

根据ePAPR,在dts中可以定义kernel启动参数

比如

```
chosen {  
  
    bootargs = "mem=64M console=ttyS0,115200n8 loglevel=8 debug  
earlycon=uart8250,mmio,0xf8060000,115200n8";  
  
};
```

但现实中kernel收到的bootargs很有可能与chosen中的大相径庭。原因：u-boot会修改dtb中的bootargs!

在u-boot 2014.01中

in arch/arm/lib/bootm.c

do\_bootm\_linux()

|

|

\\

boot\_prep\_linux()

|

|

\\

image\_setup\_linux()

|  
|  
\\

image\_setup\_libfdt()

|  
|  
\\

fdt\_chosen()

int fdt\_chosen(void \*fdt, int force)

{

.....

/\*

\* Create /chosen properties that don't exist in the fdt.

\* If the property exists, update it only if the "force" parameter

\* is true.

\*/

**str = getenv("bootargs");**

if (str != NULL) {

    path = **fdt\_getprop**(fdt, nodeoffset, "bootargs", NULL);

    if ((path == NULL) || force) {

        err = **fdt\_setprop**(fdt, nodeoffset,

```

        "bootargs", str, strlen(str)+1);

    if (err < 0)

        printf("WARNING: could not set bootargs %s.\n",

               fdt_strerror(err));

    }

}

.....

}

```

即如果u-boot的bootargs environment variable非空，同时dtb中"bootargs" is empty (这个条件在这里不成立)或者force非零（而这个条件成立，因为fdt\_chosen传入的force参数为hardcode的const value 1），则会用u-boot bootargs environment variable中的值替换dtb中的值。

由于目前的code中force总是1，所以无论原来dtb中是否定义了bootargs 参数，只要u-boot中定义了，都会被替换。

比如在Gr2 / Gs2 的u-boot中，config后自动生成的include/autoconf.mk中

```
CONFIG_BOOTARGS="console=ttyS0,115200n8 earlyprintk=serial,ttyS0,115200"
```

则该bootarg将替换在dtb的/chosen中指定的bootarg。

只要注释掉在include/confogs/pegmatite.h中的如下macro，则fdt\_chosen()就不会替换dtb中的设置了。

（ pegmatite.h是config u-boot时自动生成的 ）

```
#define CONFIG_BOOTARGS "console=ttyS0,115200n8 earlyprintk=serial,ttyS0,115200"
```

