Product Manual

SylixOS 启动手册

PM0010020003 V1.01 Date: 2016/11/02

产品使用手册

类别	内容
关键词	MBR、u-boot、PMON、grub
摘　要	介绍 SylixOS 启动方式

翼辉信息 ACOINFO

修订历史

版本	日期	原因
V1.00	2018/3/27	创建文档

目 录

第 1 章 BootLoader 简介

BootLoader 就是在操作系统运行之前运行的一段小程序，通过这段小程序，可以初始化硬件设备，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统做好准备。

1.1 grub 简介

grub 是引导操作系统的程序，它会根据自己的配置文件，去引导内核，当内核被加载到内存以后，内核会根据 grub 配置文件中的配置，找到根分区所使用的文件系统对应的驱动，通过根分区文件系统对应的驱动，挂载根分区，从而达到启动操作系统的目的。

grub 的代码分三个阶段被加载.

- grub stage 1

存在启动硬盘的 0 柱面，0 磁道，第一个扇区中，即 MBR 中，MBR 的前 446 个字节为以引导代码，也就是 grub stage 1，所谓的 stage1 ，作用只有一个，就是找到 grubstage1.5，然后将其加载的到内存。

- grub stage 1.5

它的作用就是用来识别常见的不同类型的文件系统,从而找到"/boot 目录所在的分区"对应的文件系统的驱动，驱动多大、存在于哪些扇区中，这些都是在安装操作系统的时候根据用户的设置自动生成的，我们有了"/boot 目录所在分区"的文件系统驱动，那么 /boot/grub/stage2 这样较大的文件可以直接操作了。

- grub stage 2

grub 真正的核心程序，能让用户以菜单方式将操作系统加载、新增参数、修改选项，这些全都是 stage2 的功用，前面的 grub stage 1 和 grub stage 1.5 所做的事情就是为了运行 grub stage 2，然后由 stage 2 借助 grub.conf 再去引导系统启动。

1.1.1 MBR 简介

MBR（master boot record），即主引导记录，有时也称主引导扇区。位于整个硬盘的 0 柱面 0 磁头 1 扇区（可以看作是硬盘的第一个扇区），BIOS 在执行自己固有的程序以后就会跳转到 MBR 中的第一条指令。将系统的控制权交由 MBR 来执行。在总共 512byte 的主引导记录中，MBR 的引导程序占了其中的前 446 个字节（偏移 0H~偏移 1BDH），随后的 64 个字节（偏移 1BEH~偏移 1FDH）为 DPT（Disk Partition Table，硬盘分区表），最后的两个字节“55 AA”（偏移 1FEH~偏移 1FFH）是分区结束标志。

主引导记录（MBR，Main Boot Record）是位于磁盘最前边的一段引导（Loader）代码。它负责磁盘操作系统(DOS)对磁盘进行读写时分区合法性的判别、分区引导信息的定位，它由磁盘操作系统(DOS)在对硬盘进行初始化时产生的。

主引导记录中包含了硬盘的一系列参数和一段引导程序。其中的硬盘引导程序的主要作用是：（1）检查分区表是否正确；（2）系统硬件完成自检以后引导具有激活标志的分区上的操作系统，并将控制权交给启动程序。MBR 是由分区程序（fdisk）所产生的，它不依赖任何操作系统，意即不同的操作系统可能会存在相同的 MBR，即使不同，MBR 也不会夹带操

作系统的性质。具有公共引导的特性，而且硬盘引导程序也是可以改变的，从而实现多系统共存。

1.1.2 grub 配置文件

把 grldr 及菜单配置文件 menu.lst 复制到系统盘/boot 即可。其中 grldr 会自动搜索菜单配置文件并加载，但是 menu.lst 最好存放在 C 盘根目录，免得到时候搜索不到不能加载引起错误)。把需要 grldr 引导的文件（如 DOS.IMG、GHOST83.IMG 等镜像文件）保存在硬盘。

SylixOS 在 x86 平台的启动使用 grub 作为启动 loader，下面是一个 menu.lst 的例子。(以#开始的行，表示注释，不执行)：

```
timeout 5
default 1
title  SylixOS(UP)
kernel /bspx86.elf ncpus=1 hz=1000 hhz=1000 console=/dev/ttyS0 kdlog=no
kderror=yes kfpu=no heapchk=yes utc=no rfsmap=/boot:/media/hdd0,/:/media/hdd1
video=uvesafb:ywrap,mtrr:3,640x480-32@60
title  SylixOS(NORMAL)
kernel /bspx86.elf hz=1000 hhz=1000 console=/dev/ttyS0 kdlog=no kderror=yes
kfpu=no heapchk=yes utc=no rfsmap=/boot:/media/hdd0,/:/media/hdd1
video=uvesafb:ywrap,mtrr:3,640x480-32@60
```

- #默认延迟时间

```
timeout 5
```

- #第一项为默认值

```
default 1
```

- #title 单核启动（UP）

```
title  SylixOS(UP)
kernel  /bspx86.elf
```

- #title 多核启动（NORMAL）

```
title  SylixOS(NORMAL)
kernel  /bspx86.elf
```

- #如果 ncpus 存在，得知 CPU 个数，系统不需要去探测

```
ncpus=1 // 单 CPU 工作
```

- #系统 tick 频率，默认为 100(推荐 100~10000 中间)，此处使用 1000

```
hz=1000
```

- #高速定时器频率，默认与 hz 相同（需 BSP 支持）

```
hhz=1000
```

- #使用 ttyS0 串行端口终端

```
console=/dev/ttyS0
```

- #内核 DEBUG　LOG 信息打印

```
kdlog=no
```

- #内核 DEBUG ERROR 信息打印

```
kderror=yes
```

- #内核态对浮点支持（推荐为 no）

```
kfpu=no
```

- #内存堆越界检查

```
heapchk=yes
```

- #关闭 UTC 为本地时间

```
utc=no
```

- #这是根文件系统映射关系选项，用逗号隔开，/boot 等为可选映射，/为必须映射

```
rfsmap=/boot:/media/hdd0,/:/media/hdd1
```

- #设置视频参数

```
video=uvesafb:ywrap, mtrr:3,640x480-32@60
```

注：关于 grub 启动 SylixOS 系统的方法可以参考《RealEvo-IDE 使用手册》第 8 章安装 SylixOS。

1.2 u-boot 简介

u-boot 是德国 DENX 小组开发的用于多嵌入式 CPU 的 BootLoader 程序，u-boot 不仅支持嵌入式 Linux 系统的引导，还支持 SylixOS、VxWorks 等多种嵌入式操作系统。

1.2.1 u-boot 启动过程

u-boot 的启动过程是多阶段实现的，分了两个阶段：

第一阶段是用汇编写的，主要任务是：

- CPU 自身初始化：包括 MMU，Cache，时钟系统，SDRAM 控制器等的初始化；
- 重定位：把自己从非易失性存储器搬移到 RAM 中；
- 分配堆栈空间，设置堆栈指针；
- 清零 BSS 数据段；
- 跳转到第二阶段入口函数 start_armboot()。

第二阶段是由 C 语言写的，主要任务是：

- 为 u-boot 内部私有数据分配存储空间，并清零；
- 依次调用函数指针数组 init_sequence 中定义的函数进行一系列的初始化；
- 如果系统支持 NOR Flash，调用 flash_init()和 display_flash_config()初始化并显示检测到的器件信息；
- 如果系统支持 LCD 或 VFD，调用 lcd_setmem()或 vfd_setmem()计算帧缓（Framebuffer）大小，然后在 BSS 数据段之后为 Framebuffer 分配空间，初始化 gd->fb_base 为 Framebuffer 的起始地址；
- 调用 mem_malloc_init()进行存储分配系统（类似于 C 语言中的堆）的初始化和空间分配；
- 如果系统支持 NAND Flash，调用 nand_init()进行初始化；
- 如果系统支持 DataFlash，调用 AT91F_DataflashInit()和 dataflash_print_info()进行初始化并显示检测到的器件信息；
- 调用 env_relocate()进行环境变量的重定位，即从 Flash 中搬移到 RAM 中；

- 如果系统支持 VFD，调用 drv_vfd_init()进行 VFD 设备初始化；
- 从环境变量中读取 IP 地址和 MAC 地址，初始化 gd->bd->bi_ip_addr 和 gd->bd->bi_enetaddr；
- 调用 jumptable_init()进行跳转表初始化，跳转表在 global_data 中；
- 调用 console_init_r()进行控制台初始化；
- 如果需要，调用 misc_init_r()进行杂项初始化；
- 调用 enable_interrupts()打开中断；
- 如果需要，调用 board_late_init()进行单板后期初始化，对于 AT91SAM9260EK，主要是以太网初始化；
- 进入主循环：根据用户的选择启动 linux，或者进入命令循环执行用户输入的命令。

1.2.2 u-boot 命令介绍

u-boot 发展到现在命令行模式已经非常接近 Linux 下的 shell，命令行模式下支持“Tab”键的模式补全和命令的历史记录功能，而且如果输入的命令和前几个字符和别的命令不重复，那么只需要输入这几个字符即可，例如 version 命令，在所有的 u-boot 命令中没有以其他以“v”开头的命令，所有只需输入“v”即可：

```
zynq-uboot> v


U-Boot 2013.10 (May 25 2015 - 15:40:17)
arm-xilinx-linux-gnueabi-gcc (Sourcery CodeBench Lite 2011.09-50) 4.6.1
GNU ld (Sourcery CodeBench Lite 2011.09-50) 2.21.53.20110905
```

下面对 u-boot 中支持的命令做简单介绍。

命令：**help** 或 **?**

说明：查看 u-boot 命令的帮助信息。

命令：**base**

说明：打印或者设置地址偏移。

```
zynq-uboot> base
Base Address: 0x00000000
```

命令：**bdinfo**

说明：打印板子信息结构。

```
zynq-uboot> bdinfo
arch_number     = 0x0B0B518C
boot_params     = 0x10C83195
DRAM bank       = 0x00000000
-> start        = 0x00000000
-> size         = 0x40000000
eth0name        = Gem.e000b000
ethaddr         = 00:0a:35:00:01:78
current eth     = Gem.e000b000
```

```
ip_addr         = 192.168.1.111
baudrate        = 115200 bps
TLB addr        = 0x3FFF0000
relocaddr       = 0x3FF72000
reloc off       = 0x3BF72000
irq_sp          = 0x3FB51F48
sp start        = 0x3FB51F38
ARM frequency   = 1128878610 MHz
DSP frequency   = 1079866533 MHz
DDR frequency   = -1609813823 MHz
```

命令：**boot** 或者 **bootd**

说明：执行启动模式运行 bootcmd 环境变量中指定的方式。

命令：**bootelf**

说明：从内存中启动 ELF 文件格式的程序。

命令：**bootm**

说明：从内存中启动应用程序。

命令：**bootp**

说明：从网络（BOOTP/TFTP）中启动程序。

命令：**clk**

说明：打印系统时钟。

```
zynq-uboot> clk dump
clk           frequency
   armpll       1333333320
   ddrpll       1066666656
    iopll       999999990
 cpu_6or4x      666666660
 cpu_3or2x      333333330
   cpu_2x       222222220
   cpu_1x       111111110
   ddr_2x       355555552
......
```

命令：**cmp**

说明：比较内存内容。

命令：**coninfo**

说明：打印终端设备和信息。

```
zynq-uboot> coninfo
List of available devices:
```

```
serial   80000003 SIO stdin stdout stderr
ttyPS1   00000003 .IO
ttyPS0   00000003 .IO
```

命令：**cp**

说明：内存复制。

命令：**crc32**

说明：CRC 校验。

命令：**dcache**

说明：使能和禁止数据 CACHE。

命令：**echo**

说明：回显命令。

命令：**editenv**

说明：编辑环境变量。

```
zynq-uboot> editenv sylixos
edit: sylixos
zynq-uboot> print sylixos
sylixos=sylixos
```

命令：**env**

说明：环境变量操作。

```
env default [-f] -a  复位环境变量
env delete [-f]      删除环境变量
env edit name        编辑环境变量 name 的值
env export [-t | -b | -c] [-s size] addr [var ...]     导出环境变量
env import [-d] [-t | -b | -c] addr [size]             导入环境变量
env print [-a | name ...]       打印环境变量
env run var [...]               运行环境变量中的命令
env save                        保存环境变量
env set [-f] name [arg ...]     设置环境变量
```

命令：**exit**

说明：退出脚本。

命令：**ext2load**

说明：从 EXT2 文件系统中加载二进制文件。

命令：**ext2ls**

说明：列出 EXT2 文件系统中的文件（默认为根目录“/”）。

命令：**false**

说明：什么也不做，不成功。

命令：**fatinfo**

说明：打印指定设备的文件系统信息。

```
zynq-uboot> fatinfo mmc 0:1
Interface:  MMC
  Device 0: Vendor: Man 000003 Snr 4511c000 Rev: 0.7 Prod: SU04G€
            Type: Removable Hard Disk
            Capacity: 3781.5 MB = 3.6 GB (7744512 x 512)
Filesystem: FAT32 "NO NAME    "
```

命令：**fatload**

说明：从 DOS 文件系统（FAT 等）中加载二进制文件。

命令：**fatls**

说明：列出指定设备的 DOS 文件系统中的文件（默认为根目录“/”）。

命令：**fpga**

说明：加载 FPGA 镜像。

命令：**go**

说明：开始应用程序在指定地址。

命令：**icache**

说明：使能或禁止指令 CACHE。

命令：**iminfo**

说明：打印应用程序镜像的头信息。

命令：**loadb**

说明：通过串口加载二进制文件（kermit 模式）。

命令：**loads**

说明：通过串口加载 S-Record 文件。

命令：**loadx**

说明：通过串口加载二进制文件（xmodem 模式）。

命令：**loady**

说明：通过串口加载二进制文件（ymodem 模式）。

命令：**md**

说明：显示指定内存的内容。

```
zynq-uboot> md 0x20000000
20000000: 1bdfeafe bf9be3f7 2f73ba6f a3dedf71    ........o.s/q...
20000010: 85fc906f 95ece9bd 7fa80fbb b7cef83f    o............?...
20000020: bef9ff7d 7b79ff7b fff7bf39 3bfabfb7    }...{.y{9......;
20000030: d6fdbfdf f6a4e8b7 feeb7c5f ea0fbe37    ........_|..7...
20000040: 24ab9a62 fbfacefa efebfef5 edfbe2ff    b..$............
20000050: b5dfe4f2 97eb0fcf f5c5d77b d22e9f77    ........{...w...
20000060: f9bfeff7 f5afbef7 ba36abfd eafebfab    ..........6.....
```

```
20000070: e6fbedbf 37bb3795 abf7fcbf 73ee394b    .....7.7....K9.s
20000080: b675bfbf 677efdaf e6fff93f 416b8eb4    ..u...~g?.....kA
20000090: e69eedff a7abecff e5bbebfa bfa3bcaf    ................
200000a0: e37cf6eb 9dfeddaf cbf8ff65 f9d7fff7    ..|.....e.......
200000b0: dea6ec9d ff7be4ec af3b3cbf 97ce2c74    ......{..<;.t,..
200000c0: eff7f4c9 a1c6faff e9ebd9fe 27faaeff    ............'
200000d0: fffd9f9a fddfc3ff dfedeebf d5fa6ebb    ............n..
200000e0: bcfcae67 de03f9cd 8ebe9cbf 7fdbad4b    g...........K...
200000f0: edafd4ef 6be8f9ef bbb0dfed b75ca8ff    .......k......\.
```

命令：**mdio**

说明：MDIO 操作命令。

```
mdio list                                        列出 MDIO 总线信息
mdio read <phydev> [<devad>.]<reg>               读 MDIO 寄存器
mdio write <phydev> [<devad>.]<reg> <data>       写 MDIO 寄存器
```

命令：**mii**

说明：MII 操作命令。

```
mii device                          列出 MII 可用的设备
mii device <devname>                设置当前设备
mii info   <addr>                   显示 MII  PHY 信息
mii read   <addr> <reg>             读 MII  PHY 寄存器
mii write  <addr> <reg> <data>      写 MII  PHY 寄存器
mii dump   <addr> <reg>             打印地址或寄存器信息
```

命令：**mm**

说明：修改指定地址的内存信息（地址自动增加）。

命令：**mmc**

说明：MMC 存储操作命令。

```
mmc read addr blk# cnt      读取 MMC 设备内容
mmc write addr blk# cnt     写 MMC 设备
mmc erase blk# cnt          擦除 MMC 设备
mmc rescan                  扫描 MMC 设备
mmc part                    列出当前 MMC 设备的分区信息
mmc dev [dev] [part]        显示或设置当前 MMC 设备分区
mmc list                    列出当前可用的 MMC 设备
```

命令：**mmcinfo**

说明：显示当前 MMC 设备信息。

```
zynq-uboot> mmcinfo
Device: zynq_sdhci
```

```
Manufacturer ID: 3
OEM: 5344
Name: SU04G
Tran Speed: 50000000
Rd Block Len: 512
SD version 3.0
High Capacity: Yes
Capacity: 3.7 GiB
Bus Width: 4-bit
```

命令：**mw**

说明：向指定的内存地址写入内容。

命令：**nfs**

说明：通过网络（NFS 协议）启动镜像。

命令：**nm**

说明：修改指定的内存地址。

命令：**ping**

说明：测试网络的连通性（网络地址状态为 alive 时代表网络可以正常使用）。

```
zynq-uboot> ping 192.168.1.30
Gem.e000b000:7 is connected to Gem.e000b000.  Reconnecting to Gem.e000b000
Gem.e000b000 Waiting for PHY auto negotiation to complete.... done
Using Gem.e000b000 device
host 192.168.1.30 is alive
```

命令：**printenv**

说明：打印环境变量。

命令：**reset**

说明：执行 CPU 复位。

命令：**run**

说明：运行环境变量中的命令。

命令：**saveenv**

说明：保存环境变量。

命令：**sf**

说明：SPI Flash 操作命令。

```
sf probe [[bus:]cs] [hz] [mode]     初始化指定的 Flash 设备，
                                    访问 SPI Flash 前必须先进行初始化
sf read addr offset len             读 SPI Flash 设备地址
sf write addr offset len            写 SPI Flash 设备地址
sf erase offset [+]len              擦除 SPI Flash 设备
```

```
sf update addr offset len          更新 SPI Flash 设备指定地址内容
```

命令：**sleep**

说明：睡眠功能。

命令：**source**

说明：从指定的内存地址运行一个脚本。

命令：**sspi**

说明：SPI 操作命令。

命令：**tftpboot**

说明：从网络（TFTP 协议）启动系统镜像。

命令：**version**

说明：查看 u-boot 版本信息。

1.2.3 u-boot 环境变量

u-boot 的环境变量是使用 u-boot 的关键，有一些是 u-boot 已经定义好的，有一些是用户自定义的，下表中列出了一些常用的环境变量。

表 1.1 u-boot 常用环境变量

环境变量	描述
bootdelay	执行自动启动的等候秒数
baudrate	串口控制台的波特率
netmask	以太网接口的掩码
ethaddr	以太网卡的网卡物理地址
gatewayip	网关地址
bootfile	缺省的下载文件
bootargs	传递给内核的启动参数（Linux）
bootcmd	自动启动时执行的命令
serverip	服务器端的 ip 地址
ipaddr	本地 ip 地址
stdin	标准输入设备
stdout	标准输出设备
stderr	标准出错设备

表 1.2 SylixOS 约定的环境变量

环境变量	描述
boot_sylixos_tftp	通过网络（TFTP 协议）启动 SylixOS 系统
boot_sylixos_mmc	通过 MMC 设备启动 SylixOS 系统
boot_sylixos_nand	通过 NandFlash 启动 SylixOS 系统
boot_sylixos_sf	通过 SPI Flash 设备启动 SylixOS 系统
boot_sylixos_scsi	通过硬盘启动 SylixOS 系统
boot_sylixos_u	通过 u 盘启动 SylixOS 系统

1.3 PMON 简介

PMON 是一个兼有 BIOS 和 boot loader 部分功能的开放源码软件，与 BIOS 相比功能不足，与常见的 bootloader 相比，功能要丰富的多基于龙芯的系统采用。pmon 作为类 BIOS 兼 bootloader，并做了很多完善工作。

1.3.1 pmon 启动过程

接通电源后，按主板的开机按钮，主板开始工作。根据环境变量的设置情况，PMON 启动流程稍有不同。开发板默认首先从硬盘上搜索是否存在写入的镜像，如果存在则读取内容后来启动相应的内核，如果不存在，则查找是否设了 al 环境变量，如设置了，则从 al 设置的指示来启动内核，如果没设置 al，则进入 PMON console 界面。当然，在读取 boot.cfg 之前可以按任意键来中断直接进入 PMON console 界面。

1.3.2 pmon 命令介绍

进入 PMON console 界面后，想要查找相关指令信息，通过‘h’指令，得到如图 1.1 所示指令信息：

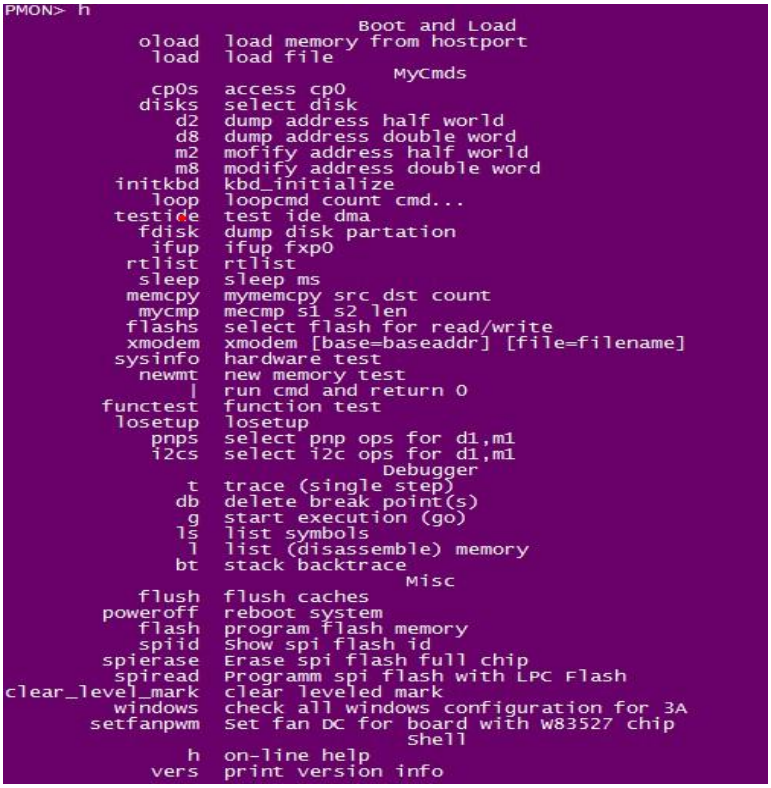

图 1.1 pmon 命令信息

如果想查看特定命令的详细说明，可以在 PMON 命令提示符下输入"h command"。 例如查看 load 命令，如图 1.2 所示格式：

```
PMON>h load
```

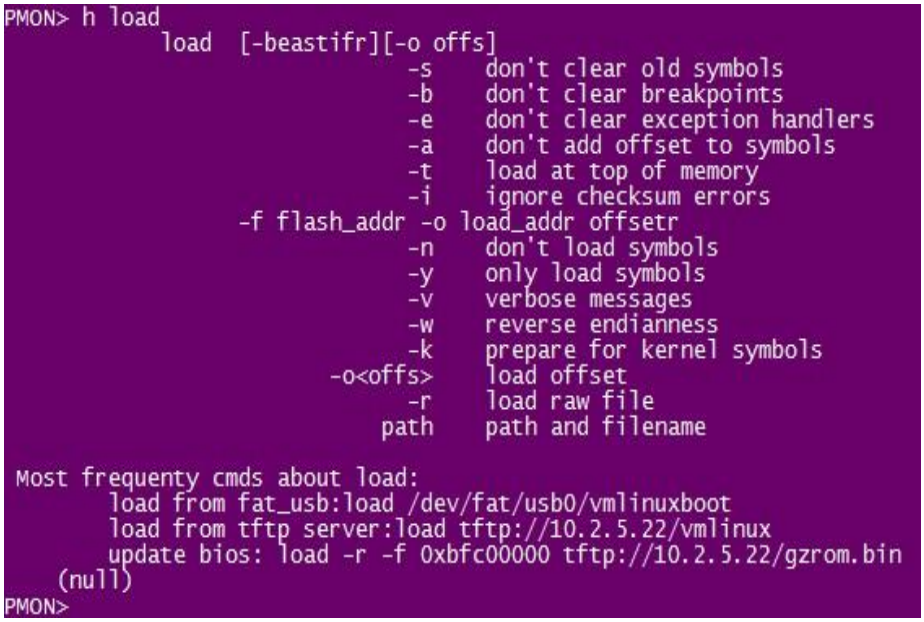

图 1.2 查看 load 指令

1.load 命令：用于下载程序和数据（例如从硬盘、优盘和 tftp 服务器等下载内核），也可以用来升级 PMON，支持 fat 和 ext2 文件格式。因此 load 可以实现多种启动方式。例如：

- tftp 启动

```
ifconfig eth0 192.168.1.85;
load -r -o80200000 tftp://192.168.1.30/bspls2hhfg.bin; flush -di;g -e 80200000;
```

其中 eth0 为开发板使用的接口名，mips 架构的开发板接口名有多种，常用的 em0（龙芯 3a3000）、syn0（龙芯 1 系列）等等，紧跟其后的接口 IP 需要设置为与镜像所在主机的 IP 在同一个网段，才能通过 tftp 协议传输。

```
-r: load raw file //想要通过 tftp 传输的文件;
-o: load_addr 偏移地址;
tftp://192.168.1.30: 192.168.1.30 为镜像所在主机 IP，需要打开 tftp 服务器;
bspls2hhfg.bin: 镜像文件名;
flush -di: 将 I-cache 和 D-cache 的内容刷新到 RAM;
g -e 80200000: 执行文件起始地址。
```

- USB 手动启动

```
load -r -o80200000 /dev/fs/fat@usb0/bsplsevm.bin;flush -di;g -e 80200000;
```

dev/fs/fat@usb0/bsplsevm.bin：镜像在 USB 下的路径。

- 硬盘手动启动

```
load -r -o80200000 /dev/fs/fat@wd0/bsplsevm.bin;flush -di;g -e 80200000;
```

dev/fs/fat@usb0/bsplsevm.bin：镜像在硬盘中的路径。

- 硬盘自动启动

```
set all "-r -o80200000 /dev/fs/fat@wd0/bsplsevm.bin;flush -di;g -e 80200000"
```

设置环境变量，该功能只适合可以保存环境变量的嵌入式平台。

2.reboot 命令，重启开发板，格式：

```
pmon > reboot
```

3.date 命令：用于查看或者设置时间。格式：

```
pmon >date [yyyymmddhhmm.ss]
```

yyymmddhhmm.ss 的格式是年月日时分秒

4. ifconfig 命令：用于设置 IP 地址。格式：

```
pmon >ifconfig  rtl0 xxx.xxx.xxx.xxx
```

5.ping 命令：用于确定本地主机是否能与另一台主机交换（发送与接收）数据报。格式：

```
pmon >ping xxx.xxx.xxx.xxx
```

6.set 命令：用来显示和设置环境变量。格式：

```
pmon >set 不加参数的可以查看所有设置的变量
pmon >set al string 用来设置 PMON 自动启动的变量
pmon >set ifconfig rtl0:xxx.xxx.xxx.xxx 设置 IP 地址
```

7.unset 命令：取消 set 命令设置的变量。格式：

```
pmon >unset al
```

8.setmac 命令：设置 MAC 地址(MAC 地址是指介质访问控制（Media Access Control，简称 MAC)地址),如果不带参数则查看当前 MAC 地址；本命令直接修改 8139 网卡的 ROM 内容，重新启动后生效。格式：

```
pmon >setmac xx:xx:xx:xx:xx:xx
```

1.3.3 pmon 环境变量

在 pmon 界面输入 env 指令可以查看环境变量信息，环境变量中有以下三种信息：

```
ethaddr = 00:00:00:00:00:00
al = /dev/fs/fat@wd0/boot/sylixos
al1 = /dev/fs/fat@wd0/boot/sylixos
```

ethaddr：MAC 地址，通过 setmac 可以设置；

al、al1：都是硬盘启动方式，通过 set al/al1 “xxxxx”来设置，优先搜索 al 变量来启动，而后搜索 al1 启动。

第 2 章 SylixOS 启动指南

不同平台的 Base 工程、Bsp 工程都是不同的，可以在翼辉信息新员工培训流程处找到相关获取方法。

2.1 制作启动盘

2.1.1 X86 制作启动盘

U 盘启动：在 RealEvo-IDE →Tools → RealEvo-SylixOS-Installer →平台（x86）、磁盘（usb）、SylixOS 镜像路径（BSP 工程下的 bspx86.elf）→ 一键安装 SylixOS → 完成启动盘的制作。

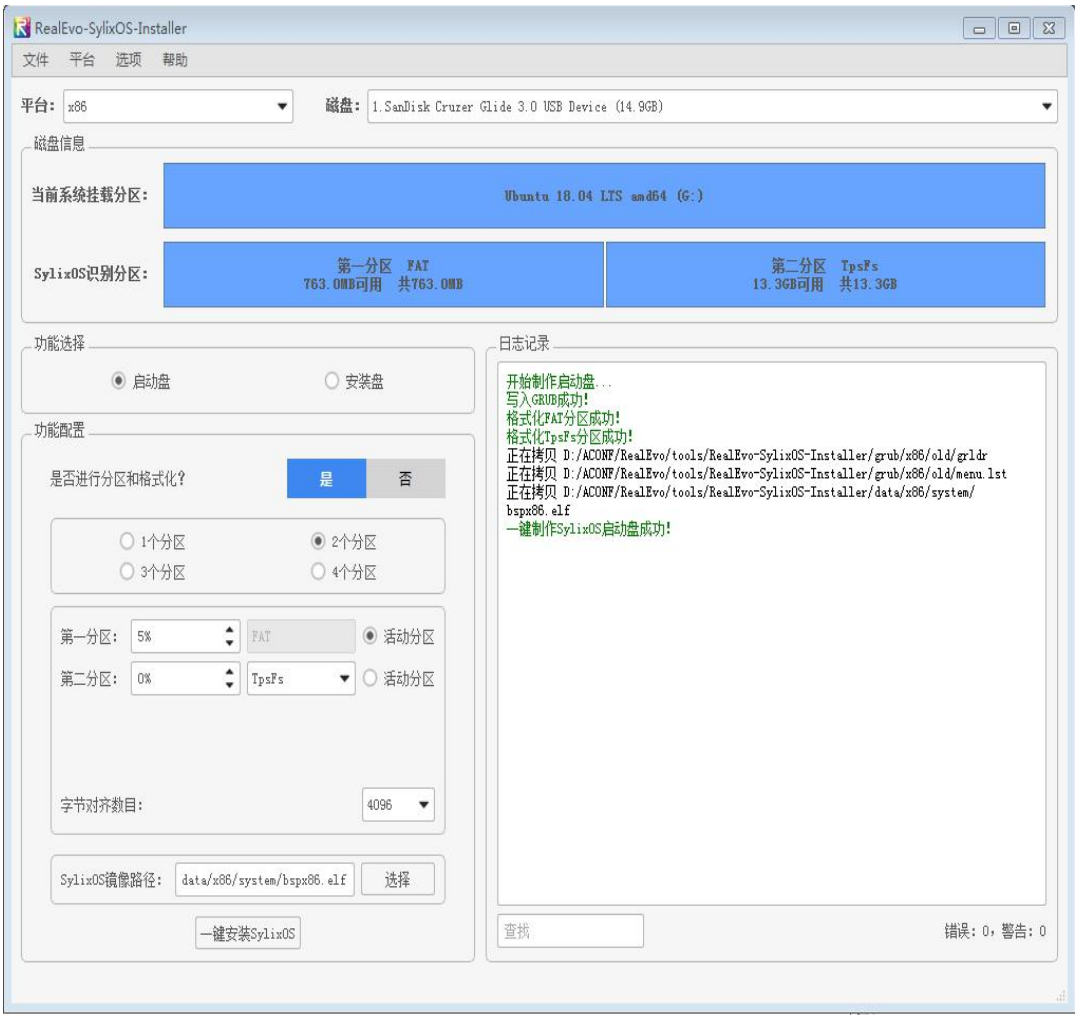

图 2.1 一键制作启动盘

启动盘里有：menu.lst、grldr、bspx86.elf。

2.1.2 AM335x 制作启动盘

(1) 第一次使用 SD 卡，必须要用 HPUSBFW 软件进行格式化，否则，am33x 开发板的 SD 卡槽无法识别。格式化过程如下图所示：（软件位置：测试事业部 → tool →am335x）

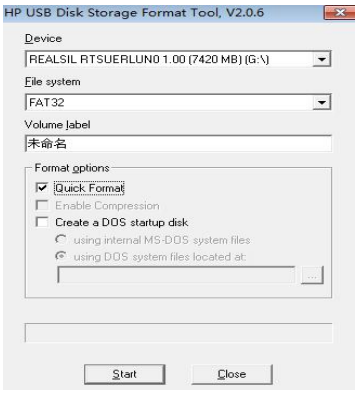

图 2.2　格式化启动盘

注意：格式化 SD 卡操作没必要每次都进行，一般一张卡进行一次格式化后，可以使用很长一段时间，只有烧写失败的情况下，可以尝试格式化 SD 卡。

（2）格式完 SD 卡，拷贝 MLO、u-boot、bspok335xs.bin 到 SD 卡中。完成了 SD 卡启动盘的制作。(MLO、u-boot 的位置：测试事业部 → tool → am335x → ok335xs;Bspok335xs.bin 的位置：在外网 GIT 上下载源码，在 RealEvo-IDE 进行编译）

2.2　制作安装盘

2.2.1　x86 制作安装盘

U 盘启动：在 RealEvo-IDE →Tools → RealEvo-SylixOS-Installer →平台（x86）、磁盘（usb）、SylixOS 镜像路径（BSP 工程下的 bspx86.elf）→ 一键安装 SylixOS → 完成启动盘的制作。

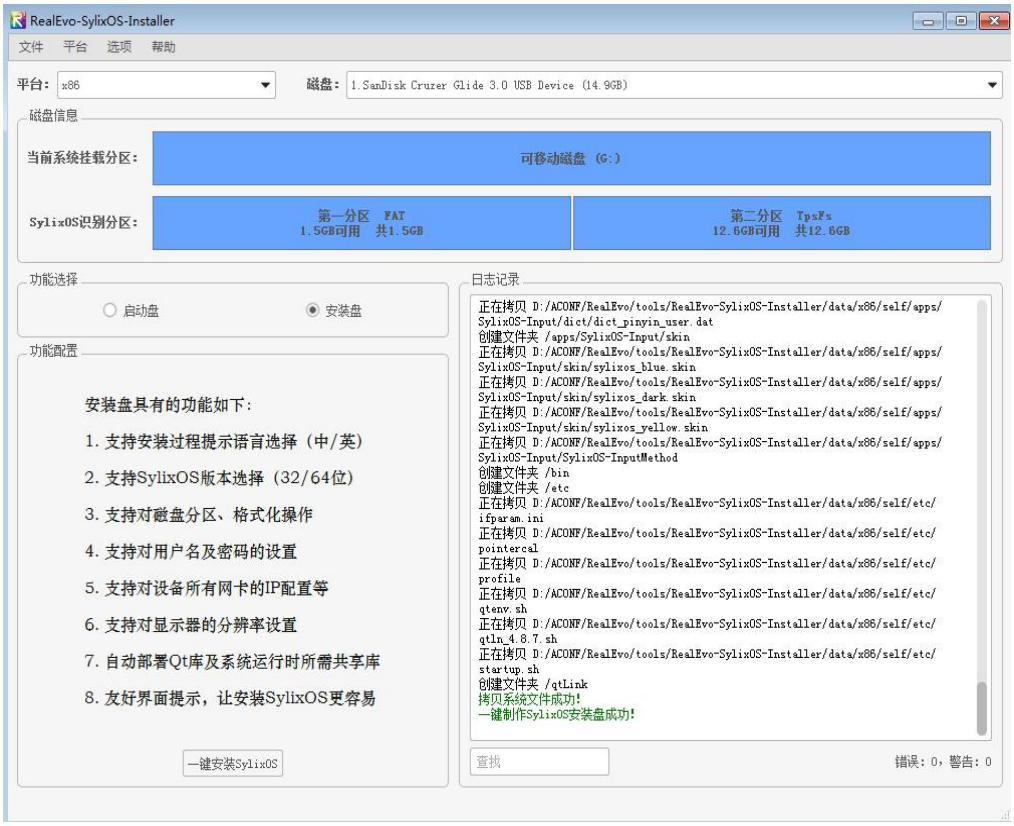

图 2.3　一键制作安装盘

安装盘里有：menu.lst、grldr、bspx86.elf、x86_installer_U.elf。

2.3 切换设备启动

2.3.1 x86 切换 U 盘启动

长按 delete → BIOS 界面 → Boot → USB(u 盘启动)。

2.3.2 AM335x 切换 SD 卡启动

SD 卡启动设置：按下启动按钮，然后给开发板上电，当听到蜂鸣器‘滴’的一声松开按钮。

注：以上启动方式适用于 OK335xS V1.2 及以后版本，如果您使用的底板是 OK335xS V1.1 版本，SD 卡启动方式是将拨码开关拨到 ON 位置然后上电，当听到‘滴’的一声后立即将拨码开关拨到 OFF 的位置。

第 3 章 SylixOS 启动方式

启动时可以参考不同平台 BSP 源码工程下的 readme.txt,不同的 BSP 工程下名字可能不同，直接找 .txt 文件。里面有启动方式和相关命令。

3.1 SylixOS 通过网络启动

在 SylixOS 开发阶段，为了能够快速方便地运行系统，通常通过网络的方式来启动 SylixOS，配置过程如下：

1. 主机端配置

主机（PC 机）需要启动 TFTP 服务器程序来启动 TFTP 传输协议，在 SylixOS 配套的开发套件(www.acoinfo.com 可以申请试用版)RealEvo-IDE 中集成了 TFTP 服务器，通过“Tools” -> “TFTP server”进行配置，如图 1.3 所示。

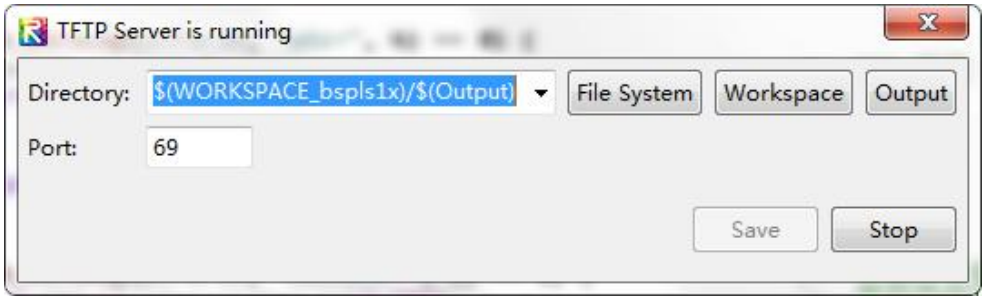

图 3.1 TFTP 服务器配置

2. u-boot 客户端配置

客户机端（运行 u-boot 的嵌入式设备）一个新的 u-boot 需要配置下面的环境变量来对网络传输的支持（TFTP 协议）：

- setenv ethaddr 12:23:34:45:56:67;
- setenv ipaddr 192.168.1.111（设置嵌入式设备的 IP 地址）;
- setenv gatewayip 192.168.1.1;
- setenv serverip 192.168.1.30（设置主机端的 IP 地址）;
- setenv loadaddr 0x10000000（设置程序的加载地址，不同的平台地址不同）;
- setenv bootfile bspimx6.bin（设置 SylixOS bsp 程序）。

以上环境变量设置成功后，可以通过 saveenv 命令将环境变量保存到相应的存储设备中，需要注意，有些板载的 u-boot 不支持 saveenv 命令，这种情况需要在每次启动都要进行一次设置。

执行完以上步骤后，我们可以通过 ping 命令来验证网络是否已经连通，如下：

```
u-boot> ping ${serverip}
host 192.168.1.30 is alive
```

显示“is alive”代表网络已经连通，执行 tftpboot 命令即可将 SylixOS bsp 程序下载到嵌入式设备，如下：

```
U-Boot > tftpboot
```

```
FEC: Link is Up 796d
Using FEC0 device
TFTP from server 192.168.1.30; our IP address is 192.168.1.111
Filename 'bspimx6.bin'.
Load address: 0x10000000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##################################################################
         ##############################################################
done
Bytes transferred = 2639528 (2846a8 hex)
```

下载完成后，执行 go ${loadaddr}将启动 SylixOS 系统。

以上过程需要每次都要执行 tftpboot 和 go 命令的操作，为了能够实现自动启动的目的，我们需要增加 boot_sylixos_tftp 和 bootcmd 环境变量的设置，如下：

```
setenv boot_sylixos_tftp 'tftpboot; go ${loadaddr}'
setenv bootcmd 'run boot_sylixos_tftp'
saveenv
```

注：自动启动只适用于能够保存环境变量的嵌入式平台。

3. pmon 客户端配置

pmon 客户端是指运行 pmon 的嵌入式设备，一个新的 pmon 需要设置与镜像所在主机在同一网段的 IP，并且通过 load 指令实现 tftp 协议的传输，前提是主机的 tftp 服务器必须打开。如图 1.3 所示：

```
ifconfig eth0 192.168.1.85;
load -r -o80200000 tftp://192.168.1.30/bspls2hhfg.bin; flush -di;g -e 80200000;
```

其中 eth0 为开发板使用的接口名，mips 架构的开发板接口名有多种，常用的 em0（龙芯 3a3000）、syn0（龙芯 1 系列）等等，紧跟其后的接口 IP 需要设置为与镜像所在主机的 IP 在同一个网段，才能通过 tftp 协议传输。

```
-r: load raw file //想要通过 tftp 传输的文件;
-o: load_addr 偏移地址;
tftp://192.168.1.30: 192.168.1.30 为镜像所在主机 IP，需要打开 tftp 服务器;
bspls2hhfg.bin: 镜像文件名;
flush -di: 将 I-cache 和 D-cache 的内容刷新到 RAM;
g -e 80200000: 执行文件起始地址。
```

pmon 客户端会收到通过 tftp 协议传输来的内核镜像，并通过引导程序启动系统。

3.2 SylixOS 通过 SD 卡启动

3.2.1 u-boot SD 卡启动 SylixOS

u-boot 希望通过 SD 来启动 SylixOS，需要配置环境变量来将镜像写入 SD，并且 u-boot 自动从 SD 启动 SylixOS（zynq7000-zturn 板子为例）。

```
setenv bootfile bspzturn.bin
setenv loadaddr 0x02800000
setenv bootcmd_sylixos_mmc "mmc dev 0; mmc rescan; fatload mmc 0 ${loadaddr}
${bootfile}; go ${loadaddr}"
setenv bootcmd "run bootcmd_sylixos_mmc"
```

其中的 mmc 是一种内存设备。

mmc rescan：探测卡的类型并且根据 mmc/sd/sdio 协议进行 comm 的初始化。

注：有的 UBOOT 版本在设置多条语句命令时，多语句之间不能用双引号，只能单引号，如：

```
  setenv bootcmd_sylixos_mmc 'mmc dev 0; mmc rescan; fatload mmc 0 ${loadaddr}
${bootfile}; go ${loadaddr}'
```

3.2.2 pmon SD 卡启动 SylixOS

如果 pmon 客户端支持 SD 卡启动 SylixOS，可以将内核镜像写入 SD 卡，写入方法常见的是通过 tftp 协议上传而后输入 sync 指令写入 SD 卡。在 pmon 模式下可以手动输入指令：

```
load -r -o80200000 /dev/fs/fat@sdcard0/bspls1x.bin;flush -di;g -e 80200000;
```

也可以设置环境变量自动启动，无需进入 pmon 模式：

```
set al "-r -o80200000 /dev/fs/fat@sdcard0/bspls1x.bin;flush -di;g -e 80200000;"
```

3.3 SylixOS 通过硬盘启动

3.3.1 硬盘启动 SylixOS（龙芯 3A）

如果 pmon 客户端支持硬盘启动 SylixOS，那么通过 FlieZilla 软件将 bspls1x.bin 从本地站点上传远程站点。设备就能加载 bspls1x.bin。然后可以将内核镜像写入硬盘，写入方法常见的是通过 ftp 协议上传而后输入 sync 指令写入硬盘。在 pmon 模式下可以手动输入指令：

```
load -r -o80200000 /dev/fs/fat@wd0/bspls1x.bin;flush -di;g -e 80200000;
```

也可以设置环境变量自动启动，无需进入 pmon 模式：

```
set al "-r -o80200000 /dev/fs/fat@wd0/bspls1x.bin;flush -di;g -e 80200000;"
```

3.3.2 硬盘启动 SylixOS（FT-1500A v1.1）

1、 进入 FT1500a#状态下，设置环境变量。

```
FT1500a# setenv cache_off "dcache flush; dcache off; icache flush; icache off"
FT1500a# setenv make_boot_arg  "mw.l 905ffee00 12345678;mw.l 905ffe08 00000002"
FT1500a# run cache_off; run make_boot_arg; fatload scsi 0:1 0x90000000
bspft1500a.bin; go 0x90000000
```

.bin 文件是通过 FileZilla 工具上传到远程站点，再通过 sync 命令写入到相应的物理设备。

2、SylixOS 系统启动成功。

注意：通过 FileZilla 上传启动文件之前，需要修改内核启动参数字符串。把用到相关平台:/media/hdd1 改为:/dev/ram。如果用的是 FT-1500　v1.1,那么如下图所示：

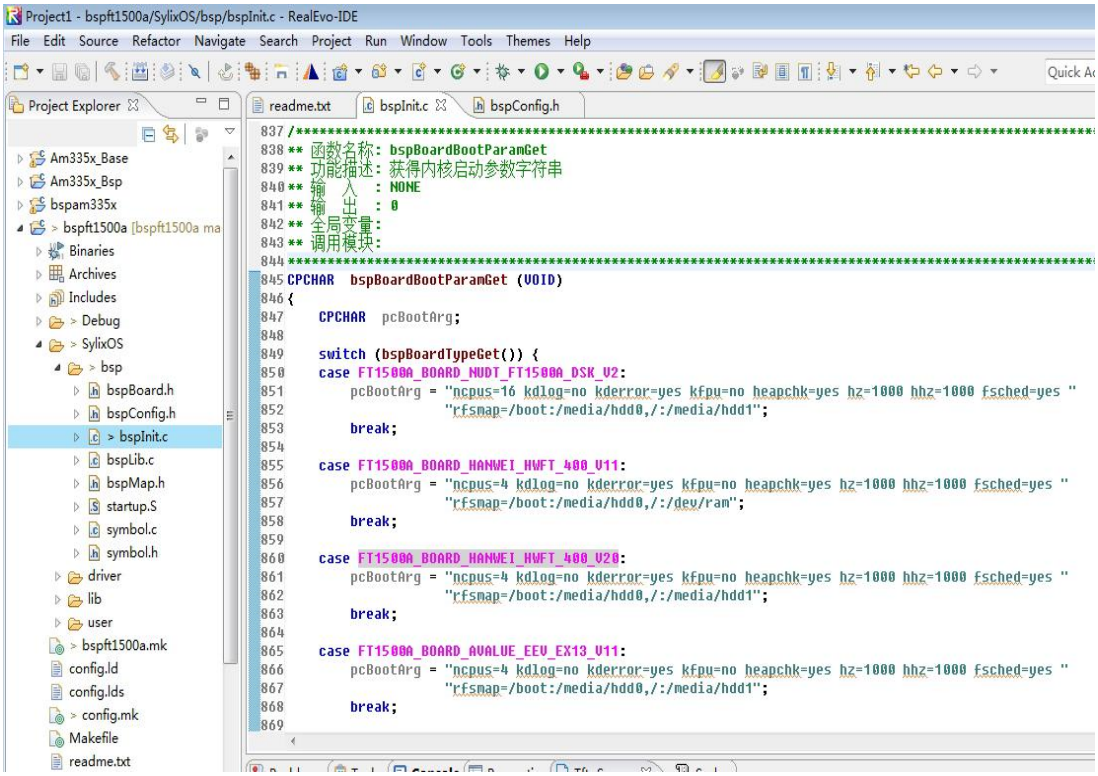

图 3.7　修改内核启动参数

FT232RL 为接口转换芯片，可以实现 USB 到串行 UART 接口的转换，也可转换到同步、异步 Bit-Bang 接口模式。

接法：RX <->TX;

TX<->RX;

GED<->GND;

3.4　SylixOS 通过 Flash 设备启动

3.4.1 固化 NandFlash（AM335x）

系统镜像固化使用如下的环境变量和和启用命令：

SD 卡根目录放入 MLO、u-boot.img、bsp$(PLATFORM).bin，开发板插入 SD 卡，从 SD 卡启动 U-Boot。

```
nand erase.chip

mmc rescan

fatload mmc 0 80A00000 MLO

nand write.i 80A00000 0 ${filesize}

fatload mmc 0 80A00000 u-boot.img

nand write.i 80A00000 800000 ${filesize}

fatload mmc 0 80A00000 bsp$(PLATFORM).bin
```

```
nand write.i 80A00000 c00000 ${filesize};
setenv bootdelay 1
setenv bootcmd 'nand read 80000000 c00000 300000; go 0x80000000'
save
boot
```

开发板上电、硬件复位、操作系统复位都将从 NAND Flash 读出 3 MiByte 系统镜像（如果系统镜像更大，需要修改上面的 300000）到内存地址 0x80000000 处运行。

注：bootdelay 不要设置为 0，否则将不能停止启动以进入 U-Boot Shell。

3.4.2 固化 SpiFlash（zynq7000）

- 固化 BootLoader 到 QSpi Flash（针对 Zynq7000 平台）

如果第一次固化，则要确保 QSpi Flash 的开始存在 BOOT.bin 文件，如果没有，则需要将 BOOT.bin 也固化到 Flash，根据实际的 BOOT.bin 的大小设置：

```
setenv qboot_addr 0x00000000
setenv boot_image BOOT.bin
setenv boot_size 200000
```

接下来，就是要读取 SD 卡里面的 BOOT.bin 读取到内存，然后再从内存读取，写入 QSPI Flash，主要使用以下两个命令：

```
fatload <interface> [<dev[:part]>]  <addr> <filename> [bytes [pos]]
sf write addr offset len
```

全部参考如下：

```
setenv qboot_addr 0x0
setenv boot_image boot.bin
setenv boot_size 0x300000
setenv qspiudp_boot "echo Update qspi boot.bin from sd card...;mmc rescan;sf probe
0 0 0;echo - Write boot.bin;fatload mmc 0 0x200000 boot.bin;sf erase ${qboot_addr}
${boot_size};sf write 0x200000 ${qboot_addr} ${boot_size}"
save
```

然后执行: run qspiudp_boot。

- 通过 tftp 或 SD 卡加载 SylixOS 镜像并固化到 QspiFlash（推荐 tftp）（针对 Zynq7000 平台）

```
tftp
sf probe 0 0 0
sf erase 0x720000  0x400000
sf write ${loadaddr}  0x720000  ${filesize}
setenv boot_sylixos_spi "sf probe 0 0 0 && sf read ${loadaddr} 0x720000 ${filesize}
&& go ${loadaddr}"
setenv bootcmd "run boot_sylixos_spi"
save
```

注：!!!如果上面的操作不正确，请留意!!!

1. 有可能有的 bootloader 处理变量的方式不同，如${loadaddr}可改为：$(loadaddr) 如：

```
sf write $(loadaddr)  0x720000  $(filesize)
```

2. 有可能有的 bootloader 处理多行语句的方式不同，可将其用引号包起来，如：

```
setenv boot_sylixos_spi "sf probe 0 0 0; sf read ${loadaddr} 0x720000 ${filesize};
go ${loadaddr}"
```

3.5 SylixOS 通过配置文件启动（只针对 uboot）

3.5.1 配置文件放到 SD

1、用 FTP 上传到到/boot 目录（media/sdcard0），如图 3.8 所示：

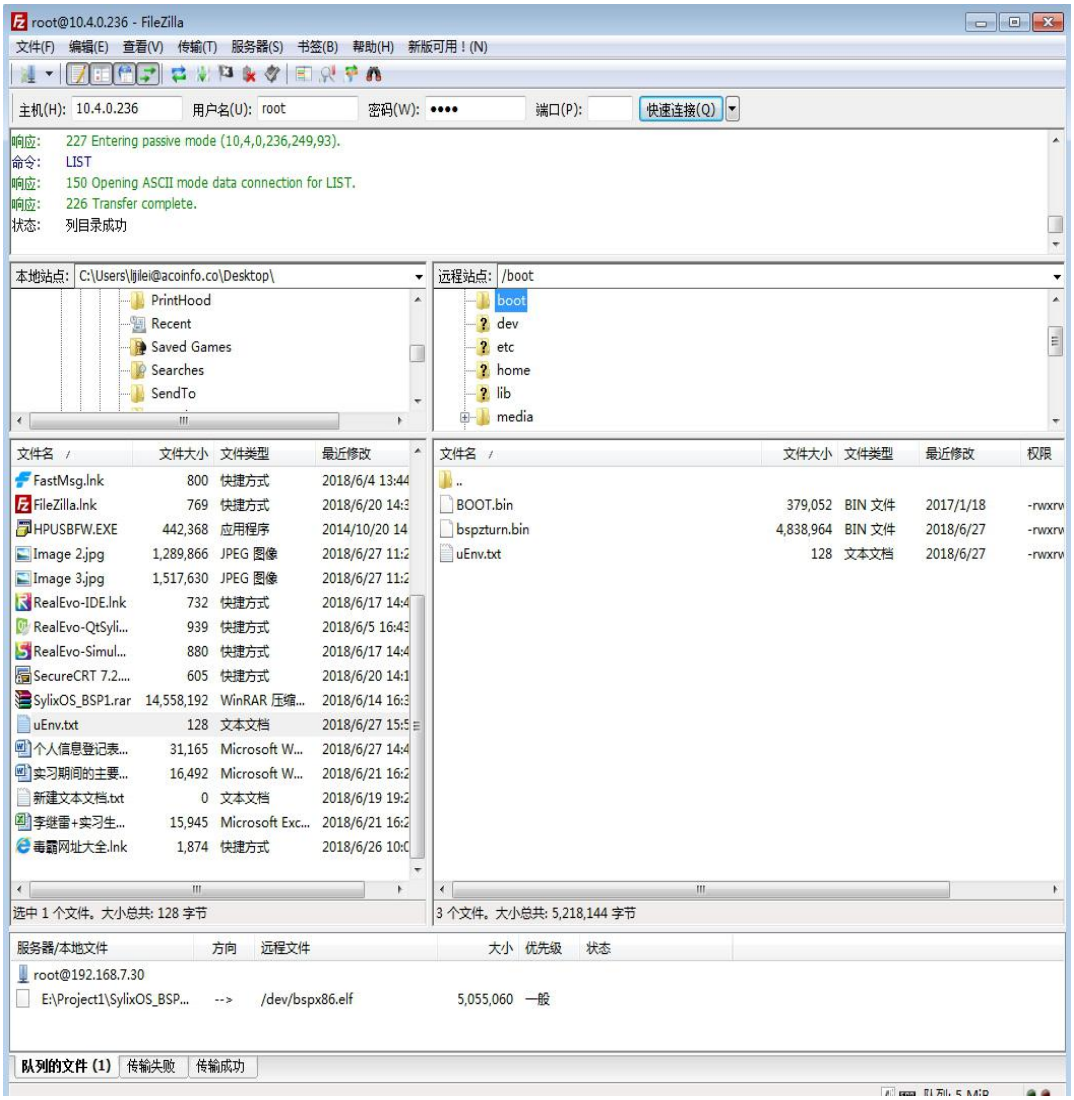

图 3.8 导入配置文件

2、读卡器读取 SD 卡 -> 拷贝到 SD 卡中。

3.5.2 引导配置文件的环境变量

以下是引导配置文件的环境变量。在 zynq-uboot 模式下，用 pri 查看这些环境变量。

```
uenvboot=if run loadbootenv; then echo Loaded environment from ${bootenv}; r
un importbootenv; fi; if test -n $uenvcmd; then echo Running uenvcmd ...; ru
n uenvcmd; fi
```

图 3.9　环境变量 uenvboot

环境变量参数：loadbootenv、bootenv、importbootenv。

```
loadbootenv=fatload mmc 0 ${loadbootenv_addr} ${bootenv}
loadbootenv_addr=0x2000000
```

图 3.10　参数 loadbootenv

```
bootdelay=3
bootenv=uEnv.txt
bootfile=bspzturn.bin
```

图 3.11　参数 bootenv

```
importbootenv=echo Importing environment from SD ...; env import -t ${loadbo
otenv_addr} $filesize
```

图 3.12　参数 importbootenv

3.5.3 启动 SylixOS

run　uenvboot 启动 SylixOS 系统。

```
zynq-uboot> setenv bootcmd 'run uenvboot'
zynq-uboot> save
Saving Environment to SPI Flash...
SF: Detected W25Q128BV with page size 256 Bytes, erase size 4 KiB, total 16 MiB
Erasing SPI flash...Writing to SPI flash...done
zynq-uboot>
zynq-uboot> run uenvboot
reading uEnv.txt
128 bytes read in 8 ms (15.6 KiB/s)
Loaded environment from uEnv.txt
Importing environment from SD ...
Running uenvcmd ...
reading bspzturn.bin
4838964 bytes read in 257 ms (18 MiB/s)
## Starting application at 0x00200000 ...
data abort
```

图 3.13　执行 uenboot

配置文件手动启动 SylixOS 系统：run uenvboot;

配置文件自动启动 SylixOS 系统：setenv bootcmd 'run uenvboot'

3.5.4 查看启动位置

查看是否从配置文件中启动。（reading uEnv.txt 说明从配置文件 uEnv.txt 中启动）

```
reading uEnv.txt
128 bytes read in 8 ms (15.6 KiB/s)
Loaded environment from uEnv.txt
Importing environment from SD ...
Running uenvcmd ...
reading bspzturn.bin
4838964 bytes read in 258 ms (17.9 MiB/s)
## Starting application at 0x00200000 ...
```

图 3.14　启动位置 uEnv.txt

3.5.5 配置文件的制作

1、 制作从 SD 卡启动的配置文件。

```
uenvcmd=setenv loadaddr 0x200000; setenv bootfile bspzturn.bin;fatload mmc 0 $mmc
0 ${loadaddr} ${bootfile}; go ${loadaddr};
```

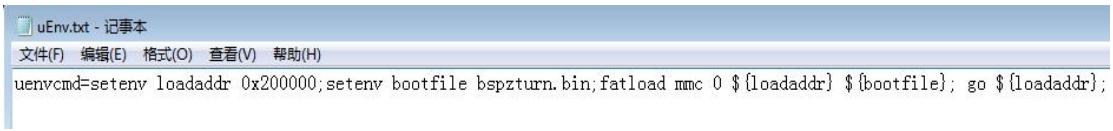

图 3.15　uEnv.txt 的制作

```
Uenvcmd=setenv loadaddr 0x80000000; setenv ipaddr 192.168.1.30;setenv bootfile
bspam4378.bin; tftp; go 0x80000000;
```

2、制作从 tftp 启动的配置文件。

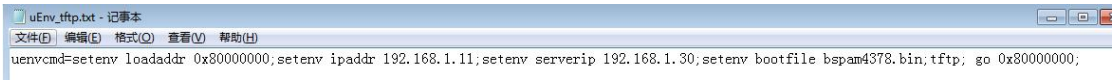

图 3.16　uEnv_tftp.txt 的制作

第 4 章 SylixOS 配置文件

4.1 starup.sh

在/etc 目录下存放着 startup.sh 启动脚本，脚本内容可以按照自己想要的信息进行添加，脚本内容最基本的内容如下（可修改）：

```
shstack 120000
modulereg /lib/modules/xsiipc.ko
modulereg /lib/modules/xinput.ko
```

- shstack 120000：设置栈的大小为 120000
- modulereg 等同于 insmod，安装驱动。

4.2 ifparam.ini

在/etc 目录下的 ifparam.ini 文件是网卡配置信息文件，可以通过设置该文件的信息来得到自己在开发板启动时想要的网卡信息，如 IP 地址、掩码等等。文件内容根据网卡不同，网卡名字也有所不同，可通过 ifconfig 指令查看网卡的网卡名，如 dw_0、e1000_0 等，内容配置大致如下：

```
[dw_0]
enable=1              //网卡使能，默认等于 1
ipaddr=10.4.100.87
netmask=255.255.0.0
gateway=10.4.0.1
default=1             //默认设备输出
mac=00:11:22:33:44:55   //MAC 地址
ipv6_auto_cfg=1       //如果 SylixOS 作为 ipv6 的路由器，值为 0
dhcp=1                //自动获取 IP 地址
```

如上所示，网卡名为 dw_0，设置其 IP 为 10.4.100.87，子网掩码为 255.255.0.0，网关为 10.4.0.1。

4.3 resolv.conf

在/etc 目录下的 resolv.conf 文件是 DNS 域名解析配置文件，可以通过设置该文件的信息来得到指定 DNS 服务器的地方。如果不指定的话，无法通过敲 www.baidu.com 来访问百度（外网），只能通过敲百度服务器的 IP 来访问。配置信息如下图所示：

```
[root@sylixos:/etc]# vi  resolv.conf
nameserver  114.114.114.114
```

DNS 就是域名解析器，当访问网站的时候，其实都是在访问一个 IP 主机，例如 www.baidu.com，敲下去的是百度的网站，但实际转换的是该主机的 IP 地址，好处是访问网

站或者说服务器时，不用记那些 IP 地址的数字串，而是记一些比较容易记的字符，如 baidu.com。而 DNS 就是把域名转换为 IP 地址的东西。

4.4 hosts

以根用户登录，或者登录切换到根用户，然后在提示符下输入 hostname 命令，可以看出当前系统的主机名为 hostname is sylixos。

```
[root@sylixos:/root]# hostname
hostname is sylixos
```

更改主机名。直接输入 hostname　sylixos1。

```
[root@sylixos:/root]# hostname sylixos1
[root@sylixos1:/root]# hostname
Hostname is sylixos1
```

第 5 章 SylixOS 环境变量

在 SylixOS 系统下，可以通过指令 env 来查看环境变量，具体信息如图 3.1（龙芯 1a）所示：

```
[root@sylixos:/root]# env
QT_QPA_FONTDIR=/qt/lib/fonts
QT_QPA_PLATFORM=sylixosfb
QT_QPA_PLATFORM_PLUGIN_PATH=/qt/plugins
QML2_IMPORT_PATH=/qt/qml
QML_IMPORT_PATH=/qt/qml
QT_QWS_FONTDIR=/qt/lib/fonts
QT_PLUGIN_PATH=/qt/plugins
POINTERCAL_FILE=/etc/pointercal
QWS_KEYBOARD=sylixosinput
QWS_MOUSE_PROTO=sylixosinput
QWS_DISPLAY=VNC:sylixosfb:/dev/fb0
QPEDIR=/qt
QTDIR=/qt
XINPUT_PRIO=199
DISPLAY=/dev/fb0
TERMCAP=/etc/termcap
TERM=vt100
LUA_CPATH=?.so;/usr/local/lib/lua/?.so;/usr/lib/lua/?.so;/lib/lua/?.so
LUA_PATH=?.lua;/usr/local/lib/lua/?.lua;/usr/lib/lua/?.lua;/lib/lua/?.lua
VPROC_EXIT_FORCE=0
LOGINBL_REP=3
LOGINBL_TO=120
DEBUG_CPU=-1
PATH_LOCALE=/usr/share/locale
LC_ALL=
LANG=C
LD_LIBRARY_PATH=/qt/lib:/qt/lib:/qt/lib:/qt/lib:/qt/lib:/usr/lib:/lib:/usr/local/lib
PATH=/usr/bin:/bin:/usr/pkg/sbin:/sbin:/usr/local/bin
NFS_CLIENT_PROTO=udp
NFS_CLIENT_AUTH=AUTH_UNIX
SYSLOGD_HOST=0.0.0.0:514
KERN_FLOAT=0
SO_MEM_PAGES=8192
TSLIB_CALIBFILE=/etc/pointercal
TSLIB_TSDEVICE=/dev/input/touch0
MOUSE=/dev/input/touch0:/dev/input/mse0
KEYBOARD=/dev/input/kbd0
STARTUP_WAIT_SEC=1
TZ=CST-8:0:0
TMPDIR=/tmp/
LICENSE=SylixOS license: Commercial & GPL.
VERSION=1.6.5
SYSTEM=SylixOS kernel version: 1.6.5 Code name: Octopus
USER=root
HOME=/root
```

图 5.1　环境变量信息

如上图所示，可知系统已经部署 QT 库，环境变量自动生成相应的信息。环境变量解析如下：

QT_QPA_FONTDIR：部署的 QT 库字体目录；

QT_QPA_PLATFORM：这个插件通过 SylixOS 的 fbdev 子系统直接写入 framebuffer；

QT_QPA_PLATFORM_PLUGIN_PATH：QT 插件所在目录；

QML2_IMPORT_PATH：一种重用 QML 类型的方式：文件系统上的目录，导入包含 QML 文档的目录到 QML 文档；

QT_QWS_FONTDIR：QT 窗口系统显示中文字体目录，存放 ttf 文件；

POINTERCAL_FILE：指定包含用于校验指针设备的文件；

QWS_KEYBOARD：QT 窗口键盘输入类型；

QWS_MOUSE_PROTO：QT 窗口鼠标输入类型；

QWS_DISPLAY=VNC:sylixosfb:/dev/fb0 ：窗口显示通过 VNC、通过/dev 下的 framebuffer；

XINPUT_PRIO：t_xinput 线程的优先级；

TERM：设置为终端机，取决于仿真类型；

TERMCAP：终端性能数据库；

LUA_CPATH：lua 依赖的库文件搜索路径；

LUA_PATH：lua 搜索路径；

VPROC_EXIT_FORCE：主线程退出自动删除子线程；

LOGINBL_REP：（FTP、Telnet 登录）连续出现几次加入黑名单；

LOGINBL_TO：FTP、Telnet 登录）黑名单刷新时间；

DEBUG_CPU：是否将被调用对象锁定在一个 CPU；

PATH_LOCALE：设置语言环境；

LANG：系统默认将 locale 定义为“c”；

LD_LIBRARY_PATH：依赖库文件路径

PATH：PATH 启动时默认路径

NFS_CLIENT_PROTO：NFS 客户端使用协议类型默认为 UDP；

NFS_CLIENT_AUTH：NFS 客户端使用 AUTH_UNIX；

SYSLOGD_HOST：syslog 服务器地址（格式：IP:PORT）

KERN_FLOAT：内核是否支持浮点；

SO_MEM_PAGES： 动态内存虚拟页面数量；

TSLIB_CALIBFILE：指定触摸屏校准文件；

TSLIB_TSDEVICE：指定触摸屏设备；

MOUSE：指定鼠标对应设备；

KEYBOARD：指定键盘对应设备；

STARTUP_WAIT_SEC：执行 startup.sh 延迟时间（s）；

TZ：时区设置，默认为东八区；

TMPDIR：临时目录挂载文件目录；

LICENSE：SylixOS 操作系统注册 license 信息；

VERSION：SylixOS 操作系统版本信息；

SYSTEM：SylixOS 操作系统系统信息；