GPIO 模块使用说明文档 V1.0

目录

1 通用和复用 GPIO 4
1.1 内部结构示意图 4
1.2 功能描述 5
1.2.1 数字输入功能 6
1.2.2 数字输出功能 6
1.2.3 模拟输入功能 6
1.2.4 中断功能 7
1.2.5 防倒灌 GPIO 7
1.2.6 复用功能 7
1.3 接口函数描述 8
1.3.1 宏定义描述 8
1.3.2 GpioSpimIoConfig 9
1.3.3 GpioFuartRxIoConfig 10
1.3.4 GpioFuartTxIoConfig 11
1.3.5 GpioBuartRxIoConfig 11
1.3.6 GpioBuartTxIoConfig 12
1.3.7 GpioBuartCtsIoConfig 13
1.3.8 GpioBuartRtsIoConfig 14
1.3.9 GpioSdIoConfig 14
1.3.10 GpioLcdIoConfig 15
1.3.11 GpioPwcIoConfig 15
1.3.12 GpioIrIoConfig 16
1.3.13 GpioFshcIoConfig 17
1.3.14 GpioSdramIoConfig 17
1.3.15 GpioClk32kIoConfig 18
1.3.16 GpioClk12m6mIoConfig 18
1.3.17 GpioPcmSyncIoConfig 19

1.3.18 GpioI2sIoConfig 20
1.3.19 GpioMclkIoConfig 20
1.3.20 GpioSwIoConfig 21
1.3.21 GpioSpisIoConfig 21
1.3.22 GpioCtsCmpoutIoConfig 22
1.3.23 GpioSdSpiSel 22
1.3.24 GpioSetReg 22
1.3.25 GpioGetReg 23
1.3.26 GpioSetRegOneBit 23
1.3.27 GpioClrRegOneBit 23
1.3.28 GpioSetRegBits 24
1.3.29 GpioClrRegBits 24
1.3.30 GpioIntEn 25
1.3.31 GpioIntDis 25
1.3.32 GpioIntClr 25
1.3.33 GpioIntFlagGet 26
1.3.34 GpioA0SetIcs 28
1.3.35 GpioA0SetMode 29
1.3.36 GpioPorSysReset 26
1.4 使用流程举例 27

GPIO 模块说明

1 通用和复用 GPIO

芯片提供了三组 GPIO，分别为 GPIOA、GPIOB、GPIOC。这三组 GPIO 除了作为普通 GPIO 外，还可复用为其他功能，例如 FUART、SPI、PWM 等。具体复用关系请参考对应芯片的数据手册。

1.1 内部结构示意图

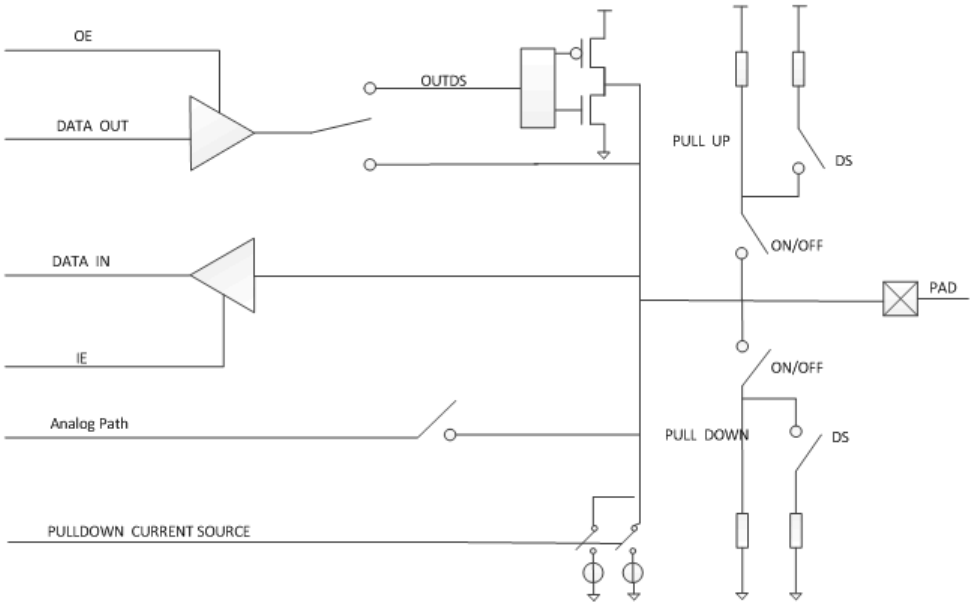

图 1.1 通用 GPIO 内部结构示意图

每个 GPIO 端口都是有几个寄存器来控制，故可以通过软件配置成多种模式。图 1.1 为 GPIO 内部结构示意图，OE 为输出使能寄存器，DATA OUT 为输出数据寄存器，IE 为输入使能寄存器，DATA IN 为输入数据寄存器，Analog Path 为模拟通道，PULLDOWN CURRENT SOURCE 为下拉电流源寄存器，PULL UP/PULL DOWN 为上拉/下拉寄存器；DS 为配置上下拉的强弱寄存器，OUTDS 为输出驱动能力设置寄存器，PAD 为封装出的引脚。

表格 1.1 通用 GPIO 配置表，表格 1.2 通用 GPIO 上下拉配置表，表格 1.3 下拉电流源输出电流值表[注解 3]。

表格 1.1 通用 GPIO 配置表

配置模式	状态	OE	IE	OUTDS	DS	PU	PD

数字输出	强驱动输出注解 1	1	0	1	*	*	*
	弱驱动输出	1	0	0	*	*	*
	强上/下拉输出注解 2	1	0	*	1	*	*
	弱上/下拉输出	1	0	*	0	*	*
数字输入	强上/下拉输入	0	1	0	0	*	*
	弱上/下拉输入	0	1	0	1	*	*
模拟输入	模拟输入	0	0	*	*	0	1

注解 1： OUTDS=0 : 8mA (VOL=0.4V, VOH=2.4V)； OUTDS=1 : 24mA (VOL=0.4V, VOH=2.4V)

注解 2： DS = 1： 70uA； DS = 0：20uA

表格 1.2 通用 GPIO 上下拉配置表

配置功能	PU	PD
上拉	0	0
模拟通道	0	1
无上拉无下拉	1	0
下拉	1	1

表格 1.3 下拉电流源输出电流值表注解 3

下拉电流源电流值(mA)	PULLDOWN1	PULLDOWN2
0	0	0
1.7	1	0
2.4	0	1
4.1	1	1

注解 3： 测试条件：PU = 1，PD = 0，IE = 1，OE = 0，DS = 0，OUTDS = 0

1.2 功能描述

GPIO 可以配置成数字输入/输出功能、模拟输入功能、中断功能以及复用功能。

表格 1.4 GPIO 默认上电状态

引脚名称	引脚类型	默认上下拉状态	默认电平
GPIOA[1：0]	防倒灌 GPIO	Pull-Down	L
GPIOA[10：2]	普通 GPIO	Pull-Up	H
GPIOA[12：11]	防倒灌 GPIO	Pull-Down	L
GPIOA[21：13]	普通 GPIO	Pull-up	H
GPIOA[25：22]	普通 GPIO	No pull up/No pull down	L

GPIOA[31：26]	普通 GPIO	Pull-up	H
GPIOB[31:0]	普通 GPIO	Pull-up	H
GPIOC[12：0]	普通 GPIO	Pull-up	H
GPIOC13	普通 GPIO	Pull-Down	L
GPIOC14	普通 GPIO	Pull-up	H

1.2.1 数字输入功能

当 GPIO 作为数字输入引脚使用时:

- 可以配置成上拉、下拉或无上下拉状态。
- 可配置成强拉状态、弱拉状态。

在配置 GPIO 为上/下拉状态同时，可根据实际需要配成 70uA 的强拉状态或 20uA 弱上/下拉状态。芯片上电默认状态为 20uA 的弱拉状态。

- 可以读取每一个通用 GPIO 数字输入状态

1.2.2 数字输出功能

当 GPIO 作为数字输出引脚使用时:

- 可以配置成高电平输出或低电平输出状态。
- 可以配置成上拉、下拉或无上下拉状态。
- 可配置成强拉状态、弱拉状态。

在配置 GPIO 为上/下拉状态同时，可根据实际需要配成 70uA 的强拉状态或 20uA 弱上/下拉状态。芯片上电默认状态为 20uA 的弱拉状态。

- 可配置成强输出状态、弱输出状态。

在配置 GPIO 作为输出引脚的时候，可以根据实际应用情况，将 GPIO 配置成强驱动输出状态或弱驱动输出状态。强驱动输出状态下 GPIO 引脚可以输出高达 24mA 驱动电流，弱驱动输出状态下 GPIO 引脚可输出 8mA 驱动电流，芯片上电默认状态为弱驱动电流输出。

1.2.3 模拟输入功能

当 GPIO 作为模拟输入引脚使用时:

- 不可配置上/下拉。
- 可以被 ADC 模块使用。
- 可以被 USB PHY 模块使用。

- 可以被 LCD 段码屏模块使用。
- 可以被 Touch Key In 使用。
- 可以被 VCOM 使用。
- 可以被 FM IN 使用。

1.2.4 中断功能

三组 GPIO 都可以作为 GPIO 中断使用，当实际情况需要利用 GPIO 中断时，需要首先配置 GPIO 为数字输入状态。

- 可以使能 GPIO 中断触发或禁止 GPIO 中断触发。
- 触发极性分为上升沿触发中断和下降沿触发中断。

  芯片上电默认触发极性为下降沿触发，可根据实际情况进行配置。
- 可读取每一个 GPIO 引脚的中断状态。

  可以读取每一个 GPIO 的中断状态，但是中断状态标志需要通过软件进行清除。

1.2.5 防倒灌 GPIO

芯片提供了 4 个防倒灌 GPIO，分别为 GPIOA[1:0]，GPIOA[12:11]。防倒灌 GPIO 是指在系统断电情况下，外部有电压时不会有电流通过这些引脚流入芯片内部。该引脚可以用于一些需要较高电压场应用。

在芯片工作时，防倒灌 GPIO 不能直接作为 IO 使用。如果需要将其作为 GPIO 使用前，想就要将防倒灌的 Charge Pump 模块关闭。一般系统在上电之后会自动关闭该功能。

1.2.6 复用功能

GPIO 除了具有通过 GPIO 功能之前，还可以将其映射成其他功能。可根据实际功能需要进行对应引脚的映射。芯片上电后除了 SW 对应的 GPIO 引脚被映射成 SW 之外，其余 GPIO 均默认进入通用模式。当 GPIO 可映射复用模式有:

- 可以映射为 4 组 SPIM 接口引脚。
- 可以映射为 3 组 Fuart 接收/发送接口引脚。
- 可以映射成 4 组 Buart 接收/发送接口引脚。
- 可以映射成 3 组 Buart Rts 接口引脚。
- 可以映射成 3 组 Buart Cts 接口引脚。
- 可以映射成 4 组 SD 接口引脚。

- 可以映射成 1 组 LCD 接口引脚。
- 可以映射成 2 组 PWC 引脚。
- 可以映射成 3 组 Ir 引脚。
- 可以映射成 2 组 Fshc 接口引脚。
- 可以映射成 1 组 Sdram 接口引脚。
- 可以映射成 3 组 32KHz Clk 输出引脚。
- 可以映射成 3 组 12MHz/16MHz Clk 输出引脚。
- 可以映射成 3 组 PCM 接口引脚。
- 可以映射成 3 组 I2S 接口引脚。
- 可以映射成 4 组 Mclk 引脚。
- 可以映射成 1 组 SW 接口引脚。
- 可以映射成 1 组 SPIS 接口引脚。
- 可以映射成 1 组 CTS（Capacitor touch sensor）比较信号引脚。

在映射为其他功能之后，如果需要重新映射回通用 GPIO 功能，需要禁能该引脚的复用映射，否则 GPIO 无法直接使用。另外 GPIO 可能允许被多个模块复用，所以必须保证同一时刻同一个 GPIO 只能作为一个功能使用。

1.3 接口函数描述

1.3.1 宏定义描述

在 gpio.h 文件中定义了所有 GPIO 接口函数需要用到的宏定义或类型定义，现在以 GPIOA 做举例说明。

- GPIO_A_IN：GPIOA 输入数据寄存器索引。
- GPIO_A_OUT：GPIOA 输出数据寄存器索引。
- GPIO_A_IE：GPIOA 输入使能寄存器索引。
- GPIO_A_OE：GPIOA 输出使能寄存器索引。
- GPIO_A_DS：GPIOA 上/下拉能力寄存器索引。
- GPIO_A_OUTDS：GPIOA 输出驱动能力寄存器索引。
- GPIO_A_PU：GPIOA 上拉寄存器索引。
- GPIO_A_PD：GPIOA 下拉寄存器索引。

- GPIO_A_PULLDOWN1/ GPIO_A_PULLDOWN2：下拉电流源大小设置寄存器索引。
- GPIOA1：和 GPIOA 端口寄存器一块使用，表示该寄存器的 BIT1 位。
- GPIO_A_INT：GPIOA 中断寄存器索引。
- GPIO_NEG_EDGE_TRIGGER：GPIO 中断触发方式为下降沿触发。
- GPIO_POS_EDGE_TRIGGER：GPIO 中断触发方式为上升沿触发。
- #define SET_C11_ANALOG_IN() GpioClrRegBits(GPIO_C_PU, GPIOC11),\
  GpioSetRegBits(GPIO_C_PD, GPIOC11),\
  GpioClrRegBits(GPIO_C_IE, GPIOC11),\
  GpioClrRegBits(GPIO_C_OE, GPIOC11)

  配置 GPIOC11 为模拟输入
- #define SET_C12_ANALOG_IN() GpioClrRegBits(GPIO_C_PU, GPIOC12),\
  GpioSetRegBits(GPIO_C_PD, GPIOC12),\
  GpioClrRegBits(GPIO_C_IE, GPIOC12),\
  GpioClrRegBits(GPIO_C_OE, GPIOC12)

  配置 GPIOC12 为模拟输入
- #define SET_C13_ANALOG_IN() GpioClrRegBits(GPIO_C_PU, GPIOC13),\
  GpioSetRegBits(GPIO_C_PD, GPIOC13),\
  GpioClrRegBits(GPIO_C_IE, GPIOC13),\
  GpioClrRegBits(GPIO_C_OE, GPIOC13)

  配置 GPIOC13 为模拟输入
- #define SET_C14_ANALOG_IN() GpioClrRegBits(GPIO_C_PU, GPIOC14),\
  GpioSetRegBits(GPIO_C_PD, GPIOC14),\
  GpioClrRegBits(GPIO_C_IE, GPIOC14),\
  GpioClrRegBits(GPIO_C_OE, GPIOC14)

  配置 GPIOC14 为模拟输入

1.3.2 GpioSpimIoConfig

原型：

void GpioSpimIoConfig(uint8_t ModeSel)

参数：

ModeSel:GPIO 与 SPI Master 复用关系选择。其中参数值 0~3 为复用为 Spi Master 引脚，0xff 为还原复用的 GPIO 为通用 GPIO，对应关系如表格 1.5 GPIO 与 SPI Master 复用关系表所示。

表格 1.5 GPIO 与 SPI Master 复用关系表

ModeSel	MOSI	CLK	MISO	功能
0	GPIOA2	GPIOA3	GPIOA4	复用为 Spi Mster
1	GPIOA21	GPIOA20	GPIOA19	复用为 Spi Mster
2	GPIOB5	GPIOB4	GPIOB3	复用为 Spi Mster
3	GPIOB20	GPIOB21	GPIOB22	复用为 Spi Mster
0xff	还原 SPI Master 引脚为通用 GPIO 引脚			

描述：

该函数用于将 GPIO 配置成 Spi Master 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~3，来选择哪一组 GPIO 为 Spi Master 接口引脚；往 ModelSel 中写入 0xff 来还原 Spi Master 引脚为通用 GPIO 引脚。

返回值：

无

1.3.3 GpioFuartRxIoConfig

原型：

void GpioFuartRxIoConfig(uint8_t ModeSel)

参数：

ModeSel：GPIO 与 Fuart Rx 引脚的复用关系。其中参数 0~2 为复用为 FuartRx 引脚，0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.6 GPIO 与 FuartRx 引脚复用关系所示。

表格 1.6 GPIO 与 FuartRx 引脚复用关系

ModeSel	FuartRx	功能
0	GPIOA1	复用为 FuartRx 引脚
1	GPIOB6	复用为 FuartRx 引脚
2	GPIOC4	复用为 FuartRx 引脚
0xff	还原 FuartRx 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 FuartRx 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 FuartRx 接口引脚；通往 ModelSel 中写入 0xff 来还原 FuartRx 引脚为通用 GPIO 引脚。

返回值:

无

1.3.4 GpioFuartTxIoConfig

原型:

void GpioFuartTxIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 Fuart Tx 引脚的复用关系。其中参数 0~2 为复用为 FuartTx 引脚, 0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.7 GPIO 与 FuartTx 复用关系表所示。

表格 1.7 GPIO 与 FuartTx 复用关系表

ModeSel	FuartTx	功能
0	GPIOA0	复用为 FuartTx 引脚
1	GPIOB7	复用为 FuartTx 引脚
2	GPIOC3	复用为 FuartTx 引脚
0xff	还原 FuartTx 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 FuartTx 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 FuartTx 接口引脚；通往 ModelSel 中写入 0xff 来还原 FuartTx 引脚为通用 GPIO 引脚。

返回值:

无

1.3.5 GpioBuartRxIoConfig

原型:

void GpioBuartRxIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 BuartRx 引脚的复用关系。其中参数 0~3 为复用为 BuartRx 引脚, 0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.8 GPIO 与 Buart Rx 引脚复用关系所示。

表格 1.8 GPIO 与 Buart Rx 引脚复用关系

ModeSel	BuartRx	功能
0	GPIOA13	复用为 BuartRx 引脚
1	GPIOA24	复用为 BuartRx 引脚
2	GPIOB8	复用为 BuartRx 引脚
3	GPIOB29	复用为 BuartRx 引脚
0xff	还原 BuartRx 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 BuartRx 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~3，来选择哪一个 GPIO 配置为 BuartRx 接口引脚；通往 ModelSel 中写入 0xff 来还原 BuartRx 引脚为通用 GPIO 引脚

返回值:

无

1.3.6 GpioBuartTxIoConfig

原型:

void GpioBuartTxIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 BuartTx 引脚的复用关系。其中参数 0~3 为复用为 BuartTx 引脚, 0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.8 GPIO 与 Buart Rx 引脚复用关系所示。

表格 1.9 GPIO 与 Buart Tx 引脚复用关系

ModeSel	BuartTx	功能
0	GPIOA16	复用为 BuartTx 引脚
1	GPIOA25	复用为 BuartTx 引脚
2	GPIOB9	复用为 BuartTx 引脚

3	GPIOB28	复用为 BuartTx 引脚
0xff	还原 BuartTx 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 BuartTx 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~3，来选择哪一个 GPIO 配置为 BuartTx 接口引脚；通往 ModelSel 中写入 0xff 来还原 BuartTx 引脚为通用 GPIO 引脚

返回值:

无

1.3.7 GpioBuartCtsIoConfig

原型:

void GpioBuartCtsIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 BuartCts 引脚的复用关系。其中参数 0~2 为复用为 BuartCts 引脚，0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.10 GPIO 与 BuartCts 引脚复用关系所示。

表格 1.10 GPIO 与 BuartCts 引脚复用关系

ModeSel	**BuartCts**	**功能**
0	GPIOA11	复用为 BuartCts 引脚
1	GPIOB30	复用为 BuartCts 引脚
2	GPIOC0	复用为 BuartCts 引脚
0xff	还原 BuartCts 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 BuartCts 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 BuartCts 接口引脚；通往 ModelSel 中写入 0xff 来还原 BuartCts 引脚为通用 GPIO 引脚

返回值:

无

1.3.8 GpioBuartRtsIoConfig

原型:

void GpioBuartRtsIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 BuartRts 引脚的复用关系。其中参数 0~2 为复用为 BuartRts 引脚，0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.11 GPIO 与 BuartRts 引脚复用关系所示。

表格 1.11 GPIO 与 BuartRts 引脚复用关系

ModeSel	BuartRts	功能
0	GPIOA12	复用为 BuartRts 引脚
1	GPIOB31	复用为 BuartRts 引脚
2	GPIOC1	复用为 BuartRts 引脚
0xff	还原 BuartRts 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 BuartRts 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 BuartRts 接口引脚；通往 ModelSel 中写入 0xff 来还原 BuartRts 引脚为通用 GPIO 引脚

返回值:

无

1.3.9 GpioSdIoConfig

原型:

void GpioSdIoConfig(uint8_t ModeSel)

参数:

ModeSel:GPIO 与 SD 复用关系选择。其中参数值 0~3 为复用为 SD 引脚，0xff 为还原复用的 GPIO 为通用 GPIO，对应关系如表格 1.12 GPIO 与 SD 复用关系表所示。

表格 1.12 GPIO 与 SD 复用关系表

ModeSel	SD_CMD	SD_CLK	SD_DAT	功能
0	GPIOA2	GPIOA3	GPIOA4	复用为 SD 接口引脚

1	GPIOA21	GPIOA20	GPIOA19	复用为 SD 接口引脚
2	GPIOB5	GPIOB4	GPIOB3	复用为 SD 接口引脚
3	GPIOB20	GPIOB21	GPIOB22	复用为 SD 接口引脚
0xff	还原 SD 引脚为通用 GPIO 引脚			

描述:

该函数用于将 GPIO 配置成 SD 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~3，来选择哪一组 GPIO 为 SD 接口引脚；往 ModelSel 中写入 0xff 来还原 SD 引脚为通用 GPIO 引脚。

返回值:

无

1.3.10 GpioLcdIoConfig

原型:

void GpioLcdIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 LCD 接口复用选择,其中参数为 0 时为复用为 LCD 接口,参数为 0xff 时还原为通用 GPIO 引脚。

描述:

该函数用于将 GPIO 配置成 LCD 接口引脚或者还原成通用 GPIO 引脚。

返回值:

无

1.3.11 GpioPwcIoConfig

原型:

void GpioPwcIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 Pwc 引脚的复用关系。其中参数 0~1 为复用为 Pwc 引脚，0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.13 GPIO 与 Pwc 引脚复用关系所示。

表格 1.13 GPIO 与 Pwc 引脚复用关系

ModeSel	Pwc	功能

0	GPIOA10	复用为 Pwc 引脚
1	GPIOB7	复用为 Pwc 引脚
0xff	还原 Pwc 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 Pwc 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~1，来选择哪一个 GPIO 配置为 Pwc 接口引脚；通往 ModelSel 中写入 0xff 来还原 Pwc 引脚为通用 GPIO 引脚

返回值:

无

1.3.12 GpioIrIoConfig

原型:

void GpioIrIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 Ir 引脚的复用关系。其中参数 0~2 为复用为 Ir 引脚，0xff 为还原复用引脚为通用 GPIO 引脚。对应关系如表格 1.13 GPIO 与 Pwc 引脚复用关系所示。

表格 1.14 GPIO 与 Ir 引脚复用关系

ModeSel	**Pwc**	**功能**
0	GPIOA10	复用为 Ir 引脚
1	GPIOB7	复用为 Ir 引脚
2	GPIOC2	复用为 Ir 引脚
0xff	还原 Ir 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 Ir 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 Ir 接口引脚；通往 ModelSel 中写入 0xff 来还原 Ir 引脚为通用 GPIO 引脚

返回值:

无

1.3.13 GpioFshcIoConfig

原型:

void GpioFshcIoConfig(uint8_t ModeSel)

参数:

ModeSel:GPIO 与 Fshc 复用关系选择。其中参数值 0~1 为复用为 Fshc 引脚，0xff 为还原复用的 GPIO 为通用 GPIO，对应关系如表格 1.15 GPIO 与 Fshc 复用关系表所示。

表格 1.15 GPIO 与 Fshc 复用关系表

ModeSel	Fsh_hold	Fsh_sck	Fsh_si	Fsh_wp	Fsh_so	Fsh_cs	功能
0	GPIOA13	GPIOA14	GPIOA15	GPIOA16	GPIOA17	GPIOA18	4-bits mode Fshc
1	---	GPIOA14	GPIOA15	---	GPIOA17	GPIOA18	1-bits mode Fshc
0xff							

描述:

该函数用于将 GPIO 配置成 Fshc 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~1，来选择哪一组 Fshc 的模式时 1-bits 还是 4-bits；往 ModelSel 中写入 0xff 来还原 Fshc 引脚为通用 GPIO 引脚。

返回值:

无

1.3.14 GpioSdramIoConfig

原型:

void GpioSdramIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 Sdram 接口复用选择，其中参数为 0 时为复用为 Sdram 接口，参数为 0xff 时还原为通用 GPIO 引脚。

描述:

该函数用于将 GPIO 配置成 Sdram 接口引脚或者还原成通用 GPIO 引脚。

返回值:

无

1.3.15 GpioClk32kIoConfig

原型:

void GpioClk32kIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 32KHZ CLK 输出引脚复用选择，其中参数 0~2 为复用 32KHZ CLK 输出引脚，0xff 为还原通用 GPIO 引脚。

表格 1.16 GPIO 与 32KHZ CLK 输出引脚复用关系

ModeSel	Clk32k	功能
0	GPIOA11	复用为 32k clk 输出引脚
1	GPIOB30	复用为 32k clk 输出引脚
2	GPIOC0	复用为 32k clk 输出引脚
0xff	还原 clk32 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 32KHZ CLK 输出接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 32KHZ CLK 输出接口引脚；通往 ModelSel 中写入 0xff 来还原 32KHZ CLK 输出引脚为通用 GPIO 引脚

返回值:

无

1.3.16 GpioClk12m6mIoConfig

原型:

void GpioClk12m6mIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 12M/16M CLK 输出引脚复用选择，其中参数 0~2 为复用 12M/16M CLK 输出引脚，0xff 为还原通用 GPIO 引脚。具体对应关系如表格 1.17 GPIO 与 12M/16M CLK 输出引脚复用关系所示。

表格 1.17 GPIO 与 12M/16M CLK 输出引脚复用关系

ModeSel	Clk12m16m	功能
0	GPIOA12	复用为 12M/16M CLK 输出引脚

1	GPIOB31	复用为 12M/16M CLK 输出引脚
2	GPIOC1	复用为 12M/16M CLK 输出引脚
0xff	还原 Clk12m16m 引脚为通用 GPIO 引脚	

描述:

该函数用于将 GPIO 配置成 12M/16M CLK 输出接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一个 GPIO 配置为 12M/16M CLK 输出接口引脚；通往 ModelSel 中写入 0xff 来还原 12M/16M CLK 输出引脚为通用 GPIO 引脚

返回值:

无

1.3.17 GpioPcmSyncIoConfig

原型:

void GpioPcmSyncIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 Pcm 复用关系选择。其中参数值 0~2 为复用为 Pcm 引脚，0xff 为还原复用的 GPIO 为通用 GPIO，如表格 1.18 GPIO 与 PCM 对应的复用关系所示。

表格 1.18 GPIO 与 PCM 对应的复用关系

ModeSel	Pcm_Sync	Pcm_Clk	Pcm_Din	Pcm_Do	功能
0	GPIOB3	GPIOB4	GPIOB6	GPIOB5	复用为 PCM 接口引脚
1	GPIOB24	GPIOB25	GPIOB27	GPIOB26	复用为 PCM 接口引脚
2	GPIOC9	GPIOC10	GPIOC12	GPIOC11	复用为 PCM 接口引脚
0xff	还原 PCM 引脚为通用 GPIO 引脚				

描述:

该函数用于将 GPIO 配置成 PCM 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一组 GPIO 为 PCM 接口引脚；往 ModelSel 中写入 0xff 来还原 PCM 引脚为通用 GPIO 引脚。

返回值:

无

1.3.18 GpioI2sIoConfig

原型:

void GpioI2sIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 I2S 复用关系选择。其中参数值 0~2 为复用为 I2S 引脚，0xff 为还原复用的 GPIO 为通用 GPIO，如表格 1.19 GPIO 与 I2s 对应的复用关系所示。

表格 1.19 GPIO 与 I2s 对应的复用关系

ModeSel	I2s_lrclk	I2s_bclk	I2s_din	I2s_Do	功能
0	GPIOB3	GPIOB4	GPIOB6	GPIOB5	复用为 I2s 接口引脚
1	GPIOB24	GPIOB25	GPIOB27	GPIOB26	复用为 I2s 接口引脚
2	GPIOC9	GPIOC10	GPIOC12	GPIOC11	复用为 I2s 接口引脚
0xff	还原 I2s 引脚为通用 GPIO 引脚				

描述:

该函数用于将 GPIO 配置成 I2s 接口引脚或还原通用 GPIO 引脚。往 ModelSel 参数中写入 0~2，来选择哪一组 GPIO 为 I2s 接口引脚；往 ModelSel 中写入 0xff 来还原 I2s 引脚为通用 GPIO 引脚。

返回值:

无

1.3.19 GpioMclkIoConfig

原型:

void GpioMclkIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 MCLK 输入或输出引脚复用选择，其中参数 0~3 为复用 MCLK 输入或输出引脚，0xff 为还原通用 GPIO 引脚。具体对应关系如表格 1.20 GPIO 与 MCLK 输入/输出引脚复用关系所示。

表格 1.20 GPIO 与 MCLK 输入/输出引脚复用关系

ModeSel	I2s_Mclk(o)	I2s_Mclk(i)	功能
0	GPIOB2	---	复用为 MCLK 输出引脚

1	GPIOC8	---	复用为 MCLK 输出引脚
2	---	GPIOB2	复用为 MCLK 输入引脚
3	---	GPIOC8	复用为 MCLK 输入引脚
0xff	还原 MCLK 引脚为通用 GPIO 引脚		

描述:

该函数用于将 GPIO 配置成 MCLK 输入或者输出，或者还原为通用 GPIO 引脚。当参数为 0~1 时配置为 MCLK 为输出引脚，2~3 时配置为 MCLK 为输入引脚，0XFF 时还原为通用 GPIO 引脚。

返回值:

无

1.3.20 GpioSwIoConfig

原型:

void GpioSwIoConfig(uint8_t ModeSel)

参数:

ModeSel：GPIO 与 Sw 复用关系选择，当参数为非 0xFF 时为复用为 Sw 引脚，当参数为 0xFF 时为还原为通用 GPIO 引脚。

描述:

该函数用于配置 GPIO 为 Sw 功能或者还原为通用 GPIO 引脚功能。

返回值:

无

1.3.21 GpioSpisIoConfig

原型:

void GpioSpisIoConfig(uint8_t ModeSel)

参数:

ModeSel:GPIO 与 Spis 复用关系选择,当参数值为 0 时复用为 Spis 引脚,当参数为 0xFF 时还原为通用 GPIO。

描述:

该函数用于配置 GPIO 为 Spis 引脚或还原为通用 GPIO 引脚。

返回值:

无。

1.3.22 GpioCtsCmpoutIoConfig

原型:

void GpioCtsCmpoutIoConfig(uint8_t ModeSel)

参数:

ModeSel: GPIO 与 Cts 复用关系选择，当参数值为 0 时复用为 Cts 引脚，当参数为 0xFF 时还原为通用 GPIO。

描述:

该函数用于配置 GPIO 为 Cts 引脚或还原为通用 GPIO 引脚。

返回值:

无。

1.3.23 GpioSdSpiSel

原型:

void GpioSdSpiSel(uint8_t Sel)

参数:

ModeSel：配置 VMEM 是被 Sdio 总线还是 Spi 总线获取。当参数为 0 时，Sdio 允许获取 VMEM 数据，当参数为 1 时， Spi 总线允许获取 VMEM 数据。

描述:

该函数用于配置 VMEM 是被 Sdio 总线还是 Spi 总线获取。

返回值:

无

1.3.24 GpioSetReg

原型:

void GpioSetReg(uint8_t RegIndex, uint32_t Val)

参数:

RegIndex：GPIO 对应的寄存器索引，详见宏定义描述。

Val: 寄存器被写入的数值。

描述:

设置 GPIO 对应索引的寄存器。

返回值:

无

1.3.25 GpioGetReg

原型:

uint32_t GpioGetReg(uint8_t RegIndex)

参数:

RegIndex：GPIO 对应的寄存器索引，详见宏定义描述。

描述:

读取 GPIO 对应寄存器索引的数值。

返回值:

GPIO 对应寄存器索引的数值。

1.3.26 GpioSetRegOneBit

原型:

void GpioSetRegOneBit(uint8_t RegIndex, uint32_t GpioIndex)

参数:

RegIndex：GPIO 对应的寄存器索引。

GpinIndex: Bit 位索引，详见宏定义描述。

描述:

设置 GPIO 寄存器的某一位为 1。

返回值:

无。

1.3.27 GpioClrRegOneBit

原型:

void GpioClrRegOneBit(uint8_t RegIndex, uint32_t GpioIndex)

参数:

RegIndex：GPIO 对应的寄存器索引。

GpinIndex: Bit 位索引，详见宏定义描述。

描述:

清除 GPIO 寄存器的某一位为 0。

返回值:

无。

1.3.28 GpioSetRegBits

原型:

void GpioSetRegBits(uint8_t RegIndex, uint32_t mask)

参数:

RegIndex: GPIO 对应的寄存器索引。

Mask：需要对寄存器操作的 Bit 位，1 为有效，可以有多个 Bit 位为 1.

描述:

设置 GPIO 寄存器的某些 bit 位为 1，并且可以一次设置多个 bit 位。该函数内部会关闭中断后再操作，操作完毕之后再次开启中断。中断关闭期间只执行一句代码，解决代码不可重入问题。

返回值:

无。

1.3.29 GpioClrRegBits

原型:

void GpioClrRegBits(uint8_t RegIndex, uint32_t mask)

参数:

RegIndex: GPIO 对应的寄存器索引。

Mask：需要对寄存器操作的 Bit 位，1 为有效，可以有多个 Bit 位为 1.

描述:

设置 GPIO 寄存器的某些 bit 位为 0，并且可以一次设置多个 bit 位。该函数内部会关闭中断后再操作，操作完毕之后再次开启中断。中断关闭期间只执行一句代码，解决代码不可重入问题。

返回值:

无。

1.3.30 GpioIntEn

原型:

void GpioIntEn(uint8_t RegIndex, uint32_t GpioIndex, uint8_t EdgeSel)

参数:

RegIndex: GPIO 端口寄存器索引, 有效值 GPIO_A_INT,GPIO_B_INT,GPIO_C_INT 分别为 GPIOA 中断、GPIOB 中断、GPIOC 中断。

GpioIndex:GPIO 引脚索引，有效值 GPIOA0~GPIOA31， GPIOB0~GPIOB31，GPIOC0~GPIOC31。

EdgeSel：中断触发方式。

描述:

开启 GPIO 某一引脚中断，同时设置该引脚的中断触发方式。

返回值:

无。

1.3.31 GpioIntDis

原型:

void GpioIntDis(uint8_t RegIndex, uint32_t GpioIndex)

参数:

RegIndex: GPIO 端口寄存器索引, 有效值 GPIO_A_INT,GPIO_B_INT,GPIO_C_INT 分别对应 GPIOA 中断、GPIOB 中断、GPIOC 中断。

GpioIndex: GPIO 引脚索引, 有效值 GPIOA0~GPIOA31， GPIOB0~GPIOB31，GPIOC0~GPIOC31。

描述:

关闭 GPIO 某一引脚中断触发。

返回值:

无。

1.3.32 GpioIntClr

原型:

void GpioIntClr(uint8_t RegIndex, uint32_t GpioIndex)

参数:

RegIndex: GPIO 端口寄存器索引, 有效值 GPIO_A_INT,GPIO_B_INT,GPIO_C_INT 分别对应 GPIOA 中断、GPIOB 中断、GPIOC 中断。

GpioIndex: GPIO 引脚索引, 有效值 GPIOA0~GPIOA31, GPIOB0~GPIOB31, GPIOC0~GPIOC31。

描述:

清除 GPIO 某一引脚中断标志位。

返回值:

无。

1.3.33 GpioIntFlagGet

原型:

uint32_t GpioIntFlagGet(uint8_t RegIndex)

参数:

RegIndex: GPIO 端口寄存器索引, 有效值 GPIO_A_INT,GPIO_B_INT,GPIO_C_INT 分别对应 GPIOA 中断、GPIOB 中断、GPIOC 中断。

描述:

获取 GPIO 某一端口中断标志寄存器。

返回值:

GPIO 某一端口中断标志寄存器内容。

1.3.34 GpioPorSysReset

原型:

void GpioPorSysReset(bool RestSrc)

参数:

RestSrc:复位源的选择,有效值为 GPIO_RSTSRC_SYSREST,GPIO_RSTSRC_PORREST,默认为 GPIO_RSTSRC_SYSREST。GPIO_RSTSRC_SYSREST 为 GPIO 引脚通过系统复位或者上电 POR 复位，GPIO_RSTSRC_PORREST 为 GPIO 引脚只有 POR 上电复位。

描述:

设置 GPIO 引脚复位源。

返回值:

无。

1.4 使用流程举例

1、设置 GPIOA6 输出高电平，强上拉，输出驱动能力为 24mA。

```
//第一步：设置 GPIOA6 输出使能
GpioClrRegOneBit(GPIO_A_IE, GPIOA6);
GpioSetRegOneBit(GPIO_A_OE, GPIOA6);
//第二步：设置为强拉状态
GpioSetRegBits(GPIO_A_DS, GPIOA6);
//第三步：设置为上拉状态
GpioClrRegOneBit(GPIO_A_PU,GPIOA6);
GpioClrRegOneBit(GPIO_A_PD,GPIOA6);
//第四步：设置为强驱动输出状态
GpioSetRegBits(GPIO_A_OUTDS, GPIOA6);
```

2、设置 GPIOB6 为上升沿触发中断

```
//第一步：设置 GPIOB6 为输入状态
GpioClrRegOneBit(GPIO_B_OE, GPIOB6);
GpioSetRegOneBit(GPIO_B_IE, GPIOB6);
//第二步：设置为无上下拉状态。
GpioSetRegOneBit(GPIO_B_PU, GPIOB6);
GpioClrRegOneBit(GPIO_B_PD, GPIOB6);
//第三步：使能 GPIOB6 中断
GpioIntEn(GPIO_B_INT, GPIOB6, GPIO_NEG_EDGE_TRIGGER);
//第四步：使能 GPIO IRQC 中断
NVIC_EnableIRQ(GPIO_IRQn)
```

3、设置 GPIO 复用为 Spi Master，其中 GPIOA2 复用为 Spi Master 的 MOSI 引脚，GPIOA3 复用为 Spi Master 的 CLK 引脚，GPIOA4 复用为 Spi Master 的 MISO 引脚。

GpioSpimIoConfig(0);

2 特殊 GPIO 功能

2.1 防倒灌功能

芯片提供了三组 GPIO，分别为 GPIOA、GPIOB、GPIOC，其中 GPIOA0，GPIOA1，GPIOA11，GPIOA12 四个 GPIO 引脚除了作为普通的 GPIO 外，还具有防倒灌的功能。

防倒灌的好处在于不仅可以在芯片处于 PowerDown 之后能够有效防止电流倒灌，还可以在芯片处于 PowerOn 时，如果外部电压过大，也可以有效防止电流倒灌。在使用防倒灌功能时，需要开启 Charge Pump 模块，在山景提供的 boot 代码中已经默认开启该功能

2.2 充电指示功能

GPIOA0 除了作为普通的 GPIO，具有防倒灌功能之外，还具有充电指示作用。充电指示功能是用防倒灌 IO 实现的，目的是芯片在 PowerDown 之后，还能够启动充电指示功能。

充电指示功能可以实现：

- 通过内 LED 电路周期性（频率可调）地控制下拉电流源的开关，从而实现充电闪烁功能。闪烁频率范围：1HZ~16HZ.
- 当充电完成时，即 VIN 高于 4.15V 时，充电结束，软件可以配置极性来使下拉电流源常开或常关，从而表现出充电结束后指示灯常亮或常灭。
- GPIOA0 可以配置使能硬件输入恒流源为：1.7mA, 2.4mA, 4.1mA.

2.3 接口函数描述

2.3.1 GpioA0SetIcs

原型：

void GpioA0SetIcs(ICS_VAL IcsVal)

参数：

IcsVal: GPIOA0 做下拉电流源时设置下拉电流源输出电流的大小。有效值 ICS_IMA7,ICS_2MA4,ICS_4MA1 分别对应输出的电流值为 1.7mA，2.4mA，4.1mA.

描述：

当 GPIOA0 做下拉电流源使用时，设置下拉电流源输出电流大小。

返回值:

无。

2.3.2 GpioA0SetMode

原型:

void GpioA0SetMode(A0_MODE_AF A0ModeAf)

参数:

A0ModeAf：GPIOA0 被选择的功能模式，取值有 A0_FOR_GPIO,A0_FOR_ICS,A0_FOR_CHARGER,A0_FOR_PWM,分别设置的模式为普通 IO 模式，下拉电流源模式，充电指示灯模式以及 PWM 输出模式。

描述:

设置 GPIOA0 的工作类型。

返回值:

无。

2.3.3 ChargerSetMode

原型:

void ChargerSetMode(uint8_t LedWidth, uint8_t LedPolarity)

参数:

LedWidth：Led out 高低脉冲的宽度，范围：1~16hz。

LedPolarity：充电完成后，Led 输出极性，0：常亮，1：常灭

描述:

设置 GPIOA0 作为 LED 充电指示功能，该函数在 ledcharger.h 中。

返回值:

无。

2.4 使用流程举例

设置 GPIOA0 作为 LED 充电指示功能，GPIOA0 内接 1.7mA 恒流源，充电过程中 0.5s 闪烁一次，充电完成时常亮。

```
//Set 1.7mA to GPIOA0
GpioA0SetIcs(ICS_1MA7);
```

```
//Pulse LedWidth:(7+1)*62.5ms = 500ms
ChargerSetMode(8, 0);
```