# C/C++ Compile Guide

**(Version 1.0.0)**





For more information, please visit our website at http://www.wiznet.io/

# Document Revision History

| Date | Revision | Changes |
|---|---|---|
| 2019-11-25 | 1.0 | Release |
| | | |
| | | |
| | | |

# Contents

# 1. Overview

This document will guide users how to cross compile the C/C++ program for OpenWRT on WizFi630S.

# 2. Download

## 2.1 Prerequisites

The build system operates in Linux, BSD, MacOSX OS, and a file system that differentiates uppercase and lowercase letters is required. Windows is not supported because it cannot differentiate uppercase and lowercase letters of Cygwin.
Disk space of 10~15GB and 2GB of RAM is needed for the default firmware package.

All commands will be processed as user without root permission.



## 2.2 Packages for Building Environment

The below packages need to be installed for each OS.

- **Arch Linux**

```
pacman -S --needed asciidoc bash bc binutils bzip2 fastjar flex git gcc util-linux gawk
intltool zlib make cdrkit ncurses openssl patch perl-extutils-makemaker rsync unzip wget
gettext libxslt boost libusb bin86 sharutils b43-fwcutter findutils time
```

- **Alpine Linux**

```
apk add asciidoc bash bc binutils bzip2 cdrkit coreutils diffutils findutils flex g++ gawk
gcc gettext git grep intltool libxslt linux-headers make ncurses-dev patch perl python2-dev
tar unzip  util-linux wget zlib-dev
```

- **Debian 7 Wheezy**

```
apt-get install libncurses5-dev zlib1g-dev gawk
```

- **Debian 8 Jessie**

```
sudo apt-get install build-essential libncurses5-dev gawk git libssl-dev gettext unzip
zlib1g-dev file python
```

- **Debian 9.4 Stretch**

```
sudo apt install build-essential libncurses5-dev gawk git libssl-dev gettext zlib1g-dev swig
unzip time
```

- **Debian 10**

```
sudo apt install build-essential libncurses5-dev gawk git libssl-dev gettext zlib1g-dev swig
unzip time
```

- **Fedora 24**

```
dnf install binutils bzip2 gcc gcc-c++ gawk gettext git-core flex ncurses-devel ncurses-
compat-libs zlib-devel zlib-static make patch unzip perl-ExtUtils-MakeMaker perl-Thread-
Queue glibc glibc-devel glibc-static quilt sed sdcc intltool sharutils bison wget openssl-
devel
```

- **Fedora 29**

```
dnf install @c-development @development-tools @development-libs zlib-static wget python2
```

- **openSUSE 13.2**

```
zypper install asciidoc bash bc binutils bzip2 fastjar flex git-core gcc-c++ gcc util-linux
gawk intltool zlib-devel mercurial make genisoimage ncurses-devel libopenssl-devel patch
perl-ExtUtils-MakeMaker python-devel rsync sdcc unzip wget gettext-tools libxslt-tools zlib-
devel
```

- **openSUSE 42.3**

```
zypper install patterns-openSUSE-devel_basis zlib-devel-static git-core
```

- **openSUSE 15**

```
zypper install patterns-devel-base-devel_basis  zlib-devel-static git-core
```

- **Ubuntu 18.04 LTS**

```
sudo apt-get install subversion build-essential libncurses5-dev zlib1g-dev gawk git ccache
gettext libssl-dev xsltproc zip python3-distutils
```

- **Centos x86-64 (some packages require EPEL)**

```
yum install binutils bzip2 gcc gcc-c++ gawk gettext flex ncurses-devel zlib-devel zlib-
static make patch unzip perl-ExtUtils-MakeMaker glibc glibc-devel glibc-static ncurses-libs
sed sdcc intltool sharutils bison wget git-core openssl-devel xz python3-distutils
```



## 2.3 OpenWRT Firmware Repository

The OpenWRT firmware has two branches, 'stable release' and 'development.'

The source codes of OpenWRT can be downloaded at the OpenWRT Git repository.

```
git clone https://github.com/openwrt/openwrt.git
```

Users can switch via 'git checkout' command If a user needs a particular version of branch.

For LEDE 17.01:

```
git checkout lede-17.01
```

For OpenWrt 18.06:

```
git checkout openwrt-18.06
```

## 2.4 Menuconfig

Users must go through the 'build' process after 'menuconfig' in order to set and download the Cross Compiler of OpenWRT – this process can take up to hours depending on the environment.

```
make menuconfig
```

Complete settings at the 'menuconfig' screen as shown below.

Then save and exit.

- Target System: MediaTek Ralink MIPS
- Subtarget: MT76x8 based boards
- Target Profile: WIZnet WizFi630S

Download and compile the files related to the 'make' command.

```
make V=s -j5
```

# 3. Write C Code

## 3.1 Helloworld

Write 'helloworld' program at workspace.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ cat helloworld.c
#include <stdio.h>


int main()
{
  printf("Hello World\r\n");
  return 0;
}
```

## 3.2 Make the Environment Script

Save the Cross Compile Toolchain related path as script.

OPENWRT_ROOT will enter the location where OpenWRT SDK is downloaded as shown below.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ cat config
# set up paths and environemt for cross compiling for openwrt
export OPENWRT_ROOT=/home/daniel/workspace/WizFi630S/openWRT/openwrt1806
export STAGING_DIR=$OPENWRT_ROOT/staging_dir
export TOOLCHAIN_DIR=$STAGING_DIR/toolchain-mipsel_24kc_gcc-7.4.0_musl
export LDCFLAGS=$TOOLCHAIN_DIR/usr/lib
export LD_LIBRARY_PATH=$TOOLCHAIN_DIR/usr/lib
export PATH=$TOOLCHAIN_DIR/bin:$PATH
```

# 4. Cross Compile

## 4.1 Run Environment Script

Run the 'config' that was created as script above before running Cross Compile.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ source config
```

## 4.2 Cross Compile

Cross compile 'helloworld.c' using the below command.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ mips-openwrt-linux-gcc -o
helloworld helloworld.c
```

Once the compile is completed, a output binary file named 'helloworld' will be created.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ ls
config  helloworld  helloworld.c
```

# 5. Run Helloworld

## 5.1 Prerequisites

The PC must be connected to the network in order to run the output binary on WizFi630S.

Transfer the binary that was compiled via SCP.

The default IP of WizFi630S is 192.168.1.1.

## 5.2 Copy the Binary to the WizFi630S

Transfer the binary file (helloworld) using SCP command.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ scp helloworld
root@192.168.1.1:hello
helloworld                                    100% 8324   141.7KB/s   00:00
```

## 5.3 Connect to WizFi630S

Connect to WizFi630S via SSH command.

Users can find the file that was copied via SCP command from the file list.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ ssh root@192.168.1.1


BusyBox v1.30.1 () built-in shell (ash)

  _______                     ________        __
 |       |.-----.-----.-----.|  |  |  |.----.|  |_
 |   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
 |_______||   __|_____|__|__||________||__|  |____|
          |__| W I R E L E S S   F R E E D O M
 -----------------------------------------------------
 OpenWrt SNAPSHOT, r9850-2101002b3d
 -----------------------------------------------------
=== WARNING! =====================================
```

```
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
--------------------------------------------------
root@wizfi630s:~# ls
hello
root@wizfi630s:~#
```

## 5.4 Run Helloworld

Run the 'Hello' file.

```
root@wizfi630s:~# ./hello
Hello World
root@wizfi630s:~#
```