



# 低功耗测试说明

# 低功耗测试说明

测试项	功耗	休眠状态	唤醒源
BT深睡+M0休眠	2.55ua	BT不广播无法维持连接，唤醒芯片复位	RTC定时唤醒，IO唤醒
BT深睡,M0降频到12M	2.807ma	BT不广播无法维持连接，唤醒芯片不复位	M0降频，程序仍在运行，需要软件逻辑上操作进行唤醒
BT浅睡，M0休眠 (蓝牙未连接进入休眠)	90ua	BT广播以一秒间隔维持，M0休眠，唤醒复位不需要重新初始化BT	RTC定时唤醒，IO唤醒，BT连接唤醒
BT浅睡，M0休眠 (蓝牙连接进入休眠)	370ua (蓝牙连接后连接间隔由app端决定，不同app连接后进入休眠功耗会有所不同)	BT保持连接,M0休眠。唤醒复位不需要重新初始化BT	RTC定时唤醒，IO唤醒，BT发送数据唤醒和断开连接唤醒
POWER OFF	0.33ua	整个芯片下电	插入USB或者POWERKEY开机

# 低功耗测试DEMO说明

标注2为读取标志位  
判断蓝牙唤醒后是否  
需要进行初始化蓝牙  
操作，测试项case3  
中会调用  
**SSC\_LPMKeyWrite ()**  
写入标志位，唤醒复  
位后读取该标志位信  
息。

```
int main(void)
{
    UART_Configuration();
    MyPrintf("YC3121 Sleep Mode Demo V1.0!\n");
    SSC_LPMKeyRead(bpk_buf, FLAG_LEN, 0);
    SysTick_Config(CPU_MHZ/100);
    BT_ConfigInit(bpk_buf[0]);

    if(bpk_buf[0] == TRUE) /*为TRUE 蓝牙不初始化唤醒，唤醒后清0*/
    {
        bpk_buf[0] = FALSE;
        SSC_LPMKeyWrite(bpk_buf, FLAG_LEN, 0);
    }
}
```

# 低功耗测试DEMO说明

```
MyPrintf("=====\n");  
MyPrintf(" Please input the following numbers to configure the sleep mode !\n");  
MyPrintf(" 1: M0 Sleep and BT Deep Sleep\n");  
MyPrintf(" 2: M0 Div Frequency and BT Deep sleep!\n");  
MyPrintf(" 3: M0 Sleep and BT Light Sleep!\n");  
MyPrintf("=====\n");  
uint8_t uartretval;
```

# 低功耗测试说明

---

休眠分为3个测试case:

1. BT深睡+M0休眠
2. BT深睡, M0降频到12M
3. BT浅睡, M0休眠。进入休眠状态分为蓝牙连接后进入休眠和蓝牙不连接进入休眠, 两者功耗不同

# 低功耗测试说明

**CASE1休眠配置:**  
直接配置IO脚 PD,  
调用CM0\_Sleep,唤醒  
条件配置为定时10S  
唤醒和GPIO33高电平  
唤醒

```
case '1':  
    MyPrintf(" M0 Sleep and BT Deep Sleep....\n");  
    Disable_Trng();  
    GPIO_Unused_Pd();  
    GPIO_Config(GPIOA, GPIO_Pin_1, GPCFG_PD);  
    GPIO_Config(GPIOA, GPIO_Pin_0, GPCFG_PD);  
    GPIO_Config(GPIOA, GPIO_Pin_14, GPCFG_PD);  
    GPIO_Config(GPIOA, GPIO_Pin_15, GPCFG_PD);  
    CM0_Sleep(10,0,1<<1,0,1);  
    break;  
case '2':
```



# 低功耗测试说明

**CASE2睡眠配置:**  
直接配置IO脚 PD,  
M0时钟降频, 关闭  
相应时钟, 关闭蓝牙。  
唤醒条件软件配置为  
IO33高唤醒

```
case '2':  
    MyPrintf(" M0 Div Frequency and BT Deep sleep...\n");  
    GPIO_Config(GPIOC,GPIO_Pin_1,GPCFG_PD);  
    SYSCTRL_HCLKConfig(SYSCTRL_HCLK_Div8);  
    Disable_Trng();  
    GPIO_Unused_Pd();  
    GPIO_Config(GPIOA, GPIO_Pin_1, GPCFG_PD);  
    GPIO_Config(GPIOA, GPIO_Pin_0, GPCFG_PD);  
    GPIO_Config(GPIOA, GPIO_Pin_14, GPCFG_PD);  
    GPIO_Config(GPIOA, GPIO_Pin_15, GPCFG_PD);  
    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_UART | \  
        SYSCTRL_AHBPeriph_CP, DISABLE);  
    BT_Hibernate();  
    while(1)  
    {  
        if(GPIO_ReadInputDataBit(GPIOC,GPIO_Pin_1) == 1)  
        {  
            IpcInit();  
            SYSCTRL_AHBPeriphClockCmd(SYSCTRL_AHBPeriph_INTR | \  
                SYSCTRL_AHBPeriph_UART | SYSCTRL_AHBPeriph_CP, ENABLE);  
            GPIO_Config(UART0_TX_PORT, UART0_TX_PIN, UART0_TXD);  
            GPIO_Config(UART0_RX_PORT, UART0_RX_PIN, UART0_RXD);  
            SYSCTRL_HCLKConfig(SYSCTRL_HCLK_Div2);  
            MyPrintf("Wake up !\n");  
            break;  
        }  
    }
```

# 低功耗测试说明

## CASE3休眠配置:

直接配置IO脚 PD,配置蓝牙进入低功耗模式,广播间隔设为1s,调用CM0\_Sleep,唤醒条件配置为蓝牙唤醒。  
(蓝牙连接, 断开, 发送数据)

```
break;
case '3':
    MyPrintf(" M0 Sleep and BT Light Sleep....\n");
    GPIO_Unused_Pd();
    GPIO_Config(GPIOA, GPIO_Pin_1, GPCFG_PD);
    GPIO_Config(GPIOA, GPIO_Pin_0, GPCFG_PD);
    GPIO_Config(GPIOA, GPIO_Pin_14, GPCFG_PD);
    GPIO_Config(GPIOA, GPIO_Pin_15, GPCFG_PD);
    BT_SetLpmMode(0x01);
    BT_SetLEAdvInterval(0x640);
    bpk_buf[0] = TRUE;
    SSC_LPMKeyWrite(bpk_buf, FLAG_LEN, 0);
    CM0_Sleep(0, 0, 0, 0, 0);
    break;
default:
```