

confidential for Frilling

confidential for Frame

CVITEK RTC 操作指南

Version: 2.0.0

Release date: 2023-02-08

© 2020 北京晶视智能科技有限公司 本文件所含信息归<u>北京晶视智能科技有限公司</u>所有。 未经授权,严禁全部或部分复制或披露该等信息。

修订记录

Revision Date Author Description 1.0.1 2022/05/08 Mark-Hsieh 初稿 1.1 2022/08/02 Jin Li Change version 2.0.0 2022/02/08 Jinyu zhao cv180x/cv181x 文档兼容 Confidential For 来別が	Spec	cifications ar	e subject to char	nge without notice		KIC 探作指用
1.0.1 2022/06/08 Mark.Hsieh 初稿 1.1 2022/08/02 Jin Li Change version 2.0.0 2022/02/08 Jinyu zhao cv180x/cv181x 文档兼容						
1.0.1 2022/06/08 Mark.Hsieh 初稿 1.1 2022/08/02 Jin Li Change version 2.0.0 2022/02/08 Jinyu zhao cv180x/cv181x 文档兼容		修订证	记录	: 21 50	7//	ear for the
1.0.1 2022/06/08 Mark.Hsieh 初稿 1.1 2022/08/02 Jin Li Change version 2.0.0 2022/02/08 Jinyu zhao cv180x/cv181x 文档兼容		U. L. L.		confidentis	C 027	Eidentie
1.1 2022/08/02 Jin Li Change version 2.0.0 2022/02/08 Jinyu zhao cv180x/cv181x 文档兼容		Revision	Date	Author	Description	
2.0.0 2022/02/08 Jinyu zhao cv180x/cv181x 文档兼容		1.0.1	2022/06/08	Mark.Hsieh	初稿	
		1.1	2022/08/02	Jin Li	Change version	
onfidential for Withhill Confidential for Withhill Confidential for Withhill Confidential for		2.0.0	2022/02/08	Jinyu zhao	cv180x/cv181x 文档兼容	
onfidential for Killimet Confidential for Killimet						
onfidential for ikillimet Confidential for ikillimet Confidential for ikillimet						
onfidential for This dential for The de						
onfidential for ikillimet Confidential for ikillimet Confidential for ikillimet						
onfidential for *** thinket Confidential for *** thinket Confidential for *** thinket						

for深圳随

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。

for 深圳雁

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。

Confidential for Fithing t

for 深圳種土

or 深圳雁十 目录

	602		£0°2			£0°2	
1	RTC 操作指序	有	S.>.	•••••		5	
	1.1.	模块介绍			Co_{ij}	5	
	1.2.	计数时钟频率				5	
	1.3.	操作准备				5	
	1.4.	使用方式				5	
	1.4.1.	loctl 控制 RTC				5	
	1.4.2.	ioctl 使用范例				6	
	1.4.3.	结构体				7	
2	Linux 系统中	PRTC 命令				8	
	2.1 date	e 和 hwclock				8	

EOT 美新州旗手

Confidential for Athlet

confident

al for multiplet

RTC 操作指南

1.1. 模块介绍

RTC(real time clock)硬件时钟,为系统提供与记录时间。若 RTC 由电池供电,当芯片处于下 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。

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。

1.2. 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 秒 = 49710 天 = 136 年

1.3. 操作准备

- RTC 的操作准备如下:
- AJ採作准备如下: 使用 SDK 发布的 kernel 插入模块: insmod cv180

1.4. 使用方式

1.4.1.loctl 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

ioctl 指令功能描述		THINGS TO SEE		
指令		描述	COT TRAIN	
RTC_ALM_READ	: 2)	读取闹钟时间	al to	
RTC_ALM_SET	. denti	设置闹钟时间) ·	
RTC_RD_TIME	Court	读取时间与日期		

Spec	incations are subject to change without holic			
			<u> </u>	
	指令	描述	(深)	
	RTC_SET_TIME	读取时间与日期	1 E02	
	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		

1.4.2.ioctl 使用范例

```
static const char default_rtc[] = "/dev/rtc0";
struct rtc_time rtc_tm;
                                                                  Confidential for 深圳随着
                          Confidential for ist Milet
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);
                                                                              Confidential for TRIMET
            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:
```

Confidential

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

1.4.3.结构体

rtc_time

```
struct rtc_time {
               int tm_sec;
               int tm_min;
               int tm_hour;
               int tm_mday;
               int tm_mon;
                                                                                            ntial for *** things t
                                                  ential for *** things
               int tm_year;
               int tm_wday;
               int tm_yday;
               int tm_isdst;
confider);
                                                                                  Court
```

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

confidential for William Confidential

Linux 系统中 RTC 命令

2.1 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)

Sat Oct 17 09:56:03 2020 0.0000000 seconds Confidential fo

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)

for深圳随