Application Note

Crystal-less Application Note

Introduction

USB interfaces are commonly used in embedded product development. Due to the high clock accuracy requirements of USB interfaces, an external high-speed crystal oscillator is often used as the clock source to ensure clock accuracy. Of course, in some applications, there is no external high-speed crystal oscillator, at this point you can use the USB crystal-less mode to ensure USB clock precision and the normal transmission of USB data.

This document mainly introduces the USB crystal-less mode, MCU does not need to connect to high-speed external crystal Oscillator.

This document is only applicable to NSING MCU products, currently supported product series are N32G43x series, N32L40x series and N32L43x series.

Contents

1. Overview

The USB2.0 full-speed protocol specifies a transfer rate of 12Mbps (12Mbps ± 0.025%(2500ppm)). In our chip, the USB core clock is four times the transfer rate, which is 48MHz. For a high-precision 48MHz clock, there are two ways to achieve this in N32G43x series, N32L40x series and N32L43x series. One is to use external HSE crystal to obtain an accurate 48MHz clock by frequency doubling and frequency division. The other is to use the UCDR module inside the chip to get an accurate 48MHz clock.

This document describes how to obtain an accurate 48MHz clock by using the UCDR module inside the chip.

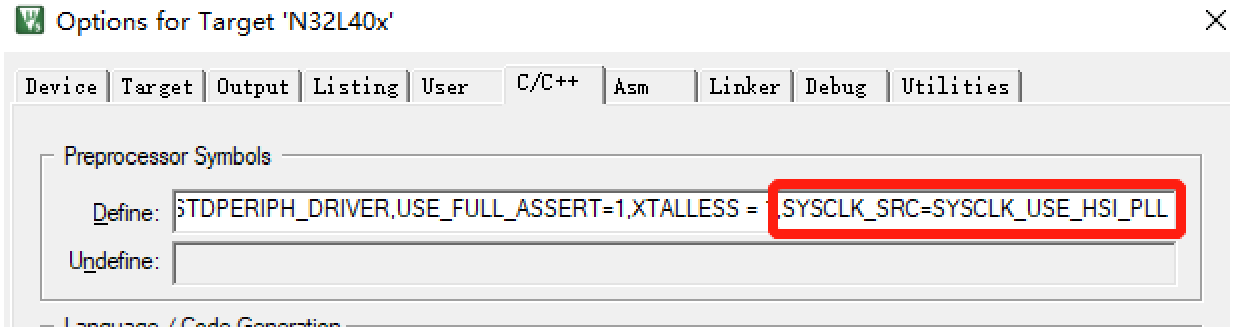
The following uses the N32L40x series as an example.

2. Configure the Sytem Clock And UCDR Module

2.1 Modify System Clock

As there is no external high-speed crystal oscillator, the system clock is selected as SYSCLK_USE_HSI_PLL.

Figure 2-1 Options for Symbols



Options for Target 'N32L40x'

Preprocessor Symbols

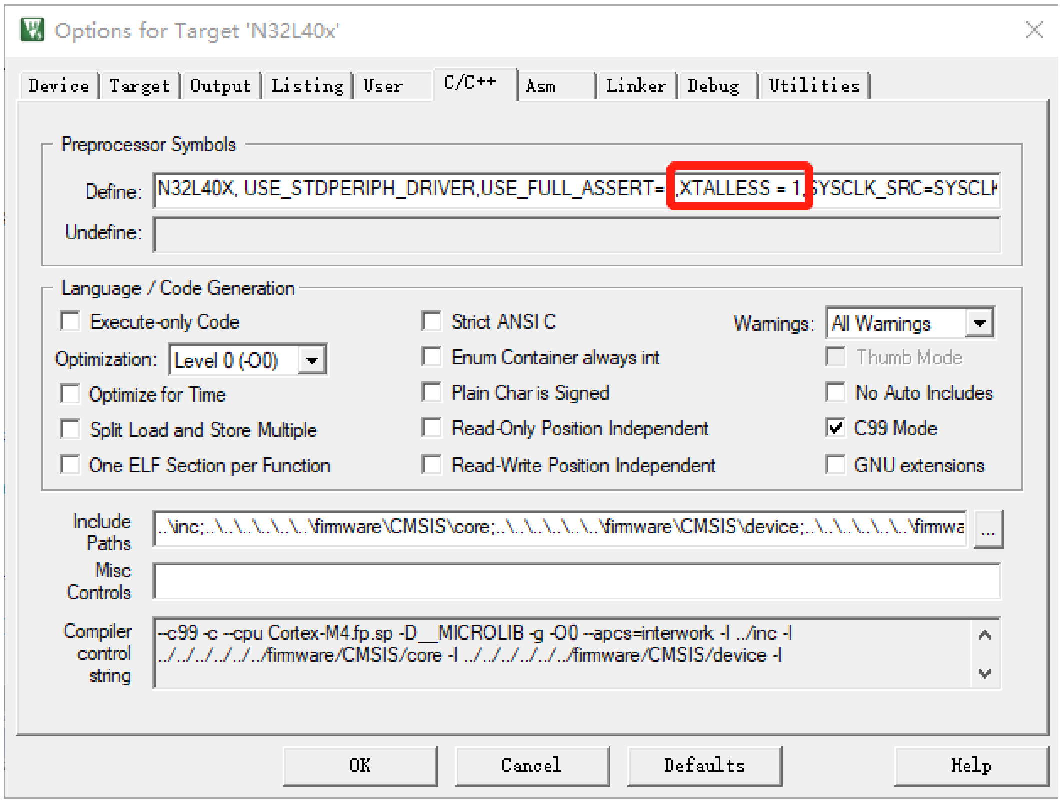
Define: STDPERIPH_DRIVER,USE_FULL_ASSERT=1,XTALLESS = 1,SYSCLK_SRC=SYSCLK_USE_HSI_PLL

Undefine:

2.2 Select Crystal-less Mode

The crystal-lessmode can be selected by setting the macro "XTALLESS = 1".

Figure 2-2 Options for Xtall-less



Options for Target 'N32L40x'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

Preprocessor Symbols

Define: N32L40X, USE_STDPERIPH_DRIVER,USE_FULL_ASSERT=,XTALLESS = 1,SYSCLK_SRC=SYSCLK

Undefine:

Language / Code Generation

- ☐ Execute-only Code
- Optimization: Level 0 (-O0)
- ☐ Optimize for Time
- ☐ Split Load and Store Multiple
- ☐ One ELF Section per Function
- ☐ Strict ANSI C
- ☐ Enum Container always int
- ☐ Plain Char is Signed
- ☐ Read-Only Position Independent
- ☐ Read-Write Position Independent
- Warnings: All Warnings
- ☐ Thumb Mode
- ☐ No Auto Includes
- ☑ C99 Mode
- ☐ GNU extensions

Include Paths: ..\inc;..\..\..\..\..\..\firmware\CMSIS\core;..\..\..\..\..\..\firmware\CMSIS\device;..\..\..\..\..\..\firmwa

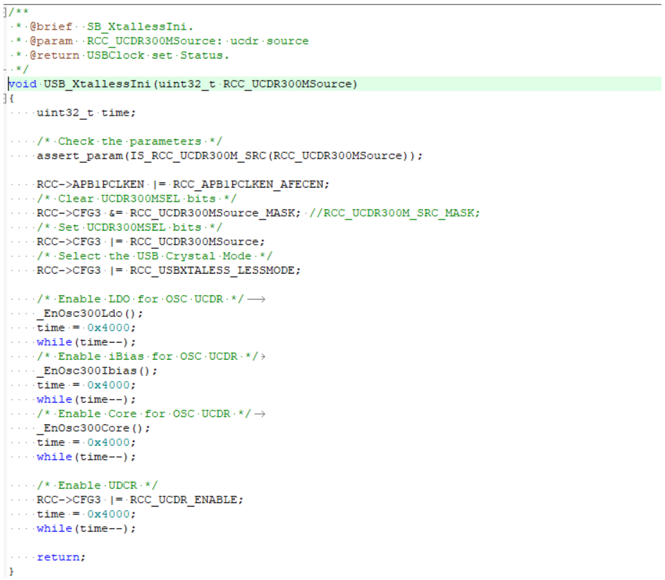
Misc Controls:

Compiler control string:

```
--c99 -c --cpu Cortex-M4.fp.sp -D__MICROLIB -g -O0 --apcs=interwork -I ../inc -I
../../../../../../firmware/CMSIS/core -I ../../../../../../firmware/CMSIS/device -I
```

2.3 Configure UCDR Module

Figure 2-3 Crystal-less Init



```c
/**
 * @brief  SB_XtallessIni.
 * @param  RCC_UCDR300MSource: ucdr source
 * @return USBClock 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 UDCR */
    RCC->CFG3 |= RCC_UCDR_ENABLE;
    time = 0x4000;
    while(time--);

    return;
}
```

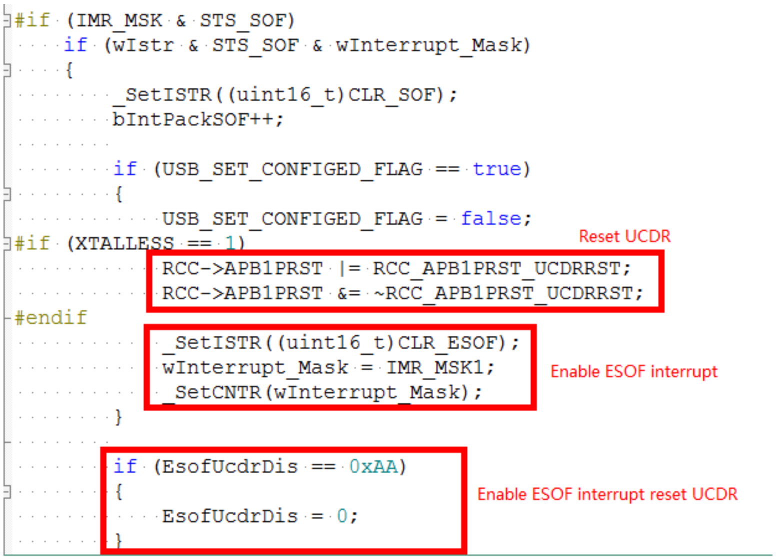
After the UCDR configuration is completed, it will detect the bus signals input by the USB host to obtain the precise USB data bit width time. Then, by dividing the OSC 300MHz, a clock with the same width as this data bit time is obtained, which can serve as the USB communication clock. The USB of crystal-less initialization process is the same as in xtal mode.

After the UCDR configuration is complete, don’t enable ESOF interrupt first. Wait until an SOF frame is received, then enable the ESOF interrupt, see Figure 2-4 **Crystal-less Mode**.

To ensure the quality of USB communication, reset the UCDR module in the following cases:

- When the USB device is just inserted, the UCDR module is reset when the first SOF frame is received after each enumeration because the signal is unstable, seeFigure 2-4;

Figure 2-4 Crystal-less Mode



```
#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;
#endif
            _SetISTR((uint16_t)CLR_ESOF);
            wInterrupt_Mask = IMR_MSK1;
            _SetCNTR(wInterrupt_Mask);
        }

        if (EsofUcdrDis == 0xAA)
        {
            EsofUcdrDis = 0;
        }
```

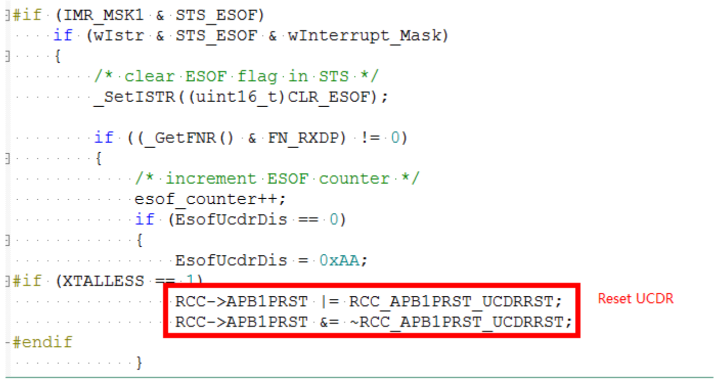
Reset UCDR

Enable ESOF interrupt

Enable ESOF interrupt reset UCDR

- Reset the UCDR module after receiving an ESOF interrupt. To prevent repeated resetting of the UCDR module, after completing the UCDR reset during an ESOF interrupt, restart the UCDR module after receiving the SOF frame during an ESOF interrupt, see Figure 2-4 **Crystal-less Mode** andFigure 2-5.

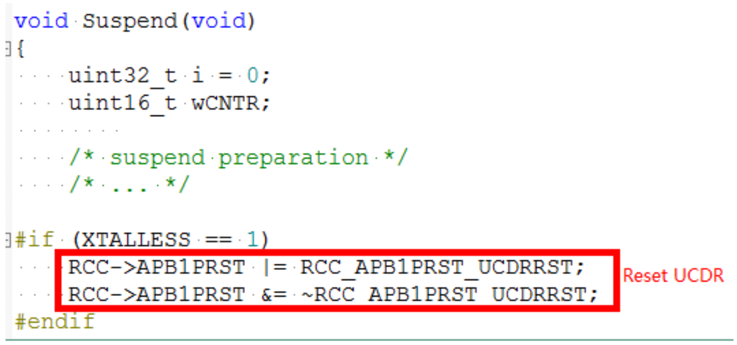
Figure 2-5 Crystal-less Mode



```
#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;
#endif
            }
```

- After the USB is suspend, reset the UCDR module, see Figure 2-6.

Figure 2-6 Crystal-less Mode



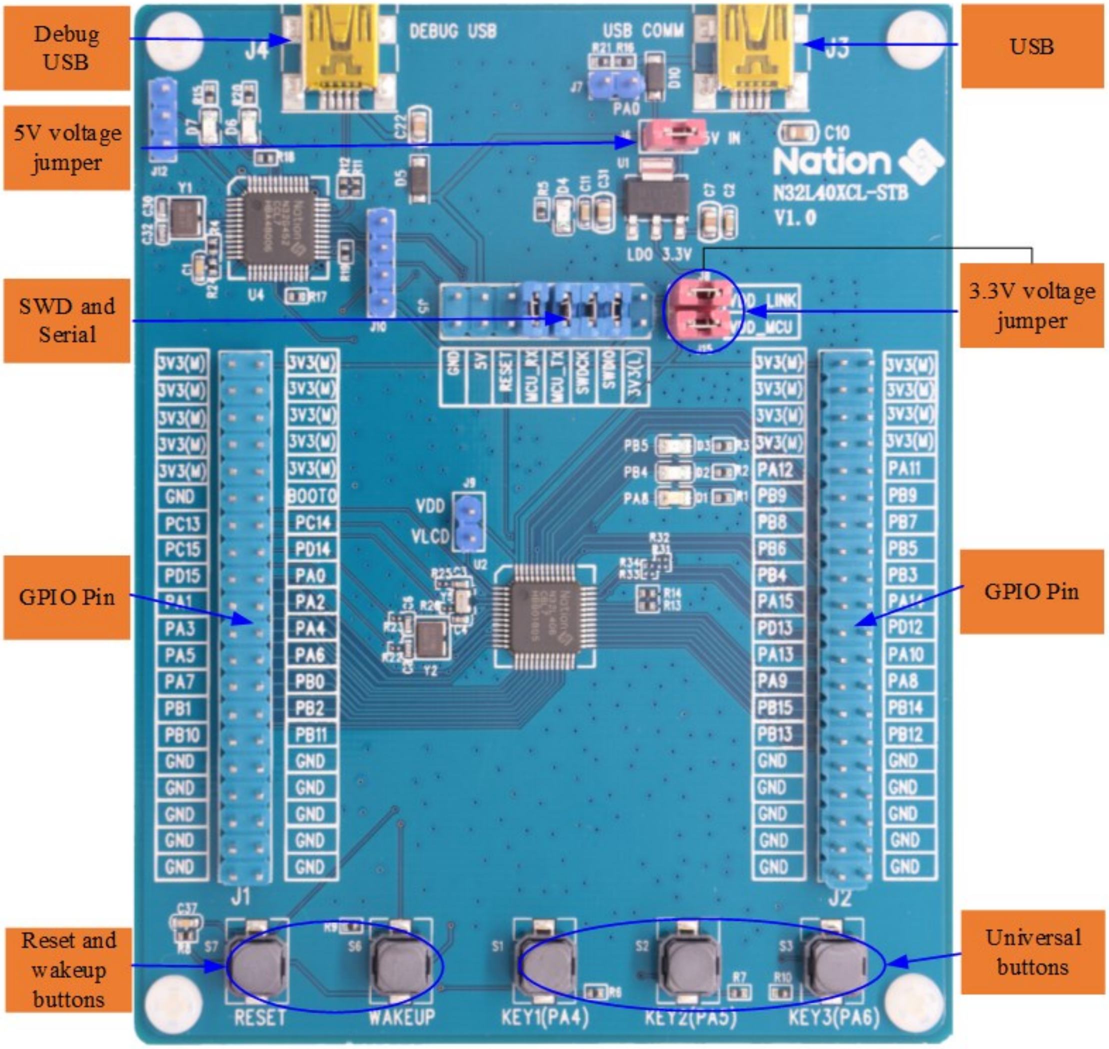
```
void Suspend(void)
{
    uint32_t i = 0;
    uint16_t wCNTR;

    /* suspend preparation */
    /* ... */

#if (XTALLESS == 1)
    RCC->APB1PRST |= RCC_APB1PRST_UCDRRST;
    RCC->APB1PRST &= ~RCC_APB1PRST_UCDRRST;
#endif
```

For more details, see DEMO in Application Notes please.

3. Demo presentation

Select N32L40x series minimum system development board N32L40XCL-STB V1.0. Figure 3-1 shows N32L40XCL-STB V1.0 minimum system development board. Please refer to “UG_N32L40XCL-STB Development Board Hardware Usage Guide” for the use of development board.

Figure 3-1 Development Board Layout



J4 on the development board is the USB download and debugging interface, and J3 is the USB device interface. After the code is compiled and downloaded to the board through J4, reset and run, and then connect J3 port to the computer, it can be seen that the computer recognizes the keyboard device, press KEY1(PA4) button, and the computer enters "a".

4. Version History

Version	Date	Changes
V1.0.0	2022.07.26	Initial release

5. Disclaimer

This document is the exclusive property of NSING TECHNOLOGIES PTE. LTD.(Hereinafter referred to as NSING). This document, and the product of NSING described herein (Hereinafter referred to as the Product) are owned by NSING under the laws and treaties of Republic of Singapore and other applicable jurisdictions worldwide. The intellectual properties of the product belong to Nations Technologies Inc. and Nations Technologies Inc. does not grant any third party any license under its patents, copyrights, trademarks, or other intellectual property rights. Names and brands of third party may be mentioned or referred thereto (if any) for identification purposes only. NSING reserves the right to make changes, corrections. enhancements, modifications, and improvements to this document at any time without notice. Please contact NSING and obtain the latest version of this document before placing orders. Although NATIONS has attempted to provide accurate and reliable information, NATIONS assumes no responsibility for the accuracy and reliability of this document. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. In no event shall NATIONS be liable for any direct, indirect, incidental, special, exemplary, or consequential damages arising in any way out of the use of this document or the Product.

NATIONS Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, Insecure Usage’. Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, all types of safety devices, and other applications intended to supporter sustain life. All Insecure Usage shall be made at user's risk. User shall indemnify NATIONS and hold NATIONS harmless from and against all claims, costs, damages, and other liabilities, arising from or related to any customer's Insecure Usage Any express or implied warranty with regard to this document or the Product, including, but not limited to. The warranties of merchantability, fitness for a particular purpose and non-infringement are disclaimed to the fullest extent permitted by law. Unless otherwise explicitly permitted by NATIONS, anyone may not use, duplicate, modify, transcribe or otherwise distribute this document for any purposes, in whole or in part.