Openwrt通用固件make menuconfig配置

选择CPU型号

Target System --->AtherosAR7xxx/AR9xxx

选择路由型号

Target Profile --->TP-LINK TL-WR843N/ND (QCA9531) 要省空间可去掉 dnsmasq和firewall

添加luci相关项

LuCI ---> Collections ---> <\*> luci 添加luci(web界面管理)

LuCI ---> Applications ---> <\*> luci-app-adkill 98 添加去广告

LuCI ---> Applications ---> <\*> luci-app-commands 添加luci的网页shell

LuCI ---> Applications ---> <\*> luci-app-ddns 添加动态域名

LuCI ---> Applications ---> <\*> luci-app-firewall 添加防火墙

LuCI ---> Applications ---> <\*> luci-app-hd-idle 添加硬盘休眠

LuCI ---> Applications ---> <\*> luci-app-syncdial 98 虚拟网卡 原macvlan

LuCI ---> Applications ---> <\*> luci-app-mjpg-streamer 98 添加视频监控

LuCI ---> Applications ---> <\*> luci-app-mmc-over-gpio 添加SD卡操作界面

LuCI ---> Applications ---> <\*> luci-app-multiwan 网络叠加 nwan、pppoe多拨

LuCI ---> Applications ---> <\*> luci-app-mwan3 网络叠加

LuCI ---> Applications ---> <\*> luci-app-ntpc 时间同步服务器

LuCI ---> Applications ---> <\*> luci-app-ocserv VPN Server

LuCI ---> Applications ---> <\*> luci-app-openvpn 98 添加openvpn

LuCI ---> Applications ---> <\*> luci-app-p910nd 添加打印服务器

LuCI ---> Applications ---> <\*> luci-app-usb-printer 添加打印服务器 hackpascal 优化

LuCI ---> Applications ---> <\*> luci-app-qos 添加服务质量 可选石像鬼QOS

LuCI ---> Applications ---> <\*> luci-app-samba 添加网络共享

LuCI ---> Applications ---> <\*> luci-app-redsock2 9812 科学上网

LuCI ---> Applications ---> <\*> luci-app-shadowsocks 9812 科学上网

LuCI ---> Applications ---> <\*> luci-app-splash 客户端弹窗

LuCI ---> Applications ---> <\*> luci-app-transmission BT下载

LuCI ---> Applications ---> <\*> luci-app-upnp 添加通用即插即用

LuCI ---> Applications ---> <\*> luci-app-vsftpd 9812 添加ftp服务

LuCI ---> Applications ---> <\*> luci-app-webshell 98 添加网页命令行终端

LuCI ---> Applications ---> <\*> luci-app-wol 添加网络唤醒

LuCI ---> Applications ---> <\*> luci-app-xunlei 98 迅雷下载

LuCI ---> Applications ---> <\*> luci-app-xware3 98 迅雷下载 32bit

LuCI ---> Themes ---> <\*> luci-theme-bootstrap 默认主题

LuCI ---> Translations ---> <\*> luci-i18n-chinese 添加luci的中文语言包

Modules ---> Translations ---> <\*> Simplified Chinese (zh-cn)新版本中文语言包位置

*红色部分官方源默认没有，使用981213的 feeds.conf.default 合并后可看到98的选项，9812的在单独的 RA-MOD ---> LuCI Applications 下！*

添加高速数据支持

Kernel modules ---> Block Device ---> <\*> kmod-block2mtd

Kernel modules ---> Block Device ---> <\*> kmod-scsi-core

Kernel modules ---> Block Device ---> <\*> kmod-scsi-generic （usb 转IDE ，SATA）

添加硬盘格式支持

Kernel modules ---> Filesystems ---> <\*> kmod-fs-ext4

Kernel modules ---> Filesystems ---> <\*> kmod-fs-nfs

Kernel modules ---> Filesystems ---> <\*> kmod-fs-nfs-common

Kernel modules ---> Filesystems ---> <\*> kmod-fs-ntfs (只读)

Kernel modules ---> Filesystems ---> <\*> kmod-fs-vfat

Kernel modules ---> Filesystems ---> <\*> kmod-fuse

Utilities --->Filesystem ---> <\*> ntfs-3g (比ntfs更好用，但CPU占用略高)

添加编码

Kernel modules ---> Native Language Support ---> <\*> kmod-nls-cp437 (FAT-fs 文件系统需要cp437支持)

Kernel modules ---> Native Language Support ---> <\*> kmod-nls-iso8859-1

Kernel modules ---> Native Language Support ---> <\*> kmod-nls-cp936

Kernel modules ---> Native Language Support ---> <\*> kmod-nls-utf8

添加SD卡支持

Kernel modules ---> Other modules ---> <\*> kmod-mmc

Kernel modules ---> Other modules ---> <\*> kmod-sdhci

Kernel modules ---> Other modules ---> <\*> kmod-sdhci-mt7620

添加USB扩展支持

Base system ---> <\*> block-mount 添加USB挂载

Base system ---> <\*> librt (libpthread 会自动联动选中) 添加USB挂载

Kernel modules ---> USB Support ---> <\*> kmod-usb-hid （usb键盘鼠标支持） <\*> kmod-usb-ohci

<\*> kmod-usb-storage (usb storage 驱动)

<\*> kmod-usb-storage-extras <\*> kmod-usb-uhci (usb 1.1 驱动) <\*> kmod-usb2

<\*> kmod-usb3

添加USB转串口驱动

Kernel modules ---> USB Support ---> <\*> kmod-usb-acm (Lanuchpad ,Arduino UNO驱动)

Kernel modules ---> USB Support ---> <\*> kmod-usb-serial

<\*> kmod-usb-serial-ch341

<\*> kmod-usb-serial-cp210x

<\*> kmod-usb-serial-ftdi

Kernel modules ---> USB Support ---> <\*> kmod-usb-serial-pl2303

添加打印驱动

Kernel modules ---> USB Support ---> <\*> kmod-usb-printer （驱动有点大，小容量rom不建议选)

添加网络配置

Network ---> SSH ---> <\*> openssh-client （SSH客户端）

Network ---> <\*> ppp-mod-pppoe （PPPOE拨号模式）

Network ---> <\*> ppp-mod-pptp （VPN客户端）

Network ---> <\*> wpad (为hostapd和wpa-supplicant的集合，支持802.1x认证，替换 wpad-mini)

Kernel modules ---> Network Devices ---> <\*> Kmod-vmxnet3 (x86\_vmware 网卡驱动)

Kernel modules ---> Network Devices ---> <\*> Kmod-pcnet32 (x86\_vmware 网卡驱动)

添加视频支持

Kernel modules ---> Video Support ---> <\*> kmod-video-core

<\*> kmod-video-uvc

Multimedia ---> <\*> mjpeg-streamer

添加通讯协议支持

Network ---> <\*> ser2net 用于和单片机通讯

Network ---> SSH ---> <\*> openssh-sftp-server sftp协议支持，xftp可用，不依赖 vsftpd

添加BT下载工具

Network --->BitTorrent ---> <\*> transmission-daemon

Network --->BitTorrent ---> <\*> transmission-remote

Network --->BitTorrent ---> <\*> transmission-web

添加FTP

Network ---> File Transfer ---> <\*> vsftpd-pam （官方源码只有 vsftpd）

添加pam支持

Libraries ---> SSL ---> <\*> libopenssl

Libraries ---> <\*> libdb47

Libraries ---> <\*> libpam-db

Utilities ---> <\*> db47-utils

添加lsusb命令

Utilities ---> <\*> usbutils

Libraries ---> <\*> libusb-1.0

添加无线网卡驱动

Kernel modules ---> Wireless Drivers ---> <\*> kmod-lib80211

Kernel modules ---> Wireless Drivers ---> <\*> kmod-rt2800-usb (3070支持)

Kernel modules ---> Wireless Drivers ---> <\*> kmod-rtl8187

Kernel modules ---> Wireless Drivers ---> <\*> kmod-rtl8192se

Kernel modules ---> Wireless Drivers ---> <\*> kmod-zd1211rw

添加应用程序配置

Utilities ---> Compression ---> <\*> unrar （解压缩工具）

Utilities ---> Compression ---> <\*> unzip （解压缩工具）

Utilities ---> Compression ---> <\*> zip （压缩工具）

Utilities ---> Filesystem ---> <\*> badblocks （支持ext2文件系统）

Utilities ---> Filesystem ---> <\*> e2fsprogs （支持ext2/ext3/ext4格式化工具）

Utilities ---> disc ---> <\*> blkid （可以列出分区类型卷标等）

Utilities ---> disc ---> <\*> fdisk （分区工具）

Utilities ---> disc ---> <\*> lsblk （列出块设备，还能显示他们之间的依赖关系）

Utilities ---> <\*> bzip2 （解压缩工具）

Utilities ---> <\*> lrzsz （上传下载工具）

Utilities ---> <\*> restorefactory （reset键支持(长按5秒以上就可以恢复固件默认设置)）

*981213源码中无该项，官方源码有，981213源码更新到官方代码后会消失！*

Utilities --> <\*> wifitoggle （添加一键开关无线(按一下WPS键放开无线就打开或者关闭)）

添加其他功能（WIFI破解）

Network --> wireless --> <\*> aircrack-ng

Network --> wireless --> <\*> mdk3

Network --> wireless --> <\*> reaver

Utilities ---> <\*> screen *还需要wireless-tools，libpcap可能需要降级到1.1.1版本*

Network --> VPN --> <\*> openvpn-polarssl （使用PolarSSL开源VPN解决方案）

Libraries --> <\*> libffmpeg-full （流媒体服务器）

<\*> luci-app-minidlna （流媒体服务器）

解决DNS污染（源码中没有）

dnscrypt-proxy （是opendns使用椭圆曲线加密算法）

pdnsd （是给 dnscrypt-proxy 做加速的，每次都从opendns 加密查询dns，虽然很有保障，但是会很慢。pdnsd 监听1053端口，用 dnscrypt-proxy 作为上级DNS服务器，将查询结果缓存起来，可以缓存最多一周）

dnsmasq （屏蔽 运营商的dns，查询pdnsd。利用 dnsmasq，可以让连到路由器上的客户端 都应用 pdnsd。）

make V=99 2>&1 |tee build.log |grep -i error 生成编译报告日志文件 make –j 2 V=s 多线程编译

make -j 2 V=s 2>&1 | tee build.log | grep -i error

路由器固件受ROM容量限制，可先编译x86版本进行测试！

以下是个性设置：基于WR703N

1.添加一键无线

在如下位置新建一个文件，文件名为01onoff

/target/linux/ar71xx/base-files/etc/hotplug.d/button/01onoff

内容如下：

#! /bin/sh

[ "$BUTTON" = "wps" ]&& [ "$ACTION" = "pressed" ] && {

SW=$(uci get wireless.@wifi-device[0].disabled)

[ $SW == '0' ] && uci setwireless.@wifi-device[0].disabled=1

[ $SW == '0' ] || uci setwireless.@wifi-device[0].disabled=0

wifi

}

保存后设置权限为0777

在终端下进入button文件夹，然后执行 chmod 777 –R 01onoff 命令

进入文件夹命令为 cd target/linux/ar71xx… ，最前面的路径前没有‘/’符号！

2.添加 3322 DDNS 动态域名解析

/feeds/packages/net/ddns-scripts/files/usr/lib/ddns/services 添加

"3322.org" “http://[USERNAME]:[PASSWORD]@members.3322.org/dyndns/update?system=dyndns&hostname=[DOMAIN]&myip=[IP]&wildcard=OFF"

修改配置项

/feeds/packages/net/ddns-scripts/files/etc/config/ddns

config service "myddns"

option enabled "1"

option service\_name "3322.org"

option domain "xxxx.3322.org"

3.修改防火墙添加开放端口

/trunk/package/network/config/firewall/files/firewall.config 添加内容：

config 'rule'

option 'target''ACCEPT'

option '\_name' 'tr'

option 'src' 'wan'

option 'proto''tcpudp'

option 'dest\_port''51413'

config 'rule'

option 'target''ACCEPT'

option '\_name''9091'

option 'src' 'wan'

option 'proto' 'tcp'

option 'dest\_port''9091'

4.修改无线默认启动发射功率及加密

/package/mac80211/files/lib/wifi/mac80211.sh

修改内容：注意对齐,参考修改 在文件最后修改以下内容：

config wifi-device radio$devidx

option type mac80211

option channel ${channel}

option macaddr $(cat/sys/class/ieee80211/${dev}/macaddress)

option hwmode 11${mode\_11n}${mode\_band}

$ht\_capab

# REMOVETHIS LINE TO ENABLE WIFI:

option disabled 0

option txpower 17

option htmode HT40-

option noscan 1

option country CN

config wifi-iface

option device radio$devidx

option network lan

option mode ap

option ssid OpenWrt\_$(cat/sys/class/ieee80211/${dev}/macaddress|tr "[a-z]""[A-Z]"|sed 's/://g'|cut -c7-12)

option encryption psk2 -----加密方式 ( option encryption none 无密码)

option key xxxxxxxx ----密码 (8位) xxxxxxxx

EOF

devidx=$(($devidx + 1))

done

}

5.修改路由器名字和时区

/package/base-files/files/etc/config/system 修改内容：

config system

option hostname OpenWrt

option zonename Asia/Shanghai

option timezone CST-8

6.默认启动DHCP(703n用来当二级路由)

/trunk/package/network/services/dnsmasq/files/dhcp.conf

config dhcp lan

option interface lan

option start 100

option limit 150

option leasetime 12h

option ignore 0 -----------------添加这个

7.修改Transmission配置文件

/feeds/packages/net/transmission/files/transmission.config 修改内容：

option rpc\_authentication\_required true

option umask 0

8.修改network配置，配置成二级路由

/trunk/package/base-files/files/etc/config/network 修改内容：

# Copyright (C) 2006 OpenWrt.org

config interface loopback

option ifname lo

option proto static

option ipaddr 127.0.0.1

option netmask 255.0.0.0

config interface lan

option type bridge

option proto static

option ipaddr 192.168.3.1

option netmask 255.255.255.0

config interface wan

option ifname eth0

option \_orig\_ifname eth0

option \_orig\_bridge false

option proto dhcp

9. 添加利用reset键的一键切换路由工作模式

在如下位置新建4个文件，在保存后，请修改权限为777

**/target/linux/ar71xx/base-files/etc/hotplug.d/button/00-button**

./etc/functions.sh

do\_button () {

local button

local action

local handler

local min

local max

config\_get button $1 button

config\_get action $1 action

config\_get handler $1 handler

config\_get min $1 min

config\_get max $1 max

[ "$ACTION" = "$action" -a "$BUTTON" = "$button" -a -n "$handler" ] && {

[ -z "$min" -o -z "$max" ] && eval $handler

[ -n "$min" -a -n "$max" ] && {

[ $min -le $SEEN -a $max -ge $SEEN ] && eval $handler

} } }

**config\_load system**

**config\_foreach do\_button button**

**/target/linux/ar71xx/base-files/etc/hotplug.d/button/change2ap**

#!/bin/sh

uci delete network.wan

uci delete network.lan

uci set network.lan=interface

uci set network.lan.ifname=eth0

uci set network.lan.type=bridge

uci set network.lan.proto=static

uci set network.lan.ipaddr=192.168.1.2

uci set network.lan.netmask=255.255.255.0

uci set network.lan.gateway=192.168.1.1

uci set network.lan.dns=192.168.1.1

uci commit network

uci set dhcp.lan.ignore=1

uci commit dhcp

reboot

**/target/linux/ar71xx/base-files/etc/hotplug.d/button/change23dhcp**

#!/bin/sh

uci delete network.wan

uci delete network.lan

uci set network.lan=interface

uci set network.lan.type=bridge

uci set network.lan.proto=static

uci set network.lan.ipaddr=192.168.10.1

uci set network.lan.netmask=255.255.255.0

uci set network.wan=interface

uci set network.wan.ifname=eth0

uci set network.wan.proto=dhcp

uci set network.wan.\_orig\_ifname=eth0

uci set network.wan.\_orig\_bridge=false

uci commit network

uci delete dhcp.lan.ignore

uci commit dhcp

reboot

**/target/linux/ar71xx/base-files/etc/hotplug.d/button/change23g**

#!/bin/sh

uci delete network.wan

uci delete network.lan

uci set network.lan=interface

uci set network.lan.ifname=eth0

uci set network.lan.type=bridge

uci set network.lan.proto=static

uci set network.lan.ipaddr=192.168.1.1

uci set network.lan.netmask=255.255.255.0

uci set network.wan=interface

uci set network.wan.ifname=ppp0

uci set network.wan.proto=3g

uci set network.wan.maxwait=0

uci set network.wan.service=evdo

uci set network.wan.device=/dev/ttyUSB0

uci set network.wan.username=ctnet@mycdma.cn

uci set network.wan.password=vnet.mobi

uci set network.wan.auto=1

uci commit network

uci delete dhcp.lan.ignore

uci commit dhcp

reboot

**修改/package/base-files/files/etc/config/system，在尾部添加**

config button

option button 'reset'

option action 'released'

option handler '/etc/hotplug.d/button/change23g'

option min '0'

option max '2'

config button

option button 'reset'

option action 'released'

option handler '/etc/hotplug.d/button/change2ap'

option min '3'

option max '8'

config button

option button 'reset'

option action 'released'

option handler '/etc/hotplug.d/button/change2dhcp'

option min '9'

option max '99'

这样以后按住reset 2秒内， 3-8秒 和 9秒以上，会自动切换到对应的网络配置，并自动重启路由器。 按住reset0-2秒放开，切换到3g配置，

按住reset3-8秒放开，切换到ap模式，

reset 9秒以上放开，切换到二级路由dhcp模式。

10. 修改挂载点

路径：./openwrt/trunk/package/block-mount/files

修改 10-swap, 20-fsck, 40-mount 的权限为777

11.添加对中文编码cp936的支持,openwrt实际上是支持cp936的，只是没开放而已

修改文件openwrt/trunk/package/kernel/modules/nls.mk 在文件尾部添加

define KernelPackage/nls-cp936

SUBMENU:=Native Language Support

TITLE:=Codepage 936 (China)

KCONFIG:=CONFIG\_NLS\_CODEPAGE\_936

FILES:=$(LINUX\_DIR)/fs/nls/nls\_cp936.ko

AUTOLOAD:=$(call AutoLoad,25,nls\_cp936)

$(call AddDepends/nls)

endef

define KernelPackage/nls-cp936/description

Kernel module for NLS Codepage 936 (Chinese)

endef

$(eval $(call KernelPackage,nls-cp936))

保存,这样在以后执行make menuconfig时，

在Kernel modules ---> Native Language Support 下面会多一个

< > kmod-nls-cp936................ Codepage 936 (china) (NEW)

12. 8m固件支持（新版本）

tools/firmware-utils/src/mktplinkfw.c 中代码:

static struct flash\_layout layouts[] = {

{

.id = "4M",

.fw\_max\_len = 0x3c0000,

.kernel\_la = 0x80060000,

.kernel\_ep = 0x80060000,

.rootfs\_ofs = 0x140000,

}, {

.id = "4Mlzma",

.fw\_max\_len = 0x3c0000,

.kernel\_la = 0x80060000,

.kernel\_ep = 0x80060000,

.rootfs\_ofs = 0x100000,

}, {

.id = "8M",

.fw\_max\_len = 0x7c0000,

.kernel\_la = 0x80060000,

.kernel\_ep = 0x80060000,

.rootfs\_ofs = 0x140000,

}, {

.id = "8Mlzma",

.fw\_max\_len = 0x7c0000,

.kernel\_la = 0x80060000,

.kernel\_ep = 0x80060000,

.rootfs\_ofs = 0x100000,

}, {

/\* terminating entry \*/

}

};

static struct board\_info boards[] = { }, {

.id = "TL-WR703Nv1",

.hw\_id = HWID\_TL\_WR703N\_V1,

.hw\_rev = 1,

.layout\_id = "4Mlzma", //只改此, 8Mlzma }, { }

13. Luci 主题修改

编辑 ./feeds/luci/modules/luci-base/root/etc/config/luci

默认主题： option mediaurlbase /luci-static/bootstrap

默认语言： option lang zh-cn

14. Luci 页面文字修改

编辑 ./feeds/luci/modules/luci-base/po/zh-cn/base.po

进阶设置

1. /trunk/tools/firmware-utils/src/mktplinkfw.c

“fw max len”，默认3c0000是4M的;改成7c0000是8M; 改成fc0000是16M

1. 修改target/linux/ar71xx/files/arch/mips/ar71xx/mach-tl-wr841n.c（找相应文件）

|  |  |  |
| --- | --- | --- |
| 4M | 8M | 16M |
| 41 .name = "u-boot",   42 .offset = 0,   43 .size = 0x020000,   44 .mask\_flags = MTD\_WRITEABLE,   45 }, {   46 .name = "kernel",   47 .offset = 0x020000,   48 .size = 0x140000,   49 }, {   50 .name = "rootfs",   51 .offset = 0x160000,   52 .size = 0x290000,   53 }, {   54 .name = "art",   55 .offset = 0x3f0000,   56 .size = 0x010000,   57 .mask\_flags = MTD\_WRITEABLE,   58 }, {   59 .name = "firmware",   60 .offset = 0x020000,   61 .size = 0x3d0000,   62 }   63 }; | 41 .name = "u-boot",   42 .offset = 0,   43 .size = 0x020000,   44 .mask\_flags = MTD\_WRITEABLE,   45 }, {   46 .name = "kernel",   47 .offset = 0x020000,   48 .size = 0x140000,   49 }, {   50 .name = "rootfs",   51 .offset = 0x160000,   52 .size = 0x690000,   53 }, {   54 .name = "art",   55 .offset = 0x7f0000,   56 .size = 0x010000,   57 .mask\_flags = MTD\_WRITEABLE,   58 }, {   59 .name = "firmware",   60 .offset = 0x020000,   61 .size = 0x7d0000,   62 }   63 }; | 41 .name = "u-boot",   42 .offset = 0,   43 .size = 0x020000,   44 .mask\_flags = MTD\_WRITEABLE,   45 }, {   46 .name = "kernel",   47 .offset = 0x020000,   48 .size = 0x140000,   49 }, {   50 .name = "rootfs",   51 .offset = 0x160000,   52 .size = 0xe90000,   53 }, {   54 .name = "art",   55 .offset = 0xff0000,   56 .size = 0x010000,   57 .mask\_flags = MTD\_WRITEABLE,   58 }, {   59 .name = "firmware",   60 .offset = 0x020000,   61 .size = 0xfd0000,   62 }   63 }; |
| "u-boot"  0X00000000-0X00020000  "kernel"  0X00020000-0X00160000  "rootfs"  0X00160000-0X003f0000  "rootfs\_data"  0X00304000-0X003f0000  "art"  0X003f0000-0X00400000  "firmware"  0X00020000-0X003f0000 | "u-boot"  0X00000000-0X00020000  "kernel"  0X00020000-0X00160000  "rootfs"  0X00160000-0X007f0000  "rootfs\_data"  0X00620000-0X007f0000  "art"  0X007f0000-0X00800000  "firmware"  0X00020000-0X007f0000 | "u-boot"  0X00000000-0X00020000  "kernel"   0X00020000-0X00160000  "rootfs"   0X00160000-0X00ff0000  "rootfs\_data"   0X00380000-0X00ff0000  "art"   0X00ff0000-0X01000000  "firmware"  0X00020000-0X00ff0000 |

操作记录

基于虚拟机VirtualBox 虚拟安装CentOS 7 mini, 配置好网络. openwrt 15.05

$ sudo yum install git

$ git clone git://git.openwrt.org/15.05/openwrt.git

$ ./scripts/feeds update –a

$ ./scripts/feeds install –a

$ make defconfig

此步会提示缺少哪些组件, 重复执行 sudo yum –y install xxx yyy zzz, 然后再执行 make defconfig,

知道没有错误

$ make menuconfig

只选三项,其余默认, 以后可以添加

选择CPU型号Target System—–Atheros AR71xx/AR7240/AR913x/AR934x

选择路由型号Target Profile—-TP-LINK TL-MR3420 v1

添加luci

LuCI—>Collections—– <\*> luci

LuCI—>Translations—- <\*> luci-i18n-chinese

// 此时按需要修改固件大小,见上.

$ make V=99