

INTERNATIONAL STANDARD

IEC
62394

First edition
2006-06

**Service diagnostic interface for consumer
electronics products and networks –
Implementation for ECHONET**



Reference number
IEC 62394:2006(E)

Publication numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series. For example, IEC 34-1 is now referred to as IEC 60034-1.

Consolidated editions

The IEC is now publishing consolidated versions of its publications. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

Further information on IEC publications

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology. Information relating to this publication, including its validity, is available in the IEC Catalogue of publications (see below) in addition to new editions, amendments and corrigenda. Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is also available from the following:

- **IEC Web Site** (www.iec.ch)

- **Catalogue of IEC publications**

The on-line catalogue on the IEC web site (www.iec.ch/searchpub) enables you to search by a variety of criteria including text searches, technical committees and date of publication. On-line information is also available on recently issued publications, withdrawn and replaced publications, as well as corrigenda.

- **IEC Just Published**

This summary of recently issued publications (www.iec.ch/online_news/justpub) is also available by email. Please contact the Customer Service Centre (see below) for further information.

- **Customer Service Centre**

If you have any questions regarding this publication or need further assistance, please contact the Customer Service Centre:

Email: custserv@iec.ch
Tel: +41 22 919 02 11
Fax: +41 22 919 03 00

INTERNATIONAL STANDARD

IEC 62394

First edition
2006-06

Service diagnostic interface for consumer electronics products and networks – Implementation for ECHONET

© IEC 2006 — Copyright - all rights reserved

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 the publisher.

International Electrotechnical Commission, 3, rue de Varembé, PO Box 131, CH-1211 Geneva 20, Switzerland
Telephone: +41 22 919 02 11 Telefax: +41 22 919 03 00 E-mail: inmail@iec.ch Web: www.iec.ch



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

PRICE CODE

XA

For price, see current catalogue

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms, definitions and abbreviations	8
3.1 Terms and definitions	8
3.2 Abbreviations	8
4 Different types of service diagnostics	8
4.1 Stand-alone products	8
4.2 Facilities or household appliances network.....	8
4.3 Remote diagnosis.....	9
5 SDI requirements	9
5.1 Hardware	9
5.2 Software.....	9
6 Tester software requirements	10
6.1 Reading the property diagnostic unit.....	10
7 Control protocol.....	11
7.1 Message structure (frame format).....	11
8 ECHONET objects: detailed specifications	40
8.1 Basic concept.....	40
8.2 ECHONET properties: basic specifications.....	41
8.3 Device object super class specifications.....	43
9 Supplementary information.....	49
9.1 Property map description format.....	49
9.2 All router data description format.....	50
9.3 Instance list description format.....	50
9.4 Class list description format	50
Figure 1 – ECHONET frame for plain data format	12
Figure 2 – EHD detailed specifications.....	13
Figure 3 – Configuration of SEA and DEA when an individual address is specified	14
Figure 4 – DEA (broadcast-stipulated) address configuration.....	14
Figure 5 – Broadcast target stipulation code	15
Figure 6 – Node group stipulation bit specifications	15
Figure 7 – OHD detailed specifications	16
Figure 8 – EOJ detailed specifications	16
Figure 9 – EPC detailed specifications.....	18
Figure 10 – ESV detailed specifications	18
Figure 10 – EpESV configuration	32
Figure 12 – Relationship between write request (requiring no response) and write "process-not-possible" response.....	35
Figure 13 – Relationship among write request (requiring a response), write "accepted" response, and write "process-not-possible" response	36

Figure 14 – Relationship among read request (requiring a response), read "accepted" response, and read "process not possible" response	38
Figure 15 – Relationship between notification request and notification response.....	38
Figure 16 – Relationship between property value notification (requiring a response) and property value notification response	39
Figure 17 – Processing target property counter for three requests	39
Figure 18 – Property data counter.....	40
Table 1 – List of class group codes.....	17
Table 2 – List of ESV codes for requests	20
Table 3 – List of ESV codes for response/notification.....	20
Table 4 – List of ESV codes for "response-not-possible" responses	21
Table 5 – List of CpESV codes for request/notification.....	34
Table 6 – List of CpESV codes for "accepted" response	34
Table 7 – List of CpESV codes for "process-not-possible" response	34
Table 8 – Data types, data sizes, and overflow/underflow codes.....	42
Table 9 – List of device object super class configuration properties.....	44
Table 10 – Relationship between the space name of the installation location and the assigned bit	46
Table 11 – Fault content property value assignments.....	47

INTERNATIONAL ELECTROTECHNICAL COMMISSION

SERVICE DIAGNOSTIC INTERFACE FOR CONSUMER ELECTRONICS PRODUCTS AND NETWORKS – IMPLEMENTATION FOR ECHONET

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62394 has been prepared by IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this standard is based on the following documents:

FDIS	Report on voting
100/1077/FDIS	100/1102/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

Withdrawn
IECNORM.COM: Click to view the full PDF of IEC 62394:2006

INTRODUCTION

Consumer products are often repaired by service workshops, which service a wide range of products developed by different manufacturers.

For highly complex products, fault diagnosis becomes increasingly difficult and time-consuming. To make diagnosis possible, manufacturers often develop built-in diagnostic software, which can be used for fault-finding together with an external diagnostic unit through a service diagnostic interface (SDI).

To avoid the need for a service workshop to purchase several different diagnostic units from different manufacturers for different products, a standardized SDI is proposed for use by all manufacturers and in all products in which such diagnostic interfaces are required. The result will be that only one SDI is needed in the service workshops.

The SDI should also be suitable for diagnosis in a network (facilities or household appliances network) in which different products from different manufacturers are connected together. The interface should also allow for future development.

The standard SDI which has to be specified, should

- be usable in future products;
- be easily connectable to a product or a network;
- be cheap;
- not limit product design.

SERVICE DIAGNOSTIC INTERFACE FOR CONSUMER ELECTRONICS PRODUCTS AND NETWORKS – IMPLEMENTATION FOR ECHONET

1 Scope

This International Standard specifies requirements for service diagnostic software to be implemented in products that incorporate a digital interface. It does not specify requirements for carrying out remote diagnosis or for manufacturer-dependent software.

The SDI requires the use of a controller (exclusive controller or general-purpose controller/PC) into which service diagnostic software can be loaded. Part of this controller software should be standardized while another part of this controller software is manufacturer-/product-related.

To reach a common approach in servicing all products from all manufacturers it is necessary to standardize specific items in the products (device under test (DUT)) as well as in the diagnostic software on the controller.

The SDI is based upon the ECHONET specification because this interface will be used in most future products. The use of this connection and existing communication protocols enable implementation in products at low cost and gives maximum flexibility and efficiency.

The SDI consists of

- the specific hardware and software requirements of the DUT;
- the specific requirements of the controller:
 - the service software;
 - an ECHONET interface (to be built in if not already present);
- the connection between the controller and the DUT.

This specification is the minimal specification necessary to be able to carry out computerized diagnosis and covers the standardized software of the controller as well as the standardized software and provisions in the DUT.

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.

ECHONET Specification:2002, Version 2.11

3 Terms, definitions and abbreviations

3.1 Terms and definitions

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

3.1.1

ECHONET specifications

ECHONET specifications were designed to enable the use of various kinds of transmission media (for example, power line, low-power radiofrequency, ETHERNET, Bluetooth®)

NOTE Ethernet is a registered trademark of the Xerox Corporation. Bluetooth® is a trademark owned by Bluetooth SIG, Inc.

3.1.12

remote diagnosis

diagnosis of a product via telephone, Internet, etc.

3.2 Abbreviations

EHD	ECHONET headers
SEA	Source ECHONET address
DEA	Destination ECHONET address
EBC	ECHONET byte counter
EDATA	ECHONET data
OHD	Object message header
EOJ	ECHONET objects
EPC	ECHONET property
ESV	ECHONET service
EDT	ECHONET property value data
CpESV	Compound ECHONET service
DUT	Device under test
OEM	Original equipment manufacturer
PC	Personal computer
ROM	Read-only memory
SDI	Service diagnostic interface

4 Different types of service diagnostics

4.1 Stand-alone products

For stand-alone products, a connection is made between the diagnostic controller and the DUT, where the DUT is from any manufacturer and of any type.

4.2 Facilities or household appliances network

In a facilities or household appliances network, a connection is made between the diagnostic controller and a network of facilities or household appliances. Several different facilities or household appliances are interconnected and not all of them are necessarily from the same manufacturer.

In this case, the SDI shall list the products on the network, detect which facilities or appliance is causing problem, and diagnose the product concerned.

4.3 Remote diagnosis

In addition to the configurations described in 4.1 and 4.2, a link can be made (for example, via telephone, the Internet, etc.) between the diagnostic controller in the workshop and a DUT/network at the customer's home. Therefore, if a product has both an ECHONET interface and a remote connection capability, this product should be able to transfer the diagnostic data, as described in this standard, through the remote connection.

5 SDI requirements

The SDI consists of

- hardware and software, both in the DUT and in the test equipment ("tester");
- the connection between the tester and the DUT.

The total SDI can be divided into the parts described in 5.1 and 5.2.

5.1 Hardware

5.1.1 Tester hardware

The hardware used for testing shall be a controller exclusive computer or general-purpose controller (for example, desktop or laptop PC) provided with at least one suitable network interface which enables the transfer of the ECHONET frame, as specified in 7.1, and running the necessary diagnostic software.

NOTE The minimum specification for the tester hardware depends on the respective tester platform.

5.1.2 Facilities or household appliances network

For the connection between the tester and the DUT, the "facilities or household appliances network" shall be used. For the diagnosis of the DUT using the network, the tester shall be connected to the facilities or household appliances network that conforms to the requirements of 7.1.

5.1.3 DUT hardware

5.1.3.1 General

The DUT shall be provided with at least one network interface which enables the transfer of the ECHONET frame as specified in 7.1.

5.1.3.2 Facilities or household appliances network

For diagnosis on a network, the tester shall, where possible, be connected to a "facilities or household appliances network" that conforms to the requirements of 7.1.

5.2 Software

NOTE The software for the SDI can be divided into two parts (tester and DUT) of which each part again can be divided into mandatory (SDI common) software and non-mandatory (manufacturer-dependent) software.

5.2.1 Tester software

The software platform of the tester shall be able to handle the ECHONET frame as specified in 7.1.

The SDI common software on the tester shall have the following functionalities:

- a) to initiate a service of “property value read request”, as specified in 7.1.9;
- b) to read out the service of “property value read response” and “property value notification” of all products, as specified in 7.1.9;
- c) to display a list of all products connected to the facilities or household appliances network to which the tester is connected. On the display shall be listed the
 - manufacturer code property;
 - place-of-business code property;
 - product code property;
 - serial number property;
 - date-of-manufacture property;
- d) to display an indication of the fault status property which describes the occurrence of an error in an actual device. The property code used as a property value is 0 × 41 when an error exists or 0 × 42 when no error exists and is found to be “OK” or “Not OK” as specified in 8.3.5;
- e) to display an indication of the fault content property which describes the content of an error in an actual device as specified in 8.3.6.

5.2.2 DUT software requirements for the SDI

The DUT shall be able to handle the ECHONET frame as specified in 7.1.

In addition, the SDI common software in the DUT shall be able to

- a) run a self-test routine;
- b) receive a service of “property value read request” as specified in 7.1.9 which is initiated by the tester and response a service of “property value read response” as specified in 7.1.9;
- c) initiate a service of “property value notification” as specified in 7.1.9.

6 Tester software requirements

6.1 Reading the property diagnostic unit

6.1.1 General

The common application shall be able to retrieve from the SDI-compliant devices and display the information specified in 6.1.1 to 6.1.3.

6.1.2 General information (product identification)

The manufacturer code property, the place-of-business code property, the product code property and the serial number property shall be read from the DUT and displayed. These property data shall always be available as specified in 8.3. The tester shall display this information for all devices in the system.

NOTE The manufacturer code displayed might not be the same as the name on the physical device.

6.1.3 Diagnosis information

After start-up of the general information software, the diagnosis information shall be displayed.

7 Control protocol

7.1 Message structure (frame format)

The ECHONET specifications were designed to enable the use of various kinds of transmission media (for example, power line, low-power radiofrequency, ETHERNET, Bluetooth®). Slow transmission speeds discourage large data transfers, and it is desirable to reduce the mounting load on simple devices. In the light of this situation, ECHONET specifies the frame format for the ECHONET communication middleware block to minimize the message size while fulfilling the requirements of the communications layer structure.

7.1.1 Frame format

Figure 1 shows the content of the ECHONET communication middleware frame format. Detailed specifications for each message component will be provided in the following subclauses.

7.1.1.1 Message configuration for exchange between ECHONET communications processing blocks

In the ECHONET communication middleware specifications, messages exchanged between ECHONET communications processing blocks are called ECHONET frames. ECHONET frames are roughly divided into two types depending on the specified EHD: the secure message format, of which the EDATA section is enciphered, and the plain message format, of which the EDATA section is not enciphered. The secure message format and the plain message format are subdivided into three formats depending on the specified EHD (see Table 2). Therefore, the following six different message formats are available for ECHONET frames.

a) Plain basic message format

Insecure communication is performed so that one message is used to view or change the contents of one property.

b) Plain compound message format

Insecure communication is performed so that one message is used to view or change the contents of two or more properties.

c) Plain arbitrary message format

Insecure communication is performed so as to exchange information that complies with vendor-unique specifications.

d) Secure basic message format

Secure communication is performed so that one message is used to view or change the contents of one property.

e) Secure compound message format

Secure communication is performed so that one message is used to view or change the contents of two or more properties.

f) Secure arbitrary message format

Secure communication is performed so as to exchange information that complies with vendor-unique specifications.

Figure 1 shows the ECHONET frame structure for the plain message format.

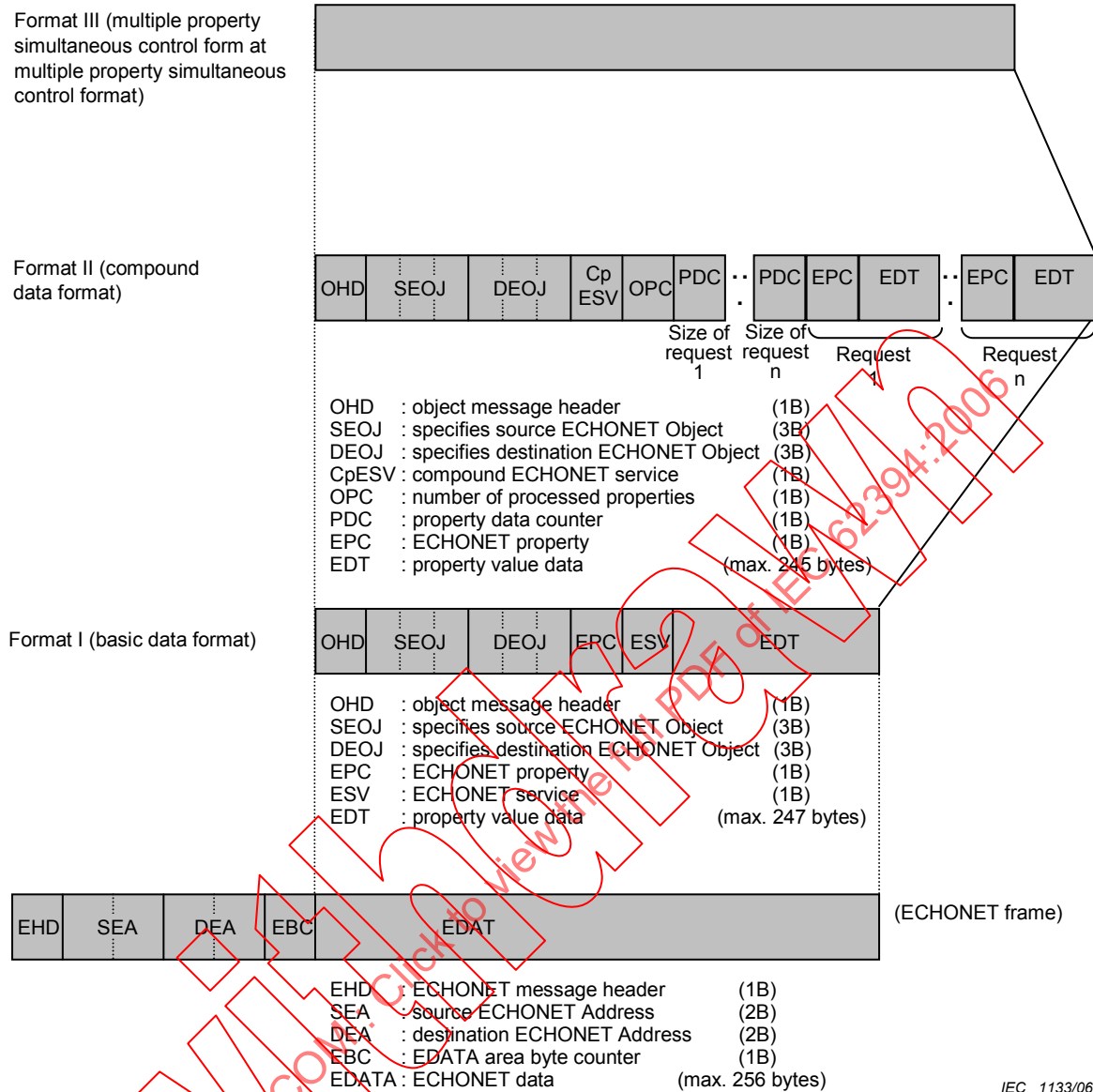
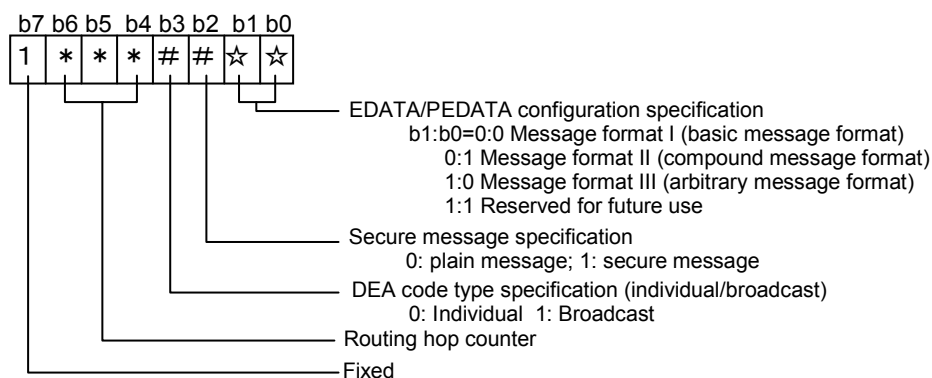


Figure 1 – ECHONET frame for plain data format

7.1.2 ECHONET headers (EHD)

This subclause provides detailed specifications for the ECHONET header (EHD) shown in Figures 1 and 2.



IEC 1134/06

NOTE When b7=0, b0 to b6 will be specified separately (reserved for future use).

Figure 2 – EHD detailed specifications

The combination of b1 and b0 specifies the message format for EDATA/PEDATA. When b1:b0 = 0:0, it indicates Message Format I (basic message format), which allows one message to operate on one property of one object. When b1:b0 = 0:1, it indicates Message Format II (compound message format), which allows one message to operate on two or more properties of one object. When b1:b0 = 1:0, it indicates Message Format III (arbitrary message format), of which EDATA/PEDATA section is in an arbitrary format.

Bit b2 indicates whether the EDATA section is enciphered or not. When b2 = 1, it means that the EDATA section is enciphered. When b2 = 0, it means that the EDATA section is not enciphered. Detailed information about enciphered and other secure messages is set forth in Clause 10.

Bit b3 specifies whether the DEA (destination ECHONET address) shown in Figures 3 and 4 is a broadcast address or an individual address. When b3 = 1, it indicates that a broadcast address is stipulated by the DEA code. When b3 = 0, it indicates that an individual address is stipulated by the DEA code. Broadcast address codes are discussed in 7.1.3.

Bits b4, b5, and b6 constitute a routing hop counter, which can be manipulated only by ECHONET routers. When a message received at one subnet of an ECHONET router is forwarded to another subnet, the counter is incremented. For every transmission from an ordinary node, a hop count of 0 is used. The relationship between b4, b5, and b6 and the hop count is shown in the table below. The number of hops can be set to a value between 0 and 7.

b6	b5	b4	Hop count (router passes)
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

7.1.3 Source/Destination ECHONET address (SEA/DEA)

This subclause provides detailed specifications for the source ECHONET address (SEA) and destination ECHONET address (DEA) shown in Figure 3. Figure 4 shows the configuration of the source ECHONET address (SEA) and the destination ECHONET address (DEA) prevailing when an individual address is stipulated by setting b3 of EHD to 0.

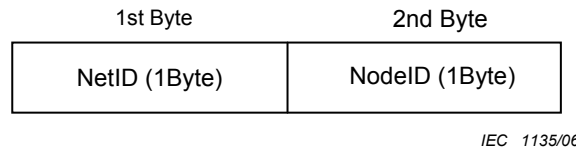


Figure 3 – Configuration of SEA and DEA when an individual address is specified

When b3 of EHD is set to 1 to specify a broadcast, the destination ECHONET address (DEA) becomes a code indicating a broadcast message for a specific ECHONET address group (including a general broadcast). The DEA configuration in this case is shown in Figure 4. The broadcast target stipulation code is shown in Figures 5 and 6.



Broadcast type stipulation code	Broadcast target stipulation code	Remarks
0x00	Specifies the node groups to be targeted for a broadcast within all subnets. For node group selection, see Figure 5	An intra-domain broadcast. In all subnets within a domain, a broadcast is sent to the nodes stipulated by the broadcast target stipulation code
0x01	Specifies the node groups to be targeted for a broadcast within its own subnet. For node group selection, see Figure 5	An intra-own-subnet broadcast. In the own subnet, a broadcast is sent to the nodes stipulated by the broadcast target stipulation code
0x02	All nodes within the subnet having the Net ID code stipulated by the "broadcast target stipulation code" are targeted	A general broadcast within a specified subnet. A broadcast is sent to all nodes within the subnet stipulated by the broadcast target stipulation code
0x03~0x7F	Reserved for future use	
0x80~0xFF	Open to user	Used when a system manager will manage the system in a collective housing unit or small office building

IEC 1136/06

Figure 4 – DEA (broadcast-stipulated) address configuration

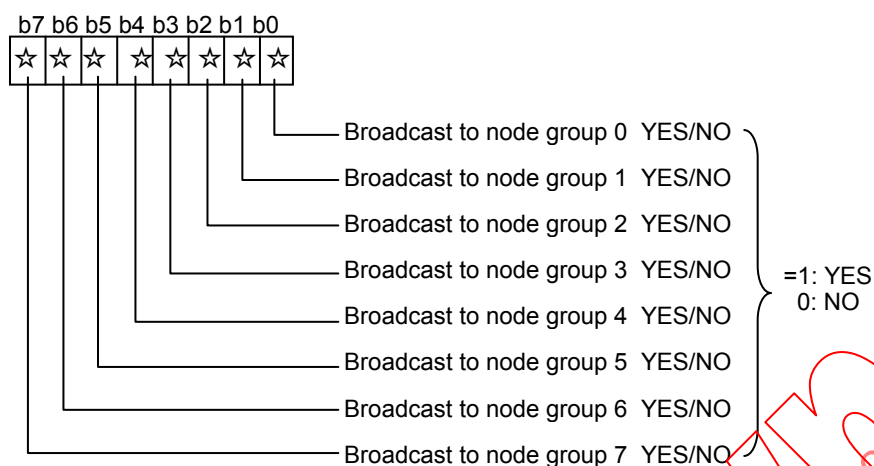


Figure 5 – Broadcast target stipulation code

	0	8	4	C	2	A	6	E	1	9	5	D	3	B	7	F	
0																	Group 0
8																	Group 1
4																	Group 2
C																	Group 3
2																	Group 4
A																	Group 5
6																	Group 6
E																	Group 7
1																	
9																	
5																	
D																	
3																	
B																	
7																	
F																	

IEC 1138/06

Figure 6 – Node group stipulation bit specifications

7.1.4 ECHONET byte counter (EBC)

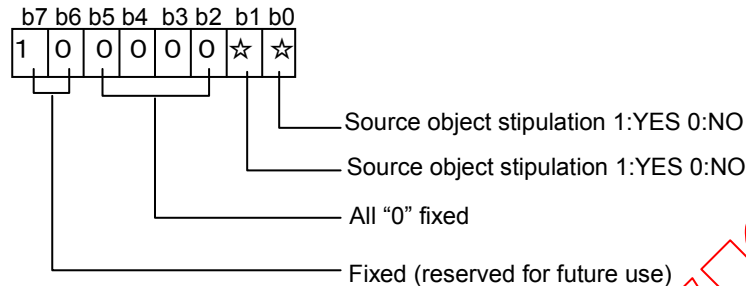
EBC Indicates the size of the ECHONET data region (EDATA region) shown in Figure 1. The size is variable in 1-byte increments. The acceptable EDATA region size ranges from 6 to 256 bytes (0x06 to 0xFF; 0x00 = 256). The lower limit is 6 bytes, which indicates that a message consists of at least 6 bytes. The reason is that either the SEOJ or the DEOJ needs to be specified with the EPC to ESV options specified for a plain message. A 6-byte message can be a message requesting an ESV with the DEOJ specified or a message carrying a "response of processing impossible" for ESV with the SEOJ specified.

7.1.5 ECHONET data (EDATA)

The DATA region for messages exchanged by the ECHONET communication middleware. Maximum size: 256 bytes.

7.1.6 Object message header (OHD)

This subclause provides detailed specifications for the object message header (OHD) shown in Figure 1. The state in which b1 and b0 are both 0 will never occur.



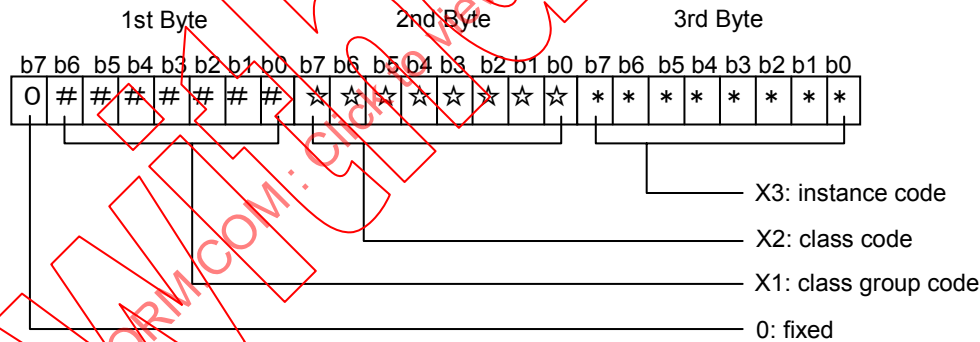
IEC 1139/06

NOTE When b6 and b7 have values other than b6 = 0 and b7 = 1, b0 to b5 will have different meanings. The meanings of bits b0 to b5 when b6 and b7 have values other than b6 = 0 and b7 = 1 will be stipulated in the future (reserved for future use).

Figure 7 – OHD detailed specifications

7.1.7 ECHONET objects (EOJ)

This subclause provides detailed specifications for the source ECHONET object (SEOJ) code and destination ECHONET object (DEOJ) code shown in Figure 1.



IEC 1140/06

NOTE The meanings of the bits when b7 of the 1st byte is 1 will be stipulated in the future (reserved for future use).

Figure 8 – EOJ detailed specifications

ECHONET objects are described using the format [X1.X2] and [X3], with these formats to be specified as shown below. (However, "." is used only for descriptive purposes and does not mean a specific code.) The object class is designated by the combination of X1 and X2, while X3 shows the class instance. A single ECHONET node may contain more than one instance of the same class, in which case X3 is used to identify each one.

The specific items in Table 1 were specified on the basis of JEM 1439 (see Clause 9). Detailed specifications for the objects shown here will be developed over time and, during this phase, specifications for the objects themselves (i.e., present/not present) will be further reviewed.

The instance code 0x00 is regarded as a special code (code for specifying all instances). When a DEOJ for which this code is specified is received, it is handled as a code specifying a broadcast to all instances of a specified class.

- X1 : class group code 0x00-0x7F. For details, refer to Table 1.
- X2 : class code 0x00-0xFF. For detailed examples, refer to Tables 2 to 8.
- X3 : instance code 0x00-0xFF.

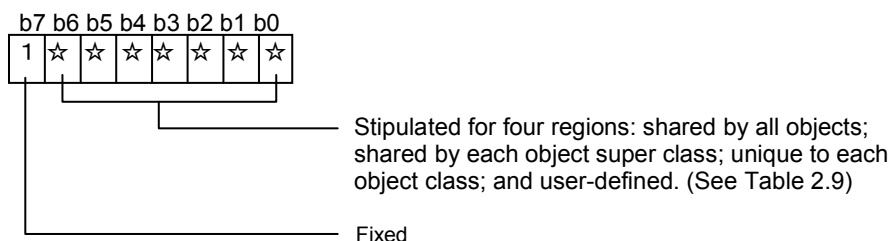
The identifier code is used when more than one of the same class specified by [X1.X2] exists within the same node. However, 0x00 is used as a general broadcast to all instances of class specified with [X1.X2].

Table 1 – List of class group codes

Class group code	Group name	Remarks
0x00	Sensor-related device class group	
0x01	Air conditioner-related device class group	
0x02	Housing/facility-related device class group	Includes lighting
0x03	Cooking/housework-related device class group	
0x04	Health-related device class group	
0x05	Management/control-related device class group	
0x06	AV-related device class group	
0x07~0x0C	Reserved for future use	
0x0D	Service class group	
0x0E	Profile class group	
0x0F	User definition class group	
0x10~0x1F	Communications definition class group for stipulation of status notification method	
0x20~0x2F	Communications definition class group for stipulation of setting control reception method	
0x30~0x3F	Communications definition class group for linked settings (action settings)	
0x40~0x4F	Communications definition class group for linked settings (trigger settings)	
0x50~0x5F	Secure communication access property set-up class	
0x60~0x7F	Reserved for future use	

7.1.8 ECHONET property (EPC)

This subclause provides detailed specifications for the ECHONET property (EPC) code shown in Figure 1. The EPC specifies a service target function. Each object stipulated by X1 (class group code) and X2 (class code), described in 7.1.7, is specified here. (When a specified object changes, the target function also changes even when the code remains unchanged. However, the detailed specifications are designed to ensure that, whenever possible, the same functions will have the same code.) Specific code values for each object are stipulated in 8.3. These codes correspond to the object property identifiers in the object definitions.



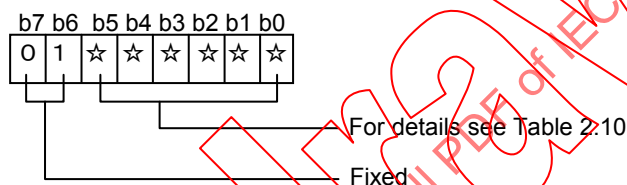
IEC 1141/06

NOTE When b7 = 0, the other bits will be defined differently.

Figure 9 – EPC detailed specifications

7.1.9 ECHONET service (ESV)

This subclause provides detailed specifications for the ECHONET service (ESV) code shown in Figure 1.



IEC 1142/06

NOTE In cases other than when b7:b6 = 0:1, the meaning of values b0 – b5 will be specified separately.

Figure 10 – ESV detailed specifications

This code stipulates manipulation of the properties stipulated by EPC. The three main kinds of operations are shown below. There are also two kinds of responses: the “response,” which is given when the stipulated properties exist; and the “response not possible,” which is given when the requested properties (including array elements) do not exist or when the stipulated service cannot be processed.

“Request”/“Response” (response/response not possible)/“Notification”

A “response” is considered to be a reply to a “request” that requires a response; when the object stipulated in the DEOJ exists, as a rule it is either “response” or “response not possible” (stipulated processing cannot be accepted, or the stipulated object exists but the property does not). When the request requires no response and the stipulated object does not exist, no response is made.

There are two types of “notification”: one for transmitting the own-property information autonomously and the other for sending a response to a notification request. However, these two types have the same code.

Three specific operations are provided: write (response required/no response required), read, and notification (notification/notification with response required). The 12 operations shown below are set in consideration of whether or not the content of the given property is an array.

- Property value write (response required/no response required)
- Property value read
- Property value notification

- d) Property value array-element-stipulated write (response required/no response required)
- e) Property value array-element-stipulated read
- f) Property value array-element-stipulated notification
- g) Property value array-element-stipulated addition (response required/no response required)
- h) Property value array-element-stipulated deletion (response required/no response required)
- i) Property value array-element-stipulated existence confirmation
- j) Property value array element addition (response required/no response required)
- k) Property value notification (response required)
- l) Property value array-element-stipulated notification (response required)

The relationship between the message configuration (presence or absence of SEOJ and DEOJ) and EPC and ESV is described below.

- The EPC in an ECHONET message stipulating only SEOJ indicates the properties of the sender object specified in SEOJ. Here, ESV contains an autonomous “notification” or “notification” or “response” in response to a request for properties specified in SEOJ and EPC. If ESV is a “request” in such a case, the received message is treated as an illegal message.
- The EPC in an ECHONET message stipulating only DEOJ indicates the properties of the destination object specified in DEOJ. Here, ESV contains a “request” regarding the properties specified in DEOJ and EPC. If ESV is a “response” or a “notification” in such a case, the received message is treated as an illegal message.
- For ECHONET messages stipulating both SEOJ and DEOJ, the ESV value is used to determine whether the EPC is stipulated by the SEOJ or the DEOJ. When the ESV is a “response” or a “notification”, the EPC is considered to be a component of the object specified by SEOJ and is viewed as a “response” or “notification” directed towards the object stipulated in the DEOJ. When the ESV is a “request,” the EPC is considered to be a component of the DEOJ and is viewed as a “request” from the object stipulated in the SEOJ.

Tables 1 through 3 show specific ESV code assignments based on the content described above. Specific descriptions of a) through l) above are provided in (1) through (12) of the remarks column in the relevant table. In the figures given in (1) through (12), the DEOJ for “requests” is shown as an individually stipulated code. However, when the DEOJ indicates a broadcast to all instances of a specified class (when the DEOJ's X3 = 0x00), a response is transmitted with both “process-not-possible” response and “response” configured for each target instance. Note that in the table, the “array elements” described above are presented as “elements.”

Table 2 – List of ESV codes for requests

Service code (ESV)	ECHONET service content	Symbol	Remarks
0x60	Property value write request (no response required)	SetI	(1)
0x61	Property value write request (response required)	SetC	
0x62	Property value read request	Get	(2)
0x63	Property value notify request	INF_REQ	(3)
0x64	Property value element-stipulated write request (no response required)	SetMI	(4)
0x65	Property value element-stipulated write request (response required)	SetMC	
0x66	Property value element-stipulated read request	GetM	(5)
0x67	Property value element-stipulated notify request	INFM_REQ	(6)
0x68	Property value element-stipulated add request (no response required)	AddMI	(7)
0x69	Property value element-stipulated add request (response required)	AddMC	
0x6A	Property value element-stipulated delete request (no response required)	DelMI	(8)
0x6B	Property value element-stipulated delete request (response required)	DelMC	
0x6C	Property value element existence confirm request	CheckM	(9)
0x6D	Property value element add request (no response required)	AddMSI	(10)
0x6E	Property value element add request (response required)	AddMSC	
0x6F	Reserved for future use		

Table 3 – List of ESV codes for response/notification

Service code (ESV)	ECHONET service content	Symbol	Remarks
0x71	Property value write response	Set_Res	ESV=0x61 response (1)
0x72	Property value read response	Get_Res	ESV=0x62 response (2)
0x73	Property value notification	INF	*1 (3)
0x74	Property value notification (response required)	INFC	(11)
0x75	Property value element-stipulated write response	SetM_Res	ESV=0x65 response (4)
0x76	Property value element-stipulated read response	GetM_Res	ESV=0x66 response (5)
0x77	Property value element-stipulated notify	INFM	*2 (6)
0x78	Property value element-stipulated notify (response required)	INFMC	(12)
0x79	Property value element-stipulated add response	AddM_Res	ESV=0x69 response (7)
0x7A	Property value notify response	INFC_Res	ESV=0x74 response (11)
0x7B	Property value element-stipulated delete response	DelM_Res	ESV=0x6B response (8)
0x7C	Property value element-stipulated existence confirm response	CheckM_Res	ESV=0x6C response (9)
0x7D	Property value element-stipulated notify response	INFMC_Res	ESV=0x78 response (12)
0x7E	Property value element add response	AddMS_Res	ESV=0x6E response (10)
0x70, 0x7F	Reserved for future use		

NOTE 1 Used for autonomous property value notification and for 0x63 response.

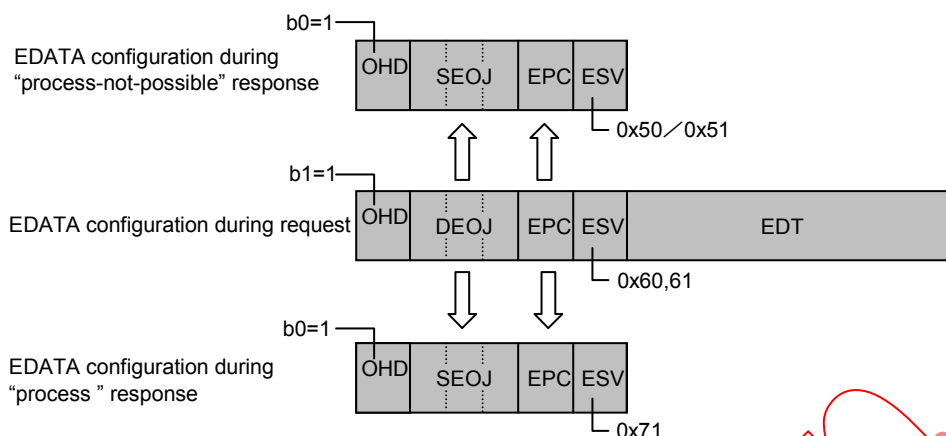
NOTE 2 Used for autonomous property value notification and for 0x67 response.

Table 4 – List of ESV codes for “response-not-possible” responses

Service code (ESV)	ECHONET service content	Symbol	Remarks
0x50	Property value write “process-not-possible” response	SetI_SNA	ESV=0x60 response not possible (1)
0x51	Property value write “process-not-possible” response	SetC_SNA	ESV=0x61 response not possible (1)
0x52	Property value read “process-not-possible” response	Get_SNA	ESV=0x62 response not possible (2)
0x53	Property value notify “process-not-possible” response	INF_SNA	ESV=0x63 response not possible (3)
0x54	Property value element-stipulated write request “process-not-possible” response	SetMI_SNA	ESV=0x64 response not possible (4)
0x55	Property value element-stipulated write request “process-not-possible” response	SetMC_SNA	ESV=0x65 response not possible (4)
0x56	Property value element-stipulated read request “process-not-possible” response	GetM_SNA	ESV=0x66 response not possible (5)
0x57	Property value element-stipulated notify request “process-not-possible” response	INFM_SNA	ESV=0x67 response not possible (6)
0x58	Property value element-stipulated add request “process-not-possible” response	AddMI_SNA	ESV=0x68 response not possible (7)
0x59	Property value element-stipulated add request “process-not-possible” response	AddMC_SNA	ESV=0x69 response not possible (7)
0x5A	Property value element-stipulated delete request “process-not-possible” response	DelMI_SNA	ESV=0x6A response not possible (8)
0x5B	Property value element-stipulated delete request “process-not-possible” response	DelMC_SNA	ESV=0x6A response not possible (8)
0x5C	Property value element-stipulated existence confirm request “process-not-possible” response	CheckM_SNA	ESV=0x6C response not possible (9)
0x5D	Property value element add request “process-not-possible” response	AddMSI_SNA	ESV=0x6D response not possible (10)
0x5E	Property value element add request “process-not-possible” response	AddMSC_SNA	ESV=0x6E response not possible (10)
0x5F	Reserved for future use		

a) Property value write service [0x60,0x61,0x71,0x50,0x51]

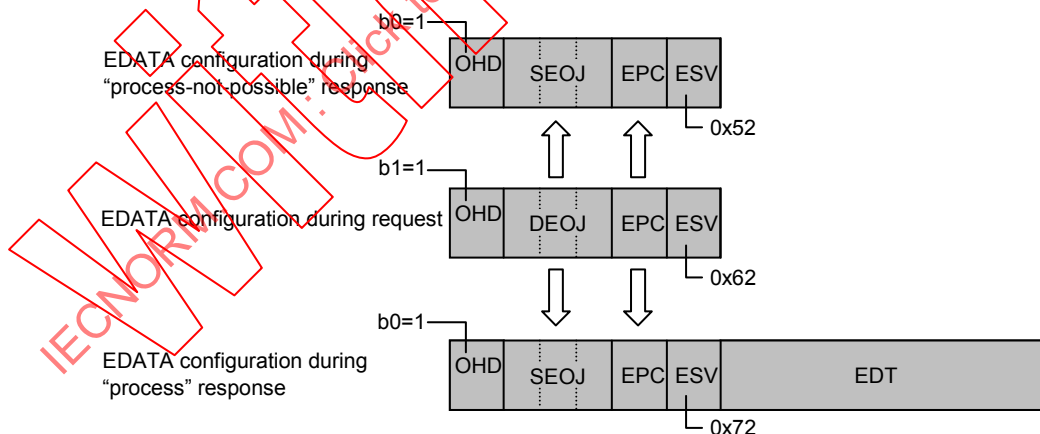
In the case of a “request” (0x60,0x61), this indicates a request to write the content shown in EDT to the property stipulated in the EPC of the object stipulated in DEOJ. In response to this “request,” when a value indicating a response is stipulated (0x61) and the request is to be (or has already been) received, “response” (0x71) is returned. This “response” is not a processing implementation response. When the request is not to be received, or when the stipulated DEOJ exists but the stipulated EPC does not exist, “response not possible” (0x50,0x51) is returned. In the response frame format, SEOJ represents the value of the object stipulated by the request, and the relevant property is set in EPC. When the relevant object itself does not exist, neither “response” nor “response not possible” is returned. Also, the “response” message DEA is defined as the requesting entity (i.e., the request message SEA).



When EDATA stipulates SEOJ during a "request," the EOJ stipulated by SEOJ in EDATA during the "request" is allocated as a DEOJ (b1 of OHD is also set to 1), in the case of both "response not possible" and "response."

b) Property value read service [0x62,0x72,0x52]

In the case of a "read" (0x62), this indicates a request to read the content of the property stipulated in the EPC of the object stipulated in the DEOJ. In response to this "read," when the request is to be (or has already been) accepted, "response" (0x72) is returned. When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, "response not possible" (0x52) is returned. In the response frame format, the value of the object stipulated by the request is set in SEOJ, the requested property is set in EPC, and the value of the requested property (i.e., the read content) is set in EDT. When "response not possible" is returned, nothing is written to the EDT. When the relevant object itself does not exist, neither "response" nor "response not possible" is returned. Also, the "response" message DEA is defined as the requesting entity (i.e., the request message SEA).

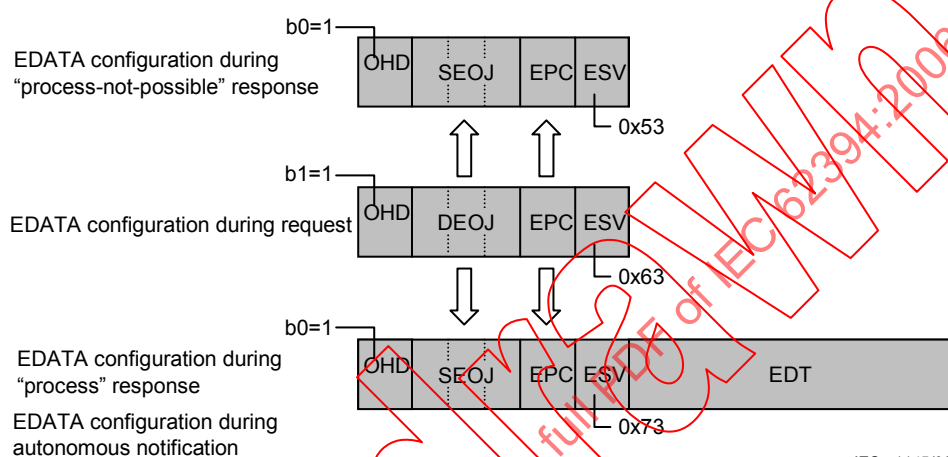


When EDATA stipulates SEOJ during a "request," the EOJ stipulated by SEOJ in EDATA during the "request" is allocated as a DEOJ (b1 of OHD is also set to 1), in the case of both "response not possible" and "response."

c) Property value notification service [0x63,0x73,0x53]

There are two types of "notification": the notification sent as a response to a "notify request" (0x63) and the autonomous notification which is unrelated to notify requests. The codes for the two types are identical. (Here, notification in response to a "notify request" signifies an announcement that does not specify the property value [content], while an autonomous notification is a voluntary announcement that was not made in response to a request.) In the case of a "notify request" (0x63), this indicates a request to notify (by general broadcast;

hereafter “announce” will signify a general broadcast to the entire domain) the content of the property stipulated in the EPC of the object stipulated in the DEOJ. In response to this “notify request,” when the request was accepted, a “response” (0x73) value is notified; when the request is not to be accepted, a “response not possible” response (0x53) value is returned. In the response frame format, the value of the object stipulated by the request is set in SEOJ, the requested property is set in EPC, and the value of the requested property (i.e., the notification content) is set in EDT. Here, DEA is set to general broadcast, but when “response not possible” is returned, nothing is written to the EDT, and the DEA sets the EA value of the requester. When the relevant object itself does not exist, neither “response” nor “response not possible” is returned. In the case of an autonomous “notification”, the DEA is set to a general broadcast for a required status change notification. In the other cases, however, the DEA can be set as desired regardless of whether “broadcast” or “individual” is selected.



IEC 1145/06

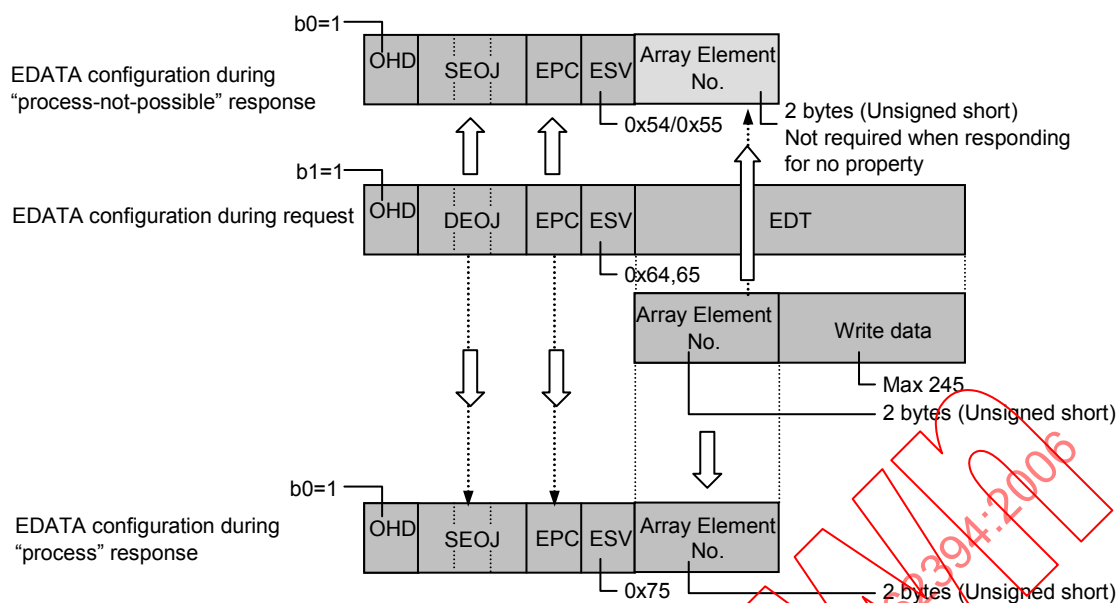
When EDATA stipulates SEOJ during a request, the EOJ stipulated by SEOJ in EDATA during the “request” is allocated as a DEOJ. In the case of both “response not possible” and “process,” the EOJ stipulated in the SEOJ in the EDATA during “request” is allocated as a DEOJ within the EDATA (b1 of OHD is also set to 1). In the case of autonomous notification, the required notification of status change does not add a DEOJ; in all other cases, the addition of a DEOJ is optional.

d) Property value element-stipulated write service [0x64,0x65,0x75,0x54,0x55]

In the case of a “request” (0x64, 0x65), this indicates a request to write the value stipulated in the EDT (includes array element number and write request value data) of the property stipulated in the EPC of the object stipulated in the DEOJ. In response to this “request,” when a value to process the response is stipulated, and when the request is to be (or has already been) accepted, a “response” (0x75) is returned. However, this “response” is not a processing implementation response. When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, and when the stipulated DEOJ and EPC exist but the array element does not, “response not possible” (0x54, 0x55) is returned.

In the frame format for response, the value of the object stipulated by the request is SEOJ, and the relevant property is set in EPC. When the relevant object itself does not exist, neither “response” nor “response not possible” is returned.. Also, the “response” message DEA is defined as the requesting entity (i.e., the request message SEA).

When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, the EDT of “response not possible” is the array element number of a “request” and when the stipulated DEOJ exists but the stipulated EPC does not exist, the EDT of “response not possible” does not exist.

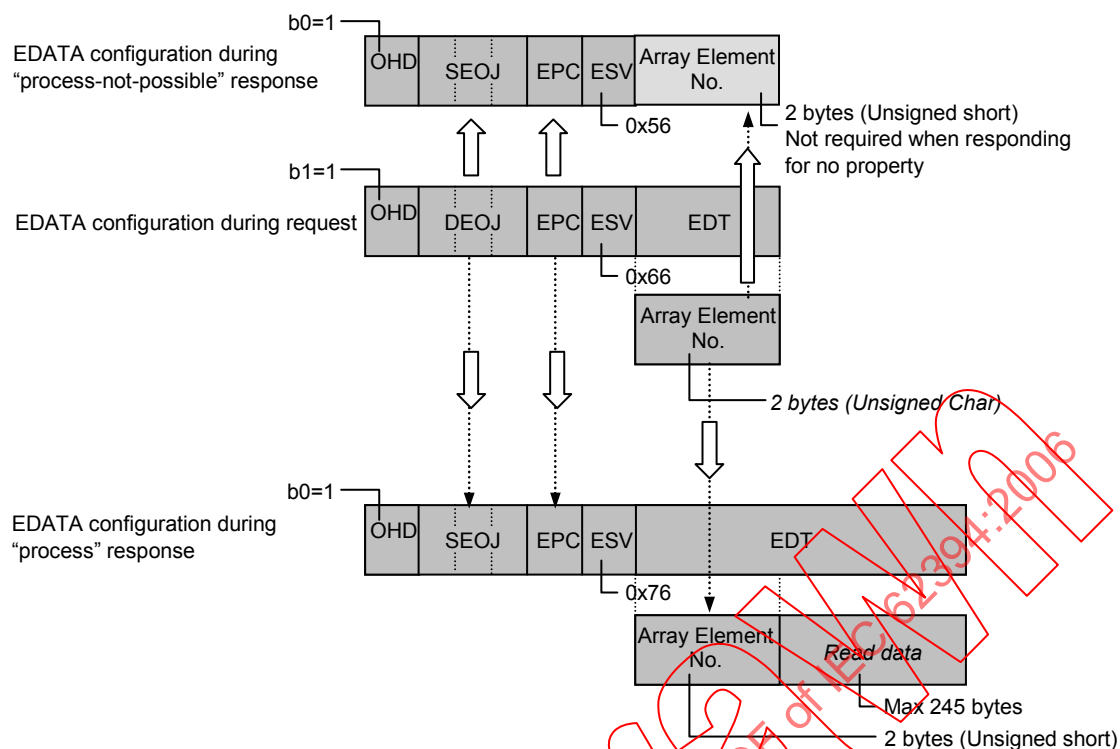


IEC 1146/06

The content of each array element number in an array format property is defined separately for each property. When the stipulated (array) element does not exist, "response not possible" is returned. Also, when the EDATA stipulates SEOJ during a "request," the EOJ stipulated in SEOJ by EDATA during the "request" is allocated as a DEOJ within EDATA (b1 of OHD is also set to 1) in the case of both "response not possible" and "response."

e) Property value element-stipulated read service [0x66,0x76,0x56]

In the case of a "read" (0x66), this indicates a request to read the content stipulated in the array element indicated in the EDT (includes array element number data to be read) of the property stipulated in the EPC of the object stipulated in the DEOJ. In response to this "read," when the request is to be (or has already been) accepted, "response" (0x76) is returned. When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not, and when the stipulated DEOJ and EPC exist but the array element does not, "response not possible" (0x56) is returned. In the frame format for response, the value of the object stipulated by the request is set in SEOJ, and the relevant property is set in EPC and the value of the relevant property (data to be read) is set in EDT. In the case of "response not possible", when the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, EDT of "response not possible" is the array element number of a "request" and when the stipulated DEOJ exists but the stipulated EPC does not exist, "response not possible" EDT of "response not possible" does not exist. When the relevant object itself does not exist, neither "response" nor "response not possible" is returned. Also, the "response" message DEA is defined as the requesting entity (i.e., the request message SEA).



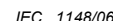
IEC 1147/06

The content of each array element number in an array format property is defined separately for each property. When the stipulated array element (element) does not exist, "response not possible" is returned. Also, when EDATA stipulates SEOJ during a "request," the EOJ stipulated in the SEOJ by EDATA during the "request" is allocated as a DEOJ within the EDATA (b1 of OHD is also set to 1) in the case of both "response not possible" and "response."

f) Property value element-stipulated notification service [0x67,0x77,0x57]

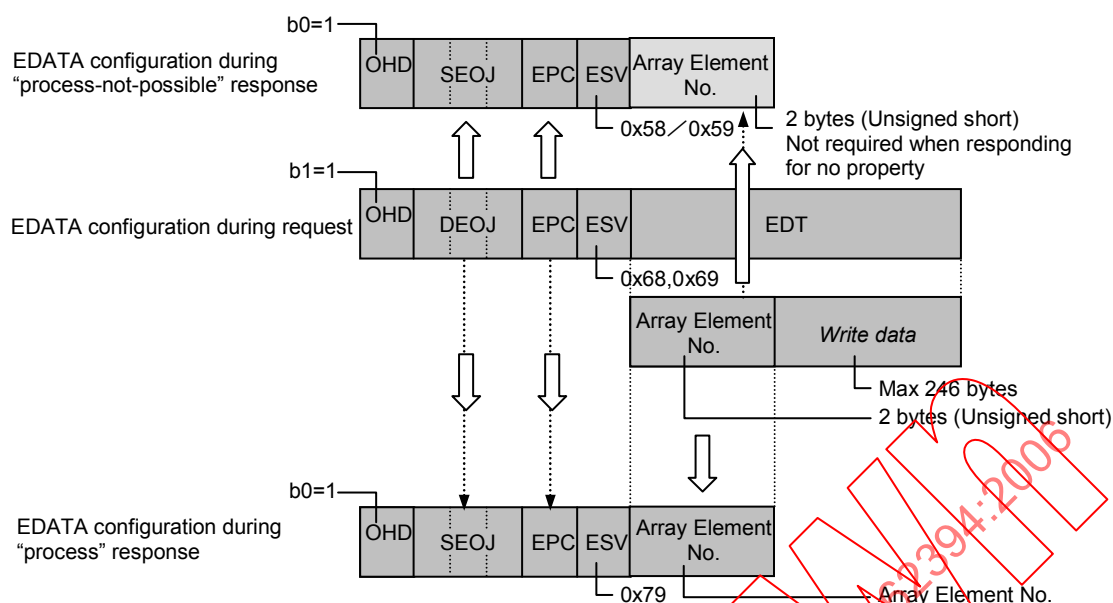
There are two types of "notification": notification sent in response to a "notify request" (0x67); and autonomous notification, which is unrelated to notify requests. The two types are not distinguished from each other in the codes. (Here, notification in response to a "notify request" signifies an announcement that does not specify the property value [content], while an autonomous notification is a voluntary announcement that was not made in response to a request from someone.) In the case of a "notify request" (0x67), this indicates a request to notify (announce) the content of the array element number stipulated in the EDT of the property stipulated in the EPC of the object stipulated in the DEOJ. In response to this "notify request," when the request was accepted, an array element value (content) is announced as a "response" (0x77). When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not, and when the stipulated DEOJ and EPC exist but the array element does not, "response not possible" (0x57) is returned. In the frame format for response, the value of the object stipulated by the request is set in SEOJ, the requested property is set in EPC, and the value of the requested array element number and its array element value (i.e., the notification content) is set in EDT. Here, DEA is set to general broadcast, but when "response not possible" is returned, and the DEA sets the EA value of the requester. When the relevant object itself does not exist, neither "response" nor "response not possible" is returned.

When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, EDT of "response not possible" is the array element number of a "request" and when the stipulated DEOJ exists but the stipulated EPC does not exist, the EDT of "response not possible" does not exist.



g) Property value element-stipulated addition [0x68,0x69,0x58,0x59,0x79]

When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, EDT of “response not possible” is the array element number of a “request” and when the stipulated DEOJ exists but the stipulated EPC does not exist, the EDT of “response not possible” does not exit.



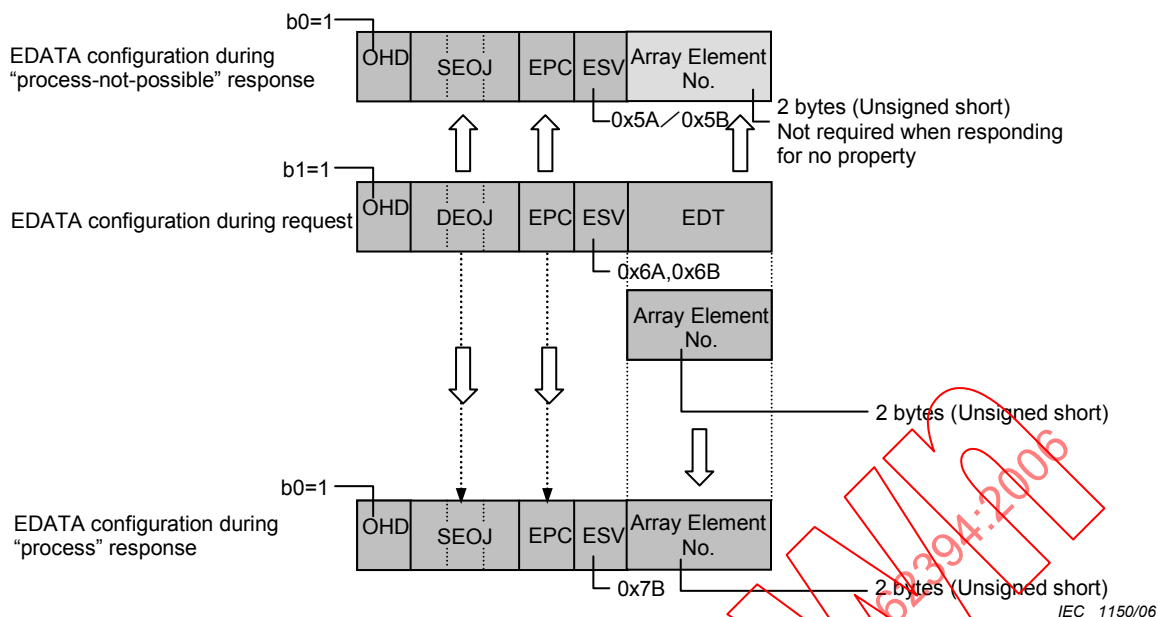
IEC 1149/06

The content of each array element number in an array format property is defined separately for each property. When the stipulated array element (element) does not exist, "response not possible" is returned. Also, when EDATA stipulates SEOJ during a "request," the EOJ stipulated in the SEOJ by EDATA during the "request" is allocated as a DEOJ within the EDATA (b1 of OHD is also set to 1) in the case of both "response not possible" and "response."

h) Property value element-stipulated deletion [0x6A, 0x6B, 0x5A, 0x5B, 0x7B]

In the case of a "request" (0x6A, 0x6B), this indicates a request to delete the array element indicated in the EDT (array element number) from the property stipulated in the EPC of the object stipulated in the DEOJ. In response to this "request," when a value indicating implementation of the response (0x6B) is stipulated, and when the request is to be (or has already been) accepted, a "response" (0x7B) is returned. However, this "response" is not a processing implementation response. When the request is not to be accepted (including cases in which the deletion is not to be implemented), or when the stipulated DEOJ exists but the stipulated EPC does not, "response not possible" (0x5A, 0x5B) is returned. In the frame format for response, the value of the object stipulated by the request is set in SEOJ, and the relevant property is set in EPC. When the relevant object itself does not exist, neither "response" nor "response not possible" is returned. Also, the "response" message DEA is defined as the requesting entity (i.e., the request message SEA).

When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, EDT of "response not possible" is the array element number of a "request" and when the stipulated DEOJ exists but the stipulated EPC does not exist, the EDT of "response not possible" does not exist.

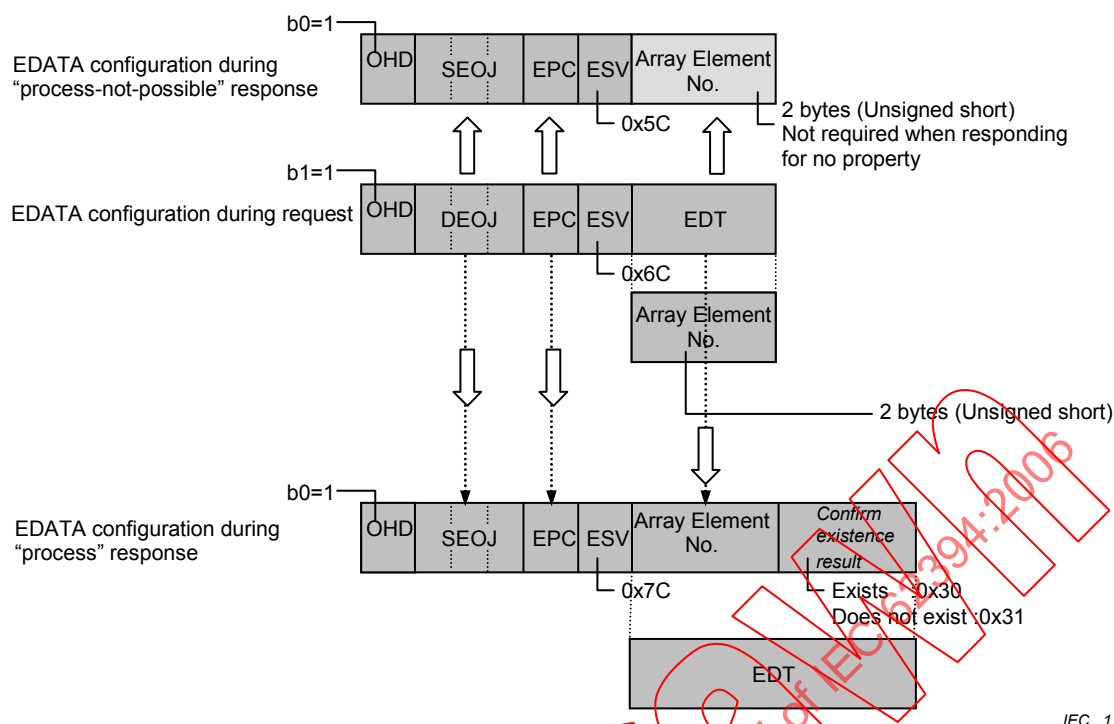


The content of each array element number in an array format property is defined separately for each property. When the stipulated array element (element) does not exist, "response not possible" is returned. Also, when EDATA stipulates SEOJ during a "request," the EOJ stipulated in the SEOJ by EDATA during the "request" is allocated as a DEOJ within the EDATA (b1 of OHD is also set to 1) in the case of both "response not possible" and "response."

i) Property value element-stipulated existence confirmation [0x6C, 0x5C, 0x7C]

In the case of a "request" (0x6C), this indicates a request to confirm the existence of the array element indicated in the EDT (includes array element number value information) in the property stipulated in the EPC of the object stipulated in the DEOJ. When the request is to be (or has already been) accepted, a "response" (0x7C) is returned. When the request is to be rejected (cannot be processed by the ESV) or when the specified DEOJ exists but the specified EPC does not exist, a "process not possible" (0x5C) is returned. In the frame format for response, the value of the object stipulated by the request is set in SEOJ, and the relevant property is set in EPC. When the relevant object itself does not exist, neither "response" nor "response not possible" is returned. Also, the "response" message DEA is defined as the requesting entity (i.e., the request message SEA).

When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not exist, EDT of "response not possible" is the array element number of a "request" and when the stipulated DEOJ exists but the stipulated EPC does not exist, the EDT of "response not possible" does not exist.



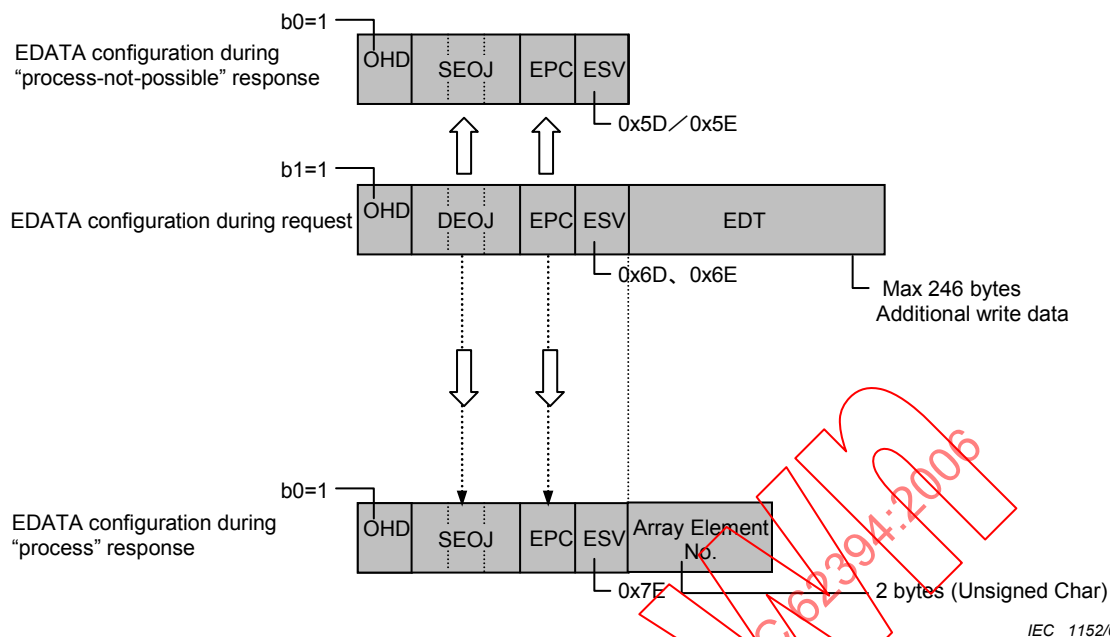
IEC 1151/06

The content of each array element number in an array format property is defined separately for each property. Also, when EDATA stipulates SEOJ during a "request," the EOJ stipulated in the SEOJ by EDATA during the "request" is allocated as a DEOJ within the EDATA (b1 of OHD is also set to 1) in the case of both "response not possible" and "response."

j) Property value element addition [0x6D, 0x6E, 0x5D, 0x5E, 0x7E]

In the case of a "request" (0x6D, 0x6E), this indicates a request to newly add an array element to the property stipulated in the EPC of the object stipulated in the DEOJ, and to write to the newly added array element the value data stipulated in the EDT. In response to this "request," when a value indicating implementation of the response (0x6E) is stipulated, and when the request is to be (or has already been) accepted, a "response" (0x7F) is returned. However, this "response" is a processing implementation response, and the added array element number is returned as an EDT. When the request is not to be accepted, or when the stipulated DEOJ exists but the stipulated EPC does not, "response not possible" (0x5D, 0x5E) is returned. In the frame format for response, the value of the object stipulated by the request is set in SEOJ, and the relevant property is set in EPC. When the relevant object itself does not exist, neither "response" nor "response not possible" is returned. Also, the "response" message DEA is defined as the requesting entity (i.e., the request message SEA).

For "response not possible", EDT does not exit.



IEC 1152/06

The content of each array element number in an array format property is defined separately for each property. Also, when EDATA stipulates SEOJ during a "request," the EOJ stipulated in the SEOJ by EDATA during the "request" is allocated as a DEOJ within the EDATA (b1 of OHD is also set to 1) in the case of both "response not possible" and "response."

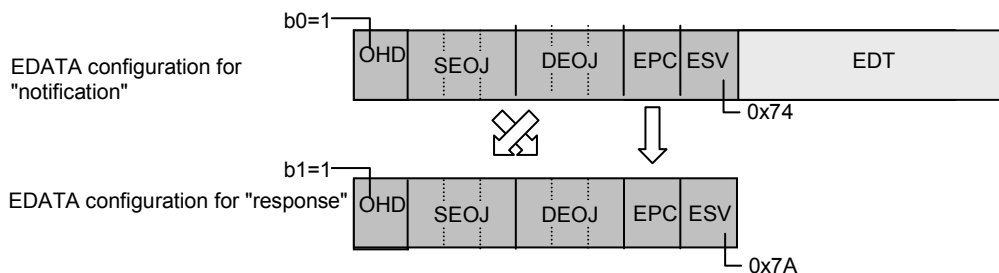
k) Property value notification (response required) [0x74, 0x7A]

The "notification (response required)" (0x74) autonomously notifies a specific node of the property value stipulated by the EPC of the SEOJ stipulated object and requests a response. The response process for this "notification (response required)" varies depending on whether the DEOJ is specified.

When the DEOJ is not specified, the "response" (0x7A) for autonomous notification reception is returned at all times.

When the DEOJ is specified, on the other hand, the subsequent process varies depending on whether the specified DEOJ exists. If the specified DEOJ exists, the "response" (0x7A) for autonomous notification reception is returned. If the specified DEOJ does not exist, the message is discarded.

If a node receives a "notification (response required)" for which a broadcast is specified, the node discards the message.



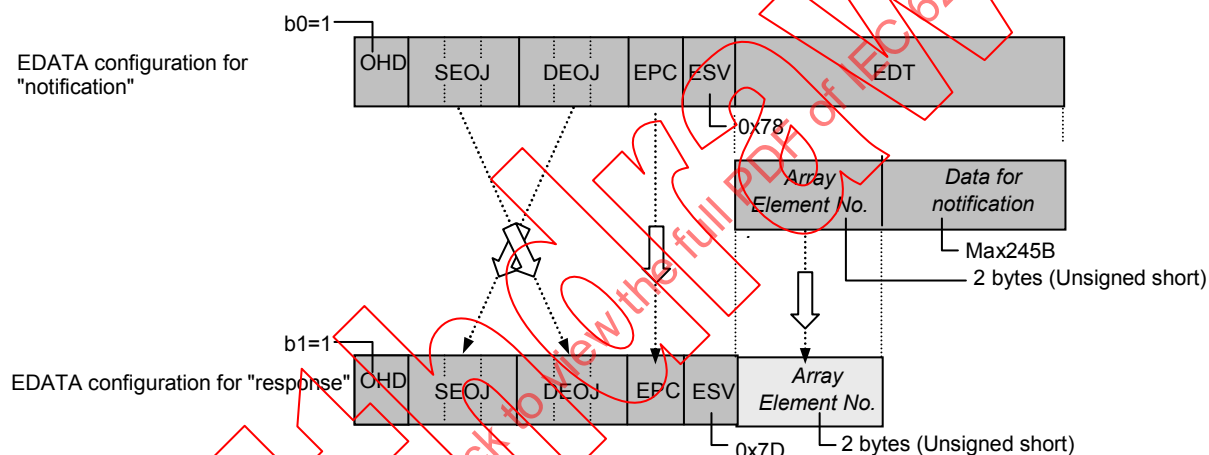
l) Property value element-stipulated notification (response required) [0x78, 0x7D]

The "notification (response required)" (0x78) autonomously notifies a specific node of the array element value stipulated by the EDT (array element number) of the property stipulated by the EPC of the SEOJ-stipulated object, and requests an acknowledgment. The response message format and response process for this "notification (response required)" varies depending on whether the DEOJ is specified.

When the DEOJ is not specified, the "response" (0x7D) for notification reception is returned at all times.

When the DEOJ is specified, on the other hand, the subsequent process varies depending on whether the specified DEOJ exists. If the specified DEOJ exists, the "response" (0x7D) for notification reception is returned. If the specified DEOJ does not exist, the message is discarded.

If a node receives a "notification (response required)" for which a broadcast is specified, the node discards the message.



IEC 1153/06

The services shown in Tables 2 through 4 are specified for each property. Regarding those stipulated as services that must be incorporated in each property, if they have the functions of that property and disclose via communications (read/write notification, etc.), this indicates that they must be processed. Processing of services for each property is specified in 8.3 and in the access rules column of the object class detailed specification tables. Access rules indicate all services that can be implemented. In this specification, the following nine access rules are specified.

Set	Processes services related to write requests for non-array property values (Performs processing indicated in (1))
Get	Processes services related to read requests for non-array property values (Performs processing indicated in (2) (3) and (11))
SetM	Processes services related to write requests for array property values (Performs processing indicated in (4))
GetM	Processes services related to read requests for array property values (Performs processing indicated in (5) (6) and (12))
AddM	Processes services related to element-stipulated add requests for array property values (Performs processing indicated in (7))
DelIM	Processes services related to delete requests for array property values (Performs processing indicated in (8))
CheckM	Processes services related to existence confirm requests for array property value elements (Performs processing indicated in (9))
AddMS	Processes services related to non-array-element-stipulated add requests for array property values (Performs processing indicated in (10))
Anno	Processes non-array property value notification services (Performs processing indicated in (3) and (11))
AnnoM	Processes array property value notification services (Performs processing indicated in (6) and (12))

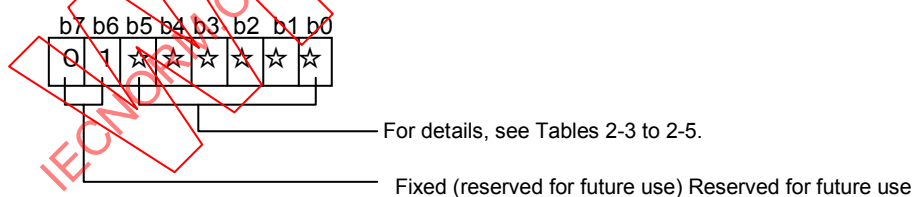
The above processing is specified for each property; there is no mixed stipulation of Set and SetM or of Get and GetM.

7.1.10 ECHONET property value data (EDT)

This subclause presents detailed specifications for the code for the ECHONET property value data (EDT) range shown in Figure 1. EDT consists of data for the relevant ECHONET property (EPC), such as status notification or specific setting and control by an ECHONET service (ESV). Detailed specifications are provided for the size, code value, etc., of EDT for each EPC (see Clause 8).

7.1.11 Compound ECHONET Service (CpESV)

This subclause provides detailed specifications for the compound ECHONET service (CpESV) code shown in Figure 1.



IEC 1154/06

Figure 10 – EpESV configuration

NOTE When bits b7 and b6 are 0 and 1, respectively, the meanings of bits b0 to b5 are stipulated separately.

The service provided by this code is used when the compound message format is used. It specifies a simultaneous action for two or more properties stipulated by the EPC. However, it does not stipulate the order of operations. The order of property operations is an implementation issue.

Three types of operations are provided: request, response, and notification. The response is subdivided into two types: "accepted" response and "process-not-possible" request. The "accepted" response is used when the service request in relation to all the EPC-stipulated properties is accepted. The "process not possible" request is used when one or more

specified properties do not exist or when the specified service cannot be processed for one or more properties.

- Request
- Response ("accepted" response/"process not possible" response)
- Notification

The "response" is a response to a "request" that requires a response. It shall be returned when a DEOJ-stipulated object exists. When the service processing request related to all the EPC-stipulated properties is accepted, the "accepted" response shall be returned. If the processing request related to one or more specified properties cannot be accepted or if the object exists but one or more properties do not exist, "process not possible" shall be returned. When the "request" does not require any response or when the specified object does not exist, no "response" will be returned.

Further, "write" (response-required write/no-response-required write), "read" and "notification" (autonomous notification/response-required notification) are regarded as specific operations. Therefore, the following five types are set. Regarding the OpESV for compound messages, array element properties are not targeted.

- a) Property value write request (no response required)
- b) Property value write request (response required)
- c) Property value read request
- d) Property value notification
- e) Property value notification (response required)

The CpESV and message configurations (presence of SEOJ and DEOJ) and their relationship to EPC and ESV are described below.

- (a) The EPC of an ECHONET message in which only the SEOJ is specified indicates the property of the SEOJ-stipulated source object. In this case, the "response", "notification", or autonomous "notification" concerning the "request" related to two or more SEOJ-/EPC-stipulated properties is positioned in the CpESV. When the CpESV is a "request" while this configuration is employed, the associated message shall be handled as an erroneous message.
- (b) The EPC of an ECHONET message in which only the DEOJ is specified indicates the property of the DEOJ-stipulated destination object. In this case, the "request" related to two or more DEOJ-/EPC-stipulated properties is positioned in the CpESV. When the CpESV is a "response" or "notification" while this configuration is employed, the associated message shall be handled as an erroneous message.
- (c) The EPC of an ECHONET message in which the SEOJ and DEOJ are both specified is such that the CpESV value determines whether the target object is stipulated by the SEOJ or DEOJ. When the CpESV is a "response" or "notification", it is concluded that the EPC forms a SEOJ-stipulated object and that the "response" or "notification" is addressed to a DEOJ-stipulated object. When the CpESV is a "request", on the other hand, it is concluded that the EPC forms a DEOJ and that the "request" is issued from a SEOJ-stipulated object.

Tables 5 to 7 show specific CpESV code assignments. The details of items a) through e) above are given in (1) through (5) of the remarks column of the tables). The figures in (1) through (5) presume that the DEOJ for a "request" is an individually specified code. However, when the DEOJ indicates an instance general broadcast, a response is transmitted with both "process not possible" response and "response" configured for each target instance. Figure 3 shows a sequence diagram, which indicates the relationships between individual CpESVs. The codes marked "reserved for future use" in the tables are to be stipulated in the future and shall not be used.

Table 5 – List of CpESV codes for request/notification

Service code (CpESV)	ECHONET service content	Symbol	Remarks
0x60	Property value write request (no response required)	CpSetI	(1)
0x61	Property value write request (response required)	CpSetC	(2)
0x62	Property value read request	CpGet	(3)
0x63~0x6F	Reserved for future use		

Table 6 – List of CpESV Codes for "accepted" response

Service code (CpESV)	ECHONET service content	Symbol	Remarks
0x71	Property value write "accepted" response	CpSet_Res	CpESV=61 response (2)
0x72	Property value read "accepted" response	CpGet_Res	CpESV=62 response (3)
0x73	Property value notification	CpINF_Res	(4)
0x74	Property value notification (response required)	CpINFC	(5)
0x7A	Property value notification response	CpINFC_Res	CpESV=74 response (5)
0x75~0x79, 0x7B~0x7F	Reserved for future use		

Table 7 – List of CpESV codes for "process-not-possible" response

Service code (CpESV)	ECHONET service content	Symbol	Remarks
0x50	Property value write "process not possible" response (1)	CpSetI_SNA	CpESV=60 "process not possible" response (1)
0x51	Property value write "process not possible" response (2)	CpSetC_SNA	CpESV=61 "process not possible" response (2)
0x52	Property value read "process not possible" response	CpGet_SNA	CpESV=62 "process not possible" response (3)
0x5F	Message length excessive	CpOverflow	Response to be returned when the response message is too long
0x53~0x5E	Reserved for future use		

a) Property value write request (requiring no response) service [0x60, 0x50]

The write request requiring no response (CpESV = 0x60) requests that the EDT-stipulated contents be written into the EPC-stipulated properties of the DEOJ-stipulated object. The order of write operations is not stipulated. The response from a request-processing node is as indicated below.

- 1) When a processing request for all properties are accepted
No response will be made.
- 2) When one or more properties relevant to the request do not exist, a processing request to one or more properties cannot be accepted, or an array property is targeted
A write "process not possible" response (1) (CpESV = 0x50) will be returned.

3) When the object relevant to the request does not exist

No response will be made.

4) When two or more identical properties exist in the request message

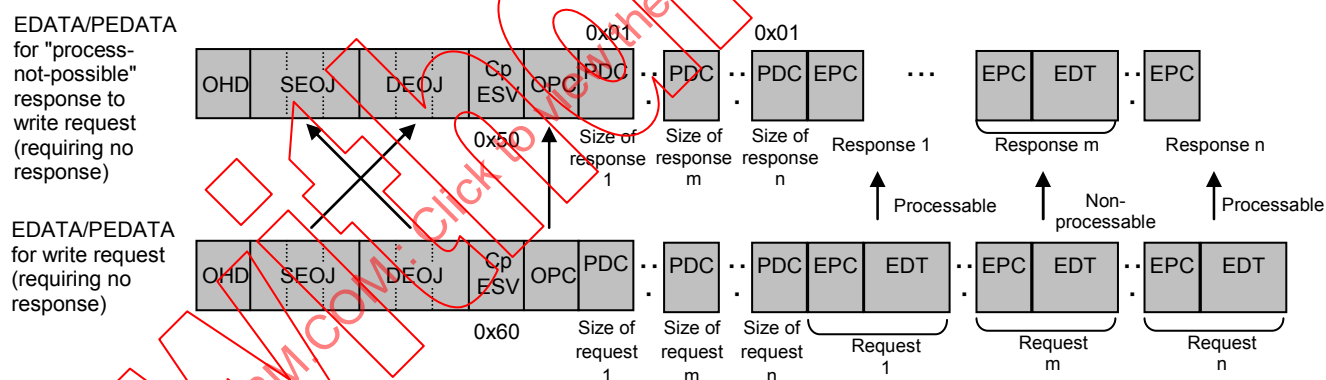
Individual processes will be performed on the presumption that differing requests are issued. A response will be made in accordance with the processing results.

NOTE The order of processes depends on the implementation. Therefore, the resulting final property status and value also depend on the implementation.

The message structure of a write "process-not-possible" response to a property value write request (requiring no response) is such that the object code of the request destination becomes the SEOJ and that the object code of the request source becomes the DEOJ. The OPC takes the same value as in the request message.

For requests (request 1 to request n) that relate to non-existent properties and process requests that are rejected, both the PDC and EDT use the same values as those used in the write request. For requests related to properties for which processing requests are accepted, the PDC value is 0x01 and the EDT value is omitted. As for the EPC, the EPC in the request message is used as is. If the target object does not exist, neither the "response" nor the "process not possible" response is returned.

An appropriate value for the OHD shall be specified in accordance with the SEOJ/DEOJ configuration in the message. Figure 12 shows the relationship between a write request requiring no response and write addition response for situations where request m cannot be accepted. The EPC sequence in the request message shall be equal to the EPC sequence in the write "process-not-possible" response message.



IEC 1155/06

Figure 12 – Relationship between write request (requiring no response) and write "process-not-possible" response

b) Property value write request (requiring a response) service [0x61, 0x71, 0x51]

The write request requiring a response (CpESV = 0x61) requests that the EDT-stipulated contents be written into the EPC-stipulated properties of the DEOJ-stipulated object. The order of write operations is not stipulated. The response from a request-processing node is as indicated below.

1) When a processing request for all properties are accepted

A write "accepted" response (CpESV = 0x71) will be returned.

2) When one or more properties relevant to the request do not exist, a processing request to one or more properties cannot be accepted, or an array property is targeted

A write "process not possible" response (CpESV = 0x51) will be returned.

3) When the object relevant to the request does not exist

No response will be made.

4) When two or more identical properties exist in the request message

Individual processes will be performed on the presumption that differing requests are issued. A response will be made in accordance with the processing results.

NOTE The order of processes depends on the implementation. Therefore, the resulting final property status and value also depend on the implementation.

The message structure of a write "process-not-possible" response to a property value write request (requiring a response) is such that the object code of the request destination becomes the SEOJ and that the object code of the request source becomes the DEOJ. The OPC takes the same value as in the request message.

For requests (request 1 to request n) that relate to non-existent properties and process requests that are rejected, both the PDC and EDT use the same values as those used in the write request. For requests related to properties for which processing requests are accepted, the PDC value is 0x01 and the EDT value is omitted. As for the EPC, the EPC in the request message is used as is. If the target object does not exist, neither the "response" nor the "process not possible" response is returned.

The message structure of a write "accepted" response is such that the object code of the request destination becomes the SEOJ and the object code of the request source becomes the DEOJ. The OPC and subsequent values are omitted.

An appropriate value for the OHD shall be specified in accordance with the SEOJ/DEOJ configuration in the message. Figure 13 shows the relationships among a write request requiring a response, a write "accepted" response, and a write "process-not-possible" response for situations where request m cannot be accepted. The EPC sequence in the request message must be equal to the EPC sequence in the write "process-not-possible" response message.

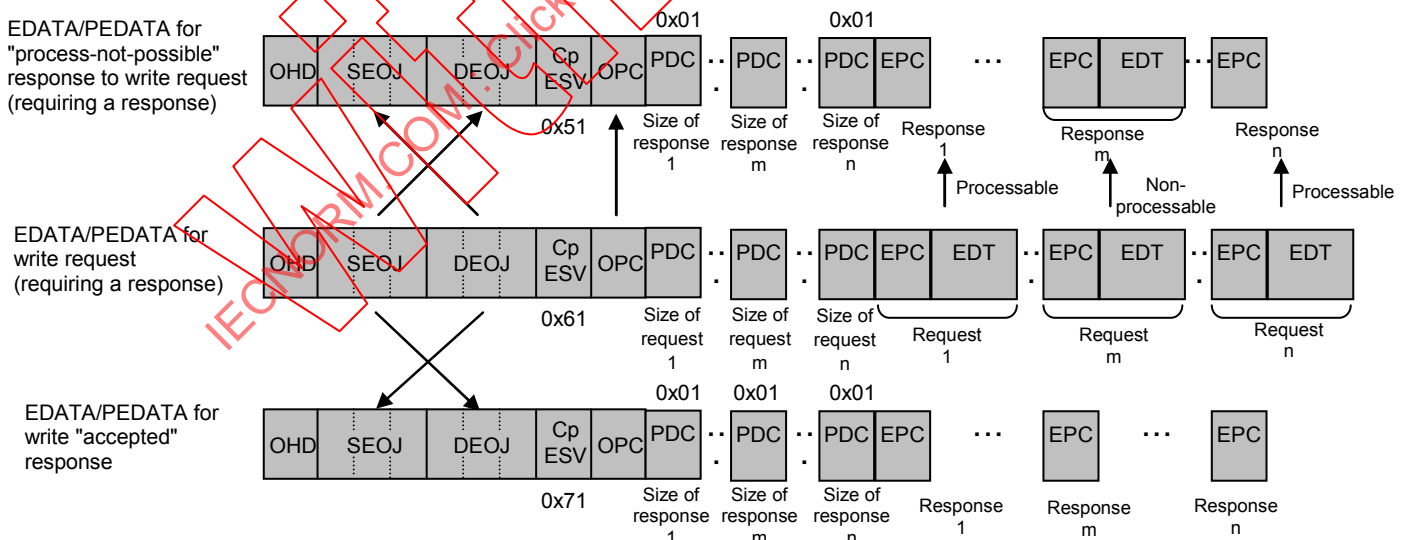


Figure 13 – Relationship among write request (requiring a response), write "accepted" response, and write "process-not-possible" response

c) Property value read request service [0x62, 0x72, 0x52, 0x5F]

The property value read request (CpESV = 0x62) requests that the contents of EPC-stipulated properties of the DEOJ-stipulated object be read. The order of read operations is not stipulated. The response from a request-processing node is as indicated below.

- 1) When a processing request for all properties are accepted
A read "accepted" response (CpESV = 0x72) will be used to return all the read values.
- 2) When one or more properties relevant to the request do not exist, a processing request to one or more properties cannot be accepted, or an array property is targeted
A write "process not possible" response (CpESV = 0x52) will be used to return the values of the read properties.
- 3) When the object relevant to the request does not exist
No response will be made.
- 4) When two or more identical properties exist in the request message
Individual processes will be performed on the presumption that differing requests are issued. A response will be made in accordance with the processing results.

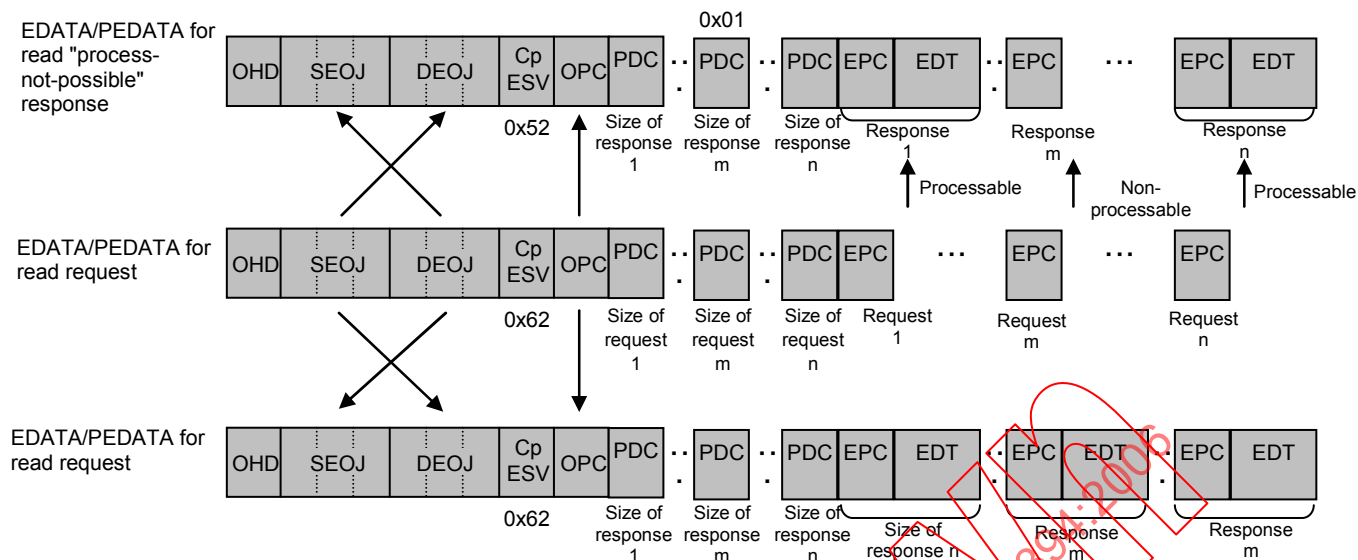
NOTE The order of processes depends on the implementation. Therefore, if two or more property states are read, the resulting final status depends on the implementation.

The message structure of a read "process-not-possible" response is such that the object code of the request destination becomes the SEOJ and the object code of the request source becomes the DEOJ. The OPC takes the same value as in the request message.

For requests (request 1 to request n) that relate to non-existent properties and process requests that are rejected, the PDC value is 0x01 and the EDT value is omitted. For requests related to properties for which processing requests are accepted, the read value is placed in the EDT and the total number of EPC and EDT bytes is regarded as the PDC. If the target object does not exist, neither the "response" nor the "process-not-possible" response is returned.

The message structure of a read "accepted" response is such that the object code of the request destination becomes the SEOJ and the object code of the request source becomes the DEOJ. The read value is placed in the EDT, and the total number of EPC and EDT bytes is regarded as the PDC.

An appropriate value for the OHD must be specified in accordance with the SEOJ/DEOJ configuration in the message. Figure 14 shows the relationships among a read request, a read "accepted" response, and a read "process-not-possible" response for situations where request m cannot be accepted. The EPC sequence in the request message must be equal to the EPC sequence in the read "accepted" response and read "process-not-possible" response messages.



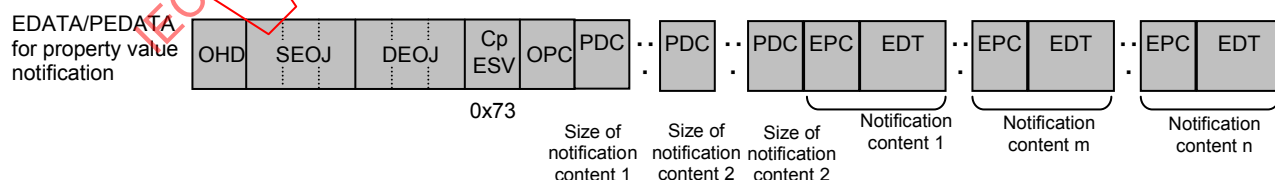
IEC 1157/06

Figure 14 – Relationship among read request (requiring a response), read "accepted" response, and read "process-not-possible" response

As is obvious from Figure 14, the read "accepted" response message is longer than the read response message. Therefore, the maximum permissible message length may be exceeded when an attempt is made to return all the property values that are read in compliance with the request. In such a situation, a response will be made using the message length overflow service code (CpESV = 0x5F). In this case, the responding side can determine the number of property values to be returned; however, the sequence of such properties shall be the same as in the request message.

d) Property value notification service [0x73]

The property value notification (CpESV = 0x73) reads the contents of EPC-stipulated properties and reports them to the DEOJ-stipulated object. When the DEOJ is not contained in the message, it is a notification to nodes. Either "individual" or "broadcast" can be selected for addressing purposes. The order of property value notifications is not stipulated. Nodes receiving this message will not return a response.



IEC 1158/06

Figure 15 – Relationship between notification request and notification response

e) Property value notification (requiring a response) service [0x74, 0x7A]

The property value notification requiring a response (CpESV = 0x74) reads the contents of EPC-stipulated properties and reports them to the DEOJ-stipulated object. When the DEOJ is not contained in the message, it is a notification to a node. Only "individual" is available for addressing purposes. The order of property value notifications is not stipulated. The response from a node receiving this message is as indicated below.

1) When a notification is accepted

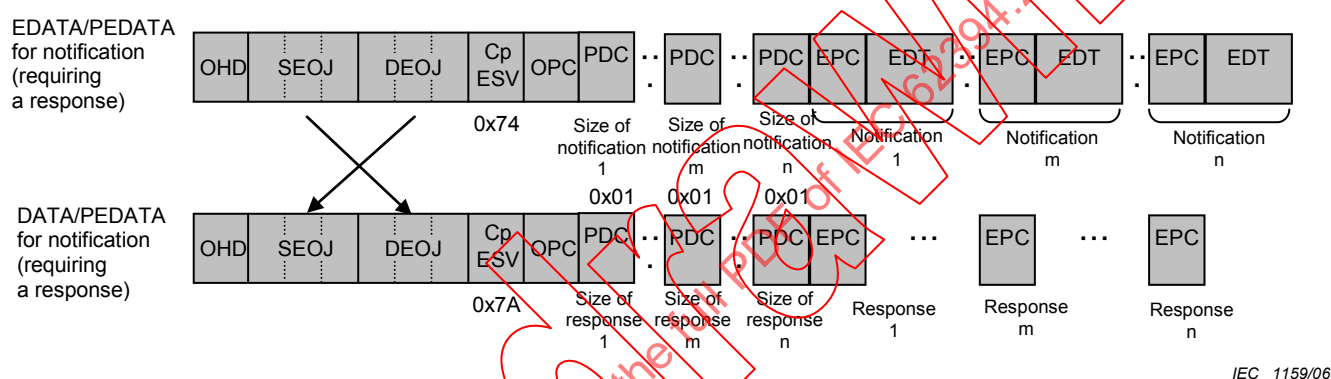
A property value notification response (CpESV = 0x7A) will be returned.

2) When the DEOJ-stipulated object does not exist

No response will be made.

The message structure of the notification response is such that the object code of the request destination becomes the SEOJ and the object code of the request source becomes the DEOJ. The OPC takes the same value as in the request message.

An appropriate value for the OHD must be specified in accordance with the SEOJ/DEOJ configuration in the message. Figure 16 shows the relationship between the property value notification (requiring a response) service and property value notification response service. The EPC sequence in the property value notification request service message shall be equal to the EPC sequence in the property value notification response service message.



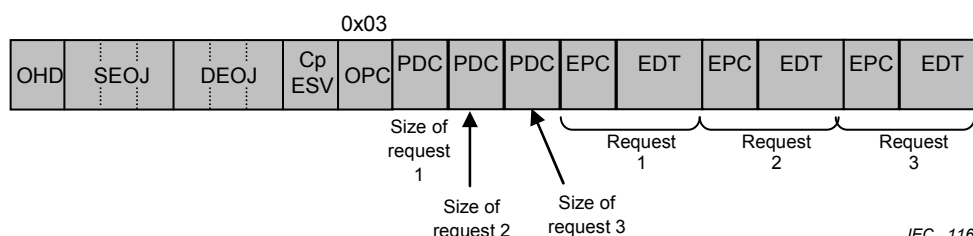
IEC 1159/06

Figure 16 – Relationship between property value notification (requiring a response) and property value notification response

7.1.12 Processing target property counter (OPC)

The processing target property counter is used in the compound message format only. It consists of one byte. In a compound message, the processing target property counter retains the number of properties targeted for a write or read operation. This counter can retain the value 1 or greater. Therefore, a compound message is allowed to exist even when the number of simultaneously operable properties is only one. The maximum number of simultaneously operable properties is limited by the maximum permissible message length.

If, for instance, there are three requests as shown in Figure 17, the processing target property counter is 0x03.



IEC 1160/06

Figure 17 – Processing target property counter for three requests

7.1.13 Property data counter (PDC)

The processing data counter is used in the compound message format only. It retains the number of bytes in the ECHONET property code (EPC) and ECHONET data (EDT), which follow the proper data counter. If, for instance, the ECHONET data sizes for requests 1, 2, and 3 are 2 bytes, 1 byte, and 5 bytes, respectively, the values placed in the first, second, and third property data counters are 0x03, 0x02, and 0x06, respectively, as shown in Figure 18.

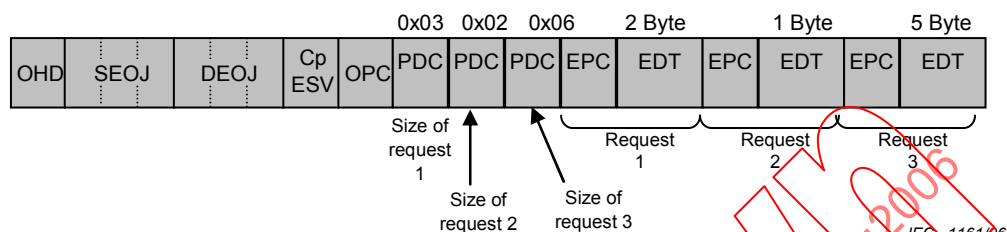


Figure 18 – Property data counter

8 ECHONET objects: detailed specifications

8.1 Basic concept

This subclause will specify specific values for the class codes of ECHONET objects processed in the ECHONET Communication Middleware, whose types and overview were given in Clause 7, along with property configurations and detailed specifications for property configurations. In the case of class codes, rather than providing entirely new specifications, standards already being studied by the industry were applied whenever possible to capitalize on past work. Regarding object properties, the operands (control content) of JEM-1439 were analysed and referred to. ECHONET objects described in this subclause and in the appendix are divided into three main classes: device objects, profile objects, and communications definition objects. In terms of the code structure, they will be divided into the class groups shown below. This subclause presents the shared ECHONET property specifications and object super classes that form ECHONET objects.

a) Device objects

- Sensor-related device class group
- Air-conditioning-related device class group
- Housing-related device class group
- Cooking/housework-related device class group
- Health-related device class group
- Management and control-related device class group
- AV-related device class group

b) Profile objects

- Profile class group

c) Communications definition objects

- Sensor-related device communications definition class group
- Air-conditioning-related device communications definition class group
- Housing-related device communications definition class group
- Cooking/housework-related device communications definition class group
- Health-related device communications definition class group

- Management and control-related device communications definition class group
- Profile communications definition class group
- AV-related device communications definition class group

Each ECHONET node must implement a device object for at least one representative device.

8.2 ECHONET properties: basic specifications

This subclause will discuss the specifications shared by all ECHONET object classes, of which details are provided in this subclause and in 8.3.

8.2.1 ECHONET property value data types

The ECHONET property value is expressed as an unsigned integer when the value is a non-negative integer value; it is expressed as a signed integer when the value is an integer value containing negatives.

When the value is a small value, it is handled as a fixed point type; when it is a non-negative small value, it is treated as an unsigned integer; and when it is a small value containing negatives, it is treated as a signed integer. Data types and sizes are specified individually for each property.

Although the property data size is specified individually for each property, property value data of 2 bytes or larger comprises ECHONET communication middleware messages as ECHONET property value data (EDT) beginning from the significant byte.

8.2.2 Property value range

The definition range for the ECHONET properties specified in this subclause and in the appendix, and the treatment of property values when the actual device value operating range differs from them, is specified below.

- When the actual device property value operating range is smaller than the ECHONET property definition range and the actual device property value assumes the upper and lower limit values, the upper and lower limit values of the operating range are considered to be the property values.

Assuming that the ECHONET property definition range is 0x00-0xFD (0°C–253°C) and the corresponding actual device operating range is 0x0A–0x32 (10°C–50°C), when the actual device value is the upper limit value (50 °C) of the operating range, the upper limit value 0x32 (50 °C) of the actual device operating range is considered to be the ECHONET property value, and when the actual device property value is the lower limit value (10°C), the lower limit value 0x0A (10°C) is considered to be the ECHONET property value.

- When the actual device property value operating range as ECHONET property is larger than the ECHONET property definition range and the actual device property value assumes a value outside the ECHONET property definition range, a code showing an underflow or overflow becomes the property value.
- Assuming that the ECHONET property definition range is 0x00-0xFD (0°C–253°C) and the corresponding actual device operating range is -10°C–300°C, when the actual device value assumes a value below the ECHONET property definition range, the underflow code 0xFE becomes the property value; when the actual device property value assumes a value above the ECHONET property definition range, the overflow code 0xFF becomes the property value.

Table 8 shows the underflow and overflow codes for each data type.