RL78 Family

PMBus communication sample software using SIS

Introduction

This application note describes a sample application for PMBus communication using the PMBus Master module SIS (Software Integration System) and the PMBus Slave module SIS (Software Integration System).

Supported standards are as follows

- PMBus Specification Rev. 1.4 Part I
- PMBus Specification Rev. 1.4 Part II
- System Management Bus (SMBus) Specification Version 3.2

For details on PMBus communication standards, refer to the respective standard documents listed above.

Target Device

RL78/G24

Contents

1. Specifications Overview

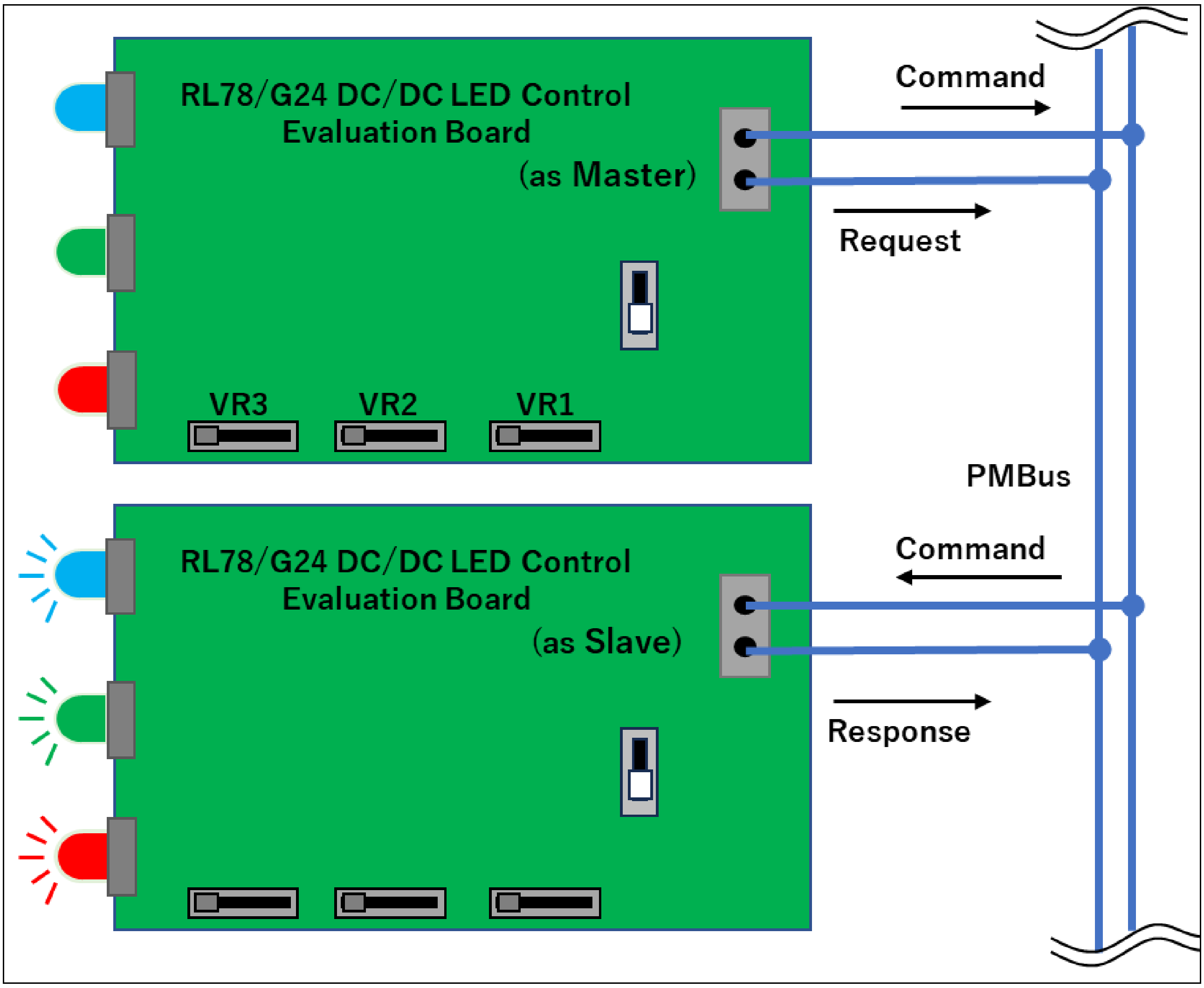
This application note describes sample applications that realize master and slave features in PMBus communication. PMBus (Power Management Bus) is an open standard protocol for power management and defines the means of communication with power devices.

The PMBus Master module SIS and PMBus Slave module SIS are used in the PMBus communication protocol stack in the sample application. Refer to the following documents for detailed information on PMBus Master Module SIS and PMBus Slave Module SIS.

- RL78 Family PMBus Master module Software Integration System (R01AN7188)
- RL78 Family PMBus Slave module Software Integration System (R01AN6985)

The following figure shows a schematic diagram of the sample application. The sample application runs on the RL78/G24 DC/DC LED Control Evaluation Board. The three LEDs on the slave board are used as power supply units. The PMBus command is sent in conjunction with the volume switches (VR1, VR2, VR3) on the master board. The LEDs on the slave board are dimmed according to the commands and data received from the master board.

Figure 1-1 PMBus Sample Application Operation Diagram



2. Operation Check Conditions

The sample application has been tested in the following environments.

Table 2.1 Operation Check Conditions

Item	Description
Microcontroller used	RL78/G24 (R7F101GLG2DFB)
Board used	RL78/G24 DC/DC LED Control Evaluation Board from Renesas Electronics Corp.
Operating frequency	• High-speed on-chip oscillator clock (f_{HOCO}): 8MHz • PLL clock (f_{PLL}): 96MHz • CPU/peripheral hardware clock (f_{CLK}): 48MHz
Operating voltage	5.0V
Integrated development environment (CS+)	CS+ for CC V8.10.00 from Renesas Electronics Corp.
C compiler (CS+)	CC-RL V1.12.01 from Renesas Electronics Corp.
Integrated development environment (IAR)	IAR Embedded Workbench IDE V5.10.3 from IAR Systems
C compiler (IAR)	IAR C/C++ Compiler for Renesas RL78 V5.10.3 from IAR Systems
Smart Configurator (SC)	V1.8.0
Emulator	E2 Emulator Lite

3. Description of the Hardware

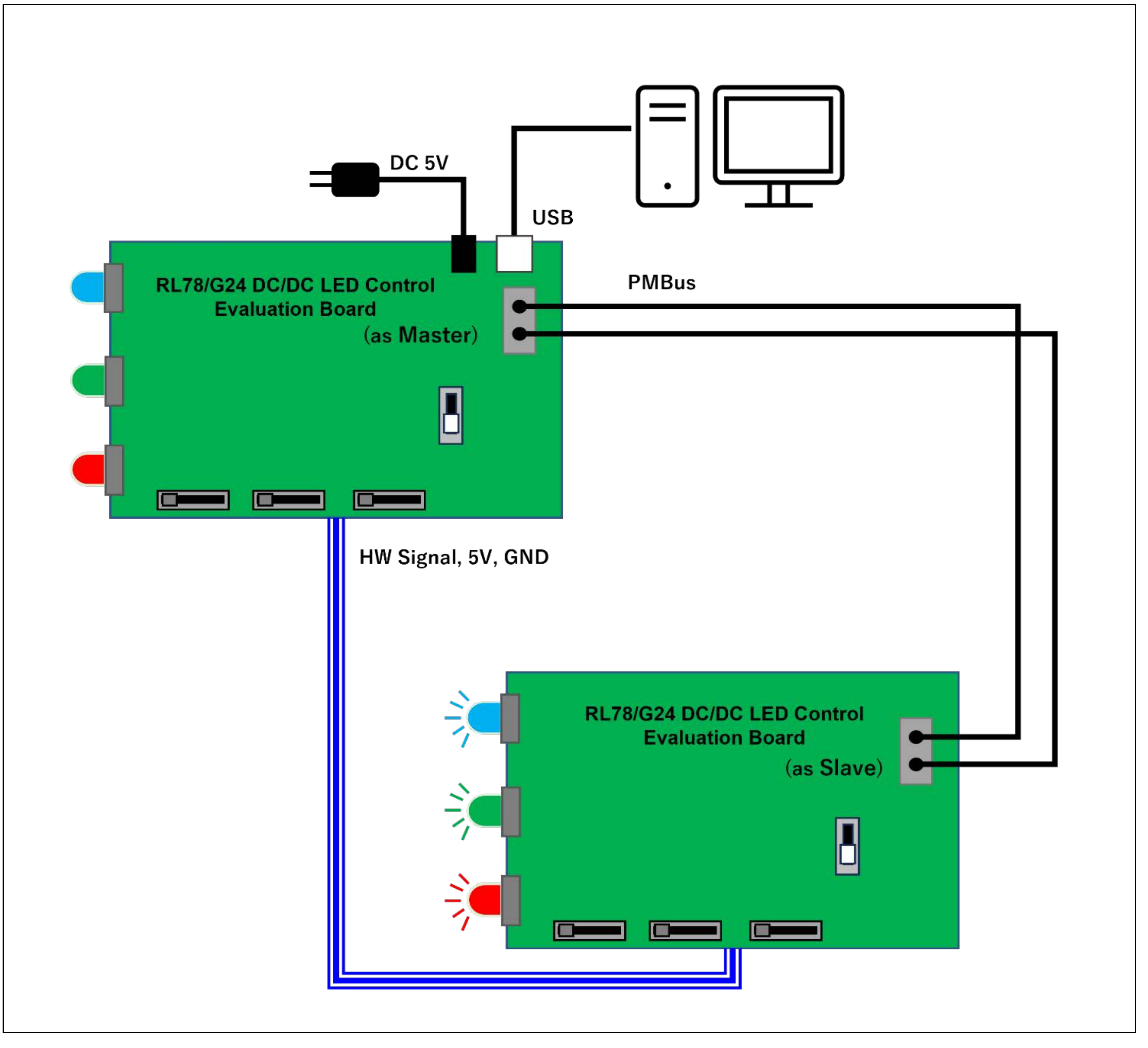
3.1 Description of the Hardware

The system configuration is shown below.

Connect a master (controller) device and a slave device as shown in the figure below. The master and slave devices operate on the RL78/G24 DC/DC LED Control Evaluation Board.

By connecting the master device and PC with a USB cable, the PMBus communication log between the master and slave devices can be displayed on the PC screen.

Figure 3-1 System Configuration Diagram



3.2 List of pins used

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

Table 3.1 Pin used and features (master)

Pin Name	I/O	Description
P60 / SCLA0	Input/Output	I^{2}C Serial Clock
P61 / SDAA0	Input/Output	I^{2}C Serial Data Bus
P10	Output	Control Signal Output
P140	Input	Control Signal value switch (SW2)

Table 3.2 Pin used and features (slave)

Pin Name	I/O	Description
P60 / SCLA0	Input/Output	I^{2}C Serial Clock
P61 / SDAA0	Input/Output	I^{2}C Serial Data Bus
P10	Input	Control Signal Input
P140	Input	Write Protect Input

*Although not used in this sample, the PMBus Slave module SIS also supports alert processing using the SMBALERT# pin. Refer to the following document for details.

- RL78 Family PMBus Master Module Software Integration System (R01AN7188)
- RL78 Family PMBus Slave Module Software Integration System (R01AN6985)

4. Description of the Software

4.1 Environment Construction

4.1.1 Software Installation

The flash writing software Renesas Flash Programmer (hereinafter referred to as RFP) is required to write the firmware. Download the software from the Renesas Electronics website.

Refer to the following User's Manual for basic RFP operations.

- Renesas Flash Programmer V3.13 Flash memory programming software User’s Manual (R20UT5312)

4.1.2 Writing firmware

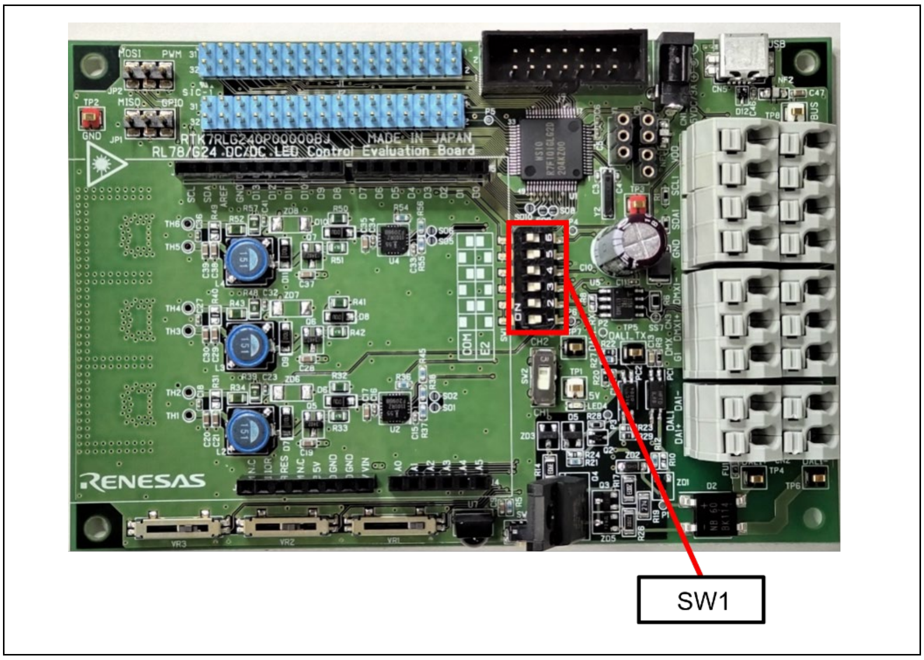
Write the target firmware to the master and slave boards respectively. The Motorola S-type format file is stored in the project folder, so use RFP to write the flash. Refer to 4.6.1 Folder Structure for information on storage folders.

Master: PMBusMaster_Sample.mot

Slave: PMBusSlave_Sample.mot

When writing firmware to the RL78/G24 DC/DC LED Control Evaluation Board, it is necessary to set SW1 on the board according to the connection method.

Figure 4-1 RL78/G24 DC/DC LED Control Evaluation Board SW1



RTK7RLG240P00000BJ MADE IN JAPAN
RL78/G24 DC/DC LED Control Evaluation Board
RENESAS
SW1

Table 4.1 RL78/G24 DC/DC LED Control Evaluation Board SW1 setting

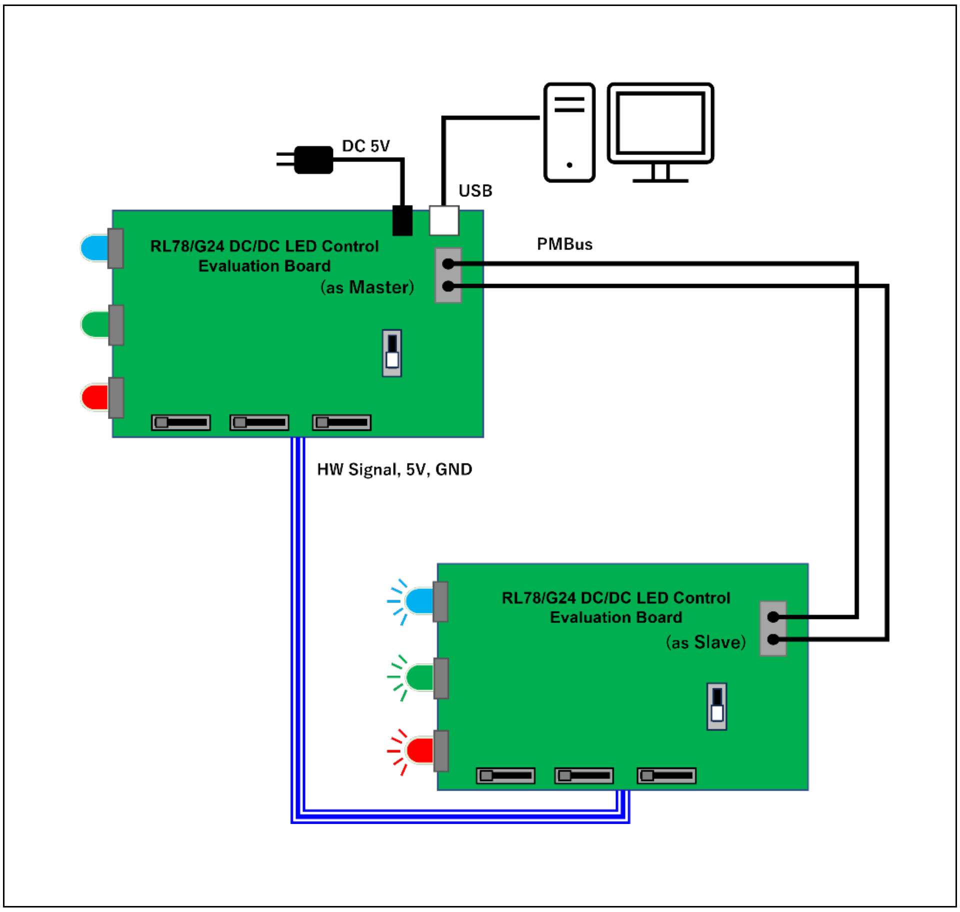
SW1	COM port	E2
1	ON	OFF
2	OFF	ON
3	ON	OFF
4	OFF	ON
5	ON	OFF
6	OFF	ON
7	ON	OFF
8	OFF	ON

4.1.3 How to connect the sample application to hardware

This section describes how to connect the hardware. Connect an AC adapter to either the master board side or the slave board side to supply power. Also, by connecting the USB terminal on the master board to a PC, the PMBus communication log can be monitored.

How to connect a master board and a slave board is explained on the next page.

Figure 4-2 Hardware configuration diagram

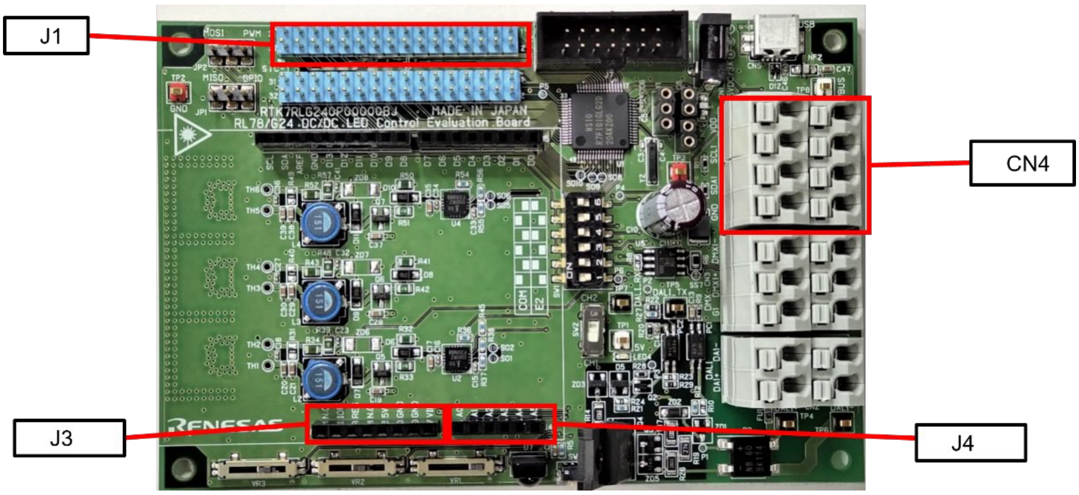


The connections between the master and slave boards are described below.

Table 4.2 Master/Slave Board Connection

Item	Master board	Slave board
SMBCLK	CN4-5 (SCLA0)	CN4-5 (SCLA0)
SMBDAT	CN4-3 (SDAA0)	CN4-3 (SDAA0)
Control	J4-6 (P10)	J4-6 (P10)
DC5V	J3-5	J3-5
GND	J3-7	J3-7

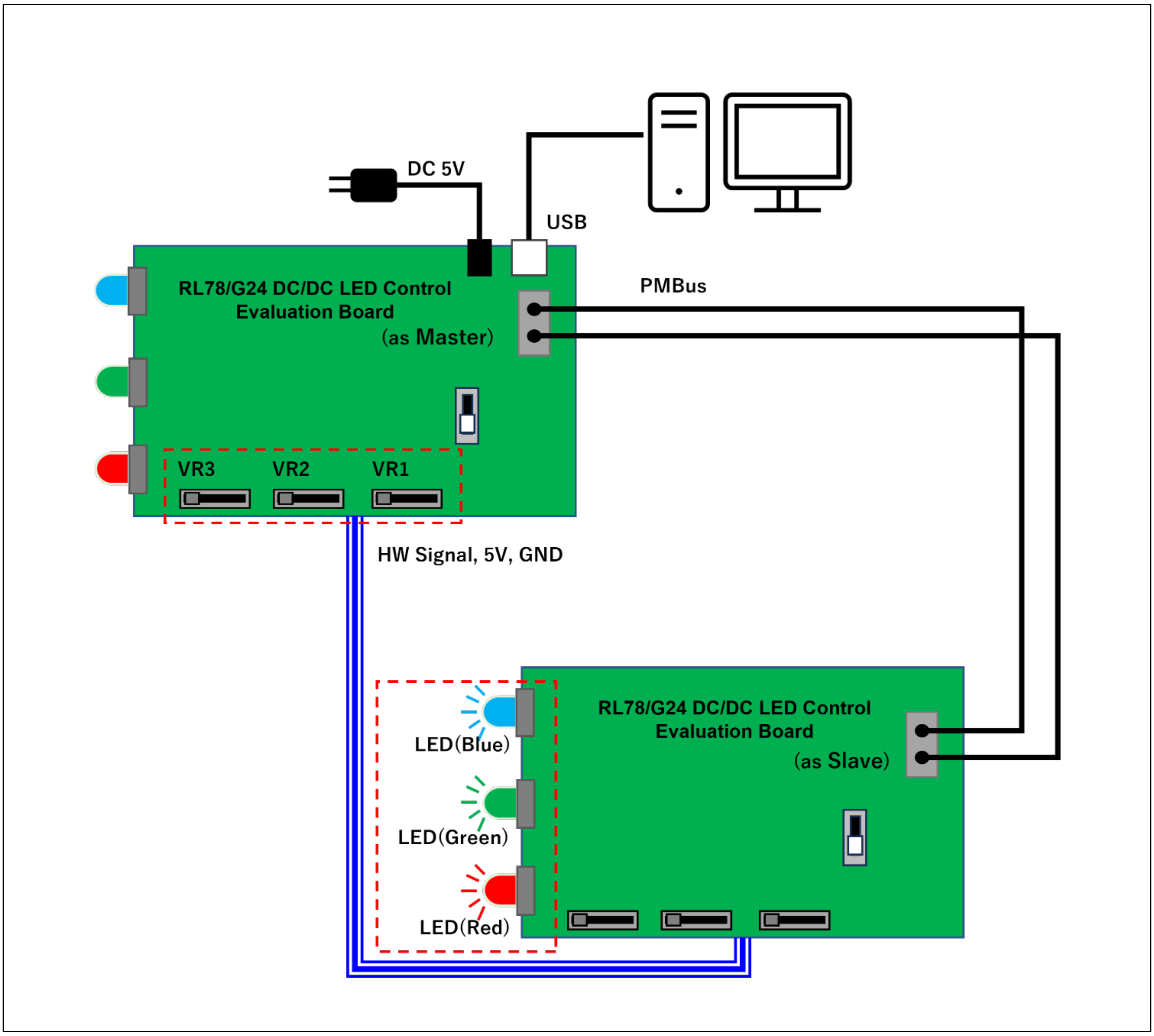
Figure 4-3 Pins on the RL78/G24 DC/DC LED Control Evaluation Board



4.2 Overview of Operation

The following is an overview of the sample application.

Figure 4-4 System Configuration


DC 5V
USB
PMBus
RL78/G24 DC/DC LED Control Evaluation Board (as Master)
VR3
VR2
VR1
HW Signal, 5V, GND
LED(Blue)
LED(Green)
LED(Red)
RL78/G24 DC/DC LED Control Evaluation Board (as Slave)

The dimming levels of the three LEDs (red, green, and blue) on the slave board are switched in conjunction with the volume switches (VR1, VR2, and VR3) on the master board. If there is a change in the volume switch level, the master sends a PMBus command to the slave to write the LED output level (target value). In response, the slave switches the LED output level.

In addition, the master periodically sends commands to know the LED output level (actual output value) of the slave and reads data from the slave.

By connecting the PC and the master board with a USB cable, these command communication logs can be displayed on the PC screen.

4.2.1 Communication Format Specifications

This section describes the PMBus communication format used in this sample application.

Legend:

S : Start condition

Sr : Repeat start condition

Rd : Read(1)

Wr : Write(0)

ACK : Acknowledge

NACK : Not acknowledge

PEC : Packet Error Code

P : Stop condition

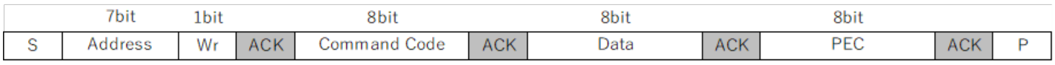
□ : Master (or Host) → Slave

▨ : Slave → Master (or Host)

4.2.1.1 Write Byte/Word

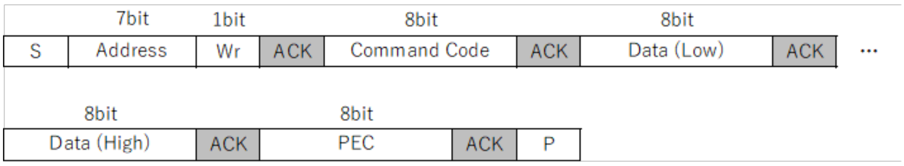
Sends a command code and corresponding write data to an arbitrary target address.

Figure 4-5 Communication format Write (Single byte)



	7bit	1bit		8bit		8bit		8bit		
S	Address	Wr	ACK	Command Code	ACK	Data	ACK	PEC	ACK	P

Figure 4-6 Communication format Write (Multi byte)



	7bit	1bit		8bit		8bit		
S	Address	Wr	ACK	Command Code	ACK	Data (Low)	ACK	…

8bit		8bit		
Data (High)	ACK	PEC	ACK	P

4.2.1.2 Read Byte/Word

Sends a command code to an arbitrary target address. Next, the response data corresponding to the command code is read from the target device.

Figure 4-7 Communication format Read (Single byte)

	7bit	1bit		8bit	
S	Address	Wr	ACK	Command Code	ACK

	7bit	1bit		8bit		8bit		
Sr	Address	Rd	ACK	Data	ACK	PEC	NACK	P

Figure 4-8 Communication format Read (Multi byte)

	7bit	1bit		8bit	
S	Address	Wr	ACK	Command Code	ACK

	7bit	1bit		8bit			8bit		8bit		
Sr	Address	Rd	ACK	Data (Low)	ACK	…	Data (High)	ACK	PEC	NACK	P

4.2.1.3 Host Notify

When a slave device detects a communication error or device failure, it sends its own address information and status information (STATUS_WORD) in the Host Notify communication format below.

Figure 4-9 Communication format Host Notify

	7bit	1bit		8bit		8bit		8bit		
S	SMBus Host Address	Wr	ACK	Device Address	ACK	STATUS_WORD(Low)	ACK	STATUS_WORD(High)	ACK	P

4.2.1.4 Other Communication Formats

Although not used in this sample, the PMBus Slave module SIS also supports receiving commands in communication formats other than those listed above. For details, refer to the following document.

- RL78 Family PMBus Master Module Software Integration System (R01AN7188)
- RL78 Family PMBus Slave Module Software Integration System (R01AN6985)

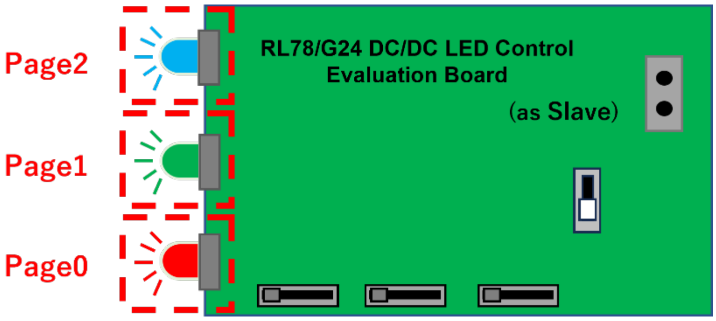
4.2.2 PMBus Command Specifications

This section describes the PMBus commands used in this sample application.

4.2.2.1 PAGE

The PAGE command provides a feature to assign multiple outputs within a single device. In this sample application, pages are assigned to each LED on the slave board, and the PAGE command is used to switch the LED to be controlled. The command is sent in the aforementioned Write Byte/Word communication format, and 1-byte data (page value) is written.

Figure 4-10 Page Assignment



4.2.2.2 VOUT_COMMAND

VOUT_COMMAND is a command to set the output voltage of a device to a specified value. In this sample application, the output level of the LED to be controlled is set. The command is sent in the aforementioned Write Byte/Word communication format, and 2 bytes of data (output target value) are written.

4.2.2.3 READ_VOUT

READ_VOUT is a command to respond the actual measured output voltage. In this sample application, the actual output level of the LED to be controlled is responded. The command is sent in the Read Byte/Word communication format described above and reads 2 bytes of data (current output level value) from the slave.

4.2.3 Application Operation Specifications

This section describes the operation specifications of this sample application.

4.2.3.1 LED Dimming Operation

When there is a change in the volume switch (VR1, VR2, VR3) input value on the master board, the following commands are sent from the master to the slaves to switch the LEDs to be controlled and to indicate LED output.

- PAGE
- VOUT_COMMAND

The slave determines the target LED based on the PAGE command received and reflects the output value set by VOUT_COMMAND to the LED to be controlled.

4.2.3.2 Reading LED Output Values

The master sends the following command to the slave at regular cycles (5-second intervals) to obtain the actual output values of the LEDs.

- PAGE
- READ_VOUT

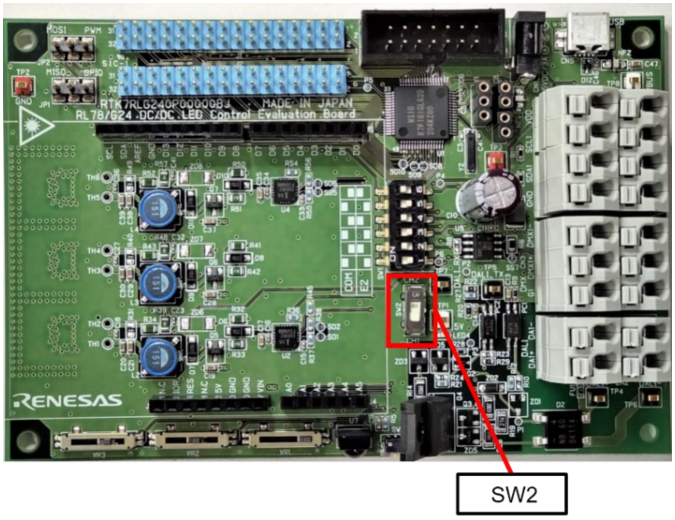
The slave determines the target LED from the PAGE command received and responds to the READ_VOUT command with the current output value of the LED.

4.2.3.3 Control Signal

Turning SW2 on the master board ON/OFF switches the output level of the Control Signal connected between the master and slave.

The slave turns off the LED when the Control Signal is OFF.

Figure 4-11 RL78/G24 DC/DC LED Control Evaluation Board SW2



4.2.3.4 Write Protect

Turning SW2 on the slave board ON/OFF switches the Write Protect signal, and when the Write Protect signal is ON, rewriting of internal information by commands is prohibited and is not reflected even if PAGE or VOUT_COMMAND is received. If a Write command such as PAGE or VOUT_COMMAND is received while the rewriting of internal information is inhibited, the aforementioned Host Notify is sent from the slave as a communication error.

4.2.3.5 PMBus Communication Log

PMBus communication logs can be monitored by connecting the master board to a PC.

Use serial communication software (e.g. Tera Term) to select and connect the USB COM port connected to the RL78/G24 DC/DC LED Control Evaluation Board. The connection settings are as follows.

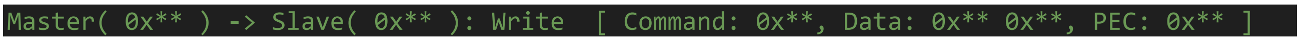
Table 4.3 Serial communication connection settings

Item	Settings
Communication baud rate	9600
Data bits	8bit
Parity	none
Stop bits	1bit
Flow Control	none

This section describes the format for displaying the communication log.

(1) Write

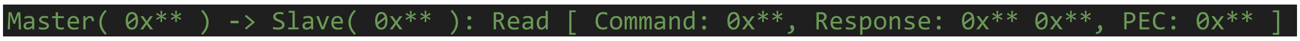
When a command in the Write communication format is sent, the following communication log is output. Source and destination address information, command codes, write data, and PEC Byte information are displayed.



```
Master( 0x** ) -> Slave( 0x** ): Write  [ Command: 0x**, Data: 0x** 0x**, PEC: 0x** ]
```

(2) Read

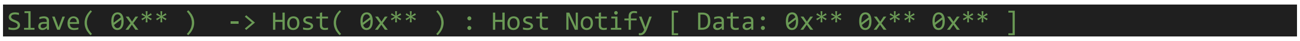
When a Read communication format command is sent, the following communication log is output. Source and destination address information, command codes, response data, and PEC Byte information are displayed.



```
Master( 0x** ) -> Slave( 0x** ): Read [ Command: 0x**, Response: 0x** 0x**, PEC: 0x** ]
```

(3) Host Notify

When a Host Notify message is detected, the following communication log is output. Source and destination address information and transmission data (STATUS_WORD information) are displayed.



```
Slave( 0x** )  -> Host( 0x** ) : Host Notify [ Data: 0x** 0x** 0x** ]
```

4.3 Operating Procedure

The following is an example of the operating procedure for this application.

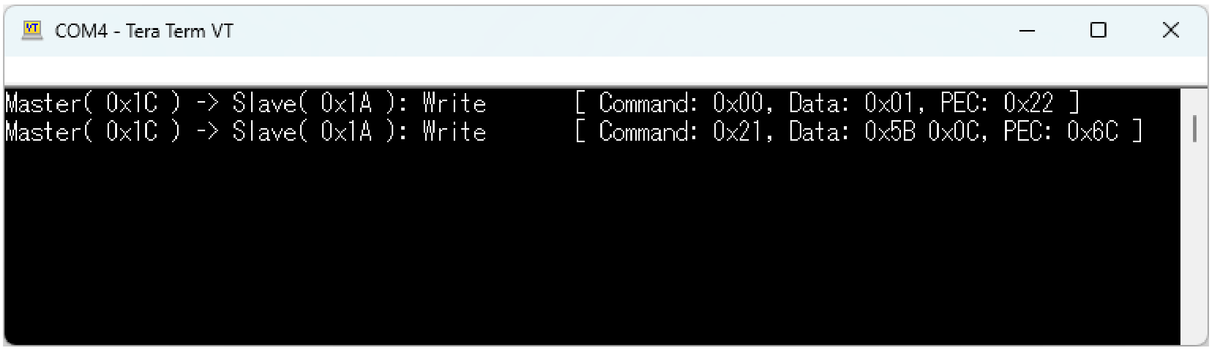
4.3.1 Common Procedure

1. Set SW2 on the master board to ON (CH2 side).
2. Set SW2 on the slave board to OFF (CH1 side).
3. Connect each board and supply power according to 4.1 Environment Construction.
4. Connect the master board and PC serially according to 4.2.3.5 PMBus Communication Log.

4.3.2 Check LED dimming operation by Write communication

1. Set the volume SW (VR1, VR2, VR3) on the master board to the arbitrary level.
2. Check the light intensity of the LEDs on the slave board changes in conjunction with the SW operation.
3. In addition, check that the PAGE and VOUT_COMMAND commands are sent on the serial communication screen.

Figure 4-12 Write Communication Log

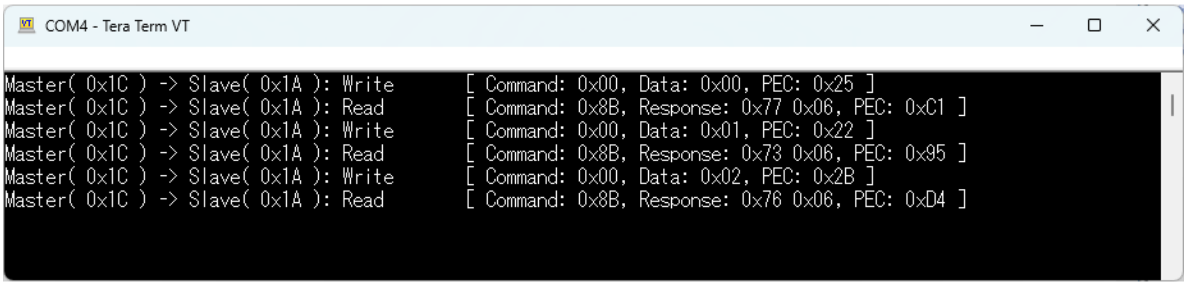


```
Master( 0x1C ) -> Slave( 0x1A ): Write       [ Command: 0x00, Data: 0x01, PEC: 0x22 ]
Master( 0x1C ) -> Slave( 0x1A ): Write       [ Command: 0x21, Data: 0x5B 0x0C, PEC: 0x6C ]
```

4.3.3 Check LED current output value by Read communication

1. Check on the serial communication screen that PAGE and READ_VOUT commands are periodically sent at 5s intervals.
2. The current output level value of the LEDs can be checked in the Read communication of the serial communication screen.
3. When the LED output level (target value) is changed using the volume SW (VR1, VR2, VR3) on the master board, the current output value of the LEDs will also change accordingly.

Figure 4-13 Read Communication Log



```
Master( 0x1C ) -> Slave( 0x1A ): Write       [ Command: 0x00, Data: 0x00, PEC: 0x25 ]
Master( 0x1C ) -> Slave( 0x1A ): Read        [ Command: 0x8B, Response: 0x77 0x06, PEC: 0xC1 ]
Master( 0x1C ) -> Slave( 0x1A ): Write       [ Command: 0x00, Data: 0x01, PEC: 0x22 ]
Master( 0x1C ) -> Slave( 0x1A ): Read        [ Command: 0x8B, Response: 0x73 0x06, PEC: 0x95 ]
Master( 0x1C ) -> Slave( 0x1A ): Write       [ Command: 0x00, Data: 0x02, PEC: 0x2B ]
Master( 0x1C ) -> Slave( 0x1A ): Read        [ Command: 0x8B, Response: 0x76 0x06, PEC: 0xD4 ]
```

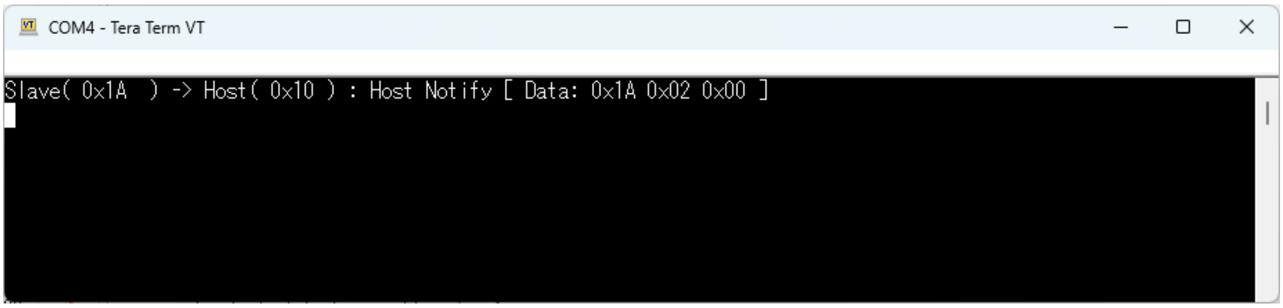
4.3.4 Check Control Signal operation

1. Set the volume SW (VR1, VR2, VR3) on the master board to the maximum level and check that the LED on the slave board is lit.
2. When SW2 on the master board is turned off, the Control Signal connected between the master and slave boards becomes inactive. At this time, check that the LED on the slave board turns off.
3. When SW2 on the master board is turned on, check that the LEDs on the slave board turn on again.

4.3.5 Check Write Protect operation

1. Turn on SW2 on the slave board. The Write Protect signal connected on the board becomes active.
2. Check that the light intensity of the LEDs on the slave board does not change when the volume SW (VR1, VR2, VR3) on the master board is changed.
3. In addition, check on the serial communication screen that Host Notify is issued from the slave board as a communication error.

Figure 4-14 Host Notify Communication Log



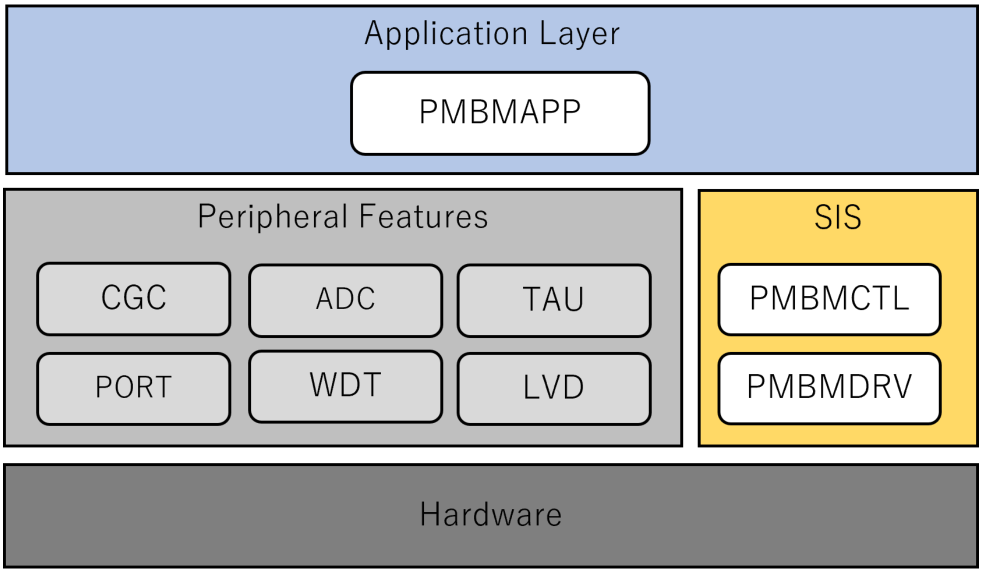
```
Slave( 0x1A  ) -> Host( 0x10 ) : Host Notify [ Data: 0x1A 0x02 0x00 ]
```

4.4 Feature Overview (PMBus Master)

4.4.1 Feature Block Diagram

The sample application's feature block diagram is shown below.

Figure 4-15 Feature Block Diagram



4.4.2 SIS (Software Integration System)

4.4.2.1 PMBMDRV (PMBus Master Driver)

This is the driver part of PMBus Master module SIS. This module provides the PMBus master send and receive operation using the serial interface IICA.

Refer to the following document for detailed information on the module.

- RL78 Family PMBus Master Module Software Integration System (R01AN7188)

4.4.2.2 PMBMCTL (PMBus Master Controller)

This is the middleware part of the PMBus Master module SIS. It accepts command and data sending requests from applications.

Refer to the following document for detailed information on the module.

- RL78 Family PMBus Master Module Software Integration System (R01AN7188)

The configuration settings in this sample application are shown below.

Table 4-4 PMBus Master Driver Settings List

Item	Setting value	Description
Bus Speed	100kHz	Specifies the bus speed.
IICA Input Mode	SMBus	Specifies the IICA input mode.
SDAA and SCLA signal falling times (tF)[ns]	0	Sets the fall time of the SDAA and SCLA signals. (Note1)
SDAA and SCLA signal rising times (tR)[ns]	0	Sets the rise time of the SDAA and SCLA signals. (Note1)
Digital filter	OFF	Specifies whether the digital filter is enabled or not. (Note2)
Interrupt level for INTIICA0	Level 3(Lowest)	Sets the INTIICA0 interrupt priority.
Pin for Control Signal	P10	Selects the Control Signal pin.
Pin for SMBALERT#	Unused	Selects the SMBALERT#pin.
Timer resource for device timeout measurement	TAU0_0	Selects the Timed Source for measuring device timeout.
Host Notify Supported	Supported	Selects whether SMBus Host Notify is supported.

Table 4-5 PMBus Master Controller Settings List

Item	Setting value	Description
Support the SMBus Packet Error Checking (PEC)	Supported	Selects whether to support PEC

4.4.3 Application Layer

4.4.3.1 PMBMAPP (PMBus Master Application)

This is the application part of the PMBus communication feature. It sends commands to the slave at regular intervals to acquire LED output values. When there is a change in the input value of the volume switch, a command is sent to the slave to indicate the LED output level. In addition, the PMBus communication log is output to the serial communication screen.

(1) Application task processing

(a) LED dimming operation

LED dimming operation is performed at 1000ms intervals. If there are any changes in the volume switch (VR1, VR2, VR3) values, the API function "RM_PMBMCTL_WriteData" provided by PMBMCTL is used to specify the command and page values, and the PAGE command is sent, PMBus communication log is output to the serial screen. Next, the API function "RM_PMBMCTL_WriteData" is used to specify the command and LED output level, and send the VOUT_COMMAND command and output the PMBus communication log to the serial screen.

(b) LED output value readout

Reads LED output values at 5000ms intervals. The API function "RM_PMBMCTL_WriteData" provided by PMBMCTL is used to specify the command and page value, send the PAGE command, and output the PMBus communication log to the serial screen. Next, the API function "RM_PMBMCTL_ReadData" is used to specify the command and send the READ_VOUT command to obtain the LED output values. The PMBus communication log is output to the serial communication screen.

(c) Control Signal setting

Control Signal is set at 100ms intervals. It obtains the status of SW2 and set the Control Signal by specifying active if the port is Hi level, inactive if the port is Low level, using the API function "RM_PMBMCTL_SetControl" provided by PMBMCTL.

(2) Processing at the time of a callback from the PMBus Master module

(a) HostNotifyReceived callback

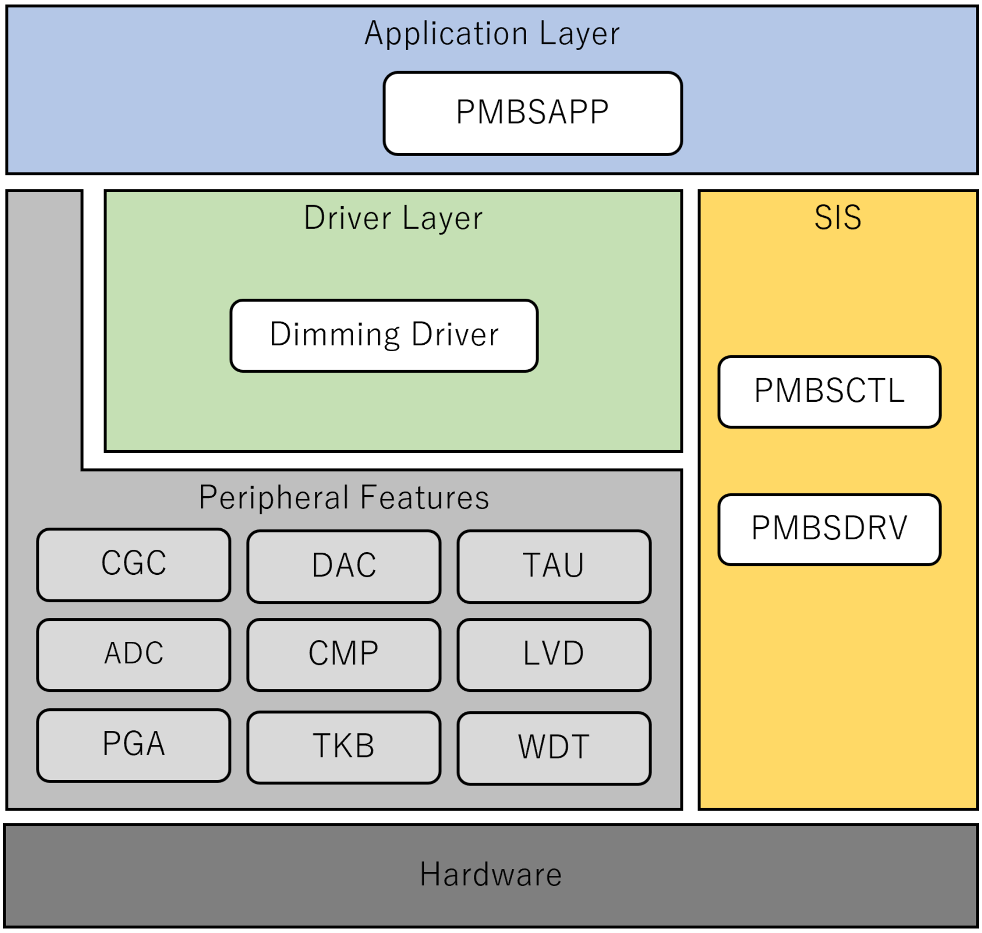
The callback is issued by PMBMCTL at the timing when Host Notify is received. The notified STATUS_WORD information is acquired and the PMBus communication log is output to the serial screen.

4.5 Feature Overview (PMBus Slave)

4.5.1 Feature Block Diagram

The sample application's feature block diagram is shown below.

Figure 4-16 Feature Block Diagram



4.5.2 Driver Layer

4.5.2.1 Dimming Driver

This driver performs dimming control for three LED colors (red, green, and blue).

The following peripheral features are used to control the dimming of each LED.

- ADC
- PGA
- DAC
- CMP
- TKB

Refer to the following document for more information on dimming control.

- LED control with RL78/G24 (R01AN6673)

4.5.3 SIS (Software Integration System)

4.5.3.1 PMBSDRV (PMBus Slave Driver)

This is the driver part of PMBus Slave module SIS. It realizes PMBus slave send/receive operation using the serial interface IICA.

Refer to the following document for detailed information on the module.

- RL78 Family PMBus Slave Module Software Integration System (R01AN6985)

4.5.3.2 PMBSCTL (PMBus Slave Controller)

This is the middleware part of the PMBus Slave module SIS, which constructs packet data and interprets communication formats based on the PMBus communication specifications.

Refer to the following document for detailed information on the module.

- RL78 Family PMBus Slave Module Software Integration System (R01AN6985)

The configuration settings in this sample application are shown below.

Table 4-6 PMBus Slave Driver Settings List (1/2)

Item	Setting value	Description
Bus Speed	100kHz	Specifies the bus speed.
IICA Input Mode	SMBus	Specifies the IICA input mode.
SDAA and SCLA signal falling times (tF)[ns]	0	Sets the fall time of the SDAA and SCLA signals. (Note1)
SDAA and SCLA signal rising times (tR)[ns]	0	Sets the rise time of the SDAA and SCLA signals. (Note1)
Digital filter	OFF	Specifies whether the digital filter is enabled or not. (Note2)

Table 4-7 PMBus Slave Driver Settings List (2/2)

Item	Setting value	Description
Interrupt level for INTIICA0	Level 3(Lowest)	Sets the INTIICA0 interrupt priority.
How to set the device address	Setting by GUI	Specifies how to set the device address.
Device address	0x1A	Sets the device address.
Pin for setting device address 1	P11	Selects the device address (bit1) pin.
Pin for setting device address 2	P12	Selects the device address (bit2) pin.
Pin for setting device address 3	P13	Selects the device address (bit3) pin.
Pin for setting device address 4	P14	Selects the device address (bit4) pin.
Pin for setting device address 5	P15	Selects the device address (bit5) pin.
Pin for setting device address 6	P16	Selects the device address (bit6) pin.
Pin for setting device address 7	P17	Selects the device address (bit7) pin.

Table 4-8 PMBus Slave Controller

Item	Setting value	Description
Support the SMBus Packet Error Checking (PEC)	Supported	Selects whether to support PEC

4.5.4 Application Layer

4.5.4.1 PMBSAPP (PMBus Slave Application)

This is the application part of the PMBus communication function. It interprets received commands and performs operations when receiving commands. It also receives Control Signal and Write Protect signals, and determines whether device output is ON or OFF and updates data.

(1) Processing at callback from PMBus Slave module

Callback notifications are received from the middleware section of the PMBus Slave module SIS (hereinafter referred to as PMBSCTL) when PMBus commands are received or when a communication error is detected. The following is a description of the processing for each callback.

(a) CheckCommandSupported callback

This is called back from PMBSCTL when a command code is received. It determines if the notified command is supported by the application and returns PMBSCTL_CBK_RTN_OK as the return value of the callback function if it is a supported command. If it is an unsupported command, it returns PMBSCTL_CBK_RTN_CMD_NOT_SUPPORTED as the return value of the callback function.

(b) WriteDataReceived callback

This is called back from PMBSCTL at the timing when a Write command is received. If it is a valid value, the internal data is updated and PMBSCTL_CBK_RTN_OK is returned as the return value of the callback function. If it is an invalid value, it does not update the internal data and returns PMBSCTL_CBK_RTN_DATA_NG as the return value. Also, when the Write Protect signal is active, the internal data is not updated and PMBSCTL_CBK_RTN_DATA_NG is returned.

(c) ReadDataReceived callback

This is called back from PMBSCTL when a Read command is received. It sets the data to be responded and returns PMBSCTL_CBK_RTN_OK.

(d) FaultNotification callback

This is called back from PMBSCTL when a communication error is detected. It updates the STATUS_WORD information based on the notified fault information and issues a Host Notify.

Refer to the following document for details on each callback function and sequence information.

- RL78 Family PMBus Slave module Software Integration System (R01AN6985)

(2) Application Task Processing

Application task processing is performed at 5ms intervals and includes the following processing.

(a) Acquisition of HW signal information

The API function "RM_PMBSCTL_GetHWSignal" provided by PMBSCTL is used to acquire HW signal information and update internal data (Control Signal and Write Protect signal level).

(b) Acquisition of LED current output value

Acquires the current output value of each LED from the dimming driver and updates the internal data. This internal data is set as response data by the aforementioned ReadDataReceived callback when the READ_VOUT command is received.

(c) LED Target Value Setting

Reflects the internal data value as the target value of each LED to the dimming driver. This internal data is updated by the aforementioned WriteDataReceived callback when VOUT_COMMAND is received.

4.6 Software Configuration

This section describes the software configuration of the sample application.

4.6.1 Folder Structure

The folder structure of this sample application is shown below.

Table 4.9 Folder Structure (Master)

Folder, File Name	Description
PMBusMaster_Sample<DIR>	Sample application folder
\App<DIR>	Application program storage folder
r_main.c	Main source file
r_pmbm_app.c	PMBusMaster application source file
r_pmbm_app.h	PMBusMaster application header file
\Utility<DIR>	Common program storage folder
r_timer16.c	Timer module source file
r_timer16.h	Timer module header file
r_usertype.h	Type definition header file
\src<DIR>	Smart configurator generation file storage folder
\smc_gen<DIR>	Smart configurator generation folder (Omitted below) *PMBus Master module SIS code is generated under
\DefaultBuild<DIR> (CC-RL)	Project output file storage folder (CC-RL development environment only)
PMBusMaster_Sample.mot	Motorola S-type format file used for writing to firmware
\Release\Exe<DIR> (IAR)	Project output file storage folder (IAR development environment only)
PMBusMaster_Sample.mot	Motorola S-type format file used for writing to firmware

Table 4.10 Folder Structure (Slave)

Folder, File Name	Description
PMBusSlave_Sample<DIR>	Sample application folder
\App<DIR>	Application program storage folder
r_main.c	Main source file
r_pmbs_app.c	PMBusSlave application source file
r_pmbs_app.h	PMBusSlave application header file
\Driver<DIR>	Driver program storage folder
r_led.c	Dimming driver source file
r_led.h	Dimming driver header file
r_led1.c	Dimming driver source file for LED1(Red)
r_led1.h	Dimming driver header file for LED1(Red)
r_led2.c	Dimming driver source file for LED2(Green)
r_led2.h	Dimming driver header file for LED2(Green)
r_led3.c	Dimming driver source file for LED3(Blue)
r_led3.h	Dimming driver header file for LED3(Blue)
\Utility<DIR>	Common program storage folder
r_timer16.c	Timer module source file
r_timer16.h	Timer module header file
r_usertype.h	Type definition header file
\src<DIR>	Smart configurator generation file storage folder
\smc_gen<DIR>	Smart configurator generation folder (Omitted below) *PMBus Slave module SIS code is generated under
\DefaultBuild<DIR> (CC-RL)	Project output file storage folder (CC-RL development environment only)
PMBusSlave_Sample.mot	Motorola S-type format file used for writing to firmware
\Release\Exe<DIR> (IAR)	Project output file storage folder (IAR development environment only)
PMBusSlave_Sample.mot	Motorola S-type format file used for writing to firmware

4.6.2 List of Option Byte Settings

The following table shows the option byte settings for the sample application.

Table 4.11 List of Option Byte Settings

Address	Set value	Contents
000C0H / 010C0H	01111101B	Watchdog timer operation enabled (count starts after reset is released)
000C1H / 010C1H	11111011B	LVD reset mode (falling: 2.97V, rising: 2.91V)
000C2H / 010C2H	11101010B	HS mode, high-speed on-chip oscillator: 8MHz
000C3H / 010C3H	10000100B	On-chip debugging permitted

5. Notes

- This sample application is configured and operated in accordance with the RL78/G24 DC/DC LED Control Evaluation Board.
- The operation provided by this sample application is only a sample. The application layer should be redesigned and evaluated according to the intended use.

6. Documents for Reference

RL78/G24 User’s Manual: Hardware (R01UH0961)

RL78 Family PMBus Master module Software Integration System (R01AN7188)

RL78 Family PMBus Slave module Software Integration System (R01AN6985)

Renesas Flash Programmer V3.12 Flash memory programming software User’s Manual (R20UT5312)

(The latest version can be downloaded from the Renesas Electronics website.)

Technical Updates/Technical News

(The latest information can be downloaded from the Renesas Electronics website.)

SMBus, PMBus Standards

PMBus Specification Rev. 1.4 Part I
PMBus Specification Rev. 1.4 Part II
System Management Bus (SMBus) Specification Version 3.2

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Aug. 28, 2023	-	First edition issued
1.01	Apr. 18, 2024	-	Added 4.4 Feature Overview (PMBus Master)

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

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

1. Precaution against Electrostatic Discharge (ESD)

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

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

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

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

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

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

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

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

Notice

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

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

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

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

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

Trademarks

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

Contact information

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