

专题2-U-Boot入门

一、U-Boot编译

1.1、编译操作

1.1.1、配置U-Boot

OK6410: `make forlinx_nand_ram256_config`

在Makefile文件中可以找到对应的板子信息

```
root@localhost:/home/S5-driver/lesson6/u-boot
File Edit View Search Terminal Help
1923 smdk2412_config :      unconfig
1924     @$(MKCONFIG) $(@:_config=) arm s3c24xx smdk2412 samsung s3c2412
1925
1926 smdk2440_config :      unconfig
1927     @$(MKCONFIG) $(@:_config=) arm s3c24xx smdk2440 samsung s3c2440
1928
1929 smdk2460_config :      unconfig
1930     @$(MKCONFIG) $(@:_config=) arm s3c24xx smdk2460 samsung s3c2460
1931
1932 smdk24a0_config :      unconfig
1933     @$(MKCONFIG) $(@:_config=) arm s3c24xx smdk24a0 samsung s3c24a0
1934
1935 tabla_config :        unconfig
1936     @$(MKCONFIG) $(@:_config=) arm s3c64xx tabla samsung mdirac3
1937
1938 smdk6400_config :      unconfig
1939     @$(MKCONFIG) $(@:_config=) arm s3c64xx smdk6400 samsung s3c6400
1940
1941 smdk6410_config :      unconfig
1942     @$(MKCONFIG) $(@:_config=) arm s3c64xx smdk6410 samsung s3c6410
1943
1944 forlinx_nand_ram128_config : unconfig
1945     @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 NAND ram128
1946
1947 forlinx_sd_ram128_config : unconfig
1948     @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 SD ram128
1949
1950 forlinx_nand_ram128_hdmi_config : unconfig
1951     @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 NAND ram128 hdmi
1952
1953 forlinx_nand_ram256_config : unconfig
1954     @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 NAND ram256
1955
1956 forlinx_sd_ram256_config : unconfig
1957     @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 SD ram256
1958
1959 forlinx_nand_ram256_hdmi_config : unconfig
1960     @$(MKCONFIG) smdk6410 arm s3c64xx smdk6410 samsung s3c6410 NAND ram256 hdmi
/6410
root@localhost:/home...
```

【1】对U-Boot进行配置

`make forlinx_nand_ram256_config`

【2】执行编译操作（指定平台，交叉工具链）

`make ARCH=arm CROSS_COMPILE=arm-linux-`

【3】得到所需平台的U-Boot文件，使用dnw下载在开发板中。

`/home/dnw u-boot.bin 0x50008000`

1.1.2、下载与运行（这是在MMU打开之后的内存地址）

OK6410: `tftp 0xc0008000 uImage`

二、U-Boot命令详解

2.1、help（帮助命令）

2.2、printenv（查看环境变量）

2.3、setenv（添加、修改、删除环境变量）

```
v setenv name value ...  
- add/modify environment variable 'name' to 'value '  
v setenv name  
- delete environment variable 'name'
```

2.4、saveenv (保存环境变量)

将当前定义的所有变量及其值存入flash 中。

2.5、tftp (通过网络下载文件)

注意：使用tftp，需要先配置好网络

```
#setenv ethaddr 12:34:56:78:9A:BC  
#setenv ipaddr 192.168.0.3  
#setenv serverip 192.168.0.2 ( tftp服务器的地址 )
```

```
#ping 192.168.0.2(tftp服务器的地址)
```

范例：

```
#tftp 0xc0008000 uImage  
把tftp服务器上的uImage下载到0xc0008000处。
```

2.6、bootm (执行程序)

```
bootm {addr} {arg}
```

执行固定格式的2进制程序

范例：

```
#bootm 0xc0008000
```

2.7、md (显示内存区的内容)

md采用十六进制和ASCII码两种形式来显示存储单元的内容。 这条命令还可以采用长度标识符 .l, .w和.b：

```
md [.b, .w, .l] address
```

范例：

```
md.w 100000  
00100000: 2705 1956 5050 4342 6f6f 7420 312e 312e  
00100010: 3520 284d 6172 2032 3120 3230 3032 202d
```

2.8、mm (修改内存，地址自动递增)。

```
mm [.b, .w, .l] address
```

mm 提供了一种互动修改存储器内容的方法。它会显示地址和当前值，然后提示用户输入。如果你输入了一个合法的十六进制数，这个新的值将会被写入该地址。然后提示下一个地址。如果你没有输入任何值，只是按了一下回车，那么该地址的内容保持不变。如果想结束输入，则输入空格，然后回车。

```
=> mm 100000  
00100000: 27051956 ? 0  
00100004: 50504342 ? AABCCDD
```

2.9、nand erase (擦除nand flash)

```
nand erase 起始地址start 长度len
```

擦除start处开始的，长度为len的区域

范例：

```
#nand erase 0x400000 0x500000
```

2.10、nand write/nand read (写/读nand flash)

```
nand write 内存起始地址 flash起始地址 长度len
```

将内存起始地址处，长度为len的数据，写入flash起始地址处。

范例：

```
#nand write c0008000 400000 500000
```

```
nand read 内存起始地址 flash起始地址 长度len
```

将flash起始地址处，长度为len的数据，读到内存起始地址处。

范例：

```
#nand read c0008000 400000 500000
```

2.11、设置自启动

1: 设置从nand flash自动启动

```
#setenv bootcmd nand read c0008000 400000 500000 \; bootm c0008000
```

2：设置自动下载内核到内存后启动

```
#setenv bootcmd tftp c0008000 uImage \; bootm c0008000
```

