

君正 uCOS-II 开发手册

Revision: 1.0
Date: June 2007



北京君正集成电路有限公司
Ingenic Semiconductor Co. Ltd

君正 uCOS-II 开发手册

Copyright © Ingenic Semiconductor Co. Ltd 2006. All rights reserved.

Release history

Date	Revision	Change
June. 2007	1.0	First release

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co. Ltd

**E801C, Power Creative Building,
No.1, Shangdi East Road,
Haidian District, Beijing 100085, China
Tel: 86-10-58851002
Fax: 86-10-58851005
Http: //www.ingenic.cn**

内容

1	概述.....	1
2	开发环境.....	3
2.1	安装交叉编译工具链.....	3
2.2	启动 TFTP 服务.....	3
3	uCOS-II 结构和编译.....	5
3.1	uCOS-II 的目录结构.....	5
3.2	配置和编译 uCOS-II.....	7
4	uCOS-II 驱动的使用和测试.....	11
4.1	网络驱动.....	11
4.2	摄像头驱动.....	12
4.3	UDC 驱动.....	13
4.4	LCD 驱动.....	13
4.5	音频驱动.....	14
4.6	触摸屏驱动.....	14

1 概述

JZ4730 和 JZ4740 是北京君正推出的两款高性能低功耗的 MIPS 内核 32 位嵌入式处理器，主频 350MHz，支持 MMU，包括 16KB 的 I-Cache 和 16KB 的 D-Cache，同时还集成了大量的外围设备。在君正处理器平台上可以运行 Linux、WinCE 和 uCOS-II 等操作系统，适合用来开发个人数字助理（PDA）、便携媒体播放器（PMP）、卫星导航仪（GPS）、数码学习机/电子辞典、数码像框、电子书和智能手机等多媒体和通信设备。

本文将向读者详细介绍基于君正处理器平台进行 uCOS-II（2.83）开发的过程和方法，引导开发人员快速进行 uCOS-II 开发，包括建立交叉编译环境、uCOS-II 的配置和编译、设备驱动的使用和测试等等。

2 开发环境

2.1 安装交叉编译工具链

在君正处理器 JZ4730/JZ4740 平台上利用 uCOS-II 进行开发之前,首先需要安装好 MIPS 的交叉编译环境。针对 uCOS-II 开发,君正提供基于 GNU GCC 3.3.1 的 MIPS 交叉编译工具链,可分别安装在 Windows 和 Linux 主机上。Windows 上的交叉编译工具链需要在 Cygwin 下使用,因此,在 Windows 上进行开发前,请先安装好 Cygwin 环境。

在 Windows 主机上安装 MIPS 交叉编译工具的步骤如下:

首先,安装和配置好 Cygwin 环境。请从网上下载 Cygwin 的相关软件包和资料,并参考 Cygwin 的文档进行安装。当安装好 Cygwin 后,启动 Cygwin 的 Shell 程序,创建一工作目录,并把工具链包解压到该目录下,然后设置好环境变量。具体步骤如下:

```
$ mkdir -p /opt
$ cd /opt
$ tar -xjf mipsel-jz-cygwin-nopic.tar.bz2
$ export GCC_EXEC_PREFIX=/opt/mipsel-nopic/lib/gcc-lib/
$ export PATH=/opt/mipsel-nopic/bin:$PATH
```

在 Linux 主机上安装 MIPS 交叉编译工具的步骤如下:

创建一工作目录,并把工具链包解压到该目录下,然后设置好环境变量。具体步骤如下:

```
$ mkdir -p /opt
$ cd /opt
$ tar -xjf mipsel-jz-linux-nopic.tar.bz2
$ export GCC_EXEC_PREFIX=/opt/mipsel-nopic/lib/gcc-lib/
$ export PATH=/opt/mipsel-nopic/bin:$PATH
```

按照上面步骤建立好交叉编译环境后,可以编译简单的 helloworld.c 测试一下:

```
$ mipsel-linux-gcc -o helloworld helloworld.c
```

如果编译通过,说明刚刚安装的交叉编译工具链可以工作了。

2.2 启动 TFTP 服务

TFTP 是用来下载远程文件的网络传输协议,引导程序 U-Boot 支持 TFTP 下载功能。在开发阶段,如果主机启动了 TFTP 服务,引导程序 U-Boot 就可以通过 TFTP 下载 uCOS-II 到目标板运行,这样将极大的方便用户进行开发。因此,建议用户在用来开发的 Linux 或者 Windows 主机上启动 TFTP 服务。具体方法请参考 Ingenic JDI 用户手册 (JDI-Manual_CN_1.2.pdf)。

3 uCOS-II 结构和编译

君正移植了 uCOS-II 内核和主要的设备驱动程序，可运行在君正处理器的各种开发平台上。本文将具体描述 uCOS-II 的结构、配置、编译以及驱动程序的使用和测试。

3.1 uCOS-II 的目录结构

- mips/: MIPS 体系相关文件目录
 - os_cpu_a.S、os_cpu.h、os_cpu_c.c: uCOS-II 移植接口文件
 - os_dbg.c: uCOS-II 内核调试相关程序，通过设置 os_cfg.h 中的 OS_DEBUG_EN 值决定是否调试
 - includes.h: uCOS-II 应用程序及其内核文件都应该包含的头文件
 - context.h: 任务切换时，MIPS 寄存器压栈和出栈的宏函数定义文件
 - libc_s.S、libc.c: 针对 MIPS 的标准库函数文件
 - mipsregs.h、mips.h、archdefs.h、sysdefs.h: MIPS 体系相关宏定义
- jz4730/: JZ4730 处理器相关目录和文件
- include/: JZ4730 处理器相关头文件目录
 - reg.h: JZ4730 各种外设的寄存器地址的宏定义文件
 - ops.h: 对 JZ4730 各种外设的寄存器操作的宏定义文件
 - frmwrk.h: 全局宏定义文件
 - clock.h: 其中定义了检测 JZ4730 各类时钟的内联函数
 - udc/: USB 设备控制器驱动目录
 - camera/: 摄像头驱动目录
 - head.S: mips 初始化及异常、中断入口程序
 - init.c: JZ4730 片上外设初始化等程序
 - cache.c: CACHE 操作代码
 - excpt.c: 异常处理（简单地将所有通用寄存器的值打印输出以便调试）程序
 - gpio.c: 针对 PMP2.2 板的 GPIO 初始化程序
 - intc.c: 中断管理程序
 - uart.c: 串口驱动程序
 - clock.c: 时钟检测及 udelay() 代码
 - timer.c: uCOS-II 操作系统时钟相关程序
 - sleep.S: JZ4730 休眠/唤醒管理程序
 - pm.c: 时钟/电源管理程序
 - main.c: 主函数程序文件
 - makefile: makefile 文件
 - link.xn: 链接脚本文件
 - DrvMemMgr.c 驱动层内存分配管理
 - nandboot/: NandBoot Nand 引导程序
 - sample/: 一些驱动调试例程
 - drv/: 各驱动目录
 - codec/: 音频驱动

- lcd/: lcd 驱动
- touch/: 触摸屏驱动
- i2c/: I2C 控制器驱动
- jz4740/: JZ4740 处理器相关目录和文件
- include/: JZ4740 处理器相关头文件目录
 - Jz4740.h: JZ4740 各种外设的寄存器地址的宏定义文件
 - bsp.h: pavo 板级相关定义
 - udc/: USB 设备控制器驱动目录
 - head.S: mips 初始化及异常、中断入口程序
 - init.c: JZ4730 片上外设初始化等程序
 - cache.c: CACHE 操作代码
 - excpt.c: 异常处理（简单地将所有通用寄存器的值打印输出以便调试）程序
 - gpio.c: 针对 PMP2.2 板的 GPIO 初始化程序
 - intc.c: 中断管理程序
 - uart.c: 串口驱动程序
 - clock.c: 时钟检测及 udelay()代码
 - timer.c: uCOS-II 操作系统时钟相关程序
 - sleep.S: JZ4730 休眠/唤醒管理程序
 - pm.c: 时钟/电源管理程序
 - main.c: 主函数程序文件
 - makefile: makefile 文件
 - link.xn: 链接脚本文件
 - DrvMemMgr.c 驱动层内存分配管理
 - nandboot/: NandBoot Nand 引导程序
 - sample/: 一些驱动调试例程
 - drv/:各驱动目录
 - codec/: 音频驱动
 - lcd/: lcd 驱动
 - touch/: 触摸屏驱动
 - key/: 键盘驱动
 - rtc/: rtc 驱动
- lwip/: 网络相关文件目录
- src/: uCOS-II 内核源码目录
 - app_cfg.h: uCOS-ii 2.81 版之后都需要在应用程序中包含该头文件，其中定义了各任务的优先级、堆栈的大小与其它应用相关的配置信息
 - os_tmr.c: uCOS-ii 2.81 新添的定时器管理程序
 - 其它文件的介绍略。
- sample/: 主函数例程目录
 - main_lwip.c: 网络驱动例程
 - main_fs.c: uc/FS 文件系统例程
 - main_mmc.c: 基于 MMC/SD 卡的 uc/FS 文件系统例程
 - main_os.c: 最基本的 uCOS-ii 例程

- mmc/: MMC/SD 卡驱动目录
- ucfs/: uc/Fs 文件系统（基于 MMC/SD 卡）目录
- audio/: wav 解码例程
- madplay: mp3 解码例程

3.2 配置和编译 uCOS-II

在一工作目录下解开 uCOS-II 的源码：

```
$ tar -xjf ucossii-jz-yyyymmdd.tar.bz2
```

解压后可见源码根目录为 ucossii。（因为目前 uCOS-II 源码包较小，所以新版本的 uCOS-II 不发布补丁，直接发布全部的源码。）

现在就可以配置和编译 ucossii 了。

通过简单地编辑 ucossii/jz4730/makefile，可以配置哪些驱动参与编译。makefile 文件的开头部分放置了驱动编译选项，JZ4730 如下：

```
CAMERA = 0           CIM 控制驱动
LWIP    = 0           网络驱动
UCGUI   = 1           UCGUI
UCFS    = 1           FAT 文件系统
MMC     = 1           SD 驱动
JPEG    = 1           JPEG 解码
TOUCH   = 0           触摸屏驱动
NAND    = 0           NAND 驱动
MP3     = 1           MP3 解码
I2C     = 1           I2C 控制器驱动
#LCDTYPE :
#         0. No LCD Driver
#         1. Samsung tft 480 * 272
#         2. Truly-tft-g320*240drsw
#         3. FoxConn_pt035tn01
LCDTYPE = 1
#CODECTYPE :
#         0. No Codec Driver
#         1. jz4740 Codec
#         2. akm4642
#         3. ac97
CODECTYPE = 2
#wip
LWIP30 = 0
```

CAMERA 代表摄像头驱动，LWIP 代表网络驱动，UCGUI 代表 uc/GUI，UCFS 代表 uc/FS 文件系统，MMC 代表 MMC/SD 驱动，JPEG 用于选择 JPEG 库是否编译。将相应模块置 1，表示编译该模块；让相应模块等号右边为空，表示不编译该模块。等式右边以“#”起始的文字是注释，不影响编译。比如：

```
UCGUI = # denpedent on JPEG
```

```
CAMERA =1 # denpedent on JPEG
```

表示不编译 uc/GUI，编译摄像头驱动。

JZ4740 如下：

```
# -----
JZ4740_PAV = 1
#
#TOUCHTYPE :
#          0. No Touch driver
#          1. AK4182
#          2. JZ4740SADC
#
TOUCHTYPE = 0

#LCDTYPE :
#          0. No LCD Driver
#          1. Samsung tft 480 * 272
LCDTYPE = 0
#RTCTYPE :
#          0. No RTC Driver
#          1. jz4740RTC
RTCTYPE = 0

#CODECTYPE :
#          0. No Codec Driver
#          1. jz4740 Codec
CODECTYPE = 0
WAVDECODE = 0
#KeyCTYPE :
#          0. No Key Driver
#          1. jz4740 Key
KEYTYPE = 0

CAMERA = 0
LWIP = 0
UCGUI = 0
UCFS = 0
MMC = 0
JPEG = 0
```

```
UDC    = 1
MP3    = 0
NAND   = 0
#Driver Manager
DM     = 0
#-----
```


4 uCOS-II 驱动的使用和测试

4.1 网络驱动

网络驱动的结构

网络模块的代码主要包括四部分：

1. tcp/ip 协议栈 lwip1.1.1
2. lwip 与 uCOS-II 的接口代码(使用的是 Michael Anburaj 提供的代码,他的个人主页是 <http://geocities.com/michaelanburaj/>)
3. lwip 与 JZ4730 网口控制器的接口 (ucosii/lwip/src/netif/ethif_jz4730.c)
4. 基于 lwip 提供的 socket api 编写的网络应用程序 tftp (ucosii/lwip/src/tftp.c) 和网络例程主函数程序(ucosii/sample/main_lwip.c)。

网络驱动的使用

1. 首先要在 makefile 中设置 LWIP = 1, 让网络模块参与编译。
2. 通过编辑 lwip 配置选项文件 ucosii/lwip/src/include/lwipopt.h, 来配置基本的网络参数。

比如配置是否使用 DHCP 动态申请 IP, 以及 IP、TCP、UDP 的一些参数等, 具体参数的配置说明可参见 lwipopt.h 的注释部分。建议网络缓冲区的大小保持默认设置, 如果设置得更小的话可能不能正常工作。

3. 使用到了网络驱动的应用程序需要包含 lwip.h 头文件, 即需要 include “lwip.h”。并要做好 lwip 的初始化工作, 初始化函数是 void lwip_init(void), 具体代码请参见 main_lwip.c。

4. 通过调用函数 sys_thread_new(void (*function)(void *arg), void *arg, int prio) 来建立与网络相关的任务, prio 表示该任务的优先级, 注意该优先级的值要大于 TCPIP_THREAD_PRIO (在 lwipopt.h 中定义), 因为 tcpip_thread 任务是保证网络正常工作最基本的任务, 该任务通过初始化函数 tcpip_init(NULL, NULL) 来启动。

与网络相关的任务的堆栈大小和最大任务数在 ucosii/lwip/src/include/arch/sys_arch_opts.h 中定义:

```
#define LWIP_MAX_TASKS 4          /* Number of LwIP tasks */
#define LWIP_STACK_SIZE 512*10   /* Stack size for LwIP tasks */
```

网络驱动的测试

使用例程 ucosii/sample/main_lwip.c 作为主函数程序, 运行之后可以用四种方式进行网络功能测试。在 u-boot 引导 uCOS-ii 运行之后, 板子根据 IP 分配方式获取 IP, 如果设置为 DHCP 方式, 则动态分配 IP, 否则静态分配 IP 为 192.168.1.140。板子上电后, 可根据板子通过串口输出的运行信息, 得知板子的 IP, 下面的测试都假设板子的 IP 为 192.168.1.140。

a. ping 测试

lwip 协议栈自带了 ICMP 软件包, 因此只要对 lwip 进行了初始化, 就能 ping 通板子。可在电脑或终端上 ping 板子:

```
ping 192.168.1.140
```

若板子网络正常, 电脑或终端将会重复显示如下信息:

```
64 bytes from 192.168.1.183: icmp_seq=2 ttl=64 time=0.4 ms
```

b. http 测试

在例程 `main_lwip.c` 中实现了一个简单的 HTTP/1.0 server，在电脑或终端上的 web 浏览器地址栏中输入板子的 IP，回车，即可看到如下文字：

```
It's a LWIP test!
```

同时，板子会通过串口打印输出如下文字：

```
Receive a http request and send some text.
```

c. telnet 测试

在例程 `main_lwip.c` 中实现了一个简单的 telnet server，在电脑或终端上运行命令：

```
telnet 192.168.1.140
```

若网络正常可接收到如下文字：

```
Hello, world!
```

同时，板子会通过串口打印输出如下文字：

```
Telnet server: got connection from x.x.x.x
```

其中，`x.x.x.x` 表示访问板子的电脑或终端的 IP。

d. tftp 测试

不同于 http 和 telnet 的代码在 `main_lwip.c` 中实现，tftp 源代码因为比较大，所以用单独的文件 `ucosii/lwip/src/tftp.c` 实现。在 `tftp.c` 中有两个主要的子函数：

```
void getfile(char *dfilename, char *filename);
```

```
void putfile(char *filename, char *sfilename);
```

`getfile(char *dfilename, char *filename)` 用于从 TFTP 服务器下载文件到板子上的 SD 卡中，参数 `dfilename` 表示要下载到 SD 卡中的目标文件的文件名，参数 `filename` 表示服务器上要被下载的源文件的文件名。

`putfile(char *filename, char *sfilename)` 用于从板子上的 SD 卡上传文件到 TFTP 服务器中，参数 `sfilename` 表示 SD 卡中的要被上传的源文件的文件名，参数 `filename` 表示要上传到服务器的目标文件的文件名。

上述文件名可带路径。

举例：如果要把 TFTP 服务器 jpeg 文件夹下的 `a.jpg` 下载到 SD 卡根目录下的 `a.jpg`，那么需要在 `tftp.c` 的 `void tftp(void *data)` 任务函数中添加语句 “`getfile("a.jpg", "jpeg/a.jpg");`”。相反，如果要把 SD 卡根目录下的 `a.jpg` 上传到 TFTP 服务器 jpeg 文件夹下的 `a.jpg`，那么添加语句 “`putfile("jpeg/a.jpg", "a.jpg");`”。

在 `main_lwip.c` 中通过执行语句 “`sys_thread_new(tftp, (void *)0, TFTP_PRIO);`” 来建立 tftp 任务，运行时可以通过板子的串口输出看到有关下载和上传的数据信息。

4.2 摄像头驱动

摄像头驱动的结构

该驱动代码基本都在 `ucosii/jz4730/camera` 下：

`cim.c`: JZ4730 摄像头接口模块驱动

`camera.c`: 摄像头应用接口程序

`hv7131.c`: hv7131 摄像头配置程序

ksmov7649.c: ksmov7649 摄像头配置程序

ov7660.c: ov7660 摄像头配置程序

sensorlib.c: i2c 驱动与摄像头接口程序

snapshot.c: 摄像头应用程序

JPEG 库放在 ucoshii/ucgui/JPEG 文件夹下(该 JPEG 库同时用于摄像头的 jpeg 压缩和 uC/GUI 的 jpeg 解压缩)

另外, ucoshii/jz4730/include/camera.h 定义了使用哪种摄像头, 是主函数程序需要包含的头文件。

ucoshii/sample/main_camera.c 是摄像头例程

摄像头驱动的使用

1. 在 makefile 中设置 CAMERA =1, 让摄像头驱动代码参与编译
2. 在 camera.h 中定义使用哪种摄像头, 例如, 如果使用 ov7660 摄像头, 则:
#define CONFIG_OV7660 1. 并在主函数程序文件中包含该头文件。
3. 在主函数程序文件中, 在使用摄像头之前, 首先需要调用 camera_open()打开摄像头, 然后可以调用 putjpg(void *filename)函数读取摄像头输出的 RGB565 图像数据并保存为文件名为 filename 的 jpeg 格式文件, 或者直接把 RGB565 图像数据送给 LCD 的 frame buffer 从而直接在 LCD 上显示图像。在不需要使用摄像头时, 可以调用 camera_close()关闭摄像头。
4. 在 snapshot.c 中, 可通过修改 putjpg()函数中变量 i 的初始值(默认是 30)来设置拍摄几幅图片才保存一幅, 该值设得过小容易使得图像偏绿。在 put_image_jpeg()中, 可通过设置 cjpeg.dct_method 来选择使用哪种 jpeg 压缩方法, DCT_IFAST (精度较低, 速度较快) 或 DCT_ISLOW (无损压缩, 速度较慢), 默认的是 DCT_ISLOW。

摄像头驱动的测试

使用 main_camera.c 可以对摄像头驱动进行测试, 它默认使用 ov7660 摄像头, 通过调用 putjpg("a.jpg") 将图像保存在 SD 卡的 a.jpg 中。然后, a.jpg 通过 TFTP 任务中的 putfile("a.jpg", "a.jpg") 上传到 TFTP 服务器根目录的 a.jpg。在服务器上通过看图软件浏览 a.jpg。capture(void * data)任务函数通过循环调用 docapture() 将拍到的图像在 LCD 上实时显示出来。

4.3 UDC 驱动

UDC 驱动 在各芯片级目录下均有 UDC 驱动。还包括一个 MASSSTORAGE 的协议栈, 一个以 RAM 为例程的 U 盘接口。使用时, 只需要初始化就可以了。要在相应得 makefile 中把 UDC 设置为 1

另在 JZ4740 里, 还包括一个 GPIO Detect 功能。

4.4 LCD 驱动

LCD 驱动 在各芯片级 drv 中包含有 LCD 的驱动,

LCDC.c 是 LCDC 控制器的驱动, 以及含有的相应屏的驱动, 其中在屏的驱动中还包括 PWM 背光的控制。在 LCDC 中导出一套与 FRAMEBUF 相关的指针供 GUI 以及视频解码等使用。

在 JZ4730 中支持的 LCD 有: foxconxx_pt035tn trully_tft-g320240dsw

在 JZ4740 中支持的 LCD 有: foxconxx_pt035tnserial Samsung_tft-g480272

另在 JZ47XX/Sample 下均有相应 LCD 的驱动测试程序。

4.5 音频驱动

在 DRV/CODEC 下。

在 JZ4730 中：包括 AC97 I2S AKM4642 两类 CODEC 驱动。

在 JZ4740 中：包括 JZ4740 CODEC 驱动。

其中均导出 `pcm_write` `pcm_read` 两个函数接口。其驱动测试可通过播放 MP3 WAV 时。另在 JZ47XX/Sample 下还包括录音并回放的测试程序。

4.6 触摸屏驱动

在 DRV/TOUCH 下。

在 JZ4730 中：包括 AC97 SPI AK4182 两类触摸屏驱动。

在 JZ4740 中：包括 JZ4740 SADC 驱动。其次试，可通过 UCGUI 或 JZ47XX/Sample 下测试。