

1. Openwrt Version

The version name: Barrier Breaker, version No.: 14.07

2. Openwrt Develop Environment

The develop virtual machine in Baidu network disk (using vmware virtual machine):

Link: https://pan.baidu.com/s/1ljqsI9xn-6zRkM_LKq22RA

Extraction code: gpxg, this is the disk image of vmware. The username of the virtual machine is hlk, Password is 12345678

Link: <https://pan.baidu.com/s/1RqU9sRh2iZZ-Y3WNPfYpdg>

Extraction code: 55sy, this is the disk image of virtualbox. The username of the virtual machine is hlk, Password is 12345678

The virtual machine comes with the Openwrt 14.07 SDK source code, /home/hlk/mtkopenwrt, which already has the default configuration and can be compiled directly.

SDK source package link: <https://pan.baidu.com/s/1X4Om05zAaNAOurfiGuUPxA>

Extraction code is unhb

3. Openwrt configuration compiler

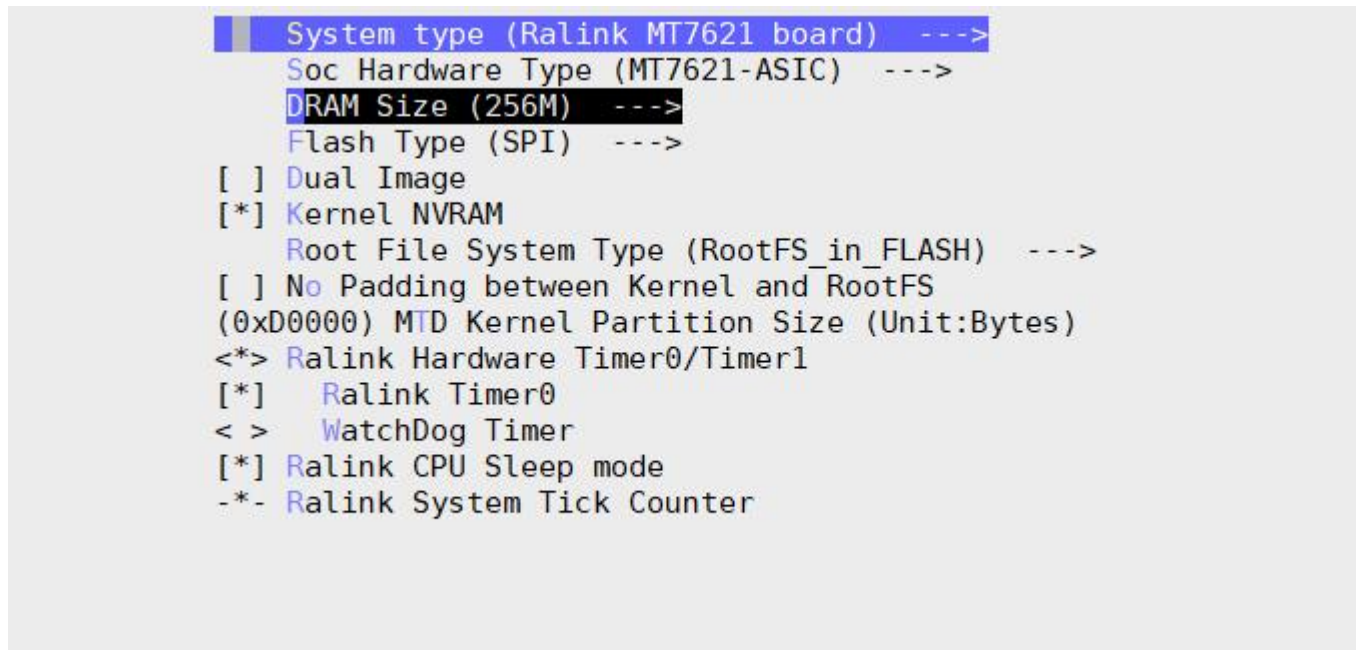
There is a default configuration in the SDK, which satisfies the basic functions of routing. Customers can also customize the configuration according to their own needs.

Command: make menuconfig



Modify the configuration of the default memory size

make kernel_menuconfig ->Machine selection ---> DRAM Size (256M) --->



Use the command make V=99

The compilation results are saved in the bin/ramips/ directory

Generate firmware name: openwrt-ramips-mt7621-mt7621-squashfs-sysupgrade.bin

We have not tried other configuration items, so we don't know the other specific functions

Notes:

Kmod-mt7628sta cannot be used, please do not select, if you need sta function, please select ap-client support under kmod-mt7628 driver

```
<> kmod-mt7628sta..... MTK MT7628 wifi STA driver --->
```

Use the command make V=99

The compilation result is saved in the bin/ramips/ directory

Generate firmware name: openwrt-ramips-mt7628-mt7628-squashfs-sysupgrade.bin

4. Use the reg command to control the registers of 7621

5. Openwrt factory configuration restoration method

Enter at the command line:

```
umount /dev/mtdblock6; firstboot
```

firstboot enter Y to confirm to restore the default

Openwrt will clear the existing configuration information and restore to the default factory configuration

6. Network port configuration in Openwrt

7. Add your own application in Openwrt and compile it into the firmware

The following is an example of helloworld : how to add an application in openwrt

Create a directory helloworld under package

Create a Makefile file in the helloworld directory: see the attachment for the file:



Makefile

Makefile rule description, detailed documentation can refer to :
<https://openwrt.org/docs/guide-developer/packages>

```

include $(TOPDIR)/rules.mk
PKG_NAME:=helloworld
PKG_RELEASE:=1

PKG_BUILD_DIR:=$(BUILD_DIR)/helloworld
include $(INCLUDE_DIR)/package.mk

define Package/helloworld
  SECTION:=HELLO_WORLD
  CATEGORY:=HELLO_WORLD
  TITLE:=Hello World App
endef

define Package/helloworld/description
  Hello World Application
endef

define Build/Prepare
  mkdir -p $(PKG_BUILD_DIR)
  cp -rfd ./src/* $(PKG_BUILD_DIR)/
endef

define Build/Compile
  $(MAKE) -C $(PKG_BUILD_DIR)
endef

define Package/helloworld/install
  $(INSTALL_DIR) $(1)/usr/bin
  $(INSTALL_DIR) $(1)/etc/init.d
  $(INSTALL_DIR) $(1)/etc/config
  $(INSTALL_BIN) $(PKG_BUILD_DIR)/helloworld $(1)/usr/bin
  $(INSTALL_BIN) ./files/hello.sh $(1)/usr/bin
  $(INSTALL_BIN) ./files/hello.init $(1)/etc/init.d/helloworld
endef

$(eval $(call BuildPackage,helloworld))

```

编译目录，实际位置在：
build_dir/target-mipsel_24kec+dsp_uClibc-0.9.33.2/helloworld/

把package/helloworld/src目录下的代码copy到编译目录下，等待编译

编译命令

INSTALL_DIR，应用需要在固件的/usr/bin目录下安装应用，需要先创建/usr/bin目录

如果应用需要开机启动，需要在/etc/init.d目录下添加启动脚本，脚本见hello.init



helloworld.tar.gz

See the attachment for the sample code package:

Unzip directly to the package of openwrt

The src directory under the Helloworld directory is where the source code is stored:

The Makefile in the src directory in the example is actually used to compile the code.

Customers can also write makefiles according to their own needs, there are a few things that need to be paid attention to

a. The name of the openwrt compilation tool is: cross compiler starting with mipsel-openwrt-linux-uclibc-

Cannot directly use mipsel-openwrt-linux-uclibc-ld for linking

B. If you use the Makefile included in the example, it is possible for simple projects, and it can edit the src directory

All .c files are compiled

Description of Makefile

```

# Date: 2012/10/24
#
# Usage:
# $ make          Compile and link (or archive)
# $ make clean    Clean the objectives and target.
#####
CROSS_COMPILE=mipsel-openwrt-linux-uclibc
OPTIMIZE := #-O2
WARNINGS := #-Wall -Wno-unused -Wno-format -Wno-strict-aliasing
DEFS :=
EXTRA_CFLAGS := #-DDEBUG #-m32
EXTRA_CFLAGS :=

# 定义一些变量，编译链接时使用
INC_DIR = ./include #include目录
SRC_DIR = ./common #源代码目录，表示编译时去当前SRC_DIR目录下查找源文件
OBJ_DIR = out #目标文件目录，表示编译后生成的.o文件存放目录，用以链接使用
EXTRA_SRC = #一些额外增加或者文件比较分散的源文件
EXCLUDE_FILES = #排除的文件列表

SUFFIX = c #源文件后缀，表示源代码可以支持的源文件后缀
TARGET := helloworld #生成目标名称
#TARGET_TYPE := ar #目标类型，表示是静态链接目标文件
#TARGET_TYPE := so #表示是动态链接目标文件
TARGET_TYPE := app #表示是生成应用程序

#####
# Do not change any part of them unless you have understood this script very well #
# This is a kind remind. #
#####

#FUNC# Add a new line to the input stream.
#定义一个变量
define add_newline
$1

```

openwrt的交叉编译工具的前缀

工程中需要包含头文件的路径

可以在这源代码路径

需要排除不编译的文件列表

需要编译的文件后缀，这里是只编译C文件

编译生成的程序名

编译为应用还是库，库可以使用so

After the file is created, use make menuconfig to configure openwrt:



After selecting, enter the project directory and perform make. After the compilation is completed, the firmware will be upgraded.

Compile-time library dependency issues:

If you use the pthread multithreading library

You can add the following in the outer Makefile

```

include $(TOPDIR)/rules.mk

PKG_NAME:=helloworld
PKG_RELEASE:=1

PKG_BUILD_DIR:=$(BUILD_DIR)/helloworld

include $(INCLUDE_DIR)/package.mk

define Package/helloworld
    SECTION:=HELLO_WORLD
    CATEGORY:=HELLO_WORLD
    TITLE:=Hello World App
    DEPENDS := +libpthread
endef

define Package/helloworld/description
    Hello World Application
endef

define Build/Prepare
    mkdir -p $(PKG_BUILD_DIR)
    cp -rfd ./src/* $(PKG_BUILD_DIR)/
endef

define Build/Compile
    $(MAKE) -C $(PKG_BUILD_DIR)
endef

define Package/helloworld/install
    $(INSTALL_DIR) $(1)/usr/bin
    $(INSTALL_DIR) $(1)/etc/init.d
    $(INSTALL_DIR) $(1)/etc/config
    $(INSTALL_BIN) $(PKG_BUILD_DIR)/helloworld $(1)/usr/bin
    $(INSTALL_BIN) ./files/hello.sh $(1)/usr/bin
    $(INSTALL_BIN) ./files/hello.init $(1)/etc/init.d/helloworld
endef

$(eval $(call BuildPackage,helloworld))
~
~

```

There is another way to cheat the openwrt compilation process at compile time

Add in Makefile

```

define Package/helloworld/extra_provides
    echo "libpthread.so.0"

```

In this way, when compiling, there will be no errors due to lack of a library. But the application can't actually run, you need to copy the corresponding library to the system library directory.

Power-on startup script:

Openwrt's power-on executes the scripts under /etc/init.d and executes them in sequence according to the size of the START variable in the script.



hello.init

Refer to the following writing of the startup script:


```

luke@ub:~/mtkopenwrt/package/helloworld/files$ cat hello.init
#!/bin/sh /etc/rc.common

START=99
start() {
    echo "start helloworld" >> /tmp/helloworld
}
stop() {
    echo "stop helloworld" >> /tmp/helloworld
}
luke@ub:~/mtkopenwrt/package/helloworld/files$

```

启动数字，数字越大越晚启动

启动时执行的shell函数名字

执行/etc/init.d/helloworld stop 调用此函数脚本 停止应用执行

Uboot

The factory uboot source code download from the open source version github,
address: <https://github.com/gnubee-git/GnuBee-MT7621-uboot.git>



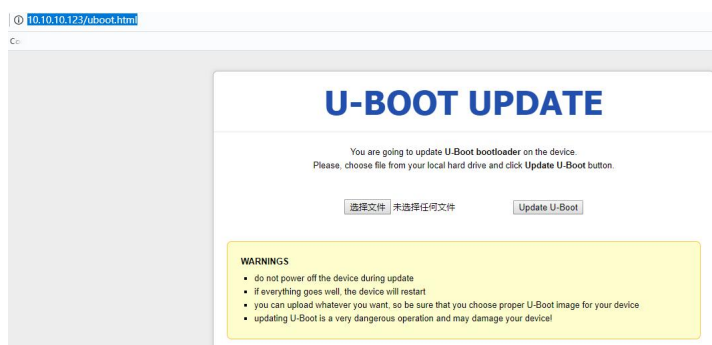
uboot.bin

Compiled uboot:

Uboot Upgrade method:

1.Press and hold the WPS button during startup, uboot enters the upgrade mode:

Note: The IP of the network port connected to the module on the PC side needs to be set to an IP page address on the same network segment as 10.10.10.123: <http://10.10.10.123/uboot.html>



2.Upgrade via serial command line

Type the number 9 to enter the TFTP upgrade uboot mode :

```

9: System Load Boot Loader then write to Flash via TFTP.
Warning!! Erase Boot Loader in Flash then burn new one. Are you sure?(Y/N)

```

Type Y to enter into parameter configuration

```
9: System Load Boot Loader then write to Flash via TFTP.  
Warning!! Erase Boot Loader in Flash then burn new one. Are you sure?(Y/N)  
Please Input new ones /or Ctrl-C to discard  
Input device IP (10.10.10.123) ==:10.10.10.123
```

Input device IP : Configure the IP of the 7621 module

Press enter

```
9: System Load Boot Loader then write to Flash via TFTP.  
Warning!! Erase Boot Loader in Flash then burn new one. Are you sure?(Y/N)  
Please Input new ones /or Ctrl-C to discard  
Input device IP (10.10.10.123) ==:10.10.10.123  
Input server IP (10.10.10.3) ==:10.10.10.3
```

:

Input Server IP : Configure the IP on the PC network port connected to the 7621

```
9: System Load Boot Loader then write to Flash via TFTP.  
Warning!! Erase Boot Loader in Flash then burn new one. Are you sure?(Y/N)  
Please Input new ones /or Ctrl-C to discard  
Input device IP (10.10.10.123) ==:10.10.10.123  
Input server IP (10.10.10.3) ==:10.10.10.3  
Input Uboot filename ==:uboot.bin
```

Input Uboot filename: Enter the file name of the uboot that needs to be upgraded