

文档编号: AN109

上海东软载波微电子有限公司

---

# 应用笔记

## ES7P1391

## 修订历史

版本	修订日期	修改概要
V1.0	2019-1-28	初版发布
V1.1	2019-04-03	<ol style="list-style-type: none"> <li>1. 变更 Logo</li> <li>2. 修改 GIE 和 GIEL 位处理说明</li> </ol>
V1.2	2019-12-02	<ol style="list-style-type: none"> <li>1. 增加 LDO 作为 ADC 输入通道的例程；</li> <li>2. SPI 主机以 IDIF 作为 SPI 发送完成的标志；</li> <li>3. 添加 1.10 硬件乘法器注意事项。</li> </ol>

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

<b>第 1 章</b>	<b>ES7P1391 应用注意</b> .....	<b>4</b>
1.1	BOR 复位.....	4
1.2	低功耗模式.....	4
1.2.1	唤醒方式操作.....	4
1.3	UART 模块.....	4
1.4	I2C 从动模块操作.....	4
1.5	T31 模块 PWM 控制模式.....	4
1.6	烧录数据 FLASH.....	5
1.7	ADC 模块.....	5
1.8	芯片配置字 PWRTEB 设置.....	5
1.9	ISP 端口.....	5
1.10	硬件乘法器.....	5
<b>第 2 章</b>	<b>ES7P1391 模块例程</b> .....	<b>7</b>
2.1	8 位定时/计数器 (T8N).....	7
2.1.1	定时模式.....	7
2.1.2	计数模式.....	7
2.2	16 位多功能定时器 (T21).....	8
2.2.1	定时模式.....	8
2.2.2	捕捉模式.....	8
2.2.3	比较模式.....	8
2.2.4	多精度 PWM 模式.....	9
2.3	16 位多功能定时/计数器 (T31).....	9
2.3.1	定时模式.....	9
2.3.2	捕捉器模式.....	10
2.3.3	比较器模式.....	11
2.3.4	PWM 模式.....	11
2.4	ADC 程序模块.....	12
2.4.1	普通测量.....	12
2.4.2	使用精确的内部 LDO.....	12
2.5	外部中断程序模块.....	13
2.6	向量中断.....	13
2.7	GIE 位和 GIEL 位处理.....	14
2.8	内部 Flash 读写模块.....	14
2.9	IAP 读芯片唯一识别码.....	15
2.10	UART 通讯程序模块.....	15
2.11	I2C 从动模块.....	16
2.12	SPI 主控器.....	16
2.13	SPI 从动器.....	17
2.14	WDT 看门狗.....	17
2.15	SLEEP 低功耗.....	17

## 第1章 ES7P1391 应用注意

### 1.1 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低于所设定的电压范围，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，可以通过配置字选择 BOR 复位电压点，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。

### 1.2 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。

#### 1.2.1 唤醒方式操作

- ◆ 在进入 IDLE 之前，需注意清零相关中断标志位，并关闭相应中断使能位，以免误唤醒芯片。
- ◆ IDLE 唤醒后，芯片硬件会自动使能 RCEN，需要用户软件清唤醒中断标志位后才能关闭 RCEN（软件对 RCEN 的关闭只在 IDLE 状态下有效，即通过软件关闭 WDT 时，在芯片运行状态下 WDT 仍保持工作，只有在进入 IDLE 状态后才会被关闭）。

### 1.3 UART模块

为了保证通讯的正确性，波特率最高不能超过 57600bps。

在仿真调试状态下，如果 TX 正在发送数据，而这时用户暂停调试，TX 口依然会不停输出数据。用户在调试时要注意这方面的影响。

### 1.4 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。

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 置位的方式复位 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。

### 1.5 T31 模块PWM控制模式

T31 模块，在 ICD 调试模式下，需软件固定设置 T31C0H <6> (HTOE OFF) 为 1，调试暂停时，PWM 输出控制由 PWEN<0>寄存器决定。当设置 PWEN<0> (HALT\_PWM) 为 1 时，在调试暂停时，计数器停止计数，会关断 PWM 输出，此时 PWM 输出端口的输入/输出状态，由相应的端口方向寄存器 PxT 决定，如果 PxT 为输入，相应的端口为高阻态，如果 PxT 为输出，相应端口状态由端口寄存器决定输出 0 还是 1。当设置 PWEN<0> (HALT\_PWM) 为 0 时，在调试暂停时，

计数器继续计数，仍保持 PWM 输出。

## 1.6 烧录数据FLASH

编译器支持由用户程序填写初始化值到芯片的 DataFlash 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 DataFlash 中。

具体操作方法是，在 iDesigner 软件项目工程中，添加汇编程序。程序中采用“eeprom”伪指令定义 DataFlash 存储器的起始地址，随后用“DW”指令定义具体数据。

### 初始化 DataFlash 程序示例：

```
eeprom 0x1F00  
  
UserData DW 0x1234, 0x2345, 0x3456 ;数据内容，3 个字  
  
END
```

- ◆ 如果烧录程序前在 ESBurner\编辑(编辑缓冲区数据)\ 中勾选“除去 DataFlash”，烧录时 DataFlash 数据将保留。默认为全部区域，即烧录时 DataFlash 区数据也会被擦除。(在 iDesigner 中由配置字 WRDataFlash 决定。配置为 Enable，烧录时 DataFlash 区数据也会被擦除，否则保留 DataFlash。)

## 1.7 ADC模块

- ◆ 建议 AD 输入通道管脚对地接 0.1uF 滤波电容。
- ◆ 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。
- ◆ 当 ADC 使用外部参考电压时，参考电压不能低于 3.0V@VDD=5.0V，2.5V@ VDD=3V。
- ◆ 建议常开启 ADC 模块的使能位 ADEN=1，以免频繁的禁止和使能导致 ADC 转换有误。

## 1.8 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 MRSTN 管脚用于外部复位时，PWRTEB 可设置为使能或者禁止。
- ◆ 当 MRSTN 管脚用于数字 IO 时，PWRTEB 在芯片内部已固定为使能。

## 1.9 ISP端口

芯片在仿真调试或编程时，ISPCK 和 ISPDA 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。

## 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 ES7P1391 模块例程

### 2.1 8 位定时/计数器 (T8N)

#### 2.1.1 定时模式

---

使用芯片的 T8N 定时器模块，在 PA4 端口输出一个周期为 2s，占空比 50%的方波。

设定 T8N 为定时器模式，计数器溢出时间为 4ms。定义一个变量每次溢出中断时自增一次，大于等于 250 次该变量清零，同时翻转 PA4。从而实现 PA4 每  $4\text{ms} \times 250 = 1.0\text{s}$  翻转一次

芯片使用内部 16MHz 做为系统时钟，则对应的 T8N 定时器时钟源频率为  $F_{osc}/2=8\text{MHz}$ 。将预分频器分频比设为 1: 128。T8N 寄存器初始值的计算公式应为：

$$(255+1-T8N) / (F_{osc}/2/128) = 4\text{ms}, \text{ 计算得到计数器初值 } T8N = 6.$$

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) 使能定时器；
- e) 中断服务程序中判断中断标志，确定是 T8N 中断后则清除 T8NIF 标志位；
- f) 执行 PA4 端口输出取反，并重新向 T8N 寄存器赋初值。

#### 2.1.2 计数模式

---

- a) 初始化系统和端口；
- b) 初始化 T8N 定时器为同步计数模式；
- c) 使能 T8NIE，GIE 中断；
- d) 使能定时器；
- e) T8NCKI 端口输入波形，T8N 计数器溢出中断；
- f) 中断服务程序中判断中断标志，确定是 T8N 中断后则清除 T8NIF 标志位；
- g) 执行 PA5 端口输出取反，清中断标志。

## 2.2 16 位多功能定时器 (T21)

### 2.2.1 定时模式

---

#### 功能说明:

使用 T21 定时功能, 计数器时钟频率  $F_{osc}=16\text{MHz}$ , 预分频 1:16, 后分频次数 16, 周期寄存器值 62500, 则定时器溢出时间  $t=16*16*62500/16\text{MHz} = 1\text{s}$ , 定时器溢出后在中断服务程序中翻转 PA4 端口电平。

#### 实现步骤:

- a) 初始化系统和端口;
- b) 初始化 T21 为定时器模式;
- c) 使能 T21VIE(溢出中断), GIE 中断;
- d) 使能定时器, 等待计数器溢出中断;
- e) 中断服务程序中判断中断标志, PA4 端口输出取反, 清中断标志。

### 2.2.2 捕捉模式

---

#### 功能说明:

使用 T21 捕捉功能, T21\_CH0(PA0)上升沿捕捉, T21\_CH1(PA7)下降沿捕捉, T21\_CH2(PB7)双沿捕捉。

#### 实现步骤:

- a) 初始化系统和端口;
- b) 初始化 T21 为捕捉模式, T21\_CH0 上升沿捕捉、T21\_CH1 下降沿捕捉、T21\_CH2 双沿捕捉;
- c) 使能端口复用功能, 选择 PA0 作为 T21\_CH0, PA7 作为 T21\_CH1, PB7 作为 T21\_CH2;
- d) 使能 T21\_CH0、T21\_CH1、T21\_CH2 捕捉中断, 使能 GIE 中断;
- e) 使能定时器, 等待捕捉中断, 进中断服务程序;
- f) 中断服务程序中判断中断标志, 保存捕捉时捕捉寄存器的值;
- g) 清中断标志位。

### 2.2.3 比较模式

---

#### 功能说明:

使用 T21 比较功能, 当通过计数器与比较器匹配时在中断服务程序中配置  $T21M<3:0>=1000$  或 1001 来实现下一次匹配时, 对输出端口的电平取反操作, 从而在 T21\_CH2(PB7)上输出方波。



**实现步骤:**

- a) 初始化系统和端口;
- b) 初始化 T21 为比较模式, 使能端口复用功能, 设置比较器值;
- c) 使能比较器 2 中断, 使能 GIE 中断;
- d) 使能定时器, 等待比较中断;
- e) 比较中断服务程序中清零计数器, 取反 T21M<0>用于在下一次计数器匹配时取反端口电平;
- f) 清中断标志位。

## 2.2.4 多精度PWM模式

**功能说明:**

使用 T21 的多精度 PWM 功能, PA0 和 PA7 输出 PWM 互补波形, PB7 输出占空比连续变化的 PWM 波形。

**实现步骤:**

- a) 初始化系统和端口;
- b) 初始化 T21 为多精度 PWM 模式, 使能端口复用功能, 设置周期寄存器和精度寄存器值;
- c) 使能定时器 PWM 周期中断, 使能 GIE 中断;
- d) 使能定时器, PA0、PA7 输出 PWM 互补波形, 等待周期中断;
- e) 周期中断服务程序中连续改变精度寄存器 2 的值, 用以连续改变 T21\_CH2 占空比;
- f) 清中断标志位。

## 2.3 16 位多功能定时/计数器 (T31)

### 2.3.1 定时模式

**功能说明(使用内部高速时钟):**

使用 T31 的定时功能, 定时器溢出后在中断服务程序中翻转 PA4 端口电平。

使用内部时钟计数, 预分频比 T31PRS=0x0F(1:16), 重载值 T31CNTLD=62500(0xF424), 后分频比 T31POS = 0x0F(1:16), 则溢出时间:

$$T = T31CNTLD / (16MHz / (T31POS+1) / (T31PRS+1)) = 1s;$$

**实现步骤:**

- a) 初始化所有端口为数字口, 输出低电平;
- b) 初始化 T31 为定时模式, 初始化周期、预分频比、后分频比;
- c) 打开 T31 溢出中断, 打开 T31 总中断, 打开全局中断;
- d) 使能 T31 计数器, 等待溢出中断;

- e) 中断服务程序中对 PA4 端口取反操作，清溢出标志位，清 T31 总中断标志位。

#### 功能说明(使用外部高速时钟):

使用 T31 的定时功能，定时器溢出后在中断服务程序中翻转 PA4 端口电平。

使用外部时钟计数，预分频比 T31PRS=0x00(1:1)，重装载值 T31CNTLD=1000(0x03E8)，后分频比 T31POS = 0x00(1:1)，依据选择的不同外部时钟模式，在对应管脚输入 1kHz 方波。

#### 实现步骤:

- 初始化所有端口为数字口，输出低电平；
- 初始化 T31 为定时器外部时钟计数模式，初始化周期、预分频比、后分频比；
- 打开 T31 溢出中断，打开 T31 总中断，打开全局中断；
- 使能 T31 计数器，等待溢出中断；
- 中断服务程序中对 PA4 端口取反操作，清溢出标志位，清 T31 总中断标志位。

### 2.3.2 捕捉器模式

#### 功能说明(测量脉冲周期):

使用 T31 的捕捉功能，成功捕捉后在中断服务程序中翻转 PA4 端口电平，捕捉时会自动将计数器的值复制到捕捉寄存器 T31CHnR (捕捉/比较寄存器)。

使用内部时钟计数，预分频比 T31PRS=0x0F(1:16)。假如捕捉波形为 1kHz，依据配置，双沿捕捉/上升沿捕捉/下降沿捕捉 PA4 输出方波的周期为 1ms/2ms/ 2ms。

#### 实现步骤:

- 初始化所有端口为数字口，输出低电平；
- 设初始化 T31 为捕捉模式，初始化内部计数器预分频比、选择输入通道、捕捉边沿；
- 打开 T31 捕捉中断，打开 T31 总中断，打开全局中断；
- 使能 T31 计数器，等待捕捉中断；
- 中断服务程序中对 PA4 端口取反操作，清溢出标志位，清 T31 总中断标志位。

#### 功能说明(测量脉冲周期和占空比):

使用 T31 的捕捉功能，将芯片的同一个端口 T31\_CH1 作为两路捕捉通道的输入，两路捕捉器选择不同的边沿捕捉(一个选择上升沿，另一个选择下降沿)，且选择在上升沿(也可是下降沿)捕捉时复位计数器。由于发生捕捉时会自动将计数器的值复制到 T31CHnR (捕捉/比较寄存器)，这样对于同一输入而言，T31CH1R 和 T31CH2R 的值就代表着 PWM 的周期和占空比。

建议：设置合理的预分频，防止在计数器溢出时仍未捕捉到一个完整的 pwm 脉冲。即最好保证 pwm 周期小于计数器溢出周期。

#### 实现步骤:

- 初始化所有端口为数字输出低电平；
- 设初始化 T31 为捕捉模式，初始化内部计数器分频比、选择输入通道、捕捉边沿；
- 打开 T31 捕捉 1 中断，打开 T31 总中断，打开全局中断；

- d) 使能 T31 计数器，等待捕捉中断；
- e) 在执行读取捕捉器的语句之前打断点，查看周期和占空比；
- f) 清除捕捉中断标志位，清除 T31 总中断标志位。

### 2.3.3 比较器模式

#### 功能说明(方波输出):

使用 T31 的比较功能，计数器匹配 T31CH1R(捕捉/比较寄存器 1)时翻转，T31 通道输出端口翻转，在 T31\_ETR(PB4)高电平时输出低电平。

#### 实现步骤:

- a). 初始化所有端口为数字口，输出低电平，T31\_ETR(PB4)设为输入下拉；
- b). 初始化 T31 为比较模式，初始化内部计数器分频比、周期等；
- c). 使能定时器输出功能(CHOE=1)，使能 T31 计数器；
- d). 示波器观察 T31\_CH1N(PB2)波形，PB4 高电平时输出低电平。

#### 功能说明(单脉冲输出):

使用 T31 的比较功能，在 T31\_CH2(PB6)上升沿时刻延时 1ms 后，在 T31\_CH1(PB5)产生一个持续 0.5ms 的宽脉冲。

#### 实现步骤:

- a). 初始化所有端口为数字口，输出低电平，T31\_CH1/T31\_CH2 设为输入；
- b). 初始化 T31 为单脉冲模式，初始化内部计数器分频比、选择输入通道、触发边沿、触发条件产生时的延时时间、脉冲宽度；
- c). 使能定时器输出功能(CHOE=1)，等待 T31\_CH2 上升沿触发计数器开始计数；
- d). 输入上升沿(或下降沿，依据配置)，示波器查看波形。

### 2.3.4 PWM模式

#### 功能说明(带死区互补输出):

使用 T31 的 PWM 功能，T31\_CH1(PB5)/ T31\_CH1N(PB2)输出带死区互补 pwm，死区时间 31.75us。PB4 高电平时输出低电平，PB0 高电平刹车。可配置调试模式暂停状态下是否输出 PWM。

#### 实现步骤:

- a). 初始化所有端口为数字口，输出低电平，T31\_ETR(PB4)和 T31\_BRK(PB0)设为输入下拉；
- b). 初始化 T31 为 PWM 模式，初始化内部计数器分频比、周期、占空比等参数；
- c). 配置调试模式暂停状态下是否仍输出 PWM；
- d). 使能定时器输出功能(CHOE=1)，使能 T31 计数器；
- e). 示波器观察 T31\_CH1(PB5)/ T31\_CH1N(PB2)波形，PB4 高电平时输出低电平，PB0 高电

平刹车。

### 功能说明(PWM 周期连续变化):

使用 T31 的 PWM 功能, T31\_CH1(PB5)/ T31\_CH2(PB2)输出频率连续变化的 PWM 波形。频率范围 100kHz~205kHz, 每 10ms 以 1kHz 步进。可配置调试模式暂停状态下是否输出 PWM。全速运行时周期连续变化, 如果调试暂停时仍输出 PWM, 则输出 PWM 频率固定为暂停前的状态。

实现步骤:

- a). 初始化所有端口为数字口, 输出低电平, 初始化 T8N, 用于定时 10ms;
- b). 初始化 T31 为 PWM 模式, 初始化内部计数器分频比、周期、占空比等;
- c). 使能周期寄存器(T31CNTLD)缓冲功能, 写入该寄存器的值在下次定时器溢出时生效;
- d). 配置调试模式暂停状态下是否仍输出 PWM;
- e). 使能定时器输出功能(CHOE=1), 使能 T8N, 使能 T31 计数器, 10ms 时间到更新周期寄存器;
- f). 示波器观察 T31\_CH1(PB5)/ T31\_CH2(PB2)波形。

## 2.4 ADC程序模块

### 2.4.1 普通测量

功能说明:

使用 ES7P1391 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。

ES7P1391 最多支持 8+1 个输入通道, 最大转换分辨率为 12bit。本例程采用宏定义的方式选择 VDD 作为基准测量通道 5 的电压, 或选择用外部 VREFP 作为基准测量通道 5 的电压。通过 UART 的方式发送到上位机, UART 波特率 9600bps。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 5 进行 A/D 转换, 得到 A/D 转换值;
- d) 或选择 VREFP 作为基准测量通道 5 的电压, 得到 A/D 转换值;
- e) 利用 UART 通信进行数据传输。

### 2.4.2 使用精确的内部LDO

功能说明:

ES7P1391 的 ADC 含有一个以内部 LDO 为输入的通道, 其电压已知且比较稳定, 具体请参考内部 LDO 输出电压特性表。

本例程首先采用内部 LDO 作为 ADC 输入通道, 然后推算出此时的 VDD 电压, 最后再以 VDD 为基准测量通道 5 的电压。通过 UART 的方式发送到上位机, UART 波特率 9600bps。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 LDO 通道作为输入进行 A/D 转换, 得到 A/D 转换值, 然后再计算出此时的 VDD 电压作为后续的参考电压;
- d) 选择通道 5 进行 A/D 转换, 以 VDD 为基准得到 A/D 转换值;
- e) 利用 UART 通信进行数据传输。

## 2.5 外部中断程序模块

**功能说明:**

使用 ES7P1391 的外部中断功能, 对 PINT1 (PA3 端口) 外部电平变化进行判别。PINT1 (PA3 端口) 的下降沿会产生外部中断标志, 若使能外部中断和全局中断, 则会进入中断函数。

**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) 清除相应的外部按键中断标志;
- e) 使能全局中断;
- f) 中断服务程序中读取端口进行识别, 清除中断标志;
- g) 主程序中判别按键去抖。

## 2.6 向量中断

采用四种模块作为向量中断的例程, 分别是 I2C(高优先级)、UART(高优先级)、T21(低优先级)、T8N(低优先级), 优先级依次递减, 高优先级可以嵌套低优先级。I2C 接收中断、UART 发送接收中断、T21 定时 1s 中断、T8N 定时中断。

注意: 使用向量中断模式前, 需在配置字界面选择->向量中断模式, 且需右键工程->属性->编译->Support interrupt vectors->True, 设置完成后切记保存设置;

**实现步骤:**

- a) 设置所有端口都为数字端口, 选择向量中断模式, 配置中断向量表 INTV;
- b) 分别初始化 T8N 模块、T21 模块、I2C 模块、UART 模块;
- c) 根据中断地址配置向量中断函数, 注意选择的向量地址要正确;
- d) 分别打开低优先级中断、全局中断, 分别使能各个模块, 等待中断。

## 2.7 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。

## 2.8 内部Flash读写模块

### 操作说明：

ES7P1391 内部有 7936 Word FLASH 存储器和 256 Word 数据存储器，共 8K Word。

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，单个地址编程时间至少为 20us，此期间芯片处于暂停状态。

对 DataFLASH 存储器读/写操作前，需配置字使能 FREN，并选择好 FLASH 存储器页面选择位。

建议用户设置合理的 WDT 溢出时间，并合理清狗。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》。

### 功能说明：

例程中 Flash 操作分为 IAP 程序区操作和 IAP 数据区操作，IAP 程序区操作采用宏定义指定起始地址的方式，向指定区域的地址写入 0x55(高字节)、0xAA(低字节)。IAP 数据区同理，两者主要的不同点在于程序区 IAP 操作可能要擦除多个 Flash 区域。

**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, 满足地址取反逻辑并设置寄存器 ROMDL 和 ROMDH 确定需要更新的数据;
- 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 IAP读芯片唯一识别码

---

**功能说明:**

通过 IAP 方式读 INFO1 区芯片唯一识别码(UID)。将读取到的 UID 保存至数组中。

**实现步骤:**

- a) 清零 RAM, IO 初始化;
- b) 选择 IAP 操作 INFO1 区, 读 0093H~0097H 地址数据;
- c) 将读出内容保存到数组中;

## 2.10 UART通讯程序模块

---

**功能说明:**

程序中采用宏定义的方式选择使用查询方式发送、接收或选择使用中断方式发送、接收。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:  $BRGH=0, \text{Baudrate}=\text{Fosc}/(64 \times (\text{BRR} \langle 7:0 \rangle + 1))$

高速模式:  $BRGH=1, \text{Baudrate}=\text{Fosc}/(16 \times (\text{BRR} \langle 7:0 \rangle + 1))$

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 9600bps, 低速模式, 系统时钟 16MHz,  $9600\text{bps} = 16\text{M}/(64 \times (\text{BRR} \langle 7:0 \rangle + 1))$ , 得  $\text{BRR}=25$ ;
- c) 设置 8 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，通过接收中断接收完整的一帧数据。

## 2.11 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，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。

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 的例程使用 I2C 模拟 64 字节的 eeprom。主机写数据：写命令+片内写起始地址+n 个字节数据；主机读数据：写命令+片内读起始地址+n 个读命令。

### 数据传输实现步骤：

- 设置 SCL，SDA 所在的端口为输入口；
- 设置从机的地址；
- 初始化从 I2C 模块，使能 I2C 模块及 I2C 通讯；
- 设置 I2C 中断使能寄存器 I2CIEC；
- 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；
- 若主机读从机，从机写 I2CTB 数据缓冲器发送数据；
- 若主机写从机，从机读 I2CRB 数据缓冲器接收数据；
- 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。

## 2.12 SPI主控器

### 功能说明：

使用芯片 SPI 模块，配置为主机模式，实现与 SPI 从机的通信。

发送数据采用查询，等待发送移位寄存器空(IDIF=1)时写入下一帧数据，发送数据如果不关心接收的数据，可将接收关闭(REN=0)。接收数据可采用中断或者轮询，在接收前，首先要发送一帧数据，然后等待接收缓冲器非空，将数据读出。

### 实现步骤：

- 配置 SPI 使用到的端口为数字口，NSS,SCK,MOSI 为输出，MISO 为输入；
- 选择 SPI 的波特率，Fosc/16；
- 选择 SPI 的通讯数据格式；
- 选择主机模式；



- e) 使能接收，如果采用中断接收，还要打开接收中断，并打开总中断；
- f) 使能 SPI，将数据放到数据发送寄存器，SPI 自动将数据发出，等待数据发送寄存器空时写入下一帧要发送的数据，发送数据的同时会接收到数据，如果不关心，在发送时可将接收关闭。接收时首先要发送一帧数据，然后检查是否接收到数据，如果接收到，读出数据。

## 2.13 SPI从动器

### 功能说明：

使用芯片 SPI 模块，配置为从机模式，实现与 SPI 主机的通信。

从机将接收到的数据保存在数组中，并放到发送数据寄存器中，当主机读数时返回主机。

### 实现步骤：

- a) 配置 SPI 使用到的端口为数字口，NSS,SCK,MOSI 为输入，MISO 为输出；
- b) 选择数据传输格式；
- c) 选择从机模式；
- d) 打开接收及接收中断，以及总中断和 SPI 中断；
- e) 使能 SPI；
- f) 将接收到的数据保存到数组中，并放到发送数据寄存器中。

## 2.14 WDT看门狗

### 功能说明：

使用芯片 WDT 模块，实现硬件看门狗功能。

芯片上电首先 PA4 快速翻转一小段时间，然后进入正常工作模式，PA4 每 0.5s 翻转一次。当程序中未在设置的时间内清狗，芯片复位，程序重新运行

### 实现步骤：

- a) 在配置字中使能看门狗；
- b) 打开预分频器，选择分频系数为 128；
- c) 初始化中快速翻转 PA4 端口电平
- d) 程序主循环中清狗，PA4 每 0.5s 翻转一次。

## 2.15 SLEEP低功耗

### 功能说明：

芯片可配置进入三种睡眠状态，深度睡眠，浅睡眠和睡眠快速唤醒模式。在进入睡眠前，PA4 管脚翻转十次，然后进入睡眠状态，管脚停止翻转，保持在睡眠前的状态。可选择是否在睡眠前关闭 WDT，使用外部管脚中断唤醒。

### 实现步骤：

- a) 选择睡眠模式；
- b) 配置外部中断管脚；
- c) 使用指令 IDLE 进入睡眠状态；

d) 外部管脚中断唤醒。