

文档编号: AN_069

上海东软载波微电子有限公司

应用笔记

HR7P169B

修订历史

版本	修订日期	修改概要
V1.0	2015-10-15	初版
V1.1	2016-06-03	增加模拟 IIC 主程序
V1.2	2016-07-07	增加 ADC 使用注意
V1.3	2016-12-6	<ol style="list-style-type: none"> 修正了 ADC 内部参考电压的使用说明。 增加了外部硬件电路对调试端口（ISP 端口）影响的说明。 增加通过配置字可选择 UART 串口单线分时通信。
V1.4	2017-3-06	<ol style="list-style-type: none"> 增加 PWM 周期和占空比连续调整的更新描述； 增加输入捕捉模块例程； 更新 PWM 模块例程； ADC 程序模块新增 AIN14 通道的 AD 检测。
V1.5	2017-6-09	<ol style="list-style-type: none"> 更新 Keyint 例程，输入口由 PB7 改为 PB4； 修改了 PWM 例程及描述； 删除了汇编代码。
V1.6	2017-09-11	<ol style="list-style-type: none"> 删除全部代码； 增加 ADC14 与 EPWM 模块例程。
V1.7	2018-01-17	<ol style="list-style-type: none"> 修正了 PWM 程序模块(T11)功能说明； 完善 I2C 传输相关应用说明。
V1.8	2018-05-02	<ol style="list-style-type: none"> 新增 IAP 擦写操作前关闭总中断使能位 GIE 说明 新增读 Flash 操作说明 新增 GIE 位和 GIEL 位处理说明 新增 ADC 转换建立过程说明及使用注意事项
V1.9	2018-09-05	<ol style="list-style-type: none"> 增加 BOR 电压设置描述
V1.10	2019-04-02	<ol style="list-style-type: none"> 变更 Logo 修改 GIE 和 GIEL 位处理说明
V1.11	2020-9-18	<ol style="list-style-type: none"> 增加硬件乘法器使用说明 增加 IAP 临界保护范围

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

第 1 章	HR7P169B 应用注意	4
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	RCEN 控制位	5
1.3	低功耗模式	5
1.4	唤醒方式操作	5
1.5	中断处理	6
1.5.1	外部按键中断	6
1.5.2	中断标志的清除	6
1.5.3	GIE 位和 GIEL 位处理	6
1.6	PWM 输出极性选择	7
1.7	PWM 周期和占空比的连续调整	7
1.8	I2C 从动模块操作	8
1.9	FLASH IAP 注意事项	9
1.10	烧录数据 FLASH	9
1.11	ADC 模块	9
1.12	UART 模块	10
1.13	芯片配置字 PWRTEB 设置	10
1.14	ISP 端口	10
第 2 章	HR7P169B 模块例程	11
2.1	定时器程序模块 (T8N)	11
2.2	输入捕捉模块 (T11)	11
2.3	PWM 程序模块 (T11)	12
2.4	增强型 EPWM 模块 (T12)	12
2.5	ADC 程序模块	13
2.6	ADC14 程序模块	13
2.7	外部按键中断程序模块	13
2.8	内部 Flash 读写程序模块	14
2.9	UART 通讯程序模块	15
2.10	I2C 从动模块	16
2.11	模拟比较器及可编程脉冲发生器模块	17
2.12	WDT 溢出唤醒 IDLE	17

第1章 HR7P169B应用注意

1.1 内部振荡器

HR7P169B 芯片在出厂时已做好内部振荡器的校准，校准精度 $16\text{MHz}\pm 1\% @ 25^\circ\text{C}, 5\text{V}$ 。

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，内部电路会自动完成校准操作。

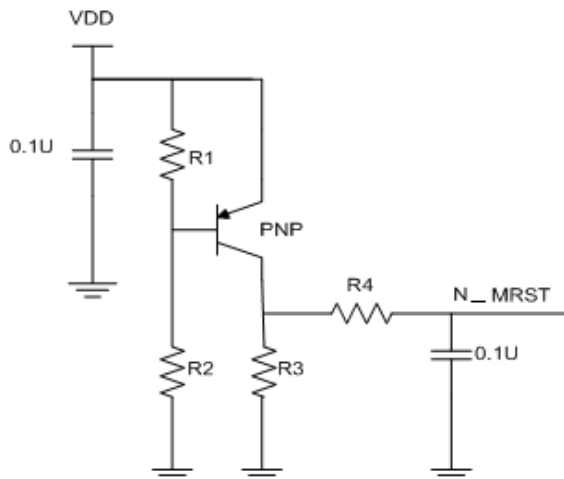
1.2 复位模块

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 $10\text{K}\Omega$ 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，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 $[R1/(R1+R2)] \times VDD < 0.7\text{V}$ 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，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。

BOR 固定为使能，建议客户设置 BORVS 在合理的电压点，以免芯片因外界干扰或电源波动而工作异常。如果客户系统芯片供电电源为 5V，推荐将 BOR 电压设置在 3.3V 或 4.0V。

1.2.3 WDT 复位

建议客户在设计产品时使能 WDT 功能。

1.2.4 RCEN 控制位

建议程序中不能关闭 RCEN，除非有特殊的操作（例如 IAP 或 IDLE），当操作完毕后，也必须使能 RCEN。

为了防止内部 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 RCEN 程序示例:

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

1.3 低功耗模式

- ◆ 如果产品封装引脚数小于 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，如果引出的 I/O 管脚未使用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，需注意清零相关中断标志位，并关闭中断使能位，以免误唤醒芯片。
- ◆ 唤醒 IDLE 后硬件自动使能 RCEN，需要清除唤醒中断标志位后才能关闭 RCEN；
- ◆ 如果 LVD 作为唤醒源，需要使能 RCEN；

1.5 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。

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。

1.6 中断处理

1.6.1 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。

1.6.2 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。

1.6.3 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：

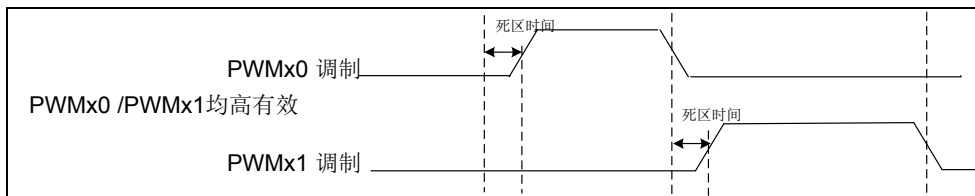
```
while(GIE == 1)
{
    GIE = 0;
}
while(GIEL == 1)           //仅使用了向量中断才需要此语句
{
    GIEL = 0;
}
.....
GIEL = 1;                  //仅使用了向量中断才需要此语句
GIE = 1;
```

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。

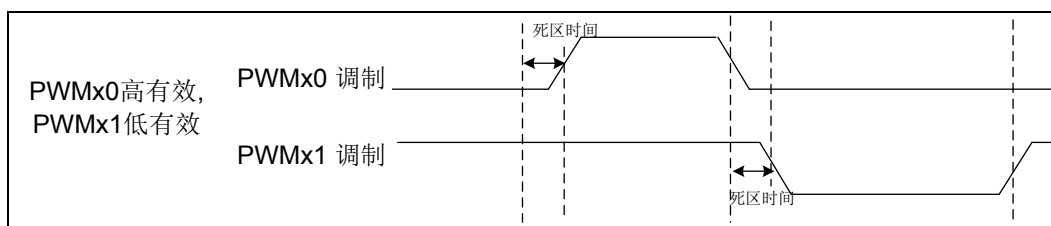
1.7 PWM输出极性选择

用户通过 PWM 配置寄存器 PWMxC 中的 PWMxM<1:0> 可以选择 PWM 输出极性。由于是互补输出，所以 PWMx0 和 PWMx1 输出的对应关系如下：

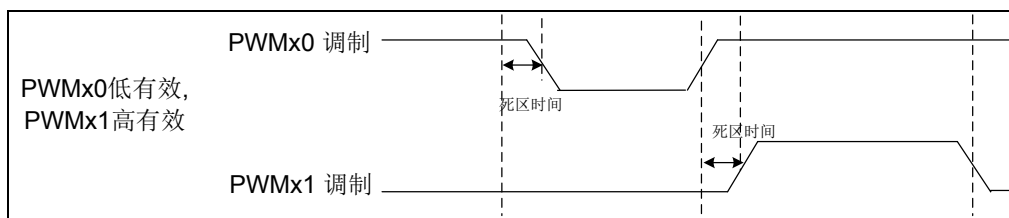
- ◆ PWMxM<1:0> = 00: PWMx0、PWMx1 高有效



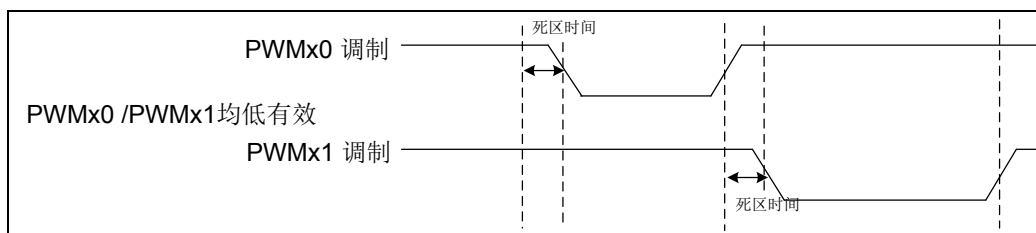
- ◆ PWMxM<1:0> = 01: PWMx0 高有效、PWMx1 低有效



- ◆ PWMxM<1:0> = 10: PWMx0 低有效、PWMx1 高有效



- ◆ PWMxM<1:0> = 11: PWMx0、PWMx1 低有效



1.8 PWM周期和占空比的连续调整

PWM 周期连续调整，必须在每次更新 PWM 周期寄存器后，设置 PWM 缓冲器即时更新使能位为 1，推荐的实现方式如下：

在 PWM 中断服务程序中，更新 PWM 周期寄存器，并设置对应的 PWM 缓冲器即时更新使能位 (PWM1XUD, PWM2XUD 或 PWM3XUD) 为 1，则当前 PWM 周期按照新的周期寄存器值进行调整（同时硬件清零 PWM 缓冲器即时更新使能位）。

PWM 占空比连续调整可以通过以下两种方式实现：

方式一：在主程序或 PWM 中断服务程序中，更新 PWM 精度寄存器，并设置 PWM 缓冲器即时更新使能位为 1，则当前 PWM 脉冲的占空比按照新的精度和周期寄存器值进行调整（同时硬件清零 PWM 缓冲器即时更新使能位）；

方式二：在主程序或 PWM 中断服务程序中，更新 PWM 精度寄存器，并设置 PWM 缓冲器即时更新使能位为 0，则当前 PWM 脉冲的占空比不变，下一个 PWM 脉冲的占空比按照新的精度寄存器值进行调整。

1.9 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，如果发送缓冲器未空，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。用户需要使用最新编译器，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）。

1.10 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。在使用 IAP 进行 flash 空间的读/擦/写期间，需要临时关闭中断，即在需要操作 FRAH/FRAL/ROMDH/ROMDL 寄存器进行一轮 IAP 读/擦/写之前临时关闭中断，直到一轮 IAP 读/擦/写结束后才可恢复中断。

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，“cseg”伪指令定义数据 FLASH 存储器的起始地址，随后以“DW”字类型定义具体数据。

初始化数据 FLASH 程序示例：

```
cseg 0x1E10
UserData DW 0x1234, 0x2345, 0x3456      ;数据内容，3 个字
END
```

1.12 ADC 模块

- ◆ 系统电源 VDD 纹波不宜过大，否则会影响 ADC 转换结果。
- ◆ AD 输入通道 AIN 管脚对地接 0.1uF 电容。
- ◆ 仿真调试时仿真端口通信可能会导致 VDD 纹波过大，从而影响 ADC 转换结果，需确认系统脱机运行结果是否正常，如果正常，则可忽略仿真调试过程中的 ADC 转换问题。
- ◆ ADC 模块设置寄存器 ADHSEN=1，ADVCMHS=1，选择 AD 高速转换模式，ADC 转换时钟，建议选择 512KHz~2MHz 之间。
- ◆ 支持内部参考电压作为 ADC 的模拟输入通道选择，即支持 ADCHS<3:0>=1110（内部参考电压 VREF）（参考 HR7P169B datasheet 6.6.6 节参考例程 2）。
- ◆ 当 ADC 使用外部参考电压时，参考电压不能低于 1.3V。
- ◆ 如果使用查询方式判断 ADC 转换是否完成，则调试时，不能在 ADC 启动代码及查询判断的循环代码段执行单步或设置断点。
- ◆ 在 VREF_EN，A/D 转换使能位 EN 使能后，ADC 需要先完成自身工作建立，才能得到正确的转换结果。上述使能控制信号使能后，延时 100us 以上，启动第一次 ADC 转换（TRIG=1），转换结束后，再延时 50us 以上，ADC 工作建立完成。

由于 ADC 建立过程中得到的转换结果与理论值偏差极大且不可预知，所以在应用程序中需要丢弃 VREF_EN，A/D 转换使能位 EN 使能后的第一次转换结果。

因每次 VREF_EN, A/D 转换使能位 EN 重新使能后, 均需要执行上述 ADC 工作建立过程, 所以应用中, 在芯片正常运行时不建议关闭上述使能控制信号, 保持为 1, 只在进入深睡眠模式前, 关闭 ADC。

1.13 UART 模块

- ◆ 支持 UART 串口单线分时通信, 通过编程器界面配置字 CFG_WORD<8>选择 RX/TX 是否复用为 PC1。

1.14 芯片配置字 PWRTEB 设置

芯片上电/低电压定时器使能位 PWRTEB 的设置:

- ◆ 当 N_MRST 管脚用于外部复位时, PWRTEB 可设置为使能或者禁止。
- ◆ 当 N_MRST 管脚用于数字输入时, PWRTEB 在芯片内部已固定为使能。

1.15 ISP 端口

芯片在仿真调试或编程时, ISPCK 和 ISPDAT 管脚要和外部电路隔开, 尤其是红外或 UART 串口等输入信号, 避免外部电路对其通信产生干扰, 而引起误操作。

第2章 HR7P169B 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 5ms, 占空比 50% 的方波。

设定 T8N 为定时器模式, 定时时间为 0.25ms。在 T8N 定时中断服务程序中取反 PA1 端口电平, 实现 50% 占空比, 0.5ms 周期的方波输出。

芯片使用 16MHz 系统时钟, 则对应的 T8N 定时器时钟源周期为 0.125us。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:

$$0.25\text{ms}/0.125\text{us} = (255 - \text{T8N} + 1) \times 8 \text{ (预分频比)}, \text{ 计算得到 } \text{T8N} = 6(0x06)。$$

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 输入捕捉模块 (T11)

功能说明:

采用 T11 捕捉信号高低脉冲的宽度, 数组连续捕捉 100 个脉宽后从头开始。

主频 16MHz, 设定捕捉时钟分频 1:32, 则时钟周期 2us, 按 12 位计数值计算, 可捕捉的时间范围为 2us~8ms。

实现步骤:

- a) 初始化系统和端口
- b) 初始化 T11 为输入捕捉模式
- c) 使能 T11IE, GIE 中断
- d) 中断服务程序, 判断中断标志
- e) 清定时器初值, 读取捕捉时间值存入数组
- f) 修改触发沿方向并清中断标志位

2.3 PWM程序模块 (T11)

功能说明:

使用芯片的 T11 扩展功能模块 PWM, 在 PA5 端口实现频率为 10KHz~28.5KHz (周期为 100us~35us), 占空比为 50%的方波输出。

芯片使用 16MHz 系统时钟, 则对应的 T11 定时器时钟源周期为 1/16MHz。T11 的预分频采用 1:8, T11 周期寄存器 T11P 初始值的计算公式应为:

PWM 周期 100us = (T11P+1) × (1/16MHz) × 8 (预分频比), 计算得到 T11P 初值 period=199(0xC7)。

PWM 脉宽 50us = T11RL × (1/16MHz) × 8 (预分频比), 计算得到 T11RL 初值 duty =100(0x64)。

PWM 占空比 50% = T11RL / (T11P+1)。

实现步骤:

- a) 初始化系统和端口
- b) 初始化 T11 及其扩展模块并对相应的寄存器赋值
- c) 使能 T11 的 PWM 模式
- d) 中断服务程序中更新周期和占空比, 并使能精度和周期缓冲器的实时更新使能位

2.4 增强型EPWM模块 (T12)

功能说明:

使用芯片的 T12 扩展功能模块 PWM, 在 PB0 口和 PB1 口实现周期为 50us, 占空比为 50%的方波, 其中死区时间 16us, 最终输出波形精度为 9us。

芯片使用 16MHz 系统时钟, 则对应的 T12 时钟源为 1/16MHz。T12 的预分频采用 1: 8, T12 周期寄存器 T12P 初始值计算公式应为:

PWM 周期 50us = (T12P+1) × (1/16MHz) × 8 (预分频比), 计算 T12P 初值为 99 (0x0063)。

PWM 周期 25us = (T12RL+1) × (1/16MHz) × 8 (预分频比), 计算 T12RL 初值为 52 (0x0032)。

PWM 占空比 50% = T12RL / (T12P+1)。

PWM 死区延时 16us = PDD2C (延时计数位) × 8 (预分频比) × (1/16MHz)。

实现步骤:

- a) 初始化 PB 端口
- b) 设置周期、精度初值以及死区计数值
- c) 设置 PWM 工作模式
- d) 使能 PWM 输出

2.5 ADC程序模块

功能说明:

使用 HR7P169B 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。

HR7P169B 最多支持 14 个输入通道, 最大转换分辨率为 12bit。本例程采用单通道 (通道 2) 进行 ADC 转换演示。

ADC 转换包括采样和转换两个过程。

实现步骤:

- a) 初始化系统和端口
- b) 使能 ADC 模块, 启动 A/D 转换, 并且使能 UART 通信
- c) 选择通道 2 进行 A/D 转换, 采样 18 次取中间 16 组数据, 取平均, 得到 A/D 转换值
- d) 利用 UART 通信进行数据传输

2.6 ADC14 程序模块

功能说明:

使用 HR7P169B 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。

HR7P169B 最多支持 14 个输入通道, 最大转换分辨率为 12bit。采用双通道 (通道 2、通道 14) 进行 ADC 转换演示, 其中通道 14 为内部参考 VREF。

ADC 转换包括采样和转换两个过程。

实现步骤:

- a) 初始化系统和端口
- b) 使能 ADC 模块, 启动 A/D 转换
- c) 进行第一次的 ADC 转化等待转化结束赋值于相应寄存器, 并舍弃第一次转换值
- d) 选择通道 2 进行采样, 得到 ADC 转换值
- e) 选择通道 14 进行采样转换, 得到 ADC 转换值
- f) 保存 A/D 转换结果

2.7 外部按键中断程序模块

功能说明:

使用 HR7P169B 的外部按键中断功能, 对 KIN3 (PB4 端口) 外部电平变化进行判别。在全局中断使能的条件下, KIN3 (PB4 端口) 的电平变化会产生外部按键中断。

请参考本文外部按键中断中断标志清除的内容, 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

2.8 内部Flash读写程序模块

功能说明:

HR7P169B 内部有 1K 字节数据 FLASH 存储器，分 2 页，每页 512 字节，支持查表读写操作，地址范围为 1E00H~1FFFH。

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。

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> 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; </pre>	<pre> u8 ReadFlash() { while(GIE == 1) { GIE = 0; } FRAH = 0x40; FRAL = 0x00; __asm { TBR } FlashData.Byte[1] = ROMDH; FlashData.Byte[0] = ROMDL; FRAH=0x47; </pre>

<pre> retrun ROMDL; } </pre>	<pre> 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。

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。

对数据 FLASH 存储器读/写操作前，芯片配置字中需使能 FREN 位，并选择好 DPAGES<1:0> FLASH 存储器页面选择位，本例中选择 page1~2。

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。

建议用户在执行擦写操作前关闭总中断使能位 GIE，避免执行擦写操作时因中断导致芯片误操作。

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

2.9 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， $9600=2M/(16 \times (BRR < 7:0 > + 1))$ ，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

2.10 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

2.11 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。

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 T12 定时器

2.12 WDT溢出唤醒IDLE

功能说明：

WDT 溢出唤醒 IDLE，需要在配置字中使能 WDT 功能。

实现步骤：

- a) 初始化系统和端口
- b) 初始化 WDT 模块，溢出时间约 2s
- c) WDT 溢出翻转 PA4 端口电平
- d) 进入 IDLE，等待 WDT 溢出唤醒