

目录

目录.....	1
第一部分：固件库概述.....	2
一、前言.....	2
二、固件库架构规则.....	2
三、固件库函数调用方法.....	4
第二部分：固件库函数说明.....	6
一、GPIO 固件库函数.....	6
二、EXTI 固件库函数.....	8
三、PWM 固件库函数.....	9
四、ADC 固件库函数.....	12
五、TIMER0 固件库函数.....	15
六、TIMER1 固件库函数.....	17
七、TIMER2 固件库函数.....	20
八、UART0 固件库函数.....	22
九、SSI 固件库函数.....	25
十、IAP 固件库函数.....	30
十一、WDT 固件库函数.....	31
十二、BTM 固件库函数.....	32
十三、PWR 固件库函数.....	33
十四、CHKSUM 固件库函数.....	34
十五、OPTION 固件库函数.....	35
规格更改记录.....	38

第一部分：固件库概述

一、前言

赛元 92 系列 MCU 的固件库，提供了一套 92 系列通用的 API 接口，以及 API 接口的应用实例，从而实现赛元 92 系列 MCU 的程序标准化。

固件库中，每个外设驱动都由一组函数组成，这组函数覆盖了该外设所有功能。每个器件的开发都由一个通用 API 驱动，API 对该驱动程序的结构，函数和参数名称都进行了标准化。

赛元 92 系列 MCU 固件库的使用，使用户无需深入掌握外设配置细节，也可轻松应用每个外设，大大减少用户程序编写时间，降低开发成本。同时，由于 92 系列 MCU 的 API 一致，只需替换相关库文件，无需修改程序细节，便能实现程序项目在不同 MCU 之间的快速替换。

二、固件库架构规则

1、缩写含义

ADC	模数转换器
BTM	低频时钟定时器
CHKSUM	校验和
DDIC	显示驱动集成电路（LED/LCD 驱动）
EXTI	外部中断事件控制器
GPIO	通用输入输出
IAP	在应用编程（EEPROM/FLASH 编程）
MDU	乘除法器单元
PWR	电源/功耗控制
PWM	脉宽调制
SSI	三选一串行接口（SPI/TWI/UART 三选一）
TIM	定时器
TOUCH	触控电路
UART	通用异步收发器
WDT	看门狗
Option	Customer Option 区域（用户烧写设置）

2、命名规则

固件库函数遵从以下命名规则：

PPP 表示任一外设缩写，例如：ADC。

系统、源文件和头文件命名都以“sc92fxxxx_”作为开头，例如：sc92f8003_adc.c, sc92f8003_conf.h。

常量仅被应用于一个文件的，定义于该文件中；被应用于多个文件的，在对应头文件中定义。所有常量都由英文字母大写书写。

外设函数的命名以该外设的缩写加下划线为开头。每个单词的第一个字母都由英文字母大写书写，例如：UART0_SendData8。在函数名中，只允许存在一个下划线，用以分隔外设缩写和函数名的其他部分。

名为 PPP_Init 的函数，功能为初始化外设 PPP，例如：TIM0_Init。

名为 PPP_DeInit 的函数，功能为复位外设 PPP 的所有寄存器至缺省值，例如 TIM1_DeInit。

名为 PPP_Cmd 的函数，功能为使能外设 PPP，例如：PWM_Cmd。

名为 PPP_ITConfig 的函数，功能为使能或失能来自外设 PPP 的中断源，并设置中断优先级，例如 TIM2_ITConfig。

名为 PPP_GetFlagStatus 的函数，功能为检查外设 PPP 某标志位被设置与否，例如：SSI_GetFlagStatus。

名为 PPP_ClearFlag 的函数，功能为清除外设 PPP 的标志位，例如 EXT11_ClearFlag。

3、编码规则

(1) 变量类型

```
typedef signed char      int8_t;
typedef signed short    int16_t;
typedef signed long     int32_t;
typedef unsigned char   uint8_t;
typedef unsigned short  uint16_t;
typedef unsigned long   uint32_t;
typedef int32_t         s32;
typedef int16_t         s16;
typedef int8_t          s8;
typedef uint32_t        u32;
typedef uint16_t        u16;
typedef uint8_t         u8;
#define __I              volatile const
#define __O              volatile
#define __IO             volatile
```

(2) 布尔型

布尔型被定义如下：

```
typedef enum {FALSE = 0, TRUE = !FALSE} bool;
```

(3) 状态类型

标志位状态、中断状态、位状态被定义如下：

```
typedef enum {RESET = 0, SET = !RESET} FlagStatus, ITStatus, BitStatus;
```

功能状态被定义如下：

```
typedef enum {DISABLE = 0, ENABLE = !DISABLE} FunctionalState;
```

错误状态被定义如下：

```
typedef enum {ERROR = 0, SUCCESS = !ERROR} ErrorStatus;
```

优先级状态被定义如下：

```
typedef enum {LOW = 0, HIGH = !LOW} PriorityStatus;
```

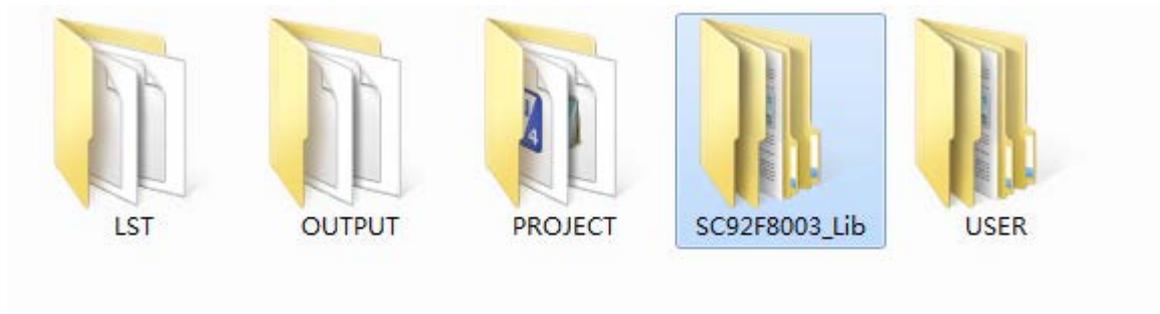
4、固件库函数文件结构（以 SC92F8003 芯片为例）

sc92f8003_yyy.c	外设的驱动源文件，包含了该外设的通用 API。8003 代表该系列 IC，yyy 代表外设缩写。该文件包含头文件 sc92f8003_yyy.h
sc92f8003_ _yyy.h	外设的驱动头文件，包含 API 相关定义。该文件包含头文件 sc92f8003.h
sc92f8003.h	固件库的通用头文件，包含整个固件库通用的类型说明及定义等。该文件包含头文件 SC92F8003_C.H
SC92F8003_C.H	赛元提供的 IC 标准头文件，包含了 IC 的寄存器定义等
sc92f8003_it.c/h	中断服务函数源/头文件，包含了 IC 所以中断服务函数，中断处理在这里执行

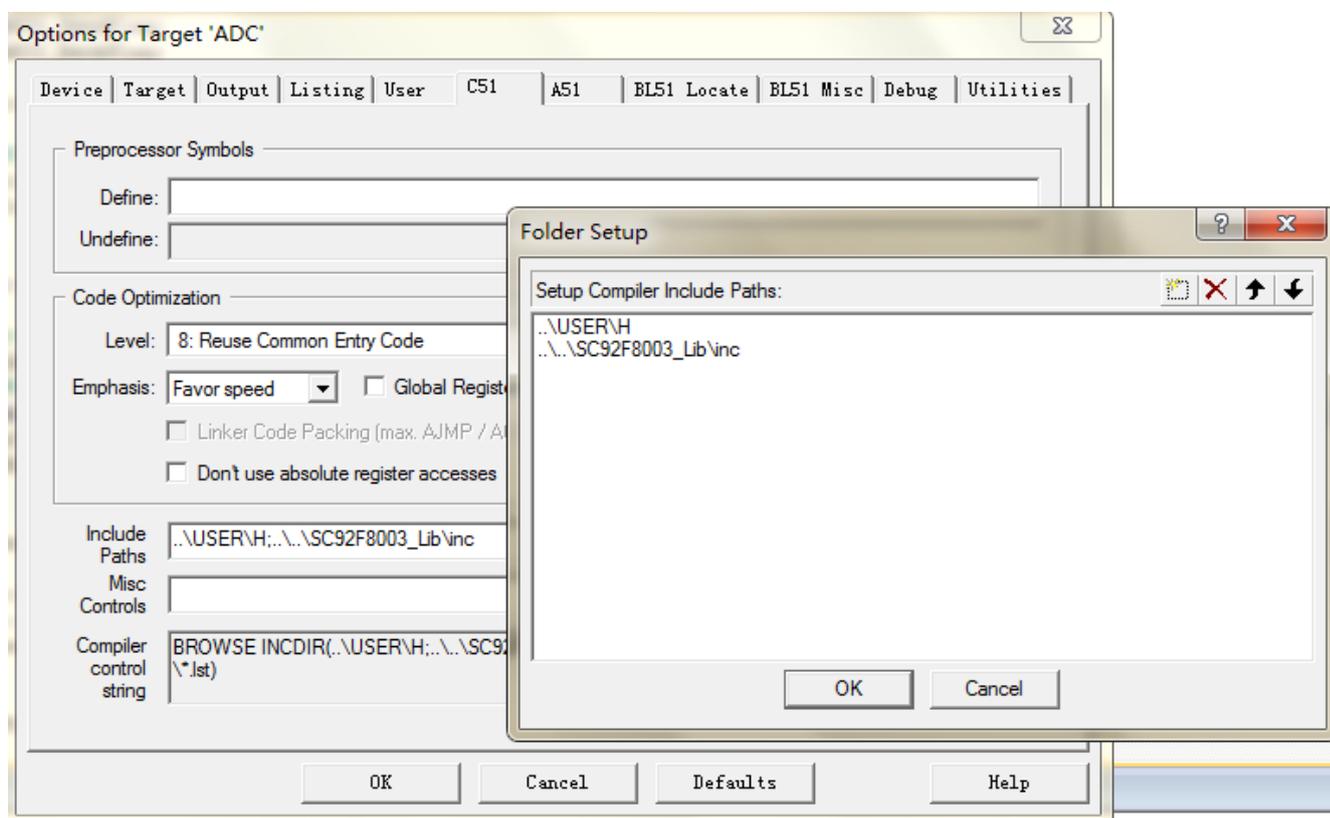
三、固件库函数调用方法

固件库函数调用方法参考 SC92F8003_LIB_EXAMPLE 中的应用示例进行说明。

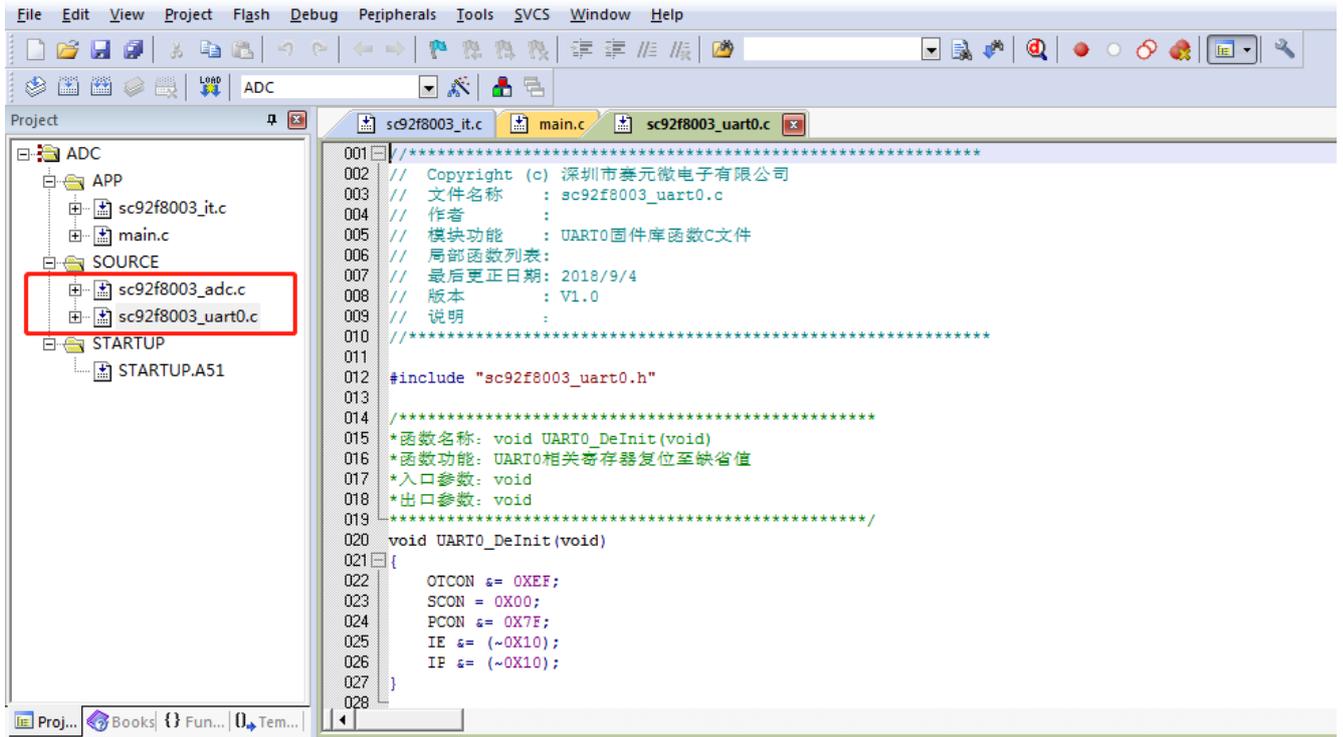
- 1、在工程相关的文件夹添加固件库 SC92F8003_Lib。



- 2、在工程内添加固件库 SC92F8003_Lib 中头文件的路径，路径指向 inc 文件夹。

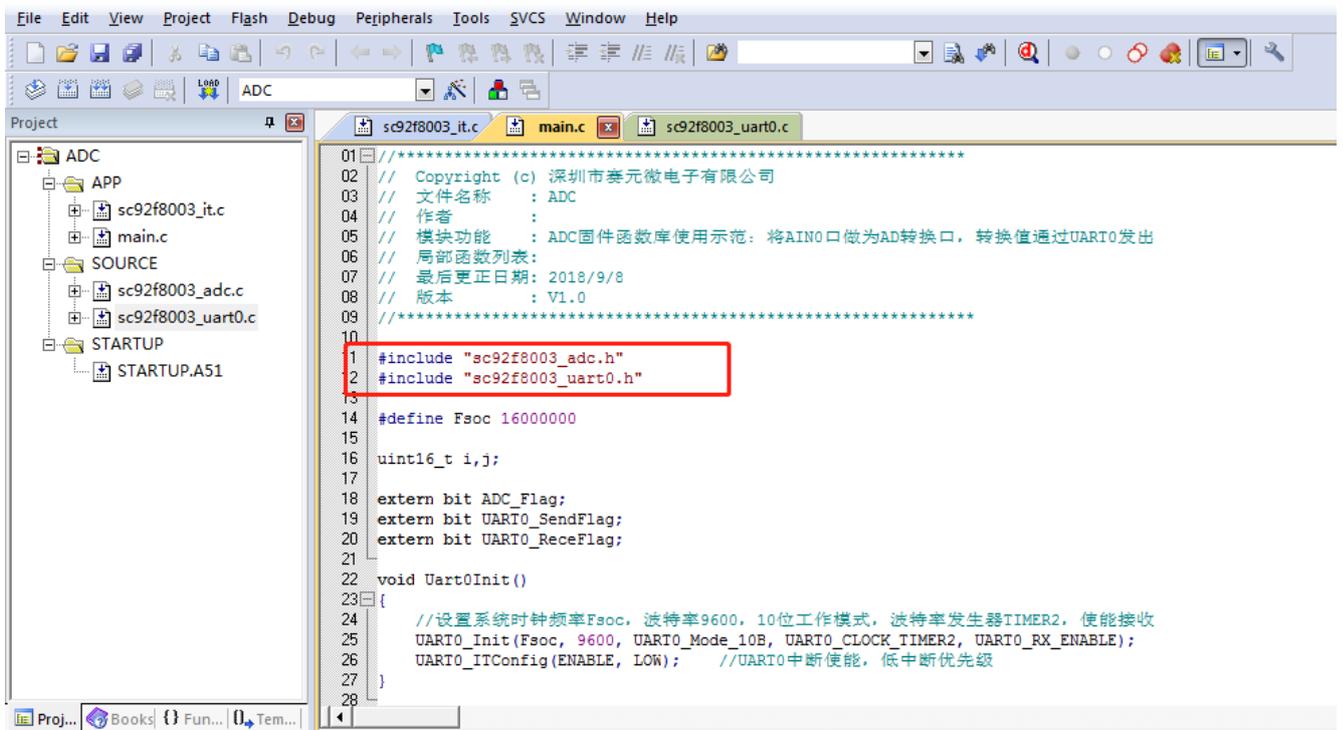


3、在工程内调用 SC92F8003_Lib 中所需的源文件。



```
001 //*****
002 // Copyright (c) 深圳市赛元微电子有限公司
003 // 文件名称 : sc92f8003_uart0.c
004 // 作者 :
005 // 模块功能 : UART0固件库函数C文件
006 // 局部函数列表:
007 // 最后更正日期: 2018/9/4
008 // 版本 : V1.0
009 // 说明 :
010 //*****
011
012 #include "sc92f8003_uart0.h"
013
014 //*****
015 *函数名称: void UART0_DeInit(void)
016 *函数功能: UART0相关寄存器复位至缺省值
017 *入口参数: void
018 *出口参数: void
019 //*****/
020 void UART0_DeInit(void)
021 {
022     OTCN  &= 0XEF;
023     SCN   = 0X00;
024     PCN  &= 0X7F;
025     IE   = (~0X10);
026     IF   = (~0X10);
027 }
028
```

4、调用固件库函数的文件内包含相关的头文件。



```
01 //*****
02 // Copyright (c) 深圳市赛元微电子有限公司
03 // 文件名称 : ADC
04 // 作者 :
05 // 模块功能 : ADC固件库函数使用示范: 将AIN0口做为AD转换口, 转换值通过UART0发出
06 // 局部函数列表:
07 // 最后更正日期: 2018/9/8
08 // 版本 : V1.0
09 //*****
10
11 #include "sc92f8003_adc.h"
12 #include "sc92f8003_uart0.h"
13
14 #define Fsoc 16000000
15
16 uint16_t i,j;
17
18 extern bit ADC_Flag;
19 extern bit UART0_SendFlag;
20 extern bit UART0_ReceFlag;
21
22 void Uart0Init()
23 {
24     //设置系统时钟频率Fsoc, 波特率9600, 10位工作模式, 波特率发生器TIMER2, 使能接收
25     UART0_Init(Fsoc, 9600, UART0_Mode_10B, UART0_CLOCK_TIMER2, UART0_RX_ENABLE);
26     UART0_ITConfig(ENABLE, LOW); //UART0中断使能, 低中断优先级
27 }
28
```

至此，完成对固件库函数的调用。

第二部分：固件库函数说明

一、GPIO 固件库函数

SC92F8003 提供了最多 18 个可控制的双向 GPIO 端口，输入输出控制寄存器用来控制各端口的输入输出状态。此 18 个 IO 口同其他功能复用。

1、GPIO 固件库函数列表

函数名	描述
GPIO_DeInit	GPIO 相关寄存器复位至缺省值
GPIO_Init	GPIO 模式配置初始化
GPIO_Write	GPIO 口 Px 赋值
GPIO_WriteHigh	GPIO 口管脚 Pxy 置位
GPIO_WriteLow	GPIO 口管脚 Pxy 复位
GPIO_ReadPort	读 GPIO 口 Px 的值
GPIO_ReadPin	读 GPIO 口 Pxy 的值

2、GPIO 固件库函数详解

(1) 函数 GPIO_DeInit

函数名	GPIO_DeInit
函数原型	void GPIO_DeInit(void)
功能描述	GPIO 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) 函数 GPIO_Init

函数名	GPIO_Init
函数原型	void GPIO_Init(GPIO_TypeDef GPIOx, uint8_t PortPins, GPIO_Mode_TypeDef GPIO_Mode)
功能描述	GPIO 口模式配置初始化
输入参数 1	GPIOx 选择操作的 GPIO 口
输入参数 2	PortPins 选择 GPIO 管脚 Pxy
输入参数 3	GPIO_Mode 选择 GPIO 口模式
返回值	无

GPIOx 可选值

GPIOx	描述
GPIO0	选择 P0 口
GPIO1	选择 P1 口
GPIO2	选择 P2 口
GPIO5	选择 P5 口

PortPins 可选值

PortPins	描述
GPIO_PIN_0	IO 口的 PIN0 脚
GPIO_PIN_1	IO 口的 PIN1 脚
GPIO_PIN_2	IO 口的 PIN2 脚
GPIO_PIN_3	IO 口的 PIN3 脚
GPIO_PIN_4	IO 口的 PIN4 脚

GPIO_PIN_5	IO 口的 PIN5 脚
GPIO_PIN_6	IO 口的 PIN6 脚
GPIO_PIN_7	IO 口的 PIN7 脚
GPIO_PIN_LNIB	IO 口的 PIN0~PIN3 脚
GPIO_PIN_HNIB	IO 口的 PIN4~PIN7 脚
GPIO_PIN_ALL	IO 口的全部 PIN 脚

GPIO_Mode 可选值

GPIO_Mode	描述
GPIO_MODE_IN_HI	高阻输入模式
GPIO_MODE_IN_PU	带上拉的输入模式
GPIO_MODE_OUT_PP	强推挽输出模式

(3) 函数 GPIO_Write

函数名	GPIO_Write
函数原型	void GPIO_Write(GPIO_TypeDef GPIOx, uint8_t PortVal)
功能描述	GPIO 口赋值
输入参数 1	GPIOx 选择操作的 GPIO 口
输入参数 2	PortVal GPIO 口的值
返回值	无

(4) 函数 GPIO_WriteHigh

函数名	GPIO_WriteHigh
函数原型	void GPIO_WriteHigh(GPIO_TypeDef GPIOx, uint8_t PortPins)
功能描述	GPIO 口管脚 Pxy 置位
输入参数 1	GPIOx 选择操作的 GPIO 口
输入参数 2	PortPins 选择 GPIO 口管脚 Pxy
返回值	无

(5) 函数 GPIO_WriteLow

函数名	GPIO_WriteLow
函数原型	void GPIO_WriteLow(GPIO_TypeDef GPIOx, uint8_t PortPins)
功能描述	GPIO 口管脚 Pxy 复位
输入参数 1	GPIOx 选择操作的 GPIO 口
输入参数 2	PortPins 选择 GPIO 口管脚 Pxy
返回值	无

(6) 函数 GPIO_ReadPort

函数名	GPIO_ReadPort
函数原型	uint8_t GPIO_ReadPort(GPIO_TypeDef GPIOx)
功能描述	读 GPIO 口 Px 的值
输入参数	GPIOx 选择操作的 GPIO 口
返回值	uint8_t 返回 Px 的值

(7) 函数 GPIO_ReadPin

函数名	GPIO_ReadPin
函数原型	BitStatus GPIO_ReadPin(GPIO_TypeDef GPIOx, GPIO_Pin_TypeDef PortPins)
功能描述	读 GPIO 口 Pxy 的值
输入参数 1	GPIOx 选择操作的 GPIO 口
输入参数 2	PortPins 选择操作的 PIN 口

返回值	BitStatus 返回 Pxy 的值
-----	---------------------

二、EXTI 固件库函数

SC92F8003 提供三个外部中断，可以分别设定其中每个中断源的触发条件为上升、下降或上下沿。

1、EXTI 固件库函数列表

函数名	描述
EXTI_DeInit	INTx 相关寄存器复位至缺省值
EXTI_SetExtInt0xTriggerMode	INT0x 初始化
EXTI_SetExtInt1xTriggerMode	INT1x 初始化
EXTI_SetExtInt2xTriggerMode	INT2x 初始化
EXTI_ITConfig	INTx 中断使能与失能配置

2、EXTI 固件库函数详解

(1) 函数 EXTI_DeInit

函数名	EXTI_DeInit
函数原型	void EXTI_DeInit(EXTIx_Typedef INTx)
功能描述	INTx 相关寄存器复位至缺省值
输入参数	INTx 选择外部中断 INTx
返回值	无

INTx 可选值

INTx	描述
INT0	选择外部中断 0
INT1	选择外部中断 1
INT2	选择外部中断 2

(2) 函数 EXTI_SetExtInt0xTriggerMode

函数名	EXTI_SetExtInt0xTriggerMode
函数原型	void EXTI_SetExtInt0xTriggerMode(uint8_t INT0x, EXTI_TriggerMode_Typedef TriggerMode)
功能描述	外部中断 0 初始化
输入参数 1	INT0x 选择外部中断 0 的管脚 INT0x
输入参数 2	TriggerMode 选择中断触发方式
返回值	无

INT0x 可选值

INT0x	描述
INT00	将 P00 脚设置为外部中断脚
INT01	将 P01 脚设置为外部中断脚

TriggerMode 可选值

TriggerMode	描述
EXTI_TRIGGER_RISE_ONLY	外部中断触发方式为上升沿
EXTI_TRIGGER_FALL_ONLY	外部中断触发方式为下降沿
EXTI_TRIGGER_RISE_FALL	外部中断触发方式为上升沿与下降沿
EXTI_TRIGGER_DISABLE	关闭外部中断触发

(3) 函数 EXTI_SetExtInt1xTriggerMode

函数名	EXTI_SetExtInt1xTriggerMode
函数原型	void EXTI_SetExtInt1xTriggerMode(uint8_t INT1x, EXTI_TriggerMode_Typedef TriggerMode)
功能描述	外部中断 1 初始化
输入参数 1	INT1x 选择外部中断 1 的管脚 INT1x
输入参数 2	TriggerMode 选择中断触发方式
返回值	无

INT1x 可选值

INT1x	描述
INT10	将 P10 脚设置为外部中断脚
INT11	将 P11 脚设置为外部中断脚
INT12	将 P12 脚设置为外部中断脚
INT13	将 P13 脚设置为外部中断脚
INT14	将 P14 脚设置为外部中断脚
INT15	将 P15 脚设置为外部中断脚
INT16	将 P16 脚设置为外部中断脚

(4) 函数 EXTI_SetExtIn2xTriggerMode

函数名	EXTI_SetExtIn2xTriggerMode
函数原型	void EXTI_SetExtInt2xTriggerMode(uint8_t INT2x, EXTI_TriggerMode_Typedef TriggerMode)
功能描述	外部中断 2 初始化
输入参数 1	INT2x 选择外部中断 2 的管脚 INT2x
输入参数 2	TriggerMode 选择中断触发方式
返回值	无

INT2x 可选值

INT2x	描述
INT21	将 P21 脚设置为外部中断脚
INT22	将 P22 脚设置为外部中断脚
INT23	将 P23 脚设置为外部中断脚
INT24	将 P24 脚设置为外部中断脚
INT25	将 P25 脚设置为外部中断脚
INT26	将 P26 脚设置为外部中断脚
INT27	将 P27 脚设置为外部中断脚

(5) 函数 EXTI_ITConfig

函数名	EXTI_ITConfig
函数原型	void EXTI_ITConfig(EXTIx_Typedef INTx, FunctionalState NewState, PriorityStatus Priority)
功能描述	外部中断初始化
输入参数 1	INTx 选择要打开的外部中断
输入参数 2	NewState 外部中断使能、关闭状态, 可取值 ENABLE 或 DISABLE
输入参数 3	Priority 外部中断优先级, 可取值 HIGH 或 LOW
返回值	无

三、PWM 固件库函数

SC92F8003 提供了最多 7 路共用周期、单独可调占空比的 10 位 PWM 输出: PWM0~6。

1、PWM 固件库函数列表

函数名	描述
PWM_DeInit	PWM 相关寄存器复位至缺省值
PWM_Init	PWM 初始化配置函数
PWM_OutputStateConfig	PWMx 输出使能/失能配置
PWM_PolarityConfig	PWMx 正/反向输出配置
PWM_IndependentModeConfig	PWMx 独立工作模式配置
PWM_ComplementaryModeConfig	PWMxPWMy 互补工作模式配置函数
PWM_DeadTimeConfig	死区时间配置
PWM_Cmd	PWM 功能开关配置
PWM_ITConfig	PWM 中断初始化
PWM_GetFlagStatus	获取 PWM 中断标志状态
PWM_ClearFlag	清除 PWM 中断标志状态

2、PWM 固件库函数详解

(1) 函数 PWM_DeInit

函数名	PWM_DeInit
函数原型	void PWM_DeInit(void)
功能描述	PWM 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) 函数 PWM_Init

函数名	PWM_Init
函数原型	void PWM_Init(PWM_PresSel_TypeDef PWM_PresSel, uint16_t PWM_Period)
功能描述	PWM 初始化
输入参数 1	PWM_PresSel 预分频选择
输入参数 2	PWM_Period PWM 周期配置
返回值	无

PWM_PresSel 可选值

PWM_PresSel	描述
PWM_PRESSEL_FOSC_D1	PWM 时钟 1 分频
PWM_PRESSEL_FOSC_D2	PWM 时钟 2 分频
PWM_PRESSEL_FOSC_D8	PWM 时钟 8 分频
PWM_PRESSEL_FOSC_D32	PWM 时钟 32 分频

(3) 函数 PWM_OutputStateConfig

函数名	PWM_OutputStateConfig
函数原型	void PWM_OutputStateConfig(uint8_t PWM_OutputPin, PWM_OutputState_TypeDef PWM_OutputState);
功能描述	PWMx 输出使能/失能配置
输入参数 1	PWM_OutputPin PWMx 选择
输入参数 2	PWM_OutputState PWM 输出状态配置
返回值	无

PWM_OutputPin 可选值

PWM_OutputPin	描述
PWM0	选择 PWM0 输出
PWM1	选择 PWM1 输出
PWM2	选择 PWM2 输出

PWM3	选择 PWM3 输出
PWM4	选择 PWM4 输出
PWM5	选择 PWM5 输出
PWM6	选择 PWM6 输出

PWM_OutputState 可选值

PWM_OutputState	描述
PWM_OUTPUTSTATE_DISABLE	该 PIN 脚作为 GPIO
PWM_OUTPUTSTATE_ENABLE	该 PIN 脚作为 PWM 输出脚

(4) 函数 PWM_PolarityConfig

函数名	PWM_PolarityConfig
函数原型	void PWM_PolarityConfig(uint8_t PWM_OutputPin, PWM_Polarity_TypeDef PWM_Polarity);
功能描述	PWMx 正/反向输出配置
输入参数 1	PWM_OutputPin PWMx 选择
输入参数 2	PWM_Polarity 选择 PWM 输出正向、反向
返回值	无

PWM_Polarity 可选值

PWM_Polarity	描述
PWM_POLARITY_NON_INVERT	PWM 输出不反向
PWM_POLARITY_INVERT	PWM 输出反向

(5) 函数 PWM_IndependentModeConfig

函数名	PWM_IndependentModeConfig
函数原型	void PWM_IndependentModeConfig(PWM_OutputPin_TypeDef PWM_OutputPin, uint16_t PWM_DutyCycle)
功能描述	PWMx 独立工作模式配置
输入参数 1	PWM_OutputPin PWMx 选择
输入参数 2	PWM_DutyCycle PWM 占空比, 取值范围为 0~1023
返回值	无

(6) 函数 PWM_ComplementaryModeConfig

函数名	PWM_ComplementaryModeConfig
函数原型	void PWM_ComplementaryModeConfig(PWM_ComplementaryOutputPin_TypeDef PWM_ComplementaryOutputPin, uint16_t PWM_DutyCycle)
功能描述	PWMxPWMy 互补工作模式配置
输入参数 1	PWM_ComplementaryOutputPin PWMxPWMy 互补通道选择
输入参数 2	PWM_DutyCycle PWM 占空比, 取值范围为 0~1023
返回值	无

PWM_ComplementaryOutputPin 可选值

PWM_ComplementaryOutputPin	描述
PWM0PWM3	互补通道选择: PWM0、PWM3
PWM1PWM4	互补通道选择: PWM1、PWM4
PWM2PWM5	互补通道选择: PWM2、PWM5

(7) 函数 PWM_DeadTimeConfig

函数名	PWM_DeadTimeConfig
-----	--------------------

函数原型	void PWM_DeadTimeConfig(uint8_t PWM012_RisingDeadTime, uint8_t PWM345_fallingDeadTime)
功能描述	死区时间配置
输入参数 1	PWM012_RisingDeadTime PWM 死区上升时间, 取值范围为 0~15
输入参数 2	PWM345_fallingDeadTime PWM 死区下降时间, 取值范围为 0~15
返回值	无

(8) 函数 PWM_Cmd

函数名	PWM_Cmd
函数原型	void PWM_Cmd(FunctionalState NewState)
功能描述	PWM 功能启动/关闭选择
输入参数	NewState PWM 功能启动/关闭选择, 可取值 ENABLE 或 DISABLE
返回值	无

(9) 函数 PWM_ITConfig

函数名	PWM_ITConfig
函数原型	void PWM_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	PWM 中断使能配置
输入参数 1	NewState 中断使能/关闭选择, 可取值 ENABLE 或 DISABLE
输入参数 2	Priority 中断优先级选择, 可取值 HIGH 或 LOW
返回值	无

(10) 函数 PWM_GetFlagStatus

函数名	PWM_GetFlagStatus
函数原型	FlagStatus PWM_GetFlagStatus(void)
功能描述	获得 PWM 中断标志状态
输入参数	无
返回值	FlagStatus PWM 中断标志状态

(11) 函数 PWM_ClearFlag

函数名	PWM_ClearFlag
函数原型	void PWM_ClearFlag(void)
功能描述	清除 PWM 中断标志状态
输入参数	无
返回值	无

四、ADC 固件库函数

SC92F8003 内建一个 12bit 的 11 通道高精度逐次逼近型 ADC, 内部还可选择 1/4VDD 输入通道, 配合内部 2.4V 参考电压用于测量 VDD 电压。

1、ADC 固件库函数列表

函数名	描述
ADC_DelInit	ADC 相关寄存器复位至缺省值
ADC_Init	ADC 初始化
ADC_ChannelConfig	ADC 输入口配置
ADC_Cmd	ADC 功能开关配置
ADC_StartConversion	启动一次 ADC 转换
ADC_GetConversionValue	获得一次 ADC 转换值
ADC_ITConfig	ADC 中断初始化

ADC_GetFlagStatus	获得 ADC 中断标志状态
ADC_ClearFlag	清空转换标志

2、ADC 固件库函数详解

(1) 函数 ADC_DeInit

函数名	ADC_DeInit
函数原型	void ADC_DeInit(void)
功能描述	ADC 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) 函数 ADC_Init

函数名	ADC_Init
函数原型	void ADC_Init(ADC_PresSel_TypeDef ADC_PrescalerSelection, ADC_Cycle_TypeDef ADC_Cycle)
功能描述	ADC 初始化
输入参数 1	ADC_PrescalerSelection 预分频选择
输入参数 2	ADC_Cycle 采样时钟周期选择
返回值	无

ADC_PrescalerSelection 可选值

ADC_PrescalerSelection	描述
ADC_PRESSEL_FHRC_D32	fADC = fHRC/32
ADC_PRESSEL_FHRC_D24	fADC = fHRC/24
ADC_PRESSEL_FHRC_D16	fADC = fHRC/16
ADC_PRESSEL_FHRC_D12	fADC = fHRC/12
ADC_PRESSEL_FHRC_D8	fADC = fHRC/8
ADC_PRESSEL_FHRC_D6	fADC = fHRC/6
ADC_PRESSEL_FHRC_D4	fADC = fHRC/4
ADC_PRESSEL_FHRC_D3	fADC = fHRC/3

ADC_Cycle 可选值

ADC_Cycle	描述
ADC_Cycle_6Cycle	ADC 采样时间为 6 个 ADC 采样时钟周期
ADC_Cycle_36Cycle	ADC 采样时间为 36 个 ADC 采样时钟周期

(3) 函数 ADC_ChannelConfig

函数名	ADC_ChannelConfig
函数原形	void ADC_ChannelConfig(ADC_Channel_TypeDef ADC_Channel, FunctionalState NewState)
功能描述	ADC 输入口配置
输入参数 1	ADC_Channel ADC 输入口选择
输入参数 2	NewState ADC 输入口使能或关闭
返回值	无

ADC_Channel 可选值

ADC_Channel	描述
ADC_CHANNEL_0	选择 AIN0 做 AD 输入口
ADC_CHANNEL_1	选择 AIN1 做 AD 输入口
ADC_CHANNEL_2	选择 AIN2 做 AD 输入口
ADC_CHANNEL_3	选择 AIN3 做 AD 输入口
ADC_CHANNEL_4	选择 AIN4 做 AD 输入口

ADC_CHANNEL_5	选择 AIN5 做 AD 输入口
ADC_CHANNEL_6	选择 AIN6 做 AD 输入口
ADC_CHANNEL_7	选择 AIN7 做 AD 输入口
ADC_CHANNEL_8	选择 AIN8 做 AD 输入口
ADC_CHANNEL_9	选择 AIN9 做 AD 输入口
ADC_CHANNEL_VDD_D4	选择 1/4V _{DD} 做 AD 输入口

(4) 函数 ADC_Cmd

函数名	ADC_Cmd
函数原形	void ADC_Cmd(FunctionalState NewState)
功能描述	ADC 功能开关配置
输入参数	NewState ADC 功能开关选择
返回值	无

(5) 函数 ADC_StartConversion

函数名	ADC_StartConversion
函数原形	void ADC_StartConversion(void)
功能描述	开始一次 AD 转换
输入参数	无
返回值	无

(6) 函数 ADC_GetConversionValue

函数名	ADC_GetConversionValue
函数原形	uint16_t ADC_GetConversionValue(void)
功能描述	获得一次 AD 转换数据
输入参数	无
返回值	uint16_t AD 转换结果

(7) 函数 ADC_ITConfig

函数名	ADC_ITConfig
函数原形	void ADC_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	使能或关闭 ADC 中断
输入参数 1	NewState 中断使能或关闭选择
输入参数 2	Priority 中断优先级选择
返回值	无

(8) 函数 ADC_GetFlagStatus

函数名	ADC_GetFlagStatus
函数原形	FlagStatus ADC_GetFlagStatus(void)
功能描述	获得 ADC 中断标志状态
输入参数	无
返回值	FlagStatus ADC 中断标志状态

(9) 函数 ADC_ClearFlag

函数名	ADC_ClearFlag
函数原形	void ADC_ClearFlag(void)
功能描述	清除 ADC 中断标志状态
输入参数	无

返回值	无
-----	---

五、TIMER0 固件库函数

TIMER0 为一个 16 位定时/计数器，具有计数方式和定时方式两种工作模式。

T0 有 4 种工作模式：

- (1) 模式 0：13 位定时器/计数器模式
- (2) 模式 1：16 位定时器/计数器模式
- (3) 模式 2：8 位自动重载模式
- (4) 模式 3：两个 8 位定时器/计数器模式

1、TIMER0 固件库函数列表

函数名	描述
TIM0_DelNit	TIM0 相关寄存器复位至缺省值
TIM0_TimeBaselNit	配置 TIM0 基本设置
TIM0_WorkMode0Config	TIM0 工作模式 0 配置
TIM0_WorkMode1Config	TIM0 工作模式 1 配置
TIM0_WorkMode2Config	TIM0 工作模式 2 配置
TIM0_WorkMode3Config	TIM0 工作模式 3 配置
TIM0_Mode0SetReloadCounter	TIM0 工作模式 0 计数值重载
TIM0_Mode1SetReloadCounter	TIM0 工作模式 1 计数值重载
TIM0_SetTL0Counter	TL0 赋值函数
TIM0_SetTH0Counter	TH0 赋值函数
TIM0_Cmd	TIM0 计数使能或关闭设置
TIM0_ITConfig	TIM0 中断设置
TIM0_GetFlagStatus	获得 TIM0 中断标志状态
TIM0_ClearFlag	清除 TIM0 中断标志

2、TIMER0 固件库函数详解

(1) TIM0_DelNit 函数

函数名	TIM0_DelNit
函数原型	void TIM0_DelNit(void)
功能描述	将 TIM0 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) TIM0_TimeBaselNit 函数

函数名	TIM0_TimeBaselNit
函数原型	void TIM0_TimeBaselNit(TIM0_PresSel_TypeDef TIM0_PrescalerSelection, TIM0_CountMode_TypeDef TIM0_CountMode)
功能描述	TIM0 基本设置
输入参数 1	TIM0_PrescalerSelection 预分频选择
输入参数 2	TIM0_CountMode 计数/定时模式选择
返回值	无

TIM0_PrescalerSelection 可选值

TIM0_PrescalerSelection	描述
TIM0_PRESSEL_FSYS_D12	TIMER0 计数源来自系统时钟 12 分频
TIM0_PRESSEL_FSYS_D1	TIMER0 计数源来自系统时钟

TIM0_CountMode 可选值

TIM0_CountMode	描述
TIM0_MODE_TIMER	TIMER0 做定时器
TIM0_MODE_COUNTER	TIMER0 做计数器

(3) TIM0_WorkMode0Config 函数

函数名	TIM0_WorkMode0Config
函数原型	void TIM0_WorkMode0Config(uint16_t TIM0_SetCounter)
功能描述	TIM0 工作模式 0 配置
输入参数	TIM0_SetCounter TIM0 计数初值
返回值	无

(4) TIM0_WorkMode1Config 函数

函数名	TIM0_WorkMode1Config
函数原型	void TIM0_WorkMode1Config(uint16_t TIM0_SetCounter)
功能描述	TIM0 工作模式 1 配置
输入参数	TIM0_SetCounter TIM0 计数初值
返回值	无

(5) TIM0_WorkMode2Config 函数

函数名	TIM0_WorkMode2Config
函数原型	void TIM0_WorkMode2Config(uint8_t TIM0_SetCounter)
功能描述	TIM0 工作模式 2 配置
输入参数	TIM0_SetCounter TIM0 计数初值
返回值	无

(6) TIM0_WorkMode3Config 函数

函数名	TIM0_WorkMode3Config
函数原型	void TIM0_WorkMode3Config(uint8_t TIM0_SetCounter, uint8_t TIM1_SetCounter)
功能描述	TIM0 工作模式 3 配置
输入参数 1	TIM0_SetCounter TIM0 初值设置
输入参数 2	TIM1_SetCounter TIM1 初值设置
返回值	无

(7) TIM0_Mode0SetReloadCounter 函数

函数名	TIM0_Mode0SetReloadCounter
函数原型	void TIM0_Mode0SetReloadCounter(uint16_t TIM0_SetCounter)
功能描述	TIMER0 工作模式 0 计数值重载
输入参数	TIM0_SetCounter TIM0 计数值重载
返回值	无

(8) TIM0_Mode1SetReloadCounter 函数

函数名	TIM0_Mode1SetReloadCounter
函数原型	void TIM0_Mode1SetReloadCounter(uint16_t TIM0_SetCounter)
功能描述	TIMER0 工作模式 1 计数值重载
输入参数	TIM0_SetCounter TIM0 计数值重载
返回值	无

(9) TIM0_Cmd 函数

函数名	TIM0_Cmd
-----	----------

函数原型	void TIM0_Cmd(FunctionalState NewState)
功能描述	TIM0 使能或关闭
输入参数	NewState TIM0 使能或关闭
返回值	无

(10) TIM0_ITConfig 函数

函数名	TIM0_ITConfig
函数原型	void TIM0_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	TIM0 中断设置
输入参数 1	NewState 中断使能或关闭
输入参数 2	Priority 中断优先级设置
返回值	无

(11) TIM0_GetFlagStatus 函数

函数名	TIM0_GetFlagStatus
函数原型	FlagStatus TIM0_GetFlagStatus(void)
功能描述	获得 TIM0 中断标志状态
输入参数	无
返回值	FlagStatus 中断标志状态

(12) TIM0_ClearFlag 函数

函数名	TIM0_ClearFlag
函数原型	void TIM0_ClearFlag(void)
功能描述	清除 TIM0 中断标志
输入参数	无
返回值	无

(13) TIM0_SetTL0Counter 函数

函数名	TIM0_SetTL0Counter
函数原型	void TIM0_SetTL0Counter(uint8_t TIM0_SetCounter)
功能描述	清除 TIM0 中断标志
输入参数	TIM0_SetCounter 配置 TL0
返回值	无

(14) TIM0_SetTH0Counter 函数

函数名	TIM0_SetTH0Counter
函数原型	void TIM0_SetTH0Counter(uint8_t TIM0_SetCounter)
功能描述	清除 TIM0 中断标志
输入参数	TIM0_SetCounter 配置 TH0
返回值	无

六、TIMER1 固件库函数

TIMER1 为一个 16 位定时/计数器，他们具有计数方式和定时方式两种工作模式。

T1 有 3 种工作模式（模式 3 不存在）：

- (1) 模式 0：13 位定时器/计数器模式
- (2) 模式 1：16 位定时器/计数器模式
- (3) 模式 2：8 位自动重载模式

1、TIMER1 固件库函数列表

函数名	描述
TIM1_DelNit	TIM1 相关寄存器复位至缺省值
TIM1_TimeBaselnit	配置 TIM1 基本设置
TIM1_WorkMode0Config	TIM1 工作模式 0 配置
TIM1_WorkMode1Config	TIM1 工作模式 1 配置
TIM1_WorkMode2Config	TIM1 工作模式 2 配置
TIM1_Mode0SetReloadCounter	TIM1 工作模式 0 计数值重载
TIM1_Mode1SetReloadCounter	TIM1 工作模式 1 计数值重载
TIM1_Cmd	TIM1 计数使能或关闭设置
TIM1_ITConfig	TIM1 中断设置
TIM1_GetFlagStatus	获得 TIM1 中断标志状态
TIM1_ClearFlag	清除 TIM1 中断标志

2、TIMER1 固件库函数详解

(1) TIM1_DelNit 函数

函数名	TIM1_DelNit
函数原型	void TIM1_DelNit(void)
功能描述	将 TIM1 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) TIM1_TimeBaselnit 函数

函数名	TIM1_TimeBaselnit
函数原型	void TIM1_TimeBaselnit(TIM1_PresSel_TypeDef TIM1_PrescalerSelection, TIM1_CountMode_TypeDef TIM1_CountMode)
功能描述	TIM1 基本设置
输入参数 1	TIM1_PrescalerSelection 预分频选择
输入参数 2	TIM1_CountMode 计数/定时模式选择
返回值	无

TIM1_PrescalerSelection 可选值

TIM1_PrescalerSelection	描述
TIM1_PRESSEL_FSYS_D12	TIMER1 计数源来自系统时钟 12 分频
TIM1_PRESSEL_FSYS_D1	TIMER1 计数源来自系统时钟

TIM1_CountMode 可选值

TIM0_CountMode	描述
TIM1_MODE_TIMER	TIMER1 做定时器
TIM1_MODE_COUNTER	TIMER1 做计数器

(3) TIM1_WorkMode0Config 函数

函数名	TIM1_WorkMode0Config
函数原型	void TIM1_WorkMode0Config(uint16_t TIM1_SetCounter)
功能描述	TIM1 工作模式 0 配置
输入参数	TIM1_SetCounter TIM1 计数初值
返回值	无

(4) TIM1_WorkMode1Config 函数

函数名	TIM1_WorkMode1Config
函数原型	void TIM1_WorkMode1Config(uint16_t TIM1_SetCounter)

功能描述	TIM1 工作模式 1 配置
输入参数	TIM1_SetCounter TIM1 计数初值
返回值	无

(5) TIM1_WorkMode2Config 函数

函数名	TIM1_WorkMode2Config
函数原型	void TIM1_WorkMode2Config(uint8_t TIM1_SetCounter)
功能描述	TIM1 工作模式 2 配置
输入参数	TIM1_SetCounter TIM1 计数初值
返回值	无

(6) TIM1_Mode0SetReloadCounter 函数

函数名	TIM1_Mode0SetReloadCounter
函数原型	void TIM1_Mode0SetReloadCounter(uint16_t TIM1_SetCounter)
功能描述	TIMER1 工作模式 0 计数值重载
输入参数	TIM1_SetCounter TIM1 计数值重载
返回值	无

(7) TIM1_Mode1SetReloadCounter 函数

函数名	TIM1_Mode1SetReloadCounter
函数原型	void TIM1_Mode1SetReloadCounter(uint16_t TIM1_SetCounter)
功能描述	TIMER1 工作模式 1 计数值重载
输入参数	TIM1_SetCounter TIM1 计数值重载
返回值	无

(8) TIM1_Cmd 函数

函数名	TIM1_Cmd
函数原型	void TIM1_Cmd(FunctionalState NewState)
功能描述	TIM1 使能或关闭
输入参数	NewState TIM1 使能或关闭
返回值	无

(9) TIM1_ITConfig 函数

函数名	TIM1_ITConfig
函数原型	void TIM1_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	TIM1 中断设置
输入参数 1	NewState 中断使能或关闭
输入参数 2	Priority 中断优先级设置
返回值	无

(10) TIM1_GetFlagStatus 函数

函数名	TIM1_GetFlagStatus
函数原型	FlagStatus TIM1_GetFlagStatus(void)
功能描述	获得 TIM1 中断标志状态
输入参数	无
返回值	FlagStatus 中断标志状态

(11) TIM1_ClearFlag 函数

函数名	TIM1_ClearFlag
-----	----------------

函数原型	void TIM1_ClearFlag(void)
功能描述	清除 TIM1 中断标志
输入参数	无
返回值	无

七、TIMER2 固件库函数

TIMER2 是一个可通过可编程预分频器驱动的 16 位自动装载计数器。其有 4 种工作模式：

- (1) 模式 0：16 位捕获模式
- (2) 模式 1：16 位自动重载定时器模式
- (3) 模式 2：波特率发生器模式
- (4) 模式 3：可编程时钟输出模式

1、TIMER2 固件库函数列表

函数名	描述
TIM2_DeInit	TIM2 相关寄存器复位至缺省值
TIM2_TimeBaseInit	配置 TIM2 基本设置
TIM2_WorkMode0Config	TIM2 工作模式 0 配置
TIM2_WorkMode1Config	TIM2 工作模式 1 配置
TIM2_WorkMode3Config	TIM2 工作模式 3 配置
TIM2_SetEXEN2	EXEN2 配置
TIM2_Cmd	TIM2 计数使能或关闭设置
TIM2_ITConfig	TIM2 中断设置
TIM2_GetFlagStatus	获得 TIM2 中断标志状态
TIM2_ClearFlag	清除 TIM2 中断标志

2、TIMER2 固件库函数详解

(1) TIM2_DeInit 函数

函数名	TIM2_DeInit
函数原型	void TIM2_DeInit (void)
功能描述	TIM2 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) TIM2_TimeBaseInit 函数

函数名	TIM2_TimeBaseInit
函数原型	TIM2_TimeBaseInit(TIM2_PresSel_TypeDef TIM2_PrescalerSelection, TIM2_CountMode_TypeDef TIM2_CountMode, TIM2_CountDirection_TypeDef TIM2_CountDirection)
功能描述	配置 TIM2 基本设置
输入参数 1	TIM2_PrescalerSelection 预分频选择
输入参数 2	TIM2_CountMode 计数/定时模式选择
输入参数 3	TIM2_CountDirection 计数方向选择
返回值	无

TIM2_PrescalerSelection 可选值

TIM2_PrescalerSelection	描述
TIM2_PRESSEL_FSYS_D12	TIMER2 计数源来自系统时钟 12 分频
TIM2_PRESSEL_FSYS_D1	TIMER2 计数源来自系统时钟

TIM2_CountMode 可选值

TIM2_CountMode	描述
TIM2_MODE_TIMER	TIMER2 做定时器
TIM2_MODE_COUNTER	TIMER2 做计数器

TIM2_CountDirection 可选值

TIM2_CountDirection	描述
TIM2_COUNTDIRECTION_UP	向上计数模式
TIM2_COUNTDIRECTION_DOWN_UP	向上/向下计数模式

(3) TIM2_WorkMode0Config 函数

函数名	TIM2_WorkMode0Config
函数原型	void TIM2_WorkMode0Config(uint16_t TIM2_SetCounter)
功能描述	TIM2 工作模式 0 配置
输入参数	TIM2_SetCounter TIM2 计数初值配置
返回值	无

(4) TIM2_WorkMode1Config 函数

函数名	TIM2_WorkMode1Config
函数原型	void TIM2_WorkMode1Config(uint16_t TIM2_SetCounter)
功能描述	TIM2 工作模式 1 配置
输入参数	TIM2_SetCounter TIM2 计数初值配置
返回值	无

(5) TIM2_WorkMode3Config 函数

函数名	TIM2_WorkMode3Config
函数原型	void TIM2_WorkMode3Config(uint16_t TIM2_SetCounter)
功能描述	TIM2 工作模式 3 配置
输入参数	TIM2_SetCounter TIM2 计数初值配置
返回值	无

(6) TIM2_SetEXEN2 函数

函数名	TIM2_SetEXEN2
函数原型	void TIM2_SetEXEN2(FunctionalState NewState)
功能描述	EXEN2 配置
输入参数	NewState 置位/复位 EXEN2, 可取值 ENABLE 或 DISABLE
返回值	无

(7) TIM2_Cmd 函数

函数名	TIM2_Cmd
函数原型	void TIM2_Cmd(FunctionalState NewState)
功能描述	TIM2 使能或关闭
输入参数	NewState 使能或关闭, 可取值 ENABLE 或 DISABLE
返回值	无

(8) TIM2_ITConfig 函数

函数名	TIM2_ITConfig
函数原型	void TIM2_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	TIM2 中断设置
输入参数 1	NewState 中断使能或关闭, 可取值 ENABLE 或 DISABLE
输入参数 2	Priority 中断优先级设置, 可取值 HIGH 或 LOW
返回值	无

(9) TIM2_GetFlagStatus 函数

函数名	TIM2_GetFlagStatus
函数原型	FlagStatus TIM2_GetFlagStatus(TIM2_Flag_TypeDef TIM2_Flag)
功能描述	获得 TIM2 的中断标志状态
输入参数	TIM2_Flag 标志位选择
返回值	FlagStatus 标志位状态

TIM2_Flag 可选值

TIM2_Flag	描述
TIM2_FLAG_TF2	标志位 TF2
TIM2_FLAG_EXF2	标志位 EXF2

(10) TIM2_ClearFlag 函数

函数名	TIM2_ClearFlag
函数原型	void TIM2_ClearFlag(TIM2_Flag_TypeDef TIM2_Flag)
功能描述	清空 TIM2 中断标志
输入参数	TIM2_Flag 标志位选择
返回值	无

八、UART0 固件库函数

SC92F8003支持一个全双工的串行口UART0的功能及特性如下:

- (1) 三种通讯模式可选: 模式 0、模式 1 和模式 3
- (2) 可选择定时器 1 或定时器 2 作为波特率发生器
- (3) 发送和接收完成可产生中断 RI/TI, 该中断标志需要软件清除

使用 UART0 需要用到 TIMER1 或 TIMER2 用作波特率发生器, 当使用 TIMER1 时, TIMER1 必须停止计数 (TR1=0), 以保证 UART0 的波特率正常。

1、UART0 固件库函数列表

函数名	描述
UART0_DeInit	将 UART0 寄存器设置为缺省值
UART0_PinSelection	UART0 引脚选择
UART0_Init	UART0 初始化配置函数
UART0_ReceiveData8	获得 SBUF 中的值
UART0_ReceiveData9	获得 SBUF 中的值及第九位的值
UART0_SendData8	发送 8 位的数据
UART0_SendData9	发送 9 位数据
UART0_ITConfig	使能或关闭 UART0 的中断
UART0_ClearFlag	清空 UART0 中断标志
UART0_GetITStatus	获得 UART0 中断标志状态

2、UART0 固件库函数详解

(1) UART0_DeInit 函数

函数名	UART0_DeInit
函数原型	void UART0_DeInit(void)
功能描述	将 UART0 寄存器设置为缺省值
输入参数	无
返回值	无

(2) UART0_PinSelection 函数

函数名	UART0_PinSelection
函数原型	void UART0_PinSelection(UART0_PinSelection_TypeDef PinSeletion)
功能描述	UART0 引脚选择
输入参数	PinSeletion 选择 UART0 引脚为 P15P16 或 P11P20
返回值	无

PinSeletion 可选值

PinSeletion	描述
UART0_PinSelection_P15P16	TIMER1 做 UART0 时钟源
UART0_PinSelection_P11P20	TIMER2 做 UART0 时钟源

(3) UART0_Init 函数

函数名	UART0_Init
函数原型	void UART0_Init(uint32_t Uart0Fsys, uint32_t BaudRate, UART0_Mode_Typedef Mode, UART0_Clock_Typedef ClockMode, UART0_RX_Typedef RxMode)
功能描述	UART0 初始化配置函数
输入参数 1	Uart0Fsys 系统主频
输入参数 2	BaudRate 波特率 (模式 0 下只可选择 UART0_BaudRate_FsysDIV12 或 UART0_BaudRate_FsysDIV4)
输入参数 3	Mode 工作模式
输入参数 4	ClockMode 时钟源选择
输入参数 5	RxMode RX 使能开关
返回值	无

BaudRate 可选值

BaudRate	描述
UART0_BaudRate_FsysDIV12	仅模式 0 可用, 串行端口在系统时钟的 1/12 下运行
UART0_BaudRate_FsysDIV4	仅模式 0 可用, 串行端口在系统时钟的 1/4 下运行
Uint32_t	10/11 位模式波特率

Mode 可选值

Mode	描述
UART0_Mode_8B	UART0 选为 8 位模式
UART0_Mode_10B	UART0 选为 10 位模式
UART0_Mode_11B	UART0 选为 11 位模式

ClockMode 可选值

ClockMode	描述
UART0_CLOCK_TIMER1	TIMER1 做 UART0 时钟源
UART0_CLOCK_TIMER2	TIMER2 做 UART0 时钟源

RxMod 可选值

RxMod	描述
UART0_RX_ENABLE	允许 UART0 接收
UART0_RX_DISABLE	禁止 UART0 接收

(4) UART0_ReceiveData8 函数

函数名	UART0_ReceiveData8
函数原型	uint8_t UART0_ReceiveData8(void)
功能描述	获得 SBUF 中的值
输入参数	无

返回值	uint8_t
-----	---------

(5) UART0_ReceiveData9 函数

函数名	UART0_ReceiveData9
函数原型	uint16_t UART0_ReceiveData9(void)
功能描述	获得 SBUF 中的值及第九位的值
输入参数	无
返回值	uint16_t

(6) UART0_SendData8 函数

函数名	UART0_SendData8
函数原型	void UART0_SendData8(uint8_t Data)
功能描述	发送 8 位数据
输入参数	Data 发送数据
返回值	无

(7) UART0_SendData9 函数

函数名	UART0_SendData9
函数原型	void UART0_SendData9(uint16_t Data)
功能描述	发送 9 位数据
输入参数	Data 发送 8 位数据及第 9 位
返回值	无

(8) UART0_ITConfig 函数

函数名	UART0_ITConfig
函数原型	void UART0_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	使能或关闭 UART0 中断
输入参数 1	NewState 使能或关闭, 可取值 ENABLE 或 DISABLE
输入参数 2	Priority 优先级, 可取值 LOW 或 HIGH
返回值	无

(9) UART0_ClearFlag 函数

函数名	UART0_ClearFlag
函数原型	void UART0_ClearFlag(UART0_Flag_TypeDef UART0_FLAG)
功能描述	清空 UART0 中断标志
输入参数	UART0_FLAG 选择 UART0 中断标志
返回值	无

UART0_FLAG 可选值

UART0_FLAG	描述
UART0_FLAG_TI	选择发送完成标志位
UART0_FLAG_RI	选择接收完成标志位

(10) UART0_GetFlagStatus 函数

函数名	UART0_GetFlagStatus
函数原型	FlagStatus UART0_GetFlagStatus(UART0_Flag_TypeDef UART0_Flag)
功能描述	获得 UART0 中断标志状态
输入参数	UART0_FLAG 选择 UART0 中断标志
返回值	无

九、SSI 固件库函数

SC92F8003 内部集成一个三选一串行接口电路 SSI，该接口可配置为 SPI、TWI、UART 中任意一种通信模式。其特点如下：

- (1) SPI 模式可配置为主模式或从属模式中的一种
- (2) TWI 模式通信时只能做从机
- (3) UART 模式可工作在模式 1（10 位全双工异步通信）和模式 3（11 位全双工异步通信）

1、SSI 固件库函数列表

函数名	描述
SSI_DeInit	SSI 相关寄存器复位至缺省值
SSI_SPI_Init	SPI 初始化配置函数
SSI_SPI_Cmd	SPI 功能开关函数
SSI_SPI_SendData	SPI 发送数据
SSI_SPI_ReceiveData	获得 SSDAT 中的值
SSI_TWI_Init	TWI 初始化配置函数
SSI_TWI_AcknowledgeConfig	TWI 接收应答使能函数
SSI_TWI_GeneralCallCmd	TWI 通用地址响应使能函数
SSI_TWI_Cmd	TWI 功能开关函数
SSI_TWI_SendData	TWI 发送数据
SSI_TWI_ReceiveData	获得 SSDAT 中的值
SSI_UART1_Init	UART1 初始化配置函数
SSI_UART1_SendData8	UART1 发送 8 位数据
SSI_UART1_ReceiveData8	获得 SSDAT 中的值
SSI_UART1_SendData9	UART1 发送 9 位数据
SSI_UART1_ReceiveData9	获得 SSDAT 中的值及第九位的值
SSI_ITConfig	SSI 中断初始化
SSI_GetFlagStatus	获得 SSI 中断标志状态
SSI_ClearFlag	清除 SSI 标志状态
SSI_PinSelection	SSI 引脚选择

2、SSI 固件库函数详解

(1) SSI_DeInit 函数

函数名	SSI_DeInit
函数原型	void SSI_DeInit(void)
功能描述	SSI 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) SSI_SPI_Init 函数

函数名	SSI_SPI_Init
函数原型	void SSI_SPI_Init(SPI_FirstBit_TypeDef FirstBit, SPI_BaudRatePrescaler_TypeDef BaudRatePrescaler, SPI_Mode_TypeDef Mode, SPI_ClockPolarity_TypeDef ClockPolarity, SPI_ClockPhase_TypeDef ClockPhase, SPI_TXE_INT_TypeDef SPI_TXE_INT)
功能描述	SPI 初始化配置函数
输入参数 1	FirstBit 优先传送位选择（MSB/LSB）
输入参数 2	BaudRatePrescaler SPI 时钟频率选择
输入参数 3	Mode SPI 工作模式选择
输入参数 4	ClockPolarity SPI 时钟极性选择

输入参数 5	ClockPhase	SPI 时钟相位选择
输入参数 6	SPI_TXE_INT	发送缓存器中断允许选择
返回值	无	

FirstBit 可选值

FirstBit	描述
SPI_FIRSTBIT_MSB	MSB 优先发送
SPI_FIRSTBIT_LSB	LSB 优先发送

BaudRatePrescaler 可选值

BaudRatePrescaler	描述
SPI_BAUDRATEPRESCALER_4	SPI 时钟速率为系统时钟除以 4
SPI_BAUDRATEPRESCALER_8	SPI 时钟速率为系统时钟除以 8
SPI_BAUDRATEPRESCALER_16	SPI 时钟速率为系统时钟除以 16
SPI_BAUDRATEPRESCALER_32	SPI 时钟速率为系统时钟除以 32
SPI_BAUDRATEPRESCALER_64	SPI 时钟速率为系统时钟除以 64
SPI_BAUDRATEPRESCALER_128	SPI 时钟速率为系统时钟除以 128
SPI_BAUDRATEPRESCALER_256	SPI 时钟速率为系统时钟除以 256
SPI_BAUDRATEPRESCALER_512	SPI 时钟速率为系统时钟除以 512

Mode 可选值

Mode	描述
SPI_MODE_MASTER	SPI 为主设备
SPI_MODE_SLAVE	SPI 为从设备

ClockPolarity 可选值

ClockPolarity	描述
SPI_CLOCKPOLARITY_LOW	SCK 在空闲状态下为低电平
SPI_CLOCKPOLARITY_HIGH	SCK 在空闲状态下为高电平

ClockPhase 可选值

ClockPhase	描述
SPI_CLOCKPHASE_1EDGE	SCK 的第一沿采集数据
SPI_CLOCKPHASE_2EDGE	SCK 的第二沿采集数据

SPI_TXE_INT 可选值

SPI_TXE_INT	描述
SPI_TXE_DISINT	TXE 为 0 时不允许发送中断
SPI_TXE_ENINT	TXE 为 1 时允许发送中断

(3) SSI_SPI_Cmd 函数

函数名	SSI_SPI_Cmd
函数原型	void SSI_SPI_Cmd(FunctionalState NewState)
功能描述	SPI 功能开关函数
输入参数	NewState 使能或关闭, 可取值 ENABLE 或 DISABLE
返回值	无

(4) SSI_SPI_SendData 函数

函数名	SSI_SPI_SendData
函数原型	void SSI_SPI_SendData(uint8_t Data)

功能描述	SPI 发送数据
输入参数	Data 发送的数据
返回值	无

(5) SSI_SPI_ReceiveData 函数

函数名	SSI_SPI_ReceiveData
函数原型	uint8_t SSI_SPI_ReceiveData(void)
功能描述	获得 SSDAT 中的值
输入参数	无
返回值	uint8_t SSDAT 中的值

(6) SSI_TWI_Init 函数

函数名	SSI_TWI_Init
函数原型	void SSI_TWI_Init(uint8_t TWI_Address)
功能描述	TWI 初始化配置函数
输入参数	TWI_Address 7 位从机地址配置
返回值	无

(7) SSI_TWI_AcknowledgeConfig 函数

函数名	SSI_TWI_AcknowledgeConfig
函数原型	void SSI_TWI_AcknowledgeConfig(FunctionalState NewState)
功能描述	TWI 接收应答使能函数
输入参数	NewState 使能或关闭, 可取值 ENABLE 或 DISABLE
返回值	无

(8) SSI_TWI_GeneralCallCmd 函数

函数名	SSI_TWI_GeneralCallCmd
函数原型	void SSI_TWI_GeneralCallCmd(FunctionalState NewState)
功能描述	TWI 通用地址响应使能函数
输入参数	NewState 使能或关闭, 可取值 ENABLE 或 DISABLE
返回值	无

(9) SSI_TWI_Cmd 函数

函数名	SSI_TWI_Cmd
函数原型	void SSI_TWI_Cmd(FunctionalState NewState)
功能描述	TWI 功能开关函数
输入参数	无 NewState 使能或关闭, 可取值 ENABLE 或 DISABLE
返回值	无

(10) SSI_TWI_SendData 函数

函数名	SSI_TWI_SendData
函数原型	void SSI_TWI_SendData(uint8_t Data)
功能描述	TWI 发送数据
输入参数	Data 发送的数据
返回值	无

(11) SSI_TWI_ReceiveData 函数

函数名	SSI_TWI_ReceiveData
函数原型	uint8_t SSI_TWI_ReceiveData(void)
功能描述	获得 SSDAT 中的值
输入参数	无
返回值	uint8_t SSDAT 中的值

(12) SSI_UART1_Init 函数

函数名	SSI_UART1_Init
函数原型	void SSI_UART1_Init(uint32_t UART1Fsys, uint32_t BaudRate, UART1_Mode_TypeDef Mode, UART1_RX_TypeDef RxMode)
功能描述	UART1 初始化配置函数
输入参数 1	UART1Fsys 系统时钟频率
输入参数 2	BaudRate 波特率
输入参数 3	Mode UART1 工作模式
输入参数 4	RxMode 接收允许选择
返回值	无

Mode 可选值

Mode	描述
UART1_Mode_10B	UART1 选为 10 位模式
UART1_Mode_11B	UART1 选为 11 位模式

RxMode 可选值

RxMode	描述
UART1_RX_ENABLE	允许 UART1 接收
UART1_RX_DISABLE	禁止 UART1 接收

(13) SSI_UART1_SendData8 函数

函数名	SSI_UART1_SendData8
函数原型	void SSI_UART1_SendData8(uint8_t Data)
功能描述	UART1 发送 8 位数据
输入参数	Data 发送的数据
返回值	无

(14) SSI_UART1_ReceiveData8 函数

函数名	SSI_UART1_ReceiveData8
函数原型	uint8_t SSI_UART1_ReceiveData8(void)
功能描述	获得 SSDAT 中的值
输入参数	无
返回值	uint8_t SSDAT 中的值

(15) SSI_UART1_SendData9 函数

函数名	SSI_UART1_SendData9
函数原型	void SSI_UART1_SendData9(uint16_t Data)
功能描述	UART1 发送 9 位数据
输入参数	Data 发送的数据
返回值	无

(16) SSI_UART1_ReceiveData9 函数

函数名	SSI_UART1_ReceiveData9
函数原型	uint16_t SSI_UART1_ReceiveData9(void)
功能描述	获得 SSDAT 中的值及第九位的值
输入参数	无
返回值	uint16_t SSDAT 中的值及第九位的值

(17) SSI_ITConfig 函数

函数名	SSI_ITConfig
函数原型	void SSI_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	SSI 中断初始化
输入参数 1	NewState 使能或关闭，可取值 ENABLE 或 DISABLE
输入参数 2	Priority 优先级，可取值 LOW 或 HIGH
返回值	无

(18) SSI_GetFlagStatus 函数

函数名	SSI_GetFlagStatus
函数原型	FlagStatus SSI_GetFlagStatus(SSSI_Flag_TypeDef SSI_FLAG)
功能描述	获得 SSI 中断标志状态
输入参数	SSI_FLAG 所需获取的标志位
返回值	FlagStatus SSI 中断标志状态

SSI_FLAG 可选值

SSI_FLAG	描述
SPI_FLAG_SPIF	SPI 数据传送标志位 SPIF
SPI_FLAG_WCOL	SPI 写入冲突标志位 WCOL
SPI_FLAG_TXE	SPI 发送缓存器空标志 TXE
TWI_FLAG_TWIF	TWI 中断标志位 TWIF
TWI_FLAG_GCA	TWI 通用地址响应标志位 GCA
UART1_FLAG_TI	UART1 发送中断标志位 TI
UART1_FLAG_RI	UART1 接收中断标志位 RI

(19) SSI_ClearFlag 函数

函数名	SSI_ClearFlag
函数原型	void SSI_ClearFlag(SSSI_Flag_TypeDef SSI_FLAG)
功能描述	清除 SSI 标志状态
输入参数	SSI_FLAG 所需清除的标志位
返回值	无

(20) SSI_PinSelection 函数

函数名	SSI_PinSelection
函数原型	void SSI_PinSelection(SSSI_PinSelection_TypeDef PinSeletion)
功能描述	SSI 引脚选择
输入参数	PinSeletion 选择 SSI 引脚为 P10P27P26 或 P21P22P23
返回值	无

PinSeletion 可选值

PinSeletion	描述
SSI_PinSelection_P10P27P26	SSI 共用引脚为 P10, P27, P26
SSI_PinSelection_P21P22P23	SSI 共用引脚为 P21, P22, P23

十、IAP 固件库函数

SC92F8003 的 IAP 操作空间范围有两种模式可选：

- (1) 128Byte EEPROM 可以作为数据存储使用
- (2) IC 的 Code 区域（范围可选）及 128Byte EEPROM 内都可进行 In Application Programming(IAP)操作，主要用作远程程序更新使用。

IAP 操作模式选择作为 Code Option 在编程器写入 IC 时选择。

1、IAP 固件库函数列表

函数名	描述
IAP_DelInit	IAP 相关寄存器复位至缺省值
IAP_SetHoldTime	IAP 操作 CPU Hold Time 配置函数
IAP_ProgramByte	IAP 写入一个字节
IAP_ReadByte	IAP 读一个字节

2、IAP 固件库函数详解

(1) 函数 IAP_DelInit

函数名	IAP_DelInit
函数原型	void IAP_DelInit(void)
功能描述	IAP 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) 函数 IAP_SetHoldTime

函数名	IAP_SetHoldTime
函数原型	void IAP_SetHoldTime(IAP_HoldTime_TypeDef IAP_HoldTime)
功能描述	IAP 操作 CPU Hold Time 配置函数
输入参数	IAP_HoldTime CPU Hold Time 选择
返回值	无

IAP_HoldTime 可选值

IAP_HoldTime	描述
IAP_HOLDTIME_4MS	设定 CPU Hold Time 为 4MS
IAP_HOLDTIME_2MS	设定 CPU Hold Time 为 2MS
IAP_HOLDTIME_1MS	设定 CPU Hold Time 为 1MS

(3) 函数 IAP_ProgramByte

函数名	IAP_ProgramByte
函数原型	void IAP_ProgramByte(uint16_t Address, uint8_t Data, IAP_MemType_TypeDef IAP_MemType, uint8_t WriteTimeLimit)
功能描述	IAP 写入一个字节
输入参数 1	Address IAP 操作地址
输入参数 2	Data 写入的数据
输入参数 3	IAP_MemType IAP 操作对象 (ROM、IFB、EEPROM)
输入参数 4	WriteTimeLimit IAP 操作时限 (非零值)
返回值	无

IAP_MemType 可选值

IAP_MemType	描述
-------------	----

IAP_MEMTYPE_ROM	IAP 操作区域为 ROM
IAP_MEMTYPE_IFB	IAP 操作区域为 IFB
IAP_MEMTYPE_EEPROM	IAP 操作区域为 EEPROM

(4) 函数 IAP_ReadByte

函数名	IAP_ReadByte
函数原型	uint8_t IAP_ReadByte(uint16_t Address,IAP_MemType_TypeDef IAP_MemType)
功能描述	IAP 读一个字节
输入参数 1	Address IAP 操作地址
输入参数 2	IAP_MemType IAP 操作对象 (ROM、IFB、EEPROM)
返回值	uint8 读到的字节数据

(5) 函数 IAP_SetOperateRange

函数名	IAP_SetOperateRange
函数原型	void IAP_SetOperateRange(IAP_OperateRange_TypeDef IAP_OperateRange)
功能描述	允许 IAP 操作的范围设置
输入参数 1	IAP_OperateRange IAP 操作范围
返回值	无

十一、WDT 固件库函数

SC92F8003 有一个看门狗 WDT，其时钟源为内部的 128kHz 振荡器。

1、WDT 固件库函数列表

函数名	描述
WDT_Delnit	WDT 相关寄存器复位至缺省值
WDT_Init	WDT 初始化配置函数
WDT_SetReload	WDT 喂狗

2、WDT 固件库函数详解

(1) 函数 WDT_Init

函数名	WDT_Delnit
函数原型	void WDT_Delnit(void)
功能描述	WDT 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) 函数 WDT_Init

函数名	WDT_Init
函数原型	void WDT_Init(WDT_OverflowTime_TypeDef OverflowTime)
功能描述	WDT 初始化配置函数
输入参数	OverflowTime WDT 溢出时间选择
返回值	无

OverflowTime 可选值

OverflowTime	描述
WDT_OverflowTime_500MS	看门狗溢出时间为 500MS
WDT_OverflowTime_250MS	看门狗溢出时间为 250MS
WDT_OverflowTime_125MS	看门狗溢出时间为 125MS

WDT_OverflowTime_62_5MS	看门狗溢出时间为 62.5MS
WDT_OverflowTime_31_5MS	看门狗溢出时间为 31.5MS
WDT_OverflowTime_15_75MS	看门狗溢出时间为 15.75MS
WDT_OverflowTime_7_88MS	看门狗溢出时间为 7.88MS
WDT_OverflowTime_3_94MS	看门狗溢出时间为 3.94MS

(3) 函数 WDT_SetReload

函数名	WDT_SetReload
函数原型	void WDT_SetReload(void)
功能描述	看门狗喂狗，计数值清 0
输入参数	无
返回值	无

十二、BTM 固件库函数

SC92F8003 内建一个频率为 128kHz 的 RC 振荡电路，作为低频时钟定时器 Base Timer 的时钟源。该振荡器直接连接一个 Base Timer，可以把 CPU 从 STOP mode 唤醒，并且产生中断。

1、BTM 固件库函数列表

函数名	描述
BTM_DelInit	BTM 相关寄存器复位至缺省值
BTM_Init	BTM 初始化
BTM_Cmd	使能或者使能 BTM
BTM_ITConfig	使能或者失能 BTM 中断
BTM_GetFlagStatus	获得 BTM 中断标志状态
BTM_ClearFlag	清除 BTM 中断标志位

2、BTM 固件库函数详解

(1) 函数 BTM_DelInit

函数名	BTM_DelInit
函数原型	void BTM_DelInit(void)
功能描述	BTM 初始化
输入参数	无
返回值	无

(2) 函数 BTM_Init

函数名	BTM_Init
函数原型	void BTM_Init(BTM_Timebase_TypeDef BTM_Timebase)
功能描述	低频时钟定时器初始化
输入参数	BTM_TimeBase BTM 中断时间选择
返回值	无

BTM_TimeBase 可选值

BTM_TimeBase	描述
BTM_TIMEBASE_15625US	低频时钟中断时间为 15.625MS
BTM_TIMEBASE_31250US	低频时钟中断时间为 31.25MS
BTM_TIMEBASE_62500US	低频时钟中断时间为 62.5MS
BTM_TIMEBASE_125MS	低频时钟中断时间为 125MS
BTM_TIMEBASE_250MS	低频时钟中断时间为 250MS

BTM_TIMEBASE_500MS	低频时钟中断时间为 500MS
BTM_TIMEBASE_1S	低频时钟中断时间为 1S
BTM_TIMEBASE_2S	低频时钟中断时间为 2S
BTM_TIMEBASE_4S	低频时钟中断时间为 4S

(3) 函数 BTM_Cmd

函数名	BTM_Cmd
函数原型	void BTM_Cmd(FunctionalState NewState)
功能描述	使能或者使能 BTM
输入参数	NewState BTM 使能或失能，可取值 ENABLE 或 DISABLE
返回值	无

(4) 函数 BTM_ITConfig

函数名	BTM_ITConfig
函数原型	void BTM_ITConfig(FunctionalState NewState, PriorityStatus Priority)
功能描述	使能或者失能 BTM 中断
输入参数 1	NewState BTM 中断使能或失能，可取值 ENABLE 或 DISABLE
输入参数 2	Priority BTM 中断优先级设置，可取值 HIGH 或 LOW
返回值	无

(5) 函数 BTM_GetFlagStatus

函数名	BTM_GetFlagStatus
函数原型	FlagStatus BTM_GetFlagStatus(void)
功能描述	获得 BTM 中断标志状态
输入参数	无
返回值	无

(6) 函数 BTM_ClearFlag

函数名	BTM_ClearFlag
函数原型	void BTM_ClearFlag(void)
功能描述	清除 BTM 中断标志位
输入参数	无
返回值	无

十三、PWR 固件库函数

SC92F8003 提供了一个特殊功能寄存器 PCON。配置该寄存器的 bit0 和 bit1 可控制 MCU 进入不同的工作模式。

1、PWR 固件库函数列表

函数名	描述
PWR_DeInit	PWR 相关寄存器复位至缺省值
PWR_EnterSTOPMode	MCU 进入 STOP 模式
PWR_EnterIDLEMode	MCU 进入 Idle 模式

2、PWR 固件库函数详解

(1) 函数 PWR_DeInit

函数名	PWR_DeInit
函数原型	void PWR_DeInit(void)

功能描述	MCU 进入 STOP 模式
输入参数	无
返回值	无

(2) 函数 PWR_EnterSTOPMode

函数名	PWR_EnterSTOPMode
函数原型	void PWR_EnterSTOPMode(void)
功能描述	MCU 进入 STOP 模式
输入参数	无
返回值	无

(3) 函数 PWR_EnterIDLEMode

函数名	PWR_EnterIDLEMode
函数原型	void PWR_EnterIDLEMode(void)
功能描述	MCU 进入 Idle 模式
输入参数	无
返回值	无

十四、CHKSUM 固件库函数

SC92F8003 内建了一个 check sum 校验模块，可用来实时生成程序代码的 16 位 check sum 值，用户可利用此 check sum 和理论值比较，监测程序区的内容是否正确。

1、CHKSUM 固件库函数列表

函数名	描述
CHKSUM_DeInit	CHKSUM 相关寄存器复位至缺省值
CHKSUM_StartOperation	触发一次 check sum 计算
CHKSUM_GetCheckValue	获取一次 check sum 计算值

2、CHKSUM 固件库函数详解

(1) 函数 CHKSUM_DeInit

函数名	CHKSUM_DeInit
函数原型	void CHKSUM_DeInit(void)
功能描述	CHKSUM 相关寄存器复位至缺省值
输入参数	无
返回值	无

(2) 函数 CHKSUM_StartOperation

函数名	CHKSUM_StartOperation
函数原型	void CHKSUM_StartOperation(void)
功能描述	触发一次 check sum 计算
输入参数	无
返回值	无

(3) 函数 CHKSUM_GetCheckValue

函数名	CHKSUM_GetCheckValue
函数原型	uint16_t CHKSUM_GetCheckValue(void)

功能描述	获取一次 check sum 计算值
输入参数	无
返回值	uint16_t check sum 计算值

十五、OPTION 固件库函数

SC92F8003 内部有单独的一块 Flash 区域用于保存客户的上电初始值设置，此区域称为 Customer Option 区域。用户在烧写 IC 事将此部分代码写入 IC 内部，IC 在复位初始化时，就会将此设置调入 SFR 作为初始设置。

对 OPTION 的操作可以对系统时钟、LVR、看门狗、外部晶振、复位管脚、ADC 参考电压等设置项做出改变。

1、OPTION 固件库函数列表

函数名	描述
OPTION_WDT_Cmd	WDT 功能开关函数
OPTION_XTIPLL_Cmd	外部高频晶振使能
OPTION_XTIPLL_SetRange	外部高频晶振频率范围
OPTION_SYSCLK_Init	系统时钟分频初始化
OPTION_RST_PIN_Cmd	外部复位管脚使能
OPTION_LVR_Init	LVR 电压选择
OPTION_ADC_VrefConfig	ADC 参考电压选择
OPTION_IAP_SetOperateRange	允许 IAP 操作的范围设置

2、OPTION 固件库函数详解

(1) 函数 OPTION_WDT_Cmd

函数名	OPTION_WDT_Cmd
函数原型	void OPTION_WDT_Cmd(FunctionalState NewState)
功能描述	WDT 功能开关函数
输入参数	NewState 功能启动/关闭选择
返回值	无

(2) 函数 OPTION_XTIPLL_Cmd

函数名	OPTION_XTIPLL_Cmd
函数原型	void OPTION_XTIPLL_Cmd(FunctionalState NewState)
功能描述	外部高频晶振使能
输入参数	NewState 功能启动/关闭选择
返回值	无

(3) 函数 OPTION_XTIPLL_SetRange

函数名	OPTION_WDT_Cmd
函数原型	void OPTION_XTIPLL_SetRange(XTIPLL_Range_TypeDef XTIPLL_Range)
功能描述	外部高频晶振频率范围选择
输入参数	XTIPLL_Range 外部晶振频率选择
返回值	无

XTIPLL_Range 可选值

XTIPLL_Range	描述
XTIPLL_HIGHER_THAN_12M	外接晶振振荡频率大于等于 12M
XTIPLL_UNDER_12M	外接晶振振荡频率小于 12M

(4) 函数 OPTION_SYSClk_Init

函数名	OPTION_SYSClk_Init
函数原型	void OPTION_SYSClk_Init(SYSCLK_PresSel_TypeDef SYSCLK_PresSel)
功能描述	系统时钟分频初始化
输入参数	SYSCLK_PresSel 选择系统时钟分频
返回值	无

SYSCLK_PresSel 可选值

SYSCLK_PresSel	描述
SYSCLK_PRESSEL_FOSC_D1	预分频 $F_{sys} = F_{osc}/1$
SYSCLK_PRESSEL_FOSC_D2	预分频 $F_{sys} = F_{osc}/2$
SYSCLK_PRESSEL_FOSC_D4	预分频 $F_{sys} = F_{osc}/4$
SYSCLK_PRESSEL_FOSC_D12	预分频 $F_{sys} = F_{osc}/12$

(5) 函数 OPTION_RST_PIN_Cmd

函数名	OPTION_RST_PIN_Cmd
函数原型	void OPTION_RST_PIN_Cmd(FunctionalState NewState)
功能描述	外部复位管脚 (P17) 使能
输入参数	NewState 使能/关闭选择
返回值	无

(6) 函数 OPTION_LVR_Init

函数名	OPTION_LVR_Init
函数原型	void OPTION_LVR_Init(LVR_Config_TypeDef LVR_Config)
功能描述	LVR 电压选择
输入参数	LVR_Config 选择 LVR 电压
返回值	无

LVR_Config 可选值

LVR_Config	描述
LVR_INVALID	LVR 无效
LVR_2_3V	LVR 2.3V 复位
LVR_2_9V	LVR 2.9V 复位
LVR_3_7V	LVR 3.7V 复位
LVR_4_3V	LVR 4.3V 复位

(7) 函数 OPTION_ADC_VrefConfig

函数名	OPTION_ADC_VrefConfig
函数原型	void OPTION_ADC_VrefConfig(ADC_Vref_TypeDef ADC_Vref)
功能描述	ADC 参考电压选择
输入参数	ADC_Vref 选择 ADC 参考电压
返回值	无

ADC_Vref 可选值

ADC_Vref	描述
ADC_VREF_VDD	选择 VDD 做 ADC 参考电压
ADC_VREF_2_4V	选择内部 2.4V 做 ADC 参考电压

(8) 函数 OPTION_IAP_SetOperateRange

函数名	OPTION_IAP_SetOperateRange
-----	----------------------------

函数原型	void IAP_OperateRange) OPTION_IAP_SetOperateRange(IAP_OperateRange_TypeDef
功能描述	允许 IAP 操作的范围设置
输入参数	IAP_OperateRange IAP 操作范围
返回值	无

IAP_OperateRange 可选值

IAP_OperateRange	描述
IAP_OPERATERANGE_ONLY_EEPROM	只允许 EEPROM 进行 IAP 操作
IAP_OPERATERANGE__LAST_0_5K_CODEREGION	允许 ROM 最后 0.5k 和 EEPROM 进行 IAP 操作
IAP_OPERATERANGE__LAST_1K_CODEREGION	允许 ROM 最后 1k 和 EEPROM 进行 IAP 操作
IAP_OPERATERANGE__ALL_CODEREGION	允许 ROM 和 EEPROM 所有区域进行 IAP 操作

规格更改记录

版本	记录	日期
V0.1	初版	2019 年 1 月
V0.2	EXTI/PWM B 部分入参类型与库不一致	2019 年 1 月