青风带你玩蓝牙 nRF51822 系列教程

---------- 青风出品必属精品

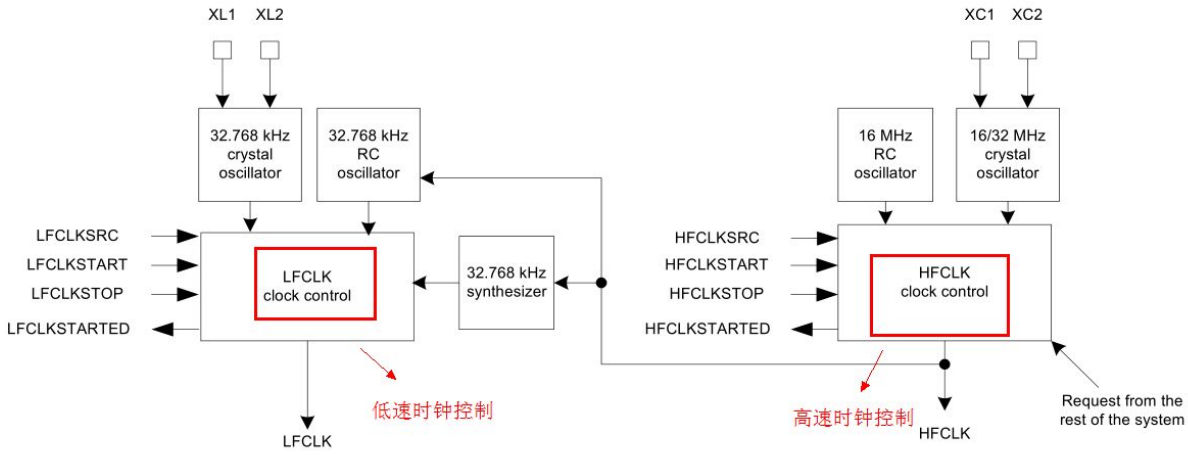
出品论坛：　www.qfv8.com　青风电子社区

作者： 青风
出品论坛: www.qfv8.com
淘宝店：http://qfv5.taobao.com
QQ 技术群：346518370
硬件平台：青云 QY-nRF51822 开发板

2.2 系统时钟设置

蓝牙 nRF51822 虽然是蓝牙设备，但是其内部包含的内核还是 ARM CORTEX M0。你需要知道你的 CPU 跑在什么样的速度，使用什么样的时钟，如何设置。

2.2.1 原理分析：



如上图所示，蓝牙 nRF51822 模块的系统时钟有两部分，一个部分是高速时钟源(HFCLK)，对应 16MHZ 的时钟频率。一个是低速时钟源(LFCLK)，对应 32.768 kHz 的时钟频率。

高速时钟源(HFCLK)：

系统高速时钟源有两种的提供方式：外接 16/32 MHz 外部晶体振荡器或者直接使用内部 16 MHz RC 振荡器。

当接外部 16/32 MHz 外部晶体振荡器时，并联谐振模式下，晶体管脚分别连接 XC1 和 XC2 管脚。如果使用 I32 MHz 晶体，需要设置 XTALFREQ 寄存器

当系统进入 ON 模式，16 MHz RC 振荡器会自动的提供高速时钟给 CPU 和其他其

他需要时钟的外设。

使用外部晶振的时候，通过触发 HFCLKSTART 任务打开外部高速时钟振荡器。触发 HFCLKSTOP 任务关掉外部高速时钟振荡器。

当你一旦打开外部高速时钟振荡器，将会产生 HFCLKSTARTED 事件。同时外部高速时钟振荡器一旦运行，16 MHz RC 振荡器将自动的被关闭。停止外部高速时钟振荡器，16 MHz RC 振荡器将会被切换回来。

当系统不需要使用 16NHZ 的时钟的时候，为了省电，系统会自动的把 16 MHz RC 振荡器关掉：比如 CPU 休眠，或者外设要求 HFCLK 停止工作。如果不需要省电，16 MHz RC 振荡器会自动重启。因此最好是通过 16 MHz RC 振荡器提供高速时钟。

但是当使用 RADIO 和 32.768 时钟校准功能时，高速时钟必须由外部高速时钟振荡器提供。

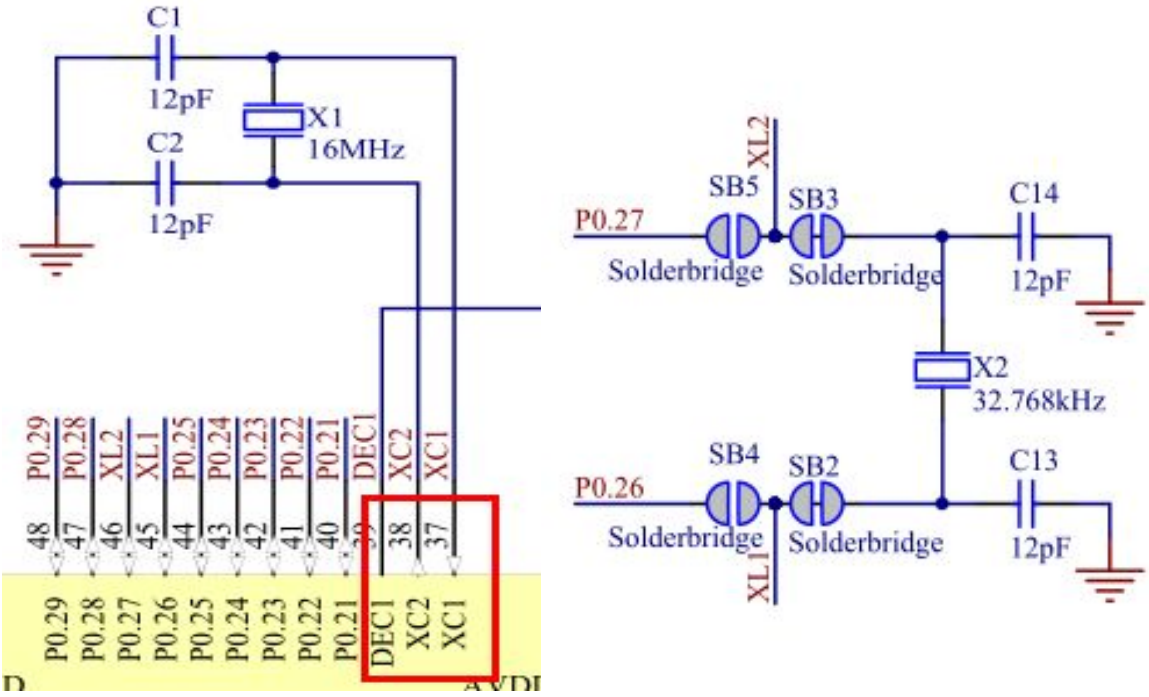
默认情况下，当传播模式从 OFF 到 ON 状态，低速时钟源将被关闭。

低速时钟源(LFCLK)：

系统高速时钟源有两种的提供方式：外接 32.768KHz 外部晶体振荡器或者直接使用内部 32.768KHz RC 振荡器。详细的讲解将在低功耗部分说明。

2.1.2 硬件准备：

如下图所示： 青云 nrf51822 开发板上，通过管脚 XC1 和管脚 XC2 连接 16MHZ 外部晶振，通过管脚 P0.27 和管脚 P0.26 连接 32.768KHZ 晶振。



2.2.3 软件准备以及编写：

我们使用前面一节介绍的 LED 灯的工程进行说明，主要是探讨下官方给出的这个延迟函数的使用;工程目录树如下图所示：

上图红色框框中的几个文件都是 ST 官方给我们编好的库函数。首先看看系统设置文件 system_nrf51.c 文件。这个文件设置了系统设置默认状态下的频率：

```
#define __SYSTEM_CLOCK          (16000000UL)        /*系统时钟设为 16MHZ


#if defined ( __CC_ARM )
        uint32_t SystemCoreClock __attribute__((used)) = __SYSTEM_CLOCK;
#elif defined ( __ICCARM__ )
        __root uint32_t SystemCoreClock = __SYSTEM_CLOCK;
#elif defined     ( __GNUC__ )
        uint32_t SystemCoreClock __attribute__((used)) = __SYSTEM_CLOCK;
#endif

void SystemCoreClockUpdate(void)
{
        SystemCoreClock = __SYSTEM_CLOCK;
}
```

默认状态下跑在 16MHZ，也就是 CPU 的运行速度，这里面在延迟函数文件里得到了体现，打开 delay 文件，看下面的延迟 1US 的方式：

```
static __ASM void __INLINE nrf_delay_us(uint32_t volatile number_of_us)
{
loop
            SUBS      R0, R0, #1 //2 个字节
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP
```

```
28.             NOP
29.             NOP
30.             NOP
31.             NOP             //12 个空操作
32.             BNE      loop//1 个字节
33.             BX       LR//分支跳转到链接寄存器  1 个字节
34. }
35.
```

整个延迟 1us 操作实际上就是 16 个字节，也就是花了 16 个时钟周期。16*(1/16mhz)=1us。因此可以确定系统工作在 16MHZ 状态下。