# Part VII
# ECHONET Communications Equipment Specification

Revision record

| | | | |
|---|---|---|---|
| • Version 1.00 | March 18th | 2000 released | Open to consortium members |
| | July | 2000 | Open to the public |
| • Version 1.01 | May 23rd | 2001 | Open to the public<br>Version 1.0 addendum & corrigendum |
| • Version 2.00 | August 7th | 2001 | Open to consortium members<br>Version 1.0 addendum & corrigendum |
| • Version 2.01 | December 19th | 2001 | Open to consortium members.<br>Errors in Version 2.00 were corrected. |
| • Version 2.10 Preview | December 28th | 2001 | Open to consortium members |
| • Version 2.10 Draft | February 15th | 2002 | Open to consortium members<br>The following table-of-contents entries were revised: |

| | Revised entry | Revision/addition |
|---|---|---|
| 1 | 3.6.7, Table 3.6 | - The "protocol difference absorption processing block status acquisition" service was changed to an optional service.<br>- The "stop notice" service and its response "stop notice process response" were added. |
| 2 | 3.6.7 (3) | - The "(19) Stop notice service" was added.<br>- The "(20) Complete initialization request service" was added.<br>- The "(21) Communication stop request service" was added.<br>- The "(22) Stop request service" was added.<br>- The "(5) Reset request service" was renamed to "(5) Start request service".<br>- Explanations of all services were added. |
| 3 | 3.6.2 | - The name of type 4 was changed to "RS-232C/UART type interface". |
| 4 | 3.6.3 (1) (2) (4) | - For type 4, the stipulation about an open-collector type was added. |
| 5 | 3.6.4 (1) | - For type 4, the stipulation about an open-collector type was added. |
| 6 | 3.6.4 (4) | - The type 4 logic description was divided into "RS-232C level" and "open collector" logic descriptions. |
| 7 | 3.6.4 (5) | - The type 4 signal level description was divided into "RS-232C level" and "open collector" signal level descriptions. |
| 8 | 3.6.5 (1) | - A note was added to the description of the type 4 RTS/CTS control method. |
| 9 | 3.6.3 (5) | - For type 4 (open collector), the connector shape description was added. |
| 10 | 3.6.6 | - The reference circuit diagram for type 4 adapter communication interface was added. |

- Version2.10 March 7th 2002 Open to consortium members

  The following table-of-contents entries were revised:

| | Revised entry | Revision/addition |
|---|---|---|
| 1 | 3.6.8 | - The "(5) Reset request service" was renamed "(5) Warm start request service".<br>- "(19) Stop notice service" was added.<br>- "(20) Complete initialization request service" was added.<br>- "(21) Communication stop request service" was added.<br>- "(22) Stop request service" was added.<br>- Explanations of all services were added. |
| 2 | 3.6.7 | - An explanation of lower-layer communication software ID was added. |
| 3 | 3.6.7 | - The number of software programs offering the lower-layer communication software mounting information request service was fixed at 0x01. |
| 4 | 3.6.8 | - Section 3.6.8 was specially added to explain the adapter communication interface service. |

- Version 2.11 April 26th 2002 Open to consortium members

  The following table-of-contents entries were revised:

| | Revised entry | Revision/addition |
|---|---|---|
| 1 | 3.6.2 | - The name of type 4 was changed to “RS232C/UART interface.” |
| 2 | 3.6.5 | - Explanation of service request collision was added. |
| 3 | 3.6.7 (1) | - Explanation of adapter vendor stipulation service was added. |
| 4 | 3.6.7 (1) Fig. 3.25 | - b0: The logical 0 precedence in relation to the collision detection bit was deleted. |
| 5 | 3.6.7 (2) | - Typographical errors relating to range and number of types of the stipulation service code were corrected. |
| 6 | 3.6.7 (2) | - The phrase “(processing at time of service reception is mandatory)” was added to the end of the sentence “Shaded portions of four high-order service code bits of 0 to 3 must be mounted.” |
| 7 | 3.6.7 (2) | - Explanation of service codes 0x11 through 0x14 was corrected. The phrase “for future reserved” was corrected. |
| 8 | 3.6.8 (1) | - An incorrect fixed value for device_num in (Note) was corrected. |
| 9 | 3.6.8 (2) | - “Cold start” was corrected to “cold start (2).” |
| 10 | 3.6.8 (5) | - Explanation of SD (1) through SD (5) was added to the service request frame. An incorrect SC value was corrected. |
| 11 | 3.6.8 (9) | - Incorrect SD (3), SD (5) and SD (9) LF values were corrected. |
| 12 | 3.6.8 (19) | - “SD (1)” was deleted from the service request frame. |
| 13 | 3.6.8 (20) | - “Cold start” was corrected to “cold start (1).” |

# Contents

# Chapter1 Overview of ECHONET Communications Equipment Specification

## 1.1 Basic Concept

@Part 7 states the communications equipment specifications for an ECHONET node, ECHONET device adapter, ECHONET gateway, and ECHONET router. It also stipulates the details and functional definitions of an ECHONET device adapter and ECHONET device interface.

## 1.2 Communications Equipment Specification Overview of ECHONET Node

The term "ECHONET node" is a generic name representing a communication terminal that permits direct information exchange through the ECHONET network. It is used to indicate a communication terminal without identifying its functionality. The requirements for the ECHONET node are stated below:

- ECHONET lower-layer communication software
- ECHONET communication middleware

## 1.3 Communications Equipment Specification Overview of ECHONET Device Adapter

The ECHONET device adapter adds ECHONET node functionality to a device that cannot function as an ECHONET node by itself. In the ECHONET Standard, the ECHONET device adapter is defined as an adapter that connects a device without ECHONET lower-layer communication software and protocol difference absorption processing block to the ECHONET network. Therefore, the requirements for the ECHONET device adapter are as stated below:

- Single ECHONET lower-layer communication software
- Protocol difference absorption processing block
- Adapter communication software

The adapter communication software provides communications between an ECHONET device adapter and ECHONET device. Its specifications are stated in Chapter 3.

## 1.4 Communications Equipment Specification Overview of ECHONET Gateway

The term "ECHONET gateway" refers to an ECHONET node that is capable of connecting an ECHONET domain to an external network. More specifically, the ECHONET gateway is an ECHONET node that has essential functions of a gateway basic block as service middleware.

Note, however, that the ECHONET gateway does not have to be a dedicated ECHONET node having gateway functionality. An ECHONET node having various functions in addition to the gateway functionality can serve as an ECHONET gateway. When the ECHONET gateway is viewed as a communication device, it is not different from an ECHONET node. Therefore, no particular communications equipment specifications are established for the ECHONET gateway.

## 1.5 Communications Equipment Specification Overview of ECHONET Router

The ECHONET router is an ECHONET Node that (1) connects each subnet to be controlled as a range in which ECHONET Lower-Layer Communication Software guarantees the seamless unity of MAC addresses by the ECHONET Communication Middleware protocol and (2) performs data routing processing. Like the ECHONET gateway, the ECHONET router need not always be a dedicated ECHONET Node; it may provide both this function and some other function.

Considering the ECHONET router as communication equipment, its requirements differ depending on the type of ECHONET Lower-Layer Communication Software.

In the case of power lines, low-power radios, extended HBS, and LonTalk®, requirement do not differ from those of ECHONET Nodes. Accordingly, the communication equipment specification of the ECHONET router is not provided separately, and the functional requirements for the ECHONET router are described in Chapter 5.

Considering the IrDA Control as ECHONET Lower-Layer Communication Software, the functional requirements native to IrDA Control must be satisfied for routing. Special functional requirements are described in Chapter 6.

# Chapter2 ECHONET Nodes

## 2.1 Basic Concept

An ECHONET Node is specified as a communication terminal that permits direct information exchange through the ECHONET network. For a device to be recognized as an ECHONET terminal, it must be an ECHONET Node. Accordingly, the full ECHONET device, ECHONET gateway, and ECHONET router are each ECHONET Nodes.

## 2.2 Function Definition

Mandatory functions of an ECHONET Node are described below.

(1) Function to distinguish other ECHONET Nodes from the Self-node

This function is to distinguish the Self-node from other ECHONET Nodes in the same domain and to specify the Self-node on the ECHONET network. Therefore, it must be possible to specify the subnet to which the Self-node belongs using NetID, and to specify the Self-node in the subnet to which the Self-node belongs using NodeID.

(2) Input/output function with transmission media

This function inputs and outputs data via transmission media. Therefore, one or more transceivers capable of handling the ECHONET Lower-Layer Communication Software are functionally required.

(3) Data processing function

This function assembles and disassembles data in each ECHONET communications layer and inputs and outputs data between layers.

Optional functions of an ECHONET Node are described below.

(4) NetID server function

When the ECHONET network consists of multiple subnets, this function assigns a NetID to each subnet with any one ECHONET Node as a master router and distributes this NetID to the ECHONET routers.

To implement this processing, the ECHONET Node must satisfy the following two requirements.

- ECHONET Lower-Layer Communication Software
- ECHONET Communication Middleware

This is shown in Fig. 2.1. In the figure, the shaded portions are requirements.

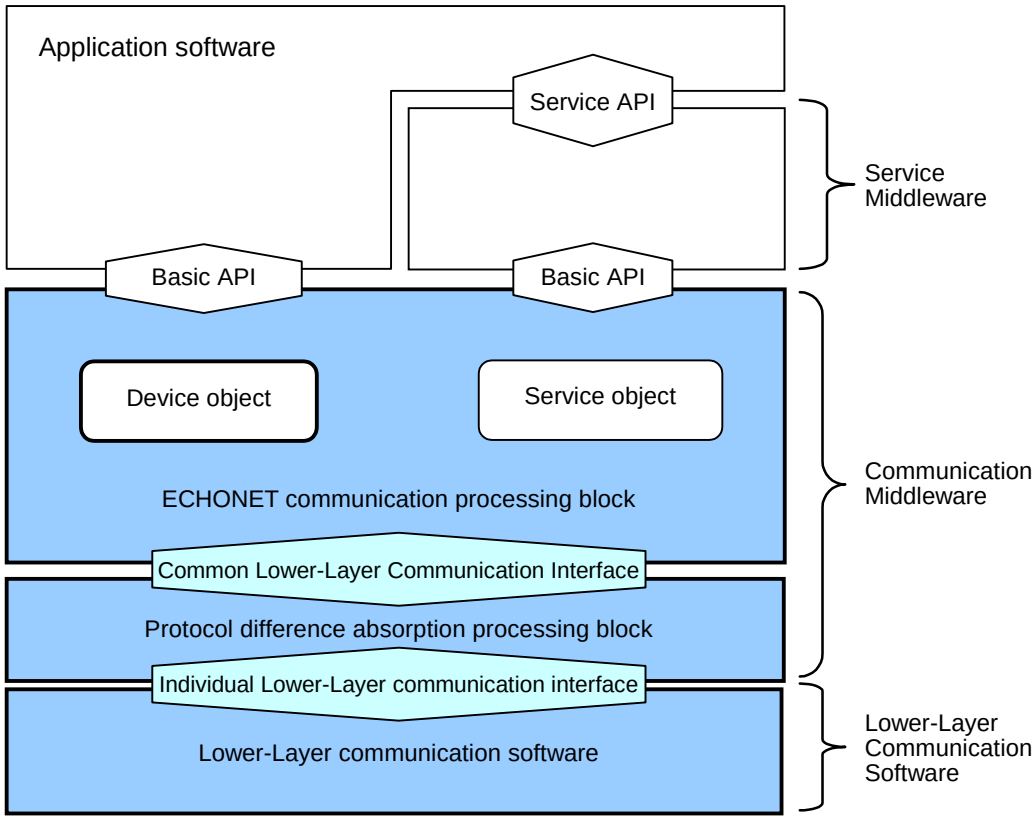



**Fig. 2.1 Requirements for ECHONET Node**

## 2.3 Mechanical and Physical Characteristics

Regarding the specification of connections with transmission media, the specification in Part 3 shall be observed in accordance with Lower-Layer Communication Protocols corresponding to the ECHONET Node. Other mechanical and physical characteristics of an ECHONET Node are specified below.

### 2.3.1 Shape

Regarding the shape, specification shall be provided except for the connection block with the transmission media of the radio system. The shape of this connection block shall conform to the specification of the ECHONET Lower-Layer Communication Software to be used.

### 2.3.2 Display block

When an LED is equipped to display the operation status as ECHONET Node communication equipment, the following minimum requirements must be satisfied. For display methods using means not specified here, the specification native to the product shall be applicable.

- Number of LEDs

  1 (for operation status display)

  • LED color

  Green

  • Status display method

  Normal operation : ON

  Initial processing : Blink (long cycle)

  Error : Blink (short cycle)

  Non-operation : OFF

  * Long cycle ..... Repetition of ON for about 2 sec and OFF for about 0.5 sec

  * Short cycle ..... Repetition of ON for about 0.5 sec and OFF for about 0.5 sec

Note: Initial processing means a cold start (full reset start) and a warm start (hardware executes reset processing while keeping acquired addresses and initial setting information). The initial processing state includes the initialization process state, startup standby state, and suspension state.

## 2.4 NetID Server Function

When the employed configuration contains two or more subnets, the unique NetID server function exists within a domain. The ECHONET node having the NetID server function must have a switch that specifies whether or not to initiate the "Basic Sequence for Parent Router Startup" (Part 2, Section 5.4.1). However, this need not be a mechanical switch. Methods using keyboard instructions or icons will not be specified here.

## 2.5 ECHONET Nodes and subnets

ECHONET Nodes can belong to only one subnet. That is, a single ECHONET Node can have only one ECHONET address. Accordingly, an ECHONET router consists of two or more ECHONET Nodes.

## 2.6 ECHONET Nodes and Domains

ECHONET Nodes cannot belong to multiple domains simultaneously. Accordingly, an ECHONET gateway for connecting two domains is defined as a device having two ECHONET Nodes (see Fig. 2.2).

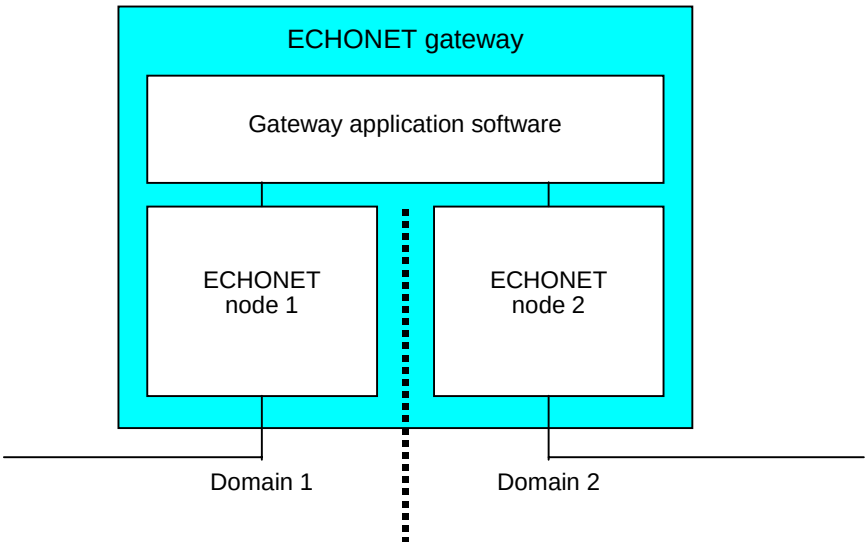



**Fig. 2.2 Relationship between ECHONET Gateway and Nodes**

## 2.7 Limitation on Number of Connections

Assuming that an ECHONET Node is considered as an address specification layer of level 2 subsequent to the subnet, the following limitation is applied:

- The maximum number of ECHONET Nodes that can exist in a single subnet is 256.
  Accordingly, the maximum value of ECHONET Nodes existing in a domain is represented by:

  $n \times m$

  n : Maximum number of subnets in a domain 255

  m : Maximum number of ECHONET Nodes in a subnet 256

  $255 \times 256 = 65{,}280$

However, this value is only the theoretical maximum and may be made smaller by limitations of the ECHONET Lower-Layer Communication Software, etc.

# Chapter3 ECHONET Device Adapter

## 3.1 Basic Concept

In ECHONET, there are two ways of giving a device an ECHONET Node function:

(1) Accommodate the requirements for an ECHONET Node in the device itself.
(2) Add an adapter to a device that is not provided with an ECHONET Node function, thereby satisfying the requirements for the ECHONET Node.

Method (2) can be further divided into two methods based on the interface between the adapter and device:

(2-1) Use an adapter with an interface specified as the ECHONET standard.
(2-2) Use an adapter with a unique interface that is not specified in the ECHONET standard.

In the ECHONET standard, among such adapters shown in (2-1), an adapter to connect a device without ECHONET Lower-Layer Communication Software and a Protocol Difference Absorption Processing Block to the ECHONET network is specified as an ECHONET device adapter. However, only one ECHONET Lower-Layer Communication Software shall be capable of being mounted on one ECHONET device adapter.

The ECHONET device adapter is an adapter that connects a device without ECHONET Lower-Layer Communication Software and a Protocol Difference Absorption Processing Block (flex ECHONET device) to the ECHONET network.

The specification of the interface between a device and an ECHONET device adapter specifies four types: a low-speed system (type 1: 600 bps) and three high-speed systems (type 2, type 3, type 4: 9600 bps). Furthermore, these types are identified by the ECHONET Lower-Layer Communication Software corresponding to the ECHONET device adapter. Accordingly, the ECHONET device adapter is indicated as shown in Fig. 3.2 by the corresponding ECHONET Lower-Layer Communication Software and transmission rate, and this shall be indicated on the ECHONET device adapter as the specification. The following are examples:

(Example 1) When the interface is of type 2 and the ECHONET Lower-Layer Communication Software corresponds to LonTalk®: EAT6-2
(Example 2) When the interface is of type 2 and the ECHONET Lower-Layer Communication Software is an extended HBS: EAT4-4

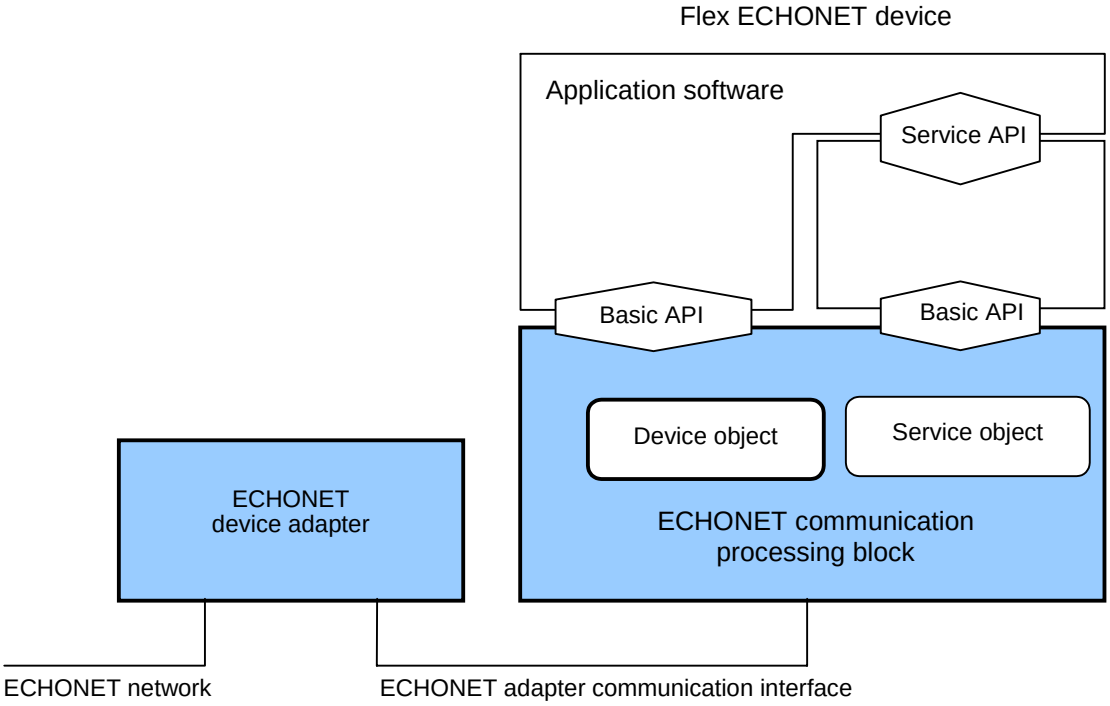



**Fig. 3.1 ECHONET Adapter and Flex ECHONET Device**

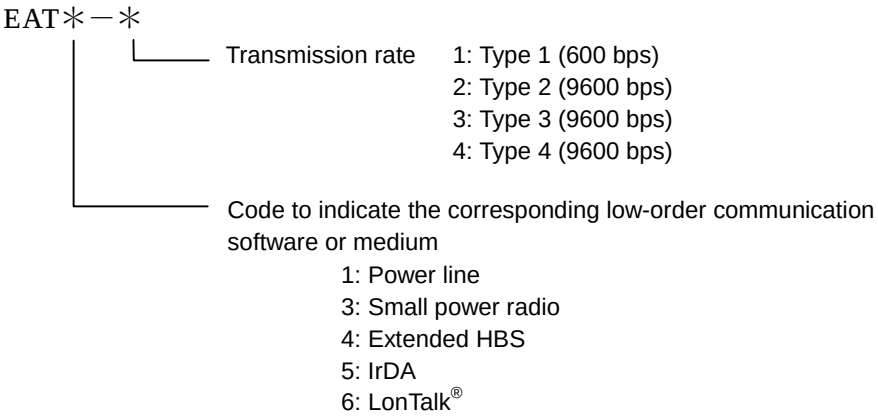



**Fig. 3.2 Type Indication of ECHONET Device Adapter**

## 3.2 Function Definition

The functions required for the ECHONET device adapter are specified as follows:

(1) Input/output function with transmission media

A function to input/output data with transmission media in accordance with the Lower-Layer Communication Protocol specification provided in Part 3. This function is executed by the ECHONET Lower-Layer Communication Software. That is, a single transceiver that can handle the ECHONET Lower-Layer Communication Protocol is required as a function.

(2) Protocol difference absorption function

A function to perform the processing specified in Part 2, Chapter 7 “Protocol Difference Absorption Processing Block Processing Specification”, which performs mutual translation between the ECHONET Lower-Layer Communication Software and the ECHONET Communication Processing Block Protocol. This function is executed by the Protocol Difference Absorption Processing Block.

(3) Adapter communication interface function

A function specified in 3.6 “Adapter Communication Software” in this Section. This function translates the ECHONET Communication Processing Block Protocol input from the Protocol Difference Absorption Processing Block through the common Lower-Layer Communication Interface into an Adapter Communication Interface protocol and then outputs it. In addition, this function outputs the input Adapter Communication Interface protocol into an ECHONET Communication Processing Block Protocol and then outputs it to the common Lower-Layer Communication Interface. This function is processed by the Adapter Communication Software.

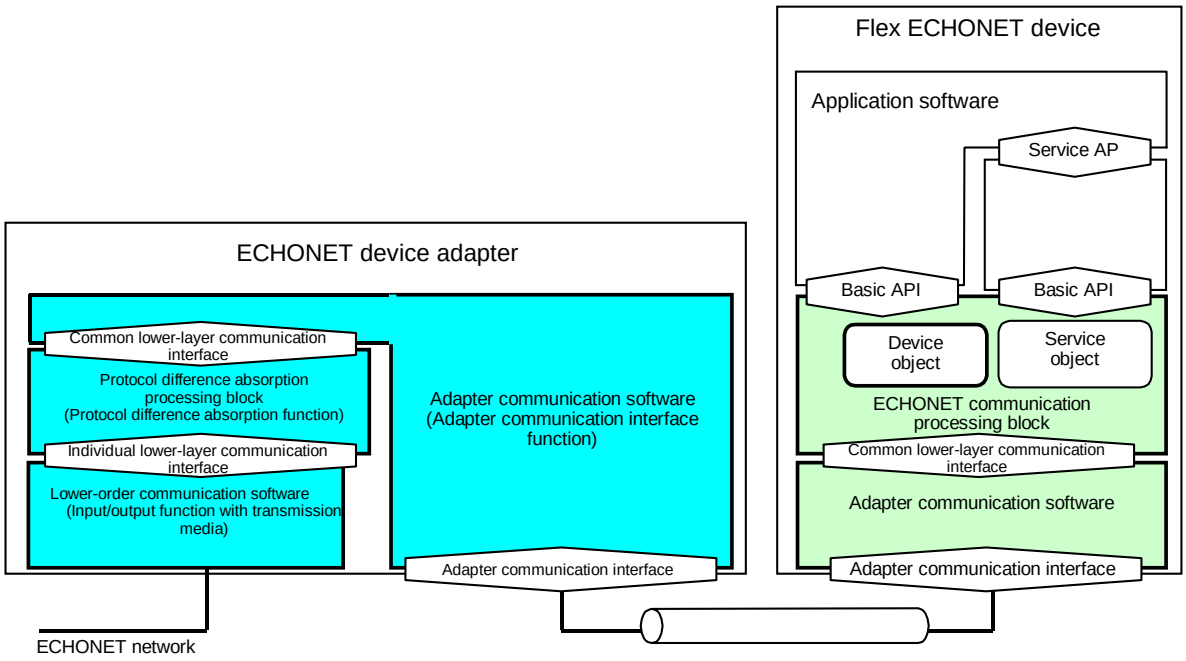



**Fig. 3.3 Functions of ECHONET Device Adapter**

## 3.3 Mechanical and Physical Characteristics

Regarding the specification of connections with transmission media, the specifications in Part 3 shall be observed in accordance with each ECHONET Lower-Layer Communication Software corresponding to the ECHONET device adapter. Specifications for connections with ECHONET devices is provided in Chapter 6. Other mechanical and physical characteristics of ECHONET device adapters are specified below.

### 3.3.1 Shape

Shape shall not be specified except for the connection block with the ECHONET device via the Adapter Communication Interface. The shape of the connection block shall conform to the specification of the ECHONET Lower-Layer Communication Software to be used.

### 3.3.2 Display block

When an LED is provided to display the operation status of the ECHONET device adapter, the following minimum specifications must be satisfied. For a display method using a means not specified here, the specifications native to each product shall be applicable.

- Number of LEDs

  1 (for operation status display)

  • LED color

  Green

  • Status display method

  Normal operation : ON

  Initial processing : Blink (long cycle)

  Error : Blink (Short cycle)

  Non-operation : OFF

  * Long cycle ..... Repetition of ON for about 2 sec and OFF for about 0.5 sec

  * Short cycle ..... Repetition of ON for about 0.5 sec and OFF for about 0.5 sec

Note: Initial processing means a cold start (full reset start) and a warm start (hardware executes reset processing while keeping acquired addresses and initial setting information).

## 3.4 Electrical Characteristics

Regarding the specification of connections with transmission media, the specifications in Part 3 shall be observed in accordance with each ECHONET Lower-Layer Communication Software corresponding to the ECHONET device adapter. Specifications for connections with ECHONET devices are provided in Chapter 6.

## 3.5 Logical Conditions

The logical conditions for Adapter Communication Software are specified in Chapter 6. For the logical conditions related to the ECHONET Lower-Layer Communication Software and Protocol Difference Absorption Processing Block, see Parts 3 and 2, respectively.

## 3.6 Adapter Communication Software

The Adapter Communication Software is defined as follows:

① This software runs on ECHONET device adapters and flex ECHONET devices.

② Differences in the Common Lower-Layer Communication Interface implementation method between the ECHONET device adapter and the flex ECHONET device (e.g. APIs with different specifications) are absorbed.

The Adapter Communication Software handles the Adapter Communication Software Protocol that is an intermediate step of the above translation. The Adapter Communication Interface, Adapter Communication Software Protocol, and their handling are specified below.

### 3.6.1 Overview of Adapter Communication Software

Fig. 3.4 shows data exchange by the Adapter Communication Software between the ECHONET device adapter and a device. Fig. 3.5 shows the relationship between the common lower-layer interface service, Adapter Communication Software Protocol, and Adapter Communication Interface protocol. In addition to services specified as common Lower-Layer Communication Interface services in this standard, the services originally specified by the adapter vendor shall also be handled. Distinctions between different services are made by the service header (SHD).

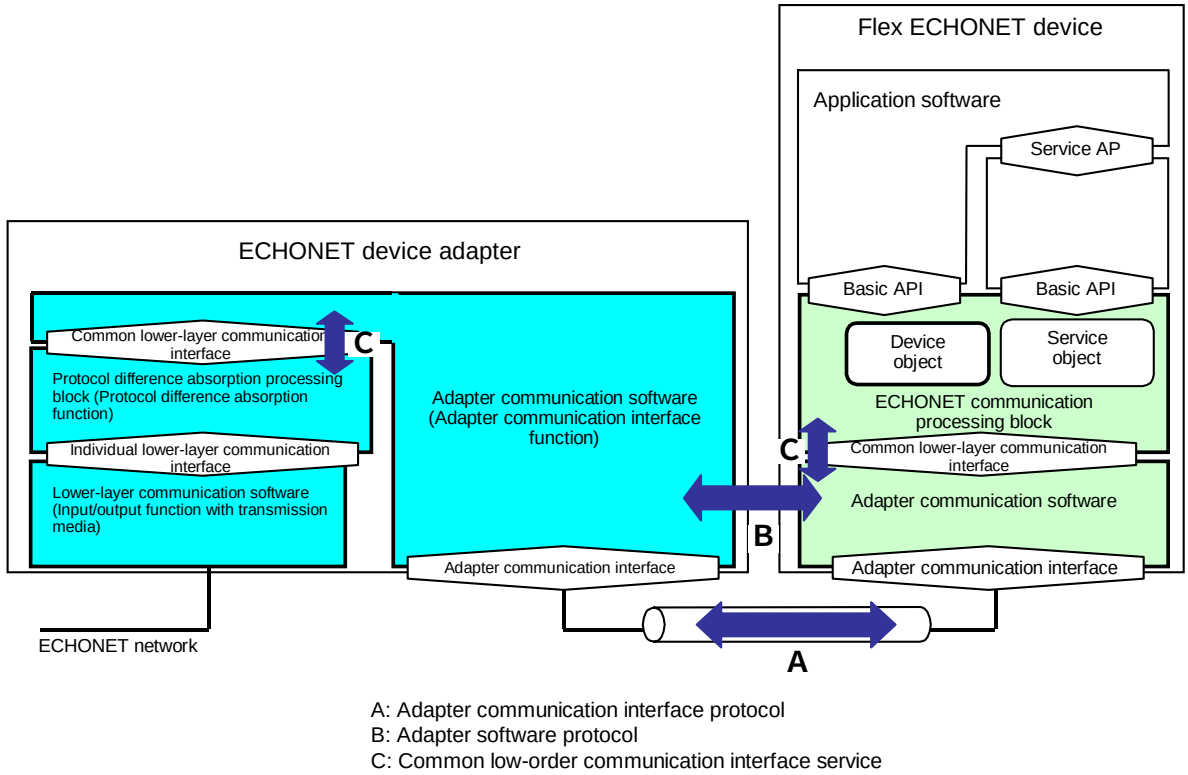



**Fig. 3.4 Data Exchange between ECHONET Device Adapter and Device**

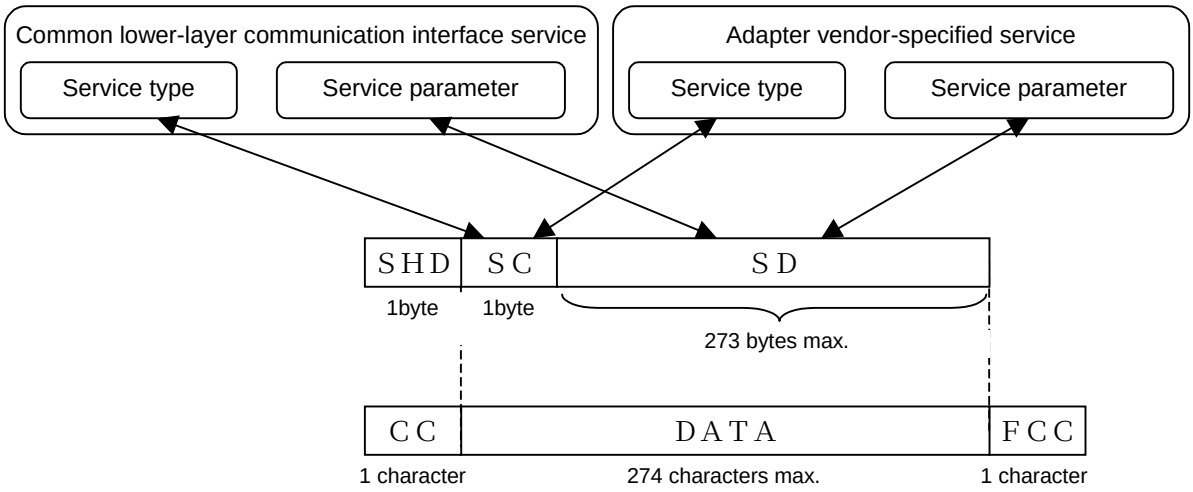



**Fig. 3.5 Relationship between Services and Protocols**

### 3.6.2 Adapter Communication Interface

The Adapter Communication Interface is an interface between the ECHONET device adapter and a device. The four items shown in the tables below are specified.

- Type 1 (Low-speed non-polarity type interface)
- Type 2 (High-speed interface)
- Type 3 (High-speed coaxial interface)
- Type 4 (RS-232C/UART interface)

Type 1 supports a transmission rate of 600 bps, and types 2, 3, and 4 support a transmission rate of 9600 bps.

The type 2 interface permits a flex ECHONET device to supply power to an ECHONET adapter.

### 3.6.3 Mechanical and physical characteristics of adapter communication interface

The mechanical and physical characteristics of the adapter communication interface are stipulated below:

(1) Transmission media

Transmission media using the Adapter Communication Interface are specified in Table 3.1.

**Table 3.1 Transmission Media of Adapter Communication Interface**

| | Medium name | Number of cables | Specification |
|---|---|---|---|
| Type 1 | Twisted pair cable | 1 pair | Conductor diameter 0.65mm |
| Type 2 | Twisted pair cable | 1 pair | Conductor diameter 0.65mm |
| Type 3 | Coaxial cable | 1 cable | S-4C-FB, TVEFCX |
| Type 4 | Conforming to RS-232C | 5 cables | Conforming to RS-232C (RS-232C level) |
| | Multi-conductor cable | 6 cables | The conductor diameter is not stipulated (open collector). |

(2) Cable length

The maximum cable length that can be used for transmission media is specified in Table 3.2.

**Table 3.2 Maximum Cable Length of Transmission Media**

| | Medium name | Maximum cable length |
|---|---|---|
| Type 1 | Twisted pair cable | 30m |
| Type 2 | Twisted pair cable | 200m |
| Type 3 | Coaxial cable | 200m |
| Type 4 | Medium conforming to RS-232C | 15m |
| | Multi-conductor cable | 2m |

(3) Connection form

For all types (1, 2, 3, and 4), one flex ECHONET device is used for one ECHONET device adapter per Adapter Communication Interface. In other words, a one-to-one connection is used.

(4) Connector shape

① Connector on the Flex ECHONET device side

The connector shape is stipulated in Table 3.3.

**Table 3.3 Connector on the Flex ECHONET Device Side**

| | Connector shape | |
|---|---|---|
| Type 1 | 2.5 mm 2-pin metric mutual connection system | |
| Type 2 | Modular type 6-split 2-pin connector | |
| Type 3 | Screw type RCA pin connector | |
| Type 4 | RS-232C level | 9-pin D-SUB male connector (recommended) |
| | Open collector | Original male connector (recommended) |

For type 4 (RS-232C level), the use of a 9-pin D-SUB male connector is recommended and not mandatory. The connector can also be switched to a 9-pin D-SUB male connector outside of a device. For type 4 (open collector), the use of the original male connector shown in Fig. 3.11 is recommended. Connection examples are shown in Fig. 3.6 and Fig. 3.7.

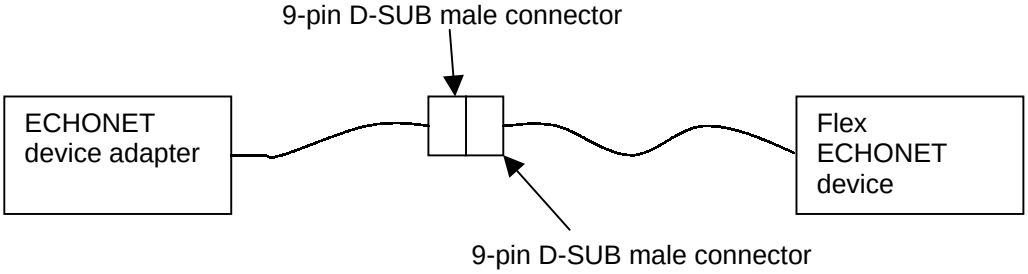
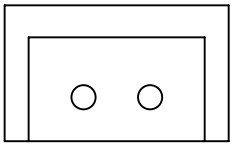



**Fig. 3.6 Device/Adapter Connection Example for Type 4 (RS-232C Level)**


**Fig. 3.7 Device/Adapter Connection Example for Type 4 (Open Collector)**

② Connector on the ECHONET device adapter side
No particularly stipulated.

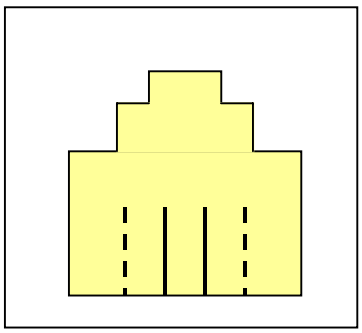
(5) Relationship between connectors and signals
Fig. 3.8 and Fig. 3.9 show the modular connector's jack-end signal assignment and cable color scheme stipulated for type 1 interface and type 2 interface, respectively. Fig. 3.10 shows the type 4 interface pin assignment for the RS-232C level. Fig. 3.11 shows the type 4 interface connector pin assignment for open collector.



| Terminal name | D | D |
|---|---|---|
| Terminal No. | 2 | 1 |

D (Black): signal line

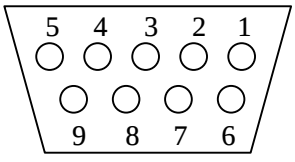
**Fig. 3.8 Type 1 Interface**



| Terminal name | D2 | D1 |
|---|---|---|
| Terminal No. | 2 | 1 |

D1 (Red) : signal and power supply (+)
D2 (Gray) : signal and power supply (–)

**Fig. 3.9 Type 2 Interface**




1：NC
2：RXD
3：TXD
4：NC
5：SG
6：NC
7：RTS
8：CTS
9：NC

**Fig. 3.10 Type 4 Interface (RS-232C level)**

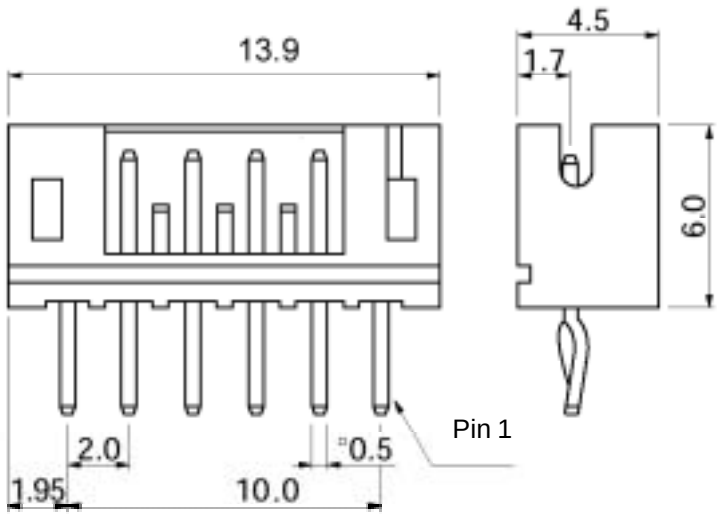



**Fig. 3.11 Type 4 Interface (Open Collector)** reference document (1)

For the type 3 interface (screw type RCA pin connector), the terminals are specified as follows:

D1 (+) : Inner conductor
D2 (–) : Outer conductor

(6) Cable used for type 4 interface

A cross cable shall be used for type 4. The signal correlations between a flex ECHONET device and ECHONET device adapter are indicated in Fig. 3.12.

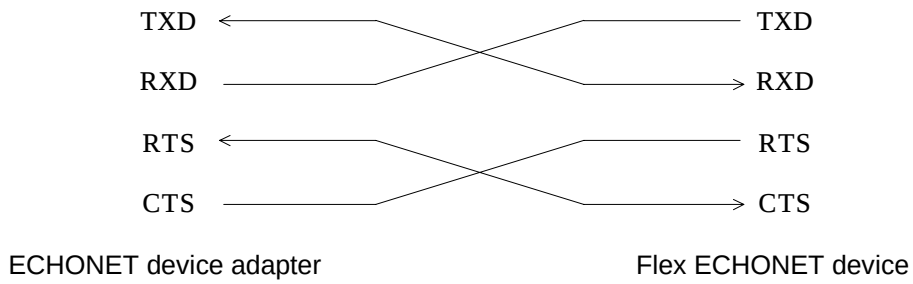



**Fig. 3.12 Type 4 Interface Signal Correlations**

### 3.6.4 Electrical characteristics of Adapter Communication Interface

The electrical characteristics of the Adapter Communication Interface are specified below.

(1) Characteristic cable impedance (nominal value)

The characteristic cable impedance is specified in Table 3.4.

**Table 3.4 Characteristic Cable Impedance**

| | Medium name | Characteristic impedance (nominal value) |
|---|---|---|
| Type 1 | Twisted pair cable | 300Ω |
| Type 2 | Twisted pair cable | 300Ω |
| Type 3 | Coaxial cable | 75Ω |
| Type 4 | Conforming to RS-232C | Not specified |
| | Multi-conductor cable | Not specified |

(2) Load resistance (type 2/type 3 interface)

In type 2 and type 3 interfaces, a load resistance shall be provided on the ECHONET device adapter to suppress waveform distortion. In the type 2 interface, a condenser must be connected in series with the load resistance to cut the direct current in consideration of a power feed (see Fig. 3.13). The load resistance value is shown below.

- Load resistance value R= 39Ω
- DC cut condenser C= 10 to 47μF (type 2 interface only)

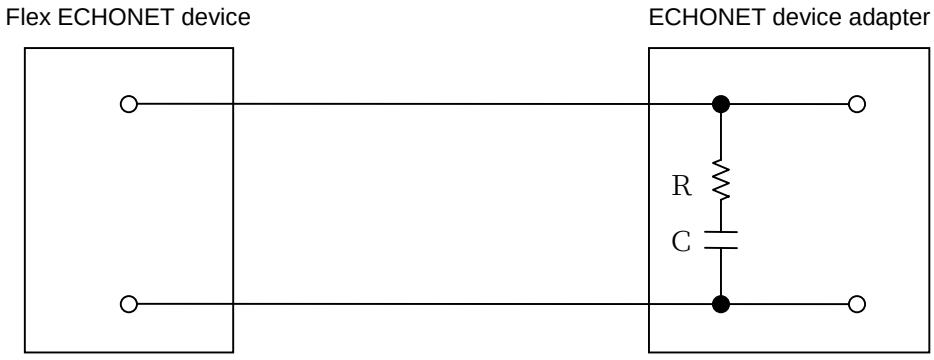



**Fig. 3.13 Load Resistance**

(3) Signal transmission rate

Signal transmission rates are specified as follows:

- Type 1 600 bps ± 1%
- Type 2 9600 bps ± 2.0%
- Type 3 9600 bps ± 2.0%
- Type 4 9600 bps ± 2.0%

(4) Signal transmission system and transmission waveform

The signal transmission system and transmission waveform are specified as follows:

◇ Type 1 and type 4

① Transmission system : Base band transmission

② Transmission waveform : NRZ

③ Logic

Type 1 : Negative logic

Type 4 (RS-232C level)

Data signal line : RS-232C level negative logic, TTL level positive logic

Control line : RS-232C level positive logic, TTL level negative logic

Type 4 (open collector)

Data signal line : Negative logic, TTL level positive logic

Control line : Positive logic, TTL level negative logic

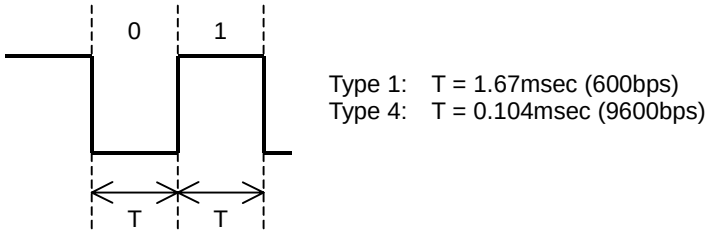



**Fig. 3.14 Signal Transmission System and Transmission Waveform of Type 1/Type 4 Adapter Communication Interface**

◇ Type 2 and type 3

① Transmission system : Base band transmission

② Transmission waveform : AMI (Alternate Mark Inversion)
Duty ratio: 50% $^{+4\%}_{-2\%}$

③ Logic : Negative logic

④ The start bit is sent out from the positive (+) side.

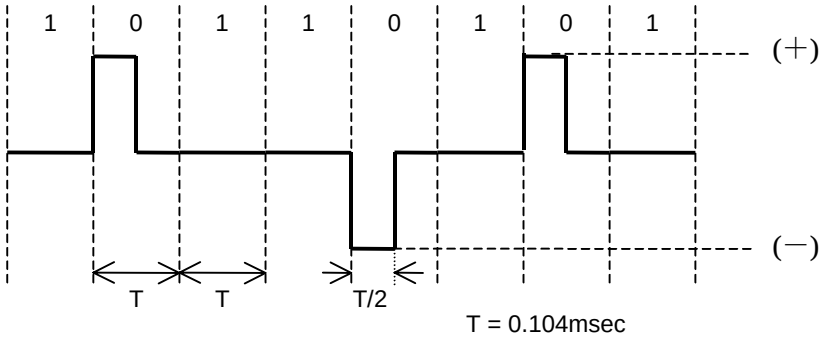



**Fig. 3.15 Signal Transmission System and Transmission Waveform of Type 2/Type 3 Adapter Communication Interface**

(5) Signal transmitting/receiving level

The electrical characteristics of the Adapter Communication Interface are specified as follows:

◇ Type 1

① ECHONET device adapter receiving level

Logic 0 transmission : 4 to 5V
Logic 1 transmission : 2.5 to 3V
Holding : 2.5 to 3V

② Device receiving level

Logic 0 transmission : 1.5 V or less
Logic 1 transmission : 2.5 to 3V
Holding : 2.5 to 3V

* Regarding the transmitting level, a value that satisfies the receiving level shall be specified in consideration of transmission loss, EMI, etc.

◇ Type 2 and type 3

Receiving level

Logic 0 transmission : 1.4 V or less

Logic 1 transmission : 0.6 or less

Holding : 0.6 or less

(Common to both ECHONET device adapter and device)

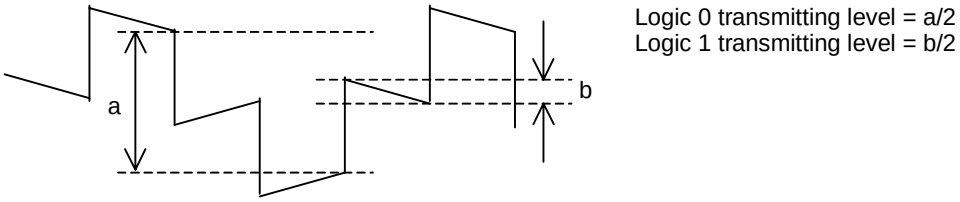



**Fig. 3.16 Signal Transmission System and Transmission Waveform of Type 2 Adapter Communication Interface**

* Regarding the transmitting level, a value that satisfies the receiving level shall be specified in consideration of transmission loss, EMI, etc.

◇ Type 4 (RS-232C level)

Data signal: RS-232C level negative logic, TTL level positive logic

Logic 0 (OFF) transmission : +5V to +15V

Logic 1 (ON) transmission : -5V to -15V

Control signal: RS-232C level positive logic, TTL level negative logic

Logic 0 (OFF) transmission : -5V to -15V

Logic 1 (ON) transmission : +5V to +15V

◇ Type 4 (open collector)

Open collector

Data signal: Negative logic, TTL level positive logic

Control signal: Positive logic, TTL level negative logic

(6) Power feed (type 2/type 4 (open collector) interface only)

For the type 2 interface, a DC current fed to the signal line is permitted. The power feed method is specified in Table 3.5.

**Table 3.5 Power Feed Method for Type 2 Interface**

| | |
|---|---|
| Power feed voltage | 10±1V |
| Receiving power voltage | 7 to 10V |
| Receiving power capacity | 50 mA or less |

Further, the type 2 interface power feed method must satisfy the following requirements:

- The Flex ECHONET device shall be a power feeder, and the ECHONET device adapter shall be a power receiver.
- A protective circuit shall be installed on the power feeder side.
- A reverse-current prevention device shall be installed in the power feeder.

The type 4 (open collector) interface permits the feed of DC power. The power feed method is stipulated as indicated in Table 3.6.

**Table 3.6 Type 4 (Open Collector) Interface Power Feed Method**

| | |
|---|---|
| Power feed voltage | 5V |

Further, type 4 (open collector) must satisfy the following requirements:

- A Flex ECHONET device must be used as a power feeder. An ECHONET device adapter must be used as a power receiver.

### 3.6.5 Logical conditions of Adapter Communication Interface

The logical conditions for the Adapter Communication Interface are specified below.

(1) Control system

◇ Type 1, type 2 and type 3

Device-side preferential survival type CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

◇ Type 4

There are no particular stipulations. However, the following procedure must be followed for RTS/CTS use:

① Check before a transmission that the CTS is OFF.

② When the CTS is OFF, turn ON the RTS, and notify the distant station of a request for transmission.

③ Output data to the TXD.

④ After completion of output, turn OFF the RTS.

Note: Note that the above procedure is different from the procedure for RS-232C RTS/CTS control.

Further, provision must be made so that the device adapter's CTS input is normally OFF in consideration of situations where the flex device does not use RTS/CTS. The one that issued the service request first has precedence. However, in the event of a service request collision, the device adapter's service request has precedence.

(2) Synchronization method

The synchronization is specified as follows:

Synchronization method: Start-stop synchronization

① Character structure (common to types 1 2, 3 and 4)

Start bit : 1 bit
Data : 8 bits
Parity : 1 bit
Stop bit : 1 bit (11 bits in total)

② Start bit transmitting polarity (type 2 and type 3 only)

The start bit transmitting polarity of the type 2 Adapter Communication Interface shall be the positive (+) side.

③ Data transmitting sequence (common to types 1, 2, 3, and 4)

LSB first

④ Start bit (common to types 1, 2, 3, and 4)

Logic 0

⑤ Stop bit (common to types 1, 2, 3, and 4)

Logic 1

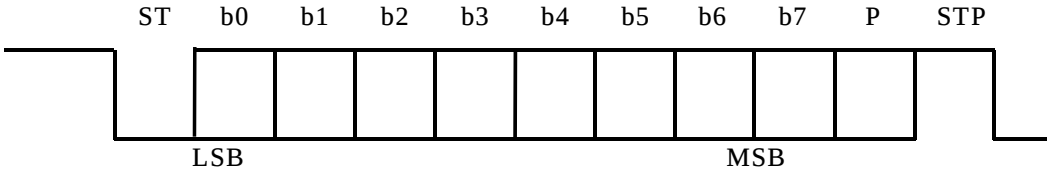
⑥ Parity (common to types 1, 2, 3, and 4)

Even parity

⑦ Character spacing

- Common to types 1, 2, and 3
  There must be no spacing interval between the stop bit and the next character.
- Type 4
  The spacing interval between the stop bit and the next character must not exceed 10 msec.

[Character structure of type 1/type 4 Adapter Communication Interface]




[Character structure of type 2/type 3 Adapter Communication Interface]

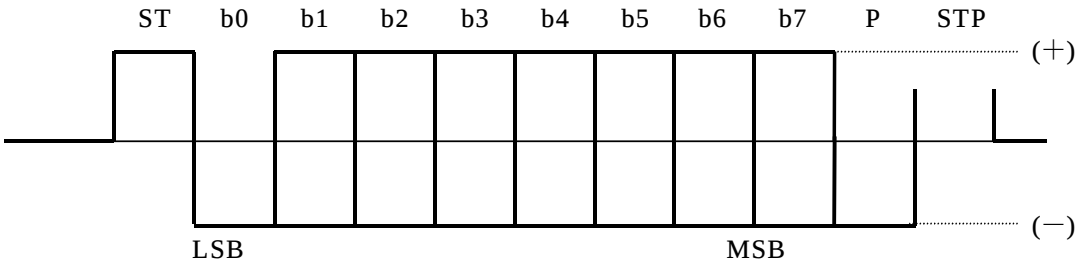
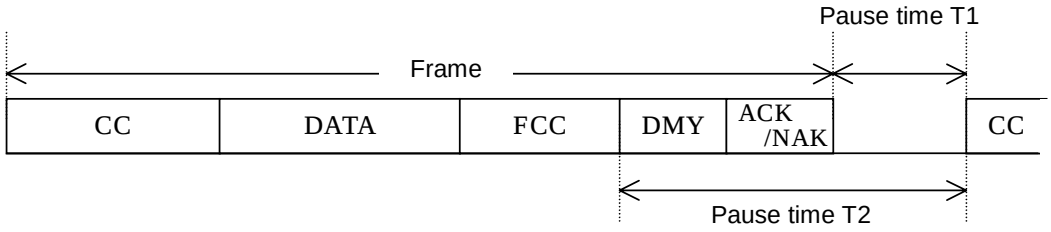



**Fig. 3.17 Character Structure of Adapter Communication Interface**

(3) Basic format of signal

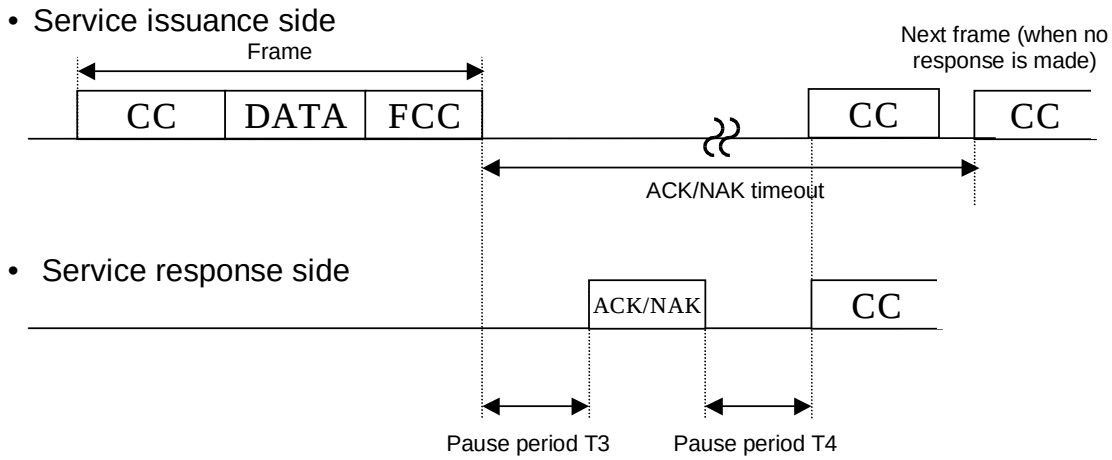
The signal basic format, which is common to types 1, 2, and 3, are stipulated as indicated below:




CC : Control code
DATA : Data field (274 characters max.)
FCC : Check code
DMY : Dummy
ACK/NAK : ACK/NAK code

**Fig. 3.18 (a) Signal Basic Format**

For type 4, the signal basic format is stipulated as indicated below:




CC : Control code
DATA : Data field (274 characters max.)
FCC : Check code
ACK/NAK : ACK/NAK code

**Fig. 3.18 (b) Signal Basic Format (Type 4)**

(4) Pause time and pause period

The pause time and pause periods for types 1, 2, 3, and 4 are stipulated as indicated below:

① Pause time T1

◇ Type 1

Time equivalent to 11 bits from the end of ACK/NAK stop bit

18.37 msec (600 bps)

◇ Type 2, type 3, and type 4

10 msec from the end of ACK/NAK stop bit

② Pause period T2

◇ Type 2 and type 3

Time equivalent to 22 bits plus 10 msec from the end of FCC stop bit

12.29 msec

③ Pause period T3

◇ Type 4

Longer than 10 msec and shorter than 200 msec from the end of FCC stop bit

(10 msec < T3 < 200 msec)

④ Pause period T4

◇ Type 4

Longer than 10 msec from the end of ACK/NAK stop bit

(10 msec < T4)

(5) Control code (CC)

Control code bit assignments, which are common to types 1, 2, 3, and 4, are stipulated as indicated below:

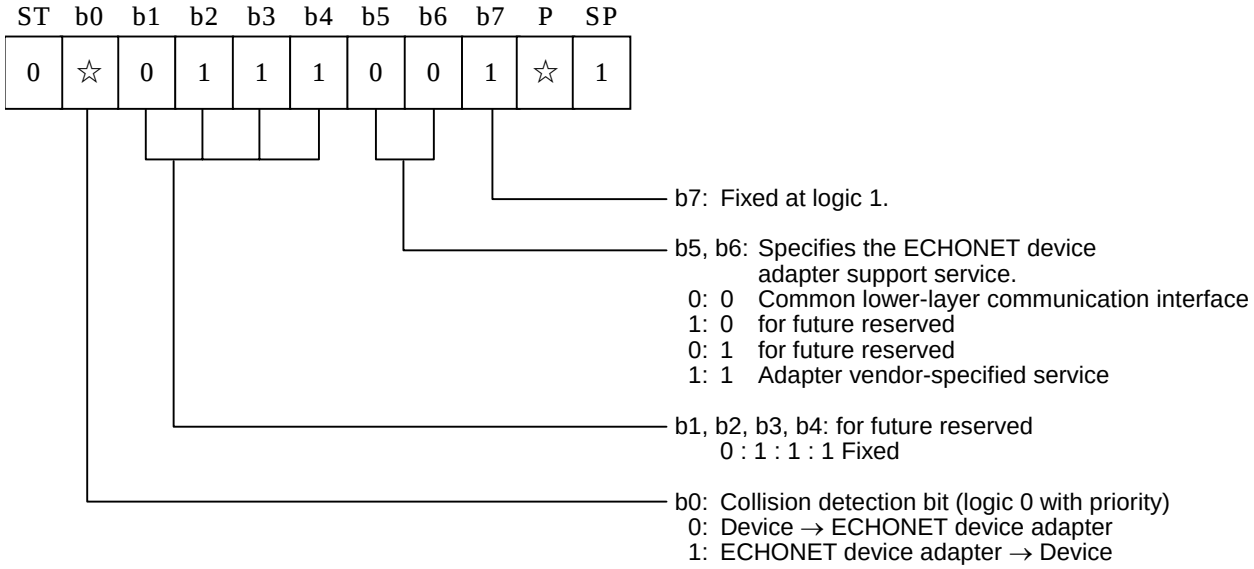



**Fig. 3.19 Control Code**

① Collision detection bit (b0)

For type 1, type 2, and type 3, when packets are sent out from both an ECHONET device adapter and a device, the surviving packet is determined as follows:

- Collision detection bit value

  Device output packet : b0=0

  ECHONET device adapter output packet : b0=1

- Collision detection point

  Point at which a time of 1/4 bit has elapsed after transmission of collision detection bit:

  0.418 msec later (600 bps)

  0.026 msec later (9600 bps)

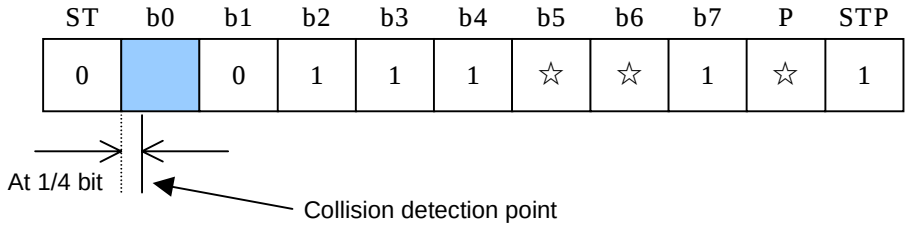



**Fig. 3.20 Collision Detection Point**

② ECHONET device adapter support specification bits (b5, b6)

These bits indicate the type of service to be carried by the packet. The following two services are specified:

- Common lower-layer interface service (b5 = 0, b6 = 0)

  Indicates that the packet is used to exchange the service specified in the common Lower-Layer Communication Interface between the flex ECHONET device and the ECHONET device adapter.

- Adapter vendor-specified service (b5 = 1, b6 = 1)

  Indicates that the packet is specified originally by the adapter vendor.

Regarding b5 = 0, b6 = 1 and b5 = 1, b6 = 0 “reserved for future use” is specified.

(6) Collision detection at contention (type 1, type 2, and type 3)

When packets are sent out simultaneously from both the ECHONET device adapter and the device, resulting in contention, the surviving packet is determined by the following procedure for types 1, 2, and 3.

① At the collision detection point, transmitted data and received data are collated.

② As a result of collation, when a mismatch is detected between the transmitted data and the received data, the transmission is stopped immediately and reception is started. Bit data of logic 0 has priority over bit data of logic 1. (The device side has priority.)

③ The loss-detected side (ECHONET device adapter side) stops transmission and starts reception. When transmission is enabled, this side starts the transmission.

(7) Data division (DATA)

The data division is a device adapter software protocol divided into byte units, to each of which are added a start bit, parity, and stop bit.

(8) Check code (FCC)

The check code shall be a 2's complement of a total of character values existing in the data division for frame transmission error detection.

(9) Packet end detection

If a start bit is not detected after a stop bit is detected, it is considered to be the end of the packet.

(10) Dummy

A dummy is assigned as an error check calculation time, as shown below. During this period, neither packet nor character exists.

◇ Type 1: Time equivalent to 2 bits
3.34 msec (600 bps)

◇ Type 2 and type 3: Time equivalent to 11 bits
1.15 msec (9600 bps)

(11) ACK/NAK

When both the device and ECHONET device adapter receive a signal of 3 characters or more in normal operation mode, error detection is performed for the received signal frame. If the signal is correctly received, the ACK signal is returned. If the signal is not correctly received, the NAK signal is returned. For ACK/NAK, the following characters are assigned:

ACK : 0x06

NAK : 0x15

(12) Error detection and error control

Error detection and error control are provided as indicated below:

- Common to types 1, 2, and 3
  - ① Error detection

    One bit is provided as a parity for each byte, and one byte is provided as a check code for the whole packet to increase the reliability of the received packet. Even parity shall be used.

  - ② Error control

    The error detection result is indicated by the ACK/NAK code subsequent to DMY. When there is no error, the ACK code is returned. When there is an error, the NAK code is returned.
    - Any code other than ACK/NAK is regarded as NAK. Non-response is regarded as NAK.
    - When the packet transmitting side receives NAK after DMY, it resends a packet after the pause time elapses. The maximum number of resend processing times shall be 3.

- Type 4
  - ① Error detection

    One bit is provided as a parity for each byte, and one type is provided as a check code for the whole packet to increase the reliability of the received packet. Even parity shall be used.

② Error control

The error detection result is indicated by the ACK/NAK code. When there is no error, the ACK code is returned. When there is an error, the NAK code is returned.

- All codes other than ACK and NAK are regarded as NAK codes.
- When the packet transmitting side receives NAK, it retransmits a packet after the pause period T4 elapses. It also retransmits a packet if the no-response time exceeds the ACK/NAK timeout time as shown in Fig. 3.18 (b). The ACK/NAK timeout time shall be calculated from the timeout time stipulated in Section 3.6.15.
- The maximum permissible number of retransmissions shall be 3.

### 3.6.6 Adapter communication interface circuit (reference circuit)

Fig. 3.21, Fig. 3.22 and Fig 3.23 show, as an example, a reference circuit to implement the Adapter Communication Interface.

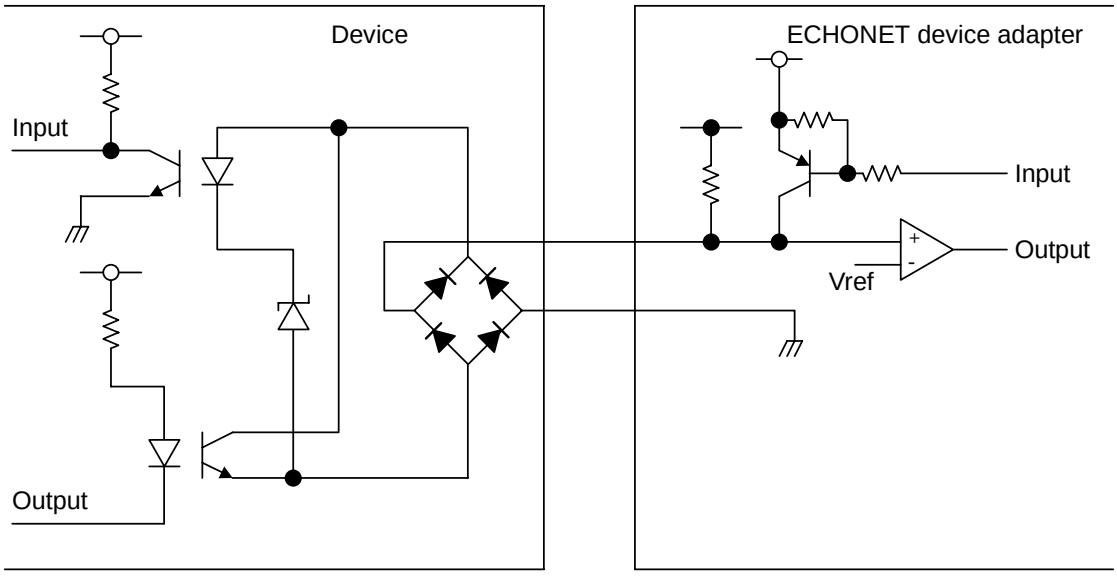
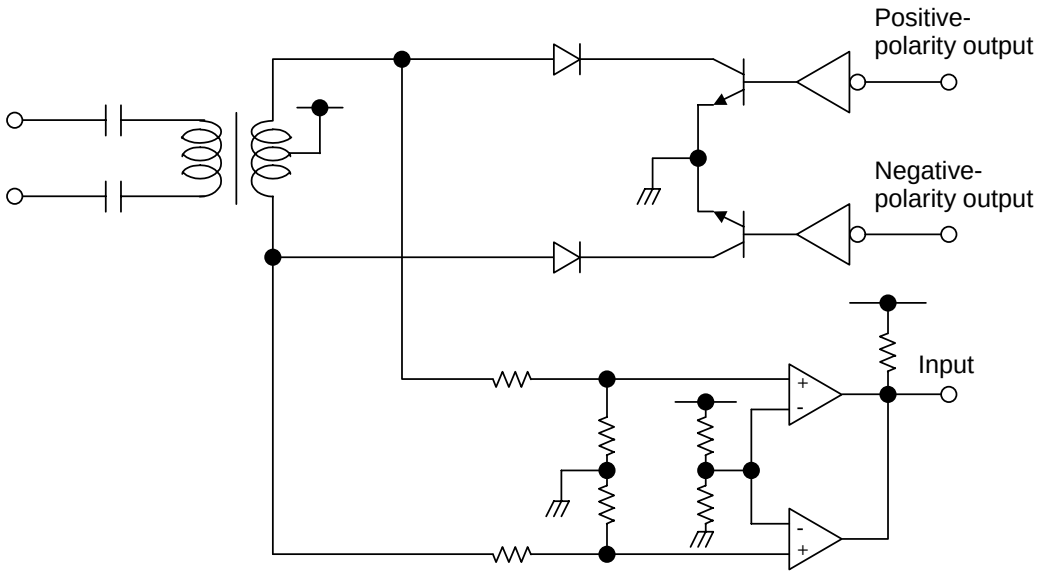



**Fig. 3.21 Example of Type 1 Adapter Communication Interface Circuit**




* Common circuit for both device and ECHONET device adapter

**Fig. 3.22 Type 2/Type 3 Adapter Communication Interface Reference Circuit**

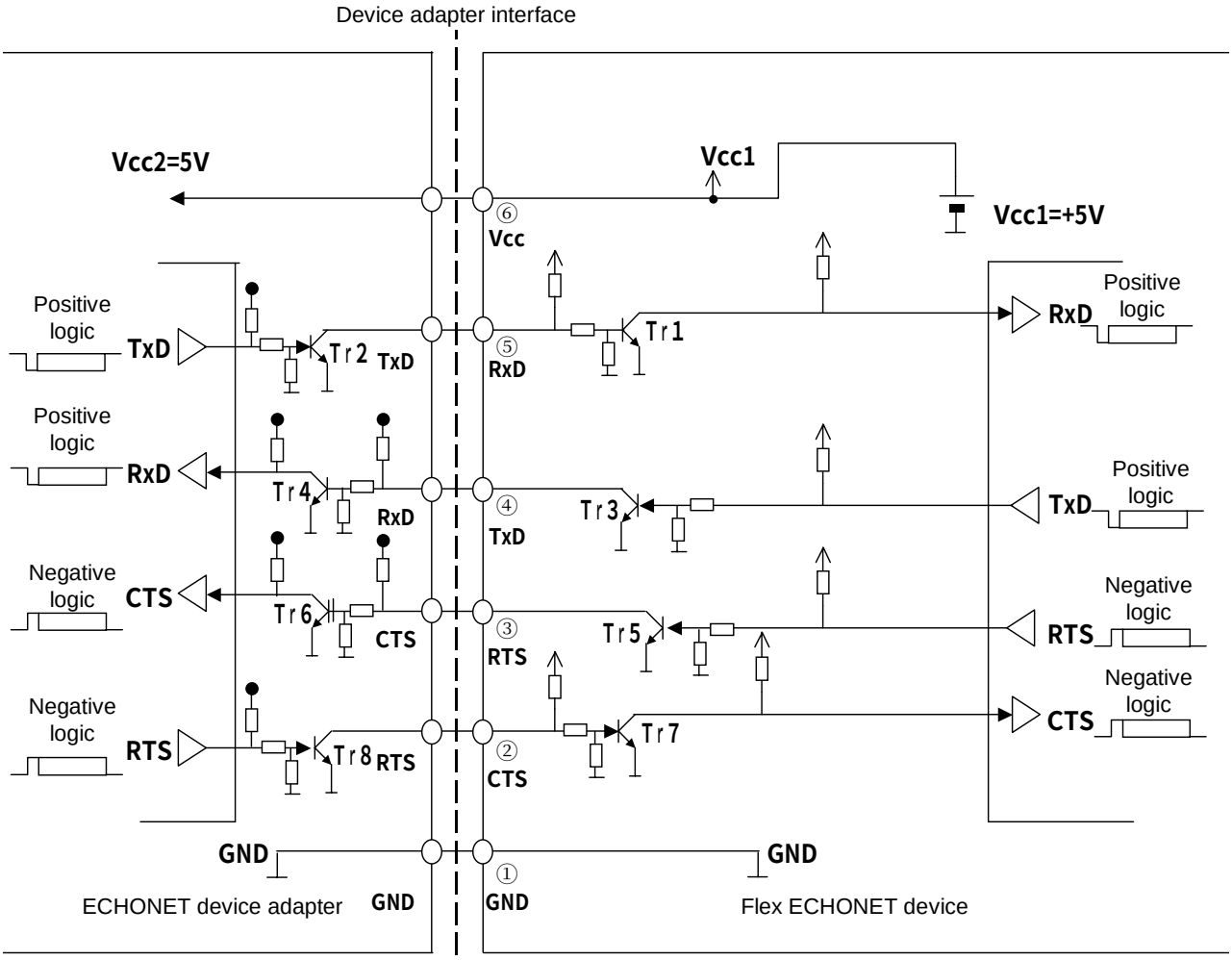



**Fig. 3.23 Reference Circuit for Type 4 Adapter (Open Collector) Interface**

### 3.6.7 Adapter Communication Software Protocol

The Adapter Communication Software Protocol configuration is specified in Fig. 3.24. The service code (SC) and service data (SD) comprise DATA of the Adapter Communication Interface protocol, and the service header (SHD) becomes CC in the Adapter Communication Interface protocol.

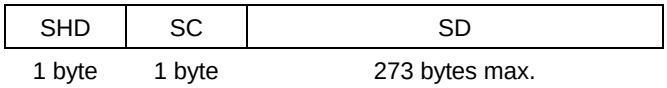



**Fig. 3.24 Adapter Communication Software Protocol**

(1) Service header (SHD)

The service header indicates whether the type of service code (SC) is based on the adapter communication interface service and shows the direction of communication. It is stipulated as indicated in Fig. 3.25. When both b5 and b6 are 1, the adapter vendor may specify its own service code.

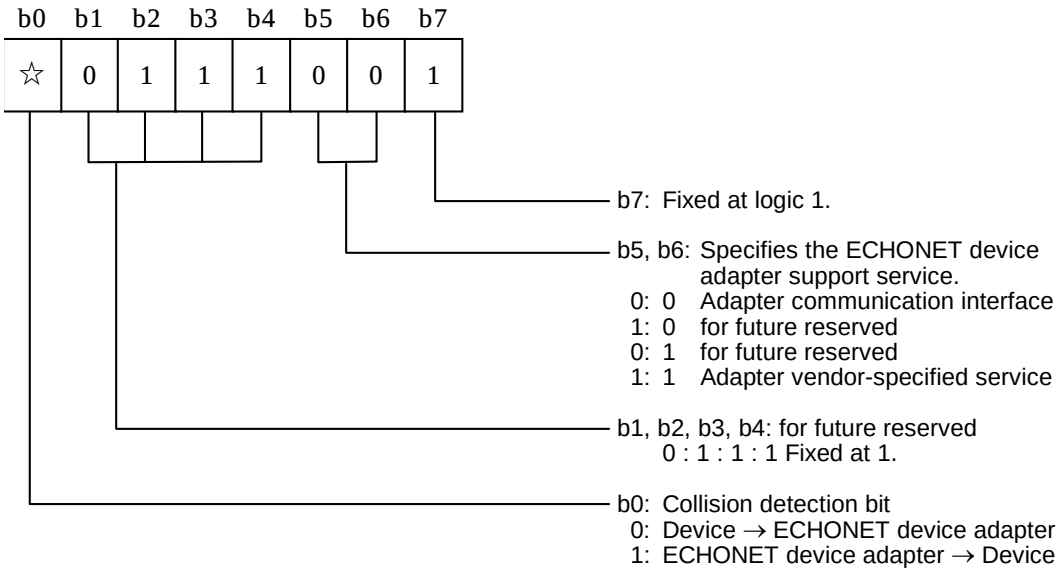



**Fig. 3.25 Service Header**

(2) Service code (SC)

The service code is a 1-byte code that specifies the Adapter Communication Interface or a service (prototype) originally specified by the adapter vendor. In this standard, 21 service codes, 0x00 to 0x14, are specified in Table 3.7 to corresponding to Adapter Communication Interface services. Service codes 0x20 to 0x34 are used to return a response as a result of services 0x00 to 0x14. The service code 0x3f is a response for a service that cannot be processed. As regards an optional adapter communication interface service, it may not always be supported by the processing side even when it is supported by the service requesting source. In such a case, this is notified by a non-processable service notice (0x3f). Shaded portions of four high-order service code bits of 0 to 3 must be mounted (processing at time of service reception is mandatory). Four high-order bits of 0 to 3 having no service code assignments can be reserved for future use. Further, four high-order bits of 4 to F are reserved for future use.

**Table 3.7 Service Codes for Adapter Communication Interface Service (1/2)**

| 4 lower-order bits \ 4 high-order bits | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Request for Lower-Layer Communication Software mounting information | Request for NodeID setting | Processing result of request for Lower-Layer Communication Software mounting information | Response of NodeID setting request processing | | | | | | | | | | | | |
| 1 | Request for initialization | Stop notice | Response of initialize processing | Response of stop notice processing | | | | | | | | | | | | |
| 2 | Request for operation start | Request for complete initialization | Response of operation start processing | Response of complete initialization request processing | | | | | | | | | | | | |
| 3 | Fault notice | Request for communication stop | Response of fault notice processing | Response of communication stop request processing | | | | | | | | | | | | |
| 4 | Request for warm start | Request for complete stop | Response of warm start processing | Response of complete stop request processing | | | | | | | | | | | | |
| 5 | Request for suspension | | Response of suspend processing | | | Reserved for future use | | | | | | | | | | |
| 6 | Request for operation restart | | Response of operation restart processing | | | | | | | | | | | | | |
| 7 | Protocol Difference Absorption Processing Block profile acquisition | | Result of Protocol Difference Absorption Processing Block profile acquisition | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 8 | Lower-layer communication software profile acquisition | | Response of Lower-Layer Communication Software profile get processing | | | | | | | | | | | | | |

**Table 3.7 Service Codes for Adapter Communication Interface Service (2/2)**

| 4 lower-order bits \ 4 high-order bits | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Protocol Difference Absorption Processing Block status acquisition | | Response of Protocol Difference Absorption Processing Block status get processing | | | | | | | | | | | | | |
| A | Lower-layer communication software status acquisition | | Response of Lower-Layer Communication Software status get processing | | | | | | | | | | | | | |
| B | Request for data transmission | | Response of data transmit processing | | | | | | | | | | | | | |
| C | Transmission result acquisition | | Response of transmission result get processing | | | | | | | | | | | | | |
| D | Request for transmission stop | | Response of transmission stop processing | | | | | | | | | | | | | |
| E | Request for received data | | Response of received data processing | | | | Reserved for future use | | | | | | | | | |
| F | NodeID acquisition request | | Response of NodeID acquisition request processing | on-processable service notice | | | | | | | | | | | | |

(3) Service data (SD)

Service data is data for Adapter Communication Interface or for services originally specified by the adapter vendor. For the Adapter Communication Interface, the data corresponds to the argument of API. Each unit of service data consists of two fields, namely, a data length field (LF) and a data field (DF). The data length field is 1 byte long and indicates the number of bytes comprising the data field. The data field is a field for input/output data specified in the common Lower-Layer Communication Interface. A data array of 2 bytes or more is regarded as a big endian. Fig. 3.26 shows a service data structure.

Regarding service data that is optional and not supported, the data field length value shall be 0x00, and the next service data length field shall continue immediately after it.

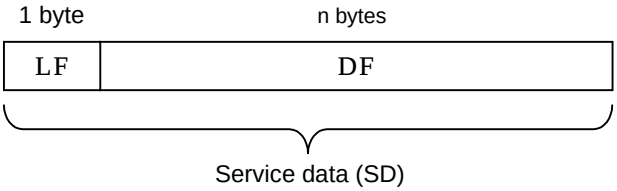



**Fig. 3.26 Service Data Format**

The service data corresponding to each service code is shown together with the Adapter Communication Software Protocol. Shaded service data shall be mandatory and must be mounted. In Fig. 3.26, the number of bytes is for cases in which service data is supported. Service data that is not supported has only 1 byte in the data length field and no data field.

(4) Lower-layer communication software ID (device_id)

The lower-layer communication software ID (device_id) indicates the type of lower-layer communication software mounted in an ECHONET device adapter. Since the ECHONET device adapter can mount only one lower-layer communication software program, it has only one lower-layer communication software ID (device_id). A Flex ECHONET device must use the "lower-layer communication software mounting information request service" to obtain a lower-layer communication software ID (device_id) from an ECHONET device adapter and use the obtained ID whenever the lower-layer communication software ID (device_id) is requested in relation to the subsequent use of various services.

### 3.6.8 Adapter Communication Interface Services

This section details various adapter communication interface services.

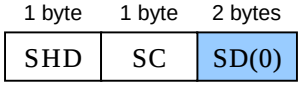
(1) Lower-layer communication software mounting information request service (Required)

Acquires the information about the lower-layer communication software mounted in an ECHONET device adapter (the number of mounted lower-layer communication software programs and their IDs).

◇ Service direction

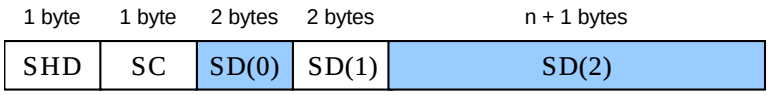
Flex ECHONET device → ECHONET device adapter

◇ Service request




SHD ：0x9C
SC ：0x00
SD(0) ：LF 0x01 DF Dummy (0x88)

◇ Processing result




SHD ：0x9D
SC ：0x20
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01:FALSE)
SD(1) ：LF 0x01 DF Number of mounted Lower-layer communication software (device_num)
SD(2) ：LF n DF Lower-layer communication software ID (device_id)
※ n ＝device_num
device_id

| | |
|---|---|
| 0x11～0x1F | Power line |
| 0x31～0x3F | Low power wireless |
| 0x41～0x4F | Extended HBS |
| 0x51～0x5F | IrDA_Control |
| 0x61～0x6F | LonTalk® |

Note: Since the current version permits an ECHONET device adapter to mount only one lower-layer communication software program, device_num, which is DF of SD(1) , is fixed at 0x01.

(2) Initialization request service (Required)

Requests an ECHONET device adapter to cold start (2) the lower-layer communication software related to a specified lower-layer communication software ID, and place it in the communication stop state. In this case, the MAC address is acquired again.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes | 3 bytes | 3 bytes | 3 bytes | 3 bytes | 2 bytes |
|---|---|---|---|---|---|---|---|
| SHD | SC | SD(0) | SD(1) | SD(2) | SD(3) | SD(4) | SD(5) |

SHD ：0x9C
SC ：0x01
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)
SD(1) ：LF 0x02 DF Transmitting buffer size (in bytes)
SD(2) ：LF 0x02 DF Receiving buffer size (in bytes)
SD(3) ：LF 0x02 DF Maximum holding time for transmission data (msec)
SD(4) ：LF 0x02 DF Maximum holding time for received data (msec)
SD(5) ：LF 0x01 DF Operation mode specification

0x00: Normal operation
0x01: Test/maintenance mode

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x21
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

(3) Operation start request service (Required)

Requests an ECHONET device adapter to place the lower-layer communication software related to a specified lower-layer communication software ID in the normal operation state.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9C
SC ：0x02
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x22
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

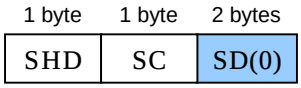
(4) Fault notice service (Optional)

Notifies a device adapter that a trouble occurred in a flex ECHONET device or was cleared.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

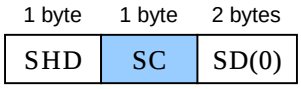
◇ Service request




SHD ：0x9C
SC ：0x03
SD(0) ：LF 0x01 DF Trouble No. (trouble_no)

0x01: An application software abnormality was found.
0x02: The ECHONET communication processing block was abnormal.
0xFF: A trouble was cleared.

◇ Processing result




SHD ：0x9D
SC ：0x23
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

(5) Warm start request service (Required)

Requests an ECHONET device adapter to warm start the lower-layer communication software related to a specified lower-layer communication software ID, and place it in the communication stop state. When this process is performed, the MAC address remains unchanged.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes | 3 bytes | 3 bytes | 3 bytes | 3 bytes | 2 bytes |
|---|---|---|---|---|---|---|---|
| SHD | SC | SD(0) | SD(1) | SD(2) | SD(3) | SD(4) | SD(5) |

SHD ：0x9C
SC ：0x01

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Lower-layer communication software ID (device_id) |
| SD(1) | ：LF | 0x02 | DF | Transmitting buffer size (in bytes) |
| SD(2) | ：LF | 0x02 | DF | Receiving buffer size (in bytes) |
| SD(3) | ：LF | 0x02 | DF | Maximum holding time for transmission data (msec) |
| SD(4) | ：LF | 0x02 | DF | Maximum holding time for received data (msec) |
| SD(5) | ：LF | 0x01 | DF | Operation mode specification |

0x00: Normal operation
0x01: Test/maintenance mode

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x24

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Processing result (0x00：TRUE, 0x01：FALSE) |

(6) Suspension request service (Optional)

Requests a device adapter to place the lower-layer communication software related to a specified lower-layer communication software ID in the suspension state.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9C
SC ：0x05
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x25
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

(7) Operation restart request service (Optional)

Requests an ECHONET device adapter to change the status of the lower-layer communication software related to a specified lower-layer communication software ID from the suspension state to the normal operation state.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9C
SC ：0x06
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x26
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

(8) Protocol Difference Absorption Processing Block profile acquisition service (Required)

Requests an ECHONET device adapter to furnish the profile data about the protocol difference absorption processing block.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9C
SC ：0x07
SD(0) ：LF 0x01 DF Dummy (0x88)

◇ Processing result

| 1 byte | 1 byte | 2 bytes | 4 bytes | 4 bytes | 3 bytes | 3 bytes |
|---|---|---|---|---|---|---|
| SHD | SC | SD(0) | SD(1) | SD(2) | SD(3) | SD(4) |

SHD ：0x9D
SC ：0x27
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)
SD(1) ：LF 0x03 DF Version information about protocol difference absorption processing block
SD(2) ：LF 0x03 DF Manufacturer code
SD(3) ：LF 0x02 DF Number of transmittable data bytes
SD(4) ：LF 0x02 DF Number of receivable data bytes

(9) Lower-layer communication software profile acquisition service (Required)

Requests an ECHONET device adapter to furnish the profile data about the lower-layer communication software related to a specified lower-layer communication software ID.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9C
SC ：0x08
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result

| 1 byte | 1 byte | 2 bytes | 4 bytes | 4 bytes | 2 bytes | 8 bytes | 2 bytes |
|---|---|---|---|---|---|---|---|
| SHD | SC | SD(0) | SD(1) | SD(2) | SD(3) | SD(4) | SD(5) |

| 9 bytes | 3 bytes | 3 bytes | 2 bytes | 3 bytes |
|---|---|---|---|---|
| SD(6) | SD(7) | SD(8) | SD(9) | SD(10) |

SHD ：0x9D
SC ：0x28
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)
SD(1) ：LF 0x03 DF Lower-layer communication software version information
SD(2) ：LF 0x03 DF Manufacturer code
SD(3) ：LF 0x02 DF Mac address bit length
SD(4) ：LF 0x07 DF Mac address information (mac_address)
SD(5) ：LF 0x02 DF House code bit length
SD(6) ：LF 0x08 DF House code information
SD(7) ：LF 0x02 DF Number of transmittable data bytes
SD(8) ：LF 0x02 DF Number of receivable data bytes
SD(9) ：LF 0x02 DF Existence/non-existence of broadcast function (0x00: Nonexistence, 0x01: Existence)
SD(10) ：LF 0x02 DF Transmission rate (bps)

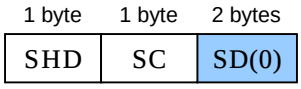
(10) Protocol Difference Absorption Processing Block status acquisition service (Optional)

Requests an ECHONET device adapter to furnish the status information about the protocol difference absorption processing block.

◇ Service direction

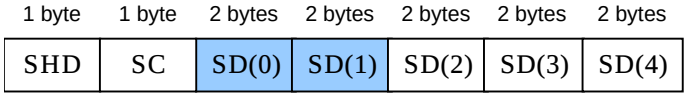
Flex ECHONET device → ECHONET device adapter

◇ Service request




SHD ：0x9C
SC ：0x09
SD(0) ：LF 0x01 DF Dummy (0x88)

◇ Processing result




SHD ：0x9D
SC ：0x29
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)
SD(1) ：LF 0x01 DF Protocol difference absorption processing block transition state information (state)
0: Stop state
1: Initializing state
2: Normal operation state
3: Error stop state
SD(2) ：LF 0x01 DF Protocol high-order layer fault (upper_trouble)
SD(3) ：LF 0x01 DF Protocol difference absorption processing block fault (low_trouble)
SD(4) ：LF 0x01 DF Protocol difference absorption processing block operation mode (low_mode)
0: Normal operation mode
1: Test mode
2: Monitoring mode

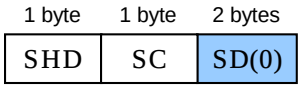
(11) Lower-layer communication software status acquisition service (Required)

Requests an ECHONET device adapter to furnish the status information about the lower-layer communication software related to a specified lower-layer communication software ID.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

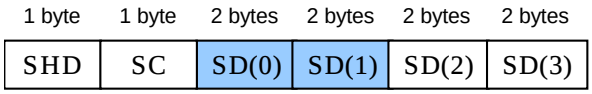
◇ Service request




SHD ：0x9C
SC ：0x0A

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Lower-layer communication software ID (device_id) |

◇ Processing result




SHD ：0x9D
SC ：0x2A

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Processing result (0x00：TRUE, 0x01：FALSE) |
| SD(1) | ：LF | 0x01 | DF | Lower-layer communication software transition state information (state)<br>0: Stop state<br>1: Initializing state<br>2: Normal operation state<br>3: Error stop state |
| SD(2) | ：LF | 0x01 | DF | Lower-layer communication software fault (low_trouble) |
| SD(3) | ：LF | 0x01 | DF | Protocol difference absorption processing block operation mode (low_mode)<br>0: Normal operation mode<br>1: Test mode<br>2: Monitoring mode |

(12) Data transmission request service (Required)

Requests an ECHONET device adapter to transmit data with the lower-layer communication software related to a specified lower-layer communication software ID.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes | 2 bytes | 2 bytes | 3 bytes | |
|---|---|---|---|---|---|---|
| SHD | SC | SD(0) | SD(1) | SD(2) | SD(3) | SD(4) |

SHD ：0x9C
SC ：0x0B

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Lower-layer communication software ID (device_id) |
| SD(1) | ：LF | 0x01 | DF | Transmitting destination NodeID information |
| SD(2) | ：LF | 0x01 | DF | Broadcast specification information<br>0x00: Does not specify a broadcast.<br>0xFF: Specifies a broadcast. |
| SD(3) | ：LF | 0x02 | DF | Number of transmittable data bytes (data_len) |
| SD(4) | ：LF | n | DF | Transmission data (send_data)<br>n = data_len (0xFF for 255 bytes or more) |

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x2B

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Processing result<br>(0x00: Buffer full,<br>0x01: Transmission acceptable,<br>0x02: Buffer size error,<br>0x03: Low-order communication software error,<br>0x04: Processing failure.) |

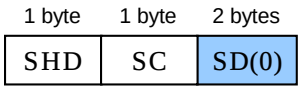
(13) Transmission result acquisition service (Optional)

Requests an ECHONET device adapter to furnish the status information about a data transmission request process that was performed by the lower-layer communication software related to a lower-layer communication software ID specified immediately before this request.

◇ Service direction

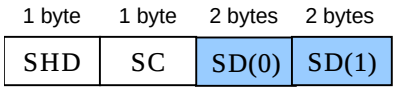
Flex ECHONET device → ECHONET device adapter

◇ Service request




SHD ：0x9C
SC ：0x0D
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result




SHD ：0x9D
SC ：0x2C
SD(0) ：LF 0x01 DF Processing result
(0x00: Transmission stop,
0x01: Normal,
0x02: Transmitting status,
0x03: Lower-layer communication software error)
SD(1) ：LF 0x01 DF Transmission result
(0x00: Success in transmission,
0x01: Failure in transmission,
0xFF: No response,
0x04: Processing failure.)

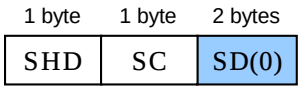
(14) Transmission stop request service (Optional)

Requests an ECHONET device adapter to stop the ongoing data transmission process performed by the lower-layer communication software related to a specified lower-layer communication software ID in compliance with a data transmission request that was issued immediately before this request.

◇ Service direction

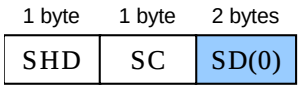
Flex ECHONET device → ECHONET device adapter

◇ Service request




SHD ：0x9C
SC ：0x0D
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result




SHD ：0x9D
SC ：0x2D
SD(0) ：LF 0x01 DF Processing result
(0x00: Termination of transmission,
0x01: Normal,
0x03: Lower-layer communication software error,
0x04: Processing failure.)

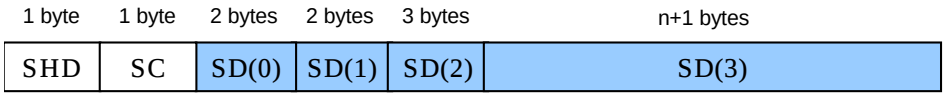
(15) Data reception request service (Required)

Notifies a Flex ECHONET device of data received by the lower-layer communication software related to a specified lower-layer communication software ID.

◇ Service direction

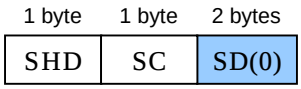
ECHONET device adapter→ Flex ECHONET device

◇ Service request




| | | | | |
|---|---|---|---|---|
| SHD | ：0x9D | | | |
| SC | ：0x0E | | | |
| SD(0) | ：LF | 0x01 | DF | Lower-layer communication software ID (device_id) |
| SD(1) | ：LF | 0x01 | DF | Transmitting source NodeID |
| SD(2) | ：LF | 0x02 | DF | Number of received data bytes (data_len) |
| SD(3) | ：LF | n | DF | Received data (receive_data)<br>n = data_len (0xFF for 255 bytes or more) |

◇ Processing result




| | | | | |
|---|---|---|---|---|
| SHD | ：0x9C | | | |
| SC | ：0x2E | | | |
| SD(0) | ：LF | 0x01 | DF | Processing result (0x00：TRUE, 0x01：FALSE) |

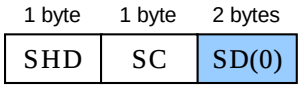
(16) NodeID acquisition service (Required)

Requests an ECHONET device adapter to furnish a node ID that is based on the MAC address currently retained by the lower-layer communication software related to a specified lower-layer communication software ID.

◇ Service direction

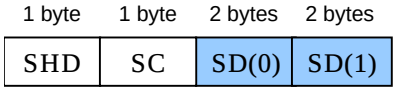
Flex ECHONET device → ECHONET device adapter

◇ Service request




SHD ：0x9C
SC ：0x0F
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result




SHD ：0x9C
SC ：0x2F
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)
SD(1) ：LF 0x01 DF NodeID

(17) Node ID setup request service (Optional)

Requests an ECHONET device adapter to set a node ID for the lower-layer communication software related to a specified lower-layer communication software ID.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 byte | 2 byte |
|---|---|---|---|
| SHD | SC | SD(0) | SD(1) |

SHD ：0x9C
SC ：0x10
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)
SD(1) ：LF 0x01 DF NodeID

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x30
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

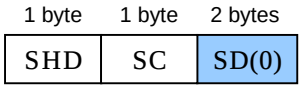
(18) Non-processable service notice (Required)

Notifies that there is a non-processable service.

◇ Service direction

Flex ECHONET device ⇔ ECHONET device adapter

◇ Notice format




SHD ：0x9C or 0x9D
SC ：0x3F
SD(0) ：LF 0x01 DF Requested service code

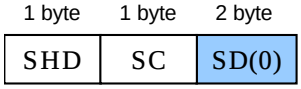
(19) Stop notice service (Required)

When the lower-layer communication software related to a specified lower-layer communication software ID is in the stop state, this service notifies a Flex ECHONET device that the lower-layer communication software is in the stop state.

◇ Service direction

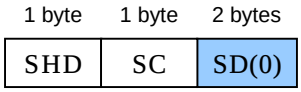
ECHONET device adapter → Flex ECHONET device

◇ Service request




SHD ：0x9D
SC ：0x11
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result




SHD ：0x9D
SC ：0x31
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

(20) Complete initialization request service (Optional)

Requests an ECHONET device adapter to cold start (1) the lower-layer communication software related to a specified lower-layer communication software ID and place it in the communication stop state. In this case, the house code information and MAC address are acquired again.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes | 3 bytes | 3 bytes | 3 bytes | 3 bytes | 2 bytes |
|---|---|---|---|---|---|---|---|
| SHD | SC | SD(0) | SD(1) | SD(2) | SD(3) | SD(3) | SD(5) |

SHD ：0x9C
SC ：0x12

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Lower-layer communication software ID (device_id) |
| SD(1) | ：LF | 0x02 | DF | Transmitting buffer size (in bytes) |
| SD(2) | ：LF | 0x02 | DF | Receiving buffer size (in bytes) |
| SD(3) | ：LF | 0x02 | DF | Maximum holding time for transmission data (msec) |
| SD(4) | ：LF | 0x02 | DF | Maximum holding time for received data (msec) |
| SD(5) | ：LF | 0x01 | DF | Operation mode specification |

0x00: Normal operation
0x01: Test/maintenance mode

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x32

| | | | | |
|---|---|---|---|---|
| SD(0) | ：LF | 0x01 | DF | Processing result (0x00：TRUE, 0x01：FALSE) |

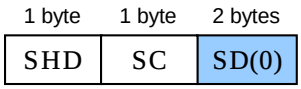
(21) Communication stop request service (Optional)

Requests an ECHONET device adapter to place the lower-layer communication software related to a specified lower-layer communication software ID in the communication stop state.

◇ Service direction

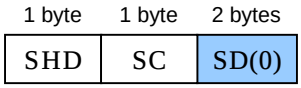
Flex ECHONET device → ECHONET device adapter

◇ Service request




SHD ：0x9C
SC ：0x13
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result




SHD ：0x9D
SC ：0x33
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

(22) Complete stop request service (Optional)

Requests an ECHONET device adapter to place the lower-layer communication software related to a specified lower-layer communication software ID in the stop state.

◇ Service direction

Flex ECHONET device → ECHONET device adapter

◇ Service request

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9C
SC ：0x14
SD(0) ：LF 0x01 DF Lower-layer communication software ID (device_id)

◇ Processing result

| 1 byte | 1 byte | 2 bytes |
|---|---|---|
| SHD | SC | SD(0) |

SHD ：0x9D
SC ：0x34
SD(0) ：LF 0x01 DF Processing result (0x00：TRUE, 0x01：FALSE)

### 3.6.9 Protocol translate processing

The translate processing to be performed between the Adapter Communication Software Protocol (ACSP) and the Adapter Communication Interface protocol (ACIP) is specified as follows:

(1) Translation from ACSP to ACIP

Translation processing is shown in Fig. 3.27.

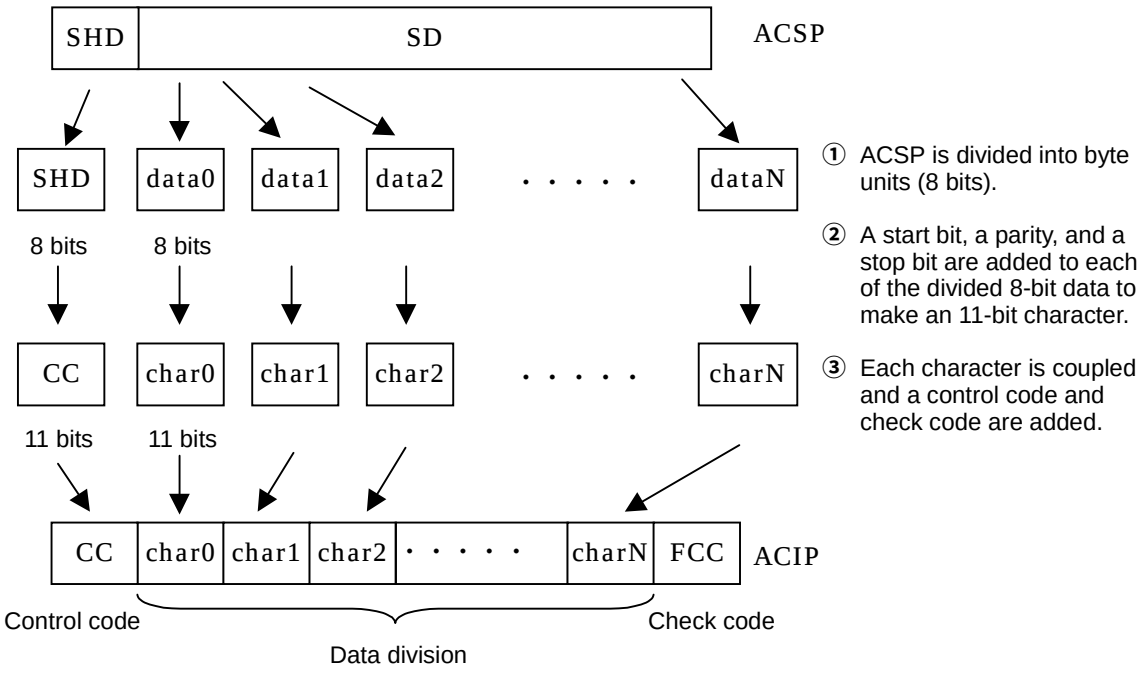



**Fig. 3.27 Translation Processing from ACSP to ACIP**

(2) ACIP to ACSP translation processing

Fig. 3.28 shows the translation processing.

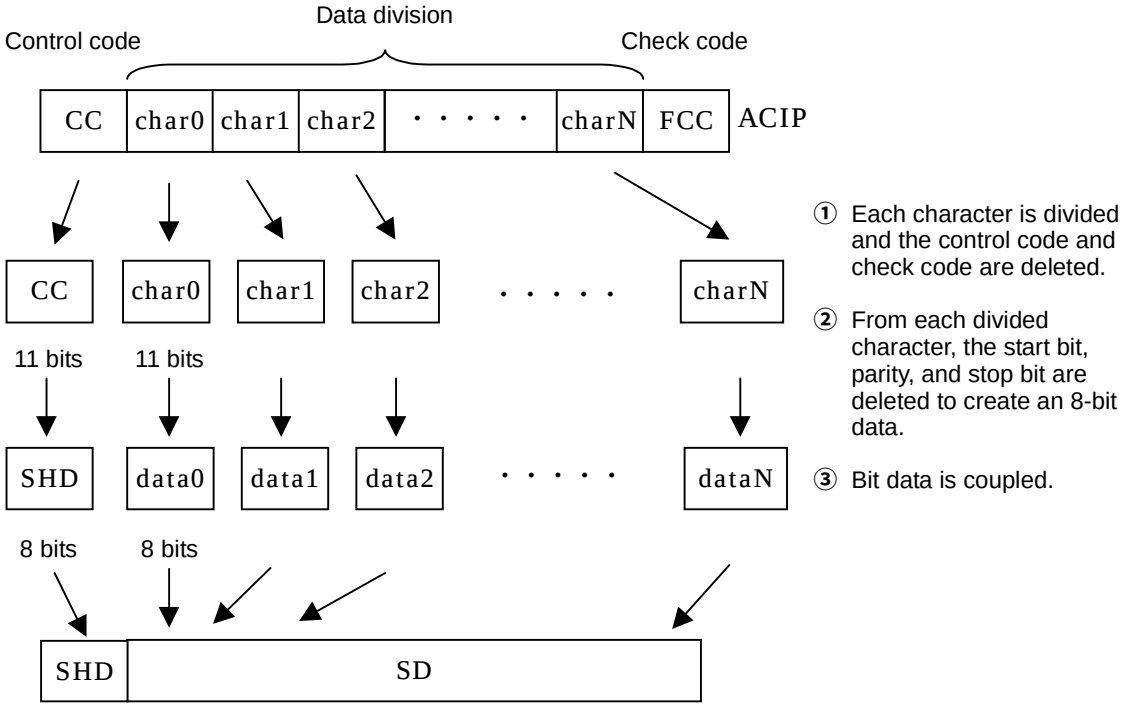



**Fig. 3.28 ACIP to ACSP Translation Processing**

### 3.6.10 Operation sequence

The operation sequence of the Adapter Communication Software depends on the common Lower-Layer Communication Interface processing mounted in the ECHONET communication processing block or the protocol difference absorption processing connected to the Adapter Communication Software through the common Lower-Layer Communication Interface. In the case of a request for data reception, for example, the following processing methods can be considered.

① The ECHONET communication processing block issues a request for data reception to the Protocol Difference Absorption Processing Block (polling processing).

② The Protocol Difference Absorption Processing Block issues a trigger to notify the receipt of data when such data has been received (event processing).

A case in which the ECHONET communication processing block and the Protocol Difference Absorption Processing Block mounting the different methods above are connected. The Adapter Communication Software must take this into consideration. Fig. 3.29 shows an example of the operation sequence for the data reception request service when the ECHONET communication processing block performs event processing and the Protocol Difference Absorption Processing Block performs polling processing.

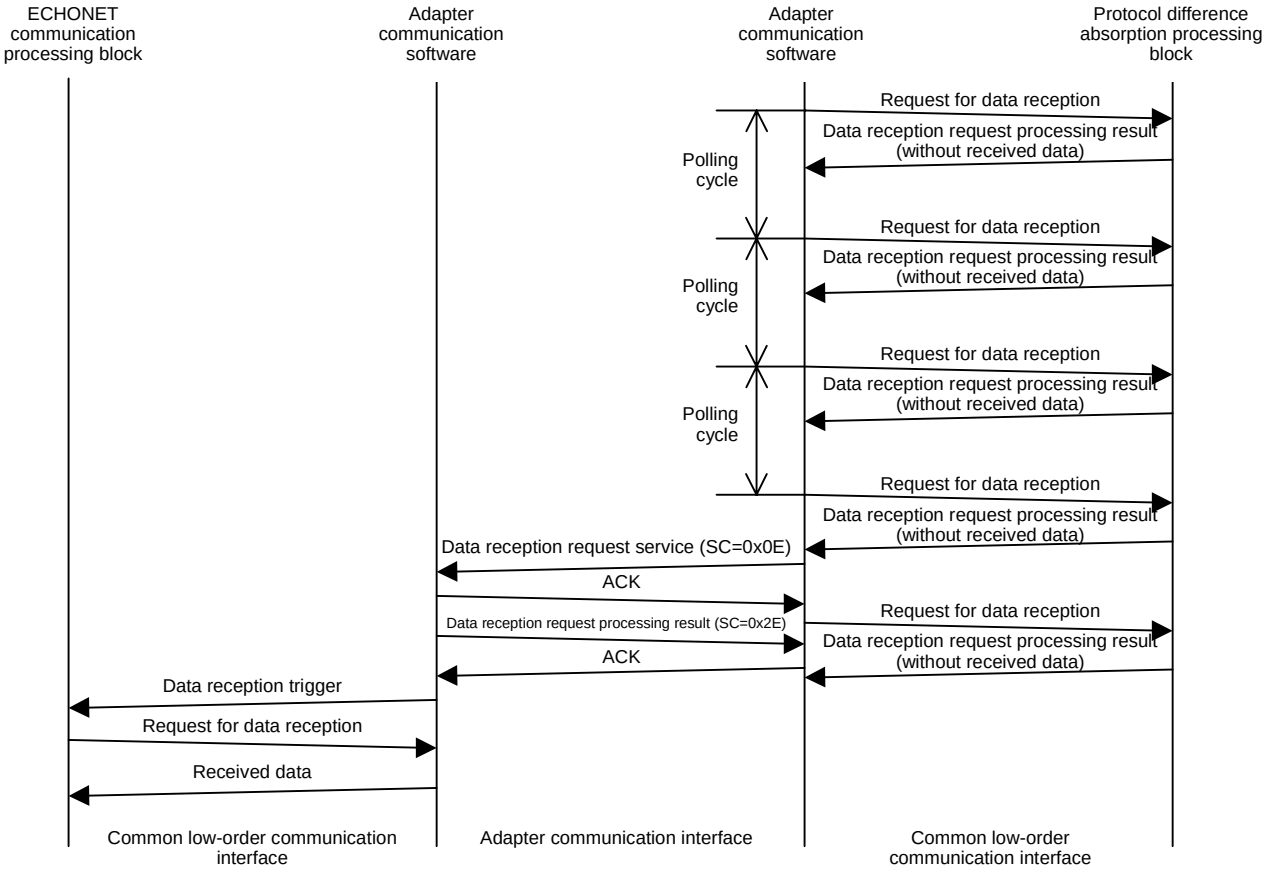



**Fig. 3.29 Adapter Communication Software Sequence**

### 3.6.11 Optional service handling

The common Lower-Layer Communication Interface supports some optional services. Accordingly, in the Adapter Communication Interface, the service responding side may not prepare a service issued by the service issuing side. In this case, the Adapter Communication Interface software on the service responding side must return "Non-processing service notice" (0x3F) as the processing result.

### 3.6.12 Optional data handling

The common Lower-Layer Communication Interface supports some types of optional data. Accordingly, the following two cases may occur in the Adapter Communication Interface.

① The service data processing for the service issued by the service issuing side is not prepared on the service responding side.

② The service data required by the service responding side does not exist in the service issued by the service issuing side.

The Adapter Communication Software must take this into consideration. An outline of the processing to be executed is described below.

<Case ①>

The Adapter Communication Software on the service responding side executes the service of the common Lower-Layer Communication Interface disregarding the non-processable service data.

<Case ②>

The Adapter Communication Software on the service responding side executes the service of the common Lower-Layer Communication Interface after compensating for the lack of service data by the default value.

### 3.6.13 Inhibition of simultaneous service issue

Only one service can be issued to the adapter communication interface at a time. More specifically, when a service request is issued by a flex ECHONET device or ECHONET device adapter, neither the request receiving side nor the request issuing side can issue any subsequent service until the previously issued service is completed. If a Flex ECHONET device and ECHONET device adapter simultaneously issue a service, the service issued by the ECHONET device adapter takes precedence. The term "simultaneous service issue" means a situation where:

a service request is received before the response to the preceding service request.

Here, the term "service" refers to the information exchange between a Flex ECHONET device and ECHONET device adapter during the time interval between the instant at which the service start conditions are established as stipulated in Section 3.6.14 and the instant at which the service ends. This time interval is referred to as the service processing period.

### 3.6.14 Service start/end conditions

In the adapter communication interface, a Flex ECHONET device and ECHONET device adapter shall conclude that a service is started or ended when the following conditions are established:

(1) Service start conditions

Service issuing side

- A service request is issued.

Service responding side

- A service request is received.

(2) Service end conditions

Service issuing side

- A service response is received.
- The timeout time has elapsed.
- The ACK/NAK timeout time for retransmission service request has elapsed.
- A new service request is received during service processing (flex ECHONET device only).

Service responding side

- The ACK signal for a service response is received.
- The ACK/NAK timeout time for a retransmitted service response has elapsed.
- A new service request is received during service processing.

### 3.6.15 Timeout

- Common to types 1, 2, and 3

  If no processing response is returned after 100 msec (timeout period) has elapsed following the issue of a service request, the next service request can be issued. Fig. 3.30 (a) shows the timeout period.

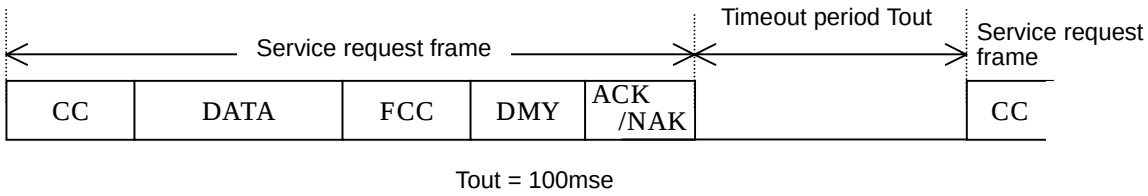



**Fig. 3.30 (a) Timeout Time (Common to Types 1, 2, and 3)**

- Types 4

  If no process response is received within 200 msec (timeout time) after ACK signal reception, a service frame can be issued. The timeout time is indicated in Fig. 3.30 (b).

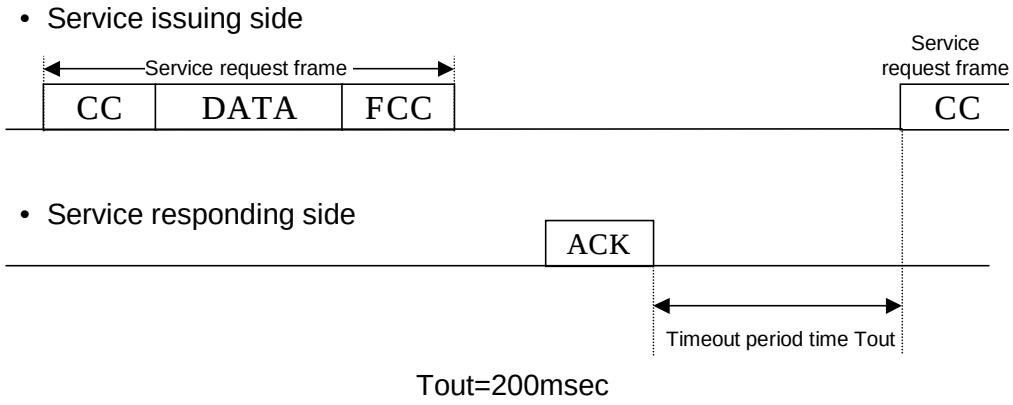



**Fig. 3.30 (b) Timeout Time (Type 4)**

# Chapter4 ECHONET Gateway

## 4.1 Basic Concept

The application software for connecting an ECHONET domain with an external system using the ECHONET protocol is called a gateway. Devices mounting this gateway are called gateway equipment. In ECHONET, however, processing to be executed by the application is not specified at present. Accordingly, the connection between ECHONET domains and external systems depends on the application software functions.

When the system is installed in an ordinary residence, we recommend that users prepare a security function, including a verification function and access control function, for the gateway application to ensure the security of the ECHONET domain. The functional definitions in such a case are described in Part 9.

# Chapter5 ECHONET Router

## 5.1 Basic Concept

ECHONET permits different types of networks to be connected as one system for operation. The ECHONET router makes the connection between two networks. The ECHONET router is not a TCP/IP router but an ECHONET-dedicated device that can perform ECHONET communication processing. However, the ECHONET router need not always be special equipment that performs only routing; for example, a PC or controller provided with multiple ECHONET communication interfaces may be operated as a router with a routing function. (As another example, an air conditioner may be provided with a function for routing between infrared and a power line. Any device type may be operated as a router with a routing function.)

Accordingly, the ECHONET router becomes communication equipment having two or more ECHONET addresses and consists of two or more nodes as necessary.

The requirements for the use of IrDA Control as ECHONET Lower-Layer Communication Software are described in Chapter 6.

## 5.2 Function Definition

The ECHONET router shall be provided with the following minimum conditions and functions:

(1) The ECHONET router is physically connected to two or more subnets. These subnets can have either different Lower-Layer Communication Protocols or same transmission media communication protocols. For each subnet to be connected, MAC addresses and ECHONET addresses must be held and managed. (The ECHONET router consists of two or more nodes.)

(2) The ECHONET communication processing block is provided with a routing function, and the communications definition object is provided with a routing table. This routing function conforms to the routing specifications explained in Part 2.

### 5.2.1 Mechanical and physical characteristics

Regarding the specification of connections with transmission media, the specification in Part 3 shall be observed in accordance with each communication protocol corresponding to the ECHONET router. Other mechanical and physical characteristics for ECHONET routers are specified below.

(1) Display block

To display the operation status of the ECHONET router, the following minimum specifications must be satisfied. For display methods using means not specified here, the specification native to the product shall be applicable. Regarding the operation status, see Part 2, Chapter 5.

- Number of LEDs

  1 LED (for operation status display)

  - LED color

  Green

  - Status display method

  Normal operation (NetID acquires) : ON
  Normal operation (NetID not acquired) : Blink (cycle 1)
  Initial operation : Blink (cycle 2)
  Error : Blink (cycle 3)
  Non-operation : OFF

  * Cycle 1.....Repetition of ON for 2 sec and OFF for 2 sec
  * Cycle 2.....Repetition of ON for 2 sec and OFF for 0.5 sec

* Cycle 3 Repetition of ON for 0.5 sec and OFF for 0.5 sec

Note: The term "initial processing" means a cold start or warm start (which is a startup achieved by performing a hardware reset process while retaining the previously acquired address and initial setup information).

### 5.2.2 Electrical characteristics

For transmission media connections, the specifications in Part 3 shall be observed for each Lower-Layer Communication Protocol corresponding to the ECHONET router.

### 5.2.3 Logical specification

As regards the logical specifications for transmission media communication protocols, the specifications provided in Part 3 for individual transmission media communication protocols supported by the ECHONET router are observed. For the Protocol Difference Absorption Processing Block, the logical specifications provided in Chapter 7 “Protocol Difference Absorption Processing Block Specification” in Part 2 are observed. For the routing specification, what conforms to the routing specification explained in Part 2 is specified.

# Chapter6 IrDA Control Router

## 6.1 Basic Concept

In ECHONET, routing processing with a subnet adjacent to subnets consisting of IrDA Controls must satisfy the requirements native to the IrDA Control in addition to the requirements for a general ECHONET router. That is, the functions as the ECHONET router must be implemented on the IrDA Control host. This is intended to absorb restrictions on IrDA Control communication functions when the IrDA Control host functions as a router.

In this Section, only the contents of the specification native to “IrDA Control Router” are described. Accordingly, see Chapter 5 for contents common to “General Router”.

### 1) Restrictions*

The IrDA Control is designed based on the specification provided for communication between a PC (host) and a peripheral device (peripheral) by infrared, and is not provided with the following functions:

① Communication between peripheral devices (peripherals) (because communications between the mouse and the keyboard are not required)
② Simultaneous broadcast communications
③ Bind start request from the host side (because communications are started with input on a peripheral device)

### 2) Restriction absorbing method

To compensate for the functions in items ① to ③ above, ECHONET specifies the following functions:

- For item ①: Communications between peripheral devices are implemented by the host’s relay of data (as described in Section **6.2 “Communications between Peripherals”**).
- For item ②: For simultaneous broadcast data, the host transmits data individually (as described in Section **6.3 “Communications of Broadcast-specified Data”**).
- For item ③: The peripheral side starts binding periodically to solve the problem of item ③ above. It is specified that the host should be provided with a receiving buffer (as described in Section **6.4 “Communication to a Peripheral in Unbind Status”**).

## 6.2 Communications between Peripherals

As described in the previous section, the IrDA cannot perform direct communications between peripherals. Communications between peripherals (so-called N:M communication) can be performed by the host's relay of data. This section describes only individual specified data. Broadcast-specified data is described in the next section.

Fig. 6.1 shows a procedure for transmitting data from peripheral A to peripheral B.

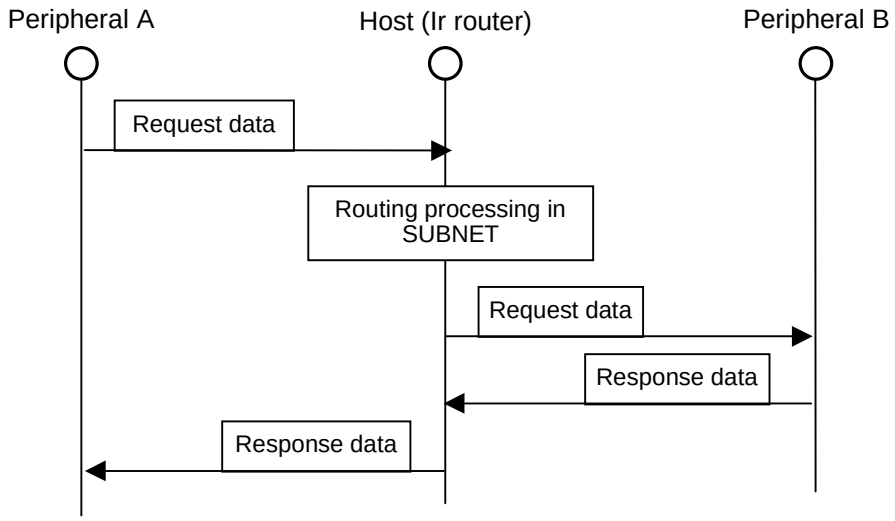



**Fig. 6.1 Communication Sequence between Peripherals**

Thus, peripheral A transmits data to the host. The host that has received data from peripheral A performs routing processing in the subnet and transmits the received data as transmission data to peripheral B. In cases requiring response data as shown in Fig. 6.1, the host can relay data in the same way as it can request data.

Fig. 6.2 shows an outline of processing between layers when peripheral-to-peripheral communication is performed. Characteristic processing for IrDA Control (host routing processing, virtual MAC address, address control table, etc.) is mainly explained as follows:

(Processing in peripheral A of the transmitting source)

① In the ECHONET Communication Middleware of peripheral A, transmission data is created. At this time, specify SEA = self NodeID and DEA = transmitting destination NodeID. (SEA = 0x02, DEA = 0x03 in the case shown in Fig. 6.2)

② The protocol difference absorption layer performs address translation processing. Specify host NodeID (= MAC address) instead of DEA NodeID as the MAC address. Next, the data is delivered to data divide processing and transmit processing. A request for data transmission is sent to the ECHONET Lower-Layer Communication Software.

(Relay processing by the host)

③ The peripheral address (PADD, 4 bits) of the transmitting source is translated into the "virtual MAC address" (8 bits) by referencing the address control table under control of the host.

④ In the Protocol Difference Absorption Processing Block, the "virtual MAC address" of the transmitting source is translated as "NodeID" of the transmitting source.

⑤ In the received data judgment processing block, when both transmitting source NetID and transmitting destination NetID are specified as those of a self-subnet, the data is determined to be intra-subnet communication data, and intra-subnet routing processing is performed. In this case, the number of EHD hops is not added, and the received data is transferred to address translation processing as transmission data.

⑥ In the address translation processing block, DEA NodeID is extracted and address translation processing is performed. In this case, DEA NodeID and transmitting destination MAC address (virtual MAC address) have a 1:1 association, so no translation is required.

⑦ In the ECHONET Lower-Layer Communication Software, the transmitting destination MAC address (virtual MAC address) is translated into PADD (4 bits) by referencing the address control table. (If bind processing has already been performed, said virtual MAC address is translated into PADD by referencing the address control table. If bind processing has not been performed, the transmission data is held in the transmitting buffer.) At this time, because the PADD of the transmitting destination peripheral is not available, the virtual MAC address must be held.

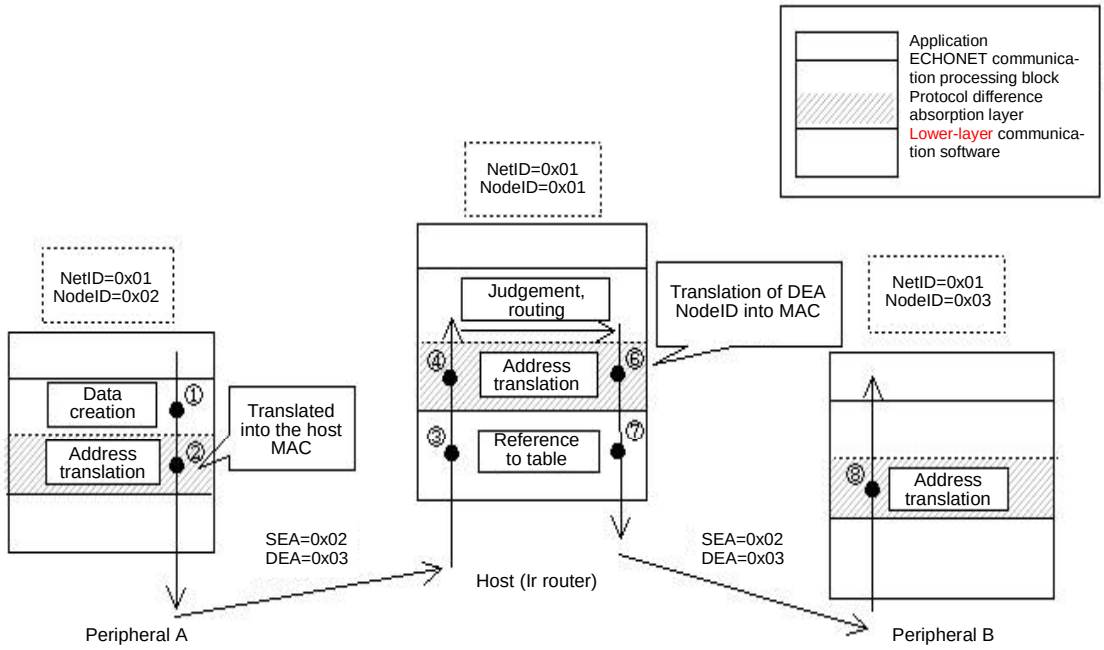



**Fig. 6.2 Outline of In-layer Processing of Transmission Sequence between Peripherals**

## 6.3 Rules of Broadcast-specified Data Communication

### 6.3.1 Overview

The IrDA Control must not be provided with a broadcast-specified data function as described in the previous section.

When the host receives broadcast-specified data including the self-subnet, it must transmit data individually to peripherals in the subnet. This section describes the processing sequence for when the host receives broadcast data.

The self-subnet broadcast specification is classified into the following two cases:

(1) When receiving broadcast-specified data from outside IrDA subnet.
(2) When receiving broadcast-specified data from inside IrDA subnet.

Each of these cases is described below.

### 6.3.2 When receiving broadcast-specified data from outside IrDA subnet

Fig. 6.3 shows a processing sequence for when broadcast-specified data, including the IrDA Control subnet, is received from outside the IrDA Control subnet. This example explains the case in which broadcast-specified data is transmitted from node X of the adjacent subnet to the IrDA Control subnet. The sequence shown in Fig. 6.3, “Response data (Note) (from host)”, is only for cases in which the host must return a response.

When the host receives request data (broadcast) oriented to the IrDA Control subnet, internal processing is as follows:

- In data judgement processing, when the EHD of received data is the broadcast specification, and the destination of the broadcast data includes “self-subnet NetID” by referencing “Broadcast specification type code” and “Broadcast target specification code”, “routing processing” is performed in the same way as the general router.
- In “routing processing”, the EHD hop count is incremented by 1, and the received data is transferred as transmission data to the Protocol Difference Absorption Processing Block through the common Lower-Layer Communication Interface.
- The protocol difference absorption layer transfers the request for transmission received from the predecessor to the ECHONET Lower-Layer Communication Software of the IrDA Control together with broadcast specification information through the Individual Lower-Layer Communication Interface.
- The ECHONET Lower-Layer Communication Software transmits broadcast data individually to transmittable peripherals (see Note 1) other than the transmitting source peripheral. In this case, the MAC address of the peripheral is the information held by the host at bind processing and requires no translation.

Note 1: "Transmittable peripherals" are peripherals in a bind status for the host. Data transmission to peripherals in unbind status is described in Section 6.4.

When transmission to all peripherals is completed or the holding time has elapsed, data in the buffer is abandoned and processing is terminated.

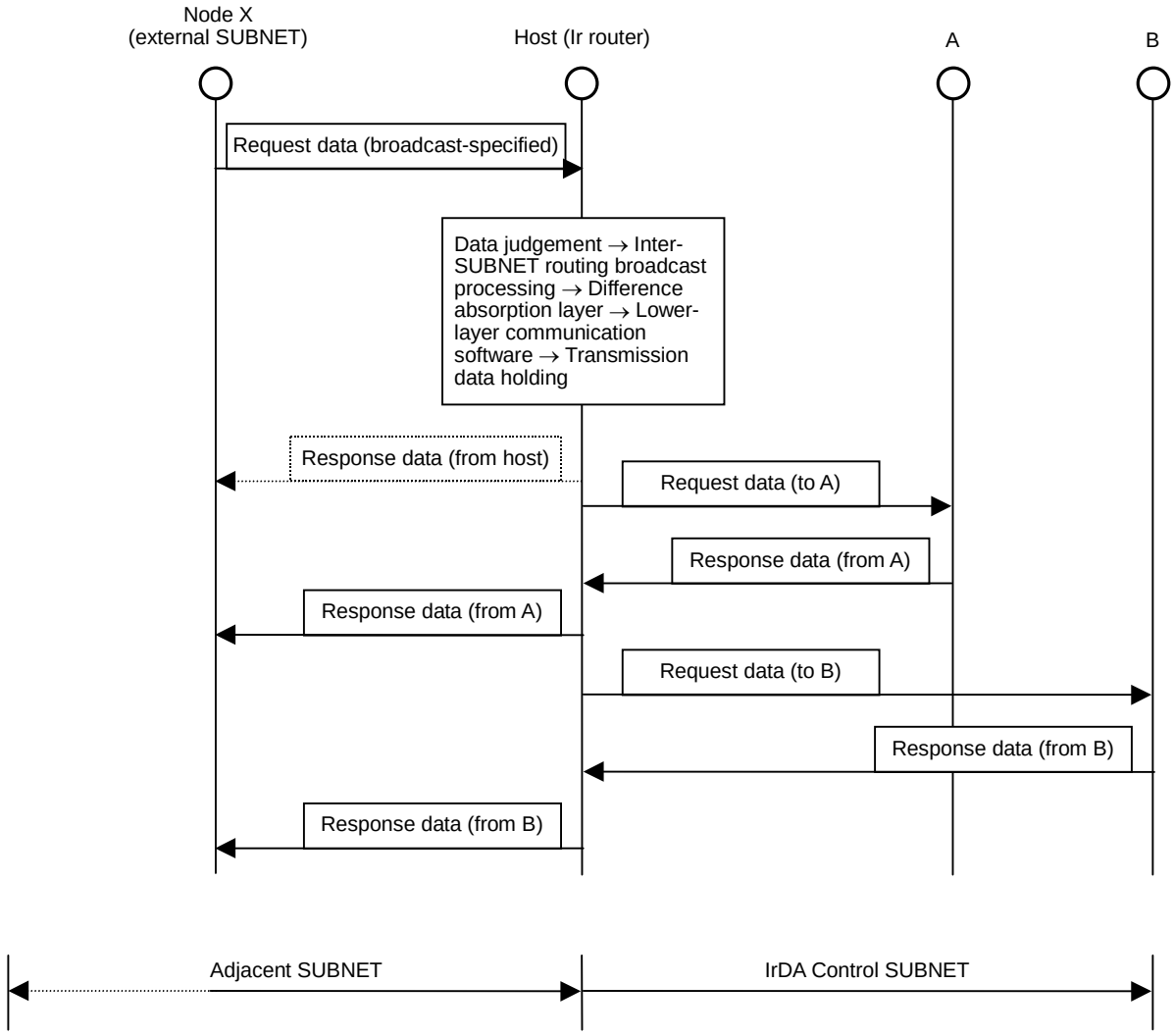



**Fig. 6.3 Broadcast-specified Data Communication Sequence from Outside IrDA Control subnet**

### 6.3.3 When receiving broadcast-specified data from inside IrDA subnet

Fig. 6.4 shows the processing sequence for when broadcast-specified data is received from the self-IrDA Control subnet.

When the host receives request data (broadcast) oriented to the IrDA Control subnet, internal processing in the host is performed as follows:

- In data judgement processing, when the EHD of the received data is the broadcast specification, the destination of the broadcast data includes “self-subnet NetID” by referencing “Broadcast specification code” and “Broadcast target specification code”, and the transmitting source NetID matches the self-subnet NetID, “Intra-subnet routing processing” native to the IrDA Control is performed. (However, routing processing to the data transmitting destination node is not performed.)
- In “Intra-subnet routine processing”, the EHD hop count is not incremented, and the received data is transferred as transmission data to the Protocol Difference Absorption Processing Block through the common Lower-Layer Communication Interface.
- The protocol difference absorption layer transfers the request for transmission received from the predecessor to the ECHONET Lower-Layer Communication Software of the IrDA Control together with broadcast specification information through the individual Lower-Layer Communication Interface.
- The ECHONET Lower-Layer Communication Software transmits broadcast data individually to transmittable peripherals (see Note 1). In this case, the MAC address of the peripheral is the information held by the host at bind processing and requires no translation.

Note 1: “Transmittable peripherals” are peripherals in a bind status for the host. Data transmission to peripherals in the unbind status is described in Section 6.4. When transmission to all peripherals is completed or the holding time has elapsed, data in the buffer is abandoned and processing is terminated.

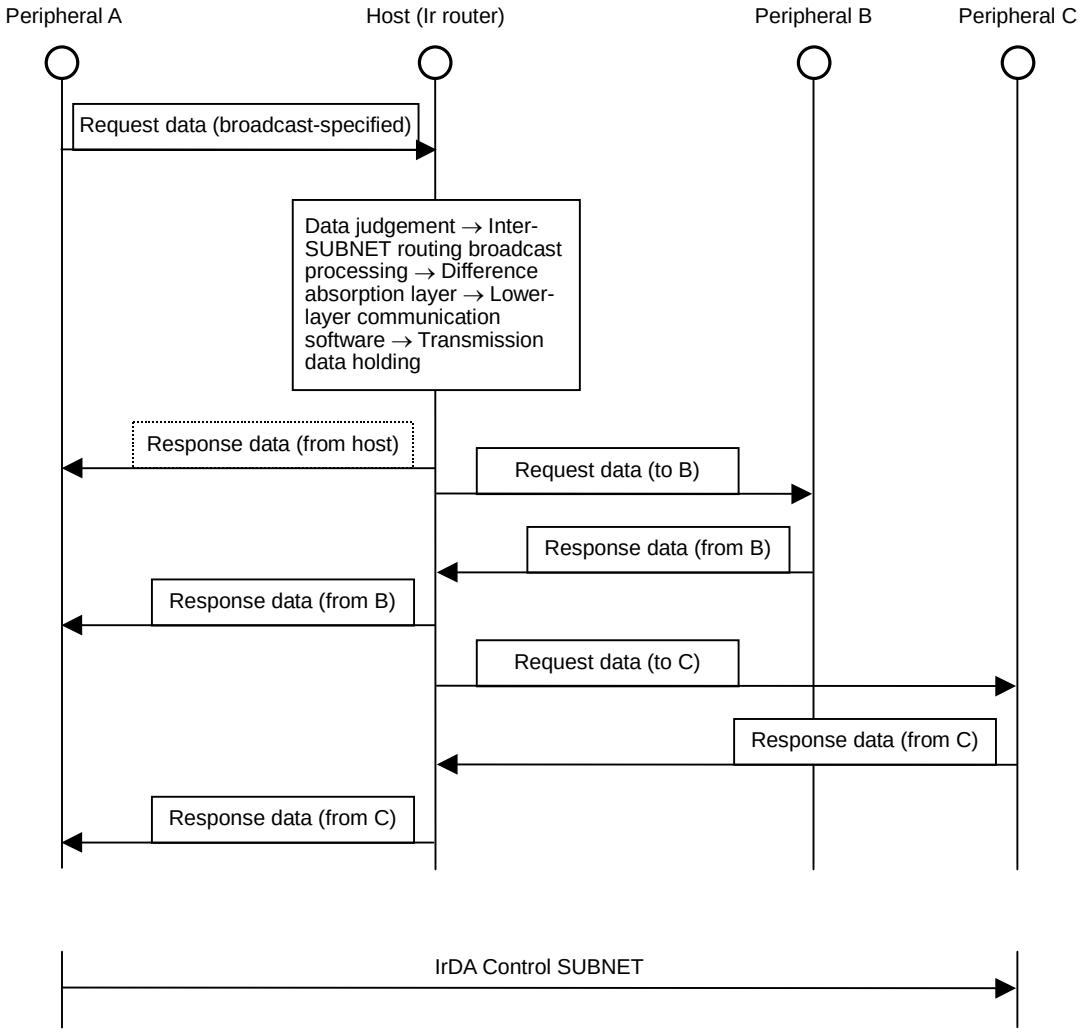



**Fig. 6.4 Broadcast-specified Data Communication Sequence from Inside IrDA Control subnet**

## 6.4 Communication to a Peripheral in the Unbind Status

### 6.4.1 Basic Concept

As described in 1) Restrictions in Section 6.1, the IrDA Control cannot perform a bind start request function from the host side. In this situation, data communication cannot be performed from the host in the unbind status (idle status) or another subnet. In ECHONET, a means for compensating for this problem has been adopted in the specification to secure bi-directionality of communication start in pseudo form. Therefore, the ECHONET Lower-Layer Communication Software must be provided with the following functions:

Peripheral-dedicated Lower-Layer Communication Software specification

- It is mandatory to mount a function to make a bind request to the host periodically in accordance with “Bind request interval (*1)” that can be set by an application. However, this interval can be set arbitrarily. An infinite interval shall also be allowed.

Host-dedicated communication software specification

- The host can hold data oriented to the self-subnet received from an external subnet or the self-SIUBNET during the “Data holding time (*2)” set for each peripheral. Mounting of this function shall also be mandatory.

### 6.4.2 Sequence

Fig. 6.5 shows a processing sequence for data from another subnet to a peripheral in the unbind status. The sequence is explained below.

- The host receives data oriented to the self-subnet.
  (Includes both individual and broadcast; data from self-subnet is the same.)
- The host performs data judgement and routing processing as described in the previous section and transfers the request for transmission to the ECHONET Lower-Layer Communication Software.
- If the transmitting destination peripheral is in the bind status, data is transmitted immediately. If it is in the unbind status (including broadcast), data is held in the buffer.
- Each peripheral mounts a fixed-time communication function and makes a bind request to the host at the set interval.
- When receiving a bind request from the peripheral, the host starts bind processing for the peripheral.
- Data is transmitted to a peripheral in the bind status. If a response is required, the response is returned.

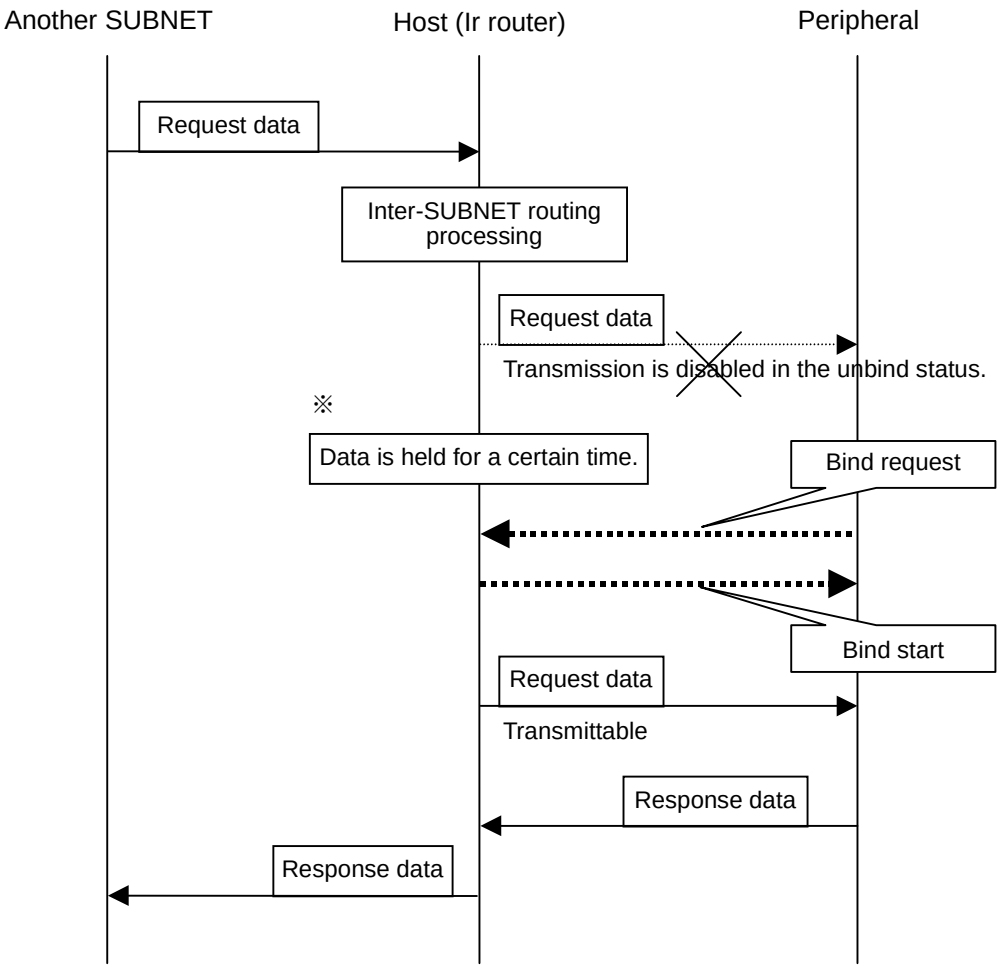



**Fig. 6.5 Communication Sequence to a Peripheral in the Unbind Status**

"Bind Interval (*1)"

The ECHONET standard does not specify any "Bind interval (*1)" setting. However, the following point must be considered in system design.

The IrDA Control Specification specifies that current status switches to unbind status if non-communication status continues for 5 or 30 seconds between the host and a peripheral. (This value can be set to five from thirty seconds.) Accordingly, to keep the bind status at all times in ECHONET, the transition time to the unbind status must be set to "30 sec" and "Bind interval (*1)" must be set to 30 sec or less. Usually, this setting is desirable when constructing a system.

When the main purpose of an application is to notify the central monitoring equipment of the occurrence of an event by using a body sensor when a peripheral is driven by a battery, “Bind interval (*1)” should be set to “Infinite”, thereby minimizing battery power consumption and extending battery life.

When setting “Data holding time (*2)”, the following two issues should be considered from the viewpoint of system design and operation:

1. When “Bind interval (*1)” is set so as not to switch to the unbind status
   The host can transmit received data to the corresponding peripheral immediately, so the data holding time may be set to several seconds.
2. When “Bind interval (*1)” is set so as to switch to the unbind status
   “Data holding time (*2)” must be set to a longer value than “Bind interval (*1)”, or the host will not be able to transmit received data to the corresponding peripheral.

# Appendix 1 Reference Document

(1) “PH-CONNECTOR” issued by JST Mfg. Co., Ltd.