

应用笔记

AN_N32G43x_N32L40x_N32L43x_USB 无晶体模式应用笔记

简介

在嵌入式产品开发中，USB 设备会经常使用到，由于 USB 的时钟精度要求，大部分情况下会使用外部高速晶体作为时钟源以保证时钟精度，当然在一些应用中，没有外部高速晶体，此时可以使用 USB 无晶体模式以保证 USB 时钟精度和 USB 正常传输数据。

文档主要介绍 USB 无晶体模式，MCU 无需接外部高速晶体，本文档仅适用于国民技术 MCU 产品，目前支持的产品系列有 N32G43x 系列、N32L40x 系列和 N32L43x 系列。

目录

1. 概述..... 3

2. 配置系统时钟和 UCDR 模块 4

 2.1 修改系统时钟..... 4

 2.2 选择无晶体模式..... 4

 2.3 配置 UCDR 模块 5

3. DEMO 演示 7

4. 版本历史..... 8

5. 声明..... 9

1.概述

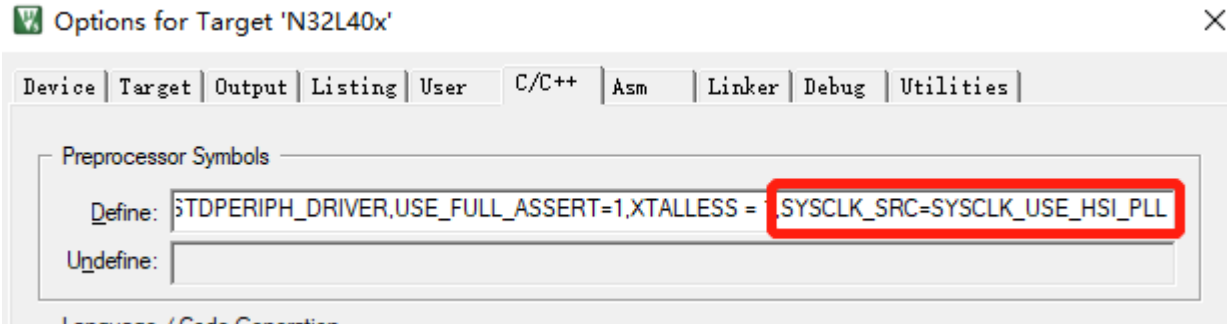
USB2.0 全速协议传输速率为 12Mbps ($12\text{Mbps} \pm 0.025\%(2500\text{ppm})$), USB 的系统时钟是传输速率的四倍, 即 48MHz。为得精确的 48MHz 时钟, 在 N32G43x 系列、N32L40x 系列和 N32L43x 系列有两种方式可以实现, 一种是使用外部 HSE 晶体, 通过倍频再分频得到精确的 48MHz 时钟; 另外一种是使用芯片内部 UCDR 模块, 得到精确的 48MHz 时钟。

本文档主要介绍通过芯片内部 UCDR 模块得到精确的 48MHz 时钟, 下文以 N32L40x 系列为例介绍。

2.配置系统时钟和 UCDR 模块

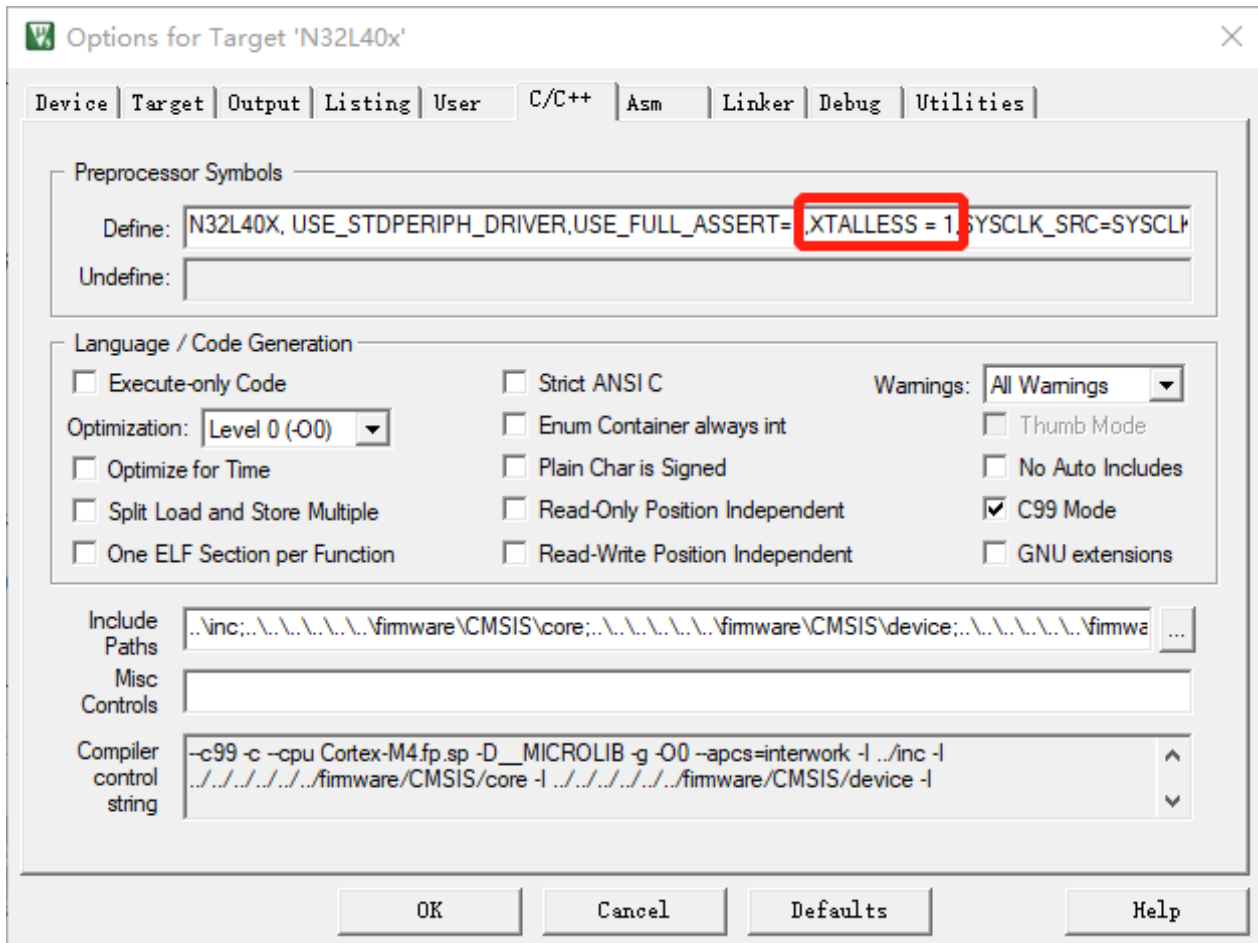
2.1 修改系统时钟

由于没有外部高速晶体，系统时钟选择 SYSCLK_USE_HSI_PLL。



2.2 选择无晶体模式

可以通过设置宏“XTALLESS = 1”来选择无晶体模式。



2.3 配置 UCDR 模块

选择 USB 为无晶体模式，配置并使能 UCDR 模块。

```

/**
 * @brief SB_XtallessIni.
 * @param RCC_UCDR300MSource: ucdr.source
 * @return USB_Clock.set.Status.
 */
void USB_XtallessIni(uint32_t RCC_UCDR300MSource)
{
    uint32_t time;

    /* Check the parameters */
    assert_param(IS_RCC_UCDR300M_SRC(RCC_UCDR300MSource));

    RCC->APB1PCLKEN |= RCC_APB1PCLKEN_AFECEN;
    /* Clear UCDR300MSEL bits */
    RCC->CFG3 &= RCC_UCDR300MSource_MASK; // RCC_UCDR300M_SRC_MASK;
    /* Set UCDR300MSEL bits */
    RCC->CFG3 |= RCC_UCDR300MSource;
    /* Select the USB Crystal Mode */
    RCC->CFG3 |= RCC_USBXTALESS_LESSMODE;

    /* Enable LDO for OSC UCDR */ →
    _EnOsc300Ldo();
    time = 0x4000;
    while(time--);
    /* Enable iBias for OSC UCDR */ →
    _EnOsc300Ibias();
    time = 0x4000;
    while(time--);
    /* Enable Core for OSC UCDR */ →
    _EnOsc300Core();
    time = 0x4000;
    while(time--);

    /* Enable UCDR */
    RCC->CFG3 |= RCC_UCDR_ENABLE;
    time = 0x4000;
    while(time--);

    return;
}

```

UCDR 配置完成后，会对 USB host 输入的总线信号进行检测，得到精确的 USB 数据 bit 位宽，再通过 OSC300MHz 分频得到精确的 USB 通信时钟，USB 的初始化流程与有晶体模式相同。

UCDR 配置完成后，先不开启 ESOF 中断，待收到 SOF 帧后，再开启 ESOF 中断，如图 2-1 所示；

为保证 USB 通信质量，在以下情况需要对 UCDR 模块进行复位：

1. 刚插入 USB 设备时，由于信号不稳定，在每次枚举完成后，收到第一个 SOF 帧时复位 UCDR 模块，如图 2-1 所示；

图 2-1

```
#if (IMR_MSK & STS_SOF)
... if (wIstr & STS_SOF & wInterrupt_Mask)
... {
...     _SetISTR((uint16_t)CLR_SOF);
...     bIntPackSOF++;
...     if (USB_SET_CONFIGED_FLAG == true)
...     {
...         USB_SET_CONFIGED_FLAG = false;
...     }
...     if (XTALLESS == 1)
...     {
...         RCC->APB1PRST |= RCC_APB1PRST_UCDRRST;
...         RCC->APB1PRST &= ~RCC_APB1PRST_UCDRRST;
...     }
...     _SetISTR((uint16_t)CLR_ESOF);
...     wInterrupt_Mask = IMR_MSK1;
...     _SetCNTR(wInterrupt_Mask);
... }
... if (EsofUcdrDis == 0xAA)
... {
...     EsofUcdrDis = 0;
... }
```

Reset UCDR

Enable ESOF interrupt

Enable ESOF interrupt reset UCDR

2. 在接收到 ESOF 中断后，复位 UCDR 模块，为防止重复复位 UCDR 模块，在 ESOF 中断中完成 UCDR 复位后，需要在接收到 SOF 帧时，再开启 ESOF 中断中的复位 UCDR 模块，如图 2-1 和图 2-2 所示；

图 2-2

```
#if (IMR_MSK1 & STS_ESOF)
... if (wIstr & STS_ESOF & wInterrupt_Mask)
... {
...     /* clear ESOF flag in STS */
...     _SetISTR((uint16_t)CLR_ESOF);
...     if ((_GetFNR() & FN_RXDP) != 0)
...     {
...         /* increment ESOF counter */
...         esof_counter++;
...         if (EsofUcdrDis == 0)
...         {
...             EsofUcdrDis = 0xAA;
...         }
...     }
...     if (XTALLESS == 1)
...     {
...         RCC->APB1PRST |= RCC_APB1PRST_UCDRRST;
...         RCC->APB1PRST &= ~RCC_APB1PRST_UCDRRST;
...     }
... }
```

Reset UCDR

3. USB 挂起后，复位 UCDR 模块，如图 2-3 所示；

图 2-3

```
void Suspend(void)
{
... uint32_t i = 0;
... uint16_t wCNTR;
... /* suspend preparation */
... /* ... */
... if (XTALLESS == 1)
... {
...     RCC->APB1PRST |= RCC_APB1PRST_UCDRRST;
...     RCC->APB1PRST &= ~RCC_APB1PRST_UCDRRST;
... }
... }
```

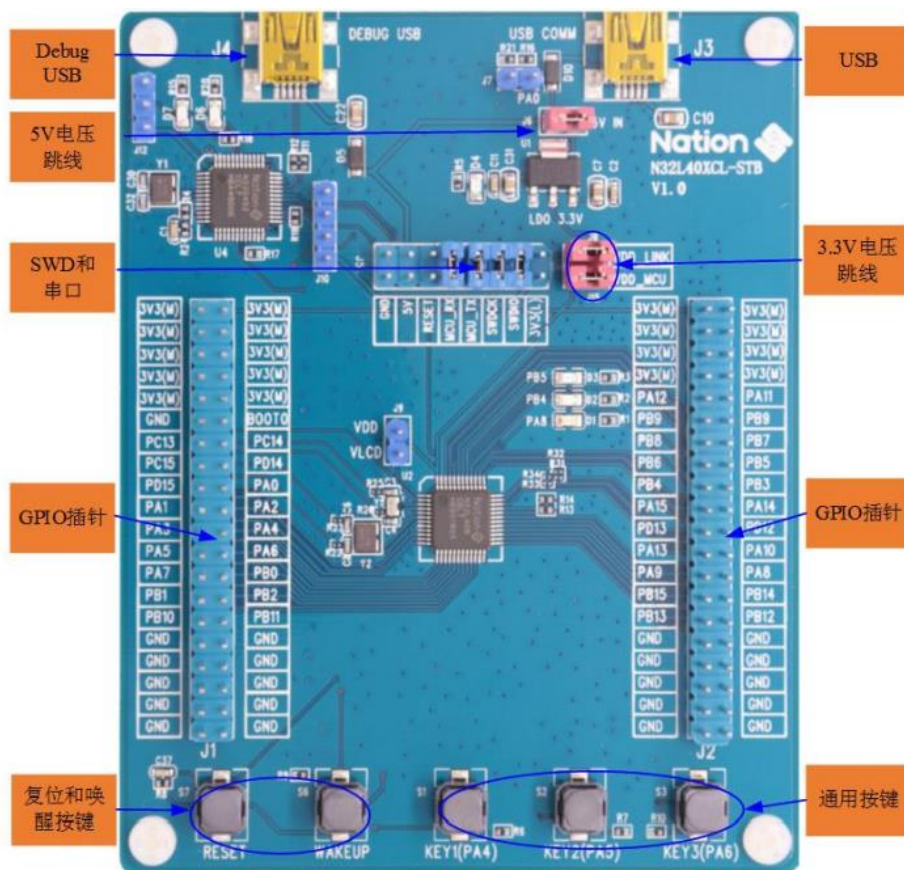
Reset UCDR

其他具体操作代码可参考应用笔记中的 DEMO。

3.Demo 演示

选择 N32L40x 系列最小系统开发板 N32L40XCL-STB V1.0，如图 3-1 是 N32L40XCL-STB V1.0 最小系统开发板，开发板的使用指南请查看《UG_N32L40XCL-STB 开发板硬件使用指南》

图 3-1 开发板布局



开发板上 J4 为 USB 下载调试接口，J3 为 USB 设备接口，将代码编译通过 J4 下载到板子后复位运行，再将 J3 口接到电脑上，可以看到电脑识别出键盘设备，按下 KEY1(PA4)按钮，电脑输入“a”。

4.版本历史

日期	版本	修改
2022/07/26	V1.0	初始版本

5. 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用者承担，同时使用者应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证责任，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。