

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/28、M16C/29 群

### 多主控I<sup>2</sup>C bus接口与I<sup>2</sup>C串行EEPROM的接口设计

#### 1. 要点

本资料说明了M16C/28、M16C/29 群单片机如何通过多主控I<sup>2</sup>C bus接口，连接I<sup>2</sup>C串行EEPROM。从应用的角度出发，本资料具体说明了I<sup>2</sup>C串行EEPROM的工作原理和单片机对I<sup>2</sup>C串行EEPROM的操作过程。（I<sup>2</sup>C bus是荷兰PHILIPS公司的注册商标）

本文涉及的应用示例，所使用的单片机属于M16C/29群，所使用的I<sup>2</sup>C串行EEPROM为2K容量的HN58X2402SI。应用程序的功能是：以字节为单位向EEPROM顺序写入8个字节以内的数据；查询EEPROM内部写入周期状态；顺序读出任意字节的数据。

应用示例中的硬件结构、软件参数等信息仅供参考；用于评估及试验时，请根据具体情况进行确认。

#### 2. 说明

本篇资料，适用于 M16C/28、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/28、M16C/29 群具有相同 SFR（特殊功能寄存器）定义的产品。

由于 M16C 族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. 基本功能简介

#### 3.1 多主控 I<sup>2</sup>C bus 接口

多主控 I<sup>2</sup>C bus 接口是以飞利浦公司 I<sup>2</sup>C bus 的数据发送格式为基础进行串行通信的电路，具有仲裁失败检测功能和同步功能。多主控 I<sup>2</sup>C bus 接口功能一览如表 3.1 所示。

多主控 I<sup>2</sup>C bus 接口由 I<sup>2</sup>C0 地址寄存器 (S0D0)、I<sup>2</sup>C0 数据移位寄存器 (S00)、I<sup>2</sup>C0 时钟控制寄存器 (S20)、I<sup>2</sup>C0 控制寄存器 0 (S1D0)、I<sup>2</sup>C0 控制寄存器 1 (S3D0)、I<sup>2</sup>C0 控制寄存器 2 (S4D0)、I<sup>2</sup>C0 状态寄存器 (S10)、I<sup>2</sup>C0 开始/停止条件控制寄存器 (S2D0) 和其他控制电路构成。

表 3.1 多主控 I<sup>2</sup>C bus 接口功能

项目	功能
格式	符合飞利浦公司的 I <sup>2</sup> C bus 规格 <ul style="list-style-type: none"> <li>• 7 位寻址格式</li> <li>• 高速时钟模式</li> <li>• 标准时钟模式</li> </ul>
通信模式	符合飞利浦公司的 I <sup>2</sup> C bus 规格 <ul style="list-style-type: none"> <li>• 主发送</li> <li>• 主接收</li> <li>• 从发送</li> <li>• 从接收</li> </ul>
SCL 时钟频率	16.1 kHz ~ 400 kHz (V <sub>IIC</sub> * = 4 MHz)
输入/输出引脚	串行数据线 SDAMM (SDA) 串行时钟线 SCLMM (SCL)

\*V<sub>IIC</sub> = I<sup>2</sup>C 总线系统时钟

在本应用示例中，总线的拓扑结构为一主一从；所使用的多主控 I<sup>2</sup>C bus 接口功能为标准时钟模式，SCL 时钟频率 100 kHz，通信模式为主发送模式和主接收模式；所使用的单片机输入输出引脚为串行数据线 SDAMM 和串行时钟线 SCLMM。

#### 3.2 瑞萨 I<sup>2</sup>C 串行 EEPROM 的使用方法

瑞萨科技拥有存储容量从 2K 至 1M 的多种不同封装的带 I<sup>2</sup>C 串行接口的 EEPROM。这一系列 EEPROM 在结构、功能和使用方法上有很大的相似性。本应用示例中所使用的 EEPROM 型号为 HN58X2402SI，其存储容量为 2Kbits。HN58X2402SI 的封装及引脚如图 3.1 所示；其引脚定义如表 3.2 所示。

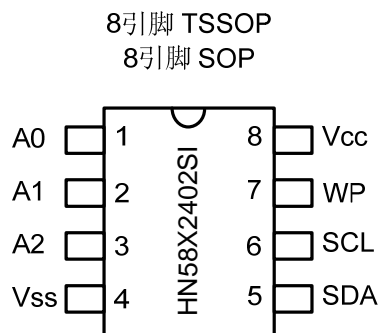


图 3.1 HN58X2402SI 的引脚图

表 3.2 HN58X2402SI 的引脚定义

项目	功能
A2, A1, A0	器件地址
SCL	I <sup>2</sup> C 总线的时钟线
SDA	I <sup>2</sup> C 总线的数据线
WP	写入保护
Vcc	电源
Vss	地

对于 HN58X2402SI 器件，总线上某个 EEPROM 的器件地址为 7 位，即  $(1010A_2A_1A_0)_2$ ；前 4 位器件地址固定为  $(1010)_2$ ，最后三位地址由 A2、A1、A0 引脚电平决定。I<sup>2</sup>C 总线上最多可支持 8 个不同地址的 HN58X2402SI 器件。

在本应用示例中，器件地址 A2、A1、A0 设定为“0”、“1”、“1”。WP 引脚接地（不使用写入保护功能）。

### 3.2.1 字节写入

字节写入是最基本的写入功能，其操作时序如图 3.2 所示。其中，紧随器件地址写入的是读写位，此位为“0”即“写从器件”，为“1”即“读从器件”。当 EEPROM 收到停止位后，EEPROM 进入内部写入周期。在内部写入周期结束之前，EEPROM 不接收时钟线和数据线上的信息。写入周期结束后，EEPROM 回到待机模式。

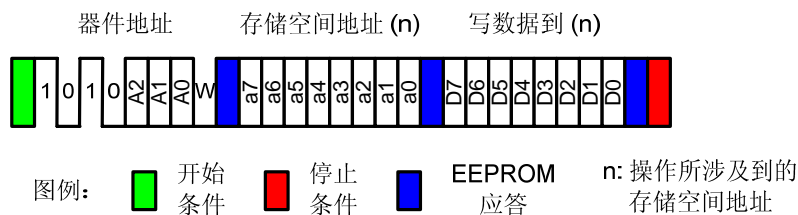


图 3.2 字节写入时序

### 3.2.2 页写入

HN58X2402SI 内部设有写入缓存，支持一次写入多个字节的数据。使用页写入功能，可以在一个写入周期内连续写入多达 8 个字节的数据。页写入的时序和字节写入的时序相同，不同之处只是多写入几个数据。

当 EEPROM 接收到一个以上的数据时，就进入页写入状态。此时，内部存储空间地址计数器的 a2 至 a0 位随着写入数据的个数自动递增。但是，当 a2 至 a0 位地址达到当前页的尾地址时（HN58X2402SI 内部以 8 字节为一页），便自动转向当前页的首地址，依次循环；此时再写入数据，将覆盖当前页的原始数据。因此，在使用页写入功能时，应通过程序控制好写入的存储空间地址。

当 EEPROM 收到停止位后，EEPROM 进入内部写入周期。

页写入的时序如图 3.3 所示。

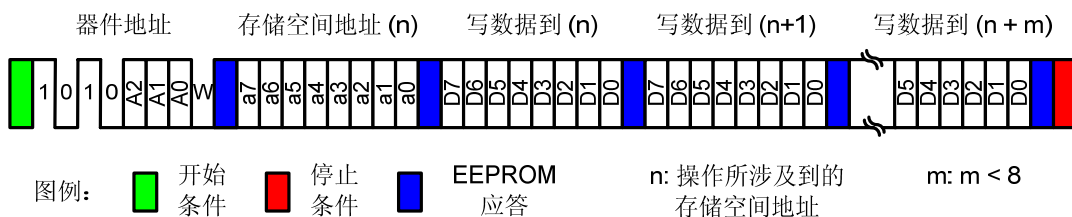


图 3.3 页写入时序

### 3.2.3 应答轮询

无论是字节写入方式，还是页写入方式，在写入操作结束后（单片机发送停止条件后），EEPROM 都会进入内部写入状态，在此期间不响应时钟线和数据线信号。此时，如需对 EEPROM 进行下一步操作，就需要判断其状态，以确定器件是否处于内部写入状态。

有两种方法可以实现这一目的，第一种是延时，第二种应答轮询。

根据 HN58X2402SI 硬件手册，当 V<sub>cc</sub> 为 2.7V ~ 5.5V 时，器件的内部写入周期最长为 10ms。如果以此为依据，用延时的方法，会浪费很多时间。

应答轮询则可以节省时间；但唯一的问题就是这段时间内，I<sup>2</sup>C 总线被占用。如果总线上有多个器件，其他通信可能会暂时受阻。在本应用示例中，由于总线结构为简单的一主一从结构，因此用应答轮询的方法最合适。

应答轮询的实现方法如图 3.4 所示。其过程是在写入结束后，单片机发出 EEPROM 器件地址和读/写位（置为“写”），等待 EEPROM 应答。如果此时 EEPROM 在内部写入周期中，就不会返回应答；否则，EEPROM 会返回应答。以此为依据，即可判断 EEPROM 的状态。

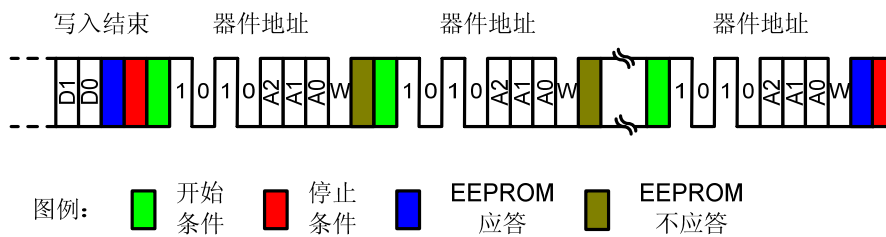


图 3.4 应答轮询时序

### 3.2.4 当前地址读取

EEPROM 内部地址计数器储存有最后一次读/写操作所涉及的最后一个存储空间地址的下一个地址。

当上一个操作为读操作并使用至最后一页的页尾地址时，当前地址将回归零地址；当上一个操作为写操作并使用至某页的页尾地址时，当前地址将回归当前页的首地址。

只要电源供电正常，当前地址就不会丢失。当器件上电时，当前地址则是不定的。

当前地址读取的时序如图 3.5 所示。

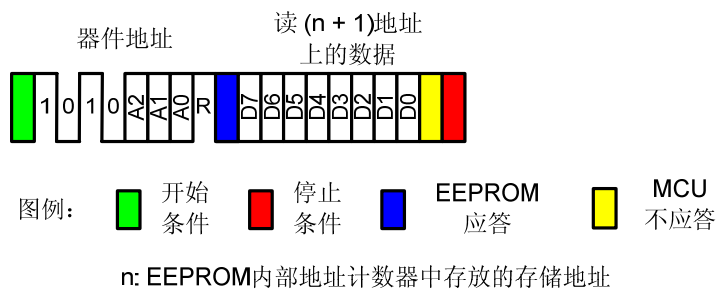


图 3.5 当前地址读取时序

### 3.2.5 任意地址读取

任意地址读取需要首先设定将要读取的存储空间地址。设定过此地址后，EEPROM 输出存储空间内的数据。当 EEPROM 收到 MCU 的无应答信号及停止条件后，EEPROM 停止当前操作并进入待机状态。

任意地址读取的时序如图 3.6 所示。

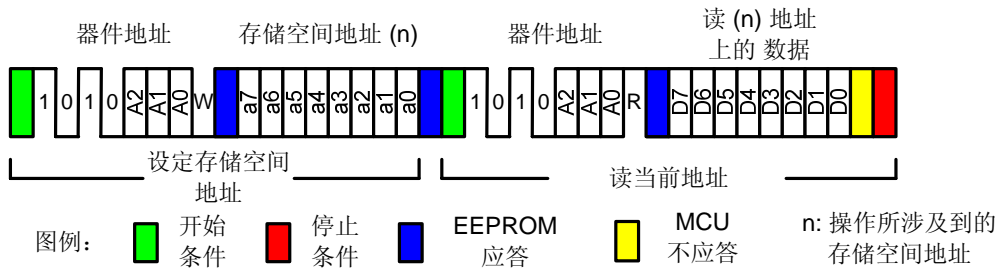


图 3.6 任意地址读取时序

### 3.2.6 连续读取

连续读取可以通过当前地址读取或任意地址读取启动。被读取的存储空间的地址随着读取操作而自动递增。当存储空间地址达到最后一页的页尾地址时，存储空间地址自动返回零地址；此后，连续读取可以继续运行。

当 EEPROM 收到单片机的无应答及其后的停止条件后，连续读取操作停止。

连续读取的时序如图 3.7 所示。

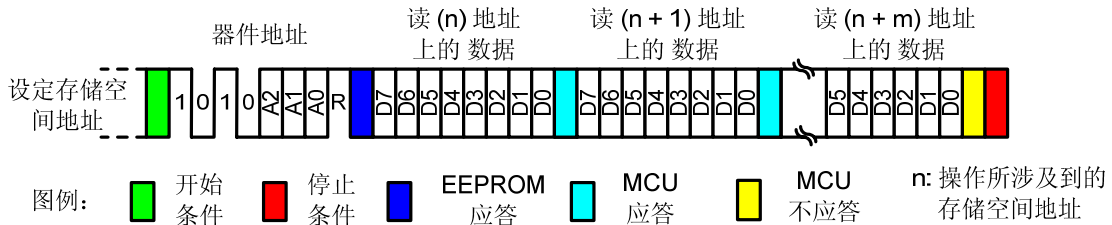


图 3.7 连续读取时序

## 4. 应用示例的说明

### 4.1 电路结构

本例中，电路结构如图 4.1 所示。

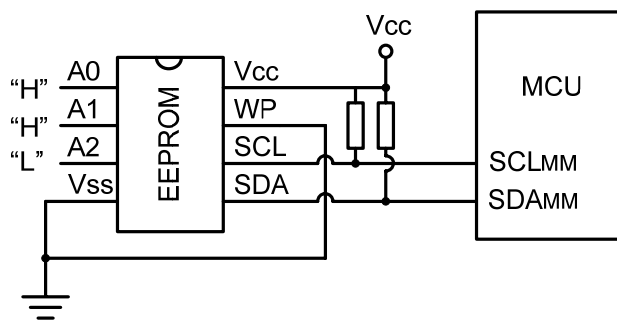


图 4.1 应用示例的参考电路结构

图中所示电路中，EEPROM 器件的器件地址引脚 A2、A1、A0 置为“0”、“1”、“1”。请根据具体电路参数，参考 I<sup>2</sup>C 规范，选择合适的上拉电阻。

### 4.2 程序流程及时序

此例中，程序的基本功能是以字节为单位向 EEPROM 顺序写入 8 字节以内的数据；查询 EEPROM 内部写入周期状态；顺序读出任意字节的数据。

实现这一功能的 I<sup>2</sup>C 处理程序的状态转换图如 4.2 所示。

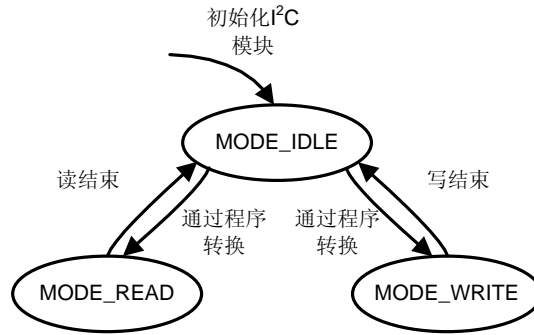


图 4.2 I<sup>2</sup>C 处理程序的状态转换图

图 4.3 所示为带查询的写入模式的时序（字节写入或者页写入均可）；与图 4.3 相对应，图 4.4 所示为写入模式下的子模式状态转换图，其中子模式的转换由中断触发；触发所涉及的中断发生的位置、类型在图 4.3 上标出。

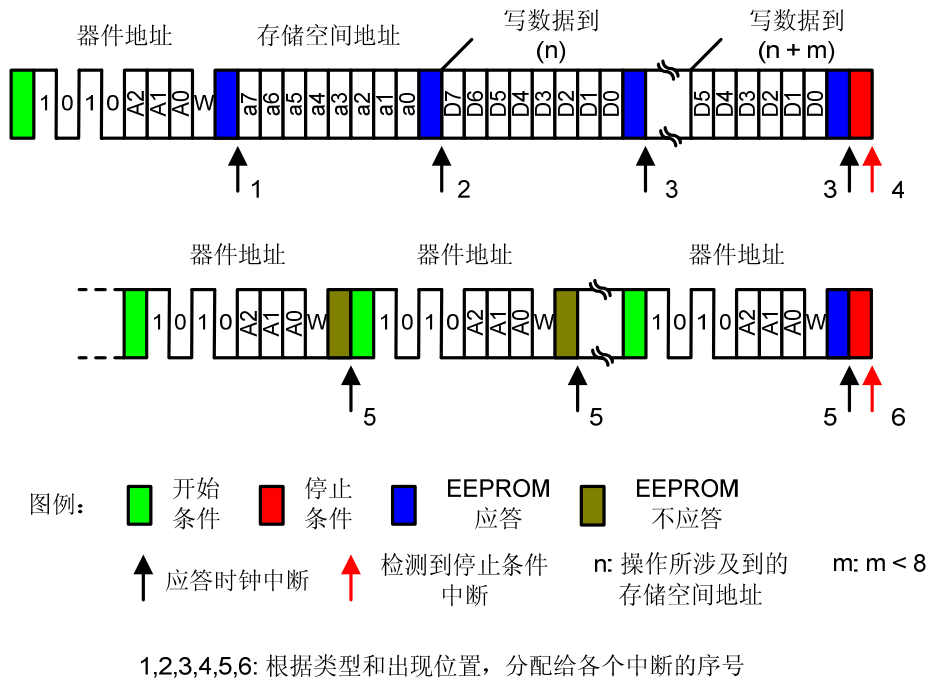


图 4.3 写入模式的时序图



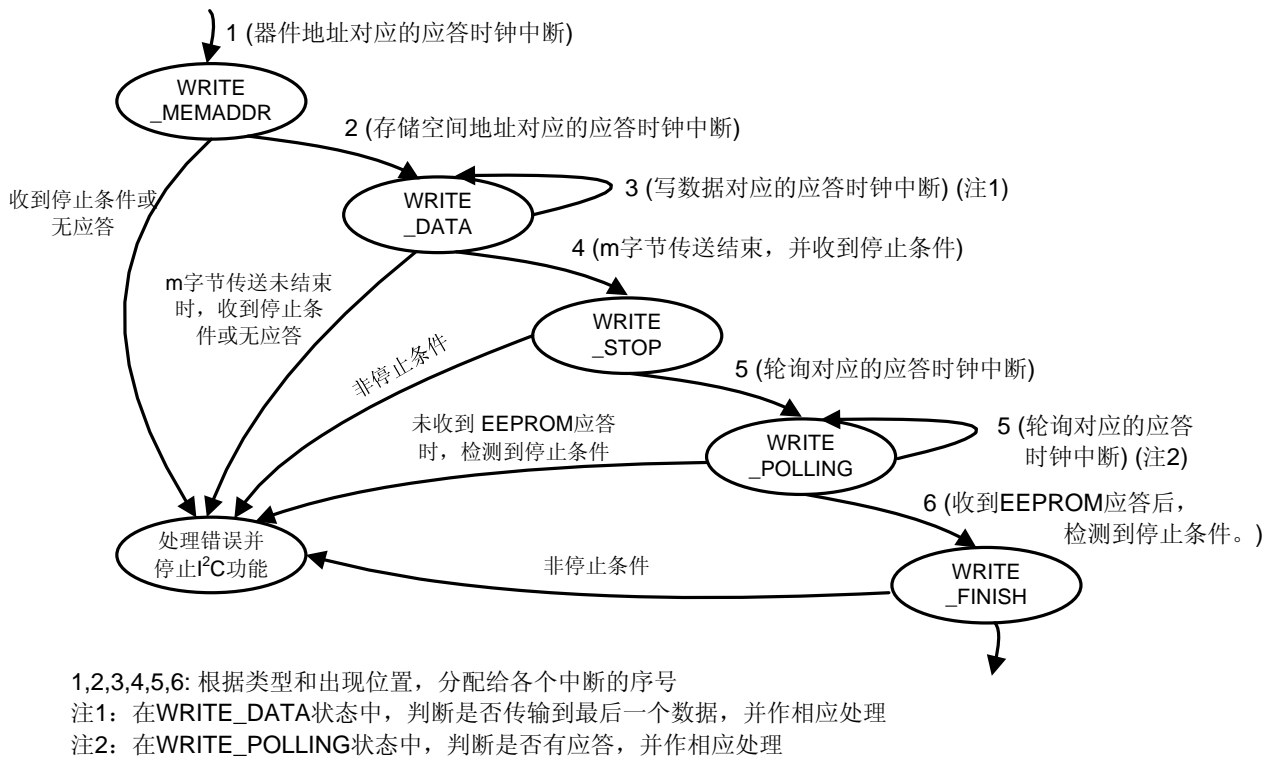


图 4.4 写入模式下的子模式状态转换图

图 4.5 所示为读取模式的时序；与图 4.5 相对应，图 4.6 所示为读取模式下的子模式状态转换图，其中子模式的转换由中断触发；触发所涉及的中断发生的位置、类型在图 4.5 上标出。

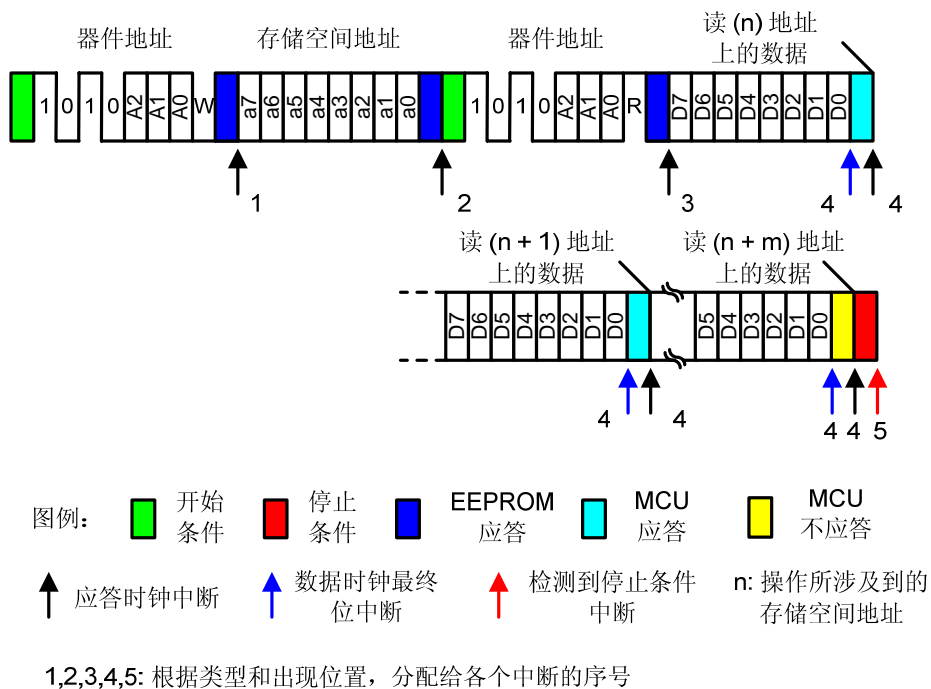
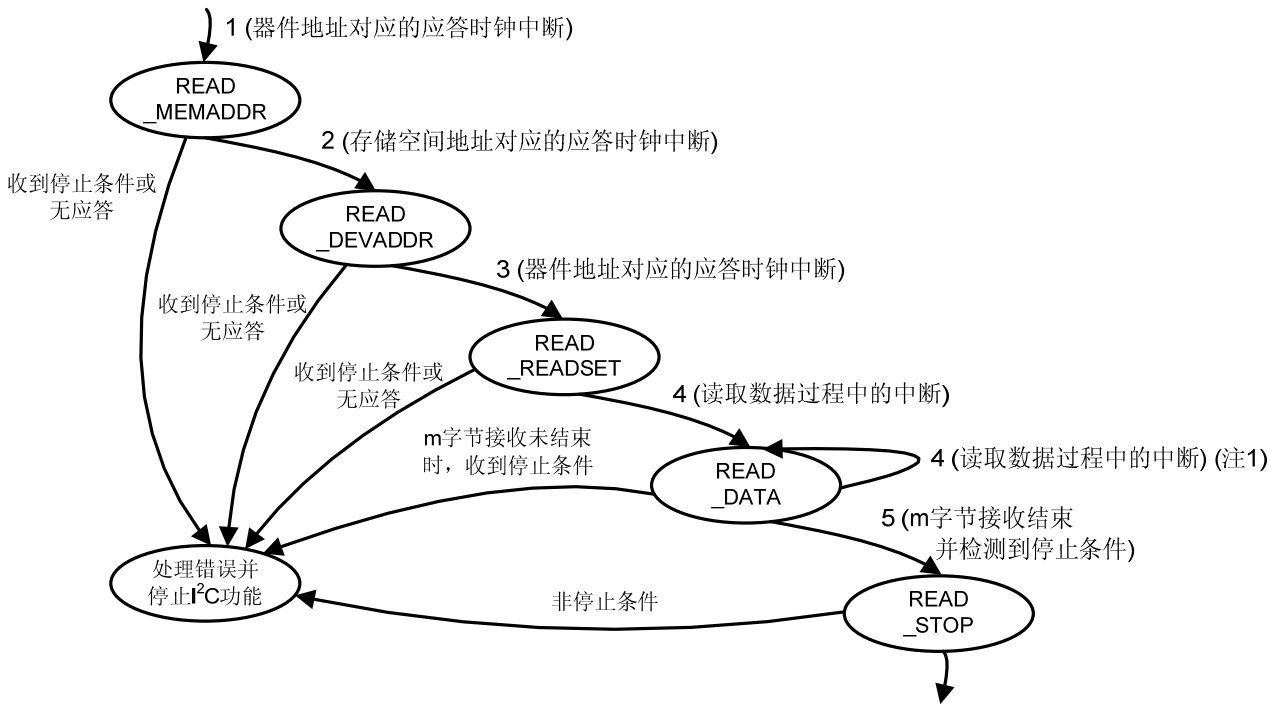


图 4.5 读取模式的时序图



1,2,3,4,5: 根据类型和出现位置, 分配给各个中断的序号  
注1: 具体判断和处理将在READ\_DATA状态中进行

图 4.6 读取模式下的子模式状态转换图

## 5. 参考例程

```

/*****/
/* 项目名称      : M16C/28, M16C/29 群应用说明          */
/*              : 多主控 I2C bus 接口与 I2C 串行 EEPROM 的接口设计 */
/* MCU 型号      : M30290FCTHP                          */
/* C 编译器      : NC30WA, 版本 5.40                    */
/* 文件名        : i2c_eeprom.c                          */
/* 功能简介      : 实现 I2C 对 EEPROM 的读写操作        */
/* 版本          : 1.0                                    */
/*                                                      */
/* Copyright (C) 2007 Renesas Technology Corp.          */
/* All right reserved.                                  */
/*****/

/*****/
/* 包含文件                                             */
/*****/
#include "sfr29.h"

/*****/
/* 定义宏                                             */
/*****/
#define EEPROM_ADDR    0b01010011      /* EEPROM 地址 (低 7 位)  */
#define WRITE          0x0              /* 写                      */
#define WRITE_ADDR     0b11111000     /* 写地址                  */
#define READ           0x1              /* 读                      */
#define READ_ADDR      0b11111000     /* 读地址                  */

/*****/
/* 函数声明                                             */
/*****/
void main(void);

extern void iic_ini(unsigned char);
extern void init_main(void);
extern unsigned char eeprom_operation(unsigned char slave,
                                     unsigned char memory,
                                     unsigned char *buf,
                                     unsigned char len,
                                     unsigned char rw);
extern void eeprom_delay(void);

/*****/
/* 定义变量                                             */
/*****/

```

```

/*****/
unsigned char iic_tr[8] = {1,2,3,4,5,6,7,8};      /* 发送缓存 */
unsigned char iic_re[8] = {0,0,0,0,0,0,0,0};      /* 接收缓存 */

/*****/
名称      : main
功能      : 主函数
输入参数  : 无
返回值   : 无
调用      : init_main(void)
           ini_ini(unsigned char)
           unsigned char eeprom_operation(unsigned char slave,
                                           unsigned char memory,
                                           unsigned char *buf,
                                           unsigned char len,
                                           unsigned char rw);
*****/
void main()
{
    init_main();          /* 初始化单片机      */
    iic_ini(1);           /* 初始化 I2C bus  */
    asm("fset I");       /* 开中断          */

    /* 写数据          */
    if(eeprom_operation(EEPROM_ADDR,WRITE_ADDR,iic_tr,8,WRITE)!=0) {;
    }
    else {;
    }

    eeprom_delay();      /* 等待写入成功 */

    /* 读数据          */
    if(eeprom_operation(EEPROM_ADDR,READ_ADDR,iic_re,8,READ)!=0) {;
    }
    else {;
    }

    eeprom_delay();      /* 等待读出结束 */

    iic_ini(0);          /* 关闭 I2C bus 功能 */

    while(1){
    }
}

```

```

/*****
    名称      : init_main
    功能      : 初始化 MCU
    输入参数  : 无
    返回值   : 无
    调用     : 无
*****/
void init_main(void)
{
    /* 设置系统时钟相关寄存器 */
    prcr = 0x01;          /* 允许写 cm0,cm1,cm2 等寄存器 */
    cm2 = 0x00;          /* 初始化振荡停止检测寄存器 */
    cm1 = 0x20;          /* 初始化系统时钟控制寄存器 1 */
                        /* XIN - XOUT 驱动能力为高 */
    cm0 = 0x08;          /* 初始化系统时钟控制寄存器 0 */
                        /* XCIN - XCOUT 驱动能力为高 */
    prcr = 0x00;          /* 禁止写 cm0,cm1,cm2 等寄存器 */

    prcr = 0x04;          /* 允许写 pacr 等寄存器 */
    pacr = 0x03;          /* 80 引脚版本 */
    prcr = 0x00;          /* 禁止写 pacr 等寄存器 */
}

```

```

/*****/
/* 项目名称      : M16C/28, M16C/29 群应用说明          */
/*              : 多主控 I2C bus 接口与 I2C 串行 EEPROM 的接口设计 */
/* MCU 型号      : M30290FCTHP                          */
/* C 编译器      : NC30WA, 版本 5.40                    */
/* 文件名        : i2c_functions.c                      */
/* 功能简介      : 实现 I2C 对 EEPROM 的读写操作        */
/* 版本          : 1.0                                    */
/*                                                      */
/* Copyright (C) 2007 Renesas Technology Corp.          */
/* All right reserved.                                  */
/*****/

/*****/
/* 包含文件      :                                       */
/*****/
#include "sfr29.h"

/*****/
/* 定义宏        :                                       */
/*****/
#define BYTE_LIMIT 8 /* 本器件的页写入的字节数限制 */

unsigned char iic_mode; /* I2C bus 工作模式 */
#define MODE_IDLE 0x00
#define MODE_READ 0x01
#define MODE_WRITE 0x02

unsigned char submode; /* I2C bus 工作子模式 */
#define WRITE_MEMADDR 0x00
#define WRITE_DATA 0x01
#define WRITE_STOP 0x02
#define WRITE_POLLING 0x03
#define WRITE_FINISH 0x04
#define READ_MEMADDR 0x05
#define READ_DEVADDR 0x06
#define READ_READSET 0x07
#define READ_DATA 0x08
#define READ_STOP 0x09

unsigned char ErrorCode; /* 错误代码 */
#define NORMAL 0x00 /* 正常 */
#define TOF_ERROR 0x01 /* 超时错误 */
#define STOP_ERROR 0x02 /* 收到非正常停止条件错误 */

```

```

#define NACK_ERROR      0x03  /* 未收到应答错误          */
#define UNKNOWN_ERROR  0x04  /* 其他错误                */

/*****/
/*    函数声明                                */
/*****/

void iic_ini(unsigned char);
unsigned char eeprom_operation(unsigned char slave,unsigned char memory,
                               unsigned char *buf,unsigned char len,
                               unsigned char rw);

void eeprom_delay(void);
void iic_int(void);
static void master_transfer(void);
static void master_receive(void);

/*****/
/*    定义变量                                */
/*****/
typedef union{
    struct{
        unsigned char b0:1;
    }bit;
    unsigned char all;
}byte_dt;

static byte_dt iic_sl;                /* 地址及控制字          */
#define iic_slave iic_sl.all
#define iic_rw iic_sl.bit.b0        /* 0:写    1:读          */

static unsigned char iic_length;     /* 发送/接收数据的个数(字节) */
static unsigned char iic_memaddr;   /* 存储空间地址          */
static unsigned char *iic_pointer;  /* 发送/接收缓存指针    */

/*****/
名称      : iic_ini
功能      : IIC 总线初始化
输入参数  : 0: 停止 I2C bus 功能
           1: 启动 I2C bus 功能
返回值   : 无
调用     : 无
/*****/
void iic_ini(unsigned char ini)
{
    asm("pushc FLG"); /* 压栈, 保护标志寄存器 */
}

```

```

if(ini == 1) {          /* 初始化 I2C bus 模块, 启动          */
    s1d0 = 0x00;        /* 8 位数据, 禁止 IIC 功能 (稍后启动)      */
                        /* 寻址格式, 复位解除, IIC 总线输入      */
    s10 = 0x00;        /* 初始化 I2C0 状态寄存器                  */
    s20 = 0x85;        /* 有 ACK 时钟, 有 ACK 应答, 标准时钟模式  */
                        /* 时钟频率 100 kHz (当 VIIC=4MHZ 时)     */
    s3d0 = 0x03;        /* 允许停止条件检测中断                    */
                        /* 允许数据接收结束中断                    */
    s4d0 = 0x19;        /* VIIC = 1/5 fIIC                          */
                        /* 允许超时检测, 长超时                    */
    s2d0 = 0x98;        /* 设定开始/停止条件的检测条件            */

    asm("fclr I");      /* 禁止中断                                  */
    ifsr27 = 1;         /* 设定 I2C bus 中断优先级                  */
    iicic = 0x01;       /* 允许 I2C bus 中断                        */

    iic_mode = MODE_IDLE; /* 设定 I2C bus 模式为空闲模式            */
    ErrorCode = NORMAL; /* 设定错误检测模式为正常模式              */
    es0 = 1;           /* 允许 I2C bus 功能                        */
}
else {                 /* 停止 I2C bus 模块                        */
    asm("fclr I");      /* 禁止中断                                  */
    iicic = 0x00;       /* 禁止 I2C bus 中断                        */
    iic_mode = MODE_IDLE; /* 设置 I2C bus 模式为空闲模式            */
    es0 = 0;           /* 禁止 I2C bus 功能                        */
}
asm("popc FLG");        /* 恢复标志寄存器 (允许中断发生)          */
}

/*****
名称      : eeprom_operation
功能      : 启动 I2C bus 具体操作
输入参数  : unsigned char slave      : 器件地址
           unsigned char memaddr     : 器件存储空间地址
           unsigned char *buffer     : 参与读写的缓存首地址
           unsigned char length      : 读写的字节数
           unsigned char rw          : 读/写
返回值   : 0: 操作启动失败
           1: 操作启动成功
调用     : 无
*****/
unsigned char eeprom_operation(unsigned char slave, unsigned char memaddr,
                               unsigned char *buffer, unsigned char length,
                               unsigned char rw)

```



```

{
    if((bb == 1) || (iic_mode != MODE_IDLE)) {
        return(0); /* 操作启动失败 */
    }
    else {
        asm("pushc FLG"); /* 压栈保护标志寄存器 */
        asm("fclr I"); /* 禁止中断发生 */
        iic_slave = slave << 1; /* 设置器件地址 */
        iic_rw = 0; /* 设置读写标志 */
        iic_length = length; /* 设置字节数 */
        iic_pointer = buffer; /* 设置缓存位置 */
        iic_memaddr = memaddr; /* 设置存储空间地址 */

        if(rw == 0) {
            if(iic_length > BYTE_LIMIT) { /* 限制页写入字节数 */
                iic_length = BYTE_LIMIT;
            }
            iic_mode = MODE_WRITE; /* 写入模式 */
            submode = WRITE_MEMADDR; /* 设定子模式 */
        }
        else {
            iic_mode = MODE_READ; /* 读取模式 */
            submode = READ_MEMADDR; /* 设定子模式 */
        }

        s10 = 0xE0; /* 产生开始条件 */
        s00 = iic_slave;
        asm("popc FLG"); /* 恢复标志寄存器 */
        return(1); /* 操作启动成功 */
    }
}

/*****
名称      : iic_int
功能      : I2C bus 中断处理
输入参数  : 无
返回值   : 无
调用      : void master_transfer(void)
           void master_receive(void)
           void iic_ini(void)
*****/
#pragma INTERRUPT iic_int
void iic_int(void)
{

```

```

if(tof) {
    ErrorCode = TOF_ERROR;
    iic_ini(0);
    return;
}

/* 检查 I2C bus 模式 */
switch (iic_mode) {
    case MODE_WRITE:
        master_transfer();
        break;
    case MODE_READ:
        master_receive();
        break;
    case MODE_IDLE:
        break;
    default:
        break;
}
}

/*****
名称      : master_transfer
功能      : I2C bus 发送操作及内部写入状态查询
输入参数  : 无
返回值   : 无
调用     : 无
*****/
static void master_transfer(void)
{
    /* 检查写入模式下的子模式 */
    switch(submode) {

        case WRITE_MEMADDR:

            if(scpin) {
                scpin = 0;
                ErrorCode = STOP_ERROR;
                iic_ini(0);
                return;
            }

            if(lrb == 1) {

```

```

        ErrorCode = NACK_ERROR; /* 设定错误代码 */
        iic_ini(0); /* 关闭 I2C bus 功能 */
        return; /* 退出函数 */
    }
    else {
        s00 = iic_memaddr; /* 写入存储空间地址 */
        submode = WRITE_DATA; /* 更改子模式 */
    }
    break;

case WRITE_DATA:

    if(scpin) { /* 收到停止条件中断 */
        scpin = 0; /* 清除相关中断标志 */
        ErrorCode = STOP_ERROR; /* 设定错误代码 */
        iic_ini(0); /* 关闭 I2C bus 功能 */
        return; /* 退出函数 */
    }

    if(lrb == 1) { /* 未收到应答 */
        ErrorCode = NACK_ERROR; /* 设定错误代码 */
        iic_ini(0); /* 关闭 I2C bus 功能 */
        return; /* 退出函数 */
    }
    else {
        if(iic_length == 0) { /* 全部数据发送完毕 */
            s10 = 0xC0; /* 发送停止条件 */
            s00 = 0xFF; /* 虚写 */
            submode = WRITE_STOP; /* 更改子模式 */
        }
        else{
            s00 = *iic_pointer; /* 发送 1 字节数据 */
            iic_pointer++; /* 缓存地址递增 */
            iic_length--; /* 发送数量递减 */
        }
    }
    break;

case WRITE_STOP:

    if(scpin) { /* 收到停止条件中断 */
        scpin = 0; /* 清除相关中断标志 */
        s10 = 0xE0; /* 发送开始条件 */
        s00 = iic_slave;
    }

```

```

        submode = WRITE_POLLING; /* 更改子模式          */
    }
    else {
        ErrorCode = UNKNOWN_ERROR; /* 设定错误代码    */
        iic_ini(0); /* 关闭 I2C bus 功能 */
        return; /* 退出函数          */
    }
    break;

case WRITE_POLLING:

    if(scpin) {
        scpin = 0; /* 清除相关中断标志    */
        ErrorCode = STOP_ERROR; /* 设定错误代码    */
        iic_ini(0); /* 关闭 I2C bus 功能 */
        return; /* 退出函数          */
    }

    if(lrb == 0){
        s10 = 0xC0; /* 发送停止条件      */
        s00 = 0xFF; /* 虚写              */
        submode = WRITE_FINISH; /* 更改子模式      */
    }
    else{
        s10 = 0xE0; /* 发送开始条件      */
        s00 = iic_slave;
    }
    break;

case WRITE_FINISH:

    if(scpin) {
        scpin = 0; /* 清除相关中断标志    */
        iic_mode = MODE_IDLE; /* 设定 I2C bus 模式 */
    }
    else {
        ErrorCode = UNKNOWN_ERROR; /* 设定错误代码    */
        iic_ini(0); /* 关闭 I2C bus 功能 */
        return; /* 退出函数          */
    }
    break;

default:
    break;

```

```

    }
}

/*****
    名称      : master_receive
    功能      : I2C bus 读取操作
    输入参数  : 无
    返回值   : 无
    调用     : 无
*****/
static void master_receive(void){

    /* 检查读取模式下的子模式 */
    switch(submode){

        case READ_MEMADDR:

            if(scspin) {                /* 收到停止条件中断 */
                scspin = 0;             /* 清除相关中断标志 */
                ErrorCode = STOP_ERROR; /* 设定错误代码 */
                iic_ini(0);             /* 关闭 I2C bus 功能 */
                return;                 /* 退出函数 */
            }

            if(lrb == 1) {              /* 未收到应答 */
                ErrorCode = NACK_ERROR; /* 设定错误代码 */
                iic_ini(0);             /* 关闭 I2C bus 功能 */
                return;                 /* 退出函数 */
            }
            else {
                s00 = iic_memaddr;      /* 写入存储空间地址 */
                submode = READ_DEVADDR; /* 更改子模式 */
            }
            break;

        case READ_DEVADDR:

            if(scspin) {                /* 收到停止条件中断 */
                scspin = 0;             /* 清除相关中断标志 */
                ErrorCode = STOP_ERROR; /* 设定错误代码 */
                iic_ini(0);             /* 关闭 I2C bus 功能 */
                return;                 /* 退出函数 */
            }
    }
}

```

```

        if(lrb == 1) {
            ErrorCode = NACK_ERROR; /* 未收到应答 */
            iic_ini(0); /* 设定错误代码 */
            return; /* 关闭 I2C bus 功能 */
        }
        else {
            iic_rw = 1; /* 退出函数 */
            s10 = 0xE0; /* 更改读写位 */
            s00 = iic_slave; /* 产生开始条件 */
            submode = READ_READSET; /* 写存储地址及读写位 */
        }
        break;

case READ_READSET:

    if(scpin) {
        scpin = 0; /* 收到停止条件中断 */
        ErrorCode = STOP_ERROR; /* 清除相关中断标志 */
        iic_ini(0); /* 设定错误代码 */
        return; /* 关闭 I2C bus 功能 */
    }

    if(lrb == 1) {
        ErrorCode = NACK_ERROR; /* 未收到应答 */
        iic_ini(0); /* 设定错误代码 */
        return; /* 关闭 I2C bus 功能 */
    }
    else {
        s10 = 0xA0; /* 接收模式设定 */
        s00 = 0xFF; /* 虚写 */
        submode = READ_DATA; /* 更改子模式 */
    }
    break;

case READ_DATA:

    if(scpin) {
        scpin = 0; /* 收到停止条件中断 */
        ErrorCode = STOP_ERROR; /* 清除相关中断标志 */
        iic_ini(0); /* 设定错误代码 */
        return; /* 关闭 I2C bus 功能 */
    }

    if(wit == 1) {
        /* 数据时钟最终位中断 */
    }

```

```

        iic_length--;
        if(iic_length == 0) {
            ackbit = 1;          /* 下一个应答中断不应答 */
        }
        else {
            ackbit = 0;          /* 下一个应答中断应答 */
        }
        return;                  /* 退出函数 */
    }
else {                          /* 应答时钟中断 */
    *iic_pointer = s00;         /* 接收数据 */
    iic_pointer++;             /* 接收堆栈指针递增 */

    if(iic_length == 0) {      /* 接收到最后一个数据 */
        ackbit = 0;            /* 下一个应答中断应答 */
        submode = READ_STOP;   /* 更改子模式 */
        s10 = 0xC0;            /* 发送停止条件 */
        s00 = 0xFF;           /* 虚写 */
    }
    else {                      /* 仍有数据要接收 */
        s00 = 0xFF;           /* 虚写 */
    }
}
break;

case READ_STOP:

    if(scpin) {                 /* 收到停止条件中断 */
        scpin = 0;              /* 清除相关中断标志 */
        iic_mode = MODE_IDLE;   /* 设定 I2C bus 模式 */
    }
    else {                      /* 未收到停止条件中断 */
        ErrorCode = UNKNOWN_ERROR; /* 设定错误代码 */
        iic_ini(0);             /* 关闭 I2C bus 功能 */
        return;                 /* 退出函数 */
    }
    break;

default:
    break;
}
}

/*****

```

名称 : eeprom\_delay  
功能 : 当总线忙或 I2C bus 状态为非空闲时, 等待  
输入参数 : 无  
返回值 : 无  
调用 : 无

```
*****/  
void eeprom_delay(void){  
    while((bb == 1) || (iic_mode != MODE_IDLE)){  
    }  
}
```



## 6. 参考文献

硬件手册

M16C/28 群硬件手册

M16C/29 群硬件手册

HN58X2402SI 芯片手册

(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>

[support.china@renesas.com](mailto:support.china@renesas.com)

## 修订记录

Rev.	发行日	修订内容	
		页	要点
1.00	2007.04.05	—	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、机动车辆、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。