



A20 Standby 使用文档

V1.0

2013-03-15

Revision History

Version	Date	Changes compared to previous issue
v1.0	2013-03-15	初建版本



目录

1. 概述-----	4
1.1. 编写目的-----	4
1.2. 适用范围-----	4
1.3. 相关人员-----	4
2. 模块介绍-----	5
2.1. 模块功能介绍-----	5
2.2. 相关术语介绍-----	5
2.3. 模块配置介绍-----	5
2.3.1. 内核对 super standby 支持-----	5
2.3.2. 普通模块对 super standby 的支持-----	5
2.3.3. Late_resume 部分的特别处理-----	6
2.4. 源码结构介绍-----	6
3. 接口描述-----	8
3.1. 用户与内核交互接口-----	8
3.1.1. Linux 层-----	8
3.1.2. Android-----	8
3.2. 模块与内核交互接口-----	8
4. demo-----	10
4.1. 普通模块对 super standby 的支持-----	10
4.2. Late_resume 部分的特别处理-----	10
5. 总结-----	12
5.1. 模块与内核交互接口的改变-----	12
6. Declaration-----	13

1. 概述

1.1. 编写目的

该设计文档的目的在于，简要介绍，为了支持超级 standby，各模块负责人需要完成的工作。

1.2. 适用范围

介绍本模块设计适用于基于 A20 的平台。

1.3. 相关人员

预期读者为 standby 客户案支持人员。

2. 模块介绍

2.1. 模块功能介绍

对于嵌入式设备尤其是移动设备来说，功耗是系统的重要指标，系统设计的重要目标之一就是尽可能地降低功耗。

Standby 模块，在特定策略控制下，通过与 CPU，内存、显示屏，gpu 等相协调，支持对一系列电压和频率的快速调节，从而降低嵌入式系统的功耗。

Standby 模块，支持两种 standby 模式：normal standby + super standby.

2.2. 相关术语介绍

Normal Standby (standby) : CPU and RAM are powered but not executed

super standby: mean Suspend to RAM (mem) , cpu is powered off, while RAM is powered and the running content is saved to RAM

超级 standby: 等同于 suspend to ram;。

2.3. 模块配置介绍

要开启 super standby 的支持，请确认在 sys_config.fex 中进行了配置，正确的配置如下：

```
[pm_para]
standby_mode      = 1
```

2.3.1. 内核对 **super standby** 支持

内核导出了两个符号：standby_type, standby_level 以利于各模块：

根据目标区分 normal standby 和 super standby.

根据掉电情况，区别对待设备；

standby_type: 表目标，支持 NORMAL_STANDBY, SUPER_STANDBY;

standby_level : 表结果，支持 STANDBY_WITH_POWER_OFF , STANDBY_WITH_POWER;

2.3.2. 普通模块对 **super standby** 的支持

各模块需要实现 suspend+resume 接口，以支持超级 standby, 确保系统唤醒后，能正常稳定的工作；

1. 包含头文件 linux/pm.h;

2. 判断 standby 类型,并进行相应处理;

示例:

```
if(NORMAL_STANDBY== standby_type){  
    //process for normal standby  
}else if(SUPER_STANDBY == standby_type){  
    //process for super standby  
}
```

2.3.3. Late_resume 部分的特别处理

Early_suspend + late_resume 部分的特别处理:

对于处于 early_suspend 部分的模块,在唤醒时,模块可能处于两种状态:

1. 系统并没有进入 super standby 状态,就因为某种原因唤醒了。
2. 系统进入了 super standby 状态,由用户或其他信号唤醒。

取决于系统是否进入了 super standby 状态,模块在唤醒时就有两种可能的状态:带电或掉电;为了让模块清楚的知道,模块唤醒时,系统是否真正进入 super standby,从而影响了模块的带电状态,内核导出了另一个全局变量: standby_level;若模块认为有需要对带电和不带电两种状态进行区分处理,从而加速唤醒过程,可借助此变量的帮助。

示例:

```
if(NORMAL_STANDBY== standby_type){  
    //process for normal standby  
}else if(SUPER_STANDBY == standby_type){  
    //process for super standby  
    if(STANDBY_WITH_POWER_OFF == standby_level){  
        //处理模块经过掉电后的恢复  
    }else if(STANDBY_WITH_POWER == standby_level){  
        //处理模块带电状态的恢复  
    }  
}
```

2.4. 源码结构介绍

内核提供的公用代码:

linux-3.3\kernel\power\

linux-3.3\drivers\base\power\

linux-3.3\drivers\base\

平台相关代码:



linux-3.3\arch\arm\mach-sun6i\pm

linux-3.3\arch\arm\mach-sun6i\pm**.[hcS]”: normal standby, super standby 公用代码;

linux-3.3\arch\arm\mach-sun6i\pm\standby**.[hcS]”: normal standby 相关代码;

linux-3.3\arch\arm\mach-sun6i\pm\standby\super**.[hcS]”: super standby 相关代码;

各驱动模块源码:

Confidential

3. 接口描述

3.1. 用户与内核交互接口

通过 sys 文件系统的节点： /sys/power/state 与上层应用交互；

3.1.1. Linux 层

通过 echo mem > /sys/power/state ， 控制系统进入 standby 状态；

3.1.2. Android

按 power 键，执行 early_suspend 部分；若无 usb 连接，会在 5s 后，进入 kernel 的 suspend，直至掉电；

3.2. 模块与内核交互接口

各模块需实现 suspend, resume 接口；

功能说明：

1. Suspend: 将所属模块带入低功耗状态，受此影响，该模块可能不能正常工作，其对应的软件服务单元，也应该被禁止运行；
2. Resume: 恢复所属模块至进入 suspend 之前的状态。同时，该模块对应的软件服务单元，也应根据进入 suspend 前的状态，决定是否处于服务状态。

接口说明：

```
static struct bus_driver l3gd20_driver = {
    .class = I2C_CLASS_HWMON,
    .suspend = XXXX;-----→ (2)
    .resume = XXXXX;
    .driver = {
        .owner = THIS_MODULE,
        .name = L3GD20_GYR_DEV_NAME,
        .pm = &l3gd20_pm;-----→ (1)
    },
    .probe = l3gd20_probe,
    .remove = __devexit_p(l3gd20_remove),
    .id_table = l3gd20_id,
    .address_list = normal_i2c,
```


};

各模块需初始化(1)或(2)处的 suspend, resume 接口，
若 (1) 被初始化，则调用 (1)，否则
若 (2) 被初始化，则调用 (2)，否则不执行 suspend, resume 操作；

Confidential

4. demo

给出模块的使用 demo。如有必要，给出 demo 的设计。

4.1. 普通模块对 **super standby** 的支持

参考 sw-ts 对 resume 操作的处理：

```
static void sun4i_ts_resume(struct early_suspend *h)
{
    /*int ret;
    struct sun4i_ts_data *ts = container_of(h, struct sun4i_ts_data, early_suspend);
    */
    /*process for normal standby*/
    if(NORMAL_STANDBY == standby_type) {
        writel(STYLUS_UP_DEBOUNCE|STYLUS_UP_DEBOUCE_EN|TP_DUAL_EN|TP_
MODE_EN,TP_BASSADDRESS + TP_CTRL1);
        /*process for super standby*/
    } else if(SUPER_STANDBY == standby_type) {
        tp_init();
    }
    return ;
}
```

4.2. **Late_resume** 部分的特别处理

参考 gt818_ts 对 resume 操作的处理：

```
if(STANDBY_WITH_POWER_OFF == standby_level){
    //reset
    ctp_ops.ts_reset();
    //wakeup
    ctp_ops.ts_wakeup();

    //set to input floating
    gpio_set_one_pin_io_status(gpio_wakeup_hdle, 0, "ctp_wakeup");

    ret=goodix_init_panel(ts);

} else if(STANDBY_WITH_POWER == standby_level){
    if(ts->use_irq) {
```



```
ret = ctp_ops.set_irq_mode("ctp_para", "ctp_int_port", CTP_IRQ_MODE);
if(0 != ret){
    printk("%s:ctp_ops.set_irq_mode err. \n", __func__);
    return ret;
}
}
else {
    //.....
}
}
```

因为 wakeup, reset 这类操作需要较长的延时,仅在掉电的条件下,才进行 ctp 的 reset, wakeup 操作,可有效加快未进入 super standby 状态时的唤醒速度.

5. 总结

5.1. 模块与内核交互接口的改变

模块与内核交互接口较多，本文档，只描述了最通常的交互接口，后期会根据需要及对休眠唤醒理解的加深，启用某些接口，相应的，本文档会修订。

6. Declaration

This(A20 Standby 使用文档) is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.