

Tiny4412 用户手册

版本：2013-08-05

(本手册适用于 Tiny4412 开发板)



Copyright@2013



追 求 卓 越 创 造 精 品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

版权声明

本手册版权归属广州友善之臂计算机科技有限公司（以下简称“友善之臂”）所有，并保留一切权力。非经友善之臂同意(书面形式)，任何单位及个人不得擅自摘录本手册部分或全部，违者我们将追究其法律责任。

敬告：

在售开发板的手册会经常更新，请在 <http://www.arm9.net> 网站查看最近更新，并下载最新手册，不再另行通知。



更新说明:

日期	说明
2013-08-05	<p>2013-08-05更新如下:</p> <ol style="list-style-type: none">1)将Android无操作时自动关屏的时间默认延长至3周;2) 增加背光调节功能;3) 增加支持中兴MF210联通3G模块, 支持开机自动3G拨号上网,支持短信息收发 ;4) 增加完全开源的U-boot-Tiny4412;5) 增加Superboot-4412:<ul style="list-style-type: none">- 支持通过脚本指定烧写文件和启动参数- 支持SD卡脱机快速烧写- 支持烧写进度LCD显示- 支持蜂鸣器和LED状态指示- 支持烧写EXT3/EXT4文件系统6) 优化了HDMI输出, 默认输出分辨率为720P, 连接高清电视可即插即用; 配合HD700(1280x720)使用效果最佳, 可与LCD完美同步显示;7) 增加三星原厂资料,供用户参考;8) 优化了GPU驱动, 2D/3D性能提升40%, 配HD700高清屏, 安兔兔跑分可达14960, 流畅度与Galaxy S3手机相当。
2013-07-09	本手册第一次发布



目录

TINY4412 用户手册.....	- 1 -
第一章 TINY4412 开发板介绍.....	- 6 -
1.1 TINY4412 核心板介绍.....	- 7 -
1.1.1 Tiny4412 核心板资源特性	- 7 -
1.1.2 Tiny4412 核心板引脚定义	- 8 -
1.1.3 Tiny4412 核心板接口说明	- 11 -
1.2 TINY4412 SDK 底板介绍.....	- 12 -
1.2.1 Tiny4412 SDK 底板硬件资源特性	- 13 -
1.2.2 Tiny4412 SDK 布局及跳线	- 14 -
1.3 TINY4412 开发底板接口说明.....	- 15 -
1.3.1 电源接口和插座.....	- 15 -
1.3.2 串口.....	- 16 -
1.3.3 USB 接口	- 17 -
1.3.4 网络接口.....	- 18 -
1.3.5 音频接口.....	- 18 -
1.3.6 用户按键.....	- 19 -
1.3.7 LCD 接口.....	- 19 -
1.3.8 ADC 输入	- 20 -
1.3.9 PWM 控制蜂鸣器	- 20 -
1.3.10 I2C-EEPROM	- 21 -
1.3.11 SD 卡.....	- 21 -
1.3.12 GPIO/SDIO 接口	- 21 -
1.3.13 CMOS CAMERA 接口	- 22 -
1.3.14 CPLD-JTAG 接口.....	- 23 -
1.4 TINY4412 的软件特性.....	- 23 -
1.4.1 Android 4.2.1 系统资源特性.....	- 23 -
第二章 准备工作.....	- 26 -
2.1 开发板设置及连接.....	- 26 -
2.1.1 启动模式选择.....	- 26 -
2.1.2 外部接口连接.....	- 27 -
2.1.3 设置超级终端.....	- 27 -
2.2 制作用于安装系统的SD卡.....	- 31 -
2.2.1 使用SD-Flasher工具烧写Superboot到SD卡	- 31 -
2.2.2 恢复SD卡到原始状态.....	- 34 -
2.2.3 注意事项.....	- 35 -
2.2.4 复制光盘的images目录到SD卡上	- 36 -



追求卓越 创造精品

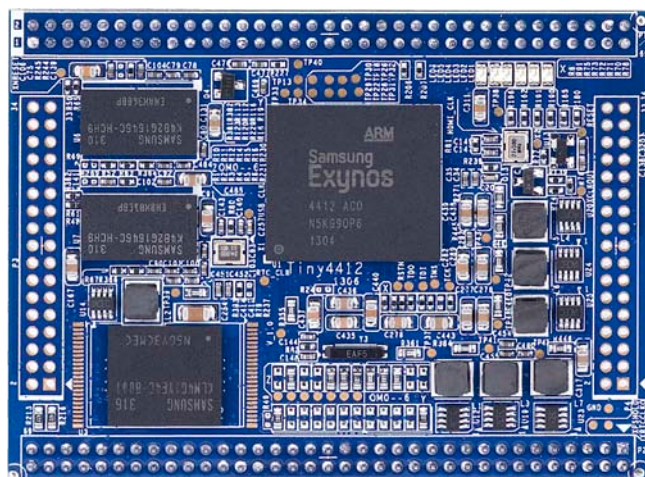
TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

2.3 使用SD卡脱机烧写系统.....	- 36 -
第三章 ANDROID使用指南	- 38 -
3.1 ANDROID的使用	- 38 -
3.1.1 Anroid下的按键	- 38 -
3.1.2 Android日常开发常用命令.....	- 38 -
3.1.3 利用重力感应模块自动旋转屏幕.....	- 39 -
3.1.4 访问SD卡.....	- 40 -
3.1.5 播放mp3.....	- 40 -
3.1.6 调节音量大小.....	- 41 -
3.1.7 录音功能.....	- 41 -
3.1.8 使用WiFi无线上网.....	- 42 -
3.1.9 3G拨号上网及短信息收发	- 44 -
3.1.10 HDMI同步输出高清图像到电视	- 46 -
3.1.11 播放高清视频.....	- 46 -
3.1.12 背光调节设置.....	- 47 -
3.1.13 串口助手.....	- 48 -
3.1.14 LED测试.....	- 50 -
3.1.15 PWM蜂鸣器测试	- 51 -
3.1.16 ADC测试	- 52 -
3.1.17 I2C-EEPROM测试	- 52 -
3.2 建立ANDROID编译环境	- 53 -
3.2.1 安装Ubuntu12.04.2 64bit系统.....	- 53 -
3.2.2 设置Ubuntu系统.....	- 59 -
3.2.3 以root用户登录	- 59 -
3.2.4 安装交叉编译器.....	- 62 -
3.2.5 解压安装Andorid4.2.1 源代码.....	- 64 -
3.3 配置和编译LINUX内核	- 64 -
3.4 从源代码开始创建ANDROID	- 65 -
3.5 制作安装或运行文件系统映像.....	- 66 -
3.6 在ANDORID程序中访问硬件	- 66 -
3.6.1 如何使用函数库(libfriendlyarm-hardware.so)?	- 66 -
3.6.2 函数库(libfriendlyarm-hardware.so)接口说明	- 68 -
3.6.3 示例程序说明.....	- 73 -

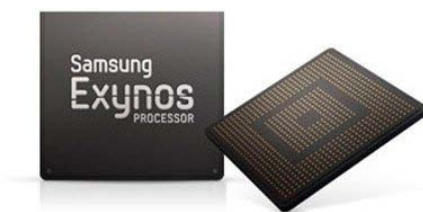
第一章 Tiny4412 开发板介绍



(图: Tiny4412 核心板)

Tiny4412 是高性能的 Cortex-A9 核心板, 它们由广州友善之臂设计、生产和发行销售。采用三星高性能的 Exynos 4412 四核处理器。

Samsung Exynos
4212 PROCESSOR



Exynos 4412 内部集成了 GPU 为 Mali-400 MP 的高性能图形引擎, 支持 3D 图形流畅运行, 并可播放 1080P 大尺寸高清视频, 流畅运行 Android 等高级操作系统, 非常适合开发高端物联网终端, 广告多媒体终端, 智能家居, 高端监控系统, 游戏机控制板等设备。

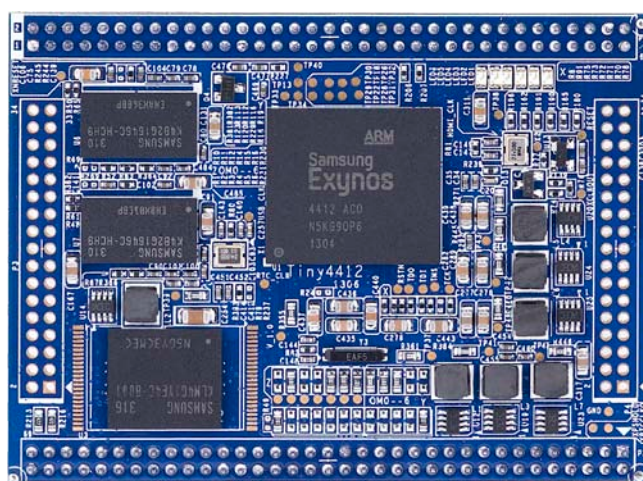
用户可以到我们网站浏览最新通知及下载更新最新的手册和系统网址: <http://www.arm9.net>

1.1 Tiny4412 核心板介绍

Tiny4412 核心板也是采用了 2.0mm 间距的双排针(P1, P2, P3, P4)，引出了大部分 CPU 功能引脚，尺寸为(74x55mm)。其中 P1 和 P2 排针为标配焊接，它们已经包含了大部分常用的功能；P3 和 P4 空焊，方便用户扩展开发使用。

Tiny4412 标配 1G DDR3 内存和 4GB 高速 eMMC 闪存。

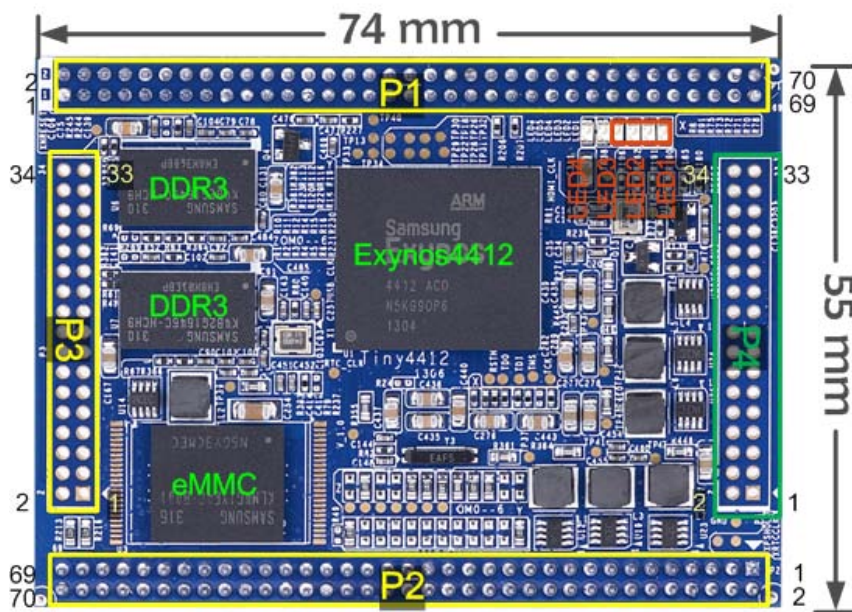
1.1.1 Tiny4412 核心板资源特性



Item	Description
CPU 处理器	<ul style="list-style-type: none"> ● Samsung ARM Cortex-A9 四核 Exynos 4412 Quad-core 处理器，运行主频 1.5GHz ● 内置 ARM Mali-400 双核 GPU ● 最高可支持 1080p@30fps 硬件解码视频流畅播放，格式可为 MPEG4, H.263, H.264 等 ● 最高可支持 1080p@30fps 硬件编码(Mpeg-2/VC1)视频输入
DDR3 RAM 内存	<ul style="list-style-type: none"> ● Size: 1G ● 32bit 数据总线，单通道
FLASH 存储	<ul style="list-style-type: none"> ● 标配 4GeMMC
接口资源	<ul style="list-style-type: none"> ● 2 个 70 Pin 2.0mm space DIP connector ● 2 个 34 Pin 2.0mm space DIP connector

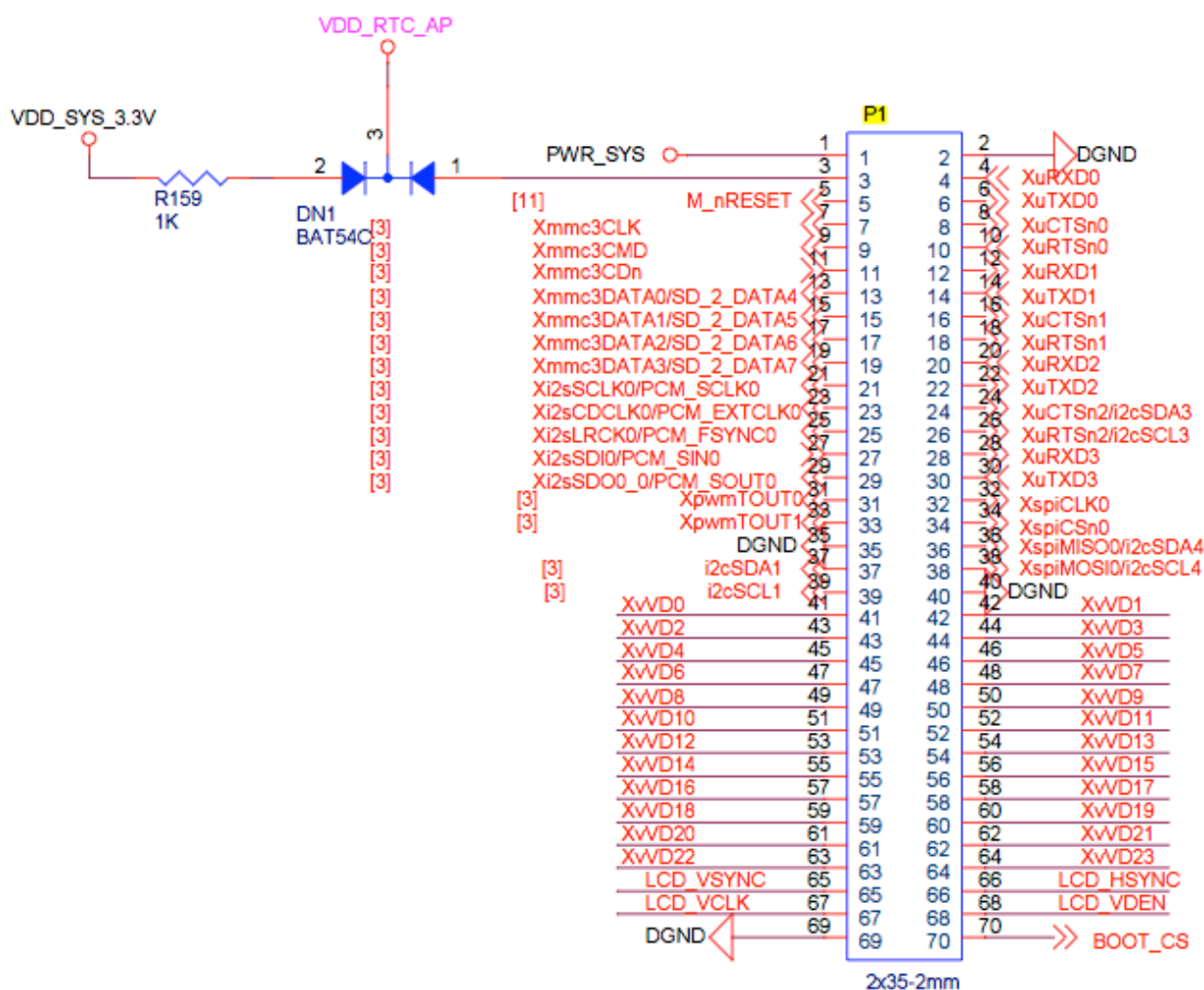
在板资源	<ul style="list-style-type: none"> 4 x User Leds(Green) 2 x Power Led(Green)
电源供电	<ul style="list-style-type: none"> Supply Voltage from 3.7V to 6V(支持睡眠唤醒模式)
PCB 规格尺寸	<ul style="list-style-type: none"> 6 层高密度电路板，采用沉金工艺生产 Size: 74 x 55 11(mm)

1.1.2 Tiny4412 核心板引脚定义



各接口定义：

P1





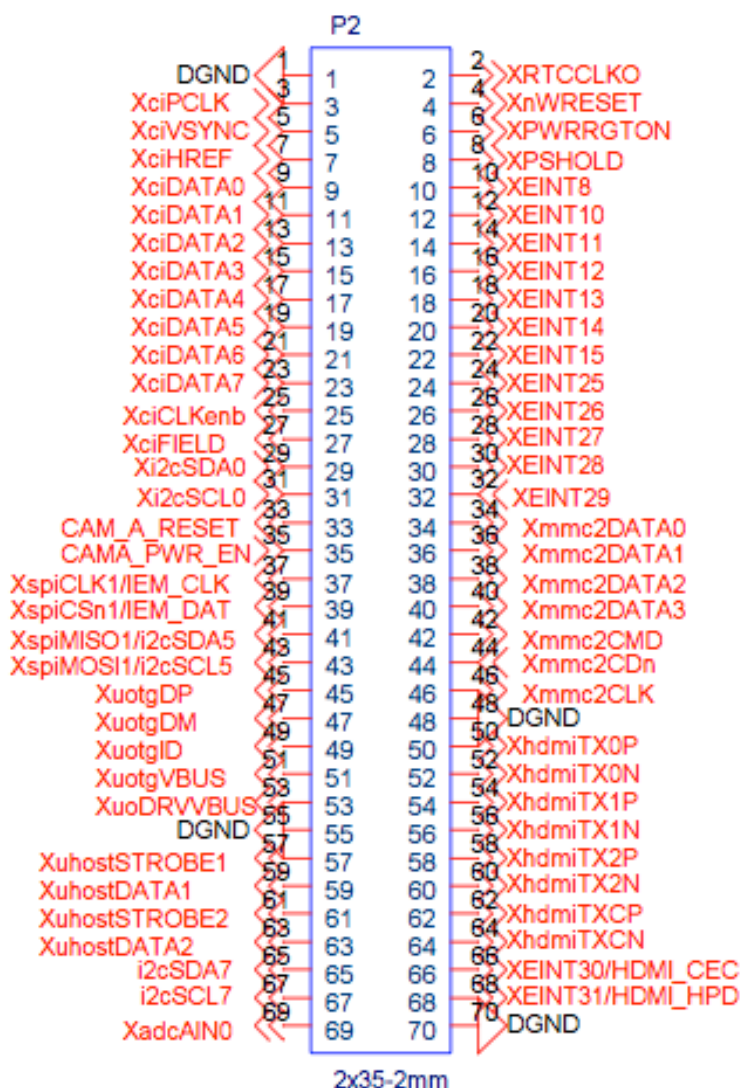
追求卓越 创造精品

TO BE BEST

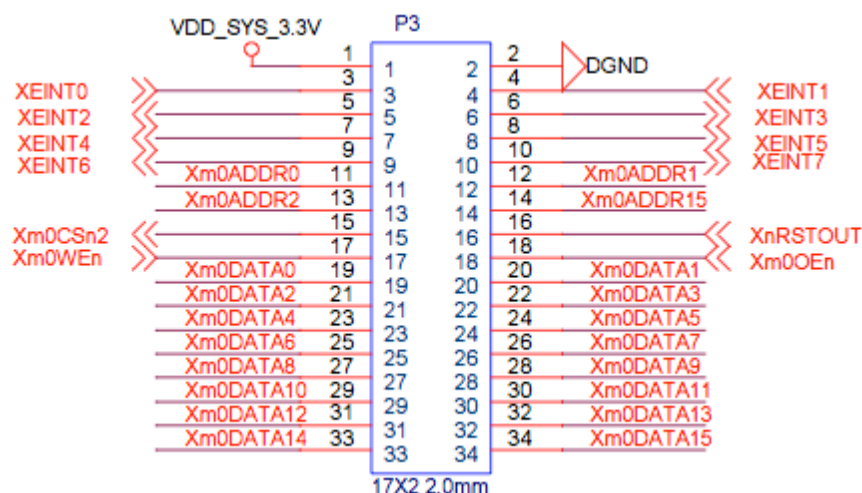
TO DO GREAT

广州友善之臂计算机科技有限公司

P2



P3



地址：广州市天河区龙口西路龙苑大厦A1栋1705

网址：<http://www.arm9.net>

电话：+86-20-85201025(售前、售后咨询)

技术支持(Tel): 13719442657

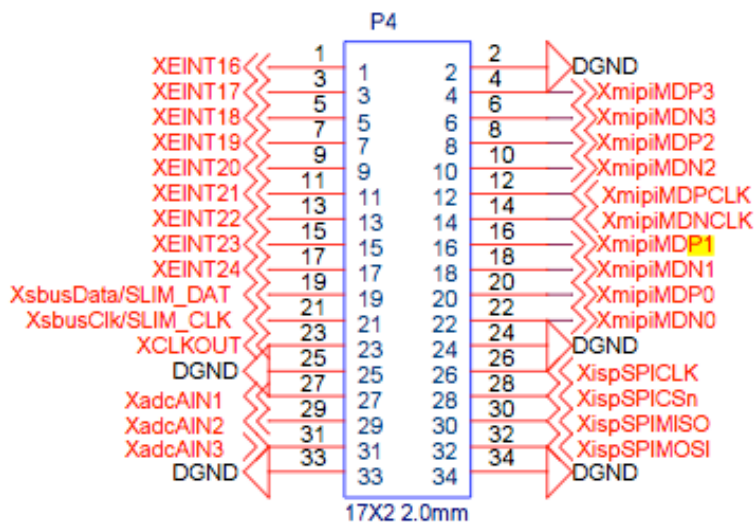
传真：+86-20-85261505

E-Mail: capbily@163.com(商务或项目合作)

dev_friendlyarm@163.com (技术支持)

第- 10 -页


P3



1.1.3 Tiny4412 核心板接口说明

1.1.3.1 用户LED

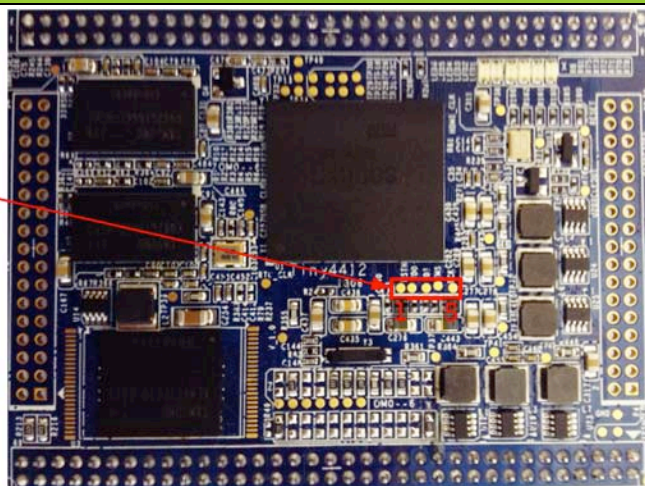
LED 是开发中最常用的状态指示设备,本开发板具有 4 个用户可编程 LED,它们直接与 CPU 的 GPIO 相连接,低电平有效(点亮),详细的资源占用如下表。

					
		LED1	LED2	LED3	LED4
	GPIO 资源	GPJ_0	GPJ_1	GPJ_2	GPJ_3

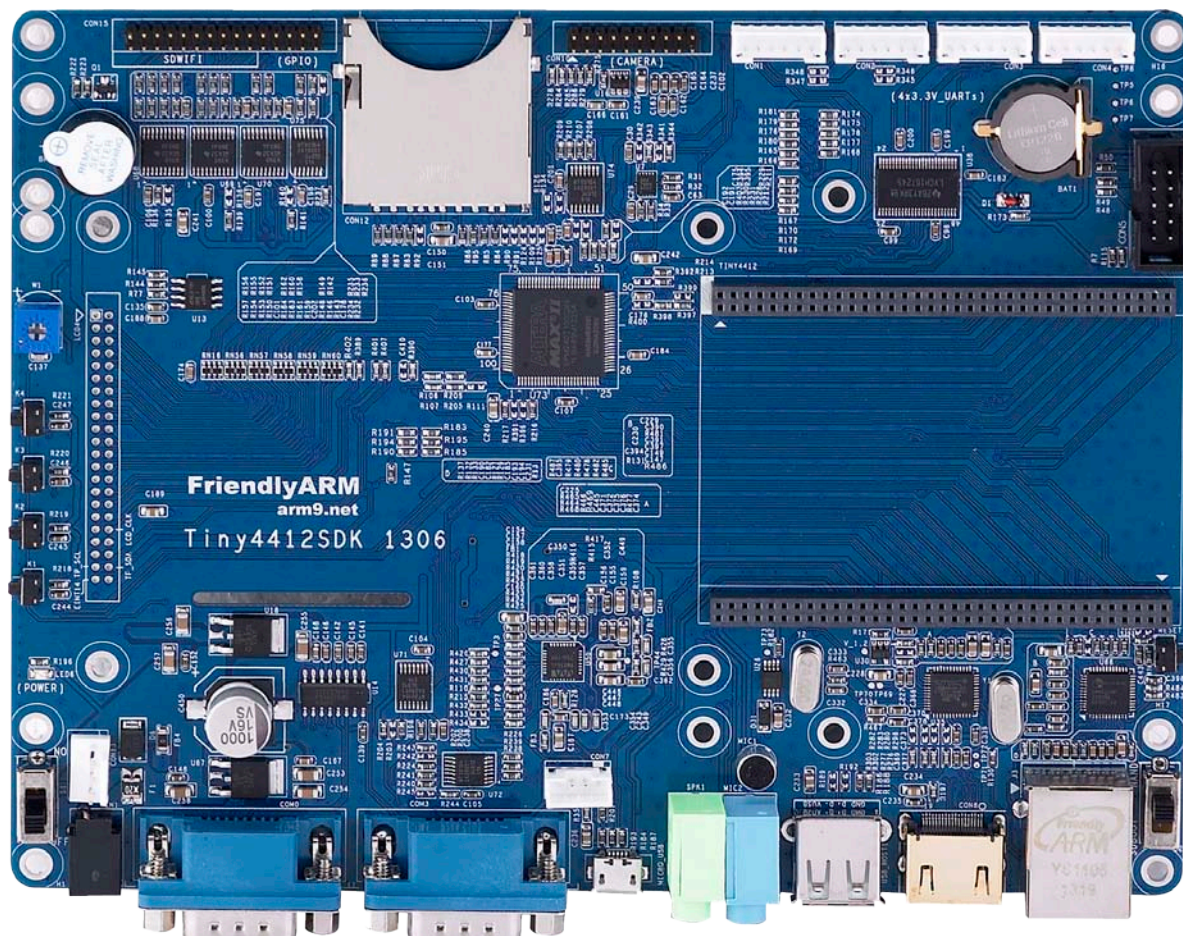
1.1.3.2 Jtag口接口说明

目前大部分高端 CPU 都可以支持 SD 卡启动, JTAG 接口已经用途不大,并且芯片厂商提供的 JTAG 接口资料和软件也更少,甚至是没有。但仍然有些用户钟情于 JTAG 调试开发,因电路板尺寸有限, Tiny4412 特留出相应的 JTAG 测试点,有需要的用户可以自行引出使用,如下图所示。

- 1) XjTDO
- 2) XjTDi
- 3) XjTMS
- 4) XjRStn
- 5) XjTCK



1.2 Tiny4412SDK底板介绍



1.2.1 Tiny4412 SDK底板硬件资源特性

LCD 显示	<ul style="list-style-type: none"> ● LCD1(背面): 45Pin, 0.5mm 间距, 兼容 Mini2440/Mini6410/Mini210 LCD 显示屏, 支持一线协议背光调节和电容触摸 ● LCD3(背面): 40Pin, 0.5mm 间距, 兼容 Mini2440/Mini6410/Mini210 LCD 显示屏, 支持一线触摸 ● LCD4(正面未引出): 44Pin, 兼容 Mini2440/Mini6410/Mini210 LCD 显示屏, 支持一线触摸和电容触摸 ● HDMI 高清接口(Type A) ● LCD 可支持从 3.5 寸到 12.1 寸, 支持高清屏
网络	<ul style="list-style-type: none"> ● 1 个 10/100M 自适应以太网 RJ45 接口(采用 DM9621)



追求卓越 创造精品

TO BE BEST

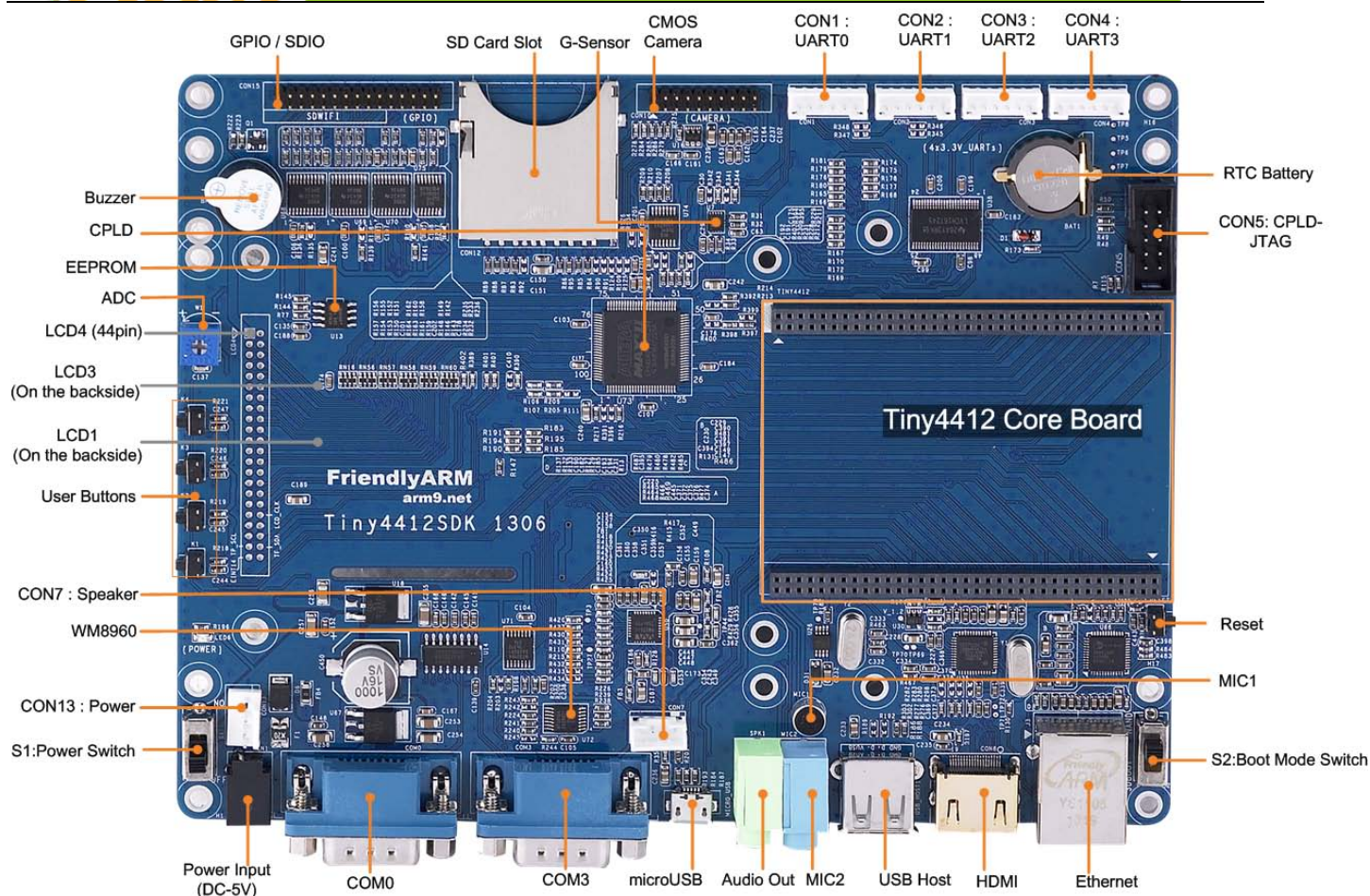
TO DO GREAT

广州友善之臂计算机科技有限公司

标准接口资源	<ul style="list-style-type: none">● 2 个 DB9 式 RS232 串口(另有 4 个 TTL 电平串口)● 1 个 micro USB Slave 2.0 接口● 1 路 3.5mm 立体声音频输出接口, 1 路在板麦克风输入● 1 路 USB Host 2.0 接口● 1 个标准 SD 卡座● 5V 直流电压输入:接口座型号为 DC-23B
在板即用资源	<ul style="list-style-type: none">● 1 个 I2C-EEPROM 芯片(256byte), 主要用于测试 I2C 总线● 4 个用户按键(中断式资源引脚)● 1 个 PWM 控制蜂鸣器● 板载实时时钟备份电池● 板载重力感应芯片
外扩接口资源	<ul style="list-style-type: none">● 4 个串口座 :TTL 电平● 1 个 GPIO 接口 (含 SDIO)● 1 个 CMOS 摄像头接口
PCB 规格尺寸	<ul style="list-style-type: none">● 层数 :2● Size: 180 x 140(mm)
软件支持	<ul style="list-style-type: none">● Linux Kernel 3.5● Android 4.2.1

1.2.2 Tiny4412 SDK布局及跳线

Tiny4412 底板接口布局如下图所示。



1.3 Tiny4412 开发底板接口说明

本小节详细介绍了开发板上每个接口或模块的引脚定义和占用的 CPU 资源，光盘中另有本开发板的完整 PDF 格式原理图，以供参考使用。

更详细的接口和引脚说明，请参考光盘中的原理图，设计参考也应以原理图为准。

1.3.1 电源接口和插座

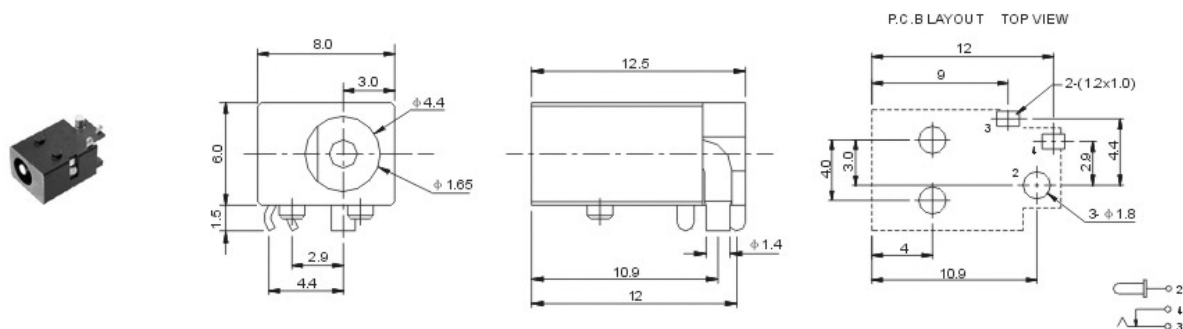
本开发板采用 5V 直流电源供电，提供了 2 个电源输入口，CN1 为附带的 5V 电源适配器插座，白色的 CON5 为 4Pin 插座，方便板子放入封闭机箱时连接电源。

CON13	NO.	引脚定义
-------	-----	------

	1	VDD5V
	2	GND
	3	GND
	4	VDDIN
说明：此种接法方便当采用引线连接时，电源开关 S1 也是有效的。		

电源座型号及尺寸

Type: DC023B

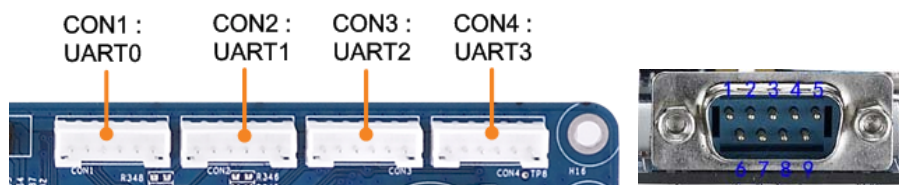


1.3.2 串口

Exynos4412 本身总共有 4 个串口，其中 UART1 为四线的功能串口，UART0、2、3 为两线串口。

在本开发板上，UART0 和 UART3 已经经过 RS232 电平转换，它们分别对应于 COM0 和 COM3，你可以通过附带的交叉串口线和 PC 互相通讯。

CON1, CON2, CON3, CON4 在开发板上的位置和原理图中的连接定义对应关系如下图所示。



CON0 ~ CON4	引脚定义(TTL)	COM0	引脚定义(RS232)
----------------	-----------	------	-------------

1	RTSn	1	NC
2	CTSn	2	RSRXD
3	TXD	3	RSTXD
4	RXD	4	NC
5	5V	5	GND
6	GND	6	NC
		7	NC
		8	NC
		9	NC


COM3	引脚定义(RS232)
1	NC
2	RSRXD
3	RSTXD
4	NC
5	GND
6	NC
7	RSCTSn
8	RSRTSn
9	NC

1.3.3 USB接口


本开发板具有一个 USB Host(2.0)接口，它和普通 PC 的 USB 接口是一样的，可以接 USB 摄像头、USB 键盘、USB 鼠标、优盘等常见的 USB 外设；

本开发板还带有一个 microUSB(2.0)接口，主要用于 Android 系统下的 ADB 功能，用于软件安装和程序调试。

microUSB 的接口定义如下：

	miniUSB	引脚定义
	5	GND
	4	OTGID
	3	D+
	2	D-
	1	Vbus

USB Host 的接口定义如下:

	USB Host	引脚定义
	1	5V
	2	D-
	3	D+
	4	GND

1.3.4 网络接口

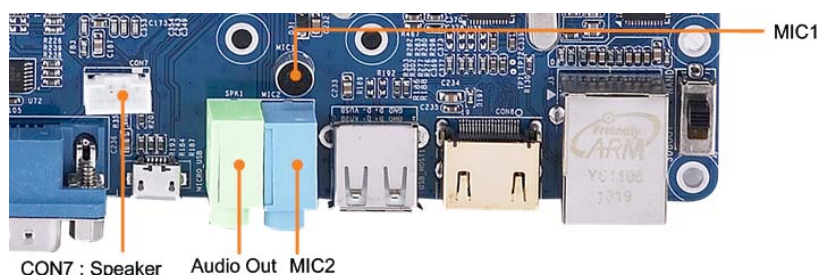
本开发板的有线网络采用了 DM9621 网卡芯片, 它可以自适应 10/100M 网络, RJ45 连接头内部已经包含了耦合线圈, 因此不必另接网络变压器, 使用普通的网线即可连接本开发板至你的路由器或者交换机。

1.3.5 音频接口

Exynos4412 支持 I2S/PCM/AC97 等音频接口, 本开发板采用的是 I2S0 接口, 它外接了 WM8960 作为 CODEC 解码芯片, 可支持 HDMI 音视频同步输出, WM8960 芯片在 Tiny4412SDK 底板上。

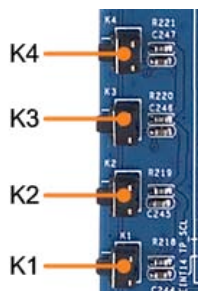
音频系统的输出为开发板上的常用 3.5mm 绿色孔径插座。

为方便学习开发使用, 我们直接在板上提供了麦克风输入, 注意: 本开发板并非专业的录音设备, 音频输入的处理电路很简单, 录音时尽量把音源靠近麦克风。



1.3.6 用户按键

本开发板总共有 4 个用户测试用按键,它们均从 CPU 中断引脚直接引出,属于低电平触发,8 个按键的定义如下:



按键	K1	K2	K3	K4
对应的中断	EINT26	EINT27	EINT28	EINT29
可复用为 GPIO	GPX3_2	GPX3_3	GPX3_4	GPX3_5

1.3.7 LCD接口

为了方便用户使用, Tiny4412SDK 带有三个 LCD 接口, 其中一个 45pin (LCD1), 可连接电容触摸屏。

LCD 接口座中包含了常见 LCD 所用的大部分控制信号(行场扫描、时钟和使能等), 和完整的 RGB 数据信号(RGB 输出为 8: 8: 8, 即最高可支持 1600 万色的 LCD); 为了用户方便试验, 还引出了 PWM 输出, 和复位信号(nRESET), 其中 LCD_PWR 是背光开关控制信号。

注意: 因为采用了一线精准触摸, LCD1 座中并不包含 CPU 自带的四线电阻触摸引脚, 而是增设了 I2C 和中断脚(见 LCD1-41、42、43、44), 这样设计是为了连接电容触摸屏。

LCD1	引脚说明	LCD1	引脚说明
1	VDD_5V	2	VDD_5V
3	VD0	4	VD1
5	VD2	6	VD3
7	VD4	8	VD5
9	VD6	10	VD7
11	GND	12	VD8
13	VD9	14	VD10
15	VD11	16	VD12
17	VD13	18	VD14

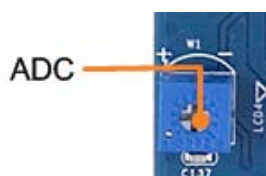
19	VD15	20	GND
21	VD16	22	VD17
23	VD18	24	VD19
25	VD20	26	VD21
27	VD22	28	VD23
29	GND	30	PWM1/GPD0_1
31	XEINT10/GPH1_2	32	nRSTOUT
33	VDEN	34	VSYNC
35	HSYNC	36	VCLK
37	I2CSCL2	38	XEINT14/GPH1_6
39	I2CSDA2	40	XEINT15/GPH1_7
41	GND		

注意：Exynos4412 总共有 3 路 I2C，此处使用的是 I2C2

说明：各个引脚和 CPU 的连接关系，请以原理图为准，此处定义标称仅供参考

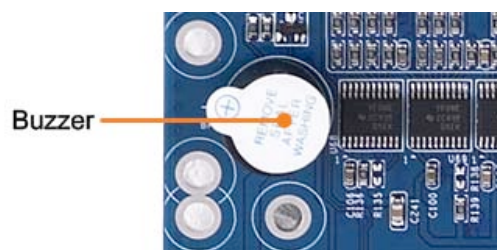
1.3.8 ADC输入

Tiny4412 核心板带有 4 路 ADC 转换通道，其中 AIN0 连接到了开发底板的可调电阻 W1，其它通道尚未在 Tiny4412SDK 底板上引出。



1.3.9 PWM控制蜂鸣器

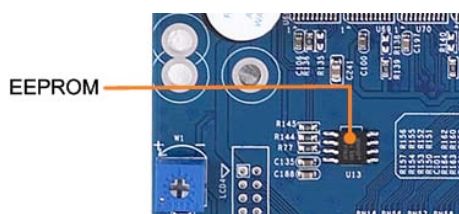
本开发板的蜂鸣器 Buzzer 是通过 PWM0 控制的，原理图如下所示，其中 PWM0 对应 GPD0_0，该引脚可通过软件设置为 PWM 输出，也可以作为普通的 GPIO 使用。



1.3.10 I2C-EEPROM

本开发板具有一个直接连接 CPU 之 I2C0 信号引脚的 EEPROM 芯片 AT24C08，它的容量有 256 byte，在此主要是为了供用户测试 I2C 总线而用，它并没有存储特定的参数。

注意：Exynos4412 总共有 8 路 I2C，此处使用了 I2C0

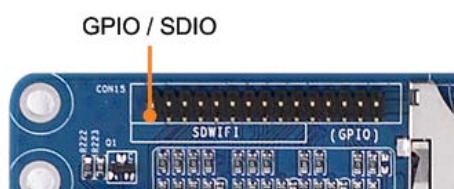


1.3.11 SD卡

Tiny4412 引出 2 路 SDIO 接口，在本开发底板中，SDIO2 被用作普通 SD 卡接口使用，该接口可以支持 SDHC，也就是高速大容量卡。

1.3.12 GPIO/SDIO接口

GPIO 是一个 2.0mm 间距的 30Pin 插针座，见下图：



其中,前 20pin 是一个 SDIO 接口，一般用于连接 SD-WiFi 模块。

GPIO 接口的引脚定义如下：

CON9	引脚定义	CON9	引脚定义
1	VDD3.3V	2	GND
3	TXD2	4	RXD2
5	I2CSCL	6	I2CSDA

7	SPIMOSI0	8	SPIMISO0
9	SPICLK0	10	SPICSn1
11	EINT13	12	EINT12
13	SD3_CLK	14	SD3_CMD
15	SD3_nCD	16	EINT11
17	SD3_DAT0	18	SD3_DAT1
19	SD3_DAT2	20	SD3_DAT3
21	SPIMISO1	22	EINT26
23	SPIMOSI1	24	EINT27
25	SPICLK1	26	EINT28
27	SPICSn1	28	EINT29
29	VDD5V	30	GND

说明：各个引脚和 CPU 的连接关系，请以原理图为准，此处定义标称仅供参考

1.3.13 CMOS CAMERA接口

Tiny4412 带有一个 CMOS 摄像头接口，它是一个 20 脚 2.0mm 间距的针座，用户可以直接使用我们提供的 CAM130 摄像头模块；其实 CAM130 摄像头模块上面没有任何电路，它只是一个转接板，它直接连接使用了型号为 ZT130G2 摄像头模块。

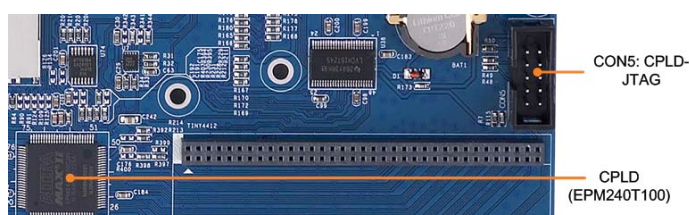
说明：CAMER 接口是一个复用端口，它可以通过设置相应的寄存器改为 GPIO 使用，下表是它对应引脚的 GPIO 列表



CMOS CAMERA			
CAMERA	引脚定义	CAMERA	引脚定义
1	I2CSDA0	2	I2CSCL0
3	XciFIELD	4	CAM_RESET/GPJ3_1
5	CAM_CLK	6	CAM_HREF
7	CAM_VSYNC	8	CAM_PCLK
9	CAM_DATA7	10	CAM_DATA6
11	CAM_DATA5	12	CAM_DATA4

13	CAM_DATA3	14	CAM_DATA2
15	CAM_DATA1	16	CAM_DATA0
17	VDD_3.3V	18	VDD_2.45-2.8V
19	VDD_1.8V	20	GND
说明：各个引脚和 CPU 的连接关系，请以原理图为准，此处定义标称仅供参考			

1.3.14 CPLD-JTAG接口



该接口用于烧写 Tiny4412SDK 上的 CPLD（EPM240T100）的固件。

1.4 Tiny4412 的软件特性

1.4.1 Android 4.2.1 系统资源特性

交叉编译器	arm-linux-gcc-4.5.1-v6-vfp	和 Mini6410/Mini210 通用，编译内核时会自动探测采用 armv7 指令集，支持硬浮点运算，独家提供
Bootloader	提供 Superboot，和开源的 uboot	
	支持 SD 卡脱机快速烧写，速度极快 (1.8M/秒)	
	支持 Android 标准的 fastboot USB 烧写系统	
	支持 SD 卡分区	



追 求 卓 越 创 造 精 品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

Android 内核	版本: Linux-3.5	
	支持 EXT3/YAFFS2/CRAMFS/FAT32 等格式的文件系统	
	看门狗驱动	
	RTC 驱动	
	4 个 LED 驱动	
	4 个用户按键驱动	
	SPI 驱动	
	I2C-EEPROM 驱动	
	PWM 控制蜂鸣器驱动	
	ADC 驱动(通道: AIN0)	
	CPU 本身所带触摸屏控制器驱动	
	电容屏触摸驱动	
	LCD 背光驱动, 支持 127 级可调	
	LCD 驱动(高清 7", 普屏 7")	
	USB Host 驱动: 支持优盘, 蓝牙等	
	USB Device 驱动: 支持 USB ADB/fastboot	
	SD 卡驱动	
	4 个串口驱动	
	USB WiFi 驱动: 内核自带, 对某些类型的卡支持不是太好	
	USB WiFi 驱动: 可支持更多型号 USB 无线网卡, 更加完善	
	音频驱动(WM8960: 支持录音和放音, ALSA 接口, 支持 D 类功放)	



追 求 卓 越 创 造 精 品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

	以太网网络 (DM9621)	
	FIMC 驱动	
	JPEG 驱动	
	MFC 多媒体驱动	
	HDMI 驱动	
	3D 加速	
	2D 加速	
	USB 转串口驱动	
Android 系统	版本: Android 4.2.1	
应用特性	支持 2D/3D 加速	流畅运行各种 2D/3D 游戏
	支持 WiFi 上网	
	支持 3G 上网和短信息收发	支持中兴 MF210 3G 模块
	支持 HDMI 音视频同步输出	可设置分辨率, 最大支持 1080p
	支持背光 127 级可调, 实现炫酷渐熄效果	
	等等...	

第二章 准备工作

出厂之前，如果客户未加说明，开发板中已经烧写了缺省的 Android4.2.1 系统(对应光盘 /images/Android 文件夹中二进制文件:zImage、ramdisk-u.img、system.img，这样你拿到板子之后即可上电开机运行系统，以便了解和测试整个硬件系统。

在开始之前，我们建议你先了解一下本章 1, 2 节的内容，一般将来做参考。

2.1 开发板设置及连接

2.1.1 启动模式选择

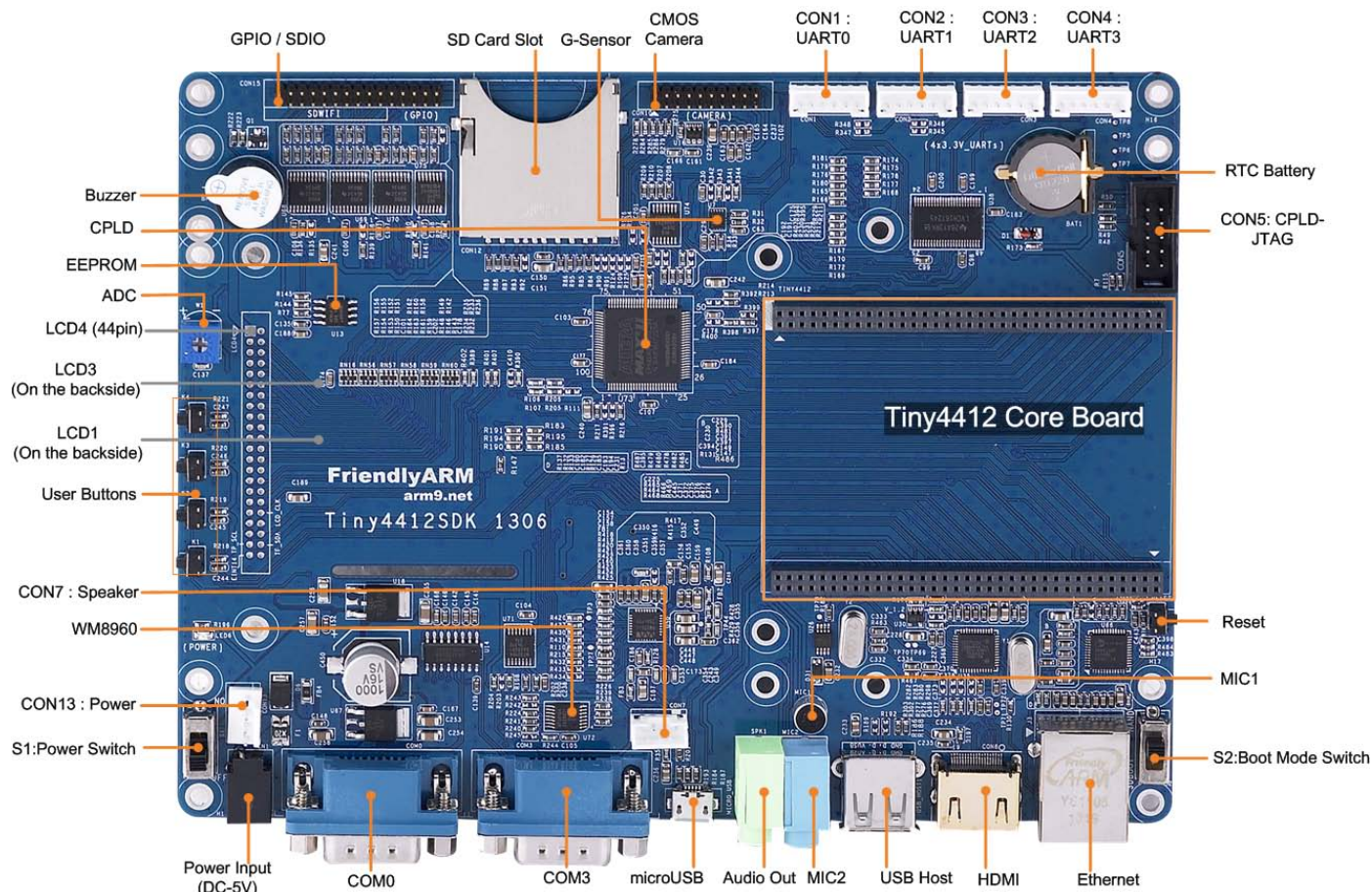
Tiny4412 支持 SD 卡和 eMMC 两种启动模式，通过 S2 开关来进行切换启动模式，如下图所示：

图示	说明	作用
	S2 拨至 NAND 标识一侧时，系统将从 eMMC 启动。	正常启动系统
	S2 拨至 SDBOOT 标识一侧时，系统将从 SD 卡启动；	用于烧写系统，或者从 SD 卡启动系统

一般开发板在日常使用时，S2 应拨向 NAND 那一侧，除非你需要烧写系统或者要从 SD 卡启动系统。

2.1.2 外部接口连接

Tiny4412SDK 有以下主要的外部接口：



初次使用 Tiny4412 开发板，可参照以下步骤连接开发板：

- 请使用我们提供的交叉串口线(蓝色头)连接开发板的串口 0(图中的 **COM0**)到 PC 机的串口
- 用我们提供的 5V 电源适配器连接到板上的 5V 输入插座(注意不要太用力，以免损坏电源插座)
- 把音箱或者耳机的插头接入板上的绿色音频输出口
- 如果您有液晶屏，请按照数据线头的方向与开发板的 LCD 接口相连 (LCD 接口在背面)

2.1.3 设置超级终端

注意：有的用户使用 USB 转串口线来扩展串口，但注意有的 USB 转串口线是会出现乱码的，这说明它的性能和功能并不好，我们的代理大部分都提供了这样的转接线，用户可以直接

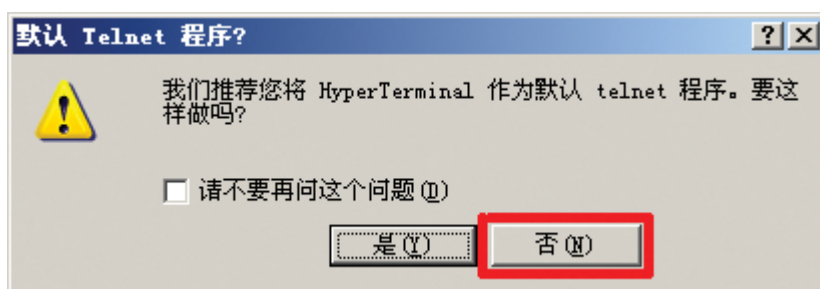
联系代理购买可用的转接线。

另外，请务必使用随机附带的串口交叉线，如果你使用其它的串口线，请使用万用表检测确定为交叉线即可。

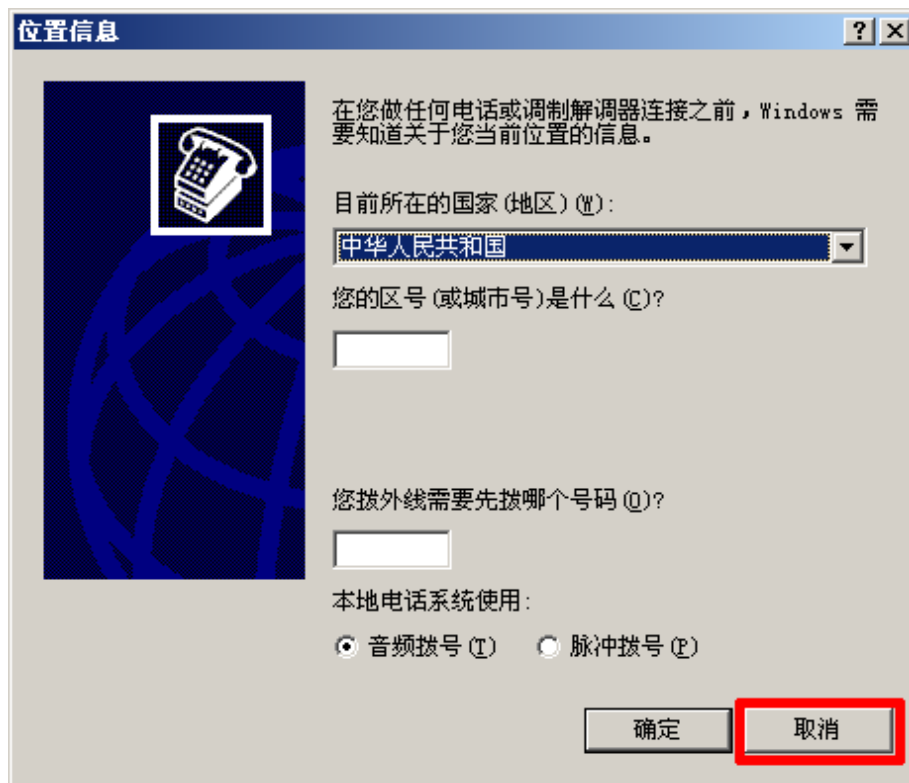
为了通过串口连接开发板，必须使用一个模拟终端程序，几乎所有的类似软件都可以使用，其中MS-Windows 自带的超级终端是最常用的选择，当你安装Windows9x 时需要自定义选择安装该项，Windows2000 及更高版本则已经缺省安装，Windows7可以使用putty。

一般桌面版Linux系统也自带了类似的串口终端软件，叫minicom，它是基于命令行的程序，对于初学者而言使用比较复杂一些，感兴趣的用户可以在网上找一下这方面的介绍。

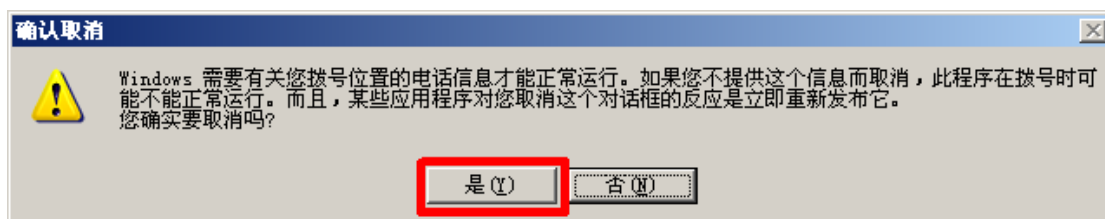
在此着重介绍一下Windows 自带的超级终端程序并以WindowsXP 为例，或许其他Windows 版本的程序界面有所不同。超级终端程序通常位于“开始->程序->附件->通讯”中，选择运行该程序，一般会跳出如图所示窗口，询问你是否要将Hypertrm作为默认的telnet程序，此时你不需要，因此点“否”按钮。



接下来，会跳出如下窗口，点“取消”



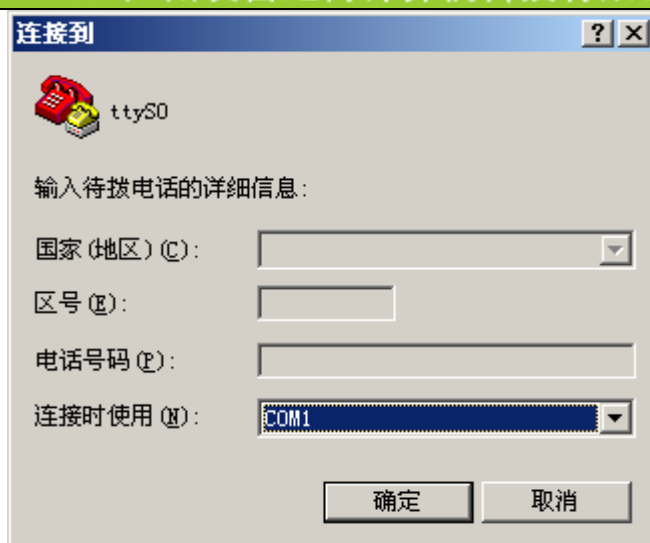
此时系统提示“确认取消”，点“是”即可，接着点提示窗口的“确定”，进入下一步。



超级终端会要求你为新的连接取一个名字，如图所示，这里我取了”ttyS0”，Windows 系统会禁止你取类似”COM1”这样的名字，因为这个名字被系统占用了。



当你命名完以后，又会跳出一个对话框，你需要选择连接开发板的串口，我这里选择了串口 1，如图所示：



最后，最重要的一步是设置串口，注意必须选择无流控制，否则，或许你只能看到输出而不能输入，另外板子工作时的串口波特率是 **115200**，如图所示。



当所有的连接参数都设置好以后，打开电源开关，终端会出现系统启动界面。

选择超级终端“文件”菜单下的“另存为...”，保存该连接设置，以便于以后再连接时就不必重新执行以上设置了。

2.2 制作用于安装系统的SD卡

一般 Tiny4412 出厂时已经安装好 Android4.2.1 系统，以后要重新烧写系统，则需要 SD 卡来烧写系统，无论是通过 USB 烧写，还是 SD 卡脱机烧写，都需要使用 SD 卡启动开发板来进行。

制作用于安装系统的 SD 卡的本质是：

将 Tiny4412 的 Bootloader(Superboot4412.bin)烧写到 SD 卡的第一个分区，这样以后 Tiny4412 就可以通过 SD 卡启动了，利用 Superboot 的 SD 卡脱机烧写功能，将系统烧写到 eMMC，达到重新安装系统的目的。

2.2.1 使用SD-Flasher工具烧写Superboot到SD卡

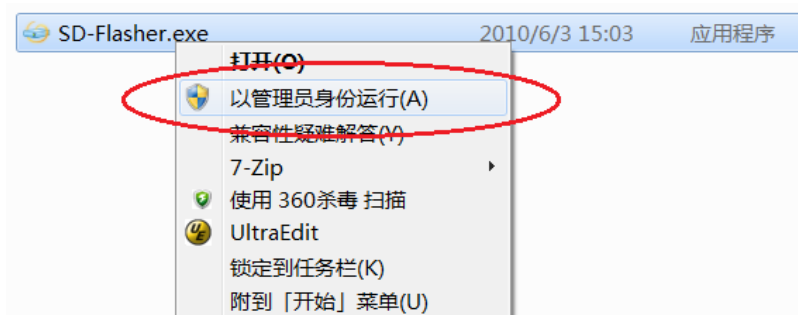
本步骤将执行以下操作：

SD-Flasher.exe 会对 SD 卡进行分区，第一个分区为 130M 用于存放 Superboot4412，剩下的空间格式化为 FRIENDLYARM 分区，用于存放系统文件 images。

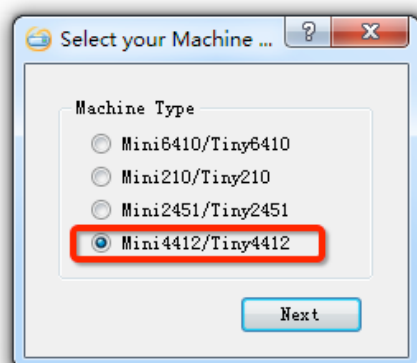
有些小于 256M 的 SD 卡是无法使用的，我们推荐使用至少 4G 的 SD 卡。

以下是详细步骤，基于 Windows7 环境：

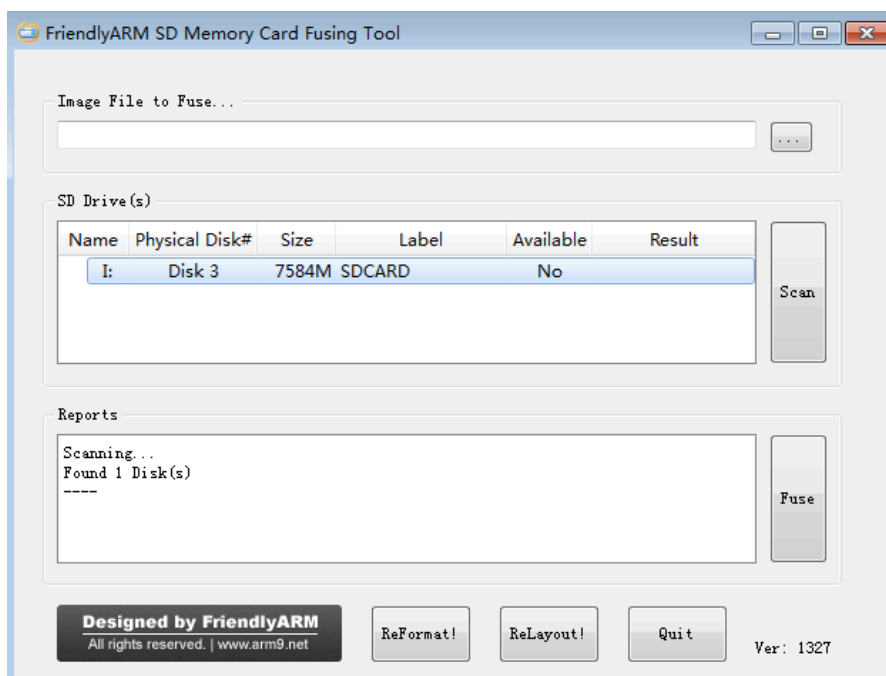
Step1: 打开光盘\tools\目录，解压 SD-Flasher-1327.7z 得到 SD-Flasher.exe 烧写软件,启动它，请注意，你需要通过管理员身份来打开该软件才行，如图



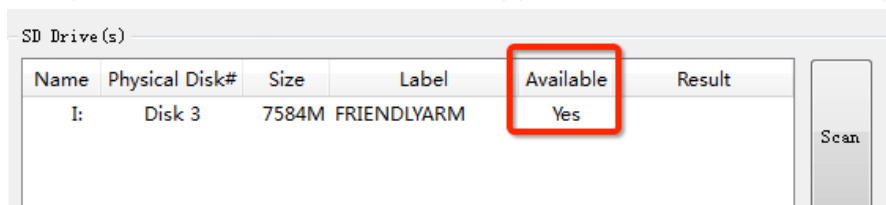
启动 SD-Flasher.exe 软件时，会弹出 “Select your Machine...” 对话框，请在其中选择 “Mini4412/Tiny4412” 项：




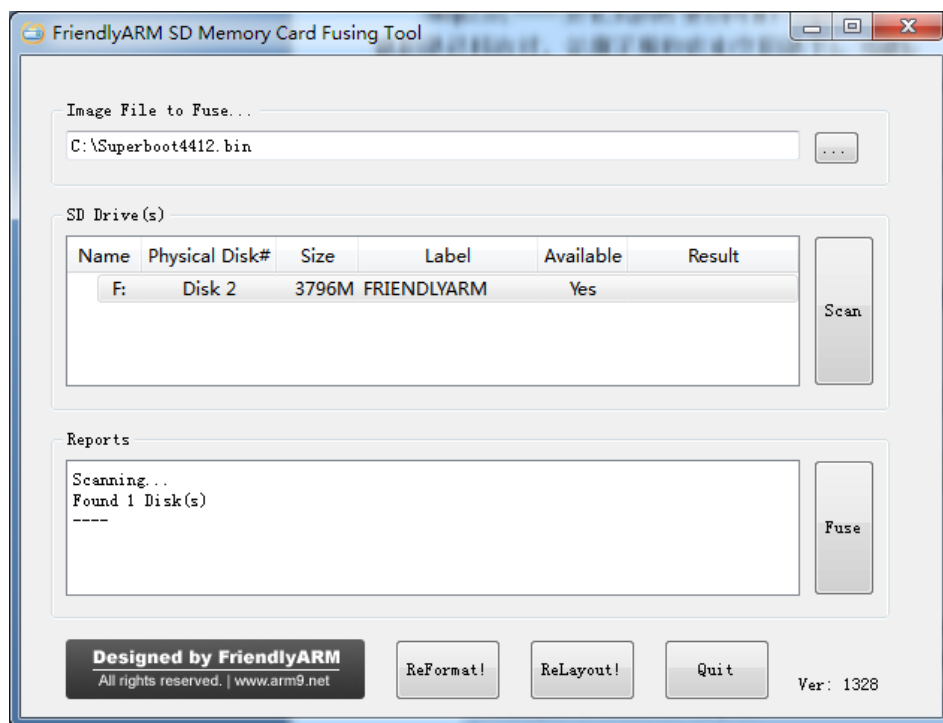
点 Next 后将弹出 SD-Flasher 主界面，点击一下 Scan，将列出你的 SD 卡，选中它，然后点“ReLayout”对 SD 卡进行分割，注意，本操作会将 SD 卡内的数据清空然后重新分区。



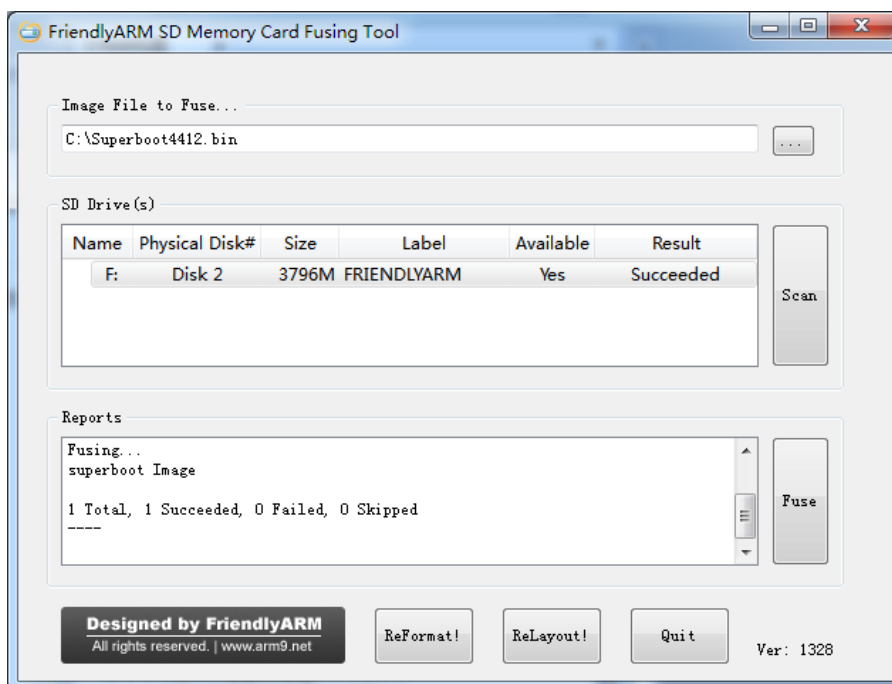
分割完成后，再点一下“Scan”，你会看到 Available 已变为可用状态，表示已分割完成：



Step2: 点  按钮找到所要烧写的 Superboot4412.bin(默认位于光盘 images/ 目录下，如光盘目录已移动过，注意不要放在中文目录下)，如图



Step3:点“Fuse”，Superboot4412.bin 就会被安全地烧写到 SD 卡的无格式区中了，以后你再使用 SD-Flasher 制作 SD 卡时，无需再执行 ReLayout 操作，FAT32 分区中的数据会得以保留。



Bootloader 被写入 SD 卡后是无法看到的，该如何检测呢？

方法是把 SD 卡插到已连接串口的开发板上，并把开发板上 S2 开关设置为“SDBOOT”模式，开机后，串口上有信息输出，例如：

```
Superboot-4412 V0.02(20130722) by FriendlyARM
```

表示 Bootloader 已被写入 SD 卡，如果串口没有输出，说明没有烧写成功。

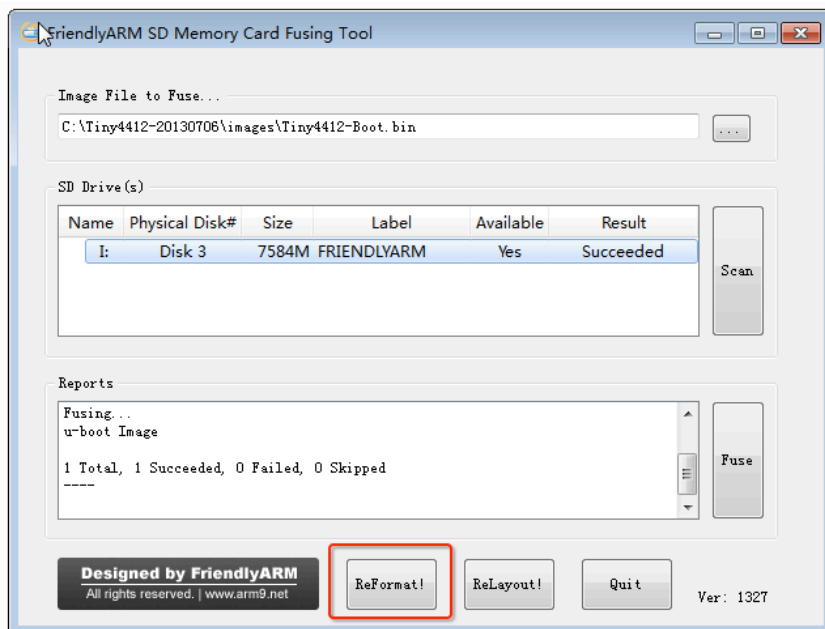
以下几种原因均有可能导致不行，并有解决方法：

1. 使用了笔记本自带的读卡器，建议使用外接的 usb 读卡器，注意，有的外接 usb 读卡器也是不行的
2. 使用了山寨的 SD 卡，请使用正品，最好是 4G 或以上的 SDHC
3. 使用了 microSD+卡套，请使用普通的 SD 卡，因为卡套的方式非常容易导致接触不良
4. 因接触不良导致无法使用 SD 卡启动的可能性比较大，建议多插拔几次试试，这包括(1)核心板和底板的接触，(2)SD 卡座本身的接触

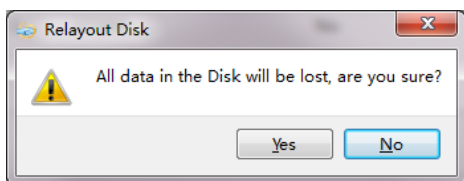
2.2.2 恢复SD卡到原始状态

此功能需要在 Windows7 下操作。

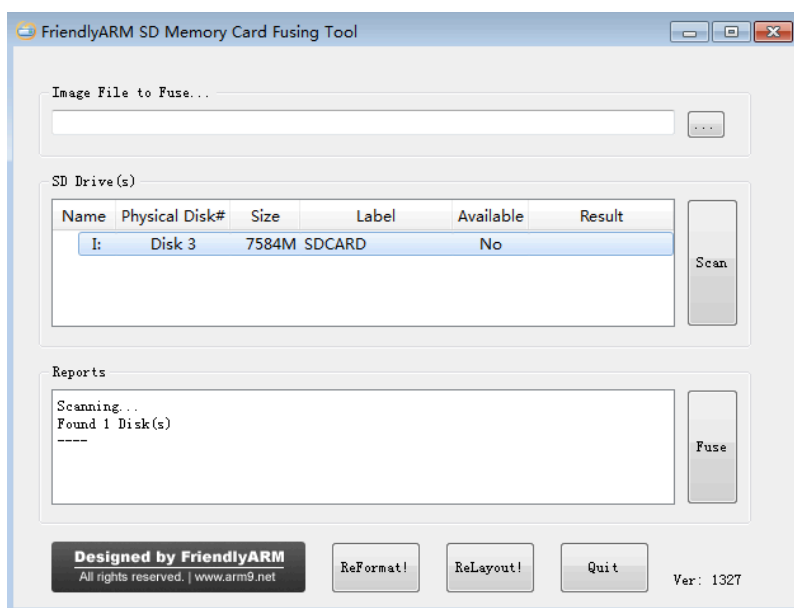
SD-Flasher.exe 会分割并预留 130M 空间用于烧写 Superboot4412.bin，当你的 SD 不再用于开发板时，你可能想恢复 SD 卡为原始状态，可参考上面章节的方法再次启动 SD-Flasher.exe，通过界面上的“ReFormat!”功能恢复 SD 卡为原始状态：



要恢复时，先点“Scan”扫描一下 SD 卡，然后点击“ReFormat!”按钮，会跳出一个提示框，如图，提示你 SD 卡中的所有数据将会丢失：



点“Yes”，开始恢复，这需要稍等一会，恢复完成后，再点一次“Scan”，这时 SD 卡的状态为不可烧写，即恢复到原始的状态了，如下图所示：



2.2.3 注意事项

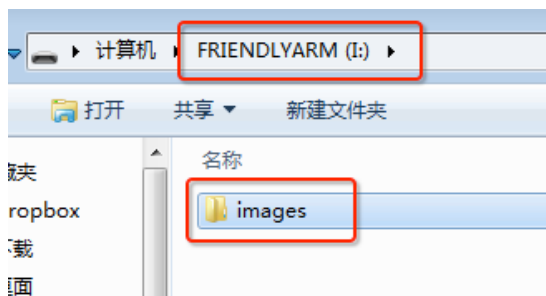
由于用户一般都会使用 SD 卡来存放一些其它的数据，因此，基于此考虑，SD-Flasher 软件在 Vista/Winows7 中运行时，会把 SD 卡自动分割为普通的 FAT32 格式区(自动命名卷标为“FriendlyARM)和无格式区(占 130M)两部分，烧写软件将会依据卷标名称作为标志，把 bootloader 烧写到无格式区，这样就不会破坏普通 FAT32 格式区中的数据了。

实际上，Vista/Windows7 系统本身的安全性很高，普通用户是无法在 Vista/Windows7 系统上强制烧写 SD 卡的，因此必须要先分割才能写入。

另外，烧写 Superboot4412.bin 请务必使用 Windows7 运行 SD-Flasher 来操作，如果使用 Windows XP，可能会烧写失败导致无法通过 SD 卡启动。

2.2.4 复制光盘的images目录到SD卡上

如需要使用 SD 卡脱机烧写系统功能，则可以将光盘上的 images 目录整个拷贝到 SD 卡的根目录下：



2.3 使用SD卡脱机烧写系统

注：此步骤假设你的 SD 已经烧写了 Superboot4412.bin，并且从光盘中拷贝了相应的烧写文件到 SD 卡上，拷贝方法是将光盘中整个 images 目录拷贝到 SD 卡的根目录下，步骤可参考章节：2.2。

Step1: 把 SD 卡插入电脑，双击打开”images\FriendlyARM.ini”文件，将 FriendlyARM.ini 的内容修改成如下内容(默认已经如此)：

```
#This line cannot be removed. by FriendlyARM(www.arm9.net)
```

```
CheckOneButton=No
```

```
Action = Install
```

```
OS = Android
```

```
LowFormat = No
```

```
VerifyNandWrite = No
```

```
LCD-Mode = No
```

```
CheckCRC32=No
```

```
StatusType = Beeper | LED
```



```
##### Android #####
```

```
Android-BootLoader = Superboot4412.bin
```

```
Android-Kernel = Android/zImage
```

```
Android-CommandLine = console=ttySAC0,115200n8 androidboot.console=ttySAC0
```

```
Android-RamDisk = Android/ramdisk-u.img
```

```
Android-RootFs-InstallImage = Android/system.img
```

Step2:检查 SD 上至少存在如下文件,如果没有,从光盘中拷贝到 SD 卡(将光盘的 images 目录整个目录拷到 SD 卡的根目录即可):

文件名	说明
images\Superboot4412.bin	Bootloader
images\Android\zImage	Android 内核(Linux Kernel 3.5)
images\Android\ramdisk-u.img	Android 根分区映象
Images\Android\system.img	Andorid 系统分区映象
images\FriendlyARM.ini	系统烧写配置文件

Step3: 取出 SD 卡插到开发板的 **SD 插槽**上, 参照下图将 S2 开关切换至 SD 卡启动, 然后上电开机, 开始烧写系统时, LCD 和串口终端会有进度显示。



Step4: 系统烧写完成后, 参照下图把开发板 S2 开关设置为“Nand”启动, 然后重新开机即可启动新的 Android 系统了。



第三章 Android使用指南

Tiny4412 选用 **Android4.2.1** 作为软件平台(Linux 内核版本为 3.5),并将我们在 6410 和 210 平台上针对 Android 所开发的增值软件移植到了 Tiny4412 中,强大的软件结合 Exynos4412 强大的硬件,满足你的项目需求的同时,也可让你专注于上层开发。

3.1 Android的使用

3.1.1 Anroid下的按键

关于 **Android** 下的按键布局

本开发板总共有 4 个用户按键,它们在 android 系统中的定义如下表所示:

按键编号	功能定义
K1	Back (返回键)
K2	Home (回到首页, 长按会列出后台程序)
K3	Menu (菜单键, 长按可旋转屏幕)
K4	OK (确定键)

(注: Android 可通过修改配置重新定义按键的含义)

3.1.2 Android日常开发常用命令

3.1.2.1 进入Android系统命令行并获得root权限

进入 Android 系统的命令行下,可方便进行的一些配置。

如要使用 USB 连接方式进入命令行,可先在你的电脑上安装 Android SDK,可自行网上找安装方法参考安装,安装后,输入 `adb shell` 即可进入 Android 系统命令行,在命令行下,是具有 root 权限的,可执行所有命令。

也可以通过串口终端来进入 Andorid 命令行,这种方法最简单,无需安装 Android SDK,但



默认情况下是没有 root 权限的，输入 su 命令一次即可获得 root 权限。

3.1.2.2 让system分区可读写

Android 系统默认情况下，system 分区是只读 mount 的，因为无法进行往里写数据的，可通过在板子上，输入以下命令重新 mount 一下 system 分区令其可读可写：

```
# mount -o remount /dev/block/mmcblk0p2 /system
```

该命令只对当前启动有效，重启后会恢复为只读。

如想 system 分区一直处于可读写，可能过修改 Android 源代码中的以下文件：

device/friendly-arm/tiny4412/fstab.tiny4412

将文件以下内容：

/dev/block/mmcblk0p2	/system	ext4	ro	wait
----------------------	---------	------	----	------

改为

/dev/block/mmcblk0p2	/system	ext4	rw	wait
----------------------	---------	------	----	------

即可。

3.1.2.3 从PC上传文件到开发板上

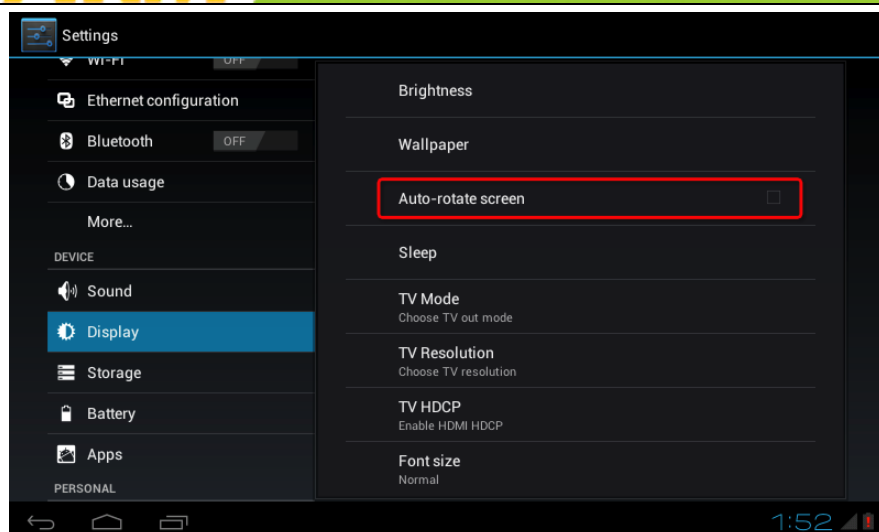
使用 Android SDK 中的 adb 命令，可将文件通过 USB 上传到开发板上，如要将 sensors.tiny4412.so 传到开发板的/system/lib/hw/目录，只要执行以下命令即可：

```
adb push sensors.tiny4412.so /system/lib/hw/
```

注意，上传文件到 system 目录，需要参考上一个章节将 system 分区设置为可读可写才可以。

3.1.3 利用重力感应模块自动旋转屏幕

Tiny4412SDK 带重力感应模块，无需设置，默认即可支持利用重力感应自动旋转屏幕，如果要禁用此功能，进入 Android 的设置->Display（显示），关闭自动屏幕旋转选项即可。

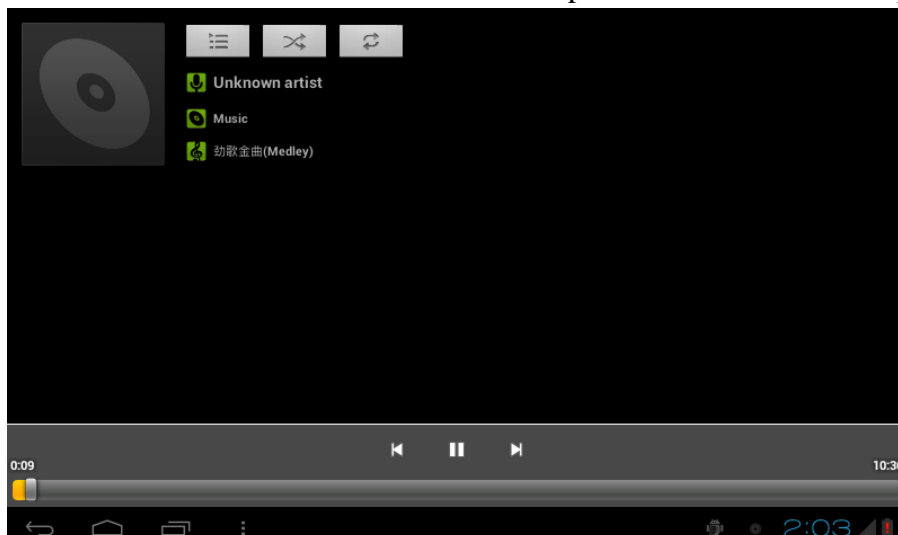


3.1.4 访问SD卡

SD 卡的目录为: /storage/sd_external/;
安装一个 ES Filebrowser 软件的打开这个目录即可。

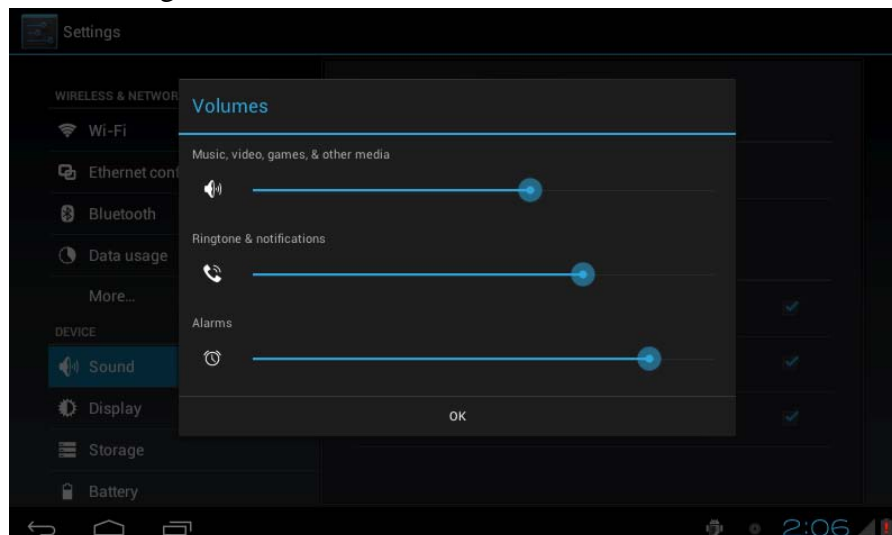
3.1.5 播放mp3

Android 系统可以自动识别 SD 卡中的 mp3 文件，下面是播放 mp3 的界面。



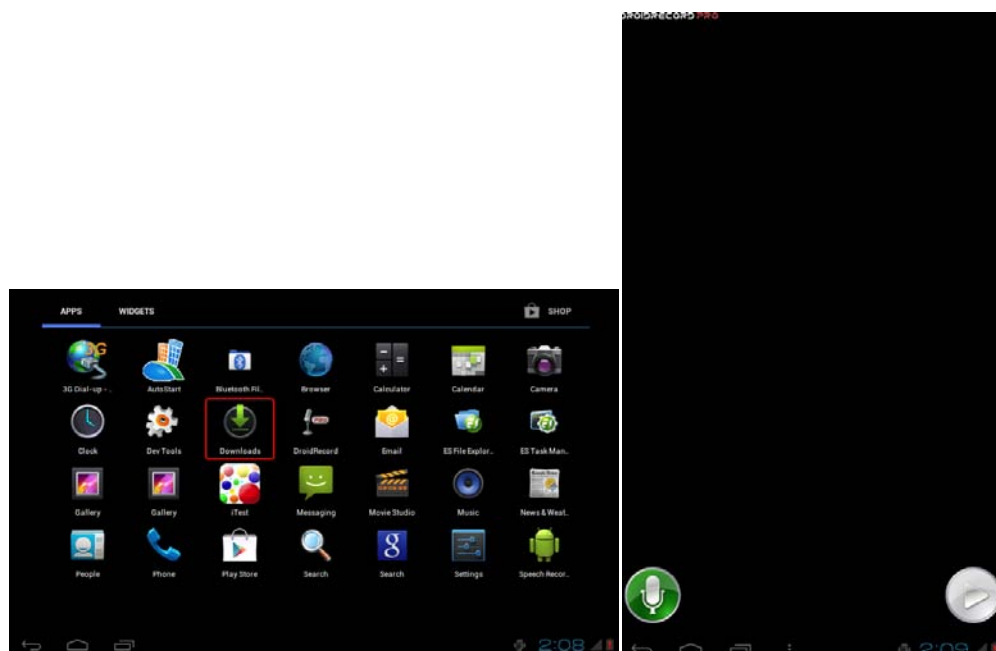
3.1.6 调节音量大小

进入 Setting -> Sound 可调节音量大小，如下图所示：



3.1.7 录音功能

我们在 Android 中内置了 DroidRecord 录音软件，可以用它进行录音与回放，程序图标如下面左边的图片所示，点击它即可启动录音程序，录音程序启动后，界面如右图所示：



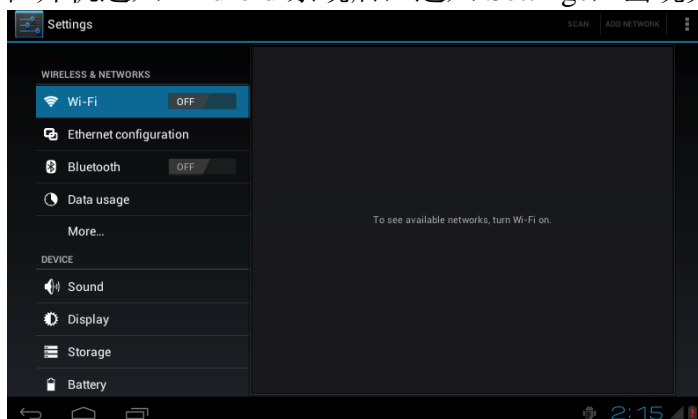
请参考下图的步骤启动录音,以及回收录音:



3.1.8 使用WiFi无线上网

本开发板可外接 USB WiFi 模块进行 WiFi 无线上网,在 Android 中连接 WiFi 的方法如下:

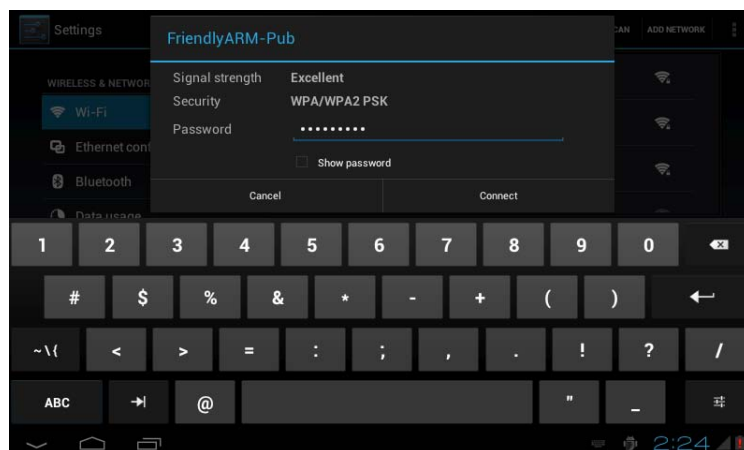
在开机进入 Android 系统后,进入 Settings,出现如图界面菜单



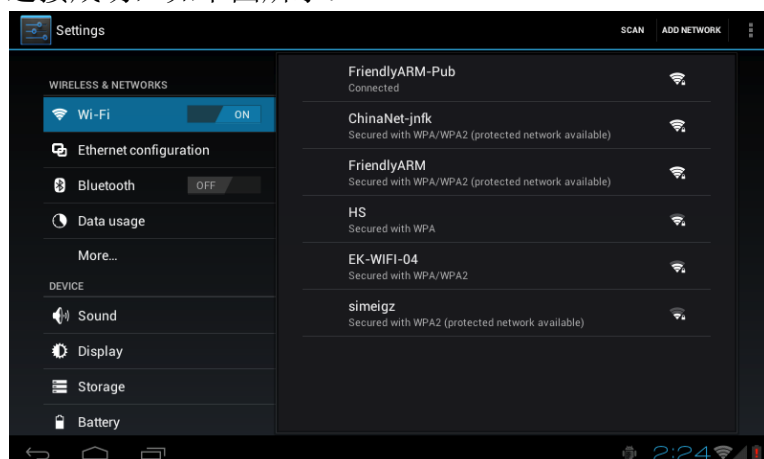
点 Wi-Fi,再点一下右边的 ON/OFF 开关开启 WiFi,开启后会自动搜索附近的无线网接入点,如图:



点你所要连接的无线网接入点，输入密码，如图



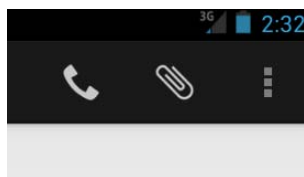
连接成功，如下图所示：



3.1.9 3G拨号上网及短信息收发

Tiny4412 目前支持中兴 MF210 模块（WCDMA）的 3G 拨号上网，使用官方提供的 RIL 驱动，因此功能稳定，可放心在项目当中使用，国内支持中国联通的 3G 上网和短信收发。

在 Android4 上无需进行任何设置，只要插上 MF210 模块开机，3G 就会自动连接上，右上角会出现 3G 图标，如下图所示：



这时打开自带的 News& Weather 应用，可以上网看新闻啦：



打开信息应用，短信息功能也是好的：



需要注意的是，目前市场上的 MF210 可能有好几个版本，目前我们测试过的设备 ID 主要有以下两款，其它的中兴 WCDMA 模块可能也支持，你可以自行测试一下：

VID: 19d2 PID: 0117

VID: 19d2 PID: 2003



另外，由于中兴 MF210 模块的接口是 miniPCIe 接口，而 Tiny4412 底板并没有 miniPCIe 接口引出，因此需要通过友善之臂设计的转换器来连接开发板的 USB 接口使用，转接板如下图所示：



内部电路板和相关连接口如下图所示，如你所见，有 SIM 卡插口，miniPCIe 接口和天线：



3.1.10 HDMI同步输出高清图像到电视

用 HDMI 线将本开发板与电视进行连接，开发板上的画面即可同步输出到电视上，如果你播放视频，声音也会同步输出到电视上。

目前默认输出的分辨率是 720P，如果你需要更改变分辨率，例如改为 1080P，可修改 ANDROID 源代码中的 hardware/samsung_slsi/exynos4/libhwc/hwc.cpp，将：

```
#define HDMI_DEFAULT_PRESET_ID V4L2_DV_720P60
```

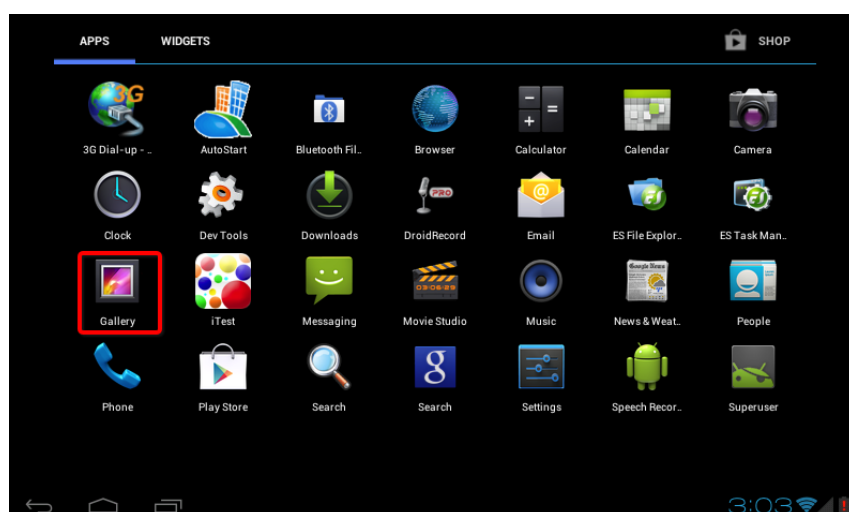
改成

```
#define HDMI_DEFAULT_PRESET_ID V4L2_DV_1080P60
```

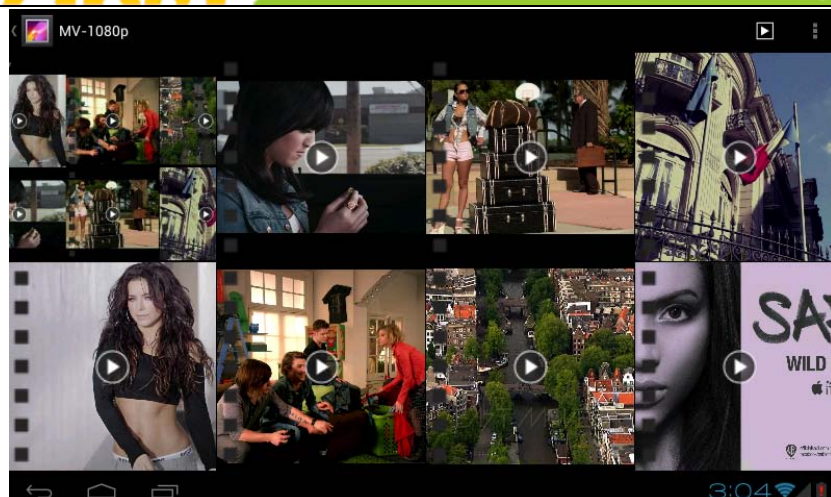
然后重新编译 libhwc 模块即可。

3.1.11 播放高清视频

将要播放的视频拷贝到 SD 卡（注：视频要求是 mp4 格式，音频压缩使用 aac 格式，我们提供了一些可用的测试视频存放在光盘的 Test Video 目录下），然后打开 Gallery 应用：



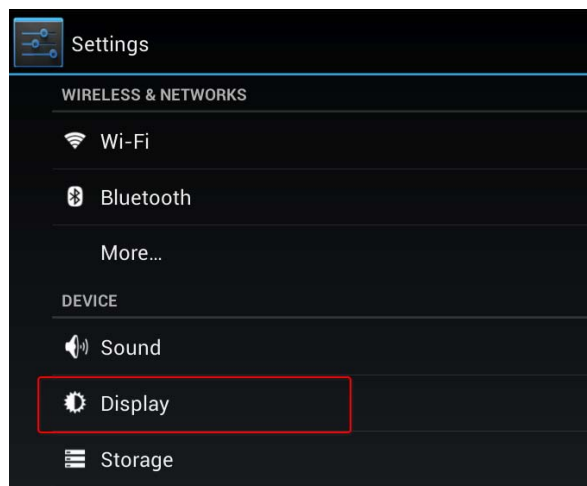
Gallery 会列出所探测到的视频，如下图所示：



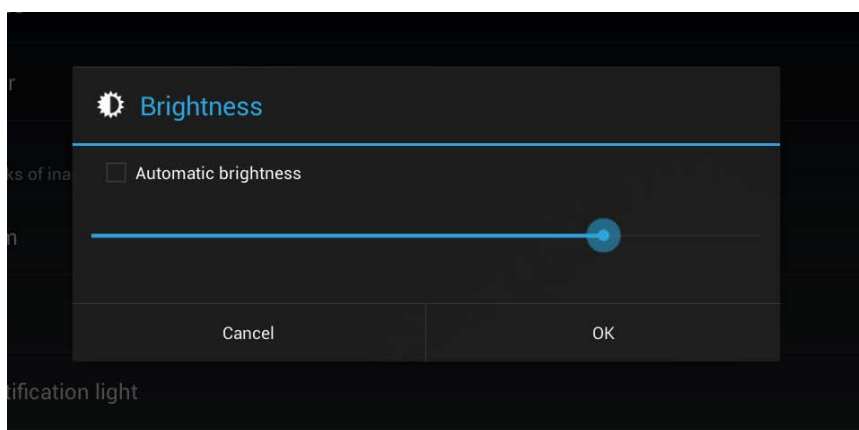
点击要播放的视频即可播放，如果连接了 HDMI 输出到电视，画面和声音将会同步在电视上显示, Exynos4412 最高支持 1080p 高清硬解播放。

3.1.12 背光调节设置

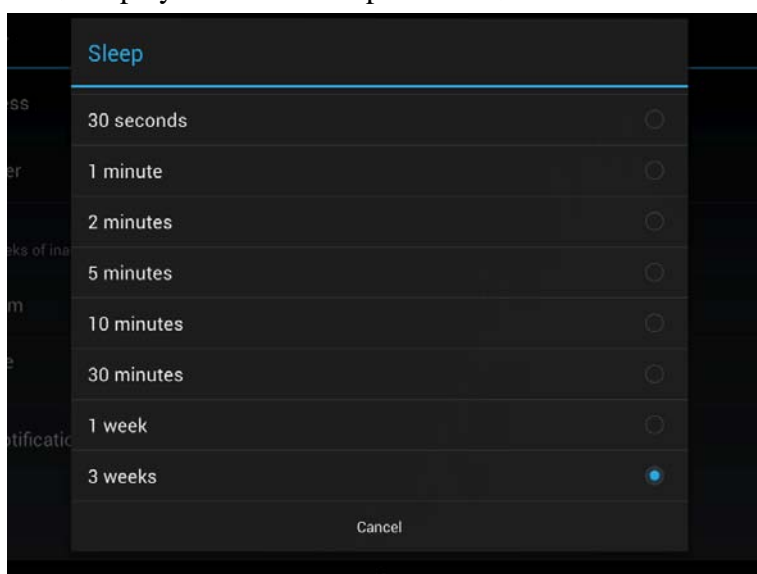
在以上使用 Android 系统的过程中,或许你已经注意到了,当在一段时间内没有点击屏幕时,背光会逐渐熄灭,在系统设置主界面中,点“Display”:



再点“Brightness”，打开背光设置窗口，在此你可以设置背光亮度，



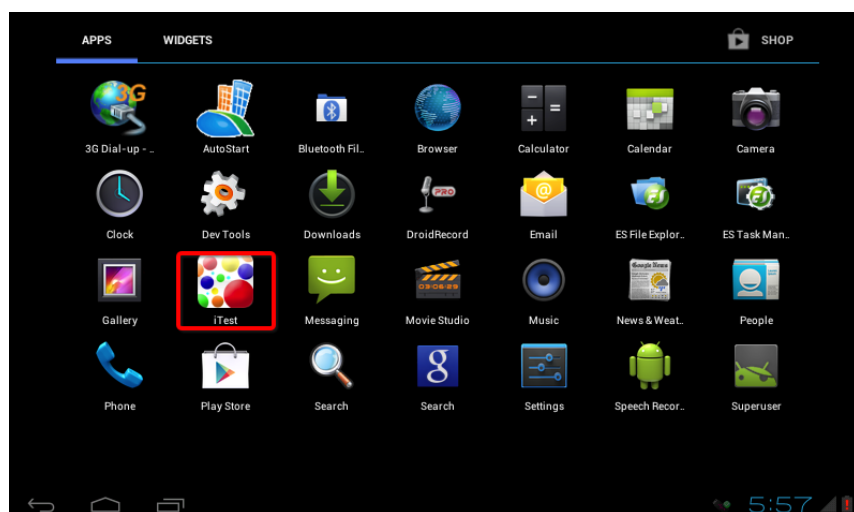
回到 Display 界面点“Sleep”，在此窗口，你可以设置背光延时关闭的时间



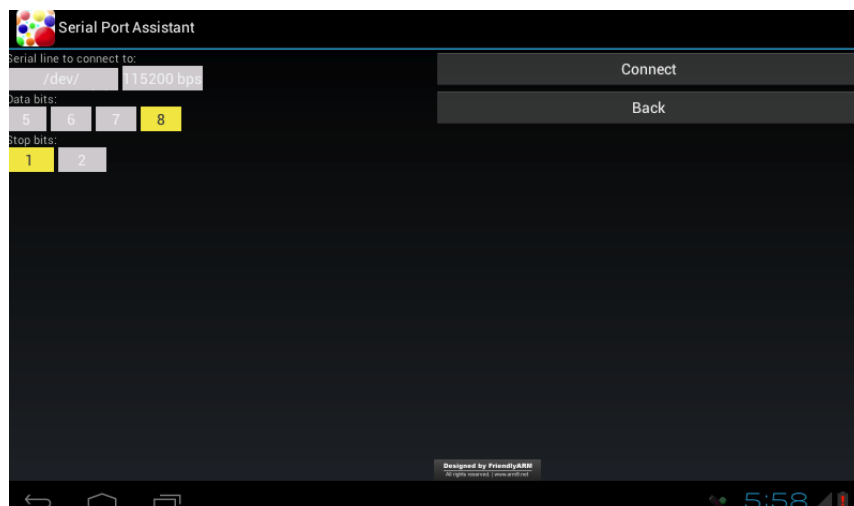
最长可将 Screen timeout 设置为 3 周（默认值）。

3.1.13 串口助手

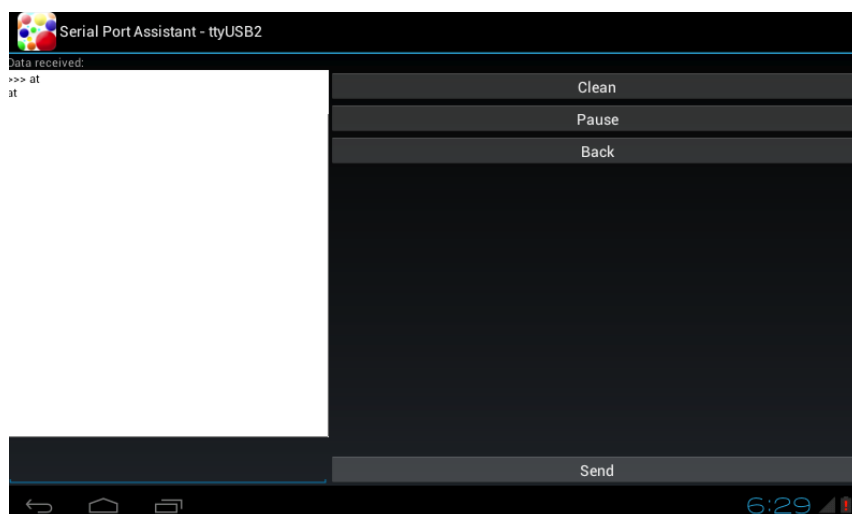
要使用串口助手功能，可在首页上点击 iTest 的图标启动 iTest 程序：



然后点击 Serial Port Assistant 启动串口助手，启动后，在左侧可设置串口的波特率等参数：



设置完成后，点击 Connect 按钮即可连接串口，连接成功后，可进行串口测试了，如下图所示：



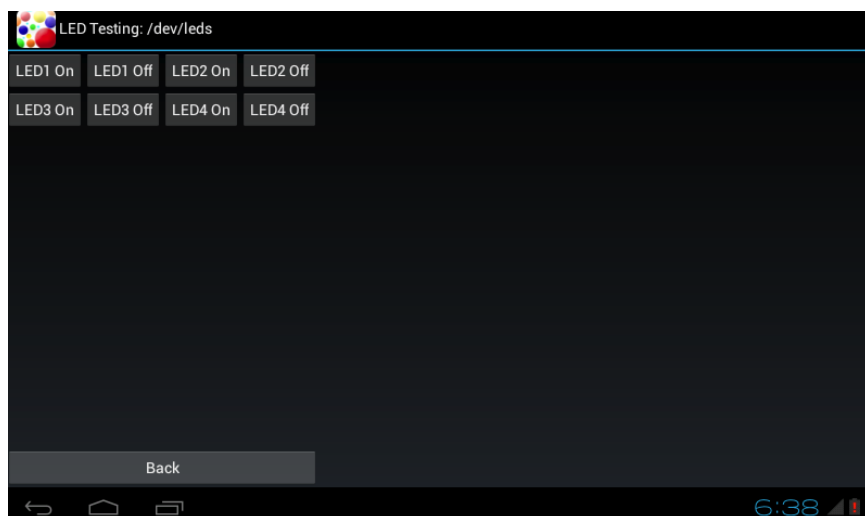
要发送数据到串口，可在 Send 左边的文本框进行输入，然后点发送即可。
点 Pause 是暂停消息的滚动，点 Clean 是清空接收到的消息。

注意事项：

- 1) 如果某个串口无法连接，在命令行用命令“fuser 文件名”测试一下设备是否被其他程序占用。
- 2) 如没有占用仍无法打开，则用“ls -l 文件名”查看一下权限，可用命令“chmod 777 文件名”更改一下设备文件的权限再试。
- 3) s3c2410_serial0 默认与 COM0 调试串口是相连的，因此，不要使用 s3c2410_serial0。

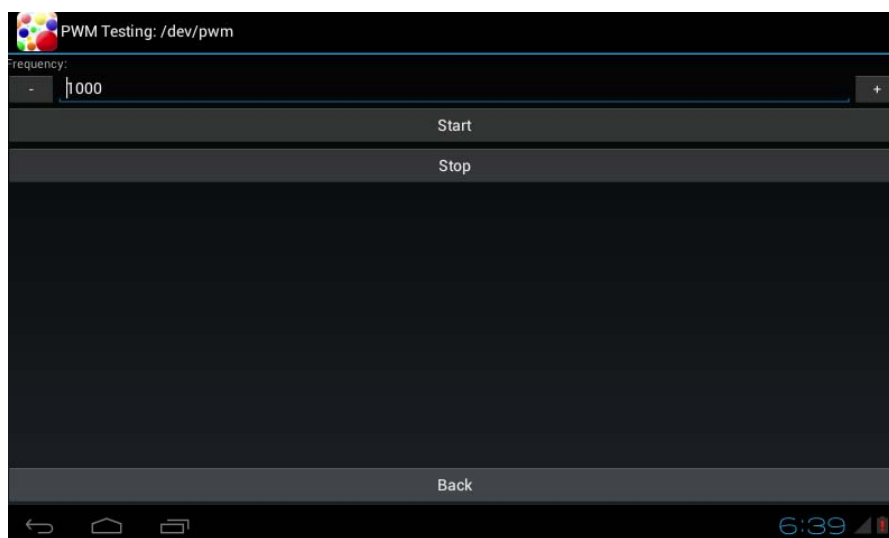
3.1.14 LED测试

要使用测试 LED，可在首页上点击 iTest 的图标启动 iTest 程序，
然后点击 LED Testing 将出现 LED 测试界面，如下图所示，直接点击按钮开关相应的 LED 即可：



3.1.15 PWM蜂鸣器测试

要使用测试 PWM，可在首页上点击 iTest 的图标启动 iTest 程序，然后点击 PWM Testing 将出现 PWM 测试界面，如下图所示：



在界面上，你可以手动输入频率，然后点 Start 令蜂鸣器发声，也可以通过+和-按钮调节频率。

点击 Stop 将停止发声。

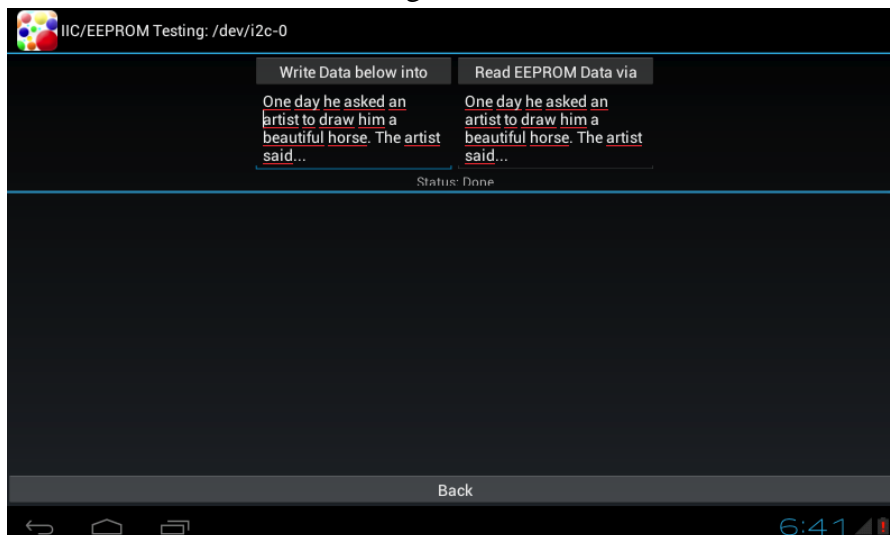
3.1.16 ADC测试

要进行 ADC 测试，即查看 A/D 转换的结果，可在首页上点击 iTest 的图标启动 iTest 程序，然后点击 A/D Convert 将出现 ADC 的转换结果显示界面，如下图所示：



3.1.17 I2C-EEPROM测试

要进行 I2C-EEPROM 的读写测试，可在首页上点击 iTest 的图标启动 iTest 程序，然后点击 IIC/EEPROM Testing 将出现 EEPROM 的测试界面，如下图所示：



先点击左侧的“Write Data below into EEPROM”按钮将左侧文本框中的文字写入到 EEPROM 中，然后再点击右侧的“Read EEPROM Data via IIC”的按钮可将 EEPROM 中的文字



进行读出，并存放在右边的文本框中。

你可以在文本框中更改你想要写入到 EEPROM 的文字。

3.2 建立Android编译环境

此处的 Android 编译环境，是指编译创建 Android 内核及基本系统所需要的开发，搭建编译环境主要有以下几步：

Step1: 安装 Ubuntu12.04.2 64bit，注意，是 Ubuntu12.04.2，不是 Ubuntu12.04，后面还有一个小版本 2。

Step2: 在 Ubuntu12.04.2 上安装 Android 编译环境

Step3: 从光盘中安装友善之臂提供的 Android 4.2.1 源代码、内核源代码、交叉编译器等，用于编译内核和 Android4.2.1；

Step4: 安装 fastboot 等相关的 Android 工具；

3.2.1 安装Ubuntu12.04.2 64bit系统

请在网络上自行下载得到 Ubuntu12.04.2 64bit 的 DVD 光盘映像，下载地址：<http://releases.ubuntu.com/precise/>，文件名为：**ubuntu-12.04.2-desktop-amd64.iso**，然后刻成光盘。

小贴士：由于 Ubuntu 安装时会联网检查更新，需要访问境外服务器，速度比较慢，所以为了加快安装速度，可先拔掉网线来安装系统，安装完再插上网线。

- **Step1:**将 Ubuntu12.04.2 光盘插入光驱，开机进入 BIOS，设置为光盘启动，顺利的话将进入 Ubuntu12.04.2 安装界面的语言选择界面，如下图所示，可按个人喜好选择语言，这里我选择“简体中文”：

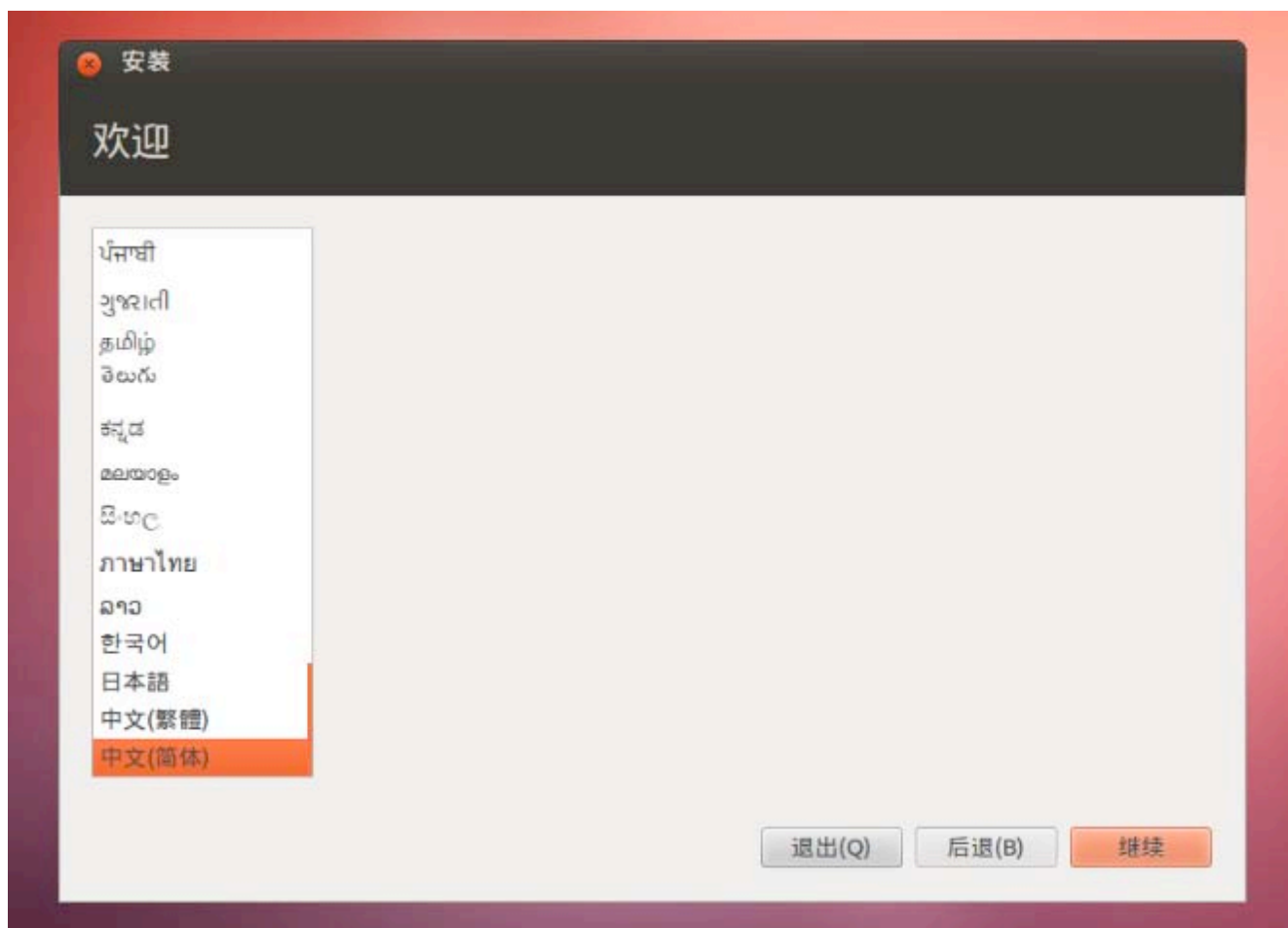
Language			
Amharic	Gaeilge	Malayalam	Thai
Arabic	Galego	Marathi	Tagalog
Asturianu	Gujarati	Nepali	Türkçe
Беларуская	ᲀᲠᲚᲰᲣ	Nederlands	Uyghur
Български	Hindi	Norsk bokmål	Українська
Bengali	Hrvatski	Norsk nynorsk	Tiếng Việt
Bosanski	Magyar	Punjabi (Gurmukhi)	中文(简体)
Català	Bahasa Indonesia	Polski	中文(繁體)
Čeština	Íslenska	Português do Brasil	
Dansk	Italiano	Português	
Deutsch	日本語	Română	
Dzongkha	தமிழ்	Русский	
Ελληνικά	Қазақ	Sámegiellii	
English	Khmer	සිංහල	
Esperanto	ಕನ್ನಡ	Slovenčina	
Español	한국어	Slovenščina	
Eesti	Kurdî	Shqip	
Euskara	Lao	Српски	
فارسی	Lietuviškai	Svenska	
Suomi	Latviski	Tamil	
Français	Македонски	தமிழ்	

F1 Help F2 Language F3 Keypad F4 Modes F5 Accessibility F6 Other Options

- **Step2:**在出现的如下界面中, 选“安装 Ubuntu(I)”:



- **Step3:**在出现的如下界面中，选“继续”：



- **Step4:**接下来是设置是否安装第三方软件和是否连网更新，什么都不选，点继续：



- **Step5:**接下来是分区设置，在虚拟机安装，建议选择第一项，否则，请选择其他选项进行分区，分区后，点“现在安装”：



- **Step6:**按下来就是安装了，中途可以设置下时区、用户名等，按向导操作即可：



- **Step7:**安装完，点重启即可：



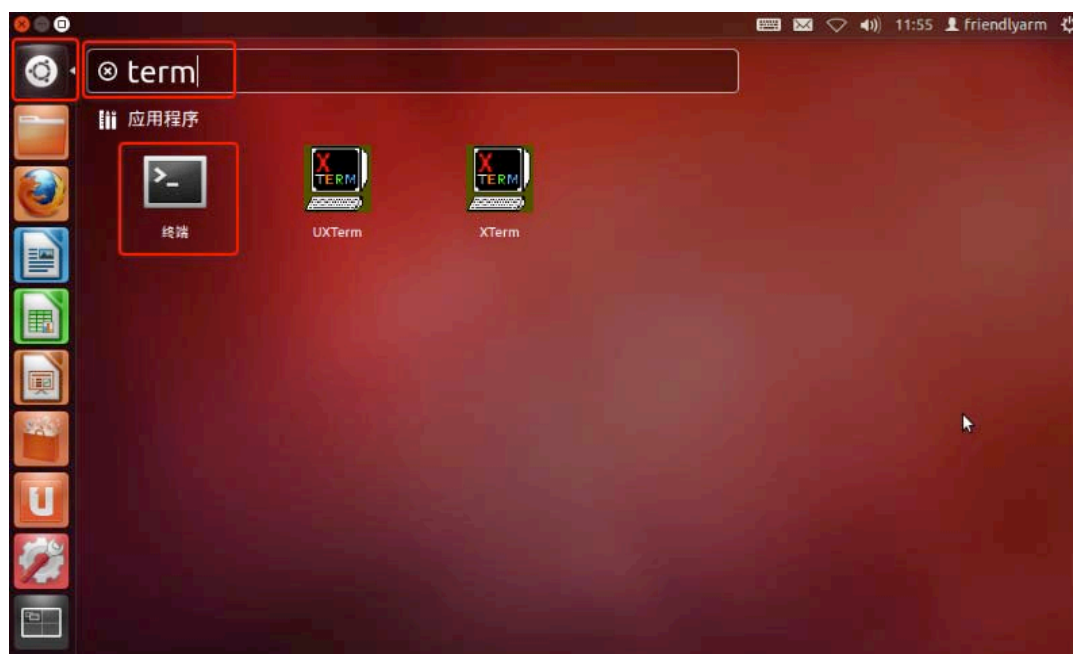
重启后，现在可以插入网线了，进入你设置的用户名和密码登入系统即可。

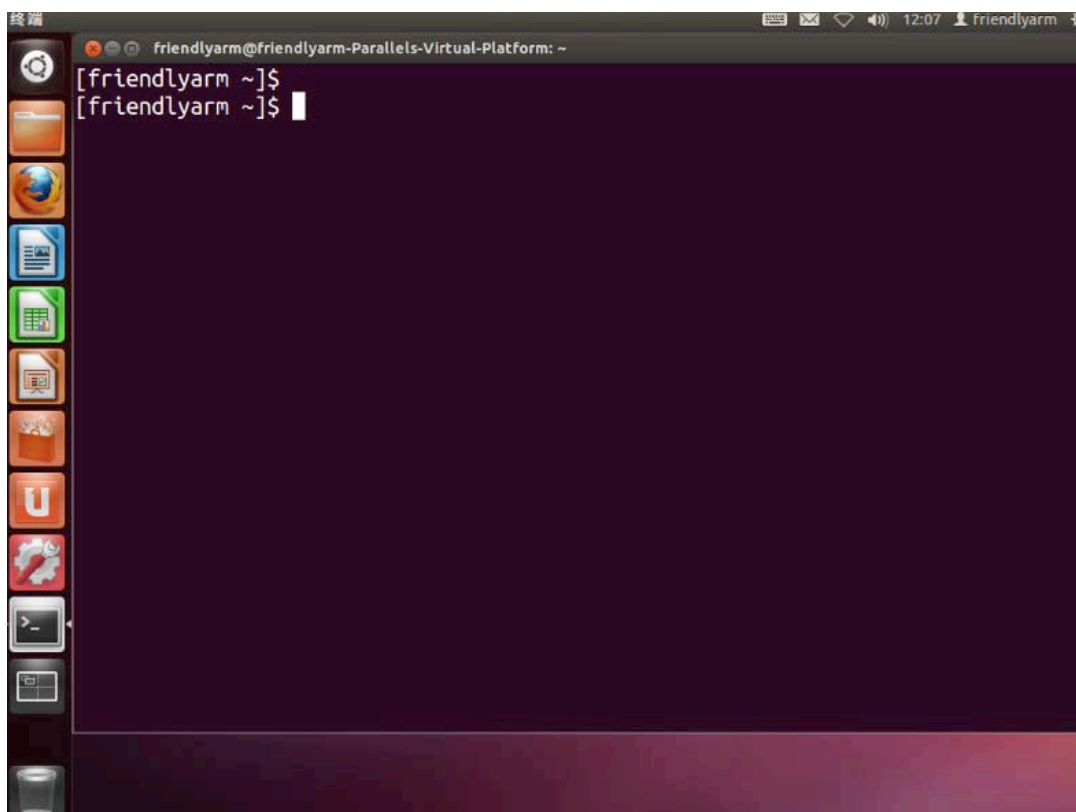
3.2.2 设置Ubuntu系统

3.2.3 以root用户登录

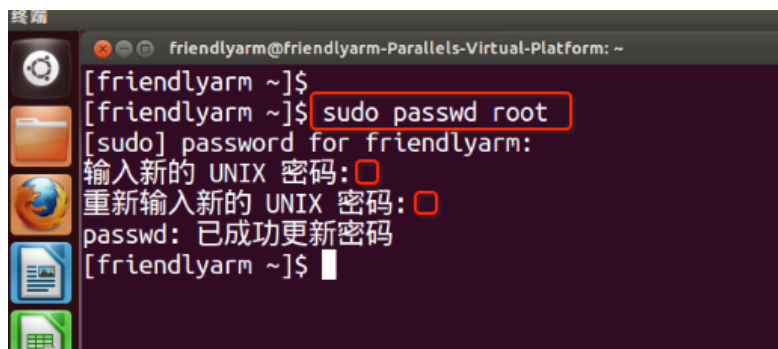
由于做嵌入式开发经常需要使用一些系统级工具，例如 minicom 和 ifconfig，因此，为了开发方便，推荐使用 root 用户来进行日常开发，Ubuntu 默认是不能通过 root 用户进行登录的，但按以下步骤设置后就可以用 root 用户登录了。

以下步骤需要在命令行操作，因为先参考以下方法进入终端命令行模式，点左侧任务栏上的第一个按钮，输入 term，终端工具将出现在搜索结果中，点击“终端”启动它：





首先，需要先设定一个 root 的密码，在终端上输入 `sudo passwd root`，按提示输入 root 用户的密码两次即可，如下图所示：



接着，输入 `su root`，以 root 用户登录，由于命令提示符太长，这里输入 `export PS1='[\u@\h \W]$'` 重新设置了一下提示符，然后输入 `cp -p /etc/lightdm/lightdm.conf /etc/lightdm/lightdm.conf.bak` 备份一下 lightgdm 配置：

```
[friendlyarm ~]$ su root
密码: 
root@friendlyarm-Parallels-Virtual-Platform:/home/friendlyarm# export PS1='[\u \W]$'
[root friendlyarm]$ cp -p /etc/lightdm/lightdm.conf /etc/lightdm/lightdm.conf.bak
[root friendlyarm]$
```

接下来用 vi /etc/lightdm/lightdm.conf 命令用 vi 编辑器打开/etc/lightdm/lightdm.conf:

```
[root friendlyarm]$ 
[root friendlyarm]$ vi /etc/lightdm/lightdm.conf
```

在文件末尾加入一行:

greeter-show-manual-login=true

最后内容如下图所示:

```
root@friendlyarm-Parallels-Virtual-Platform: /home/friend
[SeatDefaults]
greeter-session=unity-greeter
user-session=ubuntu
greeter-show-manual-login=true
```

保存重启, 重启后, 在登录界面上, 点“登录”:



然后输入 root 回车, 再输入密码即可以 root 用户登录系统了。

3.2.3.1 安装Android源代码编译所需的软件包

Ubuntu 默认安装后是不带软件开发相关的组件的，因此需要先安装一系列的软件开发组件，请先保存你的电脑能正常访问互联网。

第一步，先从 Tiny4412 光盘中拷贝 tools 目录下的 ubuntu 目录到 tmp 目录下备用，如果你是从网上下载的 iso 文件，可以通过以下命令加载 iso，然后执行拷贝：

```
# mkdir -p /mnt/iso  
# mount -o loop Tiny4412-20130707.iso /mnt/iso  
# cp /mnt/iso/tools/ubuntu /tmp/ -a
```

(注：#号是提示符，无需输入)

第二步，安装 jdk6:

```
# cd /tmp/ubuntu/jdk6/  
# chmod 755 install-sun-java6.sh  
# ./install-sun-java6.sh
```

最后一步，通过执行 install-devel-packages.sh 脚本，联网安装软件开发相关的软件包：

```
# cd /tmp/ubuntu/  
# chmod 755 install-devel-packages.sh  
# ./install-devel-packages.sh
```

中途会询问是否下载软件包，输入 Y 回车即可。

3.2.4 安装交叉编译器

我们使用的是 arm-linux-gcc-4.5.1（和 Mini210 相同），它在编译内核时会自动采用 armv7 指令集，支持硬浮点运算，下面是安装它的详细步骤。

Step1: 将光盘 Android 目录中的 arm-linux-gcc-4.5.1-v6-vfp-YYYYMMDD.tgz 复制到/tmp 目录，然后进入到该目录，执行解压命令：

```
#cd /tmp  
#tar xvzf arm-linux-gcc-4.5.1-v6-vfp-YYYYMMDD.tgz -C /
```

注意：C 后面有个空格，并且 C 是大写的，它是英文单词“Change”的第一个字母，在此是改变目录的意思。

执行该命令，将把 arm-linux-gcc 安装到/opt/FriendlyARM/toolschain/4.5.1 目录。



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

Step2: 把编译器路径加入系统环境变量, 运行命令

#gedit ~/.bashrc

编辑 ~/.bashrc 文件, 注意 “bashrc” 前面有一个 “.”, 修改最后一行为 **export PATH=\$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin**, 注意路径一定要写对, 否则将不会有效。

如图, 保存退出。

```
root@tom:/opt/FriendlyARM/toolschain/4.5.1
File Edit View Terminal Tabs Help
# .bashrc
# User specific aliases and functions

alias rm='rm -i'
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

export PATH=$PATH:/opt/FriendlyARM/toolschain/4.5.1/bin
```

重新登录系统(不必重启机器, 开始->logout 即可), 使以上设置生效, 在命令行输入 arm-linux-gcc -v, 会出现如下信息, 这说明交叉编译环境已经成功安装。

```
root@tom:/opt/FriendlyARM/toolschain/4.5.1
File Edit View Terminal Tabs Help
[root@tom 4.5.1]# arm-linux-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gcc
COLLECT_LTO_WRAPPER=/opt/FriendlyARM/toolschain/4.5.1/libexec/gcc/arm-none-linux-gnueabi/4.5.1/lto-w
rapper
Target: arm-none-linux-gnueabi
Configured with: /work/toolchain/build/src/gcc-4.5.1/configure --build=i686-build_pc-linux-gnu --hos
t=i686-build_pc-linux-gnu --target=arm-none-linux-gnueabi --prefix=/opt/FriendlyARM/toolschain/4.5.1
--with-sysroot=/opt/FriendlyARM/toolschain/4.5.1/arm-none-linux-gnueabi/sys-root --enable-languages
=c,c++ --disable-multilib --with-cpu=arm1176jzf-s --with-tune=arm1176jzf-s --with-fpu=vfp --with-flo
at=softfp --with-pkgversion=ctng-1.8.1-FA --with-bugurl=http://www.arm9.net/ --disable-sjlj-exception
ns --enable-cxa_atexit --disable-libmudflap --with-host-libstdcxx=-static-libgcc -Wl,-Bstatic,-ls
tdc++, -Bdynamic -lm' --with-gmp=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-mpf
r=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-ppl=/work/toolchain/build/arm-non
e-linux-gnueabi/build/static --with-cloog=/work/toolchain/build/arm-none-linux-gnueabi/build/static
--with-mpc=/work/toolchain/build/arm-none-linux-gnueabi/build/static --with-libelf=/work/toolchain/b
uild/arm-none-linux-gnueabi/build/static --enable-threads=posix --with-local-prefix=/opt/FriendlyARM
/toolschain/4.5.1/arm-none-linux-gnueabi/sys-root --disable-nls --enable-symvers=gnu --enable-c99 --
enable-long-long
Thread model: posix
gcc version 4.5.1 (ctng-1.8.1-FA)
[root@tom 4.5.1]#
```

3.2.5 解压安装Andorid4.2.1 源代码

首先创建工作目录/opt/FriendlyARM/tiny4412/android
在命令行执行

```
#mkdir -p /opt/FriendlyARM/tiny4412/android
```

再从 Tiny4412 光盘中拷贝 Android 目录到 tmp 目录下备用，如果你是从网上下载的 iso 文件，可以通过以下命令加载 iso，然后执行拷贝：

```
# mkdir -p /mnt/iso  
# mount -o loop Tiny4412-20130707.iso /mnt/iso  
# cp /mnt/iso/Android /tmp/ -a
```

(注：#号是提示符，无需输入)

(1)解压安装 Android4 内核源代码

在工作目录/opt/FriendlyARM/tiny4412/android 中执行：

```
#cd /opt/FriendlyARM/tiny4412/android  
#tar xvzf /tmp/Android/linux-3.5-YYYYMMDD.tgz
```

将创建生成 linux-3.5 目录，里面包含了完整的内核源代码

说明：YYYYMMDD 是发行更新日期标志，请以光盘中实际日期尾缀为准。

(2)解压安装 Android4.2.1 源代码包

在工作目录/opt/FriendlyARM/tiny4412/android 中执行：

```
#cd /opt/FriendlyARM/tiny4412/android  
#tar xvzf /tmp/Android/android-4.2.1_r1-fs-YYYYMMDD.tar.gz
```

将创建 android-4.2.1_r1 目录。

说明：YYYYMMDD 是发行或更新日期标志，请以光盘中实际日期尾缀为准。

3.3 配置和编译Linux内核

编译 Linux3.5 内核：

```
#cd /opt/FriendlyARM/tiny4412/android/linux-3.5  
#cp tiny4412_android_defconfig .config ;注意 config 前面有个”.”
```

你可以执行 make menuconfig 对配置进行修改，修改完成后，输出 make 进行编译：

```
# make
```



最后会在 arch/arm/boot 目录下生成 zImage，将它替换掉 SD 卡 images/Android/下的 zImage 烧写到 Tiny4412 即可。

3.4 从源代码开始创建Android

为了使用户能更方便地编译 Android 源代码，我们特意准备好了现成的源代码包，并且制作了一些脚本用来编译源代码和创建烧写映像，这此脚本的作用如下：

脚本	作用	调用示例
setenv	设置 Android 编译相关的环境变量	. setenv ;注意 “.” 后面有一个空格
gen-img.sh	生成可用于 fastboot 和 SD 卡烧写的系统映像文件 system.img 和 ramdisk-u.bin	./gen-img.sh
burn-img.sh	执行 USB 烧写：当 Tiny4412 开发板正处于 fastboot 下载模式时，在连接 USB 线后，执行该脚本会将系统映像文件 system.img 和 ramdisk-u.bin 通过 USB 下载到 Tiny4412 开发板上。	./burn-img.sh

要编译 Android 源代码，只要在命令行执行以下命令即可：

编译 Android 4.2.1_r1 源代码：

```
#cd /opt/FriendlyARM/tiny4412/android/ android-4.2.1_r1
#. setenv ;注意 “.” 后面有一个空格
# make
```

(小贴士：在 make 后面加上-j 参数可利用 CPU 多核加速编译速度，例如 4 核的机器，可输入 make -j4 来编译)

编译 Android 需要等待很长的时间，建议不要使用虚拟机编译，使用多核的 CPU 加真实的 Linux 系统会快很多。

3.5 制作安装或运行文件系统映像

编译成功后，执行以下命令生成系统映像文件 system.img 和 ramdisk-u.img:

```
#!/gen-img.sh
```

执行 gen-img.sh 下会在 Android 源代码当前目录下生成 system.img 和 ramdisk-u.img:

```
[root android-4.2.1_r1]$ pwd
/opt/FriendlyARM/tiny4412/android-4.2.1_r1
[root android-4.2.1_r1]$
[root android-4.2.1_r1]$ ls
abi          cts          external    libcore     packages    setenv
bionic       dalvik       frameworks  libnativehelper  pdk         system
bootable     development  gdk         Makefile    prebuilts  system.img
build        device       gen-img.sh  ndk         ramdisk-u.img  tools
burn-img.sh  docs        hardware    out         sdk          vendor
[root android-4.2.1_r1]$
```

可将 system.img 和 ramdisk-u.img 拷到 SD 卡的 images/Android 目录下进行脱机烧写，也可以执行 burn-img.sh 进行 fastboot USB 烧写。

3.6 在Andorid程序中访问硬件

为方便用户开发需要访问开发板硬件资源的 Android 应用程序，友善之臂为用户开发了一个函数库(命名为 libfriendlyarm-hardware.so)，用于访问 Tiny4412 上的硬件资源，目前支持的硬件设备包括：串口设备、蜂鸣器设备、EEPROM、ADC 设备等。

iTest 应用程序是使用该函数库来进行开发的，你可以通过运行 Android 上的 iTest 应用程序来了解这个函数库的功能。

本章节主要介绍如何在 Android 应用程序中使用 libfriendlyarm-hardware.so 函数库。

3.6.1 如何使用函数库(libfriendlyarm-hardware.so)?

友善之臂移植的 Android 内置了 libfriendlyarm-hardware.so 库，该库文件位于 Android 源代码目录的以下路径:

```
vendor/friendly-arm/exynos4412/rootdir/system/lib/libfriendlyarm-hardware.so
```




在开发板上位于 /system/lib/libfriendlyarm-hardware.so 目录下。

如果你是参考本文档的方法用 Eclipse 开发 Android 应用程序，可以通过以下方法使用 libfriendlyarm-hardware.so:

- 1) 定位到你的 Android 应用程序目录，在应用程序目录下创建 libs 目录，再进入 libs 目录下创建 armeabi 目录，然后将 libfriendlyarm-hardware.so 库文件拷贝到 armeabi 目录下。
- 2) 再回到你的应用程序目录，进入 src 目录下分别创建 com\friendlyarm\AndroidSDK 三层目录，然后在 AndroidSDK 目录下用文件编辑器新增一个源代码文件并命名为 HardwareControler.java，在该文件中输入以下代码：

```
package com.friendlyarm.AndroidSDK;
import android.util.Log;

publicclass HardwareControler
{
    /* Serial Port */
    staticpublicnativeint openSerialPort( String devName, long baud, int dataBits, int stopBits );

    /* LED */
    staticpublicnativeint setLedState( int ledID, int ledState );

    /* PWM */
    staticpublicnativeint PWMPlay(int frequency);
    staticpublicnativeint PWMStop();

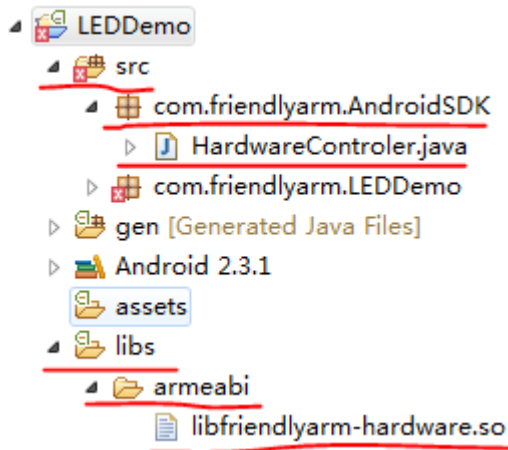
    /* ADC */
    staticpublicnativeint readADC();
    staticpublicnativeint readADCWithChannel(int channel);
    staticpublicnativeint[] readADCWithChannels(int[] channels);

    /* I2C */
    staticpublicnativeint openI2CDevice();
    staticpublicnativeint writeByteDataToI2C(int fd, int pos, byte byteData);
    staticpublicnativeint readByteDataFromI2C(int fd, int pos);

    /* IO */
    staticpublicnativeint write(int fd, byte[] data);
    staticpublicnativeint read(int fd, byte[] buf, int len);
    staticpublicnativeint select(int fd, int sec, int usec);
    staticpublicnativevoid close(int fd);
```

```
/* return 6410 or 210 */  
static public native int getBoardType();  
  
static {  
    try {  
        System.loadLibrary("friendlyarm-hardware");  
    } catch (UnsatisfiedLinkError e) {  
        Log.d("HardwareControler", "libfriendlyarm-hardware library not found!");  
    }  
}
```

部署完毕后，启动 Eclipse，在 Eclipse 左侧右击你的项目列表，选择“Refresh”刷新一下项目，这时，项目的相关的目录应该如下图红色标出的部分所示方为正确：



要使用 HardwareControler 的接口，首先需要在代码中加入如下代码导入 HardwareControler 类：

```
import com.friendlyarm.AndroidSDK.HardwareControler;
```

然后直接调用 HardwareControler 类的接口即可，下个章节将逐个说明 HardwareControler 类中的函数接口。

3.6.2 函数库(libfriendlyarm-hardware.so)接口说明

在应用层，可透过上一章节中的 HardwareControler 类来调用 libfriendlyarm-hardware.so 库中的接口，下面中列出 HardwareControler 类中的接口的定义，这些接口都是类方法，因此不需要



创建 HardwareControler 对象实例:

3.6.2.1 串口通讯的接口说明

与串口相关的接口如下表所示:

接口名称	参数与返回值说明	功能说明
<code>int openSerialPort(String devName, long baud, int dataBits, int stopBits)</code>	<p>参数说明:</p> <p>devName: 串口设备文件名, 可选的值有:</p> <p> /dev/s3c2410_seria 11 /dev/s3c2410_seria 12 /dev/s3c2410_seria 13 /dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3</p> <p>baud: 波特率</p> <p>dataBits: 数据位 (取值 5~8, 一般用 8)</p> <p>stopBits: 停止位 (取值 1~2, 一般用 1)</p> <p>返回值说明:</p> <p>成功打开串口时, 将返回串口的文件描述符, 用该描述符可进行 read、write 和 select 等操作, 如果打开失败, 则返回 -1。</p>	打开指定的串口设备, 并返回文件描述符。
<code>int write(int fd, byte[] data)</code>	<p>参数说明:</p> <p>fd: 要写入数据的文件描述符</p> <p>data: 要写入的数据</p> <p>返回值说明:</p> <p>成功返回写入的字节数, 出错返回 -1。</p>	向打开的设备或文件中写数据。
<code>int read(</code>	参数说明:	从打开的设备或文件中读取数



<pre>int fd, byte[] buf, int len)</pre>	<p>fd: 要读出数据的文件描述符 buf: 存储数据的缓冲区 len: 要读取的字节数</p> <p>返回值说明: 成功返回读取的字节数, 出错返回-1, 如果在调 read 之前已到达文件末尾, 则这次 read 返回 0。</p>	据。
<pre>int select(int fd, int sec, int usec)</pre>	<p>参数说明: fd: 要查询的文件描述符 sec: 阻塞等待数据多长时间 (单位: 秒) usec: 阻塞等待数据多长时间 (单位: 纳秒, 1 毫秒=1000 纳秒)</p> <p>返回值说明: 如果 fd 有数据可读, 返回 1, 如果没有数据可读, 返回 0, 出错时返回-1。</p>	查询打开的设备或文件是否有数据可读。
<pre>void close(int fd)</pre>	<p>参数说明: fd: 要关闭的文件描述符</p> <p>返回值说明: 无</p>	关闭指定的文件描述符

接口的使用说明:

先通过调用 openSerialPort 打开串口设备, 然后可以在线程中、或者用 timer 通过调用 select 接口轮询串口设备是否有数据到来, 如果有, 则调用 read 接口读取数据。

要往串口中写入数据, 调用 write 接口即可。

串口使用完毕后, 需要调用 close 关闭串口。

3.6.2.2 开关LED的接口说明

LED 操作的接口如下表所示:

接口名称	参数与返回值说明	功能说明
------	----------	------



int setLedState(int ledID, int ledState)	参数说明: ledID: 指定要开关哪一个 LED (取值 0~3) ledState: 1 表示亮, 0 表示灭 返回值说明: 成功返回 0, 失败返回-1	该接口用于开关 LED 灯。
--	--	----------------

3.6.2.3 让PWM蜂鸣器发声和停止发声的接口说明

蜂鸣器操作的接口如下表所示:

接口名称	参数与返回值说明	功能说明
int PWMPlay(int frequency);	参数说明: frequency: 要发声的频率 返回值说明: 成功返回 0, 失败返回-1	按指定的频率让蜂鸣器发声
int PWMStop();	参数说明: 无 返回值说明: 成功返回 0, 失败返回-1	让蜂鸣器停止发声

3.6.2.4 读取ADC的转换结果的接口说明

ADC 操作的接口如下表所示:

接口名称	参数与返回值说明	功能说明
int readADC()	参数说明: 无 返回值说明: 成功返回 ADC 转换的结果, 失败返回-1	读取 ADC 转换的结果
int readADCWithChannel(int channel)	参数说明: channel: 读取指定通道的 ADC 的值, 可选参数有 0, 1, 4, 5 返回值说明:	读取指定通道的 ADC 转换的结果



追求卓越 创造精品

TO BE BEST

TO DO GREAT

广州友善之臂计算机科技有限公司

	成功返回 ADC 转换的结果，失败返回-1	
<code>int[] readADCWithChannels(int[] channels);</code>	参数说明： channels: 要通道的 ADC 频道数组 返回值说明： 成功返回多个 ADC 结果(数组)，错误返回空	一次性读取多个频道的结果，性能好

3.6.2.5 EEPROM数据的写入与读取的接口说明

EEPROM 操作的接口如下表所示:

接口名称	参数与返回值说明	功能说明
<code>int openI2CDevice();</code>	参数说明： 无 返回值说明： 成功打开 IIC 设备时，将返回 IIC 设备的文件描述符，如果打开失败，则返回 -1。	打开 IIC 设备，并返回文件描述符。打开设备后，可使用 <code>writeByteDataToI2C</code> 和 <code>readByteDataFromI2C</code> 函数对 EEPROM 进行读写。
<code>int writeByteDataToI2C(int fd, int pos, byte byteData);</code>	参数说明： fd: 由 <code>openI2CDevice</code> 返回的文件描述符 pos: 指定数据在 EEPROM 的位置 (0~255) byteData: 要写入的数据 返回值说明： 成功返回写入的字节数，出错返回-1。	往 EEPROM 中写入数据 (每次只能写一个 byte)。 注意，该函数是个耗时的函数 (约 10 毫秒)，需要在工作线程中调用它。
<code>int readByteDataFromI2C(int fd, int pos);</code>	参数说明： fd: 由 <code>openI2CDevice</code> 返回的文件描述符 pos: 指定数据在 EEPROM 的位置 (0~255)	从打开的设备或文件中读取数据。 注意，该函数是个耗时的函数 (约 10 毫秒)，需要在工作线程中调用它。



	<p>返回值说明： 成功返回读取的数据（可强， 出错返回-1，如果在调 read 之前已到达文件末尾，则这次 read 返回 0。 返回值的类型是 int，你需要转换成 byte。</p>	
<code>void close(int fd)</code>	<p>参数说明： fd: 要关闭的文件描述符</p> <p>返回值说明： 无</p>	关闭指定的文件描述符

接口的使用说明：

先通过调用 `openI2CDevice` 打开 IIC 设备，然后需要创建一个新线程，在线程中调用 `writeByteDataToI2C` 将数据写入 EEPROM，或者调用 `readByteDataFromI2C` 从 EEPROM 读出数据，为什么要创建新线程呢？因为 `writeByteDataToI2C` 和 `readByteDataFromI2C` 函数在读写之后都会延时 10 毫秒左右，如果在 GUI 线程中调用会导致界面短暂阻塞。

EEPROM 可存储 256 个字节的数据，所以在读写时需要指定的位置范围是 0~255，每次只能读写一个字节。

EEPROM 操作完毕后，需要调用 `close` 关闭文件描述符。

3.6.3 示例程序说明

在光盘的“Android”目录下，带有一个示例程序 `LEDDemo`，可在 Windows 用 Eclipse 打开该工程来了解如何使用 `libfriendlyarm-hardware.so`。

用 miniUSB 线连接开发板后，在 Eclipse 可直接下载示例到开发板上运行并调试。