

External Clock Configuration Application Note V1.0

Publication Release Date: Sep. 2008

Support Chips:

NUC745A

NUC710A

Support Platforms:

uClinux-2.4.x

uClinux-2.6.x

Bootloader



The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

Table of Contents

1. Introduction	4
1.1. Timer.....	4
1.2. UART.....	4
1.3. PLL	4
2. Bootloader	5
2.1. PLL	5
2.2. Timer.....	5
2.3. UART.....	5
3. uClinux-2.4.x.....	6
3.1. Timer.....	6
3.2. UART.....	6
4. uClinux-2.6.x.....	8
4.1. Timer.....	8
4.2. UART.....	8
5. Revision History	10

1. Introduction

NUC710A and NUCA745 EV boards use 15 MHz crystal as external clock source. So the drivers in BSP configure some IPs using 15MHz as clock source. If other frequency is used as external clock source, these drivers must be updated according to the external clock frequency. This application note describes how to make the changes for bootloader, uClinux-2.4.x and uClinux-2.6.x.

There are some IPs are directly effected by external clock, Timer, UART, and PLL. Bootloader needs to configure all of the setting, and uClinux kernel needs to configure first two of them.

1.1. Timer

To keep timer works in the same frequency, register **TICR0** and/or **TICR1** need to adjust the initial value according to the external clock. For example, assume the *PRESCALE* is 0, to fix the timer frequency to 100 HZ, these register should be set to 150000 while external clock is 15 MHz, 120000 while external clock is 12 MHz.

1.2. UART

UART baudrate is calculated from following equation:

Baudrate = External clock / (16 * (Divisor + 2)), where Divisor is (**UART_DLM** & 0xff) << 8 + (**UART_DLL** & 0xff).

Please note not every frequent used baudrate could be generated from every external clock frequency. For example, while the external clock frequency is 16 MHz, UART is impossible to generate 115200 bps. From previous equation, the output baudrate is 111111bps and 125000bps while divisor is 6 and 7 respectively. The deviation could make the UART failed to receive and transmit with peer.

1.3. PLL

System clock could switch between external clock and PLL by set or clear bit *ECLKS* of register **CLKSEL**. After power on reset, the content of *ECLKS* is the Power-On Setting value. If the external clock frequency is faster than 15 MHz, this bit must be clear after boot up. Otherwise system clock will be faster than 80 MHz and caused system unstable. To change system clock to 80 MHz while external clock is not 15 MHz, please follow the steps list below:

1. Make sure bit *ECLKS* is cleared.
2. Set proper value to register **PLLCON** according to external clock.
3. Wait for at least 2ms and set bit *ECLKS*.

2. Bootloader

2.1. PLL

Below is an example to enable PLL as system clock. Users can call it in any location of bootloader

```
// Make sure ECLKS is cleared
*(unsigned int volatile*)(0xfff0000c) &= ~0x10;
// config PLL, 0x2C01 is for 16 MHz crystal
*(unsigned int volatile*)(0xfff00008) = 0x2C01;
// Delay 10 ms
sleep(1);
// enable PLL
*(unsigned int volatile*)(0xfff0000c) |= 0x10;
```

2.2. Timer

Timer is configured two places in shell/timer.c. Simply update the value given to register **TICR0**, and timer will keep its original frequency.

2.3. UART

Two sets of UART baudrate divisors are defined in bootloader header files. Please make sure to keep them consistent. The divisors are defined in WBLV1_1/Inc/platform.h and Embedded/Inc/platform.h. They are defined as “#define ARM_BAUD_baudrate (divisors)”. Below is an example.

```
#define ARM_BAUD_38400 (24) // Baudrate 38400 bps while ext clock is 16 MHz
```

3. uClinux-2.4.x

3.1. Timer

Two files need to be modified for changing external clock. They're listed below

```
include/asm-armnommu/arch-<IC NAME>/time.h

void __inline__ setup_timer(void)
{
    .....
    // <init value> should be external clock / (100 * 100)
    CSR_WRITE(TICR0, <init value>);
    .....
}
```

```
Include/asm-armnommu/arch-<IC NAME>/hardware.h

// for example, 16 * MHz for 16 MHz external crystal
#define fMCLK_MHz    ( <external clock/ 1M> * MHz)
```

3.2. UART

UART drivers and one header file needs to be update for new external clock::

```
include/asm-armnommu/arch-<IC NAME>/serial.h

static struct serial_buadtable uart_baudrate[] =
{
    {1200 , <divisor for 1200bps>},
    .....
    {460860, <divisor for 460860bps>}
}
```

```
}
drivers/char/<IC NAME>_uart_<NUM>.c

static void autoconfig(struct serial_state *state)
{
    .....
    serial_outp(info, COM_DLL_<NUM>, <UART_DLL for default baud>);
    serial_outp(info, COM_DLM_<NUM>, <UART_DLM for default baud>);
    .....
}

static int __init serial_console_setup(struct console *co, char *options)
{
    .....
    serial_outp(info, COM_DLL_<NUM>, <UART_DLL for default baud>);
    serial_outp(info, COM_DLM_<NUM>, <UART_DLM for default baud>);
    .....
}
```

4. uClinux-2.6.x

4.1. Timer

Two files are related to timer. The codes need to be modified are listed below:

```
linux-2.6.x/arch/armnommu/mach-<IC NAME>/time.c

void __init <IC NAME>_time_init(void)
{
    .....
    // <init value> should be external clock / (100 * 100)
    CSR_WRITE(TICR0, <init value>);
    .....
}
```

```
linux-2.6.x/include/asm-armnommu/arch-<IC NAME>/<IC NAME>.h

// for example, 16 * MHz for 16 MHz external crystal
#define fMCLK_MHz    ( <external clock/ 1M> * MHz)
```

4.2. UART

UART driver also needs to be modified. Below is an example.

```
Linux-2.6.x/driver/serial/serial_<IC NAME>.c

static void __<IC NAME>_init(const struct uart_port *port, int baud)
{
    .....
    divisor = <external clock frequency> / (baud * 16) - 2;
    .....
}
```



```
}  
  
static struct uart_port __<IC NAME>_ports[UART_NR] = {  
    .....  
    uartclk:    <external clock frequency>  
    .....  
}
```

5. Revision History

Version	Date	Description
V1.0	Sept. 2008	• Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur. Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*
