YC31xx LPM 应用说明

V1.0

Yichip Microelectronics

Revision History

Version	Date	Author	Description
V1.0	2020-2-20	Fanglingxue	Initial version

Confidentiality Level:

confidential

目录

1 文档说明

1.1 编写目的

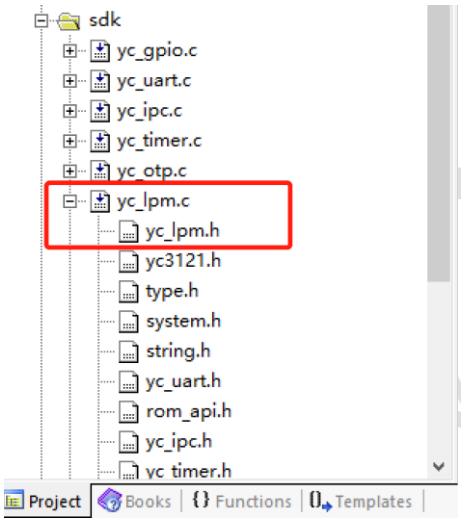
为 demo 中 LPM 模式相关示例代码及 API 提供指南

1.2 适用范围

3121 系列芯片

1.3 文件说明

本说明基于以下文件:
LPM Demo 路径为 ModuleDemo\LPM
LPM 库文件 yc_lpm.c yc_lpm.h 文件路径为 Librarier\sdk
Lpm 配置同时会设置 Trng、IPC、 SYSCLK 以及 GPIO 引脚，这些模块在对应文档中会进行详细介绍。



2 函数说明

2.1 lpm_read

函数原型: uint32_t lpm_read(volatile int *addr)
函数说明: 读对应寄存器地址数据

参数	方向	说明
addr	IN	需要读取数据的地址

返回值	说明
uint32_t num	对应 addr 地址中数据

2.2 lpm_write

函数原型：void lpm_write(volatile int *addr,uint32_t value)
函数说明：写对应寄存器地址数据

参数	方向	说明
addr	IN	需要写数据的地址
value	IN	需要写入地址中的数据

返回值	说明
None	None

2.3 lpm_bt_write

函数原型：void lpm_bt_write(uint8_t type,uint32_t val)
函数说明：写蓝牙寄存器值

参数	方向	说明
type	IN	需要写的蓝牙寄存器号
val	IN	需要写入蓝牙寄存器的数据

返回值	说明
None	None

2.4 lpm_bt_read

函数原型：uint32_t lpm_bt_read(uint8_t type)
函数说明：读蓝牙寄存器值

参数	方向	说明
type	IN	需要读得蓝牙寄存器号

返回值	说明
uint32_t	读得得蓝牙寄存器数据

2.5 lpm_sleep

函数原型：void lpm_sleep(void)
函数说明：进入 LPM 模式

参数	方向	说明
None	None	None

返回值	说明
None	None

2.6 setlpmval

函数原型：void setlpmval(volatile int * addr,uint8_t startbit,uint8_t bitwidth,uint32_t val)
函数说明：写寄存器指定 bit 位数据

参数	方向	说明
addr	IN	写入寄存器地址
startbit	IN	开始 bit 位
bitwidth	IN	bit 宽度
val	IN	写入寄存器的值

返回值	说明
None	None

2.7 readlpmval

函数原型：uint32_t readlpmval(volatile int * addr,uint8_t startbit,uint8_t bitwidth)
函数说明：读寄存器指定 bit 位数据

参数	方向	说明
addr	IN	写入寄存器地址
startbit	IN	开始 bit 位
bitwidth	IN	bit 宽度

返回值	说明
uint32_t num	读得得数据值

2.8 GPIO_Unused_Pd

函数原型：void GPIO_Unused_Pd(void)
函数说明：设置 GPIO 引脚为下拉状态

参数	方向	说明
None	None	None

返回值	说明
None	None

2.9 get_otp

函数原型：static uint32_t get_otp(void)
函数说明：获得芯片 otp

参数	方向	说明
None	None	None

返回值	说明
uint32_t num	芯片 otp

2.10 BT_Hibernate

函数原型：void BT_Hibernate(void)

函数说明：蓝牙休眠

参数	方向	说明
None	None	None

返回值	说明
None	None

2.11 Chip_Speedstep

函数原型：void Chip_Speedstep(void)
函数说明：时钟降频

参数	方向	说明
None	None	None

返回值	说明
None	None

2.12 CM0_Sleep

函数原型：void CM0_Sleep(uint32_t time, GPIO_TypeDef GPIOx, uint16_t GPIO_Pin, uint8_t islow_wakeup, uint8_t is_powerdownbt)
函数说明：M0 核睡眠

参数	方向	说明
time	IN	等待唤醒时间单位：秒
GPIOx	IN	GPIO 引脚端口
GPIO_Pin	IN	GPIO_Pin 引脚
islow_wakeup	IN	唤醒方式：0：高电平唤醒 1：低电平唤醒
is_powerdownbt	IN	0：失能蓝牙睡眠 1：使能蓝牙睡眠

返回值	说明
None	None

3 示例代码

第一步：关闭 trng
第二步：使能蓝牙
第三步：关闭没有使用的时钟 clk
第四步：下拉没有使用的 GPIO 引脚

根据配置的 GPIO 对应引脚上拉，选择(1)：将蓝牙进入睡眠模式
(2)：对芯片时钟降频
(3)：进入 LPM 模式

```
int main(void)
{
    UART_Configuration();

    MyPrintf("Yichip Yc3121 LPM test Demo V1.0.\r\n");

    //step1:close trng
    Disable_Trng();

    //step2:enable BT
    IpcInit();
    NVIC_EnableIRQ(BT_IRQn);

    //step3:close unused clk
    SYSCTRL_AHBPeriphClockCmd(SYSCTRL_AHBPeriph_INTR|SYSCTRL_AHBPeriph_SHA|\
                              SYSCTRL_AHBPeriph_CRC|SYSCTRL_AHBPeriph_PWM|\
                              SYSCTRL_AHBPeriph_WDT|SYSCTRL_AHBPeriph_USB|\
                              SYSCTRL_AHBPeriph_SPI|SYSCTRL_AHBPeriph_DES|\
                              SYSCTRL_AHBPeriph_RSA|SYSCTRL_AHBPeriph_ASE|\
                              SYSCTRL_AHBPeriph_7816|SYSCTRL_AHBPeriph_SM4|\
                              SYSCTRL_AHBPeriph_7811|SYSCTRL_AHBPeriph_ADC7811|\
                              SYSCTRL_AHBPeriph_CP, DISABLE);

    //step4:close unused gpio
    GPIO_Unused_Pd();
```

```
    GPIO_Config(GPIOC, GPIO_Pin_7, PULL_DOWN);
    GPIO_Config(GPIOC, GPIO_Pin_8, PULL_DOWN);
    GPIO_Config(GPIOC, GPIO_Pin_9, PULL_DOWN);

    static uint8_t first = 0;

    while (1)
    {
        if(GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_7) == 1 && first == 0)
        {
            MyPrintf("BT start sleep.....\n\n");

            BT_Hibernate();
            NVIC_DisableIRQ(BT_IRQn);
//          IpcInit();  通过 Ipcinit 唤醒 BT
//          NVIC_EnableIRQ(BT_IRQn);
            first++;
        }
        if(GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_8) == 1)
        {
            MyPrintf("M0 start sleep0.....\n\n");
            Chip_Speedstep();
//      SYSCTRL_HCLKConfig(SYSCTRL_HCLK_Div2);  恢复时钟即可恢复正常
        }
        if(GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_9) == 1)
        {
            MyPrintf("M0 start sleep1.....\n\n");
            CM0_Sleep(0, GPIOA, GPIO_Pin_11, 0, 1);
        }
    }
}
```