RTC
Application Guide

Issue 09

Date 2019-05-15

HiSilicon (Shanghai) Technologies Co., Ltd.

Address: New R&D Center, 49 Wuhe Road, Bantian,
Longgang District,
Shenzhen 518129 P. R. China

Website: http://www.hisilicon.com/en/

Email: support@hisilicon.com

About This Document

Purpose

This document describes the RTC correction scheme, ensuring that the RTC counts correctly.

NOTE

This document uses the Hi3536 as an example. Unless otherwise stated, Hi3521A/20D V300, Hi3531A, Hi3518E V20X/16C V200, Hi3519 V100, Hi3519 V101, Hi3559 V100, Hi3556 V100, Hi3531D V100, Hi3521D V100, Hi3536C V100, Hi3536D V100, Hi3520DV400, Hi3516C V300, Hi3559A V100, Hi3559C V100, Hi3519A V100, Hi3556A V100, Hi3516C V500, Hi3516D V300, Hi3559 V200, Hi3556 V200, Hi3516E V200, Hi3516E V300, Hi3518E V300, Hi3516DV200, and Hi3536 contents are consistent.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3536	V100
Hi3521A	V100
Hi3520D	V300
Hi3531A	V100
Hi3518E	V200
Hi3518E	V201
Hi3516C	V200
Hi3519	V100
Hi3516C	V300
Hi3519	V101
Hi3559	V100
Hi3556	V100
Hi3531D	V100
Hi3521D	V100

Product Name	Version
Hi3536C	V100
Hi3536D	V100
Hi3520D	V400
Hi3559A	V100
Hi3559C	V100
Hi3519A	V100
Hi3556A	V100
Hi3516C	V500
Hi3516D	V300
Hi3516A	V300
Hi3559	V200
Hi3556	V200
Hi3516E	V200
Hi3516E	V300
Hi3518E	V300
Hi3516D	V200

Intended Audience

This document is intended for technical support personnel.

Change History

Changes between document issues are cumulative. The latest document issue contains all changes made in previous issues.

Issue 09 (2019-05-15)

This issue is the ninth official release, which incorporates the following changes:

Sections 2.2, 2.3 and 5.1 are updated.

Chapters 3 and 4 are deleted.

The method of adjusting the RTC drive capability is added.

Issue 08 (2018-11-30)

This issue is the eighth official release, which incorporates the following changes:

The contents related to the Hi3516E V300, Hi3516E V200, and Hi3518E V300 are added.

Issue 07 (2018-09-26)

This issue is the seventh official release, which incorporates the following changes:

The contents related to the Hi3559 V200 and Hi3556 V200 are added.

Issue 06 (2018-05-15)

This issue is the sixth official release, which incorporates the following changes:

The contents related to the Hi3559A V100 and Hi3559C V100 are added.

Issue 05 (2017-10-18)

This issue is the fifth official release, which incorporates the following changes:

Note is added to chapter 5.

Chapter 6 is added.

Issue 04 (2017-09-08)

This issue is the fourth official release, which incorporates the following changes:

The description of the Hi3536D V100 is added.

Section 5.1 and section 5.2 are modified.

Issue 03 (2017-07-14)

This issue is the third official release, which incorporates the following changes:

Section 1.3 is deleted.

Section 2.1 and 2.2 are updated.

Chapter 6 is added.

Issue 02 (2017-04-10)

This issue is the second official release, which incorporates the following changes:

The descriptions of the Hi3536C V100 and Hi3516A V200 are added.

Issue 01 (2016-12-01)

This issue is the first official release.

Issue 00B02 (2015-11-05)

This issue is the second draft release.

Issue 00B01 (2015-1-19)

This issue is the first draft release.

Contents

Figures

Tables

1 Introduction

1.1 RTC Classification

Typically, real-time clocks (RTCs) are classified into three types:

- Non-integrated RTC: This RTC has only the RTC timing circuit but no integrated crystal oscillator and temperature compensation circuit. The timing accuracy of this RTC depends on the accuracy of the external crystal oscillator and is susceptible to temperature change. Typically, the timing accuracy is high at ambient temperature, and the timing deviation gradually increases when the temperature increases or decreases.
- RTC with integrated crystal oscillator: This RTC has integrated RTC timing circuit and crystal oscillator but no temperature compensation circuit. The timing accuracy of this RTC reaches the highest at ambient temperature but is also affected by temperature change.
- Integrated RTC: This RTC has integrated RTC timing circuit, crystal oscillator, and temperature compensation circuit (including the temperature sensor). It needs to be calibrated before delivery. As this RTC has the temperature compensation circuit, it features high timing accuracy and is slightly affected by temperature change.

1.2 RTC Working Mode

The embedded RTC of the Hi3536 supports the fixed frequency-division mode.

Similar to the non-integrated RTC, the embedded RTC of the Hi3536 uses the frequency-division clock generated by the external crystal oscillator and oscillation circuit. The frequency divider is fixed during working. In this mode, the RTC timing accuracy depends on the frequency accuracy of the external crystal oscillator and is affected by ambient temperature change. You can replace the non-integrated RTC with the embedded RTC of the Hi3536 to reduce costs.

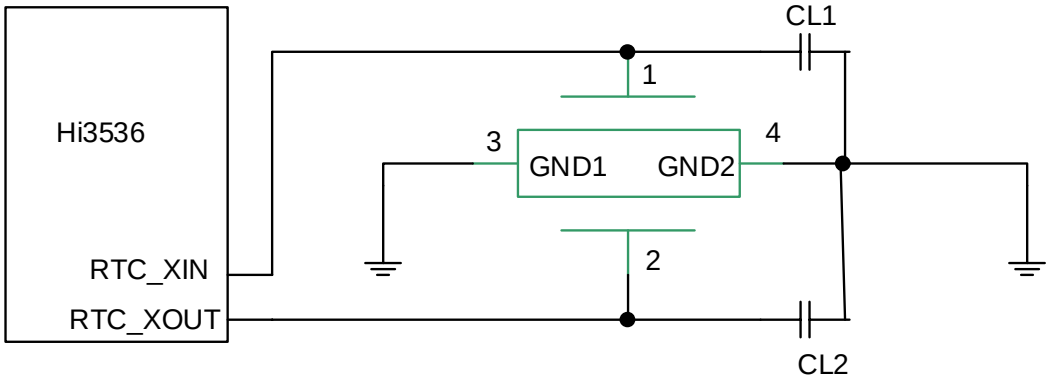
The Hi3536 RTC does not have internal temperature compensation circuit, and works only in fixed frequency-division mode. If the frequency offset of the RTC is too large, you can adjust the RTC frequency divider to fine-tune the RTC frequency. If high timing accuracy is required, you are advised to select the external RTC with an embedded crystal oscillator or the RTC with the temperature compensation function.

2 Hardware Reference Circuit of the Crystal

2.1 Hardware Reference Circuit

Crystals and capacitors need to be selected for the hardware reference circuit, as shown in Figure 2-1.

Figure 2-1 Hardware reference circuit of the crystal



2.2 Selecting Crystal Oscillators

The following specifications must be taken into account when you select crystal oscillators:

- Standard load capacitance (CL): Crystal oscillators impose strict requirements on the load capacitance. The crystal oscillator frequency reaches the nominal value only when the actual load capacitance is the same as the load capacitance defined in crystal oscillator specifications.
- Series resistance, Rs, or crystal resonance equivalent series resistance (ESR): Greater ESR indicates that the crystal is more difficult to drive. The typical and maximum values of Rs are specified in crystal specifications.

  The crystal oscillator whose maximum series resistance is less than 70 kilohms is recommended for the crystal oscillation circuit of the chip.

- Maximum drive level (DL): Indicates the maximum crystal oscillation amplitude. If the oscillation amplitude exceeds a specified amplitude, the crystal oscillator is easy to be damaged.

Table 2-1 shows the constraints on crystal selection.

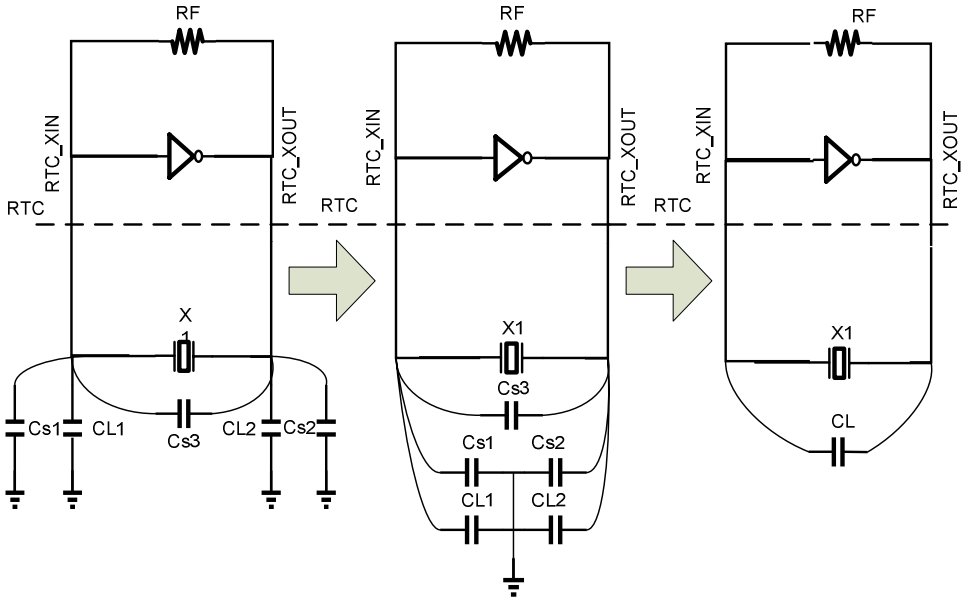
Table 2-1 Constraints on crystal selection

Parameter	Symbol	Specifications			
		MIN	TYP	MAX	Unit
Series Resistance	ESR	-	-	70	kΩ
Load Capacitance	CL	-	12.5	-	pF
Shunt Capacitance	C0	1	-	2	pF
Motional Capacitance	C1	2	-	6	fF

2.3 Selecting Capacitors

Figure 2-2 shows the actual CL diagram.

Figure 2-2 Actual CL diagram



CL_SPEC=CL1//CL2+Cs1//Cs2+Cs3

In the Pierce oscillator, CL1 and CL2 are generally set to the same value.

Specifically:

- CL_SPEC is the standard load capacitor specified in the crystal specifications.
- CL1 and CL2 are matched capacitors of the crystal.
- Cs1, Cs2, and Cs3 are parasitic capacitors. Generally, they are set to 3–5 pF.

Take a 12.5 pF crystal as an example. The capacitance of CL1 and CL2 is calculated as follows: (12.5 pF–3 pF) x 2 = 19 pF

The parasitic capacitance varies with the PCB design and the selection of the crystal and MCU. Therefore, after the PCB design, you can select the CL1 with different capacitance and test the frequency offsets to obtain the output frequency closest to 32.768 kilohms. For details about the test method, see chapter 5 "Test Methods for Crystal Specifications."

3 HI_RTC Driver Usage

NOTE

The Hi3536D V100, Hi3559A V100, Hi3559C V100, Hi3519A V100, Hi3556A V100, Hi3516C V500, Hi3516D V300, Hi3559 V200, Hi3516E V200, Hi3516E V300, Hi3518E V300, Hi3516DV200, and Hi3556 V200 do not support the hi_rtc driver.

3.1 Compilation

Run the following commands in the RTC directory to generate the hi35xx_rtc.ko driver, hi_rtc.ko driver, and sample program test:

```
cd rtc
make
cd test
make
```

The hi35xx_rtc.ko and hi_rtc.ko drivers respectively comprise the following:

- hi35xx_rtc.ko: Hi3518EV20X/ Hi3516CV200, Hi3519 V100, Hi3519 V101, Hi3559 V100, Hi3556 V100, and Hi3516C V300
- hi_rtc.ko: Hi3521A/ Hi3520DV300, Hi3531A, Hi3536, Hi3521D V100, Hi3520DV400, Hi3531D V100, and Hi3536C V100

3.2 Usage

Run the following command to insert the hi35xx_rtc.ko driver module:

```
insmod hi35xx_rtc.ko
```

Run the following command to insert the hi_rtc.ko driver module:

```
insmod hi_rtc.ko
```

RTC driver functions are described in the test sample program running on the board, as shown in Figure 3-1.

Figure 3-1 Test sample program usage

```
Usage: ./test [options] [parameter1] ...
Options:
        -s(set)              Set time/alarm,      e.g '-s time 2012/7/15/13/37/59'
        -g(get)              Get time/alarm,      e.g '-g alarm'
        -w(write)            Write RTC register,  e.g '-w <reg> <val>'
        -r(ead)              Read RTC register,   e.g '-r <reg>'
        -a(alarm)            Alarm ON/OFF',       e.g '-a ON'
        -reset               RTC reset
        -b(battery monitor)  battery ON/OFF,      e.g '-b ON'
        -f(requency)         frequency precise adjustment, e.g '-f <val>'
```

Configuring and Obtaining the RTC Time

Run the following command to configure the RTC time:

```
./test -s time <year/month/day/hour/minute/second>
```

Run the following command to obtain the RTC time:

```
./test -g time
```

Configuring and Obtaining the RTC Alarm Time

Run the following command to configure the RTC alarm time:

```
./test -s alarm <year/month/day/hour/minute/second>
```

Run the following command to obtain the RTC alarm time:

```
./test -g alarm
```

Run the following command to set whether an interrupt is generated when the alarm time reaches. The driver interrupt routine is added by users.

```
./test -a ON/OFF
```

Reading and Configuring Registers in the RTC

Run the following command to read a register in the RTC. This function is used for auxiliary tests.

```
./test -r <reg>
```

Run the following command to configure the register in the RTC. This function is used for auxiliary tests.

```
./test -w <reg> <value>
```

For details about the **reg** value, see the real-time clock contents in the user guide for each chip.

Resetting the RTC

Run the following command to reset the RTC:

```
./test -reset
```

Fine-Tuning the Frequency Divider in Fixed Frequency-Division Mode

Run the following command to set the frequency divider for adjusting the clock forward or backward:

```
./test -f <val>
```

<val> is 10000 times the frequency divider to be set. For example, if the frequency divider is 327.60, **val** is **3276000**. Run **./test –f** to view the current frequency divider. Due to calculation errors, there may be little deviation between the obtained frequency divider and the configured frequency divider. The frequency divider ranges from 327.60 to 327.70.

Enabling and Disabling Battery Level Monitoring

Run the following command to enable the RTC battery level monitoring function:

```
./test -b ON
```

Run the following command to disable the RTC battery level monitoring function:

./test –b OFF

NOTICE

This feature applies only to Hi3536, Hi3519 V100, Hi3519 V101, Hi3516A V200, Hi3516C V300, Hi3559 V100/Hi3556 V100, Hi3521D V100, Hi3531D V100, Hi3520DV400, and Hi3536C V100.

Adjusting the RTC Drive Capability

There are four levels of drive capabilities. To adjust the drive capability, you can set the lower 2 bits of the SPI_RW register. The address of the register varies with the SoC. For details, see section 3.9 "RTC" in the *Hi35xx Vxx SoC Data Sheet*. The following uses Hi3516E V200 as an example. The base address of the register is 0x120E0004 and the offset address is 0x1B.

Step 1 Run the following commands to unlock the default value (valid only for Hi3559A V100/Hi3559C V100/Hi3519A V100/Hi3556A V100 /Hi3516E V200/Hi3516E V300/Hi3518E V300/Hi3516D V200):

```
himm 0x120E0004 0x016400CD
himm 0x120E0004 0x016500AB
himm 0x120E0004 0x0166005A
himm 0x120E0004 0x0167005A
```

Step 2 Run out of the following four commands as required to set the level:

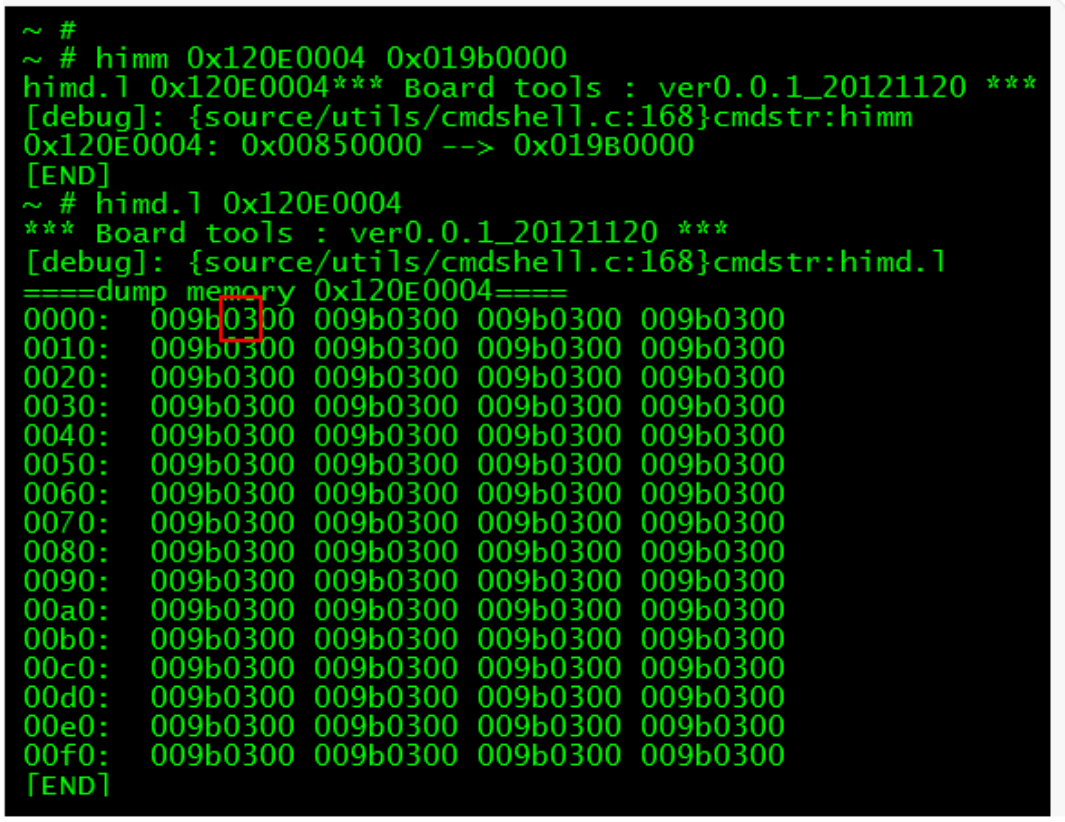
```
himm 0x120E0004 0x011b0000 (level 0)
```



```
himm 0x120E0004 0x011b0001 (level 1)
himm 0x120E0004 0x011b0002 (level 2)
himm 0x120E0004 0x011b0003 (level 3)
```

Step 3 Read the register value to check whether the configuration is successful. The readback value in the red box in the following figure indicates the configured level.

```
himm 0x120E0004 0x019b0000
himd.l 0x120E0004
```



----End

Modifying the RTC Drive Capability Code

In the **hi_rtc.c** file, configure the register marked in the following figure. Set the register to a level with the drive capability adjusted. 0x0–0x3 correspond to levels 0–3, respectively.


```
static int __init rtc_init(void)
{
    int ret = 0;

    ret = misc_register(&rtc_char_driver);
    if (0 != ret)
    {
        HI_ERR("rtc device register failed!\n");
        return -EFAULT;
    }

    ret = request_irq(RTC_IRQ, &rtc_interrupt, 0,
            "RTC Alarm", NULL);
    if(0 != ret)
    {
        HI_ERR("hi35xx rtc: failed to register IRQ(%d)\n", RTC_IRQ);
        goto RTC_INIT_FAIL1;
    }

#ifdef HI_FPGA
    /* config SPI CLK */
    writel(0x1, (volatile void *)SPI_CLK_DIV);
#else
    /* clk div value = (apb_clk/spi_clk)/2-1, for asic ,
          apb clk = 62.5MHz, spi_clk = 15.625MHz,so value= 0x1 */
    writel(0x1, (volatile void *)SPI_CLK_DIV);
#endif

    /* enable total interrupt. */
    spi_rtc_write(RTC_IMSC, 0x4);
    spi_rtc_write(RTC_CLK_CFG, 0x01);

    return 0;

RTC_INIT_FAIL1:
    misc_deregister(&rtc_char_driver);
    return ret;
} « end rtc_init »
```

User Interface

See the **hi_rtc.h** file.

4 Usage of Standard RTC Kernel Driver Under Linux

4.1 Compilation

The Hi3536C V100, Hi3536D V100, Hi3559A V100, Hi3519A V100, Hi3556A V100, Hi3516C V500, Hi3516D V300, Hi3559C V100, Hi3559 V200, Hi3516E V200, Hi3516E V300, Hi3518E V300, Hi3516D V200, and Hi3556 V200 support the RTC kernel driver. The RTC option in the kernel compilation is enabled by default.

4.2 Usage

Burn the kernel to the board, boot the board, and execute the following command:

```
ls /dev/rtc0
```

If the RTC devices can be viewed, the RTC kernel driver is properly loaded.

The RTC kernel driver supports calling the ioctl function from the system.

Table 4-1 ioctl instructions and functions supported by the RTC kernel driver

Instruction	Function
RTC_ALM_READ	Reads the alarm time.
RTC_ALM_SET	Reads the time and date.
RTC_SET_TIME	Sets the time and date.
RTC_PIE_ON	Enables the RTC global interrupt.
RTC_PIE_OFF	Disables the RTC global interrupt.
RTC_AIE_ON	Enables the RTC alarm interrupt.
RTC_AIE_OFF	Disables the RTC alarm interrupt.
RTC_UIE_ON	Enables the RTC update interrupt.
RTC_UIE_OFF	Disables the RTC update interrupt.

Instruction	Function
RTC_IRQP_SET	Sets the interrupt frequency.

Configuring and Obtaining the RTC Time

- Run the following command to configure the RTC time:

```
ioctl(fd, RTC_SET_TIME, &rtc_tm);
```

- Run the following command to obtain the RTC time:

```
ioctl(fd, RTC_RD_TIME, &rtc_tm);
```

Adjusting the RTC Drive Capability

There are four levels of drive capabilities. To adjust the drive capability, you can set the lower 2 bits of the SPI_RW register. The address of the register varies with the SoC. For details, see section 3.9 "RTC" in the *Hi35xx Vxx SoC Data Sheet*. The following uses Hi3516E V200 as an example. The base address of the register is 0x120E0004 and the offset address is 0x1B.

Step 1 Run the following commands to unlock the default value (valid only for Hi3559A V100/Hi3559C V100/Hi3519A V100/Hi3556A V100 /Hi3516E V200/Hi3516E V300/Hi3518E V300/Hi3516D V200):

```
himm 0x120E0004 0x016400CD
himm 0x120E0004 0x016500AB
himm 0x120E0004 0x0166005A
himm 0x120E0004 0x0167005A
```

Step 2 Run out of the following four commands as required to set the level:

```
himm 0x120E0004 0x011b0000 (level 0)
himm 0x120E0004 0x011b0001 (level 1)
himm 0x120E0004 0x011b0002 (level 2)
himm 0x120E0004 0x011b0003 (level 3)
```

Step 3 Read the register value to check whether the configuration is successful. The readback value in the red box in the following figure indicates the configured level.

```
himm 0x120E0004 0x019b0000
himd.l 0x120E0004
```



```
~ #
~ # himm 0x120E0004 0x019b0000
himd.l 0x120E0004*** Board tools : ver0.0.1_20121120 ***
[debug]: {source/utils/cmdshell.c:168}cmdstr:himm
0x120E0004: 0x00850000 --> 0x019B0000
[END]
~ # himd.l 0x120E0004
*** Board tools : ver0.0.1_20121120 ***
[debug]: {source/utils/cmdshell.c:168}cmdstr:himd.l
====dump memory 0x120E0004====
0000:   009b0300 009b0300 009b0300 009b0300
0010:   009b0300 009b0300 009b0300 009b0300
0020:   009b0300 009b0300 009b0300 009b0300
0030:   009b0300 009b0300 009b0300 009b0300
0040:   009b0300 009b0300 009b0300 009b0300
0050:   009b0300 009b0300 009b0300 009b0300
0060:   009b0300 009b0300 009b0300 009b0300
0070:   009b0300 009b0300 009b0300 009b0300
0080:   009b0300 009b0300 009b0300 009b0300
0090:   009b0300 009b0300 009b0300 009b0300
00a0:   009b0300 009b0300 009b0300 009b0300
00b0:   009b0300 009b0300 009b0300 009b0300
00c0:   009b0300 009b0300 009b0300 009b0300
00d0:   009b0300 009b0300 009b0300 009b0300
00e0:   009b0300 009b0300 009b0300 009b0300
00f0:   009b0300 009b0300 009b0300 009b0300
[END]
```

----End

Modifying the RTC Drive Capability Code

In the **\drivers\rtc\rtc-hibvt.c** file, configure the register marked in the following figure. Set the register to a level with the drive capability adjusted. 0x0–0x3 correspond to levels 0–3, respectively.


```
static int hibvt_rtc_init(struct hibvt_rtc *rtc)
{
    void *spi_reg = rtc->regs;
    int ret = 0;
    unsigned char val = 0;
    /*
     * clk div value = (apb_clk/spi_clk)/2-1,
     *  apb clk = 100MHz, spi_clk = 10MHz,so value= 0x4
     */
    writel(CLK_DIV_DEFAULT, (spi_reg+SPI_CLK_DIV));

    ret |= hibvt_spi_rtc_write(spi_reg, RTC_IMSC, INT_MSK_DEFAULT);
    ret |= hibvt_spi_rtc_write(spi_reg, RTC_SAR_CTRL, LV_CTL_DEFAULT);


    ret |= hibvt_spi_rtc_write(spi_reg, RTC_CLK_CFG, 0x01);

    /* default FREQ COEF */
    ret |= hibvt_spi_rtc_write(spi_reg, RTC_FREQ_H, FREQ_H_DEFAULT);
    ret |= hibvt_spi_rtc_write(spi_reg, RTC_FREQ_L, FREQ_L_DEFAULT);

    ret |= hibvt_spi_rtc_read(spi_reg, RTC_INT_RAW, &val);
    //ret |= hibvt_spi_rtc_read(spi_reg, RTC_CLK_CFG, &val2);
    if (ret) {
        dev_err(&rtc->rtc_dev->dev, "IO err.\n");
        return ret;
    }

    if (val & LV_INT_MASK) {
        dev_err(&rtc->rtc_dev->dev,
            "low voltage detected, date/time is not reliable.\n");
        hibvt_spi_write(rtc->regs, RTC_INT_CLR, 1);
    }

    return ret;
} « end hibvt_rtc_init »
```

User Interface

See the **rtc.txt** file in the **Documentation** directory of the kernel.

5 Test Methods for Crystal Specifications

5.1 Frequency Offset Test

- Connect the frequency meter to a chip clock test pin (TEST_CLK) through a coaxial cable. In the U-Boot, configure the register, make the pin multiplexed as the CLK_TEST function, and then select the RTC signal for output to test the RTC frequency.
- It is highly recommended that you do not use the oscilloscope on the crystal oscillator pin for direct measurement. The capacitance value of a typical passive probe is less than 10 pF and the input impedance is about 10 MΩ. Both values greatly influence the operation mode of the crystal oscillator.
- You can determine the frequency offset range of the crystal oscillator based on the product requirements.

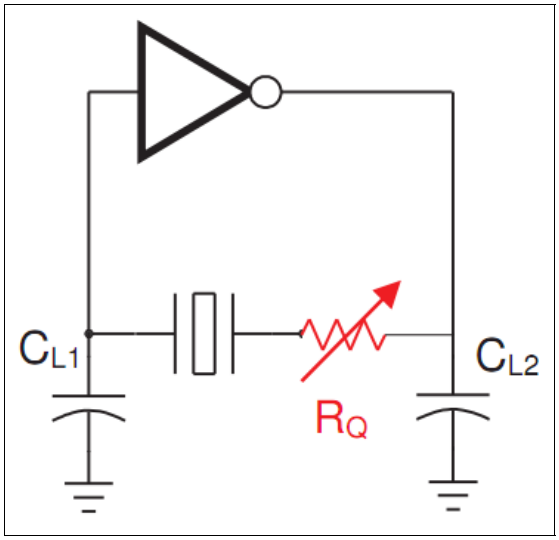
NOTICE

The test_clk pin should be connected to the test points during the board design to facilitate future crystal specification tests.

5.2 Safety Factor Test

As shown in Figure 5-1, add an RQ resistor that is connected with the crystal oscillator in series. Add the RQ value till the crystal oscillator does not work, then the RQ resistor reaches its maximum vaule, RQ_{max} (It is recommended that the RQ_{max} value is verified by the SMD resistor instead of an adjustable potentiometer).

Figure 5-1 Safety factor test diagram



By leveraging the preceding test, you can measure the oscillation margin (OA) and safety factor (SF) of the crystal oscillator circuit.

$OA = RQ_{max} + ESR$

SF:

$$SF = \frac{OA}{ESR} = \frac{RQ_{max} + ESR}{ESR}$$

Table 5-1 Safety factor constraints

Safety Factor	Constraints
SF<2	Insecure
2 ≤ SF < 3	Applicable
3 ≤ SF < 5	Secure
SF ≥ 5	Very secure

Note: It is required that the safety factor is greater than 3 at least.

For example, if the crystal ESR is 60 K and the RQ_{max} of the additional resistor in series is 120 K, it just reaches the secure level.

5.3 DL Test

The oscillation amplitude of the pins RTC_XIN and RTC_XOUT are specified on the Hi3536 crystal oscillation circuit. The actual drive level (less than the maximum drive level in crystal oscillator specifications) when the circuit works is calculated as follows:

$DL\_actual=0.35*Rs\_max*(\pi*f*Vpp_{XOUT}*CL)^2/2$

in which

- **Rs_max** is the maximum series resistance in crystal specifications.
- **f** indicates the crystal resonance frequency.
- **Vpp_{XOUT}** indicates the peak-to-peak voltage for the RTC_XOUT pin measured by using the oscilloscope.
- **CL** is the standard load capacitance in crystal specifications.

NOTE

Due to the parasitic resistance and capacitance effect of the oscilloscope probe during the voltage test, the test result deviation may appear. Therefore, the preceding formula is just a simpler method for DL estimation. You can ask crystal manufacturer for more accurate tests.

5.4 Oscillator Startup Time Test

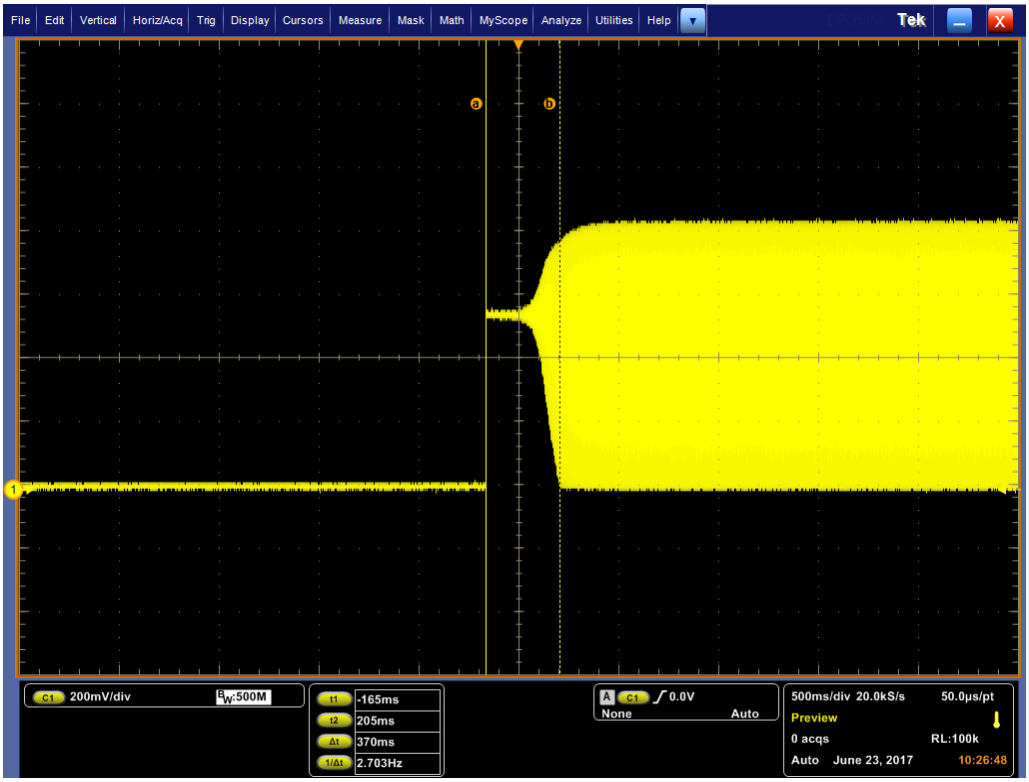
Generally, for RTC oscillation circuits, a startup time between hundreds of milliseconds and several seconds is normal. The startup time of the crystal oscillator is determined by different factors:

- The lower the frequency is, the longer the startup time becomes (compared with 24 MHz system clock).
- The larger the load capacitor is, the longer the startup time becomes.
- The greater the crystal ESR is, the longer the startup time becomes (focused during crystal selection).
- The larger the OA is, the faster the oscillator starts (namely, the larger the SF is, the faster oscillator starts).
- The larger the crystal parasitic inductance is, the longer the oscillator startup time becomes.

In view of above factors, give priority to SF and frequency offset tests before testing the oscillator startup time.

Use an oscilloscope to measure the RTC_Xout waveform, capture the first rising edge of the waveform, and measure the time from the first rising edge to stable frequency outputs, as shown in Figure 5-2.

Figure 5-2 Oscillator satartup time diagram



6 Q&A

6.1 No Oscillation on the Oscillator

[Symptom]

The 32.768 kHz clock has no output, and the value of the second register on the RTC timing circuit is constant.

[Analysis]

Use the oscilloscope probe to check oscillation waveforms on the RTC_XIN pin, and various oscillation waveforms are caused in the following situations:

- If there is no oscillation waveform on the pin, the crystal oscillator may be damaged.
- If about 32 kHz sine waves are detected and the peak-to-peak amplitude is less than 600 mV, capacitance of CL1 and CL2 may be large, causing the oscillation circuit drive capability to be insufficient and the peak-to-peak amplitude is smaller than that at 32.768 kHz frequency. Therefore, oscillation waveforms cannot pass the subsequent Schmitt trigger.
- If about 200 kHz sine waves are detected and the peak-to-peak amplitude is less than 600 mV, capacitance of CL1 and CL2 may be small, causing the oscillation circuit to oscillate to 200 kHz frequency at which the amplitude is smaller than that at 32.768 kHz frequency. Therefore, oscillation waveforms cannot pass the subsequent Schmitt trigger.

[Solution]

- Replace the crystal oscillator if it is damaged.
- If about 32 kHz sine waves are detected and the peak-to-peak amplitude is small, check whether capacitance of CL1 and CL2 is large and replace the capacitors with appropriate capacitors.
- If about 200 kHz sine waves are detected and the peak-to-peak amplitude is small, check whether capacitance of CL1 and CL2 is small and replace the capacitors with appropriate capacitors.

6.2 200 kHz Frequency Output by the Oscillator

[Symptom]

The frequency output by the 32.768 kHz clock is approximate to 200 kHz, and the value of the second register on the RTC timing circuit increases by 6 per second.

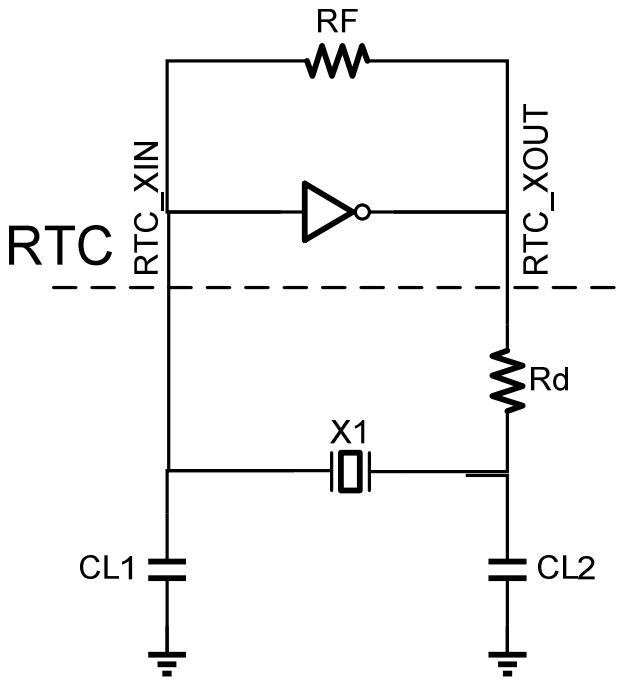
[Analysis]

If exceptions occur on the 32.768 kHz crystal oscillator, the crystal oscillator may oscillate near to six times of the fundamental frequency because the resonance point of 6.1 times of the fundamental frequency is available on this crystal oscillator.

[Solution]

Check whether capacitance of CL1 and CL2 is small. If the capacitances of the two capacitors are appropriate and the oscillation frequency is 200 kHz, add an Rd whose value is 1/(2 π x 32768 x CL2) to the circuit, as shown in Figure 6-1. The Rd and CL2 form an RC filter, reducing the loop gain at 6.1 times of the fundamental frequency.

Figure 6-1 Circuit for resolving the 200 kHz output frequency issue



NOTE

You are not advised to add an Rd to the circuit. Ensure that the signal amplitude on the RTC_XOUT pin is not small before adding an Rd to the circuit.

6.3 Incorrect Oscillation Frequency

[Symptom]

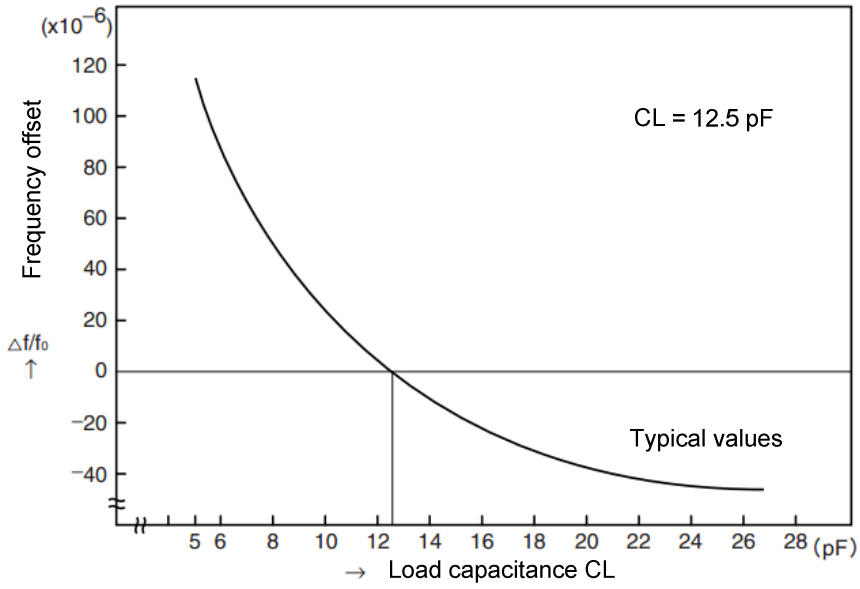
The frequency output by the 32.768 kHz clock is not 32.768 kHz.

[Analysis]

The crystal oscillator and load capacitance work together to determine the oscillation frequency on the oscillation circuit. The crystal oscillator determines the frequency range

(frequencies corresponding to the frequency offset 0 in Figure 6-2, and the actual load capacitance determines the frequency offset (the value of the frequency offset 0 in Figure 6-2).

Figure 6-2 Relationship between the frequency offset and the actual load capacitance CL



[Solution]

Firstly, ensure that crystal oscillator pin bending does not apply stress to the internal crystal oscillator and the soldering temperature complies with specifications in the data sheet. Secondly, check whether capacitance of CL1 and CL2 are appropriate and replace the two capacitors with appropriate ones.

