RL78/I1A

Lighting Communications Using RL78/I1A (Reception)

R01AN1115EJ0300
Rev.3.00
Mar. 31, 2016

Purpose

The purpose of this application note is to describe how to implement different communication interfaces for lighting control within lighting systems such as DALI, DMX512 and IR remote control using the RL78/I1A features.

Readers

This document is intended for lighting system engineers who design and develop lighting systems with communication capabilities.

The target products are as follows:

- 20-pin: R5F1076C
- 30-pin: R5F107AE, R5F107AC
- 38-pin: R5F107DE

Contents

1. Introduction

RL78/I1A microcontrollers support different communication protocols for lighting control of LED lightingsystems:

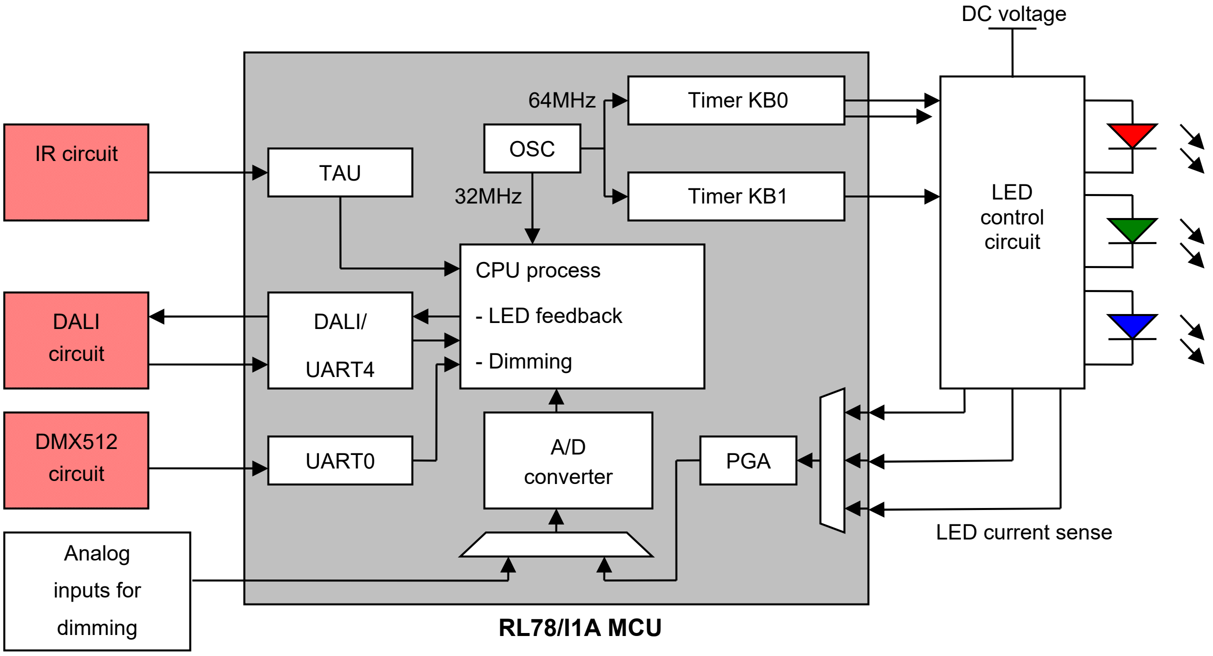
- DALI communications supported by an embedded Manchester encoder peripheral mounted onchannels 0 and 1 of Serial Array Unit 4, DALI/UART4 (transmitting and receiving frame: 8,16,17,24-bit).
- DMX512 communications supported via the UART0 serial interface and the 16-bit Timer Array Unit.
- Infrared (IR) remote control signal reception can be managed by hardware using input signal high / low-level width measurement features of the 16-bit Timer Array Unit.

The Applilet EZ for HCD Controller software automatically generates sample code to control these lighting control interfaces. This application note describes the sample code generated by Applilet EZ for HCD Controller Ver.9.0(hereinafter referred to as Applilet EZ for HCDController).

Applilet EZ for HCD Controller can generate sample code for the *RL78/I1A DC/DC LED Control Evaluation Board*. The red blocks on the left side of the board block diagram (**Figure1-1**) shows the three communication circuits introduced above, and gives an overview of their connections to the RL78/I1A peripherals:

- DALI circuit → DALI/UART4 interface
- DMX512 circuit → UART0 interface and TAU channels in low-level width measurement mode
- IR remote control circuit → TAU channels in high-level width measurement mode

Figure1-1 RL78/I1A DC/DC LED Control Evaluation Board Block Diagram



2. Applilet EZ for HCDController

2.1 Overview

Applilet EZ for HCD Controller is a tool for LED lighting /software auto-generation of a microcomputer for illuminations and program writing in.

By only specifying the dimming operation and communication mode on the GUI, you can easily generate a microcomputer of the software that controls the LED with a constant current. In addition, you write generated software in a flash memory of a microcomputer automatically via a USB cable and checkthe operations easily on an evaluation board.

Figure2-1 Structure of using Applilet EZ for HCD Controller

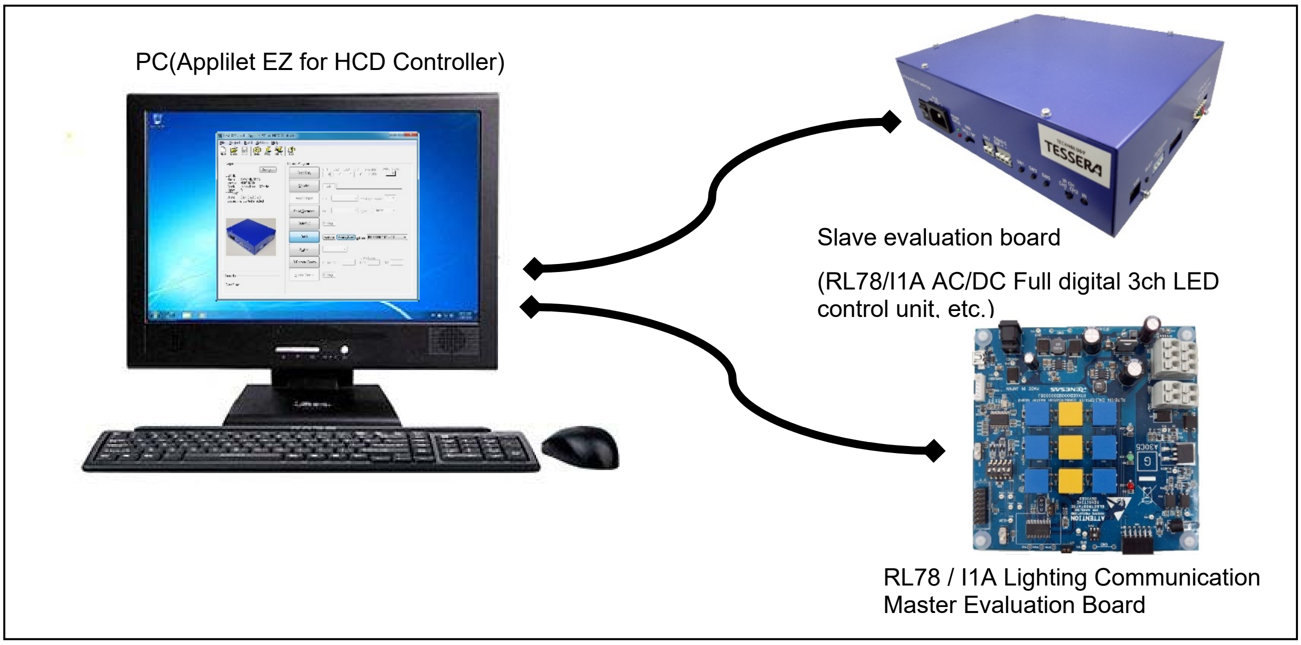
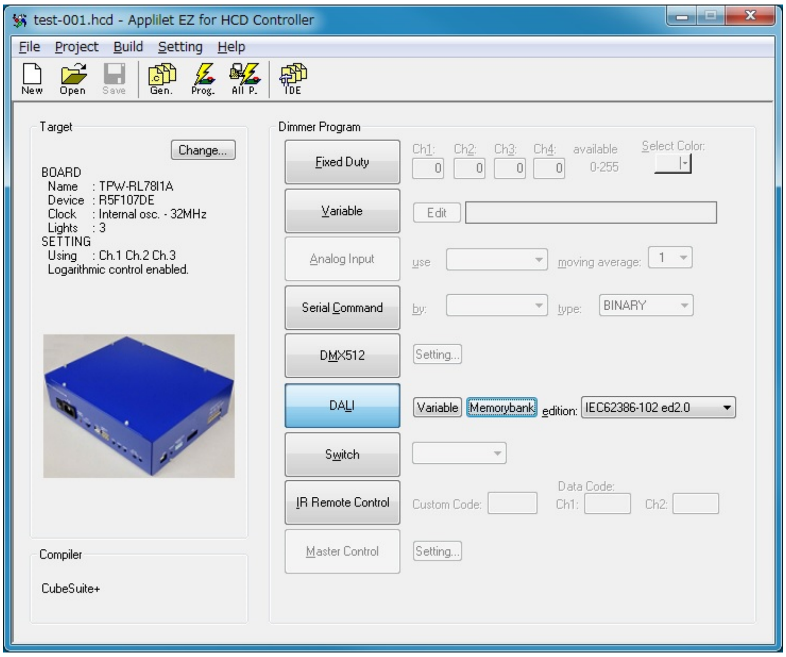

PC(Applilet EZ for HCD Controller)
Slave evaluation board
(RL78/I1A AC/DC Full digital 3ch LED control unit, etc.)
RL78 / I1A Lighting Communication Master Evaluation Board

Figure2-2 Screen of Applilet EZ for HCD Controller



BOARD
Name : TPW-RL78I1A
Device : R5F107DE
Clock : Internal osc. - 32MHz
Lights : 3
SETTING
Using : Ch.1 Ch.2 Ch.3
Logarithmic control enabled.

Ch1: 0 Ch2: 0 Ch3: 0 Ch4: 0

moving average: 1

type: BINARY

edition: IEC62386-102 ed2.0

Compiler: CubeSuite+

2.2 Supported protocols

Table2-1 shows lighting communication protocol supported in Applilet EZ for HCD Controller.

Table2-1 Supported protocols

Protocol name	Overview
DALI	DALI (Digital Addressable Lighting Interface) is an international open communication protocol for lighting control and is mainly used for controlling and toning multiple fluorescent lights and LED lamps. DALI isa standard used to achieve communication between products of different manufacturers.
DMX512	DMX512 is a wired communication protocol for digital data transmission, and it is used widely as industrial lightingsuch as stage lighting and display lighting (the device equipped with dimmer, scanner, moving lights and strobe etc.).
IR	IR is a wireless communication performed by transmitting and receiving signals using the infrared, it supports the NEC format in Applilet EZ for HCD Controller. NEC format is one of the infrared transmission protocols that are widely used in the industry around the world.Several bytes of information is sent at low speed using infrared of about 950 nm in infrared remote control of NEC.

For detail, see Applilet EZ for HCD Controller V9.0 User’s Manual (R20UT0435EJ1300).

2.3 Supported board

Table2-2 shows the list of supported board in Applilet EZ for HCD Controller.

Table2-2 Supported board

Board name	Target component	Overview
RL78/I1A DC/DC LED Control Evaluation Board	Control gear	It is an evaluation board of the LED which adopted RL78/I1A. It's equipped with an LED with 3 colors of Red, Green, Blue.It is possible to control by RL78 / I1A and FET without the constant-current driver IC.
RL78/I1A AC/DC Full digital 3ch LED control unit	Control gear	It is a LED power supply evaluation device manufactured by Tessera technology equipped with the RL78 / I1A.It is possible to control PFC and LED up to 3ch.Writing to the microcomputer and debugging is performed by using the on-board USBIF or E1.
RL78/I1A AC/DC 1 converter LED evaluation unit	Control gear	It is an evaluation board of LED control by non-insulation type 1 converter system as which RL78/I1A was adopted.Writing in to a microcomputer and debugging are performed using on-board USBIF or E1.
RL78/I1A AC/DC 2 converter LED evaluation unit	Control gear	It is an evaluation board of LED control by non-insulation type 2 converter system as which RL78/I1A was adopted.Writing in to a microcomputer and debugging are performed using on-board USBIF or E1.
Lighting Communication Master Evaluation Board	Control device (Application controller)	It is possible to use as a communication master board for controlling various lighting evaluation board.Each interface of DALI protocol communication, DMX512 protocol communication and infrared remote control are supported.In addition, it is also able to communicate only by switch operation on the master board.

Note In this application note, it is not included in the scope of the EZ-0012 previous LED evaluation board (EZ-0005, EZ-0006, etc.) and lighting communication master evaluation board (EZ-0008).

3. DALI Communication

In this application note, unless otherwise noted, it describes the IEC62386101ed1.0 and IEC62386-102ed1.0.
For the configuration of the software and the function, it is described with reference to the one for IEC62386-102ed1.0 of evaluation board (RL78/I1A DCDC LED Control Evaluation Board).
IEC62386-101 ed.2.0 and IEC62386-102 ed.2.0 released in November 2014 are changed about support of multi-master and at the timing of communication, etc. For details, see Appendix A and Appendix B.

3.1 Overview of DALI

3.1.1 What is DALI?

DALI (Digital Addressable Lighting Interface) is an international open communication protocol for lighting control and is mainly used for controlling multiple fluorescent lights and LED lamps. DALI is a standard used to achieve communication between products of different manufacturers.

3.1.2 DALI standard configuration

DALI is prescribed in IEC62386.

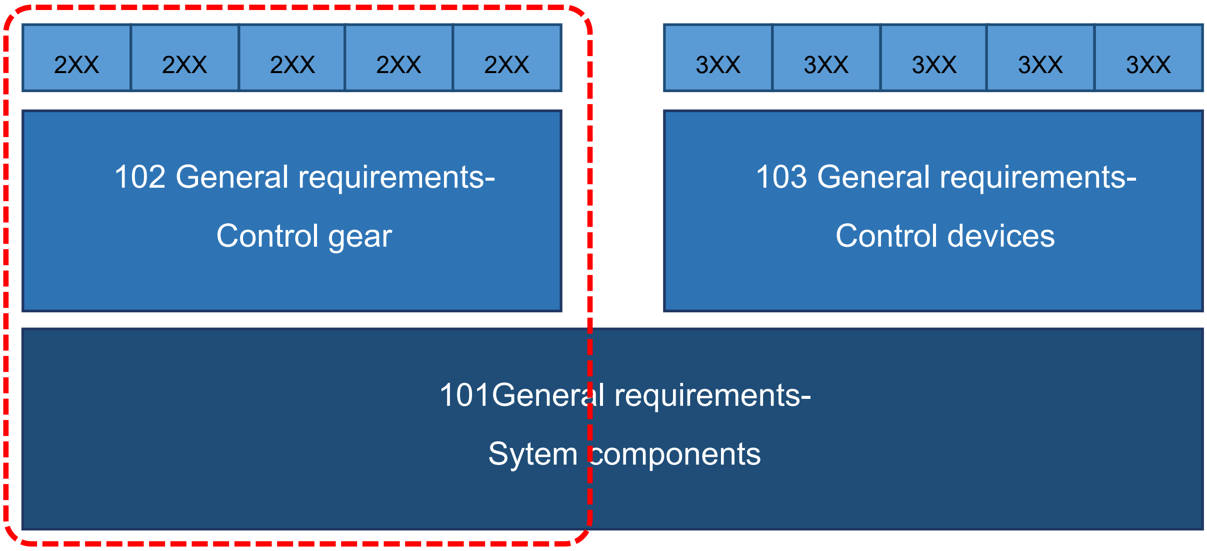
3.1.2.1 Configuration Overview

DALI standard configuration is shown below.

IEC62386 contains some of the Part called the series.

-Part 101	General terms about the system's components
-Part 102	General terms for the Control Gear (slave)
-Part 103	General terms for the Control Device (master)
-Part2xx	Extension peculiar to a source of light about Control Gear (slave)
-Part3xx	Extension peculiar to Input Device about Control Device (master)

Figure3-1 Figure of overview of IEC62386



* The red frame is within a range targeted for this application note.

3.1.2.2 Extension Overview

The extension overview for Part102 and Part103 are shown in the following table.

Table3-1 Overview of Part2xx

Part number	Description
201	Fluorescent lamps (device type 0)
202	Built-in emergency lighting (device type 1)
203	Discharge lamps (excluding fluorescent lamps) (device type 2)
204	Low-voltage halogen lamps (device type 3)
205	The power supply voltage controller for incandescent lamps (device type 4)
206	Conversion to the DC voltage of the digital signal (device type 5)
207	LED model (device type 6)
208	Switching feature (device type 7)
209	Color control (device type 8)
210	Sequencer (device type 9)

Table3-2 Overview of Part3xx

Part number	Description
301	Push button
302	Switch &Slider
303	Presence detector
304	Optical sensor
305	Color sensor
306	IP interface
307	Rotary
332	Feedback
333	Manual setting

3.1.3 DALI system configuration

DALI system configuration is shown below.

3.1.3.1 System configuration

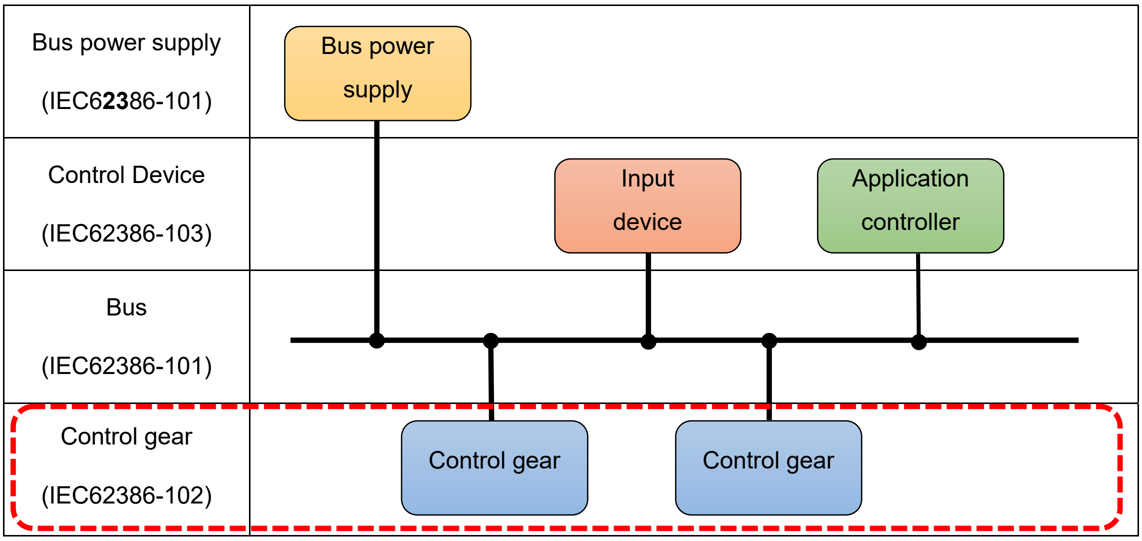
The system in accordance with DALI standard must be comprised of a component to show in Table3-3.

Table3-3 System component

Component	The number	Reference of the detailed information
Bus power supply	1 or more	IEC62386-101
Control gear	0 or more	IEC62386-102
Application controller	1 or more	IEC62386-103
Input device	0 or more	IEC62386-103
Bus	1	IEC62386-101

An example of the configuration of the system is shown in Figure3-2.

Figure3-2 Example of the configuration of the system



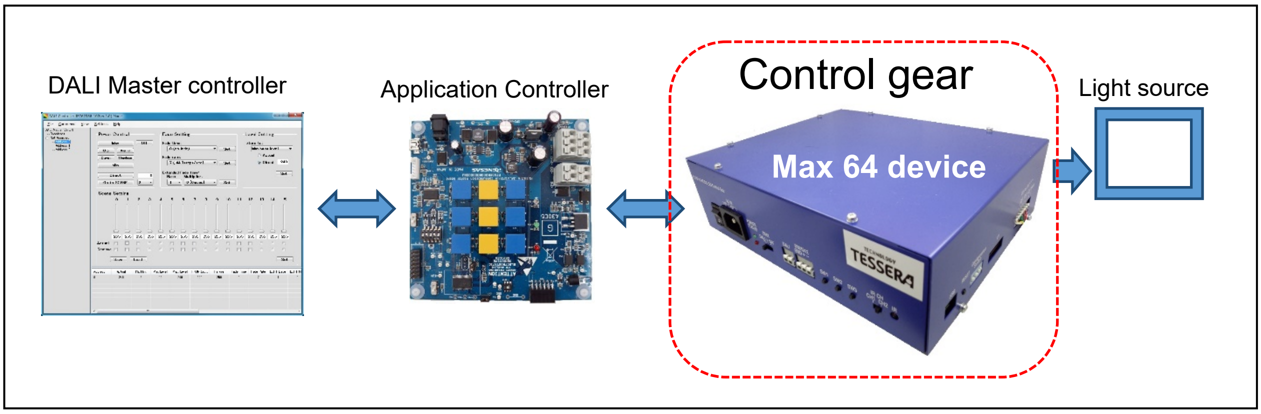
* The red frame is within a range targeted for this application note.

3.1.3.2 Control Gear

Control gear is a device that receives commands from Control device (Application controller) and for setting output (light source) or dimming in order to control at least one output (light source).

Control gear can connect at most 64 including a logic device in one Bus. Even in the case a plurality of Application controller is present (multi-master) on a Bus, the maximum number of connections Control gear is 64 units.

Figure3-3 Example of the configuration of the system



For detail about Control gear, see IEC62386-102.

Note Multi-Master is supported on IEC62386-102 ed.2.0 later.

3.1.4 Features of DALI communication

- One master can connect to up to 64 slaves
- Communication using a 2-wire, half duplex system at 1200 ± 10% (bits/sec)
- Slaves can be grouped by using a network
  - ➢ Up to 64 short addresses
  - ➢ Up to 16 group addresses
- 254-step (8-bit accuracy) lighting control levels, up to 16 of which can be saved or switched between as lighting scenes

3.1.5 Overview of DALI communication

3.1.5.1 Data structure and frame structure

(1) Data structure (bit definition)

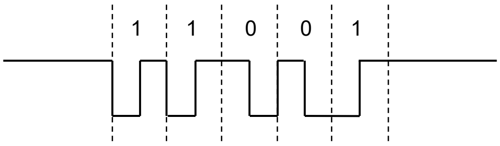
- DALI communications are Manchester encoded.

  Manchester code:

  - ➢ Bit 1 and bit 0 are not defined as the H/L voltage level but are expressed as the edge of a voltage transition.
  - ➢ Bits are defined as 0 for falling edges and as 1 for rising edges.
- The signal maintains H level when no communication is taking place.

(2) Frame structure

Figure3-4 Manchester Code Example



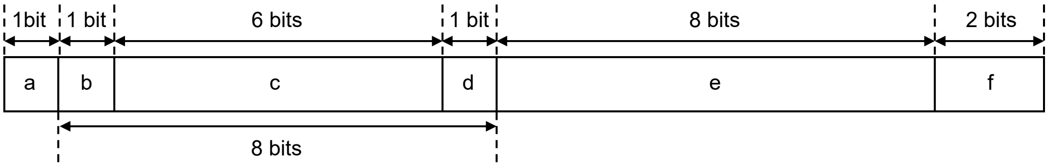
The frame structure of the DALI communication protocol is defined by Forward frame and Backward frame.

- Forward frame

  Forward frame is transmitted from the master and is composed of 19 bits.

  Figure3-5 shows the structure of Forward frame.

Figure3-5 Structure of Forward Frame



a: Start bit (1 bit, the same waveform as "1"); indicates the start of the frame

bcd: Address byte (8 bits) Note 1; specifies the frame destination or special command

b: <0> short address
<1> group address/broadcast

c: Address bit

d: Select bit
<0>Direct Arc Power Control command
<1> Other commands

e: Data byte (8 bits) Note 2; Direct Arc Power Control command data that specifies the command or the lighting

f: Stop bit (2 bits fixed at a high level); indicates the end of the frame

Notes 1: See (1) Address byte in **3.1.5.4 Commands** for details on address byte.
2: See (2) Commands in **3.1.5.4 Commands** for details on data byte.

- Backward frame
  Backward frame is transmitted from slaves and is composed of 11 bits.
  Backward frame responds to the master.

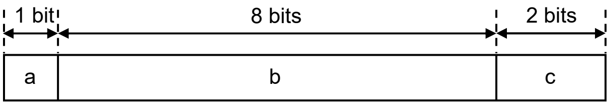
a: Start bit (1 bit, the same waveform as "1"); indicates the start of the frame

b: Data byte (8 bits); responds to the master

c: Stop bit (8 bits); indicates the end of the frame

Figure3-6 shows the structure of Backward frame.

Figure3-6 Structure of Backward Frame



3.1.5.2 Settling time

Settling time information is shown below.

(1) IEC62386-101ed1.0

Figure3-7 and Figure3-8 show about Settling time of IEC62386-101ed1.0.

Figure3-7 Forward to backward frames

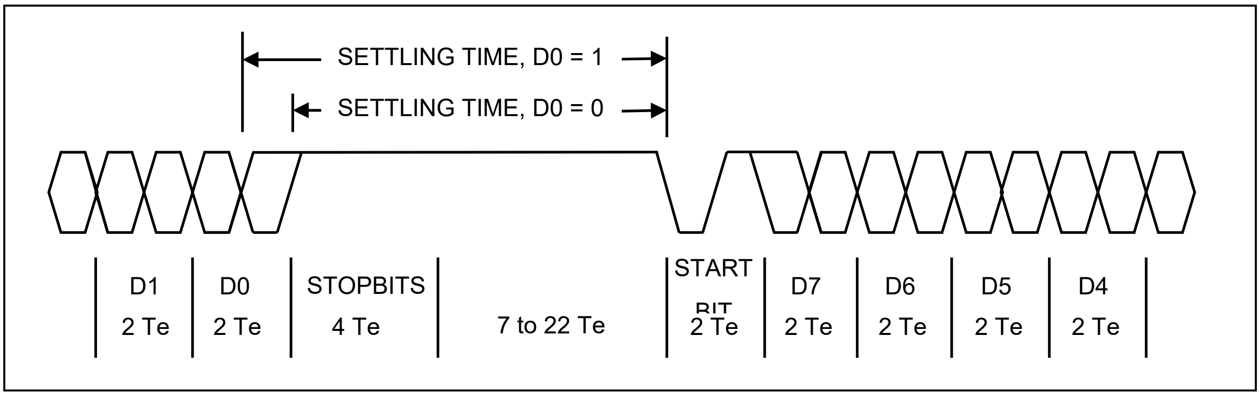
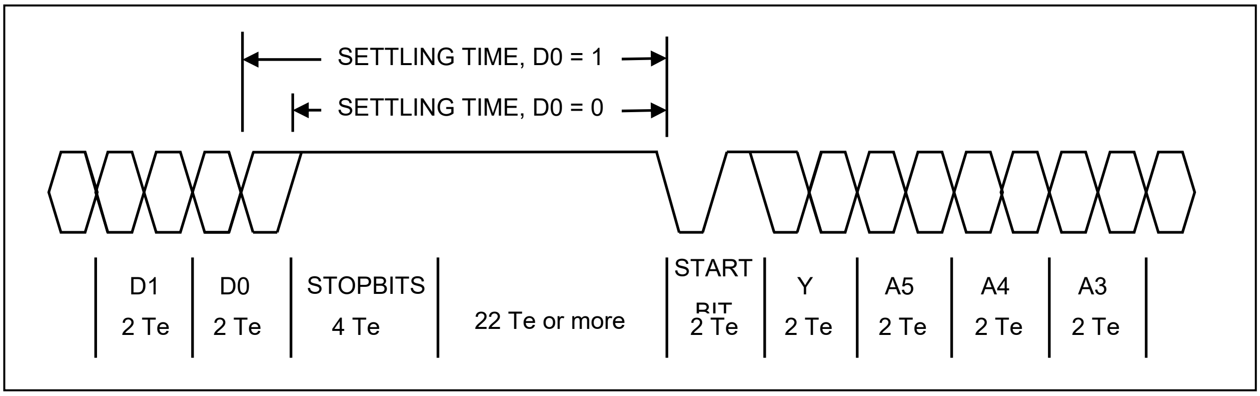


Figure3-8 Backward to forward or forward to forward frames



Settling time of IEC62386-101ed1.0 changes 1Te length on whether the final bit of data is 0 or 1.

For example, the transmission interval from Forward to Backward frame,

Settling time when last bit D0 is 0

Settling time = StopBits(4 Te) +(7Te～22 Te)

Settling time when last bit D0 is 1

Settling time = 1Te + StopBits(4 Te) +(7Te～22 Te)

As such, it changes by the last bit of data.

It prescribes the time to StartBit start from the StopBits end in the timing between the frame in IEC62386-101ed1.0. (See 3.1.5.3 Timing of transmission and reception.)

Note Te = 416.67μs

Remark For Settling time of IEC62386-101ed2.0, see Appendix B DALI (IEC62386-101,102) ed2.0 communication timing.

3.1.5.3 Timing of transmission and reception

(1) Frame baud rate

DALI communication baud rate: 1200bps

Bit width[Note]: 1bit = 833.3μs±10%

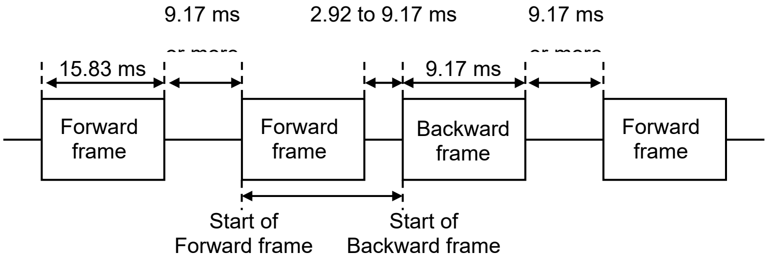
Note: The bit width is the same regardless of the Forward and Backward frames.

(2) Timing between frames(101ed1.0)

The following timing control is required for each DALI frame:

- Forward frame width: 15.83 ms±10%
- Backward frame width:9.17 ms±10%
- Communication interval between the Forward and Backward frames Forward: 2.92 to 9.17 ms
- Interval between one Forward frame and the next Forward frame: 9.17 ms or more
- Interval between one backward frame and the next forward frame:9.17 ms or more

Figure3-9 Timing Between Frames



Remark For timing between IEC62386-101ed2.0 of the frame,see Appendix B DALI (IEC62386-101,102) ed2.0 communication timing.

3.1.5.4 Commands

(1) Address byte

The DALI communication protocol has three address modes, broadcast, group, and single, for controlling slave devices. The address byte might indicate the special command used.

Table3-4 Addresses

Address type	Address byte
Broadcast address	1111111S
64 short address	0AAAAAAS (AAAAAA=0-63)
16 group address	100AAAAS (AAAA=0-15)
Special commands	101CCCC1 110CCCC1
A: Address bit S: Select bit for selecting the Direct Arc Power Control command or other commands S = ‘0’: The Direct Arc Power Control command. The data byte is the lighting control level setting. S = ‘1’: The data byte is the command number of a different command. C: Special command numbers	

Example command:

When lighting control level 254 is set for the group address 9 slave

Address byte	Data byte
10010010	11111110

Explanation:

- The group address is specified because the upper three bits of the address byte is "100".The group address 9 slave is selected because bits 4 to1 are "1001".
- The select bit (the lowest bit) is 0; therefore, this command is the Direct Arc Power Control command, and the lighting control level is directly specified by the data byte (the second byte) as 254 (the maximum lighting control level).

(2) Commands

The main commands of the DALI communication protocol are shown below.

See 3.4 DALI Commands for all commands.

Table3-5 Main commands of the DALI communication protocol

	Command Type	Command Example	Function	Address	Command/Data (8 bits)
1	**Arc power control commands** Command for manipulating the lighting control level	DIRECT ARC POWER CONTROL	Directly specifies the lighting control level with a number(with fade)	YAAAAAA0	XXXX XXXX
		OFF	Turns lighting control off	YAAAAAA1	0000 0000
		SET UP	Adds 1 to the current lighting control level (nofade), no on/off	YAAAAAA1	0000 0011
2	**Configuration commands** Command for specifying slave settings	RESET	Initializes the slave settings	YAAAAAA1	0010 0000
		ADD TO GROUP	Adds the slave specified by the address to group XXXX	YAAAAAA1	0110 XXXX
3	**Query commands** Command for obtaining slave status	QUERY STATUS	Returns STATUSINFORMATION	YAAAAAA1	10010000
4	**Special commands** Command for specifying address settings	INITIALISE	Starts address detection (specifythe slave with XXXX)	10100101	XXXX XXXX
5	**Extending special commands** Command for expansion	ENABLE DEBICE TYPE X	Add a special device XXXX	1100 0001	XXXX XXXX
6	**Application extended commands** Command for device expansions and standard updates	QUERY EXTENDED VERSION NUMBER	Returns the device type and the communication standard version supporting the device	YAAAAAA1	1111 1111

3.2 Realizing DALI Communication with RL78/I1A

3.2.1 RL78/I1A features used for DALI communication: DALI/UART4 interface

The RL78/I1A microcontroller supports a DALI/UART4 serial interface, enabling transmitting and receiving as DALI communication slaves using hardware.

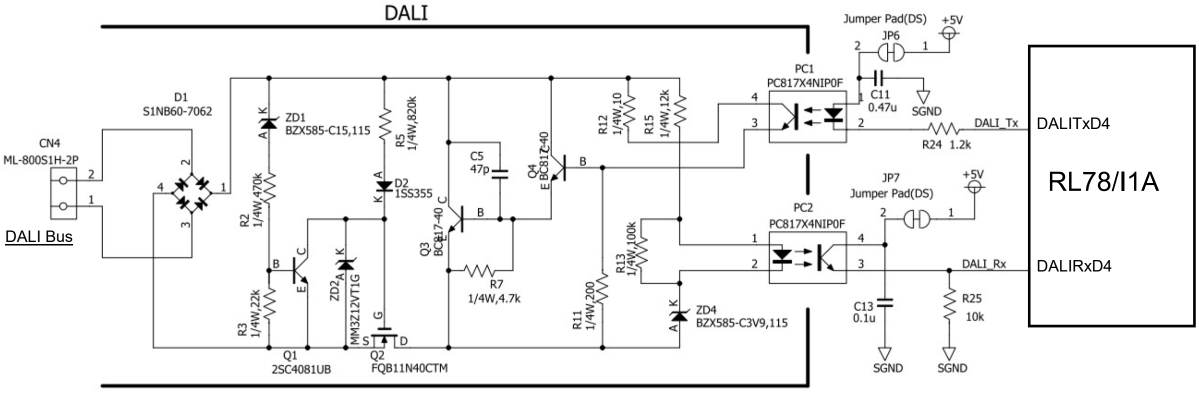
This results in reduced software processing and CPU loads during DALI communication.

3.2.1.1 Communication circuit

Figure3-10 shows the structure of the DALI communication circuit.

There are two terminals required for DALI communication: DALI receive input (RxD4 pin) and DALI transmit output (TxD4 pin).

Figure3-10 Structure of the DALI Communication Circuit



3.2.1.2 Data communication timing chart

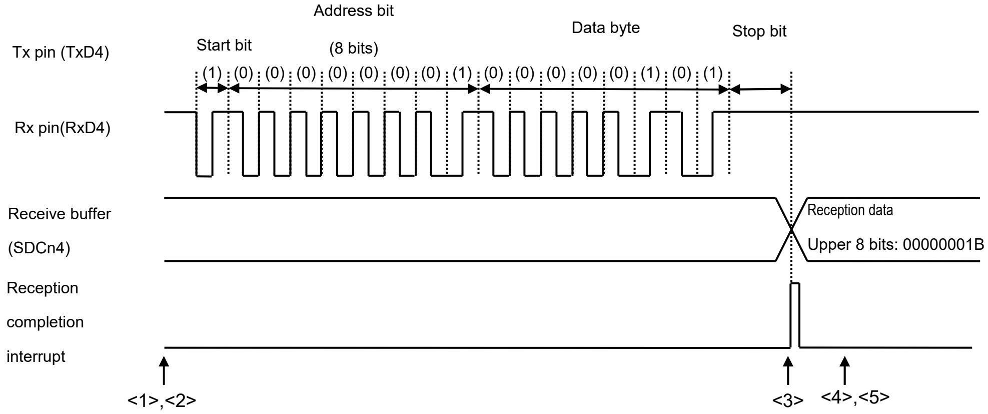
(1) When receiving data

Figure3-11shows an example timing chart when receiving Forward frame from the master.

In this example, ‘RECALL MAX LEVEL’ was received by the short address.

Command:00000001 00000101

Figure3-11 Forward Frame Reception Timing Chart Example for DALI Communication



The following is an overview of the receive operation.

<Preparing to receive data>

<1> Initialize DALI/UART4.

- Start the supply of the clock to DALI/UART4 (set the DALIEN bit of the PER1 register).
- Wait for at least 4 f_{CLK} clocks, and then specify the operation clock using the SPS4 register.
- Set up the DALI mode, operating mode, communication format, and transfer baudrate using the SOC4, SMR4n, SMR4r, SCR4n, and SDR4n registers.

<2> Set the communication wait state.

- Set the SS40/SS41 bit of the corresponding channel to set the communication waitstate.

Note Also specify interrupt and other settings as needed.

<Reception processing>

<3> After the start bit of the reception data is detected, the data is received.

- When data reception is successful, an INTSTDL4 interrupt is generated, and the received data is stored in the SDR41 register. The received data is read from the SDR41 register and processed.
- If a reception error occurs, an INTSREDL4 interrupt is generated, and the receptionerror status is stored in the SSR41 register.
- Clear the interrupt requests as needed.

Note Figure3-11 illustrates the timing of a successful reception.

<Stopping reception>

<4> Write 1 to the ST40/ST41 bit to stop communications.

<5> Disable the supply of the clock to DALI/UART4 (clear the DALIEN bit of the PER1 register).

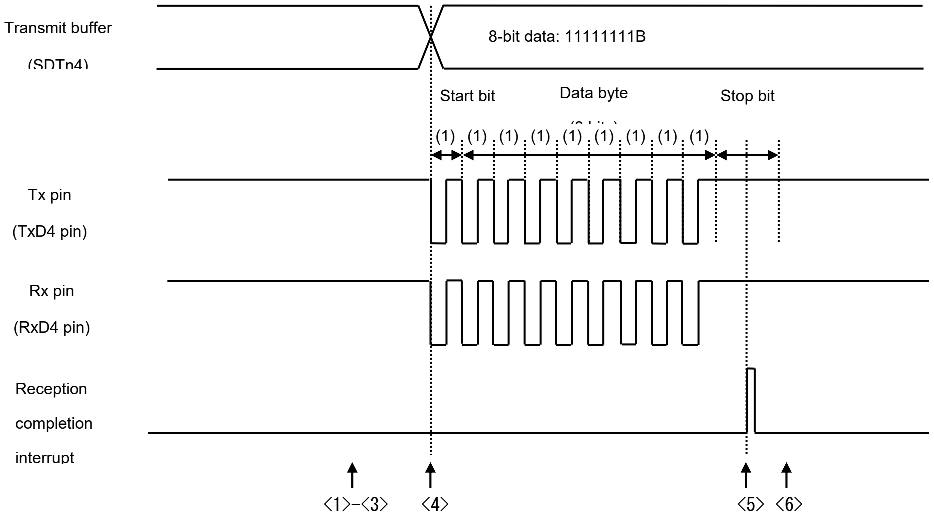
(2) When transmitting data

Figure3-12 shows an example timing chart when transmitting Backward frame to the master.

In this example, "Yes" is returned for the command received from the master (example:QUERY LAMP FAILURE that queries if there is a lighting problem).

'Yes' : 1111 1111

Figure3-12 Backward Frame Reception Timing Chart Example for DALI Communication



The following is an overview of the transmit operation.

<Preparing to transmit data>

<1> Initialize DALI/UART4.

- Start the supply of the clock to DALI/UART4 (set the DALIEN bit of the PER1 register).
- Wait for at least 4 f_{CLK} clocks, and then specify the operation clock using the SPS4 register.
- Set the DALI mode, operating mode, communication format, transfer baud rate, and output using the SOC4, SMR4n, SMR4r, SCR4n, SDR4n, SO4, and SOE4 registers.

<2> Set the communication wait state.

- Set the SS40/SS41 bit of the corresponding channel to set the communication waitstate.

<Transmission processing>

<3> Set the transmit data into the SDTL4 and SDTH4 registers, and then start the communication.

<4> When the transmission finishes, an INTSRDL4 interrupt is generated. Clear the interrupt requests as needed.

<Stopping transmission>

<5> Write 1 to the ST40/ST41 bit to stop communications.

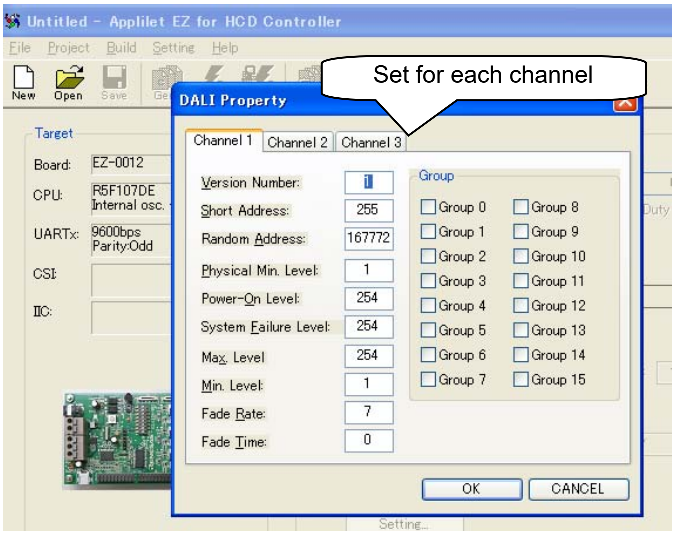
<6> Disable the supply of the clock to DALI/UART4 (clear the DALIEN bit of the PER1 register).

3.2.2 Saving DALI communication parameters

In DALI communication, some parameters must be non-volatile. Saving parameters is achieved by using the EEPROM emulation library[Note], which uses RL78/I1A data flash memory.

- The sample codes generated with Applilet EZ for HCD store the scene, faderate, fadetime, and other parameters for each slave channel using EEPROM emulation. The default valuescan be set using DALI Property. **Figure3-13** shows the settings panel.

Figure3-13 The DALI Parameter Settings Panel in Applilet EZ for HCD



Untitled - Applilet EZ for HCD Controller
File Project Build Setting Help
New Open Save
Target
Set for each channel
DALI Property
Channel 1 Channel 2 Channel 3
Group
OK CANCEL
Setting...

Board: EZ-0012
CPU: R5F107DE Internal osc.
UARTx: 9600bps Parity:Odd
CSI:
IIC:

Version Number: 1
Short Address: 255
Random Address: 167772
Physical Min. Level: 1
Power-On Level: 254
System Failure Level: 254
Max. Level: 254
Min. Level: 1
Fade Rate: 7
Fade Time: 0

☐ Group 0 ☐ Group 8
☐ Group 1 ☐ Group 9
☐ Group 2 ☐ Group 10
☐ Group 3 ☐ Group 11
☐ Group 4 ☐ Group 12
☐ Group 5 ☐ Group 13
☐ Group 6 ☐ Group 14
☐ Group 7 ☐ Group 15

- These parameters are saved as the dali_variables structure in the r_dali.h and r_dali_user.c header files. **Table3-6** shows a list of save data.

Table3-6 Parameters saved by using the EEPROM emulation feature

Item	DALI_Variables member name	Size [byte]	Item	DALI_Variables member name	Size [byte]
Version Number	version_number	1	Short Address	short_address	1
Physical Min Level	physical_min_level	1	Random Address h	random_address_h	1
Device Type	device_type	1	Random Address h	random_address_m	1
Power On Level	power_on_level	1	Random Address h	random_address_l	1
System Failure Level	system_failure_level	1	Group settings (0 to 7)	group_0_7	1
Min Level	min_level	1	Group settings (8 to 15)	group_8_15	1
Max Level	max_level	1	scene	scene	16
Fade Rate	fade_rate	1	Actual Level	actual_level	1
Fade Time	fade_time	1			

Note: See "EEPROM Emulation Library" (R01AN0707ED0100) for details on the EEPROM emulation library provided by Renesas Electronics.

3.2.3 Overview of operations

This section describes the operation of slaves in DALI communication.

Slaves receive Forward frame from the master, analyze the frame, and then perform the processing for lighting control operations and Backward frame transmissions (responses).

The peripherals that are used with the lighting control operation using DALI communication are asfollows:

- Hardware used for DALI communication: DALI/UART4; TAU; data flash memory (EEPROM emulation)
- Hardware used for lighting control: A/D converter; TAU; PGA; 16-bit timers KB0, KB1, KB2

The following shows the settings of the peripherals.

- Peripherals used and their settings
  - Channel 0 of the 16-bit timer array unit
    - Count clock f_{CLK} = 32 MHz
    - Set as a 1-ms interval timer
  - Channel 1 of the 16-bit timer array unit
    - Count clock f_{CLK} = 32 MHz
    - Set as a 100μs interval timer
  - A/D converter
    - Set the A/D conversion time to 2.97 μs
  - Programmable gain amplifier (PGA)
    - Set the gain to 8 times
    - Set input channel ANI2
  - 16-bit timer KB
    - Count clock f_{PLL} = 64 MHz
    - Set the operation mode of TKBO and TKB1 to single
    - For the timer output (TKBO00, TKBO01, and TKBO10) used, set the default level to lowand the active level to high.
    - Use the PWM output dithering feature
    - Set the PWM output frequency to 250 kHz

- TAU operation

[TAU00]

TAU00 is a 1-ms interval timer used to implement the timing control required for DALI communication. The main timing controls are as follows:

- Wait of Forward frame and Backward frame interval
  (Set to 4 ms in the program to match it with DALI communication standard)
- Time limit until receiving a second configuration command(command numbers 32 to 129) (100 ms)
- Time limit for the time allowed for processing address commands (command numbers 259 to 270) (15 min.)
- Timing of the fade processing execution (10 ms)
- DAPC sequence time limit (200 ms)
- Time measurement for automatically saving DALI communication parameters
  (saves when no changes are made to lighting control level for 100 ms after a parameter change)
- Monitoring the signal lines between the microcontroller and the DALI master board
  (a system failure is determined if the signal is low for the specified period of time) (500 ms)
- Measuring the time during which command reception is disabled
  (commands that may transmit Backward frame are disabled reception for 19 ms after receiving acommand)

[TAU01]

TAU01 is a 100-μs interval timer used to process LED lighting control feedback[Note]. See "LED Control Using RL78/I1A" (R01AN10875E) Application Note for details on feedback processing.

3.3 Organization of DALI Communication Lighting Control Software

[Software file configuration]

The following table shows the sample code file configuration of the DALI communication software.

This software can generate sample code files with Applilet EZ for HCD. See the Applilet EZ for HCD User's Manual for details on the generation process.

Table3-7 Sample Code File Configuration

Feature	File	Description
Optionbyte settings	r_init.asm	- Sets the option byte that sets the basic operations of the microcontroller. Sets the watchdog timer. Sets the operation mode and high-speed internal oscillator. Sets on-chip debugging. **Note** The option byte settings can also be specified in an integrated development environment.
LED lighting control	r_usermain.c r_LED.c r_LED1.c r_LED2.c r_LED3.c	- Controls the 3-channel LED using a 250 kHz PWM. Performs feedback processing every 100 μs and changes PWM output according to the target value. - See "LED Control Using RL78/I1A" Application Note for details.
System clock initialization processing	r_systeminit.c r_cgc.c r_lvd.c	- Sets the clock and voltage detector (LVD). Sets the operation of the high-speed internal oscillator and the PLL feature. Sets the input clock supply for the peripherals. Sets the operation mode of the A/D converter, serial array unit 0,timer array unit, serial array unit 4 (DALI/UART4),comparator/programmable gain amplifier, 16-bit timer KB Sets operation mode of LVD (reset mode).
Watchdog timer processing	r_wdt.c	- Clears the watchdog timer counter.
Timer interrupt INTTM00	r_timer.c	- Sets the interval timer mode (1 ms) to TAU00 and performs the time management for the operation of the DALI feature
DALI communication protocol processing	r_dali.c r_dali_timer.c r_dali_analyze.c r_dali_command.c r_dali_memorybank.c r_dali_user.c r_dali_variable.c	- File function group for DALI communication. Includes the processing for initializing the DALI communication feature, analyzing received commands, and lighting control levelcontrol of the specified channels. - See 3.5 **Functions for Applilet EZ for HCD DALI Communication (RL78/I1A DC/DC LED Control Evaluation Board)** for the included functions and their descriptions.

3.3.1 Operation and software flowchart

This section describes the structure of the DALI communication program in detail.

Figure3-14 shows the general flow of the DALI communication software.

- General Flowchart

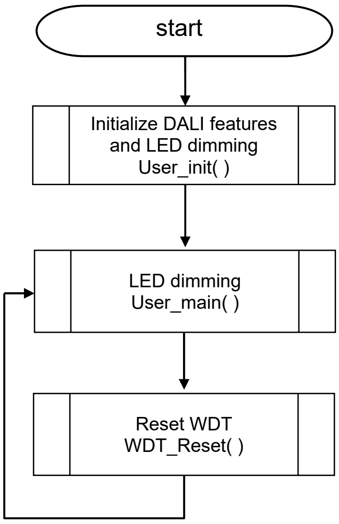
  This program can be divided into three general parts: initialization, LED lighting control, and WDT reset.

[Summary]

The program first performs initialization and then repeatedly executes LED lighting control and WDT reset clear.

DALI processing is performed as part of the LED lighting control. If data is received, acommand analysis is performed, and a response or lighting control is executed when required.

Figure3-14 General Flowchart



- Initialization processing flowchart

  The initialization processing User_init( ) executes the following two processes. A flowchart for these processes is shown in **Figure3-15**.

  Initializing the peripherals related to LED lighting control[Note]: LED_init()

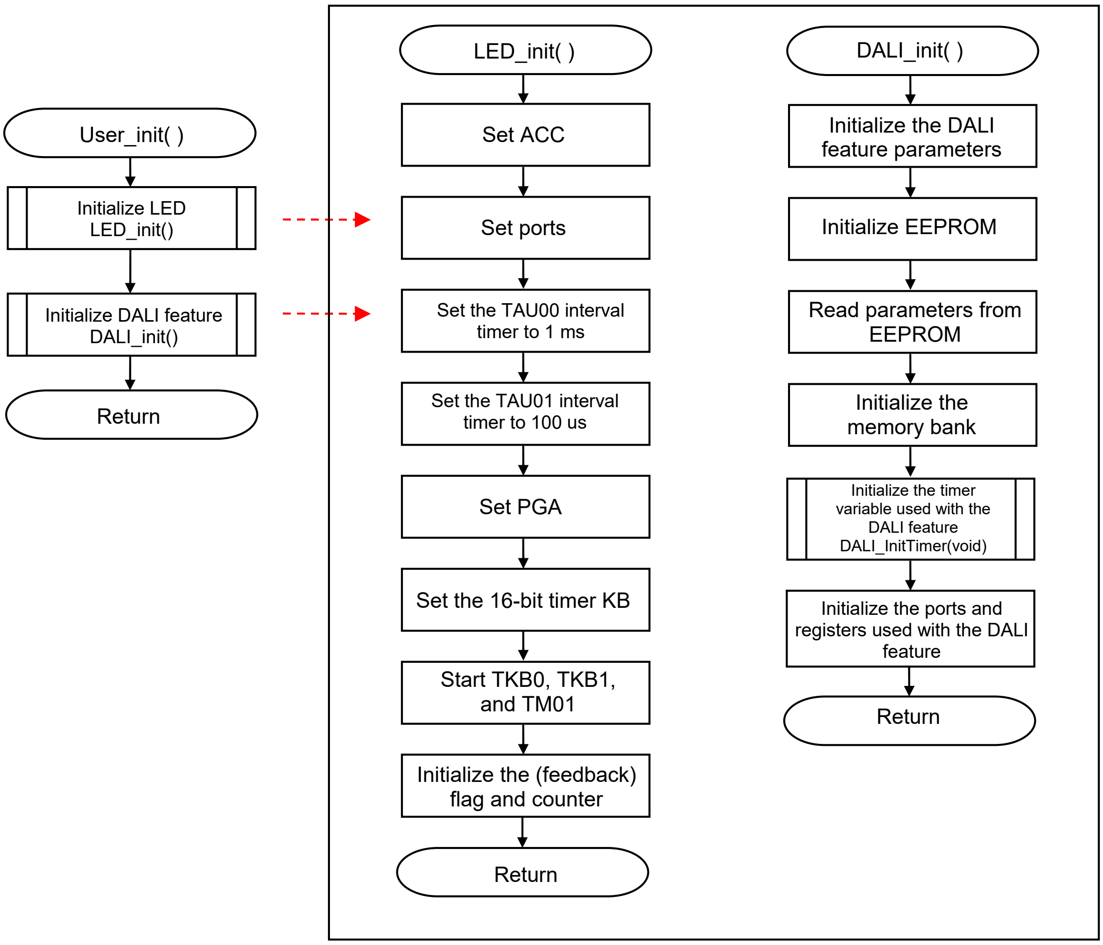
  Initializing the peripherals related to the DALI communication feature: DALI_init()

  Note The initialization of peripherals related to LED lighting control is described in the "LED Control Using RL78/I1A" Application Note. See this document for additional details.

[Summary]

User_init() includes LED_init() and DALI_init(). DALI_init() initializes the parameters and EEPROM emulation. DALI_init_Timer() initializes the timer variables used by DALI features.

Figure3-15 Initialization Processing Flowchart


User_init( )
Initialize LED
LED_init()
Initialize DALI feature
DALI_init()
Return
LED_init( )
Set ACC
Set ports
Set the TAU00 interval timer to 1 ms
Set the TAU01 interval timer to 100 us
Set PGA
Set the 16-bit timer KB
Start TKB0, TKB1, and TM01
Initialize the (feedback) flag and counter
Return
DALI_init( )
Initialize the DALI feature parameters
Initialize EEPROM
Read parameters from EEPROM
Initialize the memory bank
Initialize the timer variable used with the DALI feature
DALI_InitTimer(void)
Initialize the ports and registers used with the DALI feature
Return

- DALI periodic processing flowchart

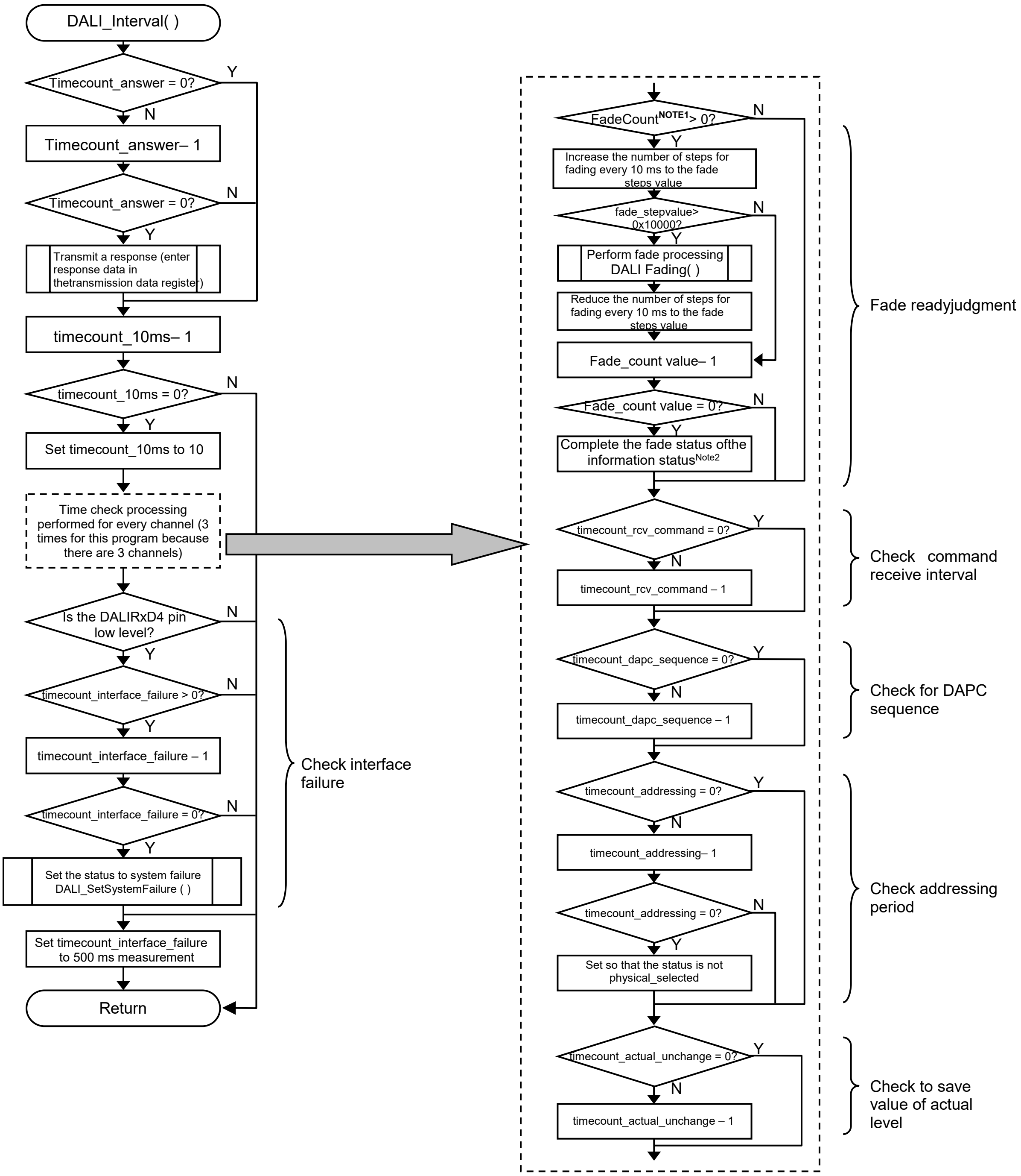
  DALI_Interval() is the basic function of the software timers used with DALI communication processing. A flowchart for this process is shown in **Figure3-16**.

[Summary]

DALI_Interval() is a function that performs periodic processing at 1 ms using TAU00.

This function performes the configuration and control multiple software timer inside, and it is a function of the standard of process that requires time management in the DALI communication processing. It is shown in **Table3-8** for the software timer variable.

Figure3-16 DALI Interval Processing (1 ms) Flowchart


DALI_Interval( )
Timecount_answer = 0?
Y
N
Timecount_answer– 1
Timecount_answer = 0?
N
Y
Transmit a response (enter response data in thetransmission data register)
timecount_10ms– 1
timecount_10ms = 0?
N
Y
Set timecount_10ms to 10
Time check processing performed for every channel (3 times for this program because there are 3 channels)
Is the DALIRxD4 pin low level?
N
Y
timecount_interface_failure > 0?
N
Y
timecount_interface_failure – 1
timecount_interface_failure = 0?
N
Y
Set the status to system failure DALI_SetSystemFailure ( )
Check interface failure
Set timecount_interface_failure to 500 ms measurement
Return
FadeCountNOTE1> 0?
N
Y
Increase the number of steps for fading every 10 ms to the fade steps value
fade_stepvalue> 0x10000?
N
Y
Perform fade processing DALI Fading( )
Reduce the number of steps for fading every 10 ms to the fade steps value
Fade_count value– 1
Fade_count value = 0?
N
Y
Complete the fade status ofthe information statusNote2
Fade readyjudgment
timecount_rcv_command = 0?
Y
N
timecount_rcv_command – 1
Check command receive interval
timecount_dapc_sequence = 0?
Y
N
timecount_dapc_sequence – 1
Check for DAPC sequence
timecount_addressing = 0?
Y
N
timecount_addressing– 1
timecount_addressing = 0?
N
Y
Set so that the status is not physical_selected
Check addressing period
timecount_actual_unchange = 0?
Y
N
timecount_actual_unchange – 1
Check to save value of actual level

Notes1. A fade count value of n means fade time＝10 ms×n.

Notes2. This refers to the STATUS INFORMATION byte.

bit 4 fade ready;

‘0’ = fade is ready;

‘1’ = fade is running

Table3-8 List of Software Timer Variables

List of Variables	
timecount_answer	Timer counter for waiting for Backward frame return following Forward frame reception
timecount_10ms	Timer counter for generating a 10 ms interval
timecount_rcv_command	Timer counter for measuring the time until two configuration commands have been received
timecount_dapc_sequence	Timer counter for measuring the time limit during the DAPC sequence
timecount_addressing	Timer counter for measuring the period available for processing address commands
timecount_actual_unchange	Timer counter for acquiring the timing for saving DALI parameters
timecount_interface_failure	Timer counter for checking the DALI master board and its communication lines
timecount_prohibit_reception	Timer counter for measuring the period for which command reception is disabled

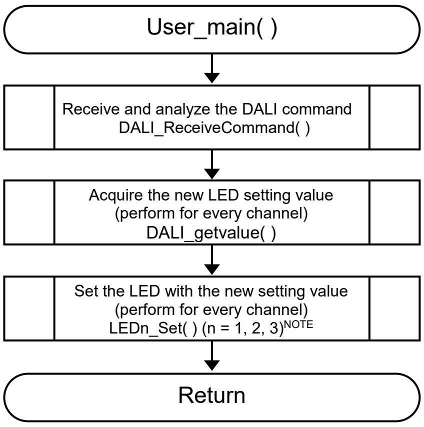
- LED lighting control processing flowchart

  User_main() receives and analyzes the most important command as a DALI communication protocol and performs LED lighting control. A flowchart of these processes is shown in **Figure3-17**.

[Summary]

User_main() is composed of DALI_ReceiveCommand(), which receives and analyzes DALI commands; DALI_getvalue(), which acquires the LED lighting control level for each channel;and LEDn_set(), which sets a new lighting control level. An LEDn_set() function exists for each channel.

Figure3-17 LED Lighting Control Processing Flowchart


User_main( )
Receive and analyze the DALI command
DALI_ReceiveCommand( )
Acquire the new LED setting value
(perform for every channel)
DALI_getvalue( )
Set the LED with the new setting value
(perform for every channel)
LEDn_Set( ) (n = 1, 2, 3)NOTE
Return

Note For details about LEDn_Set( ), see "LED Control Using RL78/I1A" Application Note.

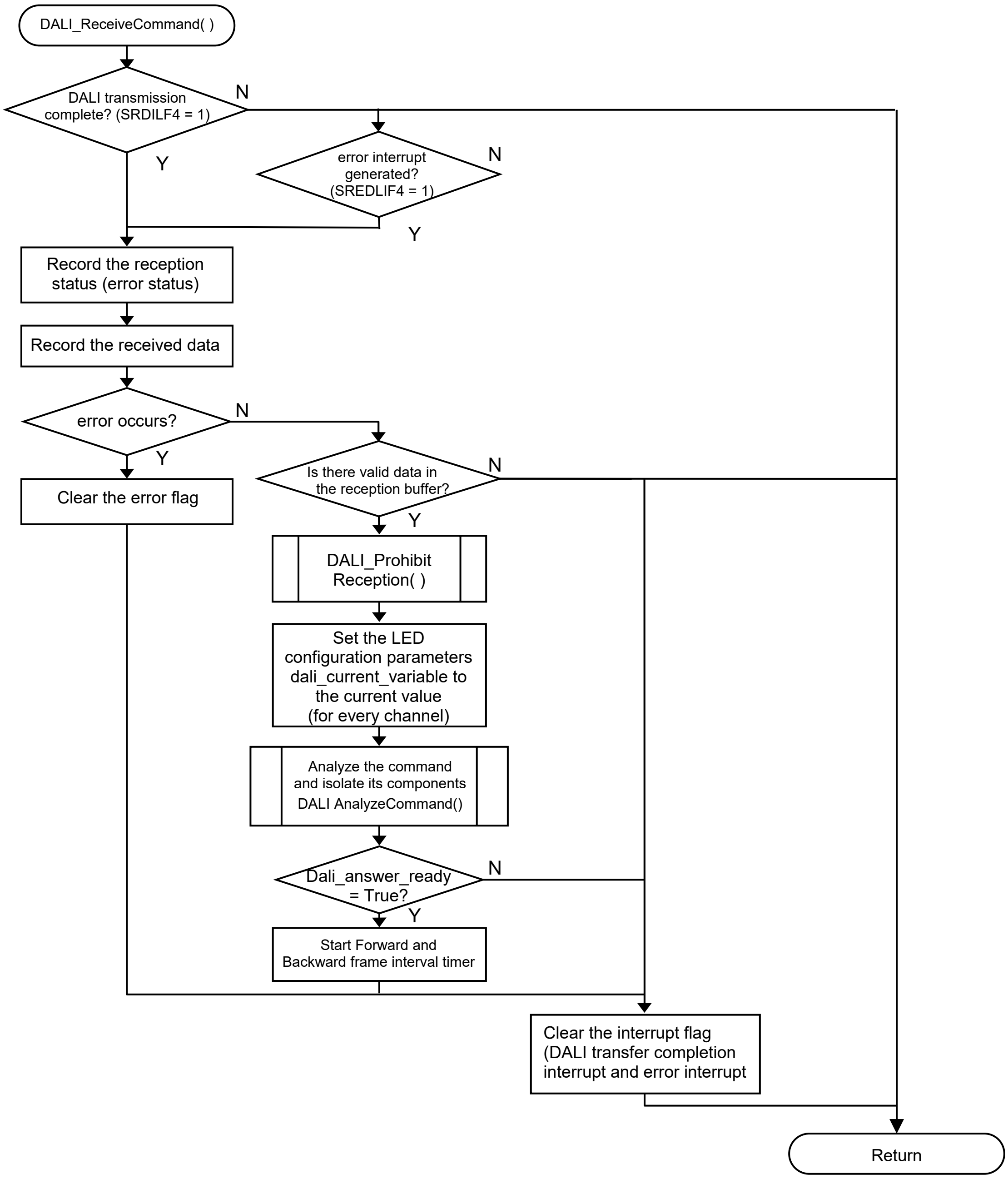
- DALI command reception and analysis flowchart

  The flowchart for DALI_ReceiveCommand(), which receives and analyzes DALI commands, is shown in **Figure3-18**.

[Summary]

The reception of DALI commands with the DALI_ReceiveCommand() performs checks by polling interrupt flags (SRDLIF4 and SREDLIF4). The flags are checked, and if a commandwas normally received, the received data is analyzed. If a response is necessary, the timercounter for Forward frame and Backward frame interval wait (timecount_answer) is set to 4 ms for response time management. If the command received from the DALI_ProhibitReception() function requires a response, the time expected until the response is sent and the reception disable status are set.

Figure3-18 DALI Command Reception and Analysis Flowchart


DALI_ReceiveCommand( )
DALI transmission complete? (SRDILF4 = 1)
N
Y
error interrupt generated? (SREDLIF4 = 1)
N
Y
Record the reception status (error status)
Record the received data
error occurs?
N
Y
Clear the error flag
Is there valid data in the reception buffer?
N
Y
DALI_Prohibit Reception( )
Set the LED configuration parameters dali_current_variable to the current value (for every channel)
Analyze the command and isolate its components DALI AnalyzeCommand()
Dali_answer_ready = True?
N
Y
Start Forward and Backward frame interval timer
Clear the interrupt flag (DALI transfer completion interrupt and error interrupt
Return

- Command analysis and isolation processing flowchart

  DALI_AnalyzeCommand() executes the command analysis and isolation processing within DALI_ReceiveCommand().A flowchart of these processes is shown in **Figure3-19**.

[Summary]

The received DALI command contains an address and a command in two bytes of data. The address and command are separated and processed.

○ Address analysis:

The address type is determined by the size of the address value (see **3.1.5.4 Commands**.)
The type is determined as a special command if it is not a broadcast address, short address, or group address.
The special command related processing, DALI_SpecialCommand(), is executed if the type is determined to be a special command.
If the type is not a special command (not a broadcast address, short address, or groupaddress), the slave determines whether its address is targeted by the processing. If it is a target, the slave starts the command analysis.

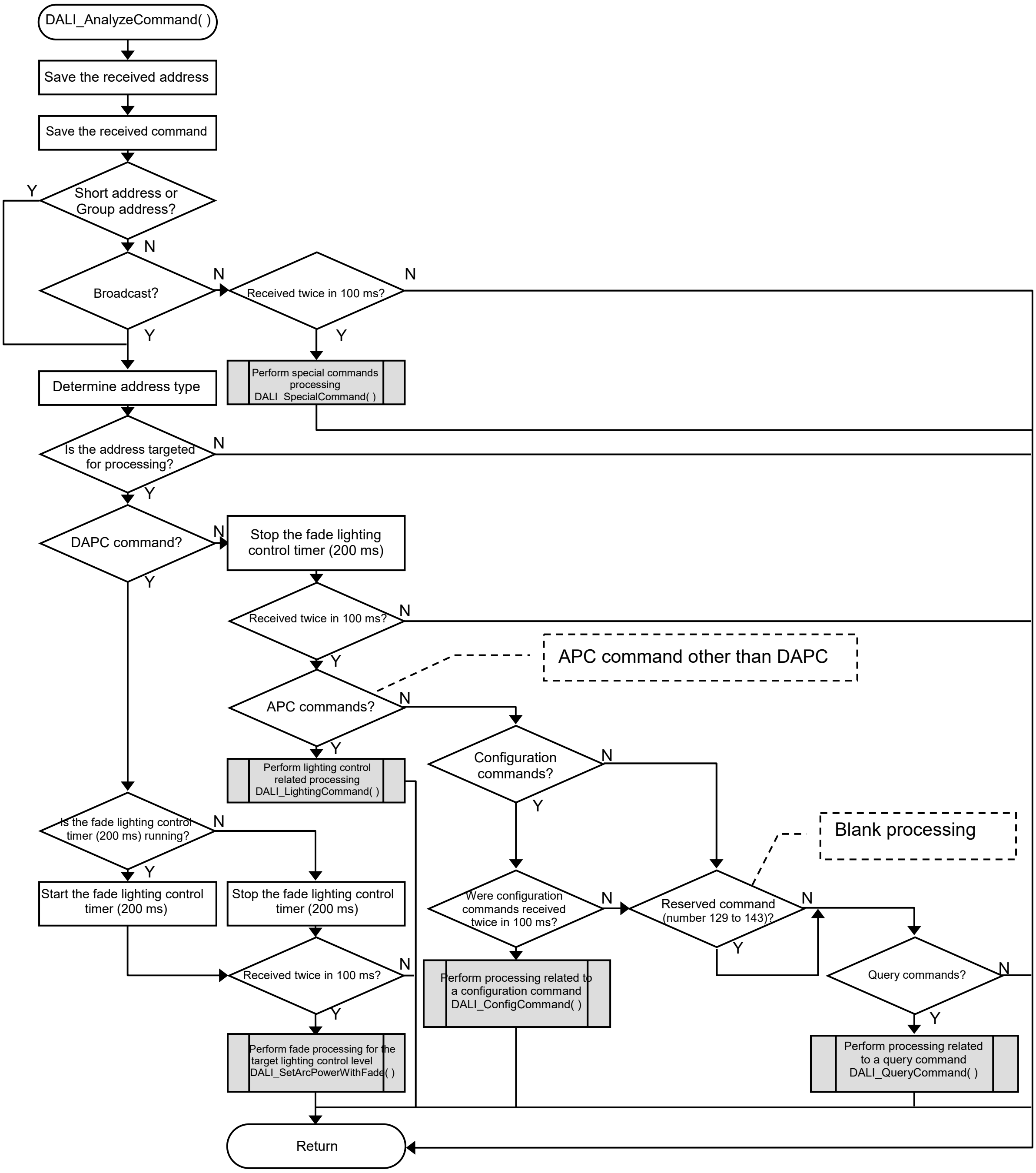
○ Command analysis:

The processing is performed according to the command type.
There are the following four types of commands. The function of these commands are contained in r_dali_command.c.

- If the type is a Direct arc power control command, *DALI_SetArcPowerWithFade()* is executed to perform the fade processing for the target lighting control level.
- If the type is an Arc power command that is not DAPC, *DALI_LightingCommand()* is executed to perform lighting control related processing[Note].
- If the type is a Configuration Command and is received twice within 100 ms,*DALI_ConfigCommand()* is executed to perform processing related to the Configuration Command[Note].
- If the type is a Query Command, *DALI_QueryCommand()* is executed to perform processing related to Query Commands[Note].

Note Some processes are not depicted in the flowchart. See the program for details.

Figure3-19 Command Analysis and Isolation Processing Flowchart

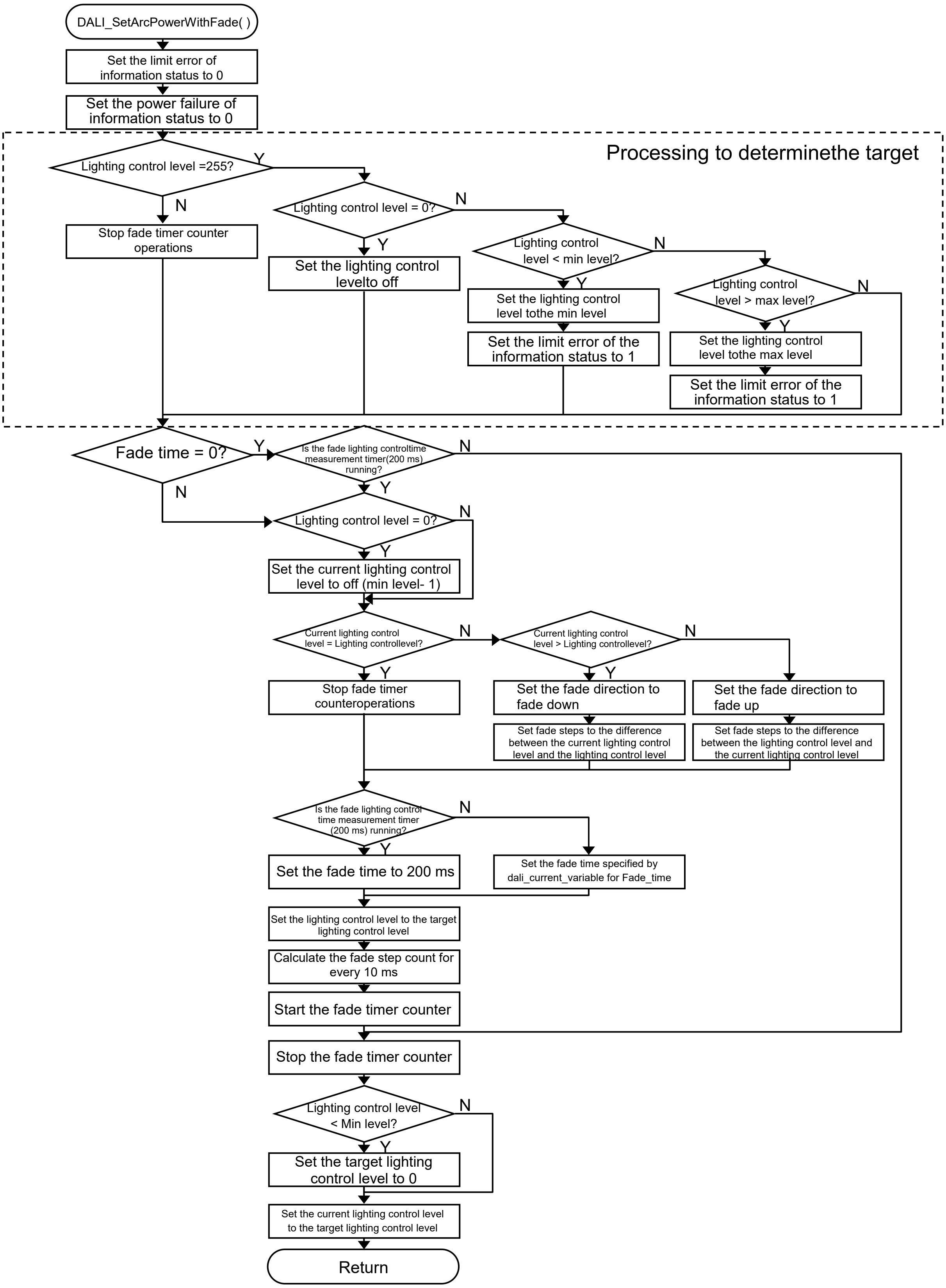

DALI_AnalyzeCommand( )
Save the received address
Save the received command
Short address or Group address?
Y
N
Broadcast?
N
Received twice in 100 ms?
N
Y
Y
Determine address type
Perform special commands processing DALI_SpecialCommand( )
Is the address targeted for processing?
N
Y
DAPC command?
N
Stop the fade lighting control timer (200 ms)
Y
Received twice in 100 ms?
N
Y
APC command other than DAPC
APC commands?
N
Y
Perform lighting control related processing DALI_LightingCommand( )
Configuration commands?
N
Y
Blank processing
Is the fade lighting control timer (200 ms) running?
N
Y
Start the fade lighting control timer (200 ms)
Stop the fade lighting control timer (200 ms)
Were configuration commands received twice in 100 ms?
N
Reserved command (number 129 to 143)?
N
Y
Received twice in 100 ms?
N
Y
Perform processing related to a configuration command DALI_ConfigCommand( )
Query commands?
N
Y
Perform fade processing for the target lighting control level DALI_SetArcPowerWithFade( )
Perform processing related to a query command DALI_QueryCommand( )
Return

- DAPC command and fade processing flowchart
  DALI_SetArcPowerWithFade() is a function for executing the target lighting control level and time management when a DAPC command is received. A flowchart ofthese processes is shown in **Figure3-20**.

[Summary]

DALI_SetArcPowerWithFade() is called from DALI_AnalyzeCommand() and uses the received DAPC command as the lighting control level. The lighting control level is restricted so that it remains within the maximum and minimum levels in the function. This restricted value becomes the target lighting control level, and fade is performed by using the time set by fadetime. When the light is off, the light turns on at the target lighting control level.

Figure3-20 DAPC Command and Fade Processing Flowchart


DALI_SetArcPowerWithFade( )
Set the limit error of information status to 0
Set the power failure of information status to 0
Processing to determinethe target
Lighting control level =255?
Y
N
Stop fade timer counter operations
Lighting control level = 0?
N
Y
Set the lighting control levelto off
Lighting control level < min level?
N
Y
Set the lighting control level tothe min level
Set the limit error of the information status to 1
Lighting control level > max level?
N
Y
Set the lighting control level tothe max level
Set the limit error of the information status to 1
Fade time = 0?
Y
N
Is the fade lighting controltime measurement timer(200 ms) running?
N
Y
Lighting control level = 0?
N
Y
Set the current lighting control level to off (min level- 1)
Current lighting control level = Lighting controllevel?
N
Y
Current lighting control level > Lighting controllevel?
N
Y
Stop fade timer counteroperations
Set the fade direction to fade down
Set the fade direction to fade up
Set fade steps to the difference between the current lighting control level and the lighting control level
Set fade steps to the difference between the lighting control level and the current lighting control level
Is the fade lighting control time measurement timer (200 ms) running?
N
Y
Set the fade time to 200 ms
Set the fade time specified by dali_current_variable for Fade_time
Set the lighting control level to the target lighting control level
Calculate the fade step count for every 10 ms
Start the fade timer counter
Stop the fade timer counter
Lighting control level < Min level?
N
Y
Set the target lighting control level to 0
Set the current lighting control level to the target lighting control level
Return

- LED lighting control level acquisition processing flowchart

  DALI_GetValue() is a function for acquiring the LED lighting control level. A flowchart ofthis process is shown in **Figure3-21**.

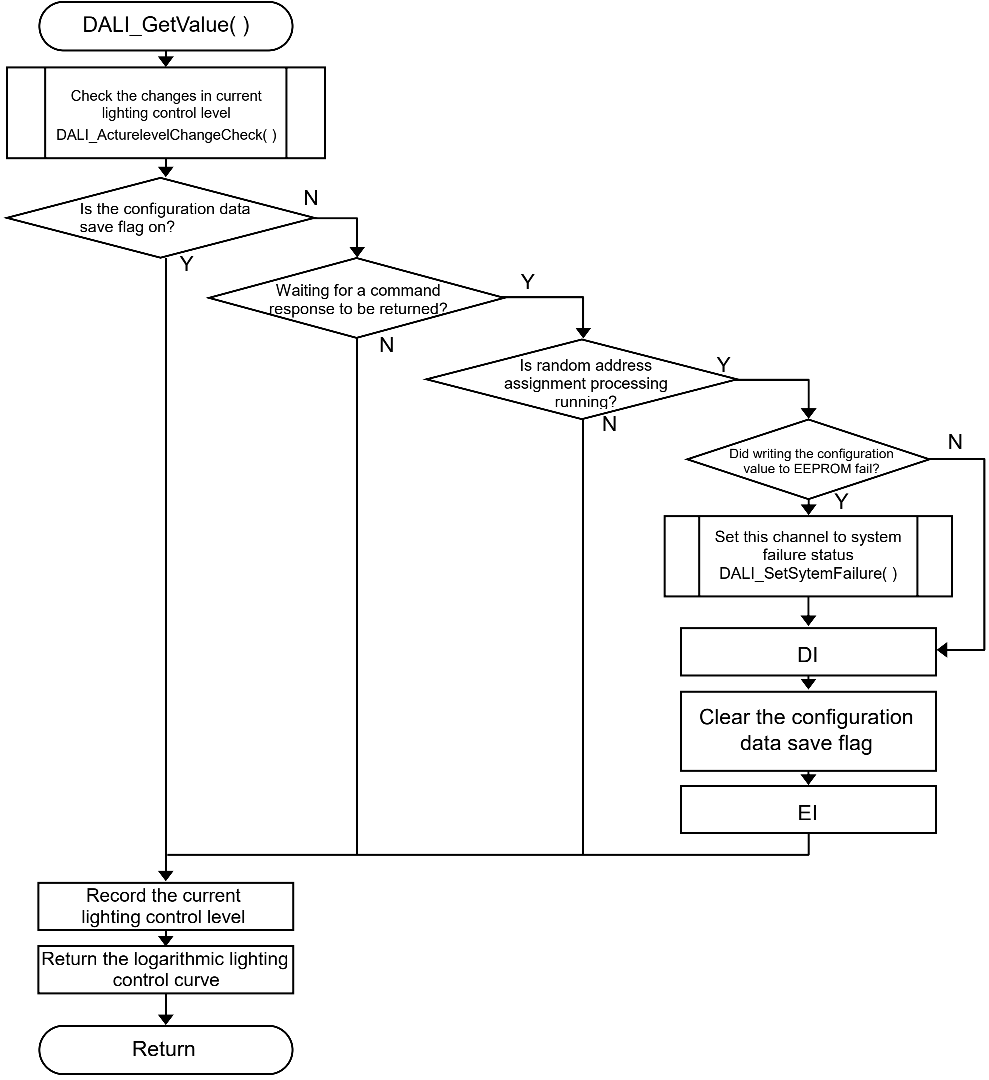
[Summary]

DALI_GetValue() is a function called from within User_main() and returns the LED lighting control level. This value is obtained from the process of receiving and analyzing the DALI command by the DALI_ReceiveCommand().Configuration data is saved here as well because DALI_GetValue() is constantly being executed. Configuration data is saved when all of the following conditions are satisfied: the configuration data save flag is on (the configuration data or the lighting control level is being changed), no DALI command response is waiting to be sent, and random address assignment processing is not being performed.
A system failure status results when saving the configuration data fails.

○ Current lighting control level change check processing (DALI_ActureLevelChangeCheck())

The lighting control level when the power is turned on must be the lighting control level that was last set when POWER ON LEVEL of the configuration data is 255. Therefore, the lighting control level (Actual level) is saved when it changes. Saving occurs when the lighting control level does not change for 500 ms following the last change.

Figure3-21 LED Lighting Control Level Acquisition Processing


DALI_GetValue( )
Check the changes in current lighting control level
DALI_ActurelevelChangeCheck( )
Is the configuration data save flag on?
N
Y
Waiting for a command response to be returned?
Y
N
Is random address assignment processing running?
Y
N
Did writing the configuration value to EEPROM fail?
N
Y
Set this channel to system failure status
DALI_SetSytemFailure( )
DI
Clear the configuration data save flag
EI
Record the current lighting control level
Return the logarithmic lighting control curve
Return

- Flowchart for setting system failure status

DALI_SetSystemFailure() is a function for setting system failure status when an error occurs,such as when writing the configuration data fails. A flowchart of this process is shown in **Figure3-22**.

[Summary]

The DALI_SetSystemFailure() sets system failure status when a specified channel is set with the reset value saved in EEPROM.

The process for this operation is as follows:

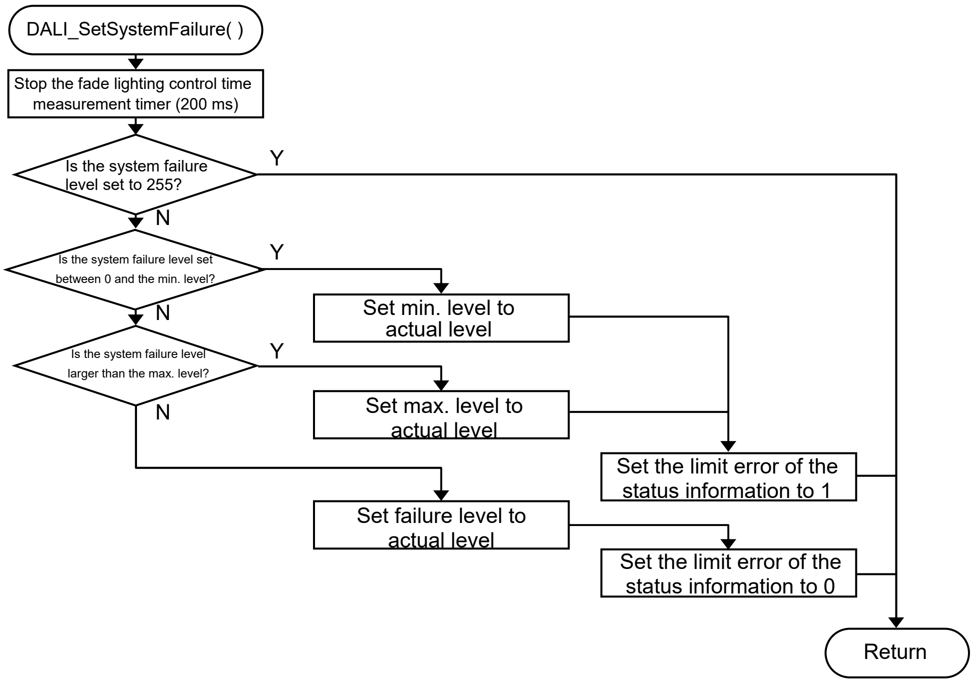
- Stop the fade processing measurement timing (200 ms)
- Change the Actual level and the information status according to the Failure level and the following conditions:
- When Failure_level = 255, do not change Actual level.
- When 0 < Failure_level ≦ Min level, set Actual level to Min level.
- When Failure_level > Max level, set Actual level to Max level.
- When Min level ≦ Failure_level ≦ Max level, or, when Failure_level = 0, set Actual level toFailure level.
- The information status bit 3: Limit Error[Note] changes based on the above processing.

Note Limit Error

0: The previous value of Arc power is between the Min and Max levels or Arc power isoff.

1: The previous value of Arc power is not within the Min and Max levels.

Figure3-22 Processing for Setting System Failure Status


DALI_SetSystemFailure( )
Stop the fade lighting control time measurement timer (200 ms)
Is the system failure level set to 255?
Y
N
Is the system failure level set between 0 and the min. level?
Y
N
Set min. level to actual level
Is the system failure level larger than the max. level?
Y
N
Set max. level to actual level
Set the limit error of the status information to 1
Set failure level to actual level
Set the limit error of the status information to 0
Return

3.4 DALI Commands

(1) Arc power control commands

These commands are used to adjust the lighting control level.

Number	Code	Name	Description
—	YAAA AAA0 XXXX XXXX	DIRECT ARC POWER CONTROL	Adjusts the lighting control level to any level XXXX XXXX according to the Fade time.
0	YAAA AAA1 0000 0000	OFF	Turns off lighting.
1	YAAA AAA1 0000 0001	UP	Increases the lighting control level for 200 ms according to the Fade rate[Note1].
2	YAAA AAA1 0000 0010	DOWN	Decreases the lighting control level for 200 ms according to the Fade rate[Note1].
3	YAAA AAA1 0000 0011	STEP UP	Increments the lighting control level (without fade)[Note1].
4	YAAA AAA1 0000 0100	STEP DOWN	Decrements the lighting control level (without fade)[Note1].
5	YAAA AAA1 0000 0101	RECALL MAX LEVEL	Maximizes the lighting control level (without fade) [Note2].
6	YAAA AAA1 0000 0110	RECALL MIN LEVEL	Minimizes the lighting control level (without fade) [Note2].
7	YAAA AAA1 0000 0111	STEP DOWN AND OFF	Decrements the lighting control level and turns off lighting if the level is at the minimum (without fade).
8	YAAA AAA1 0000 1000	ON AND STEP UP	Increments the lighting control level and turns on lighting if lighting is off (with fade).
9	YAAA AAA1 0000 1001	ENABLE DAPC SEQUENCE	It shows the repeat start of the DAPC command.
10	YAAA AAA1 0000 1010	GO TO LAST ACTIVE LEVEL	Adjusts the lighting control level to the last light control level according to the Fade time. (Command that exist only in IEC62386-102ed2.0)
11 to 15	YAAA AAA1 0000 11XX	RESERVED	[Reserved]
16 to 31	YAAA AAA1 0000 XXXX	GO TO SCENE	Adjusts the lighting control level for Scene XXXX according to the fade time.

Remarks Y: <0>Short address
<1>Group address/broadcast address
A: Address bit
X: Data

Note1: The slave does not move from the turn-off state (Actual level = 0) to the turn-on state, or vice versa.

Note2: If the slave is in the turn-off state (Actual Level = 0), it moves to the turn-off state.

(2) Configuration commands

These commands are used to change the slave settings.

Number	Code	Name	Description
32	YAAA AAA1 0010 0000	RESET	Makes a slave an RESET state.
33	YAAA AAA1 0010 0001	STORE ACTUAL LEVEL IN THE DTR (STORE ACTUAL LEVEL IN DTR0)	Saves the current lighting control level to the DTR (DTR0). (In the parenthesis is a name in IEC62386-102ed2.0)
34	YAAA AAA1 0010 0010	SAVE PERSISTENT VARIABLES	Saves a variable in nonvolatile memory (NVM). (Command that exist only in IEC62386-102ed2.0)
35	YAAA AAA1 0010 0011	SET OPERATING MODE	Sets data of DTR0 as an operating mode. (Command that exist only in IEC62386-102ed2.0)
36	YAAA AAA1 0010 0100	RESET MEMORY BANK	Changes to the reset value the specified memory bank in DTR0. (Command that exist only in IEC62386-102ed2.0)
37	YAAA AAA1 0010 0101	IDENTIFY DEVICE	Starts the identification state of the device. (Command that exist only in IEC62386-102ed2.0)
38 to 41	YAAA AAA1 0010 XXXX	RESERVED	[Reserved]
42	YAAA AAA1 0010 1010	STORE THE DTR AS MAX LEVEL (SET MAX LEVEL)	Specifies the DTR data as the maximum lighting control level. (In the parenthesis is a name in IEC62386-102ed2.0)
43	YAAA AAA1 0010 1011	STORE THE DTR AS MIN LEVEL (SET MIN LEVEL)	Specifies the DTR data as the minimum lighting control level. (In the parenthesis is a name in IEC62386-102ed2.0)
44	YAAA AAA1 0010 1100	STORE THE DTR AS SYSTEM FAILURE LEVEL (SET SYSTEM FAILURE LEVEL)	Specifies the DTR data as the "FAILURELEVEL". (In the parenthesis is a name in IEC62386-102ed2.0)
45	YAAA AAA1 0010 1101	STORE THE DTR AS POWER ON LEVEL (SET POWER ON LEVEL)	Specifies the DTR data as the "POWER ONLEVEL". (In the parenthesis is a name in IEC62386-102ed2.0)
46	YAAA AAA1 0010 1110	STORE THE DTR AS FADE TIME (SET FADE TIME)	Specifies the DTR data as the Fade time. (In the parenthesis is a name in IEC62386-102ed2.0)
47	YAAA AAA1 0010 1111	STORE THE DTR AS FADE RATE (SET FADE RATE)	Specifies the DTR data as the Fade rate. (In the parenthesis is a name in IEC62386-102ed2.0)
48	YAAA AAA1 0011 0000	SET EXTENDED FADE TIME	Specifies the DTR data as the Extended Fade Time. (Command that exist only in IEC62386-102ed2.0)
49 to 63	YAAA AAA1 0011 XXXX	RESERVED	[Reserved]
64 to 79	YAAA AAA1 0100 XXXX	STORE THE DTR AS SCENE (SET SCENE)	Specifies the DTR data as Scene XXXX. (In the parenthesis is a name in IEC62386-102ed2.0)
80 to 95	YAAA AAA1 0101 XXXX	REMOVE FROM SCENE	Deletes the Scene XXXX setting. (Specifies 1111 1111 for the scene register.)
96 to 111	YAAA AAA1 0110 XXXX	ADD TO GROUP	Adds the slave to Group XXXX.
112 to 127	YAAA AAA1 0111 XXXX	REMOVE FROM GROUP	Deletes the slave from Group XXXX.
128	YAAA AAA1 1000 0000	STORE DTR AS SHORT ADDRESS (SET SHORT ADDRESS)	Specifies the DTR data as a Short Address. (In the parenthesis is a name in IEC62386-102ed2.0)
129	YAAA AAA1 1000 0001	ENABLE WRITE MEMORY	Allows writing of the memory bank.
130 to 143	YAAA AAA1 1000 XXXX	RESERVED	[Reserved]

Remarks Y: <0>Short address

<1>Group address/broadcast address

A: Address bit

X: Data

DTR: Data Transfer Register

(3) Query commands

These commands are used to query the status of slave.

Number	Code		Name	Description
144	Fw: Bw:	YAAA AAA1 1001 0000 STATUS INFORMATION	QUERY STATUS	Returns "STATUS INFORMATION"[Note1].
145	Fw: Bw:	YAAA AAA1 1001 0001 YES'/'NO'	QUERY CONTROL GEAR (QUERY CONTROL GEAR PRESENT)	Is there a slave that can communicate? (In the parenthesis is a name in IEC62386-102ed2.0)
146	Fw: Bw:	YAAA AAA1 1001 0010 YES'/'NO'	QUERY LAMP FAILURE	Is there a lamp problem? [Note2]
147	Fw: Bw:	YAAA AAA1 1001 0011 YES'/'NO'	QUERY LAMP POWER ON	Is a lamp on?
148	Fw: Bw:	YAAA AAA1 1001 0100 YES'/'NO'	QUERY LIMIT ERROR	Is the specified lighting control level out of the range from the minimum to the maximum values?
149	Fw: Bw:	YAAA AAA1 1001 0101 YES'/'NO'	QUERYRESETSTATE	Is the slave in 'RESET STATE'?
150	Fw: Bw:	YAAA AAA1 1001 0110 YES'/'NO'	QUERY MISSING SHORT ADDRESS	Does the slave not have a short address?
151	Fw: Bw:	YAAA AAA1 1001 0111 (Standard number)	QUERY VERSION NUMBER	What is the corresponding IEC standard number?
152	Fw: Bw:	YAAA AAA1 1001 1000 (DTR content)	QUERY CONTENT DTR (QUERY CONTENT DTR0)	What is the DTR content? (In the parenthesis is a name in IEC62386-102ed2.0)
153	Fw: Bw:	YAAA AAA1 1001 1001 (Device type)	QUERY DEVICE TYPE	What is the device type? [Note3] (fluorescent lamp:0000 0000) (IEC62386-207 is 6 fixed)
154	Fw: Bw:	YAAA AAA1 1001 1010 (Minimum limit on the hardware)	QUERY PHYSICAL MINIMUM LEVEL	What is the minimum lighting control level specified by the hardware?
155	Fw: Bw:	YAAA AAA1 1001 1011 YES'/'NO'	QUERY POWER FAILURE	Has the slave operated without the execution of reset-command or the adjustment of the lighting control level?
156	Fw: Bw:	YAAA AAA1 1001 1100 (DTR1 content)	QUERY CONTENT DTR1	What is the DTR1 content?
157	Fw: Bw:	YAAA AAA1 1001 1101 (DTR2 content)	QUERY CONTENT DTR2	What is the DTR2 content?
158	Fw: Bw:	YAAA AAA1 1001 1110 (OperatingMode)	QUERY OPERATING MODE	What is the Operating Mode? (Only IEC62386-102ed2.0)
159	Fw: Bw:	YAAA AAA1 1001 1111 (Light source type)	QUERY LIGHT SOURCE TYPE	What is the Light source type? (Only IEC62386-102ed2.0)
160	Fw: Bw:	YAAA AAA1 1010 0000 (ACTUAL LEVEL)	QUERY ACTUAL LEVEL	What is the "ACTUAL LEVEL" (the current lighting control level)?
161	Fw: Bw:	YAAA AAA1 1010 0001 (Maximum lighting control level)	QUERY MAX LEVEL	What is the maximum lighting control level?
162	Fw: Bw:	YAAA AAA1 1010 0010 (Minimum lighting control level)	QUERY MIN LEVEL	What is the minimum lighting control level?
163	Fw: Bw:	YAAA AAA1 1010 0011 (POWER ON LEVEL)	QUERY POWER ON LEVEL	What is the "POWER ON LEVEL" (the lighting control level when the power is turned on)?
164	Fw: Bw:	YAAA AAA1 1010 0100 (FAILURE LEVEL)	QUERY SYSTEM FAILURE LEVEL	What is the "SYSTEM FAILURE LEVEL" (the lighting control level when a failure occurs)?
165	Fw: Bw:	YAAA AAA1 1010 0101 <Higher> Time <Lower> Rate	QUERY FADE TIME/FADE RATE	What are the Fade time and Fade rate?
166	Fw: Bw:	YAAA AAA1 1010 0110 (SpesificMode)	QERY MANUFACTURER SPECIFIC MODE	What is the Specific Mode? (Command that exist only in IEC62386-102ed2.0)
167	Fw: Bw:	YAAA AAA1 1010 0111 (Next DeviceType)	QUERY NEXT DEVICE TYPE	What is the next Device Type? (Command that exist only in IEC62386-102ed2.0)

Number		Code	Name	Description
168	Fw: Bw:	YAAA AAA1 1010 1000 (SpesificMode)	QUERY EXTENDED FADE TIME	What is the Extended Fade Time? (Command that exist only in IEC62386-102ed2.0)
169	Fw: Bw:	YAAA AAA1 1010 1010 (SpesificMode)	QUERY CONTROL GEAR FAILURE	Does a slave have the abnormality? (Command that exist only in IEC62386-102ed2.0)
170 to 175		YAAA AAA1 1010 XXXX	RESERVED	[Reserved]
176 to 191	Fw: Bw:	YAAA AAA1 1011 XXXX (Lighting control level)	QUERY SCENE LEVEL (SCENES 0-15)	What is the lighting control level for SCENE XXXX?
192	Fw: Bw:	YAAA AAA1 1100 0000 <0>No or <1>Yes for each bit	QUERY GROUPS 0-7	Does the slave belong to a group among groups 0 to 7? (Each bit corresponds to agroup.)
193	Fw: Bw:	YAAA AAA1 1100 0001 <0>No or <1>Yes for each bit	QUERY GROUPS 8-15	Does the slave belong to a group among groups 8 to 15? (Each bit corresponds to agroup.)
194	Fw: Bw:	YAAA AAA1 1100 0010 Random address (high)	QUERY RANDOM ADDRESS (H)	What are the higher 8 bits of the random address?
195	Fw: Bw:	YAAA AAA1 1100 0011 Random address (middle)	QUERY RANDOM ADDRESS (M)	What are the middle 8 bits of the random address?
196	Fw: Bw:	YAAA AAA1 1100 0100 Random address (low)	QUERY RANDOM ADDRESS (L)	What are the lower 8 bits of the random address?
197	Fw: Bw:	YAAA AAA1 1100 0101 Memory location content	READ MEMORY LOCATION	What is the memory location content?
198 to 223		YAAA AAA1 110X XXXX	RESERVED	[Reserved]

Remarks Y: <0>Short address

<1>Group address/broadcast address

A: Address bit

X: Data

DTR: Data Transfer Register

‘YES’: 1111 1111

‘NO’: Without backward frame

Note1: ‘STATUS INFORMATION’ :

8-bit data indicating the status of a ballast. The meanings of the bits are as follows:

bit0 Status of control gear<0>=OK

bit1 Lamp failure(cmd146) <0>=OK

bit2 Lamp arc power on(cmd147) <0>=OFF

bit3 Query Limit Error(cmd148) <0>=YES

bit4 Fade running<0>=fade is ready <1>=fade is running

bit5 Query RESET STATE(cmd149) <0>=No

bit6 Query Missing short address(cmd150) <0>=No

bit7 Query POWER FAILURE(cmd155) <0>=No

Note2: ‘LAMP FAILURE’ :

The Lamp Failure status of IEC62386-207 time is set on the following conditions.

The following bit (0-4) of FAILURE STATUS will be determined to LampFailure in the case that has been set even 1.

bit0	short circuit	<0>=NO
bit1	open circuit	<0>=NO
bit2	load decrease	<0>=NO
bit3	load increase	<0>=NO
bit4	current protector active	<0>=NO

Note3: 'DEVICE TYPE' :

DEVICE TYPE returns the value of Table3-9.

Table3-9 Device Type

Part number	Device Type	Description
201	0	Fluorescent lamps (device type 0)
202	1	Built-in emergency lighting (device type 1)
203	2	Discharge lamps (excluding fluorescent lamps) (device type 2)
204	3	Low-voltage halogen lamps (device type 3)
205	4	The power supply voltage controller for incandescent lamps (device type 4)
206	5	Conversion to the DC voltage of the digital signal (device type 5)
207	6	LED model (device type 6)
208	7	Switching feature (device type 7)
209	8	Color control (device type 8)
210	9	Sequencer (device type 9)

If Part2xx has not been incorporated into the system, and responds as follows.

-IEC62386-102ed1.0

It is not assumed there is no extended standard on the standard.

It is a specification to return 6 by default in Applilet EZ for HCD. When not selecting 207 modes, it is not supported 207.

-IEC62386-102ed2.0

254：Part2xx is not implemented

(4) Application extended configurationcommands

Extended command of a specific set

Number	Code	Name	Description
224	YAAA AAA1 11100000	REFERENCE SYSTEM POWER	Starts power measurement (Command that exist only in IEC62386-207)
225	YAAA AAA1 11100001	ENABLE CURRENT PROTECTOR	Enables the current protection. (Command that exist only in IEC62386-207)
226	YAAA AAA1 11100010	DISABLE CURRENT PROTECTOR	Disables the current protection. (Command that exist only in IEC62386-207)
227	YAAA AAA1 11100011	SELECT DIMMING CURVE	Selects Dimming curve (Command that exist only in IEC62386-207)
228	YAAA AAA1 11100100	STORE DTR AS FAST FADE TIME	Sets the DTR of the data as Fast Fade Time.(Command that exist only in IEC62386-207)
229 to 236	YAAA AAA1 1110 XXXX	RESERVED	[Reserved]

Remarks Y: <0>Short address
<1>Group address/broadcast address
A: Address bit
H,M,L: Search address
X: Data
DTR: Data Transfer Register
'YES': 1111 1111
'NO': Without backward frame
Fw: Forward
Bw: Backward

(5) Application extended query commands

Extended command of a specific set

Number	Code		Name	Description
237	Fw: Bw:	YAAA AAA1 11101101 (GEAR TYPE	QUERY GEAR TYPE	Returns ‘GEAR TYPE’[Note1] (Command that exist only in IEC62386-207)
238	Fw: Bw:	YAAA AAA1 11101110 (Dimming curve number)	QUERY DIMMING CURVE	Returns ’Dimming curve’in use (Command that exist only in IEC62386-207)
239	Fw: Bw:	YAAA AAA1 11101111 (POSSIBLE OPERATION MODE)	QUERY POSSIBLE OPERATING MODE	Returns ‘POSSIBLEG OPERATING MODE’[Note] (Command that exist only in IEC62386-207)
240	Fw: Bw:	YAAA AAA1 11110000 (FEATURE)	QUERY FEATURES	Returns ‘FEATURES’[Note3] (Command that exist only in IEC62386-207)
241	Fw: Bw:	YAAA AAA1 11110001 (FAILURE STATUS)	QUERY FAILURE STATUS	Returns ‘FAILURE STATUS’[Note4] (Command that exist only in IEC62386-207)
242	Fw: Bw:	YAAA AAA1 11110010 ‘YES'/'NO'	QUERY SHORT CIRCUIT	Returns bit0 short circuit of ‘FAILURE STATUS’[Note4](Command that exist only in IEC62386-207)
243	Fw: Bw:	YAAA AAA1 11110011 ‘YES’/’NO’	QUERY OPEN CIRCUIT	Returns bit1 open circuit of ‘FAILURE STATUS’[Note4](Command that exist only in IEC62386-207)
244	Fw: Bw:	YAAA AAA1 11110100 ‘YES’/’NO’	QUERY LOAD DECREASE	Returns bit2 load decrease of ‘FAILURE STATUS’[Note4](Command that exist only in IEC62386-207)
245	Fw: Bw:	YAAA AAA1 11110101 ‘YES’/’NO’	QUERY LOAD INDREASE	Returns bit3 load increase of‘FAILURE STATUS’[Note4](Command that exist only in IEC62386-207)
246	Fw: Bw:	YAAA AAA1 11110110 ‘YES’/’NO’	QUERY CURRENT PROTECTOR ACTIVE	Returns bit4 current protector active of ‘FAILURE STATUS’[Note4] (Command that exist only in IEC62386-207)
247	Fw: Bw:	YAAA AAA1 11110111 ‘YES’/’NO’	QUERY THERMAL SHUTDOWN	Returns bit5 thermal shut down of ‘FAILURE STATUS’[Note4] (Command that exist only in IEC62386-207)
248	Fw: Bw:	YAAA AAA1 11111000 ‘YES’/’NO’	QUERY THERMAL OVERLOAD	Returns bit6 thermal overload with light level reduction of ‘FAILURE STATUS’[Note4] (Command that exist only in IEC62386-207)
249	Fw: Bw:	YAAA AAA1 11111001 ‘YES’/’NO’	QUERY REFARENCE RUNNING	Returns whetherReference System Power is in operation (Command that exist only in IEC62386-207)
250	Fw: Bw:	YAAA AAA1 11111010 ‘YES’/’NO’	QUERY REFERENCE MEASURMENT FAILED	Returns bit7 reference measurement failed of ‘FAILURE STATUS’[Note4] (Command that exist only in IEC62386-207)
251	Fw: Bw:	YAAA AAA1 11111011 ‘YES’/’NO’	QUERY CURRENT PROTECTOR ENABLE	Returns state of Curent protector (Command that exist only in IEC62386-207)
252	Fw: Bw:	YAAA AAA1 11111100 (OPERATION MODE)	QUERY OPERATING MODE	Returns ‘OPERATING MODE’[Note5] (Command that exist only in IEC62386-207)
253	Fw: Bw:	YAAA AAA1 11111101 (Fast fade time)	QUERY FAST FADE TIME	Returns set Fast fade time (Command that exist only in IEC62386-207)
254	Fw: Bw:	YAAA AAA1 11111110 (Minumum fast fade time)	QUERY MIN FAST FADE TIME	Returns set Minimum fast fade time (Command that exist only in IEC62386-207)
255	Fw: Bw	YAAA AAA1 1111 1111 (1 or ‘NO’)	QUERY EXTENDED VERSION NUMBER	The version number of the extended support? IEC62386-207: 1 Other: NO(no response)

Remarks Y: <0>Short address

<1>Group address/broadcast address

A: Address bit

X: Data

H,M,L: Search address

‘YES’: 1111 1111

‘NO’: Without backward frame

Fw: Forward

Bw: Backward

Note1: ‘GEAR TYPE’ :

8-bit data indicating GEAR TYPE. The meanings of the bits are as follows:

bit0	LED power supply integrate	<0>=NO
bit1	LED module integrated	<0>=NO
bit2	a.c. supply possible	<0>=NO
bit3	d.c. supply possible	<0>=NO
bit4-7	unused	

Note2: ‘POSSIBLE OPERATING MODE’ :

8-bit data indicating POSSIBLE OPERATING MODE. The meanings of the bits are as follows:

bit0	PWM mode is possible	<0>=NO
bit1	AM mode is possible	<0>=NO
bit2	output is current controlled	<0>=NO
bit3	high current pulse mode	<0>=NO
bit4-7	unused	

Note3: ‘FEATURES’ :

8-bit data indicating FEATURES. The meanings of the bits are as follows:

bit0	short circuit detection can be queried	<0>=NO
bit1	open circuit detection can be queried	<0>=NO
bit2	detection of load decrease can be queried	<0>=NO
bit3	detection of load increase can be queried	<0>=NO
bit4	current protestor is implemented and can be queried	<0>=NO
bit5	thermal shut down can be queried	<0>=NO
bit6	light level reduction due to over temperature can be queried	<0>=NO
bit7	physical selection supported	<0>=NO

Note4: ‘FAILURE STATUS’ :

8-bit data indicating FEATURES. The meanings of the bits are as follows:

bit0	short circuit	<0>=NO
bit1	open circuit	<0>=NO
bit2	load decrease	<0>=NO
bit3	load increase	<0>=NO
bit4	current protestor active	<0>=NO
bit5	thermal shut down	<0>=NO

bit6	thermal overload with light level reduction	<0>=NO
bit7	reference measurement failed	<0>=NO

Note5: ‘OPERATING MODE’ :

8-bit data indicating OPARATING MODE. The meanings of the bits are as follows:

bit0	PWM mode active	<0>=NO
bit1	AM mode active	<0>=NO
bit2	output is current controlled	<0>=NO
bit3	high current pulse mode is active	<0>=NO
bit4	non-logarithmic dimming curve active	<0>=NO
bit5-7	unused	

(6) Special commands

These commands are used to specify addresses.

Number	Code	Name	Description
256	1010 0001 0000 0000	TERMINATE	Releases the INITIALISE state.
257	1010 0011 XXXX XXXX	DATA TRANSFER REGISTER(DTR)(DTR0)	Stores the data XXXX XXXX to the DTR(DTR0). (In the parenthesis is a name in IEC62386-102ed2.0)
258	1010 0101 XXXX XXXX	INITIALISE	Sets the slave[Note 1] to the INITIALISE status for15 minutes. Commands 259 to 270 are enabled only for a slave in this status.
259	1010 0111 0000 0000	RANDOMISE	Generates a random address.
260	Fw：1010 1001 0000 0000 Bw：‘YES’/’NO’	COMPARE	Is the random address smaller or equal to the search address?
261	1010 1011 0000 0000	WITHDRAW	Excludes slaves for which the random address and search address match from the Compare process.
262	1010 1101 0000 0000	RESERVED	[Reserved]
263	1010 1111 0000 0000	PING	Ignores in the slave. (Command that exist only in IEC62386-102ed2.0)
264	1011 0001 HHHH HHHH	SEARCHADDRH	Specifies the higher 8 bits of the search address.
265	1011 0011 MMMM MMMM	SEARCHADDRM	Specifies the middle 8 bits of the search address.
266	1011 0101 LLLL LLLL	SEARCHADDRL	Specifies the lower 8 bits of the search address.
267	1011 0111 0AAA AAA1	PROGRAM SHORT ADDRESS	The slave[Note 2] shall store the received 6-bit address (AAA AAA) as a short address if it is selected.
268	Fw：1011 1001 0AAA AAA1 Bw：‘YES’/’NO’	VERIFY SHORT ADDRESS	Is the short address AAA AAA?
269	Fw：1011 1011 0000 0000 Bw：0AAA AAA1	QUERY SHORT ADDRESS	What is the short address of the slave[Note 2]being selected?
270	1011 1101 0000 0000	PHYSICAL SELECTION	Sets the slave to Physical Selection Mode and excludes the slave from the Compare process. (Excluding IEC62386-102ed2.0) (Command that exist only in IEC62386-102ed1.0, -207ed1.0)
271	1011 1111 XXXX XXXX	RESERVED	[Reserved]

Note1: Target slave specification of INITIALISE (XXXX XXXX)

0000 0000:	All slaves is target
0AAA AAA1:	Address AAAAAA is target
1111 1111:	A slave without Short Address is target

Note2: It is a slave with a random address same as a search address or a slave of Physical Selection Mode.

Remarks Y: <0>Short address
<1>Group address/broadcast address
A: Address bit
X: Data
H,M,L: Search address
‘YES’: 1111 1111
‘NO’: Without backward frame
Fw: Forward
Bw: Backward

(7) Extending special commands

These commands are used for feature expansion.

Number	Code	Name	Description
272	1100 0001 XXXX XXXX	ENABLE DEBICE TYPE X	Adds the device XXXX (a special device).
273	1100 0011 XXXX XXXX	DATA TRANSFER REGISTER1 (DTR1)	Stores data XXXX into DTR1.
274	1100 0101 XXXX XXXX	DATA TRANSFER REGISTER2 (DTR2)	Stores data XXXX into DTR2.
275	1100 0111 XXXX XXXX	WRITE MEMORY LOCATION	Write data into the specified address of the specified memory bank. (There is BW) (DTR(DTR0)：address, DTR1：memory bank number)
276	1100 1001 XXXX XXXX	WRITE MEMORY LOCATION – NO REPLY	Write data into the specified address of the specified memory bank. (There is no BW) (DTR(DTR0)：address, TR1：memory bank number) (Command that exist only in IEC62386-102ed2.0)
276 to 349		RESERVED	[Reserved]

3.5 Functions for Applilet EZ for HCD DALI Communication (RL78/I1A DC/DC LED Control Evaluation Board)

This section provides a list of functions used for DALI communication, including their I/O and features, and is intended to support the understanding of the program for DALI communication.

List of functions

3.5.7 r_dali_hw.c

3.5.7.1 DALI_InitHW

3.5.7.2 DALI_GetCommand

3.5.7.3 DALI_SendAnswer

3.5.7.4 DALI_ProhibitReception

3.5.7.5 DALI_PermitReception

3.5.7.6 DALI_CheckProhibit

3.5.7.7 DALI_CHeckInterfaceDown

3.5.1 r_dali.c

3.5.1.1 DALI_init

Format	void DALI_init(void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	Initializes DALI functions		

3.5.1.2 DALI_getValue

Format	uint8_t DALI_getValue(uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	8 bits (unsigned char)	Specifies the channel (1 to 3) for acquiring the lighting control level
Return value		Data type	Overview
		unsigned char	LED lighting control level
Feature	Returns the LED lighting control level. The LED lighting control level is the value already set by the DALI_ReceiveCommand(). In addition, save processing of the configuration data is performed when all of the following conditions are met: the configuration data save flag is on, there is no DALI command response waiting to be returned, and there is no active random address allocation processing.		

3.5.1.3 DALI_ActualLevelChangeCheck

Format	void DALI_ActualLevelChangeCheck(uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	8 bits (unsigned char)	Specifies the channel (1 to 3) for acquiring the lighting control level
Return value		Data type	Overview
		void	–
Feature	Confirms the changes to ActualLevel for saving ActualLevel when there was no change for acertain period of time.		

3.5.1.4 DALI_RevceiveCommand

Format	void DALI_ReceiveCommand		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	Receives the DALI command and calls the analysis processing		

3.5.1.5 DALI_Fading

Format	void DALI_Fading (uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	8 bits (unsigned char)	Specifies the channel (1 to 3)
Return value		Data type	Overview
		void	–
Feature	Performs the fade processing for the specified channel		

3.5.1.6 DALI_UpdateVariables

Format	void DALI_UpdateVariable (uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	8 bit (unsigned char)	Specifies the channel (1 to 3)
Return value		Data type	Overview
		void	–
Feature	Sets a save flag for the configuration data of the specified channel		

3.5.1.7 DALI_SetSystemFailure

Format	void DALI_SetSystemFailure (uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	8 bit (unsigned char)	Specifies the channel (1 to 3)
Return value		Data type	Overview
		void	–
Feature	Sets the specified channel to SystemFailure status		

3.5.1.8 DALI_ResetValue

Format	void DALI_ResetValue (void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	Sets the configuration value to the reset value		

3.5.1.9 DALI_CheckReset

Format	void DALI_CheckReset (void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	Checks whether the configuration value is the reset value and updates RESET STATUS.		

3.5.1.10 DALI_RandmInit

Format	void DALI_RandmInit (void)		
Parameter			
Parameter	I/O	Data type	Overview
–	–	–	
Return value		Data type	Overview
		void	–
Feature	Generates a random seed value.		

3.5.2 r_dali_analyze.c

3.5.2.1 DALI_CheckConfigCommand

Format	uint8_t DALI_CheckConfigCommand(uint8_t command)		
Parameter			
Parameter	I/O	Data type	Overview
command	I	8 bits	Command
Return value		Data type	Overview
		unsigned char	TRUE: Received a duplex transmission command FALSE: Unavailable command
Feature	Confirms that the DALI configuration command (command numbers 32 to 129) is transmitted twice within 100 ms. TRUE is returned if, when the command is received, it is the second configuration command received and it was received within 100 ms. FALSE is returned if thisdoes not occur. A timer is set if it is the first reception of the configuration command.		

3.5.2.2 DALI_Check2ndCommand

Format	uint8_t DALI_Check2ndCommand(uint8_t command)		
Parameter			
Parameter	I/O	Data type	Overview
command	I	8 bits	Command
Return value		Data type	Overview
		unsigned char	TRUE: Received a duplex transmission command FALSE: Unavailable command
Feature	Confirms that the DALI configuration command (command numbers 32 to 129) is transmitted twice within 100 ms. TRUE is returned if, when the command is received, it is the second configuration command received and it was received within 100 ms. FALSE is returned if this does not occur. A timer is set if it is the first reception of the configuration command.		

3.5.2.3 DALI_AnalyzeCommand

Format	void DALI_AnalyzeCommand(uint16_t command)		
Parameter			
Parameter	I/O	Data type	Overview
command	I	16bit	Command
Return value		Data type	Overview
		Void	–
Feature	Analyzes the command and isolates the processing		

3.5.2.4 DALI_CheckAddress

Format	uint8_t DALI_CheckAddress(uint8_t address)		
Parameter			
Parameter	I/O	Data type	Overview
address	I	8 bits	Address value within the DALI command
Return value		Data type	Overview
		unsigned char	TRUE: Completed successfully FALSE: Unavailable
Feature	Determines the type (BROADCAST, SHORT ADDRESS, or GROUP) from the 8-bit format of the address,determines whether the address is a target of the processing, and then returns TRUE or FALSE		

3.5.3 r_dali_command.c

3.5.3.1 DALI_LightingCommand

Format	void DALI_LightingCommand(uint8_T cmd)		
Parameter			
Parameter	I/O	Data type	Overview
cmd	I	8 bits	Command
Return value		Data type	Overview
		–	–
Feature	Performs lighting-control-related command processing		

3.5.3.2 DALI_QueryCommand

Format	void DALI_QueryCommand(uint8_t cmd)		
Parameter			
Parameter	I/O	Data type	Overview
cmd	I	8 bits	Command
Return value		Data type	Overview
		–	–
Feature	Performs query-related command processing		

3.5.3.3 DALI_ConfigCommand

Format	void DALI_ConfigCommand(uint8_t cmd)		
Parameter			
Parameter	I/O	Data type	Overview
cmd	I	8 bits	Command
Return value		Data type	Overview
		–	–
Feature	Performs configuration-related command processing		

3.5.3.4 DALI_Fade200ms

Format	void DALI_Fade200ms(uint8_t fade_rate, uint8_t fade_direction)		
Parameter			
Parameter	**I/O**	Data type	Overview
fade_rate	I	unsigned char	Fade rate
fade_direction	I	unsigned char	Fade direction
Return value		Data type	Overview
		–	–
Feature	Performs 200 ms fade processing		

3.5.3.5 DALI_SetArcPowerWithFade

Format	void DALI_SetArcPowerWithFade(uint8_t level_new)		
Parameter			
Parameter	**I/O**	Data type	Overview
level_new	I	unsigned char	Target lighting control level
Return value		Data type	Overview
		–	–
Feature	Performs the fade processing by using the target lighting control level specified by the parameter		

3.5.3.6 DALI_SpecialCommand

Format	void DALI_SpecialCommand(uint8_t cmd, uint8_t data)		
Parameter			
Parameter	**I/O**	Data type	Overview
cmd	I	unsigned char	Command
data	I	unsigned char	Data
Return value		Data type	Overview
		–	–
Feature	Performs special command processing		

3.5.4 r_dali_memorybank.c

3.5.4.1 DALI_InitMemorybank

Format	void DALI_InitMemorybank		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	Initializes the memory bank variable		

3.5.4.2 DALI_WriteMemorybank

Format	uint8_t DALI_WriteMemorybank(uint8_t bank,uint8_t,address,uint8_t data)		
Parameter			
Parameter	I/O	Data type	Overview
bank	I	unsigned char	Bank number
address	I	unsigned char	Address in the bank
data	I	unsigned char	Data to be written
Return value		Data type	Overview
		unsigned char	TRUE: The data was writeen successfully. FALSE: The data was not writeen.
Feature	Writes the data to the specified location in the memory bank		

3.5.4.3 DALI_EnableMemorybank

Format	void DALI_EnableMemorybank(uint8_t enable)		
Parameter			
Parameter	I/O	Data type	Overview
enable	I	unsigned char	TRUE: memory bank write enable FALSE: memory bank write disabled
Return value		Data type	Overview
		void	
Feature	Sets writing of the memory bank enabled or disabled.		

3.5.4.4 DALI_ReadMemorybank

Format	uint8_t DALI_ReadMemorybank(uint8_t bank,uint8_t,address,uint8_t* data)		
Parameter			
Parameter	I/O	Data type	Overview
bank	I	unsigned char	Bank number
address	I	unsigned char	Address in the bank
data	O	unsigned char	Read data
Return value		Data type	Overview
		void	TRUE: The data was read successfully. FALSE: The data was not read.
Feature	Reads data from the specified location in the memory bank		

3.5.4.5 DALI_CheckMemorybankSaving

Format	Uint8_t DALI_CheckMemorybankSaving(uint8_t ch)		
Parameter			
Parameter	I/O	Data type	Overview
ch	I	unsigned char	Channel number
Return value		Data type	Overview
		unsigned char	TRUE: Completed successfully
Feature	When the information of the memory bank of the specified channel is updated, writes memory bank data to data flash memory.		

3.5.4.6 DALI_GetChecksum

Format	uint8_t DALI_GetChecksum(uint8_t *membank)		
Parameter			
Parameter	I/O	Data type	Overview
bank	I	unsigned char *	Checksum calculation memory bank head pointer
Return value		Data type	Overview
		unsignedchat	Checksum value
Feature	Calculates the checksum of the specified memory bank and returns the checksum value.		

3.5.5 r_dali_timer.c

3.5.5.1 DALI_InitTimer

Format	void DALI_InitTimer(void)		
Parameter			
Parameter	I/O	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	Initialize the timer variable used with the DALI feature.		

3.5.5.2 DALI_Interval

Format	void DALI_Interval(void)		
Parameter			
Parameter	I/O	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		void	–
Feature	This command is called every 1 ms of the timer. Performs timer count processing		

3.5.5.3 DALI_StartTimer

Format	void DALI_StartTimer(uint8_t type)		
Parameter			
Parameter	I/O	Data type	Overview
type	I		Type of timer counter to be used
Return value		Data type	Overview
			–
Feature	Starts counting of the timer counter by specified type		

3.5.5.4 DALI_StopTimer

Format	void DALI_StopTimer(uint8_t type)		
Parameter			
Parameter	I/O	Data type	Overview
type	I	8 bits	The type of the timer counter to be used
Return value		Data type	Overview
		void	–
Feature	Stops counting of the timer counter by specified type		

3.5.5.5 DALI_IsTimerRunning

Format	uint8_t DALI_IsTimerRunning(uint8_t type)		
Parameter			
Parameter	**I/O**	Data type	Overview
type	I	8 bits	Type of timer counter to be used
Return value		Data type	Overview
		void	TRUE: Running FALSE: Stopped
Feature	Returns whether the timer counter of the specified type is running		

3.5.5.6 DALI_StartFadeTimer

Format	void DALI_StartFadeTimer(uint32_t fadestep, uint16_t fadetime)		
Parameter			
Parameter	**I/O**	Data type	Overview
fadestep	I	unsigned int	Fade step
fadetime	I	unsigned short	Fade time
Return value		Data type	Overview
		–	–
Feature	Starts counting of the fade timer count		

3.5.5.7 DALI_StopFadeTimer

Format	void DALI_StopFadeTimer(uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	unsigned char	Channel number
Return value		Data type	Overview
		–	–
Feature	Stops counting of the fade timer count		

3.5.5.8 DALI_IsFading

Format	uint8_t DALI_IsFading(uint8_t channel)		
Parameter			
Parameter	**I/O**	Data type	Overview
channel	I	unsigned char	Channel number
Return value		Data type	Overview
		unsigned char	TRUE: Running FALSE: Stopped
Feature	Returns whether the fade timer count is running or stopped		

3.5.5.9 DALI_GetRandomValue

Format	uint16_t DALI_GetRandomValue(uint16_t size)		
Parameter			
Parameter	**I/O**	Data type	Overview
size	I	unsigned short	Return value range
Return value		Data type	Overview
		unsigned short	Random value
Feature	Generates a random value based on the timer variable		

3.5.6 r_dali_variable.c

3.5.6.1 DALI_InitEmulation

Format	unsigned char DALI_InitEmuration(void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		unsigned char	Status
Feature	Initializes the EEPROM emulation		

3.5.6.2 DALI_ReadVariables

Format	uint8_t DALI_ReadVariables(uint8_t DataNumber)		
Parameter			
Parameter	**I/O**	Data type	Overview
DataNumber	I	unsigned char	–
Return value		Data type	Overview
		unsigned char	Status
Feature	Reads the configuration value from the data flash memory		

3.5.6.3 DALI_SaveVariables

Format	uint8_t DALI_SaveVariables(uint8_t DataNumber)		
Parameter			
Parameter	**I/O**	Data type	Overview
DataNumber	I	unsigned char	–
Return value		Data type	Overview
		unsigned char	Status
Feature	Writes the configuration value to the data flash memory		

3.5.6.4 DALI_SetEELMode

Format	void DALI_SetEELMode(uint8_t mode)		
Parameter			
Parameter	**I/O**	Data type	Overview
mode	I	unsigned char	EEL_MODE_ENFORCED: execution completion wait EEL_MODE_POLLING: run immediate return
Return value		Data type	Overview
		void	–
Feature	Sets the mode of operation of the EEL.		

3.5.6.5 DALI_EELPolling

Format	uint8_t DALI_EELPolling(void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	void	–
Return value		Data type	Overview
		unsigned char	STATUS_OK: Normal end. STATUS_NG: Abnormal end
Feature	Performs continuation run processing of EEL. For the data flash memory, when the pool is full,performs clean-up processing, and when there is a mismatch in the block,performs format processing.		

3.5.7 R_dali_hw.c

3.5.7.1 DALI_InitHW

Format	void DALI_InitHW (void)		
Parameter			
Parameter	**I/O**	Data type	Overview
void	I	–	–
Return value		Data type	Overview
		–	–
Feature	Sets DALI communication.		

3.5.7.2 DALI_GetCommand

Format	uint8_t DALI_GetCommand(uint16_t* received_data)		
Parameter			
Parameter	**I/O**	Data type	Overview
received_data	O	unsigned short*	Variable pointer for the storage of reception data
Return value		Data type	Overview
		unsigned char	TRUE: Normal reception FALSE: No reception, or Reception error
Feature	Performs confirmation of data reception and getting the received data.		

3.5.7.3 DALI_SendAnswer

Format	void DALI_SendAnswer (uint8_t answer)		
Parameter			
Parameter	**I/O**	Data type	Overview
answer	I	unsigned char	Answer data
Return value		Data type	Overview
		void	–
Feature	Performs the transmission of the response data.		

3.5.7.4 DALI_ProhibitReception

Format	void DALI_ProhibitReception (uint16_t received_data)		
Parameter			
Parameter	**I/O**	Data type	Overview
Received_data	I	Unsigned char	The received command
Return value		Data type	Overview
		–	–
Feature	When the received command is a one that may return Backward frame, it makes the reception rejection state until Backward frame is transmitted. When the address specified by the command isn't for itself, it makes the subject of processing.		

3.5.7.5 DALI_PermitReception

Format	void DALI_PermitReception (void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		–	–
Feature	Releases the reception rejection state.		

3.5.7.6 DALI_CheckProhibit

Format	uint8_t DALI_CheckProhibit (void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		unsigned char	TRUE: Normal reception FALSE: No reception, or Reception error
Feature	Detects a fall edge at the time of the DALI reply data transmission, and 10 ms extends DALI data reception prohibition time.		

3.5.7.7 DALI_CheckInterfaceDown

Format	uint8_t DALI_CheckInterfaceDown (void)		
Parameter			
Parameter	**I/O**	Data type	Overview
–	–	–	–
Return value		Data type	Overview
		unsigned char	TRUE：Communication line HI FALSE：Communication line LOW
Feature	Confirms the state of the reception port and, in the case of HI, returns TRUE, in the case of LOW, returns FALSE.		

4. DMX512 Communication

This section introduces the DMX512 communication protocol and presents a solution for reception that can be implemented using the RL78/I1A microcontroller. An external RS-485 compatible transceiver is required to complete the application schematic.

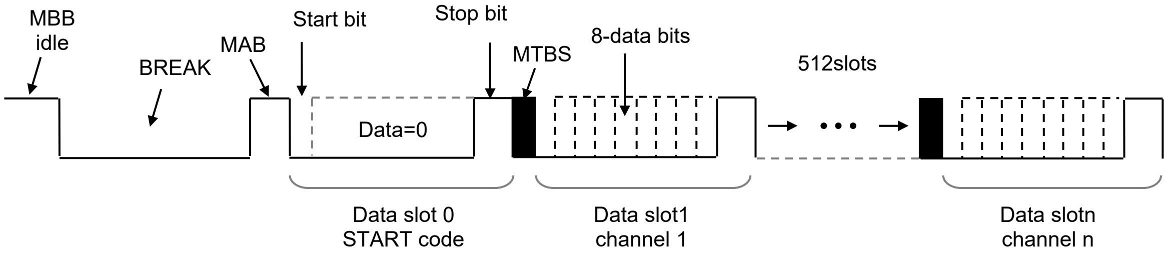
4.1 DMX512 Lighting Communication Protocol

4.1.1 Overview of the DMX512 standard

DMX512 is a wired communication protocol for digital data transmission used extensively inindustrial lighting applications such as theatre stage lighting and exhibition lighting (devices include dimmers, scanners, moving lights, strobes…). A DMX512 system has only one transmitter (also known as master or host) and multiple receivers.
The DMX512 standard covers electrical characteristics (based on the EIA/TIA–485 standard), data format, data protocol and connector type. This standard is intended to provide for interoperability aboth communication and mechanical levels with controllers made by different manufacturers.DMX512 naming comes from Digital MultipleX with 512 data slots.

Data is transmitted at a 250 kbit/s rate (each bit is 4 μs long) using a physical interface compatible with the RS-485 transmission standard over a physical interface consisting of 3 wires, with a data signal constructed using 2 differential lines and a ground (0 V).
The DMX512 data slots are transmitted sequentially in an asynchronous serial format, beginning with data slot 1 and ending with the last data slot 512.
Before the first data slot is transmitted, a starting sequence (RESET sequence) is necessary, consisting of a BREAK, a MARK AFTER BREAK and a START code (1 byte). Therefore a total of513 slots are transmitted when including the START code. Valid DMX512 data slot values rangefrom 0 to 255.

Figure4-1 DMX512 Receiver Timing Chart



	MIN.	TYP.	MAX.	Unit
BREAK	88	176	–	μs
MAB	8	–	1 000 000	μs
Slot width	–	44	–	μs
Start/data/stop bits	3.92	4	4.08	μs
MTBS	0	–	1 000 000	μs
MBB	0	–	1 000 000	μs

The data packet consists of the following time slots:

- **BREAK**

  It indicates the start of a new packet, typical value is 176 µs.

- **MARK AFTER BREAK (MAB)**

  It separates the BREAK and START code time slots. Values can be between 8 µs and 1second.

- **START CODE**

  The first slot (slot 0) following a MARK AFTER BREAK.

  It identifies the function of the subsequent data bytes in the packet. For lighting control commands, the START code value is 0x00, therefore it is also known as a NULL START code.Its purpose is to tell all the DMX devices to expect a brightness level to be sent.

- **DATA SLOTS**

  Subsequent data bytes are where the lighting control levels for each receiving device areplaced.

  There shall be no minimum number of data slots on the data link. DMX512 data packets with fewer than 512 slots may be transmitted, subject to the minimum timing requirements.Receivers operate correctly when receiving packets with break to break spacing of at least the minimum value of 1196 µs up to the maximum value of 1.25 s.

  Time between any two data slots may vary from 0 up to 1 second. This time is known asMARK TIME BETWEEN SLOTS (MTBS).

  Each data slot consists of 1 start bit, 8 data bits and 2 stop bits. A start bit is always alogical 0 and a stop bit is a logical 1.

- **MARK BEFORE BREAK (MBB)**

  This is the time between the second stop bit of the last data slot of a given data packet and the falling edge of the BREAK of the next data packet. This time may vary from 0 up to 1 second.Every data packet transmitted over the data link starts with BREAK, MARK AFTER BREAK, and START code sequence defined as RESET sequence.

  The DMX512 protocol requires the transmitter to continuously repeat (at least once a second) the transmission of a packet.

4.1.2 Hardware control interface

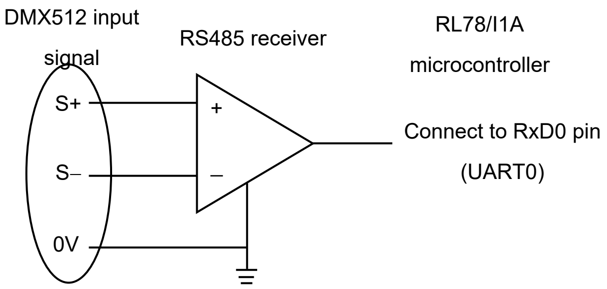
When ordering to implement DMX512 communication with the RL78/I1A microcontroller, the hardware control interface consists only of a RS-485 transceiver connected to the RxD0 reception pin of the UART0 serial interface.

As previously mentioned, the RS-485 standard uses three wires to transmit bits:

- The + signal wire (S+)
- The − signal wire (S−)
- The ground wire (0 V)

Figure4-2 shows the DMX512 receiver hardware interface.

Figure4-2 DMX512 Receiver Hardware Interface



The DMX512 protocol is half-duplex, that means data for one DMX device is either being receivedor being transmitted, but never simultaneously. The S+/S− signals are differential signals that are180° out of phase, a logical 0 is recognized when S− > S+ for 4 µs and a logical 1 when S+ > S−for 4 µs. Transmitting on the signal lines requires to physically toggle the S+/S− between +6 V and0 V. A receiver must support voltage levels between +12 V and -7 V.

Typical DMX512 systems are based on the following principle:

(1) The multiple receivers are connected to the DMX transmitter in a daisy-chain way and every packet goes through every receiver in its entirety.

(2) Each receiver receives the differential signal via an RS-485 transceiver. In our case, the RL78/I1A microcontroller then receives the packets through the RxD0 pin.

(3) Each receiver is programmed with a specific DMX address (from 1 to 512) so that it knowswhich particular frame it has to extract from each packet. It has to count the number of bytes being transmitted by the DMX controller and only capture the byte corresponding to its address.

(4) The receiver interprets the data and performs the appropriate lighting control action, in our case the RL78/I1A modifies the duty cycle of the PWM output according to the data received.

4.2 RL78/I1A Features for DMX512 Communication

4.2.1 Peripheral functions

The process of receiving DMX512 packets can be broken down into three different parts.

- The synchronization of the receiver with the beginning of a new data packet identified by aprolonged BREAK signal.
- Once the BREAK signal is identified and acknowledged, the receiver needs to wait for the lineto return to the idle state (MARK AFTER BREAK) and for a first byte of data to arrive.
- The last part consists of a loop where the receiver captures up to 512 bytes of data and stores them sequentially in a reception buffer.

The RL78/I1A microcontroller is able to perform the above actions and check whether signals received in DMX512 communication match the DMX512 communication format using specific peripheral functions:

- TAU channel 7: BREAK Signal Detection and BREAK Signal Length Measurement
- TAU channel 0: MARK AFTER BREAK and MARK TIME BETWEEN SLOTS Measurement
- UART0 Interface: START code and data slot Reception

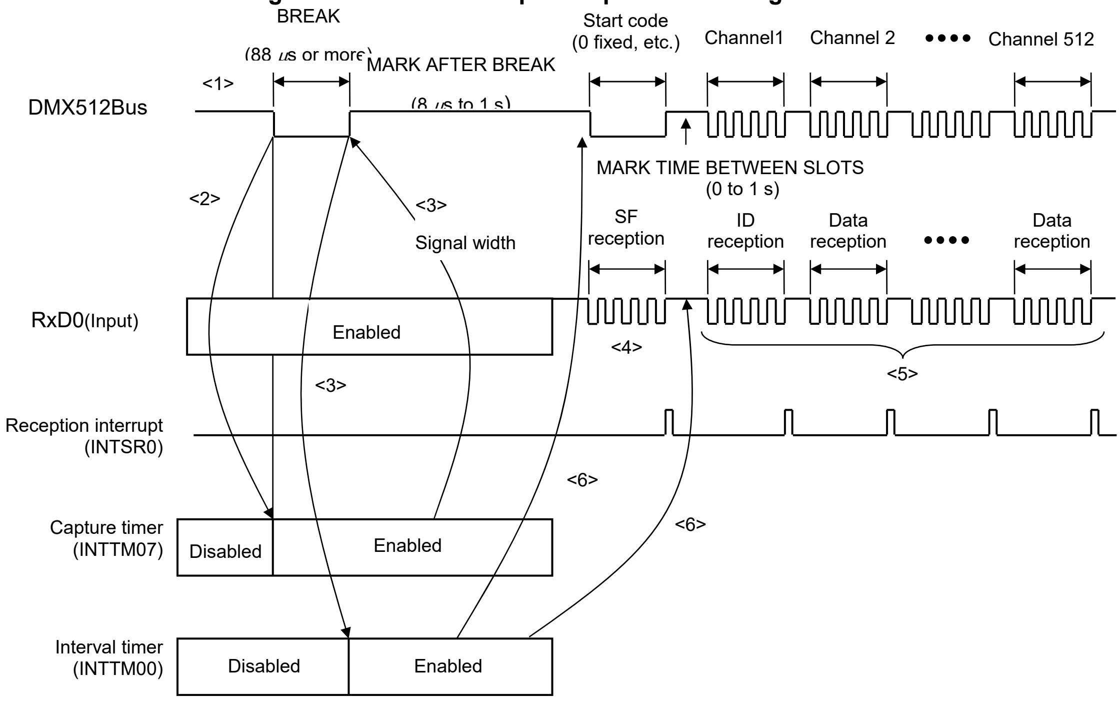
4.2.2 Operation overview

The method to perform DMX communication using the RL78/I1A peripheral functions described above is explained here in greater details. Timing charts are also presented on the next page toclarify the process.

- **TAU channel 7: BREAK Signal Detection and BREAK Signal Length Measurement**
  The input signal low-level width measurement function of the Timer Array Unit channel 7 canbe used to detect the falling edge of a BREAK period and then to measure and acknowledgethe BREAK signal width (low level for at least 88 μs). The timer needs to start counting at the falling edge of the signal input on the serial data input pin (RxD0) of UART0 and then the countvalue of the timer is captured at the rising edge. In this way, a low-level width can be measured.If the low-level width is greater than a specific value, it is recognized as a BREAK signal.
- **TAU channel 0: MARK AFTER BREAK and MARK TIME BETWEEN SLOTS Measurement**
  The interval timer mode of the Timer Array Unit channel 0 can then be used to measure and acknowledge the MARK AFTER BREAK signal width (high level for 8 μs to 1 s). This timer is also used to measure the MARK TIME BETWEEN SLOTS and ensure this time is less than 1s.
- **UART0 Interface: START code and data slot Reception**
  After the BREAK signal is detected, UART0 interface is used to receive the START code and the data slots. The RL78/I1A MCU waits for the signal reception on RxD0 pin. Then serial datais sequentially stored in the RXD0 register (= bits 7 to 0 of serial data register 01 (SDR01)) atthe specified baud rate. When the stop bit is detected, the reception end interrupt request (INTSR0) is generated.

The timing charts below show the appropriate sequence of settings to implement in order to enable DMX512 slot detection.

Figure4-3 DMX512 Reception Operation Timing Chart



(1) Waiting state (idle)

INTTM07 interrupt masked, INTSR0 interrupt unmasked, 1 s interval timer stopped

(2) Start of BREAK signal

INSR0 interrupt with error → INTTM07 unmasked

(3) End of BREAK signal

INTTM07 interrupt: BREAK signal length measurement

If length < 80 μs → state 1

If length ≥ 80 μs → start 1 s interval timer

(4) START code reception

INSR0 reception interrupt: check data

If reception error → state 1

If no reception error → restart the 1 s interval timer

(5) Data slot reception

INSR0 reception interrupt: check data

If reception error → state 1

If no reception error → restart the 1 s interval timer

* repeat state 5 for each data slot

(6) MARK AFTER BREAK and MARK TIME BETWEEN SLOTS checking

1 s interval timer interrupt (when timer exceeds 1 s) → state 1

4.3 DMX Lighting Control Software Description

4.3.1 Initialization of the internal peripheral functions

The initialization of the peripheral used for DMX512 reception operation includes the following settings:

- Setting CPU clock frequency to 32 MHz using PLL (16 times the internal high-speed oscillation clock fIH x ½)
- Peripheral function clock supply setting
- I/O port setting
- UART0 interface setting
  - Setting operation clock to f_{CLK} (32 MHz)
  - Setting Serial Array Unit 0 ch0 to UART transmission mode
  - Setting Serial Array Unit 0 ch1 to UART reception mode (Falling edge = start bit)
  - Switching input of TAU channel 7 to input signal of the RXD0 pin (ISC register)
- 16-bit TAU channel 7 setting
  - Setting count clock to f_{CLK} (32 MHz)
  - Setting to capture & one-count mode (Counting up) with both the edges of the TI0n pininput used as a start trigger and a capture trigger
  - Setting to low-level width measurement with the following condition:
    - Start trigger: falling edge, Capture trigger: rising edge
  - Unmasking the interrupt INTTM07 and starting TAU channel 7
- 16-bit TAU channel 0 setting
  - Masking the interrupt INTTM00 (previously set in 1 s interval timer mode)

Below is the DMX512 reception peripheral initialization extracted from the sample program ("DMX_init()" function from the "r_dmx.c" file).

```
SPS0     = 0x0000;                      /* CK00 select 32 MHz          */
SMR00    = 0x0022;                      /* Unit0 ch.0 UART mode        */
SMR01    = 0x0122;                      /* Unit0 ch.1 UART mode        */
SCR01    = 0x4097;                      /* 1 stop bit                  */
SDR01    = 0x7E00;                      /* 250 Kbps                    */
SIR01    = 0x0007;                      /* error clear                 */
NFEN0    = 0x01;                        /* noise filter on             */
PM1.1    = 1;                           /* P1.1 UART RX mode           */
PIM1.1   = 1;
ISC      = 0x02;                        /* RXD0 = TAU                  */
TPS0     = TPS0 | 0x0000;               /* 32 MHz                      */
TMR07    = 0x828C;                      /* timer7 mode set             */
NFEN1    = 0x80;
TMIF00   = 0;                           /* interrupt flg clear         */
TMMK00   = 1;                           /* INTTM00 disable             */
TMIF07   = 0;                           /* interrupt flg clear         */
TMMK07   = 1;                           /* INTTM07 disable             */
SRIF0    = 0;                           /* interrupt flg clear         */
SRMK0    = 0;                           /* INTSR0 enable               */
TS0      = 0x0080;                      /* timer ch.7 start            */
SS0      = 0x0002;                      /* uart0 ch.1 start            */
```

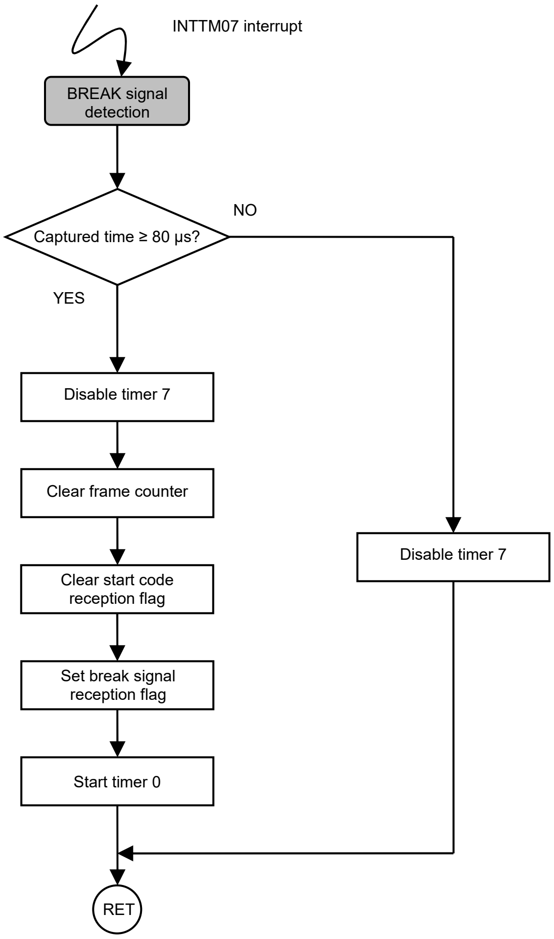
4.3.2 Operation description & software flow charts

This section presents the flow charts of the lighting control demonstration code based on the DMX512 communication interface. In this example, the 3 LED channels are controlled using independent addresses.

The first step of the process is the BREAK signal detection using Timer Array Unit channel 7, the timer starts counting when a falling edge on the RxD0 reception pin is detected, and then generates an interrupt at the next signal rising edge. The captured timer value is used to measure the BREAK signal length, if the captured time is greater than 80 µs, the BREAK signal is acknowledged and the BREAK signal reception flag is set. Timer Array Unit 0 is also started tocheck the MARK AFTER BREAK time.

The flow chart featured below gives a detailed description of this process.

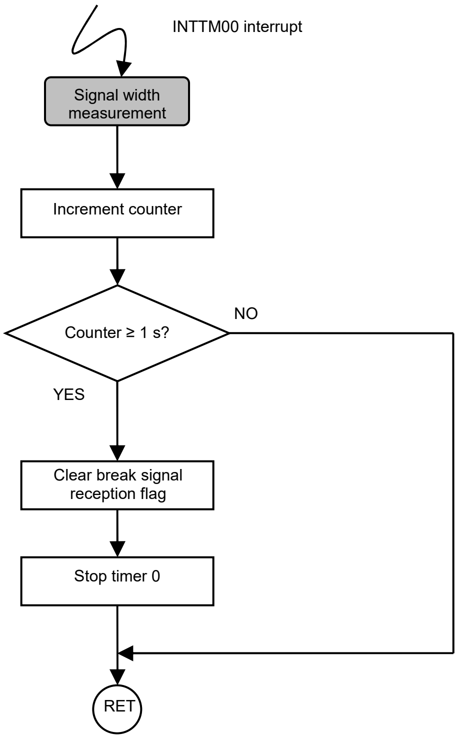
Figure4-4 BREAK Signal Detection Flow Chart



Timer Array Unit channel 0 is used to ensure that MARK AFTER BREAK time and MARK TIMEBETWEEN SLOTS are always under 1 s. If the interval timer exceeds this value, the BREAK signal reception flag is cleared.

The flow chart featured below gives a detailed description of this process.

Figure4-5 MARK AFTER BREAK & MARK TIME BETWEEN SLOTS Measurement Flow Chart

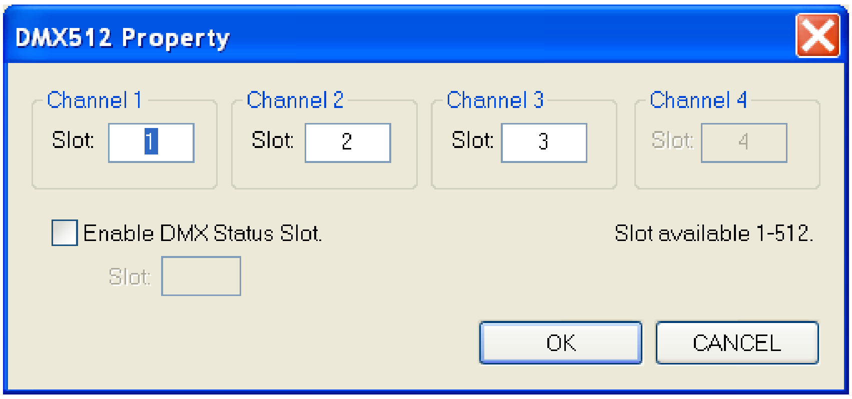


The INTSR0 interrupt service routine checks that the correct START code (0x00) is received and then stores the lighting control levels for the 3 LED channels by only capturing the data slots (1 slot= 1 byte) corresponding to the LED channel addresses. By default, the slots allocated to the LED channels are defined as follows:

Channel 1: slot = 1, channel 2: slot = 2 and channel 3: slot = 3.

These values can be modified in the Applilet EZ for HCD Controller main window, when the dimmer program is set to DMX512 it is possible to customize the slot address allocated to each ofthe 3 LED channels. To do so, open the "DMX512 Property" dialogue box by selecting the "Project" menu and clicking on "DMX512…" or simply by clicking on the "Setting" button.

Figure4-6 Applilet EZ for HCD Controller DMX512 Property Dialogue Box



DMX512 Property

Channel 1
Slot: 1

Channel 2
Slot: 2

Channel 3
Slot: 3

Channel 4
Slot: 4

□ Enable DMX Status Slot.
Slot:

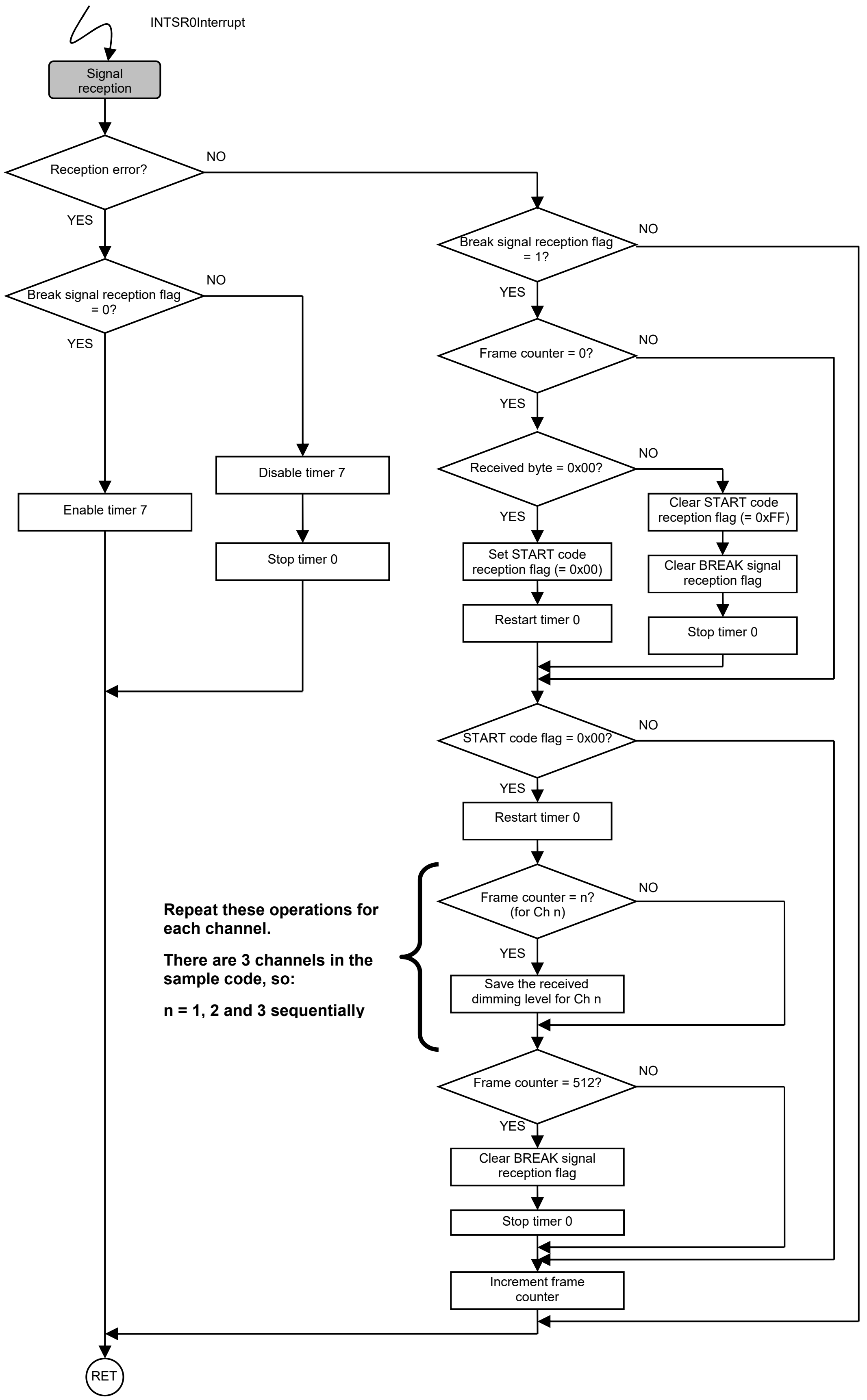
Slot available 1-512.

When changing the allocated slots, the corresponding #define macros in the "r_user.h" file of the sample code project is modified accordingly:

```
#define DMX_CHANNEL_LED1            1
#define DMX_CHANNEL_LED2            2
#define DMX_CHANNEL_LED3            3
```

After the valid RESET sequence (BREAK signal, MARK AFTER BREAK and START code) and the DMX512 packet time are detected, the received packet is processed. A particular data slot is selected from the packet received according to the LED channel address programmed, and the function "DMX_getValue()" is called to update the target lighting control value of each LED channel.The functions "LEDn_set()" (n = 1, 2, 3) then apply the updated LED lighting control values. Theduty cycle of the PWM is varied such that a "255" value means 100% duty cycle and "0" value means 0% duty cycle.

Figure4-7 START Code and Data Slot Reception Flow Chart



Repeat these operations for each channel.

There are 3 channels in the sample code, so:

n = 1, 2 and 3 sequentially

5. IR Communication

This section introduces the NEC IR communication protocol and presents a solution for reception that can be implemented using the RL78/I1A microcontroller, only an IR transceiver is required to enable a connection of the RL78/I1A to an IR transmitter.

5.1 IR Communication Protocol

5.1.1 Overview of the NEC IR protocol

NEC Infrared remote control uses an infrared ray with wavelength of approximately 950 nm to transmit several bytes of information at low speed. Although infrared rays are used to transmitbinary (0/1) data, this is not simply a matter of representing binary values by the ON/OFF status of infrared rays.

The NEC format is an example of IR transmission protocol widely used in the worldwide industry and is described below.

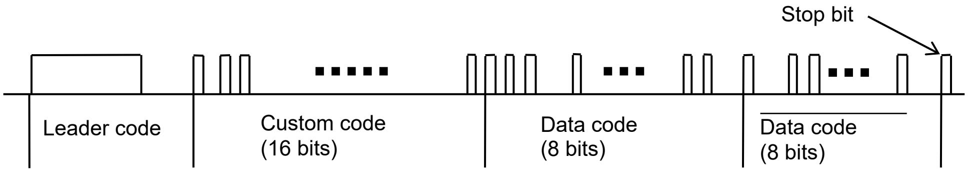
General Format

The infrared remote control signal starts with a leader code.After the leader code, the frame includes a 16-bit custom code (also called address), then an 8-bit data code (also called command) and an inverted binary 8-bit code, and finally a stop bit to signify the end of a message transmission.

An example of the NEC infrared remote control format is shown below.

This signal is followed by a frame space during which no infrared rays are emitted. The total frame length (including everything from the leader code to the frame space) is 108 ms.

Figure5-1 Example of NEC Format for Infrared Remote Control



Leader Code

The leader code stays ON for a 9 ms period, then becomes OFF for a 4.5 ms period. Since the timing of this part of the waveform differs greatly from the following data code section, it makes the leader code easier to recognize.

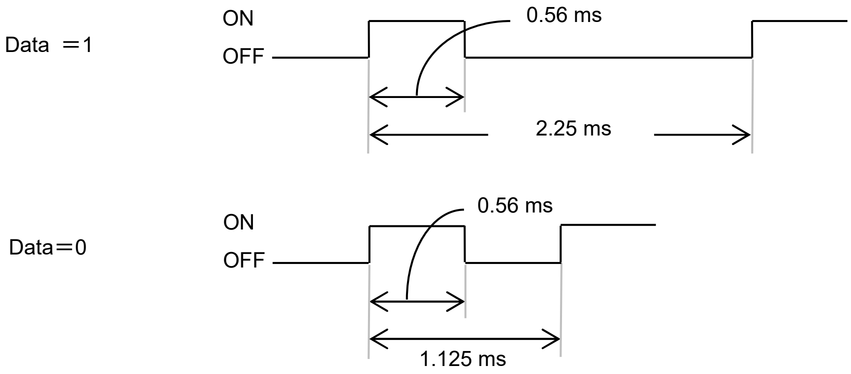
When repeating, the OFF period is only 2.25 ms, and the stop bit comes next, omitting the custom code and data codes.

Transmission Data

The custom code and data code sections contain the binary data (0 or 1). Data in each of these sections are transmitted LSB first.

The distinction between binary data (0/1) is not simply based on infrared ON/OFF status but ratheron the bit length (on the OFF status length to be exact). Therefore, the length of the custom codesection varies according to the data. It also varies for the data code; however, since inverted datacode is also transmitted, the total data length including data code and inverted data code is always the same (the total number of data bits "0" and "1" is eight).

Figure5-2 Difference between "0" and "1" Data Bit Values in Remote Control Signals

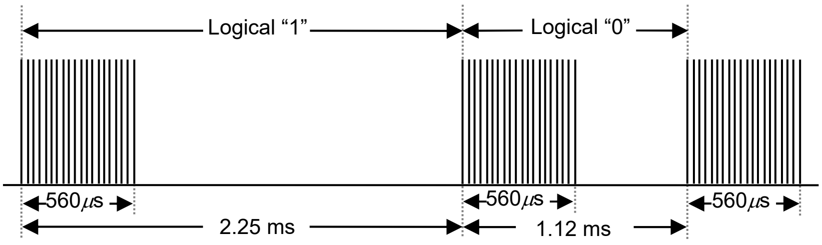


Modulation in Carrier Frequency

Infrared rays are not output consecutively during the entire ON period. Instead, infrared ON periods repeatedly alternate with infrared OFF periods at a constant frequency (called the "carrier frequency"). The standard carrier frequency is 38 kHz. The recommended carrier duty factor is 1/3. These settings help to minimize power consumption.

So the NEC IR transmission protocol uses pulse distance encoding of the message bits. Eachpulse burst is 562.5 µs in length, at a carrier frequency of 38 kHz (26.3 µs, about 21 cycles). Alogical "1" takes 2.25 ms to transmit, while a logical "0" is only half of that, being 1.125 ms.

Figure5-3 Modulation in Carrier Frequency



Data Transmission Sequence

The structure of remote control signals transmitted via this method consists of custom code and data code.

The custom code, which is transmitted first, is 16-bit long but it is divided into two 8-bit sections. Inearly versions of remote control devices, the custom code was only 8 bits long (C0 to C7), and the logically inverted data (C'0 to C'7) was transmitted via the next 8 bits for reliability. Now this C'0 to C'7 section has been reassigned as the second section of the custom code so that the custom code is 16-bit long. When transmitting, the custom code is sent LSB first (C0 to C7), then the custom code' is also sent LSB first (C0' to C7').

Figure5-4 Transmission Sequence of Custom Code Sections

↓ Transmission starts from here

C0	C1	C2	C3	C4	C5	C6	C7	C’0	C’1	C’2	C’3	C’4	C’5	C’6	C’7
Custom Code								Custom Code’							

The data being transmitted is 8-bit long. The logically inverted 8-bit data is transmitted consecutively, so a total of 16 bits are used to transmit the data. When these data are received, the inverted 8-bit data code should be checked as being the logical inversion of the first 8-bit data code, as a reliability tool in order to check that no error has occurred.

Figure5-5 Transmission Sequence of Data Code Sections

↓ Transmission starts from here

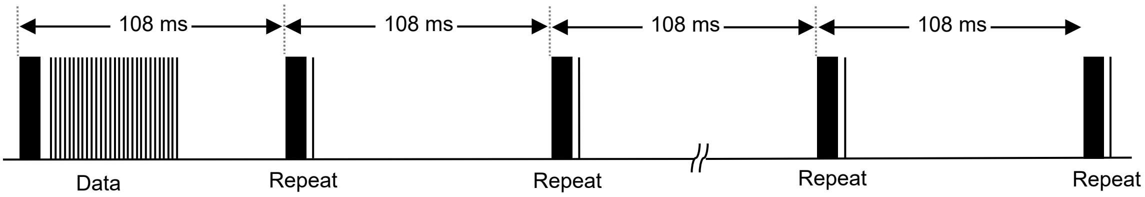
D0	D1	D2	D3	D4	D5	D6	D7	$\overline{\text{D0}}$	$\overline{\text{D1}}$	$\overline{\text{D2}}$	$\overline{\text{D3}}$	$\overline{\text{D4}}$	$\overline{\text{D5}}$	$\overline{\text{D6}}$	$\overline{\text{D7}}$
Data Code								$\overline{\text{Data Code}}$							

Repeat Code

If the key on the remote controller is kept depressed, a repeat code will be issued, typically around 40 ms after the pulse burst that signified the end of the message. A repeat code will continue to besent out at 108 ms intervals, until the key is finally released.
A data code is transmitted only once, so the repeat code consists of only the following, in order:

- a 9 ms leading pulse burst;
- a 2.25 ms space;
- a 562.5 µs pulse burst (stop bit) to mark the end of the space (and hence end of the transmitted repeat code).

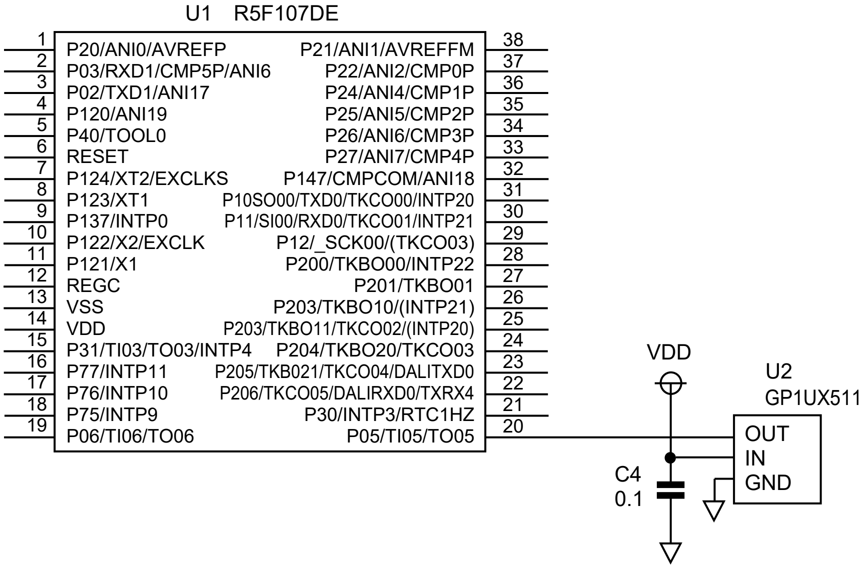
Figure5-6 Transmission of Repeat Codes After an Initial Message Frame



5.1.2 Hardware control interface

A simple 5 V 38 kHz IR remote control receiver is connected to the timer input pin 5 and no other interface circuitry is required between the RL78/I1A input pin and the IR receiver.

Figure5-7 IR Control Interface



5.2 RL78/I1A Features for IR Communication

5.2.1 Peripheral functions

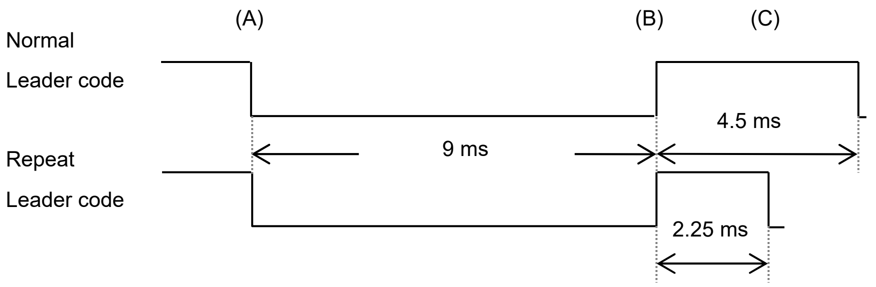
To perform IR signal detection and reception, channels 5 and 6 of the Timer Array Unit are used. Their configurations as well as an overview of the operation are described in the next section.

- Channel 6 configured in Interval Timer: Leader Code Detection.
- Channel 5 configured in Capture & One-Count Mode: Custom and Data Code Bit Length Calculation

5.2.2 Operation overview

Signal edges are used to measure the length of each signal period in order to interpret the received remote control signals.
Since the data that is output from the infrared remote control preamplifier is negative-logic data, these signals are described below as negative-logic input signals.

Figure5-8 Leader Code Detection



TAU channel 6: Leader Code Detection

Since there is a 9 ms OFF period for the leader code, a timer interrupt function checking the reception pin status is used to detect the leader code falling edge of the input signal (marked as A in **Figure5-8**). The timer is TAU channel 6 in interval timer mode.
The rising edge of the input signal (marked as B in Figure5-8) is also checked using the same interrupt function also configured as a time counter, so that the period between both edges can be measured. A minimum time of 7 ms is used when seeking to detect this OFF period.
Then the program measures the period to the next falling edge (marked as C in Figure5-8), and determines whether this is a normal leader code or repeat leader code. A time of at least 3 ms isthe criteria used to distinguish between normal leader code and repeat leader code.
When a correct lead code is detected, TAU channel 5 is started to detect the custom code and data codes. When a repeat leader code is detected the function changing the LED lighting controllevel is called again.

TAU channel 5: Custom Code and Data Code Detections

When detection of the leader code is completed, next action is to calculate the bit length of the custom code and data code.

Although both ON and OFF periods would need to be checked to ensure precision, the data value (0 or 1) can simply be judged by checking the ON period. Thus, it is only a matter of distinguishing the 1.69 ms (2.25 ms – 0.56 ms) ON period of a data value "1" and a 0.565 ms (1.125 ms – 0.56 ms) ON period of a data value "0", using a mid-point value of 1 ms. To perform such a high-level width measurement, TAU channel 5 is configured in capture &one-count mode, with both edges of the TI05 input pin used as start trigger and capture trigger, the rising edge being the start trigger and the falling edge being the capture trigger.

It is then important to precisely check for 32-bit signals. Once a 32-bit signal has been detected the function changing the LED lighting control level is called.

Frame Space

Although the most precise way to check the frame space would be to check for a total frame period of 108 ms, in this case, simply checking whether or not the data length exceeds 32 bits (4 bytes) is enough. Once a 32-bit data is received, a custom code, data code, and inverted data code check and decoding is performed. The last step is the lighting control command execution.

5.3 IR Lighting Control Software

5.3.1 Initialization of the internal peripheral functions

The initialization of the peripheral used for IR reception operation includes the following settings:

- Setting CPU clock frequency to 32 MHz using PLL (16 times the internal high-speed oscillation clock f_{IH} x ½)
- Peripheral function clock supply setting
- I/O port setting
- 16-bit TAU channel 5 setting
  - Setting count clock to f_{CLK} (32 MHz)
  - Setting to capture & one-count mode (Counting up) with both the edges of the TI0n pin input used as a start trigger and a capture trigger
  - Setting to high-level width measurement with the following condition
    - Start trigger: rising edge, capture trigger: falling edge
  - Masking the interrupt INTTM05
- 16-bit TAU channel 6 setting
  - Setting count clock to f_{CLK} (32 MHz)
  - Setting to interval timer mode, start by software
  - Setting the interval time to 100 μs ((TDR06 + 1) / fCLK)
  - Unmasking the interrupt INTTM06

Below is the IR reception peripheral initialization extracted from the sample program ("IR_init()"function from the "r_ir.c" file).

```
PM0.5     = 1;
TMR05     = 0x02CC;        /* Low/High level width                          */
TMIF05    = 0;             /* interrupt flg clear                           */
TMMK05    = 1;             /* INTTM05 disable                               */

TMR06     = 0x0000;        /* @32 MHz                                       */
TDR06     = 3199;          /* The initial value of the period is set to 100us @ 32 MHz  */
TMIF06    = 0;             /* interrupt flg clear                           */
TMMK06    = 0;             /* INTTM06 enable                                */
TS0       = 0x0040;        /* timer ch.6 start                              */
```

5.3.2 Operation description & software flow charts

The following section provides program flow charts and a description of the functions to give the user a clear picture of the IR reception operation using RL78/I1A Timer Array Unit channels.

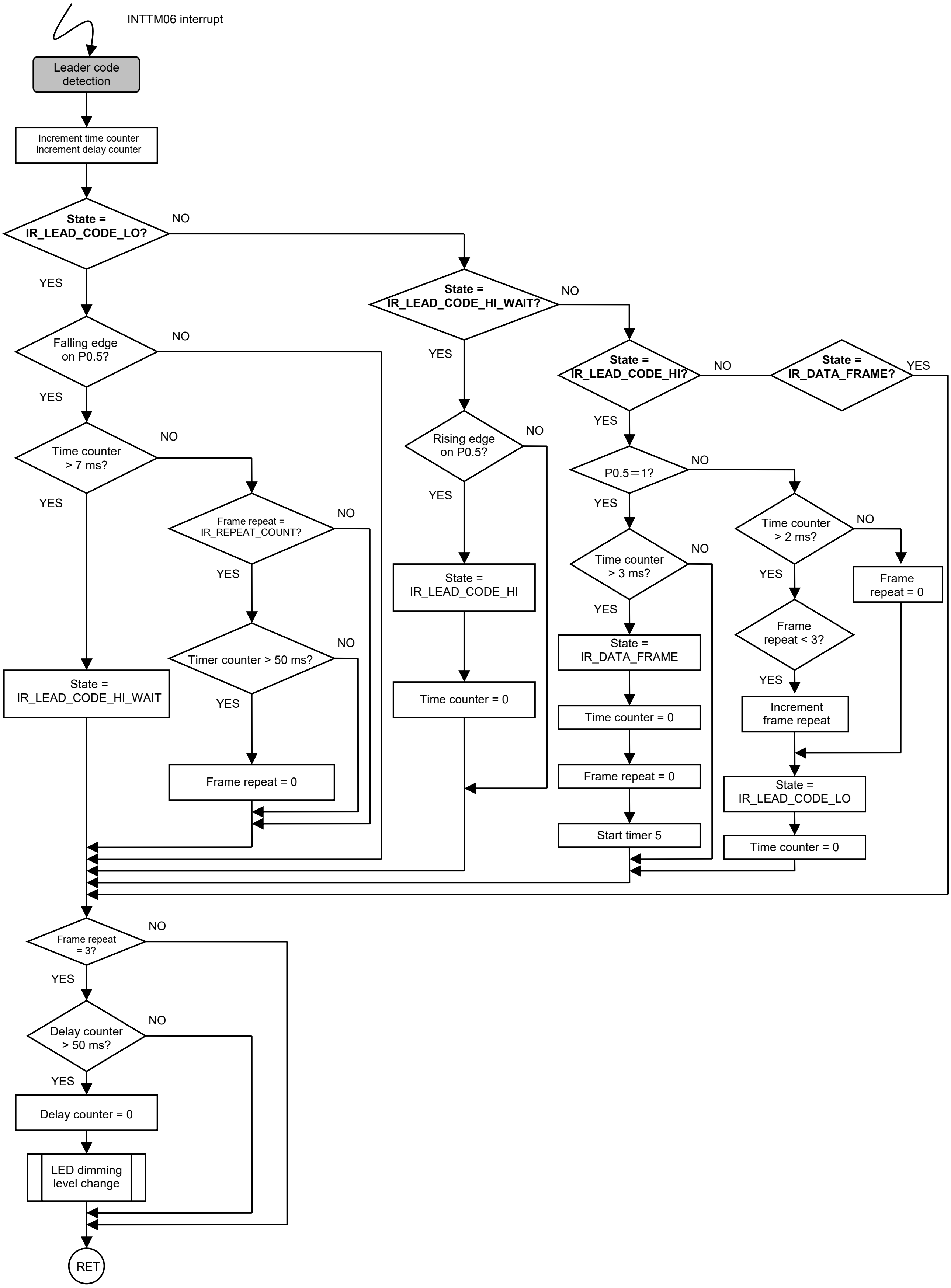
As seen previously, the IR reception process includes the leader code detection following by the bit length detection for "0" or "1" distinction, and each part uses a different timer channel.

The leader code detection process is split into 4 different states:

- Low level detection state (IR_LEAD_CODE_LO): checking for falling edge
- High level waiting state (IR_LEAD_CODE_HI_WAIT): checking for rising edge
- High level width check state (IR_LEAD_CODE_HI): checking for high level width to differentiate normal leader code to repeat leader code
- Custom and data code reception state (IR_DATA_FRAME): exiting the timer channel 6 interrupt service routine and starting timer channel 5 when normal leader code is detected

The flow chart featured below gives a detailed description of this process.

Figure5-9 Leader Code Detection Flow Chart



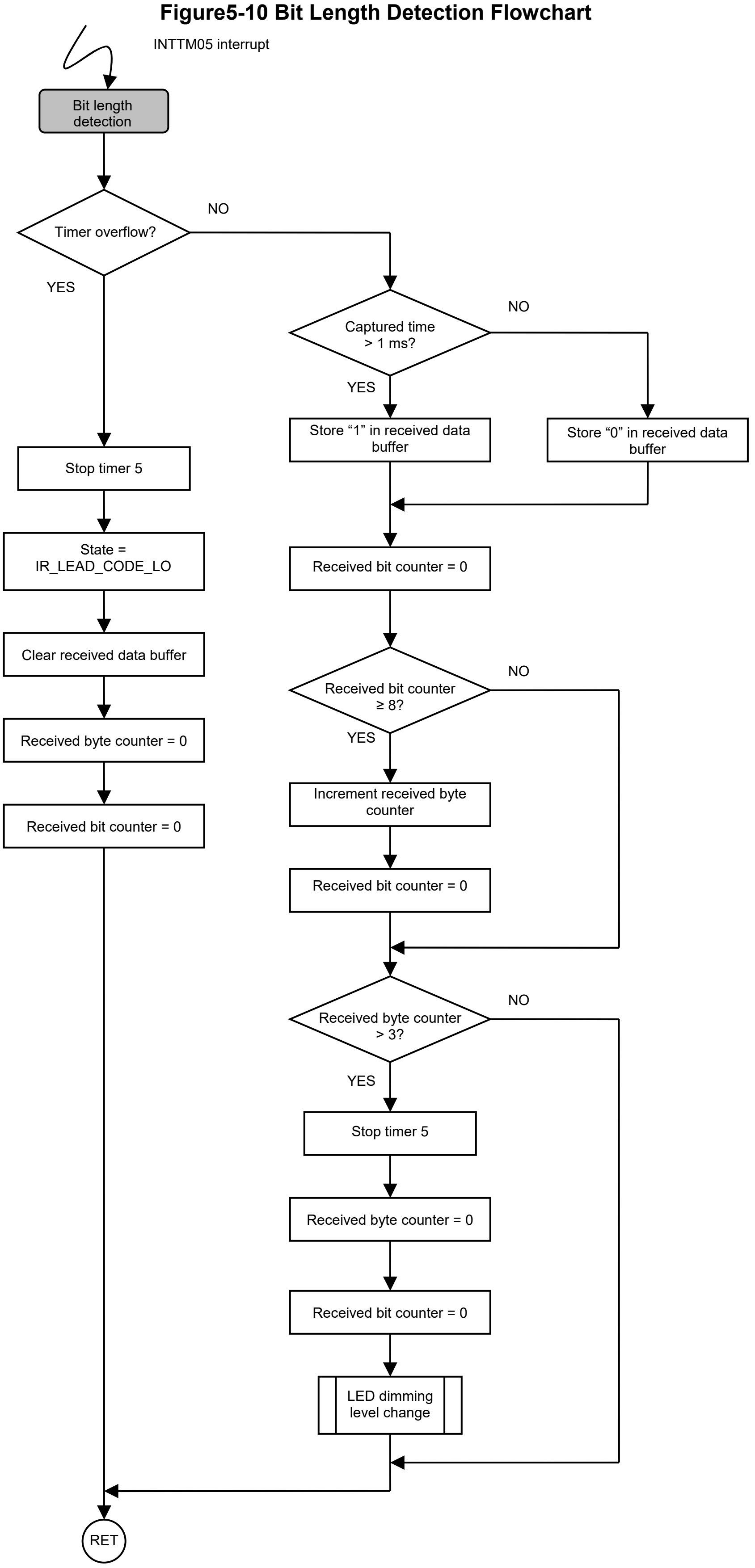
The second action consisting of custom and data code bit length detection is performed simply by checking the ON period:

- When ON time > IR_HIGH_LEVEL_TIME: bit is detected and stored as a logical "1"
- When ON time < IR_HIGH_LEVEL_TIME: bit is detected and stored as a logical "0"

Upon reception of 32 bits (4 bytes), INTTM05 interrupt is disabled and timer channel 5 operation is stopped. The initial low level detection state (IR_LEAD_CODE_LO) is selected againand the function interpreting the received custom and date codes ("IR_ControlLED()") is called to executethe appropriate LED lighting control commands.

The flowchart featured below gives a detailed description of this process.

Figure5-10 Bit Length Detection Flowchart

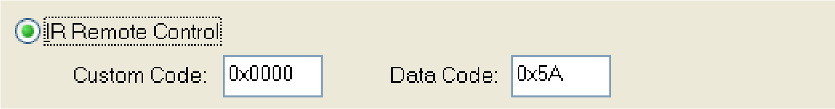


The function "IR_ControlLED()" simply checks that the custom and data codes matches the expected ones. By default, they are defined as follows:

Custom code = 0x0000 and Data code = 0x5A for channel 1 (and 0xDA for channel 2).

These values can be modified in the Applilet EZ for HCD Controller main window, when the dimmer program is set to IR Remote Control it is possible to customize the "Custom Code" and "Data Code" according to the user’s needs.

Figure5-11 Applilet EZ for HCD Controller IR Remote Control Parameters



● IR Remote Control

Custom Code: 0x0000

Data Code: 0x5A

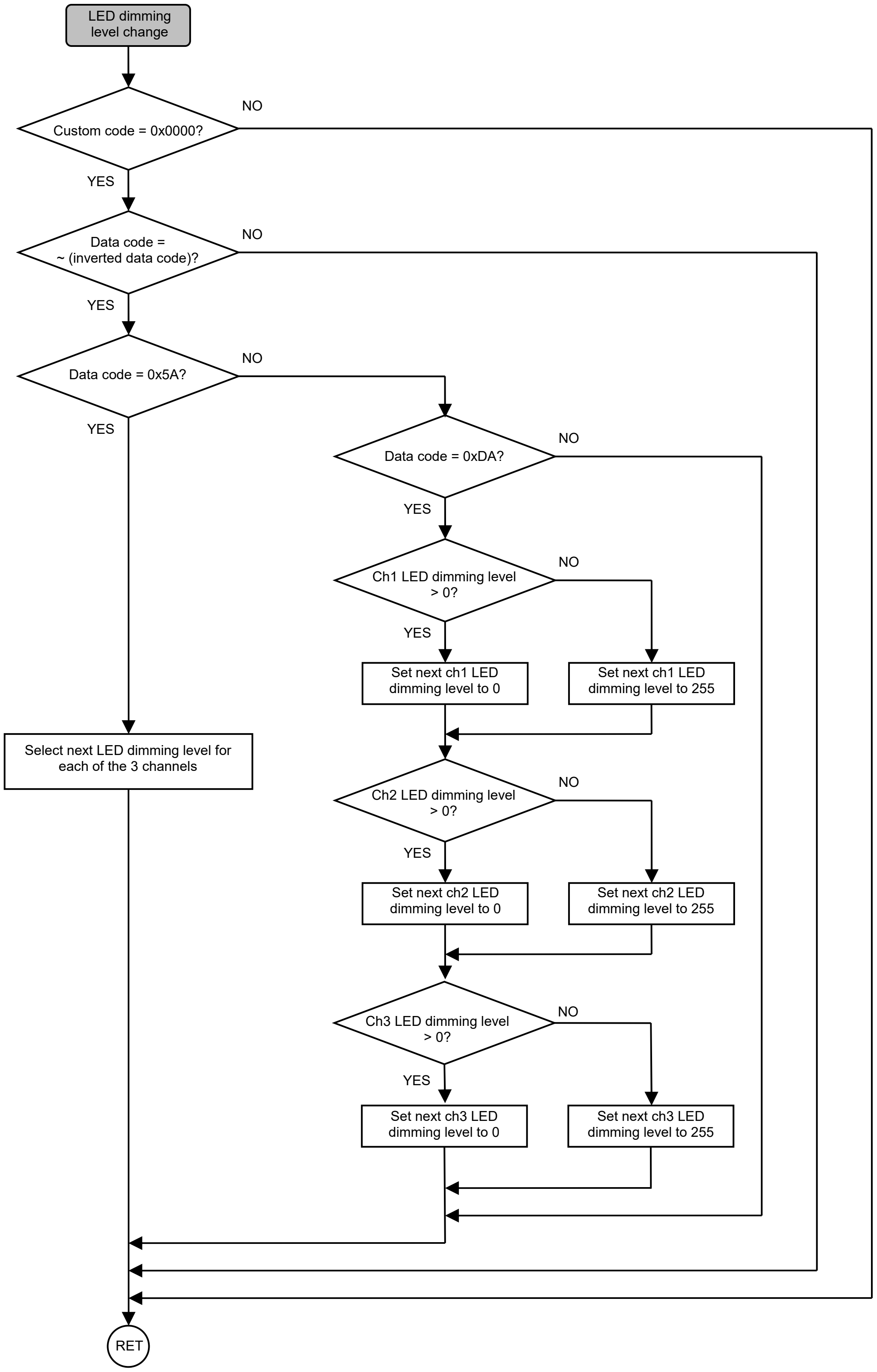
When changing the custom and/or data codes, the corresponding #define macros in the "r_user.h"file of the sample code project is modified accordingly:

```
#define IR_CUSTOM_CODE                    0x0000
#define IR_DATA_CODE                      0x5A
#define IR_DATA_CODE                      0xDA
```

When matching the channel 1 data code (0x5A by default), the function selects the next pre-defined lighting control level and another function takes care of setting each of the 3 LED channels to this new lighting control level (0 to 255).
By default, there are six pre-defined LED lighting control steps that applied one after the other in the following order and in a continuous loop each time a valid IR packet is received: 0, 85, 170,255, 170 and 85.
When matching the channel 2 data code (0xDA), the function toggles the LED lighting control level between 0 and the maximum lighting control level (255).

The flowchart featured below gives a detailed description of this process.

Figure5-12 LED Lighting Control Level Change Flowchart


LED dimming level change
Custom code = 0x0000?
NO
YES
Data code = ~ (inverted data code)?
NO
YES
Data code = 0x5A?
NO
YES
Data code = 0xDA?
NO
YES
Ch1 LED dimming level > 0?
NO
YES
Set next ch1 LED dimming level to 0
Set next ch1 LED dimming level to 255
Select next LED dimming level for each of the 3 channels
Ch2 LED dimming level > 0?
NO
YES
Set next ch2 LED dimming level to 0
Set next ch2 LED dimming level to 255
Ch3 LED dimming level > 0?
NO
YES
Set next ch3 LED dimming level to 0
Set next ch3 LED dimming level to 255
RET

Appendix A　Multi-master

The support to a multi-master was added in IEC62386-101ed2.0, IEC62386-102ed2.0 and IEC62386-103ed1.0[Note]revised and added in 2014.

The specifications about the multi-master are described in IEC62386-101ed2.0 and IEC62386-103ed1.0.

For collision correspondence that is required in Application controller to correspond to multi-master,

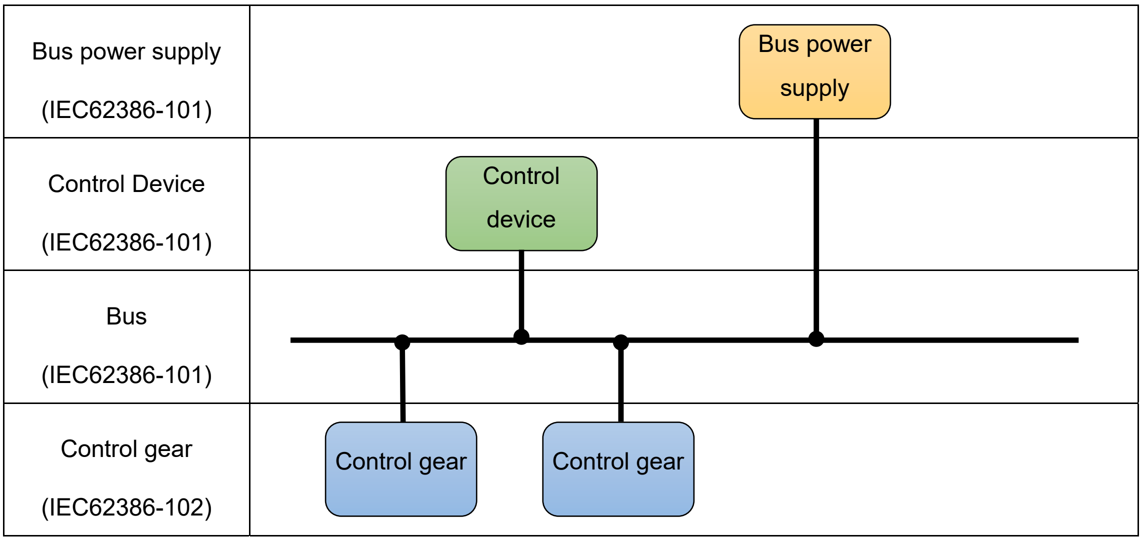
see "Lighting Communication Using RL78 / I1A (transmittion) (R01AN3193EJ0100)".

Note: IEC62386-103ed1.0 was added in the revision of 2014.

It does not exist in separate standards about Control device in the 2009 edition.

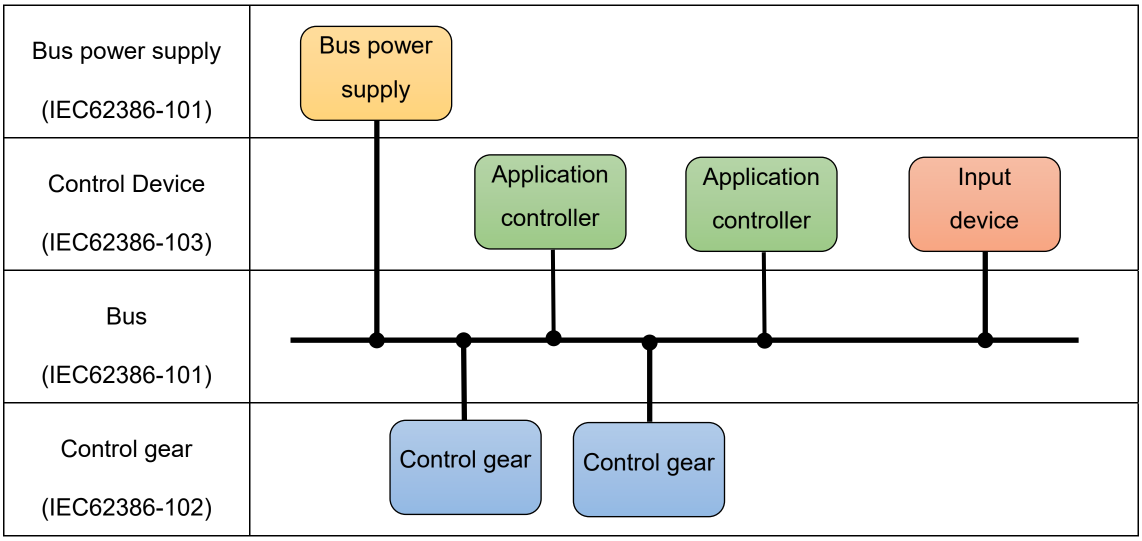
Example of system configuration is shown.

Figure A-0 Example of a single-master system configuration



Only configuration of 1 to n (single-master configuration) to connect up to 64 Control gear (the reception side) to under one Control device (the transmission side) is prescribed in IEC62386101ed1.0 and IEC62386-102ed1.0 of the version in 2009.

Figure A-1 Examples of multi-master system configuration



The configuration of n to n (multi-master configuration) to connect up to 64 Control gear (the reception side) to under up to 64 Control device (the transmission side) was enabled in IEC62386101ed2.0 and IEC62386-103ed1.0 of the revised edition in 2014.

Control device of the 2009 version be changed to Application Controller, and Control device is changed to the generic name on the master side including Input device added newly.

Only Application controller can perform the communication with Control gear.

In the case of multi-master configuration, it is possible to communicate to One Control gear from a plurality of Application controller.

It's necessary to include detection of collision and restored sequence from occurrence in Control device to perform multi-master.

Appendix B DALI(IEC62386-101,102)ed2.0 timing of communication

It's mentioned about a timing change part about communication in IEC62386-101ed2.0.

change part

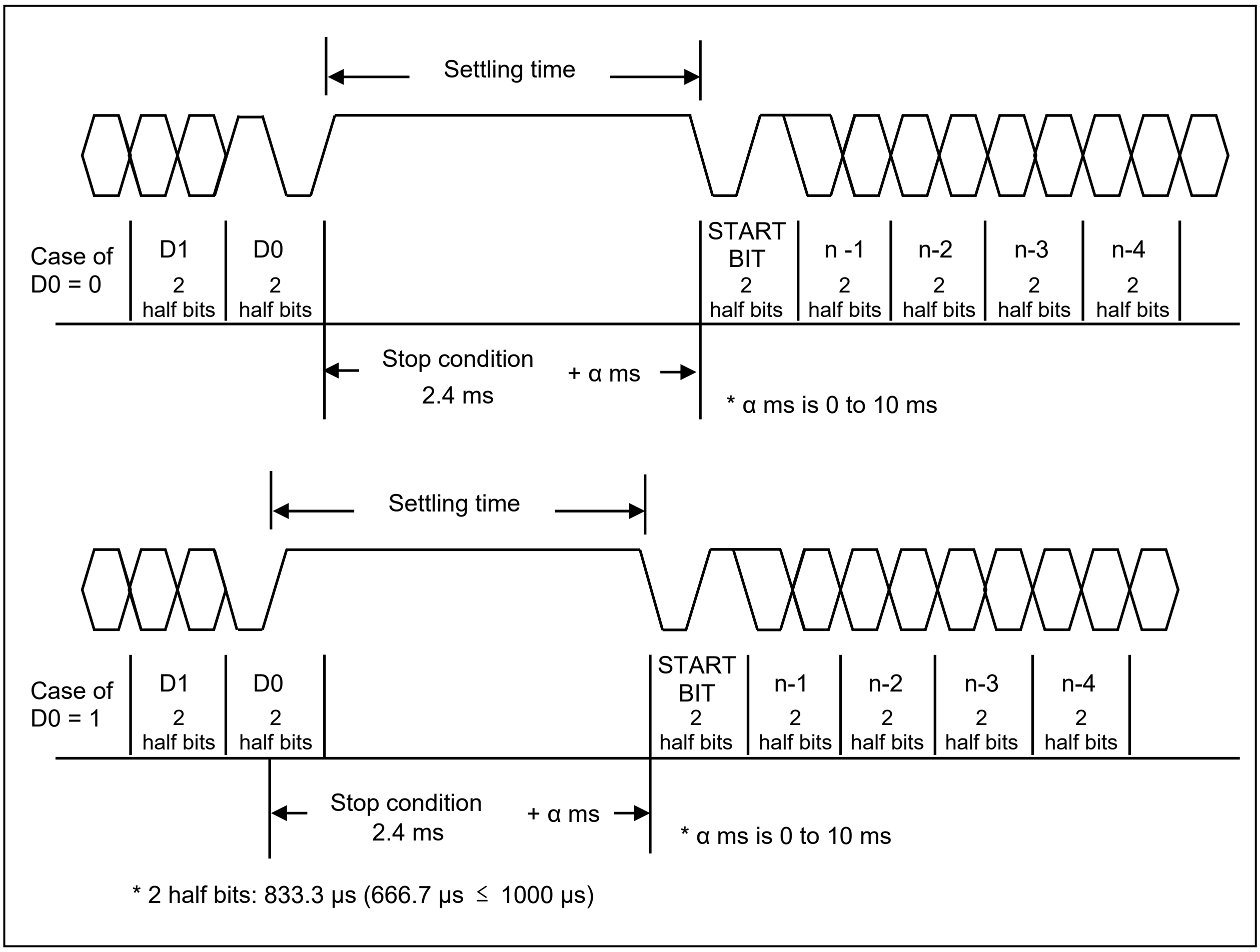
- Settling time
- timing of communication

Note Please check the specifications for details standards.

(1) Settling time

Settling time of IEC62386-101ed2.0 is shown.

Figure B-0 Settling time



Settling time of IEC62386-101ed2.0 is constant regardless of data of last bit.

In IEC62386-101ed2.0, it has defined the start of StopCondition is the last Bit of a rising edge.

Settling time is always from the start of the StopCondition until the start of the StartBit.

The timing between the frame in IEC62386-101ed2.0 is defined in Settling time.

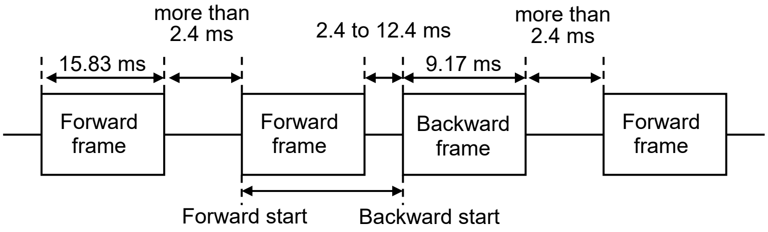
(See Appendix B timing between the frame.)

(2) timing between the frame

DALI is the frame unit and the next timing control is necessary.

- Forward frame width: 15.83 ms(12.66 to 19.00 ms)
- Backward frame width: 9.17 ms(7.33 to 11.00 ms)
- Communication interval between the Forward frame and the Backward frame: 2.4 to 12.4 ms (Settling time)
- Interval between the Forward frame and next Forward frame: more than 2.4 ms(Settling time)
- Interval between the Backward frame and next Forward frame: more than 2.4 ms(Settling time)

Figure B-1timing between the frame



* All of the timing between the frame represents at Settling time.

(3) Signal rise time and fall time

The rise and fall time must be adapted to the Table B-0 of conditions.

Figure B-2 and Figure B-3 shows the level used to measure the tRISE and tFALL.

Table B-0 Signal rise and fall times

	Minimum	Typical	Maximum
tRISE, tFALL for transmitter and multi-master transmitter	3 μ s		
tRISE, tFALL for transmitter			25 μ s
tRISE, tFALL for multi-master transmitter			15 μ s

Note Check the specifications for details of conditions related to timing.

Figure B-2 Maximum signal rise and fall time measurements

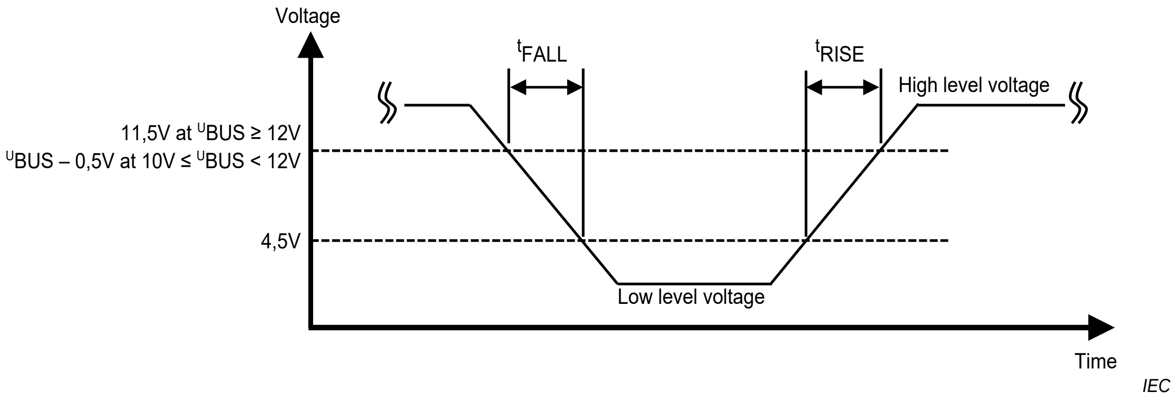
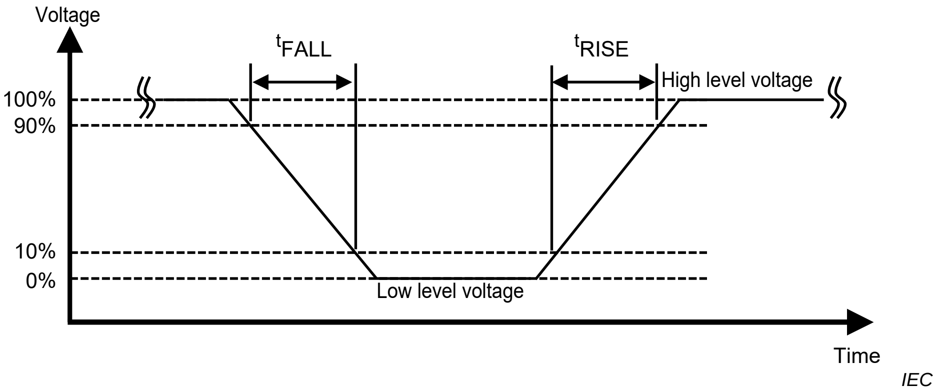


Figure B-3 Minimum signal rise and fall time measurements



(4) Transmitter bit timing

Bit timing of the Transmitter must comply with the limits shown in Table B-1.

Figure B-4 shows a portion of a typical frame.

Regardless of the Low-level voltage and the High-level voltage, timing is measured at the level of 8.0V.

Figure B-4 Bit timing example

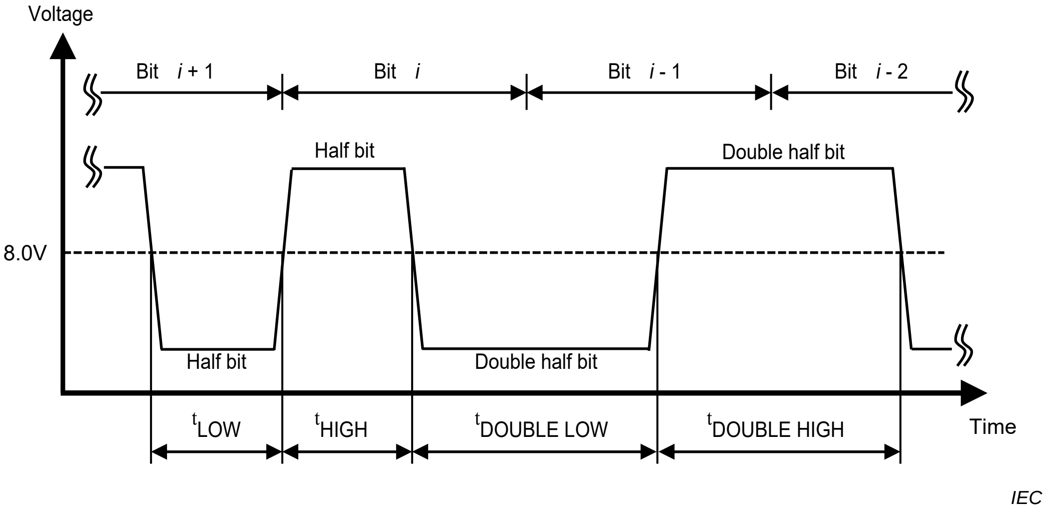


Table B-1 Transmitter bit timing

	Minimum	Typical	Maximum
Half bit time ^{t}HIGH, ^{t}LOW	336,7 μ s	416,7 μ s	466,7 μ s
Double halh bit time ^{t}DOUBLE LOW, ^{t}DOUBLE HIGH	733,3 μ s	833,3 μ s	933,3 μ s
Stop condition time ^{T}STOP	2450 μ s		

(5) Transmitter frame sequence timing

Figure B-5 shows the Settling time between consecutive frames.

For Settling time, it must conform to the values shown in Table B-6.

Figure B-5 Stting time illustraton

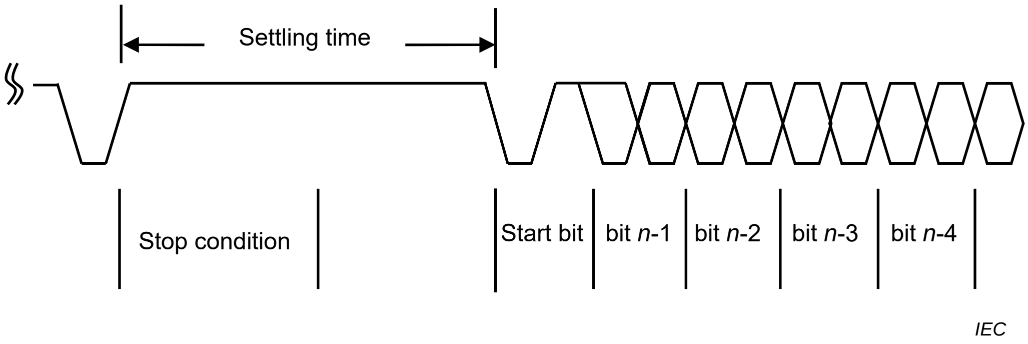


Table B-6 Transmitter settling time value

	Minimum	Typical	Maximum
Settling time between a forward frame and a backward frame	5.5 ms		10.5 ms
Settling time between any other frame and a forward frame	13.5 ms		75.0 ms

Note Check the specifications for details of conditions related to timing.

(6) Receiver bit timing

Receiver determines whether to accept or drop the frame by the following bits timing conditions.

For logical bit that starts at the edge, it must conform to the period of Table B-7 from the start edge to the next edge.

It must conform to the period of Table B-8 from the edge of the logical bit until the next edge.

For Table B-7, there is a possibility of Start bit, Stop condition or the other first Half bit of logic bit. For Table B-8, there is a possibility of Half bit, Double half bit or Stop condition.

Figure B-6 shows an example of which period table B-7 and B-8 are applied.

Table B-7 Receiver timing starting at the beginning of a logical bit

Minimum	Typical	Maximum	Description
		<333.3μs	Gray area
333.3μs	416.7μs	500μs	Half bit
> 500μs		< 750μs	Gray area
750μs		1400μs[a] 45 ms [b]	Bit timing violation
>1400μs[a]		< 2400μs[a]	Gray area
2400μs[a]			Stop condition

a Only applicable for idle state.

b Only applicable for active state. Active state longer than 45 ms shall be interpreted as bus power down.

Table B-8 Receiver timing starting at an edge inside of a logical bit

Minimum	Typical	Maximum	Description
		<333.3μs	Gray area
333.3μs	416.7μs	500μs	Half bit
> 500μs		<666.7μs	Gray area[c]
666.7μs	833.3μs	1000μs	2 hail bit
>1000μs		< 1200μs	Gray area
1200μs		<1400μs[a] 45 ms [b]	Bit timing violation
>1400μs[a]		< 2400μs[a]	Gray area
2400μs[a]			Stop condition

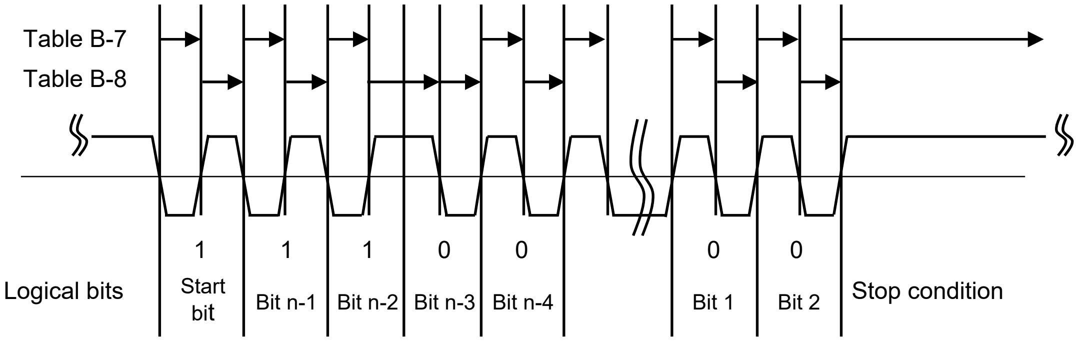
a Only applicable for idle state.

b Only applicable for active state. Active state longer than 45 ms shall be interpreted as bus power down.

c If an edge occurs after a time within the grey area, the receiver can conclude that it is a timing violation. This can be caused for example, by overlapping backward frames.

Note Check the specifications for details of conditions related to timing.

Figure B-6 Receiver timing decision example



(7) Receiver frame sequence timing

Decoding of the new frame must be started only after the detection of the Stop condition.

Receiver must conform to the frame sequence that contains the Settling time given in Table B-9.

Table B-9 Receiver settling time values

	Minimum	Typical	Maximum	Description
Settling time between forward frame and backward frame	>1.4 ms		< 2.4 ms	Gray area
	2.4 ms		12.4 ms	Frame shall be accepted as backward frame.
	>12.4 ms		< 13.4 ms	Gray area
	13.4 ms			Frame shall not be interpreted as backward frame.
Settling time between forward frame and forward frame	>1.4 ms		< 2.4 ms	Gray area
	2.4 ms			Frame shall be accepted as forward frame.
Settling time between first and second forward frame of send-twice forward frames	>1.4 ms		< 2.4 ms	Gray area
	2.4 ms		94 ms	Frames shall be accepted as send-twice forward frames.
	>94 ms		< 105 ms	Gray area
	105 ms			Frames shall be accepted as two separate forward frames.
Settling time between backward frame and forward frame	>1.4 ms		<2.4 ms	Gray area
	2.4 ms			Frame shall be accepted as forward frame.

Note This requirement ensures such that if the Receiver during the transmission of 24bit Forward frame has started, 24bit Forward frame is not to be construed as 16bit Forward frame.

Note Check the specifications for details of conditions related to timing.

(8) Collision detection

Collision detection is applied during the transmission of any of the Forward frame.

When a signal different from the value of Table B-13 was included in the one Multi-master transmitter transmit on the bus, Multi-master transmitter has to stop a transmission immediately.

When it can be guaranteed that a signal made before stopping a transmission doesn't meet the condition of Destroy areaof Table B-12 and Table B-13, Transmitter which stopped a transmission has to return to Collision avoidance.

Figure B-7 shows an example of which period Table B-12 and B-13 is applied to.

Table B-12 Checking a logical bit, starting at an edge at the beginning of the bit

Minimum	Typical	Maximum	Description
		< 100μs	Gray area
100μs		356.7μs	Destroy area [a]
>356.7μs		<400.0μs	Gray area
400.0μs		433,3μs	Valid half bit
>433.3μs		<476,7μs	Gray area
476.7μs			Destroy area [a,b]

a Signals within the destroy area shall lead to collision recovery.

b Only applicable for active state.

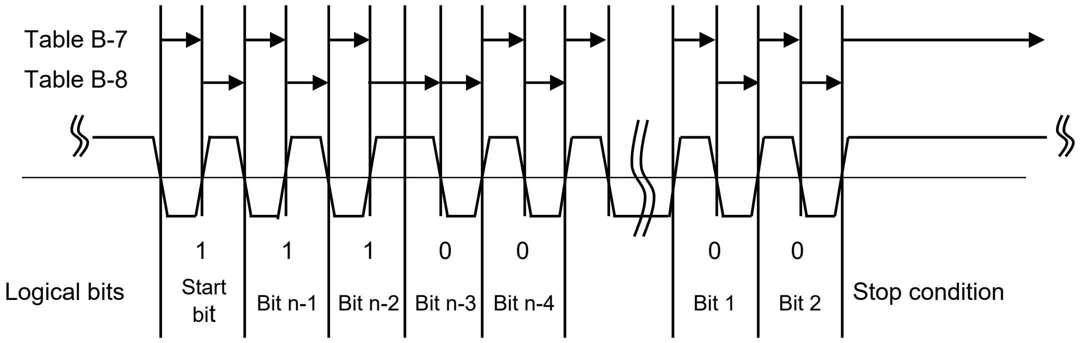
Table B-13 Checking a logical bit, starting at an edge inside the bit

Minimum	Typical	Maximum	Description
		< 100μs	Gray area
100μs		356.7μs	Destroy area [a]
>356.7μs		<400.0μs	Gray area
400.0μs		433.3μs	Valid half bit
>433.3μs		<476.7μs	Gray area
476.7μs		723.3μs	Destroy area [a]
>723.3μs		< 800μs	Gray area
800.0μs	833.3μs	866.7μs	2 valid half bit
>866,7μs		<943.3μs	Gray area
943,3μs			Destroy area [ab]

a Signals within the destroy area shall lead to collision recovery.

b Only applicable for active state.

Figure B-7 Collision detection timing decision example



Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/contact/

REVISION HISTORY

Rev.	Page	Description	
		Page	Summary
1.00	Mar. 28, 2012	ー	First edition issued
2.01	Mar. 26, 2013	p.1	Change of description in **Readers**
		p.6	**Figure 2-2. Structure of the Forward Frame** Change of description of address byte (8 bits) and data byte (8 bits)
		p.7	Modification of value in **Figure 2-4. Timing Between Frames**
		p.7，8	Change of description in **2.1.3.3 (1) Address byte**
		p.11	Change of **Figure 2 7. Backward Frame Reception Timing Chart Example for DALI Communication**
		p.12	Addition of Fade Time to **Table 2 3. Parameters saved by using the EEPROM emulation function**
		p.17	**2.3.1 Operation and software flowchart** - Change of description in [Summary] for Initialization processing flowchart
		p.28,29,31, 32	Change of description on "Y" in Remark in **2.4 DALI Commands**
		p.33	Addition of Remark to **2.4 (5) Application extended commands**
		p.46	**3.1.2 Hardware control interface** Change of DALI/UART4 to UART0 Change of DALIRxD4 to RxD0
3.00	Mar. 31, 2016	ー	Change of the title to the LightingCommunication using RL78/I1A(Reception)
		p. 5	Addition of a description of **Applilet EZ for HCD Controller**
		p. 7, 8, 9	Addition of description of **3.1.2 DALI standard configuration**and **3.1.3 DALI system configuration**
		p. 12,13	Addition of description of **DALISettling time**
		p. 14	Addition of description ofthe transmission and reception timing of IEC62386-102ed2.0
		p. 36-	Addition of an item description of the command list of IEC62386-102ed2.0
		p. 47-	Modification of the list of functions
		p. 68	Change of Figure 4-6 to the image of Applilet EZ for HCD V9.0
		p. 81	Change Figure 5-11 to the image of Applilet EZ for HCD V9.0
		p. 82-91	Addition of Appendix A, Appendix B

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of an MPU or MCU in the same group but having a different part number may differ in terms of the 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HALII Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 5.0