

# INTERNATIONAL STANDARD

# ISO 10368

Second edition  
2006-02-15

---

---

## Freight thermal containers — Remote condition monitoring

*Conteneurs à caractéristiques thermiques — Système de pilotage à  
distance des groupes frigorifiques*



Reference number  
ISO 10368:2006(E)

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

© ISO 2006

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

Page

Foreword.....	iv
Introduction .....	v
1 Scope .....	1
2 Normative references .....	1
3 Terms and definitions.....	1
4 Performance requirements .....	3
4.1 General.....	3
4.2 Requirements .....	3
4.2.1 System components.....	3
4.2.2 Performance function.....	4
4.2.3 Performance constraints .....	8
5 System compatibility requirements .....	9
5.1 General.....	9
5.2 Communications protocol .....	9
5.2.1 General.....	9
5.2.2 MMU to MDCU communications .....	9
5.2.3 MDCU to LRCD communications .....	33
5.2.4 MDCU to HRCD communications.....	42
5.3 MMU/Device communications .....	50
5.3.1 Headers .....	50
5.3.2 Other MMU/Device messages.....	50
5.4 Low data rate physical requirements — LDCU to LRCD .....	51
5.4.1 Frequency.....	51
5.4.2 Modulation method.....	51
5.4.3 Baud rate .....	51
5.4.4 Transmission mode .....	51
5.4.5 Injection mode.....	51
5.4.6 Receiver sensitivity .....	51
5.4.7 Non-transmission Impedance .....	51
5.4.8 Bit synchronization.....	51
5.4.9 Carrier setup time .....	51
5.4.10 Out-of-band filtering for HDR compatibility .....	51
5.5 High data rate physical requirements — HDCU to HRCD.....	52
5.5.1 Modulation method — Broad band .....	52
5.5.2 Transmission mode .....	52
5.5.3 Injection mode.....	53
5.5.4 Output/input impedance.....	53
5.5.5 Power density function .....	53
5.5.6 Synchronization method .....	53
5.5.7 Demodulation method .....	53
5.5.8 Receiver sensitivity .....	53
5.5.9 Data link protocol.....	54
5.5.10 Out-of-band filtering requirements for “LDR” compatibility .....	61
Bibliography .....	64

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 10368 was prepared by Technical Committee ISO/TC 104, *Freight containers*, Subcommittee SC 2, *Specific purpose containers*.

This second edition cancels and replaces the first edition (ISO 10368:1992), which has been technically revised.

## Introduction

In revising this International Standard, material relating to the RCD/Controller interface has been deleted as it is not relevant to the powerline interface, and the section on data logging formats has been deleted as it is not used in the industry. Where necessary, other smaller additions, deletions and corrections have been applied.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning the low and high data rate systems given in 5.4 and 5.5 respectively. ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. (The declarations have not yet been received by ISO.) For the low data rate system, information may be obtained from:

Thermo King Corporation,  
314 W 90th Street  
Minneapolis, Minnesota 55420  
USA

For the high data rate system, information may be obtained from:

Adaptive Networks Incorporated  
1505 Commonwealth Ave.  
Suite 30  
Brighton, Massachusetts 02135  
USA

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

.....

# Freight thermal containers — Remote condition monitoring

## 1 Scope

This International Standard establishes the information and interfaces required to permit complying central monitoring and control systems employed by one carrier or terminal to interface and communicate with complying remote communication devices of differing manufacture and configuration used by other carriers and terminals.

The data-logging formats and message protocols outlined in this International Standard apply to all currently available data rate transmission techniques. These formats and protocols also apply to all future techniques designed to be an ISO International Standard-compatible system.

The performance requirements for the monitoring, communication and control system are given in Clause 4. The system compatibility requirements are given in Clause 5. All sections of this International Standard apply to all implementations, except where specified.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 1496-2:1996, *Series 1 freight containers — Specification and testing — Part 2: Thermal containers*

ISO 9711-2, *Freight containers — Information related to containers on board vessels — Part 2: Telex data transmission*

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

### 3.1

#### **remote communications device**

#### **RCD**

device which is physically a part of the refrigeration machinery and which communicates with any complying **CMCS** using the refrigeration machinery power distribution system as the data transmission medium

NOTE 1 See Figures 1 and 2.

NOTE 2 There are two distinct types of RCD:

- an **sRCD** (see 3.9); and
- an **iRCD** (see 3.10).

### 3.2

#### **central monitoring and control system CMCS**

system consisting of hardware and software which monitors and controls one or more **RCDs**

NOTE A typical system consists of at least:

- a) operator interface devices;
- b) an **MMU**; and
- c) power line data link equipment, such as an **MDCU**.

### 3.3

#### **master monitoring unit MMU**

central processing unit such as a computer which contains specific hardware and software to control the entire remote condition monitoring system

NOTE It is the interface between the human operator and the network.

### 3.4

#### **multiple data rate central control unit MDCU**

device which forms the link between the **MMU** and the three-phase power line bus which contains the individual **RCDs**

NOTE An **MDCU** consists of two components as follows:

- a central control unit capable of receiving and transmitting at the data rates which meet the requirements of this International Standard; and
- a central control interface.

### 3.5

#### **high data rate remote communications device HRCD**

**RCD** which transmits data at a high data rate, e.g. 19,200 baud

### 3.6

#### **low data rate remote communications device LRCD**

**RCD** which communicates data at a low data rate, e.g. 1 200 baud

### 3.7

#### **high data rate central control unit HDCU**

device which links the **MMU** and the power line network, communicating with the **HRCDs**

### 3.8

#### **low data rate central control unit LDCU**

device which links the **MMU** and the power line network, communicating with the **LRCDs**

### 3.9

#### **stand-alone remote communications device sRCD**

slave remote communications device **RCD** which, with limited capabilities, merely monitors a container refrigeration unit

NOTE An sRCD can be either high or low data rate.

**3.10****integrated remote communications device****iRCD**

slave remote **RCD** which interfaces to a refrigeration unit controller via an EIA R5232-C serial interface and can control the refrigeration machinery

NOTE An **iRCD** can be either high or low data rate.

**3.11****controller**

microprocessor device that monitors and controls the refrigeration machinery

**4 Performance requirements****4.1 General**

This clause specifies the performance requirements of central monitoring and control systems (CMCSs) necessary for them to interface and communicate with complying remote communications devices (RCDs).

**4.2 Requirements****4.2.1 System components****4.2.1.1 Remote condition monitoring system components**

A single remote condition monitoring system consists of a maximum of one master monitoring unit (MMU) and one multiple data rate central control unit (MDCU). (See Figure 1, Configuration A.)

**4.2.1.2 MDCU**

An MDCU may include one high data rate central control unit (HDCU) and one low data rate central control unit (LDCU). If an HDCU and an LDCU are both present, the single remote condition monitoring system consists of a maximum of one MMU and one MDCU. (See Figure 1, Configuration A. The HDCU and LDCU are joined together by a central control unit (CCU) interface, and the three components together form the MDCU.

**4.2.1.3 MMU/MDCU interface**

The preferred method of connecting the MMU to the MDCU complex is through a single port as shown in Figure 1, Configuration A. However, certain expansion paths may require multiple connections as shown in Figure 1, Configuration B.

**4.2.1.4 Remote communications devices (HRCDs and LRCDs)**

HRCDs and LRCDs shall be able to coexist on the same power line network and not interfere with simultaneous communications with either the HDCU or the LDCU.

**4.2.1.5 MDCU components**

An MDCU may consist of either a single HDCU (to communicate with the HRCDs on the network) or a single LDCU (to communicate with LRCDs on the network). However, all signalling protocols, data-logging formats, power levels, insertion rates and other physical requirements shall be identical to that which would be used for a combined system and therefore shall be compatible. Refer to 5.2 and 5.3 for the required protocol and data-logging formats.

## 4.2.2 Performance function

### 4.2.2.1 Standard message

All RCDs shall respond to a minimum list of standardized enquiries (see 4.2.2.4) and commands with a standardized reply or acknowledgement.

### 4.2.2.2 Acknowledgement message

The RCD shall send an acknowledgement message for all commands and enquiries that are received and understood.

### 4.2.2.3 "Not able" message

If the RCD is not capable of executing a command received or of responding to an enquiry because of the configuration of the RCD and the thermal control machinery, it shall respond with a "Not able" message.

### 4.2.2.4 Required enquiries

#### 4.2.2.4.1 General

All RCDs shall respond to the following required enquiries given in 4.2.2.4.2 to 4.2.2.4.7.

#### 4.2.2.4.2 Identification number

For an integrally refrigerated or thermal container, this shall be the container ISO number comprising a 4-letter alphabetical prefix and a 7-digit suffix (including the check digit). Where a demountable marine clip-on unit is used, the identification number shall be the MDCU number in ISO format.

#### 4.2.2.4.3 Porthole container number

This response shall be in addition to the identification number for MDCU systems.

#### 4.2.2.4.4 Porthole number change

This shall be recorded in the RCD memory in alphanumerical format, together with the time of the change.

#### 4.2.2.4.5 Return air temperature

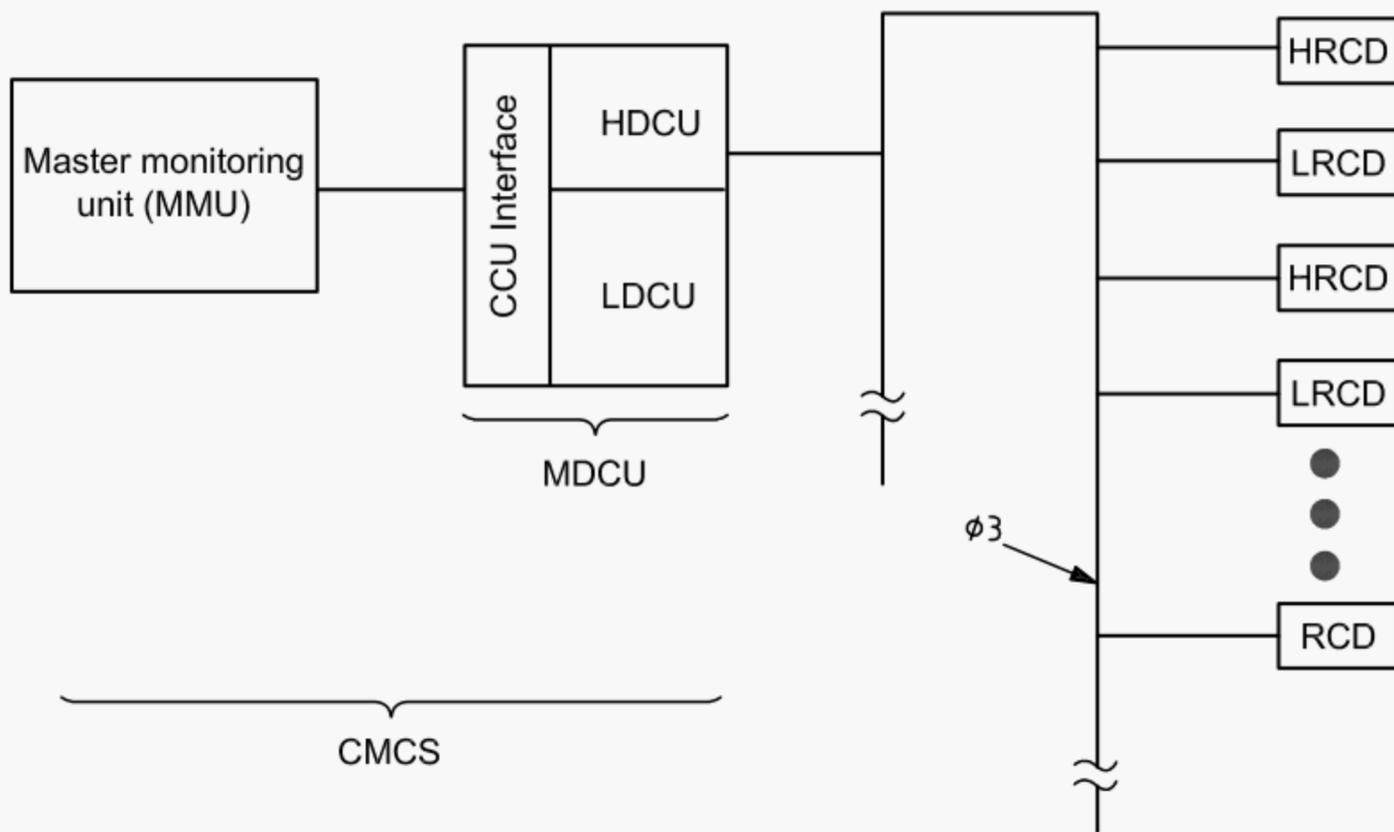
This shall be recorded in the form of a positive or negative value, expressed in degrees Celsius to one decimal place, within the range  $-30,0\text{ }^{\circ}\text{C}$  to  $+38,0\text{ }^{\circ}\text{C}$ .

#### 4.2.2.4.6 Supply air temperature

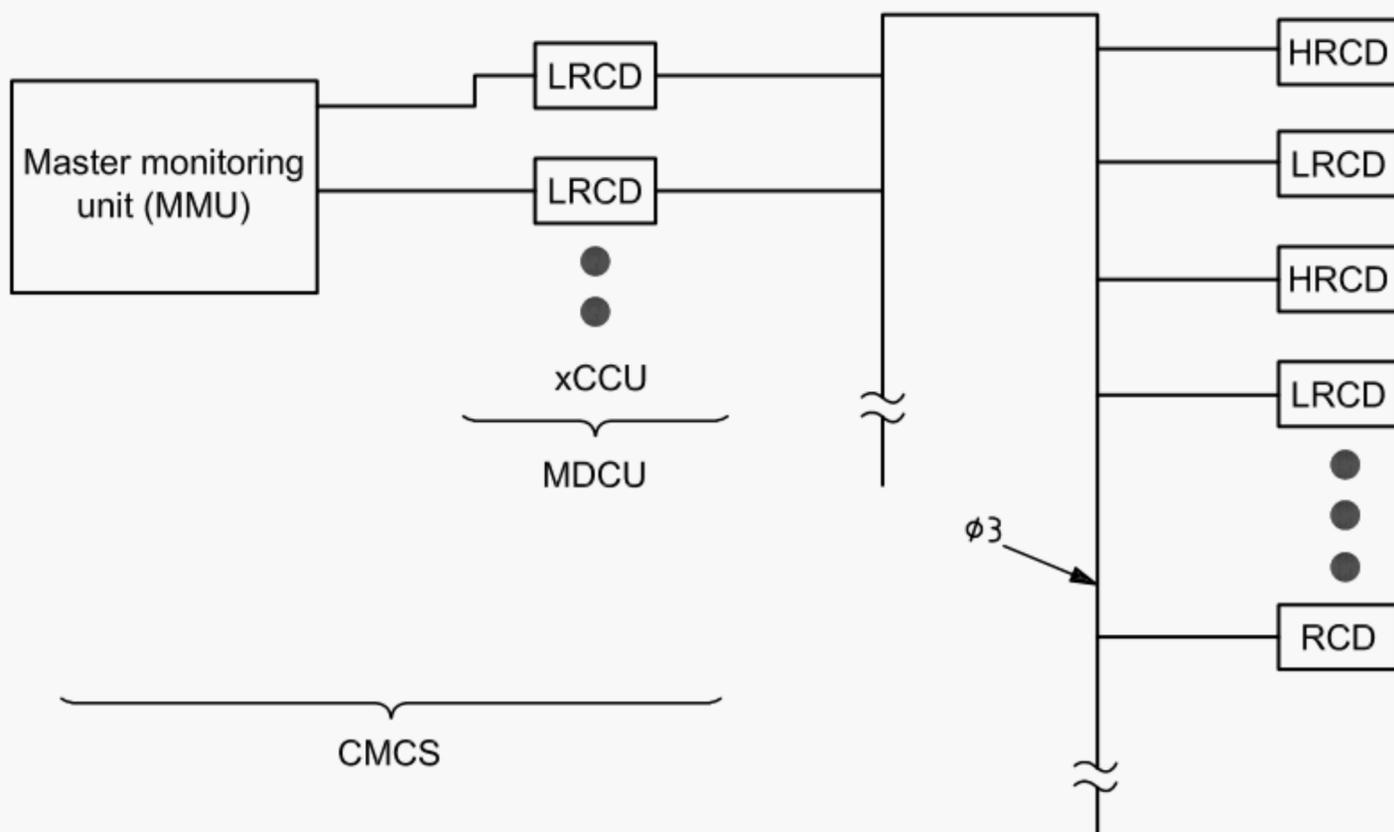
This shall be expressed in the same format as 4.2.2.4.5.

#### 4.2.2.4.7 RCD manufacturer and type

This shall consist of a unique identification number registered and controlled by ISO, and for which ISO/TC 104 is the registration authority.



a) Configuration A



b) Configuration B

**Components**

MMU	master monitoring unit	MDCU	multiple data rate central control unit
Ø3	three-phase power mains	HRCD	high data rate remote communications device
CCU	central control unit	LRCD	low data rate remote communications device
LDCU	low data rate central control unit		
HDCU	high data rate central control unit		
RCD	remote communications device		

**Figure 1 — Remote condition monitoring system components layout**

#### 4.2.2.5 Optional standard enquiries

##### 4.2.2.5.1 General

Other optional enquiries shall be standardized. RCDs and refrigeration machinery so equipped shall respond to the enquiries given in 4.2.2.5.2 to 4.2.2.5.14. RCDs not so equipped shall respond "Not able" (see 4.2.2.3).

##### 4.2.2.5.2 Operating mode

These include Full cool, Partial or Lower capacity cool, Modulated cool, Fans only or Null mode, Defrost, Heat, Off.

##### 4.2.2.5.3 Set-point temperature

This shall be expressed in the same format as 4.2.2.4.5.

##### 4.2.2.5.4 Alarms

These include High refrigeration pressure, Temperature out of range, Low compressor oil pressure, Defrost/Heat/Overheat, Compressor overload, Controller failure, Sensor failure — Return air, Sensor failure — Supply air, Power off, Amperage draw too high, Amperage draw too low, Defrost (out of time). (There is capacity for future development, e.g. controlled atmosphere.)

##### 4.2.2.5.5 All current alarms

These shall be in sequence of occurrence.

##### 4.2.2.5.6 Product temperatures

These include, for example, tank, poultry.

##### 4.2.2.5.7 Data-logger interval

This shall be one digit in half-hour intervals up to a maximum of 12 h.

##### 4.2.2.5.8 Amperage

This shall be 0 to 63,75 A in 0,25 A intervals.

##### 4.2.2.5.9 Destination

This may be up to five alphanumerical digits. If the destination changes, both the old and the current destination may be declared.

##### 4.2.2.5.10 Port of discharge

This may be up to five alphanumerical digits.

##### 4.2.2.5.11 Origin

This may be up to five alphanumerical digits.

##### 4.2.2.5.12 Report results of self-check level 1

These shall be one digit: 0 = Fail, 1 = Pass.

**4.2.2.5.13 Report results of self-check level  $n$** 

This shall be in the format of up to 256 ASCII characters, where  $n$  is a single character between two and nine.

**4.2.2.5.14 Vessel and voyage designation**

(See ISO 9711-2.)

**4.2.2.6 Commands****4.2.2.6.1 General**

RCDs and refrigeration machinery if so equipped shall respond to the commands given in 4.2.2.6.2 to 4.2.2.6.9. RCDs not so equipped shall respond "Not able" (see 4.2.2.3).

**4.2.2.6.2 Change set-point temperature**

This shall be expressed in the same format as 4.2.2.4.5.

**4.2.2.6.3 Initiate self-check level 1**

The self-check level 1 shall be initiated.

**4.2.2.6.4 Initiate self-check level  $n$** 

This shall be expressed in the same format as 4.2.2.5.13, where  $n$  is in the range two to nine.

**4.2.2.6.5 Change identification number**

This shall be expressed in the same format as 4.2.2.4.2.

**4.2.2.6.6 Change data-logger interval**

This shall be expressed in the same format as 4.2.2.5.7.

**4.2.2.6.7 Set data-logger time and date**

This shall have the date expressed in the format year/month/day.

**4.2.2.6.8 Change operating mode**

This shall be expressed in the same format as 4.2.2.5.2.

**4.2.2.6.9 Change porthole container number**

This shall be expressed in the same format as 4.2.2.4.4.

**4.2.2.7 Indecipherable or unserviceable messages**

Indecipherable or unserviceable messages shall not cause the RCD or CMCS to "crash" or "hang up". Also, failures of an electronic device in any RCD shall not cause the system to "crash" or "hang up".

#### 4.2.2.8 Verification of container identification number

The CMCS, if so equipped, shall verify the container identification number, using the check digit (the seventh digit of the numerical suffix) and an algorithm selected.

#### 4.2.3 Performance constraints

##### 4.2.3.1 Power interference

RCDs and CMCSs shall not interfere with the proper functioning of power supply regulating or controlling devices, such as voltage regulators or protective relaying equipment.

##### 4.2.3.2 Marine device Interference

CMCSs and RCDs, individually or as a system, shall not interfere with standard marine navigation and communication devices.

##### 4.2.3.3 System size

All CMCSs shall be suitable to coordinate and report on a system of 1 024 RCDs active at the same time on one CMCS.

##### 4.2.3.4 Status update

The MMU/MDCU system shall generate RCD updated status in accordance with 4.2.2.4 at least once per hour per container for a system of up to 1 024 containers active at the same time on one CMCS.

##### 4.2.3.5 Automatic RCD system list

The population or database of RCDs on the CMCS shall be self-generating. No input to the MMU, whether from an operator or from another computer, shall be necessary to determine the RCDs connected to that system.

##### 4.2.3.6 Identification of new RCDs

The MMU/MDCU system shall be designed to identify an average of at least one new container every 10 s, or 6 per minute.

##### 4.2.3.7 Voltage and frequency requirements

RCDs shall be suitable for operating on the voltage systems specified in ISO 1496-2.

##### 4.2.3.8 RCD connection

The RCD shall be connected on the line side of the refrigeration machinery disconnect or circuit breaker, if any, so that communication is possible when the refrigeration machinery is switched off. The RCD may have its own disconnect switch for servicing.

##### 4.2.3.9 Error rates

4.2.3.9.1 All CMCSs and RCDs shall be designed to meet the following error rate criteria.

The RCD/MDCU communication system may have two different types of "undetected and uncorrected" communication errors. An "undetected and uncorrected" communication error is one which is not detected and corrected within 5 min after occurrence.

**4.2.3.9.2** An error whereby an RCD executes a command which was not commanded by the MMU shall not occur more often than one time in  $25 \times 10^6$  messages (i.e. any power line disturbance which the receiver interprets as a message), or more often than once in 10 years for each CMCS, whichever is greater.

**4.2.3.9.3** An error whereby a CMCS misinterprets a message (i.e. any power line disturbance which the receiver interprets as a message) shall not occur more often than one time in  $25 \times 10^5$  messages.

## 5 System compatibility requirements

### 5.1 General

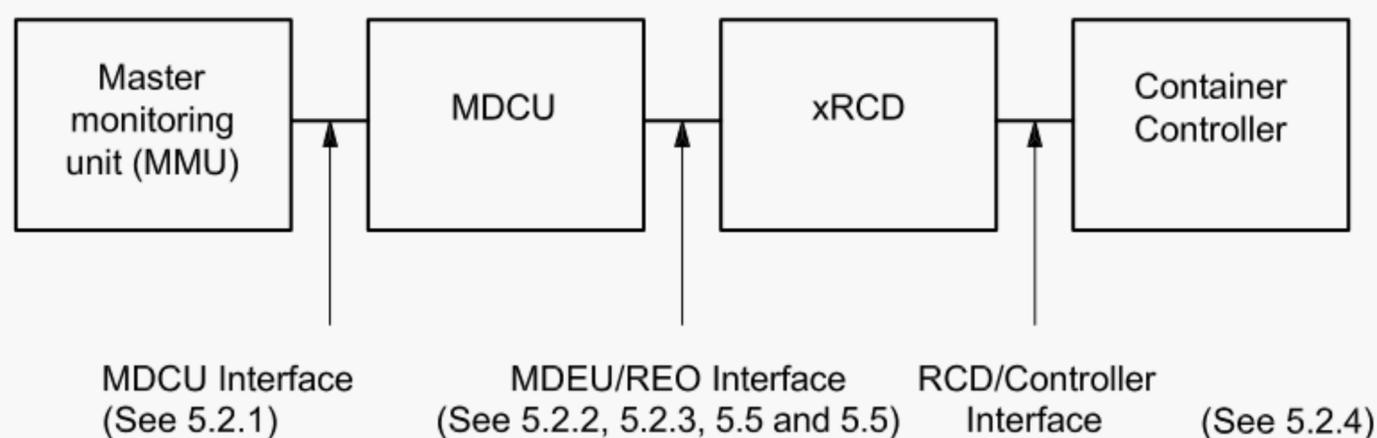
This clause specifies the interface requirements for communications protocol, data-logging formats, message definitions, and physical requirements for low data rate and high data rate (CU and CD) systems.

### 5.2 Communications protocol

#### 5.2.1 General

Each remote condition monitoring system has three interface areas as follows (see Figure 2):

- MMU to MDCU interface;
- MDCU to RCD interface;
- RCD to refrigeration machinery controller interface.



**Figure 2 — Remote condition monitoring — Communications interfaces**

#### 5.2.2 MMU to MDCU communications

##### 5.2.2.1 General

This subclause, in part, defines the communications protocol to be used when the MDCU is implemented as a discrete system component which is separate from the MMU architecture. The requirements given in this subclause do not preclude the use of bus-based open architecture MDCU applications where the EIA R5232-C is not appropriate.

The MMU communicates with the MDCU via a full duplex EIA RS232-C serial interface. The baud rate shall be at least twice the baud rate of the fastest RCD in the system. A typical communications baud rate is 4 800 baud. Each character transferred requires 1 start bit (low logic level), 8 data bits, and 1 stop bit (high logic level). The minimum time delay required between packets is one character time. This, therefore, restricts deadtime between any 2 bytes in a packet to less than one character delay.



The fields in Figures 4 and 5 are defined as follows:

- SYNC — An optional synchronization field. It may be any number of bytes, the contents of which are 16H. The MDCU strips all SYNC characters from the start.
- STX — Delimits the start of a valid message. It indicates that the next 2 bytes are the packet length.
- Packet length — A 2-byte unsigned integer which gives the packet length in bytes not including the SYNC, STX or CRC fields. It is transmitted high byte followed by low byte.
- Task No. — A 1-byte number assigned to the job before transmission to the MDCU. It is not used by the MDCU and is defined by the application. The MDCU reply contains the same value in this field. Task numbers shall be assigned sequentially by the MMU from 0 to 254 and then resume at 0. Task number 255 is reserved for the MDCU reset command.
- Xmit type — A 1-byte command directing the MDCU as to the type of processing which is to be performed on the data. The assignments are discussed in 5.2.2.2.
- Data — Variable length data field in which the contents are dependent on the type of message to be processed. The contents are defined in 5.2.2.2 and 5.2.2.3 for each appropriate command.
- CRC-A 16 bit error check field generated by using the CRC-16 polynomial:

$$X^{16} + X^{15} + X^2 + 1$$

The optional SYNC characters are not included in the CRC calculation.

## 5.2.2.2 MMU to MDCU transmit command types

### 5.2.2.2.1 General

There are two groups of commands which are sent to the MDCU: those directed to the MDCU for internal processing only and those to be transferred over the power line. The former group is given in Table 1, while the latter group is given in Table 2. An explanation of each command follows.

### 5.2.2.2.2 MDCU directed command types

#### 5.2.2.2.2.1 General

This group of commands requires processing within the MDCU, i.e. the power line communications network is not accessed. The commands currently supported by the MDCU are given in Table 1 and explained in 5.2.2.2.2.2 to 5.2.2.2.2.4.

**Table 1 — MDCU interrogation directed command types**

Command	Value
DCU reset	20H
MDCU parameters	22H
MDCU status	23H

#### 5.2.2.2.2 MDCU reset command

Upon request, the MDCU shall perform its internal diagnostics and initialize with default parameters. Any data associated with this command shall be ignored. Since the MDCU is reset, the initial status condition with default parameters shall be returned to the MMU as defined in 5.2.2.2.3. Task number 255 will be used in the reply.

#### 5.2.2.2.3 MDCU parameters command

The MDCU parameters command loads the MDCU with information passed in the data field portion of the command. A data field containing more than 15 bytes will cause the MDCU to ignore the command. The parameters to be loaded are defined below. Note that the parameters used for the low data rate system only are prefixed with an L while the parameters used for the high data rate system only are prefixed with an H.

First byte transmitted:

- Parameter 0 — Test status byte is a read-only location and, therefore, this parameter is ignored by the MDCU.
- Parameter 1-2 — MDCU software version/subversion is a read-only location and, therefore, this parameter is ignored by the MDCU.
- Parameter 3 — The MMU handshaking flag byte between the MDCU and MMU is changed. Each bit is discussed below.
  - Bit 7 — This is MDCU external UART CTS status (MMU communications). This bit is always received by the MMU in the active, or logic on, state since the UART is in the CTS mode (i.e. communicating with the MMU). This bit is a read-only bit; the MMU cannot set/reset it.
  - Bit 6-0 — Although accessible to the MMU, at present it is not used by the MDCU.
- Parameter 4-5 — Change is available for buffer size high (4) and low (5) bytes. It decreases/expands power line buffers. If the new size is below/above the MDCU's minimum/maximum value, its minimum/maximum value is used. A value of zero, or a value received which is equivalent to the previous setting, retains the previous setting. If a change in the buffer size is performed, all jobs within queue are lost.
- Lparameter 6 — The maximum size of the packets (or blocks) to be transferred on the power line is changed. If received larger than the available buffer size setting (described above), that value will be substituted. A value of zero retains the previous setting (unless that setting is greater than the new available buffer size). It is recommended that the packet size not be set greater than its default value (128 or 80H) unless the power line message timeout settings within the MDCU are changed accordingly.
- Lparameter 7 — The number of retransmissions to be attempted for a given job on the power line is changed.
- Lparameter 8 — The counter indicating the number of retries which have been performed on the power line is changed.
- Lparameter 9 — The counter indicating the number of successful normal polls which required one retry on the power line is changed.
- Lparameter 10 — The counter indicating the number of successful normal polls which required two retries on the power line is changed.
- Parameter 11 — The time delay base value used within an interactive command (5.2.2.2.3.4 and 5.2.2.2.3.9) that is required between the MDCU transmission of data to an RCD and the interactive poll used to obtain a response from the RCD is changed. Each count provides a 0,1 s incremental delay.

- Parameter 12-14 — This is presently undefined. The information will be copied into the status buffer as received. Since the first 3 bytes of the status buffer are read-only locations, any data field containing less than 3 bytes will not change any parameters in the MDCU status buffer. Also, Parameter 4-5 is 2 bytes in length and a byte count which transfers only one of these bytes will not update this parameter. Regardless of whether any parameters were changed, the MDCU reply will return the current status value to the MMU in a general valid type reply, as defined in 5.2.2.3.2.

#### 5.2.2.2.4 MDCU status command

The MDCU status command simply requests the MDCU to reply with its present status value. Any data associated with this command is ignored. The definition of this status buffer corresponds to the assignments discussed in 5.2.2.2.3 (MDCU parameters command), or as defined in 5.2.2.3.2.

#### 5.2.2.2.3 MDCU power line command types

##### 5.2.2.2.3.1 General

This group of commands requires communication with RCDs over the power line communications network. The commands currently supported by the MDCU are given in Table 2. The format of the power line is given in 5.2.2.3.

**Table 2 — MDCU power line directed command types**

Command	Value
RCD poll	30H
RCD xmit1	31H
RCD xmit2	32H
RCD interactive1	33H
RCD interactive2	34H
RCD map	35H
RCD interactive poll	36H
Device poll	40H
Device xmit1	41H
Device xmit2	42H
Device interactive1	43H
Device interactive2	44H
Device map	45H
Device interactive poll	46H
Extended device poll	47H

In all power line communication commands, the first byte of the data field is the routing byte. The routing byte's identification is derived from the original RCD map response (see 5.2.2.2.3.7) and is used by the MDCU to route the message to the proper data rate modem. Bits 7-6 define routing information for the low data rate RCDs and high data rate RCDs as described in Table 3. The remaining bits (5-0) are defined in the appropriate low or high data rate subclauses, 5.2.3 and 5.2.4 respectively.

**Table 3 — Routing byte data rate bit assignments**

Bits 7-6	Definition
00	Illegal
01	ISO LRCD
10	ISO HRCD
11	Reserved

The routing byte is always followed by an 11-byte ASCII RCD network address. Two address fields are reserved for special use in the RCDs. A universal address consisting of all ASCII Os, or all 30H, is recognized by all RCDs (i.e. a broadcast address) while a field containing nine ASCII Os followed by two ASCII??s, or nine 30Hs plus two 3IHs, is an address substituted in an RCD containing an invalid personal address. The command or message type, routing byte and RCD network address are called the power line prefix and shall comprise ASCII alphanumeric characters. This provides certainty for distinguishing these prefix characters from control characters.

#### 5.2.2.2.3.2 RCD poll command

The RCD poll command requests the status buffer from the selected RCD. The data field sent to the MDCU from the MMU contains the routing byte followed by the 11-byte ASCII RCD network address, i.e.

Data field to MDCU = Routing byte/address

A valid reply from the MDCU contains the routing byte/address and the RCD status buffer in the data field of a general valid type reply (see 5.2.2.3.2), i.e.

Data field from MDCU valid = Routing byte/address + RCD status values

The RCD status buffer contains the information defined below. Note that the status bytes used for the low data rate system only are prefixed with an L while the status bytes used for the high data rate system only are prefixed with an H.

First byte transmitted:

- Status0 — RCD status is written during initialisation of the RCD. A bit which is set indicates that an error condition occurred during the particular test, as shown in Table 4. This is a read-only location.

**Table 4 — RCD status state bit definitions**

Bit	RCD state
7	RGD type status: Set - Integrated RCD Reset - Stand-alone RCD
6	Datalog initialization incomplete
5	Reserved
4	RCD tests had an error:
3	Internal RAM error
2	External RAM error
1	Non-volatile memory error
0	Program check sum error

- Status 1-2 — RCD software version (1) and subversion (2) status is a read-only location.
- Status 3 — This is the handshake byte, used by the host to reset internal RCD buffers and also to prevent data tearing when multiple transfers of a particular buffer are obtained. Each bit is discussed below.
- Bit 7 — This is the RCD external UART CTS status (controller communications for the iRCD and external portable data collection computer for the sRCD). Set active indicates a logic one when the UART is in the CTS mode, i.e. currently capable of transfers. This bit is a read-only bit, the MMU cannot set/reset it.
- Bit 6 — This is the RCD device status buffer status. Set active indicates that the buffer contains information available to the MMU. The MMU may reset this status by transmitting logic zero in this bit position, while a logic one will not change the buffer's state.
- Bit 5 — This is the RCD device response buffer status. Set active indicates that the buffer contains information available to the MMU. The MMU may reset this status by transmitting logic zero in this bit position, while a logic one will not change the buffer's state. Note that a buffer in the process of being loaded by the controller (iRCD) will not abort the transmission.
- Bit 4 — This is the RCD device datalog buffer status. Set active indicates that the buffer contains information available to the MMU. The MMU may reset this status by transmitting logic zero in this bit position, while a logic one will not change the buffer's state. Note that a buffer in the process of being loaded by the controller (iRCD) will not abort the transmission.
- Bit 3 — This is the RCD response buffer status. Set active indicates that the buffer contains information available to the MMU.
- Bit 2 — For an sRCD only, this is the RCD external portable data collection computer buffer status, if applicable. Set active to a logic one when either the external serial channel UART receive or the transmit buffer contains valid information. This information is not exchanged on the power line communications network. This bit can be reset by the MMU, but the serial channel communications are not affected.
- Bit 1 — This is at present undefined.
- Bit 0 — This is the RCD data-logger buffer update handshake bit. The RCD sets this bit whenever the RCD has updated its internal data-logger buffer. The MMU resets the bit prior to transfer and checks it after the transfer is complete, thus providing a means to check possible data tearing of the buffer.
- Status 4-5 — This is the available power line buffer size high (4) and low (5) bytes.
- Lstatus 6 — This indicates the maximum size for any data packet (or block) transmitted by the RCD on the power line network. This number will never exceed the available buffer size. Its default is set at 128 or 80H.
- Lstatus 7 — This indicates the number of attempts the RCD will perform for a particular job in which it received an invalid response on the power line from the MDCU. The default is set at 02.
- Lstatus 8 — This indicates the number of retries required on the power line. Rollover OFFH to OOH will not occur. It can be reset using the RCD parameters command, as described in 5.2.2.4.3.
- Status 9 — This indicates the current baud rate the iRCD is communicating with the controller. For the sRCD, this indicates the current baud rate the device is communicating with an external device, if applicable. The corresponding baud rates are given in Table 5.
- Status 10 — This indicates the maximum baud rate the iRCD can communicate with the controller. For the sRCD, this indicates the maximum baud rate the device can communicate with an external device, if applicable. The corresponding baud rates are given in Table 5.

Table 5 — Baud rate definitions

RT	New baud rate
00	Default rate
01	75 baud
02	110 baud
03	300 baud
04	600 baud
05	1 200 baud
06	2 400 baud
07	4 800 baud
08	9 600 baud
09	19,2 kilobaud
10	38,4 kilobaud
11	67,2 kilobaud
12	100 kilobaud

Status 11-12 — Controller Software Version:

Data type: unsigned integer

Data format: stored as a 2 byte value in BCD (binary coded decimal) format

Example: rev 5117 = 01010001 00010111

Status 3 — Manufacturer Code: Assigned by ISO, Carrier Transicold — type “1”, type “0” reserved [restricted to numeric (non alpha) characters]:

Data type: unsigned byte

Data format: stored as an unsigned one byte value

Example: 1 = 00000001

Status 14 — Manufacturer Type Indicator: Assigned by manufacturer (restricted to numeric (non alpha) characters):

Data type: unsigned byte

Data format: stored as an unsigned one-byte value

Example: 82 = 01010010

Reserved values: The following values shall be reserved for ISO use:

0000H  
 FF00H  
 00FFH  
 FFFFh

### 5.2.2.2.3.3 RCD xmit1 and xmit2 commands

The RCD xmit1 and xmit2 command transfers text data to a selected RCD slave. The xmit1 versus xmit 2 selection defines the method of transfer on the power line and is further described for the appropriate low data rate or high data rate modem (5.2.3 and 5.2.4 respectively). From the point of view of the MMU, these process differences are invisible, except for the time of response. The data field sent to the MDCU from the MMU contains the routing byte followed by the 11-byte ASCII RCD network address. The information following this address is the text data, or message, to be given to the selected RCD, i.e.:

Data field to MDCU = Routing byte/address + RCD message

The RCD message field is discussed in 5.2.2.4. Inscribed in it is a subcommand which the RCD evaluates and processes accordingly. If this subcommand requires a response from the RCD, the RCD buffers it in an RCD response buffer. An RCD interactive poll command (as described in 5.2.2.2.3.6) is then used to obtain this buffered response. These xmit commands, however, only return a general valid type reply (see 5.2.2.3.2) from the MDCU to the MMU after the MDCU receives the proper acknowledgement of transfer from the slave RCD. The routing byte/address are returned in the data field of this reply, i.e.:

Data field from MDCU valid = Routing byte/address

#### 5.2.2.2.3.4 RCD Interactive1 and interactive2 commands

The RCD interactive1 and interactive2 commands transfer text data to a selected RCD slave in exactly the same way as the RCD xmit commands described in 5.2.2.2.3.3. As with the xmit commands, the interactive1 versus interactive2 selection defines the method of transfer on the power line and is further described for the appropriate low data rate or high data rate modem (5.2.3 and 5.2.4 respectively). From the point of view of the MMU, these process differences are invisible, except for the time of response. The interactive commands provide the RCD with a subcommand inscribed in the text data, which the RCD evaluates and processes accordingly. (The RCD processes the interactive commands in the same way as the xmit commands.) If this subcommand requires a response from the RCD, the RCD buffers it in an RCD response buffer. The difference between the RCD xmit and interactive commands occurs here in the MDCU processing. Instead of returning a valid reply to the MMU when the MDCU receives a proper acknowledgement from the slave RCD for the transfer of text data, the MDCU queues an RCD interactive poll command, as described in 5.2.2.2.3.6, without intervention from the MMU. A time delay later, as specified by a byte in the interactive command's data field, this interactive poll is sent to the RCD. Thus the data field sent to the MDCU is:

Data field to MDCU = Routing byte/address + Response delay + RCD message

In the above data field, the response delay is a 1-byte value. Each count of this value causes an approximate incremental delay, calculated as this number times the base value established in the MDCU Parameter11 byte described in 5.2.2.2.2.3, between the completion of the interactive command and the transmission of the interactive poll. The response of the RCD to this poll is then returned to the MMU as the reply for the interactive command and, therefore, its format is as given in 5.2.2.2.3.6. This reply will be dependent on the amount of RCD processing required for the RCD message subcommand. Note that subcommands not requiring RCD responses should not use an interactive command since the interactive poll would receive a negative acknowledgement from the RCD (i.e. no response buffer will be available).

#### 5.2.2.2.3.5 RCD map command

The RCD map command broadcasts a polling message from the MDCU requesting, with varying probabilities, an RCD device to return its network address. The purpose of this command is to log in, or map, any newly acquired or unpolled RCDs on the power line network. Since the mapping technique uses a polling-type process on the network, the format of the data field from the MMU to the MDCU is similar to the poll commands described in 5.2.2.2.3.3 and 5.2.2.2.3.7, i.e. the routing byte followed by the 11-byte ASCII RCD network address. However, being a broadcast type of message transfer, it cannot simply select a particular RCD by using a unique individual network address. Instead, the 11-byte ASCII field is broken into ASCII subfields supplying all RCDs on the power line network with a mapping strategy. The format of this field is shown in Figure 6.

	Alpha network address	Type repeat	Map subcommand	Unused HRCD destination address	Shift variable	Destination address individual
No. of bytes	4	3	1	2	1	2

Figure 6 — Format of RCD map command

The fields in Figure 6 are defined as follows:

- Alpha network address — This is comprise of 4-byte ASCII characters defining an RCD alpha portion (usually denoting the container operator) of the RCD network address on the power line network. All RCDs recognize the map command by using the universal address, or four ASCII 0s, or 30H, 30H, 30H, 30H.
- Type repeat — This is a 3-byte check field filled with a map command (045H). The device map command value is used instead of the RCD map command to allow an alphanumeric value to reside in the normally numeric polling field.
- Map subcommand — This is a 1-byte ASCII character which provides an RCD with a processing procedure in the mapping strategy. These procedures, and their associated values, are given below.
- Unused HRCD destination address — This is a 2-byte hexadecimal field which provides an HRCD with a network address. Only the HRCD(s) responding to the map command use this address. This field is used only when the alpha network address specifies the universal address, i.e. four ASCII 0s, or 30H, 30H, 30H, 30H.
- Destination address individual — This is a 2-byte hexadecimal field which provides an HRCD with a network address. Only the HRCD responding to the map command uses this address. This field is present only when the map address specifies an individual 11-byte address.
- Shift variable — This is the upper nibble of the byte and shall be 1111 B. The lower nibble contains a number from 0 to 15 which indicates the number of bits to mask from an RCD's 16-bit random number (0 will provide a 1111111111111111 B bit mask, while 15 will provide 0000000000000001 B). An RCD which obtains a value of zero by performing the logical AND of this mask with the 16-bit random number will attempt to return its network address provided that its internal respond to map flag is set (see below). Thus, this shift variable provides the mapping strategy with the capabilities of adjusting the range, or size, of the expected number of RCD responding.

Any field described above which does not contain a value as required will ignore the command. Thus, the data field sent to the MDCU from the MMU is:

Data field to MDCU = Network address tag/map information

An RCD will not accept a map command unless the alpha network address characters match those of the particular RCD or the universal address. This provides for selection of different classes of RCDs using the same data rate and thus the same routing byte. When accepted, the RCD decodes the map subcommand to perform different strategies of the mapping sequence. These subcommands are listed in Table 6. Any other value received by the RCD in the subcommand will force the RCD to ignore the command. Each RCD addressed will perform the map function received regardless of the respond to map state. Only after execution of the map function does the RCD check the respond to map state. If a map response is indicated, the RCD performs the logical AND of the shift variable mask with its 16-bit random number. A value of zero will allow the RCD to return its network address.

**Table 6 — Map subcommand definitions**

Subcommand	ASCII value	HEX value	RCD process
Initiate new map		49	Sets its respond to map flag, generate a new 16 bit random, and set up a new shift variable.
General new map	G	47	Generate a new 16 bit random and set up a new shift variable.
Change shift map	5	53	Set up a new shift variable.
Decrement map	D	44	Decrement its random (variable rollover will occur) and set up a new shift.

An RCD will set its internal respond to map flag in one of three ways:

- a) The RCD starts a power-up sequence, either by initially being turned on, or by an internal dead-man reset occurrence by an RCD directed reset command from the MMU device (see 5.2.2.4).
- b) The RCD has not received an individually addressed command within 1 h.
- c) The RCD receives an initialize map subcommand (defined above).

The flag is reset each time the RCD receives an individually addressed command. At this time the RCD initiates the 1 h timer discussed in b) above.

When the MDCU receives a network address from the RCD, it returns a general valid type reply (see 5.2.2.3.2) to the MMU with the data field containing this RCD network address following the map command message it used, i.e.:

Data field from MDCU value = Routing byte/map information + RCD network address

#### 5.2.2.3.6 RCD interactive poll command

The RCD interactive poll command is used to retrieve information from an RCD after the RCD has been sent an RCD xmit 1 or xmit 2 command (see 5.2.2.3.3) in which its subcommand in the RCD message field caused the RCD to buffer a response (see 5.2.3.2). The format of the data field from the MMU to the MDCU for the interactive poll is exactly the same as that of a normal poll command, i.e.:

Data field to MDCU = Routing byte/address

The time between the xmit and interactive poll is the responsibility of the MMU, and shall be enough to allow the RCD to process and buffer its response. If an RCD response buffer does not contain information, a negative acknowledge (see 5.2.2.3.5) is returned from the RCD to the MDCU, and consequently to the MMU. If, however, the RCD has buffered information, this information is returned to the MMU via a general valid type reply (see 5.2.2.3.2) with this information preceded in the data field by the network tag and the RCD's address, i.e.:

Data field from MDCU valid = Routing byte/address + RCD message

Once the RCD transfers this buffered data, it resets (and thus empties) its RCD response buffer, thereby allowing another RCD xmit or interactive command to be accepted by the RCD.

#### 5.2.2.3.7 Device poll command

The device poll command requests the quick status buffer of the controller which is attached to the selected iRCD or the internal quick status of the sRCD. The data field sent to the MDCU from the MMU contains the routing byte followed by the 11-byte ASCII RCD network address and the expected reply length, i.e.:

Data field to MDCU = Routing byte/address

A valid reply from the MDCU contains the routing byte/address and the controller (iRCD) or internal (sRCD) quick status buffer in the data field of a general valid type reply (see 5.2.2.3.2), i.e.:

Data field to MDCU valid = Routing byte/address + Quick status

For an iRCD, an image of the controller quick status buffer is contained within the iRCD and updated approximately every 30 s. For both the iRCD and the sRCD, the reply format of the data field of this message is presented in Figure 7.

#### 5.2.2.3.8 Device xmit1 and xmit2 commands

The device xmit1 and xmit2 commands transfer text data to a selected RCD slave. The xmit1 versus xmit2 selection defines the method of transfer on the power line and is further described for the appropriate low data rate or high data rate modem (5.2.3 and 5.2.4 respectively). From the point of view of the MMU, these process differences are invisible, except for the time of response.

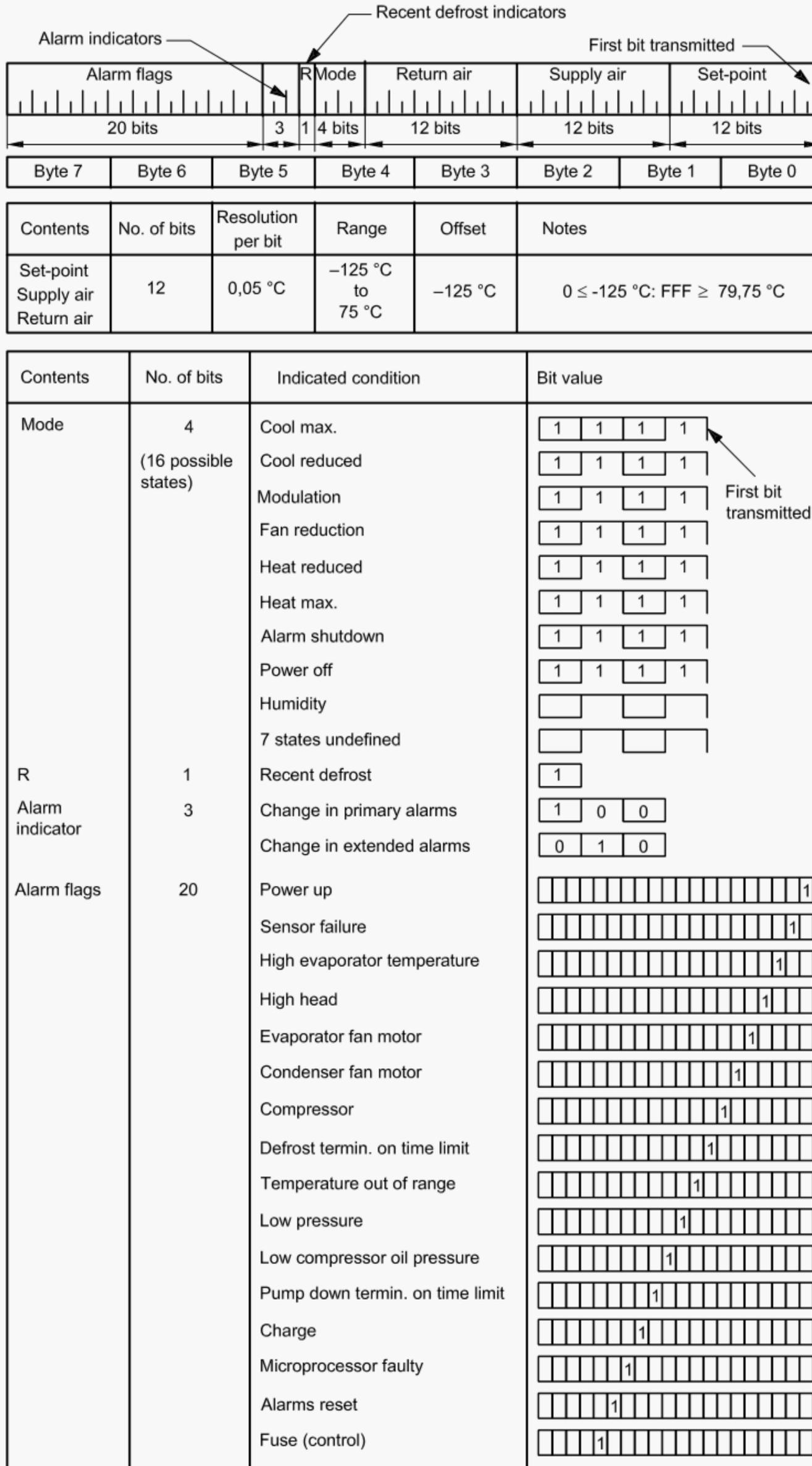


Figure 7 — Device poll (total 8 bytes or 64 bits)

The data field sent to the MDCU from the MMU contains the routing byte followed by the byte ASCII RCD network address. The information following this address is the text data, or message, to be given to the selected iRCD for transfer to the controller, or the selected sRCD for device input/output processing, i.e.:

Data field to MDCU = Routing byte/address + Device message

This device message field is discussed in 5.2.2.6. Inscribed in it is a subcommand which the RCD evaluates and processes accordingly. If this subcommand requires a response from the controller (iRCD) or the input/output function (sRCD), the RCD buffers it in a device response buffer. A device interactive poll command, as described in 5.2.2.3.11, is then used to obtain this buffered response. These xmit commands, however, only return a general valid type reply (see 5.2.2.3.2) from the MDCU to the MMU after the MDCU receives the proper acknowledgement of transfer from the RCD (this acknowledgement is an indication of the MDCU/RCD transfer and has no bearing on, for instance, the iRCD/controller transfer status). The routing byte/address are returned in the data field of this reply, i.e.:

Data field to MDCU = Routing byte/address

#### 5.2.2.3.9 Device interactive1 and interactive2 commands

The device interactive1 and interactive2 commands transfer text data to a selected RCD slave in exactly the same way as the device xmit commands described in 5.2.2.3.8. As with the xmit commands, the interactive1 versus interactive2 selection defines the method of transfer on the power line and is further described for the appropriate low data rate or high data rate modem (5.2.3 and 5.2.4 respectively). From the point of view of the MMU, these process differences are invisible, except for the time of response. The interactive commands provide the RCD with a subcommand inscribed in the text data, which the RCD evaluates and processes accordingly. (The RCD processes the interactive commands in the same way as the xmit commands.) If this subcommand requires a response from the controller (iRCD) or the input/output function (sRCD), the RCD buffers it in an RCD response buffer. The difference between the device xmit and interactive commands, like the difference between the RCD commands described in 5.2.2.3.3 and 5.2.2.3.4, occurs here in the MDCU processing. Instead of returning a valid reply to the MMU when the MDCU receives a proper acknowledgement from the RCD for the transfer of text data, the MDCU queues a device interactive poll command, as described in 5.2.2.3.11, without intervention from the MMU. A time delay later, as specified by a byte in the interactive command's data field, this interactive poll is sent to the RCD. Thus the data field sent to the MDCU is:

Data field from MDCU valid = Routing byte/address + Response delay + Device message

In the above data field, the response delay is a 1-byte value. Each count of this value causes an approximate incremental delay, calculated as this number times the base value established in the MDCU Parameter 1 byte described in 5.2.2.2.3, between the completion of the interactive command and the transmission of the interactive poll. The response of the RCD to this poll is then returned to the MMU as the reply for the interactive command and, therefore, its format is as given in 5.2.2.3.6. This reply will be dependent on the RCD processing required for the device message subcommand. Subcommands not requiring device responses should not use an interactive command since the interactive poll would receive a negative acknowledgement from the RCD (i.e. no response buffer will be available).

#### 5.2.2.3.10 Device map command

The device map command is indistinguishable from the RCD map command (see 5.2.2.3.5).

#### 5.2.2.3.11 Device interactive poll command

The device interactive poll command is used to retrieve information from an RCD after the RCD has been sent a device xmit1 or xmit2 command (see 5.2.2.3.8) in which its subcommand in the data field required the iRCD to buffer a controller response or the sRCD to buffer a device input/output process (see 5.2.2.6). The format of the data field from the MMU to the MDCU for the interactive poll is exactly the same as that of a normal poll command, i.e.:

Data field to MDCU = Routing byte/address

The time between the xmit and interactive poll is the responsibility of the MMU, and shall be enough to allow the RCD to process and buffer its response. If the RCD's device response buffer does not contain information, a negative acknowledge (see 5.2.2.2.3.5) is returned from the RCD to the MDCU and consequently to the MMU. If, however, the RCD has buffered information, this information is returned to the MMU via a general valid type reply (see 5.2.2.3.2) with this information preceded in the data field by the routing byte/address, i.e.:

Data field to MDCU valid = Routing byte/address + Message

Once the RCD transfers this buffered data, it resets (and thus empties) its device response buffer, thereby allowing another device xmit or interactive to be accepted by the slave RCD. (Note that the RCD's device response buffer is independent of the RCD's RCD response buffer discussed in 5.2.2.3.6.)

#### **5.2.2.2.4 Extended device poll**

##### **5.2.2.2.4.1 General**

The purpose of this International Standard is to propose the architecture for an enhanced device poll message. The current device poll command is defined in 5.2.2.2.3.7. This command allows the retrieval of limited information from the refrigeration unit's controller. The proposed enhanced device poll will enable the retrieval of information common in today's refrigeration units, including USDA, humidity and controlled atmosphere sensors and their set-points. Implementation of this architecture will improve both the content and performance of the polling process by providing access to all relevant data in one message.

##### **5.2.2.2.4.2 Background**

A configurable version of a type of enhanced device poll was implemented on the Carrier MicroLink 2i controller in early 1998 (rev 5109) and is currently being used by several remote monitoring vendors. This version allows the user to select individual data items or "sets" of data (USDAs, humidity, controlled atmosphere, etc.) by sending a bit array to the controller. Multiple sensor resolutions are supported in this architecture as well.

In order to maintain a structure similar to the current device poll, all current data items (set-point, supply, return and alarms) shall be consistent with respect to position and offset within the response packet. All data items shall be included in the response packet in a fixed format. Individual data items shall not be selectable. The length of the response packet is such that it may be supported with one message on either the HDR or LDR systems.

##### **5.2.2.2.4.3 Backward compatibility**

The Enhanced device poll message will supplement the current device poll message. It shall not impact the current device poll in any manner and therefore be backward compatible to existing systems.

##### **5.2.2.2.4.4 Message Architecture**

The Enhanced device poll as shown in Figure 8 and Table 7 shall be assigned a command value of 47H (reference section 5.2.2.2.3, Table 2 "MDCU power line directed command types" of this International Standard). The current device poll command has a value of 40H.

NOTE The 47H command is currently not used and is assigned as the next available value in the table.

# Bits	12	12	12	4	1	3	20	12	12	12	12	12	12	12	12	12	12	12	12	12	12
Parameter	Set-point	Supply	Return	Operating mode	Recent Defrost	Alarm indicators	Alarms flags	US DA 1	US DA 2	US DA 3	Cargo	Humidity	Humidity set-point	O2 concentration	CO2 concentration	N2 concentration	CO2 set-point	O2 set-point	Future expansion	Future expansion	Future expansion
Notes	See Figure 7, 103 68	See Table 1	See Table 1	See Table 1	See Table 1	See Table 1	See Table 1	See Table 1	See Table 1	See Table 1											

Figure 8 — Enhanced device poll — Field definitions (total 29 bytes or 232 bits)

Table 7 — Data format definitions

Contents	# Bits	Res/Bit	Range	Offset	Notes
Supply	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
Return	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
Set-point	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
USDA1	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
USDA2	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
USDA3	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
Cargo	12	0,05 C	-125C to 75C	-125C	0<=-125C; FFF>= 79,75C
Humidity (sensor)	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
Humidity (set-point)	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
O <sub>2</sub> concentration	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
CO <sub>2</sub> concentration	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
N <sub>2</sub> concentration	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
CO <sub>2</sub> set-point	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
O <sub>2</sub> set-point	12	0,025 %	0 – 100 %	0 %	000 = 0 %, FA0 = 100 %, FA0 > FFF reserved
Future	12				
Future	12				
Future	12				

5.2.2.3 MDCU to MMU reply command types

5.2.2.3.1 General

The reply types from the MDCU to the MMU convey the success/failure status of the jobs processed by the MDCU, and are presented in Table 8 and explained in 5.2.2.3.2 to 5.2.2.3.6.

Table 8 — MDCU reply types

MDCU reply	
General valid	20H
MDCU reset	27H
General invalid	28H
Invalid NAK on power line	29H
Invalid no response on power line	2AH

### 5.2.2.3.2 General valid reply

The general valid reply from the MDCU to the MMU is used to indicate that the job, or command requested, as specified by the task number, was executed without error. Commands directed for MDCU interrogation will reply only with the MDCU status buffer. (The format in which the MDCU reset command will return the status buffer is similar to that for the other MDCU interrogation commands with the exception that it will contain the default parameters, and that the reply type and task number will be as described in 5.2.2.3.3.) The definition of this status buffer is given below. The status bytes used for the low data rate system only are prefixed with an L while the status bytes used for the high data rate system only are prefixed with an H.

First byte transmitted:

- Status0 — MDCU status is written during initialization of the MDCU. A bit which is set indicates that an error condition occurred during the particular test, as shown in Table 9. Since this is a read-only location, the error status cannot be reset by the MMU.

Table 9 — MDCU status state bit definitions

Bit	MDCU state
7-5	Reserved
4	RCD tests had an error:
3	Internal RAM error
2	External RAM error
1	EEPROM (NOVRAM) error
0	Program check sum error

- Status1-2 — MDCU software version (1) and subversion (2) status is a read-only location.
- Status-3 — This is a handshake byte, used by the MMU to reset internal MDCU buffers and also to prevent data tearing when multiple transfers of a particular buffer are obtained. Each bit is discussed below.
- Bit7 — This is MDCU UART OTS status (MMU communications the OTS mode (i.e. communicating with the MMU). This bit is a read-only bit, the MMU cannot set/reset it.
- Bit6-0 — This is at present undefined.
- Status4-5 — This includes available power line buffer size high (4) and low (5) bytes.
- LStatus6 — This indicates the maximum size for any data packet (or block) transmitted by the MDCU on the power line network. This number will never exceed the available buffer size. Its default is set at 128 or 80H.

- LStatus7 — This indicates the number of attempts the MDCU will perform for a particular job in which it received an invalid response on the power line from the RCD. The default is set at 02.
- LStatus8 — This indicates the number of retries required on the power line. Rollover OFFH to OOH will not occur. It can be reset using the MDCU parameters command, as described in 5.2.2.2.2.3.
- LStatus9 — This indicates the number of successful normal polls which required one retry on the power line. Rollover OFFH to OOH will not occur. It can be reset using the MDCU parameters command, as described in 5.2.2.2.2.3.
- LStatus10 — This indicates the number of successful normal polls which required two retries on the power line. Rollover OFFH to OOH will not occur. It can be reset using the MDCU parameters command, as described in 5.2.2.2.2.3.
- Status11 — This indicates the time delay base value used within an interactive command (5.2.2.2.3.4 and 5.2.2.2.3.9) that is required between the MDCU transmission of data to an RCD and the interactive poll used to obtain a response from the RCD. Each count provides a 0,1 s incremental delay.
- Status12-14 — This is at present undefined.

Commands directed for power line communications reply with data fields which are dependent on the specific command and, therefore, are discussed within the subclause relevant to each specific command.

#### 5.2.2.3.3 MDCU reset reply

The MDCU reset reply is a special case of the general valid reply for MDCU interrogated commands since it also returns the MDCU status buffer as described in 5.2.2.3.2. The status buffer, however, will contain the MDCU default parameters since the MDCU has undergone a hardware reset. The task number returned will also be the default reset task number, or 255 (OFFH). (Therefore, since the MDCU may be reset without MMU intervention via its deadman feature, this task number should be reserved for this function only, and not used by MMU/MDCU interactions.)

#### 5.2.2.3.4 General invalid reply

The general invalid reply from the MDCU to the MMU is used to indicate that the job, or command requested, as specified by the task number, could not be completed. Unlike the special negative replies used to indicate incomplete jobs on the power line (as described in 5.2.2.3.5 and 5.2.2.3.6), this reply is used for a non-specific general purpose negative reply. Examples of its use are as follows:

- a) unrecognized command type received by the MDCU;
- b) invalid packet length received by the MDCU;
- c) no response from a mapping request. This is used instead of a no response reply to distinguish a map from other processing. The map procedure has a much higher probability of collisions on the power line and since a collision is not distinguished from a no response, the no response is not used.

As with all negative replies, no data field is associated with the reply.

#### 5.2.2.3.5 Invalid NAK on power line reply

The invalid NAK reply indicates the unsuccessful completion of a job, or command requested, as specified by the task number. This job was intended for power line interactions. The attempts to complete the job failed such that, although the RCD responded to the MDCU at least once (retries included), invalid information was received by the MDCU. In most cases, the MDCU received negative acknowledges indicating that the RCD did not have valid information to return to the MDCU or that an RCD buffer was not currently available to receive the information from the MDCU. As with all negative replies, no data field is associated with the reply.

**5.2.2.3.6 Invalid no response on power line reply**

The invalid no response reply indicates the unsuccessful completion of a job, or command requested, as specified by the task number. This job was intended for power line interactions. The attempts to complete the job failed such that no response from the RCD was received by the MDCU even though all retries were attempted. The timeout period for each command type is dependent on which data rate modem is being accessed on the power line and is, therefore, described for the appropriate low data rate or high data rate modem (5.2.3 and 5.2.4 respectively). In most cases this indicates the absence of the addressed RCD on the power line. As with all negative replies, no data field is associated with the reply.

**5.2.2.4 RCD directed power line commands**

**5.2.2.4.1 General**

This group of commands requires communications with RCD devices over the power line communications network. These commands are buried in the data field of an MMU/MDCU message and are deciphered by the RCD. The format of this portion of the MMU/MDCU data field is referred to as the RCD message. The first byte of the RCD message field is the RCD command. Those commands at present supported by the RCD are given in Table 10, and each is described below in its respective subclause. Table 10 also indicates whether or not the RCD will buffer data in its RCD response buffer on acceptance of each command. The bytes which follow the command in the message, if required, are discussed for each command in the respective subclause. Both types of RCD, the RCD and the sRCD, process these RCD directed commands similarly.

**Table 10 — RCD directed command types**

RCD command	Value	Buffering
RCD reset	20H	No
RCD parameters write	22H	Yes
RCD address/set-point read	24H	Yes
RCD address write	26H	Yes
RCD set-point write	28H	Yes
RCD RAM absolute address read	30H	Yes
RCD RAM volatile offset address read	32H	Yes
RCD RAM battery offset address read	34H	Yes
RCD RAM datalog offset address read	36H	Yes
RCD RAM absolute address write	40H	Yes
RCD RAM offset address write	42H	Yes
RCD RAM battery offset address write	44H	Yes
RCD RAM datalog offset address write	46H	Yes

Since processing of the RCD message occurs after accepting the information from the MDCU over the power line, the RCD acknowledge, or acceptance, of the network command occurs at the network level and not at the RCD message level. For example, an RCD may accept a command from the MDCU with an invalid RCD message type for the RCD. Any processing of a message after such acceptance from the MDCU, even for a valid message type which the RCD finds illegal, will be ignored by the RCD. An RCD will not accept a valid RCD network command only if its RCD response buffer already contains data waiting to be returned to the MDCU from a previous RCD network command.

#### 5.2.2.4.2 RCD reset command

The RCD reset command shall cause the RCD to initialize itself in the same way as if a power-up sequence was performed. The RCD will perform its internal diagnostics and initialize with default parameters. Any data associated with this command will be ignored. Since the RCD is reset, no data response is buffered in the RCD response buffer.

#### 5.2.2.4.3 RCD parameters command

The RCD parameters read command loads the RCD status buffer with information passed in the message following this parameter's command byte. Since the RCD status buffer is only 15 bytes in length, a message containing more than 15 bytes of information will cause the RCD to ignore the command. The parameters to be loaded are defined below. The parameters used for the low data rate system only are prefixed with an L while the parameters used for the high data rate system only are prefixed with an H.

First byte transmitted:

- ParameterO — Test status byte is a read-only location and, therefore, this parameter is ignored by the RCD.
- Parameterl-2 — RCD software version/subversion is a read-only location and, therefore, this parameter is ignored by the RCD.
- Parameter3 — The MMU handshaking flag byte is changed. This byte is used by the MMU to reset internal RCD buffers and also to prevent data tearing when multiple transfers of a particular buffer are obtained. Each bit is discussed below.
- Bit7 — This is for an iRCD only, the UART OTS status (controller communications). Active is set to a logic one when the UART is in the OTS mode that is currently capable of transfers. This bit is a read-only bit; the MMU cannot set/reset it.
- Bit6 — This is the RCD device status buffer status. Set active indicates that the buffer contains information available to the MMU. The MMU may reset this status by transmitting logic zero in this bit position, while a logic one does not change the buffer's state.
- Bit5 — This is the RCD device response buffer status. Set active indicates that the buffer contains information available to the MMU. The MMU may reset this status by transmitting logic zero in this bit position, while a logic one will not change the buffer's state. For an iRCD, a buffer in the process of being loaded by the controller does not abort the transmission.
- Bit4 — This is the RCD device datalog buffer status. Set active indicates that the buffer contains information available to the MMU. The MMU may reset this status by transmitting logic zero in this bit position, while a logic one will not change the buffer's state. For an iRCD, a buffer in the process of being loaded by the controller does not abort the transmission.
- Bit3 — This is the RCD response buffer status. Set active indicates that the buffer contains information available to the MMU. This bit cannot be reset by this command since this command will actually buffer this status here.
- Bit2 — For a sRCD only, this is the RCD external portable data collection computer buffer status, if applicable. Set active to a logic one when either the external serial channel UART receive or the transmit buffer contains valid information. This information is not exchanged on the power line communications network. This bit can be reset by the MMU, but the serial channel communications are not affected.
- Bit1 — This is at present undefined.
- Bit0 — This is the RCD data-logger buffer update handshake bit. The RCD sets this bit whenever the RCD has updated its internal data-logger buffer. The MMU resets the bit prior to transfer and checks it after the transfer is complete, thus providing a means to check possible data tearing of the buffer.

- Parameter4-5 — Available buffer size high (4) and low (5) bytes are changed. Power line buffers are decreased/expanded. Zero indicates no change. If the new size is below/above the RCD's minimum/maximum value, its minimum/maximum value is used.
- LParameter6 — The maximum size of the data packets (or blocks) to be transferred on the power line is changed. If received larger than the available buffer size setting (described above), that value will be substituted. A value of zero will retain the previous setting (unless that setting is greater than the new available buffer size). It is recommended that the packet size not be set greater than its default value (128 or 80H) unless the message timeout settings within the RCD are changed accordingly.
- LParameter7 — The number of retransmissions to be attempted for a given job on the power line is changed.
- LParameter8 — The counter indicating the number of retries which have been performed on the power line is changed.
- LParameter9 — The current baud rate at which the iRCD is communicating with the controller is changed. If this requested rate is greater than the iRCD maximum rate given in LParameter10, this maximum rate is substituted. The corresponding baud rates are given in Table 5. For an sRCD communicating with an external device (if applicable), this is a read-only location.
- Lparameter10 — Maximum baud rate the iRCD can communicate with the controller or the sRCD can communicate with an external device, if applicable. The corresponding baud rates are given in Table 5. This is a read-only location.
- Bytes 11-14 — These are at present undefined.

Since the first 3 bytes of the status buffer are read-only locations, any data field containing less than 3 bytes will not change any parameters in the RCD status buffer. Also, Parameter4-5 is 2 bytes in length and a byte count which transfers only one of these bytes will not update this parameter. Regardless of whether any parameters were changed, once the RCD has accepted to process the information, the RCD will place the new status buffer into the RCD response buffer following the RCD status change reply byte (see 5.2.2.5.2).

#### 5.2.2.4.4 RCD address/set-point read command

The RCD address/set-point read command loads the RCD response buffer with the RCD address/set-point read reply byte (see 5.2.2.5.3) followed by the present contents of the RCD's non-volatile memory device, which contains the 11-byte RCD network address plus the 4-byte RCD set-point (5.2.2.4.6). If an invalid address is resident in the RCD's non-volatile memory device (i.e. this information's check sum is found to be erroneous), the universal error address of nine ASCII Os followed by two ASCII Is, or nine 3OHs plus two 3IHs, will be buffered. Since information is not required for this read command, any data received by the RCD with this command will be ignored.

#### 5.2.2.4.5 RCD address change command

The RCD address change command changes the assigned address of the particular RCD with information passed in the message following these address change command bytes. Since an RCD network address is 11 ASCII bytes in length, a message containing more than 11 bytes of information, or information received that is not ASCII numerics/alphanumerics, will cause the RCD to ignore the command. The RCD will also ignore the command if it has been addressed with the universal address and the initial address enable device jumper removed.

Regardless of whether any bytes within the address field were changed, once the RCD has accepted to process the information, the RCD will place the new 11-byte network address into the RCD response buffer following the RCD address change reply byte (see 5.2.2.5.4). The interactive commands should not be used when changing an RCD address while addressing that RCD with its individual address since the interactive poll command queued by the MDCU to obtain the RCD response buffer will contain the old or newly invalid RCD address.

#### 5.2.2.4.6 RCD set-point change command

The RCD set-point change command changes the assigned set-point values of the particular RCD with information passed in the message following this set-point change command byte. This set-point information consists of 4 bytes which are stored in a non-volatile memory device. At present, this information is not defined and, therefore, not used by the RCD. Since there are 4 bytes, only an information field of between 1 and 4 bytes will be processed by the RCD. Any other number of bytes received will cause the RCD to ignore the command.

Regardless of whether any bytes within the set-point field were changed, once the RCD has accepted to process the information, the RCD will place the number of set-point bytes transferred into the RCD response buffer following the RCD set-point change reply byte (see 5.2.2.5.5).

#### 5.2.2.4.7 RCD RAM read commands

The RCD read commands place a specified number of bytes in the RCD response buffer. Either an absolute or an offset starting address of these data is also specified. Starting addresses are calculated for each of these commands specifically as an offset from the first location of its designated buffer. Absolute addressing may be considered as a calculated offset address from an OOOOH base location. The assignment of addresses is hardware specific and is outside the scope of this International Standard.

Access to the data-logger buffer while it is being updated is not allowed. The RCD will not transfer the datalog information to its Internal RCD response buffer. Therefore, a NAK will be returned to the MDCU when trying to retrieve this information. Since the MMU communication and data-logging updates are independent, one or more blocks of data may be affected during an update. The transfer lockout ensures that if the MMU obtains a contiguous block of the data-logger, this will not result in data tearing. However, if the MMU shall obtain various portions of the data-logger, another lockout is required. A bit in the RCD status buffer (see 5.2.2.4.3 and 5.2.2.5.2) is used, in conjunction with the MMU, to provide an interlock. The MMU should reset this bit prior to obtaining multiple block area transfers and, when the transfers are completed, should check that it has remain unchanged. The RCD will set this bit when it updates any portion, or multiple portions, of the data-logger. An update which requires data in distinctly different areas of the data-logger (i.e. block 0 pointers and the block affected by those pointers) can be accessed with confidence by the MMU.

The RCD does not perform address checking on any given location and, therefore, it is the responsibility of the MMU to provide an existing area of memory. (Support peripherals on the RCD, such as an external UART, which is memory mapped, are also accessible.) The starting address and byte count are 16-bit parameters, passed in that order, following the RCD read commands, i.e.:

Message field to RCD = Command + Start address + Byte count

In the above, the 16-bit parameters are passed high byte followed by low byte. The RCD will not accept the command if the number of bytes in the messages does not equal five or the total length requested by the byte count is greater than the RCD response buffer.

Once the RCD has accepted to process the information, the RCD will place the number of bytes requested into the RCD response buffer following the RCD RAM read reply byte, the starting address parameter and the byte count (see 5.2.2.4.6).

#### 5.2.2.4.8 RCD RAM write commands

The RCD write commands place a specified number of bytes in RCD memory. Either the absolute or the offset starting location for placing these data is also specified. Starting addresses are calculated for each of these commands specifically as an offset from the first location of its designated buffer. Residences of these buffers are hardware specific and, therefore, outside the scope of this International Standard. All of these write commands except, possibly, some defined non-volatile memory, have the potential of destroying internal RCD processing procedures and, therefore, are not allowed unless the initial address enable device jumper is inserted. Since most of these writes are performed with the initial address enable device jumper inserted, the RCD does not perform address checking on any of these given locations and, therefore, it is the responsibility of the MMU to provide an existing area of memory. (Support peripherals on the RCD, such as external UART, which may be memory mapped, are also accessible.) The non-volatile memory write which may be reserved

to store input data directly from the MMU, however, may be performed during system execution and thus its range of address to be written shall be checked. If any byte to be transferred is outside this usable external range of memory, the command is aborted. The starting address and byte count are 16-bit parameters, passed in that order, following the RAM write command. The data to be written follows this information, i.e.:

Message field to RCD = Command + Start address + Byte count + Data

In the above, the 16-bit parameters are passed high byte followed by low byte. The RCD will not accept the command if the number of bytes in the message is not greater than five. Also, the RCD will not accept the command if the byte count does not match the number of data bytes received. Since the power line buffers are variable, dependent on their setting in their status buffers (see 5.2.2.4.3), the size of the message is limited.

Once the RCD has accepted to process information, the RCD will place the RCD RAM write reply byte, along with the starting address parameter and the byte count received, in the RCD response buffer to inform the MMU that the transfer has been accepted (see 5.2.2.5.7).

### 5.2.2.5 RCD directed power line slave replies

#### 5.2.2.5.1 General

The RCD message reply types from the RCD to the MDCU convey the successful status of a previously accepted RCD message command received by the RCD. This reply has been buffered in the RCD response buffer and shall be retrieved before the RCD can accept another RCD message. The buffer is retrieved using an RCD interactive poll command (see 5.2.2.2.3.6), either exclusively from the MMU, or inclusively via an RCD interactivel or RCD interactive2 command (see 5.2.2.2.3.4). The RCD returns the buffered data in text block packets over the power line communications network to the MDCU. The first byte of this RCD message field is the RCD reply type. The reply types currently supported by the RCD are given in Table 11, and each is described below in its respective subclause. Note that the reply has the same value as its corresponding command presented in 5.2.2.4 with the least significant bit set. The bytes which follow the reply type in the message are discussed for each reply in the respective subclause.

**Table 11 — RCD directed slave reply types**

RCD reply	Value
RCD status change	23 H
RGD address/set-point read	25H
RGD address change	27 H
RCD set-point change	29 H
RCD RAM absolute address read	31H
RCD RAM offset address read	33H
RCD RAM battery offset address read	35H
RCD RAM datalog offset address read	37H
RCD RAM absolute address write	41 H
RCD RAM offset address write	43H
RCD RAM battery offset address write	45H
RCD RAM datalog offset address write	47H

On receipt of the RCD text message from the RCD, the MDCU will forward the information to the MMU. This is returned in the data field of a general valid reply (see 5.2.2.3.2), preceded by the RCD routing byte/address of the RCD interactive poll command. The full data field to the MMU is, therefore:

Data field to MMU = Routing byte/address + Reply command + RCD message data

The individual replies discussed below, however, will be in terms of the RCD message field, which is in relation to the RCD/MDCU transfer. The information received at the MMU will have the routing byte/address added.

#### 5.2.2.5.2 RCD status change reply

The information returned to the MDCU from the RCD following an RCD parameters command (see 5.2.2.4.3) contains the RCD status. This information is transferred from the RCD status buffer to the RCD response buffer immediately after the parameters are changed in the status buffer. Therefore, the delay in retrieving this reply is proportional to the age of the valid data (but obviously this relates only to those bytes being updated in RCD processing). Except for this aging of buffered data, the information returned is the same as that returned by the RCD for a network RCD poll command (see 5.2.2.2.3.2), i.e. the RCD status buffer definitions. One other slight difference between the two responses is that this reply precedes the status buffer information with the reply type byte, i.e.:

Message field to MDCU = Reply command + RCD status

#### 5.2.2.5.3 RCD address/set-point read reply

The information returned to the MDCU from the RCD following an RCD address/set-point read command (see 5.2.2.4.4) contains the 11-byte RCD network address followed by the RCD set-point bytes, all of which are stored in an RCD non-volatile memory device. This information is transferred to the RCD response buffer immediately after the command is accepted. This 15-byte ASCII information is returned to the MDCU preceded by the reply type byte, i.e.:

Message field to MDCU = Reply command + RCD network address + RCD set-point

#### 5.2.2.5.4 RCD address change reply

The information returned to the MDCU from the RCD following an RCD address change command (see 5.2.2.4.5) contains the newly accepted RCD network address. This information is transferred from the RCD non-volatile memory device to the RCD response buffer immediately after the non-volatile memory device has been updated. This 11-byte ASCII information is returned to the MDCU preceded by the reply type byte, i.e.:

Message field to MDCU Reply command + RCD's new network address

#### 5.2.2.5.5 RCD set-point change reply

The information returned to the MDCU from the RCD following an RCD set-point change command (see 5.2.2.4.6) contains the newly accepted set-point byte or bytes. This information is transferred from the RCD non-volatile memory device to the RCD response buffer immediately after the non-volatile memory device has been updated. The 1 to 4 bytes of information are returned to the MDCU preceded by the reply type byte, i.e.:

Message field to MDCU = Reply command + RCD's new set-point data

#### 5.2.2.5.6 RCD RAM read replies

The information returned to the MDCU from the RCD following an RCD RAM read command (see 5.2.2.4.7) contains the reply type byte, the starting or offset address and byte count received in the request, and the corresponding number of bytes of text data requested. The starting address and byte count are both 16-bit parameters and are returned as verification that the text data requested is that returned. Thus, the format of the RCD reply is:

Message field to MDCU = Reply command + Starting address + Byte count + Data

In the above, the starting or offset address and byte count are transferred high byte followed by low byte, and the number of data bytes is given by the byte count.

### 5.2.2.5.7 RCD RAM write replies

The information returned to the MDCU from the RCD following an RCD RAM write command (see 5.2.2.4.8) contains the reply type byte and the starting or offset address and byte count received in the request. The starting address and byte count are both 16-bit parameters, transferred high byte followed by low byte. Although the write command does not require the RCD to return actual text data, this reply is nevertheless used as a verification of the next data written into RCD memory. Thus, the format of the RCD reply is:

Message field to MDCU = Reply command + Starting address + Byte count

### 5.2.2.6 Device directed power line commands

This group of commands, like the RCD directed commands discussed in 5.2.2.4, requires communications with RCD devices over the power line communications network. Similarly, these commands are buried in the data field of an MDCU/MMU message and are deciphered by the RCD. Processing of the commands differs slightly, however, since this set of commands is used to interact directly with an iRCD's controller or the sRCD's input/output functions. Therefore, the format of this portion of the MDCU/MMU data field is referred to as a device message. The device message structure is given as:

Device message to MDCU = Return byte count + Device command + Data field

In the above, the return byte count is a 2-byte value (high byte first) which provides the RCD with the number of bytes the MMU will expect in the reply. This value is equivalent to the reply type byte count (which is one) plus the number of data bytes in the reply message. The return byte count is passed in the device message so that an iRCD does not have to fully decipher the contents of the device message, i.e. this message is intended specifically for the controller. By providing this count, the iRCD interacts correctly with the controller with only a general knowledge of the overall RCD to controller protocol. The RCD provides the SYNC characters preceding the Device command + Data field and follows this with the proper CRC characters of the protocol. It also checks, with simple masking, that the proper reply type is received for a given transmitted device command. The particular device command and its associated data field need not be deciphered. To imitate the RCD/controller's format, the message structure of the sRCD will be the same. Therefore, the device message will also contain the same information. Since the sRCD may only support a limited number of controller functions, those functions not supported by an sRCD are simply not utilized (an sRCD will ignore any received field which is not supported and return zeros in any field in a return message which is not utilized). Since processing is internal to the sRCD and does not require a communications medium to a controller, the sRCD also checks that the return byte count is the expected number of bytes to be returned in the reply message. If not, the sRCD should ignore the device command. The sRCD will process the device commands which it supports as does the controller (see 5.4). Since processing of the device message occurs after accepting the information from the MDCU over the power line, the RCD acknowledge, or acceptance, of the network command occurs at the network level and not at the device message level. For example, an RCD may accept a command from the MDCU with an invalid device message type for the RCD. Any processing of a message after such acceptance from the MDCU, even for a valid message type which the RCD finds illegal, will be ignored by the RCD. An RCD will not accept a valid device network command only if its device response buffer already contains data waiting to be returned to the MDCU from a previous network command.

### 5.2.2.7 Device directed power line slave replies

The device message reply types from the RCD to the MDCU convey the successful status of a previously accepted device message command received by the RCD. This reply has been buffered in the device response buffer and shall be retrieved before the RCD can accept another device message. (This buffer can be reset using the RCD parameters command, see 5.2.2.4.3.) The buffer is retrieved using a device interactive poll command (see 5.2.2.2.3.11), either exclusively from the MMU, or inclusively via a device interactive1 or device interactive2 command (see 5.2.2.2.3.9). The RCD returns the buffered data in text block packets over the power line communications network to the MDCU.

The first byte of this device message field is the device reply type. It should be noted that the reply has the same value as its corresponding command with the most significant bit (7) set. The bytes which follow this reply type constitute the data field of the reply message and are defined in 5.4.

On receipt of the device message text from the RCD, the MDCU will forward the information to the MMU. This is returned in the data field of a general valid reply (see 5.2.2.2.3.2), preceded by the RCD routing byte/address of the device interactive poll command. The full data field to the MMU is therefore:

Data field to MMU = Routing byte/address + Reply command + Data field

### 5.2.3 MDCU to LRCD communications

#### 5.2.3.1 General

The physical requirements of LDCU communications with the slave LRCDs are described in 5.5. This serial communications system consists of a single control station, called an LDCU, which initiates all transactions (initial master device) to interrogate a multiple tributary stations environment (slave devices). The tributary stations are called LRCDs. Control of mastership is transferred to a selected tributary device when the control station requests information. The information is returned to the control station only, i.e. no tributary-to-tributary communications are allowed. Completion of the transfer returns control of mastership back to the LDCU control station.

#### 5.2.3.2 LDCU power line communications components

##### 5.2.3.2.1 General

There are four basic components available to the transmission function on the power line. These include the preamble, the communications control characters, the prefix and the data. All transmissions shall be initiated by the preamble and shall contain some type of control character(s). A prefix or data field may or may not be present in a transmission.

##### 5.2.3.2.2 Preamble

The preamble delimits the start of a valid message. It consists of a variable number (minimum of three) of transitions followed by 10 or more logic one bits and is required to synchronize communications between devices on the power line. This provides a unique sequence of bits which is illegal for normal information transfers since each byte transferred contains a start bit (low logic level) followed by a maximum of nine logic one bits. The occurrence of this sequence shall force re-synchronization (at any instance within a transfer) of receiving slave devices on the power line. Thus, the deadtime between any two bytes in a transfer is restricted to less than one-half bit length.

##### 5.2.3.2.3 Communications control characters

###### 5.2.3.2.3.1 General

Ten ASCII communications control characters are reserved to provide control functions during communications. These characters are given in Table 10 and described in 5.2.3.2.3.2 to 5.2.3.2.3.11.

Table 12 — Network control characters

Control character	HEX value	Definition
SOH <sup>a</sup>	01	Start of heading
STX	02	Start of text
ETX <sup>a</sup>	03	End of text
EOT	04	End of transmission
ETB <sup>a</sup>	17	End of transmission block (packet) in a multiple data block transfer
ENQ	05	Enquiry
ACK <sup>a</sup>	30 or 31	Acknowledgements: 30H = ACKO 31H = ACKI
NAK	15	Negative acknowledge
SYNC	16	Synchronous idle
DLE	10	Data link escape

<sup>a</sup> Control function is provided only when preceded by a DLE control character (termed a control character sequence).

#### 5.2.3.2.3.2 SOH (Start of heading)

The DLE SOH control character sequence is transmitted by the master device at the beginning of the data portion of a transfer to delimit the start of a message heading. This heading consists of addressing or routing information. The heading may or may not be present for a data portion of a transfer.

#### 5.2.3.2.3.3 STX (Start of text)

There are two ways in which the DLE STX control character sequence is used. The first condition is when the optional header and its corresponding DLE SOH control character sequence are used. In this case the function of the DLE STX control character sequence is used as a delimiter which terminates the header information field and signals the beginning of the text information portion of the data transfer (see 5.2.3.2.3.2).

The second condition occurs when the optional header is not present. In this case, the DLE STX simply signals the beginning of the text information portion of the data transfer. In both cases, the DLE STX control character sequence is always used to indicate the start of the transfer of text information in either a single or multiple block transfer.

The text portion of a transfer is a sequence of characters which is to be treated as an entity and is transmitted in its entirety to the ultimate destination. The DLE STX shall always be present in the data portion of a text transfer, although it may not appear in the first block(s) when header information is included in a multiple block transfer.

#### 5.2.3.2.3.4 ETX (End of text)

The DLE ETX control character sequence is transmitted by the master device at the end of the data portion of a transfer to terminate the transfer of information which contains one or more blocks of heading, if present, and text. A block of information which transfers a single heading and text block, or a single text block (i.e. all of the information is transferred in a single block), is terminated by this control sequence.

**5.2.3.2.3.5 EOT (End of transmission)**

The EOT control character is transmitted by the master device (or by the LDCU control station to re-establish control of the power line network) to terminate a transmission that may have contained one or more blocks of texts and any associated heading. The EOT shall immediately follow a preamble, i.e. no prefix or other control may precede it. Detection of the EOT control requires all tributary stations to return to their receive mode, thus allowing the LDCU control station to become the master, or only device capable of initiating a transmission.

**5.2.3.2.3.6 ETB (End of transmission block)**

The DLE ETB control character sequence is transmitted by the master device at the end of the data portion of a transfer to terminate a sequence of characters for any block except the last of a multiple block transfer. This block transfer started with the DLE SOH or DLE STX control character sequence.

**5.2.3.2.3.7 ENQ (Enquiry)**

The ENQ control character is transmitted by the LDCU control station to initiate a polling session to a tributary station. It is also transmitted by a master device to request a slave device receiving a data transfer to retransmit its previous ACK/NAK sequence. In these cases, the ENQ control character is preceded by the prefix. A DLE ENQ control character sequence within a data block portion of a transfer signifies to the receiving slave device that the transmitting master device is aborting the data heading/text transfer. In this instance, the receiving slave device should discard the block and return a NAK response to the aborted data block transfer.

**5.2.3.2.3.8 ACK (Acknowledgement)**

The DLE ACK control character sequence is transmitted by a slave device to acknowledge the acceptance of the tributary station addressed by the prefix/ENQ control character sequence within a selection with a response handshaking procedure (see 3.2.2.2). The tributary station will acknowledge with an ACKO (DLE followed by a 30H). A DLE ACK control character sequence is also transmitted by a slave device to acknowledge the receipt of a block of information in the data portion of a transfer. The first block is acknowledged with an ACKI (DLE followed by a 31H), while successive blocks of heading/text data are acknowledged with subsequently alternating ACKO and ACKI. In all cases, the DLE ACK control character sequence is preceded by the prefix.

**5.2.3.2.3.9 NAK (Negative acknowledge)**

The NAK control character is transmitted by a slave device to acknowledge negatively the receipt of a block of information in the data portion of a transfer. The transmitting device of the block transfer (and thus the receiving device of this NAK control sequence) may either retransmit the previous block of information or terminate the transfer with an EOT control sequence. The NAK control character is always preceded by the prefix.

**5.2.3.2.3.10 SYNC (Synchronous idle)**

The SYNC control character is reserved as a "time-fill" character during periods in a synchronous communications system when no other characters are available for transmission. Since the power line communications medium is asynchronous, this control character is not required and should cause the receiving slave device to abort reception of the present information packet.

**5.2.3.2.3.11 DLE (Data link escape)**

The DLE control character is used to distinguish a small number of characters as either control or data. A DLE character shall precede an SOH, STX, ETX, ETB and ACK character to allow these characters to perform as control characters. A DLE character preceding another DLE character indicates a single DLE data character. Thus, a single DLE character in a sequence of characters indicates that the character to follow provides a control function, while two DLE characters in sequence require the removal of one in the data block and the use of the second as a data character. The DLE sequence will only be found in the data portion of a transfer

and is termed ‘DLE stuffing’. Thus, a single DLE character followed by an indistinguishable control character in a data portion of a transfer shall cause the receiving device to abort the reception of the data.

**5.2.3.2.4 Prefix**

The prefix is a string of 13 ASCII alphanumeric characters (this ensures that these prefix characters can be distinguished from the communications control characters specified in 5.2.3.2.3). The prefix is arranged in the following format:

Message type/Routing byte/Network address

The prefix is always followed by a control character or a control character sequence. The prefix fields are defined as follows:

- Message type — ASCII select control byte. Directs which type of transfer is being requested by the LDCU control station (see Table 13).
- Routing byte — ASCII routing byte. Selects a particular LDCU requested modification on the communications medium which the responding LRCD will use. The definition of the bits for the low data rate network is given in Table 13.

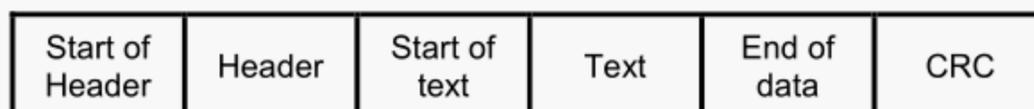
**Table 13 — Low data rate routing byte definitions**

7	6	5	4	3	2	1	0
Modem type		Standard specific					
01		0 = ISO		00 = 1 200 baud 01 = Reserved 10 = Reserved		Burst rate capability of the LRCD taken from Table 5 with offset of + 4	
				11 = Maximum burst			
01		1 = MFG		Manufacturer specific modifications			

- Network address — 11 byte ASCII characters defining the address of (a) tributary station(s) on the power line network. The bytes are categorized as the container ISO number with a 4-byte alpha field followed by two 3-byte numeric fields (numeric1 and numeric2) and a 1-byte check digit. A universal address consisting of all ASCII Os, or all 30H, is recognized by all LRCDs (i.e. a broadcast address) while a field containing nine ASCII Os followed by an ASCII I and another ASCII 0, or nine 30Hs plus 31H plus 30H, is an address substituted in an LRCD containing an invalid personal address. Response to the initial transfer establishment requiring a prefix (such as ACK or NAK) will return the same information.

**5.2.3.2.5 Data**

The data portion of a transfer consists of the information to be exchanged between two devices (master to slave) on the power line network. Control character sequences are used to initiate and terminate various portions of the information transfer. A transfer of information may be subdivided into multiple blocks, or packets, and transferred. The structure of a block is as shown in Figure 9.



**Figure 9 — Block structure**

The data fields in Figure 9 are defined as follows:

- Start of header — A DLE SOH control character sequence which is present if a header field is to be included in the data block.
- Header — Variable length field consisting of addressing or routing information. This information is transferred in raw data form and requires DLE stuffing. A heading may or may not be present. A header shall be followed by text data in a transfer. However, (a) block(s) in a multiple block transfer may contain only header information, but shall be followed by text data information.
- Start of text — A DLE STX control character sequence which is present if a text data field is to be included in the data block. If only a header is transferred in a particular block of a multiple block transfer, this field will not be present.
- Text — Variable length field consisting of the information required to be transferred from the master device to a slave device on the power line network. This information is transferred in raw data form and requires DLE stuffing. The contents are dependent on the type of message to be processed. The contents of this field are described in 5.2.2.
- End of data — A DLE ETX or DLE ETB control character sequence used to terminate the information portion of a data transfer. The DLE ETB is used to terminate a data sequence of characters for any block except the last of a multiple block transfer. The DLE ETX is used to terminate the transfer of information which contains one or more blocks of heading, if present, and text. This includes a block of information which transfers a single heading and text block, or a single text block (i.e. all the information is transferred in a single block).
- CRC — A 16-bit error check field generated by using the CRC-16 polynomial. The CRC generation is started by the DLE SOH control sequence or, if not present, the DLE STX sequence. It is terminated by either the DLE ETX or the DLE ETB control sequence. The DLE included in any control sequence, or one of the DLE characters used for DLE stuffing, is not included in the generation. The SOH character is not included in the generation either. The STX character is included only if a header, and, therefore, an SOH control character, is present. The ETX or ETB character is also included in the calculation.

### 5.2.3.3 Power line communications sessions

#### 5.2.3.3.1 General

Communications between the LDCU control station and LRCD tributary stations are always initiated by the LDCU. A session begins with the LDCU having a master status and none of the tributary stations having slave status. The LDCU master initiates power line communication by various methods as determined by the message type field. The message types available at present for power line communications are given in Table 14.

**Table 14 — Prefix message types**

Network command	Transaction
Poll	The control station polls a particular tributary station to receive basic information.
Xmit1	Xmit data from the control station to a particular tributary station using fast select.
Xmit2	Xmit data from the control station to a particular tributary station using selection with response.
Interactive1	Xmit1 then, after a delay, interactive poll the same tributary station to receive response.
interactive2	Xmit2 then, after a delay, interactive poll the same tributary station to receive response.
Map	Map tributary stations on the network.
Interactive poll	The control station polls a particular tributary station to receive information from a previous Xmit.

The LDCU control station only relinquishes mastership to a tributary station, and thus can obtain information, following the transmission of a polling sequence (poll, map or interactive poll, although the interactive1 and interactive2 use an embedded interactive poll sequence). The xmit sequences (including those embedded in the interactive1 and interactive2 sequences) only transfer information to a tributary station and do not therefore relinquish mastership.

The 1 or 2 in xmit and interactive indicates the type of LDCU/LRCD handshaking select sequence used to transmit the data. The 1 indicates a fast select sequence, while the 2 indicates a selection with response sequence. The fast select method of transfer simply sends the data to the addressed LRCD tributary station along with the addressing information (i.e. a prefix preceding the first block of data without the ENQ control character). The selection with response initially selects a tributary station using a polling process (i.e. a prefix followed by an ENQ character). Once the tributary station acknowledges its selection, the (first) block of data is transferred. From the point of view of the MMU, the differences in these processes are invisible, except for the time of response.

The type of power line command processing, e.g. select with response versus fast select or xmit versus interactive, should be carefully selected to coincide with the handling of the LRCD/device message. For instance, a command not required to buffer, and therefore not to return any data to the LDCU, should use the xmit power line command so that the LDCU may respond with the LRCD's acknowledge/nacknowledge reply to the MMU. If an interactive power line command that did not require the LRCD to buffer a response was sent, the LRCD would reply with an ACK or a NAK. Then when the LDCU requested the buffered response, the buffer would be empty and the LRCD would reply with a NAK. The MMU would not know whether the NAK came from the command or the request for the buffered response. Passing this status onto the MMU would not determine whether the LRCD had accepted the command or not, i.e. whether the negative acknowledge came from the first (interactive) or second (interactive poll) command.

The data portion of the transfer session, which comprises the information discussed in 5.2.3.2.5, is accomplished similarly in all message types. A block of information is transferred from the master device to the slave device. An unsuccessful reception of the information will cause the slave device to return a NAK prefix/control character sequence, thus allowing the master device either to retransmit or to abort with an EOT control sequence. A successful reception by the slave device will return the proper ACK prefix/control character sequence, thus allowing the master device to transmit the next data block of information, if required. If all data have been transmitted, the master device terminates the interaction with an EOT control character sequence, returning mastership to the LDCU control station. Also, a master device may abort the data transfer at any instance by transmitting a DLE ENQ control character sequence within a data block portion of a transfer.

The LDCU shall provide a timeout or waiting period for each expected response from the LRCD. These time periods are used for each retry, i.e. the timeout shall occur for each transaction on the power line; it is not accumulative over all the retries of a single command. A timeout will require the LRCD to transmit a preamble/EOT control sequence onto the power line network to abort all network processes and to allow the LDCU to regain mastership. The timeout period for each command type, and its expected response type, are given in Table 15.

**Table 15 — Power line timeout variables**

Command type	Timeout period	LDCU's expected LRCD response
Poll	0,68 s	Poll data
Xmit	0,68 s	Acknowledge
Interactive	0,68 s	Acknowledge
Interactive poll	4,50 s	Message
Map	0,60 s	Map data

### 5.2.3.3.2 Poll

The poll message session allows the LDCU control station to obtain basic information from an RCD tributary station. The LDCU will address the LRCD by transmitting the prefix immediately followed by an ENQ control character. Mastership is assumed by the addressed LRCD and text data are transmitted to the LDCU. (If a broadcast network address is used to select a tributary station, all stations will attempt to respond.) An example showing the flow of information is presented in Figure 10. This example assumes no DLE SOH/header fields (which, if present, would occur between the preamble and DLE STX in line 2). Also, a single block contains all the text data to be transferred. An invalid reception by the LDCU from line 2 would have resulted in a preamble/prefix/NAK transfer by the LDCU in line 3. The LRCD would then retry the transmission of line 2 or terminate the session with line 4.

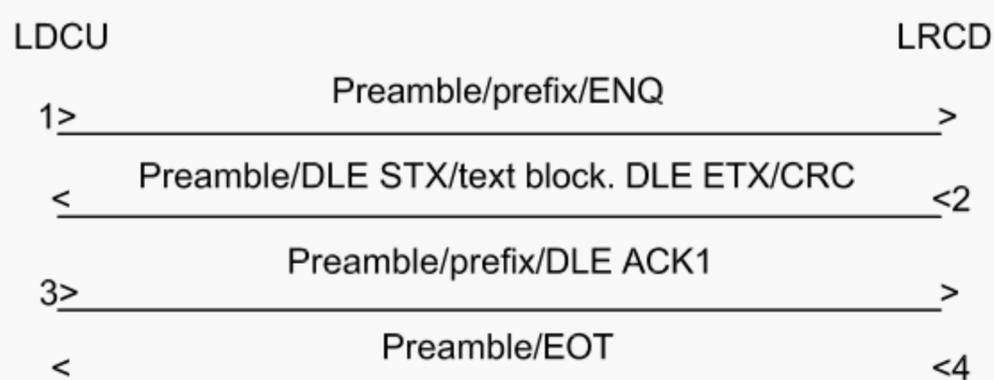


Figure 10 — Poll session transfer diagram

### 5.2.3.3.3 Xmit

The xmit1 message session transfers text data from the LDCU control station to a selected LRCD tributary station using the fast select process. This fast select method of transfer simply sends the text data to the addressed LRCD tributary station along with the addressing information, i.e. a prefix is transmitted (and not followed by an ENQ control character) immediately preceding the first block of data. Since the LRCD only receives data, it never obtains mastership of the power line. An example showing the flow of information is presented in Figure 11. Note that the negative acknowledgement by the LRCD in line 4 requires the LDCU to repeat line 3 in line 5. The acknowledgement for line 5 remains the same (ACKO) as would have been sent in line 3.

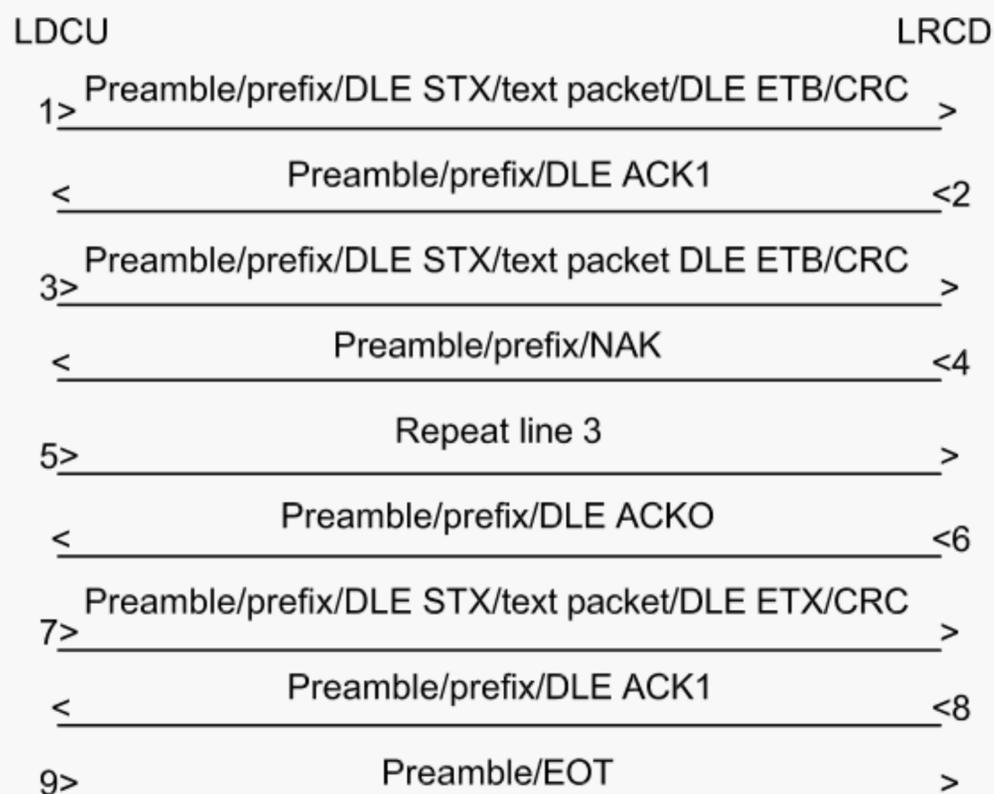


Figure 11 — Xmit1 session transfer diagram

5.2.3.3.4 Xmit2

The xmit2 message session transfers text data from the LDCU control station to a selected LRCD tributary device using the selection with response process. This selection with response method of transfer initially selects an LRCD using a polling process (prefix followed by an ENQ character). Once the tributary station acknowledges its selection, the block(s) of data is(are) transferred. Since the LRCD only receives data, it never obtains mastership of the power line. An example showing the flow of information is presented in Figure 12.

5.2.3.3.5 Interactive1

The interactive1 message transfer initiates an xmit1 message transfer session (see 5.2.3.3.3). The LDCU control station then provides a delay before initiating an interactive poll session (see 5.2.3.3.8) to the same tributary station to obtain information requested and buffered from the xmit1 session. The LDCU control station is free to communicate with other tributary stations during the delay period between the xmit1 session and interactive poll session to this tributary station. The flow of information is thus a dual function and can be exemplified by Figure 11 followed by Figure 15.

5.2.3.3.6 Interactive2

The interactive2 message transfer initiates an xmit2 message transfer session (see 5.2.3.3.4). The LDCU control station then provides a delay before initiating an interactive poll session (see 5.2.3.3.8) to the same tributary station to obtain information requested and buffered from the xmit2 session. The LDCU control station is free to communicate with other tributary stations during the delay period between the xmit2 session and interactive poll session to this tributary station. The flow of information is thus a dual function and can be exemplified by Figure 12 followed by Figure 15.

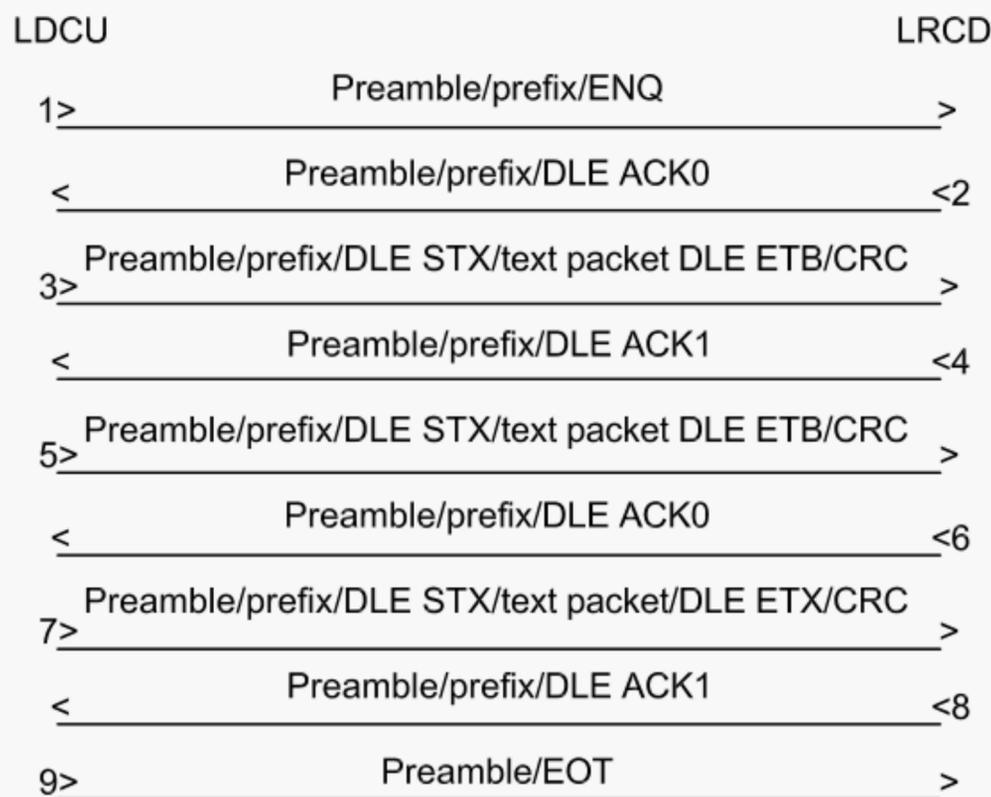
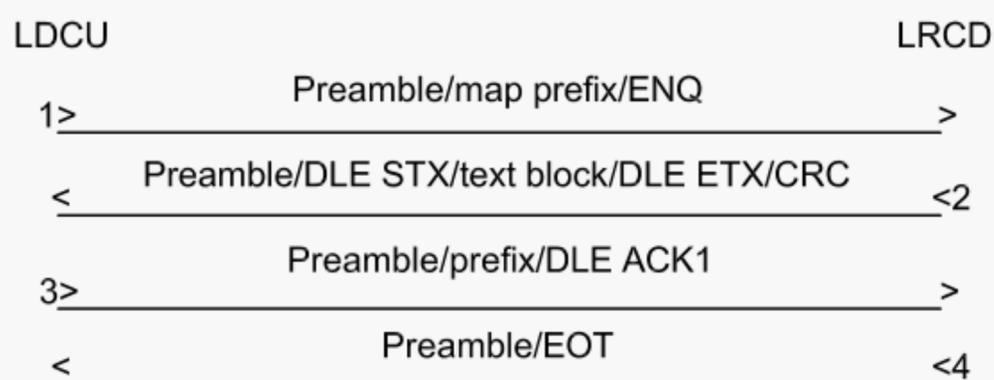


Figure 12 — Xmit2 session transfer diagram

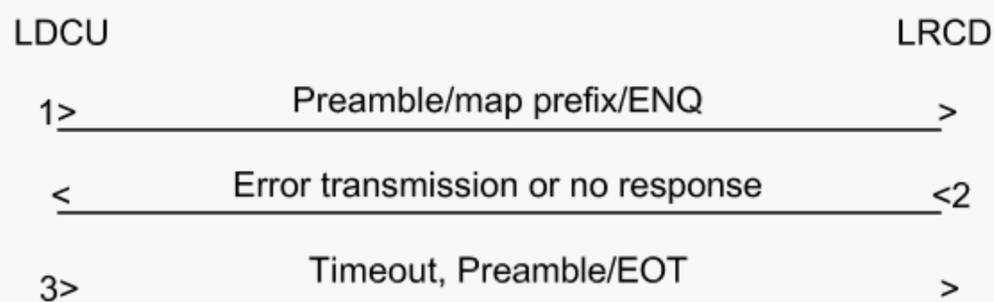
5.2.3.3.7 Map

The map message session broadcasts a polling session from the LDCU control station requesting, with varying probabilities, an LRCD tributary station to return its routing byte/network address. The purpose of this command is to log in, or map, any newly acquired or unpolled LRCDs on the power line network.

Since this mapping technique uses a polling type session on the network, the format of the information transmitted from the LDCU is similar to the poll message session described in 5.2.3.3.2, i.e. a prefix followed by the ENQ control character. However, since it is a broadcast type of message, it cannot simply select a particular LRCD tributary slave by using a unique individual network address. Instead, the 11-byte ASCII network address field described in 5.2.3.2.4 is broken into ASCII subfields supplying all LRCDs on the power line network with a mapping strategy (as defined in 5.2.3.2.3.5). An individual LRCD selected by this mapping strategy obtains mastership of the power line network and transmits a text data block containing its unique routing byte/network address to the LDCU. Since, however, this type of broadcast technique inherently suggests the possibility of selecting more than one tributary station, multiple masters may temporarily drive the power line communications medium. This results in an indistinguishable reception in the LDCU slave unit. In this instance, the LDCU slave transmits an EOT control character sequence to regain master status rather than returning the NAK prefix/control character sequence. A distinguishable transmission reception indicates a single tributary station has mapped onto the system and thus the LDCU slave returns an ACK prefix/control character sequence. Examples of the flow of information for an identified and an unidentified map are presented in Figure 13 and Figure 14 respectively. The information in the text data block will contain the individual LRCD 12 byte ASCII routing byte/network address.



**Figure 13 — Identified map session transfer diagram**



**Figure 14 — Unidentified map session transfer diagram**

#### 5.2.3.3.8 Interactive poll

The interactive poll message session is used by the LDCU control station to retrieve information from an LRCD tributary station after the LRCD has been sent an xmit1 or xmit2 message session (see 5.2.3.3.3 and 5.2.3.3.4 respectively) containing information which required the LRCD to buffer response information. This process is executed in the same manner as the poll message session of 5.2.3.3.2. The LDCU will address the LRCD by transmitting the prefix immediately followed by an ENQ control character. Mastership is assumed by the addressed LRCD and the buffered text data are transmitted to the LDCU. An example of the flow of information is presented in Figure 15.

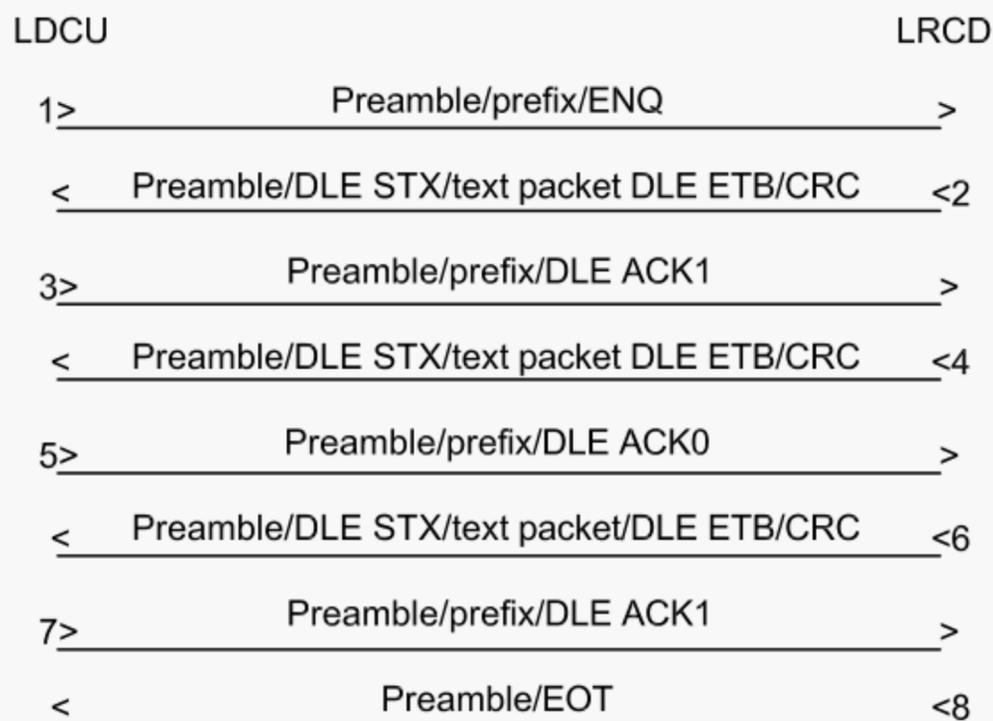


Figure 15 — Interactive poll session transfer diagram

## 5.2.4 MDCU to HRCD communications

### 5.2.4.1 General

The physical requirements and low level data link protocol for HDCU communications with the slave HRCDs is described in 5.5. Two basic types of communication are possible between an HDCU and an HRCD: data packet exchanges (see 5.5.9.3 and 5.5.9.4) and control exchanges (see 5.5.9.5). Data packet exchanges transfer information from a logically higher layer in an MMU/HDCU or controller/HRCD to its peer layer in an HRCD/controller or HDCU/MMU; control exchanges are used by an HDCU to grant an HRCD temporary access to the power line network to perform a data packet exchange. Control of the power line network then returns to the HDCU.

### 5.2.4.2 HDCU(HRCD power line communications components)

#### 5.2.4.2.1 General

The fields transferred for both a data packet exchange and a control exchange are discussed in 5.2.4.2.2 and 5.2.4.2.3. The actual format of the transfer over the power line, including the low level data link control bytes, is detailed in 5.5.

#### 5.2.4.2.2 Data packet exchange

##### 5.2.4.2.2.1 General

The actual data transferred during a data packet exchange contain three components: the data packet header, the data field to be transferred, and the packet check sum field:

| Header Data | Check sum

The protocol governing the exchange is as described in 5.5.9.3 and 5.5.3.

#### 5.2.4.2.2 Data packet header

The data packet header contains the fields required for transfer of the data packet to the designated destination. They are as follows:

| Destination address | Source address | Packet length

- Destination address — A 2-byte network address in the range 0000 to FFFE (hexadecimal) selecting a unique HRCD (0001 to EFFE) or the HDCU (0000). Address 0000 is reserved for the HDCU address and address FFF (hexadecimal) is reserved for the broadcast address. Addresses are assigned to the HRCDs by the HDCU during the login procedure. The HDCU maintains an internal lookup table associating each 11-byte ASCII RCD address (Container ID) on the power line network with a unique destination address.
- Source address — The unique 2-byte network address in the range 0000 to FFFE (hexadecimal) specifying the transmitting HRCD (0001 to FFFE) or HDCU (0000).
- Packet length — A 2-byte unsigned integer which gives the length of the data field.

#### 5.2.4.2.3 Data field

The data field contains the information to be transferred to the destination HRCD or HDCU. The format of the data field is as follows:

| Message type | Routing byte | Network address | Data |

- Message type — A 1-byte command directing what type of transfer is being requested by the MMU and corresponding to the MMU/MDCU xmit type command listed in Table 2.
- Routing byte — ASCII routing byte defined as shown in Figure 16.
- Network address — 11-byte ASCII characters defining the address of (an) RCD(s) on the power line network, i.e. the container ISO number as defined in 4.2.2.4.2.
- Data — Variable length data field for which the contents are dependent on the type of message to be processed. The contents are defined in 5.2.2.4.

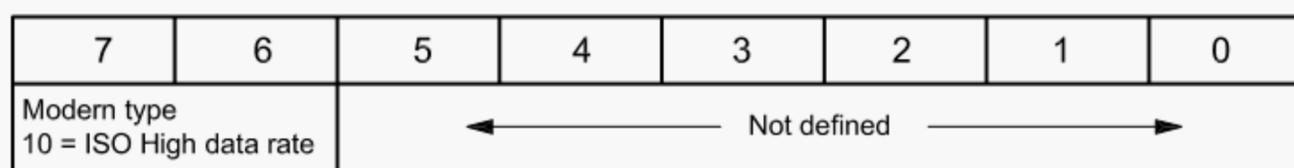


Figure 16 — Routing byte definition

#### 5.2.4.2.4 Packet check sum field

The packet check sum field contains an 8-bit exclusive-or check sum of all data packet header and data fields used in addition to the (32, 8) ECO and EDC encoding (see Table 17) used on each byte of all the data packet header and data fields.

#### 5.2.4.2.5 Control exchange

The HDCU uses a control exchange to grant an HRCD temporary access to the power line network to perform a data packet exchange. Control of the power line network then returns to the HDCU. The header field consists of the destination address only. The data field is empty.

| Header data | Destination address

The protocol governing the exchange is as described in 5.5.9.5.

5.2.4.3 HDCU/HRCD communications

5.2.4.3.1 General

Communications between the HDCU and HRCDs are always initiated by the HDCU. The HDCU uses the message type field to determine the type of session. The 14 message types available are grouped into five different categories as given in Table 16.

Table 16 — Message types

Message types	Transactions
RCD/device poll	The HDCU uses a data packet exchange to request basic information. It follows with a control exchange to retrieve the requested information.
RCD/device xmit1/2	The HDCU uses a data packet exchange to transmit data to an RCD/controller.
RCD/device interactive1/2	RCD/device xmit1/2 then, after a delay, the HDCU uses a control exchange to retrieve the RCD/controller response.
RCD/device map	The HDCU uses a broadcast data packet exchange, a data packet exchange and a control exchange to log in new RCDs on the power line network.
RCD/device interactive poll	The HDCU uses a packet exchange to transmit this request to the HRCD. It follows with a control exchange to retrieve information from an HRCD from a previous RCD/device xmit1/2.

5.2.4.3.2 RCD/device poll

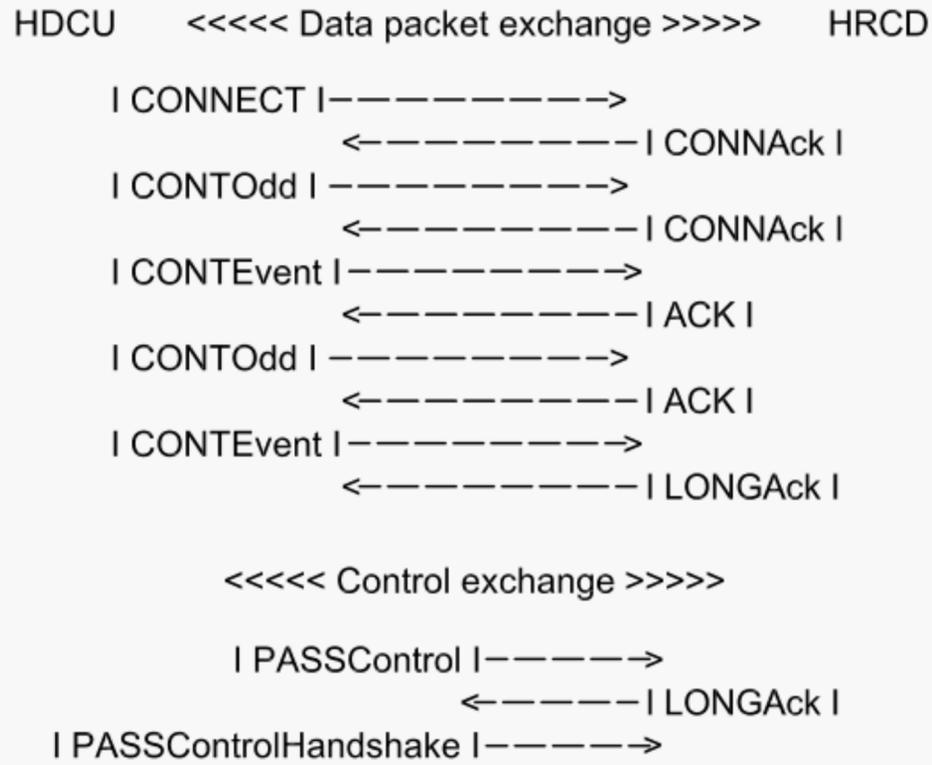
The RCD/device poll message session allows the HDCU to request basic information from a selected HRCD. The HDCU uses a data packet exchange to transmit this request to the HRCD. The HDCU then retrieves the HRCD's response to this request using a control exchange. An example showing the flow of information is presented in Figure 17.

5.2.4.3.3 RCD/device xmit1/2

The RCD/device xmit1/2 message session transfers data from the HDCU to a selected HRCD using a data packet exchange. The data can be a message requiring an HRCD response, in which case the HRCD buffers it in the HRCD response buffer. An interactive poll (see 5.2.4.3.6) is then used to retrieve this buffered response. An example showing the flow of information is presented in Figure 18.

5.2.4.3.4 RCD/device interactive1/2

The RCD/device interactive1/2 message session initiates an xmit1/2 message transfer session (see 5.2.4.3.3) to a selected HRCD. The HDCU then initiates a control exchange (see 5.2.4.3.6) to the same HRCD after a time delay as specified in the response delay field of the interactive1/2 command (see 5.2.2.2.3.4) to retrieve the information requested and buffered from the xmit1/2 session. The HDCU is free to communicate with other HRCDs during the delay period between the xmit1/2 session and the control exchange to this HRCD. An example of the flow of information is presented in Figure 19.



The HRCD accepts control from the HDCU and transmits its response to the HDCU

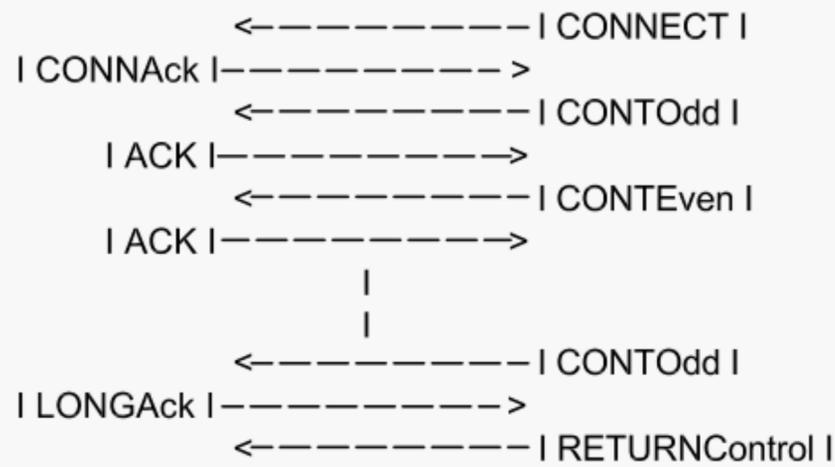


Figure 17 — RCD device poll session transfer diagram

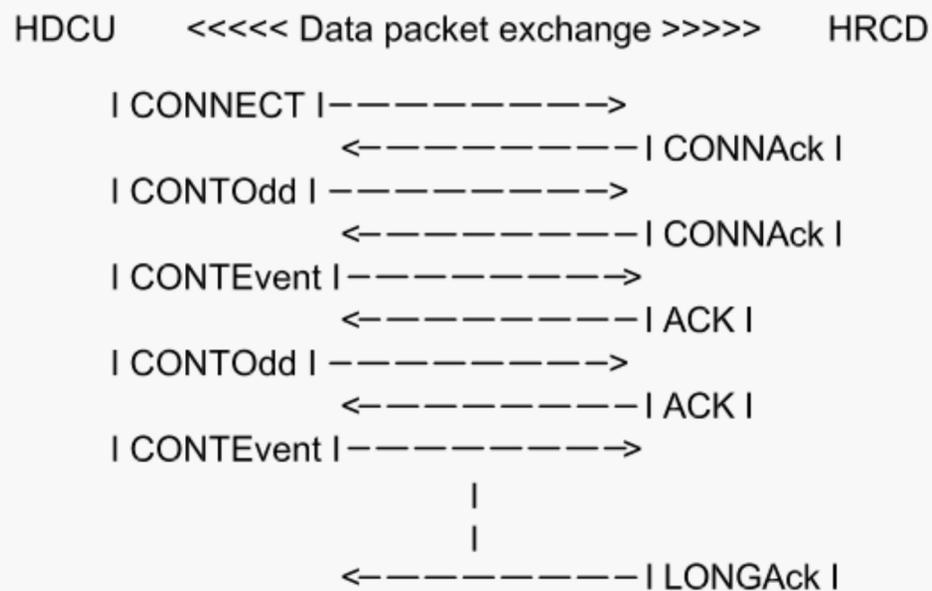


Figure 18 — RCD/device xmit1/2 session transfer diagram

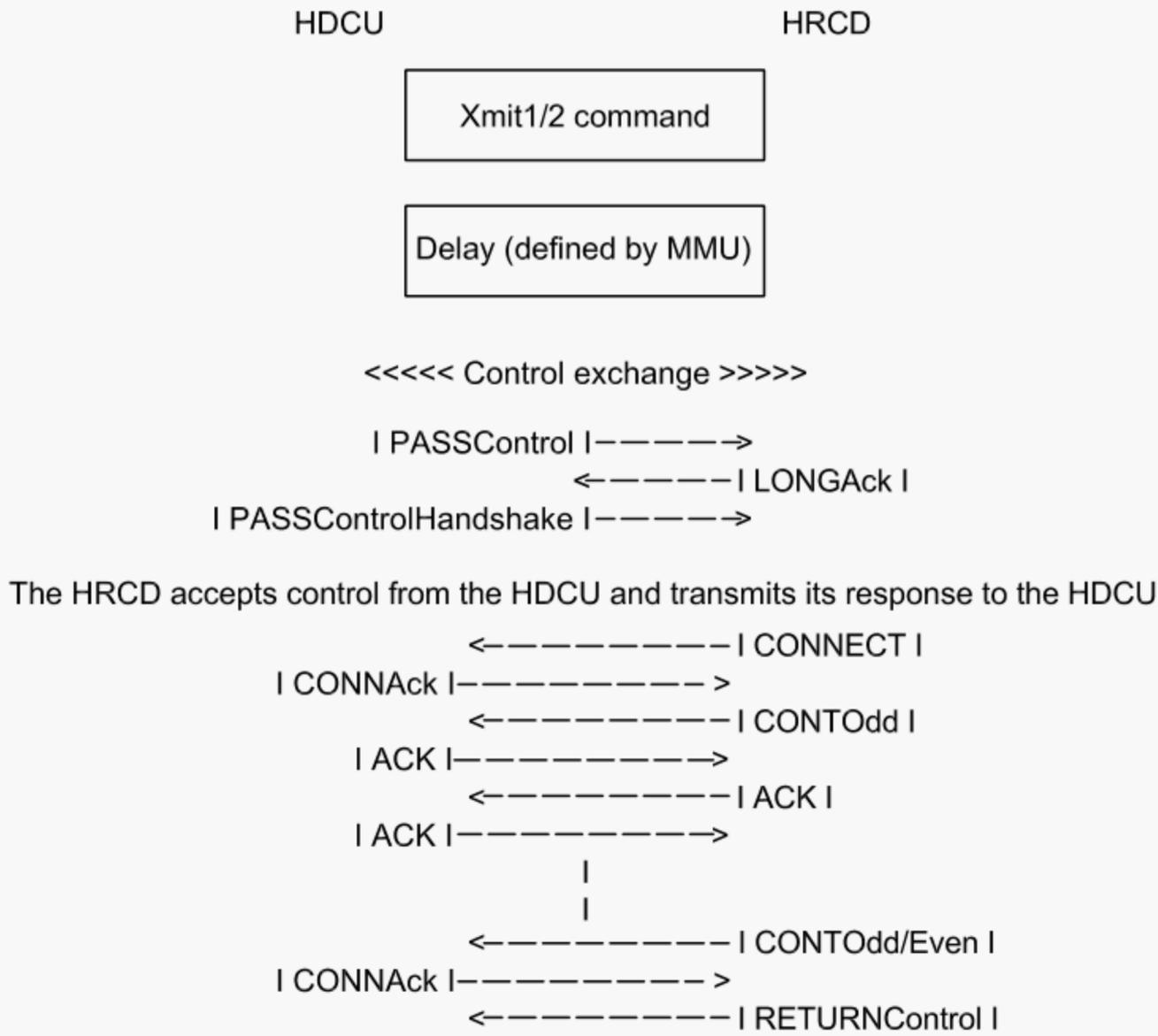


Figure 19 — RCD/device interactive1/2 session transfer diagram

5.2.4.3.5 RCD/device map

The RCD/device map message session broadcasts an MMU-originated map command from the HDCU, using a broadcast data packet exchange as defined in 5.5.9.4, requesting, with varying probabilities, an HRCD to respond. The purpose of this command is to log in, or map, any newly acquired or unpolled HRCDs on the power line network. Since it is a broadcast type of message, it cannot simply select a particular HRCD by using a unique individual network address. Instead, the 11-byte ASCII network address field described in 5.2.4.2.2.3 is broken into ASCII subfields supplying all HRCDs on the power line network with a mapping strategy (as defined in 5.2.2.2.3.5). An “unused” 2-byte network address is included for use by a (the) responding HRCD(s). A (the) responding HRCD(s) will use this “unused” 2-byte network address for 20 s. The HDCU then attempts to send a command, e.g. a device poll or an RCD poll, using a data packet exchange addressed to this 2-byte network address, with the container ID set to the universal address. This type of broadcast technique, however, suggests the possibility of more than one HRCD attempting to receive this command. If the HDCU does not successfully complete this data packet exchange, the HDCU replies to the MDCU with the appropriate reply type (described in 5.2.2.3). A successful data packet exchange indicates that a single HRCD has mapped onto the system and thus the HDCU grants temporary access to the power line network to this HRCD via a control exchange. The HRCD accepts temporary control and transmits its 11-byte ASCII network address to the HDCU via a data packet exchange and refreshes its 20 s timer. The login sequence is completed when the HRCD receives an individually addressed command with its actual container ID within 20 s. An example showing the flow of information for a successful map is shown in Figure 20.

The assignment of a new 2-byte RCD network address to a specific RCD with a known 11-byte address can be used to minimize mapping response collisions. This case can address RCDs which have dropped off-line owing to power loss or lack of interrogation within the previous hour.

5.2.4.3.6 RCD/device interactive poll

The RCD/device interactive poll message session is used by the HDCU to retrieve information from an HRCD after the HRCD has been sent an xmit1/2 message session (see 5.2.4.3.3) which required the HRCD to buffer response information. This process is executed in the same manner as the poll message session of 5.2.4.3.2. The HDCU uses a data packet exchange to transmit this request to the HRCD. The HDCU then retrieves the HRCD's buffered response using a control exchange. For an example of the flow of information, see Figure 17.

<<<<< Map broadcast data packet exchange >>>>>

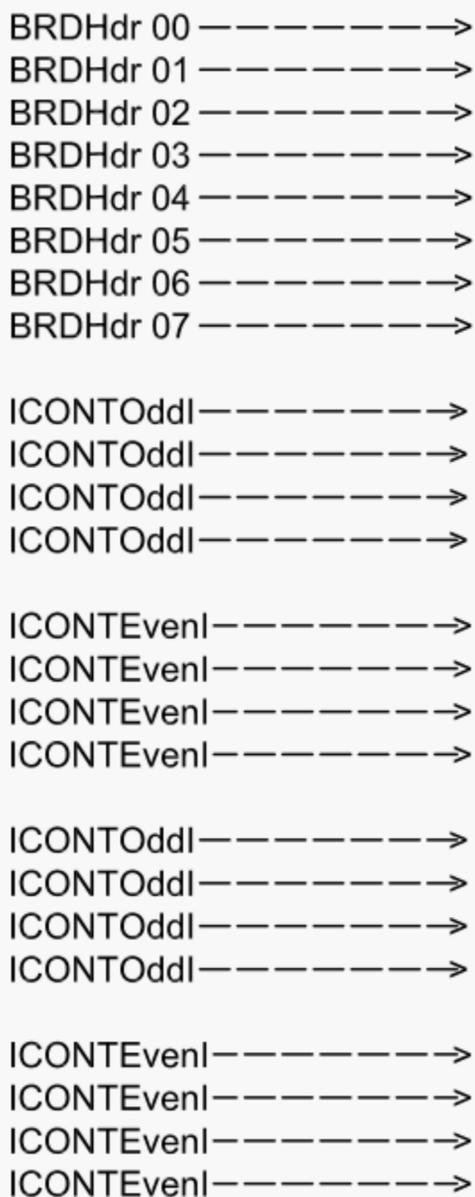


Figure 20 — RCD/device map session transfer diagram (continued)

The HDCU broadcasts the MMLI-originated map command. The 2-byte unused HRCD destination address or destination address individual included in the map command is used by an HRCD that responds to the map.

Source address = DCU address. Data consist of 1-byte for the message type, 1-byte for the routing byte, and the 11-byte ASCII RCD network address. The optional 2-byte destination address individual is included when the 11-byte ASCII RCD network address specifies an individual container ID. The exchange is sent as specified in 5.5.9.3

<<<<< Data packet exchange >>>>>

ICONNECTI  
 ICONNAckI  
 ICONTOddI  
 IACKI  
 ICONTEvenI  
 I LONGAckI

The HDCU sends a command, e.g. a device poll or an RCD poll, addressed to the 2-byte unused HRCD destination address/destination address individual address included in the previous map command. The 11-byte ASCII RCD network address used in this command is the universal container address. It is used by the HDCU to determine whether an HRCD responded to the previous map command and if so the HRCD returns its container ID.

Destination address = Unused I-IRCD destination address/destination address individual from previous map broadcast data packet.

Source address = HDCU address.

Data is the command using the 11-byte ASCII universal address of 30H 30H 30H 30H followed by 7-digits ASCII digits (30H 39H).

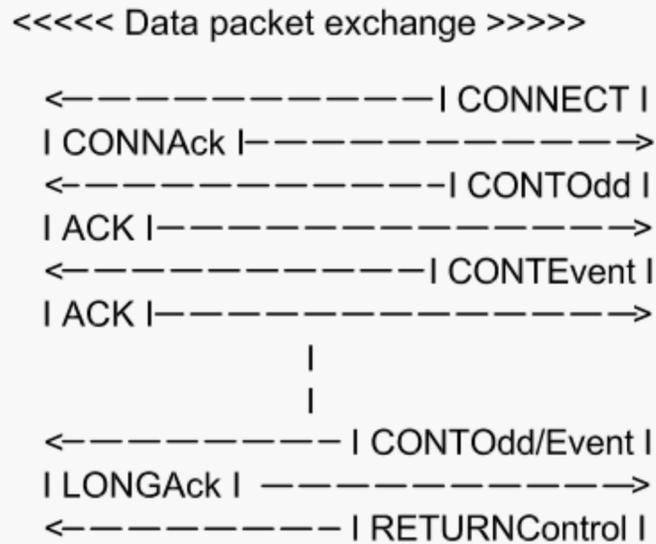
<<<<< Control exchange >>>>>

I PASSControl -----  
 <----- I LONGAck I  
 I PASSControlHandshake I----->

HDCU uses a control exchange addressed to the 2-byte unused HRCD destination address/destination address individual address included in the previous map command to retrieve the HRCD's container ID.

Destination address = Unused HRCD destination address/destination address individual from previous map broadcast data packet.

**Figure 20 — RCD/device map session transfer diagram (continued)**

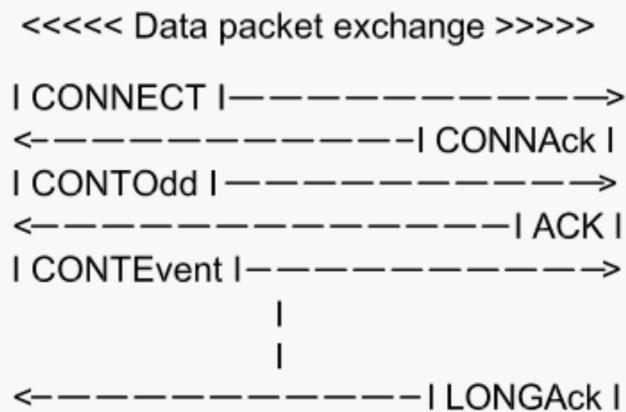


The HRCD accepts control from the HDCU and transmits its 11-byte ASCII RCD network address to the HDCU.

Destination address = HDCU address.

Source address = Unused HRCD destination address previous map broadcast data packet.

Data is the response to the command used in the previous data packet exchange which includes the HRCD's container ID.



The MMU/HDCU must send an individually addressed command to the HRCD with its container ID to complete the login procedure.

Destination address = Newly mapped HRCD destination address individual from previous map command.

Source address = HDCU address.

Data is the command using the 11-byte ASCII RCD network address retrieved from the previous map sequence.

<<<<< End of map session >>>>>

**Figure 20 — RCD/device map session transfer diagram**

## 5.3 MMU/Device communications

### 5.3.1 Headers

#### 5.3.1.1 General

Headers are entered using the data log write message. The data field consists of a 1-byte control word and 3-byte preamble followed by text fields. Each field is a string of variable length using standard ASCII characters separated with AA in hex and padded with AA in hex to a combined length of 128 bytes. The recording device shall interpret the control word and place the header, as transmitted, into its storage area.

#### 5.3.1.2 Header fields

A header contains the following fields in the order given:

- 1) Container
- 2) Vessel voyage
- 3) Origin
- 4) Shipper
- 5) Temperature set-point
- 6) Air exchange setting
- 7) Humidity set-point
- 8) Operator
- 9) Date
- 10) Time
- 11) Product
- 12) Intermediate destination
- 13) Final destination
- 14) Booking
- 15) Comments.

#### 5.3.1.3 Header field sizes

Each field in the header shall be less than 13 characters. The total number of characters in a header, not including the control word and preamble, shall equal 124. The last byte shall be hex AA.

### 5.3.2 Other MMU/Device messages

The pass-through or manufacturer specific method shall be described.

## 5.4 Low data rate physical requirements — LDCU to LRCD

### 5.4.1 Frequency

The power line signals will be sent at 55 kHz with a variation of  $\pm 2\%$  (i.e. in the range 53,9 kHz to 56,1 kHz).

### 5.4.2 Modulation method

The messages shall be frequency shift keyed (FSK). Each character transferred requires one start bit (low logic level), eight data bits and one stop bit (high logic level). A packet transfer includes a start preamble of three transitions followed by 10 logic one bits. This preamble restricts the deadtime between any two bytes in a packet to be less than one-half bit length.

### 5.4.3 Baud rate

The communications shall be sent at 1 200 baud or 1 200 bits/s.

### 5.4.4 Transmission mode

The signals shall be sent half duplex. The signals shall be transmitted at 6 V rms maximum into a line impedance of  $(15 + j15) \Omega$ .

### 5.4.5 Injection mode

The injection mode of the signal shall be "voltage injection", the difference being signal levels between the three phases and the ground.

### 5.4.6 Receiver sensitivity

The receiver sensitivity shall be 1 mV rms at 55 kHz  $\pm$  10 kHz.

### 5.4.7 Non-transmission Impedance

The signal shall be transmitted over lines with impedance levels of less than 3 k  $\Omega$  at 55 kHz.

### 5.4.8 Bit synchronization

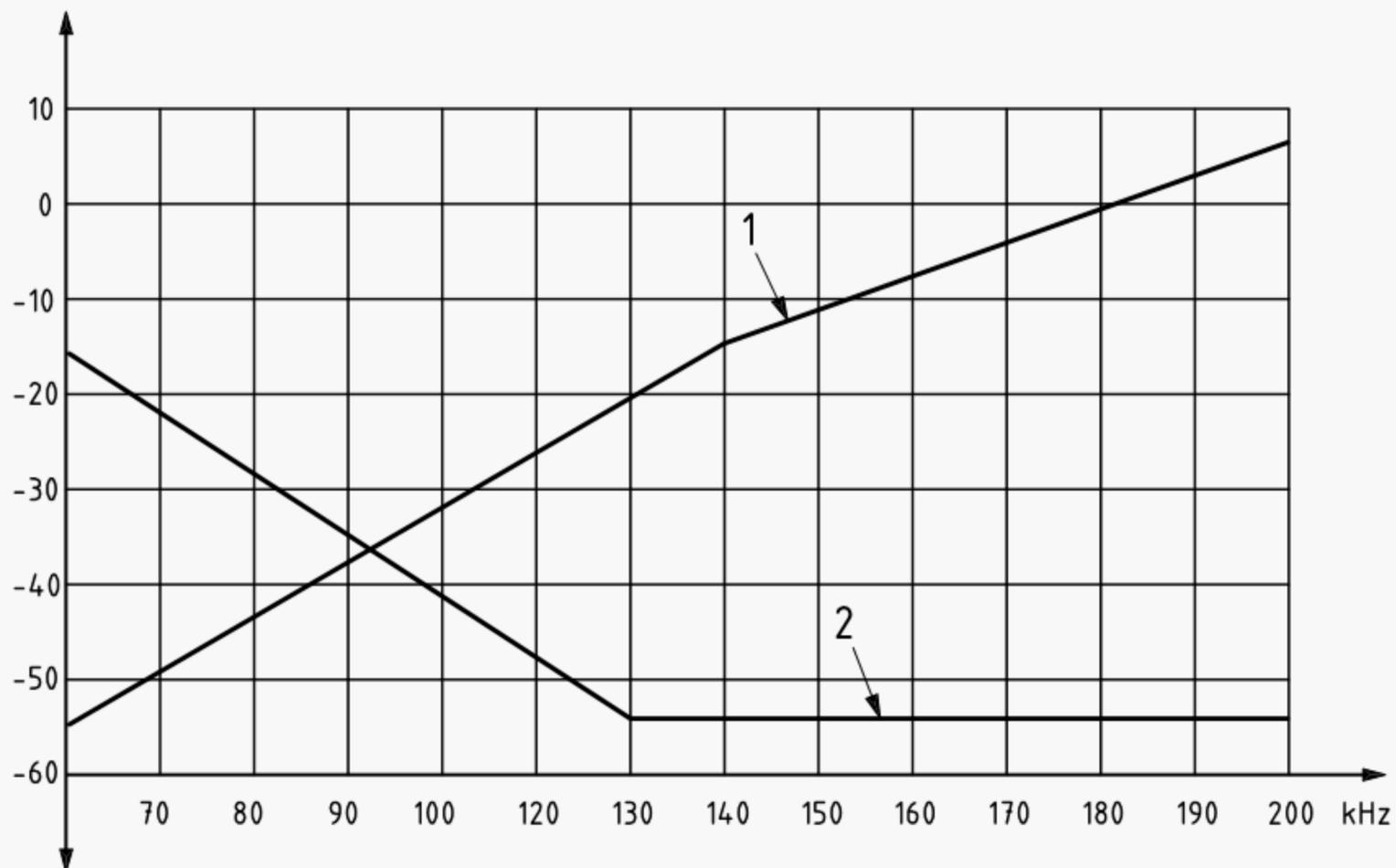
The messages sent shall be asynchronous.

### 5.4.9 Carrier setup time

The signal setup time shall be 10 ms.

### 5.4.10 Out-of-band filtering for HDR compatibility

The out-of-band power spectrum shall not exceed 2 mV rms from 130 kHz to 400 kHz and shall not exceed the linear plot shown in Figure 21 of decibel volts against frequency from 180 mV rms at 70 kHz to 2 mV rms at 130 kHz as measured into an 18  $\sim$  resistive load in any 10 kHz bandwidth.

**Key**

- 1 HDR
- 2 LDR

**Figure 21 — Filter specifications — High data rate compatibility**

## 5.5 High data rate physical requirements — HDCU to HRCD

### 5.5.1 Modulation method — Broad band

#### 5.5.1.1 General

The transmitter shall generate a broad band signal to conform to the signal defined by 5.5.1.2 and 5.5.1.3 in the signal bandwidth of 140 kHz to 400 kHz.

#### 5.5.1.2 Spreading method

An anti-symmetric waveform is digitally synthesized using 16 bits (chips) of a 4,300 8 MHz clock. Two periods of the anti-symmetric waveform (corresponding to 32 chips of the clock) define one raw data bit interval. The levels of the 16 chips may be chosen in any manner such that the resulting waveform is anti-symmetric and the power spectrum of one raw data bit matches that specified in 5.5.5.

#### 5.5.1.3 Raw data bit modulation

Each raw data bit interval is binary phase shift keyed (BPSK) using a non-return-to-zero level (NRZ-L) format, resulting in a raw data bit rate of 134,4 Kbits s<sup>-1</sup>.

### 5.5.2 Transmission mode

There is a maximum output power of 100 mW per 10 kHz bandwidth over a bandwidth of 140 kHz to 400 kHz for a line impedance of 18 Ω.

### 5.5.3 Injection mode

There is line-impedance-controlled voltage injection between three phases and ground.

### 5.5.4 Output/input impedance

The output impedance shall not exceed  $18\ \Omega$  phase-to-phase and  $21\ \Omega$  phase-to-ground. The input impedance (non-transmit) shall be greater than  $200\ \Omega$ .

### 5.5.5 Power density function

The null-to-null bandwidth of one raw data bit shall be from a maximum of  $140\ \text{kHz}$  to a minimum of  $400\ \text{kHz}$ . The null-to-null bandwidth of transmitted data shall be from  $140\ \text{kHz}$  to  $400\ \text{kHz}$ . The power density function of transmitted data shall be as shown in Figure 22.

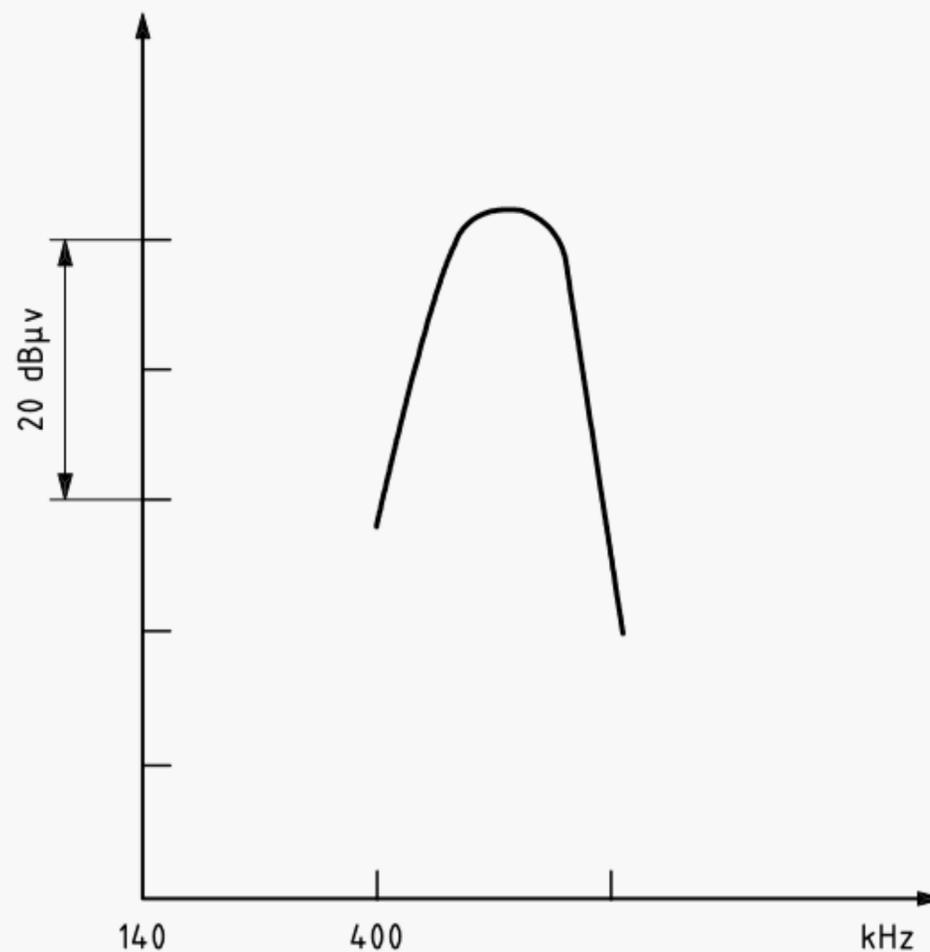


Figure 22 — Power density function

### 5.5.6 Synchronization method

The receiver shall synchronize to the synchronization preamble defined in 5.5.9.2.1. The preamble synchronization time shall not exceed  $0,3\ \text{ms}$  from the start of the preamble.

### 5.5.7 Demodulation method

The receiver shall demodulate raw data as modulated in 5.5.1. The receiver may use any method of equalization that can correct for an attenuation of at least  $15\ \text{dB}$  over  $60\ \text{kHz}$ .

### 5.5.8 Receiver sensitivity

The receiver shall operate with less than a  $2\ \text{mV}$  received signal amplitude.

## 5.5.9 Data link protocol

### 5.5.9.1 General

Codes for data link fields are given in Table 17. (32, 8) ECC and EDC codes are given in Table 18.

### 5.5.9.2 Frame formats

#### 5.5.9.2.1 Preamble

The preamble delimits the start of a valid message. It contains the necessary sequences to synchronize communications between devices on the power line:

|SYNC|SYNC|SYNC|SYNC|SOF| — total of 5 bytes

where

- SYNC is the synchronization sequence; and
- SOF is the start of frame field which may be SOFLong or SOFShort.

Each field is 1 byte in length.

Preamble(L) denotes a Preamble with a SOFLong field.

Preamble(S) denotes a Preamble with a SOFShort field.

#### 5.5.9.2.2 Data frames

##### 5.5.9.2.2.1 Unicast data frames

CONNECT and FIRSTFrameReTx signal the beginning of a packet. The first transmission of a packet uses a CONNECT. Subsequent retries of the same packet use FIRSTFrameReTx.

|Preamble(L) |HDRLong|DSTHigh|DSTLow|SRCHigh|SRCLow| — total of 25 bytes

where

- HDRLong is the long (encoded) header field (= CONNECT or FIRSTFrameReTx);
- DSTHigh is the destination address field high-order byte;
- DSTLow is the destination address field low-order byte;
- SRCHigh is the source address field high-order byte;
- SROLow is the source address field low-order byte.

Each field is encoded by (32, 8) ECC and EDC.

CONTOdd is the frame succeeding a CONNECT or FIRSTFrameReTx. It contains the remaining packet header fields and the first three bytes of the data field.

IPreamble(S)|HDRShort|NDBHigh|NDBLow|DATA1|DATA2|DATA3| — total of 26 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (CONTOdd);
- NDBHigh is the number of data bytes field high-order byte;
- NDBLow is the number of data bytes field low-order byte;
- DATA1,... DATA3 are data fields.

Each field (non-header) is encoded by (32, 8) ECC and EDC.

#### 5.5.9.2.2.2 Broadcast data frames

BRDHdr signals the beginning of a new broadcast packet. The BRDHdr frame is repeated eight times to allow all receiving units an ample opportunity to initiate a broadcast packet reception since receiver requests for retransmission are not possible. The sequence field is initialized to 00 and is incremented for each subsequent transmission.

|Preamble(L)|HDRLong|Sequence| — total of 13 bytes,

where

- HDRLong is the long (encoded) header field (= BRDHdr);
- sequence is the repetition count of consecutive BRDHdr transmissions and ranges in value from 00 to 07.

Each field is encoded by (32, 8) ECC and EDC.

CONTOdd is the frame succeeding the BRDHdr. It contains the remaining packet header fields and the first byte of the data field:

|Preamble(S)|HDRShort|SRCHigh|SRCLow|NDBHigh|NDBLow|DATA1| — total of 26 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (CONTOdd);
- SRCHigh is the source address field high-order byte;
- SROLow is the source address field low-order byte;
- NDBHigh is the number of data bytes field high-order byte;
- NDBLow is the number of data bytes field low-order byte;
- DATA1 is a data field.

Each field (non-header) is encoded by (32, 8) ECC and EDC.

#### 5.5.9.2.2.3 Unicast/broadcast data frames

Unicast/broadcast data frames alternate between CONTEven and CONTOdd for consecutive frames. A frame can contain 5 bytes of the data field for intermediate frames, and 0 to 4 bytes of the data field for the last frame of the packet. The last byte of the last frame shall contain the XOR check sum field.

|Preamble(S)|HDRShort|DATA1|... |DATA5| — total of 26 bytes (intermediate frames)

|Preamble(S)|HDRShort|DATA1|... |DATA4|XOR| — total of 10 bytes to 26 bytes (last frame)

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (= CONTEven or CONTOdd);
- DATA1...DATA5 are data fields;
- XOR is the XOR check sum field exclusive-or of DST, SRC, NDB, and all DATA fields.

Each field (non-header) is encoded by (32, 8) ECC and EDC.

#### 5.5.9.2.3 Control transfer frames

PASSControl represents the attempt of the master to poll a slave unit:

|Preamble(L)|HDRLong|DSTHigh|DSTLow| — total of 17 bytes

where

- HDRLong is the long (encoded) header field (= PASSControl);
- DSTHigh is the destination address field high-order byte;
- DSTLow is the destination address field low-order byte.

Each field is encoded by (32, 8) ECC and EDC.

#### 5.5.9.2.4 Response frames

##### 5.5.9.2.4.1 General responses

LONGAck is the response to one of the following:

- the last frame of a data packet, signalling the end of the current packet;
- a FIRSTFrameReTx data frame, indicating the detection of the packet as a duplicate by the receiver;
- a PASSControl message from the master, signalling the need of the slave to acquire control to send data to the master.

|Preamble(S)|HDRShort|HDRLong|SRCHigh|SRCLow| — total of 18 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length ( NOTAck);
- HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC ( LONGAck);
- SRCHigh is the source address field high-order byte;
- SRCLow is the source address field low-order byte.

Each SRC field is encoded by (32, 8) ECC and EDC.

NAK is the response of the receiving unit indicating that it has detected errors in a received data frame or the response of a slave to a PASSControl message from the master when the slave does not have data to send.

|Preamble(S)|HDRShort|HDRLong| — total of 10 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (= NOTAck);
- HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC (=NAK).

#### 5.5.9.2.4.2 Data frame only responses

CONNAck is the response to a CONNECT indicating that it has successfully received the first frame of the packet.

|Preamble(S)|HDRShort|HDRLong| — total of 10 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (= NOTAck);
- HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC ( CONNAck).

ACK is the response of the receiving unit indicating that it has successfully received a data frame.

|Preamble(S)|HDRShort| — total of 6 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (= ACK);
- BUSYNak is the response to a CONNECT or the CONTOdd following a CONNECT or FIRSTFrameReTx indicating a temporary lack of buffer resources at the receiving unit.

|Preamble(S)|HDRShort|HDRLong| — total of 10 bytes

where

- HDRShort is the short (non-encoded) header field of 1 byte in length (= NOTAck);
- HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC (=BUSYNak).

#### 5.5.9.2.4.3 Control transfer responses

PASSControlHandshake succeeds the acknowledgement of a PASSControl message by the receiving unit (a slave). The PASSControlHandshake signals the slave to start transmitting its data.

|Preamble(L)|HDRLong| — total of 9 bytes

where

- HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC (= PASSControlHandshake).

RETURNControl follows the transmission of a data packet by a slave which has been temporarily transferred control by the master. RETURNControl signals the master that it can resume control of the network.

|Preamble(L)|HDRLong| — total of 9 bytes

where

- HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC (= RETURNControl).

**5.5.9.2.5 Control frames**

ACTIVITYFrame is transmitted by the master to let all other attached units (slaves) know that there is an operating master on the power line network. The master transmits this control frame while in the idle state, i.e. while not engaged in a data packet exchange or control exchange. The master will wait 1 ms between successive ACTIVITYFrames.

|Preamble(L)|HDRLong| — total of 9 bytes

where

— HDRLong is the long (encoded) header field encoded by (32, 8) ECC and EDC (= ACTIVITYFrame).

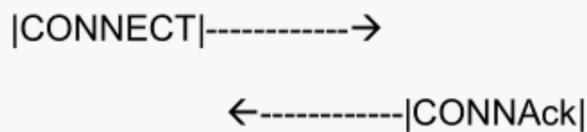
**5.5.9.3 Unicast data packet exchange**

**5.5.9.3.1 First frame exchange**

**5.5.9.3.1.1 General**

First frame exchange describes the initial exchange of frames for a new packet. When the receiving unit acknowledges the first frame, the exchange continues; otherwise, an attempt is made at retransmitting the first frame.

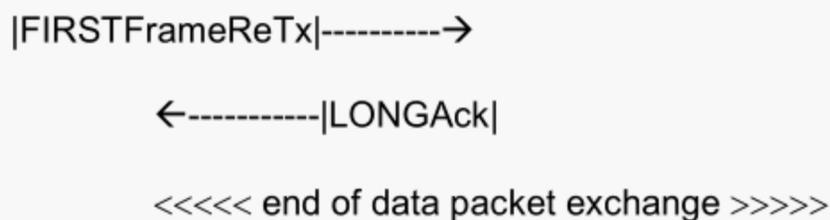
**5.5.9.3.1.2 Without retransmission**



**5.5.9.3.1.3 With retransmission**

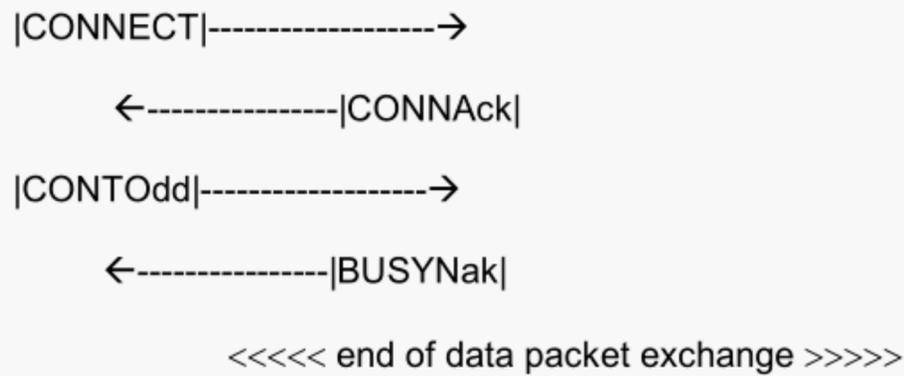


**5.5.9.3.1.4 With duplicate detection**

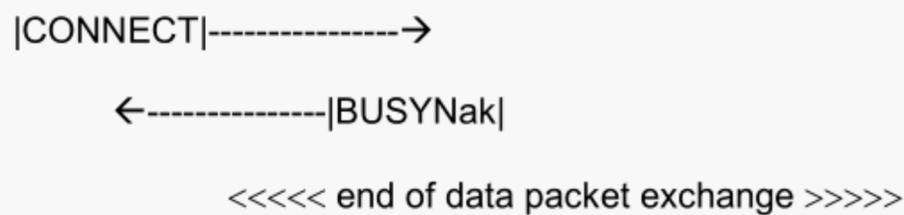


The transmitter only uses a FIRSTFrameReTx if, during a previous attempt at the packet transmission it transmitted the last frame. If the last frame was not transmitted, the receiver could not have successfully received the entire data packet, and a CONNECT should be used for the packet retransmission.

### 5.5.9.3.1.5 With lack of receiver buffer resources



After receiving the NDB fields in the CONTOdd following the CONNECT, the receiver determines that it has insufficient buffer resources to accommodate the data packet. It indicates this to the transmitter by responding with a BUSYNak response frame.



After receiving the CONNECT, the receiver determines that it has insufficient buffer resources to accommodate even a single byte data packet. It indicates this to the transmitter by responding with a BUSYNak response frame.

### 5.5.9.3.2 Intermediate frame exchange

#### 5.5.9.3.2.1 Without retransmission

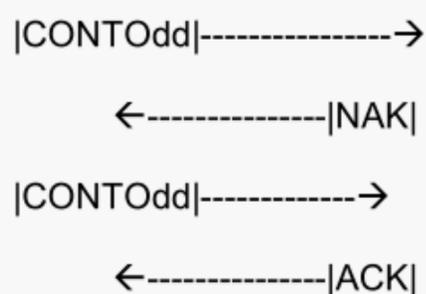


Final frame:



This is the most common form of data exchange between units on the power line. The frame headers alternate between CONTOdd and CONTEven for consecutive frames. Note that the response to the last frame is a LONGAck, signalling to the transmitting unit that the entire packet was received successfully.

#### 5.5.9.3.2.2 With retransmission



The receiving unit uses a NAK control frame to request a retransmission of a data frame received with errors.

**5.5.9.4 Broadcast data packet exchange**

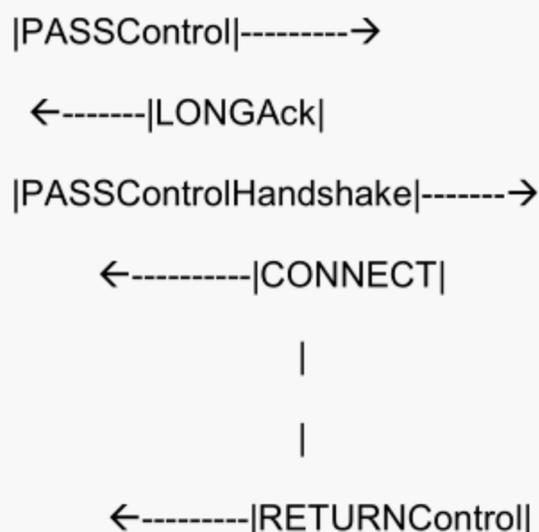
BRDHdr	00
BRDHdr	01
BRDHdr	02
BRDHdr	03
BRDHdr	04
BRDHdr	05
BRDHdr	06
BRDHdr	07
CONTOdd	
CONTOdd	
CONTOdd	
CONTOdd	
CONTEven	
CONTEven	
CONTEven	
CONTEven	

This describes the transmission of a broadcast data packet where the receiving units cannot confirm the receipt of a transmission or request a retransmission. The BRDHdr frame is repeated eight times to allow all receiving units an ample opportunity to initiate a broadcast packet reception. All subsequent data frames, both intermediate and last, are repeated four times, again to allow all receiving units to receive at least one error-free frame from each repeated data frame.

There shall be a minimum of 1-byte time (59,5 µs) and a maximum of 2-byte times (119 µs) between consecutive transmitted frames.

**5.5.9.5 Control exchange — Master polling a slave**

**5.5.9.5.1 Slave has data to transmit**



The master polls the slave by sending a PASSControl. If the slave has any data to send, it responds with a LONGAck and the master, in turn, sends a PASSControlHandshake. This allows the slave to start transmitting its data packet. On completion of its transmissions, the slave returns control to the master with a RETURN control.

**5.5.9.5.2 Slave does not have data to transmit**



The master polls the slave by sending a PASSControl. If the slave does not have any data to send, it responds with a NAK.

**5.5.9.6 Timeouts**

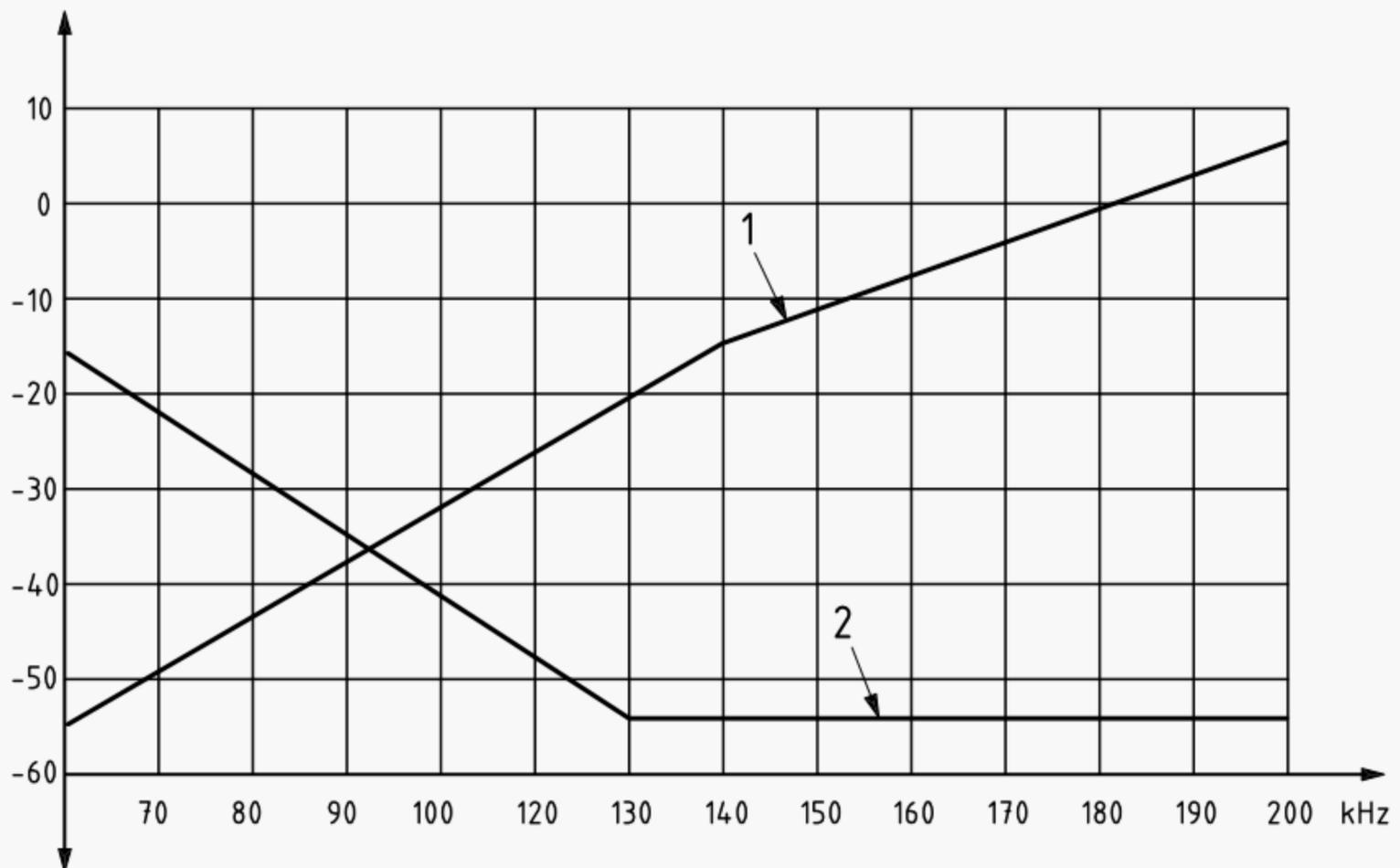
The transmitter will wait for at most 1 ms to receive a response from the destination. If none is received within this time period, it will retransmit the frame (data or control).

**5.5.9.7 Data rate**

An error-corrected data rate of 19,2 Kbits/s with an error rate of less than  $10^{-9}$  is achieved, and is determined by the specifications above.

**5.5.10 Out-of-band filtering requirements for “LDR” compatibility**

The out-of-band power spectrum shall not exceed 2 mV rms from 40 kHz to 70 kHz and shall not exceed the linear plot shown in Figure 23 of decibel volts against frequency from 2 mV rms at 70 kHz to 180 mV rms at 140 kHz and the linear plot shown in Figure 23 of decibel volts against frequency from 180 mV rms at 140 kHz to 2 V rms at 200 kHz as measured into an 18 load in any 10 kHz bandwidth.



- Key**  
1 HDR  
2 LDR

**Figure 23 — Filter specifications — Low data rate compatibility**

Table 17 — Codes for data link protocol

Frame and field(s)	Hexadecimal value of field
Preamble (see 5.5.9.2.1) SYNC SOFLong SOFSHORT	01 B6 DA
Data frames (see 5.5.9.2.2) CONNECT FIRSTFrameReTx BRDHdr CONTOdd CONTEven	50 A5 FB C6 39
Control transfer frames (see 5.5.9.2.3) PASSControl	16
Response frames (see 5.5.9.2.4) NOTAck LONGAck NAK CONNAck ACK BUSYNak PASSControlHandshake RETURNControl	4B 6D 58 EC B4 10 05 24
Control frames (see 5.5.9.2.5) ACTIVITYFrame	54

Table 18 — (32, 8) ECC and EDC codes

Databyte-code		Databyte-code		Databyte-code		Databyte-code	
00	5A5A5A5A	40	5ABC892E	80	5A31A5CE	C0	5AA27676
01	5D5AA589	41	5DBC76BB	81	5D315AB0	C1	5DA289A5
02	A55A5D31	42	A5B03144	82	A5316943	C2	A5A29669
03	695A695D	43	69B096D1	83	69315DA2	C3	69A23196
04	BC5AD196	44	BCBC44A2	84	BC3143D1	C4	BCA2A25D
05	435A4369	45	43B0A243	85	4331D144	C5	43A24431
06	2E5A2EA5	46	2EBCBBBC	86	2E31BCBB	06	2EA20E89
07	D15AB076	47	D1BCCECE	87	D1312E2E	07	D1A2BB5A
08	315A7644	48	31B0A531	88	31318969	08	31A25A43
09	895A89D1	49	89BC5A5D	89	89317696	09	89A2A5A2
0A	965A962E	4A	96B0695A	8A	96313176	CA	96A25DCE
0B	765A31BB	4B	76B05D89	8B	763196A5	CB	76A269B0
0C	A25AA2BC	4C	A2BC43A5	80	A2314489	CC	A2A2D1BB
0D	0E5A440E	4D	CEBCD176	8D	CE31A25A	CD	0EA2432E
0E	445A0EA2	4E	44B0BC96	8E	4431BB5D	CE	44A22ED1
0F	BB5ABB43	4F	BBBC2E69	8F	BB31CE31	CF	BBA2BC44

Table 18 (continued)

Databyte-code		Databyte-code		Databyte-code		Databyte-code	
10	5A5D69D1	50	5A43965D	90	5A895D96	D0	5A0E31A2
11	5D5D5D44	51	5D433131	91	5D896969	D1	5D0E9643
12	A55DA5BB	52	A5437689	92	A5895AA5	D2	A5CE89BC
13	695D5A2E	53	6943895A	93	6989A576	D3	690E76CE
14	BC5DBCCE	54	B0430E76	94	BC892E5A	D4	BCCEBB2E
15	435D2EB0	55	4343BBA5	95	4389BC89	D5	43CECEBB
16	2E5D4343	56	2E43A269	96	2E89D131	D6	2E0E4444
17	D15DD1A2	57	D1434496	97	D189435D	D7	D1CEA2D1
18	315D3189	58	31435DBB	98	318996BC	D8	310E69A5
19	895D965A	59	8943692E	99	8989310E	D9	890E5D76
1A	965D895D	5A	96435AD1	9A	968976A2	DA	960EA596
1B	765D7631	5B	7643A544	9B	76898943	DB	760E5A69
10	A25DBB69	5C	A2432E43	90	A2890E44	DC	A2CEBC31
1D	CE5DCE96	5D	0E43B0A2	9D	CE89BBD1	DD	CECE2E5D
1E	445D4476	5E	4443D1CE	9E	4489A22E	DE	440E435A
1F	BB5DA2A5	5F	BB4343BC	9F	BB8944BB	DF	BBCED189
20	5AA52E44	60	5A2EBB31	A0	5A96B069	E0	5A44CE43
21	5DA5BCD1	61	5D2ECE5D	A1	5D962E96	E1	5D44BBA2
22	A5A5D12E	62	A52E445A	A2	A5964376	E2	A544A2CE
23	69A543BB	63	692EA289	A3	6996D1A5	E3	694444BC
24	BCA55DBC	64	BC2E31A5	A4	B0966989	E4	B04496BB
25	43A569CE	65	432E9676	A5	43965D5A	E5	4344312E
26	2EA55AA2	66	2E2E8996	A6	2E96A55D	E6	2E4476D1
27	D1A5A543	67	D12E7669	A7	D1965A31	E7	D1448944
28	31A5CE5A	68	312EB02E	A8	3196BBCE	E8	31442E76
29	89A5BB89	69	892E2EBB	A9	8996CEBC	E9	8944B0A5
2A	96A5A231	6A	962E4344	AA	96964443	EA	9644D169
2B	76A5445D	6B	762ED1D1	AB	7696A2A2	EB	76444396
2C	A2A59696	60	A22E69A2	AC	A29631D1	EC	A2445D5D
2D	CEA53169	6D	CE2E5D43	AD	CE969644	ED	CE446931
2E	44A576A5	6E	442EA5B0	AE	449689BB	EE	44445A89
2F	BBA58976	6F	BB2E5ACE	AF	BB96762E	EF	BB44A55A
30	5A694389	70	5AD1A2BB	B0	5A76D1BC	F0	5ABB44A5
31	5D69D15A	71	5DD1442E	B1	5D7643CE	F1	5DBBA276
32	A569BC5D	72	A5D1CED1	B2	A5762EA2	F2	A5BBBB96
33	69692E31	73	69D1BB44	B3	6976BC43	F3	69BB0E69
34	B069A569	74	BCD17643	B4	BC765A44	F4	B0BB8931
35	43695A96	75	43D189A2	B5	4376A5D1	F5	43BB765D
36	2E696976	76	2ED1960E	B6	2E765D2E	F6	2EBB315A
37	D1695DA5	77	D1D131BC	B7	D17669BB	F7	D1BB9689
38	316944D1	78	31D1D15D	B8	3176A296	F8	31BB43A2
39	8969A244	79	89D14331	B9	89764469	F9	89BBD143
3A	9669BBBB	7A	96D12E89	BA	9676CEA5	FA	96BBBCBC
3B	76690E2E	7B	76D1BC5A	BB	7676BB76	FB	76BB2ECE
3C	A269890E	7C	A2D15A76	BC	A276765A	FC	A2BBA52E
3D	0E6976B0	7D	CED1A5A5	BD	CE768989	FD	CEBB5ABB
3E	44693143	7E	44D15D69	BE	44769631	FE	44BB6944
3F	BB6996A2	7F	BBD16996	BF	BB76315D	FF	BBBB5DD1

## Bibliography

- [1] ISO 1745:1975, *Information processing — Basic mode control procedures for data communication systems*
- [2] ISO 2111:1985, *Data communication — Basic mode control procedures — Code independent information transfer*
- [3] ISO 2628:1973, *Basic mode control procedures — Complements*
- [4] ISO 2629:1973, *Basic mode control procedures — Conversational information message transfer*
- [5] ISO 3309:1991, *Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures — Frame structure*
- [6] ISO 6346:1995, *Freight containers — Coding, identification and marking*
- [7] ISO 7498:1984, *Information processing systems — Open Systems Interconnection — Basic Reference Model*



