# 概述

在河道流速测速仪项目中，为了达成低功耗设计，在非工作状态下，需要关闭CPU，这样就需要有定时机制能定时打开CPU供电。由于AM4379在断电状态下是不能通过内部RTC唤醒的，所以在方案中选用了外部RTC芯片DS1374，通过I2C和CPU连接。本文档描述了如何完成DS1374的Linux驱动设计

# 版本信息

本文档中描述的操作基于创龙的AM4379开发板配套SDK进行修改，Linux内核版本号为3.14.43

# U-boot

u-boot部分的修改完成管脚分配部分的修改，通过修改board/ti/am43xx/mux.c文件完成：

创龙的开发板原设计中之使用了I2C0一条I2c总线，而流速仪设计中将DS1374挂接到I2C1中，因此需要增加I2C1的管脚配置：

static struct module\_pin\_mux i2c1\_pin\_mux**[]** **=** **{**

**{**OFFSET**(**spi0\_d1 **),** **(**MODE**(**2**)** **|** PULLUP\_EN **|** RXACTIVE **|** SLEWCTRL**)},**

**{**OFFSET**(**spi0\_cs0**),** **(**MODE**(**2**)** **|** PULLUP\_EN **|** RXACTIVE **|** SLEWCTRL**)},**

**{-**1**},**

**};**

I2C1的管脚使用的是spi0\_d1和spi0\_cs0这两个信号（T20,T21），其中spi0\_cs0信号在开发板中是作为gpio管脚使用的，因此还需要将对应的定义注释：

*//static struct module\_pin\_mux gpio5\_7\_pin\_mux[] = {*

*// {OFFSET(spi0\_cs0), (MODE(7) | PULLUP\_EN)}, /\* GPIO5\_7 \*/*

*// {-1},*

*//};*

然后在enable\_board\_pin\_mux函数中对i2c1进行配置，同时注释掉gpio5\_7的配置语句：

void enable\_board\_pin\_mux**(**void**)**

**{**

configure\_module\_pin\_mux**(**mmc0\_pin\_mux**);**

configure\_module\_pin\_mux**(**i2c0\_pin\_mux**);**

configure\_module\_pin\_mux**(**i2c1\_pin\_mux**);**

configure\_module\_pin\_mux**(**mdio\_pin\_mux**);**

**if** **(**board\_is\_gpevm**())** **{**

*// configure\_module\_pin\_mux(gpio5\_7\_pin\_mux);*

修改完成后对u-boot进行重新编译

# Kernel

## Devicetree（设备树）文件

在dts文件中找到&i2c1子项，将status参数设为“okay”，同时增加时钟频率的参数设置，在流速仪的设计中，i2c1总线上只挂接了DS1374一个设备，因此直接将时钟设置为DS1374的频率400K；

然后增加DS1374对应的配置说明，对应的名称“ds1374”，与驱动程序rtc-ds1374中i2c\_device\_id中的名称保持一致；0x68是ds1374的I2C地址；

&i2c1 {

clock-frequency = <400000>;

status = "okay";

pinctrl-names = "default", "sleep";

pinctrl-0 = <&i2c1\_pins\_default>;

pinctrl-1 = <&i2c1\_pins\_sleep>;

ds1374: ds1374@68 {

compatible = "dallas,ds1374";

reg = <0x68>;

};

之前提到，I2C1下只下挂了DS1374一个设备，因此将I2C1子项中其他设备的描述删除。注意，由于删除了ov2659和tlv320这两个设备的描述，同时要删除DTS文件中其它部分对这个两个描述的引用，对应：

ov2659：vpfe0子项中port的描述

&vpfe0 {

status = "disabled";

pinctrl-names = "default", "sleep";

pinctrl-0 = <&vpfe0\_pins\_default>;

pinctrl-1 = <&vpfe0\_pins\_sleep>;

#if 0

/\* Camera port \*/

port {

vpfe0\_ep: endpoint {

remote-endpoint = <&ov2659\_1>;

if\_type = <2>;

bus\_width = <8>;

hdpol = <0>;

vdpol = <0>;

};

};

#endif

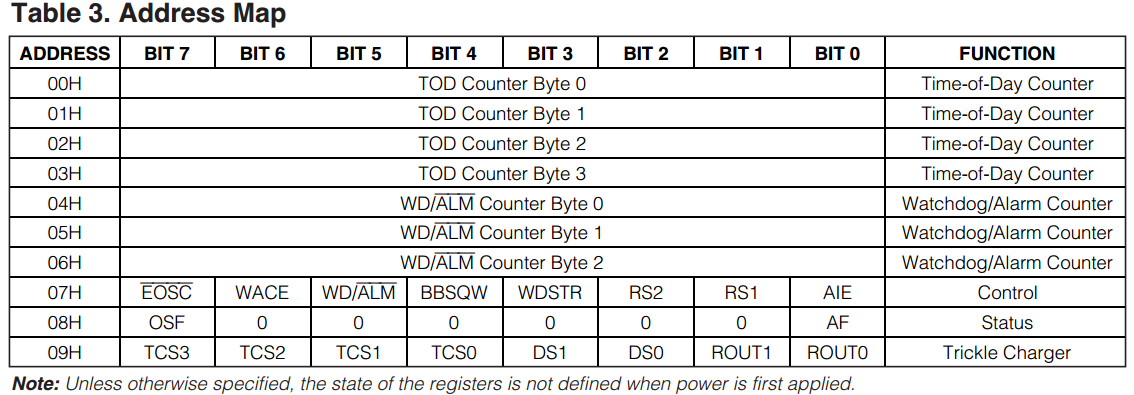
};

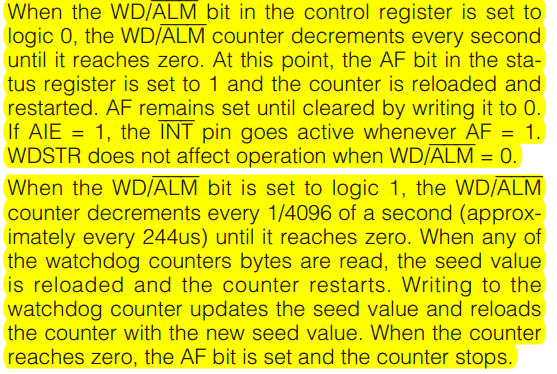
tl320：将sound0子项整个注释掉（测速仪中没有使用声卡）；

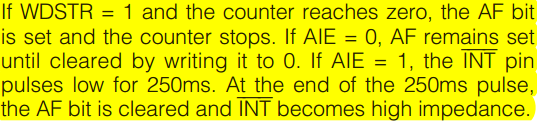
## .config

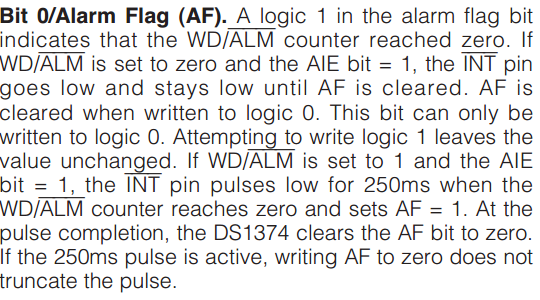
通过make menuconfig修改内核编译的配置，在device driver->real time clock中打开ds1374的编译选项

完成上述修改后，重新生成dtb和编译内核（注意先使用make distclean进行清理），烧写到flash后启动系统，进入/dev中应该可以查看到增加了一个rtc设备/dev/rtc1，这个对应的就是DS1374，至此，完成DS1374驱动的修改工作。



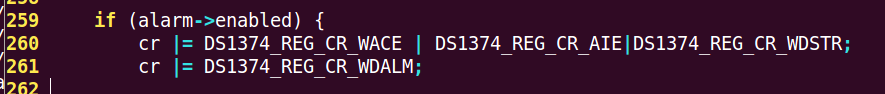






根据手册 在内核中修改drivers/rtc/rtc-ds1374.c中的

static int ds1374\_set\_alarm(struct device \*dev, struct rtc\_wkalrm \*alarm)



与static int ds1374\_alarm\_irq\_enable(struct device \*dev, unsigned int enabled)

