32-bit microcontroller

HC32L110 / HC32F003 / HC32F005

Internal RC clock for series

Suitable

series	Product number
HC32L110	HC32L110C6UA HC32L110C6PA HC32L110C4UA HC32L110C4PA HC32L110B6PA HC32L110B4PA
HC32F003	HC32F003C4UA HC32F003C4PA
HC32F005	HC32F005C6UA HC32F005C6PA HC32F005D6UA

content

1 Summary

This application note mainly introduces the internal RC clock of HC32L110 / HC32F003 / HC32F005 series.

This application note mainly includes:

ÿ Introduction of internal RC clock

ÿ Internal RC clock on

ÿ Internal RC clock port output

ÿ LPUART clock source 38.4khz

ÿ Internal high-speed RC 22.12Mhz clock in UART/LPUART application

ÿ Internal RC clock calibration

Notice:

- This application note is an application supplement for the HC32L110 / HC32F003 / HC32F005 series and is not intended to replace the user manual.

  Please refer to the user manual for specific functions and register operations.

2 Function introduction

The internal RC clock of HC32L110 / HC32F003 / HC32F005 series can be used for system clock, peripheral module clock source. To ensure the accuracy of the internal RC clock source, the internal clock has a calibration function.

3 Internal RC clock module

3.1 Introduction of Internal RC Clock

3.1.1 Introduction of Internal High Speed RC

The internal high-speed clock has 4M (default system clock), 8M, 16M, 22M, 24M frequency values. Users need different frequencies

When the value is used as the system clock or peripheral clock source, you need to set the register RCH_CR->TRIM bit first.

Before the chip leaves the factory, the calibration value of each frequency value will be stored in the corresponding address of FLASH.

Get the calibration value from the corresponding address and write it into the RCH_CR->TRIM register.

The storage address of each frequency calibration value of RCH:

ÿ 24M ÿ0x00100C00

ÿ 22.12M ÿ0x00100C02

ÿ 16M ÿ0x00100C04

8M ÿ0x00100C06

ÿ 4M ÿ0x00100C08

3.1.2 Introduction of Internal Low Speed RC

The internal low-speed clock has two frequency values of 38.4khz and 32.8khz. When the user needs a different frequency value as the system clock or peripheral

When the clock source is used, the register RCL_CR->TRIM bit needs to be set first.

Before the chip leaves the factory, the calibration value of each frequency value will be stored in the corresponding address of FLASH, the user only needs to select the required RCL clock,

Get the calibration value from the corresponding address and write it into the RCL_CR->TRIM register.

The storage address of each frequency calibration value of RCL:

ÿ 38.4Khzÿ0x00100C20

ÿ 32.8Khzÿ0x00100C22

3.2 Internal RC clock on

Before configuring SYSCTRL0 related registers, you need to configure the startup sequence: SYSCTRL2=0x5A5A, then write SYSCTRL2=0xA5A5 is enough.

Configure SYSCTRL0->RCH_EN=1 to enable internal high-speed clock; configure SYSCTRL0->RCL_EN=1 to enable internal low clock speed clock.

3.3 Internal RC clock port output

Enable the corresponding internal RC clock as described in Sections 3.1 and 3.2.

After checking that the internal clock is stable, configure RC as the system clock:

SYSCTRL0->Clk_sw4_sel=0x00 (internal high speed) or 0x02 (internal low speed)

Configuration port: P24_SEL->P24_sel=0x03, AHB bus clock output signal

GPIO_CTRL1->hclk_sel, frequency division configuration is required for high-speed clock

GPIO_CTRL1->hclk_en=1, hclk output enable

The user can observe the waveform frequency of the P24 port with an oscilloscope to confirm the accuracy of the internal RC clock.

3.4 LPUART clock source 38.4khz

According to chapters 3.1 and 3.2, enable the internal RCL low-speed 38.4Khz clock, in deep sleep mode, the high-speed clock Disabled. If the user needs to receive data correctly in this mode, the internal low speed needs to be enabled as the clock source of the LPUART.

The LPUART can correctly receive data and wake up in deep sleep mode.

LPUART module clock source configuration:

SCON->SCLKSEL=0x03, select the internal RCL as the baud rate clock source.

SCON->PRS=0x07, the configuration frequency division coefficient is 1.

According to the baud rate calculation formula in deep sleep mode:

$$\text{BaudRate} = \frac{\text{Fsclk}}{\text{PreScale}*4}$$

BaudRate = 1*4 $\frac{38400}{}$ =9600bps baud rate. Users can get 4800bps, 2400bps and other constants by configuring the frequency division coefficient.

Use baud rate.

Notice:

- **The LPUART** module only supports **the HC32L110** series.

3.5 Application of Internal High Speed RC 22.12Mhz Clock in UART/LPUART

The HC32L110 / HC32F003 / HC32F005 series has a unique 22.12Mhz high-speed clock source inside the system, which is for customers

There is a requirement for a high-speed communication baud rate of 115200bps.

Calculate the formula according to the baud rate

32 * (65536-TM) $\frac{(SCON.DBAUD + 1) * Freq}{}$ BaudRate =

According to this formula

SCON.DBAUD=0, Freq=22.12Mhz, BaudRate=115200, calculated TM=65530

According to TM=65530, the baud rate is calculated as 115208.333

Bit error rate=(115208.333-115200)/115200=0.72%

If the customer wants to achieve 115200bps, it is recommended to use 22.12M as the baud rate clock source.

Notice:

- **The LPUART** module only supports **the HC32L110** series.

3.6 Internal RC Clock Calibration

This product has a built-in clock calibration circuit, the user needs to write the TRIM value in the corresponding address of FLASH into the clock calibration when leaving the factory

register, refer to Section 3.1.

If the FLASH TRIM value is lost, it can also be calibrated using the built-in clock calibration circuit.

4 Reference samples and drivers

Through the above introduction, together with the user manual of HC32L110 / HC32F003 / HC32F005 series, we

The function and operation method of the internal RC clock module of the MCU have been further mastered.

Huada Semiconductor (HDC) officially provides the application sample and driver library of this module at the same time. Users can open the sample by opening the

The project is further intuitively familiar with the application of the module and the driver library, and can also directly refer to the sample and use in the actual development

Driver library to quickly implement the operation of this module.

ÿ Example reference: ~/HC32L110_DDL/example/inner_rc

~/HC32F003_DDL/example/inner_rc

~/HC32F005_DDL/example/inner_rc

ÿ Driver library reference: ~/HC32L110_DDL/driver/…/clk

~/HC32F003_DDL/driver/…/clk

~/HC32F005_DDL/driver/…/clk

5 Summary

The above chapters briefly introduce the basic functions of the internal RC clock of the HC32L110 / HC32F003 / HC32F005 series.

In the actual application development process, if you need to have a deeper understanding of the use method and operation matters of this module, you should use the corresponding

The user manual shall prevail. The samples and driver libraries mentioned in this article can be used as further experiments and learning by users, or

Direct application in actual development.

6 Other information

Technical support information: www.hdsc.com.cn

7 Version Information & Contact Information

date	Version revision record	
2018/6/1	The first version of Rev1.0 is released.	

If you have any comments or suggestions in the process of purchasing and using, please feel free to contact us.

Email:ymcu@hdsc.com.cn

Website: www.hdsc.com.cn

Mailing address: No. 39, Lane 572, Bibo Road, Zhangjiang Hi-Tech Park, Shanghai

Postcode: 201203