User's Manual

R-IN32M4-CL3
User's Manual CC-Link IE Field

Document number: R18UZ0071EJ0100
Issue date: Nov 1, 2019

Renesas Electronics
www.renesas.com

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

General Precautions in the Handling of Products

The following usage notes are applicable to CMOS devices from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

   - The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   - The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
     In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
     In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   - The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   - When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

How to Use This Manual

1. Purpose and Target Readers

This manual is intended for users who wish to understand the functions of industrial Ethernet communications ASSP (Application Specific Standard Product) "R-IN32M4-CL3" (R9J03G019GBG) and design application systems using it.
Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.

When designing an application system that includes this MCU, take all points to note into account. Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual. The mark “<R>” in the text indicates the major revisions to this version. You can easily find these revisions by copying “<R>” and entering it in the search-string box for the PDF file.

Literature — Literature may be preliminary versions. Note, however, that the following descriptions do not indicate "Preliminary". Some documents on cores were created when they were planned or still under development. So, they may be directed to specific customers. Last four digits of document number (described as ****) indicate version information of each document. Please download the latest document from our web site and refer to it.

Documents related to R-IN32M4-CL3

Document Name	Document Number
R-IN32M4-CL3 User’s Manual: Hardware	R18UZ0073EJ****
R-IN32M4-CL3 User’s Manual: Gigabit Ethernet PHY	R18UZ0075EJ****
R-IN32M4-CL3 User’s Manual: Board Design	R18UZ0074EJ****
R-IN32M4-CL3 User’s Manual: CC-Link IE TSN	R18UZ0070EJ****
R-IN32M4-CL3 User’s Manual: CC-Link IE Field	This Manual
R-IN32M4-CL3 Programming Manual: Driver	R18UZ0076EJ****
R-IN32M4-CL3 Programming Manual: OS	R18UZ0072EJ****

2. Numbers and Symbols

Data significance: Higher digits on the left and lower digits on the right

Active low representation:

xxxZ (capital letter Z after pin or signal name)

or xxx_N (capital letter _N after pin or signal name)

or xxnx (pin or signal name contains small letter n)

Note:

Footnote for item marked with Note in the text

Caution:

Information requiring particular attention

Remark:

Supplementary information

Numeric representation:

Binary … xxxx , xxxxB or n'bxxxx (n bits)

Decimal … xxxx

Hexadecimal … xxxxH or n'hxxxx (n bits)

Prefix indicating power of 2 (address space, memory capacity):

K (kilo) … $2^{10} = 1024$

M (mega) … $2^{20} = 1024^2$

G (giga) … $2^{30} = 1024^3$

Data Type:

Word … 32 bits

Halfword … 16 bits

Byte … 8 bits

Contents

1. INTRODUCTION

1.1 Introduction

1.1.1 CC-Link Partner Association (CLPA)

(1) Specifications

The materials related to this manual include the specifications published by the CC-Link Partner Association below. For CC-Link IE Field Network and SLMP details, download and refer to the following documents from the CC-Link Partner Association website.

Table 1.1 CC-Link IE Field Network Specification List (CC-Link Partner Association)

Document Title	Document No.
CC-Link IE Field Network Specification (Overview)	BAP-C2005-001
CC-Link IE Field Network Specification (Physical Layer and Data Link Layer)	BAP-C2005-002
CC-Link IE Field Network Specification (Application Layer Service)	BAP-C2005-003
CC-Link IE Field Network Specification (Application Layer Protocol)	BAP-C2005-004
CC-Link IE Field Network Specification (Communication Profile)	BAP-C2005-005
CC-Link IE Field Network Specification (Implementation Rules)	BAP-C2005-006
CC-Link IE Field Network Specification (Device Profile)	BAP-C2005-007
SLMP Specification (Overview)	BAP-C2006-001
SLMP Specification (Services)	BAP-C2006-002
SLMP Specification (Protocol)	BAP-C2006-003

(2) CC-Link IE Field Utility

The CC-Link IE Field Utility is a tool that simulates the master station on a personal computer. This tool allows you to simulate a simple master station when you do not have a master station but do have a PC. Feel free to download and utilize the tool from the CC-Link Partner Association website.

Table 1.2 CC-Link IE Field Utility

Document Title / Relate Tool	Document No.
CC-Link IE Field Utility	—
CC-Link IE Field Utility Operation manual	—

(3) Conformance Test

When a product is developed based on the information in this manual, the product must undergo a conformance test implemented by the CC-Link Partner Association. For conformance test details, download and refer to the following documents from the CC-Link Partner Association website.

Table 1.3 CC-Link IE Field Network Conformance Test Specifications

Document Title/Related Tool	Document No.
CC-Link IE Field Network Intelligent Device Station Conformance Test Specifications	BAP-C0401-037
CC-Link IE Field Network Remote Device Station Conformance Test Specification	BAP-C0401-041

(a) Creating a Control & Communication System Profile (CSP+)

The conformance test includes verification of CSP+. CSP+ must be created in advance. For CSP+ details, download and refer to the following documents from the CC-Link Partner Association website. From this same site, you can also download and utilize CSP+ Creation Guidelines, CSP+ Creation Support Tool, and other documents.

Table 1.4 Control & Communication System Profile (CSP+)

Document Title/Related Tool	Document No.
Control & Communication System Profile Specification	BAP-C2008-001
Control & Communication System Profile Creation Guidelines	—
CSP+ Profile Creation support Tool	—
Sample CSP+ Files	—
CSP+ Templates	—

(b) Implementing the 1000BASE-T Compliance Test

The conformance test includes verification by a 1000BASE-T compliance test (a waveform test based on IEEE802.3 specifications), which requires implementation by the user.

There are test labs capable of implementing the 1000BASE-T compliance test. For details, contact the CC-Link Partner Association.

(4) Inquiries

To request materials published by the CC-Link Partner Association (CLPA) and for conformance test details, please contact the following:

CC-Link Partner Association	TEL: +81-52-919-1588
	FAX: +81-52-916-8655
	E-mail: info@cc-link.org
	Web: http://www.cc-link.org/

1.1.2 Generic Terms and Abbreviations

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Table 1.5 Generic Terms and Abbreviations for This Manual

(1/2)

Generic Term/Abbreviation	Description
R-IN32M4-CL3	A Gigabit Ethernet PHY (GbE-PHY) built-in communication LSI for intelligent device stations and remote device stations of CC-Link IE Field Network
R-IN32M4-CL3 application circuit	A communication circuit of CC-Link IE Field Network that consists of R-IN32M4-CL3 and peripheral devices
R-IN32M4-CL3 application product	A CC-Link IE Field Network compatible product manufactured with reference to this manual
R-IN32M4-CL3 driver	A program group that controls R-IN32M4-CL3. A driver consists of R-IN32M4-CL3 driver interface functions, R-IN32M4-CL3 driver callback functions, and an R-IN32M4-CL3 driver main body.
User program	An application program created by a user. A program is used as reference for checking the logic of an intelligent device station or remote device station, and therefore needs to be customized.
Station	An element that constitutes the network and sends, receives, and transfers data. The term, node, is used with the same meaning.
Master station	A station that controls CC-Link IE Field Network. This station can perform cyclic and transient transmission with all stations.
Local station	A station that performs cyclic and transient transmission with the master station and other local stations. This station receives data, RX, RY, RWr, and RWw, of other slave stations by cyclic transmission.
Intelligent device station	A station that exchanges I/O signals (bit data) and I/O data (word data) with the master station by cyclic transmission. This station responds to a transient transmission request from another station and also issues a transient transmission request to another station.
Remote device station	A station that exchanges I/O signals (bit data) and I/O data (word data) with the master station by cyclic transmission. This station responds to a transient transmission request from another station.
Remote I/O station	A station that exchanges I/O signals (bit data) with the master station by cyclic transmission
Slave station	A generic term for stations other than a master station: local station, remote I/O station, remote device station, and intelligent device station
Own station	An intelligent device station or remote device station to be developed based on this manual
Other station	A station other than the own station
Station number	An identifier for uniquely identifying a station in the network. The term, node number, is used with the same meaning.
Cyclic transmission	A function by which data are periodically exchanged among stations on the same network using link devices (RX, RY, RWw, and RWr)
Transient transmission	A function of communication with another station, which is used when requested by a user application
Data link	A generic term for cyclic transmission and transient transmission
Remote input (RX)	Bit data input from a slave station to the master station
Remote output (RY)	Bit data output from the master station to a slave station
Remote register (RWr)	Word data input from a slave station to the master station
Remote register (RWw)	Word data output from the master station to a slave station

Table 1.5 Generic Terms and Abbreviations for This Manual

(2/2)

Generic Term/Abbreviation	Description
Link device	A generic term for RX, RY, RWw, and RWr
Device	A device (X, Y, M, D, or others) in a programmable controller CPU, or memory in a user application, where data communicated with R-IN32M4-CL3 are stored
Buffer memory	Memory in a user application, where data (such as setting values and monitoring values) are stored
Baton pass	A token to send data over a network
Disconnection	A process of stopping a data link when a data link error occurs
Return	A process of restarting data link when a station recovers from an error
SLMP	The abbreviation for "SeamLess Message Protocol". This protocol is used to access an SLMP-compatible device or a programmable controller connected to an SLMP-compatible device from an external device.
User	A manufacturer who develops and sells CC-Link family compatible products based on this manual. The terms, vendor and partner manufacturer, are used with the same meaning.
End user	A purchaser and user of CC-Link family compatible products developed by users
GX Works2 / GX Works3	The product name of the software package for the MELSEC programmable controllers
Master/local module	A generic term for the CC-Link IE Field Network master/local modules: RJ71GF11-T2, QJ71GF11-T2, and LJ71GF11-T2 CC-Link IE Field Network master/local module, and for the modules: RJ71EN71 and RnENCPU when the CC-Link IE Field Network function is used
Hold/Clear processing	Processing for holding or clearing data (RY) normally received up to the present time when cyclic transmission stops or disconnection occurs during cyclic transmission.
RIRD	The abbreviation for the master/local module dedicated instructions JP.RIRD and GP.RIRD
RIWT	The abbreviation for the master/local module dedicated instructions JP.RIWT and GP.RIWT
CSP+	The abbreviation for Control & Communication System Profile. This is the specifications for describing information required for CC-Link Family compatible device startup, operation, and maintenance.

1.1.3 Related Manuals

This manual does not include CC-Link Field Network details such as terminology and functions. Please refer to following manuals as necessary.

Table 1.6 CC-Link IE Field Network Related Manuals

Related Manual Titles	Manual Number (Model Name Code)
[MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)] Describes the Ethernet, CC-Link IE Controller Network and CC-Link IE Field Network specifications, procedures from preparation to operation, system configuration, wiring, and communication examples.	SH(NA)-081256ENG (13JX09)
[MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)] Describes the CC-Link IE Field Network functions, parameter settings, programming, troubleshooting, input/output signals, buffer memory, and the like.	SH(NA)-081259ENG (13JX18)
[MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual] Describes the CC-Link IE Field Network and MELSEC-Q series master/local module specifications, procedures from preparation to operation, system configuration, installation, setup, functions, programming, and troubleshooting.	SH(NA)-080917ENG (13JZ47)
[MELSEC-L CC-Link IE Field Network Master/Local Module User's Manual] Describes the CC-Link IE Controller Network and MELSEC-L series master/local module specifications, procedures from preparation to operation, system configuration, installation, settings, functions, programming, and troubleshooting.	SH(NA)-080972ENG (13JZ54)
[MELSEC-L CC-Link IE Field Network Head Module User's Manual] Describes head module specifications, procedures from preparation to operation, system configuration, installation, setup, and troubleshooting.	SH(NA)-080919ENG (13JZ48)
[SLMP Reference Manual] Describes the protocol (SLMP) used for data reading and writing with SLMP compatible devices from an external device.	SH(NA)-080956ENG (13JV23)

1.1.4 Usage Precautions

(1) The specifications and other information described in this manual are subject to change without notice.

(2) The sample code described in this manual is for the development of an intelligent device station or remote device station using R-IN32M4-CL3. The sample code indicates an example of use of the materials herein; its operation is not guaranteed by Mitsubishi.

(3) The intellectual property rights of the information provided on the CD-ROM and the information downloaded from the Mitsubishi Electric FA site are the property of Mitsubishi Electric Corporation. Reprinting the information without the consent of Mitsubishi Electric Corporation and reproduction of the information for any purpose other than the development of an intelligent device station or remote device station is prohibited. Distribution of a product (object code or the like) after sample code compilation, however, is unrestricted.

1.2 Overview

This manual describes how to develop an intelligent device station or remote device station for CC-Link IE Field Network using "CC-Link IE Field Network intelligent device stations and remote device stations communication LSI R-IN32M4-CL3 with GbE-PHY".

The main information included in this manual is as follows:

- User program design (see Section 4 "CREATING THE USER PROGRAM")
- R-IN32M4-CL3 driver specifications (see Section 5 "R-IN32M4-CL3 DRIVER RELATED SPECIFATIONS")

1.2.1 R-IN32M4-CL3 Performance Specifications

(1) Performance Specifications of CC-Link IE Field Network

The following table lists the R-IN32M4-CL3 performance specifications related to CC-Link IE Field Network.

Table 1.7 CC-Link IE Field Network Performance Specifications

Item			Description	
Station type			Intelligent device station or remote device station	
Station number			1 to 120	
Network number			1 to 239	
Communication speed			1 Gbps	
Network topology			Line, star, and ring (Coexistence of line topology and star topology is possible.)	
Connection cable			Ethernet cable that satisfies 1000BASE-T standards: Category 5e or higher (double shielded, STP), straight cable	
Maximum station-to-station distance			100 m	
Overall cable distance			Link topology: 12000 m (when cables are connected to 1 master station and 120 slave stations) Star topology: Depends on the system configuration. Ring topology: 12100 m (when cables are connected to 1 master station and 120 slave stations)	
Number of cascade connections			Up to 20	
Cyclic transmission			Intelligent device station	Remote device station
	Maximum number of link points per station	RX	2048 points (2048 bits), 256 bytes	128 points (128 bits), 16 bytes
		RY	2048 points (2048 bits), 256 bytes	128 points (128 bits), 16 bytes
		RWr	1024 points (1024 words), 2048 bytes	64 points (64 words), 128 bytes
		RWw	1024 points (1024 words), 2048 bytes	64 points (64 words), 128 bytes
Transient transmission			Intelligent device station	Remote device station
	Client function		Supported	Not supported
	Server function		Supported	Supported
	Data size[Note]		2048 bytes	1024 bytes
	Dedicated instruction		RIRD/RIWT can be received by customizing the sample code.	
Other functions			—	
	Diagnostic function		CC-Link IE Field Network diagnostic function	
	Temporary error invalid station		Supported	
	Fast linkup		Supported	

Note. When the transient frame to be sent or received exceeds 1518 bytes, the transient frame is divided and sent or received in blocks.

1.2.2 Sample Code Overview

(1) Sample Code Overview

The following describes sample code overview.

Table 1.8 File Overview

Folder Name		Details
cie_intelligent_device	Sample Driver	Sample code for intelligent device station. •User programs •R-IN32M4-CL3 driver main body •R-IN32M4-CL3 driver interface functions •R-IN32M4-CL3 driver call back functions
	IAR	IAR Embedded Workbench project files (compiler setting, linker setting, build setting)
cie_remote_device	Sample Driver	Sample code for remote device station, IAR Embedded Workbench project files (intelligent device station too)
	IAR	

The sample code consists of the user program, R-IN32M4-CL3 driver interface functions, R-IN32M4-CL3 driver callback functions, and the R-IN32M4-CL3 driver main body.

Caution: The sample code describes only the information related to CC-Link IE Field Network (communication function information).

(1) The user program is an application program created by the user. The program in the sample code is provided for your reference for checking intelligent device station or remote device station logic. Customize the program in accordance with user requirement specifications. (See Section 4 "CREATING THE USER PROGRAM".)

(2) R-IN32M4-CL3 driver interface functions are functions called when an R-IN32M4-CL3 driver function is used by the user program. Customization is not required. (See Section 5.2 "R-IN32M4-CL3 Driver Interface Function List".

(3) R-IN32M4-CL3 driver callback functions describe examples of processing on the user program side in response to events that occur on the R-IN32M4-CL3 driver side. Customize the functions in accordance with user requirement specifications. (See Section 5.4 "R-IN32M4-CL3 Driver Callback Function List".)

(4) The R-IN32M4-CL3 driver main body is the main body of the driver area that is called by R-IN32M4-CL3 driver interface functions and controls R-IN32M4-CL3. Customization is not required.

(2) System Configuration

The following describes an example of the software configuration of an R-IN32M4-CL3 application product. With usage of the OS driver library[Note1] and various functions provided by the R-IN32M4-CL3 driver, the user program can utilize various R-IN32M4-CL3 functions, such as cyclic transmission and transient transmission.

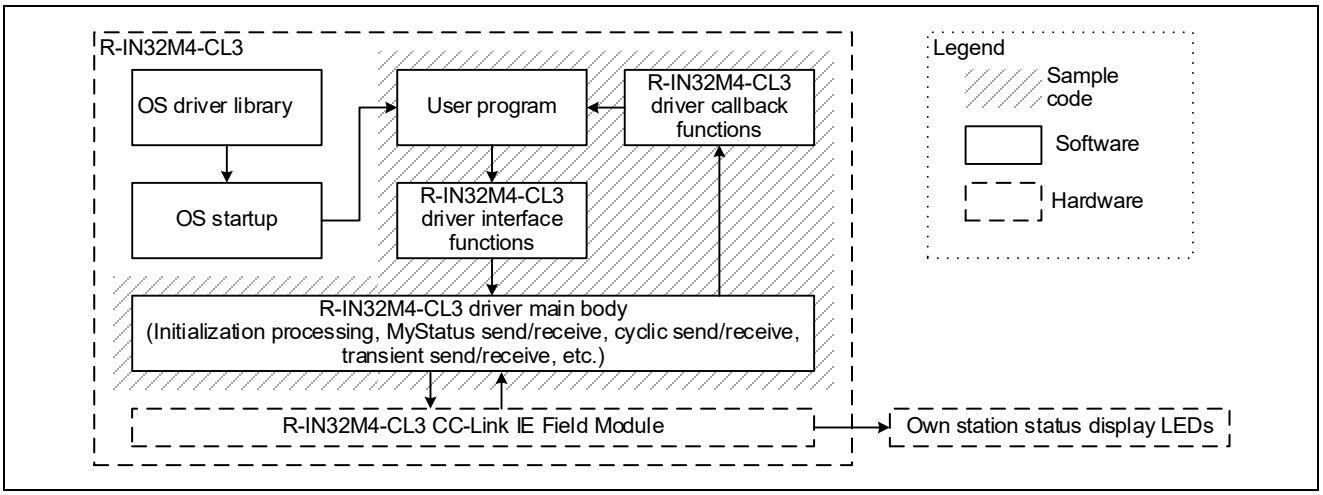


Figure 1.1 Software Configuration Overview

2. INVESTIGATING AND PREPARING SPECIFICATIONS PRIOR TO DEVELOPMENT

This section describes the specifications to be investigated and preparations to be made when developing an R-IN32M4-CL3 application product.

The following is an example of the user development process. The specifications to be investigated and the preparations to be made in each step are described in the sections that follow.

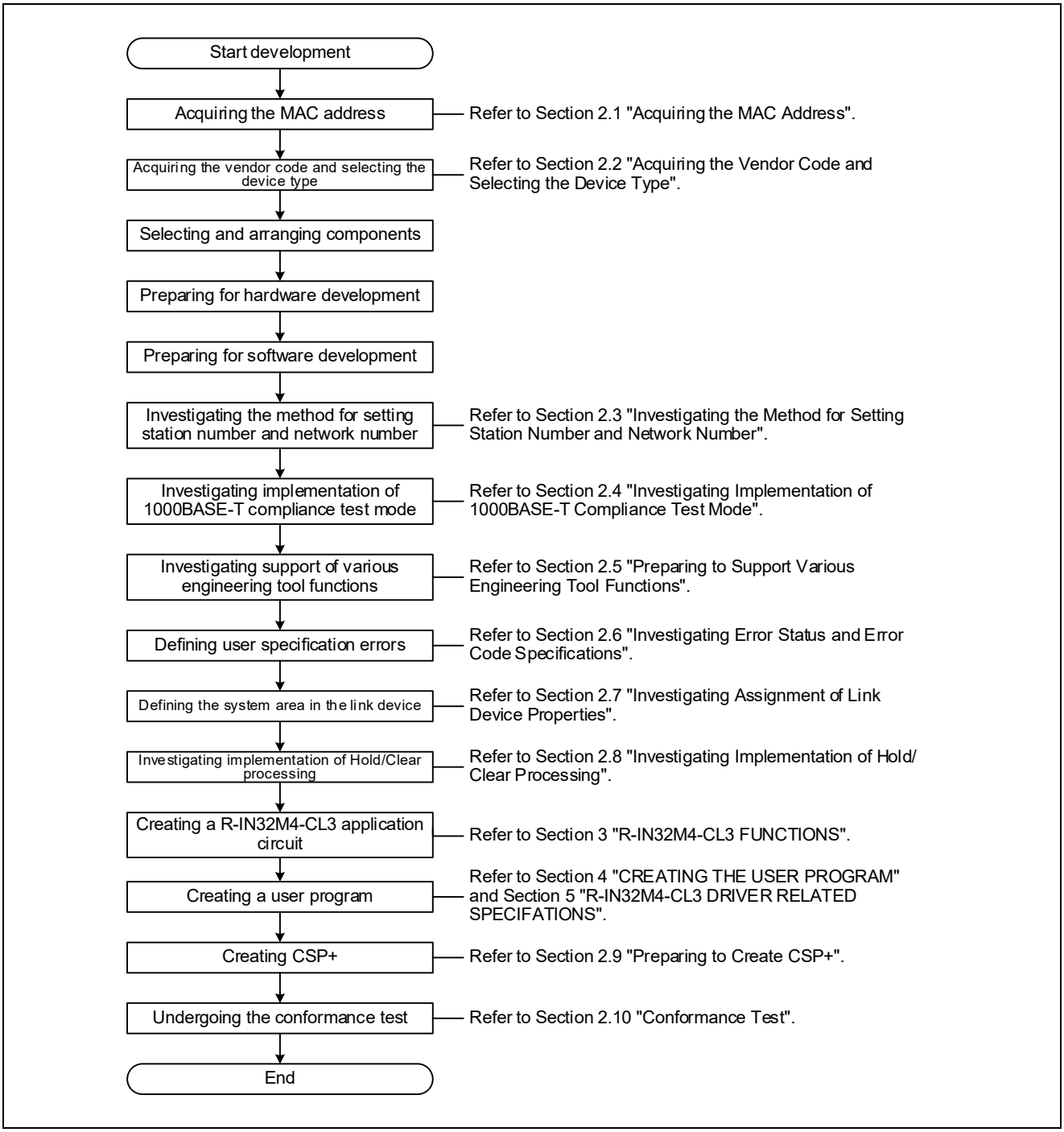


Figure 2.1 Development Process Example

2.1 Acquiring the MAC Address

R-IN32M4-CL3 application products are Ethernet (IEEE 802.3ab) compliant. Be sure to acquire a MAC address MA-L (MAC Address Block Large) unique to the device to be developed.

To acquire a MAC address, contact the following authority (department) in the USA.

The IEEE Registration Authority
445 Hoes Lane Piscataway, NJ 08854 USA
Phone : +1 (732) 465-6481
Fax : +1 (732) 562-1571
Web : https://standards.ieee.org/products-services/regauth/oui/ (As of November 2019)

2.2 Acquiring the Vendor Code and Selecting the Device Type

R-IN32M4-CL3 application products require registration of a vendor code and device type. The vendor code and device type are assigned and managed by the CC-Link Partner Association. If you have any questions, contact the CC-Link Partner Association.

Table 2.1 Vendor Code and Device Type

Item	Description
Vendor code (vendorCode)	ID number (fifth to eighth digits) issued when the vendor joined the CC-Link Partner Association.Note
Device type (deviceType)	Select the applicable device type from the "CC-Link IE Field Network Specification (Device Profile)". If an applicable device type does not exist, consult with the CC-Link Partner Association.

Note. If the ID number is 123-456-7890, the vendor code is 5678.

2.3 Investigating the Method for Setting Station Number and Network Number

To create a data link to the own station, a station number and network number need to be set in R-IN32M4-CL3. Investigate method for setting the station number and network number in accordance with the specifications of the R-IN32M4-CL3 application product in advance.

For example, the station number and network number can be set by using a hardware switch or by using the engineering tool (peripheral device) of the R-IN32M4-CL3 application product.

For either method, call "Station number and network number setting" (Section 5.3.1 (3) "gerR_IN32_SetNodeAndNetworkNumber") of the R-IN32M4-CL3 driver interface functions in the user program "iUserInitialization" (Section 4.2.2 "Initialization Processing").

For reference, the following describes examples of how to set the station number and network number.

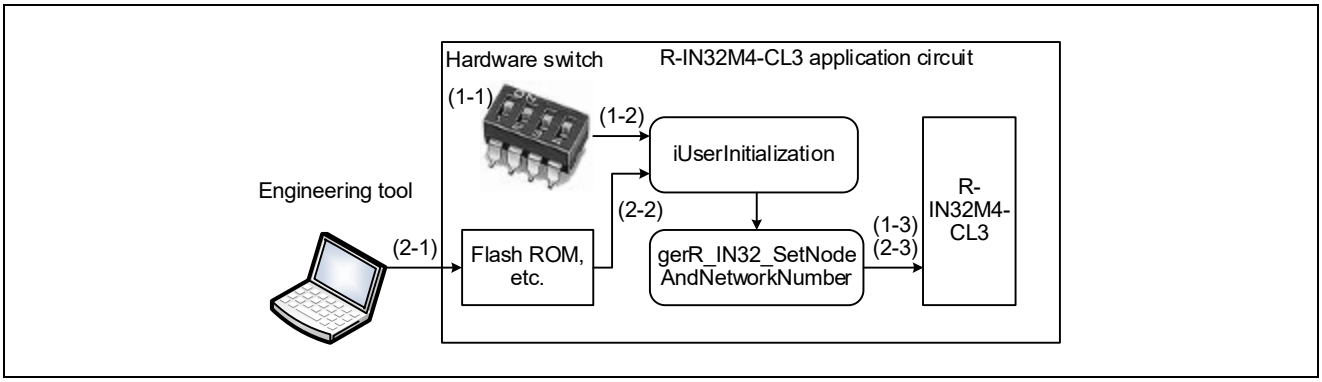


Figure 2.2 Image of Setting Station Number and Network Number

Table 2.2 <Example 1: Using a hardware switch>

Step	Description
(1-1)	Set the station number and network number using a hardware switch.
(1-2)	The user program "iUserInitialization" acquires the current values of the hardware switch, and sets the values in the arguments of the R-IN32M4-CL3 driver interface function "gerR_IN32_SetNodeAndNetworkNumber". The process for acquiring the current values of the hardware switch is not written in the user program "iUserInitialization". Add the process in accordance with user specifications.
(1-3)	The R-IN32M4-CL3 driver interface function "gerR_IN32_SetNodeAndNetworkNumber" sets the argument values in R-IN32M4-CL3.

Select a hardware switch that corresponds to the value range of the station number and network number.

Table 2.3 Hardware Switch Range

Item	Value Range
Station number	01h to 78h (1 to 120)
Network number	01h to EFh (1 to 239)

Table 2.4 <Example 2: Using the engineering tool>

Step	Description
(2-1)	Set the station number and network number data in Flash ROM or the like using the engineering tool.
(2-2)	The user program "iUserInitialization" acquires the data set in Flash ROM, and sets the data in the arguments of the R-IN32M4-CL3 driver interface function "gerR_IN32_SetNodeAndNetworkNumber". The process for acquiring the data set in Flash ROM is not written in the user program "iUserInitialization". Add the process in accordance with user specifications.
(2-3)	The R-IN32M4-CL3 driver interface function "gerR_IN32_SetNodeAndNetworkNumber" sets the argument values in R-IN32M4-CL3.

2.4 Investigating Implementation of 1000BASE-T Compliance Test Mode

CC-Link IE Field Network compatible products are 1000BASE-T compliant, and require implementation of the 1000BASE-T compliance test based on IEEE802.3ab specifications.

Remark: Verify the results of the 1000BASE-T compliance test using the CC-Link Partner Association conformance test.

The 1000BASE-T compliance test measures four test waveforms from the Ethernet ports as verification of transmission path waveforms. Investigate implementing functions and processings in the R-IN32M4-CL3 application product that switch the test waveform at desired timings during testing.

For example, the test waveform can be switched by using a hardware switch or by using the engineering tool (peripheral device) of the R-IN32M4-CL3 application product. For either method, call the R-IN32M4-CL3 driver interface function "IEEE802.3ab compliance test" (Section 5.3.14 (1) "gerR_IN32_IEEETest") from the user program "UserIEEETest" (Section 4.4.1 "Hardware Test (IEEE802.3ab Compliance Test)").

Table 2.5 Example: Switching the 1000BASE-T Compliance Test Mode

Step	Description
(1)	Implement a function that switches the mode to a "compliance test mode (offline mode)", which is other than the mode (online mode) used during normal operation, from an external source (such as a hardware switch) of the R-IN32M4-CL3 application product.
(2)	Implement processing that can specify the arguments "MODE1" to "MODE4" of the R-IN32M4-CL3 driver interface function "gerR_IN32_IEEETest" from an external source (such as a hardware switch) of the R-IN32M4-CL3 application product.

2.5 Preparing to Support Various Engineering Tool Functions

The CC-Link IE Field Network diagnostics and parameter processing/command execution of slave stations can be performed using the engineering tool. Investigate whether or not the specifications of the R-IN32M4-CL3 application product (slave station) will support engineering tool functions in advance.

2.5.1 CC-Link IE Field Network Diagnostics

The engineering tool graphically displays the status of CC-Link IE Field Network. For details, see Section 3.9 "CC-Link IE Field Network Diagnostics".

2.5.2 Parameter Processing/Command Execution of Slave Stations

By using the engineering tool, parameter setup and command execution can be performed on the developed device without programming. For details, see Section 2.9.1 "Parameter Processing/Command Execution of Slave Stations".

The above two functions are performed using transient transmission (SLMP frames).
The R-IN32M4-CL3 application product (slave station) needs to respond to SLMP request frames from the master station. Investigate implementing SLMP frame send/receive in advance.

Table 2.6 Engineering Tool Functions

No.	Engineering Tool Function		Items Required in R-IN32M4-CL3 Application Product
1	CC-Link IE Field Network diagnostics		SLMP frame request reception and response send processing
	a	Selected station communication status monitor	
	b	Communication test	
	c	Cable test	
2	Parameter processing/command execution of slave stations		Write CSP+ up to the scope [3] in the figure illustrated in Section 2.9 "Preparing to Create CSP+". SLMP frame request reception and response send processing

For SLMP frame send/receive, see Section 4.3 "User Program Details (Transient Transmission Related)" and the sections that follow. (SLMP frame send/receive processing is described in the sample code. Use the processing described.)

2.6 Investigating Error Status and Error Code Specifications

When an error occurs in an R-IN32M4-CL3 application product, the error status is reported to other stations. In addition, when a request frame is abnormally received during transient transmission, the error code is reported to the request source. Investigate the specifications of error status and error code in advance in accordance with the specifications of the R-IN32M4-CL3 application product.

For error status specifications, see Section 2.6.1 "Investigating Error Status Specifications". For transient transmission error code specifications, see Section 2.6.2 "Investigating Specifications of Error Codes Stored in Return Codes (RSTS) and End Codes (End Code)".

[Error codes for errors other than transient transmission errors]
Error codes include those for R-IN32M4-CL3 hardware errors and for user application area[Note] errors.

Note. User-defined function areas (I/O areas, sensor areas, temperature control areas, etc.)

The error codes for R-IN32M4-CL3 hardware errors are detected by the R-IN32M4-CL3 driver but not reported to other stations. The error processing after detection is optional. (See Section 5.5 (1) "gR_IN32_CallbackFatalError".)

The error codes for user application area errors are not reported to other stations. The definitions and implementation are optional.

2.6.1 Investigating Error Status Specifications

The R-IN32M4-CL3 driver stores its own station error status in the detailed application error status of the MyStatus frame, and reports the status to other stations. (See Section 3.8 "MyStatus Send/Receive Function".)

Define minor, moderate, and major error statuses in accordance with R-IN32M4-CL3 application product specifications. For a reference when defining error status, the error statuses of the programmable controller CPU module are classified as follows:

- Minor error: An error in which the CPU module continues operation, such as a battery error
- Moderate error: An error in which the CPU module stops operation, such as a WDT error
- Major error: An error in which the CPU module stops operation, such as a RAM error
  (Errors that are more serious than moderate errors that may involve hardware failure are considered as major errors.)

2.6.2 Investigating Specifications of Error Codes Stored in Return Codes (RSTS) and End Codes (End Code)

When an R-IN32M4-CL3 application product abnormally receives Transient2 request frame, the error code is stored in the return code (RSTS) of the response frame and sent to the request source to report the error and cause.

Store error codes in the return code (RSTS) in accordance with the specifications of the R-IN32M4-CL3 application product so that the end user can check the return code and take action using the user's manual of the developed device. For details, see Section 7.3 "CC-Link Compatible Transient Frame".

When an R-IN32M4-CL3 application product abnormally receives SLMP request frame, similar to the Transient2 request frame, store the error code in the end code (End Code) in accordance with the specifications of the R-IN32M4-CL3 application product.
For details, see Section 7.4.2 "Details of End Code (End Code)" in Section 7.4 "SLMP Frame".

2.7 Investigating Assignment of Link Device Properties

Link devices are classified into three property groups. Investigate the property groups to be assigned to the link devices of the R-IN32M4-CL3 application product in advance. (For details, refer to "CC-Link IE Field Network Specification (Device Profile)".)

2.7.1 Direct Input/Output Group

The direct input/output group uses link devices for general input/output and not for specific functions. (For example, input/output of master/local modules, remote I/O modules, and others.)

2.7.2 System Input/Output Group

The system input/output group uses link devices for interlock between the master station and its own station, notification of own station status. (For details, see Section 6 "LINK DEVICE SYSTEM AREA".)

Table 2.7 Example of Link Device Property Definitions (System Area)

Link Device	Name	Link Device	Name
RX07	Warning status flag	—	—
RX08	Initial data processing request flag	RY08	Initial data processing complete flag
RX09	Initial data setting complete flag	RY09	Initial data setting request flag
RX0A	Error status flag	RY0A	Error reset request flag
RX0B	Remote ready	—	—

2.7.3 Vendor Input/Output Group

The vendor input/output group uses arbitrarily defined link devices.

Table 2.8 Example of Link Device Property Definitions (Inverter)

Link Device	Name	Link Device	Name
RX10	Forward rotation status/stop status	RY10	Forward rotation command/stop command
RX11	Reverse rotation status/stop status	RY11	Reverse rotation command/stop command
RWr00	Output frequency status	RWw00	Output frequency setting

2.8 Implementation of Hold/Clear Processing

In an R-IN32M4-CL3 application product that is controlling output to an external device, "hold/clear processing" refers to processing in which the R-IN32M4-CL3 application product is made to continue (hold) or stop (clear) the output, when cyclic transmission has stopped for reasons such as data link disconnection, master station shutdown, or a master station error.

Taking the cautionary note below into consideration, decide upon the implementation of hold or clear processing as a failsafe when data link disconnection, master station shutdown, or a master station error occurs.

For the hold/clear processing when the master station is shut down or has an error, see Section 4.2.7 "MyStatus from Master Station and Cyclic Receive Processing".

The master station status can be monitored by receiving MyStatus frames. For details on the master station information acquired by the MyStatus frame, see Section 3.8.2 "Receiving MyStatus ". For the hold/clear processing when the data link is disconnected, see Section 4.2.10 "Communication Status Update Processing ".

Caution: This note applies when the slave station (own station) is using the R-IN32M4-CL3 driver (gerR_IN32_GetReceivedCyclicData) to acquire cyclic reception data.

The acquired cyclic data differs according to whether the the master station is operating or has an error and the state of the data link.

For details on the acquired cyclic data, see Section 4.2.7 "MyStatus from Master Station and Cyclic Receive Processing" and Section 4.2.10 "Communication Status Update Processing ".

2.9 Preparing to Create CSP+

CSP+ is specifications for describing required information for starting, operating, and maintaining CC-Link Family compatible products. Providing CSP+ to the end users of the R-IN32M4-CL3 application product allows them to manage all stations of CC-Link IE Field Network using one engineering tool.

For CSP+ details, refer to "Control & Communication System Protocol Specification". To create CSP+, use "CSP+ Creation Support Tool".

The following shows the scope in which CSP+ files are to be created for the intelligent device station or the remote device station. The conformance test includes CSP+ verification. Be sure to create CSP+ of scope [1]. Investigate which functions (creation scopes [2] and [3]) of the engineering tool are to be supported by the specifications of the R-IN32M4-CL3 application product in advance.

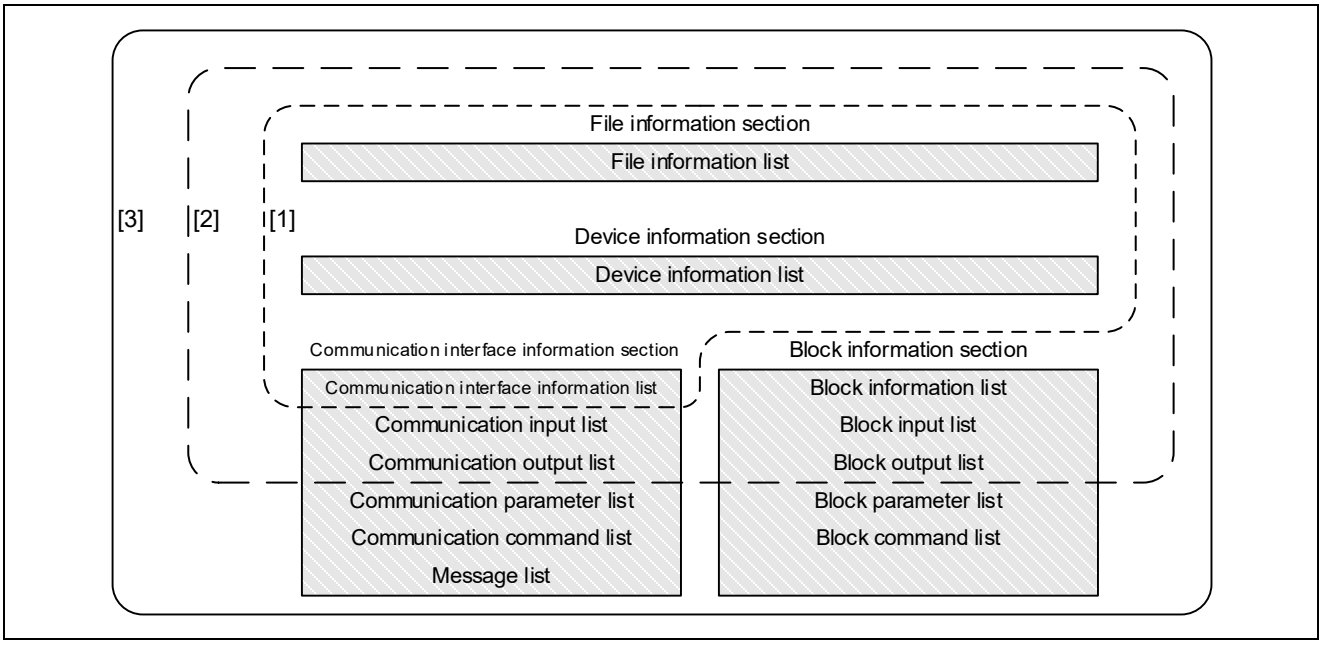


Figure 2.3 CSP+ File Section Configuration

Scope	Description	Necessity
[1]	Information required for verifying mandatory items in the CC-Link Partner Association conformance test [GX Works2, GX Works 3] R-IN32M4-CL3 application products are displayed in the CC IE Field Configuration window and the network configuration can be easily created.	Required
[2]	Information required for displaying slave station link device and master station device assignments	Optional
[3]	Information required for executing parameter processing/command execution of slave stations[Note] [GX Works2, GX Works 3] The parameters of CC-Link IE Field Network compatible products can be easily set from the CC IE FIELD configuration window.	Optional

Note. For details, see Section 2.9.1 "Parameter Processing/Command Execution of Slave Stations".

2.9.1 Parameter Processing/Command Execution of Slave Stations

R-IN32M4-CL3 application products support parameter processing/command execution of slave stations, making it possible to reduce the programming required for parameter setup and command execution by the end user of the R-IN32M4-CL3 application product.

Investigate whether or not the specifications of the R-IN32M4-CL3 application product will support parameter processing/command execution of slave stations by the engineering tool in advance.

[Example of parameter processing of slave station]

The following parameters can be set up on a Mitsubishi Electric remote I/O station (NZ2GF2B1-***) without programming.

	Name	Initial Value	Read Value	Write Value	Setting Range	Unit
Station parameter						
☑	Input response time setting	5: 10ms	5: 10ms	5: 10ms		
☑	Output HOLD/CLEAR setting	0: CLEAR	0: CLEAR	0: CLEAR		
☑	Cyclic data update watch tim...	0	0	0	0 to 20	x100ms
☑	Mode switch	9: Automatic...	9: Automatic...	9: Automatic...		
☑	Initial operation setting	0: with initial...	0: with initial...	0: with initial...		
Basic module parameter						
☑	⊟ Synchronous Input Timing Ac...					
	Synchronous Input Timing ...	0: Disable	0: Disable	0: Disable		
☑	⊟ Input OFF delay setting					
	Input OFF delay setting X0	0	0	0	0 to 150000	x400us

Figure 2.4 Example of Slave Station Parameter Processing by CC IE Field Configuration Window

Parameter processing/command execution of slave stations can be achieved by satisfying the following:

- Describing CSP+ up to scope [3] in the figure illustrated in Section 2.9 "Preparing to Create CSP+"
- Implementing the SLMP frame send/receive processing described by CSP+ in the R-IN32M4-CL3 application product

2.10 Conformance Test

The conformance test is a test implemented for each device in order to ensure high reliability in the communication of CC-Link IE Field Network compatible products. The test verifies that the user product satisfies the CC-Link IE Field Network communication specifications and is connectable to the network.

Acquire the conformance test specifications when preparing for development, and design the R-IN32M4-CL3 application product so that it satisfies the test requirement specifications.

A CC-Link IE Field Network compatible product that passes the conformance test can be included as a recommended product in the "CC-Link Partner Product Catalog" and other medium.

3. R-IN32M4-CL3 FUNCTIONS

3.1 Reset

This function initializes the CPU and GbE-PHY areas of R-IN32M4-CL3. "Power-on reset" and "system reset" in this manual refers to resetting from the following sources.

[Power-on reset]

- Reset by signal input from the PONRZ pin (including initialization of R-IN32M4-CL3 internal RAM)

[System reset]

- Reset by signal input from the HOTRESETZ pin (excluding internal PLL)
- Reset by system reset register (SYSRESET) (equivalent to signal input from the HOT RESETZ pin)

For details, refer to the related document "R-IN32M4-CL3 User's Manual: Hardware".

3.2 Fast Linkup

"Fast linkup" refers to a way of shortening the time from turning on the power to linkup of the R-IN32M4-CL3 application product.

The linkup time can be shortened by disabling the auto-negotiation of GbE-PHY (the function for automatically selecting the optimal speed and mode of communications). A system using a tool changer on the tip of an industrial robot arm is used to illustrate an example of this function.

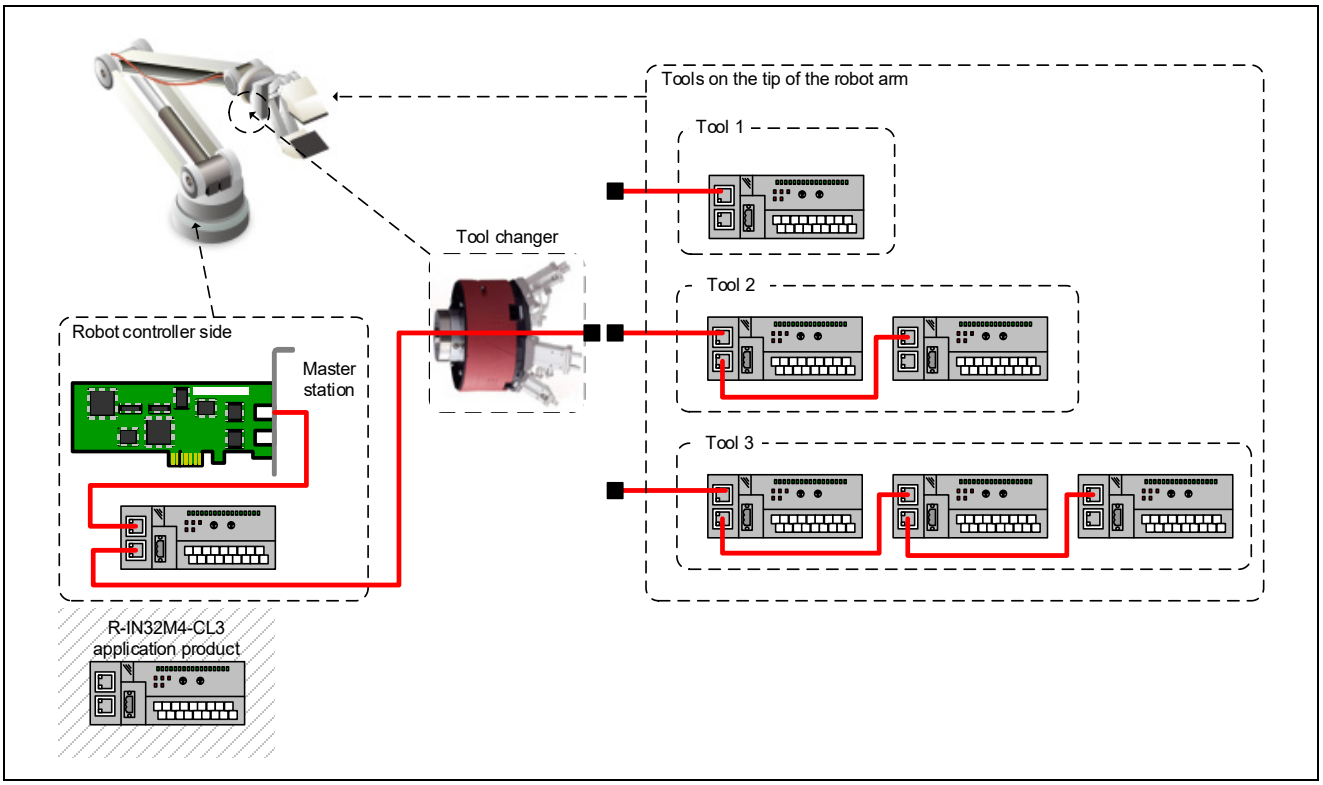


Figure 3.1 Example of Fast Linkup Usage (for an Industrial Robot Arm)

The following items are required to realize this function.

[Hardware]

A hardware switch is required to switch fast linkup on or off. For details, see Section 3.2.1 "Setting Fast Linkup".

[Software]

Include iUserInitialization (see Section 4.2.2 "Initialization Processing") as additional processing in the user program, to read and check the value of said hardware switch. If a hardware switch is not to be used, include additional processing to check the state of fast linkup from the engineering tools (peripheral devices) of the R-IN32M4-CL3 application product.

If fast linkup is enabled, make the fast linkup setting (set the initial value for PHY) in initialization of the R-IN32M4-CL3. For details, see Section 5.3.1(2) "gerR_IN32_Initialize" and the entry on the initial value of PHY setting in Table 5.7 "R_IN32_UNITINIT_T List".

[Caution]

When this function is enabled for both the own station port and the adjacent other-party station port, the linkup time between the ports is reduced. It does not shorten the linkup time if fast linkup is enabled for one port but not the other, or if the other port does not support fast linkup.

3.2.1 Setting Fast Linkup

When fast linkup is to be made operational, set the function for respective ports.

Linkup is not possible if another-party port has the function disabled or does not support it. Therefore, in consideration of cases when an adjacent port has the function disabled or does not support it, and of cases where the situation regarding support is not clear, the end user requires a simple way of enabling and disabling fast linkup.

For reference, the following table shows how to set up fast linkup when a switch for this purpose is implemented as part of the R-IN32M4-CL3 application circuit. The dedicated switch used here is an 8-bit general hardware switch block, the state of which is acquired in the sample software by the get_board_sw function.

Table 3.1 Switch Block for Making the Fast Linkup Settings

Bit	Name	Description
0	F LINK P1	ON: The fast linkup of port 1 is enabled OFF: The fast linkup of port 1 is disabled
1	F LINK P2	ON: The fast linkup of port 2 is enabled OFF: The fast linkup of port 2 is disabled
2	— (Reserved)	Reserved for future functional expansion
3	— (Reserved)	Reserved for future functional expansion
4	— (Reserved)	Reserved for future functional expansion
5	— (Reserved)	Reserved for future functional expansion
6	— (Reserved)	Reserved for future functional expansion
7	— (Reserved)	Reserved for future functional expansion

Table 3.2 Making the Fast Linkup Settings

Step	Description
(1)	Use the fast linkup settings switch block to set the states of ports 1 and 2.
(2)	The user program acquire the states of the hardware switches by a call of the get_board_sw function in iUserInitialization.
(3)	Initialize the R-IN32M4-CL3 by making the settings for the initial value of PHY (see Table 5.8 "R_IN32_PHY_SETTING_T List") as shown below with the driver interface function gerR_IN32_Initialize in iUserInitialization of the user program. [Value of bit 0 acquired in step (2)] ・When the bit is ON (enabled), set the MDI of port 1 to forced MDI and set the 1000BASE-T clock as a forced master. ・When the bit is OFF (disabled), settings for the MDI of port 1 and the 1000BASE-T clock are to be made automatically [Value of bit 1 acquired in step (2)] ・When the bit is ON (enabled), set the MDI of port 2 to forced MDI-X and set the 1000BASE-T clock as a forced slave. ・When the bit is OFF (disabled), settings for the MDI of port 2 and the 1000BASE-T clock are to be made automatically
(4)	The R-IN32M4-CL3 driver proceeds with linkup according to the settings in step (3).

[Caution]

The states of the switches as referred to above on activation of the R-IN32M4-CL3 application product determine the states of the ports in terms of fast linkup. Changing the state of a fast linkup setting switch after activation of the R-IN32M4-CL3 application product does not change the state in terms of fast linkup. To update the state of a setting after activation, turn the power off and then on again.

3.2.2 System Configuration with Fast Linkup

A system configuration with fast linkup is shown below, taking an industrial robot arm as an example.

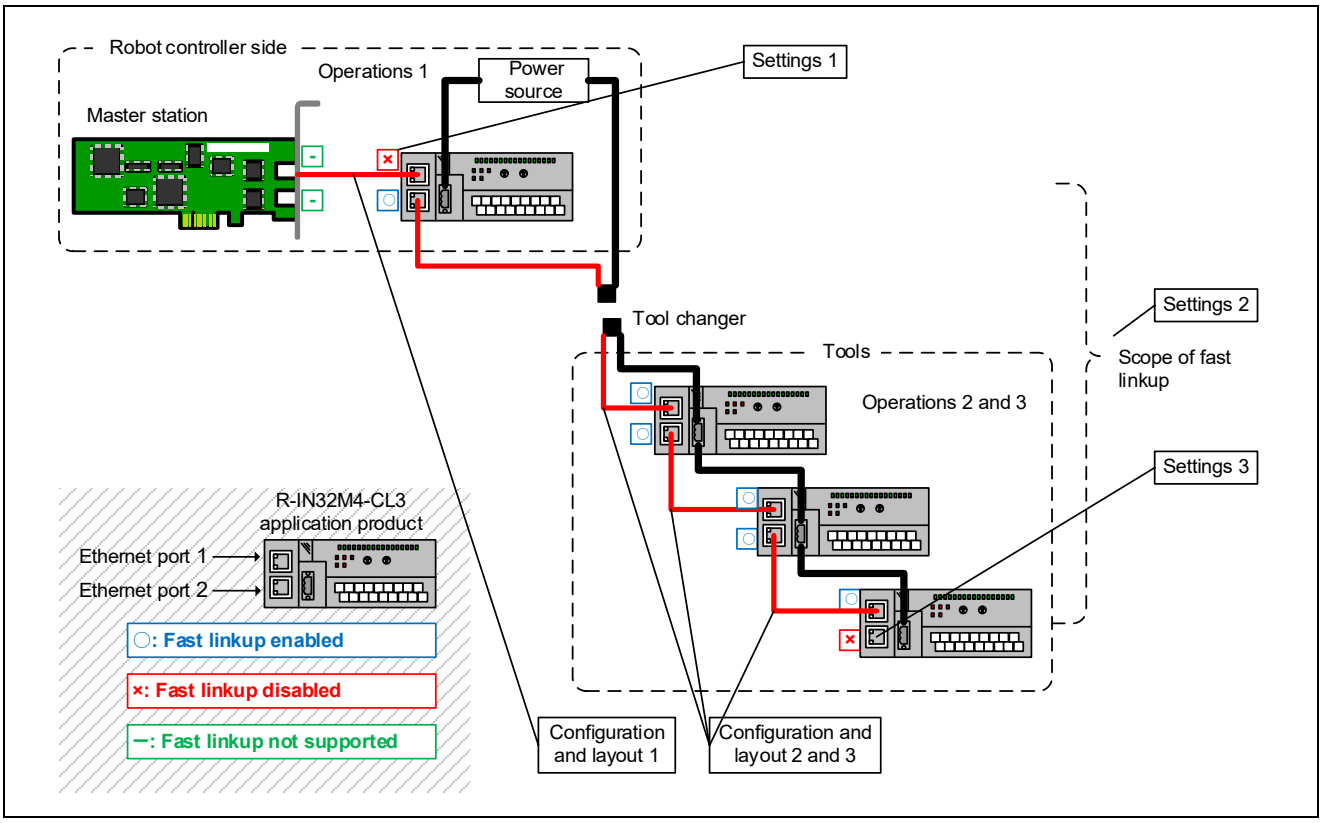


Figure 3.2 System Configuration with Fast Linkup (for an Industrial Robot Arm)

[Configuration and layout]

1. Connect port 1 of the R-IN32M4-CL3 application product on the robot controller side to the master station, and port 2 to the tool side.

In the R-IN32M4-CL3 application product, port 1 and port 2 are fixed as the master and the slave, respectively, in terms of setting the 1000BASE-T clock. The clock setting is performed for the slave first, and then for the master. Setting the clock for port 2 (slave) first can reduce the time required for linkup with the tool.

2. When connecting two R-IN32M4-CL3 application products with fast linkup, connect port 2 of the upstream station to port 1 of the downstream station.

In the R-IN32M4-CL3 application product, as described above, port 1 and port 2 are fixed as the master and the slave, respectively, in terms of setting the 1000BASE-T clock. Therefore, even if fast linkup is enabled, connecting port 1 to port 1 or port 2 to port 2 does not support linkup.

3. Use linear connections for the R-IN32M4-CL3 application products in the tool block.

If these products are connected in a network topology other than linear, linkup may take more time. In addition, the time required for the linkup will increase with the number of connected products.

[Settings]

1. For port 1 of the R-IN32M4-CL3 application product, which is to be connected to the station that does not support fast linkup (the master station), fast linkup should be disabled.

   When a port with fast linkup enabled is connected to a port that has fast linkup disabled or does not support fast linkup, linkup will not proceed.

2. Fast linkup should be enabled for all ports of the R-IN32M4-CL3 application product that are within the scope of fast linkup.

3. In the case of the R-IN32M4-CL3 application products placed at the tool end, set the non-connected ports to have fast linkup disabled.

[Operations]

1. Operate the master station on the robot controller side and the R-IN32M4-CL3 application products with the power on at all times.

2. After switching to a tool, turn on the power simultaneously for all of the R-IN32M4-CL3 application products in the given tool.

3. When connecting the cable and the power again after switching to a tool, the master station should detect[Note] the disconnection of the previous tool before making the new connection.

   Connecting the cable to the power supply before the master detects the disconnection may lengthen the time required for linkup.

   The GbE-PHY in the R-IN32M4-CL3 takes about 0.4 to 0.8 seconds to detect the disconnection after a tool is disconnected.

   Turning on the power (connecting) before the disconnection has been detected may lengthen the time required for linkup because the disconnection was not detected with the correct timing.

Note. Disconnection can be detected by monitoring the states of the station data links (SW00B0 to SW00B7) with the master station.

3.2.3 Notes on Fast Linkup

If fast linkup is to be used, the degree to which the times are shortened depends on the R-IN32M4-CL3 application circuit and the user program.

Shorten the following items.

- The time required for release from the reset state
- The time required to load the firmware from the boot memory
- The time required for the firmware to make the settings related to fast linkup

In the sample software for fast linkup, in addition to the fast linkup settings made by "R_IN32_PHY_SETTING_T List", referred to by Table 3.2, reading of the serial flash memory in fast read mode is used to shorten the time required to load the firmware from the boot memory. To use the fast linkup processing in the sample software, select "SFlash Boot FastLinkUp" in the build configuration.

3.3 Status Display Function

3.3.1 Status Display by LEDs

R-IN32M4-CL3 can display the status of its own station and the status of the Ethernet PORT using LEDs.
From the viewpoint of ease of use by the end user, mounting all LEDs other than the User LEDs in the table below is recommended.

Mount the LEDs so that the LED lights are visible from the housing of the R-IN32M4-CL3 application product. The LED colors and shapes are not specified. Select the LEDs in accordance with user specifications.

Table 3.3 LED Status Display List

Type	LED Name		Description
Own station status display	RUN		Indicates the operating status.
		On	Operating normally.
		Off	A hardware failure or a watchdog timer error has occurred.
	RD		Displays the reception status of data.
		On	Receiving data.
		Off	Data not received.
	SD		Displays the sending status of data.
		On	Sending data
		Off	Data not sent.
	D LINK		Indicates the status of the data link.
		On	Data link in operation (cyclic transmission in progress)
		Off	Data link not performed (disconnected)
		Blinking	Data link in operation (cyclic transmission stopped)
	ERR.		Indicates the R-IN32M4-CL3 error status.
		On	Error in own station
		Off	Normal operation
	L ERR.		Indicates the error status of the received data and the line, and loopback status. When this LED is on, you can check the port that detected the error using the L ER LED.
		On	Abnormal data received or loopback in progress
		Off	Normal data received or loopback not performed
	User LED1, 2		Indicates a user-defined status.
Ethernet PORT1 status display	LINK	On	Link up
		Off	Link down
	L ER	On	Abnormal data received or loopback in progress
		Off	Normal data received or loopback not performed
Ethernet PORT2 status display	LINK	On	Link up
		Off	Link down
	L ER	On	Abnormal data received or loopback in progress
		Off	Normal data received or loopback not performed

3.3.2 Controlling the LEDs

Some LEDs are controlled by hardware and some LEDs are controlled by software.

The LEDs controlled by hardware are turned on/off by the CPU area or GbE-PHY area of R-IN32M4-CL3, in accordance with the status of the own station. These LEDs do not need to be controlled by software.

The LEDs controlled by software are turned on/off by R-IN32M4-CL3 driver interface functions called by the user program, in accordance with the status of the own station. (For R-IN32M4-CL3 driver interface functions, see Section 5.3.7 "LED Control".)

The following table lists the LED control sources and LED control at reset/error.

Table 3.4 LED Control List

Type	LED Name	R-IN32M4-CL3 Output Signal Name	H/S Classification	Control Source	Power-on Reset Note1	System Reset Note1	Internal WDT / External WDT / Own Station Error Note2
Own station status display	RUN	CCI_RUNLEDZ	H/S	CPU area or Driver	Off	Off	Off
	RD	CCI_RDLEDZ	H	CPU area	Off	—	—
	SD	CCI_SDLEDZ	H	CPU area	Off	—	—
	D LINK	CCI_DLINKLEDZ	H/S	CPU area or Driver	Off	Off	Off
	ERR.	CCI_ERRLEDZ	H/S	CPU area or Driver	Off	Off	On
	L ERR.	—	H/S	CPU area or Driver	—	—	—
	User LED1	RP20	S	Driver	Off	Off	Off
	User LED2	RP21	S	Driver	Off	Off	Off
Ethernet PORT1 status display	LINK	PHY0_LED0	H	GbE-PHY area	Off	Off	— Note3
	L ER	CCI_LERR0LEDZ	H/S	CPU area or Driver	Off	Off	Off
Ethernet PORT2 status display	LINK	PHY1_LED0	H	GbE-PHY	Off	Off	— Note3
	L ER	CCI_LERR1LEDZ	H/S	CPU area or Driver	Off	Off	Off

Remark: H: Controlled by hardware
S: Controlled by software
H/S: Controlled by hardware and software
Driver: R-IN32M4-CL3 driver interface functions, see Section 5.3.7 "LED Control".

Notes 1. For reset details, see Section 3.1 "Reset".
2. This is an error that occurs for user application reasons. For details, see Section 4.2.4 "Own Station Error Processing".
3. These LEDs turn on if the mode is bypass mode.

3.3.3 Controlling User LEDs

User LEDs can be freely defined in accordance with the specifications of the R-IN32M4-CL3 application product. For example, the on/off/blinking status of a User LED can be controlled to indicate the following:

- Status of online/offline mode (hardware test mode) of own station
- Normal/Error status of various tests such as the hardware test

Control the User LEDs using the R-IN32M4-CL3 driver interface functions "gerR_IN32_SetUSER1LED" and "gerR_IN32_Set USER2LED". For details of the R-IN32M4-CL3 driver interface functions, see Section 5.3.7 "LED Control".

3.3.4 Enabling/Disabling the LED Function

The function of the LEDs in the table below can be enabled and disabled.

Table 3.5 LEDs for which Function Can Be Enabled/Disabled

LED Name	Description
RUN	Operating status
ERR	Error status
D LINK	Data link status
User LED 1, User LED 2	User-defined status
L ER 1, L ER 2	Ethernet PORT 1, 2 reception data error status

Determine the LED function enable/disable specifications as necessary.

Remark: (Example)
Disable the L ER.LEDs of Ethernet PORT1 and PORT2 in a link down state since the LED sometimes stays ON when the link is down.

To disable the LED function, use the function "gerR_IN32_DisableLED".
To enable the LED function, use the function "gerR_IN32_EnableLED".
For details of the functions "gerR_IN32_DisableLED" and "gerR_IN32_EnableLED", see Section 5.3.7 "LED Control".

3.4 Bypass Mode

Bypass mode maintains network connection (linkup), even when the system resets or an error that affects communication occurs in a line or ring topology, so that communication with downstream stations from the own station is not affected. R-IN32M4-CL3 transitions its own station to bypass mode when the causes below occur.

[Causes for mode change to bypass mode]

- System reset (See Section 3.1 "Reset".)
- WDT error (internal WDT error or CCI_WDTIZ (multiplexed with P12) input)
- Own station error (An error that occurs for user application reasons. For details, see Section 4.2.4 "Own Station Error Processing".)

3.5 MIB Information

MIB information is information collected by R-IN32M4-CL3, such as the Ethernet PORT1 and PORT2 frame reception count and error frame reception count, used to manage the communication status. The user program uses MIB information to identify the communication status of the Ethernet ports of its own station. For MIB information details, see Section 4.2.13 "MIB Information Acquisition Processing".

3.6 Cyclic Transmission Function

The cyclic transmission function periodically exchanges data with the master station using link devices.

The status of each link device (RY and RWw) of the master station is stored in the link device (RY and RWw) of its own station by a link scan. The status of each link device (RX and RWr) of its own station is stored in the link device (RX and RWr) of the master station by a link scan.

The following illustrates the flow of cyclic data.

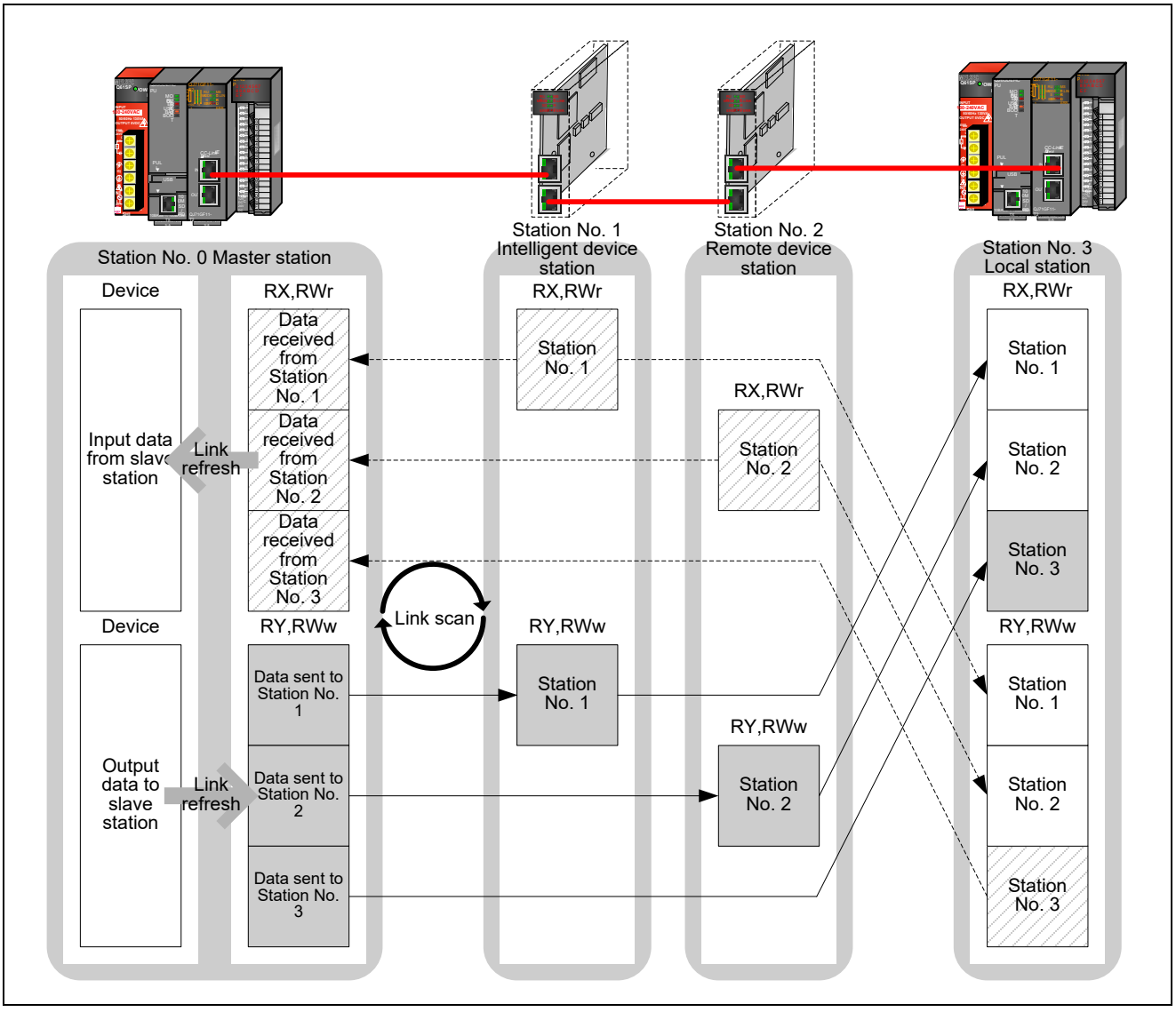


Figure 3.3 Flow of Cyclic Data

For cyclic transmission, see Section 4.2.7 "MyStatus from Master Station and Cyclic Receive Processing" for reception and Section 4.2.9 "Cyclic Send Processing" for send.

3.7 Transient Transmission Function

Transient transmission communicates data when there is a communication request from another station or its own station[Note]. The function directly accesses the device/buffer memory of the other station and communicates the data.

Note. When a read/write instruction is executed by the engineering tool (peripheral device) of the R-IN32M4-CL3 application product

Transient transmission achieves send/received easier than cyclic transmission in the following cases:

- When reading and writing a large volume of data that exceeds the number of own/other station link device points
- When there is no send/reception area for general-purpose data (error history, parameter setting values, etc.) in the own/other station link device

The following illustrates the flow of transient data with a read instruction.

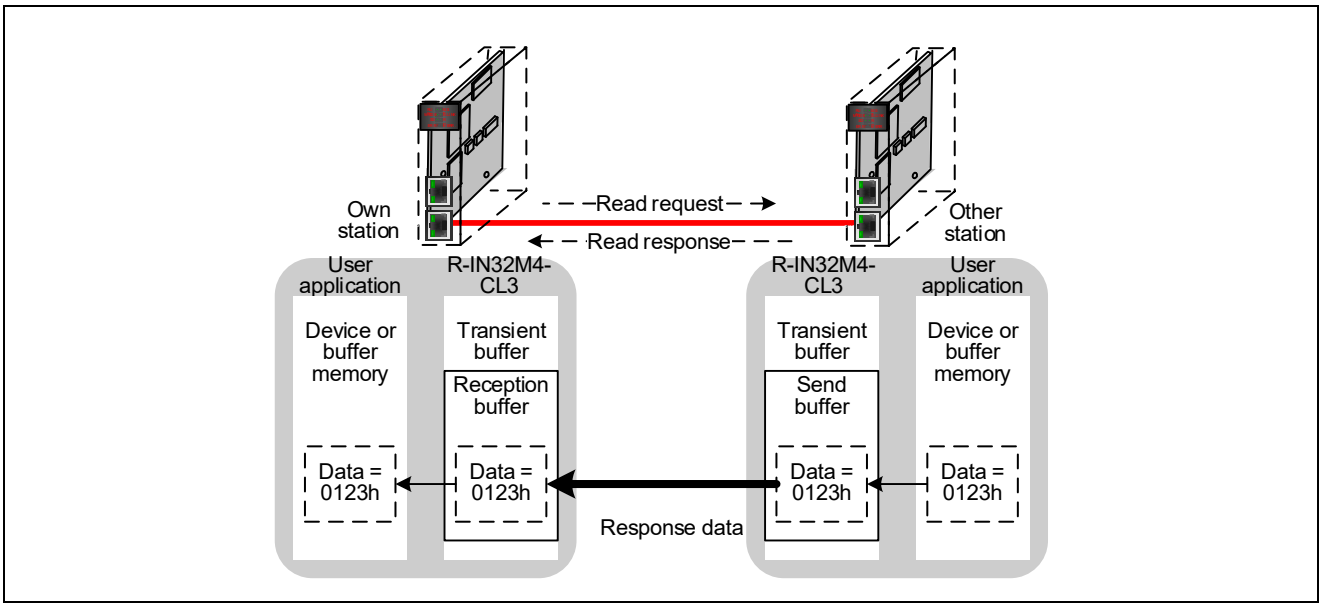


Figure 3.4 Flow of Transient Data

For transient transmission processing details, see Section 4.3 "User Program Details (Transient Transmission Related)" and subsequent sections.

Caution: To realize the transient transmission function, set "blTransientReceiveEnable" (transient reception function) in initial settings to "R_IN32_TRUE".
(See No. 6 in Table 5.7 "R_IN32_UNITINIT_T List" in Section 5.3.1 (2) "gerR_IN32_Initialize".)

3.7.1 Transient Transmission Client and Server Functions

Transient transmission includes a client function and server function.

The client function sends transient requests to stations with a server function.

The server function sends transient responses to transient requests from stations with a client function.

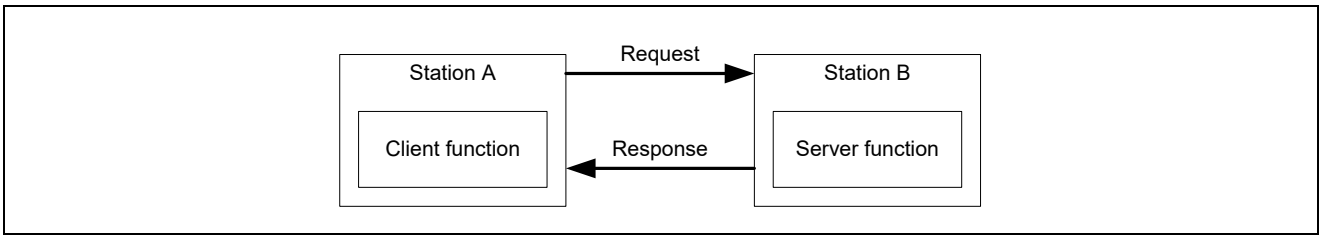


Figure 3.5 Transient Client/Server Function

The intelligent device station can implement a client function and a server function.

The remote device station can implement a server function only.

For a processing overview of client and server functions, see Section 4.3 "User Program Details (Transient Transmission Related)" Section 4.3.1 "Transient Transmission Processing Overview".

3.7.2 Transient Transmission Frames

The following table lists the frames of transient transmission supported by R-IN32M4-CL3 application products, and indicates whether the send/receive processing for each frame needs to be implemented.

Table 3.6 Transient Transmission Frame List and Need for Implementation

No.	Frame Name[Note1]	Frame Type (FType)		Data Type (DataType)		Data Sub Type (DataSubType)		Implementation
1	CC-Link IE Field specific transient	22h	Transient1	07h	CC-Link IE Field specific	0002h	System specific	Required
2	SLMP	22h	Transient1	05h	Network common	0002h	SLMP	Optional
3	CC-Link compatible transient	25h	Transient2	04h	CC-Link compatible	—	—	Optional
4	TransientAck	23h	TransientAck	Note2	Note2	Note2	Note2	Required

Notes 1. In this manual, each frame is described using the above names.

2. Extracted from the received frame.

1. The CC-Link IE Field specific transient frame is used by the master station to collect slave station information and manage the network.
2. The SLMP frame is used by extension functions (CC-Link IE Field Network diagnostics, parameter processing/command execution of slave stations, etc.) that use the engineering tool.
3. The CC-Link compatible transient frame is mainly used in communication between user products. The frames are compatible with CC-Link transient frames.
4. TransientAck is used to issue verification responses to the send source when Transient1 and Transient2 frames are received.

3.7.3 Transient Transmission Commands

The following table lists the transient transmission commands supported by R-IN32M4-CL3 application products, and indicates whether or not implementation by the client or server function is required for each command. The remote device station does not require the client function of the commands in the table below.

Table 3.7 Transient Transmission Command List and Need for Implementation

Frame Name	Command Type	ID Station		RD Station	Remarks
		Client Function	Server Function	Server Function	
CC-Link IE Field specific transient	Node information distribution	×	△	×	Note1, 6
	Statistics information acquirement	×	△	△	Note2, 6
	Detailed node information acquisition	×	◎	◎	Note3, 6
	Option information acquisition	×	○	○	Note4, 6
SLMP	Selected station information acquisition	×	○	○	Note5, 6, 7
	Communication test	×	○	○	
	Cable test	×	○	○	
	Remote reset	×	△	△	
	Memory read	△	△	△	—
	Memory write	△	△	△	—
CC-Link compatible transient	Memory access information acquisition	△	△	△	Note8
	RUN	△	△	△	—
	STOP	△	△	△	—
	Memory read	△	△	△	RIRD equivalent
	Memory write	△	△	△	RIWT equivalent

Remark: ◎: Required ○: Recommended △: Optional x: Not required
ID: Intelligent Device RD: Remote Device

Notes 1. Node information distribution is a command by which the master station distributes information indicating station number and MAC address correspondence by multicast. Node information is used when an intelligent device station sends a transient request to another station. Note that a TransientAck and response are not required for this command. Only processing for receiving the distributed node information is required.

2. Statistical information acquisition is a command by which the master station collects error information related to Ethernet PORT1 and PORT2 of the slave station.

3. Detailed node information acquisition is a command by which the master station collects information specific to the slave station. For detailed node information, see Table 5.6 "R_IN32_UNITINFO_T List".

4. Option information acquisition is a command by which the master station confirms the presence or non-presence of slave station options. Option information is information indicating the support of extension functions of CC-Link IE Field Network, such as SLMP frame transmission/reception and CC-Link IE Field Network diagnostics.

5. See Section 3.9 "CC-Link IE Field Network Diagnostics".

6. The R-IN32M4-CL3 driver interface functions (Section 5.3.11 "Transient Request Reception") perform receive processing.

7. Required for CC-Link IE Field Network diagnostics.

8. Required when access codes are used

3.8 MyStatus Send/Receive Function

The MyStatus send/receive function is used by R-IN32M4-CL3 to periodically send and receive MyStatus frames. MyStatus is used to report the status of stations connected to the network. Sending and receiving MyStatus makes it possible to report the status of the own station to the master station and monitor the status of the master station.

3.8.1 Sending MyStatus

Setting the information (own station status) outlined in the table below in the arguments of the R-IN32M4-CL3 driver interface function "Own station status setting" (Section 5.3.5 (1) "gerR_IN32_SetNodeStatus") in the user program "UserSendMyStatus" (Section 4.2.8 "MyStatus Send Processing") makes it possible to report the status of the own station to the master station.

Table 3.8 Information Related to Sending MyStatus

No.	Item	Description
1	Detailed application operation status	Stores the operation status of the user application. 0000h:Detailed application operation status notification not supported 0001h:Application stopped 0002h:Application running 0003h:Application does not exist Other than the above: Not used
2	Detailed application error status	Stores the error status when a user application error occurs. 0000h:No error 0001h:Minor error 0002h:Moderate error 0003h:Major error Other than the above:Not used

3.8.2 Receiving MyStatus

The status of the master station can be monitored by acquiring the information (master station status) in the table below from the address specified in the R-IN32M4-CL3 driver interface function "Master station status acquisition" (Section 5.3.4 (4) "gerR_IN32_GetMasterNodeStatus") in the user program "UserReceiveCyclic" (Section 4.2.7 "MyStatus from Master Station and Cyclic Receive Processing").

Table 3.9 Information Related to Receiving MyStatus

No.	Item	Description
1	State of operation of the master station application	Indicates the state of operation of the master station application. 0b: Application stopped 1h: Application running
2	Error state of the master station application	Indicates the error state of the master station application. 0h: No error 1h: Error occurred

Remark: For a Mitsubishi Electric master station, the state of the master station application is based on the state of the sequencer CPU unit as described below.

1. State of operation of master station application

[Application stopped]

The sequencer program has stopped operating.

That is, the setting of the RUN/STOP switch is to "STOP" the operation, or operation is in the midst of a moderately serious or major error.

[Application running]

The sequencer program is operating.

That is, the setting of the RUN/STOP switch is to "RUN" the operation.

2. Error state of master station application

[No error]

No error has occurred, or a minor error such as a battery error has occurred with the CPU unit such that operation has continued.

[Error occurred]

An error of the CPU unit that was serious enough to stop the operation has occurred. Such errors include moderately serious errors such as WDT errors and major errors such as hardware failures.

3.9 CC-Link IE Field Network Diagnostics

The CC-Link IE Field Network diagnostics graphically displays the status of CC-Link IE Field Network using the engineering tool. Error locations, error causes, corrective actions, and event history can be checked using the engineering tool. For function details, refer to the user's manual of the master/local module.

This function displays the R-IN32M4-CL3 application product on the CC-Link IE Field Network diagnostics window by responding to SLMP frame requests from the master station. The function also allows you to execute various tests and operations.

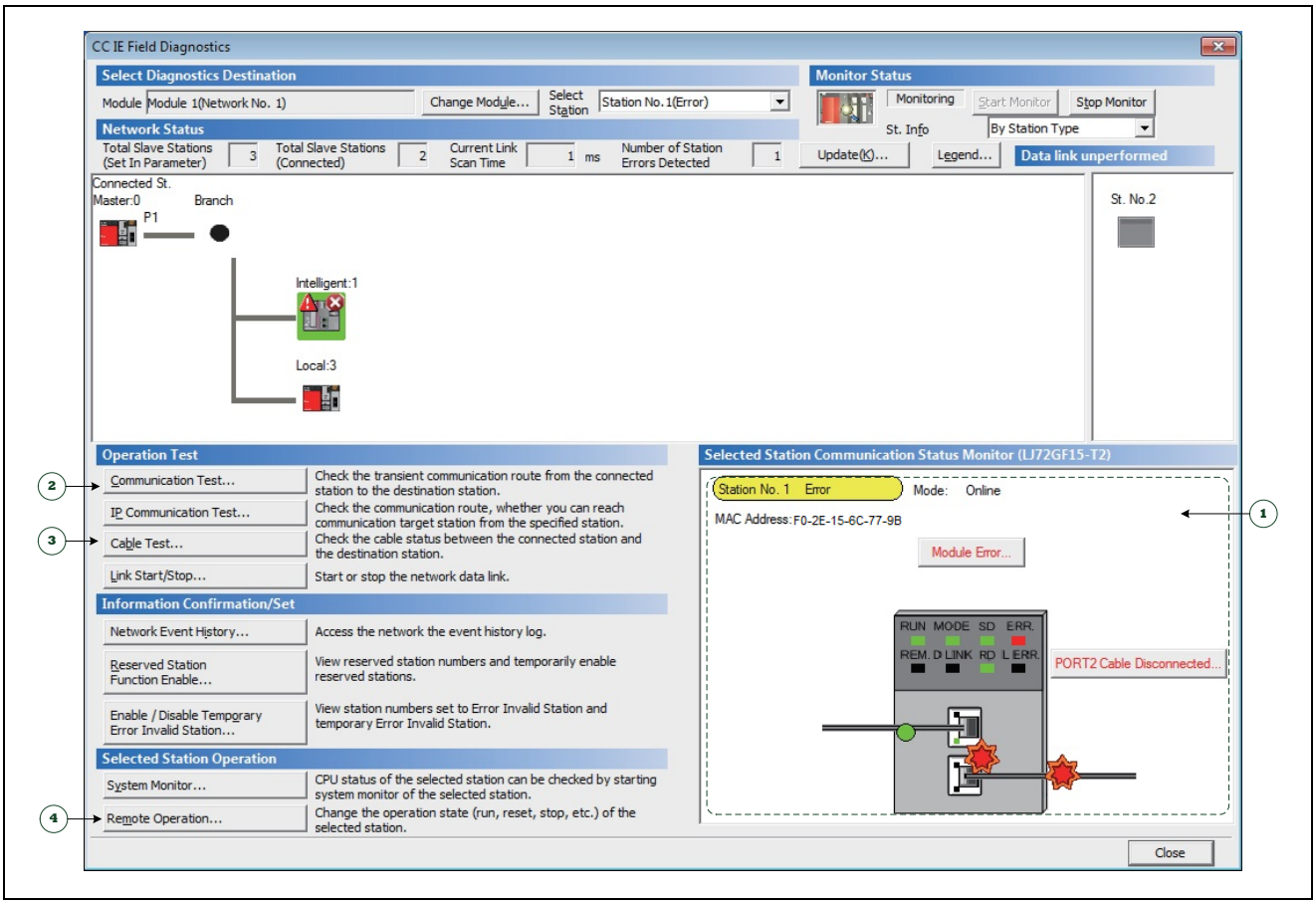


Figure 3.6 Diagnostic Window/Operation Locations (GX Works2)

Table 3.10 Diagnostic Window/Operation Locations and SLMP Requests

No.	Item	Description	SLMP Request Frame (Command)
1	Selected station communication status monitor	Displays the status of the selected station and error details.	Selected station communication status request (3119h)
2	Communication test	Tests the communication path of transient transmission from the own station to the communication destination.	Communication test request (3040h)
3	Cable test	Tests cable disconnection and no connection.	Cable test request (3050h)
4	Remote operation	Resets the status of the R-IN32M4-CL3 application product without hardware switch operation.	Remote reset request (1006h)

[SLMP request frame response]

In the user program "UserHandleReceivedTransient1" (Section 4.3.3 "Transient1 Receive Data Processing"), the R-IN32M4-CL3 driver interface functions ((5) to (8) in the Section 5.3.11 "Transient Request Reception") performs the applicable SLMP frame response processing (request frame receive processing).

The processing of the above Nos. 1 to 4 is described in the sample code. (Implementation of the above Nos. 1 to 3 is recommended.)

3.9.1 Selected Station Communication Status Monitor LEDs

The LED status of the own station can be displayed on the selected station communication status monitor by creating LED information in "UserHandleReceivedTransient1" (Section 4.3.3 "Transient1 Receive Data Processing") and issuing a response to the selected station communication status request.

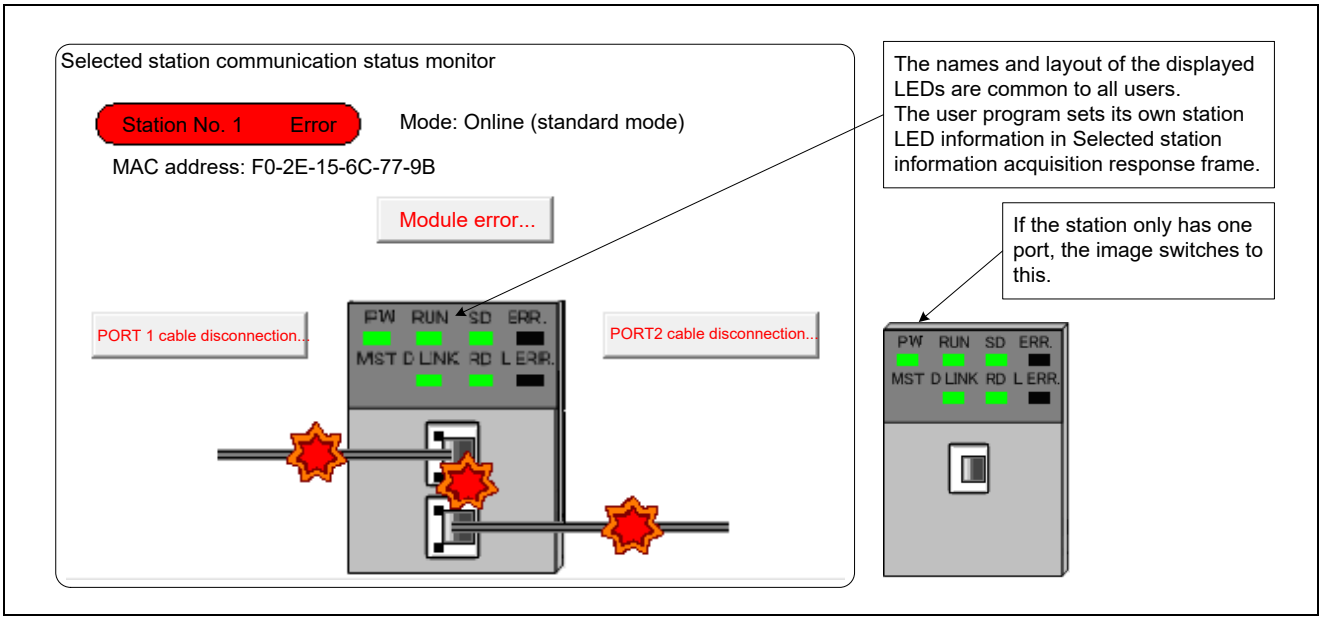


Figure 3.7 Display Example of Selected Station Communication Status Monitor

[Example of LED Use of Selected Station Communication Status Monitor]

When the LED status of the R-IN32M4-CL3 application product is not visible during end user troubleshooting, the LED status can be checked by using CC-Link IE Field Network diagnostics.

[Displayable LEDs]

The LED names[Note] and LED layout that can be displayed on the selected station communication status monitor are as shown in the figure above.

Note. PW, RUN, SD, ERR., MST (not used; grayed out), D LINK, RD, L ERR.

For details on creating LED information, see Section 4.3.3 (3) "Creating LED Information".

4. CREATING THE USER PROGRAM

This section provides an overview of user program processing and customization.

The user program is sample processing for verifying the logic of the communication process of an intelligent device station or remote device station. Customize the program in accordance with user specifications.

4.1 User Program List

The following is a list of the user programs available in the "¥cie_intelligent_device¥root¥Japanese¥sample" or "¥cie_remote_device¥root¥Japanese¥sample" folders. This list also indicates whether or not implementation in the intelligent device station (ID) or remote device station (RD) is required for each program.

**Caution: The R_IN32M4_Transient.c file describes Transient2 memory reading/writing and SLMP memory reading/writing as sample processing of each command.
If you want to implement commands other than the above, add the processing for the command while referring to Section 7 "FRAME FORMAT" and the related manual "SLMP Reference Manual" (SH(NA)-080956ENG)).**

Table 4.1 User Program List (Initial/Cyclic Transmission Related)

(1/2)

No.	File	Function Name	Overview	Implementation Required	
				ID	RD
1	R_IN32M4_sample.c	iUserMainRoutine	Main processing	O	
2		iUserInitialization	Initialization processing	O	
3		iUserStart	Communication start processing	O	
4		UserForceStop	Own station error processing	△	
5		UserStopCyclic	Cyclic transmission stop processing	△	
6		iUserExecuteMain	Event processing	O	
7		UserReceiveCyclic	MyStatus from master station and cyclic receive processing	O	
8		UserSendMyStatus	MyStatus send processing	O	
9		UserSendCyclic	Cyclic send processing	O	
10		UserUpdateStatus	Communication status update processing	O	
11		UserUpdateLed	LED update processing	O	
12		UserGetCyclicStatus	Cyclic transmission status update processing	△	
13		UserGetMIB	MIB information acquisition processing	△	

Remark: O: Required; △: Optional; ×: Not required

Table 4.2 User Program List (Transient Transmission Related)

(2/2)

No.	File	Function Name	Overview	Implementation Required	
				ID	RD
1	R_IN32M4_ Transient.c	UserReceiveTransient	Transient1, Transient2, and TransientAck receive processing	O	
2		UserHandleReceivedTransient1	Transient1 receive data processing	O	
3		UserStartMakingReceivedTransient1	Transient1 receive data reconstruction start processing	O	
4		blUserMakeReceivedTransient1	Transient1 receive data reconstruction processing	O	
5		erUserHandleReceivedMemRead Request	SLMP memory read request frame receive processing	△	
6		erUserHandleReceivedMemWrite Request	SLMP memory write request frame receive processing	△	
7		UserHandleReceivedTransient2	Transient2 receive data processing	△	
8		erUserCheckReceivedTransient2	Transient2 receive data check processing	△	
9		erUserHandleReceivedTransient2 _RequestSetMemory	Transient2 memory write request receive processing	△	
10		ulUserSetTransient2_Response	Transient2 response frame creation processing	△	
11		UserHandleReceivedTransientAck	TransientAck receive data processing	O	
12		blUserSetTransientAck	TransientAck frame creation processing	O	
13		UserSendTransient	Transient1, Transient2, and TransientAck send processing	O	
14		erUserSetSlmpMemRead_Request	SLMP memory read request frame creation processing	△	×
15		pvUserJudgeTransient1Divide	Transient1 request send division determination processing	△	×
16		erUserSetTransient1_Requeset	Transient1 request frame creation processing	△	×
17		erUserHandleReceivedMemRead Response	SLMP memory read response receive processing	△	×
18		UserSetTransient2_Request	Transient2 request frame creation processing	△	×
19		ulUserSetTransient2_ RequestGetMemory	Transient2 memory read request frame creation processing	△	×
20		erUserHandleReceivedTransient2_ ResponseGetMemory	Transient2 memory read response receive processing	△	×

Remark: O: Required; △: Optional; ×: Not required

Table 4.3 User Program List (Hardware Test Related)

No.	File	Function Name	Overview	Implementation Required	
				ID	RD
1	R_IN32M4_HWTest.c	UserIEEETest	Hardware test (IEEE802.3ab compliance test)	O	
2		UserLoopBackTest	Hardware test (loopback communication test)	△	

Remark: O: Required; △: Optional; ×: Not required

4.2 User Program Details (Initial/Cyclic Transmission Related)

4.2.1 Main Processing

The following shows the general flow of the main processing.

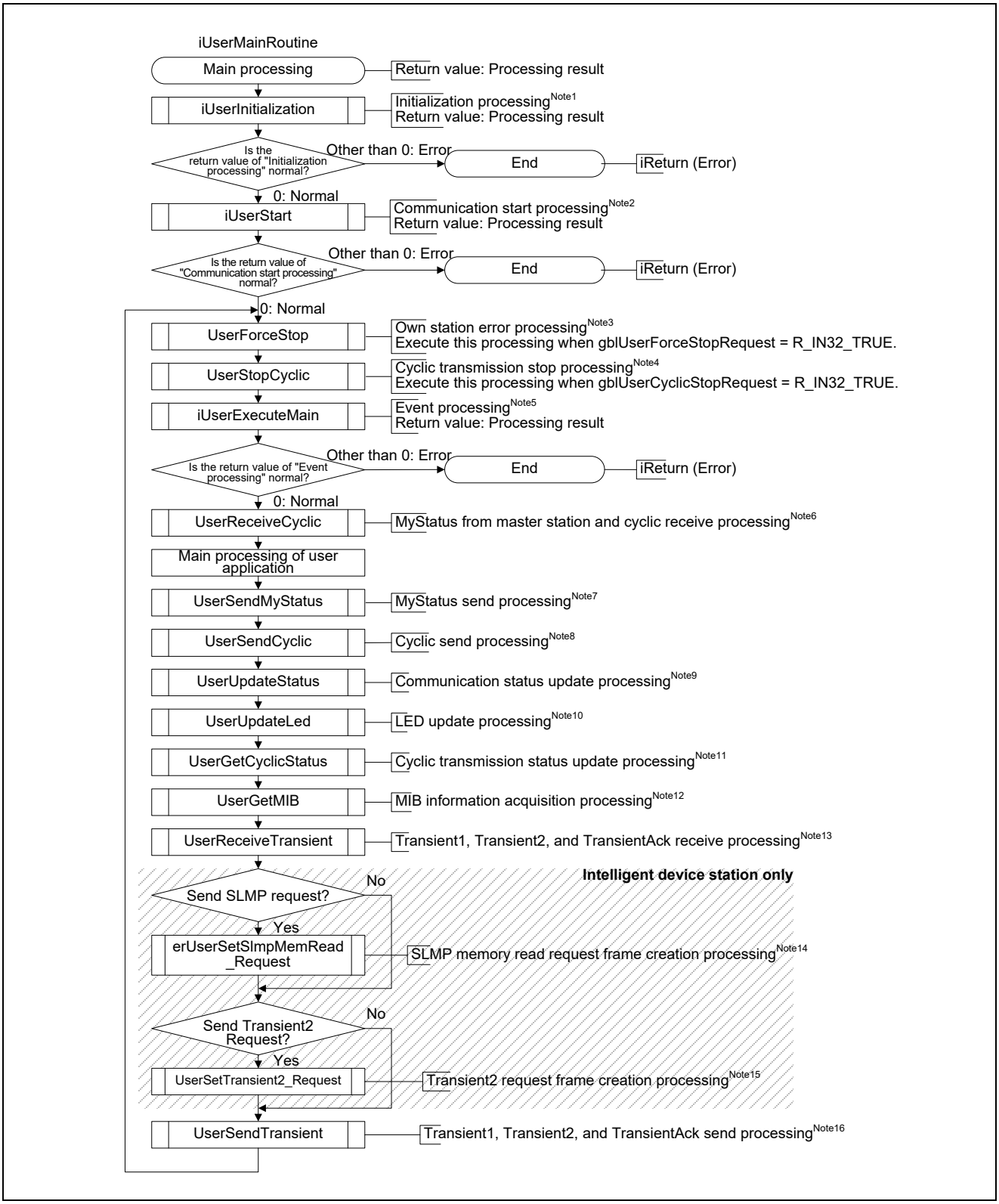


Figure 4.1 Flowchart for Main Processing

Notes 1. For details, see Section 4.2.2 "Initialization Processing".
2. For details, see Section 4.2.3 "Communication Start Processing".
3. For details, see Section 4.2.4 "Own Station Error Processing".
4. For details, see Section 4.2.5 "Cyclic Transmission Stop Processing".
5. For details, see Section 4.2.6 "Event Processing".
6. For details, see Section 4.2.7 "MyStatus from Master Station and Cyclic Receive Processing".
7. For details, see Section 4.2.8 "MyStatus Send Processing".
8. For details, see Section 4.2.9 "Cyclic Send Processing".
9. For details, see Section 4.2.10 "Communication Status Update Processing".
10. For details, see Section 4.2.11 "LED Update Processing".
11. For details, see Section 4.2.12 "Cyclic Transmission Status Update Processing".
12. For details, see Section 4.2.13 "MIB Information Acquisition Processing".
13. For details, see Section 4.3.2 "Transient1, Transient2, and TransientAck Receive Processing".
14. For details, see Section 4.3.15 "SLMP Memory Read Request Frame Creation Processing".
15. For details, see Section 4.3.19 "Transient2 Request Frame Creation Processing".
16. For details, see Section 4.3.14 "Transient1, Transient2, and TransientAck Send Processing".

4.2.2 Initialization Processing

This function initializes R-IN32M4-CL3, enables and disables the R-IN32M4-CL3 internal WDT, and sets the station number and network number.

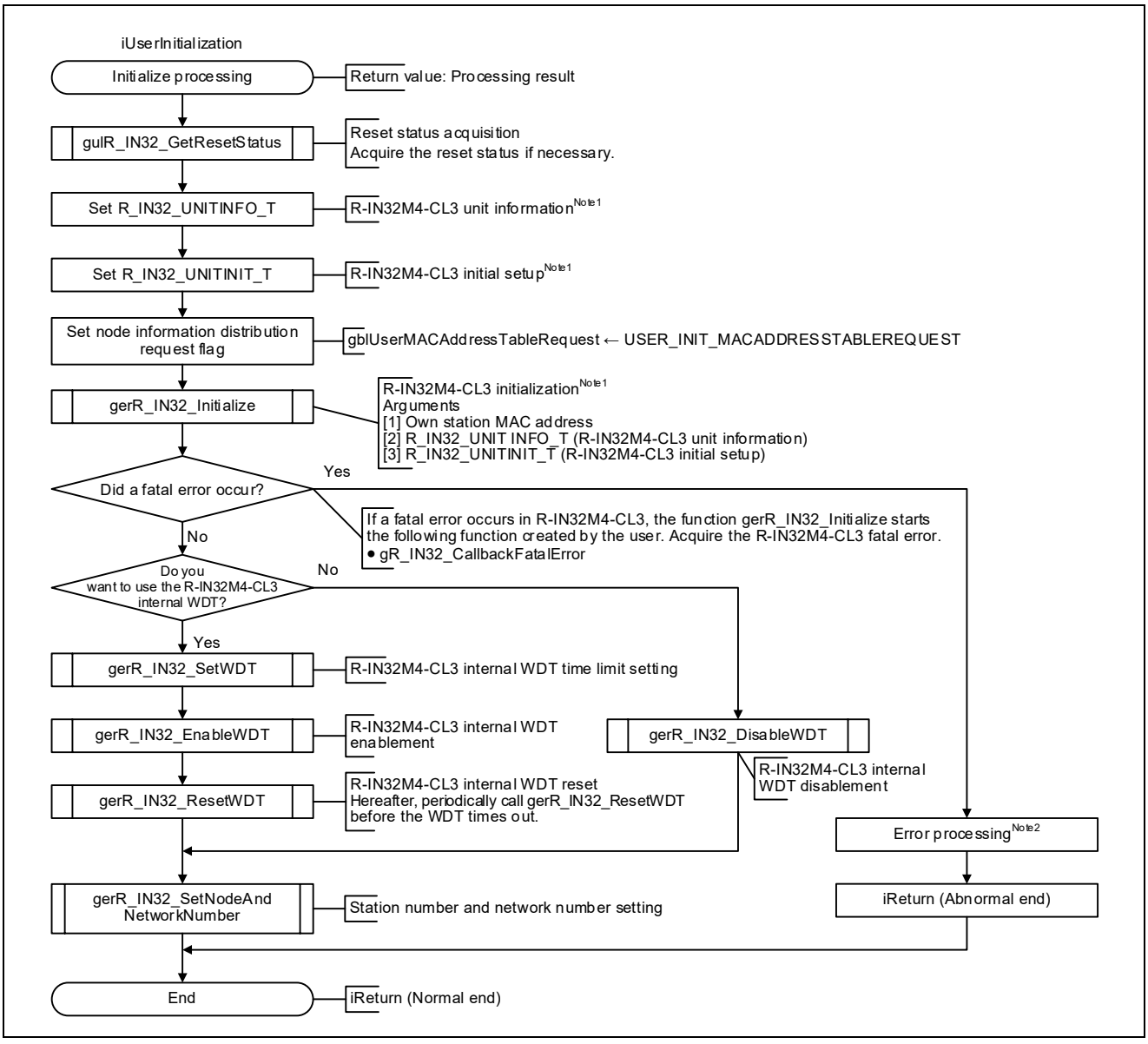


Figure 4.2 Flowchart for Initialization Processing

Notes 1. For details, see Section 5.3.1(2) "gerR_IN32_Initialize".

2. For example, add processing such as calling UserForceStop (Own station error processing), and setting its own station to bypass mode.

[About gblUserMACAddressTableRequest]

"gblUserMACAddressTableRequest" (node information distribution request flag) is used to determine whether or not the own station receives Node information distribution frame is to be received. (See Section 4.3.3 "Transient1 Receive Data Processing".)

- When own station wants to receive node information (when own station wants to send a transient request)
  Set both "blMACAddressTableRequest" (initial value of node information distribution request) and "gblUserMACAddressTableRequest" (node information distribution request flag) of R_IN32_UNITINIT_T to "R_IN32_TRUE".
- When own station does not want to receive node information (when own station does not want to send a transient request)
  Set both "blMACAddressTableRequest" (initial value of node information distribution request) and "gblUserMACAddressTableRequest" (node information distribution request flag) of R_IN32_UNITINIT_T to "R_IN32_FALSE".

4.2.3 Communication Start Processing

This function instructs R-IN32M4-CL3 to start communication.

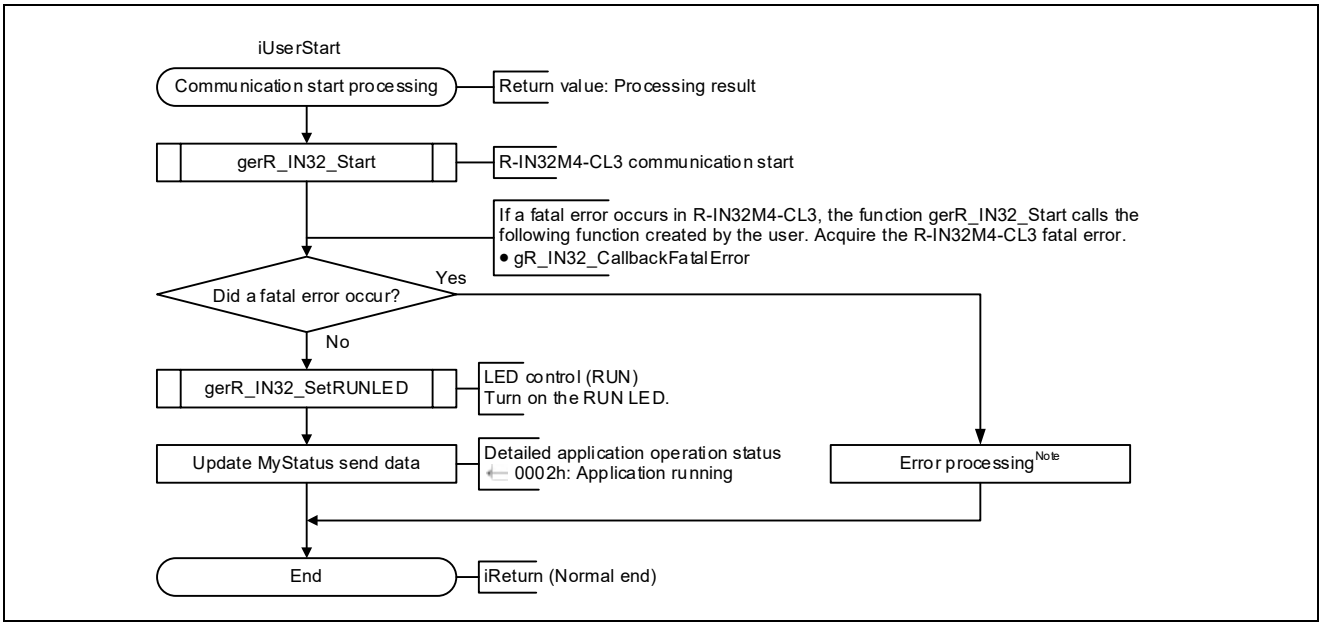


Figure 4.3 Flowchart for Communication Start Processing

Note. For example, add processing such as calling UserForceStop (Own station error processing), and setting its own station to bypass mode.

4.2.4 Own Station Error Processing

This function changes the state of its own station to an error when a (user defined) error occurs in the user program. (This processing is optional.)

When "gerR_IN32_ForceStop" is called, an R-IN32M4-CL3 own station enters an error state and the R-IN32M4-CL3 changes to bypass mode. In bypass mode, communication frames that have entered the port are not received by R-IN32M4-CL3 but are forwarded as is to another port.

When the error on its own station is cleared, the system must be reset by power-on reset or system reset.

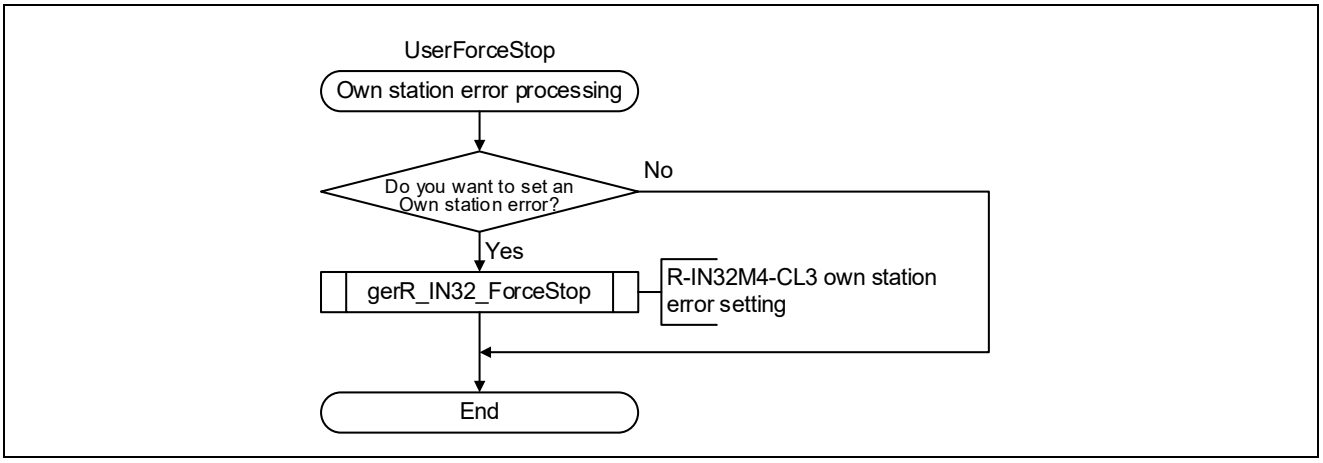


Figure 4.4 Flowchart for Own Station Error Processing

4.2.5 Cyclic Transmission Stop Processing

This function allows you to control the stopping and restarting of cyclic transmission for user application-side reasons. (This processing is optional.)

Even if you stop cyclic transmission, transient transmission is possible (token passing continues).

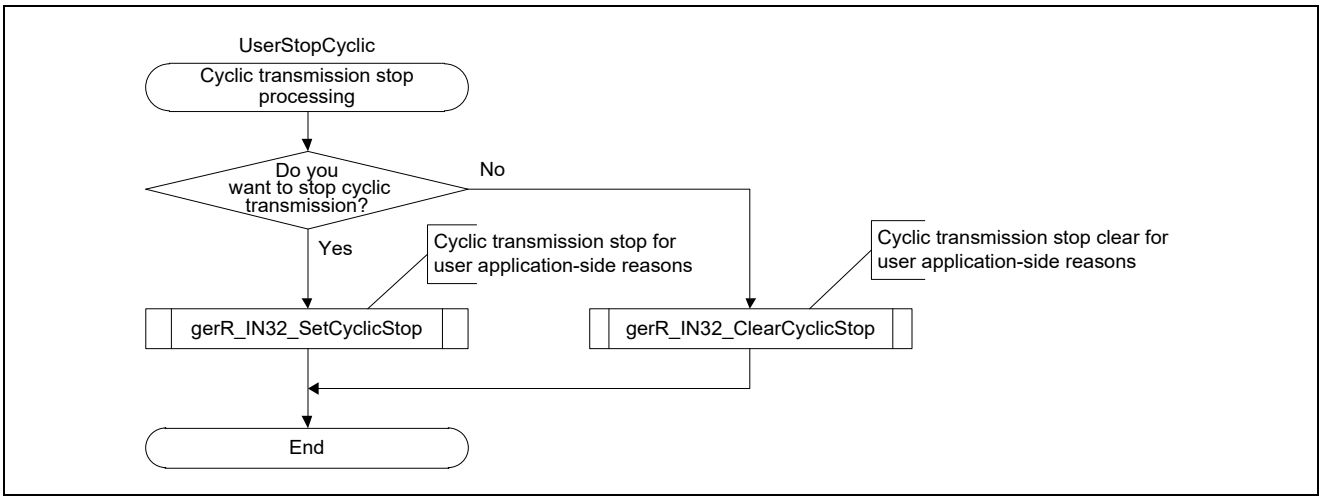


Figure 4.5 Flowchart for Cyclic Transmission Stop Processing

4.2.6 Event Processing

This function detects R-IN32M4-CL3 events (interrupts causes), processes the events, and updates MIB information.

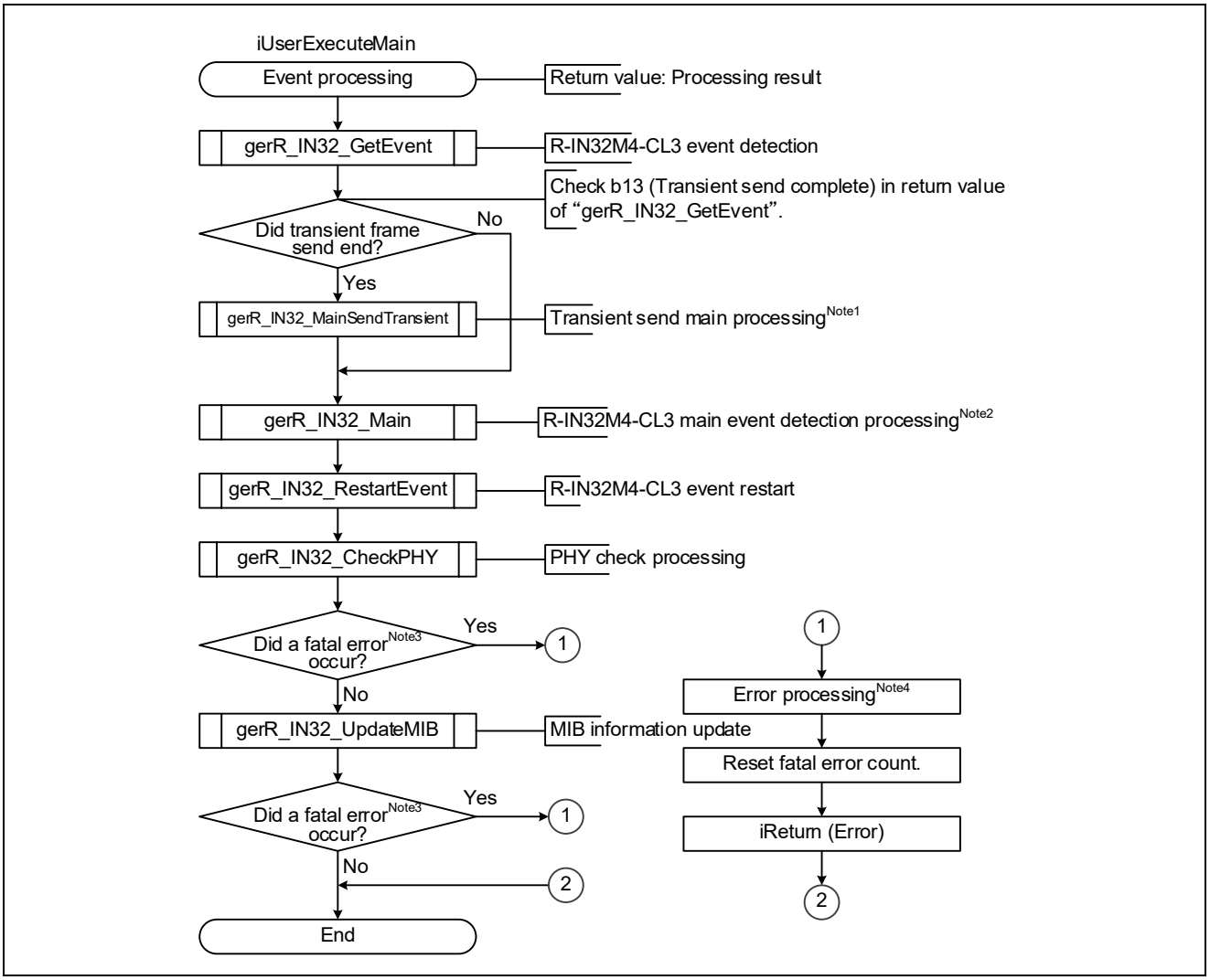


Figure 4.6 Flowchart for Event Processing

Notes 1. This function calls the following function created by the user to report the target send descriptor status (send result).
- **gerR_IN32_CallbackTransientSendingComplete**

2. When a command is received from the master station, the function gerR_IN32_Main calls the following function created by the user. Be sure to acquire the command.
- **gerR_IN32_CallbackCommandFromMaster**

3. If a fatal error occurs in R-IN32M4-CL3, the function gerR_IN32_UpdateMIB calls the following function created by the user. Be sure to acquire the R-IN32M4-CL3 fatal error.
- **gR_IN32_CallbackFatalError**

4. For example, add processing such as calling UserForceStop (Own station error processing), and setting its own station to bypass mode.

4.2.7 MyStatus from Master Station and Cyclic Receive Processing

This function acquires the status of the master station from the received MyStatus frame and acquires cyclic data from the received cyclic frame. Perform "hold/clear processing" in accordance with the status of the master station that is acquired from the MyStatus frame (in accordance with whether the master station is stopped, an error occurred, or the like).

In an R-IN32M4-CL3 application product that is controlling output to an external device, "hold/clear processing" refers to processing in which the R-IN32M4-CL3 application product is made to continue (hold) or stop (clear) the output, when cyclic transmission has stopped for reasons such as data link disconnection, master station shutdown, or a master station error.

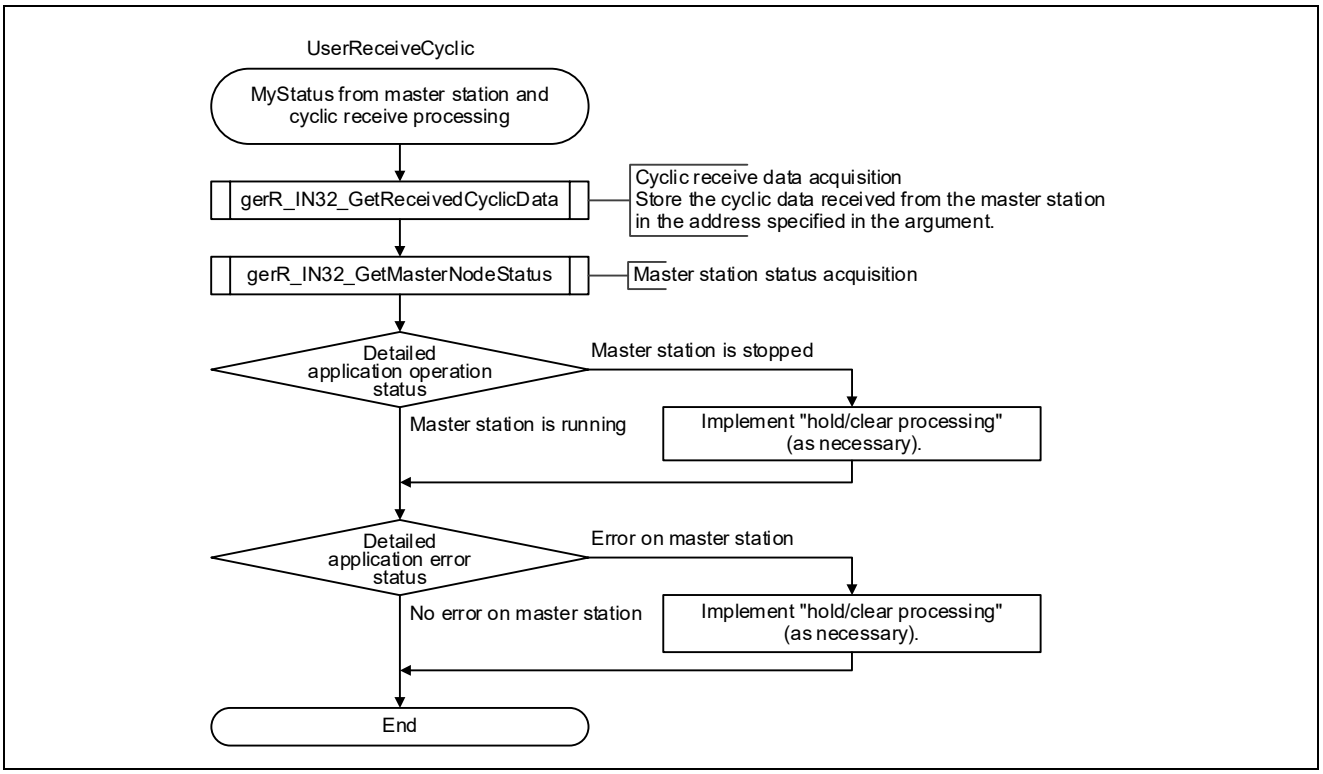


Figure 4.7 Flowchart for MyStatus from Master Station and Cyclic Receive Processing

Caution: Taking the cautionary notes (1) and (2) below into consideration, decide upon the implementation of hold or clear processing as a failsafe.

(1) Cyclic data transmitted by the master station

When the master station application is shut down or has an error, the transmission of cyclic data by the master station is held or cleared according to the relevant setting of the master station.

For a master station from Mitsubishi Electric, the hold/clear processing is set in "Output state setting on CPU module STOP" or "Output state setting on CPU stop error".

The slave station (own station) cannot detect whether the transmission of cyclic data by the master station is to be held or cleared in advance.

(2) Cyclic reception data acquired by the R-IN32M4-CL3 driver according to the state of the master station application

This note applies when the slave station (own station) is using the R-IN32M4-CL3 driver (gerR_IN32_GetReceivedCyclicData) to acquire cyclic reception data.

The acquired cyclic data differs according to whether the the master station is operating or has an error.

Master Station Application		Cyclic Reception Data Acquired by the R-IN32M4-CL3 Driver
State of Operation	Error State	
Running	No error	The cyclic data that is currently being transmitted by the master station is acquired.
Stopped	No error	
Running	Error occurred Note	Not acquired The cyclic data before the master station application had an error remains at the address specified by the argument.
Stopped	Error occurred	

Note. For the sequencer CPU unit of a master station from Mitsubishi Electric, there is no "running with an error" state.

4.2.8 MyStatus Send Processing

This function sets the MyStatus frame. The set frame is automatically sent by R-IN32M4-CL3.

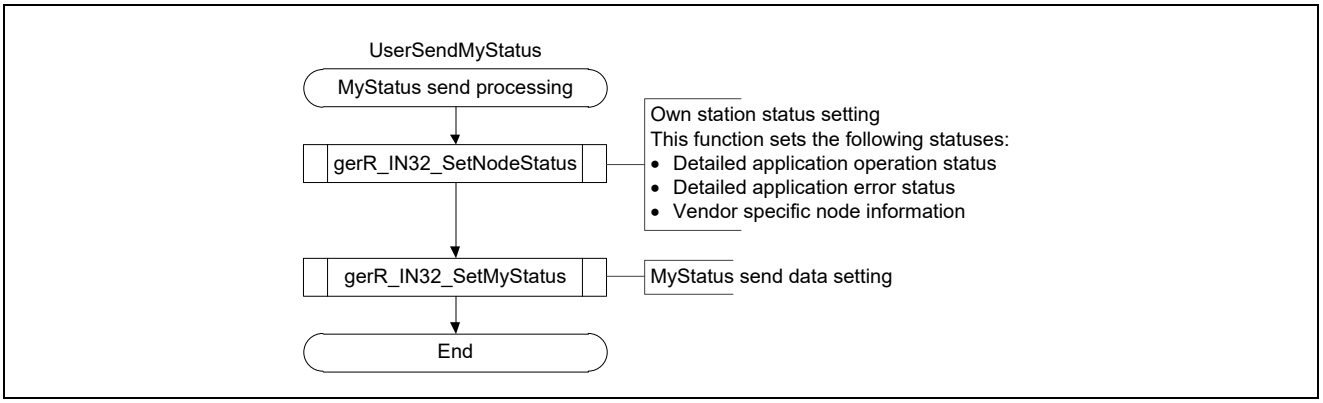


Figure 4.8 Flowchart for MyStatus Send Processing

4.2.9 Cyclic Send Processing

This function sends cyclic send data (RX and RWr).

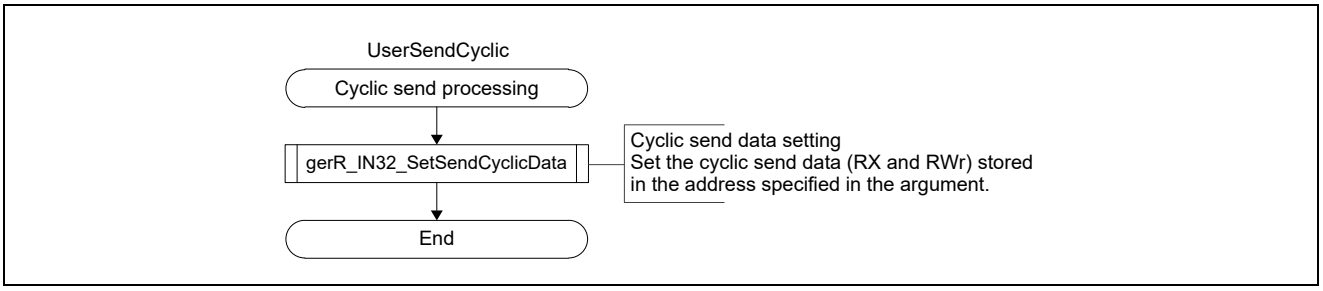


Figure 4.9 Flowchart for Cyclic Send Processing

4.2.10 Communication Status Update Processing

This function acquires the data link status of its own station, and sets the hold/clear processing and the ERR. LED control flag in accordance with the data link status.

The ERR. LED control flag is used to control the ERR. LED in the LED update processing. (See Section 4.2.11 "LED Update Processing".)

In an R-IN32M4-CL3 application product that is controlling output to an external device, "hold/clear processing" refers to processing in which the R-IN32M4-CL3 application product is made to continue (hold) or stop (clear) the output, when cyclic transmission has stopped for reasons such as data link disconnection, master station shutdown, or a master station error.

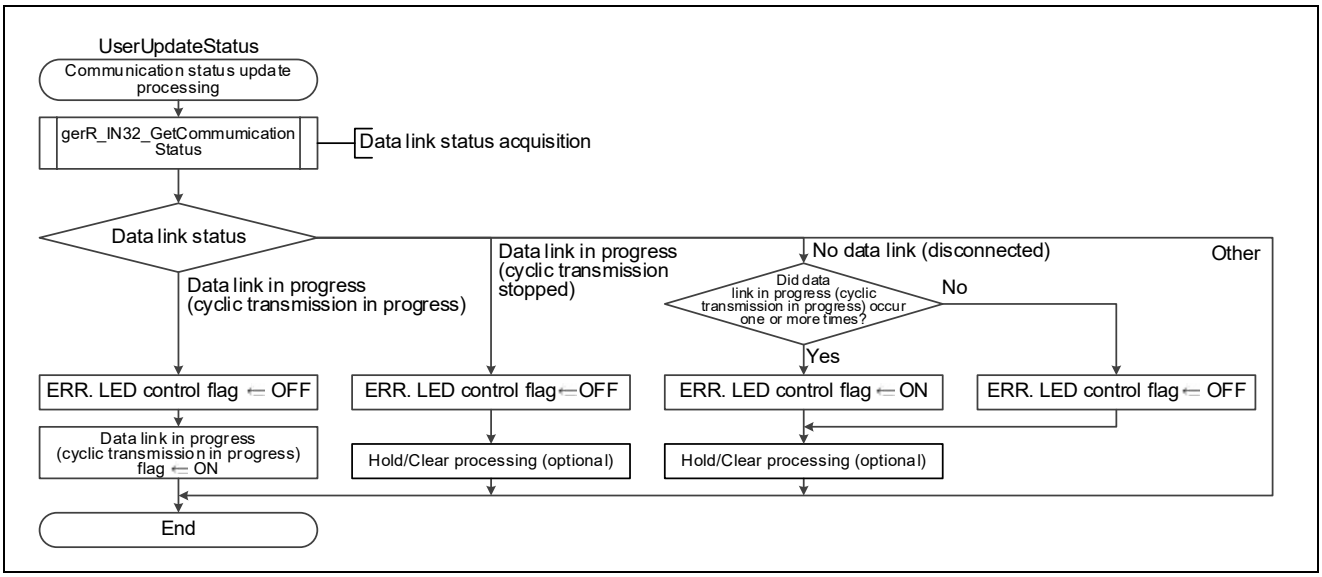


Figure 4.10Flowchart for Communication Status Update Processing

Caution: Taking the cautionary note below into consideration, decide upon the implementation of hold or clear processing as a failsafe.

Cyclic reception data acquired by the R-IN32M4-CL3 driver in accordance with the data link status

This note applies when the slave station (own station) is using the R-IN32M4-CL3 driver (gerR_IN32_GetReceivedCyclicData) to acquire cyclic reception data.

The acquired cyclic data depends on the state of the data link.

Data Link State	Cyclic Reception Data Acquired by the R-IN32M4-CL3 Driver
Data link is not set up (Own station is disconnected)	Not acquired The cyclic data before the master station application had an error remains at the address specified by the argument.
Data link is present (Own station has stopped cyclic transmission) Note	The cyclic data that is currently being transmitted by the master station is acquired.

Note. The slave station receives RY and RWw, and does not transmit RX or RWr.

4.2.11 LED Update Processing

This function controls the on, off, and blinking state of the D LINK LED and L ER LED in accordance with the data link status of its own station by calling the R-IN32M4-CL3 driver interface function. Additionally, this function controls the on and off state of the ERR. LED if needed by the user application, in accordance with the ERR. LED control flag status. Note that the ERR. LED control flag is set in the Communication status update processing (see Section 4.2.10 "Communication Status Update Processing "), but the value of the flag may be changed if needed by the user application.

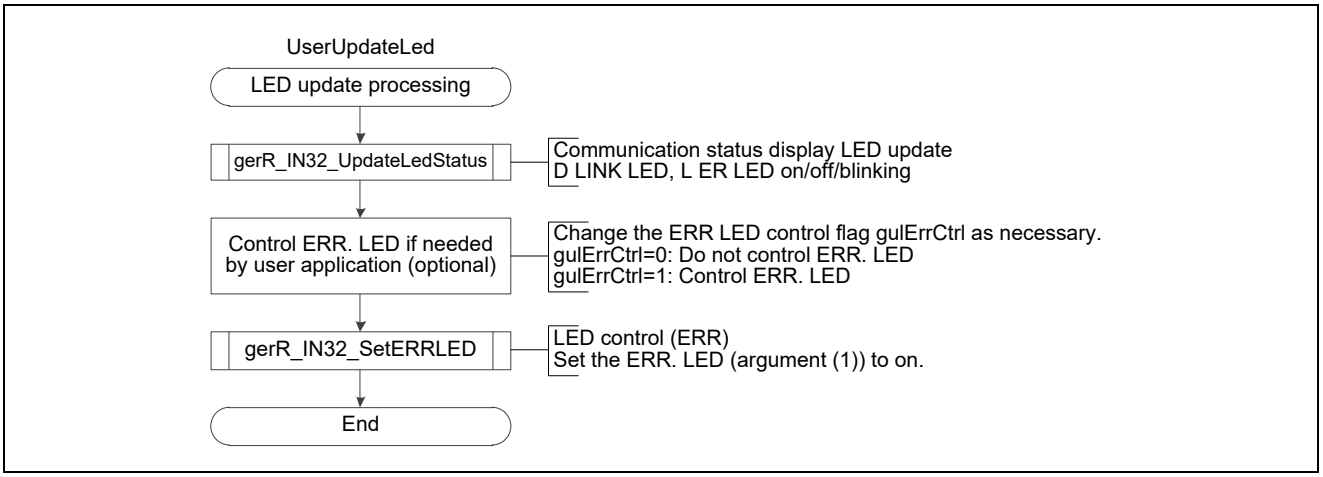


Figure 4.11Flowchart for LED Update Processing

4.2.12 Cyclic Transmission Status Update Processing

This function acquires the cyclic transmission size specified by the master station and the cyclic transmission status. (This processing is optional.)

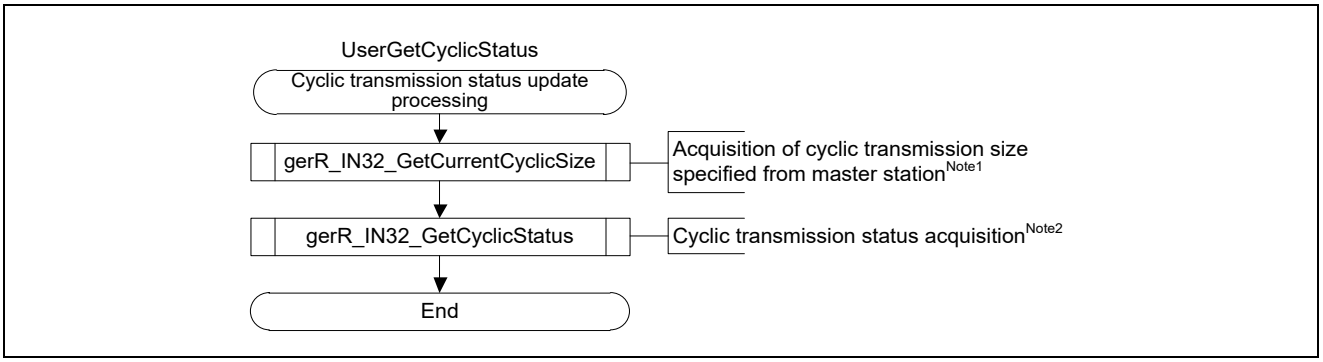


Figure 4.12Flowchart for Cyclic Transmission Status Update Processing

Notes 1. Cyclic transmission is performed by R-IN32M4-CL3 based on the cyclic transmission size specified by the master station. If own station wants to acquire the cyclic transmission size specified by the master station, call "gerR_IN32_GetCurrentCyclicSize".

2. Cyclic transmission is processed by the R-IN32M4-CL3 driver. The user program does not need to acquire the cyclic transmission status or perform processing in accordance with the status. If own station wants to acquire the cyclic transmission status, call "gerR_IN32_GetCyclicStatus".

4.2.13 MIB Information Acquisition Processing

This function acquires MIB information. (This processing is optional.)
MIB information can be used in processing such as the following:

Remark: Example: Processing in which the user program monitors the frame send/receive status and issues a report or warning in accordance with the normal state or error frequency.

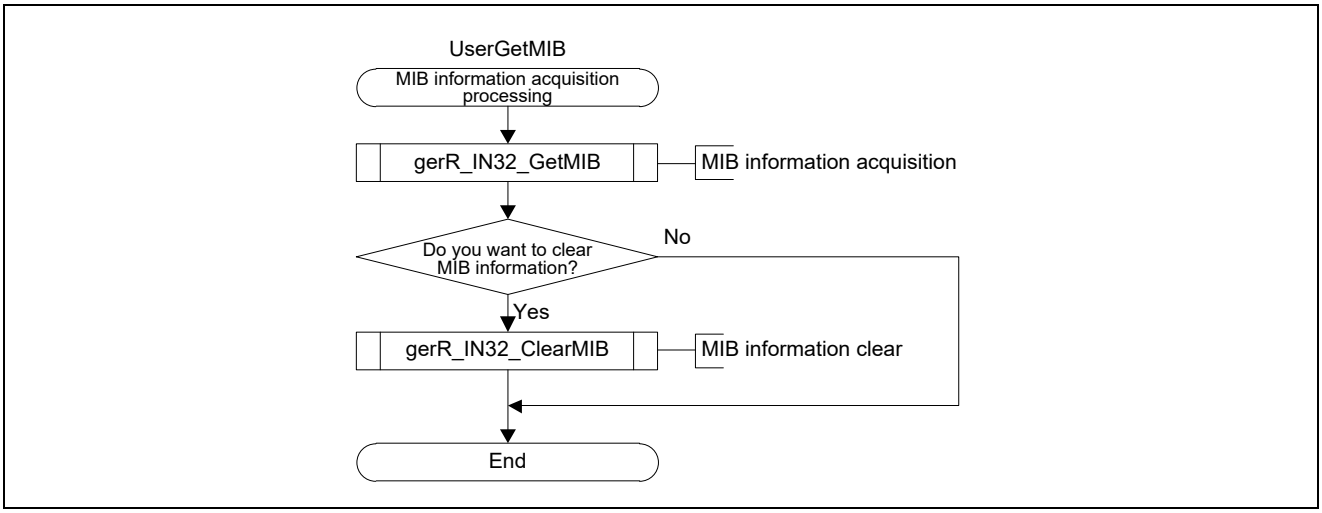


Figure 4.13Flowchart for MIB Information Acquisition Processing

Caution: MIB information is non-disclosed information. Do not disclose the information to the end user.

(1) List of MIB Information of Ring Control Area

Table 4.4 List of MIB Information of Ring Control Area

No.	MIB Information	Description
1	No. of HEC error frames	Counts the number of HEC errors in received frames.
2	No. of DCS/FCS error frames	Counts the number of DCS/FCS errors in received frames.
3	No. of undersize error frames	Counts the number of received error frames with a size less than 28 bytes.
4	No. of forwarded frames	Counts the number of forwarded frames.
5	No. of upper layer transmission frames	Counts the number of frames transmitted to upper layers.
6	No. of discarded frames due to full forward buffer	Counts the number of frames discarded due to a full forward buffer.
7	No. of discarded frames due to full upper layer transmission buffer	Counts the number of frames discarded due to a full upper layer transmission buffer.

(2) List of MIB Information of MAC IP Area

Table 4.5 List of MIB Information of MAC IP Area

No.	MIB Information	Description
1	No. of received frames	Counts all frame receptions, including error frames. Error frames: FCS error, undersized, oversized frames
2	No. of sent frames	Counts the number of sent frames.
3	No. of received undersized frames	Counts the number of received frames with a size less than 64 bytes.
4	No. of received oversized frames	Counts the number of received frames with a size exceeding 1, 518 bytes.
5	No. of received frame FCS errors	Counts the number of received frames with an FCS error.
6	No. of received frame fragment errors	Counts the number of received frames with fragment errors. Fragment error: A frame with less than 64 bytes and an FCS error
7	No. of frames detected within minimum IFG	Counts the number of frames detected within the minimum inter-frame gap (IFG).
8	No of received frames with SFD or less	Counts the number of received frames that ended at a field up to SFD and were not recognized as a valid frame.
9	No. of reception code errors	Counts the number of GMII reception data errors detected (RECV_*_ERR=1[Note]). Counts a RECV_*_ERR[Note] that occurred multiple times in an idle state (RECV_*_DV=1[Note]) as one error.
10	No. of received invalid carrier errors	Counts the number of invalid carriers that occurred in an idle state. Counts multiple invalid carriers that occurred in an idle state as one error.
11	No. of received invalid carrier extension errors	Counts the number of invalid carrier extensions that occurred in an idle state. Counts multiple invalid carrier extensions that occurred in an idle state as one error.

Note. The asterisk ("*") indicates a wild character. (A: Port 0, B: Port 1)

(3) List of Other MIB Information

Table 4.6 List of Other MIB Information

No.	MIB Information	Description
1	No. of link downs (port 1)	Counts the number of link downs of port 0.
2	No. of link downs (port 2)	Counts the number of link downs of port 1.
3	No. of master watch timer errors	Counts the number of timeouts of the master watch timer.
4	No. of received cyclic frames	Counts the number of cyclic frames received by R-IN32M4-CL3.
5	No. of received transient frames	Counts the number of transient frames received by R-IN32M4-CL3.
6	No. of received transient frames discarded	Counts the number of received transient frames discarded by R-IN32M4-CL3.

4.3 User Program Details (Transient Transmission Related)

4.3.1 Transient Transmission Processing Overview

[When own station is server]

The following shows an image of the processing procedure in which the server sends a response frame in response to a request frame from a client.

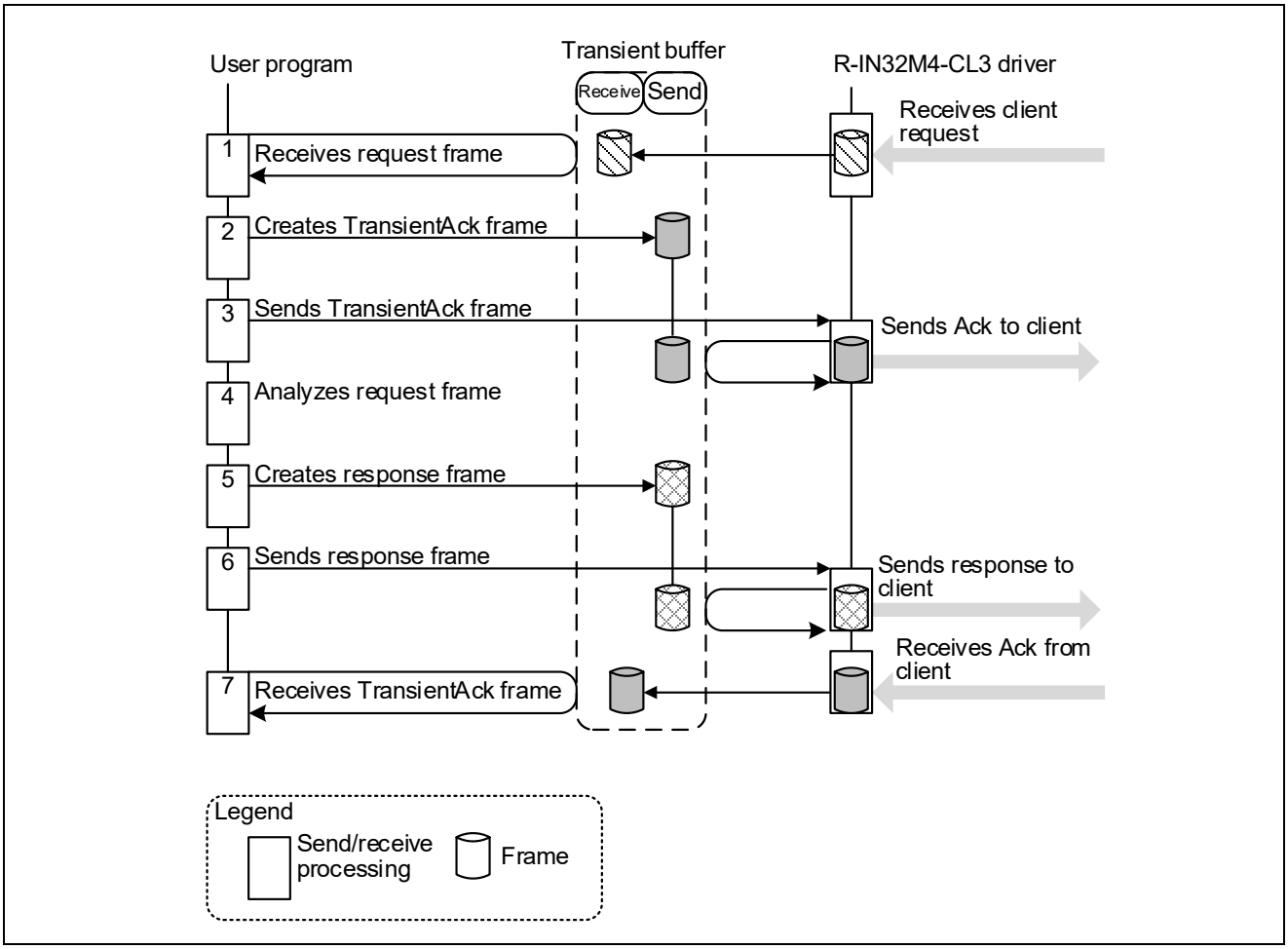


Figure 4.14Server Processing Procedure

No.	Processing	Reference Section
1	Receives Transient frame	4.3.2 "Transient1, Transient2, and TransientAck Receive Processing"
2	Creates TransientAck frame	4.3.13 "TransientAck Frame Creation Processing"
3	Sends TransientAck frame	4.3.14 "Transient1, Transient2, and TransientAck Send Processing"
4	Analyzes received request frame and performs processing for each frame	4.3.3 "Transient1 Receive Data Processing" 4.3.8 "Transient2 Receive Data Processing"
5	Creates response frame in accordance with command	4.3.6 "SLMP Memory Read Request Frame Receive Processing" 4.3.7 "SLMP Memory Write Request Frame Receive Processing" 4.3.10 "Transient2 Memory Write Request Receive Processing"
6	Sends response frame	4.3.14 "Transient1, Transient2, and TransientAck Send Processing"
7	Receives TransientAck frame	4.3.12 "TransientAck Receive Data Processing"

[When own station is client]

The following shows an image of the processing procedure in which the client sends a request frame and then receives a response frame from the server.

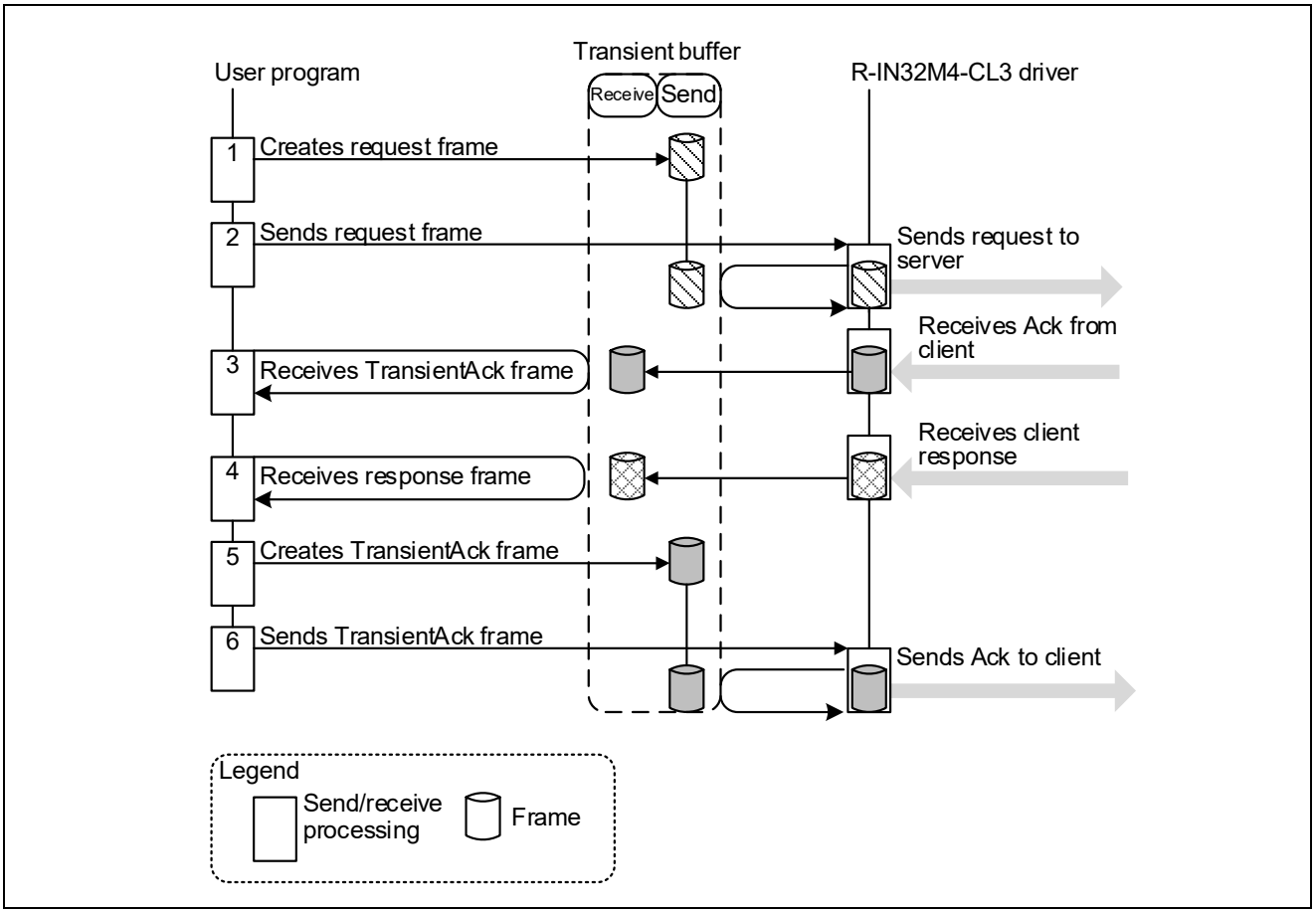


Figure 4.15Client Processing Procedure

No.	Processing	Reference Section
1	Creates request frame[Note]	4.3.15 "SLMP Memory Read Request Frame Creation Processing" 4.3.19 "Transient2 Request Frame Creation Processing"
2	Sends request frame	4.3.14 "Transient1, Transient2, and TransientAck Send Processing"
3	Receives TransientAck frame	4.3.2 "Transient1, Transient2, and TransientAck Receive Processing" 4.3.12 "TransientAck Receive Data Processing"
4	Receives response frame	4.3.3 "Transient1 Receive Data Processing" 4.3.18 "SLMP Memory Read Response Receive Processing" 4.3.8 "Transient2 Receive Data Processing" 4.3.21 "Transient2 Memory Read Response Receive Processing"
5	Creates TransientAck frame	4.3.13 "TransientAck Frame Creation Processing"
6	Sends TransientAck frame	4.3.14 "Transient1, Transient2, and TransientAck Send Processing"

Note. Transient2 memory read and SLMP memory read are described in sample code as the sample processing of each command. Implement commands other than the above by customizing the user program.

4.3.2 Transient1, Transient2, and TransientAck Receive Processing

This function receives Transient1, Transient2, and TransientAck frames and processes the data.

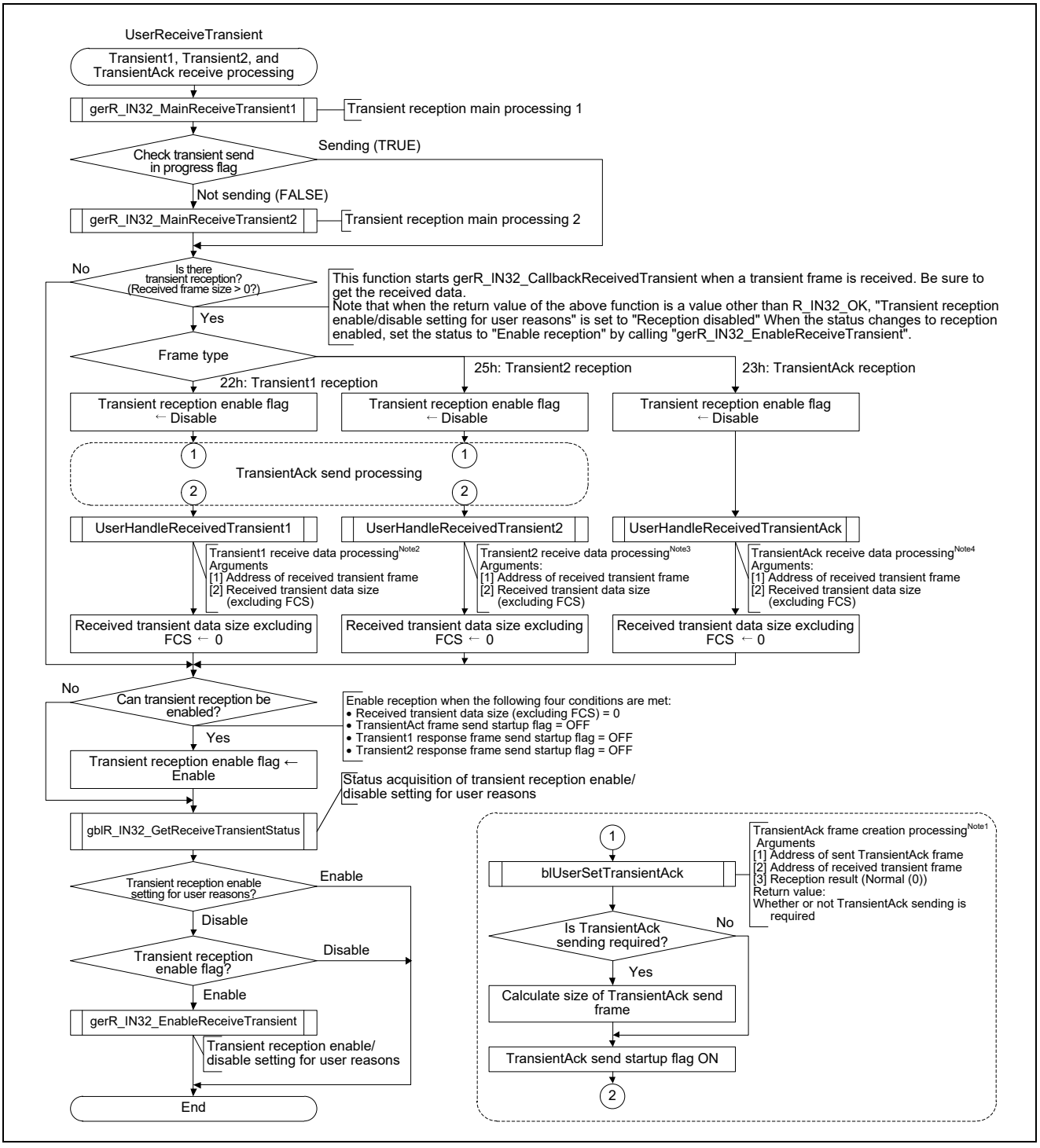


Figure 4.16Flowchart for Transient1, Transient2, and TransientAck Receive Processing

Notes 1. For details, see Section 4.3.13 "TransientAck Frame Creation Processing".

2. For details, see Section 4.3.3 "Transient1 Receive Data Processing".

3. For details, see Section 4.3.8 "Transient2 Receive Data Processing".

4. For details, see Section 4.3.12 "TransientAck Receive Data Processing".

4.3.3 Transient1 Receive Data Processing

This function analyzes a received Transient1 frame and performs processing in accordance with the analysis result. In addition, this function reconstructs data when a Transient1 frame is received divided.

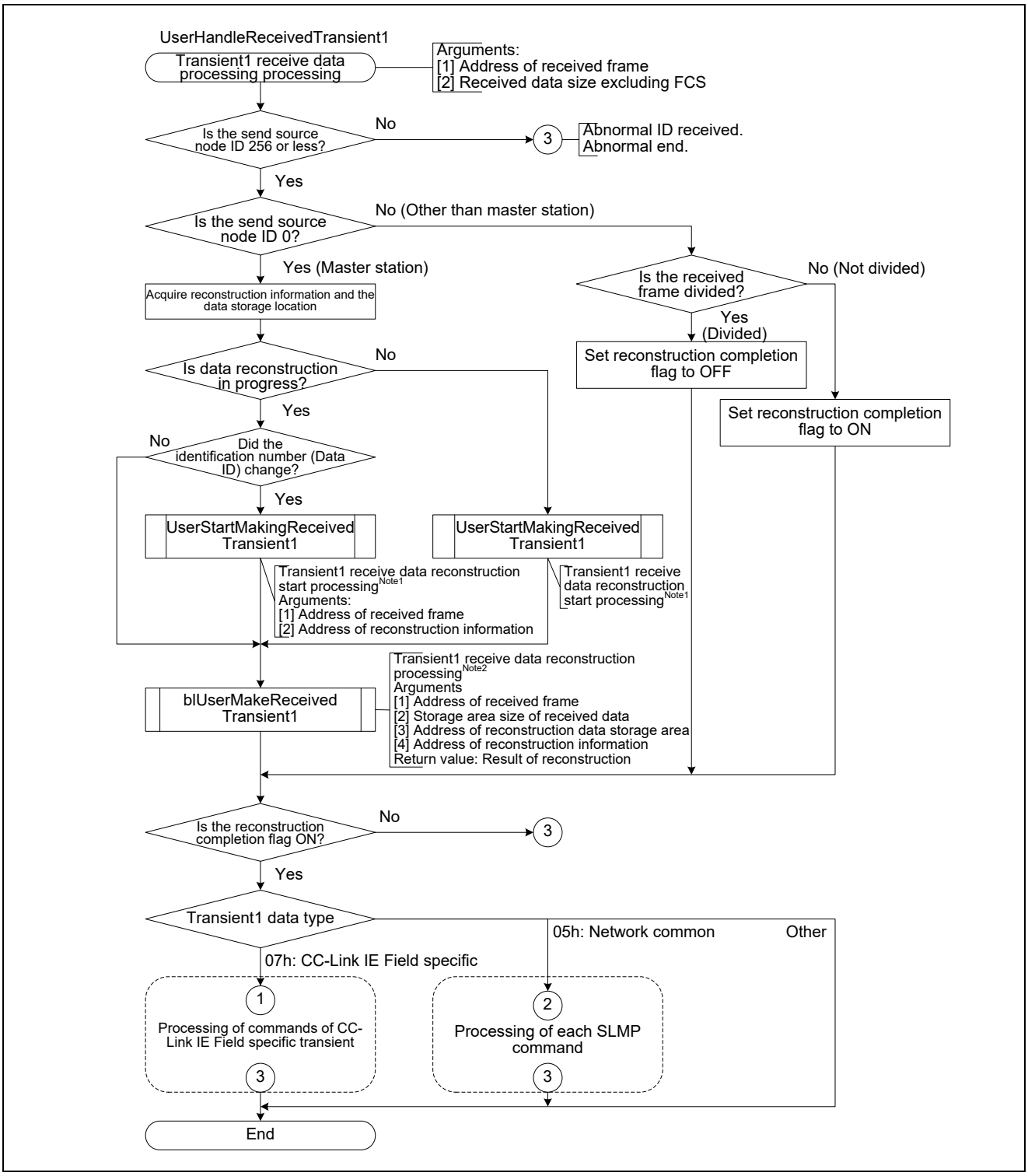


Figure 4.17Flowchart for Transient1 Receive Data Processing (1/4)

Notes 1. For details, see Section 4.3.4 "Transient1 Receive Data Reconstruction Start Processing".
2. For details, see Section 4.3.5 "Transient1 Receive Data Reconstruction Processing".

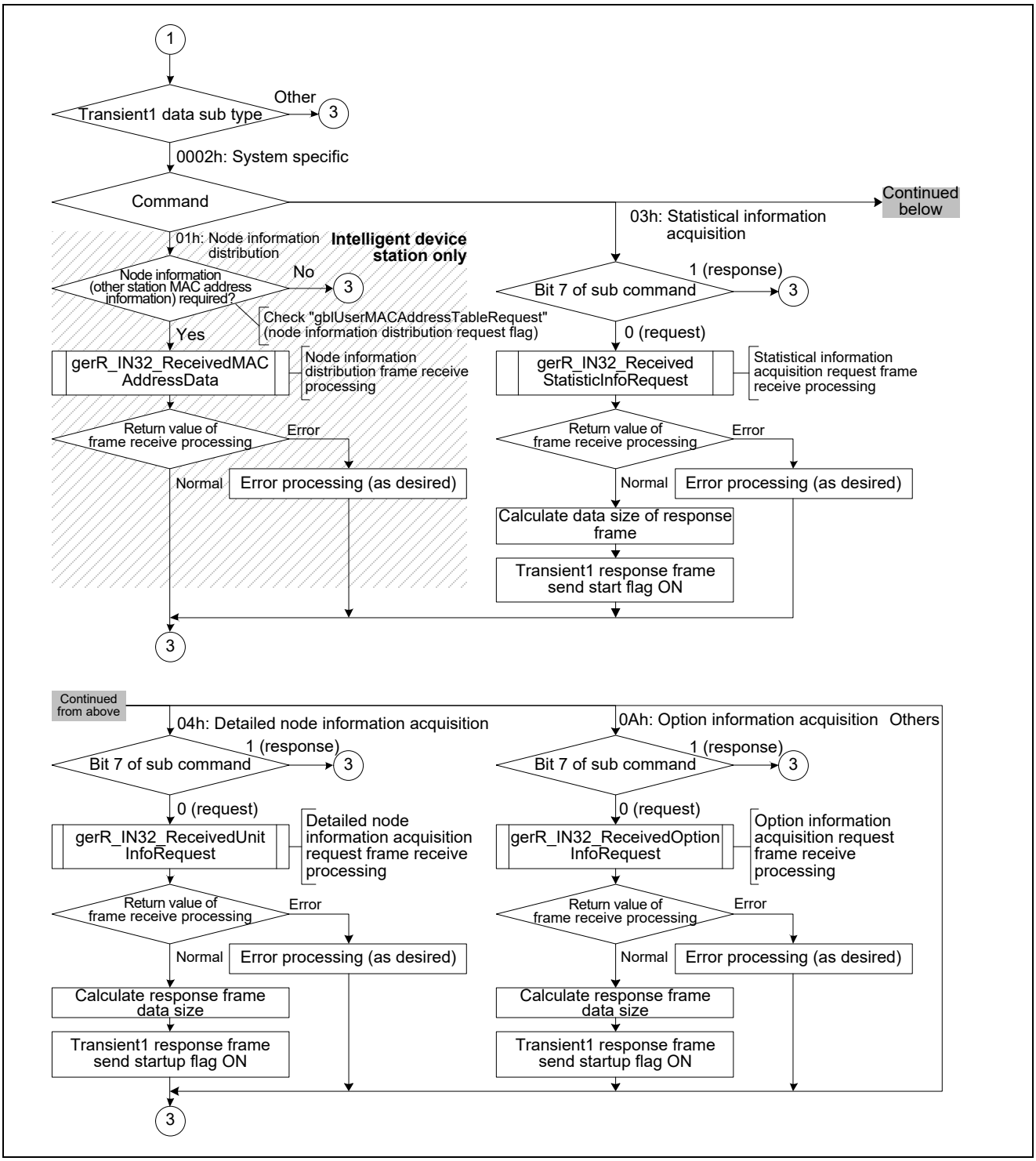


Figure 4.17Flowchart for Transient1 Receive Data Processing (2/4)

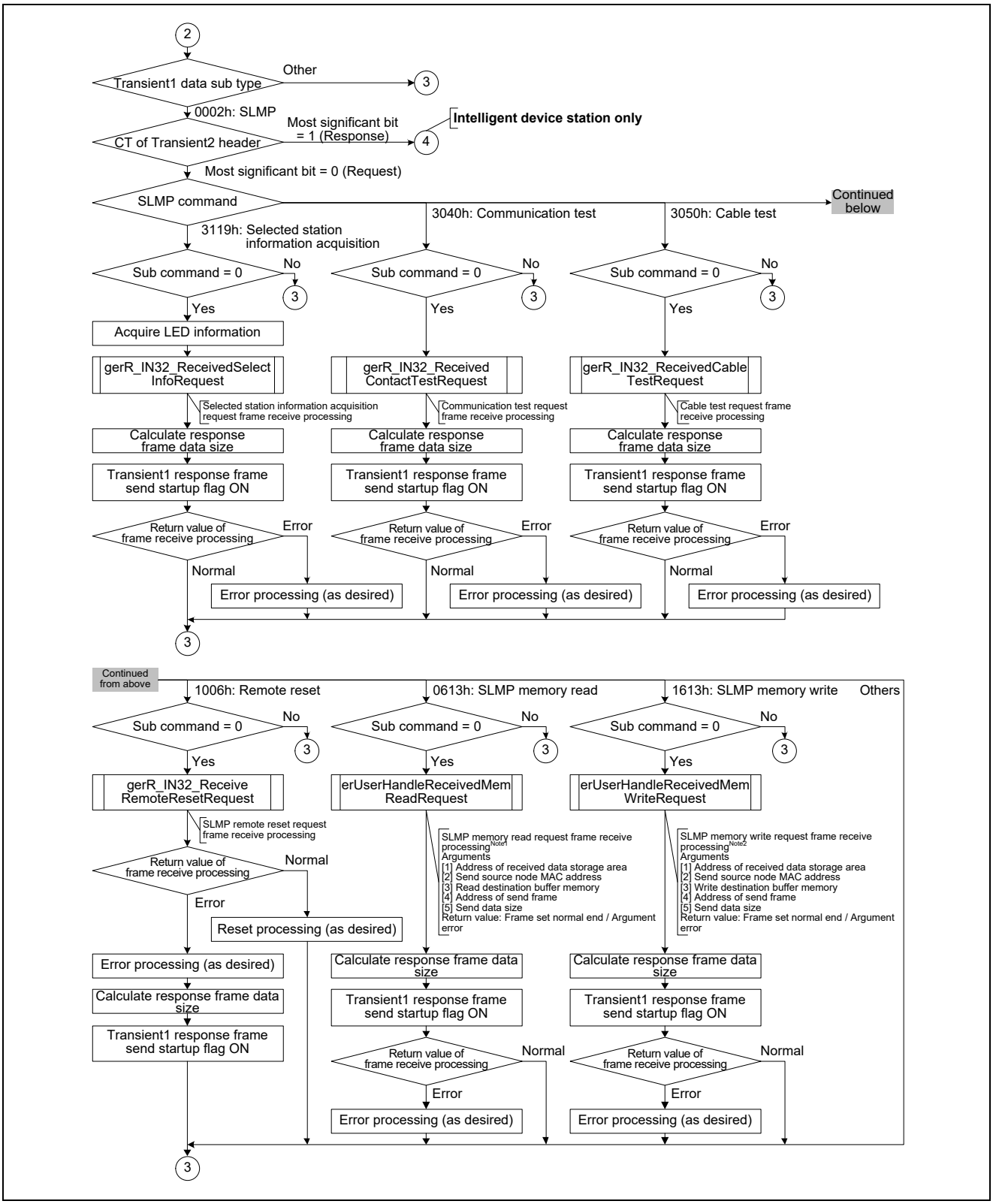


Figure 4.17Flowchart for Transient1 Receive Data Processing (3/4)

Notes 1. For details, see Section 4.3.6 "SLMP Memory Read Request Frame Receive Processing".
2. For details, see Section 4.3.7 "SLMP Memory Write Request Frame Receive Processing".

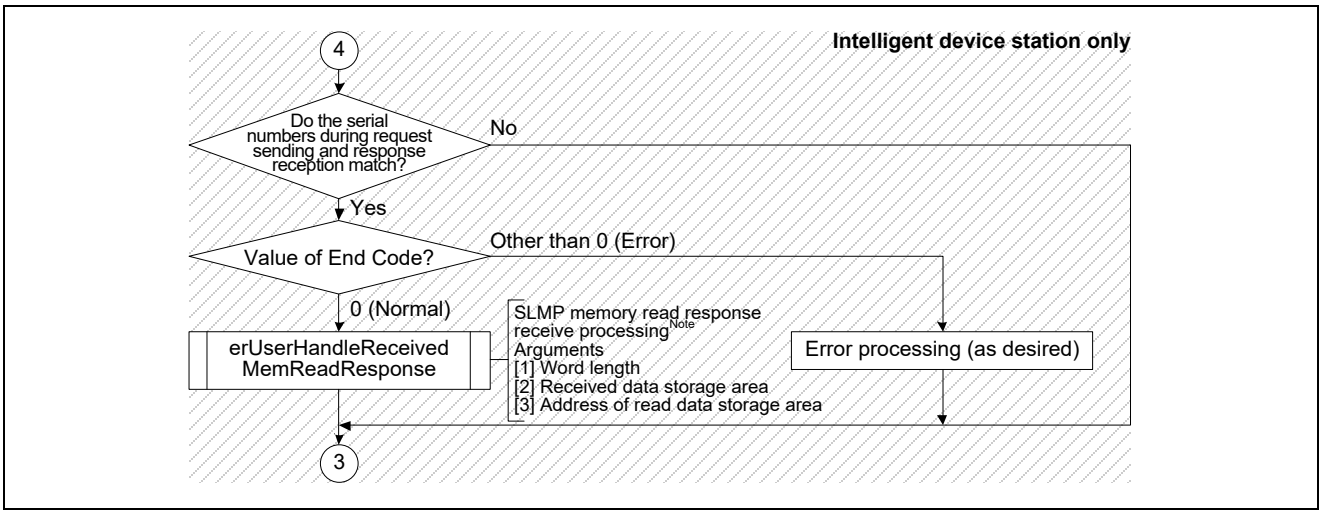


Figure 4.17Flowchart for Transient1 Receive Data Processing (4/4)

Note. For details, see Section 4.3.18 "SLMP Memory Read Response Receive Processing".

(1) Node Information Distribution

Node information is information that indicates the correspondence between other station numbers and MAC addresses. Node information is used when an intelligent device station sends a transient request to another station (when own station is client).

- When own station wants to receive node information (when you want to send a transient request)

  In iUserInitialization (Section 4.2.2 "Initialization Processing"), set both "blMACAddressTableRequest" (initial value of node information distribution request) and "gblUserMACAddressTableRequest" (node information distribution request flag) to "R_IN32_TRUE". Note that a TransientAck and transient response are not required for received node information distribution frames.

- When own station does not want to receive node information (when own station does not want to send a transient request)

  In iUserInitialization (Section 4.2.2 "Initialization Processing"), set both "blMACAddressTableRequest" (initial value of node information distribution request) and "gblUserMACAddressTableRequest" (node information distribution request flag) to "R_IN32_FALSE".

(2) SLMP Request Reception from Master Station

The CC-Link IE Field Network diagnostics and parameter processing/command execution of slave stations can be performed using the engineering tool. These functions can be used by its own station responding to an SLMP request frame from the master station.

The following illustrates reception based on the selected station information acquisition command. The processing for sending and receiving is the same as that for the communication test, cable test, remote reset command, and the commands described in CSP+.

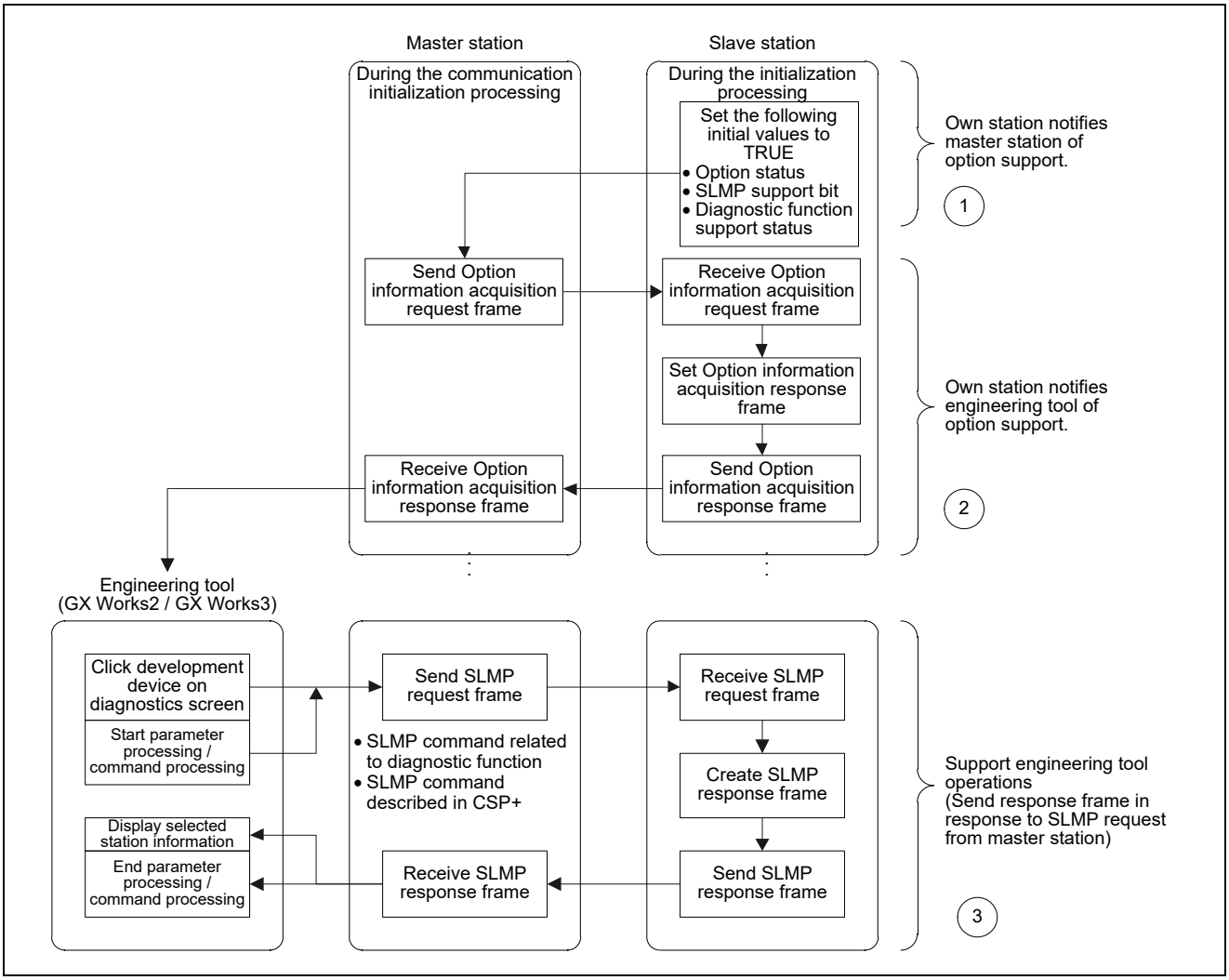


Figure 4.18SMP Request Reception Procedure

1. R_IN32_UNITINIT_T setup (R-IN32M4-CL3 initial setup)
   Set the following members of R_IN32_UNITINIT_T to "R_IN32_TRUE". For details, see Table 5.7 "R_IN32_UNITINIT_T List".
   - ulOptionSupport (Initial value of option status)
   - ulSlmpSupport (Initial value of SLMP support bit)
   - ulSlmpDiagnosisSupport (Initial value of diagnostic function support status initial value)
2. Response to Option information acquisition request frame
   The R-IN32M4-CL3 driver interface function "gerR_IN32_ReceivedOptionInfoRequest" responds to the Option information acquisition request frame from the master station.
3. Response to Selected station information acquisition request frame
   The R-IN32M4-CL3 driver interface function "gerR_IN32_ReceivedSelectInfoRequest" responds to the Selected station information acquisition request frame from the master station.

(3) Creating LED Information

To display the LED status of its own station on the selected station communication status monitor, store the LED information corresponding to its own station status in "R_IN32_SELECTINFO_LED_INFO_T". Make sure the LED information matches the status of the LEDs mounted in the R-IN32M4-CL3 application circuit.

Table 4.7 R_IN32_SELECTINFO_LED_INFO_T

No.	Member		Description
1	UCHAR	uchRow	No. of LED array rows (fixed to 04h)
2	UCHAR	uchColumn	No. of LED array columns (fixed to 02h)
3	R_IN32_LED_INFO_T	stLedInf[8]	LED information 1 to 8

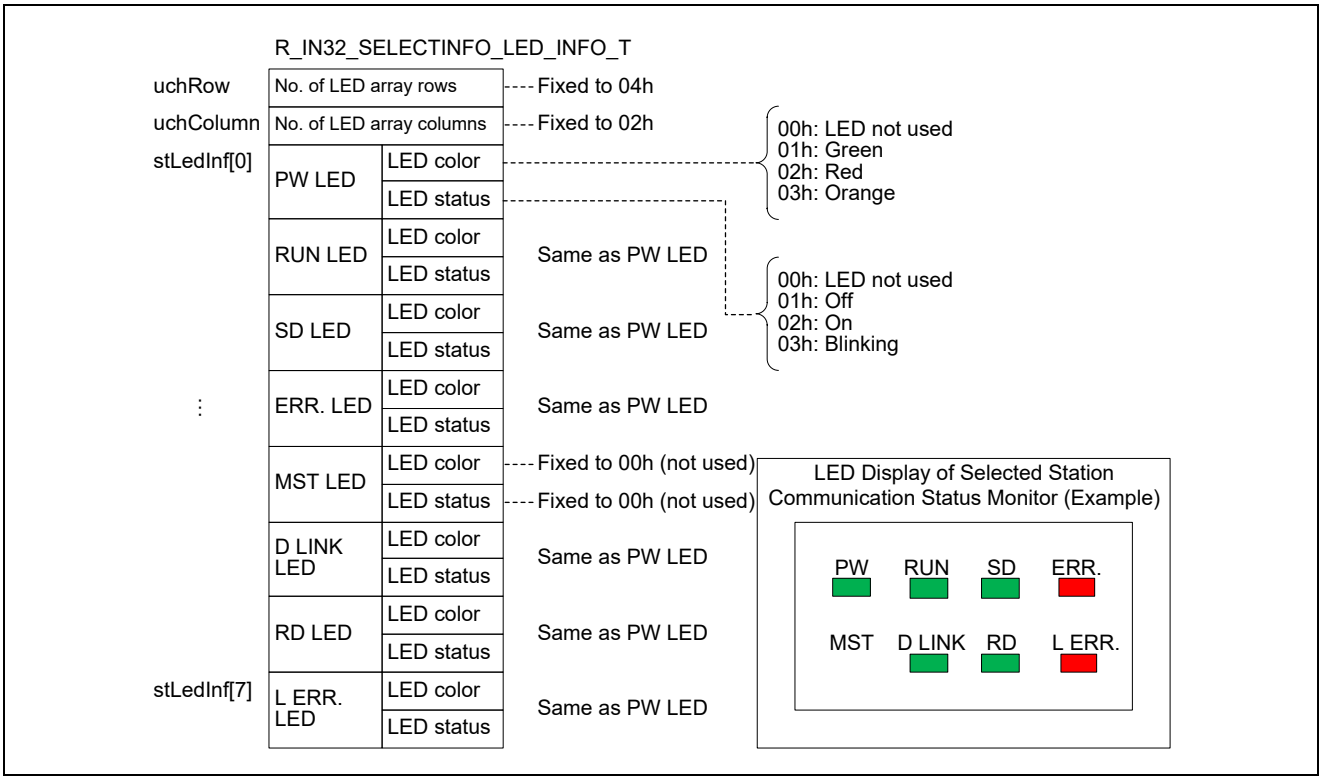


Figure 4.19Creating LED information

Caution: When the LED status (on/off/blinking) of the R-IN32M4-CL3 application circuit changes at an interval shorter than the communication interval of selected station information acquisition, the change in the LED status is not transmitted to the engineering tool (such as with LEDs that repeated turn on and off at high speed, such as SD and RD).
When GX Works2 and GX Works3 are used, the communication interval of selected station information acquisition is approximately 5 seconds, and therefore the LED display on the diagnostics screen differs from the actual LED status.

4.3.4 Transient1 Receive Data Reconstruction Start Processing

This function starts reconstructing the divided Transient1 receive frame.

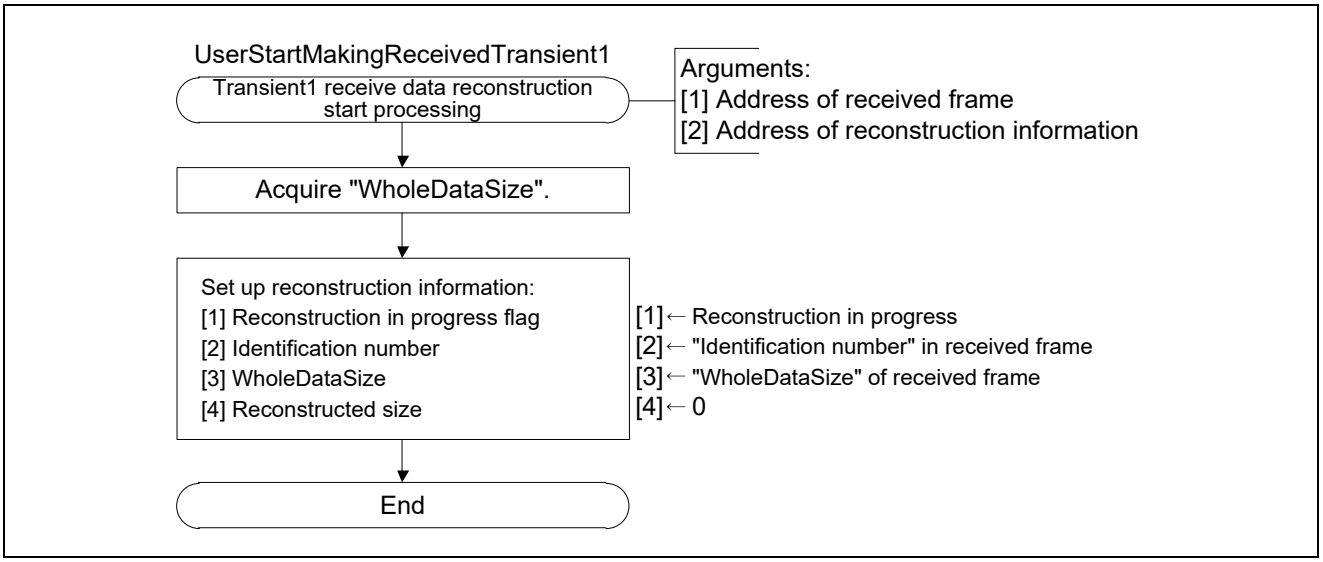


Figure 4.20Flowchart for Transient1 Receive Data Reconstruction Start Processing

4.3.5 Transient1 Receive Data Reconstruction Processing

This function reconstructs the data of the Transient1 frame.

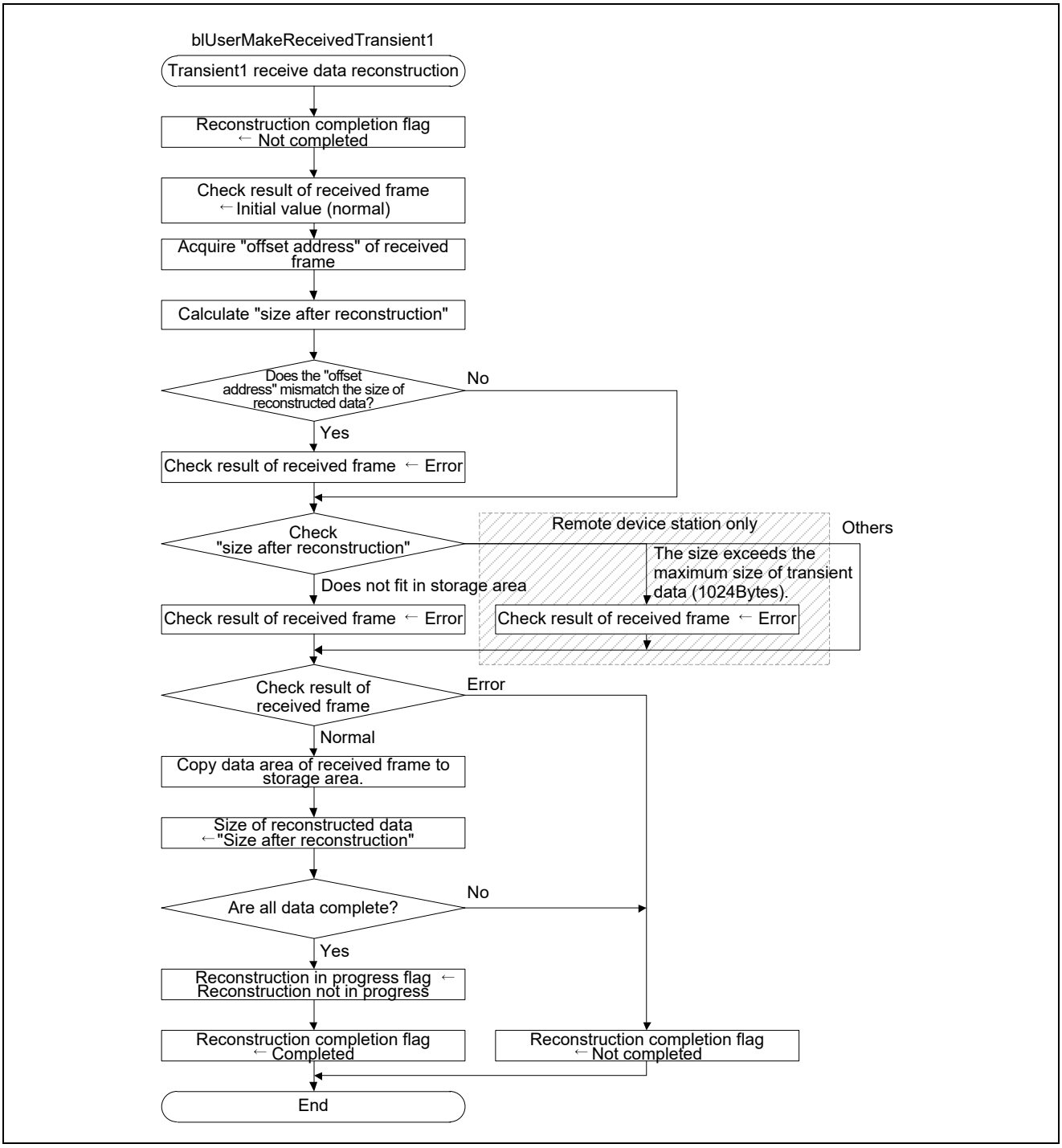


Figure 4.21Flowchart for Transient1 Receive Data Reconstruction Processing

4.3.6 SLMP Memory Read Request Frame Receive Processing

This function performs frame reception processing when SLMP memory read request frame is received in its own station from another station.

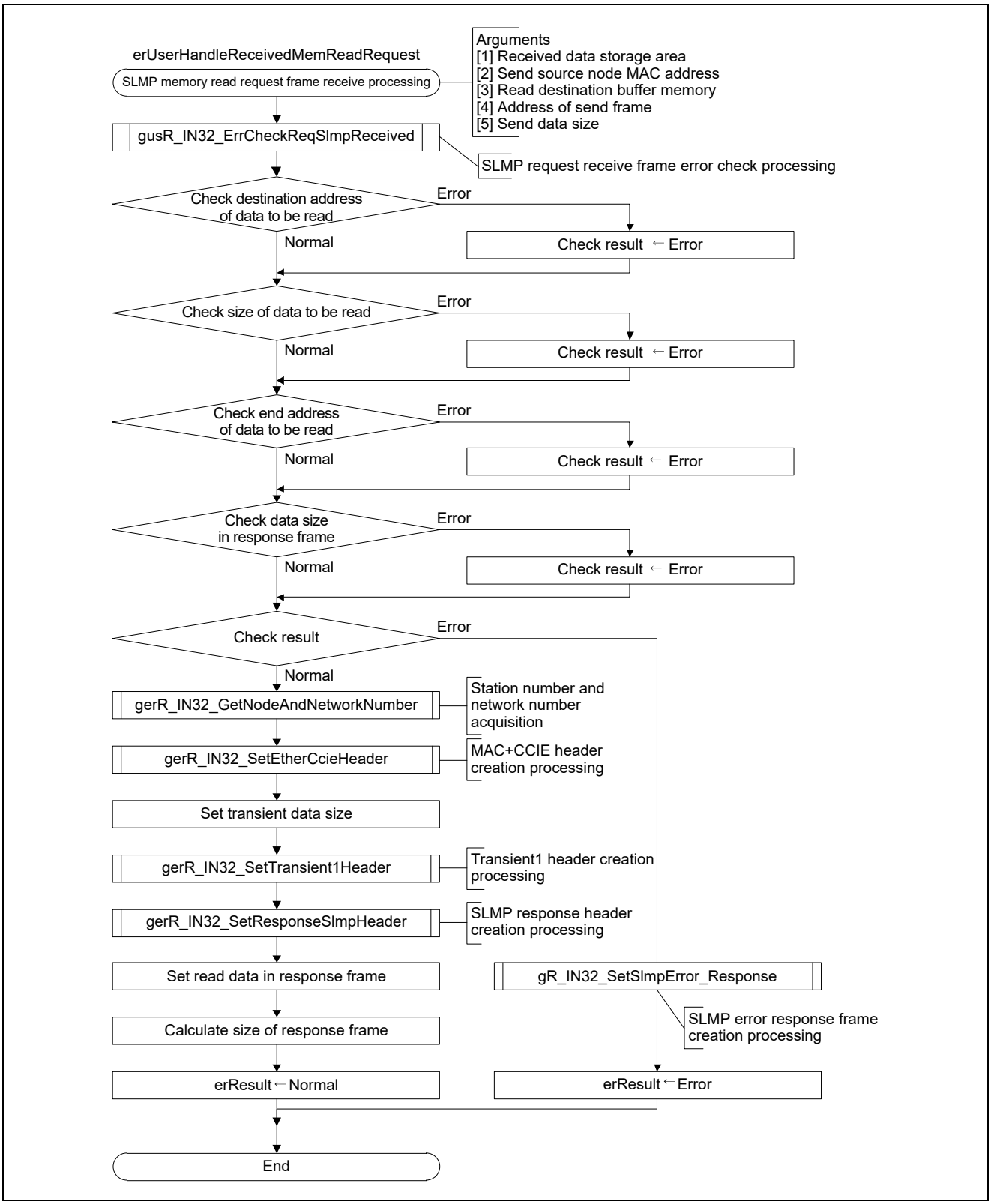


Figure 4.22Flowchart for SLMP Memory Read Request Frame Receive Processing

4.3.7 SLMP Memory Write Request Frame Receive Processing

This function performs frame receive processing when SLMP memory write request frame is received in its own station from another station.

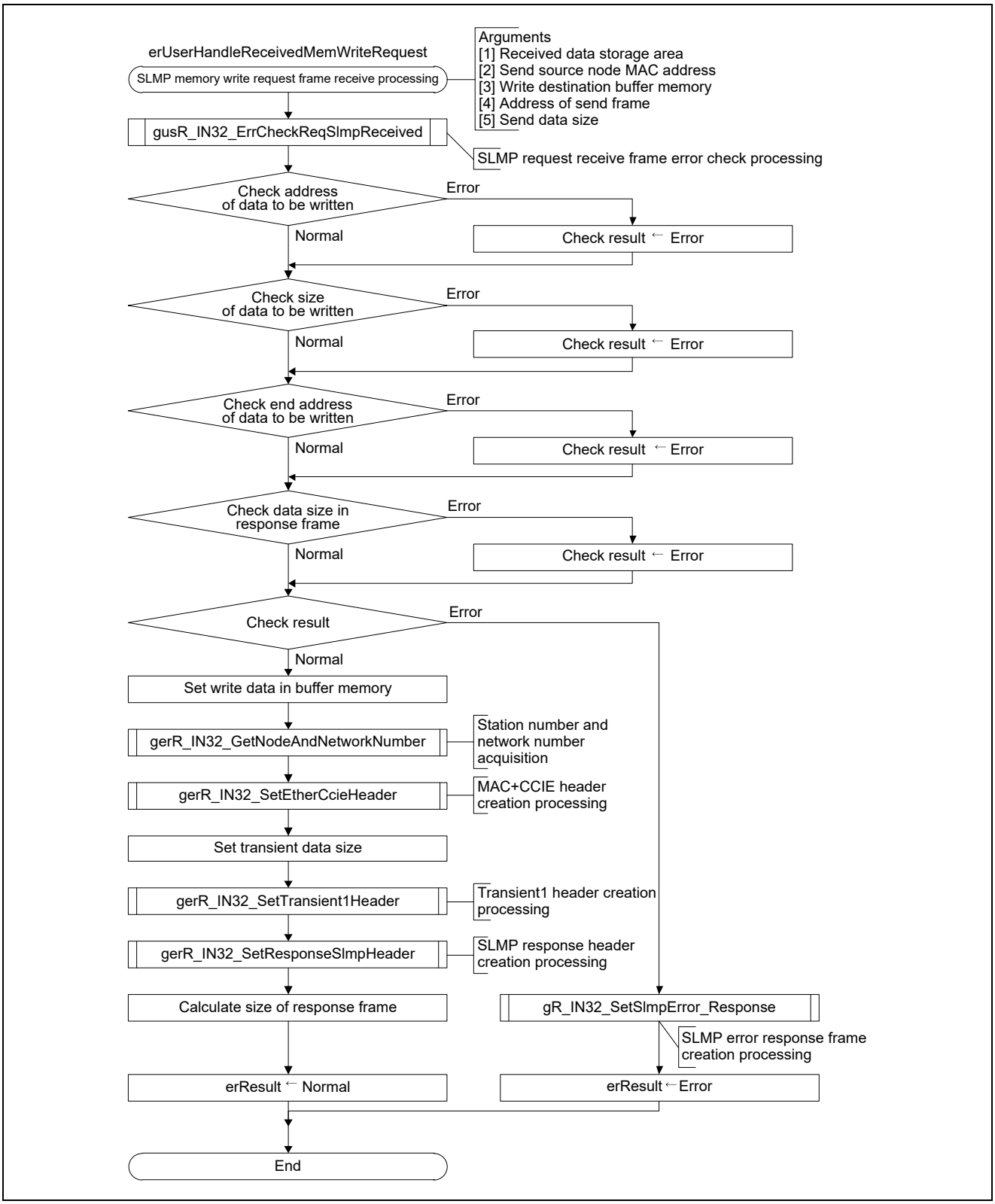


Figure 4.23Flowchart for SLMP Memory Write Request Frame Receive Processing

4.3.8 Transient2 Receive Data Processing

This function analyzes a received Transient2 frame and creates or receives a response frame in accordance with the analysis results.

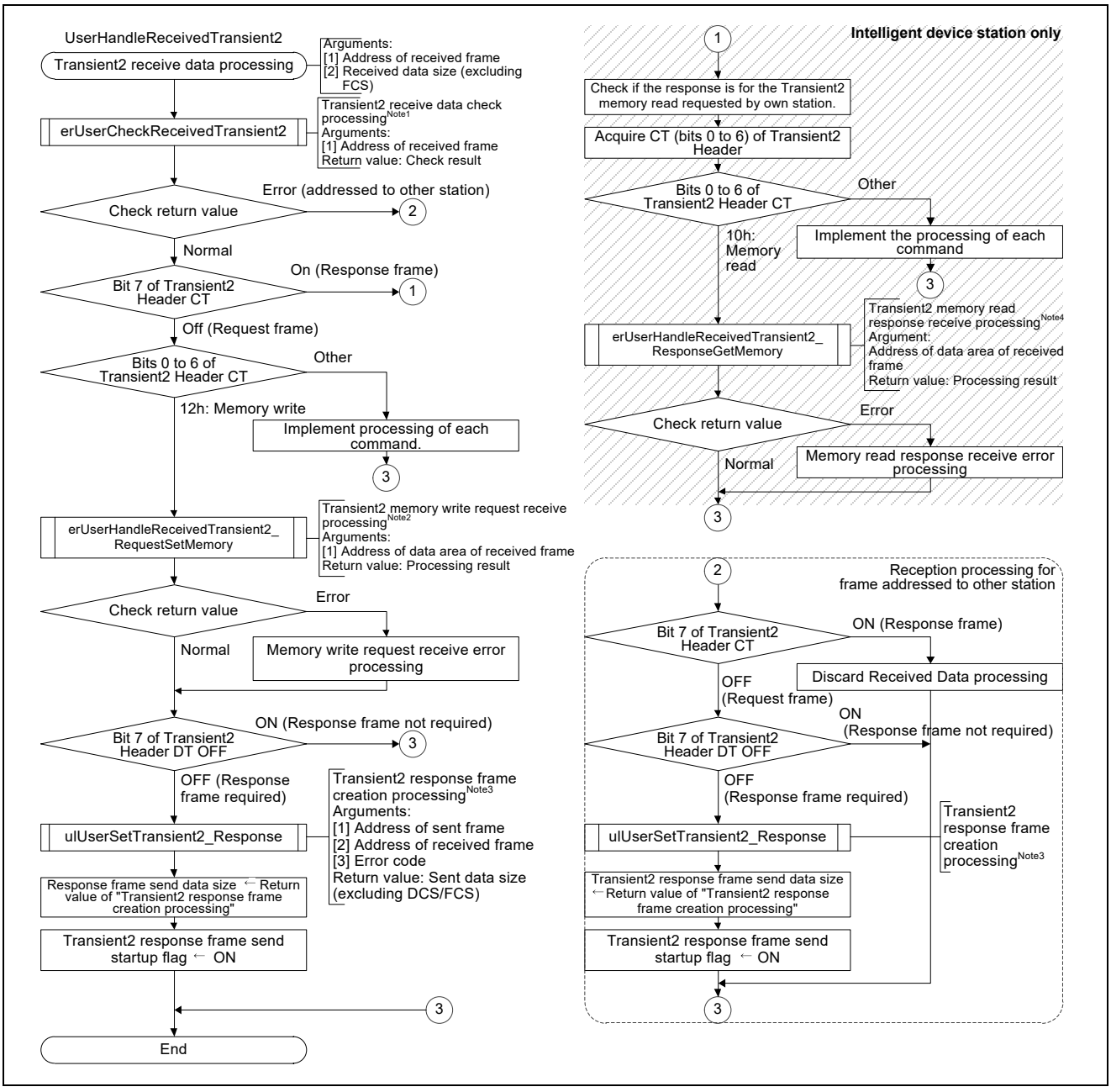


Figure 4.24Flowchart for Transient2 Receive Data Processing

Notes 1. For details, see Section 4.3.9 "Transient2 Receive Data Check Processing".
2. For details, see Section 4.3.10 "Transient2 Memory Write Request Receive Processing".
3. For details, see Section 4.3.11 "Transient2 Response Frame Creation Processing".
4. For details, see Section 4.3.21 "Transient2 Memory Read Response Receive Processing".

4.3.9 Transient2 Receive Data Check Processing

This function checks if the received Transient2 frame is addressed to its own station, and checks the destination station number (DA/DS) and destination network number (DNA).

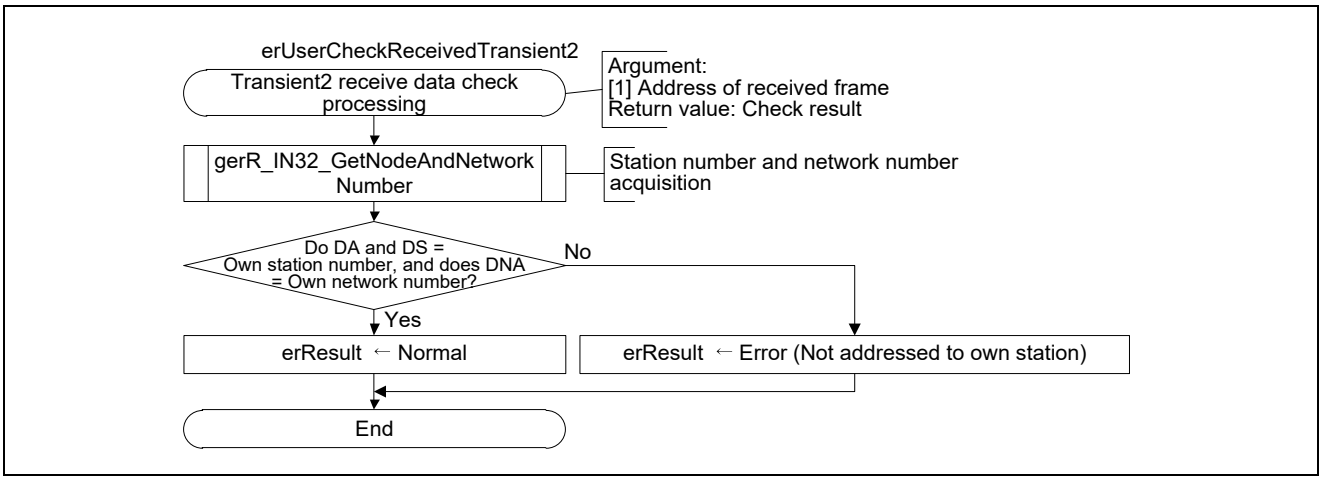


Figure 4.25Flowchart for Transient2 Receive Data Check Processing

4.3.10 Transient2 Memory Write Request Receive Processing

This function performs frame reception processing when a frame requesting to set Transient2 memory in its own station is received from another station.

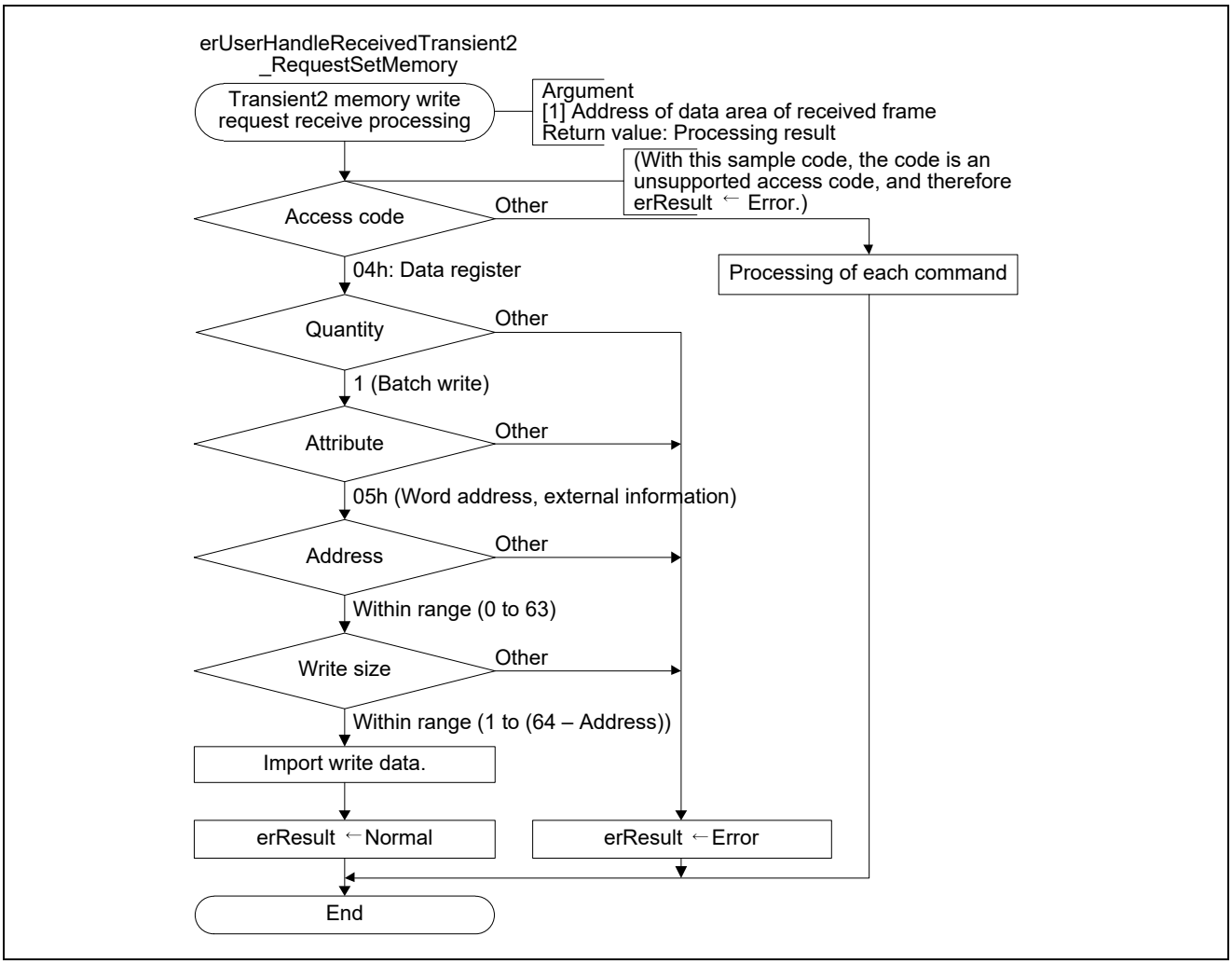


Figure 4.26Flowchart for Transient2 Memory Write Request Receive Processing

The flow above illustrates the command processing for the settings below. Any other setup results in error.

(1) Memory write
(2) Access code: 04h (Data register)
(3) Attribute: 05h (Word access, external information)
(4) Address (start address of write destination): 0
(5) Write size: 1 to (64 – Address)

4.3.11 Transient2 Response Frame Creation Processing

This function creates a Transient2 response frame.

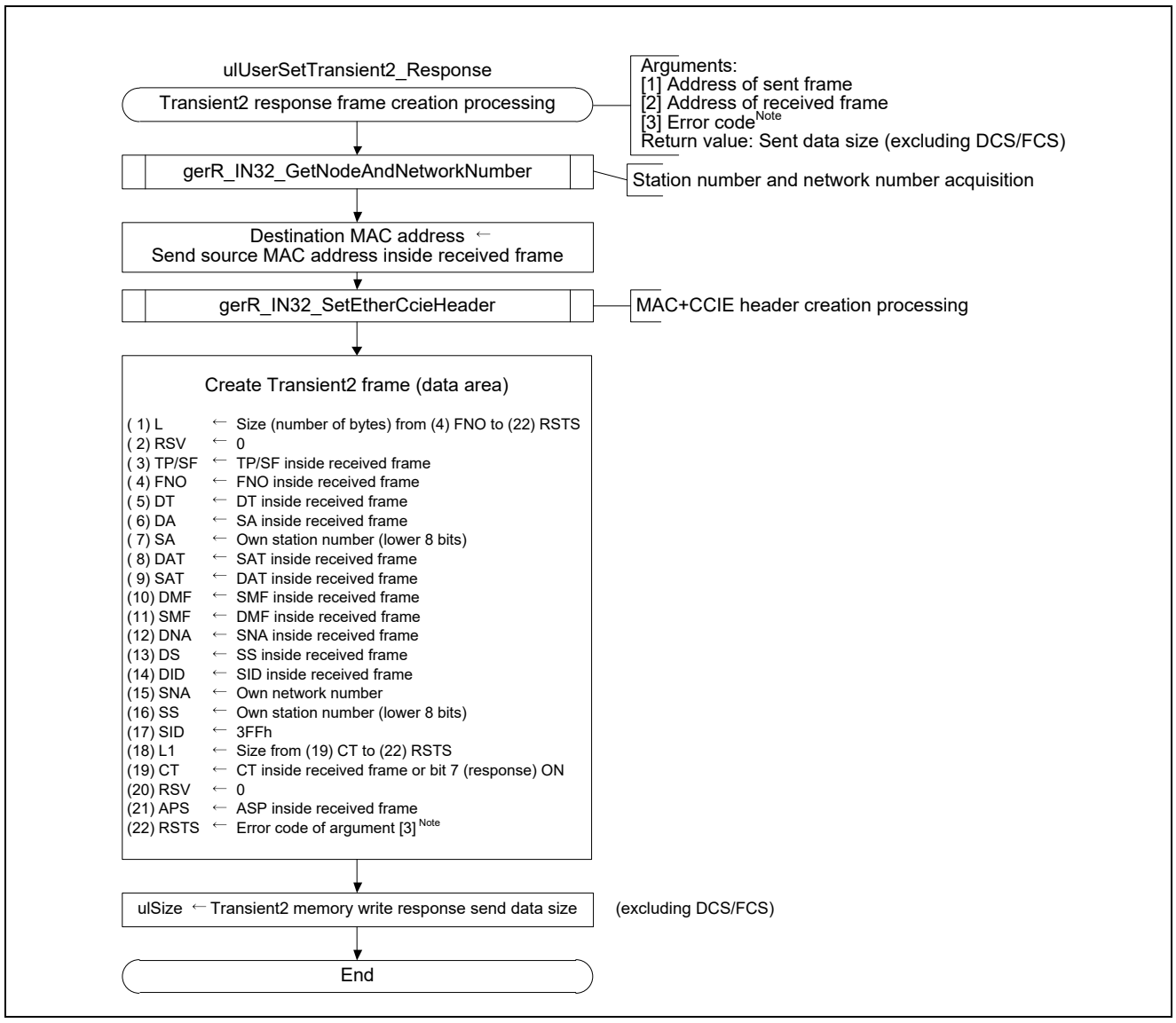


Figure 4.27Flowchart for Transient2 Response Frame Creation Processing

Note. For (3) Error Code details, see Section 7.3.3 "Details of Return Code (RSTS)".

4.3.12 TransientAck Receive Data Processing

This function analyzes the received TransientAck frame and adds processing corresponding to the analytical results.

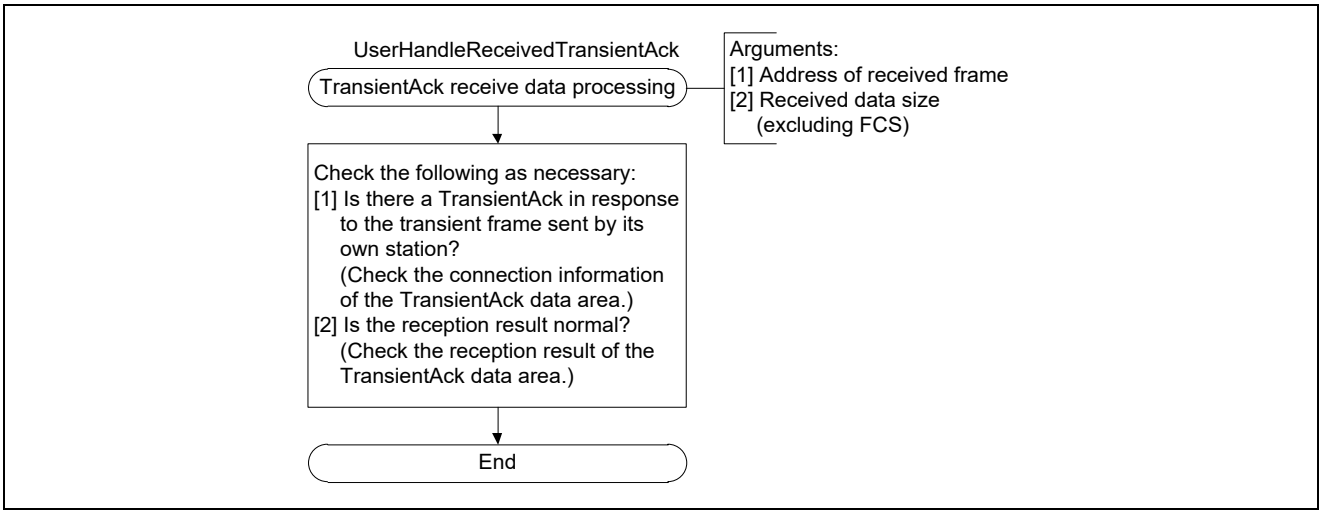


Figure 4.28Flowchart for TransientAck Receive Data Processing

4.3.13 TransientAck Frame Creation Processing

This function creates a TransientAck frame.

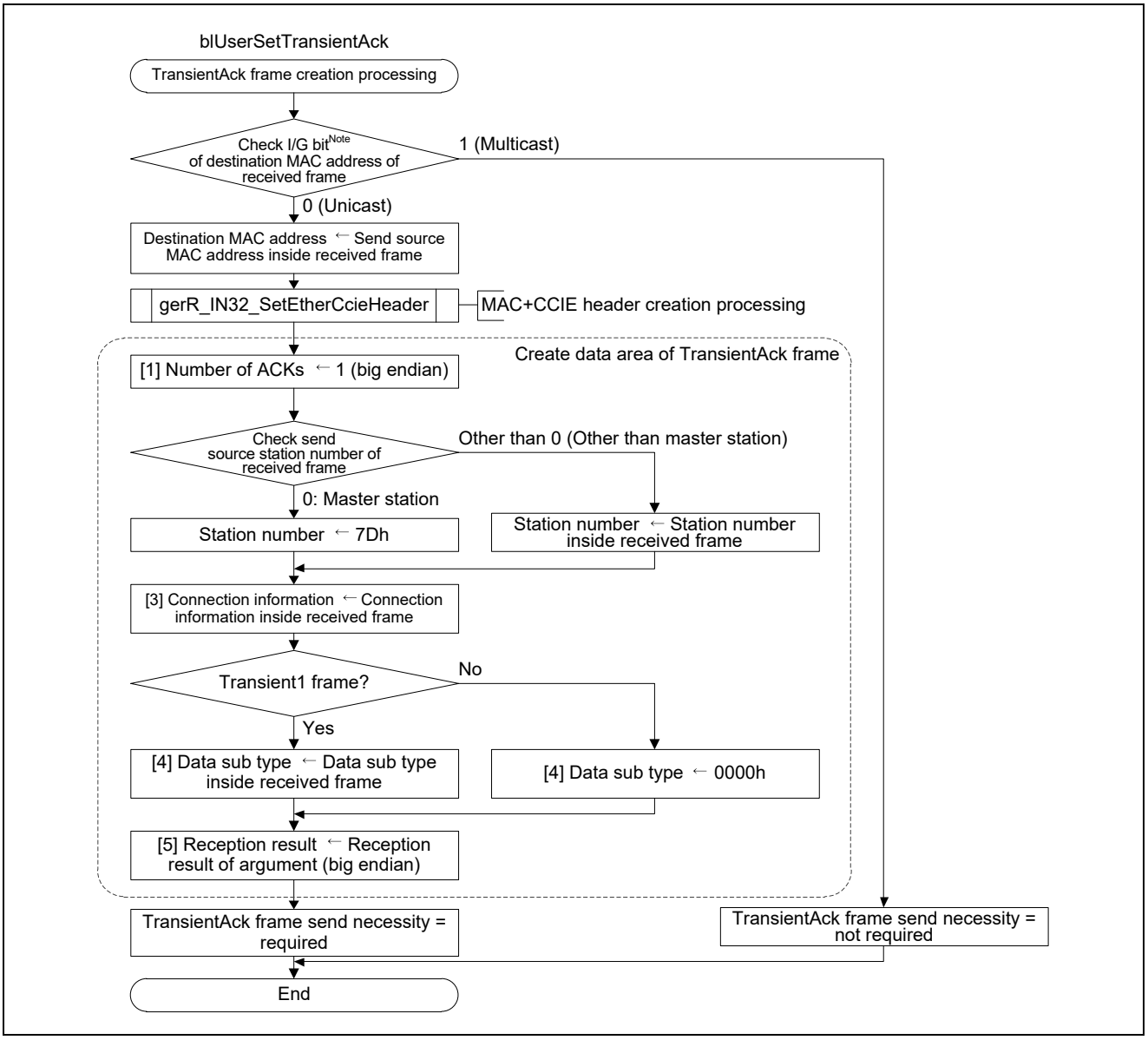
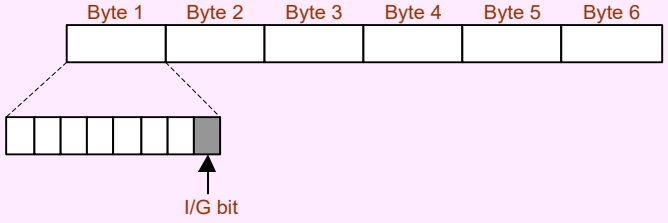


Figure 4.29Flowchart for TransientAck Frame Creation Processing

Note. The I/G bit is the least significant bit of the first byte (octet) of the MAC address.



4.3.14 Transient1, Transient2, and TransientAck Send Processing

This function sends Transient1, Transient2, and TransientAck frames.

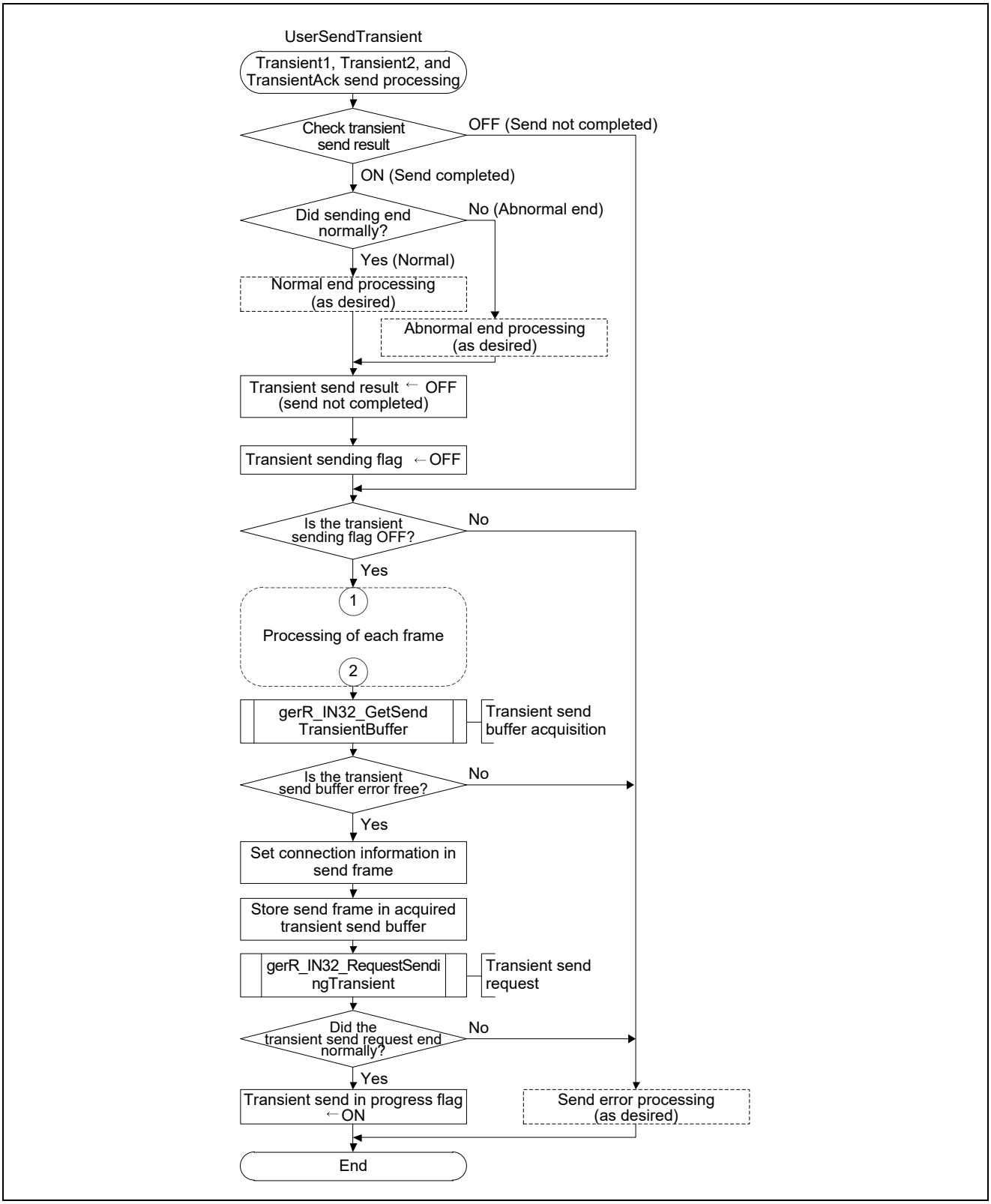


Figure 4.30Flowchart for Transient1, Transient2, and TransientAck Send Processing (1/2)

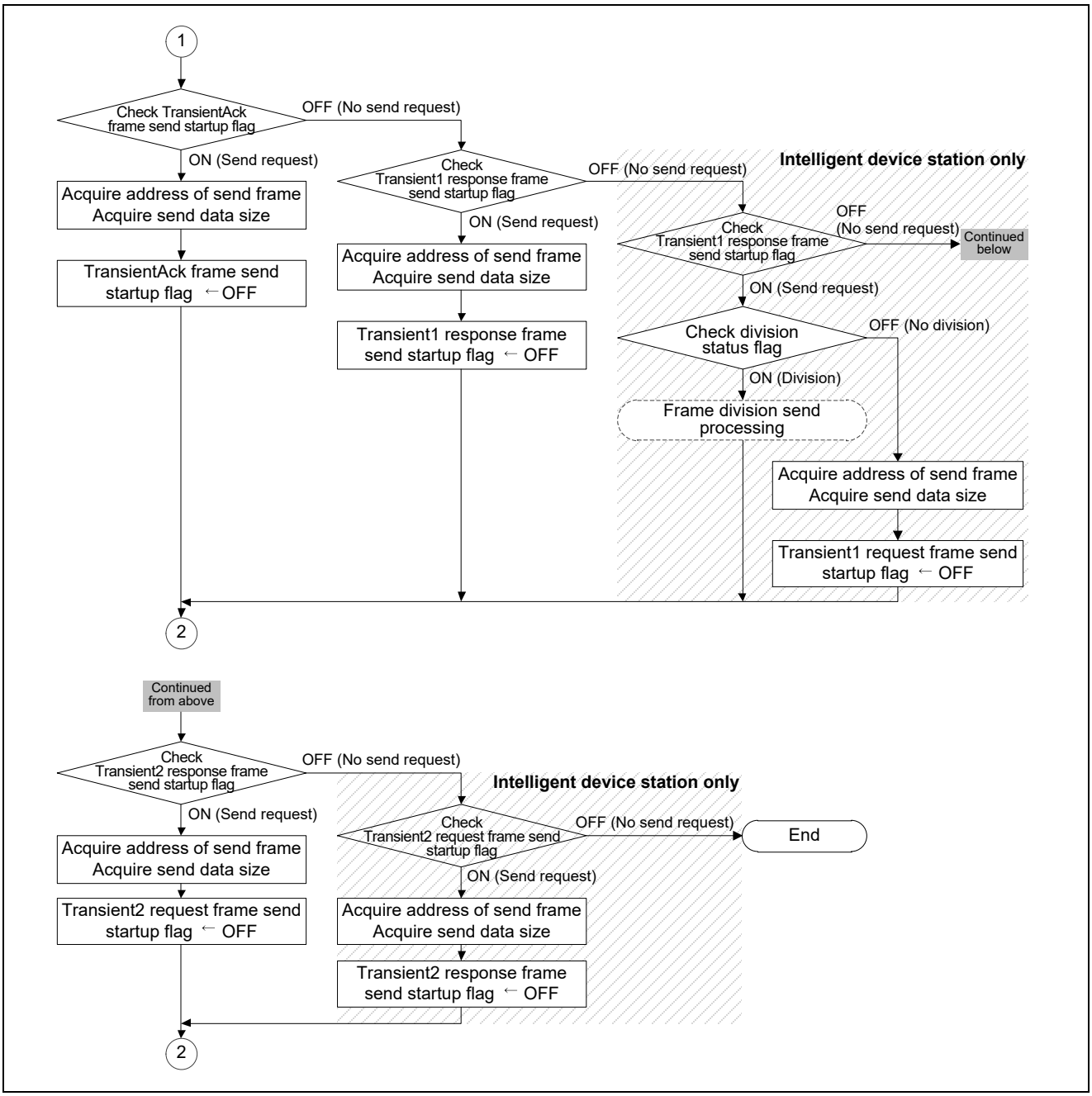


Figure 4.30Flowchart for Transient1, Transient2, and TransientAck Send Processing (2/2)

[Sending data by dividing data into blocks]

When the transient data requested to be sent is 1466 to 2048 bytes, the transient data can be divided and sent. Implement this processing in accordance with specifications of the R-IN32M4-CL3 application product. The following shows an image of the process for divided sending. For details regarding the Transient1 frame, see Section 7.2 "Transient1 Frame".

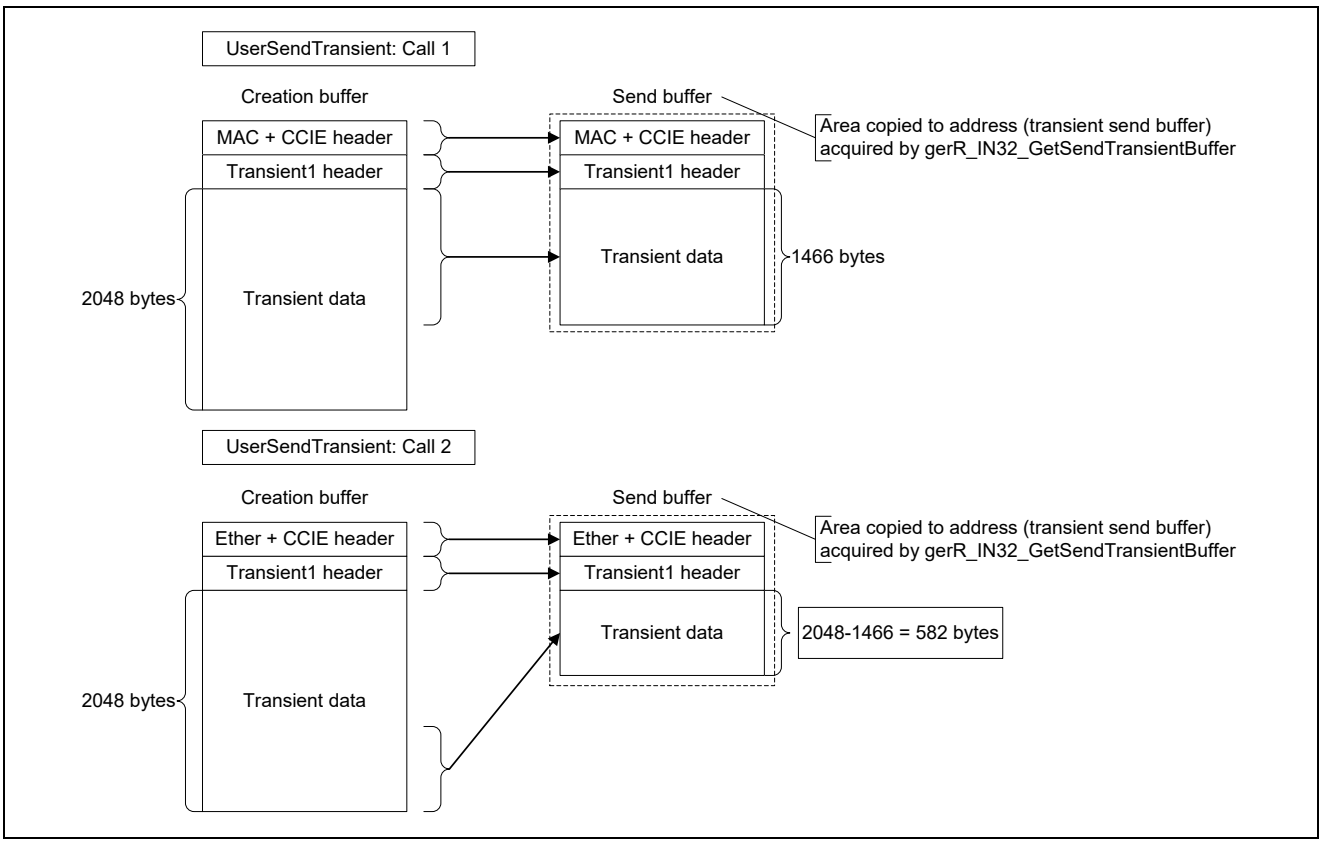


Figure 4.31When Dividing and Transferring Transient Data Having a Size of 2048 Bytes (When the Division Flag is ON)

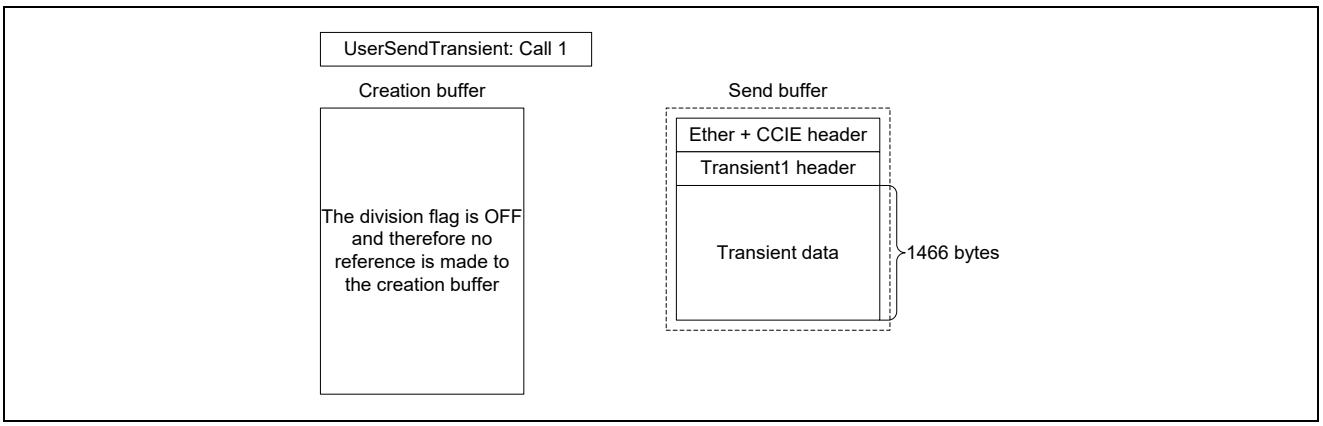


Figure 4.32When Dividing and Transferring Transient Data Having a Size of 1466 Bytes or Less (When the Division Flag is OFF)

4.3.15 SLMP Memory Read Request Frame Creation Processing

This function creates SLMP memory read request frame to be sent to another station. This processing is an example of the processing for creating SLMP request frame. For other commands, add processing as required.

Implement this processing only when its own station is an intelligent device station.

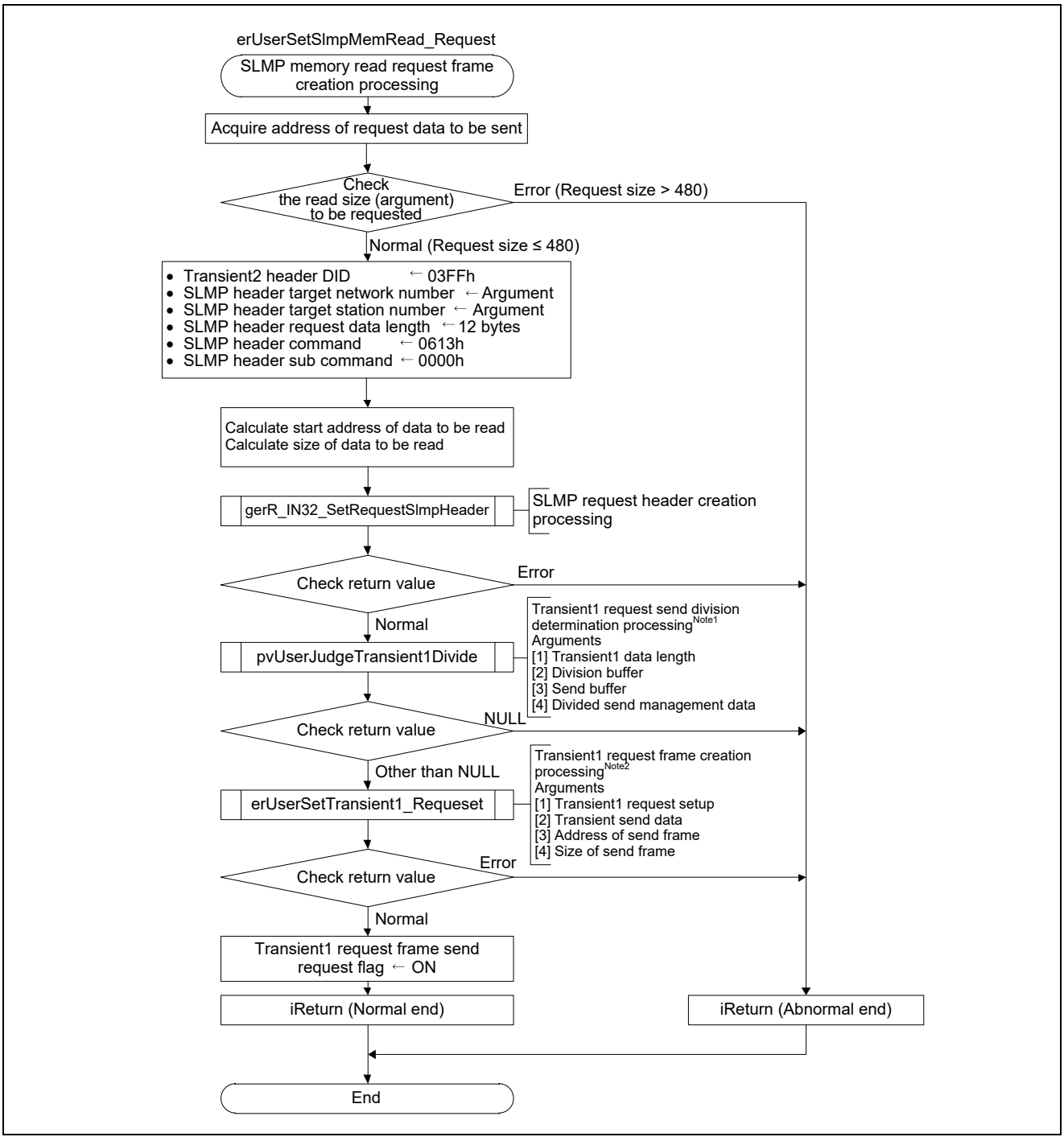


Figure 4.33Flowchart for SLMP Memory Read Request Frame Creation Processing

Notes 1. For details, see Section 4.3.16 "Transient1 Request Send Division Determination Processing".

2. For details, see Section 4.3.17 "Transient1 Request Frame Creation Processing".

4.3.16 Transient1 Request Send Division Determination Processing

This function determines if a frame should be divided prior to sending when creating a Transient1 request frame.

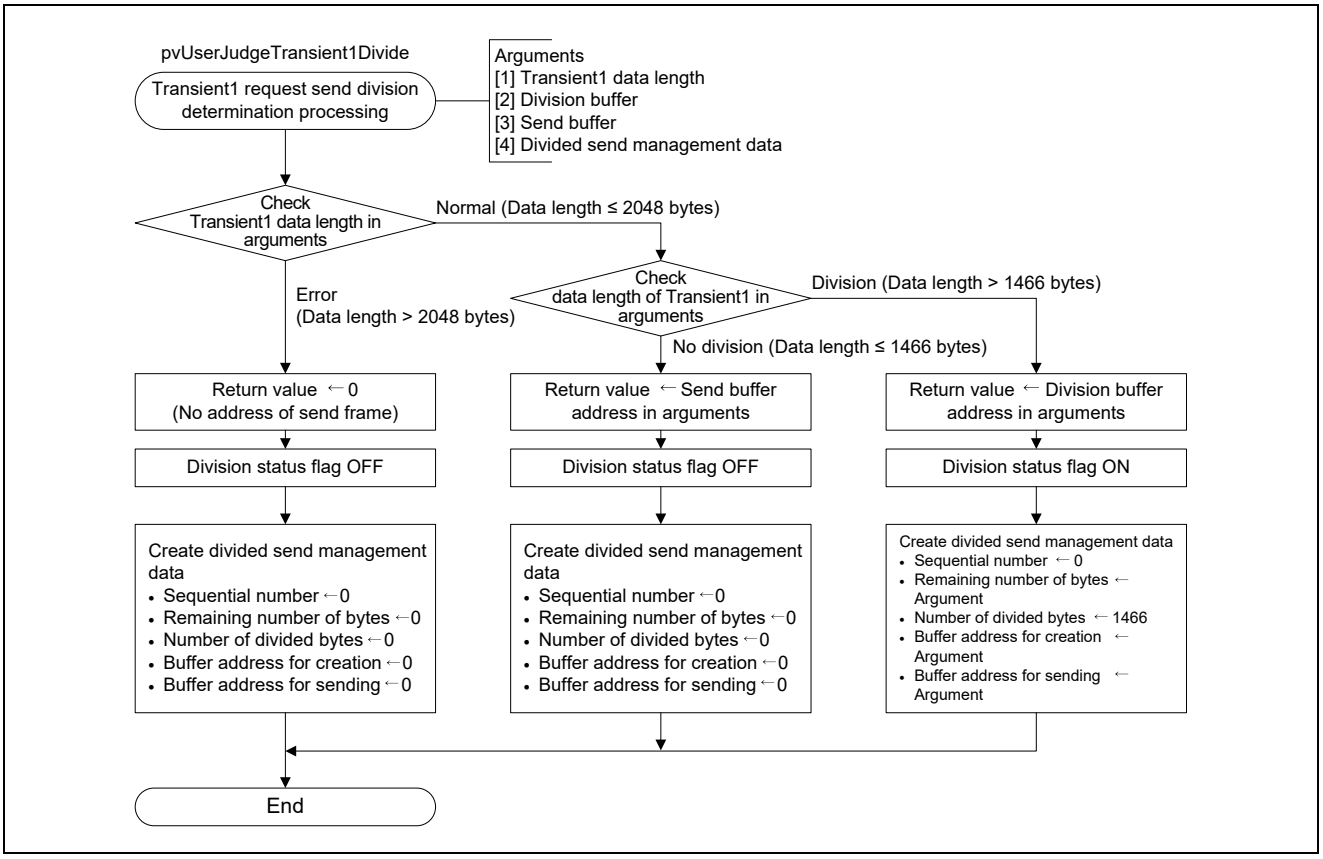


Figure 4.34Flowchart for Transient1 Request Send Division Determination Processing

4.3.17 Transient1 Request Frame Creation Processing

This function creates a request frame (from the MAC header to the Transient1 header) when an SLMP memory read request is sent from its own station to another station.

Implement this processing only when its own station is an intelligent device station.

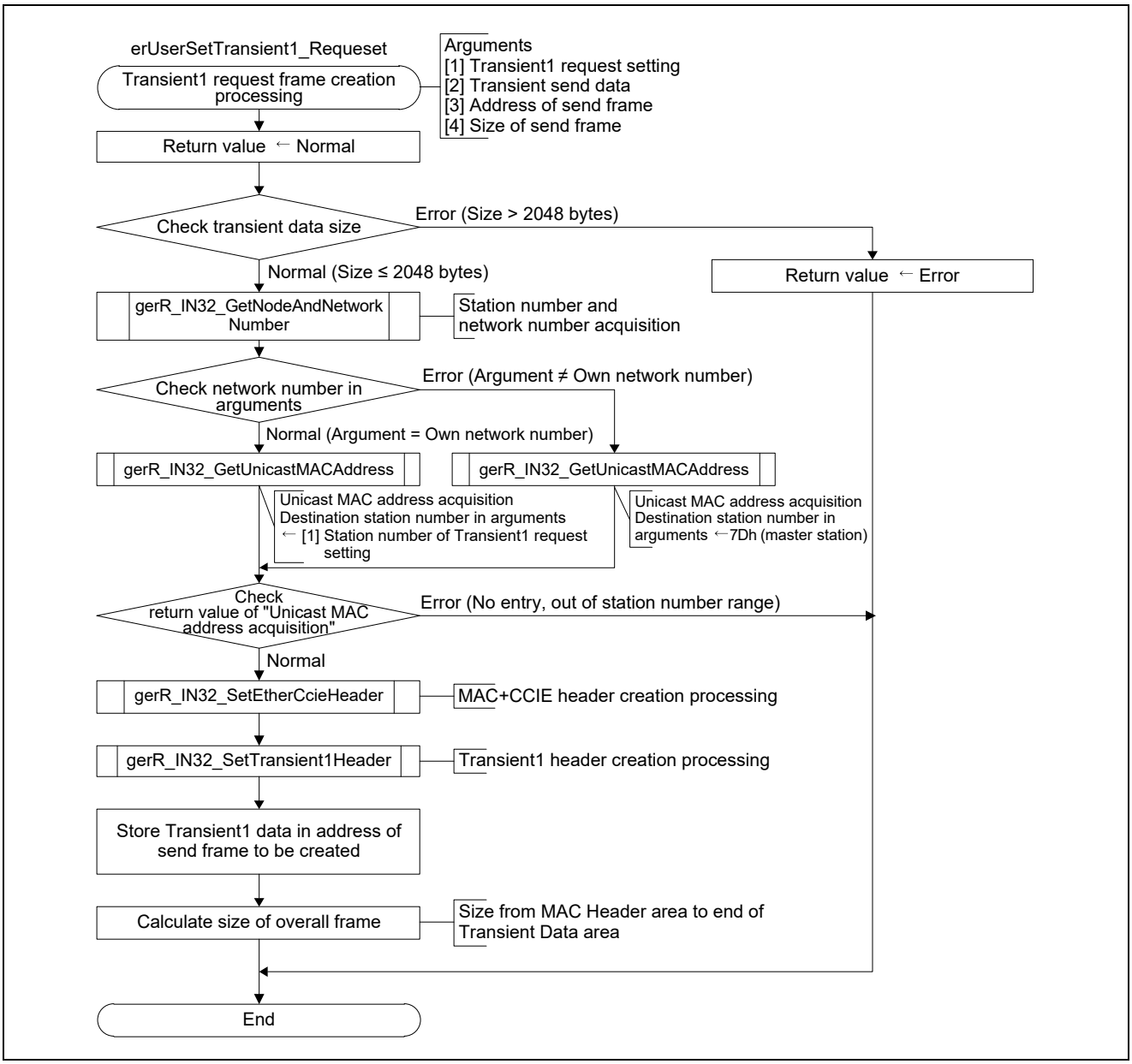


Figure 4.35Flowchart for Transient1 Request Frame Creation Processing

4.3.18 SLMP Memory Read Response Receive Processing

This function receives response frames for SLMP memory read requested by its own station to other stations.
Implement this processing only when its own station is an intelligent device station.

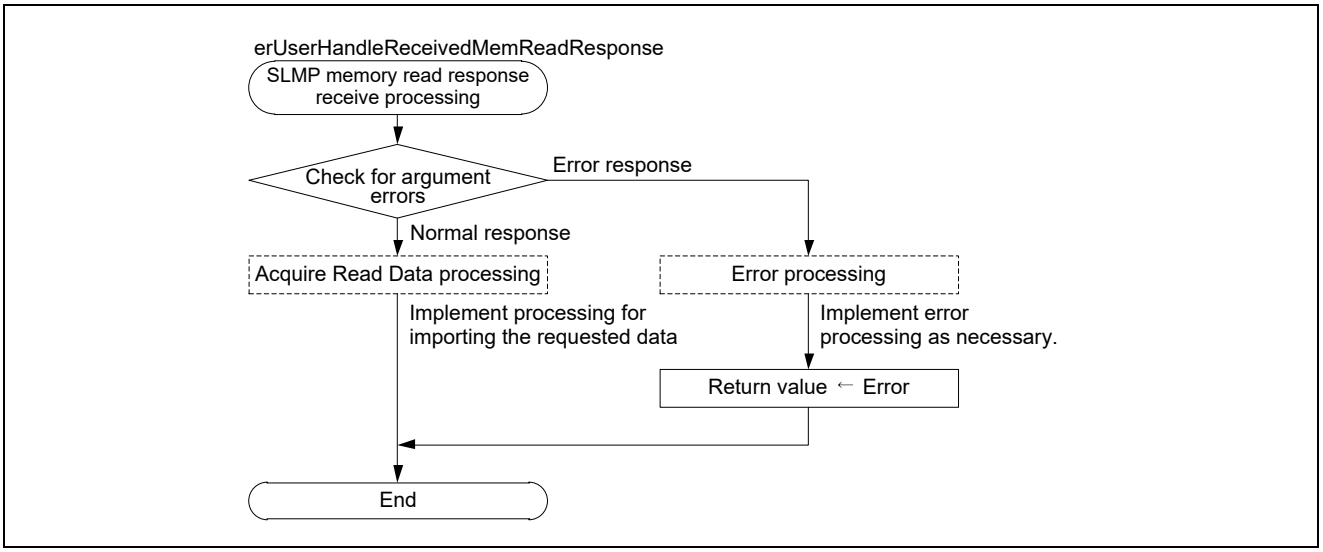


Figure 4.36Flowchart for SLMP Memory Read Response Receive Processing

4.3.19 Transient2 Request Frame Creation Processing

This function creates Transient2 request frame to be sent to another station. This processing is an example of the processing for creating Transient2 request frame. For other commands, add processing as required.

Implement this processing only when its own station is an intelligent device station.

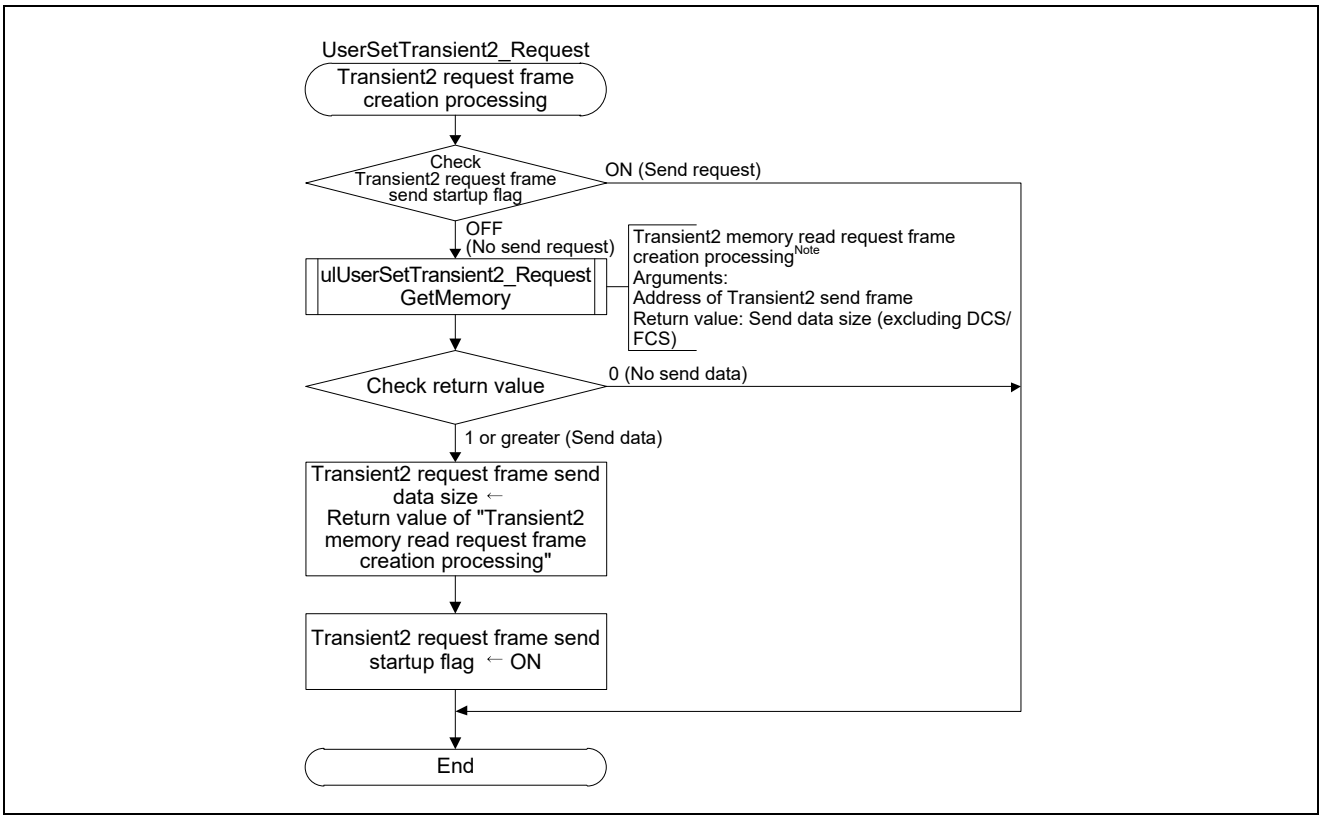


Figure 4.37Flowchart for Transient2 Request Frame Creation Processing

Note. For details, see Section 4.3.20 "Transient2 Memory Read Request Frame Creation Processing".

4.3.20 Transient2 Memory Read Request Frame Creation Processing

This function creates a request frame when a Transient2 memory read request is to be sent from its own station to another station. Implement this processing only when its own station is an intelligent device station.

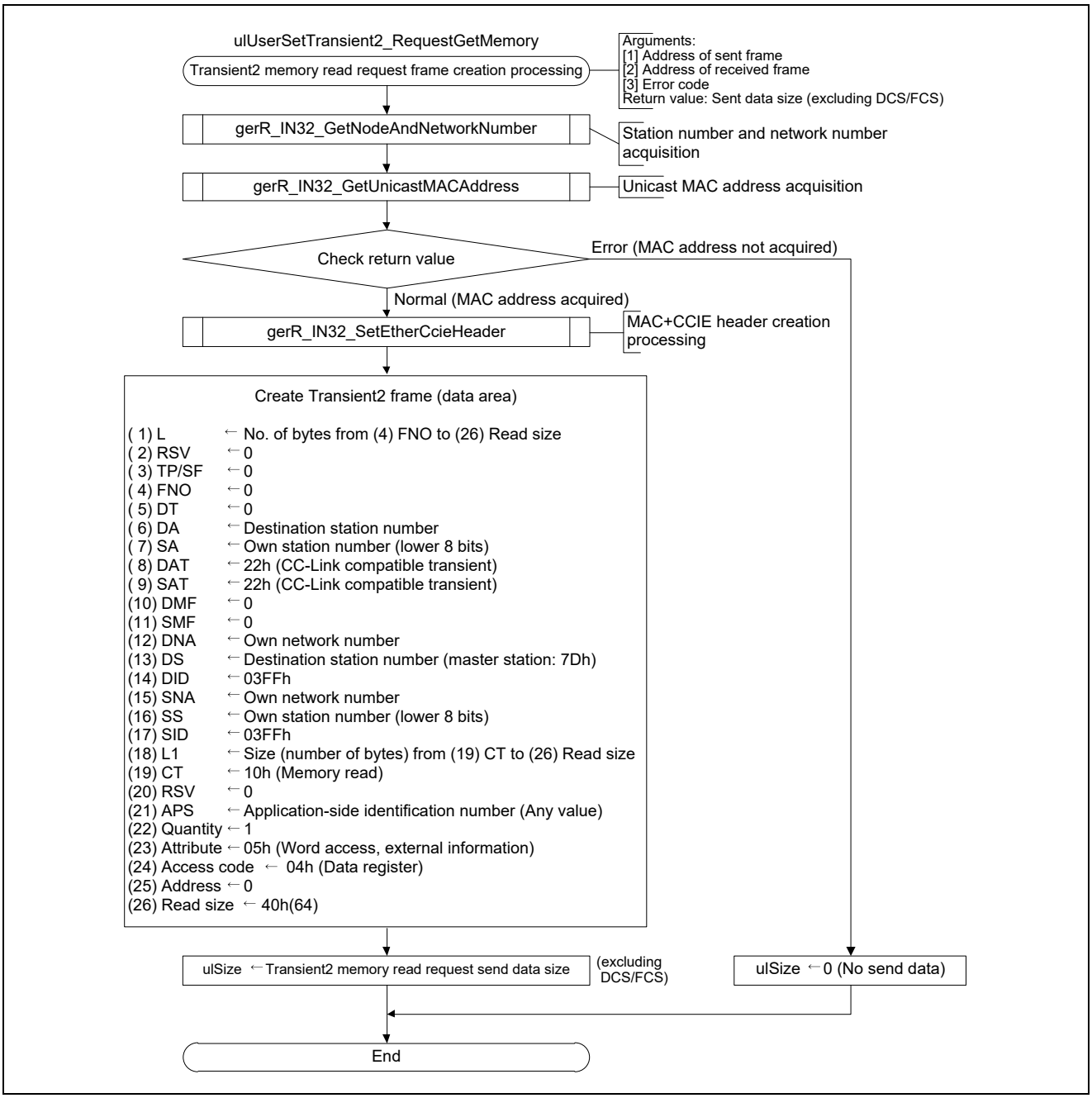


Figure 4.38Flowchart for Transient2 Memory Read Request Frame Creation Processing

The flow above illustrates the command processing for the settings below. Any other setup results in error.

(1) Destination station number: 7Dh (Master station)
(2) Access code: 04h (Data register)
(3) Attribute: 05h (Word access, external information)
(4) Address (start address of read destination): 0
(5) Read size: 40h (64)

4.3.21 Transient2 Memory Read Response Receive Processing

This function receives response frames for Transient2 memory read requested by its own station to another station. Implement this processing only when its own station is an intelligent device station.

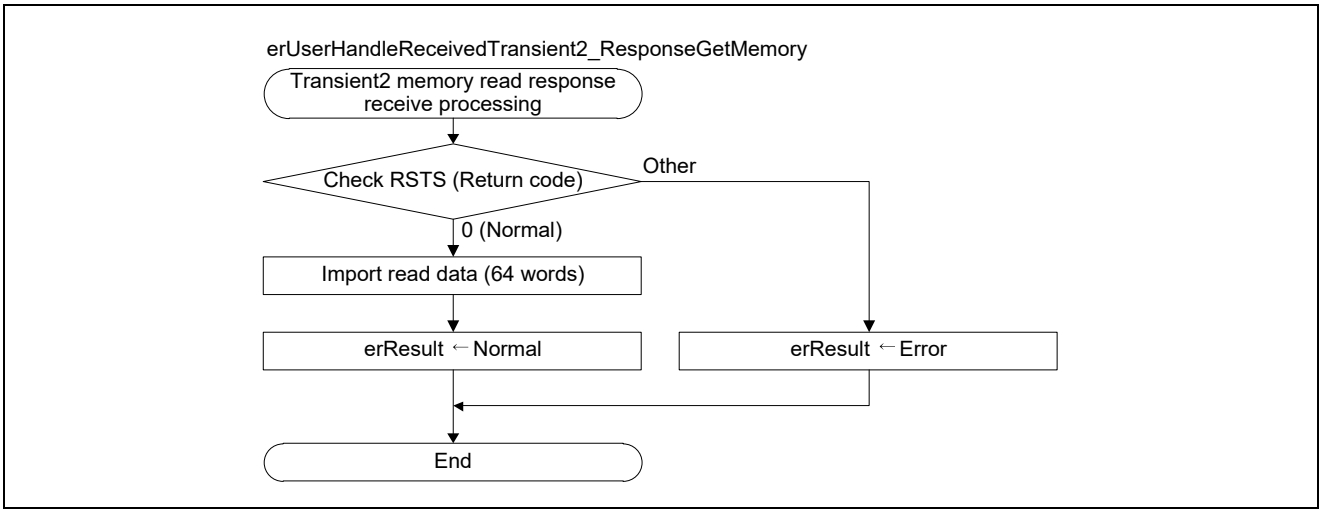


Figure 4.39Flowchart for Transient2 Memory Read Response Receive Processing

The flow above illustrates the command processing for the settings below. Any other setup results in error.

(1) Destination station number: 7Dh (Master station)
(2) Access code: 04h (Data register)
(3) Attribute: 05h (Word access, external information)
(4) Address (start address of read destination): 0
(5) Read size: 40h (64)

4.4 User Program Details (Hardware Test Related)

Implement the hardware test as independent processing, not as main processing (iUserMainRoutine). (Implementation examples: From outside (station number / network number switch or engineering tool), implement a function that switches to "Hardware Test Mode (Online Mode)" other than the normal operation mode (online mode).)

4.4.1 Hardware Test (IEEE802.3ab Compliance Test)

The hardware test needs to be implemented to implement the "1000BASE-T compliance test" of the conformance test.

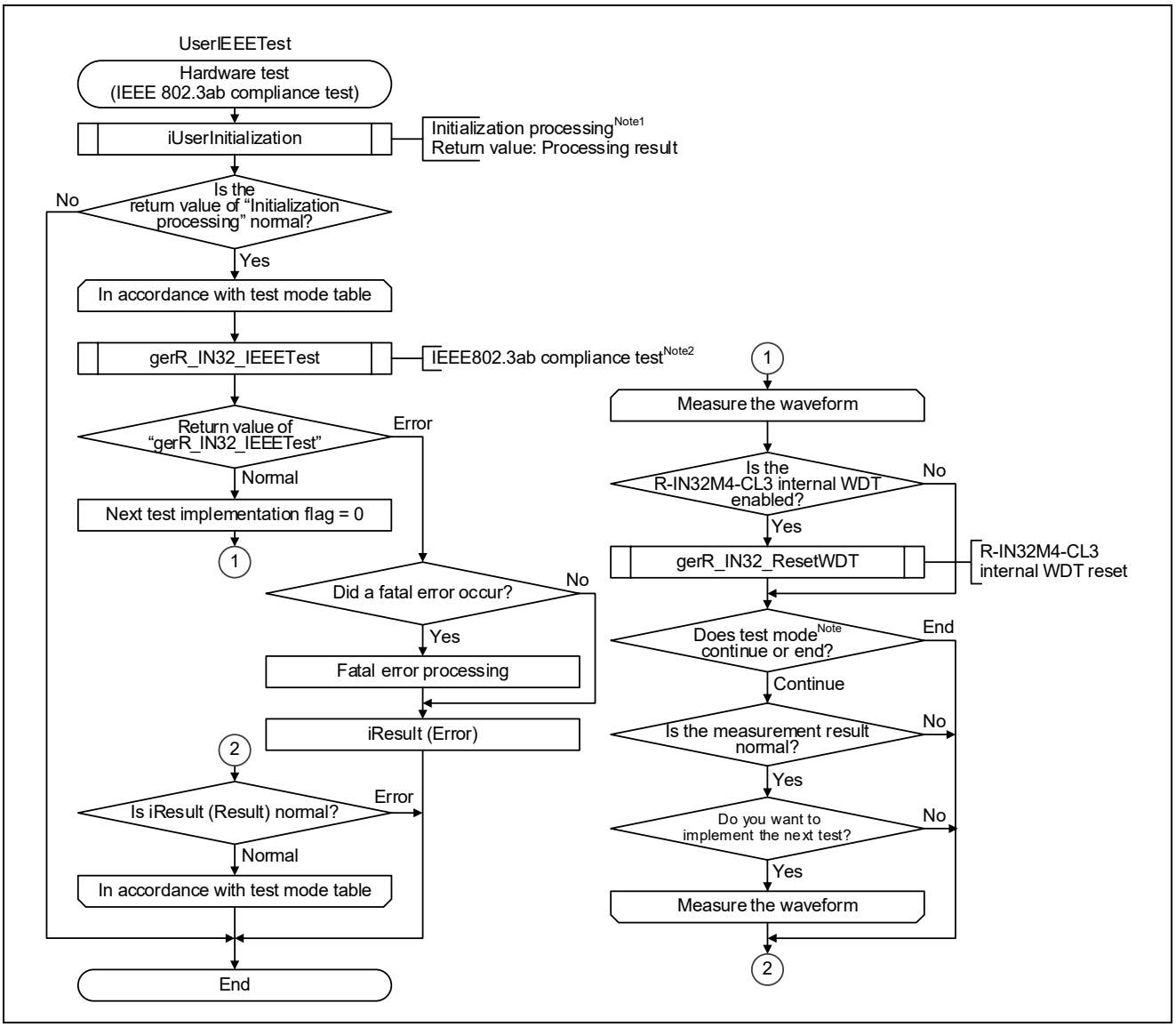


Figure 4.40Flowchart for Hardware Test (IEEE802.3ab Compliance Test)

Notes 1. For details, see Section 4.2.2 "Initialization Processing".

2. When a fatal error occurs in R-IN32M4-CL3, the gerR_IN32_IEEETest function calls the function below created by the user. Be sure to acquire the R-IN32M4-CL3 fatal error.

- **gR_IN32_CallbackFatalError**

4.4.2 Hardware Test (Loopback Communication Test)

The Hardware test (loopback communication test) checks if there is a hardware error in R-IN32M4-CL3.
Implement this test with Ethernet ports 1 and 2 connected by an Ethernet cable.

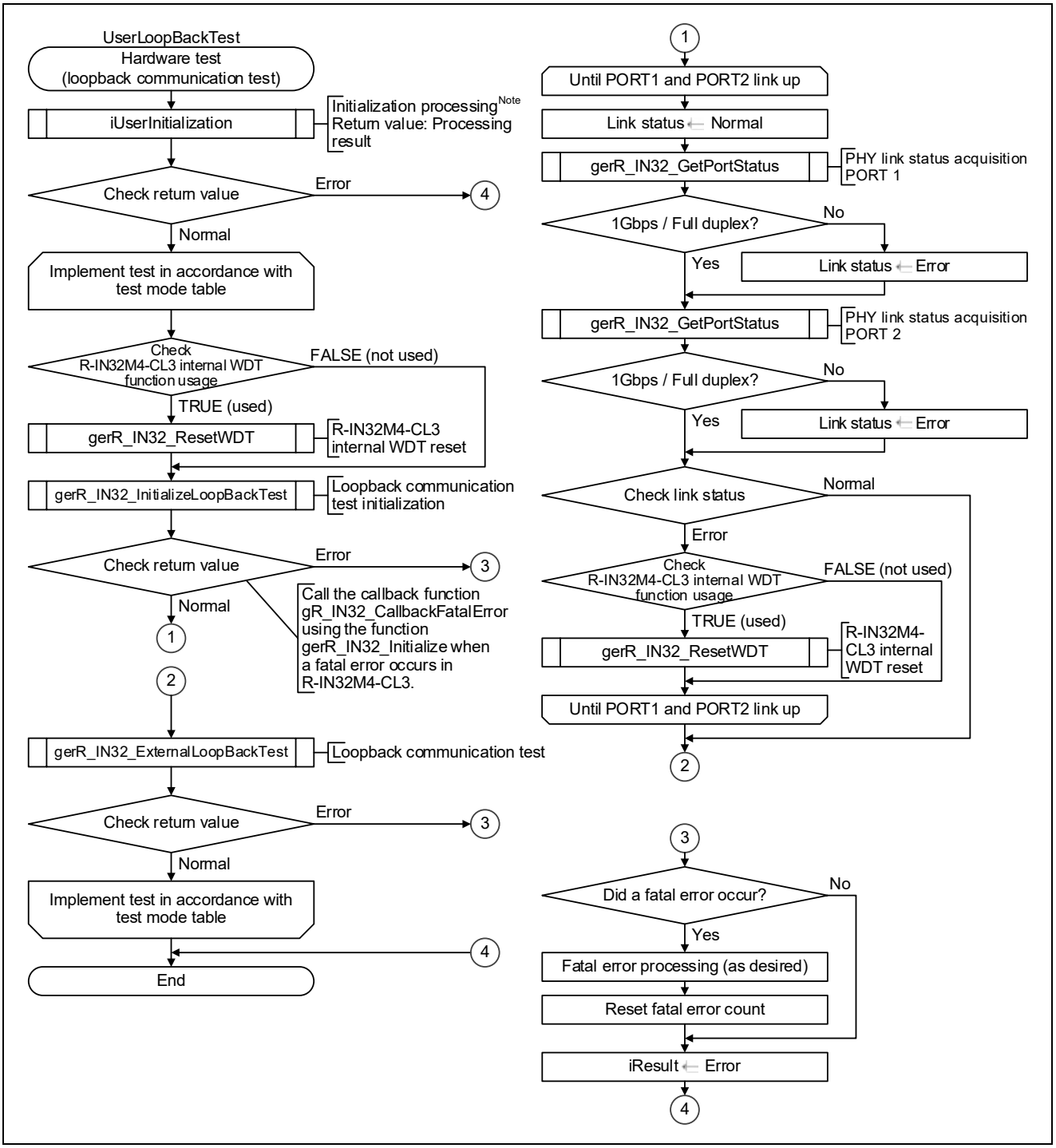


Figure 4.41Flowchart for Hardware Test (Loopback Communication Test)

Note. For details, see Section 4.2.2 "Initialization Processing".

5. R-IN32M4-CL3 DRIVER RELATED SPECIFATIONS

This section describes the specifications of the R-IN32M4-CL3 driver interface functions and R-IN32M4-CL3 driver callback functions that make up the R-IN32M4-CL3 driver.

5.1 Overview of Each Function

5.1.1 Overview

The table below provides an overview of each function and indicates whether or not function changes are required.

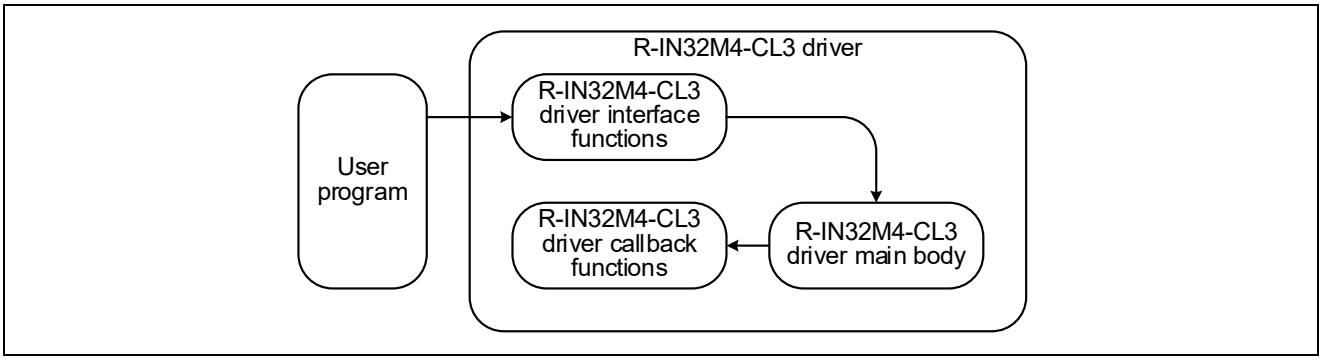


Figure 5.1 Relationship Between Functions

Table 5.1 Overview of Each Function

Program Part Name	Overview	Need for Change
R-IN32M4-CL3 driver interface functions	A function called when a function of the R-IN32M4-CL3 driver is used from the user program. (File: R_IN32_Interface.c)	×
R-IN32M4-CL3 driver callback functions	A function used when the user program requests callback from the R-IN32M4-CL3 driver. Describes the processing on the user program side for events that occur in the R-IN32M4-CL3 driver. (File: R_IN32M4_Callback.c)	O
R-IN32M4-CL3 driver main body	The main body of the driver area that is called by R-IN32M4-CL3 driver interface functions and controls R-IN32M4-CL3. (Files: Files below the driver folder excluding R_IN32_Interface.c.)	×

5.1.2 Type Definition and Error Code

The following indicates the types and error codes defined in the R-IN32M4-CL3 driver.

Table 5.2 R-IN32M4-CL3 Driver Type List

No.	Defined Type	Implementation
1	VOID	void
2	CHAR	char
3	UCHAR	unsigned char
4	SHORT	short
5	USHORT	unsigned short
6	INT	int
7	UINT	unsigned int
8	LONG	long
9	ULONG	unsigned long
10	ERRCODE	int
11	BOOL	int

Table 5.3 R-IN32M4-CL3 Driver Error Code List

No.	Symbol	Value	Description
1	R_IN32_OK	0	Normal
2	R_IN32_ERR	–1	Abnormal end (status error / mismatch)
3	R_IN32_ERR_OTHER	–2	(Error occurred in driver inside library)
4	R_IN32_ERR_OUTOFRANGE	–3	Out of range
5	R_IN32_ERR_EMPTY	–4	Empty
6	R_IN32_ERR_OVERFLOW	–5	Overflow
7	R_IN32_ERR_NOENTRY	–6	No entry
8	R_IN32_ERR_NOPERMIT	–7	Not permitted
9	R_IN32_ERR_NODATA	–8	No data
10	R_IN32_ERR_NOMYSTATUS	–9	No valid MyStatus

5.1.3 Description Specifications

The following describes the description specifications of each function.

Table 5.4 Source Code Description Specifications

Item	Description	Remarks
C language standard	ANSI C compliant	Partially uses extended compiler maker specifications.
Character encoding	ASCII, Japanese (Shift_JIS)	—
Tab length	4	—
Return code	CR+LF	—

5.2 R-IN32M4-CL3 Driver Interface Function List

The following lists the interface functions of the R-IN32M4-CL3 driver.

Table 5.5 R-IN32M4-CL3 Driver Interface Function List

(1/3)

Function Category (Reference Section)	Function Name	Function Type	Overview
Initial setup (Section 5.3.1 "Initial Setup")	gulR_IN32_GetResetStatus	ULONG	Reset status acquisition
	gerR_IN32_Initialize	ERRCODE	R-IN32M4-CL3 initialization
	gerR_IN32_SetNodeAndNetworkNumber	ERRCODE	Station number and network number setting
	gerR_IN32_Start	ERRCODE	R-IN32M4-CL3 communication start
WDT (Section 5.3.2 "Watchdog Timer")	gerR_IN32_ResetWDT	ERRCODE	R-IN32M4-CL3 internal WDT reset
	gerR_IN32_DisableWDT	ERRCODE	R-IN32M4-CL3 internal WDT disablement
	gerR_IN32_EnableWDT	ERRCODE	R-IN32M4-CL3 internal WDT enablement
	gerR_IN32_SetWDT	ERRCODE	R-IN32M4-CL3 internal WDT time limit setting
Event (Section 5.3.3 "Event")	gerR_IN32_GetEvent	ERRCODE	R-IN32M4-CL3 event detection
	gerR_IN32_Main	ERRCODE	R-IN32M4-CL3 event detection main processing
	gerR_IN32_RestartEvent	ERRCODE	R-IN32M4-CL3 event restart
	gerR_IN32_UpdateMIB	ERRCODE	MIB information update
Cyclic transmission (Section 5.3.4 "Cyclic Transmission")	gerR_IN32_SetCyclicStop	ERRCODE	Cyclic transmission stop for user application-side reasons
	gerR_IN32_ClearCyclicStop	ERRCODE	Cyclic transmission stop clear for user application-side reasons
	gerR_IN32_GetReceivedCyclicData	ERRCODE	Cyclic receive data acquisition
	gerR_IN32_GetMasterNodeStatus	ERRCODE	Master station status acquisition
	gerR_IN32_SetMyStatus	ERRCODE	MyStatus send data setting
	gerR_IN32_SetSendCyclicData	ERRCODE	Cyclic send data setting
Own station status setup (Section 5.3.5 "Own Station Status Setup")	gerR_IN32_SetNodeStatus	ERRCODE	Own station status setting
	gerR_IN32_ForceStop	ERRCODE	R-IN32M4-CL3 own station error setting
Own station status acquisition (Section 5.3.6 "Own Station Status Acquisition")	gerR_IN32_GetNodeAndNetworkNumber	ERRCODE	Station number and network number acquisition
	gerR_IN32_GetCurrentCyclicSize	ERRCODE	Acquisition of cyclic transmission size specified from master station
	gerR_IN32_GetCommumicationStatus	ERRCODE	Data link status acquisition
	gerR_IN32_GetPortStatus	ERRCODE	PHY link status acquisition
	gerR_IN32_GetCyclicStatus	ERRCODE	Cyclic transmission status acquisition
	gerR_IN32_GetMIB	ERRCODE	MIB information acquisition
	gerR_IN32_ClearMIB	ERRCODE	MIB information clear

Table 5.5 R-IN32M4-CL3 Driver Interface Function List

(2/3)

Function Category (Reference Section)	Function Name	Function Type	Overview
LED control (Section 5.3.7 "LED Control")	gerR_IN32_SetERRLED	ERRCODE	LED control (ERR)
	gerR_IN32_SetUSER1LED	ERRCODE	LED control (User LED 1)
	gerR_IN32_SetUSER2LED	ERRCODE	LED control (User LED 2)
	gerR_IN32_SetRUNLED	ERRCODE	LED control (RUN)
	gerR_IN32_DisableLED	ERRCODE	LED control function disablement
	gerR_IN32_EnableLED	ERRCODE	LED control function enablement
	gerR_IN32_UpdateLedStatus	ERRCODE	Communication status display LED update
Network time (Section 5.3.8 "Network Time")	gerR_IN32_GetNetworkTime	ERRCODE	Network time (serial value) acquisition
	gerR_IN32_SetNetworkTime	ERRCODE	Network time (serial value) setting
	gerR_IN32_NetworkTimeToDate	ERRCODE	Network time (serial value) to clock information conversion
	gerR_IN32_DateToNetworkTime	ERRCODE	Clock information to network time (serial value) conversion
MDIO access (Section 5.3.9 "MDIO Access")	gerR_IN32_EnableMACIPAccess	ERRCODE	MAC IP access enablement
	gerR_IN32_DisableMACIPAccess	ERRCODE	MAC IP access disablement
	gerR_IN32_WritePHY	ERRCODE	PHY internal register write
	gerR_IN32_ReadPHY	ERRCODE	PHY internal register read
	gerR_IN32_CheckPHY	ERRCODE	PHY check processing
Transient reception (Section 5.3.10 "Transient Reception")	gerR_IN32_MainReceiveTransient1	ERRCODE	Transient reception main processing 1
	gerR_IN32_MainReceiveTransient2	ERRCODE	Transient reception main processing 2
	gerR_IN32_EnableReceiveTransient	ERRCODE	Transient reception enable/disable setting for user reasons
	gblR_IN32_GetReceiveTransientStatus	BOOL	Status acquisition of transient reception enable/disable setting for user reasons
	gerR_IN32_SetMACAddressTableData	ERRCODE	Node information distribution data (MAC address table) setting

Table 5.5 R-IN32M4-CL3 Driver Interface Function List

(3/3)

Function Category (Reference Section)	Function Name	Function Type	Overview
Transient request reception (Section 5.3.11 "Transient Request Reception")	gerR_IN32_ReceivedMACAddressData	ERRCODE	Node information distribution frame receive processing
	gerR_IN32_ReceivedStatisticInfoRequest	ERRCODE	Statistical information acquisition request frame receive processing
	gerR_IN32_ReceivedUnitInfoRequest	ERRCODE	Detailed node information acquisition request frame receive processing
	gerR_IN32_ReceivedOptionInfoRequest	ERRCODE	Option information acquisition request frame receive processing
	gerR_IN32_ReceivedSelectInfoRequest	ERRCODE	Selected station information acquisition request frame receive processing
	gerR_IN32_ReceivedContactTestRequest	ERRCODE	Communication test request frame receive processing
	gerR_IN32_ReceivedCableTestRequest	ERRCODE	Cable test request frame receive processing
	gerR_IN32_ReceiveRemoteResetRequest	ERRCODE	SLMP remote reset request frame receive processing
	gR_IN32_SetSlmpError_Response	ULONG	SLMP error response frame creation processing
	gerR_IN32_ErrCheckReqFieldNetworkReceived	ERRCODE	CC-Link IE Field specific request receive frame error check processing
	gusR_IN32_ErrCheckReqSlmpReceived	USHORT	SLMP request receive frame error check processing
Transient send frame header creation (Section 5.3.12 "Transient Send Frame Header Creation")	gerR_IN32_SetEtherCcieHeader	ERRCODE	MAC+CCIE header creation processing
	gerR_IN32_SetTransient1Header	ERRCODE	Transient1 header creation processing
	gerR_IN32_SetRequestSlmpHeader	ERRCODE	SLMP request header creation processing
	gerR_IN32_SetResponseSlmpHeader	ERRCODE	SLMP response header creation processing
Transient send (Section 5.3.13 "Transient Send")	gerR_IN32_GetUnitInformation	ERRCODE	Unit information acquisition
	gusR_IN32_GetNodeID	USHORT	Node ID acquisition
	gerR_IN32_GetMulticastMACAddress	ERRCODE	Multicast MAC address acquisition
	gerR_IN32_GetUnicastMACAddress	ERRCODE	Unicast MAC address acquisition
	gerR_IN32_GetSendTransientBuffer	ERRCODE	Transient send buffer acquisition
	gerR_IN32_RequestSendingTransient	ERRCODE	Transient send request
	gerR_IN32_MainSendTransient	ERRCODE	Transient send main processing
Hardware test (Section 5.3.14 "Hardware Test")	gerR_IN32_IEEETest	ERRCODE	IEEE802.3ab compliance test
	gerR_IN32_InitializeLoopBackTest	ERRCODE	Loopback communication test initialization
	gerR_IN32_ExternalLoopBackTest	ERRCODE	Loopback communication test

5.3 R-IN32M4-CL3 Driver Interface Function Details

This section describes how to use the R-IN32M4-CL3 driver interface functions and the details of related functions.

5.3.1 Initial Setup

(1) gulR_IN32_GetResetStatus

Function	Reset status acquisition			
Call Format	ULONG gulR_IN32_GetResetStatus (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_RESET_PWRON(1): Power-on reset R_IN32_RESET_SYSTEM(2): System reset			
Description	This function acquires the reset status. Call this function before gerR_IN32_Initialize (Section 5.3.1(2)).			

(2) gerR_IN32_Initialize

Function	R-IN32M4-CL3 initialization			
Call Format	ERRCODE gerR_IN32_Initialize (const UCHAR* puchMACAddr, const R_IN32_UNITINFO_T *pstUnitInfo, const R_IN32_UNITINIT_T *pstUnitInit)			
Arguments	Name	Variable Name	Description	I/O
	const UCHAR	*puchMACAddr	Own station MAC address Set as follows for 12-34-56-78-90-AB: puchMACAddr[0]: 12h puchMACAddr[1]: 34h puchMACAddr[2]: 56h puchMACAddr[3]: 78h puchMACAddr[4]: 90h puchMACAddr[5]: ABh	Input
	const R_IN32_UNITINFO_T	*pstUnitInfo	R-IN32M4-CL3 unit information For details, see Table 5.6 "R_IN32_UNITINFO_T List".	Input
	const R_IN32_UNITINIT_T	*pstUnitInit	R-IN32M4-CL3 initial setup For details, see Table 5.7 "R_IN32_UNITINIT_T List".	Input
Return Value	R_IN32_OK: Normal end			
Description	This function performs R-IN32M4-CL3 initialization and PHY reset. Calling this function disables the R-IN32M4-CL3 internal WDT. When you want to use the R-IN32M4-CL3 internal WDT, call gerR_IN32_EnableWDT (Section 5.3.2(3)).			

Caution: When a fatal error occurs in R-IN32M4-CL3, this function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

The following describes the configuration of the argument R_IN32_UNITINFO_T of gerR_IN32_Initialize.

Table 5.6 R_IN32_UNITINFO_T List

(1/2)

No.	Member		Overview	Setting Description
1	ULONG	ulMaxRySize	RY size (bytes)	Specifies the RY size (bytes) communicable by its own station in increments of 1 byte. The maximum value for an intelligent device station is 256 bytes. The maximum value for a remote device station is 16 bytes.
2	ULONG	ulMaxRWwSize	RWw size (words)	Specifies the RWw size (words) communicable by its own station in increments of 2 words. The maximum value for an intelligent device station is 1,024 words. The maximum value for a remote device station is 64 words.
3	ULONG	ulMaxRxSize	RX size (bytes)	Specifies the RX size (bytes) communicable by its own station in increments of 1 byte. The maximum value for an intelligent device station is 256 bytes. The maximum value for a remote device station is 16 bytes.
4	ULONG	ulMaxRWrSize	RWr size (words)	Specifies the RWr size (words) communicable by its own station in increments of 2 words. The maximum value for an intelligent device station is 1, 024 words. The maximum value for a remote device station is 64 words.
5	ULONG	ulMyStationPort TotalNumber	No. of own station ports	Specifies the number of physical CC-Link IE Field Network ports of its own station. Set "2" or "1".
6	ULONG	ulTokenHoldTime;	Token hold time	Specifies the maximum time its own station holds a token after token passing begins, in µs. Set 23 µs.
7	ULONG	ulIOType	Node information (I/O type)	Specifies the I/O type. 00b indicates mixed, 01b indicates input, 10b indicates output, and 11b indicates composite. Mixed is used in a case when the input and output are mixed and the input and output use the same address. Composite is used in a case where the input and output are mixed and the input and output do not use the same address.
8	ULONG	ulNetVersion	Network firmware version	Specifies the firmware version of the network.
9	ULONG	ulNetModelType	Network model type	Specifies the model type (deviceType) specified by the CC-Link Partner Association.[Note]
10	ULONG	ulNetUnitModelCode	Network model code	Specifies the model code of the network. The model code is any code defined by the user. Manage the code so that it is unique within the same vendor code.
11	ULONG	ulNetVendorCode	Network vendor code	Specifies the vendor code (vendorCode) acquired when the vendor became a member of the CC-Link Partner Association, in BCD. (If the vendor code is 5678, 5678h is specified.)[Note]

Note. See Section 2.2 "Acquiring the Vendor Code and Selecting the Device Type".

Remark: Network: Refers to the communication section configured in R-IN32M4-CL3 and peripheral circuitry of its own station.
Controller: Refers to user-defined function sections (I/O section, sensor section, temperature adjustment section, etc.) of its own station.

Table 5.6 R_IN32_UNITINFO_T List

(2/2)

No.	Member		Overview	Setting Description
12	UCHAR	auchNetUnitModel Name[20]	Network model name	Specifies the model name of the network (in 20-byte character string (ASCII code)). The model name is any name defined by the user. Manage the name so that it is unique within the same vendor code.
13	UCHAR	auchNetVendorName [32]	Network vendor name	Specifies the vendor name of the network (in 32-byte character string (ASCII code)). The vendor name is any name (company name, brand name, etc.) defined by the user.
14	USHORT	usHwVersion	Network hardware version	Specifies the network hardware version.
15	USHORT	usDeviceVersion	Device version	Sets the device version (Version.) The device version (Version) indicates the version of the functions of the developed device. Used for associating CSP+ files.Note2
16	BOOL	blInformationFlag	Controller information status flag	Enables/Disables Nos. 14-21 in this table. R_IN32_False indicates disable, and R_IN32_True indicates enable. Disabled when there is only a communication function (example: QJ71GF11-T2).
17	ULONG	ulCtrlVersion	Controller firmware version	Specifies the firmware version of the controller.
18	ULONG	ulCtrlModelType	Controller model type	Specifies the model type (deviceType) specified by the CC-Link Partner Association.Note1
19	ULONG	ulCtrlUnitModelCode	Controller model name	Specifies the model name of the controller. The model name is any name defined by the user. Manage the name so that it is unique within the vendor code.
20	ULONG	ulCtrlVendorCode	Controller vendor code	Specifies the vendor code (vendorCode) acquired when the vendor became a member of the CC-Link Partner Association, in BCD. (If the vendor code is 5678, 5678h is specified.)Note1
21	UCHAR	auchCtrlUnitModel Name[20]	Controller model code	Specifies the model code of the controller (in 20-byte character string (ASCII code)). The model code is any code defined by the user. Manage the code so that it is unique within the same vendor code.
22	UCHAR	auchCtrlVendorName [32]	Controller vendor name	Specifies the vendor name of the controller (in 32-byte character string (ASCII code)). The vendor name is any name defined by the user.
23	ULONG	ulVendorInformation	Controller vendor device specific information	Specifies the vendor device specific information of the controller. The vendor device specific information is any information defined by the user.

Notes 1. See Section 2.2 "Acquiring the Vendor Code and Selecting the Device Type".

2. The device version of CSP+ is further described below. For details, refer to "DEVICE_INFO Part" in the "Control & Communication System Profile Specification".

Remark: Device Version: Supplemental Information

[Background]

When the software version of an R-IN32M4-CL3 application product is upgraded, specification changes sometimes occur, such as the addition of slave station parameter processing or command execution. When the specifications of an R-IN32M4-CL3 application product change, the CSP+ file also needs to be updated in accordance with the specification change.

[Purpose of device version]

The information that identifies the specifications before and after a change is the device version. The device version is used to indicate the specifications of the R-IN32M4-CL3 application product that correspond to each CSP+ file.

(a) Purpose of use by engineering tool
The engineering tool manages all CSP+ files having different device versions, making it possible to provide optimum functions and UI in accordance with the used version of the R-IN32M4-CL3 application product.

(b) Purpose of use by end user
The end user can select the CSP+ file for the device actually used upon comparing the device versions described in the CSP+ file and the version of the R-IN32M4-CL3 application product used.

The following shows the configuration of the argument "R_IN32_UNITINIT_T" of gerR_IN32_Initialize.

Table 5.7 R_IN32_UNITINIT_T List

(1/3)

No.	Member		Overview	Setting Description
1	BOOL	blNMIUse	NMI interrupt use	Specify "R_IN32_TRUE" when you want to use the R-IN32M4-CL3 internal WDT function, and "R_IN32_FALSE" when you do not. Specifying "R_IN32_TRUE" changes the CCI_NMIZ to "Low" when the R-IN32M4-CL3 internal WDT overflows.
2	BOOL	blInterruptUse	CPU interrupt function use	Specify "R_IN32_TRUE" when you want to use the R-IN32M4-CL3 CPU interrupt function, and "R_IN32_FALSE" when you do not. Specifying "R_IN32_TRUE" changes the CCI_INTZ to "Low" when an R-IN32M4-CL3 interrupt occurs.
3	BOOL	blFailedProcess1	Failed process setting 1	Specify "R_IN32_TRUE". When any of the signals below are true, R-IN32M4-CL3 changes to bypass mode. (Communication frames are neither sent nor received. A received frame is forwarded as is to the other port.) [1] When the WDTIL signal is True (Low) [2] When the R-IN32M4-CL3 internal WDT times out When you want to clear bypass mode, power-on reset or system reset is required.
4	BOOL	blFailedProcess2	Failed process setting 2	Specify "R_IN32_TRUE". When a forced stop is executed (gerR_IN32_ForceStop), R-IN32M4-CL3 changes to bypass mode. (Communication frames are neither sent nor received. A received frame is forwarded as is to the other port.) When you want to clear a forced stop, power-on reset or system reset is required. For gerR_IN32_ForceStop details, see Section 5.3.5 (2) "Own Station Status Setup".
5	ULONG	ulNodeType	Node type	Specifies the node type of its own station. Specify (33h) for an intelligent device station. Specify (34h) for a remote device station.
6	BOOL	blTransient ReceiveEnable	Transient reception function	Specify "R_IN32_TRUE". Specifies whether or not the transient reception function is present. R_IN32_TRUE indicates the function is present, and R_IN32_FALSE indicates the function is not present.

Table 5.7 R_IN32_UNITINIT_T List

(2/3)

No.	Member		Overview	Setting Description
7	BOOL	blMACAddress TableRequest	Initial value of Node information distribution request	[When intelligent device station] Set this to "R_IN32_TRUE" when a transient transmission client function is implemented, and to "R_IN32_FALSE" when it is not. [When remote device station] A transient transmission client function is not present. Set this to "R_IN32_FALSE". [Node information] Node information indicates the correspondence between the MAC addresses and station numbers of other stations. When this is set to "R_IN32_TRUE", node information is distributed from the master station by multicast. When this is set to "R_IN32_FALSE", discard the received Node information distribution frames using the user program. [When Transient2 frames are sent] When transient frames are actively sent (with client), node information is used. When a response is returned to the send source (with server), the response can be returned using the send source MAC address, and therefore node information is not used.
8	ULONG	ulRunStatus	Initial value of detailed application operation status	Specifies the initial value of the detailed application operation status within nodeStatus of the MyStatus frame. R_IN32_RUNSTS_UNSUPPORTED (0000h): Detailed application operation status notification not supported R_IN32_RUNSTS_STOP (0001h): Application stopped R_IN32_RUNSTS_RUN (0002h): Application running R_IN32_RUNSTS_NOTEXIST (0003h): Application substance does not exist
9	ULONG	ulErrorStatus	Initial value of detailed application error status	Sets the initial value of the detailed application error status of the nodeStatus field of the MyStatus frame. R_IN32_ERRSTS_NONE (0000h): No error R_IN32_ERRSTS_WARNING (0001h): Minor error R_IN32_ERRSTS_ERROR (0002h): Moderate error R_IN32_ERRSTS_FATALERROR (0003h): Major error
10	ULONG	ulUserInformation	Initial value of vendor specific node information	Specifies the initial value of vendorSpfNodeInfo of the MyStatus frame.
11	ULONG	ulOptionSupport	Initial value of option status	Set this to "R_IN32_TRUE" (recommended) when options are supported, and to "R_IN32_FALSE" when options are not supported. [Option] An option is an extended function of CC-Link IE Field Network, and includes the SLMP frame send/receive function and CC-Link IE Field Network diagnostic function.

Table 5.7 R_IN32_UNITINIT_T List

(3/3)

No.	Member		Overview	Setting Description
12	ULONG	ulSlmpSupport	Initial value of SLMP support bit	Set this to "R_IN32_TRUE" (recommended) when SLMP frames are sent and received, and to "R_IN32_FALSE" when they are not.Note1
13	ULONG	ulSlmpDiagnosis Support	Initial value of diagnostic function support	Set this to "R_IN32_TRUE" (recommended) when the CC-Link IE Field Network diagnostic function is supported, and to "R_IN32_FALSE" when it is not.Note2
14	R_IN32_PHY_ SETTING_T	stPHYSetting[2]	Initial value of PHY setting	Set MDI/MDI-X and master/slave for the respective ports. For details, see Table 5.8 "R_IN32_PHY_SETTING_T List".

Notes 1. To send and receive SLMP frames, set both this and the "Initial value of option status" to "R_IN32_TRUE".

2. To support the CC-Link IE Field Network diagnostic function, set this as well as the "Initial value of option status" and the "Initial value of SLMP support bit" to "R_IN32_TRUE".

Table 5.8 R_IN32_PHY_SETTING_T List

No.	Member		Overview	Setting Description
1	ULONG	ulMDI	MDI setting	Set MDI/MDI-X for the respective ports. When fast linkup is to be enabled, set R_IN32_MDI_FORCED_MDI for port 1 and R_IN32_MDI_FORCED_MDIX for port 2. R_IN32_MDI_AUTO (0000h): Automatic R_IN32_MDI_FORCED_MDI (0001h): Forced MDI R_IN32_MDI_FORCED_MDIX (0002h): Forced MDI-X
2	ULONG	ulClk	1000BASE-T clock setting	Set master/slave for the respective ports. When fast linkup is to be enabled, set R_IN32_CLOCK_MASTER for port 1 and R_IN32_CLOCK_SLAVE for port 2. When this function is to be disabled, set R_IN32_CLOCK_AUTO. R_IN32_CLOCK_AUTO (0000h): Automatic R_IN32_CLOCK_MASTER (0001h): Forced master R_IN32_CLOCK_SLAVE (0002h): Forced slave

(3) gerR_IN32_SetNodeAndNetworkNumber

Function	Station number and network number setting			
Call Format	ERRCODE gerR_IN32_SetNodeAndNetworkNumber (UCHAR uchNetworkNumber, USHORT usNodeNumber)			
Arguments	Name	Variable Name	Description	I/O
	UCHAR	uchNetworkNumber	Network number (value range: 1-239)	Input
	USHORT	usNodeNumber	Station number (value range: 1-120)	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end (error status in library) R_IN32_ERR_OUTOFRANGE: Station number out of range or network number out of range			
Description	This function sets the station number and network number in R-IN32M4-CL3. When the return value is R_IN32_ERR_OUTOFRANGE, the station number and network number are not set. Add error processing to the call source function.			

Caution: This function needs to be called after iUserInitialization (Section 4.2.2 "Initialization Processing") before calling gerR_IN32_Start (Section 5.3.1(4)) by iUserStart (Section 4.2.3 "Communication Start Processing"). When this function is called before the above processing, an R_IN32_ERR: abnormal end (status error in library) occurs.

(4) gerR_IN32_Start

Function	R-IN32M4-CL3 communication start			
Call Format	ERRCODE gerR_IN32_Start(VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end			
Description	This function provides instructions to start communication to R-IN32M4-CL3.			

Caution: When a fatal error occurs in R-IN32M4-CL3, this function calls gR_IN32_CallbackFatalError created by the user. Be sure to execute error processing in accordance with the error code.

5.3.2 Watchdog Timer

(1) gerR_IN32_ResetWDT

Function	R-IN32M4-CL3 internal WDT reset			
Call Format	ULONG gerR_IN32_ResetWDT (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function resets the R-IN32M4-CL3 internal WDT.			

Caution: If you want to call a function within Section 5.3.2 "Watchdog Timer " after this function is called, wait 1.032 µs or longer.

(2) gerR_IN32_DisableWDT

Function	R-IN32M4-CL3 internal WDT disablement			
Call Format	ULONG gerR_IN32_DisableWDT (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function disables the R-IN32M4-CL3 internal WDT.			

Caution: If you want to call a function within Section 5.3.2 "Watchdog Timer" after this function is called, wait 1.032 µs or longer. R-IN32M4-CL3 enables the R-IN32M4-CL3 internal WDT immediately after reset. (Initial value of R-IN32M4-CL3 internal WDT time limit setting: 3.2 s.) The R-IN32M4-CL3 internal WDT is disabled in gerR_IN32_Initialize (Section 5.3.1(2)). Implement one of the following when the period until startup of gerR_IN32_Initialize takes time:

- **Call this function to disable the R-IN32M4-CL3 internal WDT.**
- **Call gerR_IN32_ResetWDT (Section 5.3.2(1)) to reset the R-IN32M4-CL3 internal WDT. (Make sure that the R-IN32M4-CL3 internal WDT does not time out.)**

(3) gerR_IN32_EnableWDT

Function	R-IN32M4-CL3 internal WDT enablement			
Call Format	ULONG gerR_IN32_EnableWDT (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function enables the R-IN32M4-CL3 internal WDT.			

Caution: If you want to call a function within Section 5.3.2 "Watchdog Timer" after this function is called, wait 1.032 µs or longer. R-IN32M4-CL3 disables the R-IN32M4-CL3 internal WDT when gerR_IN32_Initialize (Section 5.3.1(2)) is called. Be sure to implement this function when you want to use the R-IN32M4-CL3 internal WDT.

(4) gerR_IN32_SetWDT

Function	R-IN32M4-CL3 internal WDT time limit setting			
Call Format	ULONG gerR_IN32_SetWDT (USHORT usWDTCOUNT)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usWDTCOUNT	R-IN32M4-CL3 internal WDT time limit setting 0000h: 100ms 0001h: 200ms 0002h: 300ms … 0001Fh: 3.2 s	Input
Return Value	R_IN32_OK: Normal end			
Description	This function sets the R-IN32M4-CL3 internal WDT time limit.			

Caution: If you want to call a function in this section after this function is called, wait 1.032 µs or longer. If the R-IN32M4-CL3 internal WDT time limit setting is changed by this function while the R-IN32M4-CL3 internal WDT is running (after gerR_IN32_EnableWDT (Section 5.3.2(3)) is called), the R-IN32M4-CL3 internal WDT runs using the new time limit setting when gerR_IN32_ResetWDT (Section 5.3.2(1)) is called. (Until gerR_IN32_ResetWDT is called, the R-IN32M4-CL3 internal WDT runs using the R-IN32M4-CL3 internal WDT time limit setting prior to the change.)

5.3.3 Event

(1) gerR_IN32_GetEvent

Function	R-IN32M4-CL3 event detection			
Call Format	ERRCODE gerR_IN32_GetEvent (R_IN32_EVTPRM_INTERRUPT_T *pstEvent)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_EVTPRM_INTERRUPT_T	*pstEvent	Interrupt cause For details, see Table 5.9 "R_IN32_EVTPRM_INTERRUPT_T List".	Output
Return Value	R_IN32_OK: Normal end			
Description	This function detects R-IN32M4-CL3 events.			

The following describes the configuration of R_IN32_EVTPRM_INTERRUPT_T.

Table 5.9 R_IN32_EVTPRM_INTERRUPT_T List

No.	Member		Overview
1	ULONG	b1ZCommConnect	Connect communication
2	ULONG	b1ZCommDisconnect	Disconnect communication
3	ULONG	b1ZCommConnectToDisconnect	Connect communication → Disconnect communication
4	ULONG	b1ZCommDisconnectToConnect	Disconnect communication → Connect communication
5	ULONG	b1ZChangeStNoNetNo	Change station number and network number
6	ULONG	b1ZChangeActCommand	Change run command
7	ULONG	b1ZPrmFrmRcv_OK	Parameter frame reception
8	ULONG	b1ZReserve1	Reserved
9	ULONG	b1ZPrmChkFrmRcv_OK	ParamCheck frame reception (when parameters match)
10	ULONG	b3ZReserve2	Reserved
11	ULONG	b1ZRecvNonCyclic	Transient reception
12	ULONG	b1ZSendFinNonCyclic	Transient send complete
13	ULONG	b7ZReserve3	Reserved
14	ULONG	b1ZMasterWatchTimeout	Master watch timer timeout occurred
15	ULONG	bAZReserve4	Reserved

(2) gerR_IN32_Main

Function	R-IN32M4-CL3 event detection main processing			
Call Format	ERRCODE gerR_IN32_Main (const R_IN32_EVTPRM_INTERRUPT_T *pstEvent)			
Arguments	Name	Variable Name	Description	I/O
	const R_IN32_EVTPRM_INTERRUPT_T	*pstEvent	Interrupt cause	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end (status error in library)			
Description	This function performs processing in response to an R-IN32M4-CL3 event.			

Caution: This function needs to be called after IUserInitialization (Section 4.2.2 "Initialization Processing") and iUserStart (Section 4.2.3 "Communication Start Processing"). Calling this function before executing the above processing results in an R_IN32_ERR (abnormal end; status error in library).

(3) gerR_IN32_RestartEvent

Function	R-IN32M4-CL3 event restart			
Call Format	ERRCODE gerR_IN32_RestartEvent (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function restarts events stopped by gerR_IN32_GetEvent (Section 5.3.3(1)).			

(4) gerR_IN32_UpdateMIB

Function	MIB information update			
Call Format	ERRCODE gerR_IN32_UpdateMIB (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end [MIB information collection error (status error in library / mismatch)] R_IN32_ERR_OTHER: Abnormal end [MIB information collection error (error occurred in driver inside library)]			
Description	This function updates the MIB information.			

Caution: When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

5.3.4 Cyclic Transmission

(1) gerR_IN32_SetCyclicStop

Function	Cyclic transmission stop for user application-side reasons			
Call Format	ERRCODE gerR_IN32_SetCyclicStop (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function stops cyclic transmission for device-side reasons. If you want to clear the stop status, call the function gerR_IN32_ClearCyclicStop (Section 5.3.4(2)).			

(2) gerR_IN32_ClearCyclicStop

Function	Cyclic transmission stop clear for user application-side reasons			
Call Format	ERRCODE gerR_IN32_ClearCyclicStop (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function clears cyclic transmission stop that was called by the function gerR_IN32_SetCyclicStop (Section 5.3.4(1)).			

(3) gerR_IN32_GetReceivedCyclicData

Function	Cyclic receive data acquisition			
Call Format	ERRCODE gerR_IN32_GetReceivedCyclicData (VOID *pRyDst, VOID *pRWwDst, BOOL blEnable)			
Arguments	Name	Variable Name	Description	I/O
	VOID	*pRyDst	RY area	Output
	VOID	*pRWwDst	RWw area Note1	Output
	BOOL	blEnable	Enables/Disables copying. R_IN32_TRUE: Enable R_IN32_FALSE: Disable	Input
Return Value	R_IN32_OK: Normal end (received data present) R_IN32_ERR: Abnormal end (no received data)			
Description	This function stores cyclic receive data from the master station in the addresses indicated by pRyDst and pRWwDst. Note, however, that when blEnable is set to R_IN32_FALSE, the cyclic receive data is discarded. The return value changes to R_IN32_ERR.Note2			

Notes 1. Set the head address of the RWw area in increments of 4 bytes (0 or multiple of 4).

2. R_IN32_ERR: Abnormal end (no received data)
While an R_IN32_ERR occurs when no cyclic communication is received from the previous call of gerR_IN32_GetReceivedCyclicData to the current call of gerR_IN32_GetReceivedCyclicData, this does not indicate an error.

(4) gerR_IN32_GetMasterNodeStatus

Function	Master station status acquisition			
Call Format	ERRCODE gerR_IN32_GetMasterNodeStatus (BOOL *pblRunSts, BOOL *pblErrSts, ULONG *pulErrCode)			
Arguments	Name	Variable Name	Description	I/O
	BOOL	*pblRunSts	State of operation of the master station application R_IN32_TRUE: Running R_IN32_FALSE: Stopped	Output
	BOOL	*pblErrSts	Error state of the master station application R_IN32_TRUE: Error R_IN32_FALSE: No error	Output
	ULONG	*pulErrCode	Master station error code	Output
Return Value	R_IN32_OK: Normal end (MyStatus frame received from master station) R_IN32_ERR: Abnormal end [MyStatus frame not received from master station due to no data link (data link disconnected)]			
Description	This function acquires the status of the master station from the MyStatus frame received from the master station. When the MyStatus frame is not received from the master station due to no data link (data link disconnected), the arguments are as follows: pblRunSts : R_IN32_FALSE pblErrSts: R_IN32_FALSE pulErrCode: 0			

(5) gerR_IN32_SetMyStatus

Function	MyStatus send data setting			
Call Format	ERRCODE gerR_IN32_SetMyStatus (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function sets its own station status specified by gerR_IN32_SetNodeStatus in R-IN32M4-CL3 (Section 5.3.5(1)).			

(6) gerR_IN32_SetSendCyclicData

Function	Cyclic send data setting			
Call Format	ERRCODE gerR_IN32_SetSendCyclicData (const VOID *pRxSrc, const VOID *pRWwSrc, BOOL blEnable)			
Arguments	Name	Variable Name	Description	I/O
	const VOID	*pRxSrc	RX area	Input
	const VOID	*pRWwSrc	RWw area Note	Input
	BOOL	blEnable	Enables/Disables update. R_IN32_TRUE: Enable R_IN32_FALSE: Disable	Input
Return Value	R_IN32_OK: Normal end			
Description	This function sets the cyclic send data stored in the addresses indicated in pRxSrc and pRWwSrc in R-IN32M4-CL3. Note, however, that when blEnable is set to R_IN32_FALSE, cyclic send data is not set. (The return value changes to R_IN32_ERR.)			

Note. Set the head address of the RWw area in increments of 4 bytes (0 or multiple of 4).

5.3.5 Own Station Status Setup

(1) gerR_IN32_SetNodeStatus

Function	Own station status setting			
Call Format	ERRCODE gerR_IN32_SetNodeStatus (ULONG ulRunSts, ULONG ulErrSts, ULONG ulErrCode, ULONG ulUserInformation)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulRunSts	Detailed application operation status R_IN32_RUNSTS_UNSUPPORTED(0): Detailed application operation status notification not supported R_IN32_RUNSTS_STOP(1): Application stopped R_IN32_RUNSTS_RUN(2): Application running R_IN32_RUNSTS_NOTEXIST(3): Application substance does not exist	Input
	ULONG	ulErrSts	Detailed application error status R_IN32_ERRSTS_NONE(0): No error R_IN32_ERRSTS_WARNING(1): Minor error R_IN32_ERRSTS_ERROR(2): Moderate error R_IN32_ERRSTS_FATALERROR(3): Major error	Input
	ULONG	ulErrCode	Error code	Input
	ULONG	ulUserInformation	Vendor specific node information	Input
Return Value	R_IN32_OK: Normal end			
Description	This function sets its own station status as information to be sent in a MyStatus frame.			

(2) gerR_IN32_ForceStop

Function	R-IN32M4-CL3 own station error setting			
Call Format	ERRCODE gerR_IN32_ForceStop (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function sets an own station error in R-IN32M4-CL3. To clear the own station error, power-on reset or system reset is required.			

5.3.6 Own Station Status Acquisition

(1) gerR_IN32_GetNodeAndNetworkNumber

Function	Station number and network number acquisition			
Call Format	ERRCODE gerR_IN32_GetNodeAndNetworkNumber (USHORT *pusNodeNumber, UCHAR *puchNetworkNumber)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	*pusNodeNumber	Station number	Output
	UCHAR	*puchNetworkNumber	Network number	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the station number and network number.			

(2) gerR_IN32_GetCurrentCyclicSize

Function	Acquisition of cyclic transmission size specified from master station			
Call Format	ERRCODE gerR_IN32_GetCurrentCyclicSize (R_IN32_CYCLIC_SIZE_T *pstCyclicSize)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_CYCLIC_SIZE_T	*pstCyclicSize	Cyclic transmission size ulRySize: RY size (bytes) ulRWwSize: RWw size (bytes) ulRxSize: RX size (bytes) ulRWrSize: RWr size (bytes)	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the cyclic transmission size specified from the master station in the parameter frame. The functions gerR_IN32_GetReceivedCyclicData (Section 5.3.4(3)) and gerR_IN32_SetSendCyclicData (Section 5.3.4(6)) input and output cyclic send and reception data in the size acquired by this function.			

(3) gerR_IN32_GetCommumicationStatus

Function	Data link status acquisition			
Call Format	ERRCODE gerR_IN32_GetCommumicationStatus (ULONG *pulCommSts)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	*pulCommSts	Data link status R_IN32_COMMSTS_CYC_DLINK(2): Data link in progress (cyclic transmission in progress) R_IN32_COMMSTS_TOKEN_PASS(1): Data link in progress (cyclic transmission stopped) R_IN32_COMMSTS_DISCONNECT(0): No data link (disconnected)	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the data link status. Turn the D LINK LED on/off according to the data link status. R_IN32_COMMSTS_CYC_DLINK(2): LED on R_IN32_COMMSTS_TOKEN_PASS(1): LED blinking R_IN32_COMMSTS_DISCONNECT(0): LED off			

Caution: For D LINK LED on/off control, see UserUpdateLed (Section 4.2.11 "LED Update Processing").

(4) gerR_IN32_GetPortStatus

Function	PHY link status acquisition				
Call Format	ERRCODE gerR_IN32_GetPortStatus (ULONG ulPort, ULONG *pulLinkStatus, ULONG *pulSpeed, ULONG *pulDuplex)				
Arguments	Name	Variable Name	Description		I/O
	ULONG	ulPort	Port specification	R_IN32_PORT1(0): PORT1 R_IN32_PORT2(1): PORT2	Input
	ULONG	*pulLinkStatus	Link status	R_IN32_LINKUP(1): Link up R_IN32_LINKDOWN(0): Link down	Output
	ULONG	*pulSpeed	Speed Note	R_IN32_SPEED_1G(0): 1Gbps R_IN32_SPEED_100M(1): 100Mbps R_IN32_SPEED_10M(2): 10Mbps	Output
	ULONG	*pulDuplex	Full duplex / Half duplex Note	R_IN32_DUPLEX_FULL(0): Full duplex R_IN32_DUPLEX_HALF(1): Half duplex	Output
Return Value	R_IN32_OK: Normal end				
Description	This function acquires the PHY link status. (When pulLinkState is LinkDown: Don't care.)				

Note. Enabled when the second argument *pulLinkStatus is R_IN32_LINKUP (1). Do not use this when the second argument is R_IN32_LINKDOWN (0).

(5) gerR_IN32_GetCyclicStatus

Function	Cyclic transmission status acquisition			
Call Format	ULONG gerR_IN32_GetCyclicStatus (R_IN32_CYCLIC_STA_T *pstCyclicStatus)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_CYCLIC_STA_T	*pstCyclicStatus	Cyclic transmission status For details, see Table 5.10 "R_IN32_CYCLIC_STA_T List".	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the cyclic transmission status.			

The following describes the configuration of R_IN32_CYCLIC_STA_T.

Table 5.10 R_IN32_CYCLIC_STA_T List

No.	Member		Bit	Overview	Description
1	USHORT	b3ZComonParamkeepCond	b2-0	Cyclic transmission parameter hold status	001b: Parameter normally received 010b: Not received or ID mismatch 011b: Checking 100b: Parameter abnormally received
2	USHORT	b1ZParamCheckCond	b3	Cyclic transmission parameter check status	0b: Checked 1b: Checking
3	USHORT	b1ZMyNodeNoRangeOut	b4	Station number invalid setting status	0b: In range 1b: Out of range
4	USHORT	b1ZMyNodeReserveSetup	b5	Reserved station setting status	0b: Non-reserved node 1b: Reserved node
5	USHORT	b1ZCyclicOpeInstructPackage	b6	Cyclic transmission implementation instruction (batch) setting status	0b: Run 1b: Stop
6	USHORT	b1ZCyclicOpeInstructVarious	b7	Cyclic transmission implementation instruction (individual) setting status	0b: Run 1b: Stop
7	USHORT	b1ZReserved1	b8	Reserved	—
8	USHORT	b1ZMyMpuAbnomal	b9	Cyclic transmission continuation not possible error status	0b: No error 1b: Cyclic transmission not possible error
9	USHORT	b1ZMyNodeNumberDuplicate	b10	Station number duplication status	0b: No duplication 1b: Duplication
10	USHORT	b1ZReserved2	b11	Reserved	—
11	USHORT	b1ZNodeTypeWrong	b12	Station type invalid / Specified size invalid status	0b: Normal 1b: Invalid
12	USHORT	b1ZReserved3	b13	Reserved	—
13	USHORT	b1ZDLinkState	b14	Disconnection status	0b: Not disconnected (cyclic transmission in progress or token passing in progress) 1b: Disconnected
14	USHORT	b1ZCyclicState	b15	Stop status due to own reasons	0b: Not stopped 1b: Cyclic transmission stopped due to reason other than the above

(6) gerR_IN32_GetMIB

Function	MIB information acquisition			
Call Format	ERRCODE gerR_IN32_GetMIB (R_IN32_MIB_T *pstMIB)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_MIB_T	*pstMIB	R-IN32M4-CL3 MIB information	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires MIB information.			

(7) gerR_IN32_ClearMIB

Function	MIB information clear			
Call Format	ERRCODE gerR_IN32_ClearMIB (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function clears MIB information.			

Caution: MIB information is non-disclosed information. Do not disclose this information to the end user.

5.3.7 LED Control

(1) gerR_IN32_SetERRLED

Function	LED control (ERR)			
Call Format	ERRCODE gerR_IN32_SetERRLED (ULONG ulCtrl)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulCtrl	LED control R_IN32_LED_OFF: LED off R_IN32_LED_ON: LED on R_IN32_LED_BLINK: LED blinking	Input
Return Value	R_IN32_OK: Normal end			
Description	This function turns on and off the ERR LED.			

Caution: The LED cannot be turned off or set to blinking when an R-IN32M4-CL3 internal WDT, external WDT, or own station error occurs.

(2) gerR_IN32_SetUSER1LED

Function	LED control (User LED 1)			
Call Format	ERRCODE gerR_IN32_SetUSER1LED (ULONG ulCtrl)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulCtrl	LED control R_IN32_LED_OFF: LED off R_IN32_LED_ON: LED on R_IN32_LED_BLINK: LED blinking	Input
Return Value	R_IN32_OK: Normal end			
Description	This function turns on and off User LED 1.			

Caution: The LED cannot be turned on or set to blinking when an R-IN32M4-CL3 internal WDT, external WDT, or own station error occurs.

(3) gerR_IN32_SetUSER2LED

Function	LED control (User LED 2)			
Call Format	ERRCODE gerR_IN32_SetUSER2LED (ULONG ulCtrl)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulCtrl	LED control R_IN32_LED_OFF: LED off R_IN32_LED_ON: LED on R_IN32_LED_BLINK: LED blinking	Input
Return Value	R_IN32_OK: Normal end			
Description	This function turns on and off User LED 2.			

Caution: The LED cannot be turned on or set to blinking when an R-IN32M4-CL3 internal WDT, external WDT, or own station error occurs.

(4) gerR_IN32_SetRUNLED

Function	LED control (RUN)			
Call Format	ERRCODE gerR_IN32_SetRUNLED (ULONG ulCtrl)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulCtrl	LED control R_IN32_LED_OFF: LED off R_IN32_LED_ON: LED on	Input
Return Value	R_IN32_OK: Normal end			
Description	This function turns on and off the RUN LED.			

Caution: The LED cannot be turned on when an R-IN32M4-CL3 internal WDT, external WDT, or own station error occurs.

(5) gerR_IN32_DisableLED

Function	LED control function disablement			
Call Format	ERRCODE gerR_IN32_DisableLED (USHORT usBitPattern)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usBitPattern	LED control function disablement (ON: Disable, OFF: Hold previous value) Bit 0: Disable RUN LED Bit 2: Disable User LED 2 Bit 4: Disable User LED 1 Bit 6: Disable D LINK LED Bit 8: Disable ERR. LED Bit10: Disable port 1 L ER LED Bit11: Disable port 2 L ER LED (Bits 1, 3, 5, 7, 9, and 12-15: Not used)	Input
Return Value	R_IN32_OK: Normal end			
Description	This function disables the LED function.			

Caution: The function cannot be disabled when an R-IN32M4-CL3 internal WDT, external WDT, or own station error occurs.

(6) gerR_IN32_EnableLED

Function	LED control function enablement			
Call Format	ERRCODE gerR_IN32_EnableLED (USHORT usBitPattern)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usBitPattern	LED control function enablement (ON: Enable, OFF: Hold previous value) Bit 0: Enable RUN LED Bit 2: Enable User LED 2 Bit 4: Enable User LED 1 Bit 6: Enable D LINK LED Bit 8: Enable ERR. LED Bit10: Enable port 1 L ER LED Bit11: Enable port 2 L ER LED (Bits 1, 3, 5, 7, 9, and 12-15: Not used)	Input
Return Value	R_IN32_OK: Normal end			
Description	This function enables the LED function.			

(7) gerR_IN32_UpdateLedStatus

Function	Communication status display LED update			
Call Format	ERRCODE gerR_IN32UpdateLedStatus (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function controls the D LINK LED and ERR. LED of its own station in accordance with the data link status of the argument acquired by the Data link status acquisition processing (Section 5.3.6(3) gerR_IN32_GetCommumicationStatus).			

Data Link Status	D LINK LED	ERR. LED
Data link in progress (cyclic transmission in progress)	On	Off
Data link in progress (cyclic transmission stopped)	Blinking	Off

In addition, the function controls the L ER1 LED and L ER2 LED in accordance with the result of token passing during ring connection.

Token Passing	L ER1 LED	L ER2 LED
Token not passed (disconnected)	Off	Off
Token passing in progress and both ports enabled	Off	Off
Token passing in progress and only PORT1 enabled	Off	Off
Token passing in progress and only PORT2 enabled	Off	Off

5.3.8 Network Time

(1) gerR_IN32_GetNetworkTime

Function	Network time (serial value) acquisition			
Call Format	ERRCODE gerR_IN32_GetNetworkTime (USHORT *pusSerial)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	*pusSerial	Network time pusSerial[0]: Network time (bits 15-0) pusSerial[1]: Network time (bits 31-16) pusSerial[2]: Network time (bits 47-32)	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the network time (serial value in increments of 15.2587890625 µs given a starting point of January 1, 2000, 00:00:00).			

(2) gerR_IN32_SetNetworkTime

Function	Network time (serial value) setting			
Call Format	ERRCODE gerR_IN32_SetNetworkTime (const USHORT *pusSerial)			
Arguments	Name	Variable Name	Description	I/O
	const USHORT	*pusSerial	Network time pusSerial[0]: Network time (bits 15-0) pusSerial[1]: Network time (bits 31-16) pusSerial[2]: Network time (bits 47-32)	Input
Return Value	R_IN32_OK: Normal end			
Description	This function sets the network time (serial value in increments of 15.2587890625 µs given a starting point of January 1, 2000, 00:00:00).			

(3) gerR_IN32_NetworkTimeToDate

Function	Network time (serial value) to clock information conversion			
Call Format	ERRCODE gerR_IN32_NetworkTimeToDate (R_IN32_TIMEINFO_T *pstTimeInfo, const USHORT *pusSerial)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_TIMEINFO_T	*pstTimeInfo	Clock information For details, see Table 5.11 "R_IN32_TIMEINFO_T List".	Output
	const USHORT	*pusSerial	Network time pusSerial[0]: Network time (bits 31-16) pusSerial[1]: Network time (bits 47-32)	Input
Return Value	R_IN32_OK: Normal end			
Description	This function converts the network time (serial value in increments of seconds given a starting point of January 1, 2000, 00:00:00) to clock information [year/month/day/hour/minute/second/microsecond (fixed to 0)/day of the week].			

The following describes the configuration of R_IN32_TIMEINFO_T.

Table 5.11 R_IN32_TIMEINFO_T List

No.	Member		Overview	Remarks
1	USHORT	usYear	Year (2000 - 2136)	—
2	USHORT	usMonth	Month (1-12)	—
3	USHORT	usDay	Day (1-31)	—
4	USHORT	usHour	Hour (0-23)	—
5	USHORT	usMin	Minute (0-59)	—
6	USHORT	usSec	Second (0-59)	—
7	USHORT	usMsec	Millisecond (0-999)	—
8	USHORT	usWday	Day of week (0 (Sunday) - 6 (Saturday))	—

(4) gerR_IN32_DateToNetworkTime

Function	Clock information to network time (serial value) conversion			
Call Format	ERRCODE gerR_IN32_DateToNetworkTime (const R_IN32_TIMEINFO_T *pstTimeInfo, USHORT *pusSerial)			
Arguments	Name	Variable Name	Description	I/O
	const R_IN32_TIMEINFO_T	*pstTimeInfo	Clock information	Input
	USHORT	*pusSerial	Network time pusSerial[0]: Network time (bits 15-0) pusSerial[1]: Network time (bits 31-16) pusSerial[2]: Network time (bits 47-32)	Output
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end			
Description	This function converts clock information (year/month/day/hour/minute/second) to network time (serial value in increments of seconds given a starting point of January 1, 2000, 00:00:00). (ausSerial[0]: Network time (bits 15-0) is fixed to 0.)			

Caution: A year other than 2000-2136 results in an R_IN32_ERR. The R-IN32M4-CL3 driver does not check for any errors other than the above. Implement error processing in the user program to ensure that there are no leap year or date errors.

5.3.9 MDIO Access

(1) gerR_IN32_EnableMACIPAccess

Function	MAC IP access enablement			
Call Format	ERRCODE gerR_IN32_EnableMACIPAccess (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end (MDIO command end wait error)			
Description	This function enables MAC IP access.			

Caution: Shorten the period from MAC IP access enablement to MAC IP access disablement (gerR_IN32_DisableMACIPAccess (Section 5.3.9(2)) to the extent possible. (If the user uses interrupts, use the function with the interrupts disabled from MAC IP access enablement to MAC IP access disablement.) When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

(2) gerR_IN32_DisableMACIPAccess

Function	MAC IP access disablement			
Call Format	ERRCODE gerR_IN32_DisableMACIPAccess (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function disables the MAC IP access.			

(3) gerR_IN32_WritePHY

Function	PHY internal register write			
Call Format	ERRCODE gerR_IN32_WritePHY (ULONG ulPort, ULONG ulAddr, ULONG ulData)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulPort	Port subject to register writing R_IN32_PORT1(0): PORT1 R_IN32_PORT2(1): PORT2	Input
	ULONG	ulAddr	PHY register address	Input
	ULONG	ulData	Data to be written to PHY	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end (MDIO command end wait error)			
Description	This function writes to the PHY internal register in MDIO.			

Caution: Use this function during the period from gerR_IN32_EnableMACIPAccess (Section 5.3.9(1) to gerR_IN32_DisableMACIPAccess (Section 5.3.9(2)). When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

(4) gerR_IN32_ReadPHY

Function	PHY internal register read			
Call Format	ERRCODE gerR_IN32_ReadPHY (ULONG ulPort, ULONG ulAddr, ULONG *ulData)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulPort	Port subject to register reading R_IN32_PORT1(0): PORT1 R_IN32_PORT2(1): PORT2	Input
	ULONG	ulAddr	PHY register address	Input
	ULONG	*ulData	Data read from PHY	Output
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end (MDIO command end wait error)			
Description	This function reads the PHY internal register in MDIO.			

Caution: Use this function during the period from gerR_IN32_EnableMACIPAccess (Section 5.3.9(1)) to gerR_IN32_DisableMACIPAccess (Section 5.3.9(2)). When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

(5) gerR_IN32_CheckPHY

Function	PHY check processing			
Call Format	R_IN32_OK: Normal end			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the PHY link status.			

5.3.10 Transient Reception

(1) gerR_IN32_MainReceiveTransient1

Function	Transient reception main processing 1			
Call Format	ERRCODE gerR_IN32_MainReceiveTransient1 (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the transient frames received by R-IN32M4-CL3.			

(2) gerR_IN32_MainReceiveTransient2

Function	Transient reception main processing 2			
Call Format	ERRCODE gerR_IN32_MainReceiveTransient2 (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function delivers the received transient frames acquired by gerR_IN32_MainReceiveTransient1 (Section 5.3.10(1)) to the user program using gerR_IN32_CallbackReceivedTransient (Section 5.5(3)).			

(3) gerR_IN32_EnableReceiveTransient

Function	Transient reception enable/disable setting for user reasons			
Call Format	ULONG gerR_IN32_EnableReceiveTransient (BOOL blEnable)			
Arguments	Name	Variable Name	Description	I/O
	BOOL	blEnable	Reception enable/disable setting R_IN32_TRUE: Enable reception R_IN32_FALSE: Disable reception	Input
Return Value	R_IN32_OK: Normal end			
Description	This function enables or disables transient reception for user reasons. When the return value of gerR_IN32_CallbackReceivedTransient created by the user is R_IN32_ERR, "Transient reception enable/disable setting for user reasons" is set to "Disable reception". Be sure to set reception to "Enable reception" using this function once reception becomes possible.			

(4) gblR_IN32_GetReceiveTransientStatus

Function	Status acquisition of transient reception enable/disable setting for user reasons			
Call Format	BOOL gblR_IN32_GetReceiveTransientStatus (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	Status of reception enable/disable setting R_IN32_TRUE: Reception enabled R_IN32_FALSE: Reception disabled			
Description	This function acquires the status of transient reception enable/disable setting for user reasons.			

(5) gerR_IN32_SetMACAddressTableData

Function	Node information distribution data (MAC address table) setting			
Call Format	ERRCODE gerR_IN32_SetMACAddressTableData (UCHAR uchSeqNumber, R_IN32_MACADDRESSDATA_T *pstMacAddrDat)			
Arguments	Name	Variable Name	Description	I/O
	UCHAR	uchSeqNumber	Sequential distribution number (value range: 1-7)	Input
	R_IN32_MACADDRESSDATA_T	*pstMacAddrDat	Information such as MAC address (MAC address table)	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR_OUTOFRANGE: Station number out of range or sequential distribution number out of range			
Description	This function sets the information (MAC address table), such as the MAC address, acquired by node information distribution from the master station, and the sequential distribution number.			

Caution: Register the station number of the master station as 7Dh.
If R_IN32_FALSE is set by the initial value of Node information distribution request (No. 7 of Table 5.7 "R_IN32_UNITINIT_T List"), this function does not need to be called.

The following describes the configuration of R_IN32_MACADDRESSDATA_T.

Table 5.12 R_IN32_MACADDRESSDATA_T List

No.	Member		Overview	Remarks
1	USHORT	usNodeNumber	Station number (1-120, Master station: 7Dh)	—
2	UCHAR	uchTransientReceiveEnable	Transient reception function (R_IN32_ENABLE/R_IN32_DISABLE)	—
3	UCHAR	auchMacAddress[6]	MAC address	—

5.3.11 Transient Request Reception

(1) gerR_IN32_ReceivedMACAddressData

Function	Node information distribution frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedMACAddressData (const VOID* pvReceivedData, ULONG ulDataSize)			
Arguments	Name	Variable Name	Description	I/O
	const VOID*	pvReceivedData	Received data storage area	Input
	ULONG	ulDataSize	Received data size	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create the MAC address data of the other station using gerR_IN32_SetMACAddressTableData (Section 5.3.10(5)).			

(2) gerR_IN32_ReceivedStatisticInfoRequest

Function	Statistical information acquisition request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedStatisticInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create Statistical information acquisition response frame using the received data specified in the argument.			

(3) gerR_IN32_ReceivedUnitInfoRequest

Function	Detailed node information acquisition request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedUnitInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create Detailed node information acquisition response frame using the received data specified in the argument.			

(4) gerR_IN32_ReceivedOptionInfoRequest

Function	Option information acquisition request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedOptionInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA, const USHORT usSupportFunction)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
	const USHORT	usSupportFunction	Option function support	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create Option information acquisition response frame using the received data specified in the argument.			

(5) gerR_IN32_ReceivedSelectInfoRequest

Function	Selected station information acquisition request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedSelectInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA, const VOID* pvR_IN32TLedInfo)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Address of send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
	const VOID*	pvR_IN32TLedInfo	Own station LED information	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create Selected station information acquisition response frame using the received data specified in the argument.			

(6) gerR_IN32_ReceivedContactTestRequest

Function	Communication test request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedContactTestRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create Communication test response frame using the received data specified in the argument.			

(7) gerR_IN32_ReceivedCableTestRequest

Function	Cable test request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceivedCableTestRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Create Cable test response frame using the received data specified in the argument.			

(8) gerR_IN32_ReceiveRemoteResetRequest

Function	SLMP remote reset request frame receive processing			
Call Format	ERRCODE gerR_IN32_ReceiveRemoteResetRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Check if the station number and network number of the received SLMP remote reset response frame is within the set range. A response frame is not sent when the SLMP remote reset request is normally received.			

(9) gR_IN32_SetSlmpError_Response

Function	SLMP error response frame creation processing			
Call Format	VOID gR_IN32_SetSlmpError_Response (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA, USHORT usFinCode)			
Arguments	Name	Variable Name	Description	I/O
	VOID*	pvSendFrame	Send frame	Output
	ULONG*	pulDataSize	Send data size	Output
	const VOID*	pvReceivedData	Received data storage area	Input
	const UCHAR*	puchSA	Send source node MAC address	Input
	USHORT	usFinCode	End code	
Return Value	None			
Description	Create SLMP error response frame. Store the argument (usFinCode) in the end code of the created response frame.			

(10) gerR_IN32_ErrCheckReqFieldNetworkReceived

Function	CC-Link IE Field specific request receive frame error check processing			
Call Format	ERRCODE gerR_IN32_ErrCheckReqFieldNetworkReceived (const VOID* pvReceivedData)			
Arguments	Name	Variable Name	Description	I/O
	const VOID*	pvReceivedData	Received data storage area	Input
Return Value	R_IN32_OK: Normal R_IN32_ERR: Destination station number error or destination network number error in received data			
Description	Check if the destination station number and network number of the received CC-Link IE Field specific request frame is within the set range.			

(11) gesR_IN32_ErrCheckReqSlmpReceived

Function	SLMP request receive frame error check processing			
Call Format	USHORT gusR_IN32_ErrCheckReqSlmpReceived (const VOID* pvReceivedData)			
Arguments	Name	Variable Name	Description	I/O
	const VOID*	pvReceivedData	Received data storage area	Input
Return Value	R_IN32_SLMP_FINCODE_OK: Normal (0000h) Other than above: Destination station number error or destination network number error in received data			
Description	Check if the destination station number and network number of the received SLMP request frame is within the set range.			

5.3.12 Transient Send Frame Header Creation

(1) gerR_IN32_SetEtherCcieHeader

Function	MAC+CCIE header creation processing			
Call Format	ERRCODE gerR_IN32_SetEtherCcieHeader (const UCHAR* puchSndMac, const UCHAR* puchMyMac, UCHAR uchFrameClassification, UCHAR uchDataClassification, R_IN32_NONCICLIC_FRAME_T* pstCOMMON)			
Arguments	Name	Variable Name	Description	I/O
	CONST UCHAR*	puchSndMac	Destination MAC address	Input
	const UCHAR*	puchMyMac	Own MAC address	Input
	UCHAR	uchFrameClassification	Frame type	Input
	UCHAR	uchDataClassification	Data type	Input
	R_IN32_NONCICLIC_FRAME_T *	pstCOMMON	Address of created MAC + CCIE header	Output
Return Value	Return value of erR_IN32T_TxFrame_CreateEtherCcieHeader (R_IN32_OK: Normal end only)			
Description	Create MAC header and CCIE header for pstCOMMON by calling erR_IN32T_TxFrame_CreateEtherCcieHeader.			

(2) gerR_IN32_SetTransient1Header

Function	Transient1 header creation processing			
Call Format	ERRCODE gerR_IN32_SetTransient1Header (USHORT usDataSubClassification, USHORT usTransientDataSize, R_IN32_TRAN1_HEAD_T* pstHEAD)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usDataSubClassification	Data sub type	Input
	USHORT	usTransientDataSize	Size of transient data	Input
	R_IN32_TRAN1_HEAD_T*	pstHEAD	Address of created Transient1 header	Output
Return Value	Return value of erR_IN32T_TxFrame_CreateTransient1Header (R_IN32_OK: Normal end, R_IN32_Error: Error in argument)			
Description	Create Transient1 header for pstHEAD by calling erR_IN32T_TxFrame_CreateTransient1Header.			

(3) gerR_IN32_SetRequestSlmpHeader

Function	SLMP request header creation processing			
Call Format	ERRCODE gerR_IN32_SetRequestSlmpHeader (R_IN32_SLMP_REQUSET_SETTING_T* pstSlmpReqSetting, R_IN32_SLMP_REQUEST_FRAME_T* pstSlmpExHead, ULONG* pulAllDataSize, USHORT* pusReqSerialNo)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_SLMP_REQUSET_SETTING_T*	pstSlmpReqSetting	Target station setting	Input
	R_IN32_SLMP_REQ_FRAME_T*	pstSlmpExHead	Address of created Transient2 + SLMP header to be created	Output
	ULONG*	pulAllDataSize	Entire SLMP data size	Output
	USHORT*	pusReqSerialNo	Serial number	Output
Return Value	Return value of erR_IN32T_TxFrame_CreateRequestSlmpHeader (R_IN32_OK: Normal end, R_IN32_ERR: Error in argument)			
Description	Create Transient2 header+ SLMP header (for request) for pstSlmpExHead by calling erR_IN32T_TxFrame_CreateRequestSlmpHeader.			

(4) gerR_IN32_SetResponseSlmpHeader

Function	SLMP response header creation processing			
Call Format	ERRCODE gerR_IN32_SetResponseSlmpHeader (const R_IN32_SLMP_REQUEST_FRAME_T* pstReqSlmpExHead, USHORT usSlmpDataSize, R_IN32_SLMP_RESPONSE_FRAME_T* pstSlmpExHead)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_SLMP_REQ_FRAME_T*	pstReqSlmpExHead	Received SLMP request frame	Input
	ULONG	usSlmpDataSize	Size of SLMP data area	Input
	R_IN32_SLMP_RES_FRAME_T*	pstSlmpExHead	Created Transient2 + SLMP header	Output
Return Value	Return value of erR_IN32T_TxFrame_CreateResponseSlmpHeader (R_IN32_OK: Normal end, R_IN32_ERR: Error in argument)			
Description	Create Transient2 header + SLMP header (for response) for pstSlmpExHead by calling erR_IN32T_TxFrame_CreateResponseSlmpHeader.			

5.3.13 Transient Send

(1) gerR_IN32_GetUnitInformation

Function	Unit information acquisition			
Call Format	ERRCODE gerR_IN32_GetUnitInformation (R_IN32_UNITINFO_T *pstUnitInfo, R_IN32_UNITNETWORKSETTING_T *pstUnitNetworkSetting)			
Arguments	Name	Variable Name	Description	I/O
	R_IN32_UNITINFO_T	*pstUnitInfo	Unit information	Output
	R_IN32_UNITNETWORKSETTING_T	*pstUnitNetworkSetting	Network operation setting	Output
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the setting information of its own station. The acquired setting information is used when creating Detailed node information acquisition response frame.			

The following describes the configuration R_IN32_UNITNETWORKSETTING_T.

Table 5.13 R_IN32_UNITNETWORKSETTING_T Table

No.	Member		Overview
1	ULONG	ulFrameSendCount	No. of sends during token hold
2	ULONG	ulFrameSendInterval	Frame send interval
3	ULONG	ulTokenSendCount	No. of token sends

(2) gusR_IN32_GetNodeID

Function	Node ID acquisition			
Call Format	USHORT gusR_IN32_GetNodeID (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	Node ID			
Description	This function acquires the node ID. The acquired node ID is used when performing transient send.			

(3) gerR_IN32_GetMulticastMACAddress

Function	Multicast MAC address acquisition			
Call Format	ERRCODE gerR_IN32_GetMulticastMACAddress (UCHAR *puchMACAddr)			
Arguments	Name	Variable Name	Description	I/O
	UCHAR	*puchMACAddr	Multicast address When 13-34-56-78-90-AB is set, the following addresses are returned: puchMACAddr[0]: 13h puchMACAddr[1]: 34h puchMACAddr[2]: 56h puchMACAddr[3]: 78h puchMACAddr[4]: 90h puchMACAddr[5]: ABh	Output
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end [The multicast MAC address cannot be acquired due to no data link (data link disconnected).]			
Description	This function acquires the multicast MAC address. The acquired multicast MAC address is used as the destination address when transient send is performed to all stations connected to the network.			

(4) gerR_IN32_GetUnicastMACAddress

Function	Unicast MAC address acquisition			
Call Format	ERRCODE gerR_IN32_GetUnicastMACAddress (USHORT usNodeNumber, UCHAR *puchMACAddr)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usNodeNumber	Station number (value range: 1-120, master station: 7Dh)	Input
	UCHAR	*puchMACAddr	Unicast address When 13-34-56-78-90-AB is set, the following addresses are returned: puchMACAddr[0]: 12h puchMACAddr[1]: 34h puchMACAddr[2]: 56h puchMACAddr[3]: 78h puchMACAddr[4]: 90h puchMACAddr[5]: ABh	Output
Return Value	R_IN32_OK: Normal end R_IN32_ERR_NOENTRY: No entry R_IN32_ERR_OUTOFRANGE: Station number out of range			
Description	This function acquires the unicast MAC address corresponding to the station number from Node information distribution frame received from the master station. Set the master station number to 7Dh.			

Caution: When there is no data link (data link disconnected), the unicast MAC address cannot be acquired (the return value becomes R_IN32_ERR_NOENTRY).

(5) gerR_IN32_GetSendTransientBuffer

Function	Transient send buffer acquisition			
Call Format	ERRCODE gerR_IN32_GetSendTransientBuffer (USHORT usSize, VOID** ppvSendBuffAddr, UCHAR *puchSendBuffNo, UCHAR *puchConnectionInfo)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usSize	Send data size excluding DCS/FCS	Input
	VOID	**ppvSendBuffAddr	Transient send buffer address	Output
	UCHAR	*puchSendBuffNo	Transient send buffer number	Output
	UCHAR	*puchConnectionInfo	Transient connection information	Output
Return Value	R_IN32_OK: Normal end (transient send buffer acquired) R_IN32_ERR: Abnormal end (transient send buffer acquisition error)			
Description	This function inquires whether or not there is space in the transient send area for send of the "send data size, " and returns the following information if there is space: • Transient send buffer address • Transient send buffer number • Transient connection information When you want to perform transient send, execute the following: • Acquire the transient send buffer number using this function. • Store the send data in the acquired transient send buffer. • Request transient send using gerR_IN32_RequestSendingTransient (Section 5.3.13(6)).			

Caution: In the following cases, transient send cannot be performed and the process ends in error (R_IN32_ERR: abnormal end):
- **When there is no data link (data link disconnected)**
- **When the send data size is greater than 1,510 bytes**

(6) gerR_IN32_RequestSendingTransient

Function	Transient send request			
Call Format	ERRCODE gerR_IN32_RequestSendingTransient (UCHAR uchSendBuffNo, USHORT usSize)			
Arguments	Name	Variable Name	Description	I/O
	UCHAR	uchSendBuffNo	Transient send buffer number	Input
	USHORT	usSize	Send data size excluding DCS/FCS	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end (transient send request error)			
Description	This function specifies send to the transient send buffer number acquired by gerR_IN32_GetSendTransientBuffer (Section 5.3.13(5)). Before executing this function, execute the following: • Acquire the transient send buffer using gerR_IN32_GetSendTransientBuffer. • Store the send data in the acquired transient send buffer. Any error that occurs after send is requested by this function is notified by the return value of gerR_IN32_MainSendTransient (Section 5.3.13(7)). Set the send data size to the same size as the value specified in gerR_IN32_GetSendTransientBuffer.			

Caution: In the following case, transient send cannot be performed and the process ends in error (R_IN32_ERR: Abnormal end):
- **When there is no data link (data link disconnected)**

(7) gerR_IN32_MainSendTransient

Function	Transient send main processing			
Call Format	ULONG gerR_IN32_MainSendTransient (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the transient send end result. This function calls gerR_IN32_CallbackTransientSendingComplete (Section 5.5(4)) to issue a notification regarding the status (send result) of the target send descriptor.			

5.3.14 Hardware Test

(1) gerR_IN32_IEEETest

Function	IEEE802.3ab compliance test			
Call Format	ERRCODE gerR_IN32_IEEETest (USHORT usMode)			
Arguments	Name	Variable Name	Description	I/O
	USHORT	usMode	IEEE 802.3ab compliance test mode R_IN32_IEEE_MODE1(1): MODE1 R_IN32_IEEE_MODE2(2): MODE2 R_IN32_IEEE_MODE3(3): MODE3 R_IN32_IEEE_MODE4(4): MODE4 R_IN32_IEEE_END(5): Test end	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end			
Description	This function sets the waveform output for test mode in PHY in accordance with the IEEE 802.3ab compliance test mode of the argument.			

Caution: When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

(2) gerR_IN32_InitializeLoopBackTest

Function	Loopback communication test initialization			
Call Format	ERRCODE gerR_IN32_InitializeLoopBackTest (VOID)			
Arguments	Name	Variable Name	Description	I/O
	—	—	—	—
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end			
Description	This function performs initialization for executing the loopback communication test.			

Caution: When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

(3) gerR_IN32_ExternalLoopBackTest

Function	Loopback communication test			
Call Format	ERRCODE gerR_IN32_ExternalLoopBackTest (ULONG ulPort)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulPort	Test target port R_IN32_PORT1(0): PORT1 R_IN32_PORT2(1): PORT2	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end			
Description	This function sends a frame from the test target port specified in the argument, and verifies the received result using the other port. When implementing this test, connect Ethernet PORT 1 and Ethernet PORT 2 using an Ethernet cable.			

Caution: When the return value of this function is a value other than R_IN32_OK, the function calls gR_IN32_CallbackFatalError (Section 5.5(1)) created by the user. Be sure to execute error processing in accordance with the error code.

5.4 R-IN32M4-CL3 Driver Callback Function List

The following describes the R-IN32M4-CL3 driver callback functions

Table 5.14 R-IN32M4-CL3 Driver Callback Function List

No.	Function Classification	Function Name	Function Type	Overview
1	Error processing	gR_IN32_CallbackFatalError	VOID	R-IN32M4-CL3 fatal error acquisition
2	Own station status acquisition	gerR_IN32_CallbackCommandFromMaster	ERRCODE	Command acquisition from master station
3	Transient send/receive	gerR_IN32_CallbackReceivedTransient	ERRCODE	Received transient frame acquisition
4		gerR_IN32_CallbackTransientSendingComplete	ERRCODE	Transient send completion status acquisition

5.5 R-IN32M4-CL3 Driver Callback Function Details

The internal processing of R-IN32M4-CL3 driver callback functions needs to be customized by the user.
The following describes the details of the R-IN32M4-CL3 driver callback functions.

(1) gR_IN32_CallbackFatalError

Function	R-IN32M4-CL3 fatal error acquisition			
Call Format	VOID gR_IN32_CallbackFatalError (ULONG ulErrorCode, ULONG ulErrorInfo)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	ulErrorCode	Fatal error code	Input
	ULONG	ulErrorInfo	Fatal error information (Address of function when error occurred.)	Input
Return Value	None			
Description	This function acquires R-IN32M4-CL3 fatal errors. The R-IN32M4-CL3 driver calls this function when an R-IN32M4-CL3 fatal error is detected. Function internal processing is freely implemented by the vendor.			

Table 5.15 List of Fatal Error Codes of gR_IN32_CallBackFatalError Function

Fatal Error Code (ulErrorCode)	Fatal Error Information (ulErrorInfo)	Fatal Error Description	Processing
D529	Driver internal call source function Address of gerR_IN32D_ClearTxRxRAM	Communication LSI error	The error is most likely a malfunction caused by noise, etc. Check the distance between lines and cables as well as device grounding, and implement noise countermeasures accordingly. Implement a hardware test. If the error occurs again, most likely the hardware is faulty.
D52A	Driver internal call source function Address of erR_IN32D_MDIO_WaitCommandComplete	Communication LSI error	
D52B	Driver internal call source function Address of erR_IN32D_ResetMAC	Communication LSI error	
D52C	Driver internal call source function Address of gerR_IN32D_StartRing	Communication LSI error	

(2) gerR_IN32_CallbackCommandFromMaster

Function	Command acquisition from master station			
Call Format	ERRCODE gerR_IN32_CallbackCommandFromMaster (ULONG pulCommand)			
Arguments	Name	Variable Name	Description	I/O
	ULONG	pulCommand	Command status from master station For details, see Table 5.16 "ulCommand List".	Input
Return Value	R_IN32_OK: Normal end			
Description	This function acquires commands by parameter frame reception from the master station. The R-IN32M4-CL3 driver calls this function when a parameter frame is received from the master station. Function internal processing is freely implemented by the user.			

The following describes the configuration of ulCommand.

Table 5.16 ulCommand List

No.	Bit	Overview	Description
1	b0	Stop cyclic transmission instruction (station number out of range)	1b: Stop instruction 0b: No stop instruction
2	b1	Stop cyclic transmission instruction (reserved station setting)	1b: Stop instruction 0b: No stop instruction
3	b2	Stop cyclic transmission instruction (master station instruction)	1b: Stop instruction 0b: No stop instruction
4	b3	Stop cyclic transmission instruction (station number duplication)	1b: Stop instruction 0b: No stop instruction
5	b15-4	Reserved	—
6	b16	Station type invalid (own station type does not match station type specified by master station)	1b: Station type invalid 0b: Station type valid
7	b17	Specified size invalid[Note]	1b: Specified size invalid 0b: Specified size valid
8	b31-18	Reserved	—

Note. The cyclic transmission size specified by the master station is greater than the allowable maximum size (size specified by gerR_IN32_Initialize (Section 5.3.1(2)) for own station cyclic transmission.)

(3) gerR_IN32_CallbackReceivedTransient

Function	Received transient frame acquisition			
Call Format	ERRCODE gerR_IN32_CallbackReceivedTransient (VOID *pvRcv, USHORT usFrameSize)			
Arguments	Name	Variable Name	Description	I/O
	VOID	*pvRcv	Reception buffer	Input
	USHORT	usFrameSize	Frame size excluding FCS	Input
Return Value	R_IN32_OK: Normal end R_IN32_ERR: Abnormal end			
Description	This function acquires received transient frames. The R-IN32M4-CL3 driver calls this function when a transient frame is received. Function internal processing is freely implemented by the user.			

Caution: Set the head address of the reception buffer in increments of 4 bytes (0 or multiple of 4). When the return value is a value other than R_IN32_OK, "Transient reception enable/disable setting for user reasons" is set to "Disable reception" Be sure to set transient reception for users reasons to "Enable reception" by calling gerR_IN32_EnableReceiveTransient (Section 5.3.10(3)) once reception becomes possible.

(4) gerR_IN32_CallbackTransientSendingComplete

Function	Transient send completion status acquisition			
Call Format	ERRCODE gerR_IN32_CallbackTransientSendingComplete (UCHAR uchSendBuffNo, ERRCODE erSendStatus)			
Arguments	Name	Variable Name	Description	I/O
	UCHAR	uchSendBuffNo	Transient send buffer number	Input
	ERRCODE	erSendStatus	Status of target transient send buffer (send result) R_IN32_OK: Transient send normal completion R_IN32_ERR: Transient send abnormal completion	Input
Return Value	R_IN32_OK: Normal end			
Description	This function acquires the send status (send result) of the transient send buffer. The R-IN32M4-CL3 driver calls this function when send of a transient frame ends. Function internal processing is freely implemented by the vendor.			

6. LINK DEVICE SYSTEM AREA

A part of link devices in a slave station connected to CC-Link IE Field Network can be defined as a system area. A system area is used to notify other stations of the status of the own station and to instruct operation from the master station to the own station.

Defining a part of link devices as a system area is optional. To define a system area, assign the bits of link devices as indicated in Table 6.1 "System Area Bit Assignments (Example)".

The following table shows an example of defining a system area for the remote input (RX) and the remote output (RY). When defining a system area for the remote registers (RWr, RWw), replace RX with RWw and RY with RWw.

Table 6.1 System Area Bit Assignments (Example)

Bit	Name	Bit	Name
RX(S+0)	Reserved	RY(S+0)	Reserved
RX(S+1)		RY(S+1)	
RX(S+2)		RY(S+2)	
RX(S+3)		RY(S+3)	
RX(S+4)		RY(S+4)	
RX(S+5)		RY(S+5)	
RX(S+6)		RY(S+6)	
RX(S+7)	Warning status flag	RY(S+7)	
RX(S+8)	Initial data processing request flag	RY(S+8)	Initial data processing complete flag
RX(S+9)	Initial data setting complete flag	RY(S+9)	Initial data setting request flag
RX(S+A)	Error status flag	RY(S+A)	Error reset request flag
RX(S+B)	Remote ready	RY(S+B)	Reserved
RX(S+C)	Reserved	RY(S+C)	
RX(S+D)		RY(S+D)	
RX(S+E)		RY(S+E)	
RX(S+F)		RY(S+F)	

Remark: S: Start number of system area

6.1 System Area Details

This section describes the details of each bit in the system area, using the remote input (RX) and the remote output (RY) as an example.

(1) Remote Ready: RX(S+B)

This bit indicates that data can be sent and received between the master station and the own station.

Turn on the bit after the R-IN32M4-CL3 application product is powered on or reset.

Turn off the bit when data cannot be sent or received between the master station and the own station due to Error status flag.

Note that leave the bit on when Warning status flag is on.

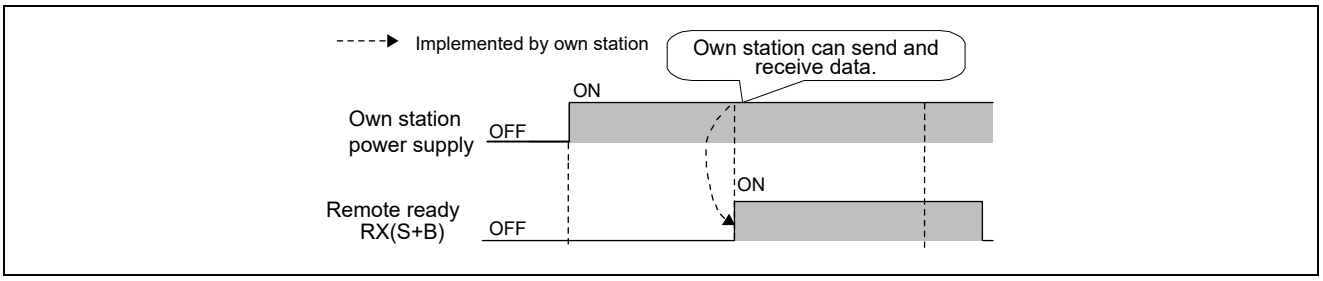


Figure 6.1 Timing Chart: Remote Ready

(2) Initial data processing request flag: RX(S+8), Initial data processing complete flag: RY(S+8)

These bits are used to request initial data processing from the own station to the master station after the R-IN32M4-CL3 application product is powered on or reset. After the initial data processing completes, turn on Remote ready.

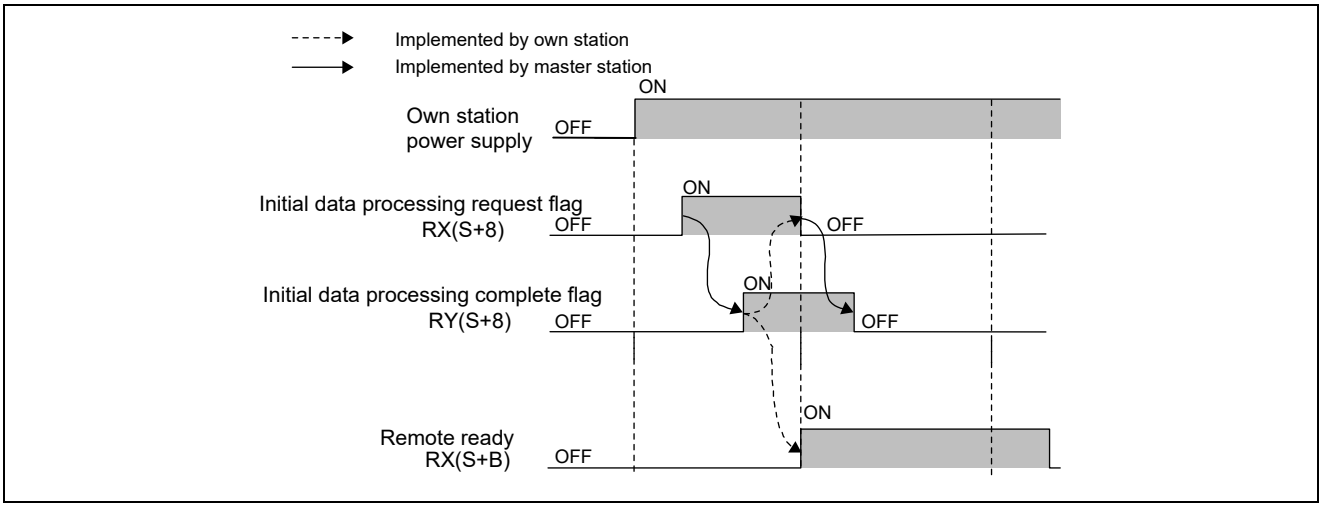


Figure 6.2 Timing Chart: Initial Data Processing Request/Complete Flag

(3) Initial Data Setting Complete Flag: RX(S+9), Initial Data Setting Request Flag: RY(S+9)

These bits are used to request initial data setting from the master station to the own station.
After the initial data are set, turn on Remote ready.

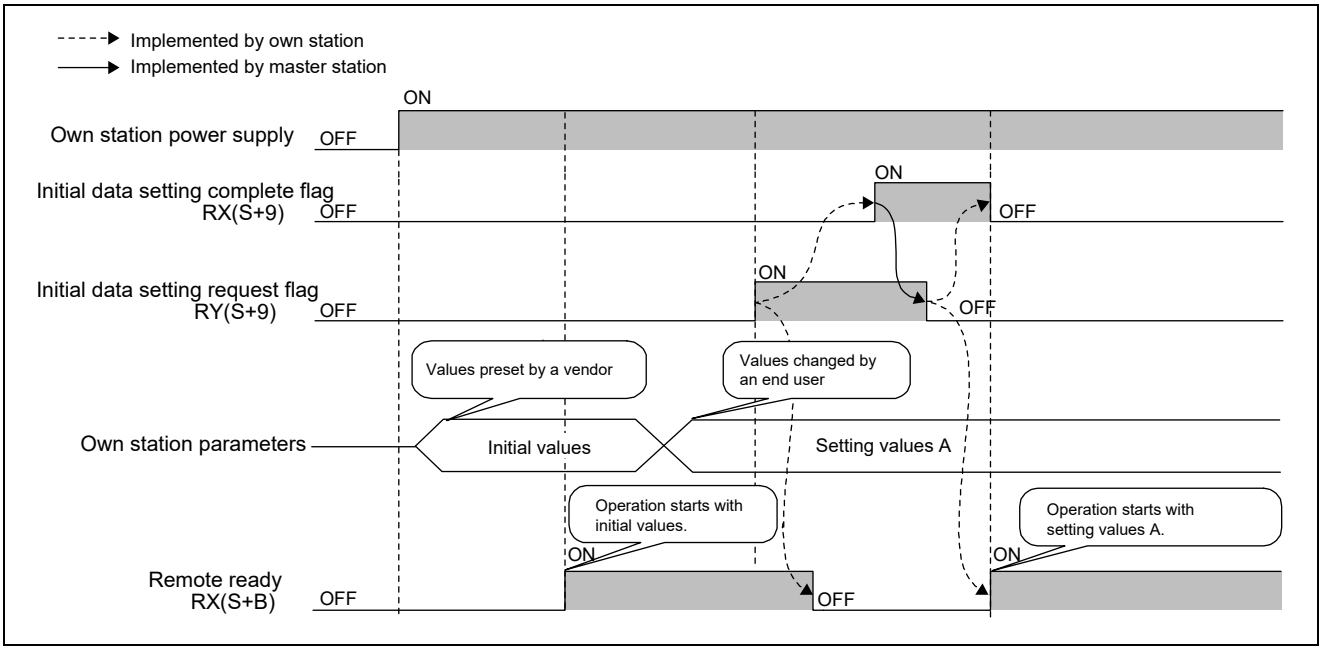


Figure 6.3 Timing Chart: Initial Data Setting Complete/Request Flag

(4) Implementation of Initial Data Processing Request/Complete Fag and Initial Data Setting Complete/Request Flag

When these flags are implemented, turn on Remote ready after both the initial data processing and the initial data setting processing complete.

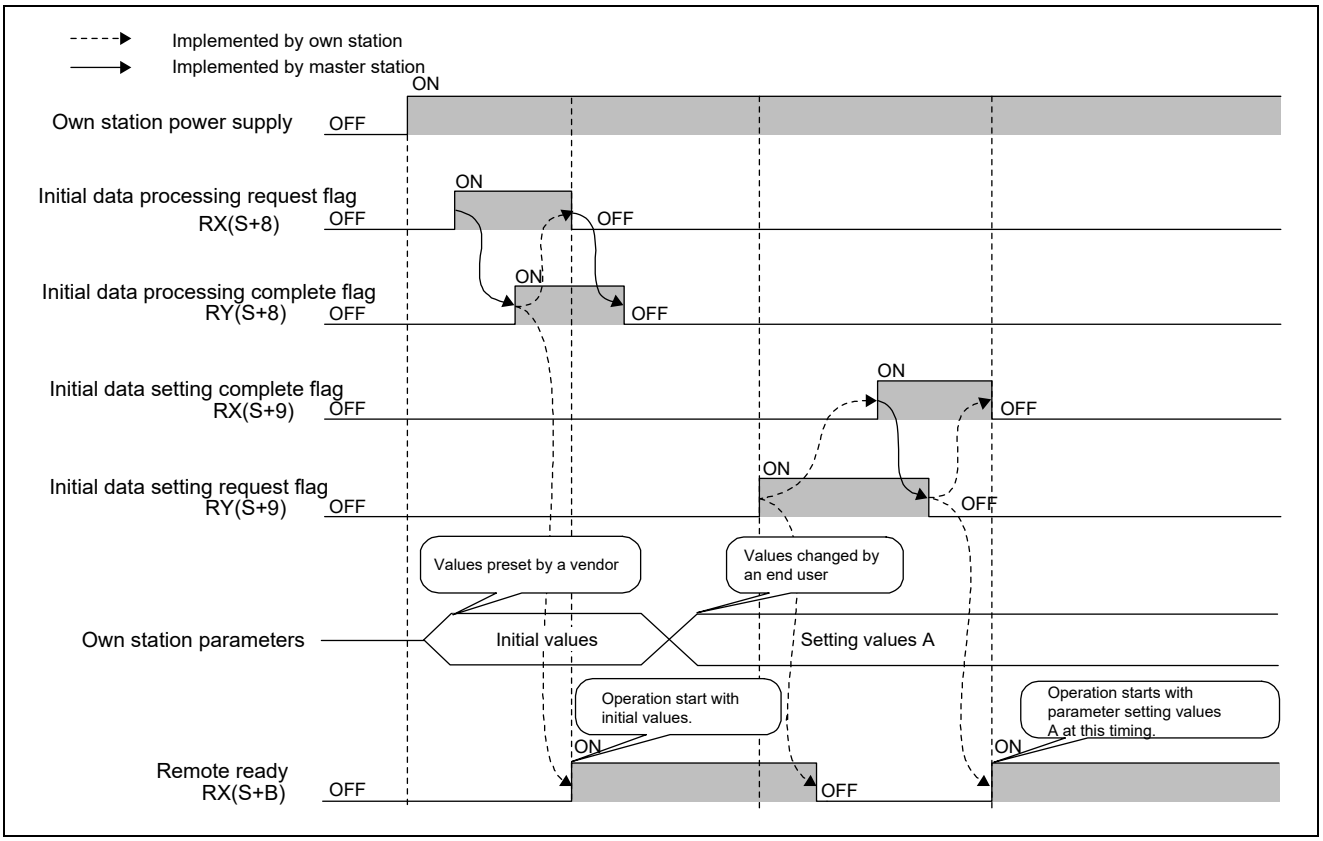


Figure 6.4 Timing Chart: Initial Data Processing and Setting

(5) Error Status Flag: RX(S+A), Error Reset Request Flag: RY(S+A)

These bits are used to notify or clear a moderate/major error of the own station. (The station can no longer continue its operation.)

Turn on Error status flag when a moderator/major error occurs in the own station. The master station clears the error and turns on Error reset request flag.

The own station turns off Error status flag and clears the error code storage area. Turn off Remote ready from error occurrence to error clear.

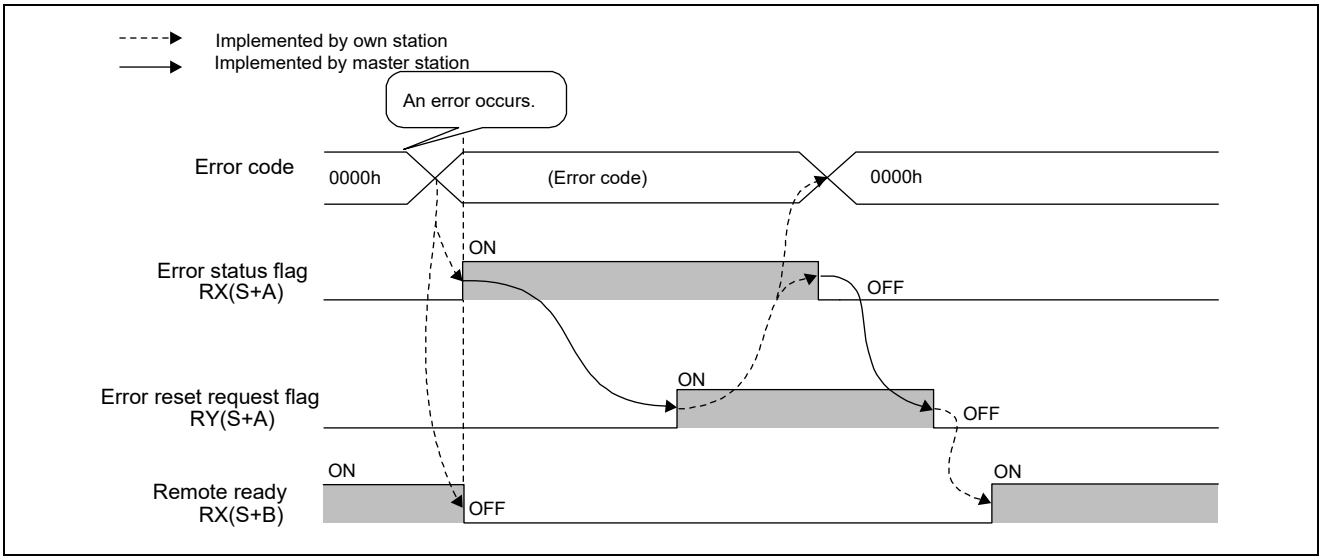


Figure 6.5 Timing Chart: Error Status Flag, Error Reset Request Flag

(6) Warning Status Flag: RX(S+7)

This bit is used to notify a minor error of the own station. (The station can continue its operation.) Turn on this flag when a minor error occurs in the own station.

When the master station eliminates the error cause, the own station clears the warning code and turn off this flag.

Leave Remote ready on from warning occurrence to warning clear.

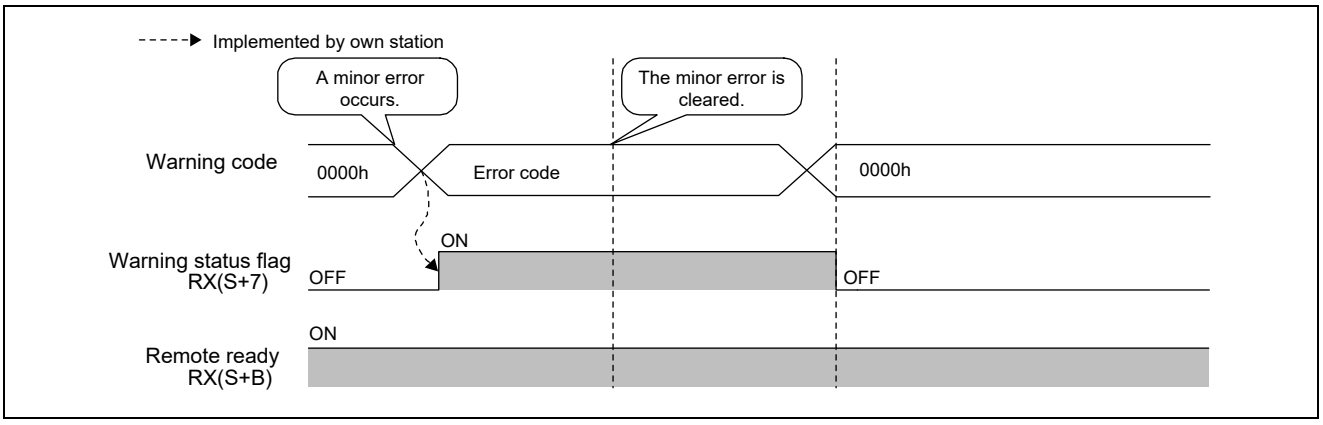


Figure 6.6 Timing Chart: Warning Status Flag

7. FRAME FORMAT

The following table lists frames that are created during sending or analyzed during receiving by the user program. This section describes the formats of frames in the following table for your reference to create and analyze frames.

Table 7.1 List of Frames Created/Analyzed by User Program

No.	Frame Name	Frame Type (FType)		Data Type (DataType)		Data Sub-Type	
1	SLMP	22h	Transient1	05h	Network common	0002h	SLMP
2	CC-Link compatible transient	25h	Transient2	04h	CC-Link compatible	—	—
3	TransientAck	23h	TransientAck	Note	Note	Note	Note

Note. TransientAck sends an acknowledgement response using the data type and data sub-type of the received frame.

7.1 Common Format

The frames of the CC-Link IE Field Network are IEEE802.3 Ethernet frame compatible. The following shows the format common to the frames of the CC-Link IE Field Network.

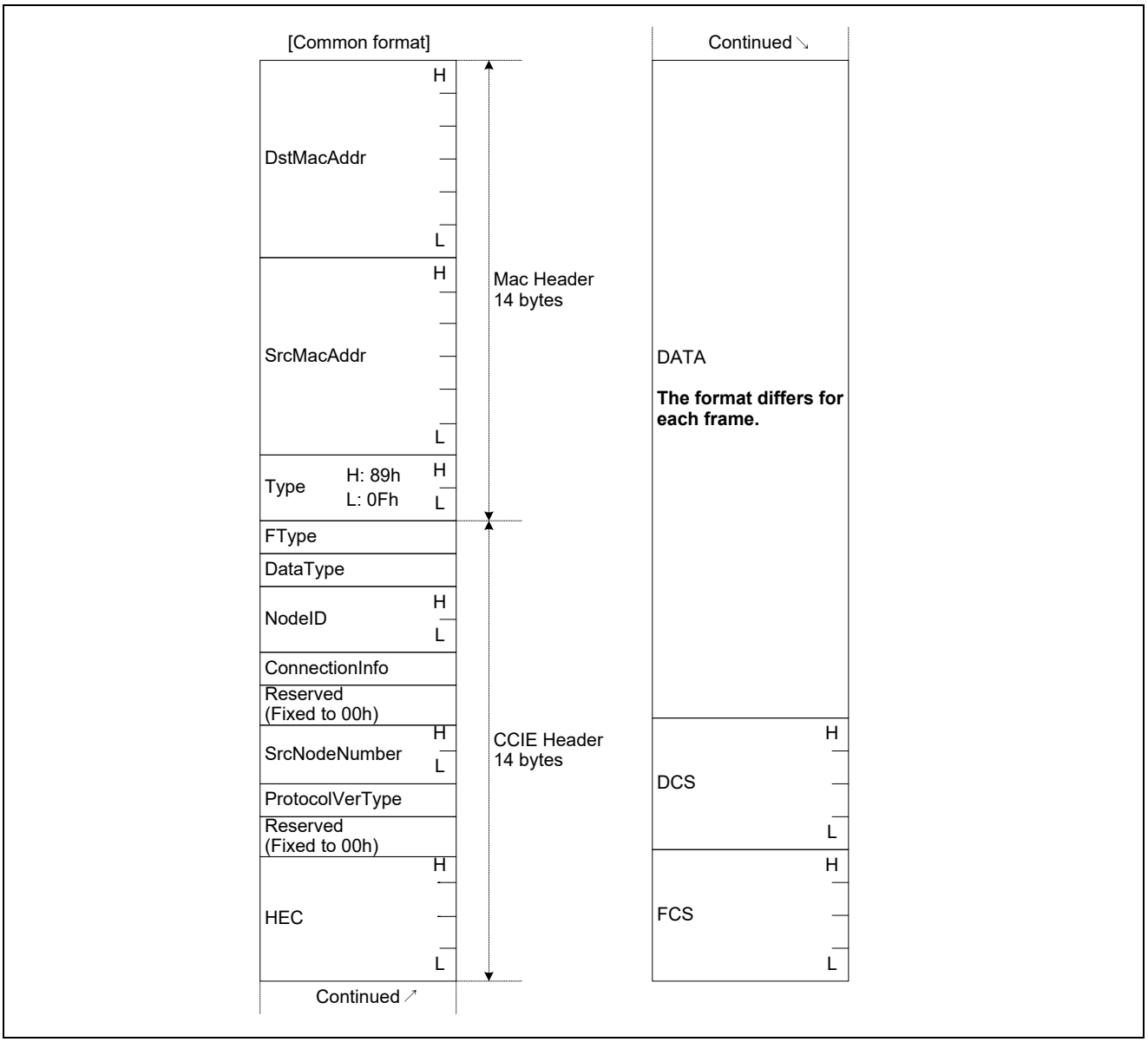


Figure 7.1 Overview of Frame Common Format

The following table describes the details of items defined in the frame common format. All items are set when a frame is sent by the R-IN32M4-CL3 driver.

Table 7.2 Details of Frame Common Format

Item	Description	Value	Remarks
Dst/SrcMacAddr (first octet)	MAC address of send destination/source	Value managed by IEEE[Note1]	01h when the MAC address is 01-23-45-67-89-AB
Dst/SrcMacAddr (second octet)	MAC address of send destination/source		23h when the MAC address is 01-23-45-67-89-AB
Dst/SrcMacAddr (third octet)	MAC address of send destination/source		45h when the MAC address is 01-23-45-67-89-AB
Dst/SrcMacAddr (fourth octet)	MAC address of send destination/source	Value managed by user[Note1]	67h when the MAC address is 01-23-45-67-89-AB
Dst/SrcMacAddr (fifth octet)	MAC address of send destination/source		89h when the MAC address is 01-23-45-67-89-AB
Dst/SrcMacAddr (sixth octet)	MAC address of send destination/source		ABh when the MAC address is 01-23-45-67-89-AB
Type	Type	Fixed to 890Fh[Note1]	Indicates that the frame is a CC-Link IE Field Network frame.
FType	Frame type	[Note2]	—
DateType	Data type		—
NodeID	Node identifier	00h to F0h (0 to 240)[Note1]	Management information of each slave station connected to the master station (The number differs from a station number.)
ConnectionInfo	Transient identification information	01h to FFh (1 to 255)	Information for identifying the transient frame sent during one token hold
SrcNodeNumber	Own station number	0001h to 0078h (1 to 120)[Note1]	—
ProtocolVerType	Bits 7-4 Protocol version	Fixed to 0h	—
	Bits 3-0 Protocol type	Fixed to 1h	—
HEC	Header Error Control	—	—
DCS	Data Check Sequence	—	—
FCS	Frame Check Sequence	—	—

Notes 1. Set using big endian.

2. See Table 7.1 "List of Frames Created/Analyzed by User Program".

7.2 Transient1 Frame

The following shows the basic format of a Transient1 frame.

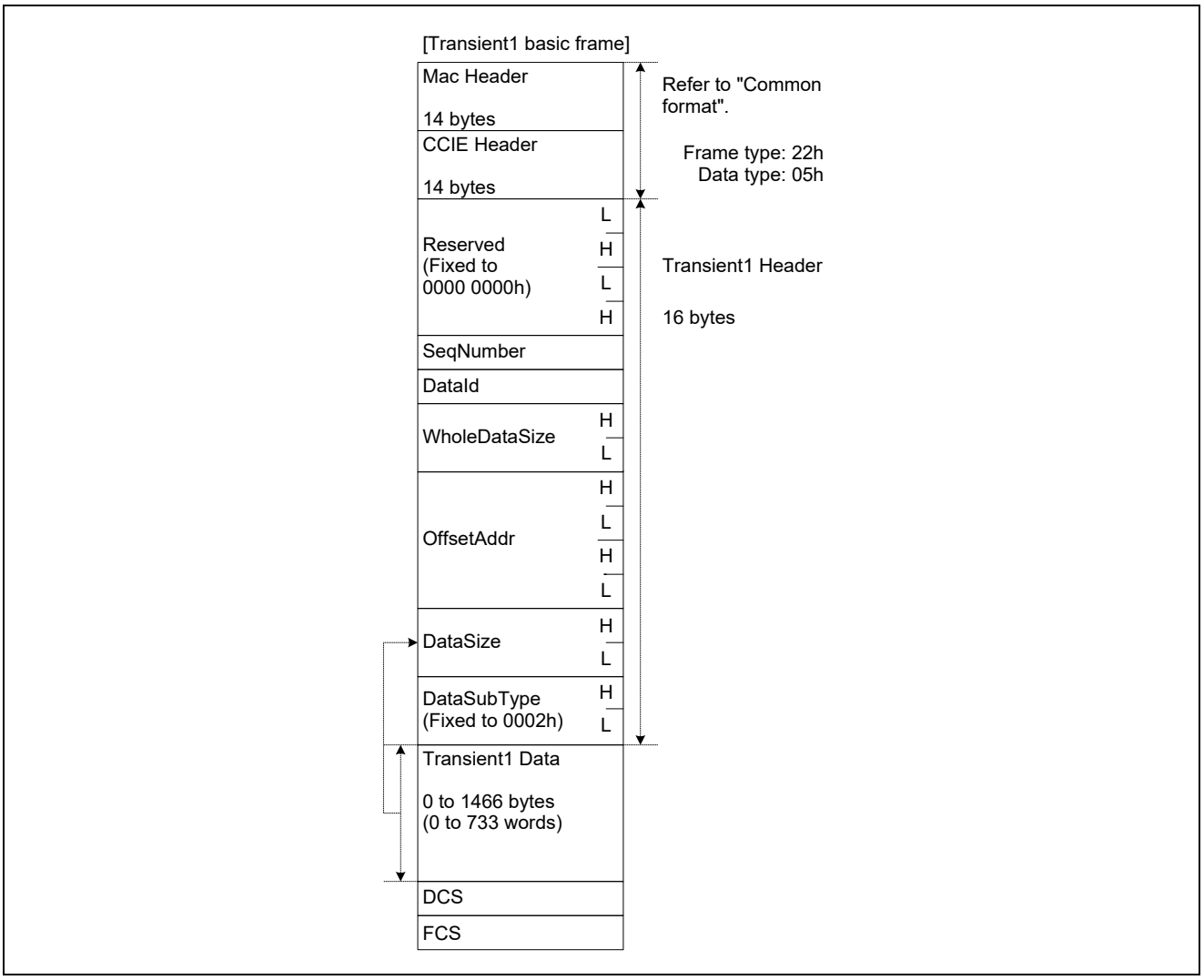


Figure 7.2 Overview of Transient1 Frame Basic Format

The following table describes the details of items defined in the Transient1 frame basic format.

Table 7.3 Details of Transient1 Frame Basic Format

Item	Description		Value	Remarks
SeqNumber	Bit 7	Final frame identification	0b: Divided frame 1b: Final divided frame	A number assigned when transient data is divided
	Bits 6-0	Transient1 frame sequential number	00h to 7Fh (0 to 127)	
DataId	Transient data identification number		00h to FFh (0 to 255)	Set the same identification number for the frames after divided.
WholeDataSize	Entire transient data size	Intelligent device station	0000h to 0800h (0 to 2048)Note	Entire transient data size (in bytes) before divided
		Remote device station	0000h to 0400h (0 to 1024)Note	
OffsetAddr	Offset address from the start of entire transient data		0000 0000h to 7FFF FFFFhNote	When not divided: Fixed to 0 Start frame when divided: Fixed to 0 For the second frame and later, the storage location within the entire transient data is indicated using an offset address from the start of the data.
DataSize	Size of transient data in the frame		0000h to 05BAh (0 to 1466)Note	Transient data size (in bytes) after divided
DataSubType	Data sub-type		0002h: SLMPNote	SLMP only in this manual.

Note. Set using big endian.

The following shows the format when a Transient1 frame is divided. In the following example, the transient data is SLMP[Note] and the transient data area addresses are set to 1000h to 1800h.

Note. For the details of SLMP, see Section 7.4 "SLMP Frame".

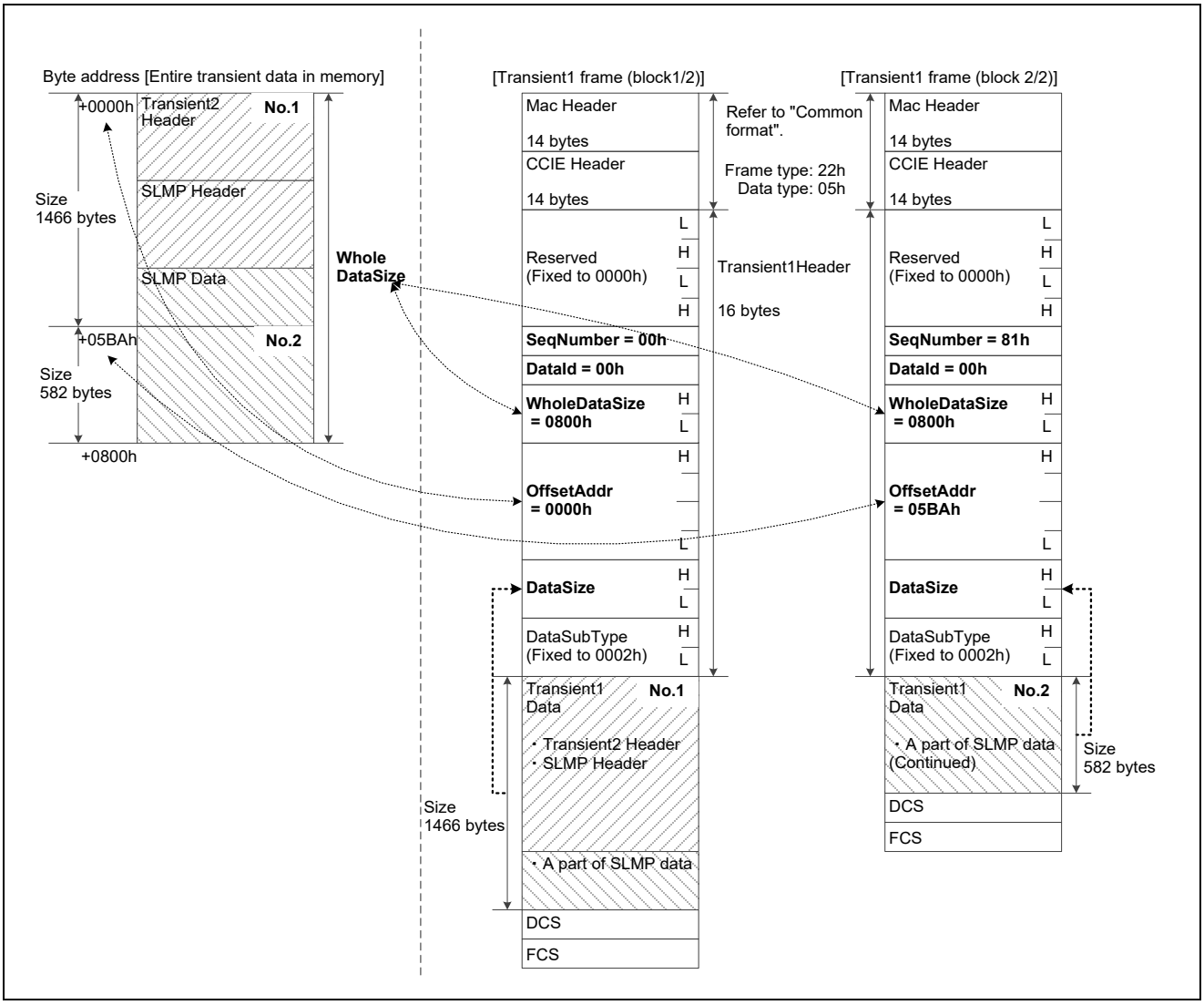


Figure 7.3 Example of Transient1 Frame Division

7.3 CC-Link Compatible Transient Frame

The following shows the basic format of a CC-Link compatible transient frame.

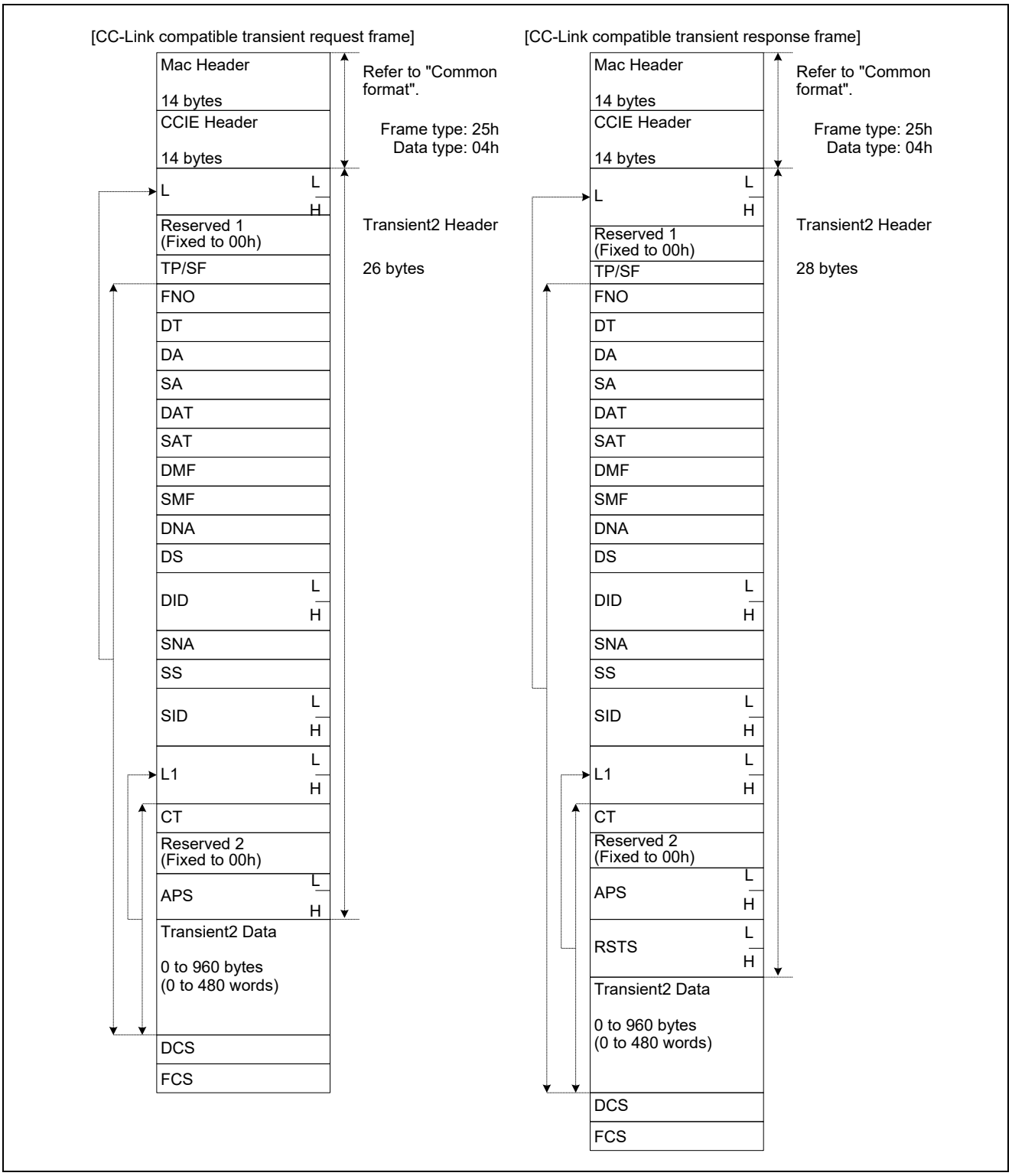


Figure 7.4 Overview of CC-Link Compatible Transient Frame Basic Format

7.3.1 Details of CC-Link Compatible Transient Frame Basic Format

The following table describes the details of items defined in the CC-Link transient frame basic format.

Table 7.4 Details of CC-Link Compatible Transient Frame Basic Format

Item	Description		Value	Remarks
L	Frame length		—	Set the data length after FNO in bytes.
TP/SF	Type/Sequence number		Fixed to 00h	—
FNO	Start frame identification number/Divided frame number		Fixed to 00h	—
DT	Priority/Response frame necessity		Fixed to 00h	—
DA	Destination station number		01h to 78h (1 to 120): Station number 7Dh: Specified control station/master station 7Eh: Current control station/master station FFh: Global request	Set the station number of the destination station. (Same value as DS.)
SA	Source station number		01h to 78h (1 to 120): Station number	Set the station number of the execution station. (Same value as SS.)
DAT	Destination application type		Fixed to 22h	—
SAT	Source application type		Fixed to 22h	—
DMF	Execution module destination flag		00h: Transient2 03h: SLMP	—
SMF	Execution module source flag		00h: Transient2 03h: SLMP	—
DNA	Destination network number		01h to EFh (1 to 239)	Set the network number of the destination station.
DS	Destination station number 02		01h to 78h (1 to 120): Station number 7Dh: Specified control station/master station 7Eh: Current control station/master station FFh: Global request	Set the station number of the destination station. (Same value as DA.)
DID	Destination identification number		Fixed to 03FFh	—
SNA	Source network number		01h to EFh (1 to 239)	Set the network number of the execution station.
SS	Source station number 02		01h to 78h (1 to 120): Station number	Set the station number of the execution station. (Same value as SA.)
SID	Source identification number		Fixed to 03FFh	—
L1	Data length		—	Set the data length after CT in bytes.
CT	Command type		04h to 1Fh: Transient2 30h: SLMP request B0h: SLMP response	For details of Transient2 command type, see Section 7.3.2 "Details of Command Type (CT)".
APS	Application number	Bits 15-8	Fixed to 00h	Set a number to identify the frame when the source station sends a request.
		Bits 7-0	00h to FFh	
RSTS	Return code		0000h: Normal Other than 0000h: Error code	During response only. For details of return code (RSTS), see Section 7.3.3 "Details of Return Code (RSTS)".

7.3.2 Details of Command Type (CT)

The following shows the data structure of the command type (CT).

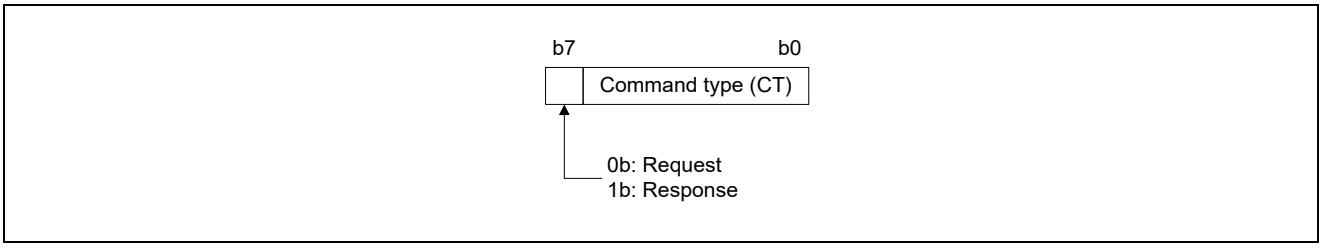


Figure 7.5 Data Structure of CC-Link Compatible Transient Command Type

The following table lists the CC-Link compatible transient commands described in this manual. For the frame format of each command, see Section 7.3.4 "Memory Access Information Acquisition" and subsequent sections.

Table 7.5 CC-Link Compatible Transient Command List

Command Type	Description
04h	Memory access information acquisition request
84h	Memory access information acquisition response
08h	Remote RUN request
88h	Remote RUN response
09h	Remote STOP request
89h	Remote STOP response
10h	Memory read request
90h	Memory read response
12h	Memory write request
92h	Memory write response

7.3.3 Details of Return Code (RSTS)

The return code (RSTS) is an area where the server stores the error code in the response frame when an error exists in the client request frame.

[When the own station is a client]

During the response frame receive processing, the error code of an error detected in the request frame sent by the own station is stored. Refer to the user's manual of the request destination device (MELSEC or partner manufacturer product) and correct the request frame creation processing or the request send processing.

[When the own station is a server]

During the response frame send processing, store the error code of an error detected in the request frame sent by the client. The error code can be defined by a user. The following table lists error code examples. (Error codes of Nos. 2 to 7 in the table are described in the sample code.)

Table 7.6 Examples of Error Codes Stored in Return Code

No.	Error Code	Description (Definition in the R_IN32M4_Transient.h file)	Action
1	0000h	Normal	—
2	D203h	Transient data read/write address specification error (USER_ERR_WRREQ_ADDRESS_OUTOFRANGE)	Correct the read/write addresses in the transient request source, and perform the processing again.
3	D213h	Transient data command error (USER_ERR_WRREQ_COMMAND_OUTOFRANGE)	Correct the request command in the transient request source, and perform the processing again.
4	D218h	Transient data read/write data size error (USER_ERR_WRREQ_SIZE_OUTOFRANGE)	Correct the read/write data size in the transient request source, and perform the processing again.
5	D219h	Transient data attribute code error (USER_ERR_WRREQ_ATTRIBUTE_OUTOFRANGE)	Correct the attribute code in the transient request source, and perform the processing again.
6	D21Ah	Transient data access code error (USER_ERR_WRREQ_ACCESSCODE_OUTOFRANGE)	Correct the access code in the transient request source, and perform the processing again.
7	D2AEh	Transient data destination station number error (USER_ERR_NOTTOONESELF)	Transient data addressed to a different network/station number has been received. Check the network number and the destination station number, and perform the processing again.
8	D2A0h	Receive buffer full error	Check the network status by executing the CC-Link IE Field Network diagnostics. When transient data reception of the destination station is overloaded, have the send source send the data after a desired period of time has elapsed.
9	D2A1h	Send buffer full error	Decrease the transient transmission frequency, and perform the processing again. Check that there is no error in the cable and switching hub connections in the request source.
10	D2A3h	Transient data frame length (L) error	Correct the corresponding error in the Transient2 header, and perform the processing again.
11	D2A4h	Transient data reserved (RSV) error	
12	D2A5h	Transient data destination station number (DA) error	
13	D2A6h	Transient data source station number (SA) error	
14	D2A7h	Transient data destination application type (DAT) error	
15	D2A8h	Transient data source application type (SAT) error	
16	D2A9h	Transient data destination network number (DNA) error	
17	D2AAh	Transient data destination station number (DS) error	
18	D2ABh	Transient data source network number (SNA) error	
19	D2ACh	Transient data source number (SS) error	
20	D2ADh	Transient data length (L1) error	

7.3.4 Memory Access Information Acquisition

Executing the memory access information acquisition command (CT=04h) allows you to acquire applicable devices of the destination controller and access codes. The following shows the format of a memory access information acquisition frame.

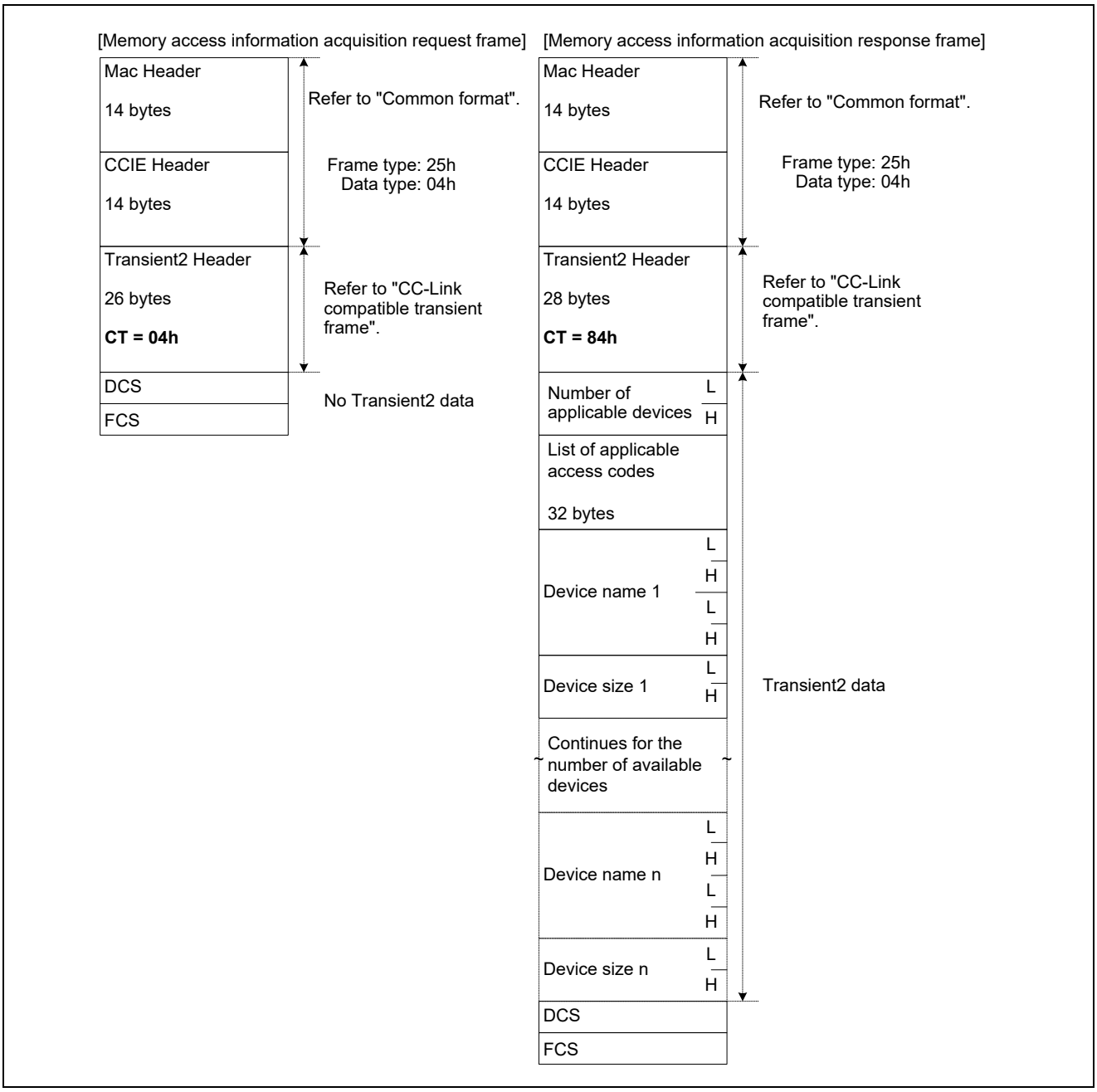


Figure 7.6 Overview of Memory Access Information Acquisition Frame Format

The following table describes the details of items defined in the memory access information acquisition frame format.

Table 7.7 Details of Memory Access Information Acquisition Frame Format

<table>
<tr><th>Item</th><th>Description</th><th>Value</th><th>Remarks</th></tr>
<tr><td>Number of applicable devices</td><td>Stores the number of applicable devices that can be used by the memory read/write commands.</td><td>—</td><td>—</td></tr>
<tr><td>List of applicable access codes</td><td>Stores whether or not a device of the destination station controller of the memory read/write commands can be accessed in the following bit pattern. The devices are indicated by 00h to FFh access codes.

<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b3</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>0Fh</td><td>0Eh</td><td>0Dh</td><td>0Ch</td><td>0Bh</td><td>0Ah</td><td>09h</td><td>08h</td><td>07h</td><td>06h</td><td>05h</td><td>04h</td><td>03h</td><td>02h</td><td>01h</td><td>00h</td></tr>
<tr><td>1Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>10h</td></tr>
<tr><td>2Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>20h</td></tr>
<tr><td>3Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>30h</td></tr>
<tr><td>4Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>40h</td></tr>
<tr><td>5Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>50h</td></tr>
<tr><td>6Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>60h</td></tr>
<tr><td>7Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>70h</td></tr>
<tr><td>8Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>80h</td></tr>
<tr><td>9Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>90h</td></tr>
<tr><td>AFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>A0h</td></tr>
<tr><td>BFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>B0h</td></tr>
<tr><td>CFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>C0h</td></tr>
<tr><td>DFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>D0h</td></tr>
<tr><td>EFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>E0h</td></tr>
<tr><td>FFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>F0h</td></tr>
</table>
0b: Inapplicable
1b: Applicable</td><td>—</td><td>—</td></tr>
<tr><td>Device name 1</td><td>Stores the device name using ASCII code, four characters maximum.</td><td>—</td><td>—</td></tr>
<tr><td>Device size 1</td><td>Stores the access size of the device of the destination controller in words.</td><td>—</td><td>—</td></tr>
<tr><td>…</td><td>Repeated for the number of applicable devices.</td><td>—</td><td>—</td></tr>
<tr><td>Device name n</td><td>Stores the device name using ASCII code, four characters maximum.</td><td>—</td><td>Note</td></tr>
<tr><td>Device size n</td><td>Stores the access size of the device of the destination controller in words.</td><td>—</td><td>Note</td></tr>
</table>

Note. The start address of the device name n and the device size n can be obtained using the following formulas.

Device name n: +3Eh + (06h × n)

Device size n: +3Eh + (06h × n) + 04h

The following is the definition of an access code.

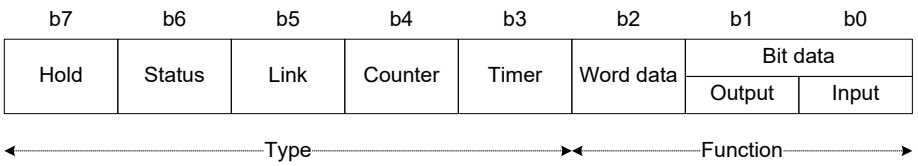


Figure 7.7 Access Code Definition

7.3.5 Remote RUN

Executing the remote RUN command (CT=08h) allows you to change the operating status of another station controller to RUN. The following shows the format of a remote RUN frame.

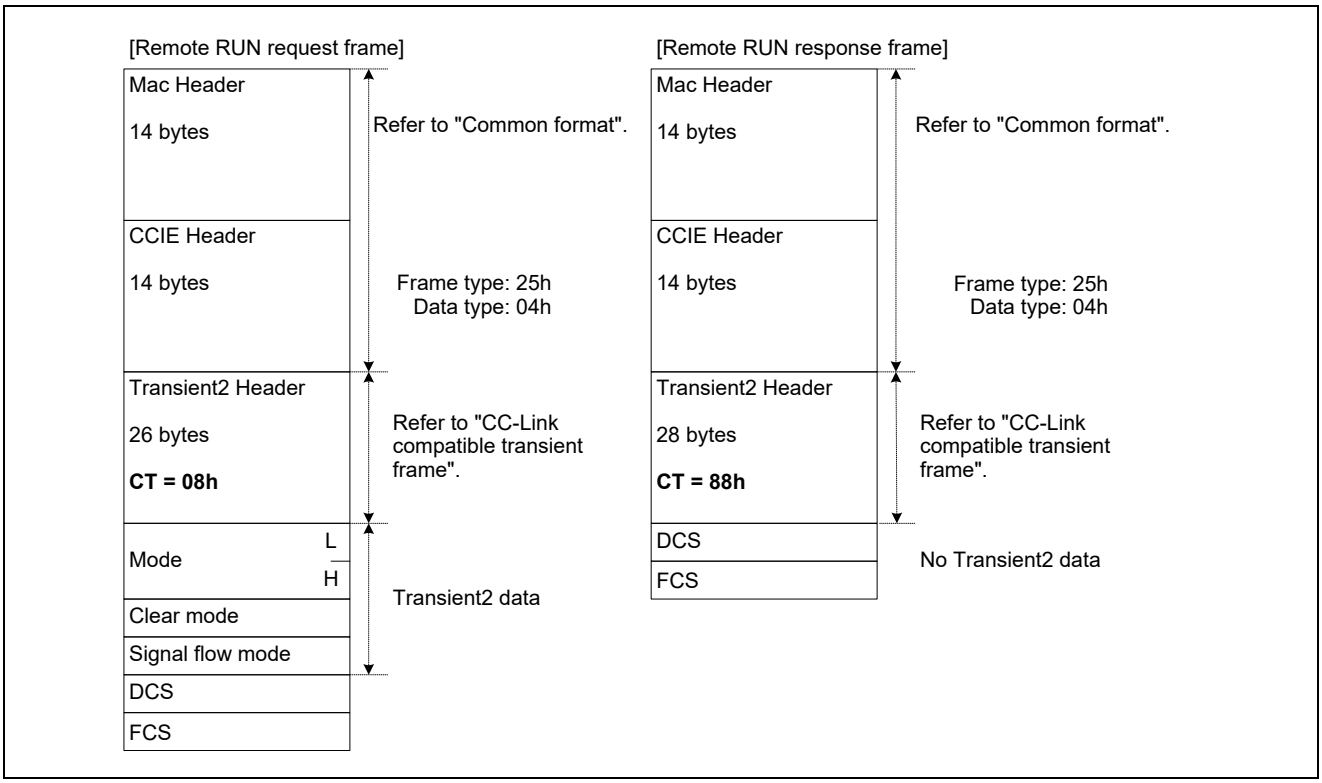


Figure 7.8 Overview of Remote RUN Frame Format

The following table describes the details of items defined in the remote RUN frame format.

Table 7.8 Details of Remote RUN Frame Format

Item	Description	Value	Remarks
Mode	Set whether or not to forcibly set another station controller that is in a STOP state to a RUN state.	1: Forced RUN 3: Normal RUN	—
Clear mode	Set whether or not to initialize the devices of the controller during remote RUN execution.	0: Do not initialize 1: Initialize devices other than latch devices 2: Initialize all devices	—
Signal flow mode	Specify the method of initialization set under "Clear mode".	0: Hold status before command execution 1: All 0 (off) 2: All 1 (on)	—

7.3.6 Remote STOP

Executing the remote STOP command (CT=09h) allows you to change the operating status of another station controller to STOP. The following shows the format of a remote STOP frame.

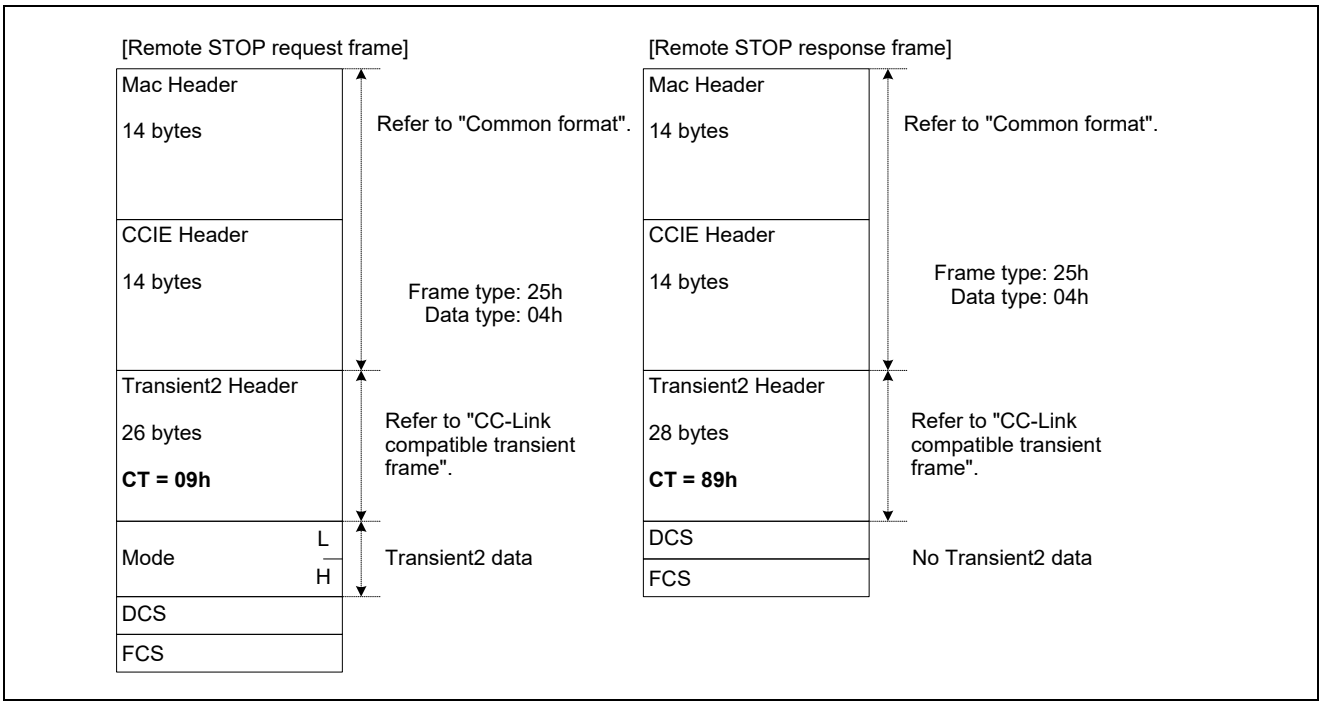


Figure 7.9 Overview of Remote STOP Frame Format

The following table describes the details of items defined in the remote STOP frame format.

Table 7.9 Details of Remote STOP Frame Format

Item	Description	Value	Remarks
Mode	Set whether or not to forcibly set another station controller that is in a RUN state to a STOP state.	1: Forced STOP 3: Normal STOP	—

7.3.7 Memory Read

Executing the memory read command (CT=10h) allows you to retrieve data from devices of another station controller. The following shows the format of a memory read frame. Note that this frame is sent when the dedicated instruction RIRD is executed in a Mitsubishi Electric programmable controller.

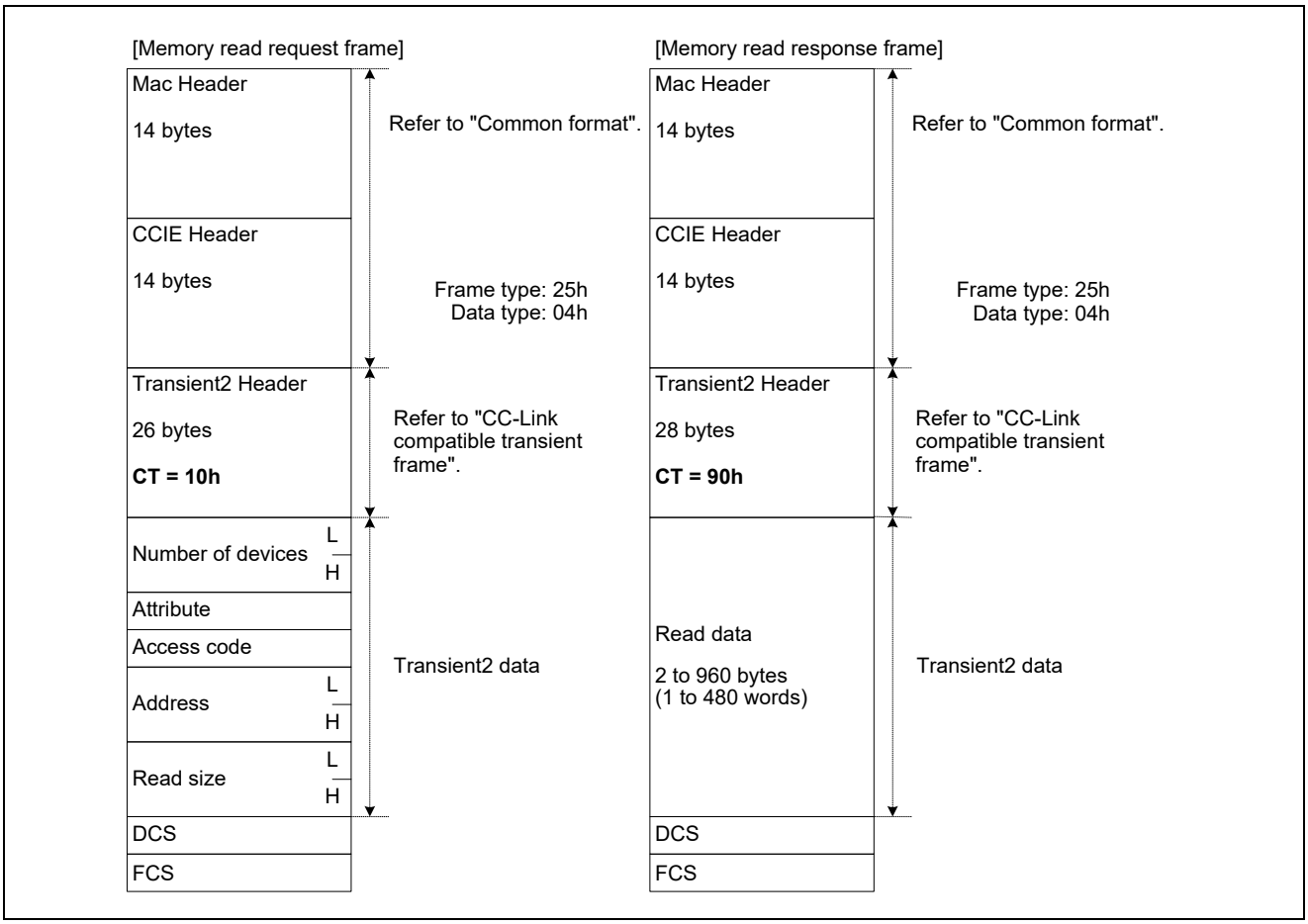


Figure 7.10Overview of Memory Read Frame Format

The following table describes the details of items defined in the memory read frame format.

Table 7.10 Details of Memory Read Frame Format

Item	Description	Value	Remarks
Number of devices	Set the number of devices to be read.	Fixed to 0001h	—
Attribute	Set the attribute of the target device of the request destination.	—	For details, see Section 7.3.9 "Access Codes and Attributes".
Access code	Set the access code of the target device of the request destination.	—	For details, see Section 7.3.9 "Access Codes and Attributes".
Address	Set the start address of the device to be read. (Set a value within the access range.)	Any offset value	Specify 0 or a multiple of 16 when the device type is bit, and 0 or a multiple of 2 when the device type is byte.
Read size	Set the size of the data to be read (in words).	1 to 480 words	—

7.3.8 Memory Write

Executing the memory write command (CT=12h) allows you to write data to devices of another station controller. The following shows the format of a memory write frame. Note that this frame is sent when the dedicated instruction RIWT is executed in a Mitsubishi Electric programmable controller.

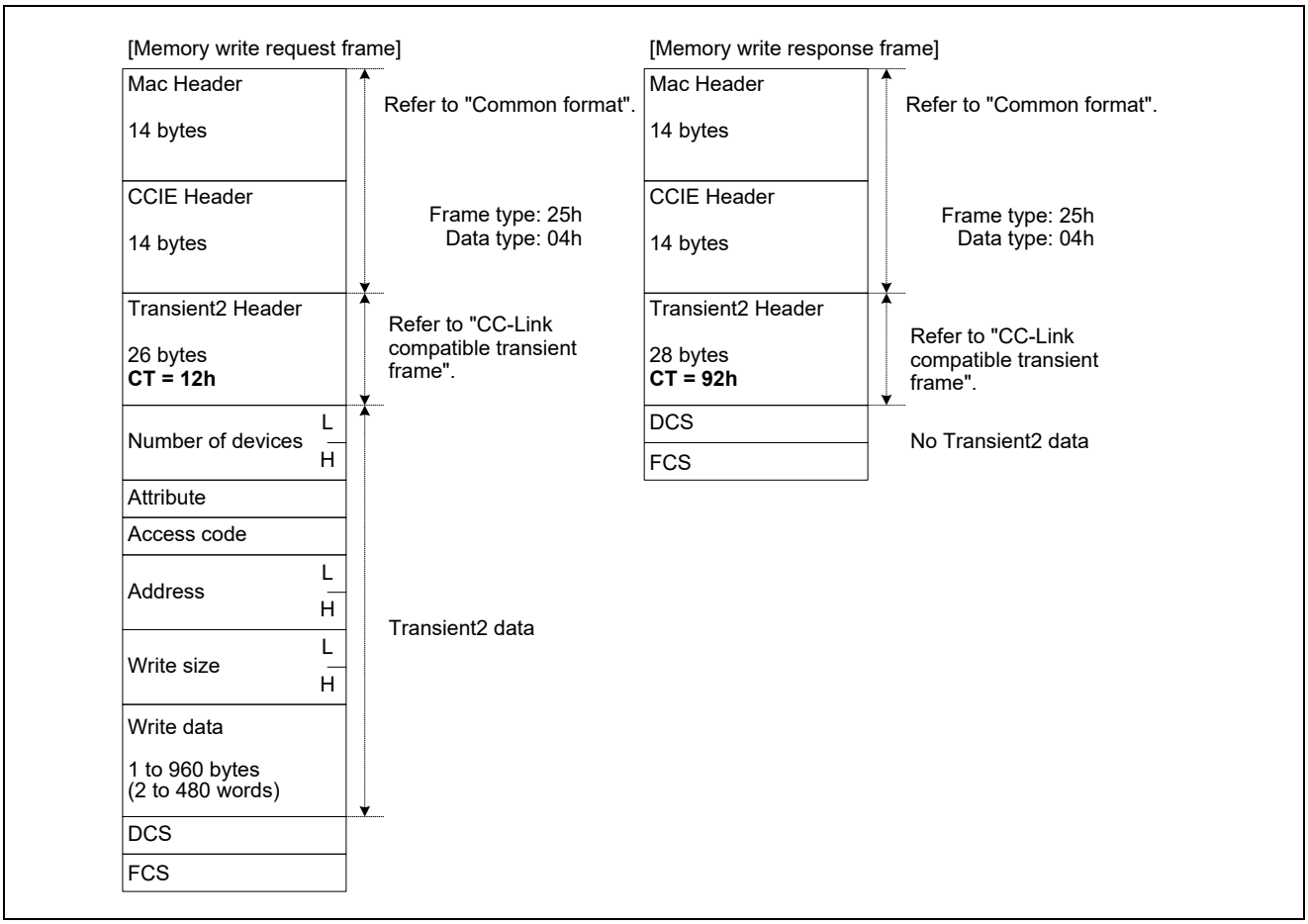


Figure 7.11Overview of Memory Write Frame Format

The following table describes the details of items defined in the memory write frame format.

Table 7.11 Details of Memory Write Frame Format

Item	Description	Value	Remarks
Number of devices	Set the number of devices to be written.	Fixed to 0001h	—
Attribute	Set the attribute of the target device of the request destination.	—	For details, see Section 7.3.9 "Access Codes and Attributes".
Access code	Set the access code of the target device of the request destination.	—	For details, see Section 7.3.9 "Access Codes and Attributes".
Address	Set the start address of the device to be written. (Set a value within the access range.)	Any offset value	Specify 0 or a multiple of 16 when the device type is bit, and 0 or a multiple of 2 when the device type is byte.
Write size	Set the size of the data to be written (in words).	1 to 480 words	—
Write data	Set the data to be written.	Any value to be written	—

7.3.9 Access Codes and Attributes

The following are the definitions of an access code and an attribute.

Define the device/buffer memory areas of the own station so that another station can access them by using the memory read/write commands.

Another station: MELSEC product and CC-Link family compatible product developed by users

Own station: R-IN32M4-CL3 application product

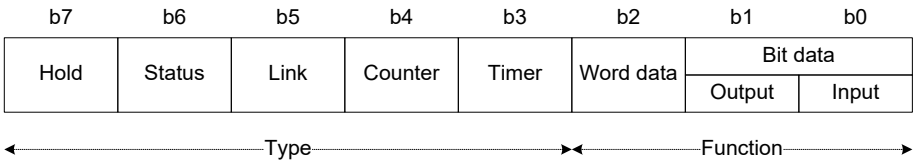


Figure 7.12Access Code Definition

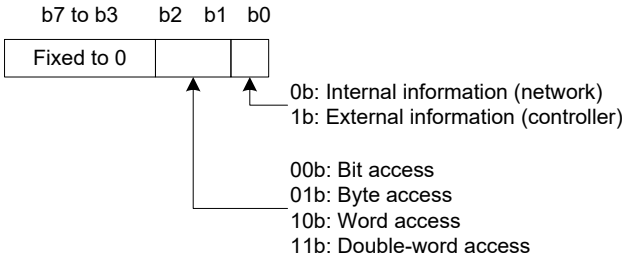


Figure 7.13Attribute Definition

The following table lists the access codes and attribute of MELSEC products.

Refer to the table when accessing another station (MELSEC product) from the own station by using the memory read/write commands.

The number of device points (size) differs depending on the programmable controller. For the accessible range, refer to the user's manual of the programmable controller used.

Table 7.12 MELSEC Product Access Code List

Device	Symbol	Device Type		Unit	Access Code[Note1]	Attribute Code[Note1]
		Bit	Word			
Input relay	X	√	—	Hexadecimal	01h	05h
Output relay	Y	√	—	Hexadecimal	02h	
Special relay	SM	√	—	Decimal	43h	
Special register	SD	—	√	Decimal	44h	
Internal relay	M	√	—	Decimal	03h	
Latch relay	L	√	—	Decimal	83h	
Timer (contact)	T	√	—	Decimal	09h	
Timer (coil)	T	√	—	Decimal	0Ah	
Timer (current value)	T	—	√	Decimal	0Ch	
Retentive timer (contact)	ST	√	—	Decimal	89h	
Retentive timer (coil)	ST	√	—	Decimal	8Ah	
Retentive timer (current value)	ST	—	√	Decimal	8Ch	
Counter (contact)	C	√	—	Decimal	11h	
Counter (coil)	C	√	—	Decimal	12h	
Counter (current value)	C	—	√	Decimal	14h	
Data register	D [Note2]	—	√	Decimal	04h	
File register	R	—	√	Decimal	84h	
Link relay	B	√	—	Hexadecimal	23h	
Link register	W [Note2]	—	√	Hexadecimal	24h	
Link special relay	SB	√	—	Hexadecimal	63h	
Link special register	SW	—	√	Hexadecimal	64h	

Notes 1. If the target station is a station other than the master/local module, refer to the user's manual of the target station for the access codes and attribute codes.

2. The extended data register (D65536 and later) and the extended link register (W10000 and later) cannot be specified.

7.4 SLMP Frame

The following figure shows the basic format of an SLMP frame.

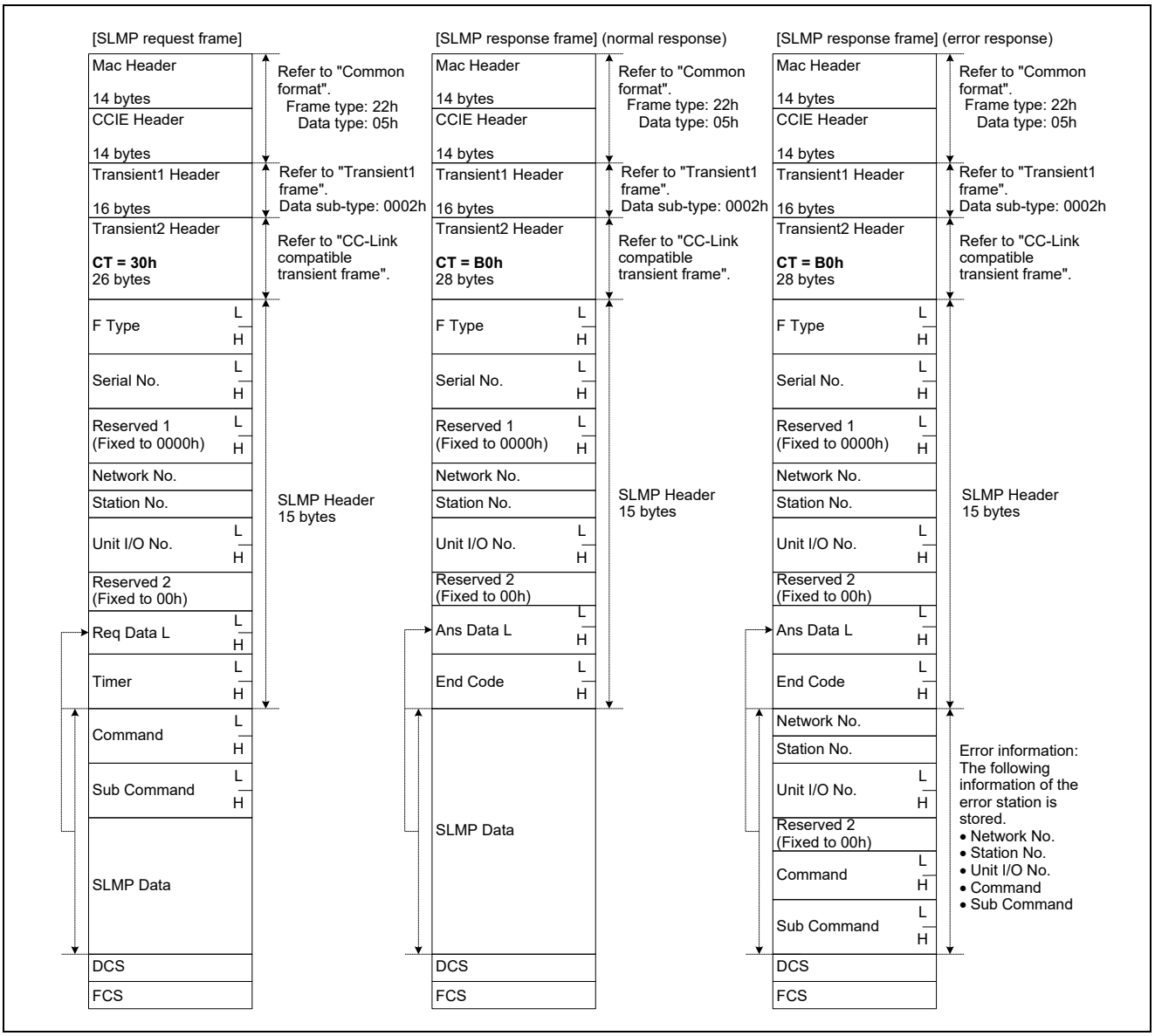


Figure 7.14Overview of SLMP Frame Basic Format

7.4.1 Details of SLMP Frame Basic Format

The following table describes the details of items defined in the SLMP frame basic format.

Table 7.13 Details of SLMP Frame Basic Format

Item	Description	Value	Remarks
F Type	Frame type	0054h: Request 00D4h: Response	—
Serial No.	Serial number	0000h to FFFFh	Set a number to identify the frame. Set the same value for a request frame and the corresponding response frame.
Network No.	Destination station network number	00h: Own station 01h to EFh (1 to 239): Other station	Set the network number of the destination station.
Station No.	Destination station number	01h to 78h (1 to 120): Station number 7Dh: Specified control station/master station 7Eh: Current control station/master station FFh: Own station[Note]	Set the destination station number.[Note]
Unit I/O No.	Destination module I/O number	03FFh: Fixed	Set the access destination CPU module.
Req Data L	Request data length	—	Set the request data size, from Timer to the end of the data area, in bytes.
Ans Data L	Response data length	—	Set the response data size, from End Code to the end of the data area, in bytes.
Timer	Monitoring timer	0001h to FFFFh 0000h: Unlimited	Request frame only. Set the wait time (in increments of 250ms) for the client to receive a response from the server. Recommended values: Own station: 0001h to 0028h (0.25 to 10s) Other stations: 0002h to 00F0h (0.5 to 60s)
End Code	End code	0000h: Normal end Other than 0000h: Error code	Response frame only. For details of end codes, see Section 7.4.2 "Details of End Code (End Code)".

Note. Effective only when the Network No. is set to 00h.

7.4.2 Details of End Code (End Code)

The end code (End Code) is an area where the server stores the error code in the response frame when an error exists in the client request frame.

[When the own station is a client]

During the response frame receive processing, the error code of an error detected in the request frame sent by the own station is stored. Refer to the user's manual of the request destination device (MELSEC or partner manufacturer product) and correct the request frame creation processing or the request send processing.

[When the own station is a server]

During the response frame send processing, store the error code of an error detected in the request frame sent by the client. The error code can be defined by a user. The following table lists error code examples.

Table 7.14 Examples of Error Codes Stored in End Code

No.	Category	Error Code	Description	Action
1	Successful completion	0000h	The request was processed normally.	—
2	General error	C059h	• There is an error in the command/subcommand specification. • A command other than that in the specified sequence was received.	Correct the command/subcommand, and send the request again.
3		C05Ch	There is an error in the request message.	Correct the request message, and send the request again.
4		C061h	The request data length and data size do not match.	Correct the request data or the request data length, and send the request again.
5		CEE0h	Another request is being executed. The request cannot be processed.	Wait for a while, and send the request again.
6		CEE1h	The request message size exceeds the range that can be processed.	Correct the request message, and send the request again.
7		CEE2h	The response message size exceeds the range that can be processed.	Correct the request message, and send the request again.
8	Server information	CF10h	The specified server information number does not exist.	Correct the server information number, and send the request again.
9	Communication settings	CF20h	An item that cannot be set is included in the request message.	Correct the setting item (CSP+), and send the request again.
10	Parameter settings	CF30h	The specified parameter ID does not exist.	Correct the parameter and the parameter ID (CSP+), and send the request again.
11		CF31h	The write exclusive start processing has not been performed. The request cannot be processed.	Execute the write exclusive start processing, and send the request again.
12	Communication status	CF70h	An error occurred in the communication path of the relay destination. The request cannot be processed.	Check the communication path, and send the request again.
13		CF71h	A timeout occurred. The processing was interrupted.	Check the status of the destination device, and send the request again.

7.4.3 SLMP Memory Read Frame

Executing the SLMP memory read command (Command=0613h, Sub Command=0000h) allows you to retrieve data from the buffer memory of another station (SLMP-compatible device).

The following shows the format of an SLMP memory read frame.

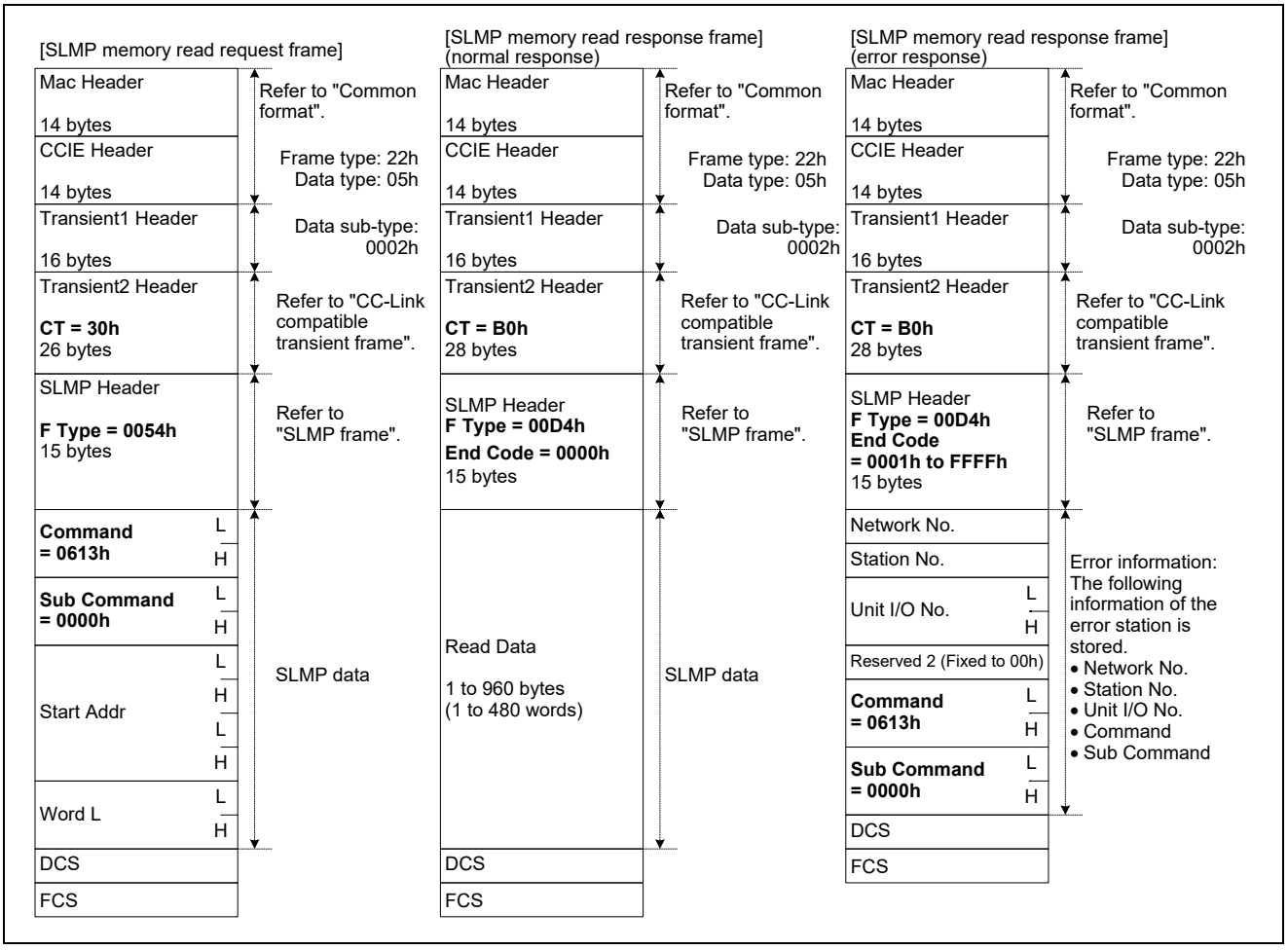


Figure 7.15Overview of SLMP Memory Read Frame Format

The following table describes the details of items defined in the SLMP memory read frame format.

Table 7.15 Details of SLMP Memory Read Frame Format

Item	Description	Value	Remarks
Command	Command	Fixed to 0613h	—
Sub Command	Subcommand	Fixed to 0000h	—
Start Addr	Start address	—	Specify the start address of the buffer memory to be read.
Word L	Word length	1h to 1E0h (1 to 480)	Specify the word length of the buffer memory to be read.

7.4.4 SLMP Memory Write Frame

Executing the SLMP memory write command (Command=1613h, Sub Command=0000h) allows you to write data to the buffer memory of another station (SLMP-compatible device).

The following shows the format of an SLMP memory write frame.

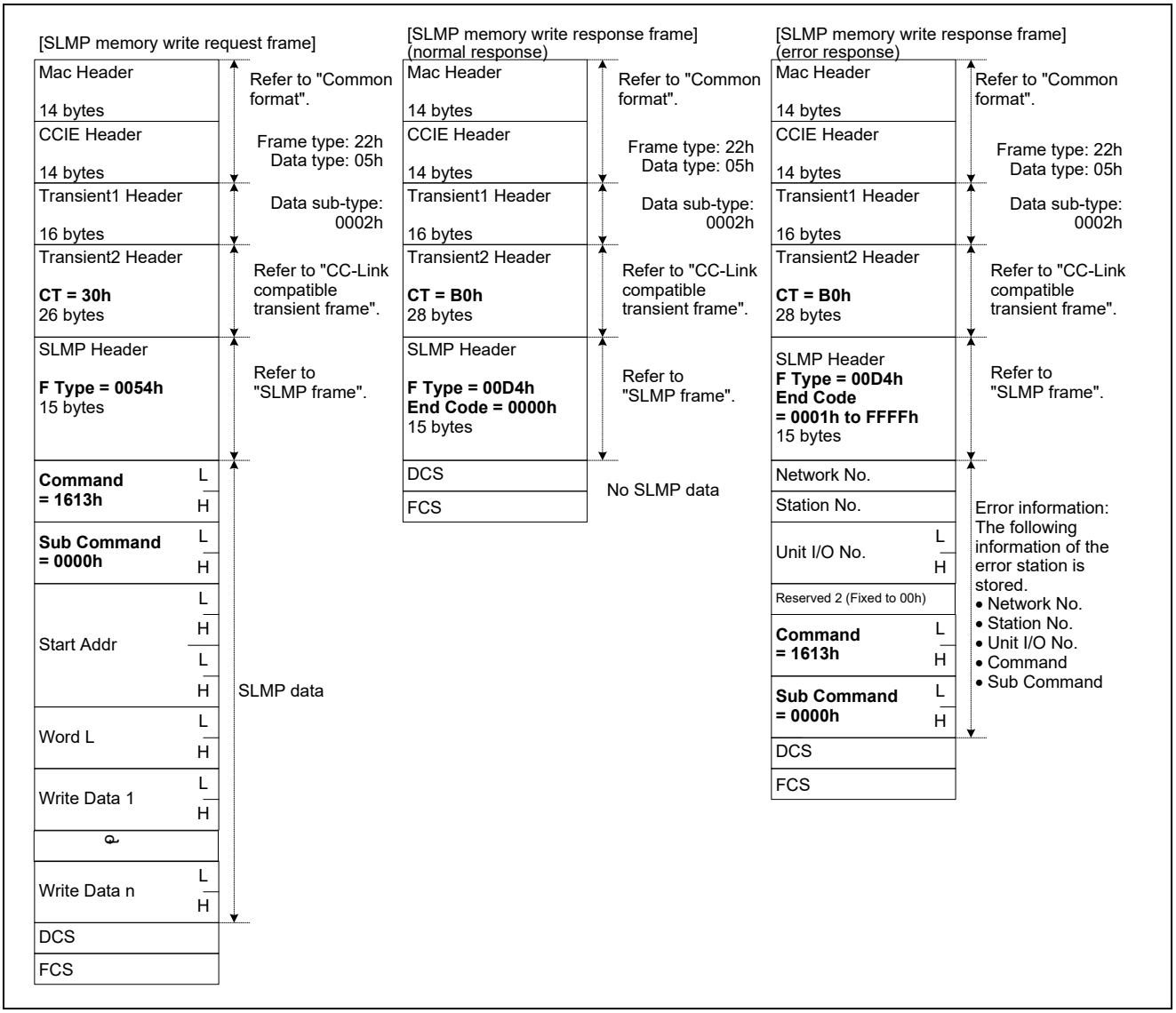


Figure 7.16Overview of SLMP Memory Write Frame Format

The following table describes the details of items defined in the SLMP memory write frame format.

Table 7.16 Details of SLMP Memory Write Frame Format

Item	Description	Value	Remarks
Command	Command	Fixed to 1613h	—
Sub Command	Subcommand	Fixed to 0000h	—
Start Addr	Start address	—	Specify the start address of the buffer memory to be written.
Word L	Word length	1h to 1E0h (1 to 480)	Specify the word length of the buffer memory to be written.

7.4.5 TransientAck Frame

The following shows the format of a TransientAck frame.

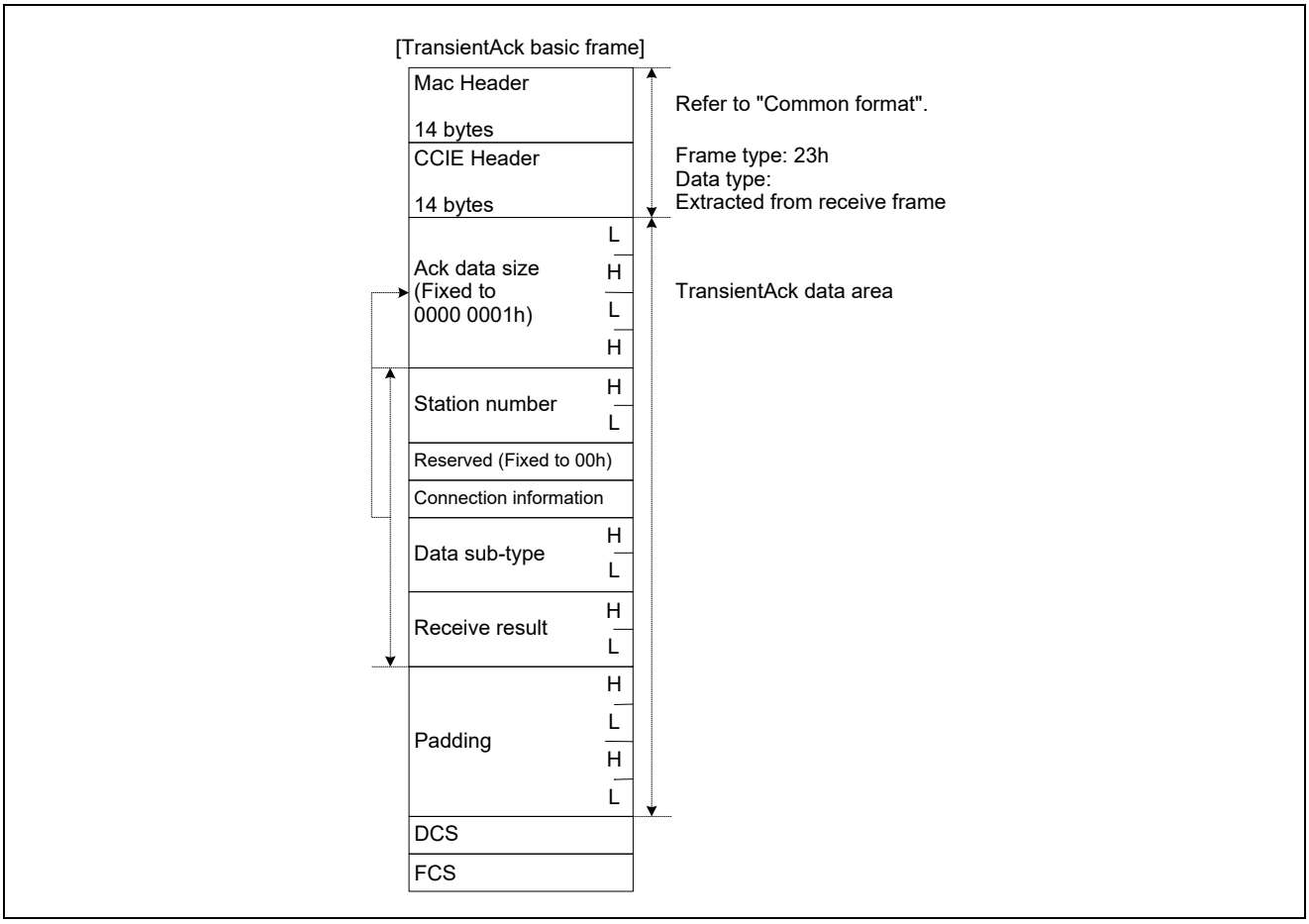


Figure 7.17Overview of TransientAck Frame Format

The following table describes the details of items defined in the TransientAck frame format.

Table 7.17 Details of TransientAck Frame Format

Item	Description	Value	Remarks
Ack data size	Data size from station number to receive result	Fixed value: 0000 0001h	—
Station number	TransientAck frame destination station number	Station number of received transient frame send source	Note 1
Connection information	Connection information loopback value of Ack send target frame	Connection information of received transient frame	—
Data sub-type	Data sub-type of received transient frame	Transient1: 0002h Transient2: 0000h	Note 2
Receive result	Receive result (RET) of transient frame	Normal: 0000h Error: Other than 0000h	—
Padding	Padding (16 bytes)	—	Note 3

Notes 1. When a transient frame is received from the master station (send source station number: 0000h), set the destination station number after converting the value to "007Dh".

2. There is no data sub-type for Transient2 frames.

Set "0000h" for an acknowledgement response to a Transient2 frame.

3. To satisfy the minimum Ethernet frame size of 64 bytes, padding is automatically performed by R-IN32M4-CL3.

REVISION HISTORY	R-IN32M4-CL3 User's Manual: CC-Link IE Field

Rev.	Date	Description	
		Page	Summary
1.00	Nov.1, 2019	-	First edition issued

[Memo]

R-IN32M4-CL3 User's Manual:

Publication Date: Rev.1.00 Nov. 1, 2019

Published by: Renesas Electronics Corporation

R-IN32M4-CL3 User's Manual

CC-Link IE Field

Renesas Electronics Corporation