Adding New Routers to the Automated Build System

There may come a time when you wish to make a new firmware for a router that has OpenWrt support but is not yet included in the Automated Build System. This document will explain how to go about doing this.

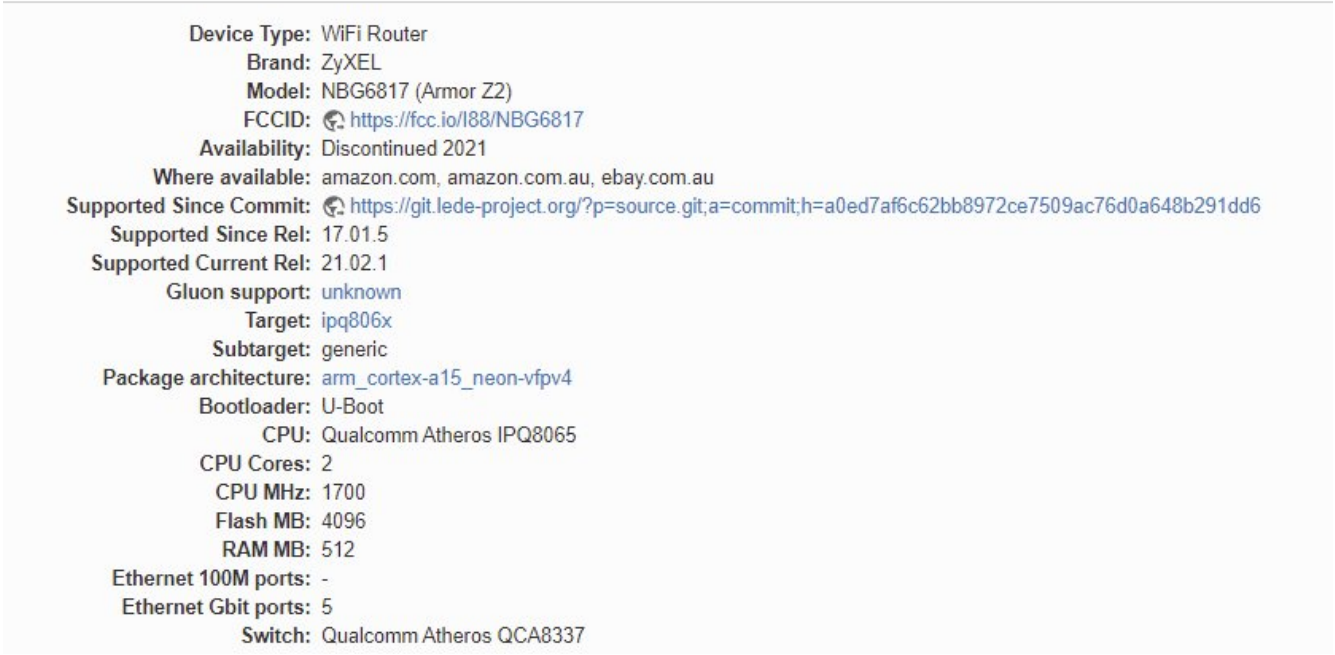
The Automated Build System consists of a configuration file for the router and an entry in a json file that defines all the information needed to build a firmware for the router.

You need to know information about the router and the place to get that is from

https://openwrt.org/toh/start

This web site contains information on all OpenWrt supported routers. In the far right colum is a link labeled *View/Edit data*. Go to that page for the router you are interested in.

Here you will find information on flash and RAM memory, processor type, wifi chipsets and the name of the image file(s) produced by the build system. This is all important information that is needed to add router support to the Automated Build System.



Device Type:	WiFi Router
Brand:	ZyXEL
Model:	NBG6817 (Armor Z2)
FCCID:	https://fcc.io/I88/NBG6817
Availability:	Discontinued 2021
Where available:	amazon.com, amazon.com.au, ebay.com.au
Supported Since Commit:	https://git.lede-project.org/?p=source.git;a=commit;h=a0ed7af6c62bb8972ce7509ac76d0a648b291dd6
Supported Since Rel:	17.01.5
Supported Current Rel:	21.02.1
Gluon support:	unknown
Target:	ipq806x
Subtarget:	generic
Package architecture:	arm_cortex-a15_neon-vfpv4
Bootloader:	U-Boot
CPU:	Qualcomm Atheros IPQ8065
CPU Cores:	2
CPU MHz:	1700
Flash MB:	4096
RAM MB:	512
Ethernet 100M ports:	-
Ethernet Gbit ports:	5
Switch:	Qualcomm Atheros QCA8337

Creating the Configuration File

When creating a new configuration file for a router you must first delete the existing configuration file from the build system. To do this look in the main build system folder for a file named *.config* (note the leading period in the name). Delete this file. This will reset the new configuration file to default.

Open the Terminal program and enter this at the command line.

make menuconfig

This will open a router configuration program that allows you to create a new configuration file for a router. We will now take the information we found on the *View/Edit data* page for the router and create a template configuration for it.

There are just a few things that need setting to make the template configuration file.

- Target System - this is the *Target* entry on the page. In this case it is *IPQ806x*. Enter this submenu and search for the *Target* there. The *Target System* may not be named exactly the name of *Target* as there can be a manufacturers name as a prefix. In this case it is called *Qualcomm IPQ806x*. Scroll through the list until you find the correct Target.
- Subtarget - this is the *Subtarget* on the page. Select the one from the submenu that matches this.
- Target Profile - This is the make and model of the router and this can be found on the web page as *Brand* and *Model*. In this example the Brand is *ZyXEL* and the Model is *NBG6817*. Search in the submenu for the Brand and Model, *ZyXEL NBG6817 and select that.*
- Luci - as a precaution against creating an image without the web GUI you should also select the Luci package by going to *Luci→Collections* and selecting *Luci*. This will include the GUI in any image that you make using the template configuration.

Now exit the configuration program and save the file when asked. The last step is to rename the configuration file and place it in the correct folder.

The configuration file you have just created is named *.config* (note the leading period in the name). This needs to be renamed to a unique name for this router. By convention it is renamed to *.config_xxx* where *xxx* is the name assigned to the router. For this example it will be named *.config_z6817*. Again note the leading period in the name.

Once the file is renamed it can be moved into the */configfiles/template* folder. This completes the creation of a template confguration file for the new router. Using this configuration file in the build system will create a basic OpenWrt image with the web GUI present. The Automated Build System will use this template to create proper ROOter images.

Adding the Entry to the Json File

With the configuration file created and stored in the correct location we can move on to telling the Automated Build System how to go about creating a ROOter image for this router.The first step in this is to again look at the *View/Edit data* page for the router.

This will tell us about the image file(s) name and it's location in the build system.

With this information in hand we must to the json flie used to define the router. This file is found in the main directory of the build system and will be named *routerxxx.json* where the *xxx* varies acording to the version of the build system.

- routers.json - 19.07.6 build system
- router-1806.json - 18.06.7 build system
- router-2102.json - 21.02 build system
- router-GL.json - Gl.inet build system

These files are a standard json file that uses brackets to define a router entry. As an example this is the entry for the ZyXEL router we have been looking at.

```
"Z6817": {
        "type": "2",
        "config": ".config_z6817",
        "imagepath": "/ipq806x/generic/",
        "mod": "ZyXEL-NBG6817",
        "image1": "openwrt-ipq806x-generic-zyxel_nbg6817-squashfs-
```

sysupgrade.bin",
"image2": "openwrt-ipq806x-generic-zyxel_nbg6817-squashfs-factory.bin",
"ext1": "-upgrade.bin",
"ext2": "-factory.bin"
}

This is an entry that has been stripped of all unneeded information.

The first line defines the name of the router as used by the Automated Build System. It is always in upper case letters but when making an image it can be used in lower case. For example, *./build z6817* is used to build the image.

There are 4 different types of routers that can have images made for them. This are :

- Type 1 – only an upgrade image is produced.
- Type 2 – a factory and an upgrade image are produced.
- Type 3 – Raspberry Pi routers.
- Type 4 – a single image is produced that is inside a *img.gz* file.

The *type* line defines which type of router we are working with, in this case *Type 2*.

The *config* line defines the name of the configuration file we created for this router and placed in */configfiles/template*.

The *imagepath* line tells us where the newly created image file is located. This is in the */bin/targets* folder and the path must start and end with a backslash (/). This particular image is found in the */bin/targets/ipq806x/generic/* folder.

The *mod* line is the name that is used on the image file and the Zip file containing the images. This can be anything that describes the router.

The next lines are used to describe the actual name of the image file(s) produced by the build system. These lines will vary depending on the router type.

For *Type 1* there will be an *image* line.

"image": "openwrt-ramips-mt7620-alfa-network_tube-e4g-squashfs-sysupgrade.bin",

This is the image that is used to upgrade the router.

For *Type 2* there will be an *image1* and an *image2* line.

"image1": "openwrt-ipq806x-generic-zyxel_nbg6817-squashfs-sysupgrade.bin",
"image2": "openwrt-ipq806x-generic-zyxel_nbg6817-squashfs-factory.bin",

These are the image used to upgrade to ROOter from the factory firmware and the image used to upgrade from ROOter to ROOter.

For *Type 4* there will also be an *image1* and *image2* line.

"image1": "openwrt-x86-generic-combined-squashfs.img.gz",
"image2": "openwrt-x86-generic-combined-squashfs.img",

The *image1* line is the image file produced by the build system and the *image2* line is the name of the file inside the *image1* file.

Finally there are a line or lines describing the extension used when renaming the image files produced by the build system into ROOter format.

For a *Type 1* router there is a single extension line.

"ext": "-upgrade.bin"

The image that ROOter produces will consist of the *mod* (Alfa-Tube-E4G), the date with a GO prefix (GO2021-12-30) and the ext (-upgrade.bin) for a final name of *Alfa-Tube-E4G-GO2021-12-30-upgrade.bin.* This is what the *image* file will be renamed to.

For *Type 2* there will be an *ext1* and *ext2* extension line.

"ext1": "-upgrade.bin",
"ext2": "-factory.bin"

When renaming the image file *ext1* is used with *image1* and *ext2* is used with *image2*. The *image1* file will be renamed to *mod*, date and *ext1* like *ZyXEL-NBG6817-GO2021-12-30-upgrade.bin*. The *image2* file will be renamed using the *ext2* line like *ZyXEL-NBG6817-GO2021-12-30-factory.bin*.

Type 4 routers have no extension line as the extension is fixed.

The name of the *image*, *image1* and *image2* files can be found from the *View/Edit data* web page for the router.

Firmware OpenWrt Install URL :
https://downloads.openwrt.org/releases/21.02.1/targets/ipq806x/generic/openwrt-21.02.1-ipq806x-generic-zyxel_nbg6817-squashfs-factory.bin

This tells us that the factory to ROOter image (*image2*) is named

openwrt-21.02.1-ipq806x-generic-zyxel_nbg6817-squashfs-factory.bin

If there are OpenWrt versions in the name they must be removed. This will leave you with an image name of

openwrt-ipq806x-generic-zyxel_nbg6817-squashfs-factory.bin

In some cases you will find that this name is incorrect due to changes made to the image name between OpenWrt versions. This will appear when you try to use the Automated Build System and you get an error at the end stating it can't find the image. You then need to go to */bin/targets/xxxx/yyyy* and find the correct name.
Once you have completed the entries in the json file you are ready to build your new image.

Please note that the format of the entry in the json file is very rigid and it must be followed exactly. All data in the lines must be surrounded by double quotes (") and every line except the last must have a comma (,) at the end.

Use another router entry as a template when addng a new router to avoid making syntax errors. Also note that each router entry is separated by a comma (,) except the final one in the file.

For reference the json file is laid out as follows :

```
{
	"ROUTER1": {
		"type": "1",
		"config": "config file",
		"mod": "router name",
		"imagepath": "path to image file",
		"image": "name of upgrade image file",
		"ext": "extension name"
	},
	"ROUTER2": {
		"type": "2",
		"config": "config file",
		"mod": "router name",
		"imagepath": "path to image files",
		"image1": "name of factory image file",
		"image2": "name of upgrade image file",
		"ext1": "extension name",
		"ext2": "extension name"
	},
	"ROUTER3": {
		"type": "4",
		"config": "config file",
		"mod": "router name",
		"imagepath": "path to image file",
		"image1": "name of GZ image file",
		"image2": "name of IMG image file"
	}
}
```