# Linux驱动\_pinctrl子系统

## 1.pinctrl子系统初始化

### 1.1 dts树中关于pinctrl的定义

pinctrl@56000000 {

compatible = "samsung,s3c2440-pinctrl";

};

pinctrl\_0: pinctrl@56000000 {

reg = <0x56000000 0x1000>;

wakeup-interrupt-controller {

compatible = "samsung,s3c2410-wakeup-eint";

interrupts = <0 0 0 3>,

<0 0 1 3>,

<0 0 2 3>,

<0 0 3 3>,

<0 0 4 4>,

<0 0 5 4>;

};

};

&pinctrl\_0 {

/\*

\* Pin banks

\*/

gpa: gpa {

gpio-controller;

#gpio-cells = <2>;

};

gpb: gpb {

gpio-controller;

#gpio-cells = <2>;

};

gpc: gpc {

gpio-controller;

#gpio-cells = <2>;

};

gpd: gpd {

gpio-controller;

#gpio-cells = <2>;

};

gpe: gpe {

gpio-controller;

#gpio-cells = <2>;

};

gpf: gpf {

gpio-controller;

#gpio-cells = <2>;

interrupt-controller;

#interrupt-cells = <2>;

};

gpg: gpg {

gpio-controller;

#gpio-cells = <2>;

interrupt-controller;

#interrupt-cells = <2>;

};

gph: gph {

gpio-controller;

#gpio-cells = <2>;

};

gpj: gpj {

gpio-controller;

#gpio-cells = <2>;

};

/\*

\* Pin groups

\*/

uart0\_data: uart0-data {

samsung,pins = "gph-0", "gph-1";

samsung,pin-function = <2>;

};

i2c0\_bus: i2c0-bus {

samsung,pins = "gpe-14", "gpe-15";

samsung,pin-function = <EXYNOS\_PIN\_FUNC\_2>;

};

nand\_pinctrl: nand\_pinctrl {

samsung,pins = "gpa-17", "gpa-18", "gpa-19",

"gpa-20", "gpa-22";

samsung,pin-function = <1>;

};

};

### 1.2 2440 pinctrl子系统初始化

samsung,s3c2440-pinctrl的注册代码如下：

static const struct of\_device\_id samsung\_pinctrl\_dt\_match**[]** **=** **{**

**{** **.**compatible **=** "samsung,s3c2440-pinctrl"**,**

**.**data **=** s3c2440\_pin\_ctrl **},**

**};**

static struct platform\_driver samsung\_pinctrl\_driver **=** **{**

**.**probe **=** samsung\_pinctrl\_probe**,**

**.**driver **=** **{**

**.**name **=** "samsung-pinctrl"**,**

**.**of\_match\_table **=** samsung\_pinctrl\_dt\_match**,**

**.**suppress\_bind\_attrs **=** **true,**

**},**

**};**

static int \_\_init samsung\_pinctrl\_drv\_register**(**void**)**

**{**

register\_syscore\_ops**(&**samsung\_pinctrl\_syscore\_ops**);**

**return** platform\_driver\_register**(&**samsung\_pinctrl\_driver**);**

**}**

匹配后samsung\_pinctrl\_probe函数调用。

static int samsung\_pinctrl\_probe**(**struct platform\_device **\***pdev**)**

**{**

struct samsung\_pinctrl\_drv\_data **\***drvdata**;**

const struct samsung\_pin\_ctrl **\***ctrl**;**

struct device **\***dev **=** **&**pdev**->**dev**;**

struct resource **\***res**;**

int ret**;**

//分配samsung\_pinctrl\_drv\_data结构体空间

drvdata **=** devm\_kzalloc**(**dev**,** **sizeof(\***drvdata**),** GFP\_KERNEL**);**

//初始化samsung\_pinctrl\_drv\_data

ctrl **=** samsung\_pinctrl\_get\_soc\_data**(**drvdata**,** pdev**);**

//drvdata关联dev

drvdata**->**dev **=** dev**;**

//pinctrl操作对应的虚拟地址

res **=** platform\_get\_resource**(**pdev**,** IORESOURCE\_MEM**,** 0**);**samsung\_pinctrl\_parse\_dt

drvdata**->**virt\_base **=** devm\_ioremap\_resource**(&**pdev**->**dev**,** res**);**

//获取pinctrl对应的irq号，这里没有

res **=** platform\_get\_resource**(**pdev**,** IORESOURCE\_IRQ**,** 0**);**

**if** **(**res**)**

drvdata**->**irq **=** res**->**start**;**

//注册到gpio子系统中

ret **=** samsung\_gpiolib\_register**(**pdev**,** drvdata**);**

//注册到pinctrl子系统中

ret **=** samsung\_pinctrl\_register**(**pdev**,** drvdata**);**

//gpio相关中断注册到中断子系统中

**if** **(**ctrl**->**eint\_gpio\_init**)**

ctrl**->**eint\_gpio\_init**(**drvdata**);**

**if** **(**ctrl**->**eint\_wkup\_init**)**

ctrl**->**eint\_wkup\_init**(**drvdata**);**

//设置device的私有数据

platform\_set\_drvdata**(**pdev**,** drvdata**);**

/\* Add to the global list \*/

list\_add\_tail**(&**drvdata**->**node**,** **&**drvdata\_list**);**

**return** 0**;**

**}**

probe函数对pinctrl、中断、gpio子系统都进行了操作。