Energy Harvesting Launch Operation Demo using Quick Start Mode Enabling Effective MCU Utilization during Charging Time for RE01 256KB Group

Summary

This application note provides sample code to demonstrate energy harvesting and RE01 Family's unique quick-start mode using the EK-RE01 256KB (Evaluation Kit RE01 256KB), a development kit for the RE01 256KB Group.

In this sample code, power is supplied from an energy harvesting control circuit (EHC) powered by an energy harvesting element (solar panel/thermoelectric generator). By using the power generated by EHC, the MCU is able to maintain continuous MIP LCD (SMIP) operation.

The device also has a quick start function that is unique to the EHC circuit. This function enables the operation before the secondary battery is fully charged. In this demo, SMIP display is started during the quick start period.

Please refer to the project included in the package for the sample code.

Target Device

RE01 256KB Group

Note

When applying the sample code covered in this application note to another microcomputer, modify the code according to the specifications for the target microcomputer and conduct an extensive evaluation of the modified program.

Related Document

RE01 1500KB, 256KB Group Getting Started Guide to Development Using CMSIS Package (R01AN4660).

Contents

1. What is Quick Start for Energy Harvesting Circuit for RE Family? 3

2. Specifications 3
2.1 Description of Project 3
2.2 Overview 3
2.3 Pins Used 5
2.4 Procedure for Operating Sample Code 6
2.5 Jumper settings and power generator element connections 7

3. Conditions for Checking Operation 9

4. Description of Software 10

5. Specifications of Driver APIs 12

6. Usage Notes 12
6.1 VCOM and SCS pin for MIP LCD 12
6.2 DMAC Interrupts 12
6.3 Setting DMAC Transfer Source and Transfer Destination Addresses 13

7. Troubleshooting 14
7.1 Occurrence of Build Error 14
7.2 Occurrence of HardFault Error when API of CMSIS Driver Is Called 14
7.3 Peripheral Function Fails to Operate when API Is Called 14
7.4 Normal API Return Value But No Pin Output from Peripheral Function 14
7.5 Peripheral Function’s Input or Output Does Not Operate as Expected 14

8. Sample Code 14

9. Reference Documents 15

Revision History 16

1. What is Quick Start for Energy Harvesting Circuit for RE Family?

Quick start is a function that is unique to EHC circuit in RE Family. This function enables the users to immediately use the device before the secondary battery is fully charged.

Quick start mode is especially useful when secondary battery is used together with EHC circuit. Depending on the battery size, secondary battery charging from solar cell or other energy harvesting elements may take up to several days. This function enables the users to skip this waiting time and to use the device as soon as the power generated by the power generating element is available.

During the battery charging period, the product can also be easily mistaken as a broken product because the users cannot see the status of the charging. By using quick start mode, the MCU can notify the users about the status of the charging. In this demo, the status of the secondary battery charging is shown in SMIP during the quick start period. The timing of the quick start mode is described in Figure 1.1.

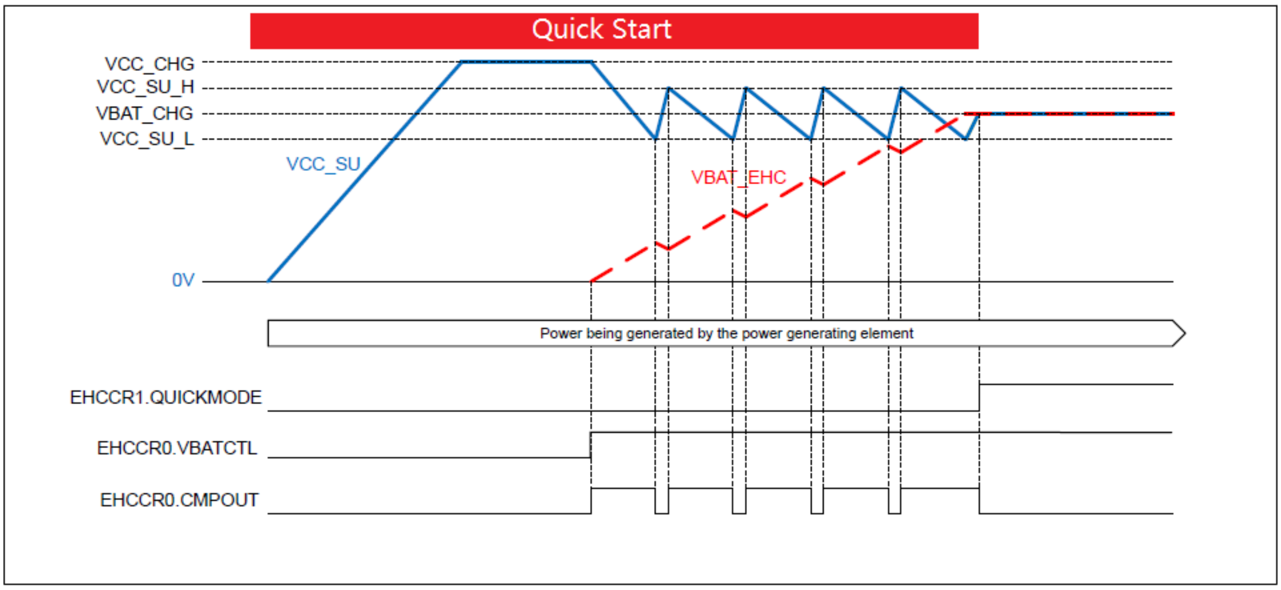


Figure 1.1 Quick Start Timing

2. Specifications

2.1 Description of Project

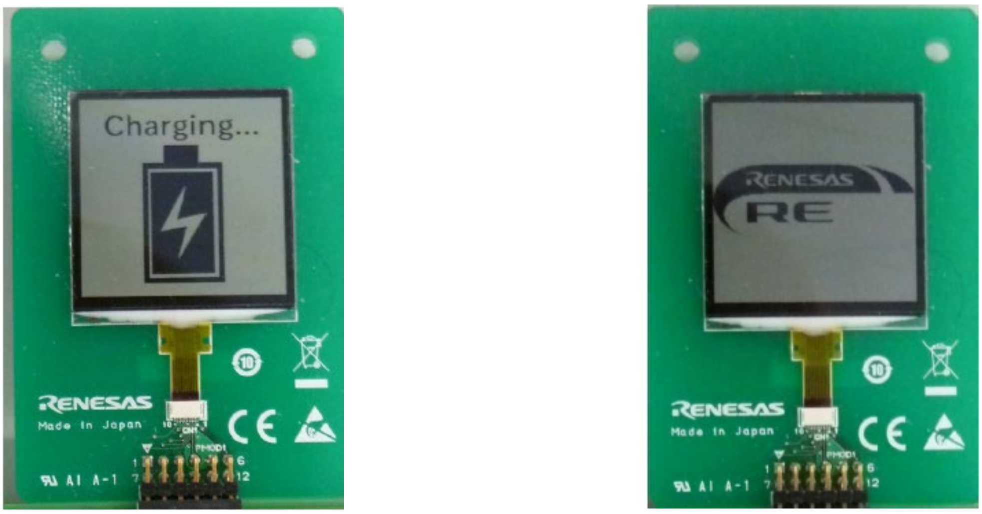
A sample code project "r01an5406_re_ehc_quickstart_smip.zip" is delivered with this application note.

r01an5406_re_ehc_quickstart_smip.zip is a project which operation has been confirmed on the Evaluation Kit RE01 256KB Board. This project is configured to match the settings of RE01 256KB (R7F0E01182CFP) mounted on the Evaluation Kit RE01 256KB Board. When using another RE01 256KB device, change the device settings in the project to those of the target device.

2.2 Overview

By using this sample code, the board can be powered up by an energy harvester (solar panel/thermoelectric generator. The program will send image data to SMIP display, depending on the status of secondary battery charging progress:

- While waiting for the secondary battery to finish charging, the program sends image data to SMIP. The images used while waiting for secondary battery charging are shown in Figure 2.1.



(a) during charging in quick start mode (b) after charging is completed

Figure 2.1 SMIP Display during Quick Start Up Sequence

- After the secondary battery is charged, the program goes to main.c and SMIP alternately shows the original image and color-inverted image. The images are shown in Figure 2.2.

(a) Original Image (b) Color-inverted Image

Figure 2.2 RE Logo in SMIP Display

The overview of the functions used in this program is shown below.

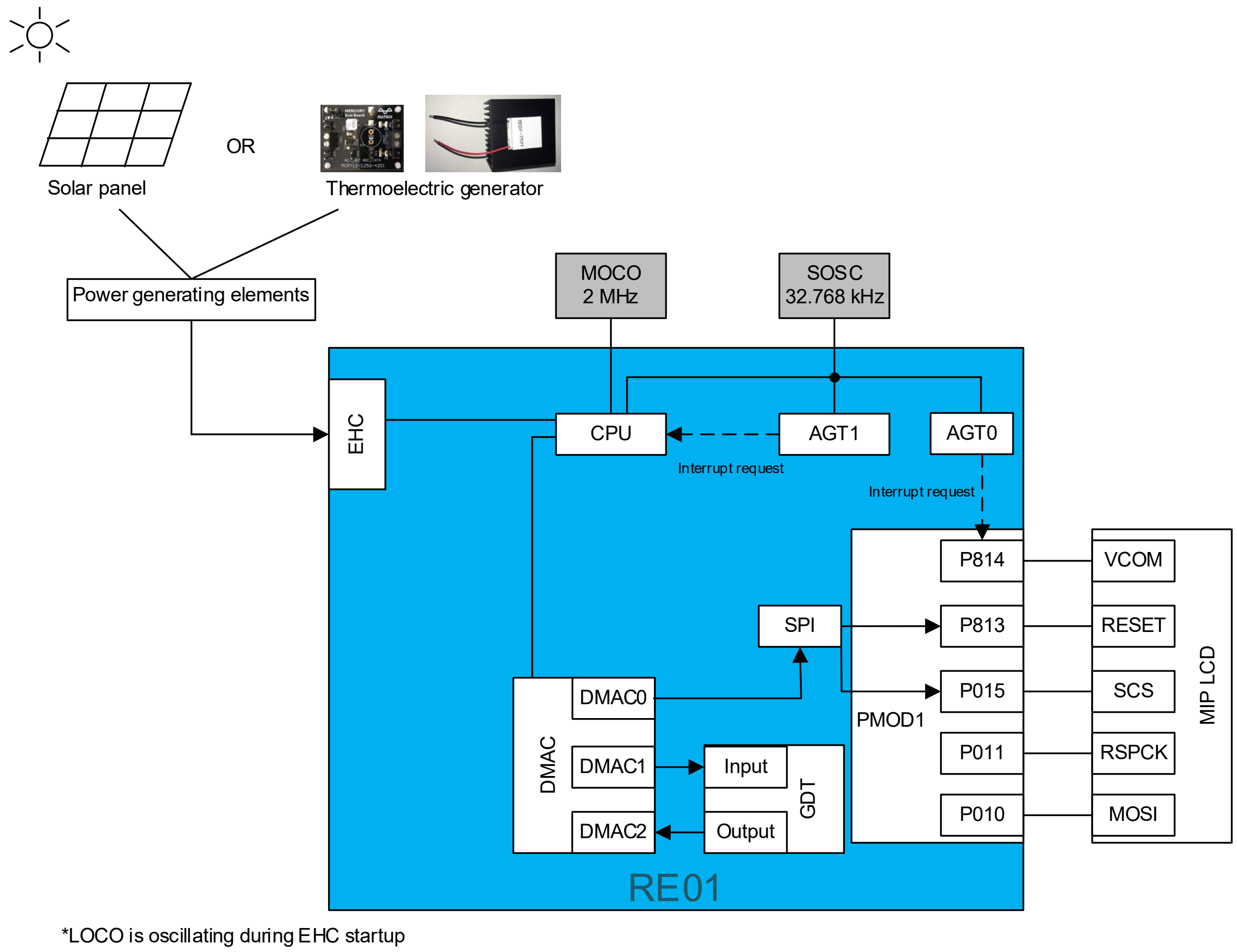


Figure 2.3 Program Overview

2.3 Pins Used

The pins used by the sample code are shown below.

Pin Used	Purpose of Use
P010	MOSIA_B (SPI)
P011	RSPCKA_B (SPI)
P015	SCS (SPI)
P813	RESET (SPI)
P814	VCOM (SPI)

2.4 Procedure for Operating Sample Code

Follow the steps described below to operate the sample code. Refer to the startup guide for more details of how to build and download the program.

1. Build the sample code
2. Follow the board setting shown in Figure 2.4 before downloading the program *1
3. Connect the debugger and download the program to the MCU.
4. Disconnect the debugger.
5. Change the board setting before operating in energy harvesting mode *1

   In this sample code, users can select either a solar panel or thermoelectric generator harvester as the power generation element.

   a. When using solar panel

      Evaluation Kit RE01 256KB Board and a solar panel are required. Connect the required components by following the setting shown in Figure 2.5.

   b. When using thermoelectric generator harvester

      Evaluation Kit RE01 256KB Board, thermoelectric generator harvester (sold separately), and MERCURY Eval Board (sold separately) are required. Connect the required components by following the setting shown in Figure 2.6

6. Start EHC operation

   Sample program will start when power is supplied to the VSC_VCC pin from the power generation elements.

7. Restarting EHC operation

   After the charging of the secondary battery is completed, the program transitions to a steady-state operation period and continuously inverse the image shown in the SMIP display. If the power supply to the MCU is interrupted and the secondary battery does not have enough voltage to maintain operation, the operation of the MCU is intentionally stopped. The MCU can automatically restart as soon as the power from the power generation elements is re-supplied to the VSC_VCC pin. *2

Note: *1 The board setting and connection during program download, energy harvesting mode operation with solar panel, and energy harvesting mode operation with thermoelectric generator harvester are different. Be sure to follow the board setting described for each respective mode.

*2 The EHC circuit starts initial operation when there is a voltage difference between the VCC and VCC/IOVCC pins. Therefore, if the discharging process of the charge on the VCC/IOVCC pins takes a long time due to components connected to the VCC/IOVCC pins, etc., it is not possible to create a voltage difference to start the initial operation of the EHC circuit. For this reason, design the circuit so that the VCC/IOVCC pins are quickly discharged. The evaluation kit has a discharge switch (SW7 of the RE01 256KB) that can be used manually for quick discharge at the prototype stage.

2.5 Jumper settings and power generator element connections

・Fig. 2.4 shows the board setting when downloading the program.
・Fig. 2.5 shows the board setting during energy harvesting operation (using solar panel)
・Fig. 2.6 shows the board setting during energy harvesting operation (using thermoelectric generator harvester)

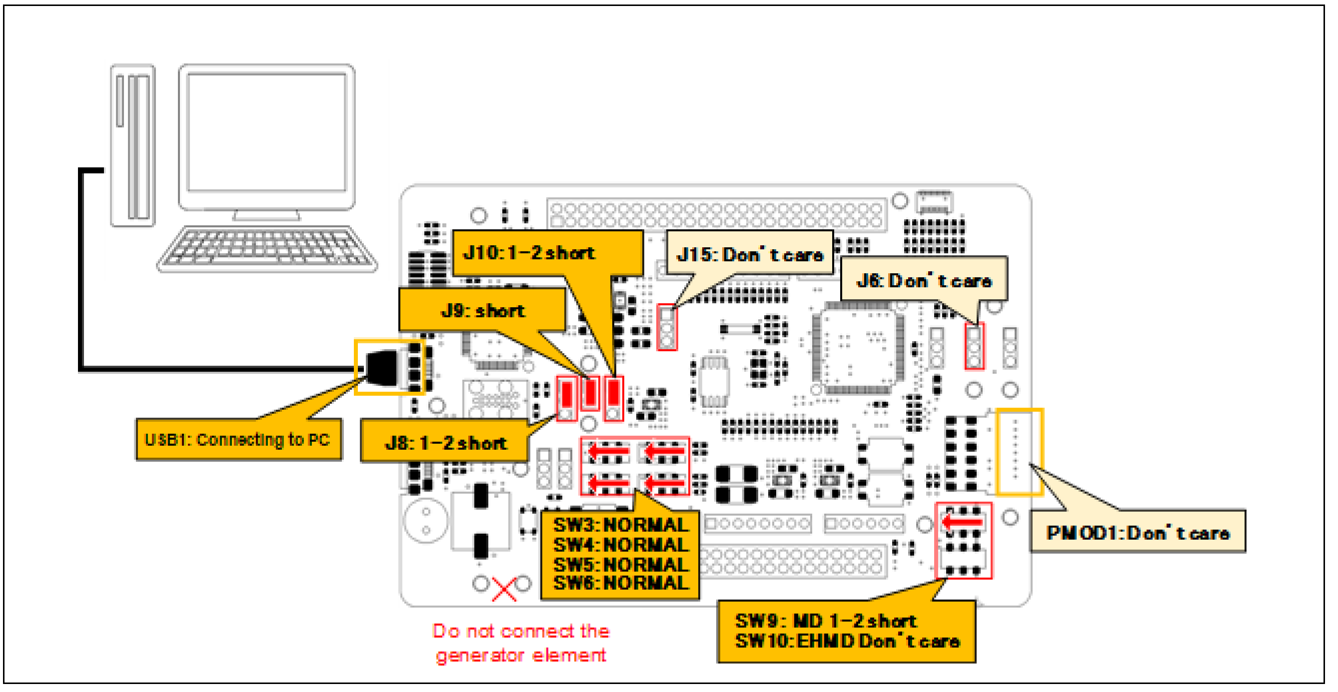


Figure 2.4 Board setting when downloading the program

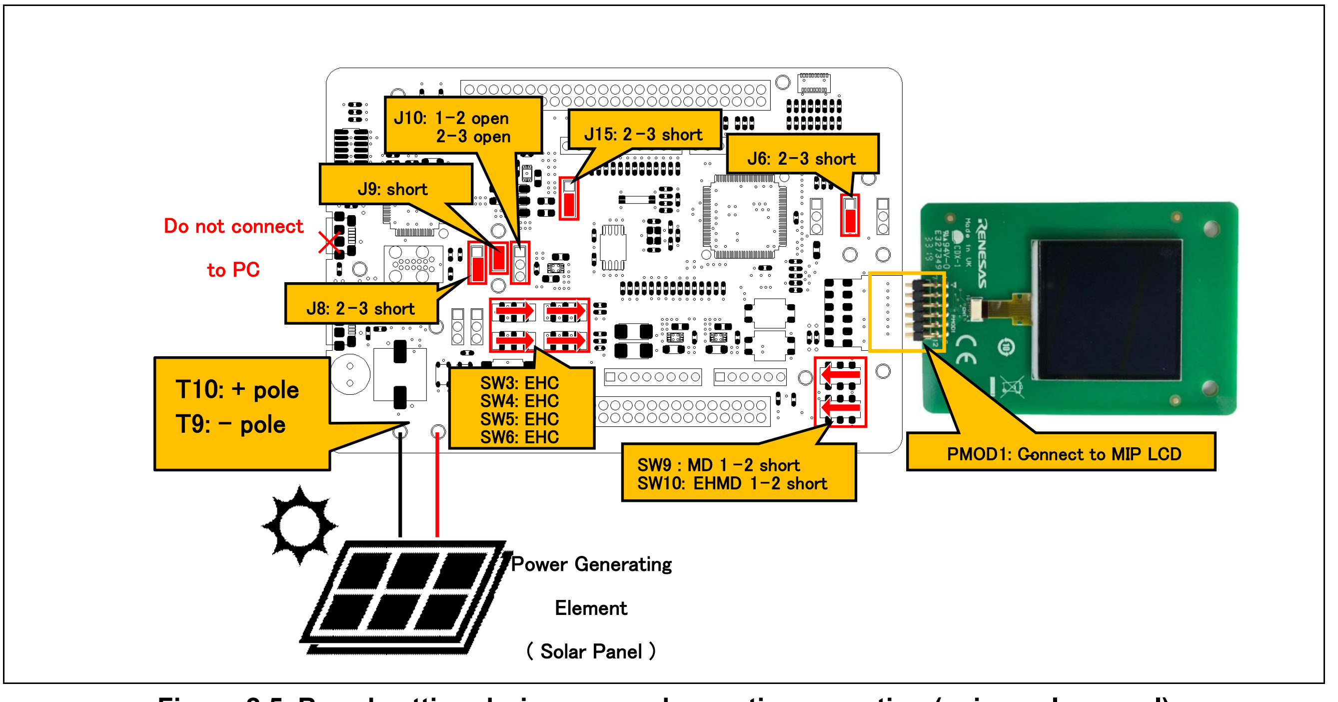


Figure 2.5 Board setting during energy harvesting operation (using solar panel)

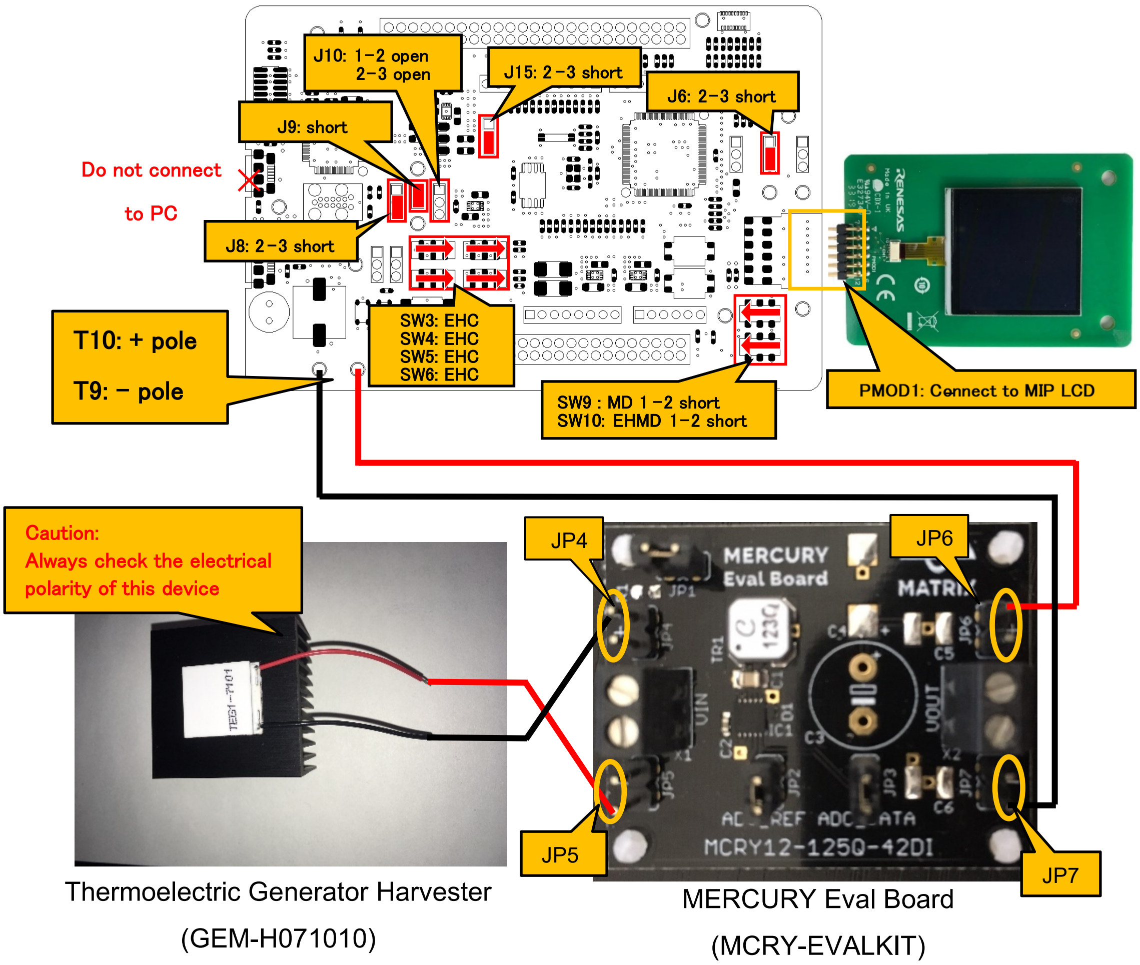


Figure 2.6 Board settings during energy harvesting operation (using thermoelectric generator harvester)

This sample code can use a thermoelectric generator as a power generation element. A thermoelectric generator is a device that uses temperature difference to generate electricity. In this demo, the power is harvested from the temperature difference between human body and room temperature. To power up the MCU, place a finger to the thermoelectric generator harvester.

To run the demonstration, connect the MERCURY Eval Board, thermoelectric generator harvester, and Evaluation Kit RE01 256KB Board as shown in Figure. 2.6. Thermoelectric generator harvester harvests the power generated from temperature difference between human body and room temperature. MERCURY Eval Board is used to increase the voltage generated by the thermoelectric generator harvester to a level that can support Evaluation Kit RE01 256KB Board operation.

Caution: Always check the electrical polarity of the cables when connecting the thermoelectric generator harvester to MERCURY Eval Board. In the device used this demonstration (GEM-H071010), the red wire is the negative pole and the black wire is the positive pole. Make sure to connect the components according to its respective correct electrical pole.

3. Conditions for Checking Operation

The operation of the sample code delivered with this application note has been checked under the following conditions (Table 3-1).

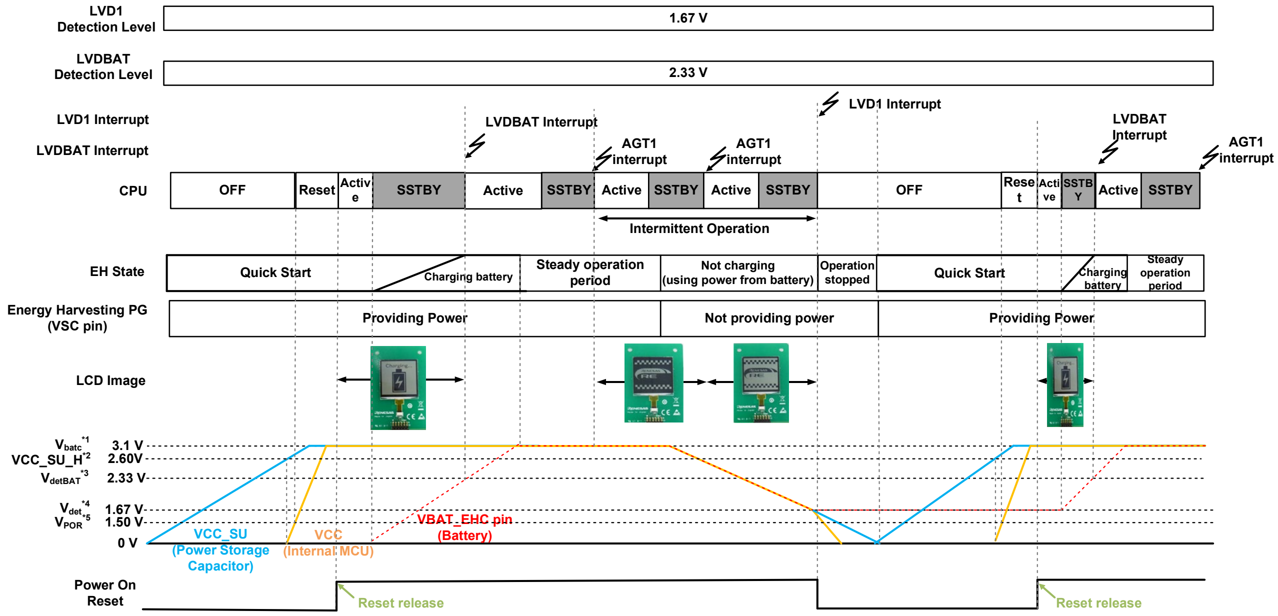
Table 3-1 Conditions for Checking Operation

Item		Description
Microcontroller used		R7F0E01182CFP (RE01 256KB Group)
Operating frequency		ICLK: MOCO 2 MHz PCLKB: MOCO 2MHz
Operating voltage		• 3.3 V
Target board		Evaluation Kit RE01 256KB Board
Display		SMIP display (RTK70E015DB00000BE) *Included in the Evaluation Kit
Solar energy harvester		Solar Panel (AM-1815CA) *Included in the Evaluation Kit
Thermoelectric generator harvester		Thermoelectric generation element (GEM-H071010)
		MERCURY Eval Board (MCRY-EVALKIT)
Development environment	GCC	e^2 studio 2020-07 made by Renesas Electronics
	IAR	IAR Embedded Workbench for ARM Version 8.40.2.22891
C compiler	GCC	GCC ARM Embedded V6.3.1.20170620 GNU 6-2017-q2-update
	IAR	IARC/C++ Compiler for ARM Version 8.40.2 or later
Version of I/O header		Rev. 1.00
Version of CMSIS driver package		Rev. 1.00

4. Description of Software

This sample code demonstrates EHC quick start up and operation powered by a solar panel. In EH mode, users can choose to connect 2.4 V, 2.5 V, 2.6 V, 2.7 V, 2.8 V, 2.9 V, 3.0 V, and 3.1 V secondary battery or super capacitor. Figure 4.1 shows the timing chart of this sample code when using 3.1 V secondary battery.

EH also has an overcharge protection circuit to prevent overcharging of the secondary battery connected to the VBAT_EHC pin and the storage capacitor connected to the VCC_SU pin by the power generating element connected to the VSC_VCC pin. In this sample code, the threshold voltage for charging protection of VBAT is 3.070 V (when 3.1 V VBAT charging voltage is used).



Notes:
*1 V_{batc}: Secondary battery (VBAT) Charging Voltage
*2 VCC_SU_H: Startup threshold voltage at the time of starting up the energy harvest mode
*3 V_{detBAT}: Voltage detection level for LVDBAT (voltage monitoring BAT) circuit
*4 V_{det}: Voltage detection level for LVD1 (voltage monitoring 1) circuit
*5 V_{POR}: Voltage detection level for power-on reset circuit

Figure 4.1 Timing Chart for EHC Operation (using 3.1 V secondary battery)

The sample code performs the following operations.

- When solar panel is connected to the board, the secondary battery charging period starts. During secondary battery charging period, SMIP shows the status of the charging (Figure 2.1).
- After secondary battery charging is finished, the program goes to main file and sends image data to SMIP. The display is toggled between original image and color-inverted image (Figure 2.2)

Figure 4.2 shows the program flow during the secondary battery charging period. Figure 4.3 shows the program flow after going to main (after secondary battery charging period ends).

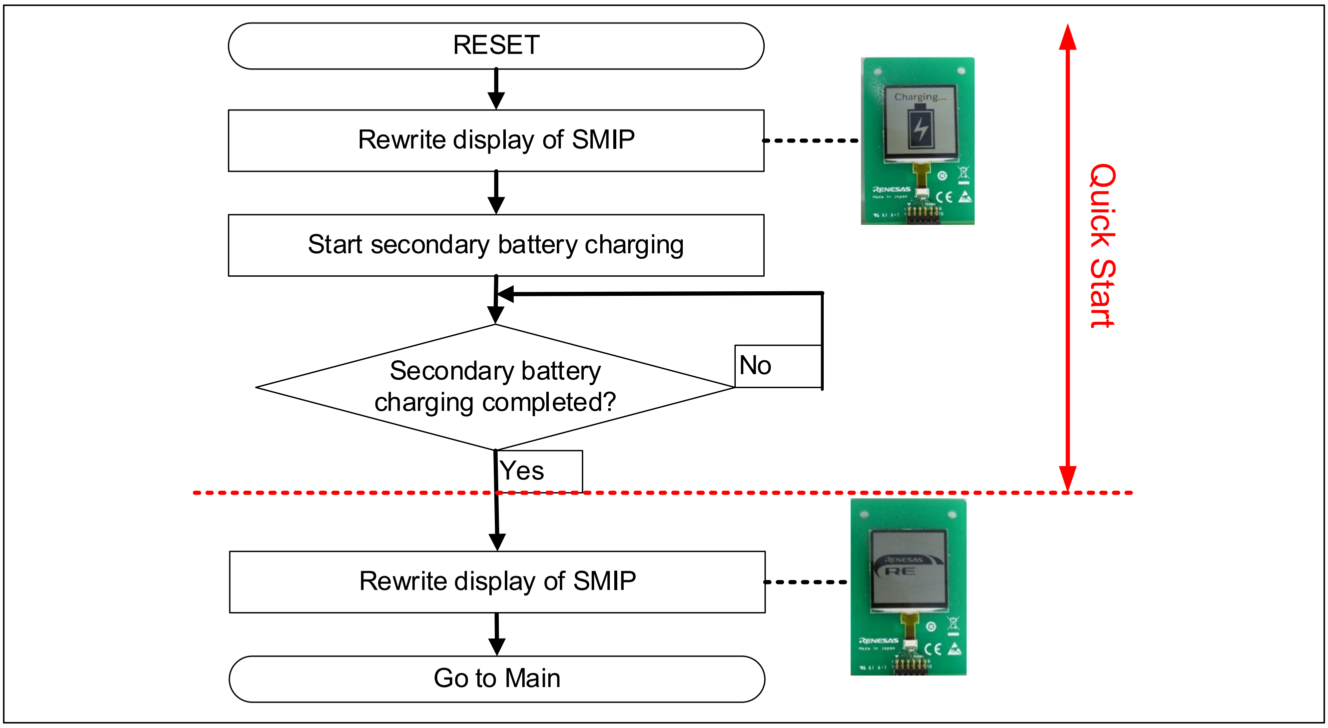


Figure 4.2 Program Flow during Secondary Battery Charging Period

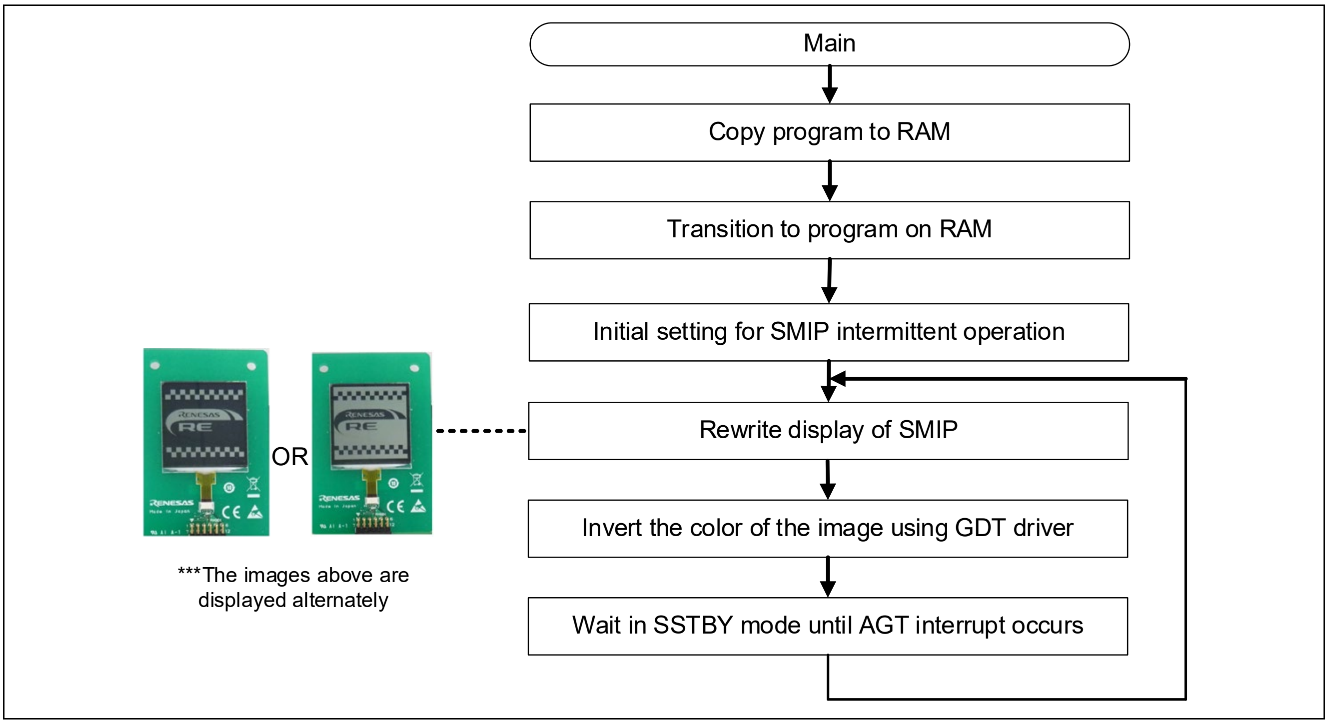


Figure 4.3 Program Flow after Secondary Battery Charging Period Ends

5. Specifications of Driver APIs

The specifications of drivers in the CMSIS driver package can be found in the directory shown below. Refer to the specifications and notes for each driver inside the folder.

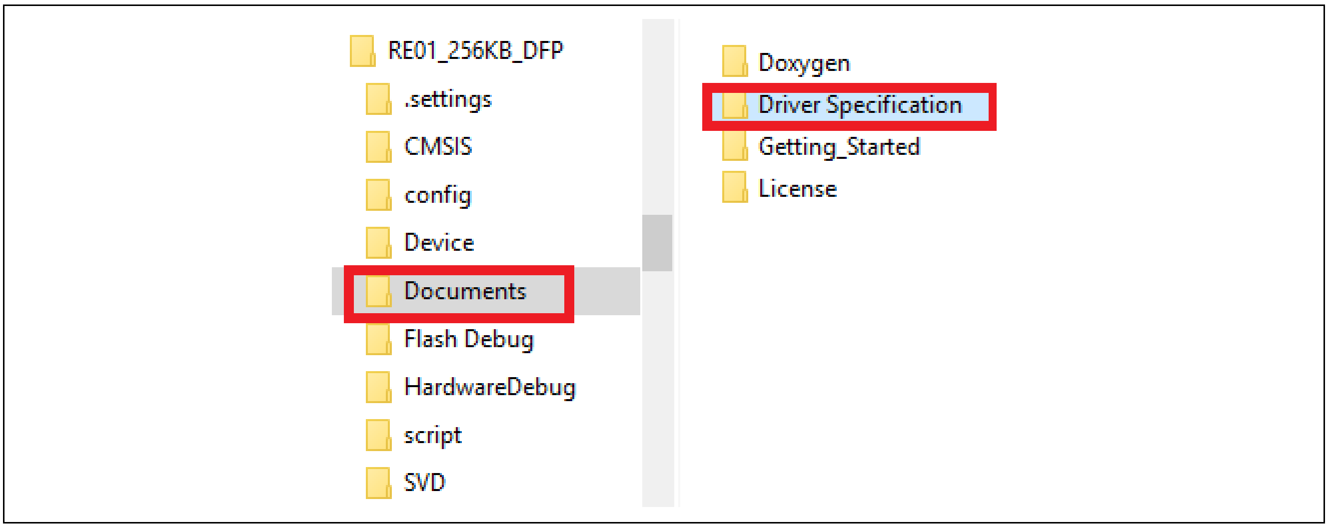


Figure 5.1 Location of Driver Specification

6. Usage Notes

Only the main notes related to the DMAC and MIP LCD pin setting are introduced in this chapter. Note that not all notes are given here.

For other notes, see the external specification document described in "5. Specifications of Driver APIs".

6.1 VCOM and SCS pin for MIP LCD

When using serial MIP LCD, VCOM pin output level must be set to high before setting SCS pin to high. Also note that waiting time is required when changing VCOM output level.

For more information about VCOM waiting time, see TN0181ANVNANN-*N*03 published by Kyocera.

6.2 DMAC Interrupts

When data transfer of the specified count is complete (DMCRA register), an interrupt is signaled to the CPU. There are 2 types of DMA activation sources: software trigger and interrupt requests from peripheral modules/trigger from external interrupt input pins.

To use interrupt requests from peripheral modules as DMA activation source, register it to NVIC in r_system_cfg.c and set the interrupt request for DMA activation source in DELSRn register.

Figure 6.1 shows an example of registering interrupts to the NVIC. Figure 6.2 shows an example of enabling DMACn_INT interrupt.

```
...
#define SYSTEM_CFG_EVENT_NUMBER_PORT_IRQ0
        (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)        /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_DMAC0_INT
        (SYSTEM_IRQ_EVENT_NUMBER0)                 /*!< Numbers 0/4/8/12/16/20/24/28 only */
#define SYSTEM_CFG_EVENT_NUMBER_DTC_COMPLETE
        (SYSTEM_IRQ_EVENT_NUMBER_NOT_USED)         /*!< Numbers 0/4/8/12/16/20/24/28 only */
...
```

Figure 6.1 Example of Registering Interrupts to NVIC

```
interrupt_init()
{
      …

     /*set DMAC source: SPI1_SPTI*/
    ICU->DELSR0_b.DELS=0x9F;

   /*DMAC transfer end interrupt*/
   /*Set priority*/
   R_NVIC_SetPriority(SYSTEM_CFG_EVENT_NUMBER_DMAC0_INT,3);

   /*Set IRQ channel 0 for DMAC0_INT interrupt*/
   R_SYS_IrqEventLinkSet(SYSTEM_CFG_EVENT_NUMBER_DMAC0_INT, 0x02, dmac0_int_callback);

   /*Clear interrupt flag*/
   R_SYS_IrqStatusClear(SYSTEM_CFG_EVENT_NUMBER_DMAC0_INT);

   /*Enable NVIC_ISER bit*/
   R_NVIC_EnableIRQ(SYSTEM_CFG_EVENT_NUMBER_DMAC0_INT);

     …
}

main()
{
    …
   interrupt_init();
      …
}

/*******************************************************************************************************************
* callback function
*******************************************************************************************************************/
static void dmac0_int_callback)(void)
{
    /* Write the code for DMAC0 transfer end interrupt below */
}
```

Figure 6.2 Example of Enabling DMAC Interrupt

6.3 Setting DMAC Transfer Source and Transfer Destination Addresses

Specify transfer source and transfer destination addresses in DMSAR and DMDAR register. The address alignment in these registers must match the transfer data size value selected in SZ bits in DMTMD register.

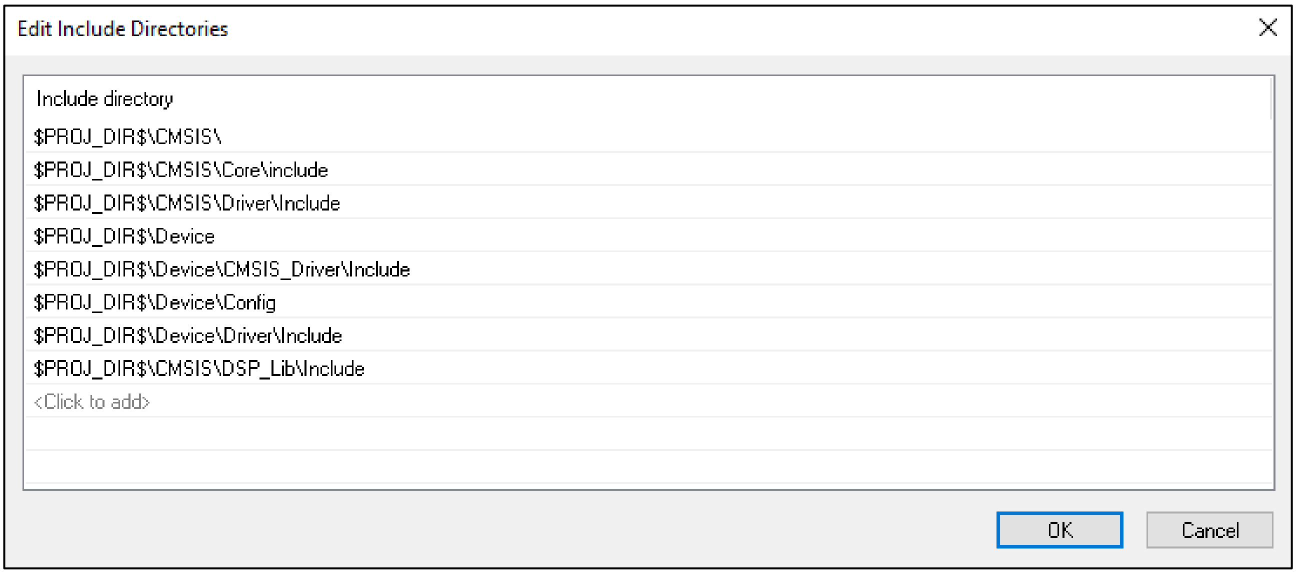
7. Troubleshooting

7.1 Occurrence of Build Error

A-1) Have the include directories been specified correctly?

When using EWARM, we recommend that the include directories be specified as shown in the example below.

The include directories can be specified from IDE Options [C/C++ Compiler] → [Preprocessor].



7.2 Occurrence of HardFault Error when API of CMSIS Driver Is Called

A) The API has possibly not been copied to RAM.

Before calling an API function that was assigned to RAM, make sure that it has been copied to RAM by the R_SYS_CodeCopy function. For details, refer to the related document No. R01AN4660.

7.3 Peripheral Function Fails to Operate when API Is Called

A) Has the API been set up correctly?

Check the API's return value to see if an error has occurred.

In particular, errors are often caused by problems related to interrupts not being set in r_system_cfg.h. For details, refer to the related document No. R01AN4660.

7.4 Normal API Return Value But No Pin Output from Peripheral Function

A) Are the pin settings correct?

Check to make sure the pins have been set up correctly by the functions in pin.c.

For details, refer to the related document No. R01AN4660.

7.5 Peripheral Function's Input or Output Does Not Operate as Expected

A) Check to make sure the VOCR register has been set up correctly before making the initial settings for peripheral functions.

For details, refer to the related document No. R01AN4660.

8. Sample Code

Sample code can be downloaded from the Renesas Electronics website.

9. Reference Documents

User's Manual: Hardware

RE01 Group User's Manual: Hardware

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

Technical Update/Technical News

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

User's Manual: Development Tools

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

Website and Support

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

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

Revision History

Rev.	Date	Description	
		Page	**Summary**
1.00	Aug. 25, 2020	—	First edition issued

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.

Corporate Headquarters

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

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/.