

# Mini6410 Android2.3 编程开发指南

2011-03-22

(本手册适用于 Mini6410/Tiny6410)



Copyright © 2007-2010 FriendlyARM

All rights reserved.



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

## 简介

本手册由广州友善之臂计算机科技有限公司(简称“友善之臂”)创建和维护,并作为标准用户手册的一个补充,仅供嵌入式爱好者学习参考使用,友善之臂目前并不对本手册的内容提供任何解释和解答服务,用户可以在论坛中反馈你所遇到的问题和疑问,我们将在以后的更新中修正或者采纳您的建议,本手册主要以首页日期为版本标志。

本手册由友善之臂软件开发工程师编写制作,以 Mini6410 和 Tiny6410 作为开发平台,讲解如何建立 Android 开发环境,包括 Android SDK 和 Eclipse 集成开发环境的下载及安装,以及如何使用 Mini6410 和 Tiny6410 开发板作为真机调试程序,非常适合 Android 初学者学习和参考。

Mini6410 是一款十分精致的低价高品质一体化 ARM11 开发板,由广州友善之臂设计、生产和发行销售。它采用三星 S3C6410 作为主处理器,在设计上承袭了 Mini2440 “精于心,简于形”的风格,而且布局更加合理,接口更加丰富,十分适用于开发 MID、汽车电子、工业控制、导航系统、媒体播放等终端设备;也可适用于 高校教学、嵌入式培训、个人研究学习和 DIY 等。

Tiny6410 是由友善之臂出品的一款以 ARM11 芯片(三星 S3C6410)作为主处理器的嵌入式核心板,它采用高密度 6 层板设计,尺寸为 64 x 50mm,它集成了 128M DDR RAM, 256M/1GB SLC Nand Flash 存储器,采用 5V 供电,在板实现 CPU 必需的各种核心电压转换,还带有专业复位芯片,通过 2.0mm 间距的排针,引出各种常见的接口资源,以供不打算自行设计 CPU 板的开发者进行快捷的二次开发使用。

因为 Mini6410 和 Tiny6410 的硬件接口和资源都是基本相同的,所以本手册完全适用于 Tiny6410 开发板。

为了方便用户,本手册的所用到的软件包放在光盘 A 的 Android 目录下,用户不需要自己去下载。

我们欢迎各位网友复制传播本手册,但不得擅自摘抄部分或全部内容用作商业用途,违者必究,友善之臂保留本手册的解释和修改权。

友善之臂公司网址: <http://www.arm9.net>

本手册由 ARM9 之家论坛(<http://www.arm9home.net>)发布,转载请注明出处,手册内难免有遗漏和不足之处,欢迎大家提出宝贵意见,请发邮件至: [qt\\_friendlyarm@163.com](mailto:qt_friendlyarm@163.com)。



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

#### 更新说明

2011-03-22	增加硬件访问接口
2011-02-14	更新 Android 版本为 2.3.1，开发平台使用 Windows 7
2011-01-27	本手册第一次发布



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

## 目 录

MINI6410 ANDROID2.3 编程开发指南 .....	- 1 -
目 录.....	- 4 -
第一章 MINI6410 和 TINY6410 开发板简介 .....	- 5 -
1.1 MINI6410 开发板 .....	- 5 -
1.2 TINY6410 开发板 .....	- 6 -
第二章 建立 ANDROID 应用开发环境 .....	- 8 -
2.1 步骤一：下载并安装 JDK (JAVA SE DEVELOPMENT KIT) .....	- 8 -
2.2 步骤二：下载并安装 ANDROID SDK .....	- 9 -
2.3 步骤三：下载并安装 ANDROID 2.3 的相关 PACKAGES .....	- 10 -
2.4 步骤四：安装 ECLIPSE 集成开发环境 .....	- 11 -
2.5 步骤五：安装 ANDROID DEVELOPMENT TOOLS 插件 .....	- 12 -
2.6 步骤六：配置 ECLIPSE .....	- 13 -
2.7 步骤七：创建 ANDROID 模拟器 .....	- 13 -
2.8 步骤八：开发第一个 ANDROID 程序 (验证开发环境是否搭建成功) .....	- 15 -
2.8.1 创建 HelloMini6410 工程 .....	- 15 -
2.8.2 在模拟器运行 HelloMini6410 程序 .....	- 18 -
2.9 步骤九：建立 MINI6410 调试环境 .....	- 18 -
2.9.1 安装 USB ADB 驱动程序 .....	- 18 -
2.9.2 在 Mini6410 上测试 ADB 功能 .....	- 21 -
2.9.3 通过 USB ADB 在 Mini6410 上运行程序 .....	- 23 -
2.9.4 在 Mini6410 上调试 Android 程序 .....	- 26 -
第三章 在 ANDROID 程序中访问硬件 .....	- 27 -
3.1 如何使用函数库(LIBFRIENDLYARM-HARDWARE.SO)? .....	- 27 -
3.2 函数库(LIBFRIENDLYARM-HARDWARE.SO)接口说明 .....	- 29 -
3.2.1 串口通讯的接口说明 .....	- 29 -
3.2.2 开关 LED 的接口说明 .....	- 31 -
3.2.3 让 PWM 蜂鸣器发声和停止发声的接口说明 .....	- 31 -
3.2.4 读取 ADC 的转换结果的接口说明 .....	- 31 -
3.2.5 EEPROM 数据的写入与读取的接口说明 .....	- 32 -
3.3 示例程序说明 .....	- 33 -

## 第一章 mini6410 和 tiny6410 开发板简介

### 1.1 Mini6410 开发板



Mini6410 是一款十分精致的低价高品质一体化 ARM11 开发板，由广州友善之臂设计、生产和发行销售。它采用三星 S3C6410 作为主处理器，在设计上承袭了 Mini2440 “精于心，简于形”的风格，而且布局更加合理，接口更加丰富，十分适用于开发 MID、汽车电子、工业控制、导航系统、媒体播放等终端设备；也可适用于高校教学、嵌入式培训、个人研究学习和 DIY 等。

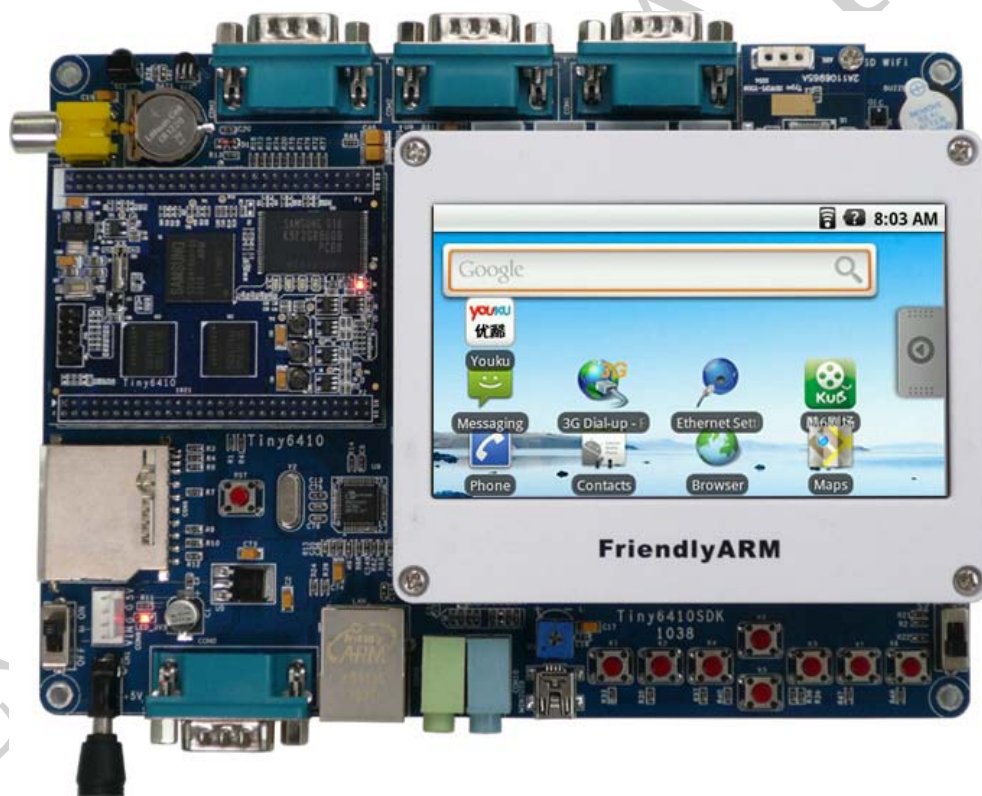
具体而言，Mini6410 具有双 LCD 接口、4 线电阻触摸屏接口、100M 标准网络接口、标准 DB9 五线串口、Mini USB 2.0-OTG 接口、USB Host 1.1、3.5mm 音频输出口、在板麦克风、标准 TV-OUT 接口、弹出式 SD 卡座、红外接收等常用接口；另外还引出 4 路 TTL 串口、CMOS Camera 接口、40pin 总线接口、30pin GPIO 接口(可复用为 SPI、I2C、中断等，另含 3 路 ADC、1 路 DAC)、SDIO2 接口(可接 SD WiFi)、10pin Jtag 接口等；在板的还有蜂鸣器、I2C-EEPROM、备份电池、AD 可调电阻、8 按键(可引出)、4LED

等；所有这些，都极大地方便了开发者的评估和使用，再加上我们按照 Mini6410 尺寸专门定制的 4.3"LCD 模块，真正做到让您“一手掌握所有”！

我们还充分地发挥了 6410 支持 SD 卡启动这一特性，使用我们精心研制的 superboot，无需连接电脑，只要把目标文件拷贝到 SD 卡中(可支持大于 2G 的高速大容量卡)，你就可以在开发板上极快极简单地自动安装或运行各种嵌入式系统(WindowsCE6/Linux/Android/Ubuntu 等)；甚至无需烧写，就可以在 SD 卡上直接运行它们，这一切，简直太酷了！

要了解 Mini6410 开发板的详细信息，请访问：<http://www.arm9.net/mini6410-feature.asp>

## 1.2 Tiny6410 开发板



Tiny6410 是一款以 ARM11 芯片(三星 S3C6410)作为主处理器的嵌入式核心板，该 CPU 基于 ARM1176JZF-S 核设计，内部集成了强大的多媒体处理单元，支持 Mpeg4, H.264/H.263 等格式的视频文件硬件编解码，可同时输出至 LCD 和 TV 显示；它还并带有 3D 图形硬件加速器，以实现 OpenGL ES 1.1 & 2.0 加速渲染，另外它还支持 2D 图形图像的平滑缩放，翻转等操作。

Tiny6410 采用高密度 6 层板设计，尺寸为 64 x 50mm，它集成了 128M DDR RAM，256M/1GB SLC Nand



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

Flash 存储器，采用 5V 供电，在板实现 CPU 必需的各种核心电压转换，还带有专业复位芯片，通过 2.0mm 间距的排针，引出各种常见的接口资源，以供 不打算自行设计 CPU 板的开发者进行快捷的二次开发使用。

Tiny6410SDK 是采用 Tiny6410 核心板的一款参考设计底板，它主要帮助开发者以此为参考进行核心板的功能验证以及扩展开发。该底板具有三 LCD 接口、4 线电阻触摸屏接口、100M 标准网络接口、标准 DB9 五线串口、Mini USB 2.0 接口、USB Host 1.1、3.5mm 音频输入输出口、标准 TV-OUT 接口、SD 卡座、红外接收等常用接口；另外还引出 4 路 TTL 串口，另 1 路 TV-OUT、SDIO2 接口（可接 SD WiFi）接口等；在板的还有蜂鸣器、I2C-EEPROM、备份电池、AD 可调电阻、8 个中断式按键等。

在布局上安排上，我们尽量考虑把常用尺寸的 LCD 模块能够固定在底板上，比如 3.5”，4.3”LCD，7” LCD 等，这样用户在使用时不至于把各种电线搅在一起，更增加了开发套件的便携性。

我们还充分地发挥了 6410 支持 SD 卡启动这一特性，使用我们精心研制的 Superboot，无需连接电脑，只要把目标文件拷贝到 SD 卡中（可支持高达 32G 的高速大容量卡），你就可以在开发板上极快极简单地自动安装各种嵌入式系统（WindowsCE6/Linux/Android/Ubuntu /uCOS2 等）；甚至无需烧写，就可以在 SD 卡上直接运行它们！

要了解 Tiny6410 开发板的详细信息，请访问：<http://www.arm9.net/tiny6410.asp>

## 第二章 建立 Android 应用开发环境

本章节将介绍如何在 Windows 7 系统中搭建 Android 开发环境, 本文将介绍 Android 2.3 下的 ADB 功能, 所以请确认你的 Android 版本不低于 Android 2.3, 如果你的光盘中的 Android 版本低于 2.3, 建议你先从友善之臂官方网站下载最新的光盘, 并更新 Android 为 Android 2.3。

在文中所用到的所有软件包都可以在光盘 A 中的 Android 目录上找到。

### 2.1 步骤一：下载并安装 JDK (Java SE Development Kit)

由于 Android SDK 和 Eclipse 都是用 Java 编写的, 因此需要先在 Windows 7 上安装 JDK, JDK 可按以下方法下载:

访问网站 <http://www.oracle.com/technetwork/java/javase/downloads/index.html>, 在页面中点击 JDK:



在打开的页面中, 选择 Platform 为 Windows, 并选中同意 License Agreement, 然后点击 Continue:



接着就是根据页面的提示下载 JDK 的安装程序，下载完成后，双击安装程序，根据向导的提示完成安装即可。

## 2.2 步骤二：下载并安装 Android SDK

首先，前往从网站 <http://developer.android.com/sdk/> 获取 Windows 平台的 最新的 Android SDK 套件，下载完成会得到一个安装文件，下载页面如下图所示，请下载红色方框框出的文件：

### Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview. If you're already using the Android SDK, you should update to the latest tools or platform using the starter package. See [Adding SDK Components](#).

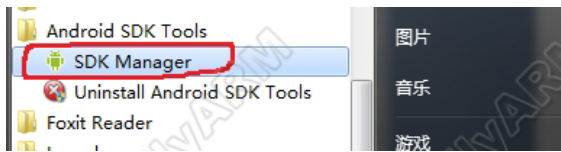
Platform	Package	Size	MD5 Checks
Windows	<a href="#">android-sdk_r09-windows.zip</a>	32779808 bytes	1a1bb8fad801
	<a href="#">installer_r09-windows.exe (Recommended)</a>	32828818 bytes	a0185701ac0
Mac OS X (intel)	<a href="#">android-sdk_r09-mac_x86.zip</a>	28829553 bytes	ef3102fdbbbb
Linux (i386)	<a href="#">android-sdk_r09-linux_x86.tgz</a>	26917824 bytes	9fefac5ff85d3

下载的好处是保证能获得最新版本，但不想或无法下载的用户，也可以使用 mini6410 光盘上的版本，位于光盘 A 的 Android 目录下，文件名为 **installer\_r09-windows.exe**。

双击下载得到的安装程序 **installer\_r09-windows.exe**，根据安装向导的提示安装即可，默认将安装在 C:\Program Files\Android\android-sdk-windows 目录下，安装完成后，SDK Manager 默认会自动启动。

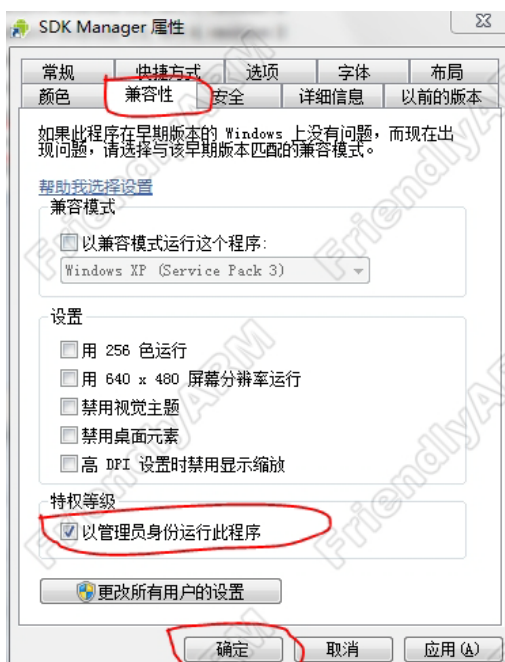
SDK Manager 启动时，由于我们还没有下载任何的 Packages，因此会弹出一个 Choose Packages to Install 的对话框，询问你是否安装所有可用的 Packages，由于我只想安装 Android 2.3 相关的 Packages，因此，这里我点击 Cancel 关闭对话框，回到 Android SDK and AVD Manager 的主界面。

以后可以通过在开始菜单中找到 **Android SDK Tools**，然后点击 **SDK Manager** 来启动 **SDK Manager**，如下图所示：



由于默认 SDK 是安装在 C 盘，所以最好将 SDK Manager 以管理员身份运行，否则在下载并安装 Packages 时会因为无法往 C 盘写入数据而导致失败，通过以下方法设置在每次启动 SDK 时，都以管理员身份启动：

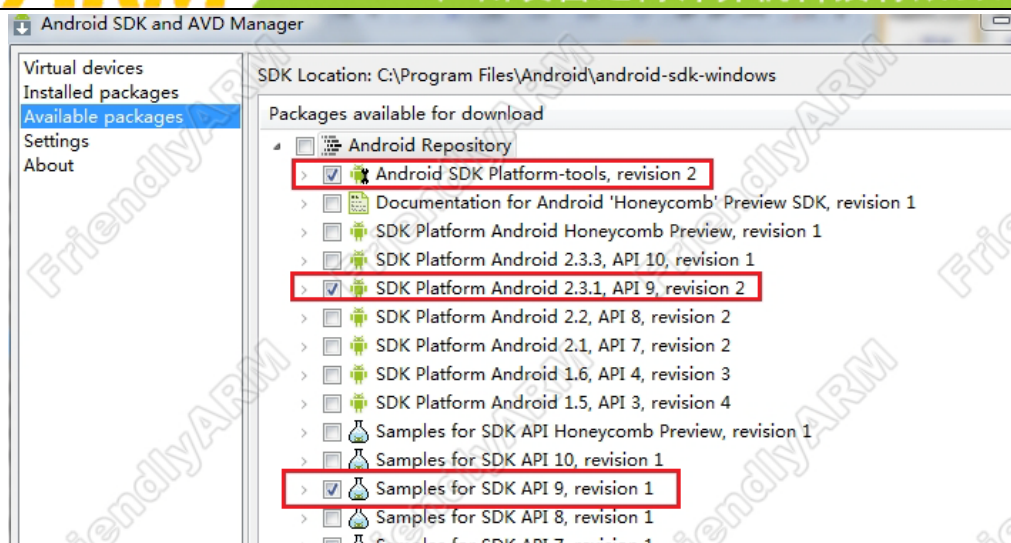
- 1) 在开始菜单中右击 SDK Manager 的快捷方式，点属性
- 2) 在 SDK Manager 快捷方式的属性对话框，点击“兼容性”的选项卡，选中“以管理员身份运行此程序”，如下图所示：



以后每次点击快捷方式运行 SDK 时，都会弹出权限确认对话框，点击“是”即可以管理员身份运行程序。

## 2.3 步骤三：下载并安装 Android 2.3 的相关 Packages

用管理员身份启动 SDK Manager，在 Android SDK and AVD Manager 的主界面上，选择“Available Packages”，点击 Android Repository 前面的“>”图标展开选项，参考下图选中 Android 2.3.1 相关的选项：



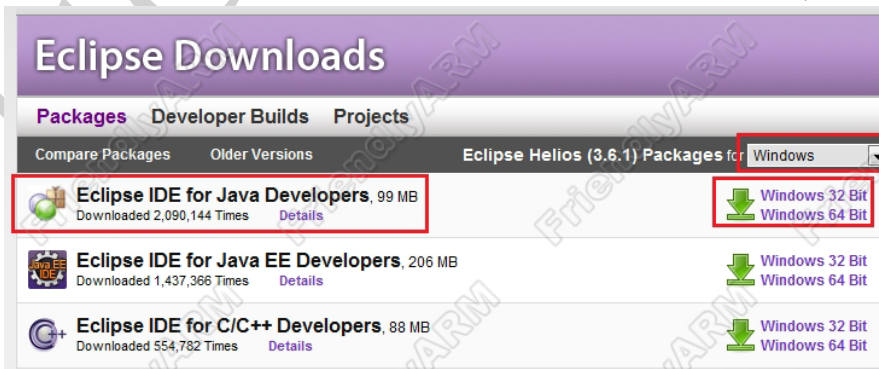
在上面的对话框中点击“Install Selected”按钮，在弹出的 Choose Packages to Install”对话框中，选中“Accept All”单选框，点击“Install”按钮，将进入下载过程，下载速度视你的网速而定，请耐心等待下载完成。

下载完成后，会弹出询问是否重启 ADB 的对话框，点击 Yes 即可。

## 2.4 步骤四：安装 Eclipse 集成开发环境

Android 推荐的 IDE 为 Eclipse，访问以下地址下载 Eclipse：  
<http://www.eclipse.org/downloads/>：

在下载页面中，参考下图，选择 Windows 类别，然后下载 Eclipse IDE for Java Developers 版本(下载 32bit 还是 64bit 版本则根据你的 Windows7 版本的实际情况而定)，如下图所示：

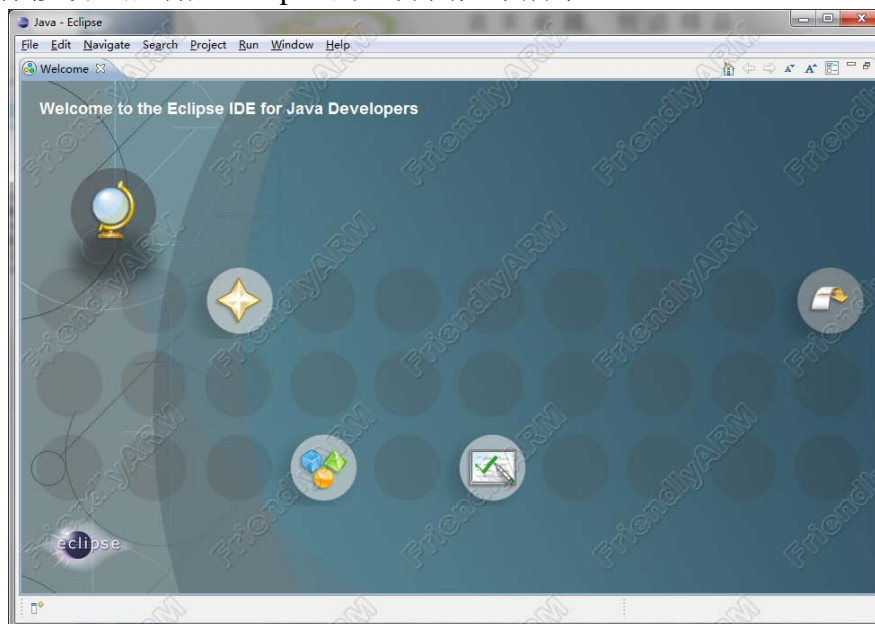


不想或无法下载的用户，也可以使用 mini6410 光盘上的版本，位于光盘 A 的 Android

目录下，文件名为 **eclipse-java-helios-SR1-win32.zip**。

下载后会得到一个名为 **eclipse-java-helios-SR1-win32.zip** 的压缩包(随着 eclipse 版本的更新，文件名可能有所不同)，Eclipse 是不需要安装的，直接解压就可以用了，推荐在 D:\根目录下解压这个压缩包，解压后进入 eclipse 目录，双击 **eclipse.exe** 即可启动 eclipse。

启动时，首先会让你设置 **Workspace** 的路径，这个路径用于保存你所创建所有程序的源代码，你可以自行修改，启动后 Eclipse 的主界面如下所示：



## 2.5 步骤五：安装 Android Development Tools 插件

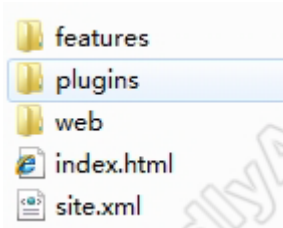
在上面的步骤中，我们分别安装了 Android SDK 和 Eclipse，但要在 Eclipse 中使用 Android SDK，还需要在 Eclipse 中安装一个 ADT 插件，安装方法如下：

访问网址 <http://developer.android.com/sdk/eclipse-adt.html#installing>，在网页中找到 ADT 的下载链接下载它，如下所示：

1. Download the current ADT Plugin zip file from the table below (do not unpack it).

Name	Package	Size	MD5 Checksum
ADT 9.0.0	<a href="#">ADT_9.0.0.zip</a>	4433536 bytes	bc2757f2a5a11d131390ce547bae154b

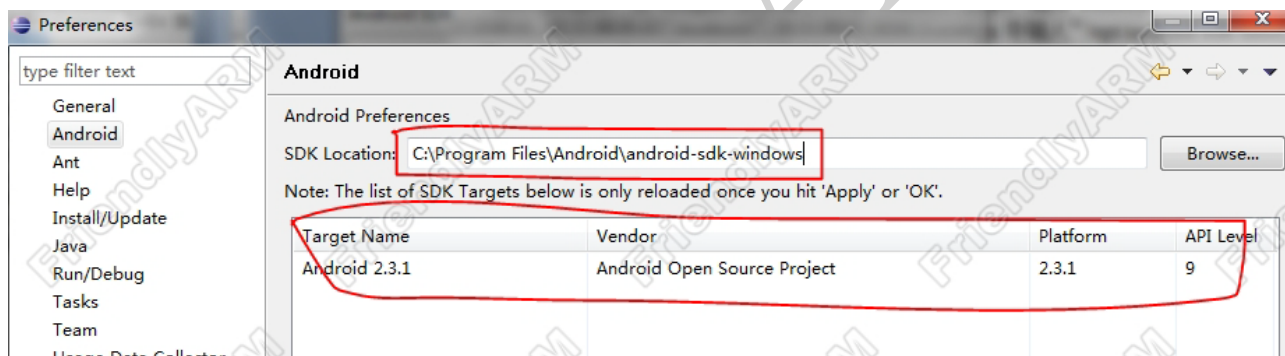
下载后得到文件 **ADT\_9.0.0.zip**，解压后得到如下文件：



先退出 Eclipse，然后将这些文件复制并覆盖 Eclipse 目录下即可完成安装。

## 2.6 步骤六：配置 Eclipse

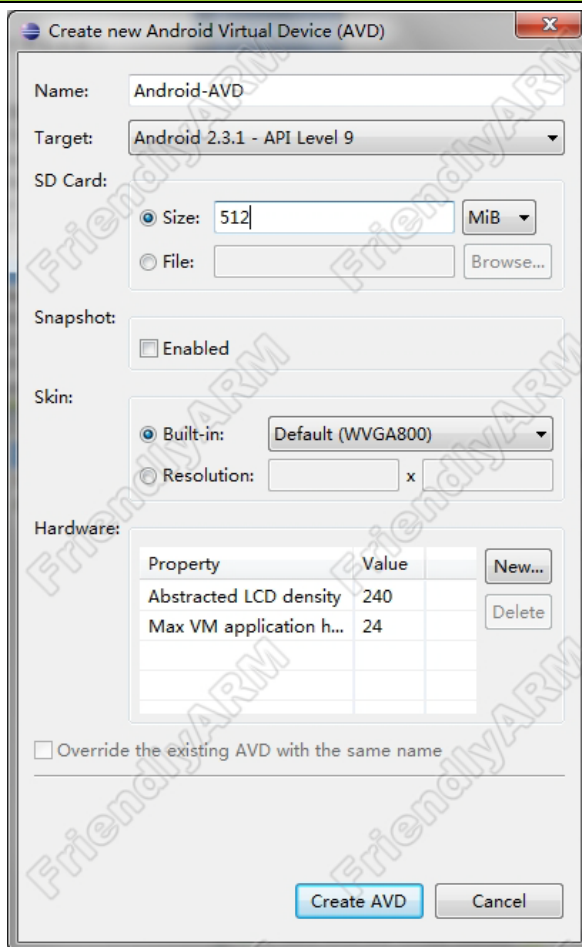
启动 Eclipse，在 Eclipse 主界面上点击 Window 菜单，选择 Preferences，在 Preferences 对话框中，在左侧选中“Android”，在右侧的 SDK Location 右边点击 Brower 按钮，选择 SDK 的安装路径，默认的路径为 C:\Program Files\Android\android-sdk-windows，如下图所示：



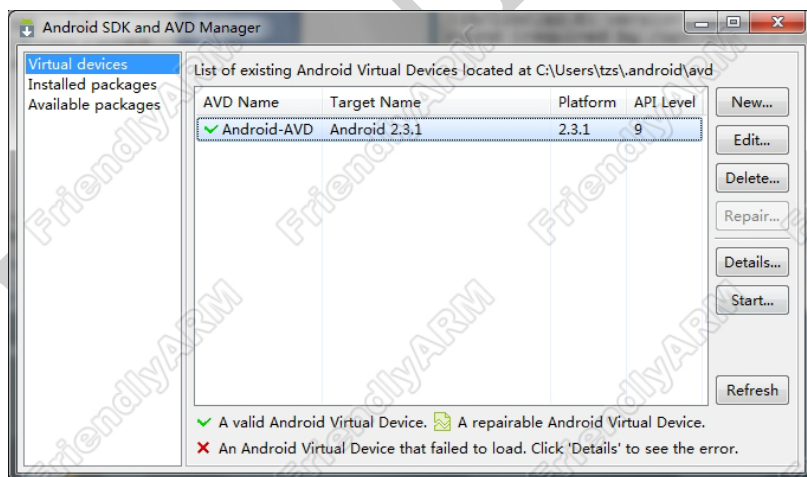
点击 OK 完成配置。

## 2.7 步骤七：创建 Android 模拟器

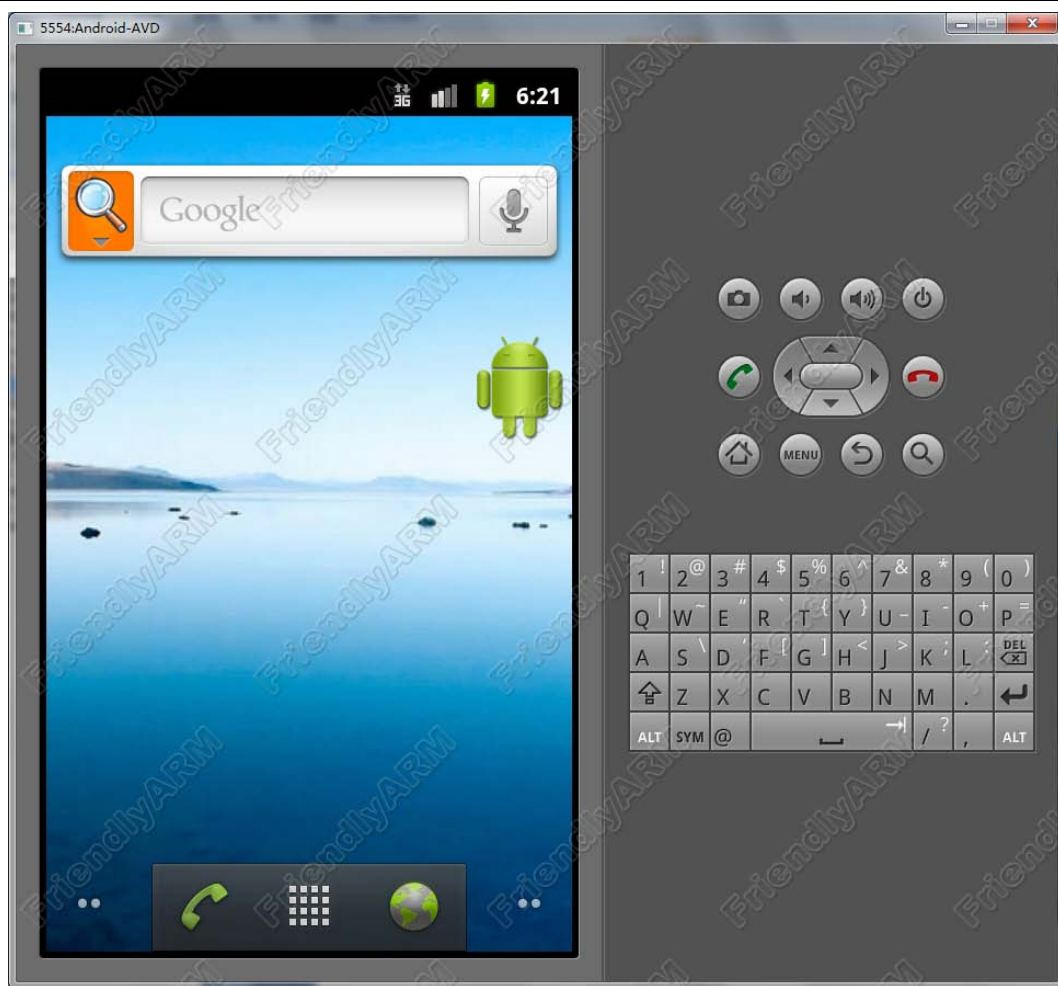
在开始菜单中找到 Android SDK Tools，然后点击 SDK Manager 来启动 SDK Manager，将弹出“Android SDK and AVD Manager”对话框，点击“New...”按钮，将弹出“Create new Android Virtual Device(AVD)”对话框，在 Name 中输入：Android-AVD，Target 中选择：Android 2.3.1 - API Level 9，SD Card 输入 512，其它选项默认，点击“Create AVD”按钮即可，如下图所示：



创建完成后，将在列表中列出该模拟器，如下图所示：



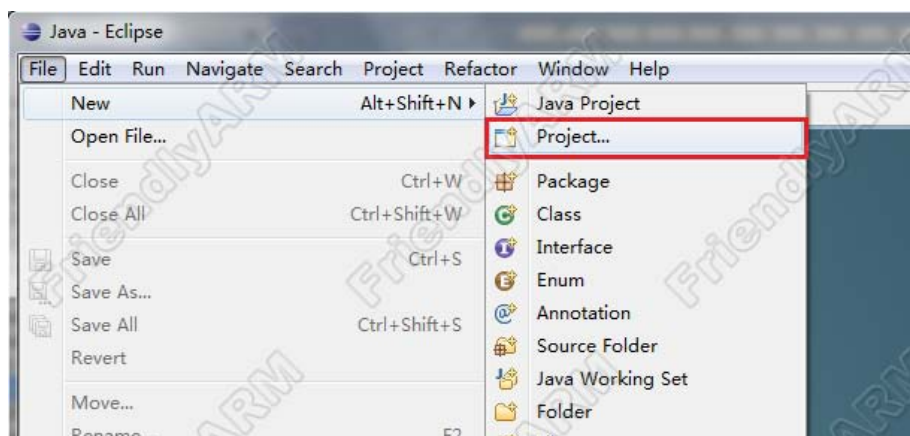
可在列表选中它，然后点击 **Start**，在弹出的 Launch Option 对话框中点击 **Launch** 启动模拟器，效果如下所示：



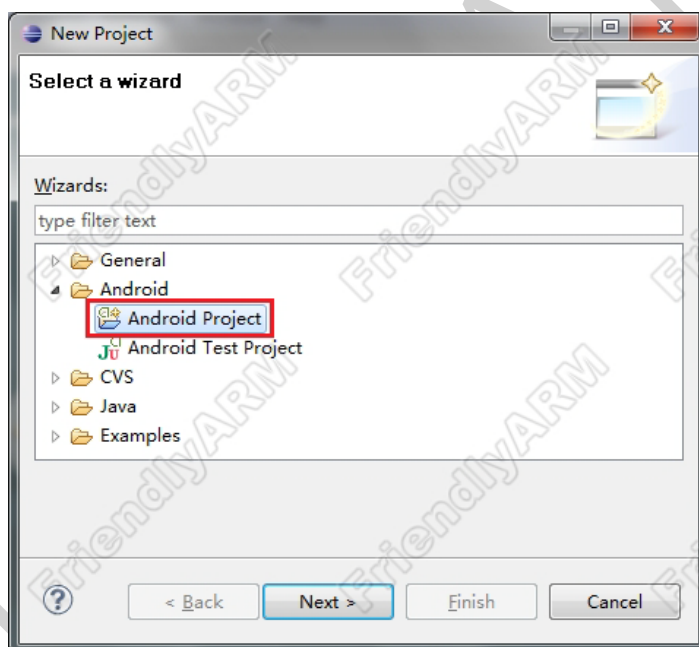
## 2.8 步骤八：开发第一个Android程序（验证开发环境是否搭建成功）

### 2.8.1 创建 HelloMini6410 工程

接下来我们新建一个 Android 项目 HelloMini6410，以验证开发环境是否搭建成功。在 Eclipse 主界面上依次选择菜单：File->New->Project...如下图所示：



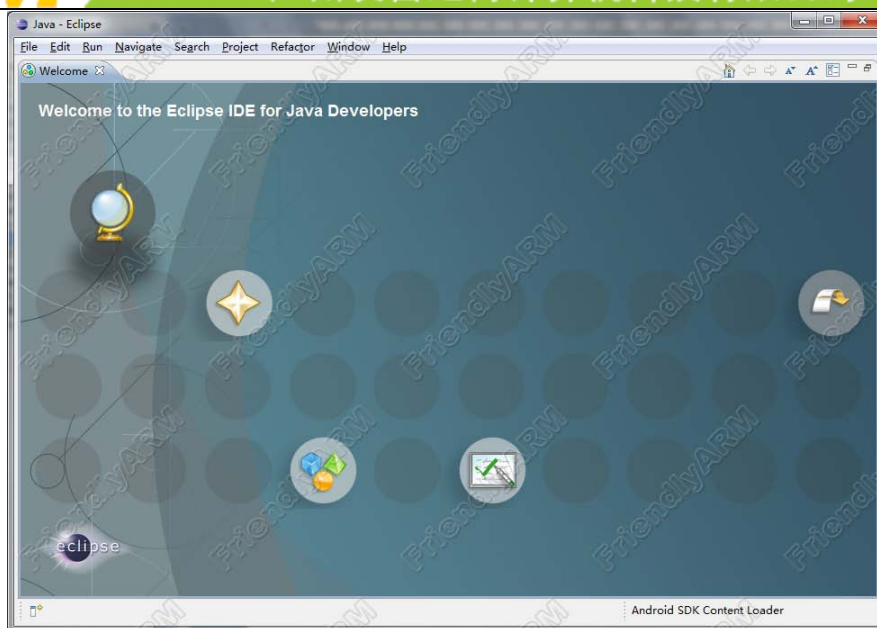
在弹出“New Project”对话框，选中：Android->Android Project，然后点击“Next”按钮，如下图所示：



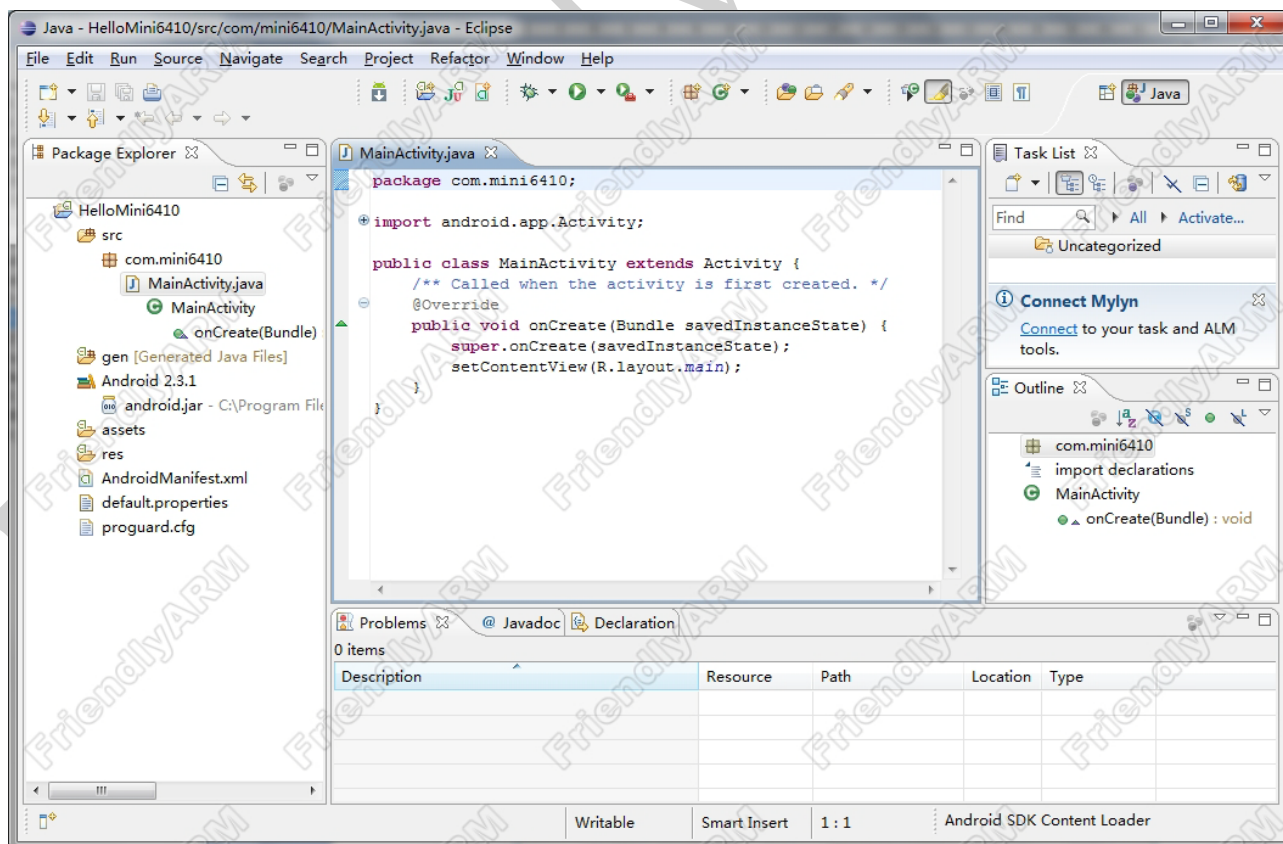
界面上将弹出“New Android Project”对话框，在对话框中进行如下输入：

- 1) Project Name 中输入：HelloMini6410
- 2) Build Target 中选择：Android 2.3.1 复选框
- 3) Appication name 中输入：HelloMini6410
- 4) Package name 中输入：com.mini6410
- 5) Create Activity 中输入：MainActivity

如下图所示，输入完成后，直接点击“Finish”按钮完成新建项目向导，将回到主界面：



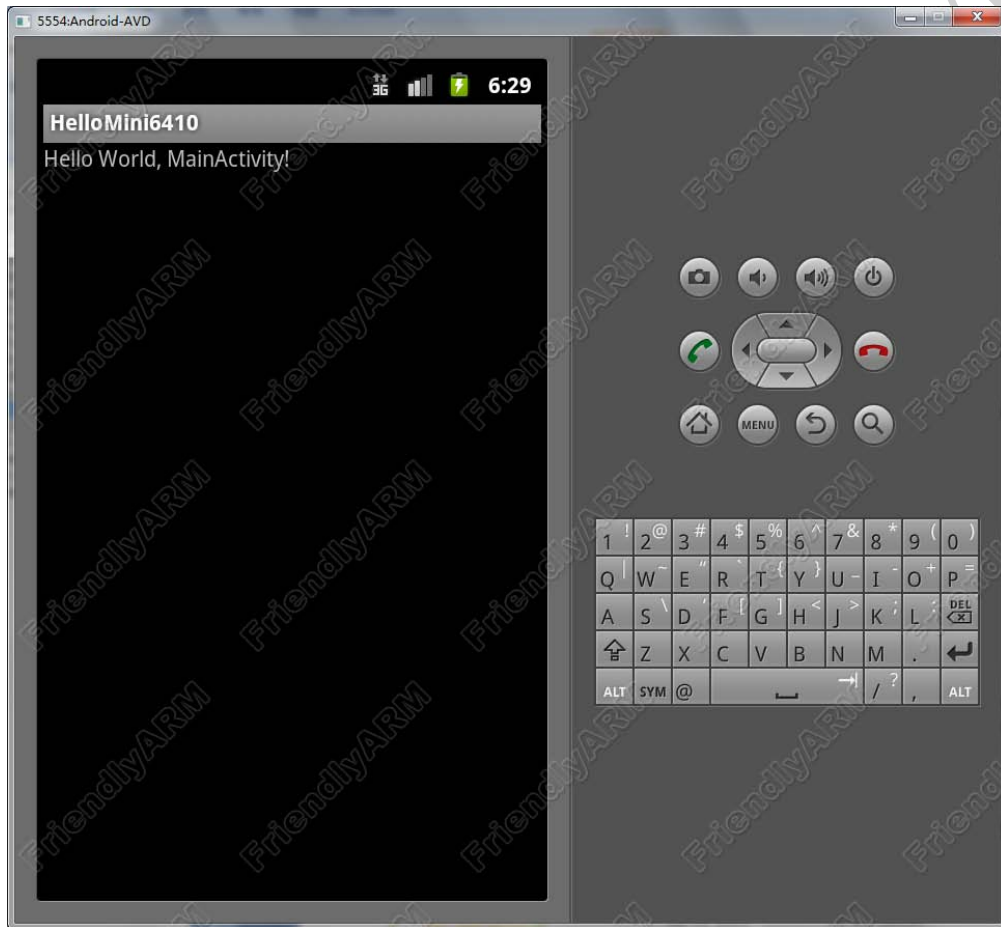
点击 Welcome 后面 X 按钮关闭欢迎界面，将启动工程视图，点击左侧的 src -> com.mini6410 -> MainActivity.java 就可以看到源代码啦：



## 2.8.2 在模拟器运行 HelloMini6410 程序

要编译并运行 HelloMini6410 程序，先在 Package Explorer 中选中 HelloMini6410 工程名称，然后点击工具栏的运行按钮，或选择菜单：Run->Run As->Android Application 即可。

将会自动启动 Android 模拟器，启动过程因为需要启动 Android 系统，所以要耐心等待，Android 系统启动完成后，将会自动运行 HelloMini6410 程序，如下图所示：



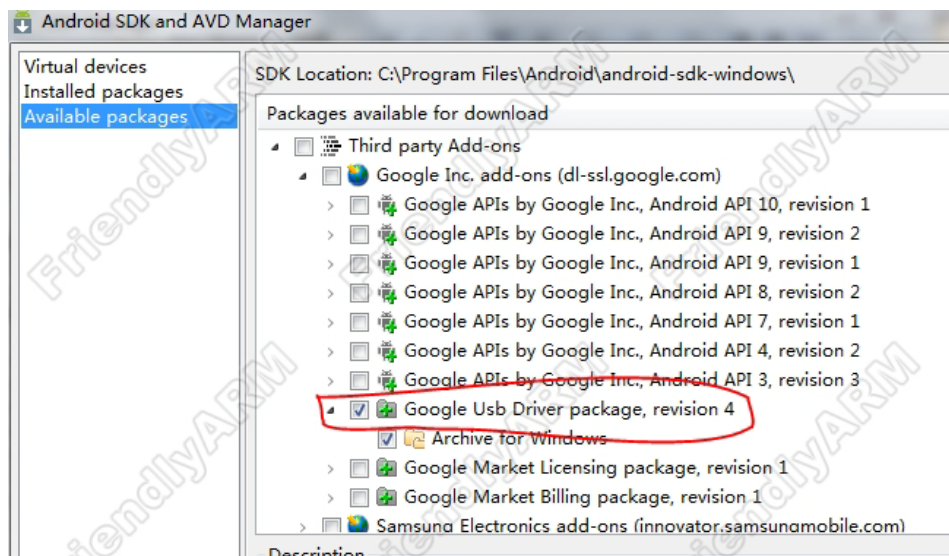
在接下来的章节，我们将实现把这个 HelloMini6410 程序在 mini6410 上调试和运行。

## 2.9 步骤九：建立 Mini6410 调试环境

### 2.9.1 安装 USB ADB 驱动程序

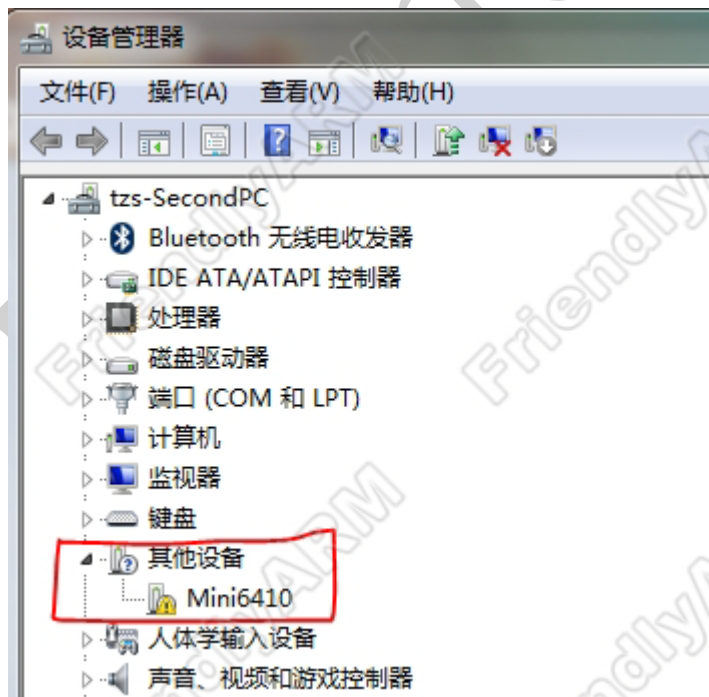
用管理员身份启动 SDK Manager，在 Android SDK and AVD Manager 的主界面上，选择

“Available Packages”，点击 Third party Add-ons 前面的“>”图标展开选项，参考下图选中 Google Usb Driver package 选项：



在上面的对话框中点击“Install Selected”按钮，在弹出的 Choose Packages to Install”对话框中，选中“Accept All”单选框，点击“Install”按钮，将进入下载过程，下载速度视你的网速而定，耐心等待下载完成。

下载完成后，将 Mini6410 开机，在 Android 启动完毕后，插入 MiniUSB 线与 PC 相连，这时，Windows7 会提示正在安装驱动程序，并稍后会提示“驱动程序安装失败”，这时，右击“我的电脑”，选择“属性”，再点击“设备管理器”，会看到一个 Mini6410 的设备：



右击“Mini6410”，选择“更新驱动程序软件”，在弹出的对话框中选择“浏览计算机上的驱动程序文件”，再点击“浏览”，在 Android SDK 安装路径中选择 USB 驱动程序的路径，默认情况下是“C:\Program Files\Android\android-sdk-windows\google-usb\_driver”，选择路径后点击“下一步”进行安装，将弹出如下的安装提示：



在上述对话框中点击“安装”按钮，稍等片刻，得到如下提示表示已安装完成：



Windows 已经成功地更新驱动程序文件

Windows 已经完成安装此设备的驱动程序软件:



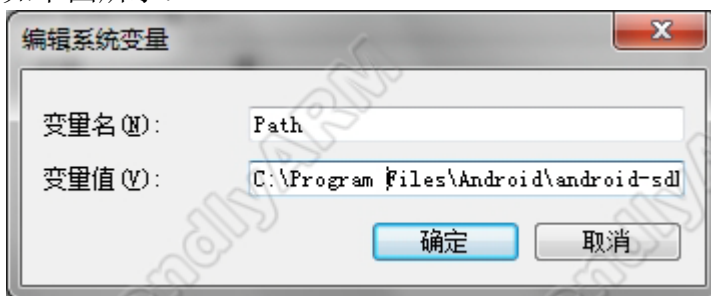
Android Composite ADB Interface

## 2.9.2 在 Mini6410 上测试 ADB 功能

### 2.9.2.1 将 adb 命令添加到 Path 环境变量中

通过下面的方法将 adb 命令所在的路径添加到 Path 环境变量中：

- 1) 右击“我的电脑”->属性，再选择左边导航的“高级系统设置”选项。
- 2) 点击右下角的“环境变量”选项。
- 3) 在“系统变量”中，找到 Path 环境变量，双击它，在变量值前面追加以下内容：“C:\Program Files\Android\android-sdk-windows\platform-tools;”，注意后面有一个分号，如下图所示：



- 4) 点击“确定”完成环境变量设置。

#### 测试一下是否找到 adb 命令

通过点击开始菜单，在开始菜单下方的搜索框中输入 cmd，在 cmd.exe 上按回车来启动 DOS 窗口，在 DOS 窗口中，输入 adb 按回车，如果显示以下信息表示环境变量设置 OK：



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7600]
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

C:\Users\tzs>adb
Android Debug Bridge version 1.0.26

-d                - directs command to the only connected USB device
e                returns an error if more than one USB device is
present.
-e                - directs command to the only running emulator.
returns an error if more than one emulator is r
unning.
-s <serial number> - directs command to the USB device or emulator w
ith
the given serial number. Overrides ANDROID_SERI
AL
environment variable.
-p <product name or path> - simple product name like 'sooner', or
a relative/absolute path to a product
out directory like 'out/target/product/sooner'.
If -p is not specified, the ANDROID_PRODUCT_OUT
```

## 2.9.2.2 测试 ADB 的功能

### 查看设备连接状态

在 mini6410 上启动 Android, 然后用 mini USB 线将 mini6410 与 PC 相连, 在 DOS 窗口上输入以下命令验证开发板是否已连接:

```
# adb devices
```

显示以下内容表示成功连接到 Mini6410 设备:

```
C:\Users\tzs>adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
MINI6410_2011W08      device
```

### 进入 ADB Shell

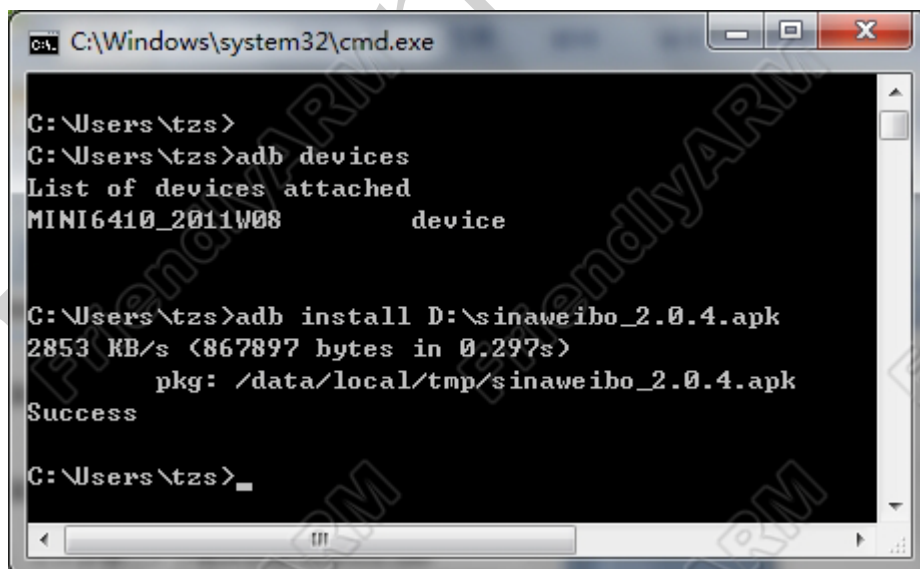
使用以下命令可以进入 mini6410 终端:

```
# adb shell
```

在 mini6410 终端上输入 exit 退回到 DOS 提示符。

### 用 ADB 安装软件

以安装 D:\sinaweibo\_2.0.4.apk 的程序为例, 在 DOS 窗口中输入 **adb install D:\sinaweibo\_2.0.4.apk** 进行安装, 如下图所示:



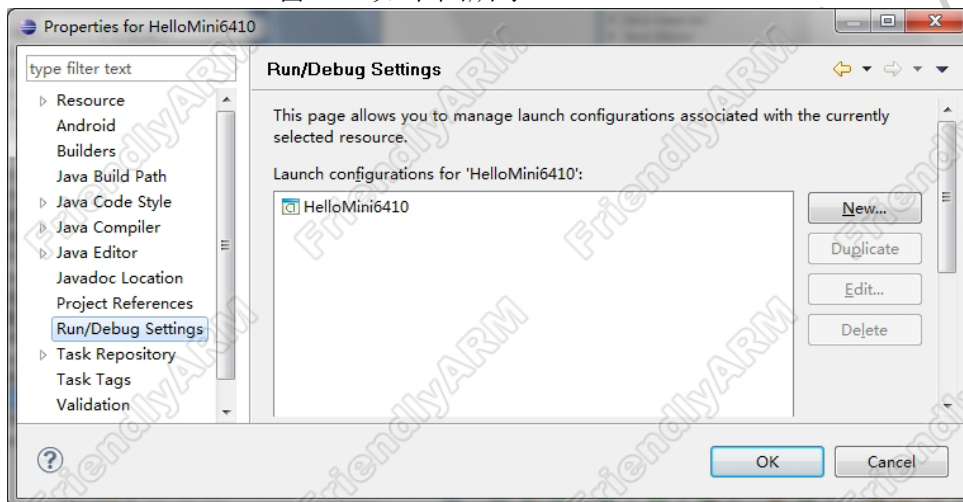
### 其它功能

ADB 功能非常强大，除了安装软件、调试、Shell 功能外，还可以往开发板上传送文件等，读者可以自己挖掘。

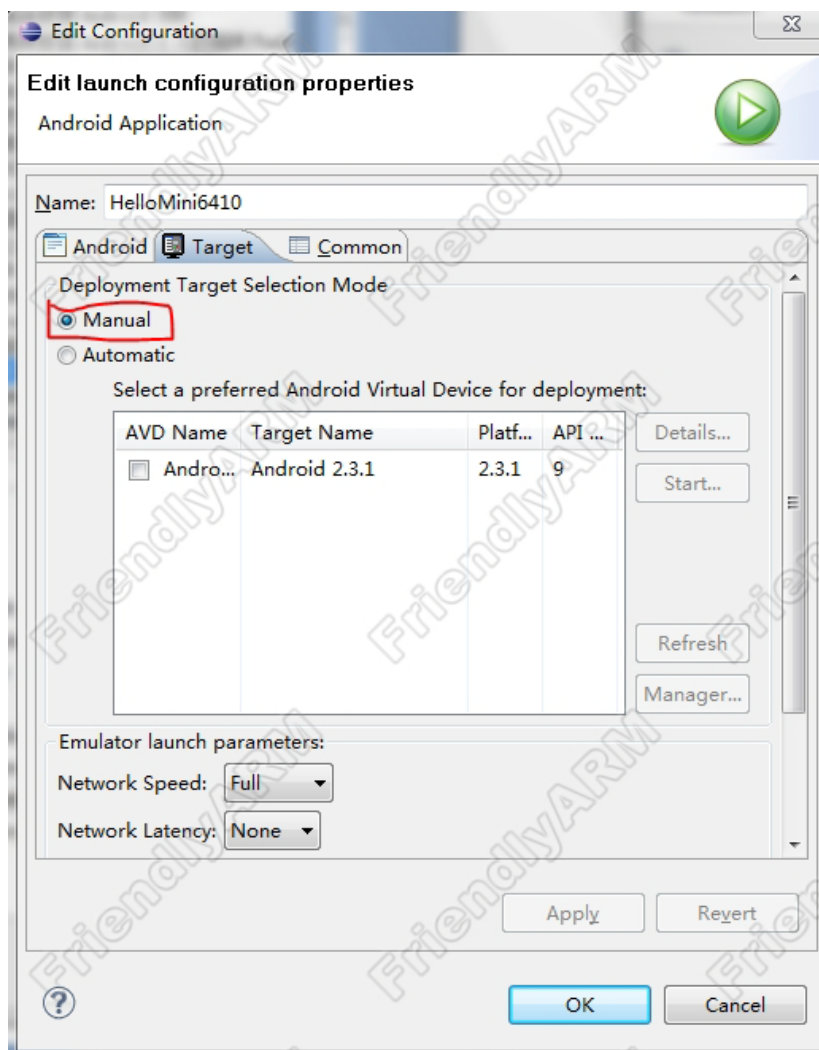
### 2.9.3 通过 USB ADB 在 Mini6410 上运行程序

启动 Eclipse，将自动打开 HelloMini6410 工程，如果没有打开，可手动打开。

在 Eclipse 主界面左侧的 Package Explorer 中右击 HelloMini6410 项目，点 Properties，将弹出 Properties for HelloMini6410 窗口，如下图所示：

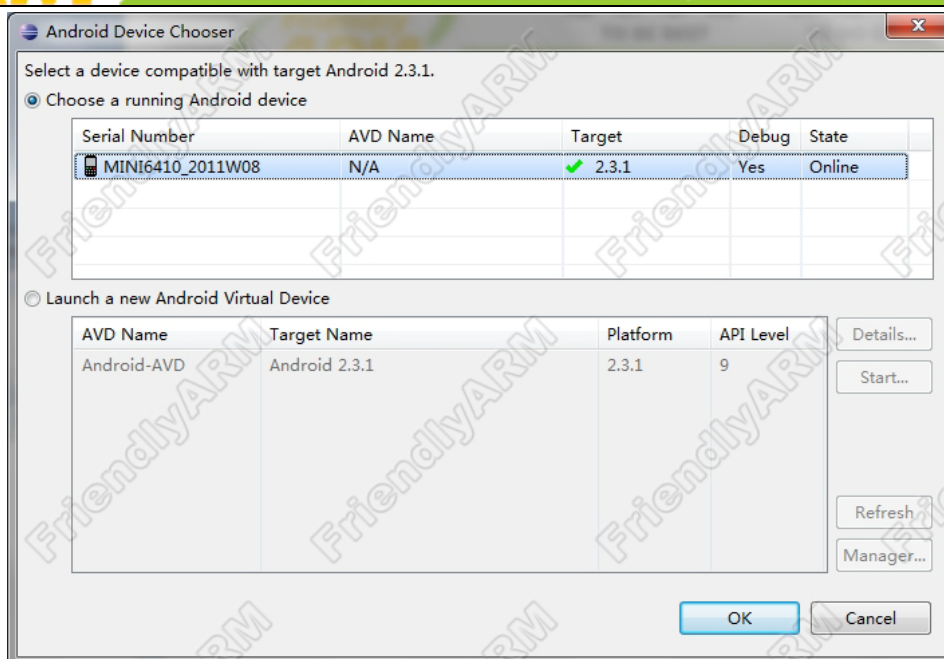


在窗口中点击“Run/Debug Settings”，选择中间列表中的 HelloMini6410，然后在右边点击“Edit...”按钮，将弹出 Edit Configuration 窗口 点击“Target”，在 Deployment Target Selection Mode 上选择 Manual，如下所示：



点击 OK 保存并退出。

接下来是激动人心的时刻，选中 HelloMini6410 工程名称，然后点击工具栏的运行按钮，或选择菜单：Run->Run As->Android Application，会弹出“Android Device Chooser”对话框，在其中选择“Choose a running Android device”，然后在列表中选中 Target 为 2.3.1 的设备(也就是 mini6410 啦)，如下图所示，完成后点击“OK”按钮：

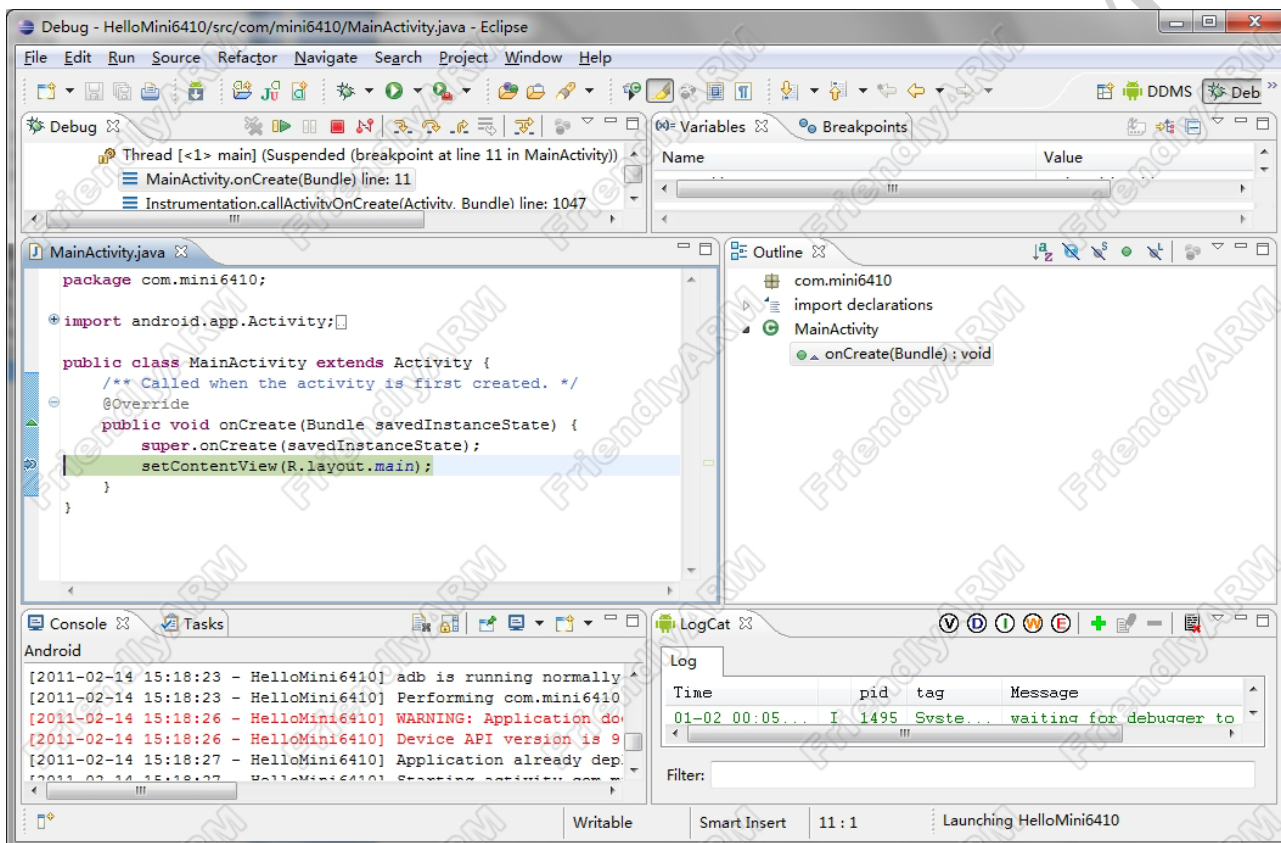


略等片刻，HelloMini6410 就在 mini6410 上运行起来啦：



## 2.9.4 在 Mini6410 上调试 Android 程序

先在 Mini6410 上退出 HelloMini6410 程序，然后在 Eclipse 主界面上，点击菜单“Open Perspective”下的“Debug”，即可进入调试视图，我们尝试在源代码上设置一个断点，最后点击菜单“Run”下的“Debug”重新运行程序，如下图所示，程序运行在我们设置的断点就暂停了，表示真机调试环境建立成功：



## 第三章 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 Mini6410 或者 Tiny6410 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。

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

友善之臂在 2011 年 3 月的 Android 更新中，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/mini6410/prebuilt/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;

public class HardwareControler
{
    /* Serial Port */
    static public native int openSerialPort( String devName, long baud, int dataBits,
int stopBits );
```



```
/* LED */
static public native int setLedState( int ledID, int ledState );

/* PWM */
static public native int PWMPlay(int frequency);
static public native int PWMStop();

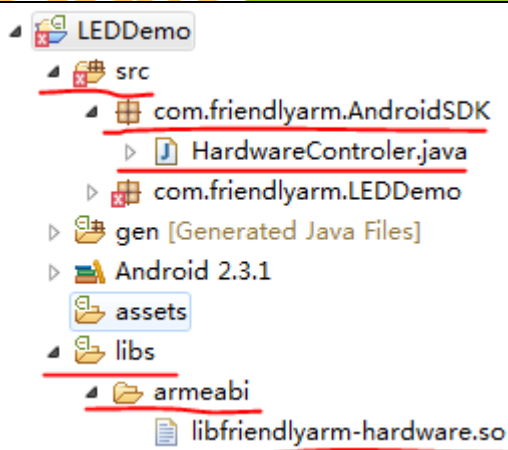
/* ADC */
static public native int readADC();

/* I2C */
static public native int openI2CDevice();
static public native int writeByteDataToI2C(int fd, int pos, byte byteData);
static public native int readByteDataFromI2C(int fd, int pos);

/* 通用接口 */
static public native int write(int fd, byte[] data);
static public native int read(int fd, byte[] buf, int len);
static public native int select(int fd, int sec, int usec);
static public native void close(int fd);

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。

## 3.2 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：

### 3.2.1 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
<b>int</b> openSerialPort( String devName, <b>long</b> baud, <b>int</b> dataBits, <b>int</b> stopBits )	参数说明： devName: 串口设备文件名，可选的值有： /dev/s3c2410_serial1 /dev/s3c2410_serial2 /dev/s3c2410_serial3 /dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2	打开指定的串口设备，并返回文件描述符。



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

	<p>/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>	
<pre>int write(     int fd,     byte[] data)</pre>	<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。
<pre>int read(     int fd,     byte[] buf,     int len)</pre>	<p>参数说明:</p> <p>fd: 要读出数据的文件描述符</p> <p>buf: 存储数据的缓冲区</p> <p>len: 要读取的字节数</p> <p>返回值说明:</p> <p>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>	从打开的设备或文件中读取数据。
<pre>int select(     int fd,     int sec,     int usec)</pre>	<p>参数说明:</p> <p>fd: 要查询的文件描述符</p> <p>sec: 阻塞等待数据多长时间 (单位: 秒)</p> <p>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:</p> <p>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:</p> <p>fd: 要关闭的文件描述符</p> <p>返回值说明:</p> <p>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。

### 3.2.2 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
<code>int setLedState(     int ledID,     int ledState )</code>	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。

### 3.2.3 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
<code>int PWMPlay(int frequency);</code>	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
<code>int PWMStop();</code>	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

### 3.2.4 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
<code>int readADC()</code>	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

### 3.2.5 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>	<p>参数说明： 无</p> <p>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。</p>	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>	<p>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</p> <p>返回值说明： 成功返回写入的字节数，出错返回-1。</p>	<p>往 EEPROM 中写入数据 (每次只能写一个 byte)。</p> <p>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>
<code>int readByteDataFromI2C(int fd, int pos);</code>	<p>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)</p> <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 <code>read</code> 之前已到达文件末尾, 则这次 <code>read</code> 返回 0。 返回值的类型是 <code>int</code>, 你需要转换成 <code>byte</code>。</p>	<p>从打开的设备或文件中读取数据。</p> <p>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>
<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。



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

EEPROM 操作完毕后，需要调用 `close` 关闭文件描述符。

### 3.3 示例程序说明

在光盘 A 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。