



HCRTOS rtc at8563使用说明文档

1. 文档履历

版本号	日期	制/修订人	制/修订记录
1.0	2023.09.01	邱浩佳	新增文档说明

HCRTOS rtc at8563使用说明文档

1. 文档履历
2. 概述
 - 2.1 编写目的
 - 2.2 读者对象
3. 模块介绍
4. 模块接口说明
5. 模块测试用例与Sample Code
 - 5.0.1 设备树节点的添加
 - 5.0.2 menuconfig的配置
 - 5.0.2.1 驱动的配置
 - 5.0.2.2 测试命令的配置
 - 5.0.3 sample code
6. 模块调试方法
7. 常见问题

2. 概述

2.1 编写目的

介绍如何在hcrtos上使用rtc设备at8563。

2.2 读者对象

hcartos开发工程师和技术支持工程师

3. 模块介绍

- 支持时间的设置和读取。
- 支持报警功能的设置。
- 支持报警触发中断功能（触发中断时输出低电平）。

4. 模块接口说明

本模块通过ioctl进行操作，支持以下cmd。

RTC_SET_TIME: 设置时间的秒、分、时、星期、月、日、年。

RTC_RD_TIME: 可获取时间的秒、分、时、星期、月、日、年。

RTC_ALM_SET: 设置rtc的alarm功能，支持设置对应的日、小时、分钟，设置后默认开启alarm功能。

RTC_ALM_READ: 获取rtc设置的alarm，可获取到日、小时、分钟。

RTC_AIE_ON: 开启alarm功能。

RTC_AIE_OFF: 关闭alarm功能。

5. 模块测试用例与Sample Code

介绍本模块相关的测试用例及相关Sample Code

5.0.1 设备树节点的添加

```
1  i2c@0{
2      pinmux-active = <PINPAD_L28 3 PINPAD_L29 3>;
3      device_type = "hichip,hcartos-setup-setbit";
4      reg_bit = <0xb8800094 18 1>;
5      devpath = "/dev/i2c0";
6      baudrate = <100000>;
7      mode = "master";
8      status = "okay";
9  };
10 at8563{
11     irq-gpio = <PINPAD_B02 0>;           //at8563所连接的中断引脚
12     devpath = "/dev/at8563";             //at8563生成的设备节点路径
13     i2c-devpath = "/dev/i2c0";          //at8563所用的i2c节点
14     i2c-addr = <0x51>;                   //at8563的7bit地址
15     status = "okay";                     //节点状态
```

5.0.2 menuconfig的配置

5.0.2.1 驱动的配置

根据下面路径勾选at8563驱动，勾选后执行make kernel-rebuild all进行编译。

```

1  There is no help available for this option.
2  Symbol: CONFIG_HC_AT8563 [=y]
3  Type   : bool
4  Prompt: AT8563 rtc driver
5  Location:
6      -> Components
7          -> kernel (BR2_PACKAGE_KERNEL [=y])
8              -> Drivers
9                  -> rtc driver (CONFIG_DRV_RTC [=y])
10 Defined at rtc:1
11 Depends on: BR2_PACKAGE_KERNEL [=y] && CONFIG_DRV_RTC [=y]
```

5.0.2.2 测试命令的配置

根据下面路径勾选at8563测试命令，勾选后执行make cmds-rebuild all进行编译。

```

1  There is no help available for this option.
2  Symbol: CONFIG_CMDS_RTC_AT8536 [=y]
3  Type   : bool
4  Prompt: rtc at8536 test operations
5  Location:
6      -> Components
7          -> Cmds (BR2_PACKAGE_CMDS [=y])
8  Defined at source:257
9  Depends on: BR2_PACKAGE_CMDS [=y]
```

在串口控制台输入对应的命令，即可看到日期时间。

```

volume is 50--> usb notify connect v0BDApF179

hc1512a@dbB200#
hc1512a@dbB200#
hc1512a@dbB200# at8563_test
RTC_ALM_READ return (day:01) (hour:00):(min00)
RTC_RD_TIME return 2024-12-31 23:59:55
Invalid command "at8563_test"
hc1512a@dbB200# at8563_test
RTC_ALM_READ return (day:02) (hour:00):(min00)
RTC_RD_TIME return 2025-01-01 00:00:03
Invalid command "at8563_test"
hc1512a@dbB200#
```

5.0.3 sample code

sample code位于sdk绝对路径: hcrtos/components/cmds/source/rtc/rtc8537_test.c

```
1  #include <stdint.h>
2  #include <unistd.h>
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <string.h>
6  #include <getopt.h>
7  #include <fcntl.h>
8  #include <sys/ioctl.h>
9  #include <kernel/delay.h>
10 #include <kernel/lib/console.h>
11
12 #include <freertos/FreeRTOS.h>
13 #include <freertos/task.h>
14 #include <freertos/semphr.h>
15 #include <freertos/queue.h>
16 #include <kernel/lib/console.h>
17
18 #include <hcuapi/iocbase.h>
19 #include <uapi/linux/rtc.h>
20
21 static const char *device = "/dev/at8563";
22 static int i = 0;
23
24 int at8563_test(int argc, char * argv[])
25 {
26     int fd;
27     int ret;
28     struct rtc_time rtc_tm = { 0 };
29     struct rtc_time rtc_tm_temp = { 0 };
30
31     //闰月
32     rtc_tm.tm_year = 2024 - 1900; /* 需要设置的年份, 需要减1900 */
33     rtc_tm.tm_mon = 12 - 1; /* 需要设置的月份, 需要确保在0-11范围*/
34     rtc_tm.tm_mday = 31; /* 需要设置的日期*/
35     rtc_tm.tm_hour = 23; /* 需要设置的时间*/
36     rtc_tm.tm_min = 59; /* 需要设置的分钟时间*/
37
38     rtc_tm.tm_sec = 55; /* 需要设置的秒数*/
39
40     fd = open(device, O_RDWR);
41
42     if (fd < 0) {
43         printf("can't open device\n");
44         return -1;
45     }
46 }
```

```

45     struct rtc_time rtc_alarm_tm = { 0 };
46
47     struct rtc_time rtc_alarm_tm_temp = { 0 };
48
49     if (i == 0) {
50         /* set rtc time*/
51         if (ioctl(fd, RTC_SET_TIME, &rtc_tm) < 0) {
52             printf("RTC_SET_TIME failed\n");
53             close(fd);
54             return -1;
55         }
56         i++;
57     }
58
59     rtc_alarm_tm.tm_year = 0; /* 闹钟忽略年设置*/
60     rtc_alarm_tm.tm_mon = 0; /* 闹钟忽略月设置*/
61     rtc_alarm_tm.tm_mday = 1; /* 闹钟忽略日期设置*/
62     rtc_alarm_tm.tm_hour = 0; /* 需要设置的时间*/
63     rtc_alarm_tm.tm_min = 0; /* 需要设置的分钟时间*/
64     rtc_alarm_tm.tm_sec = 0; /* 需要设置的秒数,没有按秒触发alarm,大于0, 会在
min+1*/
65
66     /* set alarm time and will enabled alarm */
67     if (ioctl(fd, RTC_ALM_SET, &rtc_alarm_tm) < 0) {
68         printf("RTC_ALM_SET failed\n");
69     }
70
71     /* get alarm time */
72     if (ioctl(fd, RTC_ALM_READ, &rtc_alarm_tm_temp) < 0) {
73         printf("RTC_ALM_READ failed\n");
74     }
75
76     printf("RTC_ALM_READ return (day:%02d) (hour:%02d):(min%02d)\n",

```

```
74         rtc_alarm_tm_temp.tm_mday, rtc_alarm_tm_temp.tm_hour,
75         rtc_alarm_tm_temp.tm_min);
76
77 #if 0
78     /* enable alarm time */
79     if (ioctl(fd, RTC_AIE_ON, 1) < 0) {
80         printf("RTC_AIE_ON failed\n");
81     }
82
83     /* disable alarm time */
84     if (ioctl(fd, RTC_AIE_OFF, 0) < 0) {
85         printf("RTC_AIE_OFF failed\n");
86     }
87 #endif
88
89     /* read rtc time */
90     if (ioctl(fd, RTC_RD_TIME, &rtc_tm_temp) < 0) {
91         printf("RTC_RD_TIME failed\n");
92         close(fd);
93         return -1;
94     }
95     printf("RTC_RD_TIME return %04d-%02d-%02d %02d:%02d:%02d\n",
96         rtc_tm_temp.tm_year + 1900, rtc_tm_temp.tm_mon + 1,
97         rtc_tm_temp.tm_mday, rtc_tm_temp.tm_hour, rtc_tm_temp.tm_min,
98         rtc_tm_temp.tm_sec);
99
100     close(fd);
101
102     return ret;
103 }
```

104

105 `CONSOLE_CMD(at8563_test, NULL, at8563_test, CONSOLE_CMD_MODE_SELF, "rtc
at8563")`

6. 模块调试方法

暂无。

7. 常见问题

暂无。