

文档编号: AN\_057

上海东软载波微电子有限公司

# 应用笔记

---

**HR7P275**

## 修订历史

版本	修改日期	更改概要
V1.0	2015-06-09	初版
V1.1	2015-07-28	统一修改公司名称、logo 及网址等
V1.2	2015-10-29	修改了 EEPROM 的操作步骤及注意事项
V1.3	2016-02-23	修改复位模块和低功耗模式的描述
V1.4	2016-06-03	增加了内部 EEPROM 操作需使能 HRC 注意事项
V1.5	2016-12-7	删除 T11/T12 预分频和后分配小节
V1.6	2017-09-20	1. 增加未使用 IO 端口的处理方法； 2. 删除 LVDIN 相关描述。 3. 补充说明 ICD 调试端口配置，建议选择 PC4/PC5 作为调试端口。
V1.7	2018-01-18	删除程序模块代码。
V1.8	2018-05-02	1. 新增读 Flash 操作说明 2. 新增 GIE 位和 GIEL 位处理说明
V1.9	2019-04-02	1. 变更 Logo 2. 修改 GIE 和 GIEL 位处理说明
V1.10	2019-12-23	1. 添加 1.10 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/>

版权所有©

### 上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不承担或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系

## 目 录

### 内容目录

<b>第 1 章</b>	<b>HR7P275 应用注意</b> .....	<b>4</b>
1.1	内部振荡器 .....	4
1.2	复位模块 .....	4
1.2.1	外部复位 .....	4
1.2.2	BOR 复位 .....	4
1.2.3	WDT 复位 .....	5
1.2.4	SBOREN/LRCEN 控制位 .....	5
1.3	低功耗模式 .....	5
1.4	中断处理 .....	5
1.4.1	外部按键中断 .....	5
1.4.2	中断标志的清除 .....	5
1.4.3	GIE 位和 GIEL 位处理 .....	5
1.5	未使用的 IO 端口处理 .....	6
1.6	PWM 输出极性选择 .....	6
1.7	FLASH IAP 注意事项 .....	6
1.8	内部 EEPROM 操作 .....	7
1.8.1	内部 EEPROM 写操作 .....	7
1.8.2	内部 EEPROM 读指令 .....	7
1.8.3	掉电时保存 EEPROM 数据正确的外围电路要求 .....	7
1.9	ICD 调试端口配置 .....	7
1.10	硬件乘法器 .....	8
<b>第 2 章</b>	<b>HR7P275 模块例程</b> .....	<b>9</b>
2.1	8 位定时器/计数器程序模块 (T10) .....	9
2.2	8 位定时器程序模块 (T11/T12) .....	9
2.3	16 位门控定时器程序模块 (T20/T21) .....	10
2.4	单边 PWM 程序模块 (T11/T12) .....	10
2.5	双边 PWM 程序模块 (T20/21) .....	11
2.6	外部中断程序模块 .....	12
2.7	ADC 程序模块 .....	12
2.8	外部按键中断程序模块 .....	12
2.9	通用异步收发器程序模块 .....	13
2.10	程序存储器访问程序模块 .....	13
2.10.1	读 FLASH .....	13
2.10.2	写 FLASH .....	13
2.10.3	擦除 FLASH .....	14
2.11	EEPROM 程序模块 .....	14
2.11.1	读 EEPROM .....	14
2.11.2	写 EEPROM 并校验 .....	15
2.11.1	烧录 EEPROM .....	15
2.12	LVD 程序模块 .....	15

## 第1章 HR7P275 应用注意

### 1.1 内部振荡器

HR7P275 芯片在出厂时已做好内部振荡器的校准，内部高速 RC 振荡器校准精度  $16\text{MHz} \pm 2\% @ 25^\circ\text{C}$ ， $3.0\text{V} \sim 5.5\text{V}$ ，内部低速 RC 振荡器校准精度  $32\text{KHz} \pm 10\% @ 25^\circ\text{C}$ ， $3.0\text{V} \sim 5.5\text{V}$ 。

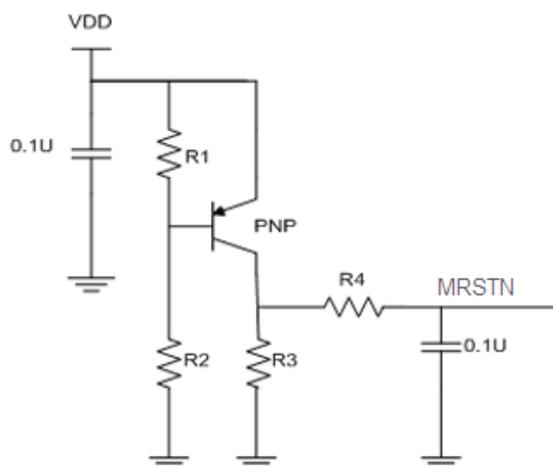
如果用户选择芯片内部振荡器作为系统时钟源，在芯片上电复位完成后，内部电路会自动把校准值加载到校准寄存器 HRCCAL 与 LRCCAL，完成校准操作，因此不需要通过软件进行赋值。

### 1.2 复位模块

#### 1.2.1 外部复位

用户应避免把 MRSTN 引脚直接连接到 VDD 上，可通过  $10\text{K}\Omega$  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  $< 0.7\text{V}$  时，PNP 晶体管截止，MRSTN 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  $[R1/(R1+R2)] \times VDD < 0.7\text{V}$  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  $R1=2\text{K}$ ， $R2=10\text{K}$ ， $R3=20\text{K}$ ， $R4=1\text{K}$ ，复位电压应满足：

$[2\text{K}/(2\text{K}+10\text{K})] \times VDD < 0.7\text{V}$ ，通过计算可以得到，当  $VDD < 4.2\text{V}$  时，PNP 晶体管处于截止状态，MRSTN 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 SBOREN/LRCEN控制位

---

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

#### 设置 SBOREN 和 LRCEN 程序示例:

```
BSS    PWEN, SBOREN    ;SBOREN=1
BSS    PWEN, LRCEN     ;LRCEN=1
```

## 1.3 低功耗模式

---

- 如果产品封装数小于 44，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，LRCEN 不能清零。

## 1.4 中断处理

---

### 1.4.1 外部按键中断

---

当用户使用了芯片的外部按键中断功能，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。

### 1.4.2 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。

### 1.4.3 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。

## 1.5 未使用的IO端口处理

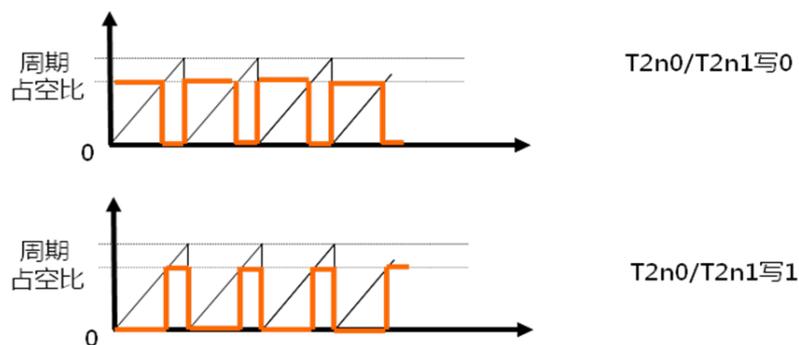
系统中未使用的 IO 管脚建议设置为输出固定电平并悬空，若设置为输入，须加上拉或下拉电阻接到电源或地。

## 1.6 PWM输出极性选择

T1n 和 T2n 作为 PWM 可输出正向或反向逻辑。

在使能 PWM 前，若向相应端口寄存器写“0”，则输出正向逻辑，即 PWM 脉宽为高电平；若向相应端口寄存器写“1”，则输出反向逻辑，即 PWM 脉宽为低电平。

单边 PWM 输出极性示意图如下：



## 1.7 FLASH IAP注意事项

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
<pre> u8 ReadFlash() {     GIE = 0;     FRAH = 0x40;     FRAL = 0x00;     __asm     {         TBR     }     FlashData.Byte[1] = ROMDH;     FlashData.Byte[0] = ROMDL;     GIE=1;     retron  ROMDL; }         </pre>	<pre> u8 ReadFlash() {     GIE = 0;     FRAH = 0x40;     FRAL = 0x00;     __asm     {         TBR     }     FlashData.Byte[1] = ROMDH;     FlashData.Byte[0] = ROMDL;     FRAH=0X47;     FRAL=0X00;     GIE=1;     return ROMDL; }         </pre>

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。

## 1.8 内部EEPROM操作

### 1.8.1 内部EEPROM写操作

内部 EPROM 存储器写操作时，用户程序必须对当前访问地址的所属页进行页数据校验，若读出校验不对，需再次将页数据写入并读出校验，直到校验正确为止。

为保证操作时序，读写 EEPROM 时需打开内部高速振荡器并待其稳定后方可操作。

### 1.8.2 内部EEPROM读指令

读 HR7P275 内部 EEPROM 时，仅支持 TBR/TBR1#指令，不支持 TBR#1/ TBR\_1 等指令。

### 1.8.3 掉电时保存EEPROM数据正确的外围电路要求

为保证系统在突然掉电情况下能成功保存一页数据，Vdd 引脚需要至少接 15uF 电容。

## 1.9 ICD调试端口配置

当选择 PA4/PA5 用于 ICD 调试端口时，如果软件配置为模拟端口，即 PAS[5:4]=11，将会导致无法进行 ICD 调试。

建议选择 PC4/PC5 用于 ICD 调试端口。

## 1.10 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 HR7P275 模块例程

### 2.1 8 位定时器/计数器程序模块 (T10)

#### 功能说明:

使用芯片的 T10 定时器模块, 在 PC1 端口输出一个周期为 2ms, 占空比 50% 的方波。

设定 T10 为定时器模式, 定时时间为 1ms。在 T10 定时中断服务程序中取反 PC1 端口电平, 实现 50% 占空比, 2ms 周期的方波输出。

芯片使用 8MHz 系统时钟, 则对应的 T10 定时器时钟源周期为 0.25us。将预分频器分配给 T10 定时器, 分频比为 1: 16。T10 的初始计数值的计算公式应为:

$$1\text{ms} / 0.25\text{us} = (255 - T10 + 1) \times 16, \text{ 计算得到 } T10 = 6(0x06)。$$

#### 实现步骤:

- a) 初始化系统和端口
- b) 初始化 T10 定时器
- c) 使能 T10IE, GIE 中断
- d) 中断服务程序。判断中断标志, 确定是 T10 中断后则清除 T10IF 标志位
- e) 执行 PC1 端口取反输出, 并重新向 T10 寄存器赋初值

### 2.2 8 位定时器程序模块 (T11/T12)

#### 功能说明:

使用芯片的 T11 (T12 方法类同, 以下不再赘述) 定时器模块, 在 PC1 端口输出一个高低电平各为 2ms 的方波 (周期 4ms, 占空比 50%)。

设定 T11 为定时器模式, 定时时间为 2ms。在 T11 定时中断服务程序中, 取反 PC1 端口输出电平, 实现 2ms 宽度高低电平的输出。

芯片使用 2MHz 系统时钟, 则对应的 T11 定时器时钟源周期为 1us。T11 的预分频采用 1: 16, 后分频为用 1: 1。T11 采用递增计数, 当 T11 的计数值达到周期寄存器 T11P 的设定值, 且满足后分频器的设定值后, 可以产生中断 (中断使能条件下)。T11P 寄存器值的计算公式应为:

$$2\text{ms} / 1\text{us} = (T11P+1) \times 16, \text{ 计算得到 } T11P = 124(0x7C)。$$

#### 实现步骤:

- a) 初始化系统和端口
- b) 初始化 T11 定时器并对 T11P 赋初值
- c) 使能 T11, PEIE, GIE 中断
- d) 中断服务程序。判断中断标志, 确定是 T11 中断后则清除 T11F 标志位

- e) 执行 PC1 端口取反输出

## 2.3 16 位门控定时器程序模块 (T20/T21)

### 功能说明:

使用芯片的 T20 (T21 方法类同, 以下不再赘述) 定时器, 在 PC1 端口输出一个高低电平各为 4ms 的方波 (周期 8ms, 占空比 50%)。

设定 T20 为定时器模式, 定时时间为 4ms。在 T20 定时中断服务程序中取反 PC1 端口电平, 实现 50% 占空比, 8ms 周期的方波输出。

芯片使用 8MHz 系统时钟, 则对应的 T20 定时器时钟源周期为 0.25us。T20 预分频比设定为 1:16。T20 的初始计数值的计算公式应为:

$4\text{ms} / 0.25\text{us} = (65535 - T20 + 1) \times 16$ , 计算得到  $T20 = 64536(0xFC18)$ , 则  $T20H = 0xFC$ ,  $T20L = 0x18$ 。

注意 16 位门控定时器重载初值时需要关闭定时器使能位。

### 实现步骤:

- a) 初始化系统和端口
- b) 初始化 T20 定时器
- c) 使能 T20 中断, PEIE 中断和 GIE 中断
- d) 中断服务程序。判断中断标志, 确定是 T20 中断后则清除 T20IF 标志位
- e) 执行 PC1 端口取反输出, 并重新向 T20 寄存器赋初值

## 2.4 单边 PWM 程序模块 (T11/T12)

### 功能说明:

初始化中当 PWM 输出端口寄存器写“0”时, PWM 输出高有效, 即 PWM 脉宽为高电平。当 PWM 输出端口寄存器写“1”时, PWM 输出低有效, 即 PWM 脉宽为低电平。

当  $T1nM<1:0> = 10$  时, T1n 模块为单边 PWM 模式。

使用 T11 的单边 PWM 模块在 PB1(T110)端口实现频率为 1.25KHz (周期为 800us), 占空比为 50% 的方波输出, PB1 初始化为 0, PWM 输出高有效。

使用 T12 的单边 PWM 模块在 PC0(T120)端口实现频率为 1.25KHz (周期为 800us), 占空比为 25% 的波形输出, PC0 初始化为 0, PWM 输出高有效。

芯片使用 2MHz 系统时钟, 则对应的 T1n 定时器时钟源周期为 1us。T1n 的预分频采用 1:4 (后分频器无效), T1n 周期寄存器 T1nP 值和占空比寄存器 T1nR 值的计算公式应为:

$$\text{周期} / 1\text{us} = (T1nP + 1) \times 4$$

$$\text{占空比} = (T1nR \times 2 + 1 + T1nREX) / ((T1nP + 1) \times 2)$$

PWM 模块对应的周期寄存器和占空比寄存器值分别为:

$800\mu\text{s} / 1\mu\text{s} = (T11P + 1) \times 4$ ，计算得到  $T11P = 199(0xC7)$ 。

$50\% = (T11R \times 2 + 1 + T11REX) / ((T11P+1) \times 2)$ ，计算  $T11R = 99(0x63)$ ， $T11REX = 1$ 。

$800\mu\text{s} / 1\mu\text{s} = (T12P + 1) \times 4$ ，计算得到  $T12P=199(0xC7)$ 。

$25\% = (T12R \times 2 + 1 + T12REX) / ((T12P+1) \times 2)$ ，计算  $T12R = 49(0x31)$ ， $T12REX = 1$ 。

#### 实现步骤:

- 初始化系统和端口，确定 PWM 输出极性
- 初始化 T1n 及其 PWM 模式，并对相应的寄存器赋值
- 配置 PWM 输出端口并使能 PWM 功能

## 2.5 双边PWM程序模块 (T20/21)

#### 功能说明:

初始化中当 PWM 输出端口寄存器写“0”时，PWM 输出高有效，即 PWM 脉宽为高电平。当 PWM 输出端口寄存器写“1”时，PWM 输出低有效，即 PWM 脉宽为低电平。

当  $T2nM<3:0> = 1101$  时，芯片为双边 PWM 模式。

使用芯片的 T20 双边 PWM 模块在 PB4(T200)端口实现频率为 1.25KHz（周期为 800us），占空比为 50%，而方波输出。

使用芯片的 T21 双边 PWM 模块在 PD6(T210)端口实现频率为 1.25KHz（周期为 800us），占空比为 25%的波形。

芯片使用 2MHz 系统时钟，则对应的 T2n 定时器时钟源周期为 1us。

T2n 的预分频采用 1: 4（后分频器无效），T2n 周期寄存器 T2nP 值和占空比寄存器 T2nR 值的计算公式应为：

$$\text{周期} / 1\mu\text{s} = T2nP \times 4 \times 2$$

$$\text{占空比} = (T2nR \times 2 + 1 + T2nREX) / ((T2nP + 1) \times 2)$$

PWM 模块对应的周期寄存器和占空比寄存器值分别为：

$$800\mu\text{s} / 1\mu\text{s} = T20P \times 4 \times 2$$
，计算得到  $T20P = 100(0x64)$ 。

$$50\% = (T20R \times 2 + T20REX) / (T20P \times 2)$$
，计算  $T20R = 50(0x32)$ ， $T20REX = 0$ 。

$$800\mu\text{s} / 1\mu\text{s} = T21P \times 4 \times 2$$
，计算得到  $T21P=100(0x64)$ 。

$$25\% = (T21R \times 2 + T21REX) / (T21P \times 2)$$
，计算  $T21R = 25(0x19)$ ， $T21REX = 0$ 。

#### 实现步骤:

- 初始化系统和端口
- 初始化 T2n 及其 PWM 模式，并对相应的寄存器赋值
- 配置 PWM 输出端口并使能 PWM 功能

## 2.6 外部中断程序模块

---

### 功能说明:

使用 HR7P275 的 PC0~PC3 端口外部中断功能，设定外部中断端口下降沿触发产生中断，在中断服务程序中取反 PB0~PB3 端口的输出电平。

### 实现步骤:

- a) 初始化系统和端口
- b) 设置外部中断下降沿触发
- c) 使能外部中断 GIE/PIEx
- d) 中断服务程序中，清除中断标志位，取反端口的输出电平

## 2.7 ADC程序模块

---

使用 HR7P275 芯片的 ADC 模块，实现对 AN0 通道模拟输入电压的数字量转换，其它通道关闭。

为提高 ADC 采样的精度，建议在 ADC 采样通道输入阻抗比较大的场合下（兆欧级），尽可能加长 ADC 采样时间，可以有效提高 ADC 采样精确性。

### 实现步骤:

- a) 初始化系统和端口
- b) 初始化 AD 转换器
- c) AD 转换执行及结果显示

## 2.8 外部按键中断程序模块

---

### 功能说明:

使用 HR7P275 的外部按键中断模块，打开外部按键端口的内部弱上拉功能，确认发生外部按键中断后取反 PB1 端口的输出电平。

### 实现步骤:

- a) 初始化系统和端口
- b) 使能外部按键中断端口的内部弱上拉电阻
- c) 使能外部按键中断和总中断
- d) 中断服务程序中按键消抖判别，清除外部按键中断标志

## 2.9 通用异步收发器程序模块

---

### 功能说明:

使用 HR7P275 芯片的通用异步收发器 (UART) 模块, 初始发送 1 个固定数据 0x55, 再将接收的数据回传给上位机。

HR7P275 支持 2 路通用异步收发器 (即 UART) 外设, 支持 8/9 位数据格式, 支持全双工模式, 可以兼容 RS-232/RS-442/RS-485 的通讯接口。

UART 模块波特率 (高速模式) 的计算公式为: 波特率 =  $F_{osc} / (16 \times (BRnR < 7:0 > + 1))$ 。

芯片使用 4MHz 系统时钟, 本程序使用 UART1 模块, 波特率设定为 1200bps, 计算得到 BR1R 初始值为 207 (0xCF)。

- a) 初始化系统和端口
- b) 初始化 UART 模块
- c) 使能 UART 接收中断
- d) 接收字节, 使能发送中断
- e) 回送已接收的字节, 禁止发送中断

## 2.10 程序存储器访问程序模块

---

### 功能说明:

HR7P275 FLASH 程序存储器为 16bits\*32K, 读写 FLASH 需事先知道地址的低 8 位和高 8 位, 将其分别放到 FRAL 和 FRAH 中。

当配置位 IAPEN 使能时, HR7P275 的程序存储器可以进行页擦除和单个字 (16 位) 的写入和读取的访问操作。如果某个地址已经被写入过数据, 则改写该地址的数据前, 需要预先对该地址所属页进行页擦除操作。

### 2.10.1 读FLASH

---

#### 实现步骤:

- a) 被读单元的地址低 8 位和高 8 位, 将其分别放到 FRAL 和 FRAH 中
- b) 执行 TBR 指令
- c) 取 ROMDL 和 ROMDH 中的数据

### 2.10.2 写FLASH

---

#### 实现步骤:

- a) 被写单元的地址低 8 位和高 8 位, 将其分别放到 FRAL 和 FRAH 中
- b) 选择 Flash 编程操作
- c) 关全局总中断

- d) 执行单写时序
- e) 启动编程操作

### 2.10.3 擦除FLASH

---

#### 实现步骤:

- a) 被擦单元的高 8 位地址赋给 FRAH
- b) 判断 FRAH 是否为有效的页地址
- c) 选择 Flash 擦除操作
- d) 关全局总中断
- e) 执行擦写时序
- f) 启动擦写操作

## 2.11 EEPROM程序模块

---

#### 功能说明:

HR7P275 内部集成 512 字节的 EEPROM 数据存储单元,映射到程序寻址空间 FE00H~FFFFH。512 字节的 EEPROM 又分为 32 页,每页 16 字节。

读写 EEPROM 需使能配置位 IAPEN。

为保证操作时序,读写 EEPROM 操作时 HRCEN 位需使能,并判断 HRCON 位是否置位。

无论是否有数据,EEPROM 写入前不需要进行 IAP 擦除操作。

EEPROM 的 IAP 写入和读操作都是以字节为单位。当 EEPROM 进行 IAP 写入操作时,内核和外设处于正常运行状态,当 EEPROM 写入完成后将置位 IAP 中断标志,用户可使能 IAP 中断使能位和全局中断使能位响应中断请求读 EEPROM。

在对 EEPROM 存储器编程操作时,用户程序必须对当前访问地址的所属页进行页数据校验,若读出校验不对,需再次将页数据写入并读出校验,直到校验正确为止。

编译器支持由程序填写初始化值烧录到芯片的 EEPROM 存储器,编译后会包含在 Hex 文件中,具体操作方法是在 iDesigner 软件项目工程中添加汇编程序 eepinit.asm。

### 2.11.1 读EEPROM

---

#### 实现步骤:

- a) 被读单元的地址低 8 位和高 8 位,将其分别放到 FRAL 和 FRAH 中
- b) 使能 HRC 并判断 HRCON 位是否为 1
- c) 执行 TBR 指令
- d) 执行 6 条 NOP 指令
- e) 取 ROMDL 中的数据
- f) 关闭 HRC

## 2.11.2 写EEPROM并校验

实现步骤:

- a) 读取待写入地址所在页的原整页数据
- b) 写更新数据到待写入的地址单元
- c) 读取更新后的整页数据并校验
- d) 如果校验失败，则对整页数据重新写入并校验
- e) 重复步骤 c 和 d，如果仍然失败，返回写入失败的 EEPROM 地址

## 2.11.1 烧录EEPROM

eepinit.asm 程序中采用“EEPROM”伪指令和“DW”字类型定义具体数据。

编译器默认从 EEPROM 空间的起始地址开始依次存储。

```
;初始化 EEPROM 数据  
EEPROM      ;伪指令  
tmp DW 0x1234,0x5678,0x9abc,0xdef0      ;DW 定义，高字节在高地址，低字节在低地址  
DW 0x1234,0x5678,0x9abc,0xdef0;  
END
```

## 2.12 LVD程序模块

功能说明:

HR7P275 支持低电压检测 LVD 功能，用来检测电源电压 VDD 的电压跌落。

应用中当检测到电源电压低于设定的阈值时程序将关键数据保存到片内的 EEPROM 中。

为提高 LVD 模块判断的准确性，需要打开 LVD 模块的滤波器，将 LVDC 寄存器的第 4 位置 1，在使能该位后需延时 300uS 再清 LVDIF 标志位。

实现步骤:

1. 初始化 LVD 模块和端口
2. 延时 300uS 以上
3. 清 LVDIF 中断标志位
4. 使能 LVDIE 中断
5. 使能全局中断 GIE
6. 中断服务程序中，清 LVDIF 中断标志位，保存数据到片内的 EEPROM