



CV186AH RTC 使用手册

Version: 1.0

Release date: 2023/12

©2022 北京晶视智能科技有限公司
本文件所含信息归北京晶视智能科技有限公司所有。
未经授权，严禁全部或部分复制或披露该等信息。

目录

1	声明	2
2	RTC 操作指南	3
2.1	模块介绍	3
2.2	计数时钟频率	4
2.3	操作准备	4
2.4	使用方式	4
2.4.1	Ioctl 控制 RTC	4
2.4.2	ioctl 使用范例	5
2.4.3	结构体	5
3	Linux 系统中 RTC 命令	7
3.1	date 和 hwclock	7
4	RTC 问题排查	8
4.1	软件问题	8
4.2	硬件问题	8

修订记录

Revision	Date	Description
1.0	2023/09/15	初稿

1 声明



法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

联系我们

地址 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

电话 +86-10-57590723 +86-10-57590724

邮编 100094（北京）518100（深圳）

官方网站 <https://www.sophgo.com/>

技术论坛 <https://developer.sophgo.com/forum/index.html>

2 RTC 操作指南

2.1 模块介绍

RTC(real time clock) 硬件时钟，为系统提供与维护当前的日期和时间。它的主要作用是提供一个可靠的时间基准，使系统能够实现以下功能：

- 准确记录时间

RTC 提供了一个准确的时间来源，可用于记录事件的时间戳、创建日志、执行时间敏感的任务和调度，以及确保计算机系统的时间数据是可靠的。

- 制定计划和定时任务

RTC 允许系统根据特定的时间计划任务和事件。这对于定期执行特定操作、闹钟、计时器和周期性任务非常有用。

- 持久性的时间存储

RTC 通常具有一个电池备份，允许它在系统关闭或断电时继续运行。这确保了 RTC 数据的持久性，包括日期、时间和闹钟设置。也就是说当处理器处于下电关机或休眠状态时，RTC 仍会继续计数并维护时间信息不丢失。

- 安全性

RTC 可以用于实现时间敏感的安全性功能，例如生成时间戳、验证证书的有效期、处理数字签名等。

- 系统唤醒和上下电操作

RTC 可以配置为在指定的时间唤醒计算机系统，这对于定时唤醒休眠或睡眠模式下的设备非常有用，以执行特定任务。也可以通过操作寄存器使 RTC 以外的 IP 模块进行上下电的操作。

Linux 内核将 RTC 作为时间与日期维护器，当 Linux 系统启动时，内核读取 RTC 时间以初始化系统（软件）时钟达成时间同步。内核在需要时，亦可将时间与日期回写到 RTC 中。RTC 在各种应用中都有用武之地，包括计算机、服务器、嵌入式系统、移动设备、网络设备、工业自动化、医疗设备、汽车等

2.2 计数时钟频率

RTC 的计数时钟采用 32.768KHz 时钟，运行基于一个 32-bit 加法计数器提供秒计数，计数最大时间为：

$$2^{32} \text{ 秒} = 49710 \text{ 天} = 136 \text{ 年}$$

2.3 操作准备

RTC 的操作准备如下：

- 使用 SDK 发布的 kernel
- 插入模块: insmod soph_rtc.ko

2.4 使用方式

2.4.1 Ioctl 控制 RTC

应用层可以通过 ioctl 控制 RTC，设备节点为/dev/rtc0

使用方式如下：

```
int ioctl(int fd, ind cmd);
```

ioctl 指令功能描述

指令	描述
RTC_ALM_READ	读取闹钟时间
RTC_ALM_SET	设置闹钟时间
RTC_RD_TIME	读取时间与日期
RTC_SET_TIME	设置时间与日期
RTC_PIE_ON	开 RTC 全局中断
RTC_PIE_OFF	关 RTC 全局中断
RTC_AIE_ON	使能 RTC 闹钟中断
RTC_AIE_OFF	禁止 RTC 闹钟中断
RTC_UIE_ON	使能 RTC 更新中断
RTC_UIE_OFF	禁止 RTC 更新中断
RTC_IRQP_SET	设置中断频率

2.4.2 ioctl 使用范例

```
static const char default_rtc[] = "/dev/rtc0";
struct rtc_time rtc_tm;
int fd;

fd = open(rtc, O_RDONLY);
if (fd == -1) {
    perror(rtc);
    exit(errno);
}
```

通过如下命令可获取 RTC 时间：

```
/* Read the RTC time/date */
retval = ioctl(fd, RTC_RD_TIME, &rtc_tm);
if (retval == -1) {
    perror("RTC_RD_TIME ioctl");
    exit(errno);
}
fprintf(stderr, "\n\nCurrent RTC date/time is %d-%d-%d, %02d:%02d:%02d.\n",
    rtc_tm.tm_mday, rtc_tm.tm_mon + 1, rtc_tm.tm_year + 1900,
    rtc_tm.tm_hour, rtc_tm.tm_min, rtc_tm.tm_sec);
```

通过如下命令可设置 RTC 时间：

```
retval = ioctl(fd, RTC_SET_TIME, &rtc_tm);
if (retval == -1) {
    perror("RTC_RD_TIME ioctl");
    exit(errno);
}
```

2.4.3 结构体

· rtc_time

```
struct rtc_time {
    int tm_sec;
    int tm_min;
    int tm_hour;
    int tm_mday;
    int tm_mon;
    int tm_year;
    int tm_wday;
    int tm_yday;
    int tm_isdst;
};
```

tm_mday: 一个月中的日期，取值区间为 [1,31]

tm_wday: 一个星期的第几天，星期日为 0，星期一为 1，以此类推

tm_yday: 一年中的第几天，取值区间为 [0,365]，其中 0 代表 1 月 1 日，1 代表 1 月 2 日，以此类推

tm_isdst: 判断是否为夏令时, 1 是夏令时; 0 非夏令时

3 Linux 系统中 RTC 命令

3.1 date 和 hwclock

- date 命令可以查询/更改目前 Linux 上的系统 (软件) 时钟

例: 写入时间

```
# date "2020-10-17 9:48:30"  
Sat Oct 17 09:48:30 CST 2020
```

例: 读取时间

```
# date  
Sat Oct 17 09:48:34 CST 2020
```

- Hwclock 则用于查询/更改硬件时钟 (RTC) 的时间

例: 查询硬件时间 (RTC 时间)

```
# hwclock  
Sat Oct 17 09:56:03 2020 0.000000 seconds
```

例: 将系统时间写到 RTC 时间

```
# hwclock -w
```

例: 将 RTC 时间写入系统时间

```
# hwclock -s
```

(可以在/etc/inittab 中添加 /bin/hwclock -s 系统开机时自动读取 RTC 时间, 同步到系统时钟)

4 RTC 问题排查

4.1 软件问题

- 查看设备树（对于基于设备树的系统）：

如果系统使用设备树来描述硬件，确保 RTC 设备在设备树中正确配置。检查设备节点和属性是否正确。

- 检查驱动程序和配置：

确保 RTC 设备的驱动程序正确加载并配置。检查内核日志以查看与 RTC 相关的任何错误或警告消息。如果需要，重新加载或卸载并重新加载 RTC 驱动程序。

- 使用命令行工具：

使用命令行工具（如 `hwclock`）来检查 RTC 设备的状态和时间设置。运行命令来获取 RTC 时间和日期信息。使用命令行工具设置 RTC 时间，以确保它与系统时间同步。

- 检查中断和时钟源：

如果 RTC 设备有中断输出，确保中断线正确连接并已启用。如果 RTC 设备依赖外部时钟源，确保时钟源的稳定性和正确性。

- 监视日志和警告：

定期监视系统日志以查看与 RTC 设备相关的任何警告或错误消息。借助这些消息分析 RTC 问题。

4.2 硬件问题

- 检查电源和电池

确保 RTC 设备的电源正常供应。检查电源线是否连接正确，电池是否充电或需要更换。如果 RTC 设备依赖电池备份，确保电池的电量足够以维持 RTC 数据，尤其是在系统关机或断电时。

- 确认硬件连接

检查处理器的物理连接，包括数据线、时钟线等。尤其是当 RTC 的时钟源由外部时钟提供时，确保连接稳定且正确。

- 考虑硬件故障

考虑 RTC 设备本身可能存在硬件故障的可能性。如果其他排查步骤无法解决问题，考虑更换相关处理器或进行硬件诊断。

（如果在以上步骤中未能解决问题，建议与供应商联系，寻求技术支持和建议。）