文档编号：AN_037

上海东软载波微电子有限公司

应用笔记

HR7P169/170

修订历史

版本	修订日期	修改概要
V1.0	2014-04-28	初版
V1.1	2014-06-30	修改从动 I2C 模块的说明和例程
V1.2	2014-11-19	1. 更新文档模板 2. 更新 I2C 从动模块例程
V1.3	2015-02-05	增加芯片 B 版和 D 版差异说明
V1.4	2015-03-16	1. 删除芯片 B 版和 D 版差异说明 2. 增加烧录数据 FLASH 存储器的方法
V1.5	2015-05-18	修改内部 Flash 读写和 UART 模块例程
V1.6	2015-07-28	统一修改公司名称、logo 及网址等
V1.7	2015-11-11	增加 ADC 输入缓冲使用注意事项
V1.8	2016-02-23	修改复位模块和低功耗模式的描述
V1.9	2018-01-02	1. 增加 I2C 高速传输说明 2. 删除程序模块代码。
V1.10	2018-05-02	1. 新增读 Flash 操作说明 2. 新增 GIE 位和 GIEL 位处理说明
V1.11	2019-01-27	1. 增加 PWM 死区与脉宽的说明
V1.12	2019-04-02	1. 变更 Logo 2. 修改 GIE 和 GIEL 位处理说明
V1.13	2019-12-02	1. 添加 1.13 硬件乘法器注意事项

地　址：中国上海市龙漕路 299 号天华信息科技园 2A 楼 5 层
邮　编：200235
E-mail：support@essemi.com
电　话：+86-21-60910333
传　真：+86-21-60914991
网　址：http://www.essemi.com/

目 录

内容目录

第1章 HR7P169/170 应用注意

1.1 内部振荡器

HR7P169/170 芯片在出厂时已做好内部振荡器的校准，校准精度 16MHz±2%@25℃，3V~5.5V。

如果用户选择芯片内部振荡器作为系统时钟源，在芯片上电复位完成后，内部电路会自动把校准值加载到校准寄存器 OSCCALH 与 OSCCALL，完成校准操作，因此不需要通过软件进行赋值。

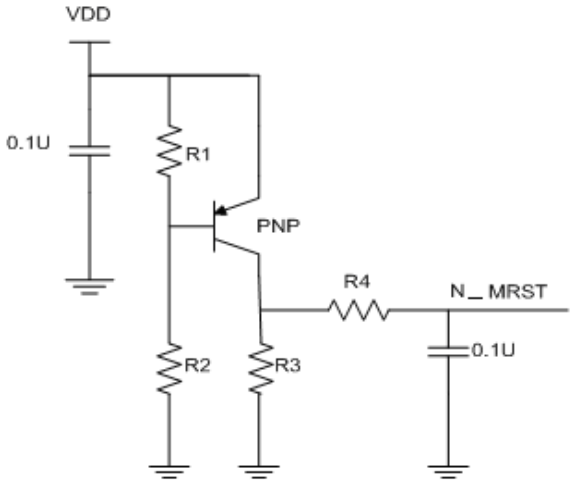
1.2 复位模块

1.2.1 外部复位

用户应避免把 N_MRST 引脚直接连接到 VDD 上,,可通过 10KΩ 电阻上拉到 VDD 的方式连接。

我们建议用户使用具有低电压检测功能的复位芯片作为外部复位电路。

当系统有低成本的要求时，用户也可以采用以下电路替代电压检测芯片。



电压检测原理：当 VDD 电压下降，导致 R1 两端电压< 0.7V 时，PNP 晶体管截止，N_MRST 引脚被 R3 电阻下拉至低电平，使芯片处于复位状态。

复位电压点应满足 [R1/(R1+R2)]×VDD < 0.7V 这个条件，用户可以此进行复位电压和匹配电阻的计算。

举例：

选定 R1=3.3K，R2=10K，R3=20K，R4=1K，复位电压应满足：

[2K/(2K+10K)]×VDD < 0.7V，通过计算可以得到，当 VDD < 4.2V 时，PNP 晶体管处于截止状态，N_MRST 被拉至低电平，可保证芯片处于复位状态。

1.2.2 BOR复位

BOR 掉电复位模块监控施加于芯片电源上的电压，一旦芯片的工作电压超出所设定的电压范围，则产生欠压复位，这样可以防止芯片 IO 端口的非正常输入/输出，有效增强系统的抗干扰性能，

提高系统的稳定性。

建议客户在设计产品时使能 BOR，并设置 BORVS 在合理的电压点。在芯片运行和进入 IDLE 模式时，都不要通过软件关闭 BOR 功能，以免芯片因外界干扰或电源波动而工作异常。

1.2.3 WDT复位

建议客户在设计产品时使能 WDT 功能。

1.2.4 SREN/RCEN控制位

为了防止低电压检测复位使能位 SREN 和内部 RC 时钟使能位 RCEN 受干扰被清 0，建议在程序主循环中置 1。

设置 SREN 和 RCEN 程序示例:

```
BSS     PWEN, SREN          ;SREN=1
BSS     PWEN, RCEN          ;RCEN=1
```

1.3 低功耗模式

- 如果产品封装数小于 20，未引出的 I/O 管脚需设置为输出低电平。
- 实际应用系统中，未使用的 I/O 管脚需设置为输出低电平。
- 在 IDLE 模式，当客户使用 WDT 唤醒时，RCEN 不能清零。

1.4 唤醒方式操作

在进入 IDLE 之前，需注意清零 T8P1/T8P2/UART 中断标志位，并关闭中断使能位，以免误唤醒芯片。

1.5 中断处理

1.5.1 外部按键中断

用户在使用外部按键中断功能时，在中断程序中清除中断标志位前，必须对所有使能的外部按键输入端口进行一次读/写操作，使比较参考电平与当前输入电平保持一致，否则标志位无法被清除。

1.5.2 中断标志的清除

用户在打开中断前需先清除相应的中断标志，避免中断的误触发。

除只读的中断标志（由硬件清除）外，其余的中断标志必须通过软件清除。

1.5.3 GIE位和GIEL位处理

用户通过软件对中断使能位 GIE 或 GIEL 进行写零操作的时刻，如果同时发生了中断响应，则芯片会优先响应中断，本次软件写零操作无效。为确保对中断使能位 GIE 和 GIEL 的软件写零操作成功，推荐的实现方式如下：

```
while(GIE == 1)
{
    GIE = 0;
}
while(GIEL == 1)            //仅使用了向量中断才需要此语句
{
    GIEL = 0;
}
.....
GIEL = 1;                   //仅使用了向量中断才需要此语句
GIE = 1;
```

用户在对 GIE 和 GIEL 的操作中，一定要严格按照上面例程的顺序进行。

1.6 未使用的IO端口处理

系统中未使用的 IO 管脚建议设置为输出固定电平并悬空， 若设置为输入， 须加上拉或下拉电阻接到电源或地。

1.7 PWM

1.7.1 PWM周期寄存器

由于每个 PWM 模块的周期寄存器没有相应的缓冲寄存器，在启动 PWM 模块后可以随时对精度寄存器的缓冲器赋值，但是不能对 PWM 模块周期寄存器进行赋值操作。只有停止 PWM 模块输出后，才能对相应的 PWM 模块周期寄存器进行赋值操作。

1.7.2 PWM死区与脉宽的关系

死区时间由设置的系统时钟的分频值(由 PDDxPR<1:0>选择)和死区控制寄存器 PDDxC<6:0>的值决定。Tdelay = Tosc×死区时钟分频比×（PDDxC<6:0>）。死区时间必须小于 PWM 的脉宽，否则 PWM 会一直输出低电平。

1.8 I2C从动模块操作

I2C 从动模块支持 7 位从机地址匹配，由 I2C 主机控制发送或接收数据。

当主机向从机发送数据时，从机通常判断 I2CRBIF 标志，如果接收缓冲器不空，即接收到主机数据，则读接收缓冲器的数据。

当主机读取从机数据时，从机通常判断 I2CTBIF 标志，如果发送缓冲器未满，则依次写入需要发送的数据。

为了避免误发数据，建议每次完整的通讯结束（例如收到 STOP 标志），就采用 I2CRST 置位复位一次 I2C 模块来清空接收和发送数据缓冲器，同时再重新初始化 I2CC 和 I2CIEC 寄存器，为下次 I2C 通讯做好准备。

例程中低速传输用的是默认的中断模式，需要注意的是默认中断模式的中断入口只有一个，不同的中断源所产生的中断都会进入到同一个中断函数进行处理，如果前一个中断相关的程序执行的太久便会影响别的中断相关程序的执行。为了降低 I2C 做从机时不同中断处理对其传输速度影响，在设计程序时尽量首先判断 I2C 的中断标志位，以首先执行 I2C 中断相关程序；尽量减少每个中断源对应的中断程序运行的时间；尽量在判断不同中断源（例如定时器和 I2C 中断）时，使用如下格式的语句判断：

```
“if(I2CIF&&I2CIE)
  {;}
else if(T8NIF && T8NIE)
  {;}”。
```

使用默认中断模式传输时，考虑到不同中断源之间的影响，建议传输速度不要大于 10KHz。

实际开发中，当主机读从机，I2C 从机模块进入到地址匹配中断（I2CSRIF==1）之后，从机往往会有一些操作（比如对即将发送的数据做处理），造成数据短时间不能写入到 I2CTB 寄存器内，如果延迟时间超出了主机的最大等待时间，传输便会失败。低速传输时，主机有着充足的时间等待从机发送数据，当高速传输时，往往就会出现上述问题。解决这个问题可以使用 iDesigner 自带的优化处理方法。用户需要在编译器“项目->编译->Support interrupt vectors”选择“true”和“项目->编译->IIC slave high speed mode”选择“true”。使用编译器优化之后，当主机读从机时，从机只需要在写入数据到 I2CTB 寄存器之后释放 SCL 引脚（置位 I2CTE），主机接下来便会开始接收从机所发的数据。

需要特殊注意的是，若使用编译器优化，在编程时用户必须使用向量中断，中断全局寄存器 INTG 中的 INTV<1:0>必须为 0b11，使得 I2C 的中断优先级位于最高位，而优先级比 I2C 中断 IG6 高一级的 IG7 中断对 I2C 的传输速度也会产生影响，所以用户应尽量避免使用 IG7 中断。不同的应用环境下 I2C 的传输速度会有差别，高速传输时建议传输的最大速度小于 600KHz，当用户传输的速度大于 400KHz 时，建议关闭 I2C 滤波功能（I2CX16 = 0）。

1.9 唤醒方式操作

在进入 IDLE 之前，需注意清零 T8P1/T8P2/UART 中断标志位，并关闭中断使能位，以免误唤醒芯片。

1.10 FLASH IAP操作

当 FLASH 存储器进行 IAP 擦除或 IAP 写入操作时 CPU 内核暂停执行，外设可按预设状态继续运行，外设的中断请求将置位相应的中断标志。当 IAP 擦除或 IAP 写入操作完成时，CPU 内核恢复执行。

在编译中断子程序的过程中，编译器会保存 FRAH/FRAL 的地址，并在退出中断时，通过 TBR 指令，再次读取 FRAH/FRAL 对应地址单元的值到 ROMDH/ROMDL 中。因此不建议用户在完成读、

写、擦操作后更改 FRAH/FRAL 的值。从而对程序的功能造成影响。

推荐	不推荐

例 1：u8 ReadFlash()
{
while(GIE == 1)
{
GIE = 0;
}
 	
FRAH = 0x40;
FRAL = 0x00;
__asm
{
TBR
}
FlashData.Byte[1] = ROMDH;
FlashData.Byte[0] = ROMDL;
GIE=1;
retrun ROMDL;
}
	u8 ReadFlash()
	{
	while(GIE == 1)
	{
	GIE = 0;
	}
	
	FRAH = 0x40;
	FRAL = 0x00;
	__asm
	{
	TBR
	}
	FlashData.Byte[1] = ROMDH;
	FlashData.Byte[0] = ROMDL;
	FRAH=0X47;
	FRAL=0X00;
	GIE=1;
	return ROMDL;
	}

在上面的例程中，执行右边的程序,会发现最后返回的 ROMDL 结果并不一定是 0x4000 地址单元存储的数据，在打开 GIE 后，如果此时有一个中断需要响应，在退出中断子程序之前会执行“隐含”的 TBR 指令，而操作对象，正是在打开 GIE 之前刚被修改的 0x4700 地址，因此在执行完中断子程序再返回到这段代码后，程序返回的将是 0x4700 地址单元的数据。

1.11 烧录数据FLASH

编译器支持由程序填写初始化值到芯片的数据 FLASH 存储器，编译后初始化值会包含在 Hex 文件中。当用户调用该 Hex 文件对芯片进行编程时，初始化值将被固化到指定地址的数据 FLASH 中。

具体操作方法是，在 iDesigner 软件项目工程中，添加汇编程序，例如：eeprom_init.asm。程序中采用“EEPROM”伪指令定义数据 FLASH，随后以“DW”字类型定义具体数据。需要注意的是，在“EEPROM”伪指令前，需要增加一条“ORG”伪指令定义数据 FLASH 存储器的起始地址，否则数据将从数据 FLASH 存储器的首地址开始存储。

初始化数据 FLASH 程序示例：

```
ORG 0x4010
```

```
EEPROM
UserData DW 0x1234, 0x2345, 0x3456          ;数据内容，3 个字
END
```

1.12 ADC输入缓冲器

当 ADC 模块的输入缓冲器 AINEN=1 使能时，能够检测到的电压范围为 0.2V~（VDD-0.2）V，此时无法检测到 mV 级的小信号（详见数据手册 ADC 参数寄存器 ADCTST）。

如果需要 ADC 检测全电压范围时，应禁止 ADC 模块的输入缓冲器，即 AINEN=0。

1.13 硬件乘法器

用户在使用芯片硬件乘法器时，要注意中断服务程序可能会改变乘数寄存器 MULA 和 MULB，最终导致程序运行获取到一个错误的乘积。用户有二种方式来规避这种风险。

方式一：用户在使用硬件乘法器之前，先禁止全局中断使能（GIE=0），以免在中断处理过程中，乘数寄存器被改写。乘法运算完成后，将乘积读出，再恢复全局中断使能（GIE=1）。

方式二：用户在使用硬件乘法器之前，先将乘数和被乘数备份在特定的变量中。这样，编译器会在中断服务程序中自动备份和恢复乘数寄存器。注：需使用 v1.2.0.113 及以上版本的工具链。

方式二示例如下：

```
unsigned char __MULA__ @0x0;
unsigned char __MULB__ @0x1;
#define SET_MULA(a) {__MULA__ = (a); MULA = __MULA__;}
#define SET_MULB(a) {__MULB__ = (a); MULB = __MULB__;}

main()
{
SET_MULA(12);
SET_MULB(15);
}
```

为了方便用户使用，变量声明和宏定义，都已经添加在芯片的头文件中了。用户在实际使用时，只需执行 SET_MULA(12)和 SET_MULB(15)这二处即可。

第2章 HR7P169/170 模块例程

2.1 定时器程序模块（T8N）

功能说明：

使用芯片的 T8N 定时器模块，在 PA1 端口输出一个周期为 4ms，占空比 50%的方波。

设定 T8N 为定时器模式，定时时间为 2ms。在 T8N 定时中断服务程序中取反 PA1 端口电平，实现 50%占空比，4ms 周期的方波输出。

芯片使用 2MHz 系统时钟，则对应的 T8N 定时器时钟源周期为 1us。将预分频器分配给 T8N 定时器，分频比为 1：8。T8N 寄存器初始值的计算公式应为：

2ms/1us =（255-T8N+1）×8（预分频比），计算得到 T8N = 6(0x06)。

实现步骤：

a) 初始化系统和端口

b) 初始化 T8N 定时器

c) 使能 T8NIE，GIE 中断

d) 中断服务程序。判断中断标志，确定是 T8N 中断后则清除 T8NIF 标志位

e) 执行 PA1 端口取反输出，并重新向 T8N 寄存器赋初值

2.2 标准PWM程序模块（T8P1）

功能说明：

使用芯片的 T8P1 扩展功能模块 PWM，在 PA1 端口实现频率为 2.5KHz（周期为 400us），占空比为 50%的方波输出。

芯片使用 2MHz 系统时钟，则对应的 T8P1 定时器时钟源周期为 1us。T8P1 的预分频采用 1：4，T8P1 周期寄存器 T8P1P 初始值的计算公式应为：

PWM 周期 400us =（T8P1P+1）×1us×4（预分频比），计算得到 T8P1P = 99(0x63)。

PMM 脉宽 200us = T8P1RL ×1us×4（预分频比），计算得到 T8P1RL =50(0x32)。

PWM 占空比 50% = T8P1RL/ （T8P1P+1）。

实现步骤：

a) 初始化系统和端口

b) 初始化 T8P1 及其扩展模块并对相应的寄存器赋值

c) 使能 T8P1 的 PWM 模式

2.3 增强型EPWM程序模块（T8P2）

功能说明：

使用芯片的 T8P2 的 EPWM 模块，在 PB0、PB1 端口输出正向半桥 EPWM 信号。PWM 频率为 2.5KHz（周期为 400us），占空比为 50%。

设 EPWM 以 T8P2 为时基。芯片使用 2MHz 系统时钟，则对应的 T8P2 定时器时钟源周期为 1us。T8P2 的预分频采用 1：4，T8P2 周期寄存器 T8P2P 的初始值的计算公式应为：

PWM 周期 400us =（T8P2P+1）×1us×4（预分频比），计算得到 T8P2P = 99(0x63)。

PMM 脉宽 200us = T8P2RL ×1us×4（预分频比），计算得到 T8P2RL =50(0x32)。

PWM 占空比 50% = T8P2RL/ （T8P2P+1）。

设死区延时时间 Tdelay 为 50us，

根据公式 Tdelay = 2×Tosc×PDD2C<6:0> = 1us×PDD2C，计算得到 PDD2C = 50(0x32)。

实现步骤：

a) 初始化 T8P2 及其 PWM 模式并对相应的寄存器赋值

b) 初始化输出端口

c) 使能 T8P2 的 EPWM 模式

d) 设置 EPWM 占空比/死区/自动关断触发信号及触发后输出信号，启动 EPWM 功能

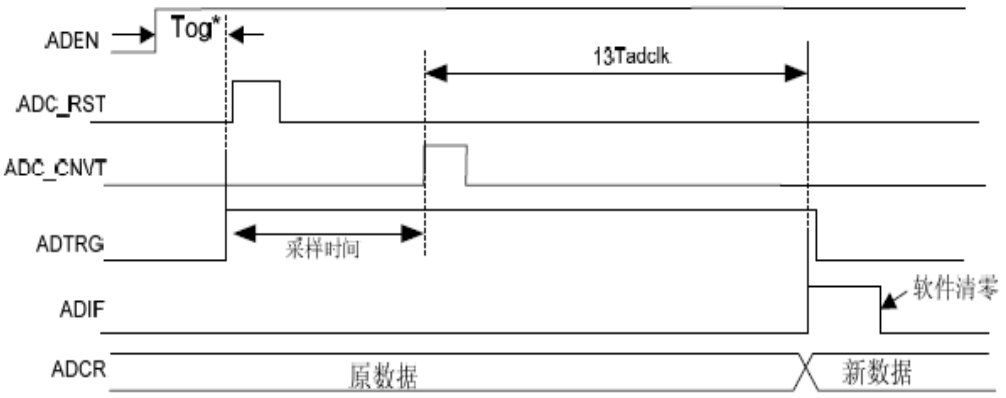
2.4 ADC程序模块

功能说明：

使用 HR7P169 芯片的 ADC 模块，采用查询方式实现对模拟输入电压的数字量转换。

HR7P169 最多支持 14 个通道输入，最大转换分辨率为 12bit。

ADC 转换包括采样和转换两个过程，高速时采样时间不得低于 0.5us，低速时采样时间不得低于 5us，转换时间为 13 个 Tad（见下图）。



注 1：Tog>0；
注 2：不推荐使用软件控制采样时间；
注 3：AD 转换时钟周期 Tadclk。

实现步骤：

a) 初始化系统和端口
b) 初始化 ADC 模块并对相应的寄存器赋值
c) 使能 ADC 模块，启动 A/D 转换
d) 查询并等待 A/D 转换结束
e) 保存 A/D 转换结果

2.5 外部按键中断程序模块

功能说明：

使用 HR7P169 的外部按键中断功能，对 KIN3（PB7 端口）外部电平变化进行判别。在全局中断使能的条件下，KIN3（PB7 端口）的电平变化会产生外部按键中断。

请参考本文章节 1.4 外部按键中断和章节 1.5 中断标志的清除的内容，在清除中断标志前，必须对所有使能的外部按键输入端口进行一次访问操作，使比较参考电平与当前输入电平一致，否则标志位可能无法被清除。

实现步骤：

a) 设置所有端口都为数字端口，并将相应的外部按键中断端口设为输入口
b) 使能外部按键中断端口，配置相应的控制寄存器
c) 使能外部按键中断端口的内部弱上拉电阻
d) 对 PB 按键端口访问操作一次，使比较参考电平和当前输入电平一致
e) 清除相应的外部按键中断标志

f) 使能全局中断

g) 中断服务程序中，读取端口进行识别，清除中断标志

h) 主程序中判别按键去抖

2.6 内部Flash读写程序模块

功能说明：

HR7P169 内部有 4K 字节数据 FLASH 存储器， 分 4 页，每页 1K 字节，支持查表读写操作，地址范围为 4000H~47FFH。

查表读操作通过查表读指令将 FRA 所指向的地址单元中的一个字读入 ROMD 中。

数据 FLASH 的写以字为单位，写操作前必须先擦除所写单元所在的页，故写数据 FLASH 时包含三个基本操作：数据备份，页擦除和字编程。

芯片页擦除时间至少为 2ms，此期间芯片处于暂停状态。

建议用户在执行页擦除操作时关闭 WDT 内部 RC 时钟，避免执行擦除操作时因看门狗复位导致芯片误操作。

关于 Flash 存储器的可靠性操作方法详见《AN062_应用笔记_MCU 片内非易失性存储器操作》。

数据 FLASH 写实现步骤：

a) 用查表读指令将一页内容备份至数据存储空间

b) 修改数据存储空间需更新的数据

c) 执行页擦除（需按固定指令序列进行）

d) 通过设置寄存器 FRAL 和 FRAH 选择需要更新的地址，及设置寄存器 ROMDL 和 ROMDH 需要更新的数据

e) 将 ROMDL 和 ROMDH 中的内容按固定的写序列写入 FRA 所指定的页中地址

f) 重复 d)，e)直至完成整页编程

g) 用查表读指令进行写入区校验

2.7 UART通讯程序模块

功能说明：

异步收发器支持 8/9 位数据格式，支持高速低速模式。

波特率计算公式：

低速模式：BRGH=0, baud rate=Fosc/(64×(BRR<7:0>+1))

高速模式：BRGH=1, baud rate=Fosc/(16×(BRR<7:0>+1))

UART 通讯时，发送/接收中断标志通过写/读对应的 Buffer 硬件自动清零

实现步骤：

a) 设置所有端口都为数字口，并将 TX 方向设成输出，RX 方向设成输入

b) 设置波特率 9600，高速模式，系统时钟 2Mhz，9600=2M/(16×(BRR<7:0>+1))，得 BRR=12

c) 设置 8/9 位数据格式

d) 使能发送/接收器

e) 发送时，发送缓存为空时写入发送数据；接收时，通过查询 RXIF 中断标志位来判断是否收到完整的一帧数据

2.8 I2C从动模块

功能说明：

当 I2CSRIF 标志位置位，且 I2CRW 位为 0 时表示主机写从机，即从机接收数据。从机接收数据缓冲器满时，中断标志位 I2CRBIF 置位，读取一次接收数据缓冲器 I2CRB，I2CRBIF 标志位自动清零，其他使用注意事项请参考芯片数据手册。

当 I2CSRIF 标志位置位，且 I2CRW 位为 1 时表示主机读从机，即从机发送数据。I2CTB 发送数据缓冲器不满时，中断标志位 I2CTBIF 置位，写发送数据缓冲器 I2CTB，若数据缓冲器写满，I2CTBIF 标志位自动清零，若发出一个字节，数据缓冲器未满，则中断标志位 I2CTBIF 置位。

从机接收到 STOP 信号时，收发数据缓冲器不会自动清零，需置位 I2CRST 位来软件复位 I2C 模块。复位后，需重新初始化 I2C 模块。

关于 I2C 的传输共有两个例程即高速 I2C 传输和低速 I2C 传输用例，每个例程主机写一个数据给从机，再读回两个数据。从机将接收到的数据以原码和反码的形式发回给主机。

低速传输实现步骤：

a) 设置 SCL，SDA 所在的端口方向为输入

b) 设置从机的地址

c) 初始化从 I2C 模块，使能 I2C 模块及 I2C 通讯

d) 设置 I2C 中断使能寄存器 I2CIEC

e) 若收到地址匹配中断，清地址匹配中断标志， I2CRW 为 0 时写从机，使能 I2CRBIE，禁止 I2CTBIE；I2CRW 为 1 时读从机，禁止 I2CRBIE，使能 I2CTBIE

f) 若主机读从机，从机写 I2CTB 数据缓冲器发送数据

g) 若主机写从机，从机读 I2CRB 数据缓冲器接收数据

h) 若收到 STOP 中断， 清 STOP 中断标志， I2CRST 执行一次置位，复位 I2C 模块，清空收发数据缓冲器，禁止 I2C 中断使能寄存器，然后重新初始化 I2CIEC 和 I2CC 寄存器

高速传输实现步骤：

a) 设置编译器“项目->编译->Support interrupt vectors”为“true”；

“项目->编译-> IIC slave high speed mode”加入“true”

b) 设置 SDA 所在的端口方向为输入，SCL 引脚为输出，低电平

c) 设置从机的地址

d) 初始化从 I2C 模块，使能 I2C 模块及 I2C 通讯

e) 设置 I2C 中断使能寄存器 I2CIEC

f) 若收到地址匹配中断，清地址匹配中断标志， I2CRW 为 0 时写从机，使能 I2CRBIE，禁止 I2CTBIE；I2CRW 为 1 时读从机，禁止 I2CRBIE，使能 I2CTBIE ，释放 SCL 引脚

g) 若主机读从机，从机写 I2CTB 数据缓冲器发送数据，程序需要在写入 I2CTB 数据之后释放 SCL 引脚，主机在从机释放 SCL 引脚以后开始读数据

h) 若主机写从机，从机读 I2CRB 数据缓冲器接收数据

i) 若收到 STOP 中断， 清 STOP 中断标志， I2CRST 执行一次置位，复位 I2C 模块，清空收发数据缓冲器，禁止 I2C 中断使能寄存器，然后重新初始化 I2CIEC 和 I2CC 寄存器

2.9 模拟比较器及可编程脉冲发生器模块

功能说明：

比较器模块除实现通用比较功能外，还可实现过压，浪涌，过流和欠压等故障检测，故障检测输入比较器选择请参考数据手册。当故障事件发生时，比较器的输出会触发故障检测电路，当一定时间内故障没解除，则比较器延时输出会暂停或关闭。

PPG 输出是 C1OUT_DLY 经过比较器故障检测调整后的输出。

PPG 实现步骤：

a) 初始化系统和端口

b) 初始化 PPG 模块

c) 使能 PPG 模块

d) 初始化 T8P2 定时器