

C/C++ Compile Guide

(Version 0.0.1)

WIZnet <https://wiznet.io>
WIZnet <https://wizwiki.net>



© 2019 WIZnet Co., Ltd. All Rights Reserved.

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

Document Revision History

Date	Revision	Changes
2019-09-09	1.0	Release

 <https://wiznet.io>
<https://wizwiki.net>

Contents

1. Overview	4
2. Download	4
2.1 Prerequisites	4
2.2 Packages for build environment.....	4
2.3 OpenWRT Firmware repository	6
2.4 Menuconfig	7
3. Write C Code.....	7
3.1 Helloworld	7
3.2 Make the Environment Script	8
4. Cross Compile	8
4.1 Run Environment Script.....	8
4.2 Cross Compile.....	8
5. Run Helloworld.....	9
5.1 Prerequisites	9
5.2 Copy the binary to the WizFi630S	9
5.3 Connect to the WizFi630S.....	9
5.4 Run Helloworld.....	10

1. Overview

이 문서는 WizFi630S에 OpenWRT용 C/C++ 프로그램을 Cross Compile하고 모듈에서 실행하는 방법에 대한 가이드입니다.

2. Download

2.1 Prerequisites

OpenWRT 빌드 시스템은 Linux, BSD, MacOSX 운영체제에서 동작하며, 대소문자를 구별하는 파일 시스템이 필요합니다. Windows는 Cygwin이 대소문자를 구분하지 않기 때문에 지원하지 않습니다. 기본 펌웨어 패키지를 생성하기 위해서는 10~15GB 이상의 디스크 여유공간과 2GB 이상의 RAM이 필요합니다.

모든 명령은 root 권한없이 user로 진행해야 합니다.

2.2 Packages for build environment

빌드를 위해 각 운영체제별로 아래와 같은 패키지들이 설치되어 있어야 합니다.

- **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

OpenWRT 펌웨어는 Stable Release와 Development라는 2개의 branch로 나뉘져 있습니다. OpenWRT의 소스코드는 OpenWRT Git repository에서 다운로드 가능합니다.

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

만약 특정 버전의 Branch를 원한다면, 'git checkout' 명령으로 전환 가능합니다.

For LEDE 17.01:

```
git checkout lede-17.01
```

For OpenWrt 18.06:

```
git checkout openwrt-18.06
```

2.4 Menuconfig

OpenWRT의 Cross Compiler를 설정하고 다운로드 하기 위해서는 menuconfig 후 build 과정을 거쳐야 합니다. 이 과정은 환경에 따라 수 시간이 소요될 수 있습니다.

```
make menuconfig
```

menuconfig 화면에서 아래와 같이 WizFi630S으로 설정을 완료한 이후에 Save 후 Exit 합니다.

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

이후 make 명령으로 관련파일들을 다운로드하고 컴파일 합니다.

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

3. Write C Code

3.1 Helloworld

Workspace에서 helloworld 프로그램을 작성합니다.

```
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

Cross Compile Toolchain 관련 path를 스크립트로 저장합니다.

아래 OPENWRT_ROOT가 OpenWRT SDK가 설치된 위치를 입력합니다.

```
daniel@daniel-ubuntu:~/workspace/WizFi630S/openWRT/hello$ cat config
# set up paths and environment 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

Cross Compile을 실행하기 전 위에서 스크립트로 만들어 놓은 config를 실행합니다.

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

4.2 Cross Compile

아래 명령으로 helloworld.c를 cross compile 합니다.

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

Compile이 정상적으로 완료되면, helloworld 라는 output binary 파일이 생성됩니다.

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


5. Run Helloworld

5.1 Prerequisites

Output binary를 WizFi630S에서 실행하기 위해서는 PC와 네트워크로 연결되어야 합니다. 이 과정에서는 SCP를 사용해서 컴파일 된 binary를 WizFi630S로 복사하도록 하겠습니다.

WizFi630S의 기본 IP는 192.168.1.1로 설정되어 있습니다.

5.2 Copy the binary to the WizFi630S

SCP 명령을 사용해 binary파일(helloworld)를 WizFi630S로 복사합니다.

```
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 the WizFi630S

SSH 명령을 사용해서 WizFi630S에 접속합니다.

접속한 이후 파일 리스트를 확인해보면 위 단계에서 SCP 명령으로 복사한 파일이 복사되어 있는 것을 확인할 수 있습니다.

```
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

Hello 파일을 실행합니다.

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