

FANCONN™ 开源 NBIOT 开发平台	文档编号	产品版本	密级
	V1.0	V1.0	公开
	文档名称：LiteOne GPIO 配置		共 8 页

# **FANCONN 开源 NBIOT 开发平台**

## **LiteOne GPIO 配置**

广州准捷电子科技有限公司

版本：V1.0

## 版权声明

本手册版权属广州准捷电子科技有限公司（以下简称“准捷电子”）所有，并保留一切权利。非准捷电子同意（书面形式），任何单位及个人不得擅自摘录本手册部分或全部，违者我们将追究其法律责任。

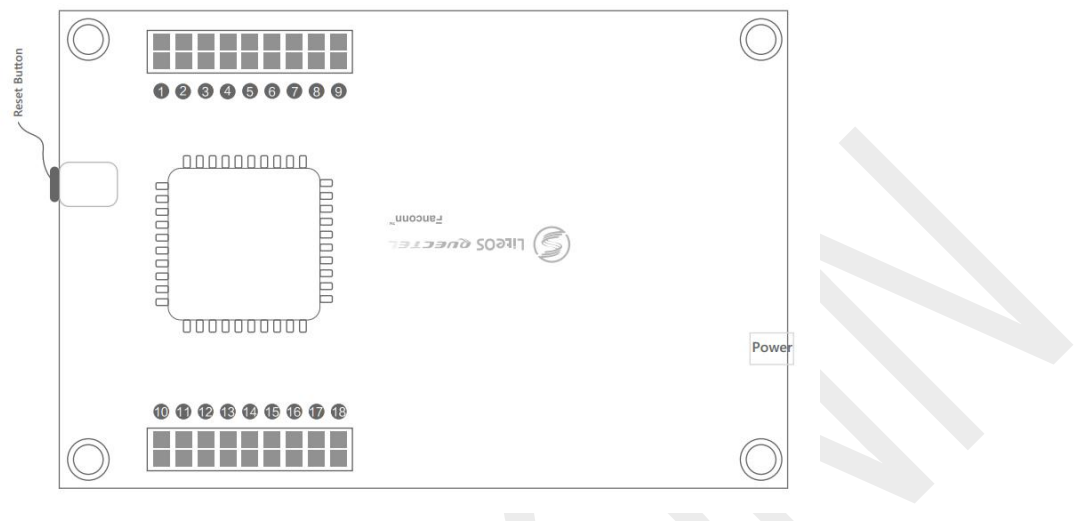
## 版本记录

版本号	作者	版本说明	日期
V1.0	九月阁	初始版本	2018-10-31

## 目录

1. GPIO 引脚对应图.....	4
2. 初始化函数.....	6
3. 设置 GPIO 电平.....	7
4. 获取 GPIO 电平.....	7
5. GPIO 特殊配置.....	8

1. GPIO 引脚对应图

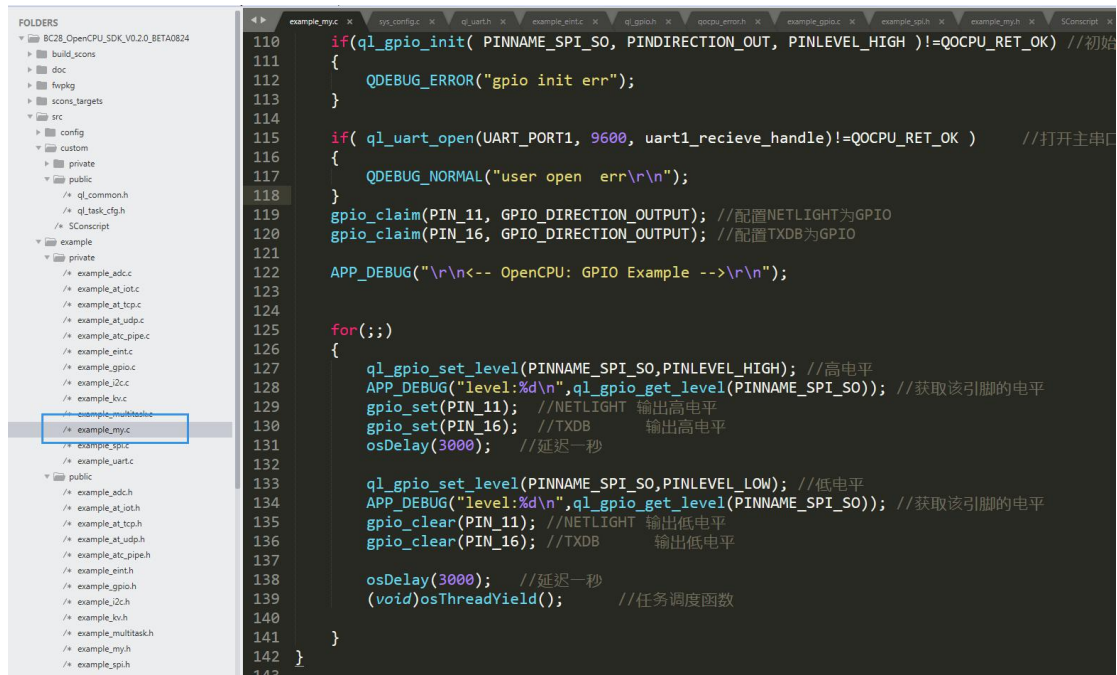


丝印的对应引脚名：		软件 SDK 对应引脚名：	
①	NC NC		
②	SWD-C SWD-D	NULL	NULL
③	RTS VDDEXT	PINNAME_RST	NULL
④	GPIO9 CTS	PINNAME_USIM_DETECT	PINNAME_CTS
⑤	RI DTR	NULL	PINNAME_DTR
⑥	RX1 TX1	NULL	NULL
⑦	REST NETLIGHT	NULL	NULL (*特殊配置)
⑧	GND GND		
⑨	5V 5V		
⑩	SPI2-SI SPI2-CKL	PINNAME_SPI_SI	PINNAME_SPI_CLK
⑪	SPI2-SO SPI2-CS	PINNAME_SPI_SO	PINNAME_SPI_CS
⑫	TX29 RX28	PINNAME_UART_TX3	PINNAME_UART_RX3
⑬	RXDB TXDB	PINNAME_DBG_RXD	NULL(*特殊配置)
⑭	ADC NC	NULL	NULL
⑮	SPI1-CS SPI1-CL	PINNAME_SPI_CS2	PINNAME_SPI_CLK2
⑯	SPI1-SI SPI1-SO	PINNAME_SPI_SI2	PINNAME_SPI_SO2
⑰	GND GND		
⑱	3.3 3.3		

可以参考开箱说明书。  
根据移远提供的资料不是很完善导致引脚丝印一些地方出现小瑕疵，具体正确引脚图请参考上面的开箱说明书为准

程序配置参考对应例程:

程序配置三个 GPIO 进行高低电平切换, 间隔时间 3 秒, 并且通过串口打印 PINNAME\_SPI\_SO 的电平。



```
110 if(ql_gpio_init( PINNAME_SPI_SO, PINDIRECTION_OUT, PINLEVEL_HIGH )!=QOCPU_RET_OK) //初始
111 {
112     QDEBUG_ERROR("gpio init err");
113 }
114
115 if( ql_uart_open(UART_PORT1, 9600, uart1_recieve_handle)!=QOCPU_RET_OK ) //打开主串口
116 {
117     QDEBUG_NORMAL("user open err\r\n");
118 }
119 gpio_claim(PIN_11, GPIO_DIRECTION_OUTPUT); //配置NETLIGHT为GPIO
120 gpio_claim(PIN_16, GPIO_DIRECTION_OUTPUT); //配置TXDB为GPIO
121
122 APP_DEBUG("\r\n<-- OpenCPU: GPIO Example -->\r\n");
123
124 for(;;)
125 {
126     ql_gpio_set_level(PINNAME_SPI_SO,PINLEVEL_HIGH); //高电平
127     APP_DEBUG("level:%d\n",ql_gpio_get_level(PINNAME_SPI_SO)); //获取该引脚的电平
128     gpio_set(PIN_11); //NETLIGHT 输出高电平
129     gpio_set(PIN_16); //TXDB 输出高电平
130     osDelay(3000); //延迟一秒
131
132     ql_gpio_set_level(PINNAME_SPI_SO,PINLEVEL_LOW); //低电平
133     APP_DEBUG("level:%d\n",ql_gpio_get_level(PINNAME_SPI_SO)); //获取该引脚的电平
134     gpio_clear(PIN_11); //NETLIGHT 输出低电平
135     gpio_clear(PIN_16); //TXDB 输出低电平
136
137     osDelay(3000); //延迟一秒
138     (void)osThreadYield(); //任务调度函数
139 }
140
141 }
142
143 }
```

## 2. 初始化函数

GPIO 初始化函数如下:

```
QOCPU_RET ql_gpio_init( Enum_PinName      pinName,  
                        Enum_PinDirection  dir,  
                        Enum_PinLevel      level  
                        );
```

pinName: GPIO 引脚

dir: 配置方向

PINDIRECTION\_IN: 输入

PINDIRECTION\_OUT: 输出

Level: 初始化 GPIO 引脚电平

PINLEVEL\_LOW: 低电平

PINLEVEL\_HIGH: 高电平

PINLEVEL\_NONE: 不配置

在配置 GPIO 输入模式的时候第三个参数选择 PINLEVEL\_NONE。

返回: 判断是否是 QOCPU\_RET\_OK, 否则初始化出错

参考配置:

```
if(ql_gpio_init(PINNAME_SPI_SO,PINDIRECTION_OUT,PINLEVEL_HIGH)!=QOCPU_RET_OK)  
{  
    QDEBUG_ERROR("gpio init err");  
}
```

其中配置 PINNAME\_SPI\_SO 引脚作为输出, 初始化电平为高电平, 判断初始化是否成功。

```
if(ql_gpio_init(PINNAME_UART_TX3,PINDIRECTION_IN, PINLEVEL_NONE)!=QOCPU_RET_OK)  
{  
    QDEBUG_ERROR("gpio init err");  
}
```

其中配置 PINNAME\_UART\_TX3 引脚作为输入, 不配置电平, 判断初始化是否成功。

### 3. 设置 GPIO 电平

GPIO 设置引脚函数如下:

```
QOCPU_RET ql_gpio_set_level(Enum_PinName pinName, Enum_PinLevel level);
```

pinName: GPIO 引脚

level: 设置电平

PINLEVEL\_LOW: 低电平

PINLEVEL\_HIGH: 高电平

参考配置:

```
ql_gpio_set_level(PINNAME_SPI_S0, PINLEVEL_HIGH); //高电平
```

```
ql_gpio_set_level(PINNAME_SPI_S0, PINLEVEL_LOW); //低电平
```

### 4. 获取 GPIO 电平

GPIO 设置引脚函数如下:

```
bool ql_gpio_get_level(Enum_PinName pinName);
```

pinName: GPIO 引脚

返回: GPIO 的引脚电平

参考配置

```
APP_DEBUG("level:%d", ql_gpio_get_level(PINNAME_SPI_S0)); //获取该引脚的电平
```

关于 APP\_DEBUG 函数需要自己定义, 可以参考 GPIO 例程, 直接输出电平的状态

## 5. GPIO 特殊配置

该配置只针对引脚对应图的那两个特殊标注的两个引脚。这两个引脚只能作为 GPIO 输出使用，不能作为输入。

丝印的对应引脚名:		软件 SDK 对应引脚名:	
①	NC NC		
②	SWD-C SWD-D	NULL	NULL
③	RTS VDDEXT	PINNAME_RST	NULL
④	GPIO9 CTS	PINNAME_USIM_DETECT	PINNAME_CTS
⑤	RI DTR	NULL	PINNAME_DTR
⑥	RX1 TX1	NULL	NULL
⑦	REST NETLIGHT	NULL	NULL (*特殊配置)
⑧	GND GND		
⑨	5V 5V		
⑩	SPI2-SI SPI2-CKL	PINNAME_SPI_SI	PINNAME_SPI_CLK
⑪	SPI2-SO SPI2-CS	PINNAME_SPI_SO	PINNAME_SPI_CS
⑫	TX29 RX28	PINNAME_UART_TX3	PINNAME_UART_RX3
⑬	RXDB TXDB	PINNAME_DBG_RXD	NULL (*特殊配置)
⑭	ADC NC	NULL	NULL
⑮	SPI1-CS SPI1-CL	PINNAME_SPI_CS2	PINNAME_SPI_CLK2
⑯	SPI1-SI SPI1-SO	PINNAME_SPI_SI2	PINNAME_SPI_SO2
⑰	GND GND		
⑱	3.3 3.3		

配置函数如下

```
gpio_claim(PIN_11, GPIO_DIRECTION_OUTPUT); //配置 NETLIGHT 为 GPIO
gpio_claim(PIN_16, GPIO_DIRECTION_OUTPUT); //配置 TXDB 为 GPIO
```

在使用下面函数之前必须确保该引脚已经配置为 GPIO

应用函数如下

```
gpio_set(PIN_11); //NETLIGHT 输出高电平
gpio_set(PIN_16); //TXDB 输出高电平
```

```
gpio_clear(PIN_11); //NETLIGHT 输出低电平
gpio_clear(PIN_16); //TXDB 输出低电平
```