RL78/G23

DALI-2 Input Device Push Button(301) Sample Application

Summary

This application note describes a sample application that performs DALI (Digital Addressable Lighting Interface) communication using the RL78/G23 microcontroller.

The sample application operates as an Input Device. The supported DALI standards are as follows

- IEC 62386-101 Edition2.1 (hereafter 101ed.2.1)
- IEC 62386-103 Edition1.0 (hereafter 103ed.1.0)
- IEC 62386-301 Edition1.0 (hereafter 301ed.1.0)

To achieve standard-compliant communication, the waveforms of Manchester encoded DALI signals are processed using the peripheral functions in the RL78/G23.

This application note assumes that you already have a working knowledge of DALI; for more information on the DALI standard, see 6. Reference documents.

Target Device

RL78/G23

Notes: When applying this application note to other microcontrollers, please modify it to suit the specifications of the microcontroller and evaluate it thoroughly.
DALI-2 certification is not a test for semiconductors or software. Please build the device as an Input Device that can support DALI-2 certification.

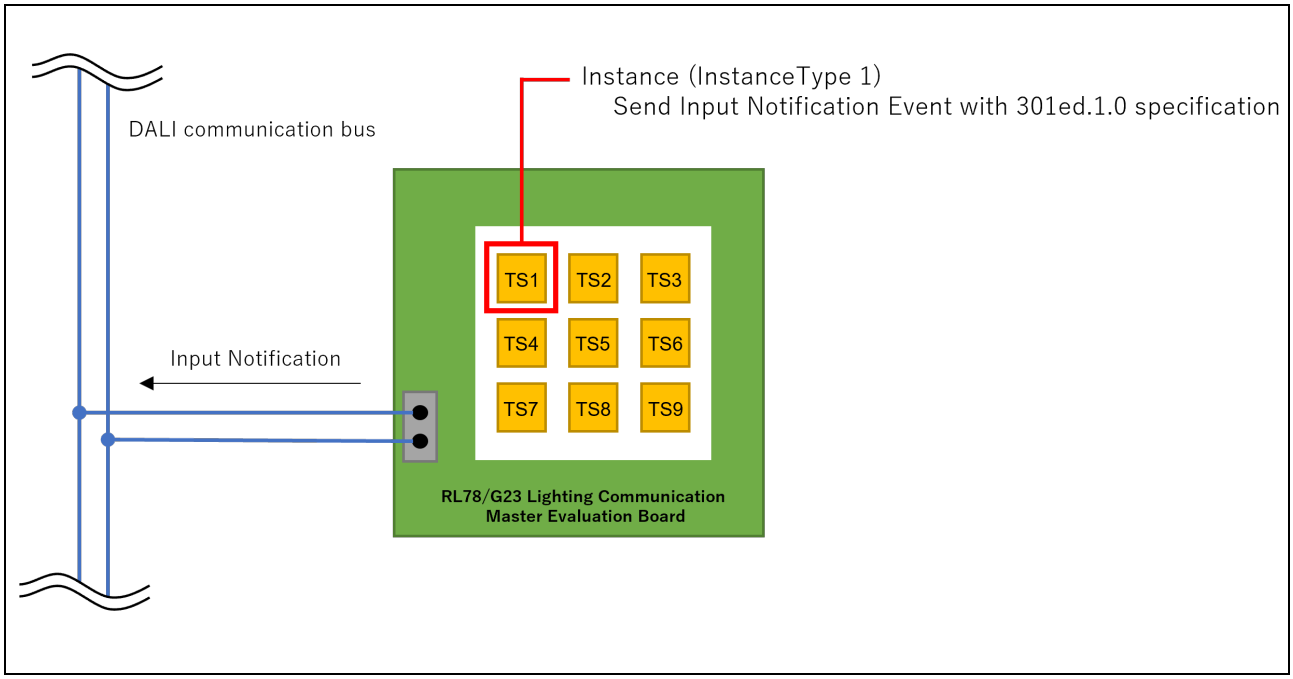
Contents

1. Overview

This application note describes a sample application that implements the Input Device function. The Input Device in this sample application has one instance of Instance Type 1 (Push Button), which is linked to the TS1 touch key as a physical signal processor. The touch key TS1 is pressed as an input, and an Input Notification Event conforming to the 301ed.1.0 specification is issued.

The Input Notification Event is a Forward Frame to notify the status of the signal processor. If you wish to dim the Control Gear on the DALI subnet, implement an application that sends a 16-bit Forward Frame corresponding to the Input Notification Event.

Figure 1-1 Overview of Input Device sample application operation



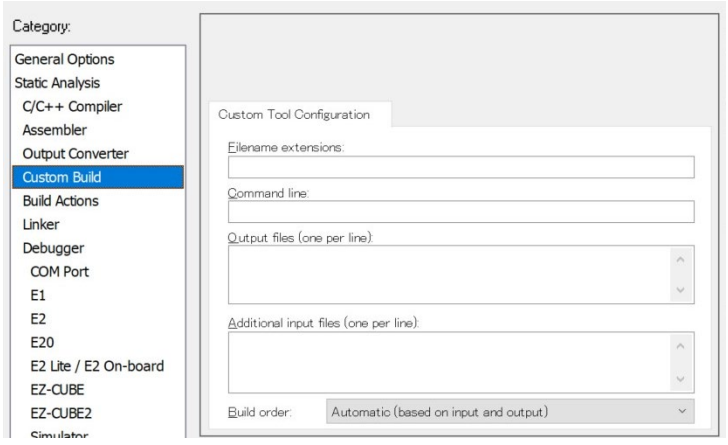
2. Operation confirmation conditions

The sample application has been tested in the following environments

Table 2.1 Operation confirmation conditions

Item	Contents
Microcontroller used	RL78/G23 (R7F100GGG2DFB)
Board used	RL78/G23 Lighting Communication Master Evaluation Board (RTK7RL23LMP00000BJ)
Operating Frequency	High-speed on-chip oscillator clock: 32 MHz
Operating Voltage	3.3V
Maximum current consumption (When DALI command is issued)	31.30mA
Integrated Development Environment (e2 studio)	Made by Renesas Electronics e2 studio 2022-04
C compiler (e2 studio)	Made by Renesas Electronics CC-RL V1.11.00
Integrated Development Environment (IAR)	Made by IAR Systems IAR Embedded Workbench IDE V4.21.4(NOTE)
C compiler (IAR)	Made by IAR Systems IAR C/C++ Compiler for Renesas RL78 V4.21.4
Library	Made by Renesas Electronics Renesas Flash Driver RL78 Type 01 V1.00 Made by Renesas Electronics EEPROM emulation software RL78 Type01 V1.00 Made by Renesas Electronics DALI103i library Gen2 V1.00 Made by Renesas Electronics DALI301 library Gen2 V1.00
Smart Configurator (SC)	V1.3.0 [Components used] Board Support Package (BSP) V1.20 SNOOZE mode sequencer (SMS) V1.0.1 Capacitive Sensing Unit driver (CTSU) V1.20 Touch middleware (TOUCH) V1.20

NOTE : When use IAR Embedded Workbench IDE V5 series, please clear all settings on "Custom tool configuration" . Please click Project - Option - Custom Build to open the custom tool configuration. Please clear below items : Filename extensions, Command line, Output files, Additional input files.



3. Hardware Description

3.1 System Configuration

An example system configuration is shown below.

The sample application operates Input Device corresponding to RL78/G23 Lighting Communication Master Evaluation Board. The Input Device is a type of master device in the DALI system that notifies the DALI system of information from signal processing devices (switches, sensors, etc.) acquired by the Input Device. A separate Application Controller and Control Gear are required to configure a DALI system that uses the Input Device.

In this application note, the RX65N Cloud Kit+DALI-2 option board or RL78/G23 Lighting Communication Master Evaluation Board is used as the Application Controller. EZ-0012+EZ-0012 expansion board is used as the Control Gear.

Figure 3-1 System configuration example (RX65N Cloud Kit + DALI-2 option board)

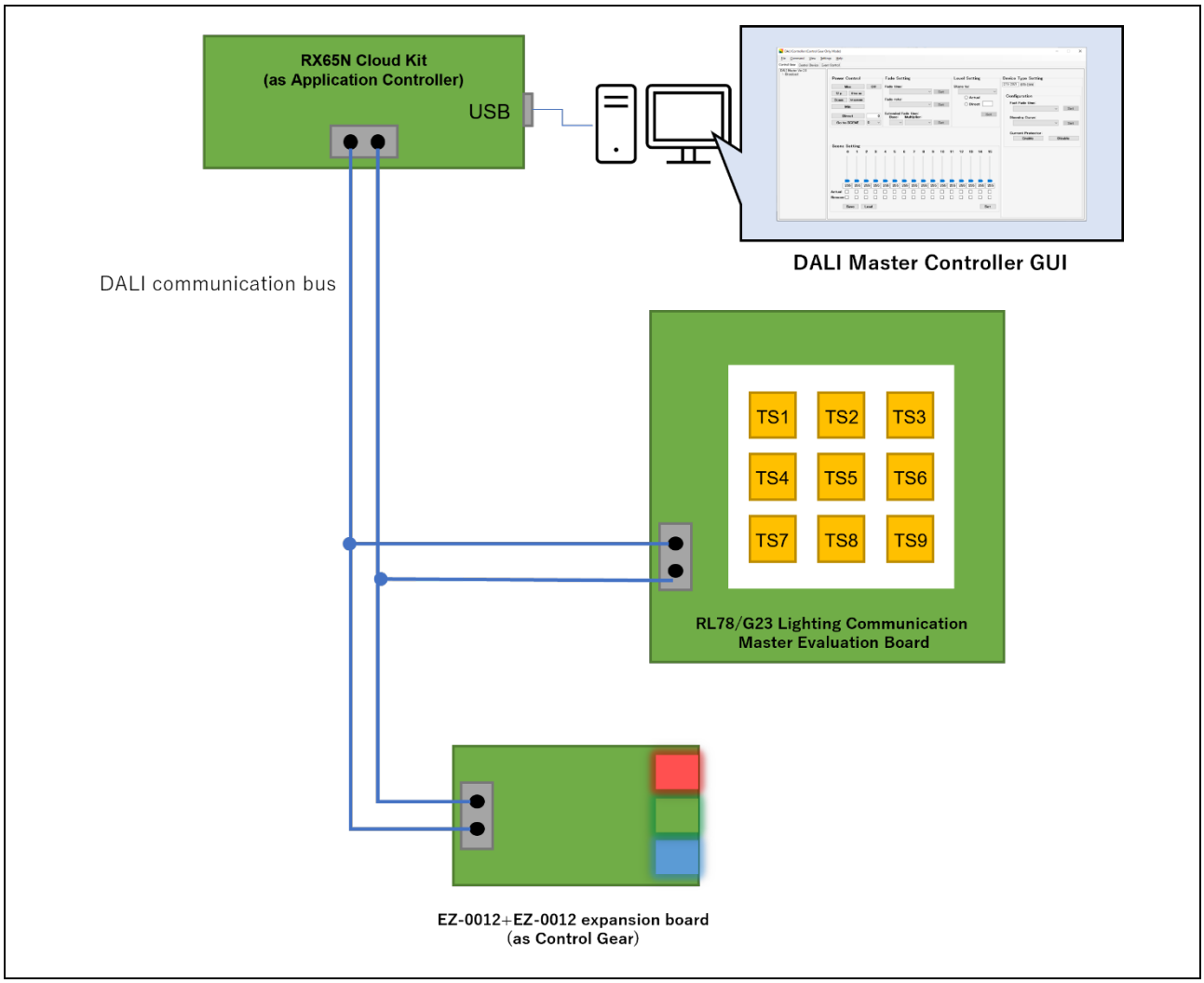
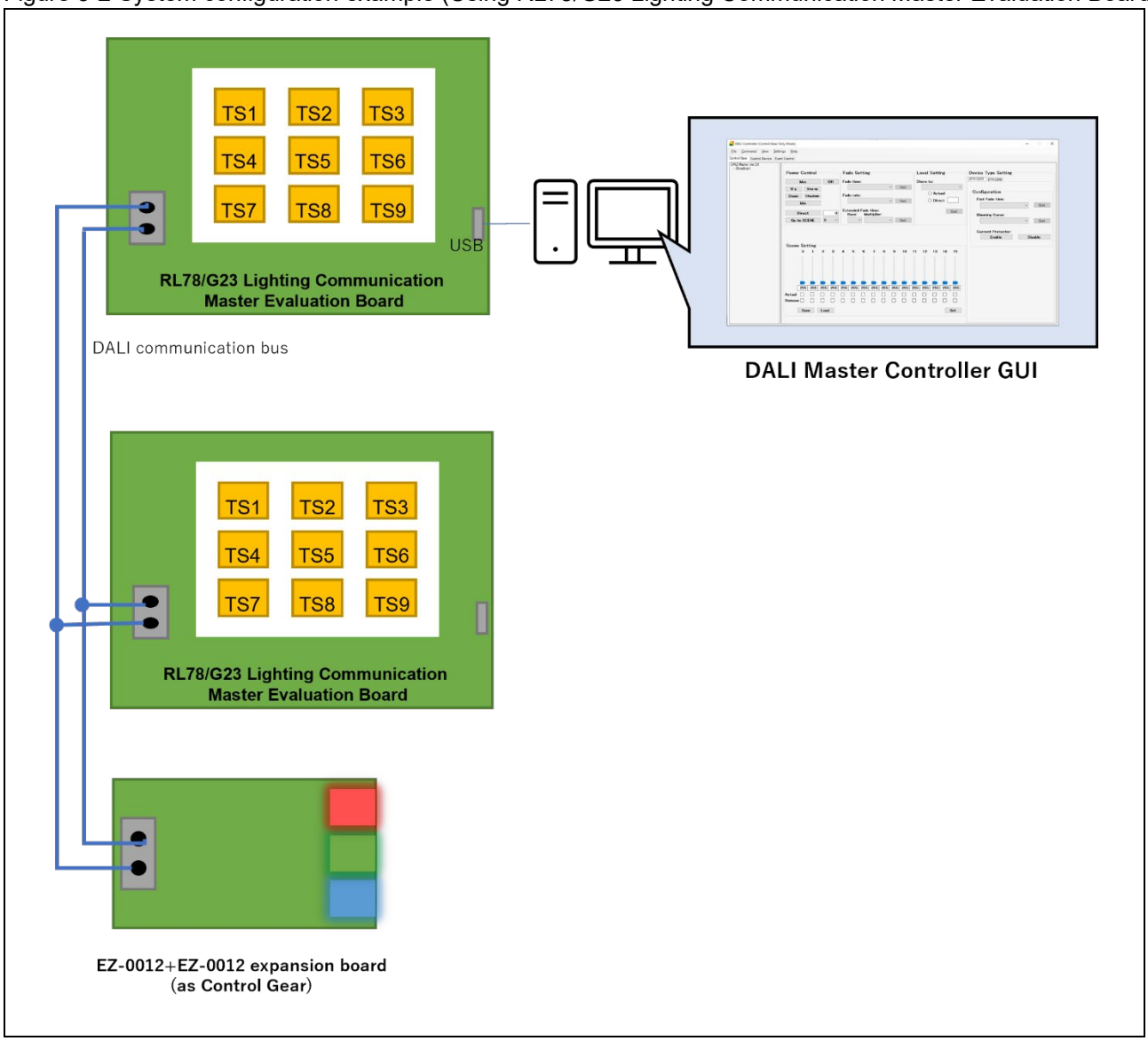


Figure 3-2 System configuration example (Using RL78/G23 Lighting Communication Master Evaluation Board)



3.2 List of pins used

The pins and features used in the sample application are shown below.

Table 3.1 Pins used and features

Pin name	Input/Output	Contents
P10	Output	DALI Communication (Transmit)
P16/TI01	Input	DALI communication (Receive)
P31/TS01	Output	Touch key scan output 1
P71/TS03	Input	Touch key scan input 1

For other features related to the RL78/G23 Lighting Communication Master Evaluation Board, refer to the following.

- RL78/G23 Lighting communication master board initial firmware DALI communication modulation by SMS (R01AN6460)
- RL78/G23 Lighting Communication Master Evaluation Board User's Manual (R20UT5072EJ0102)

4. Software Description

4.1 How to build the environment (CC-RL)

4.1.1 Download software

The following software is required to run the sample application.

Please download it from the Renesas Electronics website.

- e2 studio
- Renesas Flash Driver RL78 Type 01 for RL78/G23
- EEPROM Emulation Software for RL78/G23 RL78 Type01
- DALI Master Controller GUI V3.00 or later

4.1.2 Install e2 studio

The e2 studio is required for program development and on-chip debugging of the sample applications.

For details on installation and other basic operations, refer to the User's Manual below.

- Integrated Development Environment e2 studio 2020-04, e2 studio v7.8 User’s Manual: Getting Started Guide (R20UT4819)

4.1.3 Install Renesas Flash Driver RL78 Type 01 for RL78/G23

The sample application does not include the Renesas Flash Driver. Therefore, it is necessary to download the file from the Renesas Electronics website and register the file.

For installation, refer to the User's Manual below. For file registration, refer to section 4.1.8.

- Renesas Flash Driver RL78 Type01 User's Manual for RL78/G23 (R20UT4830)

4.1.4 Install EEPROM Emulation Software for RL78/G23 RL78 Type01

The sample application does not include EEPROM emulation software. Therefore, it is necessary to download the file from the Renesas Electronics website and register the file.

For installation, refer to the User's Manual below. For file registration, refer to section 4.1.8.

- EEPROM Emulation Software RL78 Type01 for RL78/G23 User's Manual (R20UT5008)

4.1.5 Install the DALI Master Controller GUI

The DALI Master Controller GUI is required for evaluation of sample applications.

Refer to the User's Manual below for the installation procedure.

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

The firmware must also be written to the application controller that communicates with the DALI master controller GUI. For the target firmware, use the hex and mot files enclosed in the ZIP file provided when downloading the DALI master controller GUI installer.

4.1.6 Write firmware for the EZ-0012+EZ-0012 expansion board

The firmware needs to be written as a Control Gear to check the operation of the sample application. Refer to the following for the procedure.

- RL78/I1A DALI-2 Control Gear Basic (102) LED (207) Color Control (209Tc) Sample Application (R01AN6177)

4.1.7 Import Sample Applications

1. Unzip the provided project file and place it in the desired location (folder).
2. Start e2 studio and right-click in the Project Explorer or select the [File] → [Import].
3. Select [General] → [Existing Projects into Workspace] and click [Next].
4. Select "Select root directory" and from [Browse] select the project file placed in step 1.
5. Select a sample application.
6. Click [Exit].

4.1.8 Locate Renesas Flash Driver / EEPROM Emulation Software

After importing the sample application, store the Renesas Flash Driver / EEPROM emulation software downloaded from Renesas Electronics in the following directory.

Table 4.1 Location of Renesas Flash Driver / EEPROM emulation software

\DALI103i_301_sample<DIR>	
\Library<DIR>	
\RFD<DIR>	Folder that contains the Renesas Flash Driver.
\userown	Setup for this sample application
r_rfd_common_userown.c	
\include	Copy from RFD.
\rfd	
r_rfd.h	
r_rfd_compiler.h	
r_rfd_device.h	
r_rfd_memmap.h	
r_rfd_types.h	
r_typedefs.h	
r_rfd_common_api.h	
r_rfd_common_control_api.h	
r_rfd_common_userown.h	
r_rfd_data_flash_api.h	
\source	
\commom	
r_rfd_common_api.c	
r_rfd_common_control_api.c	
\dataflash	
r_rfd_data_flash_api.c	
\EES<DIR>	Folder for storing EEPROM emulation software.
\userown	Setup for this sample application.
\include	
r_ees_descriptor.h	
r_ees_user_types.h	
r_ees_descriptor.c	
\include	Copy from EES.
\ees	
r_ees.h	
r_ees_compiler.h	
r_ees_defines.h	
r_ees_device.h	
r_ees_memmap.h	
r_ees_types.h	
r_typedefs.h	
r_ees_api.h	
r_ees_exrfd_api.h	
r_ees_sub_api.h	
\source	
\ees	
r_ees_api.c	
r_ees_exrfd_api.c	
r_ees_sub_api.c	

4.1.9 How to build the sample application

1. Right click on the project and select [Build Project].
2. Check the [Console] pane shows ‘Build complete.’ message to indicate a successful build.

4.1.10 Hardware connections and configuration switches for the sample application

The jumpers, etc. on the RL78/G23 Lighting Communication Master Evaluation Board must be set as follows when operating this sample application.

Figure 4-1 RL78/G23 Lighting Communication Master Evaluation Board Components

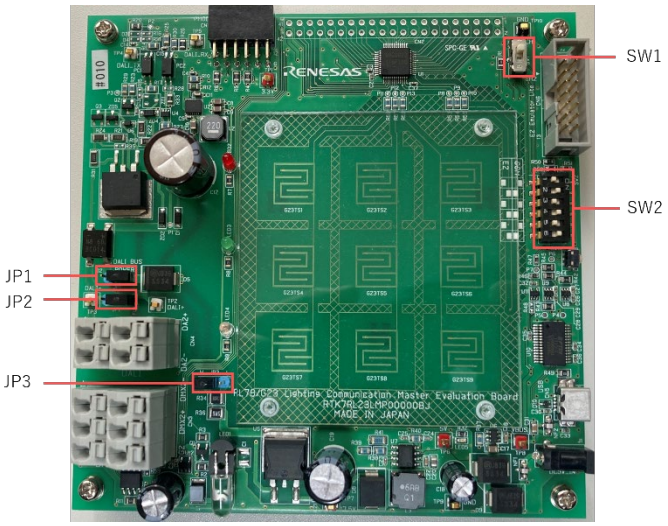


Table 4.2 List of switch settings

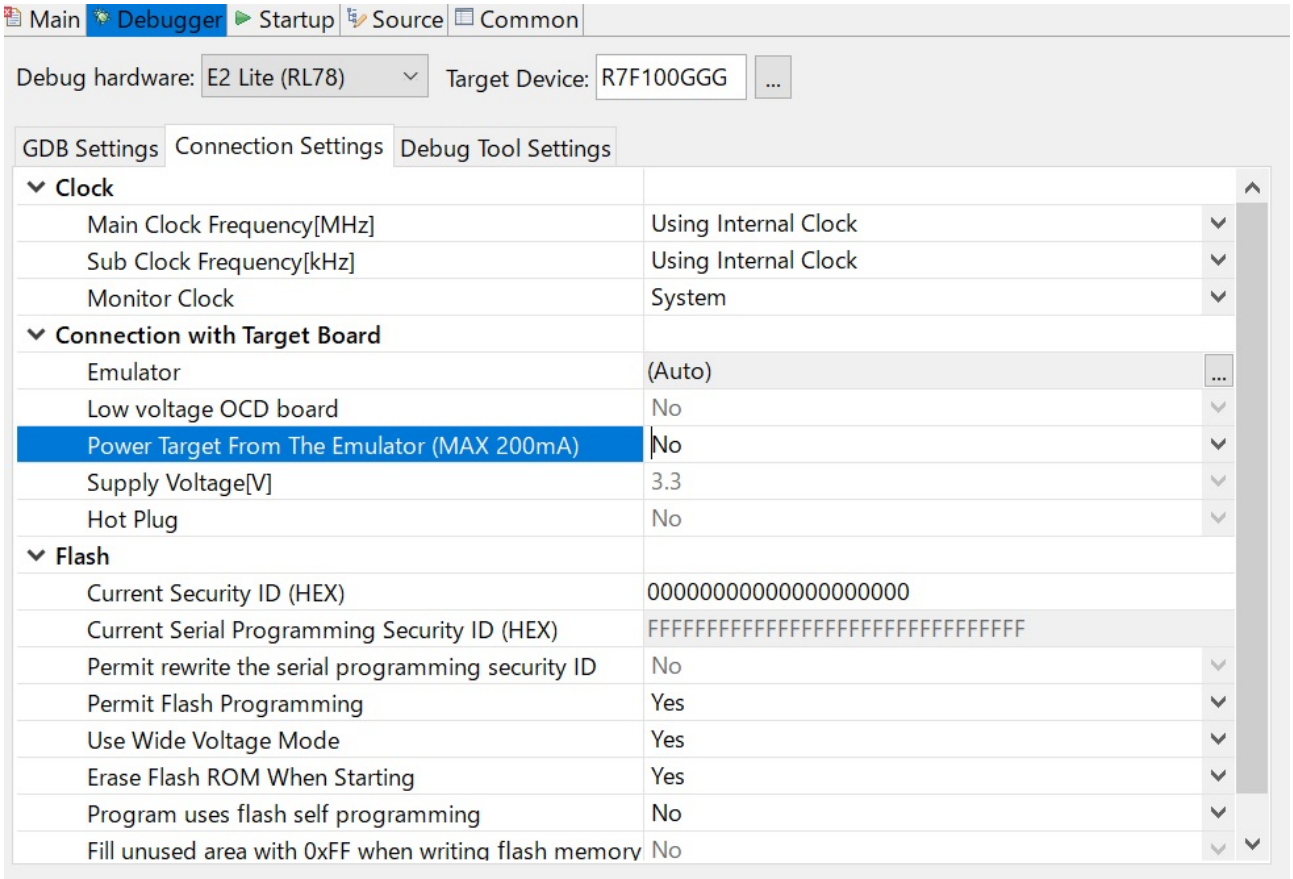
JP/SW Number	Setting
JP1	Open
JP2	Open
JP3	H side and short
SW1	don’t care
SW2-1	OFF
SW2-2	ON
SW2-3	OFF
SW2-4	ON
SW2-5	OFF
SW2-6	ON

4.1.11 How to debug the sample application

1. After importing the sample application in e2 studio, click the  button to download the program to the microcontroller.
2. Click [Run] → [Debug Configurations…] to open the “Debug Configurations” window.
3. In the "Debug Configurations" window, expand the "Renesas GDB Hardware Debugging" debug configuration view and click on an existing debug configuration.
4. Switch to the [Debugger] → [Connection Settings] tab and confirm that the settings are as shown below.

   * When using E2 emulator Lite, select Debug hardware : E2 Lite (RL78).

Figure 4-2 Debug Screen Settings



5. Select "Start Debugging" and when the "Debug" view screen appears, ready for debugging.

For details on debugging and other basic operations, please refer to the User's Manual below.

- Integrated Development Environment e2 studio 2020-04, e2 studio v7.8 User’s Manual: Getting Started Guide (R20UT4819)

4.2 Environment construction (IAR)

4.2.1 Download Software

The following software is required to run the sample application.

Download from the IAR Systems website.

4.2.2 IAR Embedded Workbench for Renesas RL78

- IAR Embedded Workbench for Renesas RL78 is required for program development and on-chip debugging of the sample applications.
  Refer to the video below for installation.

  Using Smart Configurator with IAR Embedded Workbench for RL78 (1/2) - Installation

4.2.3 Install the DALI Master Controller GUI

The DALI Master Controller GUI is required for evaluation of sample applications.

Refer to the User's Manual below for the installation procedure.

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

The firmware must also be written to the application controller that communicates with the DALI master controller GUI. For the target firmware, use the hex and mot files enclosed in the ZIP file provided when downloading the DALI master controller GUI installer.

4.2.4 Write firmware for the EZ-0012+EZ-0012 expansion board

The firmware needs to be written as a Control Gear to check the operation of the sample application. Refer to the following for the procedure.

- RL78/I1A DALI-2 Control Gear Basic (102) LED (207) Color Control (209Tc) Sample Application (R01AN6177)

4.2.5 Import Sample Applications

1. Start IAR Embedded Workbench for Renesas RL78.
2. Select [File] -> [New Workspace].
3. Select [File] -> [Save Workspace As] to place the workspace in any name and location (folder).
4. Unzip the provided project file and place it in the workspace (folder) created.
5. Select [Project] -> [Add Existing Project] and select the project file (EWP file) placed in step 9.
6. Select [File] -> [Exit].

4.2.6 Locate Renesas Flash Driver / EEPROM emulation software

After importing the sample application, store the Renesas Flash Driver / EEPROM emulation software downloaded from Renesas Electronics in the following directory.

Table 4.3 Location of Renesas Flash Driver / EEPROM emulation software

\DALI103i_301_sample<DIR>	
\Library<DIR>	
\RFD<DIR>	Folder that contains the Renesas Flash Driver.
\userown	Setup for this sample application
r_rfd_common_userown.c	
\include	Copy from RFD.
\rfd	
r_rfd.h	
r_rfd_compiler.h	
r_rfd_device.h	
r_rfd_memmap.h	
r_rfd_types.h	
r_typedefs.h	
r_rfd_common_api.h	
r_rfd_common_control_api.h	
r_rfd_common_userown.h	
r_rfd_data_flash_api.h	
\source	
\commom	
r_rfd_common_api.c	
r_rfd_common_control_api.c	
\dataflash	
r_rfd_data_flash_api.c	
\EES<DIR>	Folder for storing EEPROM emulation software.
\userown	Setup for this sample application.
\include	
r_ees_descriptor.h	
r_ees_user_types.h	
r_ees_descriptor.c	
\include	Copy from EES.
\ees	
r_ees.h	
r_ees_compiler.h	
r_ees_defines.h	
r_ees_device.h	
r_ees_memmap.h	
r_ees_types.h	
r_typedefs.h	
r_ees_api.h	
r_ees_exrfd_api.h	
r_ees_sub_api.h	
\source	
\ees	
r_ees_api.c	
r_ees_exrfd_api.c	
r_ees_sub_api.c	

4.2.7 How to build the sample application

1. Right-click on the project in the workspace and select [Make].
2. The build will start and the "console" will display the status of the build. The build is complete when the message "Total number of errors: 0" is displayed.

4.2.8 Hardware connections and configuration switches for the sample application

The jumpers, etc. on the RL78/G23 Lighting Communication Master Evaluation Board must be set as follows when operating this sample application.

Figure 4-3 RL78/G23 Lighting Communication Master Evaluation Board Components

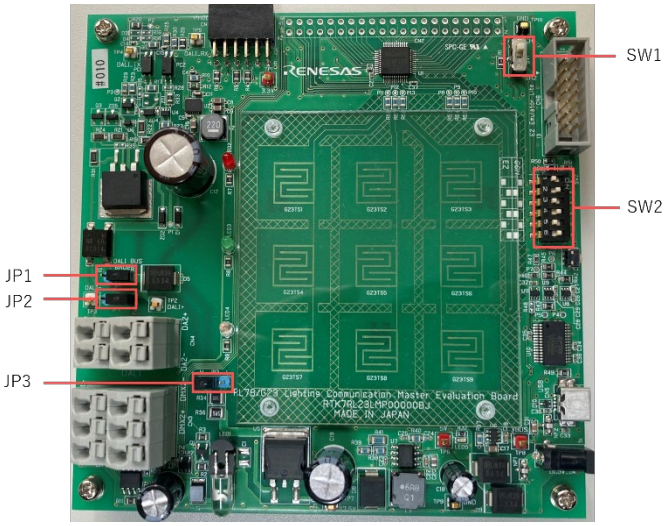


Table 4.4 List of switch settings

JP/SW Number	Setting
JP1	Open
JP2	Open
JP3	H side and short
SW1	don’t care
SW2-1	OFF
SW2-2	ON
SW2-3	OFF
SW2-4	ON
SW2-5	OFF
SW2-6	ON

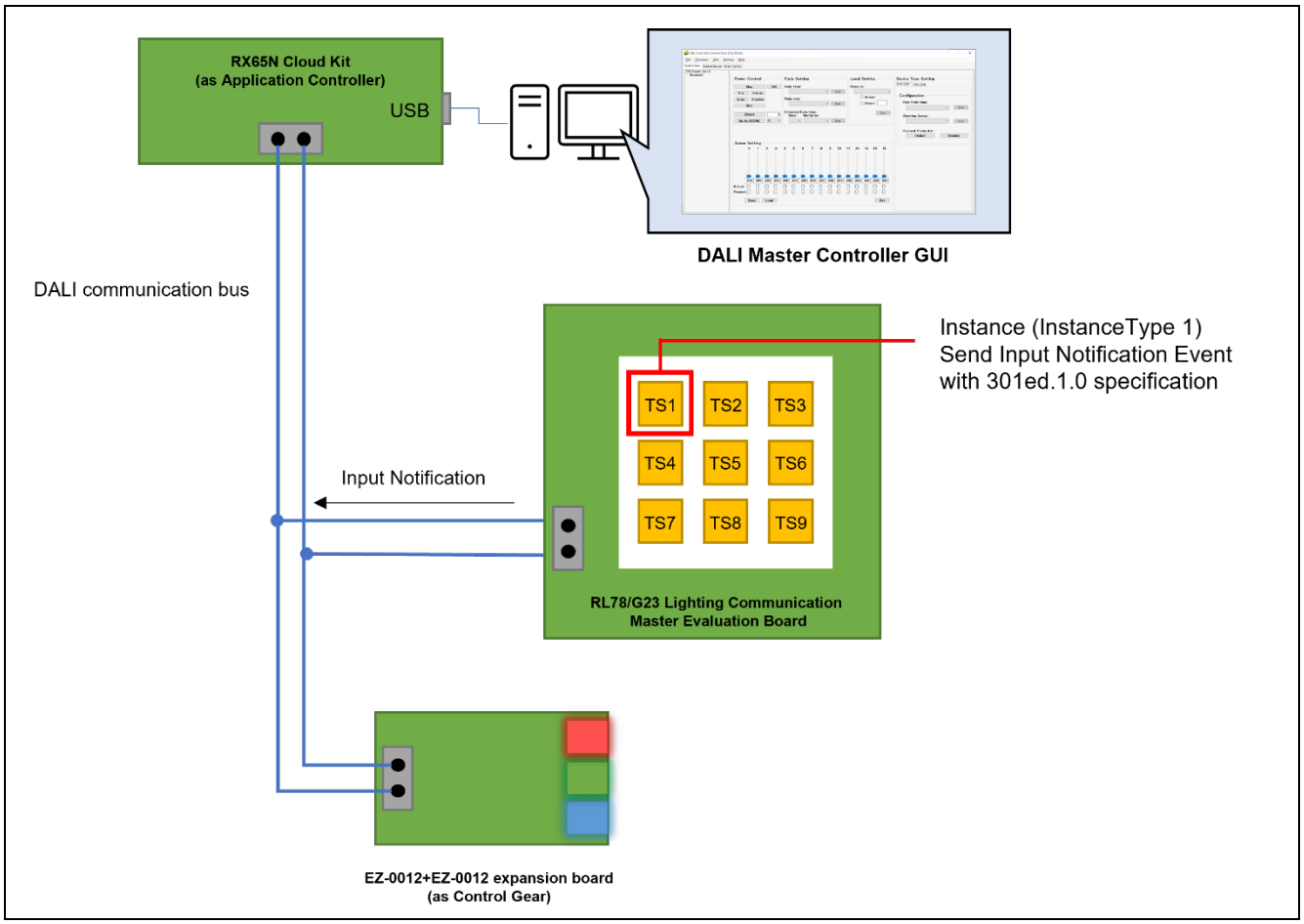
4.2.9 How to debug the sample application

1. After importing the sample application in IAR Embedded Workbench for Renesas RL78, click the  button to download the program to the microcontroller and debug it.

4.3 Overview of Operation

The following is an overview of the operation of this sample application.

Figure 4-4 System configuration example (using RX65N Cloud kit + DALI-2 option board)



This sample application operates as an "Input Device" equipment in the DALI standard. The Input Device in this sample application has one Instance of “Instance Type 1”, which is associated with a touch key and sends an Input Notification Event to the DALI communication bus according to the press state of the touch key. The Instance - Touch Key correspondence table is shown in Table 4.5.

Table 4.5 Instance - Touch Key correspondence table

Instance Number	resolution (bit)	Input Signal	Corresponding touch key
0	1	0 : Not pressed, 1: Pressed	TS1

In addition, the following Event Information is defined for Instance Type 1 Instances, and an Input Notification Event occurs when each of the conditions is met. Refer to the IEC62386-301ed1.0 standard for the conditions under which each Event occurs.

Table 4.6 List of Event Information

Event Information		Description
Event Details	Event Info value (binary value)	
Button released	00 0000 0000	Button released
Button pressed	00 0000 0001	Button pressed
Short press	00 0000 0010	Button shortly pressed
Double press	00 0000 0101	Button pressed twice
Long press start	00 0000 1001	Button long press started
Long press repeat	00 0000 1011	Press and hold button is in progress
Long press stop	00 0000 1100	Press and hold button is stopped
Button free	00 0000 1110	Button is released from sticking
Button stuck	00 0000 1111	Button stuck

The processing corresponding to the Event Message sent by the Input Device in this sample application (such as sending dimming instruction commands to the Control Gear) should be implemented in the Application Controller on the same DALI subnet.

In 4.4 Operating Procedure, four events (Short Press, Double Press, Long Press Start, and Long Press Repeat) are issued to realize an operation that associates the dimming operation of Control Gear with the issued events.

The conditions for the occurrence of each event are as follows.

In addition, this is just an example about events to be generated and commands to be associated with them. It is necessary to set events and associate commands according to the behavior to be realized by users.

Figure 4-5 Condition for occurrence of Short Press event

■Condition for occurrence of Short Press event
It occurs when the Push Button is pressed for a time shorter than T_{short}[ms] and then not pressed for T_{double}[ms].

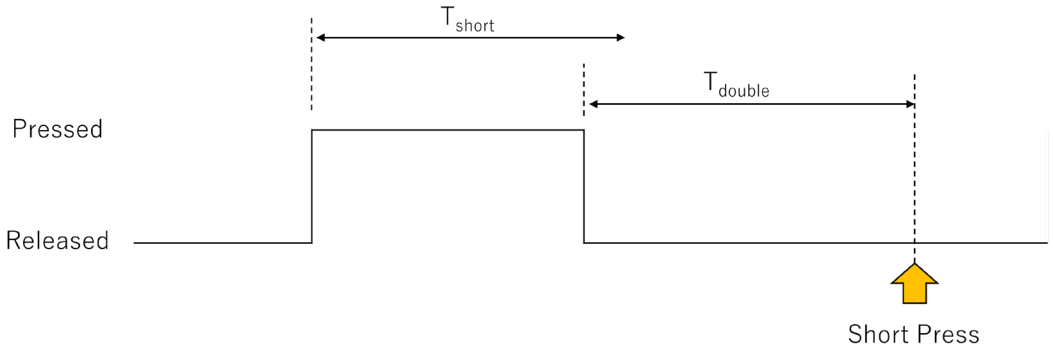


Figure 4-6 Condition for occurrence of Double Press event

■Condition for occurrence of Double Press event
It occurs when the Push Button is pressed for a time shorter than T_{short}[ms] and then pressed again within a time shorter than T_{double}[ms].

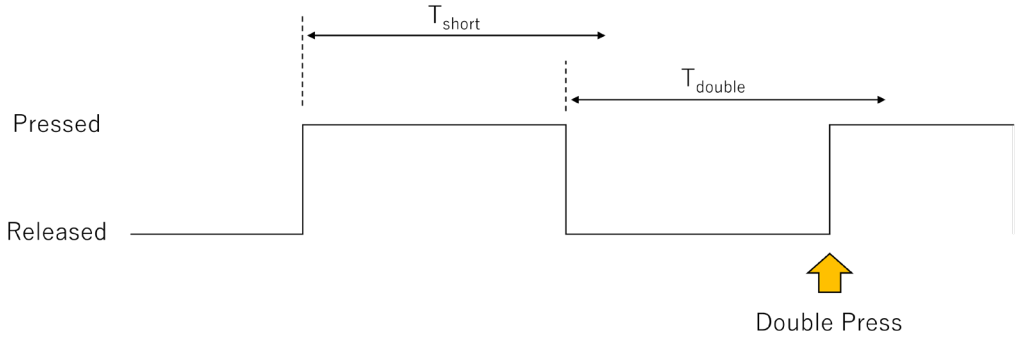


Figure 4-7 Condition for occurrence of Long Press Start event

■Condition for occurrence of Long Press Start event
It occurs when the Push Button is held down for a T_{short}[ms] time.

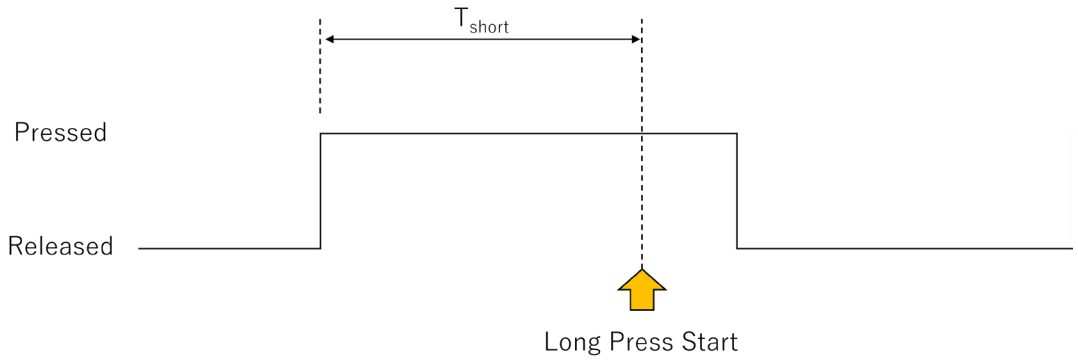
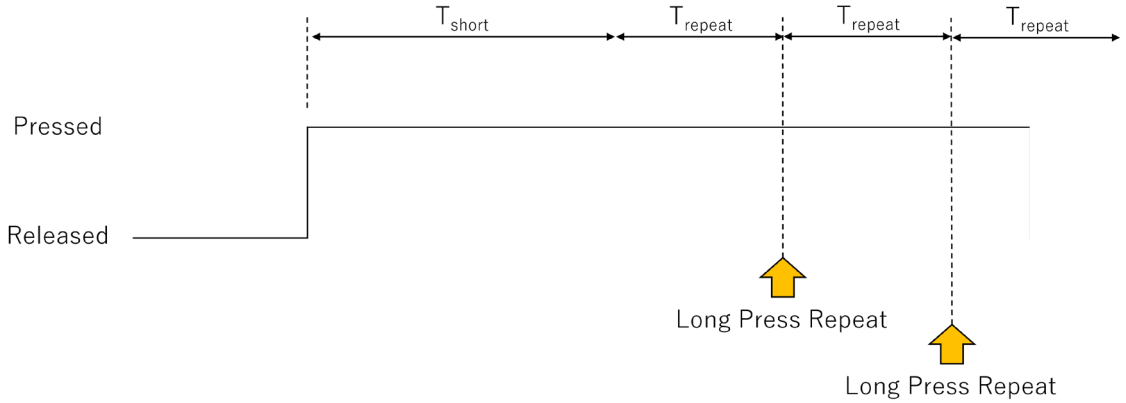


Figure 4-8 Condition for occurrence of Long Press Repeat event

■Condition for occurrence of Long Press Repeat event
It occurs every T_{repeat}[ms] if the Push Button continues to be pressed after T_{short}[ms].



In this case, the T_{short}, T_{double}, and T_{repeat} times are set as follows.

Table 4.7 Each setting time

Item	Set value of variable	T_{incr}	Set time
T_{short}	50	20 [ms]	1,000 [ms]
T_{double}	25	20 [ms]	500 [ms]
T_{repeat}	10	20 [ms]	200 [ms]

In addition, the commands associated with each occurrence event are as follows

Table 4.8 Occurrence Event - Related Command correspondence table

Occurrence event	Event information	Associated command (16bit Forward Frame)
Short Press	0b 00 0000 0010	GO TO LAST ACTIVE LEVEL (0xFF0A)
Double Press	0b 00 0000 0101	OFF (0xFF00)
Long Press Start	0b 00 0000 1001	RECALL MIN LEVEL (0xFF06)
Long Press Repeat	0b 00 0000 1011	UP (0xFF01)

This will cause the Control Gear dimming to perform the following operations when a user operation is performed.

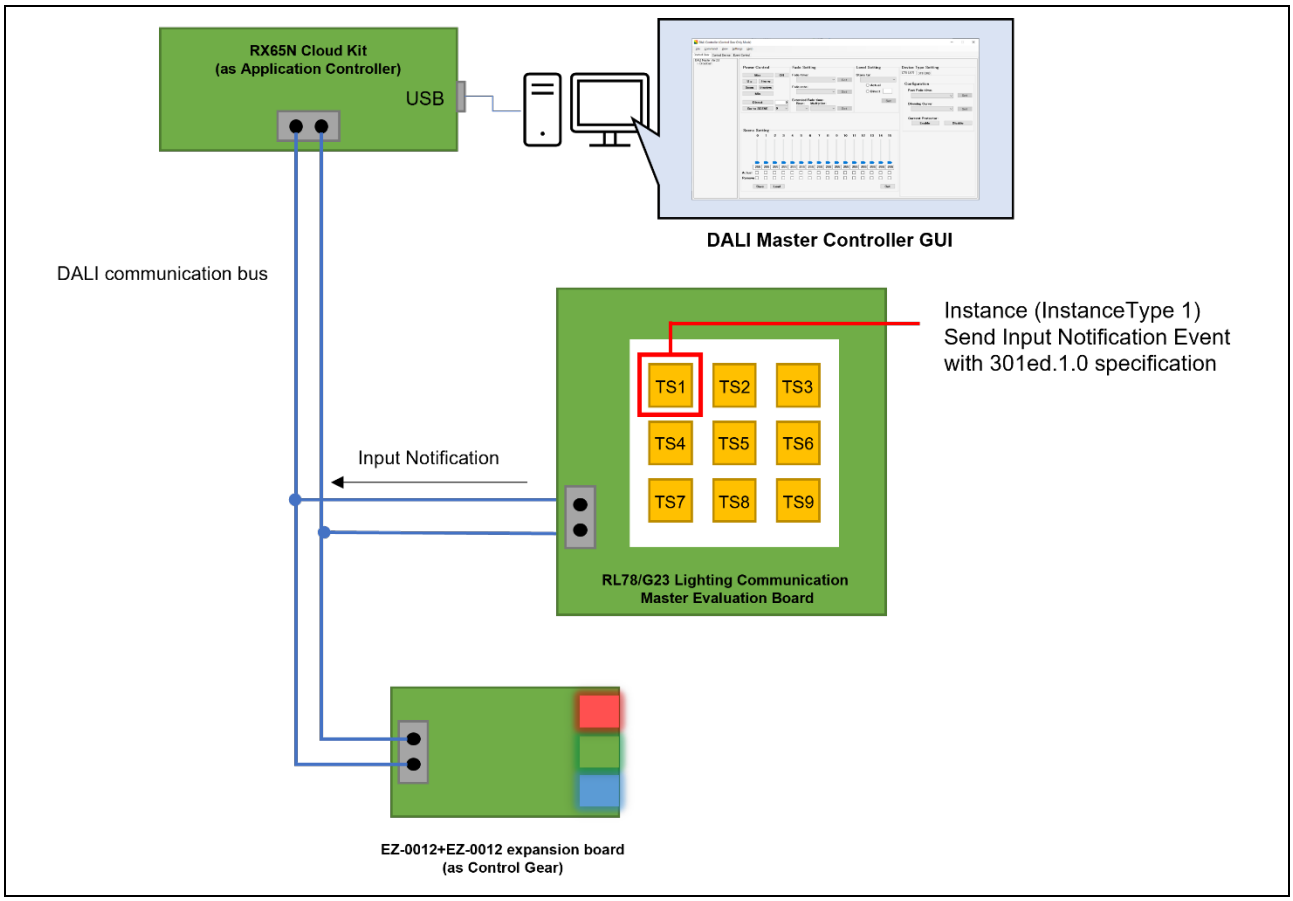
- Short press action: light on at last active level
- Twice push action: light off
- Long press: Fade operation from min level to max level during long press

4.4 Operating Procedure

This application note describes the procedure for running the sample application using the DALI master controller GUI as follows.

An example of using the RX65N Cloud kit + DALI-2 option board as the Application Controller is shown below.
When using the RL78/G23 Lighting Communication Master Evaluation Board as the Application Controller, refer to 3.1 to connect devices. Connect an AC adapter to the RX65N Cloud kit + DALI-2 Option Board and configure the settings so that the power supply to the DALI bus is sufficient.

Figure 4-9 System configuration example (using RX65N Cloud kit + DALI-2 option board)



[Procedure]

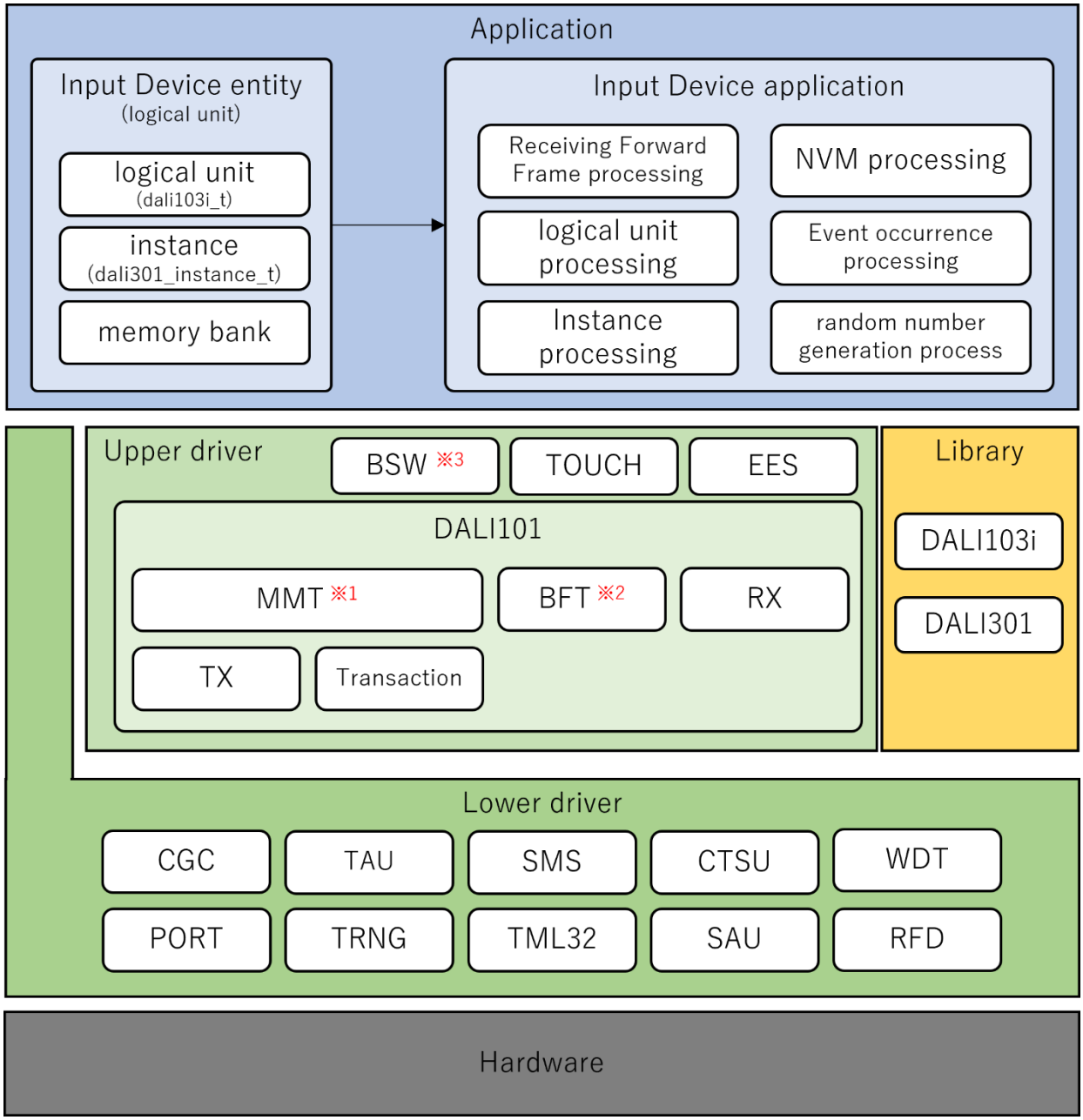
1. After connecting RX65N Cloud kit + DALI-2 option board, RL78/G23 Lighting Communication Master Evaluation Board, EZ-0012 + EZ-0012 expansion board as the Application Controller, and then supply power to each device.
2. After building the sample application, download the program from the debugger to the RL78/G23 Lighting Communication Master Evaluation Board and execute debugging.
3. After starting the DALI master controller GUI, make a serial connection to the RX65N Cloud kit + DALI-2 option board.
4. Assign a short address, group number, and instance group number to the RL78/G23 Lighting Communication Master Evaluation Board from the Control Device tab of the DALI master controller GUI.
5. Open the Manual Command (By Code) screen and select the Control Device tab.
6. Send RESET (0xFFFE10 to Twice) with Manual Command (By Code).
7. Send DTR0 (0xC13032 is sent Once) with Manual Command (By Code).
8. Send SET SHORT TIMER (Twice send 0xFFFF00) with Manual Command (By Code).
9. Send DTR0 (Once send 0xC13019) with Manual Command (By Code).
10. Send SET DOUBLE TIMER (Twice send 0xFFFF01) with Manual Command (By Code).
11. Send DTR0 (Once send 0xC1300A) with Manual Command (By Code).
12. Send SET REPEAT TIMER (Twice send 0xFFFFFF02) with Manual Command (By Code).
13. Send DTR0 (Once send 0xC1303C) with Manual Command (By Code).
14. Send SET EVENT FILTER (Twice send 0xFFFFFF68) with Manual Command (By Code).
15. Select the Event Control tab of the DALI master controller GUI.
16. Short press (less than 1000 [ms]) "TS1" on the RL78/G23 Lighting Communication Master Evaluation Board.
17. An event message is automatically registered in the "Receive Event" tab of the Event Control tab of the DALI master controller GUI. Click the "Setting" button and set GO TO LAST ACTIVE LEVEL (0xFF0A) in Broadcast as the frame for sending the received event message.
18. Press "TS1" on the RL78/G23 Lighting Communication Master Evaluation Board twice (again within 500 [ms]).
19. An event message is automatically registered in the "Receive Event" of the Event Control tab of the DALI master controller GUI. Click the "Setting" button and set OFF (0xFF00) in Broadcast as the frame to be sent for the received event message.
20. Press and hold down "TS1" on the RL78/G23 Lighting Communication Master Evaluation Board (for about 2000 [ms]).
21. An event message is automatically registered in the "Receive Event" of the Event Control tab of the DALI master controller GUI. Click the "Setting" button and set RECALL MIN LEVEL (0xFF06) in Broadcast as the frame to send in response to the received Long Press Start event message and UP (0xFF01) in Broadcast as the frame to send in response to the Long Press Repeat event message.
22. Turn on the Event Control tab of the DALI master controller GUI, all Enable checkboxes.
23. Operate "TS1" on the RL78/G23 Lighting Communication Master Evaluation Board (short press, press twice, long press) and confirm that dimming control for Control Gear is available.

4.5 Feature Overview

4.5.1 Feature block diagram

A feature block diagram of the sample application is shown below.

Figure 4-10 Feature block diagram



*1 MMT ・・・ Multi Master Transmitter

*2 BFT ・・・ Backward Frame Transmitter

*3 BSW ・・・ Binary Switch

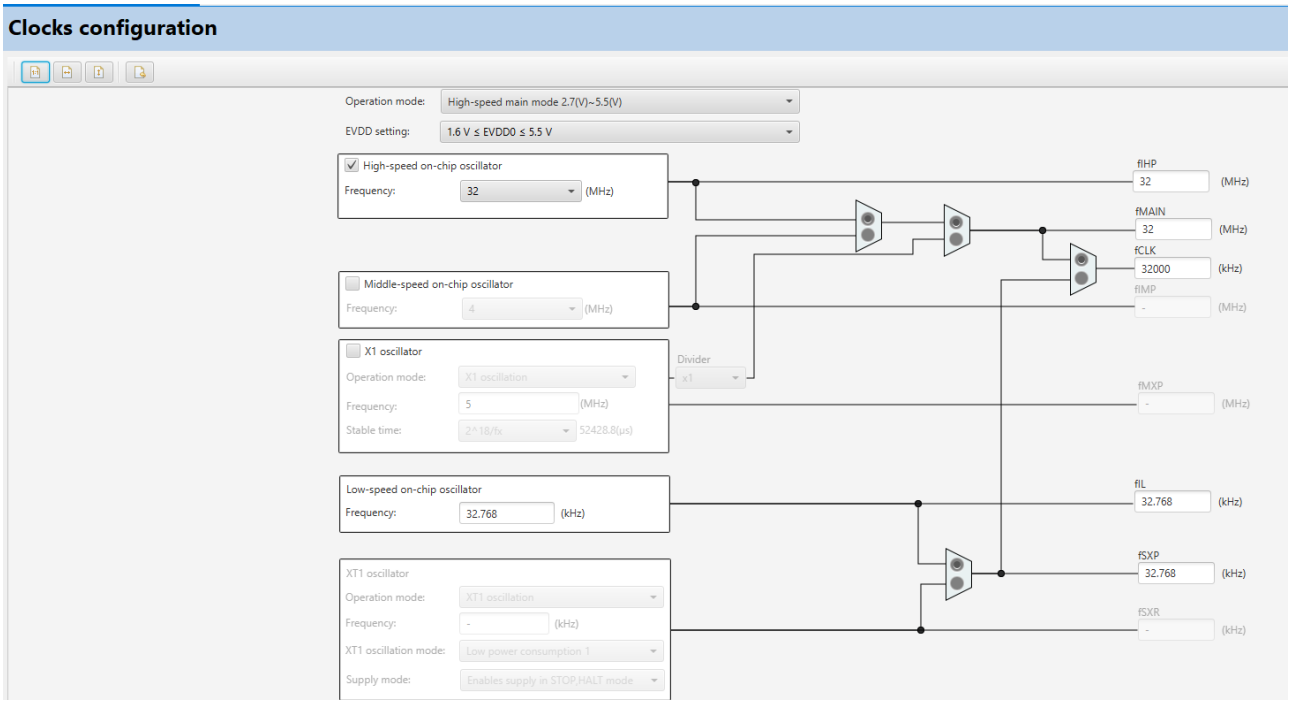
4.5.2 Lower driver layer

It is a simple driver layer using the RL78/G23's peripheral features.

4.5.2.1 CGC

This driver performs clock generation. This sample application uses the Smart Configurator (SC) and has the following settings.

Figure 4-11 clock setting



4.5.2.2 PORT

This driver enables port control of the RL78/G23 microcontroller. In this sample application, a driver that enables control of all ports of the on-board microcontroller is implemented and called as required.

Table 4.9 Port settings (1/2)

Contents	Input/Output	Usage
P01/RxD1	Input	For DMX512
P00/TxD1	Output	For DMX512
P10	Output	For DALI transmission
P11	Input	For debugger connection
P12	Output	For debugger connection
P13	Output	Unused
P14	Output	Unused
P15	Output	Unused
P16/TI01	Input	For DALI reception (input pulse width measurement)
P17	Output	Unused
P20	Output	DMX512
P21	Output	PMOD
P22	Output	PMOD
P23	Output	PMOD
P24	Output	PMOD
P25	Input	Unused
P26	Output	Unused
P27	Output	Unused
P30	Input	Unused
P31/TS01	Output	Touch key scan output 1
P40/TOOL0	Input	For debugger connection
P41/TO07	Output	Unused
P50/TS00	Input	Touch key scan input 3
P51	Output	Unused
P60/SCLA0	Output	PMOD
P61/SDAA0	Output	PMOD
P62	Output	LED (red)
P63	Output	LED (green)

Table 4.10 Port settings (2/2)

Contents	Input/Output	Usage
P70/TS02	Input	Touch key scan input 2
P71/TS03	Input	Touch key scan input 1
P72/TS04	Output	LED (white)
P73/TS05	Output	Unused
P74/TS06	Output	Touch key scan output 3
P75/TS07	Output	Touch key scan output 2
P120	Output	Unused
P121	Output	Unused
P122	Output	Unused
P123	Input (input-only port)	Unused
P124	Input (input-only port)	Unused
P130	Output (output-only port)	Unused
P137/INTP0	Input (input-only port)	Unused
P140	Output	Unused
P146 注	FOR_PROBIT_LAB macro disabled: Output FOR_PROBIT_LAB macro enabled: Input (internal pull-up resistor enabled)	FOR_PROBIT_LAB macro disabled: Unused FOR_PROBIT_LAB macro enabled: ProbitLab digital switch 1 connection status input
P147	Output	Unused
RESET	-	Unused
REGC	-	Unused
VSS	-	Unused
VDD	-	Unused

Note: The FOR_PROBIT_LAB macro is only required when carrying out a test sequence for DALI-2 certification. When carrying out the test sequence, enable the FOR_PROBIT_LAB macro, connect the P146 and GND pins to ProbitLab's digital switch 1 and carry out the test sequence. If the test sequence is not carried out, it should be disabled to avoid false detection of the Push Switch input status. The FOR_PROBIT_LAB macro is defined in the following locations

e2studio: Open the project properties, select [C/C++ Build] -> [Settings] -> [Tool Settings] -> [Compiler] -> On the [Source] screen, [Macro definition]

IAR : From the project options, on the [C/C++ Compiler] -> [Defined symbols] on the [Preprocessor] screen.

4.5.2.3 TAU

This driver controls timers using a timer array unit. In this sample application, each channel is configured as follows according to the application.

Table 4.11 Timer array unit settings

Channel	Setting	Interrupt priority	Usage
TAU0 CH0	Interval timer (1ms)	Interrupt not used	1ms periodic process synchronisation of applications
TAU0 CH1	Input pulse width measurement	Priority 0	Pulse width measurement of DALI RxD pin
TAU0 CH2	Interval timer (50us)	Priority 1	Idle/Active state continuation time measurement of DALI RxD pins
TAU0 CH3	Interval timer (1458us)	Priority 3	Active state transmission time of Corrupted Backward Frame
TAU0 CH4	Interval timer (optional)	Priority 3	For processing when a collision occurs - Active State transmission time of tBreak (1200-1400us) - Loopback time (50ms) from waveform transmitted from DALI TxD pin to received at DALI RxD pin
TAU0 CH5	Unused	-	-
TAU0 CH6	Unused	-	-
TAU0 CH7	Unused	-	-

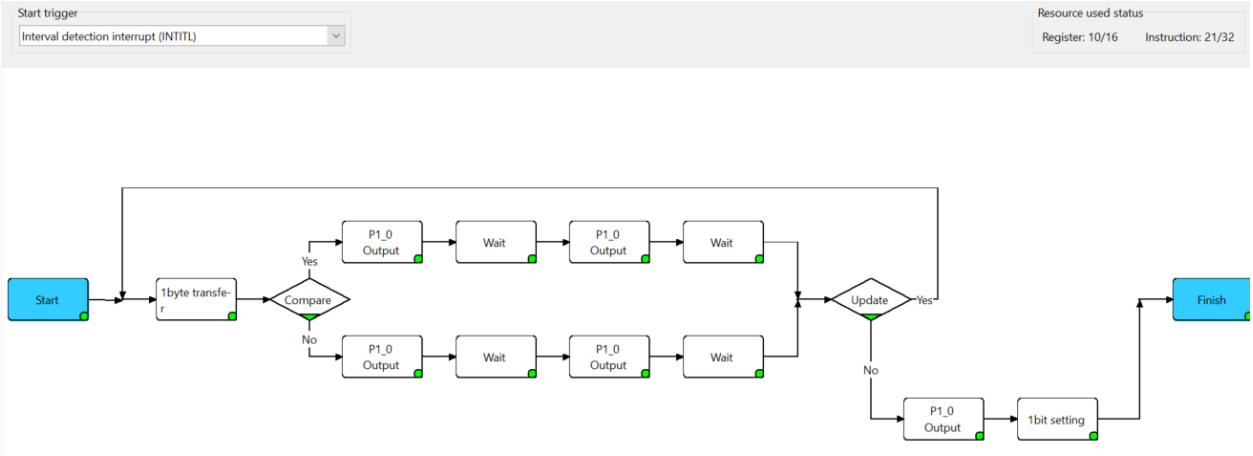
4.5.2.4 TRNG

It is a driver that uses a true random number generator.

4.5.2.5 SMS

This driver uses the SNOOSE mode sequencer (hereafter SMS). In this sample application, the SMS is used to modulate the DALI transmission. The following settings are made using the Smart Configurator (SC).

Figure 4-12 SMS settings



The specifications of the SMS process to be performed with the above settings are given below.

Table 4.12 SMS settings

Overview	DALI waveform output processing by SMS
Description	(1) The SMS is activated by an INTITL interrupt. (2) The waveform is transmitted according to the data by judging the 1-bit data from the argument. ・ Bit data == 0: Output to P10 (DALI TxD pin) in the order of 1 → 0 in the Wait time width. ・ Bit data == 1: Output to P10 (DALI TxD pin) in the order of 0 → 1 in the wait time width. (3) When all data transmission is completed, the output of port P10 is set to High , MK0H bit4 (SMSEMK) is set to Zero generating an SMS termination interrupt (INTSMSE). (4) The finish function terminates SMS process.
Argument[Note1]	address_d: Transmission data start address address_e: Transmission data end address
Return value	None
Remark	The bit width of the waveform output with reference to the DALI TxD pin varies with the rise or fall delay in the DALI communication circuit external to the microcontroller; adjustment is required so that the waveform output on the DALI communication bus complies with the standard. For details, refer to the RL78/G23 DALI-2 Input Device Basic (103) Sample Application Notes in Section 4.5.6.2.

4.5.2.6 TML32

This driver uses a 32-bit interval timer. In this sample application, each channel is configured as a 4-channel 8-bit counter mode as follows according to the application.

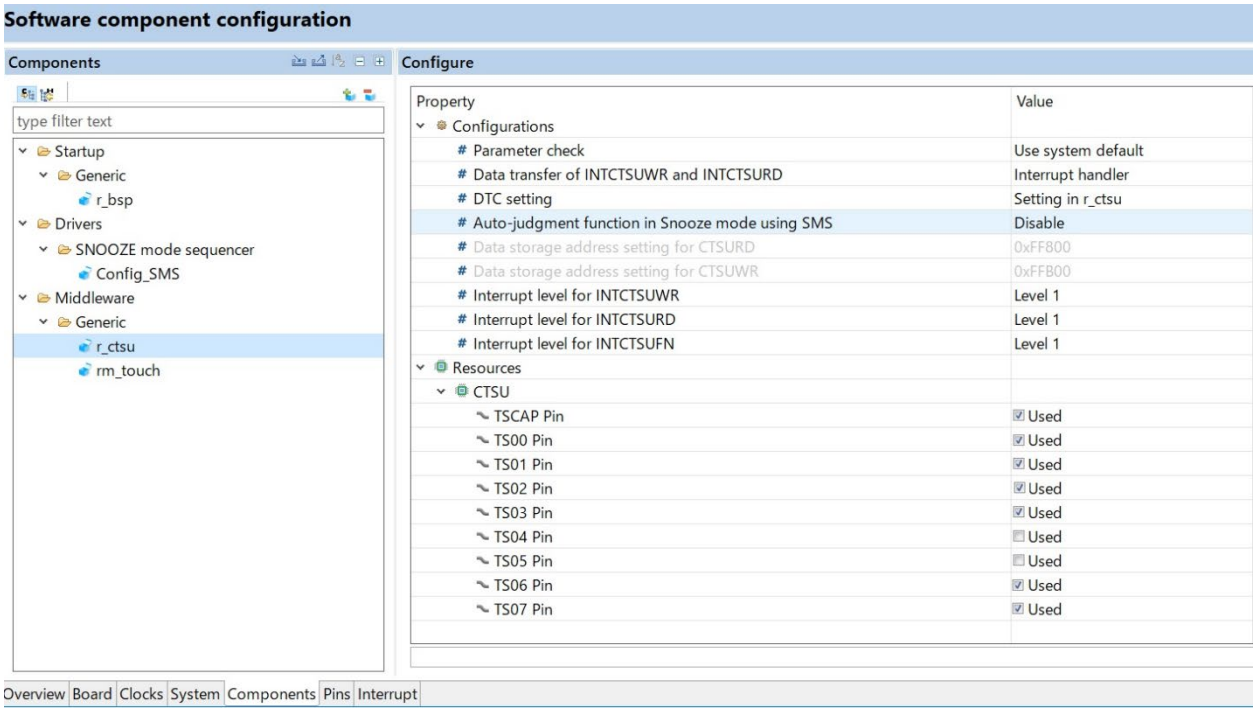
Table 4.13 32-bit interval timer settings

Channel	Setting	Interrupt priority	Usage
TML32 CH0	Interval timer (16us)	Non-interruptible	SMS activation trigger
TML32 CH1	Unused		-
TML32 CH2	Unused		-
TML32 CH3	Unused		-

4.5.2.7 CTSU

This driver uses a capacitance sensor unit. This sample application uses the Smart Configurator (SC) and has the following settings.

Figure 4-13 CTSU settings



4.5.2.8 SAU

This driver uses the Serial Array Unit.

4.5.2.9 WDT

This driver uses a watchdog timer. This driver restarts the watchdog timer.

Refer to 4.6.2 List of optional byte settings for information on setting the watchdog timer.

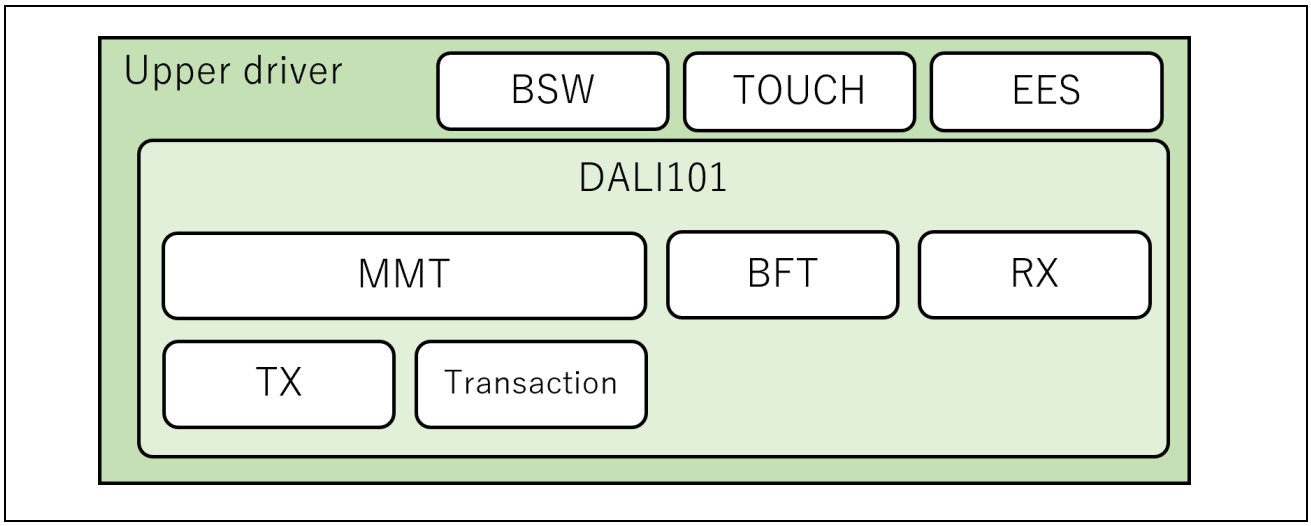
4.5.2.10 RFD

This driver reads and writes to the RL78/G23 data flash. This sample application uses Renesas Flash Driver from Renesas Electronics.

For details, refer to Renesas Flash Driver RL78 Type01 User's Manual for RL78/G23 (R20UT4830).

4.5.3 Upper driver layer

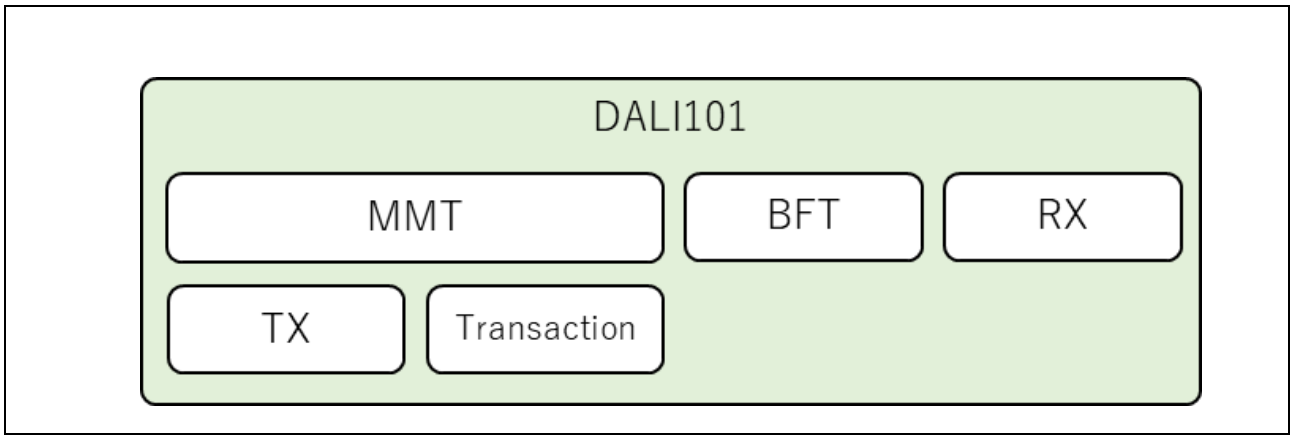
Highly functional drivers that are realized by combining lower-level drivers.



4.5.3.1 DALI101 communication driver

The DALI communication driver of this sample application has the following configuration.

Figure 4-14 DALI101 communication driver configuration



(1) DALI101

API function group for the DALI101 communication driver.

Within this module, implement features to acquire the state of the DALI RxD pin (RxD level duration and receive bit width) and pass it to the MMT and RX described below.

- DALI RxD level duration measurement

Start/restart the 50 us interval timer (Timer Array Unit 0 Channel 2) triggered by an edge detection on the DALI RxD pin. The duration of the RxD level (High/Low) is measured by adding 50 us to the duration in the timer's interrupt handler.

- DALI RxD receive bit width measurement

The receive bit width is measured using the input pulse interval measurement feature of the Timer Array Unit for the DALI RxD pin.

However, as each measurement time is based on the microcontroller terminal reference, it varies with the rising or falling delay in the DALI communication circuit external to the microcontroller, resulting in an error in the acquisition time. For this reason, a delay correction is made in consideration of the delay in the DALI communication circuit before being passed on to the MMT and RX.

Refer to 4.5.6 Threshold value and calibration for more information on corrections and thresholds.

Other functions are implemented to wrap MMT, BFT and RX function groups.

(2) Multi-Master Transmitter (MMT)

This module realizes DALI's Multi-Master Transmitter provision.

It starts transmitting the DALI Frame registered by the user at the appropriate settling time corresponding to the “priority”. It also checks the loopback received data during transmission to see if a collision has occurred.

If a collision occurs, transmission is aborted and appropriate recovery processing is performed.

(3) Backward Frame Transmitter (BFT)

This module is used to transmit Backward Frames.

Transmitting a valid 8-bit Backward Frame is achieved by using the Transmitter (TX) described below, while transmitting a Corrupted Backward Frame is achieved by setting the DALI TxD pin to output Low for 1.458ms and then High.

Corrupted Backward Frame refers to a damaged Backward Frame caused by the collision of multiple Backward Frames. When there are multiple logical DALI devices in a DALI device and each device has different Backward Frame data contents, the Backward Frames are aggregated and transmitted as a Corrupted Backward Frame.

(4) Receiver (RX)

This module realizes the Receiver provisions of DALI.

The DALI RxD receive bit width/bit level (High/Low) received from the upper driver DALI101 is sequentially stored in a ring buffer and the Stop Condition and System Failure are detected by the DALI RxD level duration time.

When a Stop Condition is detected, Frame reception is regarded as complete and whether or not the frame satisfies the DALI standard is checked from each bit width/bit level (High/Low) stored in the ring buffer until then.

If the frame is a normal frame, information on whether or not the same frame was received within the previous 100 ms is also provided.

(5) Transaction (TRANSACTION)

Buffer module for Frame transmission, including Transaction specifications (a group of frames that make sense in a series of frames).

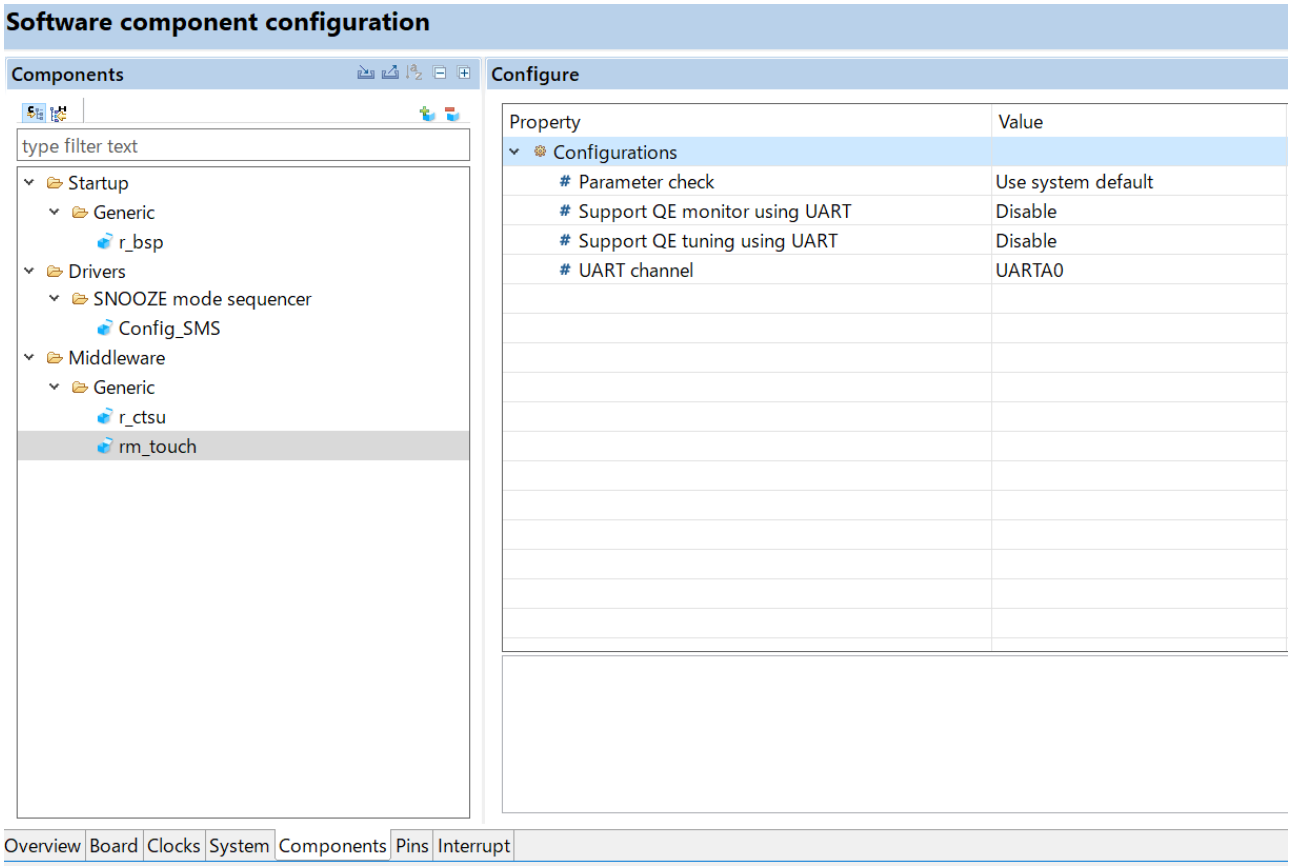
(6) Transmitter (TX)

This module transmits Frames up to 32 bit using SMS and TML32.

4.5.3.2 TOUCH

This middleware handles touch keys. This sample application uses the Smart Configurator (SC) and has the following settings.

Figure 4-15 TOUCH settings



4.5.3.3 EES

This software is designed to perform EEPROM emulation features using the RL78/G23's data flash. This sample application uses Renesas Electronics EEPROM emulation software.

For more information, refer to EEPROM Emulation Software RL78 Type01 for RL78/G23 User's Manual (R20UT5008).

4.5.3.4 BSW

Driver for acquiring the input status of Binary Switch (abbreviated BSW), including bouncing (unstable decrease that temporarily repeats ON/OFF when the setting status changes) measures by specifying the pin to which the Binary Switch is connected.

4.5.4 Library layer

The DALI103i and DALI301 libraries are used as libraries for Input Device in DALI communication.

4.5.5 Application layer

4.5.5.1 Secure Input Device entity

Secure Instance and logical unit entities according to the specifications of the Input Device. In this sample application, the elements required for one logical unit are grouped in the logical_unit_t structure before allocating the entity.

4.5.5.2 Input Device application

(1) Forward Frame processing on receipt

This analyses the received Forward Frame, executes the command and transmits the Backward Frame.

(2) logical unit processing

This sample application is a single logical unit; multiple logical units can be implemented if more than 32 Instances are required.

(3) Instance processing

This processes the signal processor. Implement the processing according to the signal processor used.

This sample application uses touch keys as the signal processor for the Push Button.

(4) NVM processing

This performs the data flash storage process. In this sample application, it is implemented in the EES and RFD libraries.

(a) Autosaving

As the DALI standard requires that the NVM variable save its state at least 30 seconds before power-off, this sample application automatically saves any changes to the NVM variable every second. If you want to reduce the number of writes to the data flash, either increase the save frequency or only write when power off is detected.

(b) Manual saving

To save NVM variables explicitly, send the "SAVE PERSISTENT VARIABLES" command from the Application controller.

After receiving this command, save all NVM variables within 300 ms.

(5) Event processing

(a) POWER NOTIFICATION

This event is used to notify power supply reconnection.

Transmit at random timings between 1.3s and 5s from the start of power supply to the Input Device.

(b) INPUT NOTIFCATION

Events can be issued according to the state of the signal processor. When implementing Instance Type 1 to 31, implement according to the corresponding standard (301 to 331); when implementing with Instance Type 0, the user should define the specification.

(6) Random number generation process

This generates random numbers for use in the DALI103i library. The conditions for the random numbers to be generated are as follows.

- It should be generated in the range [0x000000, 0xFFFFFE].
- The same value is not generated even if the generation is repeated at least as many times as the number of logical units.

* When implementing pseudo-random numbers, the seed value must be a genuine random number.

4.5.6 Threshold value and calibration

For details on the contents of this section, refer to the same section of the RL78/G23 DALI-2 Input Device Basic (103) Sample Application Notes.

4.6 Software configuration

This section describes the software configuration of the sample application.

4.6.1 Folder structure

The folder structure of this sample application is shown below.

Table 4.14 Folder structure (1/2)

Folder, File name	Description
DALI103i_301_sample	Folder containing the sample application
\App	Folder containing the application program
r_common.h	Common definition header file
r_input_device.c	Input Device module source file
r_input_device.h	Input Device module header file
r_it1_instance.c	Instance Type 1 source file
r_it1_instance.h	Instance Type 1 header file
r_main.c	Main source file
r_memory_bank.c	Memory bank module source file
r_memory_bank.h	Memory bank module header file
r_memory_banks.c	Source file for modules managing multiple memory banks
r_memory_banks.h	Header file for modules managing multiple memory banks
r_nvm.c	Source file for NVM modules supporting data flash libraries
r_nvm.h	Header file for NVM modules supporting data flash libraries
r_random.c	Source file for random number generation modules
r_random.h	Header file for random number generation modules
r_unit0_memory_bank.c	Source file for unit0 definition of memory banks
r_unit0_memory_bank.h	Header file for unit0 definition of memory banks
\Driver	Folder containing the driver program
r_bsw.c	Binary Switch driver source file
r_bsw.h	Binary Switch driver header file
r_dali101.c	DALI101ed.2.0 compliant driver source file
r_dali101.h	DALI101ed.2.0 compliant driver header file
r_dali101_bft.c	DALI101ed.2.0 compliant driver (Backward Frame transmission) source file
r_dali101_bft.h	DALI101ed.2.0 compliant driver (Backward Frame transmission) header file
r_dali101_common.h	DALI101ed.2.0 common definition header file
r_dali101_mmt.c	DALI101ed.2.0 compliant driver (Multi Master transmission) source file
r_dali101_mmt.h	DALI101ed.2.0 compliant driver (Multi Master transmission) header file
r_dali101_rx.c	DALI101ed.2.0 compliant driver (Receive) source file
r_dali101_rx.h	DALI101ed.2.0 compliant driver (Receive) header file
r_dali101_transaction.c	DALI101ed.2.0 compliant driver (transaction) source file
r_dali101_transaction.h	DALI101ed.2.0 compliant driver (transaction) header file
r_dali101_tx.c	DALI101ed.2.0 compliant driver (transmission) source file
r_dali101_tx.h	DALI101ed.2.0 compliant driver (transmission) header file
r_port.c	Source file for PORT
r_port.h	Header file for PORT
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
r_tml32.c	Source file for TML32
r_tml32.h	Header file for TML32

Table 4.15 Folder structure (2/2)

Folder, File name	Description
DALI103i_301_sample	Folder containing the sample application
\Driver	Folder containing the driver programs
r_trng.c	Source files for TRNG
r_trng.h	Header file for TRNG
r_wdt.c	Source file for WDT
r_wdt.h	Header file for WDT
\HardwareDebug	Folder containing project output files
DALI103i_301_sample.mot	Motorola S-type format files used for writing to firmware
\Library	Library storage folder
\DALI103i	Folder for DALI103ed1.0 compliant libraries
\DALI301	Folder for DALI301ed1.0 compliant libraries
\EES	Library storage folder for EES
\RFD	Library storage folder for RFD
\QE-Touch	Storage folder for touch-key configuration files
DALI103i_301_sample.tifcfg	Touch key configuration file
\qe_gen	Storage folder for QE auto-generated files
qe_touch_config.c	Touch configuration definition source file
qe_touch_config.h	Touch configuration definition header file
qe_touch_define.h	Touch configuration definition header file
\src	Storage folder for smart configurator generation files
smc_gen	Folder for smart configurator generation
Utility	Common program storage folder
r_timer16.c	Timer module source file
r_timer16.h	Timer module header file
r_usertype.h	Type definition header file

4.6.2 List of optional byte settings

The optional byte settings for the sample application are shown below.

Table 4.16 List of optional byte settings

Address	Set value	Description
000C0H / 040C0H	11101111B	Watchdog timer operation permitted ・After reset is released, counting stops ・Overflow time: $2^{17}/f_{IL}$ (3478.26 ms)
000C1H / 040C1H	11111101B	LVD reset mode (falling: 2.91 V, rising: 2.97 V)
000C2H / 040C2H	11101000B	HS mode, high-speed on-chip oscillator: 32 MHz
000C3H / 040C3H	10000100B	On-chip debugging permitted

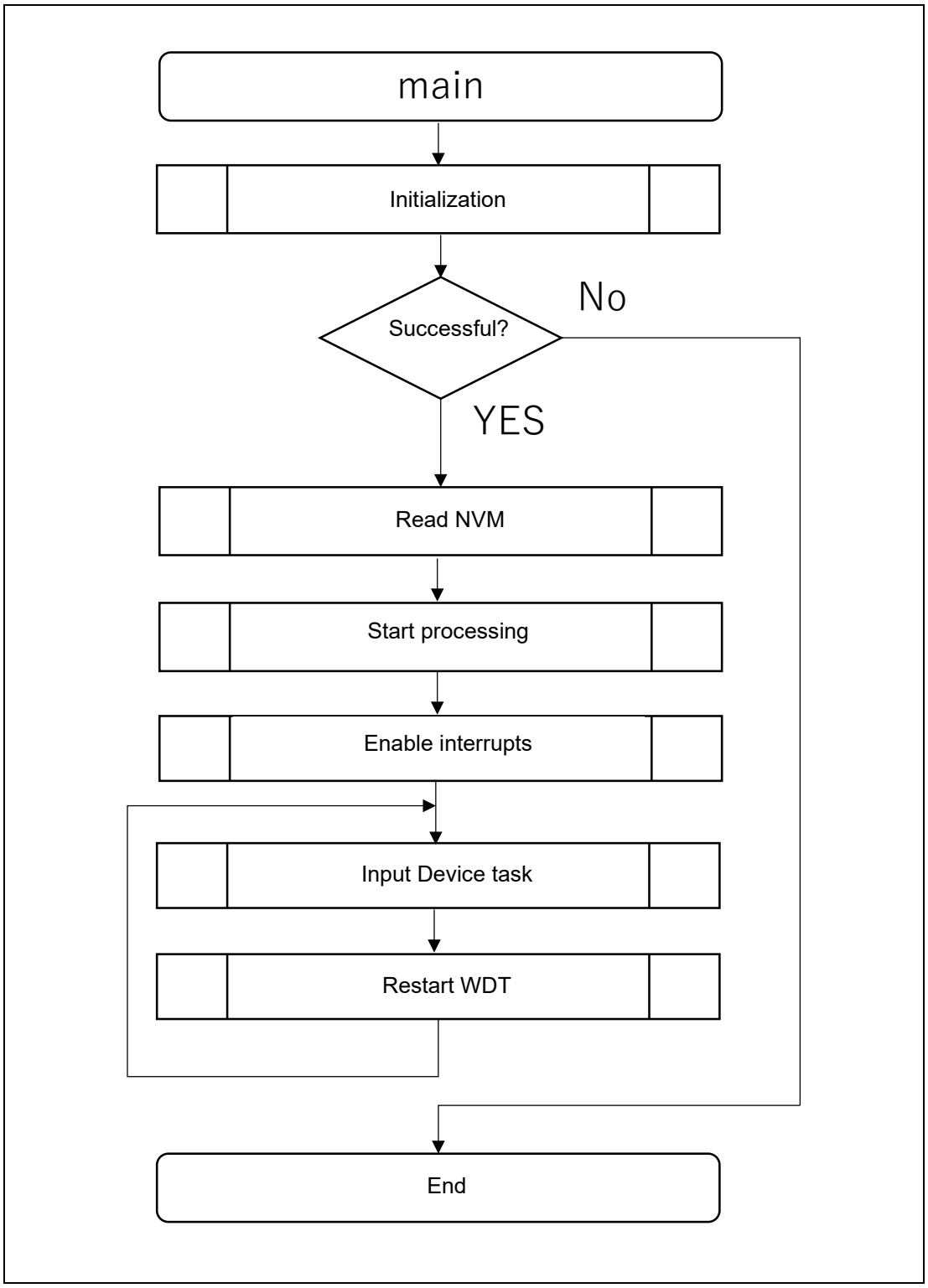
4.6.3 Flowcharts

4.6.3.1 Main flow

The main flow of the sample application is shown below.

After the initialization process is completed, the Input Device process is executed.

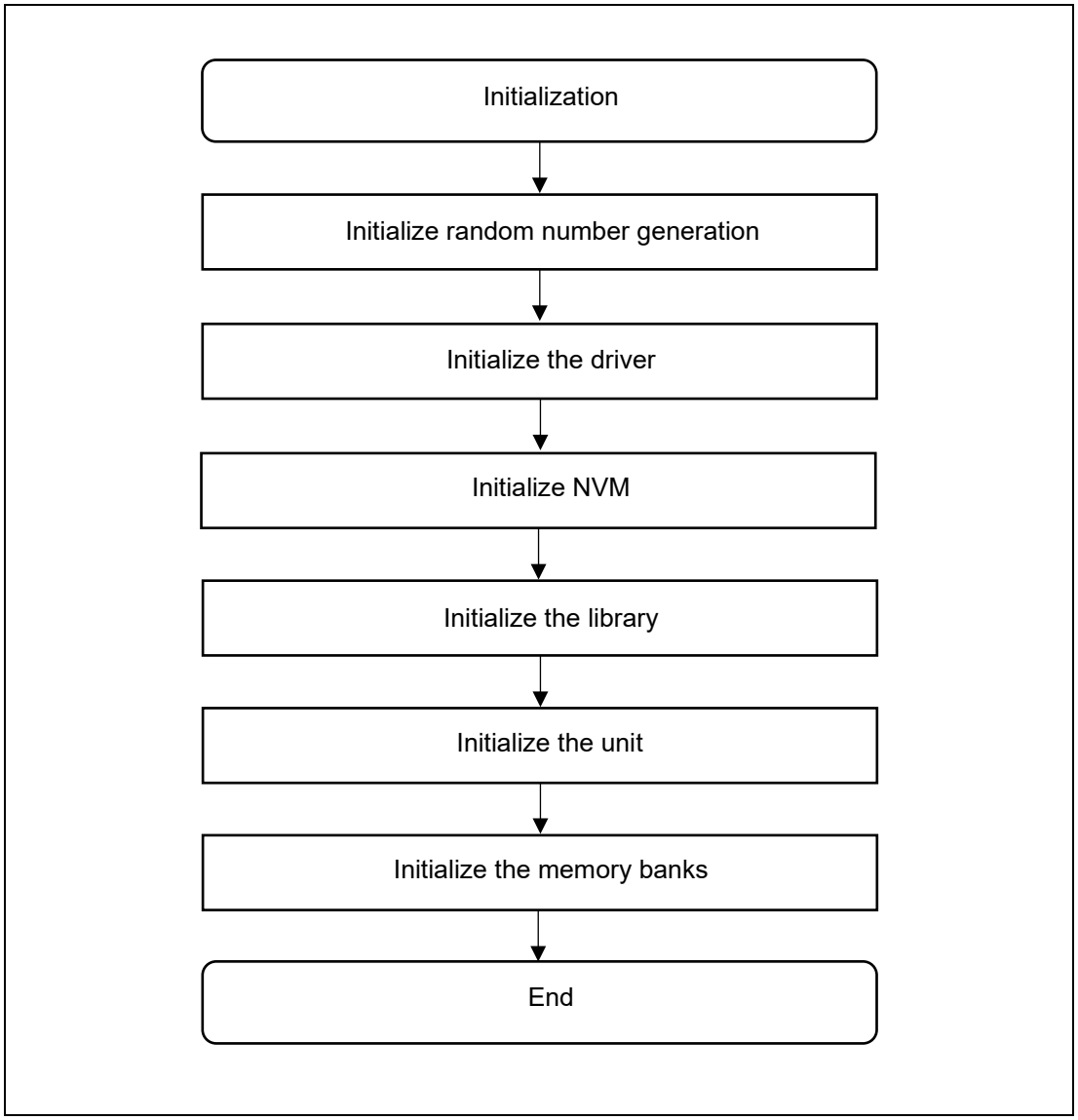
Figure 4-16 Main flow chart



4.6.3.2 Initialization

The flow of the initialization process is shown below.

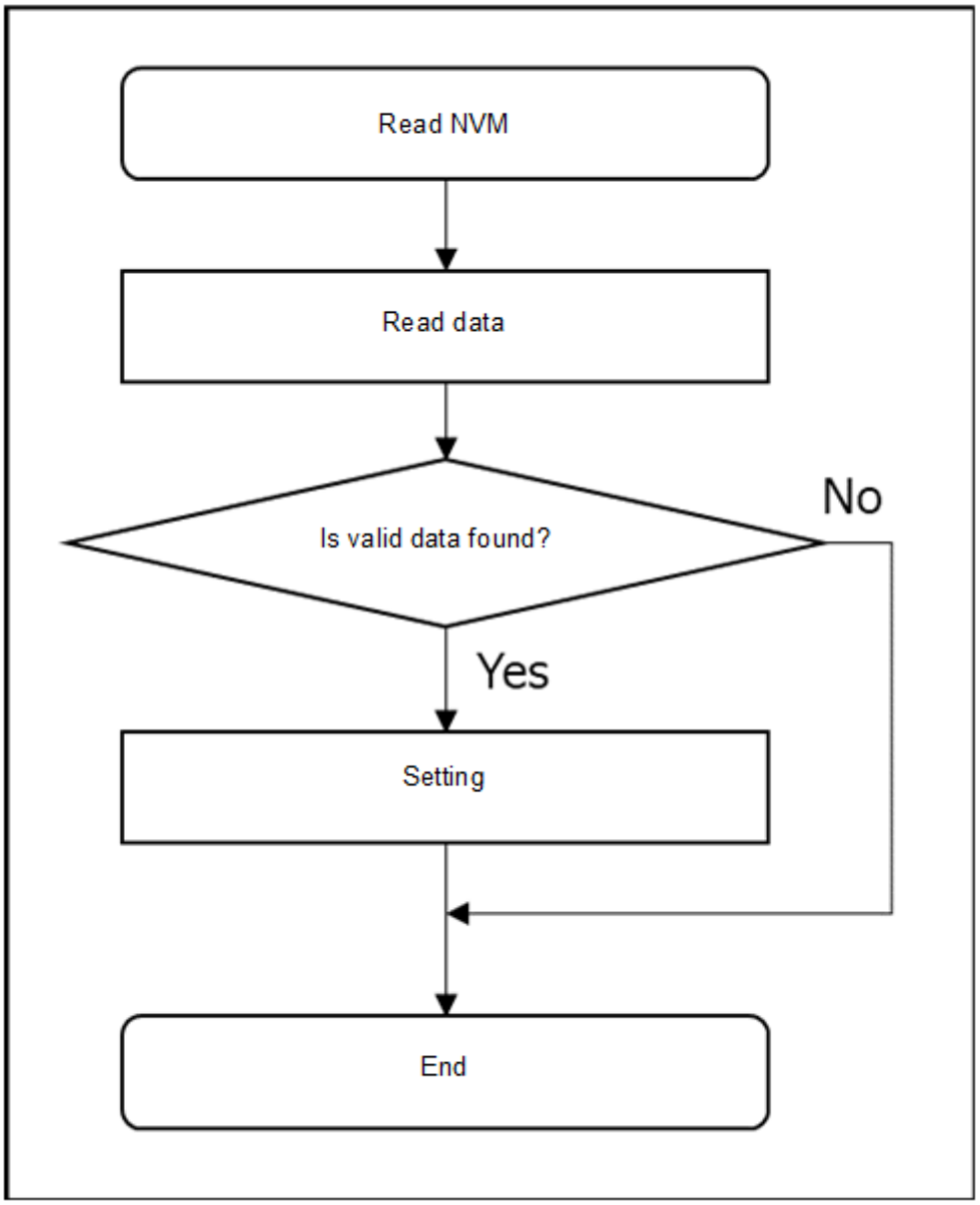
Figure 4-17 Initialization process flow diagram



4.6.3.3 NVM reading

The flow of the NVM loading process, which is carried out after the initialization process, is shown below.

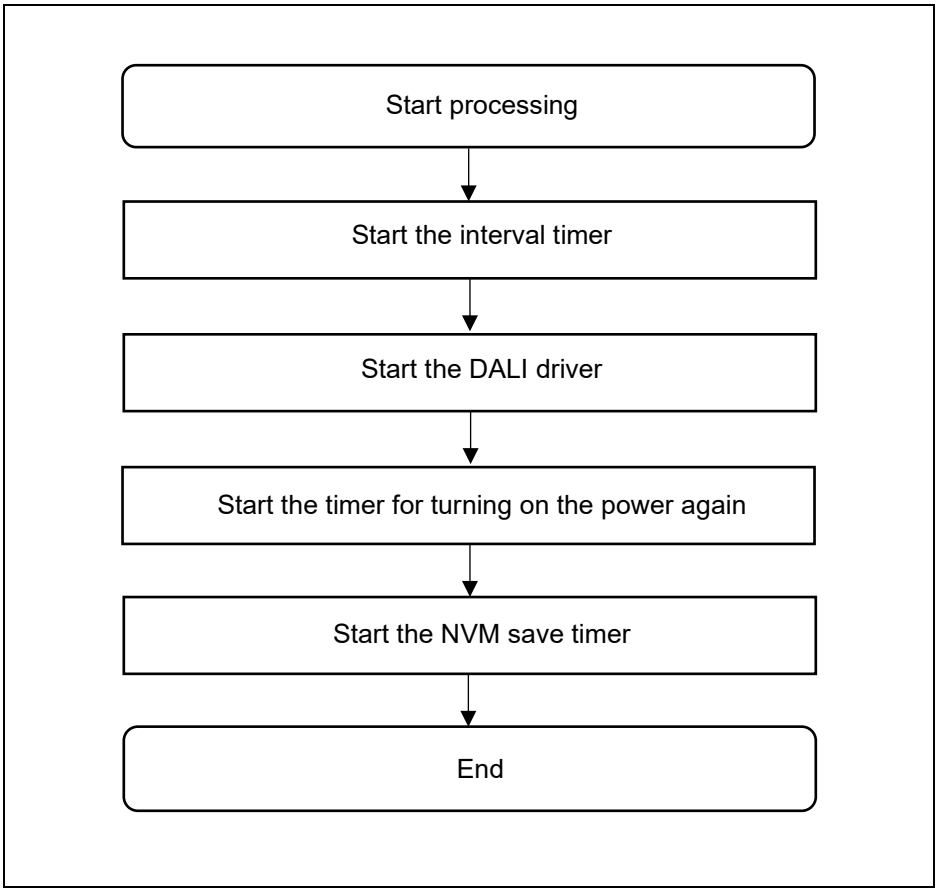
Figure 4-18 NVM reading process flow diagram



4.6.3.4 Start of processing

The flow for starting the process, which is carried out after the NVM read process, is shown below.

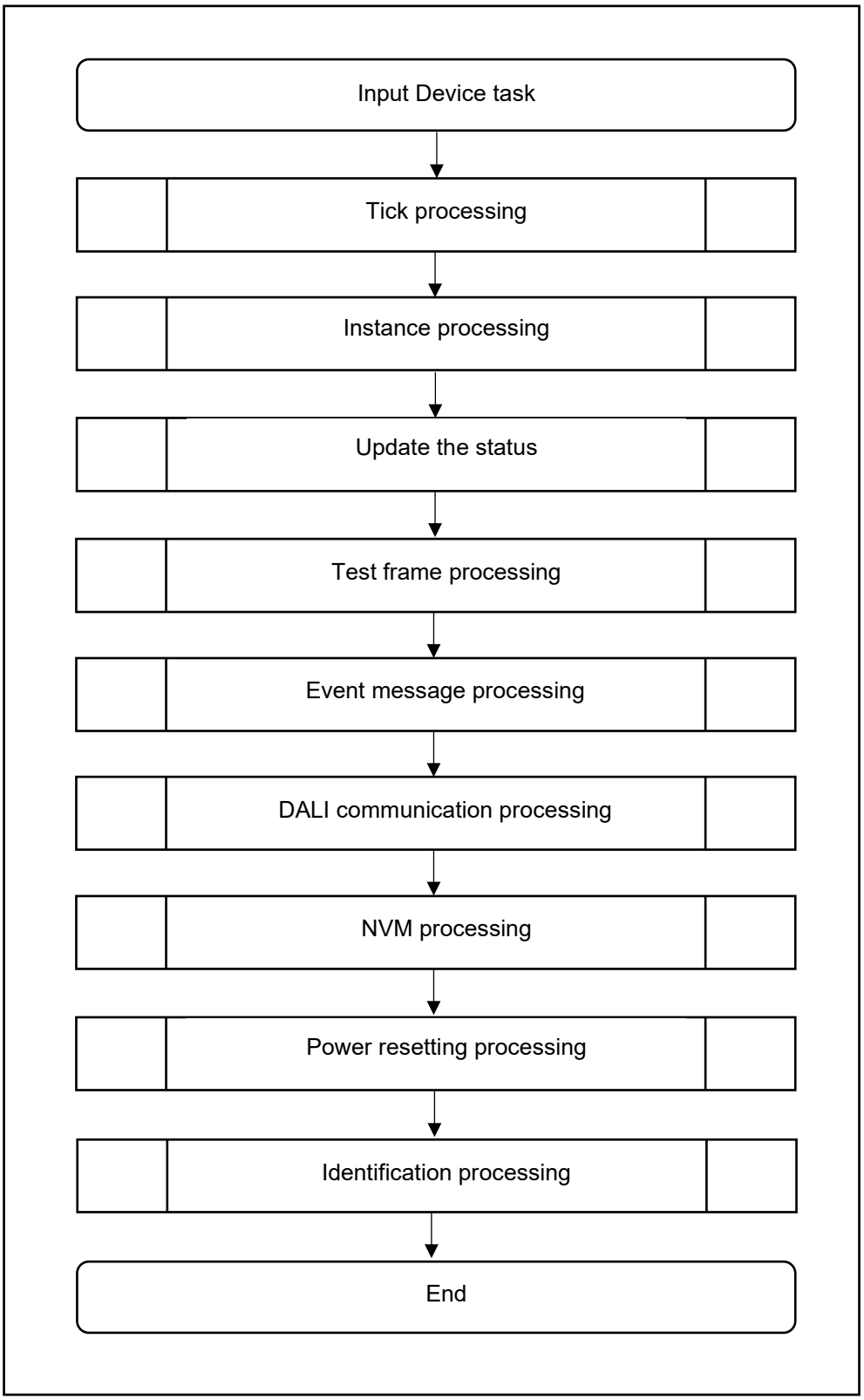
Figure 4-19 Process initiation flow diagram



4.6.3.5 Input Device task processing

The flow of Input Device task processing is shown below.

Figure 4-20 Input Device task processing flow diagram



5. Notes

- This sample application is configured and operated in accordance with the board RL78/G23 Lighting Communication Master Evaluation Board.
- The operation provided by this sample application is only a sample. The application layer should be redesigned and evaluated according to the intended use.

6. Reference documents

RL78/G23 User's Manual: Hardware (R01UH0896EJ0110)

RL78 Family User's Manual: Software (R01US0015J)

Integrated Development Environment e2 studio 2020-04, e2 studio v7.8 User's Manual: Getting Started Guide (R20UT4819)

DALI Master Controller GUI User's Manual (R20UT0715)

RL78/I1A DC/DC LED Control Evaluation Board (EZ-0012) (R01UH0363)

RL78/I1A DALI-2 Control Gear Basic (102) LED (207) Color Control (209Tc) Sample Application (R01AN6177)

(Please obtain the latest version from the Renesas Electronics website.)

Technical Update / Technical News

(Please obtain the latest version of information from Renesas Electronics website.)

DALI standards

IEC 62386-101:2014+AMD1:2018 CSV Consolidated version (ed.2.1)
IEC 62386-103:2014+AMD1:2018 CSV Consolidated version (ed.1.1)
IEC 62386-301:2017 (ed.1.0)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	2023/11/08		
1.01	2025/11/28	30	SMS setting

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/.