

CSDN新首页上线啦，邀请你来立即体验！(http://feed.csdn.net/)

**CSDN**

博客 (http://feed.csdn.net/?ref=toolbar)

学院 (http://edu.csdn.net?ref=toolbar)

下载 (http://download.csdn.net?ref=toolbar)

更多 ▾



登录 (https://passport.csdn.net/account/login?ref=toolbar) 注册 (http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobile\_register\_source=csdnblog1)

1



## 高通 msm平台GPIO相关的device tree设置

转载

2016年11月30日 00:02:21



标签：高通 (http://so.csdn.net/so/search/s.do?q=高通&t=blog) /



c语言 (http://so.csdn.net/so/search/s.do?q=c语言&t=blog) / dts (http://so.csdn.net/so/search/s.do?q=dts&t=blog) /



gpio (http://so.csdn.net/so/search/s.do?q=gpio&t=blog) / 中断 (http://so.csdn.net/so/search/s.do?q=中断&t=blog)

2721

转载自：http://blog.csdn.net/hongzg1982/article/details/47784627

(http://blog.csdn.net/hongzg1982/article/details/47784627)

GPIO相关的dvice tree设置和interrupt设置

gpoi号以及gpio相关的属性设置

以tsp的proxy\_en端口为例：

立即体验



viewsky11

+ 关注

美国产子



(http://blog.csdn.net/viewsky11)

码云

未开通

原创

粉丝

喜欢

(https://github.com/viewsky11)

227

54

0

### 他的最新文章

更多文章 (http://blog.csdn.net/viewsky11)

pbc应用分析

(/viewsky11/article/details/78144620)

mqtt C客户端

(/viewsky11/article/details/78105405)

linux内核代码特点

(/viewsky11/article/details/77966762)

Linux open系统调用流程

(/viewsky11/article/details/77934608)



内容举报



返回顶部

```

1 i2c@78b6000 { /* BLSP1 QUP2 */
2     compatible = "qcom,i2c-msm-v2";
3     ...
4     tmd3782@39 {
5         compatible = "taos,tmd3782";
6         ...
7         taos,en = <msm_gpio 8 0x1>; //
8         ...
9     }
10 }

```

在相应的驱动里边，取gpio编号以及设置输入或者输出

```

1 //在tsp驱动里边，可以通过of_get_named_gpio()来取相应的gpio号
2 pdata->enable = of_get_named_gpio(np, "taos,en", 0);
3 gpio_direction_output(pdata->enable, 1);

```

那这个gpio的active的时候和sleep的时候的 PULL\_DOWN , PULL\_UP , NO\_PULL 等属性在哪里设置呢？  
其实是在pinctrl相关的dtsi文件里边设置的



## 博主专栏



rt2860v2源码分析

(<http://blog.csdn.net/column>)

3343

(<http://blog.csdn.net/column/details/1>)

多线程编程笔记



(<http://blog.csdn.net/column>)

7332

(<http://blog.csdn.net/column/details/1>)

linux设备驱动分析



( kernel-4.7 )

(<http://blog.csdn.net/column>)

展开



内容举报

## 在线课程



返回顶部



1



```

1 tlmux_pinmux: pinctrl@1000000 {
2     compatible = "qcom,msm-tlmux-8916";//看一下相应的驱动
3     reg = <0x1000000 0x300000>;
4     interrupts = <0 208 0>;
5
6     //gp: general purpose pins
7     //此外还有两种pin type:
8         //sdc : SDC pins
9         //qds: QDSC pins
10    /*General purpose pins*/
11    gp: gp {
12        qcom,pin-type-gp;
13        qcom,num-pins = <122>; //这个pin type里包含的pin的个数
14        #qcom,pin-cells = <1>;
15        msm_gpio: msm_gpio {
16            compatible = "qcom,msm-tlmux-gp";
17            gpio-controller;//指定当前msm_gpio为一个gpio-controller
18            #gpio-cells = <2>;
19            // #gpio-cells的值指定每个msm_gpio后面跟几个数来表示一个gpio。
20            //如果是2的话，就要像上面tsp一样写成taos,en = <&msm_gpio 8 0x1>;
21            //前面的8指定gpio号，但后面的还不是很清楚，应该是表示输入输出等，这里0x1表示输出
22            interrupt-controller; //表示可以用作中断控制器
23            #interrupt-cells = <2>;//两个的话，前面的表示irq号，后面的一个optional flags
24            num_irqs = <122>;//表示可以被用作中断源的pin的个数
25        };
26    };
27 };
28
29 //定义sdc pin type
30 /* Sdc pin type */
31 sdc: sdc {
32     qcom,num-pins = <6>;
33     #qcom,pin-cells = <1>;

```



SDCC 2017  
前端技术实战线上峰会



C语言大型软件设计的面向对象  
utm\_source=blog9  
(http://blog.csdn.net/hu  
Course/detail/603?  
log9)



美国产子

## 热门文章

Openwrt/lib/functions/procd.sh脚本 (/viewsky11/article/details/45972553)

4144

Openwrt的wifi脚本分析 (/viewsky11/article/details/44998853)

4093

基本线程编程(多线程编程笔记) (/viewsky11/article/details/53419301)

4014

OpenWrt路由器通过Aria2开启迅雷离线下载[图文教程] (/viewsky11/article/details/19551965)

3754

openwrt升级命令 (/viewsky11/article/details/17148257)

3559



内容举报



返回顶部



1



```

34 };
35
36 prox_sensor_power {
37     qcom,pins = <&gp 8>; //gpio 8的上拉，下拉或者no pull等属性
38     qcom,pin-func = <0>;
39     qcom,num-grp-pins = <1>;
40     //qcom,num-grp-pins，这个表示一共有几个pin
41     //msm-pinctrl.txt里边写的是number of pins in the group.
42
43     label = "prox_sensor_power";
44     //label: name to identify the pin group to be used by a client.
45
46     //以下是pinctrl的时候的设置，下面有pinctrl的说明
47     prox_power_active: prox_power_active {
48         drive-strength = <2>;//2MA
49         bias-disable; /* No PULL *///也可以写成 bias-disable = <0>;
50         //这里可以把bias-disable替换成bias-pull-up;或者bias-pull-down;
51     };
52
53     prox_power_suspend: prox_power_suspend {
54         drive-strength = <2>; //2MA
55         bias-disable; /* No PULL */
56     };
57 };

```

在dts里边定义gpio相关的中断

以hall id相关的dtsi定义为例



内容举报



返回顶部



1



```

1 hall {
2     status = "okay";
3     compatible = "hall";
4     interrupt-parent = <&msm_gpio>;//表示当前的中断控制器用的哪个，高通平台的dts文件中，
5         //msm_gpio表示msm_tlmm_irq中断控制器。
6     interrupts = <52 0>;
7     //第一个52是指中断号，后面的0不知道表示什么，根据文档，这个值有以下意义
8     //1: low-to-high edge triggered
9     //2: high-to-low edge triggered
10    //3: active high-level-sensitive
11    //4: active low-level-sensitive
12    //Documentation/devicetree/bindings/arm/gic.txt
13
14    hall,gpio_flip_cover = <&msm_gpio 52 0>;
15
16
17 };

```

如果定义了一个以上的interrupts号该怎么弄呢？

比如：

```

1 hall {
2     ...
3     interrupts = <52 0> <62 0>;
4     ...
5 }

```

这种可以通过， platform\_get\_irq(pdev,0) 来取第一个irq号， platform\_get\_irq(pdev,1) 来取第二个irq号。

## reg相关的设置



内容举报



返回顶部

以下是 reg , address-cells , size-cells 的解释 , 但还不知道从哪里读出来这些并设置??

- reg
- #address-cells
- #size-cells

其中reg的组织形式为



reg = <address1 length1 [address2 length2] [address3 length3] ... > ,

- 1 其中的每一组address length表明了设备使用的一个地址范围。address为1个或多个32位的



整型 ( 即cell ) , 而length则为cell的列表或者为空 ( 若 \#size-cells = 0 ) 。

address 和 length 字段是可变长的, 父结点的 #address-cells 和 #size-cells 分别决定了子结点的



reg属性的address和length字段的长度。

在本例中, root结点的 #address-cells = <1>; 和 #size-cells = <1>; 决定了serial、gpio、spi等结点的address和



length字段的长度分别为1。

cpus 结点的 #address-cells = <1>; 和 #size-cells = <0>; 决定了2个cpu子结



点的address为1, 而length为空, 于是形成了2个cpu的 reg = <0>; 和 reg = <1>; 。

external-bus结点的 #address-cells = <2> 和 #size-cells = <1>; 决定了其下的

ethernet、i2c、flash的reg字段形如 reg = <0 0 0x1000>; 、 reg = <1 0 0x1000>;

和 reg = <2 0 0x4000000>; 。

其中, address字段长度为0, 开始的第一个 cell ( 0、1、2 ) 是对应的片选, 第2个 cell ( 0, 0, 0 ) 是相对该片选的基地址, 第3个 cell ( 0x1000、0x1000、0x4000000 ) 为length。

特别要留意的是i2c结点中定义的 #address-cells = <1>; 和 #size-cells = <0>;

## pinctrl相关的设置

pinctrl相关的的设置到底有什么用呢? 在驱动里边常常碰到驱动相关的一个或者几个gpio, 在醒来或者睡眠的时候需要设置成不同的类型, 不如醒来的时候是i2c端口, 但睡眠的时候可能要设置成GPIO并把输出设置成0等。



内容举报

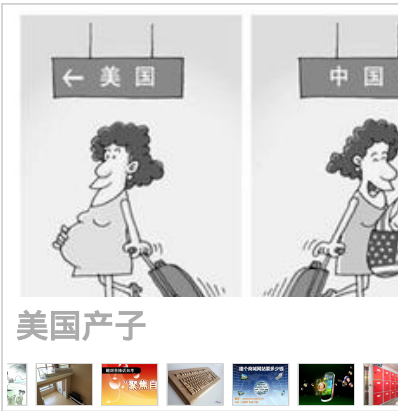


返回顶部

这个时候pinctrl就派上用场了，这个可以大大简化驱动的编写，因为这个可以像下面这样根据active和suspend来设置要配置的gpio的管脚配置，然后在驱动里边调用

```
1 devm_pinctrl_get_select(dev,"tlmm_motor_active");
```

来实现 pinctrl-0 和 pinctrl-1 里对应 tlmm\_motor\_active 和 tlmm\_motor\_suspend 的配置！！像下面的device tree配置的话，如果是 devm\_pinctrl\_get\_select(dev,"tlmm\_motor\_active"); 则应该就是把 pinctrl-0 里边的gpio相关配置都配置上去。如果是 devm\_pinctrl\_get\_select(dev,"tlmm\_motor\_suspend"); 的话就把 pinctrl-1 里边的设置都配置上去。  
具体 devm\_pinctrl\_xxx 这种接口说明也可以参考kernel下面的Documentation/pintrl.txt文件





1



```

1  &soc {
2    xxx,vibrator {
3      compatible = "haptic_vib";
4      //下面的pinctrl的设置，好像是没有在驱动里边读取并进行设置，
5      //不知道什么用~~
6      pinctrl-names = "tlmm_motor_active","tlmm_motor_suspend";
7      pinctrl-0 = <&motor_en_active &motor_pwm_active>;
8      pinctrl-1 = <&motor_en_suspend &motor_pwm_suspend>;
9
10     //下面en,pwm的设置前面已经讲过
11     xxx,vib_en = <&msm_gpio 76 0x1>;
12     xxx,vib_pwm = <&msm_gpio 50 0x1>;
13
14     xxx,vib_model = <1>;
15     xxx,is_pmic_vib_pwm = <0>;
16     xxx,pwm_period_us = <40>;
17     xxx,duty_us = <36>;
18     status = "ok";
19   };
20
21
22   //这里muic_i2c_active , muic_int_pin ,muic_chg_det这种都可以在gpio controller那里找到相应
23   //的设置，例如下面这样
24   tlmm_motor_en {
25     qcom,pins = <&gp 76>;
26     qcom,pin-func = <0>;
27     qcom,num-grp-pins = <1>;
28     label = "tlmm_motor_en";
29     motor_en_active: motor_en_active {
30       drive-strength = <2>;
31       bias-disable = <0>; /* No PULL */
32     };
33     motor_en_suspend: motor_en_suspend {

```



内容举报



返回顶部



```
34         drive-strength = <2>;
35         bias-disable = <0>; /* No PULL */
36     };
37 }
```

在dts里边定义gpio和中断的文档



- 1 还有像gpio-ranges这种没有说明，可以再Documentation/devicetree/bindings/gpio/gpio.txt里边找到说明。
- 还有一个是高通的gpio说明，在Documentation/devicetree/bindings/pinctrl/msm-pinctrl.txt。



## 🔖 在gpio和中断debug方法

💬 在debug目录下，可以查到每个gpio的输入输出设置，以及当前的值。

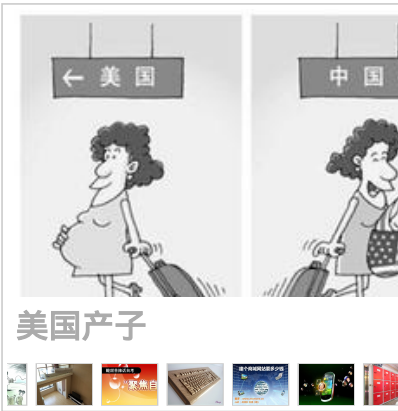


```
1 #cat /d/gpio
2 //这个命令只会显示AP设置的GPIO信息，不显示Modem设置的GPIO信息
```

如果想看更详细的GPIO设置的话

```
1 #cat /d/gpiomux
2 //显示AP,CP所有的GPIO的信息
```

例：





1



```
1 //开始操作GPIO的时候必须要先执行
2 #echo 30 > /sys/class/gpio/export
3
4 //设置GPIO 30的输入输出
5 #echo "out" > /sys/class/gpio/gpio30/direction
6 #echo "in" > /sys/class/gpio/gpio30/direction
7
8 //改变GPIO 30的值
9 #echo 1 > /sys/class/gpio/gpio30/value
10 #echo 0 > /sys/class/gpio/gpio30/value
11
12 //操作完毕需要执行如下命令
13 #echo 30 > /sys/class/gpio/unexport
14
```

查找Wakeup IRQ等

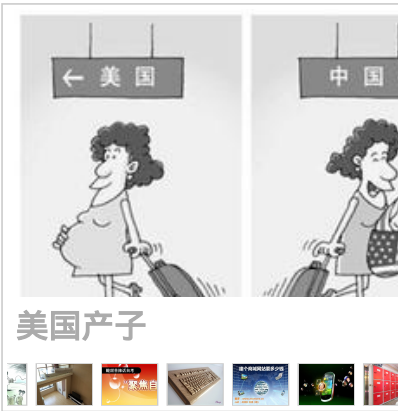


```
1 #echo 1 > /sys/module/msm_show_resume_irq/parameters/debug_mask.
2 //这样输入完之后，如果被中断唤醒就会输出如下log
3 [ 75.0xxx] pm8xxx_show_resume_irq_chip: 479 triggered
4 [ 75.0xxx] msm_gpio_show_resume_irq: 392 triggered
5 [ 75.0xxx] gic_show_resume_irq: 48 triggered
6 [ 75.0xxx] gic_show_resume_irq: 52 triggered
7
```

显示整个中断设置情况

```
1 #cat /proc/interrupts
```

# GIC 中断控制器



  
内容举报

  
返回顶部

GIC中断控制器的device tree定义的例子如下：

```

1  intc:interrupt-controller@F9000000{
2      compatible = "qcom,msm-qgic2";
3      interrupt-controller;//声明这个为一个中断控制器
4      #interrupt-cells = <3>;//高通的这里是3，具体看芯片的GIC中断控制器
5      reg = <0xF9000000 0x1000>, <0xF900200 0x1000>;
6  }
7

```

这里 interrupt-controller 的意思跟上面解释的一样，但 #interrupt-cells 的值必须是3。

高通的是这样的，当然其他芯片的还要具体看GIC控制器驱动。

申请中断的例子：

```

1  device1@f991f000{
2      compatible = "qcom,msm-device-v1";
3      reg = <0xf991f000 0x1000>;
4      interrupt-parent = <&intc>; //指定中断控制器
5      interrupts = <0 131 0>, <0 179 0>;
6      interrupt-names = "irq", "otg_irq";
7  };
8

```

这里interrupts的3个数中，后面两个和前面说的一样的，分别是中断号和中断类型。

第三个数指定的中断类型：

```

1  //1: low-to-high edge triggered
2  //2: high-to-low edge triggered
3  //3: active high-level-sensitive
4  //4: active low-level-sensitive
5  //Documentation/devicetree/bindings/arm/gic.txt
6

```



⚠  
内容举报

⬆  
返回顶部

那第一个数表示什么呢？第一个数表示中断GIC的中断类型。

0 表示：shared processor interrupts (SPI)

1 表示：Private Pripheral Interrupts (PPI)

还有interrupt mapping内容看一下Linux Device Tree GPIO文档。



1

以下是打印的某个高通平台的/proc/interrupts的内容

☰ GIC的中断号有些和 msmxxx.dtsi 里边设置的终端号不一致，一般有一定的偏移量。之前看到过的是32，还没找在哪里设置的。



内容举报



返回顶部



1



```

1 root@gtelltevw:/proc # cat interrupts
2 cat interrupts
3      CPU0   CPU1   CPU2   CPU3
4 20: 11501064 2634910 1450801 1172471   GIC arch_timer
5 35:    0     0     0     0   GIC apps_wdog_bark
6 39: 5643824 2599019 1701936 1316131   GIC arch_mem_timer
7 47:  52981     0     0     0   GIC cpr
8 56:    0     0     0     0   GIC modem
9 57: 1527948     0     0     0   GIC qcom,smd-modem
10 58:    5     0     0     0   GIC qcom,smsm-modem
11 59:    5     0     0     0   GIC smp2p
12 61:   10     0     0     0   GIC sps
13 65: 23838     0     0     0   GIC kgs1-3d0
14 75:    0     0     0     0   GIC msm_iommu_global_cfg_irq, msm_iommu_global_cfg_irq
15 76:  420     0     0     0   GIC msm_vidc
16 82:   10     0     0     0   GIC cci
17 83:    2     0     0     0   GIC csid
18 84:    2     0     0     0   GIC csid
19 89:    2     0     0     0   GIC
20 102:    0     0     0     0   GIC msm_iommu_nonsecure_irq, msm_iommu_nonsecure_irq, msm_iommu_nonsec
21 104: 650484     0     0     0   GIC MDSS
22 110:    0     0     0     0   GIC csiphy
23 111:    0     0     0     0   GIC csiphy
24 127:    0     0     0     0   GIC i2c-msm-v2-irq
25 128:  7585     0     0     0   GIC i2c-msm-v2-irq
26 130:    0     0     0     0   GIC i2c-msm-v2-irq
27 131:    0     0     0     0   GIC i2c-msm-v2-irq
28 132:    0     0     0     0   GIC i2c-msm-v2-irq
29 140: 384648     0     0     0   GIC msm_serial_hsl0
30 155: 12681898     0     0     0   GIC mmc0
31 157:    0     0     0     0   GIC mmc1
32 166:   1761     0     0     0   GIC msm_otg, msm_hsusb
33 170: 263953     0     0     0   GIC 7824900.sdhci

```



内容举报



返回顶部



1



```

34 172: 0 0 0 0 GIC msm_otg
35 174: 207 0 0 0 GIC qcom,smd-wcnss
36 175: 5 0 0 0 GIC smp2p
37 176: 0 0 0 0 GIC qcom,smsm-wcnss
38 181: 0 0 0 0 GIC wcnss
39 200: 8461599 379482 175887 113247 GIC qcom,smd-rpm
40 203: 932778 390645 274255 170499 GIC 601d0.qcom,mpm
41 216: 0 0 0 0 GIC tsens_interrupt
42 222: 7 0 0 0 GIC 200f000.qcom,spmi
43 239: 0 0 0 0 GIC sps
44 240: 946 0 0 0 GIC 1000000.pinctrl
45 253: 2 0 0 0 GIC 7864900.sdhci
46 273: 0 0 0 0 GIC msm_iommu_nonsecure_irq
47 274: 0 0 0 0 GIC msm_iommu_nonsecure_irq
48 280: 1 0 0 0 GIC mobicore
49 288: 3 0 0 0 msm_tlmm_irq sm5703
50 290: 0 0 0 0 msm_tlmm_irq 7864900.sdhci cd
51 291: 6 0 0 0 qnpn-int qnpn_kdpwr_status
52 292: 0 0 0 0 qnpn-int qnpn_resin_status
53 294: 0 0 0 0 qnpn-int qnpn_kdpwr_resin_bark
54 295: 0 0 0 0 qnpn-int qnpn_rtc_alarm
55 297: 0 0 0 0 qnpn-int pm8916_tz
56 299: 1 0 0 0 qnpn-int qnpn_adc_tm_high_interrupt
57 300: 0 0 0 0 qnpn-int qnpn_adc_tm_low_interrupt
58 330: 0 0 0 0 msm_tlmm_irq k2hh_accel
59 338: 0 0 0 0 sm5703 otffail
60 348: 3 0 0 0 sm5703 topoff
61 349: 0 0 0 0 sm5703 done
62 357: 5 0 0 0 msm_tlmm_irq sm5703 muic micro USB
63 454: 932 2 1 1 msm_tlmm_irq zt7554_ts
64 455: 0 0 0 0 msm_tlmm_irq fuelgauge-irq
65 456: 0 0 0 0 msm_tlmm_irq sx9500_irq
66 457: 0 0 0 0 msm_tlmm_irq sx9500_wifi_irq
67 458: 0 0 0 0 smp2p_gpio modem

```



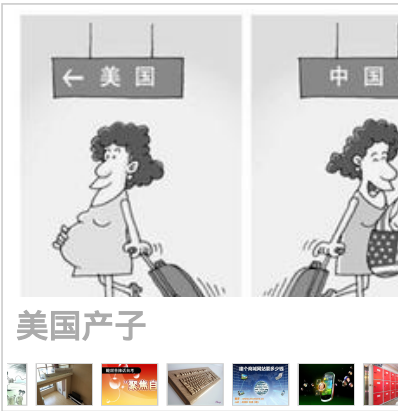
内容举报



返回顶部

- 👍
- 1
- ☰
- 🔖
- 💬
- 🔗

68	459:	1	0	0	0	smp2p_gpio	error_ready_interrupt
69	460:	1	0	0	0	smp2p_gpio	modem
70	461:	0	0	0	0	smp2p_gpio	modem
71	490:	0	0	0	0	smp2p_gpio	wcnss
72	491:	1	0	0	0	smp2p_gpio	error_ready_interrupt
73	492:	1	0	0	0	smp2p_gpio	wcnss
74	493:	0	0	0	0	smp2p_gpio	wcnss
75	522:	2	0	0	0	msm_tlmm_irq	home_key
76	523:	0	0	0	0	msm_tlmm_irq	volume_up
77	524:	0	0	0	0	msm_tlmm_irq	sec_headset_detect
78	IPI0:	0	49521	49521	49521	CPU	wakeup interrupts
79	IPI1:	263118	216085	322849	349358	Timer	broadcast interrupts
80	IPI2:	5221229	10110805	7696353	5534579	Rescheduling	interrupts
81	IPI3:	585272	2348936	2593715	2633820	Function call	interrupts
82	IPI4:	2127	403855	275707	237116	Single function call	interrupts
83	IPI5:	0	0	0	0	CPU stop	interrupts
84	IPI6:	0	0	0	0	CPU	backtrace
85	Err:	0					





相关文章推荐

- ⚠️  
内容举报
- ⬆️  
返回顶部

## powersave debug (/haichunzhao/article/details/11629269)

高通8960平台 1.判断手机是否进入睡眠，比较直观的方法是连接电源，按下电源键，察看电源上显示的电流。手机睡眠后，电流为2mA上下范围。 2.如果手机不睡眠，可以分为几个步骤去调查。 首先看一下正常...

 haichunzhao (<http://blog.csdn.net/haichunzhao>) 2013-09-13 09:49  3109



## 配置学习总结高通平台GPIO (/joymine/article/details/48161895)

1

前言：近期在BSP的工作学习中，Group Leader 给安排了一项任务，就是分别在Android项目代码里面的SBL1，LK，以及各自负责的驱动模块代码（我是Audio驱动）中间配置GPIO ...

 Joymine (<http://blog.csdn.net/Joymine>) 2015-09-01 18:59  2549





## 如何从零学习 React 技术栈

在学会 React 之后，你的能力将不止局限于浏览器，React 还可以拓宽到使用 React Native 开发原生应用，以及使用 ReactVR 开发虚拟现实等各个领域。

([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjcvPWm0lZ0qnfK9ujYzP1fvn1Dz0Aw-5Hc4nHb3rjD0TAq15HfLPWRznjb0T1Ydrju9PyczP1w9mvfdP16z0AwY5HDdnH01rj0knHc0IgF\\_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq\\_lyd8Q1R4myRknhubmhr9rHNhrHmkPWqdIAdxTvqdThP-5HDknHKBmhkEusKzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqn1DvPW0](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjcvPWm0lZ0qnfK9ujYzP1fvn1Dz0Aw-5Hc4nHb3rjD0TAq15HfLPWRznjb0T1Ydrju9PyczP1w9mvfdP16z0AwY5HDdnH01rj0knHc0IgF_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_lyd8Q1R4myRknhubmhr9rHNhrHmkPWqdIAdxTvqdThP-5HDknHKBmhkEusKzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqn1DvPW0))

## MSM8937平台bootloader调试之一 (/js\_wawayu/article/details/52419318)

MSM8937平台的AP是八核A53的处理器，分成两个cluster，最高主频支持1.4GHZ，PMIC有两种组合：PM8937+PMI8937和PM8937+PMI8952。 高通...

 js\_wawayu ([http://blog.csdn.net/js\\_wawayu](http://blog.csdn.net/js_wawayu)) 2016-09-02 21:49  3408



内容举报



返回顶部

## msm8937 64位开发概要 (/loongembedded/article/details/72674115)



1. 8937 lk采用8952的。/device/qcom/msm8937\_64/BoardConfig.mk下面可看出来BOOTLOADER\_PLATFORM := msm8952  
# use m...



LoongEmbedded (<http://blog.csdn.net/LoongEmbedded>)

2017-05-24 10:47

📖1490



## Webpack 前端工程化入门

如果你想了解在公司级别的大型项目中是如何做工程化的？如果你想深入了解 Webpack，并且想用它的各种特性来提升构建效率？那么你就有理由来了解一下这门课。



美国产子



([http://www.baidu.com/cb.php?c=lgF\\_pyfqHmknjcvPjR0IZ0qnfK9ujYzP1ndnHfY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YvrHmsPj9hrycsm1m3rjnd0AwY5HDdnH01rj0knHc0lgF\\_5y9YIZ0IQzqMpgwBUvqoQhP8QvIGIAPCmgfLmvgQydeQ2Ktandovjmv7AME](http://www.baidu.com/cb.php?c=lgF_pyfqHmknjcvPjR0IZ0qnfK9ujYzP1ndnHfY0Aw-5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YvrHmsPj9hrycsm1m3rjnd0AwY5HDdnH01rj0knHc0lgF_5y9YIZ0IQzqMpgwBUvqoQhP8QvIGIAPCmgfLmvgQydeQ2Ktandovjmv7AME))



RduHfLrHcsPjqdIAdxTvqdThP-

5HDknHK3mhkEusKzujYk0AFV5H00TZcq0KdpyfqHRLPjnvnfKEpyfqHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnHckP1b)



## GTxxxx系列-----如何读取某个gpio的当前的状态 (/u013308744/article/details/50854145)



有时候需要去读取一个gpio的状态是啥，通常在system/bin目录下有个r文件 如果怀疑设置不对，就可以读取一下看看 root

@msm8937\_evb:/system/bin # ./r...



u013308744 (<http://blog.csdn.net/u013308744>)

2016-03-11 16:35

📖957

## msm8937 porting iic device (/u013308744/article/details/50371627)

又要加touch了，重新搞一遍porting工作，还是遇到了一些问题。 首先往Dtsi文件里加iic device的时候要注意看iic bus是否已经挂载成功了 我就遇到了用i2cdetect扫描bu...



u013308744 (<http://blog.csdn.net/u013308744>)

2015-12-21 14:08

📖1945

## msm平台GPIO相关的device tree设置 (/keleming1/article/details/51034946)

转<http://www.ithao123.cn/content-8794703.html> msm平台GPIO相关的device tree设置 [摘要：GPIO相干的dvce tree...



内容举报



返回顶部

keleming1 (<http://blog.csdn.net/keleming1>) 2016-04-01 10:38 2451

## 高通GPIO配置简介 (/ldinvincible/article/details/52421640)

GPIO中断，调试和客制化 PA2 GPIO/Interrupt non-HLOS 覆盖 non-HLOS GPIO/Interrupt 软件驱动，GPIO是标准通用输入输出。Msm与外设之间最基本...

ldinvincible (<http://blog.csdn.net/ldinvincible>) 2016-09-03 10:25 3537

## msm GPIO 配置 (/u013786805/article/details/30266589)

示例：gpio\_tlmm\_config(GPIO\_CFG(PMIC\_GPIO\_INT, 5, GPIO\_CFG\_INPUT, GPIO\_CFG\_NO\_PULL, GPIO\_CFG\_2MA...

u013786805 (<http://blog.csdn.net/u013786805>) 2014-06-12 17:18 1814

## 高通msm8916 gpio笔记（基于设备树） (/hm131415/article/details/54377549)

1.通用GPIO控制函数：gpio\_set\_value (port\_num,0/1) 一般只是在这个GPIO口的寄存器上写上某个值，至于这个端口是否设置为输出，它就管不了！而gpio\_directio...

no1freedman (<http://blog.csdn.net/no1freedman>) 2017-01-16 11:42 1173

## msm GPIO setting 及debug (/a04081122/article/details/54666941)

1.MSM BLSP ownership设置 高通的低速接口I2C&SPI&UART通过BLSP控制器来控制。而BLSP控制器的ownership可以设置给不同的模块，比如HLOS,ADSP,TZ等。 ...

a04081122 (<http://blog.csdn.net/a04081122>) 2017-01-22 11:25 822

## 高通使用/system/bin/r读取msm8916的gpio配置 (/eliot\_shao/article/details/53836216)




内容举报




返回顶部

高通使用/system/bin/r读取msm8916的gpio配置 作者：Eliot shao 2016年12月22日 使用文档：80-NK807-2X msm8916 so f...


 seek\_0380 ([http://blog.csdn.net/seek\\_0380](http://blog.csdn.net/seek_0380)) 2016-12-23 10:33 1523


## 高通平台中gpio简单操作和调试 (/s\_jason/article/details/73864103)

 做底层驱动免不了gpio打交道，所以对其操作和调试进行了一下简单的梳理 一、gpio的调试方法 在Linux下，通过sysfs，获取gpio状态，也可以操作gpio。 1、获取...

 s\_jason ([http://blog.csdn.net/s\\_jason](http://blog.csdn.net/s_jason)) 2017-06-28 21:21 850

## Device Tree GPIO引脚控制 (/zhuxiaoping54532/article/details/50393828)

 第一次接触Devicetree，在调试IC驱动的时候居然发现Reset引脚一直无法控制，驱动调试被Block了，后来终于摸索出一个步骤。这个是在高通MSM8976平台上控制GPIO的步骤： 1、从o...

 zhuxiaoping54532 (<http://blog.csdn.net/zhuxiaoping54532>) 2015-12-24 13:41 2487


## linux驱动开发—基于Device tree机制的驱动编写 (/zengxianyang/article/details/50732929)

前言 Device Tree是一种用来描述硬件的数据结构，类似板级描述语言，起源于OpenFirmware(OF)。在目前广泛使用的Linux kernel 2.6.x版本中，对于不同平台、不同硬件...

 zengxianyang (<http://blog.csdn.net/zengxianyang>) 2016-02-24 20:15 4310

## 高通 lcd dtsti (/qq\_20678703/article/details/51210790)

<http://blog.csdn.net/loongembedded/article/details/50972188> 4.1. LCD 参考文档： 1) 80-NA157-174\_...

 qq\_20678703 ([http://blog.csdn.net/qq\\_20678703](http://blog.csdn.net/qq_20678703)) 2016-04-21 17:08 1439




内容举报



返回顶部

## 高通 linux 3.0 配置及控制GPIO的方式 (/yoursuperhero/article/details/52932224)

linux配置GPIO的方式 linux配置GPIO的方式 Kernel阶段的控制 1 使用Pinctrl框架控制 (1) 设备树文件相关节点 sn7326\_po...

 YOURSUPERHERO (<http://blog.csdn.net/YOURSUPERHERO>) 2016-10-26 10:52 908


## 高通方案gpio (/liwei405499/article/details/42008723)


1

gpio 1.1.1 GPIO的作用 GPIO 作为QSC60X5 与外界沟通的桥梁，主要有四个作用： 1 . 通过GPIO 向外部输出一个高/低电平，控制外部的器件或者通知外部器件某...

 liwei405499 (<http://blog.csdn.net/liwei405499>) 2014-12-18 18:23 1557

## 给定A, B两个整数，不使用除法和取模运算，求A/B的商和余数 (/peace\_power/article/details/...

 给定A, B两个整数，不使用除法和取模运算，求A/B的商和余数。 1 . 最基本的算法是，从小到大遍历： for (i = 2 to A -1)  
if (i \* B >...

 peace\_power ([http://blog.csdn.net/peace\\_power](http://blog.csdn.net/peace_power)) 2013-07-01 00:25 2666

## 利用K-means聚类算法根据经纬度坐标对中国省市进行聚类 (/llg\_llg\_llg/article/details/531928...

K-means聚类算法是一种非层次聚类算法，在最小误差的基础上将数据划分了特定的类，类间利用距离作为相似度指标，两个向量之间的距离越小，其相似度就越高。程序读取全国省市经纬度坐标，然后根据经纬度坐标进...

 LLG\_llg\_llg ([http://blog.csdn.net/LLG\\_llg\\_llg](http://blog.csdn.net/LLG_llg_llg)) 2016-11-16 22:58 5148



内容举报



返回顶部