

# 矽昌OpenWrt用户手册

---

- 矽昌OpenWrt用户手册
  - 总览
    - 文档总览
      - 主题
      - 缩略语和首字母缩略词
  - Siflower路由器软件概述
    - OpenWrt架构
      - 源码结构
      - 主要服务
      - Siflower移植代码
    - Linux Kernel
      - 源码结构
      - 修改内核
      - 将补丁添加到内核
  - Flash布局
  - 下载和构建
    - 下载源文件
    - 建立一个构建环境
    - 运行构建
      - 基本构建
      - 简单的编译内核包
      - 安装提要（可选）
      - 输出文件
      - 清理编译
  - 安装
    - 从uboot安装
    - 从openwrt安装
  - 调试
    - 串口
    - JTAG
    - GDB
    - 调试信息为包添加调试信息（例如）
      - Coredump文件

# 总览

## 文档总览

### 主题

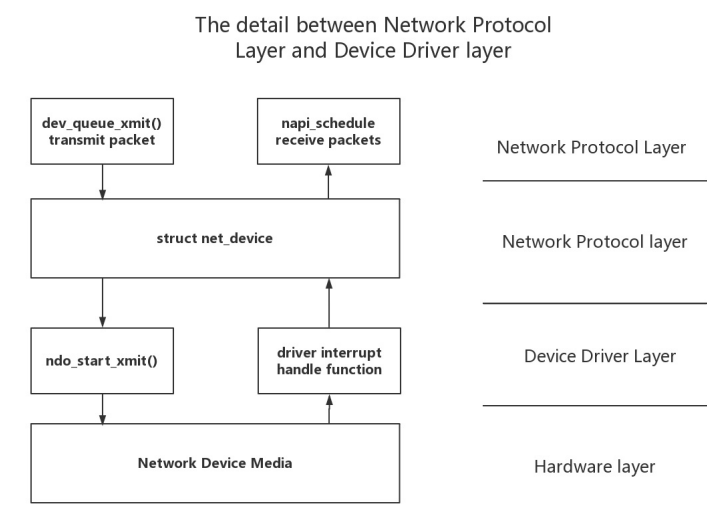
本文档的目的是描述Siflower路由器软件的体系结构，以及编译和安装的准则。 本文档针对希望在Siflower EVB板上开发路由器生产的人员。

### 缩略语和首字母缩略词

缩写	含义
EVB	工程开发板
UBoot	通用引导加载程序
UBoot env	通用引导加载程序环境变量
Spl	第二程序加载器
Rootfs	根文件系统
OpenWrt	专注于路由产品的开源软件项目
luci	OpenWRT 中的一个软件，提供web 界面

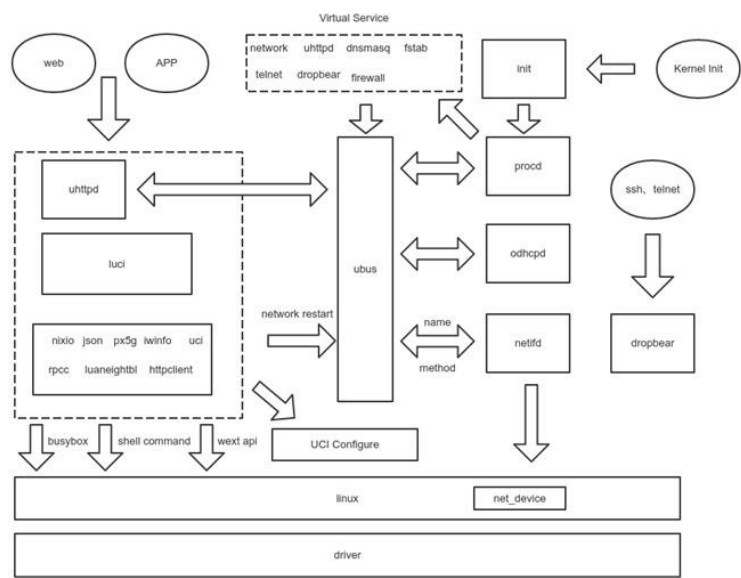
## Siflower路由器软件概述

本节介绍Siflower路由器软件解决方案的主要组件。下图显示了路由器软件的最高层次。



## OpenWrt架构

根据GNU通用公共许可证第2版的规定，OpenWrt项目是免费软件。它拥有一个完整的软件堆栈，可由OEM和其他设备实现者移植并在自己的硬件上运行。此地图显示openwrt项目提供的主要软件。



源码结构

现在我们使用chaos\_calmer\_15\_05\_1作为我们的基本openwrt项目分支，它是2016年3月由openwrt.org发布的最新稳定版本。

该表显示了OpenWrt提供的主要源文件夹和包。

文件夹	描述
tools	获取代码和编译时使用的主机端工具
toolchain	包括内核头文件，C库，交叉编译器，调试器
target	定义供应商文件和图像工具
package	包含OpenWrt提供的所有基本包
include	包含主要的Makefiles和编译规则
scripts	包括配置脚本，补丁脚本，软件源脚本
dl	编译时包含所有下载包
build_dir	编译时的临时文件以及提取的源代码
staging_dir	编译环境包括常见的头文件和工具链
feeds	所有可选软件包由openwrt或thirdparty提供
bin	包含输出文件

更多的细节可以参考<https://wiki.openwrt.org/doc/guide-developer>。

## 主要服务

此表显示OpenWrt启动时的主要服务。

服务 | 描述 |

---|---

| dropbear | 为小型内存环境设计的小型SSH2服务器/客户端。 |

| dnsmasq | 它旨在为LAN提供耦合的DNS和DHCP服务。 |

| telnetd | telnet服务器的Telnet守护进程 |

| uhttpd | 小巧的单线程HTTP服务器 |

| netifd | 网络接口管理器服务 |

| odhcpd | 用于ipv6的DHCP服务器 |

| ubusd | 进程间通信服务 |

| logd | 记录用户空间的服务 |

| ntpd | 网络时间同步守护进程 |

| hostapd | IEEE 802.1x / WPA / EAP / RADIUS 认证器。 |

## Siflower移植代码

所有用于OpenWrt的siflower移植代码都放置在target / linux / siflower / 中。 我们保持所有其他文件夹是干净的。

## Linux Kernel

现在我们使用Linux内核版本3.18.29，它是最新的chaos\_calmer\_15\_05\_1的内核版本。 Openwrt项目在标准内核上有一系列补丁，可以在网络上进行优化或支持上层文件系统。

## 源码结构

内核源代码作为一个包放在dir target / linux / 中。 源代码组织：

```
target / linux / generic / patches (openwrt官方给linux的基本补丁)
target / linux / generic / config-3.18 (基于openwrt官方对linux的基本配置)
target / linux / siflower / patches (供应商补丁到linux)
target / linux / siflower / sf16a18-mpw0 / config-3.18 (供应商配置到linux)
```

## 修改内核

编译时内核源代码将被提取到build\_dir文件夹中。 你可以在下面的目录中找到它。 您可以修改此目录中的内核代码并重建固件。 但有一点你必须注意，当你使用命令

```
make clean
```

时，这个目录将被清理干净。

```
build_dir/target-mipsel_mips-interAptiv_uClibc-0.9.33.2/linux-siflower_sf16a18-mpw0/lin
```

将补丁添加到内核

Quilt是OpenWrt使用的默认修补工具，通常我们需要一系列步骤来创建一个内核如下的新修补程序。

```
make target/linux/{clean,prepare} V=s QUILT=1 // make sure the kernel source is clean
cd to linux source dir like below:
build_dir/target-mipsel_mips-interAptiv_uClibc-0.9.33.2/linux-siflower_sf16a18-mpw0/lin
quilt series // display current patches in kernel
quilt new platform/001_test.patch // add a new patch which name should be in order
quilt add drivers/mtd/mtdpart.c // make a association between source file and current p
do whatever modification you like
quilt refresh // effect changes into patch
cd - // return to top dir
make target/linux/update // collect patches into vendor dir
ls target/linux/siflower/patches/001_test.patch // now patch is available and you can u
```

更详细的你可以参考<https://wiki.openwrt.org/doc/devel/patches>

Flash布局

Flash分区：

Layer0	raw flash 16M						
Layer1	uboot partitions				mtd4 firmware 15744K		
Layer2					mtd6 rootfs 14267K mounted: " / "		
Layer3	mtd0 spl_loader 128k	mtd1 uboot 384K	mtd2 uboot_env 64k		mtd3 factory 64K	mtd5 kernel 1477K ulmage_lzma	/dev/root mounted: " / rom " 5371K root.squashfs (increase in 256K for mkfs with block size

				256K)	
--	--	--	--	-------	--

每个分区的描述：

分区	描述
Spl	uboot的第一阶段，负责将Uboot加载到dram和init hw中
Uboot	负责从spi中提取ulmage.lzma到dram并跳转到内核
Uboot-env	存储uboot使用的通用参数，例如波特率
Factory	Store参数在重置或升级时不会被删除
Linux	标准Linux内核与硬件交互
Rootfs	除内核以外的所有openwrt文件系统
Rootfs_data	rootfs_data Jffs2 rw文件系统

## 下载和构建

### 下载源文件

Openwrt源代码树位于Git存储库中。确保已安装git并进行了正确配置。

你可以检查下面准备的源代码，你应该从PM处得到地址。

```
git clone ssh://username@192.168.1.10:29428/external/***
```

### 建立一个构建环境

OpenWrt构建系统是OpenWrt Linux发行版的构建系统。OpenWrt构建系统适用于GNU / Linux，BSD或MacOSX操作系统。区分大小写的文件系统是必需的。

要生成一个可安装的OpenWrt固件映像文件，其大小例如为 8MB，你需要：

```
ca. 200 MB of hard disk space for OpenWrt build system
ca. 300 MB of hard disk space for OpenWrt build system + OpenWrt Feeds
ca. 2.1 GB of hard disk space for source packages downloaded during build from Open
ca. 3-4 GB of available hard disk space to build (i.e. cross-compile) OpenWrt and g
ca. 1-4 GB of RAM to build Openwrt.(build x86's img need 4GB RAM)
```

为了方便地下载OpenWrt源代码，并构建工具来完成交叉编译过程：

```
sudo apt-get update
sudo apt-get install git-core build-essential libssl-dev libncurses5-dev unzip gawk zli
```

有些提要可能不适用于git，但只能通过subversion（简称：svn）或mercurial。如果你想获得他们的源代码，你需要安装svn和mercurial:

```
sudo apt-get install subversion mercurial
```

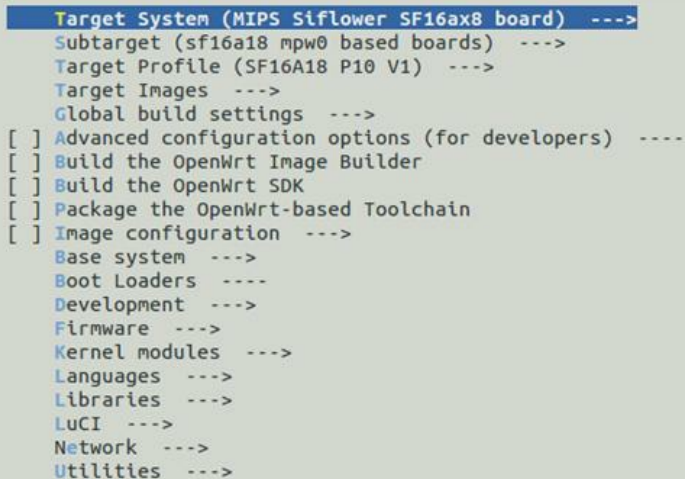
更多的细节你可以参考<https://wiki.openwrt.org/doc/howto/buildroot.exigence>

## 运行构建

### 基本构建

使用以下命令使OpenWrt构建系统检查构建系统上丢失的包:

```
make menuconfig
choose Target System: MIPS Siflowr SF16ax8 board
choose Target Profile: SF16A18 P10 V1
save configuration
```



```
Target System (MIPS Siflowr SF16ax8 board) --->
Subtarget (sf16a18 mpw0 based boards) --->
Target Profile (SF16A18 P10 V1) --->
Target Images --->
Global build settings --->
[ ] Advanced configuration options (for developers) ----
[ ] Build the OpenWrt Image Builder
[ ] Build the OpenWrt SDK
[ ] Package the OpenWrt-based Toolchain
[ ] Image configuration --->
Base system --->
Boot Loaders ----
Development --->
Firmware --->
Kernel modules --->
Languages --->
Libraries --->
LuCI --->
Network --->
Utilities --->
```

现在，所有内容都已准备好用于构建镜像，这是通过一个命令完成的:

```
make -j V=s
```

更多细节你可以参考<https://wiki.openwrt.org/doc/howto/build>。

### 简单的编译内核包

你可以在下面编译内核独立使用命令。

```
make target/linux/compile V=s
```

## 安装提要（可选）

在OpenWrt中，“feed”是一组共享相同位置的软件包。**Feed**可能位于任何可通过支持的**Feed**方式的协议上的单个名称（路径/**URL**）寻址的位置。您可以安装下面的所有**Feed**（制作之前）：

```
./scripts/feeds update -a  
./scripts/feeds install -a
```

更详细的你可以参考<https://wiki.openwrt.org/doc/devel/feeds>。

## 输出文件

Openwrt项目将文件系统和内核编译为如下的单个文件。该文件将用作**sys upgrade**映像。

```
bin/siflower/openwrt-siflower-sf16a18-mpw0-squashfs-sysupgrade.bin
```

## 清理编译

您可能需要不时地清理构建环境。以下制作目标对于这项工作很有用。

```
1)make clean  
删除目录/ bin和/ build_dir的内容。 使清洁不会删除工具链，它也可以避免清除不同于您在.config  
  
2)make dirclean  
删除目录/ bin和/ build_dir以及/ staging_dir和/ toolchain (=交叉编译工具) 和/ logs的内容。  
  
3)make distclean  
将编译或配置的所有内容都删除，并删除所有已下载的提要内容和程序包源。  
  
4) make target / linux / clean  
清理linux对象。  
  
5)make package/luci/clean  
清理luci包对象。
```

## 安装

有两种方法可以将新镜像安装到EVB板上。 他们都将使用网络浏览器将镜像上传到您的主板。

## 从uboot安装

如果您的EVB板已经被openwrt图像烧坏了，我们需要从uboot stage安装完整的openwrt镜像。但首先确保您的EVB板上有一个正确的uboot镜像，否则您将不得不从jtag安装uboot镜像。



安装步骤如下：

- 1) 确保串行端口正确连接到您的PC。设置波特率为115200。
- 2) 使用静态IP地址设置您的PC。将第一个以太网端口连接到PC。EVB板上的第一个端口是距离USB最近的端口。
- 3) 打开EVB板。uboot启动时，在串行控制台中打开任意键。现在我们将进入如下的uboot命令模式。

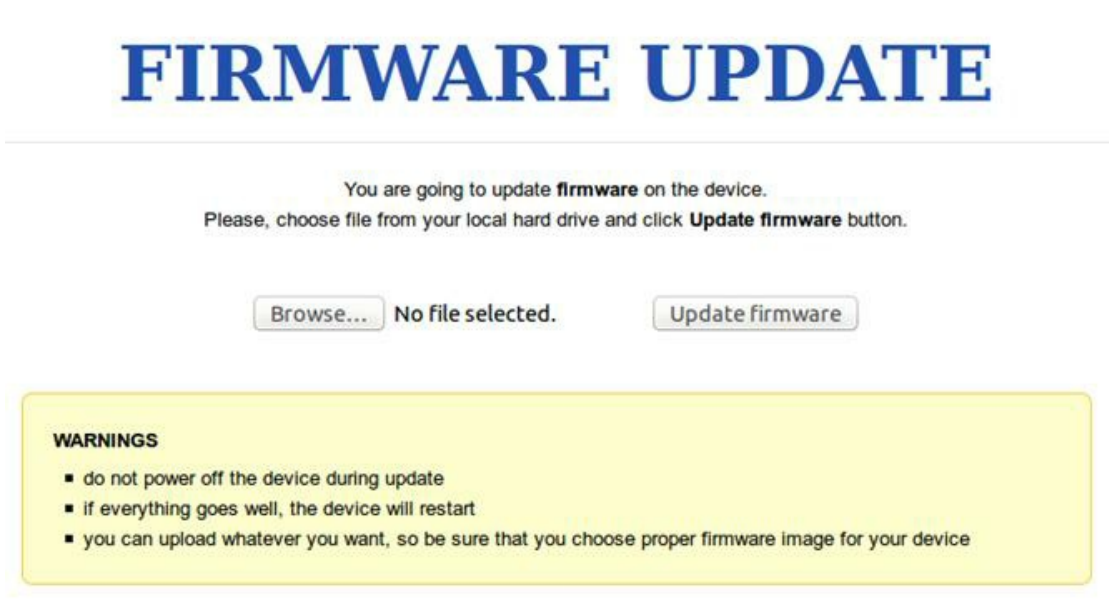
Hit any key to stop autoboot: 0

sfa18 #

- 4) 输入httpd 192.168.4.1，然后按ENTER键。这里的IP地址可以是任何与您的PC具有相同前缀的值。

sfa18 # httpd 192.168.4.1

- 5) 通过网页浏览器（如Chrome）访问网址192.168.4.1。如果一切正在进行，您将获得下面的页面。



- 6) 选择一个sysupgrade.bin，然后更新固件。
- 7) 将图像刻录到闪光灯需要几秒钟的时间。刻录后该板会自动重启。您可以从串口控制台获取详细信息。

## 从openwrt安装

如果您的EVB板上运行openwrt系统。安装新的固件镜像要容易得多。安装步骤如下所示。

- 1) 将任何以太网局域网端口连接到PC，确保您可以使用浏览器（例如chrome）访问路由器管理页面。
- 2) 按以下菜单顺序查找备份页面：系统/备份/闪存新的固件映像。
- 3) 选择一个图像并执行“Flash图像”，然后执行“继续”。
- 4) 将图像刻录到闪光灯需要几秒钟。刻录后该板会自动重启。

# 调试

## 串口

对于POSIX系统，我们推荐minicom作为默认终端软件。默认情况下，您应该将波特率设置为115200。

## JTAG

我们在soc上使用MIPS interAptiv。如果您打算使用JTAG调试EVB板，则必须准备一个MIPS Debugger并在您的计算机上安装了codescape。

## GDB

更详细的你可以参考<https://wiki.openwrt.org/doc/devel/gdb>。

## 调试信息为包添加调试信息（例如）

包/网络/ utils的/ iwininfo/生成文件

CFLAGS =“\$（TARGET\_CFLAGS）-Wall -g”

## Coredump文件

在某些进程崩溃时使用coredump进行调试。

`ulimit -c unlimited` //设置无限制的核心文件

`sudo sh -c“echo 1> /proc / sys / kernel / core_uses_pid”`//为核心文件添加pid信息

当崩溃时，您可以使用gdb分析coredump文件。

STAGING\_DIR/工具链-mipsel\_mips-interAptiv\_gcc-4.8 linaro\_uClibc-0.9.33.2/ bin中/ mipsel体系-的 OpenWrt-Linux的GDB