Linux 设备树学习笔记 (三、修改 uboot、linux-4.15 内核在 JZ2440 使用设备树) 修改设备...

1. 实验开发环境

u-boot：u-boot-2016.03

Linux 内核：linux-4.15

开发板：JZ2440 开发板

2. 添加 uboot mtd device_tree 分区保存 设备树 文件

在 uboot 的 JZ2440 开发板的配置文件 **include/configs/jz2440.h** 在 **MTDPARTS_DEFAULT** 中添加 **device_tree** 分区，代码如下：

```
#define MTDPARTS_DEFAULT      "mtdparts=jz2440-0:512k(u-boot),"   \
                              "128k(device_tree)," \  /*添加device_tree
                              "256k(params),"     \
                              "4m(kernel),"   \
                              "-(rootfs)"     \
```

3. 修改内核 mtd 分区

uboot 修改了分区，linux 也要修改对应的分区，**在 arch/arm/mach-s3c24xx/common-smdk.c** 的 **smdk_default_nand_part** 添加 **device_tree** 分区，代码如下：

```
/* NAND parititon from 2.4.18-swl5 */

static struct mtd_partition smdk_default_nand_part[] = {
    [0] = {
        .name   = "bootloader",
        .size   = SZ_512K,
```

```
        .offset = 0,
    },
    [1] = {  /*添加device_tree分区*/
        .name   = "device_tree",
        .size   = SZ_128K,
        .offset = MTDPART_OFS_APPEND,
    },
    [2] = {
        .name   = "params",
        .offset = MTDPART_OFS_APPEND,
        .size   = SZ_256K,
    },
    [3] = {
        .name   = "kernel",
        .offset = MTDPART_OFS_APPEND,
        .size   = SZ_4M,
    },
    [4] = {
        .name   = "rootfs",
        .offset = MTDPART_OFS_APPEND,
        .size   = MTDPART_SIZ_FULL,
    },
};
```

4. 修改内核代码支持 JZ2440 使用设备树

注：这部分的修改参考韦东山设备树视频配套资料的 linux-4.19-rc3_device_tree_for_jz2440.patch 补丁文件

1. 修改 arch/arm/mach-s3c24xx/mach-smdk2440.c 文件：

添加如下代码：

```
static const char *const smdk2440_dt_compat[] __initconst = {
    "samsung,smdk2440",
    NULL
};
```

```
MACHINE_START(S3C2440, "SMDK2440")
    /* Maintainer: Ben Dooks <ben-linux@fluff.org> */
    .atag_offset    = 0x100,

    .dt_compat      = smdk2440_dt_compat, /*添加*/

    .init_irq   = s3c2440_init_irq,
    .map_io     = smdk2440_map_io,
    .init_machine   = smdk2440_machine_init,
    .init_time  = smdk2440_init_time,
MACHINE_END
```

2. 修改 **arch/arm/plat-samsung/gpio-samsung.c** 的 **samsung_gpiolib_init** 函数：

注释掉以下代码：

```
//  if (of_have_populated_dt())
//      return 0;
```

3. 在 **arch/arm/boot/dts/Makefile** 添加如下代码：

```
dtb-$(CONFIG_ARCH_S3C2440) += jz2440.dtb
```

4. 把 **jz2440.dts** 文件拷贝到 **arch/arm/boot/dts/** 目录下，**jz2440.dts** 的代码如下：

```
// SPDX-License-Identifier: GPL-2.0
/*
 * SAMSUNG SMDK2440 board device tree source
 *
 * Copyright (c) 2018 weidongshan@qq.com
 * dtc -I dtb -O dts -o jz2440.dts jz2440.dtb
 */

#define S3C2410_GPA(_nr)    ((0<<16) + (_nr))
#define S3C2410_GPB(_nr)    ((1<<16) + (_nr))
#define S3C2410_GPC(_nr)    ((2<<16) + (_nr))
#define S3C2410_GPD(_nr)    ((3<<16) + (_nr))
#define S3C2410_GPE(_nr)    ((4<<16) + (_nr))
#define S3C2410_GPF(_nr)    ((5<<16) + (_nr))
#define S3C2410_GPG(_nr)    ((6<<16) + (_nr))
#define S3C2410_GPH(_nr)    ((7<<16) + (_nr))
#define S3C2410_GPJ(_nr)    ((8<<16) + (_nr))
#define S3C2410_GPK(_nr)    ((9<<16) + (_nr))
#define S3C2410_GPL(_nr)    ((10<<16) + (_nr))
#define S3C2410_GPM(_nr)    ((11<<16) + (_nr))

/dts-v1/;

/ {
    model = "SMDK24440";
    compatible = "samsung,smdk2440";

    #address-cells = <1>;
    #size-cells = <1>;

    memory@30000000 {
        device_type = "memory";
```

```
        reg =  <0x30000000 0x4000000>;
    };
/*
    cpus {
        cpu {
            compatible = "arm,arm926ej-s";
        };
    };
*/
    chosen {
        bootargs = "noinitrd root=/dev/mtdblock4 rw init=/linuxrc cons
    };


    led {
        compatible = "jz2440_led";
        reg = <S3C2410_GPF(5) 1>;
    };
};
```

5. 测试

1. 重新编译内核，设备树，在 linux 内核顶层目录下，输入以下命令：

```
make uImage -j4
make dtbs  /*编译设备树，生成jz2440.dtb文件*/
```

2. 下载内核，设备树到开发板测试：

```
nfs 30000000 192.168.0.102:/home/book/works/first_fs/uImage
nfs 32000000 192.168.0.102:/home/book/works/first_fs/jz2440.dtb

bootm 30000000 - 32000000
```

打印信息如下，可见 JZ2440 可以使用设备树了。

```
## Booting kernel from Legacy Image at 30000000 ...
   Image Name:   Linux-4.15.0
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    3485880 Bytes = 3.3 MiB
   Load Address: 30108000
   Entry Point:  30108000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 32000000
   Booting using the fdt blob at 0x32000000
   Loading Kernel Image ... OK
   Loading Device Tree to 33afb000, end 33afe1b8 ... OK

Starting kernel ...

Booting Linux on physical CPU 0x0
Linux version 4.15.0 (book@book-virtual-machine) (gcc version 4.9.4
CPU: ARM920T [41129200] revision 0 (ARMv4T), cr=c0007177
```


全文完