致尊敬的顾客

关于产品目录等资料中的旧公司名称

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：http://www.renesas.com

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（http://www.renesas.com）

【业务咨询】http://www.renesas.com/inquiry

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. 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.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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.
5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
6. 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.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "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.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. 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.
9. 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 system 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. 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.
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.

M16C/26A、29 群

CRC 计算电路

1. 要点

本篇资料介绍如何使用 M16C/26A 与 M16C/29 群单片机的 CRC 计算电路，并且说明了一种使用 CRC 计算电路进行通信错误的检测方法。

2. 说明

本篇资料，适用于 M16C/26A、M16C/29 群单片机。

本篇资料中的参考例程也适用于 M16C 族产品中与 M16C/26A、M16C/29 群具有相同 SFR（特殊功能寄存器）定义的产品。

由于 M16C 系列产品中有些功能会有所改进，请参看用户手册。如果使用本篇资料中所列功能时，请仔细检查每一步操作。

3. 规格

CRC（CRC 冗余校验）是基于 CRC 码的数据块校验方法，该方法用于通信过程中的错误检测。M16C/26A 与 M16C/29 群单片机使用内置的 CRC 计算电路，从而可以生成 CRC 校验码。

3.1 CRC 计算与 CRC 校验码

在 CRC 计算过程中将通信数据当作数值处理，这些数值被称为信息多项式。CRC 计算就是用信息多项式除以固定的生成多项式（模 2 除），CRC 的计算示例如图 1 所示。以下两个表达式可以作为生成多项式使用。

- CRC-CCITT:

$$X^{16}+X^{12}+X^{5}+1$$

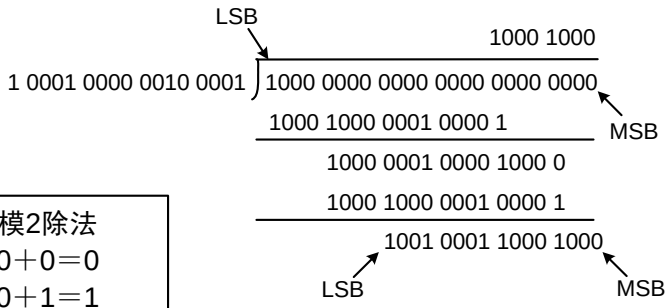
- CRC-16:

$$X^{16}+X^{15}+X^{2}+1$$

最后计算所得的余数就称为 CRC 校验码。

生成多项式: 1 0001 0000 0010 0001 ($X^{16}+X^{12}+X^5+1$)
信息多项式: 1000 0000 (由最低位发送和接收01h)

CRC校验码: 1001 0001 1000 1000 (因为从LSB开始，→1189h)



模2除法
0＋0＝0
0＋1＝1
1＋0＝1
1＋1＝0
－1＝1

生成多项式: 1 1000 0000 0000 0101 ($X^{16}+X^{15}+X^2+1$)
信息多项式: 1000 0000 (由最低位发送和接收01h)

CRC校验码: 1000 0011 0000 0011 (因为从LSB开始，→C0C1h)

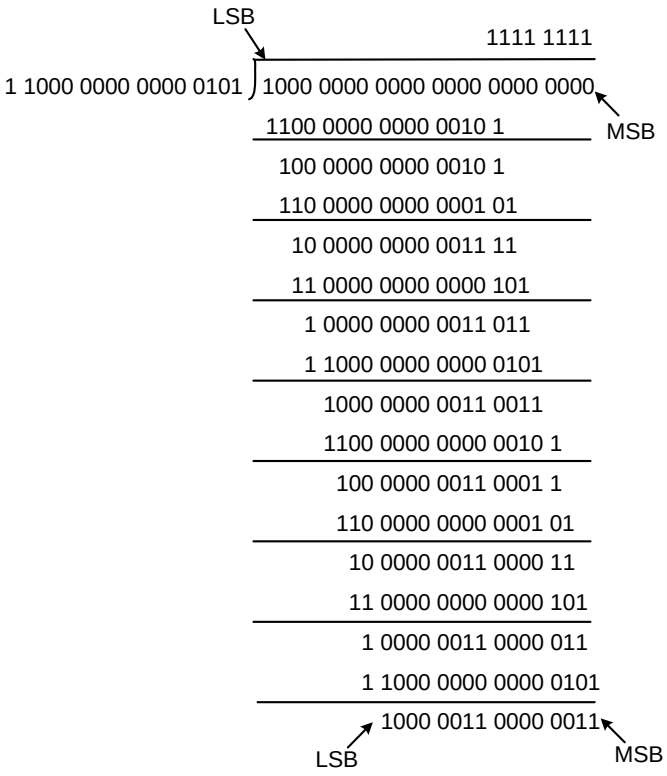


图 1. CRC 计算示例图

3.2 CRC 计算电路的硬件配置

虽然 CRC 的计算从原理上看是在做除法操作，但在硬件上是通过移位寄存器和异或电路来生成 CRC 校验码的，CRC 的计算电路示例如图 2 所示。

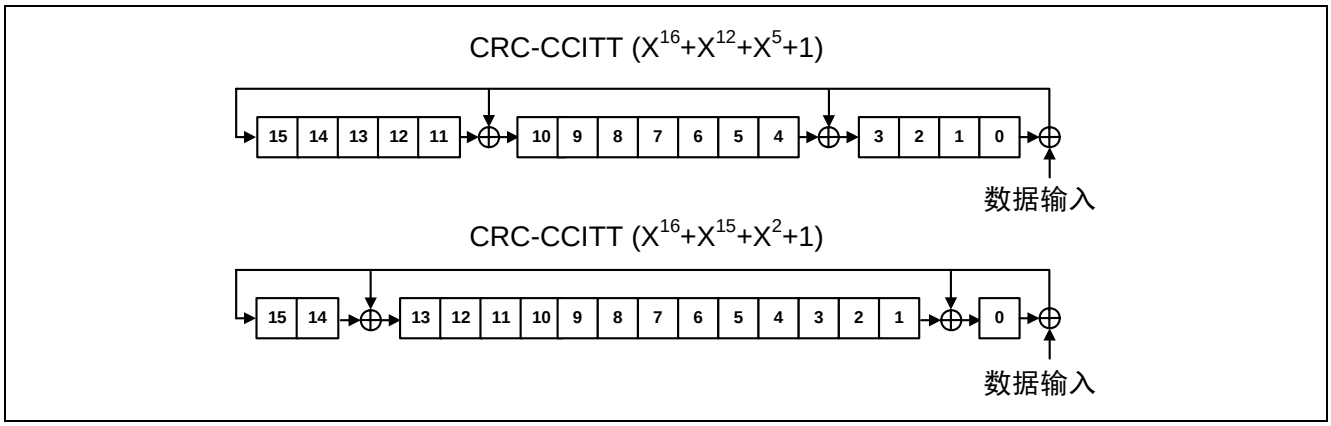


图 2. CRC 计算电路示例图

3.3 CRC 计算中的内部操作

使用 M16C/26A 与 M16C/29 群单片机的 CRC 计算方法是：当数据以字节方式写入 CRC 输入寄存器后，根据新输入的数据与 CRC 数据寄存器中的内容产生 CRC 校验码，并将计算结果存放在 CRC 数据寄存器中。因此，当传送或接收多字节数据时，传送/接收的数据依次写入 CRC 输入寄存器，而 CRC 数据寄存器最终的内容就可作为此次块校验的 CRC 校验码。

除此之外，可以通过设置 SFR 监测地址寄存器（CRCSAR）中的监测读允许位或写允许位、UART 发送缓冲寄存器或接收缓冲寄存器的地址选择位实现 CRC 计算监测功能。在监测模式下，CRC 计算与读取 UART 发送缓冲寄存器或接收缓冲寄存器是并行的，无需将发送/接收数据写入 CRC 输入寄存器中。此功能可用于通过 DMA 传送方式读取 UART 发送/接收缓冲寄存器的情况。

以下是发送/接收 12345678h（001 0010 0011 0100 0101 0110 0111 1000b）（LSB 先）并执行 CRC 计算的例子。

3.3.1 CRC-CCITT

（1）向 CRC 数据寄存器写入初始值“0000h”。

（2）向 CRC 输入寄存器写入最先发送/接收的数据“78h”。

（3）CRC 输入寄存器中的值从低有效位（LSB）开始按位依次送入 CRC 数据寄存器，算术操作的执行过程如下所示。

本次操作结束后 CRC 数据寄存器中的内容变成“FFCFh”。

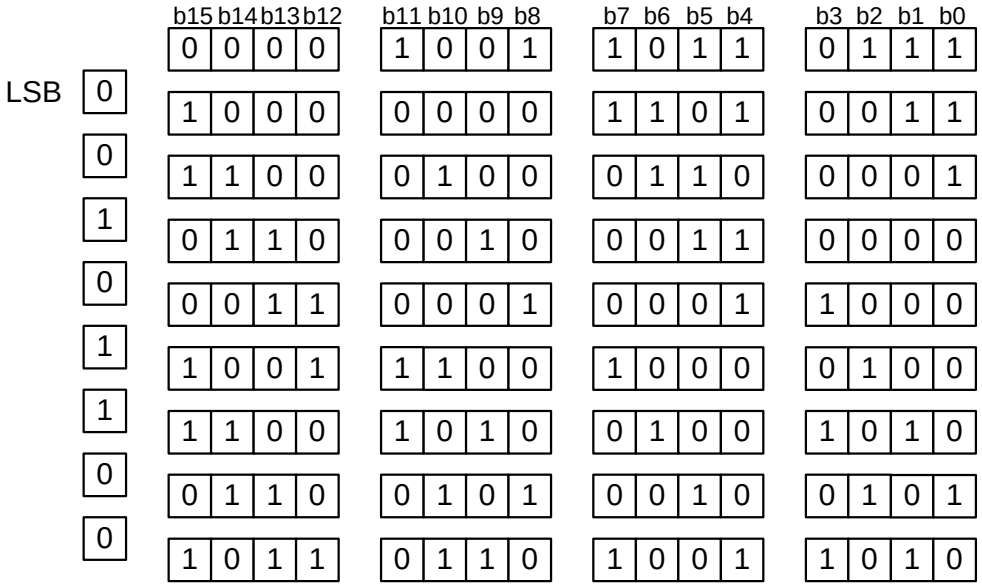
		b15 b14 b13 b12	b11 b10 b9 b8	b7 b6 b5 b4	b3 b2 b1 b0
		0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
LSB	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
	1	1 0 0 0	0 1 0 0	0 0 0 0	1 0 0 0
	1	1 1 0 0	0 1 1 0	0 0 0 0	1 1 0 0
	1	1 1 1 0	0 1 1 1	0 0 0 0	1 1 1 0
	1	1 1 1 1	0 1 1 1	1 0 0 0	1 1 1 1
	0	1 1 1 1	1 1 1 1	1 1 0 0	1 1 1 1

（4）在保持 CRC 数据寄存器内容完整的同时，向 CRC 输入寄存器写入下一个发送/接收的数据“56h”。

本次操作结束后 CRC 数据寄存器中的内容变成“09B7h”。

		b15 b14 b13 b12	b11 b10 b9 b8	b7 b6 b5 b4	b3 b2 b1 b0
		1 1 1 1	1 1 1 1	1 1 0 0	1 1 1 1
LSB	0	1 1 1 1	1 0 1 1	1 1 1 0	1 1 1 1
	1	0 1 1 1	1 1 0 1	1 1 1 1	0 1 1 1
	1	0 0 1 1	1 1 1 0	1 1 1 1	1 0 1 1
	0	1 0 0 1	1 0 1 1	0 1 1 1	0 1 0 1
	1	0 1 0 0	1 1 0 1	1 0 1 1	1 0 1 0
	0	0 0 1 0	0 1 1 0	1 1 0 1	1 1 0 1
	1	0 0 0 1	0 0 1 1	0 1 1 0	1 1 1 0
	0	0 0 0 0	1 0 0 1	1 0 1 1	0 1 1 1

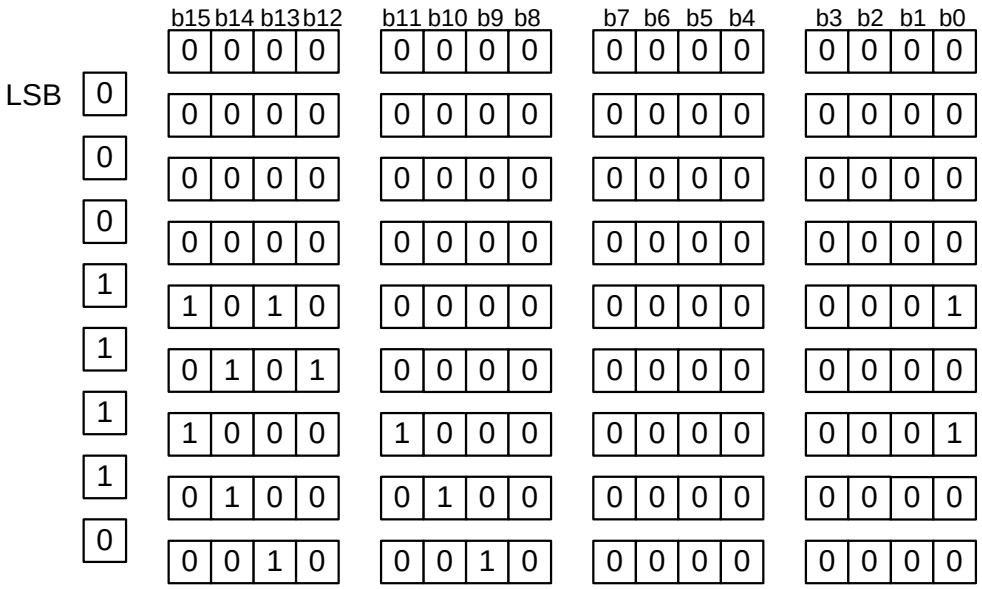
（5）在保持 CRC 数据寄存器内容完整的同时，向 CRC 输入寄存器写入下一个发送/接收的数据“34h”。
本次操作结束后 CRC 数据寄存器中的内容变成“B69Ah”。



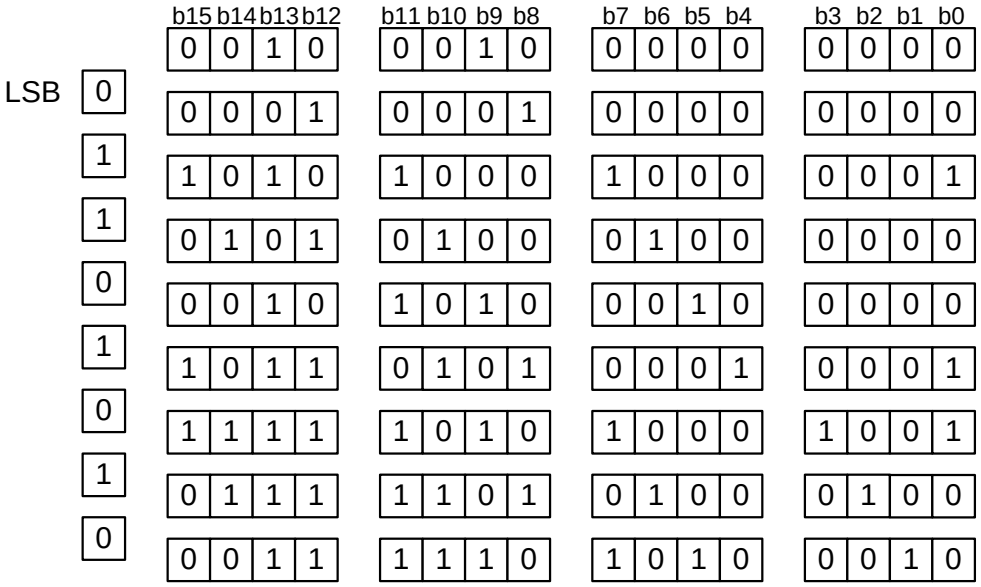
（6）在保持 CRC 数据寄存器内容完整的同时，向 CRC 输入寄存器写入下一个发送/接收的数据“12h”。
本次操作结束后 CRC 数据寄存器中的内容变成“08F6h”。

3.3.2 CRC-16

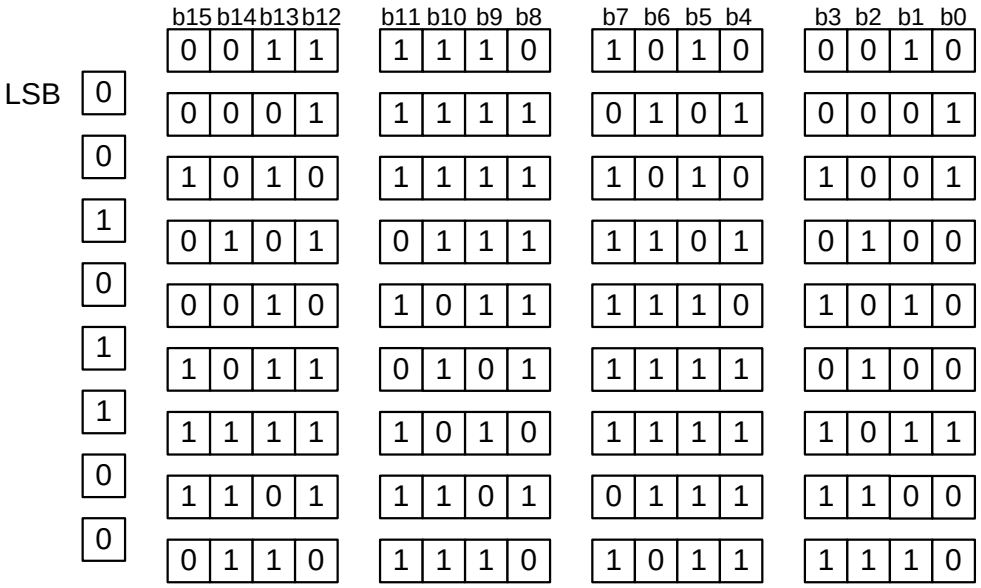
（1）向 CRC 数据寄存器写入初始值“0000h”。
（2）向 CRC 输入寄存器写入最先发送/接收的数据“78h”。
（3）CRC 输入寄存器中的值从低有效位（LSB）开始按位依次送入 CRC 数据寄存器，算术操作的执行过程如下所示。
本次操作结束后 CRC 数据寄存器中的内容变成“2200h”。



（4）在保持 CRC 数据寄存器内容完整的同时，向 CRC 输入寄存器写入下一个发送/接收的数据“56h”。本次操作结束后 CRC 数据寄存器中的内容变成“3EA2h”。



（5）在保持 CRC 数据寄存器内容完整的同时，向 CRC 输入寄存器写入下一个发送/接收的数据“34h”。本次操作结束后 CRC 数据寄存器中的内容变成“6EBEh”。

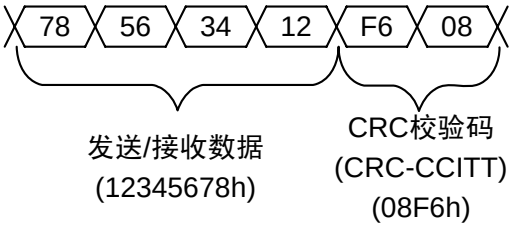


（6）在保持 CRC 数据寄存器内容完整的同时，向 CRC 输入寄存器写入下一个发送/接收的数据“12h”。本次操作结束后 CRC 数据寄存器中的内容变成“7D6Eh”。

3.4 如何使用 CRC

当发送/接收多字节数据时，通过发送/接收数据和 CRC 校验码一起实现通信错误的检测。

由LSB开始发送/接收12345678h (001 0010 0011 0100 0101 0110 0111 1000b)时，
发送/接收的格式为



3.4.1 发送方设置步骤

（1）向 CRC 数据寄存器写入初始值“0000h”。在 CRC 模式寄存器中设定 CRC 多项式和 CRC 模式。

（2）在 CRC 监测地址寄存器中设定发送缓冲寄存器地址作为 SFR 监测地址寄存器的值，同时将写监测允许位置“1”。

（3）将要传送的数据按字节依次写入发送缓冲器。

（4）当所有数据按字节传送完毕时，读取 CRC 数据寄存器的内容（CRC 校验码）并将该校验码写入发送缓冲寄存器中发送。

3.4.2 接收方设置步骤

（1）向 CRC 数据寄存器写入初始值“0000h”。在 CRC 模式寄存器中设定 CRC 多项式和 CRC 模式。

（2）在 CRC 监测地址寄存器中设定接收缓冲寄存器地址作为 SFR 监测地址寄存器的值，同时将读监测允许位置“1”。

（3）每次数据接收完毕，从接收缓冲寄存器读取接收数据。当全部数据按字节读取完毕，接收方的 CRC 数据寄存器的内容应该与发送方的相同。

（4）由于 CRC 校验码已经从发送方传送，用与上面同样的方法读取发送方 CRC 校验码。

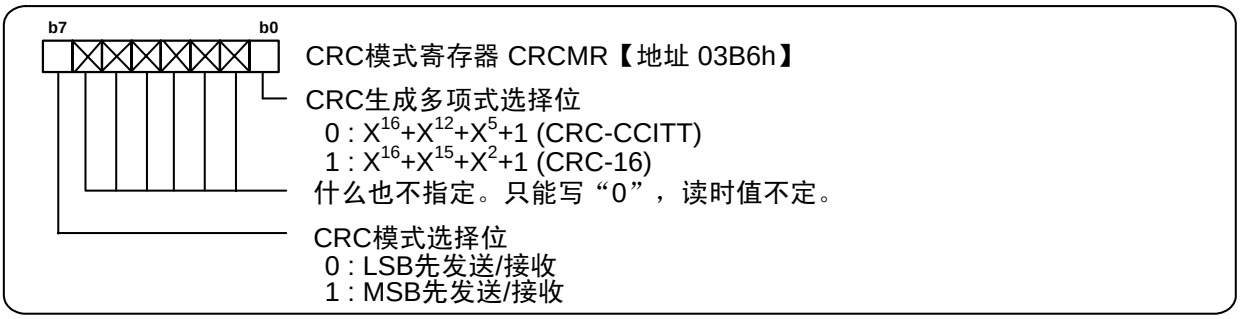
（5）当所有数据包括 CRC 校验码接收完毕时，CRC 数据寄存器的内容应该为“0000h”，表示接收过程正常。如果 CRC 数据寄存器的内容不为“0000h”，表示有通信错误发生。在此情况下，需要请求发送方重传数据。

3.5 寄存器设置

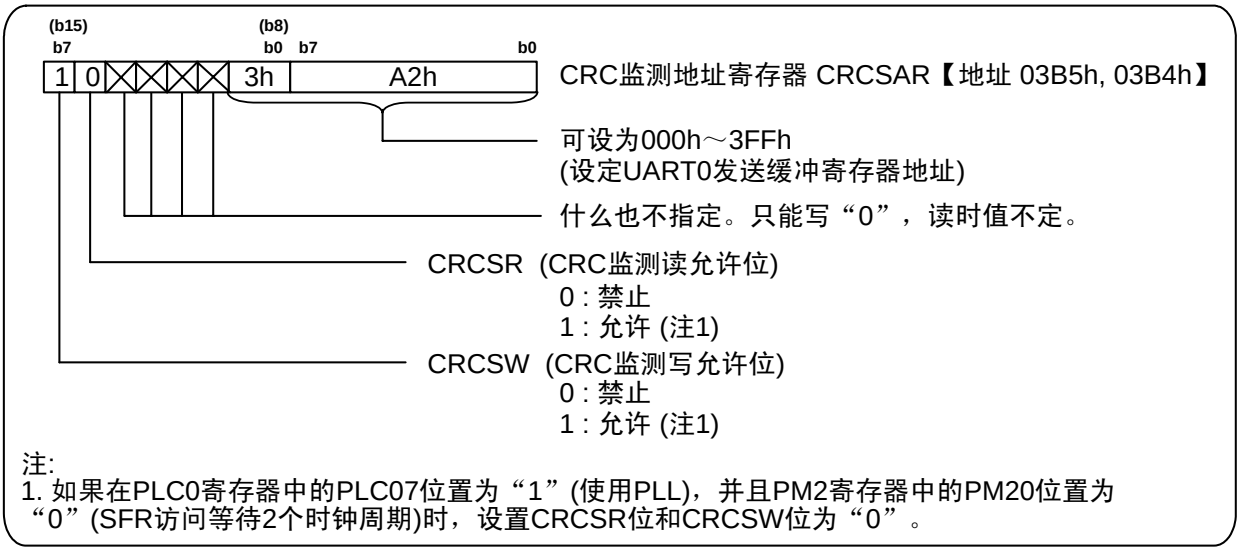
本篇应用说明中给出的操作是基于 M16C/29 群单片机产品，M16C/26A 群产品的寄存器设置步骤请参阅相关硬件手册。

3.5.1 发送方寄存器设置

(1) 设置CRC模式寄存器



(2) 设置CRC监测地址寄存器

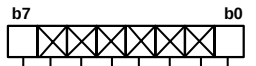


注:
1. 如果在PLC0寄存器中的PLC07位置为“1”(使用PLL)，并且PM2寄存器中的PM20位置为“0”(SFR访问等待2个时钟周期)时，设置CRCSR位和CRCSW位为“0”。

(3) 设置CRC数据寄存器

3.5.2 接收方寄存器设置

(1) 设置CRC模式寄存器



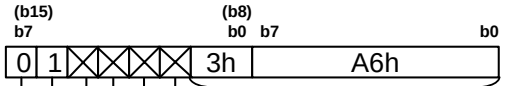
CRC模式寄存器 CRCMR【地址 03B6h】

CRC生成多项式选择位
0 : $X^{16}+X^{12}+X^{5}+1$ (CRC-CCITT)
1 : $X^{16}+X^{15}+X^{2}+1$ (CRC-16)

什么也不指定。只能写“0”，读时值不定。

CRC模式选择位
0 : LSB先发送/接收
1 : MSB先发送/接收

(2) 设置CRC监测地址寄存器



CRC监测地址寄存器 CRCSAR【地址 03B5h, 03B4h】

可设为000h～3FFh
(设定UART0接收缓冲寄存器地址)

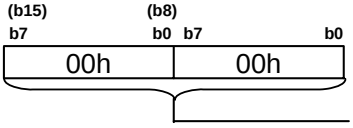
什么也不指定。只能写“0”，读时值不定。

CRCSR (CRC监测读允许位)
0 : 禁止
1 : 允许 (注1)

CRCSW (CRC监测写允许位)
0 : 禁止
1 : 允许 (注1)

注:
1. 如果在PLC0寄存器中的PLC07位置为“1”（使用PLL），并且PM2寄存器中的PM20位置为“0”（SFR访问等待2个时钟周期）时，设置CRCSR位和CRCSW位为“0”。

(3) 设置CRC数据寄存器



CRC数据寄存器 CRCD【地址 03BDh, 03BCh】

CRC计算结果输出

4. 参考例程

以下程序实现在异步串行 I/O 模式下 4 个字节数据的发送/接收，并且将通信数据的 CRC 校验码随后发送出去，参考例程中 CRC 实例的硬件连接如图 3 所示。

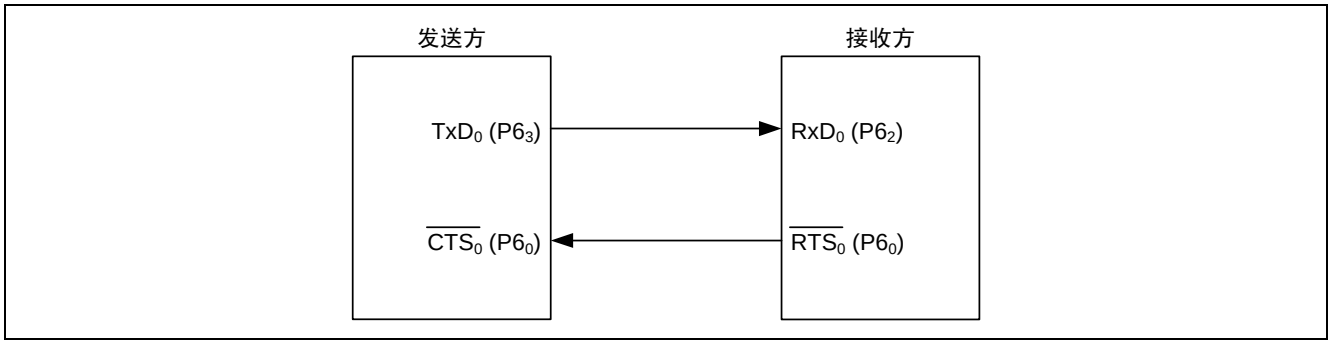


图 3 CRC 实例说明图

4.1 发送方程序

```
/************************************************************/
/*                                                          */
/* M16C/Tiny Series Program Collection                      */
/*                                                          */
/* FILE NAME   : rec05b0005-0102_snd.c                      */
/* CPU         : M16C/29 Group                              */
/* FUNCTION    : CRC Calculation Circuit                    */
/*             (Clock asynchronous serial I/O transfer)     */
/* HISTORY     : 2006.04.13 Ver 1.02                        */
/*                                                          */
/* Copyright (C) 2005. Renesas Technology Corp.             */
/* All right reserved.                                      */
/*                                                          */
/************************************************************/

/************************************************************/
/*    Include File                                          */
/************************************************************/
#include "sfr29.h"     // Special function register header file

/************************************************************/
/*    Function Definition                                   */
/************************************************************/
void init_mcu(void);   // MCU initialize
void init_uart(void);  // UART initialize
void init_crc(void);   // CRC initialize
void crc_snd(void);
void wait_10ms(void);  // Main clock oscillation stable wait routine

/************************************************************/
/*    Define Label                                          */
/************************************************************/
#define PRODUCT_TYPE 0     // 29 group: 0    26A group: 1
#define PIN_TYPE 0         // 80 pin: 0      64 pin: 1 (29 group)
                           // 48 pin: 0      42 pin: 1 (26A group)
#define PLL_ON 0           // 0: not use PLL or PM20 is 1
                           // 1: use PLL and PM20 is 0
```

```
/************************************************************/
/*    Define Const                                          */
/************************************************************/
/* port2_0 - port2_7 data : 0  1  2  3  4  5  6  7  8  9  9 */
unsigned char count_data[11] =
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xD8,0x80,0x90,0x90};

/************************************************************/
/*    Define Variable                                       */
/************************************************************/
unsigned short trans_data[4];
unsigned int i;
unsigned char crc_data_l;
unsigned char crc_data_h;

/************************************************************/
/*    Main Program                                          */
/************************************************************/
void main(void) {
   init_mcu();
   init_uart();
   init_crc();
   crc_snd();
}

/************************************************************/
/*    MCU Initialize Program                                */
/************************************************************/
void init_mcu(void) {
   prcr = 0x03;   // Protect register off

   pm20 = 1;      // Specifying wait when accessing SFR (Note)
                  // (Note) This bit can only be rewritten while the PLC07 bit is
                  // "0" (PLL Off). Also, to select a 16MHz or higher PLL clock,
                  // set this bit to "0" (2 wait).

   wait_10ms();   // Waiting for main clock oscillation stable

   cm2 = 0x00;    // System register2 Initialize
   cm1 = 0x20;    // System register1 Xcin-Xcout: high
   cm0 = 0x08;    // System register0 Xcin-Xcout: high

   prcr = 0x00;   // Protect register on

   #if PRODUCT_TYPE        // Product selection: 26A group
      ifsr2a = 1;          // Interrupt request cause select register2 IFSR2A
                           // <IFSR20>   : Reserved bit (Must be set to "1")
      prcr = 0x04;         // Protect register off
      #if PIN_TYPE         // Port setting
         pacr = 0x01;      // 42pin type
      #else
         pacr = 0x04;      // 48pin type
      #endif
      prcr = 0x00;         // Protect register on
   #else                   // Product selection: 29 group
      ifsr2a = 0;          // Interrupt request cause select register2 IFSR2A
                           // <IFSR20>   : Reserved bit (Must be set to "0")
```

```
      prcr = 0x04;         // Protect register off
      #if PIN_TYPE         // Port setting
         pacr = 0x02;      // 64pin type
      #else
         pacr = 0x03;      // 80pin type
      #endif
      prcr = 0x00;         // Protect register on
   #endif
}

/*******************************************************/
/*    Main Clock Oscillation Stable Wait 10ms Routine     */
/*******************************************************/
void wait_10ms(void) {
   ta0mr = 0x00;     // Set Timer A0 mode register (Timer mode, count source: f1)

   ta0 = 20000-1;    // Setting counter value (10msec @4MHz/2, f1)

   ta0ic = 0x00;     // Clear interrupt request bit

   tabsr = 0x01;     // Timer A0 start counting

   while (ir_ta0ic == 0){   }

   ir_ta0ic = 0;     // Clear interrupt request bit

   tabsr = 0x00;     // Timer A0 stops counting
}

/*******************************************************/
/*    UART Initialize Program                          */
/*******************************************************/
void init_uart(void) {
   u0mr = 0x45;   // UART0 transmit/receive mode register setting
                  // UART mode transfer data 8 bits long
                  // Internal clock select one stop bit
                  // Parity enabled (odd parity)

   u0c0 = 0x00;   // UART0 transmit/receive control register 0 setting
                  // CTS function select
                  // CTS/RTS function enabled
                  // TxD0 pin is CMOS output
                  // Transmission data is output at falling edge of transfer clock
                  // and receive data is input at rising edge LSB first

   ucon = 0x01;   // UART transmit/receive control register 2 setting
                  // UART0 transmit interrupt cause is selected to "Transmission
                  // completed (TXEPT=1)"
                  // CTS/RTS shared pin

   u0brg = 129;   // Setting UART0 bit rate generator (Approx 9600bps @20MHz f1)
   u0c1 = 0x01;   // UART transmit/receive control register 1 setting transmit
                  // enabled

   trans_data[0] = 0x78;
   trans_data[1] = 0x56;
```

```
   trans_data[2] = 0x34;
   trans_data[3] = 0x12;
}

/*******************************************************/
/*    CRC Initialize Program                           */
/*******************************************************/
void init_crc(void) {
   #if PLL_ON
      crcsw = 0;              // CRC snoop on write is disabled
      crcsr = 0;              // CRC snoop on read is disabled

      #else
                              // CRC calculation circuit
      crcps = 0;              // CRC mode polynomial selection bit is selected
                              // to "0" (CRC-CCITT)
      crcms = 0;              // CRC mode selection bit is selected to "0" (LSB
                              // first)

      crcsarl = (char)(&u0tb);    // Setting SFR snoop address
      crcsarh = (char)((unsigned long)(&u0tb) >> 8 );

      crcsw = 1;              // CRC snoop on write is enabled
      crcsr = 0;              // CRC snoop on read is disabled

      crcd = 0;               // CRC data register set to "0"

   #endif
}

/*******************************************************/
/*    CRC Send Program                                 */
/*******************************************************/
void crc_snd(void) {

   for (i=0;i<4;i++) {
      u0tb = trans_data[i];   // Writing transmit data

      while (!ti_u0c1) {
                     // Check & wait the status of UART0 transmit buffer empty flag
      }
   }

   crc_data_l = crcdl;
   crc_data_h = crcdh;
   u0tb = crc_data_l;         // Writing transmit CRC data

   while (!ti_u0c1) {
                     // Check & wait the status of UART0 transmit buffer empty flag
   }

   u0tb = crc_data_h;         // Writing transmit CRC data

   while (!ti_u0c1) {
                     // Check & wait the status of UART0 transmit buffer empty flag
   }
}
```

4.2 接收方程序

```
/************************************************************/
/*                                                          */
/* M16C/Tiny Series Program Collection                      */
/*                                                          */
/* FILE NAME   : rec05b0005-0102_rcv.c                      */
/* CPU         : M16C/29 Group                              */
/* FUNCTION    : CRC Calculation Circuit                    */
/*               (Clock asynchronous serial I/O receive)    */
/* HISTORY     : 2006.04.13 Ver 1.02                        */
/*                                                          */
/* Copyright (C) 2005. Renesas Technology Corp.             */
/* All right reserved.                                      */
/*                                                          */
/************************************************************/

/************************************************************/
/*    Include File                                          */
/************************************************************/
#include "sfr29.h"     // Special function register header file

/************************************************************/
/*    Function Definition                                   */
/************************************************************/
void init_mcu(void);   // MCU initialize
void init_uart(void);  // UART initialize
void init_crc(void);   // CRC initialize
void crc_rcv(void);
void wait_10ms(void);  // Main clock oscillation stable wait routine

/************************************************************/
/*    Define Label                                          */
/************************************************************/
#define PRODUCT_TYPE 0     // 29 group: 0    26A group: 1
#define PIN_TYPE 0         // 80 pin: 0      64 pin: 1 (29 group)
                           // 48 pin: 0      42 pin: 1 (26A group)
#define PLL_ON 0           // 0: not use PLL or PM20 is 1
                           // 1: use PLL and PM20 is 0

/************************************************************/
/*    Define Const                                          */
/************************************************************/
/* port2_0 - port2_7 data : 0  1  2  3  4  5  6  7  8  9  9 */
unsigned char count_data[11] =
{0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xD8,0x80,0x90,0x90};

unsigned short recv_data[6];
unsigned int i;

/************************************************************/
/*    Main Program                                          */
/************************************************************/
void main(void) {
   init_mcu();
   init_uart();
   init_crc();
```

```
   crc_rcv();
}

/*****************************************************/
/*    MCU Initialize Program                         */
/*****************************************************/
void init_mcu(void) {
   prcr = 0x03;   // Protect register off

   pm20 = 1;      // Specifying wait when accessing SFR (Note)
                  // (Note) This bit can only be rewritten while the PLC07 bit is
                  // "0" (PLL Off). Also, to select a 16MHz or higher PLL clock,
                  // set this bit to "0" (2 wait).

   wait_10ms();   // For waiting 10ms, initialize Time A0 and start counting.

   cm2 = 0x00;    // System register2 Initialize
   cm0 = 0x08;    // System register0 Xcin-Xcout: high
   cm1 = 0x20;    // System register1 Xcin-Xcout: high

   prcr = 0x00;   // Protect register on

   #if PRODUCT_TYPE        // Product selection: 26A group
      ifsr2a = 1;          // Interrupt request cause select register2 IFSR2A
                           // <IFSR20>   : Reserved bit (Must be set to "1")
      prcr = 0x04;         // Protect register off
      #if PIN_TYPE         // Port setting
         pacr = 0x01;      // 42pin type
      #else
         pacr = 0x04;      // 48pin type
      #endif
      prcr = 0x00;         // Protect register on
   #else                   // Product selection: 29 group
      ifsr2a = 0;          // Interrupt request cause select register2 IFSR2A
                           // <IFSR20>   : Reserved bit (Must be set to "0")
      prcr = 0x04;         // Protect register off
      #if PIN_TYPE         // Port setting
         pacr = 0x02;      // 64pin type
      #else
         pacr = 0x03;      // 80pin type
      #endif
      prcr = 0x00;         // Protect register on
   #endif
}

/*****************************************************/
/*    Main Clock Oscillation Stable Wait 10ms Routine     */
/*****************************************************/
void wait_10ms(void) {
   ta0mr = 0x00;     // Set Timer A0 mode register (Timer mode, count source: f1)

   ta0 = 20000-1;    // Setting counter value (10msec @4MHz/2, f1)

   ta0ic = 0x00;     // Clear interrupt request bit

   tabsr = 0x01;     // Timer A0 start counting
```

```
   while (ir_ta0ic == 0){   }

   ir_ta0ic = 0;     // Clear interrupt request bit

   tabsr = 0x00;     // Timer A0 stops counting
}

/*********************************************************/
/*    UART Initialize Program                            */
/*********************************************************/
void init_uart(void) {
   u0mr = 0x45;   // UART0 transmit/receive mode register setting
                  // UART mode transfer data 8 bits long
                  // Internal clock select one stop bit
                  // Parity enabled (odd parity)

   u0c0 = 0x04;   // UART0 transmit/receive control register 0 setting
                  // RTS function select
                  // CTS/RTS function enabled
                  // TxD0 pin is CMOS output
                  // Transmission data is output at falling edge of transfer clock
                  // and reception data is input at rising edge LSB first

   ucon = 0x01;   // UART transmit/receive control register 2 setting
                  // UART0 transmit interrupt cause is selected to "Transmission
                  // completed (TXEPT=1)"
                  // CTS/RTS shared pin

   u0brg = 129;   // Setting UART0 bit rate generator (Approx 9600bps @20MHz f1)

   u0c1 = 0x04;   // UART transmit/receive control register 1 setting Receive
                  // enabled
}

/*********************************************************/
/*    CRC Initialize Program                             */
/*********************************************************/
void init_crc(void) {
   #if PLL_ON
      crcsw = 0;              // CRC snoop on write is disabled
      crcsr = 0;              // CRC snoop on read is disabled

   #else
                              // CRC calculation circuit
      crcps = 0;              // CRC mode polynomial selection bit is selected
                              // to "0" (CRC-CCITT)
      crcms = 0;              // CRC mode selection bit is selected to "0" (LSB
                              // first)

      crcsarl = (char)(&u0rb);    // Setting SFR snoop address
      crcsarh = (char)((unsigned long)(&u0rb) >> 8 );

      crcsw = 0;              // CRC snoop on write is disabled
      crcsr = 1;              // CRC snoop on read is enabled

      crcd = 0;               // CRC data register set to "0"
```

```
   #endif
}

/*********************************************************/
/*    CRC Receive Program                                */
/*********************************************************/
void crc_rcv(void) {

   for (i=0;i<6;i++) {
      while (!ri_u0c1) {
                     // Check & wait the status of UART0 receive complete flag
      }
      recv_data[i] = u0rbl;   // Reading receive data
   }

   if ( crcd == 0 ) {
                     // CRC data check CRCD == 0x00?
      while (1) {
                     // Normal end loop
      }
   }

   else {
      while (1) {
                     // Error end loop
      }
   }
}
```

5. 参考文献

数据手册

M16C/26A 群（M16C/26A、M16C/26T）硬件手册 Rev.1.00

M16C/29 群硬件手册 Rev.1.00

（最新版本请从瑞萨科技网页上取得）

技术信息/技术更新

（最新信息请从瑞萨科技网页上取得）

公司主页和咨询窗口

瑞萨科技公司主页

http://www.cn.renesas.com

咨询

http://www.renesas.com/inquiry

修订记录

Rev.	发行日	修订内容	
		页	要点
1.00	2006.04.14	—	初版发行

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

注意

本文只是参考译文，前页所载英文具有正式效力。

请遵循安全第一进行电路设计

1. 虽然瑞萨科技尽力提高半导体产品的质量和可靠性，但是半导体产品也可能发生故障。半导体的故障可能导致人身伤害、火灾事故以及财产损害。在电路设计时，请充分考虑安全性，采用合适的如冗余设计、利用非易燃材料以及故障或者事故防止等的安全设计方法。

关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的瑞萨科技产品的参考资料，不转让属于瑞萨科技或者第三者所有的知识产权和其它权利的许可。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法以及其它应用电路的例子而引起的损害或者对第三者的权力的侵犯，瑞萨科技不承担责任。
3. 本资料所记载的产品数据、图、表、程序、算法以及其它所有信息均为本资料发行时的信息，由于改进产品或者其它原因，本资料记载的信息可能变动，恕不另行通知。在购买本资料所记载的产品时，请预先向瑞萨科技或者经授权的瑞萨科技产品经销商确认最新信息。

本资料所记载的信息可能存在技术不准确或者印刷错误。因这些错误而引起的损害、责任问题或者其它损失，瑞萨科技不承担责任。

同时也请通过各种方式注意瑞萨科技公布的信息，包括瑞萨科技半导体网站。
（http：//www.renesas.com）

4. 在使用本资料所记载部分或者全部数据、图、表、程序以及算法等信息时，在最终做出有关信息和产品是否适用的判断前，务必对作为整个系统的所有信息进行评价。由于本资料所记载的信息而引起的损害、责任问题或者其它损失，瑞萨科技不承担责任。
5. 瑞萨科技的半导体产品不是为在可能和人命相关的环境下使用的设备或者系统而设计和制造的产品。在研讨将本资料所记载的产品用于运输、交通车辆、医疗、航空宇宙用、原子能控制、海底中继器的设备或者系统等特殊用途时，请与瑞萨科技或者经授权的瑞萨产品经销商联系。
6. 未经瑞萨科技的书面许可，不得翻印或者复制全部或者部分资料的内容。
7. 如果本资料所记载的某产品或者技术内容受日本出口管理限制，必须在得到日本政府的有关部门许可后才能出口，并且不准进口到批准目的地国家以外的国家。
   禁止违反日本和（或者）目的地国家的出口管理法和法规的任何转卖、挪用或者再出口。
8. 如果需要了解本资料所记载的信息或者产品的详细，请与瑞萨科技联系。