1构建嵌入式linux系统bootloader



课程介绍

主要介绍

- 嵌入式系统结构
- 嵌入式软件开发
- Bootloader设计

一 嵌入式系统结构

嵌入式系统是一个计算机系统；

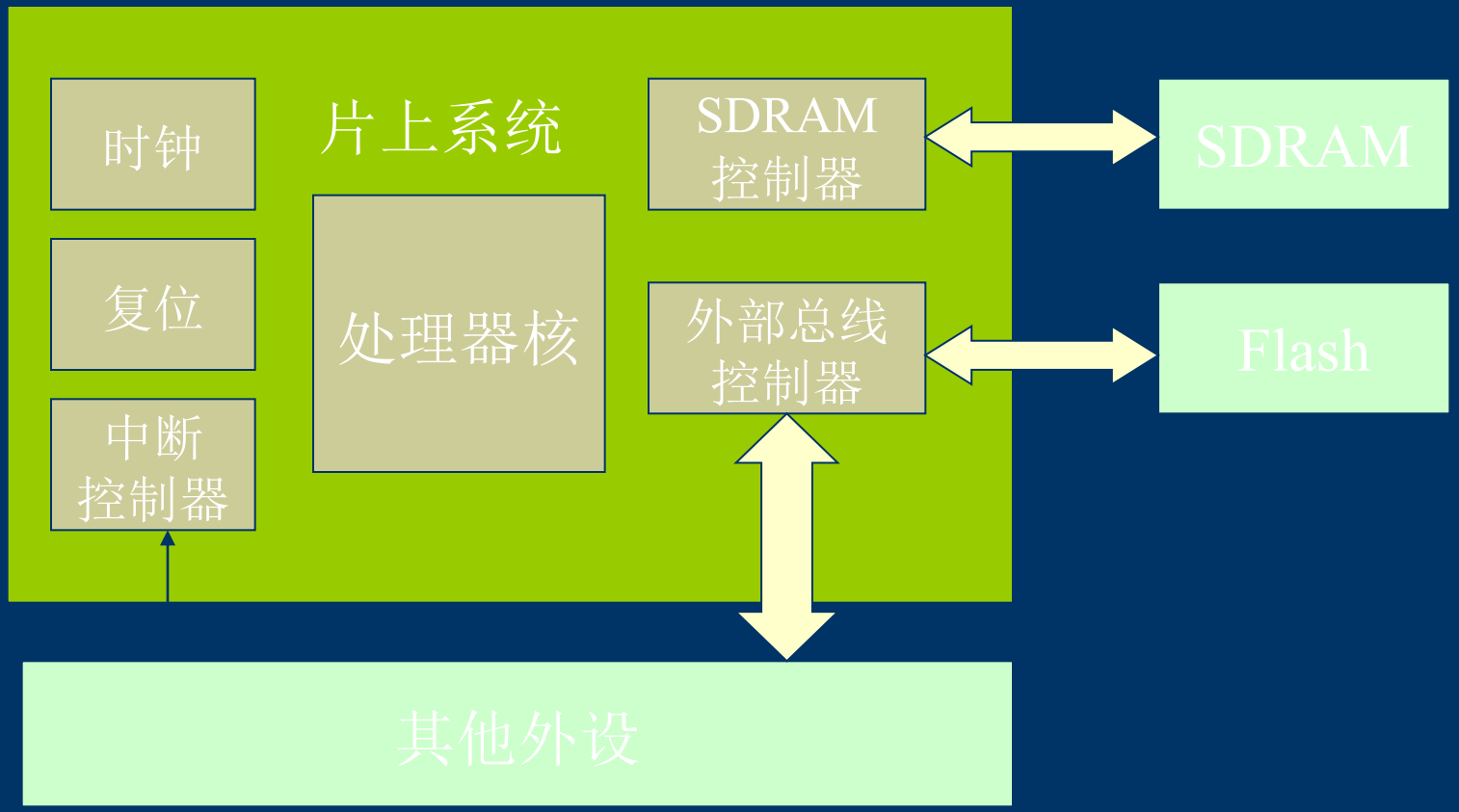
嵌入式系统一般包括硬件和软件两部分

这里的硬件是以一个高性能的处理器（通常是32位处理器）为基础；

软件是以一个多任务操作系统为基础的综合平台；

这样一个软、硬件平台是单片机无法比拟的，可称之为“嵌入式系统”。

（1）嵌入式硬件平台



（2）嵌入式系统软件

嵌入式软件可以分为两大类：

系统软件：控制和管理嵌入式系统资源，一般是内核。

应用软件：在系统软件之上，如GUI（图形界面），网络服务器软件等。

（2）嵌入式软件结构



内核软件

嵌入式软件

嵌入式软件开发

嵌入式应用开发过程

一个嵌入式应用项目的开发过程是一个硬件和软件设计的综合过程，是一个系统设计过程，主要经历以下步骤：

- 硬件设计与实现
- 引导加载程序的设计
- 操作系统的移植（主要是驱动程序移植）
- 应用程序的设计
- 系统调试，样机交付

其中，引导加载程序，操作系统移植，应用程序设计是软件设计，具有自己的特点。

嵌入式软件开发的特点

需要交叉编译

硬件平台处理器较慢，内存和外存容量小等等；

把软件开发放在高性能的PC机上进行；

PC机上CPU指令集与嵌入式CPU的指令集是不同的，因此，在PC机上开发嵌入式软件需要交叉编译。

需要移植

由于嵌入式系统是一个软硬件定制的系统，硬件平台各不相同，软件设计需要根据不同的硬件设备来添加或修改相应的代码，这就是移植。

基于linux的嵌入式软件设计

在一个硬件平台上设计一个基于linux的软件系统，需要的工作可以分为以下三部分：

Bootloader设计

Linux内核的定制移植，驱动程序的开发

应用程序的设计

根据GEC2410开发板，构建嵌入式linux系统

接下来，我们以gec2410开发板为硬件平台，介绍构建一个嵌入式linux系统的原理和过程。

系统软件有以下部分组成

- Bootloader为u-boot
- Linux内核的版本为2.6.14
- 应用程序由busybox、bash等构成

Bootloader设计

Bootloader介绍

Bootloader程序是系统加电后运行的第一段代码；

通常嵌入式CPU上电或者复位时都会跳到一个特定的地址开始执行；如：ARM是0x0地址；

通常用某种类型的固态存储设备(FLASH)映射到这个预先安排的地址上；

因此在系统加电后，CPU 将首先执行的是这个特定地址上的 Boot Loader 程序。

Bootloader依赖于硬件平台，因此bootloader程序一般都需要移植。

Bootloader程序结构

Bootloader程序的结构一般分为两个阶段：

Stage 1: 汇编语言编写，达到短小精悍的目的；

Stage 2: C语言编写，实现更加复杂的功能；

实验：结合gec2410的bootloader程序分析bootloader程序的结构；

U-boot的介绍

Uboot是德国DENX软件工程小组开发的、支持多种嵌入式CPU的bootloader程序。

可引导的嵌入式操作系统：

UBoot不仅仅支持嵌入式Linux系统的引导，当前，它还支持NetBSD, VxWorks, QNX, RTEMS, ARTOS, LynxOS 嵌入式操作系统。

支持的处理器：

UBoot支持PowerPC、MIPS、 x86、ARM、NIOS、XScale等诸多常用系列的处理器。

特点：

功能多、灵活性强、开发积极、开放源代码、属于“monitor”

U-boot的介绍

uboot的获得：

从下面地址下载uboot的源代码。

sourceforge/projects/uboot

邮件列表：

lists.sourceforge.net/lists/listinfo/u-boot-users/-boot-users/

U-boot的源码结构

- uboot的目录结构
  - 顶层有18个子目录，分别存放和管理不同的源代码。这些目录分三类：
    - 与处理器体系结构或者开发板硬件直接相关
    - 存放通用的函数或者驱动程序的目录
    - 存放uboot的应用程序、工具、文档的目录

U-boot的源码结构

uboot的目录结构

- board：存放与开发板有关的文件。每一个开发板都以一个子目录出现在当前目录中。如smdk2410
- common：实现uboot支持的命令，每一条命令都对应一个文件。例如bootm命令 -> cmd_bootm.c。
- cpu：与特定CPU架构相关目录，每一款Uboot下支持的CPU在该目录下对应一个子目录，比如有子目录arm920t等。
- doc：文档目录。Uboot有非常完善的文档，推荐大家参考阅读。
- drivers：Uboot支持的设备驱动程序都放在该目录，比如各种网卡、串口和USB等。

U-boot的源码结构

uboot的目录结构

fs: 支持的文件系统，Uboot现在支持cramfs、fat、fdos、jffs2和registerfs。

lib_xxxx: 与体系结构相关的库文件。如与ARM相关的库放在lib_arm中。

net：与网络协议栈相关的代码，BOOTP协议、TFTP协议、RARP协议和NFS文件系统的实现。

tools：生成Uboot的工具，如：mkimage, crc等等

问题

问题：u-boot支持多种体系结构的CPU,那么怎么在配置编译u-boot的时候选择自己的平台呢？

U-boot的配置编译

U-Boot 的源码是通过 GCC 和 Makefile 组织编译的。

首先顶层 Makefile 设置开发板的定义；

然后递归地调用各级子目录下的 Makefile；

最后把编译生成的目标文件链接成U-Boot 映像。

U-boot的配置编译

顶层目录下的 Makefile

负责U-Boot整体配置编译。

每一种开发板在 Makefile 都需要有板子配置的定义。

```
smdk2410_config :   unconfig
    @./mkconfig $(@:_config=) arm arm920t smdk2410 NULL s3c24x0
```

执行配置U-Boot的命令make smdk2410_config，则通过./mkconfig脚本生成include/config.mk 的配置文件。实际上是定义了ARCH、CPU、BOARD、SOC这些变量

```
ARCH    = arm
CPU     = arm920t
BOARD   = smdk2410
SOC     = s3c24x0
```

U-boot的配置编译

顶层目录下的 Makefile

Makefile文件开始的部分包含了这些变量的定义和输出。

```
# load ARCH, BOARD, and CPU configuration
include include/config.mk
export          ARCH CPU BOARD VENDOR SOC
```

U-boot的配置编译

- 顶层目录下的 Makefile
  - 硬件平台依赖的目录文件就是根据这些定义来确定。SMDK2410平台相关目录如下。
    - board/smdk2410/
    - cpu/arm920t/
    - cpu/arm920t/s3c24x0/
    - lib_arm/
    - include/asm-arm/
    - include/configs/smdk2410.h

U-boot的配置编译

顶层目录下的 Makefile

Makefile文件还定义uboot映象编译的依赖关系。

```
ALL = u-boot.srec u-boot.bin System.map
all:            $(ALL)
u-boot.hex:     u-boot
                $(OBJCOPY) ${OBJCFLAGS} -O ihex $< $
u-boot.srec:    u-boot
                $(OBJCOPY) ${OBJCFLAGS} -O srec $< $
u-boot.bin:     u-boot
                $(OBJCOPY) ${OBJCFLAGS} -O binary $< $
```

……

Makefile缺省的编译目标是all，包括：

u-boot.srec

u-boot.bin

System.map

U-boot的配置编译

开发板配置头文件

编译之前，可通过修改头文件include/configs/<board_name>.h，为开发板定义配置选项或者参数。例如smdk2410板的头文件是smdk2410.h。

这个头文件主要定义了两类变量

一类是选项，前缀是CONFIG_，用来选择处理器、设备接口、命令、属性等。

```
#define CONFIG_DRIVER_CS8900          1
```

另一类是参数，前缀是 CFG_，用来定义总线频率、串口波特率、Flash 地址等参数。

```
#define PHYS_FLASH_1          0x00000000
#define CFG_FLASH_BASE    PHYS_FLASH_1
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/218134047074006047