

RL78/G23

Lighting communication master board initial firmware DALI communication modulation by SMS

Introduction

This application note describes the initial firmware for the RL78/G23 lighting communication master board (RTK7RL23LMP00000BJ). Hereinafter, this initial firmware is referred to as "the sample application".

To achieve communication in conformance with the DALI and DMX512 standards, the sample application processes signal waveforms by using RL78/G23 on-board peripherals. For DALI sending, the waveform software processing is alleviated by outputting Manchester code from the SNOOZE mode sequencer.

This application note is intended for those who already have knowledge of the DALI and DMX512 standards. For details about these standards, refer to "6. Reference Documents".

Target Device

RL78/G23 Lighting Communication Master Evaluation Board

When applying the sample program covered in this application note to another microcomputer, modify the program according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Contents

1. Specifications	4
1.1 Standalone Feature	5
1.2 Master Controller GUI Communication Feature	6
2. Operation Confirmatino Conditions	7
3. Description of the Hardware	8
3.1 System Configuration	8
4. Description of the Software	9
4.1 Building the Environment	9
4.1.1 Downloading the Software	9
4.1.2 Installing e ² studio	9
4.1.3 Installing QE for Capacitive Touch V3.1.0	9
4.1.4 Installing a Master Controller GUI	9
4.1.4.1 DALI Master Controller GUI	9
4.1.4.2 DMX512 Master Controller GUI	9
4.1.5 Importing the Sample Application	10
4.1.6 Downloading the SIS Modules	10
4.1.7 Building the Sample Application	10
4.1.8 Hardware Configuration and Connection Example for the Sample Application	11
4.1.8.1 Configuring the Master Board	11
4.1.8.2 Configuring the EZ-0012 Evaluation Board	12
4.1.8.3 Example of Connection for DALI Communication	13
4.1.8.4 Example of Connection for DMX512 Communication	14
4.1.9 Debugging the Sample Application	15
4.2 Operation Procedure	16
4.3 Functional Overview	16
4.3.1 Functional Block Diagram	16
4.3.2 Common Information	17
4.3.2.1 Settings for the Peripherals	17
4.3.2.2 Touch Keys	20
4.3.3 DALI Communication	28
4.3.3.1 Peripheral Settings	28
4.3.3.2 DALI Driver	29
4.3.4 DMX512 Communication	36
4.3.4.1 Peripheral Settings	36
4.3.4.2 DMX512 Send Processing	37
4.3.5 Infrared Communication	38
4.3.5.1 Peripheral Settings	38

- 4.3.5.2 Infrared Send Processing..... 38
- 4.4 Software Configuration..... 41
 - 4.4.1 Folder Structure..... 41
 - 4.4.2 List of Option Byte Settings..... 43
 - 4.4.3 Flowcharts 44
 - 4.4.3.1 Main Processing..... 44
 - 4.4.3.2 DALI communication control processing..... 47
 - 4.4.3.3 DMX512 communication control processing..... 52
- 5. Notes.....55
- 6. Sample Code.....55
- 7. Reference Documents.....55
- Revision History.....57

1. Specifications

The RL78/G23 lighting communication master board (hereinafter, "the master board") supports the interfaces for DALI communication, DMX512 communication, and infrared communication.

The sample application allows the following two features for operation of the master board:

- Standalone feature
Allows the user to control operation (via DALI or infrared communication) by using nine touch keys on the master board.
- Master controller GUI communication feature
Enables communication with a master controller GUI via DALI communication or DMX512 communication.

Table 1-1 shows the communication protocols and the features that support the protocols.

Table 1-1 Communication Protocols and Features

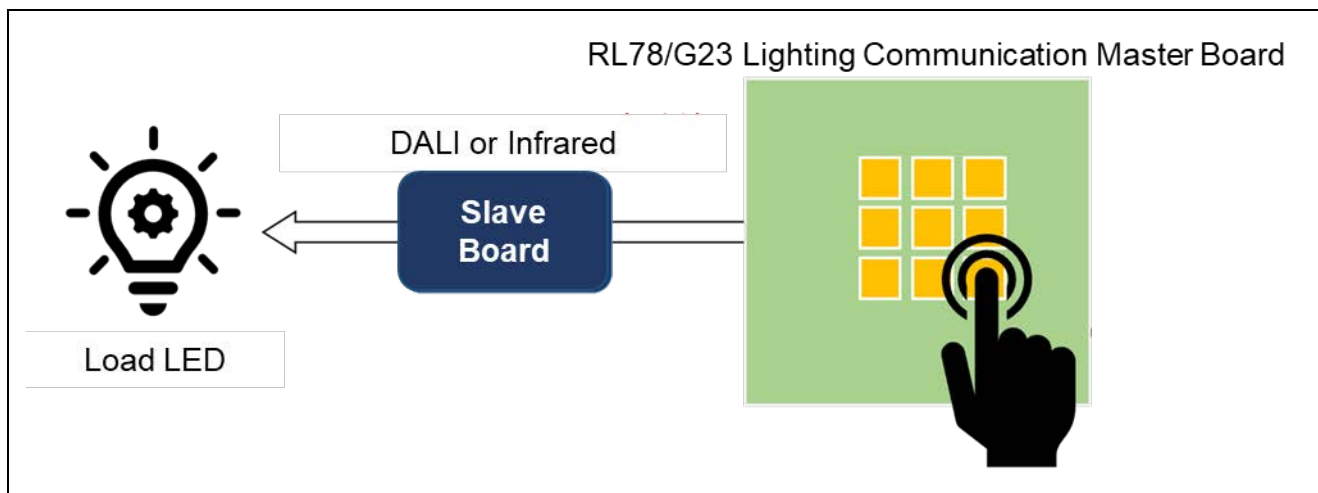
Communication protocol	Feature
DALI	Standalone feature and master controller GUI communication feature
DMX512	Master controller GUI communication feature
Infrared	Standalone feature

In this application note, the word "slave" is used to generically refer to a communications partner connected to the master board, such as a DALI control gear, DMX512 receiver, or infrared receiver.

1.1 Standalone Feature

The standalone feature allows the user to send commands by using the touch keys on the master board.

Figure 1-1 Standalone Feature



By pressing a touch key on the master board, the user can send a command that is pre-assigned ^{Note 1} to the key to the slave board.

Table 1-2 shows the communication protocols available for touch key operation.

Table 1-2 Communication Protocols Available for Touch Key Operation

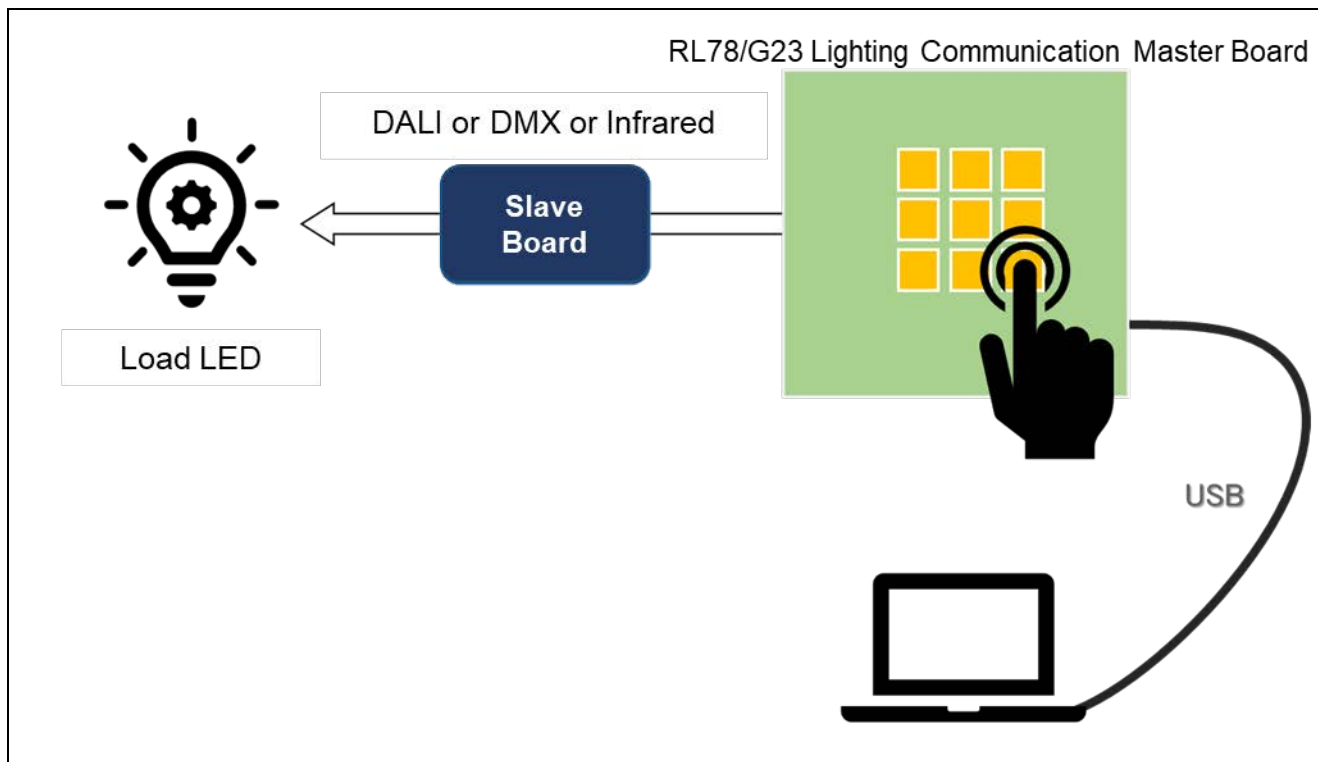
Communication protocol	Availability
DALI	✓
DMX512	-
Infrared	✓

Note 1: Command assignment to touch keys is defined by using constants coded in the sample application. Therefore, it is impossible to change the assignment while the sample application is running. For details about the command assignment to touch keys, refer to sections 4.3.2.2(5), 4.3.2.2(6), and 0.

1.2 Master Controller GUI Communication Feature

The master controller GUI communication feature allows the user to send commands to the slave board by using the GUI on the PC connected to the master board.

Figure 1-2 Master Controller GUI Communication Feature



The master controller GUI communication feature passes messages between the master board and the PC that is USB-connected with the master board via the master controller GUI ^{Note 1} installed on the PC so that the commands corresponding to the messages are sent to the slave board and the responses from the slave board are returned to the PC.

The DALI Master Controller GUI, which supports the standalone feature, can also accept operations instructed from touch keys on the master board.

Table 1-3 shows the communication protocols for which a master controller GUI is available.

Table 1-3 Communication Protocols for Which a Master Controller GUI Is Available

Communication protocol	Availability
DALI	✓
DMX512	✓
Infrared	-

Note 1: There are two master controller GUIs: DALI Master Controller GUI and DMX512 Master Controller GUI.

2. Operation Confirmatino Conditions

The operation of the sample code provided with this application note has been tested under the following conditions.

Table 2-1 Operation Confirmation Conditions

Item	Description
MCU used	RL78/G23 (R7F100GGG)
Operating frequency	High-speed on-chip oscillator clock (fIH): 32MHz
Operating voltage	3.3V
Integrated development environment (e2 studio)	e2 studio 2021-07 from Renesas Electronics Corp.
C compiler (e2 studio)	CC-RL V1.10.0 from Renesas Electronics Corp.
Plugin (e2 studio)	QE for Capacitive Touch V3.1.0 from Renesas Electronics Corp.
Smart configurator	Components: Board Support Packages. (r_bsp) V1.13 Capacitive Sensing Unit driver (r_ctsu) V1.20 SNOOZE mode sequencer V1.0.1 Touch middleware (rm_touch) V1.20
Emulator	E2 emulator Lite
Board used	RTK7RL23LMP00000BJ EZ-0012 and EZ-0012 expansion board ^{Note 1} from Renesas Electronics Corp.
Tools used	DALI Master Controller GUI V2.02 or later DMX512 Master Controller GUI V2.00 or later

Note 1: For details about the EZ-0012 expansion board, contact Renesas Electronics Corporation.

3. Description of the Hardware

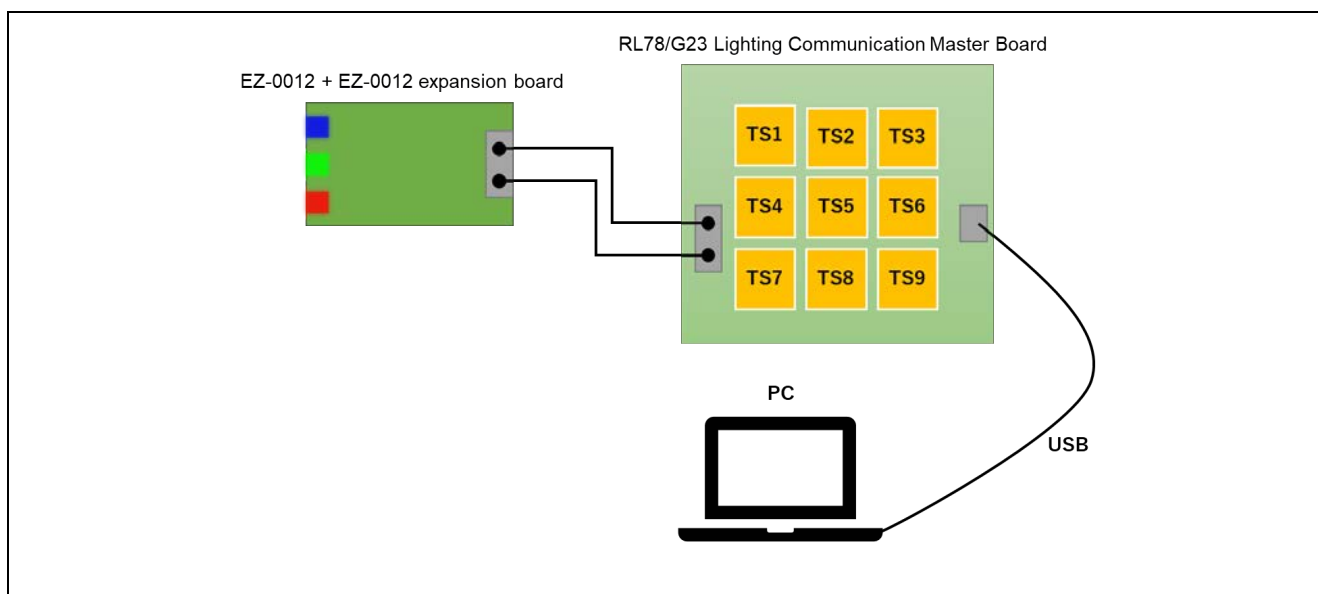
3.1 System Configuration

The following figure shows an example of the system configuration.

In this example, the sample application plays a role as the master that outputs control instruction commands to the slave.

In this application note, the RL78/G23 lighting communication master board is used as the master board and the EZ-0012 + EZ-0012 expansion board (EZ-0012 expansion board on the EZ-0012 evaluation board) is used as the slave board.

Figure 3-1 Example of the System Configuration



- RL78/G23 lighting communication master board

The product information about the RL78/G23 lighting communication master board is available on the website at the following URL:

https://www.renesas.com/rl78g23_lcm_evaluation_board/

(Please search for the product information about the RL78/G23 lighting communication master board.)

For details about how to use the RL78/G23 lighting communication master board, refer to the following manual:

RL78/G23 Lighting Communication Master Board User's Manual (R20UT5072)

- EZ-0012 + EZ-0012 expansion board

For details about the EZ-0012 evaluation board, visit the webpage at the following URL:

<https://www.renesas.com/jp/en/products/microcontrollers-microprocessors/rl78-low-power-8-16-bit-mcus/ez-0012-ez-0012-led-lighting-evaluation-board>

For details about the EZ-0012 expansion board, contact Renesas Electronics Corporation.

4. Description of the Software

4.1 Building the Environment

4.1.1 Downloading the Software

The following software products are required for the sample application to operate and can be downloaded from the Renesas Electronics website.

- e² studio
- QE for Capacitive Touch V3.1.0
- DALI Master Controller GUI version 2.02 or later
- DMX512 Master Controller GUI version 2.00 or later

4.1.2 Installing e² studio

The tool e² studio is required when you develop programs and perform on-chip debugging for the sample application. For details about how to perform installation and other operations, refer to the following manual:

- e² studio 2020-04 and e² studio v7.8 Integrated Development Environment User's Manual (R20UT4819)

4.1.3 Installing QE for Capacitive Touch V3.1.0

QE for Capacitive Touch V3.1.0 is required when you specify the initial settings of the touch interface and tune the sensitivity for the touch keys on the master board.

For details about how to perform installation and other operations, refer to the following manual:

- QE for Capacitive Touch V3.1.0 Release Note (R20UT5144)

4.1.4 Installing a Master Controller GUI

To use the master controller GUI communication feature, a master controller GUI compliant with the communication protocol is required.

4.1.4.1 DALI Master Controller GUI

The DALI Master Controller GUI is required when you evaluate the master controller GUI communication feature used with DALI communication.

For details about the installation procedure, refer to the following manual:

- DALI Master Controller GUI User's Manual (R20UT0715)

4.1.4.2 DMX512 Master Controller GUI

The DMX512 Master Controller GUI is required when you evaluate the master controller GUI communication feature used with DMX512 communication.

For details about the installation procedure, refer to the following manual:

- DMX512 Master Controller GUI User's Manual (R01US0200)

4.1.5 Importing the Sample Application

1. Uncompress the project file provided by this application note, and then move the extracted file to any folder of your choice.
2. Start e² studio and open the Project Explorer tab. Then, right-click and select [Import]. Alternatively, from the [File] menu, select [Import].
3. Open the [General] node, select [Existing Projects into Workspace], and then [Next].
4. Select the [Select root directory] radio button, click [Browse], and then select the project file that you extracted in step 1.
5. Select the sample application to be operated.
6. Click [Finish].

4.1.6 Downloading the SIS Modules

If you generate code for touch keys, you must also download the SIS modules (r_ctorsu and rm_touch).

For details about the download procedure, refer to the following application note:

- "7. Using Smart Configurator to Add Modules" in the manual "RL78 Family: Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)"

4.1.7 Building the Sample Application

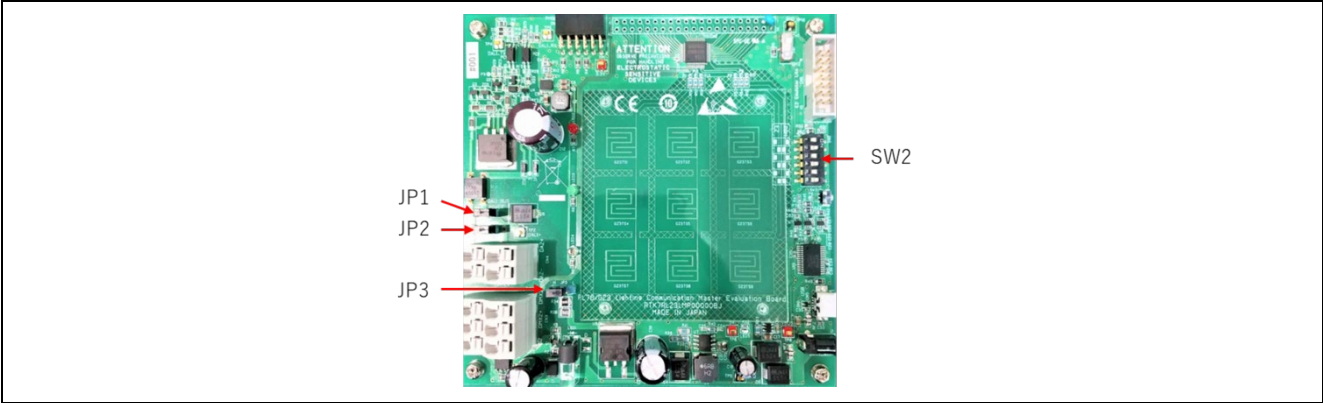
1. In the Project Explorer tab, right-click the target project, and then select [Build Project].
2. Build processing starts and the progress of the processing is displayed in the [Console] tab. When the "Build Finished" message is displayed, the build processing is complete.

4.1.8 Hardware Configuration and Connection Example for the Sample Application

4.1.8.1 Configuring the Master Board

Figure 4-1 shows the exterior view of the master board.

Figure 4-1 RL78/G23 Lighting Communication Master Board



(1) Configuring the operation mode

Configure the mode switch to set RUN mode.

Table 4-1 SW2 Configuration

	SW2					
	1	2	3	4	5	6
RUN mode	OFF	ON	OFF	ON	OFF	ON

(2) Configuring the power supply to the DALI bus line

Configure the power supply to the DALI bus line.

Table 4-2 JP1 and JP2 Configurations

Power supply to the DALI bus line	JP1	JP2
Disable	Open	Open
Enable	Short	Short

(3) Configuring the power source for the master board

Configure the power source for the master board.

Table 4-3 JP3 Configuration

Power source for the master board	JP3
USB port	L
AC adapter	H

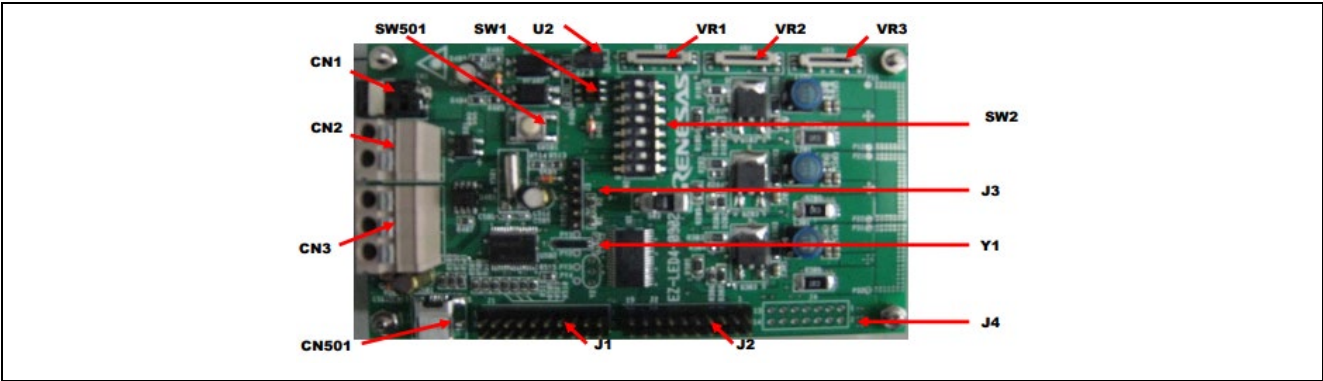
For other information about the master board, refer to the following manual:

- RL78/G23 Lighting Communication Master Board User's Guide (R20UT5072)

4.1.8.2 **Configuring the EZ-0012 Evaluation Board**

Figure 4-2 shows the exterior view of the EZ-0012 evaluation board.

Figure 4-2 RL78/I1A DC/DC LED Control Evaluation Board



(1) Configuring the communication protocol

Select the protocol to be used (DALI or DMX512) with the SW1 configuration switch on the EZ-0012 evaluation board.

To use DALI communication, set the SW1 switch to the DALI position. To use DMX512 communication, set the SW1 switch to the DMX512 position. (Refer to the following figure.)

Figure 4-3 Configuration Switch SW1



(2) Configuring the operation mode

Configure the SW2 configuration switch on the EZ-0012 evaluation board to set normal operation mode (for use of DALI or DMX512 communication).

Table 4-4 SW2 Configuration

	SW2							
	1	2	3	4	5	6	7	8
Normal operation mode	ON	ON	OFF	OFF	OFF	OFF	ON	ON

For other information about the EZ-0012 evaluation board, refer to the following manual:

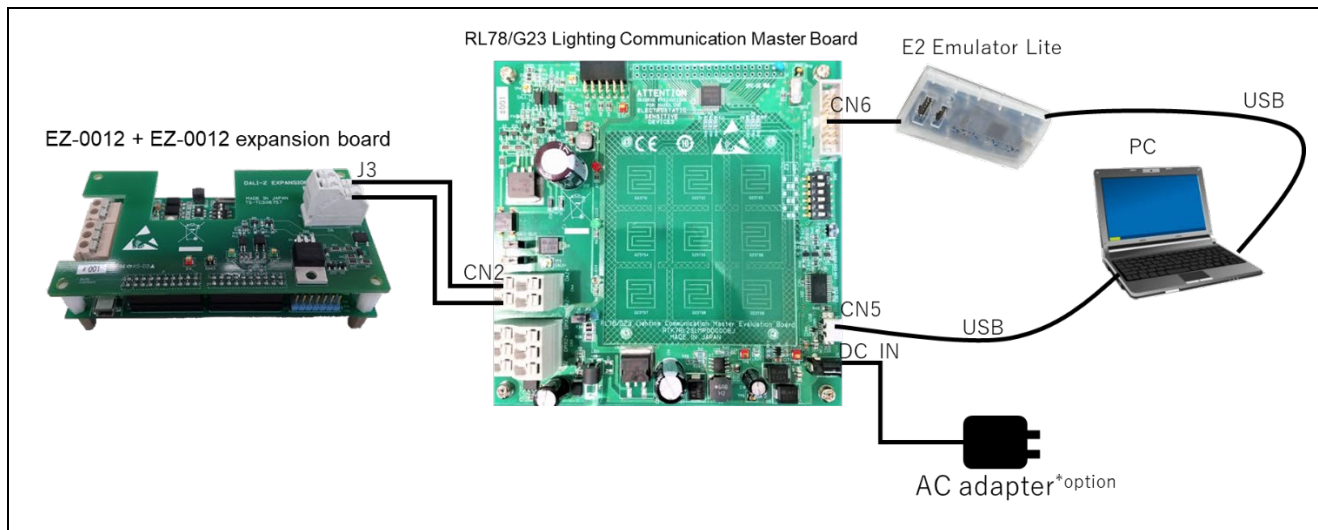
- RL78/I1A DC/DC LED Control Evaluation Board (EZ-0012) User's Manual (R01UH0363)

4.1.8.3 Example of Connection for DALI Communication

Figure 4-4 shows an example of hardware connection for DALI communication.

For the EZ-0012 expansion board, contact Renesas Electronics Corporation.

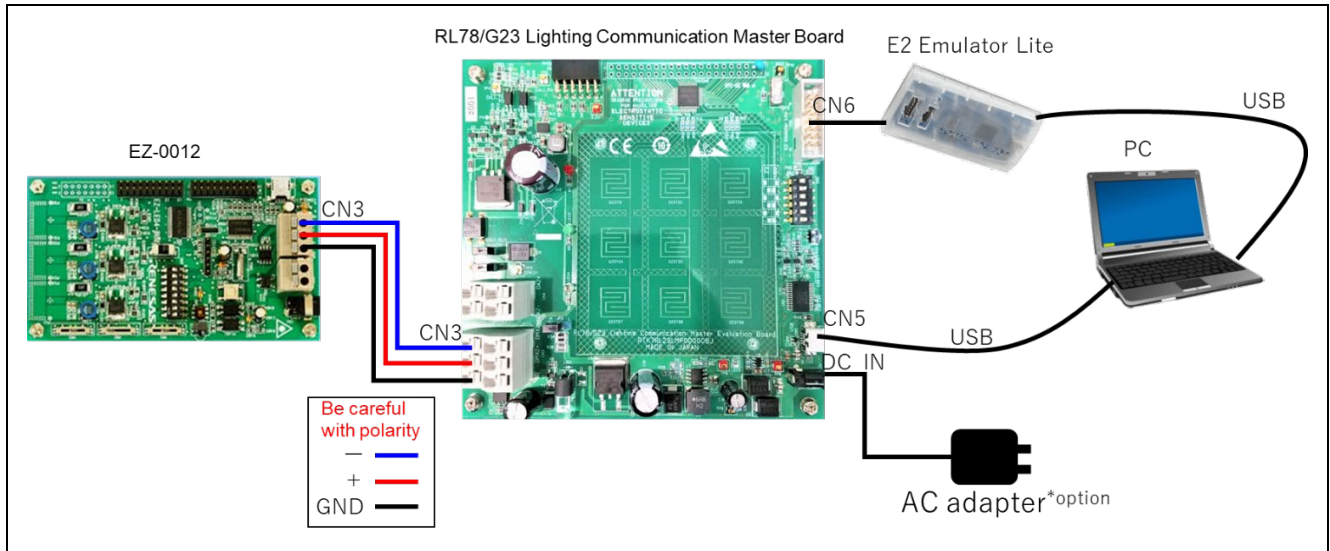
Figure 4-4 Example of Connection for DALI Communication



4.1.8.4 Example of Connection for DMX512 Communication

Figure 4-5 shows an example of hardware connection for DMX512 communication.

Figure 4-5 Example of Connection for DMX512 Communication



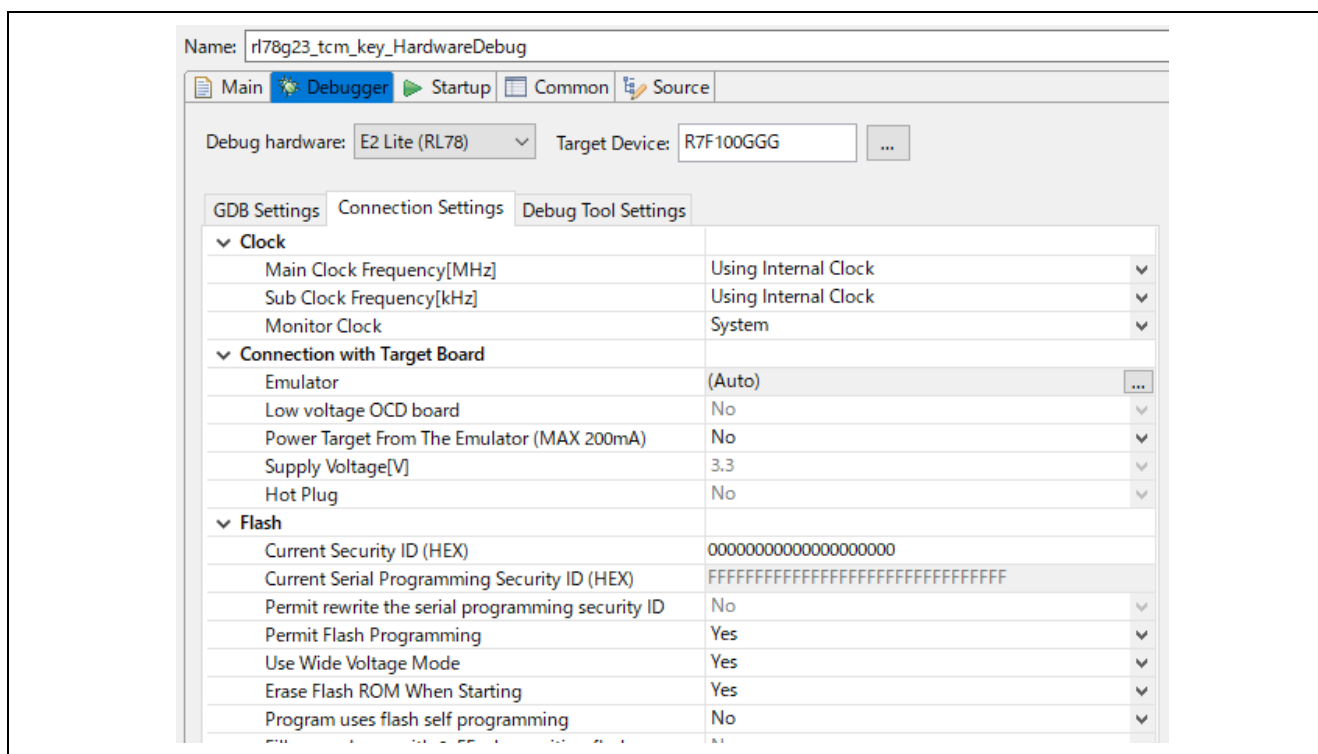
4.1.9 Debugging the Sample Application

1. After you have imported the sample application by using e² studio, click the  button to download the program to the microcontroller.
2. From the [Run] menu, select [Debug Configurations] to open the Debug Configurations window.
3. In the tree view of the window, expand the [Renesas GDB Hardware Debugging] node, and then select an existing debug configuration.
4. In the [Debugger] module, open the [Connection Settings] tab, and then confirm that the settings are specified as shown in Figure 4-6.

Note: If you are using the E2 Lite emulator, select "E2 Lite (RL78)" from the [Debug hardware] drop-down list.

If power is supplied from a USB port, select "COM Port (RL78)" from the [Debug hardware] drop-down list.

Figure 4-6 Settings in the Debug Configurations Window



5. Select [Start Debugging]. When the [Debug] view appears, you are ready for debugging.

For details about installation and other basic operations, refer to the following manual:

- e² studio 2020-04 and e² studio v7.8 Integrated Development Environment User's Manual (R20UT4819)

4.2 Operation Procedure

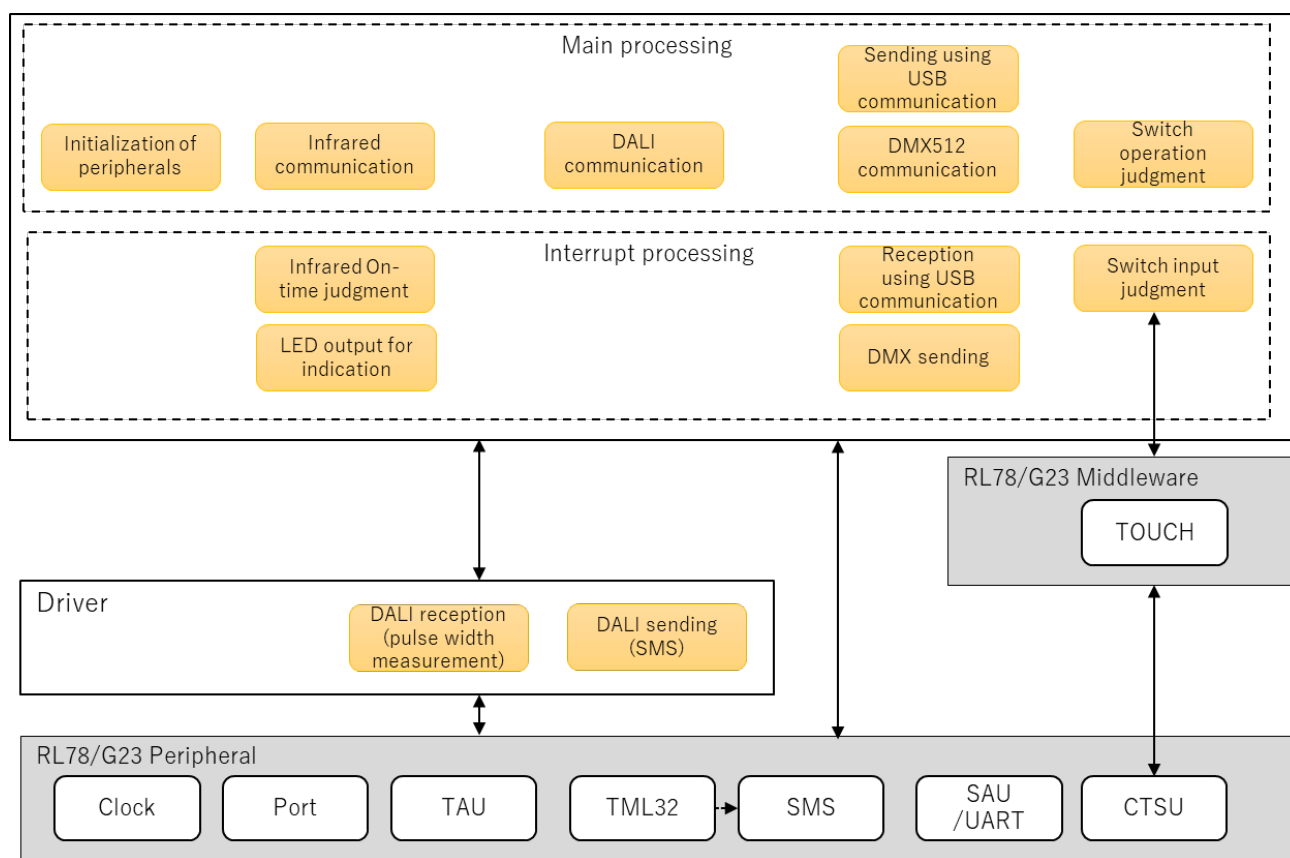
For details about the operation procedure, refer to section 3.3 in the "RL78/G23 Lighting Communication Master Board User's Manual (R20UT5072)".

4.3 Functional Overview

4.3.1 Functional Block Diagram

Figure 4-7 shows the configuration of functional blocks for the sample application.

Figure 4-7 configuration of functional blocks



The sample application consists of the main processing, interrupt processing, and the interfacing to peripherals.

The main processing includes initialization of peripherals, judgment of touch key operations, and communications.

The interrupt processing mainly uses TAU interrupts to perform DALI reception, LED indicator output, and communication type determination.

SAU handles the UART sending to, UART reception from, and DMX512 sending to, the DALI or DMX512 Master Controller GUI. The CTSU passes the measurement results that will be used for determination of the touch key input via the TOUCH module to the sample program. The SNOOZE mode sequencer (hereinafter, "SMS") triggered by an interval timer handles the DALI sending.

4.3.2 Common Information

4.3.2.1 Settings for the Peripherals

Table 4-5 and

Table 4-6 describe the settings of the peripherals.

Table 4-5 Settings of Peripherals (1/2)

Peripheral	Settings								
Clock	CPU clock: fCLK = 32 MHz								
I/O ports	For the LEDs: P62: Output to the power LED P63: Output to the communication status LED P72: Output to the touch-key-On LED Unused pins are configured as follows: <table> <tr> <td>Input/on-board pull-up</td><td>P13 to P15, P17, P51, P73, P120 to P122, P140, P146, and P147</td></tr> <tr> <td>Output Low</td><td>P20 to P24, P26 to P27, and P130</td></tr> <tr> <td>Input (pull-up circuit connection)</td><td>P60 and P61</td></tr> <tr> <td>Input (to enable prevention of flow-through current)</td><td>P137</td></tr> </table>	Input/on-board pull-up	P13 to P15, P17, P51, P73, P120 to P122, P140, P146, and P147	Output Low	P20 to P24, P26 to P27, and P130	Input (pull-up circuit connection)	P60 and P61	Input (to enable prevention of flow-through current)	P137
Input/on-board pull-up	P13 to P15, P17, P51, P73, P120 to P122, P140, P146, and P147								
Output Low	P20 to P24, P26 to P27, and P130								
Input (pull-up circuit connection)	P60 and P61								
Input (to enable prevention of flow-through current)	P137								
Capacitance Sensor Unit (CTSU)	RX family: P71/TS03: Key scan input 1 P70/TS02: Key scan input 2 P50/TS00: Key scan input 3 TX family: P31/TS01: Key scan output 1 P75/TS07: Key scan output 2 P74/TS06: Key scan output 3								

Peripheral	Settings
Timer Array Unit 0 (TAU0)	Channel 4 (TM04): Determining whether USB communication timed out • Operation mode: Interval timer • Counter clock: 500 kHz ($f_{CLK}/2^6$) • Cycle: 9.17 ms • Completion interrupt: Enabled Channel 5 (TM05): Interval timer for periodic processing • Operation mode: Interval timer • Counter clock: 32 MHz ($f_{CLK}/2^0$) • Cycle: 1 ms • Completion interrupt: Enabled

Table 4-6 Settings of Peripherals (2/2)

Peripheral	Settings
Serial Array Unit 0 (UART0)	Channel 0: USB sending • Operation mode: UART mode • Serial clock: 32 MHz • Baud rate: 250 kbps • Data direction: LSB first • Stop bit: 1 bit • Send interrupt: Disabled Channel 1: USB reception • Operation mode: UART mode • Serial clock: 32 MHz • Baud rate: 250 kbps • Data direction: LSB first • Receive interrupt: Enabled Note: The input buffers for reception pins are enabled.

4.3.2.2 Touch Keys

For the touch keys on the master board, the CTSU2L capacitive touch sensing unit is used to measure the capacitance of each electrostatic capacitance sensor.

The CTSU2L unit can detect finger touch from the capacitance change determined by software. Normally, the electrode surface of each electrostatic capacitance sensor is coated with dielectric material, and therefore fingers do not directly contact on electrodes.

The electrostatic capacitance can be detected by either the self-capacitance method or the mutual capacitance method. The master board adopts the mutual capacitance method. In this method, a sensor with two electrodes, send and receive electrodes, detects the capacitance change that arises between the two electrodes when a finger comes close to the sensor.

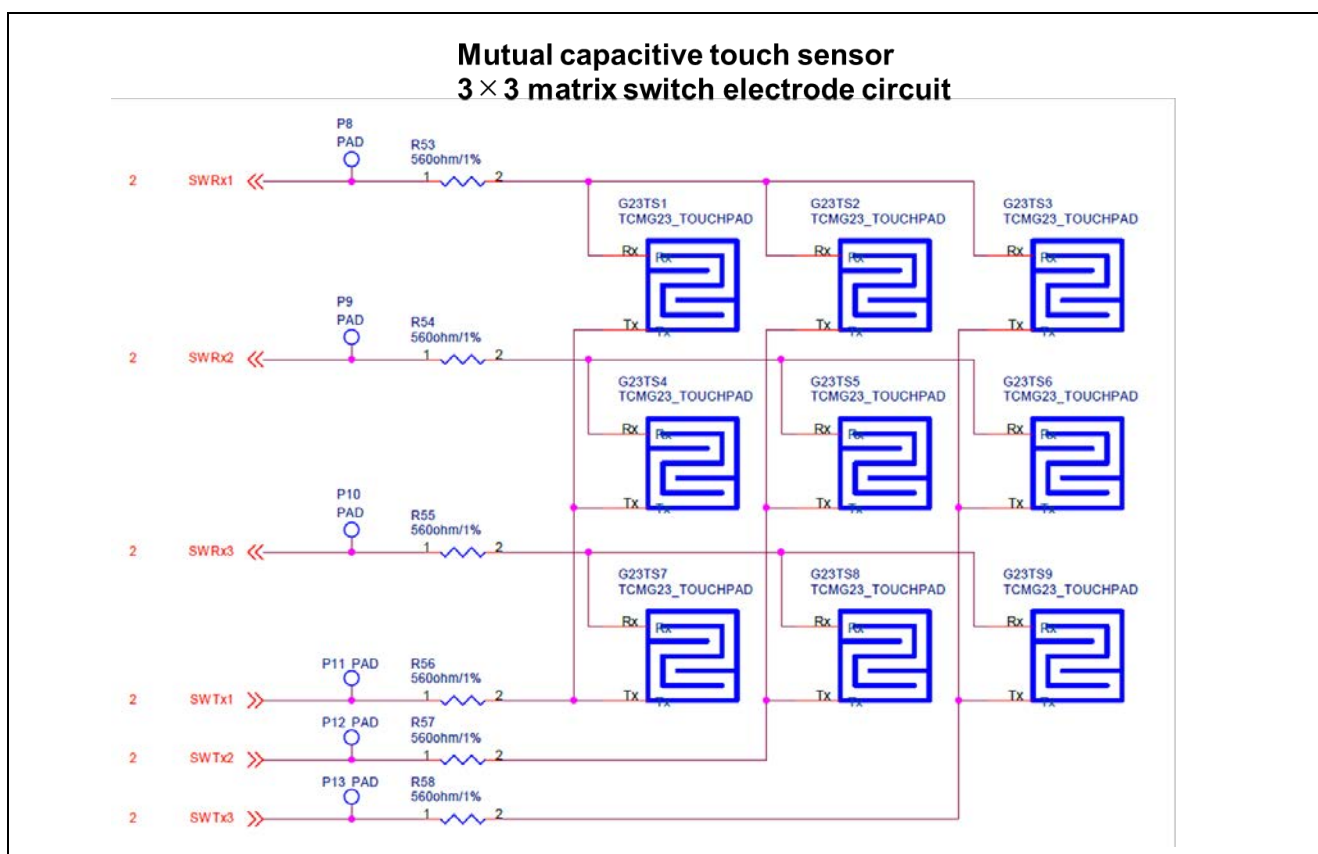
For details, refer to the following application note:

- RL78 Family Capacitive Touch Sensing Unit (CTSU2L) Operation Explanation (R01AN5744)

(1) Touch key circuit diagram

Figure 4-8 shows the circuit diagram of the touch keys on the master board.

Figure 4-8 Touch Key Status Determination Circuit



(2) Touch-key press determination

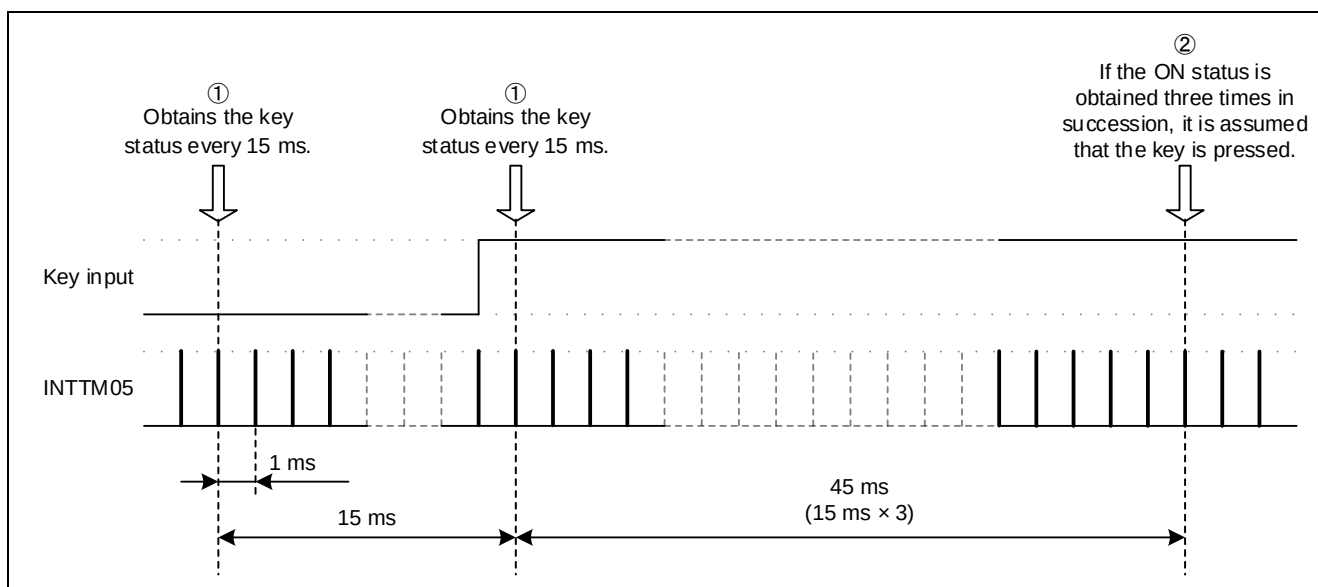
To determine whether a touch key is pressed, the CTSU module measures the electrostatic capacitance in mutual capacitance mode.

For details about mutual capacitance mode, refer to the following application note:

- "1.2.2 Mutual Capacitance Mode" in the manual "RL78 Family CTSU Module Software Integration System (R11AN0484)"

The sample application performs touch-key press determination at the times shown in Figure 4-9.

Figure 4-9 Touch-Key Press Determination Timing



1. Obtains the key input status (ON or OFF) at 15-millisecond intervals.
2. If the ON status continues for 45 ms, (15 ms × 3: prevention of misjudgment due to chattering), it is assumed that the key is pressed.

(3) Touch-key status determination

Table 4-7 describes the touch-key status determination timing and method.

Table 4-7 Touch-Key Status Determination

Item	Description
Touch-key continuous operation determination ("resend" determination)	After ON is confirmed as the key input status in step ② in Figure 4-9, if the ON status is obtained successively for 150 ms, the key touch is assumed to be an instruction for continuous operation. This determination is performed only for a touch key for which the "resend" continuous operation is set as the operation type. For the commands that support continuous operation, refer to section 4.3.2.2(5).
Touch-key continuous operation determination ("repeat" determination)	After ON is confirmed as the key input status in step ② in Figure 4-9, if the ON status continues for 108 ms[#], the key touch is assumed to be an instruction for continuous operation. This determination can be set for only infrared (IR) communication. This determination is performed only for a touch key for which the "repeat" continuous operation is set as the operation type.
Touch-key command execution determination	If ON is confirmed as the key input status for only one of touch keys TS1 to TS9 in step ② in Figure 4-9, the command assigned to the relevant key is executed. If it is determined that two or more touch keys are pressed concurrently, the sample application executes no command and discards the results, assuming that no touch keys were pressed.

[#]: The 108-millisecond threshold for determination of the "repeat" continuous operation is compliant with the repeat command send timing in the NEC format of the infrared communication protocol.

(4) Touch key settings

Table 4-8 describes the settings specified for each touch key.

Table 4-8 Touch Key Settings

Item	Description
Send data	DALI: 8 bits (address byte) + 8 bits (opcode byte) IR: 16 bits (custom code) + 8 bits (data ch1) + 8 bits (data Ch2)
Operation type	One-time operation (once): 0 Continuous operation (resend): 1 (resends the command) Continuous operation (repeat): 2 (repeats command sending) (available for IR communication only)
Operation protocol	NONE: 0 DALI: 1 IR: 2

Touch-key operations can be categorized into two types as follows:

- One-time operation: Sends a command once when a touch key is pressed once.
- Continuous operation: Sends a command multiple times when a touch key is pressed once.

There are the following two types of continuous operation:

- Resend: Sends a command again when the operation type of the touch key is determined to be continuous operation.
- Repeat: Sends a repeat command when the touch key is in the ON status for 108 ms.

Note: The "repeat" continuous operation supports only infrared (IR) communication.

(5) Commands that can be set for touch keys

Table 4-9 describes the settings that can be specified for each touch key.

Table 4-9 Commands That Can Be Set for Touch Keys

Protocol	Command	Operation type
DALI	DAPC	once
	OFF	once
	UP	once or resend
	DOWN	once or resend
	STEP UP	once
	STEP DOWN	once
	RECALL MAX LEVEL	once
	RECALL MIN LEVEL	once
	STEP DOWN AND OFF	once
	ON AND STEP UP	once
	GO TO SCENE	once
DMX512	--	--
IR	Non-programmable	once, resend, or repeat

(6) Default Command Assignment for the Touch Keys

Table 4-10 shows the default settings for the touch keys.

Table 4-10 Touch Key Default Settings

Touch key	Communication protocol	Description
TS1	DALI	Broadcast RECALL MAX LEVEL once
TS2	DALI	Broadcast RECALL MIN LEVEL once
TS3	DALI	Broadcast OFF once
TS4	DALI	Broadcast UP resend
TS5	DALI	Broadcast DOWN resend
TS6	DALI	Broadcast Scene 0 once
TS7	DALI	Broadcast ON AND STEP UP once
TS8	DALI	Broadcast STEP DOWN AND OFF once
TS9	IR	Repeat (custom code: 0x0000, ch1: 0x5A, ch2: 0xDA)

(7) Setting commands for the touch keys

The following describes how to set commands for the touch keys.
Touch key commands are implemented in r_key_define.h in the sample application.

(a) Communication protocol settings

Keywords SW1_MODE to SW9_MODE are used to set communication protocols for the touch keys.
The following table shows the values that can be set for SW1_MODE to SW9_MODE.

Table 4-11 List of Communication Protocol Macros

Macro name	Value	Description
SW_MODE_NONE	(0)	None
SW_MODE_DALI	(1)	DALI communication protocol
SW_MODE_IR	(2)	Infrared communication protocol

The following shows an example of communication protocol settings.

Figure 4-10 Example of Communication Protocol Settings

```
--
82      /* Key Mode */
83      #define SW1_MODE          (SW_MODE_DALI)
84      #define SW2_MODE          (SW_MODE_DALI)
85      #define SW3_MODE          (SW_MODE_DALI)
86      #define SW4_MODE          (SW_MODE_DALI)
87      #define SW5_MODE          (SW_MODE_DALI)
88      #define SW6_MODE          (SW_MODE_DALI)
89      #define SW7_MODE          (SW_MODE_DALI)
90      #define SW8_MODE          (SW_MODE_DALI)
91      #define SW9_MODE          (SW_MODE_IR)
--
```

(b) Command settings

Keywords SW1_CMD to SW9_CMD are used to set the commands that are sent when the corresponding touch keys are pressed.

The following table shows the formats of commands that can be set for SW1_CMD to SW9_CMD.

Table 4-12 Command Format

Communication protocol	Description
DALI communication	Address setting + command
Infrared communication	Custom code + data code + Ch1 settings + Ch2 settings

The following figure shows an example of command settings.

Figure 4-11 Example of Command Settings

```
60      /* SW setting command */
61      #define SW1_CMD                (0xFF05)
62      #define SW2_CMD                (0xFF06)
63      #define SW3_CMD                (0xFF00)
64      #define SW4_CMD                (0xFF01)
65      #define SW5_CMD                (0xFF02)
66      #define SW6_CMD                (0xFF10)
67      #define SW7_CMD                (0xFF08)
68      #define SW8_CMD                (0xFF07)
69      #define SW9_CMD                (0x00005ADA)
```

(c) Operation mode settings

Keywords SW1_FLG to SW9_FLG are used to define the operation modes that are used when the corresponding touch keys are pressed.

The following table shows the values that can be set for SW1_FLG to SW9_FLG.

Table 4-13 List of Operation Mode Macros

Macro name	Value	Description
SW_MESSAGE_REQ_SEND_ONCE	(0x01)	Sends the command once.
SW_MESSAGE_REQ_SEND_TWICE	(0x02)	Sends the command twice.
SW_MESSAGE_REQ_SEND_RESEND	(0x05)	Sends the command continuously (every 150 milliseconds).
SW_MESSAGE_REQ_SEND_REPEAT	(0x06)	Sends the repeat command continuously (every 108 milliseconds) (for IR communication only).

The following figure shows an example of operation mode settings.

Figure 4-12 Example of Operation Mode Settings

```

71      /* Key Flag */
72      #define SW1_FLG          (SW_MESSAGE_REQ_SEND_ONCE)
73      #define SW2_FLG          (SW_MESSAGE_REQ_SEND_ONCE)
74      #define SW3_FLG          (SW_MESSAGE_REQ_SEND_ONCE)
75      #define SW4_FLG          (SW_MESSAGE_REQ_SEND_RESEND)
76      #define SW5_FLG          (SW_MESSAGE_REQ_SEND_RESEND)
77      #define SW6_FLG          (SW_MESSAGE_REQ_SEND_ONCE)
78      #define SW7_FLG          (SW_MESSAGE_REQ_SEND_ONCE)
79      #define SW8_FLG          (SW_MESSAGE_REQ_SEND_ONCE)
80      #define SW9_FLG          (SW_MESSAGE_REQ_SEND_REPEAT)

```

4.3.3 DALI Communication

DALI (Digital Addressable Lighting Interface), an open global communication protocol for lighting control, is mainly used to control the intensity and color of multiple fluorescent lamps or LED illuminators. This protocol allows lighting products of different manufacturers to communicate with each other.

The sample application works as the master in DALI communication when connected to a PC on which the DALI Master Controller GUI is operating while the standalone feature or master controller GUI communication feature is in use.

The purpose of the driver section of this sample application is to verify DALI communication simply. Therefore, the behavior of the driver is partially different from a driver that is completely compliant with the DALI101 standards.

For details about DALI communication, refer to "3.1.5 Overview of DALI Communication" in the application note "Lighting Communications Using RL78/I1A (Reception)" (R01AN1115).

4.3.3.1 Peripheral Settings

Table 4-14 shows the details of the peripheral settings.

Table 4-14 Details of the Peripheral Settings

Peripheral	Settings
32-bit interval timer (TML32)	ITL000: 16 μ s interval • Operation clock: 250 kHz (fITL0/128) • Detection of completion: Enabled
Timer array unit 0 (TAU0)	Channel 0 (TM00): Reception time measurement • Operation mode: Interval timer • Counter clock: 32 MHz (fCLK/2⁰) • Cycle: 100 μs • Detection of completion: Enabled Channel 1 (TM01): Input pulse interval measurement • Counter clock: 500 kHz (fCLK/2⁶) • Completion interrupt: Enabled
SNOOZE mode sequencer (SMS)	• Start trigger: INTITL (ITL000 timer interrupt) • Output pin: P10 • Completion interrupt: Enabled

4.3.3.2 DALI Driver

The DALI driver consists of receive processing and send processing. The receive processing uses a timer array unit to process the waveform by software. The send processing uses a SNOOZE mode sequencer and 32-bit interval timer to output a waveform.

(1) Receive processing

The sample application uses channels 0 and 1 of timer array unit 0 to process the received waveform.

The sample application measures the receive bit width of the DALI_RX pin and sequentially stores the receive bit width and bit level (high or low) in a ring buffer. It detects a stop condition and system failure by measuring the duration time of the DALI_RX level.

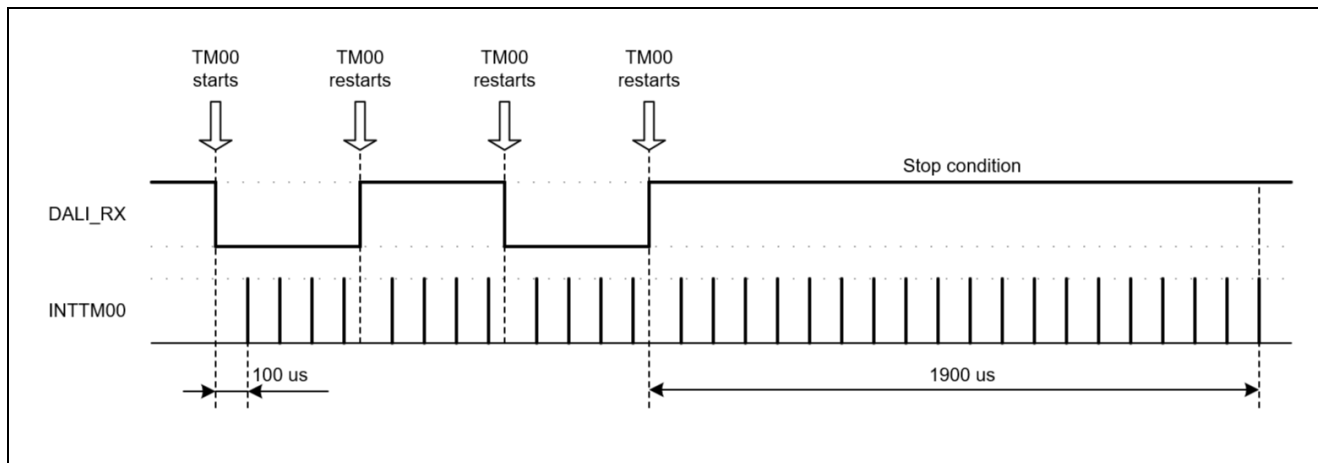
When the sample application detects a stop condition, it assumes that frame reception is complete and checks whether the received frame satisfies the DALI standards from the bit width and bit level (high or low) stored in the ring buffer.

If the received frame is normal, the sample application also provides information indicating whether the frame was received within the last 100 ms.

(a) Measurement of the reception level duration time

The sample application starts or restarts a 100 μ s interval timer (channel 0 of timer array unit 0) by using edge detection for the DALI_RX pin as a trigger. The interrupt handler of the timer adds 100 μ s to the duration time. The sample application uses this handler to measure the duration time of the reception level (high or low). The measurement results are used to detect a stop condition or system failure.

Figure 4-13 Measurement of the Reception Level Duration Time

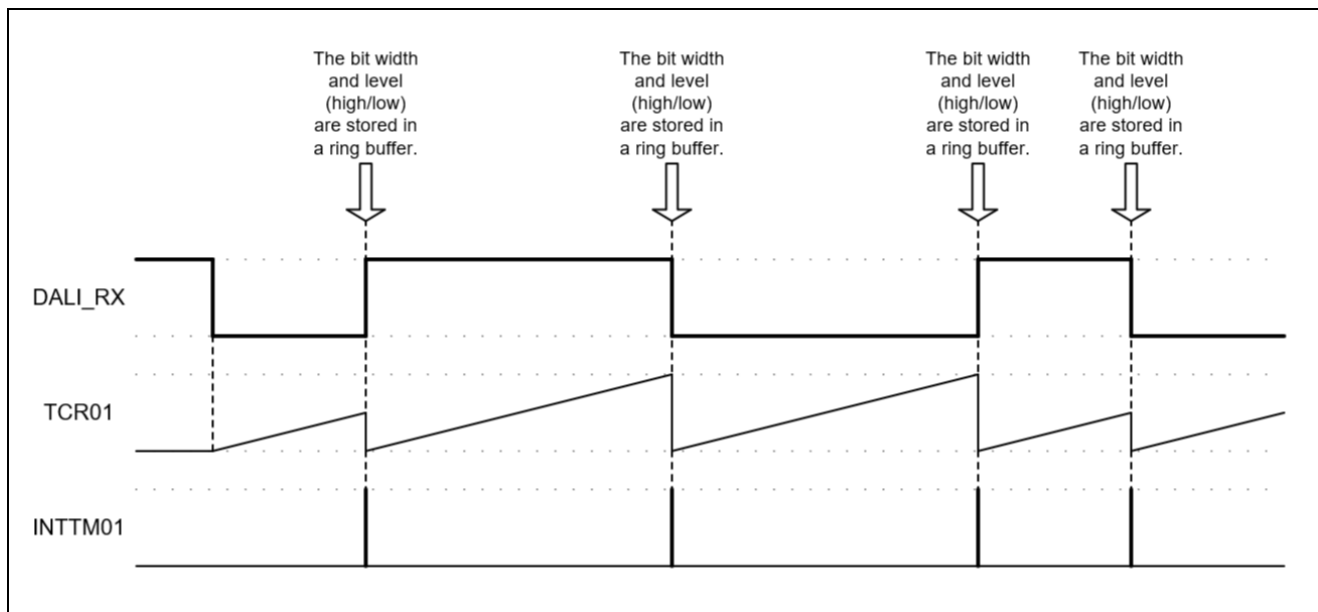


(b) Measurement of the receive bit width

The sample application uses the input pulse interval measurement feature of the timer array unit (channel 1 of timer array unit 0) to measure the receive bit width.

The receive bit width and the bit level (high or low) are stored in a ring buffer by the INTTM01 interrupt that occurs simultaneously with capturing.

Figure 4-14 Measurement of the Receive Bit Width



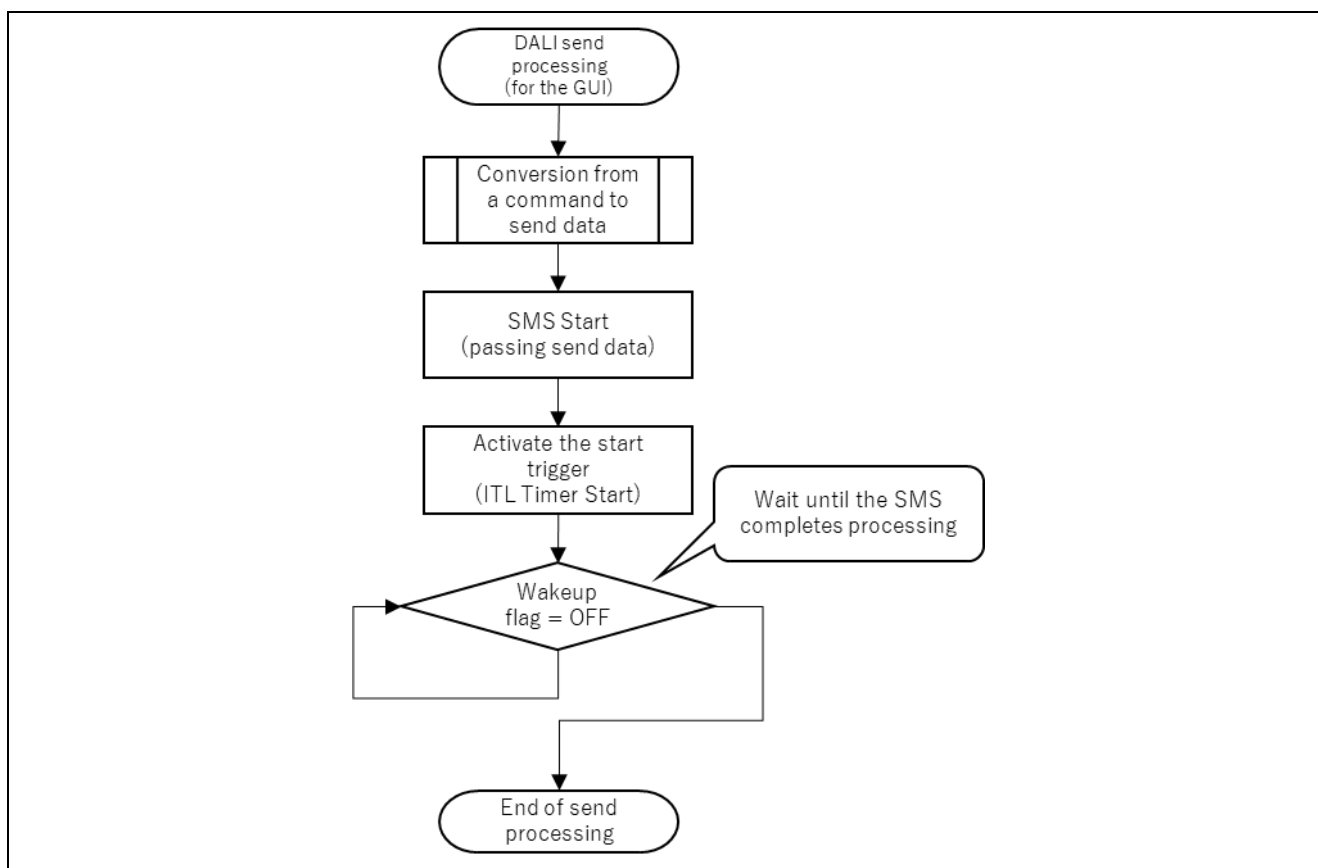
(2) Send processing

The sample application uses RL78/G23 peripherals (the SMS and 32-bit interval timer) to generate and output a DALI send waveform.

(a) SMS invocation

Figure 4-15 shows the flowchart for DALI waveform output processing.

Figure 4-15 flowchart for DALI waveform output processing using SMS



The SMS can handle data on a byte basis. Therefore, all bits of a 16-bit DALI command (uint16_t) are stored in a byte array (uint8_t[16]) on a byte basis, and the first address and last address of the byte array are specified for SMS Start.

The SMS is started by an interval timer (ITL timer). The timer starts after SMS Start passes data.

For communication with the GUI, "normal completion (NORMAL)" must be returned after a DALI waveform is sent. Therefore, an interrupt by the Wakeup function is used to return to the postprocessing for completion of waveform sending.

(b) Trigger to start the SMS

The trigger to start the SMS is the interval detection interrupt (INTITL).

The sample application sets DALI send data for the SMS and then starts the interval timer so that the SMS is started by an interval timer interrupt that occurs when a certain length of time elapses.

For details about the SMS start trigger settings, refer to Table 4-14.

(c) SMS processing

Table 4-15 describes the specifications of the SMS processing that is performed by the sample code.

Table 4-15 Specifications of the SMS Processing

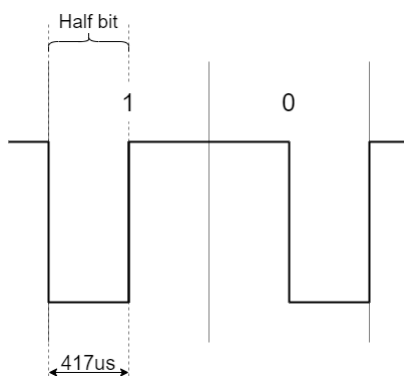
Synopsis	DALI waveform output processing by the SMS
Description	- An INTITL interrupt starts the SMS. - The program performs bitwise data determination for the argument and sends a waveform that is appropriate for each bit. Bit data == 0: Outputs 1 and then 0 to P10. Bit data == 1: Outputs 0 and then 1 to P10. - When the sending of all data bits is complete, the output of the P01 port is set to High and the Wakeup function is used to cause an SMS termination interrupt (INTSMSE).
Arguments [#]	address_d: First address of the send data address_e: Last address of the send data
Return value	None
Remarks	None

Note: These arguments are specified in the R_Config_SMS_Start function settings.

The waveform output in step (2) in Table 4-15 is output at the send bit timing defined in the 101 ed2.0 standards. In the sample application, the SMS wait time is set to 417 μ s, which is an approximate value for the typical value defined in the standards.

Table 4-16 Send Bit Timing

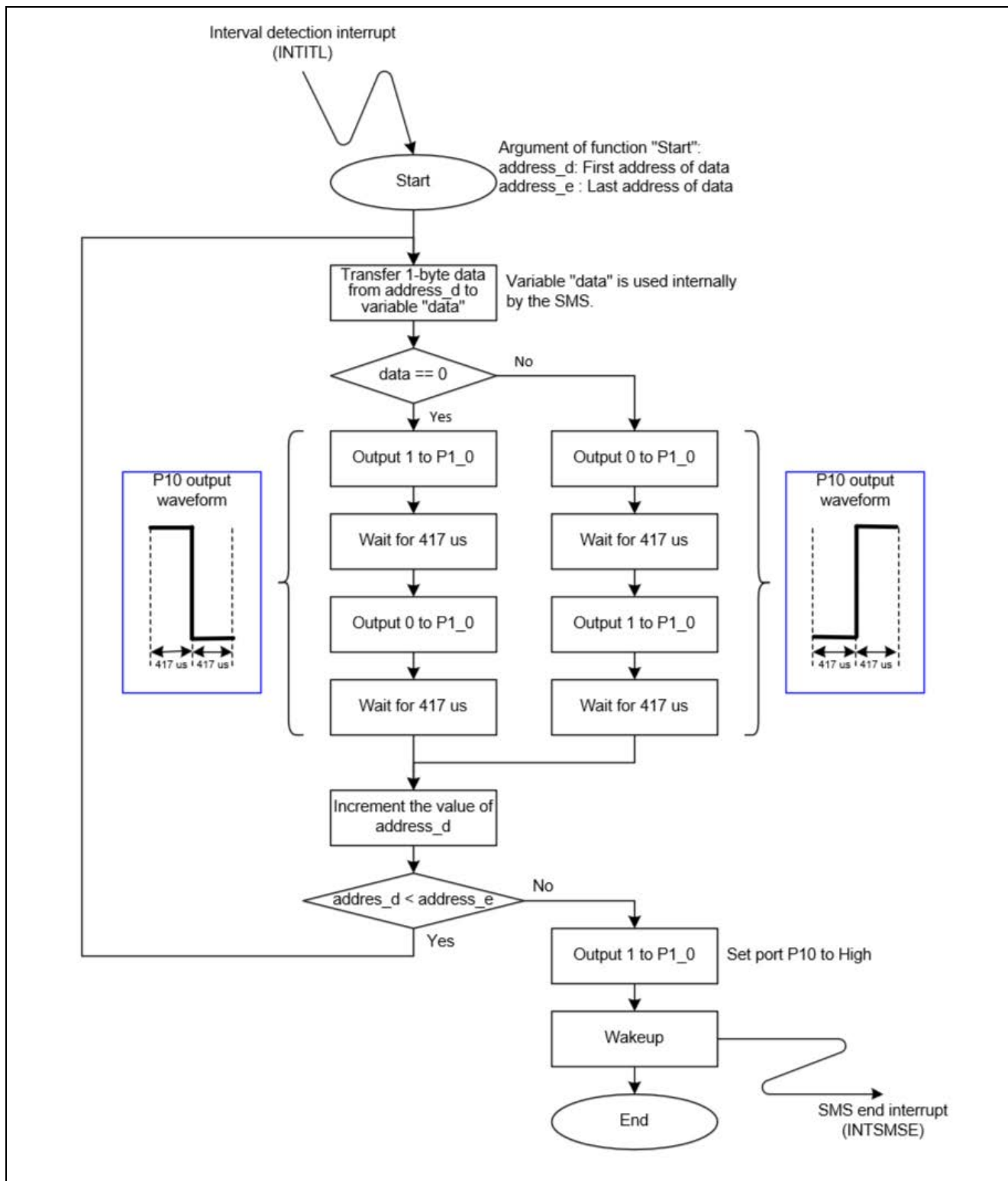
Description	Minimum	Typical	Maximum
Half bit	366.7 μ s	416.7 μ s	466.7 μ s



(d) Flowchart for the SMS Processing

Figure 4-16 shows the flowchart for the SMS processing.

Figure 4-16 Flowchart for the SMS Processing



(e) SMS control registers

Table 4-17 and Table 4-18 show the values that are set for the SMS control registers.

Table 4-17 General-Purpose Sequencer Registers 0 to 15

Register name	Set value	Remarks
SMSG0	0000H	Fixed value: 0000H
SMSG1	0000H	address_d, the first argument for the Start function
SMSG2	0	Fixed value for data comparison: 0000H
SMSG3	0000H	address_e, the second argument for the Start function
SMSG4	&ITLS0	ITLS0 address
SMSG5	&P1	P1 address
SMSG6	1	Fixed value for incrementation of address_d: 0001H
SMSG7	&smsInitValue0	Initial value for the "data" variable: 0000H
SMSG8	0000H	Not used
SMSG9	0000H	Not used
SMSG10	0000H	Not used
SMSG11	0000H	Not used
SMSG12	0000H	Not used
SMSG13	0000H	Not used
SMSG14	0000H	Not used
SMSG15	FFFFH	Fixed value: FFFFH

Table 4-18 Sequencer Instruction Registers 0 to 31

Register name	Set value	Remarks
SMSI0	3720H	MOVW SMSG2, [SMSG7+0]
SMSI1	0400H	MOV [SMSG4+0], SMSG0
SMSI2	1120H	MOV SMSG2, [SMSG1+0]
SMSI3	7202H	CMPW SMSG2, SMSG0
SMSI4	8072H	BZ \$7
SMSI5	5500H	CLR1 [SMSG5+0].0
SMSI6	9687H	WAIT 68h, 07h
SMSI7	4500H	SET1 [SMSG5+0].0
SMSI8	9687H	WAIT 68h, 07h
SMSI9	7002H	CMPW SMSG0, SMSG0
SMSI10	8052H	BZ \$5
SMSI11	4500H	SET1 [SMSG5+0].0
SMSI12	9687H	WAIT 68h, 07h

SMSI13	5500H	CLR1 [SMSG5+0].0
SMSI14	9687H	WAIT 68h, 07h
SMSI15	7160H	ADDW SMSG1, SMSG6
SMSI16	7132H	CMPW SMSG1, SMSG3
SMSI17	8F10H	BC \$-15
SMSI18	4500H	SET1 [SMSG5+0].0
SMSI19	F001H	WAKEUP
SMSI20-31	0000H	Not used

4.3.4 DMX512 Communication

DMX512 is a wired communication protocol for sending digital data and is widely used for stage lighting, exhibition lighting, and other industrial lighting purposes using devices provided with dimmers, scanners, mobile lights, or flashes.

The sample application works as the master (transmitter) in DMX512 communication when connected to a PC on which the DMX512 Master Controller GUI is operating while the master controller GUI communication feature is in use.

For details about DMX512 communication, refer to "4.1.1 Overview of the DMX512 standard" in the application note "Lighting Communications Using RL78/I1A (Reception)" (R01AN1115).

4.3.4.1 Peripheral Settings

Table 4-19 shows the details of the peripheral settings.

Table 4-19 Details of the Peripheral Settings

Peripheral	Settings
Timer array unit 0 (TAU0)	Channel 3 (TM03): Management of the output time • Operation mode: Interval timer • Counter clock: 32MHz (fCLK/2⁰) • Cycle: 88 or 8 μs • Detection of completion: Enabled
Serial array unit 0 (UART1)	Channel 2: DMX512 sending • Operation mode: UART mode • Serial clock: 32 MHz • Baud rate: 250 kbps • Data direction: LSB first • Stop bit: 2 bits • Send interrupt: Enabled Channel 3: DMX512 reception • Operation mode: UART mode • Serial clock: 32 MHz • Baud rate: 250 kbps • Data direction: LSB first • Send interrupt: Disabled

4.3.4.2 DMX512 Send Processing

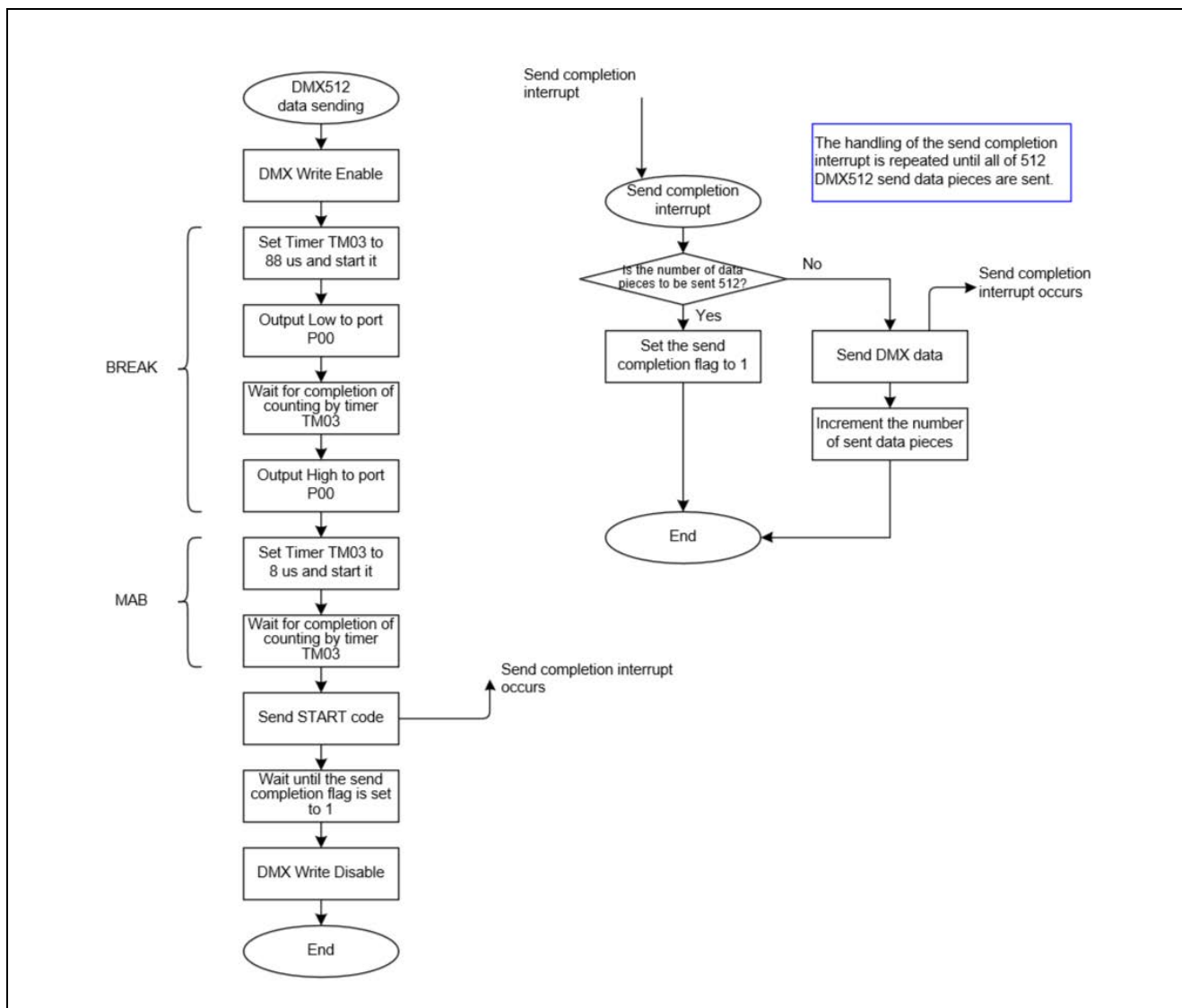
The send processing of DMX512 communication uses channel 3 of timer array unit 0 and channel 2 of serial array unit 0.

- Channel 3 of timer array unit 0 (TM03)
Uses an interval timer to measure the BREAK and MAB times.
- Channel 2 of serial array unit 0 (UART1)
Uses the send functionality of UART1 to send the data slots 0 to 512.

The sample application outputs BREAK and MAB by measuring the time based on the counting completion interrupt for TM03 and controls the level (high/low) of the P00 port. It then repeats the following sequence of operations 512 times: UART1 is used to send data from data slots 0 to 512, and then the next data is sent by using a send completion interrupt.

Figure 4-17 shows the flowchart for the DMX512 data sending.

Figure 4-17 Flowchart for the DMX512 Data Sending



4.3.5 Infrared Communication

Infrared communication is wireless communication that uses infrared light to send and receive signals. The infrared communication of this sample application supports the NEC format, an infrared transmission protocol used in industry circles worldwide. NEC-format infrared remote control uses infrared light of about 950 nm for sending bytes of information at low speed.

The sample application can communicate with slave devices via infrared communication when the standalone feature is used.

For details about infrared communication, refer to "5.1.1 Overview of the NEC IR protocol" in the application note "Lighting Communications Using RL78/I1A (Reception)" (R01AN1115).

4.3.5.1 Peripheral Settings

Table 4-20 shows the details of the peripheral settings.

Table 4-20 Details of the Peripheral Settings

Peripheral	Settings
I/O port	P25: Digital input (channel switch for infrared communication)
Timer array unit 0 (TAU0)	Channel 2 (TM02): Output time measurement • Operation mode: Interval timer • Counter clock: 32 MHz (fCLK/2⁰) • Cycle: 562.5 μs • Completion interrupt: Enabled Channel 7 (TM07): Rectangular wave output • Counter clock: 32 MHz (fCLK/1) • Cycle: 13.157 μs (=38 kHz/2) • Completion interrupt: Disabled

4.3.5.2 Infrared Send Processing

The send processing of infrared communication uses channels 2 and 7 of timer array unit 0.

- Channel 2 of timer array unit 0 (TM02): An interval timer is used to measure the time at which to output the ON/OFF period in the send waveform and the time for the whole frame.
- Channel 7 of timer array unit 0 (TM07): Uses the rectangular wave output functionality to output a waveform of 38 kHz (carrier frequency) that indicates the ON status of the output waveform.

For the sample application to output standards-compliant waveforms, it measures the time by using a counting completion interrupt generated from TM02 at 562.5 μ s intervals and controls start and stop of output from TM07.

Figure 4-18 and Figure 4-19 show the flowchart for infrared data sending.

Figure 4-18 Flowchart for Infrared Data Sending (1/2)

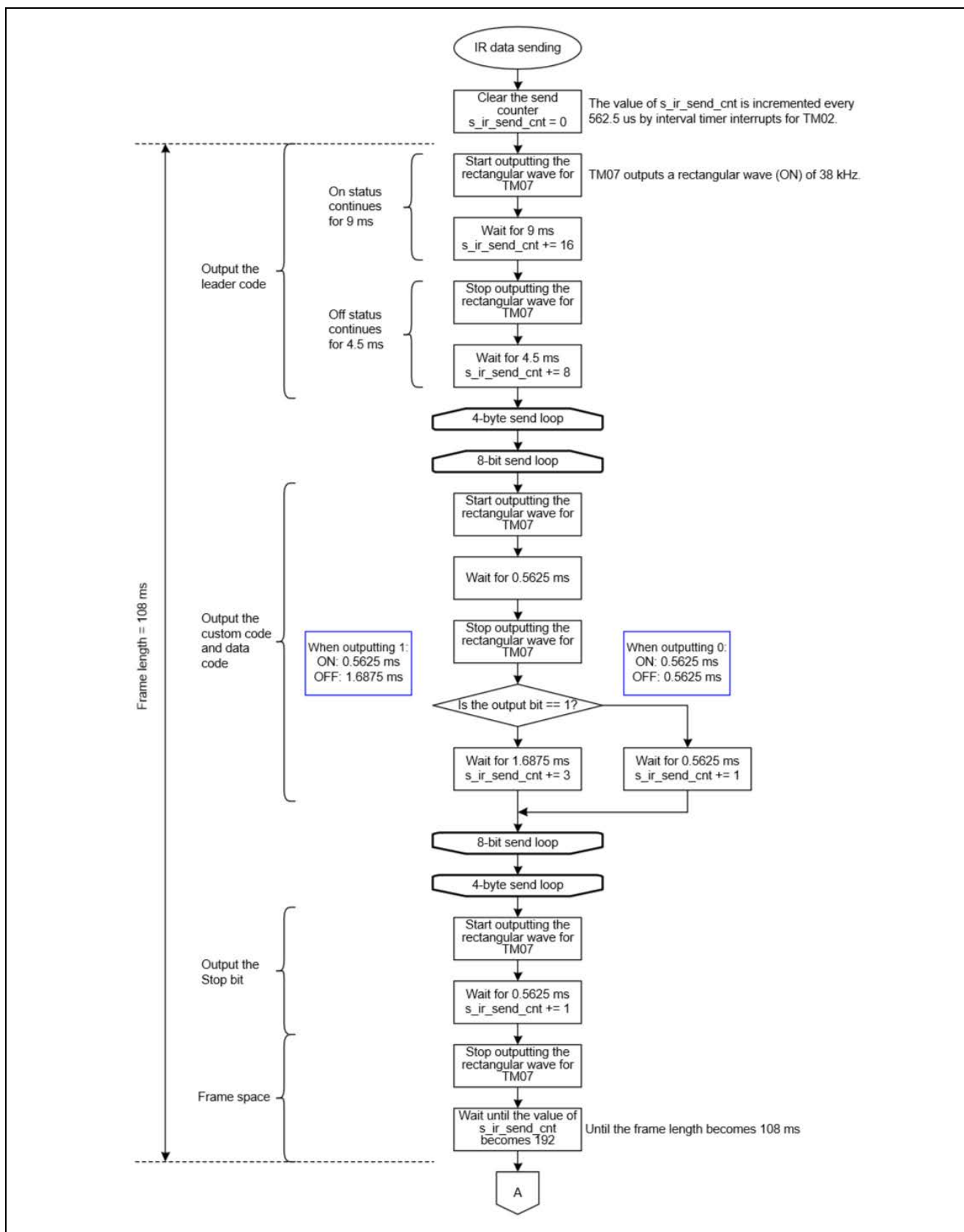
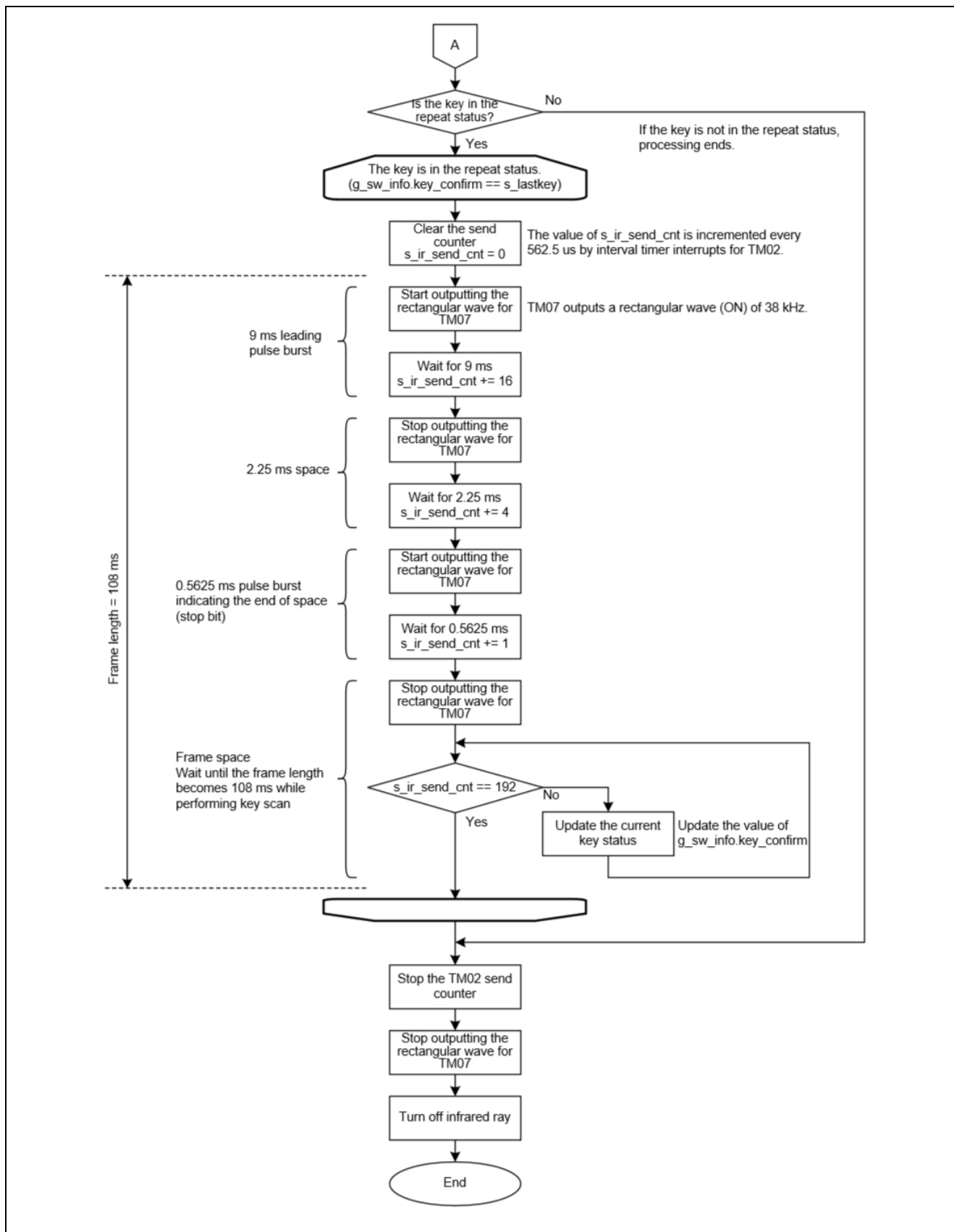


Figure 4-19 Flowchart for Infrared Data Sending (2/2)



4.4 Software Configuration

This section describes the software configuration of the sample application.

4.4.1 Folder Structure

Table 4-21 and Table 4-22 describe the folder structure of the sample application.

Table 4-21 Folder Structure (1/2)

Folder or file name	Description	Use of Smart Configurator
rl78g23_tcm_key<DIR>	Sample code folder	
\src<DIR>	Program folder	
\qe_gen<DIR>	Folder in which QE for Capacitive Touch	✓ Note
\QE-Touch<DIR>	Folder containing the configuration files for	✓ Note
r_dali.c	Source file for DALI communication	
r_dali.h	DALI communication header file	
r_dali101.c	Source file for the DALI driver	
r_dali101.h	DALI driver header file	
r_dali101_rx.c	Source file for DALI driver receive processing	
r_dali101_rx.h	Header file for DALI driver receive processing	
r_dmx.c	Source file for DMX512 communication	
r_dmx.h	Header file for DMX512 communication	
r_gui_com.c	Source file for communication with the	
r_gui_com.h	Header file for communication with the	
r_interrupt.c	Source file for interrupt functions	
r_interrupt.h	Header file for interrupt functions	
r_ir.c	Source file for infrared communication	
r_ir.h	Header file for infrared communication	
r_itl.c	Source file for interval timers	
r_itl.h	Header file for interval timers	
r_key_define.h	Header file for touch key definitions	
r_key_scan.c	Source file for touch key scanning	
r_key_scan.h	Header file for touch key scanning	
r_led_disp.c	Source file for controlling LED indication	
r_led_disp.h	Header file for controlling LED indication	
r_main.c	Source file for the "main" processing	
r_main.h	Header file for the "main" processing	
r_port.c	Source file for port configuration	
r_port.h	Header file for port configuration	
r_qe_touch.c	Source file for QE-Touch	
r_qe_touch.h	Header file for QE-Touch	
r_sau.c	Source file for SAU	
r_sau.h	Header file for SAU	
r_tau.c	Source file for TAU	
r_tau.h	Header file for TAU	

Table 4-22 Folder Structure (2/2)

Folder or file name	Description	Use of Smart Configurator
\rl78g23_tcm_key<DIR>	Sample code folder	
\src<DIR>	Program folder	
\smc_gen<DIR>	Folder generated by Smart Configurator	✓
\Config_SMS<DIR>	Folder for the programs for the SMS	✓
Config_SMS.c	Source program for the SMS	✓
Config_SMS.h	Header program for the SMS	✓
Config_SMS_ASM.smsasm	ASM source file for the SMS	✓
Config_SMS_user.c	Interrupt source file for the SMS	✓
\general<DIR>	Folder for storing the initialization program and other common programs	✓
\rm_touch<DIR>	Folder for Touch middleware programs	✓
\l_bsp<DIR>	Folder for the BSP programs	✓
\l_config<DIR>	Folder for the BSP_CFG programs	✓
\l_ctsu<DIR>	Folder for the CTSU programs	✓
\l_pincfg<DIR>	Folder for the PIN_CFG programs	✓

<DIR>: The item is a directory.

Note: Code is generated by using QE for Capacitive Touch V3.1.0.

4.4.2 List of Option Byte Settings

Table 4-23 shows the option byte settings used in the sample application.

Table 4-23 List of Option Byte Settings

Address	Set value	Description
000C0H / 010C0H	1110 1111B	The watchdog timer is disabled. (Counting stops after reset cancellation.)
000C1H / 010C1H	1111 1111B	LVD reset mode (Falling: 1.65 V, rising: 1.69 V)
000C2H / 010C2H	1110 1000B	HS mode, high-speed on-chip oscillator: 32 MHz
000C3H / 010C3H	1000 0100B	On-chip debugging is disabled.

4.4.3 Flowcharts

4.4.3.1 Main Processing

Figure 4-20 and Figure 4-21 shows the flowchart of the main processing

Figure 4-20 Main Processing (1/2)

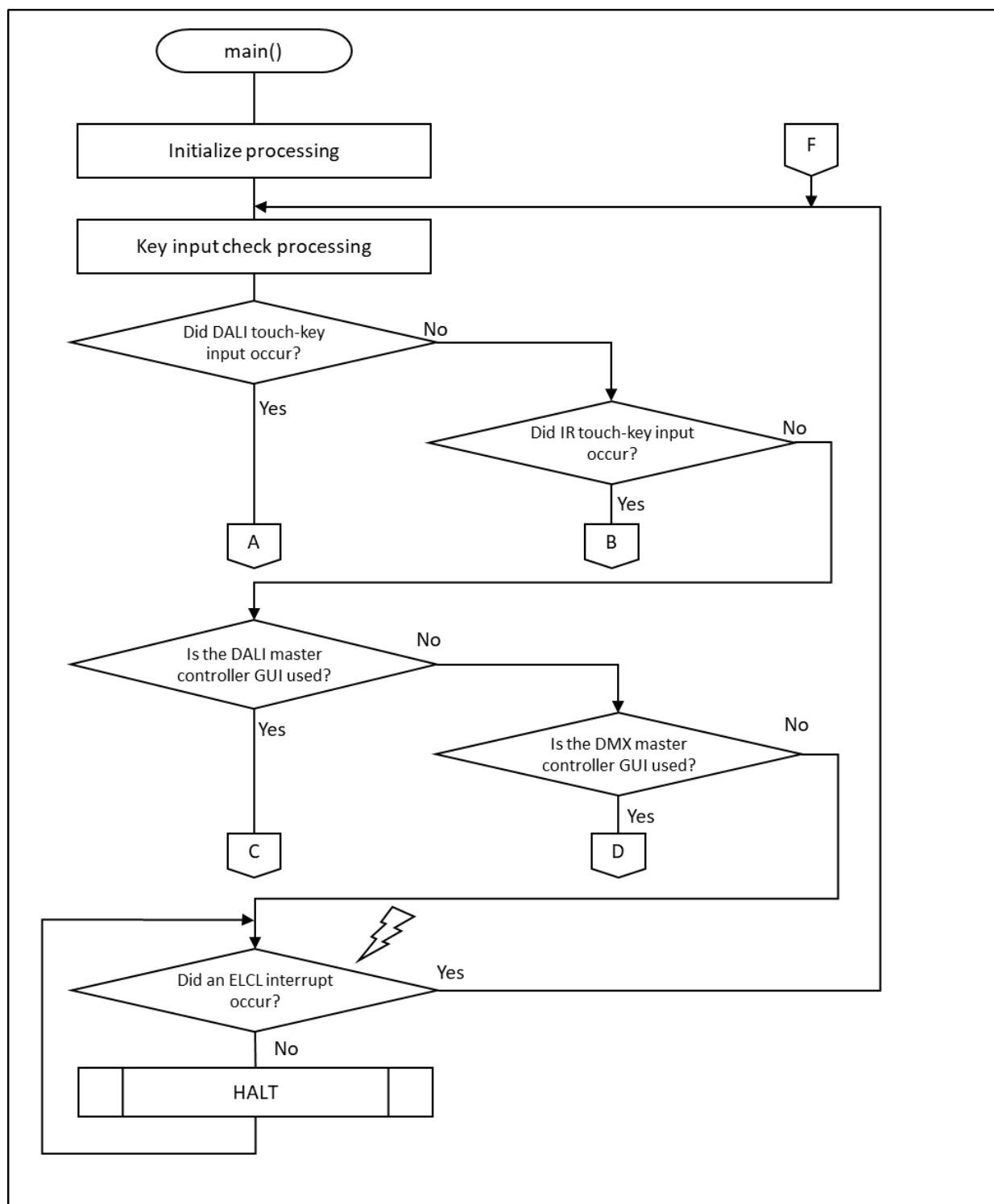
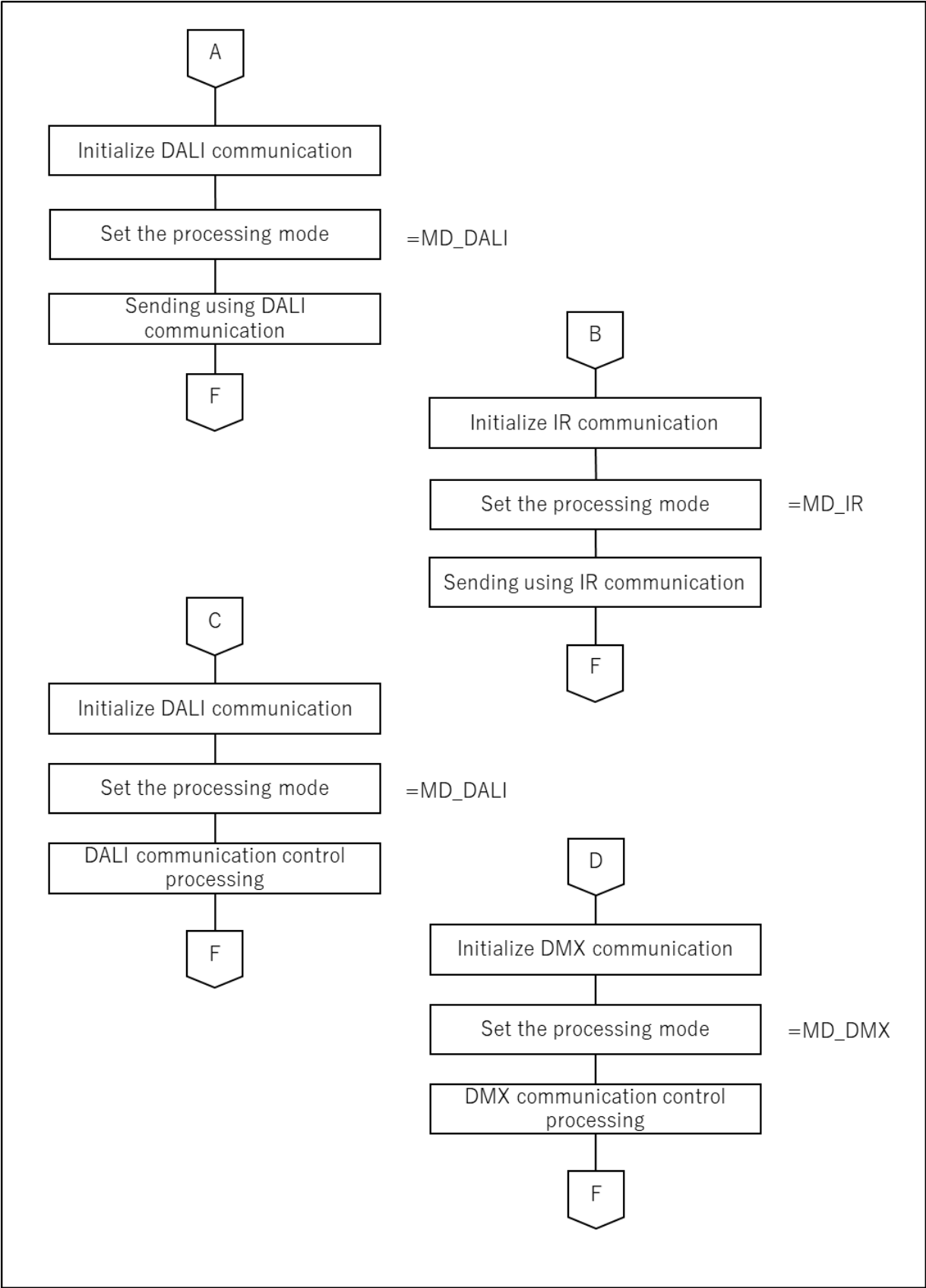


Figure 4-21 Main Processing (2/2)



Start:

After the program starts, it initializes the peripherals and variables, and then enters a loop.

Main processing loop:

Control remains in the loop until touch key input or connection with the master controller GUI occurs.

When touch key input occurs, the system obtains the information about the pressed key and performs the operation assigned to the key.

If no touch key input occurs, the system performs communication control processing for the connected master controller GUI.

Processing for the touch key input:

The system specifies the peripheral settings according to the obtained processing mode (DALI or IR).

The system sets the processing mode.

The system sends the specified settings to Control Gear.

After that, control returns to the main processing loop.

Communication control processing:

The system specifies the peripheral settings according to the connected master controller GUI (DALI or DMX).

The system sets the processing mode.

The system performs communication control processing.

Communication control processing has a processing loop in which communications with the master controller GUI and Control Gear are continuously performed. (Refer to sections 4.4.3.2 and 4.4.3.3.)

If touch key input occurs inside the communication control processing, after the request from the master controller GUI is processed, control exits the communication control processing and returns to the main loop.

The system obtains the information about the touch key input that occurred in the main loop and handles the processing for the touch key input.

Common notes:

The processing mode is set to the mode of the protocol with which processing is performed.

The processing mode is set after all initialization processing ends.

Initialization processing is performed only when the processing mode changes.

The master controller GUI to be used cannot be changed during startup.

4.4.3.2 DALI communication control processing

Figure 4-22, Figure 4-23 and Figure 4-24 shows the flowchart of the DALI communication control processing.

Figure 4-22 DALI communication control processing (1/3)

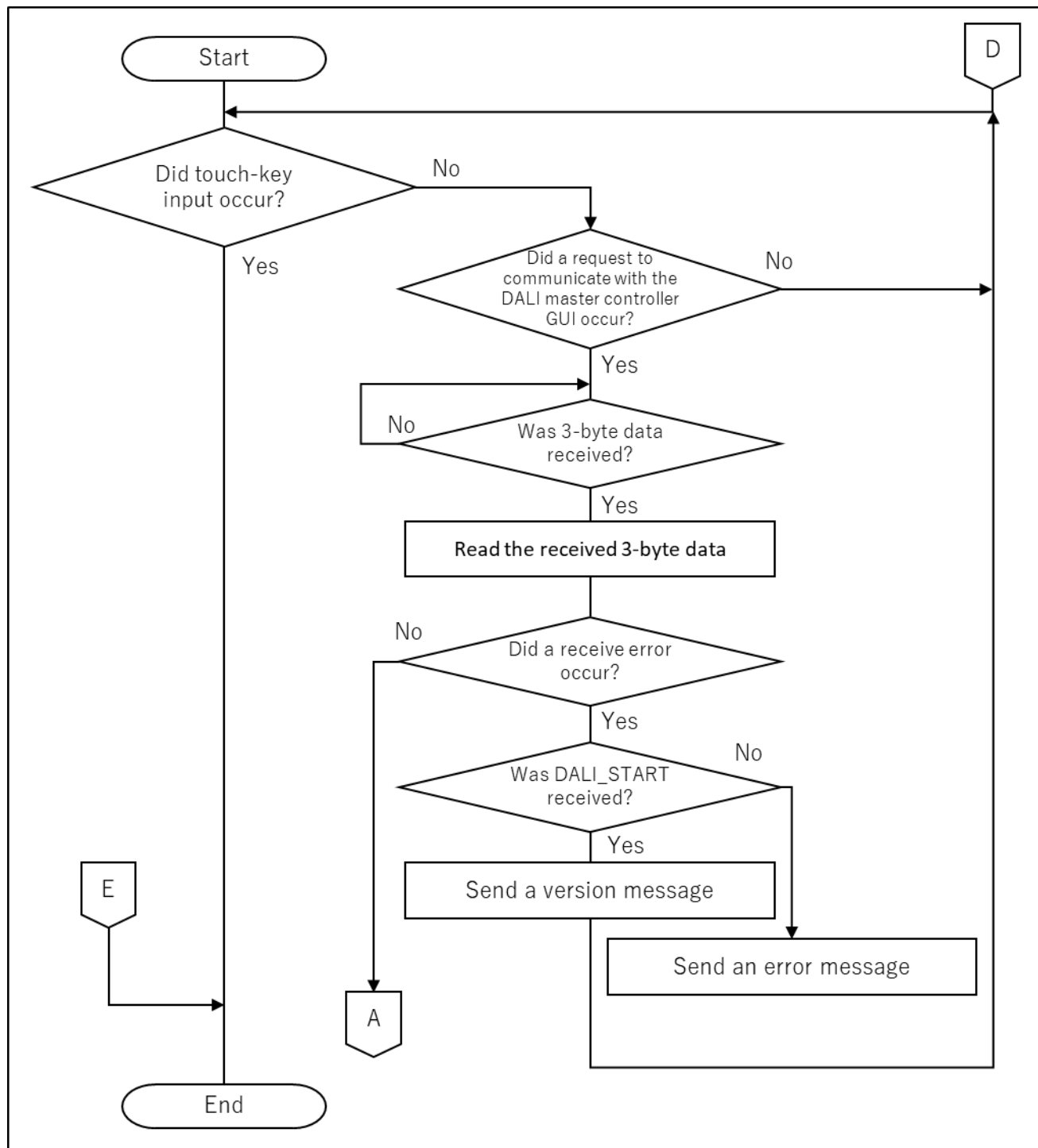


Figure 4-23 DALI communication control processing (2/3)

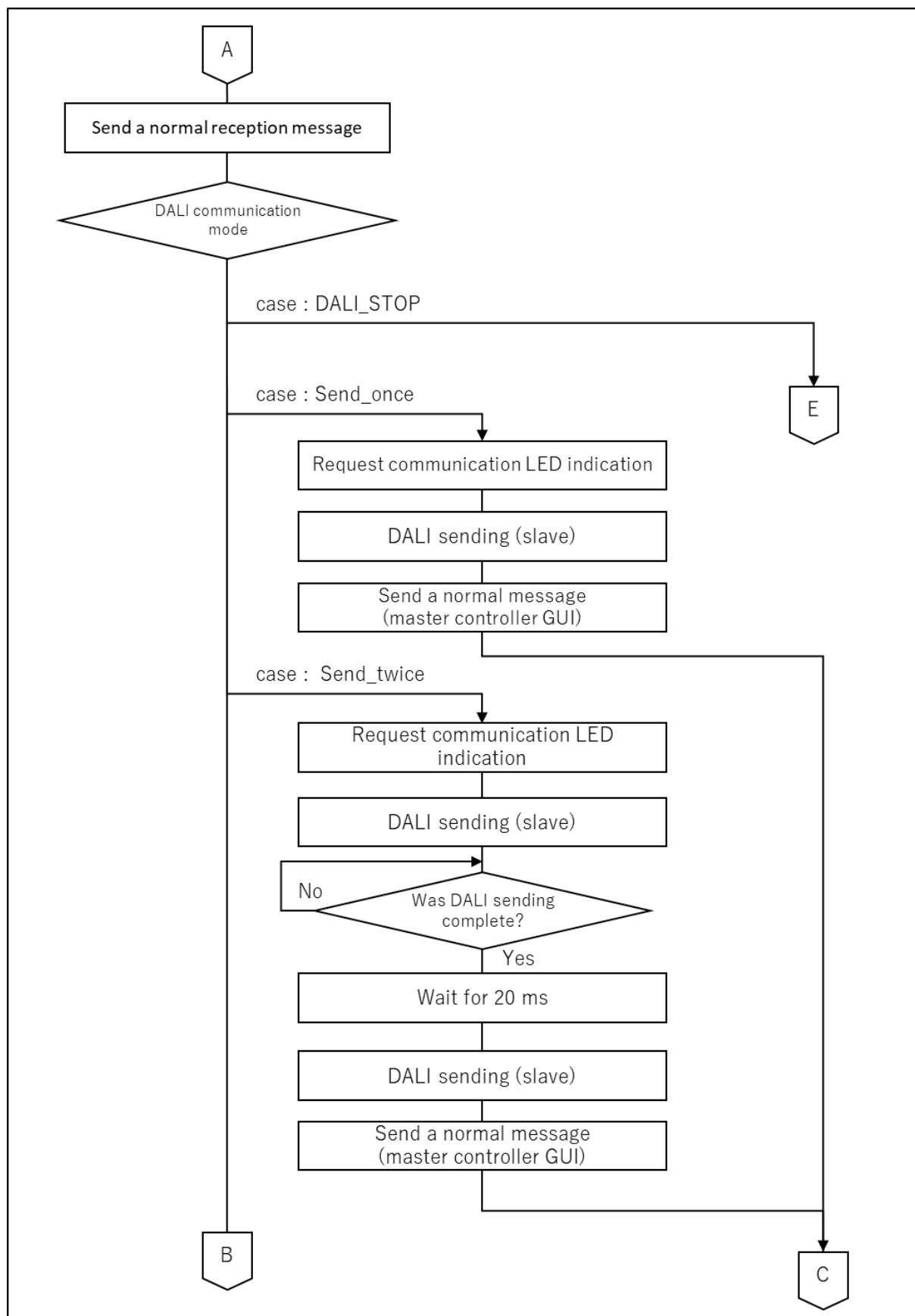
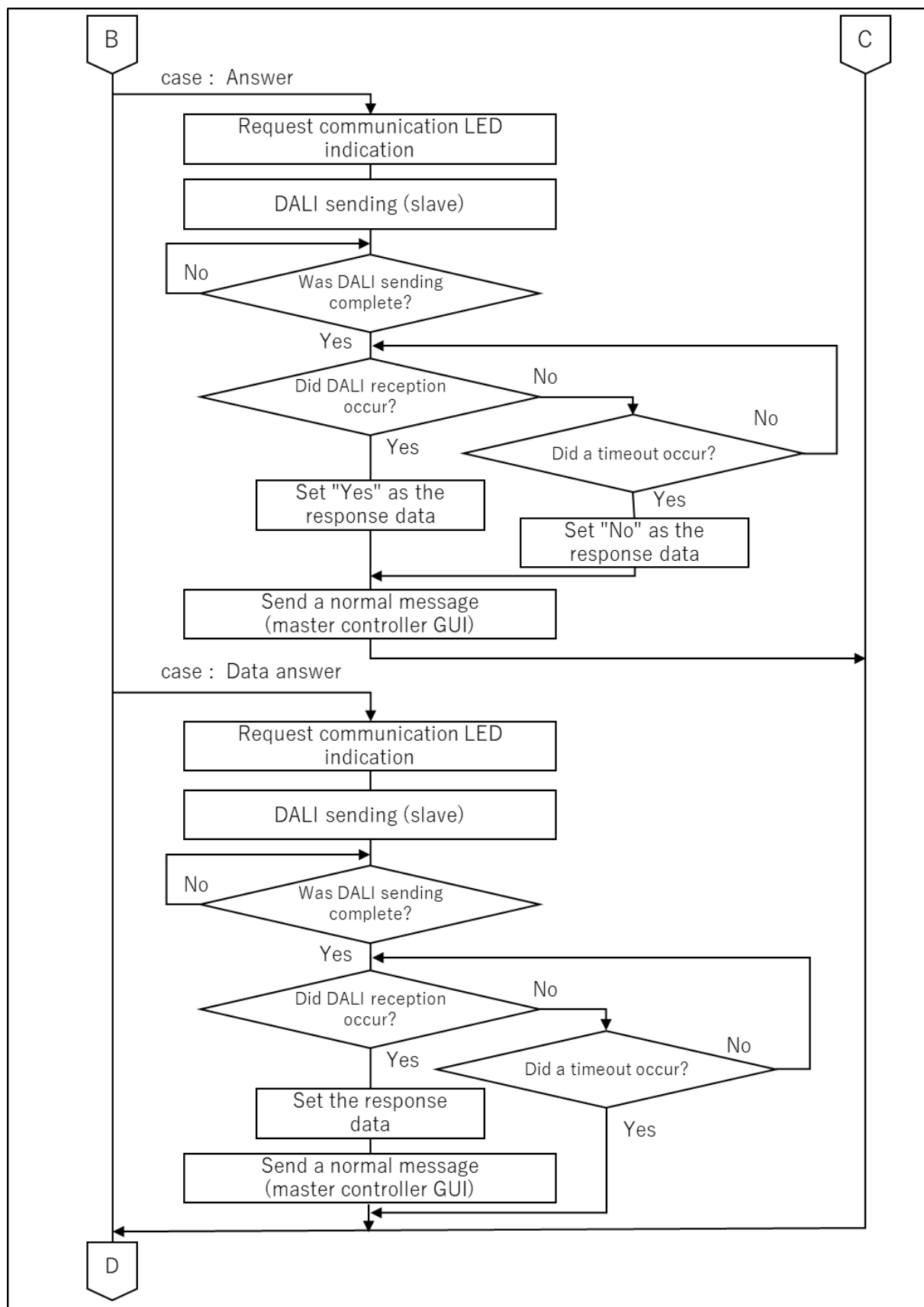


Figure 4-24 DALI communication control processing (3/3)



When a communication request from the master controller GUI occurs, DALI communication control receives three-byte data. DALI communication control then sends a forward frame to Control Gear based on the received data.

After sending a forward frame to Control Gear, if DALI communication control needs a reply, it receives a backward frame from Control Gear, and then sends the received data to the master controller GUI.

DALI communication control can request two types of information: Yes/No or data.

DALI communication control processing:

Control enters the DALI communication control processing loop.

Main processing loop for DALI communication control processing:

The system checks for occurrence of touch key input.

If touch key input occurs, the system performs the operation according to the touch key.

If touch key input does not occur and reception from the DALI Master Controller GUI occurs, the system performs DALI communication processing.

Processing for the touch key input:

When reception from the master controller GUI occurs, the system discards it.

Control exits the DALI communication control processing loop and returns to the main processing loop.

DALI communication processing:

The system waits for 3-byte data reception (reception timeout value: 9.17 ms).

If 3-byte data reception occurs, the system obtains the received data.

If reception times out and the system cannot obtain 3-byte data:

- If the first byte is DALI_START, the system sends a version message to the master controller GUI and control returns to the beginning of the processing.
- In other cases, the system sends an error message (data error) to the master controller GUI and control returns to the beginning of the processing.

The system verifies the header data.

The system performs an operation according to the value set in the received header.

DALI_STOP:

Control exits the DALI communication control processing loop and returns to the main processing loop.

Once:

The system turns on LED3 (the LED is turned off automatically 500 ms later).

The system sends a forward frame to Control Gear.

The system sends a normal message to the master controller GUI.

Twice:

The system turns on LED3 (the LED is turned off automatically 500 ms later).

The system sends a forward frame to Control Gear.

The system waits for completion of the sending.

The system waits for 20 ms.

The system sends a forward frame to Control Gear.

The system sends a normal message to the master controller GUI.

Answer:

The system turns on LED3 (the LED is turned off automatically 500 ms later).

The system sends a forward frame to Control Gear.

The system waits for completion of the sending.

After completion of the sending, the system sets a timer so that reception of a backward frame will time out after 20.0 ms.

If a backward frame is received, "Yes" (0xFF) is set as the response to the master controller GUI.

If reception times out, "No" (0x00) is set as the response to the master controller GUI.

The system sends the response value to the master controller GUI.

Data answer:

The system turns on LED3 (the LED is turned off automatically 500 ms later).

The system sends a forward frame to Control Gear.

The system waits for completion of the sending.

After completion of the sending, the system sets a timer so that reception of a backward frame will time out after 20.0 ms.

If a backward frame is received, the value of the backward frame is set as the response to the master controller GUI.

The system sends the response value to the master controller GUI.

If reception times out, the system does not respond to the master controller GUI (in anticipation of timeout of response waiting on the GUI side).

Other:

The system sends an error message (DALI Header data error) to the master controller GUI.

4.4.3.3 DMX512 communication control processing

Figure 4-25 and Figure 4-26 shows the flowchart of the DMX512 communication control processing.

Figure 4-25 DMX512 communication control processing (1/2)

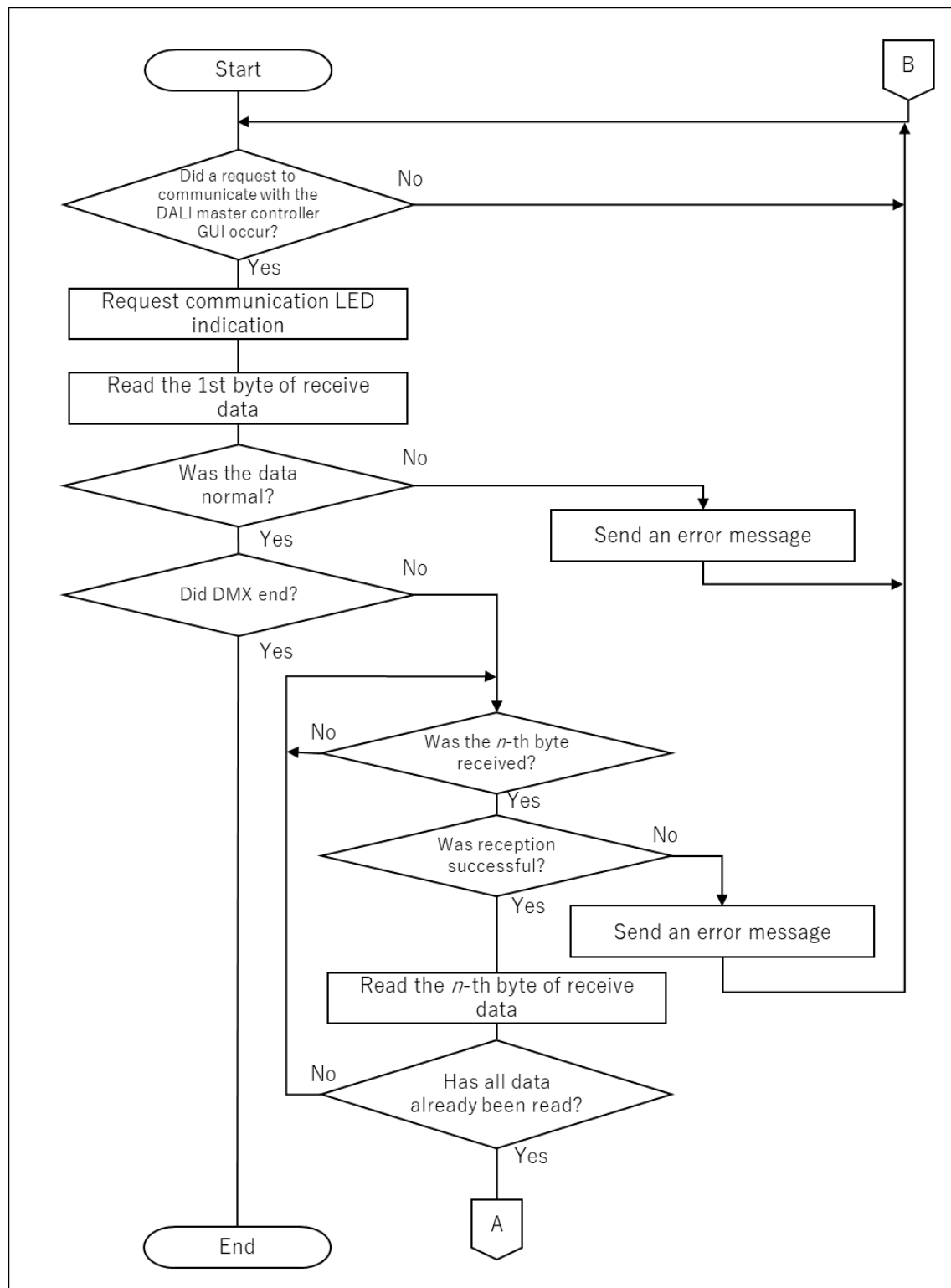
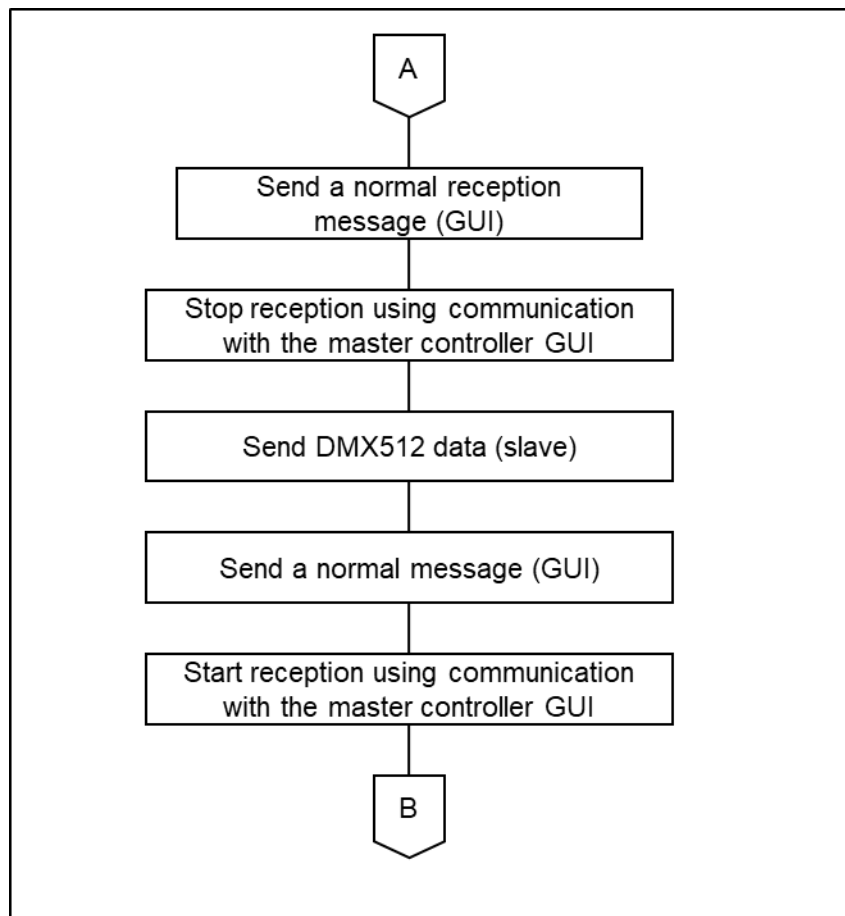


Figure 4-26 DMX512 communication control processing (2/2)



In DMX512 communication control, when the master controller GUI issues a communication request, the system receives the start code data and determines the number of bytes to be received. When the system receives the determined number of bytes, it sends all the data to the slave.

Touch key operations are unavailable.

DMX512 communication control processing:

Control enters the DMX512 communication control processing loop.

Main processing loop in the DMX512 communication control processing:

When the system receives data from the master controller GUI, it performs DMX512 communication processing.

DMX512 communication processing:

The system turns on LED3 (the LED is turned off automatically 500 ms later).

The system receives the first-byte data.

If the first-byte data is invalid, the system sends an error message (start code error) to the master controller GUI and cancels the DMX512 communication processing.

If the first-byte data is DMX start code, the system sends a version information message to the master controller GUI.

If the first-byte data is DMX stop code, control exists the DMX512 communication control processing loop.

If the first-byte data is the duty cycle 0x00, the system sets 512 as the number of bytes to be received.

The system receives the set number of bytes.

If the system fails to obtain all data due to a timeout, it sends an error message (data error) to the master controller GUI and terminates the DMX512 communication processing.

When all data is correctly received, the system performs sending to the slave.

Sending to the slave:

The system sends a normal message to the master controller GUI.

Reception from the master controller GUI is prohibited.

Data is sent to the slave.

The system sends a normal message to the master controller GUI.

The system starts reception from the master controller GUI.

5. Notes

For the DALI send waveform generated by using the SNOOZE mode sequencer (SMS), the duration of high-level or low-level output sometimes becomes longer than 417 μ s that is set as the SMS wait time. Therefore, adjust the wait time settings according to the actually output waveform.

6. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

7. Reference Documents

RL78/G23 User's Manual: Hardware (R01UH0896)

QE for Capacitive Touch V3.1.0 Release Note (R20UT5144)

RL78 Family Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)

RL78 Family Capacitive Touch Sensing Unit (CTSU2L) Operation Expalanation (R01AN5744)

RL78 Family CTSU Module Software Integration System (R11AN0484)

RL78 Family User's Manual: Software (R01US0015J)

e² studio Integrated Development Environment User's Manual: Getting Started Guide (R20UT4819)

DALI Master Controller GUI User's Manual (R20UT0715)

DMX512 Master Controller GUI User's Manual (R01US0200)

RL78/G23 Lighting Communication Master Board User's Manual (R20UT5072)

RL78/I1A DC/DC LED Control Evaluation Board User's Manual (R01UH0363)

The latest versions can be downloaded from the Renesas Electronics website.

Technical updates:

The latest versions can be downloaded from the Renesas Electronics website.

DALI Standards manuals:

IEC 62386-101:2014+AMD1:2018 CSV Consolidated version (ed.2.1)

IEC 62386-102:2014+AMD1:2018 CSV Consolidated versioned (ed.2.1)

Website and Support

Renesas Electronics Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/contact/>

All trademarks and registered trademarks are the property of their respective owners.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Nov.16.22	—	First Edition

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products 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. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance 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 the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended 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; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.