

Smart Acknowledge September 15, 2010



Smart Acknowledge (SmartACK) V1.4**Smart Acknowledge (SmartACK) V1.4****REVISION HISTORY**

The following major modifications and improvements have been made to the first version of this document:

No	Major Changes	Date	Who
1.1	Changed the candidate priority evaluation.		
1.2	Changed shortcuts for telegrams		
1.3	Added criteria within candidate evaluation		
1.4	Exported document from system spec	15.10.2010	ASt

**Published by EnOcean GmbH, Kolpingring 18a, 82041 Oberhaching, Germany
www.enocean.com, info@enocean.com, phone ++49 (89) 6734 6890**

© EnOcean GmbH
All Rights Reserved

Important!

This information describes the type of component and shall not be considered as assured characteristics. No responsibility is assumed for possible omissions or inaccuracies. Circuitry and specifications are subject to change without notice. For the latest product specifications, refer to the EnOcean website: <http://www.enocean.com>.

As far as patents or other rights of third parties are concerned, liability is only assumed for modules, not for the described applications, processes and circuits.

EnOcean does not assume responsibility for use of modules described and limits its liability to the replacement of modules determined to be defective due to workmanship. Devices or systems containing RF components must meet the essential requirements of the local legal authorities.

The modules must not be used in any relation with equipment that supports, directly or indirectly, human health or life or with applications that can result in danger for people, animals or real value.

Components of the modules are considered and should be disposed of as hazardous waste. Local government regulations are to be observed.

Smart Acknowledge (SmartACK) V1.4

1 Introduction

Smart Ack enables bidirectional communication. The communication is managed by a Controller that responds to the devices telegrams with acknowledges. It can work with also with non Smart Ack devices. Scenario with HVAC as Controller is shown in figure below.

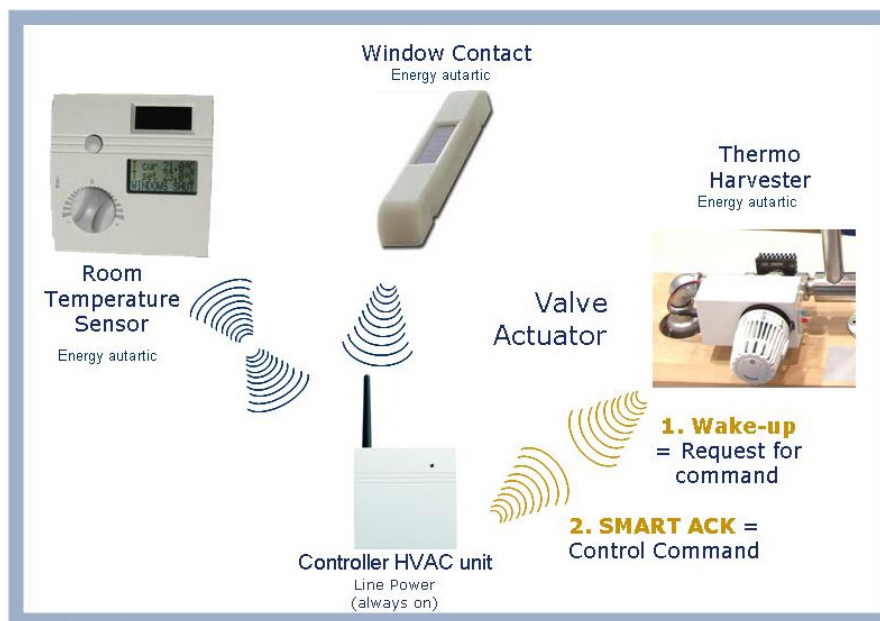


Figure 1 Smart Ack scenario

1.1 Functional description

Smart Ack is a bidirectional communication protocol between two actors. One actor is an energy autarkic Sensor, the second is a line powered Controller. Communication protocol in direction from Sensor to Controller is as it was defined before in non Smart Acknowledge devices. Smart Acknowledge describes communication in direction from Controller to autarkic Sensor. This leads us to the main challenge of Smart Acknowledge design:

- Receive mode consumes high amount of energy

To keep the receive mode on Sensor as short as possible we use message synchronization (Message flow is performed in predefined time intervals). A sensor sends its data and expects the answer telegram in a predefined very short time slot. In this time Sensor's receiver is active. For this purpose we declare Mail Boxes. Mail Box is like a letter box for Sensor. Telegrams from Controllers are collected into the Mail Box. Sensor can reclaim telegrams that are in his Mail Box.

Communication through repeaters can cause unknown time delays and synchronization will not work. Therefore the second challenge of Smart Acknowledge is:

- Unknown time delays are introduced by repeaters

Smart Acknowledge (SmartACK) V1.4

To eliminate unknown delays Mail Boxes are established at line powered devices with direct contact to autarkic device. This device is called Post Master. Post Master manages all Sensors Mail Boxes. Post Master can become Controller itself 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:

1.2.1 Post Master.

The Smart Ack attribute describe they ability to operate in Smart ACK networks. Actors are real devices that can take roles. Repeater and Controller can take the role of Post Master. The Smart Ack operations are same on Controller and Repeater, but Controller can additionally Learn Sensors and Repeater not.

1.2.2 Smart Ack Sensor

Smart Ack Sensor is an autarkic device that sends measured data and reclaim acknowledges. It can be learn 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 infinite loop.

1.2.3 Post master

Post Master is the possibly nearest Smart Ack actor to the Sensor. Post Master can become Repeater or Controller. They behave in same way as Post Master. Post Master holds Sensors Mail Boxes.

To Smart Ack activities belong:

- handle reclaims.
- capture Data send to Sensors and put them to its Mail Box

1.2.4 Controller

Controller is a line powered device. It can have back bone connection and beside Smart Ack also other functionalities. Controller can learn Sensors. Controller can take the role of Post Master. It processes Sensors Data and sends Data back to sensor.

Smart Acknowledge (SmartACK) V1.4**1.2.5 Repeater**

Repeater is a line powered device. It can take the role of Post Master. Besides Smart Ack it can have other functionalities i.e. light actuator.

The functionality of Smart Ack and repeater is not connected. One of the can be turn off without affecting the another.

1.3 Reclaim process

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

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

1.3.1 Implementation aspects

The reclaim process of acknowledges from Post Master to Sensor looks like this:

Sensor sends initial telegram. Then the reclaim process starts.

Sensor wait the response period

This time is for system to process the initial message and prepare an acknowledge

a) Sensor sends reclaim telegram.

b) Sensor wait the minimum reclaim period

This time is for Post Master to process reclaim and start sending of acknowledge.

c) Sensors turn on the receiver.

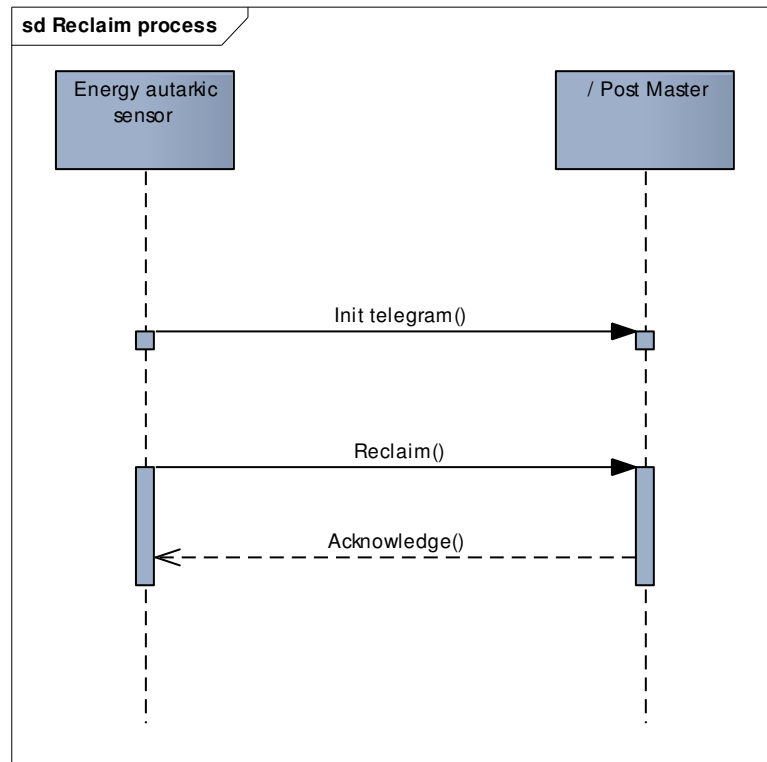
d)

— 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.

Smart Acknowledge (SmartACK) V1.4

1) Initial telegram can be Data or a Learn Request telegram.

Figure 2 Reclaim process

The time period of Sensors active receive mode is called actual reclaim period and it can vary.

The response period is different in learning and operating process:

- Learning uses Standard response period
- Operating uses response period that has been determinate by Controller.

1.3.1.1 Mail Box

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

- Temporary Mail Box

Mail Box is used in learning. Learn Acknowledge is stored in this Mail Box. There is only one temporary mail box on a Smart ack device.

- Normal Mail Box

Mail Box is used in Operating. Every Sensor has its own Mail Box. In case Sensor is learned by more Controllers there can be more Mail Boxes for this sensor. These Mail

Smart Acknowledge (SmartACK) V1.4

Boxes are separated by Mail Box index. Indexes are delegated by Post Master. For every Sensor Controller relation there can be only one Mail Box.

Post Master would send already reclaimed acknowledge twice. Instead Post Master signalizes to Sensor that its Mail Box is empty. Post Master also signalizes to Sensor when it tries to reclaim non existing Mail Box.

Table 1 Post Master signals

Name	Description
Mail Box empty	Send when Post Master has no new Acknowledge for Sensor.
Mail Box does not exists	Send by Sensors already declared Post Master. In case that Sensor tries to reclaim in operating mode a Mail Box with non existing Mail Box index

If reclaim process fails, 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.

NOTE: Other than Smart Ack devices will notice the reclaim attempts as one telegram.

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

Smart Acknowledge (SmartACK) V1.4**2 Functional modes**

Smart Acknowledge functionality is different in Learning and Operating.

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

Table 2 Smart ack telegrams

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

2.1 Learning

Learning is a process where device exchange information about each other. It can result in Learn In or Learn Out. Learning with more controllers is also enabled. It is explained 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.

- e) The autarkic Sensor is switched to learn mode by doing so it sends a “Learn Request” telegram.
- f) Post Master gets determined.
- g) Controller decides if the learning is a Learn In or Learn Out.
- h) Sensor reclaims Learn Acknowledge.

2.1.1 Determine Post Master

In determine Post Master process we try to find the nearest Smart Ack device that can hold Sensors Mail Boxes. It can be Controller itself or a Repeater. Based on this in Smart Ack we differ between two actors positioning in field:

Sensor and Controller have direct radio contact, Controller is Post Master – simple mode

- i) Sensor and Controller do not have direct radio contact, the contact is enabled by repeaters, Controller is not Post Master – advanced mode

The priority in determining Post Master is to establish simple mode. 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 promoted by Controller according a hierarchically system. Sensor can have only one Post Master. The process of finding Post

Smart Acknowledge (SmartACK) V1.4

Master repeated in every Learning. The Post Master gives back the postmastership when the Sensor is Learned Out from all controllers.

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

NOTE: Reply telegrams contain the Acknowledge.

2.1.1.1 Implementation aspects

To determine if there is a direct radio contact we use the Learn Request telegram. The telegram is send by Sensor in Learning. When device receives telegram we declare the radio contact is available. To ensure good operating result the radio signal strength index – RSSI must be at a certain level. We call this level “good enough RSSI”

When controller receives Learn Request it automatically promotes itself to Post Master. Repeaters must participate as candidate in determining Post Master. More repeaters can participate at once. Repeaters enter their information into the Learn Request telegram and repeat it. Controller collects all these telegrams and promotes one candidate to Post Master. Collecting starts by the first Learn Request telegram and lasts for a short time. This is called the Learn Request period.

Repeater information entered into Learn Request data field:

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

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

Table 3 Request code

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

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

The postmaster is eliminated by a hierarchically priority system. Every attended candidate gets a priority. Controller participates also as candidate. The candidate with highest priority is than elected.

The hierarchical priority system is determined by these criteria:

- if candidate is already post master (detected from Request code)
- j) if candidate has place for next mailbox (detected from Request code)

Smart Acknowledge (SmartACK) V1.4

- k) if RSSI from candidate to sensor is good enough
- l) if the candidate is the local device or remote

Table 4 Priorities

Priority	1. PM	2. PLACE OK	3. RSSI OK	4. Local/ Remote
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

If criteria is TRUE = 1 or FALSE = 0. Local candidate = 1 and Remote candidate = 0. First criteria is more important than the second etc.

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

with lowest hop (device) count between controller and candidate

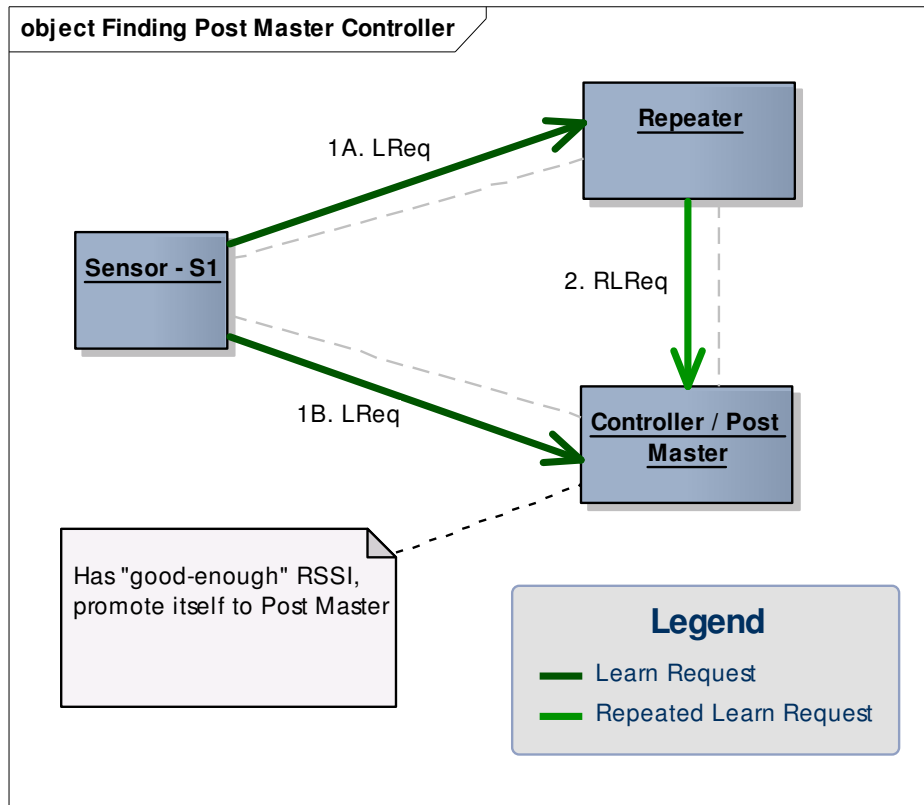
- m) if same hop count then with the highest “good-enough” RSSI value

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

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

Finding Post Master scenario that results in simple mode is on Picture bellow.

Smart Acknowledge (SmartACK) V1.4

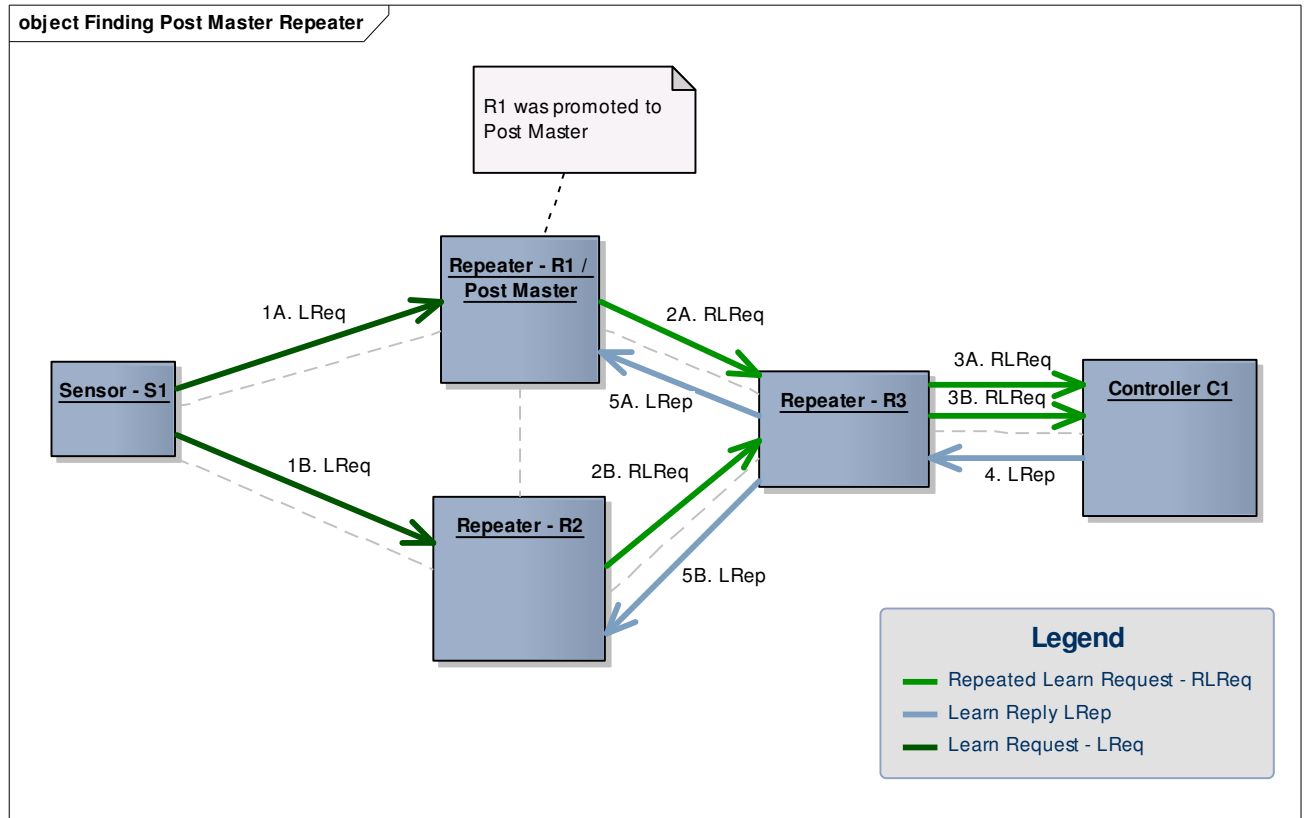


1) Controller collects the original and repeat Learn Request and then promotes itself to Post Master.

Figure 3 Finding Post Master – simple mode

Finding Post Master that results in advanced mode is on picture bellow.

Smart Acknowledge (SmartACK) V1.4



- R1 has better RSSI of received Learn Request
- R1 and R2 participates on Post Master determination. R1 is promoted to Post Master.

Figure 4 Finding Post Master – advanced mode

2.1.2 Learn In and Learn Out

Controller decides if the ongoing learning operation results in:

Learn In

Is performed when controller wants to create a relation with Sensor.

- 1) Successful Learn In
- 2) Failed Learn In

n) Learn Out.

Is performed when controller wants to drop existing relation with Sensor.

This decision is mainly based on:

- If Sensor is already learned by Controller.
- If Controller allows repeated learning of one sensor.
- If the controller accepts Sensors EEP
- Technical abilities of connection (Post Master candidate above 6 included is found)

Smart Acknowledge (SmartACK) V1.4

NOTE: The decision is made by the user application and there can be also more user defined criteria.

In Learn In Controller determines the response time that is used in reclaim process in Operating mode. The information is included in Acknowledge to Sensor.

2.1.2.1 Implementation aspects

The information about Controller decision is included in controllers answer to sensor. It is the Acknowledge code.

Table 5 Acknowledge code

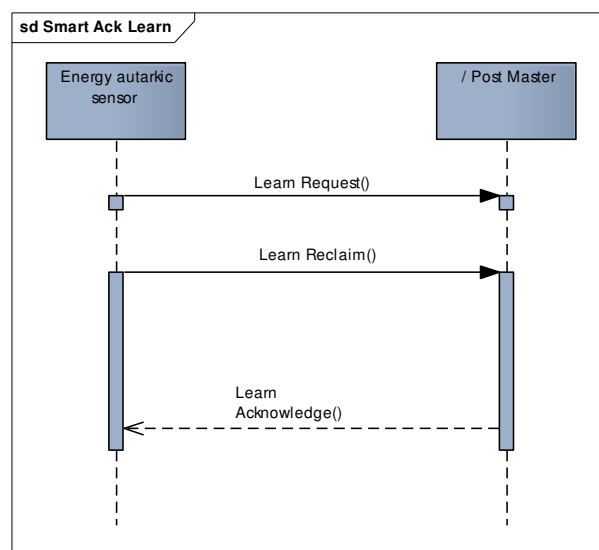
Acknowledge code	Description	Post Master action	Sensor interpretation
0x00	First Learn In successful	Create Mail Box.	Create Mail Box information.
0x01 – 0x0F	Repeated Learn In.	-	Application specific.
0x10 – 0x1F	Failed Learn In	-	Application specific
0x20	Complete Learn Out.	Drop Mail Box.	Delete Mail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

According the result Post Master creates or drops a Mail Box. In case of repeated learn of one Sensor the existing Mail Box is used. Repeated Learn In can be partially Learned out or complete Learn out at once.

When Sensor gets learned by more Controllers always the same Post Master gets elected. For every controller a new Mail Box is created at Post Master. Post Master indexes these Mail Boxes by Controllers. Then in Learning the Post Master informs the Sensor about the actual Mail Box index.

2.1.3 Learning Scenario

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



Smart Acknowledge (SmartACK) V1.4

Figure 5 Learning – Simple mode

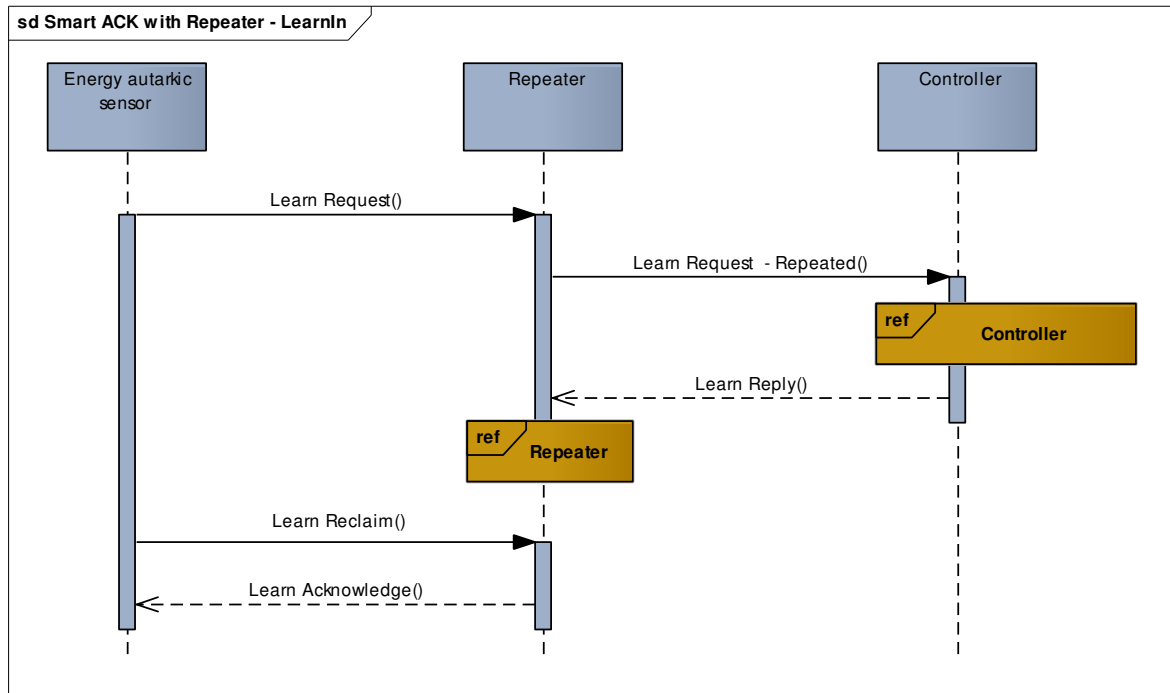


Figure 6 Learning - Advanced mode

Smart Acknowledge (SmartACK) V1.4**2.2 Operating**

Operating is the common data transfer performed. It goes in these steps:

Sensors sends data telegram.

- o) Controller prepares Data Acknowledge telegram.
- p) Sensor reclaims Data Acknowledge.

In Operating 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.

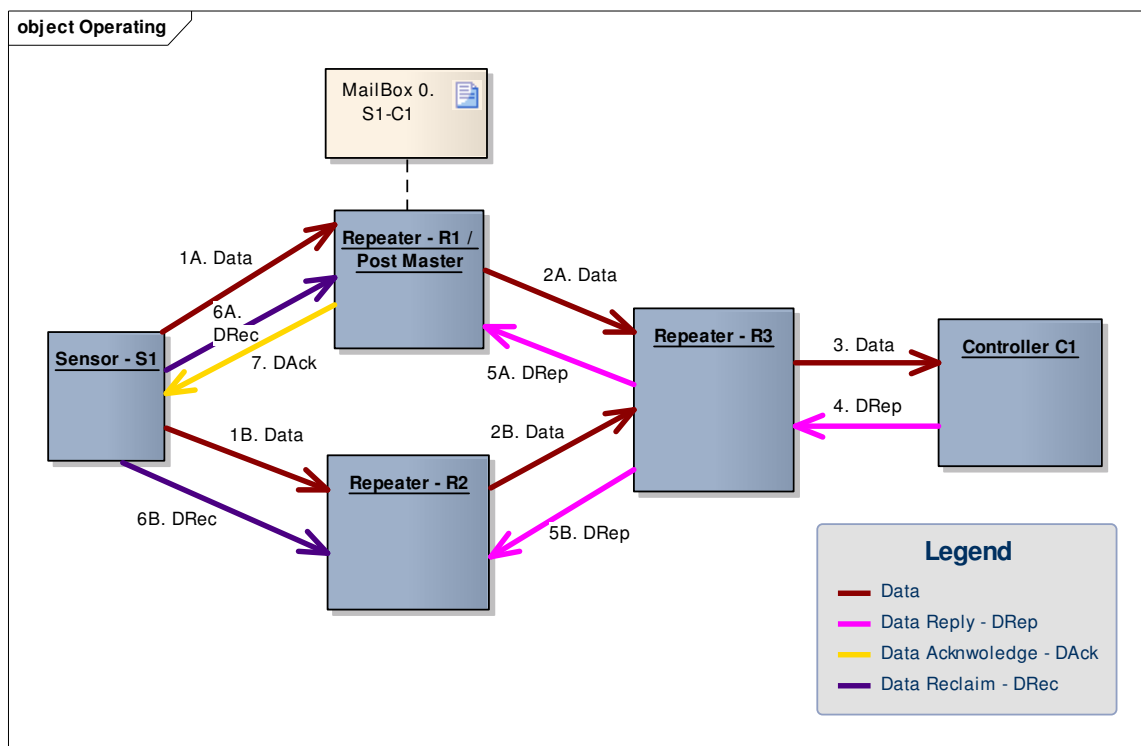
2.2.1 Implementation aspects

After sending Data telegram sensor waits the Response time and then starts reclaim process. In this time the Controller has to prepare an Acknowledge and place it on Post Master. This period was declared by controller in Learning before. In order it is as short as possible.

In data reclaim process the Sensor identifies with the Mail Box index which of its Mail Boxes it wants to reclaim. With repeating the data reclaim process sensor can reclaim all of its Mail Boxes one by one.

NOTE: Sensor has more Mail Boxes when it is learned by more Controllers.

The message flow in Operating mode is on picture below.



Smart Acknowledge (SmartACK) V1.4

1) Data Reply and Data Acknowledge have different status bytes, besides this they are same telegrams.

Figure 7 Operating message flow

2.2.2 Operating Scenario

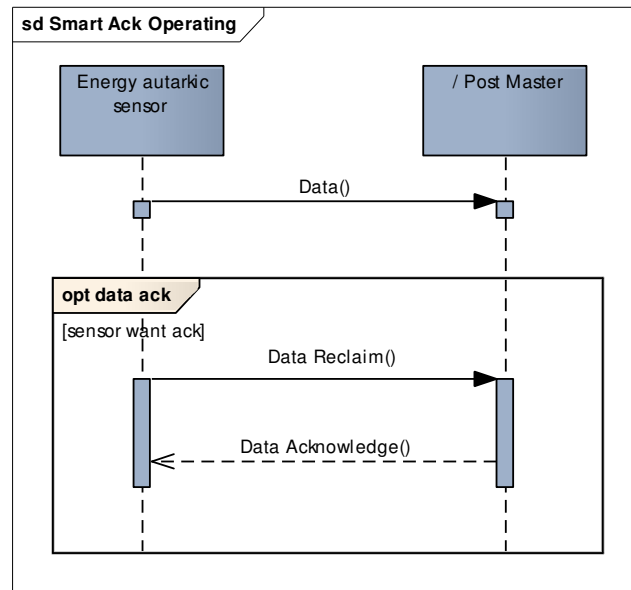


Figure 8 Operating – simple mode

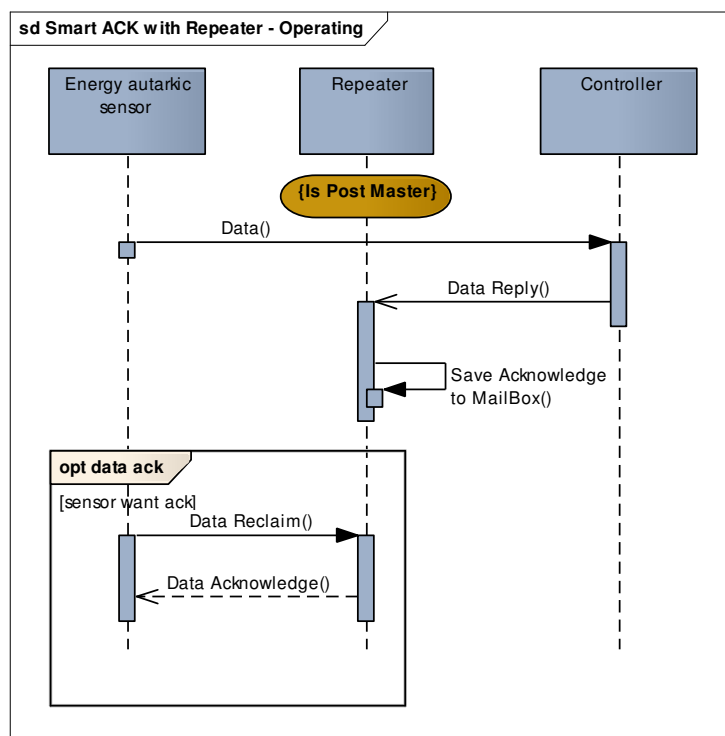


Figure 9 Operating – advanced mode

2.2.3 Remote reset

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

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

2.2.3.1 Implementation aspects

The idea is to switch the reset signal telegram for a Data Acknowledge 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 Remote Management or Controller. Controller can direct swap the Reset signal for the Data Acknowledge. When Controller is not available Remote Management call a function on Post Master to swap telegrams. When Post Master is unavailable the swap is performed by Remote Management.

Reset scenario where Controller sends the Reset signal is Figure below.

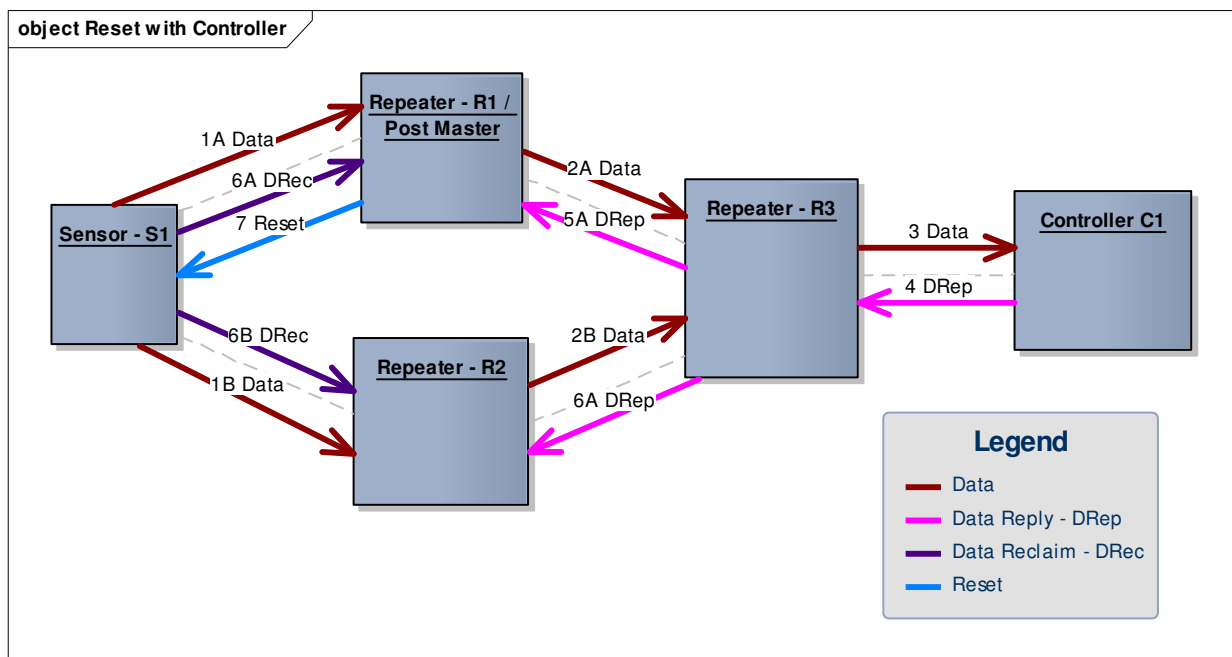


Figure 10 Reset scenario with Controller

Scenario with Remote Manager and broken Post Master is shown on Figure below.

Smart Acknowledge (SmartACK) V1.4

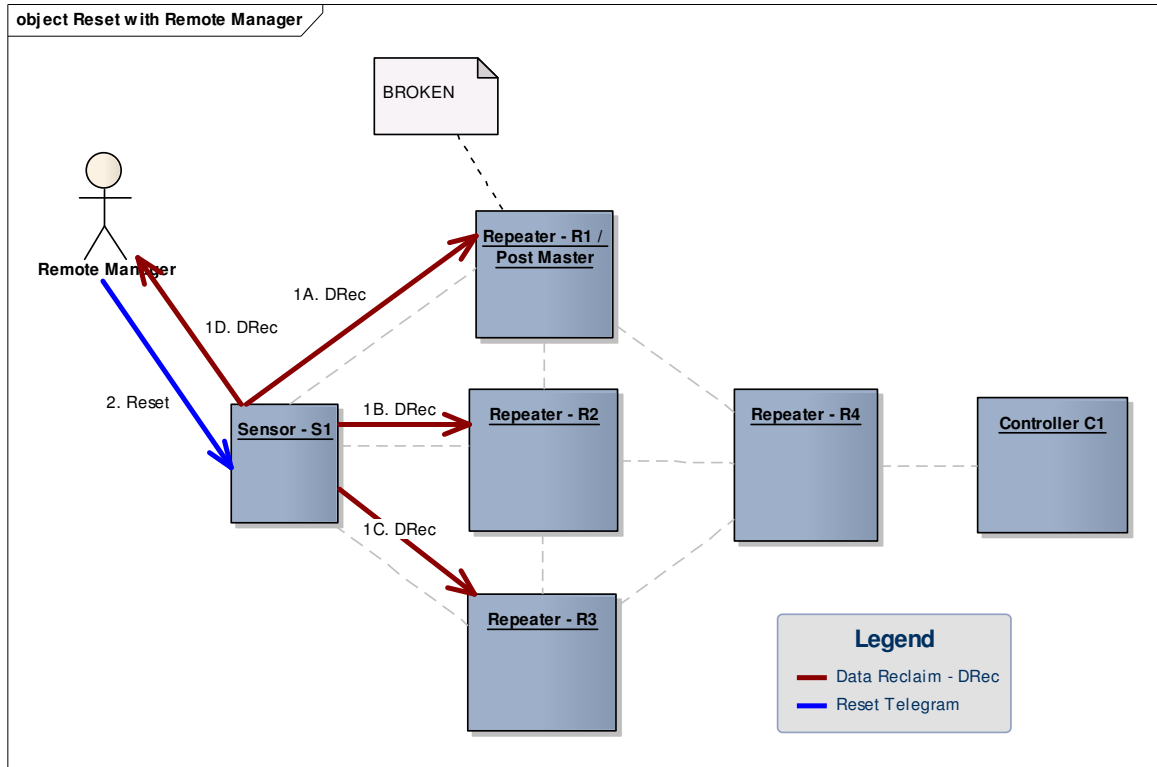


Figure 11 Reset Learn with Remote Manager

Smart Acknowledge (SmartACK) V1.4**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, 9BS)

NOTE: For more details see general telegram description.

Telegrams described in chapter functional description, are implemented by one of these telegrams.

Table 6 Smart Ack telegrams overview

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

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

Table 7 Message indexes

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

Smart Acknowledge (SmartACK) V1.4

0x01	Mail Box Empty	Signal telegram
0x02	Mail Box does not exist	Signal telegram
0x03	Reset	Signal telegram

3.1.1 Flag codes

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

Table 8 Request code

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

Table 9 Acknowledge code

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

Smart Acknowledge (SmartACK) V1.4

3.1.2 Learn Request

Table 10 Learn request description

Name	Learn Request
Used telegram	Smack Learn Request Telegram
Telegram Choice	0xC6
Message index	N/A
Data length	10 bytes
Data content	<div>Request code 5 bits</div> <div>Manufacturer ID 11 bits</div> <div>EEP (EnOcean Equipment Profile) 3 bytes</div> <div>RSSI [dBm] 1 byte</div> <div>Repeater ID 4 bytes</div>
Send with sub-telegram count	3
Repeated	no*
Send by	Sensor
Addressed to	N/A

*NOTE: Sensor sends learn request with Status 0x0F. But smack devices alter telegram and send it.

Table 11 Learn request structure

	7	6	5	4	3	2	1	0
0	0xC6							
1	REQUEST CODE							
2	MANUFACTURER ID							
3	EEP							
4								
5								
6	RSSI							
7	REPEATER ID							
8								
9								
10								
11	SENSOR ID							
12								
13								
14								
15	0x0F							
16	CHCK							

Smart Acknowledge (SmartACK) V1.4

3.1.3 Learn Reply

Table 12 Learn reply description

Name	Learn Reply		
Used telegram	Smack Learn Answer Telegram		
Telegram Choice	0xC7		
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 structure

	7	6	5	4	3	2	1	0
0	0xA6							
1	0xC7							
2	0x01							
3	RESPONSE TIME							
4								
5	ACKNOWLEDGE CODE							
6								
7	SENSOR ID							
8								
9								
10								
11	POST MASTER ID							
12								
13								
14								
15	CONTROLLER ID							
16								
17								
18								
18	0x00							
19	CHCK							

Smart Acknowledge (SmartACK) V1.4**3.1.4 Learn Acknowledge****Table 14 Learn Acknowledge description**

Name	Learn Acknowledge		
Used telegram	Smack Learn Answer Telegram		
Telegram Choice	0xC7		
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		

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

Table 15 Learn acknowledge structure

	7	6	5	4	3	2	1	0
0	0xA6							
1	0xC7							
2	0x02							
3	RESPONSE TIME							
4								
5	ACKNOWLEDGE CODE							
6	MAIL BOX INDEX							
7	NOT USED							
8								
9								
10	POST MASTER ID							
11								
12								
13								
14	CONTROLLER ID							
15								
16								
17								
18	0x0F							
19	CHCK							

Smart Acknowledge (SmartACK) V1.4**3.1.5 Learn Reclaim****Table 16 Learn reclaim description**

Name	Learn Reclaim
Used telegram	Smack Reclaim Telegram
Telegram Choice	0xA7
Message index	0b0
Data length	0 bytes
Send with sub-telegram count	1
Repeated	no
Send by	Sensor
Addressed to	N/A

Table 17 Learn reclaim structure

	7	6	5	4	3	2	1	0
0	0xA7							
1	0b0	NOT USED						
2	SENSOR ID							
3								
4								
5								
6	0x0F							
7	CHCK							

Smart Acknowledge (SmartACK) V1.4

3.1.6 Data Reclaim

Table 18 Data reclaim description

Name	Data Reclaim
Used telegram	Smack Reclaim Telegram
Telegram Choice	0xA7
Message index	0b1
Data length	7 bits
Data content	Mail Box index 7 bits
Send with sub-telegram count	1
Repeated	no
Send by	Sensor
Addressed to	N/A

Table 19 Data reclaim structure

	7	6	5	4	3	2	1	0
0	0xA7							
1	0b1	MAIL BOX INDEX						
2	SENSOR ID							
3								
4								
5								
6	0x0F							
7	CHCK							

Smart Acknowledge (SmartACK) V1.4**3.1.7 Mail Box empty message****Table 20 Mail Box empty description**

Name	Mail Box empty
Used telegram	Signal Telegram
Telegram Choice	0xD0
Message index	0x01
Send with sub-telegram count	1
Repeated	no
Send by	Post Master :: {Controller, Repeater}
Addressed to	Sensor

Table 21 Mail Box empty structure

	7	6	5	4	3	2	1	0
0	0xA6							
1	0xA7							
2	0x01							
3	SENSOR ID							
4								
5								
6								
7	POST MASTER ID							
8								
9								
10								
11	0x0F							
12	CHCK							

Smart Acknowledge (SmartACK) V1.4**3.1.8 Mail Box does not exist****Table 22 Mail Box does not exists description**

Name	Mail Box does not exists
Used telegram	Signal Telegram
Telegram Choice	0xD0
Message index	0x02
Send with sub-telegram count	1
Repeated	no
Send by	Post Master :: {Controller, Repeater}
Addressed to	Sensor

Table 23 Mail Box does not exists structure

	7	6	5	4	3	2	1	0
0	0xA6							
1	0xA7							
2	0x02							
3	SENSOR ID							
4								
5								
6								
7	POST MASTER ID							
8								
9								
10								
11	0x0F							
12	CHCK							

Smart Acknowledge (SmartACK) V1.4**3.1.9 Reset****Table 24 Reset description**

Name	Reset
Used telegram	Signal Telegram
Telegram Choice	0xD0
Message index	0x03
Send with sub-telegram count	1
Repeated	no
Send by	Controller, Post Master :: {Controller, Repeater}, Remote Device
Addressed to	Sensor

Table 25 Reset structure

	7	6	5	4	3	2	1	0
0	0xA6							
1	0xA7							
2	0x03							
3	SENSOR ID							
4								
5								
6								
7	SENDER ID							
8								
9								
10								
11	0x0F							
12	CHCK							

Smart Acknowledge (SmartACK) V1.4

3.1.10 Data

Table 26 Data description

Name	Data
Used telegram	Common EnOcean telegram
Telegram Choice	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 27 Data reply description

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

3.1.12 Data acknowledge

Table 28 Data acknowledge description

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

Smart Acknowledge (SmartACK) V1.4

4 Smart Ack case studies

Here we want to show and discuss some interesting case studies of Smart Acknowledge. They can help to understand the protocol and gain complex knowledge about it.

4.1 Field installations

Smart Acknowledge can be used in building and also in home automation. In figure below is pictured a building automation scenario. More Sensors can be learned in one gateway. Repeater is only used when the direct connection to gateway is not good enough.

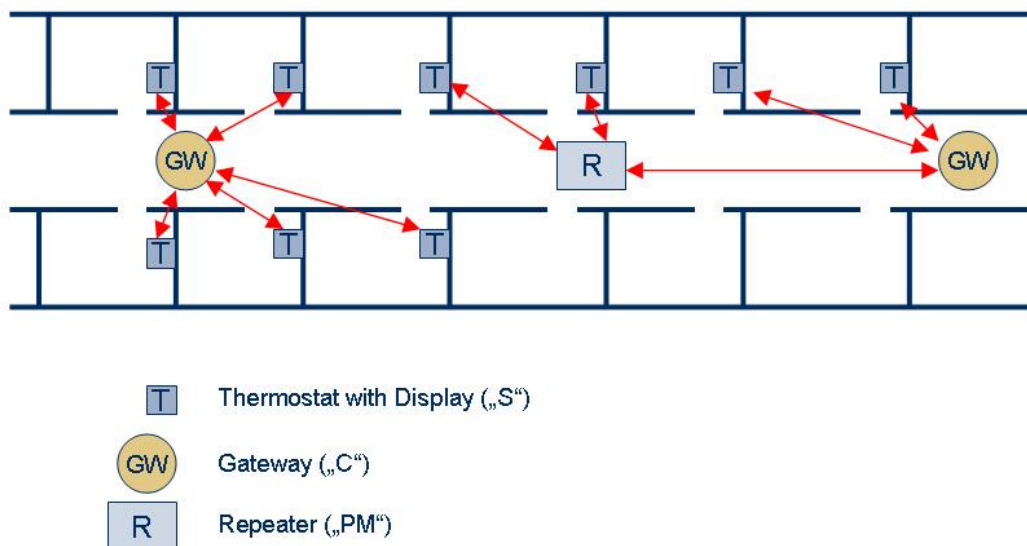
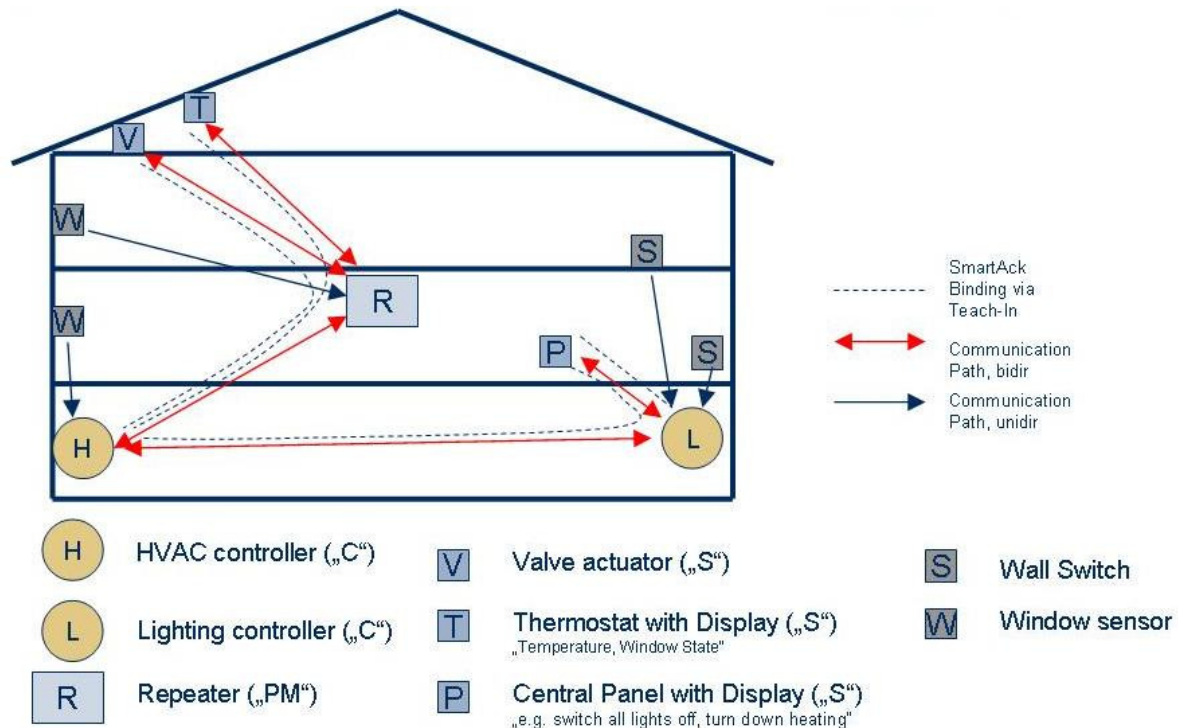


Figure 12 Building automation scenario

In figure below is a home automation scenario. Autarkic modules like 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.

Smart Acknowledge (SmartACK) V1.4**Figure 13 Home automation scenario****4.2 Operating notes****4.2.1 Does a Post Master get evaluated every time?**

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

This is the main motivation in case the sensor and controller has not a direct link. If we want the sensor to receive the controllers decision, we must evaluate a remote post master every time. According, if the learn in accepted or is discarded, the post master establish a permanent Mail Box or not.

4.2.2 Transfer of messages

Messages are more 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.

Smart Acknowledge (SmartACK) V1.4**4.2.3 Data update**

In scenarios where the Response period must be as short as possible, because of customer application requirements, 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 Learn In 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 unpleased system states can occur if ignored.

We can add sensors and repeaters in 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 behaviour**4.3.1 Controller at Learn In**

For understanding of the Smat Ack processes at learning and operating we will discuss some scenarios of the actors behaviour.

At figure below the baviour of controller at Learn In showed. The scenario is following:

- REQ#10 Controller receives Learn Request (original or repeated)
- REQ#20 Controller will collect Learn Request telegrams for the Learn request period
- REQ#30 Learn In
- REQ#40 According the result of post master election
 - Controller promotes itself to post master
 - OR
 - Remote Post Master gets elected and Controller sends Learn Reply

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

Smart Acknowledge (SmartACK) V1.4

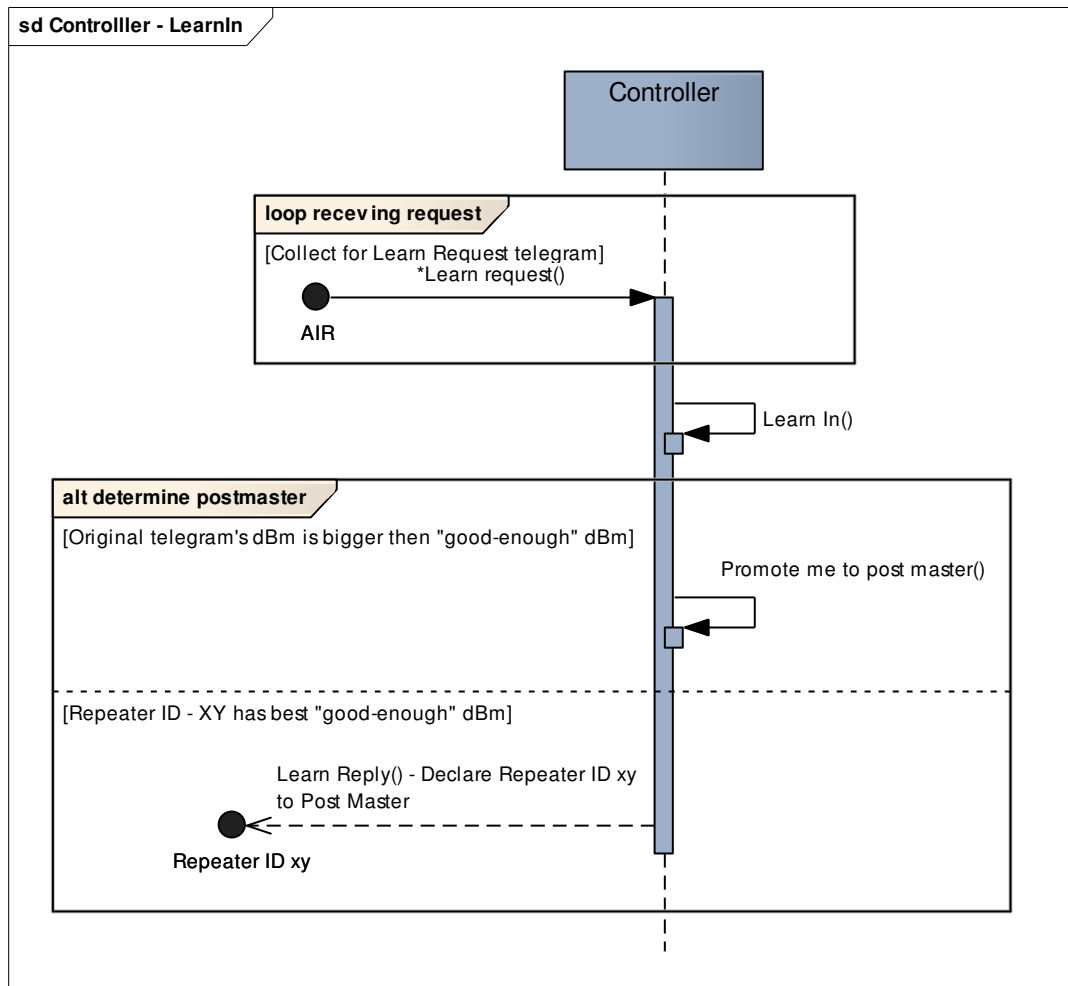


Figure 14 Controller at Learn In

4.3.2 Controller and Repeater at Learn In

When more repeaters are participating at the evaluation for Post master only one gets elected. Generic behaviour of a repeater with controller at advanced mode Learn In is shown at 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 telegram
 - Controller receives the telegram
 - Controller sends Learn reply
 - According the destination ID of the learn Reply
 - Repeater recognizes it was promoted to Post Master
- OR

Smart Acknowledge (SmartACK) V1.4

— Repeater repeats the Learn Reply

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

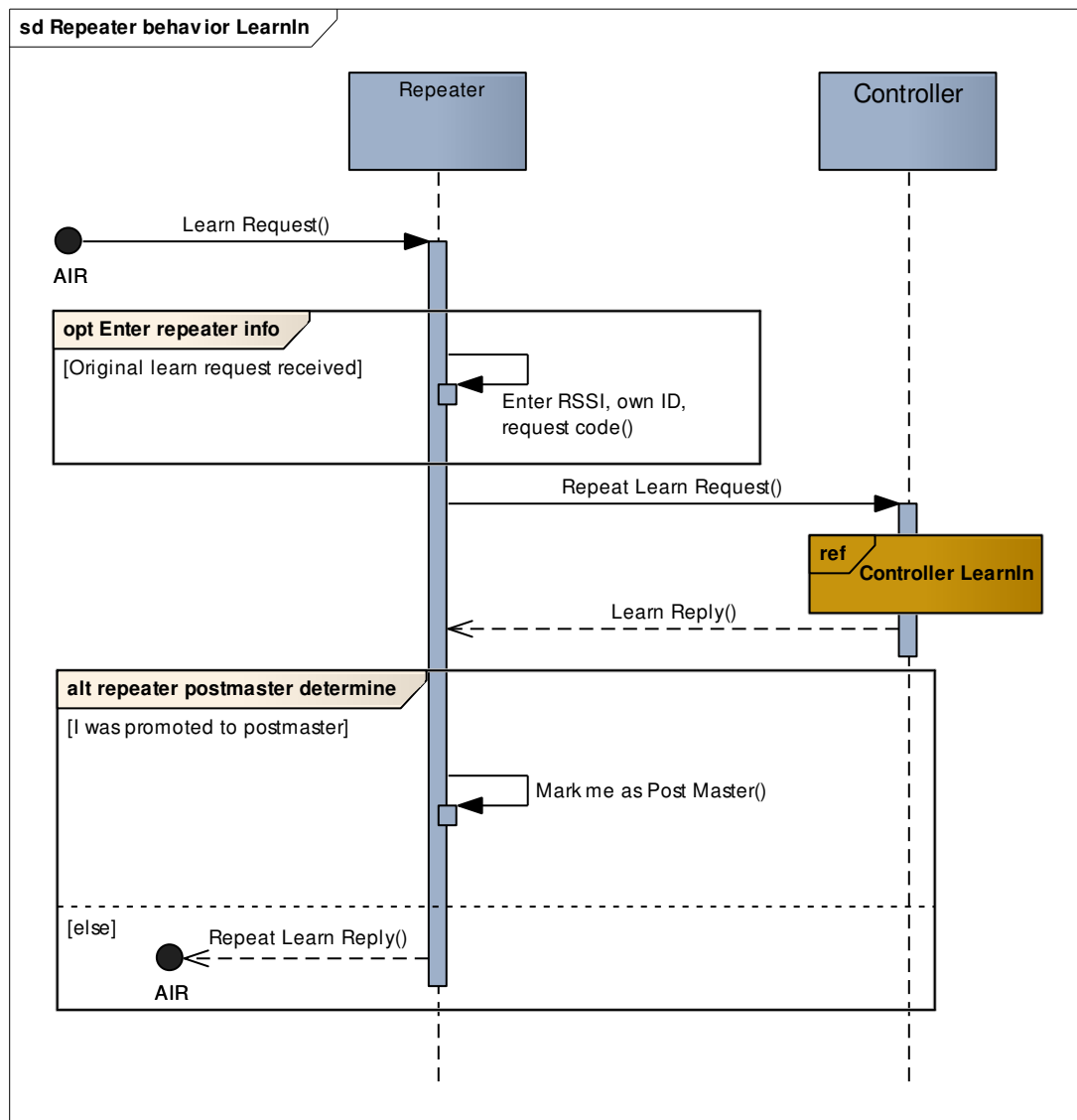


Figure 15 Controller and Repeater at Learn In

4.3.2.1 Learn In with more Controllers

For every Controller Sensor connection the learn process must be repeated. The fact in learn in with more controllers is that, the sensors have an already declared post master. It can be a controller or a repeater. The most important condition is that in next learn in with another Controller the already Post Master gets elected. Sensor can have only one Post Master, so if another Post Master gets elected system will get instable.

Smart Acknowledge (SmartACK) V1.4

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, which will learn in the Sensor. Behaviour of the already declared post Master is listened in figure below.

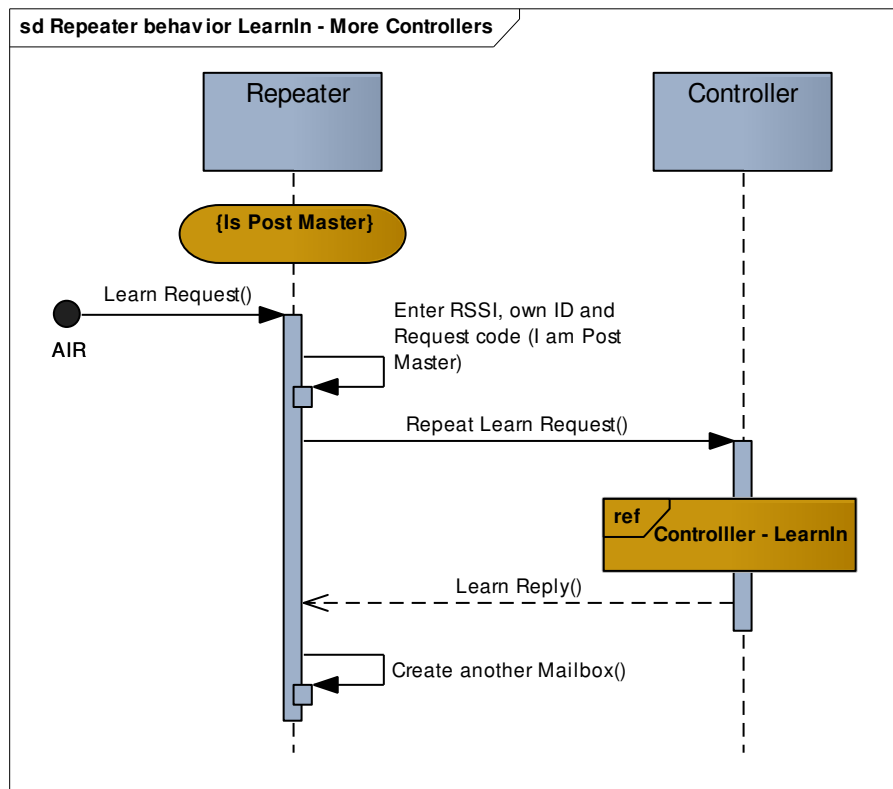


Figure 16 Post Master behaviour with next Controller

The scenario of Learn In with more controllers is shown in two steps. First one controller learns in the sensor and second the next controller learns in the sensor.

Smart Acknowledge (SmartACK) V1.4

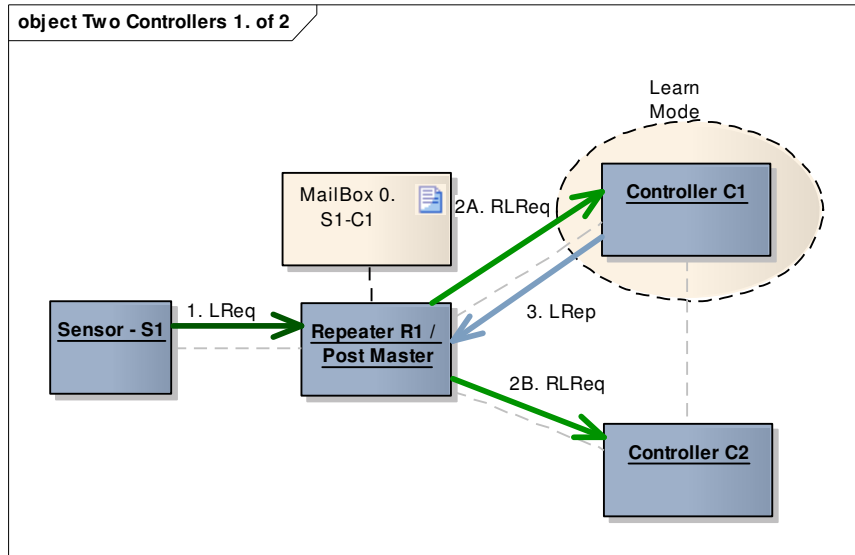


Figure 17 Learn In with two Controllers 1. of 2

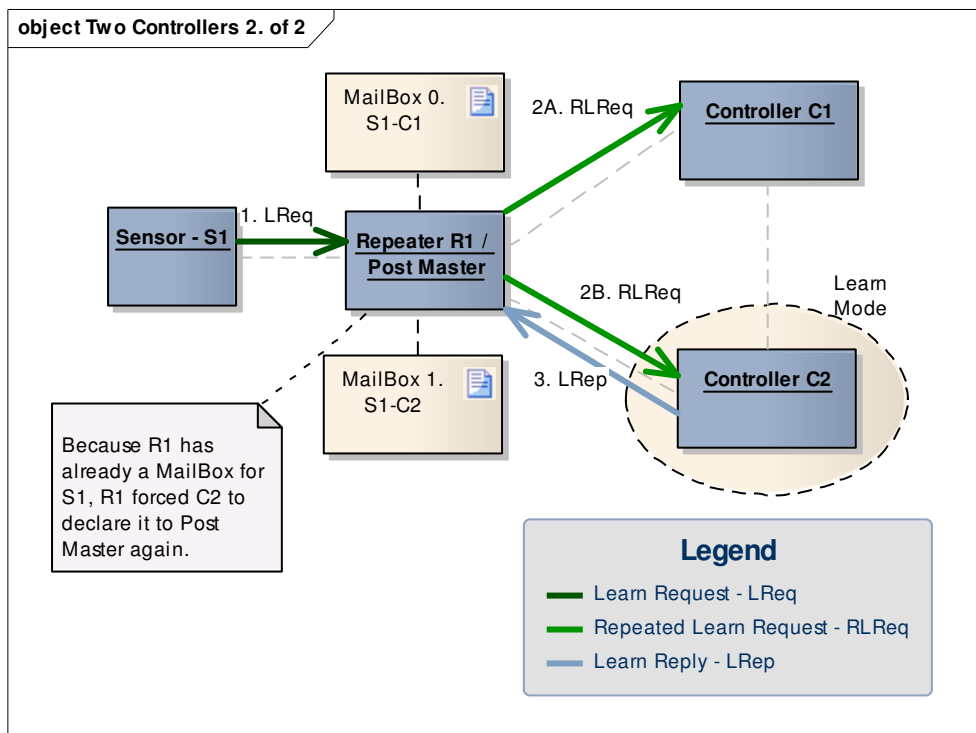


Figure 18 Learn In with two Controllers 2. of 2

In the following scenario, the postmastership gets the Controller, which learns the sensor first. The rule that the sensor can have only one Post Master must be respected at any cost. Also when there is a suitable candidate with fewer hops, it can not be evaluated.

Smart Acknowledge (SmartACK) V1.4

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

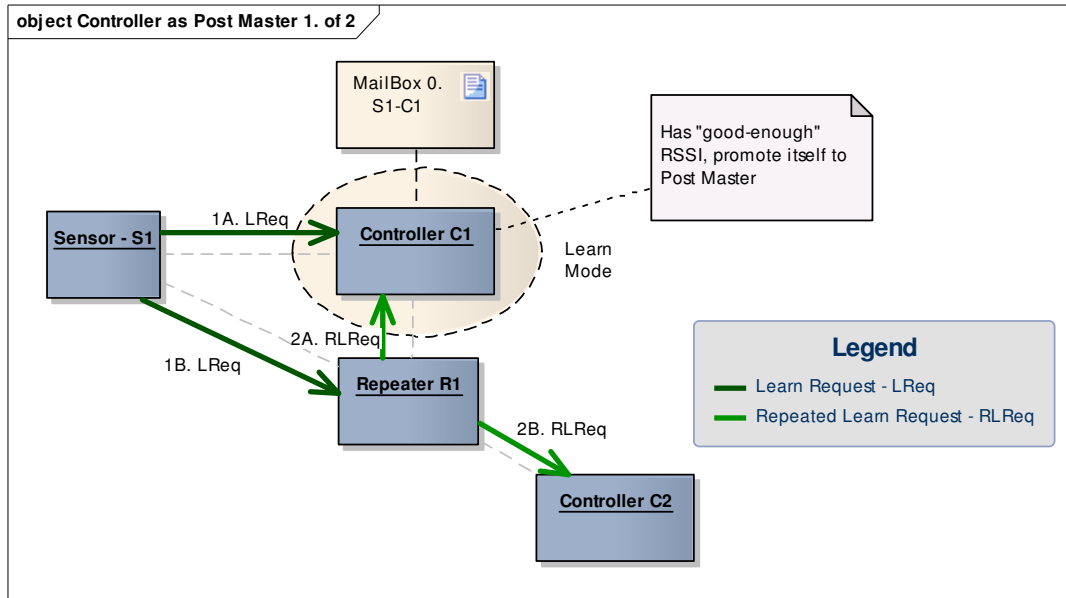


Figure 19 Controller as Post Master 1. of 2

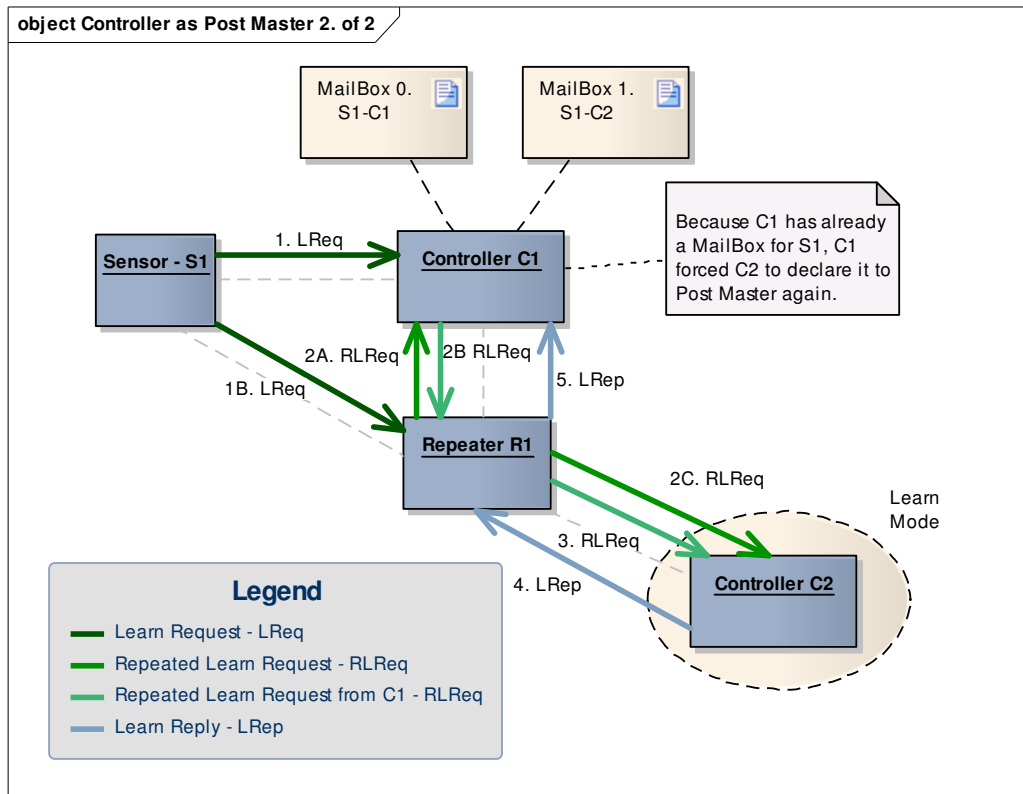


Figure 20 Controller as Post Master 2. of 2

Smart Acknowledge (SmartACK) V1.4

4.4 Learn out 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 Learn Out or as a an repeated Learn In. The decision is made by the Controller application.

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

After receiving Learn Reply with Learn Out Acknowledge code the Post Master deletes the Mail Box. It puts the Acknowledge for the Sensor into the temporary Mail Box, because we want 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 can not give back postmastership until all Mail Boxes are removed.

Remote Post Master would behave like in the figure below.

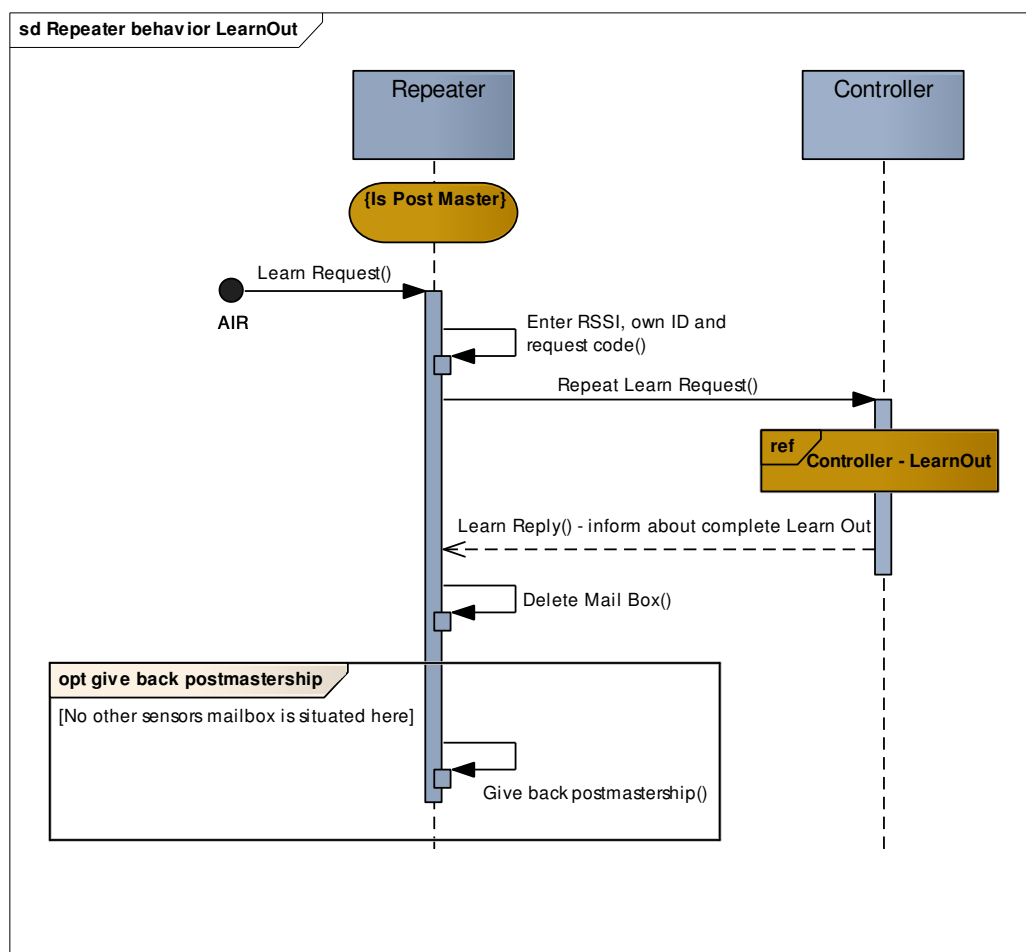


Figure 21 Controller and Repeater at Learn Out

5 Debugging

Since the Smart Ack is more complex it is susceptible to failure states. The states are results of i.e. lost telegrams or non communicating devices. These states can not be solved without additional help. So we need a possibility to active debug Smart Ack networks.

5.1 Debug process

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

These debug possibilities are provided:

1. Remote reset process
- q) Drop Mail Boxes at Post Master via Remote Management.
- r) Read Mail Box configuration at Post Master with Remote Management.
- s) Learn Out Sensor in Controller via Remote Management.
- t) Read Learn In 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 Sensor and Post Master have always a direct link. Other non Post Master repeaters do not repeat these telegrams. In repeating process we use status field and filters to evaluate if repeat or not. For purposes to avoid repeating we use the status value 0x0F that signalizes: "Never repeat again".

5.1.1.2 Post Master repeating

Declared Post Master must repeat all telegrams that come from Sensor. Otherwise they possibly can not reach the Controller. So although a Smart Ack actor is not a standard repeater it must repeat the traffic from learned Sensors.

5.1.1.3 Timing

In this chapter we look at the timing constrains and variables of Smart Ack protocol. The times are held as short as possible considering the lowest possible energy consumption but they are adjusted for the Dolphin Chip platform.

Smart Acknowledge (SmartACK) V1.4**Table 29 Periods**

Period name	Length	Description	Type
Learn Request period	250 ms	In periodController 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

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 Sensor is learned on Controller and has a Mail Box on the repeater - the Post Master is a 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 Learn In the Sensor. The Post master is the same as with the original Controller - a repeater. After 6 months we want to move the Sensor.

6.1.2 Issue:

Post Master will not give back postmastership because Learn Out can not 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:

- 1) 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 actual when the RSSI level (Sensor - Repeater) is no longer good enough.
- 2) 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 conflict and clear the mail box on 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.

Smart Acknowledge (SmartACK) V1.4**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 are repeated and carried to Controller. Controller sends 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 and sends no Data reply. We install new Smart Ack device, instead 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 be learn out the Sensor. Then the new device can be elected to Post Master and all Controllers can be learned in.

Rating: Undesired effects

6.2.2 Solution:

There are more possibilities to start learn process.

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

It is important to Learn Out the Sensor on all Controllers first and then start Learn In of the Sensor.

Smart Acknowledge (SmartACK) V1.4

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 Learn Out to free memory resources. The Learn Out can not be performed in common way because old Sensor is not available.

Rating: Feasibility of functions

6.3.2 Solution:

- 1 With remote management tell Controller to trigger Learn Out.
- 2 With remote management remove Mail Boxes from post master and tell Controller to learn out.

Smart Acknowledge (SmartACK) V1.4**6.4 Missing telegrams**

In system can rise error states 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 actual state of system. (i.e. Post Master is already declared). In the following text are described only critical error states and scenarios when messages get lost. Situations are separated by actual 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 Post Master Smack Ack actor
- any non Post Master Smack Ack actor → Controller – only actual when advanced learn enabled
- Sensor → Post Master
- Post Master → Controller – only actual when advanced learn enabled

Actual states:

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

Smart Acknowledge (SmartACK) V1.4

Table 31 Missing message issue 01

Message name	Learn Request
Sender	Sensor
Receiver	Post Master
Actual state	LearnIn with more Controllers (Post Master already declared)
Rating	RISK
<p>Description</p> <p>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.</p> <p>Problem will occur only at run-time.</p>	
<p>Detection</p> <p>Problem will occur only at run-time. Sensor gets Data Acknowledge from more Post Masters.</p> <p>Prevention</p> <p>Declare stricter "good-enough" RSSI boundaries.</p> <p>Solution</p> <p>Force finding Post Master again. (LearnOut all connections)</p>	

Table 32 Missing message issue 02

Message name	Learn Request
Sender	Post Master
Receiver	Controller
Actual state	LearnIn with more Controllers (Post Master already declared)
Rating	RISK
<p>Description</p> <p>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.</p> <p>Problem will occur only at run-time.</p>	
<p>Detection</p> <p>Problem will occur only at run-time. Sensor gets Data Acknowledge from more Post Masters.</p> <p>Prevention</p> <p>Post Master sends Learn Request more times.</p> <p>Solution</p> <p>Force finding Post Master again. (LearnOut all connections)</p>	

Smart Acknowledge (SmartACK) V1.4**6.4.2 Missing Learn Reply**

Actors connection:

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

Actual states:

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

Smart Acknowledge (SmartACK) V1.4

Sender	Controller
Receiver	Post Master
Actual state	LearnOut with one Controller, LearnOut with more Controllers
Rating	RISK
<p>Description</p> <p>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.</p> <p>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.</p>	
<p>Detection</p> <p>Sensor receives no learn acknowledge, can be signaled to user.</p>	
<p>Prevention</p> <p>Post Master sends Learn Reply more times.</p>	
<p>Solution</p> <p>LearnOut and then LearnIn Sensor on Controller.</p>	

6.4.3 Missing Learn Reclaim

Actors connection:

- Sensor → Post Master

Actual states:

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

Smart Acknowledge (SmartACK) V1.4

- Post Master → Sensor

Actual states:

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

Table 37 Analysis of missing Learn Acknowledge

Connection \ State	LearnIn with one Controller	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
Actual 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 signaled 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 MailBoxes. (in most Sensors the amount is 1)

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

Loosing one of Data messages is not a thread 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

Smart Acknowledge (SmartACK) V1.4

determination of stricter “good-enough” RSSI values and good radio infrastructure planning should be secured that Data telegrams get lost only occasionally.