reclaim command and expects then the answer telegram in a predefined very short time slot. In this time the sensor's receiver is active. For this purpose we declare Mail Boxes. A Mail Box is specific for each sensor where the controller stores the messages for the particular sensor for the time when his receiver is active. Telegrams from Controllers are collected into the Mail Box. A Sensor can reclaim telegrams that are in his Mail Box.

Communication through repeaters can cause unknown time delays and synchronization issues. Therefore the second challenge of SMART Acknowledge is:

- Unknown time delays are introduced by repeaters

To eliminate unknown delays Mail Boxes are established in line powered devices with direct radio link to autarkic devices. This device is called the Post Master. Post Masters manage all Sensors' Mail Boxes. A Post Master can also be a Controller or a repeater.

1.2 Role and actors definition

In the SMART ACK protocol we define these actors:

- SMART ACK Sensor
- SMART ACK Repeater
- SMART ACK Controller

and the role of:

- Post Master

Actors are EnOcean radio devices that can take roles. Only the Repeater or Controller can take the role of Post Master. The SMART ACK operations are the same on Controller and Repeater, but a Controller can additionally Learn Sensors and perform other tasks while a Repeater cannot.

1.2.1 SMART ACK Sensor

The SMART ACK Sensor is an autarkic device that sends measured data and reclaims acknowledges. It can be learned by Controller. Both the Controller and the Sensor know about each other when they are learned. The Sensor periodically wakes up executes its application and goes again to sleep. This is repeated in an infinite loop.

1.2.2 Post Master Role

The Post Master is the closest possible SMART ACK actor to the Sensor. The Post Master can be a Repeater or Controller. In their role as Post Master they behave the same. Post Masters hold Sensors' Mail Boxes.

General SMART ACK activities of a Post Master consist of:

- handling reclaims
- capture data sent to Sensors and store the data in its Mail Box

1.2.3 Controller

A Controller is any line powered device that can process the Sensors' Data and can send Data back to the Sensor. It can have a back bone connection and besides SMART ACK also other functionalities. A Controller can learn Sensors. A Controller can take the role of a Post Master.

1.2.4 Repeater

A Repeater is a line powered device with the capability to repeat EnOcean telegrams. It can take the role of a Post Master. Besides SMART ACK it can have other functionalities i.e. light actuator.

The functionality of SMART ACK and repeater are independent. One of them can be turned off without affecting the other.

1.3 Reclaim process

Reclaim is the process to transfer messages from the Post Master to the Sensor. The main target is to keep the receiver-on-time of the sensors as short as possible. The best way is to enable the receiver mode only for the transfer time of the telegram. For this purpose time synchronization between Sensor and Post Master is established.

The synchronization is provided by telegrams. A Sensor sends a reclaim telegram and immediately after sending turns on the receiver mode. After receiving a Reclaim, the Post Master directly sends the answer telegram – Acknowledge. Therefore it is called SMART Acknowledge.

1.3.1 Implementation aspects

The reclaim process is outlined below:

- The Sensor sends an initial data telegram. This starts the communication.
- The Sensor waits the response period. This time is for the system to process the initial data message and prepare an acknowledge
- The Sensor sends reclaim telegram.
- The Sensor waits the Minimum reclaim period. This is the time for Post Master to process the reclaim telegram and send an acknowledge. Period between end sending “Reclaim” and enabling receiver on Sensor.
- The Sensors turns on the receiver.
- The Sensor receives Acknowledge from Post Master.

OR

- After the Maximum reclaim period the Sensor turns off the receiver. This is the time-out of reclaim process.

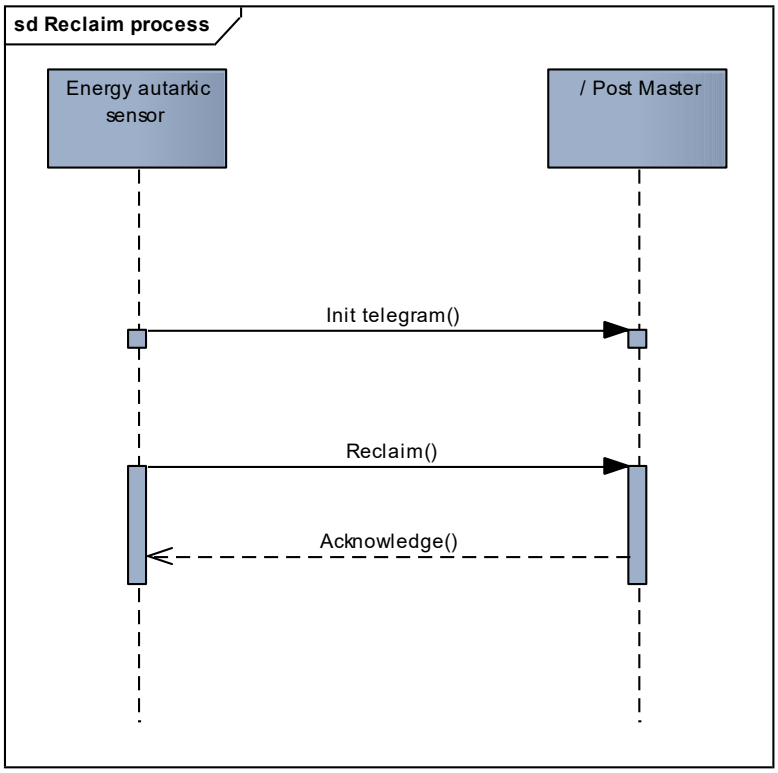


Figure 2 Reclaim process[1]

The receiver-on-time of a Sensor in the actual reclaim period shall be small as possible. It varies. Maximum is defined as Maximum reclaim period, typical value is the Actual reclaim period.

The response period is different in the learning and operating processes:

- Learning mode uses the Standard response period
- Operating mode uses a response period that has been determined by the Controller.

Details about the period's definition above can be found in chapter 2.3.

Both Reclaim and Acknowledge telegrams are not repeated. They reach their target at first hop, because the Sensor and the Post Master have always a direct link.

1.3.1.1 Mail Box

One Mail Box can only hold one telegram at a time. On a Post Master there are two types of Mail Boxes:

- Temporary Mail Box

[1] Initial telegram can be Data or a Learn Request telegram.

This Mail Box is used during LearnIn. LearnIn Acknowledge is stored in this Mail Box. There is only one temporary mail box on a SMART ACK enabled device (Repeater or Controller).

- Normal Mail Box

  This Mail Box is used during normal operation. Every Sensor and Controller pair has their own Mail Box. In the case where a Sensor is logically linked to more than one Controller there will be an additional Mail Box for each Sensor and Controller pair. These Mail Boxes are identified by a Mail Box index. Indexes are managed by the Post Master, every Sensors Mailbox is indexed (i.e. every Sensor starts with Mailbox Index 0). For every Sensor Controller pair there is exactly one Mail Box in the Post Master.

A Post Master has several possibilities how response to the reclaim messages from a Sensor. If the Sensor has a Mail Box at the Post Master the Post Master will respond with the appropriate reclaim response or will reply with the contents of the Mail Box the Sensor is reclaiming.

Name	Description
Mail Box empty	Sent when Post Master has no new Acknowledge for Sensor.
Mail Box does not exits	Sent by devices declared Post Master in the case that Sensor tries to reclaim in operating mode a Mail Box with non existing Mail Box index.

Table 1 Post Master signals

If the reclaim process fails, a Sensor will try up to 3 times to do a successful reclaim attempt of one Acknowledge. The wait period between two reclaims is defined as the period between two sub-telegrams[2].

Other than SMART ACK devices will notice the reclaim attempts as one telegram.

By the first reclaim attempt the Post Master starts a time out period for all next reclaim attempts. This period is called the Mail Box period. In this period the Sensor may transmit an additional reclaim attempt. After this period elapses the Post Master sends a Mail Box empty signal when he tries to reclaim the corresponding Mail Box. The Mail Box period duration is related to the sub-telegram timing of one telegram. The period is again restarted when the Mail Box is filled with a new acknowledge.

[2] See https://www.enocean-alliance.org/specifications/ for details on radio specification.

2 Functional modes

SMART Acknowledge functionality is different in Learning and Operating modes.

The message flow for Learning and Operating modes is similar, but the data content of the telegrams is different. The telegrams have the following conversions:

	Learning	Operating
Init telegram	Learn Request	Data
Reply	Learn Reply	Data Reply
Reclaim	Learn Reclaim	Data Reclaim
Acknowledge	Learn Acknowledge	Data Acknowledge

Table 2 SMART ACK telegrams

2.1 Learning

Learning is a process where devices exchange information about each other in order to create the logical links in the EnOcean network. It can result in LearnIn or LearnOut. Learning additional Controllers is also enabled. It is shown below how SMART ACK supports repeated Learning of one Sensor.

The Actors must be at their final installation location. The learning process goes in these steps:

- The Controller is switched to learn mode.
- The autarkic Sensor is switched to learn mode by sending a “Learn Request” telegram.
- Post Master gets determined.
- Controller decides if the learning is a LearnIn or LearnOut.
- Sensor reclaims Learn Acknowledge.

2.1.1 Determining the Post Master

To determine the Post Master we try to find the nearest SMART ACK device to the Sensor that can hold a Sensor’s Mail Boxes. It can be the Controller itself or a Repeater. Based on this in SMART ACK we differ between two actors positioned in field:

- Sensor and Controller are within each other’s transmit and receive radio range; Controller is Post Master – simple mode
- Sensor and Controller are not within each other’s transmit and receive radio range, the communication is enabled by repeaters, the Controller is not the Post Master – advanced mode
- Sensor and Controller are not within each other’s transmit and receive radio range, the contact is enabled by smart acknowledge repeating – advanced mode Repeater

The priority in determining Post Master is to establish simple mode. The recommendation is only when simple mode is not possible we try to establish advanced mode. In advanced mode

repeaters participate as candidates for Post Master. The Post Master gets selected by the Controller according to a priority policy. A Sensor can only have one Post Master. The process of finding the Post Master is repeated during every Learning. The Post Master gives up the Post Master duties for a Sensor when the Sensor is Learned Out from all controllers and thus has no more valid Mail Box indexes.

2.1.1.1 Implementation aspects

To determine if there is a direct radio link between a Controller and Sensor the Sensor sends a Learn Request telegram in Learn mode. When a Controller receives this telegram it is assumed that a direct radio link is available. To ensure good operating results the radio signal strength index – RSSI must be at a certain level. We call this level the "good enough RSSI".

When a Controller receives a Learn Request it automatically promotes itself to Post Master unless there was already a Post Master declared in previous learn process with different parties. Every Repeater must participate as a candidate in determining the Post Master. Repeaters enter their information into the Learn Request telegram and repeat it. The Controller collects all these telegrams and selects one candidate to become the Post Master. Collecting starts with the first Learn Request telegram and lasts for the Learn Request period.

Repeater information entered into the Learn Request data field:

- the RSSI value (in dBm, of received Telegram from a Sensor)
- own ID
- request code

Request code gives information how is the Sensor related to the Repeater.

Request code	Description
0b11111	Default value – sent by Sensor
0b00000	I am no Post Master and do not have place for next Mail Box.
0b00001	I am no Post Master and do have place for next Mail Box.
0b00010	I am a Post Master and do not have place for next Mail Box.
0b00011	I am a Post Master and do have place for next Mail Box.

Table 3 Request code

NOTE: When other Repeaters receive a "Learn Request" and a Request code is filled they just repeat it.

The postmaster is selected by a priority system. Every participating candidate gets a priority. Controllers also participate as candidates. The candidate with the highest priority is then selected.

If the controller runs for post master this means he is a local device. If a different Smart Ack actor runs for post master he is a remove device.

The hierarchical priority system is determined by these criteria:

- if there is already a post master for the sensor defined(detected from Request code) (weight 8)
- if the candidate has place for next mailbox (detected from Request code) (weight 4)
- if the RSSI from Sensor to the candidate is good enough (weight 2)
- if the candidate is the local device or not (weight 1)

Priority	1. PM (weight 8)	2. PLACE OK (weight 4)	3. RSSI OK (weight 2)	4. Local/ Remote (weight 1)
15	1	1	1	1
14	1	1	1	0
11	1	0	1	1
10	1	0	1	0
7	0	1	1	1
6	0	1	1	0
5	0	1	0	1
4	0	1	0	0
3	0	0	1	1
2	0	0	1	0
1	0	0	0	1
0	0	0	0	0

Table 4 Post Master Selection Priorities

If criteria is TRUE = 1 or FALSE = 0. Local candidate = 1 and Remote candidate = 0.

Special case is priority 6. In this case we want to find the remote candidate:

- with the lowest hop (repeater) count between controller and candidate
- if same hop count then with the highest RSSI value greater than the “good-enough” RSSI value

Candidates above priority 6 included are accepted as post master. When no candidate above 6 included is available learning will fail.

When a remote candidate is promoted to post master the Controller sends a Learn Reply telegram to the Post Master with information about the Sensor and to know about the Learning result.

The Post Master scenario that results in simple mode is pictured below.

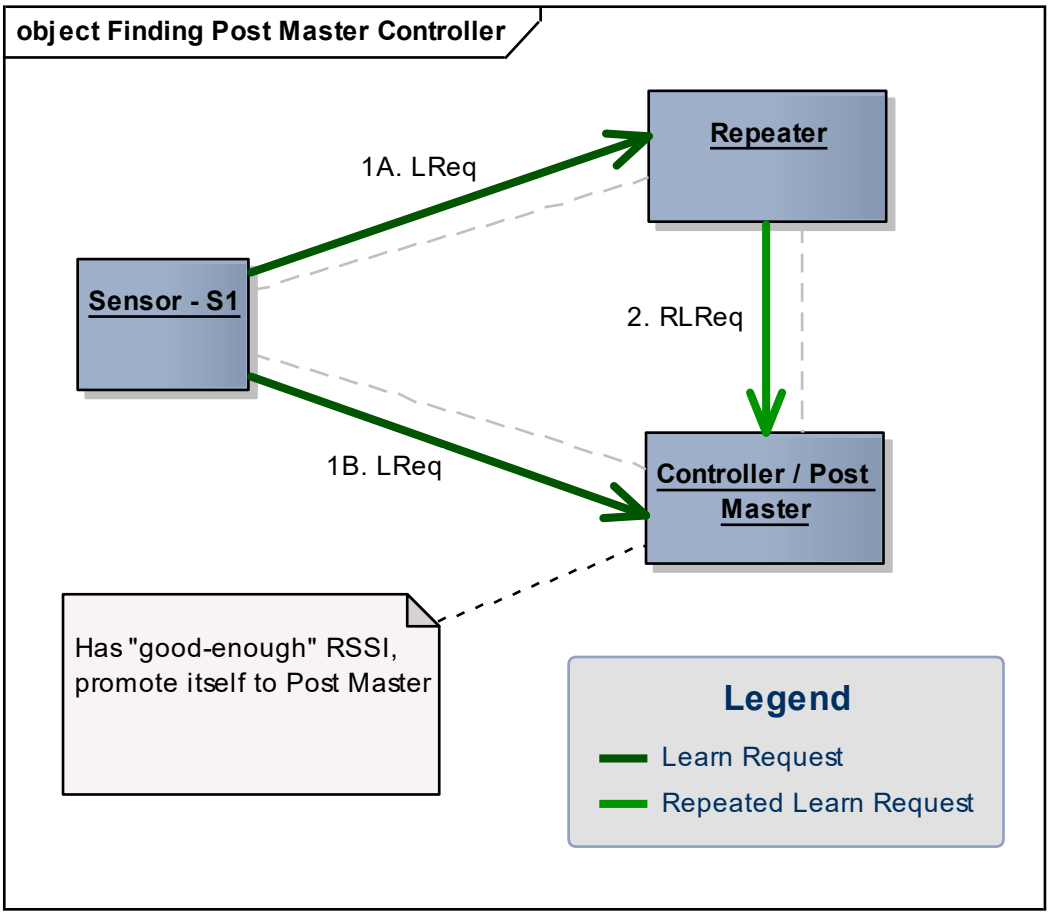
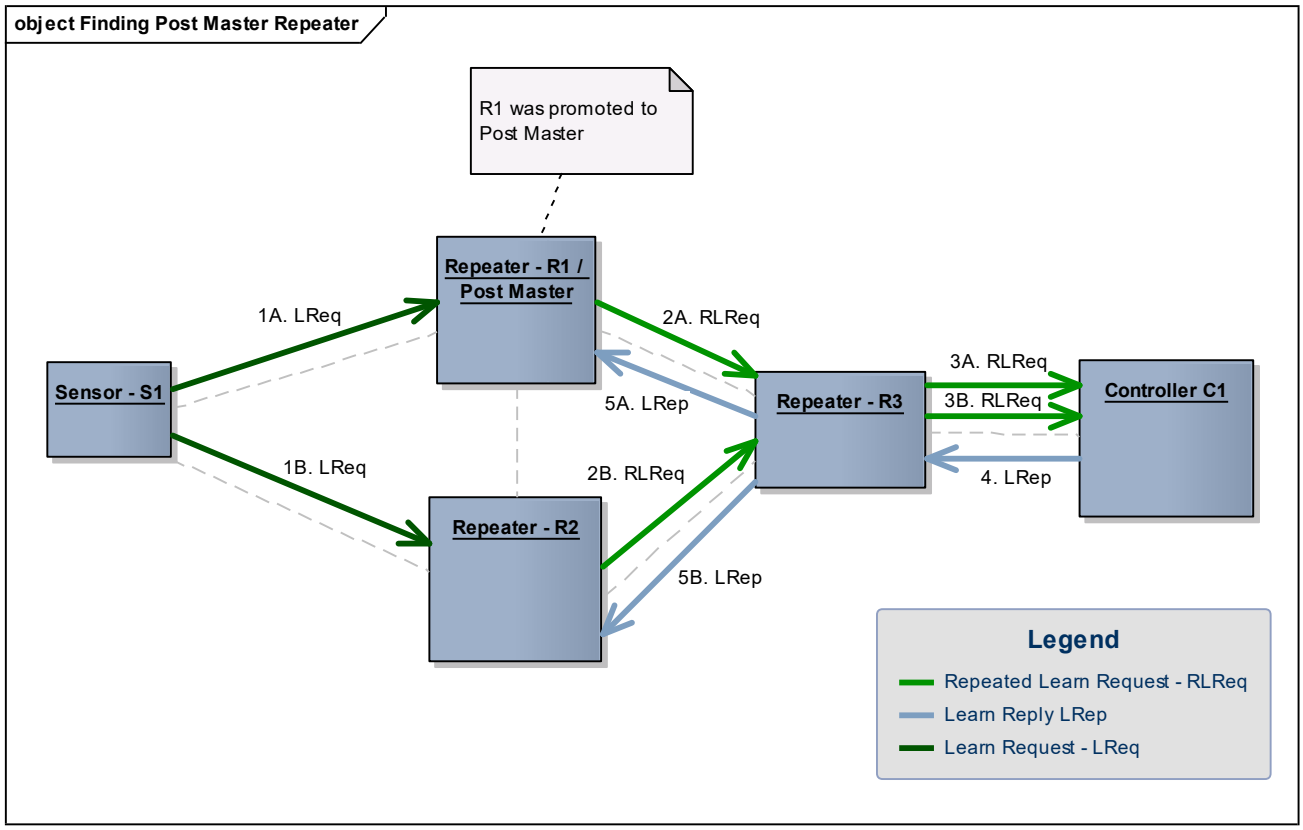


Figure 3 Finding Post Master – simple mode[3]

Finding the Post Master that results in advanced mode is pictured below.

[3] Controller collects the original and repeat Learn Request and then promotes itself to Post Master.



- R1 has better RSSI of received Learn Request
- R1 and R2 are candidates for Post Master. R1 is promoted to Post Master.

Figure 4 Finding Post Master – advanced mode

2.1.2 LearnIn and LearnOut

The Controller decides if the ongoing learning operation results in:

- First LearnIn, performed when a controller creates a link with a Sensor. Mail Box created.
- Success of and additional LearnIn or partial LearnOut
- LearnOut, performed when controller removes an existing link with a Sensor. Mail Box deleted
- Failure - any of above

This decision is based on the following criteria:

- If the Sensor is already learned by the Controller.
- If the Controller allows repeated learning of one sensor.
- If the Controller accepts the Sensor's EEP/GP
- Technical feasibility of connection (Post Master candidate above 6 included is found)

NOTE: The decision is made by the user application and there may be additional user defined criteria.

During LearnIn the Controller determines the response time that will be used in the reclaim process as part of Operating mode. The information is included in Acknowledge to the Sensor.

2.1.2.1 Implementation aspects

Information about a controller Learn decision is included in the Controllers answer to the Sensor. It is the Acknowledge code.

Acknowledge code	Description	Post Master action	Sensor interpretation
0x00	First LearnIn successful	Create Mail Box.	Create Mail Box information.
0x01 – 0x0F	Repeated LearnIn.	-	Application specific.
0x10 – 0x1F	Failed LearnIn	-	Application specific
0x20	Complete LearnOut.	Drop Mail Box.	Delete Mail Box information.
0x21 – 0x2F	Partial LearnOut.	-	Application specific

Table 5 Acknowledge code

According to the result a Post Master creates or removes a Mail Box. In the case of a repeated learn of one Sensor the existing Mail Box is used. Repeated LearnIn can be partially learned out or completely Learned out at once.

When a Sensor gets learned by several Controllers the same Post Master always gets elected. For every controller a new Mail Box is created at the Post Master. The Post Master indexes these Mail Boxes for every Sensor separately (i.e. first Mailbox from any sensor starts with index 0 and then increments). During the Learning process the Post Master informs the Sensor about the actual Mail Box index.

2.1.3 Learning Scenario

The Result of learning has no influence on the message flow. In the figures below Learning scenarios are shown for simple and advanced mode:

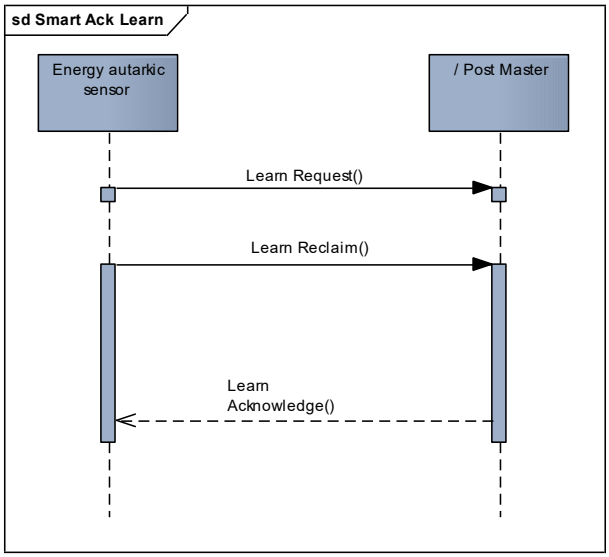


Figure 5 Learning – Simple mode

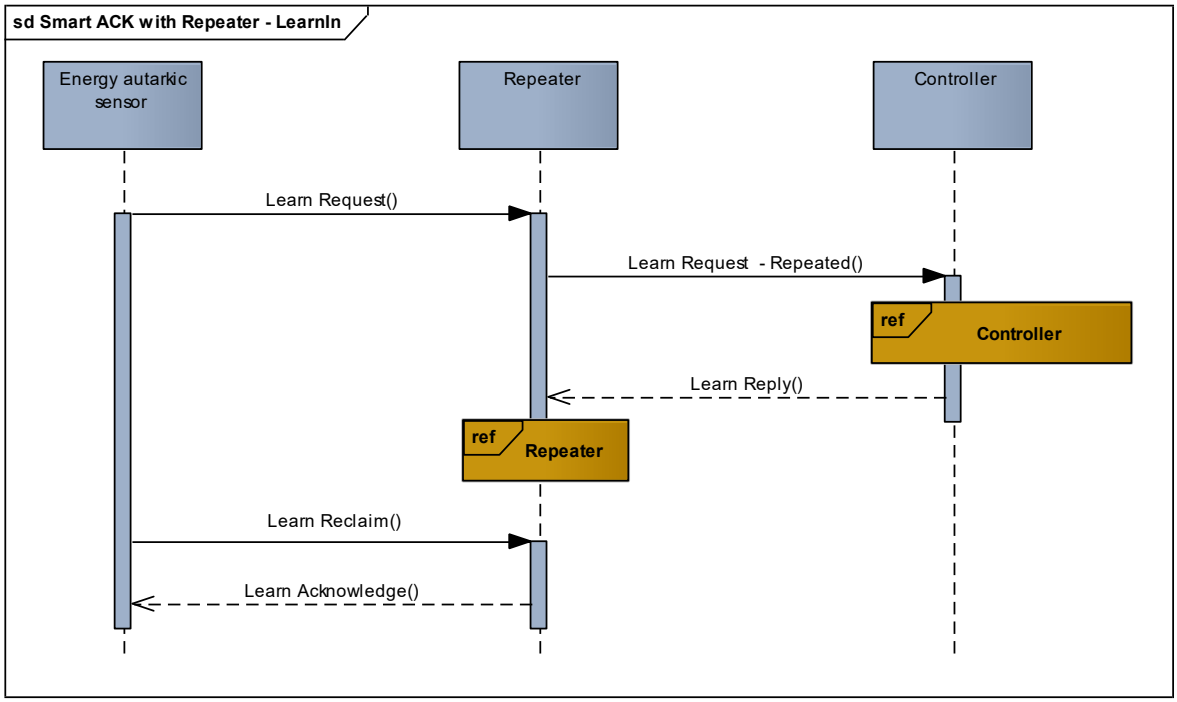


Figure 6 Learning - Advanced mode

2.2 Operating mode

Operating mode is when application data transfer is performed. It consists of these steps:

- Sensors sends a data telegram.
- Controller prepares Data Acknowledge telegram.
- Sensor reclaims Data Acknowledge.

In Operating mode the reclaim of acknowledge is optional. The Sensor user application can decide if it performs the reclaim process. The frequency of the data transfer is also defined by the Sensor user application.

See the simple (without repeater) and advance (with repeater) mode operating.

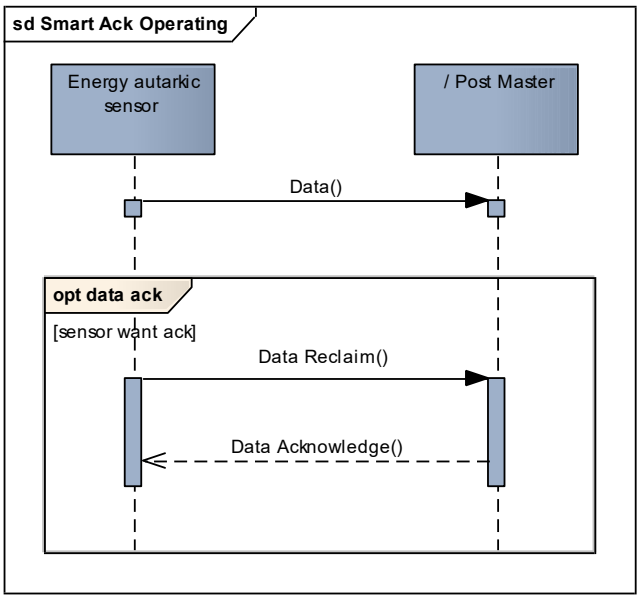


Figure 7 Operating – simple mode

When a Controller is not the Post Master his telegrams addressed to a Sensor are stored in a Mail Box on a remote Post Master. Telegrams are first sent to the Post Master and then reclaimed by the Sensor. These telegrams from the Controller to the Post Master are called Reply telegrams. When a Controller is the Post Master this step is not necessary.

NOTE: Reply telegrams contain the Acknowledge.

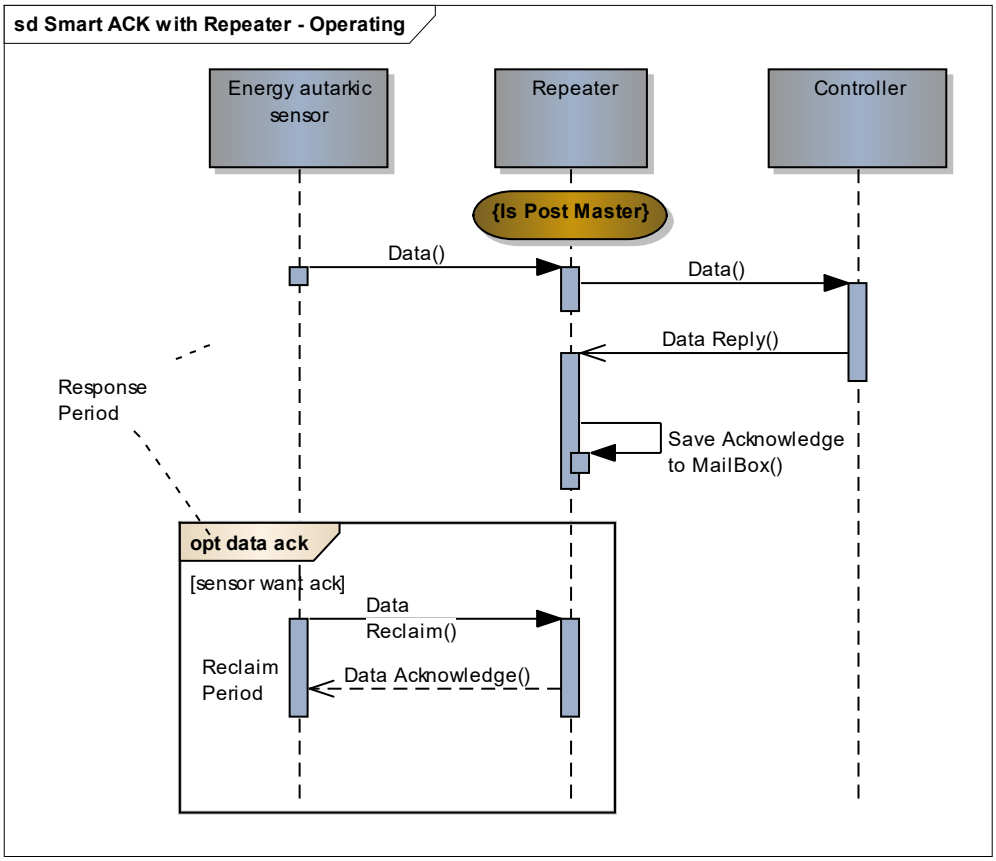


Figure 8 Operating – advanced mode

After sending a Data telegram the sensor waits the Response time and then starts the reclaim process. In this time the Controller has to prepare an Acknowledge and place it in the Mail Box. This period was declared by the controller while LearnIn before, to keep it is as short as possible.

When the Response time has elapsed the Sensor can send a Reclaim telegram. This telegram contains the Mail Box index in the Post Master. By repeating the data reclaim process Sensors can reclaim all of its Mail Boxes one by one.

NOTE: Sensor can have more than one Mail Box when it is learned by more than one Controller.

The message flow in Operating mode is in the picture below.

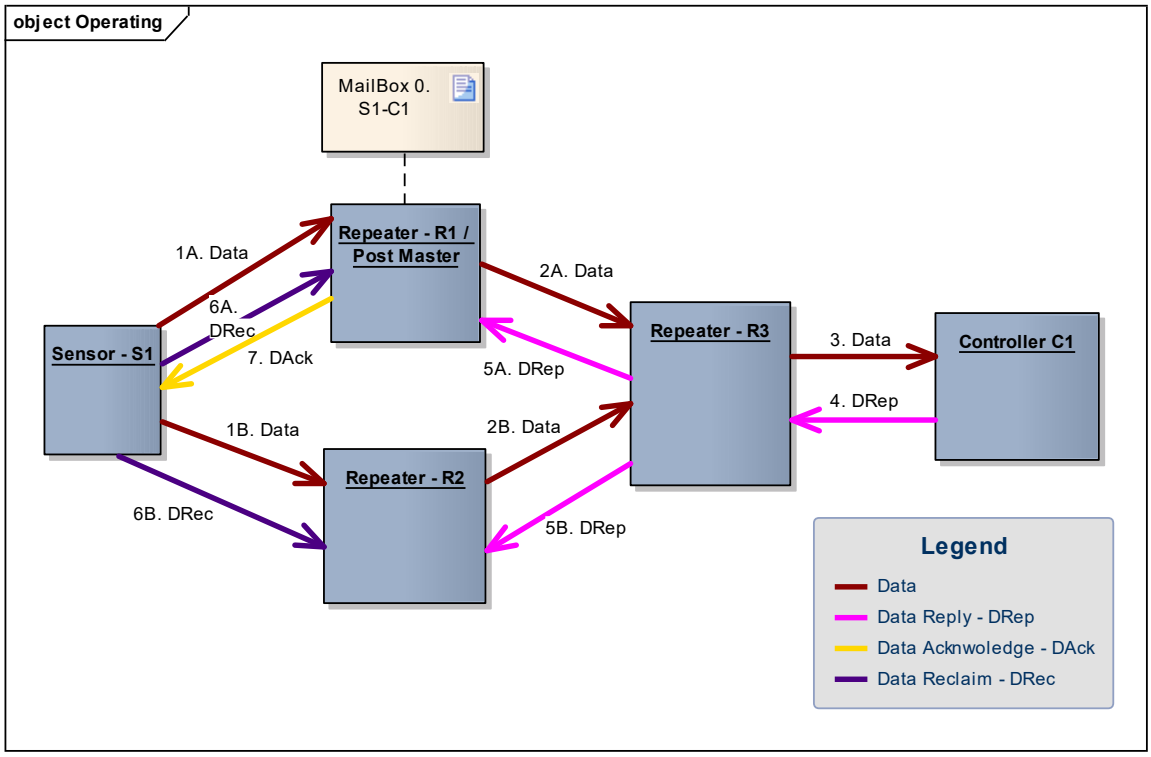


Figure 9 Operating message flow[4]

2.2.1 Remote reset

Remote reset process enables to rerun learning without having direct physical access to the Sensor. It signalizes the Sensor to start Learning.

After the sensor recognizes the remote reset signal it starts the learning by sending a Learn Request telegram.

2.2.1.1 Implementation aspects

The idea is to exchange the Data Acknowledge telegram for the reset signal telegram. It is the only possible way to reach the Sensor, because only in reclaim process it will listen for telegrams.

Sending of the reset signal message is performed by the Controller. The Controller can send a Reset signal instead of the Data Acknowledge. When the Controller is not available an external manager can calls a Remote Communization function on the Post Master to swap telegrams. This functionality is not described in this document. When the Post Master is unavailable/broken the swap is performed by an external device with specific functionalities.

A Reset scenario where the Controller sends the Reset signal is shown in the Figure below.

[4] Data Reply and Data Acknowledge have different status bytes, besides this they are same telegrams.

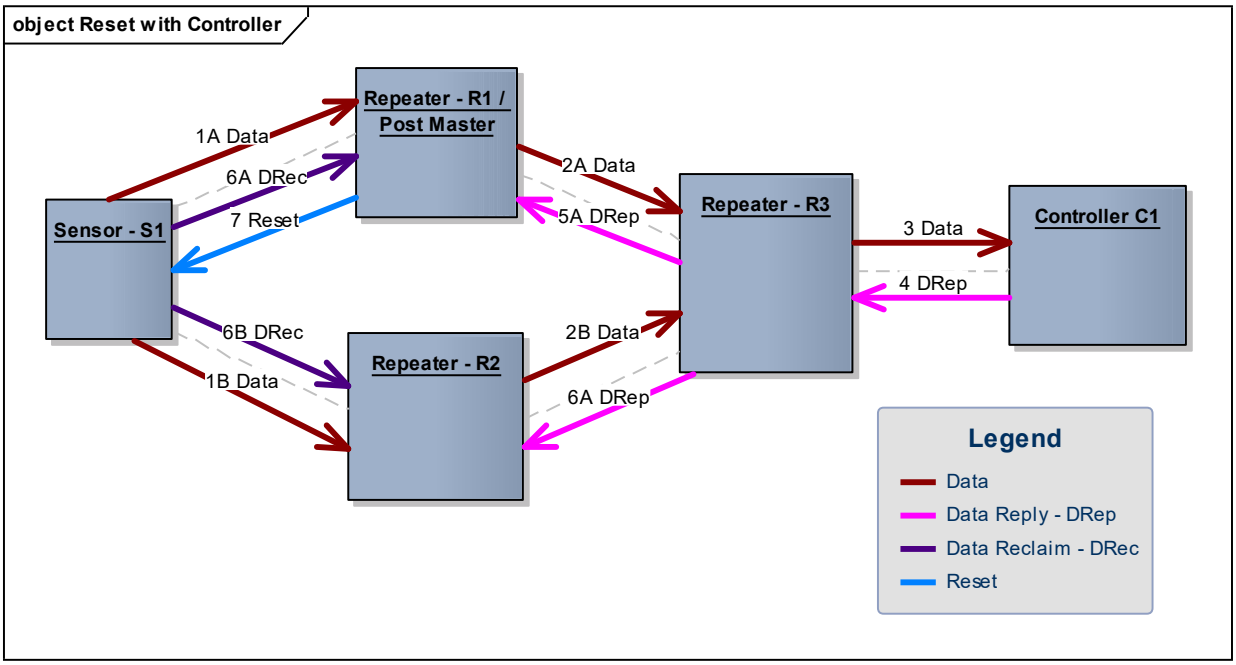


Figure 10 Reset scenario with Controller

A Scenario with the Remote Manager and broken Post Master is shown in the Figure below.

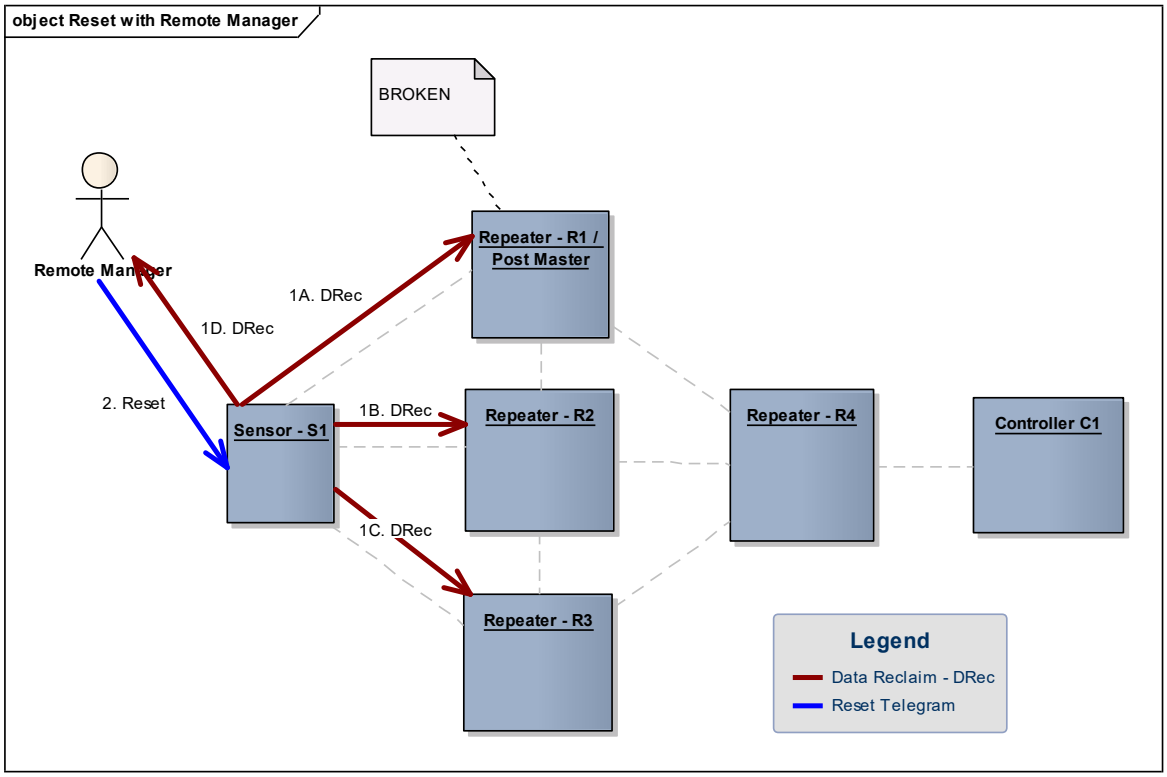


Figure 11 Reset Learn with Remote Manager

2.3 Timing definitions

The times are held as short as possible considering the lowest possible energy consumption but they initial definition was based for the Dolphin Chip platform.

Period name	Length	Description	Type
Learn Request period	250 ms	In period Controller collects Learn Request messages	constant
Standard response period	550 ms	Period between end sending “Learn Request” and start of sending “Learn Reclaim”.	constant
Response period	Min. 150 ms	Period between end sending “Data” and start sending “Data Reclaim”	Depending on Controller
Actual reclaim period	average 2,25 ms	Period of active receive mode on sensor, from enabling receiver till Acknowledge is received.	
Minimum reclaim period	2,5 ms	Period between end sending “Reclaim” and enabling receiver on Sensor	constant
Maximum reclaim period	6 ms	Maximum waiting period for Acknowledge. From enabling receiver to disabling receiver without received Acknowledge.	constant
MailBox period	120 ms	Period in that Sensor can repeated try to reclaim a mailbox, that was reclaimed before.	constant

Table 6 Periods

3 SMART ACK telegram description

SMART ACK protocol uses these telegrams:

- Smack Learn Request telegram
- Smack Learn Answer telegram
- Smack Reclaim telegram
- Signal telegram
- Common EnOcean telegrams (4BS, VLD)

NOTE: For more details see general telegram description[5].

For ERP1: Usage of CRC8 is mandatory for all SMART ACK telegrams

RORG	data field	sender id	status	crc8
1 byte	**Payload**	4 bytes	1 byte	1 byte

For ERP2:

[5] See: https://www.enocean-alliance.org/specifications/

Length	sender id	data field	crc8
1 byte	4 bytes	**payload**	1 byte

Telegrams described in the chapter functional description are implemented by one of these telegram types. Only the data payload is documented, independent of using ERP1 or ERP2.

SMART ACK description	Telegram	Shortcut
Learn Request	Smack Learn Request telegram	sm_lrn_req
Learn Reply	Smack Learn Answer telegram	sm_lrn_ans
Learn Reclaim	Smack Reclaim telegram	sm_rec
Learn Acknowledge	Smack Learn Answer telegram	sm_lrn_ans
Data	Common EnOcean data telegram	-
Data Reply	Common EnOcean data telegram	-
Data Reclaim	Smack Reclaim telegram	sm_rec
Data Acknowledge	Common EnOcean data telegram	
MailBox Empty	Signal telegram	sig
MailBox does not exist	Signal telegram	sig
Reset	Signal telegram	sig

Table 7 SMART ACK telegrams overview

Since one telegram implements several telegrams, as described before, we declare a unique index to separate the implemented telegrams. Indexes are unique within one telegram.

Index	SMART ACK description	Telegram
0b0	Learn Reclaim	Smack Reclaim telegram
0b1	Data Reclaim	Smack Reclaim telegram
-	Learn Request	Smack Learn Request telegram
0x01	Learn Reply	Smack Learn Answer telegram
0x02	Learn Acknowledge	Smack Learn Answer telegram
0x01	Mail Box Empty	Signal telegram
0x02	Mail Box does not exist	Signal telegram
0x03	Reset	Signal telegram

Table 8 Message indexes

3.1.1 Flag codes

To sum up, here are listed the request and acknowledge codes. The flag codes are directly used in telegrams.

Request code	Description
0b11111	Default value – send by Sensor
0b00000	I am not Post Master and do not have place for next Mail Box.
0b00001	I am not Post Master and do have place for next Mail Box.
0b00010	I am Post Master and do not have place for next Mail Box.
0b00011	I am Post Master and do have place for next Mail Box.

Table 9 Request code

Acknowledge code	Description	Post Master action	Sensor interpretation
0x00	First LearnIn successful	Create Mail Box.	Create Mail Box information.
0x01 – 0x0F	Repeated LearnIn.	-	Application specific.
0x10 – 0x1F	Failed LearnIn	-	Application specific
0x20	Complete LearnOut.	Drop Mail Box.	Delete Mail Box information.
0x21 – 0x2F	Partial LearnOut.	-	Application specific

Table 10 Acknowledge code

3.1.2 Learn Request

Name		Learn Request
Used telegram		SMART ACK Learn Request Telegram
R-ORG		ERP1=0xC6, ERP2= Extended telegram type 0x01
Message index		N/A
Data length		10 bytes
Data content	Request code	5 bits
	Manufacturer ID	11 bits
	EEP (EnOcean Equipment Profile)	3 bytes
	RSSI [dBm]	1 byte
	Repeater ID	4 bytes
Send with sub-telegram count		3
Repeated		no*
Send by		Sensor
Addressed to		N/A

Table 11 Learn request description[6]

	7	6	5	4	3	2	1	0
0	REQUEST CODE							
1	MANUFACTURER ID							
2	EEP							
3								
4								
5	RSSI							
6	REPEATER ID							
7								
8								
9								

Table 12 SMART ACK Learn request structure

[6] Sensor sends learn request to not repeat. But SMART ACK devices alter telegram and send it.

3.1.3 Learn Reply

Name	Learn Reply
Used telegram	Smack Learn Answer Telegram
R-ORG	ERP1=0xC7, ERP2=Extended telegram type 0x02
Message index	0x01
Data length	7 bytes
Data content	Response time [ms] 2 bytes Acknowledge code 1 byte Sensor ID 4 bytes
Send with sub-telegram count	3
Repeated	yes
Send by	Controller
Addressed to	Post Master :: Repeater

Table 13 Learn reply description

	7	6	5	4	3	2	1	0
0	0x01							
1	RESPONSE TIME							
2								
3	ACKNOWLEDGE CODE							
4	SENSOR ID							
5								
6								
7								

Table 14 Learn reply structure

3.1.4 Learn Acknowledge

Name	Learn Acknowledge
Used telegram	Smack Learn Answer Telegram
R-ORG	ERP1=0xC7, ERP2=Extended telegram type 0x02
Message index	0x02
Data length	4 bytes
Data content	Response time [ms] 2 bytes Acknowledge code 1 byte Mail Box index 1 bytes
Send with sub-telegram count	1
Repeated	no
Send by	Post Master :: {Controller, Repeater}
Addressed to	Sensor

Table 15 Learn Acknowledge description

Sender ID is always Controller ID, although the real sender is a repeater that was promoted to Post Master

	7	6	5	4	3	2	1	0
0	0x02							
1	RESPONSE TIME							
2								
3	ACKNOWLEDGE CODE							
4	MAIL BOX INDEX							

Table 16 Learn acknowledge structure

3.1.5 Learn Reclaim

Name	Learn Reclaim
Used telegram	Smack Reclaim Telegram
R-ORG	ERP1=0xA7, ERP2=no R-ORG: 5-Byte-Length-Telegram
Message index	0b0
Data length	1 bit
Data content	Learn reclaim indicator 1 bit
Data length	1 bit
Send with sub-telegram count	1
Repeated	no
Send by	Sensor
Addressed to	N/A

Table 17 Learn reclaim description

	7	6	5	4	3	2	1	0
1	0b0	NOT USED						

Table 18 Learn reclaim structure

3.1.6 Data Reclaim

Name	Data Reclaim
Used telegram	Smack Reclaim Telegram
R-ORG	ERP1=0xA7, ERP2=no R-ORG: 5-Byte-Length-Telegram
Message index	0b1
Data length	8 bits
Data content	Data reclaim indicator 1 bit Mail Box index 7 bits
Send with sub-telegram count	1
Repeated	no
Send by	Sensor
Addressed to	N/A

Table 19 Data reclaim description

	7	6	5	4	3	2	1	0
0	0b1	MAIL BOX INDEX						

Table 20 Data reclaim structure

3.1.7 Mail Box empty message

Name	Mail Box empty
Used telegram	Signal Telegram
R-ORG	ERP1=0xD0, ERP2=Telegram type 0x03
Message index	0x01
Data length	1 Byte
Data content	Message index 8 bits
Send with sub-telegram count	1
Repeated	no
Send by	Post Master :: {Controller, Repeater}
Addressed to	Sensor

Table 21 Mail Box empty description

	7	6	5	4	3	2	1	0
0	0x01							

Table 22 Mail Box empty structure

3.1.8 Mail Box does not exist

Name	Mail Box does not exists
Used telegram	Signal Telegram
R-ORG	ERP1=0xD0, ERP2=Telegram type 0x03
Message index	0x02
Data length	1 Byte
Data content	Message index 8 bits
Send with sub-telegram count	1
Repeated	no
Send by	Post Master :: {Controller, Repeater}
Addressed to	Sensor

Table 23 Mail Box does not exists description

	7	6	5	4	3	2	1	0
0	0x02							

Table 24 Mail Box does not exists structure

3.1.9 Reset

Name	Reset
Used telegram	Signal Telegram
R-ORG	ERP1=0xD0, ERP2=Telegram type 0x03
Message index	0x03
Data length	1 Byte
Data content	Message index 8 bits
Send with sub-telegram count	1
Repeated	no
Send by	Controller, Post Master :: {Controller, Repeater}, Remote Device
Addressed to	Sensor

Table 25 Reset description

	7	6	5	4	3	2	1	0
0	0x03							

Table 26 Reset structure

3.1.10 Data

Table 27 Data description

Name	Data
Used telegram R-ORG	Common EnOcean telegram defined by telegram type
Send with sub-telegram count	3
Repeated	yes
Send by	Sensor
Addressed to	N/A

3.1.11 Data reply

Table 28 Data reply description

Name	Data Reply
Used telegram R-ORG	Common EnOcean telegram defined by telegram type
Send with sub-telegram count	3
Repeated	yes
Send by	Controller
Addressed to	Sensor

3.1.12 Data acknowledge

Table 29 Data acknowledge description

Name	Data acknowledge
Used telegram R-ORG	Common EnOcean telegram defined by telegram type
Send with sub-telegram count	1
Repeated	no
Send by	Controller
Addressed to	Sensor

4 SMART ACK case studies

Here we want to show and discuss some interesting case studies utilizing SMART Acknowledge. They can further help to understand the protocol and how to implement it.

4.1 Field installations

Smart Acknowledge can be used in building and home automation or other applications. In the figure below is pictured a building automation scenario. More Sensors can be learned in one gateway. Repeaters are only used when the direct connection to the gateway is not reliable.

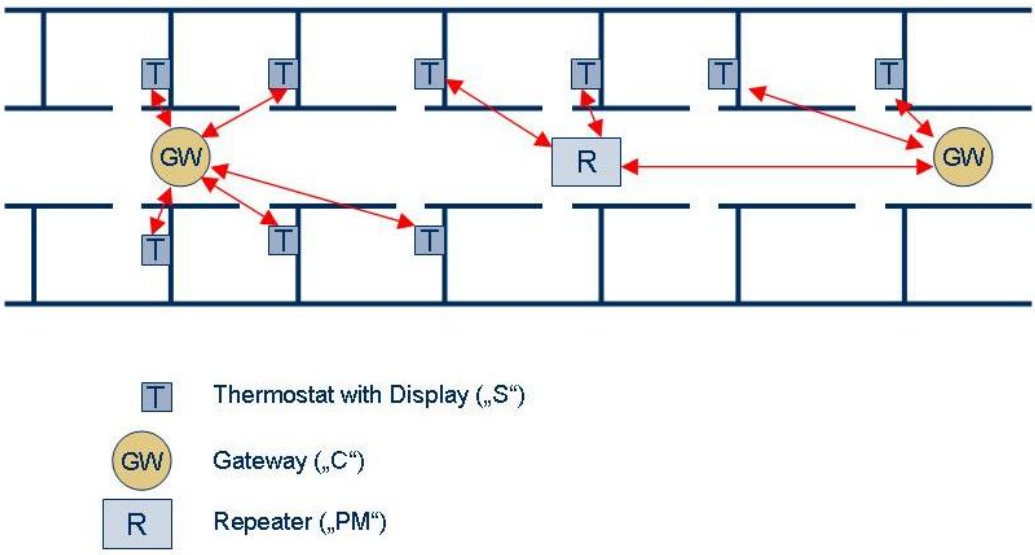


Figure 12 Building automation scenario

In the figure below is a home automation scenario. Autarkic modules like a central panel with display can regulate the lights and the heating. Therefore they are learned in more than one Controller. SMART ACK Controllers work as well with non SMART Acknowledge devices like wall switches and windows sensors.

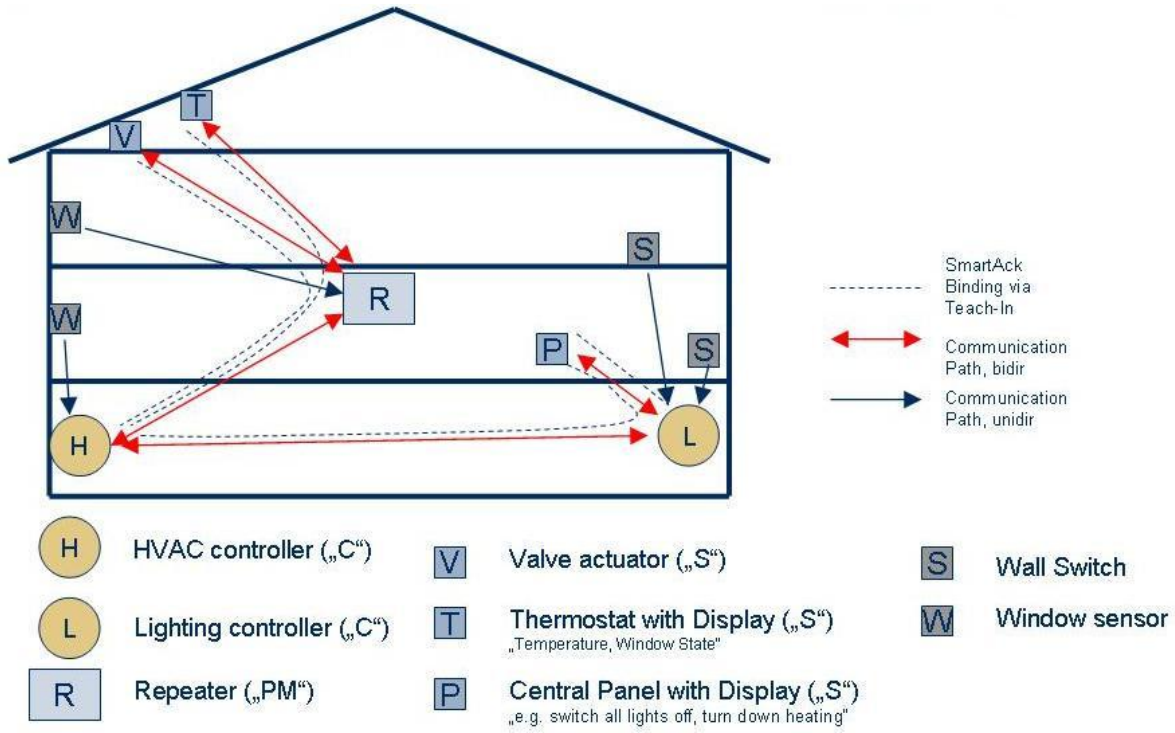


Figure 13 Home automation scenario

4.2 Operating notes

4.2.1 Does a Post Master get evaluated every time?

In learn mode a controller collects learn requests, determines the post master for the device pair, decides to LearnIn or LearnOut and prepares the learn reply or learn acknowledge. What if LearnIn is discarded, does a postmaster get evaluated? Yes, the post master gets evaluated every time, because when the learn is discarded we want the sensor to receive the information about the discarded LearnIn. The difference is, that the postmastership will only last for the learn acknowledge, no mail box is established at the promoted post master on discarded LearnIn.

This is the main motivation in case the Sensor and controller has not a direct link. If we want the sensor to receive the controller's decision, we must evaluate a remote post master every time. Accordingly, if the LearnIn is accepted or discarded, the post master establishes a permanent Mail Box or not.

4.2.2 Transfer of messages

Messages are multiple telegrams chained together. Scenarios where the Controller wants to transfer more than one "Data Acknowledge" messages after one "Data Reclaim" message may be required in operating mode. Is that possible? No, after a reclaim only one acknowledge can be transferred. However this feature is supported in the way that a Sensor can repeat several times the whole operating process and so receive more messages. The Controller can signalize in the first "Acknowledge" telegram that he wants to transfer more than one. The feature is not part of the SMART ACK protocol, it is an application depended.

4.2.3 Data update

In scenarios where the Response period must be as short as possible, we can prepare the Controller always with a "Data Reply" message. The Controller does not have to ask the back bone for an answer but as quickly as possible send the "Data Reply" message. The back bone would periodically update the Controller and provide him a "fresh data reply". This feature is application dependent.

4.2.4 Moving Sensors, changing installation location, add new devices

When a Sensor is moved, it is important to do the LearnOut process at its old installation location and then do the LearnIn at the new installation location. This is important because at the new installation location the old Post Master might not have a good-enough signal strength and error states can occur. This is important, because lots of unwanted system states can occur if ignored.

We can add sensors and repeaters in an already operating SMART ACK environment.

In simple learn mode the actors do not have to be at their final installation location, because the LearnIn process in this case is much simpler than with repeaters. But it is recommended to start LearnIn only when all devices are at their final installation location.

4.3 Actors behavior

4.3.1 Controller at LearnIn

To better understand the learning and operating modes we will discuss some scenarios involving actors.

In the below the behavior of a Controller at LearnIn is showed. The scenario is the following:

- The Controller receives Learn Request (original or repeated)
- The Controller will collect Learn Request telegrams for the Learn request period
- LearnIn
- According the result of post master election

— The Controller promotes itself to post master

OR

— The Remote Post Master gets elected and the Controller sends a Learn Reply

In the end the Sensors starts the reclaim process of the Learn Acknowledge, the declared Post Master will respond. This last step is not shown in the figure. All other steps are listened in the figure bellow.

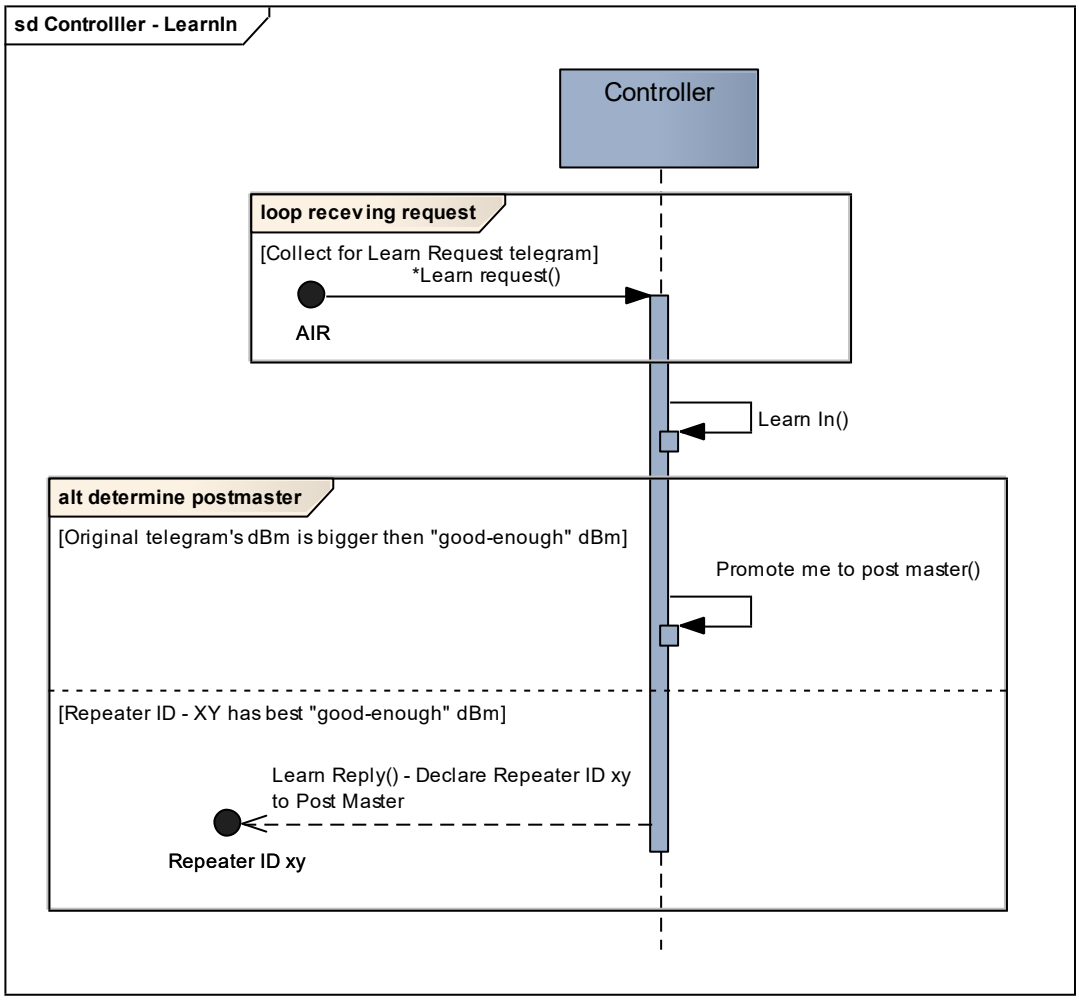


Figure 14 Controller at LearnIn

4.3.2 Controller and Repeater at LearnIn

When several repeaters are participating in the evaluation for the Post master only one gets selected. The generic behavior of a repeater with controller at advanced mode LearnIn is shown in the figure below. The scenario goes in these steps:

- Repeater receives an Learn request
- Enters the RSSI of the received telegram, his ID and the request code
- Repeater sends the altered Learn request
- Controller receives the telegram
- Controller decides on process result and is positive who will be Post Master
- Controller sends Learn reply
- Depending on the destination ID of the learn Reply

— Repeater recognizes it was promoted to Post Master

OR

— Repeater repeats the Learn Reply

Then the Sensor starts the reclaim process of the Learn Acknowledge, this step is not shown in the figure below.

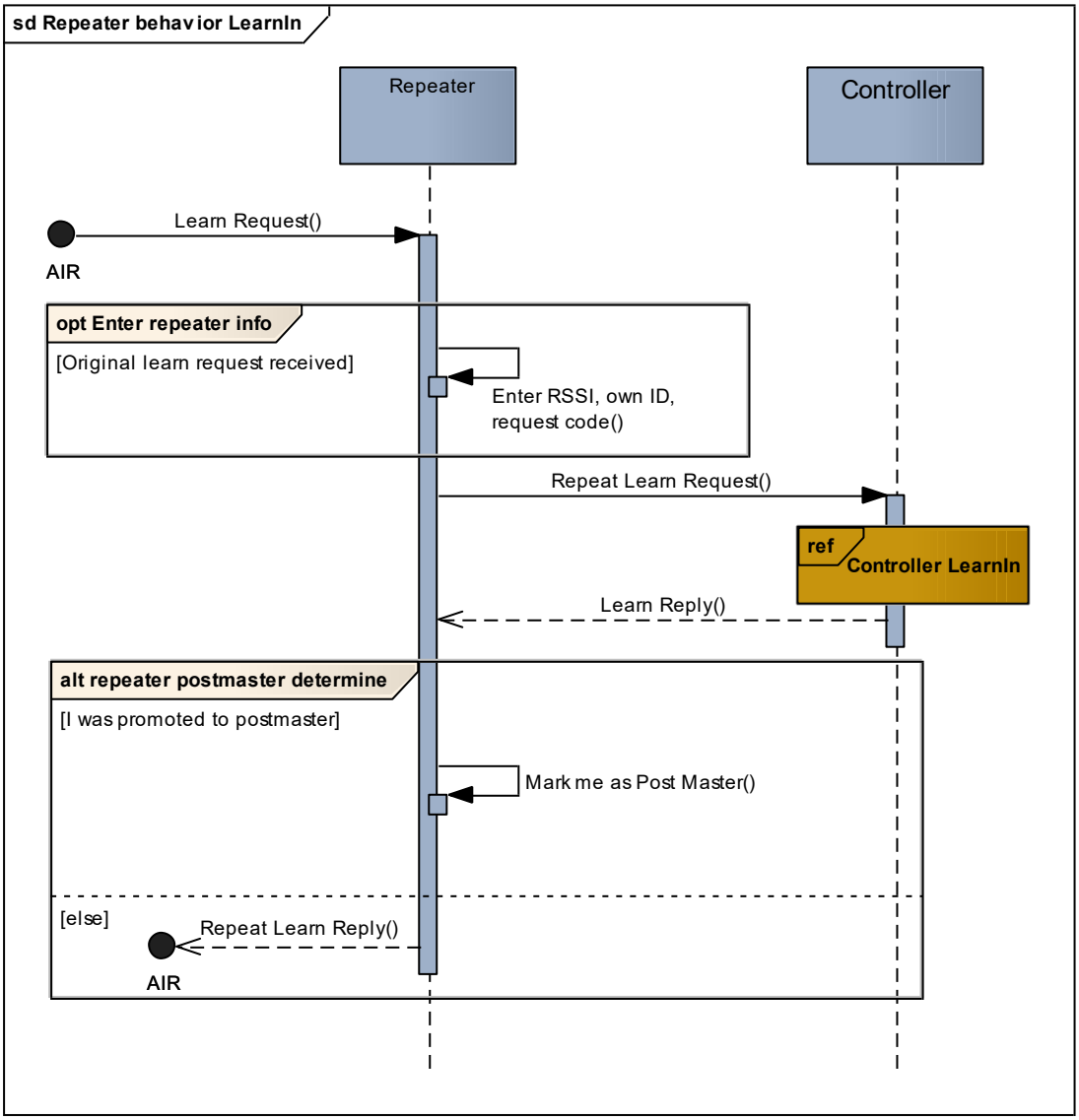


Figure 15 LearnIn with a Controller and a Repeater

4.3.2.1 LearnIn with Controllers after a Postmaster was already declared.

In a scenario with several controllers is mandatory that one and only one controller is the Post Master. Otherwise the system can behave in an unstable way. The premise of only one Poster Master in the system is achieved by learning in only one controller at a time.

Behavior of the already declared post Master is listened in Figure 16. The postmaster was declared by a previous LearnIn process by a different controller. This controller is not participating in the current process.

The already declared Post Master will attend in the election with a request code, which tells the Controller: „I am already Post Master“. The key is that the Controller needs to receive its Learn

Request. Then the already declared Post Master will create a new Mail Box for all the following Controllers.

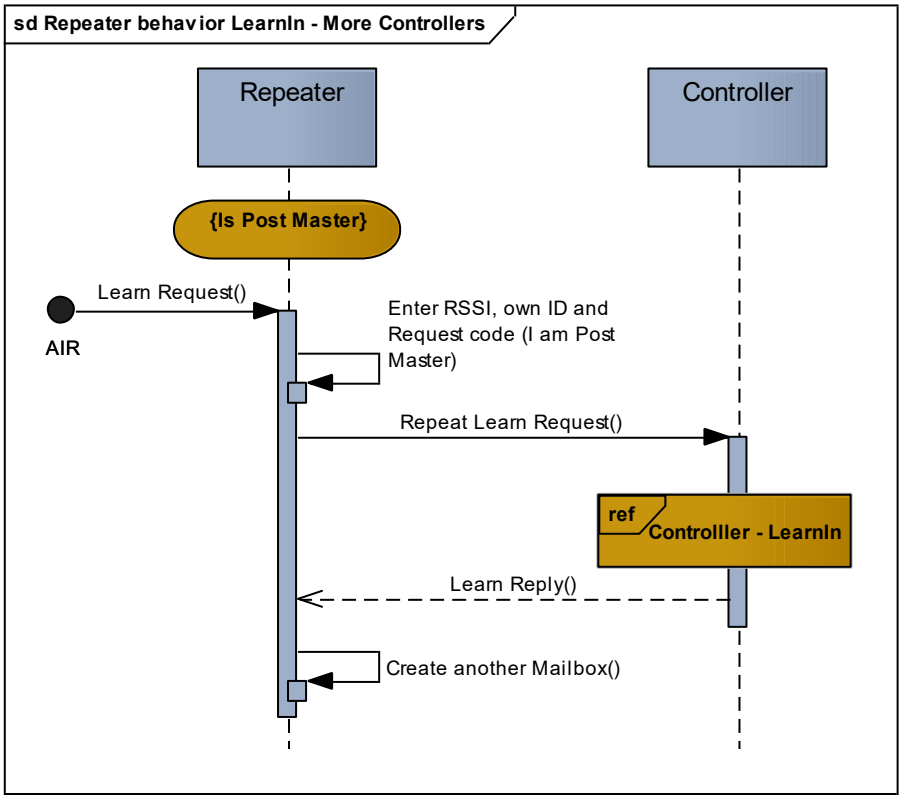


Figure 16 Post Master behavior with next Controller

4.3.2.2 LearnIn with several Controllers.

The scenario of a LearnIn with more than one controllers is shown in two steps. Firstly one controller learns in the sensor and after that the next controller learns in the sensor.

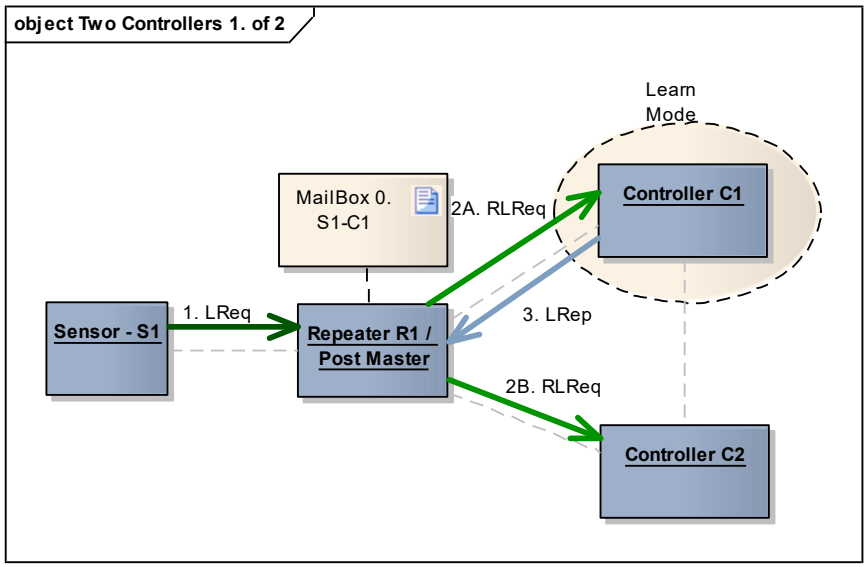


Figure 17 LearnIn with two Controllers. Step 1/2

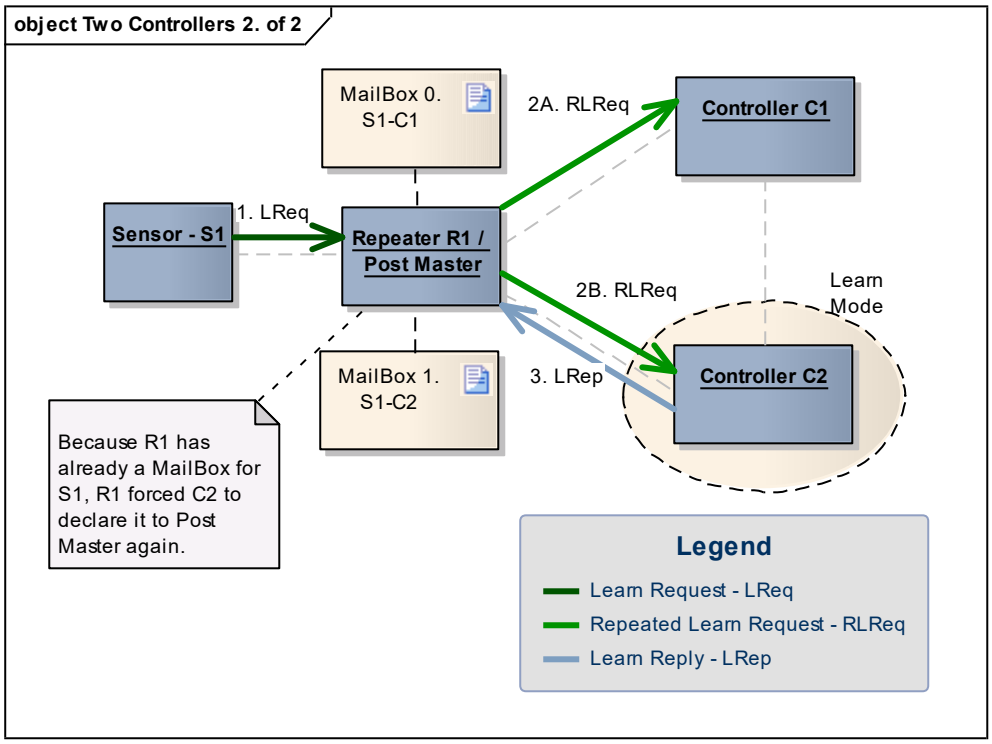


Figure 18 LearnIn with two Controllers. Step 2/2

In the following scenario, the Controller C1, which gets the Learn request first, becomes Post Master. The rule that the Sensor can have only one Post Master must always be respected. Also when there is a suitable candidate with fewer hops, it cannot be selected.

NOTE: Controller and repeater behave in the same way as post master.

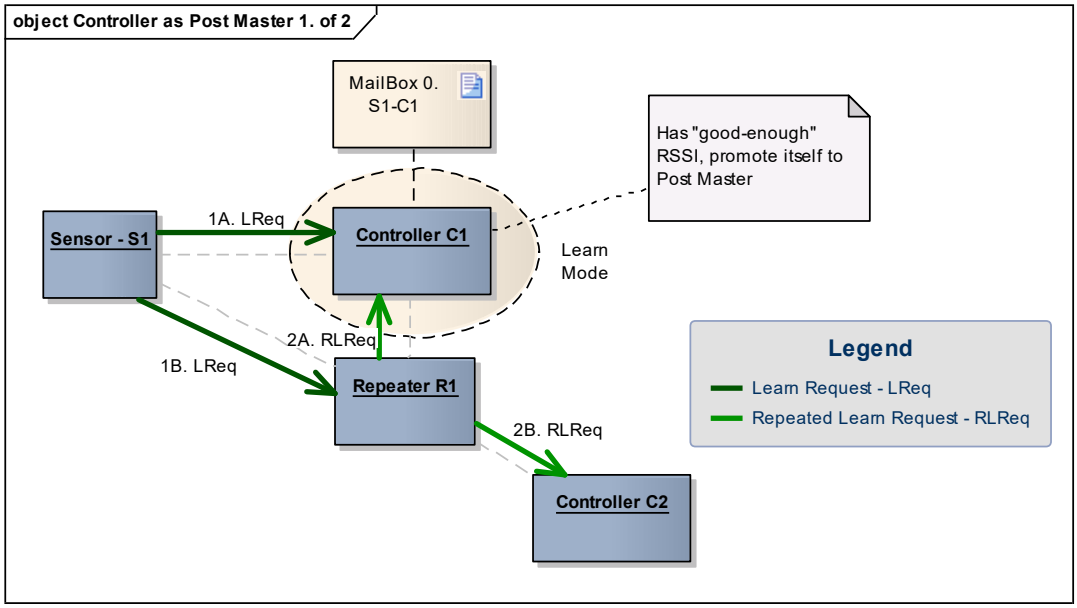


Figure 19 Controller as Post Master 1. of 2

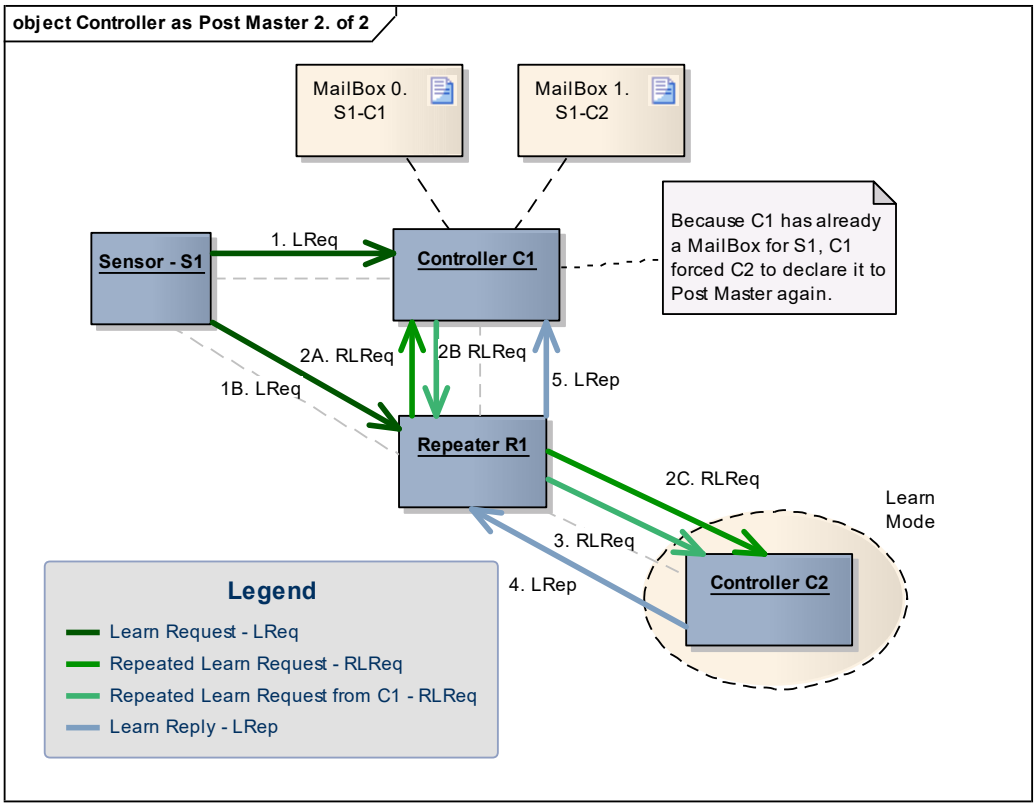


Figure 20 Controller as Post Master 2. of 2

4.4 LearnOut or repeated learn?

If an already learned Sensor sends a Learn request and the Controller is in Learn mode the learning can be interpreted as an LearnOut or as a an repeated LearnIn. The decision is made by the Controller application.

Important is that the repeated LearnIn must be also partial LearnOut, or LearnOut at once. This decision is also made by the Controller application.

After receiving a Learn Reply with LearnOut Acknowledge code the Post Master deletes the Mail Box. It puts the Acknowledge for the Sensor into the temporary Mail Box, because he wants the Sensor to receive this information. If there are no more Mail Boxes from other Controllers, the device gives back the postmastership. When a Controller is Post Master and it learns out this Sensor, it cannot give back postmastership until all Mail Boxes are removed.

A Remote Post Master would behave like in the figure below.

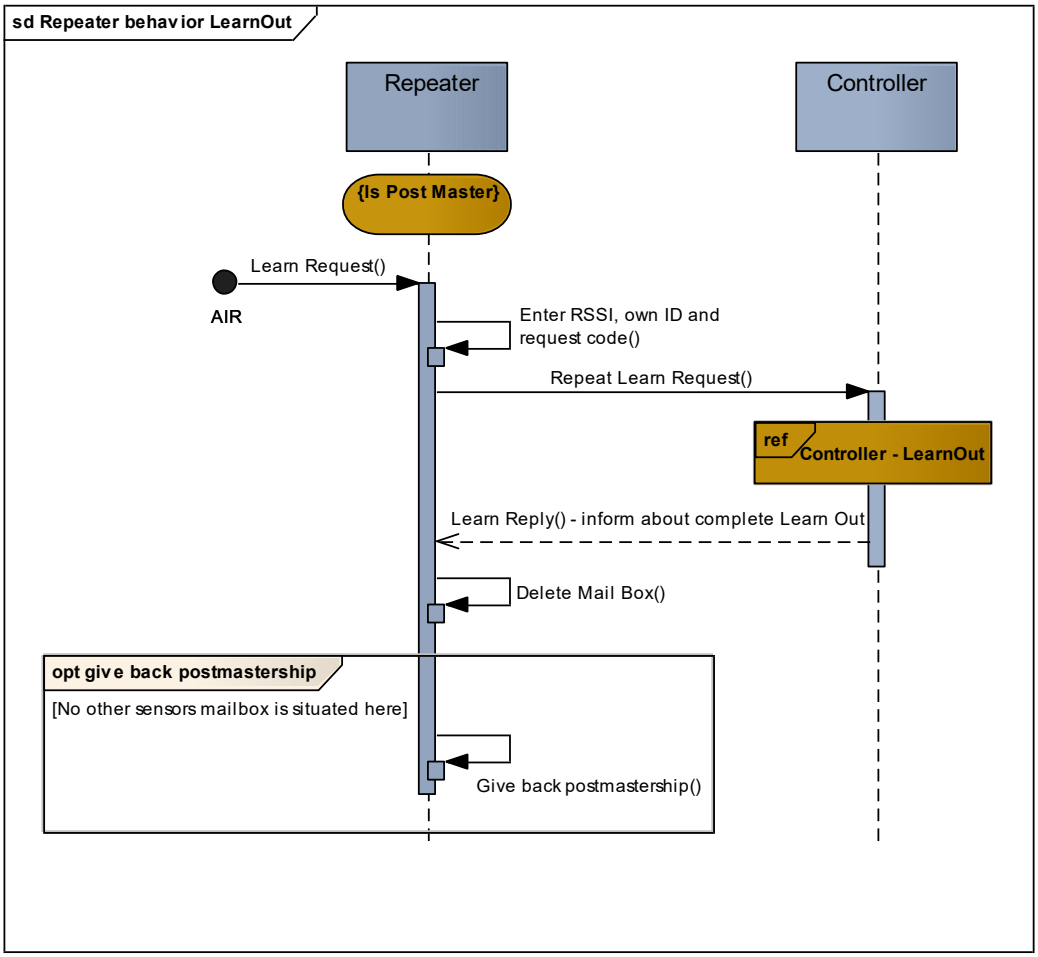


Figure 21 Controller and Repeater at LearnOut

5 Debugging

Since the SMART ACK is a complex communication protocol it is susceptible to failure states. The states are the results of i.e. lost telegrams or non-communicating devices. These states cannot be solved without additional help.

5.1 Debug process

The target is to recover from failure state and restore an optimal operating situation. Debug tools do not automate the debug process. Active user interaction is required. The user tracks down the issue and adjusts settings on the devices. These operations are performed with Remote Management.

These debug possibilities are provided:

- Remote reset process
- Drop Mail Boxes at Post Master via Remote Management.
- Read Mail Box configuration at Post Master with Remote Management.
- LearnOut Sensor in Controller via Remote Management.

- Read LearnIn Sensors on Controller via Remote Management.

Resulting requirements to enable debugging:

- Line powered actors must support Remote Management
- Debug manual
- Debug tool must have graphic user interface

5.1.1 Implementation aspects

5.1.1.1 Repeating of telegrams

Both Reclaim and Acknowledge telegrams are not repeated. They reach their target at first hop, because the Sensor and the Post Master have always a direct link. In the repeating process we use the status field and filters to evaluate if repeating is done or not. To avoid repeating the status value 0x0F signalizes: “Never repeat again”.

5.1.1.2 Post Master repeating

The Declared Post Master must repeat all data telegrams (excluding Reclaim) that come from the Sensor. Otherwise they possibly cannot reach the Controller. So although a SMART ACK actor is not a standard repeater it must repeat the traffic from learned Sensors.

6 Advanced debug issues

This chapter describes examples of issues that can occur at run time. All example configurations are derived from a classic use case where a Sensor is learned on a Controller and has a Mail Box on the Repeater - the Post Master is the repeater.

Issues are rated by their negative influence to the system:

- Feasibility of functions - FOF
- Undesired effects - UE
- Risks - R

6.1.1 Missing Controller

Original Controller is not available (i.e. broken or switched off). We install a new Controller and LearnIn the Sensor. The Post Master is the same as with the original Controller - the repeater. After 6 months we want to move the Sensor.

6.1.2 Issue:

Post Master will not give back postmastership because LearnOut cannot be executed with the not available Controller. Repeater will give back the postmastership only when all Controllers are learned out. This issue can result in problems:

- When the Sensor is at his new location and it is learned in, the repeater will claim the postmastership because it has the old Mail Box from the missing Controller. The issue is the case when the RSSI level (Sensor - Repeater) is no longer good enough.
- When the Sensor at his new location has no direct RF connection to the repeater. The mailbox will be occupied by this Sensor but not used. The resources on Post Master are used non-efficient.

Rating: risk

6.1.3 Solution:

Use remote management to operative solve the conflict and clear the mail box on the Post Master.

The Issue is not presumable to happen often. The use of remote management in this case can demand a more educated and experienced technician.

6.2 Missing Repeater – Post Master

Declared Post Master is not available (i.e. broken or switched off). Sensor sends Data telegrams. If there is another non Post Master repeater the Data is repeated and carried to the Controller. The Controller sends a Data reply but since the Post Master is not available no device will handle the Data Reclaim of the Sensor. Or if there is no other repeater the Controller does not get any data from the Sensor. The Controller therefore does not send a Data reply. We install a new SMART ACK device, replacing the not available Post Master to correct the situation.

6.2.1 Issue:

The new installed device must gain the postmastership. This can be achieved only by repeating the learn process. First all Controllers must LearnOut the Sensor. Then the new device can be selected as Post Master and all Controllers can be learned in again.

Rating: Undesired effects

6.2.2 Solution:

There are more possibilities to start the learn process.

- Manually or with a remote management command switch the Controller to learn mode.
- Manually or with a reset signal trigger LearnIn Sensor (send Learn Request).

It is important to LearnOut the Sensor on all Controllers first and then start LearnIn of the Sensor.

6.3 Missing Sensor

Learned Sensor is not available (i.e. broken or switched off). We want to replace the Sensor. The learn process must be executed with the new Sensor.

6.3.1 Issue:

Learn of new Sensor can fail because actor does not have place for next Mail Box, because it keeps resources for Mail Box for not available Sensor. The old Sensor must be learned Out to free memory resources. The LearnOut cannot be performed in a common way because the old Sensor is not available.

Rating: Feasibility of functions

6.3.2 Solution:

- With remote management tell the Controller to trigger LearnOut.
- With remote management remove Mail Boxes from the Post Master and tell the Controller to LearnOut.

6.4 Missing telegrams

In the system errors can arise when messages get lost (i.e. collisions, the target does not receive telegram). The resulting error state and influence on system depends on which message gets lost and the present state of system. (i.e. Post Master is already declared). In the following text only critical error states and scenarios when messages get lost are described. Situations are separated by current state and actors between which the message gets lost.

Issues are rated by their negative influence to the system:

- Feasibility of functions - FOF
- Undesired effects - UE
- Risks - R

6.4.1 Missing Learn Request

Actors connection:

- Sensor → any non PostMaster Smack Ack actor
- any non PostMaster Smack Ack actor → Controller – only present when advanced learn enabled
- Sensor → Post Master
- Post Master → Controller – only present when advanced learn enabled

States:

- LearnIn with one Controller
- LearnIn with more Controllers (Post Master already declared) – only present when advanced learn enabled
- Repeated LearnIn
- LearnOut with one Controller
- LearnOut with more Controllers – only present when advanced learn enabled
- Partial LearnOut

Table 30 Analysis of missing Learn Request

Connection \ State	LearnIn with one Controller	LearnIn with more Controllers	Repeated LearnIn	LearnOut with one Controller	LearnOut with more Controllers	Partial LearnOut
Sensor → non PM	FOF	FOF	FOF	FOF	N/A	FOF
non PM → Controller	FOF	FOF	FOF	FOF	N/A	FOF
Sensor → PM	N/A	R	FOF	FOF	FOF	FOF
PM → Controller	N/A	R	FOF	FOF	FOF	FOF

Table 31 Missing message issue 01

Message name	Learn Request
Sender	**Sensor**
Receiver	**Post Master**
State	**LearnIn with more Controllers (Post Master already declared)**
Rating	RISK
Description Situation is potential risk to system, but only when more candidates for Post Master have connection to Sensor. When Post Master does not receive Learn Request the Controller gets no information that Sensor has already a Post Master and declares new Post Master. Two Post Masters can cause serious system instability. Sensor and Controller can not recognize this situation although it causes problems. Both devices think that LearnIn was successful. **Problem will occur only at run-time.**	
Detection **Problem will occur only at run-time. Sensor gets Data Acknowledge from more Post Masters.** Prevention Declare stricter “good-enough” RSSI boundaries. Solution	

Force finding Post Master again. (LearnOut all connections)

Table 32 Missing message issue 02

Message name	Learn Request
Sender	**Post Master**
Receiver	**Controller**
State	**LearnIn with more Controllers (Post Master already declared)**
Rating	RISK
Description Situation is potential risk to system, but only when more candidates for Post Master have connection to Sensor. Controller gets no information that Sensor has already a Post Master and declares new Post Master. Two Post Masters can cause serious system instability. Sensor and Controller can not recognize this situation although it causes problems. Both actors think that LearnIn was successful. **Problem will occur only at run-time.**	
Detection **Problem will occur only at run-time. Sensor gets Data Acknowledge from more Post Masters.** Prevention Post Master sends Learn Request more times. Solution Force finding Post Master again. (LearnOut all connections)	

6.4.2 Missing Learn Reply

Actors connection:

- Controller → Post Master – only present when advanced learn enabled

States:

- LearnIn with one Controller
- LearnIn with more Controllers (Post Master already declared) – only present when advanced learn enabled
- Repeated LearnIn
- LearnOut with one Controller
- LearnOut with more Controllers – only present when advanced learn enabled
- Partial LearnOut

Table 33 Analysis of missing Learn Reply

Connection \ State	LearnIn with one Controller	LearnIn with more Controllers	Repeated LearnIn	LearnOut with one Controller	LearnOut with more Controllers	Partial LearnOut
Controller → PM	R	R	UE	R	R	UE

Table 34 Missing message issue 03

Message name	Learn Reply
Sender	**Controller**
Receiver	**Post Master**
State	**LearnIn with one Controller, LearnIn with more Controllers**
Rating	RISK
Description Situation is potential risk to system. The Controller assumes the Sensor is Learned in. But declared Post Master does not get Learn Reply so it does not know that it is Post Master (single Controller) or should create additional MailBox (more Controllers). Sensor has no message about the learn result. Operating will not work, because no Post Master is declared for Sensor Controller relation.	
Detection **Sensor receives no learn acknowledge, can be signalized to user.** Prevention Post Master sends Learn Reply more times. Solution LearnOut and then LearnIn Sensor on Controller.	

Table 35 Missing message issue 04

Message name	Learn Reply
Sender	**Controller**
Receiver	**Post Master**
State	**LearnOut with one Controller, LearnOut with more Controllers**
Rating	RISK

Description Situation is potential risk to system. The Controller assumes the Sensor is learned out. But Post Master does not get Learn Reply so it does not delete the MailBox. Similar Sensor has no message about the learn result. Later LearnIn of Sensor can fail. The actor with the not deleted MailBox will claim the postmastership for him. If location of actors has changed finding Post Master is corrupted. Also the non deleted MailBox occupies memory space and the actor can run out of MailBoxes.
Detection **Sensor receives no learn acknowledge, can be signalized to user.** Prevention Post Master sends Learn Reply more times. Solution LearnOut and then LearnIn Sensor on Controller.

6.4.3 Missing Learn Reclaim

Actors connection:

- Sensor → Post Master

States:

- LearnIn with one Controller
- LearnIn with more Controllers (Post Master already declared) – only present when advanced learn enabled
- Repeated LearnIn
- LearnOut with one Controller
- LearnOut with more Controllers – only present when advanced learn enabled
- Partial LearnOut

Table 36 Analysis of missing Learn Reply

Connection \ State	LearnIn with one Controller	LearnIn with more Controllers	Repeated LearnIn	LearnOut with one Controller	LearnOut with more Controllers	Partial LearnOut
Sensor→ PM	R	R	UE	UE	FOF	UE

6.4.4 Missing Learn Acknowledge

Actors connection:

- Post Master → Sensor

States:

- LearnIn with one Controller
- LearnIn with more Controllers (Post Master already declared) – only present when advanced learn enabled
- Repeated LearnIn
- LearnOut with one Controller
- LearnOut with more Controllers – only present when advanced learn enabled
- Partial LearnOut

Table 37 Analysis of missing Learn Acknowledge

Connection \ State	LearnIn with one Control-ler	LearnIn with more	Repeated LearnIn	LearnOut with one Controller	LearnOut with more Controllers	Partial LearnOut
PM → Sensor	R	R	UE	UE	FOF	UE

Table 38 Missing message issue 05

Message name	Learn Request, Learn Acknowledge
Sender	**Sensor, Post Master**
Receiver	**Post Master, Sensor**
State	**LearnIn with one Controller, LearnIn with more Controllers**
Rating	RISK
Description Situation is same if Learn Reclaim or Learn Acknowledge gets lost. Situation is potential risk although Post Master is declared and the actor knows about it. The Sensor does not know it is LearnIn in a Controller and in Operating mode does not reclaim its MailBoxes. Operating will fail.	
Detection **Sensor receives no learn acknowledge, can be signalized to user.** Prevention Declare stricter “good-enough” RSSI boundaries. Solution LearnOut and then LearnIn Sensor on Controller. Sensor application can by default after every LearnIn with no Learn Acknowledge try to reclaim supported amount of Mail-Boxes. (in most Sensors the amount is 1)	

6.4.5 Missing Data, Data Reply, Data Reclaim, Data Acknowledge

Losing one of the Data messages is not a threat for SMART Acknowledge as communication protocol. The user application determines the negative influence to the system when one of the Data, Data Reply, Data Reclaim or Data Acknowledge messages gets lost. Basically by determination of stricter “good-enough” RSSI values and good radio infrastructure planning should be secured that Data telegrams get lost only occasionally.