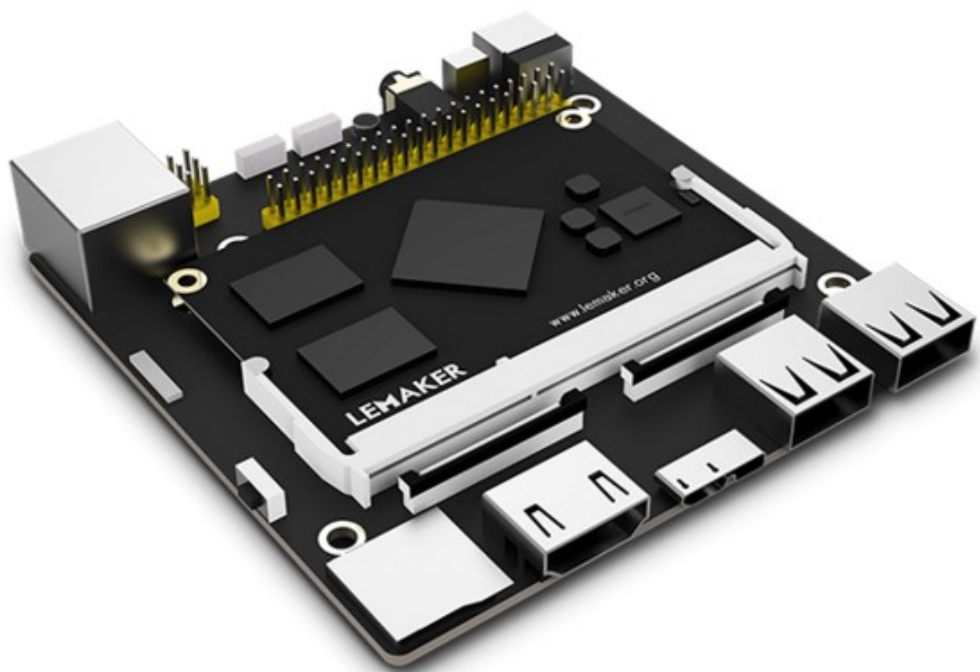


# LeMaker Guitar 用户手册

(第一版 )



## 版本历史

| 版本  | 日期         | 作者      | 描述                                          |
|-----|------------|---------|---------------------------------------------|
| 1.0 | 2015-11-12 | LeMaker | 适用于 LeMaker V1.3 与 Lemuntu for guitar v1511 |
|     |            |         |                                             |
|     |            |         |                                             |
|     |            |         |                                             |

LeMaker

## 目录

|                                           |    |
|-------------------------------------------|----|
| 1 介绍.....                                 | 4  |
| 1.1 LeMaker Guitar 介绍.....                | 4  |
| 1.2 Linux 系统介绍.....                       | 4  |
| 1.3 需要的配件.....                            | 6  |
| 1.4 配件连接.....                             | 8  |
| 1.5 更多内容.....                             | 8  |
| 2 安装系统镜像.....                             | 10 |
| 2.1 Windows 下烧写系统镜像到 MicroSD 卡.....       | 10 |
| 2.2 Linux 下烧写系统镜像到 MicroSD 卡.....         | 12 |
| 2.3 Windows 下烧写系统固件到 EMMC NAND Flash..... | 13 |
| 2.4 Linux 下烧写系统固件到 EMMC NAND Flash.....   | 15 |
| 2.5 操作系统镜像分区描述.....                       | 15 |
| 2.6 另请参阅.....                             | 16 |
| 3 基本使用.....                               | 17 |
| 3.1 显示输出.....                             | 17 |
| 3.1.1 HDMI 接口.....                        | 17 |
| 3.1.2 LVDS 接口.....                        | 18 |
| 3.2 音频.....                               | 21 |
| 3.3 WIFI.....                             | 22 |
| 3.3.1 WIFI station 模式.....                | 22 |
| 3.3.2 WIFI AP 模式.....                     | 26 |
| 3.4 蓝牙.....                               | 28 |
| 3.5 摄像头.....                              | 30 |
| 3.6 存储.....                               | 31 |
| 3.6.1 USB.....                            | 31 |
| 3.6.2 Micro SD 卡和 EMMC NAND Flash.....    | 34 |
| 3.7 ADC.....                              | 34 |
| 3.8 另请参阅.....                             | 35 |
| 4 扩展根文件系统分区.....                          | 36 |
| 5 CPU.....                                | 38 |
| 5.1 调整 CPU 主频.....                        | 38 |
| 5.2 锁频.....                               | 39 |
| 5.3 读取 CPU&PMU 内部温度.....                  | 40 |
| 5.4 另请参阅.....                             | 40 |
| 6 远程登录.....                               | 42 |
| 6.1 使用 SSH 登录.....                        | 42 |
| 6.1.1 Windows 上远程登录.....                  | 42 |
| 6.1.2 Linux 下远程登录.....                    | 44 |
| 6.2 使用 vnc 远程控制 SBC.....                  | 45 |

|                                      |    |
|--------------------------------------|----|
| 6.3 使用 RDP 登录.....                   | 46 |
| 6.4 另请参阅.....                        | 48 |
| 7 视频.....                            | 49 |
| 7.1 GPU 硬件加速.....                    | 49 |
| 7.1.1 在 LeMaker Guitar 上安装硬件加速库..... | 49 |
| 7.1.2 测试 OpenGL ES.....              | 50 |
| 7.1.3 测试 OpenCL.....                 | 51 |
| 7.2 视频解码.....                        | 54 |
| 7.3 另请参阅.....                        | 55 |
| 8 IO 控制.....                         | 56 |
| 8.1 使用 sysfs 用户空间.....               | 56 |
| 8.2 WiringLMK.....                   | 58 |
| 8.3 LMK.GPIO.....                    | 58 |
| 8.4 LeScratch.....                   | 59 |
| 8.5 另请参阅.....                        | 60 |
| 9 备份系统镜像.....                        | 61 |
| 9.1 使用 Win32Diskimager 备份系统镜像.....   | 61 |
| 9.2 使用 dd 命令备份系统镜像.....              | 62 |
| 9.3 另请参阅.....                        | 62 |
| 10 Android.....                      | 63 |
| 10.1 基本屏幕.....                       | 63 |
| 10.2 基本设置.....                       | 64 |

# 1 介绍

## 1.1 LeMaker Guitar 介绍

[LeMaker Guitar](#) 是由 LeMaker 团队设计的一款高性能，低功耗 SoM。它使我们能够建立一个个人的、多功能的系统。实际上，它就是一台完整的迷你电脑，它包含了所有必须的部件，例如处理器、内存、网络以及其它接口。LeMaker Guitar 采用 S500 SOC、4 核 ARM® Cortex™-A9 CPU、1GB/2GB DDR3 SDRAM 和 8GB EMMC Flash。

为了使用户更方便地利用 LeMaker Guitar DIY 自己的作品，有效地减少用户自己定制 LeMaker Guitar 底板的设计时间，LeMaker 团队设计一些 LeMaker Guitar 底板。如果你不想花费任何时间和精力设计底板，很幸运，LeMaker 团队也提供了 3 种底板：底板 Rev.B，底板 Rev.C 和底板 Rev.D。底板 Rev.B 是标准型底板，底板 C 是增强型底板，底板 D 是发烧型底板。推荐使用标准型底板 Rev.B。

LeMaker Guitar 能够很好的兼容各种软件。基本上所有的基于 Linux 的主流操作系统都能运行在 LeMaker Guitar 上，例如 Android，Ubuntu-MATE，Arch Linux 等。Lemuntu 和 LeMedia 是由 LeMaker 团队基于 Linux 系统，专门为 LeMaker SBC 系列开发的操作系统。

## 1.2 Linux 系统介绍

[Linux](#) 是运行在 LeMaker Guitar 上的操作系统（OS）。它的角色与 [Windows](#)，[Mac OS X](#)，Android（事实上，Android 基于 Linux 内核），iOS 或你知道的任何其它操作系统扮演的角色一样。这个角色提供了一个平台，让一切来运行。它与硬件对话，并与作为用户的你对话。

但是，什么使得 Linux 不同于任何其他操作系统呢？一开始它是免费的，非常强大的，高度可定制并且最好的一点是它由用户为用户创建的。然而把 Linux 称作“一个操作系统”是有点轻描淡写的。与 Windows 8 或 Mac OS X 是一个操作系统不同，Linux 不是“一个操作系统”。Linux 有许多操作系统，甚至上百个操作系统。

Linux 由不同的组件构成，每一个组件都具有许多不同的变体。这些都被打包成易于安装的安装包，以满足不同的需要。想要一个简单的台式机替代品吗？这里有一个 Linux 发行版的。想要一个家庭媒体服务器吗？这里也有一个 Linux 发行版的。如果你能想起来，也许有人已经

在 Linux 社区开发它了。

## Lemuntu

[Lemuntu](#) 是一个基于 Debian Jessie 版本为 LeMaker SBC 系列优化的免费开源操作系统。为了充分利用 SBC 的硬件资源，提升 SBC 用户体验，优化操作系统是有必要的。Lemuntu 默认安装了一个轻量级的图形桌面环境 LXDE。LeMaker 团队已经初步优化 Lemuntu 系统，它可以流畅地在 LeMaker Guitar 上运行。同时，为了提高 Lemuntu 的稳定性和友好性，Lemuntu 系统仍处于持续开发当中。

## Android

[Android](#) 是由 Google 公司开发的基于 Linux 内核的移动设备操作系统。Android 带有可以直接操作的用户界面，最初为触屏设备设计，例如智能手机、平板电脑、有特定用户界面的电视机、汽车和腕戴手表。Android 操作系统使用触摸输入，与真实世界的动作一致，像滑动、点击、捏合和扩大这些动作来操作屏幕上的目标和虚拟键盘。尽管最初为触摸屏设备设计，但是它也可以用于游戏机、数码照相机、个人电脑和其他电子产品。

## Ubuntu-MATE

[Ubuntu MATE](#) 是一个免费开源的 Linux 发行版本，是 Ubuntu 官方的衍生版本。它与 Ubuntu 的最大不同之处在于它默认使用基于 Ubuntu 以前版本 11.04 使用的 GNOME 开发的 MATE 桌面环境，而不是 Ubuntu 桌面默认的 Unity 图形用户界面。

## LeMedia

[LeMedia](#) 是一个基于 Debian 和 Kodi 定制，专门用于 LeMaker SBC 定制的多媒体操作系统。LeMedia 可以运行在 LeMaker SBC 上，作为多媒体中心，播放游戏和从本地或者网络存储媒体和因特网上观看大部分的视频、音乐、图片和其他数字多媒体文件。

## Arch Linux

[Arch Linux](#) 是基于 IA-32 和 x86 系统架构的 Linux 系统发行版本。它主要由免费和开源软件构成，并支持参与社区活动。Arch Linux 开发团队的设计原则遵从 KISS 原则(“keep it simple, stupid”),并把它作为通用的准则，侧重于优雅、代码正确性、简约和简单，期望用户愿意做出一些努力来了解系统的运行情况。pacman 是一个 Arch Linux 专有的软件包管理工具，用于安装、卸载和更新软件包。Arch Linux 使用滚动发行的模式，定期的系统更新需要获得最新的 Arch 软件；Arch 团队发布的安装镜像仅仅是更新了主要的系统组件快照。

**注意：**如无特殊说明，本用户手册中的所有命令和操作都是在 LeMaker Guitar 底板 Rev.B 和 Lemuntu 系统下进行的。

## 1.3 需要的配件

使用 LeMaker Guitar，我们需要一些必要的配件配合 LeMaker Guitar 使用：

表 1.1 配件表

|                          |                                                                                                            |
|--------------------------|------------------------------------------------------------------------------------------------------------|
| <b>Micro SD 卡</b>        | 建议使用 4GB 或更大的 class 10 MicroSD 卡, 建议购买 8GB 的 SD 卡。                                                         |
| <b>显示器和连接线</b>           | HDMI 或 DVI 或 VGA 接口的显示屏或电视机可以当作显示器。如果显示器使用 DVI 或 VGA 作为输入接口,需要一根 HDMI 转 DVI 线或 HDMI 转 VGA 线。               |
| <b>键盘鼠标</b>              | 标准的键盘和鼠标都可以满足使用条件。                                                                                         |
| <b>电源</b>                | 使用 5~12V@2A 电源适配器供电，插头内径 2.1 毫米，插头外径 5.5 毫米，里面正极，外面负极。                                                     |
| <b>网络连接</b>              | 为了能够更新和下载软件，建议通过以太网线或无线 WIFI 连接到 Internet。                                                                 |
| <b>USB3.0/USB2.0 连接线</b> | 为了能够从 EMMC NAND FLASH 启动操作系统， 需要使用 USB3.0/USB2.0 连接线(A/M-Micro-B)连接 LeMaker Guitar 来烧写镜像到 EMMC NAND FLASH。 |

配件图片如下：



无线键鼠



电源适配器



以太网线



HDMI 连接线



MicroSD 卡



USB3.0 连接线



HDMI 转 DVI 连接线



HDMI 转 VGA 连接线



USB2.0 连接线

图 1.1 配件图片

## 1.4 配件连接

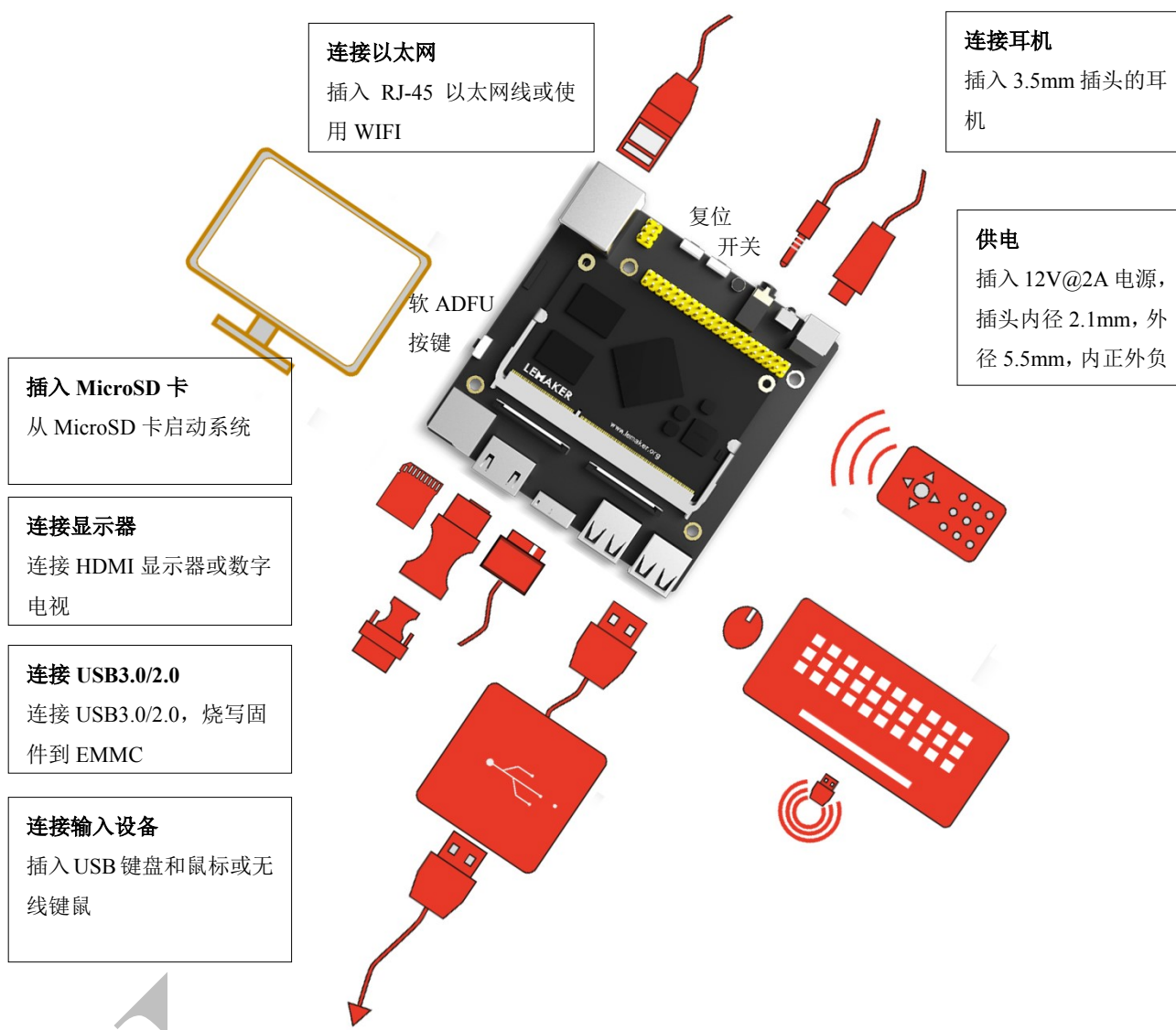


图 1.2 配件连接到 LeMaker Guitar

(警告：请不要带电热插拔核心板，否则可能会烧毁 CPU。)

## 1.5 更多内容

LeMaker 官方主页：<http://www.lemaker.org/cn/>

LeMaker 官方论坛：<http://forum.lemaker.org/cn>

LeMaker Guitar 教程：[http://wiki.lemaker.org/LeMaker\\_Guitar/zh-hans](http://wiki.lemaker.org/LeMaker_Guitar/zh-hans)

LeMaker 技术支持：[support@lemaker.org](mailto:support@lemaker.org)

LeMaker Github: <https://github.com/LeMaker>

LeMaker 博客: <http://www.lemaker.org/cn/blog>

LeMaker 微店商城: <http://weidian.com/?userid=335465484>

LeMaker

## 2 安装系统镜像

LeMaker Guitar 有两种方式启动操作系统，一种方式是从 MicroSD 卡启动操作系统，另一种方式是从板上 EMMC NAND Flash 启动操作系统。在 MicroSD 卡和 EMMC NAND Flash 都安装好操作系统的情况下，如果将 MicroSD 卡插入 MicroSD 卡插槽，操作系统将优先从 MicroSD 卡启动。如果想从 EMMC NAND Flash 启动操作系统，必须先从 MicroSD 卡插槽拔出 MicroSD 卡。LeMaker Guitar 的操作系统镜像有两种格式：一种是烧写到 MicroSD 卡的 镜像文件，另一种是烧写到 EMMC NAND Flash 的 固件文件。我们可以从 [LeMaker 资源下载页面](#) 下载系统镜像文件，并且在 Windows 系统下用 [7-Zip](#) 工具或在 Linux 系统下用 [7za](#) 工具解压压缩文件。

### Windows:

安装 7-zip 软件工具，解压操作系统压缩文件。

### Linux:

```
sudo apt-get install p7zip-full  
7za x [imagefilename].7z
```

如果想从 MicroSD 卡启动操作系统，我们需要安装操作系统镜像到 MicroSD 卡；但是如果从 EMMC NAND Flash 启动操作系统，我们需要安装操作系统镜像到 MMC NAND Flash。LeMaker Guitar 上有一个“硬 ADFU”按键，LeMaker Guitar 底板上有一个“软 ADFU”按键，用于进入 ADFU 模式。但是建议使用底板上的“软 ADFU”按键进入 ADFU 模式。下面的操作将告诉我们如何烧写系统镜像到 MicroSD 卡和 EMMC NAND Flash。

注意：“硬 ADFU”意思是通过“硬 ADFU”按键切断 EMMC 或 NAND Flash 和 SOC 之间的数据线路，导致控制器不能发现外部存储介质，然后进入 ADFU 模式。“软 ADFU”意思是 u-boot 将会检测“软 ADFU”按键的状态，然后调用相关的函数进入 ADFU 模式。如果 u-boot 出错，那么只能使用硬“ADFU”按键进入 ADFU 模式去烧写系统固件。

### 2.1 Windows 下烧写系统镜像到 MicroSD 卡

**第 1 步：**把 MicroSD 卡插入电脑或读卡器，MicroSD 卡的容量要大于系统镜像的大小，一般是 4GB 或更大。



图 2.1 MicroSD 卡

**第 2 步：**格式化 SD 卡。下载并安装 MicroSD 卡格式化工具 SD Formatter，下载链接：

[http://mirror.lemaker.org/SDFormatter\\_v4.zip](http://mirror.lemaker.org/SDFormatter_v4.zip)。以管理员身份运行 SD Formatter，在“选项设置”中，选择格式化类型为“快速格式化”，选择逻辑大小调整为“开启（ON）”，点击“OK”按钮完成设置，点击“格式化”按钮格式化 MicroSD 卡。



图 2.2 SDFormatter

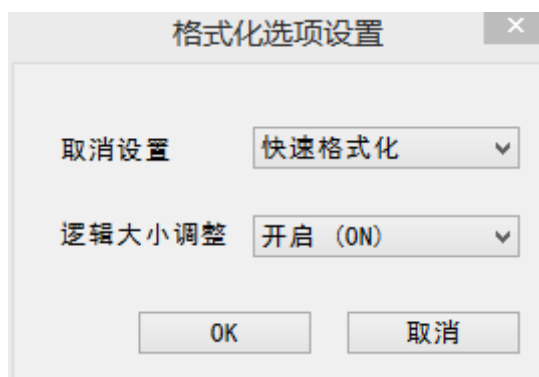


图 2.3 SDFormatter 设置

**第 3 步：**烧写镜像到 SD 卡。下载并安装系统镜像烧写工具 **Win32DiskImager**，下载链接：

[http://mirror.lemaker.org/Win32DiskImage\\_v0.9.5\\_install.zip](http://mirror.lemaker.org/Win32DiskImage_v0.9.5_install.zip)。以管理员身份运行

Win32DiskImager，打开系统镜像文件，点击“Write”按钮，烧写系统镜像到 MicroSD 卡，耐心等待，直到烧写成功。

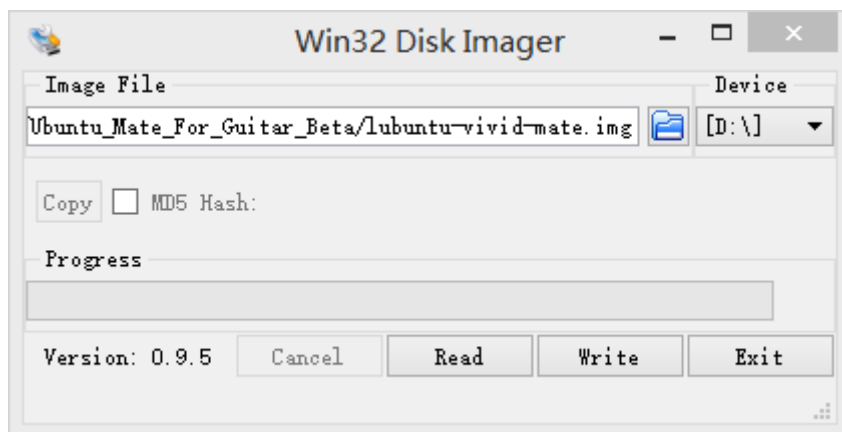


图 2.4 Win32 Disk Imager

## 2.2 Linux 下烧写系统镜像到 MicroSD 卡

**第 1 步：**把 MicroSD 卡插入电脑或读卡器，MicroSD 卡的容量要大于系统镜像的大小，一般是 4GB 或更大。



图 2.5 MicroSD 卡

**第 2 步：**在命令终端中，查看 MicroSD 卡设备。

```
sudo fdisk -l
```

**第 3 步：**安装镜像烧写工具

**Debian / Ubuntu 系统:**

```
sudo apt-get install ddrescue
```

**RedHat /Fedora /CentOS 系统:**

```
yum -y install ddrescue
```

**第 4 步：**写入系统镜像到 SD 卡。

```
sudo dd_rescue -A [path]/[imagename].img dev/sdx
sync
```

耐心等待，直到烧写成功。

## 2.3 Windows 下烧写系统固件到 EMMC NAND Flash

**第 1 步：** 下载并安装系统烧写工具 **IH 固件烧写工具**。下载链接：

[http://mirror.lemaker.org/IH\\_FW\\_Burning\\_Tool\\_For\\_Windows\\_V2.01.03.zip](http://mirror.lemaker.org/IH_FW_Burning_Tool_For_Windows_V2.01.03.zip)



图 2.6 IH 固件烧录工具

**第 2 步：** 以管理员的身份运行 IH 固件烧写工具，按住“软 ADFU”按键，用 USB3.0/2.0 连接线 LeMaker Guitar，上电大约 6s 后，IH 固件烧写工具检测到 ADFU 设备时，松开“软 ADFU”按键。如果不能检测到 ADFU 设备，请使用核心板上的“硬 ADFU”按键，重复上述过程。

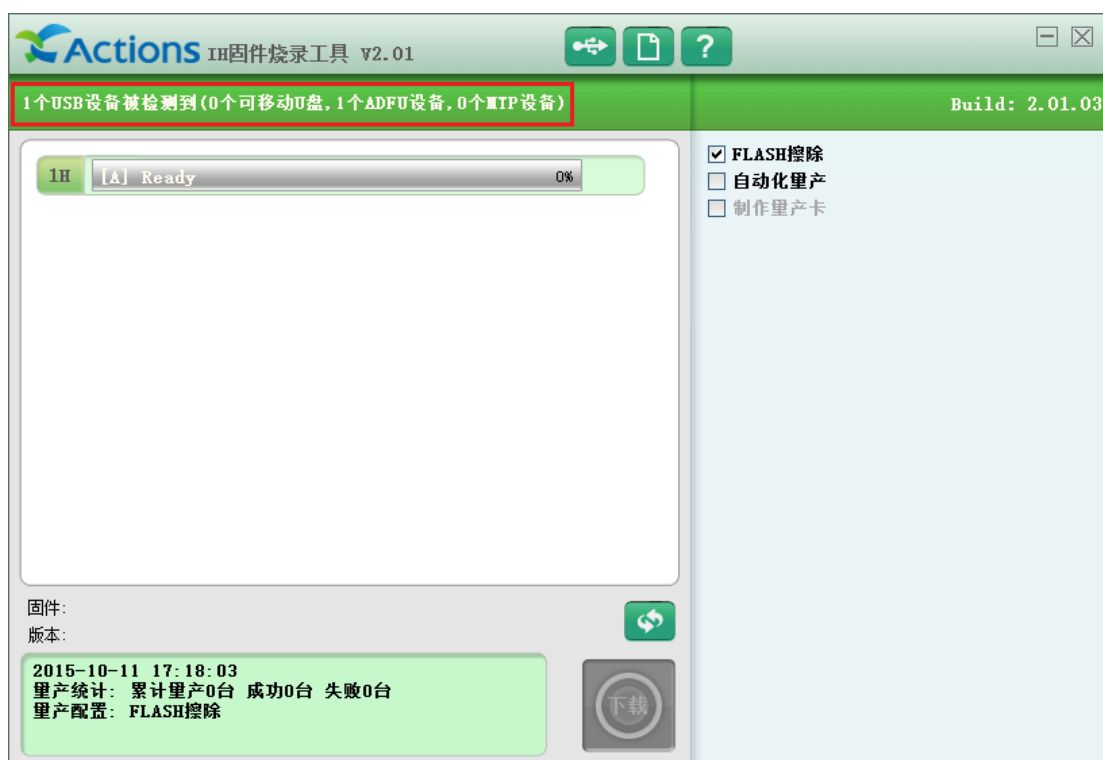


图 2.7 检测 ADFU 设备

**第3步:** 打开固件文件，点击“下载”，烧写系统固件到 EMMC，耐心等待，直到固件烧写完成。

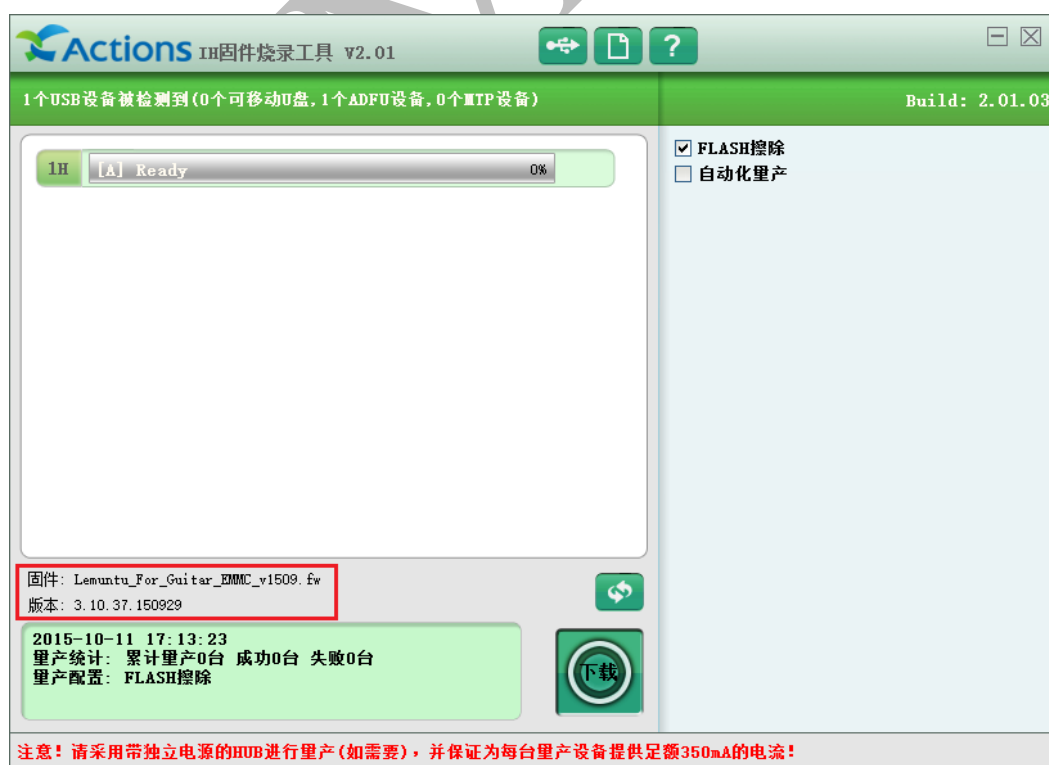


图 2.8 烧写系统固件

## 2.4 Linux 下烧写系统固件到 EMMC NAND Flash

第 1 步：下载固件烧写工具 FW Burning Tool，下载链接：

[http://mirror.lemaker.org/FW\\_Burning\\_Tool\\_For\\_Linux\\_V1.0\\_01.tar.gz](http://mirror.lemaker.org/FW_Burning_Tool_For_Linux_V1.0_01.tar.gz)。

```
wget http://mirror.lemaker.org/FW_Burning_Tool_For_Linux_V1.0_01.tar.gz
```

第 2 步：在命令终端，首先解压软件包，再安装软件。

```
sudo tar -zxvf FWBurning_Tool_For_Linux_V1.0_01.tar.gz
cd FWBurning_Tool_For_Linux_V1.0_01
sudo ./FWBurningTool-1.0.run
```

第 3 步：按住“软 ADFU”按键，用 USB3.0/2.0 连接线连接 LeMaker Guitar，然后给 LeMaker Guitar 上电大约 6s 后，用“lsmod”命令检测到 USB 设备，当找到 USB 设备时，松开“软 ADFU”按键。如果不能检测到 ADFU 设备，请使用“硬 ADFU”按键，重复上述过程。

```
sudo lsusb
...
Bus 001 Device 002: ID 10d6:10d6 Actions Semiconductor Co., Ltd
...
```

第 4 步：烧写镜像到 EMMC NAND Flash。

```
sudo python ./ActionsFWU.py --fw=[firmware_name].fw
```

## 2.5 操作系统镜像分区描述

表 2.1 Linux OS 镜像分区表

| MicroSD 卡系统镜像布局 |          |                                                              |
|-----------------|----------|--------------------------------------------------------------|
| 名字              | 起始地址     | 作用                                                           |
| Mbr             | 0        | MicroSD 卡镜像的第一个启动分区                                          |
| Gpt             | 0x200    | gpt 分区表                                                      |
| Environment     | 0x5000   | uboot 环境变量                                                   |
| Bootloader      | 0x200200 | 第一个启动阶段 boottraps (bootloader.bin)                           |
| Uboot           | 0x300000 | uboot 启动加载程序(u-boot-dtb.img)                                 |
| Misc            | 0x800000 | fat 分区，包括 kernel、initramfs、kernel 配置文件、uEnv.txt 等等(misc.img) |

|        |           |                           |
|--------|-----------|---------------------------|
| Rootfs | 0x3800000 | 文件系统( <b>rootfs.img</b> ) |
|--------|-----------|---------------------------|

## 2.6 另请参阅

[1] 7za 用法: <http://linux.die.net/man/1/7za>。

[2] 7-Zip—一个高压压缩率归档工具: <http://www.7-zip.org/>。

[3] SDFormatter—SD 卡格式化工具: [http://www.sdcard.org/downloads/formatter\\_4/](http://www.sdcard.org/downloads/formatter_4/)。

[4] Win32 Disk Imager—烧写镜像到 USB 棒和 SD/CF 卡工具:

<http://sourceforge.net/projects/win32diskimager/>。

[5] 制作系统镜像方法:

[http://wiki.lemaker.org/LeMaker\\_Guitar:How\\_to\\_make\\_LeMaker\\_Guitar\\_OS\\_image/zh-hans](http://wiki.lemaker.org/LeMaker_Guitar:How_to_make_LeMaker_Guitar_OS_image/zh-hans)。

## 3 基本使用

### 3.1 显示输出

#### 3.1.1 HDMI 接口

HDMI（高清晰度多媒体接口）是一种专有的音频/视频接口，用于从 HDMI 兼容源设备，如一个显示控制器，传送未压缩的视频数据和压缩的或未压缩的数字音频数据到兼容的计算机显示器，视频投影仪，数字电视，或数字音频设备上。HDMI 是一个数字替代模拟视频标准。LeMaker Guitar 有一个 HDMI 接口，可通过 HDMI 连接线连接显示器和 LeMaker Guitar。但是如果显示器采用 DVI 或 VGA 作为输入接口，我们需要一条 HDMI 转 DVI 线或 HDMI 转 VGA 线，照片如下所示：



图 3.1 HDMI 转 DVI 连接线



图 3.2 HDMI 转 VGA 连接线

LeMaker Guitar HDMI 输出支持多种分辨率，我们可以根据自己的 HDMI 显示器的分辨率进行配置。

表 3.1 HDMI 分辨率表

| HDMI 分辨率表 |                  |
|-----------|------------------|
| VID       | 分辨率              |
| 1         | VID640x480P_60   |
| 4         | VID1280x720P_60  |
| 16        | VID1920x1080P_60 |
| 17        | VID720x576P_50   |
| 19        | VID1280x720P_50  |
| 31        | VID1920x1080P_50 |

126

VID1280x720P\_60\_DVI

## 查看 HDMI 输出分辨率

```
cat /sys/devices/b02c0000.hdmi/vid
16      #Resolution is 1920x1080
```

## 设置 HDMI 分辨率

例如：VID1280x720P\_50

**第 1 步：**编辑 dts 文件：*linux-actions-bsp/linux-actions/arch/arm/boot/dts/lemaker\_guitar\_bbb.dts*。

设置 default\_resolution = “720P50HZ”。

**第 2 步：**编辑 dts 文件：*linux-actions-bsp/u-boot-actions/arch/arm/dts/lemaker\_guitar.dts*。设置 default\_resolution = “720P50HZ”。

**第 3 步：**重新编译 LeMaker Guitar BSP。

参考 [http://wiki.lemaker.org/LeMaker\\_Guitar:How\\_to\\_use\\_LeMaker\\_Guitar\\_BSP/zh-hans](http://wiki.lemaker.org/LeMaker_Guitar:How_to_use_LeMaker_Guitar_BSP/zh-hans) 下载并编译 LeMaker Guitar BSP，LeMaker Guitar BSP 包含了上面的两个文件。

```
cd linux-actions-bsp/
make
```

**第 4 步：**替换 kernel.dtb 和 u-boot-dtb.img 文件：

把 *linux-actions-bsp/build/s500/misc/kernel.dtb* 和

*linux-actions-bsp/build/s500/u-boot/u-boot-dtb.img* 复制到 LeMaker Guitar 的 */tmp* 目录，然后执行下面的操作：

```
cd /tmp/
sudo cp kernel.dtb /media/lemaker/misc      #or sudo cp kernel.dtb /media/misc
sudo dd if=u-boot-dtb.img of=/dev/mmcblk0 bs=512 seek=6144
sync
sudo reboot
```

## 3.1.2 LVDS 接口

LVDS (Low-voltage differential signaling), 也叫做 TIA/EIA-644。LVDS 的工作电压低，运行速度快，可以使用较便宜的双绞线传输数据。LeMaker Guitar 默认支持 7 寸 LCD 屏。如果

我们要使用其它尺寸的 LCD 屏，则要像 HDMI 一样配置它的分辨率。例如 1024\*600\_60。

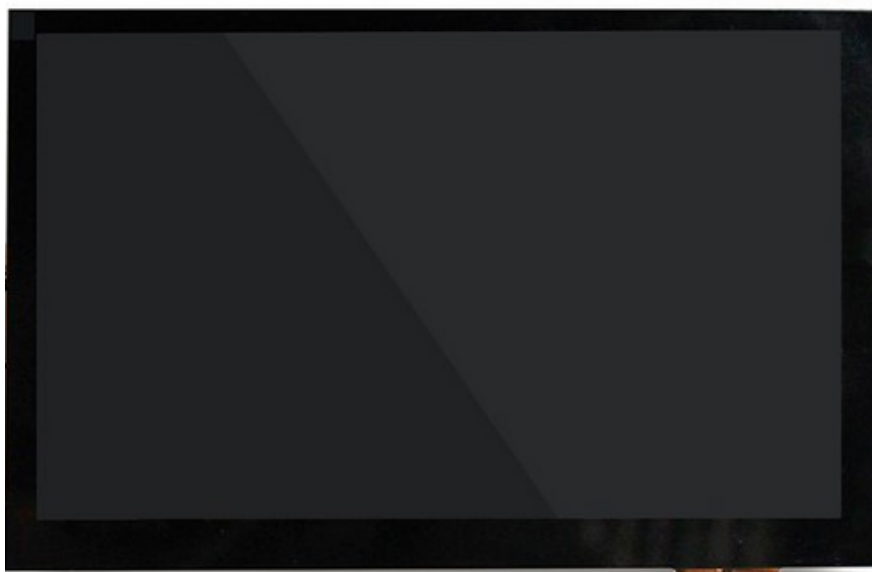


图 3.3 LCD 模块

**第 1 步:** 编辑 dts 文件: *linux-actions-bsp/linux-actions/arch/arm/boot/dts/lemaker\_guitar\_bbb.dts*。

设置 refresh = <60>, xres = <1024>, yres = <600>。

```

lcd0: lcd0@b02a0000 {
    pinctrl-names = "default";
    pinctrl-0 = <&lvds_state_default>;
    lcd_power_gpios = <&gpio 36 0>;      /*GPIOB4*/
    lcd_reset_gpios = <&gpio 49 0>;      /*GPIOB17*/
    port_type = "lvds";
    data_width = <0>;                    /* 0:24bit ,1:18bit */
    videomode-0 = <&lcd_mode0>;
    vsync_inversion = <0>;
    hsync_inversion = <0>;

    dclk_inversion = <0>;
    lde_inversion = <0>;
    lvds_ctl = <0x000a9523>;              /* last two bit: 3:8bit ,1:6bit */
    lvds_alg_ctl0 = <0xc141a030>;
    lightness = <128>;
    saturation = <7>;
    contrast = <5>;
  }

```

```

    lcd_mode0:mode@1024x600p60 {
        refresh = <60>;                /* refresh frequency*/
        xres = <1024>;                  /* width */
        yres = <600>;                   /* height*/
        /*in pico second, 0.000 000 000 001s*/
        pixclock = <20000>;
        left_margin = <56>;
        right_margin = <60>;
        upper_margin = <13>;
        lower_margin = <18>;
        hsync_len = <4>;
        vsync_len = <4>;
        /*0: FB_VMODE_NONINTERLACED, 1:FB_VMODE_INTERLACED*/
        vmode = <0>;

    };
};

```

**第2步：**编辑 dts 文件：*linux-actions-bsp/u-boot-actions/arch/arm/dts/lemaker\_guitar.dts*。设置 refresh = <60>, xres = <1024>, yres = <600>。

```

lcd0: lcd0@b02a0000 {
    pinctrl-names = "default";
    pinctrl-0 = <&lvds_state_default>;
    lcd_power_gpios = <&gpio 36 0>;    /*GPIOB4*/
    lcd_reset_gpios = <&gpio 49 0>; /*GPIOB17*/
    port_type = "lvds";
    data_width = <0>;                /* 0:24bit ,1:18bit */
    videomode-0 = <&lcd_mode0>;
    vsync_inversion = <0>;
    hsync_inversion = <0>;

    dclk_inversion = <0>;
    lde_inversion = <0>;
    lvds_ctl1 = <0x000a9523>;        /* last two bit: 3:8bit ,1:6bit */
    lvds_alg_ctl0 = <0xc141a030>;
    lightness = <128>;
    saturation = <7>;
    contrast = <5>;

    lcd_mode0:mode@1024x600p60 {
        refresh = <60>;
        xres = <1024>;

```

```
        yres = <600>;
        /*in pico second, 0.000 000 000 001s*/
        pixclock = <20000>;
        left_margin = <56>;
        right_margin = <60>;
        upper_margin = <13>;
        lower_margin = <18>;
        hsync_len = <4>;
        vsync_len = <4>;
        /*0: FB_VMODE_NONINTERLACED, 1:FB_VMODE_INTERLACED*/
        vmode = <0>;

    };

};
```

**第3步：**重新编译 LeMaker Guitar 的 BSP。

参考 [http://wiki.lemaker.org/LeMaker\\_Guitar:How\\_to\\_use\\_LeMaker\\_Guitar\\_BSP/zh-hans](http://wiki.lemaker.org/LeMaker_Guitar:How_to_use_LeMaker_Guitar_BSP/zh-hans) 下载并编译 LeMaker Guitar BSP，LeMaker Guitar BSP 包含了上面的两个文件。

```
cd linux-actions-bsp/
make
```

**第4步：**替换 kernel.dtb 和 u-boot-dtb.img。

把 `linux-actions-bsp/build/s500/misc/kernel.dtb` 和 `linux-actions-bsp/build/s500/u-boot/u-boot-dtb.img` 复制到 LeMaker Guitar 的 `/tmp` 目录下，然后执行下面的操作：

```
cd /tmp/
sudo cp kernel.dtb /media/lemaker/misc      #or sudo cp kernel.dtb /media/misc
sudo dd if=u-boot-dtb.img of=/dev/mmcblk0 bs=512 seek=6144
sync
sudo reboot
```

## 3.2 音频

LeMaker Guitar 默认通过底板上的标准 3.5mm 耳机座输出音频，可以接耳机或音箱播放音乐。也可以通过 HDMI 输出音频，但是要先安装 `pulseaudio` 工具并编辑 `/etc/pulse/default.pa` 文件（Lemuntu 系统），参考下面内容进行配置：

```
sudo apt-get install pulseaudio
sudo nano /etc/pulse/default.pa
...
### use module-udev-detect -- see below -- for doing this automatically)
```

```
#load-module module-alsa-sink device=hw:0,0 # 配置 3.5mm Jack 输出
load-module module-alsa-source device=hw:1,0 # 配置 HDMI 输出
...
```

## 3.3 WIFI

LeMaker Guitar 使用 Realtek RTL8723BS 作为板载 WIFI 模块，它集成了 WIFI+BT 功能模块。RTL8723BS 的尺寸小，且将 WIFI+蓝牙组合模块用邮票孔封装，电路板尺寸为长 14mm\*宽 14mm\*高 2mm。它可以容易的用于 SMT 工艺制造且非常适于平板 PC、超级本、移动设备和消费产品。它还具有一个 PCM 接口，可以通过 BT 控制器直接连接外部音频编解码器传输音频数据。理论上，使用 802.11n/g/n MIMO 技术的 WIFI 吞吐量可以高达 150Mbps。蓝牙支持 BT2.1+EDR/ BT3.0 和 BT4.0。

### 3.3.1 WIFI station 模式

通常，我们使用板载 WIFI 连接 WIFI AP 热点。这就是 WIFI station 模式。

**第 1 步：加载 WIFI 驱动。**

在 LeMaker Guitar 上，操作系统会自动默认加载 WIFI 模块的驱动 wlan\_8723bs。可以用以下命令查看是否加载了驱动：

```
lsmod
Module                Size  Used by
...
wlan_8723bs           1327715  0
...
```

如果没有发现 WIFI 的驱动，则要手动加载：

```
sudo modprobe wlan_8723bs
```

如果想要开机自动加载 WIFI 驱动，则在 `/etc/modules` 文件中添加以下内容：

```
sudo nano /etc/modules
wlan_8723bs
```

**第 2 步：配置 WIFI。**

#### Lumuntu

在 Lemuntu 系统下，我们可以通过 `wpa_gui` 这个图形化界面连接 WIFI。如下图：

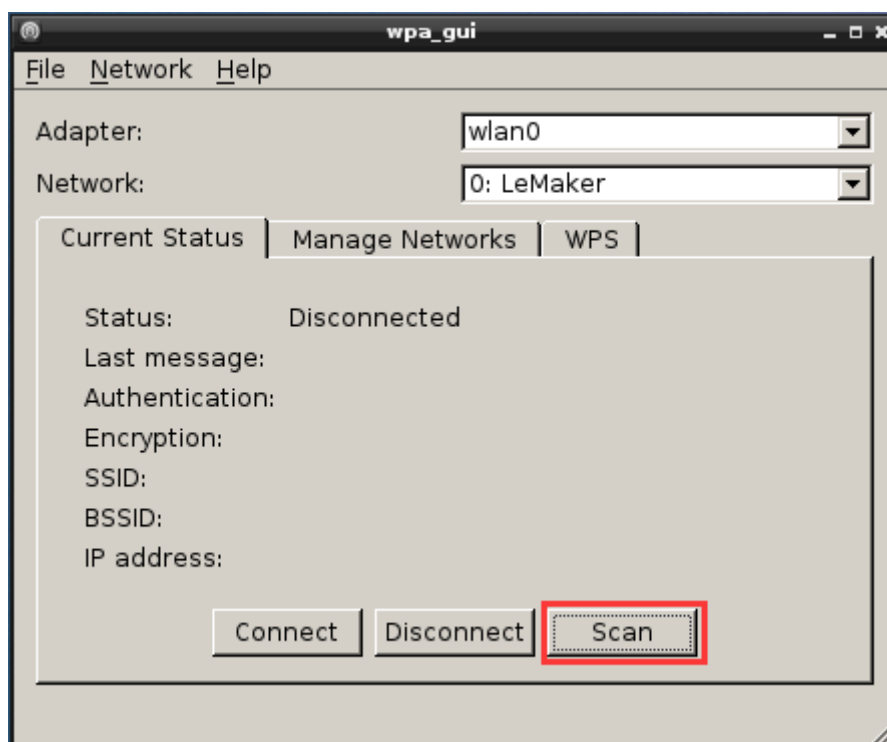


图 3.4 打开 wap\_gui

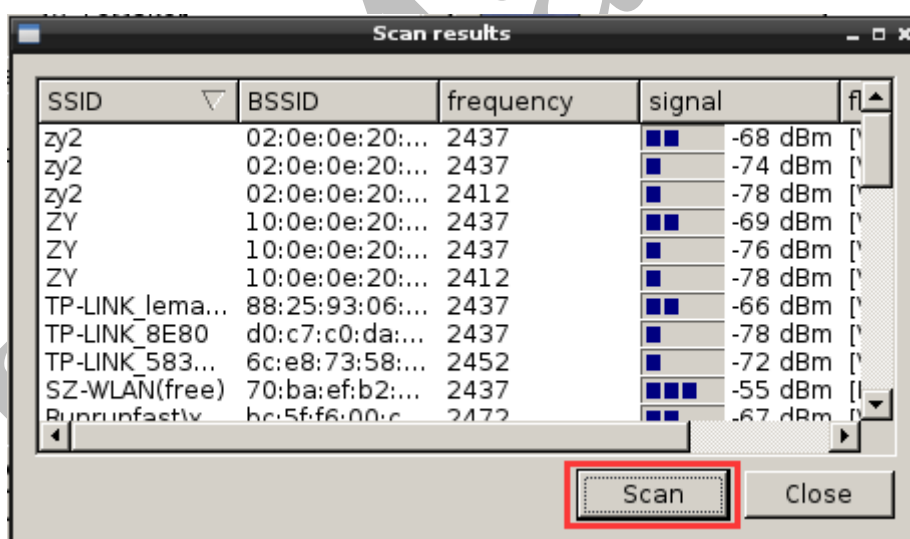


图 3.5 搜索 WIFIAP

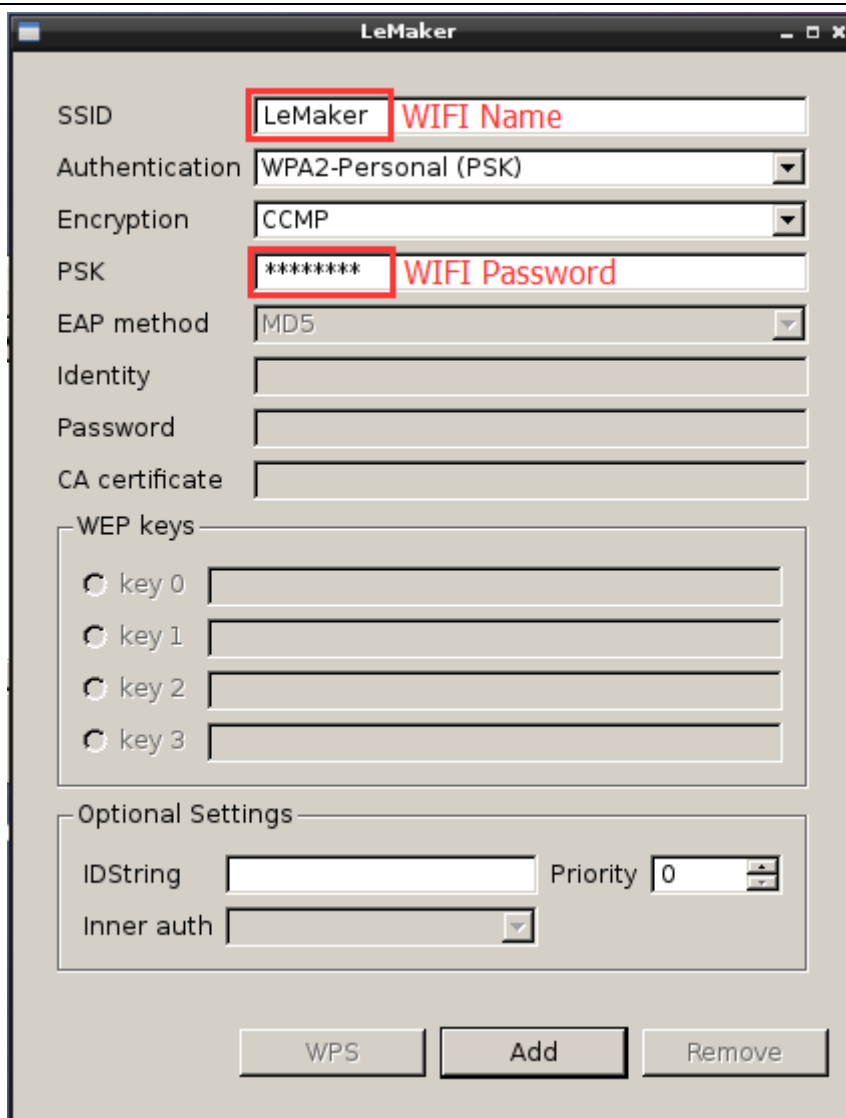


图 3.6 输入 SSID 和 PSK

配置文件保存在/etc/wpa\_supplicant/wpa\_supplicant.conf中。

```
cat /etc/wpa_supplicant/wpa_supplicant.conf
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="LeMaker"           #WIFI Name
    psk="***** "           #WIFI password
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}
```

## Ubuntu MATE

在 Ubuntu MATE 系统下，我们可以通过按右上角的网络连接按钮连接 WIFI。如下图：

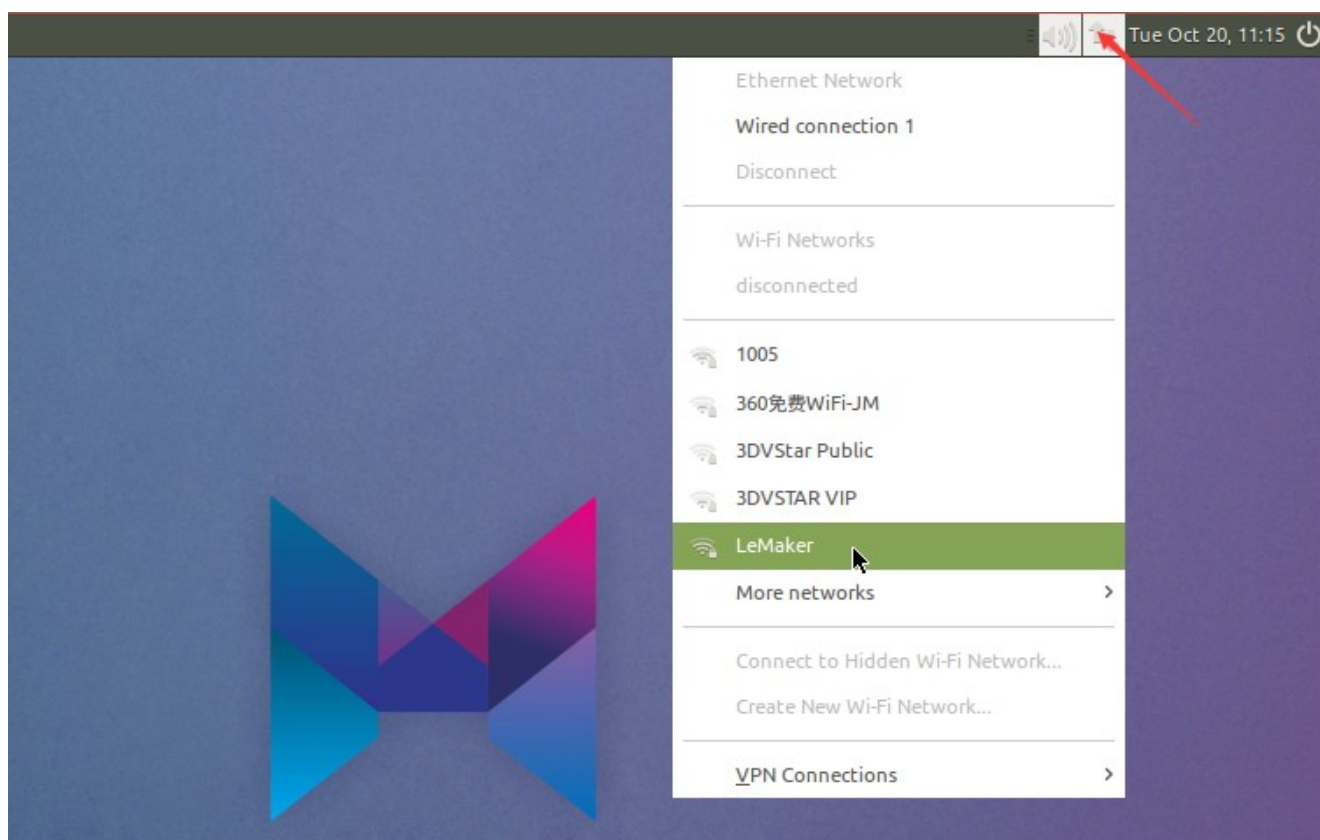


图 3.7 连接 WIFI

## 没有安装桌面环境的 Linux

如果我们在类 Linux 系统下使用命令行工作，可以参考以下内容连接 WIFI（用 wpa\_supplicant 工具）。

安装 wpa\_supplicant

```
sudo apt-get install wpasupplicant
```

编辑 wpa\_supplicant.conf，将以下内容添加到 wpa\_supplicant.conf 文件中。如果文件不存在，则手动创建它。ssid：指 WIFI 名称；psk：指 WIFI 密码。

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

ap_scan=1 # use the wpa_supplicant to scan and choose the AP
network={
    ssid="your_WIFI_name"
    psk="your_WIFI_password"
}
```

编辑/etc/network/interfaces 文件，参考以下内容进行配置：

```
sudo nano /etc/network/interfaces
auto wlanx
iface wlanx inet dhcp
pre-up wpa_supplicant -B -i wlanx -c/etc/wpa_supplicant
/wpa_supplicant.conf
pre-down killall -q wpa_supplicant
```

注意：wlanx 指 wlan0, wlan1 ...。我们可以输入 **ifconfig** 查看使用了哪个无线网口。

### 3.3.2 WIFI AP 模式

参考以下内容将 WIFI 配置为 AP 模式：

**第 1 步：**检查并加载 WIFI 驱动。

```
sudo modprobe wlan_8723bs
```

**第 2 步：**安装 hostapd 工具。

```
sudo apt-get install git #如果没有安装git, 先安装git

git clone git://github.com/tgraf/libnl-1.1-stable.git #安装libnl
cd libnl-1.1-stable/
./configure
make
sudo make install

sudo apt-get install libnl-3-dev
sudo apt-get install openssl
sudo apt-get install libssl-dev

git clone git://w1.fi/srv/git/hostap.git #安装hostap
cd hostap/hostapd/
cp defconfig .config
make
sudo make install
```

**第 3 步：**创建并编辑 hostapd.conf。

```
sudo mkdir -p /etc/hostapd
sudo nano /etc/hostapd/hostapd.conf
interface=wlanx
driver=nl80211
ssid=LeMaker_1020
```

```
channel=6
hw_mode=g
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

**第4步：**编辑/etc/network/interfaces。

```
sudo nano /etc/network/interfaces
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlanx
iface wlanx inet static
address 192.168.100.1
netmask 255.255.255.0
```

**第5步：**安装 DHCP 服务。

```
sudo apt-get install udhcpd
sudo nano /etc/udhcpd.conf
# The start and end of the IP lease block
start      192.168.100.20 #default: 192.168.0.20
end        192.168.100.254 #default: 192.168.0.254
# The interface that udhcpd will use
interface   wlanx          #default: eth0
#Examples
option subnet 255.255.255.0
opt  router 192.168.100.1
opt  wins   192.168.100.1
option dns  192.168.100.1 # appened to above DNS servers for a total of 3
option domain local
option lease 864000      # 10 days of seconds
```

**第6步：**重启系统，启动 dhcp 和 hostapd。

```
sudo reboot
udhcp /etc/udhcpd.conf
hostapd -B /etc/hostapd/hostapd.conf
```

**注意：**wlanx 指 wlan0,wlan1 ...

现在，我们可以连接 LeMaker Guitar 提供的 WIFI AP 热点了。但是我们还不能连接到因特网上。如果要连接上因特网，我们需要在 `ethx` 和 `wlanx` 之间创建一个网桥。

### 3.4 蓝牙

LeMaker Guitar 使用的 RTL8723BS 模块支持 WIFI 和蓝牙功能，下面的教程将在 Lemuntu 系统上教我们如何使用蓝牙。

**第 1 步：加载蓝牙驱动。**

```
sudo modprobe rfkill-actions_8723bs.ko
```

**第 2 步：安装相关软件。**

```
sudo apt-get install bluetooth bluez-cups python-bluez gnome-phone-manager
```

**第 3 步：下载并安装 rtl8723bs\_bt。**

```
sudo wget http://mirror.lemaker.org/rtl8723bs_bt.tar.gz
tar -zxvf rtl8723bs_bt.tar.gz
cd rtl8723bs_bt
make
make install
```

**第 4 步：启动蓝牙。**

```
sudo chmod +x start_bt.sh
sudo ./start_bt.sh
```

**第 5 步：查看并使用蓝牙。**

```
hcitool dev
Devices:
    hci0    8D:18:D9:F3:79:5D

hciconfig -a
hci0:   Type: BR/EDR  Bus: UART
        BD Address: 8D:18:D9:F3:79:5D  ACL MTU: 1021:8  SCO MTU: 255:16
        UP RUNNING PSCAN ISCAN
        RX bytes:1179 acl:0 sco:0 events:36 errors:0
        TX bytes:2052 acl:0 sco:0 commands:36 errors:0
        Features: 0xff 0xff 0xff 0xfe 0xdb 0xff 0x7b 0x87
        Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
        Link policy: RSWITCH HOLD SNIFF PARK
        Link mode: SLAVE ACCEPT
        Name: '*****'
```

```
Class: 0x000000
Service Classes: Unspecified
Device Class: Miscellaneous,
HCI Version: 4.0 (0x6) Revision: 0x1e3e
LMP Version: 4.0 (0x6) Subversion: 0xe40e
Manufacturer: Realtek Semiconductor Corporation (93)
```

```
hciconfig hci0 up
```

我们可以通过以下命令扫描周围的蓝牙设备,也可以用其它的蓝牙设备扫描到 LeMaker Guitar。

```
hcitool scan
Scanning ...
    88:C9:D0:0A:B6:8B      *
    28:E3:47:94:09:4A      *
```

**第 6 步: 安装 Bluetooth GUI 并连接 Bluetooth 设备。**

```
apt-get install blueman gnome-bluetooth
```

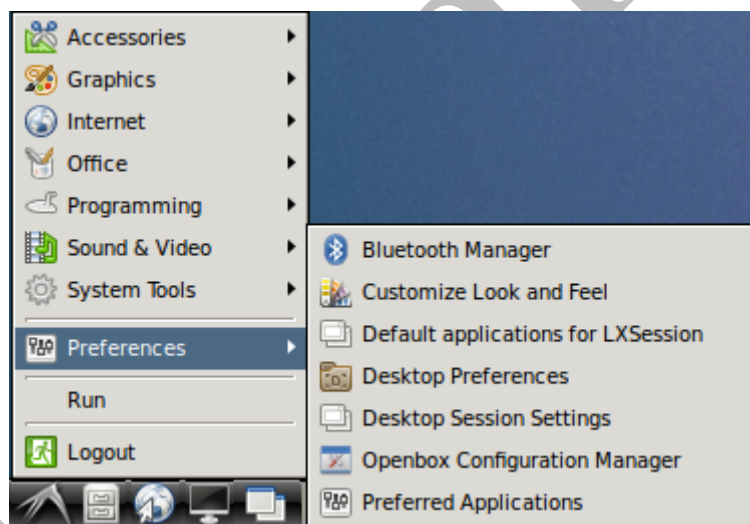


图 3.8 打开 Bluetooth 管理器

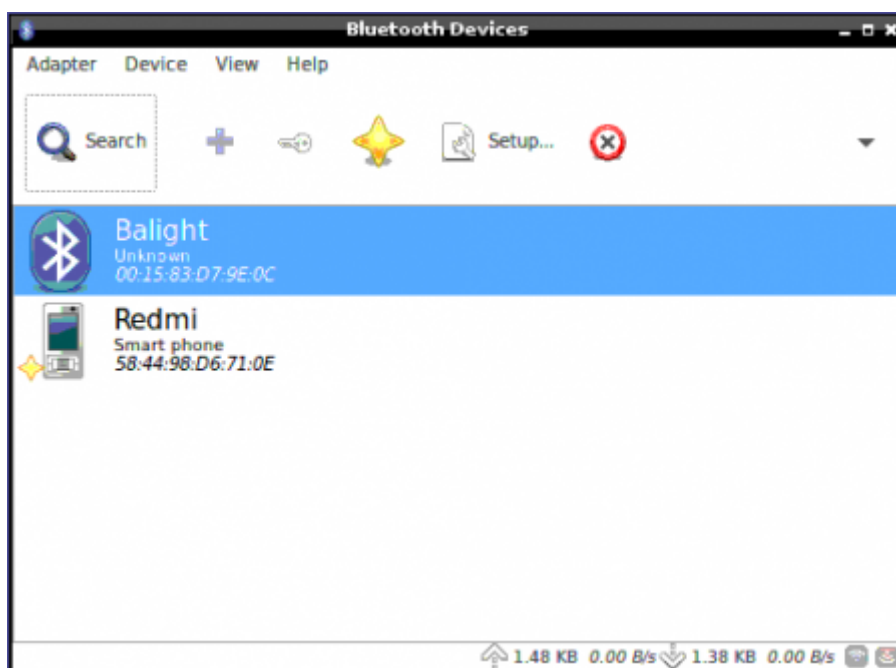


图 3.9 搜索蓝牙设备

## 3.5 摄像头

LeMaker Guitar 的摄像头接口兼容 Banana Pi/Pro，所以在 LeMaker Guitar 上我们也能够使用 Banana Pi Camera 模块，它使用了 OmniVision 公司的 5640 CMOS 影像感光技术的高分辨率摄像头。摄像头模块连接在 LeMaker Guitar 底板上的 CSI 接口上。它提供了在一个小而轻的高灵敏度，低串扰，低噪点图像捕捉的设计方案。

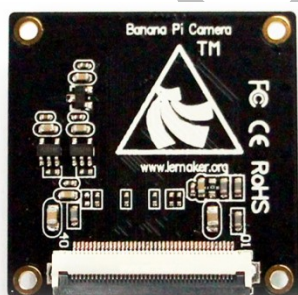


图3.10 Banana Pi 摄像头模块



图3.11 40pin FPC线

**第 1 步：** 安装摄像头模块所需要的驱动模块。

```
sudo modprobe owl_camera
sudo modprobe ov5640 rear=1
```

**第 2 步：** 检查摄像头模块驱动。

LeMaker Guitar 用户手册

可以用 `lsmod` 命令检查摄像头模块驱动是否成功加载。

```
lsmod
Module                Size  Used by
...
ov5640                34118  1
...
```

如果摄像头驱动已经成功装载进去，我们会发现“`/dev/video0`”节点。

```
ls /dev/video0
/dev/video0
```

### 第 3 步：安装 `mplayer`。

MPlayer 是一个可以在多个操作系统上运行的视频播放器，它可以播放 MPEG/VOB、AVI、Ogg/OGM、VIVO、ASF/WMA/WMV、QT/MOV/MP4、RealMedia、Matroska、NUT、NuppelVideo、FLI、YUV4MPEG、ILM、RoQ、PVA 格式文件。它支持本地的 XAnim 和 Win32 DLL 多媒体数字信号编解码器，可以观看 VideoCD、SVCD、DVD、3ivx、DivX 3/4/5、WMV 甚至 H.264 视频。

`guvcview` 是一个简单的 V4L2 全功能的视频采集器。

```
sudo apt-get install mplayer
```

### 第 4 步：使用 MPlayer 测试摄像头。

在 LeMaker Guitar 桌面上的命令终端上输入下面命令显示摄像头视频图像。

```
mplayer tv:// -fs
```

我们也可以使用一个 UVC 摄像头通过上述步骤来来显示摄像头视频图像。但是并非所有 UVC 摄像头可以正常工作。目前，LeMaker Guitar 上的 USB2.0 支持每个 microframe 有 3 个 transaction 或最后一个 transaction 很短的 microframe 的 UVC 摄像头，不支持每个 microframe 有 1 个 transaction 或 2transaction 的 UVC 摄像头。

## 3.6 存储

### 3.6.1 USB

LeMaker Guitar 底板 Rev.B 上有一个 USB 3.0 Micro-B 接口，它可以工作在 device 模式，也可以工作在 host 模式。

## Device 模式

当我们要把一个操作系统烧写到 LeMaker Guitar 的 EMMC 上时，或者要在 LeMaker Guitar 上对 Android App 进行调试时，需要把 LeMaker Guitar 连接到电脑上，对于电脑而言，LeMaker Guitar 是个外部设备，此时，USB3.0 工作在 Device 模式。但是目前 LeMaker Guitar USB3.0 的 Device 模式驱动还不够完善，所以当我们在使用 Device 模式的时候，它实际上是工作在 USB2.0 下的。

我们也可以使用 USB2.0 数据线（与大多安卓手机数据线相同）连接 LeMaker Guitar 到电脑上。



图 3.12 USB2.0 Micro-B转  
Standard A 公口连接线



图 3.13 Guitar USB2.0  
Device模式

## Host 模式

Host 模式要比 Device 模式要更加常见，因为在这个模式下，我们可以用 USB3.0 接口连接硬盘或千兆网卡等。当然，如果需要这样做的话，我们还应该有一根 USB3.0 OTG 线。（USB 3.0 Micro B 公口转 USB 3.0 Standard A 母口连接线）如下图所示：



图 3.14 USB 3.0 Micro B 公口转 USB 3.0 Standard A 母口连接线

市场上对 USB 3.0 Micro B OTG 线的引脚定义有两种，所以我们在购买的时候一定要注意这一点。表格如下所示：

表 3.1 USB3.0 引脚定义

| Micro B Plug |       | Wire        | USB3.0 Standard A Receptacle |       |
|--------------|-------|-------------|------------------------------|-------|
| Pin Number   |       | Wire Number | Pin Number                   |       |
| 1            | VBus  | 1           | 1                            | VBUS  |
| 2            | D-    | 2           | 2                            | D-    |
| 3            | D+    | 3           | 3                            | D+    |
| 4            | ID    |             |                              |       |
| 5            | GND   | 4           | 4                            | GND   |
| 6            | SSTX- | 5           | 8                            | SSTX- |
| 7            | SSTX+ | 6           | 9                            | SSTX+ |
| 8            | GND   | 7           | 7                            | GND   |
| 9            | SSRX- | 8           | 5                            | SSRX- |
| 10           | SSRX+ | 9           | 6                            | SSRX+ |

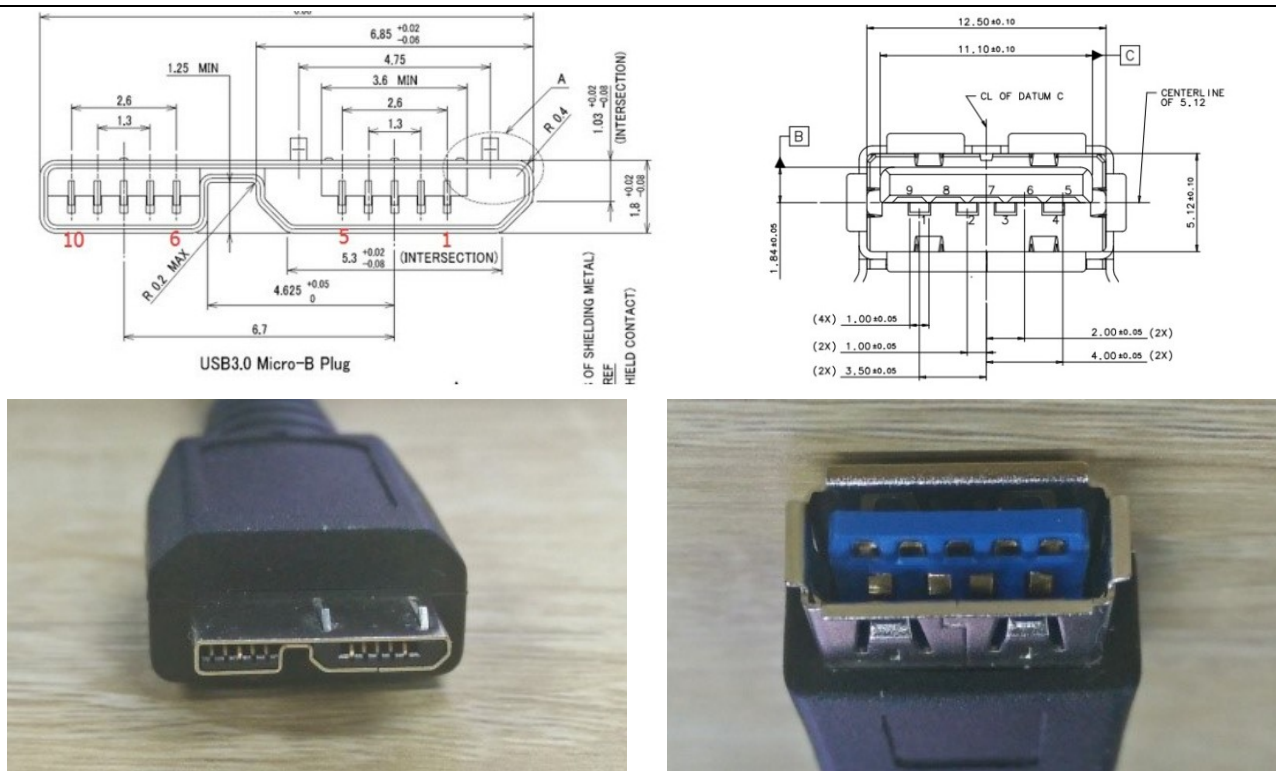


图 3.15 USB 3.0 插口

### 3.6.2 Micro SD 卡和 EMMC NAND Flash

当操作系统从 EMMC NAND Flash 启动，`/dev/mmcblk0` 是 EMMC NAND Flash 的设备节点。如果插入一张 MicroSD 卡到 MicroSD 卡插槽，我们将会发现 `/dev/mmcblk1` 变成 MicroSD 卡的设备节点。但是当操作系统从 MicroSD 启动时，`/dev/mmcblk0` 是 MicroSD 卡的设备节点，且 `/dev/mmcblk1` 是 EMMC NAND Flash 的设备节点。使用下面的命令查看系统所有的存储设备的分区：

```
sudo fdisk -l
```

或

```
lsblk
```

## 3.7 ADC

LeMaker Guitar 底板 Rev.B 上有两个 ADC 通道：ADC 和 ADC\_COM。



图 3.16 ADC 接口

输入下面的命令查看 ADC 的值:

#### ADC0:

```
cat /sys/class/hwmon/hwmon0/device/aux0
185 /1024      #ADC0_DATA = 185
```

可以使用下面的公式计算 ADC0 (AnalogVol)的电压值:

$$\text{AnalogVol} = 3.0/1024 * (\text{ADC0\_DATA})$$

#### ADC\_COM:

```
cat /sys/class/hwmon/hwmon0/device/remote_control
99 /1024      #ADC_COM_DATA = 185
```

可以使用下面的公式计算 ADC\_COM (AnalogVol)的电压值:

$$\text{alogVol} = \text{SVCC}/1024 * (\text{REM\_CON\_DATA})$$

其中, SVCC 的默认值是 3.1V。

## 3.8 另请参阅

- [1] hostapd – 一个用户态用于 AP 和认证服务器的守护进程: <http://w1.fi/hostapd/>
- [2] wpa\_supplicant – 一个无线网路配置工具: [http://w1.fi/wpa\\_supplicant/](http://w1.fi/wpa_supplicant/)
- [3] Netlink Protocol Library Suite: <http://www.infradead.org/~tgr/libnl/>
- [4] MPlayer – 电影视频播放器: <http://www.mplayerhq.hu/design7/info.html>
- [5] guvcview – 一个简单的 V4L2 视频捕捉工具: <http://sourceforge.net/projects/guvcview/>
- [6] OV5640: <http://www.ovt.com/products/sensor.php?id=177>

## 4 扩展根文件系统分区

LeMaker 提供的 Linux 系统镜像文件(.img 文件或.fw 文件)大约有 3.5GB 大小。当把系统镜像文件烧写到 8GB 的 MicroSD 卡或 EMMC NAND Flash 后,会发现 “/” 目录所在的系统分区只剩余大约 3GB 大小的可用空间。由于这些存储空间会很快被用完,这可能会引起麻烦。例如安装一个 emacs 软件就要占用 200M 的存储空间。

因此我们需要扩展根目录系统分区。例如,有一个 8GB 的 MicroSD 卡或 8GB 的 EMMC NAND Flash,我们想要把剩余的存在空间扩展到根目录所在的分区。扩张系统分区我们需要用到两个工具 parted 和 resize2fs。

**parted:** 是一个硬盘分区和分区扩展工具,它允许我们创建、撤销、扩展、移动和复制 ext2、linux-swaps、FAT、FAT32 和 reiserfs 分区。它能够创建、扩展和移动 Macintosh HFS 分区,也能够检测 jfs、ntfs、ufs 和 xfs 分区。它对于新系统创建存储空间,重新组织硬盘空间使用和复制数据到新硬盘非常有用。

**resize2fs:** 能扩展 ext2、ext3 或 ext4 文件系统。他能够扩大或缩小设备上未挂载的文件系统。如果文件系统已经挂载,也能够扩展已挂载的文件系统,只要内核支持在线扩展分区。

执行下面的操作扩展系统分区:

```
parted /dev/mmcblk0 #mmcblkx 表示 MicroSD 卡或 EMMC NAND Flash
GNU Parted 3.2
Using /dev/mmcblk0
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print #查看分区
Model: SD SL08G (sd/mmc)
Disk /dev/mmcblk0: 7948MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
  1      8389kB  58.7MB  50.3MB  fat16        primary  msftdata
  2      58.7MB  3670MB  3611MB  ext4         primary  msftdata

(parted) resizepart #扩展分区
Partition number? 2 #选第二个分区
End? [3670MB]? -1 #扩展到最后一行
```

```
(parted) print #查看分区
```

```
Model: SD SL08G (sd/mmc)
```

```
Disk /dev/mmcblk0: 7948MB
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt
```

```
Disk Flags:
```

| Number | Start  | End    | Size   | File system | Name    | Flags    |
|--------|--------|--------|--------|-------------|---------|----------|
| 1      | 8389kB | 58.7MB | 50.3MB | fat16       | primary | msftdata |
| 2      | 58.7MB | 7947MB | 7888MB | ext4        | primary | msftdata |

```
(parted) q #退出 parted
```

```
Information: You may need to update /etc/fstab.
```

然后:

```
sudo reboot
```

```
sudo resize2fs /dev/mmcblk0p2
```

(注意: parted 2.3 不支持动态分区。)

## 5 CPU

### 5.1 调整 CPU 主频

根据不同的需求，LeMaker Guitar 能够调整 CPU 频率和调频模式，使之性能高且功耗低，这是非常重要的。时钟频率越低，设备运行速度越慢，功耗越低。CPU 内核电压值依赖时钟频率，时钟频率和电压值之间存在一种映射关系，这种关系被称为 `defs_table`。仅定义一个 CPU 频率/电压映射集合和最大/最小值是不够的，还需要在它们之间切换的管理策略。`cpufreq` 管理器（`cpufreq governors`）负责管理这些策略。LeMaker Guitar 支持 Linux 内核 CPU 内核调频，我们可以使用如下命令查看所有调频管理模式和 `dvfs_table` 映射表。

查看可用主频：

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
408000 720000 900000 1104000 1308000
```

查看当前主频：

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
900000
```

查看可用调频模式：

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
conservative ondemand userspace powersave interactive performance
```

**ondemand:** CPU 根据当前负荷设置 CPU 频率。为了达到这个目的，CPU 必须能够快速切换频率。

**Conservative:** “conservative” 模式非常类似于 “ondemand”，根据 CPU 的当前负荷设置 CPU 的频率，最大的不同在于当 CPU 负荷发生变化时，CPU 的频率逐步的提升或降低，而不是直接跳到最高值。这种模式适合于电池供电情况下。

**Userspace:** “userspace” 模式允许用户和任何以 root 权限运行的程序，通过创建一个 “scaling\_setspeed” 的 sysfs 文件来设定 CPU 频率。

**Powersave:** “powersave” 模式在最小频率和最大频率范围内，设置 CPU 的频率为最小值。

**Performance:** “performance” 模式在最小频率和最大频率范围内，设置 CPU 的频率为最大值。

## 5.2 锁频

调频模式和 CPU 频率的范围都会对电量消耗和性能产生重要的影响。可以通过下面方式锁定 CPU 频率：

```
echo ondemand > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo 900000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 900000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo 35 > /sys/devices/system/cpu/cpufreq/ondemand/up_threshold
echo 10 > /sys/devices/system/cpu/cpufreq/ondemand/sampling_down_factor
echo 1 > /sys/devices/system/cpu/cpufreq/ondemand/io_is_busy
```

查看设置结果：

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
900000
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
900000
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
900000
cat /sys/devices/system/cpu/cpufreq/ondemand/up_threshold
35
cat /sys/devices/system/cpu/cpufreq/ondemand/sampling_down_factor
10
cat /sys/devices/system/cpu/cpufreq/ondemand/io_is_busy
1
```

CPU 调频配置默认存储在 `/etc/rc.local` 中

```
cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

echo ondemand > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor

echo 45 > /sys/devices/system/cpu/cpufreq/ondemand/up_threshold
echo 15 > /sys/devices/system/cpu/cpufreq/ondemand/sampling_down_factor
echo 1 > /sys/devices/system/cpu/cpufreq/ondemand/io_is_busy
```

```
exit 0
```

## 5.3 读取 CPU&PMU 内部温度

有两种方法读取 CPU 的内部温度，一种方法是直接读取 `/sys/class/thermal/thermal_zone1/temp` 的值，另一种方法是使用 `lm-sensors` 工具。我们也可以直接读取 PMU（电源管理单元）的内部温度。

### 直接读取 CPU 的内部温度

```
cat /sys/class/thermal/thermal_zone1/temp
```

### 使用 `lm-sensors` 读取 CPU 的内部温度

第 1 步：安装 `lm-sensors` 工具。

```
sudo apt-get update
sudo apt-get install lm-sensors
```

第 2 步：监测硬件。

```
sudo sensors-detect
```

第 3 步：读取 CPU 内部温度。

```
sudo sensors
battery-virtual-0
Adapter: Virtual device
temp1:          +0.0°C

owl-thermal-virtual-0
Adapter: Virtual device
temp1:          +51.0°C (crit = +105.0°C)
```

### 读取 PMU 的内部温度

```
cat /sys/devices/b0170000.i2c/i2c-0/0-0065/atc2603c-hwmon.0/ic_temperature
```

## 5.4 另请参阅

[1] Linux(TM)内核 CPU 频率电压调节代码：

<https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt>。

[2] PowerVR 介绍: [https://en.wikipedia.org/wiki/PowerVR#Series\\_5\\_28SGX.29](https://en.wikipedia.org/wiki/PowerVR#Series_5_28SGX.29)。

[3] lm-sensor-一个 Linux CPU 监控软件工具: <http://www.lm-sensors.org/>。

LeMaker

## 6 远程登录

### 6.1 使用 SSH 登录

Secure Shell (即 SSH), 是一种在远程机器上以安全方式启动的基于文本的 shell 会话的加密的网络协议。它允许用户在不接近实际的机器情况下, 在这个台机器的命令提示符下运行命令。它也允许用户在不安全的客户端-服务端的网络结构上建立一个安全通道, 连接 SSH 客户端应用程序和 SSH 服务器。常用应用包括远程命令行登录和远程命令执行, 任何网络服务都可以使用 SSH 来确保其安全。这个网络协议有两个规格, 分别是 SSH-1 和 SSH-2。

#### 6.1.1 Windows 上远程登录

**第 1 步:** Windows 上安装 Putty 或 SecureCRT。

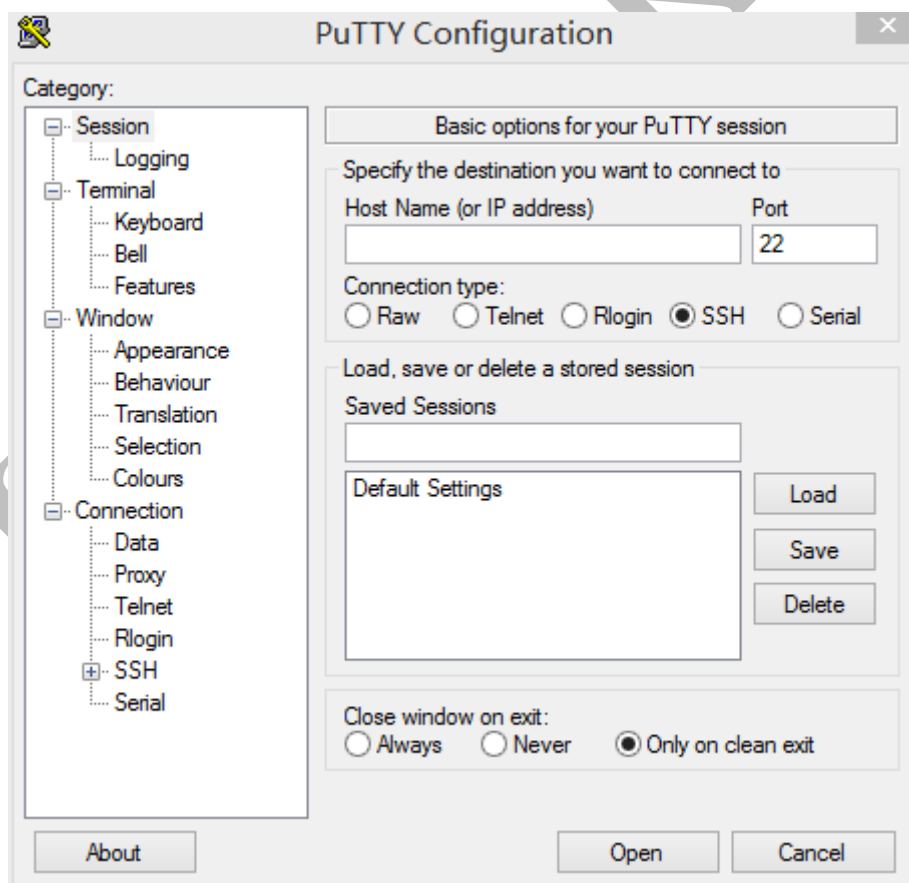


图 6.1 Putty

**第 2 步:** 通过串口命令行终端查看 SBC 的 IP 地址, 例如: 192.168.0.97。确保我们的桌面 PC

机和 LeMaker Guitar 在同一个局域网中。

`ifconfig`

**第 3 步：**选择“SSH”选项，设置 Host Name 为 SBC 的 IP 地址。

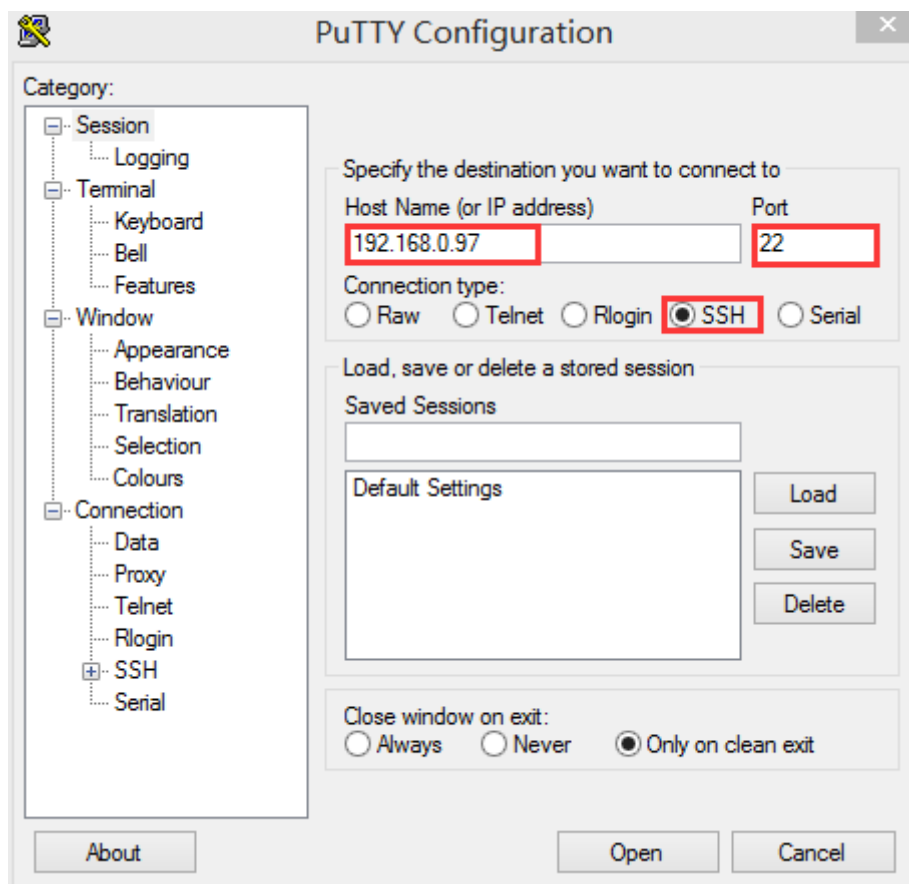


图 6.2 配置 Putty

**第 4 步：**点击“Open”按钮，远程登录到 SBC。

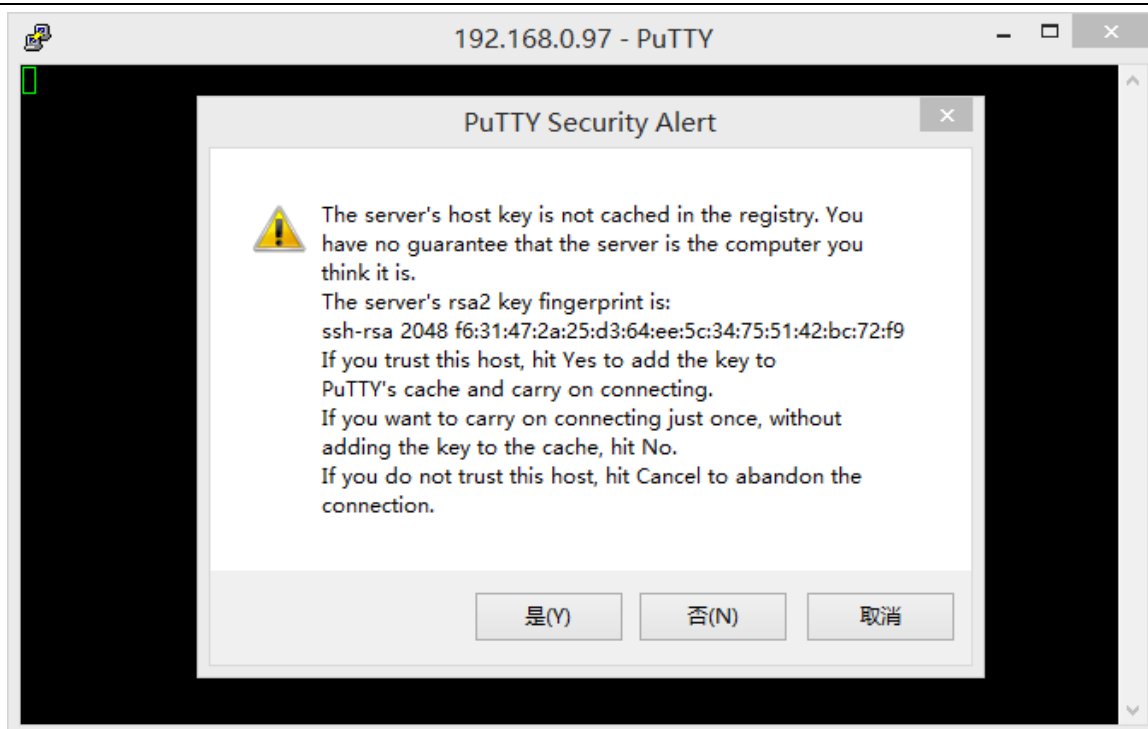


图 6.3 远程连接 LeMaker Guitar

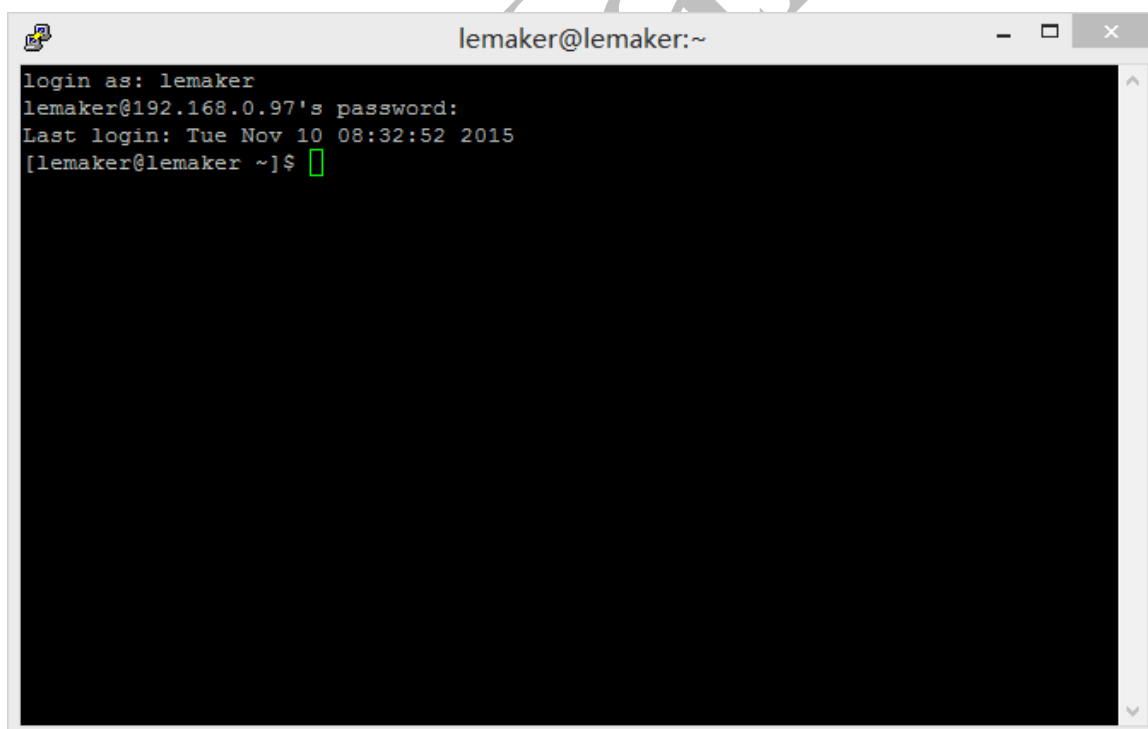


图 6.4 远程登录 SBC

## 6.1.2 Linux 下远程登录

我们也可以在 Linux 桌面 PC 机上登录 SBC。

**第 1 步：**在桌面 Linux 系统(ubuntu 12.04)上安装 Openssh 软件。

```
sudo apt-get install openssh
```

openssh-client, openssh-server 和 ssh-import-id 将会被安装。

**第 2 步：**通过 SSH 登录 SBC。

```
sudo ssh -l [username] [ipaddress]
```

username 是用户名(lemaker), ipaddress 是 SBC 的 IP 地址。

## 6.2 使用 vnc 远程控制 SBC

VNC 即虚拟网络计算机, 是一个远程控制软件, 它允许我们在互联网上的任何地方查看和远程管理另外一台计算机的桌面 (“VNC 服务器”), 只需使用另一台电脑桌面上的一个简单的客户端程序 (“VNC 查看器”)。而这两台计算机不必是同样类型的计算机, 支持跨平台进行一切相关的操作。例如, 我们可以使用 “VNC 查看器” 在家里的 Windows 电脑桌面上查看办公室里的 Linux 电脑。VNC 是免费的, 广泛应用于工业, 教育和个人用途。VNC 使与远程计算机交互变得更加简单。

**TightVNC** 是一个免费软件工具。通过 **TightVNC**, 我们可以查看远程计算机桌面, 并且使用鼠标和键盘控制这个计算机, 就像我们坐在一台计算机前面所进行的操作一样。

**第 1 步：**安装 tightvncserver。

```
sudo apt-get update  
sudo apt-get install tightvncserver
```

**第 2 步：**启动 vncserver。

```
vncserver :1
```

如果有停止 vncserver, 输入下面的命令:

```
vncserver -kill :1
```

**第 3 步：**使用下面命令, 查看 LeMaker Guitar 的 IP 地址。

```
ifconfig
```

**第 4 步：**使用 vncview 远程控制 LeMaker Guitar。

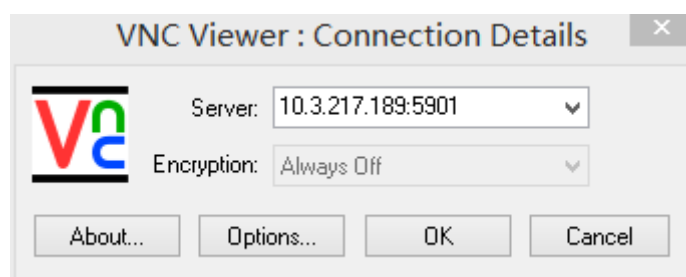


图 6.6 配置 VNC Viewer

其中 10.3.217.189 是 LeMaker Guitar 的 IP 地址；5901 是 tightvncserver 的默认端口号。

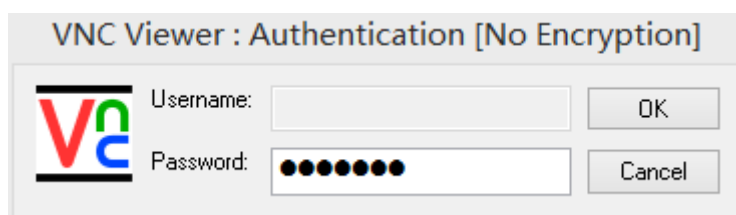


图 6.7 输入密码

输入密码，我们将会看到 LeMaker Guitar 的桌面。

## 6.3 使用 RDP 登录

RDP 是微软开发的一套专业远程桌面协议，它提供了一个图形用户界面，通过网络连接到另外一个计算机上。用户使用 RDP 客户端软件就可达远程链接目的，但是运行其它操作系统的计算机必须运行 RDP 服务器软件。

**第 1 步：**在 Lemuntu 上安装 RDP 服务。

```
sudo apt-get install xrdp
sudo apt-get install tightvncserver
sudo systemctl enable xrdp
sudo reboot
```

**第 2 步：**打开“run”命令窗口，输入“mstsc”，点击“OK”按钮。

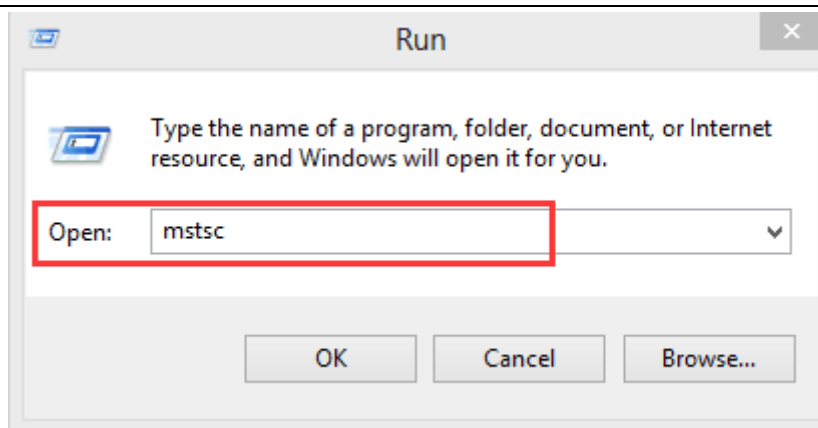


图 6.8 Run 命令窗口

**第 3 步：**输入 LeMaker Guitar 的 IP 地址， 点击“connect”按钮。

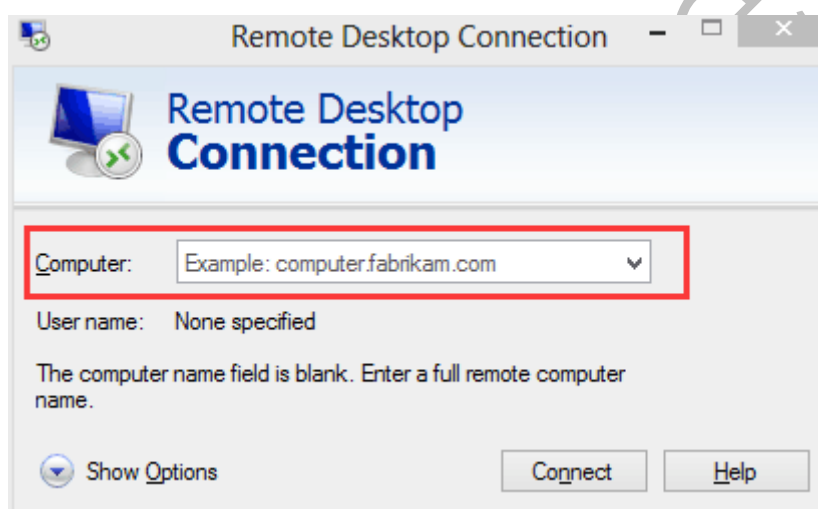


图 6.9 远程桌面连接

**第 4：**输入登录 Lemuntu 的账户。

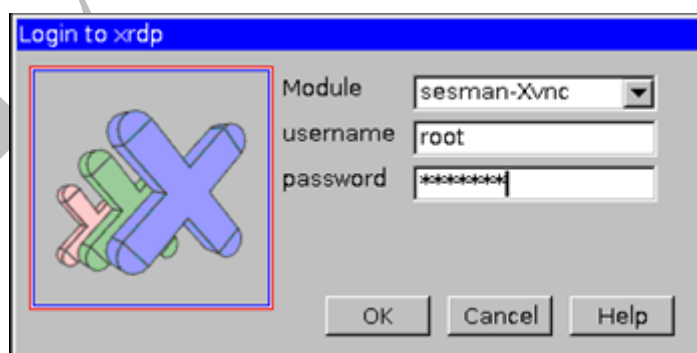


图 6.9 登录到 xrdp

**第 5 步：**点击“OK”，我们将看到连接成功的提示。

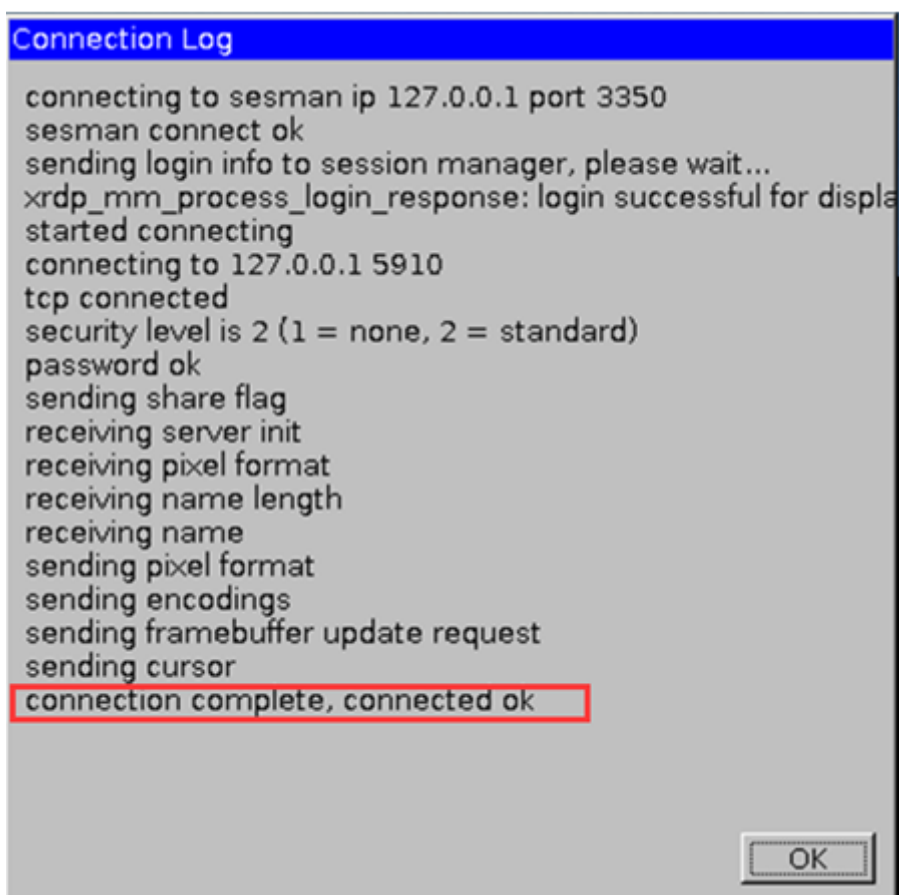


图 6.10 查看 rdp 连接成功日志

## 6.4 另请参阅

- [1] PuTTY 下载页面: <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>。
- [2] SSH – 加密网络安全登录协议: [https://en.wikipedia.org/wiki/Secure\\_Shell](https://en.wikipedia.org/wiki/Secure_Shell)。
- [3] SecureCRT – 安全的 Telnet 和 SSH Windows 客户端:  
<https://www.vandyke.com/products/securecrt/>。
- [4] Http File Server 主页: <http://www.rejetto.com/hfs/>。
- [5] tightvnc 主页: <http://www.tightvnc.com/>。

## 7 视频

### 7.1 GPU 硬件加速

PowerVR 是 Imagination Technologies 公司(原 VideoLogic)开发硬件、2D 和 3D 渲染软件、视频编解码库、图像处理技术、OpenGL ES 及 OpenCL 加速库的一个部门。PowerVR SGX544MP GPU 提供最好的游戏体验和全高清显示屏, 支持行业标准 API-OpenGL-ES1.1 和 2.0。

#### 纹理支持:

- 立方体映射
- 投影纹理
- 非平方纹理
- 体积纹理
- 结构数组

#### 纹理格式:

- RGB 8888, 565, 1555, 1565
- 单色 8, 16, 16f, 32f, 32int
- 双通道 8:8, 16:16, 16f:16f
- 压缩纹理 PVR-TC1, PVR-TC2, ETC1
- 可支持对所有 YUV 格式编程

#### 分辨率支持:

- 帧缓冲最大值 = 4096\*4096
- 纹理最大值 = 4096\*4096
- 最大体积 = 2048
- 纹理坐标范围 = 8192

#### 7.1.1 在 LeMaker Guitar 上安装硬件加速库

**第 1 步:** 下载并解压文档。

```
wget http://mirror.lemaker.org/GPU_For_LeMaker_Guitar_201511.tar.gz
sudo tar -zxvf GPU_For_LeMaker_Guitar_201511.tar.gz
```

## 第 2 步：安装软件包。

```
cd GPU_For_LeMaker_Guitar_201511
sudo chown -R root:root ./*
sudo ./install.sh
```

## 第 3 步：替换 X，Xorg。

```
cd /usr/bin
sudo mv X X.original
sudo mv Xorg Xorg.original
sudo ln -sf /usr/local/XSGX/bin/X X
sudo ln -sf /usr/local/XSGX/bin/Xorg Xorg
sudo cp /usr/local/XSGX/etc/xorg.conf /etc/
```

## 第 4 步：检测模块相依性。

```
sudo depmod
```

## 第 5 步：加载库路径。

```
sudo nano /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf
/usr/local/XSGX/lib
/usr/lib

sudo ldconfig
```

## 第 6 步：检查 GPU 是否开启，并重启系统。

```
lsmod
Module                Size  Used by
...
pvrsrvkm              336156  6
drm                   225379  6 pvrsrvkm
...
```

确保已经加载了驱动 `pvrsrvkm` 和 `drm`，而且其“Used by”选项的值不为 0。另外，图形用户界面也运行正常，则可以说明硬件加速已经正常开启了。

## 7.1.2 测试 OpenGL ES

### 第 1 步：安装一些必要的库。

```
sudo apt-get install libgegl-0.2-0 libgfortran3 libgimp2.0 libgl1-mesa-dev
libgles2-mesa-dev libglew-dev libglew1.10 libglib2.0-bin libglib2.0-data
libglib2.0-dev libglu1-mesa-dev
```

```
sudo apt-get install libpng-dev
sudo apt-get install libx11-dev
sudo apt-get install pkg-config
sudo apt-get install libjpeg-dev
```

第 2 步：编译并安装 glmark2。

```
sudo git clone https://github.com/LeMaker/glmark2.git
cd glmark2
sudo ./waf configure --with-flavors=x11-gl,x11-glesv2
sudo ./waf
sudo ./waf install
```

第 3 步：替换一些库文件。

```
sudo cp /usr/lib/libEGL.so* /usr/lib/arm-linux-gnueabi/ -a
sudo cp /usr/lib/libGLES* /usr/lib/arm-linux-gnueabi/ -a
```

第 4 步：在 Lemuntu 系统命令行终端运行 glmark2-es2。

```
glmark2-es2
```

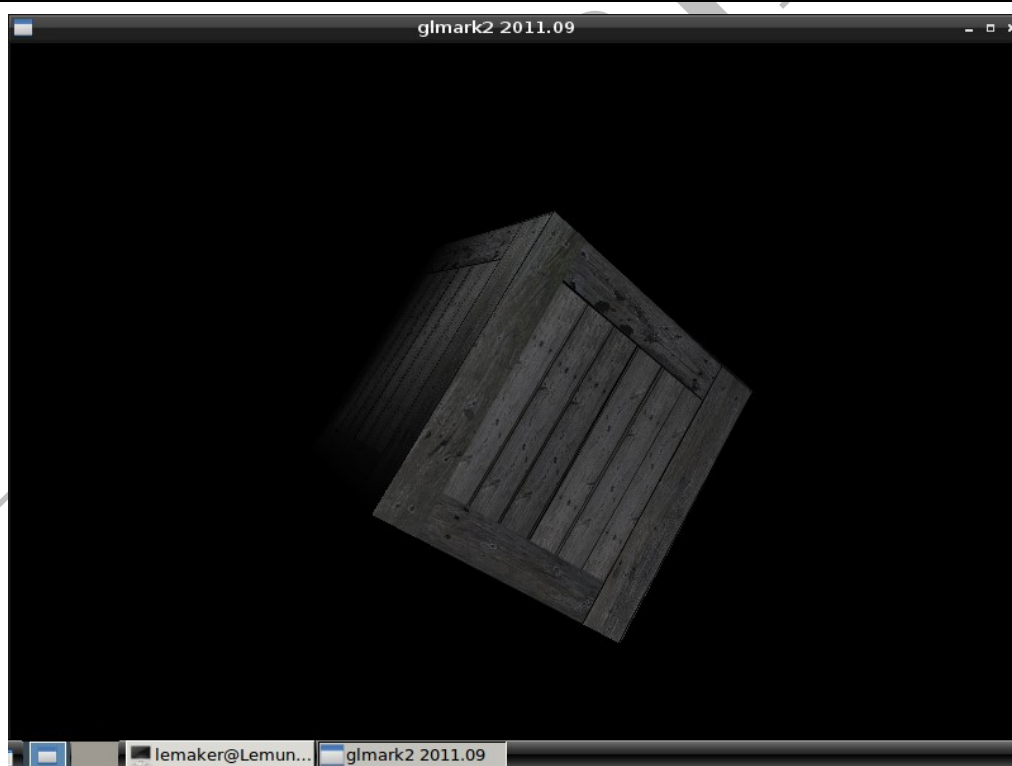


图 7.1 glmark 测试

### 7.1.3 测试 OpenCL

第 1 步：安装 OpenGL\_test 软件包。

```
sudo apt-get install unzip
wget http://mirror.lemaker.org/OpenCL_test.zip
unzip OpenCL_test.zip
```

第 2 步：运行 ocl\_filter\_test。

```
cd OpenCL_test/
cp *.so /usr/lib/
chmod 755 *_test
./ocl_filter_test
OpenCL Filter Test(s) (SGX_DDK_Linux_XOrg sgxddk MAIN@3625561,release,omap_linux) at
Tue Dec 8 10:16:31 2015
(00)
*****
****
Runs a 1x1 copyimage filter on an image using texture samples
*****
*****
copy1x1 --> passed
copy1x1: Test took 0.21 seconds (Compute: 0.01 seconds) to run 1 iterations:
copy1x1: Filter dims(1,1)      Image dims(512,512) size(1.00MB)      Pixelreads(0.26
million)
copy1x1: F/s(76.92)      s/F(0.01)      MB/s(76.92)      s/MB(0.01)      Pixelreads/s(20.09
million)
(01)
*****
****
Runs a 3x3 copyimage filter on an image using texture samples
*****
*****
copy3x3 --> passed
copy3x3: Test took 0.22 seconds (Compute: 0.01 seconds) to run 1 iterations:
copy3x3: Filter dims(3,3)      Image dims(512,512) size(1.00MB)      Pixelreads(2.33
million)
copy3x3: F/s(90.91)      s/F(0.01)      MB/s(90.91)      s/MB(0.01)
Pixelreads/s(211.98 million)
(02)
*****
****
Runs a 5x5 copyimage filter on an image using memory reads
*****
*****
...

```

### 第3步：运行 ocl\_unit\_test。

```
./ocl_unit_test
OpenCL Unit Test(s) (SGX_DDK_Linux_XOrg sgxddk MAIN@3625561,release,omap_linux)
at Tue Dec 8 10:20:02 2015
00*****
***
Platform Test:
    Checks that an OpenCL compatible platform is present
    for the unit test to run.
*****
***
Verify_Platform: Enumerating 1 platforms
Verify_Platform: CL_PLATFORM_PROFILE      EMBEDDED_PROFILE
Verify_Platform: CL_PLATFORM_VERSION      OpenCL 1.1
Verify_Platform: CL_PLATFORM_NAME         PowerVR SGX
Verify_Platform: CL_PLATFORM_VENDOR       Imagination Technologies
Verify_Platform: CL_PLATFORM_EXTENSIONS   'cl_khr_byte_addressable_store'
platform --> passed
platform: Test took 0.03 seconds to run:
platform: Verify 0.03s (100.00)%
01*****
***
Device Test:
    Checks that an OpenCL compatible device is present
    for the unit test to run.
*****
***
Verify_Device: Enumerating 1 devices
Verify_Device: CL_DEVICE_TYPE             CL_DEVICE_TYPE_GPU
Verify_Device: CL_DEVICE_NAME             PowerVR SGX 544
Verify_Device: CL_DEVICE_VENDOR          Imagination Technologies
Verify_Device: CL_DRIVER_VERSION          1.1
Verify_Device: CL_DEVICE_PROFILE          EMBEDDED_PROFILE
Verify_Device: CL_DEVICE_VERSION          OpenCL 1.1
Verify_Device: CL_DEVICE_EXTENSIONS       cl_khr_byte_addressable_store
cl_khr_global_int32_base_atomics cl_khr_global_int32_extended_atomics
cl_khr_local_int32_base_atomics cl_khr_local_int32_extended_atomics
device --> passed
device: Test took 0.00 seconds to run:
device: Verify 0.00s (100.00)%
```

```
02*****
***
Addition Kernel:
    Performs an online compilation of an integer addition kernel
    and verifies that the output buffer is correct.
*****
***
Compute_Add: Online compilation test with 4096 instances running source:
<source>
__kernel void AdditionKernel(__global int* a, __global int* b)
{
    int ith = get_global_id(0);
    a[ith] = a[ith] + b[ith];
}
...

```

## 7.2 视频解码

LeMaker Guitar VDE(Video Decoder Engine) 是基于 OpenMAX 的解码器组件。OpenMAX (Open Media Acceleration), 通常缩写为“OMX”, 是一个非专有、免费、跨平台的 C 语言编程接口, 提供用于音频、视频和静态图像处理的抽象例程。它主要用于那些需要对大量的多媒体数据(如视频编解码、图形库和其它的音视频、图像、语音的功能)以可预测方式进行高效的处理的低功耗和嵌入式设备(包括智能手机、游戏机、数字媒体播放器、机顶盒等)。OpenMAX 一共提供三层接口: 应用层(application layer: AL), 集成层(integration layer: IL) 和开发层(development layer: DL)。

OMX\_VDE 有两个端口: port1 为视频输入接口, port2 为视频输出接口。OMX\_VDE 支持许多视频格式, 例如: AVC(H264)、H263、MJPEG、MPEG2、MPEG4、WMV3 等。

LeMaker Guitar VDE (Video Decoder Engine)是一个基于 OpenMAX 的解码组件。

**第 1 步:** 下载并安装 omx lib。

```
sudo wget http://mirror.lemaker.org/omx.tar.gz
sudo tar -zxvf omx.tar.gz
cd omx
sudo ./install.sh

```

omx 头文件被安装在/usr/include/omx-include, omx 视频解码库安装在/usr/lib, 全部代码存放在/home/owlplayer。

## 第 2 步：编译 owlplayer。

```
cd /home/owlplayer  
make
```

## 第 3 步：测试解码库。

```
cd /home/owlplayer  
./owlplayer [videoname]
```

## 第 4 步：测试结果。

当执行 `./owlplayer videoname` 时，我们会在 `/home` 目录下发现 “NV12\_wxxx\_hxxx”。“NV12\_wxxx\_hxxx” 会记录 “videoname” 中的 50 帧图像。“NV12” 表示像素格式为 NV12。“xxx” 表示宽度或高度。我们可以在 Windows 电脑上运行 YUVPlayer.exe 播放 “NV12\_wxxx\_hxxx”。

## 7.3 另请参阅

[1] OpenMAX 介绍: <https://en.wikipedia.org/wiki/OpenMAX>。

[2] YUV 播放器-一个功能全面的播放器，用于播放未压缩的平面 YUV 视频文件：  
<http://www.yuvplayer.com/>。

## 8 IO 控制

LeMaker Guitar 底板 Rev.B 上有一个 40pin GPIO 接口，包括 GPIO，PWM、SPI、I2C、UART 等等。通过一些基本的方法，我们可以从用户空间控制这些 IO 的输入和输出。但是，如果我们需要更加复杂的接口通信（如 SPI 和 I2C），这是不容易实现的。因此，我们提供了可用的 GPIO 库：WiringLMK 和 LMK.GPIO。它们提供了许多的 API，让我们更加容易控制 SBC 上的 IO 接口。目前，WiringLMK C 库和 LMK.GPIO Python 库都支持 LeMaker Guitar。接下来，我们将学习如何使用物理管脚编号来使用这些库。

### 8.1 使用 sysfs 用户空间

我们可以从用户空间利用 sysfs 访问 GPIO 引脚。内核默认实现了物理 GPIO 到 sysfs 文件系统的 GPIO 操作映射。在 exporting 或 unexporting GPIO 之前，我们必须获得正确的引脚序号索引，LeMaker Guitar 底板 Rev.B 的计算方法如下：

|                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------|
| GPIO A: $\text{GPIOA} * \text{NUMBER} = 0 + * \text{NUMBER}*$ , for example, $\text{GPIOA}27 = 0 + 27 = 27$      |
| GPIO B: $\text{GPIOB} * \text{NUMBER} = 32 + * \text{NUMBER}*$ , for example, $\text{GPIOB}31 = 32 + 31 = 63$    |
| GPIO C: $\text{GPIOC} * \text{NUMBER} = 64 + * \text{NUMBER}*$ , for example, $\text{GPIOC}0 = 64 + 0 = 64$      |
| GPIO D: $\text{GPIOD} * \text{NUMBER} = 96 + * \text{NUMBER}*$ , for example, $\text{GPIOD}3 = 96 + 3 = 99$      |
| GPIO E: $\text{GPIOE} * \text{NUMBER} = 128 + * \text{NUMBER}*$ , for example, $\text{GPIOE}22 = 128 + 22 = 150$ |

我们可以简单的查看下面的“40Pin GPIO 定义”表：

## 40Pin GPIO definition

| Functions                   | Pin on S500 | Pin on Board | Pin on Board | Pin on S500 | Functions               |
|-----------------------------|-------------|--------------|--------------|-------------|-------------------------|
| VCC(3.3V)                   |             | 1            | 2            | 5V          |                         |
| TWI2_SDA                    | GPIOE3      | 3            | 4            | 5V          |                         |
| TWI2_SCK                    | GPIOE2      | 5            | 6            | GND         |                         |
| LVDS_OAP/LCD0_D19           | GPIOB18     | 7            | 8            | GPIOC27     | UART0_TX                |
| GND                         |             | 9            | 10           | GPIOC26     | UART0_RX                |
| DSI_DP3/SDIO0_CLKB/LCD0_D16 | GPIOC0      | 11           | 12           | GPIOB8      | PWM3                    |
| DSI_DN3/SDIO1_D3/LCD0_D9    | GPIOC1      | 13           | 14           | GND         |                         |
| DSI_CP/SDIO1_D1/LCD0_D1     | GPIOC4      | 15           | 16           | GPIOA25     | SIRQ1                   |
| VCC(3.3V)                   |             | 17           | 18           | GPIOC6      | MIPI_DSI_DP0            |
| SPI0_MOSI                   | GPIOC25     | 19           | 20           | GND         |                         |
| SPI0_MISO                   | GPIOC24     | 21           | 22           | GPIOC5      | DSI_CN/SDIO_D0/LCD0_D0/ |
| SPI0_SCLK                   | GPIOC22     | 23           | 24           | GPIOC23     | SPI0_SS                 |
| GND                         |             | 25           | 26           | GPIOB19     | LVDS_OAN/LCD0_D15       |
| LVDS_OBP/LCD0_D21           | GPIOB16     | 27           | 28           | GPIOB14     | LVDS_OBN/LCD0_D23       |
| LVDS_OCN/LCD0_D22           | GPIOB15     | 29           | 30           | GND         |                         |
| LVDS_OEP/LCD0_DCLK0         | GPIOB10     | 31           | 32           | GPIOB13     | LVDS_ODN/LCD0_LDE0      |
| I2S_BCLK1/PCM0_OUT          | GPIOB0      | 33           | 34           | GND         |                         |
| I2S_LRCLK1/PCM0_CLK         | GPIOB1      | 35           | 36           | GPIOA28     | I2S_BCLK0/PCM0_IN       |
| I2S_MCLK1/PCM0_SYNC         | GPIOB2      | 37           | 38           | GPIOA31     | I2S_D1                  |
| GND                         |             | 39           | 40           | GPIOA27     | I2S_D0                  |

图 8.1 40pin GPIO 定义

访问 GPIO 引脚，如 **GPIOB19** ( $32+19=51$ )，我们首先需要使用下面命令 **export** 这个引脚：

```
sudo echo 51 > /sys/class/gpio/export
```

如果操作成功，我们可以查看 sys 文件系统节点信息：

```
ls /sys/class/gpio/gpio51
active_low direction edge power subsystem uevent value
```

使用下面的命令，设置 GPIO 的方向为输出：

```
echo out > /sys/class/gpio/gpio51/direction
```

使用下面的命令，设置 GPIO 的方向为输入：

```
echo in > /sys/class/gpio/gpio51/direction
```

使用下面的命令，读取 GPIO 的值：

```
cat /sys/class/gpio/gpio51/value
```

当方向设置为输出，我们可写入 1 或 0 到 GPIO 引脚：

```
echo 1 /sys/class/gpio/gpio51/value
```

## 8.2 WiringLMK

WiringLMK 是为 Banana Pro 和 LeMaker Guitar 用 C 语言写的一个访问 GPIO 库。它是从 Drogon 编写的 WiringPi 移植过来的。LeMaker 团队对这个库进行了修改，并让 WiringLMK 与原生的 wiringPi 有相同的接口。

使用下面的命令下载并安装 WiringLMK：

```
git clone https://github.com/LeMaker/WiringLMK.git
cd WiringLMK
chmod +x ./build
sudo ./build
```

下面是一个简单的“Hello World” 例子，演示如何使 LED 灯闪：

```
#include <wiringPi.h>
int main (void)
{
    wiringPiSetupPhys () ;
    pinMode (11, OUTPUT) ;
    for (;;)
    {
        digitalWrite (0, HIGH) ; delay (500) ;
        digitalWrite (0, LOW) ; delay (500) ;
    }
    return 0 ;
}
```

更多内容，请访问：<http://wiki.lemaker.org/WiringLMK/zh-hans>。

## 8.3 LMK.GPIO

LMK.GPIO 提供了一个用 Python 语言写的用来控制 Banana Pro 和 LeMaker Guitar 上的 GPIO 的类。它是从 Drogon 编写的 WiringPi 移植过来的。LeMaker 团队对这个库进行了修改，让 LMK.GPIO 与原生的 RPi.GPIO 有相同的接口。目前的 LMK.GPIO 版本还不支持 SPI、I2C、硬件 PWM 和串行功能。

使用下面的命令下载并安装 LMK.GPIO：

```
sudo apt-get install python-dev
git clone https://github.com/LeMaker/LMK.GPIO.git
cd LMK.GPIO
python setup.py install
sudo python setup.py install
```

下面是一个简单的“Hello World”例子，展示如何使 LED 灯闪烁：

```
#!/usr/bin/env python

import LMK.GPIO as GPIO
import time

#LED Mode BOARD
PIN_NUM = 7
GPIO.setmode(GPIO.BOARD)

while True:
    try:
        GPIO.setup(PIN_NUM, GPIO.OUT)
    except:
        print("Failed to setup GPIO %d", PIN_NUM)

    GPIO.output(PIN_NUM, True)
    time.sleep(0.5)
    GPIO.output(PIN_NUM, False)
    time.sleep(0.5)
```

更多内容，请访问：<http://wiki.lemaker.org/LMK.GPIO/zh-hans>。

## 8.4 LeScratch

Scratch@MIT 教育软件：Scratch 1.4 设计的初衷是教育和学习。它可以与 LeScratch 一起运行。LeScratch 是一个运行在 LeMaker SBC 上让 Scratch 与硬件通信的 Scratch 消息处理器，可以很方便地让任何年龄阶段的人使用 LeScratch 学习微处理器和传感器知识，从而达到教育的目的。它实现了使用 Python 扩展外设中发送和接收命令的功能。下载链接：

<https://github.com/LeMaker/LeScratch/>。

更多内容，请访问：[http://wiki.lemaker.org/LeScratch\\_User\\_Guide/zh-hans](http://wiki.lemaker.org/LeScratch_User_Guide/zh-hans)。

## 8.5 另请参阅

[1] LeScratch: <https://github.com/LeMaker/LeScratch>。

[2] Scratch - 一个免费的编程语言和在线社区: <https://scratch.mit.edu/>。

LeMaker

## 9 备份系统镜像

### 9.1 使用 Win32Diskimager 备份系统镜像

我们可以用 Win32Diskimager 烧写系统镜像到 MicroSD 卡。我们也可以使用 Win32Diskimager 从 MicroSD 卡读取数据来备份操作系统镜像。

**第 1 步：**把安装好系统镜像 MicroSD 卡插入电脑或读卡器。



图 9.1 MicroSD 卡

**第 2 步：**打开 Win32DiskImager 工具，填入镜像文件名字，选择 MicroSD 卡设备，然后点击“read”按钮，开始备份系统镜像。

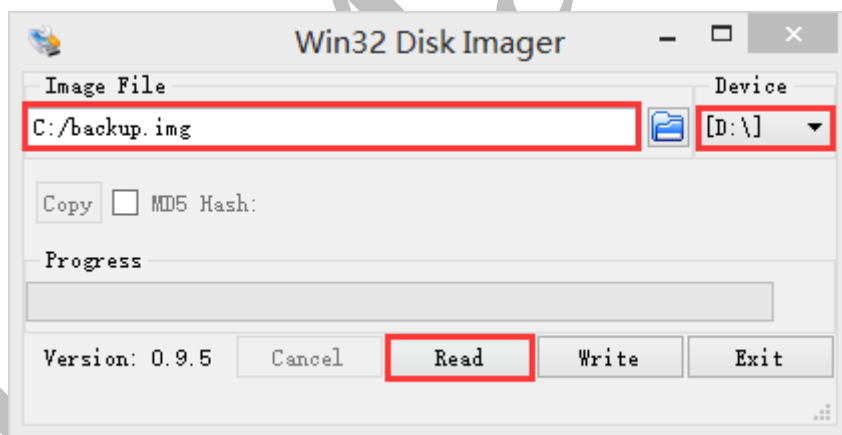


图 9.2 设置 Win32DiskImager

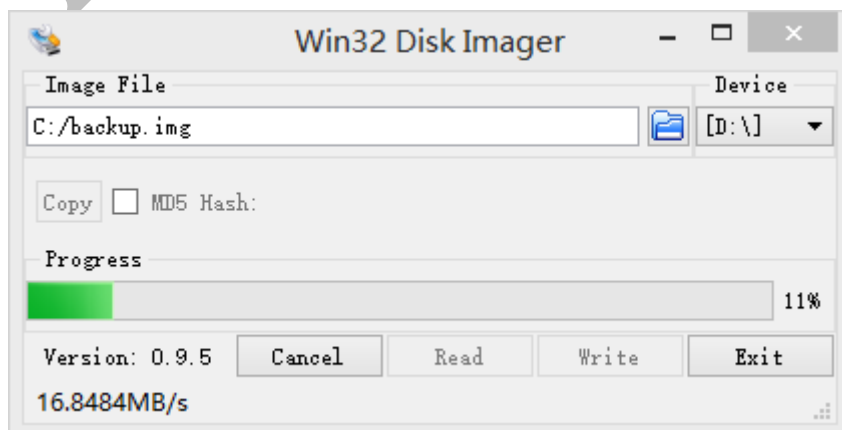


图 9.3 备份系统镜像

## 9.2 使用 dd 命令备份系统镜像

我们可以使用 Win32Diskimager 工具备份系统镜像，也可以使用在 Linux 系统下使用 **dd** 命令备份操作系统。

**第 1 步：**插入一个 U 盘到 LeMaker Guitar，用来存储系统备份镜像。U 盘将一般会挂载在 /media/lemaker/目录下。使用下面的命令查看 U 盘挂载的路径：

```
lsblk
```

**第 2 步：**使用下面的命令备份操作系统镜像。

```
cd /media/lemaker/udiskname  
sudo dd if=/dev/mmcblkx of=backup.img bs=1M  
sync  
sudo umount /media/lemaker/udiskname
```

**注意：**/dev/mmcblkx 表示/dev/mmcblk0 或/dev/mmcblk1。当操作系统从 EMMC NAND Flash 启动，/dev/mmcblk0 是 EMMC NAND Flash 的设备文件节点。如果插入 MicroSD 卡到 MicroSD 卡插槽，我们将会发现/dev/mmcblk1 变成 MicroSD 卡的设备节点。但是当操作系统从 MicroSD 卡启动，/dev/mmcblk0 则是 MicroSD 卡的设备文件节点，且/dev/mmcblk1 是 EMMC NAND Flash 的设备文件节点。因此，我们可以根据选择的 mmc 块设备文件备份 MicroSD 卡或 EMMC NAND Flash 上的系统镜像。如果要从系统备份文件恢复系统，只需要执行下面的命令：

```
dd if=backup.img of=/dev/mmcblkx bs=1M  
sync
```

## 9.3 另请参阅

[1] Win32 Disk Imager – 一个烧写镜像到 USB 棒或 SD/CF 卡工具：

<http://sourceforge.net/projects/win32diskimager/>。

## 10 Android

Android 是由 Google 公司开发的基于 Linux 内核的移动设备操作系统。Android 带有可以直接操作的用户界面，最初为触屏设备设计，例如智能手机、平板电脑、有特定用户界面的电视机、汽车和腕戴手表。Android 操作系统使用触摸输入，与真实世界的动作一致，像滑动、点击、捏合和扩大这些动作来操作屏幕上的目标和虚拟键盘。尽管最初为触摸屏设备输出设计，但是它也可以用于游戏机、数码照相机、个人电脑和其他电子产品。

### 10.1 基本屏幕

#### 主屏幕

滑动屏幕，即可切换到另一个主屏幕（共支持 5 主屏幕），点击状态栏上的“主页”按钮，切换到默认的主屏幕。

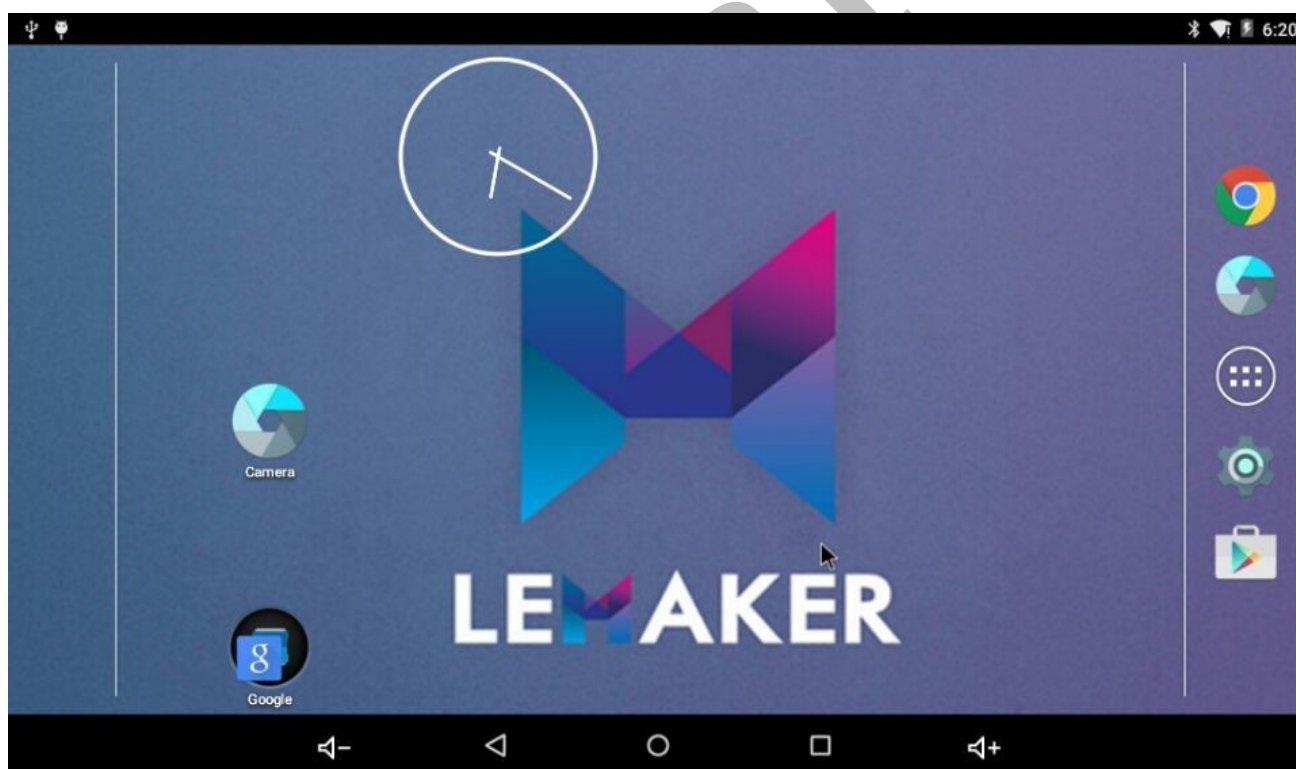


图 10.1 Android 主屏幕

#### 主菜单

点击主屏幕的右侧的  图标，即可进入主菜单。所有的应用程序都将主菜单上列出，可以通过拖动屏幕，滚动浏览所有的内容。

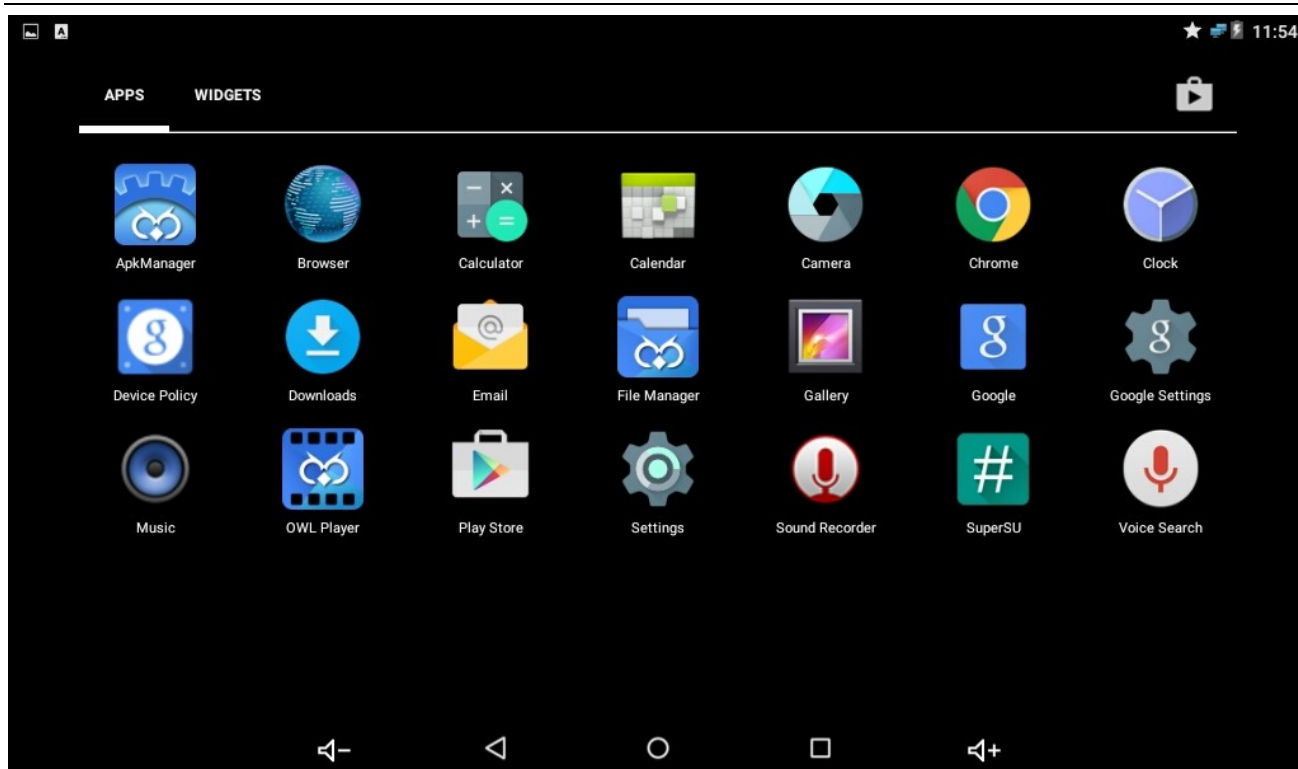


图 10.2 Android 主菜单

## 10.2 基本设置

点击  图标进入基本设置页面，包括“网络”、“设备”、“个人”和“系统”。

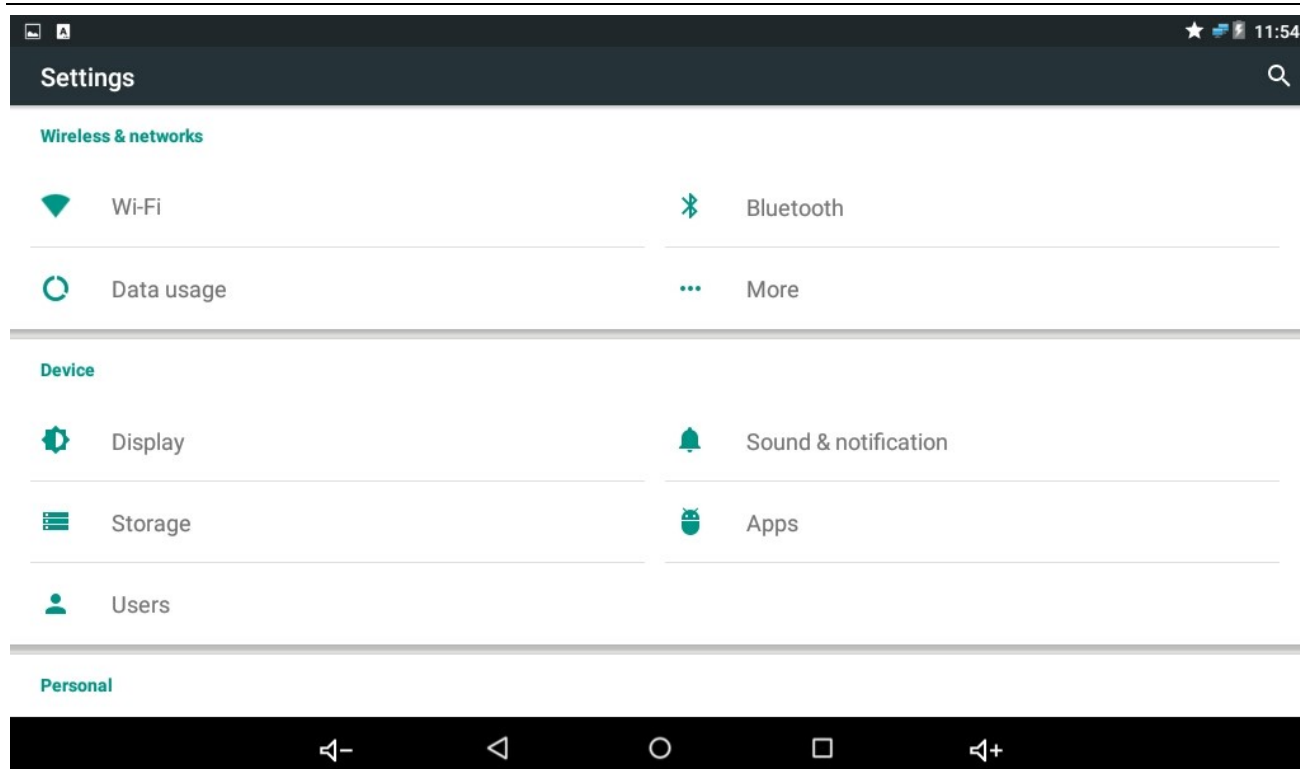


图 10.3 Android 基本设置