

龙芯中科
LOONGSON TECHNOLOGY

2K 龙芯派系列手册

实验手册



地址：北京市海淀区中关村环保科技示范园龙芯产业园2号楼
电话：+86 (10) 62546668（分机1601/1609/1610/1620/1621/1605）
客户服务邮箱：duanli@loongson.cn / wangjue@loongson.cn
网址：www.loongson.cn

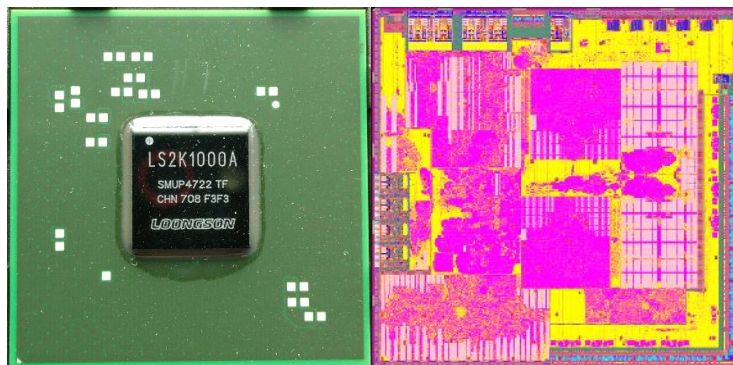
目 录

一、	2K 龙芯派设备平台简介.....	1
1.1	Loongson2K1000 处理器介绍.....	1
1.1.1	芯片规格.....	1
1.1.2	芯片结构图.....	2
1.2	2K 龙芯派介绍.....	2
1.2.1	2K 龙芯派硬件规格.....	3
1.2.2	Loongnix 操作系统.....	3
1.2.3	SylixOS 翼辉实时操作系统.....	4
二、	2K 龙芯派外设模组简介.....	4
2.1	温湿度感应模块.....	4
2.2	蓝牙通讯模块.....	4
2.3	蜂鸣器.....	5
2.4	流水灯控制模块.....	5
2.5	八位模拟数码管.....	6
2.6	触摸按键模块.....	6
三、	基础实验.....	6
3.1	GPIO 实验.....	6
3.1.1	实验设备.....	6
3.1.2	实验原理.....	7
3.1.3	参考代码.....	7
3.1.4	实验步骤.....	8
3.2	I2C 接口实验.....	9
3.2.1	实验设备.....	9
3.2.2	实验内容.....	9
3.2.3	参考资料.....	10
3.2.4	实验步骤.....	10
3.3	串口实验.....	10
3.3.1	实验设备.....	10
3.3.2	实验原理.....	10
3.3.3	参考代码.....	10
3.3.4	实验步骤.....	11
四、	SylixOS 实时操作系统实验.....	11
4.1	SylixOS 内核模块基本实验.....	11
4.1.1	实验目的.....	11
4.1.2	实验内容.....	11
4.1.3	实验环境.....	11
4.1.4	实验原理.....	11
4.1.5	实验步骤.....	11
4.2	SylixOS 八位数码管 I/O 控制实验.....	13
4.2.1	实验目的.....	13
4.2.2	实验内容.....	13

4.2.3	实验环境.....	14
4.2.4	实验原理.....	14
4.2.5	实验步骤.....	14
4.3	SylixOS 温湿度感应实验	14
4.3.1	实验目的.....	14
4.3.2	实验内容.....	14
4.3.3	实验环境.....	15
4.3.4	实验原理.....	15
4.3.5	实验步骤.....	15
4.4	SylixOS 蓝牙模块调试实验	15
4.4.1	实验目的.....	15
4.4.2	实验内容.....	16
4.4.3	实验环境.....	16
4.4.4	实验原理.....	16
4.4.5	实验步骤.....	17
4.5	SylixOS 触摸按键实验	17
4.5.1	实验目的.....	17
4.5.2	实验内容.....	17
4.5.3	实验环境.....	17
4.5.4	实验原理.....	18
4.5.5	实验步骤.....	18
4.6	SylixOS 八位流水灯实验	19
4.6.1	实验目的.....	19
4.6.2	实验内容.....	19
4.6.3	实验环境.....	19
4.6.4	实验原理.....	19
4.6.5	实验步骤.....	20
4.7	SylixOS 蜂鸣器实验	20
4.7.1	实验目的.....	20
4.7.2	实验内容.....	20
4.7.3	实验环境.....	20
4.7.4	实验原理.....	20
4.7.5	实验步骤.....	21

一、 2K 龙芯派设备平台简介

1.1 Loongson2K1000 处理器介绍

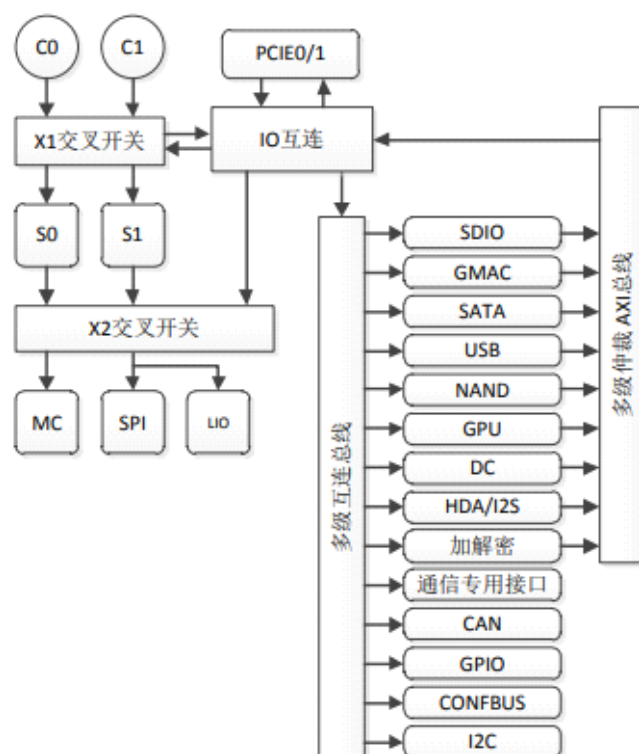


龙芯 2K1000 处理器集成两个 GS264 处理器核，芯片外围接口包括两路 x4 PCIE2.0、一路 SATA2.0、4 路 USB2.0、两路 DVO、64 位 DDR2/3，及其它各种小接口。该芯片可以满足网络安全、工业控制、小型终端等领域应用需求，并为其他扩展应用提供相应的接口。

1.1.1 芯片规格

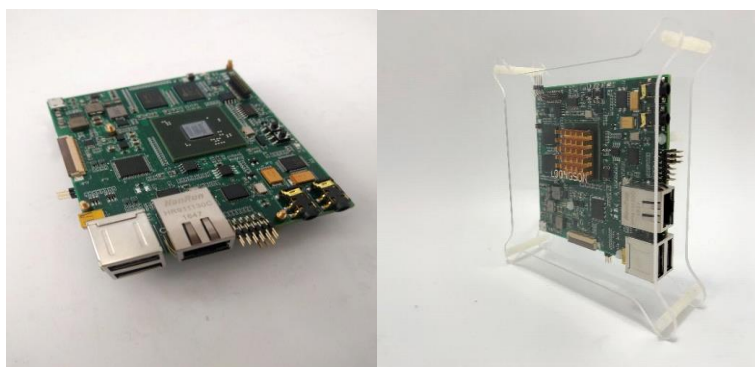
龙芯 2K1000 芯片规格	
主频	1GHz
处理器核	2 个 64 位超标量处理器核GS264； 支持MIPS64 指令集； 支持LISA64 指令集； 10 级超标量流水线；双发射乱序执行； 2 个定点单元、1 个浮点单元和1 个访存单元
高速缓存	每个处理器核包含32KB 私有一级指令缓存和32KB 私有一级数据缓存； 所有处理器核共享1MB 二级缓存
内存控制器	1 个64 位DDR2/3-1333 控制器；
高速 I/O	2 个PCIE x4 控制器； 可分别配置为4 路x1 及2 路x1 控制器
其它 I/O	4 路USB；SATA；2 路DVO；2 路GMAC NAND；最多12 路UART 2 路CAN AC97/I2S I2C SPI/SDIO GPIO PCIE 等
制造工艺	40nm CMOS 工艺
功耗管理	支持主要模块时钟动态控制；支持ACPI
典型功耗	4.5W@1GHz

1.1.2 芯片结构图



一级交叉开关连接两个处理器核、两个二级 Cache 以及 IO 子网络（Cache 访问路径）。二级交叉开关连接两个二级 Cache、内存控制器、启动模块（SPI 或者 LIO）以及 IO 子网络（Uncache 访问路径）。IO 子网络连接一级交叉开关，以减少处理器访问延迟。IO 子网络中包括需要 DMA 的模块（PCIE、GMAC、SATA、USB、HDA/I2S、NAND、SDIO、DC、GPU 和加解密模块）和不需要 DMA 的模块，需要 DMA 的模块可以通过 Cache 或者 Uncache 方式访问内存。

1.2 2K 龙芯派介绍



2K 龙芯派是首款采用 2K1000 低功耗处理器设计的嵌入式方案验证板，具有资源丰富、接口齐全、低功耗、高可靠的特点。龙芯派搭载 2K1000 处理器（主频 1GHz），板载 DDR3 颗粒，实现 DDR3 的运行存储功能。实现了 GPIO 的输入输出，中断功能。板上集成 1 个网口，集成 3 个 USB 接口，HDMI 接口，LCD 接口，音频输入/输出，集成 SD 卡接口，集成 2 个 CAN 接口，集成 RTC 计时功能。可以外扩 WIFI 模块。2K 龙芯派可以广泛应用于信息安全、电力、轨道交通、工业控制、信号处理、数据通信、信息教育等领域。

1.2.1 2K 龙芯派硬件规格

2K 龙芯派资源丰富，采用龙芯最新一代 2 号处理器 2K1000，集成存储、显示、音频、网络等功能，具有高性能，低功耗的特点。该板卡的具体特性如下：

- (1) 采用 2K1000 CPU，主频 1GHZ；
- (2) 板载 DDR3 内存 2GB，主频 400Mhz；
- (3) 板载启动 Nor FLASH 32Mb；
- (4) 集成 Micro SD 卡接口，可用于系统安装、存储数据等；
- (5) 集成 2 路 10/100/1000M 自适应网口；
- (6) 集成 1 路 2.4G WIFI 模块；
- (7) 集成 3 路 USB2.0 接口，2 个标准 USB 接口，1 个插针接口；
- (8) 集成 HDMI、LCD 显示接口；
- (9) 集成 3.5mm 音频输入/输出接口；
- (10) 集成 4 路 PWM 接口；
- (11) 集成 2 路 I2C 接口；
- (12) 集成 1 路 SPI 接口，包含 3 个片选；
- (13) 集成 4 路 TTL 串口；
- (14) 可配最多 30 个输入/输出 GPIO；
- (15) 集成 2 个 CAN2.0 接口；
- (16) 集成 EJTAG 调试接口，可用于程序下载、单步调试等；
- (17) 集成 RTC 功能，保留外置 RTC 电池接口；
- (18) 集成 CPU 温度检测功能；
- (19) 集成复位按键；
- (20) 集成开机按键；
- (21) 集成 Micro USB 5V/2A 供电接口，该板卡整体功耗<5W；
- (22) 板卡尺寸 96mm*80mm

1.2.2 Loongnix 操作系统

Loongnix 操作系统是龙芯开源社区推出的 Linux 操作系统，作为龙芯软件生态建设的成果验证和展示环境，集成在内核、驱动、图形环境等操作系统基础设施方面的最新研发成果，以“源码开放、免费下载”的形式进行发布，可直接应用于日常办公、生产、生活

等应用环境，同时可供合作厂商、科研机构及爱好者等在龙芯平台上研发其品牌软件或专用系统。

用户想要获取最新的 Loongnix 操作系统，以及 PMON、内核、编译工具等基础软件，可以访问龙芯开源社区（<http://www.loongnix.org/index.php/%E9%A6%96%E9%A1%B5>）

1.2.3 SylixOS 翼辉实时操作系统

SylixOS 是国内一款内核自主化率达到 100% 的开源大型实时操作系统；支持 ARM、MIPS、PowerPC、x86、SPARC、DSP 等处理器架构，便于在不同硬件平台之间进行系统迁移；硬实时内核，调度算法先进高效，性能强劲；应用编程接口符合 IEEE、ISO、IEC、GJB7714-2012 相关操作系统编程接口规范，便于基于 Linux、VxWorks 等操作系统应用向 SylixOS 系统的迁移。

SylixOS 提供一整套集设计、开发、调试、仿真、部署、测试于一体的开发平台，便于系统开发与调试，加快软件研发速度，缩短产品研制周期，助力用户专注应用开发。

详细信息可由[翼辉信息技术有限公司官网](http://www.acoinfo.com)(<http://www.acoinfo.com>)了解，通过翼辉信息官网 <http://www.acoinfo.com/html/experience.php> 申请龙芯翼辉集成开发套件，申请页面中的附加信息栏填写“**龙芯派开发**”，开发套件下载链接将会发送到申请时预留的邮箱。

二、 2K 龙芯派外设模组简介

2.1 温湿度感应模块

SHT20 数字温湿度传感器基于领先世界的 CMOSens[®] 数字传感技术，具有极高的可靠性和卓越的长期稳定性。全量程标定，两线数字接口，可与单片机直接相连，大大缩短研发时间、简化外围电路并降低费用。此外，该模块体积微小、响应迅速、低能耗、可浸没、抗干扰能力强、温湿一体，兼有露点测量，性价比高，使该产品能够适于多种场合的应用。

技术参数

湿度测量范围：0~100%RH

湿度测量精度：±3%RH

温度测量范围：-40~125℃

温度测量精度：±0.3℃

工作电压：2.1~3.6VDC（请注意：请勿使用 5V 供电）

输出接口：I2C 接口输出

2.2 蓝牙通讯模块

DX-BT05 蓝牙模块采用美国 TI 的 CC2541 芯片，配置 256Kb 空间，遵循 V4.0 BLE 蓝牙规范。支持 AT 指令，用户可以根据需要更改串口波特率、设备名称、配对密码等参数。该模块支持 UART 接口，并支持 SPP 蓝牙串口协议，具有成本低、体积小、功耗低、收发灵敏等特点。

技术参数

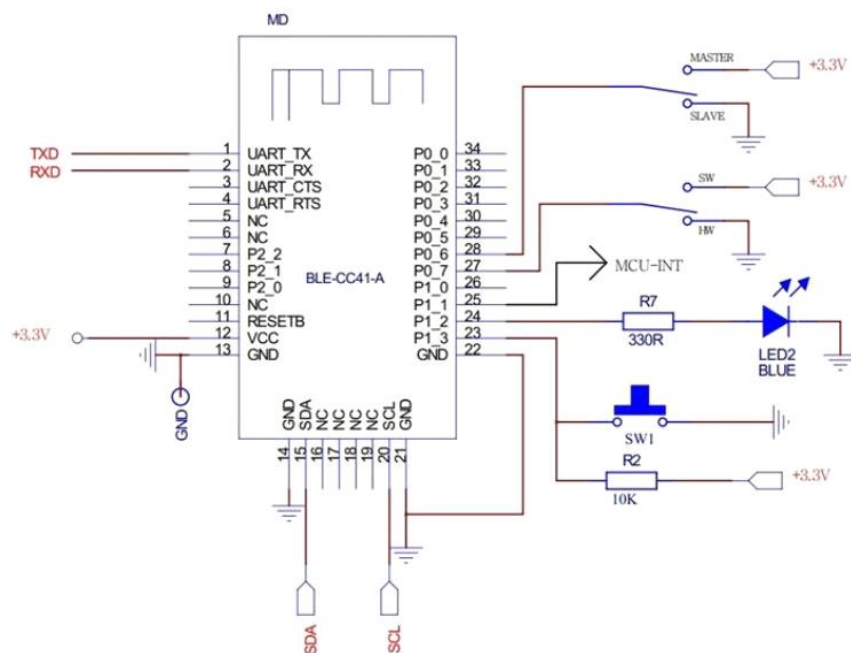
连接距离：60 米

蓝牙协议：支持标准 BLE 协议，支持蓝牙 Class1 和 Class2 模式

使用条件：iPhone4S 以上型号，iOS6 以上；Android4.3 版本以上，手机蓝牙版本 4.0

输入电压：3V

模块电路图



2.3 蜂鸣器

模块采用 S8550 三极管驱动，工作电压 3.3V~5V。当 I/O 口输入低电平时，蜂鸣器发声。

接口说明

VCC 外接 3.3V-5V 电压（可以直接与 5V 单片机和 3.3V 单片机相连）

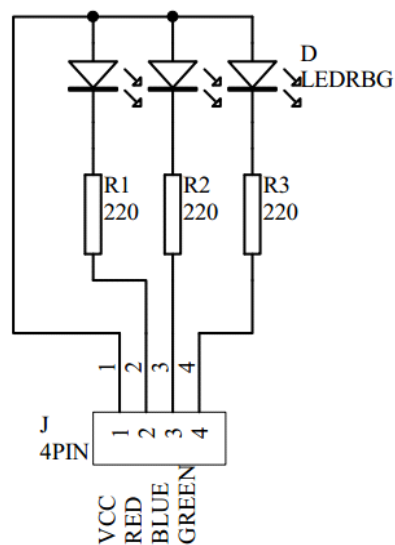
GND 外接 GND

I/O 外接单片机 IO 口

2.4 流水灯控制模块

模块配备 8 个三基色（红绿蓝）全彩 LED，通过扫描控制，与数码管控制方式相同。位选控制对应 LED 点亮，段选控制颜色。通过不同程序可以产生不同的颜色变化效果。

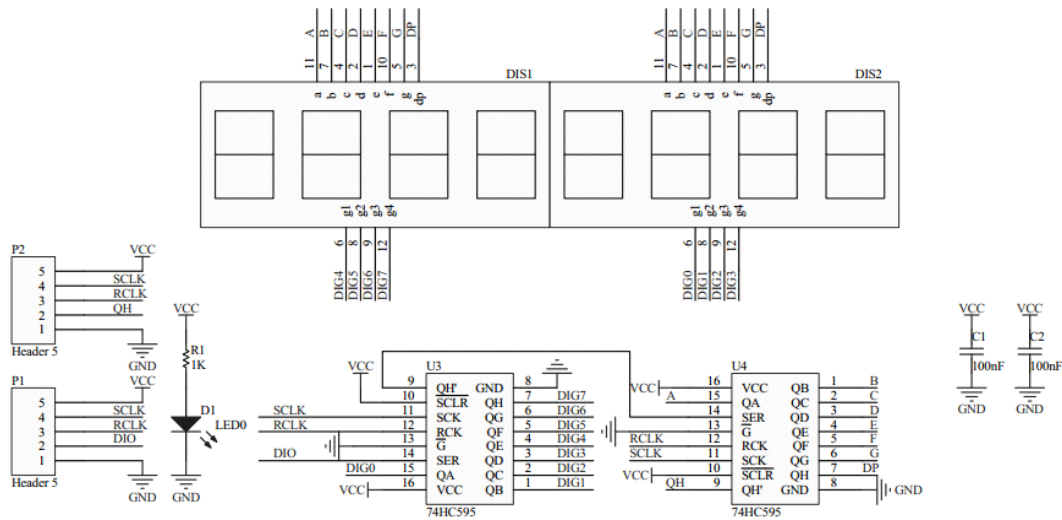
模块电路图



2.5 八位模拟数码管

采用 2 片 595 驱动数码管，根据数码管动态扫描原理进行显示。

模块电路图



2.6 触摸按键模块

TTP223 触摸按键模块是一个基于触摸检测 IC（TTP223B）的电容式点动型触摸开关模块。常态下，模块输出低电平，模式为低功耗模式；当用手指触摸相应位置时，模块会输出高电平，模式切换为快速模式；当持续 12 秒没有触摸时，模式又切换为低功耗模式。可以将模块安装在非金属材料如塑料、玻璃的表面，另外将薄薄的纸片（非金属）覆盖在模块的表面，只要触摸的位置正确，即可做成隐藏在墙壁、桌面等地方的按键。

项目	最小值	典型值	最大值	单位
电源电压 VCC	2.0	3	5.5	V
输出高电平 VOH	—	0.8VCC	—	V
输出低电平 VOL	—	—	0.3VCC	V
输出引脚灌电流 (@VCC=3V,VOL=0.6V)	—	8	—	mA
输出引脚拉电流 (@VCC=3V,VOH=2.4V)	—	-4	—	mA
响应时间（低功耗模式）	—	—	220	mS
响应时间（快速模式）	—	—	60	mS
尺寸	24X24X7.2			mm
重量	2			g

三、基础实验

3.1 GPIO 实验

3.1.1 实验设备

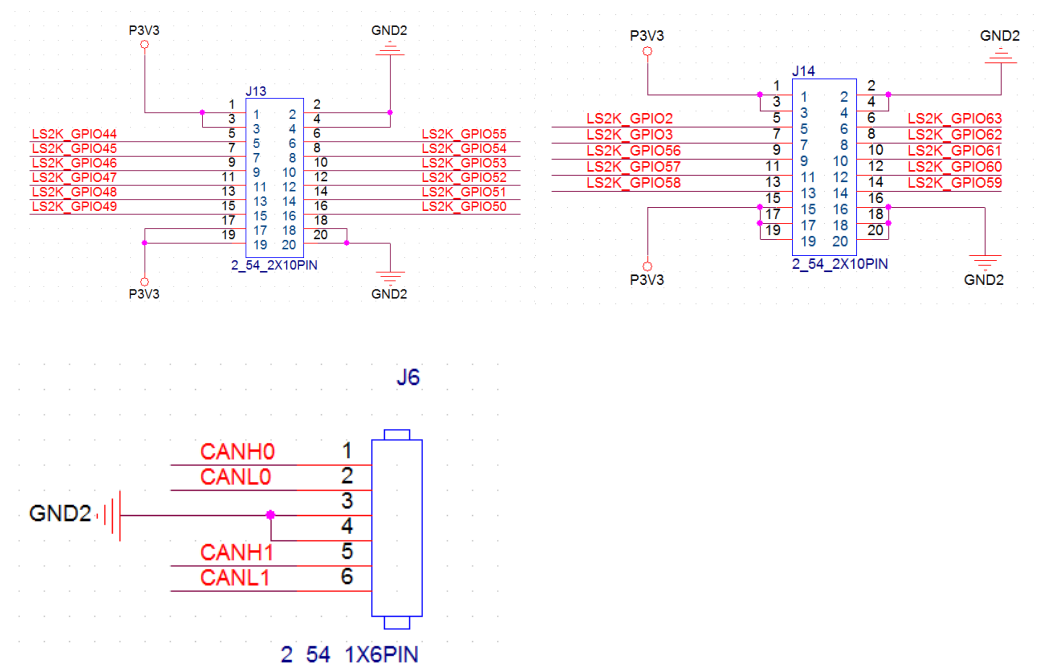
蜂鸣器，74HC595 八位数码管，2K 龙芯派，IO/CAN 转接板，杜邦线。

3.1.2 实验原理

通过 linux 内核态及应用层接口设置 GPIO 状态控制不同硬件设备工作。把 IO 模块连接到 2K 龙芯派的 IO 上即可，本实验使用 IO 转接板连接外设，IO 转接板连接如下：



扩展板的插针引脚定义如下：



3.1.3 参考代码

gpio 驱动代码位于 arch/mips/loongson2/ls2k/gpio.c
74HC595 驱动 drivers/char/74HC595.c

74HC595 应用程序 test_led.c,代码如下:

```
#include <sys/types.h>
#include <sys/stat.h>
#include <stdio.h>
#include <fcntl.h>

static char num[]={0x9, 1, 2, 3, 4, 5, 6, 7};

main()
{
int fd;
fd=open("/dev/my74hc595-0", O_RDWR); //可读写方式打开设备文件
if(fd!=-1)
{
write(fd, &num, 8); //写设备变量

close(fd); //关闭设备文件
}
else
{
printf("Device open failure\n");
perror("open my74hc595");
}
}
```

3.1.4 实验步骤

八位数码管实验

(1) 连接数码管到板卡上

GPIO59 接 DIO

GPIO57 接 SCLK

GPIO58 接 RCLK

(2) 编译及运行应用程序

mipsel-linux-gcc -static -o test_led test_led.c

用 U 盘拷贝 test_led 到板卡中执行如下命令:

chmod +x test_led

./test_led

可观察到八位数码管按 num 数组值依次点亮和熄灭。

(3) 控制数码管熄灭时间

修改驱动 write_to_74hc 函数和 LED_OUT 函数中的 msleep (xx) 函数中的 xx 值, 控制每位数码管的熄灭时间。编译内核并更新到系统 boot 目录下, 重启系统运行 test_led, 可多次重复实验。

蜂鸣器控制 (用户态接口)

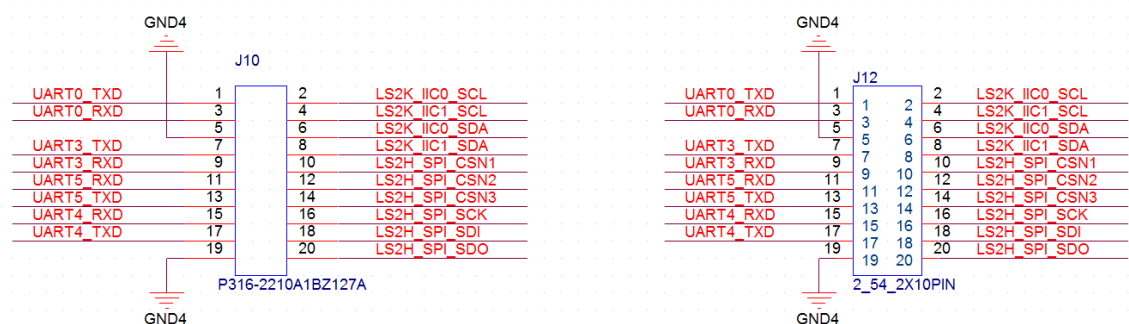
本实验通过用户态 GPIO 接口, 实现对 GPIO 高低的控制, 达到对蜂鸣器的控制。以蜂鸣器接在 GPIO1 为例, 在 fedora21 的终端下输入如下命令:

```
echo 1 > /sys/class/gpio/export 导出 GPIO1
echo out > /sys/class/gpio/gpio1/direction 设置 GPIO 方向为输出
cat /sys/class/gpio/gpio1/value 读取 GPIO 值
echo 1 > /sys/class/gpio/gpio1/value GPIO1 输出为高
echo 0 > /sys/class/gpio/gpio1/value GPIO1 输出为低
可通过输出高低的时间长短来控制蜂鸣器鸣响的时间。
```

3.2 I2C 接口实验

3.2.1 实验设备

I2C 温湿度模块，2K 龙芯派，UART/I2C/SPI 扩展板、杜邦线。扩展板连接如下：



3.2.2 实验内容

2K 龙芯派预留了两路 I2C 到 UART/I2C/SPI 转接板，软件按 sht21 规范通信协议读取温度及湿度，修改使用不同 I2C 接口及修改驱动源码读出。

3.2.3 参考资料

驱动代码位于 drivers/hwmon/sht21.c

传感器 datasheet SHT20-Datasheet(温湿度).pdf

2K 龙芯派原理图

3.2.4 实验步骤

(1) 连接传感器到板卡上

按入门手册要求把传感器接到板卡的 I2C1 接口上。

(2) 编译内核及驱动

a) 配置编译选项

make menuconfig ARCH=mips

驱动选项位于：

Device Drivers ----->

<*> Hardware Monitoring support----->

<*>Sensiron humidity and temperature sensors. SHT21 and compat.

(2) 读出未转化温度

用 u 盘拷贝内核 vmlinux 到 sd 卡的 boot 目录中，重启后驱动生效。在终端下通过如下命令可读取到温湿度。

cat /sys/class/hwmon/hwmon0/device/humidity1_input 可读取到湿度

Cat /sys/class/hwmon/hwmon0/device/temp1_input 可读取到温度

(3) 修改驱动

根据传感器 datasheet 修改，驱动程序读出摄氏温度及相对湿度

(4) 修改 I2C 接口

修 改 arch/mips/loongson2/ls2k/platform.c 中 i2c_register_board_info(I2C_BUS_1, &ls2k_sht21_info,1)的 I2C_BUS_1 为 I2C_BUS_0 可切换为 i2c0 上，通过切换不同 I2C 接口重复如上实验。

3.3 串口实验

3.3.1 实验设备

蓝牙模块，2K 龙芯派，杜邦线

3.3.2 实验原理

通过串口编程对蓝牙模块发送不同命令，以达到控制蓝牙模块工作的目的。

3.3.3 参考代码

应用程序 comm.c socket2.c

(注：代码以单独文件提供)

3.3.4 实验步骤

(1) 编译串口测试应用程序

```
mipsel-linux-gcc -static -o test_uart comm.c socket2.c
```

(本实验程序是发送数据到蓝牙)

(2) 打开手机中的蓝牙接收程序

(3) 运行 test_uart

可在手机 APP 上看到不停的有 0x55 0x11 到 0xff 数据接收。

(4) 用户可以修改应用程序及 APP 进行不同方向的数据接收及发送。

四、 SylixOS 实时操作系统实验

4.1 SylixOS 内核模块基本实验

4.1.1 实验目的

- 掌握内核模块工程设计的流程
- 对内核模块的功能实现和使用有形象的认识

4.1.2 实验内容

- 完成一个内核模块工程的创建、编译、装载和卸载

4.1.3 实验环境

- 硬件：已经部署 SylixOS 操作系统的验证平台，一台 PC 主机
- 软件：RealEvo-IDE、putty

4.1.4 实验原理

内核模块文件开始，需要使用宏定义“__SYLIXOS_KERNEL”声明当前文件为内核模块。

任何一个内核模块工程或文件都要包含两个头文件“SylixOS.h”和“module.h”，前者是 SylixOS 系统统一的头文件，包含系统内核各个模块功能定义，后者则包含内核模块装载器相关参数与 API。

装载一个内核模块的 shell 命令是 insmod，在装载模块时，会自动调用内核模块内的 module_init 函数。在内核模块中需要进行的初始化工作可以放在 module_init 函数内。编写驱动程序时需要的接口初始化、注册设备这些操作都需要放在此函数内。

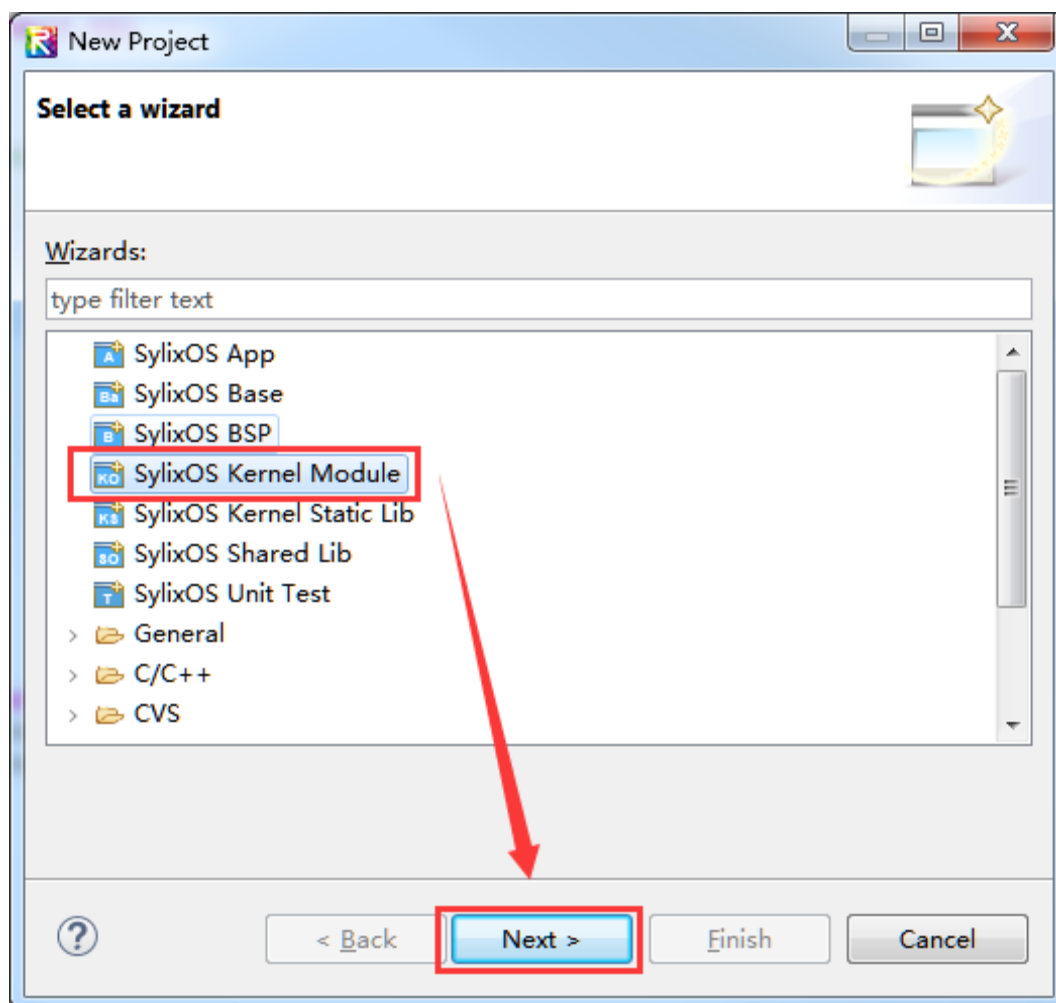
卸载一个内核模块的 shell 命令是 rmmod，在卸载模块时，会自动调用内核模块内的 module_exit 函数。内核模块中一些任务结束后需要释放系统资源，释放系统资源操作可以放在 module_exit 内完成。

一个内核模块工程中可能有很多函数，但不是所有的函数都能够被其他模块或程序使用，只有被宏“LW_SYMBOL_EXPORT”声明的函数才能够被其他模块或程序使用。编译内核模块的 SylixOS 内核版本与验证平台运行的 SylixOS 内核版本需要保持一致，否则会装载失败。

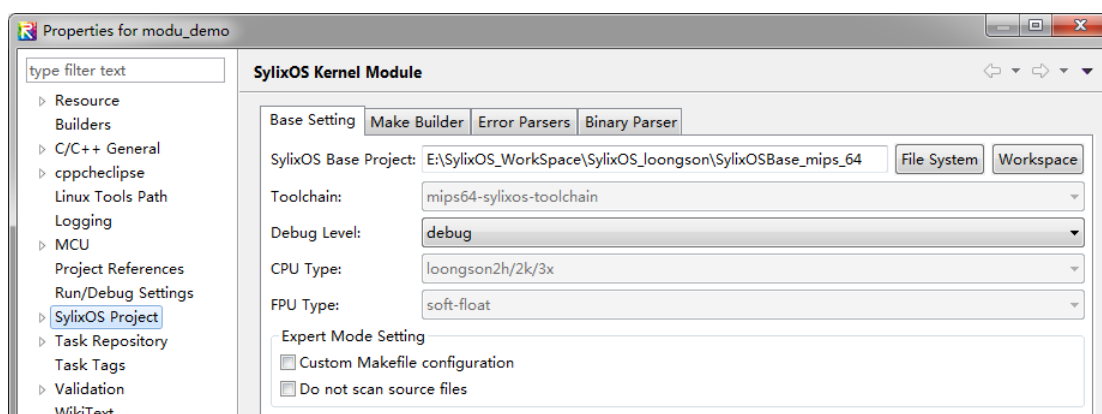
4.1.5 实验步骤

1. 创建内核模块工程

打开 RealEvo-IDE， “File” → “New” → “Project” 创建内核模块工程，选择 “SylixOS Kernel Module”。如下图所示：



点击 “Next” 按钮，添加工程命名 modu_demo，点击 “Next” 按钮，选择 “SylixOS base project” 路径。如下图所示：



之后点击 “Finish” 按钮，完成工程创建。创建后的工程，会自动生成 modu_demo.c 文件，文件包含有默认代码。本实验例程中不对生成的代码做任何修改。

内核模块工程生成的默认代码

```
#define __SYLIXOS_KERNEL
#include <SylixOS.h>
#include <module.h>
/*
 * SylixOS call module_init() and module_exit() automatically.
 */
void module_init (void)
{
    printk("hello_module init!\n");
    return (0);
}
void module_exit (void)
{
    printk("hello_module exit!\n");
}
/*
 * module export symbols
 */
LW_SYMBOL_EXPORT void hello_module_func (void)
{
    printk("hello_module_func() run!\n");
}
```

2. 装载内核模块

编译工程 `modu_demo`，将生成的程序文件 `modu_demo.ko` 上传到验证平台中。需要注意，内核模块工程的“Upload Setting”中默认路径是 `/lib/modules/modu_demo.ko`。

在控制台中，将当前工作路径切换到 `/lib/modules` 下，装载内核模块

```
[root@sylixos_station:/lib/modules]# insmod modu_demo.ko
hello_module init!
module modu_demo.ko registor ok, handle : 0xffffffff82131470
```

- 执行装载命令后，输出信息是 `module_init` 函数的打印信息
- 装载成功后，会返回 `handle`

3. 卸载内核模块

```
[root@sylixos_station:/lib/modules]# rmmod modu_demo.ko
hello_module exit!
```

4.2 SylixOS 八位数码管 I/O 控制实验

4.2.1 实验目的

- 掌握对显示数码管的操作

4.2.2 实验内容

- 阅读 `app_hc595.c` 程序代码，了解 `hc595` 的操作流程和相关函数使用

- 编译 app_hc595 工程，并将程序上传到验证平台运行查看效果

4.2.3 实验环境

- 硬件：八位串行 595 数码管，2K 龙芯派，杜邦线
- 软件：RealEvo-IDE、putty

4.2.4 实验原理

硬件接线：

模块 P1 的 1 引脚连接 J14 的 2 引脚；模块 P1 的 2 引脚连接 J14 的 14 引脚；
模块 P1 的 3 引脚连接 J14 的 13 引脚；模块 P1 的 4 引脚连接 J14 的 11 引脚；
模块 P1 的 5 引脚连接 J14 的 1 引脚。

程序清单 app_hc595.c

```
#include <stdio.h>

int main (int argc, char **argv)
{
    INT      iFd;
    UCHAR    ucBuf[8]={1, 2, 3, 4, 5, 6, 7, 8};

    iFd = open("/dev/hc595", O_RDWR, 0666);
    if (iFd < 0) {
        printf("open dev failed\r\n");
        return (-1);
    }

    write(iFd, ucBuf, sizeof(ucBuf));
    sleep(1);
    close(iFd);

    return (0);
}
```

4.2.5 实验步骤

1. 使用 IDE System FTP 上传 ls2kdrv_hc595.ko 内核模块库到验证平台
2. 在控制台中，切换到/lib/modules/drivers 目录，执行命令 insmod ls2kdrv_hc595.ko
3. 切换到/apps/app_hc595 目录，执行命令 ./app_hc595；看到数码显示管循环依次显示 1~8

4.3 SylixOS 温湿度感应实验

4.3.1 实验目的

- 掌握通过温湿度模块采集温度和湿度

4.3.2 实验内容

- 阅读 app_sht20.c 程序代码，了解该模块的操作及使用

- 编译 app_sht20 工程，并将程序上传到验证平台运行查看效果

4.3.3 实验环境

- 硬件：I2C 温湿度模块，2K 龙芯派，杜邦线
- 软件：RealEvo-IDE、putty

4.3.4 实验原理

硬件接线：

模块 scl 引脚连接到 J12 的 2 引脚；模块 sda 引脚连接到 J12 的 6 引脚；

模块 vcc 引脚连接到 J13 的 1 引脚；模块 gnd 引脚连接到 J13 的 2 引脚；

该模块是 IIC 接口，与龙芯 2K 派的 IIC 接口相连即可，该实验例程连接到 IIC0 通道。

程序清单 app_sht20.c

```
#include <stdio.h>

int main (int argc, char **argv)
{
    INT      iFd;

    iFd = open("/dev/sht20", O_RDWR, 0666);
    if (iFd < 0) {
        printf("open dev failed\r\n");
        return (-1);
    }
    while(1) {
        UCHAR    ucBuf[4]={0};
        read(iFd, ucBuf, 4);
        printf("temp:%d    humi:%d\r\n", (*(UINT16 *)ucBuf) / 1000,
                                                    (*(UINT16 *) (ucBuf + 2)) / 1000);

        sleep(1);
    }
    close(iFd);

    return (0);
}
```

4.3.5 实验步骤

1. 使用 IDE System FTP 上传 ls2kdrv_sht20.ko 内核模块库到验证平台
2. 在控制台中，切换到/lib/modules/drivers 目录，执行命令 insmod ls2kdrv_sht20.ko
3. 切换到/apps/app_sht20 目录，执行命令 ./app_sht20；终端打印当前的温度和湿度值

4.4 SylixOS 蓝牙模块调试实验

4.4.1 实验目的

- 掌握通过串口和蓝牙模块通讯

- 掌握通过手机蓝牙和蓝牙模块通讯

4.4.2 实验内容

- 理解 app_bluetooth.c 程序代码，掌握和蓝牙模块通讯的方法
- 掌握手机和蓝牙模块的通讯方法

4.4.3 实验环境

- 硬件：蓝牙模块，2K 龙芯派，安卓或苹果手机，杜邦线
- 软件：RealEvo-IDE、putty

4.4.4 实验原理

硬件接线：

模块 TXD 引脚连接到 J12 的 15 引脚；模块 RXD 引脚连接到 J12 的 17 引脚；

模块 3.3V 引脚连接到 J13 的 1 引脚；模块 GND 引脚连接到 J13 的 2 引脚；

DX-BT05 4.0 蓝牙模块是专为智能无线数据传输而打造，采用美国 TI 公司 CC2541 芯片，配置 256Kb 空间，遵循 V4.0 BLE 蓝牙规范。支持 AT 指令，用户可根据需要更改串口波特率、设备名称、配对密码等参数，使用灵活。

本模块支持 UART 接口，并支持 SPP 蓝牙串口协议，具有成本低、体积小、功耗低、收发灵敏性高等优点，只需配备少许的外围元件就能实现其强大功能。

程序清单 app_bluetooth.c

```
#include <stdio.h>
#include "pthread.h"

int iFd; /* 文件描述符返回值 */

void *pthread_tx_test (void *pvArg)
{
    int iFd = *(int *)pvArg;

    write(iFd, "AT\r\n", sizeof("AT\r\n"));
    sleep(1);
    write(iFd, "AT+VERSION\r\n", sizeof("AT+VERSION\r\n"));
    sleep(1);

    return (NULL);
}

void *pthread_rx_test (void *pvArg)
{
    int iFd = *(int *)pvArg;

    for (;;) {
        char buf[1024] = "";
        read(iFd, buf, 1024);
    }
}
```

```

        printf("\r%s\n", buf);
    }

    return (NULL);
}

int main (int  argc, char  **argv)
{
    pthread_t    Txtid, Rxtid;

    iFd = open("/dev/ttyS4", O_RDWR, 0666);

    ioctl(iFd, FIOBAUDRATE, 9600);
    ioctl(iFd, FIOSETOPTIONS, (OPT_TERMINAL & (~OPT_7_BIT) & (~OPT_ECHO)));

    pthread_create(&Txtid, NULL, pthread_tx_test, (void *)&iFd);
    pthread_create(&Rxtid, NULL, pthread_rx_test, (void *)&iFd);

    pthread_join(Txtid, NULL);
    pthread_join(Rxtid, NULL);

    return (0);
}

```

4.4.5 实验步骤

1. 编译 app_bluetooth 工程，将生成的程序文件上传到验证平台
2. 在控制台中，切换到/apps/app_bluetooth 目录，执行令 ./app_bluetooth 可以看到串口上打印 OK 和蓝牙的版本号
3. 用手机客户端 LightBlue 和蓝牙模块通讯，参考《DX-BT05 4.0 蓝牙手机操作指南》

4.5 SylixOS 触摸按键实验

4.5.1 实验目的

- 掌握通过 GPIO 来采集触摸按键的输入状态

4.5.2 实验内容

- 阅读 app_tip223b.c 程序代码，了解触摸按键的操作流程和相关函数使用
- 编译 app_tip223b 工程，并将程序上传到验证平台运行查看效果

4.5.3 实验环境

- 硬件：tip223b 触摸按键，2K 龙芯派，杜邦线
- 软件：RealEvo-IDE、putty

4.5.4 实验原理

硬件接线：

模块 VCC 引脚连接到 J13 的 1 引脚；模 GND 引脚连接到 J13 的 2 引脚；

模块 SIG 引脚连接到 J14 的 11 引脚。

该模块是一个基于触摸检测 IC（TTP223B）的电容式点动型触摸开关模块。常态下模块输出低电平，模式为低功耗模式；当用手指触摸相应位置时，模块会输出高电平，模式切换为快速模式；当持续 12 秒没有触摸时，模式又切换为低功耗模式。可以将模块安装在非金属材料如塑料、玻璃的表面，另外将薄薄的纸片（非金属）覆盖在模块的表面，只要触摸的位置正确，即可做成隐藏在墙壁、桌面等地方的按键。该模块可以让你免除常规按压型按键的烦恼。

模块特点如下：

点动型：初态为低电平，触摸为高电平，正反面均可作为触摸面；

控制接口：共 3 个引脚（GND、VCC、SIG），SIG 为数字信号输出脚；

电源指示灯：绿色 LED，上电正确即发亮；

触摸区域：类似指纹的图标内部区域，手指轻轻触摸即可触发；

定位孔：4 个 M2 螺丝定位孔，孔径为 2.2mm，便于安装定位，实现模块间组合。

程序清单 app_tip223b.c

```
#include <stdio.h>

int main (int argc, char **argv)
{
    INT      iFd;

    iFd = open("/dev/tip223b", O_RDWR, 0666);
    if (iFd < 0) {
        printf("open dev failed\r\n");
        return (-1);
    }
    sleep(5);
    close(iFd);

    return (0);
}
```

4.5.5 实验步骤

1. 使用 IDE System FTP 上传 ls2kdrv_tip223b.ko 内核模块库到验证平台
2. 在控制台中，切换到/lib/modules/drivers 目录，执行命令 insmod ls2kdrv_tip223b.ko
3. 切换到/apps/app_tip223b 目录，执行命令 ./app_tip223b，用手指触摸按键，可以看到控制台打印 the key is to be pressed，不触摸按键不打印信息

4.6 SylixOS 八位流水灯实验

4.6.1 实验目的

- 掌握通过 GPIO 来控制 LED 灯，实现流水灯的效果

4.6.2 实验内容

- 阅读 app_8bitled.c 程序代码，了解 LED 驱动的操作流程和相关函数使用
- 编译 app_8bitled 工程，并将程序上传到验证平台运行查看效果

4.6.3 实验环境

- 硬件：三基色 LED 模块，2K 龙芯派，杜邦线
- 软件：RealEvo-IDE、putty

4.6.4 实验原理

8 个三基色（红绿蓝）全彩 LED，扫描式控制方式，与数码管控制方式相同，位选控制对应 LED 点亮，段选控制颜色，通过不同的程序可以产生不同的颜色变化效果。

接线方法：

模块 D0 引脚连接到 J13 的 16 引脚；模块 D1 引脚连接到 J13 的 14 引脚；

模块 D2 引脚连接到 J13 的 12 引脚；模块 D3 引脚连接到 J13 的 10 引脚；

模块 D4 引脚连接到 J13 的 8 引脚；模块 D5 引脚连接到 J13 的 6 引脚；

模块 D6 引脚连接到 J14 的 9 引脚；模块 D7 引脚连接到 J14 的 11 引脚；

另外旁边有三个 RGB 的脚位，需要跟 GND 相连，跟 R 相连就亮红色，跟 G 相连就亮绿色，跟 B 相连就亮蓝色，三个同时连起来就全都亮。龙芯派 J13 和 J14 的 2、4、18、20 引脚都是 GND，VCC 引脚连接到 J13 的 1 引脚。

程序清单 app_8bitled.c

```
#include <stdio.h>

#define LED_STATE      3           /* LED 状态          */
#define LED_ON         1           /* LED 亮            */
#define LED_OFF        0           /* LED 灭            */

int main (int argc, char **argv)
{
    INT    iFd[8], i;
    static CHAR    *_G_pcLedDevName[8] = {
        "/dev/led0", "/dev/led1", "/dev/led2", "/dev/led3",
        "/dev/led4", "/dev/led5", "/dev/led6", "/dev/led7",
    };
    while(1) {
        for (i = 0; i < 8; i++) {
            iFd[i] = open(_G_pcLedDevName[i], O_RDWR, 0666);
            if (iFd[i] < 0) {
                printf("failed to open %s\r\n", _G_pcLedDevName[i]);
            }
        }
    }
}
```



```

        return (PX_ERROR);
    }

    ioctl(iFd[i], LED_STATE, LED_ON);
    msleep(40);

    ioctl(iFd[i], LED_STATE, LED_OFF);
    msleep(40);

    close(iFd[i]);
}
}

return (0);
}

```

4.6.5 实验步骤

1. 使用 IDE System FTP 上传 ls2kdrv_led.ko 内核模块库到验证平台
2. 在控制台中，切换到/lib/modules/drivers 目录，执行命令 insmod ls2kdrv_led.ko
3. 切换到/apps/app_8bitled 目录，执行命令 ./app_8bitled；可以看到灯依次闪烁，效果像流水一样

4.7 SylixOS 蜂鸣器实验

4.7.1 实验目的

- 掌握蜂鸣器的操作原理和使用

4.7.2 实验内容

- 阅读 app_beep.c 程序代码，了解 beep 的操作流程和相关函数使用
- 编译 app_beep 工程，并将程序上传到验证平台运行查看效果

4.7.3 实验环境

- 硬件：蜂鸣器，2K 龙芯派，杜邦线
- 软件：RealEvo-IDE、putty

4.7.4 实验原理

硬件接线：

模块 VCC 引脚连接到 J13 的 1 引脚；模 GND 引脚连接到 J13 的 2 引脚；

模块 I/O 引脚连接到 J14 的 11 引脚。

模块采用 S8550 三极管驱动，当 I/O 口输入低电平时，蜂鸣器发声。

程序清单 app_beep.c

```

#include <stdio.h>

#define BEEP_RING          0                /* BEEP 鸣叫命令 */

int main (int argc, char **argv)

```

```

{
    INT      iFd;
    ULONG    ulMsTime;
    INT      i;

    iFd = open("/dev/beep", O_RDWR, 0666);
    if (iFd < 0) {
        printf("failed to open /dev/beep\n");
        return (-1);
    }

    for (i = 0; i < 10; i++) {
        ulMsTime = 100;
        ioctl(iFd, BEEP_RING, ulMsTime);

        sleep(1);
    }

    sleep(2);
    close(iFd);

    return (0);
}

```

4.7.5 实验步骤

1. 使用 IDE System FTP 上传 ls2kdrv_beep.ko 内核模块库到验证平台
2. 在控制台中，切换到/lib/modules/drivers 目录，执行命令 insmod ls2kdrv_beep.ko
3. 切换到/apps/app_beep 目录，执行命令 ./app_beep；可以听到蜂鸣器鸣叫