# **32** -bit microcontroller

# **RTC** modules of the **HC32Lxxx** series

## Suitable

| series | Product number |
|---|---|
| **HC32Lxxx** | -- |

# content

# 1 Summary

This application note mainly introduces the RTC modules of the HC32Lxxx series.

This application note mainly includes:

ÿ RTC Calendar

ÿ RTC cycle timing function

ÿ RTC alarm clock

ÿ RTC calibration function

Notice:

- This application note is a supplementary material for the HC32Lxxx series and cannot replace the user manual, specific functions and registers Please refer to the user manual for related matters such as operation.

# 2 Function introduction

The HC32Lxxx series RTC module is an independent BCD timer/counter. RTC can realize calendar, alarm clock, cycle Timing, calibration and other functions.

# 3 RTC standard initialization process

In practical applications, some applications need to be able to continue timing after an external RESET when using RTC.

No reinitialization is required.

The RTC initialization process comprehensively considers various application requirements, which not only meets the conventional RTC initialization function, but also meets the

The function that can continue to keep the timing even when a non-power-on reset occurs, as shown in the following figure:

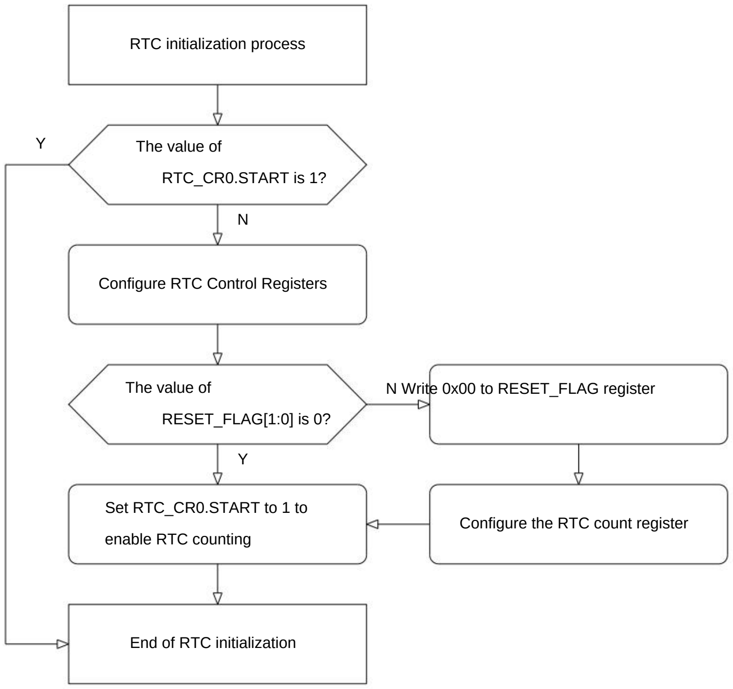
RTC initialization process
The value of RTC_CR0.START is 1?
Y
N
Configure RTC Control Registers
The value of RESET_FLAG[1:0] is 0?
N
Write 0x00 to RESET_FLAG register
Y
Set RTC_CR0.START to 1 to enable RTC counting
Configure the RTC count register
End of RTC initialization

Figure 1 RTC standard initialization flow chart

# 4 RTC Calendar

## 4.1 Clock Source

ÿ Off-chip low-speed crystal oscillator

ÿ On-chip low-speed oscillator (32.768Khz)

ÿ Off-chip high-speed crystal oscillator

## 4.2 Calendar function

ÿ Year, month, week, day, hour, minute, second (BCD format)

ÿ Automatic adjustment of days in month and leap year

ÿ Time system 12/24h can be set

## 4.3 Initialize and read and write the calendar

The RTC module is reset only once when powered on, and the module cannot be reset by various external resets.

ÿ Initialize

Stop counting, set time system, period and other configurations, start counting start=1

ÿ Read calendar

Method 1: CR1.WAIT=1, read calendar data while waiting for CR1.WAITF=1, then CR1.WAIT=0, wait

When CR1.WAITF=0, the reading is completed;

Method 2: Directly read the calendar register, read the second register, read the second register again, if the two are the same, the reading is completed

operate;

Method 3: Read the calendar register during periodic interrupt.

ÿ Write a calendar

When the RTC does not start counting, the calendar register can be written directly.

After start start, CR1.WAIT=1, write calendar data when waiting for CR1.WAITF=1, then CR1.WAIT=0,

(Complete within 1s) Wait for CR1.WAITF=0, restart counting.

# 5-cycle timing function

HC32Lxxx has two types of periodic interrupts:

ÿ PRDSEL = 1, the step is 0.5s periodic interrupt, PRDX is used to configure the periodic interval;

ÿ PRDSEL = 0, PRDS configuration period interval is 0.5s, 1s, 1min, 1h, 1d, 1month or 0, no period is generated interrupt.

# 6 Alarm clock

HC32Lxxx series MCU provides week, hour and minute alarm clock registers, which will be generated when the calendar time is equal to the three at the same time. The alarm clock is interrupted.

Notice:

- In the description of the weekly alarm clock register, b0:b6 correspond to Sunday and Saturday respectively, and the corresponding position is 1, which represents the alarm clock on this day of the week efficient.

# 7 Calibration function

## 7.1 Low speed compensation

The external crystal oscillator clock source is 32.768khz, and the average compensation accuracy per $\frac{1}{\text{second is 32768}}$ *106=30.5ppm, can not meet the high precision requirements,

Therefore, it is necessary to adjust the algorithm to expand the maximum compensation period by 32 times, so that the average compensation unit per second becomes

30.5/32=0.96ppm, but every 1hz accuracy is inaccurate. Because the expansion is 32 times, the compensation register introduces 5 bits

decimal.

Actual frequency - target frequency
compensation target value = *106 target frequency

Compensation target value [PPM]*215
Compensation register CR[8:0]= ( ) Take 2 's complement +0001.00000B 106

This compensation method cannot meet the requirement of precision per second.

## 7.2 High-speed compensation

Based on high-speed Pclk clock compensation (4M~32M), the compensation precision can be dispersed to every second. Guaranteed 0.96ppm per second compensation unit.

Regarding high-speed compensation, the relevant register configuration should be noted:

ÿ The SYSCRTL1.RTC_FREQ_ADJUST bit should be selected according to the actual pclk clock.

ÿ RTC_CR0.1HZSEL=1, high precision 1hz output, RTC_CR0.1HZOE enables 1hz output.

## 7.3 Compensation formula C language code

Since the default compensation register is 0x20, please write 0 first, and then call this formula to calculate.

Compensation range -275.5ppm~+212.9ppm.

```
// Get the compensation value through the compensation formula

RTC_Value = RTC_PPM*32768/1000000;

RTC_COM = Change_FloattoBin(RTC_Value);

//Write the compensation value to the compensation register

M0P_RTC->COMPEN_f.CR = RTC_COM;

M0P_RTC->COMPEN_f.EN = 1;
```

```
uint16_t Change_FloattoBin(float data)//Take 2's complement part

{

 float data1 = data;

 uint16_t temp_data=0;

 uint16_t temp_data1=0;

 uint8_t i, data_inter;

 if(data<0)

 {

      data=-data;

 }

 data_inter = (uint8_t)data;

 data-=data_inter;

 for(i=0;i<6;i++)

 {

  data = data*2;

  if(data>=1)

  {

   data-=1;

   temp_data|=1<<(5-i);

  }

 }

 temp_data|=(data_inter<<6);

 if(data1<0)

 {

  temp_data=~temp_data+1;

 }

 else

 {
    temp_data1=temp_data;

  }

    temp_data1=temp_data;

   temp_data1>>=1;

    temp_data1+=0x20;

    temp_data1&=0x1ff;

    return temp_data1;//Take the lower 9 bits of compensation value

}
```

# 8 Enter low power mode

Since the module register configuration uses the APB bus clock PCLK, the internal operation of the RTC module uses RCL or

There are 2~3 low-speed clocks such as XTL and other low-speed clocks from the completion of the module register configuration and enablement to the actual start of operation inside the module.

The delay of the clock, so you must pay attention in use. You cannot enter the low-power mode immediately after configuring the register. It is recommended to delay.

More than 3 low-speed clocks enter the low-power mode, otherwise the module may not start at all.

# 9 Reference samples and drivers

Through the above introduction and in conjunction with the user manual of the HC32Lxxx series, we discuss the functions and functions of the RTC module of the above series of MCUs.

The operation method has 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 Sample reference: ~/HC32Lxxx_DDL/example/rtc

ÿ Driver library reference: ~/HC32Lxxx_DDL/driver/…/rtc

# 10 Summary

The above chapters briefly introduced the RTC module of the HC32Lxxx series, and explained the functions and operation steps of this module in detail.

step. In the actual application development process, if the user needs to have a deeper understanding of the usage and operation of this module,

The corresponding user manual shall prevail. The samples and driver libraries mentioned in this chapter can be used as user's further experiments and learning.

It can also be directly applied in actual development.

# 11 Other information

Technical support information: www.hdsc.com.cn

# 12 Version Information & Contact

| date | Version revision record |
|---|---|
| 2018/5/31 | Rev1.0 initial release |
| 2020/1/10 | Rev1.1 1. Add RTC standard initialization process; 2. Support all HC32Lxxx series of products. |
| 2020/8/27 | Rev1.2 adds Chapter 8 |
| 2020/9/24 | Rev1.3 Added section 7.3 RTC compensation formula C language code |

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

Email: mcu@hdsc.com.cn

Website: http://www.hdsc.com.cn/mcu.htm

Mailing address: 10th Floor, Block A, No. 1867, Zhongke Road, Zhangjiang Town, Pudong New Area, Shanghai

Postcode: 201203