

Part V

ECHONET Common Lower-layer Communication
Interface Specification

Release information (as of August 7th 2001)

a) Version1.0	March 18 th	2000 released	Open to the consortium member
	July	2000	Open to the public
b) Version1.01	May 23 rd	2001	Open to the public
c) Version2.00	August 7 th	2001	Open to the consortium member

Notes: On and after Version2.00, Powerline communication protocol has drawn together as Powerline communication A.

Specifications published by the ECHONET CONSORTIUM are established without regard to industrial property rights(patents, utility models and so on). ECHONET CONSORTIUM has no responsibility for industrial property rights over contents of specifications.

Contents

Chapter 1	Overview	1-1
1.1	Basic Concept	1-1
1.2	Positioning on Communication Layers	1-2
Chapter 2	ECHONET Common Lower-layer Communications Interface Function Specification	2-1
2.1	List of ECHONET Common Lower-layer Communication Interface Functions.....	2-1
2.2	Level 1 ECHONET Common Lower-layer Communication Interface Detailed Specification.....	2-3
Chapter 3	Level 1 ECHONET Common Lower-layer Communication Interface Specification	3-1
3.1	List of Level 1 ECHONET Common Lower-layer Communication Interface	3-1
3.2	Level 1 ECHONET Common Lower-layer Communication Interface Detailed Specification.....	3-3
Chapter 4	Level 2 ECHONET Common Lower-layer Communication Interface Specification.....	4-1
4.1	List of Level 2 ECHONET Common Lower-layer Communication Interface for C Language	4-2
4.2	C Language-oriented Level 2 ECHONET Common Lower-layer Communication Interface Detailed Specification.....	4-3
4.2.1	ClcGetDevID	4-4
4.2.2	ClcInit.....	4-5
4.2.3	ClcRequestRun.....	4-6
4.2.4	ClcSetTrouble.....	4-7
4.2.5	ClcReset	4-8
4.2.6	ClcSuspend.....	4-9
4.2.7	ClcWakeUp	4-10
4.2.8	ClcGetProData	4-11
4.2.9	ClcGetStatus.....	4-12
4.2.10	ClcLowInit	4-13
4.2.11	ClcLowRequestRun	4-15
4.2.12	ClcLowRequest.....	4-16
4.2.13	ClcLowSuspend	4-17
4.2.14	ClcLowWakeUp.....	4-18
4.2.15	ClcGetLowProData.....	4-19
4.2.16	ClcGetLowStatus	4-21

4.2.17	ClcSendData.....	4-22
4.2.18	ClcGetSendResult	4-23
4.2.19	ClcSendCancel	4-24
4.2.20	ClcReceiveData.....	4-25
4.2.21	ClcGetNodeID	4-26
4.2.22	ClcSetNodeID	4-27

Chapter 1 Overview

1.1 Basic Concept

The ECHONET Common Lower-layer Communication Interface Specification in Part 5 is provided to specify the software interface to implement processing and information exchange between the ECHONET communications processing block and the Protocol Difference Absorption Processing Block, which are described in Section 1.2 “Positioning on Communication Layers”. This Common Lower-layer Communication Interface permits describing the processing specification of the ECHONET communications processing block in common form without consideration of differences in Lower-layer Communications Software specification. This Common Lower-layer Communication Interface specification provides rules on functions as level 1 and level 2 of the Common Lower-layer Communication Interface specification for the case in which input/output data items and concrete language are specified with regard to APIs, based on the assumption that they are supported by the Protocol Difference Absorption Processing Block. Levels 1 and 2 of the Common Lower-layer Communication Interface specification are based on the concept of levels 1 and 2 of the basic API.

1.2 Positioning on Communication Layers

The interface specification described in this Section is provided in a form that permits absorbing differences in the Lower-layer Communication Software so that the ECHONET communications processing block may control the portion under the Protocol Difference Absorption Processing Block without regard to differences in the Lower-layer Communication Software .

The shaded area in Fig. 1.1 shows the positioning of the Common Lower-layer Communication Interface .

The Common Lower-layer Communication Interface is positioned between the Protocol Difference Absorption Processing Block and the ECHONET communications processing block to implement mutual exchange.

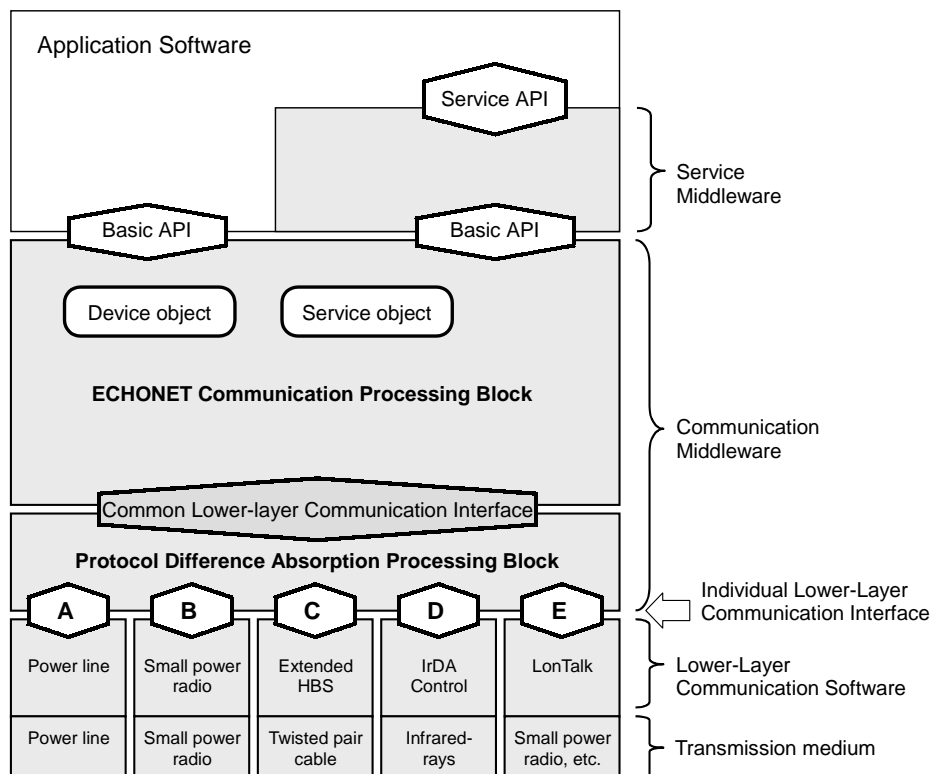


Fig. 1.1 Positioning of Common Lower-layer Communication Interface

Chapter 2 ECHONET Common Lower-layer Communication Interface Function Specification

2.1 List of ECHONET Common Lower-layer Communication Interface Functions

Table 2.1 shows a list of ECHONET Common Lower-layer Communication Interface functions supported by the Protocol Difference Absorption Processing Block. The Protocol Difference Absorption Processing Block shall be provided with these functions. The function specification of each API is explained in the following section.

Table 2.1 List of ECHONET Common Lower-layer Communication Interface Functions (1/2)

No.	API name	Function	Remarks
1	Request for initialization	Requests initialization of Protocol Difference Absorption Processing Block and Lower-layer Communication Software .	
2	Request for operation start	Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software start operation.	
3	Fault notice	Notifies Protocol Difference Absorption Processing Block of fault (error) status of high-order layer from ECHONET communications processing block.	
4	Request for reset	Asks Protocol Difference Absorption Processing Block and Lower-layer Communication Software for reset processing.	
5	Request for suspension	Asks Protocol Difference Absorption Processing Block and Lower-layer Communication Software to suspend operation.	
6	Request for operation restart	Asks Protocol Difference Absorption Processing Block and Lower-layer Communication Software to restart operation.	
7	Protocol Difference Absorption Processing Block profile getting	Gets profile data (static information) from Protocol Difference Absorption Processing Block.	
8	Lower-layer Communication Software profile getting	Gets profile data (static information) from Lower-layer Communication Software .	
9	Protocol Difference Absorption Processing Block status acquisition	Gets dynamic status (processing fault, address redundancy, etc.) information from Protocol Difference Absorption Processing Block.	
10	Lower-layer Communication Software status acquisition	Gets dynamic status (processing fault, address redundancy, etc.) from Lower-layer Communication Software .	

**Table 2.1 List of ECHONET Common Lower-layer
Communication Interface Functions (2/2)**

No.	API name	Function	Remarks
11	Request for data transmission	Requests that Protocol Difference Absorption Processing Block send data.	
12	Transmission result acquisition	Requests data transmission result from Protocol Difference Absorption Processing Block.	
13	Request for transmission stop	Requests that data transmission be stopped.	
14	Request for received data	Asks Protocol Difference Absorption Processing Block to deliver received data.	
15	NodeID acquisition	Gets NodeID recognized by Protocol Difference Absorption Processing Block.	
16	NodeID setting	Sets NodeID for Protocol Difference Absorption Processing Block	

2.2 Level 1 ECHONET Common Lower-layer Communication Interface Detailed Specification

(1) Request for initialization

Requests initialization of the Protocol Difference Absorption Processing Block and the Lower-layer Communication Software . Upon receiving this request, the Protocol Difference Absorption Processing Block executes initialization for the Protocol Difference Absorption Processing Block and the specified Lower-layer Communication Software using the specified information. However, normal operation is started when “Request for operation start” is received.

(2) Request for operation start

Requests that operation of the Protocol Difference Absorption Processing Block and the Lower-layer Communication Software be started.

(3) Fault notice

Notifies the Protocol Difference Absorption Processing Block of the fault (error) status of the high-order layer from the ECHONET communications processing block.

(4) Request for reset

Requests that the Protocol Difference Absorption Processing Block and the Lower-layer Communication Software reset. Upon receiving this request, the Protocol Difference Absorption Processing Block executes reset processing for all Lower-layer Communication Software if the request relates to the resetting of the Protocol Difference Absorption Processing Block proper, and waits in the initial status. If the request relates to the resetting of the discrete Lower-layer Communication Software , said block executes only reset processing for the specified Lower-layer Communication Software .

(5) Request for suspension

Requests suspension from the protocol difference processing block and the Lower-layer Communication Software . Upon receiving this request, the Protocol Difference Absorption Processing Block shall not accept any request other than “Request for operation restart” or “Request for reset” from the ECHONET communications processing block if the received request relates to the suspension of the Protocol Difference Absorption Processing Block proper, even when it comes from the ECHONET communications processing block or the Lower-layer Communication Software . If the request relates to the suspension of the discrete Lower-layer Communication Software , said block executes only suspend processing for the specified Lower-layer Communication Software .

- (6) Request for operation restart
Requests that the Protocol Difference Absorption Processing Block and the Lower-layer Communication Software clear suspension status and restart operation. Upon receiving this request, the Protocol Difference Absorption Processing Block restarts operation of the specified software as well as the self-block.
- (7) Protocol Difference Absorption Processing Block profile acquisition
Asks the Protocol Difference Absorption Processing Block for profile data of the Protocol Difference Absorption Processing Block. The profile data requested by this function shall be static information such as the software developer name, version No., etc.
- (8) Lower-layer Communication Software profile acquisition
Asks the Protocol Difference Absorption Processing Block for profile data of the Lower-layer Communication Software. The profile data requested by this function shall be static information such as the software developer name, version No., etc.
- (9) Protocol Difference Absorption Processing Block status acquisition
Asks the Protocol Difference Absorption Processing Block for the status data of the Protocol Difference Absorption Processing Block. The status data requested by this function shall be dynamic information of the Protocol Difference Absorption Processing Block such as error status and processing status.
- (10) Lower-layer software status acquisition
Asks the Protocol Difference Absorption Processing Block for the status data of the Lower-layer Communication Software. The status data requested by this function shall be dynamic information of the Lower-layer Communication Software such as error status and processing status.
- (11) Request for data transmission
Requests that the Protocol Difference Absorption Processing Block send the specified ECHONET data.
- (12) Transmission result acquisition
Asks the Protocol Difference Absorption Processing Block for the transmission result of the data requested by “Request for data transmission”.

(13) Request for transmission stop

Requests that the Protocol Difference Absorption Processing Block stop transmission of the data requested by “Request for data transmission”.

(14) Request for received data

Requests that the Protocol Difference Absorption Processing Block deliver the received data.

(15) NodeID getting

Asks for the NodeID information recognized by the Protocol Difference Absorption Processing Block.

(16) NodeID setting

Sets NodeID information for the Protocol Difference Absorption Processing Block.

Chapter 3 Level 1 ECHONET Common Lower-layer Communication Interface Specification

3.1 List of Level 1 ECHONET Common Lower-layer Communication Interface

Table 3.1 shows a list of level 1 ECHONET Common Lower-layer Communication Interface supported by the Protocol Difference Absorption Processing Block. Mounted APIs conforming to level 1 should be provided with the input/output data items to be specified in the next section. The details of each data item and multiple data items may be implemented as a single data item, or a single data item may be divided into multiple data items. Argument names shall be indicated for reference. The function explanation and input/output items of each API are specified in the next section.

Table 3.1 List of Level 1 ECHONET Common Lower-layer Communication Interface (1/2)

No.	API name	Function outline	Mounting specification
1	Request for Lower-layer Communication Software mounting information	Requests the types and IDs of mounted (accessible) Lower-layer Communication Software .	Required
2	Request for initialization	Requests initialization of Protocol Difference Absorption Processing Block and the Lower-layer Communication Software	Required
3	Request for operation start	Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software operation be started.	Required
4	Fault notice	Notifies Protocol Difference Absorption Processing Block of fault (error) status of high-order layer from the ECHONET communications processing block.	Optional
5	Request for reset	Requests reset processing from Protocol Difference Absorption Processing Block and Lower-layer Communication Software .	Optional
6	Request for suspension	Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software suspend operation.	Optional
7	Request for operation restart	Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software restart operation.	Optional
8	Protocol Difference Absorption Processing Block profile acquisition	Obtains static information for Protocol Difference Absorption Processing Block.	Required
9	Lower-layer Communication Software profile acquisition	Obtains static information for Lower-layer Communication Software .	Required
10	Protocol Difference Absorption Processing Block status acquisition	Obtains dynamic status (processing fault, etc.) of Protocol Difference Absorption Processing Block.	Required

**Table 3.1 List of Level 1 ECHONET Common Lower-layer
Communication Interface (2/2)**

No.	API name	Function outline	Mounting specification
11	Lower-layer Communication Software status acquisition	Obtains dynamic status (processing fault, address redundancy, etc.) of Lower-layer Communication Software .	Required
12	Request for data transmission	Requests data transmission from Protocol Difference Absorption Processing Block.	Required
13	Transmission result acquisition	Requests data transmission result from Protocol Difference Absorption Processing Block.	Optional
14	Request for transmission stop	Requests that data transmission be stopped.	Optional
15	Request for received data	Requests received data from Protocol Difference Absorption Processing Block.	Required
16	NodeID getting	Obtains NodeID recognized by Protocol Difference Absorption Processing Block.	Required
17	NodeID setting	Sets NodeID for Protocol Difference Absorption Processing Block.	Optional

3.2 Level 1 ECHONET Common Lower-layer Communication Interface Detailed Specification

For each API shown in Table 3.1 in the previous section, data input/output is shown below. In the following table, “Input” indicates that data is transferred from the ECHONET communications processing block to the Protocol Difference Absorption Processing Block (input viewed from the Protocol Difference Absorption Processing Block), while “Output” indicates that data is transferred from the Protocol Difference Absorption Processing Block to the ECHONET communications processing block (output viewed from the Protocol Difference Absorption Processing Block). Regarding mounting, the contents of this data should be provided as input/output, but the transfer method (for example, using structures or transferring pointer information for transfer buffer) is not specified for level 2.

- (1) Request for Lower-layer Communication Software mounting information (mandatory function to be mounted)

Requests the type (power line, low-power radio, etc.) of the mounted (accessible)

Lower-layer Communication Software . Table 3.2 shows input/output specifications.

**Table 3.2 List of Low-order Communication Software
Type Request API Input/Output Data**

Direction	Data name	Contents and condition	Remarks
Input	–		
Output	device_num	Specifies number of lower-layer software mounted.	Optional
Output	software_id	Notifies type and ID of Lower-layer Communication Software . Identification of the power line communications software, specific low-power radio communications software, extended HBS communications software, LON communications software, IrDA Control communications software, etc. shall be enabled. If the interface can work with multiple Lower-layer Communication Software , multiple notification shall be enabled.	Required
Output	Return Value	TRUE: Successful initialization, FALSE: Failed initialization.	Optional

- (2) Request for initialization (mandatory function to be mounted)

Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software initialize. Upon receiving this request, the Protocol Difference Absorption Processing Block initializes the Protocol Difference Absorption Processing Block and the specified Lower-layer Communication Software using the specified information. However, normal operation is started when “Request for operation start” is received. Table 3.3 shows input/output specifications.

Table 3.3 List of Initialization Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	software_id	Indicates software to be initialized. Identification of Protocol Difference Absorption Processing Block and all Lower-layer Communication Software shall be enabled.	Required
Input	p_init	Initialization parameter. This data includes the maximum holding time for transmission data and the maximum holding time for received data, etc. However, specific contents differ depending on software to be initialized.	Required
Input	lowif_info	Discrete lower-layer communication IF call function information	Optional
Output	Return Value	TRUE: Successful initialization, FALSE: Failed initialization.	Optional

(3) Request for operation start (mandatory function to be mounted)

Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software start operation. Table 3.4 shows input/output specifications.

Table 3.4 List of Operation Start Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	software_id	Indicates software to start operation There is a request for lump operation start under the Protocol Difference Absorption Processing Block and a request for operation start of individual Lower-layer Communication Software .	Optional
Output	Return Value	TRUE: Successful operation start, FALSE: Failed operation start.	Optional

(4) Fault notice

Notifies Protocol Difference Absorption Processing Block of the fault (error) status of the high-order layer from the ECHONET communications processing block. Table 3.5 shows input/output specifications.

Table 3.5 List of Fault Notice API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	trouble_no	Notice of trouble No.	Required
Output	Return Value	TRUE: Fault notice acceptable, FALSE: Fault notice not acceptable	Optional

(5) Request for reset

Requests reset processing from Protocol Difference Absorption Processing Block and Lower-layer Communication Software . Upon receiving this request, the Protocol Difference Absorption Processing Block executes reset processing for all Lower-layer Communication Software if the request relates to the resetting for the Protocol Difference Absorption Processing Block proper, and waits in the initial status. If the request relates to the resetting for the specified Lower-layer Communication Software , said block executes reset processing only for the specified Lower-layer Communication Software . Table 3.6 shows input/output specifications.

Table 3.6 List of Resetting Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	software_id	Specifies a target to be reset. There is a request for lump resetting under the Protocol Difference Absorption Processing Block and a request for resetting of individual Lower-layer Communication Software .	Required
Output	Return Value	TRUE: Resetting acceptable, FALSE: Resetting not acceptable	Optional

(6) Request for suspension

Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software suspend operation. Upon receiving this request, the Protocol Difference Absorption Processing Block shall not accept any request other than “Request for operation restart” or “Request for reset” from the ECHONET communications processing block if the received request relates to the suspension of the Protocol Difference Absorption Processing Block proper, even when it comes from the ECHONET communications processing block or the Lower-layer Communication Software . If the request relates to the suspension of the discrete Lower-layer Communication Software , said block executes only suspend processing for the specified Lower-layer Communication Software . Table 3.7 shows input/output specifications.

Table 3.7 List of Suspension Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	software_id	Indicates a target for suspension. There is a request for lump suspension under the Protocol Difference Absorption Processing Block and a request for suspension of individual Lower-layer Communication Software .	Required
Output	Return Value	TRUE: Suspension acceptable, FALSE: Suspension not acceptable	Optional

(7) Request for operation restart

Requests that Protocol Difference Absorption Processing Block and Lower-layer Communication Software clear suspension status and restart operation. Upon receiving this request, the Protocol Difference Absorption Processing Block restarts operation of the specified software, including the self-block. Table 3.8 shows input/output specifications.

Table 3.8 List of Operation Restart Request API Data

Direction	Data name	Contents and condition	Remarks
Input	software_id	Indicates software to be restarted. Identification of the Protocol Difference Absorption Processing Block and individual Lower-layer Communication Software shall be enabled.	Required
Output	Return Value	TRUE: Successful restart, FALSE: Restart disable (including failure)	Optional

(8) Protocol Difference Absorption Processing Block profile acquisition (mandatory function to be mounted)

Asks the Protocol Difference Absorption Processing Block for profile data for the Protocol Difference Absorption Processing Block. The data requested by this function shall be static information such as the software developer name and and version No. of the Protocol Difference Absorption Processing Block. Table 3.9 shows input/output specifications.

Table 3.9 List of Protocol Difference Absorption Processing Block Profile Acquisition API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	–		
Output	device_idinfo	Information on the target discrete Lower-layer Communication Software for control. For Protocol Difference Absorption Processing Blocks that can work with multiple Lower-layer Communication Software, this data is for multiple discrete Lower-layer Communication Software. "Multiple" may occur.	Required
Output	version_No	Version information for Protocol Difference Absorption Processing Block	Optional
Output	company_name	Maker information	Optional
Output	rwlen	Buffer size information	Optional
Output	Return Value	TRUE: Normal, FALSE: Error	Optional

- (9) Lower-layer Communication Software profile acquisition (mandatory function to be mounted)

Asks Protocol Difference Absorption Processing Block for Lower-layer Communication Software profile data. The profile data requested by this function shall be static information such as the software developer and version No. of the Lower-layer Communication Software . Table 3.10 shows input/output specifications.

**3.10 Low-order Communication Software Profile
Acquisition API Input/Output Data**

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which profile is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Output	NodeID	NodeID information	Required
Output	version_No	Version information for Lower-layer Communication Software	Optional
Output	company_name	Maker information	Optional
Output	mac_address	Mac address size information	Optional
Output	rwlen	Buffer size information	Optional
Output	broad	Broadcast function available/unavailable	Optional
Output	baud	Transmission rate	Optional
Output	Return Value	TRUE: Normal, FALSE: Error	Optional

- (10) Protocol Difference Absorption Processing Block status acquisition (mandatory function to be mounted)

Asks Protocol Difference Absorption Processing Block for status data of the Protocol Difference Absorption Processing Block. The status data to be requested by this function shall be dynamic data such as error status and processing status. Table 3.11 shows input/output specifications.

**Table 3.11 Protocol Difference Absorption Processing Block
Status Acquisition API Input/Output Data**

Direction	Data name	Contents and condition	Remarks
Input	–		
Output	state	Transition state information. Recognition of the status specified in Part 2 shall be enabled.	Required
Output	trouble_no	Fault information on Protocol Difference Absorption Processing Block.	Optional
Output	upper_trouble	Information recognized as fault in high-order layer	Optional
Output	Low_trouble	Information recognized as fault in low-order layer	Optional
Output	Return Value	TRUE: Normal, FALSE: Error	Optional

- (11) Lower-layer Communication Software status acquisition (mandatory function to be mounted)

Asks Protocol Difference Absorption Processing Block for status data for Lower-layer Communication Software . The status data to be requested by this function shall be dynamic data such as error status and processing status. Table 3.12 shows input/output specifications.

Table 3.12 Status Acquisition API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	software_id	Indicates software for which status is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Output	state	Transition state information. Recognition of stop status, initializing status, completion of initialization, normal processing status, and error stop status shall be enabled.	Required
Output	trouble_no	Fault information for Lower-layer Communication Software	Optional
Output	upper_trouble	Information recognized as fault in high-order layer by Lower-layer Communication Software .	Optional
Output	Return Value	TRUE: Normal, FALSE: Error	Optional

- (12) Request for data transmission (mandatory function to be mounted)

Requests that Protocol Difference Absorption Processing Block transmit the specified ECHONET data. Table 3.13 shows input/output specifications.

Table 3.13 Data Transmission Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which transmission is requested. Recognition of individual Lower-layer Communication Software shall be enabled.	Required
Input	send_data	Transmission of requested data in ECHONET data format	Required
Input	dnode_id	NodeID information of destination in SUBNET and information specifying either broadcast or individual	Required
Output	Return Value	TRUE: Normal, FALSE: Error	Optional

(13) Transmission request acquisition

Asks the Protocol Difference Absorption Processing Block for a transmission result for data requested by “Request for data transmission”. Table 3.14 shows input/output specifications.

Table 3.14 Transmission Result Acquisition API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which transmission result acquisition is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Output	result	Information on transmitting status, normal termination of transmission, termination due to transmission error, or transmission stopping status.	Required
Output	Return Value	TRUE: Normal, FALSE: Error	Optional

(14) Request for transmission stop

Requests that Protocol Difference Absorption Processing Block stop transmission of data requested by “Request for data transmission”. Table 3.15 shows input/output specifications.

Table 3.15 List of Transmission Stop Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which transmission stop is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Output	Return Value	TRUE: Stop success, FALSE: Stop failure (already transmitted)	Optional

(15) Request for data reception (mandatory function to be mounted)

Asks Protocol Difference Absorption Processing Block for received data. Table 3.16 shows input/output specifications.

Table 3.16 List of Data Reception Request API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which received data acquisition is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Output	receive_data	Information on received data	Required
Output	Return Value	TRUE: Normal, FALSE: Error (error indication code, such as no received data)	Optional

(16) NodeID acquisition (mandatory function to be mounted)

Asks for NodeID information recognized by Protocol Difference Absorption Processing Block. Table 3.17 shows input/output specifications.

Table 3.17 List of NodeID Acquisition API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which NodeID acquisition is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Output	nodeID	nodeID information	Required
Output	Return Value	TRUE: Normal, FALSE: Error (error indication code such as unset nodeID and specified device_id error)	Optional

(17) NodeID setting (mandatory function to be mounted)

Sets NodeID for Protocol Difference Absorption Processing Block. Table 3.18 shows input/output specifications.

Table 3.18 NodeID Setting API Input/Output Data

Direction	Data name	Contents and condition	Remarks
Input	device_id	Indicates device for which NodeID setting is requested. Identification of individual Lower-layer Communication Software shall be enabled.	Required
Input	nodeID	NodeID information	Required
Output	Return Value	TRUE: Normal, FALSE: Error (error indication code such as set disable)	Optional

Chapter 4 Level 2 ECHONET Common Lower-layer Communication Interface Specification

This Chapter provides the API detailed specification in consideration of the interchangeability of the software to be developed using this interface as the level 2 ECHONET Common Lower-layer Communication Interface . The specification provided in this Chapter is mainly for cases in which API processing is mounted in the Protocol Difference Absorption Processing Block (a form in which processing of the Protocol Difference Absorption Processing Block is called by the ECHONET communications processing block).

The level 2 ECHONET Common Lower-layer Communication Interface s intended for the ANSI Standard C language (hereafter referred to as C language) are specified in ECHONET Standard V 1.0.

4.1 List of Level 2 ECHONET Common Lower-layer Communication Interface for C Language

The following twenty-three functions are specified as functions of the level 2 ECHONET Common Lower-layer Communication Interface for C language. “Level 2 Optional” may not be mounted as a function. (For example, in the Protocol Difference Absorption Processing Block based on the specification permitting starting the operation of the discrete Lower-layer Communication Software in the same way when the Protocol Difference Absorption Processing Block operation start function is executed, function No.10 may not be mounted.) As for function implementation, the functions shown in this section shall be implemented to conform to level 2.

Table 4.1 List of Level 2 ECHONET Common Lower-layer Communications Interface Functions for C Language

No.	Function name	Name	Remarks
1	ClcGetDevID	Lower-layer Communication Software mounting information request function	Required
2	ClcInit	Protocol Difference Absorption Processing Block initialization function	Required
3	ClcRequestRun	Protocol Difference Absorption Processing Block operation start function	Required
4	ClcSetTrouble	High-order layer fault notice function	Required
5	ClcReset	Protocol Difference Absorption Processing Block operation start function	Required
6	ClcSuspend	Protocol Difference Absorption Processing Block operation suspension function	Required
7	ClcWakeUp	Protocol Difference Absorption Processing Block operation restart function	Required
8	ClcGetProData	Protocol Difference Absorption Processing Block profile data acquisition function	Required
9	ClcGetStatus	Protocol Difference Absorption Processing Block status data acquisition function	Required
10	ClcLowInit	Lower-layer Communication Software initialization function	Optional
11	ClcLowRequestRun	Lower-layer Communication Software operation start function	Optional
12	ClcLowReset	Lower-layer Communication Software operation start function	Optional
13	ClcLowSuspend	Lower-layer Communication Software operation suspension function	Optional
14	ClcLowWakeUp	Lower-layer Communication Software operation restart function	Optional
15	ClcGetLowProData	Lower-layer Communication Software profile data acquisition function	Required
16	ClcGetLowStatus	Lower-layer Communication Software status data acquisition function	Required
17	ClcSendData	Data transmission function	Required
18	ClcGetSendResult	Data transmission result request function	Optional
19	ClcSendCancel	Data transmission stop request function	Optional
20	ClcReceiveData	Data reception function	Required
21	ClcGetNodeID	NodeID acquisition function	Required
22	ClcSetNodeID	NodeID setting function	Required

4.2 C Language-oriented Level 2 ECHONET Common Lower-layer Communication Interface Detailed Specification

This section describes the details of each function shown in Table 4.1 for the following seven items:

- (1) Name
Indicates a function name.
- (2) Function
Explains a function.
- (3) Syntax
Indicates the syntax of the function.
- (4) Explanation
Explains detailed specifications of arguments and variables.
- (5) Return value
Indicates a return value.
- (6) Structure
Indicates the specification of a structure if it exists.
- (7) Notes/restrictions
Indicates precautions and restrictions if they exist.

4.2.1 ClcGetDevID

(1) Name

Lower-layer Communication Software mounting information request

(2) Function

Requests DeviceID information to indicate identification for cases in which there are multiple Lower-layer Communication Software types accessible through the Protocol Difference Absorption Processing Block. This request is based on the assumption that it should be called ahead of “Initialization request function: Clcinit” and “Operation start request: ClcReqeustRun”.

(3) Syntax

```

BOOL ClcInit (
    unsigned char    device_num    /* [OUT] Information on the number of operable
                                   Lower-layer Communication Software */
    unsigned char    *device_idset /* [OUT] Information on operable Lower-layer
                                   Communication Software ID */
)

```

(4) Explanation

*device_idset : Pointer to the ID information on operable Lower-layer Communication Software . Ahead of the pointer, there are pieces of information specified in device_num. The Lower-layer Communication Software ID is as follows:

Power line A system	0x11 ~ 0x1F
Power line B system	0x21 ~ 0x2F
Specific low-power radio	0x31 ~ 0x3F
Extended HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LON	0x61 ~ 0x6F

(5) Return value

0: Failed acquisition
1: Successful acquisition

(6) Structure

None

(7) Notes/restrictions

None

4.2.2 ClcInit

- (1) Name
Protocol Difference Absorption Processing Block initialization request function
- (2) Function
Requests initialization of the Protocol Difference Absorption Processing Block. Upon receiving this request, the Protocol Difference Absorption Processing Block executes its own initialization with the specified information. However, normal operation is started when “Operation start request: ClcRequestRun” is received.
- (3) Syntax

```

BOOL ClcInit (
    CLC_INIT_DATA *init_data      /* [IN] Pointer to initialization parameter */
)

```
- (4) Explanation

```

*init_data      : Pointer to initialization parameter for Protocol Difference Absorption
                  Processing Block

```
- (5) Return value

```

0:   Failed initialization
1:   Successful initialization

```
- (6) Structure

```

typedef struct {
    short      sbuf_len;      /* Transmitting buffer size */
    short      *sbuf;         /* Pointer to transmitting buffer */
    short      rbuf_len;      /* Receiving buffer size */
    short      *rbuf;         /* Pointer to receiving buffer */
    short      sholdtime,     /* Information on maximum holding time for data
                              transmitted by Protocol Difference Absorption
                              Processing Block */
    short      rholdtime,     /* Information on maximum holding time for data
                              received by Protocol Difference Absorption
                              Processing Block */
    unsigned char clc_mode,   /* Specifies operation mode */
} CLC_INIT_DATA

```

```

clc_mode      : Specifies operation mode for the Protocol Difference Absorption
                  Processing Block
                  0x00 Normal operation mode
                  0x01 Test and maintenance mode (details are not specified)

```
- (7) Notes/restrictions
None

4.2.3 ClcRequestRun

(1) Name

Protocol Difference Absorption Processing Block operation start function

(2) Function

Requests that the Protocol Difference Absorption Processing Block start operation. Upon receiving this request, the Protocol Difference Absorption Processing Block starts operation.

(3) Syntax

BOOL ClcRequestRun (void)

(4) Explanation

None

(5) Return value

0: Failed start

1: Successful start

(6) Structure

None

(7) Notes/restrictions

This function must be called before data is transmitted or received after execution of Clcinit. The function must be executed before Initial setting (ClcLowInit) and Operation start (ClcLowRequestRun) of the Lower-layer Communication Software .

4.2.4 ClcSetTrouble

(1) Name

High-order layer fault notice function

(2) Function

Notifies the Protocol Difference Absorption Processing Block of the fault (error) status of the application software and the ECHONET communications processing block.

(3) Syntax

```
BOOL    ClcSetTrouble (  
        char htrouble_no      /* [IN] High-order layer trouble No. /*)
```

(4) Explanation

When a fault is notified, the Protocol Difference Absorption Processing Block performs the following operations:

- Processing upon receipt of data
The block does not execute data receive processing or abandon received data after notifying the Lower-layer Communication Software of the high-order layer operation error.
- Transmission request from the ECHONET communication control processing block
The block rejects any request other than “Request for reset” (ClcReset) and returns an error.

htrouble_no : Trouble No.

-1 Removal of trouble

1 Application software error

2 ECHONET communications processing block error

(5) Return value

0: Failed notice

1: Successful notice

(6) Structure

None

(7) Notes/restrictions

None

4.2.5 ClcReset

(1) Name

Protocol Difference Absorption Processing Block reset request function

(2) Function

Requests resetting of the Protocol Difference Absorption Processing Block. Upon receiving this request, the Protocol Difference Absorption Processing Block makes a request to every Lower-layer Communication Software under control for resetting and executes reset processing for the self-block, waiting in the initial status.

(3) Syntax

BOOL ClcReset (void)

(4) Explanation

The following processing is executed as reset processing upon receipt of this request:

- Clear transmitting/receiving buffer
- Reset fault setting of high-order layer
- Reset various status and work areas
- Reset Lower-layer Communication Software

(5) Return value

0: Failed request
1: Successful request

(6) Structure

None

(7) Notes/restrictions

None

4.2.6 ClcSuspend

(1) Name

Protocol Difference Absorption Processing Block suspension request function

(2) Function

Requests suspension of the Protocol Difference Absorption Processing Block. Upon receiving this request, the Protocol Difference Absorption Processing Block executes suspension of the self-block and the Lower-layer Communication Software and does not accept any request other than “Request for operation restart:ClcWakeup” or “Request for reset:ClcReset” from the ECHONET communications processing block, even when it comes from ECHONET communications processing or Lower-layer Communication Software .

(3) Syntax

BOOL ClcSuspend (void)

(4) Explanation

When this request is received, the series of transmit processing is terminated and operation stop processing is executed if data is being transmitted. If data is being received, the received data are abandoned and the processing is terminated. The following operations are performed in operation stop status:

- Data reception
Not performed.
- Request from ECHONET communication control processing block
Any request other than “Request for operation restart:ClcWakeup” and “Request for reset: ClcReset” is rejected and an error code is returned.

(5) Return value

- 0: Failed suspension
- 1: Successful suspension

(6) Structure

None

(7) Notes/restrictions

None

4.2.7 ClcWakeup

(1) Name

Operation restart function

(2) Function

Requests that suspension status of communications software be cleared under the Protocol Difference Absorption Processing Block and operation restarted. Upon receiving this request, the Protocol Difference Absorption Processing Block clears suspension of self-block and Lower-layer Communication Software using the specified information and restarts operation.

(3) Syntax

BOOL ClcWakeup (void)

(4) Explanation

(5) Return value

0: Failed restart
1: Successful restart

(6) Structure

None

(7) Notes/restrictions

None

4.2.8 ClcGetProData

(1) Name

Protocol Difference Absorption Processing Block profile data acquisition function

(2) Function

Gets profile data for Protocol Difference Absorption Processing Block. Profile data requested by this function is information on property values of profile object of Protocol Difference Absorption Processing Block, such as software developer and version No.

(3) Syntax

```
BOOL    ClcGetProData (
        CLC_PRO_DATA *pro_data,
        /* [OUT] Profile data */)
```

(4) Explanation

pro_data : Profile data of Protocol Difference Absorption Processing Block

(5) Return value

0: Failed acquisition
1: Successful acquisition

(6) Structure

```
typedef struct {
    unsigned char    ver[3];        /* Version No. of Protocol Difference
                                     Absorption Processing Block */
    unsigned char    maker[3];      /* Manufacturer code */
    short            slen;           /* Transmittable data length */
    short            rlen;           /* Receivable data length */
} CLC_PRO_DATA
```

(7) Notes/restrictions

None

4.2.9 ClcGetStatus

(1) Name

Protocol Difference Absorption Processing Block status acquisition function

(2) Function

Asks Protocol Difference Absorption Processing Block for status data. Status data that can be obtained with this function is dynamic information such as error status and processing status.

(3) Syntax

```
BOOL LowGetStatus (
    CLC_STATUS *status /* [OUT] Status of Protocol Difference
                        Absorption Processing Block */)
```

(4) Explanation

status : Status of Protocol Difference Absorption Processing Block is returned.

(5) Return value

0: Failed acquisition
1: Successful acquisition

(6) Structure

```
typedef struct {
    char upper_trouble; /* High-order layer fault code (0 to 127)
                        No fault and removal of trouble (0) */
    char low_trouble; /* Lower-layer Communication Software fault code (0 to
                        127)
                        No fault and removal of trouble (0) */
    char clc_mode; /* Operation mode code
                  Normal operation (0)
                  Test mode, such as maintenance (1)
                  Monitoring mode (2) */
    short state; /* Status of Protocol Difference Absorption Processing
                Block
                CLC_STS_STOP(0) : Stop status
                CLC_STS_INI(1) : Initializing status
                CLC_STS_RUN(2) : Normal processing status
                CLC_STS_ESTOP(3) : Error stop status */
} CLC_STATUS;
```

(7) Notes/restrictions

None

4.2.10 ClcLowInit

(1) Name

Initialization request function

(2) Function

Requests initialization of communications software under Protocol Difference Absorption Processing Block. Upon receiving this request, Protocol Difference Absorption Processing Block initializes self-block and Lower-layer Communication Software using the specified information. However, normal operation is started when “Request for operation start: ClcRequestRun” is received.

(3) Syntax

```

BOOL ClcInit (
    Unsigned char device_id,          /*[IN] Target software type ID for initialization */
    CLC_INIT_DATA *clcinit_data      /*[IN] Pointer to initialization parameter (1) */
    LOW_INIT_DATA *lowinit_data, /*[IN] Pointer to initialization parameter (2) */
    void *low_init                    /*[IN] Pointer to initialization parameter (3) */
)

```

(4) Explanation

device_id : Identification information for target communications software for initialization

Power line A system	0x11 ~ 0x1F
Power line B system	0x21 ~ 0x2F
Specific low-power radio	0x31 ~ 0x3F
Extended HBS	0x41 ~ 0x4F
IrDA_Control	0x51 ~ 0x5F
LON	0x61 ~ 0x6F

*clcinit_data : Pointer to initialization parameter of Protocol Difference Absorption Processing Block

*lowinit_data : Pointer to initialization parameter of Lower-layer Communication Software common specification item

*low_init : Pointer to initialization parameter that differs with individual Lower-layer Communication Software
Contents of parameter are specified for each discrete Lower-layer Communication Software .

(5) Return value

0: Failed initialization

1: Successful initialization

(6) Structure

```
typedef struct {
    short      sbuf_len;    /* Transmitting buffer size */
    short      *sbuf;       /* Pointer to transmitting buffer */
    short      rbuf_len;    /* Receiving buffer size */
    short      *rbuf        /* Pointer to receiving buffer */
    short      sholdtime,   /* Maximum holding time for data transmitted by
                           Protocol Difference Absorption Processing Block */
    short      rholdtime,   /* Maximum holding time for data received by
                           Protocol Difference Absorption Processing Block */
    unsigned char clc_mode, /* Operation mode specification */
} CLC_INIT_DATA
clc_mode      : Operation mode specification of Protocol Difference Absorption
                Processing Block
                0x00    Normal operation mode
                0x01    Test/maintenance mode (details are not specified)

typedef struct {
    short      sfholdtime, /* Maximum holding time for data transmitted by
                           Lower-layer Communication Software */
    short      rfholdtime, /* Maximum holding time for data received by
                           Lower-layer Communication Software */
    unsigned char low_mode, /* Operation mode specification */
    short      mac_len,     /* Mac address length */
    unsigned char mac_ad[7], /* Mac address */
} LOW_INIT_DATA
Except mac_ad[7], set NULL when initialization data is not found.
When NULL is set in mac_len, mac_ad[7] shall have no significance. (When mac_len is set to
NULL, this indicates that there is no Mac address setting.)
```

(7) Notes/restrictions

None

4.2.11 ClcLowRequestRun

(1) Name

Lower-layer Communication Software operation start request function

(2) Function

Requests that Lower-layer Communication Software start operation under control of Protocol Difference Absorption Processing Block. Upon receiving this request, Protocol Difference Absorption Processing Block starts operation of specified Lower-layer Communication Software .

(3) Syntax

```
BOOL ClcLowRequestRun (
    unsigned char device_num    /* [IN] Number information for target
                                Lower-layer Communication Software for
                                initialization */
    unsigned char *device_idset /* [IN] ID information for target Lower-layer
                                Communication Software for initialization */)
```

(4) Explanation

*device_idset : Identification information for target communications software for initialization. When parameter is 0x00, this indicates the start of operation of all Lower-layer Communication Software . For details of other codes, see the ClcLowInit function.

(5) Return value

0: Failed start
1: Successful start

(6) Structure

None

(7) Notes/restrictions

After execution of ClcLowInit, this function must be called before data is transmitted or received.

4.2.12 ClcLowRequest

(1) Name

Lower-layer Communication Software reset request function

(2) Function

Requests resetting of individual Lower-layer Communication Software .

(3) Syntax

```
BOOL ClcReset (  
    unsigned char device_num    /* [IN] Number information for target  
                                communications software to be reset */  
    unsigned char *device_idset /* [IN] ID information for target software to be  
                                reset */)
```

(4) Explanation

*device_idset : Identification information for target communications software to be reset.
For details of codes, see the ClcInit function.

(5) Return value

0: Failed request
1: Successful request

(6) Structure

None

(7) Notes/restrictions

None

4.2.13 ClcLowSuspend

(1) Name

Lower-layer Communication Software suspension request function

(2) Function

Requests suspension of Lower-layer Communication Software . Upon receiving this request, the Protocol Difference Absorption Processing Block suspends the specified Lower-layer Communication Software .

(3) Syntax

```
BOOL ClcSuspend (  
    unsigned char  device_num      /* [IN] Number information for target software to  
                                   be reset */  
    unsigned char  *device_idset   /* [IN] ID information for target software to be  
                                   reset */)
```

(4) Explanation

*device_idset : Identification information on communications software to be suspended.
For details of codes, see the ClnInit function.

(5) Return value

0: Failed suspension
1: Successful suspension

(6) Structure

None

(7) Notes/restrictions

None

4.2.14 ClcLowWakeUp

(1) Name

Lower-layer Communication Software operation restart request function

(2) Function

Requests that Lower-layer Communication Software clear suspension and restart operation. Upon receiving this request, the Protocol Difference Absorption Processing Block clears the suspension status of the specified Lower-layer Communication Software and restarts operation.

(3) Syntax

```
BOOL ClcWakeup (  
    unsigned char device_num    /* [IN] Number information for target software to  
                                be restart */  
    unsigned char *device_idset /* [IN] ID information for target software to be  
                                restart */)
```

(4) Explanation

*device_idset : Identification information for target communications software to be restart. For details of codes, see the ClcInit function.

(5) Return value

0: Failed restart
1: Successful restart

(6) Structure

None

(7) Notes/restrictions

None

4.2.15 ClcGetLowProData

(1) Name

Lower-layer Communication Software profile data acquisition function

(2) Function

Gets profile data and addresses of specific processing functions to be used by Protocol Difference Absorption Processing Block for Lower-layer Communication Software .
The profile data requested by this function is information on property values of the profile object, such as the software developer and version No.

(3) Syntax

```

BOOL ClcLowGetProData (
    unsigned char device_id, /* [IN] Lower-layer Communication Software type ID */
    LOW_PRO_DATA *pro_data,
                                /* [OUT] Profile data */
    short (**chmacfunc) (unsigned char node_id, unsigned char *mac),
                                /* [OUT] NodeID Physical address translation
                                function address */
    unsigned char (**chnodefunc) (unsigned char *mac),
                                /* [OUT] Physical address NodeID translation
                                function address */
    void(**broadfunc) (const char bcast, char map[32])
                                /* [OUT] Broadcast destination acquisition function
                                address */)

```

(4) Explanation

device_id	: Identification information for Lower-layer Communication Software (For specific details of codes, see the ClcLowInit function.)
pro_data	: Profile data for specified Lower-layer Communication Software
chmacfunc	: Translation function address from NodeID to the physical address native to Lower-layer Communication Software is returned. For a medium in which there is a match between NodeID and physical address or for a medium that performs simple linear translation, NULL is returned. Specifications of the arguments of the functions to be delivered are as follows: node_id: [in] NodeID to be translated mac: [out] Physical address that was translated.
chnodefunc	: The translation function address from the physical address native to the Lower-layer Communication Software to NodeID is returned. For a medium in which there is a match between NodeID and physical address or for a medium that performs simple linear translation, NULL is returned. Specifications of the arguments of the functions to be delivered are as follows: mac: [in] Physical address to be translated This function returns nodeID as a return value.
broadfunc	: The broadcast destination acquisition function address is returned.

For Lower-layer Communication Software with a broadcast function, NULL is returned.

Specifications of the arguments of the functions to be delivered are as follows:

bcast	:	[in] Except for 0xff (simultaneous broadcast), observe codes for DEA broadcast.
map	:	[out] Address to bit map of broadcast destination node is returned. Relationship between broadcast destination addresses and bits is shown below.
map[0]-bit0	:	nodeID 0 (0x00)
map[0]-bit1	:	nodeID 1 (0x01)
.....		
map[1]-bit0	:	nodeID 8 (0x08)
map[2]-bit1	:	nodeID 9 (0x09)
.....		
map[31]-bit7	:	nodeID 255 (0xFF)

(5) Return value

0: Failed acquisition
1: Successful acquisition

(6) Structure

```
typedef struct {
    unsigned char    kind;          /* Low-order medium types
                                   Power line A: 0x31      Power line B: 0x32
                                   Low-power radio: 0x33    Extended HBS: 0x34
                                   IrDA Control: 0x35      LON: 0x36 */
    unsigned char    ver[3];        /* Lower-layer Communication Software version No. */
    unsigned char    maker[3];      /* Manufacturer code */
    short            mac_len;        /* MAC address length */
    unsigned char    mac_ad[7];     /* MAC address */
    unsigned char    mac_mask[7];   /* MAC address mask value */
    short            house_len;     /* House code length */
    short            *housecode;     /* Pointer to house code information */
    short            slen;          /* Transmittable data length */
    short            rlen;          /* Receivable data length */
    short            broad;         /* Existence/non-existence of broadcast function
                                   (0: Non-existence, 1: Existence) */
    short            baud;          /* Transmission rate */
} LOW_PRO_DATA
```

(7) Notes/restrictions

None

4.2.16 ClcGetLowStatus

(1) Name

Lower-layer Communication Software status acquisition function

(2) Function

Asks Lower-layer Communication Software for status data for Lower-layer Communication Software. The status data that can be obtained using this function dynamic information, such as error status and processing status.

(3) Syntax

```

BOOL LowGetStatus (
    unsigned char device_id,      /* [IN] Lower-layer software type ID */
    LOW_STATUS *status           /* [OUT] Lower-layer Communication Software
                                status */)

```

(4) Explanation

device_id : Identification information for Lower-layer Communication Software (for specific details of codes, see the ClcLowInit function).
status : Returns status of Lower-layer Communication Software.

(5) Return value

0: Failed acquisition
1: Successful acquisition

(6) Structure

```

typedef struct {
    char upper_trouble; /* High-order layer fault code (0 to 127)
                        No fault or removal of trouble (0) */
    char low_trouble; /* Lower-layer Communication Software block fault code (0 to
                      127)
                      No fault and removal of trouble (0) */
    char low_mode; /* Operation mode code
                  Normal operation (0)
                  Test mode such as maintenance (1)
                  Monitoring mode (2) */
    short state; /* Lower-layer Communication Software block status
                LOW_STS_STOP(0) : Stop status
                LOW_STS_INI(1) : Initializing status
                LOW_STS_RUN(2) : Normal processing status
                LOW_STS_ESTOP(3) : Error stop status */
} LOW_STATUS;

```

(7) Notes/restrictions

None

4.2.17 ClcSendData

(1) Name

Data transmission function

(2) Function

Requests that Protocol Difference Absorption Processing Block transmit specified ECHONET data.

(3) Syntax

```
short ClcSendData (
    unsigned char device_id,    /* [IN] Lower-layer Communication Software type ID
                               */
    const unsigned char *buf,   /* [IN] Pointer to transmission data */
    short snd_sz,              /* [IN] Transmission data size */
    unsigned char dnode_id,    /* [IN] Transmission destination NodeID */
    unsigned char broad,       /* [IN] Broadcast specification */
)
```

(4) Explanation

The transmission data to be delivered in this case is “EHD+SEA+DEA+EBC+EDATA”. When receiving a request for transmission using this function, the Protocol Difference Absorption Processing Block creates data in accordance with the Lower-layer Communication Software specification and transmits the data to the Lower-layer Communication Software .

device_id : Identification information for Lower-layer Communication Software (for specific details of the codes, see the ClcLowInit function).

buf : Specifies a pointer to the transmission data.

snd_sz : Specifies transmission data size.

dnode~id : Specifies transmission destination NodeID.

broad : Specifies broadcast.

0x00 : No broadcast specification

0xFF : Simultaneous broadcast

Regarding others, observe the codes for SEA broadcast.

(5) Return value

CLC_BUFFER_FULL(0)	: Buffer full error
CLC_NO_ERROR(1)	: Transmission accepted
CLC_BUFFER_SIZE_ERROR(2)	: Buffer size error
CLC_STATE_ERROR(3)	: Internal error of Lower-layer Communication Software

(6) Structure

None

(7) Notes/restrictions

None

4.2.18 ClcGetSendResult

(1) Name

Transmission result acquisition function

(2) Function

Obtains transmission result for data requested by “Request for data transmission:ClcSendData”.

(3) Syntax

```
short    ClcGetSendResult (  
          unsigned char device_id,    /* [IN] Lower-layer Communication Software type ID  
                                     */  
          unsigned char *result      /* [OUT] Transmission result */)
```

(4) Explanation

Obtains transmission result for lower-layer communications module.

The result has significance only when the return value is NO_ERROR.

device_id : Identification information for Lower-layer Communication Software (for specific details of the codes, see the ClcLowInit function).

result : Transmission result. 0x00: Successful transmission, 0x01: Failed transmission, 0xFF: No response

(5) Return value

CLC_CANCEL(0)	: Transmission stop
CLC_NO_ERROR(1)	: Normal
CLC_NO_SENDEND(2)	: Transmitting status (transmission not completed)
CLC_INTERNAL_ERROR(3)	: Internal error in Lower-layer Communication Software

(6) Structure

None

(7) Notes/restrictions

None

4.2.19 ClcSendCancel

(1) Name

Transmission stop request function

(2) Function

Requests that transmission of data requested by request for data transmission (ClcSendData) stop

(3) Syntax

```
short    ClcSendCancel (  
          unsigned char device_id    /* [IN] Lower-layer Communication Software type ID  
                                     */)
```

(4) Explanation

Requests that data transmission be stopped. Upon receiving this request, the Protocol Difference Absorption Processing Block asks the specified Lower-layer Communication Software to stop transmission.

device_id : Identification information for Lower-layer Communication Software (for specific details of the codes, see the ClcLowInit function).

(5) Return value

CLC_CANCEL(0)	: No execution of stop processing because of completion of transmission
CLC_NO_ERROR(1)	: Normal
CLC_INTERNAL_ERROR(3)	: Internal error in Lower-layer Communication Software

(6) Structure

None

(7) Notes/restrictions

None

4.2.20 ClcReceiveData

(1) Name

Data receive request function

(2) Function

Asks Protocol Difference Absorption Processing Block for received data.

(3) Syntax

```
short  ClcReceiveData (
        unsigned char device_id,    /* [IN] Lower-layer Communication Software type ID
                                     */
        unsigned char *buf,         /* [OUT] Pointer to receiving buffer */
        short  buf_sz               /* [IN] Receiving buffer size */
        short  *rcv_cz              /* [OUT] Received data size */
        unsigned char *snode_id     /* [OUT] Physical address of transmitting source */
)
```

(4) Explanation

Gets received data from Protocol Difference Absorption Processing Block.

device_id : Identification information for Lower-layer Communication Software (for specific details of the codes, see the ClcLowInit function).
buf : Specifies pointer to receiving buffer.
buf_sz : Specifies receiving buffer size.
rcv_sz : Returns the actual received data size.
snode_id : Returns the NodeID of the transmitting source.

(5) Return value

CLC_NO_RECEIVE(0) : No received data
CLC_NO_ERROR(1) : Normal (with received data)
CLC_BUFFER_SIZE_ERROR(2) : Buffer size error
CLC_INTERNAL_ERROR(3) : Internal error in Lower-layer Communication Software

(6) Structure

None

(7) Notes/restrictions

None

4.2.21 ClcGetNodeID

(1) Name

NodeID information acquisition function

(2) Function

Requests NodeID recognized by Protocol Difference Absorption Processing Block.

(3) Syntax

```
BOOL ClcGetNodeID (  
    unsigned char device_id,    /* [IN] Lower-layer Communication Software type  
                                ID */  
    unsigned char node_id,      /* [OUT] NodeID */)
```

(4) Explanation

A request is made to obtain NodeID corresponding to Lower-layer Communication Software recognized by Protocol Difference Absorption Processing Block.

device_id : Identification information for Lower-layer Communication Software (for specific details of the codes, see the ClcLowInit function).

node_id : NodeID code

(5) Return value

0: Failed NodeID acquisition

1: Successful NodeID acquisition

(6) Structure

None

(7) Notes/restrictions

None

4.2.22 ClcSetNodeID

(1) Name

NodeID information setting function

(2) Function

None

(3) Syntax

```
short    ClcSetSNodeID (  
          unsigned char device_id,    /* [IN] Lower-layer Communication Software type  
                                     ID */  
          unsigned char node_id      /* [IN] NodeID information */)
```

(4) Explanation

Sets the NodeID corresponding to the Lower-layer Communication Software recognized by the Protocol Difference Absorption Processing Block.

device_id : Identification information for Lower-layer Communication Software (for specific details of the codes, see the ClcLowInit function).

node_id : NodeID code

(5) Return value

CLC_NO_CHEANGE(0) : Unchangeable with software

CLC_NO_ERROR(1) : Normal

CLC_INTERNAL_ERROR(3) : Internal error of Lower-layer Communication Software

(6) Structure

None

(7) Notes/restrictions

None