

R7F0C004

串行阵列单元 (SAU) (通过简易 IIC 控制 EEPROM 篇)

R01AN2444CC0100
Rev. 1.00
2015.03.31

要点

本篇应用说明介绍了使用串行阵列单元 (SAU) 的简易 IIC 功能控制 EEPROM 的方法。使用简易 IIC 功能，通过中断处理实现了对 IIC 总线连接的 EEPROM 的读写。

对象 MCU

R7F0C004

在将本篇应用说明应用到其他单片机时，需结合其他单片机的规格进行变更，并进行详细的评价。

目录

1.	规格	4
2.	动作确认条件	12
3.	相关应用说明	12
4.	硬件说明	13
4.1	硬件配置示例	13
4.2	使用引脚一览表	13
5.	软件说明	14
5.1	操作概要	14
5.2	选项字节设定一览表	16
5.3	常数一览表	16
5.4	变量一览表	18
5.5	函数一览表	20
5.6	函数说明	22
5.7	流程图	31
5.7.1	初始设定函数	31
5.7.2	系统函数	32
5.7.3	CPU 时钟的设定	32
5.7.4	串行阵列单元的设定	33
5.7.5	主函数	43
5.7.6	EEPROM 的选择	46
5.7.7	停止条件发行函数	47
5.7.8	停止条件生成函数	48
5.7.9	总线释放处理函数	50
5.7.10	EEPROM 写入处理	51
5.7.11	EEPROM 地址检查处理	53
5.7.12	计算从属设备地址的处理	54
5.7.13	写入数据的页分割处理	55
5.7.14	开始条件发行处理	56
5.7.15	等待 EEPROM 写入完成的处理	58
5.7.16	EEPROM 读出处理	59
5.7.17	等待 EEPROM 读出完成的处理	61
5.7.18	从属设备地址发送完成处理	61
5.7.19	高位地址发送完成处理	62
5.7.20	重新开始处理	62
5.7.21	数据接收开始处理	63
5.7.22	数据接收处理	64
5.7.23	最后数据接收处理	65
5.7.24	数据发送开始处理	65
5.7.25	数据发送处理	66
5.7.26	下一页写入开始处理	67
5.7.27	SCL 空时钟输出处理	68

5.7.28	SCL 上升处理.....	68
5.7.29	SCL 下降处理.....	69
5.7.30	ACK 确认处理.....	69
5.7.31	SCL 低电平时间等待处理.....	70
5.7.32	SCL 高电平时间等待处理.....	70
5.7.33	INTIICr 中断处理.....	71
5.7.34	INTTM02 中断处理	74
6.	参考例程	75
7.	参考文献	75
	公司主页和咨询窗口	75

1. 规格

本篇应用说明使用串行阵列单元的简易 IIC 功能，对通过 IIC 总线连接的 EEPROM 进行读写。

- 对 EEPROM 的存取是通过将存取用信息值设定到结构体，然后调用函数来处理的。
- 考虑到作为 API 的动作，EEPROM 的控制尽量使用中断进行处理。
- 作为操作对象的 EEPROM，可以从 2K~512K 位中选择 1 种。（默认选择 16K 位的 R1EX24016A。主函数也是对应此款芯片进行的测试处理。）
- 对应于选择的 EEPROM 进行存取控制。
- 在 EEPROM 读出过程中进行了复位，EEPROM 在 SDA 信号输出低电平的状态下重新开始时，本应用也支持这种从总线占有状态到总线释放的操作。
- 使用的简易 IIC 通道以通道 00 为标准，但也考虑到了能够方便地变更指定。

本篇应用说明中使用到的外围功能和用途如表 1.1 所示，IIC 通信概要如图 1.1 所示。

使用简易 IIC 功能的 IIC 通信的时序如图 1.2~图 1.8 所示。

表 1.1 相关外围功能和用途

外围功能	用途
串行阵列单元	利用简易 IIC 功能进行 IIC 主控设备的发送接收。 （使用 SCL00 引脚和 SDA00 引脚）

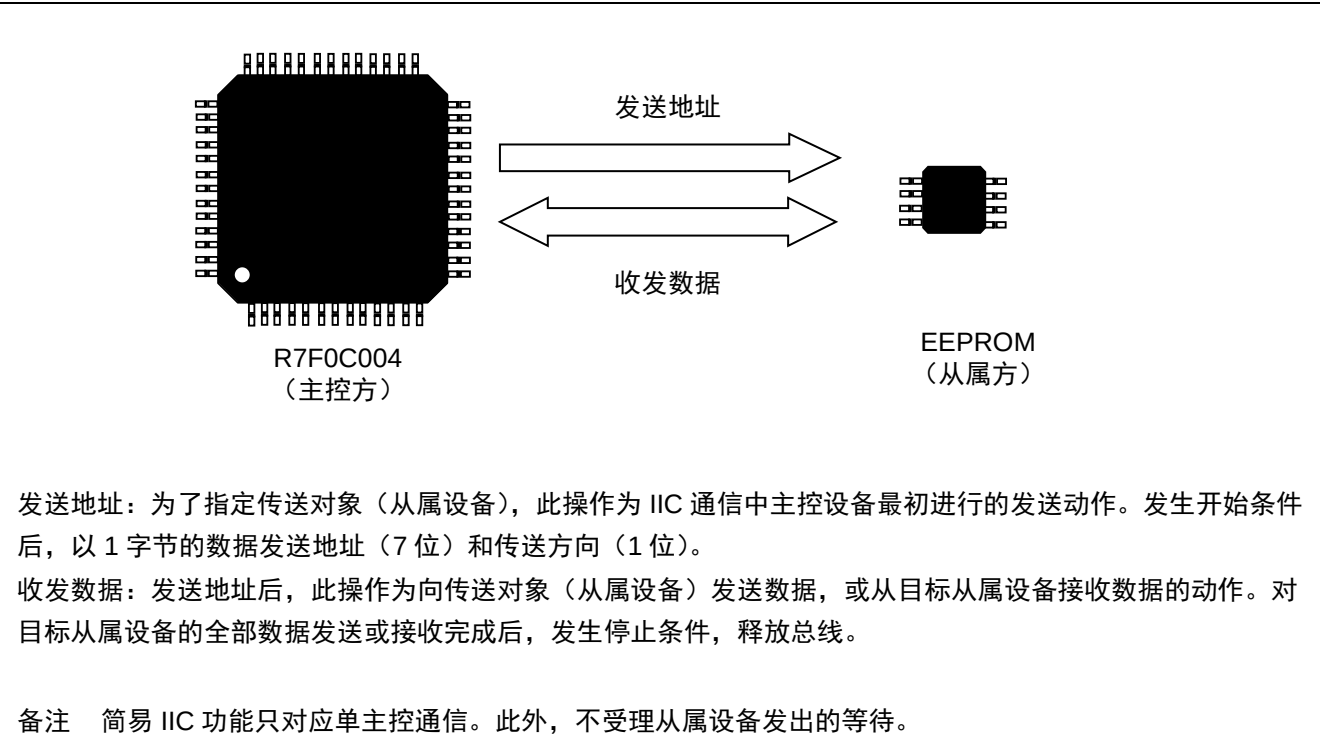


图 1.1 IIC 通信概要

(1) 主控设备→从属设备通信 1 (开始条件~地址~数据)

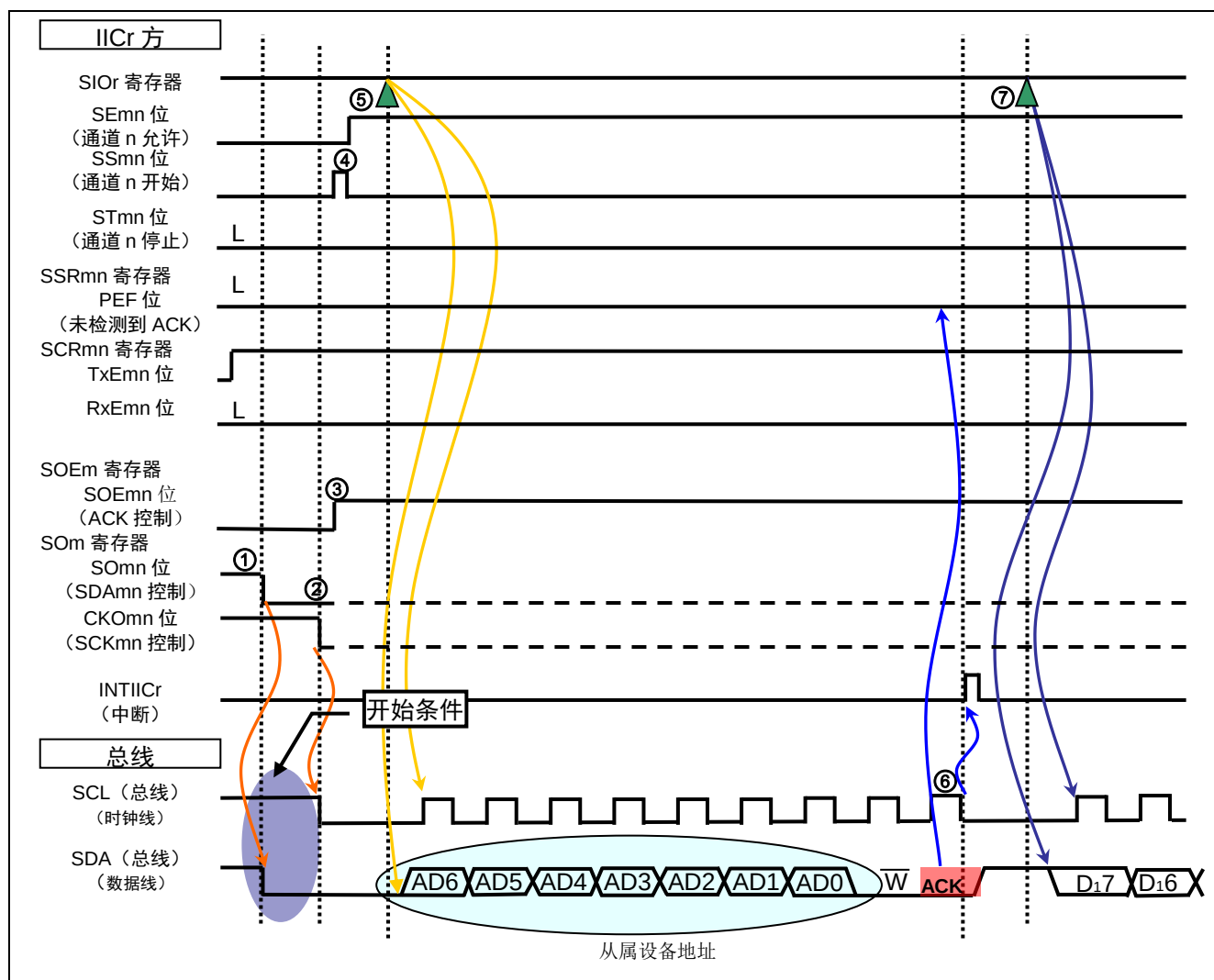


图 1.2 IIC 通信时序图 (主控设备→从属设备通信示例) (1/4)

- ① 如果 IICmn 的初始设定 (此处预先设置了 SCRmn 寄存器的 TxEmn=1, RxEmn=0) 完成了的话, 设定 SOm 寄存器的 SOMn 位为 0, 使 SDA 信号下降, 以产生开始条件。
- ② 经过开始条件的保持时间 (标准模式下为 4.0μs, 快速模式下为 0.6μs) 后, 设置 SOm 寄存器的 CKOm 位为 0, 使 SCL 信号下降。
- ③ 为了进行通信, 设置 SOEm 寄存器的 SOEmn 位为 1, 以允许输出。
- ④ 设置 SSmn 寄存器的 SSmn 位为 1, 使通道 n 为动作允许状态。
- ⑤ 向 SIOr 寄存器写入从属设备的地址后, 通信启动。
- ⑥ 一旦从属设备地址发送完成, 就会产生 INTIICr。
- ⑦ 通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。如果 PEF 位为 0, 则向 SIOr 寄存器写入发送数据。如果 PEF 位为 1, 则中止发送。

(2) 主控设备→从属设备通信 2 (地址~数据~数据)

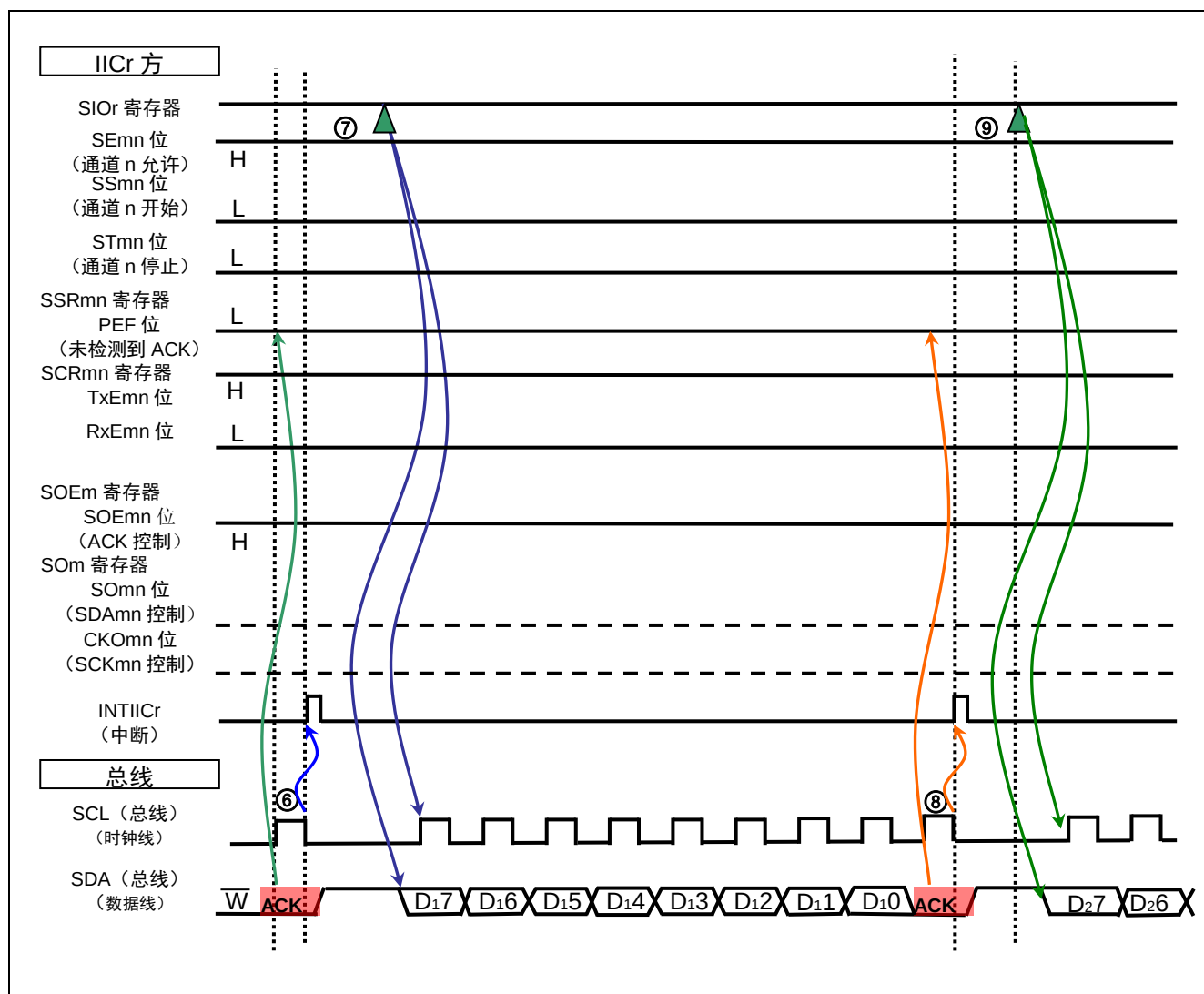


图 1.3 IIC 通信时序图 (主控设备→从属设备通信示例) (2/4)

- ⑥ 一旦从属设备地址发送完成，就会产生 INTIICr。
- ⑦ 通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。如果 PEF 位为 0，则向 SIOr 寄存器写入发送数据。如果 PEF 位为 1，则中止发送。
- ⑧ 一旦数据发送完成，就会产生 INTIICr。
- ⑨ 通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。如果 PEF 位为 0，则向 SIOr 寄存器写入发送数据。如果 PEF 位为 1，则中止发送。

(3) 主控设备→从属设备通信 3 (数据~数据~停止条件)

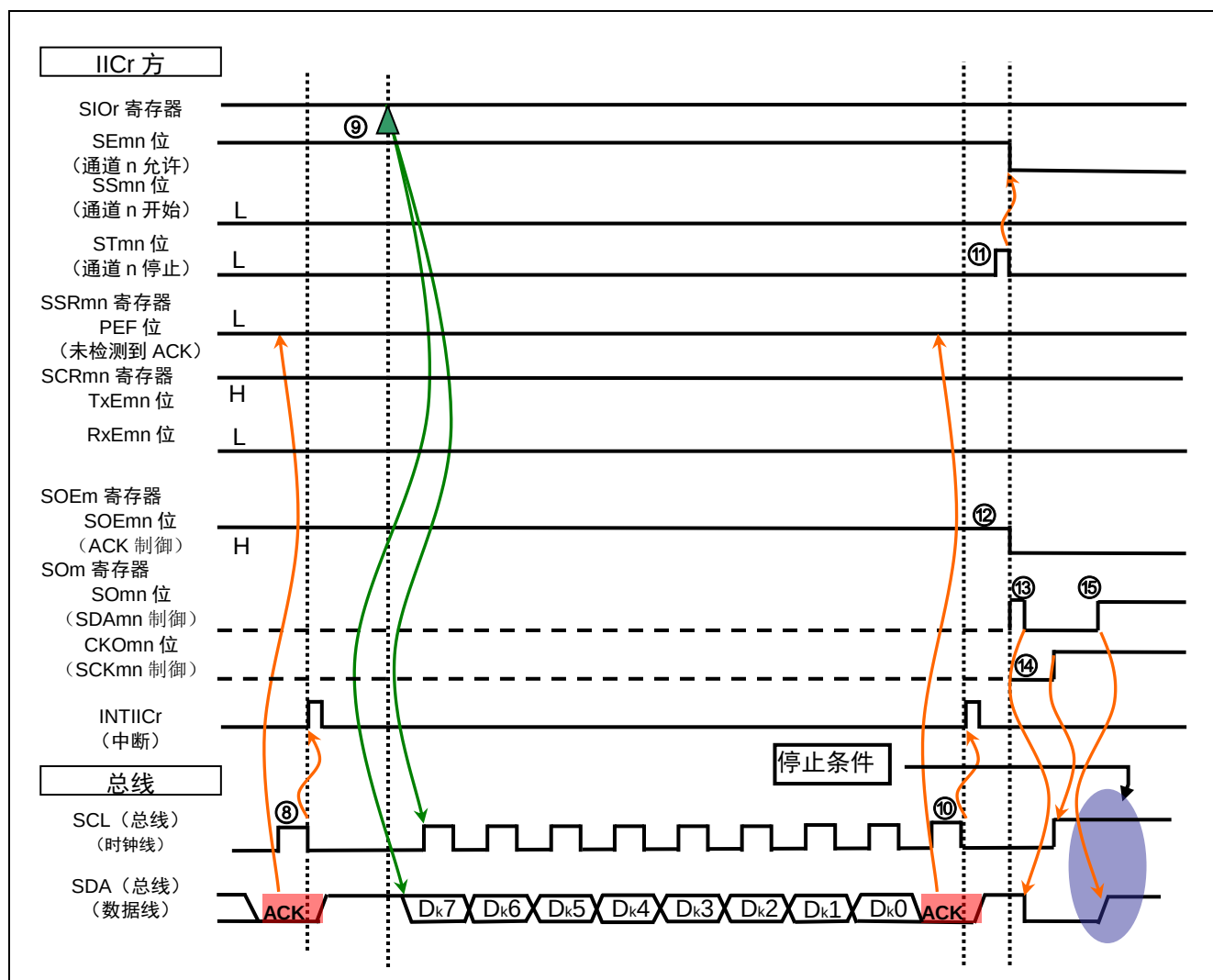


图 1.4 IIC 通信时序图 (主控设备→从属设备通信示例) (3/4)

- ⑧ 一旦数据发送完成, 就会产生 INTIICr。
- ⑨ 通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。如果 PEF 位为 0, 则向 SIOr 寄存器写入发送数据。如果 PEF 位为 1, 则中止发送。
- ⑩ 一旦数据发送完成, 就会产生 INTIICr。
- ⑪ 设置 STmn 寄存器的 STmn 位为 1, 使通道 n 为动作禁止状态。
- ⑫ 为了发行停止条件, 设置 SOEm 寄存器的 SOEmn 位为 0, 以停止输出。
- ⑬ 在停止条件发行准备时, 为了使 SDA 下降, 设置 SOM 寄存器的 SOMn 位为 0。
- ⑭ 在停止条件发行准备时, 为了使 SCL 上升, 设置 SOM 寄存器的 CKOmn 位为 1。
- ⑮ 经过停止条件的准备时间后, 通过设置 SOM 寄存器的 SOMn 位为 1 发行停止条件。

(4) 主控设备→从属设备通信 4 (数据~重新开始条件~地址)

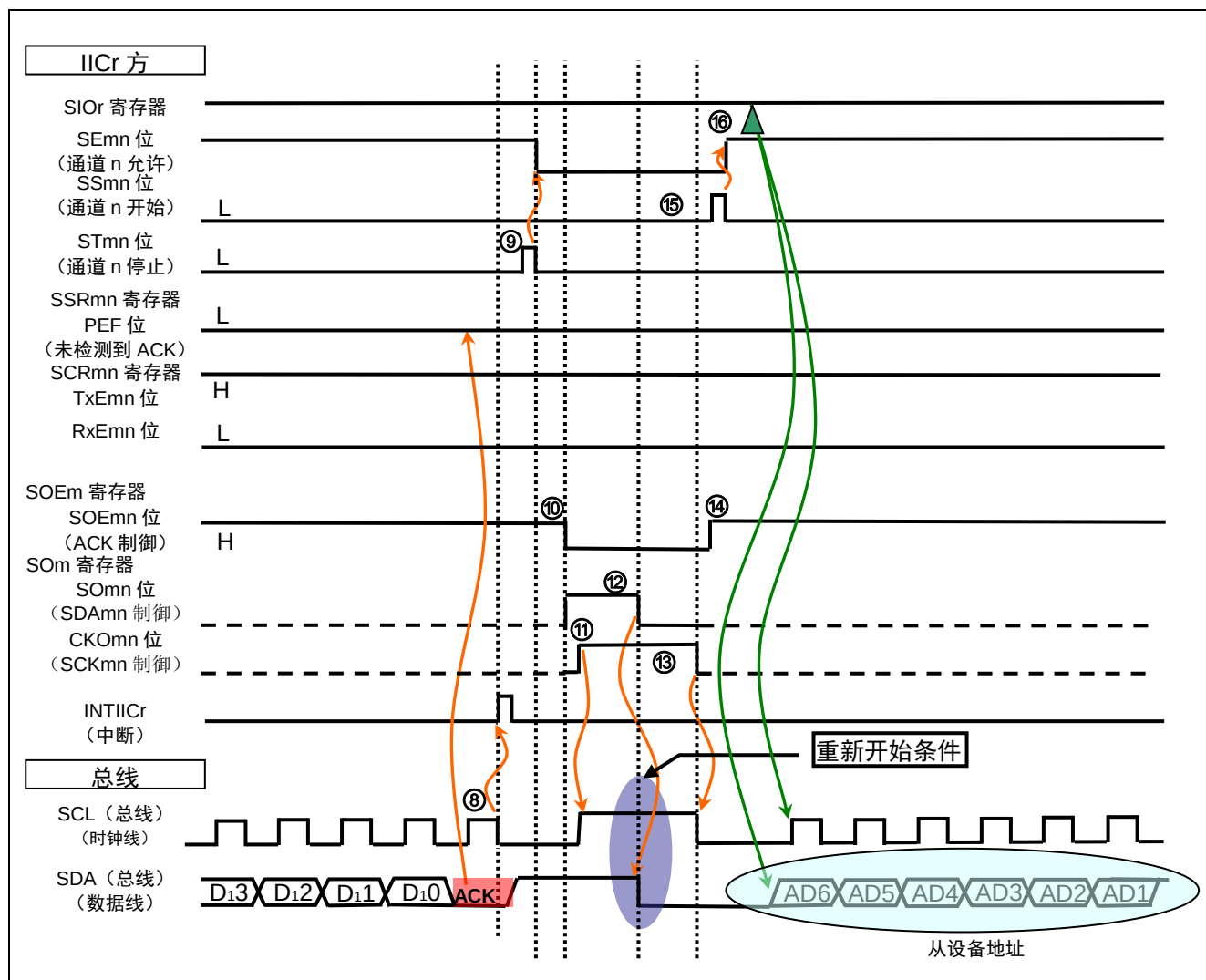


图 1.5 IIC 通信时序图 (主控设备→从属设备通信示例) (4/4)

- ⑧ 一旦数据发送完成,就会产生 INTIICr,通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。
- ⑨ 设置 STm 寄存器的 STmn 位为 1,使通道 n 为动作禁止状态。
- ⑩ 为了发行重新开始条件,设置 SOEm 寄存器的 SOEmn 位为 0,以停止输出。
- ⑪ 在重新开始条件发行准备时,为了跟随着 SDA 使 SCL 也上升,设置 SOm 寄存器的 CKOmn 位为 1。
- ⑫ 通过设置 SOm 寄存器的 SOmn 位为 0 发行重新开始条件。
- ⑬ 经过开始条件的保持时间后,设置 SOm 寄存器的 CKOmn 位为 0,使 SCL 信号下降。
- ⑭ 为了进行通信,设置 SOEm 寄存器的 SOEmn 位为 1,以允许输出。
- ⑮ 设置 SSm 寄存器的 SSmn 位为 1,使通道 n 为动作允许状态。
- ⑯ 向 SIOr 寄存器写入从属设备的地址后,通信启动。

备注 对 EEPROM 读出时,主控设备→从属设备通信的情况下,指定 EEPROM 的单元地址,然后从指定的地址读取数据时使用此处理。

(5) 从属设备→主控设备通信 1 (开始条件~地址~数据)

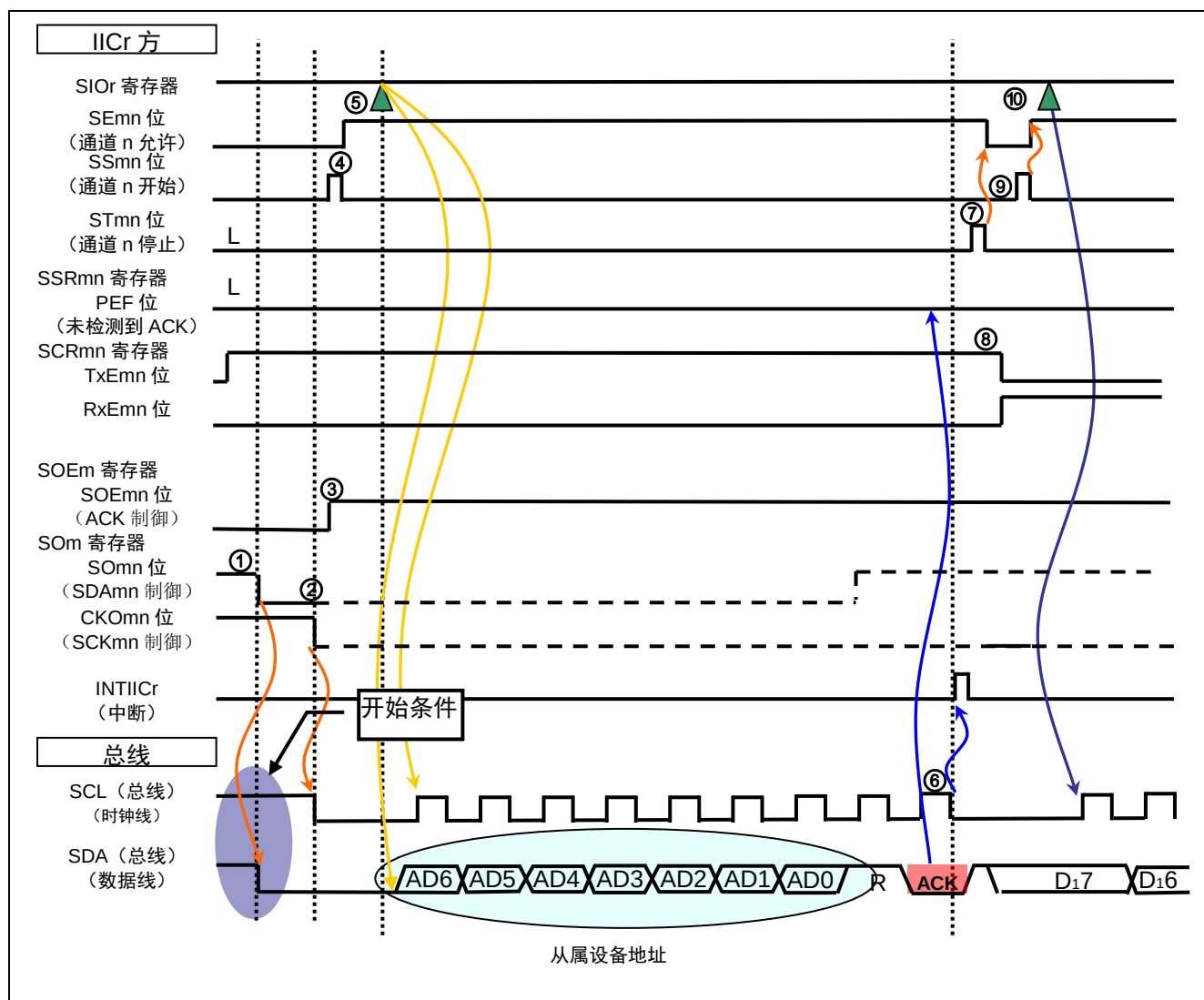


图 1.6 IIC 通信时序图 (从属设备→主控设备通信示例) (1/3)

- ① 如果 IICmn 的初始设定 (此处预先设置了 SCRmn 寄存器的 TxEmn=1, RxEmn=0) 完成了的话, 设定 SOM 寄存器的 SOMn 位为 0, 使 SDA 信号下降, 以产生开始条件。
- ② 经过开始条件的保持时间 (标准模式下为 4.0μs, 快速模式下为 0.6μs) 后, 设置 SOM 寄存器的 CKOmn 位为 0, 使 SCL 信号下降。
- ③ 为了进行通信, 设置 SOEm 寄存器的 SOEmn 位为 1, 以允许输出。
- ④ 设置 SSm 寄存器的 SSmn 位为 1, 使通道 n 为动作允许状态。
- ⑤ 向 SIOr 寄存器写入从属设备的地址后, 通信启动。
- ⑥ 一旦从属设备地址的发送完成, 就会产生 INTIICr。通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。
- ⑦ 为了切换通信方向, 设置 IICmn 的动作为禁止状态。
- ⑧ 变更 SCRmn 寄存器的 TxEmn=0, RxEmn=1。
- ⑨ 设置 IICmn 为动作允许状态。
- ⑩ 通过向 SIOr 寄存器虚写数据 (0FFH), 启动接收。

(6) 从属设备→主控设备通信 2 (地址~数据~数据)

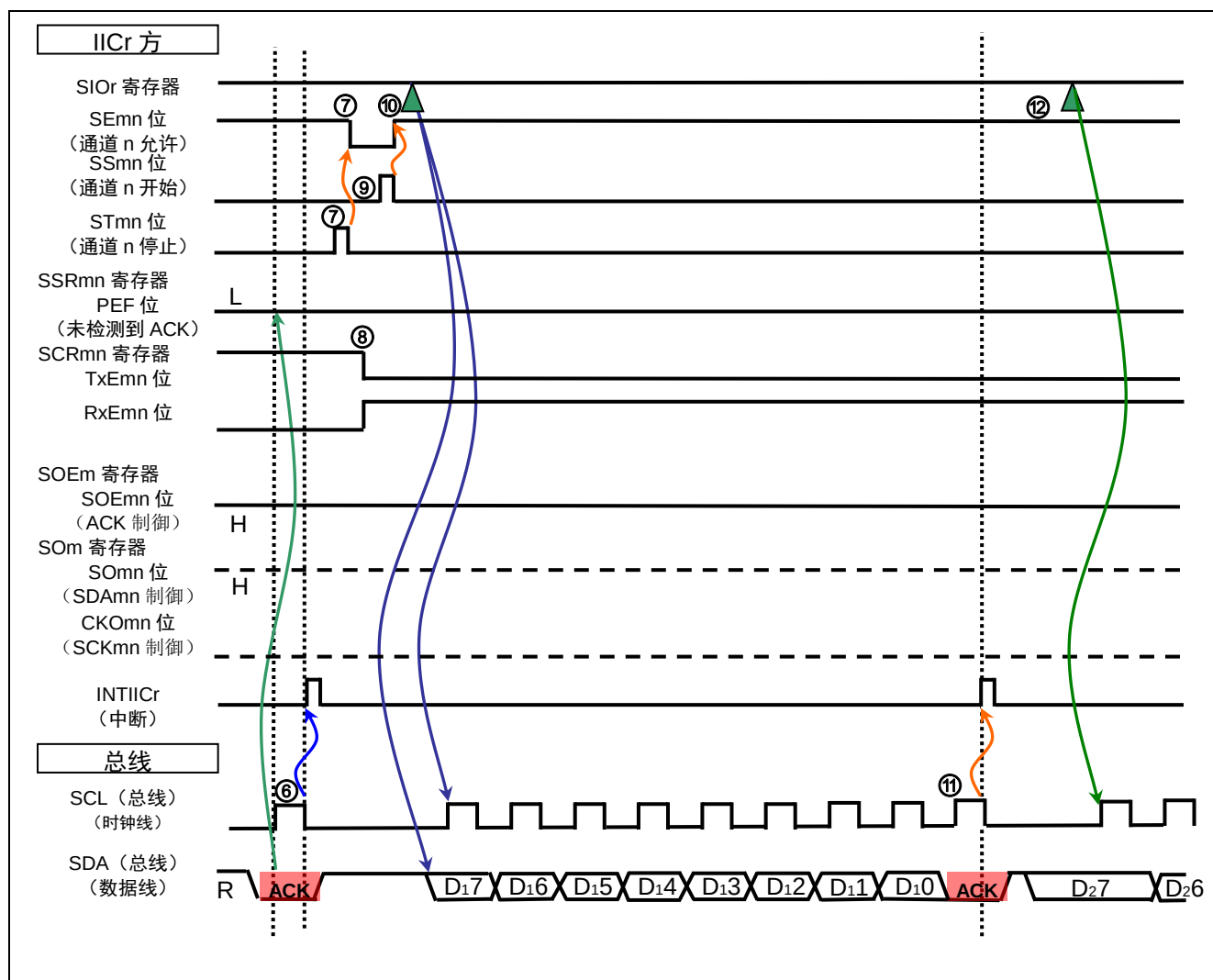


图 1.7 IIC 通信时序图 (从属设备→主控设备通信示例) (2/3)

- ⑥ 一旦从属设备地址发送完成，就会产生 INTIICr。通过 SSRmn 寄存器的 PEF 位对从属设备的 ACK 应答进行确认。
- ⑦ 为了切换通信方向，设置 IICmn 的动作为禁止状态。
- ⑧ 变更 SCRmn 寄存器的 TxEmn=0，RxEmn=1。
- ⑨ 设置 IICmn 为动作允许状态。
- ⑩ 通过向 SIOr 寄存器虚写数据 (0FFH)，启动接收。
- ⑪ 因为 SOEmn=1，在 SCL 的第 9 个时钟进行了 ACK 应答，并发生了接收完成的 INTIICr。
- ⑫ 为了接收下一个数据 (不是最后的数据)，向 SIOr 寄存器虚写数据 (0FFH)，以启动接收。

(7) 从属设备→主控设备通信 3 (数据~数据~停止条件)

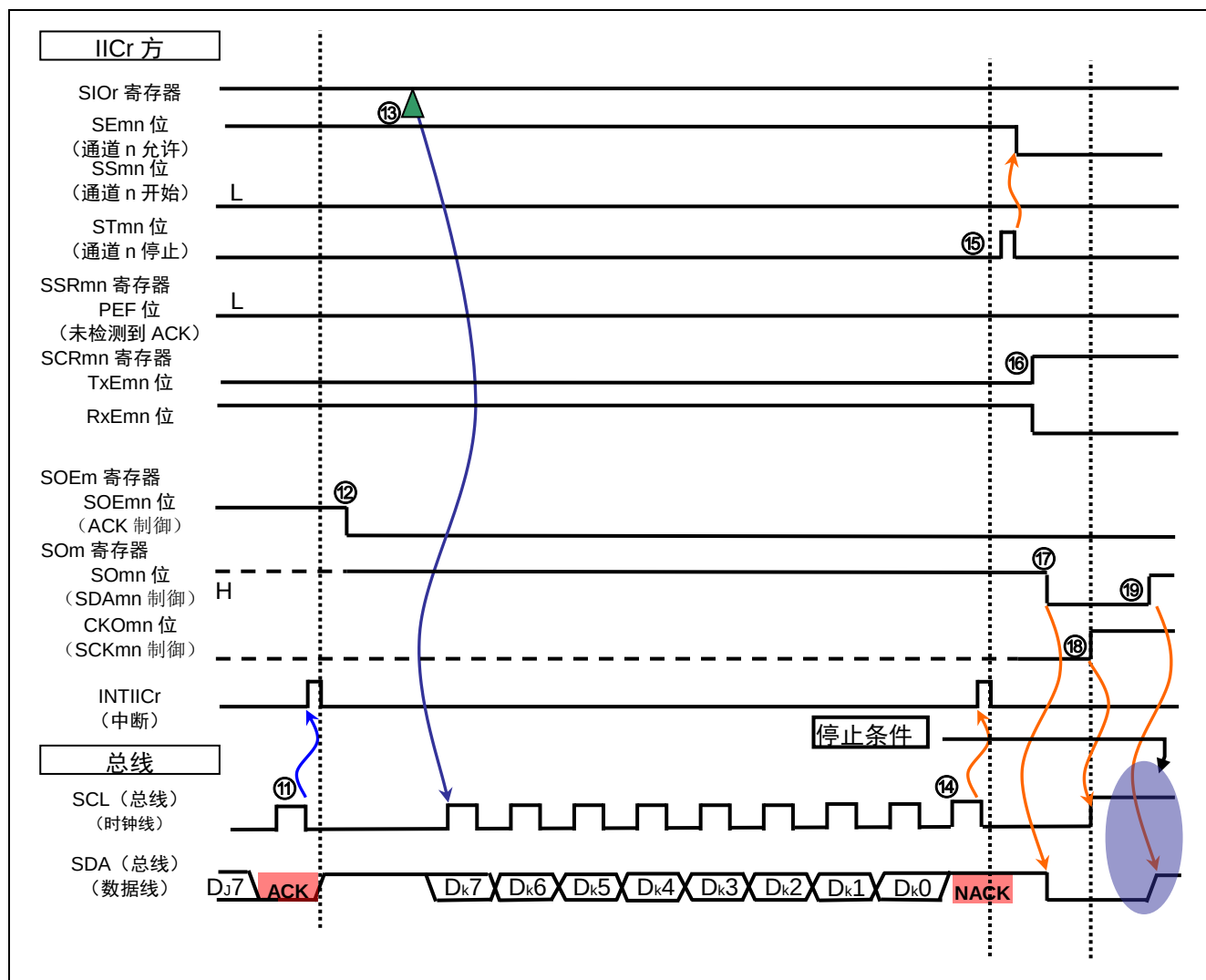


图 1.8 IIC 通信时序图 (从属设备→主控设备通信示例) (3/3)

- ⑪ 因为 SOEmn=1, 在 SCL 的第 9 个时钟进行了 ACK 应答, 并发生了接收完成的 INTIICr。
- ⑫ 为了对最后的接收数据应答 NACK, 设置 SOEm 寄存器的 SOEmn 位为 0, 以停止串行输出。
- ⑬ 通过向 SIOr 寄存器虚写数据 (0FFH), 启动接收。
- ⑭ 因为 SOEmn=0, 在 SCL 的第 9 个时钟进行了 NACK 应答, 并发生了接收完成的 INTIICr。
- ⑮ 设置 STmn 寄存器的 STmn 位为 1, 使通道 n 为动作禁止状态。
- ⑯ 为了准备下一次通信, 将 SCRmn 寄存器设置为和初始设定相同的状态。
- ⑰ 在停止条件发行准备时, 为了使 SDA 下降, 设置 SOmn 寄存器的 SOmn 位为 0。
- ⑱ 在停止条件发行准备时, 为了使 SCL 上升, 设置 SOmn 寄存器的 CKOmn 位为 1。
- ⑲ 经过停止条件的准备时间后, 通过设置 SOmn 寄存器的 SOmn 位为 1 发行停止条件。

2. 动作确认条件

本篇应用说明中的参考例程，是在下面的条件下进行动作确认的。

表 2.1 动作确认条件

项目	内容
所用单片机	R7F0C004M
工作频率	● 高速内部振荡器（HOCO）时钟：24MHz ● CPU/外围硬件时钟：24MHz
工作电压	5.0V（工作电压范围：2.9V~5.5V） LVD 工作模式：复位模式 ● V_{LVD}（上升沿）：2.81V ● V_{LVD}（下降沿）：2.75V
综合开发环境	瑞萨电子开发 CS+ for CA, CX V3.00.00
C 编译器	瑞萨电子开发 CA78K0R V1.70

3. 相关应用说明

与本应用说明相关的应用说明如下所示。请同时参考。

R7F0C009 定时器阵列单元（间隔定时器）（R01AN2439CC）应用说明

4. 硬件说明

4.1 硬件配置示例

本篇应用说明中使用的硬件配置示例，如图 4.1 所示。

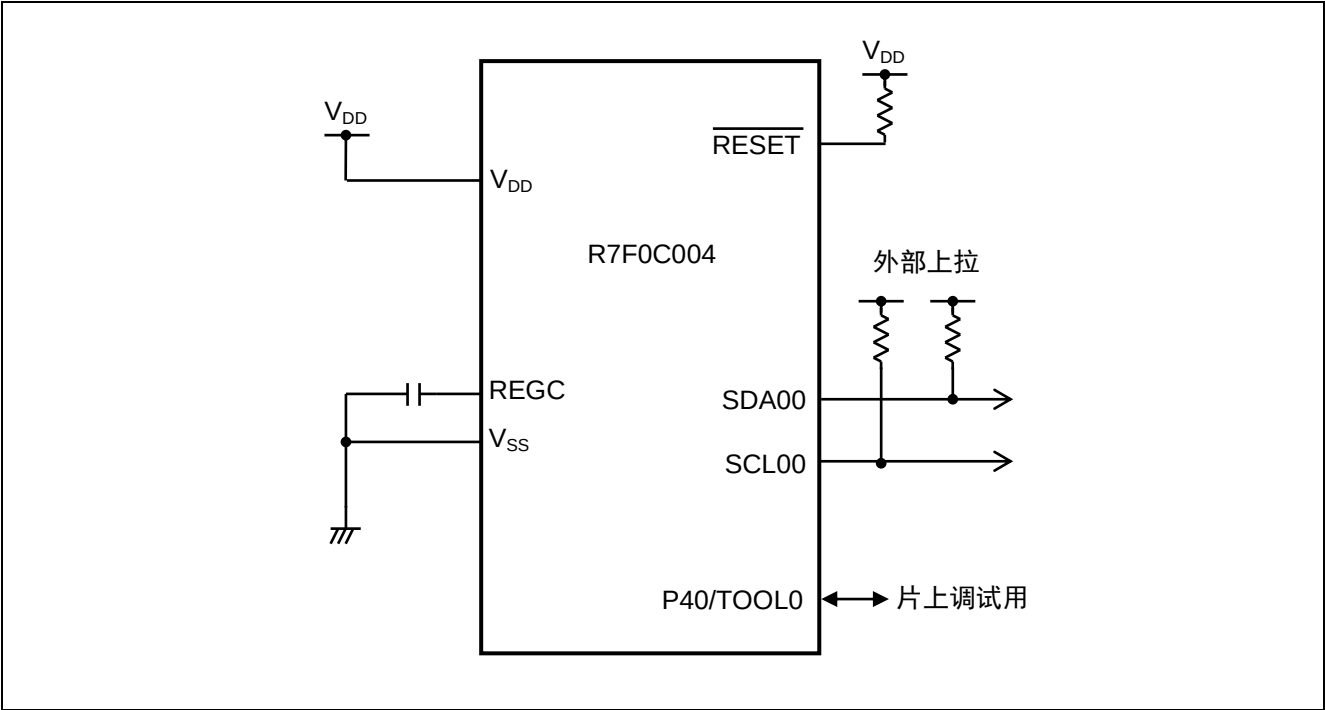


图 4.1 硬件配置

注意 1.上述硬件配置图是为了表示硬件连接情况的简化图。在实际电路设计时，请注意根据系统具体要求进行适当的引脚处理,并满足电气特性的要求(输入专用引脚请注意分别通过电阻上拉到 V_{DD} 或是下拉到 V_{SS})。

2.请将 V_{DD} 设置为大于 LVD 设定的复位解除电压 (V_{LVD})。

4.2 使用引脚一览表

使用的引脚及其功能，如表 4.1 所示。

表 4.1 使用的引脚及其功能

引脚名	输入/输出	内容
P16/SCL00	输入/输出	IIC00 的串行时钟输出引脚
P17/SDA00	输入/输出	IIC00 的串行数据收发用引脚

5. 软件说明

5.1 操作概要

本篇应用说明中, 通过串行接口 IIC00, 使用 IIC 主控设备收发信功能进行 EEPROM 的控制 (写入、读出) 操作。考虑到此控制作为 API 的使用, 尽量以中断进行处理。

(1) 对串行阵列单元 0 的通道 0 以简易 IIC 模式进行初始设定。

<设定条件>

- 设定运行时钟为 CK00 (24MHz)。
- 设定运行模式为简易 IIC。
- 设定传送结束中断。
- 设定数据和时钟的相位为类型 1。
- 设定数据长度为 8 位, 停止位为 1 位, 无校验位, MSB 优先传送。
- 设定传送时钟为快速模式的 387kHz。
- 设定 SO00, CKO00 为 1。
- 设定 P16/SCL00 引脚为传送时钟输出用, P17/SDA00 引脚为数据发送/接收用。

(2) 设定定时器阵列单元的通道 2 为 100 μ s 的间隔定时器, 用于确认写入完成。

(3) 将使用的 EEPROM (16K 位) 的信息值复制到处理用信息值的结构体中。

(4) 发行停止条件, 使总线处于开放状态。

(5) 准备出 256 字节的写入数据 (递增形式)。

(6) 设定存取用信息值 (结构体 g_PARAI)。

(7) 从 EEPROM 的 0x400 地址开始写入 256 字节的数据。

(8) 读出包括已写入数据前后各 1 字节在内的数据。

(9) 准备出相同的 16 个字节的数据 (0xkk)。(kk=00, 11, 22……77)

(10) 从 EEPROM 的 0xk00 地址开始写入这 16 个字节。

(11) 使 k 的值从 0~7 改变, 重复 (9) ~ (10) 的操作。

(12) 读出包括 (9) ~ (10) 中已写入数据前后各 8 字节在内的共 32 字节的数据。

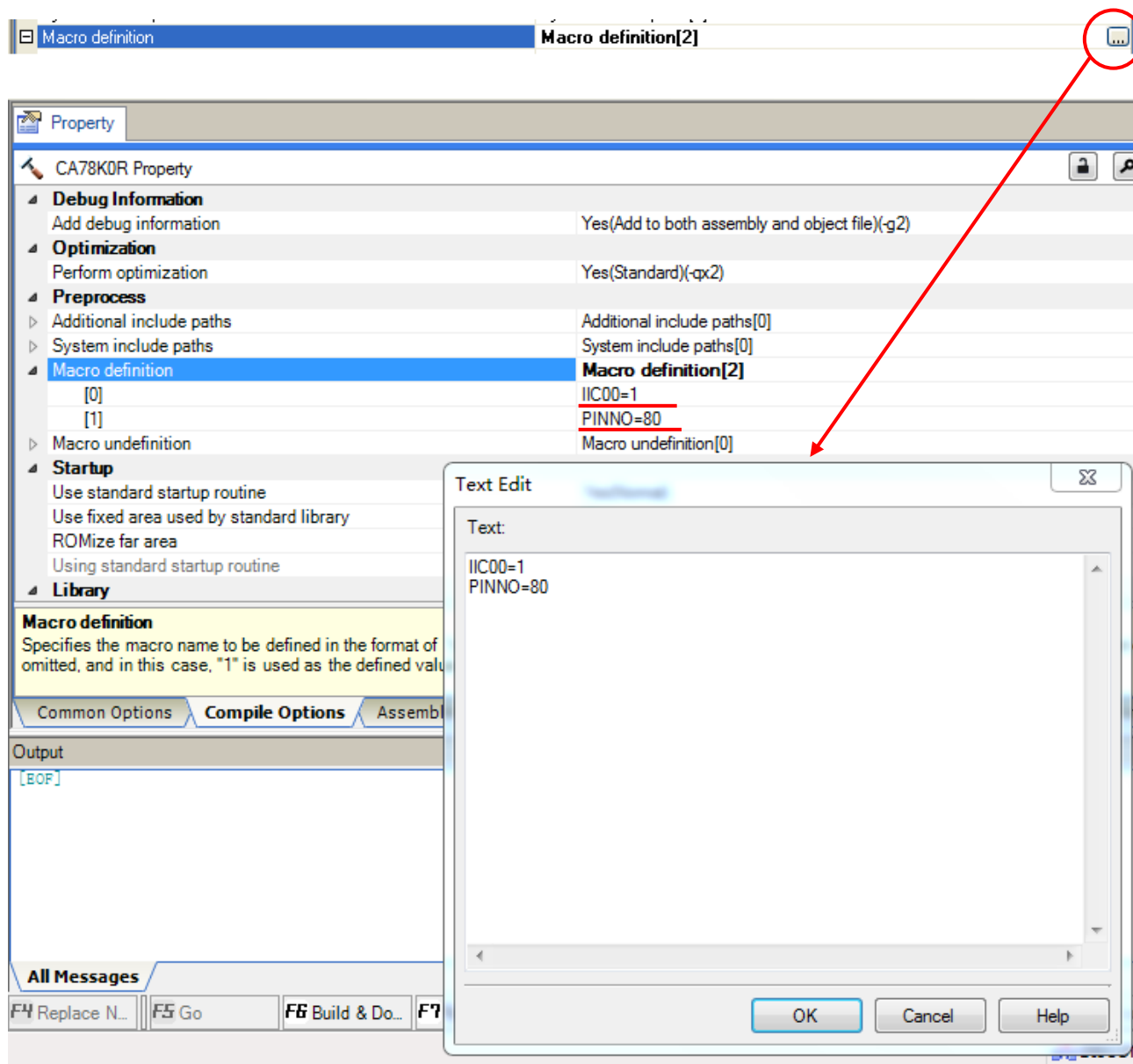
注意 本参考例程是使用 R7F0C004 的简易 IIC 通过 IIC 总线控制 EEPROM（R1EX24016，R1EX24032）的例子。对使用的通道或 EEPROM 变更时请详细评价再使用。

备注 在使用的项目中，使用 [Compile Options] 的 [Macro definition] 功能指定使用的简易 IIC 的通道和产品的引脚数。

Macro definition [0] 选择了 IIC00 作为使用的简易 IIC 的通道。例如，当使用通道 10 时，把这个设置变更为 [IIC10=1]。

Macro definition [1] 指定使用的产品的引脚数。在确认以 Macro definition [0] 指定的通道在目标产品中是否存在的基础上使用。

变更这些设置时，点击 [Macro definition] 右侧的按钮，打开 Text Edit 页面，在此处进行编辑。



5.2 选项字节设定一览表

选项字节的设定如表 5.1 所示。

表 5.1 选项字节设定

地址	设定值	内容
000C0H/010C0H	01101110B	看门狗定时器动作停止 (复位解除后, 计数停止)
000C1H/010C1H	01111111B	LVD 工作模式: 复位模式 ● V_{LVD} (上升沿): 2.81V ● V_{LVD} (下降沿): 2.75V
000C2H/010C2H	11100000B	HS 模式、HOCO: 24MHz
000C3H/010C3H	10000101B	允许片上调试

5.3 常数一览表

参考例程中使用的常数如表 5.2, 5.3 所示。用于指定使用的通道号的宏名一览表如表 5.4 所示。

表 5.2 参考例程中使用的常数 (1/2)

常数名	设定值	内容
PAGE_16	0x000F	2K~16K 位 EEPROM 每页的数据数
PAGE_32	0x001F	32K、64K 位 EEPROM 每页的数据数
PAGE_64	0x003F	128K、256K 位 EEPROM 每页的数据数
PAGE_128	0x007F	512K 位 EEPROM 每页的数据数
MEMORY_2K	0x0001	2K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_4K	0x0002	4K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_8K	0x0004	8K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_16K	0x0008	16K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_32K	0x0010	32K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_64K	0x0020	64K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_128K	0x0040	128K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_256K	0x0080	256K 位 EEPROM 的容量 (以 256 字节为单位)
MEMORY_512K	0x0100	512K 位 EEPROM 的容量 (以 256 字节为单位)
ADDR0BIT	0b00000000	从属设备地址不用作单元地址
ADDR1BIT	0b00000001	通过从属设备地址的位 1 指定 A8
ADDR2BIT	0b00000011	通过从属设备地址的位 2、1 指定 A9、8
ADDR3BIT	0b00000111	通过从属设备地址的位 3~1 指定 A10~8
I2C_OK	0x00	正常完成
PARA_ERR	0x20	信息值错误
NO_ACK1	0x40	未收到对从属设备地址的 ACK 应答
NO_ACK2	0x41	未收到对 EEPROM 地址的 ACK 应答
NO_ACK3	0x42	未收到对发送数据的 ACK 应答
BUS_ERR	0x60	总线不能置为开放状态 (SDA 为高电平)
SVAMSK	0b11111110	用于屏蔽从属设备地址的位 0 的数据
SCLLOWW	0x04	SCL 的低电平时间计时用数据
SCLHIGHW	0x02	SCL 的高电平时间计时用数据
RETRYCNT	0x09	SCL 的空时钟的脉冲数

表 5.3 参考例程中使用的常数 (2/2)

常数名	设定值	内容
R1EX24002A	0x00	用于指定使用的 EEPROM 的常数。 在枚举型常数 eeprom_name 中定义, 为了从 eeprom_info 型的结构体 EEPROM_ADDRESS 中查找对应的 EEPROM 的信息值而使用的。
R1EX24004A	0x01	
R1EX24008A	0x02	
R1EX24016A	0x03	
R1EX24032A	0x04	
R1EX24064A	0x05	
R1EX24128A	0x06	
R1EX24256A	0x07	
R1EX24512A	0x08	

表 5.4 参考例程中使用的宏名

宏名	设定值 (通道 0)	内容
SAUmEN	SAU0EN	使用的外围允许寄存器
SPSm	SPS0	串行时钟选择寄存器
SMRmn	SMR00	串行模式寄存器
SCRmn	SCR00	串行通信运行设定寄存器
SDRmn	SDR00	串行数据寄存器
SIOr	SIO00	串行数据寄存器 (收发数据用)
SSRmn	SSR00	串行状态寄存器
SIRmn	SIR00	串行标志清除触发寄存器
SSmL	SS0L	串行通道开始寄存器
STmL	ST0L	串行通道停止寄存器
TRGONn	0b00000001	用于触发的位
SOEmL	SOE0L	串行输出允许寄存器
SOEON	TRGONn	用于设定串行输出允许寄存器的位
SOEOFF	(uint8_t)(~SOEON)	用于清除串行输出允许寄存器的位
SOm	SO0	串行输出寄存器
SDAHIGH	TRGONn	用于设置 SDA 上升的位
SDALOW	~SDAHIGH	用于设置 SDA 下降的位
SCLHIGH	TRGONn × 0x100	用于设置 SCL 上升的位
SCLLOW	~SCLHIGH	用于设置 SCL 下降的位
IICIFr	IICIF00	中断请求标志寄存器
IICMKr	IICMK00	中断请求屏蔽寄存器
PFSEG_SDAr	PFSEG5.3	SDA 用 LCD 端口功能寄存器的位
PFSEG_SCLr	PFSEG5.2	SCL 用 LCD 端口功能寄存器的位
PM_IICr	PM1	IICr 用端口模式寄存器 (指定 SDA 信号的使用)
PM_SDAr	PM1.7	SDA 用端口模式寄存器的位
PM_SCLr	PM1.6	SCL 用端口模式寄存器的位
POM_IICr	POM1	端口输出模式寄存器
P_IICr	P1	IICr 用端口
P_SDAr	P1.7	SDA 用端口
P_SCLr	P1.6	SCL 用端口
SDAINMODE	0b10000000	用于指定端口模式寄存器为输入模式的位
SDAOUTMODE	0b00111111	用于指定端口模式寄存器为输出模式的位
SDASCLON	(uint8_t)(~SDAOUTMODE)	用于端口设定的位
IICPR0r	IICPR000	中断优先级设定寄存器
IICPR1r	IICPR100	

5.4 变量一览表

全局变量如表 5.5 所示。

另外，从 g_eeprom_type 开始为模块内的全局变量。

表 5.5 全局变量

类型	变量名	内容	使用此变量的函数
结构体 eeprom_paraA16	g_PARAI	用于指定 EEPROM 存取用的信息值	main() check_EEPROM_Addr()
uint8_t	g_comstatus	运行信息 / 结果标志	R_EEPROM_R() R_EEPROM_wait_read() R_IICr_Tx_addr1() R_IICr_Tx_addr2() R_IICr_Rx_RST() R_IICr_RxData_ST() R_IICr_RxData() R_IICr_Rx_LastData() R_EEPROM_W() R_EEPROM_wait_write() R_IICr_TxDataST() R_IICr_TxData() R_EEPROM_next_page() IINTIICr()
uint8_t 数组 (256)	g_data_bufferW1	写入数据缓冲	main()
uint8_t 数组 (256)	g_data_bufferW2	写入数据缓冲	main()
uint8_t 数组 (512)	g_data_bufferR1	读出数据缓冲	main()

类型	变量名	内容	使用此变量的函数
uint8_t	g_eeprom_type	使用的 EEPROM 的编号	R_device_select()
结构体 eeprom_paraA16 uint8_t slaveaddr; uint16_t eepromaddr; uint8_t *bufferaddr; uint16_t number;	g_PARAA	存取 EEPROM 用的信息值	R_EEPROM_R() R_IICr_Tx_addr1() R_IICr_Tx_addr2() R_IICr_Rx_RST() R_IICr_RxData_ST() R_IICr_RxData() R_IICr_Rx_LastData() R_EEPROM_W() R_EEPROM_wait_write() R_IICr_TxDataST() R_IICr_TxData() R_EEPROM_Devide() SINTTM02()
结构体 eeprom_paraA16	g_PARAC	用于指定 EEPROM 存取的信息值的副本	check_EEPROM_Addr() R_EEPROM_R() R_EEPROM_W() R_IICr_TxData() get_slave_Addr() R_EEPROM_Devide()
结构体 eeprom_info uint16_t page_size; uint16_t rom_size; uint8_t addr_mask; uint8_t mask2;	EEPROM_Info	用于保存使用的 EEPROM 的信息值（处理用）	R_device_select() R_IICr_Tx_addr1() get_slave_Addr() R_EEPROM_Devide()

5.5 函数一览表

函数一览表如表 5.6 所示。

表 5.6 函数一览表

函数名	概要
R_device_select	指定使用的 EEPROM。
R_EEPROM_R	根据参数传递的指针所指向的结构体中的存取用信息值，从 EEPROM 中读出数据。
R_EEPROM_wait_read	等待 EEPROM 的读出完成。
R_EEPROM_W	根据参数传递的指针所指向的结构体中的存取用信息值，向 EEPROM 写入数据。
R_EEPROM_wait_write	等待对 EEPROM 的写入完成。
check_EEPROM_Addr	检查被指定的信息值是否超出了 EEPROM 的容量范围。如果在容量范围内，则复制存取用信息值。
get_slave_Addr	EEPROM 的容量为 4K~16K 位的情况下，将单元地址的高位组合到从属设备地址中。
R_EEPROM_Devide	为了使写入时写入数据包含在一页内，修正 1 次写入的字节数的信息值。
R_IICr_Tx_addr1	进行从属设备地址的发送完成的处理，发送 EEPROM 的单元地址。
R_IICr_Tx_addr2	当 EEPROM 的单元地址为 2 字节时，发送低位地址。
R_IICr_Rx_RST	为了在发送完 EEPROM 的单元地址后读出数据，以接收模式重新开始。
R_IICr_RxData_ST	为了接收数据，设定 TxE→0、RxE→1 后启动数据接收。当读出数据为 1 字节时，设定 NACK 应答后启动接收。
R_IICr_RxData	将接收的数据保存到缓冲区，启动下一次的数据接收。
R_IICr_Rx_LastData	把最后的接收数据保存到缓冲区。为准备下一次通信，设定 TxE→1，RxE→0，然后发行停止条件，结束处理。
R_IICr_TxDataST	开始发送向 EEPROM 的写入数据。
R_IICr_TxData	数据发送完成后，对一页的写入数据全部发送完成了的话，发行停止条件，指示 EEPROM 的内部写入。全部数据的处理都完成了的话，结束处理。还有剩余数据的话，准备下一页的写入，启动用于写入完成等待的定时器。
R_EEPROM_next_page	因为 EEPROM 的内部写入已完成，停止定时器，进行 EEPROM 下一页单元地址的发送。

函数名	概要
R_IICr_StartCond	操作 S0m 寄存器以发行开始条件，并允许 IICr 的运行。
R_IICr_StopCond	发行停止条件，并确认总线是否被释放。如果总线未被释放，在 SCL 上输出空时钟，再次发行停止条件。
R_IICr_send_Stop	禁止 IICr 运行，操作 S0m 寄存器以发行停止条件。
R_IICr_wait_bus	在 SCL 信号上输出 9 个脉冲的空时钟后，确认 SDA 信号是否开放（变为高电平）。
R_IICr_SCL_pulse	在 SCL 信号上输出空时钟。
R_IICr_SCL_high	使 SCL 信号上升，等待高电平宽度的时间。
R_IICr_SCL_low	使 SCL 信号下降，等待低电平宽度的时间。
R_IICr_NACK	确认来自从属设备的 ACK/NACK 应答。
R_IICr_SCL_Time	等待 SCL 信号的低电平时间。
R_IICr_SCL_highTime	等待 SCL 信号的高电平时间。
IINTIICr	在 IICr 的发送完成中断中，确认从属设备的 ACK 应答，并跳转到下一个处理。
IINTTM02	在 TM02 的间隔中断处理中，发送从属设备地址，以确认写入完成。
R_IICr_Init	进行 IICr 和 TM02 的初始设定。

5.6 函数说明

以下所示为参考例程中的函数说明。

[函数名] R device select

概要	指定使用的 EEPROM
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	MD_STATUS R_device_select(enum eeprom_name);
说明	将通过参数指定的 EEPROM 的信息值复制到结构体 EEPROM_Info 中。
参数	[EERROM 的名称] 枚举型常数 eeprom_name 中列出的名称
返回值	[I2C_OK]时：正常结束 [PARA_ERR]时：指定的名称错误
参考	无

[函数名] R_IICr_StopCond

概要	生成停止条件
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	MD_STATUS R_IICr_StopCond(void);
说明	生成 IICr 的停止条件。总线不能释放时，在 SCL 上输出 9 个空时钟，然后确认 SDA 变成了高电平后，再次发行停止条件。
参数	无
返回值	[I2C_OK]时：总线释放完成 [BUS_ERR]时：总线未能释放
参考	无

[函数名] R EEPROM R

概要	从 EEPROM 读出的开始要求的处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	void R_EEPROM_R(struct eeprom_paraA16 *PARA);
说明	根据作为参数的指针所指向的结构体中存取用信息值的指定，进行相应的读出操作。
参数	*PARA 指向 eeprom_paraA16 型的结构体的指针
返回值	无
参考	读出完成的等待和结果在 R_EEPROM_wait_read 中进行。

[函数名] R_EEPROM_wait_read

概要	从 EEPROM 读出完成的等待处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	MD_STATUS R_EEPROM_wait_read (void);
说明	等待由 R_EEPROM_R 启动的读出完成。
参数	无 -
返回值	[I2C_OK]时: 读出正常结束 [PARA_ERR]时: 指定的信息值在 EEPROM 范围外 [NO_ACK1]时: 没有对从属设备地址的 ACK 应答 [NO_ACK2]时: 没有对 EEPROM 地址的 ACK 应答
参考	无

[函数名] R_EEPROM_W

概要	向 EEPROM 写入的开始处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	void R_EEPROM_W(struct eeprom_paraA16 *PARA);
说明	根据作为参数的指针所指向的结构体中存取用信息值的指定, 进行相应的写入操作。
参数	*PARA 指向 eeprom_paraA16 型的结构体的指针
返回值	无
参考	写入完成的等待和结果在 R_EEPROM_wait_write 中进行。

[函数名] R_EEPROM_wait_write

概要	向 EEPROM 写入完成的等待处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	MD_STATUS R_EEPROM_wait_write(void);
说明	等待由 R_EEPROM_W 启动的向 EEPROM 写入的完成。
参数	无 -
返回值	[I2C_OK]时: 写入正常结束 [PARA_ERR]时: 指定的信息值在 EEPROM 范围外 [NO_ACK1]时: 没有对从属设备地址的 ACK 应答 [NO_ACK2]时: 没有对 EEPROM 地址的 ACK 应答 [NO_ACK3]时: 没有对发送数据的 ACK 应答
参考	无

[函数名] check_EEPROM_Addr

概要	EEPROM 存取区域的检查处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	static MD_STATUS check_EEPROM_Addr(struct eeprom_paraA16 *PARA);
说明	检查根据存取 EEPROM 的信息值进行读/写的区域是否在 EEPROM 范围内。
参数	*PARA 指向 eeprom_paraA16 型的结构体的指针
返回值	[I2C_OK]时: 存取区域在 EEPROM 的范围内。 [PARA_ERR]时: 存取区域不在 EEPROM 的范围内。
参考	无

[函数名] get_slave_Addr

概要	EEPROM 的从属设备地址的计算处理
头文件	r_cg_macrodriver.h
声明	static void get_slave_Addr(void);
说明	将 4K~16K 位 EEPROM 的单元地址的高位 1~3 位组合到从属设备地址中。
参数	无 -
返回值	无
参考	将结构体 g_PARAC 的成员 eepromaddr 的值组合到成员 slaveaddr 中。

[函数名] R_EEPROM_Devide

概要	EEPROM 写入时的页分割处理
头文件	r_cg_macrodriver.h
声明	static void R_EEPROM_Devide(void);
说明	根据使用的 EEPROM 的页, 分割写入的数据。此次的写入用信息值设定到结构体 g_PARAA 中, 剩余数据的信息设定到 g_PARAC 中。
参数	无 -
返回值	无
参考	结构体 g_PARAC 的成员 eepromaddr、bufferaddr 是下次的写入用信息值, number 是下次及之后的数据数。

[函数名] R_IICr_Tx_addr1

概要	EEPROM 的地址发送处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_Tx_addr1(void);
说明	在 32K 位及以上 EEPROM 中, 发送单元地址的高位。在 16K 位及以下的产品中, 发送 1 字节的单元地址。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_Tx_addr2

概要	EEPROM 单元地址的低位的发送处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_Tx_addr2(void);
说明	在 32K 位及以上的 EEPROM 中，发送地址的低位字节。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_Rx_RST

概要	重新开始处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_Rx_RST(void);
说明	为了在发送完 EEPROM 的单元地址后读出数据，以接收模式重新开始。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_RxData_ST

概要	数据接收开始处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_RxData_ST(void);
说明	读出模式中的从属设备地址发送完成后，将 IICr 从发送（TxE=1, RxE=0）变更为接收（TxE=0, RxE=1），启动接收（在 SIOr 上写入虚数据）。读出数据为 1 字节时，设定 NACK 应答，然后启动数据接收。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_RxData

概要	数据接收处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_RxData (void);
说明	把接收到的数据保存到缓冲区，启动下一次的数据接收。读出数据剩余 1 字节时，设定 NACK 应答，然后启动数据接收。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_Rx_LastData

概要	最后数据的接收完成处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_Rx_LastData (void);
说明	将接收的数据保存到缓冲区，停止 IICr 的动作。为准备下一次通信，设定 TxE→1，RxE→0，然后发行停止条件，以结束处理。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_TxDataST

概要	数据发送开始处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_TxDataST (void);
说明	EEPROM 地址发送完成后，开始数据的发送。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_TxData

概要	数据发送处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_TxData (void);
说明	1 字节的数据传送完成了的话，就发送下一个数据。如果对一页的写入数据全部发送结束了的话，则发行停止条件，指示 EEPROM 的内部写入。如果全部数据的处理完成了的话，结束处理。还有剩余数据的话，就准备下一页的写入，启动用于写入完成等待的定时器。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_EEPROM_next_page

概要	写入等待完成，下一页的地址指定处理
头文件	r_cg_userdefine.h
声明	static void R_EEPROM_next_page(void);
说明	页写入完成后，停止写入等待定时器，进行 EEPROM 下一页的单元地址的发送。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用。检查到写入完成后，以对从属设备地址发送的 ACK 应答启动。

[函数名] R_IICr_StartCond

概要	开始条件的发行处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_StartCond(void);
说明	暂时禁止 IICr 的动作，发行开始条件，然后再允许 IICr 的动作。
参数	无 -
返回值	无
参考	IINTIICr 处理中使用

[函数名] R_IICr_StopCond

概要	总线释放处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	MD_STATUS R_IICr_StopCond(void);
说明	发行停止条件，确认总线是否被释放。总线未被释放的话，在 SCL 上输出空时钟，然后再次发行停止条件。
参数	无 -
返回值	[I2C_OK]时：完成停止条件的发行 [BUS_ERR]时：SDA 信号未变为高电平
参考	无

[函数名] R_IICr_send_Stop

概要	停止条件的发行处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_send_Stop (void);
说明	禁止 IICr 动作，操作 S0m 寄存器，以发行停止条件。
参数	无 -
返回值	无
参考	内部发行停止条件时使用。

[函数名] R_IICr_wait_bus

概要	总线释放确认处理
头文件	r_cg_macrodriver.h r_cg_userdefine.h
声明	static MD_STATUS R_IICr_wait_bus(void);
说明	停止 IICr 的状态下，在 SCL 上输出 9 个脉冲的空时钟后，确认 SDA 信号。
参数	无 -
返回值	[I2C_OK]时：总线能够释放。 [BUS_ERR]时：总线不能释放（SDA 信号未变为高电平）。
参考	无

[函数名] R_IICr_SCL_pulse

概要	SCL 信号的空时钟输出处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_SCL_pulse(void);
说明	操作 S0m 寄存器, 在 SCL 上输出低脉冲。
参数	无 -
返回值	无
参考	无

[函数名] R_IICr_SCL_high

概要	使 SCL 信号上升的处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_SCL_high(void);
说明	操作 S0m 寄存器, 使 SCL 上升, 并等待 SCL 的高电平期间。
参数	无 -
返回值	无
参考	无

[函数名] R_IICr_SCL_low

概要	使 SCL 信号下降的处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_SCL_low(void);
说明	操作 S0m 寄存器, 使 SCL 下降, 并等待 SCL 的低电平期间。
参数	无 -
返回值	无
参考	无

[函数名] R_IICr_NACK

概要	来自从属设备的 ACK/NACK 应答的确认处理
头文件	r_cg_macrodriver.h
声明	static MD_STATUS R_IICr_NACK(void);
说明	返回 SSRmn 寄存器的位 1 (PEF 位=NACK 标志位)。
参数	无 -
返回值	[0x00]时: 有 ACK 应答 [0x02]时: 无 ACK 应答
参考	只检查状态标志位, 不清除。

[函数名] R_IICr_SCL_Time

概要	SCL 信号的低电平期间等待处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_SCL_Time(void);
说明	等待 SCL 信号的低电平期间。
参数	无 -
返回值	无
参考	无

[函数名] R_IICr_SCL_highTime

概要	SCL 信号的高电平期间等待处理
头文件	r_cg_userdefine.h
声明	static void R_IICr_SCL_highTime(void);
说明	等待 SCL 信号的高电平期间。
参数	无 -
返回值	无
参考	无

[函数名] IINTIICr

概要	IICr 传送完成中断
头文件	r_cg_userdefine.h
声明	__interrupt void IINTIICr(void);
说明	由 INTIICr 启动, 根据通信的状况启动必要的处理。除了 EEPROM 的写入完成等待, 在检出来自从属设备的 NACK 应答时, 会根据处理的状态在 g_comstatus 上设定错误标志。
参数	无 -
返回值	无
参考	无

[函数名] IINTTM02

概要	100μs 间隔定时完成中断
头文件	r_cg_userdefine.h
声明	__interrupt void IINTTM02 (void);
说明	为确认写入完成, 发送开始条件和从属设备地址。
参数	无 -
返回值	无
参考	结果的确认在 IINTIICr 中进行。

[函数名] R_IICr_Init

概要	IICr 初始设定处理
头文件	r_cg_userdefine.h
声明	void R_IICr_Init (void);
说明	对应使用的 IIC 的通道进行设定。IICr 的设定完成后，将写入完成确认用的定时器 02 设定为间隔定时器。设定完成后，将使用的 EEPROM 的初始值设定为 16K 位。
参数	无 -
返回值	无
参考	无

5.7 流程图

本篇应用说明的整体流程图，如图 5.1 所示。

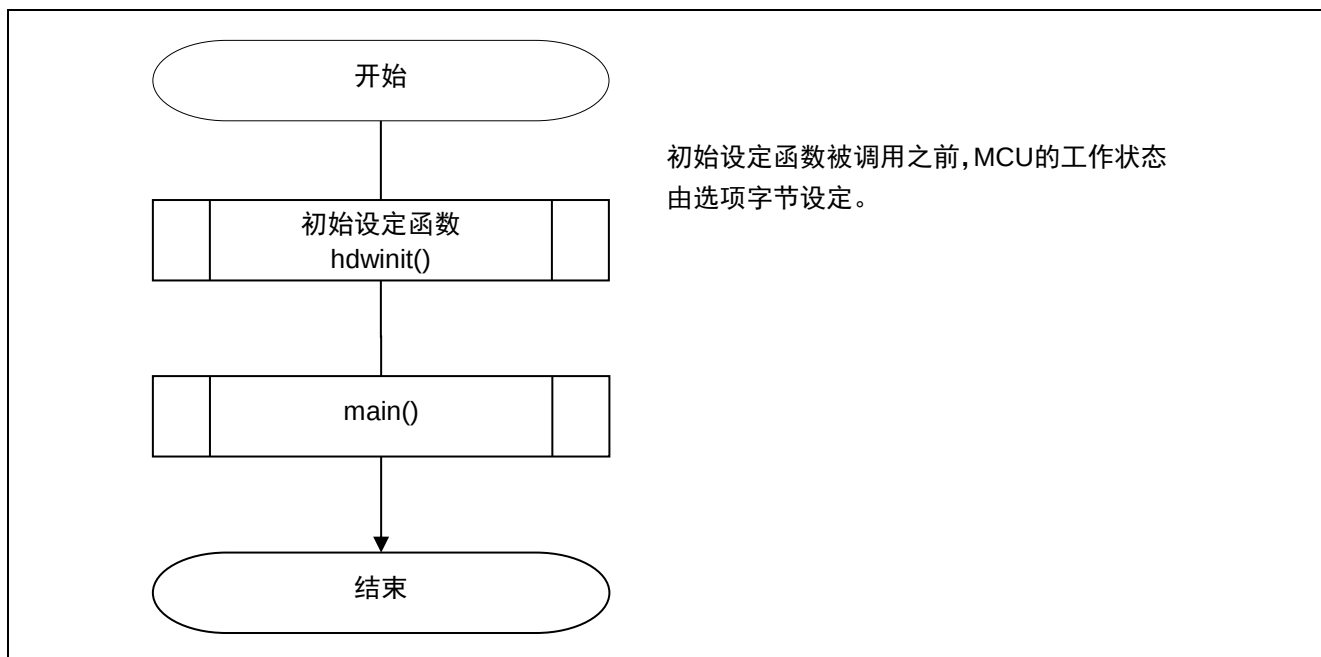


图 5.1 整体流程图

5.7.1 初始设定函数

初始设定函数的流程图，如图 5.2 所示。

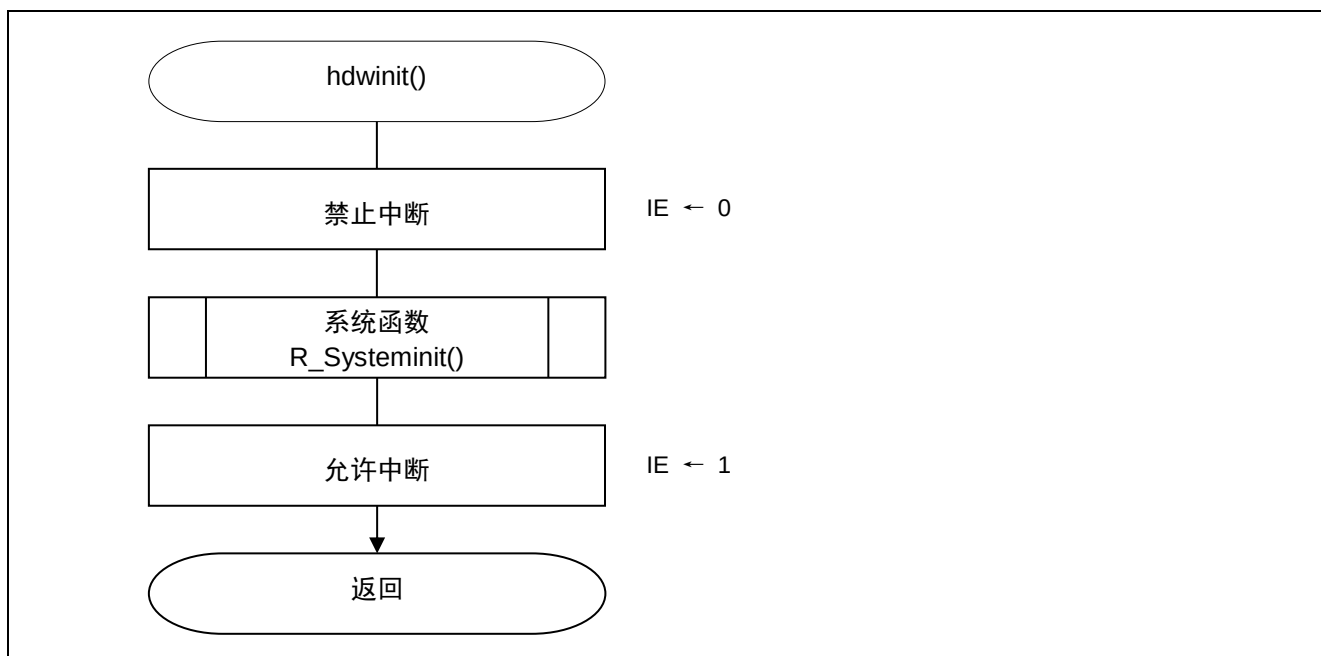


图 5.2 初始设定函数

5.7.2 系统函数

系统函数的流程图，如图 5.3 所示。

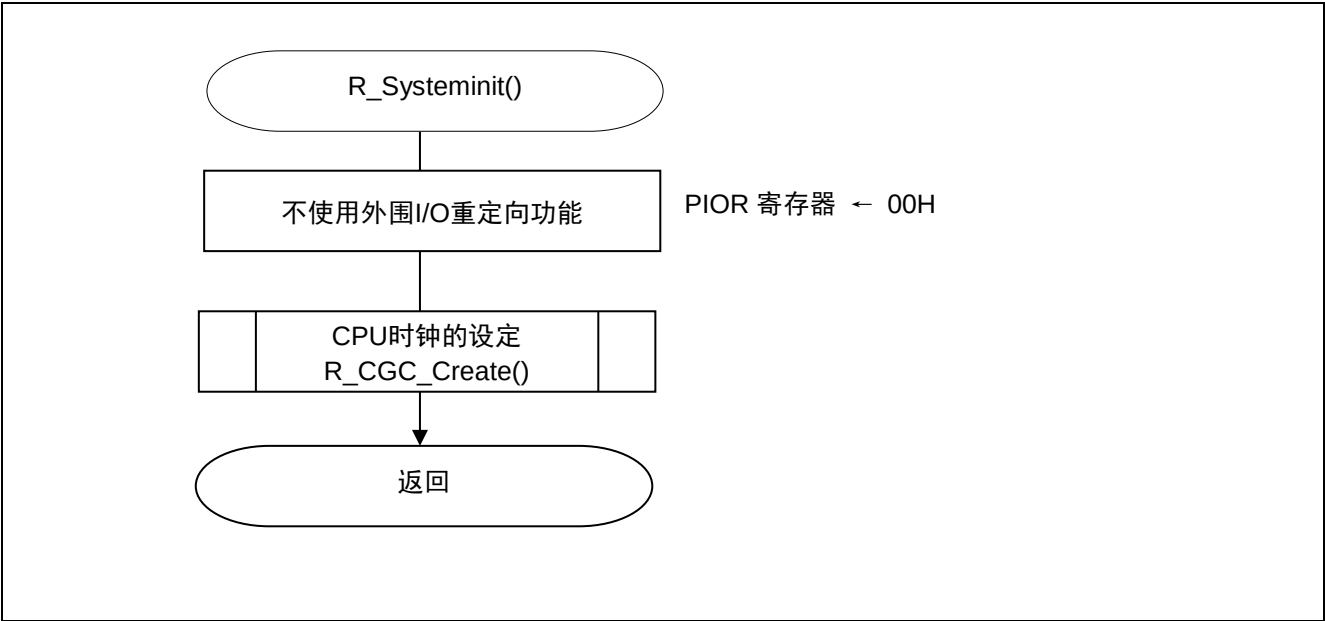


图 5.3 系统函数

备注 根据使用的 IIC 的通道不同，使用的端子会不同，所以没有进行未使用端口的设定。

5.7.3 CPU 时钟的设定

CPU 时钟的设定流程图，如图 5.4 所示。

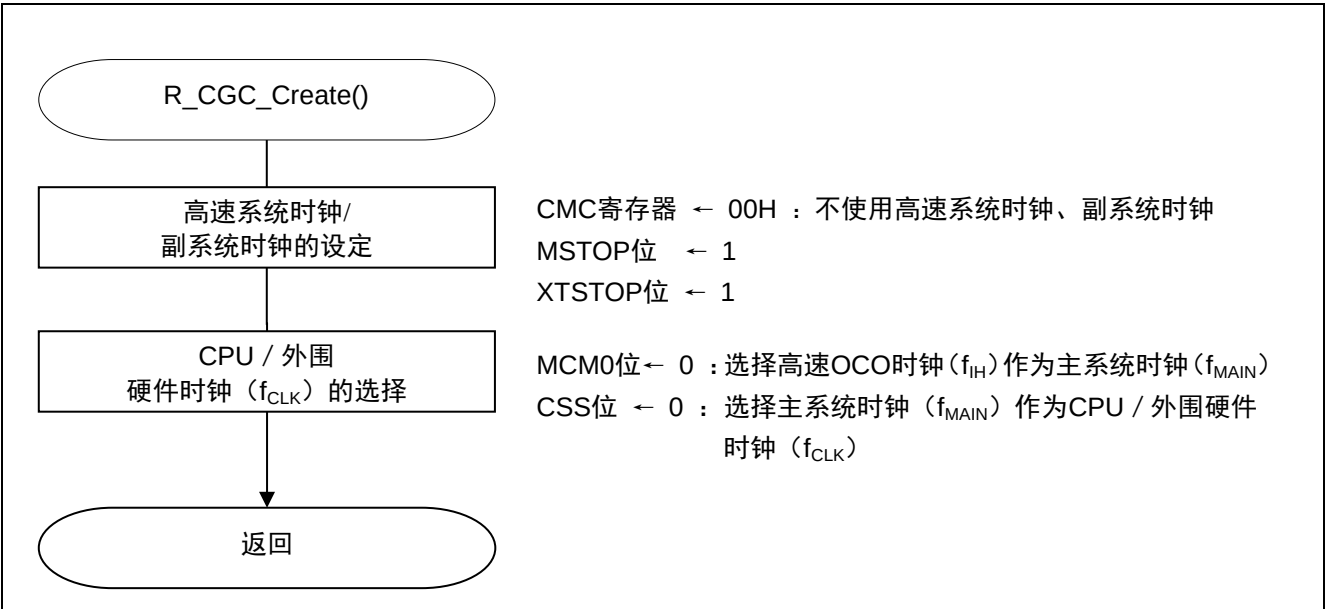


图 5.4 CPU 时钟的设定

5.7.4 串行阵列单元的设置

串行接口的设定流程图，如图 5.5 和 5.6 所示。

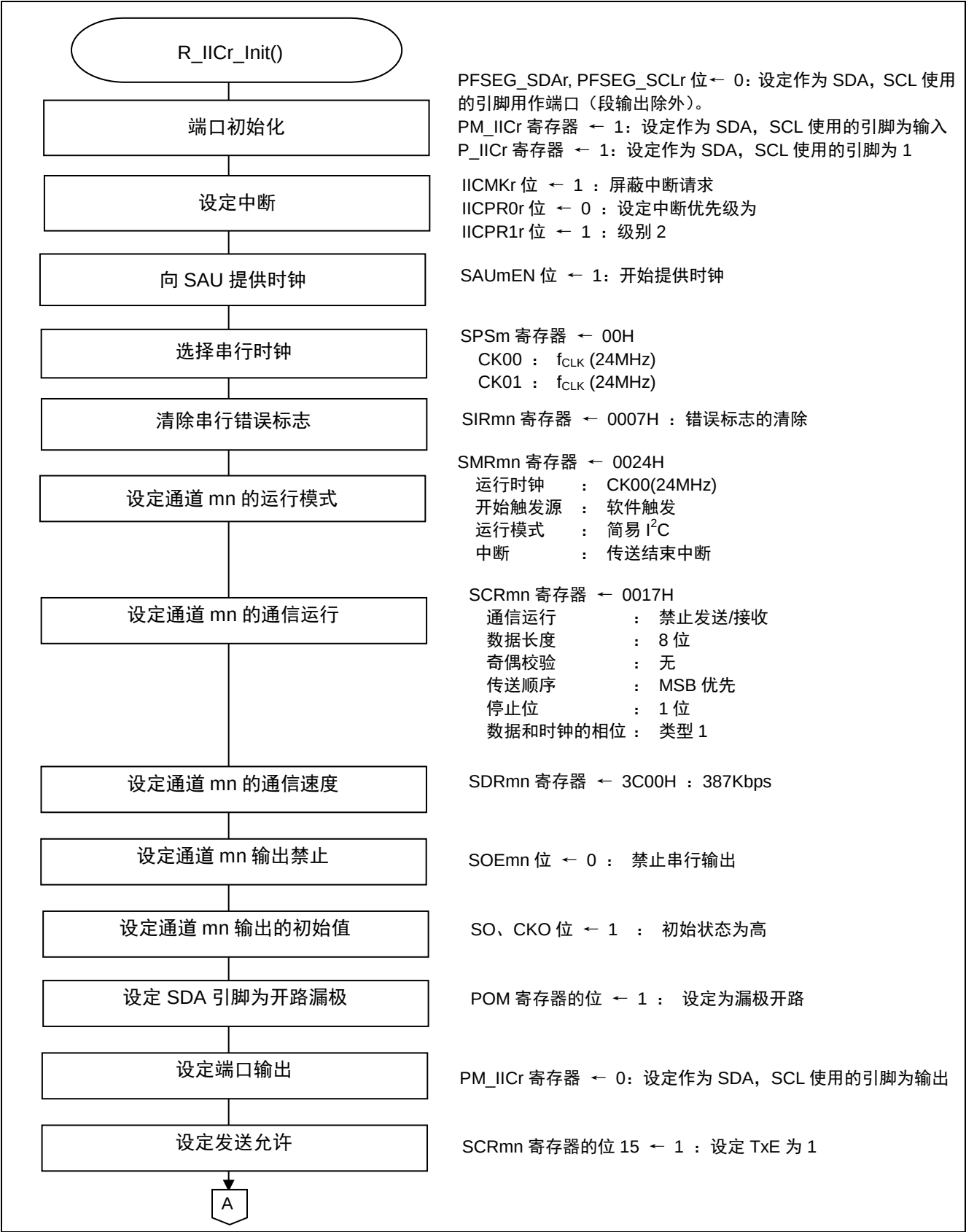


图 5.5 SAU 的设定（1/2）

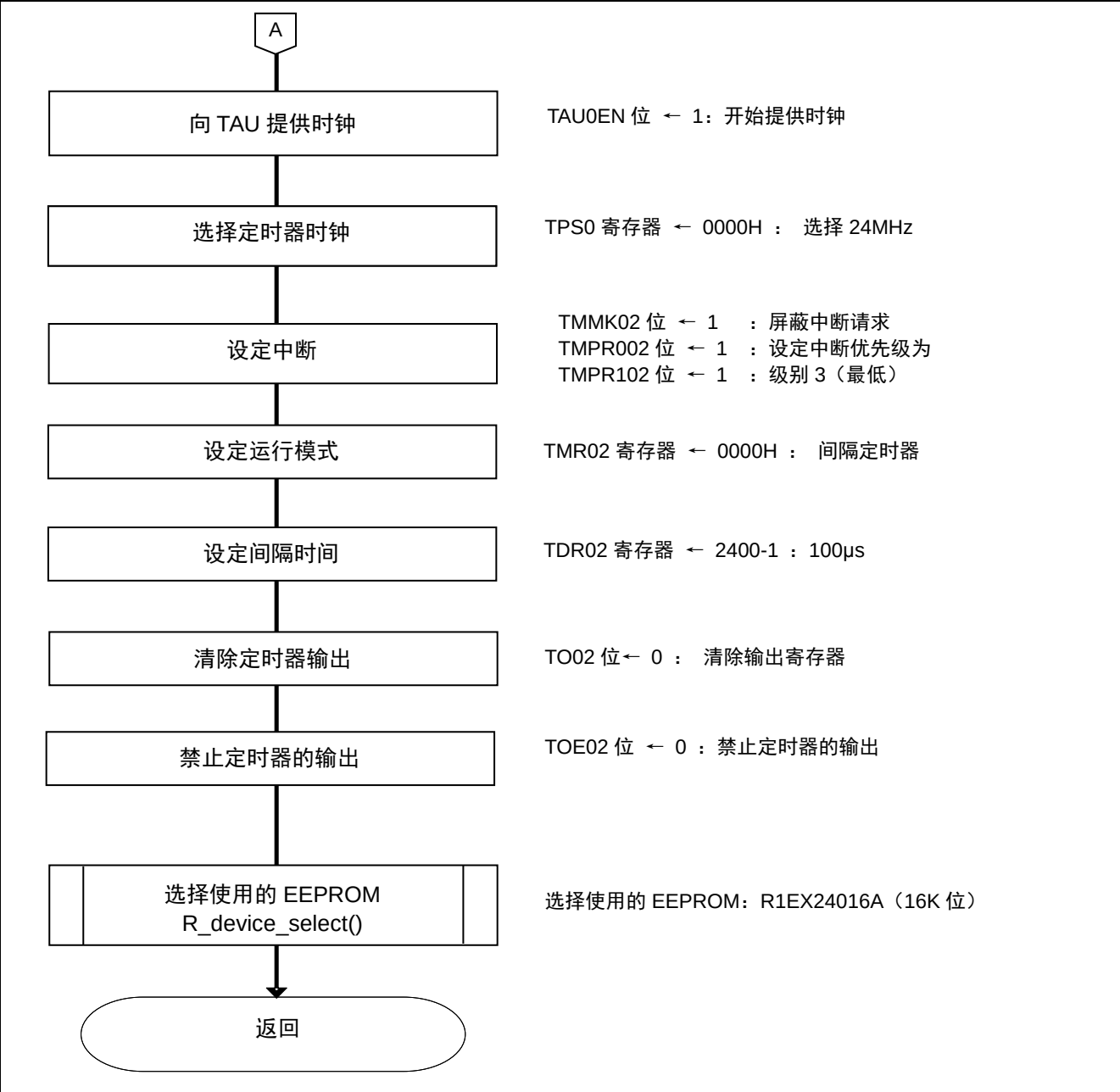


图 5.6 SAU 的设定 (2/2)

开始向串行阵列单元 SAUm 提供时钟

- 外围允许寄存器 0 (PER0)
操作 SAUmEN, 开始向 SAUm 提供时钟。

符号: PER0

7	6	5	4	3	2	1	0
RTWCEN	0	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN
x	0	x	x	1	1	0	x

位 3、2

SAUmEN	串行阵列单元 m 的输入时钟的控制
0	停止提供输入时钟
1	允许提供输入时钟

选择串行阵列单元 SAUm 的时钟

- 串行时钟选择寄存器 m (SPSm)
运行时钟: CK00 = 24MHz, CK01 = 24MHz

符号: SPSm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位 3-0

PRSm 03	PRSm 02	PRSm 01	PRSm 00		运行时钟 (CK00) 的选择				
					$f_{CLK}=$ 2MHz	$f_{CLK}=$ 5MHz	$f_{CLK}=$ 10MHz	$f_{CLK}=$ 20MHz	$f_{CLK}=$ 24MHz
0	0	0	0	f_{CLK}	2MHz	5MHz	10MHz	20MHz	24MHz
0	0	0	1	$f_{CLK}/2$	1MHz	2.5MHz	5MHz	10MHz	12MHz
0	0	1	0	$f_{CLK}/2^2$	500kHz	1.25MHz	2.5MHz	5MHz	6MHz
0	0	1	1	$f_{CLK}/2^3$	250kHz	625kHz	1.25MHz	2.5MHz	3MHz
0	1	0	0	$f_{CLK}/2^4$	125kHz	313kHz	625kHz	1.25MHz	1.5MHz
0	1	0	1	$f_{CLK}/2^5$	62.5kHz	156kHz	313kHz	625kHz	750kHz
0	1	1	0	$f_{CLK}/2^6$	31.3kHz	78.1kHz	156kHz	313kHz	375kHz
0	1	1	1	$f_{CLK}/2^7$	15.6kHz	39.1kHz	78.1kHz	156kHz	187.5kHz
1	0	0	0	$f_{CLK}/2^8$	7.81kHz	19.5kHz	39.1kHz	78.1kHz	93.8kHz
1	0	0	1	$f_{CLK}/2^9$	3.91kHz	9.77kHz	19.5kHz	39.1kHz	46.9kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95kHz	4.88kHz	9.77kHz	19.5kHz	23.4kHz
1	0	1	1	$f_{CLK}/2^{11}$	977Hz	2.44kHz	4.88kHz	9.77kHz	11.7kHz
1	1	0	0	$f_{CLK}/2^{12}$	488Hz	1.22kHz	2.44kHz	4.88kHz	5.86kHz
1	1	0	1	$f_{CLK}/2^{13}$	244Hz	610Hz	1.22kHz	2.44kHz	2.93kHz
1	1	1	0	$f_{CLK}/2^{14}$	122Hz	305Hz	610Hz	1.22kHz	1.46kHz
1	1	1	1	$f_{CLK}/2^{15}$	61Hz	153Hz	305Hz	610Hz	732Hz

注意 详细的寄存器设定方法, 请参照 R7F0C004 用户手册 硬件篇。

错误标志的清除

- 串行标志清除触发寄存器 mn（SIRmn）
清除通道 mn 的错误信息

符号：SIRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	FECT _{mn} ^注	PECT _{mn}	OVCT _{mn}
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

位 1

PECTmn	通道 mn 的奇偶校验错误标志（无 ACK 应答）的清除触发
0	不清除
1	将 SSRmn 寄存器的 PEFmn 位清 0

注 仅限奇数通道

注意 详细的寄存器设定方法，请参照 R7F0C004 用户手册 硬件篇。

设定通道 mn 的运行模式

- 串行模式寄存器 mn (SMRmn)
 - 设定运行时钟: CK00
 - 设定开始触发: 仅软件触发
 - 设定运行模式: 简易 I²C
 - 设定中断源: 传送结束中断

符号: SMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKS mn	CCS mn	0	0	0	0	0	STS mn ^注	0	SIS mn0 ^注	1	0	0	MD mn2	MD mn1	MD mn0
0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

位 15

CKSmn	通道 mn 运行时钟 (f _{MCK}) 的选择
0	SPS0 寄存器设定的运行时钟 CK00
1	SPS0 寄存器设定的运行时钟 CK01

位 14

CCSmn	通道 mn 传送时钟 (f _{TCLK}) 的选择
0	CKSmn 位指定的运行时钟 f _{MCK} 的分频时钟
1	来自 SCK00 引脚的输入时钟 f _{SCK} (CSI 模式的从属传送)

位 8

STS ^注	开始触发源的选择
0	只有软件触发有效 (在 CSI、UART 发送、简易 I ² C 时选择)
1	RxD0 引脚的有效边沿 (在 UART 接收时选择)

位 2-1

MDmn2	MDmn1	通道 mn 运行模式的设定
0	0	CSI 模式
0	1	UART 模式
1	0	简易 I ² C 模式
1	1	禁止设定

位 0

MDmn0	通道 mn 中断源的选择
0	传送结束中断
1	缓冲器空中断 (在传送数据从 SDRmn 寄存器传送到移位寄存器时发生)

注 仅限奇数通道

注意 详细的寄存器设定方法, 请参照 R7F0C004 用户手册 硬件篇。

通信格式的设定

- 串行通信运行设定寄存器 mn (SCRmn)

运行模式: 只发送

奇偶校验位的设定: 无奇偶校验

数据传送顺序: MSB 优先

停止位长度: 1 位

数据长度: 8 位

符号: SCRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	1	DLS mn1	DLS mn0
1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1

位 15-14

TXEmn	RXEmn	通道 mn 运行模式的设定
0	0	禁止通信
0	1	只接收
1	0	只发送
1	1	发送和接收

位 13-12

DAPmn	CKPmn	CSI 模式中的数据和时钟的相位选择
0	0	类型 1
0	1	类型 2
1	0	类型 3
1	1	类型 4

在 UART 模式和简易 I2C 模式中, 必须设定 DAP01, CKP01 = 0, 0。

位 10

EOCmn	错误中断信号 (INTSREx (x = 0-3)) 的屏蔽控制
0	屏蔽错误中断 INTSREx (不屏蔽 INTSRx)
1	允许发生错误中断 INTSREx (错误发生时 INTSRx 被屏蔽)

在 CSI 模式和简易 I2C 模式中或者在 UART 发送时, 必须将 EOCmn 位置 “0”。

注意 详细的寄存器设定方法, 请参照 R7F0C004 用户手册 硬件篇。

位 9-8

PTCmn1	PTCmn0	UART 模式中的奇偶校验位的设定	
		发送动作	接收动作
0	0	不输出奇偶校验位	接收时没有奇偶校验
0	1	输出零校验	不判断奇偶校验
1	0	输出偶校验	判断偶校验
1	1	输出奇校验	判断奇校验

位 7

DIRmn	CSI 和 UART 模式中的数据传送顺序的选择
0	MSB 优先的输入/输出
1	LSB 优先的输入/输出

位 5-4

SLCmn1	SLCmn0	UART 模式中的停止位的设定
0	0	无停止位
0	1	停止位长度 = 1 位
1	0	停止位长度=2 位 (只限于 mn=00、02、10、12)
1	1	禁止设定

在 UART 接收时或者在简易 I2C 模式中，必须设定为 1 位停止位 (SLCmn1、SLCmn0=0、1)。

位 1-0

DLSmn1	DLSmn0	CSI 和 UART 模式中的数据长度的设定
0	1	9 位数据长度 (保存到 SDRmn 寄存器的 bit0-8) (只在 UART 模式中可选择)
1	0	7 位数据长度 (保存到 SDRmn 寄存器的 bit0-6)
1	1	8 位数据长度 (保存到 SDRmn 寄存器的 bit0-7)
其他		禁止设定

注意 详细的寄存器设定方法，请参照 R7F0C004 用户手册 硬件篇。

运行时钟（ f_{MCK} ）的分频设定

- 串行数据寄存器 mn（SDRmn）
传送时钟： $f_{MCK}/62$ （387Kbps）

符号：SDRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
运行时钟（ f_{MCK} ）的分频							发送和接收缓冲器								
0	0	1	1	1	1	0	x	x	x	x	x	x	x	x	x

位 15-9

b15	b14	b13	b12	b11	b10	b9	通过运行时钟（ f_{MCK} ）的分频设定传送时钟
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
0	0	1	1	1	1	0	$f_{MCK}/62$
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

注意 详细的寄存器设定方法，请参照 R7F0C004 用户手册 硬件篇。

对象通道的串行数据输出允许的设定

- 串行输出允许寄存器 m (SOEm)
停止输出

符号: SOEm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	SOE m2	0	SOE m0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SOEmn	通道 mn 串行输出的允许/停止
0	停止串行通信的输出
1	允许串行通信的输出

SDA, SCL 初始输出电平的设定

- 串行输出寄存器 m (SOM)
输出电平: 1

符号: SOM

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
串行时钟输出								串行数据输出							
0	0	0	0	1	1	1	CKO m0 ^注	0	0	0	0	1	SO m2	1	SO m0
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

位 8^注

CKOm	通道 mn 的串行时钟输出
0	串行时钟的输出值为“0”
1	串行时钟的输出值为“1”

位 2、0

SOMn	通道 mn 的串行数据输出
0	串行数据的输出值为“0”
1	串行数据的输出值为“1”

注 只限于 SO0 寄存器。

注意 详细的寄存器设定方法, 请参照 R7F0C004 用户手册 硬件篇。

IICr 引脚的设定

- LCD 端口功能寄存器 (PFSEGr)
- 端口寄存器 (Px)
- 端口模式寄存器 (PMx)
- 端口输出模式寄存器 (POMx)

设定 SDA 引脚为开路漏极输出。

SCL、SDA 在输出模式下使用的引脚设定如下所示。

简易 I ² C 通道	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21	IIC30	IIC31
SCL 引脚	P16	—	P05	—	—	—	—	—
SDA 引脚	P17	—	P06	—	—	—	—	—

符号: PFSEGr

PFSEGrk	Pxn 引脚的端口 (段输出除外) 或者段输出的指定
0	用作端口 (段输出除外)
1	用作段输出

符号: Pxn

Pxn	输出数据的控制
0	输出 0
1	输出 1

符号: PMxn

PMxn	Pxn 输入/输出模式的选择
0	输出模式 (输出缓冲器 ON)
1	输入模式 (输出缓冲器 OFF)

符号: POMxn

POMxn	Pxn 输出模式的选择
0	CMOS 输出模式
1	N 沟道漏极开路输出模式

注 仅设定作为 SDA 使用的引脚

定时器 02 (间隔定时器) 的设定

- 定时器的详细设定, 请参考以下应用说明。

请参考 R7F0C009 定时器阵列单元 (间隔定时器) (R01AN2439CC) 应用说明。

5.7.5 主函数

图 5.7～图 5.9 所示为主函数的流程图。作为 API 的使用示例，主函数对 16K 位 EEPROM 进行读写测试。

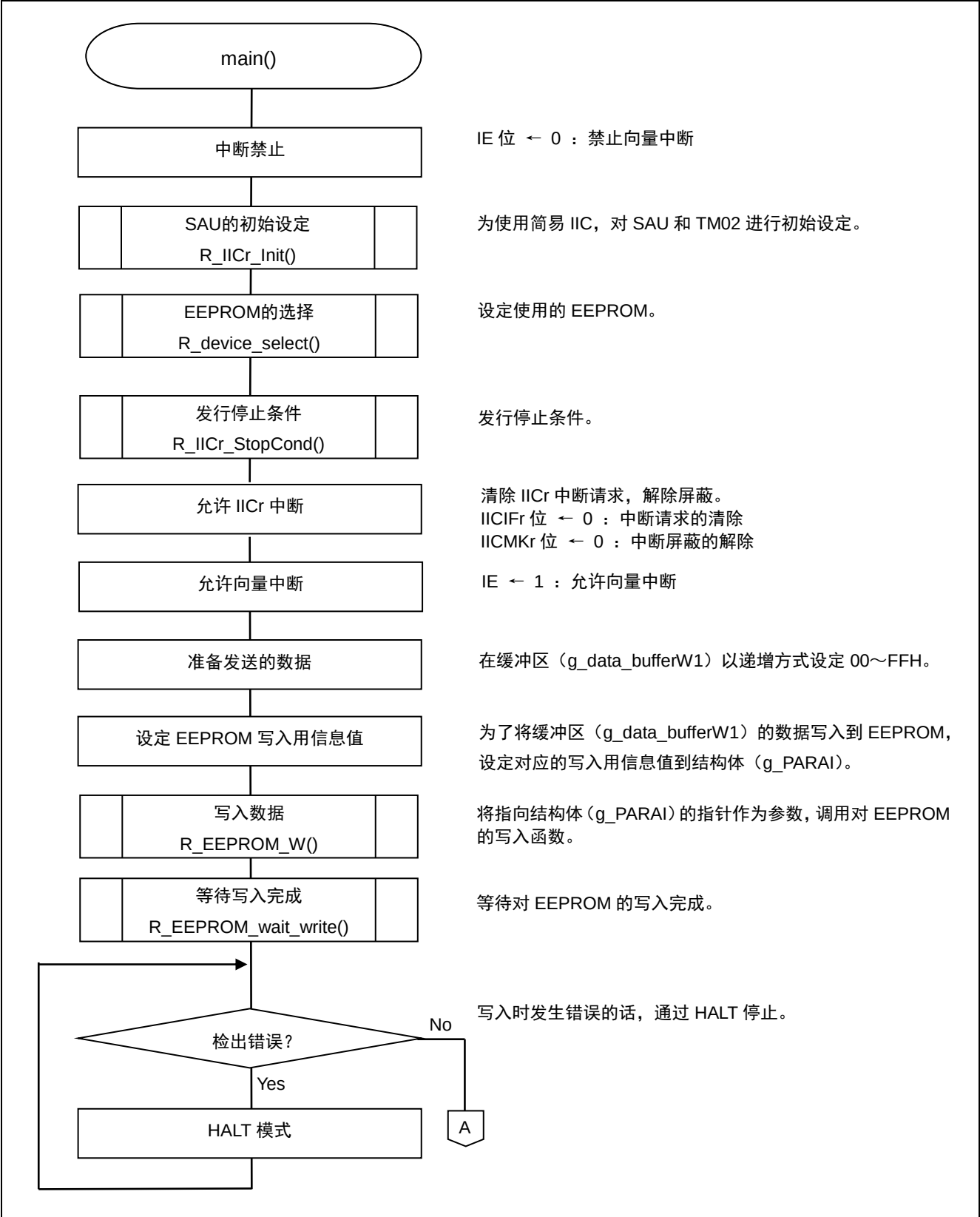


图 5.7 主函数 (1/3)

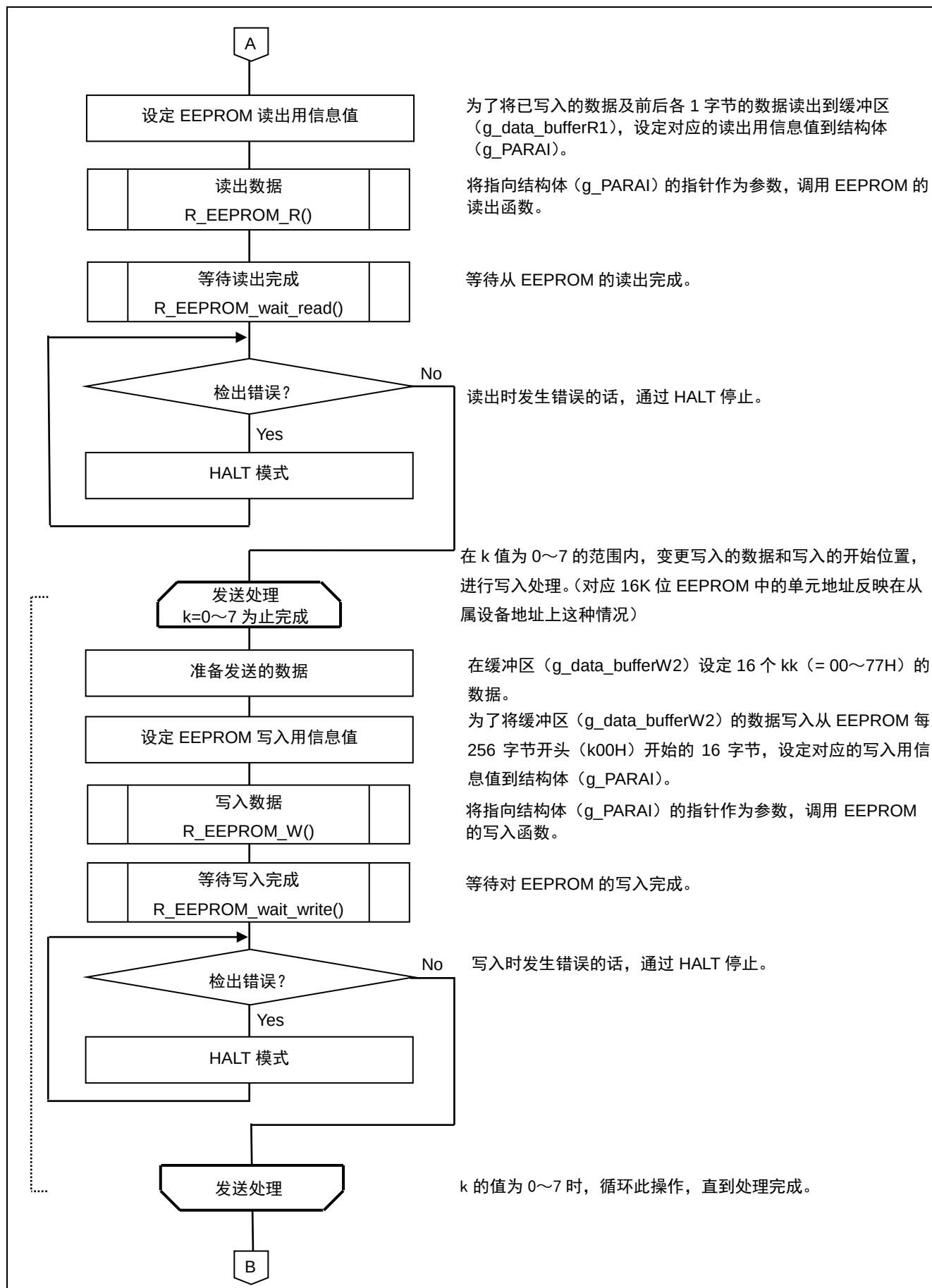


图 5.8 主函数 (2/3)

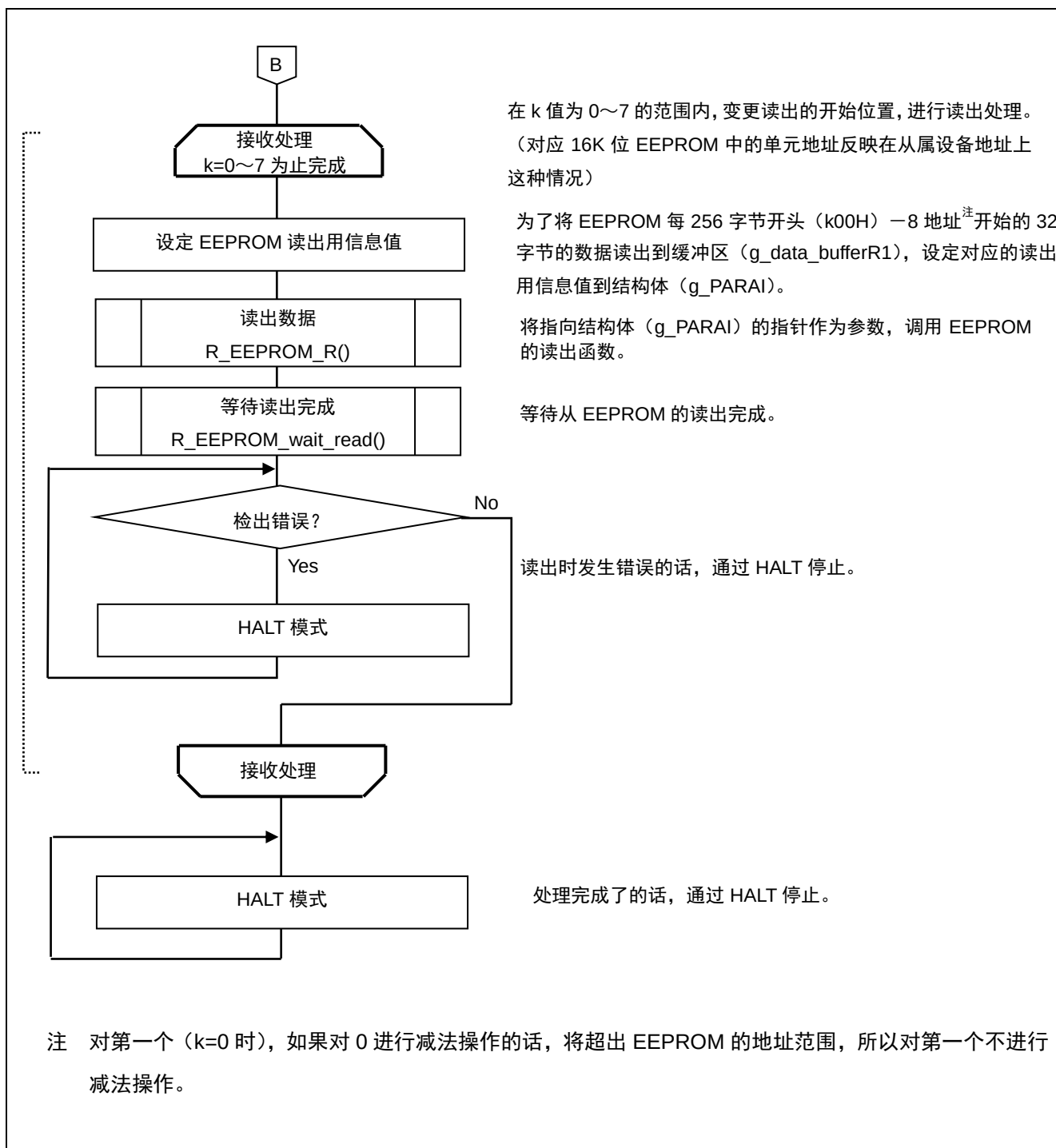


图 5.9 主函数 (3/3)

5.7.6 EEPROM 的选择

图 5.10 所示为 EEPROM 的选择的流程图。

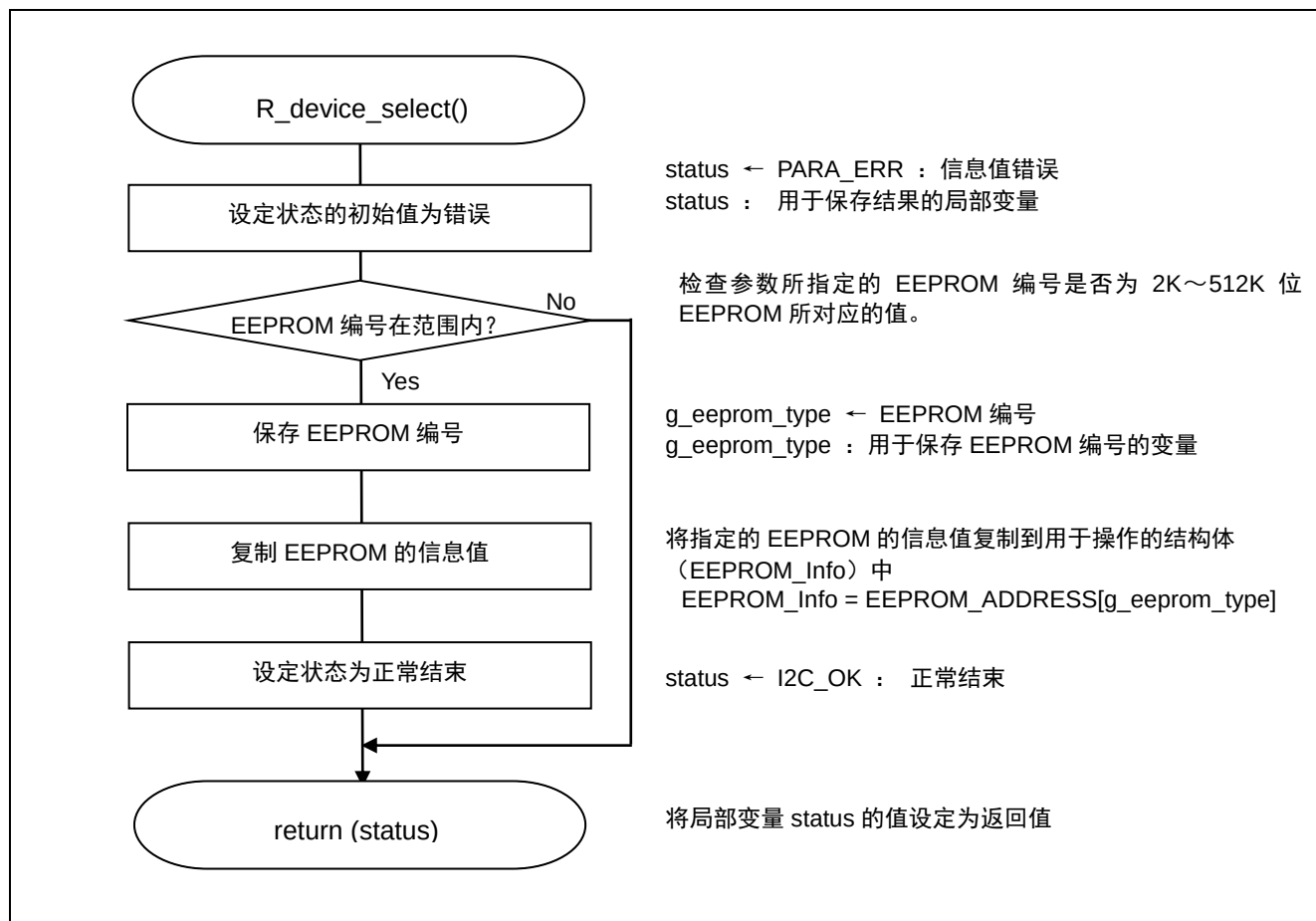


图 5.10 EEPROM 的选择

5.7.7 停止条件发行函数

图 5.11 所示为停止条件发行函数的流程图。

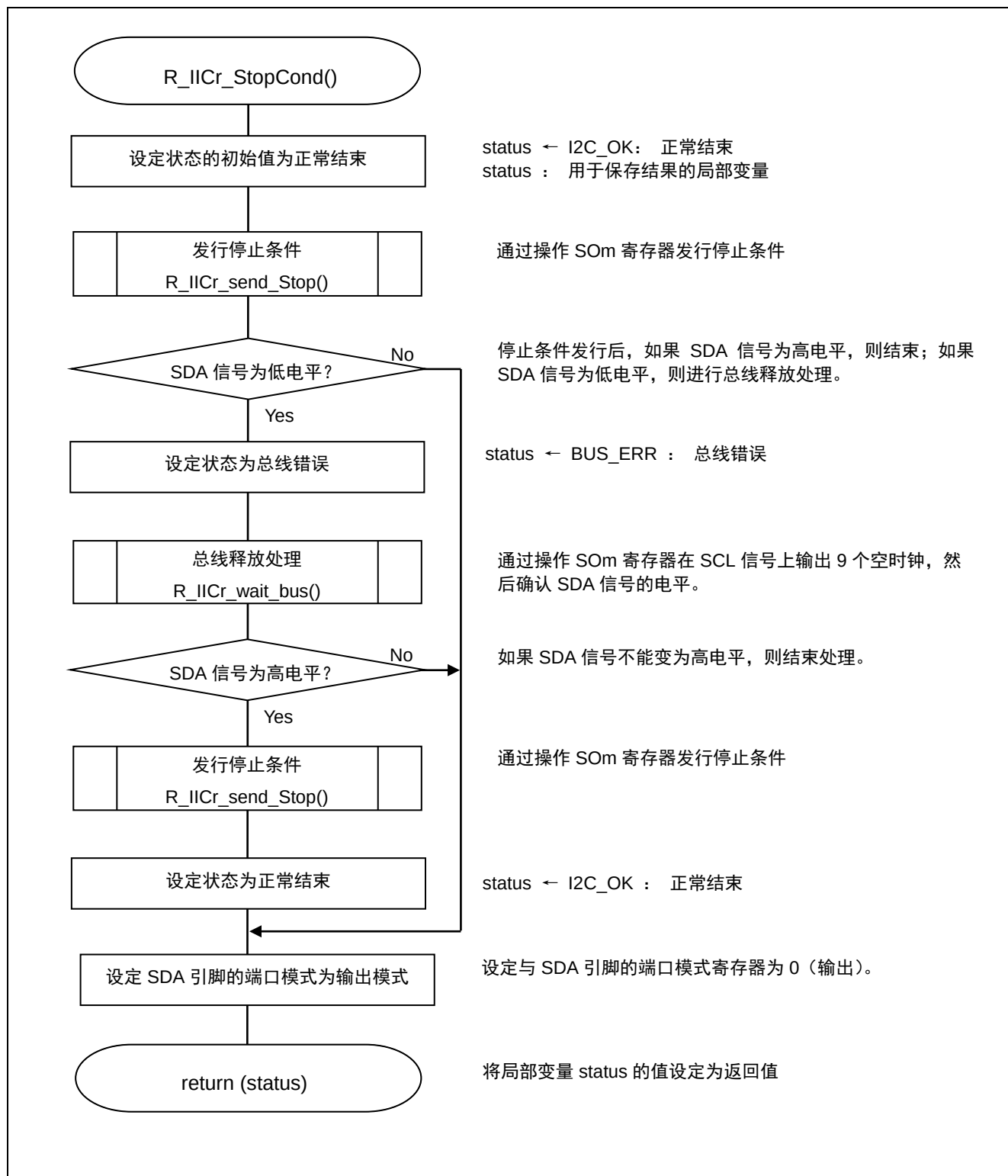


图 5.11 停止条件发行函数

5.7.8 停止条件生成函数

图 5.12 所示为停止条件生成函数的流程图。

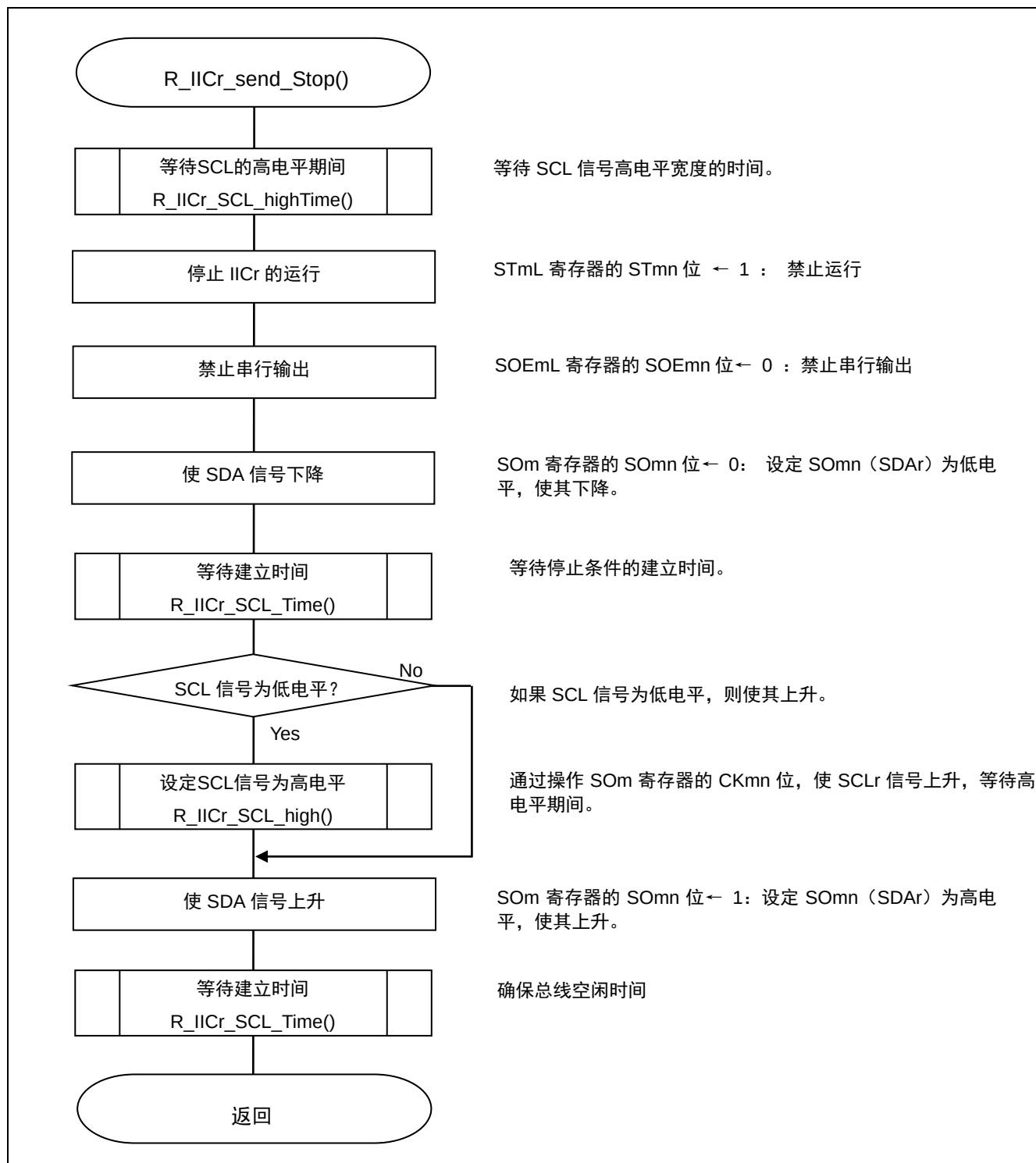


图 5.12 停止条件生成函数

串行通道的停止

- 串行通道停止寄存器 m (STm/STmL)
串行通道的停止

符号: STmL

7	6	5	4	3	2	1	0
0	0	0	0	STm3	STm2	STm1	STm0
0	0	0	0	1	1	1	1

位 3-0

STmn	通道 mn 运行停止的触发
0	没有触发
1	将 SEmn 位清 0, 停止通信运行

串行输出的禁止

- 串行输出允许寄存器 m (SOEm/SOEmL)
串行通道的输出禁止

符号: SOEmL

7	6	5	4	3	2	1	0
0	0	0	0	0	SOEm2	0	SOEm0
0	0	0	0	0	0	0	0

位 2、0

SOEmn	通道 mn 串行输出的允许/停止
0	停止串行通信的输出
1	允许串行通信的输出

串行输出的操作

- 串行输出寄存器 m (SOm)
串行输出 (SCL, SDA) 的操作

符号: SOm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	CKO m0 ^注	0	0	0	0	1	SO m2	1	SO m0
0	0	0	0	1	1	1	0/1	0	0	0	0	1	0/1	1	0/1

位 8^注

CKOmn	通道 mn 的串行时钟输出
0	串行时钟的输出值为 “0”
1	串行时钟的输出值为 “1”

位 2、0

SOMn	通道 mn 串行数据输出
0	串行数据的输出值为 “0”
1	串行数据的输出值为 “1”

注 只限于 SO0 寄存器。

注意 详细的寄存器设定方法, 请参照 R7F0C004 用户手册 硬件篇。

5.7.9 总线释放处理函数

图 5.13 所示为总线释放处理函数的流程图。

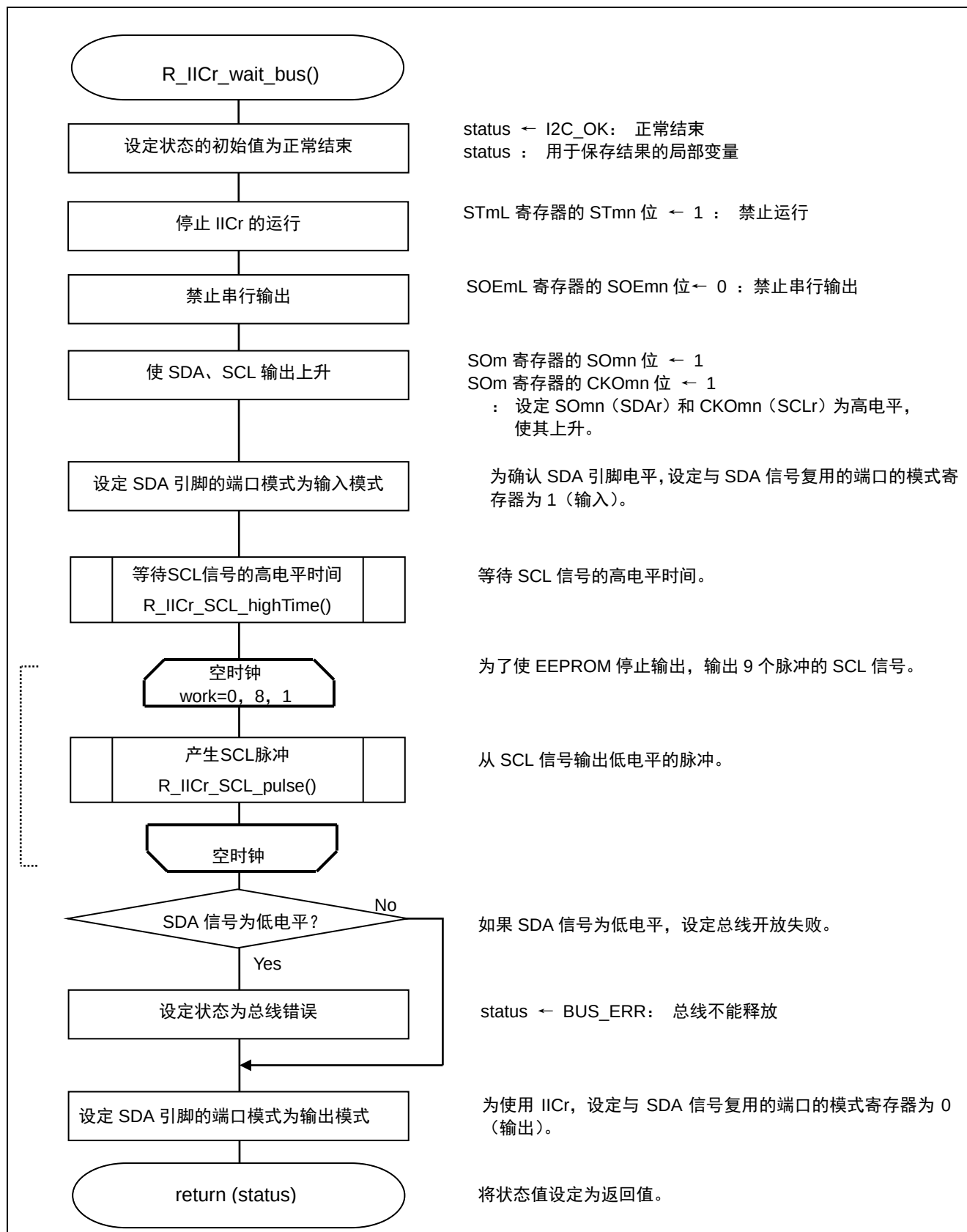


图 5.13 总线释放处理函数

5.7.10 EEPROM 写入处理

图 5.14 所示为 EEPROM 写入处理的状态转移图，图 5.15 所示为流程图。

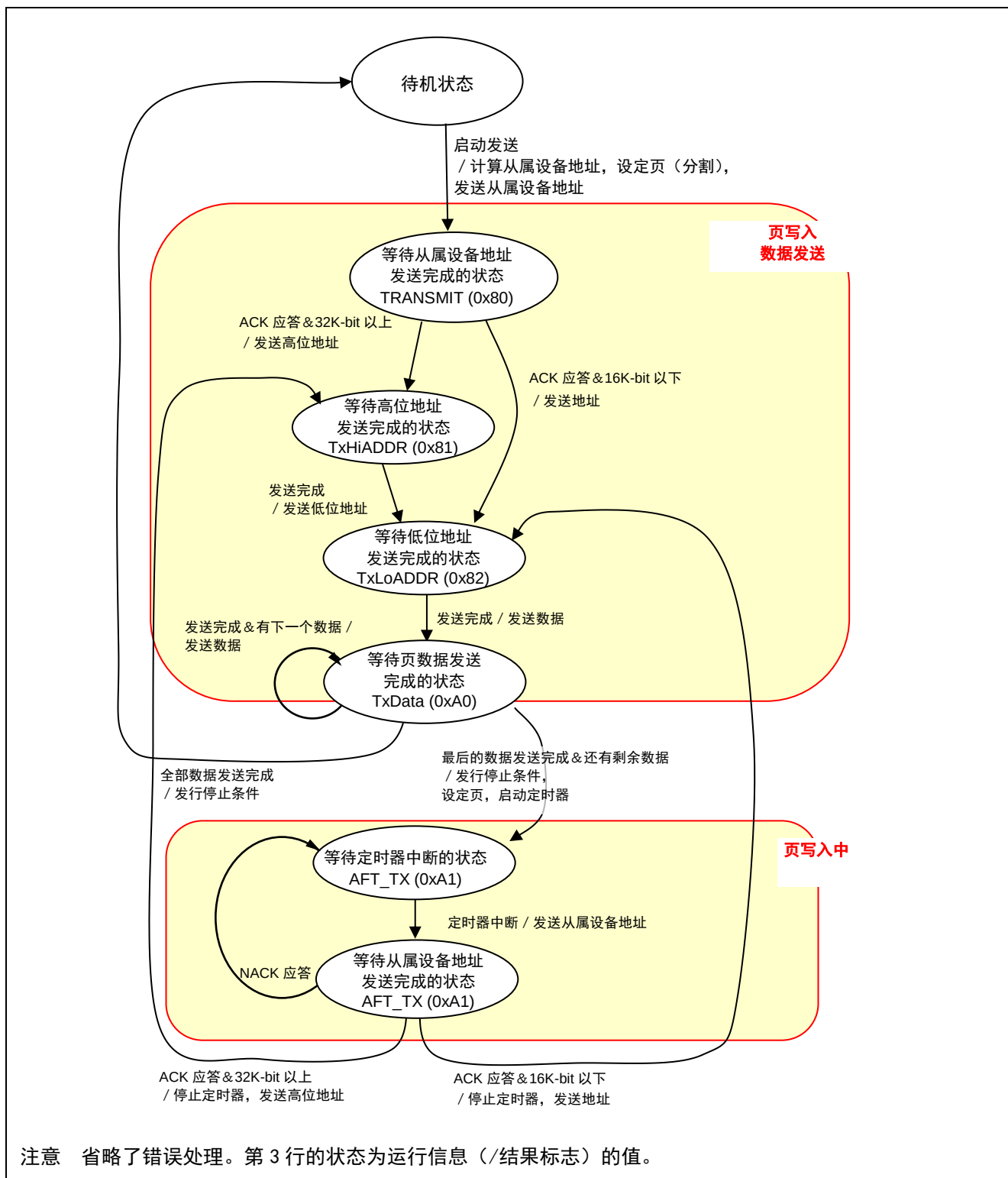


图 5.14 EEPROM 写入处理的状态转移图

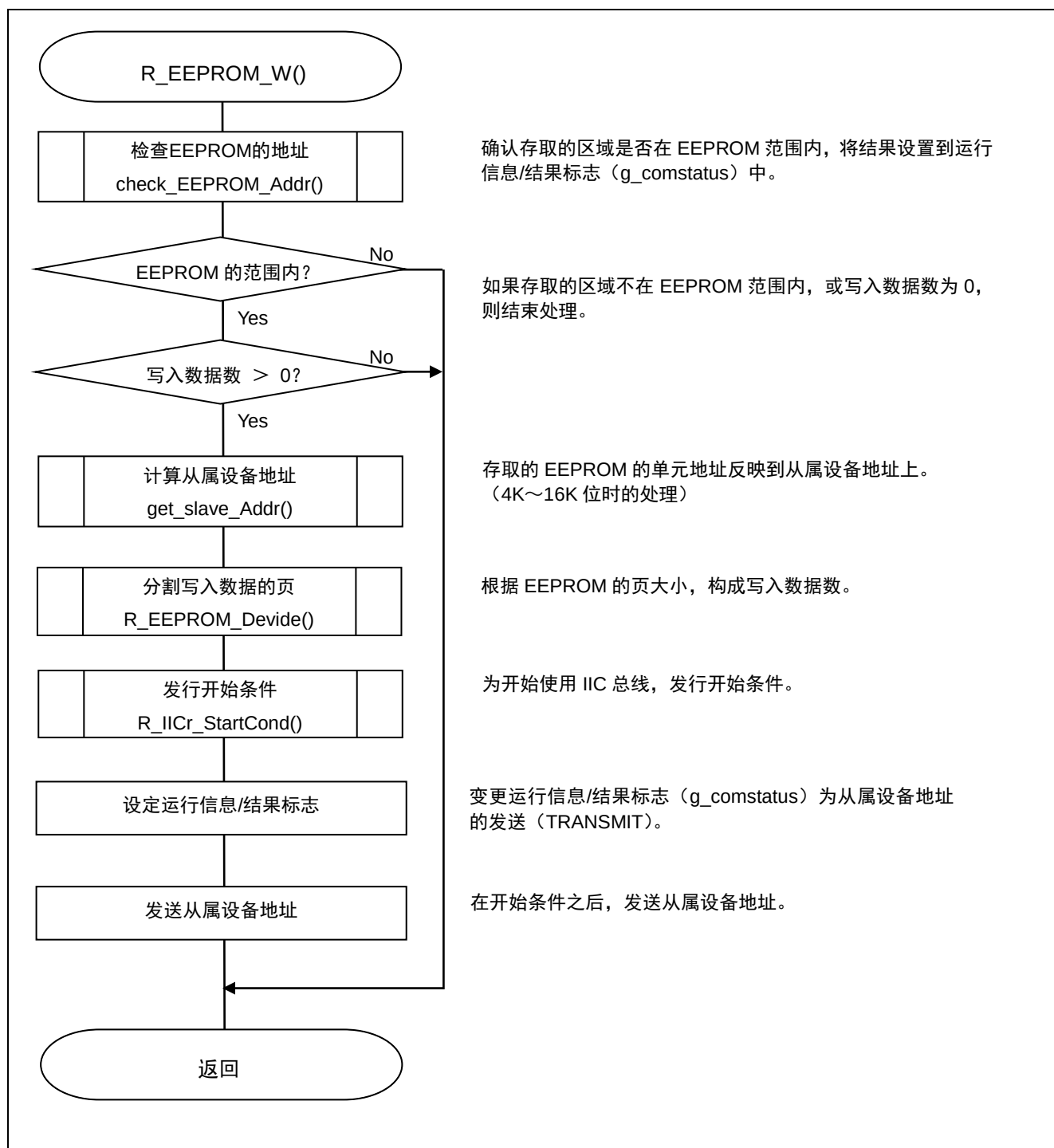


图 5.15 EEPROM 写入处理

5.7.11 EEPROM 地址检查处理

图 5.16 所示为 EEPROM 地址检查处理的流程图。

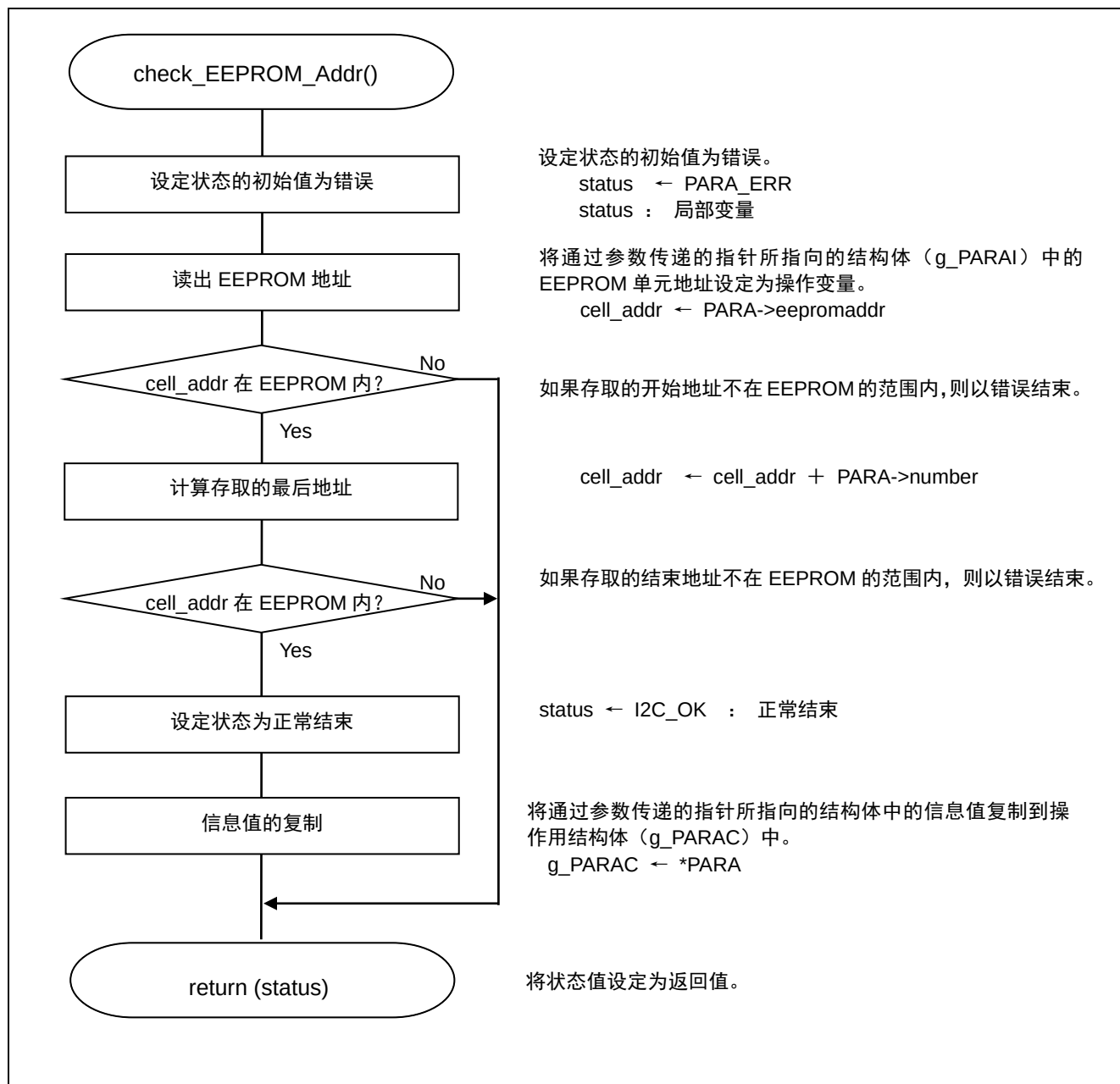


图 5.16 EEPROM 地址检查处理

5.7.12 计算从属设备地址的处理

图 5.17 所示为计算从属设备地址的处理流程图。

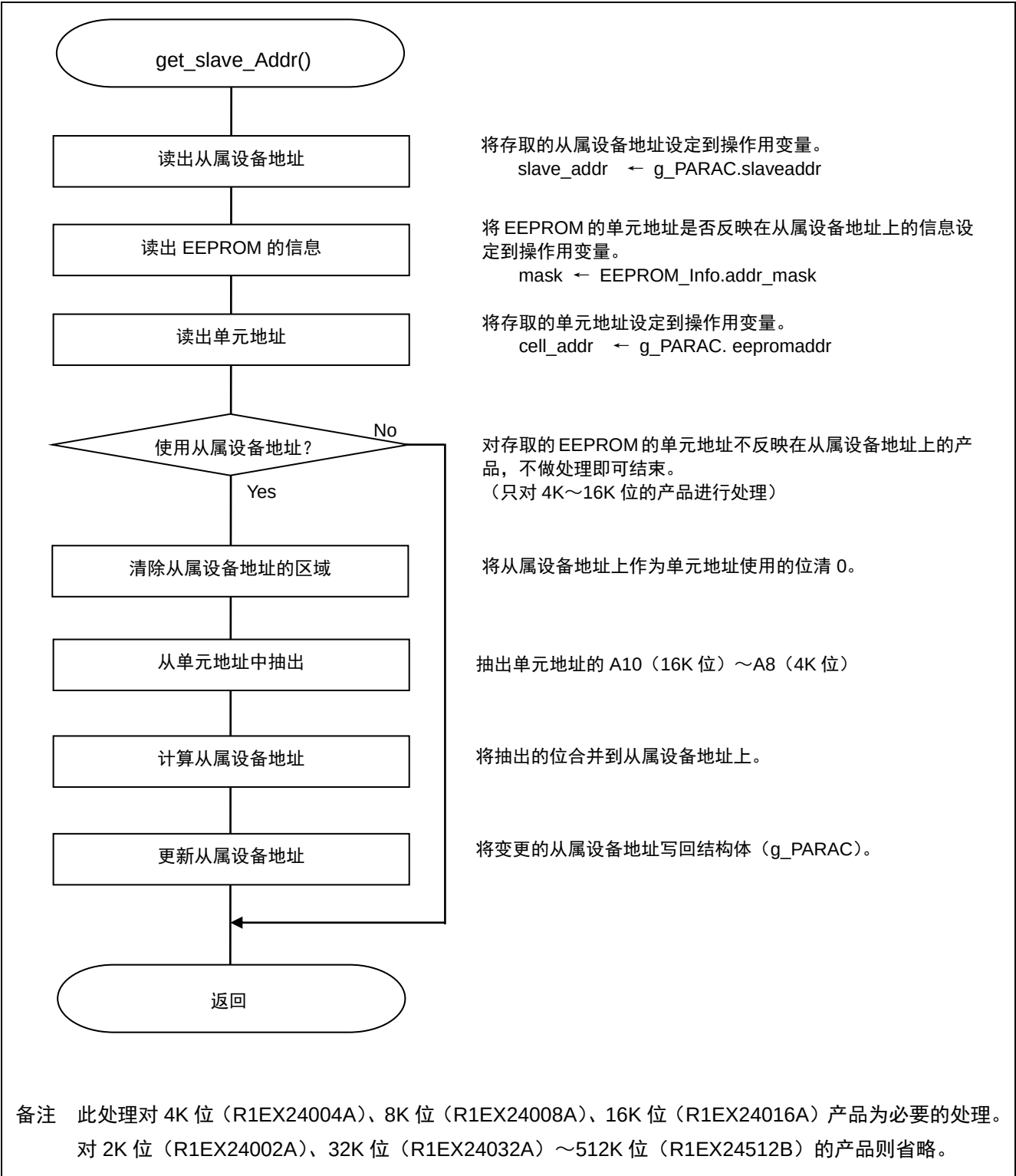


图 5.17 计算从属设备地址的处理

5.7.13 写入数据的页分割处理

图 5.18 所示为写入数据的页分割处理的流程图。

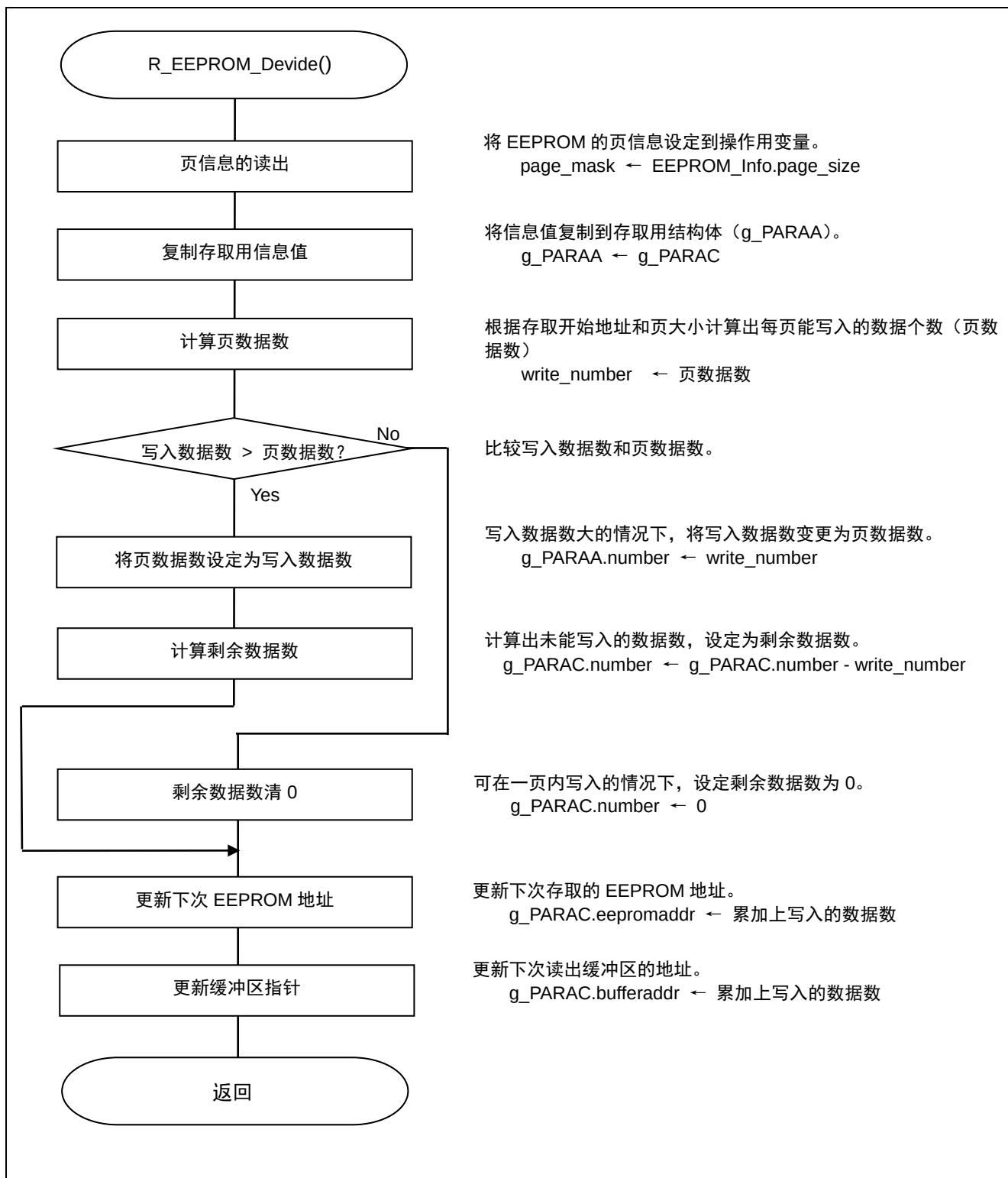


图 5.18 写入数据的页分割处理

5.7.14 开始条件发行处理

图 5.19 所示为开始条件发行处理的流程图。



图 5.19 开始条件发行处理

串行通道运行开始

- 串行通道开始寄存器 m（SSm/SSmL）
开始串行通道的运行

符号：SSmL

7	6	5	4	3	2	1	0
0	0	0	0	SSm3	SSm2	SSm1	SSm0
0	0	0	0	1	1	1	1

位 3-0

SSmn	通道 mn 运行开始的触发
0	没有触发
1	将 SEmn 位置“1”，转移到通信待机状态

注意 详细的寄存器设定方法，请参照 R7F0C004 用户手册 硬件篇。

5.7.15 等待 EEPROM 写入完成的处理

图 5.10 所示为等待 EEPROM 写入完成的处理流程图。

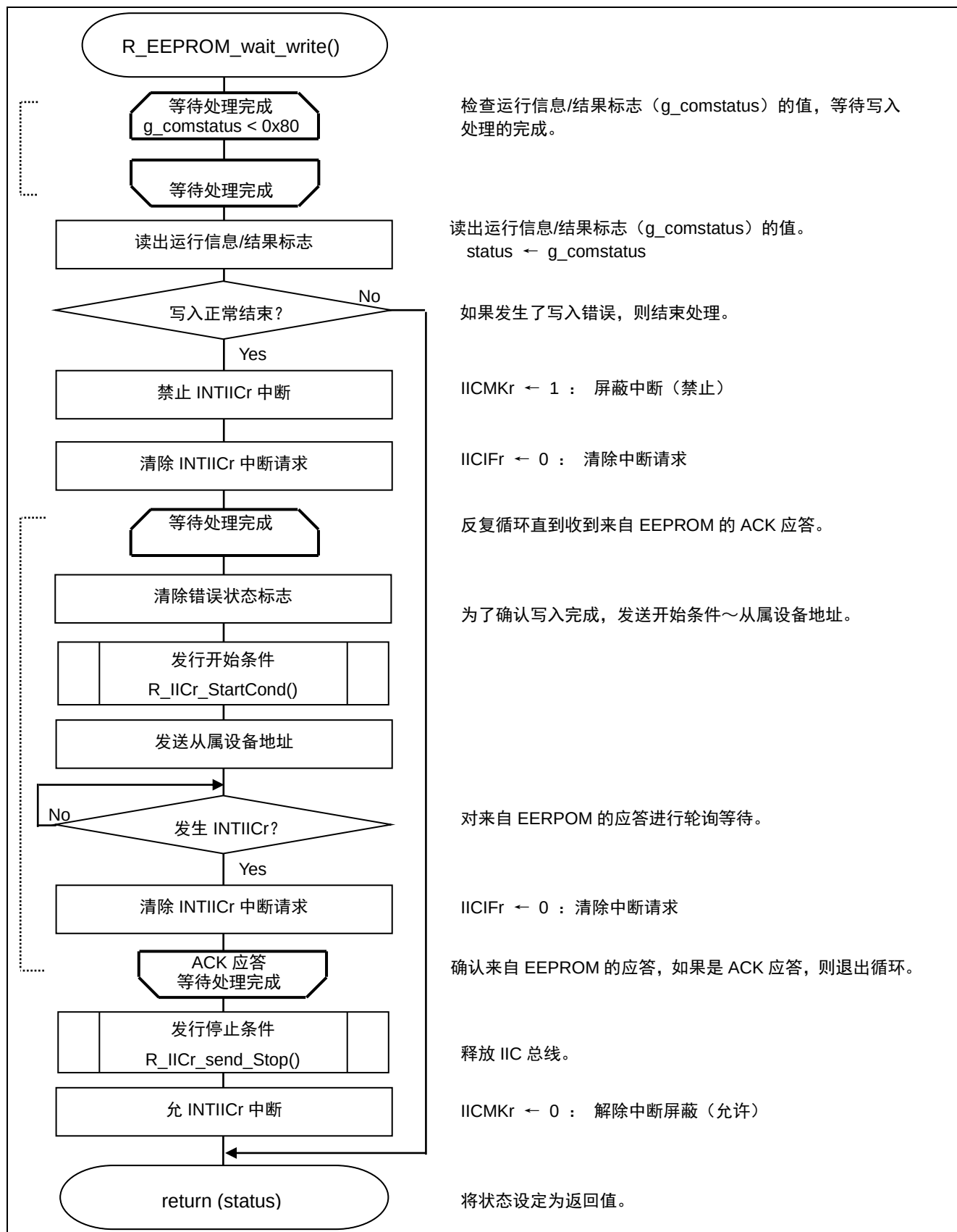


图 5.20 等待 EEPROM 写入完成的处理

5.7.16 EEPROM 读出处理

图 5.21 所示为 EEPROM 读出处理的状态转移图，图 5.22 所示为流程图。

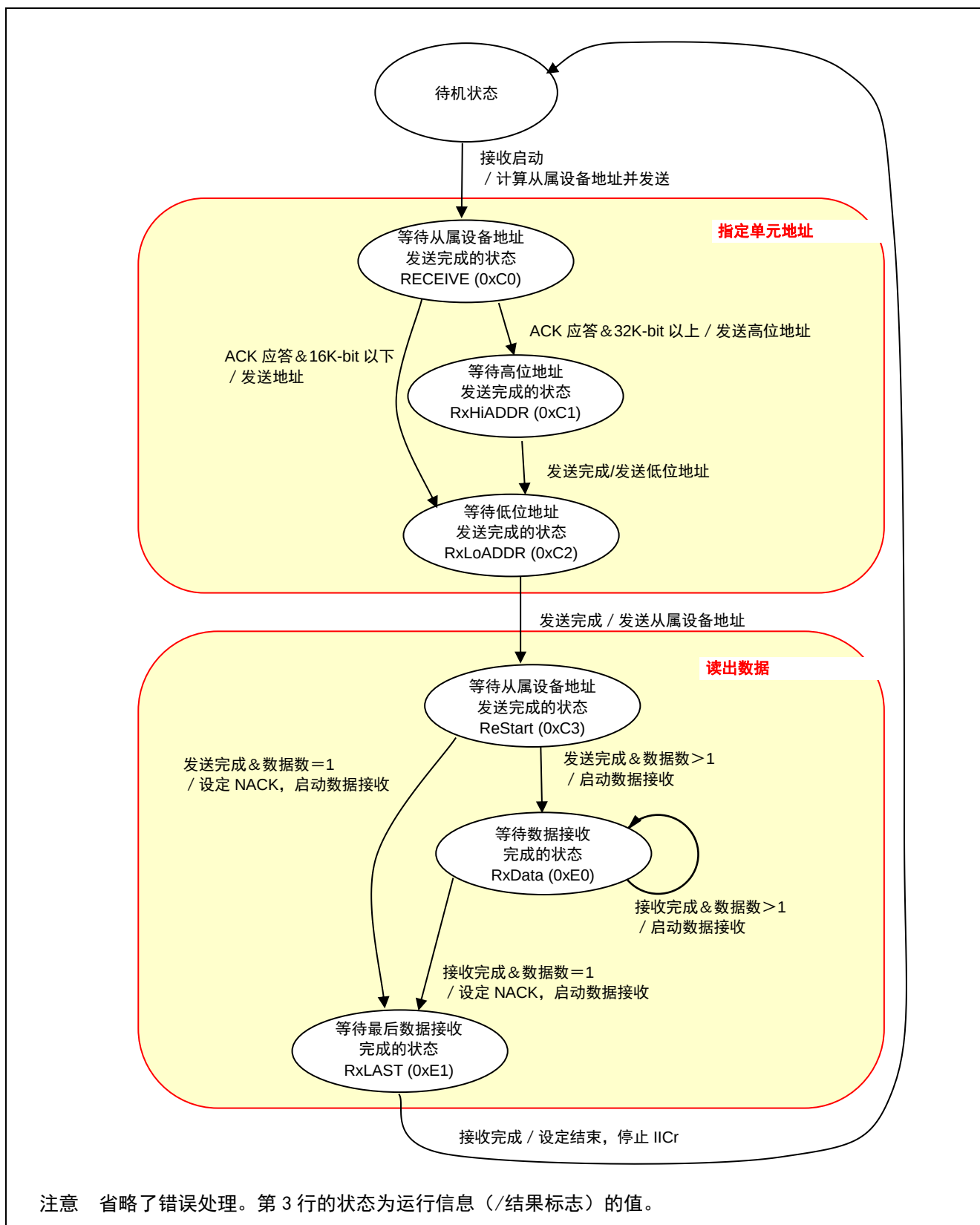


图 5.21 EEPROM 读出处理状态转移图

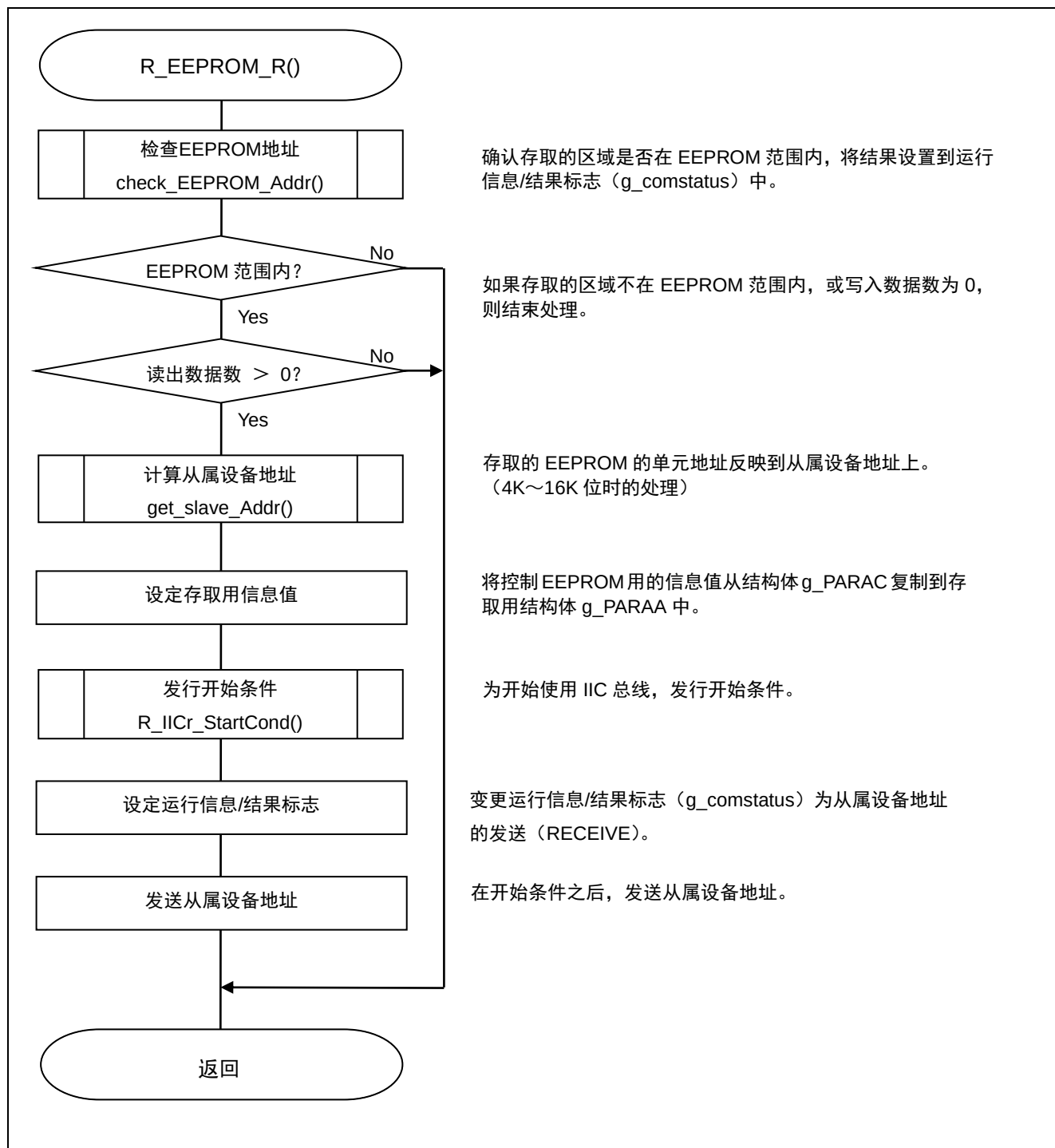


图 5.22 EEPROM 读出处理

5.7.17 等待 EEPROM 读出完成的处理

图 5.23 所示为等待 EEPROM 读出完成的处理流程图。

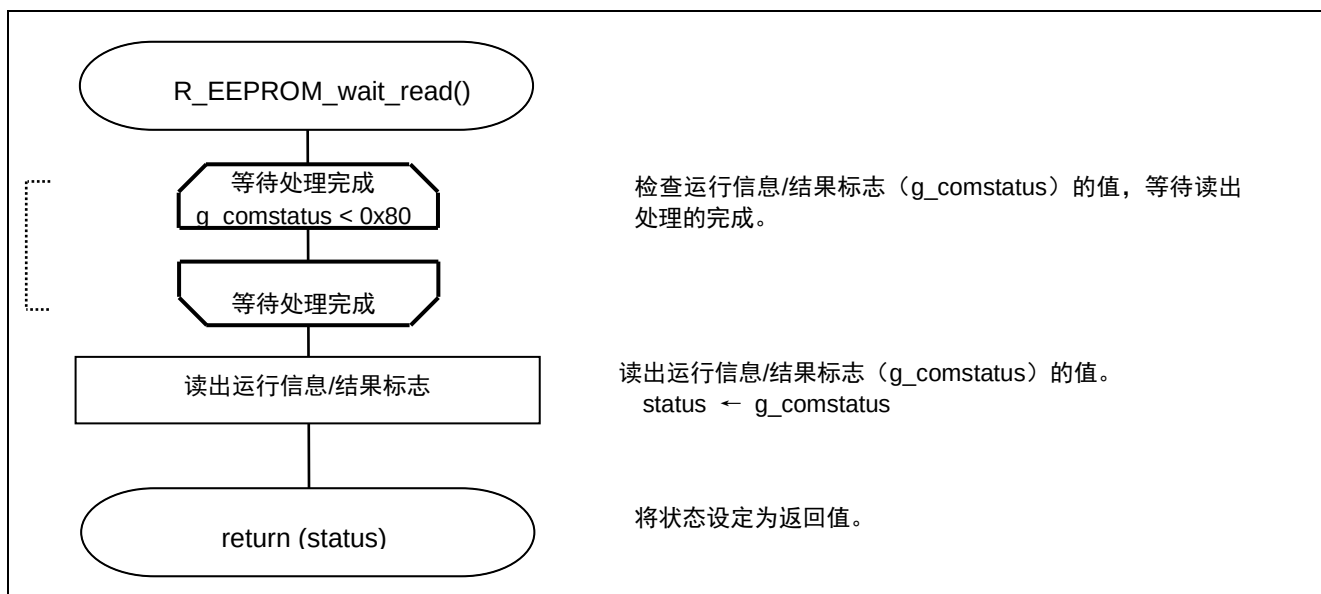


图 5.23 等待 EEPROM 读出完成的处理

5.7.18 从属设备地址发送完成处理

图 5.24 所示为从属设备地址发送完成处理。

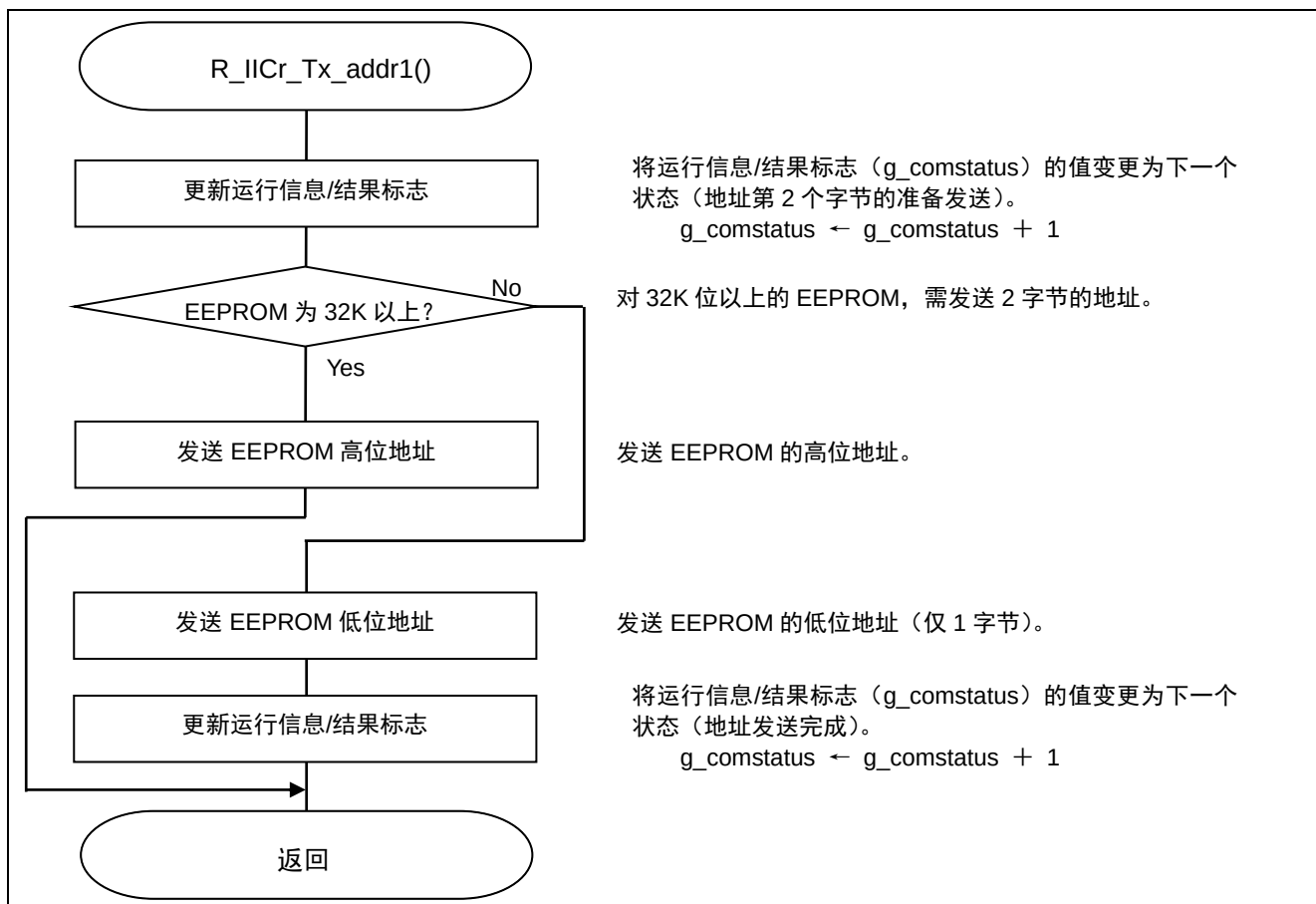


图 5.24 从属设备地址发送完成处理

5.7.19 高位地址发送完成处理

图 5.25 所示为高位地址发送完成处理的流程图。

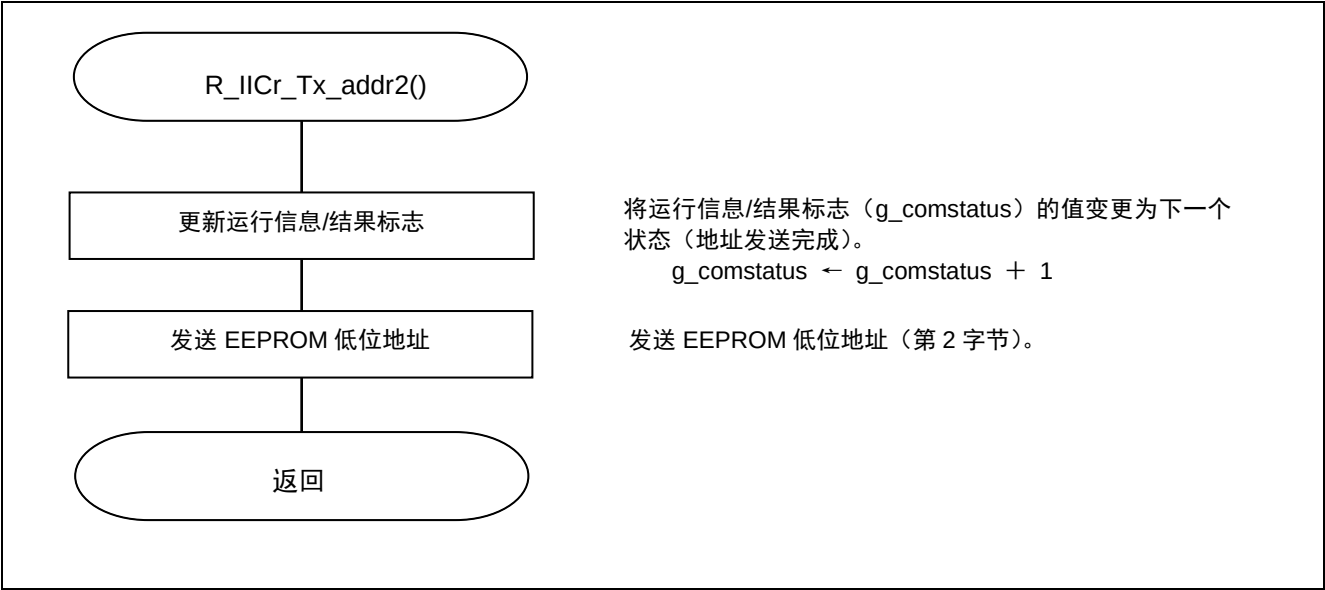


图 5.25 高位地址发送完成处理

5.7.20 重新开始处理

图 5.26 所示为重新开始处理的流程图。



图 5.26 重新开始处理

5.7.21 数据接收开始处理

图 5.27 所示为数据接收开始处理的流程图。

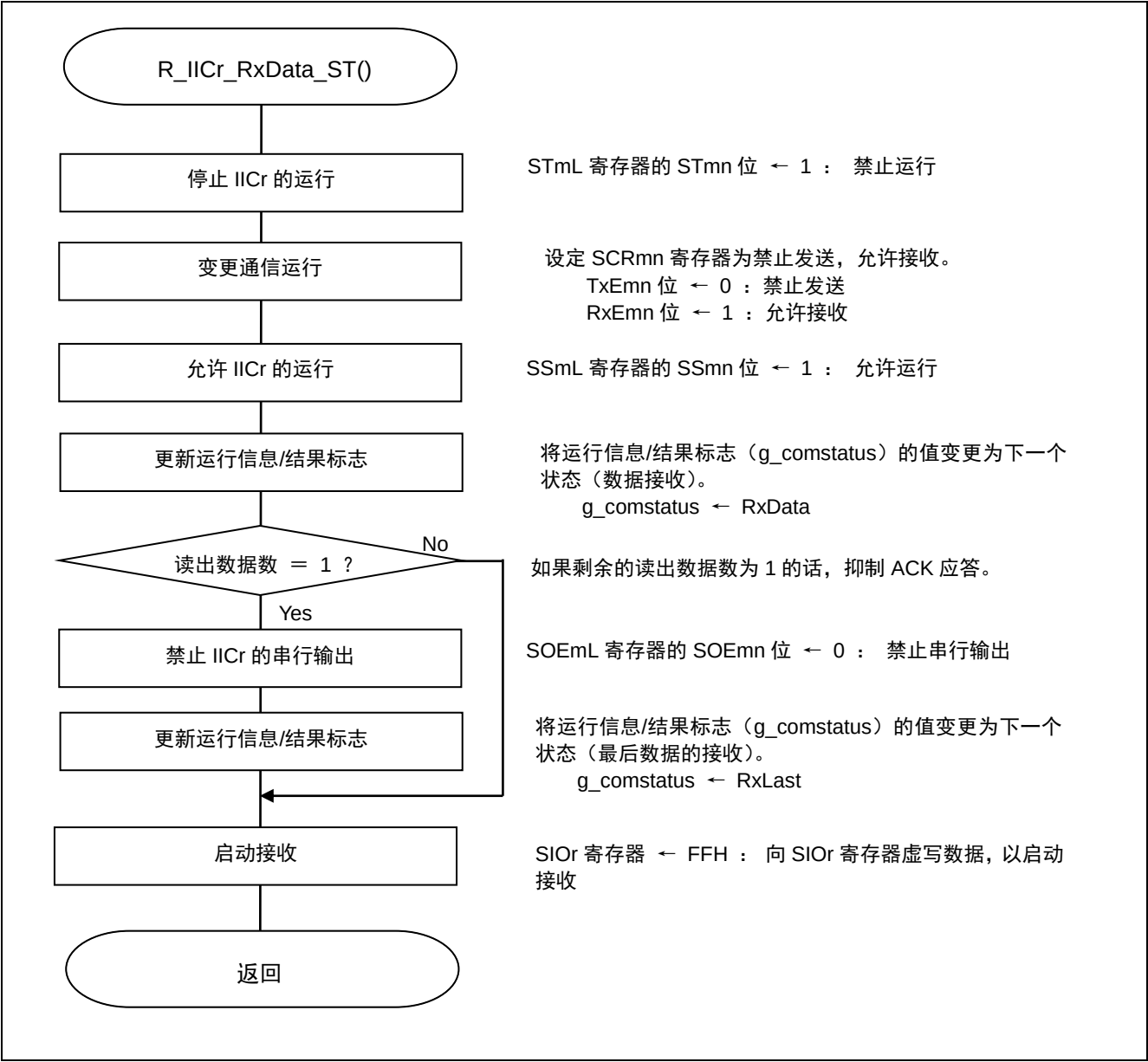


图 5.27 数据接收开始处理

5.7.22 数据接收处理

图 5.28 所示为数据接收处理的流程图。

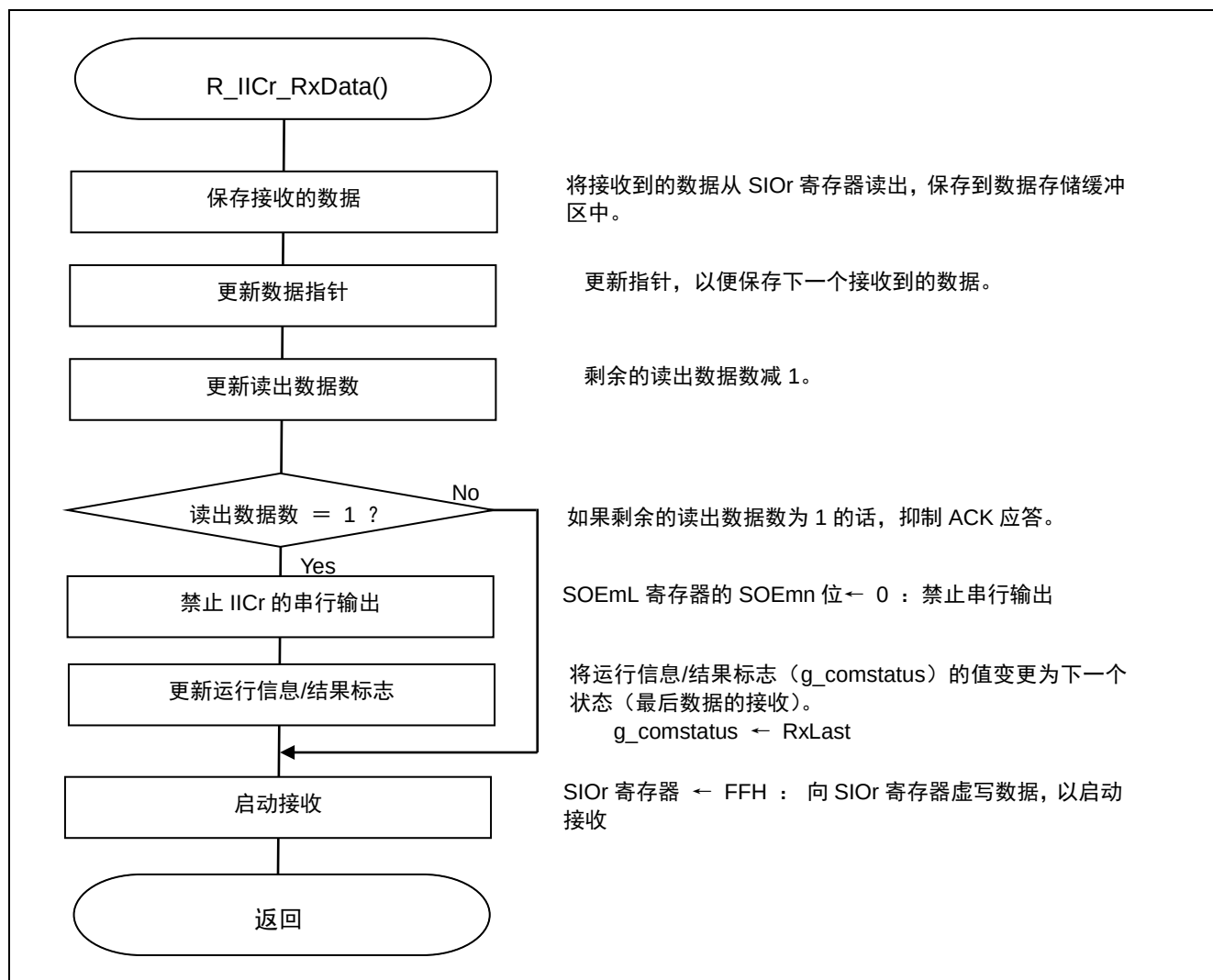


图 5.28 数据接收处理

5.7.23 最后数据接收处理

图 5.29 所示为最后数据接收处理的流程图。

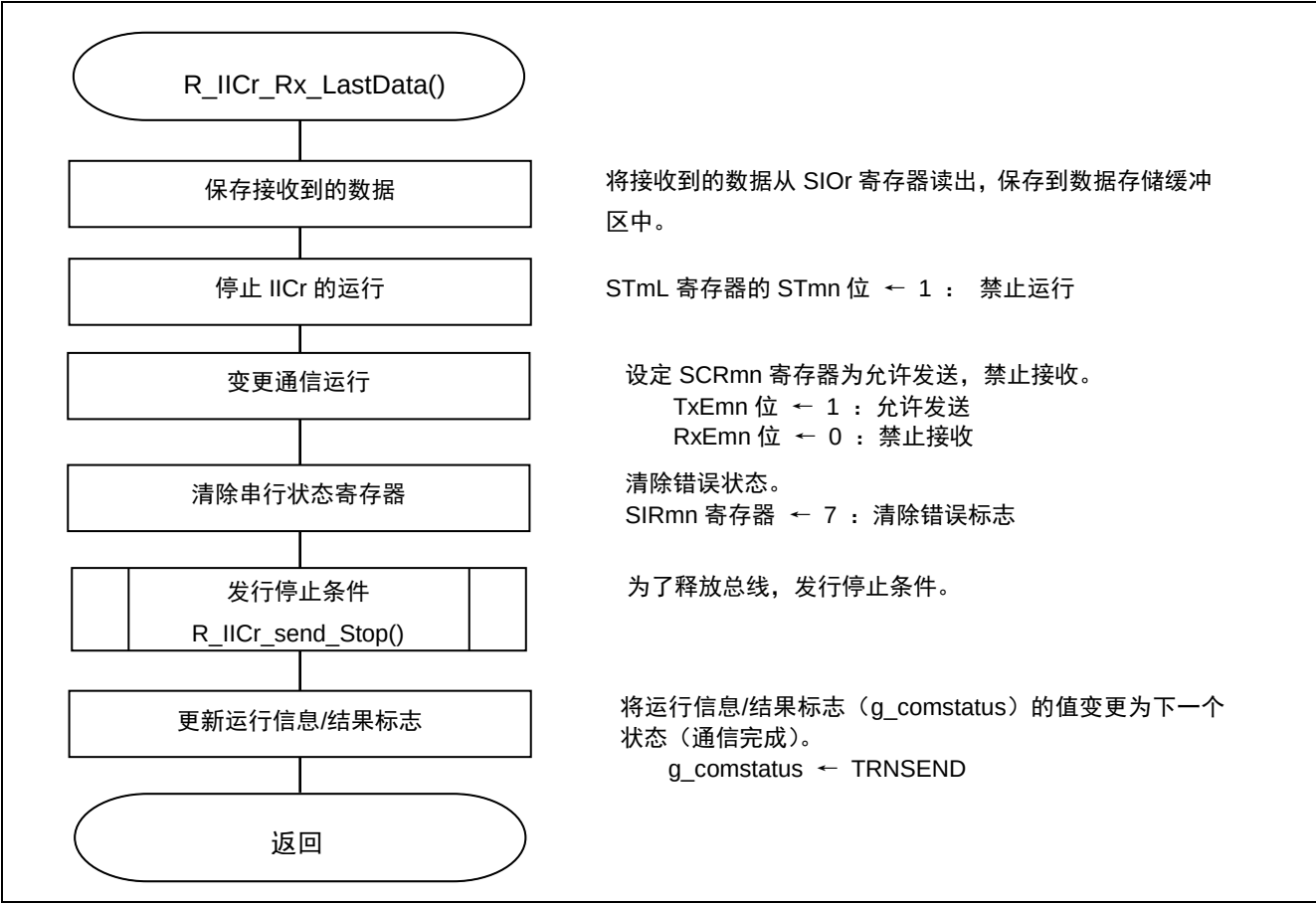


图 5.29 最后数据接收处理

5.7.24 数据发送开始处理

图 5.30 为数据发送开始处理的流程图。

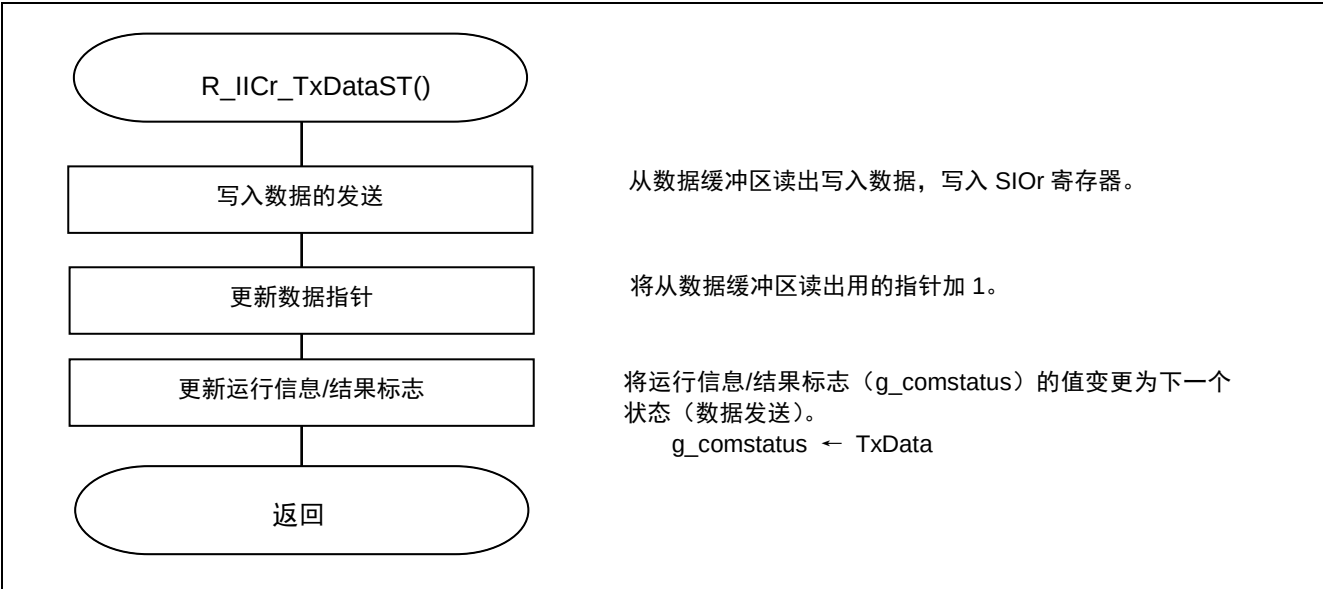


图 5.30 数据发送开始处理

5.7.25 数据发送处理

图 5.31 所示为数据发送处理的流程图。

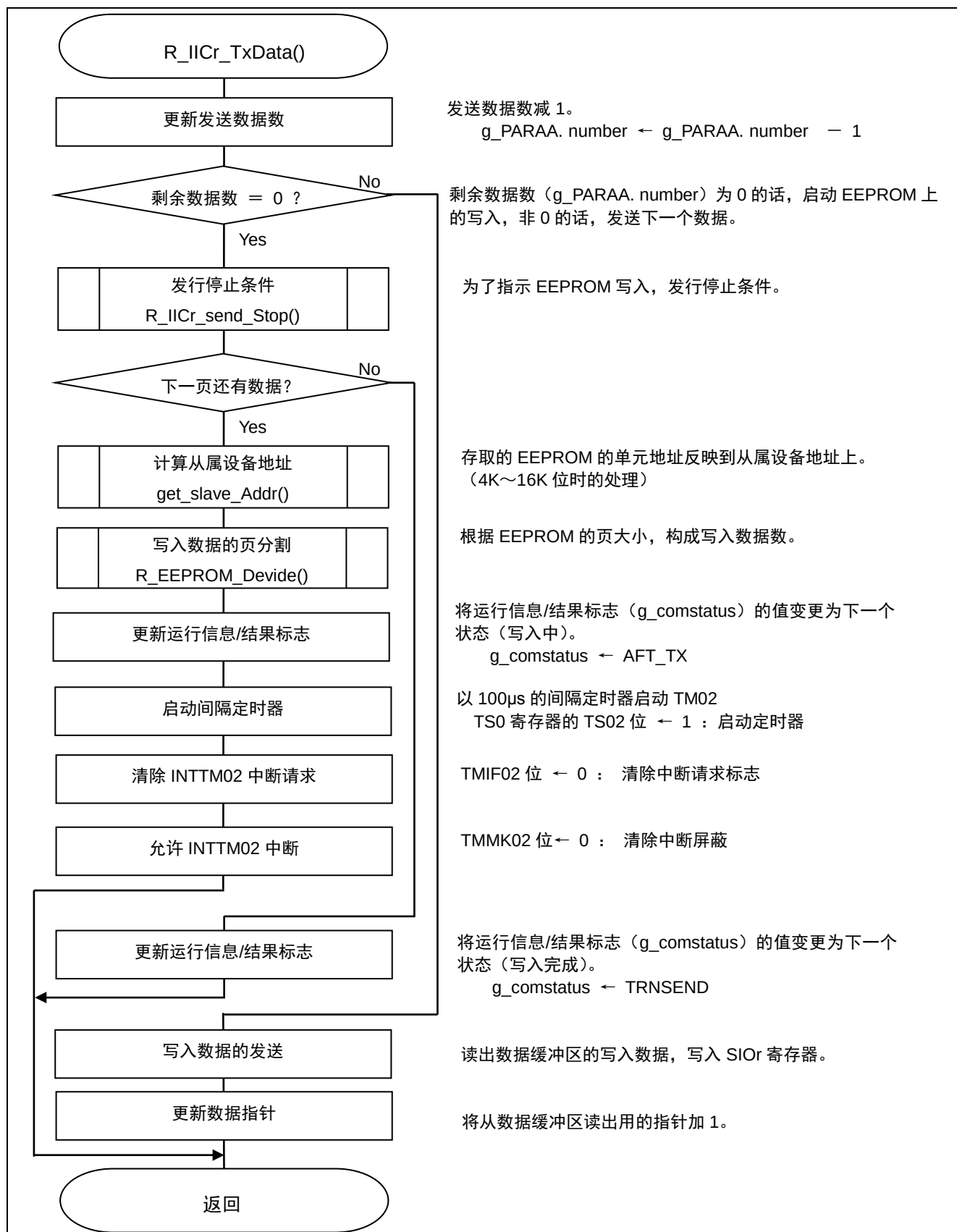


图 5.31 数据发送处理

5.7.26 下一页写入开始处理

图 5.32 所示为下一页写入开始处理的流程图。

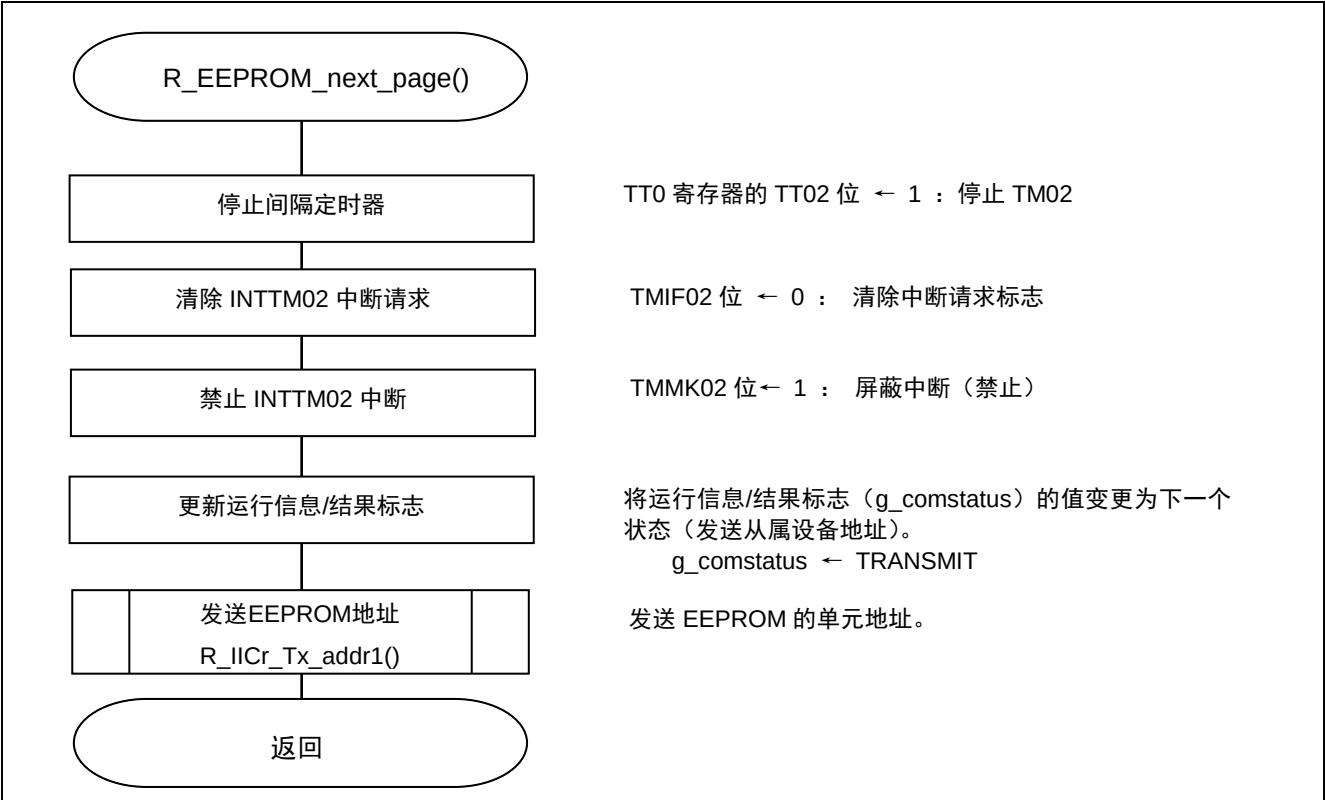


图 5.32 下一页写入开始处理

5.7.27 SCL 空时钟输出处理

图 5.33 所示为 SCL 空时钟输出处理的流程图。

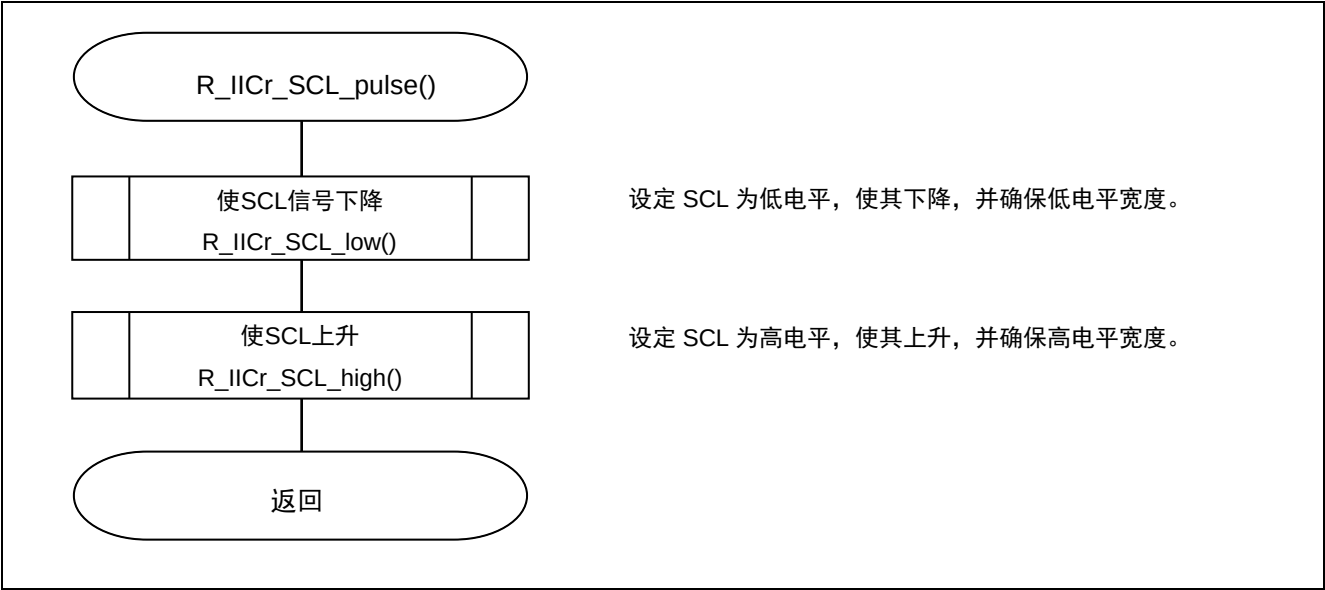


图 5.33 SCL 空时钟输出处理

5.7.28 SCL 上升处理

图 5.34 所示为 SCL 上升处理的流程图。

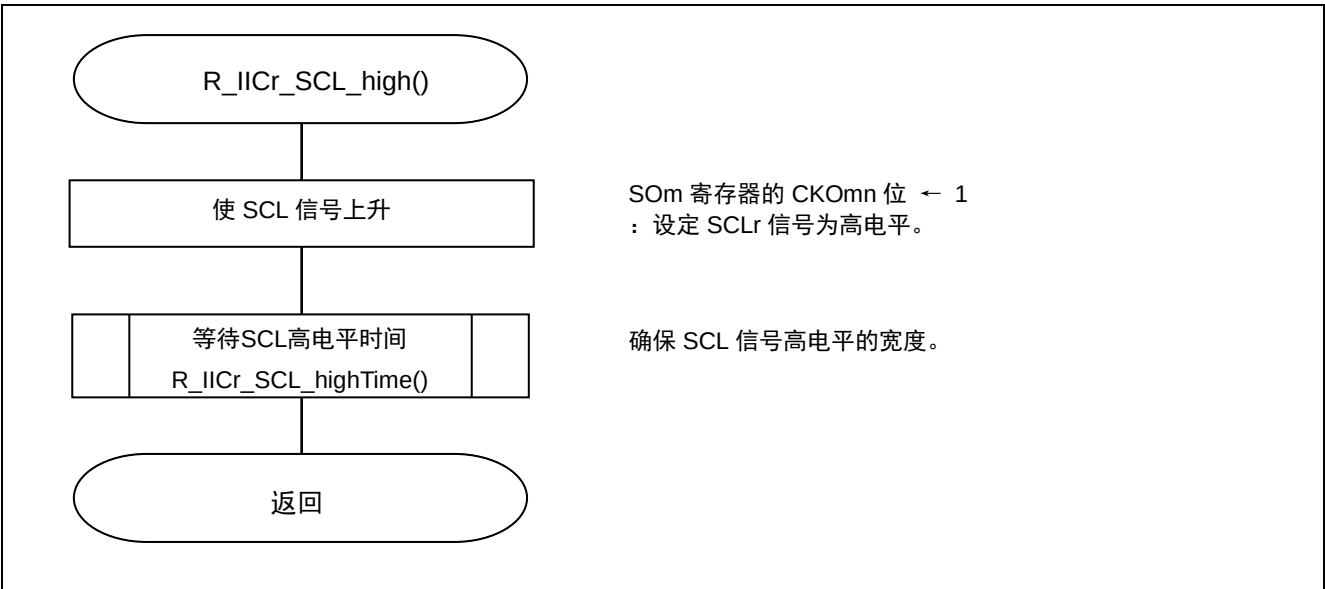


图 5.34 SCL 上升处理

5.7.29 SCL 下降处理

图 5.35 所示为 SCL 下降处理的流程图。

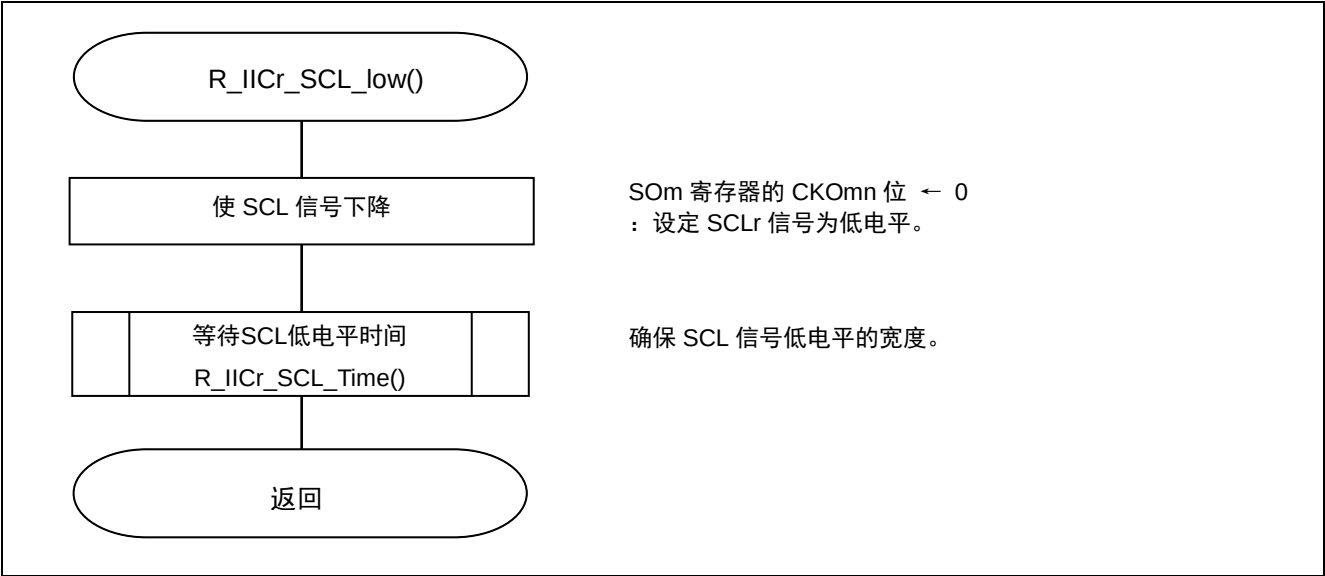


图 5.35 SCL 下降处理

5.7.30 ACK 确认处理

图 5.36 所示为 ACK 确认处理的流程图。

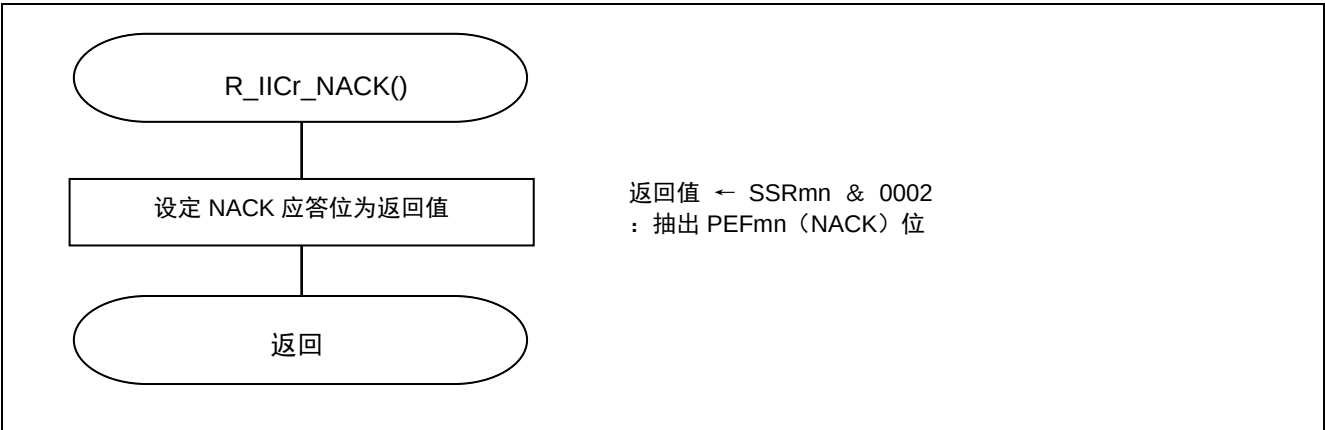


图 5.36 ACK 应答确认处理

5.7.31 SCL 低电平时间等待处理

图 5.37 所示为 SCL 低电平时间等待处理的流程图。

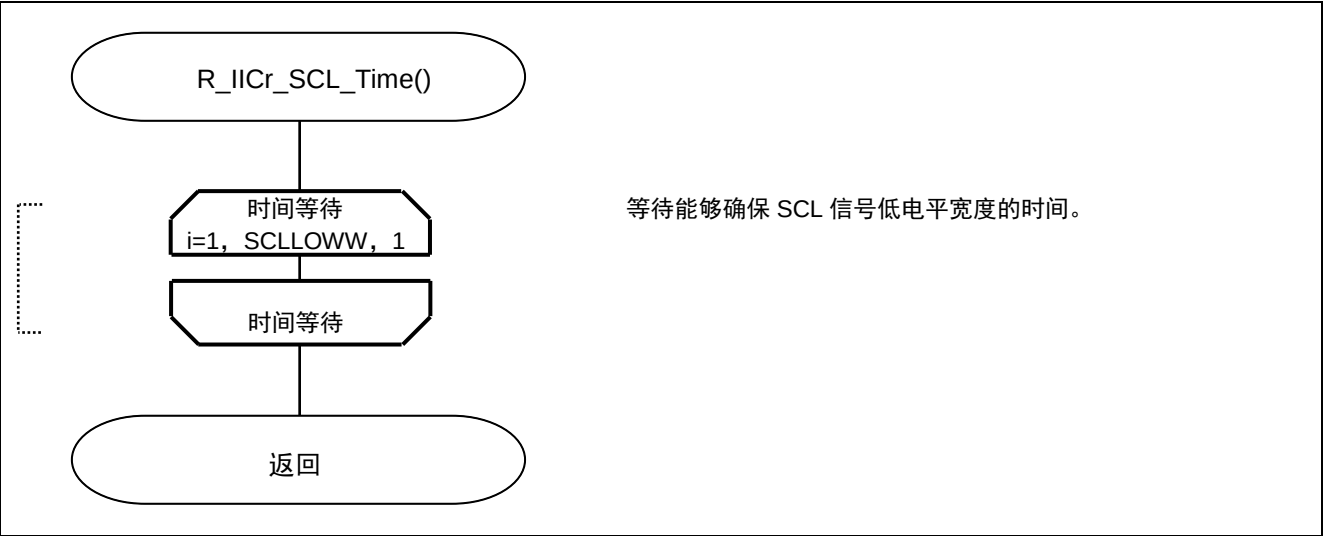


图 5.37 SCL 低电平时间等待处理

5.7.32 SCL 高电平时间等待处理

图 5.38 所示为 SCL 高电平时间等待处理的流程图。

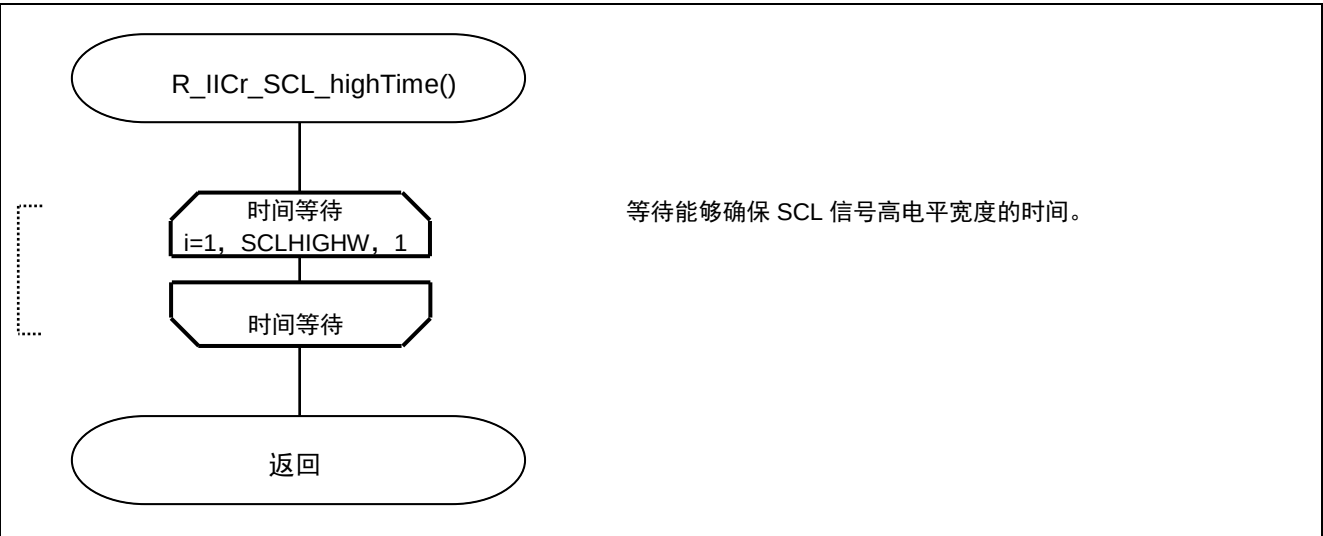


图 5.38 SCL 高电平时间等待处理

5.7.33 INTIICr 中断处理

图 5.39～图 5.41 所示为 INTIICr 中断处理的流程图。

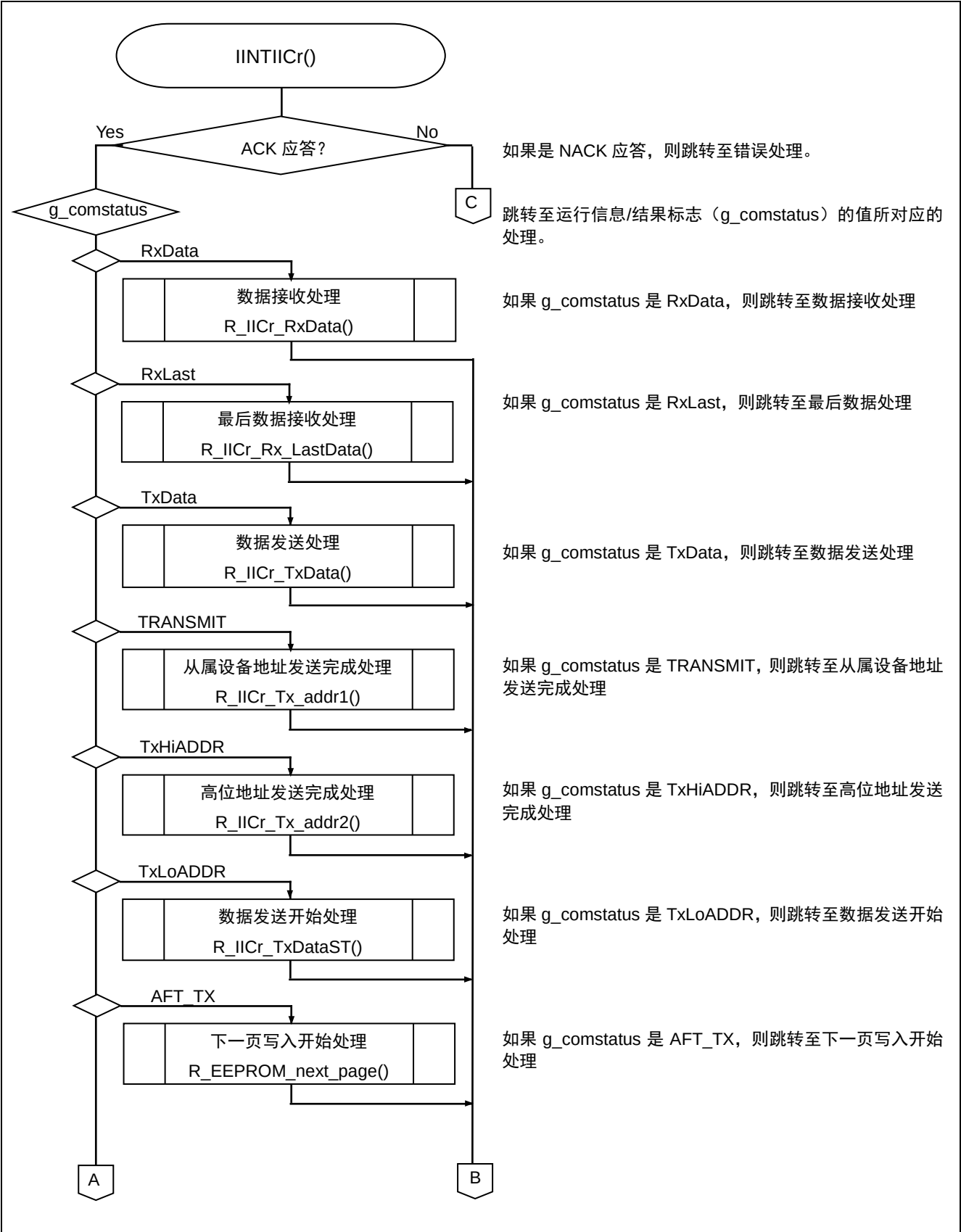


图 5.39 INTIICr 中断处理（1/3）

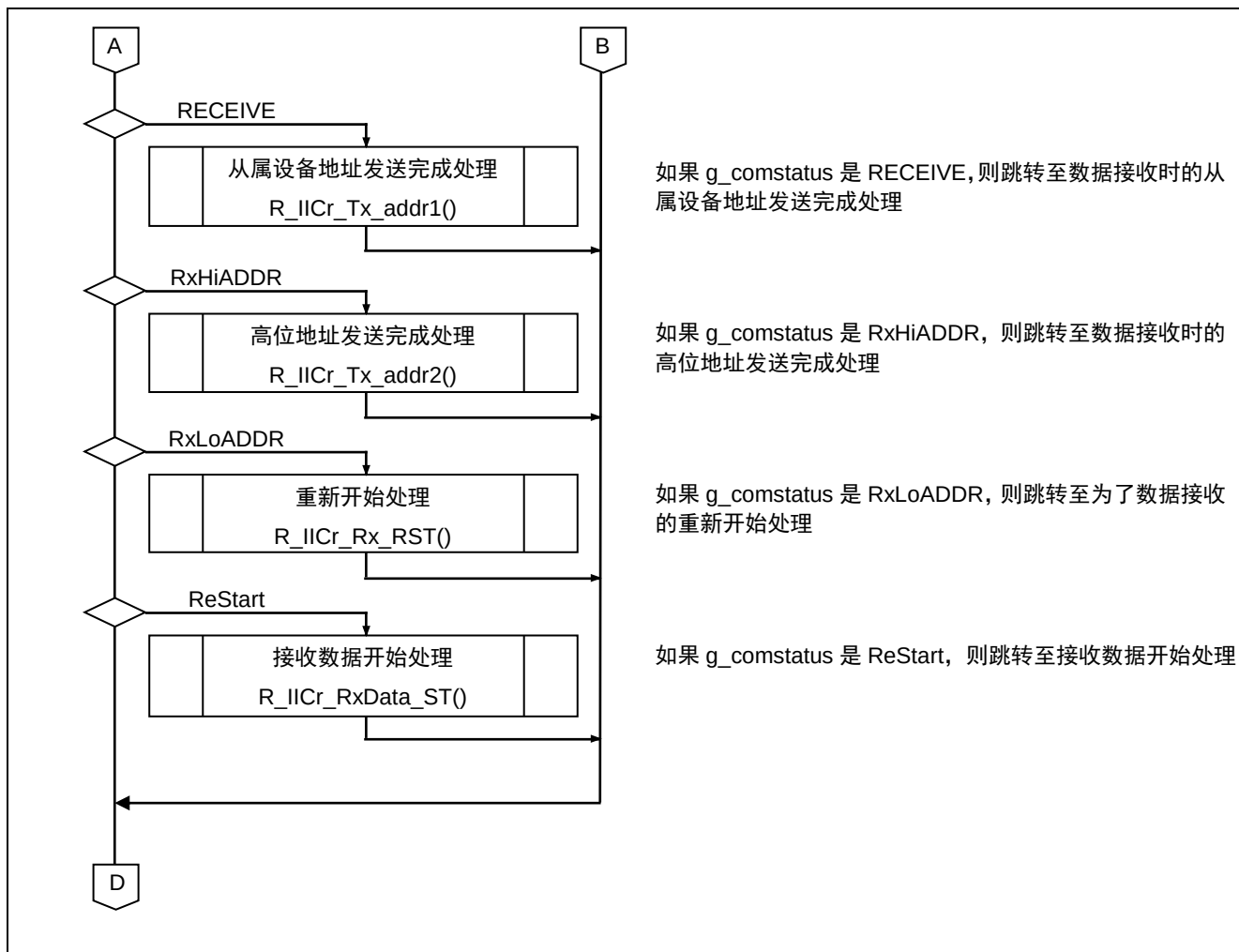


图 5.40 INTIICr 中断处理 (2/3)

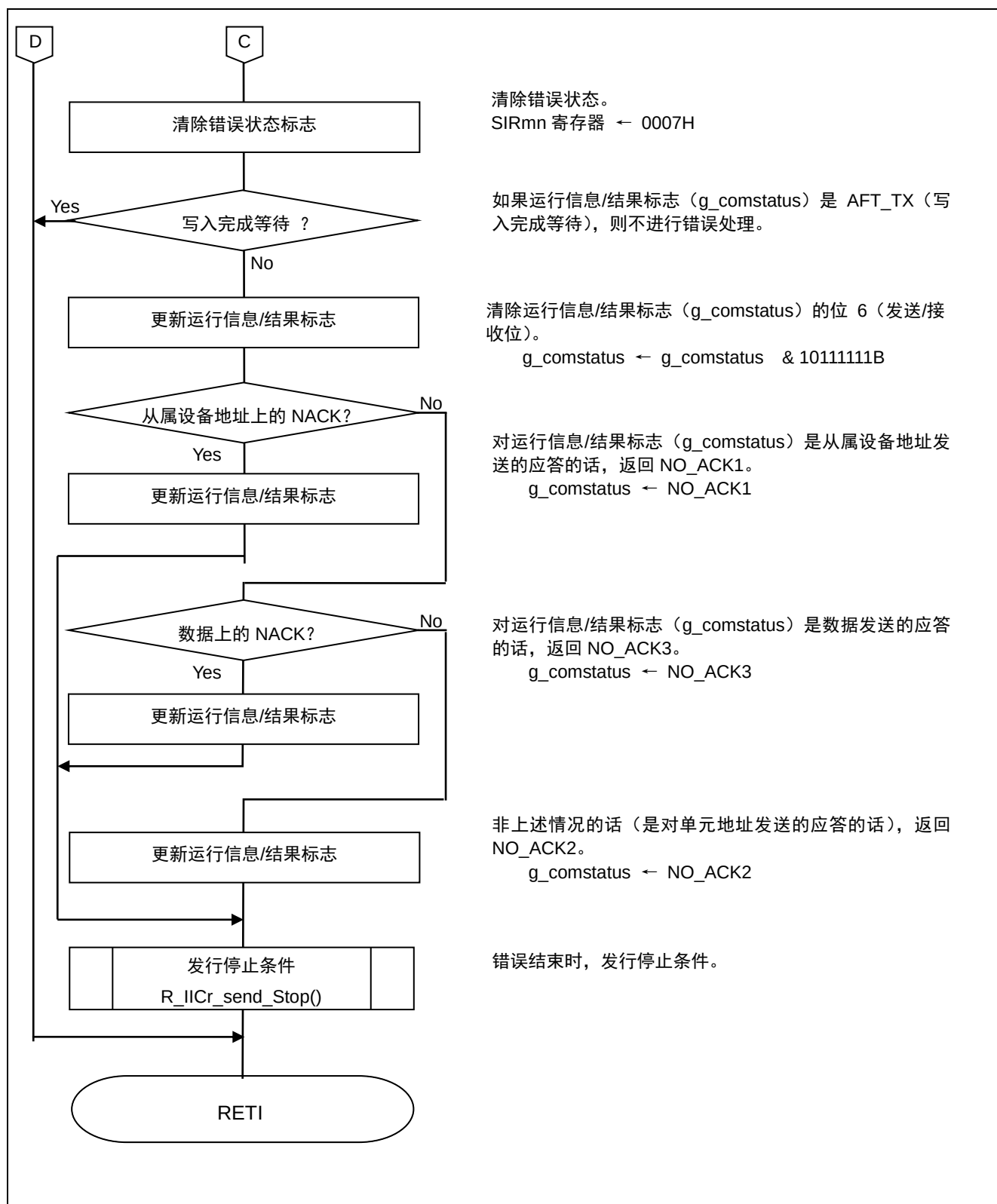


图 5.41 INTIICr 中断处理 (3/3)

5.7.34 INTTM02 中断处理

图 5.42 所示为 INTTM02 中断处理的流程图。

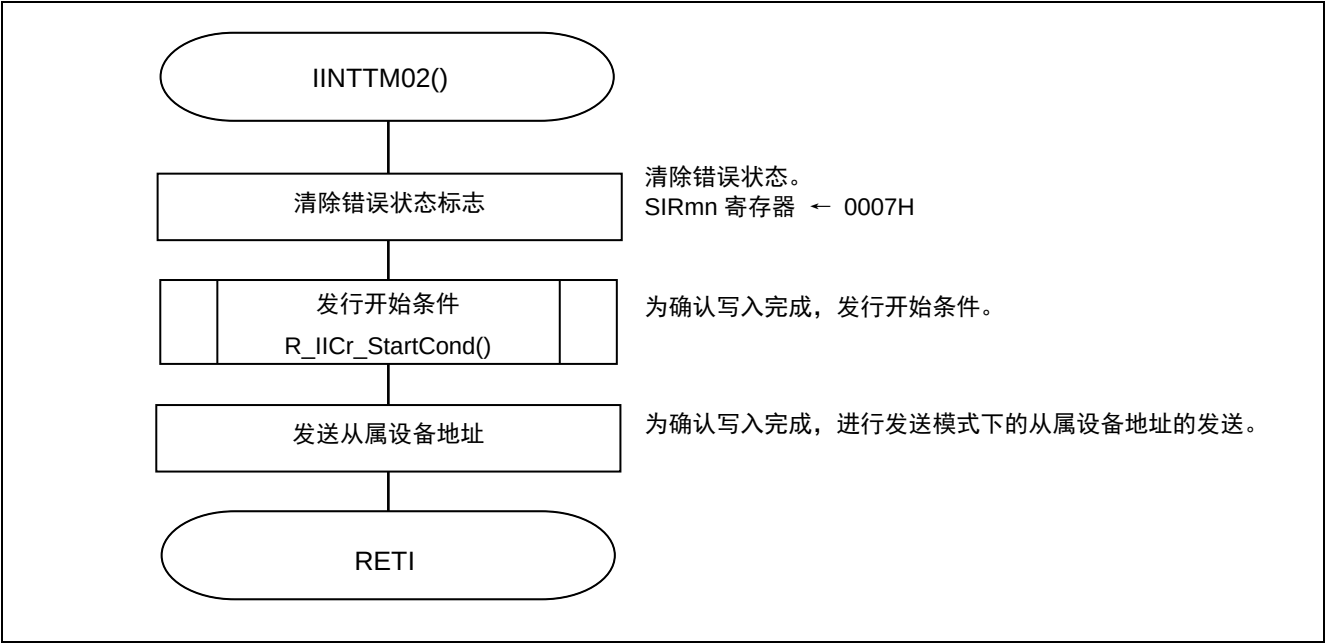


图 5.42 INTTM02 中断处理

6. 参考例程

参考例程请从瑞萨电子网页上取得。

7. 参考文献

R7F0C004 用户手册 硬件篇 (R01UH0393EJ)

RL78 family User's Manual: Software (R01US0015EJ)

(最新版本请从瑞萨电子网页上取得)

技术信息/技术更新

(最新信息请从瑞萨电子网页上取得)

公司主页和咨询窗口

瑞萨电子主页

- <http://cn.renesas.com/>

咨询

- <http://www.renesas.com/inquiry>
- contact.china@renesas.com

修订记录

Rev.	发行日	修订内容	
		页	要点
1.00	2015.03.31	—	初版发行

所有商标及注册商标均归其各自拥有者所有。

产品使用时的注意事项

本文对适用于单片机所有产品的“使用时的注意事项”进行说明。有关个别的使用时的注意事项请参照正文。此外，如果在记载上有与本手册的正文有差异之处，请以正文为准。

1. 未使用的引脚的处理

【注意】将未使用的引脚按照正文的“未使用引脚的处理”进行处理。

CMOS产品的输入引脚的阻抗一般为高阻抗。如果在开路的状态下运行未使用的引脚，由于感应现象，外加LSI周围的噪声，在LSI内部产生穿透电流，有可能被误认为是输入信号而引起误动作。未使用的引脚，请按照正文的“未使用引脚的处理”中的指示进行处理。

2. 通电时的处理

【注意】通电时产品处于不定状态。

通电时，LSI内部电路处于不确定状态，寄存器的设定和各引脚的状态不定。通过外部复位引脚对产品进行复位时，从通电到复位有效之前的期间，不能保证引脚的状态。

同样，使用内部上电复位功能对产品进行复位时，从通电到达到复位产生的一定电压的期间，不能保证引脚的状态。

3. 禁止存取保留地址（保留区）

【注意】禁止存取保留地址（保留区）

在地址区域中，有被分配将来用作功能扩展的保留地址（保留区）。因为无法保证存取这些地址时的运行，所以不能对保留地址（保留区）进行存取。

4. 关于时钟

【注意】复位时，请在时钟稳定后解除复位。

在程序运行中切换时钟时，请在要切换成的时钟稳定之后进行。复位时，在通过使用外部振荡器（或者外部振荡电路）的时钟开始运行的系统中，必须在时钟充分稳定后解除复位。另外，在程序运行中，切换成使用外部振荡器（或者外部振荡电路）的时钟时，在要切换成的时钟充分稳定后再进行切换。

5. 关于产品间的差异

【注意】在变更不同型号的产品时，请对每一个产品型号进行系统评价测试。

即使是同一个群的单片机，如果产品型号不同，由于内部ROM、版本模式等不同，在电特性范围内有时特性值、动作容限、噪声耐量、噪声辐射量等不同。因此，在变更不同型号的产品时，请对每一个型号的产品进行系统评价测试。

Notice

- 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.
- 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.
- 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.
- You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
- Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantation etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
- 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.
- Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
- 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.
- Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
- It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
- This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
- 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.

以下"注意事项"为从英语原稿翻译的中文译文，仅作参考译文，英文版的"Notice"具有正式效力。

注意事项

- 本文件中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文件中的电路、软件 and 相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，瑞萨电子不承担任何责任。
- 在准备本文件所记载的信息的过程中，瑞萨电子已尽量做到合理注意，但是，瑞萨电子并不保证这些信息都是准确无误的。用户因本文件中所记载的信息的错误或遗漏而遭受的任何损失，瑞萨电子不承担任何责任。
- 对于因使用本文件中的瑞萨电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，瑞萨电子不承担任何责任。本文件所记载的内容不应视为对瑞萨电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
- 用户不得更改、修改、复制或制作或以其他方式部分或全部地非法使用瑞萨电子的任何产品。对于用户或第三方因上述更改、修改、复制或其他方式非法使用瑞萨电子产品的行为而遭受的任何损失，瑞萨电子不承担任何责任。
- 瑞萨电子产品根据其质量等级分为两个等级：“标准等级”和“高质量等级”。每种瑞萨电子产品的推荐用途均取决于产品的质量等级，如下所示：
标准等级：计算机、办公设备、通讯设备、测试和测量设备、视听设备、家用电器、机械工具、个人电子设备以及工业机器人等。
高质量等级：运输设备（汽车、火车、轮船等）、交通控制系统、防灾系统、预防犯罪系统以及安全设备等。
瑞萨电子产品无意用于且未被授权用于可能对人类生命造成直接威胁的产品或系统及可能造成人身伤害的产品或系统（人工生命维持装置或系统、植埋于体内的装置等）中，或者可能造成重大财产损失的产品或系统（核反应堆控制系统、军用设备等）中。在将每种瑞萨电子产品用于某种特定应用之前，用户应先确认其质量等级。不得将瑞萨电子产品用于超出其设计用途之外的任何应用。对于用户或第三方因将瑞萨电子产品用于其设计用途之外而遭受的任何损害或损失，瑞萨电子不承担任何责任。
- 使用本文件中记载的瑞萨电子产品时，应在瑞萨电子指定的范围内，特别是在最大额定值、电源工作电压范围、移动电源电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用瑞萨电子产品而产生的故障或损失，瑞萨电子不承担任何责任。
- 虽然瑞萨电子一直致力于提高瑞萨电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，瑞萨电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当瑞萨电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。由于难于对微机电系统单独进行评估，所以请用户自行对最终产品或系统进行安全评估。
- 关于环境保护方面的详细内容，例如每种瑞萨电子产品的环境兼容性等，请与瑞萨电子的营业部门联系。使用瑞萨电子产品时，请遵守对管制物质的使用或含量进行管理的所有相应法律法规（包括但不限于《欧盟RoHS指令》）。对于因用户未遵守相应法律法规而导致的损害或损失，瑞萨电子不承担任何责任。
- 不可将瑞萨电子产品和技术用于或者嵌入日本国内或海外相应的法律法规所禁止生产、使用及销售的任何产品或系统中。也不可将本文件中记载的瑞萨电子产品或技术用于与军事应用或者军事用途有关的任何目的（如大规模杀伤性武器的开发等）。在将本文件中记载的瑞萨电子产品或技术进行出口时，应当遵守相应的出口管制法律法规，并按照上述法律法规所规定的程序进行。
- 向第三方分销或处分产品或者以其他方式将产品置于第三方控制之下的瑞萨电子产品买方或分销商，有责任事先向上述第三方通知本文件规定的内容和条件；对于用户或第三方因非法使用瑞萨电子产品而遭受的任何损失，瑞萨电子不承担任何责任。
- 在事先未得到瑞萨电子书面认可的情况下，不得以任何形式部分或全部转载或复制本文件。
- 如果对本文件所记载的信息或瑞萨电子产品有任何疑问，或者用户有任何疑问，请向瑞萨电子的营业部门咨询。

(注1) 瑞萨电子：在本文件中指瑞萨电子株式会社及其控股子公司。

(注2) 瑞萨电子产品：指瑞萨电子开发或生产的任何产品。



SALES OFFICES

Renesas Electronics Corporation

<http://www.renesas.com>

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

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

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

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

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

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

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

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

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

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

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

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

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-5737, Fax: +82-2-558-5141