SPECIFICATION V1.0

# EnOcean Radio Protocol

October 10, 2012

EnOcean GmbH
Kolpingring 18a
82041 Oberhaching
Germany

Phone +49.89.67 34 689-0
Fax +49.89.67 34 689-50
info@enocean.com
www.enocean.com

## REVISION HISTORY

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

| No | Major Changes |
|---|---|
| 1.0 | Document created |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

**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.
Packing: Please use the recycling operators known to you.

## TABLE OF CONTENT

# 1 Introduction

This document is the cover document of EnOcean Radio Protocol (ERP). The following table summarizes the ERP protocol in OSI layers. This document gives partial information about the Data Link Layer and Network Layer.

| Layer | Services | Data Units |
|---|---|---|
| Application (API) | EnOcean Equipment Profiles (EEP)<br>RPC/RMCC handling | DATA |
| Presentation | Radio Telegram Processing<br>Encryption | DATA |
| Session | --- not used --- | -- |
| Transport | Smart Ack<br>Remote Management | TELEGRAM/ MESSAGE |
| Network | Addressing telegrams (ADT Encapsulation/Decapsulation)<br>Switch telegram conversion (choice/status processing)<br>Repeating (status processing) | TELEGRAM |
| Data Link Layer | Subtelegram Structure<br>Control Sum Calculation<br>Subtelegram Timing<br>Listen before talk | SUBTELEGRAM |
| Physical | Encoding/Decoding (inverse bits)<br>Radio reception/transmission | BITS / FRAME |

## 2 Data unit description

The communication protocol is packet based and the data units can be of three different types:

- Frame
- Subtelegram
- Telegram

A frame is the representation of the encoded data on the physical layer. It includes control and synchronization information for the receiver. A frame is transmitted as a bit by bit serial sequence. A subtelegram is the result of a decoding process, in which this control (PRE, SOF, INV and EOF) and synchronization information are removed from the frame. The reverse mechanism to get a frame from a subtelegram is the encoding process.

The subtelegrams are handled in the data link layer. The ERP protocol is designed to work mostly as a unidirectional protocol without handshaking. To ensure transmission reliability three identical subtelegrams are transmitted within a certain time range. Each transmitted subtelegram is an atomic unit and contains all the information the composed telegram contains. The data structure of a subtelegram is shown in

Figure 1.

| 1 | 1 … X | 4 | 1 | 1 | byte |
|---|---|---|---|---|---|
| RORG | DATA | TXID | STATUS | HASH | |

**Figure 1 – Structure of a subtelegram**

The universal fields are:

- RORG/CHOICE – identifies the subtelegram type
- DATA – the payload of the transmitted subtelegram
- TXID/SourceID – identifies the transmitter, each having a unique 4 byte identity
- STATUS – identifies if the subtelegram is transmitted from a repeater and the type of integrity control mechanism used. This field is not present in a switch telegram.
- HASH/Checksum – data integrity check value of all the bytes
- The length of the subtelegram is not transmitted in the subtelegram structure. The length is determined by counting the number of bytes starting with RORG and ending with HASH.

# 3 Layer 2 – data link layer

## 3.1 Introduction

In the data link layer the transmitted data are one or more subtelegrams. The structure of a subtelegram is shown in 2.

## 3.2 Subtelegram timing

The subtelegram timing aims to avoid telegram collisions from different transmitters. Each subtelegram is transmitted in a different time range. The limits of the subtelegram timing are determined by the TX and RX maturity times. The maturity times specifies the length of the time range within which the transmission of all subtelegrams has to be completed and received. The values of the TX and RX maturity times are specified in Table 1 below.

A complete telegram consists of a maximum of 3 subtelegrams. The transmission of the start of the first subtelegram and the end of the last subtelegram by the transmitter shall not exceed the TX maturity time.

Repeaters have a different subtelegram timing range than the original transmitter. For the receiver, the time between receiving the end of the first subtelegram received and the end of the last subtelegram shall not exceed the RX maturity time also when repeaters are involved.

The LBT technique (see 3.4) makes it possible to avoid collision by controlling the subtelegram transmission timing, but it cannot completely guarantee the avoidance of a collision.

**Table 1: Maturity time parameters**

| Description | Parameter |
|---|---|
| Maximum TX maturity time | 40 ms |
| RX maturity time | 100 ms |

To schedule the subtelegram transmission the TX maturity time is divided into 4 groups; each of them with 10 time slots of the size 1 ms. The numeration of the time slots starts with 0 and ends with 39.

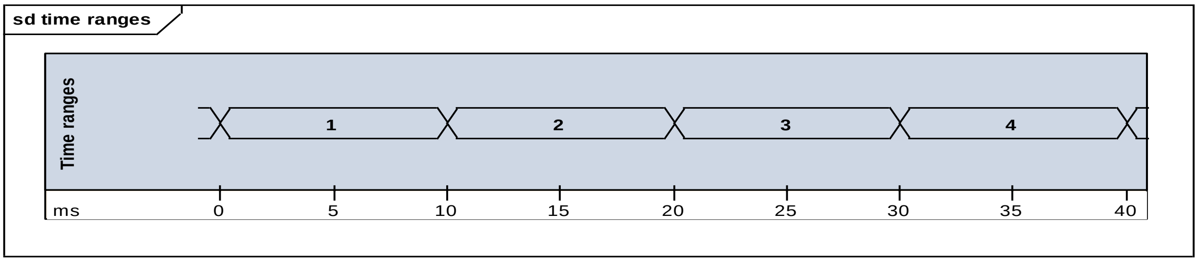


**Figure 2 – TX Maturity Time divided into four 10 ms time ranges**

These 4 ranges (see Figure 2) will be used for sending a maximum of 3 subtelegrams. The scheduling determines in what range what subtelegram number is allowed to be sent. To avoid collisions when using repeaters, the subtelegram timing of original and repeated telegrams differs depending only on the status of the repeated subtelegram and not on the configured level of the repeater. Table 2 defines in which time range, which is determined by the numbered time slots, each subtelegram may be transmitted.

**Table 2: Allocation of time slots to the different subtelegrams**

| Status of telegram | 1st subtelegram | 2nd sub telegram | 3rd sub telegram |
|---|---|---|---|
| Original | 0 | 1...9 | 20...39 |
| Level 1 repeated | 10...19 | 20...29 | |
| Level 2 repeated | 0...9 | 20...29 | |

All subtelegrams will be transmitted within these time ranges. A second or third subtelegram transmission may only start if the previous subtelegram transmission has been completed. There is no specified minimum pause between subtelegrams. The transmitter and repeater is free to use any time slot within each time range.

The transmission start of the first subtelegram of an original transmitter starts the time counting for the transmitter. The completion of the first subtelegram received (which due to disturbance is not always the first one from the transmitter) starts the counting in the receiver or the repeater.

If the wireless channel is occupied by the transmission of other transmitters, the LBT functionality can delay the transmission until the end of the TX maturity time is reached.

## 3.3 Data integrity

### 3.3.1 General

In order to check that a subtelegram has arrived intact, a hash of the telegram is calculated before transmission and attached to the subtelegram (field HASH). The attached hash value is not protected and thus only serves to detect transmission failures – not protection against malicious intent. The verification is done by the device receiving the telegram, i.e., a receiving device or a repeater. There are two public algorithms. They are 8 bits long. One is summation based and one uses an 8-bit long Cyclic Redundancy Check (CRC) algorithm.

If the verification of the intactness of the received subtelegram fails, the subtelegram is ignored.

The STATUS byte indicates which hash function is used. This is summarised in Table 3 below.

**Table 3: Identification of the hash function used in the telegram**

| Characteristics | Width | Used by telegram types |
|---|---|---|
| 8bit Checksum | 8 bit | any type of telegram when STATUS bit $2^7$ = 0 |
| 8bit CRC | 8 bit | any type of telegram when STATUS bit $2^7$ = 1 |

### 3.3.2 The 8 bit summation hash function algorithm

This clause describes the 8-bit checksum algorithm. The result of the calculation has the length of 8 bits. It is calculated by the transmitter before transmission and by the receiver after receiving the subtelegram.

The algorithm is as follows:

- The sum of the value of each byte in the subtelegram except the hash value field is evaluated ignoring overflow, i.e. all bits beyond the byte are ignored. This one byte (8 bits) sum value is the hash of the 8-bit algorithm.

### 3.3.3 The 8 bit Cyclic Redundancy Check (CRC) hash function algorithm

This hash function is based on the Cyclic Redundancy Check algorithm providing a hash value of length one byte.

The algorithm starts with the first byte of the subtelegram (RORG) and calculates the remainder of the division (modulo 2) by the generator polynomial $x^8 + x^2 + x + 1$ of the product $x^8$ multiplied by the first byte of the subtelegram.

The result of this calculation is XORed with the next byte in the subtelegram and again the remainder of the division is calculated as above.

This procedure is repeated until the last byte of the subtelegram excluding HASH is reached. The remainder of the final division is used as hash value.

## 3.4 Listen before talk

### 3.4.1 General

Listen before talk (LBT) is a technique used in wireless communications whereby a wireless transmitter or repeater first senses its wireless environment before starting a transmission. The aim is to avoid collisions with other senders. It is an optional feature of the transmitting device.

Prior to transmitting a subtelegram, the transmitting device checks, whether there is an ongoing transmission in the air. If this is the case the transmission is suspended for the delay of a random time range. After this delay the transmitter check is repeated. If no ongoing telegram transmission is detected, the subtelegram is transmitted. In case the calculated random delay would lead to the violation of the TX maturity time, the subtelegram transmission is sent irrespective of any other transmissions.

It is recommended to implement and use LBT before each subtelegram transmission, but it is not required. Some transmitting devices cannot support this feature as for example self powered products.

# 4 Layer 3 – network layer

## 4.1 Introduction

Three aspects of the repeating and addressing is described in this section.

## 4.2 Repeater

### 4.2.1 General

Repeaters are necessary when the distance between sender and receiver is too large to establish an adequate wireless connection. For bigger distances it is possible to place a maximum of two repeaters in a row. The job of the repeater is to receive the telegram from the sender or another repeater and send it again, so that the receiver of the message can get it. But before it is resent the repeater modifies the STATUS byte of the telegram. To limit the amount of repeated telegrams in an environment with more repeaters we differ between two repeater levels:

- Level 1 Repeaters repeat only received original subtelegrams.
- Level 2 Repeaters repeat only received original or once repeated subtelegrams.

If a level 2 repeater receives an original and also an once repeated subtelegram originating from the same transmitter, it will only repeat once with 3 subtelegrams.

### 4.2.2 Time response for collision avoidance

When there are repeaters in a system it is particularly important to avoid collisions. When a subtelegram is sent from a transmitter, it is thus necessary that the repeater does not repeat his received subtelegrams at the same time as another subtelegram from the original sender or a following repeater is transmitted.

Therefore a special subtelegram timing for repeaters is defined, which depends on the received subtelegram repeater level. This is described in detail in 3.2 above.

### 4.2.3 Bits of repeater level in the status byte

The STATUS field is used for a repeater to differ between subtelegrams from a transmitting device from those from a repeater, The bits $2^0$ to $2^3$ in the status field byte of each subtelegram shows the number of repeater hops of the telegram. The Table 4 shows the possible combinations:

**Table 4: STATUS byte with repeater level bits**

| Repeater level bits | | | | Description |
|---|---|---|---|---|
| $2^3$ | $2^2$ | $2^1$ | $2^0$ | |
| 0 | 0 | 0 | 0 | Original sender |
| 0 | 0 | 0 | 1 | Subtelegram was repeated 1 time |
| 0 | 0 | 1 | 0 | Subtelegram was repeated 2 times |
| 1 | 1 | 1 | 1 | Telegram shall not be repeated |

The Table 5 shows, how the repeater level bits have to be modified in the repeated sub telegram:

**Table 5: Repeating bits in STATUS byte**

| Repeater | Received subtelegram status | Repeated subtelegram status |
|---|---|---|

| | | |
|---|---|---|
| Level 1 | 0000 = original subtelegram received | 0001 = subtelegram is once repeated |
| | 0001 = once repeated subtelegram received | Subtelegram will not be repeated! |
| | 0010 = twice repeated subtelegram received | Subtelegram will not be repeated! |
| | 1111 = subtelegram shall not be repeated | Subtelegram will not be repeated! |
| Level 2 | 0000 = original subtelegram received | 0001 = subtelegram is once repeated |
| | 0001 = once repeated subtelegram received | 0010 = subtelegram is twice repeated |
| | 0010 = twice repeated subtelegram received | Subtelegram will not be repeated! |
| | 0011 = subtelegram shall not be repeated | Subtelegram will not be repeated! |

If a repeater receives subtelegrams of a telegram from a transmitter or a repeater, the status byte of the 3 repeated subtelegrams and the decision, whether the subtelegram is to be repeated, depends on the first received subtelegram according to Table 5.

## 4.3 Addressing

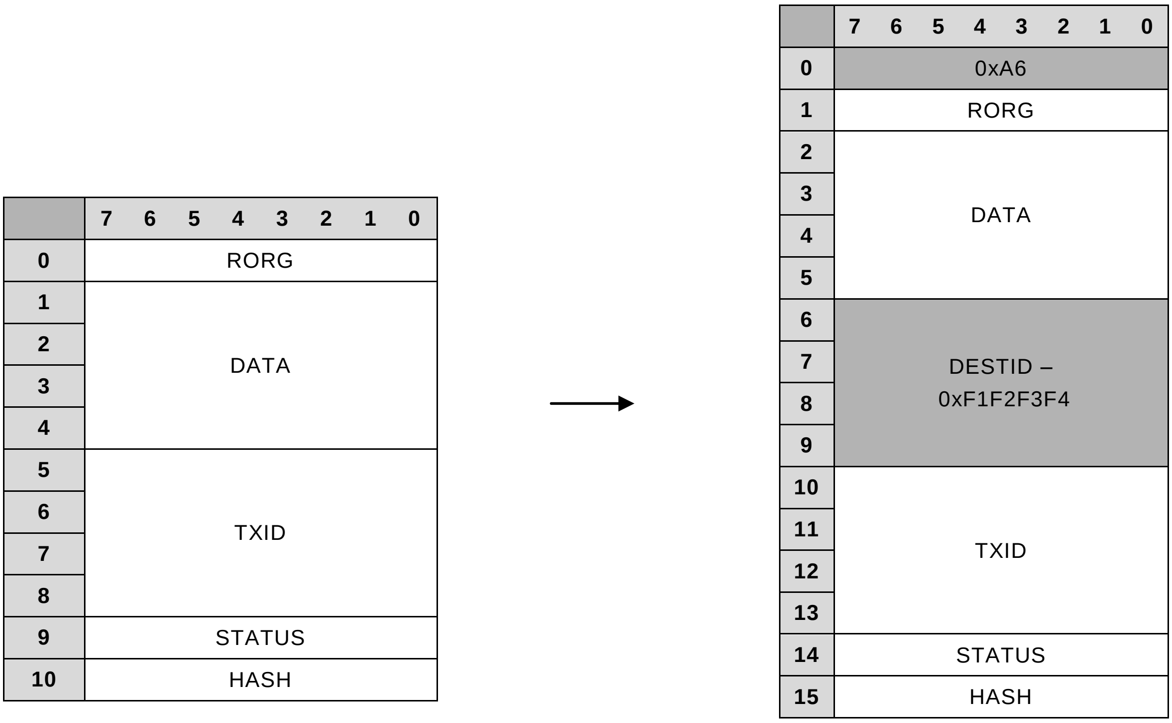
### 4.3.1 General

Addressing of telegrams is an essential feature for bidirectional communication. It is designed to enable future incorporation of additional features.

### 4.3.2 Encapsulation

The addressing of a telegram is performed by using an encapsulation mechanism. Encapsulated telegrams are recognized in the telegram type field (RORG) by the value 0xA6. The encapsulated field contains the original telegram that has to be addressed. The field destination identity DESTID of length four bytes is inserted preceding the field TXID, the transmitting identity.

Below is an example of how a telegram with destination identity DESTID 0xF1F2F3F4 would be encapsulated. Note that the value of the field DESTID is only an example:



|  | 7 6 5 4 3 2 1 0 |
|---|---|
| 0 | RORG |
| 1 | DATA |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 | TXID |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 | STATUS |
| 10 | HASH |

|  | 7 6 5 4 3 2 1 0 |
|---|---|
| 0 | 0xA6 |
| 1 | RORG |
| 2 | DATA |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 | DESTID –<br>0xF1F2F3F4 |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 | TXID |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 | STATUS |
| 15 | HASH |

**Figure 3 – An example of encapsulation**