

RL78/G24

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

Introduction

This application note describes a sample application for DALI (Digital Addressable Lighting Interface) communication using RL78/G24 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/G24.

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/G24

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	4
2. Operating conditions	6
3. Description of the Hardware	7
3.1 System configuration	7
3.2 List of Pins to be Used	8
4. Description of the Software	8
4.1 Environment setup	8
4.1.1 Downloading the Software	8
4.1.2 Installing CS+	9
4.1.3 Installing Renesas Flash Driver Type01	9
4.1.4 Installing the DALI Master Controller GUI	9
4.1.5 Importing the sample application	9
4.1.6 Placing Renesas Flash Driver	10
4.1.7 Building the sample application	11
4.1.8 Hardware connection and setting switch for sample application	11
4.1.8.1 Hardware Connection	12
4.1.9 Debugging the Sample Application	13
4.2 Operating Procedure	14
4.3 Overview of Features	17
4.3.1 Feature block diagram	17
4.3.2 Hardware independent part	17
4.3.3 Hardware dependent part	17
4.3.3.1 Common features of each sample application	18
4.3.3.2 DALI102ed.2.0+207ed.1.0 Sample Application Specific Features	22
4.3.3.3 DALI102ed.2.0+207ed.1.0+209ed.1.0 Sample Application Specific Features	24
4.4 Calibration	33
4.4.1 DALI Receive Waveform Correction	33
4.4.2 DALI Transmit Waveform Correction	34
4.5 Software Configuration	35
4.5.1 Folder Structure	35
4.5.1.1 Sample application for LED dimming control by CPU	36
4.5.1.2 Sample application for LED dimming control by FAA	39
4.5.2 List of Option Byte Settings	42
4.5.3 Flowchart	42
4.5.3.1 Main Flow	42
4.5.3.2 Initialization	43
4.5.3.3 Reading NVM	43
4.5.3.4 Start Processing	44
4.5.3.5 Control Gear Task Processing	45

5. Appendix	46
5.1 DALI driver.....	46
5.1.1 Backward Frame Transmitter (BFT).....	46
5.1.1.1 Reproduction of Corrupted Backward Frame.....	46
5.1.2 Receiver (Rx).....	46
5.1.3 Peripheral functions.....	46
5.1.4 API function	47
6. Notes.....	49
7. Reference Documents.....	49
Revision History	50

1. Specifications

The following two sample applications are described in this application note. There are two types of sample applications, a project that performs LED dimming control with a CPU and a project that performs with FAA (Flexible Application Accelerator).

- 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 RL78/G24 DC/DC LED Control Evaluation Board.

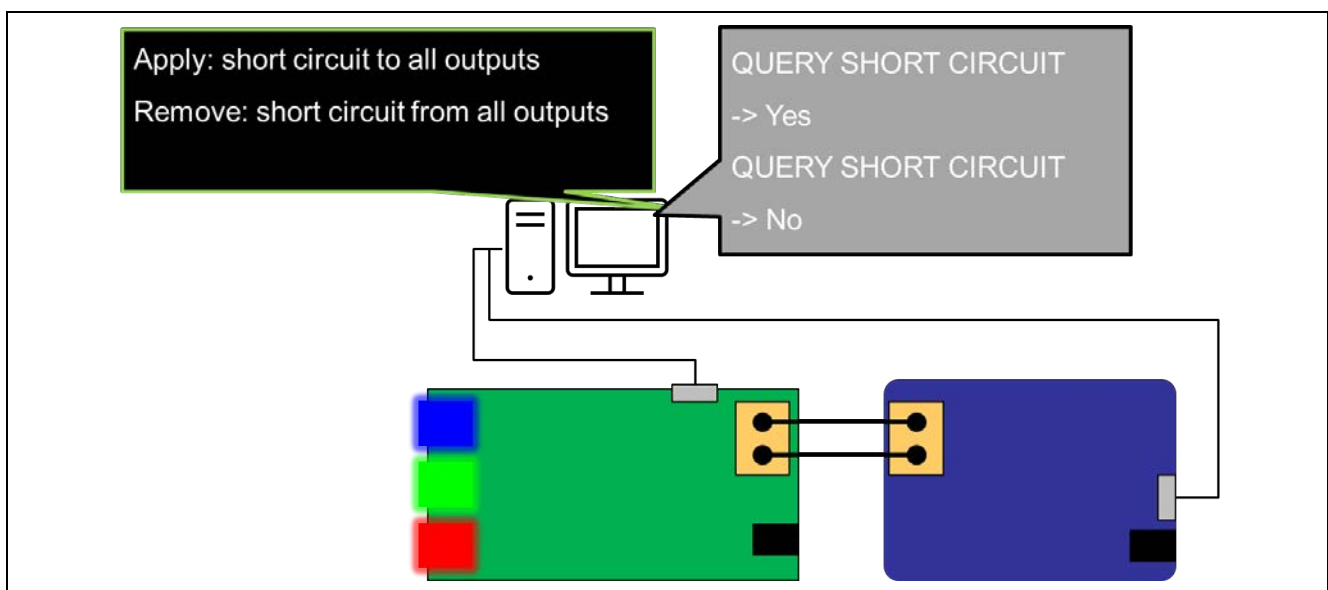


Figure 1-1 102ed.2.0+207ed.1.0 Sample Application Operation Schematic

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+207ed.1.0+209ed.1.0 sample application

This sample application incorporates the Control Gear function (102ed.2.0) with LED light source extensions (207ed.1.0) and colour control extensions (209ed.1.0).

The main LED light source extensions include linear/non-linear dimming curve switching, fade extensions, and LED light source specific features.

The colour control extension mainly defines three types (X,Y / RGBWAF / Tc) of light source dimming/control methods (colour type).

This sample application operates with one logical unit. As an extension of the LED light source, the LED failure status can be set and checked on a PC using the serial communication function on the RL78/G24 DC/DC LED Control Evaluation Board.

As an extension of the colour control, colour temperature (Tc), XY, and RGBWAF are supported, and the colour type can be changed during operation. The LEDs used for each colour type are as follows

Tc: Blue LED (high temperature range) and red LED (low temperature range) are used.

XY: Red LED, green LED, and blue LED are used.

RGBWAF: Enables/disables any of the LED colour adjustments during operation using red, green, and blue LEDs.

The actual level and the colour value of the currently enabled colour mixing method (colour value) are checked every 100ms, and the actual level and colour value are output to the serial communication monitor on the PC only when there is a change in either of these values.

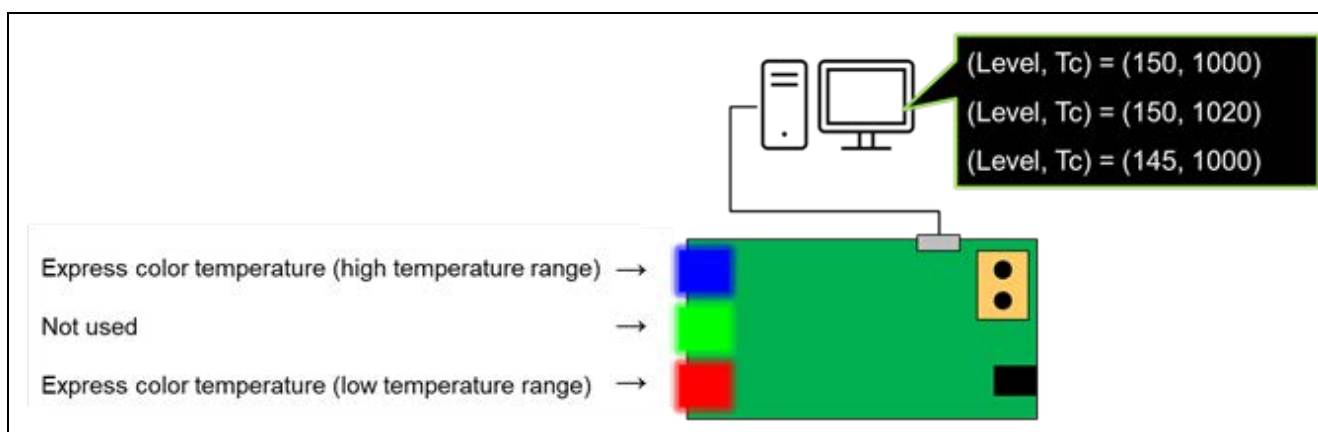


Figure 1-2 102ed.2.0+209ed.1.0 Sample Application (colour type : Tc) Operation Schematic

Note: The serial display operation is only a sample application to check the colour value of Tc , XY or RGBWAF.

Note: The level displayed on the serial communication monitor refers to the actual level.

This application note covers the 102ed.2.0 + 207ed.1.0 combination and the 102ed.2.0 + 207ed.1.0 + 209ed.1.0 combination, but you can also combine the 102ed.2.0 alone or the 102ed.2.0, 209ed.1.0 can also be used in combination.

2. Operating conditions

Each sample application has been tested in the following environments.

Table 2-1 Confirmed operation environment

Item	Description
Microcontroller	RL78/G24 (R7F101GLG)
Operating Frequency	<ul style="list-style-type: none"> • High-speed on-chip oscillator clock (f_{HOCO}): 8MHz • PLL oscillator circuit output (f_{PLL}): 96MHz • Timer KB3 operating clock: $f_{PLL} = 96\text{MHz}$
Operating Voltage	5.0V
Integrated Development Environment (CS+)	Made by Renesas Electronics CS+ for CC 8.10.00
C Compiler (CS+)	Made by Renesas Electronics CC-RL V1.12.01
Integrated Development Environment (IAR)	Made by IAR Systems IAR Embedded Workbench IDE V4.21.1
C Compiler (IAR)	
Library	Made by Renesas Electronics Renesas Flash Driver Type01 V1.10
Emulator	E2 emulator Lite
Board	Made by Renesas Electronics RL78/G24 DC/DC LED Control Evaluation Board RX65N Cloud Kit+DALI-2 Option Board RL78/G23 Lighting Communication Master Evaluation Board
Tool	DALI master controller GUI V3.00. 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 RL78/G24 DC/DC LED Control Evaluation 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, RL78/G23 Lighting Communication Master Evaluation Board or RX65N Cloud Kit + DALI-2 Option Board is used as Application Controller.

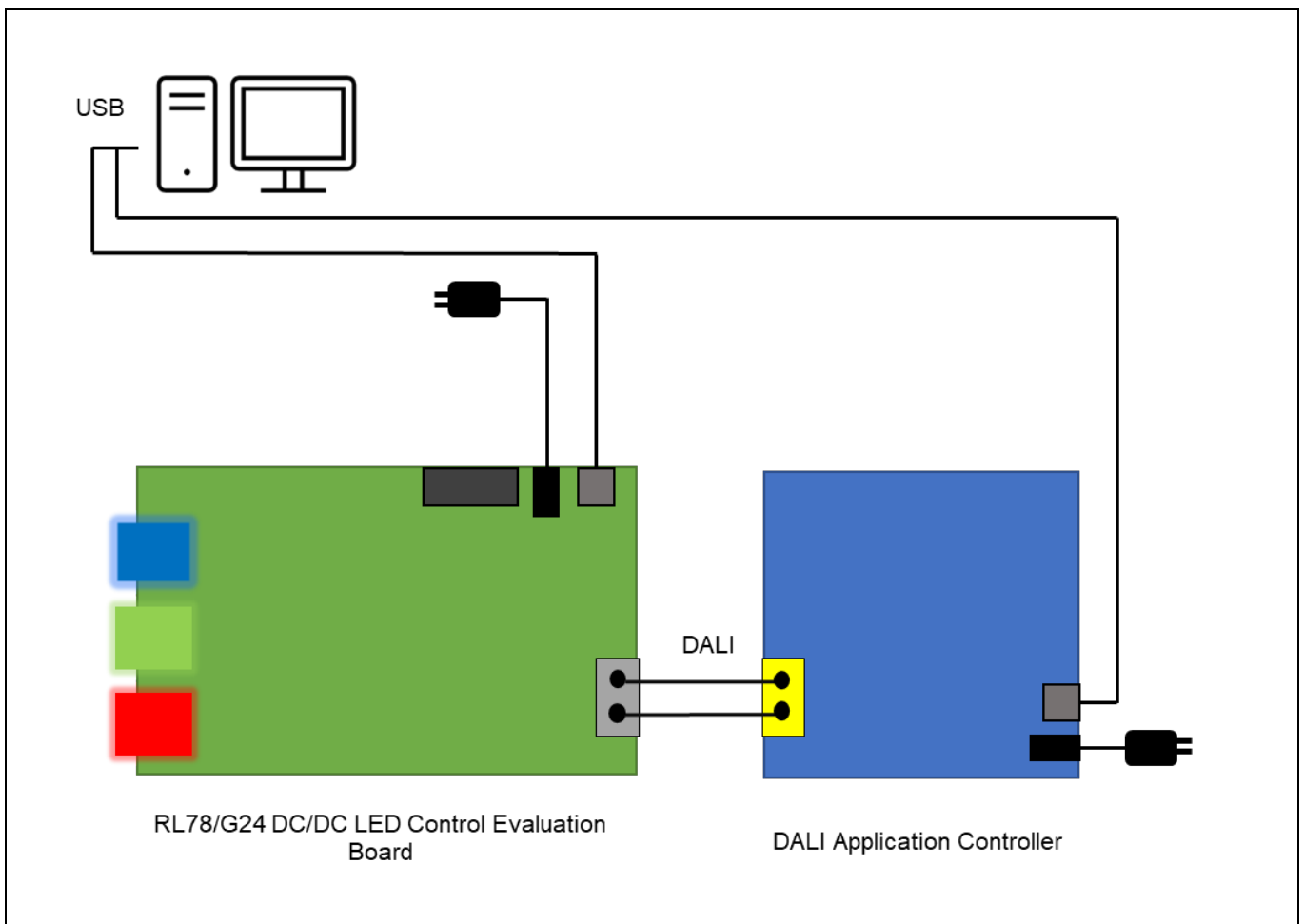


Figure 3-1 Example of the system configuration

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
P02/DALITxD0	Output	DALI communication (Transmission)
P03/DALIRxD0	Input	DALI communication (Reception)
P50/RxD0	Input	UART communication (Reception)
P51/TxD0	Output	UART communication (Transmission)
P12/TKBO00	Output	LED1 lights up
P13/TKBO01	Output	LED2 lights up
P14/TKBO10	Output	LED3 lights up
P120/IVCMP0	Input	LED1 lighting Feedback
P00/IVCMP1	Input	LED2 lighting Feedback
P01/IVCMP2	Input	LED3 lighting Feedback

For other features about RL78/G24 DC/DC LED Control Evaluation Board, refer to the following.

- RL78/G24 DC/DC LED Control Evaluation Board User's Manual (R20UT5371)

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.

- CS+
- Renesas Flash Driver type01 V1.10
- DALI Master Controller GUI V3.00 or later

4.1.2 Installing CS+

CS+ 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.

- CS+ V8.10.00 Integrated Development Environment User's Manual: Project Operation (R20UT5299)

4.1.3 Installing Renesas Flash Driver Type01

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 User's Manual for installation. For details on registering the library files, refer to chapter 4.1.6.

- RL78 Family Renesas Flash Driver RL78 Type01 User's Manual (R20UT4830)

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 the desired location (folder).
2. Start CS+, select "File" tab -> "Open File".
3. Select the MTPJ file in the folder placed in step 1.

4.1.6 Placing Renesas Flash Driver

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

¥DALI102_xxx_sample<DIR>	
¥Library<DIR>	
¥RFD<DIR>	This folder contains the Renesas Flash Driver.
¥userown	Setup according to 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_extension_api.h	
r_rfd_common_get_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	
r_rfd_common_extension_api.c	
r_rfd_common_get_api.c	
¥dataflash	
r_rfd_data_flash_api.c	

4.1.7 Building the sample application

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

4.1.8 Hardware connection and setting switch for sample application

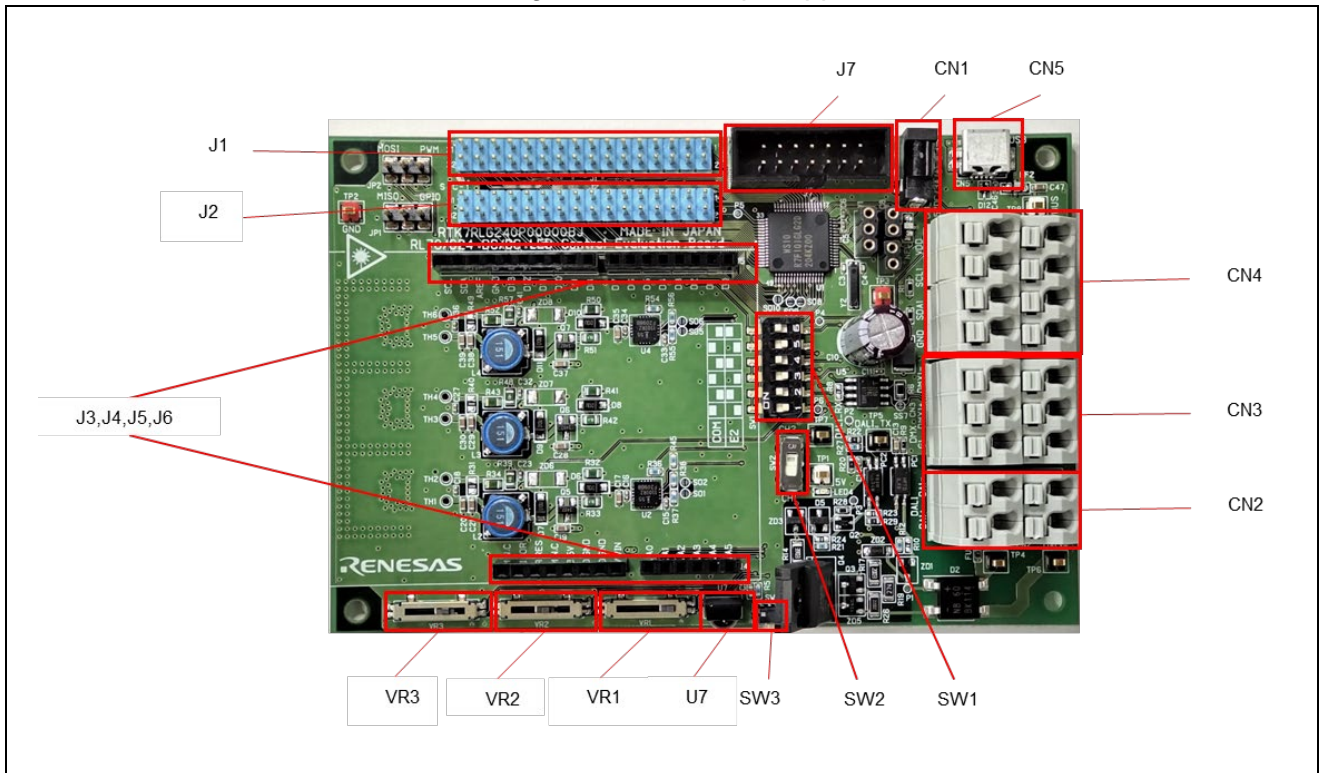


Figure 4-1 RL78/G24 DC/DC LED Control Evaluation Board

The contents of the setting switch (SW1) on the RL78/G24 DC/DC LED Control Evaluation Board must be changed. The contents of the setting switch are shown below.

Table 4-1 Setting switch SW1 (RL78/G24 DC/DC LED Control Evaluation Board)

SW1	Setting
1	OFF
2	ON
3	OFF
4	ON
5	OFF
6	ON
7	OFF
8	ON

4.1.8.1 Hardware Connection

The hardware connections are shown in Figure 4-2.

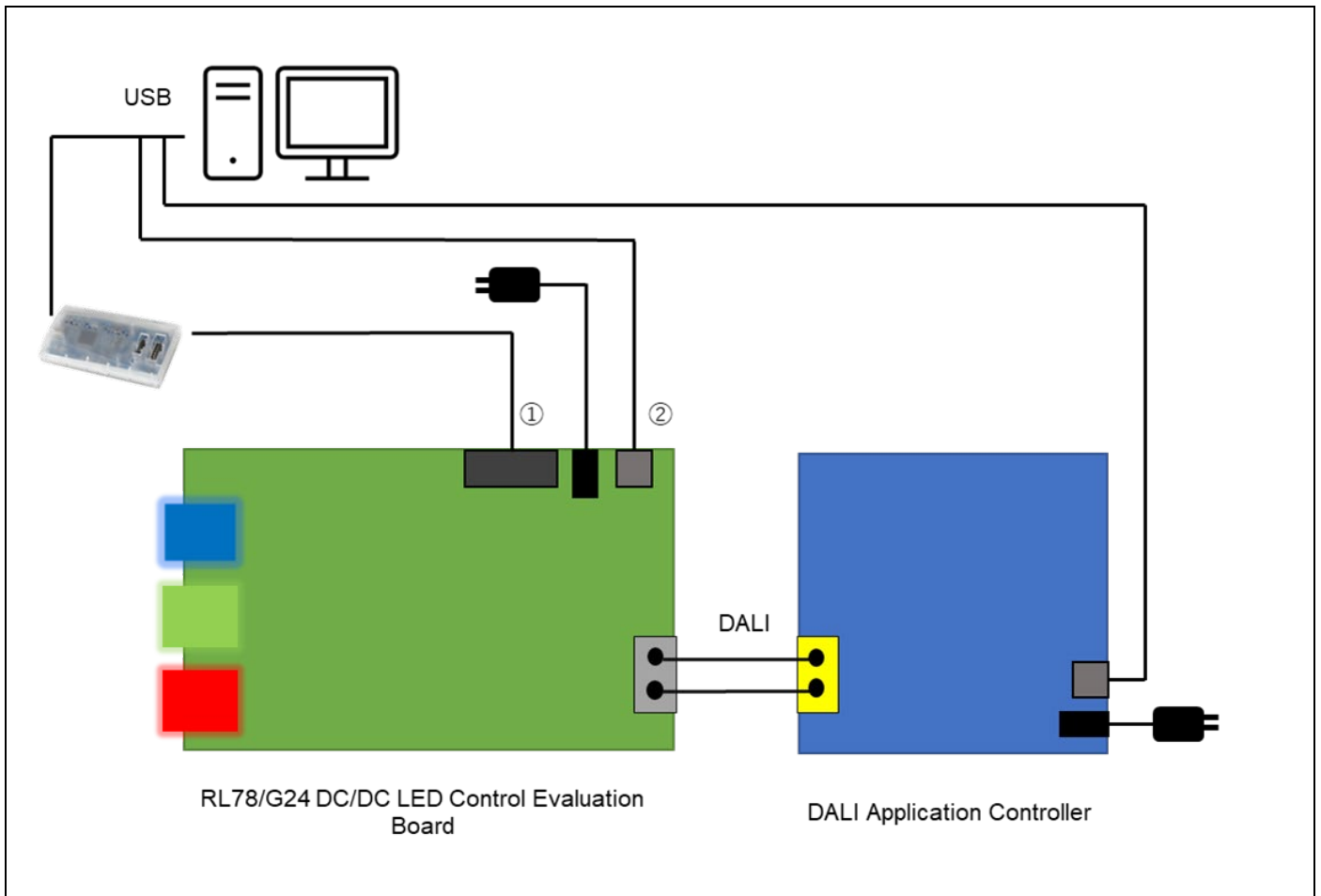


Figure 4-2 Connection diagram


Connect ① when debugging with an emulator.

Connect ② when using serial communication monitor.

4.1.9 Debugging the Sample Application

1. Right-click [RL78 E2 Lite] in the project tree, open [Property], and confirm that the settings are as shown below.

* If the data stored in the data flash is to be deleted, set [Erase flash ROM when starting] to [Yes].

2. Click the  button to download the program to the microcontroller.


3. Click on the  button to run the program.



Figure 4-3 Debug Screen Settings

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

- CS+ V8.10.00 Integrated Development Environment User's Manual: Project Operation (R20UT5299)

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 connecting the RL78/G24 DC/DC LED Control Evaluation Board and the Application Controller that operates the RL78/G24 DC/DC LED Control Evaluation Board, supply power to each device. Refer to Chapter 3.1 System configuration for the connection method.
2. Use serial communication software (Tera Term, etc.), select the USB COM port connected to RL78/G24 DC/DC LED Control Evaluation Board, and then connect. The connection settings are as follows.

Table 4-2 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 performing 4.2(1) Common operating procedures, open Manual Command in the DALI master controller GUI and select the Control Gear tab.
 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.
- Open the Manual Command of the DALI master controller GUI and select the Control Gear tab.
- (2) Set Fast fade time to 400ms
 - Transmit RESET
 - Transmit DTR0(16)
 -
 - Check [with ENABLE DEVICE TYPE command] and 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 SHORT CIRCUIT" on the DALI master controller GUI.
e.g. On the DALI master controller GUI, check [with ENABLE DEVICE TYPE command] and transmit [QUERY SHORT CIRCUIT].
 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) For DALI102ed.2.0+207ed.1.0+209ed.1.0(all) sample applications
- (a) Colour type: Colour temperature changes with Tc
 6. After performing 4.2(1) Common operating procedures, open the Manual command from Send Code... in the DALI master controller GUI and select the Control Gear tab.
 7. Send a command to change the colour temperature (Tc).
e.g.: Change colour temperature (Tc) to a warmer colour by 100[mirek] minutes.
 - Send DTR0 (100)
 - Check "with ENABLE DEVICE TYPE command" and send STORE COLOUR TEMPERATURE Tc STEP INCREMENT by broadcast Twice
 - COLOUR TEMPERATURE Tc STEP WARMER transmitted by Broadcast
 8. Verify that the lamps have variable lighting conditions.
 9. Confirm that the serial output outputs variable values in accordance with the above.

(b) Colour type: RGBWAF to change colour

6. After performing 4.2(1) Common operating procedures, open the Manual command from Send Code... in the DALI master controller GUI and select the Control Gear tab.
7. Send command to change RGBWAF.
e.g.: Change colour in RGBWAF to R=40%, G=100%, and B=60% in normalised colour control.
 - Send DTR(128)
 - Check "with ENABLE DEVICE TYPE command" and send SET TEMPORARY RGBWAF CONTROL by Broadcast
 - Send DTR0(7)
 - Check "with ENABLE DEVICE TYPE command" and Twice send STORE ENABLED CHANNELS by Broadcast
 - Send DTR0(100)
 - Send DTR1 (200)
 - Send DTR2 (150)
 - Check "with ENABLE DEVICE TYPE command" and send SET TEMPORARY RGB DIMLEVEL by Broadcast.
 - Check "with ENABLE DEVICE TYPE command" and send ACTIVATE by Broadcast.
8. Confirm that the lighting status of the lamps is variable.
9. Confirm that the serial output outputs variable values in accordance with the above.

(c) Colour type: Change colour with XY

6. After performing 4.2(1) Common operating procedures, open the Manual command from Send Code... in the DALI master controller GUI and select the Control Gear tab.
7. Sends a command to change the coordinate point XY.
e.g.: Change colour to $(x, y) = (0.5, 0.4)$ in XY space.
 - Send DTR0(0)
 - Send DTR1(128)
 - Check "with ENABLE DEVICE TYPE command" and send SET TEMPORARY x-COORDINATE by Broadcast
 - Send DTR0(102)
 - Send DTR1(102)
 - Check "with ENABLE DEVICE TYPE command" and send SET TEMPORARY y-COORDINATE by Broadcast
 - Check "with ENABLE DEVICE TYPE command" and send ACTIVATE by Broadcast
8. Confirm that the lighting status of the lamps 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.

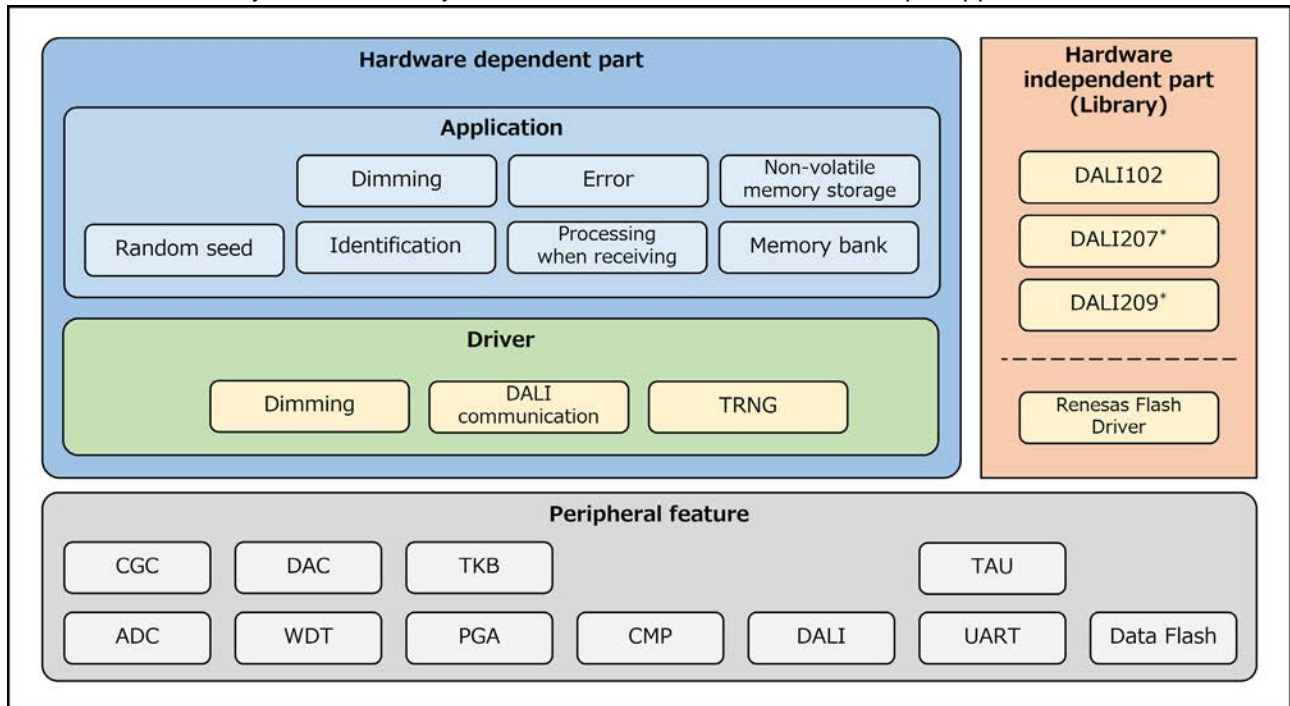


Figure 4-4 Feature block diagram

4.3.2 Hardware independent part

Hardware-independent parts are supported by the DALI library. For details, refer to the user's manual of each library.

- Renesas Flash Driver is used to store non-volatile memory; for details on Renesas Flash Driver, refer to the RL78 Family Renesas Flash Driver RL78 Type01 User's Manual for more information on the Renesas Flash Driver.

4.3.3 Hardware dependent part

The hardware dependent part is implemented to match the hardware configuration of the RL78/G24 DC/DC LED Control Evaluation 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.

The DALI module is used for transmitting and receiving DALI, and the TAU feature is used for detecting System Failure and settling time.

(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 Renesas Flash Driver Type01. 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.

* A block here refers to a virtual block (1024 bytes) that is a combination of four physical blocks (256 bytes) of data flash.

The block configuration is shown in Figure 4-5.

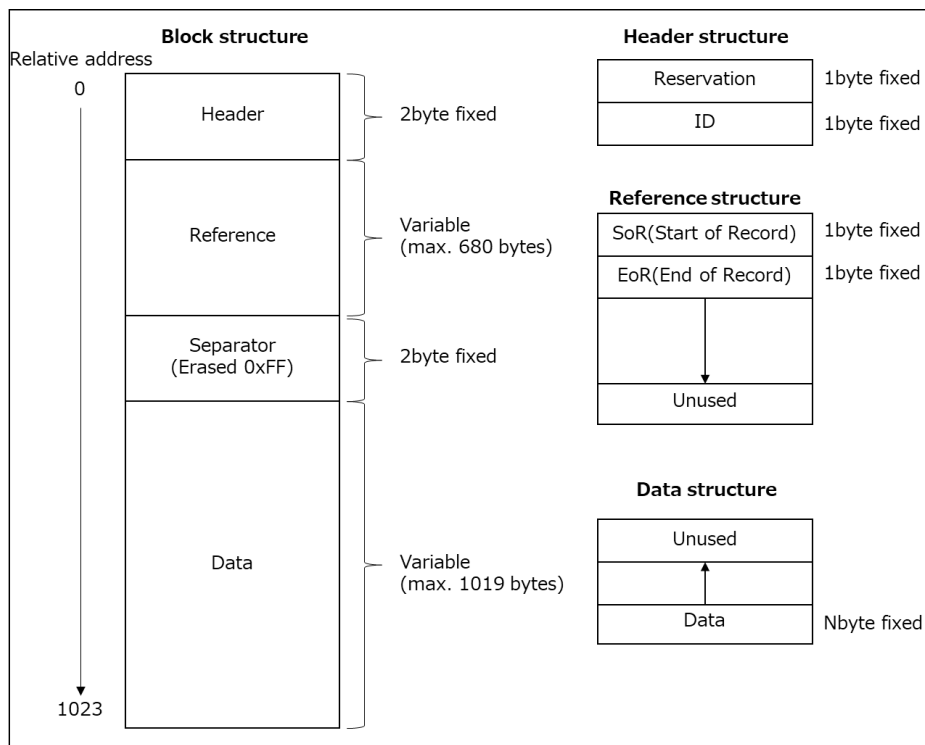


Figure 4-5 Block structure diagram

(a) Header structure

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

Table 4-3 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 4-4 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 4-5 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

Seed values of random numbers used in the DALI library are generated by a true random number generator (TRNG).

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 4-6 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+207ed.1.0+209ed.1.0 Sample Application Specific Features

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

(1) LED colour adjustment feature(Tc)

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

As an example, when the colour 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 colour 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 colour temperature (Tc) and each temperature range LED, and the relationship between total dimming ratio (actual level) and colour temperature (Tc).

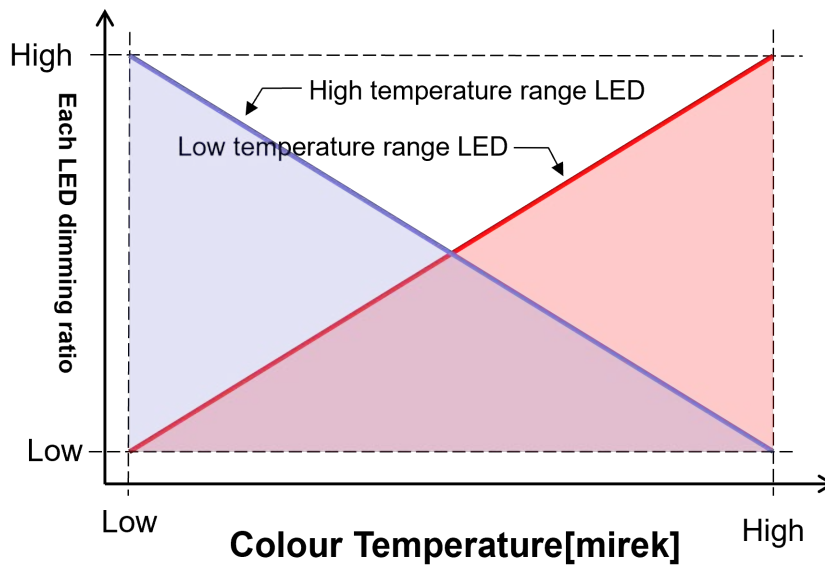


Figure 4-6 Relationship between colour temperature and each temperature range LED

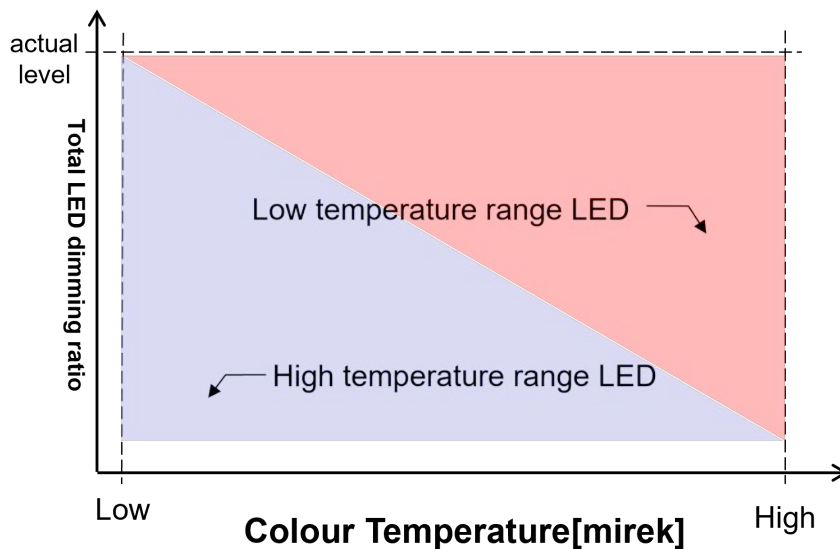


Figure 4-7 Relationship between total dimming ratio (actual level) and colour temperature (TC)

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.

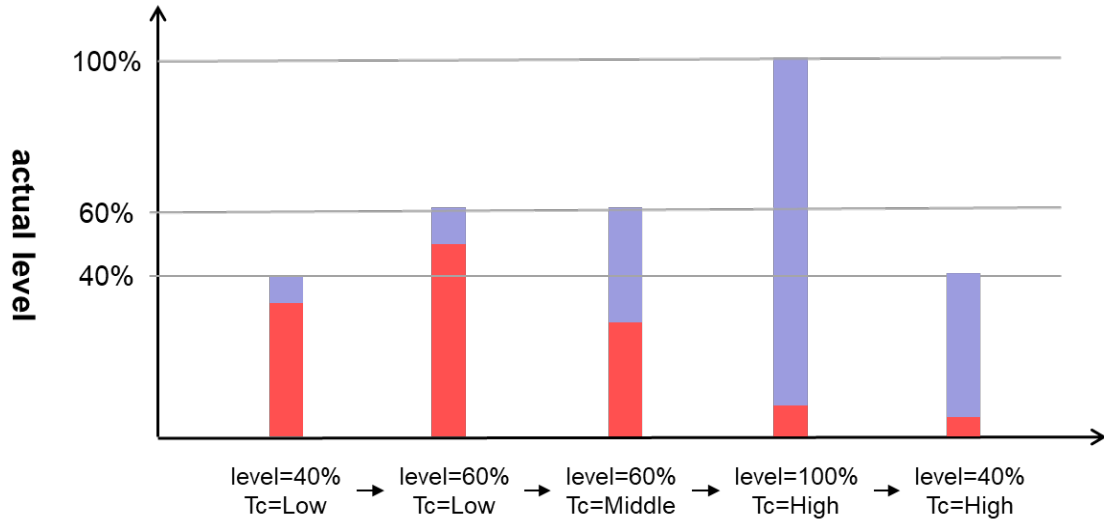


Figure 4-8 Ratio of the total dimming ratio (actual level) to the dimming ratio of each temperature range LED

The PHM is specified as the minimum physical dimming ratio for LEDs. When using the dimming/colour 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 colour 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: 0x0096
- physical warmest: 0x01C2

Colour temperature Tc	0x0064	~	0x0096 (MAX∅25%)	~	0x012C	~	0x01C2 (MAX∅75%)	~	0x01F4
actual level(PHM)	170(10.091%)								
High temperature LED dimming rate	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 rate	0% (0%*10.091%)	~	2.522% (25%*10.091%)	~	5.0455% (50%*10.091%)	~	7.568% (75%*10.091%)	~	10.091% (100%*10.091%)
Total LED Dimming Rate (Low temperature LED + high temperature LED)	10.091%	~	10.091%	~	10.091%	~	10.091%	~	10.091%

Figure 4-9 Each dimming ratio operation table that meets the physical limitations in this sample

(2) LED Colour Adjustment Feature (XY)

This sample application uses three LEDs (red, green, and blue) for each logical unit. The specified XY coordinate points are converted to RGB and the ratio of each LED's output is calculated. By multiplying this by the current dimming ratio, the sum of the dimming ratios of each LED is always set to match the dimming ratio specified in actual level. The dimming rate of each LED also varies with the actual level; changing the actual level changes the dimming rate of each LED, but does not change the ratio of the dimming rates of the LEDs.

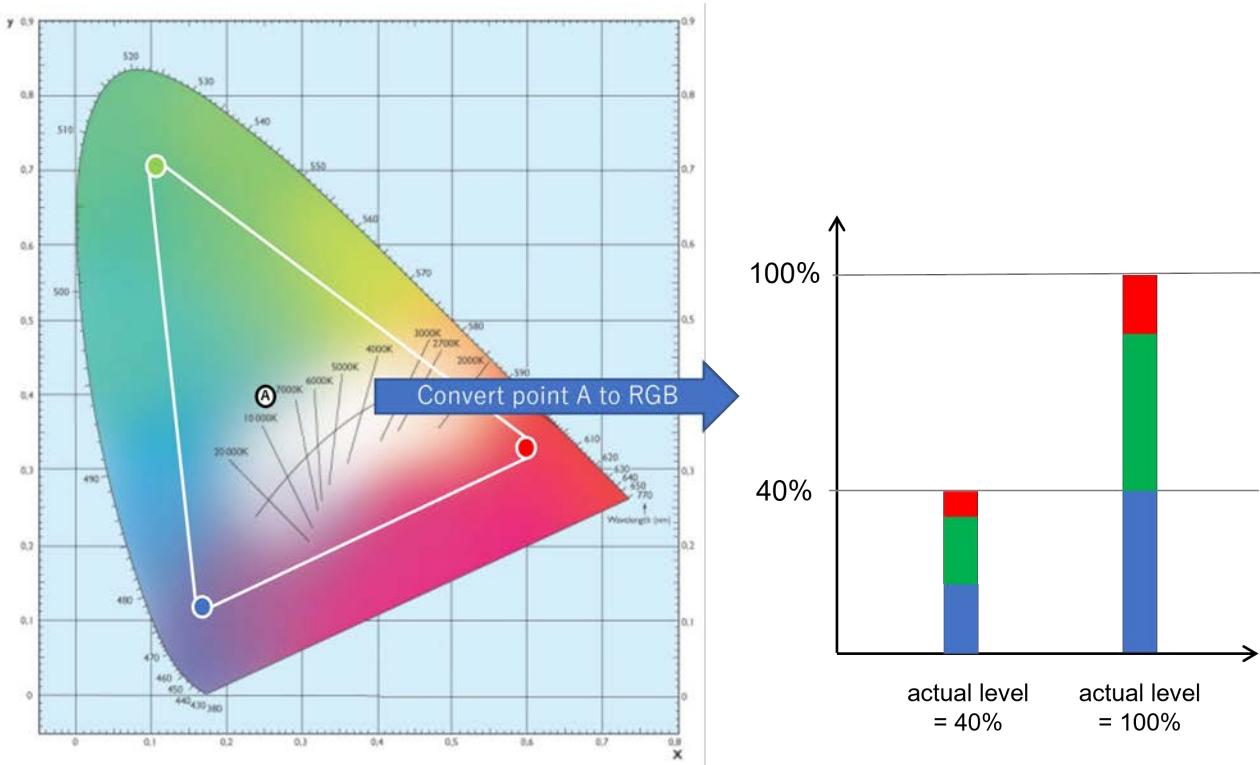


Figure 4-10 Ratio of RGB conversion of XY coordinate points to dimming ratio

In the XY colourimetric feature, it is necessary to determine whether the colour indicated by the XY coordinate points can be expressed by the light source used and to notify the library of this determination. For example, there may be a range of colours that cannot be mixed due to the position of the light source used, or the light source may have an output limit and the amount of light is too much (or not enough) to express the colour. When the XY coordinate points that cannot be used for colour representation are indicated, it is necessary to "adjust the colour along the lines where colour representation is possible" or "maintain the current colour adjustment state when the range where colour representation is not possible is reached, and resume colour adjustment when the range where colour representation is possible is reached again.

In this sample application, the "colour representation is possible" is judged based on the colour inside the triangle surrounded by the coordinate points of the three primary colours. If the output of each LED falls below the PHM (physical minimum level), the output is set to PHM.

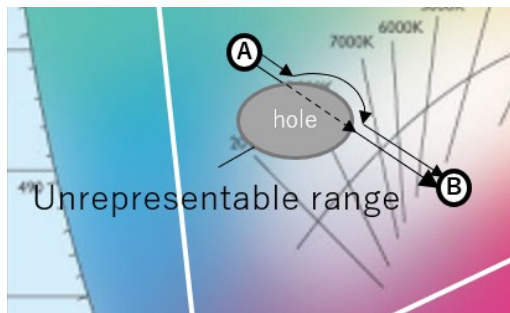
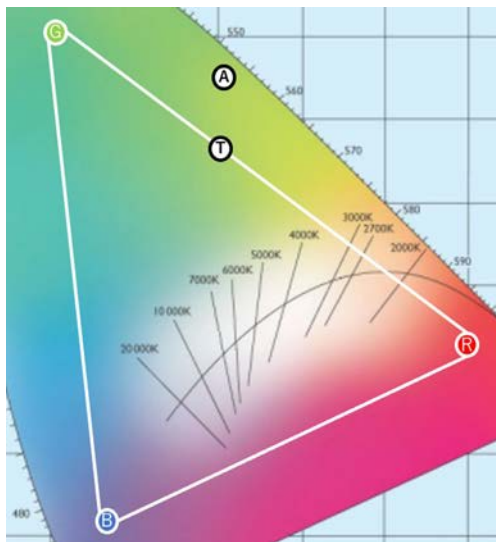


Figure 4-11 Example of operation and range where colour representation is not possible

In addition, the out-of-range judgment of coordinate points is made so that the XY coordinate points to be acquired are always "within the representable range". Accordingly, it is necessary to set the coordinate points (primary_xy_coordinate_red / primary_xy_coordinate_green / primary_xy_coordinate_blue) for each LED being used.

In this sample application, the coordinate points of each LED are set as follows to enable out-of-range judgment.

- primary_xy_coordinate_red[DALI209_INDEX_XY_X] : 0xB43B (0.70)
- primary_xy_coordinate_red[DALI209_INDEX_XY_Y] : 0x4BC0 (0.29)
- primary_xy_coordinate_green[DALI209_INDEX_XY_X] : 0x29E0 (0.16)
- primary_xy_coordinate_green[DALI209_INDEX_XY_Y] : 0xB823 (0.71)
- primary_xy_coordinate_blue[DALI209_INDEX_XY_X] : 0x212E (0.12)
- primary_xy_coordinate_blue[DALI209_INDEX_XY_Y] : 0x11CB (0.06)



A: Coordinate point outside the range
T: Coordinate point inside the range

R: Coordinate point represented by the red LED
G: Coordinate point represented by the green LED
B: Coordinate point represented by blue LED

Figure 4-12 Example of range and out-of-range/in-range coordinate points for each LED coordinate point

For details on each setting value and the function to determine if colour representation is possible, refer to the RL78 Family Control Gear Library User's Manual: Colour Control (209).

(3) LED colour control feature (RGBWAF)

In this sample application, three LEDs (red, green, and blue) are used for each logical unit. However, the RL78/G24 DC/DC LED Control Evaluation Board does not implement lamps other than red, blue, and green, so these channels are disabled. In this sample application, the available LED channels are set as follows.

- `rgbwaf_channels_present` : (`DALI209_ENABLE_CH_RGBWAF_R` | `DALI209_ENABLE_CH_RGBWAF_G` | `DALI209_ENABLE_CH_RGBWAF_B`)

The channels to be used can be disabled or enabled during operation by setting enable channels. An example of setting change is shown in the table below.

Table 4.7 Breakdown of enable Channels and example settings

bit	7	6	5	4	3	2	1	0	enable channels
Ch Breakdown	-	-	Free	Amber	White	Blue	Green	Red	
Change example 1	-	-	Invalid	Invalid	Invalid	Valid	Invalid	Valid	0x05
	0	0	0	0	0	1	0	1	
Change example 2	-	-	Invalid	Valid	Invalid	Valid	Valid	Valid	0x07 *
	0	0	0	1	0	1	1	1	
* Enabling channels that have not been enabled in <code>*rgbwaf_channels_present</code> is not reflected in the enable channels setting.									

The RGBWAF colour control feature has two control methods: normalised colour control and extended colour control. Normalised colour control calculates the dimming ratio (LightOutput) from the enabled channels, colour value, and dimming ratio. Extended colour control calculates the LightOutput from the enabled channels, colour value, dimming rate, and power ratio of the lamp. Normalised colour control and extended colour control can be switched during operation. Therefore, the control method and power ratio must be set in advance. For details on the setting value of power ratio, refer to the RL78 Family Control Gear Library User's Manual: Colour Control (209).

The obtained dimming rate (LightOutput) is converted to the dimming rate of each LED so that the sum of the dimming rates of each LED always matches the dimming rate specified in the actual level. For this reason, it is necessary to inform the library of the dimming rate of the actual level specified to calculate the LightOutput.

However, LightOutput does not take PHM (physical minimum level) into account. Also, since there is no output range limit like Tc or XY, care must be taken to maintain the output state when it falls below PHM.

In this sample application, when the output of each LED falls below PHM, the output is set to PHM.

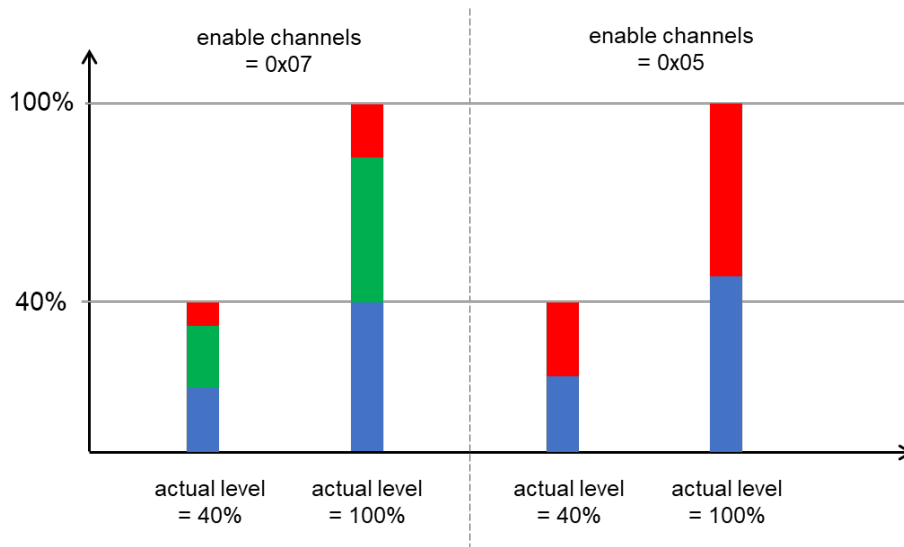


Figure 4-13 Relationship diagram between enable channels and LightOutput

(4) LED colour modulation feature (colour type change)

The colour type change feature is intended to change the currently enabled colour type during operation. This feature is valid only when the colour type to be used is set to a combination of two or more of TC, XY, or RGBWAF.

For example, if the colour type is set to support XY and Tc, and the currently valid colour type is XY, specifying the colour type Tc while operating with the colour type XY will switch the valid colour type to Tc.

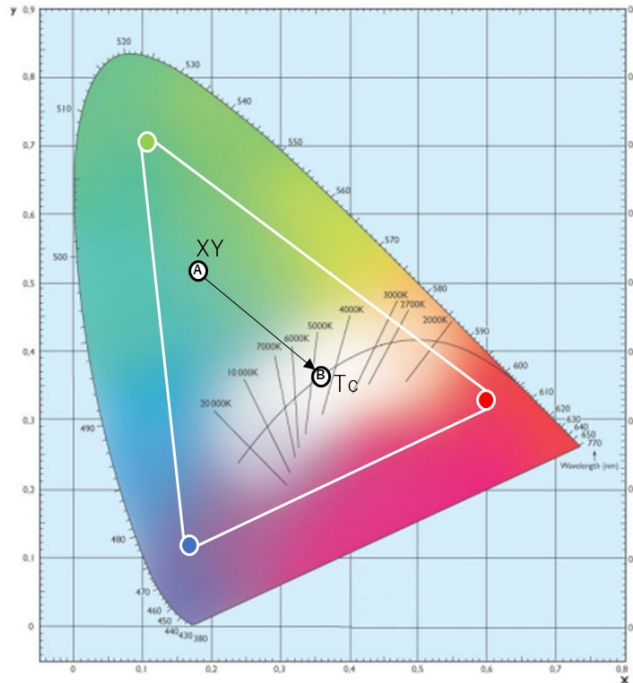


Figure 4-14 Example of colour type change operation

If a colour fade is required when changing the colour type, the conversion process corresponding to each colour type must be registered in the library. If fading is not required, the conversion process does not need to be registered. For details on conversion processing, refer to the RL78 Family Control Gear Library User's Manual: Colour Control (209).

(5) Debugging feature (dimming/colour control monitoring feature)

Serial output is used to display changes in colour (colour value) as numerical values.

At 100ms intervals, it checks the actual level and the colour value of the currently enabled colour mixing feature, and outputs the following only when there is a change in either value.

- Example of output screen for colour control by colour temperature (Tc)
Example of screen operation: When colour temperature is changed from 150[mirek] to 200[mirek] to 250[mirek] with actual level=100%.

```
(Level, Tc) = (0xFE, 0x0096)
(Level, Tc) = (0xFE, 0x00C8)
(Level, Tc) = (0xFE, 0x00FA)
```

```
.
.
.
```

Figure 4-15 Serial output screen (colour temperature Tc)

- Example of output screen during colour control by XY
Example of screen operation: When the coordinate point in XY space is changed from (x , y) = (0.58, 0.32) → (0.2, 0.63) → (0.15, 0.13) with actual level = 100%.

```
(Level, coordinate_x, coordinate_y) = (0xFE, 0x963E, 0x52DD)
(Level, coordinate_x, coordinate_y) = (0xFE, 0x3362, 0xA21E)
(Level, coordinate_x, coordinate_y) = (0xFE, 0x2742, 0x225B)
```

```
.
.
.
```

Figure 4-16 Serial output screen (XY)

- Example of output screen when colour control by RGBWAF
Example of screen operation: When the following changes are made in the Normalised colour control setting

1. enabled channels= RED, GREEN, BLUE output enabled
RED=33.33%, GREEN=33.33%, BLUE=33.33% output
2. enabled channels= RED, BLUE output enabled
RED=50%, BLUE=50% output
3. enabled channels= RED, GREEN, BLUE output enabled
RED=22.22%, GREEN=33.33%, BLUE=44.44% output

```
lightOutput(R, G, B, W, A, F) = (0x0D05, 0x0D05, 0x0D05, -, -, -)
lightOutput(R, G, B, W, A, F) = (0x1388, -, 0x1388, -, -, -)
lightOutput(R, G, B, W, A, F) = (0x08AE, 0x0D05, 0x115C, -, -, -)
```

```
.
.
.
```

Figure 4-17 Serial output screen (RGBWAF)

(6) 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 z , Z	Short circuit detection q: Set the output of the lamp (red) to short-circuit status Q: Clear the short-circuit condition of the output of lamp (red) a: Set the output of lamp (green) to short-circuited state A: Clear the short-circuited state of the output of lamp (green) z: Set the output of lamp (blue) to short-circuited state Z: Clear the short-circuited state of the output of lamp (blue)
	OPEN CIRCUIT	w , W s , S x , X	Open circuit detection w: Set the output of lamp (red) to open-circuit state W: Clear the open-circuit state of the output of the lamp (red) s: Set the output of the lamp (green) to open-circuit state S: Clear the open circuit state of the output of the lamp (green) x: Set the output of the lamp (blue) to open-circuit state X: Clear the open circuit state of the output of the lamp (blue)
	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 Calibration

If the duty cycle of the DALI receive waveform or DALI transmit waveform changes due to the characteristics or degradation of an external device (photocoupler), the corresponding waveform must be corrected using the RXWR1 and TXWR1 registers.

4.4.1 DALI Receive Waveform Correction

By setting the RXWR1 register, it is possible to extend the Low or High level of the receive waveform, and as a result, the duty ratio of the receive waveform can be corrected.

The procedure for correcting the DALI receive waveform in this sample is shown below.

1. After connecting the RL78/G24 DC/DC LED Control Evaluation Board and the Application Controller that operates the RL78/G24 DC/DC LED Control Evaluation Board, supply power to each device.
2. Enable the "RXD_CALIBRATION" macro in the sample project r_main.c and download it after building and debugging.
3. Set the oscilloscope channels as shown in the table below.

Table 4-13 Oscilloscope setup example

Channel	Destination	Setting
CH1	DALI Communication Bus	Voltage range: 5V/div
CH2	RL78/G24 DC/DC LED Control Evaluation Board J1-2 (Synchronous output of RXDMON register)	Voltage range: 2V/div
Other settings <ul style="list-style-type: none"> • A time range of about 100 us/div is recommended. • Set the trigger to trigger at around 2.5V falling edge for channel 2. 		

4. Broadcast transmit of the QUERY GEAR PRESENT command from the DALI Master Controller GUI.
5. Measure one of the following points from the waveform of CH1 acquired by oscilloscope trigger.
 - ① Low width of the half bit from the 8V point at the falling edge to the 8V point at the rising edge
6. Measure one of the following points from the waveform of CH2 acquired by oscilloscope trigger.
 - ② Low width of the half bit from the 8V point at the falling edge to the 8V point at the rising edge
7. Find the error of the measurement result ② with respect to the measurement result ①, and set a value in the RXWR1 register that corrects the error to be as close to 0 as possible.
8. Build/write with the "RXD_CALIBRATION" macro disabled in the sample project "r_main.c".

If the receive waveform cannot be fully compensated by the above RXWR1 register setting, use the DALI bit timing violation threshold register to adjust the time slot.

For details on how to set the RXWR1 register, refer to the RL78/G24 User's Manual, Hardware Edition (R01UH0961).

4.4.2 DALI Transmit Waveform Correction

1. After connecting the RL78/G24 DC/DC LED Control Evaluation Board and the Application Controller that operates the RL78/G24 DC/DC LED Control Evaluation Board, supply power to each device.
2. After building the sample project, run debugging.
3. Set the oscilloscope channels as shown in the table below.

Table 4-13 Oscilloscope setup example

Channel	Destination	Setting
CH1	DALI Communication Bus	Voltage range: 5V/div
CH2	TP5 (DALI_TX) on target board	Voltage range: 2V/div
Other settings		
<ul style="list-style-type: none"> • A time range of about 100 us/div is recommended. • Set the trigger to trigger at around 2.5V falling edge for channel 2. 		

4. Broadcast transmit of the QUERY GEAR PRESENT command from the DALI Master Controller GUI.
5. Measure the following one point from the waveform of CH1 acquired by oscilloscope trigger.
 - ① Low width of the half bit from the 8V point at the falling edge to the 8V point at the rising edge
6. Set the TXWR1 register to the value to be corrected so that the measurement result of ① comes as close as possible to the half bit Typical value.

For details on how to set the TXWR1 register, refer to the RL78/G24 User's Manual, Hardware Edition (R01UH0961).

4.5 Software Configuration

The software configuration for each sample application is described below.

4.5.1 Folder Structure

This sample application provides two types of LED dimming control, one on the CPU and the other on the FAA (Flexible Application Accelerator). The folder structure for this sample application is shown below.

4.5.1.1 Sample application for LED dimming control by CPU

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 colour temperature (TC) compatible dimming module ^{*2}
r_lamp_tc.h ^{*2}	Header file for 209ed.1.0 colour temperature (TC) compatible dimming module ^{*2}
r_lamp_xy.c ^{*2}	Source file for 209ed.1.0 XY-enabled dimming module ^{*2}
r_lamp_xy.h ^{*2}	Header file for 209ed.1.0 XY-enabled dimming module ^{*2}
r_lamp_rgbwaf.c ^{*2}	Source file for 209ed.1.0 RGBWAF-enabled dimming module ^{*2}
r_lamp_rgbwaf.h ^{*2}	Header file for 209ed.1.0 RGBWAF-enabled 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
Config_DALI.c	Source file for DALI peripheral function settings
Config_DALI.h	Header file for DALI peripheral function settings
Config_DALI_user.c	Source file for DALI peripheral function setting users
r_cg_dali.h	Header file for DALI peripheral function definition
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)
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

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_port.c	Source file for PORT
r_port.h	Header file for PORT
r_trng.c	Source files for TRNG
r_trng.h	Header files for TRNG
¥DefaultBuild<DIR> (CC-RL)	Folder to store project output files (CC-RL development environment only)
DALIXXX_sample.mot	Motorola S-type format file used for writing to firmware
¥Debug¥Exe<DIR> (IAR)	Folder to store project output files (IAR development environment only)
DALIXXX_sample.mot	Motorola S-type format file used for writing to firmware
¥Library<DIR>	Folder to store library
¥DALI102<DIR>	Folder to store DALI102ed.2.0 library
r_dali_102_cc_gen2_v1_01.lib (CC-RL)	Library file to support DALI102ed.2.0
r_dali_102_iar_gen2_v1_01.a (IAR)	
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_01.lib (CC-RL)	Library file to support DALI207ed.1.0 ^{*1}
r_dali_207_iar_gen2_v1_01.a (IAR)	
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_v2_00.lib (CC-RL)	Library file to support DALI209ed.1.0 ^{*2}
r_dali_209_iar_gen2_v2_00.a (IAR)	
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}
¥RFD<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.
¥src	Smart Configurator Generation File Storage Folder
smc_gen	Smart configurator generation folder

Table 4-11 Folder structure (3/3)

Folder, File name	Description
DALI102_20X_sample	Sample application folder
¥Utility<DIR>	Folder for common programs
r_timer16.c	Source file for timer module
r_timer16.h	Header file for timer module
r_user_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.

4.5.1.2 Sample application for LED dimming control by FAA

Table 4-12 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 colour temperature (TC) compatible dimming module ^{*2}
r_lamp_tc.h ^{*2}	Header file for 209ed.1.0 colour temperature (TC) compatible dimming module ^{*2}
r_lamp_xy.c ^{*2}	Source file for 209ed.1.0 XY-enabled dimming module ^{*2}
r_lamp_xy.h ^{*2}	Header file for 209ed.1.0 XY-enabled dimming module ^{*2}
r_lamp_rgbwaf.c ^{*2}	Source file for 209ed.1.0 RGBWAF-enabled dimming module ^{*2}
r_lamp_rgbwaf.h ^{*2}	Header file for 209ed.1.0 RGBWAF-enabled 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
Config_DALI.c	Source file for DALI peripheral function settings
Config_DALI.h	Header file for DALI peripheral function settings
Config_DALI_user.c	Source file for DALI peripheral function setting users
r_cg_dali.h	Header file for DALI peripheral function definition
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)
r_led.c	Source file for the dimming driver
r_led.h	Header file for the dimming driver

Table 4-13 Folder structure (2/3)

Folder, File name	Description
DALI102_20X_sample	Folder for sample applications
¥Driver<DIR>	Folder to store driver program
r_port.c	Source file for PORT
r_port.h	Header file for PORT
r_trng.c	Source files for TRNG
r_trng.h	Header files for TRNG
¥DefaultBuild<DIR> (CC-RL)	Folder to store project output files (CC-RL development environment only)
DALIXXX_sample.mot	Motorola S-type format file used for writing to firmware
¥Debug¥Exe<DIR> (IAR)	Folder to store project output files (IAR development environment only)
DALIXXX_sample.mot	Motorola S-type format file used for writing to firmware
¥Library<DIR>	Folder to store library
¥DALI102<DIR>	Folder to store DALI102ed.2.0 library
r_dali_102_cc_gen2_v1_01.lib (CC-RL)	Library file to support DALI102ed.2.0
r_dali_102_iar_gen2_v1_01.a (IAR)	
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_01.lib (CC-RL)	Library file to support DALI207ed.1.0 ^{*1}
r_dali_207_iar_gen2_v1_01.a (IAR)	
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_v2_00.lib (CC-RL)	Library file to support DALI209ed.1.0 ^{*2}
r_dali_209_iar_gen2_v2_00.a (IAR)	
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}
¥RFD<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.
¥src	Smart Configurator Generation File Storage Folder
smc_gen	Smart configurator generation folder

Table 4-14 Folder structure (3/3)

Folder, File name	Description
DALI102_20X_sample	Sample application folder
¥Utility<DIR>	Folder for common programs
r_timer16.c	Source file for timer module
r_timer16.h	Header file for timer module
r_user_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.

4.5.2 List of Option Byte Settings

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

Table 4-15 List of Option Byte Settings

Address	Set value	Description
000C0H / 010C0H	01110001B	Watchdog timer operation permitted (counting starts after reset is released)
000C1H / 010C1H	11111011B	LVD reset mode (falling edge: 2.97V, rising edge: 2.91V)
000C2H / 010C2H	11101010B	HS mode, high-speed on-chip oscillator: 8MHz
000C3H / 010C3H	10000100B	On-chip debug permitted

4.5.3 Flowchart

4.5.3.1 Main Flow

The main flow of each sample application is shown below.

After the initialization process is complete, the Control Gear process is executed.

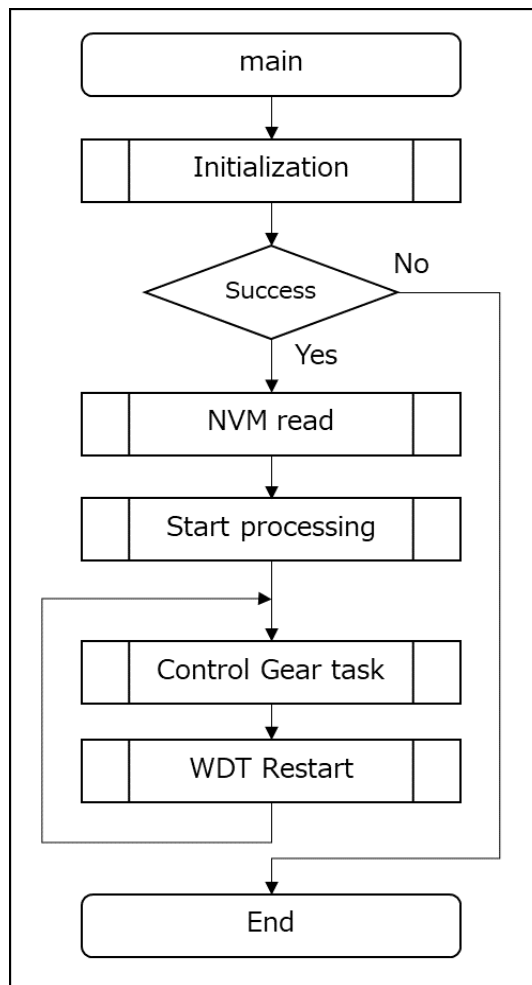


Figure 4-18 Main flow diagram

4.5.3.2 Initialization

The flow of the initialization process is shown below.

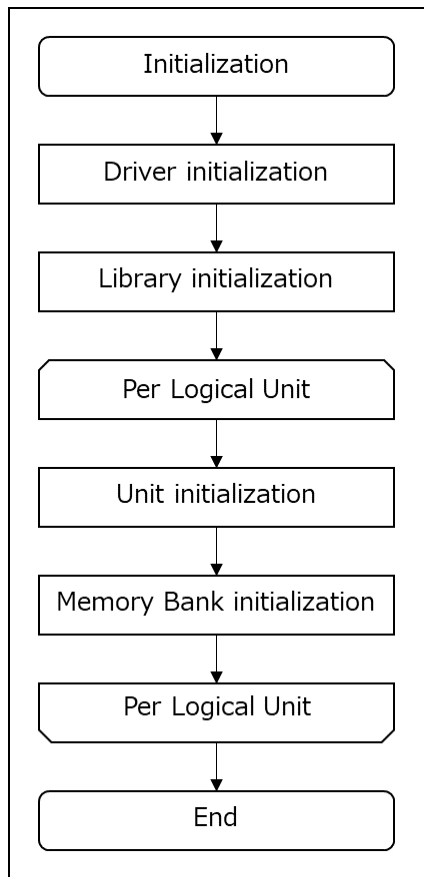


Figure 4-19 Initialization flow diagram

4.5.3.3 Reading NVM

The flow of the NVM reading process to be executed after the initialization process is shown below.

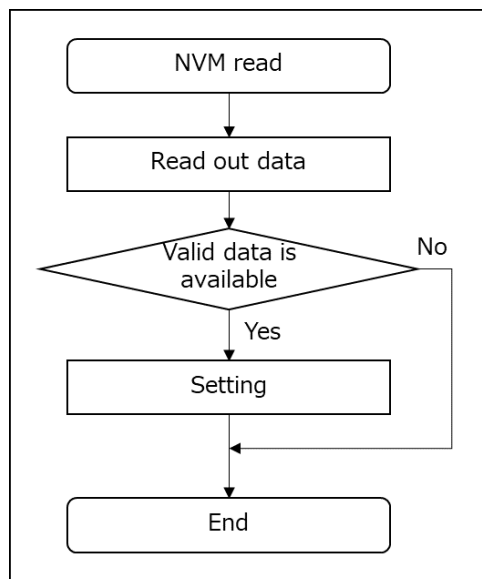


Figure 4-20 NVM reading process flow diagram

4.5.3.4 Start Processing

The flow of the control gear start process is shown below.

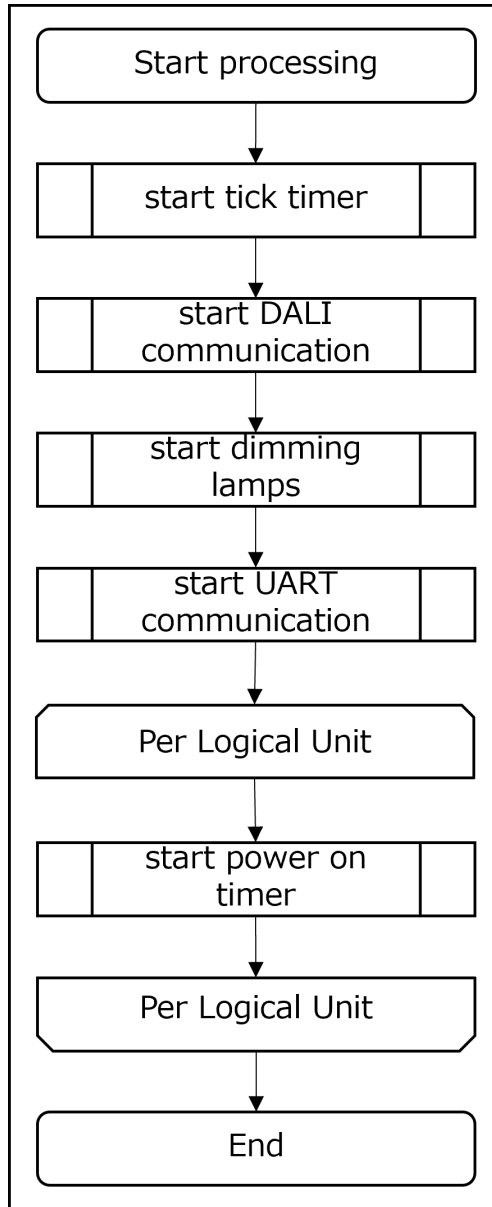


Figure 4-21 Start processing flow diagram

4.5.3.5 Control Gear Task Processing

The flow of the Control Gear task processing is shown below.

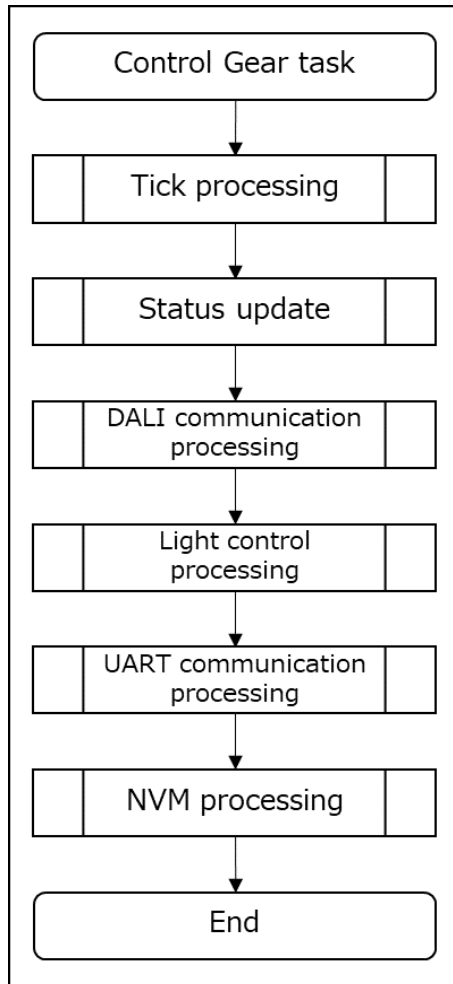


Figure 4-22 Control Gear task processing flow diagram

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)

It is implemented using a DALI module. Since the settling time (5.5 ms to 10.5 ms) between receiving the Forward Frame and transmitting the Backward Frame does not require precision, a software timer in 1 ms increments is used to adjust the waiting time before transmission.

5.1.1.1 Reproduction of Corrupted Backward Frame

Use an interval timer to force the DALI bus Low for a certain period of time (1300us to 2000us) to reproduce a corrupted Backward Frame.

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)

It is implemented using a DALI module and an interval timer.

The DALI module is used for sending and receiving DALI, and the TAU function is used for detecting System Failure and settling time.

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
P02/DALITxD0	Transmission process: DALI transmit feature
P03/DALIRxD0	Reception process: DALI receive feature
TAU0-CH2	Receive processing: Interval timer (100us)
TAU0-CH3	Transmission process: Interval timer (1458us)
DALI	DALI module

5.1.4 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_GetReceivedFrame	Receive frame
R_DALI101_SystemFailureIsDetected	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_GetReceivedFrame

Function	R_DALI101_GetReceivedFrame	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_SystemFailureIsDetected

Function	R_DALI101_SystemFailureIsDetected	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 RL78/G24 DC/DC LED Control Evaluation 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/ G24 User's Manual Hardware (R01UH0961J)

RL78 Family User's Manual Software (R01US0015J)

CS+ V8.10.00 Integrated Development Environment User's Manual: Project Operation (R20UT5299)

DALI Master Controller GUI User's Manual (R20UT0715)

RL78/G24 DC/DC LED Control Evaluation Board User's Manual (R20UT5371)

RL78 Family Renesas Flash Driver RL78 Type01 User's Manual (R20UT4830)

(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 version (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	Sep., 1, 2023	-	First edition

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

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

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

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

3. Input of signal during power-off state

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

4. Handling of unused pins

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

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

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

7. Prohibition of access to reserved addresses

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

8. Differences between products

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

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

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

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

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

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

Contact information

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

www.renesas.com/contact/