文档编号：AN_028

上海东软载波微电子有限公司

应用笔记

HR7P90H/91H/92H/90J/91J/92J

修订历史

版本	修改日期	更改概要
V1.0	2012-03-21	初版
V1.1	2012-07-19	增加 1.8 节芯片 Flash 页更新功能描述
V1.2	2013-06-18	增加 ADC 模块转换通道复用应用注意事项
V1.3	2014-12-30	重新编排章节，增补了模块例程 对所有应用注意事项进行了重新修订
V1.4	2015-07-28	统一修改公司名称、logo 及网址等
V1.5	2018-1-17	删除程序模块代码。
V1.6	2018-4-28	新增读 Flash 操作说明
V1.7	2019-03-26	1. 变更 Logo 2. 修改 GIE 和 GIEL 位处理说明
V1.8	2019-12-02	1. 添加 1.12 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 HR7P90H/91H/92H/90J/91J/92J应用注意

1.1 内部振荡器

HR7P90H/91H/92H/90J/91J/92J，芯片在出厂时已做好内部振荡器的校准，校准精度 16MHz ±2%@25℃（VDD =2.5V~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，芯片无需通过指令读取校准值，并赋值到校准寄存器 CALR 来做校准。芯片内部电路会自动把校准值加载到校准寄存器 CALR，完成校准操作。因此 CALR 寄存器不需要通过软件进行赋值。并且在用户程序中，需避免修改该寄存器，以免覆盖芯片默认的时钟校准值。

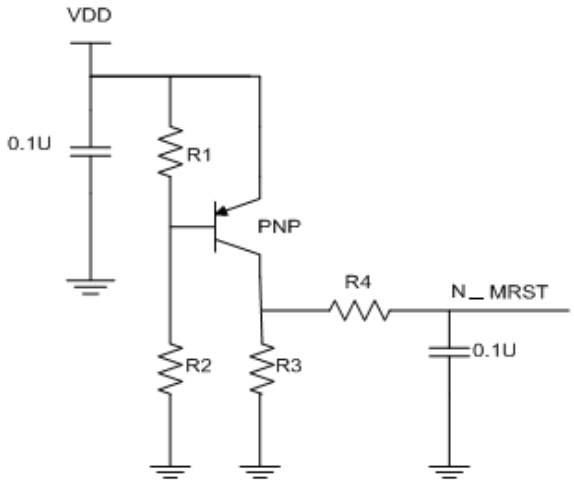
1.2 复位电路

1.2.1 内部复位

用户采用内部复位方式时，应避免把 N_MRST 引脚直接连接到 VDD 上，可以在 N_MRST 引脚上通过 10KΩ 电阻上拉到 VDD 的方式连接。

1.2.2 外部复位

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=2K，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。

1.3 振荡器复用功能

当用户使用芯片的内部振荡器模式时，芯片外部振荡器引脚可被作为普通 I/O 端口使用。在芯片上电直至复位稳定前这段时间，与芯片外部振荡器引脚 OSC2 复用的 PA6 端口会存在非预期的输出电平，可能引起系统输出的误动作。我们建议用户在方案设计中，不要将 PA6 端口作为关键信号的输出使用。

当用户使用芯片的内部振荡器模式，且 CLKOUT 不输出（INTOSCIO）时，芯片的 PA6 和 PA7 引脚可复用为 T16G 振荡器使用。在此模式下，用户需要将 T16G 模块有效（T16GEN=1），设置为 T16G 振荡器模式（T16GOSCEN=1），同时将相应的引脚 PA6 和 PA7 设置为输入模式。

当芯片被配置为使用外部振荡器时，与外部振荡器引脚复用的 I/O 端口（PA7,PA6）禁止使能内部弱上拉功能，即特殊功能寄存器 N_PAPU 的 bit7 和 bit6 位必须被置为 1，否则可能影响外部振荡器的振荡性能。

1.4 外部按键中断

1.4.1 外部按键中断的判别方法

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。

1.4.2 外部按键端口的内部弱上拉

当 PB4~PB7 作为按键输入时，在使能片内弱上拉功能后，如果在输入端口电平变化的速度过慢，会引起外部按键中断死锁现象。即当产生一次中断，程序读取 PB 端口值，然后清除中断标志后，PB 端口电平再次变化时，芯片无法产生中断，中断标志也不会被置位。

实际应用中，用户应避免外部按键中断和内部弱上拉功能的同时启用。当用户必须同时使用外部按键中断和端口内部弱上拉功能时，以下方法可以避免外部按键中断的死锁：

a）在进入电平变化中断后禁止 KIE(电平变化中断使能位)，但不清除中断标志 KIF。

b）在中断程序中设立“延时消抖标志”Tdelay_f（用户自定义标志）并将其置“1”，设立”延时寄存器”Tdelay_r（用户自定义变量）并将其清”0”，退出中断服务程序。

c）使用定时器中断对“Tdelay_r 计数，当延时时间>600us（按定时器时间而定）后将 Tdelay_f 清”0”，KIF 清”0”，使能 KIE（置”1”）

d）至此完成了一次电平变化中断。

如果按键输入端口的电平变化较快（变化时间≤600us），则建议用户禁止端口内部弱上拉功能。

注 1：参考章节 2.7 的模块例程，采用 T8P1 实现 200us 定时中断，对 Tdelay_r 计数

1.5 中断标志的清除

用户在使能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。

为避免中断的发生与中断标志清除操作冲突时，清除中断标志不成功，建议用户在进行中断标志清除操作后，对中断标志清除成功与否进行软件判断。如果操作不成功则再次进行中断标志清除操作，直到中断标志清除成功为止。用户也可以连续执行两次中断标志清除操作达到相同目的。

中断标志清除程序示例：

中断标志清除方法一：

```
……
BCC       INTC0,T8NIF     ；清除 T8N 中断标志
JBC       INTC0,T8NIF     ；判断 T8N 中断标志清除是否成功？
GOTO      $-2             ；T8N 中断标志清除不成功则重复进行
……
```

中断标志清除方法二：

```
……
BCC       INTC0,T8NIF     ；清除 T8N 中断标志
BCC       INTC0,T8NIF     ；再次清除 T8N 中断标志
……
```

1.6 GIE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 I/O端口的位操作

当执行以端口寄存器为目标的操作时，芯片实际执行读-修改-写过程，即先读取该组全部 I/O 端口的电平，修改后再写回端口寄存器。位操作也执行读-修改-写过程，因此对某一 I/O 端口的位操作有可能影响同组其它 I/O 端口的输出电平。

举例：当位操作指令的目标为一个 I/O 端口时，芯片先读取该组全部 I/O 端口的电平（不是读输出寄存器的电平），修改相应的位后再写入该组全部 I/O 端口的输出寄存器。当该组其它 I/O 端口被设定为输入状态或复用为外设功能时，这些 I/O 端口的电平状态与对应的输出寄存器值并不保持一致。当对某一 I/O 端口进行位操作时，因为芯片实际执行了同组全部 I/O 端口的读-修改-写操作，同组其它 I/O 的输出寄存器值是由读操作时当前 I/O 端口的电平决定的。

我们建议用户在对 I/O 端口的复用功能切换时，充分考虑当前 I/O 端口的输出寄存器值是否被误修改，并判断是否需要重新对这些 I/O 端口进行初始化赋值。

1.8 芯片Flash页更新

本芯片的 Flash 页更新功能被设计用于程序的在系统更新。如果用户把该功能用作非易失性存储器使用的话，需要注意在 Flash 的页写入和擦除阶段，芯片处于暂停状态，不会执行任何指令并响应任何中断。用户在把芯片应用于实时控制系统时，请慎重使用该功能。

同时，由于芯片的页擦除时间约为 22ms±8%，如果用户打开了 WDT 功能（WDT 使用独立的振荡器，仍然继续工作）并且 WDT 的溢出时间小于页擦除时间的话，由于芯片在 Flash 的页擦除阶段处于暂停状态，无法执行 CLRWDT 指令，会导致 WDT 计数器溢出产生复位，使芯片始终工作在复位状态。

由于 HR7P90H/91H/92H/90J/91J/92J 芯片的 WDT，在时钟源预分频比为 1:1 时，最大溢出时间约为 16ms，小于芯片页擦除时间，因此用户在使用页更新功能且打开 WDT 功能时，必须将预分频器分配给 WDT 寄存器，并确保使用预分频器后的 WDT 溢出时间大于芯片的页擦除时间。

用户可以使用芯片的其它定时器作为程序的时基，如果设计中只能使用 T8N 定时器作为程序时基的话，建议用户通过软件的方式延长定时时间。

1.9 芯片FLASH读操作

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;
	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;
    }
```

```
        }
        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。

1.10 芯片FLASH擦除操作对其它模块的影响及解决办法

芯片的页擦除时间约为 22ms±8%，在此过程中芯片处于暂停状态，不会执行任何指令和响应任何中断，芯片外围模块（例如 PWM、LCD 等）也停止工作。

如果用户在程序中使用芯片内置的 LCD 模块驱动液晶，同时用户在程序中也有对芯片 FLASH 进行擦除操作，此时可能会发生显示闪烁的现象。针对此问题，建议用户采用如下操作加以解决：

配置 LCD 控制寄存器 LCDC0 时，把帧频率时钟源配置成 LCD_RC/32，使用内部 RC 时钟作为 LCD 的帧频率时钟源。

1.11 T16G定时/计数器仿真注意事项

芯片的 T16GH 和 T16GL 寄存器同时受内置调试模块的控制，在芯片在线调试时，用户应避免在包含有 T16GH 和 T16GL 寄存器操作的指令处设置断点或单步运行。

1.12 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 HR7P90H/91H/92H/90J/91J/92J模块例程

2.1 T8N程序模块

功能说明：

使用芯片的 T8N 定时器模块，在 PC1 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 PC1 端口电平，实现 50%占空比，4ms 周期的方波输出。

芯片使用 4MHz 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 的初始计数值的计算公式应为：

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

实现步骤：

1、 初始化系统和端口
2、 初始化 T8N 定时器
3、 使能 T8NIE，GIE 中断
4、 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
5、 执行 PC1 端口取反输出，并重新向 T8N 寄存器赋初值

2.2 T8P1 程序模块

功能说明：

使用芯片的 T8P1(对 T8P2，T8P3 请参考芯片数据手册进行设定，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 T8P1 为定时器模式，定时时间为 2ms。在 T8P1 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 4MHz 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，后分频采用 1：5。T8P1 采用递增计数，当 T8P1 的计数值达到周期寄存器 T8P1P 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）。T8P1P 寄存器值的计算公式应为：

2ms/1us =（T8P1P+1）×4（预分频比）×5（后分频比），计算得到 T8P1P = 99(0x63)。

实现步骤：

1、 初始化系统和端口
2、 初始化 T8P1 定时器并对 T8P1P 赋初值
3、 使能 T8P1，PEIE，GIE 中断
4、 中断服务程序。判断中断标志，确定是 T8P1 中断后则清除 T8P1F 标志位
5、 执行 PC1 端口取反输出

}

2.3 T16G程序模块

功能说明：

使用芯片的 T16G 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 T16G 为定时器模式，定时时间为 4ms。在 T16G 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 4MHz 系统时钟，则对应的 T16G 定时器时钟源周期为 1us。T16G 预分频比设定为 1：4。T16G 的初始计数值的计算公式应为：

4ms/1us =（65535-T16G+1）×4（预分频比），计算得到 T16G = 64536(0xFC18)，则 T16GH = 0xFC，T16GL = 0x18。

实现步骤：

1、 初始化系统和端口
2、 初始化 T16G 定时器
3、 使能 T16G 中断，PEIE 中断和 GIE 中断
4、 中断服务程序。判断中断标志，确定是 T16G 中断后则清除 T16GIF 标志位
5、 执行 PC1 端口取反输出，并重新向 T16G 寄存器赋初值

2.4 T8P单边PWM程序模块（T8P1/2）

功能说明：

当 T8PxM<1:0> = 10 时，T8P1 模块为单边 PWM 模式。

使用 T8P1 的单边 PWM 模块在 PB0(T8P10)端口实现频率为 1.25KHz（周期为 800us），占空比为 50%的方波输出。

使用 T8P2 的单边 PWM 模块在 PB2(T8P20)端口实现频率为 1.25KHz（周期为 800us），占空比为 25%的波形输出

芯片使用 2MHz 系统时钟，则对应的 T8Px 定时器时钟源周期为 1us。T8Px 的预分频采用 1：4（后分频器无效），T8Px 周期寄存器 T8PxP 值和占空比寄存器 T8PxR 值的计算公式应为：

周期/1us =（T8PxP+1）×（预分频比）

占空比 = （T8PxR+1）/ (T8PxP+1）

PWM 模块对应的周期寄存器和占空比寄存器值分别为：

800us/1us = (T8P1P+1)×4（预分频比），计算得到 T8P1P=199(0xc7)。

50% = (T8P1R +1)/(T8P1P+1)，计算得到 T8P1R=99(0x63)。

800us/1us = (T8P2P+1)×4（预分频比），计算得到 T8P2P=199(0xc7)。

25% = (T8P2R +1)/(T8P2P+1)，计算得到 T8P2R=49(0x31)。

实现步骤：

1、 初始化系统和端口
2、 初始化 T8Px 及其 PWM 模式，并对相应的寄存器赋值
3、 配置 PWM 输出端口并使能 PWM 功能

2.5 T8P双边PWM程序模块（T8P1/2）

功能说明：

当 T8PxM<1:0> = 11 时，芯片为双边 PWM 模式，双边模式下 PWM 的实际输出周期为设定周期的两倍。

使用芯片的 T8P1 双边 PWM 模块在 PB0(T8P10)端口实现频率为 1.25KHz（周期为 800us），占空比为 50%,而方波输出，实际输出周期为 1.6ms。

使用芯片的 T8P2 双边 PWM 模块在 PB2(T8P20)端口实现频率为 1.25KHz（周期为 800us），占空比为 25%的波形，实际输出周期为 1.6ms

芯片使用 2MHz 系统时钟，则对应的 T8Px 定时器时钟源周期为 1us。

T8Px 的预分频采用 1：4（后分频器无效），T8Px 周期寄存器 T8PxP 值和占空比寄存器 T8PxR 值的计算公式应为：

周期/1us =（T8PxP+1）×（预分频比）

占空比 = （T8PxR +1）/ (T8PxP+1）

PWM 模块对应的周期寄存器和占空比寄存器值分别为：

800us/1us = (T8P1P+1)×4（预分频比），计算得到 T8P1P=199(0xc7)。

50% = (T8P1R +1)/(T8P1P+1)，计算得到 T8P1R=99(0x63)。

800us/1us = (T8P2P+1)×4（预分频比），计算得到 T8P2P=199(0xc7)。

25% = (T8P2R +1)/(T8P2P+1)，计算得到 T8P1R =49(0x31)

实现步骤：

1、 初始化系统和端口
2、 初始化 T8Px 及其 PWM 模式，并对相应的寄存器赋值
3、 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 HR7P90H_91H_92H_90J_91J_92J 的 PB0~PB7 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 PC0~PC7 端口的输出电平。

实现步骤：

1、 初始化系统和端口;
2、 设置外部中断下降沿触发;
3、 使能外部中断 GIE//PIEx;
4、 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

功能说明：

使用 HR7P90H/91H/92H/90J/91J/92J 芯片的 ADC 模块，采用查询方式实现对 AN0 通道模拟输入电压的数字量转换，采取中值滤波方式得到最终的 A/D 转换值。

实现步骤：

1、 初始化系统和端口
2、 初始化 ADC 模块并对相应的寄存器赋值
3、 使能 ADC 模块，启动 A/D 转换
4、 查询并等待 A/D 转换结束
5、 采样数据排序
6、 求 A/D 采样数据平均值
7、 返回 A/D 采样值

2.8 外部按键中断程序模块

功能说明：

使用 HR90H/91H/92H/90J/91J/92J 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，参照 1.4.2 章节的说明，确认发生外部按键中断后取反 PC0 端口的输出电平。

实现步骤：

1、 初始化系统和端口
2、 使能外部按键中断端口的内部弱上拉电阻
3、 初始化延时定时器 T8P1
4、 使能外部按键中断和总中断
5、 中断服务程序中，按照 1.4.2 章节的方式判别并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：

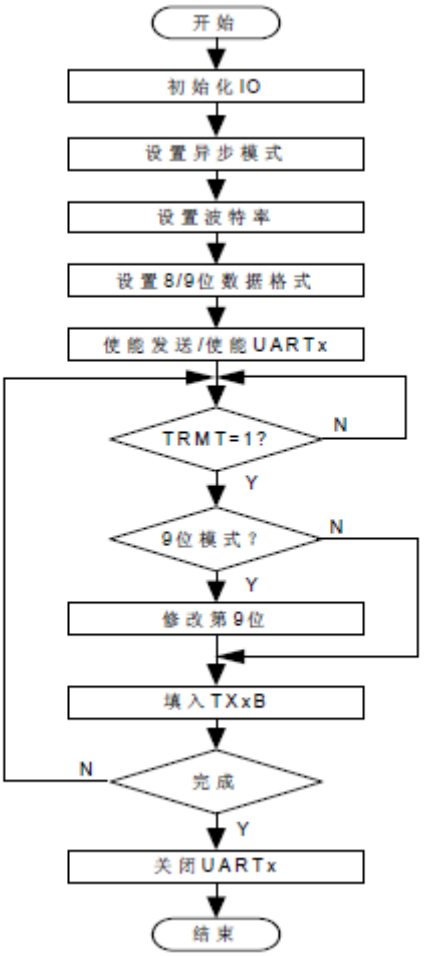
使用 HR7P90H/91H/92H/90J/91J/92J 芯片的通用异步收发器（UART）模块，定时发送 4 字节固定数据，数据接收成功则在 PC5 端口取反输出。

HR7P90H/91H/92H/90J/91J/92J 支持 3 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。

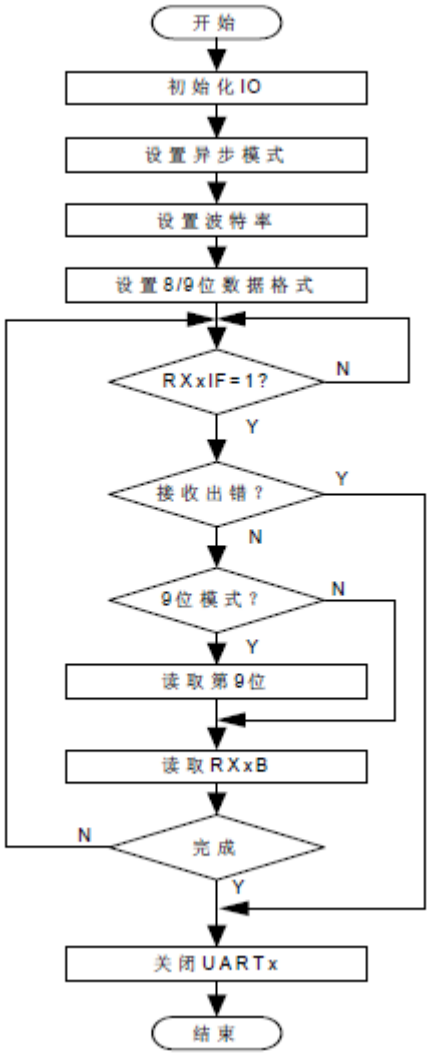
UART 模块波特率（高速模式）的计算公式为：波特率 = Fosc / (16×(BRR<7:0> + 1))。

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）。

异步发送实现步骤：



异步接收器实现流程：



2.10 程序存储器访问程序模块

功能说明：

2.10.1 读FLASH

功能说明：

HR7P90H/91H/92H/90J/91J/92J FLASH 程序存储器为 16bits*16K~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，然后执行 TBR 指令，对应地址的 16 位数据便存放到 ROMDL 和 ROMDH 中。

实现步骤：

1、 被读单元的地址低 8 位和高 8 位，将其分别放到 FRAL 和 FRAH 中
2、 执行 TBR 指令
3、 取 ROMDL 和 ROMDH 中的数据

2.10.2 写FLASH

功能说明：

以更新第 62 页数据为例给出 Demo，其中 DATA_BUF 为备份 62 页 FLASH 所用的数组（Uint08 data_buf[512]，定义时要用绝对地址方法如：section4 Uint08 data_buf[512] @0X0200;）

实现步骤：

1、 备份页至 RAM
2、 更新 RAM 数据
3、 执行页擦除
4、 执行写行