# Linux驱动\_设备树

这里仅仅对2440上的设备树进行探索。

## 1.设备树起源

Linus Torvalds在2011年3月17日的ARM Linux邮件列表宣称“this whole ARM thing is a f\*cking pain in the ass”，引发ARM Linux社区的地震，随后ARM社区进行了一系列的重大修正。在过去的ARM Linux中，arch/arm/plat-xxx和arch/arm/mach-xxx中充斥着大量的垃圾代码，相当多数的代码只是在描述板级细节，而这些板级细节对于内核来讲，不过是垃圾，如板上的platform设备、resource、i2c\_board\_info、spi\_board\_info以及各种硬件的platform\_data。读者有兴趣可以统计下常见的s3c2410、s3c6410等板级目录，代码量在数万行。

社区必须改变这种局面，于是PowerPC等其他体系架构下已经使用的Flattened Device Tree（FDT）进入ARM社区的视野。Device Tree是一种描述硬件的数据结构，它起源于 OpenFirmware (OF)。在Linux 2.6中，ARM架构的板极硬件细节过多地被硬编码在arch/arm/plat-xxx和arch/arm/mach-xxx，采用Device Tree后，许多硬件的细节可以直接透过它传递给Linux，而不再需要在kernel中进行大量的冗余编码。  
Device Tree由一系列被命名的结点（node）和属性（property）组成，而结点本身可包含子结点。所谓属性，其实就是成对出现的name和value。在Device Tree中，可描述的信息包括（原先这些信息大多被hard code到kernel中）：

CPU的数量和类别

内存基地址和大小

总线和桥

外设连接

中断控制器和中断使用情况

GPIO控制器和GPIO使用情况

Clock控制器和Clock使用情况

它基本上就是画一棵电路板上CPU、总线、设备组成的树，Bootloader会将这棵树传递给内核，然后内核可以识别这棵树，并根据它展开出Linux内核中的platform\_device、i2c\_client、spi\_device等设备，而这些设备用到的内存、IRQ等资源，也被传递给了内核，内核会将这些资源绑定给展开的相应的设备。

**从设备树导入的历史看，为了缓解不同开发板产生的大量冗余代码，使用一个树的形式将设备/总线等描述放在一个配置文件中。**

## 2. 在2440上导入设备树

1.编译设备树 make dtbs，生成s3c2440-tq2440-dt.dtb。

2.带设备树linux系统的启动

下載uImage：tftp 0x30008000 uImage**;**

下載ramdisk：tftp 0x31000000 ramdisk**.**img**;**

下載設備樹文件：tftp 0x33000000 s3c2440**-**tq2440**-**dt**.**dtb**;**

啓動：bootm 0x30008000 0x31000000 0x33000000

3.设备树结构

s3c2440**-**tq2440**-**dt**.**dts

**---->** s3c2440**.**dtsi

**---->** s3c24xx**.**dtsi

**---->** skeleton**.**dtsi

**---->** s3c2440**-**pinctrl**.**dtsi

skeleton.dtsi 存放的是一個設備樹必備的一些基本屬性。  
s3c24xx.dtsi 中存放的是整個s3c24xx系列SoC公共的一些屬性，如中斷控制器、串口、看門狗、RTC、I2C控制器等等。  
s3c2440-pinctrl.dtsi 存放的是s3c2440這款SoC中GPIO控制器、外部中斷控制器、引腳複用等信息的配置  
s3c2440.dtsi 存放的是s3c2440這個SoC跟其他s3c24xx系列不同的一些硬件信息，如clock控制器、串口等等。  
s3c2440-tq2440-dt.dts 存放的是tq2440的硬件信息。

4.修改arch/arm/boot/dts/Makefile

dtb**-**$**(**CONFIG\_ARCH\_S3C24XX**)** **+=** \

s3c2440**-**tq2440**-**dt**.**dtb

這樣在make dtbs編譯設備樹的時候就會編譯s3c2440-tq2440-dt.dts，在arch/arm/boot/dts/下生成s3c2440-tq2440-dt.dtb。

5.修改Makefile和Kconfig，添加tq2440板子的信息，以便在kernel啓動的時候能夠用從設備樹鏡像中解析到的信息匹配到tq2440板子。

修改arch/arm/mach-s3c24xx/Kconfig

config MACH\_TQ2440\_DT

bool "TQ2440 development board using device tree"

select CLKSRC\_OF

select USE\_OF

select PINCTRL

select PINCTRL\_S3C24XX

help

Say Y here to select support **for** the TQ2440**.**

修改arch/arm/mach-s3c24xx/Makefile

obj**-**$**(**CONFIG\_MACH\_TQ2440\_DT**)** **+=** mach**-**tq2440**-**dt**.**o

6.添加arch/arm/mach-s3c24xx/mach-tq2440-dt.c

#include <linux/clocksource.h>

#include <linux/irqchip.h>

#include <linux/serial\_s3c.h>

#include <asm/mach/arch.h>

#include <mach/map.h>

#include <plat/cpu.h>

#include <plat/pm.h>

#include "common.h"

static void \_\_init tq2440\_dt\_map\_io**(**void**)**

**{**

s3c24xx\_init\_io**(NULL,** 0**);**

**}**

static void \_\_init tq2440\_dt\_machine\_init**(**void**)**

**{**

s3c\_pm\_init**();**

**}**

static const char **\***const tq2440\_dt\_compat**[]** \_\_initconst **=** **{**

"samsung,s3c2440"**,**

"samsung,tq2440"**,**

**NULL**

**};**

DT\_MACHINE\_START**(**TQ2440\_DT**,** "Samsung S3C2440 (Flattened Device Tree)"**)**

**.**dt\_compat **=** tq2440\_dt\_compat**,**

**.**map\_io **=** tq2440\_dt\_map\_io**,**

**.**init\_irq **=** irqchip\_init**,**

**.**init\_machine **=** tq2440\_dt\_machine\_init**,**

MACHINE\_END

注意dt\_compat數組，其中的值要跟設備樹中的compatible匹配，这里为arch/arm/boot/dts/s3c2440-tq2440-dt.dts。

**/**dts**-**v1**/;**

#include "s3c2440.dtsi"

#include <dt-bindings/interrupt-controller/irq.h>

#include <dt-bindings/clock/s3c2410.h>

**/** **{**

model **=** "TQ2440"**;**

compatible **=** "samsung,s3c2440"**,** "samsung,tq2440"**;**

memory **{**

reg **=** **<**0x30000000 0x34000000**>;**

**};**

7.删除uboot的bootargs

如果uboot中設置了bootargs屬性的話，在boot的之前它會修改設備樹鏡像，覆蓋其中chosen節點中的bootargs屬性，爲了便於調試，我在uboot中執行setenv bootargs命令，這樣就可以刪除uboot中bootargs環境變量。

下面是設備樹（s3c2440-tq2440-dt.dts）中chosen節點的定義：

chosen**{**

bootargs **=** "root=/dev/ram0 rw rootfstype=ext2 console=ttySAC0,115200n8 init=/linuxrc ignore\_loglevel earlyprintk"**;**

**};**

## 3. 设备树文件解析