

## Application note

---

# AN\_N32G43x\_N32L43x\_N32L40x\_USB\_Xtal\_Less\_Application\_Note

---

## Introduction

In embedded product development, the USB device is often used, because the USB clock accuracy requirement, and in most cases will use external high speed crystal as the clock source to ensure accuracy, of course, in some applications, there is no external high-speed crystal, at this point you can use the USB xtal\_less mode to ensure USB clock precision and the normal transmission of USB data.

This document mainly introduces the USB xtal\_less mode, MCU does not need to connect to high-speed external crystal, this document is only applicable to nations MCU products, currently supported product series are N32G43x, N32L40x and N32L43x series.

# Contents

<b>Introduction .....</b>	<b>1</b>
<b>1. Overview .....</b>	<b>3</b>
<b>2. Configure the sytem clock and UCDR module .....</b>	<b>4</b>
2.1 Modify system clock .....	4
2.2 Select Xtal_Less mode .....	4
2.3 Configure UCDR module.....	5
<b>3. Demo presentation .....</b>	<b>7</b>
<b>4. Version history .....</b>	<b>8</b>
<b>5. Notice.....</b>	<b>9</b>

## 1. Overview

The USB2.0 full-speed protocol has a transmission rate of 12Mbps ( $12\text{Mbps} \pm 0.025\%(2500\text{ppm})$ ), and the system clock of USB is four times as long as the transmission rate, is 48MHz. In order to obtain an accurate 48MHz clock, there are two ways to achieve it 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.

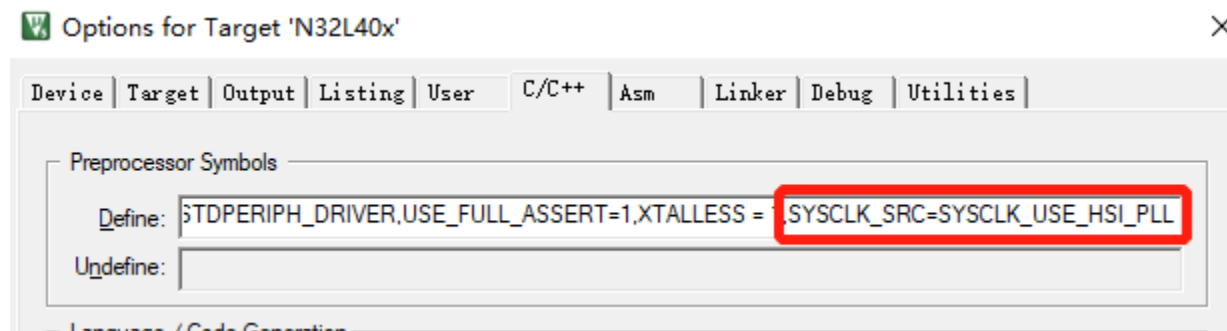
This document describes how to obtain an accurate 48MHz clock by using the UCDR module inside the chip.

The following uses the N32L40x series as an example.

## 2. Configure the sytem clock and UCDR module

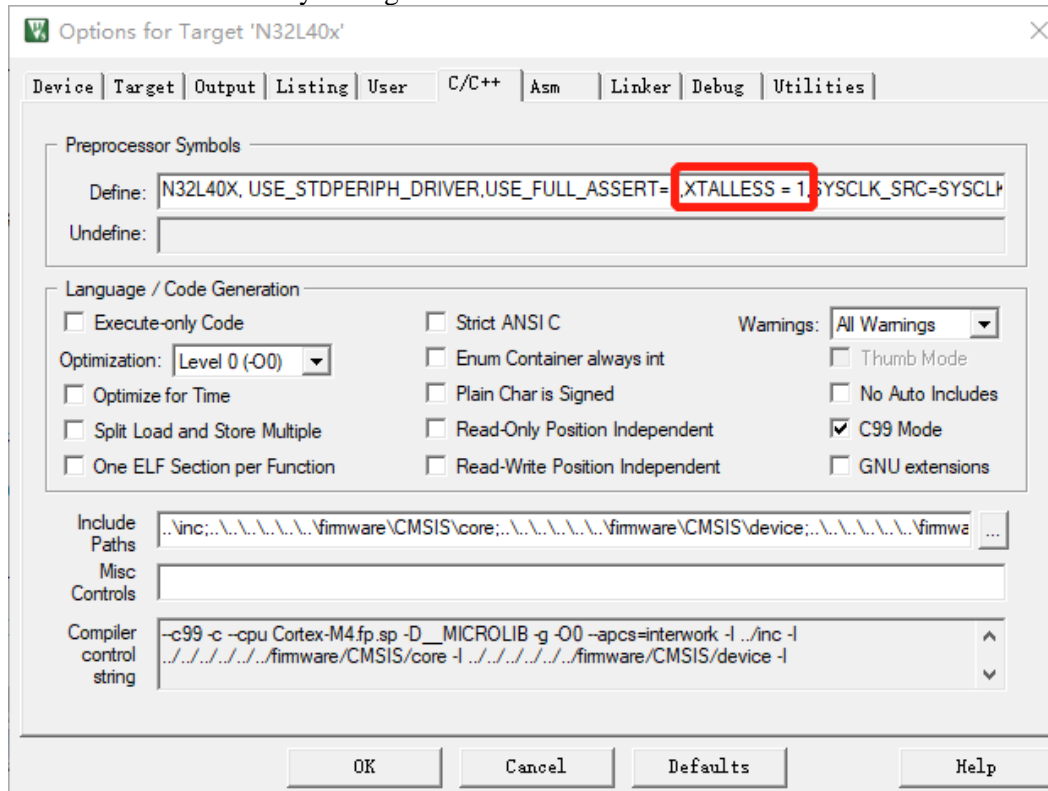
### 2.1 Modify system clock

SYSCLK\_USE\_HSI\_PLL is selected for the system clock for there id no external high-speed crystal.



### 2.2 Select Xtal\_Less mode

The xtal\_less mode can be selected by setting the macro "XTALLESS = 1".



## 2.3 Configure UCDR module

```

/**
 * @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_AFECEEN;
    /* 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;
}

```

After the UCDR configuration is completed, the bus signal input by USB host will be detected to obtain the accurate USB data bit width, and then through OSC300MHz frequency division to obtain the accurate USB communication clock. The USB initialization process is the same as in xtal mode.

After the UCDR configuration is complete, don't enable ESOFF interrupt, and then enable ESOFF interrupt after receiving SOF frame, see Figure 2-1.

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

1. 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, see Figure 2-1;

Figure 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. 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 2-1 and Figure 2-2.

Figure 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. After the USB is suspend, reset the UCDR module, see Figure 2-3.

Figure 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;
...}
...}
```

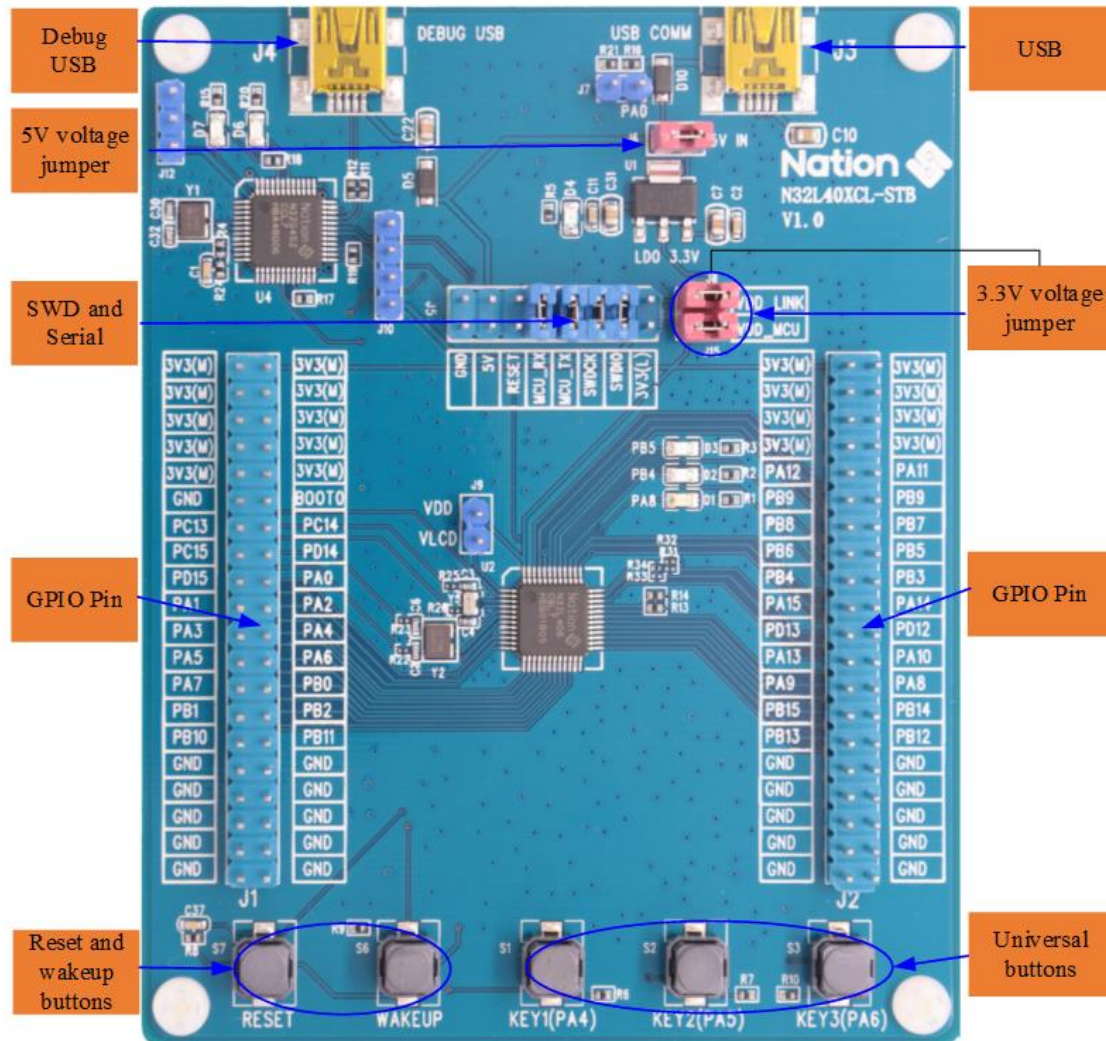
Reset UCDR

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

Date	Version	Remark
2022/07/26	V1.0	New document

## 5. Notice

This document is the exclusive property of Nations Technologies Inc. (Hereinafter referred to as NATIONS). This document, and the product of NATIONS described herein (Hereinafter referred to as the Product) are owned by NATIONS under the laws and treaties of the People's Republic of China and other applicable jurisdictions worldwide.

NATIONS does not grant 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.

NATIONS reserves the right to make changes, corrections, enhancements, modifications, and improvements to this document at any time without notice. Please contact NATIONS 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 support or 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.