电源管理

1. 实验目的

- 掌握 STC15W4K32S4 系列单片机降低功耗的方式，空闲模式和掉电模式的原理。
- 掌握单片机进入掉电模式和退出掉电模式的寄存器配置及程序设计。

2. 实验内容

- 编写程序实现外部中断唤醒掉电模式 MCU 的程序设计。
- 编写程序实现内部掉电唤醒定时器定时唤醒掉电模式 MCU 的程序设计。

3. 硬件设计

3.1. 电源管理概念介绍

单片机系统电源管理实际上就是控制单片机系统的低功耗，或者说省电。

设计低功耗的单片机系统需要从以下几个方面考虑：

1） 在可以满足功能和性能要求下，尽量选择集成功能模块多的单片机，这样可以少搭建外部硬件电路。

2） 单片机的正常工作电流要关注，还要关注单片机省电模式（空闲模式、掉电模式等）时的静态电流。

3） 在满足应用要求的前提下，选“低配型”的 单片机，减少单片机本身的功耗。

4） 单片机工作电压和频率在满足系统性能要求的情况下，尽可能选低的。

5） 单片机工作频率和单片机系统整体功耗之间要寻求一个合适点，因为工作频率越低，意味着需要更长的处理时间。

6） 控制单片机切换到省电模式时，要考虑切换时间和切换电流。

✧ 注：单片机电源管理，主要针对的就是单片机这个芯片的低功耗。

3.2. STC15W4K32S4 系列单片机电源管理介绍

STC15W4K32S4 系列单片机为了降低功耗，可使之运行于 3 种省电模式，即低速模式、空闲模式和掉电模式。如下表所示。

表 1：STC15W4K32S4 系列单片机工作模式

序号	工作模式	典型功耗	控制方式	备注
1	正常模式	2.7mA ~ 7mA	不进入省电模式	
2	低速模式	待定	PCON2 寄存器相应位控制	设定工作频率
3	空闲模式	约 1.8mA	PCON 寄存器相应位控制	IDL 位置 1 进空闲模式
4	掉电模式	<0.1uA	PCON 寄存器相应位控制	PD 位置 1 进掉电模式

■ STC15W4K32S4 系列单片机内部结构图

STC15W4K32S4 系列单片机中包括：中央处理器(CPU)、程序存储器(Flash)、数据存储器(SRAM)、定时器/计数器、掉电唤醒专用定时器、GPIO 口、A/D 转换、比较器、看门狗、UART1、UART2、UART3、UART4、CCP/PWM/PCA、高速同步串行通信端口 SPI、片内高精度 R/C 时钟及高可靠复位等模块。

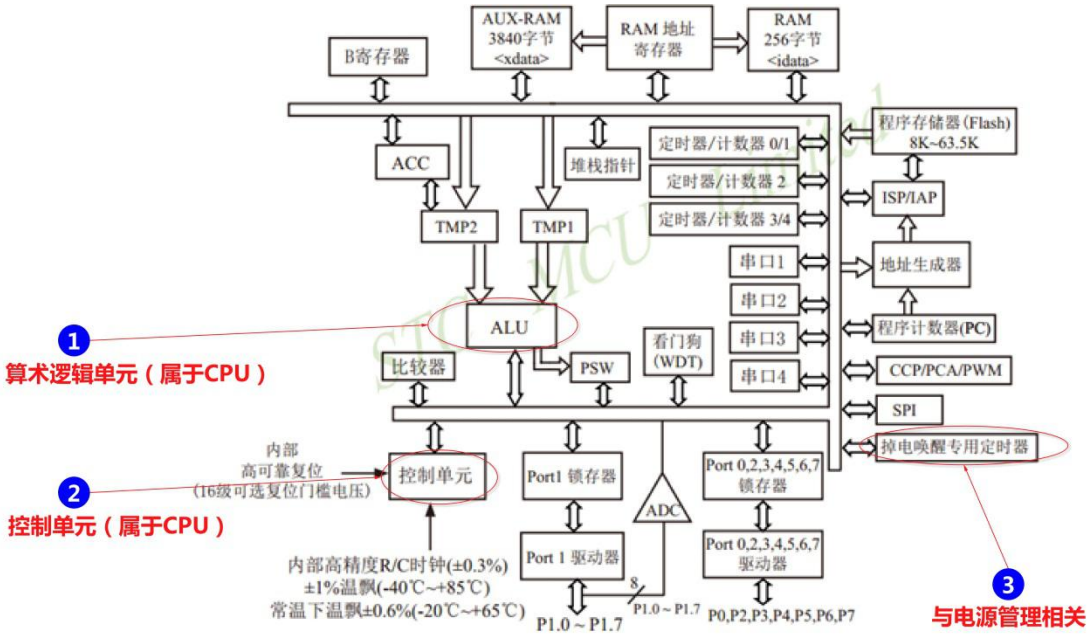

图 1：STC15W4K32S4 系列单片机内部结构图

✧ 注： CPU（Central Processing Unit）是单片机的核心部件，包括运算器和控制器。

■ STC15W4K32S4 系列单片机系统时钟与主时钟

1、 系统时钟是指对主时钟进行分频后给 CPU、定时器、串行口、SPI、CCP/PWM/PCA、A/D 转换等功能模块提供的实际工作时钟。

2、 主时钟可以是内部 R/C 时钟，也可以是外部输入的时钟或外部晶体震荡产生的时钟。

3、 系统时钟和主时钟不是一码事，系统时钟是由主时钟经过分频得出的。如何对主时钟分频，是通过配置 CLK_DIV 时钟分频寄存器（又称 PCON2）的系统时钟选择控制位实现。

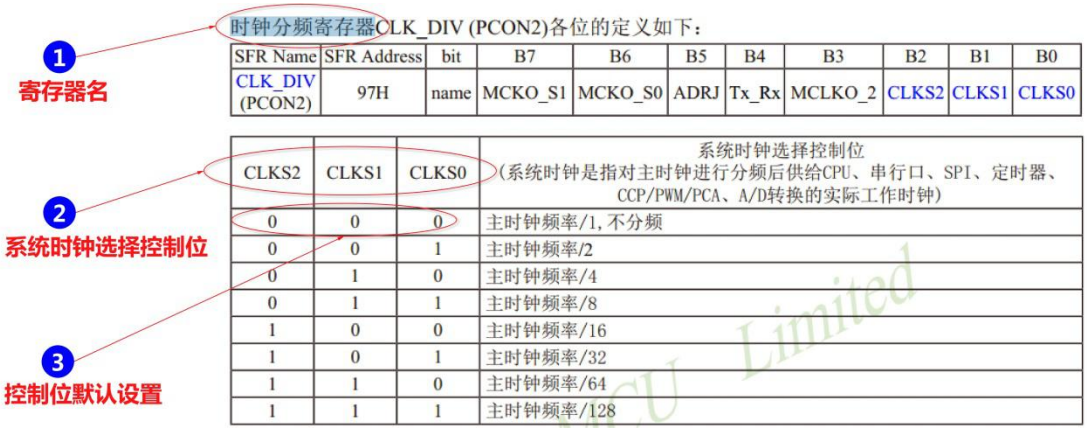

图 2：时钟分频寄存器 CLK_DIV

✧ 注：如果想降低功耗，可适当对主时钟进行分频，实现降低系统时钟频率的目的，但必须以可以实现项目功能为前提。

■ STC15W4K32S4 系列单片机工作在空闲模式

1、 该模式下，除系统不给 CPU 供时钟，CPU 不执行指令外，其余功能部件仍可继续工作。举例：如果定时器已被调用，开启空闲模式，定时器依然在计数。

2、 看门狗在空闲模式下是否工作取决于自身有一个“IDLE”模式位。

3、 空闲模式下，RAM、堆栈指针（PC）、程序状态字（PSW）、累加器（A）等寄存器都保持原有数据。

4、 空闲模式下，GPIO 口保持着空闲模式被激活前那一刻的逻辑状态。

5、 任何一个中断都会将 IDL 位清零，从而退出空闲模式。

6、 单片机从空闲模式下被唤醒，则程序会从上次设置单片机进入空闲模式语句的下一条语句开始往下执行。

图 3：电源控制寄存器 PCON

✧ 注：STC15W4K32S4 系列单片机工作在空闲模式下，单片机的功耗约 1.8mA。

■ STC15W4K32S4 系列单片机工作在掉电模式

1、 该模式下，单片机所使用的时钟（内部系统时钟或外部晶体/时钟）停振，由于无时钟源，CPU、看门狗、定时器、串行口、A/D 转换、SPI 等功能模块停止工作，但外部中断 0~外部中断 4、CCP/PWM/PCA、低压检测电路均可继续工作。

2、 掉电模式下，所有特殊功能寄存器（SFR）维持进入掉电模式前那一刻的状态不变。

3、 掉电模式下，GPIO 口保持着掉电模式被激活前那一刻的逻辑状态。

4、 特定的外部引脚可唤醒掉电模式下的 CPU，这些外部引脚有外部中断引脚、CCP0/CCP1 引脚、串口接收引脚、定时器/计数器外部输入引脚。

5、 除特定的外部引脚可唤醒掉电模式下的 CPU 外，还可通过配置特殊功能寄存器 WKTCH 和 WKTCL 唤醒掉电模式下的 CPU。

6、 单片机从掉电模式下被唤醒，则程序会从上次设置单片机进入掉电模式语句的下一条语句开始往下执行。

7、 特别注意：编写 C 语言程序时，在将进入掉电模式时，一定在设置 MCU 进入掉电模式语句后加 2~4 条空语句（即 NOP 语句）。

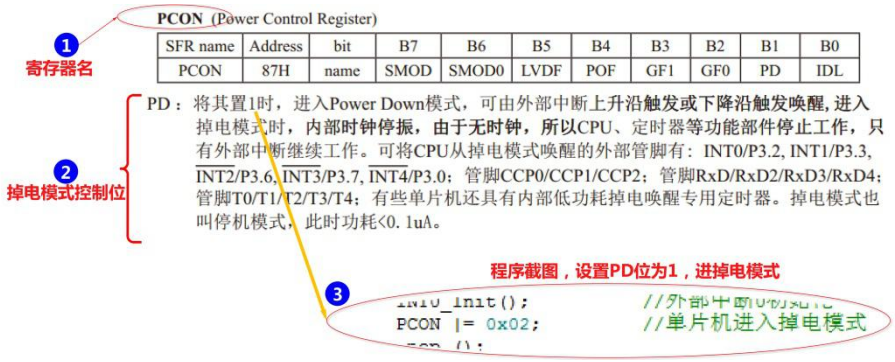

图 4：电源控制寄存器 PCON

✧ 注：STC15W4K32S4 系列单片机工作在掉电模式下，单片机的功耗小于 0.1uA。

3.3. 掉电唤醒定时器

STC15W4K32S4 系列单片机内部掉电唤醒定时器是一个 15 位的计数器（由{WKTCH[6:0]，WKTCL[7:0]}组成 15 位），用于唤醒处于掉电模式的单片机。

如果掉电唤醒定时器 WKTCH 寄存器的 WKTEN 位置 1，则单片机进入掉电模式后，掉电唤醒定时器开始计数，当计数值与用户所设置的值一样时，掉电唤醒定时器将单片机唤醒。

用户所设置的值是通过配置 WKTCL 寄存器的 B0~B7 位和 WKTCH 寄存器的 B0~B6 位实现。具体可参考下图。

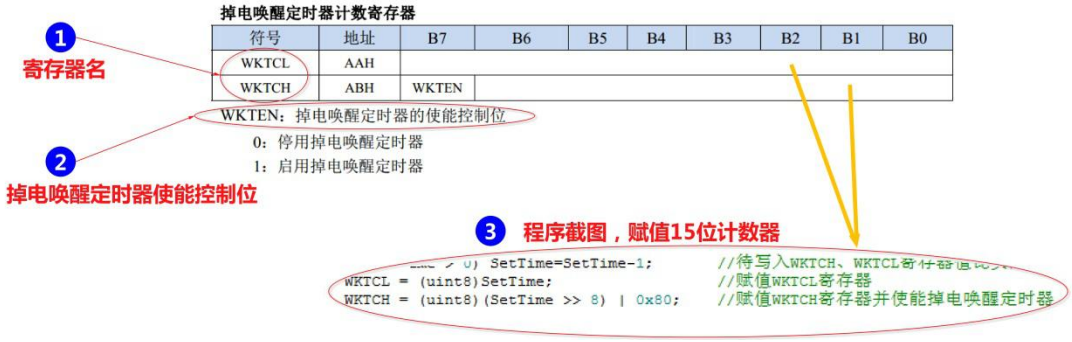

图 5：掉电唤醒定时器计数寄存器 WKTCL 和 WKTCH

✧ 注：15 位计数器的范围是：0~32767，所以写入寄存器{WKTCH[6:0]，WKTCL[7:0]}中的数值也是在 0~32767 范围内。

STC15W4K32S4 系列单片机内部掉电唤醒定时器有自己的内部时钟，其中掉电唤醒定时器计数一次的时间就是由该时钟决定。内部掉电唤醒定时器时钟频率约为 32768HZ，当然误差较大，所以计算出的掉电唤醒定时器定时时间与实际时间会有较大偏差。

下面给出掉电唤醒定时器定时时间计算公式：

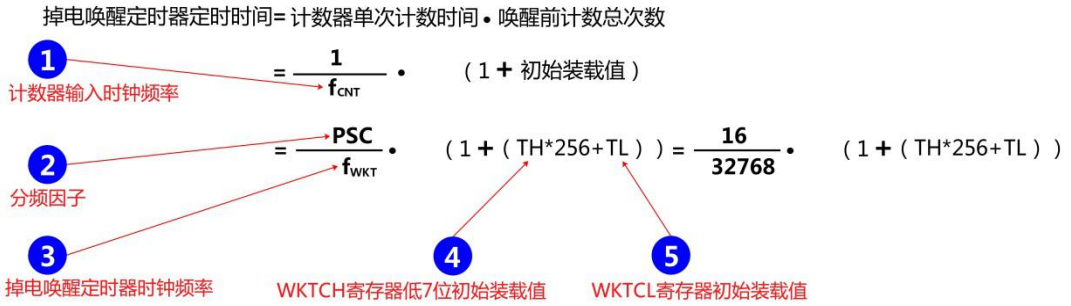

图 6：掉电唤醒定时器定时时间计算公式

✧ 注：用户通过读 RAM 区 F8H 和 F9H 的内容（F8H 存放频率的高字节，F9H 存放频率的低字节）来获取内部掉电唤醒专用定时器出厂时所记录的时钟频率。内部掉电唤醒定时器专用时钟 16 分频后用于掉电唤醒定时器计数。

举例，配置 WKTCH 寄存器低 7 位初始装载值为 0x0F，WKTCL 寄存器初始装载值为 0xFF，计算掉电唤醒定时器定时时间。

1） 十六进制 0x0F 转成十进制是 15，十六进制 0xFF 转成十进制是 255。这样初始装载值为：256*15+255=4095。

2） 掉电唤醒前计数总次数为：4095+1=4096。

3） 掉电唤醒定时器定时时间：(4096*16)/32768=2s。

4） 如果已知掉电唤醒定时器定时时间，计算 WKTCH 寄存器低 7 位初始装载值和 WKTCL 寄存器初始装载值，则是反推过来即可。

4. 软件设计

4.1. 电源管理寄存器汇集

STC15W4K32S4 系列单片机进行电源管理时会用到 4 个寄存器，如下表所示：

表 2：STC15W4K32S4 系列电源管理使用寄存器汇总

序号	寄存器名	读/写	功能描述
1	PCON	读/写	电源控制寄存器。
2	CLK_DIV（PCON2）	读/写	时钟分频寄存器。
3	WKTCH	读/写	掉电唤醒定时器计数寄存器。
4	WKTCL	读/写	

✧ 注：WKTCH 寄存器最高位 WKTEN 是掉电唤醒定时器的使能控制位。

4.2. 单片机掉电唤醒 - 外部中断 0 唤醒实验（下降沿方式）

✧ 注：本节的实验源码是在“实验 2-4-1：外部中断 0（下降沿中断方式）”的基础上修改。本节对应的实验源码是：“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”。

4.2.1. 工程需要用到的 c 文件

本例需要用到的 c 文件如下表所示，工程需要添加下表中的 c 文件。

表 3：实验需要用到的 c 文件

序号	文件名	后缀	功能描述
1	led	.c	包含与用户 led 控制有关的用户自定义函数。
2	exint	.c	外部中断有关的用户自定义函数。
3	delay	.c	包含用户自定义延时函数。

4.2.2. 头文件引用和路径设置

■ 需要引用的头文件

```
1. #include "delay.h"
2. #include "led.h"
3. #include "exint.h"
```

■ 需要包含的头文件路径

本例需要包含的头文件路径如下表：

表 4：头文件包含路径

序号	路径	描述
1	..\ Source	led.h、exint.h 和 delay.h 头文件在该路径，所以要包含。
2	..\User	15W4KxxS4.h 头文件在该路径，所以要包含。

MDK 中点击魔术棒，打开工程配置窗口，按照下图所示添加头文件包含路径。

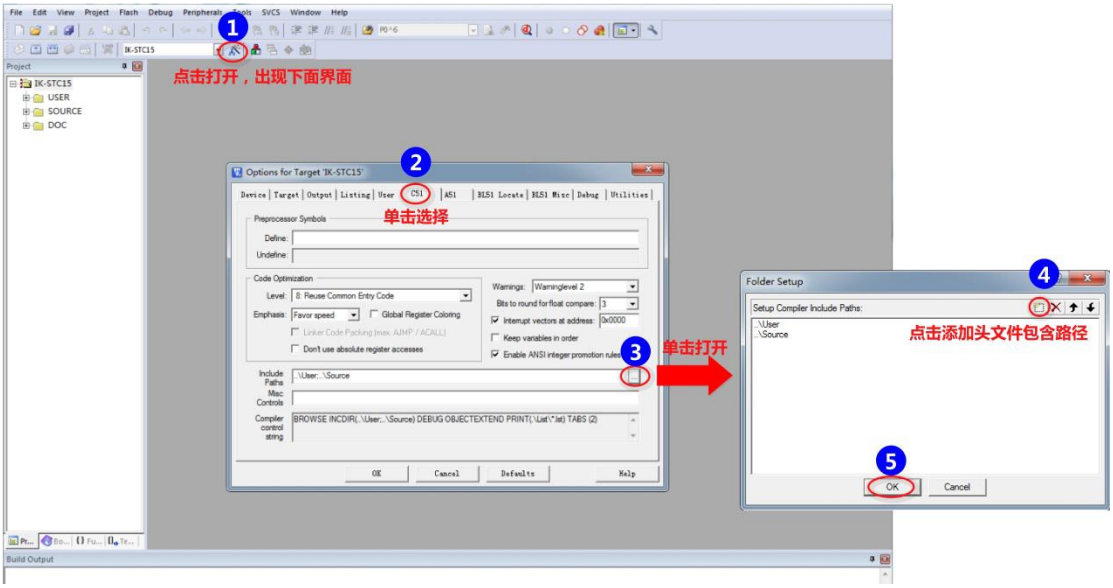

图 7：添加头文件包含路径

4.2.3. 编写代码

首先，在 exint.c 文件中编写外部中断 0 初始化函数 INT0_Init，代码如下。

程序清单：外部中断 0 初始化函数

```
/*************************************************************
功能描述：外部中断 0 初始化
入口参数：无
返回值：无
*************************************************************/
void INT0_Init(void)
{
    IE0  = 0;          //将 INT0 中断请求标志位清"0"
    EX0 = 1;           //使能 INT0 中断允许位
    IT0 = 1;           //选择 INT0 为下降沿触发方式
}
```

然后，编写外部中断服务函数，中断服务函数中无任务，代码如下。

程序清单：中断服务函数

```
/*************************************************************
功能描述：外部中断服务程序
入口参数：无
返回值：无
*************************************************************/
void INT0_Isr (void) interrupt INT0_VECTOR
{
    ;        //无任务
}
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制红灯亮蓝灯灭，约 5s 后控制红灯灭蓝灯亮、对 INT0 进行初始化、配置使单片机进入掉电模式。

代码清单：主函数

```
int main()
{
////////////////////////////////////////////////
//注意: STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
//       高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
//相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
//         P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
////////////////////////////////////////////////
    P0M1 &= 0x3F;   P0M0 &= 0x3F;     //设置 P0.6、P0.7 为准双向口
```

```
10.
11.     SleepDelay = 0;                 //初始清零
12.     EA = 1;                           //允许总中断
13.
14.     while(1)
15.     {
16.         delay_ms(1);                  //延时 1ms
17.         if(++msecond >= 1000)          //约 1 秒到
18.         {
19.             msecond = 0;                  //清 1000ms 计数
20.              led_on(LED_2);               //点亮红色指示灯 DS2
21.             led_off(LED_1);               //熄灭蓝色指示灯 DS1
22.
23.             if(++SleepDelay >= 5)         //约 5 秒到
24.             {
25.                 SleepDelay = 0;          //清零延时用时间变量
26.                 if(TRUE)
27.                 {
28.                     SleepDelay = 0;        //清零延时用时间变量
29.                     led_on(LED_1);           //点亮蓝色指示灯 DS1
30.                     led_off(LED_2);        //熄灭红色指示灯 DS2
31.                     INT0_Init();           //外部中断 0 初始化
32.                     PCON |= 0x02;          //单片机进入掉电模式
33.                     _nop_();
34.                     _nop_();
35.                     _nop_();
36.                 }
37.             }
38.         }
39.     }
40. }
```

4.2.4. 硬件连接

本实验需要使用 P0.6 驱动蓝色指示灯 DS1、P0.7 驱动红色指示灯 DS2，使用 P3.2（外部中断 0 引脚）连接到触摸按键 S4，因此需要按下图所示连接杜邦线和短路帽。

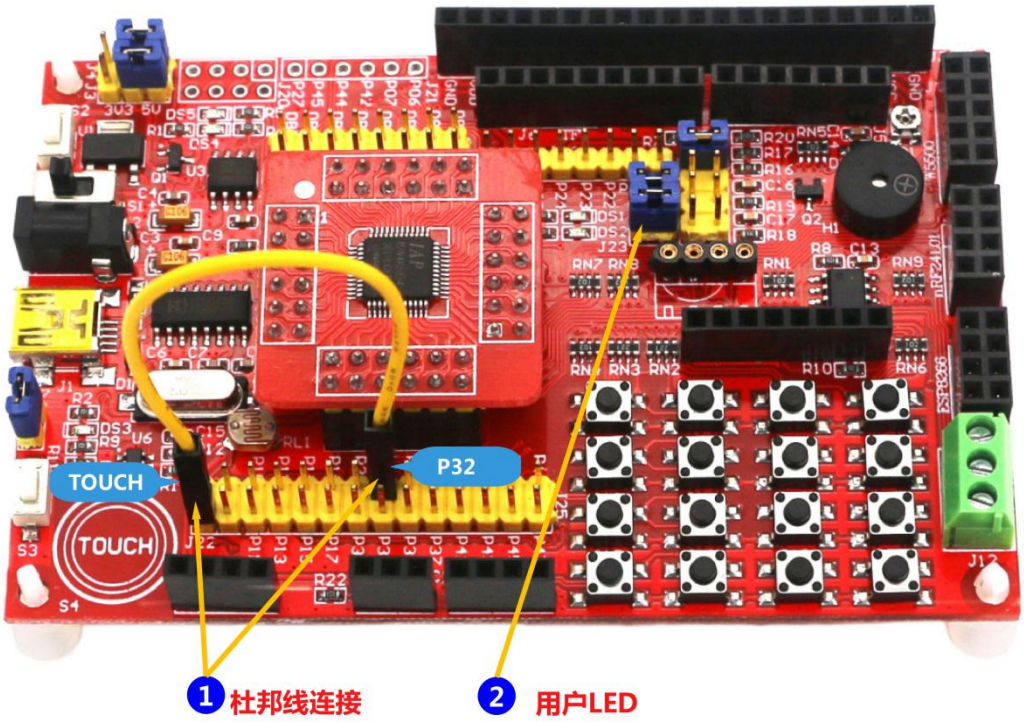

图 8：开发板连接图

4.2.5. 实验步骤

1. 解压“…\第 3 部分：配套例程源码\ 1 - 基础实验程序\”目录下的压缩文件“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\WAKEUP-INT0_FAL\projec”目录下的工程“WAKEUP-INT0_FAL.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WAKEUP-INT0_FAL.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序 。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：

   1）红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机进入掉电模式；

   2）按下触摸按键在 P32 上产生上升沿，单片机依然处于掉电模式，红灯灭蓝灯亮；

   3）触摸按键由按下到松开，在 P32 上产生下降沿，唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机再次进入掉电模式。

4.3.单片机掉电唤醒 - 外部中断 0 唤醒实验（上升沿或下降沿中断方式）

✧ 注：本节的实验源码是在“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”的基础上修改。本节对应的实验源码是：“实验 2-6-2：单片机掉电唤醒 - 外部中断 0（上升沿或下降沿方式）”。

4.3.1. 工程需要用到的 c 文件

本实验需要用到的头文件以及添加头文件包含路径的方法请参考“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”部分。

4.3.2. 编写代码

首先，在 exint.c 文件中编写外部中断 0 初始化函数 INT0_Init，代码如下。

程序清单：外部中断 0 初始化函数

```
1.  /*************************************************************
2.  功能描述：外部中断 0 初始化
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT0_Init(void)
7.  {
8.      IE0  = 0;          //将 INT0 中断请求标志位清"0"
9.      EX0 = 1;           //使能 INT0 中断允许位
10.     IT0 = 0;           //选择 INT0 为上升沿或下降沿触发方式
11. }
```

然后，编写外部中断服务函数，中断服务函数中无任务，代码如下。

程序清单：中断服务函数

```
1.  /*************************************************************
2.  功能描述：外部中断服务程序
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT0_Isr (void) interrupt INT0_VECTOR
7.  {
8.     ;         //无任务
9.  }
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制红灯亮蓝灯灭，约 5s 后控制红灯灭蓝灯亮、对 INT0 进行初始化、配置使单片机进入掉电模式。

代码清单：主函数

```
1.  int main()
```

```
{
//////////////////////////////////////////////
//注意: STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
//       高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
//相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
//         P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
//////////////////////////////////////////////
    P0M1 &= 0x3F;   P0M0 &= 0x3F;     //设置 P0.6、P0.7 为准双向口

    SleepDelay = 0;               //初始清零
    EA = 1;                         //允许总中断

    while(1)
    {
        delay_ms(1);                  //延时 1ms
        if(++msecond >= 1000)          //约 1 秒到
        {
            msecond = 0;                  //清 1000ms 计数
            led_on(LED_2);               //点亮红色指示灯 DS2
            led_off(LED_1);             //熄灭蓝色指示灯 DS1
            if(++SleepDelay >= 5)         //约 5 秒到
            {
                SleepDelay = 0;          //清零延时用时间变量
                if(TRUE)
                {
                    SleepDelay = 0;        //清零延时用时间变量
                    led_on(LED_1);           //点亮蓝色指示灯 DS1
                    led_off(LED_2);        //熄灭红色指示灯 DS2
                    INT0_Init();           //外部中断 0 初始化
                    PCON |= 0x02;          //单片机进入掉电模式
                    _nop_();
                    _nop_();
                    _nop_();
                }
            }
        }
    }
}
```

4.3.3. 硬件连接

本实验需要使用 P0.6 驱动蓝色指示灯 DS1、P0.7 驱动红色指示灯 DS2，使用 P3.2（外部中断 0 引脚）连接到触摸按键 S4，因此需要按下图所示连接杜邦线和短路帽。

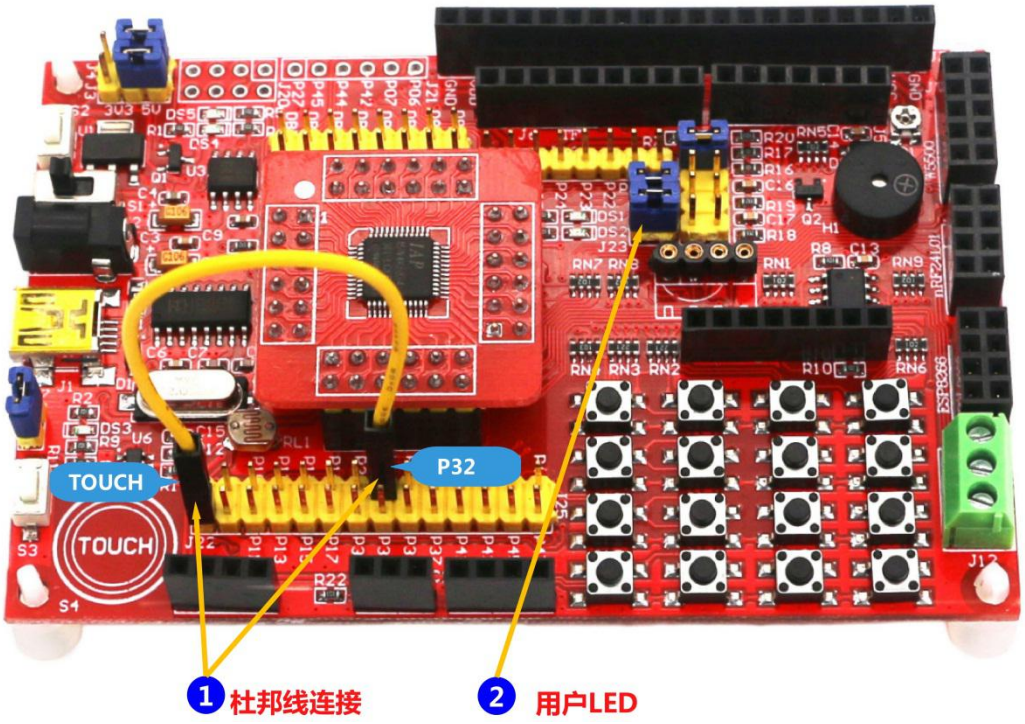

图 9：开发板连接图

4.3.4. 实验步骤

1. 解压“…\第 3 部分：配套例程源码\ 1 - 基础实验程序\”目录下的压缩文件“实验 2-6-2：单片机掉电唤醒 - 外部中断 0（上升沿或下降沿方式）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\WAKEUP-INT0_RISFAL \projec”目录下的工程“WAKEUP-INT0_RISFAL.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WAKEUP-INT0_RISFAL.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序 。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：

   1）红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机进入掉电模式；

   2）按下触摸按键在 P32 上产生上升沿，唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机再次进入掉电模式；

   3）触摸按键由按下到松开，在 P32 上产生下降沿，再次唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机又进入掉电模式。

4.4.单片机掉电唤醒 - 外部中断 1 唤醒实验（下降沿中断方式）

✧ 注：本节的实验源码是在“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”的基础上修改。本节对应的实验源码是：“实验 2-6-3：单片机掉电唤醒 - 外部中断 1（下降沿方式）”。

4.4.1. 工程需要用到的 c 文件

本实验需要用到的头文件以及添加头文件包含路径的方法请参考“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”部分。

4.4.2. 编写代码

首先，在 exint.c 文件中编写外部中断 1 初始化函数 INT1_Init，代码如下。

程序清单：外部中断 1 初始化函数

```
1.  /*************************************************************
2.  功能描述：外部中断 1 初始化
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT1_Init(void)
7.  {
8.      IE1  = 0;          //将 INT1 中断请求标志位清"0"
9.      EX1 = 1;           //使能 INT1 中断允许位
10.     IT1 = 1;           //选择 INT1 为下降沿触发方式
11. }
```

然后，编写外部中断服务函数，中断服务函数中无任务，代码如下。

程序清单：中断服务函数

```
1.  /*************************************************************
2.  功能描述：外部中断服务程序
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT1_Isr (void) interrupt INT1_VECTOR
7.  {
8.      ;         //无任务
9.  }
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制红灯亮蓝灯灭，约 5s 后控制红灯灭蓝灯亮、对 INT1 进行初始化、配置使单片机进入掉电模式。

代码清单：主函数

```
1.  int main()
2.  {
3.  ////////////////////////////////////////////////
4.  //注意: STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
5.  //       高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
6.  //相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
7.  //         P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
8.  ////////////////////////////////////////////////
9.      P0M1 &= 0x3F;   P0M0 &= 0x3F;     //设置 P0.6、P0.7 为准双向口
10.
11.     SleepDelay = 0;               //初始清零
12.     EA = 1;                         //允许总中断
13.
14.     while(1)
15.     {
16.         delay_ms(1);                  //延时 1ms
17.         if(++msecond >= 1000)          //约 1 秒到
18.         {
19.             msecond = 0;                 //清 1000ms 计数
20.              led_on(LED_2);              //点亮红色指示灯 DS2
21.             led_off(LED_1);             //熄灭蓝色指示灯 DS1
22.
23.             if(++SleepDelay >= 5)        //约 5 秒到
24.             {
25.                 SleepDelay = 0;         //清零延时用时间变量
26.                 if(TRUE)
27.                 {
28.                     SleepDelay = 0;       //清零延时用时间变量
29.                     led_on(LED_1);        //点亮蓝色指示灯 DS1
30.                     led_off(LED_2);       //熄灭红色指示灯 DS2
31.                     INT1_Init();          //外部中断 1 初始化
32.                     PCON |= 0x02;         //单片机进入掉电模式
33.                     _nop_();
34.                     _nop_();
35.                     _nop_();
36.                 }
37.             }
38.         }
39.     }
40. }
```

4.4.3. 硬件连接

本实验需要使用 P0.6 驱动蓝色指示灯 DS1、P0.7 驱动红色指示灯 DS2，使用 P3.3（外部中断 1 引脚）连接到触摸按键 S4，因此需要按下图所示连接杜邦线和短路帽。

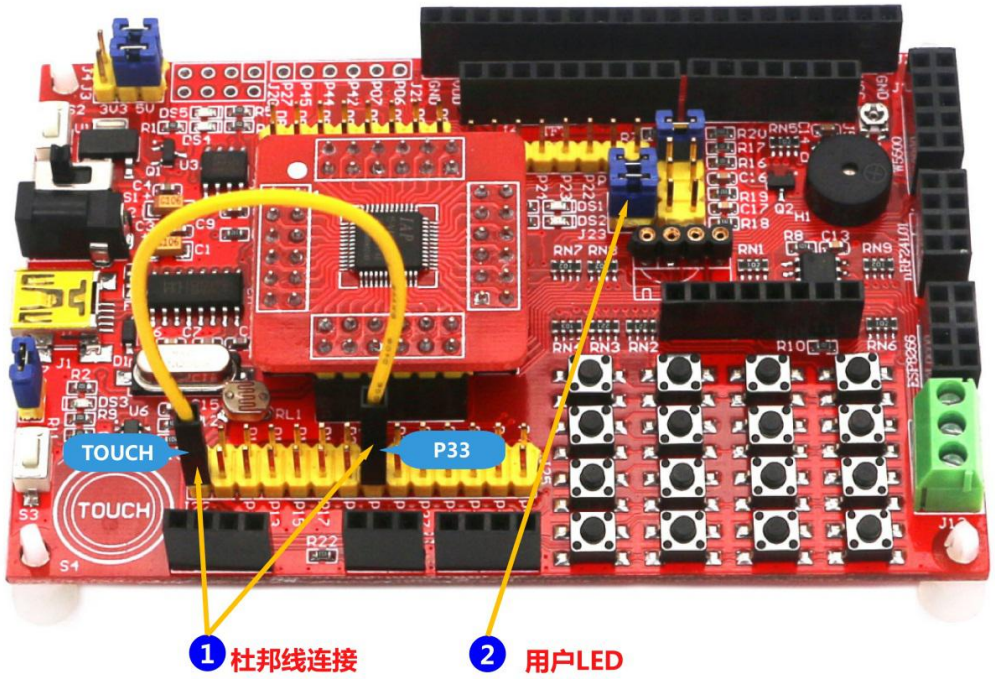

图 10：开发板连接图

4.4.4. 实验步骤

1. 解压“…\第 3 部分：配套例程源码\ 1 - 基础实验程序\”目录下的压缩文件“实验 2-4-3：外部中断 1（下降沿中断方式）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\WAKEUP-INT1_FAL \projec”目录下的工程“WAKEUP-INT1_FAL.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WAKEUP-INT1_FAL.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序 。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：

   1）红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机进入掉电模式；

   2）按下触摸按键在 P33 上产生上升沿，单片机依然处于掉电模式，红灯灭蓝灯亮；

   3）触摸按键由按下到松开，在 P33 上产生下降沿，唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机再次进入掉电模式。

4.5.单片机掉电唤醒 - 外部中断 1 唤醒实验（上升沿或下降沿中断方式）

✧ 注：本节的实验源码是在“实验 2-6-3：单片机掉电唤醒 - 外部中断 1（下降沿方式）”的基础上修改。本节对应的实验源码是：“实验 2-6-4：单片机掉电唤醒 - 外部中断 1（上升沿或下降沿方式）”。

4.5.1. 工程需要用到的 c 文件

本实验需要用到的头文件以及添加头文件包含路径的方法请参考“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”部分。

4.5.2. 编写代码

首先，在 exint.c 文件中编写外部中断 1 初始化函数 INT1_Init，代码如下。

程序清单：外部中断 1 初始化函数

```
1.  /*************************************************************
2.  功能描述：外部中断 1 初始化
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT1_Init(void)
7.  {
8.      IE1  = 0;          //将 INT1 中断请求标志位清"0"
9.      EX1 = 1;           //使能 INT1 中断允许位
10.     IT1 = 0;          //选择 INT1 为上升沿或下降沿触发方式
11. }
```

然后，编写外部中断服务函数，中断服务函数中无任务，代码如下。

程序清单：中断服务函数

```
1.  /*************************************************************
2.  功能描述：外部中断服务程序
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT1_Isr (void) interrupt INT1_VECTOR
7.  {
8.      ;         //无任务
9.  }
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制红灯亮蓝灯灭，约 5s 后控制红灯灭蓝灯亮、对 INT1 进行初始化、配置使单片机进入掉电模式。

代码清单：主函数

```
1.  int main()
```

```
{
////////////////////////////////////////////////
//注意: STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
//       高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
//相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
//         P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
////////////////////////////////////////////////
    P0M1 &= 0x3F;   P0M0 &= 0x3F;     //设置 P0.6、P0.7 为准双向口

    SleepDelay = 0;                  //初始清零
    EA = 1;                          //允许总中断

    while(1)
    {
        delay_ms(1);                   //延时 1ms
        if(++msecond >= 1000)           //约 1 秒到
        {
            msecond = 0;                    //清 1000ms 计数
             led_on(LED_2);                 //点亮红色指示灯 DS2
            led_off(LED_1);                  //熄灭蓝色指示灯 DS1

            if(++SleepDelay >= 5)          //约 5 秒到
            {
                SleepDelay = 0;           //清零延时用时间变量
                if(TRUE)
                {
                    SleepDelay = 0;         //清零延时用时间变量
                    led_on(LED_1);          //点亮蓝色指示灯 DS1
                    led_off(LED_2);         //熄灭红色指示灯 DS2
                    INT0_Init();            //外部中断 0 初始化
                    PCON |= 0x02;           //单片机进入掉电模式
                    _nop_();
                    _nop_();
                    _nop_();
                }
            }
        }
    }
}
```

4.5.3. 硬件连接

本实验需要使用 P0.6 驱动蓝色指示灯 DS1、P0.7 驱动红色指示灯 DS2，使用 P3.3（外

部中断 1 引脚）连接到触摸按键 S4，因此需要按下图所示连接杜邦线和短路帽。

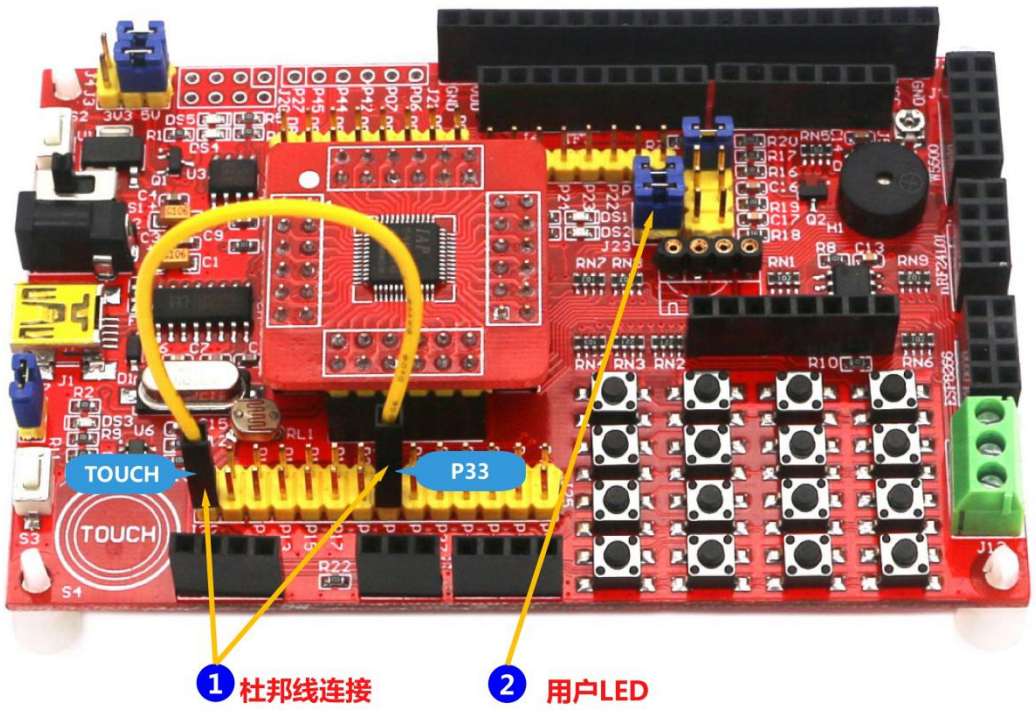

图 11：开发板连接图

4.5.4. 实验步骤

1. 解压“…\第 3 部分：配套例程源码\ 1 - 基础实验程序\”目录下的压缩文件“实验 2-4-4：外部中断 1（上升沿或下降沿中断方式）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\WAKEUP-INT1_RISFAL \projec”目录下的工程“WAKEUP-INT1_RISFAL.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WAKEUP-INT1_RISFAL.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序 。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：

   1）红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机进入掉电模式；

   2）按下触摸按键在 P33 上产生上升沿，唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机再次进入掉电模式；

   3）触摸按键由按下到松开，在 P33 上产生下降沿，再次唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机又进入掉电模式。

4.6.单片机掉电唤醒 - 外部中断 2 唤醒实验（下降沿中断方式）

✧ 注：本节的实验源码是在“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”的基础上修改。本节对应的实验源码是：“实验 2-6-5：单片机掉电唤醒 - 外部中断 2（下降沿方式）”。

4.6.1. 工程需要用到的 c 文件

本实验需要用到的头文件以及添加头文件包含路径的方法请参考“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”部分。

4.6.2. 编写代码

首先，在 exint.c 文件中编写外部中断 2 初始化函数 INT2_Init，代码如下。

程序清单：外部中断 2 初始化函数

```
1.  int main(void)
2.  /*************************************************************
3.  功能描述：外部中断 2 初始化
4.  入口参数：无
5.  返回值：无
6.  *************************************************************/
7.  void INT2_Init(void)
8.  {
9.      AUXR2 |= 0x10;    //使能 INT2 中断允许位
10. }
```

然后，编写外部中断服务函数，中断服务函数中无任务，代码如下。

程序清单：中断服务函数

```
1.  /*************************************************************
2.  功能描述：外部中断服务程序
3.  入口参数：无
4.  返回值：无
5.  *************************************************************/
6.  void INT2_Isr (void) interrupt INT2_VECTOR
7.  {
8.      ;         //无任务
9.  }
```

最后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制红灯亮蓝灯灭，约 5s 后控制红灯灭蓝灯亮、对 INT2 进行初始化、配置使单片机进入掉电模式。

代码清单：主函数

```
1.  int main()
2.  {
```

```
3.  ////////////////////////////////////////////////
4.  //注意: STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
5.  //       高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
6.  //相关 IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
7.  //         P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
8.  ////////////////////////////////////////////////
9.      P0M1 &= 0x3F;   P0M0 &= 0x3F;     //设置 P0.6、P0.7 为准双向口
10.
11.     SleepDelay = 0;               //初始清零
12.     EA = 1;                         //允许总中断
13.
14.     while(1)
15.     {
16.         delay_ms(1);                  //延时 1ms
17.         if(++msecond >= 1000)          //约 1 秒到
18.         {
19.             msecond = 0;                  //清 1000ms 计数
20.              led_on(LED_2);              //点亮红色指示灯 DS2
21.             led_off(LED_1);            //熄灭蓝色指示灯 DS1
22.
23.             if(++SleepDelay >= 5)         //约 5 秒到
24.             {
25.                 SleepDelay = 0;          //清零延时用时间变量
26.                 if(TRUE)
27.                 {
28.                     SleepDelay = 0;        //清零延时用时间变量
29.                     led_on(LED_1);           //点亮蓝色指示灯 DS1
30.                     led_off(LED_2);        //熄灭红色指示灯 DS2
31.                     INT2_Init();           //外部中断 2 初始化
32.                     PCON |= 0x02;          //单片机进入掉电模式
33.                     _nop_();
34.                     _nop_();
35.                     _nop_();
36.                 }
37.             }
38.         }
39.     }
40. }
```

4.6.3. 硬件连接

本实验需要使用 P0.6 驱动蓝色指示灯 DS1、P0.7 驱动红色指示灯 DS2，使用 P3.6（外部中断 2 引脚）连接到触摸按键 S4，因此需要按下图所示连接杜邦线和短路帽。

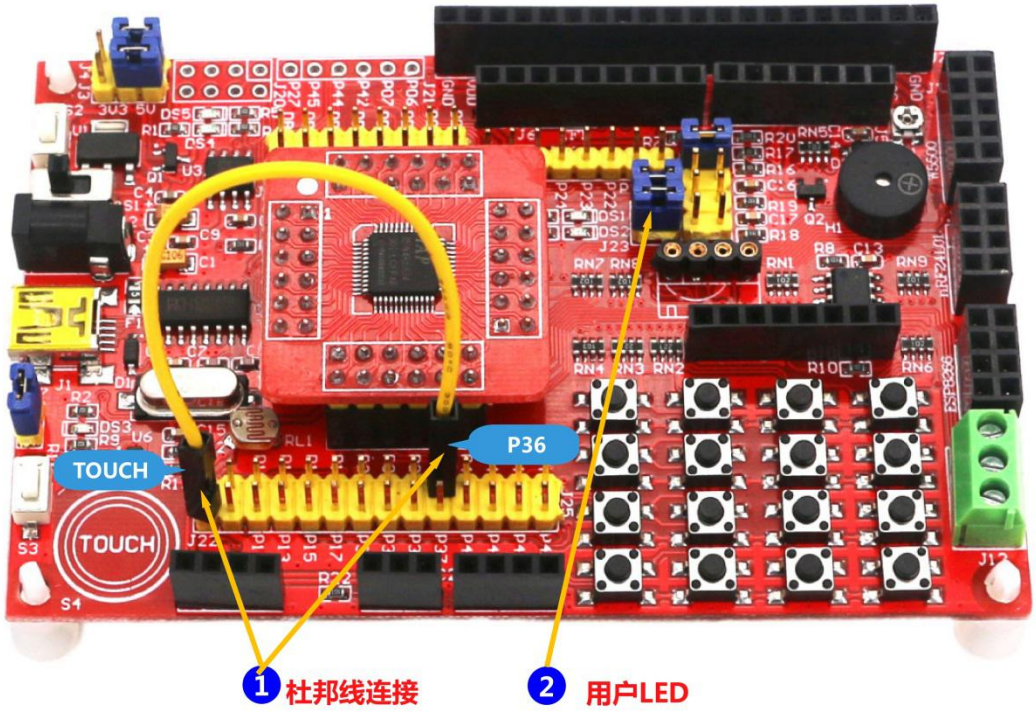

图 12：开发板连接图

4.6.4. 实验步骤

1. 解压“…\第 3 部分：配套例程源码\ 1 - 基础实验程序\”目录下的压缩文件“实验 2-4-2：外部中断 0（上升沿或下降沿中断方式）”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\WAKEUP-INT2_FAL\projec”目录下的工程“WAKEUP-INT2_FAL.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WAKEUP-INT2_FAL.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序 。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：

   1）红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机进入掉电模式；

   2）按下触摸按键在 P36 上产生上升沿，单片机依然处于掉电模式，红灯灭蓝灯亮；

   3）触摸按键由按下到松开，在 P36 上产生下降沿，唤醒单片机，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机再次进入掉电模式。

4.7.单片机掉电唤醒 - 定时唤醒实验

✧ 注：本节的实验源码是在“实验 2-6-1：单片机掉电唤醒 - 外部中断 0（下降沿方式）”的基础上修改。本节对应的实验源码是：“实验 2-6-6：单片机掉电唤醒 - 定时唤醒”。

4.7.1. 工程需要用到的 c 文件

本例需要用到的 c 文件如下表所示，工程需要添加下表中的 c 文件。

表 5：实验需要用到的 c 文件

序号	文件名	后缀	功能描述
1	led	.c	包含与用户 led 控制有关的用户自定义函数。
2	wkt	.c	内部掉电唤醒定时器有关的用户自定义函数。
3	delay	.c	包含用户自定义延时函数。

4.7.2. 头文件引用和路径设置

■ 需要引用的头文件

```
1. #include "delay.h"
2. #include "led.h"
3. #include "wkt.h"
```

■ 需要包含的头文件路径

本例需要包含的头文件路径如下表：

表 6：头文件包含路径

序号	路径	描述
1	..\ Source	led.h、wkt.h 和 delay.h 头文件在该路径，所以要包含。
2	..\User	15W4KxxS4.h 头文件在该路径，所以要包含。

MDK 中点击魔术棒，打开工程配置窗口，按照下图所示添加头文件包含路径。

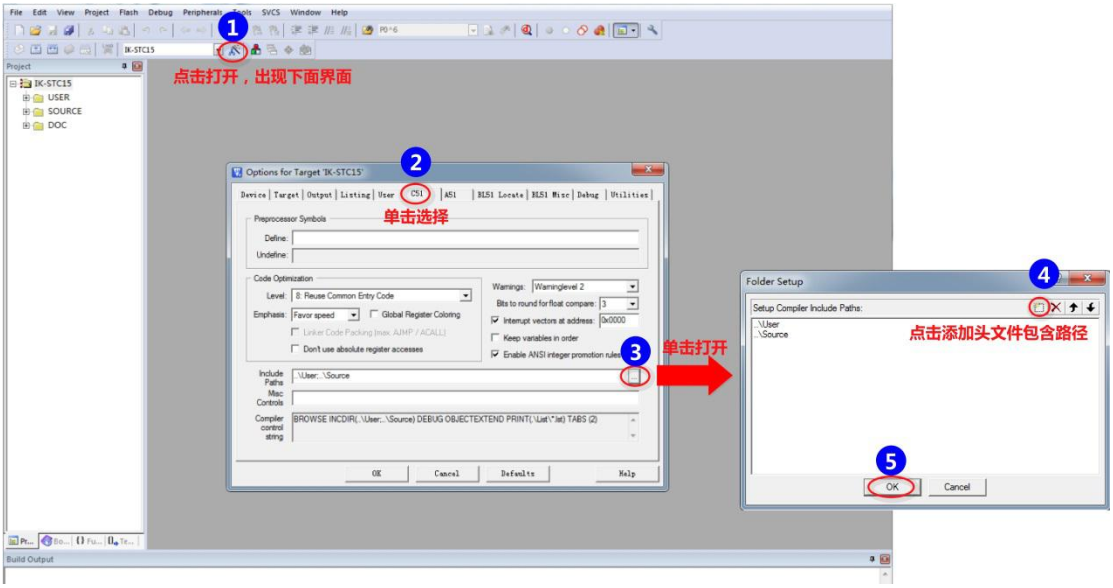

图 13：添加头文件包含路径

4.7.3. 编写代码

首先，在 wkt.c 文件中编写掉电唤醒定时器计数寄存器配置函数，可以实现将单片机内部掉电唤醒定时器开启并对 15 位计数器赋初值，代码如下。

程序清单：掉电唤醒定时器计数寄存器配置函数

```
1.  /*************************************
2.  功能描述：唤醒定时器计数寄存器配置
3.  入口参数：uint16 SetTime：实际计数值
4.  返回值：无
5.  *************************************/
6.  void    SetWakeUpTime(uint16 SetTime)
7.  {
8.      if(SetTime > 0)  SetTime=SetTime-1; //待写入 WKTCH、WKTCL 寄存器值比实际计数值少 1
9.      WKTCL = (uint8)SetTime;                    //赋值 WKTCL 寄存器
10.     WKTCH = (uint8)(SetTime >> 8) | 0x80;     //赋值 WKTCH 寄存器并使能掉电唤醒定时器
11. }
```

然后，在主函数中对 P0.6 和 P0.7 口进行模式配置，先控制红灯亮蓝灯灭，约 5s 后控制红灯灭蓝灯亮、开启掉电唤醒定时器并赋值计数器、配置使单片机进入掉电模式。

代码清单：主函数

```
1.  int main(void)
2.  {
3.  ///////////////////////////////////////////////
4.  //注意：STC15W4K32S4 系列的芯片,上电后所有与 PWM 相关的 IO 口均为
5.  //       高阻态,需将这些口设置为准双向口或强推挽模式方可正常使用
6.  //相关 IO：P0.6/P0.7/P1.6/P1.7/P2.1/P2.2
```

```
//        P2.3/P2.7/P3.7/P4.2/P4.4/P4.5
//////////////////////////////////////////////////
    P0M1 &= 0x3F;   P0M0 &= 0x3F;     //设置 P0.6、P0.7 为准双向口

    SleepDelay = 0;                   //初始清零

    while(1)
    {
        delay_ms(1);                      //延时 1ms
        if(++msecond >= 1000)              //约 1 秒到
        {
            msecond = 0;                     //清 1000ms 计数
            led_on(LED_2);                  //点亮红色指示灯 DS2
            led_off(LED_1);                  //熄灭蓝色指示灯 DS1

            if(++SleepDelay >= 5)            //约 5 秒到
            {
                    SleepDelay = 0;           //清零延时用时间变量
                    led_on(LED_1);            //点亮蓝色指示灯 DS1
                    led_off(LED_2);           //熄灭红色指示灯 DS2
                    SetWakeUpTime(4096);      //约 2 秒后唤醒
                    PCON |= 0x02;             //单片机进入掉电模式
                    _nop_();
                    _nop_();
                    _nop_();
            }
        }
    }
}
```

4.7.4. 实验步骤

1. 解压“…\第 3 部分：配套例程源码\ 1 - 基础实验程序\”目录下的压缩文件“实验 2-4-6：多个外部中断”，将解压后得到的文件夹拷贝到合适的目录，如“D\STC15”。
2. 启动 Keil C51。
3. 在 Keil C51 中执行“Project→Open Project”打开“…\WAKEUP-TIME \projec”目录下的工程“WAKEUP-TIME.uvproj”。
4. 点击编译按钮编译工程。注意查看编译输出栏，观察编译的结果，如果有错误，修改程序，直到编译成功为止。编译后生成的 HEX 文件“WAKEUP-TIME.hex”位于工程目录下的“Output”文件夹中。
5. 打开 STC-ISP 软件下载程序 。下载使用内部 IRC 时钟，IRC 频率选择为 11.0592MHZ。
6. 程序运行后，实验现象及操作步骤如下：

1）红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机进入掉电模式；

2）2s 后，掉电唤醒定时器定时时间到，单片机被从掉电模式下唤醒，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机再次进入掉电模式；

3）2s 后，掉电唤醒定时器定时时间又到，单片机被从掉电模式下唤醒，程序从设置单片机进入掉电模式的下一条语句开始执行，进主循环，红灯亮蓝灯灭，约 5s 后红灯灭蓝灯亮，单片机又一次次进入掉电模式。如此反复，单片机不断进入掉电模式，又不断被定时唤醒。