RL78/I1A

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

Introduction

This application note describes a sample application for DALI (Digital Addressable Lighting Interface) communication using RL78/I1A microcontroller.

Two sample applications are provided, each running as Control Gear. The DALI standards supported are IEC 62386-101 Edition2.0 (hereafter 101ed.2.0), IEC 62386-102 Edition2.0 (hereafter 102ed.2.0), IEC 62386-207 Edition1.0 (hereafter 207ed.1.0), IEC 62386-209 Edition1.0 (hereafter 209ed.1.0). As of October 2021, the combination of each standard is a combination that can support DALI-2 certification.

To achieve communication that complies with the standards, the waveforms of Manchester-encoded DALI signals are processed using the peripheral functions on the RL78/I1A.

This application note assumes that the user already has some knowledge of DALI; for more information on the DALI standard, refer to the 7. Reference Documents.

Target Device

RL78/I1A

Notes: 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.
DALI-2 certification are not semiconductor or software tests. Please build your own Control Gear that can handle the DALI-2 certification.

Contents

1. Specifications

The following two sample applications are described in this application note.

- DALI102ed.2.0+207ed.1.0 sample application
  This is a sample application that includes the Control Gear function (102ed.2.0) and the LED light source extension function (207ed.1.0). The main features of the extension include switching between linear and non-linear dimming curves, fade extension, and LED light source specific functions.
  This sample application runs in 3 logical units, and the failure status of the LED can be set and checked on a PC using the serial communication function of the EZ-0012.
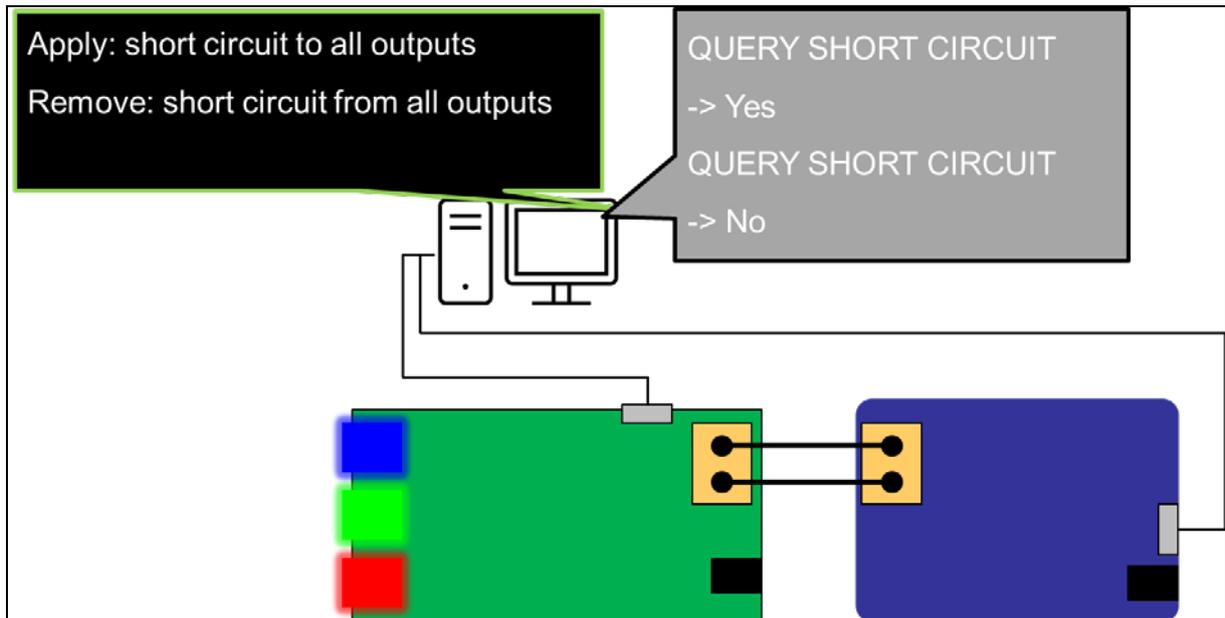
Figure 1-1 102ed.2.0+207ed.1.0 Sample Application Operation Schematic

Note: The serial failure and recovery notifications are only for the purpose of checking operation as a sample application. When building it into the actual Control Gear, build it according to the specifications.

- DALI102ed.2.0+209ed.1.0 sample application
  This is a sample application of the Control Gear feature (102ed.2.0) with the color control extension (209ed.1.0). As the main contents of the extension, 3 types of light source dimming and color control methods (X,Y / RGBWAF / Tc) are defined.
  This sample application is designed to control color temperature (Tc) and operates with 1 logical unit. It uses blue LED (high temperature range) and red LED (low temperature range) as LEDs for dimming and color control. It also checks the actual level and color temperature (Tc) every 100ms and outputs the actual level and color temperature (Tc) to the serial communication monitor on the PC only when there is a change in any of the values.
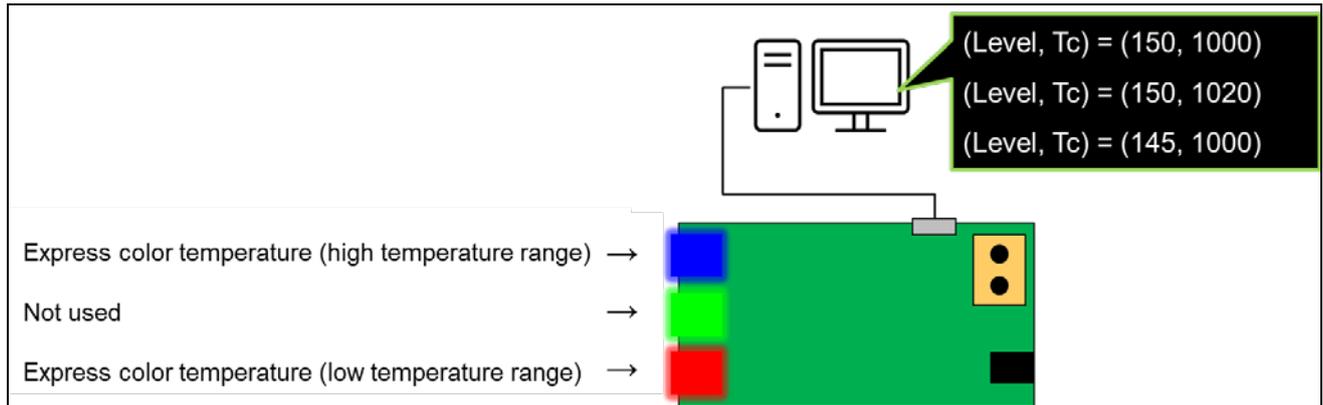
Figure 1-2 102ed.2.0+209ed.1.0 Sample Application Operation Schematic

Note: The serial display operation is only for checking the color temperature (Tc) as a sample application.
Note: The level displayed on the serial communication monitor refers to the actual level.

This application note covers the combination of 102ed.2.0 + 207ed.1.0 and the combination of 102ed.2.0 + 209ed.1.0. It is also possible to operate 102ed.2.0 alone or a combination of 102ed.2.0, 207ed.1.0, 209ed.1.0.

2. Operating conditions

Each sample application has been tested in the following environments.

Table 2-1 Confirmed operation environment

Item	Description
Microcontroller	RL78/I1A (R5F107DE)
Operating Frequency	High-speed on-chip oscillator clock: 32MHz
Operating Voltage	5.0V
Integrated Development Environment (e2 studio)	Made by Renesas Electronics 　e2 studio 2021-04
C Compiler (e2 studio)	Made by Renesas Electronics 　CC-RL V1.10.0
Integrated Development Environment (IAR)	Made by IAR Systems IAR Embedded Workbench IDE V8.5.2.7561
C Compiler (IAR)	Made by IAR Systems IAR C/C++ Compiler for Renesas RL78 V4.21.3.2447
Library	Made by Renesas Electronics Data flash libraries Type04 V2.00
Emulator	E1 emulator / E2 emulator Lite
Board	Made by Renesas Electronics 　EZ-0012＋EZ-0012 expansion board 　TCM-RL78I1A
Tool	DALI master controller GUI V2.02 or later

3. Description of the Hardware

3.1 System configuration

An example of the system configuration is shown below.

Each sample application is operated by Control Gear which is compatible with EZ-0012+EZ-0012 expansion board. The Control Gear is a slave device in the DALI system and requires a separate Application Controller which is a master device that outputs command instructions. In this application note, TCM-RL78I1A is used as Application Controller.
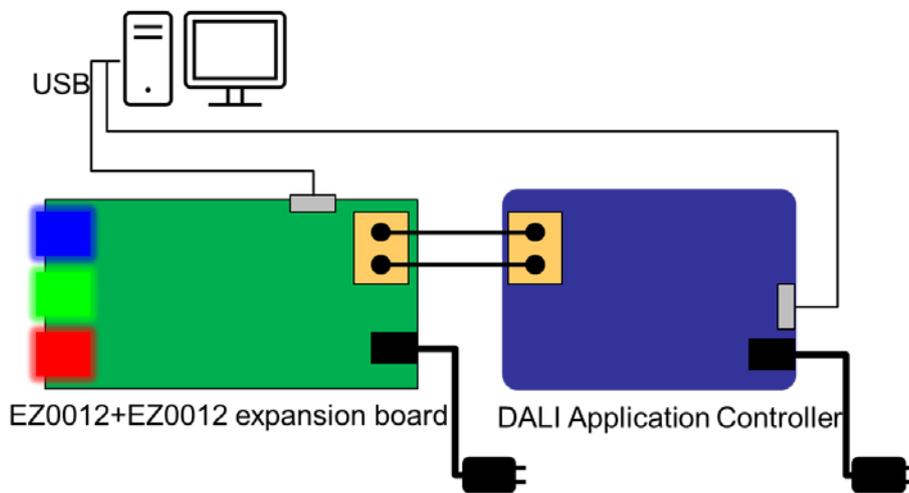
Figure 3-1 Example of the system configuration

3.2   List of Pins to be Used

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

Table 3-1 Pins to be Used and Their Features

Pin Name	I/O	Description
P10/DALITxD4	Output	DALI communication (Transmission)
P11/TI07	Input	DALI communication (Reception)
P03/RxD1	Input	UART communication (Reception)
P02/TxD1	Output	UART communication (Transmission)
P200/TKBO00	Output	LED101 lights up
P201/TKBO01	Output	LED102 lights up
P202/TKBO10	Output	LED103 lights up
P22/CMP0P	Input	LED101 lighting Feedback
P24/CMP1P	Input	LED102 lighting Feedback
P25/CMP2P	Input	LED103 lighting Feedback
RESET	Input	RESET Switch

For other features about EZ-0012, refer to the following.
- EZ-0012 RL78/I1A DC/DC LED Control Evaluation Board User's Manual (R01UH0363)

4.   Description of the Software

4.1   Environment setup

4.1.1   Downloading the Software

The following software is required to run each sample application.

Download them from the Renesas Electronics website.

- e2 studio
- Data Flash Library Type04 V2.00
- DALI Master Controller GUI V2.02 or later

4.1.2 Installing e2 studio

e2 studio is required to develop programs for each sample application and to perform on-chip debugging.

For details on installation and other basic operations, refer to the following user's manual.

- e2 studio 2020-04 and e2 studio v7.8 Getting Started Guide (R20UT4819)

4.1.3 Installing the Data Flash Library Type04

Each sample application does not include the data flash library. Therefore, it is necessary to register the library file by downloading it from the Renesas Electronics website.

Refer to the following release notes for installation. For details on registering the library files, refer to chapter 4.1.7.

- RL78 Family Data Flash Library Type04 Package Ver.2.00 Release Note (R20UT0749)

4.1.4 Installing the DALI Master Controller GUI

The DALI master controller GUI is required to evaluate each sample application.

Refer to the following user's manual for the installation procedure.

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

4.1.5 Importing the sample application

1. Unzip the provided project file and place it in a folder of your choice.
2. Start e2 studio, right-click in Project Explorer or select File tab -> Import.
3. Select "General" and then "Existing Project to Workspace" and click "Next".
4. Select "Select Root Directory" and select the project file placed in 1. from "Browse".
5. Select only the sample applications to run.
    e.g. In the case of DALI102ed.2.0+207ed.1.0 sample application
    Select the check box for dali102_207_sample and uncheck the check boxes for other sample applications.
6. Click "Finish".

4.1.6 Placing the Data Flash Library

After importing the sample application to run, store the data flash library downloaded from Renesas Electronics in the following directory.

¥DALI102_xxx_sample<DIR>			FDL folder: Folder to store the data flash library.
	¥Library<DIR>		It should contain pfdl.h / pfdl.lib (CC-RL) or pfdl.a (IAR) /pfdl_types.h according to the development environment to be used.
		¥FDL<DIR>	
		pfdl.h	
		pfdl.lib (CC-RL)	
		pfdl.a (IAR)	
		pfdl_types.h	

4.1.7 Building the sample application

1. Right click on the project in the Project Explorer and select "Build Project".
2. The build begins and the status of the build is displayed in the "Console". When the message "Build Finished" is displayed, the build is complete.

4.1.8 Hardware connection and setting switch for sample application

To use DALI communication, it is necessary to change the status of the setting switch (SW1) of EZ-0012 from the default setting.
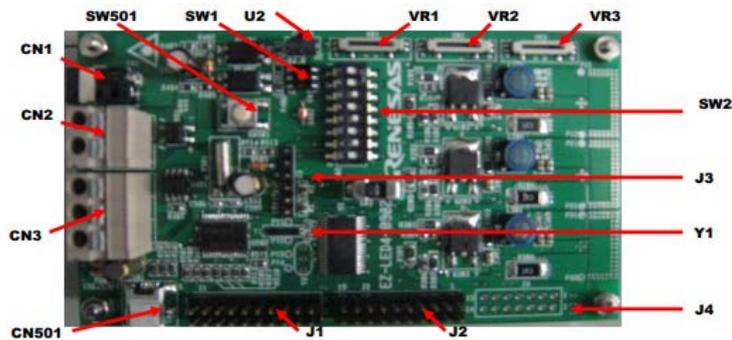
Figure 4-1 RL78 / I1A DC / DC LED Control Evaluation Board(EZ-0012) Components

SW1 should be changed to the "DALI" side as shown below.

Figure 4-2 Setting switch SW1

The hardware connection and EZ-0012 setting switch (SW2) settings are different between using EZ-0012 + EZ0012 expansion board and operating with EZ-0012 only.

Chapter 4.1.8.1 and 4.1.8.2 describes how to configure the settings according to the hardware and debug tools to be used.

4.1.8.1 Using the EZ-0012 expansion board

The basic hardware connection when using the EZ-0012 + EZ-0012 expansion board is shown in Figure 4-3.
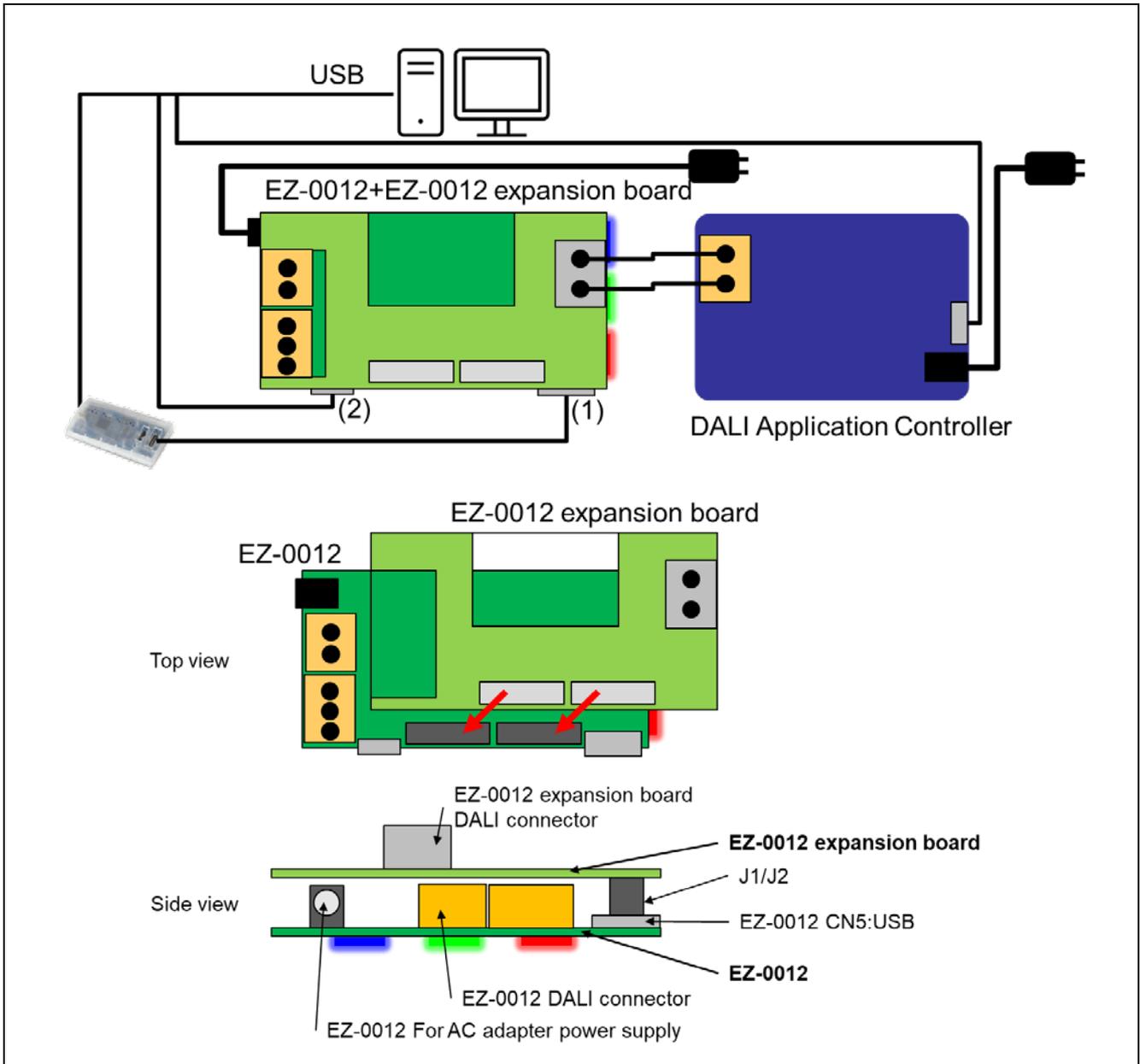
Figure 4-3 Connection diagram when using EZ-0012 expansion board

Connect (2) when debugging via USB.

Connect (1) when debugging with an emulator.

Connect (1) and (2) when debugging while using the serial communication monitor.

Depending on the tool used for debugging, it is necessary to change the contents of the setting switches of EZ-0012. The following table shows the contents of the setting switches.

Table 4-1 Setting switch SW2 (EZ-0012 expansion board)

Number	Debug Setting		
	USB	Emulator	UART + Emulator
1	OFF	OFF	OFF
2	OFF	OFF	OFF
3	ON	OFF	OFF
4	OFF	OFF	ON
5	ON	OFF	OFF
6	OFF	OFF	ON
7	ON	ON	ON
8	OFF	ON	ON

4.1.8.2 Using the EZ-0012 only

When debugging using only the EZ-0012 without the EZ-0012 expansion board, the hardware connections are as shown in Figure 4-4.
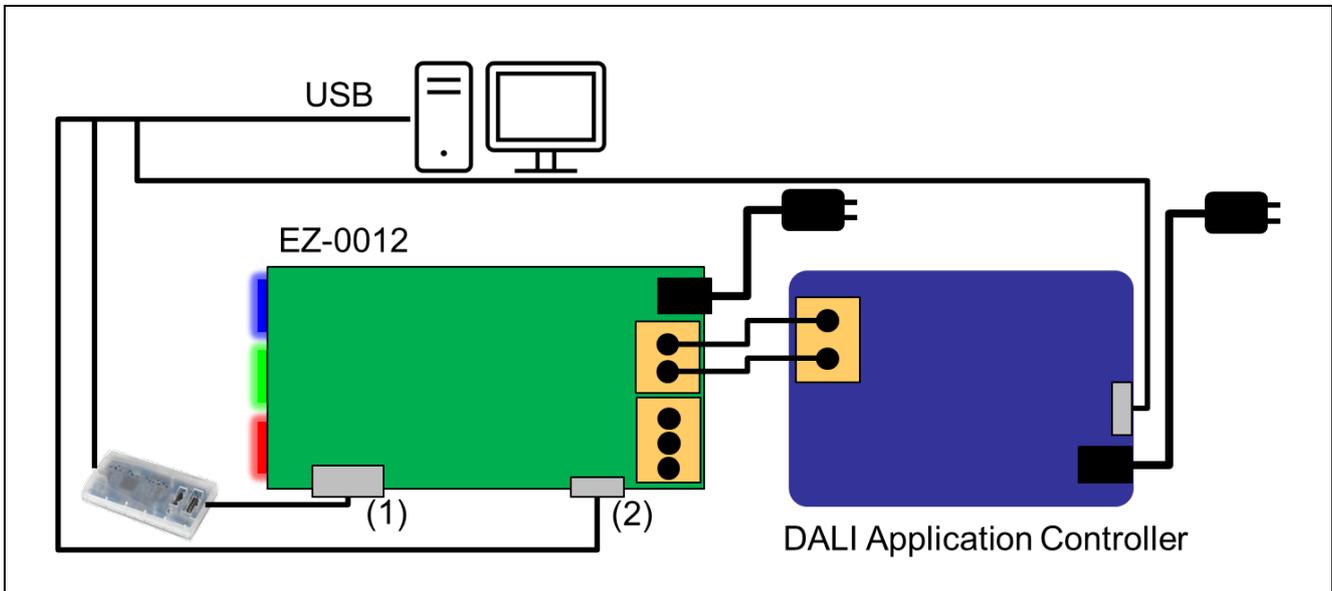
Figure 4-4 Connection diagram when using EZ-0012 only

Connect (2) when debugging via USB.

Connect (1) when debugging with an emulator.

Connect (1) and (2) when debugging while using a serial communication monitor.

Depending on the tool used for debugging, it is necessary to change the contents of the setting switches of EZ-0012. The following table shows the contents of the setting switches.

Table 4-2 Setting switch SW2 (EZ-0012)

Number	Debug Setting		
	USB	Emulator	UART + Emulator
1	ON	ON	ON
2	ON	ON	ON
3	ON	OFF	OFF
4	OFF	OFF	ON
5	ON	OFF	OFF
6	OFF	OFF	ON
7	ON	ON	ON
8	OFF	ON	ON

For other details about EZ-0012, refer to the following user's manual.

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

4.1.9 Debugging the Sample Application

1. After importing the sample application in e2 studio, click the [icon] button to download the program to the microcontroller.
2. Select "Run" -> "Debug Configuration..." to open the "Debug Configuration" window.
3. In the "Debug Configuration" window, expand the "Renesas GDB Hardware Debugging" debug configuration view and click on an existing debug configuration.
4. Switch to "Debugger" -> "Connection Settings" tab and confirm that the settings are as shown below.
   * Select "Debug hardware: E1/E20 (RL78)" when using E1 emulator.
   * Select "Debug hardware: E2 Lite (RL78)" when using E2 emulator Lite.
   * Select "Debug hardware: EZ (RL78)" when using USB.
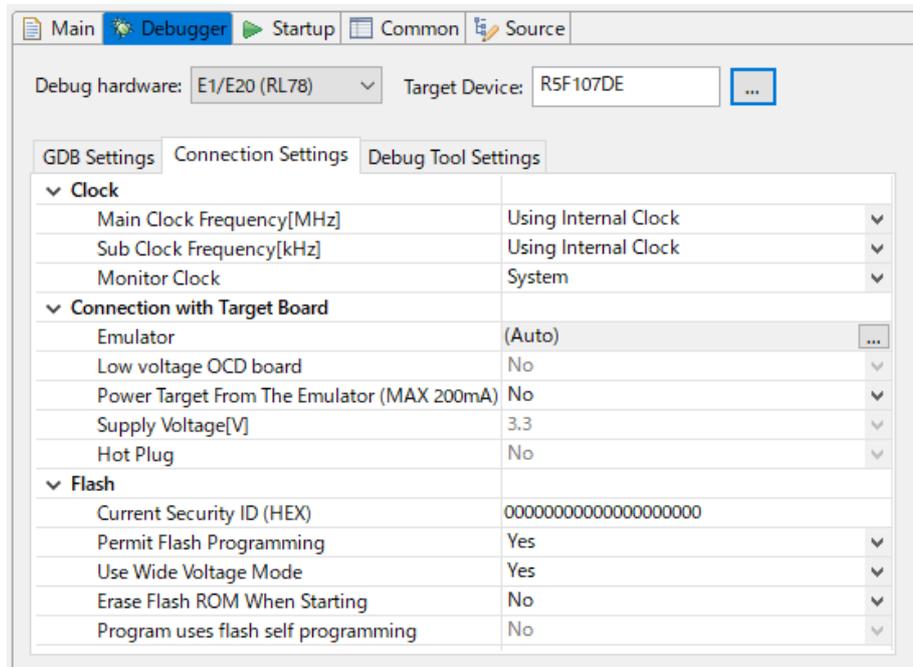
Figure 4-5 Debug screen setting contents

5. Select "Start Debug", and when the "Debug" view window appears, debugging is ready.

For details on installation and other basic operations, refer to the following user's manual.

- e2studio User's Manual: Getting Started Guide for V7.0 (R20UT4374)

4.2   Operating Procedure

In this application note, the procedure to run each sample application using the DALI master controller GUI is shown below.

(1)   Common operating procedures

1.   After the EZ-0012 and the master device that operates the EZ-0012 are connected, power is supplied to each device.
     For the system configuration, refer to Chapter 3.1 System configuration.
2.   Use serial communication software (Tera Term, etc.), select the USB COM port connected to EZ-0012, and then connect. The connection settings are as follows.

Table 4-3 Serial communication connection settings

Item	Setting contents
Communication baud rate	9600
Data bits	8bit
Parity	none
Stop bit	1bit
Flow control	none

3.   After building the sample application, download it to the debugger and run it for debugging.
4.   After launching the DALI Master Controller GUI, select the USB COM port connected to the Application Controller and connect it.
5.   Operate the DALI master controller GUI and confirm that the lighting status of the lamp changes.
     e.g. 1: To turn on all connected lamps at the maximum light intensity -> Select Broadcast and press the MAX button.
     e.g. 2: Turn off all connected lights -> Select Broadcast and press the OFF button.

(2)   In the case of the DALI102ed.2.0+207ed.1.0 sample application

6. After executing 4.2(1) Common operating procedures, set the edition of the DALI master controller GUI to 102ed.2.0 and open Manual Command.
7. Transmit a command to change the fade time.
    e.g. Change the fade time to 400ms after setting it to 4s
    (1) Set fade time to 4s

    - Transmit RESET
    - Transmit DTR0(6)
    - Transmit SET FADE TIME
    - Transmit the DAPC command and check that the lighting status changes over 4s.

    After setting the edition of the DALI master controller GUI to 102ed.1.0+207ed.1.0, open the Manual Command.
    (2) Set Fast fade time to 400ms

    - Transmit RESET
    - Transmit DATA TRANSFER REGISTER(DTR)(16)
    - Transmit ENABLE DEVICE TYPE X(6)
    - Transmit STORE DTR AS FAST FADE TIME
    - Transmit the DAPC command and confirm that the lighting status changes in 400ms.

8. To check the LED failure status, press the key described in "4.3.3.2(2) Debugging feature (pseudo-error generation feature)" to update the failure status.
    e.g. Transmit q via serial and set SHORT CIRCUIT to true
9. Transmit "QUERY XXXX CIRCUIT" in the DALI master controller GUI.
    e.g. After transmitting "ENABLE DEVICE TYPE X(6)", transmit "QUERY SHORT CIRCUIT" in DALI master controller GUI
10. Confirm that the response results in the Command Log of the DALI master controller GUI have changed according to the updated fault status.

(3)   In the case of the DALI102ed.2.0+209ed.1.0 sample application

6. After executing 4.2(1) Common operating procedures, open Manual command (By Code) from Send Code... in DALI master controller GUI.
7. Transmit a command to change the color temperature (Tc).
    e.g. Change the color temperature (Tc) to a warmer color by 100 [K]

    - Transmit DTR0 (100):
      0b10100011 0b01100100
    - Transmit ENABLE DEVICE TYPE X(8):
      0b11000001 0b00001000
    - Transmit STORE COLOUR TEMPERATURE Tc STEP INCREMENT via Broadcast Twice:
      0b11111111 0b11101111
    - Transmit ENABLE DEVICE TYPE X(8):
      0b11000001 0b00001000
    - Transmit COLOUR TEMPERATURE Tc STEP WARMER via Broadcast:
      0b11111111 0b11101001

8. Confirm that the lighting status of the lamp is variable.
9. On the serial output, confirm that the variable value is output according to the above.

For other detailed operations of the DALI master controller GUI, refer to the DALI master controller GUI user's manual (R20UT0715).

For details on commands, etc., refer to each DALI standard in Chapter 7 Reference Documents.

4.3 Overview of Features

4.3.1 Feature block diagram

The feature block diagram for each sample application is shown below.
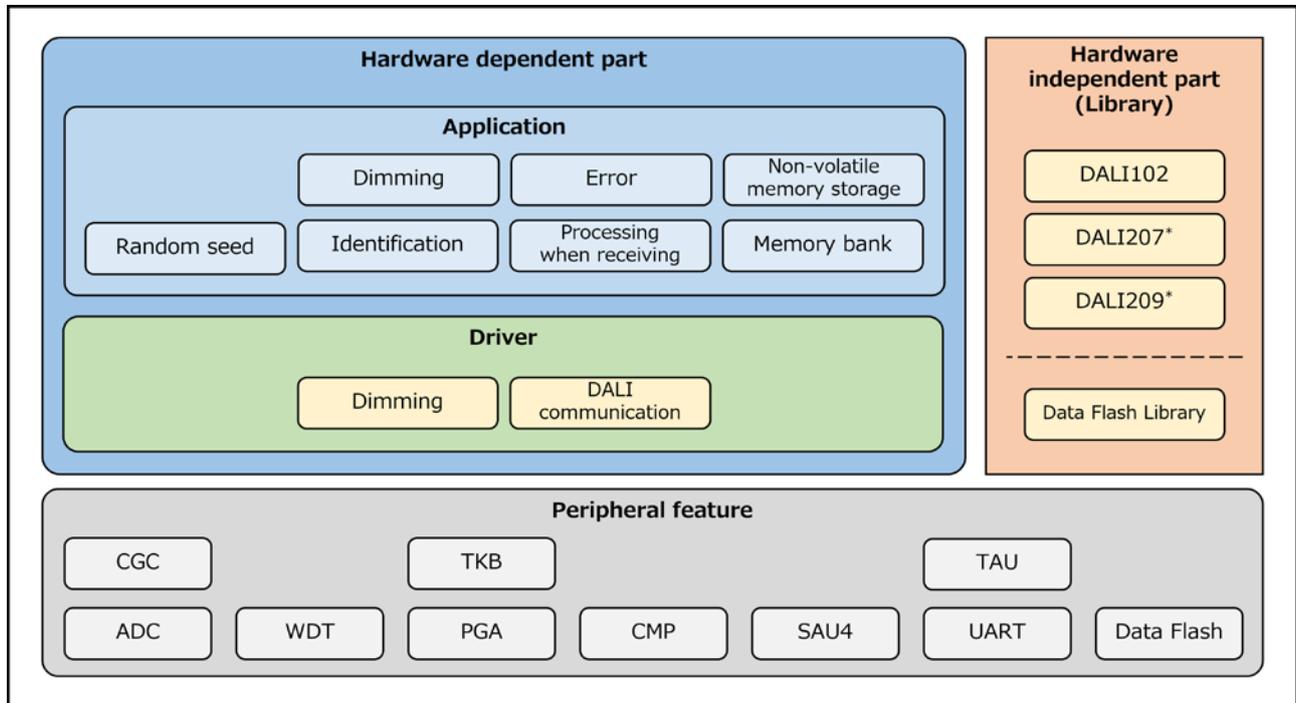
* The DALI207 library is included only in the DALI102ed.2.0+207ed.1.0 sample application, and the DALI209 library is included only in the DALI102ed.2.0+209ed.1.0 sample application.

Figure 4-6 Feature block diagram

4.3.2 Hardware independent part

The hardware-independent part is supported by the DALI library. Refer to the user's manual of each library for details.

The data flash library is used for non-volatile memory storage. For details of the data flash library, refer to RL78 Family Data Flash Library Type04 Package Ver.2.00 Release Note (R20UT0749).

4.3.3 Hardware dependent part

The hardware dependent part is implemented to match the hardware configuration of the EZ-0012+EZ-0012 expansion board. when developing DALI products, please consider and change the operation to match the standard and purpose of use and evaluate it thoroughly.

The features implemented in each sample application are shown in the following chapters.

4.3.3.1 Common features of each sample application

(1) Dimming driver

It is a driver that dims the LED lamps that are the output of the Control Gear. It uses the timer KB feature, PGA feature, and comparator feature to control the lighting on and off of each LED. There are three LEDs for dimming, and the driver acquires the status of each lamp and instructs dimming.

(2) DALI communication driver

It is a driver that meets the DALI communication specifications required by IEC62386-101ed.2.0.

On the transmission side, the TAU feature and the DALI feature of SAU4 are used to encode the transmitted data into Manchester code and to transmit backward frame and corrupted backward frame.

On the reception side, the TAU feature is used to capture and decode the received frames in Manchester. It also detects system failures on the DALI bus.

(3) Non-volatile memory storage

As for the variables that need to be saved to the data flash, they will be saved to the data flash at the following timing.

- When 1s has elapsed since the change request from the library to the variable
- When a save request by a receiving Frame is received

In addition to the above, the DALI standard recommends saving the data when the power is turned off. In this sample application, the board does not support the power-off detection, so it is designed to perform the saving process periodically.

To access the data flash, use the data flash library Type04. The data flash consists of 4 blocks (blocks 0 to 3), where blocks 0 and 1 are allocated for storing frequently updated data and blocks 2 and 3 are allocated for storing infrequently updated data.

The block configuration is shown in Figure 4-7.
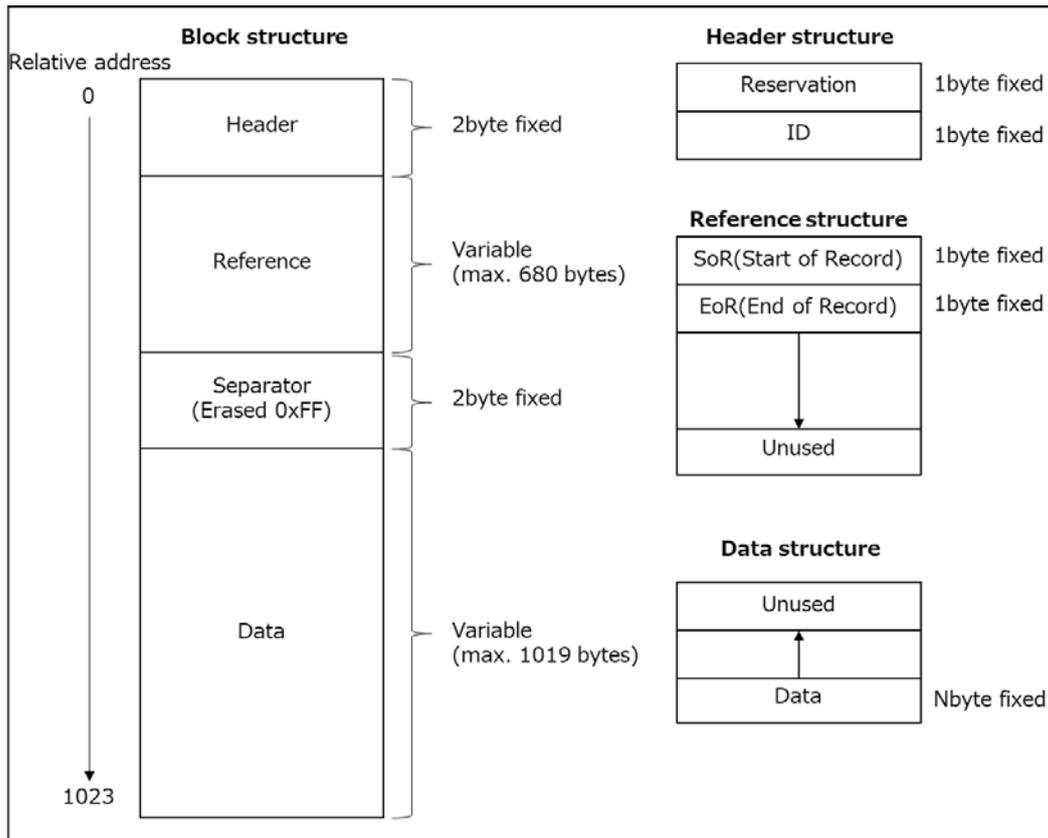
Figure 4-7 Block structure diagram

(a) Header structure

The header part is assigned an ID, and the block to be written or erased is selected from the value.

Table 4-4 Header structure

Header	Valid value	Invalid value
ID	0 or 1	Values other than valid values

If both even and odd blocks are valid in blocks 0 and 1, which are updated frequently, and blocks 2 and 3, which are updated infrequently, the following rule is used for block selection.

● If the odd and even blocks have the same ID: Select the odd block
● If the odd and even blocks have different ID: Select the even block

(b) Reference structure

Set SoR when writing data and EoR when writing is completed.

Use each value at startup to get the location where the latest data is stored.

Table 4-5 Reference structure

Reference	Valid value	Invalid value
SoR	0x12: Start writing 0xFF: Not used	Values other than valid values
EoR	0x34: Write completed 0xFF: Not completed	Values other than valid values

(c)　Data structure

Data is written from the lower address to the upper address.

(d)　Priority of non-blocking processes

Initialization and reading are blocked, but writing and erasing are non-blocking. The priority at that time is as follows.

1.　Writing data in high-frequency block
2.　Writing data in low-frequency block
3.　Erasing data from high-frequency block
4.　Erasing data from low-frequency block

(e)　Erase trigger

Erase blocks that are not valid during initialization. Erase the other block when the first write succeeds.

(4)　Memory bank

It implements the memory banks specified in the standard and the processes to store and manage the memory banks. In each sample application, the memory bank required by the standard is defined for each logical unit and managed.

(5)　Processing when Frame is received

The receive Frame received from the DALI communication driver is passed to the DALI library function, and if necessary, a backward frame is transmitted using the DALI communication driver within the valid period.

If it is necessary to transmit a corrupted backward frame, it notifies the DALI communication driver of the request to transmit the corrupted backward frame and transmits it.

(6)　Dimming process

It gets the dimming amount according to the status of each logical unit from the DALI library and reflects it to the lamp via the dimming driver. During the identification operation, blinking is processed. If an error occurs, it notifies the DALI library of the error and turns off each LED depending on the content of the error.

(7)　Identification process

Each sample application flashes the LED of the corresponding logical unit during the identification operation mode.

(8) Error process

It detects errors that occur during software operation and notifies the DALI library functions depending on the contents of the error. In this sample application, an error is generated by the debug feature (Pseudo-anomaly generation feature).

- About the pseudo-anomaly generation feature
  Normally, failure detection and recovery would be notified to the DALI library based on the status of the lamp to manage the failure status, but since the lamp is integrated and does not support power-off detection, failure detection and recovery detection are generated pseudo-actively. Fault detection and recovery detection are set by transmitting the relevant operation key via serial transmission. When the relevant key is pressed, the set contents are output to the serial communication monitor connected to the PC as a response result.
  The following table shows the faults that can be set for each sample application.

Table 4-6 List of faults common to each sample application

Item	Related failure	Operation key	Description
control gear failure	CONTROL GEAR FAILURE	0 , 1	Device failure 0: Device failure state is cleared. 1: Set to device failure state
failure procedure	SHORT CIRCUIT	-*	Short circuit detection*
	OPEN CIRCUIT	-*	Open circuit detection*
	All failures of failure status	c	Clear all failures

* Since the corresponding keys and contents are different in each sample application, refer to the pseudo-anomaly generation feature of the sample application used.

Note 1: If an unconfigured key is transmitted, "unknown key" will be displayed on the serial output screen.

Note 2: The control gear failure setting is reflected in the controlGearFailure bit of the 102 standard. This setting is not stored in the data flash memory (NVM).

Note 3: The failure procedure setting represents the physical operation required for testing.
When this is operated, it will be reflected in the lampFailure, failure status, etc., considering the settings of the actualLevel and current protector managed by the library. This setting is saved in NVM.

(9) Generation of random numbers and seed values

It passes the seed value, which is the source of the random number used in the DALI library, during initialization and reflects it in the processing.

In each sample application, the seed value is generated from the following at the timing of the Control Gear initialization process.

- Value of uninitialized RAM area
- AD value of specified voltage
- AD value of temperature sensor

4.3.3.2   DALI102ed.2.0+207ed.1.0 Sample Application Specific Features

The features supported in 207ed.1.0 are listed below.

(1)   About failure status

In addition to the 4.3.3.1(8) Error process, 207ed.1.0 optionally detects the following faults

- SHORT CIRCUIT
- OPEN CIRCUIT
- LOAD DECREASE
- LOAD INCREASE
- CURRENT PROTECTOR
- THERMAL SHUT DOWN
- THERMAL OVERLOAD
- REFERENCE SYSTEM POWER FAILED

(2) Debugging feature (pseudo-error generation feature)

207ed.1.0 manages the following faults in addition to those described in Section 4.3.3.1(8) Error process.

Set/resolve the fault status using the uppercase/lowercase letters of the corresponding operation keys. The table below shows the errors assigned to the keys.

Table 4-7 List of operation keys for 207ed.1.0 and their corresponding failures

Item	Related failure	Operation key	Description
failure procedure	SHORT CIRCUIT	q , Q	Short circuit detection q: Set all outputs to short-circuit state Q: Clear the short-circuit state of all outputs
	OPEN CIRCUIT	w , W	Open-circuit detection w: Set all outputs to open-circuit state W: Clear the open-circuit state of all outputs
	LOAD DECREASE	e , E	Load reduction detection e: Disconnect some loads E: Reconnect some loads
	LOAD INCREASE	r , R	Load increase detection r: Connect additional load R: Disconnect the added load
	CURRENT PROTECTOR	t , T	Current protection enabled state t: Set the occurrence of underload/overload. T: Clear underload/overload state
	THERMAL SHUT DOWN	y , Y	Overheat protection detection y: Detect overheat (overheating of load) Y: Clear overheat condition
	THERMAL OVERLOAD	u , U	Low light level detection due to overheating u: Detect overheat (stop load) U: Clear overheat condition
	REFERENCE SYSTEM POWER FAILED	i , I	Reference measurement result of LED load i: Set measurement result = failure I: Set measurement result = success
	All failures in failure status	c	Clear all errors
	End of reference measurement	f	Finish the reference measurement of LED load

Note 1: The reference measurement process can be completed immediately. If no action is taken, the process will be completed in 15 minutes. The measurement result is set by the failure procedure setting (i , I).

Note: In this application note, a pseudo generation function is implemented to check the error handling. When implementing it in an actual product, refer to the DALI standard document: 102ed.2.1 and 207ed.1.0 in Chapter 7 Reference Documents and implement it.

4.3.3.3 DALI102ed.2.0+209ed.1.0 Sample Application Specific Features

The features supported in 209ed.1.0 are listed below.

(1) LED color adjustment feature

In this sample application, two LEDs (one in the low temperature range and one in the high temperature range) are used to represent color control for one Logical unit.

As an example, when the color temperature (Tc) is low, set the dimming ratio of the low-temperature LEDs to high and that of the high-temperature LEDs to low. The dimming ratio of each LED will change as the color temperature (Tc) changes, but the sum of the dimming ratio of each LED should always be set to match the dimming ratio specified in the actual level.

The following diagram shows the relationship between color temperature (Tc) and each temperature range LED, and the relationship between total dimming ratio (actual level) and color temperature (Tc).
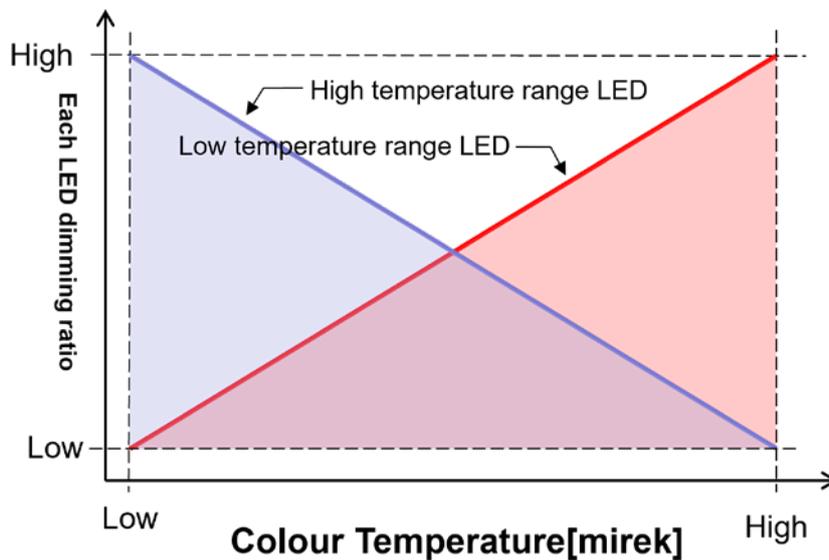
Figure 4-8 Relationship between color temperature and each temperature range LED
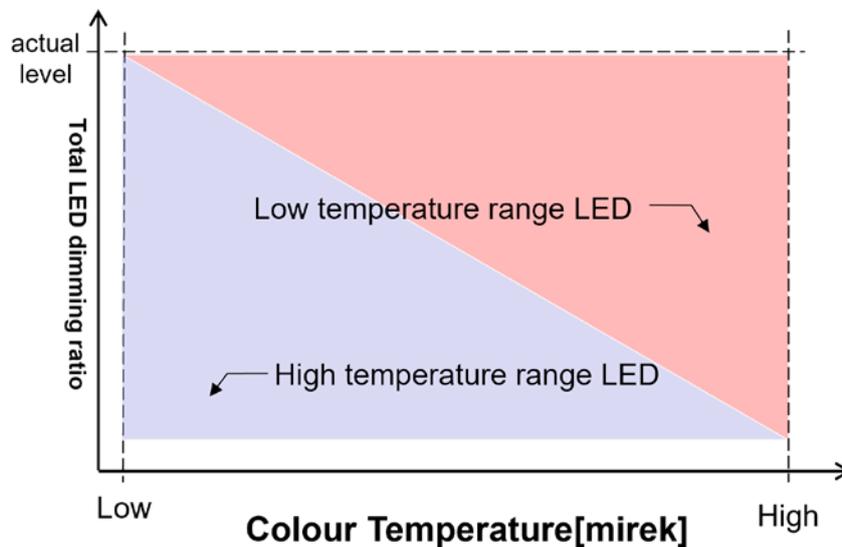
Figure 4-9 Relationship between total dimming ratio (actual level) and color temperature (TC)

The dimming ratio of each LED also changes depending on the actual level, and although the dimming ratio of each LED changes as the actual level changes, the ratio of the dimming ratio between low and high temperature range LEDs does not change.
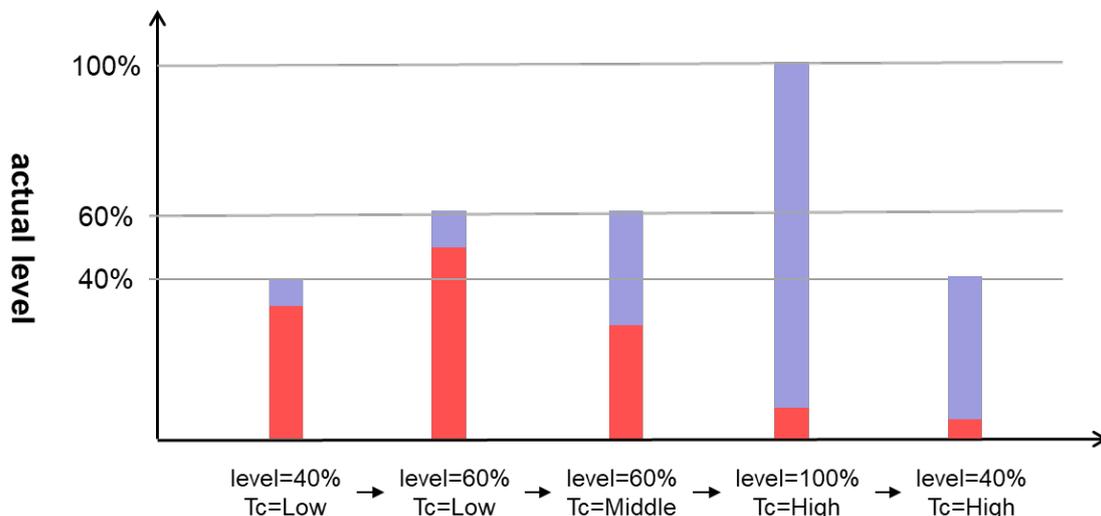
Figure 4-10 Ratio of the total dimming ratio (actual level) to the dimming ratio of each temperature range LED

The PHM is specified as the minimum physical dimming ratio for LEDs. When using the dimming/color control driver in this sample application, it is necessary to set a different value from the PHM when 209ed.1.0 is not extended. The reason for this is that the dimming ratio set at the actual level is the sum of the two dimming ratios of the low and high temperature range LEDs, so if the PHM is set at the actual level, at least one of the LEDs will have a dimming ratio below the PHM.

In addition, 209ed.1.0 specifies physical limits of color temperature (TC) called physical coolest and physical warmest. In this sample application, it is necessary to set three variables, PHM, physical coolest, and physical warmest, so that each LED does not fall below the minimum physical dimming ratio within the valid actual level range

The minimum dimming ratio (PHM) of each LED in EZ-0012 is 70 (0.658%), and this sample application uses the following settings for the three variables to meet the physical limitations.

- PHM: 170(10.091%)
- physical coolest: 0x3FFFF
- physical warmest: 0xBFFF

Colour Temperature Tc	0x0001	~	0x3FFF (25% of MAX)	~	0x7FFF	~	0xBFFF (75% of MAX)	~	0xFFFE
actual level(PHM)	170(10.091%)								
High temperature LED dimming ratio	10.091% (100%*10.091%)	~	7.568% (75%*10.091%)	~	5.0455% (50%*10.091%)	~	2.522% (25%*10.091%)	~	0% (0%*10.091%)
Low temperature LED dimming ratio	0% (0%*10.091%)	~	2.522% (25%*10.091%)	~	5.0455% (50%*10.091%)	~	7.568% (75%*10.091%)	~	10.091% (100%*10.091%)
LED dimming ratio (Low temperature LED + High temperature LED)	10.091%	~	10.091%	~	10.091%	~	10.091%	~	10.091%

Figure 4-11 Each dimming ratio operation table that meets the physical limitations in this sample

(2) Debugging feature (dimming/color control monitoring feature)

Serial output is used to check the change in color temperature (Tc) as a numerical value.

It checks the actual level and color temperature (TC) at 100ms intervals and outputs the following only when there is a change in one of the values.

```
(Level, Tc) = (150, 1000)

(Level, Tc) = (150, 1020)

(Level, Tc) = (145, 1000)
            .
            .
            .
```
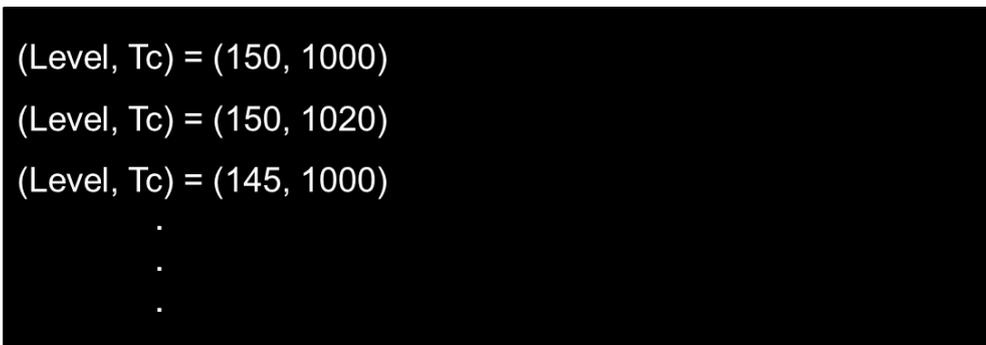
Figure 4-12 Serial output screen

(3) Debugging feature (pseudo-error generation feature)

In 209ed.1.0, the fault status is set/resolved by using the uppercase/lowercase letters of the corresponding operation keys as in chapter 4.3.3.1(8) Error process. The table below shows the errors assigned to the keys.

Table 4-8 List of operation keys for 209ed.1.0 and their corresponding failures

Item	Related failure	Operation key	Description
failure procedure	SHORT CIRCUIT	q , Q a , A	Short circuit detection q: Set the output of the high temperature area lamp to short circuit status Q: Clear the short-circuit in the output of the high-temperature lamp a: Set the output of the low temperature area lamp to short circuit A: Clear the short-circuit in the output of the low-temperature lamp
	OPEN CIRCUIT	w , W s , S	Open circuit detection w: Set the output of the high temperature range lamp to open circuit state. W: Clear the open-circuit state of the output of the high temperature range lamp. s: Set the output of the low temperature lamp to open circuit. S: Clear the open-circuit state of the output of the low temperature range lamp.
	All failures in failure status	c	Clear all errors

Note: In this application note, a pseudo generation function is implemented to check the error handling. When implementing it in an actual product, refer to the DALI standard document: 102ed.2.1 and 209ed.1.0 in Chapter 7 Reference Documents and implement it.

4.4 Software Configuration

The software configuration for each sample application is described below.

4.4.1 Folder Structure

The folder structure for this sample application is shown below.

Table 4-9 Folder structure (1/3)

Folder, File name	Description
DALI102_20X_sample	Sample application folder
¥App\<DIR>	Application program storage folder
r_cg.c	Source file for the Control Gear module
r_cg.h	Header file for the Control Gear module
r_common.h	Header file for common definitions
r_debug.c	Source file for debugging modules
r_debug.h	Header file for debugging modules
r_lamp.c[*1]	Source file for dimming module[*1]
r_lamp.h[*1]	Header file for dimming module[*1]
r_lamp_tc.c[*2]	Source file for 209ed.1.0 color temperature (TC) compatible dimming module[*2]
r_lamp_tc.h[*2]	Header file for 209ed.1.0 color temperature (TC) compatible dimming module[*2]
r_main.c	Main source file
r_memory_bank.c	Source file for memory bank module
r_memory_bank.h	Header file for memory bank module
r_memory_banks.c	Source file for modules that manage multiple memory banks
r_memory_banks.h	Header file for modules that manage 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 module
r_random.h	Header file for random number generation module
r_unit0_memory_bank.c	Source file for unit0 definition of memory bank
r_unit0_memory_bank.h	Header file for unit0 definition of memory bank
r_unit1_memory_bank.c	Source file for unit1 definition of memory bank
r_unit1_memory_bank.h	Header file for unit1 definition of memory bank
r_unit2_memory_bank.c	Source file for unit2 definition of memory bank
r_unit2_memory_bank.h	Header file for unit2 definition of memory bank
¥Driver\<DIR>	Folder to store driver program
r_adc.c	Source file for ADC
r_adc.h	Header file for ADC
r_cgc.c	Source file for CGC
r_cgc.h	Header file for CGC
r_comp.c	Source file for the comparator
r_comp.h	Header file for the comparator
r_dali101.c	Source file for DALI101ed.2.0 supported drivers
r_dali101.h	Header file for DALI101ed.2.0 supported drivers
r_dali101_bft.c	Source file for DALI101ed.2.0 supported drivers (transmission)
r_dali101_bft.h	Header file for DALI101ed.2.0 supported drivers (transmission)
r_dali101_rx.c	Source file for DALI101ed.2.0 supported drivers (reception)
r_dali101_rx.h	Header file for DALI101ed.2.0 supported drivers (reception)

Table 4-10 Folder structure (2/3)

Folder, File name	Description
DALI102_20X_sample	Folder for sample applications
¥Driver<DIR>	Folder to store driver program
r_led.c	Source file for the dimming driver
r_led.h	Header file for the dimming driver
r_led1.c	Source file for LED1 dimming driver
r_led1.h	Header file for LED1 dimming driver
r_led2.c	Source file for LED2 dimming driver
r_led2.h	Header file for LED2 dimming driver
r_led3.c	Source file for LED3 dimming driver
r_led3.h	Header file for LED3 dimming driver
r_macrodriver.h	Header file for macro driver
r_pga.c	Source file for PGA
r_pga.h	Header file for PGA
r_port.c	Source file for PORT
r_port.h	Header file for PORT
r_ram.c	Source file for seed value generation for DALI library
r_ram.h	Header file for seed value generation for DALI library
r_serial.c	Source file for SAU
r_serial.h	Header file for SAU
r_systeminit.c	Source file for initialization
r_tau.c	Source file for TAU
r_tau.h	Header file for TAU
r_tkb.c	Source file for 16-bit timer KB
r_tkb.h	Header file for 16-bit timer KB
r_wdt.c	Source file for WDT
r_wdt.h	Header file for WDT
¥generate<DIR> (CC-RL)	Folder to store project products (e2 studio development environment only)
cstart.asm	Assembler file for startup routine definition
hdwinit.asm	Assembler file for peripheral device initialization process
iodefine.h	Header file for I/O register definition
stkinit.asm	Assembler file for stack area initialization
¥HardwareDebug<DIR> (CC-RL)	Folder to store project output file (e2 studio development environment only)
DALIXXX_sample.mot	Motorola S-type format file used to write to firmware
¥Debug¥Exe<DIR> (IAR)	Folder to store project output file (IAR development environment only)
DALIXXX_sample.mot	Motorola S-type format file used to write to firmware

Table 4-11 Folder structure (3/3)

DALI102_20X_sample				Sample application folder
	¥Library<DIR>			Folder to store library
		¥DALI102<DIR>		Folder to store DALI102ed.2.0 library
			r_dali_102_cc_gen2_v1_00.lib (CC-RL) r_dali_102_iar_gen2_v1_00.a (IAR)	Library file to support DALI102ed.2.0
			r_dali102_api.h	Header file to support DALI102ed.2.0 library
			r_dali102_xxxx.h	Header file to support DALI102ed.2.0 library (used in the library)
		¥DALI207<DIR>[*1]		Folder to store DALI207ed.1.0 library[*1]
			r_dali_207_cc_gen2_v1_00.lib (CC-RL) r_dali_207_iar_gen2_v1_00.a (IAR)	Library file to support DALI207ed.1.0[*1]
			r_dali207_api.h	Header file to support DALI207ed.1.0 library[*1]
			r_dali207_xxxx.h	Header file to support DALI207ed.1.0 library (used in the library) [*1]
		¥DALI209<DIR>[*2]		Folder to store DALI209ed.1.0 library[*2]
			r_dali_209_cc_gen2_v1_00.lib (CC-RL) r_dali_209_iar_gen2_v1_00.a (IAR)	Library file to support DALI209ed.1.0[*2]
			r_dali209_api.h	Header file to support DALI209ed.1.0 library[*2]
			r_dali209_xxxx.h	Header file to support DALI209ed.1.0 library (used in the library) [*2]
		¥FDL<DIR>		Folder containing the data flash library from Renesas Electronics. * In this directory and below, it is necessary to download the data flash library from Renesas Electronics and register the library file. For details, please refer to chapter 4.1.3 and 4.1.6.
	¥Utility<DIR>			Folder for common programs
		r_timer16.c		Source file for timer module
		r_timer16.h		Header file for timer module
		r_typedefs.h		Header file for type definition

*1: It is only stored in the 207 sample application.
*2: It is only stored in the 209 sample application.

4.4.2 List of Option Byte Settings

The option byte settings for each sample application are shown below.

Table 4-12 List of Option Byte Settings

Address	Set value	Description
000C0H / 010C0H	01110001B	Watchdog timer operation permitted (counting starts after reset is released)
000C1H / 010C1H	01110011B	LVD reset mode (falling edge: 3.98V, rising edge: 4.06V)
000C2H / 010C2H	11101011B	HS mode, high-speed on-chip oscillator: 4MHz
000C3H / 010C3H	10000100B	On-chip debug permitted

4.4.3 Flowchart
4.4.3.1 Main Flow

The main flow of each sample application is shown below.

After the initialization process is complete, the Control Gear process is executed.
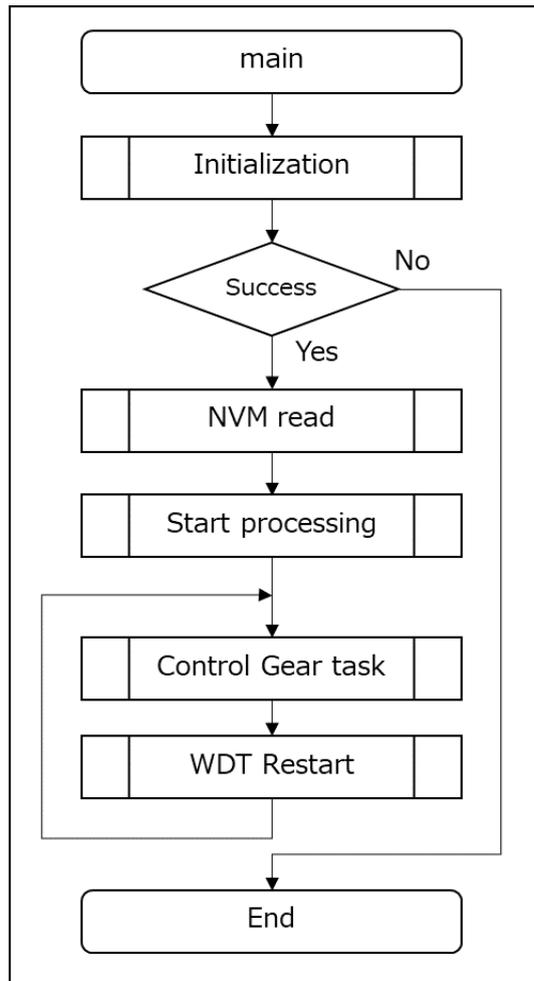
Figure 4-13 Main flow diagram

4.4.3.2 Initialization

The flow of the initialization process is shown below.
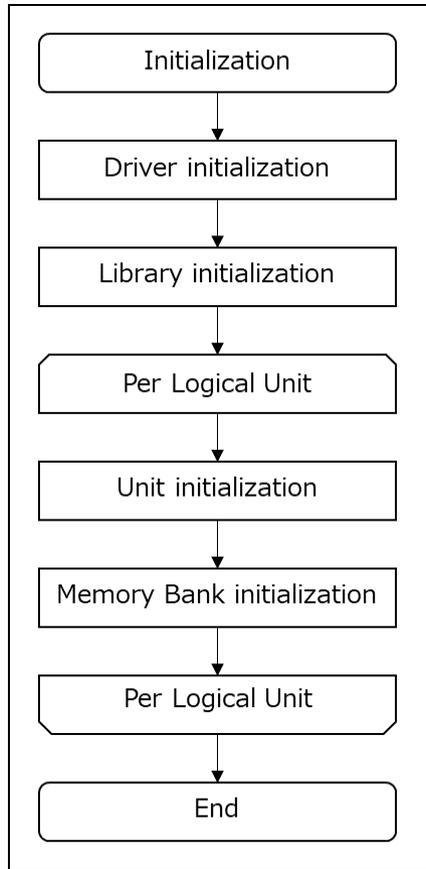
Figure 4-14 Initialization flow diagram

4.4.3.3 Reading NVM

The flow of the NVM reading process to be executed after the initialization process is shown below.
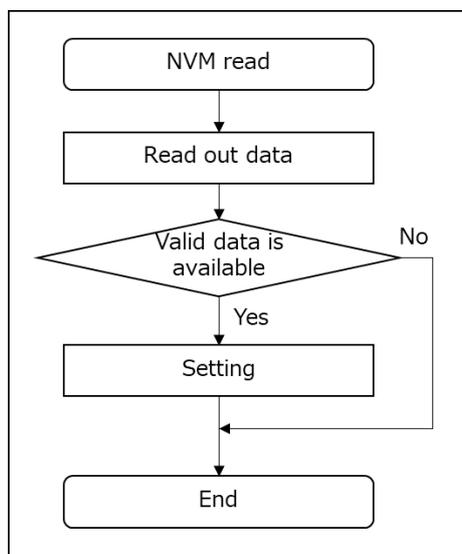
Figure 4-15 NVM reading process flow diagram

4.4.3.4 Start Processing

The flow of the control gear start process is shown below.
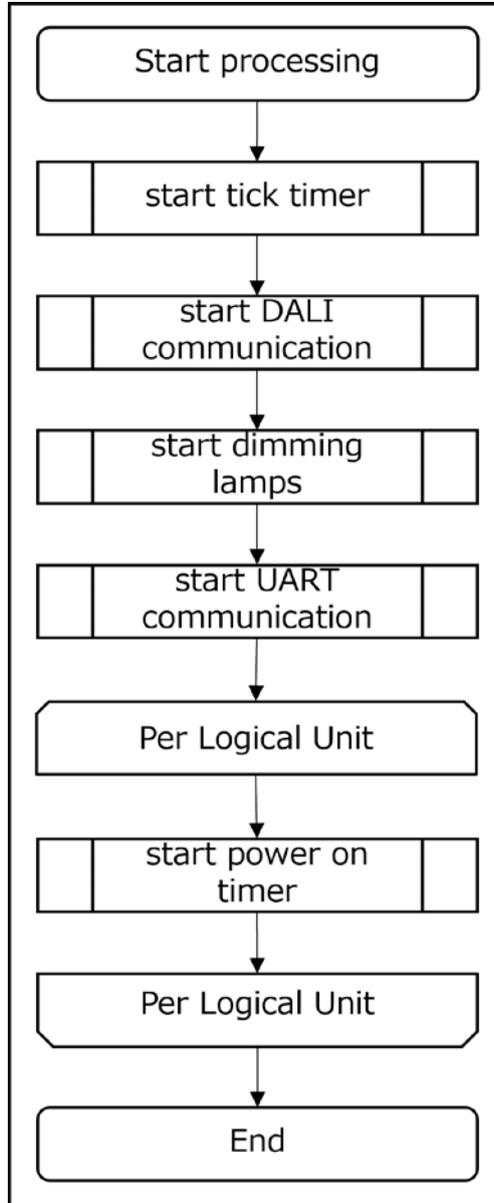
Figure 4-16 Start processing flow diagram

4.4.3.5 Control Gear Task Processing

The flow of the Control Gear task processing is shown below.
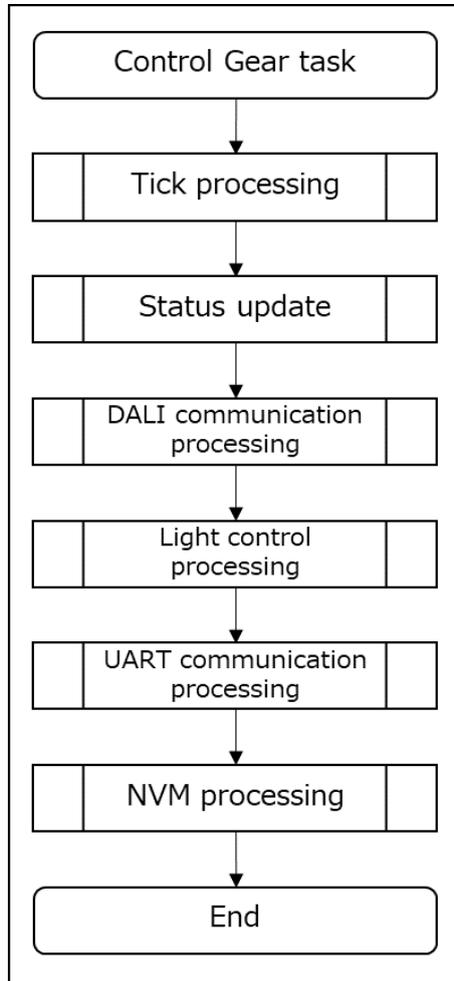
Figure 4-17 Control Gear task processing flow diagram

5. Appendix

5.1 DALI driver

The outline of the DALI driver implemented in this sample application is shown below.

5.1.1 Backward Frame Transmitter (BFT)

Implement using the DALI function of SAU4. Since the settling time (5.5ms to 10.5ms) from receiving the Forward Frame to sending the Backward Frame does not require accuracy, the waiting time until sending is adjusted with the software timer in 1ms units.

5.1.1.1 Reproduction of Corrupted Backward Frame

The interval timer is used to force the DALI bus to go low for a certain period of time (1300us to 2000us) at the same time as the transmission starts, and the damaged Backward Frame is reproduced. To prevent the Low time from changing, the transmission data is fixed at 0x00.

Note: Corrupted Backward Frame is sent when multiple Logical units respond differently. No implementation is required for a single Logical unit.

5.1.2 Receiver (Rx)

Implemented using Input pulse interval measurement and Interval timer. The information acquired by these interrupts is stored in the ring buffer, and bit timing analysis and Manchester code decoding are performed in main.
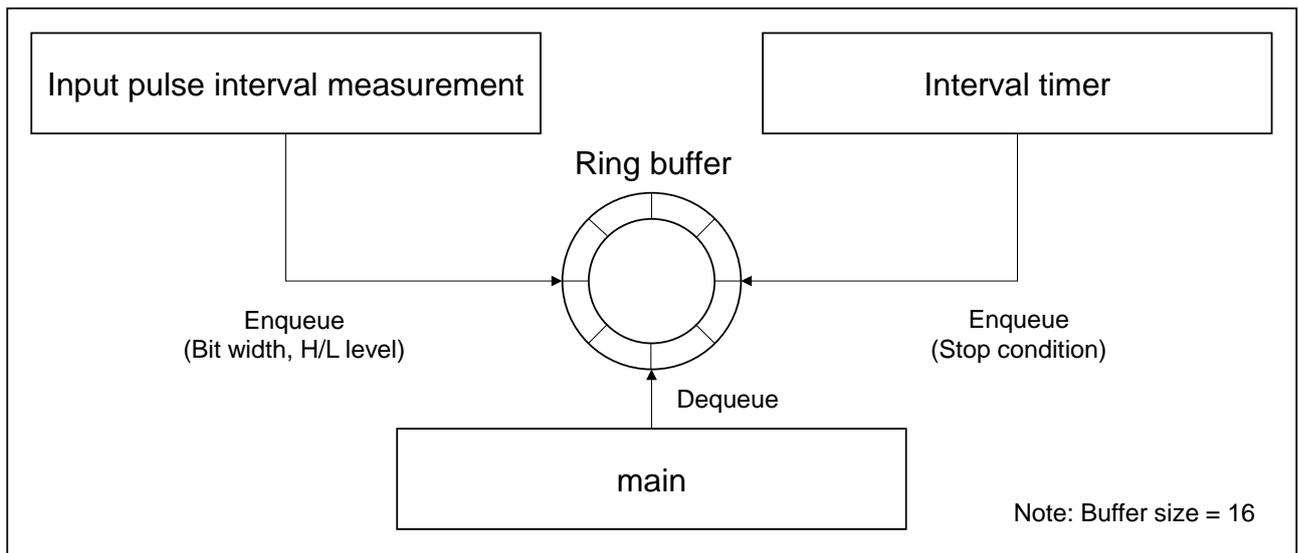
Figure 5-1 Receiver (Rx) overview

5.1.2.1 Interrupt limit

The main process is looped at least every 1ms and should not exceed the buffer size (16). Normally, since the DALI half bit is 416.7us, the input pulse interval measurement interrupt occurs only 2 to 3 times in 1ms. However, in consideration of noise etc., the input pulse interval measurement interrupt is masked for 100us after the interrupt. This limits the input pulse interval measurement interrupts to a maximum of 10 times.

Note: Stop condition is detected after the H level has passed 1900us (1400us-2400us), so it is not necessary to consider noise. (0 to 1 times in 1ms)

5.1.3 Peripheral functions

The peripheral functions used by the DALI driver are shown below.

Table 5-1 Peripheral functions used in the DALI driver

Function	Use
P10/DALITxD4	Transmission process: DALI send function
P11/TI07	Receive processing: Input pulse interval measurement
TAU0-CH2	Receive processing: Interval timer (100us)
TAU0-CH3	Transmission process: Interval timer (1458us)

5.1.3.1 API function

The API function list of the DALI driver and the outline of each function are shown below.

Table 5-2 DALI driver API function list

Function name	Description
R_DALI101_Init	Initialize DALI driver
R_DALI101_Start	Start DALI driver
R_DALI101_Tick1ms	Advance the timer by 1ms
R_DALI101_Task	DALI driver task
R_DALI101_ReceiveFrame	Receive frame
R_DALI101_DetectsSystemFailure	Detect system failure
R_DALI101_SendBackwardFrame	Send backward frame

Table 5-3 R_DALI101_Init

Function	R_DALI101_Init	Initialize DALI driver
Param	void	-
Return	void	-

Table 5-4 R_DALI101_Start

Function	R_DALI101_Start	Start DALI driver
Param	void	-
Return	void	-

Table 5-5 R_DALI101_Tick1ms

Function	R_DALI101_Tick1ms	Advance the timer by 1ms
Param	void	-
Return	void	-

Table 5-6 R_DALI101_Task

Function	R_DALI101_Task	DALI driver task
Param	void	-
Return	void	-

Table 5-7 R_DALI101_ReceiveFrame

Function	R_DALI101_ReceiveFrame	Receive frame
Param	uint32_t * p_data	Frame data
Param	bool * p_twice	Send-twice
Return	int8_t	<0: Error Frame 0: Not received >0: Number of bits in the received frame

Table 5-8 R_DALI101_DetectsSystemFailure

Function	R_DALI101_DetectsSystemFailure	Detect system failure
Param	void	-
Return	bool	true: Detected false: Not detected

Table 5-9 R_DALI101_SendBackwardFrame

Function	R_DALI101_SendBackwardFrame	Send Backward Frame
Param	uint8_t data	Backward Frame data
Return	void	-

6. Notes

- The configuration and operation of the user defined part is designed for the board EZ-0012+EZ-0012 expansion board.
- The operations provided by each sample application are only samples. The user defined part is originally the part where the user decides the operation and implements it, so please change the design and evaluate it according to the purpose of use.
- Please consider the error operation separately according to the user's purpose of use and operation.

7.   Reference Documents

RL78/I1A User's Manual Hardware (R01UH0169J)

RL78 Family User's Manual Software (R01US0015J)

e2 studio 2020-04 and e2 studio v7.8 Getting Started Guide (R20UT4819)

DALI Master Controller GUI User's Manual (R20UT0715)

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

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

Technical Update / Technical News

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

DALI Standards

IEC 62386-101:2014+AMD1:2018 CSV Consolidated version (ed.2.1)
IEC 62386-102:2014+AMD1:2018 CSV Consolidated versioned (ed.2.1)
IEC 62386-207:2009 (ed.1.0)
IEC 62386-209:2011 (ed.1.0)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	10th Jun., 2022	All	First edition issued
1.01	1st Nov. 2022	All	Correct wrong description, Revised several parts related to updating sample software

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)