

第 1 章 物联网 openwrt 开发概述.....	3
1.1 OpenWrt 系统介绍.....	3
1.1.1 OpenWrt 系统的定义和特点.....	3
1.1.2 OpenWrt 系统的发展历史.....	3
1.2 物联网介绍.....	4
1.3 RT5350 开发板介绍.....	4
第 2 章 物联网 openwrt 开发环境构建.....	7
2.1 硬件环境构建.....	7
2.1.1 主机与目标板结合的交叉开发模式.....	7
2.1.2 硬件要求.....	7
2.2 软件环境构建.....	8
2.2.1 主机 Linux 操作系统的安装.....	8
2.2.2 主机 Linux 操作系统上网络服务的配置与启动.....	19
第 3 章 相关工具、命令的使用.....	21
3.1 Windows 环境下的工具介绍.....	21
3.1.1 代码阅读与编辑工具 Source Insight.....	21
3.1.2 文件传输工具 FileZilla.....	26
3.1.3 远程登录工具 SecureCRT.....	27
3.1.4 TFTP 服务器软件 Tftpd32.....	29
3.2 Linux 环境下的工具、命令介绍.....	30
3.2.1 代码阅读、编辑工具 KScope.....	30
3.2.2 远程登录工具 C-kermit.....	34
3.2.3 编辑命令 vi.....	35
3.2.4 查找命令 grep、find 命令.....	36
3.2.5 在线手册查看命令 man.....	37
3.2.6 其他命令: tar、diff、patch.....	38
第 4 章 下载配置编译 openwrt.....	40
4.1 安装相关工具、库.....	40
4.2 下载 openwrt 源码.....	40
4.3 编译 openwrt 源码.....	41
4.4 调试串口连接和系统烧写.....	42
4.4.1 调试串口连接.....	42
4.4.2 烧写 openwrt 系统.....	45
第 5 章 物联网 openwrt 开发之 VLAN 设置.....	49
5.1 VLAN 介绍.....	49
5.1.1 VLAN 介绍.....	49
5.1.2 RT5350 开发板 VLAN 支持.....	49
5.2 灵活配置端口为 WAN、LAN.....	50
5.2.1 openwrt 中的 vlan.....	50
5.2.2 RT5350 开发板 VLAN 默认配置.....	50
5.2.3 全部设置为 LAN.....	51
第 6 章 物联网 openwrt 开发之 Web 界面 Luci 移植.....	52
6.1 WEB 界面移植.....	52
6.2 Luci.....	52
6.2.1 Luci 编译.....	52
6.2.2 Luci 页面编辑.....	54

第 1 章 物联网 openwrt 开发概述

本章目标

- 了解 openwrt、物联网发展历史
- 了解 RT5350 开发板

1.1 OpenWrt 系统介绍

1.1.1 OpenWrt 系统的定义和特点

1. OpenWrt 系统的定义

OpenWrt 可以被描述为一个嵌入式的 Linux 发行版,(主流路由器固件有 dd-wrt, tomato, openwrt 三类)而不是试图建立一个单一的、静态的系统。OpenWrt 的包管理提供了一个完全可写的文件系统,从应用程序供应商提供的选择和配置,并允许您自定义的设备,以适应任何应用程序。

对于开发人员,OpenWrt 是使用框架来构建应用程序,而无需建立一个完整的固件来支持;对于用户来说,这意味着其拥有完全定制的能力,可以用前所未有的方式使用该设备。

OpenWRT 是一个高度模块化、高度自动化的嵌入式 Linux 系统,拥有强大的网络组件和扩展性,常常被用于工控设备、电话、小型机器人、智能家居、路由器以及 VOIP 设备中。同时,它还提供了 100 多个已编译好的软件,而且数量还在不断增加,而 OpenWrt SDK 更简化了开发软件的工序。

OpenWRT 不同于其他许多用于路由器的发行版,它是一个从零开始编写的、功能齐全的、容易修改的路由器操作系统。实际上,这意味着您能够使用您想要的功能而不加进其他的累赘,而支持这些功能工作的 linux kernel 又远比绝大多数发行版来得新。

2. OpenWrt 系统的特点

如果对 Linux 系统有一定的认识,并想学习或接触嵌入式 Linux 的话,OpenWRT 很适合。而且 OpenWRT 支持各种处理器架构,无论是对 ARM, X86, PowerPC 或者 MIPS 都有很好的支持。其多达 3000 多种软件包,囊括从工具链(toolchain),到内核(linux kernel),到软件包(packages),再到根文件系统(rootfs)整个体系,使得用户只需简单的一个 make 命令即可方便快速地定制一个具有特定功能的嵌入式系统来制作固件。

一般嵌入式 Linux 的开发过程,无论是 ARM, PowerPC 或 MIPS 的处理器,都必需经过以下的开发过程:

- 1、创建 Linux 交叉编译环境;
- 2、建立 Bootloader;
- 3、移植 Linux 内核;
- 4、建立 Rootfs (根文件系统);
- 5、安装驱动程序;
- 6、安装软件;

熟悉这些嵌入式 Linux 的基本开发流程后,不再局限于 MIPS 处理器和无线路由器,可以尝试在其它处理器,或者非无线路由器的系统移植嵌入式 Linux,定制合适自己的应用软件,并建立一个完整的嵌入式产品。

1.1.2 OpenWrt 系统的发展历史

OpenWrt 项目由 2004 年 1 月开始,第一个版本是基于 Linksys 提供的 GPL 源码及 uclibc 中的 buildroot 项目,这个版本称为“stable”版,在网上至今仍有很多项目使用这个版本,较为有名

Freifunk-Firmware 和 Sip@Home.

到了 2005 年初, 一些新的开发人员加入了这项目, 几个月后他们释出了第一个 “experimental” 版本, 这和以前版本不同的是, 这版本差不多完全舍弃了 Linksys 的 GPL 源码, 使用了 buildroot2 作为核心技术, 将 OpenWrt 完全模块化, OpenWrt 使用 Linux 正式发行的核心源码 (2.4.30), 加上了一些补丁和网络驱动, 开发队伍更为 OpenWrt 添加了许多免费的工具, 可以直接把 Image 写入 Flash (mtd) 里面, 设定无线功能和 VLAN 交换功能, 这个版本名为 “White Russian”, 而 1.0 版本于 2005 年底公布。

1.2 物联网介绍

物联网是新一代信息技术的重要组成部分, 其英文名称是: “The Internet of things”。顾名思义, 物联网就是物物相连的互联网。这有两层意思: 其一, 物联网的核心和基础仍然是互联网, 是在互联网基础上的延伸和扩展的网络; 其二, 其用户端延伸和扩展到了任何物品与物品之间, 进行信息交换和通信。物联网就是 “物物相连的互联网”。物联网通过智能感知、识别技术与普适计算、广泛应用于网络的融合中, 也因此被称为继计算机、互联网之后世界信息产业发展的第三次浪潮。物联网是互联网的应用拓展, 与其说物联网是网络, 不如说物联网是业务和应用。因此, 应用创新是物联网发展的核心, 以用户体验为核心的创新 2.0 是物联网发展的灵魂。

活点定义: 利用局部网络或互联网等通信技术把传感器、控制器、机器、人员和物等通过新的方式联在一起, 形成人与物、物与物相联, 实现信息化、远程管理控制和智能化的网络。物联网是互联网的延伸, 它包括互联网及互联网上所有的资源, 兼容互联网所有的应用, 但物联网中所有的元素 (所有的设备、资源及通信等) 都是个性化和私有化的。

2013、2014 年, 智能路由产品层出不穷, 从极路由到如意云、从 360 安全路由到小米 “智能家居” 路由; 在此之前, 许多创客玩家们也都有尝试在开发板上自己制造出一台路由器。究竟一台智能路由是如何做出来的呢? 我们是否可以在众多开源组件的基础上自己来构建一台智能路由?

现在社会是物联网高速发展的社会, 物联网无处不在, 但是目前还没有一个成功的物联网开发平台, 对于刚接触这方面人更是无从下手, 物联网是把任何物品与互联网连接起来, 进行信息交换和通讯, 以实现智能化识别、定位、跟踪、监控和管理的一种网络。他不仅需要强大的网络功能, 还需要具有对物的检测和控制功能, 这就需要有一个多功能的网络监控平台来做这些东西。在众多的网络操作系统中 openwrt 具有开源、功能强大、移植性好等特点, 它拥有强大的网络组件和扩展性, 常常被用于工控设备、电话、小型机器人、智能家居、路由器以及 VOIP 设备中。同时, 它还提供了 1000 多个已编译好的软件, 而且数量还在不断增加, 而 OpenWrt SDK 更简化了开发软件的工序。这是最适合做物联网平台的操作系统。

在业内, 都说 2014 年将是物联网的元年, 而 2016 年, 将是高峰。今年, 2014 年, 正处于物联网蓬勃发展的开始阶段, 我们更应该学好 openwrt、学好物联网。

本书带领大家开启 F403 科技创意室_RT5350 开发板的学习之旅, 首先在上面移植 openwrt, 因为这里开发的第一步。

F403 科技创意室研发的 RT5350 开发板, 就给大家提供了一个很好的硬件平台, 并且还出了 openwrt 教程, 该教程会不断更新!

请选择我们, F403 科技创意室 (<http://f403tech.taobao.com/>), 跟我们一起来学习 openwrt 吧!

1.3 RT5350 开发板介绍

F403 科技创意室独立研发的 RT5350 开发板具有丰富的外围接口, 众多的外围模块, 比较小的体积, 而且完美支持 openwrt 系统, 它是目前最好的物联网开发平台。它引出大量的 gpio, 具有 spi, i2c 等常用的总线接口, 在物联网平台上面可以发挥其重要作用。

RT5350开发板

外形尺寸: 118mm x 60mm

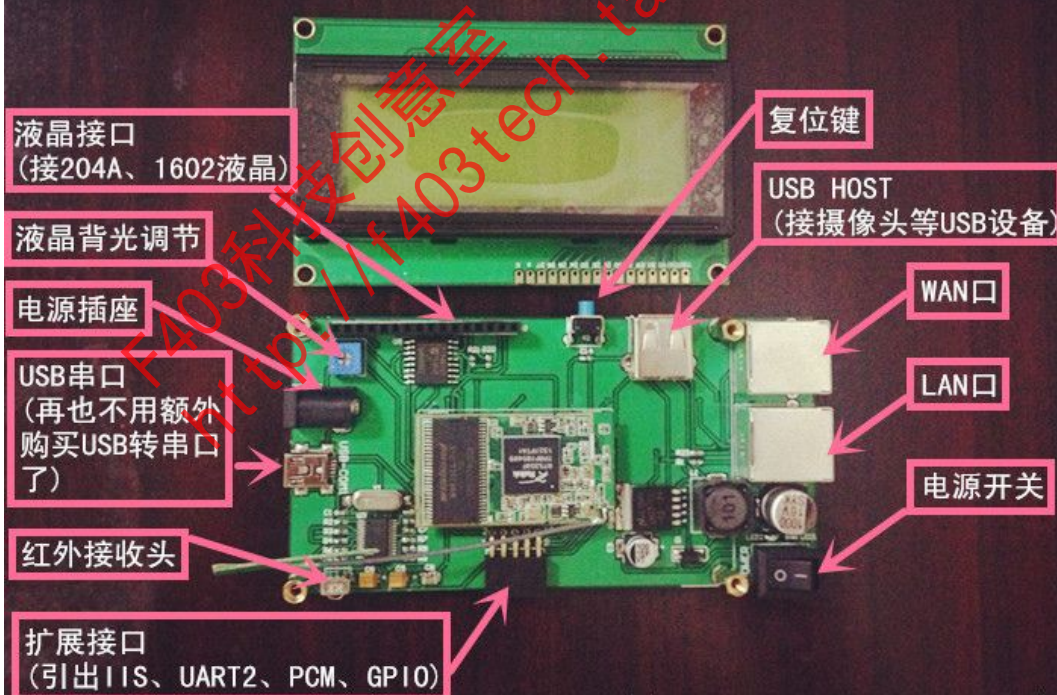
板载USB转串口、红外接收头、液晶接口、USB、WAN、LAN

板载排母:

用于接 声卡模块、矩阵键盘模块、继电器模块、RS232
串口模块



<http://f403tech.taobao.com/>



<http://f403tech.taobao.com/>



注意：

- (1) 此教程为 F403 科技创意室版权所有；
- (2) 此 openwrt 教程不不断更新，请关注我们的淘宝店：
<http://f403tech.taobao.com/>
向客服索要最新版本的教程。
- (3) 针对该教程，我们特意创建了 QQ 超级群，由教程的作者亲自指导大家学习。详情请咨询客服。

F403科技创意室
<http://f403tech.taobao.com/>

第 2 章 物联网 openwrt 开发环境构建

本章目标

- 了解物联网 openwrt 开发的交叉开发模式；
- 搭建硬件、软件开发环境；

2.1 硬件环境构建

2.1.1 主机与目标板结合的交叉开发模式

开发 PC 机上的软件时，可以直接在 PC 机上编辑、编译、调试软件，最终发布的软件也是在 PC 机上运行。对于物联网 openwrt 开发，最初的嵌入式设备是一个空白的系统，需要通过主机为它构建基本的软件系统，并烧写到设备中；另外，嵌入式设备的资源并不足以用来开发软件。所以需要用到交叉开发模式：在主机上编辑、编译软件，然后在目标板上运行、验证程序。主机指 PC 机，目标板指嵌入式设备，在本书中，目标板就是 F403 科技创意室独立研发的 RT5350 开发板，“目标板”、“开发板”、“单板”都是同一个意思。

对于 F403 科技创意室独立研发的 RT5350 开发板，进行物联网 openwrt 开发时一般可以分为以下 3 个步骤。

(1) 在主机上编译 U-boot，然后通过 SPI Flash 烧写器烧入单板。

通过 SPI Flash 烧写器烧写程序的效率非常低，而且还需要取下 Flash 芯片，它适用于烧写空白单板。为方便开发，通常选用具有串口传输、网络传输、烧写 Flash 功能的 U-boot，它可以快速地从主机获取可执行代码，然后烧入单板。

该 U-boot 在开发板生产时，已经由工厂烧入到了 SPI Flash 中。特别强调，在做物联网 openwrt 开发时，一般不建议大家自己开发、重新烧写 U-boot！否则，一不小心，开发板就会变成砖头的哦！

(2) 在主机上编译 openwrt，通过 U-boot 烧入单板或直接启动。

一个可以在单板上运行的 openwrt 是进行后续开发的基础。

(3) 在主机上编译各类应用程序，经过验证后烧入单板。

烧写、启动 U-boot 后，就可以通过 U-boot 的各类选项来下载、烧写、运行程序了。启动 Linux 后，也是通过执行各种命令来启动应用程序的。怎么输入这些命令、查看命令运行的结果呢？一般通过串口来进行输入输出。所以交叉开发模式中，主机与目标板通常需要 2 种连接：串口、网络，如图 2.1 所示。

图 2.1 交叉开发模式

2.1.2 硬件要求

1. 主机要求

一般的 PC 机就可以用来进行物联网 openwrt 开发，它应该满足以下要求。

- ① 有一个 USB 口；
- ② 支持网络；
- ③ 至少 20GB 的硬盘。

因为要通过串口来操作 Linux，所以得使用 USB-串口转换器。幸运的是，F403 科技创意室独立研发的

RT5350开发板，已经板载了该转换器。

2. 目标板要求

本书使用 F403 科技创意室 (<http://f403tech.taobao.com/>) 提供的开发板，当然，读者也可以使用其他公司的开发板，只是可能很多功能需要自己通过飞线、额外购买更多模块来实现。

由于后面会设计到大量的教程、大量的应用的移植，所以建议大家用 32MB 的内存、8MB 的 Flash。

所谓的硬件开发环境搭建很简单，将主机与目标板通过串口线(直接用 mini USB 线连接主机和目标板即可)、网线(接单板上的任意网口)连接起来，将各类设备连接到目标板上去即可。

2.2 软件环境构建

2.2.1 主机 Linux 操作系统的安装

1. 在 Windows 上安装虚拟机

本书基于 Ubuntu 12.04 进行开发，它是一个很容易安装和使用的 Linux 发行版。光盘映像文件的下载地址为 <http://releases.ubuntu.com/>。

安装方法有好几种：将映像文件刻录成光盘后安装，通过网络安装等。对于不熟悉 Linux 的读者，可以通过 VMware 虚拟机软件使用映像文件安装，这样可以在 Windows 中使用 Linux——反过来也是可以的，安装 Linux 后，再使用 VMware 安装 windows，这样就可以在 Linux 中同时使用 Windows 了。

这几种安装方法基本相同，下面介绍在 Windows 中通过 VMware 来安装 Linux 的方法。不管是哪种方法，都建议单独使用一个分区来存放本书所涉及的源码、编译结果，这可以避免当系统出错、系统重装时破坏学习成果。

从 VMware 的官方网站 <http://www.vmware.com> 下载到 VMware 工具，安装后，参照以下方法安装、设置 Linux。

(1) 在 VMware 中建立一个虚拟机器：

要建立一个虚拟机器，可以想象需要指定硬盘、内存、网络。在 VMware 中可以使用实际的硬盘，也可以使用文件来模拟硬盘。依照下面的一系列图形就可以建立了。

① 启动 VMware，如图 2.1 所示。选“Create a New Virtual Machine”。

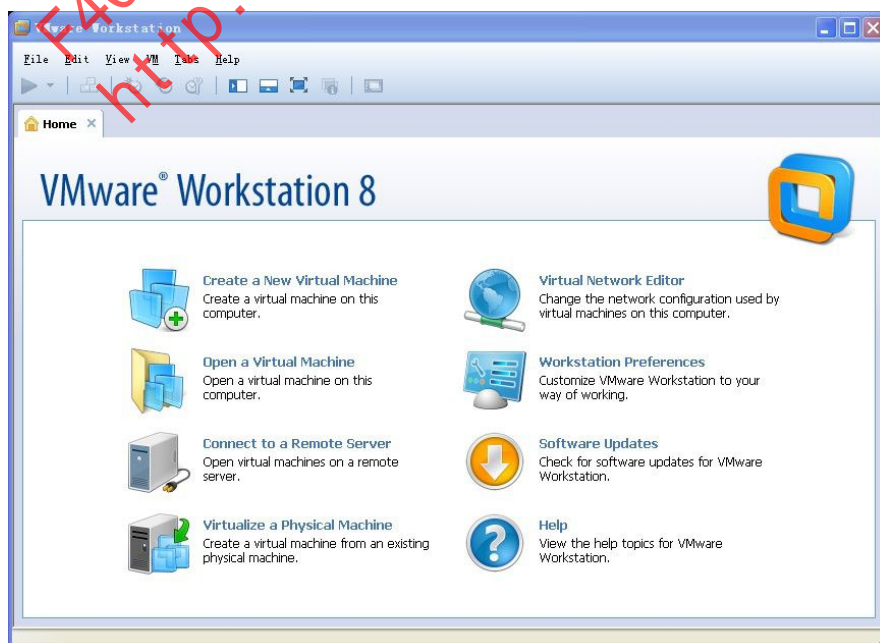


图 2.1 启动 VMware

② 在后续界面中使用默认选项，直到出现如图 2.2 所示的界面，在里面选择“Custom”自己定制虚拟机器。



图 2.2 选择定制虚拟机

③ 选择虚拟机的格式，使用默认选项即可，如图 2.3 所示。选择客户操作系统，选择“I will install the operating system later.”，如图 2.4 所示。

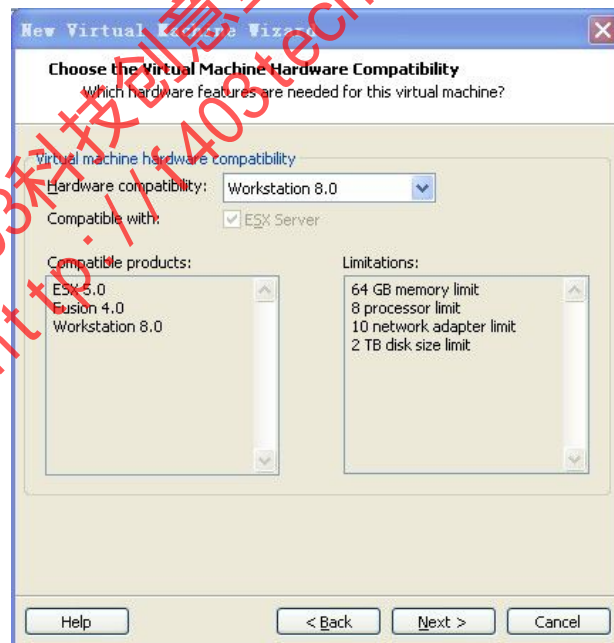


图 2.3 选择虚拟机的格式

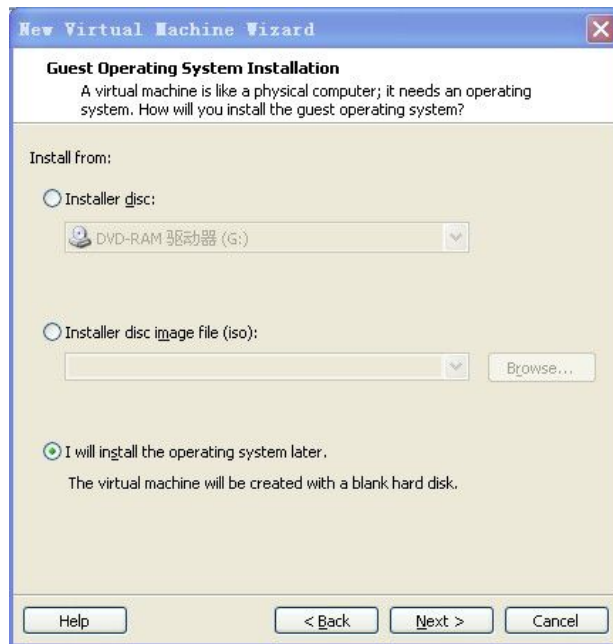


图 2.4 选择客户操作系统

④ 在 Windows 中使用 VMware 安装 Linux，Windows 被称为“Host Operatins System”（主机操作系统），Linux 被称为“Gest Operatins System”（客户操作系统）。选择 Linux 作为客户操作系统，版本为“Ubuntu”，如图 2.5 所示。

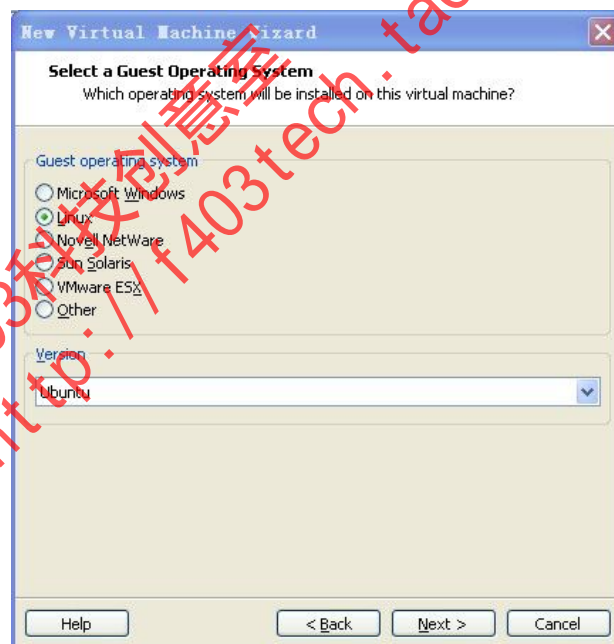


图 2.5 选择客户操作系统

⑤ 设置虚拟机的名字及存储位置，如图 2.6 所示。

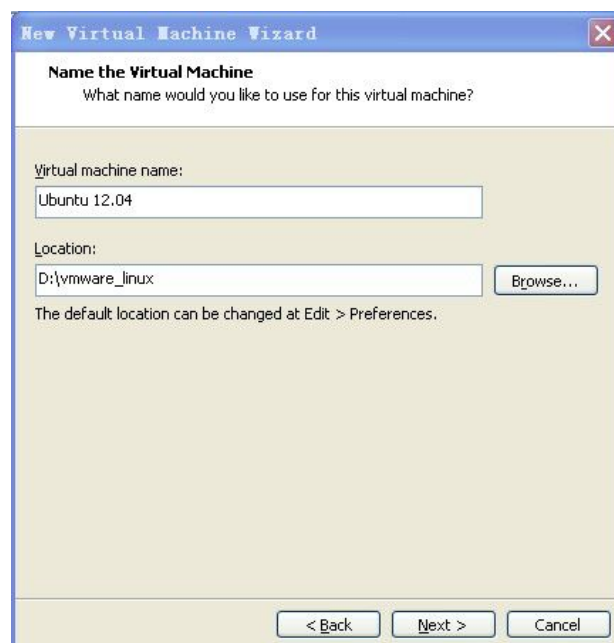


图 2.6 设置虚拟机的名字及存储位置

⑥ 指定处理器个数(根据自己的实际情况设置)，如图 2.7 所示。指定虚拟机的内存容量，在图 2.8 所示的界面中有推荐值及取值范围。

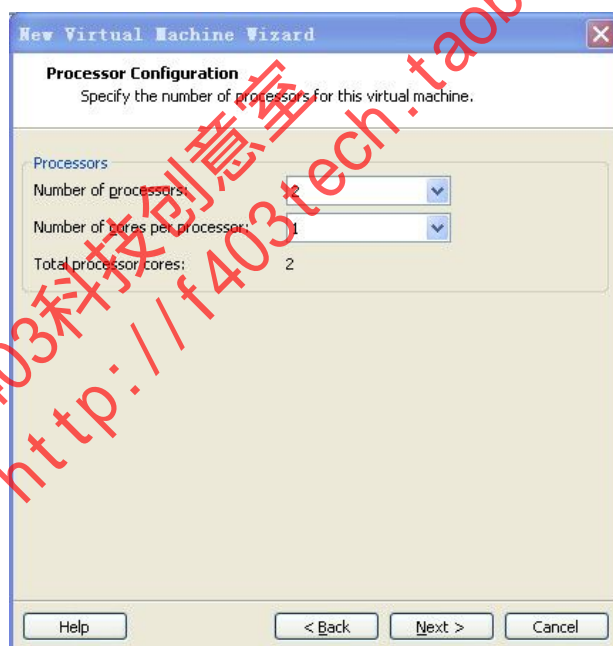


图 2.7 指定处理器个数

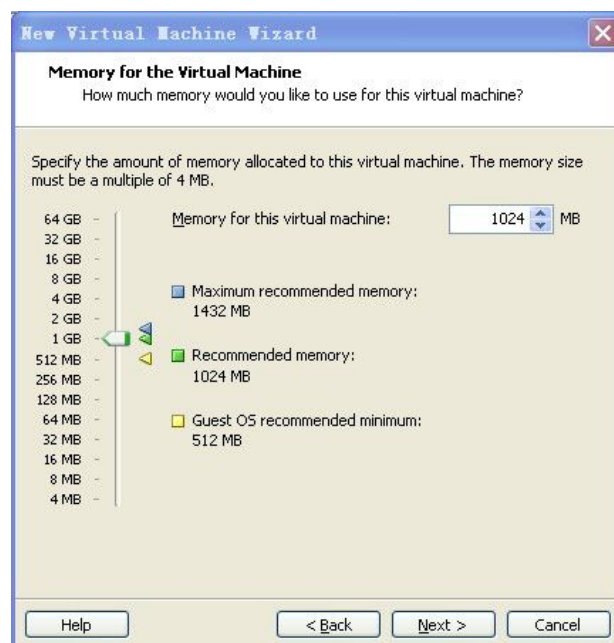


图 2.8 指定虚拟机的内存容量

⑦ 指定虚拟机的网络连接类型，一般使用桥接方式(bridge networking)，如图 2.9 所示。安装完华后可以再进行修改。

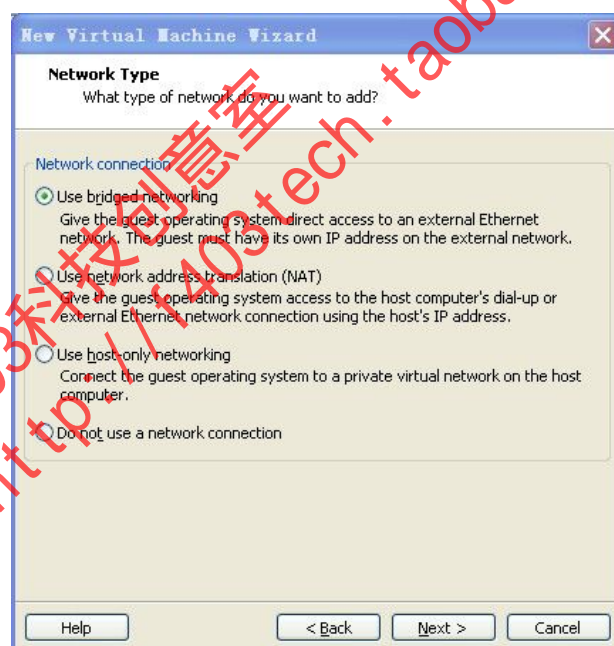


图 2.9 指定虚拟机的网络连接类型

⑧ 选择“I/O Adapter”，使用默认值，如图 2.10 所示。

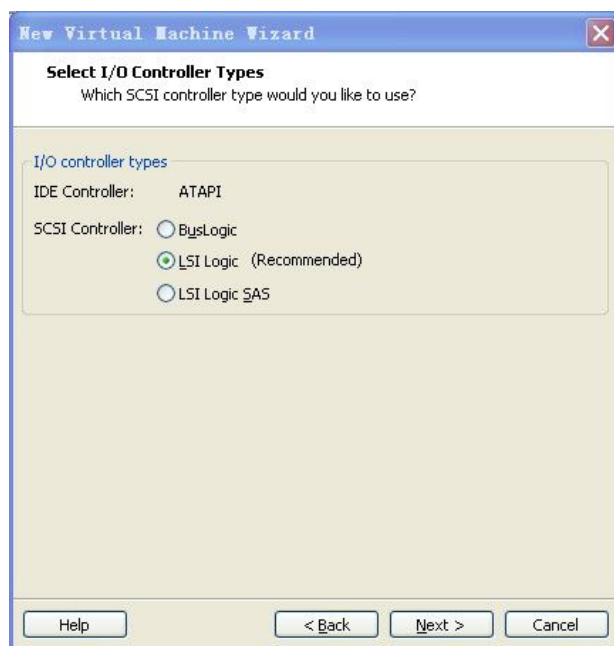


图 2.10 选择 “I/O Adapter”

⑨ 图 2.11、2.12、2.13、2.14 共 4 个图都是用来创建虚拟硬盘。在图 2.13 中，为了方便管理，建议选择 “storage virtual disk as a single file”。

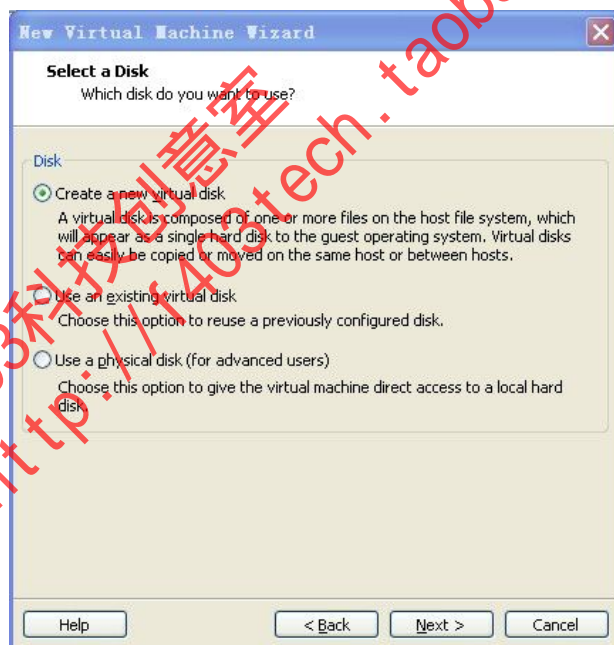


图 2.11 选择创建新的虚拟硬盘

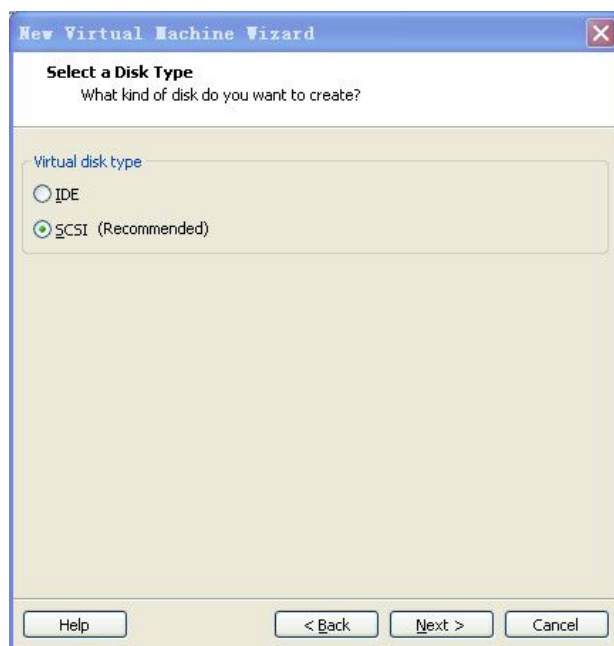


图 2.12 选择硬盘(使用默认类型)

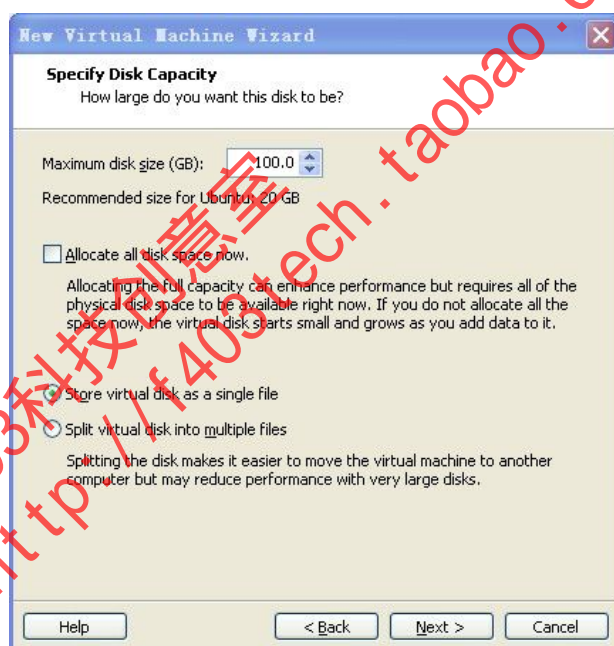


图 2.13 指定硬盘容量



图 2.14 设置虚拟硬盘的名字(在 Windows 下将新建一个文件来表示这个虚拟硬盘)

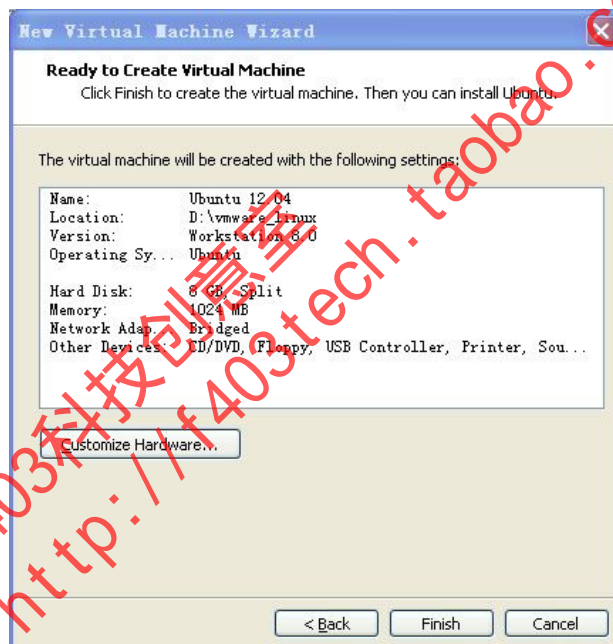


图 2.15 完成

单击“完成”按钮，现在已经创建了一个虚拟机器。

⑩ 在上图中单击“完成”按钮后，就创建了一个虚拟机器。

2. 在虚拟机上安装 Linux

本书使用 Ubuntu 12.04 的光盘文件 ubuntu-12.04.1-desktop-i386.iso 文件进行安装。下面简单介绍关键步骤，其他步骤可以参看安装时出现的说明。

(1) 在虚拟机上使用光盘文件。

如图 2.16 所示，进入虚拟机的编辑界面，选中“CD/DVD”，在右边的界面中，选择“Connect at power on”(表示开启虚拟机时就连接光盘)；然后选择“Use ISO image”，如果有实际的光盘，可以选择“Use physical drive”。

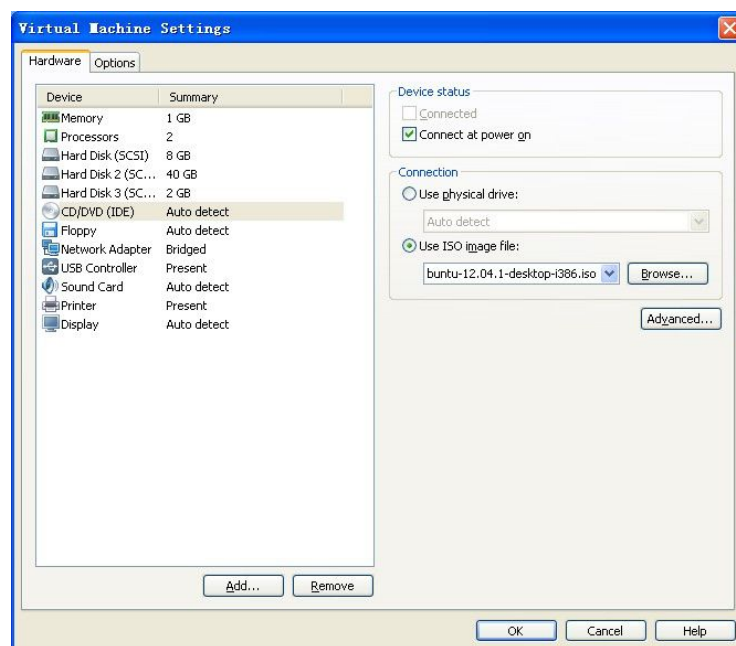


图 2.16 在虚拟机上使用光盘文件

(2) 启动虚拟机，它使用前面设置的光盘文件启动，这时候即可开始安装 Linux。

如图 2.17 所示，在虚拟机启动后，桌面有个名为“install Ubuntu”的图标，单击它进行安装。接着出现图 2.18 所示界面，使用默认设置，点“Continue”。当出现如图 2.19 所示的界面时，选择“Erase disk and install Ubuntu”。

注意：在 VMware 的操作系统中，要将鼠标释放出来(回到 Windows 中)，按“Ctrl+Alt”键即可。

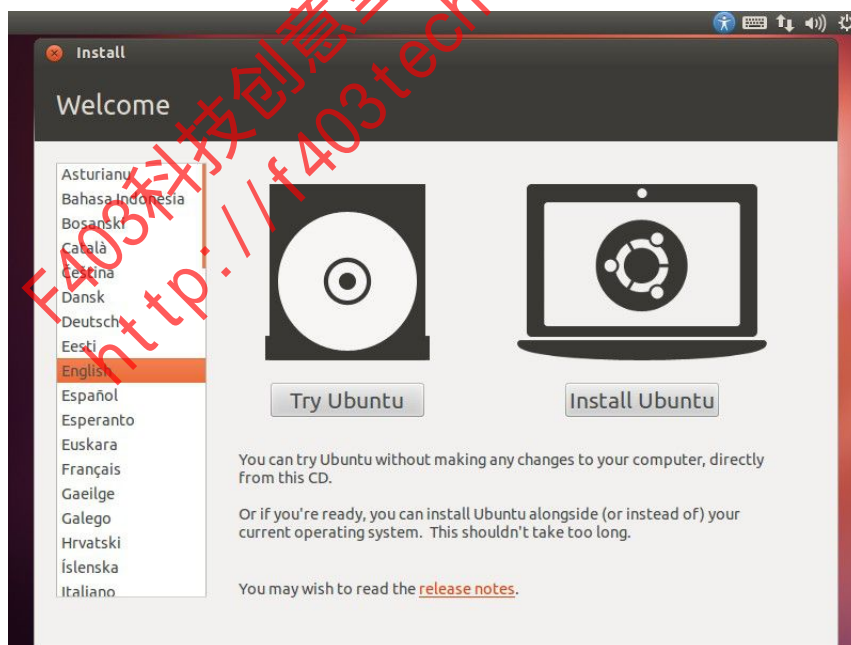


图 2.17 通过“手动”来使用硬盘

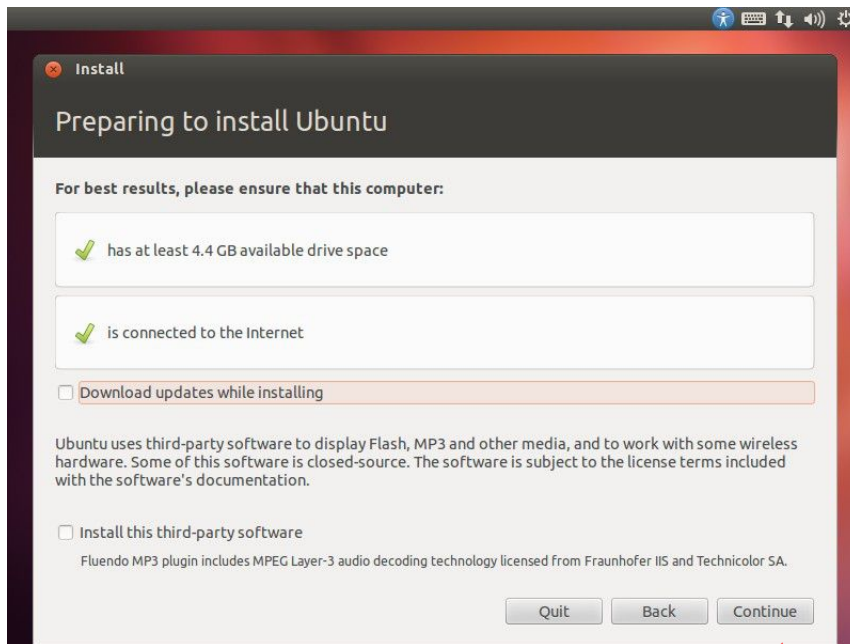


图 2.18 条件检查

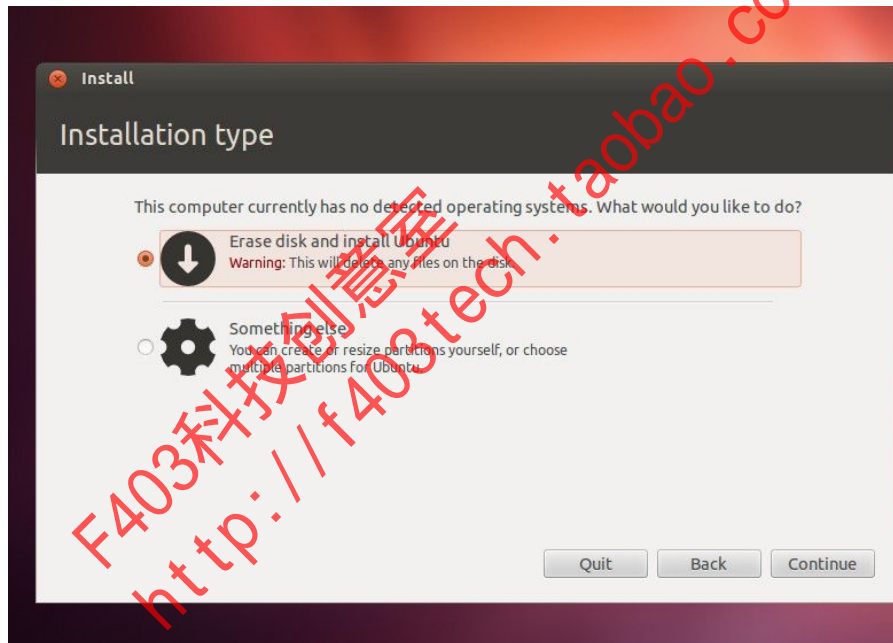


图 2.19 安装类型

③ 在紧接着出现的界面中单击“install now”，如图 2.20 所示。



图 2.20 准备分区

④ 然后在后续的界面中使用默认值，安装程序会进行格式化虚拟硬盘等操作。当出现如图 2.21 所示的界面时，在里面设置用户名及密码。

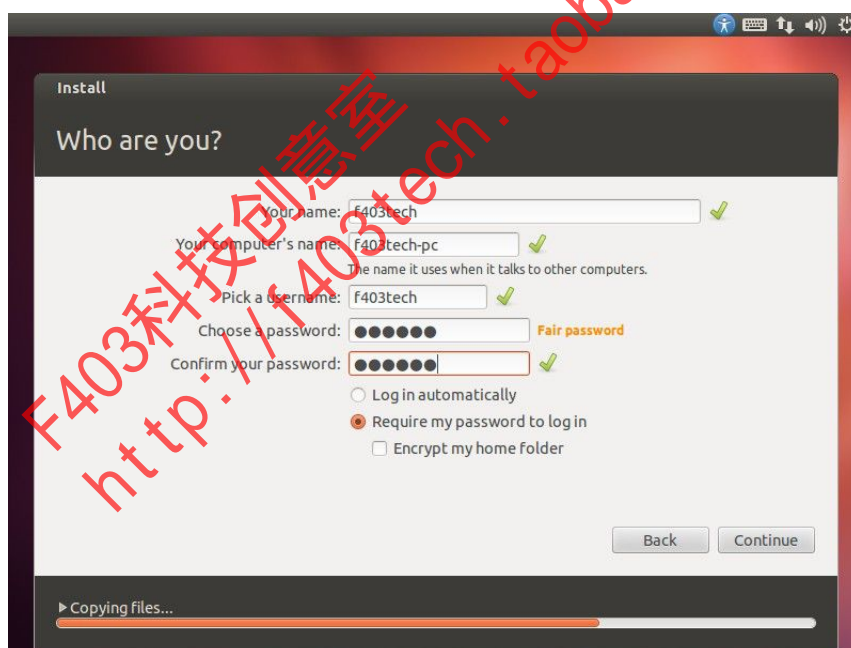


图 2.21 设置用户名和密码

⑤ 继续往下，开始安装系统。当安装完成时，出现如图 2.22 所示的界面：

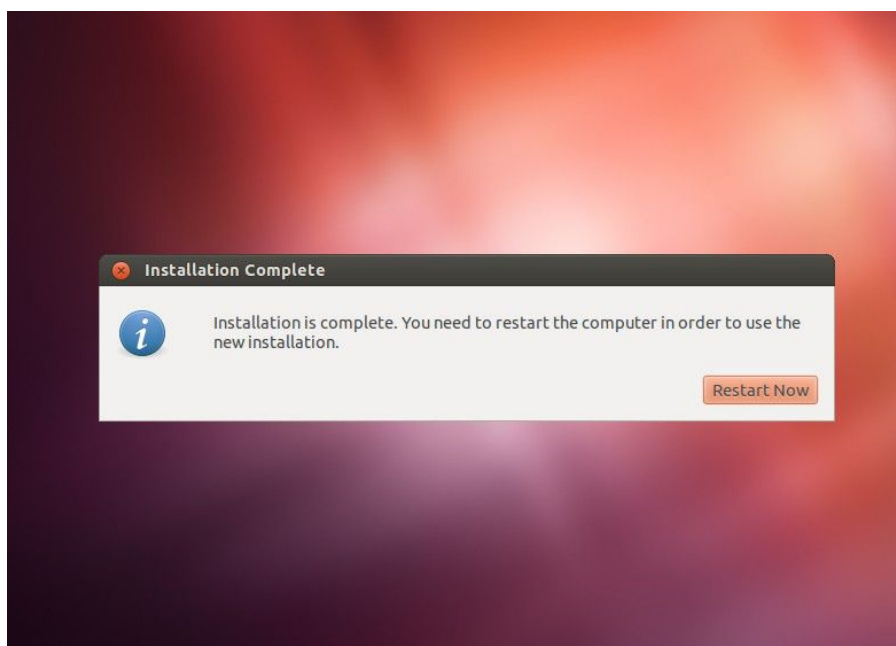


图 2.22 安装结束

这时候从 VMware 的菜单“VM”->“Setting...”进入虚拟机的设置界面，在“CD-ROM”的设置界面去掉“Connect at power on”（参考图2.16）。然后点击 Linux 安装界面的“Restart now”即可（如果不能重启，直接关闭 VMware 后再启动）。

2.2.2 主机 Linux 操作系统上网络服务的配置与启动

面配置 Linux，启动 ftp、ssh 这 2 个服务。如果不是通过远程登录 Linux，而是直接在 Linux 中进行开发，则 ftp、ssh 这两个服务不用开启。

（1）准备使用软件维护工具 apt-get。

Ubuntu 12.04 中没有安装 ftp、ssh、nfs 服务器软件，它提供了一个很方便的安装、升级、维护软件的工具：apt-get。apt-get 从光盘、网络上下载程序，并安装。所以要使用 apt-get，通常要保证 Linux 能上网。

第一次使用 apt-get 来安装程序之前，要先完成以下两件事：

① 修改/etc/apt/sources.list，将其中注释掉的网址打开

在安装 Ubuntu 12.04 的时候，如果网络无法使用，它会自动将/etc/apt/sources.list 中各项注释掉。比如需要将以下两行开头的“#”号去掉：

```
#deb http://cn.archive.ubuntu.com/ubuntu/ gutsy main restricted
#deb-src http://cn.archive.ubuntu.com/ubuntu/ gutsy main restricted
.....
```

注意：由于/etc/apt/sources.list 属于 root 用户，而 Ubuntu 12.04 中屏蔽了 root 用户的使用，要修改它，需要使用 sudo 命令。比如可以使用“sudo vi /etc/apt/sources.list”来修改它，或者使用“sudo gedit &”命令启动图形化的文本编辑器，再打开、编辑它。

② 更新可用的程序列表

执行如下命令即可，它只是更新内部的数据库以确定：有哪些程序已经安装、哪些没有安装、哪些有新版本。apt-get 程序将使用这个数据库来确定怎样安装用户指定的程序，并找到和安装它所依赖的其他程序。

下面的指令***

```
$ sudo apt-get update
```

(2) 安装、配置、启动服务。

首先说明，Ubuntu 7.10中隐藏了 root 用户，就是说不能使用 root 用户登录，这可以避免不小心使用 root 权限而导致系统崩溃。当需要使用 root 权限时，使用“sudo”命令，比如要修改/etc/exports 文件时，修改如下所示：

```
# sudo vi /etc/exports
```

现在可以使用 apt-get 来安装软件了，以下的安装、配置、启动方法在 Ubuntu 7.10自带的帮助文档中都有说明：

① 安装、配置、启动 ftp 服务。

执行以下命令安装，安装后即会自动运行：

```
$ sudo apt-get install vsftpd
```

修改 vsftpd 的配置文件/etc/vsftpd.conf，将下面几行前面的“#”号去掉。

```
#local_enable=YES
```

```
#write_enable=YES
```

上面第一行表示是否允许本地用户登录，第二行表示是否允许上传文件。

修改完毕之后，执行以下命令重启 ftp 服务：

```
$ sudo /etc/init.d/vsftpd restart
```

② 安装、配置、启动 ssh 服务。

执行以下命令安装，安装后即会自动运行：

```
$ sudo apt-get install openssh-server
```

它的配置文件为/etc/ssh/sshd_config，使用默认配置即可。

注意：

(3) 此教程为 F403 科技创意室版权所有；

(4) 此 openwrt 教程不不断更新，请关注我们的淘宝店：

<http://f403tech.taobao.com/>

向客服索要最新版本的教程。

(3) 针对该教程，我们特意创建了 QQ 超级群，由教程的作者亲自指导大家学习。详情请咨询客服。

第3章 相关工具、命令的使用

本章目标

- 掌握 Windows 下的代码阅读、编辑工具 Source Insight
- 掌握在 Windows 下与 Linux 进行交互的工具：FileZilla、SecureCRT
- 掌握 Linux 下的代码阅读、编辑工具 KScope，串口工具 C-kermit
- 掌握一些常用的 Linux 命令

3.1 Windows 环境下的工具介绍

仅仅将 Linux 作为服务器来进行开发时(比如使用 VMware 运行 Linux 以提供编译环境,日常工作仍然在 Windows 下进行),可以使用 Windows 下的几款优秀工具提高工作效率。比如使用 Source Insight 阅读、编辑代码;使用 FileZilla 与 Linux 服务器、RT5350 开发板进行文件传输;使用 SecureCRT 远程登录 Linux 进行各类操作。

3.1.1 代码阅读与编辑工具 Source Insight

Source Insight 是一款极具革命性的代码阅读、编辑工具,它内建了 C/C++、C#、Java 等多种编程语言的分析器。Source Insight 会自动分析源代码,动态地生成并更新一个数据库,并通过丰富而有效的表现形式使得阅读、编辑代码非常方便、高效。比如它会将 C 语言中全局变量、局部变量标上不同的颜色;将光标移到某个变量、函数上时,窗口下方会自动显示它们的定义;借助于不断更新的数据库,可以快速找到函数的调用关系;编辑代码时,变量名、函数名会自动补全;……。

基于这些功能,在 Windows 下阅读 Linux 内核源码这类庞大的软件时,使用 Source Insight 有助于理清各类错综复杂的变量、函数之间的关系。

从网址 <http://www.sourceinsight.com/> 上可以下载一个试用版本,它具有正式版的全部功能,试用期为30天。

下面以 Linux 内核源码为例介绍 Source Insight 的使用。

1. 创建一个 Source Insight 工程

启动 Source Insight 之后,它默认的支持文件中没有以“.S”结尾的汇编语言文件,点击菜单“Options”->“Document Options”,在弹出的对话框中选择“Document Type”为“C Source File”,在“File filter”中添加“*.S”类型,如图3.1所示。

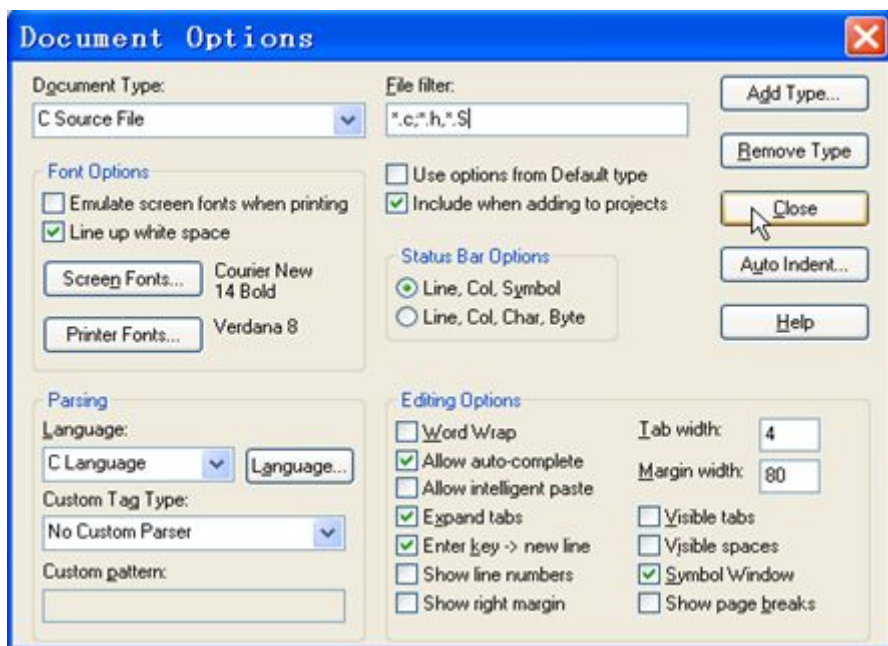


图3.1 设置 Source Insight 支持的文件类型

然后点击菜单“Projects”->“New Project”开始建立一个新的工程，界面如图3.2所示。

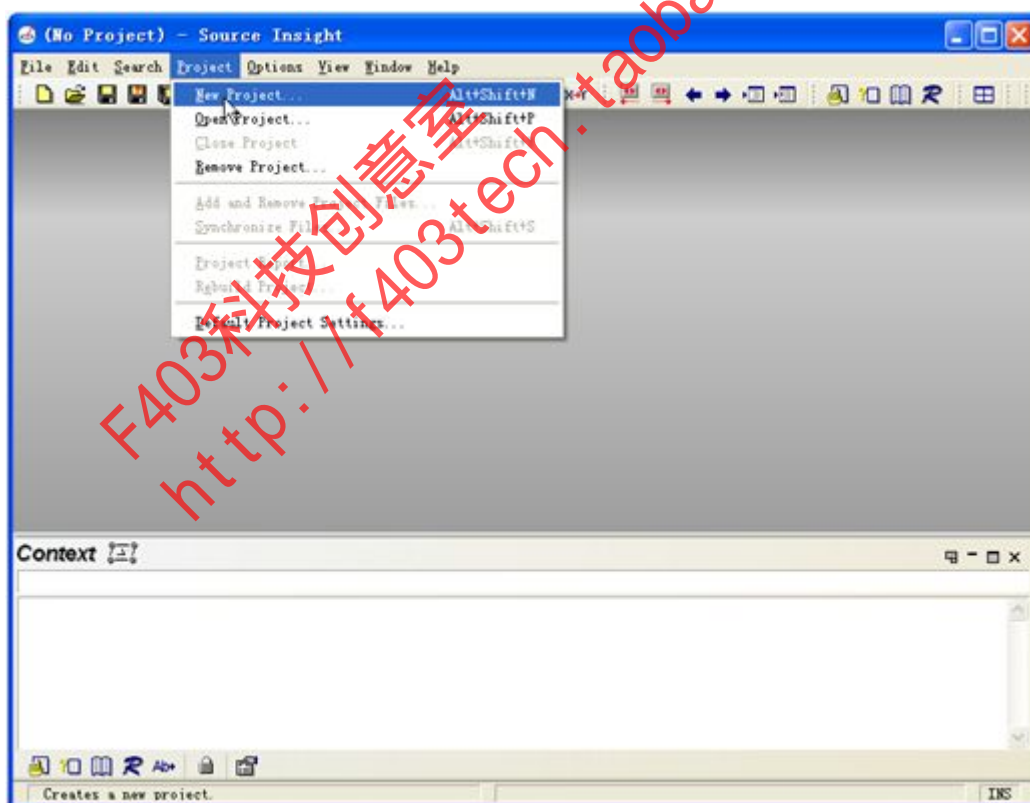


图3.2 新建一个 Source Insight 工程

在随后出现的界面中，输入工程的名称和工程数据的存放位置。本小节中，假设内核源码位置为 E:\kernel_projects\linux-2.6.22.6，将要建立的 Source Insight 工程名为 linux-2.6.22.6，在 E:\kernel_projects\sc 目录下存放工程数据，则如图 3.3 一样设置，然后点击“OK”按钮（如果 E:\kernel_projects\sc 目录还不存在，会提示是否创建这个目录）。

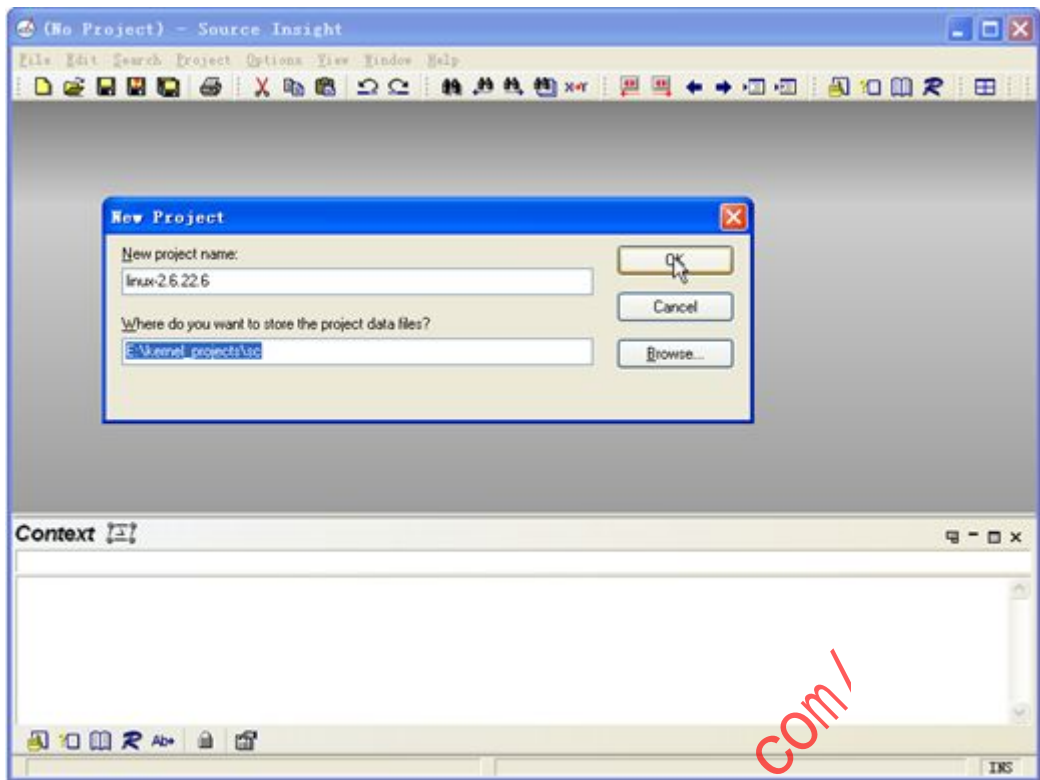


图3.3 输入 Source Insight 工程名称及保存位置

可以想象，接下来的步骤应该是指定源码的位置及添加源文件。如图3.4所示指定内核源码位置后，点击“OK”按钮进入下一个设置界面。

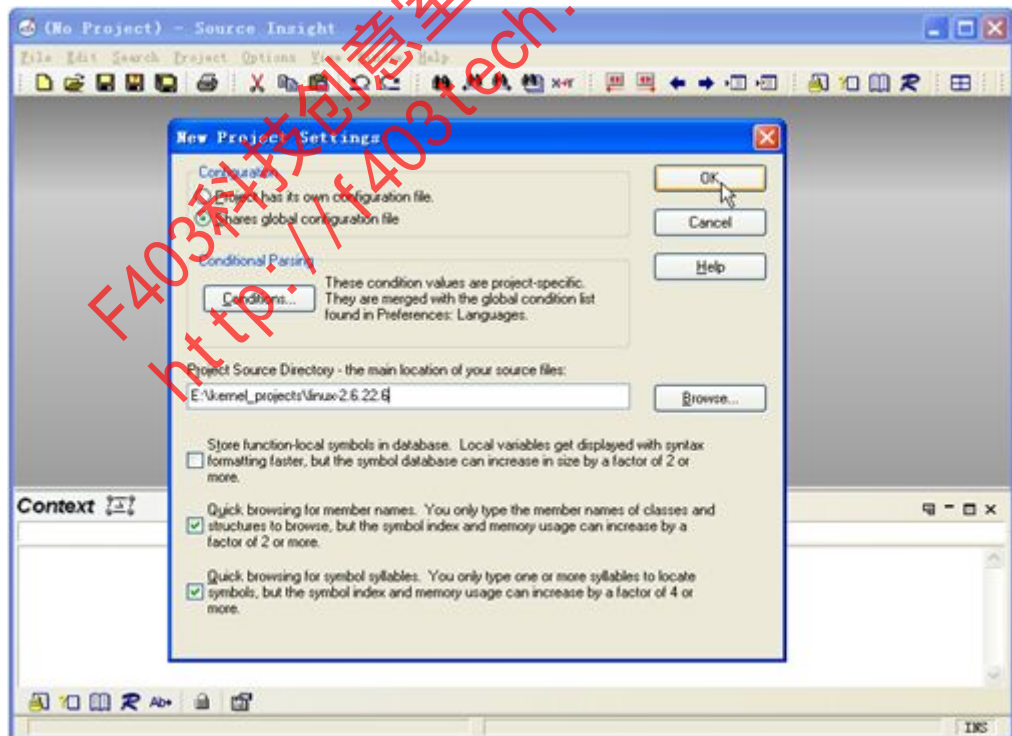


图3.4 指定源码位置

图3.5是添加源文件的操作界面：先点击“Add All”按钮，在弹出的对话框中选中“Include top level sub-directories”（表示将添加第一层子目录中的文件）、“Recursively add lower sub-directories”（表示递归地加入底层的子目录，即加入所有子目录中的文件）；然后点击“OK”控制开始加入内核的所有源文

件。

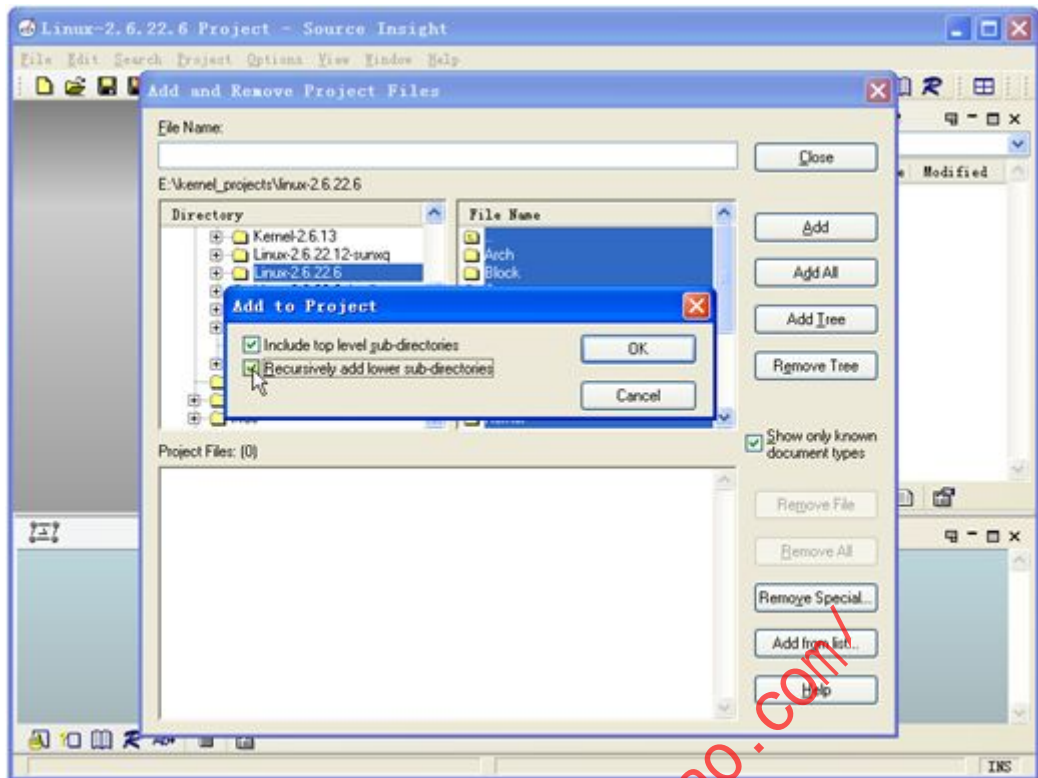


图3.5 添加源文件

实际上，由于内核支持多个架构的CPU、多个型号的目标板，而本书只关心 S3C2410、S3C2440目标板，所以可以在 Source Insight 工程中去掉其他不相关的文件。仍然在与图4.5相似的界面中(可以点击菜单“Projects”->“Add and Remove Project Files”进入)，选择某个目录后，使用“Remove Tree”按钮将整个目录下的文件从工程中移除。

要移除的目录如下，操作的示例界面如图3.6所示：

- ① arch 目录下除 arm 外的所有子目录；
- ② arch/arm 目录下以“mach-”开头的目录(除 mach-s3c2410、mach-s3c2440之外)；
- ③ arch/arm 目录下以“plat-”开头的目录(除 plat-s3c24xx 之外)；
- ④ include 目录下以“asm-”开头的目录(除 asm-arm、asm-generic 之外)；
- ⑤ include/asm-arm 目录下以“arch-”开头的目录(除 arch-s3c2410之外)。

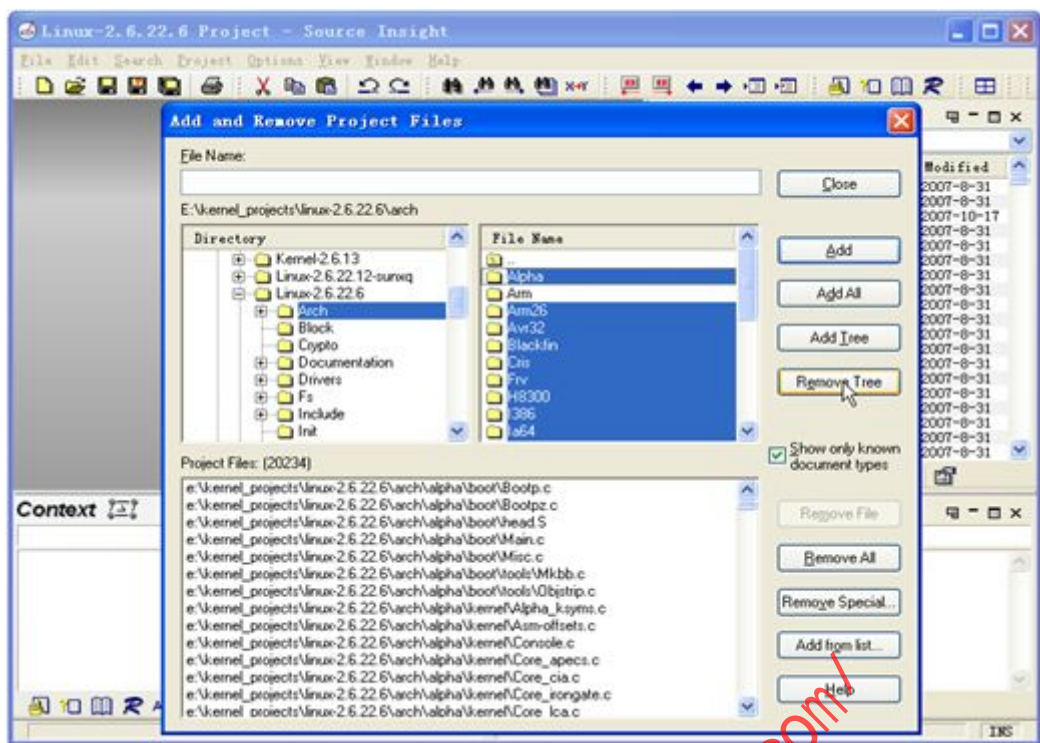


图3.6 移除源文件

至此，Source Insight 工程建立完毕。

2. “同步”源文件

所谓“同步”源文件就是在 Source Insight 工程中建立一个数据库，它里面保存有源文件中各变量、函数之间的关系，使得阅读、编辑代码时能快速提供各种辅助信息(比如以不同颜色显示不同类型的变量等)。

这个数据库会自动建立，但是对于比较庞大的源码的工程，建议初次使用时手工建立数据库，这使得 Source Insight 工程很快地建立所有源码的全面的关系图。

点击菜单“Projects”->“Synchronize Files”，会弹出如图3.7所示的对话框，选中其中的“Force all files to be re-parsed”(表示“强制分析所有文件”)，然后点击“OK”按钮即可生成数据库。

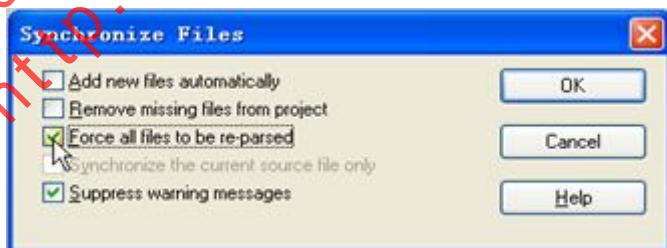


图3.7 同步源文件

3. Source Insight 工程使用示例

在 Source Insight 右边的文件列表中选择打开 s3c2410fb.c 文件，可以得到如图4.8所示的界面，它的中间是主窗口，可以在里面阅读、编辑代码；左边是“Symbol window”(符号窗口)，可以从中快速找到当前文件中的变量、函数，宏定义等；下边是“Context window”(上下文窗口)，在主窗口中将光标放在某个变量、函数、宏上面时，会在这个窗口中显示它们的定义，比如在图3.8中，这个窗口中显示了 request_irq 函数的定义。

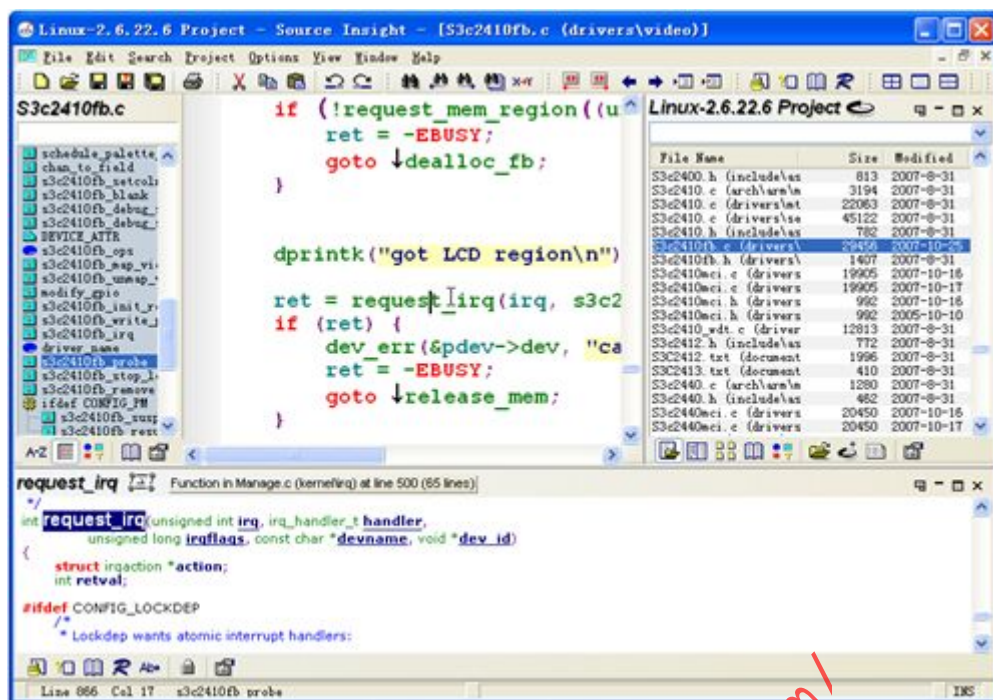


图3.8 Source Insight 的使用界面

在主窗口中，按住 Ctrl 键的同时，使用鼠标点击某个变量、函数、宏，就可以跳到定义它们的位置；双击上下文窗口也可以达到同样的效果。

同时按住 “Alt”、“,” 键可以令主窗口倒退到上一画面，同时按住 “Alt”、“.” 键可以令主窗口前进到前一个画面。

在某个变量、函数、宏上用鼠标右键点击一下，在弹出的对话菜单中选择 “Lookup References”，可以快速地在所有源文件中找到对它们的引用，这比搜索整个源码目录快多了。

Source Insight 还有很多使用技巧，上面只介绍了几种常用的技巧，读者在使用过程中可以通过各个菜单了解更多。

3.1.2 文件传输工具 FileZilla

FileZilla 是一款 SSH 客户端软件，只要在 Linux 上按照 2.2.2 小节的说明安装、启动了 SSH 服务，就可以使用 FileZilla 在 Windows 与 Linux 之间进行文件传输。

安装、启动 FileZilla 后，跳过前面的设置向导。假设 Linux 服务器的 IP 为 192.168.1.103，用户名为 f403tech，密码为 123456，如图 3.9 所示在 “Host”、“Username”、“Password” 中分别填入 192.168.1.103、f403tech、123456，然后回车就可以连接上 Linux 服务器了。

在图 4.9 所示界面中，可以在左、右两边的窗口中拖拽文件进行上传、下载。

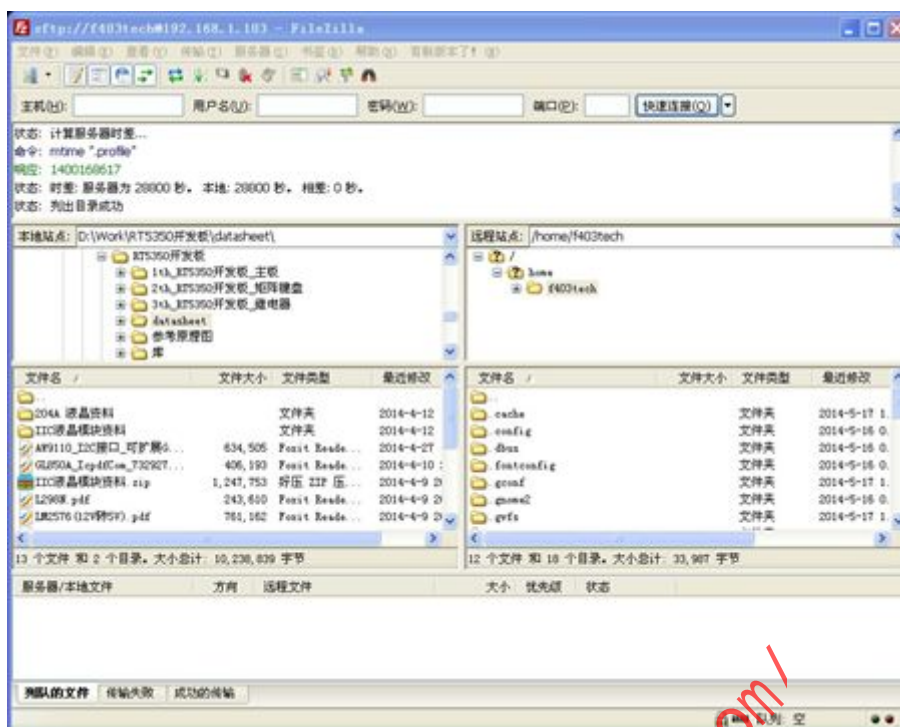


图3.9 FileZilla 的使用界面

3.1.3 远程登录工具 SecureCRT

SecureCRT 支持多种协议，比如 SSH2、SSH1、Telnet、Serial 等。可以用它来连接 Linux 服务器，作为一个远程控制台进行各类操作；也可以用它来连接串口，操作目标板。

SecureCRT 可以从网址 <http://www.vandyke.com/products/securecrt/> 中下载，有30天的免费试用期。

安装、启动 SecureCRT 后，跳过初始设置界面。点击菜单“File”->“connect”，出现如图3.10所示的界面，点击其中的“New Session”按钮开始建立新的连接。

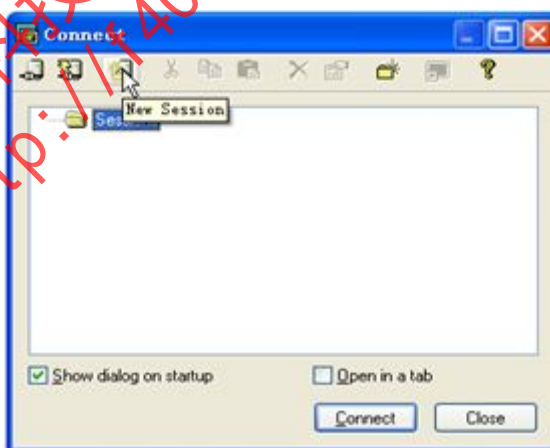


图3.10 SecurtCRT 新建连接的界面

在随后出现的对话框中选择传输协议：SSH2或 Serial，如图3.11所示。

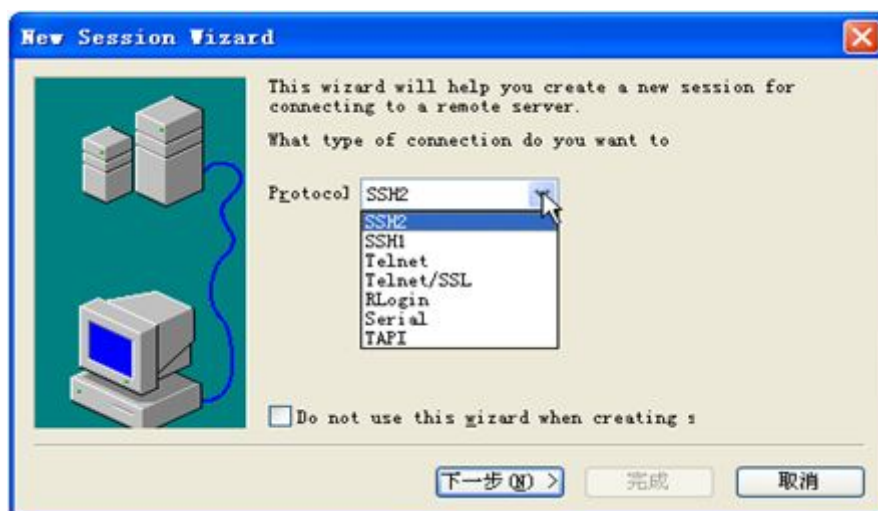


图3.11 选择传输协议

在图4.11所示的对话框中点击下一步，进行更详细的设置。对于 SSH2、Serial 协议，请分别参考图3.12、图3.13进行设置。

图4.12：输入服务器 IP(Hostname)、用户名 (Username)。

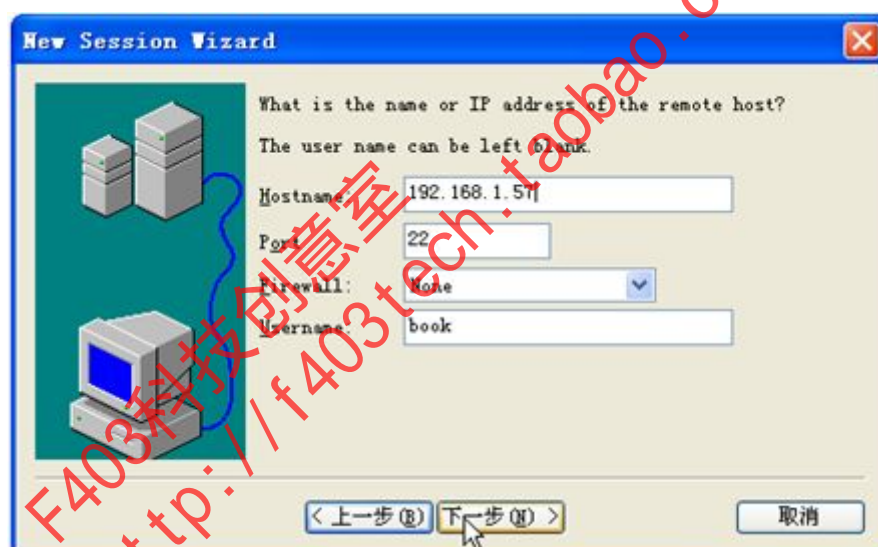


图3.12设置 SSH2连接

图3.13：选择串口1(COM1)，设置波特率(Baud rate)为115200、数据位(Data bits)为8、不使用校验位(Parity)、停止位(Stop bits)为1，不使用流控(Flow Control)。

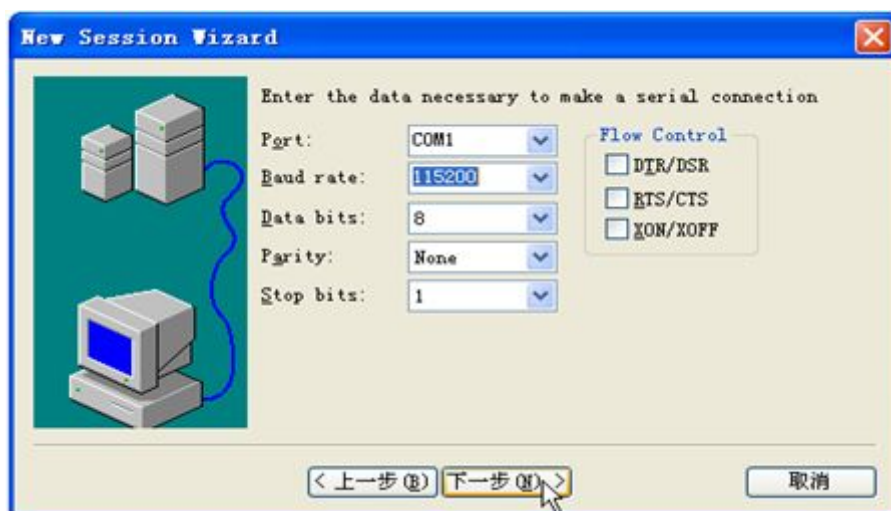


图3.13 设置 Serial 连接

按照图3.12、3.13设置完成后，点击“下一步”按钮，在后续对话框中可以设置这个新建的连接的名字。

当建立完新的连接后，可以看到如图3.14所示的对话框，也可以通过点击菜单“File”->“connect”启动这个对话框。在里面双击某个连接，即可启动它。

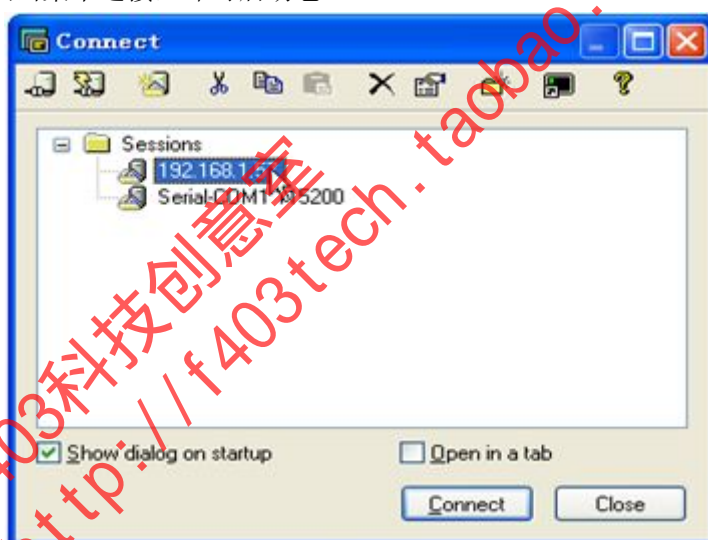


图3.14 启动连接

3.1.4 TFTP 服务器软件 Tftpd32

Tftpd32是一款轻便的 DHCP、TFTP、SNTP 和 Syslog 服务器软件，同时也是一款 TFTP 客户端软件。使用 U-Boot 时可以使用它的 TFTP 服务器功能来下载软件到目标板中(也可以使用 Linux 中的 NFS 服务代替)。

Tftpd32可以从网址 http://tftpd32.jounin.net/tftpd32_download.html 中下载。它可以直接运行，参考图3.15 进行设置：选择服务器的目录(要传输的文件放在这个目录中)、选择 IP 地址(对于有多个 IP 的系统而言，要从中选择一个)。

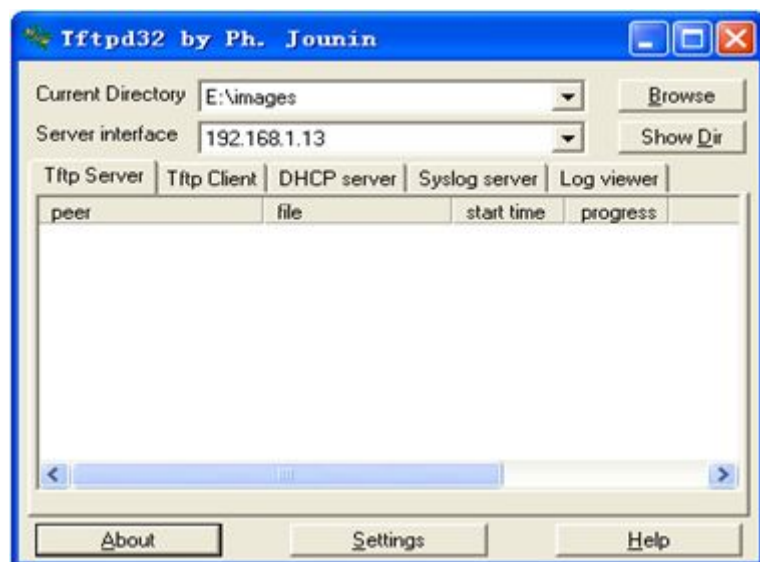


图3.15 设置 Tftpd32的 TFTP 服务

设置完毕后，以后 U-Boot 就可以通过 tftp 命令从 Windows 系统中获取文件了。

3.2 Linux 环境下的工具、命令介绍

如果读者直接在 Linux 环境下工作，可以使用本节介绍的几个工具、命令，它们与4.1节中介绍的工具功能相似，另外：它们是免费的。

3.2.1 代码阅读、编辑工具 KScope

KScope 的作用与 Source Insight 几乎一样，它也是一款源代码阅读、编辑工具。KScope 使用 Cscope 作为源代码的分析引擎，可以为编码人员提供一些有价值的信息，特别适用于使用 C 语言编写的大型项目。

下面依次介绍 KScope 的安装、使用方法。

1. 安装 KScope

确保 Linux 能连上网络，然后使用以下命令进行安装，会得到一个 kscope 命令，并且在 Linux 桌面菜单 “Applications” -> “Programming” 下生成了一个启动项 “KScope”：

```
$ sudo apt-get install kscope
```

要启动 KScope，可以在控制台中运行 kscope 命令，或者点击菜单 “Applications” -> “Programming” -> “KScope”。

2. 建立 KScope 工程

建立 KScope 工程的步骤与建立 Source Insight 工程的步骤相似，也分为这几步：设置工程名，指定工程数据的存放位置，设置支持的文件类型，指定源码的位置，添加、移除源文件，建立数据库。

以内核源码为例，假设内核源码位置为 /work/system/linux-2.6.22.6，将要建立的 KScope 工程名为 linux-2.6.22.6，在 /work/kscope_projects/ linux-2.6.22.6 目录下存放工程数据。

先建立 /work/kscope_projects/linux-2.6.22.6 目录，在控制台中执行以下命令：

```
$ mkdir -p /work/kscope_projects/ linux-2.6.22.6
```

然后启动 KScope 后，点击菜单 “Project” -> “Create Project”，弹出如图3.16所示的对话框。先在

“Detail”选项页中仿照图4. 16分别填入工程名、工程数据的存放位置、源码的位置；然后在“File Types”选项页中仿照图3. 17设置支持的文件类型为：*.c、*.h和*.S；最后点击“Create”按钮得到类似图3. 18的对话框。

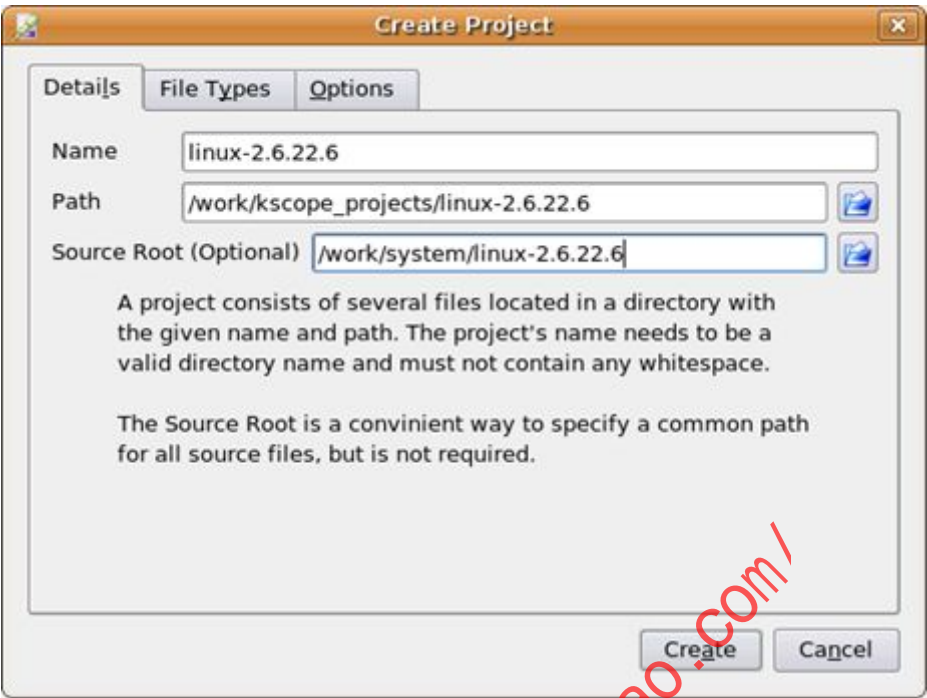


图3. 16 新建一个 KScope 工程

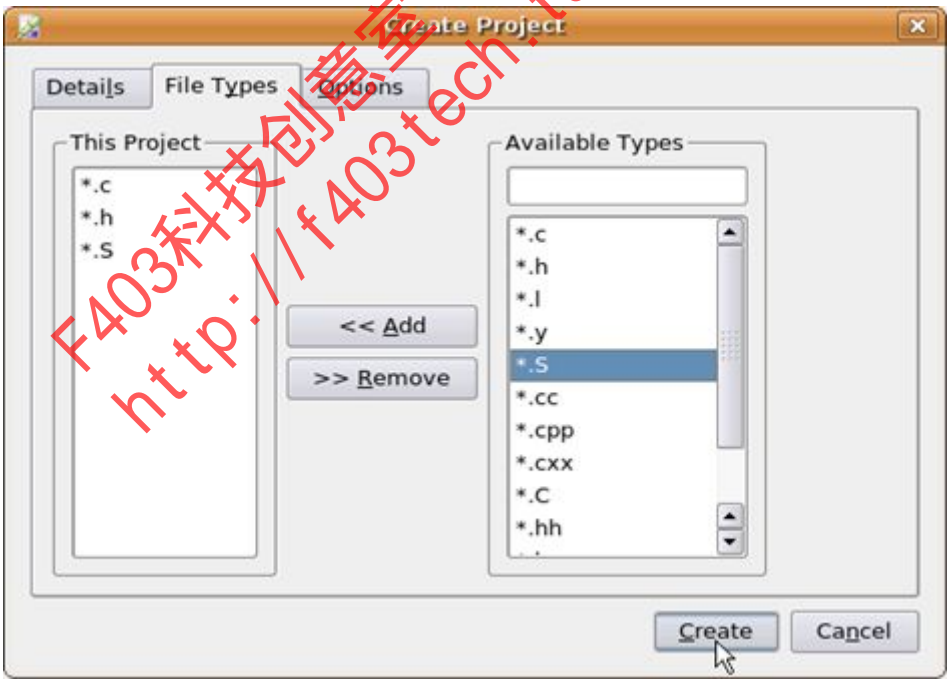


图3. 17 设置 KScope 工程支持的文件类型

类似图3. 18的对话框，也可以通过菜单“Project”->“Add/Remove Files”来启动，在这个对话框中进行源文件的添加、移除。

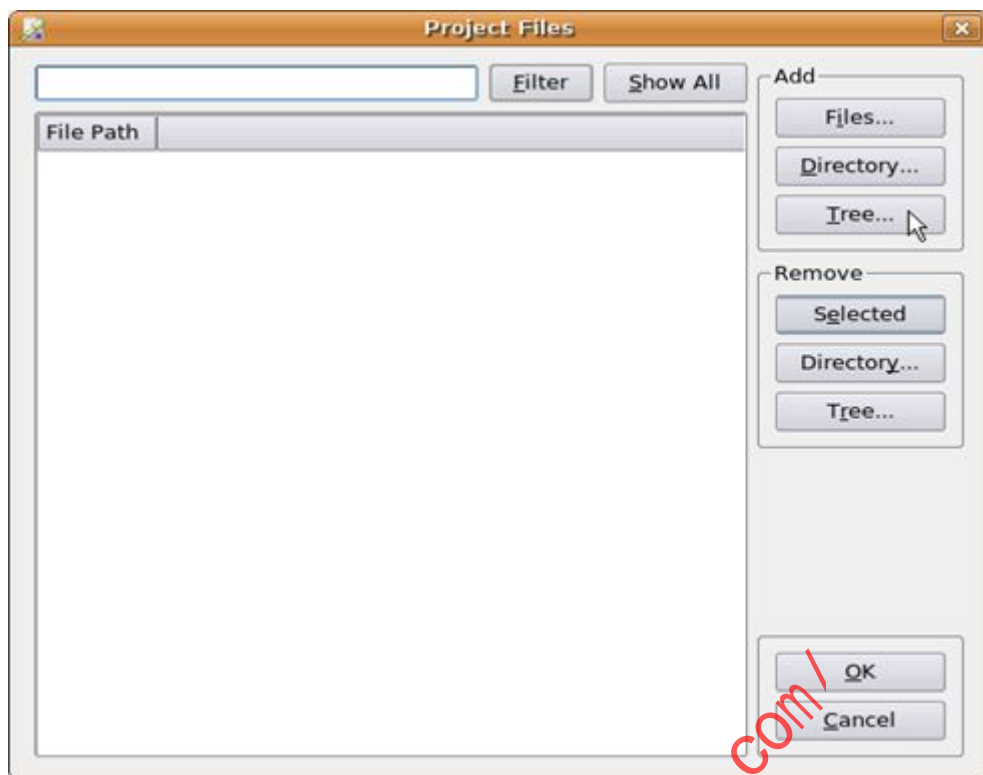


图3.18 添加/移除源文件

图3.18中，“Files”、“Directory”、“Tree”按钮分别表示添加、移除的操作以文件、目录(表示目录下的文件，不包括它的子目录)、整个目录树为单位。移除操作中的“Dircectory”、“Tree”按钮还没有实现，可以在左边的文件框中选择要去除的文件，然后点击“Selected”按钮。为了方便，不妨在建立工程之前先删除不需要的目录、文件，以本书为例，这些不需要目录如下：

- ① arch 目录下除 arm 外的所有子目录；
- ② arch/arm 目录下以“mach-”开头的目录(除 mach-s3c2410、mach-s3c2440之外)；
- ③ arch/arm 目录下以“plat-”开头的目录(除 plat-s3c24xx 之外)；
- ④ include 目录下以“asm-”开头的目录(除 asm-arm、asm-generic 之外)；
- ⑤ include/asm-arm 目录下以“arch-”开头的目录(除 arch-s3c2410之外)。

在图3.18中，点击“Add”框中的“Tree”按钮，弹出如图3.19所示的对话框。在“Folders”框中选择内核的根目录，然后点击“OK”按钮开始添加源文件。

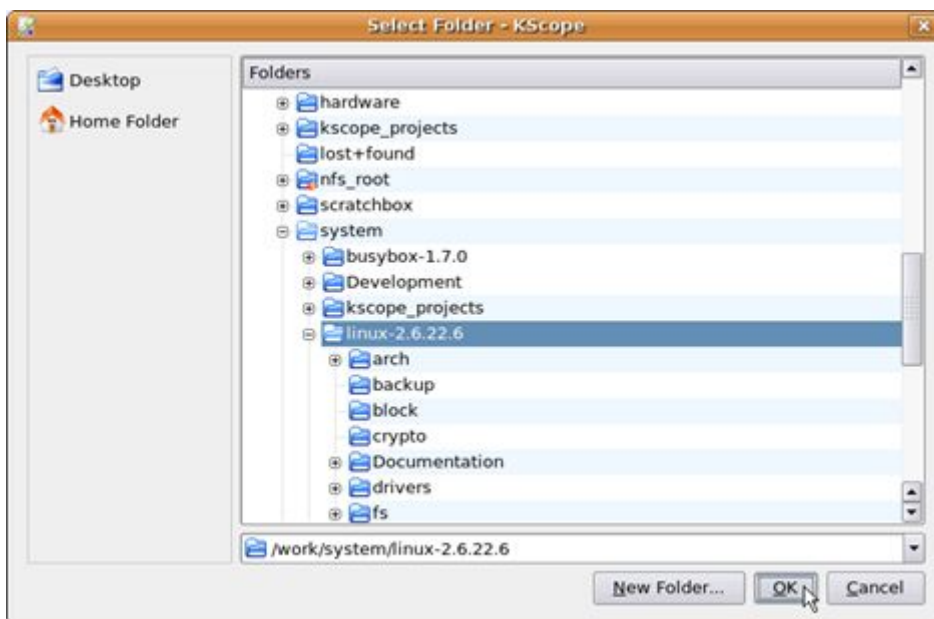


图3.19 添加源文件目录树

源文件添加完毕后，点击图3.18所示界面中的“OK”按钮，KScope 即会自动生成数据库。至此，KScope 工程建立完毕。

3. KScope 工程使用示例

在KScope 右边的文件列表中选择打开 s3c2410fb.c 文件，可以得到如图4.20所示的界面，它的中间是主窗口，可以在里面阅读、编辑代码；左边是“Tag List”（Tag 列表），可以从中快速找到当前文件中的变量、函数，宏定义等；下边是“Query Window”（查询结果窗口），在主窗口中将鼠标放在某个变量、函数、宏上面，然后使用鼠标右键选择某些操作或者按住某些快捷键，会在这个窗口中显示这些操作的结果。

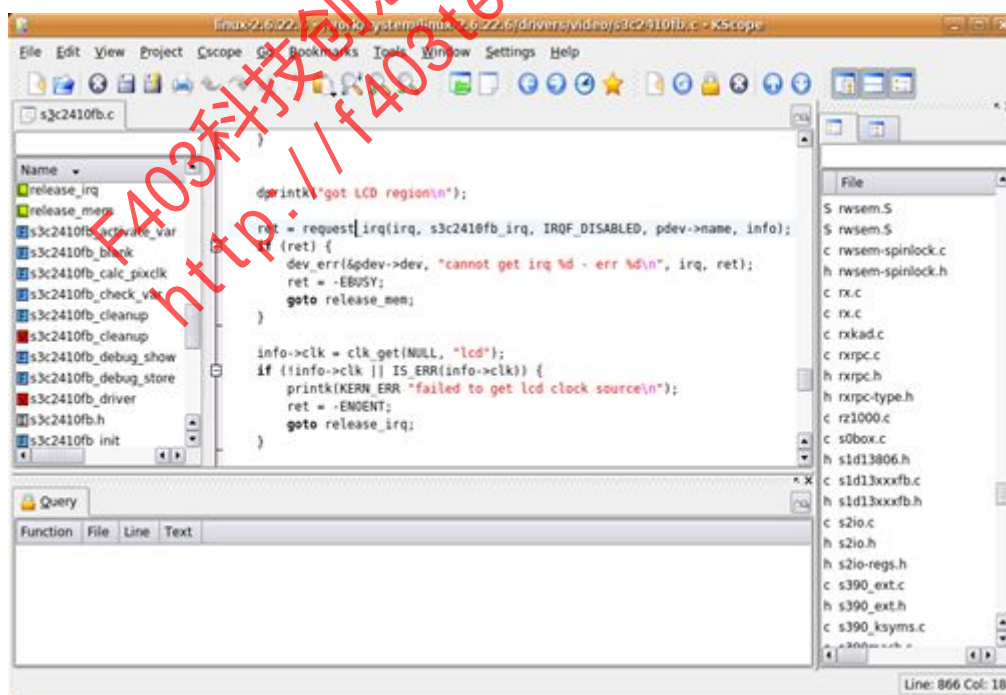


图3.20 KScope 的使用界面

下面介绍一些简单的操作示例。

在主窗口中，用鼠标点击某个函数，比如点击 s3c2410fb.c 中的“request_irq”字样，然后点鼠标右

键，将弹出一个菜单，选择其中的“Cscope”->“Quick Definition”即可快速找到它的定义。将光标移到“request_irq”字样上，然后按下快捷键“Ctrl+]”也可以达到同样的效果。

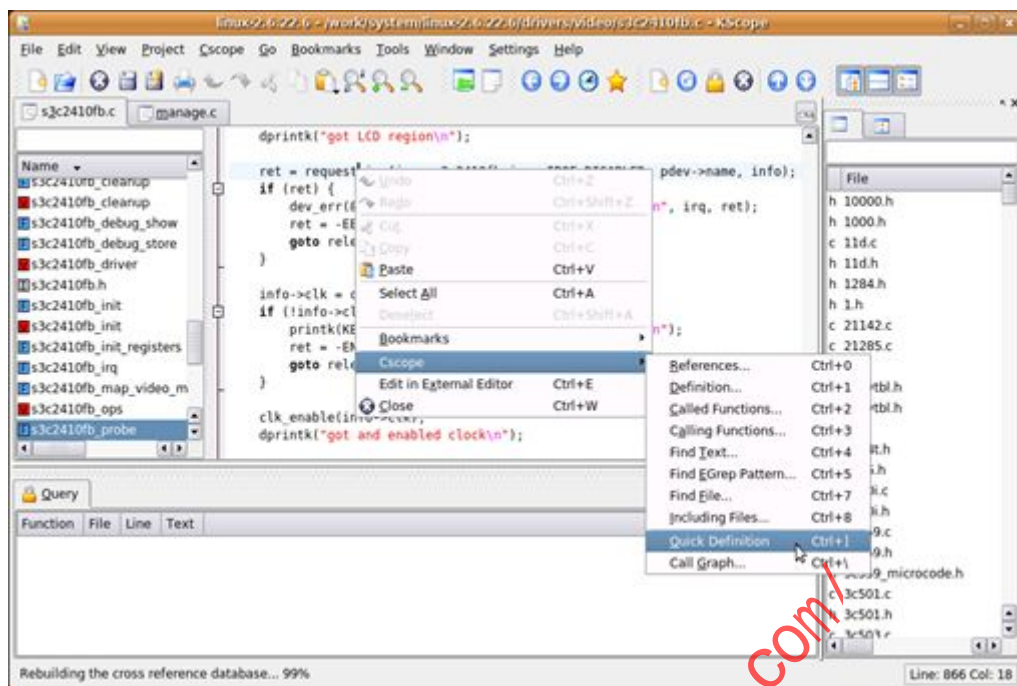


图3.21 在Kscope中查找函数定义

同时按住“Alt”、左箭头可以令主窗口倒退到上一画面，同时按住“Alt”、右箭头可以令主窗口前进到前一个画面。

Kscope 还有很多使用技巧，上面只介绍了几种常用的技巧，读者在使用过程中可以通过各个菜单了解更多。

3.2.2 远程登录工具 C-kermi

C-Kermit 是一款集成了网络通信、串口通信的工具，它有如下多种功能(本书中，只用到它的串口通信功能)：

- 支持 Kermit 文件传输协议
- 自定义了一种脚本语言，它强大而易于使用，可用于自动化工作
- 无论是网络通信、还是串口通信，操作是一致的，并支持多种硬件、软件平台
- 有安全认证、加密功能
- 内建 FTP、HTTP 客户端功能及 SSH 接口
- 支持字符集转换

下面介绍在 Linux 下安装、使用 C-kermi 的方法。

确保 Linux 能连上网络，然后使用以下命令进行安装，会得到一个 kermi 命令：

```
$ sudo apt-get install ckermit
```

使用 kermi 之前，先在 /home/book (假设用户名为 book) 目录下创建一个名为 .kermrc 的配置文件，内容如下：

```
set line /dev/ttyS0
set speed 115200
set carrier-watch off
```

```
set handshake none
set flow-control none
robust
set file type bin
set file name lit
set rec pack 1000
set send pack 1000
set window 5
```

然后，运行“\$ sudo kermit -c”命令即可启动串口；要想关闭串口，先同时按住“Ctrl”和“\”键，然后松开再按“C”键，最后输入“exit”并回车。

在Linux中，可以使用kermit连接串口以操作目标板。

3.2.3 编辑命令 vi

vi命令是字符终端下的一个文本编辑工具。对文本进行少量修改时(当然，使用熟练并且添加了其他插件的话，它的功能堪比Source Insight)，使用vi命令很方便，特别是在使用SecureCRT等工具远程登录Linux时。

vi可以执行输出、删除、查找、替换、块操作等众多文本操作，它没有菜单，只有命令，且命令繁多。

在控制台中输入“vi”或“vi filename”就可以启动vi，后者将打开或新建文件。它有三种基本工作模式：命令行模式、文本输入模式和末行模式。

1. 命令行模式

vi一被启动，它就处于命令行模式；另外，任何时候、任何模式下，只要按一下“Esc”键，即可使vi进入命令行模式。在“命令行模式”下，可以直接使用某些按键完成相应操作。

常用的命令如表4.1所示。

表4.1 vi 中常用的命令

命令(按键)	作用	命令(按键)	作用
光标移动命令			
Ctrl+f	向文件尾翻一屏	Ctrl+b	向文件首翻一屏
n+	光标下移n行(n为数字)	n-	光标上移n行(n为数字)
0(数字零)	光标移至当前行首	\$	光标移至当前行尾
nG	光标移至第n行的行首(n为数字)	:n	光标移至第n行的行首(n为数字)
文本插入命令			
i	在光标前开始插入文本	a	在光标后开始插入文本
o	在当前行之下新开一行	O	在当前行之上新开一行
u	撤销上次对文本的修改		
文本删除命令			
d0	删至行首	d\$或者D	删至行尾

x	删除光标后的一个字 符	X	删除光标前的一个 字符
ndd	删除当前行及其后 n-1行		
搜索及替换命令			
/pattern	从光标开始处向文件 尾搜索 pattern	?pattern	从光标开始处向文 件首搜索 pattern
n	在同一方向重复上一 次搜索命令	N	在反方向上重复上 一次搜索命令
:s/p1/p2 /g	将当前行中所有 p1均 用 p2替代	:n1,n2s/p1 /p2/g	将第 n1至 n2行中所 有 p1均用 p2替代
:g/p1/s/ /p2/g	将文件中所有 p1均用 p2替换		
退出/保存命令			
:w	保存文件	:wq	保存文件并退出 vi
:q	退出 vi	:q!	退出 vi,但是不保存 文件

注:

- (1) “搜索及替换命令”中的“pattern”、“p1”、“p2”表示一个正则表达式，可以用来匹配某些字符串，比如“[0-9][0-9]”表示两位数。通常直接使用字符串，比如使用命令“/lib”在文件中查找“lib”字样。
- (2) “:”开头的命令是“末行模式”中的用法，这里是为了方便读者参考才放在一起。

2. 文本输入模式

在命令模式下输入表4.1中的文本插入命令时，就会进入文本输入模式。在该模式下，用户输入的任何字符都被 vi 当做文件内容保存起来，并在屏幕上显示。在文本输入过程中，按“Esc”键即可回到命令模式。

3. 末行模式

在 vi 中，命令通常只包含几个按键，如表4.1所示；要想输入更长的命令，要进入“末行模式”。在命令模式下，用户按“:”键即可进入末行模式，此时 vi 会在显示窗口的最后一行显示一个“:”作为末行模式的提示符，等待用户输入命令。输入完成后回车，命令即会执行，然后 vi 自动回到命令模式。

末行模式下常用的命令请参考表4.1。

3.2.4 查找命令 grep、find 命令

在 Linux 下，常用 grep 命令列出含有某个字符串的文件，常用 find 命令查找匹配给定文件名的文件。

1. grep 命令

grep 命令的用法为：

```
grep [options] PATTERN [FILE...]
```

以几个例子介绍它的常用格式：

- (1) 在内核目录下查找包含“request_irq”字样的文件：

```
$ cd /work/system/linux-2.6.22.6/
// *表示查找当前目录下的所有文件、目录，-R 表示递归查找子目录
$ grep "request_irq" * -R
```

- (2) 在内核的 kernel 目录下查找包含“request_irq”字样的文件：

```
$ cd /work/system/linux-2.6.22.6/
```

```
// kernel 表示在当前目录的 kernel 子目录下查找，-R 表示递归查找它的所有子目录
$ grep "request_irq" kernel -R
```

2. find 命令

find 命令的用法为：

```
find [-H] [-L] [-P] [path...] [expression]
```

以几个例子介绍它的常用格式：

(1) 在内核目录下查找文件名中包含“fb”字样的文件：

```
$ cd /work/system/linux-2.6.22.6/
$ find -name "*fb*"
```

(2) 在内核的 drivers/net 目录下查找文件名中包含“fb”字样的文件：

```
$ cd /work/system/linux-2.6.22.6/
$ find drivers/net -name "*fb*"           // “drivers/net” 必须是 find 命令的第一个参数
```

依照上面介绍的 grep、find 命令的使用例子，基本可以满足在 Linux 下对代码、文件的查找工作。

3.2.5 在线手册查看命令 man

Linux 中包含了种类繁多的在线手册，从各种命令、各种函数的使用，到一些配置文件的设置，等等。可以使用 man 命令查看这些手册，比如执行“man grep”命令即可看到 grep 命令的使用方法。

man 命令的基本用法为：

```
man [section] name
```

其中的“section”被称为区号，当直接使用“man name”命令没有查到需要的手册时，可以指定区号。比如想查看“open”函数的用法，使用“man open”命令得到的却是一个名为 openvt 的程序的用法，这时可以使用“man 2 open”命令，表示要查看第2区（它表示系统调用）中的手册。

Linux 在线手册按照区号进行分类，如表4.2所示。

表4.2 Linux 在线手册的区号及类别

区号	类别
1	命令，比如 ls、grep、find 等
2	系统调用，比如 open、read、socket 等
3	库调用，比如 fopen、fread 等
4	特殊文件，比如/dev/目录下的文件等
5	文件格式和惯例，比如/etc/passwd 等
6	游戏
7	其他
8	系统管理命令，类似 mount 等只有系统管理员才能执行的命令
9	内核例程(这个区号基本没被使用)

最后介绍使用 man 命令的阅读技巧，即启动 man 命令后，可以通过一些热键进行翻页等操作，如表4.3所示。

表4.3 man 命令的热键

热 键	作用
h	显示帮助信息
j	前进一行
k	后退一行
空 格或 f	向前翻页
b	向后翻页
g	跳转到手册的第一行
G	跳转到手册的最后一行
?st ring	向后搜索字符串 string
/st ring	向前搜索字符串 string
r	刷屏
q	退出

3.2.6 其他命令: tar、diff、patch

1. tar 命令

tar 命令具有打包、解包、压缩、解压缩等4种功能，在本书中使用的频率很高。它常用的压缩、解压缩方式有两种: gzip、bzip2。一般而言，以“.gz”、“z”结尾的文件是用 gzip 方式进行压缩的，以“.bz2”结尾的文件是用 bzip2 方式进行压缩的，后缀名中有“tar”字样时表示这是一个文件包。

tar 命令有5个常用的选项：

- ① “c”：表示创建，用来生成文件包；
- ② “x”：表示提取，从文件包中提取文件。
- ③ “z”：使用 gzip 方式进行处理，它与“c”结合就表示压缩，与“x”结合就表示解压缩；
- ④ “j”：使用 bzip2 方式进行处理，它与“c”结合就表示压缩，与“x”结合就表示解压缩；
- ⑤ “f”：表示文件，后面接着一个文件名。

以例子说明 tar 命令的使用方法：

- ① 将某个目录 dirA 整个地制作为压缩包：

```
$ tar czf dirA.tar.gz dirA      // 将目录 dirA 压缩为文件包 dirA.tar.gz，以 gzip 方式进行压缩
$ tar cjf dirA.tar.bz2 dirA     // 将目录 dirA 压缩为文件包 dirA.tar.bz2，以 bzip2 方式进行压缩
```

- ② 将某个压缩包文件 dirA.tar.gz 解开：

```
$ tar xzf dirA.tar.gz          // 在当前目录下解开 dirA.tar.gz，先使用 gzip 方式解压缩，然后解包
$ tar xjf dirA.tar.bz2         // 在当前目录下解开 dirA.tar.bz2，先使用 bzip2 方式解压缩，然后解包
$ tar xzf dirA.tar.gz -C <dir> // 将 dirA.tar.gz 解开到<dir>目录下
$ tar xjf dirA.tar.bz2 -C <dir> // 将 dirA.tar.bz2 解开到<dir>目录下
```

2. diff、patch 命令

diff 命令常用来比较文件、目录，也可以用来制作补丁文件——所谓“补丁文件”就是“修改后的文件”与“原始文件”的差别。

常用的选项有：

- ① “-u”：表示在比较结果中输出上下文中一些相同的行，这有利于人工定位；
- ② “-r”：表示递归比较各个子目录下的文件；
- ③ “-N”：将不存在的文件当作空文件；
- ④ “-w”：忽略对空格的比较；
- ⑤ “-B”：忽略对空行的比较。

例如：假设 linux-2.6.22.6 目录中是原始的内核，linux-2.6.22.6_ok 目录中是修改过的内核，可以使用以下命令制作补丁文件 linux-2.6.22.6_ok.diff (原始目录在前，修改过的目录在后)：

```
$ diff -urNwB linux-2.6.22.6 linux-2.6.22.6_ok > linux-2.6.22.6_ok.diff
```

由于 linux-2.6.22.6 是标准的代码，可以从网上自由下载，要发布 linux-2.6.22.6_ok 中所做的修改时，只需要提供补丁文件 linux-2.6.22.6_ok.diff (它通常是很小的)。

patch 命令被用来打补丁——就是依据补丁文件来修改原始文件。比如对于上面的例子，可以使用以下命令将补丁文件 linux-2.6.22.6_ok.diff 应用到原始目录 linux-2.6.22.6 上去。假设 linux-2.6.22.6_ok.diff 和 linux-2.6.22.6 位于同一个目录下。

```
$ cd linux-2.6.22.6
$ patch -p1 < ../linux-2.6.22.6_ok.diff
```

patch 命令中最重要的选项是“-pn”：补丁文件中指明了要修改的文件的路径，“-pn”表示忽略路径中第“n”个斜线之前的目录。假设 linux-2.6.22.6_ok.diff 中有如下几行：

```
diff -urNwB linux-2.6.22.6/A/B/C.h linux-2.6.22.6_ok/A/B/C.h
--- linux-2.6.22.6/A/B/C.h 2007-08-31 02:21:01.000000000 -0400
+++ linux-2.6.22.6_ok/A/B/C.h 2007-09-20 18:11:46.000000000 -0400
.....
```

使用上述命令打补丁时，patch 命令根据“linux-2.6.22.6/A/B/C.h”寻找源文件，“-p1”表示忽略第1个斜线之前的目录，所以要修改的源文件是当前目录下的：A/B/C.h。

注意：

- (5) 此教程为 F403 科技创意室版权所有；
- (6) 此 openwrt 教程不不断更新，请关注我们的淘宝店：

<http://f403tech.taobao.com/>

向客服索要最新版本的教程。

- (3) 针对该教程，我们特意创建了 QQ 超级群，由教程的作者亲自指导大家学习。详情请咨询客服。

第 4 章 下载配置编译 openwrt

本章目标

- 掌握 openwrt 源码的下载方法
- 掌握 openwrt 的配置方法
- 掌握 openwrt 的编译方法
- 掌握如果下载固件到开发板的方法

4.1 安装相关工具、库

(1) 安装 SVN 工具

安装 SVN 工具，用于下载 openwrt 源码：

```
$ sudo apt-get install subversion
```

(2) 安装 git 工具

```
$ sudo apt-get install git-core
```

(3) 安装依赖的库文件

```
$ sudo apt-get install gcc g++ binutils patch bzip2 flex bison make autoconf gettext texinfo unzip sharutils  
ncurses-term zlib1g-dev libncurses5-dev gawk
```

4.2 下载 openwrt 源码

编译 openwrt 首先需要下载源码，openwrt 的源码是通过 svn 下载，openwrt 目前已经很好的支持了 RT5350 芯片，openwrt 在 F403 科技创意室_RT5350 开发板也一样。下面开始 openwrt 学习之旅。

新建一个 openwrt 文件夹，使用 svn 下载源码

```
$ mkdir openwrt  
$ cd openwrt/  
$ svn co svn://svn.openwrt.org/openwrt/trunk
```

等待下载即可。

```
A trunk/tools/libtool/files/libtool-v2.4.patch  
A trunk/tools/libtool/patches  
A trunk/tools/libtool/patches/000-relocatable.patch  
A trunk/tools/libtool/patches/100-libdir-fixes.patch  
A trunk/tools/libtool/patches/200-openwrt-branding.patch  
A trunk/tools/libtool/patches/110-dont-use-target-dir-for-relinking.patch  
A trunk/tools/libtool/patches/120-strip-unsafe-dirs-for-relinking.patch  
A trunk/tools/libtool/patches/001-fix-func_append.patch  
A trunk/tools/libtool/patches/150-trailingslash.patch  
A trunk/tools/libtool/Makefile  
A trunk/.gitignore  
A trunk/config  
A trunk/config/Config-build.in  
A trunk/config/Config-devel.in  
A trunk/config/Config-kernel.in  
A trunk/config/Config-images.in  
U trunk  
checked out revision 40773.  
f403tech@f403tech-pc:/work/openwrt$
```

图 4.1 openwrt 下载过程

当出现图 4.1 所示界面时，这样就下载完成。

下载完 openwrt 的源码后，为了使 openwrt 支持更多的软件，需要更新和安装其他源上面的软件：

```
$ cd trunk
$ ./scripts/feeds update -a
$ ./scripts/feeds install -a
```

一切就绪，开始编译，编译的时候需要注意，openwrt 默认是用非 root 用户名来编译的，如果你是 root 用户名需要在配置上面做点修改。需要修改 include/prereq-build.mk

```
$ gedit include/prereq-build.mk
```

把第十六行的注释掉

```
define Require/non-root
# [ "$(shell whoami)" != "root" ]
endif
```

注意：

对于我们 Ubuntu 系统，并不需要修改 prereq-build.mk。

4.3 编译 openwrt 源码

编译的时候首先执行 make menuconfig，和配置内核的命令时一样的，不出意外的话，它就进入了一个比较熟悉的界面，如图 4.2 所示，配置过内核的朋友应该不会陌生。

```
$ make menuconfig
```

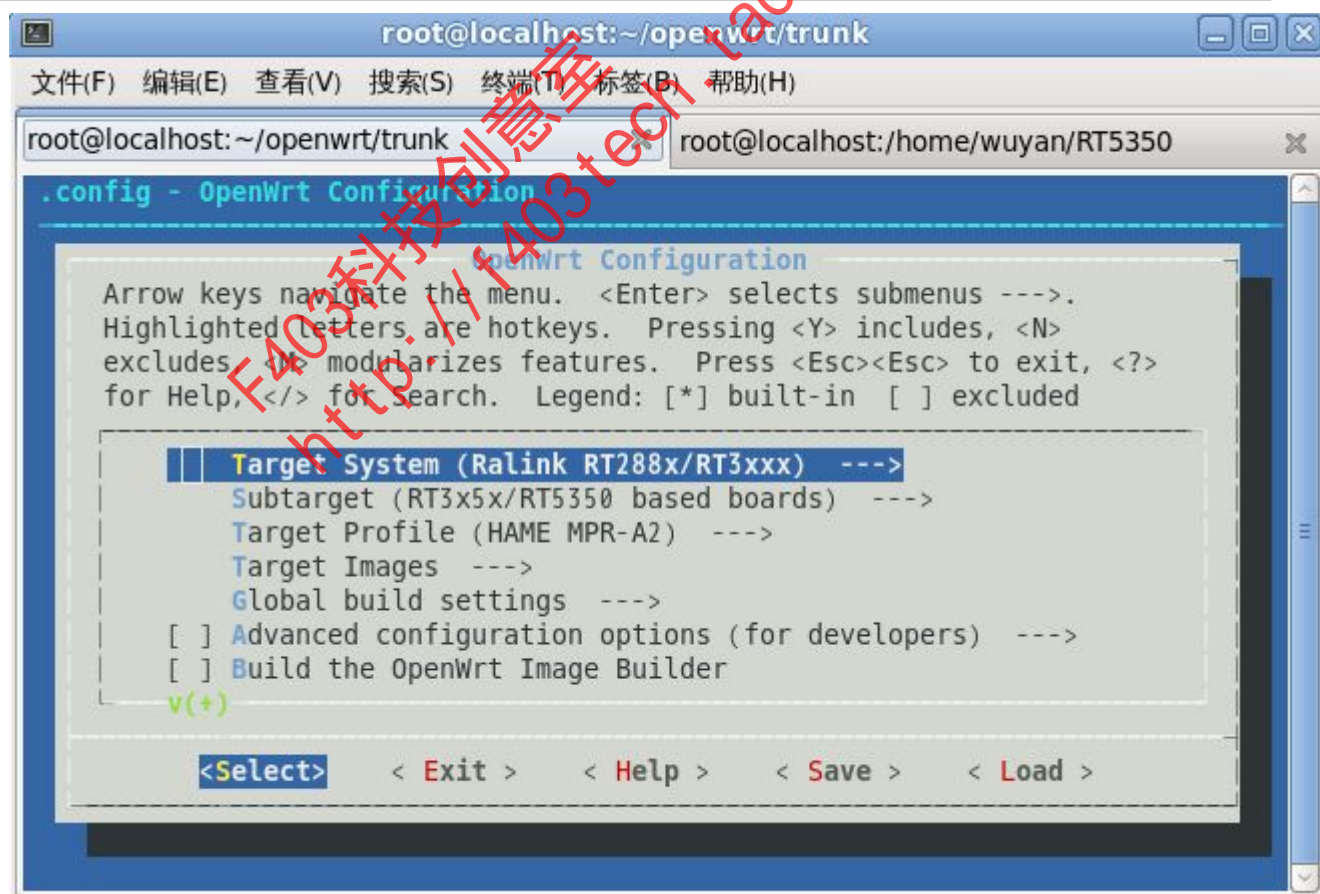


图 4.2 配置界面

F403 科技创意室研发的 RT5350 开发板(32M 内存、8M Flash)，和 HAME 的 HAME-MPR-A2 硬件配置是一样的。因此配置如下：

1、选择 CPU 型号

Target System —> Ralink RT288x/RT3xxx

2、选择 CPU 子型号

Subtarget —> RT3x5x/RT5350 based boards

3、选择具体路由器型号

Target profile —> HAME-MPR-A2

然后点击 save 保存配置，这样一个配置就算完成了，当然这个是最简单的配置，我们首先是要在开发板把程序系统跑起来，复杂的配置以后的课程会一点一点的教会大家。

最后执行 make V=99，开始漫长的编译，编译时间由电脑配置决定，由于编译时需要的软件是从网络上下载，所以这里机子必须能够上网。V=99 意思显示所有信息。

```
$ make V=99
```

大概 4、5 个小时编译就会完成，然后在源码目录 bin 下面生成镜像。

```
$ ls bin/ramips/  
md5sums  
openwrt-ramips-rt305x-MPRA2-rootfs.tar.gz  
openwrt-ramips-rt305x-mpr-a2-squashfs-sysupgrade.bin  
openwrt-ramips-rt305x-root.squashfs  
openwrt-ramips-rt305x-uImage.bin  
openwrt-ramips-rt305x-vmlinux.bin  
openwrt-ramips-rt305x-vmlinux.elf  
packages
```

其中 openwrt-ramips-rt305x-mpr-a2-squashfs-sysupgrade.bin 这个就是我们要的镜像。烧写到板子上面即可启动。这样最简单 openwrt 就可以启动了，不过好多功能都没有，一步一步来，下一讲介绍移植 web 界面。

4.4 调试串口连接和系统烧写

4.4.1 调试串口连接

F403 科技创意室研发的 RT5350 开发板已经板载了 USB 转串口芯片，因此，只需通过 mini USB 线将开发板和电脑连接起来，便可以使用串口调试功能。

(1) 安装 USB 转串口驱动

F403 科技创意室研发的 RT5350 开发板串口对应的驱动为“PL2303_Prolific_DriverInstaller_v190.exe”，安装过程很简单，直接采用默认设置即可。



图 4.3 USB 转串口驱动安装过程



图 4.4 USB 转串口驱动安装过程

(2) 连接开发板、电脑

F403 科技创意室研发的 RT5350 开发板已经板载了 USB 转串口芯片，因此，只需通过 mini USB 线将开发板和电脑连接起来，便可以使用串口调试功能。

当开发板与电脑连接上时，电脑便会自动识别出 RT5350 开发板的串口，然后我们便可以通过“我的电脑->属性->硬件->设备管理”查看出串口对应的串口号，如图 4.5 所示：



图 4.5 串口号查找

(3) 使用 SecureCRT 看串口调试信息

打开 SecureCRT，单击如图 4.6 所示的“快速连接”按钮(被红框框起的)。

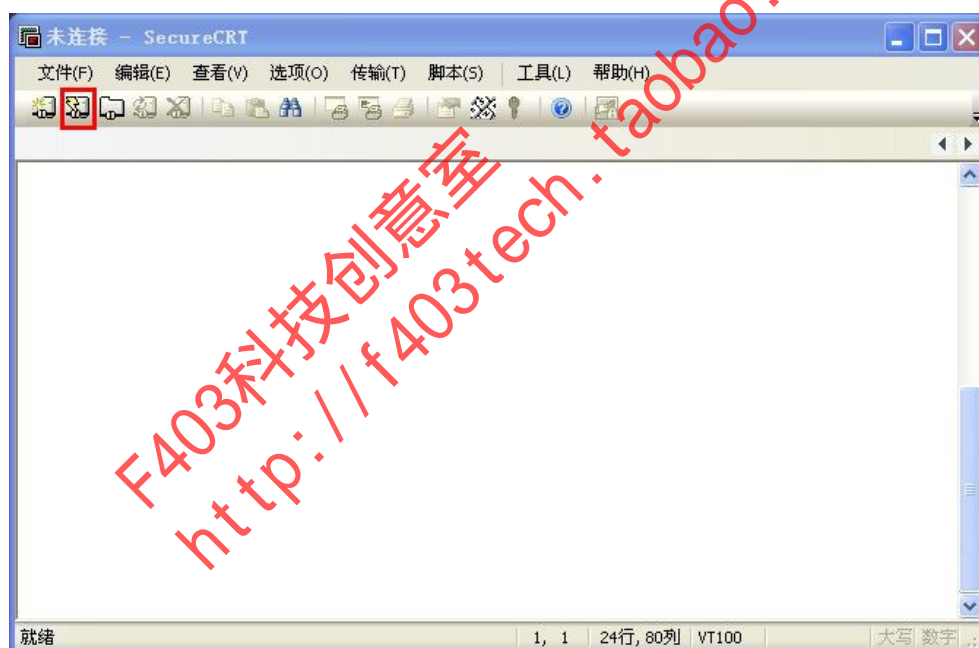


图 4.6 启动 SecureCRT

紧接着在如图 4.7 所示的界面中新建一个串口终端，串口号为 COM3(根据自己的情况确定)，波特率为 57600，8N1，单击“连接”，串口终端便创建好了。最后，开发板上电，我们便可以看到串口调试信息了，如图 4.8 所示。



图 4.7 创建串口终端

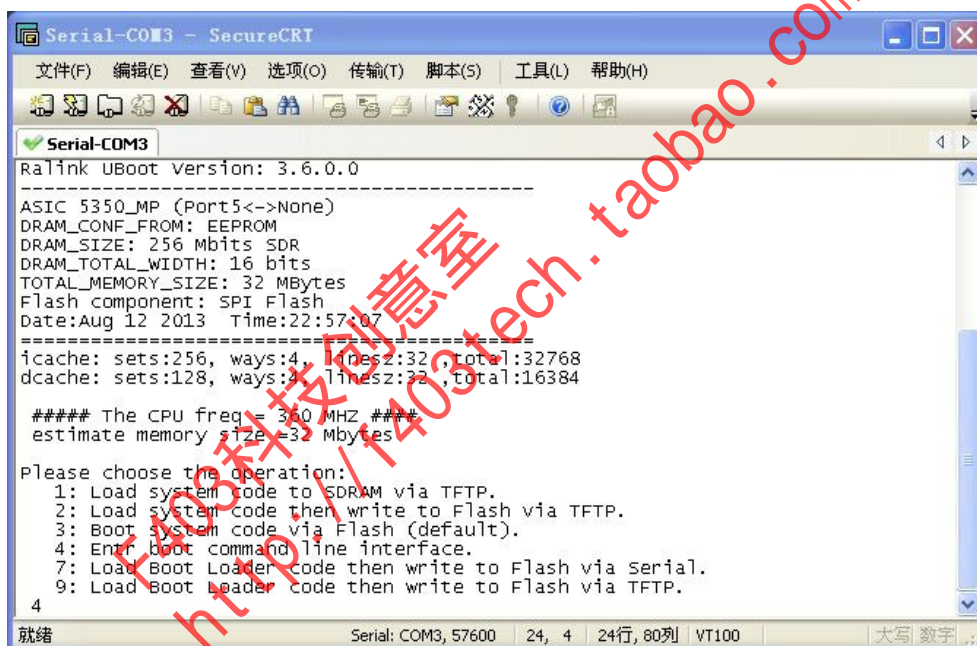


图 4.8 串口调试信息

4.4.2 烧写 openwrt 系统

接下来就给大家讲解，如何烧写 4.2 节中编译出来的系统。

(1) 通过 FileZilla 将“openwrt-ramips-rt305x-mpr-a2-squashfs-sysupgrade.bin”从 Ubuntu 12.04 中拖到 Windows 的“D:\tftp”目录。如图 4.9 所示：

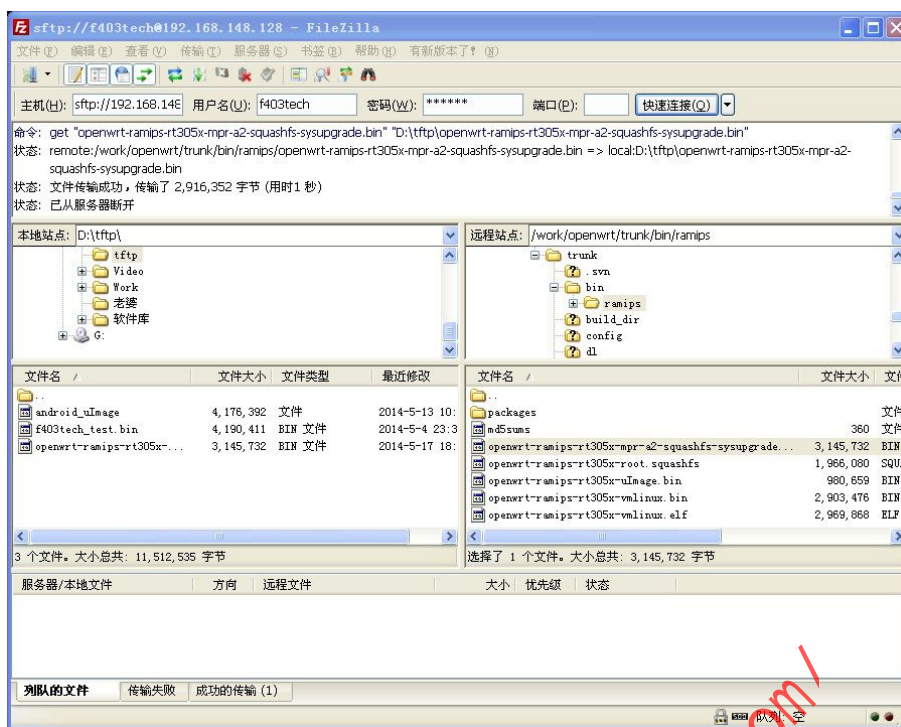


图 4.9

(2) 重启开发板，在 U-boot 的操作菜单中选择 2。如图 4.10 所示：

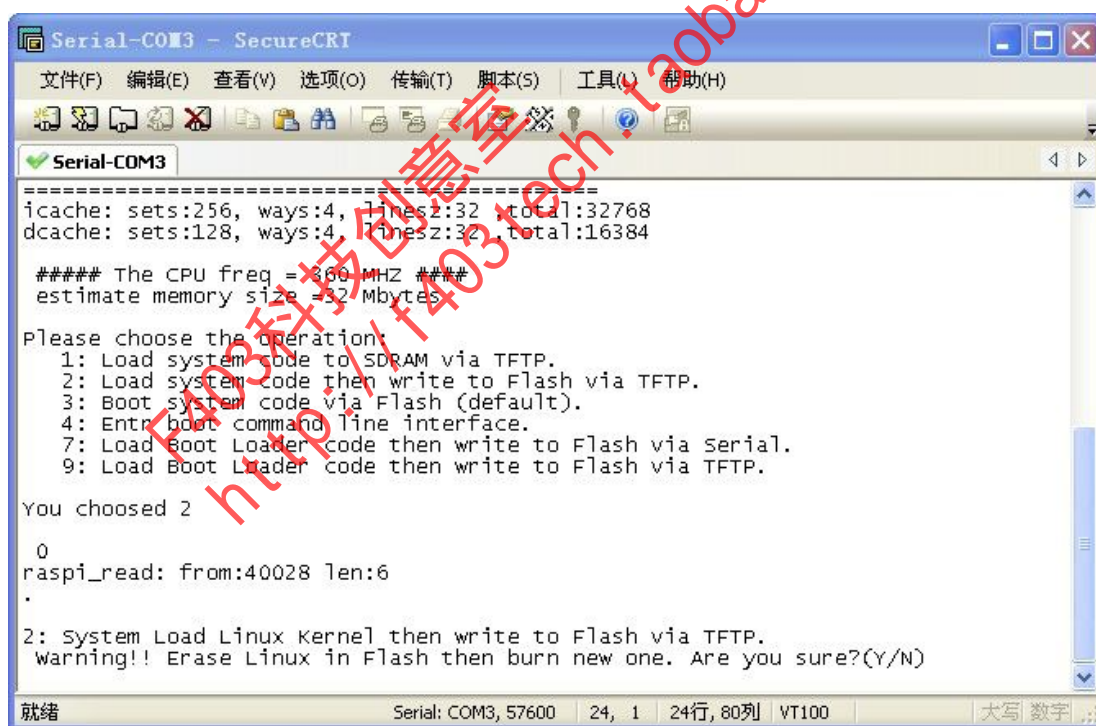


图 4.10

(3) 用网线将开发板的网口(任何一个都行)和电脑的网口连接起来。启动 tftpd32，将“Current Directory”设置为要下载的系统所在目录，这里为“D:\tftp”，注意一定不能有中文路径；将“Server interface”选为有线网卡的 IP。如图 4.11 所示：

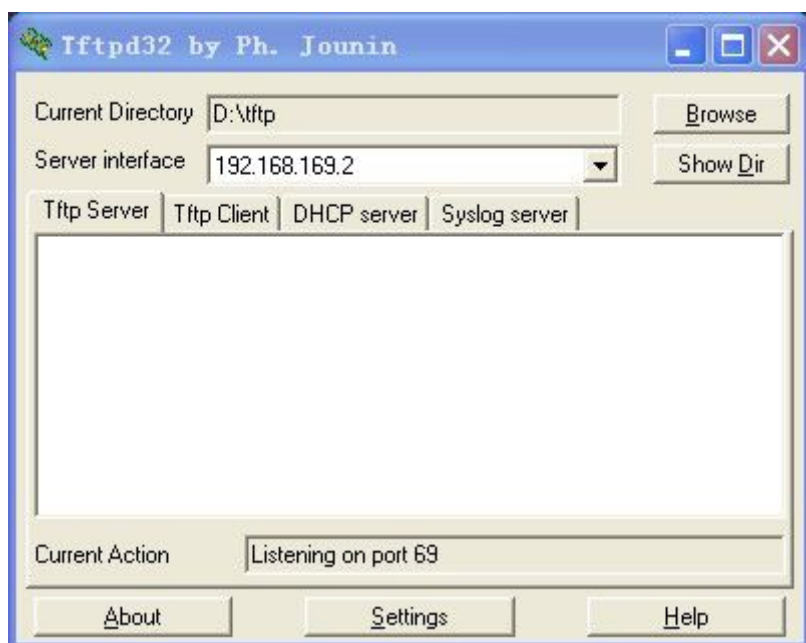


图 4.11

- (4) 回到第二步，在图 4.10 所示界面中，输入“y”，接下来在如图 4.12 所示界面中。设置如下：
- device IP:192.168.169.7 // 该 IP 和有线网卡的 IP 同一个网段即可
- server IP:192.168.169.2 // 有线网卡的 IP
- Filename:openwrt-ramips-rt305x-mpr-a2-squashfs-sysupgrade.bin // 系统名称

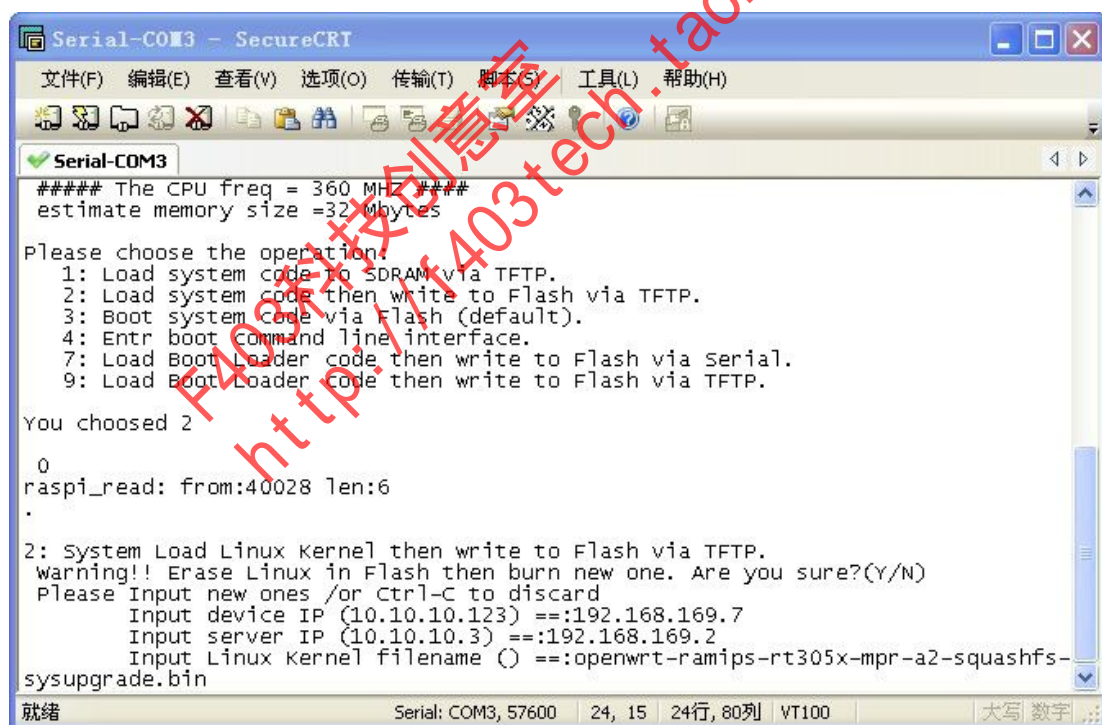
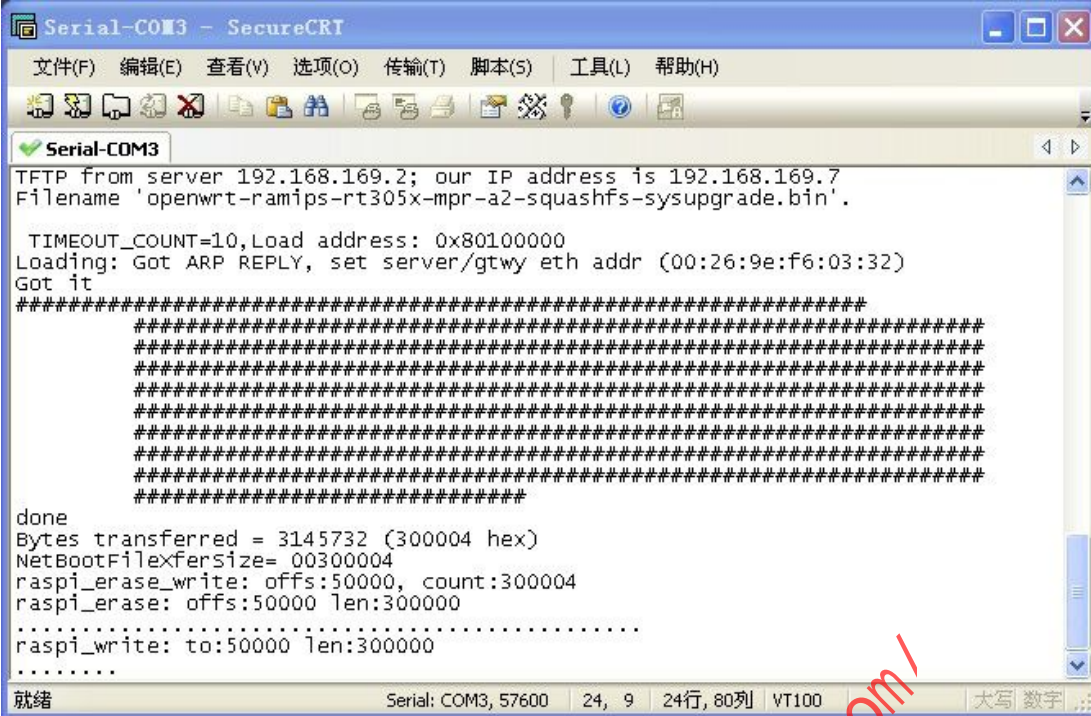


图 4.12

最后回车，便会出现如图 4.13 所示界面，便开始下载了，当系统下载成功后，便会自动运行该系统，到此，我们自己移植的 openwrt 系统，便正常运行起来了。当然，还有很多功能没有实现，再后继的章节中，给大家讲解。



```
Serial-COM3 - SecureCRT
文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 帮助(H)

Serial-COM3
TFTP from server 192.168.169.2; our IP address is 192.168.169.7
Filename 'openwrt-ramips-rt305x-mpr-a2-squashfs-sysupgrade.bin'.

TIMEOUT_COUNT=10,Load address: 0x80100000
Loading: Got ARP REPLY, set server/gtwy eth addr (00:26:9e:f6:03:32)
Got it
#####
#####
#####
#####
#####
#####
#####
#####
#####
#####
done
Bytes transferred = 3145732 (300004 hex)
NetBootFilexfersize= 00300004
raspi_erase_write: offs:50000, count:300004
raspi_erase: offs:50000 len:300000
.....
raspi_write: to:50000 len:300000
.....

就绪 Serial: COM3, 57600 24, 9 24行, 80列 VT100 大写 数字
```

图 4.13

注意：

(7) 此教程为 F403 科技创意室版权所有；

(8) 此 openwrt 教程不不断更新，请关注我们的淘宝店：

<http://f403tech.taobao.com/>

向客服索要最新版本的教程。

(3) 针对该教程，我们特意创建了 QQ 超级群，由教程的作者亲自指导大家学习。详情请咨询客服。

第 5 章 物联网 openwrt 开发之 VLAN 设置

本章目标

- 了解 VLAN 概念
- 灵活配网络置端口为 WAN 或者 LAN

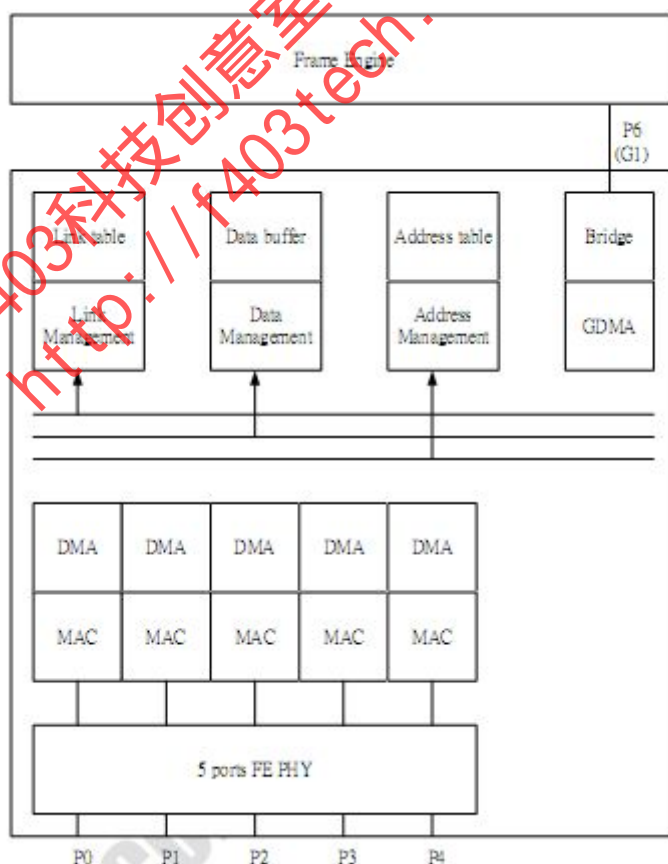
5.1 VLAN 介绍

5.1.1 VLAN 介绍

VLAN (Virtual Local Area Network) 的中文名为“虚拟局域网”。VLAN 是一种将局域网设备从逻辑上划分成一个个网段，从而实现虚拟工作组的新兴数据交换技术。这一新兴技术主要应用于交换机和路由器中，但主流应用还是在交换机之中。但又不是所有交换机都具有此功能，只有 VLAN 协议的第二层以上交换机才具有此功能，这一点可以查看相应交换机的说明书即可得知。

5.1.2 RT5350 开发板 VLAN 支持

我们的设备具有 vlan 功能，具体可参考 rt5350 的数据手册。设备具有 5 个端口，这些端口基于 vlan 技术，符合 802.1q 标准。下图是 rt5350 的 valn 示意图。



Vlan 的作用是划分不同的网络，每个网络之间都不互通。Rt5350 有 5 个端口，可以最多划分为 5 个不同的网络。每个网络即可以做 wan，也可以做 lan，一般是一个端口做 wan 口，4 个端口做 lan 口。我们的板子

上面引出了 rt5350 的 P3 端口和 P4 端口，可以把他们划分为两个不同的网络，一个做为 lan，一个作为 wan。也可以把两个都作为 lan 口。

5.2 灵活配置端口为 WAN、LAN

5.2.1 openwrt 中的 vlan

Openwrt 对于 rt5350 主机实际上有三个设备在里面：一个可配置 VLAN 的交换机、一个无线端口、一个 linux 主机。交换机和 linux 主机被内部的电路所连接，通过 VLAN 标记的包进行数据交换。主机上所有的物理端口都在单一的交换机上。VLAN 分隔这些端口到不同的组。

Openwrt 中配置文件在 /etc/config 下面，对于 vlan 设置是在 /etc/config/network 中，通过增加一个 switch 配置段来添加 vlan 的支持。因为 rt5350 在 openwrt 上面网络接口名字为 eth0，但是 switch 驱动不会检测任何物理设备，为了适用网卡，必须配置 vlan，每一个 vlan 设备必须有一个 vlan<n> 其中 n 为一个 vlan 的编号。因此你如果想有 2 个 vlan 的话，你应该有像下面的配置。在 rt5350 中 cpu 端口默认为 6。

```
1 config switch
2     option name 'rt305x'
3     option reset '1'
4     option enable_vlan '1'
5 config switch_vlan
6     option device 'rt305x'
7     option vlan '1'
8     option ports '0 1 2 3 6t'
9 config switch_vlan
10    option device 'rt305x'
11    option vlan '2'
12    option ports '4 6t'
```

其中第 4 行代表开启 vlan，6, 7, 8 行为 vlan0 的设置，10, 11, 12 行为 vlan1 的设置。

这样配置的话重启网络会自动生成 2 个设备接口，eth0.0 (vlan0)，eth0.1 (vlan1)，这样你可以配置这两个 valn 做为 lan 口或者 wan 口。其中端口 0，端口 1，端口 2，端口 3 都属于 vlan0，端口 4 属于 vlan1。

5.2.2 RT5350 开发板 VLAN 默认配置

对于本开发板，它的默认 vlan 设置如下：

```
1 config interface 'lan'
2     option ifname 'eth0.1'
3     option force_link '1'
4     option type 'bridge'
5     option proto 'static'
6     option ipaddr '192.168.1.1'
7     option netmask '255.255.255.0'
8     option ip6assign '60'
9 config interface 'wan'
10    option ifname 'eth0.2'
11    option proto 'dhcp'
12    option macaddr ':
13 config switch
14    option name 'rt305x'
```



```

15     option reset '1'
16     option enable_vlan '1'
17 config switch_vlan
18     option device 'rt305x'
19     option vlan '1'
20     option ports '0 1 2 3 6t'
21 config switch_vlan
22     option device 'rt305x'
23     option vlan '2'
24     option ports '4 6t'

```

从默认配置可以看出，端口 0、1、2、3 属于 vlan0，端口 4 属于 vlan1；第 2 行则说明是将 vlan0 设置为 lan，则端口 0、1、2、3 为 LAN；第 10 行则说明将 vlan1 设置为 WAN，则端口 4 为 WAN。

因为板开发板仅适用 vlan 端口 3 和端口 4，可以把 option ports '0 1 2 3 6t' 这一行的 0, 1, 2 去掉，这样就配置个两个 vlan，端口 3 作为第一个 vlan，端口 4 作为第 2 个 vlan，这样网络环境就配置好了。端口 3 为 LAN，端口 4 为 WAN。

5.2.3 全部设置为 LAN

如果想将开发板端口 3、端口 4 全部设置成 LAN 口，则对 /etc/config/network 做如下修改。

(1) 将

```
20     option ports '0 1 2 3 6t'
```

改为

```
20     option ports '3 4 6t'
```

(2) 将

```

21 config switch_vlan
22     option device 'rt305x'
23     option vlan '2'
24     option ports '4 6t'

```

给注释掉。

(3) 将

```

9  config interface 'wan'
10     option ifname 'eth0.2'
11     option proto 'dhcp'
12     option macaddr :

```

给注释掉。

注意：

(9) 此教程为 F403 科技创意室版权所有；

(10) 此 openwrt 教程不不断更新，请关注我们的淘宝店：

<http://f403tech.taobao.com/>

向客服索要最新版本的教程。

(3) 针对该教程，我们特意创建了 QQ 超级群，由教程的作者亲自指导大家学习。详情请咨询客服。

第 6 章 物联网 openwrt 开发之 Web 界面 Luci 移植

6.1 WEB 界面移植

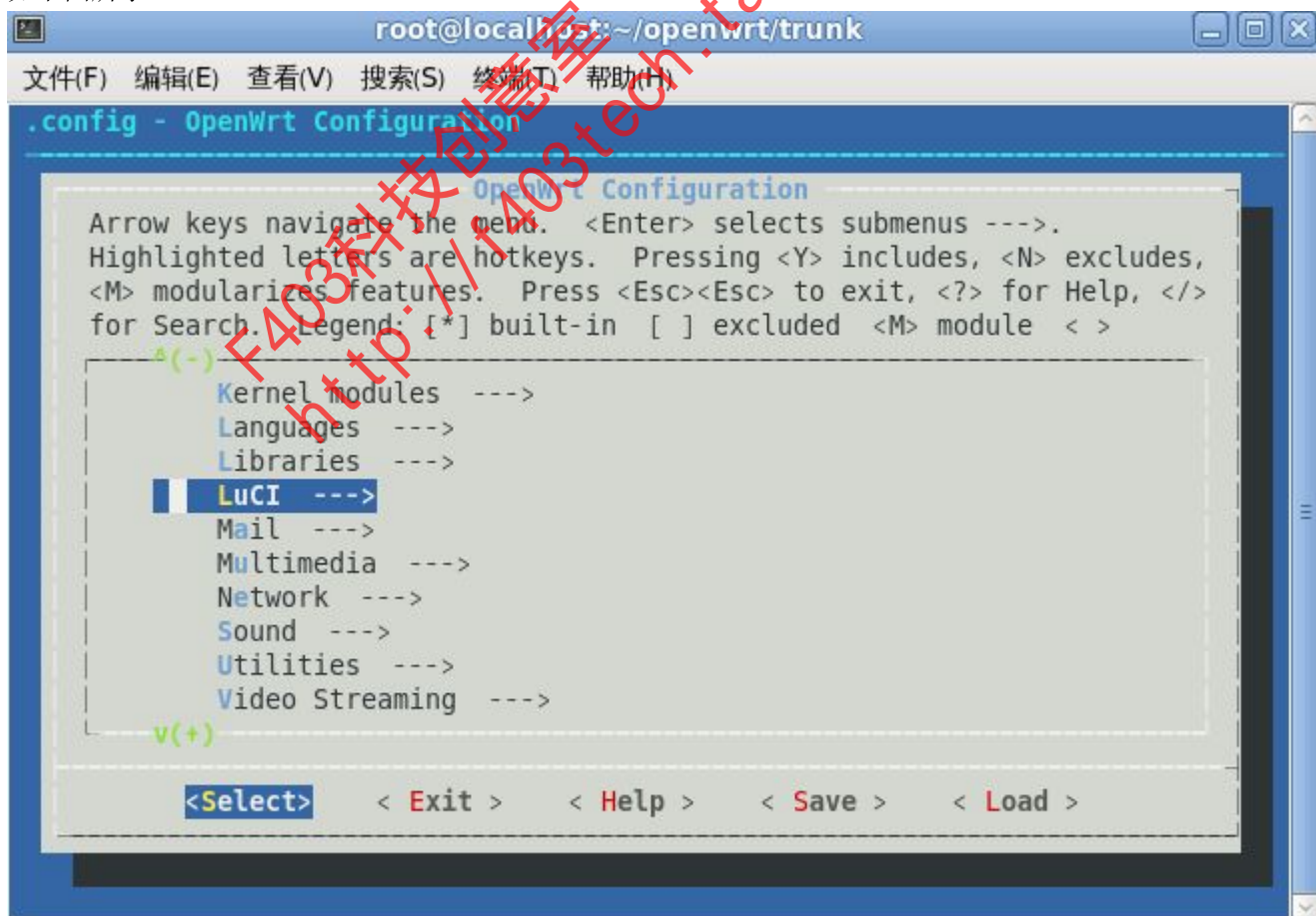
大部分路由器的提供了 web 配置方式，openwrt 也不例外，有了 web 界面就可以简单配置有关路由器的一些参数了，比如 ip 地址，无线参数等，而且如果做自己的应用的话 web 界面也少不了。Openwrt 提供了两种 web 界面，luci 和 webif，这两种界面各有千秋。Luci 是官方提供的标准界面，webif 是第三方开发的界面。他们有一个共同点是他们都是脚本，解释性语言。这就为他们的扩张提供了方便。

本章节学习在 openwrt 上面 web 的移植和开发，对于 web 的移植 openwrt 已经做了集成移植比较简单，但是 web 的开发官网上面介绍的非常少，所以这里也介绍下怎么增加一个简单界面，为以后的扩张。

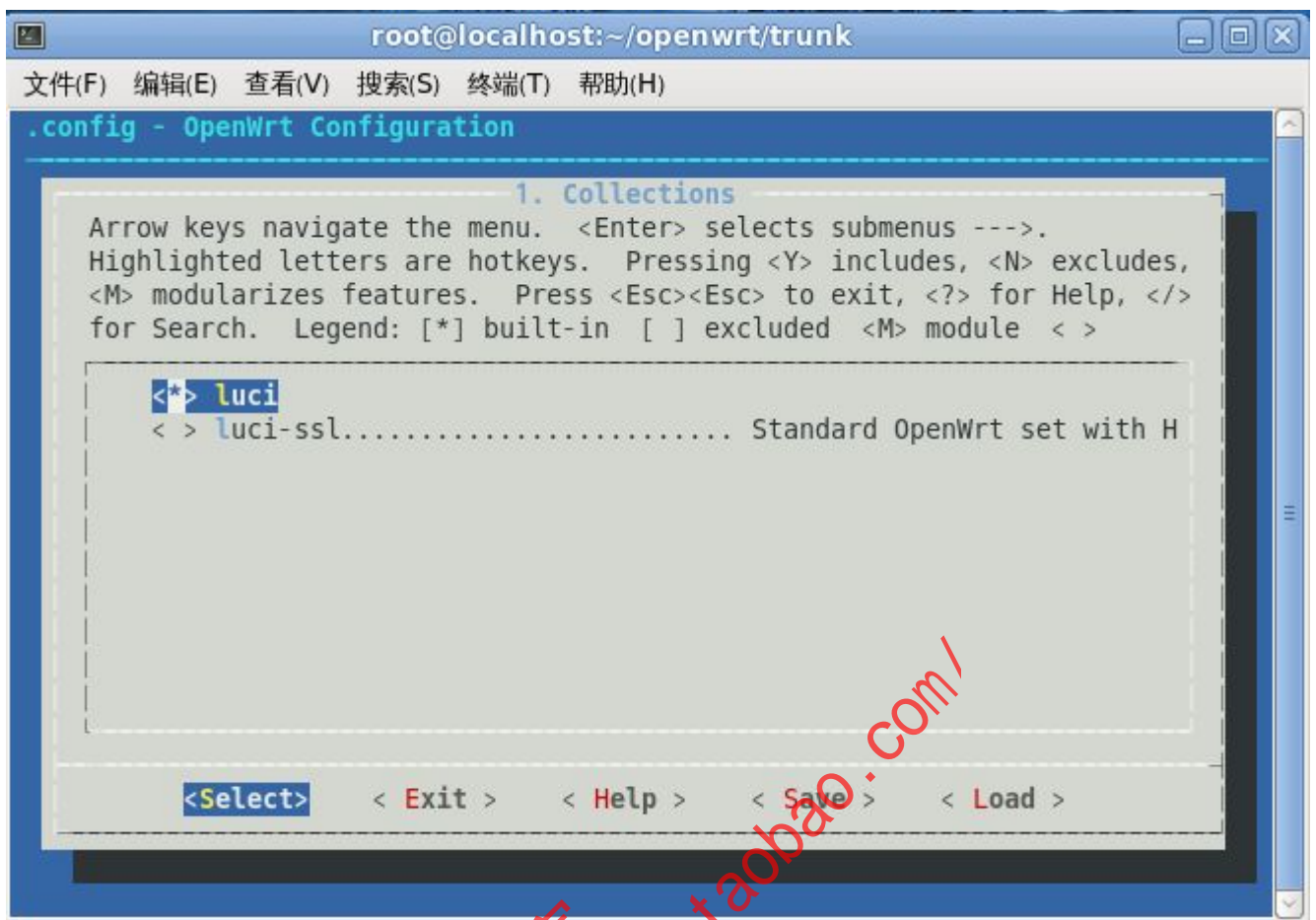
6.2 Luci

6.2.1 Luci 编译

首先介绍 luci 的编译，luci 的编译比较简单。首先进入 openwrt 的配置界面，进入 luci 的界面选择页面，如下图所示：



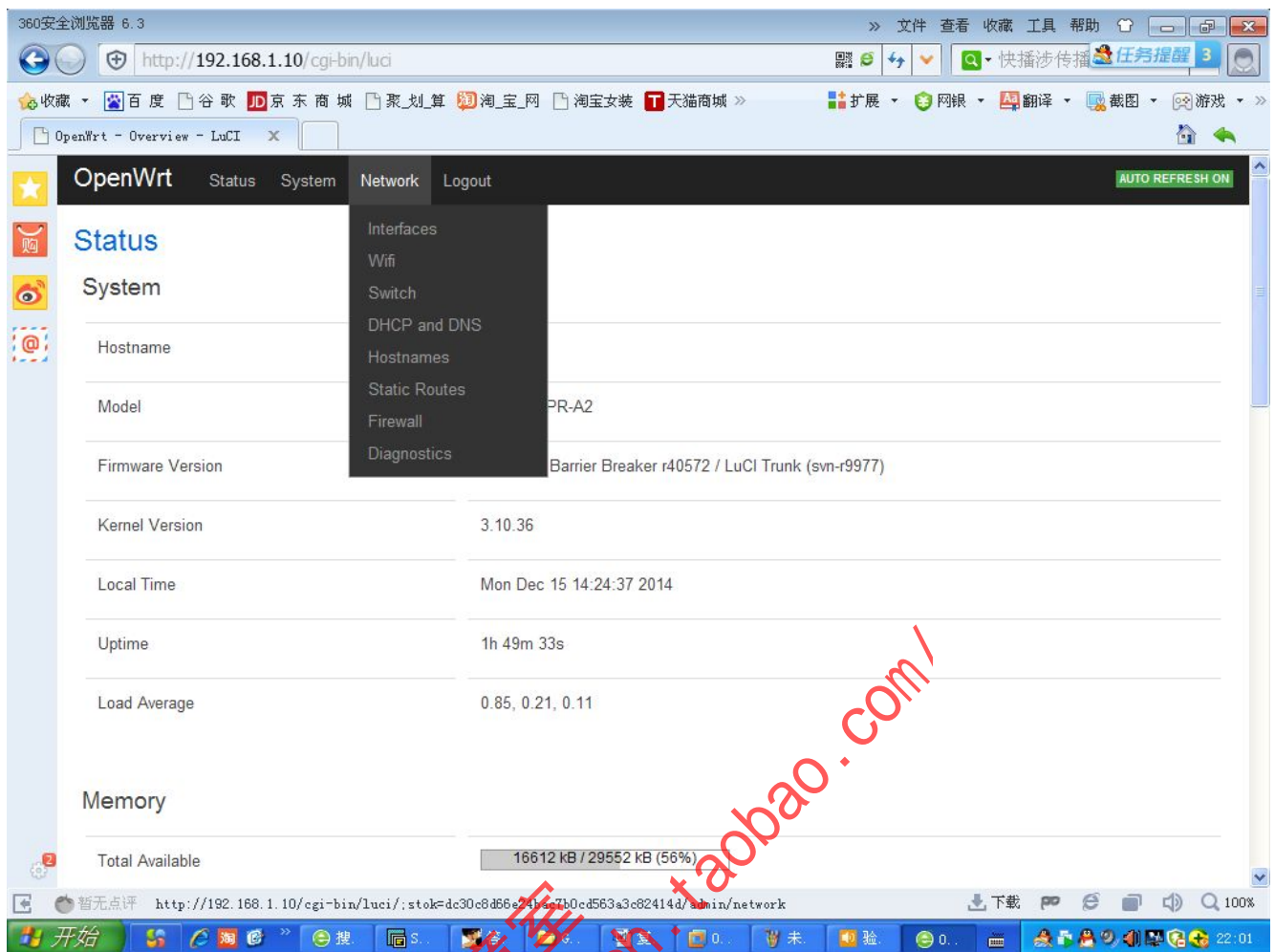
进入 luci→Collections 界面，如下图所示



选中 luci，这样基本的 luci 界面就选中了，然后保存退出编译即可，编译的话请参考编译章节。这样编译出的固件即具有了 luci 界面。

任何一个 web 界面都包括两部分内容，一部分是 web 服务器，一部分是界面语言，openwrt 的 web 服务器默认采用 uhttpd，端口 80，web 界面语言为 luci，luci 包括两部分内容，它是由 lua 和 uci 合并起来的，lua 是一种执行效率非常高的解释性语言，uci 是配置 openwrt 的默认工具。对于 uci 的使用后面章节会有介绍，这里就不说了。

编译好固件之后下载进去即可使用 luci 了，默认的路由器界面如下所示：



是不是很方便，下面咱们就解开 luci 的面纱。

6.2.2 Luci 页面编辑

Luci 的界面用户目录在 `/usr/lib/lua/luci`, luci 是 mvc 架构, `/usr/lib/lua/luci` 下有三个目录 `model`、`view`、`controller`, 它们对应 `M`、`V`、`C`。下面简单介绍生成界面的方法。其中 `M` 是已经生成的 web 控件, 其实这里就是写好的 lua 脚本, `V` 是给外部提供的一些 web 界面, `C` 控制其 `V` 和 `M` 的显示方式, 从上面的描述可以看出 `C` 部分是其入口的地方, 现在就进入 `controller` 看看他的内容, 其实它的入口地址在 `controller` 的 `admin` 目录下面, 里面的各个 lua 文件就是各个入口, 比如 `system.lua` 就对应的主界面的 `system` 栏下面, 咱们现在希望在 `system` 栏中增加一个自定义的界面, 这里为了方面描述只显示 `helloworld`。

每个功能界面的入口地址为对应的 lua 文件中 `entry` 功能函数, `entry` 功能函数的原型为

```
entry(path, target, title=nil, order=nil)
```

前面两个参数是必须的, 后面两个是可选的。Path 形如 `(admin, system, reboot)`, `entry` 根据这些属性创建节点, 比如上面内容代表在 `system` 标签中的 `reboot` 页面, `target` 是这里的重点, 它描述了当用户调用这个界面所执行的行为。这里主要由三种行为, `call`、`template`、`cbi`, 其中 `call` 为调用文件中的功能函数, `template` 为调用 `view` 中的 `html` 界面, `cbi` 为调用 `mode` 中的 `lua` 文件。Title 是在页面中显示的内容, `order` 是显示的顺序, 顺序越小显示内容越靠前。

下面在 `system.lua` 中增加一条

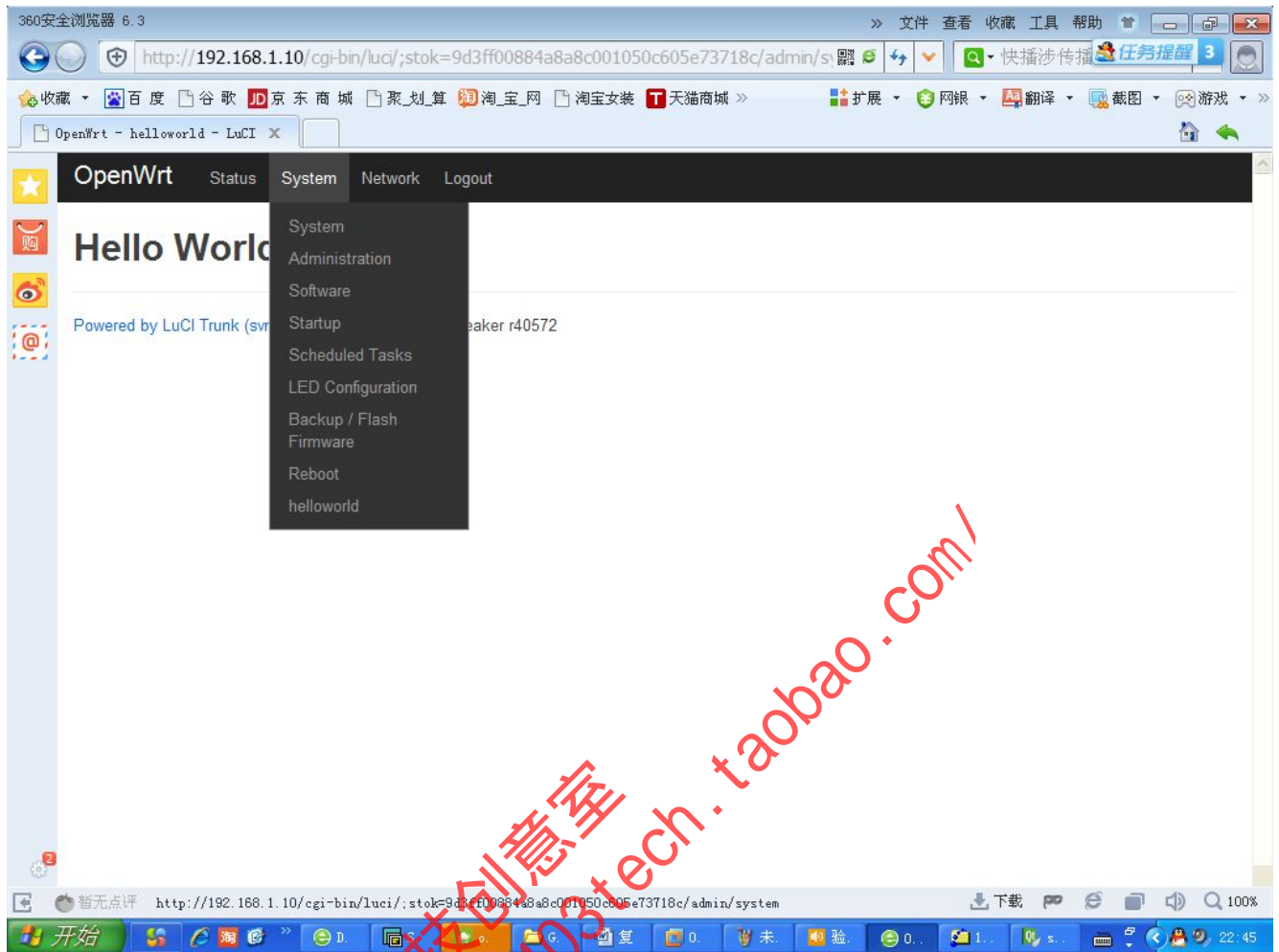
```
entry({"admin", "system", "helloworld"}, template("admin_system/helloworld"), _("helloworld"), 99)
```

其中第一个参数代表一个节点, 第二个参数, 调用 `view/admin_system/helloworld.htm` 文件, 第三个参数为显示的名字, 第四个参数为显示顺序, 这里在最后显示, 其中 `helloworld.htm` 文件的内容如下:

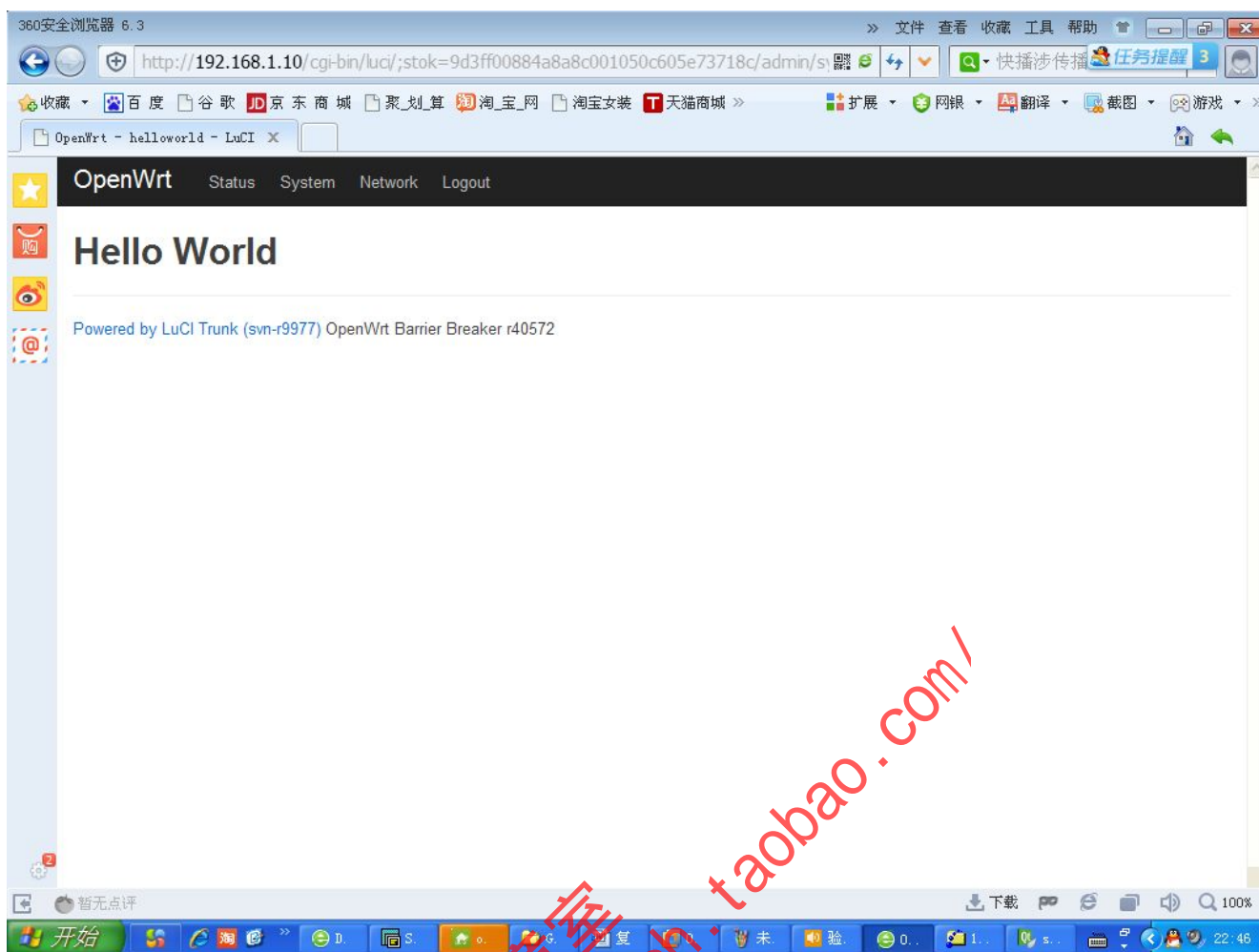
```
<%+header%>
```

```
<h1><%:Hello World%></h1>
<%+footer%>
```

执行完之后可以看下界面的显示,



可以看到 system 下面有了 helloworld 选项，点击他进入界面，如下图所示：



可以看到和界面内容保持一致。这样一个简单的界面就增加成功了，不过只是简单显示了一条语句，要增加更复杂的内容，后面会慢慢详细介绍。Luci 的开发就先简单介绍到这里。

注意：

(11) 此教程为 F403 科技创意室版权所有；

(12) 此 openwrt 教程不不断更新，请关注我们的淘宝店：

<http://f403tech.taobao.com/>

向客服索要最新版本的教程。

(3) 针对该教程，我们特意创建了 QQ 超级群，由教程的作者亲自指导大家学习。详情请咨询客服。