

Creating a ROOter Build System

To create a OpenWrt build system that can be used to create ROOter images you need a computer that is running Linux.

This can be a Virtual Machine running in Windows or Linux running on a computer. The recommended Linux distros to use for this are *Linux Mint*, *Ubuntu 18.04* or *Debian*. ROOter images are made using Linux Mint 18.

This computer must have Internet access in order to download the files needed to create an OpenWrt build system.

Using the File Manager of your Linux distro, create a folder that will contain the build system. It is possible to have multiple build systems resident in this folder if you want to build images with different OpenWrt versions.

We will name this folder **OpenWrt** but it can be called anything.

Copy the Zip file that contains the ROOter build system creator into this folder. This Zip file will be named something like **Rooter18066.zip** which identifies it as creating a build system using OpenWrt 18.06.6.

Extract this Zip inside the folder. Depending on how you do the extraction you may now have another folder appear. If so then open that folder.

Inside will be two files named ***rooter-master.tar.gz*** and ***setup18066***. Open a Terminal in the folder which contains these files.

If you have not already added the required packages to your Linux distro to allow building OpenWrt you must edit ***setup18066*** to allow the addition of these packages.

```
setup18066
File Edit Search Options Help
1 #!/bin/sh
2
3 # all commands involving downloads are run twice to ensure all files are
4
5 # uncomment the two lines below if you are using Linux Mint 18
6
7 #sudo apt-get install build-essential subversion git-core libncurses5-dev
8 #sudo apt-get install build-essential subversion git-core libncurses5-dev
9
10 # uncomment the two lines below if you are using Ubuntu 18.04.1
11
12 #sudo apt -y install build-essential libncurses5-dev python unzip gawk g
13 #sudo apt -y install build-essential libncurses5-dev python unzip gawk g
14
15 # uncomment the two lines below if you are using Debian
16
17 #sudo apt install build-essential git unzip ncurses-dev libz-dev libssl-
18 #sudo apt install build-essential git unzip ncurses-dev libz-dev libssl-
19
20 # other versions of Linux will need different dependancies added in orde
21
22 git clone https://git.openwrt.org/openwrt/openwrt.git rooter18066
```

Remove the # at the start of the two lines that correspond to your Linux distro. Then save the file.

Now open a **Terminal** in the folder which contains these files.

At the prompt in **Terminal** enter

./setup18066

and press *Enter*.

```
dairyman@coolermaster ~/OpenWrt/working
File Edit View Search Terminal Help
dairyman@coolermaster ~/OpenWrt/working $ ./setup18066
```

You will be asked for your password that you use to log into the computer. Enter that and then press *Enter*.

The build system will be created in a folder called **Rooter18066** and the script will download all the necessary files and customize the build system with the ROOter scripts.

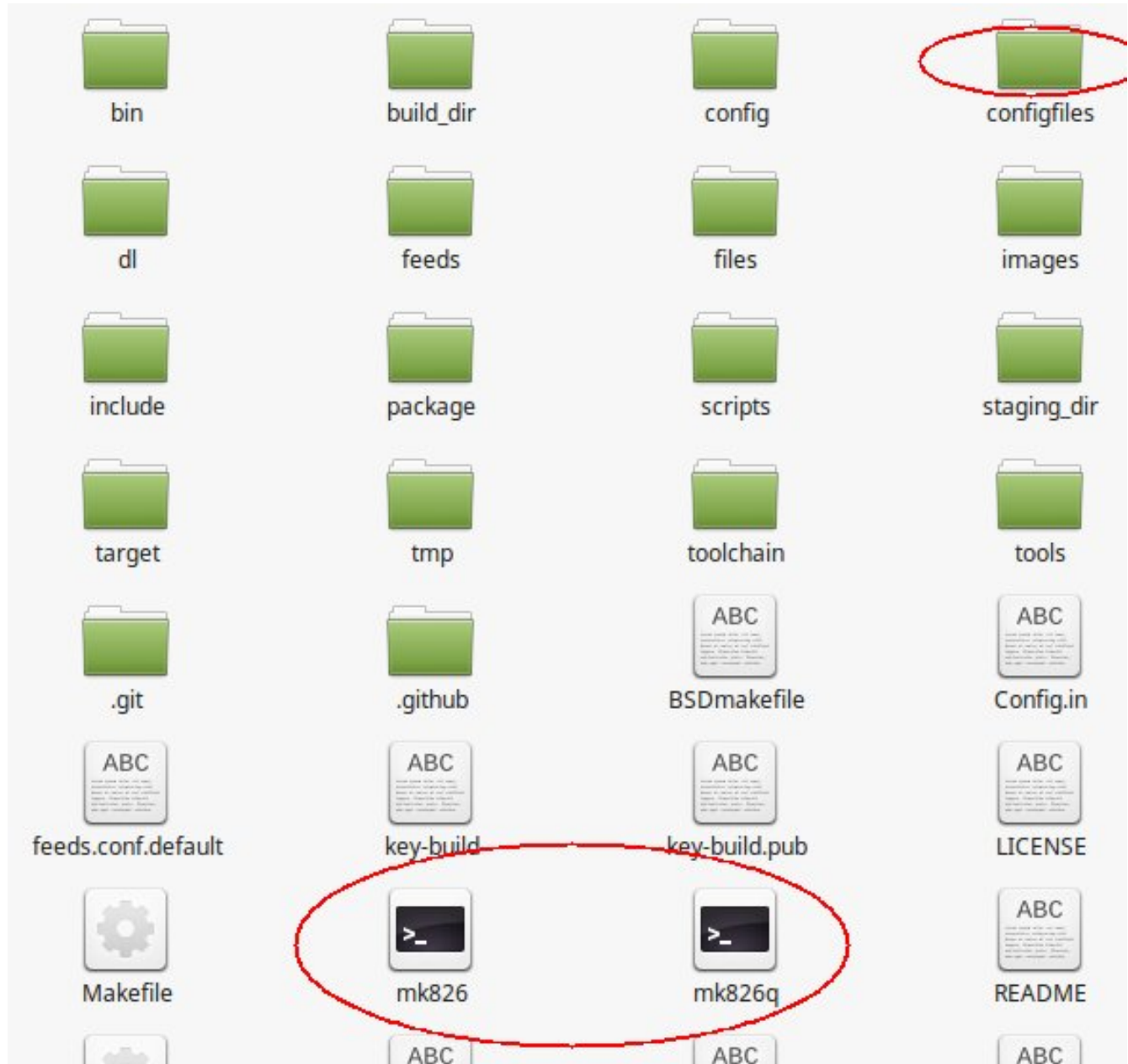
This will take some time but you can see the progress in the Terminal window. If you get a an error where it says “can’t Access site” then simply run **setup18066** again. It sometimes has problems making a connection to certain of the OpenWrt download sites.

When the creation process is complete you can close the **Terminal** window.

Use the File Manager to open the **Router18066** folder. Inside is a complete ROOter build system ready to create new images.

Creating ROOter Images

To create new firmware images that duplicate the look at feel of ROOter you must use the build scripts that are included in the build system and follow the correct procedure for creating configuration files for use by those scripts.



The important things in this folder are the **configfiles** folder and the **mk**** files.

The first thing you must do is turn on the display of hidden files in the File manager. In Linux Mint this is done by making the File Manager the focus and then pressing **Ctrl-H**. When this is done you will see some files whose name start with a period like **.gitignore**.

Configuration Files

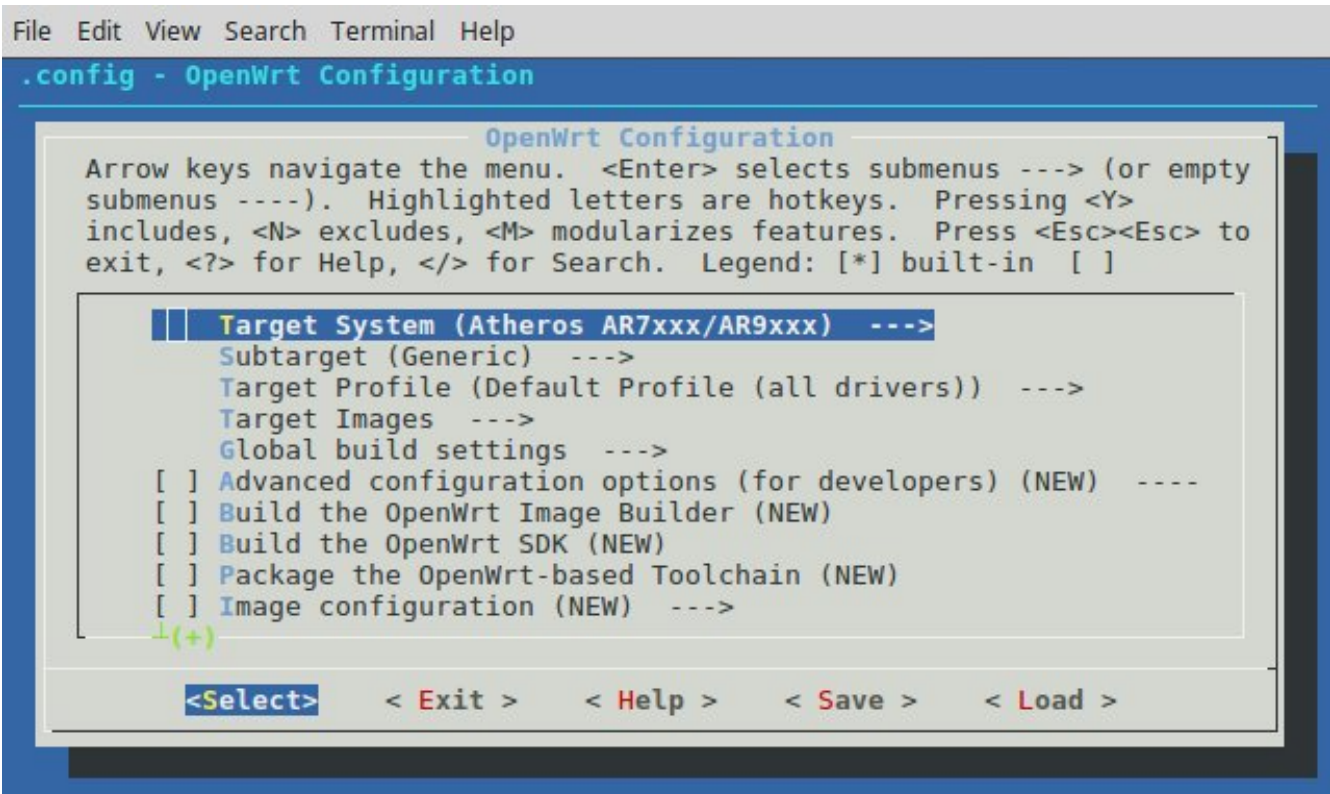
You can now start making the configuration file for your desired router. When creating a configuration file from scratch (rather than just changing an existing configuration) always delete the **.config** file from this folder. Note, this file has a period at the start of the name.

This is the file that will contain the router's configuration and, by deleting, you are starting fresh.

Open a **Terminal** in this folder and, at the prompt, enter

make menuconfig

and press enter. The **Terminal** window will then show :



```
File Edit View Search Terminal Help
.config - OpenWrt Configuration

OpenWrt Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

  [*] Target System (Atheros AR7xxx/AR9xxx) --->
      Subtarget (Generic) --->
      Target Profile (Default Profile (all drivers)) --->
      Target Images --->
      Global build settings --->
  [ ] Advanced configuration options (for developers) (NEW) ----
  [ ] Build the OpenWrt Image Builder (NEW)
  [ ] Build the OpenWrt SDK (NEW)
  [ ] Package the OpenWrt-based Toolchain (NEW)
  [ ] Image configuration (NEW) --->
  ↑(+)
```

<Select> <Exit> <Help> <Save> <Load>

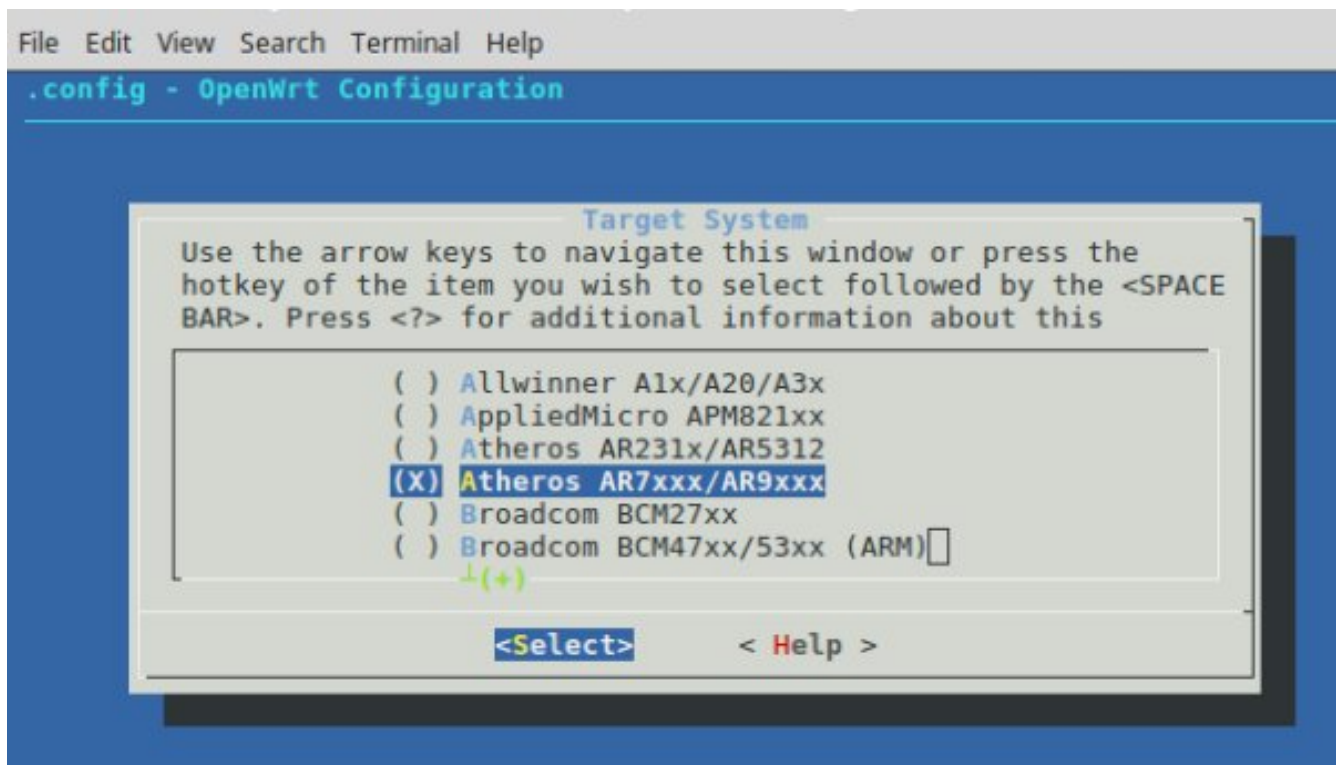
This window is navigated by using the keyboard. The **left** and **right** arrow keys move you across the commands at the bottom of the screen. The **up** and **down** arrow keys move you through the menu entries.

Pressing **Enter** will take you to the selected submenu if the highlighted bottom command is **<Select>**. If the highlighted bottom command is **<Exit>** then you will go to the previous submenu or, if at the main menu, will exit the Configuration program.

Target Selection

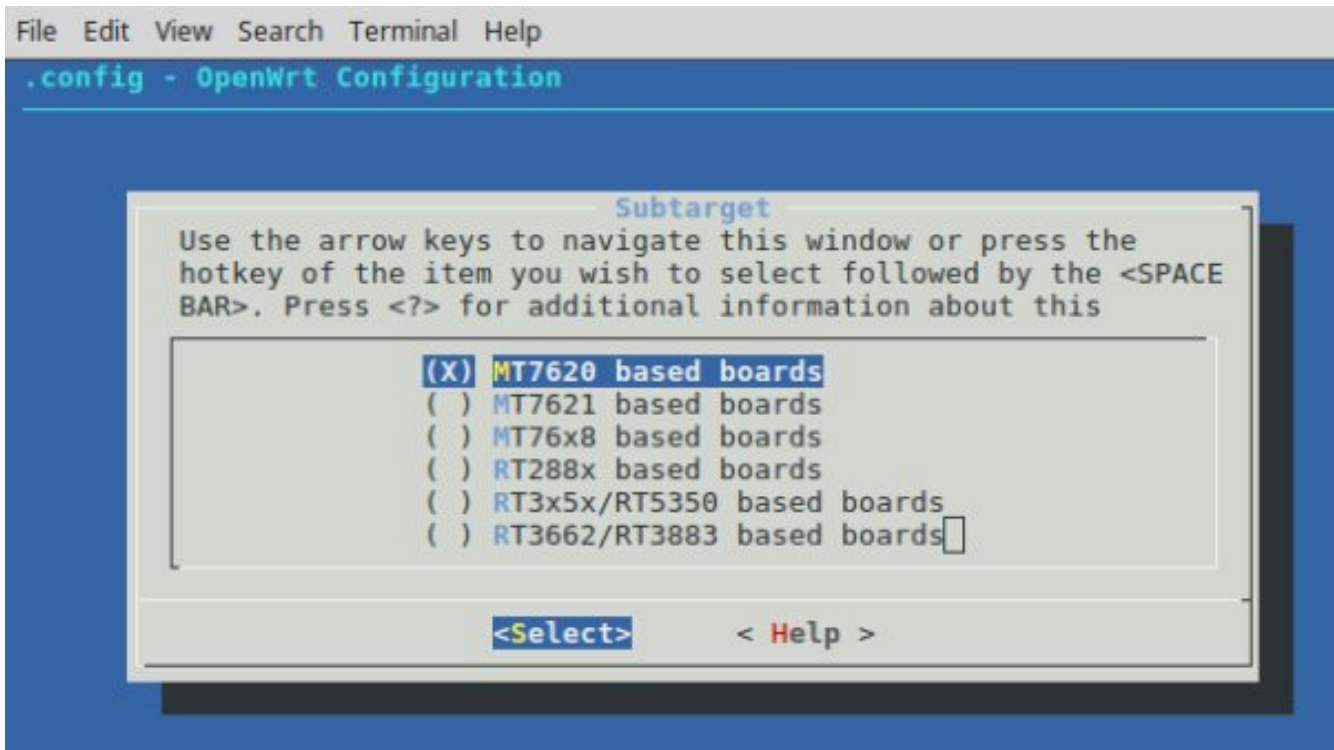
Since every image created by the build system is for a specific router you must tell the Configuration program what router you are building for. This is done by setting the correct **Target System**, **Subtarget** and **Target Profile**.

The **Target System** is used to select the processor type of the router. Use the arrow keys to highlight this menu entry and then press **Enter**.



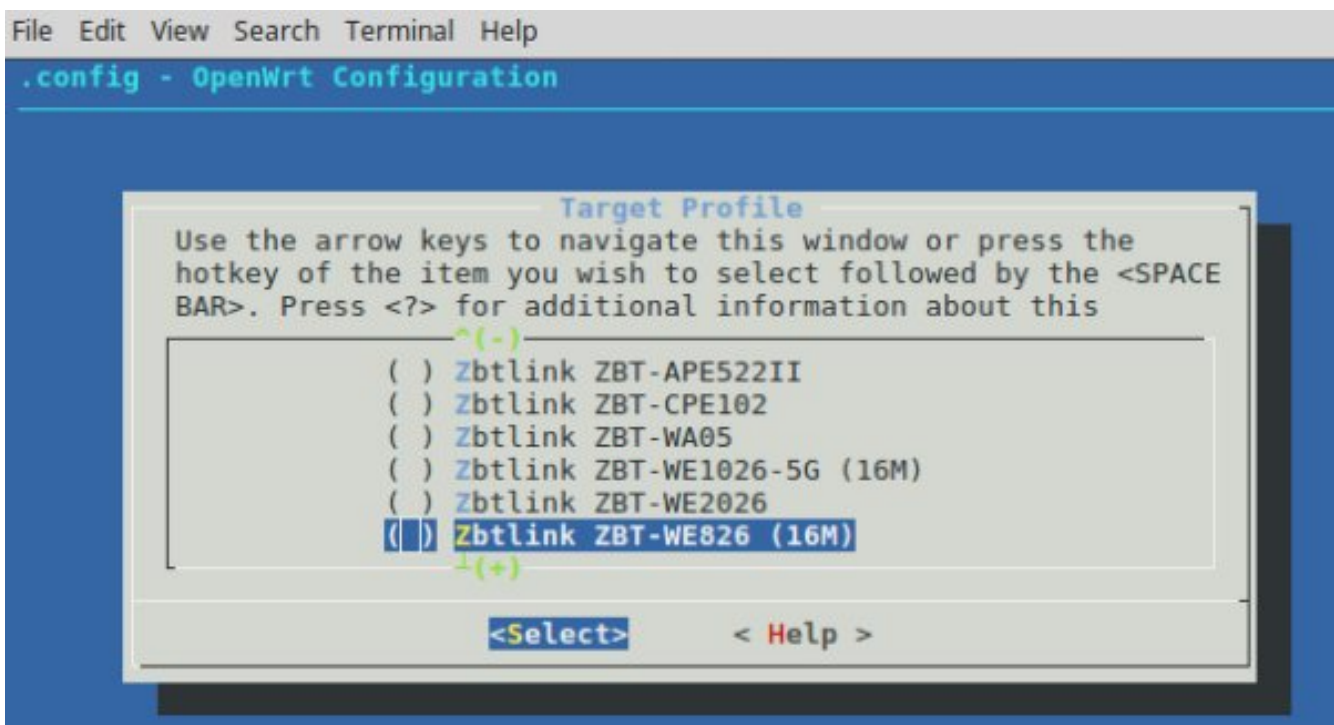
Use the arrow keys to scroll up and down this menu until you find the desired **Target System** then press **Enter** to select it.

The **Subtarget** menu (when it is present) is used to select among the different models of this processor.



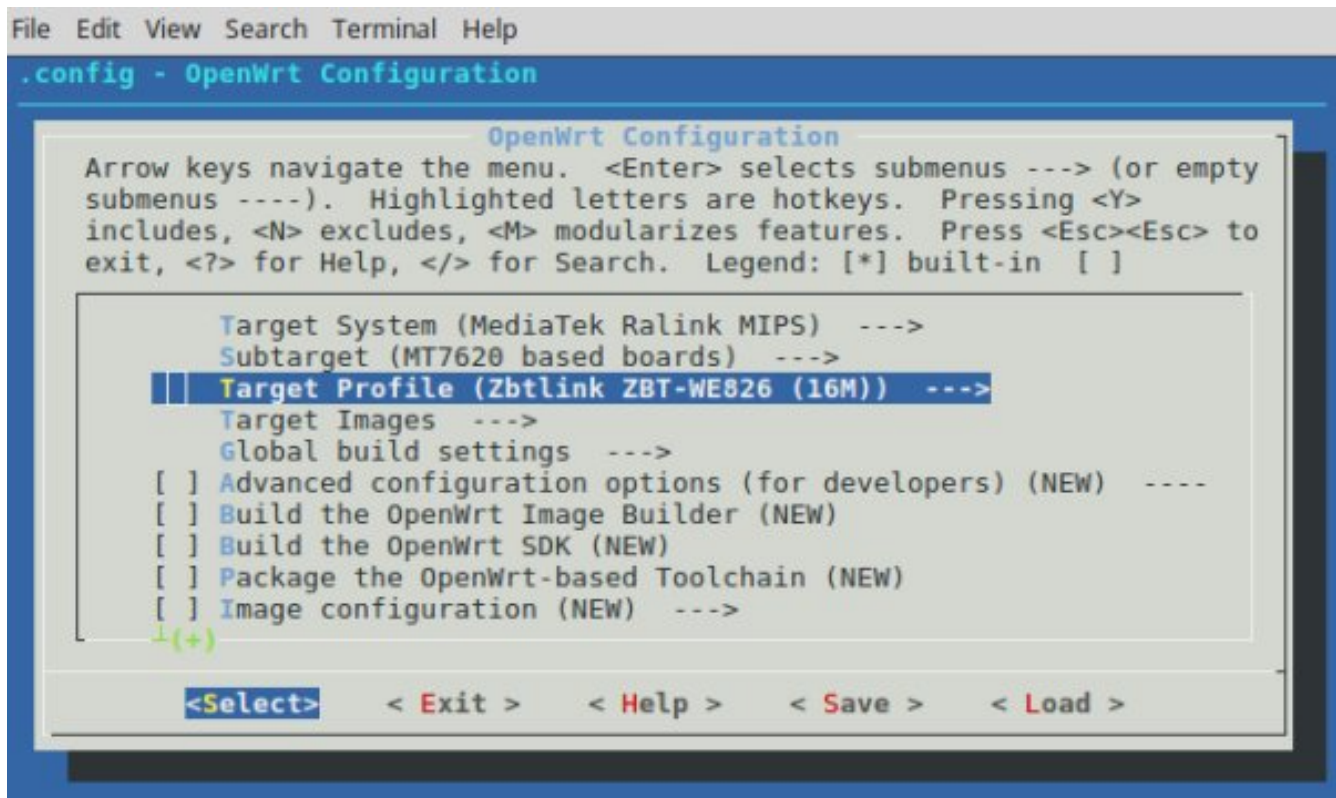
Again, use the arrow keys to scroll up and down this menu until you find the desired **Subtarget** and press **Enter** to select it.

Finally, the **Target Profile** is used to select the router model.



Use the arrow keys to scroll up and down this menu until you find the desired router model and then press **Enter** to select it.

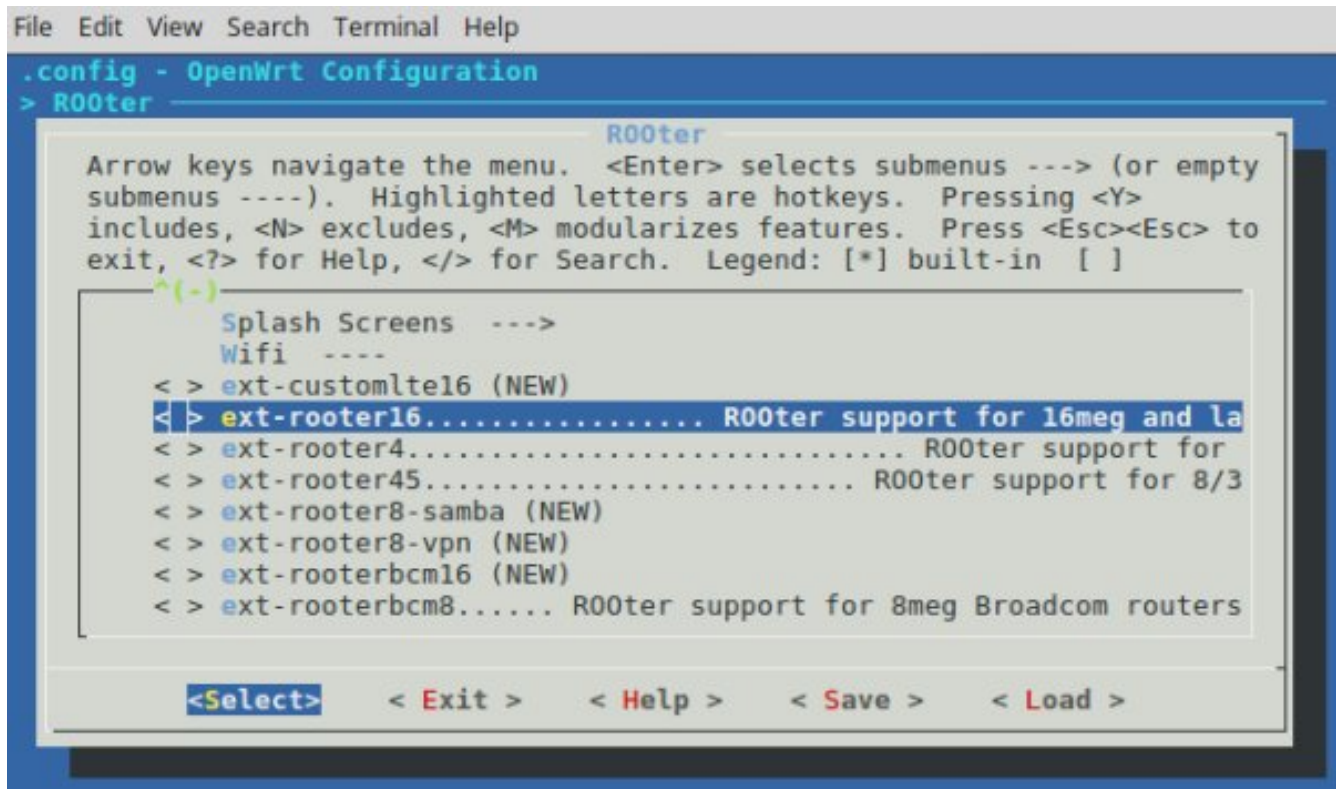
When you have completed the above steps you will be able to build a firmware for the desired router.



Package Selection

With target router selected in the Configuration program we can now select which packages we want to have included in the firmware. These packages will add extra features not found in a basic OpenWrt firmware.

To build a ROOter firmware, scroll down to the **ROOter** menu entry and press **Enter**.



```
File Edit View Search Terminal Help
.config - OpenWrt Configuration
> ROOter

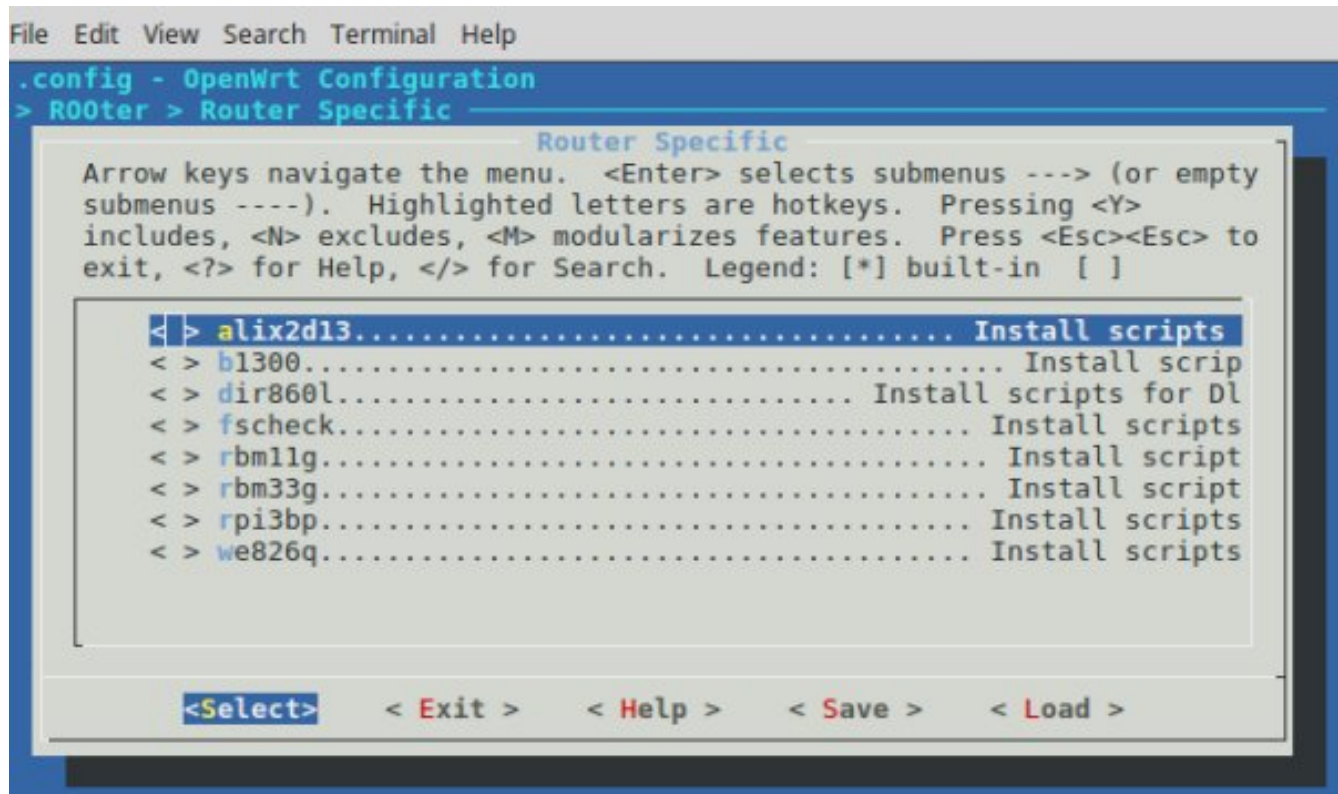
                                ROOter
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty
submenus ----). Highlighted letters are hotkeys. Pressing <Y>
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to
exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]

^(-)
  Splash Screens --->
  Wifi ----
  < > ext-customltel6 (NEW)
  < > * ext-router16..... ROOter support for 16meg and la
  < > ext-router4..... ROOter support for
  < > ext-router45..... ROOter support for 8/3
  < > ext-router8-samba (NEW)
  < > ext-router8-vpn (NEW)
  < > ext-routerbcm16 (NEW)
  < > ext-routerbcm8..... ROOter support for 8meg Broadcom routers

  <Select>  < Exit >  < Help >  < Save >  < Load >
```

Select the ROOter package that is designed for your router, based on the amount of flash memory and processor type. You do this by pressing **Y**. To deselect a package press **N**. A ***** will appear to the left of a package that has been selected.

For some routers you must also select extra ROOter packages designed specifically for those routers. To do this, scroll to the **Router Specific** menu and press **Enter**.



Use **Y** to select the extra package if one is available for your router.

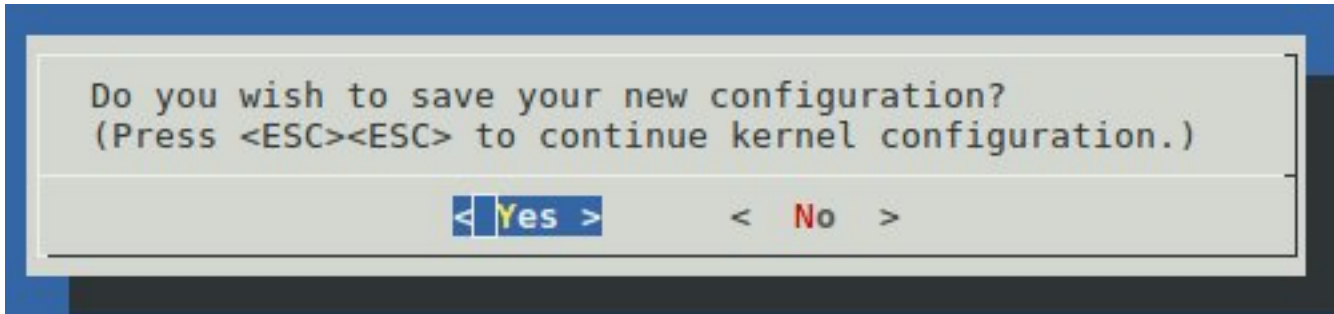
Use the left or right arrow key to highlight the **Exit** command at the bottom and press **Enter** to leave this menu.

Repeat this procedure in the ROOter menu to return to the main menu.

Exit the Configuration

With all the required packages selected you can now leave the Configuration program. Do this by using the left or right arrow key to highlight the **Exit** command at the bottom and then press **Enter**.

If you have modified the configuration file or have created a new file then you will be asked if you want to save it.



Use the arrow keys to select **Yes** and press **Enter**. Your configuration data will be saved in a file named **.config**. Note that this file name starts with a period.

Because you can build images for many routers in this build system we must keep the configuration file for each router in another folder so when you create another configuration file it doesn't overwrite the previous one.

To accomplish this we will rename the configuration file and move it to the **configfiles** folder. We will then tell our build script the name of this file so it can use it when building a new image.

The configuration file can be named anything but a good practice is to include an identifier in the name about which router this is to be used with. For example, for a configuration file for the WE826 we would rename **.config** to **.config_826** before moving it to the **configfiles** folder.

To aid in the creation of configuration files there is a table of routers, their target information and the packages required to build a working ROOter image. See **Table 1** for more information.

Build Scripts

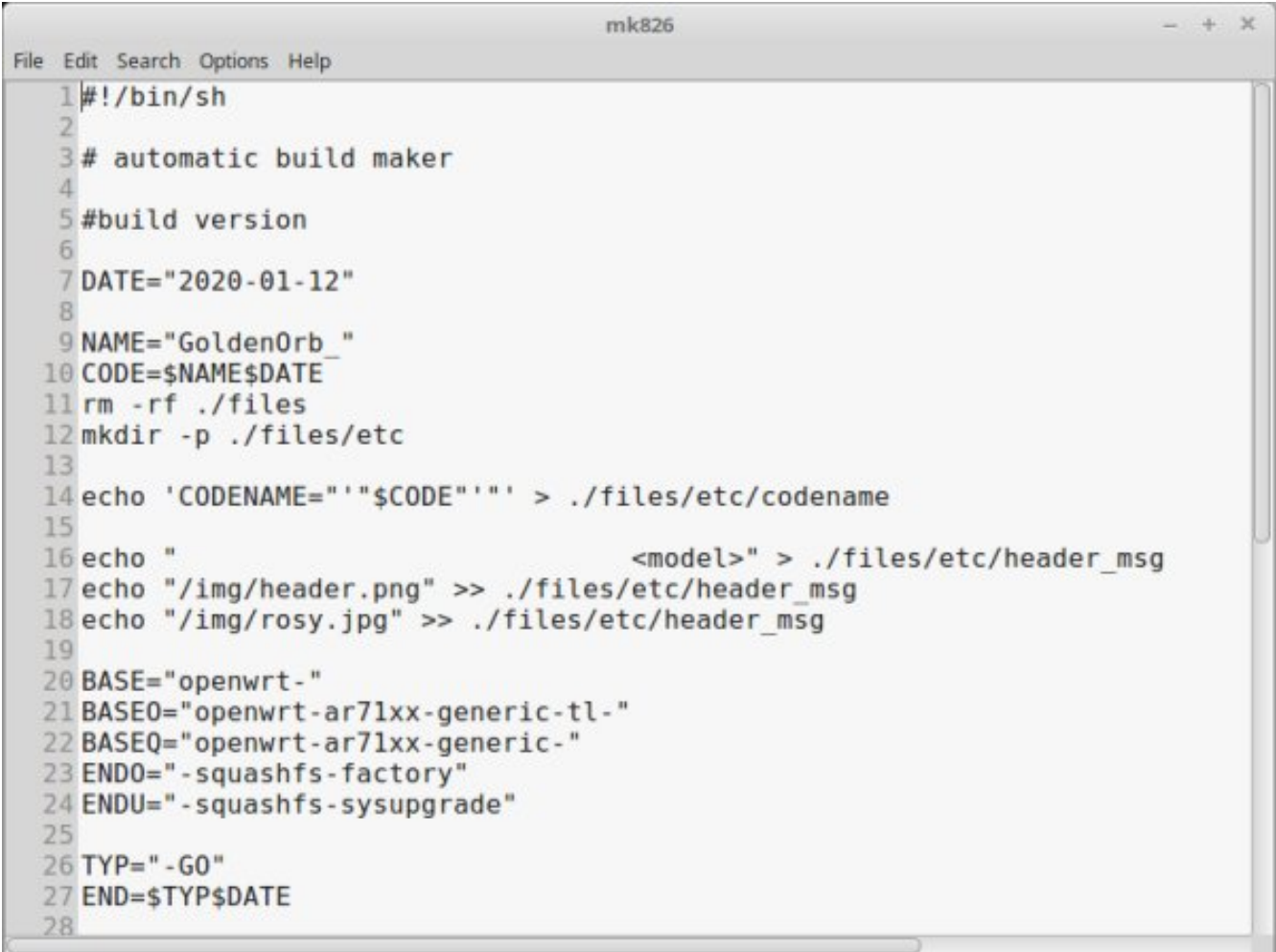
To build an image that has the look and feel of the ROOter GUI you must use the ROOter build scripts. These scripts will make the image from your configuration file, rename it to a ROOter name and Zip it up in a file for you.

The scripts also add files to the image so the GUI looks like a normal ROOter GUI.

Each image you build will have it's own build script that is customized just for it.

Let us look at how the build scripts are put together so you can see how to modify them to suit your desired router.

The first part of every script is the same. It creates several files needed by ROOter and sets the date used by the build. This date can be changed by editing line 7.



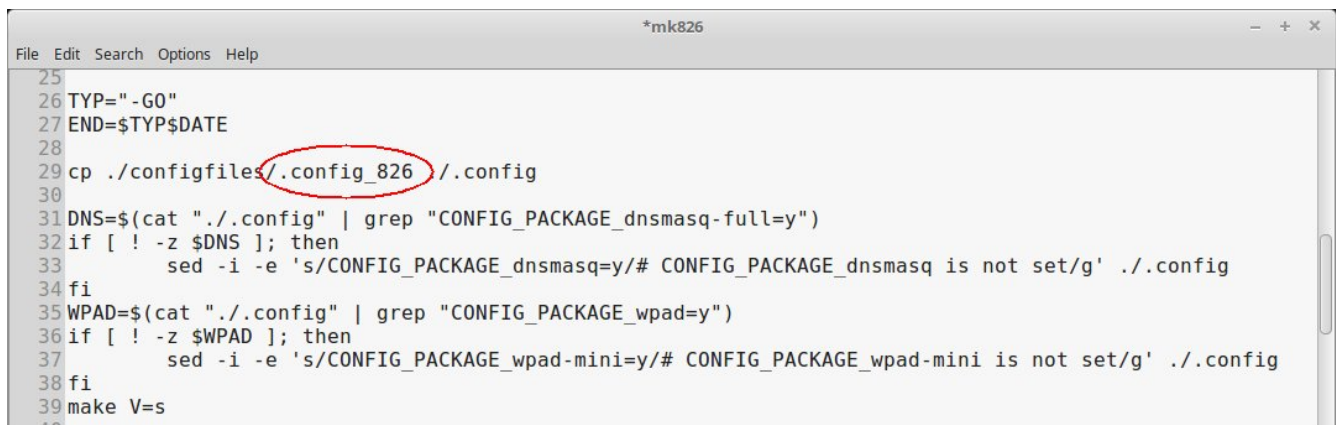
```
mk826
File Edit Search Options Help
1 #!/bin/sh
2
3 # automatic build maker
4
5 #build version
6
7 DATE="2020-01-12"
8
9 NAME="GoldenOrb_"
10 CODE=$NAME$DATE
11 rm -rf ./files
12 mkdir -p ./files/etc
13
14 echo 'CODENAME="" "$CODE"' > ./files/etc/codename
15
16 echo "                <model>" > ./files/etc/header_msg
17 echo "/img/header.png" >> ./files/etc/header_msg
18 echo "/img/rosy.jpg" >> ./files/etc/header_msg
19
20 BASE="openwrt-"
21 BASE0="openwrt-ar71xx-generic-tl-"
22 BASEQ="openwrt-ar71xx-generic-"
23 ENDO="-squashfs-factory"
24 ENDU="-squashfs-sysupgrade"
25
26 TYP="-GO"
27 END=$TYP$DATE
28
```

The second part copies the desired configuration file from the **configfiles** folder and renames it to **.config** so it can be used to build the image. It also alters the configuration file so there are no conflicts between versions of dnsmasq and wpad.

You must edit line 29 to change the name of the configuration file stored in **configfiles** to match the desired file. In this case the file is named **.config_826**.

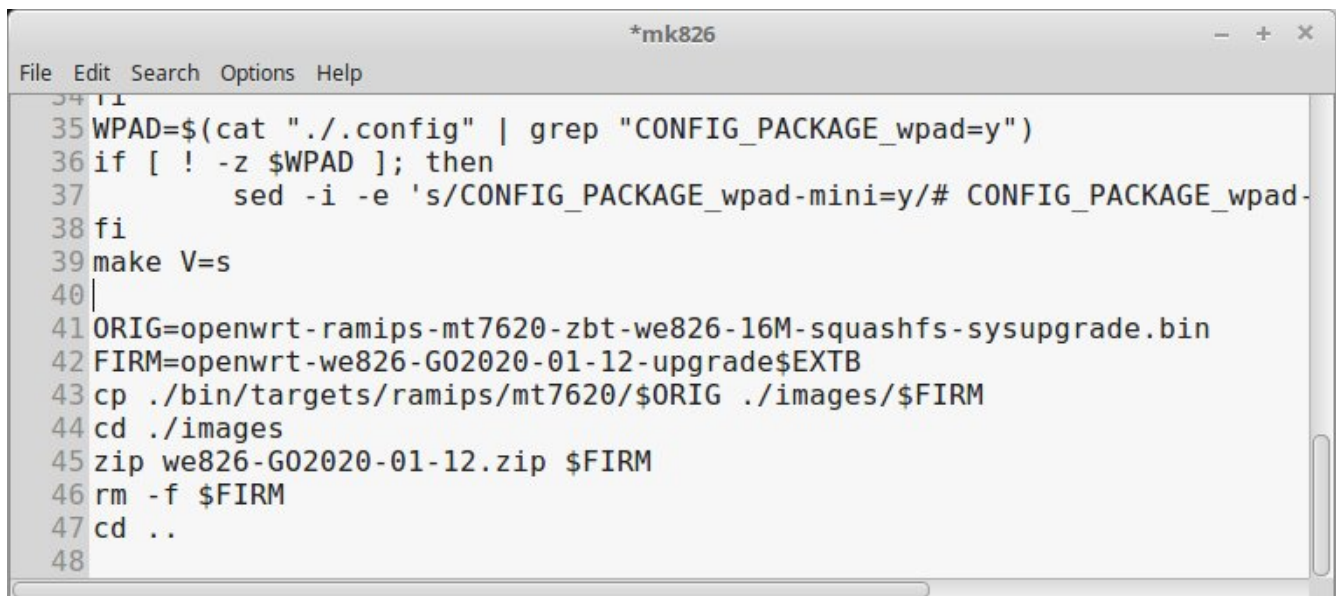
This is important.

Line 39 does the actual building of the image.



```
*mk826
File Edit Search Options Help
25
26 TYP="-G0"
27 END=$TYP$DATE
28
29 cp ./configfiles/.config_826 ./config
30
31 DNS=$(cat "./.config" | grep "CONFIG_PACKAGE_dnsmasq-full=y")
32 if [ ! -z $DNS ]; then
33     sed -i -e 's/CONFIG_PACKAGE_dnsmasq=y/# CONFIG_PACKAGE_dnsmasq is not set/g' ./config
34 fi
35 WPAD=$(cat "./.config" | grep "CONFIG_PACKAGE_wpad=y")
36 if [ ! -z $WPAD ]; then
37     sed -i -e 's/CONFIG_PACKAGE_wpad-mini=y/# CONFIG_PACKAGE_wpad-mini is not set/g' ./config
38 fi
39 make V=s
40
```

The third part of the script moves the newly created image file from where it was stored by the build system to the **images** folder, renames it in the ROOter manner and Zips it up into a file.

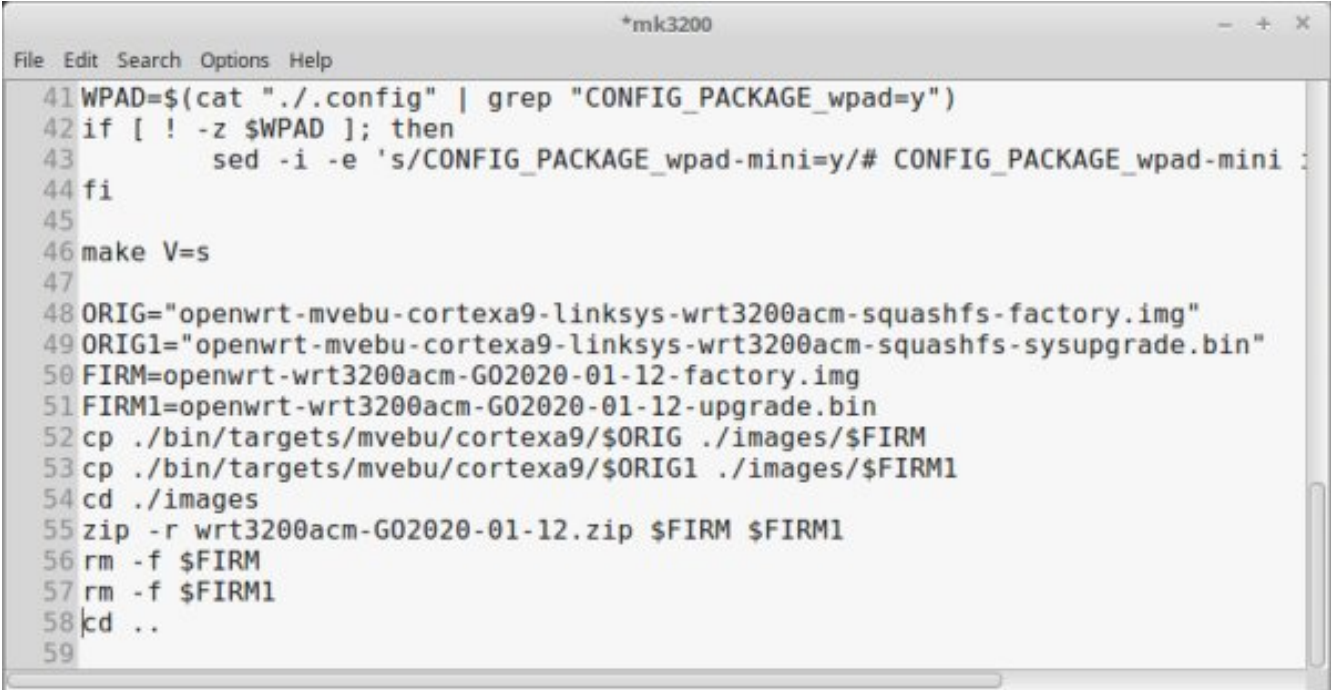


```
*mk826
File Edit Search Options Help
34 fi
35 WPAD=$(cat "./.config" | grep "CONFIG_PACKAGE_wpad=y")
36 if [ ! -z $WPAD ]; then
37     sed -i -e 's/CONFIG_PACKAGE_wpad-mini=y/# CONFIG_PACKAGE_wpad-
38 fi
39 make V=s
40 |
41 ORIG=openwrt-ramips-mt7620-zbt-we826-16M-squashfs-sysupgrade.bin
42 FIRM=openwrt-we826-G02020-01-12-upgrade$EXTB
43 cp ./bin/targets/ramips/mt7620/$ORIG ./images/$FIRM
44 cd ./images
45 zip we826-G02020-01-12.zip $FIRM
46 rm -f $FIRM
47 cd ..
48
```

This part of the build script is very specific to the router that the image is being built for.

- Line 41 is the name of the image file that was created by the build system.
- Line 42 is the name we want to call the image file.
- Line 43 copies the file from where it was stored and renames it to the new name.
- Line 45 then zips it into a Zip file.

In some cases the router needs two images, one to flash from factory to ROOter and one to update an existing ROOter. Then the final part of the script looks different.



```
*mk3200
File Edit Search Options Help
41 WPAD=$(cat "./.config" | grep "CONFIG_PACKAGE_wpad=y")
42 if [ ! -z $WPAD ]; then
43     sed -i -e 's/CONFIG_PACKAGE_wpad-mini=y/# CONFIG_PACKAGE_wpad-mini :
44 fi
45
46 make V=s
47
48 ORIG="openwrt-mvebu-cortexa9-linksys-wrt3200acm-squashfs-factory.img"
49 ORIG1="openwrt-mvebu-cortexa9-linksys-wrt3200acm-squashfs-sysupgrade.bin"
50 FIRM=openwrt-wrt3200acm-G02020-01-12-factory.img
51 FIRM1=openwrt-wrt3200acm-G02020-01-12-upgrade.bin
52 cp ./bin/targets/mvebu/cortexa9/$ORIG ./images/$FIRM
53 cp ./bin/targets/mvebu/cortexa9/$ORIG1 ./images/$FIRM1
54 cd ./images
55 zip -r wrt3200acm-G02020-01-12.zip $FIRM $FIRM1
56 rm -f $FIRM
57 rm -f $FIRM1
58 cd ..
59
```

- Line 48 is the name of the flash from factory image
- Line 49 is the name of the upgrade image
- Line 50 is the new name of the flash from factory image
- Line 51 is the new name of the upgrade image
- Line 52 copies the factory file from where it was stored and renames it to the new name.
- Line 53 copies the upgrade file from where it was stored and renames it to the new name.
- Line 55 then zips them into a Zip file.

Once you have created your new build script or are using an existing one simply open a **Terminal** in the build system folder and run the script.

The sample build scripts included in the package are named **mk826** and **mk3200** so to execute them you would enter

./mk826

in the **Terminal**. The image will then be created and the end result is placed in the **image** folder.

Be aware that the first time an image is created in a new build system it must download a number of files needed to compile the image. It also has to compile every part of the **Target System** so this will take a number of hours to complete.

After the first build, images from that **Target System** will only take a short amount of time to compile.

In order to make it easier to create your own build scripts a file named **mkimage** is included in the package. This is the first two sections of a typical build script so all you need to do is

- change the date,
- change the name of the configuration file
- add in the copy/rename/zip portion of the script.

Also included is a file named **imagecopy** which contains the copy/rename/zip portion of a build script for a number of routers. Just find the section that matches your router and copy/paste it onto the end of **mkimage** and you will have a working build script.

Table 1**Router Configuration Information**

This is a table of all supported routers with the Target information and Package information required to create a ROOter image.

The **ext-router16** and **ext-routerbcm16** packages are found in the main **ROOter** menu. The **ext-wifi** package is found in the **Wifi** submenu. The other packages are found in the **Router Specific** submenu.

The **kmod-ath9k** and **kmod-rt2800-pci** packages are found in the **Kernel** menu in the **Wifi** submenu.

Router Model	Target System	Subtarget	Target Profile	Packages
PC Engines Alix 2D13	x86	AMD Geode based systems	generic	ext-router16 ext-wifi alix2d13
Gl.iNet MT300a	Mediatek Ralink MIPS	MT7620 based boards	GL-INET MT300a	ext-router16
Gl.iNet MT300n	Mediatek Ralink MIPS	MT7620 based boards	GL-INET MT300n	ext-router16
Gl.iNet MT300n-v2	Mediatek Ralink MIPS	MT76x8 based boards	GL-INET GL- MT300N-V2	ext-router16
Huawei HG556a-A	Broadcom BCM63xx	generic	Huawei Echolife HG556a rev A	ext-router16 kmod-ath9k
Huawei HG556a-B	Broadcom BCM63xx	generic	Huawei Echolife HG556a rev B	ext-router16 kmod-ath9k
Huawei HG556a-C	Broadcom BCM63xx	generic	Huawei Echolife HG556a rev C	ext-router16 kmod-rt2800- pci
ZBT WE826-T	Mediatek Ralink MIPS	MT7620 based boards	ZBT WE826 (16M)	ext-router16
Dlink DIR860L	Mediatek Ralink MIPS	MT7621 based boards	Dlink DIR860L	ext-router16 dir860l

Linksys WRT1200AC	Marvell EBU Armada	Marvell Armada 37x/38x/xp	Linksys WRT1200AC (Caiman)	ext-router16
Linksys WRT1900AC v1	Marvell EBU Armada	Marvell Armada 37x/38x/xp	Linksys WRT1900AC (Mamba)	ext-router16
Linksys WRT1900AC v2	Marvell EBU Armada	Marvell Armada 37x/38x/xp	Linksys WRT1900AC (Cobra)	ext-router16
Linksys WRT1900ACS	Marvell EBU Armada	Marvell Armada 37x/38x/xp	Linksys WRT1900ACS (Shelby)	ext-router16
Linksys WRT3200AC M	Marvell EBU Armada	Marvell Armada 37x/38x/xp	Linksys WRT3200AC M (Rango)	ext-router16
Linksys WRT32X	Marvell EBU Armada	Marvell Armada 37x/38x/xp	Linksys WRT32X (Venom)	ext-router16
TP-link C2600	Qualcomm Atheros IPQ806x		Tplink C2600	ext-router16
ZBT WG3526	Mediatek Ralink MIPS	MT7621 based boards	ZBT WG3526	ext-router16
Netgear WNDR3700v4	Atheros AR7xxx	generic with NAND flash	Netgear WNDR3700v4	ext-router16
Netgear WNDR4300	Atheros AR7xxx	generic with NAND flash	Netgear WNDR3700v4	ext-router16
TP-Link WDR4900	Freescale MPC85xx	generic	TP-LINK TL- WDR4900	ext-router16
Netgear R6220	Mediatek Ralink MIPS	MT7621 based boards	Netgear R6220	ext-router16
Sanlinking 7620-D240	Mediatek Ralink MIPS	MT7620 based boards	Sanlinking 7620-D240	ext-router16

Linksys EA8500	Qualcomm Atheros IPQ806x		Linksys EA8500	ext-router16
Netgear R7800	Qualcomm Atheros IPQ806x		Netgear Nighthawk X4S R7800	ext-router16
PCEngines APU2C4	x86	x86_64	Generic	ext-router16 ext-wifi apu2c4
Gl.iNet AR300M16	Atheros AR7xxx	generic	GL AR300M	ext-router16
Gl.iNet AR750	Atheros AR7xxx	generic	GL AR750	ext-router16
TP-LINK Archer C7 v2	Atheros AR7xxx	generic	TP-LINK Archer C7 v2	ext-router16
TP-LINK Archer C7 v3	Atheros AR7xxx	generic	TP-LINK Archer C7 v3	ext-router16
TP-LINK Archer C7 v4	Atheros AR7xxx	generic	TP-LINK Archer C7 v4	ext-router16
Gl.iNet B1300	Qualcomm Atheros IPQ40xx		Gl.inet B1300	ext-router16 b1300
Linksys EA3500	Marvell Kirkwood		Linksys EA3500	ext-router16
Linksys E4200v2	Marvell Kirkwood		Linksys E4200v2/EA45 00 (Viper)	ext-router16
Linksys EA4500	Marvell Kirkwood		Linksys E4200v2/EA45 00 (Viper)	ext-router16
BT Home Hub 5A	Lantiq	XRX200	BT Home Hub 5A	ext-router16

Mikrotik NAND Large	Atheros AR7xxx	Mikrotik devices with NAND/NOR flash	Mikrotik Routerboard with >=128 MB NAND flash	ext-rooter16 ext-wifi
Alfa R36a	Atheros AR7xxx	generic	Alpha R36a	ext-rooter16
Gl.iNet 6416	Atheros AR7xxx	generic	Gl.iNet 6416	ext-rooter16
Gl.iNet AR150	Atheros AR7xxx	generic	Gl.iNet AR150	ext-rooter16
Gl.iNet Mifi	Atheros AR7xxx	generic	Gl.iNet Mifi	ext-rooter16
TP-Link WR842NDv3	Atheros AR7xxx	generic	TP-Link WR842NDv3	ext-rooter16
TP-Link WR1043NDv4	Atheros AR7xxx	generic	TP-Link WR1043NDv4	ext-rooter16
TP-Link WR1043NDv5	Atheros AR7xxx	generic	TP-Link WR1043NDv5	ext-rooter16
Dlink DIR825c1	Atheros AR7xxx	generic	Dlink DIR825c1	ext-rooter16
Dlink DIR835a1	Atheros AR7xxx	generic	Dlink DIR835a1	ext-rooter16
Netgear WNDR3700v2	Atheros AR7xxx	generic	Netgear WNDR3700v2	ext-rooter16
Netgear WNDR3800	Atheros AR7xxx	generic	Netgear WNDR3800	ext-rooter16
Netgear WNDRMAC	Atheros AR7xxx	generic	Netgear WNDRMAC	ext-rooter16
Netgear WNDRMACv2	Atheros AR7xxx	generic	Netgear WNDRMACv2	ext-rooter16
WD MyNet N600	Atheros AR7xxx	generic	WD MyNet N600	ext-rooter16

WD MyNet N750	Atheros AR7xxx	generic	WD MyNet N750	ext-router16
Buffalo WZR-HP-G300NH	Atheros AR7xxx	generic	Buffalo WZR-HP-G300NH	ext-router16
Buffalo WZR-HP-AG300H	Atheros AR7xxx	generic	Buffalo WZR-HP-AG300H	ext-router16
Dlink DGL-5500	Atheros AR7xxx	generic	Dlink DGL-5500	ext-router16
Turris Omnia	Marvell EBU Armada	Marvell Armada 37x/38x/XP	Turris Omnia	ext-router16
Orange Pi Zero Plus	Allwinner A1x/A20/A3x	Allwinner A64	Xunlong Orange Pi Zero Plus	ext-router16 ext-wifi fscheck
Mikrotik RBM33G	Mediatek Ralink MIPS	MT7621 based boards	Mikrotik RBM33G	ext-router16 rbm33g
Mikrotik RBM11G	Mediatek Ralink MIPS	MT7621 based boards	Mikrotik RBM11G	ext-router16 rbm11g
Raspberry Pi	Broadcom BCM27xx	BCM2708 based boards	Raspberry Pi B/B+/CM/Zero/ZeroW	ext-router16 ext-wifi fscheck
Raspberry Pi 2	Broadcom BCM27xx	BCM2709 based boards	Raspberry Pi 2 B	ext-router16 ext-wifi fscheck
Raspberry Pi 3	Broadcom BCM27xx	BCM2710 based boards	Raspberry Pi 3 B/CM	ext-router16 ext-wifi fscheck brcmfmac-firmware-43455-sdio-rpi
UniElec U7621-06	Mediatek Ralink MIPS	MT7621 based boards	UniElec U7621-06	ext-router16 ext-wifi
UniElec U7628-01	Mediatek Ralink MIPS	MT7628 based boards	UniElec U7628-01	ext-router16 ext-wifi

ZBT WG3526	Mediatek Ralink MIPS	MT7621 based boards	ZBT WG3526	ext-router16
ZBT WE3526	Mediatek Ralink MIPS	MT7620 based boards	ZBT WE826	ext-router16
ZBT WE1026- 5G	Mediatek Ralink MIPS	MT7620 based boards	ZBT WE1026- 5G	ext-router16
ZBT WE1326	Mediatek Ralink MIPS	MT7621 based boards	ZBT WE1326	ext-router16
generic x86 64 bit	x86	x86_64	Generic	ext-router16 ext-wifi
Xiaomi Mini	Mediatek Ralink MIPS	MT7620 based boards	Xiaomi MiWifi Mini	ext-router16
Xiaomi MiWifi Router 3G	Mediatek Ralink MIPS	MT7621 based boards	Xiaomi Mi Router 3G	ext-router16
Lenovo Y1	Mediatek Ralink MIPS	MT7620 based boards	Lenovo Y1	ext-router16
Lenovo Y1S	Mediatek Ralink MIPS	MT7620 based boards	Lenovo Y1S	ext-router16
Huawei HG553	Broadcom BCM63xx	generic	Huawei HG553	ext- routerbcm16
Netgear R8000	Broadcom BCM47xx/BC M53xx (ARM)		Netgear R8000	ext- routerbcm16
Asus RT- AC56U	Broadcom BCM47xx/BC M53xx (ARM)		Asus RT- AC56U	ext- routerbcm16
Asus RT- AC68U	Broadcom BCM47xx/BC M53xx (ARM)		Asus RT- AC68U	ext- routerbcm16
Asus RT- AC87U	Broadcom BCM47xx/BC M53xx (ARM)		Asus RT- AC87U	ext- routerbcm16

