

■ **ST17H66T**

GPIO Application Note

Version 1.1

Author: lenze

Security: Public

Date: 2022.10

LENZE

Copyright © 2020 Shenzhen Lenze technology Co.,LTD All rights reserved.
Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.

Revision History

Revision	Author	Date	Description
V1.1		2022.10	This document is used for ST17H66T

目录

1 简介	1
1.1 特殊 IO	2
1.1.1 P16、P18	2
1.2 GPIO 模式	2
1.2.1 GPIO 输出	2
1.2.2 GPIO 输入	2
1.2.3 GPIO retention	2
1.2.4 GPIO 上下拉电阻	2
1.2.5 中断和唤醒	3
1.3 ANALOG 模式	3
2 GPIO 典型应用	3
2.1 GPIO 输出	3
2.2 GPIO 输入	3
2.3 GPIO retention	3
2.4 GPIO 上下拉电阻	4
2.5 中断和唤醒	4

图表目录

表 1: GPIO 上电默认属性配置	1
---------------------------	----------

1.1 特殊 IO

1.1.1 P16、P18

P16、P18 默认做模拟口，接晶振、电容组成振荡电路，其中 P16 为 XTALI，P18 为 XTALO。

1.2 GPIO 模式

GPIO 模式是最常用的模式，可配置为输出并输出高低电平，可以配置为输入读取外部的高低电平。

当配置为输入时，支持中断和唤醒。

1.2.1 GPIO 输出

配置相应 GPIO 方向寄存器为输出，向输出寄存器写 1 或 0，即可输出高或低电平。

1.2.2 GPIO 输入

配置相应 GPIO 方向寄存器为输入，读取输入寄存器的值，即可获取当前 GPIO 的电平状态。

如使用中断，需要打开 GPIO 中断使能功能，并配置中断产生条件。

如使用唤醒，需要打开 GPIO 唤醒使能功能，并配置唤醒产生条件。

1.2.3 GPIO retention

当 GPIO 做输出时，可配置 retention 功能。retention 默认是关闭的。

retention 打开时，系统休眠时，GPIO 的输出特性和输出值保持不表。retention 关闭时，系统休眠时，GPIO 会恢复默认输入态。

比如，P00 运行时配置为 GPIO 输出且输出 1，如系统进入休眠时候，该 GPIO 会变为输入态，此时不会输出 1。如果想让该 GPIO 在休眠时也保持输出 1 这种状态，那么需要在休眠前配置该 GPIO 的 retention 功能。

1.2.4 GPIO 上下拉电阻

每个 GPIO 支持四种上下拉电阻配置：

- 浮空:高阻态。
- 强上拉:上拉到 VDD3，高电平，驱动电流大。上拉电阻 10kΩ 欧姆。
- 弱上拉:上拉到 VDD3，高电平，驱动电流小。上拉电阻 1MΩ 欧姆。
- 下拉:下拉到地，低电平，下拉电阻 150kΩ 欧姆。
- GPIO 默认：浮空。

1.2.5 中断和唤醒

所有 GPIO 支持中断和唤醒。

中断支持电平触发和边沿触发，唤醒支持边沿触发。

注意事项：GPIO 做唤醒源使用时，必须配置该引脚的内部上拉电阻或下拉电阻，不能是高阻态。

1.3 ANALOG 模式

只有 P11、P15、P16、P18、P20、支持模拟功能。

- 32.768K 晶振振荡电路：P16、P18 接电容、32.768K 晶振组成振荡电路。
- ADC:采集引脚上的电压，单端支持的引脚有 P11、P15、P20，差分支持的引脚有 P20P15。

2 GPIO 典型应用

2.1 GPIO 输出

配置对应 GPIO 方向寄存器(swporta_ddr)为输出，设置输出寄存器(swporta_dr)为 0 或 1。

驱动有对应的 API，直接调用即可。

将 P2 设置为输出，并输出 1。

```
gpio_dir(P2, OEN);  
gpio_write(p2, 1);
```

2.2 GPIO 输入

配置相应 GPIO 方向寄存器(swporta_ddr)为输入，读取输入寄存器(swporta_ddr)，即可获取当前 GPIO 的电平状态。

将 P2 设置为输入，并读取 GPIO 电平。

```
gpio_dir(P2, IE);  
ret = gpio_read(P2);
```

2.3 GPIO retention

当 GPIO 配置 GPIO 输出时，当系统休眠后，GPIO 输出信息将会丢失。如果想在系统休眠后，仍保持 GPIO 输出状态并保持输出的高低电平，需要使用 GPIO retention 功能。

将 P2 设置为输出，设置系统休眠时仍然保持输出高电平。

```
gpio_retention(p2);  
gpio_dir(P2, OEN);  
gpio_write(p2, 1);
```

2.4 GPIO 上下拉电阻

每个 GPIO 支持四种上下拉配置：悬空、强上拉、上拉、下拉。

将 P2 设置为输入，设置为强上拉，读取当前 GPIO 状态。

```
gpio_dir(P2, IE);  
gpio_pull_set(P2, STRONG_PULL_UP);  
ret = gpio_read(p2);
```

2.5 中断和唤醒

使用 GPIO 中断时，需要配置 GPIO 中断产生条件。中断产生后，GPIO 驱动会响应中断并调用用户配置的回调函数。

使用 GPIO 唤醒时，当系统进入休眠前，根据当前 GPIO 的电平状态设置唤醒系统的条件，当该条件产生时，系统唤醒并会调用用户配置的回调函数。

配置 GPIO 中断前需要初始化 GPIO 中断，pin_test 为全局变量。

```
ret = gpioin_init(pin_test, sizeof(pin_test)/sizeof(pin_test[0]));  
if(ret != PPlus_SUCCESS)  
{  
    LOG("gpio init error:%d\n", ret);  
}  
  
gpio_dir(GPIO_KEY_NO1, IE); //set gpio input  
gpio_pull_set(GPIO_KEY_NO1, STRONG_PULL_UP); //pull up 10k  
ret = gpioin_register(GPIO_KEY_NO1, LC_Key_Pin_IntHandler_Pos,  
LC_Key_Pin_IntHandler_Neg);
```

```
const uint8_t pin_map[GPIO_NUM] =  
{  
    0, //p0  
    1, //p1  
    2, //p2  
    3, //p3  
    7, //p7  
    9, //p9  
    10, //p10  
    11, //p11  
    14, //p14  
    15, //p15  
    16, //p16  
    17, //p17  
    18, //p18  
    20, //p20  
    23, //p23  
    24, //p24
```

```
25,    //p25
26,    //p26
27,    //p27
31,    //p31
32,    //p32
33,    //p33
34,    //p34
};

void    LC_Key_Pin_IntHandler_Pos(GPIO_Pin_e pin, IO_Wakeup_Pol_e type)
{
    uint8_t pin_idx = pin_map[pin];
    uint32_t triggered = s_gpio_wakeup_src[pin_idx/32] & BIT(pin_idx%32);

    if(triggered)
    {
        LOG("wakeup0\n");
    }
    else
    {
        LOG("Int0\n");
    }
}
void    LC_Key_Pin_IntHandler_Neg(GPIO_Pin_e pin, IO_Wakeup_Pol_e type)
{
    uint8_t pin_idx = pin_map[pin];
    uint32_t triggered = s_gpio_wakeup_src[pin_idx/32] & BIT(pin_idx%32);
    if(triggered)
    {
        LOG("wakeup1\n");
    }
    else
    {
        LOG("Int1\n");
    }
}
}
```

将 GPIO 设置为输入，配置强上拉，注册了上升沿和下降沿回调。

系统为非休眠状态时，下降沿中断的回调为“Int1”，而在系统休眠时，下降沿唤醒的回调为“wakeup0”。

系统为非休眠状态时，上升沿中断的回调为“Int0”，而在系统休眠时，上升沿唤醒的回调为“wakeup1”。