CC-Link IE Field Network
Basic Sample Code User's Manual
(Slave Station)
Version 1.02.4

Revisions

Date	No.	Revision
2016/08/01	Version 1.00	First edition
2016/10/07	Version 1.01	Corrected minor errors
2016/12/27	Version 1.01.3	Porting for Linux
2017/03/21	Version 1.02.4	1. Changing Figure 8 Flowchart 2 for CCIEF_BASIC_SLAVE.c (Example of the Sample Program) 2. Adding the function for discarding request packet of another network on Raspbian(jessie). 3. Changing build sequence.

Contents

Relevant material

The following table lists the materials relevant to this manual.

No.	Publisher	Material name	Material number
1	CC-Link Partner Association (CLPA)	CC-Link IE Field Network Basic Specification (Application Layer Protocol)	BAP-C2010-004
2	CC-Link Partner Association (CLPA)	SLMP (Seamless Message Protocol) Specification (Overview)	BAP-C2006-001
3	CC-Link Partner Association (CLPA)	SLMP (Seamless Message Protocol) Specification (Services)	BAP-C2006-002
4	CC-Link Partner Association (CLPA)	SLMP (Seamless Message Protocol) Specification (Protocol)	BAP-C2006-003

Radix notation

The following radix notation is used in this manual except as otherwise specifically provided.

No.	Radix	Description	Example
1	Decimal	Units representing cardinal numbers are not added after numeric string.	0
2	Hexadecimal	The symbol 0x representing a hexadecimal is added before a numeric string.	0x00

Integral data types

The following integral data types is used in this manual except as otherwise specifically provided.

No.	Integral data type	Sign	Bit	Byte	C programming language (32 bits)
1	int8_t	Signed	8	1	char
2	int16_t	Signed	16	2	short
3	int32_t	Signed	32	4	int, long
4	int64_t	Signed	64	8	long long
5	uint8_t	Unsigned	8	1	unsigned char
6	uint16_t	Unsigned	16	2	unsigned short
7	uint32_t	Unsigned	32	4	unsigned int, unsigned long
8	uint64_t	Unsigned	64	8	unsigned long long

Precautions

The following lists the precautions about this manual.

- Since the attached sample codes are application examples, they do not guarantee the actual operation.
- This manual does not describe the terminology explanation of the CC-Link IE Field Network Basic and SLMP or troubleshooting. Acquire the relevant manuals from vendors of each product for reference, if necessary.
- Note that the descriptions of this manual and the sample code and specifications may be changed without notice.

1 Overview

This specification is for engineers to develop the CC-Link IE Field Network Basic slave station application.

2 Terminology

This manual uses the following generic terms and abbreviations for descriptions except as otherwise specifically provided.

Generic term and abbreviation	Description
CCIEF-BASIC	An abbreviation for CC-Link IE Field Network Basic
SLMP	An abbreviation for Seamless Message Protocol
SLMP information	The structure including the following information relevant to the SLMP packet: Network number, node number, processor number, packet data length, command, subcommand, and pointer to data
Master station	An abbreviation for CC-Link IE Field Network Basic master station
Slave station	An abbreviation for CC-Link IE Field Network Basic slave station

3 Function

The sample code provides the following functions.

Table 1 Function of Sample Code

No.	Name	Description
1	CCIEF-BASIC slave station	Performs cyclic transmission with the CCIEF-BASIC master station as a CCIEF-BASIC slave station.
2	SLMP server	Performs each of the following services as an SLMP server: memory reading and writing, detection of connected devices, parameter reading and writing, and remote set.

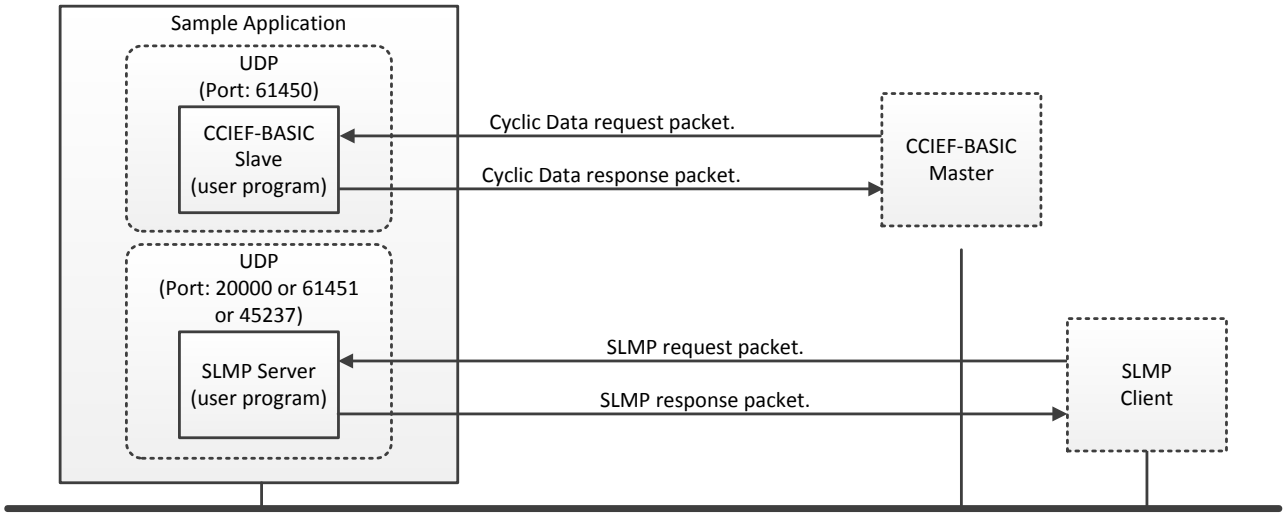

Figure 1 Function of Sample Code

4 Specifications

CCIEF-BASIC slave station

The following table lists the specifications of the CCIEF-BASIC slave station in the sample code.

Table 2 Specifications of CCIEF-BASIC Slave Station

Item		Description
Protocol		UDP
Port number		61450
IP address		IPv4 class C: Address range 192.0.0.1 to 223.255.255.254 Network address length: 24 bits, host address length: 8 bits (Subnet mask: 255.255.255.0)
Message format		SLMP with no serial number added (Binary mode)
Transmission format		Directed broadcast (Receive), Unicast (Send)
Number of occupied stations		1 to 16*1
Function		Cyclic transmission*2
Cyclic data	RX	64 bits (1 station occupied) to 1024 bits maximum (16 stations occupied)*1
	RY	64 bits (1 station occupied) to 1024 bits maximum (16 stations occupied)*1
	RWw	32 words (1 station occupied) to 512 words maximum (16 stations occupied)*1
	RWr	32 words (1 station occupied) to 512 words maximum (16 stations occupied)*1
Parameter		Text file of the CSV ("Comma-Separated Values") format. For details, refer to below.

*1 Conform to the set value of the number of occupied stations (1 to 16) of the parameters.

*2 The sample code sends back the cyclic data received from the master station.

The sample code defines the parameters of the CCIEF-BASIC slave stations as follows.
The parameters are described in the text file of the CSV ("Comma Separated Values") format.

Table 3 Parameter Specifications of Sample Code

ID	Parameter	Setting range	Description
1	IP address	192.0.0.1 to 223.255.255.254 (0: Default of network adapters)	Set the IP address of the slave station.
2	Subnet mask	255.255.255.0 (0: Default of network adapters)	Set the subnet mask of the slave station.
3	Default gateway IP address	192.0.0.1 to 223.255.255.254 (0: Default of network adapters)	Set the default gateway IP address of the slave station.
4	Number of occupied stations	1 to 16	Set the number of occupied stations of the slave station.
5	Cyclic response delay time	1 to 4294967295 [ms] (0: No delay)	Set the time to delay the response sent to the master station from the slave stations in cyclic transmission.

The following shows the examples of CCIEF-BASIC slave station parameters.

IP address = 192.168.3.1, Subnet mask = 255.255.255.0, Default gateway IP address = 192.168.3.254, Number of occupied stations = 1, Response delay time = None

[SlaveParameter.csv]

```
,,
CCIEF-BASIC Slave Sample Parameter,,
,,
ID,DATA,COMMENT
1,192.168.3.1,IP Address
2,255.255.255.0,Subnet Mask
3,192.168.3.254,Default Gateway IP Address
4,1,Occupied Station Number
5,0,Cyclic Response Wait Time
```

IP address, Subnet mask, and Default gateway IP address = 0 (Default of Network Adapters), Number of occupied stations = 3, Response delay time = 150 [ms]

[SlaveParameter.csv]

```
,,
CCIEF-BASIC Slave Sample Parameter,,
,,
ID,DATA,COMMENT
1,0,IP Address
2,0,Subnet Mask
3,0,Default Gateway IP Address
4,3,Occupied Station Number
5,150,Cyclic Response Wait Time
```

SLMP server

The following table lists the specifications of the SLMP server in the sample code.

Table 4 Specifications of SLMP Server

Item	Description
Protocol	UDP
Port number	20000*1, 61451, 45237
Message format	SLMP (Binary mode)
Transmission format	Limited broadcast (Send / Receive) or Unicast (Send / Receive)
Service	Refer to below.

*1 Change the port number according to the implementation environment.

The following table lists the services of SLMP server in the sample code.

Table 5 Service of SLMP Server

Class name	Service name	Port number	Command	Subcommand	Description
Memory	Memory Read	20000	0613	0000	Reads the internal memory*1.
	Memory Write	20000	1613	0000	Writes the data in the internal memory*1.
NodeConnect	NodeSearch	61451	0E30	0000	Responds to the server detection in the network.
	IPAddressSet	61451	0E31	0000	Sets the network information necessary for communications such as the IP address.*2
Parameter Setting	DeviceInfoCompare	20000	0E32	0000	Checks external devices.
	ParameterGet	20000	0E33	0000	Reads the parameter*3 values.
	ParameterSet	20000	0E34	0000	Writes the values of the parameter*3.
	ParameterSetStart	20000	0E35	0000	Starts the parameter write exclusive processing.
	ParameterSetEnd	20000	0E36	0000	Terminates the parameter write exclusive processing.
	ParameterSet Cancel	20000	0E3A	0000	Cancels the parameter write exclusive processing.
NodeMonitoring	Communication SettingGet	45237	0E45	0000	Reads the communication setting.
Remote Control	Read Type Name	20000	0101	0000	Reads the model name and model code.
	Remote Reset	20000	1006	0000	Resets the execution of the application.

*1 The sample code supports a memory space of 10 k words.

*2 The sample code supports the settings of the IP address and subnet mask of network adapters.

*3 The sample code supports the parameters of the CCIEF-BASIC slave station defined in the sample code.

5 Application Development

5.1 Development environment

The sample codes attached in this manual cause no compile error when "VC++ 2010 (Visual Studio 2010 Visual C++)" is used. Refer to the procedure from compilation to execution of the sample code using VC++ 2010 described in Chapter 5.

5.2 Development procedure

This section describes the procedure to develop an application using the attached sample code.
The sample code is configured with the program parts listed in Table 6. Change the SLMP library according to the implementation environment. In addition, change the contents of the user programs according to the application.

Table 6 Configuration of the Sample Code

No.	Program part	Overview
1	SLMP library	This function generates the SLMP packet and acquires the SLMP information from the packet. Change the program according to the implementation environment.
2	User program	This application program implements functions of the device. Sample codes to execute cyclic transmissions as a CC-Link IE Field Network Basic slave station using WinSock and sample codes to receive and send the SLMP packet as an SLMP server are described in this manual as an example. Change the program according to the environment.

The following describes the procedure to develop an application.

(1) Creating user program (CCIEF_BASIC_SLAVE.c, CCIEF_BASIC_SLAVE.h, SLMP_SERVER.c, SLMP_SERVER.h, SOCKET.c, SOCKET.h, TIMER.c, TIMER.h, USER_SAMPLE.c, USER_SAMPLE.h)
Create a user program. For details, refer to Section 5.5.

(2) Creating SLMP library (SLMP.c, SLMP.h)
After compiling the source code of the SLMP library included in the attached sample code, execute the librarian to create a library file.

(3) Linking user programs and library file
Link user programs and library file to create a load module file.

5.3 List of the sample code files

The following shows the directory configuration of the sample code.

```
root.   +- library      +- include    ... SLMP library header file
        |               +- src        ... SLMP library code file
        |
        +- sample       +- include    ... User program header file
                        +- src        ... User program code file
```

The following table lists the sample code files.

No.	Folder name			File name	Description
1	Root			version.txt	Version information
2				readme.txt	Help file
3		library	include	SLMP.h	SLMP library header
4			src	SLMP.c	SLMP library function
5		sample	include	CCIEF_BASIC_SLAVE.h	User program header (CCIEF-BASIC slave station)
6				SLMP_SERVER.h	User program header (SLMP server)
7				SOCKET.h	User program header (Socket)
8				TIMER.h	User program header (Timer)
9				USER_SAMPLE.h	User program header
10			src	CCIEF_BASIC_SLAVE.c	User program (CCIEF-BASIC slave station)
11				SLMP_SERVER.c	User program (SLMP server)
12				SOCKET.c	User program (Socket)
13				TIMER.c	User program (Timer)
14				USER_SAMPLE.c	User program

5.4 List of sample code functions

Table 7 lists the functions included in the sample code.

Table 7 List of Sample Code Functions

No.	Program part	File name	Function name	Function type	Overview	Disclosed/ undisclosed
1	SLMP library	SLMP.c	SLMP_MakePacketStream	int	SLMP packet generation	Disclosed
2			SLMP_GetSlmpInfo	int	SLMP information acquisition	Disclosed
3			local_itoa	uint8_t	Conversion from numeric string to ASCII	Disclosed
4			local_atoi	uint8_t	Conversion from ASCII to numeric string	Disclosed
5			SLMP_MakeErrorData	int	SLMP error response data generation	Disclosed
6	User program	CCIEF_ BASIC_ SLAVE.c	ccief_basic_slave_initialize	int	CCIEF-BASIC slave station initialization	Disclosed
7			ccief_basic_slave_terminate	void	CCIEF-BASIC slave station termination	Disclosed
8			ccief_basic_slave_main	int	CCIEF-BASIC slave station main processing	Disclosed
9			ccief_basic_slave_set_rx	int	RX data setting	Disclosed
10			ccief_basic_slave_get_ry	int	RY data acquisition	Disclosed
11			ccief_basic_slave_get_rww	int	RWw data acquisition	Disclosed
12			ccief_basic_slave_set_rwr	int	RWr data setting	Disclosed
13			ccief_basic_slave_get_ pointer	uint16_t *	Device head pointer acquisition	Disclosed
14			ccief_basic_slave_set_unit_ info	void	Own station unit information setting	Disclosed
15			ccief_basic_slave_set_err_ code	void	Error code setting	Disclosed
16			ccief_basic_slave_set_unit_ data	void	Own station management information setting	Disclosed
17			ccief_basic_slave_get_ master_info	void	Master station information acquisition	Disclosed
18			ccief_basic_slave_recv_ cyclic_data	int	Cyclic data receiving	Undisclosed
19			ccief_basic_slave_send_ cyclic_data	int	Cyclic data sending	Undisclosed
20			ccief_basic_slave_send_ cyclic_data_error	int	Cyclic error data sending	Undisclosed
21			ccief_basic_slave_ disconnection	void	Disconnection processing	Undisclosed
22			ccief_basic_slave_ disconnection_timer_timeout	void	Timeout of disconnection detection period (callback function)	Undisclosed

No.	Program part	File name	Function name	Function type	Overview	Disclosed/ undisclosed
23	User program	SLMP_ SERVER .c	slmp_server_initialize	int	SLMP server initialization	Disclosed
24			slmp_server_terminate	void	SLMP server termination	Disclosed
25			slmp_server_main	int	SLMP server main processing	Disclosed
26			slmp_server_user_port	int	Receive processing for the user-specified port of SLMP server	Disclosed
27			slmp_server_basic_port	int	Receive processing for the CCIEF-BASIC NodeConnect port of SLMP Server	Disclosed
28			slmp_server_paramset_port	int	Receive processing for the parameter setting port of SLMP server	Disclosed
29			slmp_server_set_status	void	SLMP server status setting	Disclosed
30			slmp_server_slmp_send_ response	int	SLMP response sending	Undisclosed
31			slmp_server_slmp_send_err _response	int	SLMP error response sending	Undisclosed
32			slmp_server_service	int	SLMP service execution	Disclosed
33			slmp_server_memory_read	int	Memory reading	Disclosed
34			slmp_server_memory_write	int	Memory writing	Disclosed
35			slmp_server_node_search_ basic	int	Automatic detection (for CCIEF-BASIC)	Disclosed
36			slmp_server_node_search_ send_response_timeout	void	Timeout to wait for automatic detection response sending (callback function)	Undisclosed
37			slmp_server_ip_address_set _basic	int	Communication setting (for CCIEF-BASIC)	Disclosed
38			slmp_server_device_info_ compare	int	Device connection information check	Disclosed
39			slmp_server_parameter_get	int	Parameter reading	Disclosed
40			slmp_server_parameter_set	int	Parameter writing	Disclosed
41			slmp_server_parameter_set _start	int	Start of parameter write exclusive processing	Disclosed
42			slmp_server_parameter_set _end	int	Termination of parameter write exclusive processing	Disclosed
43			slmp_server_parameter_set _cancel	int	Cancel of parameter write exclusive processing	Disclosed
44			slmp_server_communication _setting_get	int	Communication settings acquisition	Disclosed
45			slmp_server_read_type_ name	int	Model name reading	Disclosed
46			slmp_server_remote_reset	int	Remote reset	Disclosed

No.	Program part	File name	Function name	Function type	Overview	Disclosed/ undisclosed
47	User program	SOCKET .c	socket_initialize	int	Socket initialization	Disclosed
48			socket_terminate	void	Socket termination	Disclosed
49			socket_recv	int	Packet receiving	Disclosed
50			socket_send	int	Packet sending	Disclosed
51		TIMER.c	timer_initialize	void	Timer initialization	Disclosed
52			timer_terminate	void	Timer termination	Disclosed
53			timer_main	void	Timer main processing	Disclosed
54			timer_start	int	Timer start	Disclosed
55			timer_stop	void	Timer stop	Disclosed
56			timer_get_time	uint32_t	Current time acquisition	Disclosed
57			timer_broadcast_send_wait_ time	uint32_t	Broadcast send waiting time acquisition	Disclosed
58			timer_analyze_time_data	void	Clock time analysis	Disclosed
59		USER_ SAMPLE .c	main	void	Main processing	Disclosed
60			user_callback_recv_cyclic_ data	void	Cyclic data receiving (callback function)	Disclosed
61			user_callback_cyclic_ disconnection	void	Disconnection detection (callback function)	Disclosed
62			user_callback_set_ip_ address_basic	void	Communication setting (for CCIEF-BASIC) (callback function)	Disclosed
63			user_callback_parameter_ get	int	Parameter reading (callback function)	Disclosed
64			user_callback_parameter_ set	int	Parameter writing (callback function)	Disclosed
65			user_callback_parameter_ set_end	int	Parameter writing completed (callback function)	Disclosed
66			user_callback_remote_reset	int	Remote reset (callback function)	Disclosed
67			user_parameter_file_read	int	Parameter file reading	Undisclosed
68			user_parameter_file_write	int	Parameter file writing	Undisclosed
69			user_display_cyclic_ information	void	Cyclic information display	Undisclosed
70			user_get_adapter_info	int	Network adapter information acquisition	Undisclosed
71			user_set_adapter_info	int	Network adapter information setting	Undisclosed

Disclosed: The functions to be disclosed outside. Undisclosed: The functions to be used in the local file.

5.5 Creating a user program

Create a user program according to the implementation environment.
The following table lists the user program files.

Table 8 List of User Program Files

No.	File name	Description
1	CCIEF_BASIC_SLAVE.c	Performs cyclic transmission with the CCIEF-BASIC master station.
2	SLMP_SERVER.c	After receiving an SLMP request packet from an SLMP client, executes each service processing and sends the SLMP response packet to the SLMP client.
3	SOCKET.c	Provides a set of functions to execute socket processing.
4	TIMER.c	Provides the library to execute timer processing.
5	USER_SAMPLE.c	Executes initialization and main processing of the CCIEF-BASIC slave station and SLMP server, and reads and writes parameter files. This program also implements callback functions and executes cyclic data processing with the CCIEF-BASIC master station and each service processing of the SLMP server.

The following shows the points requiring changes in the program due to differences in the operating system and protocol stack in the implementation environment.

Table 9 Points Requiring Changes in the Program due to Differences in the Operating System and Protocol Stack

No.	File name	Function name	Points to be changed
1	CCIEF_BASIC_SLAVE.c	ccief_basic_slave_main	Method to implement the socket function and structure
2		ccief_basic_slave_recv_cyclic_data	Method to implement the socket function and structure
3		ccief_basic_slave_send_cyclic_data	Method to implement the socket function and structure
4	SLMP_SERVER.c	slmp_server_user_port	Method to implement the socket function and structure
5		slmp_server_basic_port	Method to implement the socket function and structure
6		slmp_server_paramset_port	Method to implement the socket function and structure
7		slmp_server_slmp_send_response	Method to implement the socket function and structure
8		slmp_server_slmp_send_err_response	Method to implement the socket function and structure
9		slmp_server_service	Method to implement the socket function and structure
10		slmp_server_memory_read	Method to implement the socket function and structure
11		slmp_server_memory_write	Method to implement the socket function and structure
12		slmp_server_node_search_basic	Method to implement the socket function and structure
13		slmp_server_node_search_send_response_timeout	Method to implement the socket function and structure

No.	File name	Function name	Points to be changed
14	SLMP_SERVER.c	slmp_server_ip_address_set_basic_basic	Method to implement the socket function and structure
15		slmp_server_device_info_compare	Method to implement the socket function and structure
16		slmp_server_parameter_get	Method to implement the socket function and structure
17		slmp_server_parameter_set	Method to implement the socket function and structure
18		slmp_server_parameter_set_start	Method to implement the socket function and structure
19		slmp_server_parameter_set_end	Method to implement the socket function and structure
20		slmp_server_parameter_set_cancel	Method to implement the socket function and structure
21		slmp_server_communication_setting_get	Method to implement the socket function and structure
22		slmp_server_remote_reset	Method to implement the socket function and structure
23	SOCKET.c	socket_initialize	Method to implement the socket function and structure* 1
24		socket_terminate	Method to implement the socket function and structure
25		socket_recv	Method to implement the socket function and structure
26		socket_send	Method to implement the socket function and structure
27	TIMER.c	timer_get_time	Method to acquire the system time
28		timer_broadcast_send_wait_time	Method to acquire the system time
29	USER_SAMPLE.c	main	Method to implement the socket function and structure
30		user_callback_parameter_get	Method to implement the socket function and structure
31		user_callback_parameter_set	Method to implement the socket function and structure
32		user_parameter_file_read	Method to implement the socket function and structure
33		user_parameter_file_write	Method to implement the socket function and structure
34		user_get_adapter_info	Method to acquire the network adapter information
35		user_set_adapter_info	Method to set the network adapter information.

*1 Non blocking mode setting is required.

The following describes the key points in creating a user program in every file.

(1) CCIEF_BASIC_SLAVE.c

This program performs cyclic transmission with the CCIEF-BASIC master station.

(a) Cyclic data

The sample code defines the cyclic data (RX, RY, RWw, RWr) internally. The user program realizes cyclic transmission with the master station by accessing the cyclic data at an arbitrary timing. As cyclic data access methods, a method to directly access cyclic data devices and a method to acquire the head pointer of each device and access with the pointer are defined.

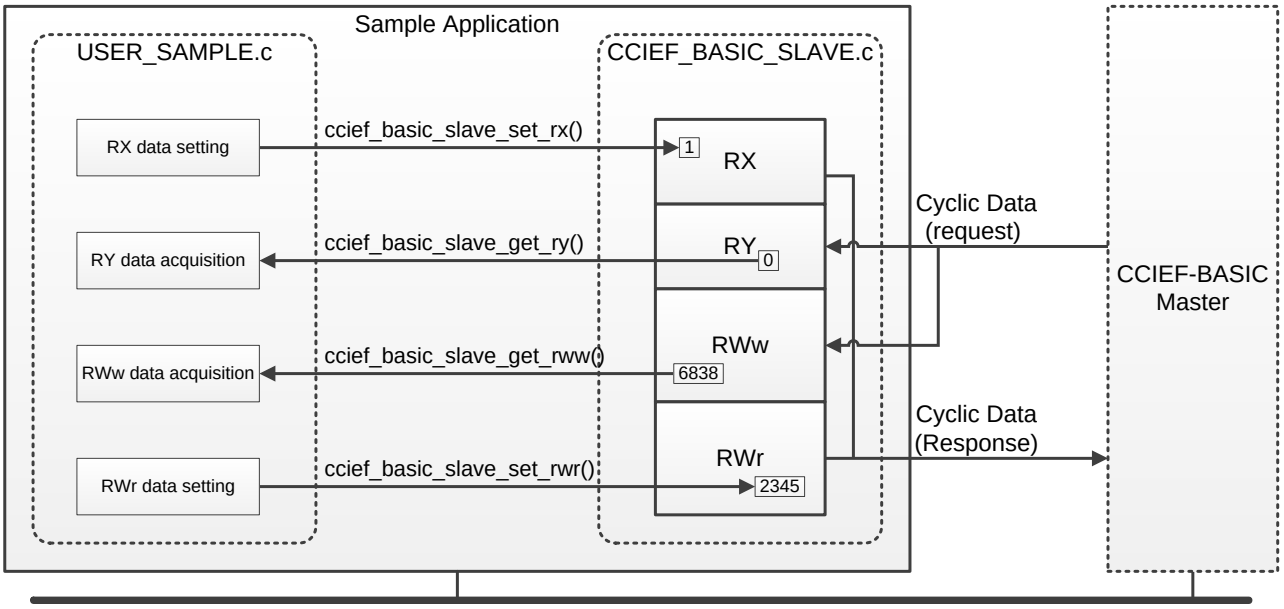

Figure 2 Method to Directly Access Cyclic Data Devices

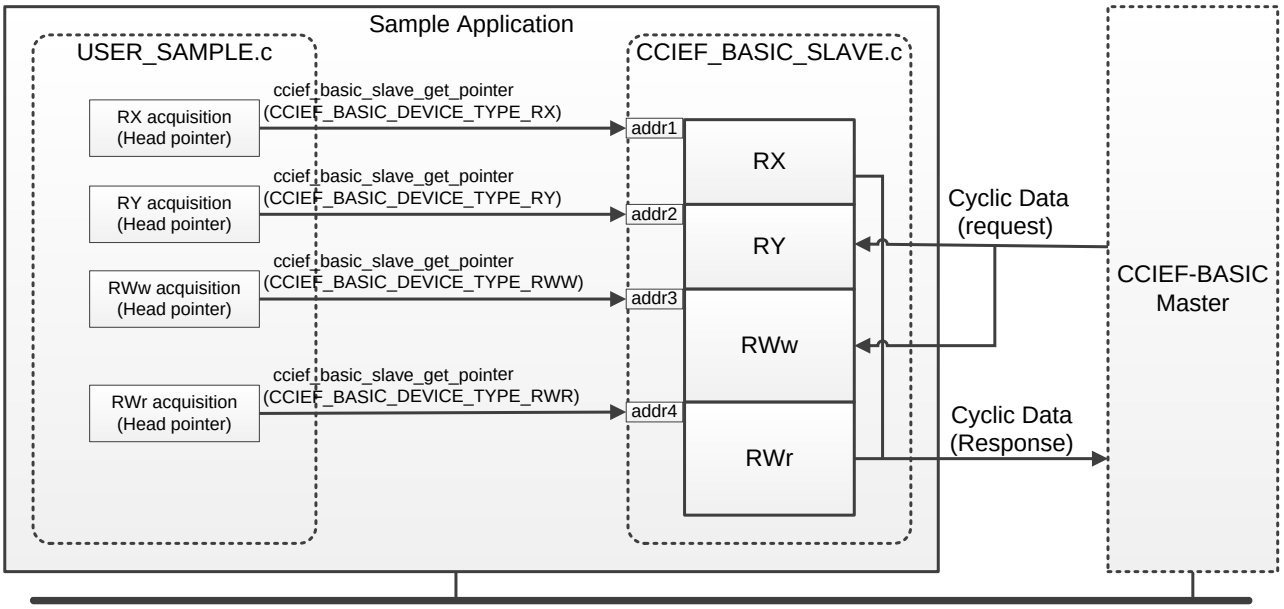

Figure 3 Method to Acquire the Head Pointer of Each Device of the Cyclic Data

(b) Callback function

The sample code defines the callback functions and executes the specified callback function in the following timing.
The user can easily develop a function by implementing the callback functions.

Table 10 List of Callback Functions

No.	Callback function	Execution timing
1	CCIEF_BASIC_SLAVE_CALLBACK_ RECV_CYCLIC_DATA	When receiving cyclic request data from the CCIEF-BASIC master station
2	CCIEF_BASIC_SLAVE_CALLBACK_ CYCLIC_DISCONNECTION	When detecting disconnection from the CCIEF-BASIC master station or cyclic stop

[CCIEF_BASIC_SLAVE.h]

```
typedef void(*CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA)
(int iCyclicState, int iOccupiedStationNumber);

typedef void(*CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION)(void);
```

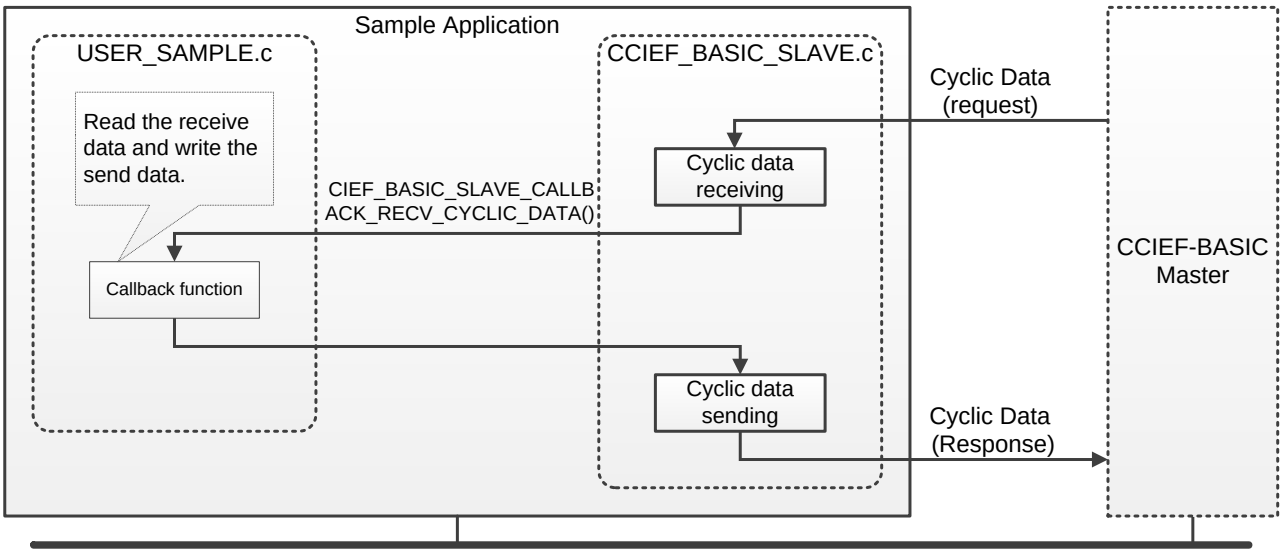

Figure 4 Image of the Callback Function

(c) Setting of slave station notification information

The sample code can set the information to notice the master station by sending the cyclic transmission response in the user program.

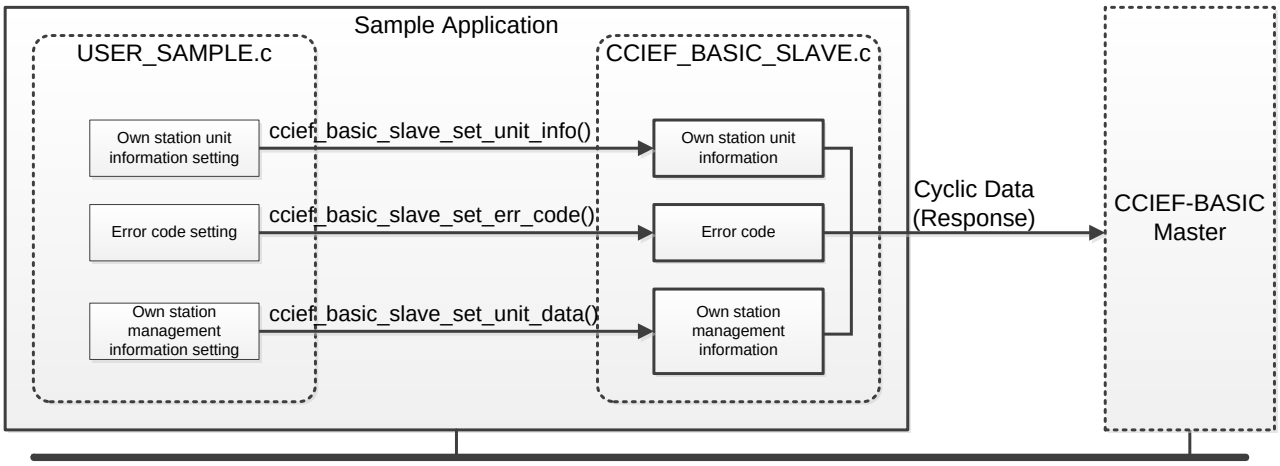

Figure 5 Setting the Slave Station Notification Information

(d) Master station information acquisition

The sample code can get the master station information received via the cyclic transmission data by the user program.

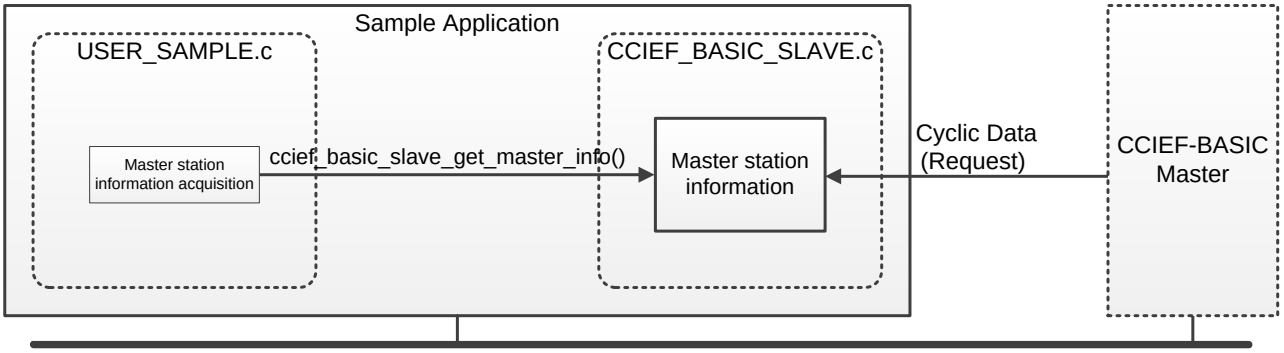

Figure 6 Acquiring the Master Station Information

(e) Flowchart

The following figures show the flowchart of the sample program.

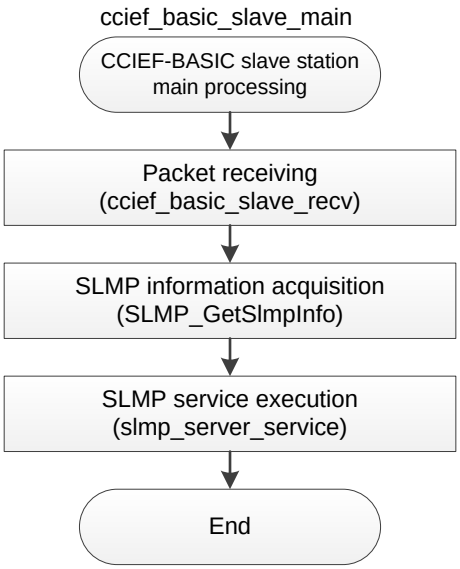

Figure 7 Flowchart 1 for CCIEF_BASIC_SLAVE.c (Example of the Sample Program)

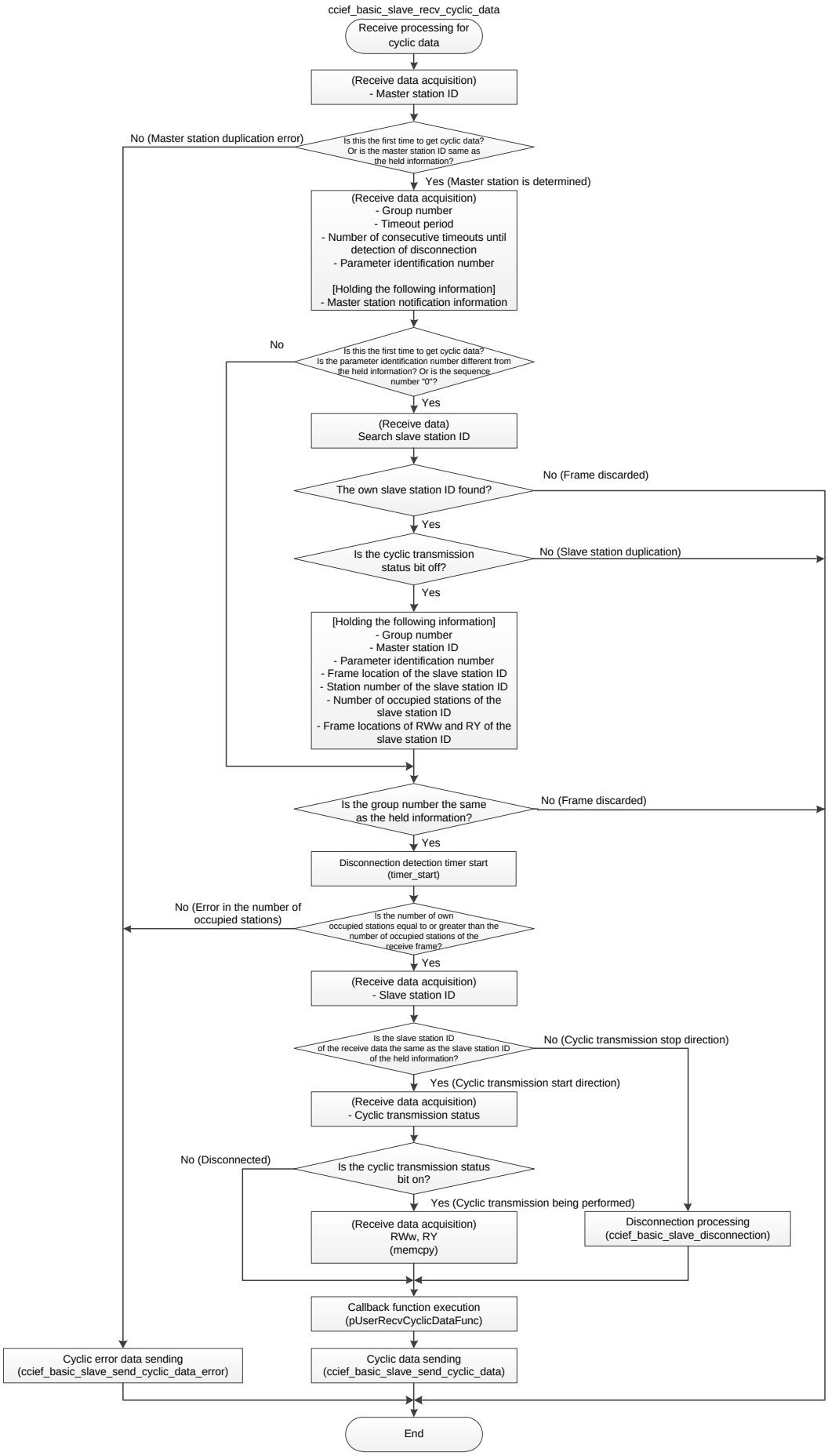

Figure 8 Flowchart 2 for CCIEF_BASIC_SLAVE.c (Example of the Sample Program)

(2) SLMP_SERVER.c

After receiving an SLMP request packet from an SLMP client, this program executes each service processing and sends the SLMP response packet to the SLMP client.

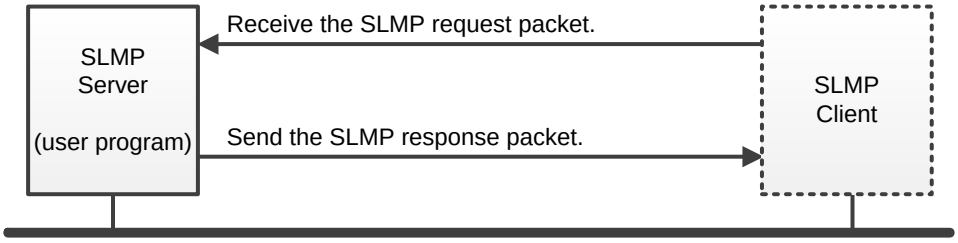

Figure 9 SLMP Server

(a) Callback function

The sample code defines the following callback functions.
The user can easily develop a function by implementing the callback functions.

Table 11 List of Callback Functions

No.	Callback function	Execution timing
1	SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC	When the SLMP server has received a command request for communication setting (0E31)
2	SLMP_SERVER_CALLBACK_PARAMETER_GET	When the SLMP server has received a command request for parameter reading (0E33)
3	SLMP_SERVER_CALLBACK_PARAMETER_SET	When the SLMP server has received a command request for parameter writing (0E34)
4	SLMP_SERVER_CALLBACK_PARAMETER_SET_END	When the SLMP server has received a command request for parameter writing completion (0E36)
5	SLMP_SERVER_CALLBACK_REMOTE_RESET	When the SLMP server has received a command request for remote reset (1006)

[SLMP_SERVER.h]

```
typedef void(*SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC)
(uint32_t ulIpAddress, uint32_t ulSubnetMask );
typedef int(*SLMP_SERVER_CALLBACK_PARAMETER_GET)(uint16_t usId, uint16_t *pusSize, uint8_t **ppucData);
typedef int(*SLMP_SERVER_CALLBACK_PARAMETER_SET)(uint16_t usId, uint16_t usSize, uint8_t *pucData);
typedef int(*SLMP_SERVER_CALLBACK_PARAMETER_SET_END)(void);
typedef int(*SLMP_SERVER_CALLBACK_REMOTE_RESET)(void);
```

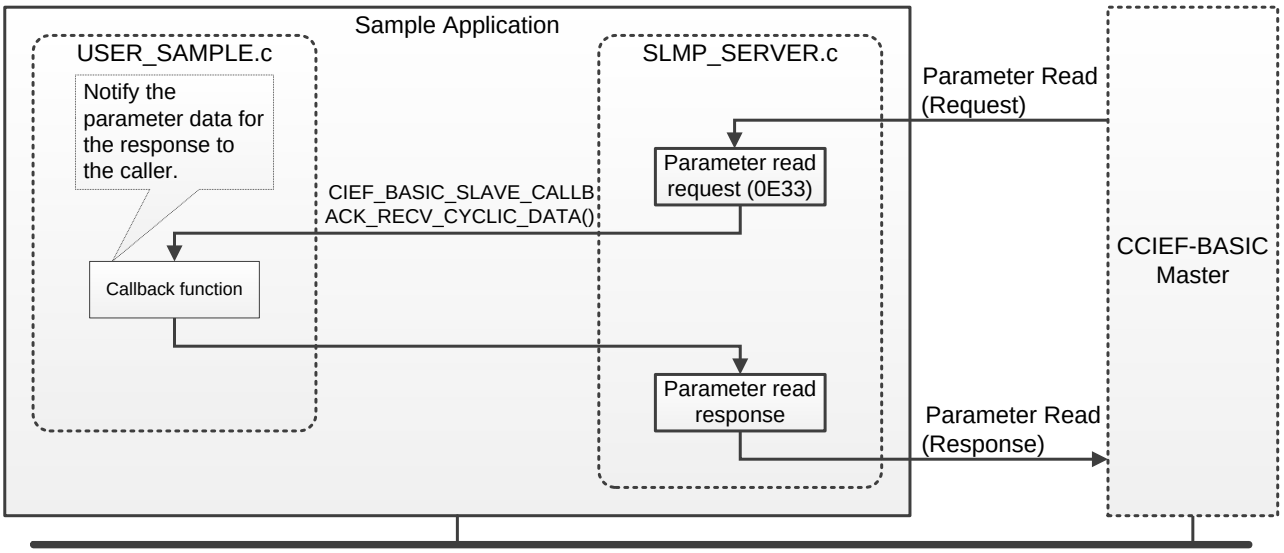

Figure 10 Image of the Callback Function

(b) Flowchart

The following figures show the flowchart of the sample program.

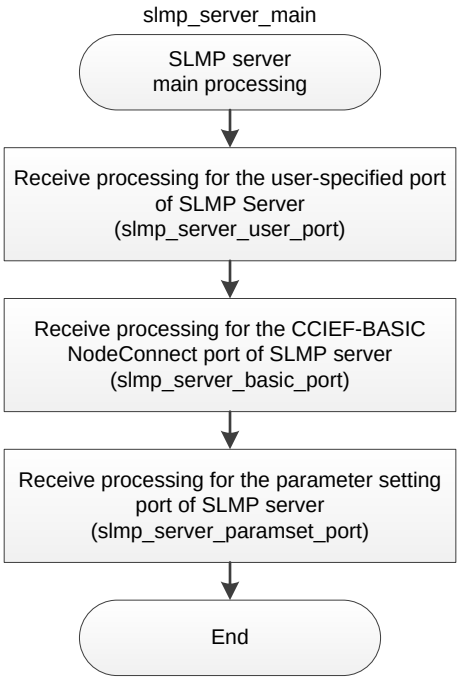

Figure 11 Flowchart 1 for SLMP_SERVER.c (Example of the Sample Program)

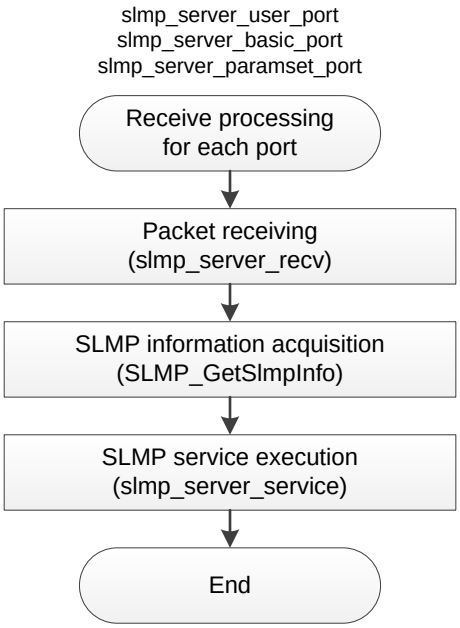

Figure 12 Flowchart 2 for SLMP_SERVER.c (Example of the Sample Program)

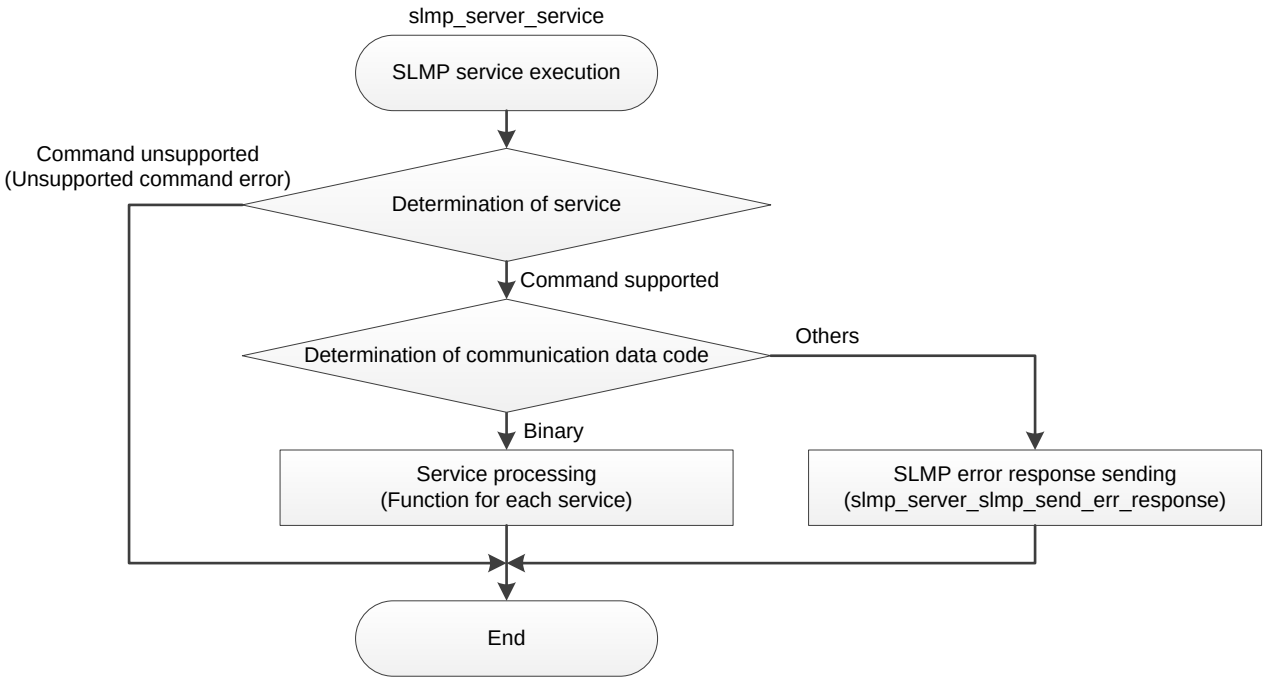

Figure 13 Flowchart 3 for SLMP_SERVER.c (Example of the Sample Program)

(3) SOCKET.c

This program provides a set of functions to execute socket processing.
* Change the program according to the implementation environment.

(4) TIMER.c

This program provides the library to execute timer processing.
* Change the method to acquire the elapsed time (processor time) or others according to the implementation environment.

(a) Callback function

The sample code defines the following callback functions. The callback function is executed when the registered timer has timed out.

[TIMER.h]

```
typedef void (*TIMER_CALLBACK)( int iId, void *pCallbackArg );
```

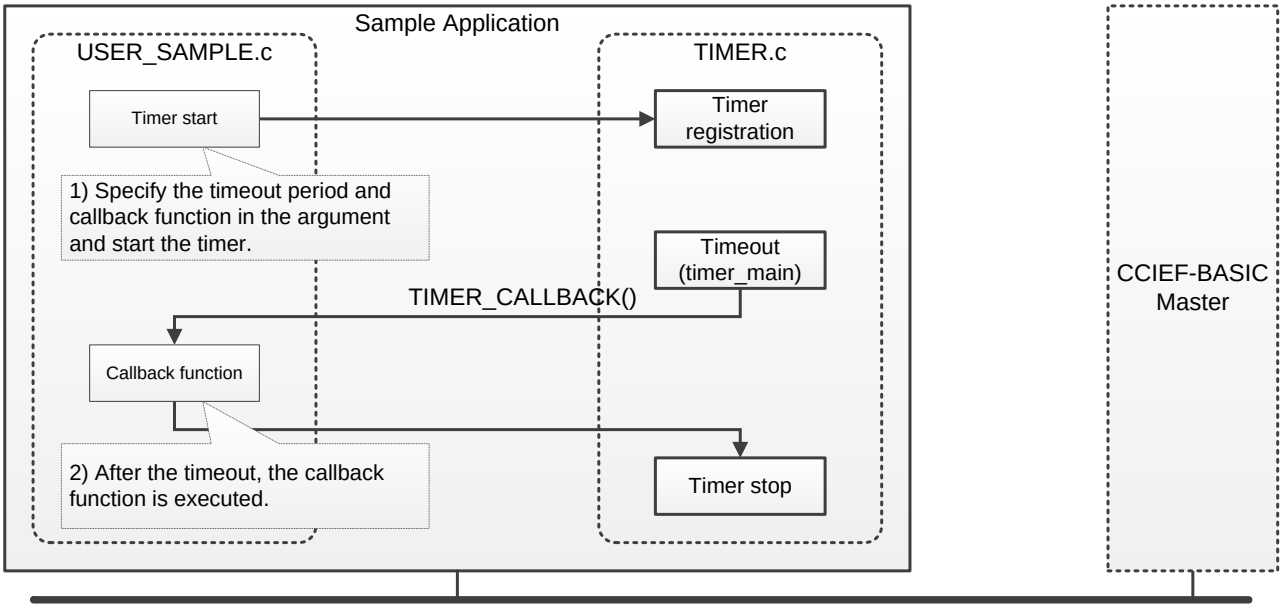

Figure 14 Image of the Callback Function

(b) Flowchart

The following figures show the flowchart of the sample program.

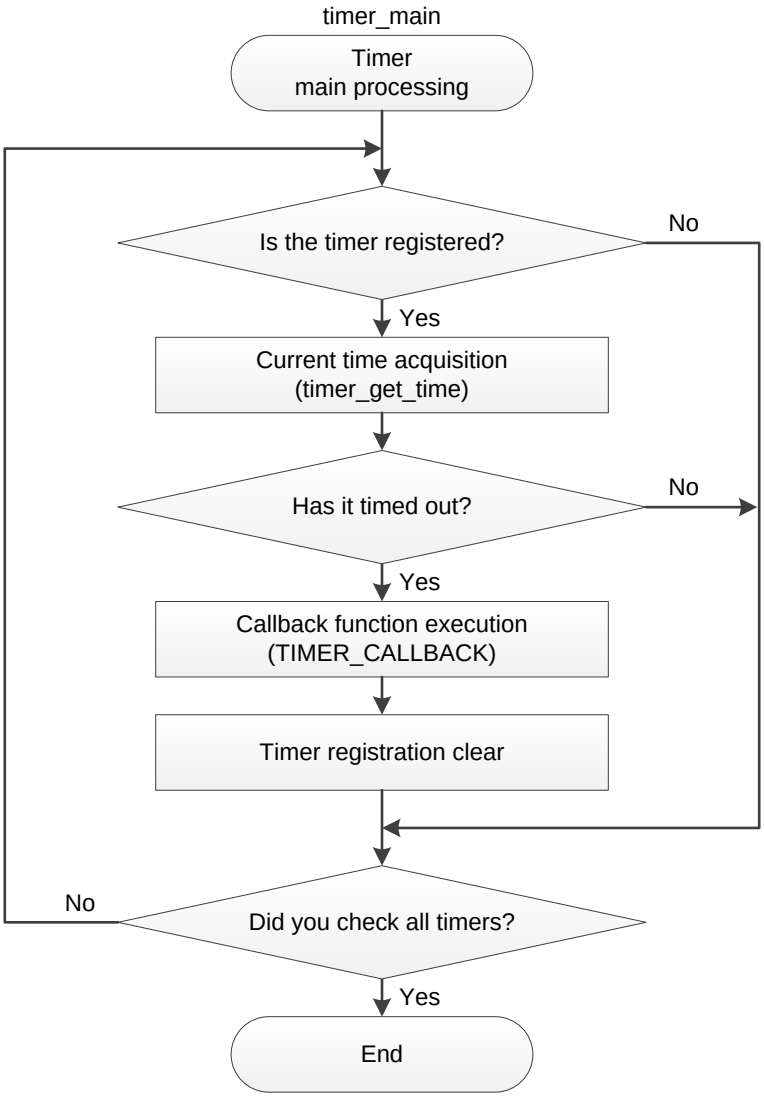

Figure 15 Flowchart for TIMER.c (Example of the Sample Program)

(5) USER_SAMPLE.c

This program execute initialization and main processing of the CCIEF-BASIC slave station and SLMP server, and reads and writes parameter files.
This program also implements callback functions and executes cyclic data processing with the CCIEF-BASIC master station and each service processing of the SLMP server.

(a) Implementing a program

The sample program implements the callback functions provided by CCIEF_BASIC_SLAVE.c and SLMP_SERVER.c. The following table lists the implementation content of the sample program.
* Change the program according to the implementation environment.

Table 12 Implementation Content of the Sample Program

No.	Program	Implementation content	Callback function of the implementation source
1	user_callback_recv_ cyclic_data	Sets the RWw and RY data of the slave station back to RWr and RX.	CCIEF_BASIC_SLAVE_CALLBACK_ RECV_CYCLIC_DATA (Refer to Table 10.)
2	user_callback_cyclic_ disconnection	Sets the own station unit information as application stopped.	CCIEF_BASIC_SLAVE_CALLBACK_ CYCLIC_DISCONNECTION (Refer to Table 10.)
3	user_callback_set_ip_ address_basic	Reflects the communication setting and executes reset. (For CCIEF-BASIC)	SLMP_SERVER_CALLBACK_ IPADDRESS_SET_BASIC (Refer to Table 11.)
4	user_callback_parameter _get	Replies the values of the parameters of the slave station.	SLMP_SERVER_CALLBACK_ PARAMETER_GET (Refer to Table 11.)
5	user_callback_parameter _set	Reflects the values into the parameters of the slave station.	SLMP_SERVER_CALLBACK_ PARAMETER_SET (Refer to Table 11.)
6	user_callback_parameter _set_end	Writes parameters into the file.	SLMP_SERVER_CALLBACK_ PARAMETER_SET_END (Refer to Table 11.)
7	user_callback_remote_ reset	Executes reset.	SLMP_SERVER_CALLBACK_REMOTE _RESET (Refer to Table 11.)

(b) Flowchart

The following figure shows the flowchart of the sample program.

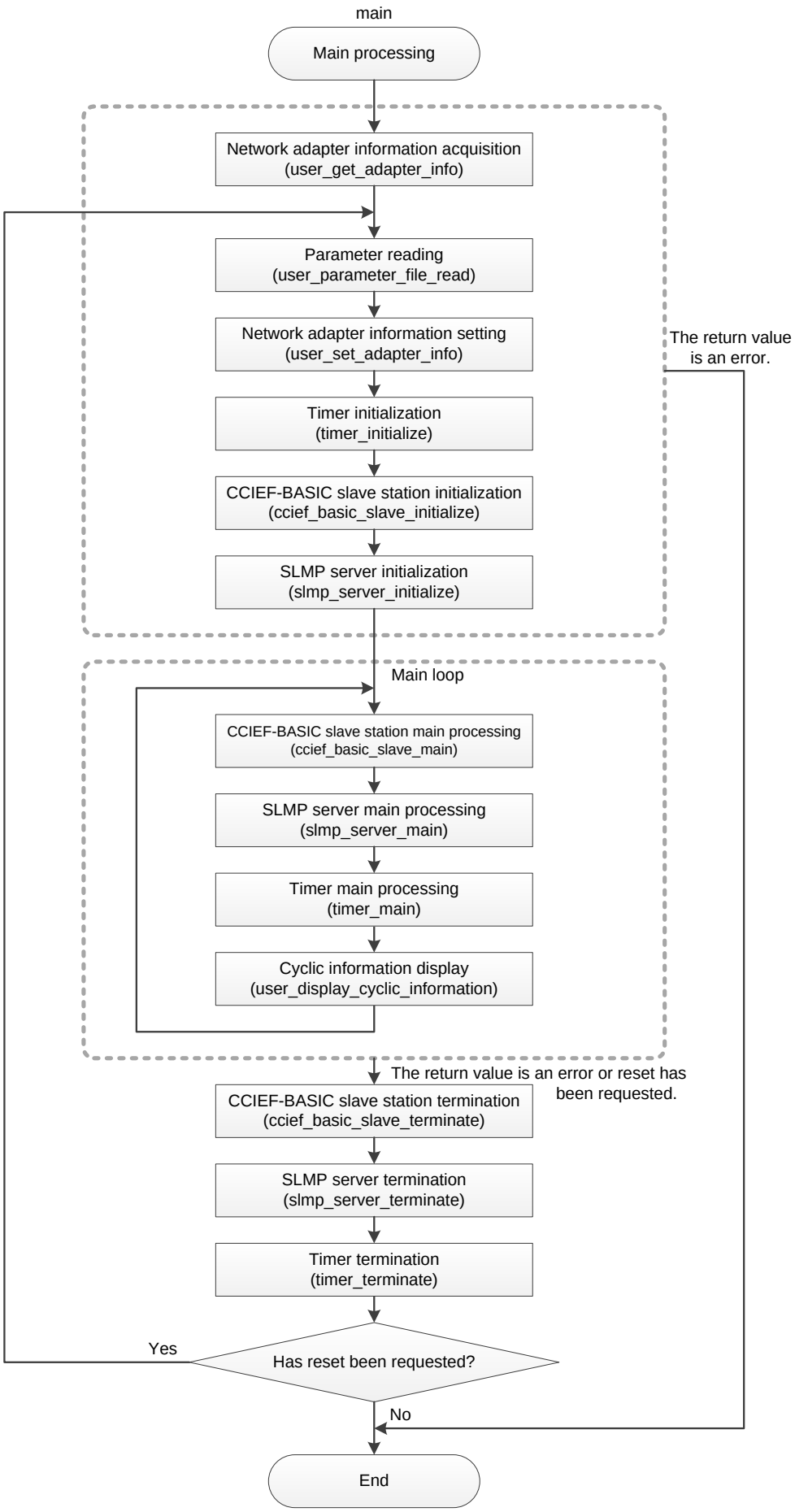

Figure 16 Flowchart for USER_SAMPLE.c (Example of the Sample Program)

5.6 Function details

5.6.1 Definition of the return value

The following codes are used as error codes and end codes returned as function return values in the SLMP library.

[SLMP.h]

```
#define SLMP_ERR_OK                                          0
#define SLMP_ERR_NG                                          (-1)
#define SLMP_ERR_COMMAND_SUBCOMMAND                          (0xC059)
#define SLMP_ERR_WRONG_DATA                                  (0xC05C)
#define SLMP_ERR_DATA_LENGTH                                 (0xC061)
#define SLMP_ERR_UNDER_EXECUTION                             (0xCEE0)
#define SLMP_ERR_REQ_DATA_SIZE                               (0xCEE1)
#define SLMP_ERR_RES_DATA_SIZE                               (0xCEE2)
#define SLMP_ERR_NO_EXIST_SERVER_NO                          (0xCF10)
#define SLMP_ERR_CAN_NOT_COMMUNICATION_SETTING               (0xCF20)
#define SLMP_ERR_NO_EXIST_PARAM_ID                           (0xCF30)
#define SLMP_ERR_CAN_NOT_PARAMETER_SET                       (0xCF31)
#define SLMP_END_DUPLICATE_MASTER                            (0xCFE0)
#define SLMP_END_INVALID_NUMBER_OF_OCCUPIED_STATIONS         (0xCFE1)
#define SLMP_END_SLAVE                                       (0xCFF0)
#define SLMP_END_DISCONNECTED_REQUEST                        (0xCFFF)
```

The following codes are used as error codes returned as function return values in the user program.

[CCIEF_BASIC_SLAVE.h]

```
#define CCIEF_BASIC_SLAVE_ERR_OK                             0
#define CCIEF_BASIC_SLAVE_ERR_NG                             (-1)
#define CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE                   (-100)
```

[SLMP_SERVER.h]

```
#define SLMP_SERVER_ERR_OK                                   0
#define SLMP_SERVER_ERR_NG                                   (-1)
#define SLMP_SERVER_ERR_UNSUPPORT_SERVICE                    (-2)
```

[SOCKET.h]

```
#define SOCKET_ERR_OK                                        0
#define SOCKET_ERR_SOCKET                                    (-100)
#define SOCKET_ERR_RECV                                      (-103)
#define SOCKET_ERR_SEND                                      (-104)
#define SOCKET_ERR_NO_RECEIVABLE                             (-200)
```

[TIMER.h]

```
#define TIMER_OK                                             0
#define TIMER_RESOURCE_NONE                                  (-1)
```

[USER_SAMPLE.h]

```
#define USER_ERR_OK                                          0
#define USER_ERR_NG                                          (-1)
```

5.6.2 SLMP_MakePacketStream

Table 13 SLMP_MakePacketStream

Function	SLMP packet generation			
File name	SLMP.c		Disclosed/undisclosed	Disclosed
Call format	int SLMP_MakePacketStream (uint32_t ulFrameType, const SLMP_INFO *p, uint8_t *pucStream)			
Argument	Type	Variable name	Description	I/O
	uint32_t	ulFrameType	Frame type	Input
	const SLMP_INFO *	p	SLMP information	Input
	uint8_t *	pucStream	Send packet	Output
Return value	SLMP_ERR_OK: Normal SLMP_ERR_NG: Error			
Description	This function generates an SLMP communication packet.			

The following shows the configuration of SLMP_INFO based on the sample code.

[SLMP.h]

```
typedef struct
{
        uint32_t    ulFrameType;            /* Frame Type */
        uint16_t    usSerialNumber;         /* Serial Number */
        uint16_t    usNetNumber;            /* Network Number */
        uint16_t    usNodeNumber;           /* Node Number */
        uint16_t    usProcNumber;           /* Processor Number */
        uint16_t    usDataLength;           /* Data Length */
        uint16_t    usTimer;                /* Timer Value */
        uint16_t    usCommand;              /* Command */
        uint16_t    usSubCommand;           /* Sub Command */
        uint16_t    usEndCode;              /* End Code */
        uint8_t     *pucData;               /* Data */
}SLMP_INFO;
```

5.6.3 SLMP_GetSlmpInfo

Table 14 SLMP_GetSlmpInfo

Function	SLMP information acquisition			
File name	SLMP.c		Disclosed/undisclosed	Disclosed
Call format	int SLMP_GetSlmpInfo (SLMP_INFO *p, const uint8_t *pucStream)			
Argument	Type	Variable name	Description	I/O
	SLMP_INFO *	p	SLMP information	Output
	uint8_t *	pucStream	Receive packet	Input
Return value	SLMP_ERR_OK: Normal SLMP_ERR_NG: Error			
Description	This function acquires SLMP information.			

5.6.4 local_itoa

Table 15 local_itoa

Function	Conversion from numeric string to ASCII			
File name	SLMP.c		Disclosed/undisclosed	Disclosed
Call format	uint8_t local_itoa (uint8_t ucInt)			
Argument	Type	Variable name	Description	I/O
	uint8_t	ucInt	Numeric string	Input
Return value	uint8_t: ASCII code			
Description	This function converts numeric string to ASCII.			

5.6.5 local_atoi

Table 16 local_atoi

Function	Conversion from ASCII to numeric string			
File name	SLMP.c		Disclosed/undisclosed	Disclosed
Call format	uint8_t local_atoi (uint8_t ucInt)			
Argument	Type	Variable name	Description	I/O
	uint8_t	ucInt	Numeric string	Input
Return value	uint8_t: Numeric string			
Description	This function converts ASCII to numeric string.			

5.6.6 SLMP_MakeErrorData

Table 17 SLMP_MakeErrorData

Function	SLMP error response data generation			
File name	SLMP.c		Disclosed/undisclosed	Disclosed
Call format	int SLMP_MakeErrorData (const SLMP_INFO *p, uint8_t *pucStream , uint16_t *pusDataSize)			
Argument	Type	Variable name	Description	I/O
	const SLMP_INFO *	p	SLMP information	Input
	uint8_t *	pucStream	Response data	Output
	uint16_t *	pusDataSize	Response data size	Output
Return value	SLMP_ERR_OK: Normal SLMP_ERR_NG: Error			
Description	This function generates SLMP error response data.			

5.6.7 ccief_basic_slave_initialize

Table 18 ccief_basic_slave_initialize

Function	CCIEF-BASIC slave station initialization			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	int ccief_basic_slave_initialize (CCIEF_BASIC_SLAVE_INFO *pSlave, CCIEF_BASIC_SLAVE_CALLBACK_RECV_CYCLIC_DATA pRecvCyclicDataFunc, CCIEF_BASIC_SLAVE_CALLBACK_CYCLIC_DISCONNECTION pCyclicDisconnectionFunc)			
Argument	Type	Variable name	Description	I/O
	CCIEF_BASIC_SLAVE_INFO *	pSlave	Slave station information	Input
	CCIEF_BASIC_SLAVE_CALLBACK _RECV_CYCLIC_DATA	pRecvCyclicData Func	Callback function (Cyclic data reception)	Input
	CCIEF_BASIC_SLAVE_CALLBACK _CYCLIC_DISCONNECTION	pCyclicDisconnection Func	Callback function (Disconnection detection)	Input
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal SOCKET_ERR_SOCKET: Socket error			
Description	This function: - initializes the CCIEF-BASIC slave station; - initializes each variables; and - generates a socket.			

The following shows the configuration of CCIEF_BASIC_SLAVE_INFO based on the sample code.

[CCIEF_BASIC_SLAVE.h]

```
typedef struct
{
        uint16_t    usVenderCode;               /* Vender code */
        uint32_t    ulModelCode;                /* Model code */
        uint16_t    usMachineVersion;           /* Machine version */
        uint32_t    ulIpAddress;                /* Slave ip address */
        int         iOccupiedStationNumber;     /* Number of occupied stations */
} CCIEF_BASIC_SLAVE_INFO;
```

5.6.8 ccief_basic_slave_terminate

Table 19 ccief_basic_slave_terminate

Function	CCIEF-BASIC slave station termination			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	void ccief_basic_slave_terminate (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function: - terminates the CCIEF-BASIC slave station; and - closes the socket.			

5.6.9 ccief_basic_slave_main

Table 20 ccief_basic_slave_main

Function	CCIEF-BASIC slave station main processing			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	int ccief_basic_slave_main (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal CCIEF_BASIC_SLAVE_ERR_NG: Error SOCKET_ERR_RECV: Socket error			
Description	This function: - receives a packet; and - checks the disconnection detection period.			

5.6.10 ccief_basic_slave_set_rx

Table 21 ccief_basic_slave_set_rx

Function	RX data setting			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	int ccief_basic_slave_set_rx (int iNumber, int iValue)			
Argument	Type	Variable name	Description	I/O
	int	iNumber	Device number	Input
	int	iValue	Set value 0 (bit off) 1 (bit on)	Input
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE: Device number error			
Description	This function sets an argument iValue in the RX of the device number specified by the argument iNumber.			

5.6.11 ccief_basic_slave_get_ry

Table 22 ccief_basic_slave_get_ry

Function	RY data acquisition			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	int ccief_basic_slave_get_ry (int iNumber, int *piValue)			
Argument	Type	Variable name	Description	I/O
	int	iNumber	Device number	Input
	int *	piValue	Data storage location pointer Stored value: 0 (bit off) 1 (bit on)	Output
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE: Device number error			
Description	This function acquires RY data of the device number specified by the argument iNumber.			

5.6.12 ccief_basic_slave_get_rww

Table 23 ccief_basic_slave_get_rww

Function	RWw data acquisition			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	int ccief_basic_slave_get_rww (int iNumber, uint16_t *pusValue)			
Argument	Type	Variable name	Description	I/O
	int	iNumber	Device number	Input
	uint16_t *	pusValue	Data storage location pointer	Output
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE: Device number error			
Description	This functions acquires RWw data of the device number specified by the argument iNumber.			

5.6.13 ccief_basic_slave_set_rwr

Table 24 ccief_basic_slave_set_rwr

Function	RWr data setting			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	int ccief_basic_slave_set_rwr (int iNumber, uint16_t usValue)			
Argument	Type	Variable name	Description	I/O
	int	iNumber	Device number	Input
	uint16_t	usValue	Set value	Input
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Norma CCIEF_BASIC_SLAVE_ERR_DEVICE_RANGE: Device number error			
Description	This function sets an argument iValue in the RWr of the device number specified by the argument iNumber.			

5.6.14 ccief_basic_slave_get_pointer

Table 25 ccief_basic_slave_get_pointer

Function	Device head pointer acquisition			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	uint16_t *ccief_basic_slave_get_pointer (int iDeviceType)			
Argument	Type	Variable name	Description	I/O
	int	iDeviceType	Device type	Input
Return value	Device head pointer			
Description	This function acquires a head pointer of the device.			

The following shows the definition of the device types based on the sample code.

[CCIEF_BASIC_SLAVE.h]

```
#define    CCIEF_BASIC_DEVICE_TYPE_RX        1        /* Type of device for RX */
#define    CCIEF_BASIC_DEVICE_TYPE_RY        2        /* Type of device for RY */
#define    CCIEF_BASIC_DEVICE_TYPE_RWW       3        /* Type of device for RWw */
#define    CCIEF_BASIC_DEVICE_TYPE_RWR       4        /* Type of device for RWr */
```

5.6.15 ccief_basic_slave_set_unit_info

Table 26 ccief_basic_slave_set_unit_info

Function	Own station unit information setting			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	void ccief_basic_slave_set_unit_info (uint16_t usUnitInfo)			
Argument	Type	Variable name	Description	I/O
	uint16_t	usUnitInfo	Unit information	Input
Return value	-			
Description	This function sets the own station unit information.			

The following shows the definition of the application operating status based on the sample code.

[CCIEF_BASIC_SLAVE.h]

```
#define CCIEF_BASIC_UNIT_INFO_APPLICATION_STOP        0x0000   /* Stopping application for setting the unit info */
#define CCIEF_BASIC_UNIT_INFO_APPLICATION _RUNNING    0x0001   /* Running application for setting the unit info */
```

5.6.16 ccief_basic_slave_set_err_code

Table 27 ccief_basic_slave_set_err_code

Function	Error code setting			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	void ccief_basic_slave_set_err_code (uint16_t usErrCode)			
Argument	Type	Variable name	Description	I/O
	uint16_t	usErrCode	Error code	Input
Return value	-			
Description	This function sets error codes.			

5.6.17 ccief_basic_slave_set_unit_data

Table 28 ccief_basic_slave_set_unit_data

Function	Own station management information setting			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	void ccief_basic_slave_set_unit_data (uint32_t ulUnitData)			
Argument	Type	Variable name	Description	I/O
	uint32_t	ulUnitData	Own station management information	Input
Return value	-			
Description	This function sets the own station management information.			

5.6.18 ccief_basic_slave_get_master_info

Table 29 ccief_basic_slave_get_master_info

Function	Master station information acquisition			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Disclosed
Call format	void ccief_basic_slave_get_master_info (CCIEF_BASIC_SLAVE_MASTER_INFO *pInfo)			
Argument	Type	Variable name	Description	I/O
	CCIEF_BASIC_SLAVE_MASTER_INFO *	pInfo	Storage location pointer for the master station information	Output
Return value	-			
Description	This function acquires the master station information.			

The following shows the configuration of CCIEF_BASIC_SLAVE_MASTER_INFO based on the sample code.

[CCIEF_BASIC_SLAVE.h]

```
typedef struct
{
        uint16_t                        usUnitInfo;         /* Information of the unit */
        uint16_t                        usReserve;          /* Reserve */
        uint8_t                         aucTimeData[8];     /* Time of the master */
} CCIEF_BASIC_MASTER_NOTIFY_INFO;

typedef struct
{
        uint32_t                        ulId;               /* Id of the master */
        uint8_t                         ucGroupNumber;      /* Group number */
        CCIEF_BASIC_MASTER_NOTIFY_INFO  NotifyInfo;         /* Notify information of the master */
} CCIEF_BASIC_SLAVE_MASTER_INFO;
```

5.6.19 ccief_basic_slave_recv_cyclic_data

Table 30 ccief_basic_slave_recv_cyclic_data

Function	Cyclic data receiving			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Undisclosed
Call format	int ccief_basic_slave_recv_cyclic_data (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - analyzes the receive data; - acquires the master station notification information; and - forwards the received data to RY and RWw.			

5.6.20 ccief_basic_slave_send_cyclic_data

Table 31 ccief_basic_slave_send_cyclic_data

Function	Cyclic data sending			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Undisclosed
Call format	int ccief_basic_slave_send_cyclic_data (SOCKET sock, const SLMP_INFO *source, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulSendAddr	Send destination IP address	Input
	uint16_t	usSendPortNumber	Send destination port number	Input
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - forwards RX and RWr to send data; and - sends the cyclic data to the master station.			

5.6.21 ccief_basic_slave_send_cyclic_data_error

Table 32 ccief_basic_slave_send_cyclic_data_error

Function	Cyclic error data sending			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Undisclosed
Call format	int ccief_basic_slave_send_cyclic_data_error (SOCKET sock, const SLMP_INFO *source, uint16_t usEndCode, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint16_t	usEndCode	End code	Input
	uint32_t	ulSendAddr	Send destination IP address	Input
	uint16_t	usSendPortNumber	Send destination port number	Input
Return value	CCIEF_BASIC_SLAVE_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function sends cyclic error data to the master station.			

5.6.22 ccief_basic_slave_disconnection

Table 33 ccief_basic_slave_disconnection

Function	Disconnection processing			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Undisclosed
Call format	void ccief_basic_slave_disconnection (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function: - executes the disconnection processing of the cyclic transmission; and - executes the callback function (disconnection detection) specified by the user.			

5.6.23 ccief_basic_slave_disconnection_timer_timeout

Table 34 ccief_basic_slave_disconnection_timer_timeout

Function	Timeout of disconnection detection period (callback function)			
File name	CCIEF_BASIC_SLAVE.c		Disclosed/undisclosed	Undisclosed
Call format	void ccief_basic_slave_disconnection_timer_timeout (int iId, void *pCallbackArg)			
Argument	Type	Variable name	Description	I/O
	int	iId	Timer ID	Input
	void *	pCallbackArg	Callback function argument (Not used)	Input
Return value	-			
Description	This callback function is executed by the timer function when the disconnection detection period of the cyclic transmission has timed out. This function executes disconnection processing of the cyclic transmission.			

5.6.24 slmp_server_initialize

Table 35 slmp_server_initialize

Function	SLMP server initialization			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_initialize (SLMP_SERVER_INFO *pServerInfo, SLMP_SERVER_CALLBACK_IPADDRESS_SET_BASIC pIpAddresSetFunc, SLMP_SERVER_CALLBACK_PARAMETER_GET pParameterGetFunc, SLMP_SERVER_CALLBACK_PARAMETER_SET pParameterSetFunc, SLMP_SERVER_CALLBACK_PARAMETER_SET_END pParameterSetEndFunc, SLMP_SERVER_CALLBACK_REMOTE_RESET pRemoteResetFunc)			
Argument	Type	Variable name	Description	I/O
	SLMP_SERVER_INFO *	pServerInfo	Server information	Input
	SLMP_SERVER_CALLBACK_ IPADDRESS_SET_BASIC	pIpAddresSetFunc	Callback function (Communication settings)	Input
	SLMP_SERVER_CALLBACK_ PARAMETER_GET	pParameterGetFunc	Callback function (Parameter reading)	Input
	SLMP_SERVER_CALLBACK_ PARAMETER_SET	pParameterSetFunc	Callback function (Parameter writing)	Input
	SLMP_SERVER_CALLBACK_ PARAMETER_SET_END	pParameterSetEndFunc	Callback function (Parameter writing completed)	Input
	SLMP_SERVER_CALLBACK_ REMOTE_RESET	pRemoteResetFunc	Callback function (Unit reset)	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SOCKET: Socket error			
Description	This function: - initialize the SLMP server; - initializes each variables; and - generates a socket.			

The following shows the configuration of SLMP_SERVER_INFO based on the sample code.

[SLMP_SERVER.h]

```
typedef struct
{
        uint16_t        usVenderCode;               /* Vender code */
        uint32_t        ulModelCode;                /* Model code */
        uint16_t        usMachineVersion;           /* Machine version */
        uint8_t         aucMacAddress[6];           /* Mac Address */
        uint32_t        ulIpAddress;                /* Server ip address */
        uint32_t        ulSubnetMask;               /* Server subnet mask */
        uint32_t        ulDefaultGatewayIPAddress;  /* Server default gateway ip address */
        uint16_t        usPortNumber;               /* Server port number */
        uint8_t         acHostname[64];             /* Hostname */
        uint16_t        usStatus;                   /* Status */
        uint8_t         acTypeName[16];             /* Type name */
        uint16_t        usTypeNameCode;             /* Type name code */
        uint16_t        *pusMemory;                 /* Pointer of the user memory */
        unsigned int    uiMemorySize;               /* Size of the user memory */
} SLMP_SERVER_INFO;
```

5.6.25 slmp_server_terminate

Table 36 slmp_server_terminate

Function	SLMP server termination			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	void slmp_server_terminate (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function: - terminates SLMP server and - closes the socket.			

5.6.26 slmp_server_main

Table 37 slmp_server_main

Function	SLMP server main processing			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_main (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_RECV: Socket error			
Description	This function calls the processing for each socket.			

5.6.27 slmp_server_user_port

Table 38 slmp_server_user_port

Function	Receive processing for the user-specified port of SLMP server			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_user_port (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_RECV: Socket error			
Description	For the packet received via the user-specified port, this function: - receives a request packet; - executes each service processing; and - sends a response packet.			

5.6.28 slmp_server_basic_port

Table 39 slmp_server_basic_port

Function	Receive processing for the CCIEF-BASIC NodeConnect port of SLMP server			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_basic_port (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_RECV: Socket error			
Description	For packets received via the CCIEF-BASIC NodeConnect port (61451), this function: - receives a request packet; - executes each service processing; and - sends a response packet.			

5.6.29 slmp_server_paramset_port

Table 40 slmp_server_paramset_port

Function	Receive processing for the parameter setting port of SLMP server			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_paramset_port (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_RECV: Socket error			
Description	For packets received via the parameter setting port (45237), this function: - receives a request packet; - executes each service processing; and - sends a response packet.			

5.6.30 slmp_server_set_status

Table 41 slmp_server_set_status

Function	SLMP server status setting			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	void slmp_server_set_status (uint16_t usStatus)			
Argument	Type	Variable name	Description	I/O
	uint16_t	usStatus	Server status	Input
Return value	-			
Description	This function sets the server status specified by the argument usStatus.			

5.6.31 slmp_server_slmp_send_response

Table 42 slmp_server_slmp_send_response

Function	SLMP response sending			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Undisclosed
Call format	int slmp_server_slmp_send_response (SOCKET sock, const SLMP_INFO *source, uint32_t ulSendAddr, uint16_t usSendPortNumber, uint8_t *pucSendData, uint16_t usSendDataSize)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulSendAddr	Send destination IP address	Input
	uint16_t	usSendPortNumber	Send destination port number	Input
	uint8_t *	pucSendData	Storage location pointer for send data	Input
	uint16_t	usSendDataSize	Send data size	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function sends SLMP response.			

5.6.32 slmp_server_slmp_send_err_response

Table 43 slmp_server_slmp_send_err_response

Function	SLMP error response sending			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Undisclosed
Call format	int slmp_server_slmp_send_err_response (SOCKET sock, const SLMP_INFO *source, uint16_t usEndCode, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint16_t	usEndCode	End code	Input
	uint32_t	ulSendAddr	Send destination IP address	Input
	uint16_t	usSendPortNumber	Send destination port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function sends SLMP error response.			

5.6.33 slmp_server_service

Table 44 slmp_server_service

Function	SLMP service execution			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_service (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber, SLMP_SERVICE *pServiceTable, int iServiceTableNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
	SLMP_SERVICE *	pServiceTable	SLMP service table	Input
	int	iServiceTableNumber	Number of SLMP services	Input
Return value	SLMP_SERVER_ERR_OK: Normal SLMP_SERVER_ERR_UNSUPPORT_SERVICE: Unsupported service error SOCKET_ERR_SEND: Socket error			
Description	This function executes SLMP service.			

The following shows the configuration of SLMP_SERVICE based on the sample code.

[SLMP_SERVER.h]

```
typedef struct
{
        uint16_t    usCommand;                                                          /* Command */
        int         (*pFunc)( SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber );
                                                                                        /* Function */
} SLMP_SERVICE;
```

5.6.34 slmp_server_memory_read

Table 45 slmp_server_memory_read

Function	Memory reading			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_memory_read (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for memory reading; and - creates and sends response data.			

5.6.35 slmp_server_memory_write

Table 46 slmp_server_memory_write

Function	Memory writing			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_memory_write (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for memory writing; and - creates and sends response data.			

5.6.36 slmp_server_node_search_basic

Table 47 slmp_server_node_search_basic

Function	Automatic detection (for CCIEF-BASIC)			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_node_search_basic (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for automatic detection; - creates response data; - acquires the waiting time for broadcast send; and - starts the timer for waiting for sending of the automatic detection response.			

The following table lists the setting values for the automatic detection response data.

Table 48 Automatic Detection Response Data

No.	Item	Setting value	Remarks
1	Server MAC address	Default value	Setting value of network adapters
2	Server IP address	Default value	Setting value of network adapters
3	Server subnet mask	Default value	Setting value of network adapters
4	Server default gateway IP address	Default value	Setting value of network adapters
5	Server host name	"SlaveSample"	The value defined with "USER_SERVER_HOSTNAME"*1
6	Server vendor code	0x1234	The value defined with "USER_PROFILE_VENDOR_CODE"*1
7	Server model code	0x00010001	The value defined with "USER_ PROFILE_MODEL_CODE"*1
8	Server device version	0x0001	The value defined with "USER_ PROFILE_MACHINE_VERSION"*1
9	IP address of the communication destination unit	0xFFFFFFFF	Fixed value
10	Port number of the communication destination unit	0xFFFF	Fixed value
11	Server status	0x0000 (Initial value)	This value can be changed in the program.*2
12	Server communication port number	61451	Fixed value (port number for CCIEF-BASIC automatic detection)
13	Server communication protocol setting	1	Fixed value (UDP)

*1 Defined in USER_SAMPLE.h.
*2 Set value of slmp_server_set_status

5.6.37 slmp_server_node_search_send_response_timeout

Table 49 slmp_server_node_search_send_response_timeout

Function	Timeout to wait for automatic detection response sending (callback function)			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Undisclosed
Call format	void slmp_server_node_search_send_response_timeout (int iId, void *pCallbackArg)			
Argument	Type	Variable name	Description	I/O
	int	iId	Timer ID (Not used)	Input
	void *	pCallbackArg	Callback function argument	Input
Return value	-			
Description	This callback function is executed by the timer function at the timeout to wait for automatic detection response sending. This function sends response data.			

5.6.38 slmp_server_ip_address_set_basic

Table 50 slmp_server_ip_address_set_basic

Function	Communication setting (for CCIEF-BASIC)			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_ip_address_set_basic (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for communication setting; - executes the callback function (communication setting) specified by the user; and - creates and sends response data.			

5.6.39 slmp_server_device_info_compare

Table 51 slmp_server_device_info_compare

Function	Device connection information check			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_device_info_compare (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for checking the device connection information; and - creates and sends response data.			

5.6.40 slmp_server_parameter_get

Table 52 slmp_server_parameter_get

Function	Parameter reading			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_parameter_get (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for reading parameters; - executes the callback function (parameter reading) specified by the user; and - creates and sends response data.			

5.6.41 slmp_server_parameter_set

Table 53 slmp_server_parameter_set

Function	Parameter writing			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_parameter_set (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for writing parameters; - executes the callback function (parameter writing) specified by the user; and - creates and sends response data.			

5.6.42 slmp_server_parameter_set_start

Table 54 slmp_server_parameter_set_start

Function	Start of parameter write exclusive processing			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_parameter_set_start (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data to start the parameter write exclusive processing; and - creates and sends response data.			

5.6.43 slmp_server_parameter_set_end

Table 55 slmp_server_parameter_set_end

Function	Termination of parameter write exclusive processing			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_parameter_set_end (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data to terminate the parameter write exclusive processing; - executes the callback function (parameter writing completed) specified by the user; and - creates and sends response data.			

5.6.44 slmp_server_parameter_set_cancel

Table 56 slmp_server_parameter_set_cancel

Function	Cancel of parameter write exclusive processing			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_parameter_set_cancel (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data to cancel the parameter write exclusive processing; and - creates and sends response data.			

5.6.45 slmp_server_communication_setting_get

Table 57 slmp_server_communication_setting_get

Function	Communication settings acquisition			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_communication_setting_get (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data to acquire the communication settings; and - creates and sends response data.			

5.6.46 slmp_server_read_type_name

Table 58 slmp_server_read_type_name

Function	Model name reading			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_read_type_name (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data to read the model name; and - creates and sends response data.			

The following table lists the setting value of the response data for reading the model name.

Table 59 Response Data for Reading the Model Name

No.	Item	Setting value	Remarks
1	Model name	"SampleCode"	Defined with "USER_TYPE_NAME".*1
2	Model code	0x1234	Defined with "USER_TYPE_NAME_CODE".*1

*1 Defined in USER_SAMPLE.h.

5.6.47 slmp_server_remote_reset

Table 60 slmp_server_remote_reset

Function	Remote reset			
File name	SLMP_SERVER.c		Disclosed/undisclosed	Disclosed
Call format	int slmp_server_remote_reset (SOCKET sock, const SLMP_INFO *source, uint32_t ulRecvAddr, uint16_t usRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	const SLMP_INFO *	source	Receive packet information	Input
	uint32_t	ulRecvAddr	Send source IP address	Input
	uint16_t	usRecvPortNumber	Send source port number	Input
Return value	SLMP_SERVER_ERR_OK: Normal SOCKET_ERR_SEND: Socket error			
Description	This function: - receives request data for remote reset; - executes the callback function (remote reset) specified by the user; and - creates and sends response data.			

5.6.48 socket_initialize

Table 61 socket_initialize

Function	Socket initialization			
File name	SOCKET.c		Disclosed/undisclosed	Disclosed
Call format	int socket_initialize (SOCKET *sock, uint32_t ulIpAddress, uint16_t usPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET *	sock	Storage location pointer for the socket descriptor	Output
	uint32_t	ulIpAddress	IP address	Input
	uint16_t	usPortNumber	Port number	Input
Return value	SOCKET_ERR_OK: Normal SOCKET_ERR_SOCKET: Socket generation error			
Description	This function: - initializes the socket; and - returns the socket descriptor. Change the program according to the implementation environment.			

5.6.49 socket_terminate

Table 62 socket_terminate

Function	Socket termination			
File name	SOCKET.c		Disclosed/undisclosed	Disclosed
Call format	void socket_terminate (SOCKET sock)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
Return value	-			
Description	This function terminates the socket. Change the program according to the implementation environment.			

5.6.50 socket_recv

Table 63 socket_recv

Function	Packet receiving			
File name	SOCKET.c		Disclosed/undisclosed	Disclosed
Call format	int socket_recv (SOCKET sock, uint8_t *pucStream, int iLength, uint32_t *pulRecvAddr, uint16_t *pusRecvPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	uint8_t *	pucStream	Receive packet	Output
	int	iLength	Receive packet length	Input
	uint32_t *	pulRecvAddr	Send source IP address	Output
	uint16_t *	pusRecvPortNumber	Send source port number	Output
Return value	SOCKET_ERR_OK: Normal SOCKET_ERR_NO_RECEIVABLE: No receive data SOCKET_ERR_RECV: Socket receive error			
Description	This function receives a packet. Change the program according to the implementation environment. * This function must be regularly executed.			

5.6.51 socket_send

Table 64 socket_send

Function	Packet sending			
File name	SOCKET.c		Disclosed/undisclosed	Disclosed
Call format	int socket_send (SOCKET sock, uint8_t *pucStream, int iLength, uint32_t ulSendAddr, uint16_t usSendPortNumber)			
Argument	Type	Variable name	Description	I/O
	SOCKET	sock	Socket descriptor	Input
	uint8_t *	pucStream	Send packet	Input
	int	iLength	Send packet length	Input
	uint32_t	ulSendAddr	Send destination IP address	Input
	uint16_t	usSendPortNumber	Send destination port number	Input
Return value	SOCKET_ERR_OK: Normal SOCKET_ERR_SEND: Socket send error			
Description	This function sends a packet. Change the program according to the implementation environment.			

5.6.52 timer_initialize

Table 65 timer_initialize

Function	Timer initialization			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	void timer_initialize (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function initializes the timer function. * The maximum number of the timer is defined with "TIMER_MAX".			

5.6.53 timer_terminate

Table 66 timer_terminate

Function	Timer termination			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	void timer_terminate (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function terminates the timer function.			

5.6.54 timer_main

Table 67 timer_main

Function	Timer main processing			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	void timer_main (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function executes timer main processing. When the time is up, this function executes the callback function specified by the user. * This function must be regularly executed.			

5.6.55 timer_start

Table 68 timer_start

Function	Timer start			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	int timer_start (uint32_t ulTime, int *piId, TIMER_CALLBACK pCallbackFunc, void *pCallbackArg)			
Argument	Type	Variable name	Description	I/O
	uint32_t	ulTime	Timeout period [ms]	Input
	int *	piId	Storage location pointer for the timer ID	Output
	TIMER_CALLBACK	pCallbackFunc	Storage source pointer for the callback function	Input
	void *	pCallbackArg	Argument of the callback function	Input
Return value	TIMER_OK: Normal TIMER_RESOURCE_NONE: Timer exhaustion			
Description	This function: - starts the timer; - sets the started timer ID to the storage location specified by the argument piId; - registers the callback function specified by the argument pCallbackFunc; and - registers the callback argument specified by the argument pCallbackArg.			

5.6.56 timer_stop

Table 69 timer_stop

Function	Timer stop			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	void timer_main (int iId)			
Argument	Type	Variable name	Description	I/O
	int	iId	Timer ID	Input
Return value	-			
Description	This function stops the timer specified by the argument piId.			

5.6.57 timer_get_time

Table 70 timer_get_time

Function	Current time acquisition			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	uint32_t timer_get_time (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	Elapsed time (processor time) [ms]			
Description	This function returns the elapsed time (processor time). Change the program according to the implementation environment.			

5.6.58 timer_broadcast_send_wait_time

Table 71 timer_broadcast_send_wait_time

Function	Broadcast send waiting time acquisition			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	uint32_t timer_broadcast_send_wait_time (uint32_t ulMaxWaitTime)			
Argument	Type	Variable name	Description	I/O
	uint32_t	ulMaxWaitTime	Maximum value for the response waiting time	Input
Return value	Send waiting time [ms]			
Description	This function acquires the waiting time for broadcast send.			

5.6.59 timer_analyze_time_data

Table 72 timer_analyze_time_data

Function	Clock time analysis			
File name	TIMER.c		Disclosed/undisclosed	Disclosed
Call format	void timer_analyze_time_data (int64_t llTime, TIMER_TIME_DATA *pTimeData)			
Argument	Type	Variable name	Description	I/O
	int64_t	llTime	Clock information (UNIX time)	Input
	TIMER_TIME_DATA *	pTimeData	Storage location pointer for clock data	Output
Return value	-			
Description	This function: -analyzes the time of the argument llTime. -store in argument pTimeData			

The following shows the configuration of the TIMER_TIME_DATA based on the sample code.

[TIMER.h]

```
typedef struct
{
        uint16_t  usYear;          /* Year */
        uint16_t  usMonth;         /* Month */
        uint16_t  usDay;           /* Day */
        uint16_t  usHour;          /* Hour */
        uint16_t  usMinute;        /* Minute */
        uint16_t  usSecond;        /* Second */
        uint16_t  usMilliseconds;  /* Milliseconds */
} TIMER_TIME_DATA;
```

5.6.60 main

Table 73 main

Function	Main processing			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	void main (int argc, char *argv[])			
Argument	Type	Variable name	Description	I/O
	int	argc	Total number of command line arguments	Input
	char *	argv[]	Command line argument	Input
Return value	-			
Description	This function: - acquires and sets the network adapter information; - initializes the CCIEF-BASIC slave station and SLMP server; - initializes the timer function; - reads and writes parameters; - executes the main processing for the CCIEF-BASIC slave station and SLMP server (loop processing); - executes the main processing of the timer function (loop processing); and - executes reset processing.			

	Change the program according to the implementation environment.

5.6.61 user_callback_recv_cyclic_data

Table 74 user_callback_recv_cyclic_data

Function	Cyclic data receiving (callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	void user_callback_recv_cyclic_data (int iCyclicState, int iOccupiedStationNumber)			
Argument	Type	Variable name	Description	I/O
	int	iCyclicState	Cyclic transmission status 0 (Invalid cyclic data) 1 (Valid cyclic data)	Input
	int	iOccupiedStationNumber	Number of occupied stations of the receive data	Input
Return value	-			
Description	This function is executed when receiving cyclic data from the CCIEF-BASIC master station. Change the program according to the implementation environment. * In the sample code, this function: - forwards RY and RWw data back to RX and RWr data; - waits for the response delay time; and - sets the own station unit information.			

5.6.62 user_callback_cyclic_disconnection

Table 75 user_callback_cyclic_disconnection

Function	Disconnection detection (Callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	void user_callback_cyclic_disconnection (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function is executed when the cyclic transmission with the CCIEF-BASIC master station is disconnected or stopped. Change the program according to the implementation environment. * In the sample code, this function: - sets the own station unit information.			

5.6.63 user_callback_set_ip_address_basic

Table 76 user_callback_set_ip_address_basic

Function	Communication setting (for CCIEF-BASIC) (callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	void user_callback_set_ip_address_basic (uint32_t ulIpAddress, uint32_t ulSubnetMask)			
Argument	Type	Variable name	Description	I/O
	uint32_t	ulIpAddress	IP address	Input
	uint32_t	ulSubnetMask	Subnet mask	Input
Return value	-			
Description	This function is executed when receiving request data for the SLMP communication setting command (0E31). Change the program according to the implementation environment. * In the sample code, this function: - sets the communication setting requested by the argument in the network adapter setting; and - resets the unit.			

5.6.64 user_callback_parameter_get

Table 77 user_callback_parameter_get

Function	Parameter reading (callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	int user_callback_parameter_get (uint16_t usId, uint16_t *pusSize, uint8_t **ppucData)			
Argument	Type	Variable name	Description	I/O
	uint16_t	usId	Parameter ID	Input
	uint16_t *	pusSize	Storage location pointer for the parameter value size	Output
	uint8_t **	ppucData	Storage location pointer for the parameter value	Output
Return value	SLMP_ERR_OK: Normal SLMP_ERR_NO_EXIST_PARAM_ID: Parameter ID error			
Description	This function is executed when receiving request data for the SLMP parameter reading command (0E33). Change the program according to the implementation environment. * In the sample code, this function: - stores the parameter ID value and size of the CCIEF-BASIC slave station parameter specified by the argument usId in the second and third argument.			

5.6.65 user_callback_parameter_set

Table 78 user_callback_parameter_set

Function	Parameter writing (callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	int user_callback_parameter_set (uint16_t usId, uint16_t usSize, uint8_t *pucData)			
Argument	Type	Variable name	Description	I/O
	uint16_t	usId	Parameter ID	Input
	uint16_t	usSize	Parameter value size	Input
	uint8_t *	pucData	Parameter value	Input
Return value	SLMP_ERR_OK: Normal SLMP_ERR_NO_EXIST_PARAM_ID: Parameter ID error			
Description	This function is executed when receiving request data for the SLMP parameter writing command (0E34). Change the program according to the implementation environment. * In the sample code, this function: - reflects the parameter ID value specified by the argument in the CCIEF-BASIC slave station parameter.			

5.6.66 user_callback_parameter_set_end

Table 79 user_callback_parameter_set_end

Function	Parameter writing completed (callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	int user_callback_parameter_set_end (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	SLMP_ERR_OK: Normal			
Description	This function is executed when receiving request data for the SLMP parameter writing completion command (0E36). Change the program according to the implementation environment. * In the sample code, this function: - writes the parameters of the CCIEF-BASIC slave station into the file.			

5.6.67 user_callback_remote_reset

Table 80 user_callback_remote_reset

Function	Remote reset request(callback function)			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Disclosed
Call format	int user_callback_remote_reset (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	SLMP_ERR_OK: Normal			
Description	This function is executed when receiving request data for the SLMP remote reset command (1006). Change the program according to the implementation environment. * In the sample code, this function: - executes the reset processing.			

5.6.68 user_parameter_file_read

Table 81 user_parameter_file_read

Function	Parameter file reading			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Undisclosed
Call format	int user_parameter_file_read (char *file_path, USER_SLAVE_PARAMETER *pParameter, USER_ADAPTER_INFO *pAdapterInfo)			
Argument	Type	Variable name	Description	I/O
	char *	file_path	File	Input
	USER_SLAVE_PARAMETER *	pParameter	Storage location pointer for CCIEF-BASIC slave station parameters	Output
	USER_ADAPTER_INFO *	pAdapterInfo	Network adapter information	Output
Return value	USER_ERR_OK: Normal USER_ERR_NG: Error			
Description	This function: - reads the file specified by the argument file_path and stores it in the argument pParameter; and - reflect the parameter setting value in the argument pAdapterInfo. Change the program according to the implementation environment.			

The following shows the configuration of the USER_SLAVE_PARAMETER based on the sample code.

[USER_SAMPLE.h]

```
typedef struct
{
        uint32_t        ulIpAddress;                    /* Slave ip address */
        uint32_t        ulSubnetMask;                   /* Subnet Mask */
        uint32_t        ulDefaultGatewayIPAddress;      /* Default Gateway IP Address */
        uint16_t        usOccupiedStationNumber;        /* Number of occupied stations */
        uint32_t        ulCyclicResponseWaitTime;       /* Wait for cyclic response time [ms] (0:Not wait) */
} USER_SLAVE_PARAMETER;
```

The following shows the configuration of the USER_ADAPTER_INFO based on the sample code.

[USER_SAMPLE.h]

```
typedef struct
{
        uint8_t         aucMacAddress[6];               /* MAC Address */
        uint32_t        ulIpAddress;                    /* IP Address */
        uint32_t        ulSubnetMask;                   /* Subnet Mask */
        uint32_t        ulDefaultGatewayIPAddress;      /* Default Gateway IP Address */
} USER_ADAPTER_INFO;
```

5.6.69 user_parameter_file_write

Table 82 user_parameter_file_write

Function	Parameter file writing			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Undisclosed
Call format	int user_parameter_file_write (char *file_path, USER_SLAVE_PARAMETER *pParameter)			
Argument	Type	Variable name	Description	I/O
	char *	file_path	File	Input
	USER_SLAVE_PARAMETER *	pParameter	Storage location pointer for CCIEF-BASIC slave station parameters	Input
Return value	USER_ERR_OK: Normal USER_ERR_NG: Error			
Description	This function writes the value of the argument pParameter into the file specified by the argument file_path. Change the program according to the implementation environment.			

5.6.70 user_display_cyclic_information

Table 83 user_display_cyclic_information

Function	Cyclic information display			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Undisclosed
Call format	void user_display_cyclic_information (void)			
Argument	Type	Variable name	Description	I/O
	-	-	-	-
Return value	-			
Description	This function displays the cyclic information on the screen. Change the program according to the implementation environment.			

5.6.71 user_get_adapter_info

Table 84 user_get_adapter_info

Function	Network adapter information acquisition			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Undisclosed
Call format	int user_get_adapter_info (USER_ADAPTER_INFO *pGetAdapterInfo)			
Argument	Type	Variable name	Description	I/O
	USER_ADAPTER_INFO *	pGetAdapterInfo	Storage location pointer for network adapter information	Output
Return value	USER_ERR_OK: Normal USER_ERR_NG: Error			
Description	This function acquires the network adapter information. Change the program according to the environment. * The sample code describes an example of acquiring the network adapter information on Windows operating system.			

5.6.72 user_set_adapter_info

Table 85 user_set_adapter_info

Function	Network adapter information setting			
File name	USER_SAMPLE.c		Disclosed/undisclosed	Undisclosed
Call format	int user_set_adapter_info (USER_ADAPTER_INFO *pSetAdapterInfo)			
Argument	Type	Variable name	Description	I/O
	USER_ADAPTER_INFO *	pSetAdapterInfo	Network adapter information	Input
Return value	USER_ERR_OK: Normal USER_ERR_NG: Error			
Description	This function sets the network adapter information. Change the program according to the environment. * The sample code describes an example of setting the network adapter information on Windows operating system.			

6 Appendix: Procedure from compilation to execution of sample code

This section describes the procedure from compilation to execution of the sample code when "gcc" is used.
This code is compiled on the CentOS based system.

6.1 Specifications

Execute the sample code under the environment shown in Figure 17.

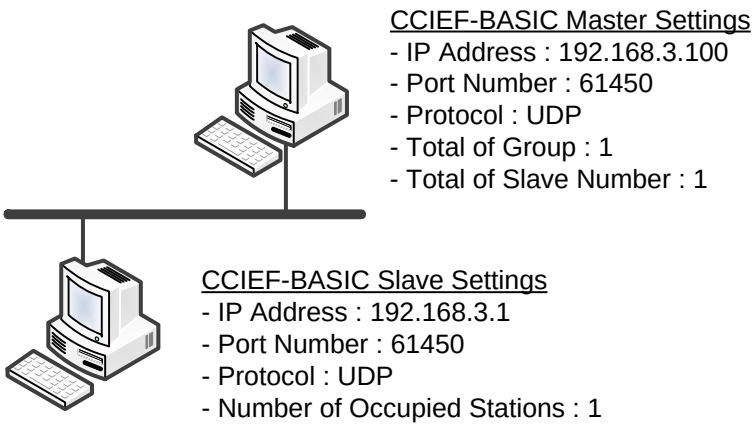

Figure 17 Execution Environment of the Sample Code

The sample code executes the cyclic transmission between the CCIEF-BASIC slave station and the CCIEF-BASIC master station (sample application*1). The status of the cyclic transmission will be displayed on the screen in 5-second intervals.
The command line argument specifies the parameter file set by the user and starts the application.*2*3
If multiple network adapters are mounted in the execution environment, the selection window of the mounted network adapter is displayed. The sample code is started with the network adapter selected by the user.

*1 For details, refer to the "CC-Link IE Field Network Basic Sample Code User's Manual (Master Station)".
*2 If the command line argument does not specify any parameter file, the application is started with the default parameters.
*3 If the IP address of the CCIEF-BASIC slave station set by the parameters is different from the default value of the network adapter, the settings of the network adapter will be changed to the set values of the parameter.

6.2 Creating an application

This section describes the procedure to create an executable module using gcc.

(1) Extract sample code tree.

(2) cd CCIEF-BASIC_Master

(3) Do the following command.

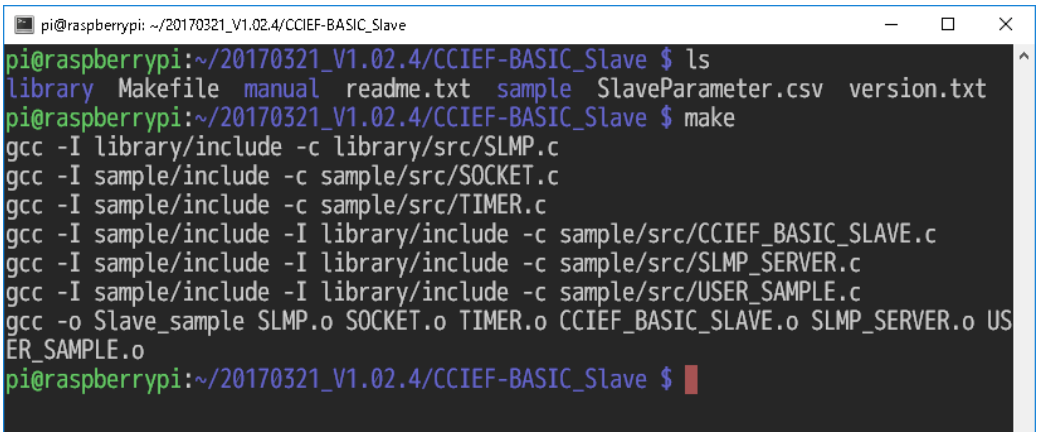

Figure 18 Compile command

6.3 Executing an application

This section describes the procedure to execute an application using gcc.

(1) Create a parameter file of the CCIEF-BASIC slave station. (For details, refer to "4 Specifications".)

[SlaveParameter.csv]

```
,,
CCIEF-BASIC Slave Sample Parameter,,
,,
ID,DATA,COMMENT
1,0,IP Address
2,0,Subnet Mask
3,0,Default Gateway IP Address
4,1,Occupied Station Number
5,0,Cyclic Response Wait Time
```

(2) Execute application as following.

When the cyclic transmission with the CC-Link IE Field Network Basic master station is executed, the output is as follows.

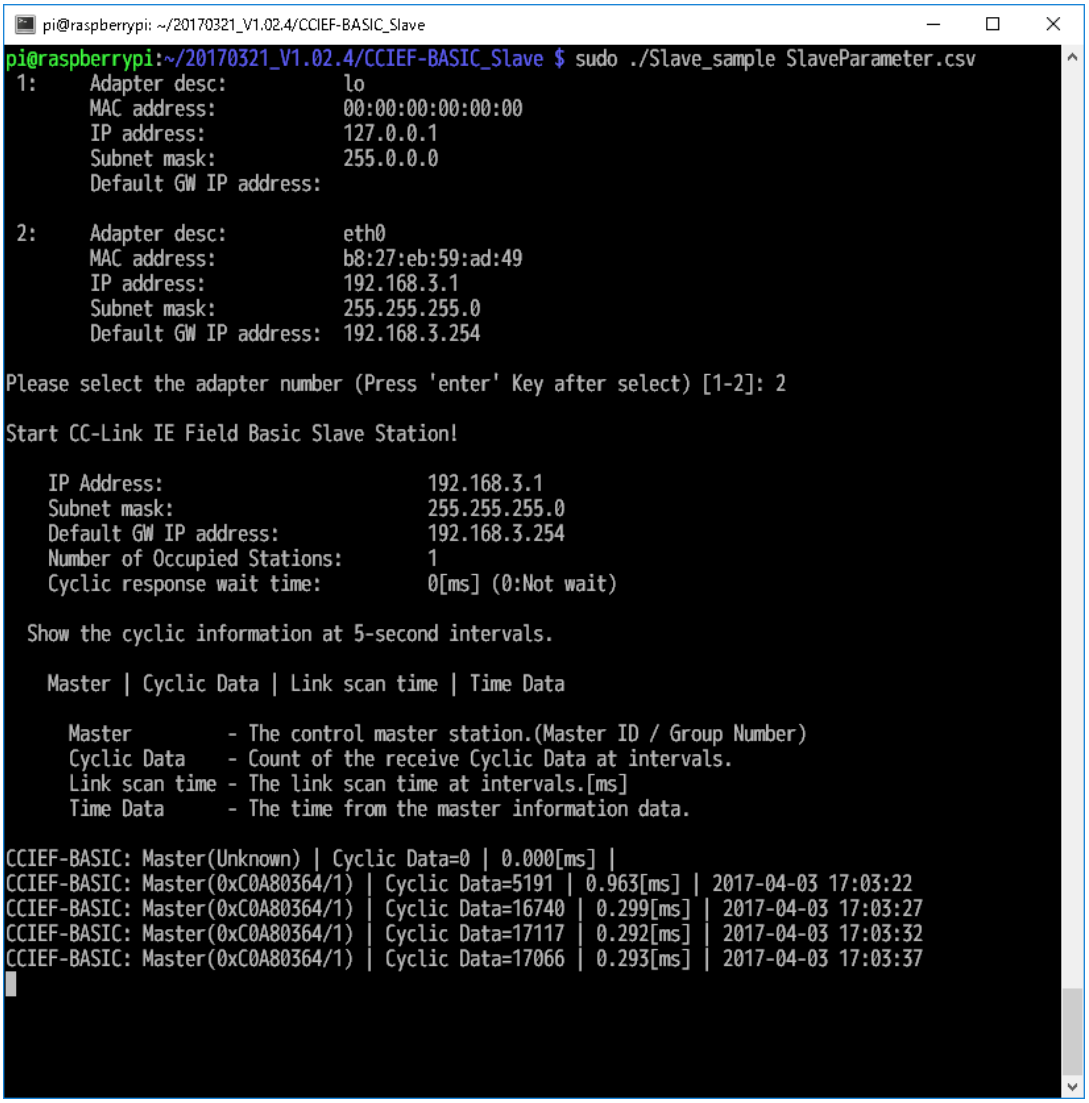

Figure 19 Execute Application