

# Linux驱动|rtc-hym8563移植笔记

原创 土豆居士 一口Linux 2022-09-27 11:50 发表于江苏

收录于合集

#瑞芯微 2 #Linux驱动 53 #所有原创 196

点击左上方蓝色“一口Linux”，选择“设为星标”

第一时间看干货文章

Linux知识图谱

 一口Linux甄选

¥25

购买

📖 【干货】[嵌入式驱动工程师学习路线](#)

📖 【干货】[Linux嵌入式知识点-思维导图](#)



本文基于瑞芯微rk3568平台，关于该平台快速入手操作，大家可以参考以下文章：

《[瑞芯微rk356x板子快速上手](#)》

## 0、什么是rtc-hym8563？

RTC：实时时钟的缩写是(Real Time Clock)。RTC 是集成电路，通常称为时钟芯片。

### 1) 产品概述

HYM8563是一款低功耗CMOS实时时钟/日历芯片，由武汉昊昱公司设计生产。



它提供一个可编程的时钟输出，一个中断输出和一个掉电检测器，所有的地址和数据都通过I2C总线接口串行传递。

最大总线速度为400Kbits/s，每次读写数据后，内嵌的字地址寄存器会自动递增。

## 2) 产品特点

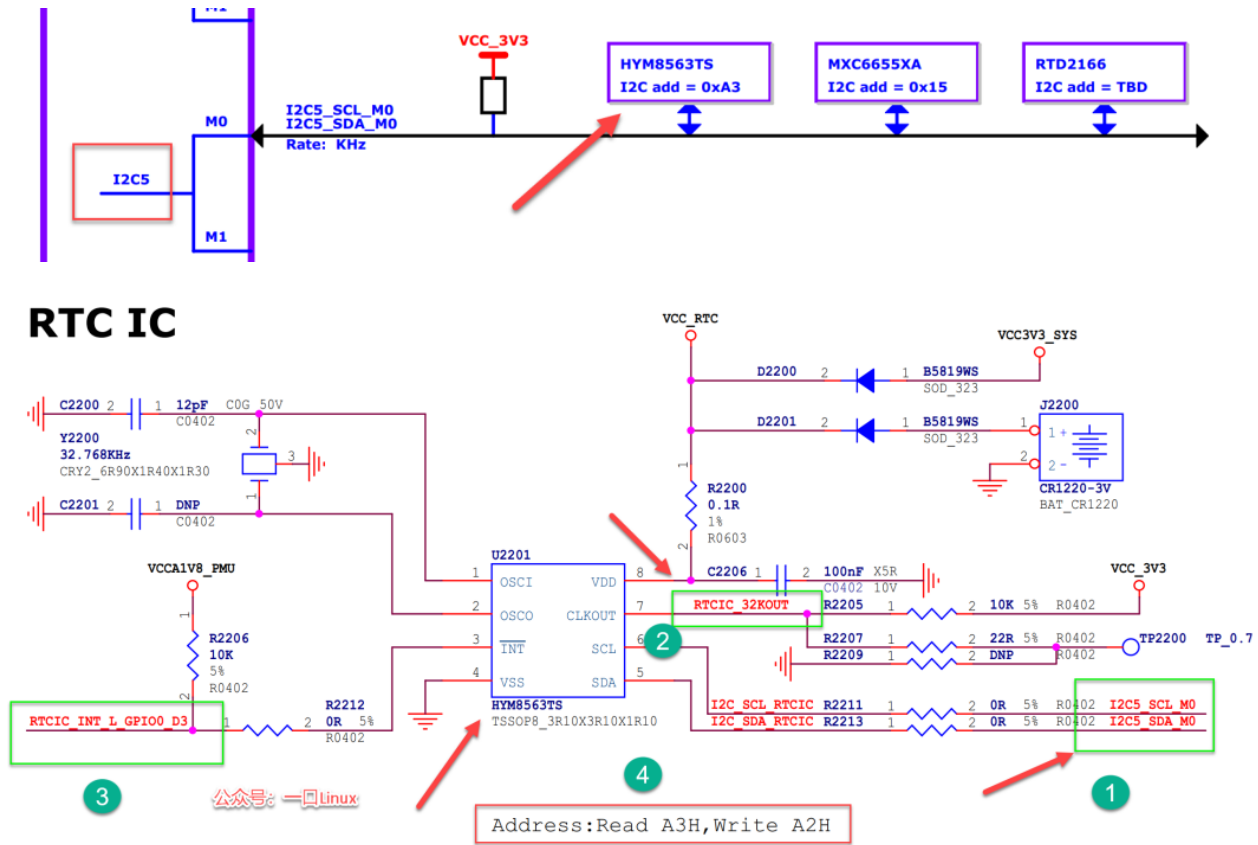
- 可计时基于32.768kHz晶体的秒，分，小时，星期，天，月和年
- 带有世纪标志
- 宽工作电压范围：1.8~5.5V
- 低休眠电流：典型值为0.25μA（VDD=3.0V，TA=25℃）
- 400kHz的I2C总线接口（VDD=1.8~5.5V时）
- 可编程时钟输出频率为：32.768kHz，1024Hz，32Hz，1H
- 报警和定时器
- 掉电检测器
- 内部集成振荡电容
- 片内电源复位功能
- I2C总线从地址：读，0A3H；写，0A2H
- 漏极开路中断引脚
- 封装形式：DIP8和SOP8

## 一、移植基于平台

```
soc : rk3568
board: EVB1-DDR4-V10
软 件: Android 11
Linux: 4.19.232
```

## 二、移植步骤

1) 电路图



由以上两个图可得硬件信息如下：

- 1. RTC连接到I2C的通道5
- 2. CLKOUT引脚对应RTCIC\_32KOUT
- 3. INT连接的是RTCIC\_INT\_L\_GPIO0\_D3，即GPIO0的bit3
- 4. 从设备地址为0x51（0101 0001），read: 1010 0011 write: 1010 0010

2) 设备树

内核中已有关于hym8563的设备树节点信息说明：

```
Documentation/devicetree/bindings/rtc/haoyu,hym8563.txt

1 Haoyu Microelectronics HYM8563 Real Time Clock
2
3 The HYM8563 provides basic rtc and alarm functionality
4 as well as a clock output of up to 32kHz.
5
6 Required properties:
7 - compatible: should be: "haoyu,hym8563"
8 - reg: i2c address
9 - #clock-cells: the value should be 0
```

```

10
11 Optional properties:
12 - clock-output-names: From common clock binding
13 - interrupts: rtc alarm/event interrupt
14
15 Example:
16
17 hym8563: hym8563@51 {
18     compatible = "haoyu,hym8563";
19     reg = <0x51>;
20
21     interrupts = <13 IRQ_TYPE_EDGE_FALLING>;
22
23     #clock-cells = <0>;
24 };
25
26 device {
27     ...
28     clocks = <&hym8563>;
29     ...
30 };

```

CSDN @一口Linux

同时在瑞芯微提供的个厂家sdk中，已经由类似的设备树节点实例，所以我们可以直接参考该内容：

```

peng@ubuntu:~/work/rk_android11.0_sdk_220718/kernel$ grep hym8563 arch/arm64/boot/dts/rockchip/* -nr
arch/arm64/boot/dts/rockchip/rk3368-orion-r68-meta.dts:207:     hym8563: hym8563@51 {
arch/arm64/boot/dts/rockchip/rk3368-orion-r68-meta.dts:208:         compatible = "haoyu,hym8563";
arch/arm64/boot/dts/rockchip/rk3368-r88-dcdc.dts:326:     hym8563: hym8563@51 {
arch/arm64/boot/dts/rockchip/rk3368-r88-dcdc.dts:327:         compatible = "haoyu,hym8563";
arch/arm64/boot/dts/rockchip/rk3368-r88.dts:63:         clocks = <&hym8563>;
arch/arm64/boot/dts/rockchip/rk3368-r88.dts:191:     hym8563: hym8563@51 {
arch/arm64/boot/dts/rockchip/rk3368-r88.dts:192:         compatible = "haoyu,hym8563";
arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v10.dtsi:166:     hym8563: hym8563@51 {
arch/arm64/boot/dts/rockchip/rk3568-nvr-demo-v10.dtsi:167:         compatible = "haoyu,hym8563";
164 &i2c1 {
165     status = "okay";
166     hym8563: hym8563@51 {
167         compatible = "haoyu,hym8563";
168         reg = <0x51>;
169
170         pinctrl-names = "default";
171         pinctrl-0 = <&rtc_int>;
172
173         interrupt-parent = <&gpio0>;
174         interrupts = <RK_PD3 IRQ_TYPE_LEVEL_LOW>;
175     };
176 };

```

下面是pinctrl关于rtc引脚的说明：

```

423 &pinctrl {
424     rk618 {
425         rk618_reset: rk618-reeset {
426             rockchip,pins = <1 RK_PB2 RK_FUNC_GPIO &pcfg_output_high>;
427         };
428         rk618_int: rk618-int {
429             rockchip,pins = <0 RK_PB0 RK_FUNC_GPIO &pcfg_pull_none>;
430         };
431     };
432
433     rtc {
434         rtc_int: rtc-int {
435             rockchip,pins = <0 RK_PD3 RK_FUNC_GPIO &pcfg_pull_up>;
436         };
437     };
438

```

结合该说明文档，根据硬件信息，填充设备树信息到文件

```
kernel\arch\arm64\boot\dts\rockchip\rk3568-evb.dtsi
```

添加信息如下：

```

1393 &i2c5 {
1394     status = "okay";
1395     hym8563: hym8563@51 {
1396         compatible = "haoyu,hym8563";
1397         reg = <0x51>;
1398
1399         pinctrl-names = "default";
1400         pinctrl-0 = <&rtc_int>;
1401
1402         interrupt-parent = <&gpio0>;
1403         interrupts = <RK_PD3 IRQ_TYPE_LEVEL_LOW>;
1404     };
1405
1406
1468 &pinctrl {
1469     rtc {
1470         rtc_int: rtc-int {
1471             rockchip,pins = <0 RK_PD3 RK_FUNC_GPIO &pcfg_pull_up>;
1472         };
1473     };
1474

```

可以看到该节点内容和硬件电路图是对应起来的。

### 3) 驱动

hym8563驱动程序位于

```
drivers/rtc/rtc-hym8563.c
```

打开该驱动

```
arch/arm64/configs/rockchip_defconfig
```

```
549 CONFIG_REGULATOR_FIXED_VOLTAGE=y
```

#### 4) 打开hctosys驱动权限

为了支持硬件的时钟(RTC)能与系统时间同步，内核需要对应驱动支持：

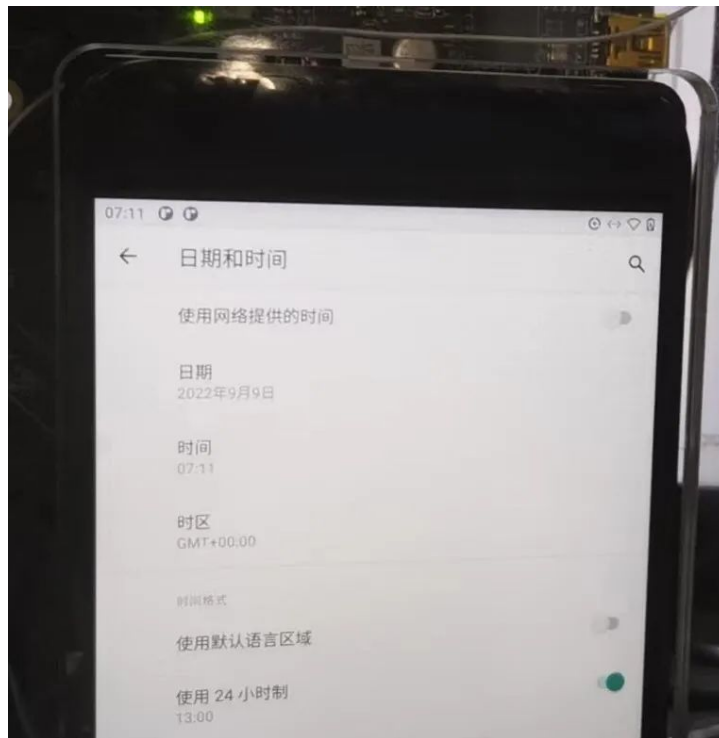
4434	CONFIG_RTC_HCTOSYS=y	允许RTC时间设置到系统时间
4435	CONFIG_RTC_HCTOSYS_DEVICE="rtc0"	默认同步时间的RTC设备
4436	CONFIG_RTC_SYSTOHC=y	允许系统时间设置到RTC
4437	CONFIG_RTC_SYSTOHC_DEVICE="rtc0"	默认同步时间的RTC设备

修改完毕，重新编译内核和设备树，烧录内核即可

### 三、测试rtc

#### 测试1) 测试rtc驱动

在UI界面上 设置->系统->日期和时间->时间 可以读取和设置新的时间，



设置完毕使用**hwclock**查看时间

如果时间一致，可以设置说明rtc驱动起了作用，

```

WM_CHAR(0x6B='k', Scan=0, IParam=0x00000001) must be processed internally in CConEm...
C:\Users\Administrator
λ
C:\Users\Administrator
λ
C:\Users\Administrator
λ adb shell
rk3568_r:/ $
rk3568_r:/ $
rk3568_r:/ # hwclock
hwclock: /sys/class/rtc: Permission denied
hwclock: /dev/misc/rtc: No such file or directory
1|rk3568_r:/ # su
rk3568_r:/ # hwclock
Wed Sep 21 02:56:06 2022 0.000000 seconds
rk3568_r:/ # hwclock Wed Sep 21 02:56:06 2022 0.000000 seconds rk3568_r:/ #
Permission denied
No such file or directory
0.000000 seconds
adb.exe Search

```

【也可以在驱动的read函数中，增加log，更加直观】

```

.read_time = hym8563_rtc_read_time,
.set_time = hym8563_rtc_set_time,

```

## 测试2) 测试rtc关机后时间同步

1. 设置当前时间和自己手机同步
2. 拔掉电源，等待几分钟开机
3. 确认板子时间是否和当前手机时间一致

## 四、其他查看rtc方法

同时在以下目录中也可以直接读取驱动相关的信息。

```
rk3568_r:/sys/class/rtc # ls
rtc0
rk3568_r:/sys/class/rtc # cd rtc0
rk3568_r:/sys/class/rtc/rtc0 # cd ..
rk3568_r:/sys/class/rtc # ls -l
total 0
lrwxrwxrwx 1 root root 0 2021-01-01 12:00 rtc0 -> ../../devices/platform/fe5e0000.i2c/i2c-5/5-0
rk3568_r:/sys/class/rtc # cd rtc0
rk3568_r:/sys/class/rtc/rtc0 # ls
date device max_user_freq power subsystem uevent wakeup8
dev hctosys name since_epoch time wakealarm
rk3568_r:/sys/class/rtc/rtc0 # cat name
rtc-hym8563 5-0051
rk3568_r:/sys/class/rtc/rtc0 # cat time
03:04:57
rk3568_r:/sys/class/rtc/rtc0 # cat hctosys
1
```

## 五、遇到的问题

### 问题1) /sys/class/rtc下有多个rtc设备，需要去掉rtc-rk808驱动

共板里除了hym8563，在pmic中还有内部的rtc-rk808，我们需要把该rtc驱动去掉，否则/sys/class/rtc下会有2个rtc设备 在以下文件

```
arch/arm64/configs/rockchip_defconfig
```



```
808 CONFIG_RTC_DRV_RK808=n
```

该驱动位于

```
drivers/rtc/rtc-rk808.c
```

## 问题2) ui界面设置的时间无法同步到hym8563中

打开开机后的logcat, 发现有如下信息

```
01-01 12:00:14.256 439 439 E AlarmManagerService: failed to open /sys/class/rtc/rtc0/hctosys
01-01 12:00:14.256 439 439 W AlarmManagerService: no wall clock RTC found
01-01 12:00:14.260 439 439 D AlarmManagerService: Kernel timezone updated to 0 minutes west
01-01 12:00:14.261 439 439 D AlarmManagerService: Setting time of day to sec=1663584459

09-19 10:47:39.000 439 439 W AlarmManagerService: Unable to set rtc to 1663584459: No such
```

该log位于以下文件:

```
frameworks\base\services\core\jni\com_android_server_AlarmManagerService.cpp
```

该log位于263行:

```
257 static bool rtc_is_hctosys(unsigned int rtc_id)
258 {
259     android::String8 hctosys_path = String8::format("%s/rtc%u/hctosys",
260         rtc_sysfs, rtc_id);
261     FILE *file = fopen(hctosys_path.string(), "re");
262     if (!file) {
263         ALOGE("failed to open %s: %s", hctosys_path.string(), strerror(errno));
264         return false;
265     }
266 }
```

```
267     unsigned int hctosys;
268     bool ret = false;
269     int err = fscanf(file, "%u", &hctosys);
270     if (err == EOF)
271         ALOGE("failed to read from %s: %s", hctosys_path.string(),
272             strerror(errno));
273     else if (err == 0)
274         ALOGE("%s did not have expected contents", hctosys_path.string());
275     else
276         ret = hctosys;
277
278     fclose(file);
279     return ret;
280 }
```

由程序可知，没有权限打开文件

```
/sys/class/rtc/rtc0/hctosys
```

增加该文件读取权限，打开文件

```
device/rockchip/common/sepolicy/vendor/genfs_contexts
```

增加以下内容：

```
28 # rk356x RTC
29 genfscon sysfs /devices/platform/fe5e0000.i2c/i2c-5/5-0051/rtc/rtc0/hctosys
```

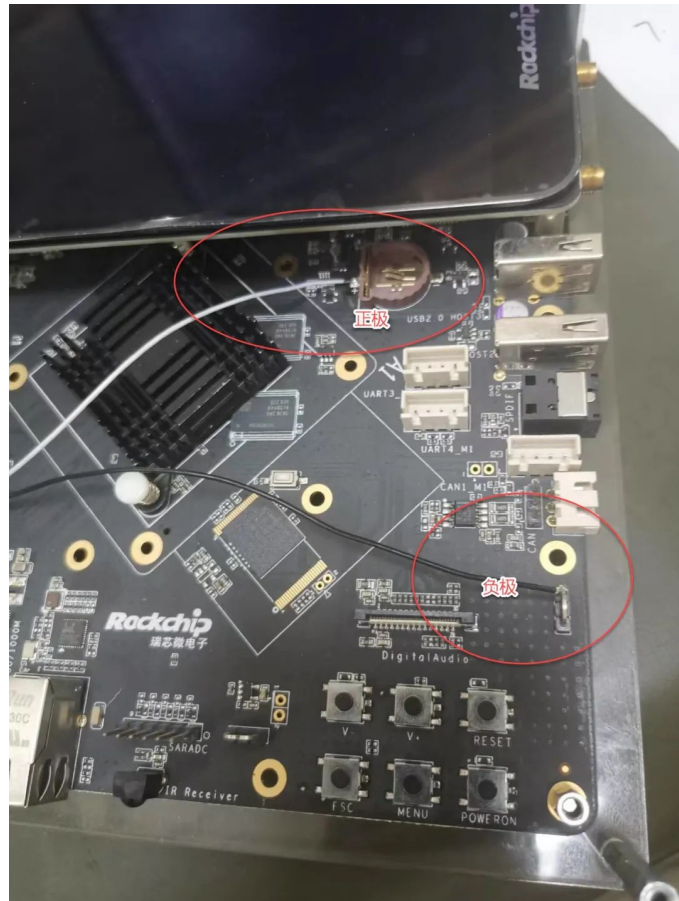
重新编译安卓即可。

注意 `/sys/class/rtc/rtc0/hctosys` 其实对应的文件位置  
是 `/sys/devices/platform/fe5e0000.i2c/i2c-5/5-0051/rtc/rtc0/hctosys`

```
rk3568_r:/sys/class/rtc # ls -l total 0 lrwxrwxrwx 1 root root 0 2021-01-01 12:00 rtc0  
-> ../../devices/platform/fe5e0000.i2c/i2c-5/5-0051/rtc/rtc0
```

### 问题3) 没有纽扣电池，需要自己飞线，接稳压电源

如下图，注意正负极，电压设置为3v即可



后续还会继续编写Linux内核的rtc 时间子系统的文章，敬请关注！

## 六、欢迎交流

一口君建立了瑞芯微的技术交流群，

大家工作中用到瑞芯微系列soc的，可以一起交流，

加群后台留言即可。

nd

关注，回复【1024】海量Linux资料赠送



一口Linux

一起学习嵌入式，Linux，网络，驱动，arm知识。  
225篇原创内容

公众号

精彩文章合集

文章推荐

- 【专辑】[ARM](#)
- 【专辑】[粉丝问答](#)
- 【专辑】[所有原创](#)
- 【专辑】[linux入门](#)
- 【专辑】[计算机网络](#)
- 【专辑】[Linux驱动](#)
- 【干货】[嵌入式驱动工程师学习路线](#)
- 【干货】[Linux嵌入式所有知识点-思维导图](#)

收录于合集 #瑞芯微 2

下一篇 · 瑞芯微rk356x板子快速上手

阅读原文

喜欢此内容的人还喜欢

Linux inxi 命令的 3 种使用方法 | Linux 中国  
Linux中国



一个披着 Windows 外壳的轻量级 Linux 系统  
K8S中文社区



Window Linux 双系统安装历程，超详细





Linux学习