

蜗窝科技

慢下来，享受技术。



博客 项目 讨论区 关于蜗窝 联系我们 登录

linux内核中的GPIO系统之（1）：软件框架

作者：linuxer 发布于：2014-7-21 14:40 分类：GPIO子系统

一、前言

作为一个工作多年的系统工程师，免不了做两件事情：培训新员工和给新员工分配任务。对于那些刚刚从学校出来的学生，一般在开始的时候总是分配一些非常简单的任务，例如GPIO driver、LED driver。往往CPU datasheet的关于GPIO或者IO ports的章节都是比较简单的，非常适合刚入行的工程师。虽然GPIO子系统相关的硬件比较简单，没有复杂的协议，不过，对于软件抽象而言，其分层次的软件思想是每个嵌入式软件工程师需要掌握的内容。

我更倾向使用GPIO系统这个名字来代替GPIO driver这个名字，GPIO driver仅仅包含了pin signal状态控制和读取的内容，而GPIO系统包括了pin multiplexing、pin configuration、GPIO control、GPIO interrupt control等内容。本文主要是以3.14内核作为例子，讲述linux kernel中GPIO系统的软件框架。

二、GPIO相关硬件有哪些差异

嵌入式工程师总是要处理各种各样的target board，每个target board上的GPIO总是存在不同，例如：

1、和CPU的连接方式不同

对于ARM的嵌入式硬件平台，SOC本身可以提供大量的IO port，SOC上的GPIO controller是通过SOC的总线（AMBA）连接到CPU的。对于嵌入式系统而言，除了SOC的IO port，一些外设芯片也可能会提供IO port，例如：

（1）有些key controller芯片、codec或者PMU的芯片会提供IO port

（2）有些专用的IO expander芯片可以扩展16个或者32个GPIO

从硬件角度看，这些IO和SOC提供的那些IO完全不同，CPU和IO expander是通过I2C（也有可能是SPI等其他类型的bus）连接的，在这种情况下，访问这些SOC之外的GPIO需要I2C的操作，而控制SOC上的GPIO只需要写寄存器的操作。不要小看这个不同，写一个SOC memory map的寄存器非常快，但是通过I2C来操作IO就不是那么快了，甚至，如果总线繁忙有可能阻塞当前进程，这种情况下，内核同步机制必须有所区别（如果操作GPIO可能导致sleep，那么同步机制不能采用spinlock）。

2、访问方式不同

SOC片内的GPIO controller和SOC片外的IO expander的访问当然不一样，不过，即便都是SOC片内的GPIO controller，不同的ARM芯片，其访问方式也不完全相同，例如：有些SOC的GPIO controller会提供一个寄存器来控制输出电平。向寄存器写1就是set high，向寄存器写0就是set low。但是有些SOC的GPIO controller会提供两个寄存器来控制输出电平。向其中一个寄存器写一就是set high，向另外一个寄存器写一就是set low。

3、配置方式不同

即便是使用了同样的硬件（例如都使用同样的某款SOC），不同硬件系统上GPIO的配置不同。在一个系统上配置为输入，在另外的系统上可能配置为输出。

4、GPIO特性不同。这些特性包括：

（1）是否能触发中断。对一个SOC而言，并非所有的IO port都支持中断功能，可能某些处理器只有一两组GPIO有中断功能。

（2）如果能够触发中断，那么该GPIO是否能够将CPU从sleep状态唤醒

（3）有些有软件可控的上拉或者下拉电阻的特性，有的GPIO不支持这种特性。在设定为输入的时候，有的GPIO可以设定debounce的算法，有的则不可以。

5、多功能复用

有的GPIO就是单纯的作为一个GPIO出现，有些GPIO有其他的复用的功能。例如IO expander上的GPIO只能是GPIO，但是SOC上的某个GPIO除了做普通的IO pin脚，还可以是SPI上clock信号线。

站内搜索

功能

留言板
评论列表

随机文章

蓝牙协议分析(4)_IPv6 Over BLE介绍
process credentials
Linux I2C framework(3)_I2C consumer
Unix的历史
显示技术介绍(3)_CRT技术

最新评论

lover713814
@wilson：可以加我微信
zhangfan747551 ...
wowo
@franc：博客和论坛的账号不能通用，另外建议大家不要登录...
franc
无法登录
wowo
@Gute：我对BLE也是边理解边记录，大家可以多交流~
wowo
@tinylaker：多谢建议，下次整理的时候，一定填上！

文章分类

Linux内核分析(10)
统一设备模型(12)
电源管理系统(40)
中断子系统(14)
进程管理(5)
内核同步机制(17)
GPIO子系统(3)
时间子系统(13)
通信类协议(4)
内存管理(5)
图形子系统(1)
Linux应用技巧(11)
软件开发(5)
基础技术(2)
蓝牙(6)
ARMv8A Arch(4)
显示(3)
基础学科(9)
技术漫谈(10)
项目专区(0)
X Project(2)

文章存档

2016年4月(7)
2016年3月(5)
2016年2月(5)
2016年1月(6)

三、硬件功能分类

ARM based SOC的datasheet中总有一个章节叫做GPIO controller（或者I/O ports）的章节来描述如何配置、使用SOC的引脚。虽然GPIO controller的硬件描述中充满了大量的寄存器的描述，但是这些寄存器的功能大概分成下面三个类别：

1、有些硬件逻辑是和I/O port本身的功能设定相关的，我们称这个HW block为pin controller。软件通过设定pin controller这个硬件单元的寄存器可以实现：

- （1）引脚功能配置。例如该I/O pin是一个普通的GPIO还是一些特殊功能引脚（例如memory bank上CS信号）。
- （2）引脚特性配置。例如pull-up/down电阻的设定，drive-strength的设定等。

2、如果一组GPIO被配置成SPI，那么这些pin脚被连接到了SPI controller，如果配置成GPIO，那么控制这些引脚的就是GPIO controller。通过访问GPIO controller的寄存器，软件可以：

- （1）配置GPIO的方向
- （2）如果是输出，可以配置high level或者low level
- （3）如果是输入，可以获取GPIO引脚上的电平状态

3、如果一组gpio有中断控制器的功能，虽然控制寄存器在datasheet中的I/O ports章节描述，但是实际上这些GPIO已经被组织成了一个interrupt controller的硬件block，它更像是一个GPIO type的中断控制器，通过访问GPIO type的中断控制器的寄存器，软件可以：

- （1）中断的enable和disable（mask和unmask）
- （2）触发方式
- （3）中断状态清除

四、如何通过软件抽象来掩盖硬件差异

传统的GPIO driver是负责上面三大类的控制，而新的linux kernel中的GPIO subsystem则用三个软件模块来对应上面三类硬件功能：

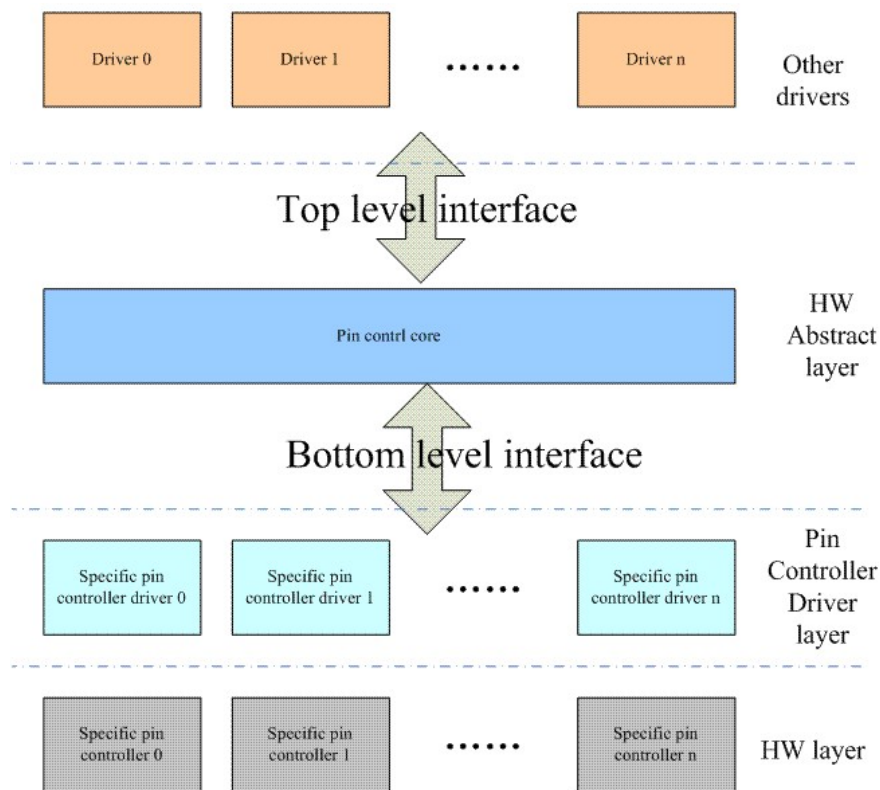
- （1）pin control subsystem。驱动pin controller硬件的软件子系统。
- （2）GPIO subsystem。驱动GPIO controller硬件的软件子系统。
- （3）GPIO interrupt chip driver。这个模块是作为一个interrupt subsystem中的一个底层硬件驱动模块存在的。本文主要描述前两个软件模块，具体GPIO interrupt chip driver以及interrupt subsystem请参考本站其他相关文档。

1、pin control subsystem block diagram

下图描述了pin control subsystem的模块图：

2015年12月(6)
2015年11月(9)
2015年10月(9)
2015年9月(4)
2015年8月(3)
2015年7月(7)
2015年6月(3)
2015年5月(7)
2015年4月(9)
2015年3月(9)
2015年2月(6)
2015年1月(6)
2014年12月(17)
2014年11月(8)
2014年10月(9)
2014年9月(7)
2014年8月(12)
2014年7月(6)
2014年6月(6)
2014年5月(9)
2014年4月(9)
2014年3月(7)
2014年2月(3)
2014年1月(4)

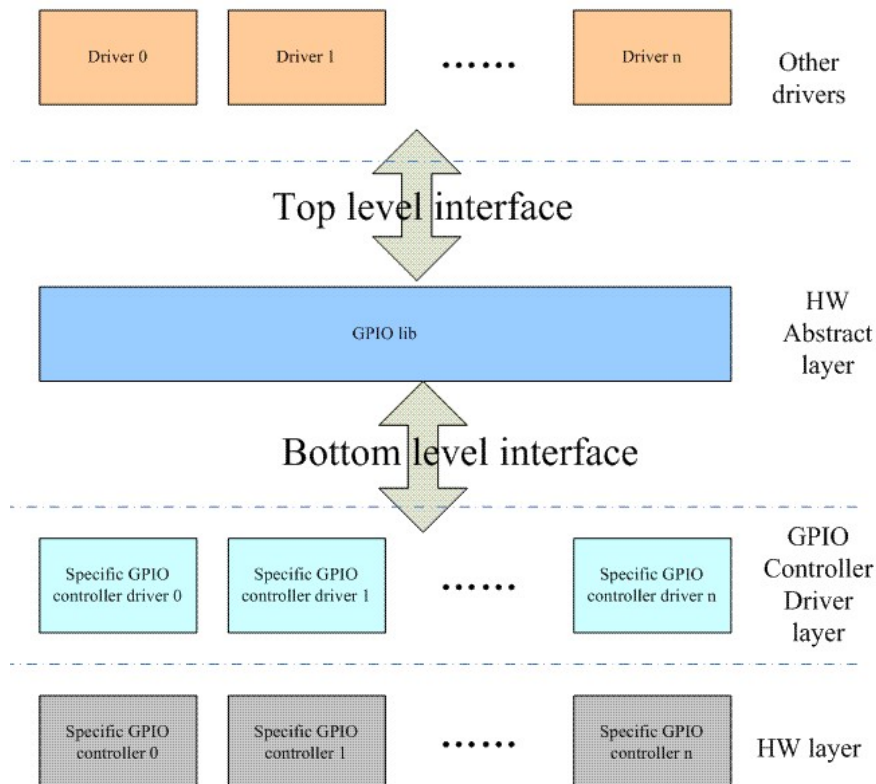




底层的pin controller driver是硬件相关的模组，初始化的时候会向pin control core模块注册pin control设备（通过pinctrl_register这个bootom level interface）。pin control core模块是一个硬件无关模块，它抽象了所有pin controller的硬件特性，仅仅从用户（各个driver就是pin control subsystem的用户）角度给出了top level的接口函数，这样，各个driver不需要关注pin controller的底层硬件相关的内容。

2. GPIO subsystem block diagram

下图描述了GPIO subsystem的模块图：



基本上这个软件框架图和pin control subsystem是一样的，其软件抽象的思想也是一样的，当然其内部具体的实现不一样，我们会在后续的文章中描述。

原创文章，转发请注明出处。蜗窝科技。http://www.wowotech.net/io-port-control.html

标签: GPIO 软件框架

« Linux内核中的GPIO系统之（3）： pin controller driver代码分析 | ACCESS_ONCE宏定义的解释»

评论：

tom

2015-09-22 15:08

pin control subsystem 和 gpio subsystem 有什么区别，

回复

emeralddream

2016-03-31 09:04

@tom：个人理解 PINCTRL更高级吧，涉及到了pimux。目前gpio子系统只保留了API。具体的回调函数已经全部移到pinctrl子系统中去了。

回复

emeralddream

2016-03-31 09:04

@tom：个人理解 PINCTRL更高级吧，涉及到了pimux。目前gpio子系统只保留了API。具体的回调函数已经全部移到pinctrl子系统中去了。

回复

xie1230

2015-08-27 17:31

作者真的很牛，佩服，收藏了

回复

fucker

2015-07-27 09:08

hi linuxer:

在先拜读文章前，先让我先表达下，小弟对你的崇拜~! ! !

```
(_)
/oo\\_____
 \ /          |---\
  V           / \ \
           \\_|__\\_|/ *
             ||  YY|
             ||  ||
```

回复

RRR

2015-06-24 16:54

哎 博主好久没有更新了...GPIO & Pinctrl & IRQ 看了一半就断层了...

好多东西不能联系到一起，好纠结

希望博主有机会能继续完结整个GPIO

回复

wowo

2015-06-25 19:56

@RRR：Linuxer同学这段时间有点忙，相信他很快就会回归的。

回复

SleepDeXiang

2015-06-10 11:15

这个硬件描述的相当完美

回复

linuxer

2015-06-11 00:17

@SleepDeXiang：汗颜！承蒙夸奖，其实这几份GPIO的文档（还有device tree那份）是真的没走心，中断子系统、时间子系统和内核同步的确是“亲生的”，我写的比较用心一些。anyway，欢迎这位欢乐的工程师，哈哈

回复

SleepDeXiang

2015-06-11 10:40

@linuxer: 哈哈，小弟的确平时挺欢乐的！本科毕业马上三年了，以前一直搞驱动(BSP)相关的，今年过来比较闲就想学学更高级的内核相关东西，然后就来到你这！哈哈，真是风水宝地。。。ps：我们部门做SOC的，也是linaro金牌会员，kernel3.8以后的网上根本搜不到资料，我去年年初接触的，第一次在你这看到这么新的知识，真是接地气，特别佩服楼主，万分膜拜！

[回复](#)

车干

2015-06-03 15:27

手上有本老罗的的android的源代码分析。感觉android不比内核简单多少，可能是自己c++，java不行，要是他肯写内核多好。主要用电脑看代码很费眼，而且不如看书能静下心来，自己习惯的问题，只能这样了。

话说今天上午正看着，突然网页怎么也上不了，之前也没存多少html。我还在想完了，难得好的资料看不不了。留个邮箱吧，我的是g5912667@163.com，以防万一。

[回复](#)

蜗蜗

2015-06-03 16:15

@车干: Android涉及到业务逻辑，所以比较繁琐，但是思路是一样的，弄清楚一个点之后，其它的都能触类旁通。

Kernel相比Android，就像理科和工科之间的关系一样，是比较基础的。

上不了是因为博客所在的空间商的IP被攻击。没关系的，我都有备份，wowotech不会不告而别的。

[回复](#)

gzz

2016-01-04 15:52

@车干: android和linux是两种境界，各有味道，研读起来都有一番风味，是C与C++的典范

[回复](#)

车干

2015-06-02 14:00

楼主你好，上次麻烦你出书，你不肯。那有推荐的书没有，毛德操，赵炯两位老师的书内核版本太老了，实际应用没那么高。较新的内核没碰到好点的书，麻烦推荐下像之前两位老师那样差不多的书，

[回复](#)

蜗蜗

2015-06-03 13:41

@车干: 说实话，基于新内核的书还真没有。现在大家比较浮躁，都去写android的了，很少有人研究kernel了啊。

[回复](#)

蜗蜗

2015-06-03 13:42

@车干: 个人意见，最好的书就是kernel source code啊。

[回复](#)

peter

2015-05-12 19:34

学些了，谢谢分享

[回复](#)

foiron

2014-07-21 18:56

hi linux

更新了啊，抓紧来看看。

I: (3) 如果是输入，可以[或者]GPIO引脚上的电平状态 ---这里是要写检测吗？

II: 具体GPIO interrupt chip driverI以及interrupt subsystem请参考本站其他相关文档 这个会在中断子系统里面讲解吗？因为现有的文档没有对这个进行描述

ps: 每天都会打开蜗窝主页，算是对你们的辛勤劳动做一点贡献吧：)

[回复](#)

linuxer

2014-07-21 21:00

@foiron: I: 多谢指正，已经修改了笔记

II: 其实是准备写一个Linux kernel的中断子系统的系列文章，不过现在还没有动笔（GIC的代码分析应该是中断子系统中的一篇，后续还会写一个GPIO type的interrupt controller的驱动，其实这两种irq chip的概念是类似的），呵呵 ~ ~ 完成之后再完整本文中的链接。

[回复](#)

发表评论：

昵称

邮件地址 (选填)

个人主页 (选填)

PMugN

发表评论