



A76xx Series Open SDK

_GPIO开发_应用指导

LTE 模组

芯讯通无线科技(上海)有限公司
上海市长宁区临虹路289号3号楼芯讯通总部大楼
电话: 86-21-31575100
技术支持邮箱: support@simcom.com
官网: www.simcom.com

名称:	A76xx Series Open SDK_GPIO开发_应用指导
版本:	V1.00
类别:	应用文档
状态:	已发布

版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2023，保留一切权利。

Version History

Version	Date	Owner	What is new
V1.00	2022-10-26		第一版

SIMCom
Confidential

About this Document

本文档适用于 A1803S open 系列、A1603 open 系列、A1606 open 系列。

SIMCom
Confidential

目录

版权声明..... 2

Version History 3

About this Document 4

目录..... 5

1GPIO 应用开发..... 6

 1.1 GPIO 介绍 6

 1.2 基于 GPIO 的开发..... 6

 1.2.1 常用 API..... 6

 1.2.2 相关头文件 8

 1.3 例子 8

 1.4 备注 8

1GPIO 应用开发

1.1 GPIO 介绍

GPIO（General Purpose Input/Output Port），即通用输入输出端口，就是芯片的一些引脚。作为输入端口时，可以通过它们读入引脚的状态--高电平或低电平，作为输出端口时，我们可以通过它们输出高电平或低电平来控制连接的外围设备。

1.2 基于 GPIO 的开发

1.2.1 常用 API

1. Gpio 属性设置

接口:	int sAPI_GpioConfig(unsigned int gpio, SC_GPIOConfiguration gpioConfig);
输入:	Gpio 引脚号, GpioConfig
输出:	无
返回值:	GPIORC_OK
NOTE:	Gpio 属性需结合结构体 SC_GPIOConfiguration 使用

2. Gpio 绑定中断设置

接口:	int sAPI_GpioConfigInterrupt(unsigned int gpio_num, SC_GPIOTransitionType type, GPIOCallback handler);
输入:	Gpio 引脚号, 中断触发方式, 中断绑定函数
输出:	无
返回值:	GPIORC_OK
NOTE:	

3. Gpio 引脚 MFPR 设置

接口:	int sAPI_setPinFunction(unsigned int gpio_num, unsigned int pin_func);
输入:	Gpio 引脚号, Pin 功能选项 (0-7)

输出:	无
返回值:	GPIORC_OK
NOTE:	使用此函数，需结合相应的 Pinmux 表

4. Gpio 输入输出方向设置

接口:	int sAPI_setGpioDirection(unsigned int gpio_num, unsigned int pin_dir);
输入:	Gpio 引脚号，方向（0：输入；1：输出）
输出:	无
返回值:	GPIORC_OK
NOTE:	

5. 获得某个 Gpio 的输入输出方向

接口:	int sAPI_getGpioDirection(unsigned int gpio_num);
输入:	Gpio 引脚号
输出:	无
返回值:	0 或 1
NOTE:	返回值 0 表示输入，1 表示输出

6. Gpio 输出电平设置

接口:	int sAPI_setGpioValue(unsigned int gpio_num, unsigned int pin_level);
输入:	Gpio 引脚号，电平值(0:低电平；1：高电平)
输出:	无
返回值:	GPIORC_OK
NOTE:	该函数只在引脚设置为输出方向时生效

7. 获得某个 Gpio 输出电平

接口:	int sAPI_getGpioValue(unsigned int gpio_num);
输入:	Gpio 引脚号
输出:	无
返回值:	返回值 0 表示低电平 1 表示高电平
NOTE:	0

8. Gpio 输出电平设置保存，重启后依旧

接口:	int sAPI_setGpioPinConfig(SC_GPIONumAndConfiguration gpiocfg);
输入:	SC_GPIONumAndConfiguration
输出:	无

返回值:

无

NOTE:

该函数实现对 Gpio 输出电平值的保存，以及在重启后依旧保持之前的状态

1.2.2 相关头文件

GPIO 头文件

```
#include <sc_gpio.h>
```

1.3 例子

参考 demo_gpio.c(UI_demo)

1.4 备注

1.变量

gpio、gpio_num 为 gpio 引脚号,使用类型为 SC_MODULE_GPIO_X(x:1 2 3 ……)

SC_GPIOTransitionType 的使用类型为:

SC_GPIO_NO_EDGE ,//无边沿检测

SC_GPIO_RISE_EDGE, //上升沿检测

SC_GPIO_FALL_EDGE, //下降沿检测

SC_GPIO_TWO_EDGE, //双边沿检测

pin_dir(输入输出方向)的使用类型为:

SC_GPIO_IN_PIN, //输入

SC_GPIO_OUT_PIN, //输出

2.结构体

typedef enum

{

SC_GPIO_IN_PIN = 0,

SC_GPIO_OUT_PIN = 1

}SC_GPIOPinDirection;

typedef enum

{

SC_GPIO_PULL_DISABLE = 0,

SC_GPIO_PULLUP_ENABLE,

SC_GPIO_PULLEDN_ENABLE

}SC_GPIOPullUpDown;

typedef enum


```
{  
    SC_GPIO_NO_EDGE = 0,  
    SC_GPIO_RISE_EDGE,  
    SC_GPIO_FALL_EDGE,  
    SC_GPIO_TWO_EDGE,  
}SC_GPIOTransitionType;
```

typedef struct

```
{  
    SC_GPIOPinDirection    pinDir;  
    UINT32                  initLv;  
    SC_GPIOPullUpDown       pinPull;  
    SC_GPIOTransitionType   pinEd;  
    GPIOCallback            isr;  
    GPIOCallback            wu;  
} SC_GPIOConfiguration;
```

typedef struct

```
{  
    UINT32 gpioNum;  
    SC_GPIOConfiguration gpioAttribute;  
}SC_GPIONumAndConfiguration;
```

结构体的使用参照 API