WM_W60X_SDK_GCC Compiling Guide

V1.2

Beijing Winner Microelectronics Co., Ltd.

Address: 18th Floor, Yindu Building, No.67 Fucheng Road, Haidian District, Beijing, China

Tel: +86-10-62161900

Website: www.winnermicro.com

Document History

Version	Completion Date	Revision Record	Author	Auditor
V0.1	2018/9/20	Initial Release	Leihy	
V1.0	2018/9/28	Add building cross-compiling environment under Linux	Huangll	
V1.1	2018/11/2	Update author information	Cuiych	
V1.2	2018/12/15	Modify W600 to W60X	Ray	

Content

1 Building cross-compiling environment

We suggest developer to do cross-compiling under Linux. No matter under Cygwin or MSYS environment, the compiling rate are both slower than Linux environment.

1.1 Linux Environment (binary package mode)

When Ubuntu system is installed by default mode, the data packages for system installation are different in each time. CentOS should be used under Linux environment. This document uses CentOS7.5.1804 and users can get the free image file from CentOS official website.

1.1.1 Setting under CentOS7

Pay attention to following items during installing this system:

◆ In part of【Select Software】, we suggest to choose the type of basic environment with【**Develop and Generate workstations**】, and tick all the attached options.

1.1.1.1 Preparation Work

The selected cross-compiling tools is 32bit ELF, but there are many 64bit equipment now. So some software packages should be used to support running 32bit program on 64bit system. Following is the detailed operation by root (pay attention to the account authority):

```
# yum install glibc.i686 -y
# mkdir /opt/toolchain/
# chown YOU-ACCOUNT.YOU-ACCOUNT /opt/toolchain/
```

YOU-ACCOUNT is the general account created by user, which is convenient to decompress by toolchain.

1.1.1.2 Get and Process the cross-compiling tools

Following command should be executed:

```
$ wget https://launchpad.net/gcc-arm-embedded/4.9/4.9-2014-q4-major/+download/gcc-arm-none-eabi-4_9-2014q4-20141203-linux.tar.bz2
$ tar -xjf gcc-arm-none-eabi-4_9-2014q4-20141203-linux.tar.bz2 -C /opt/toolchain/
$ ln -s gcc-arm-none-eabi-4_9-2014q4 /opt/toolchain/gcc-arm-none-eabi
```

1.1.1.3 Modify Variables of Environment

Some toolchain information should be added in PATH variables. Following is the detailed command (pay attention to the account authority):

```
# echo "export PATH=\$PATH:/opt/toolchain/gcc-arm-none-eabi/bin" >> /etc/bashrc
```

```
$ source ~/.bashrc
```

1.1.2 Environment Verification

a) Check arm-gcc absolute path

```
$ which arm-none-eabi-gcc
/opt/toolchain/gcc-arm-none-eabi/bin/arm-none-eabi-gcc
$
```

b) Verify arm-gcc version

Using following command to verify arm-gcc version number 4.9.3:

```
$  arm-none-eabi-gcc  -v
Using  built-in  specs.
COLLECT_GCC=arm-none-eabi-gcc
COLLECT_LTO_WRAPPER=/opt/toolchain/gcc-arm-none-eabi-4_9-2014q4/bin/../lib/gcc/arm-none-eabi/4.9.3/lto-wrapper
Target:  arm-none-eabi
Configured  with:  /home/build/work/GCC-4-9-build/src/gcc/configure  --target=arm-none-eabi  --prefix=/home/build/work/GCC-4-9-build/install-native  --libexecdir=/home/build/work/GCC-4-9-build/install-native/lib  --infodir=/home/build/work/GCC-4-9-build/install-native/share/doc/gcc-arm-none-eabi/info  --mandir=/home/build/work/GCC-4-9-build/install-native/share/doc/gcc-arm-none-eabi/man  --htmldir=/home/build/work/GCC-4-9-build/install-native/share/doc/gcc-arm-none-eabi/html  --pdfdir=/home/build/work/GCC-4-9-build/install-native/share/doc/gcc-arm-none-eabi/pdf  --enable-languages=c,c++  --enable-plugins  --disable-decimal-float  --disable-libffi  --disable-libgomp  --disable-libmudflap  --disable-libquadmath  --disable-libssp  --disable-libstdcxx-pch  --disable-nls  --disable-shared  --disable-threads  --disable-tls  --with-gnu-as  --with-gnu-ld  --with-newlib  --with-headers=yes  --with-python-dir=share/gcc-arm-none-eabi  --with-sysroot=/home/build/work/GCC-4-9-build/install-native/arm-none-eabi  --build=i686-linux-gnu  --host=i686-linux-gnu  --with-gmp=/home/build/work/GCC-4-9-build/build-native/host-libs/usr  --with-mpfr=/home/build/work/GCC-4-9-build/build-native/host-libs/usr  --with-mpc=/home/build/work/GCC-4-9-build/build-native/host-libs/usr  --with-isl=/home/build/work/GCC-4-9-build/build-native/host-libs/usr  --with-cloog=/home/build/work/GCC-4-9-build/build-native/host-libs/usr  --with-libelf=/home/build/work/GCC-4-9-build/build-native/host-libs/usr  --with-host-libstdcxx='-static-libgcc  -Wl,-Bstatic,-lstdc++,-Bdynamic  -lm'  --with-pkgversion='GNU  Tools  for  ARM  Embedded  Processors'  --with-multilib-list=armv6-m,armv7-m,armv7e-m,armv7-r
Thread  model:  single
```

```
gcc version 4.9.3 20141119 (release) [ARM/embedded-4_9-branch revision 218278] (GNU Tools for ARM Embedded Processors)
$
```

2 Using xxx-gcc to Compile Project

2.1.1 Default Conditions

Default to compile under Linux environment. The W60X SDK’s LWIP version is 2.0.3.

2.1.2 Makefile Functions

Besides basic functions of default target and clean target, the Makefile in current project also supports following 2 functions:

- No parameter: Silent Compilation
- V=s: output all information

2.1.3 Compilation

2.1.3.1 Compilation under Linux

```
$ cd Tools/GNU/
$ make
```

or

```
$ cd Tools/GNU/
$ make V=s
```

After compilation succeed, the target file will be saved in Bin directory in the project’s root directory.

If secondary developing users want to add files, they should save such files in app directory, and modify Makfile in Tools/GNU at the same time.

3 Appendix

3.1 Appendix A: Samba Server Installation and Application

3.1.1 Install Samba server under CentOS 7

3.1.1.1 Samba Configuration

Modify the content in /etc/samba/smb.conf based on following command:

```
# diff -Nur /etc/samba/smb.conf.bak /etc/samba/smb.conf
--- /etc/samba/smb.conf.bak         2018-03-05 16:32:05.306000047 +0800
+++ /etc/samba/smb.conf 2018-03-05 16:33:06.809001634 +0800
@@ -246,9 +246,11 @@
 #============================ Share Definitions
==============================


 [homes]
-            comment = Home Directories
-            browseable = no
+             comment = %S Home Directories
+             browseable = yes
             writable = yes
+             create mask = 0777
+             directory mask = 0777
 ;          valid users = %S
 ;          valid users = MYDOMAIN\%S

#
```

Add Samba account and password, using following command:

```
# smbpasswd -a ssss
New SMB password:
Retype new SMB password:
#
```

For the security purpose, the password can not be echoed by Linux. We suggest using the same password with ssss account login password.

3.1.1.2 Firewall and SELinux

Modify the content in /etc/selinux/config base on following command:

```
# diff -Nur /etc/selinux/config.bk /etc/selinux/config
--- /etc/selinux/config.bk          2018-09-28 07:40:20.992648619 -0400
+++ /etc/selinux/config 2018-09-28 07:40:30.503650417 -0400
```

```
@@ -4,7 +4,7 @@
 #       enforcing - SELinux security policy is enforced.
 #       permissive - SELinux prints warnings instead of enforcing.
 #       disabled - No SELinux policy is loaded.
-SELINUX=enforcing
+SELINUX=disabled
 # SELINUXTYPE= can take one of three two values:
 #       targeted - Targeted processes are protected,
 #       minimum - Modification of targeted policy. Only selected processes are protected.
#
```

It will not take effect immediately after modifying above files. It will take effect in the next time starting up. To modify the current running status of SELinux, following command can be used:

```
# setenforce 0
```

According to firewall operation, Samba open interface should be added during production process. Following command is how to close firewall:

```
# systemctl stop firewalld
# systemctl disable firewalld
```

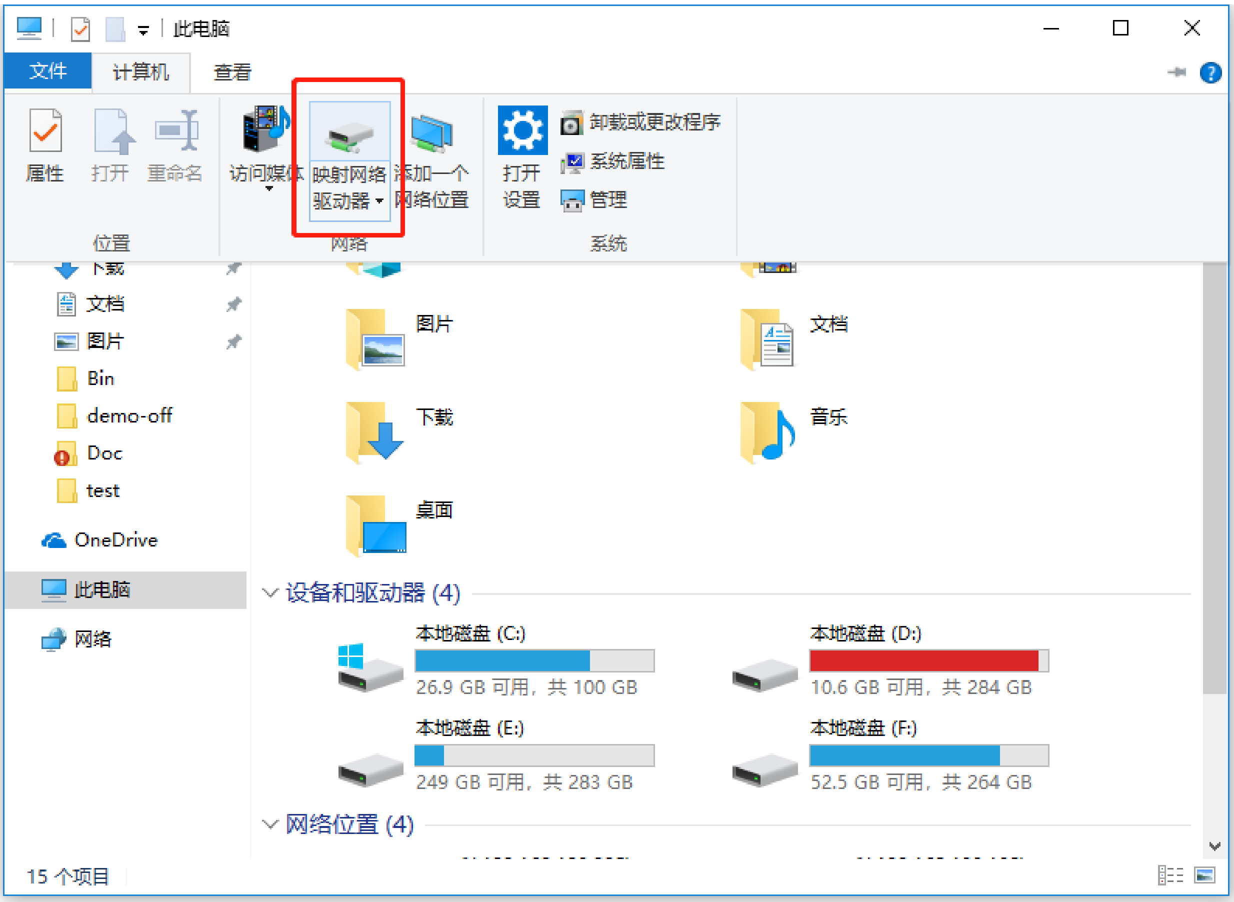
3.1.1.3 Startup Samba Service

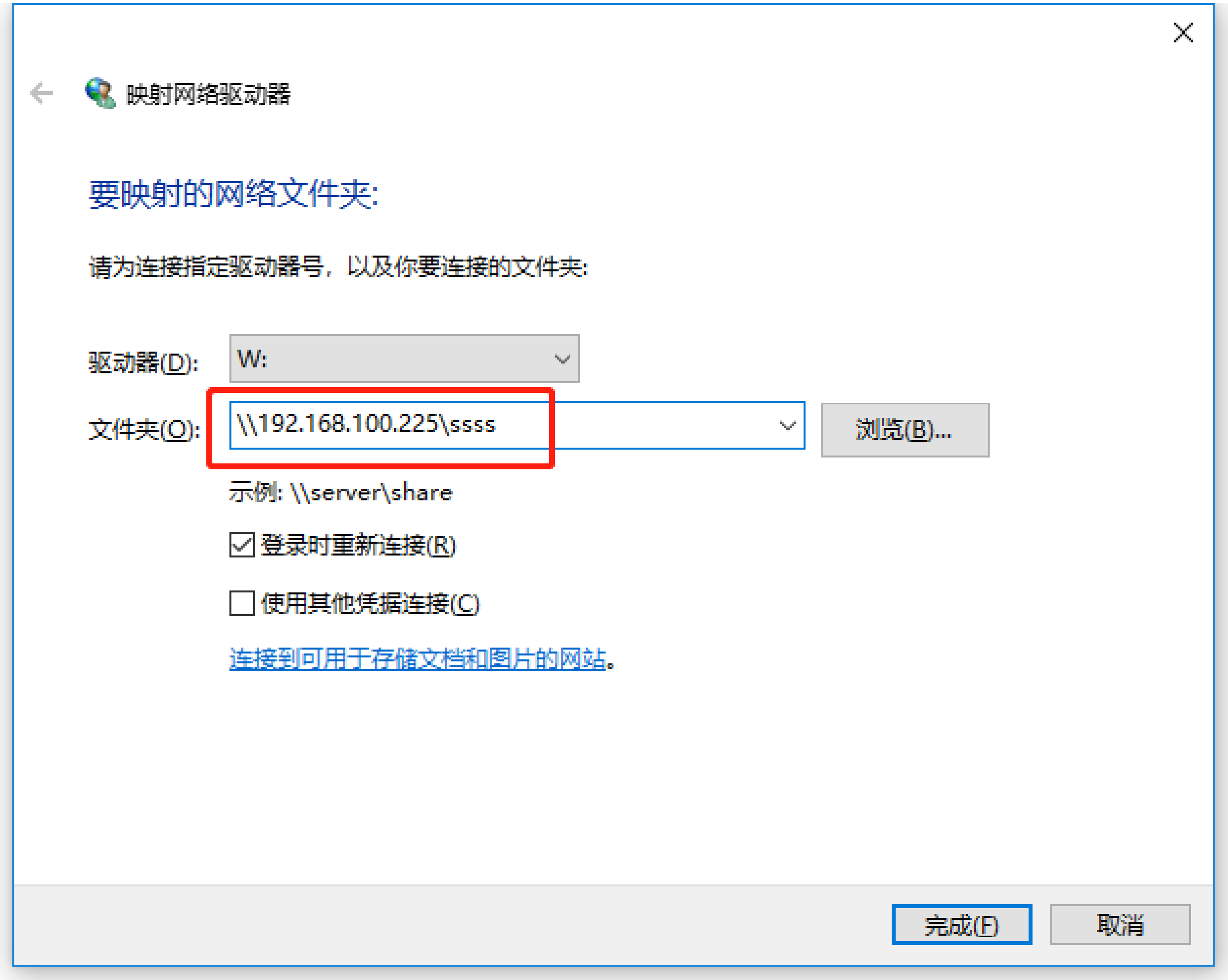
Following 4 command can help to startup Samba:

```
# systemctl enable smb
# systemctl enable nmb
# systemctl restart nmb.service
# systemctl restart smb.service
```

3.1.2 Visit Linux Samba server under Windows:

Following steps are the process to visit Linux Samba server under Win10 (visit with Samba account ssss):


此电脑
文件
计算机
查看
属性
打开
重命名
位置
访问媒体
映射网络驱动器
添加一个网络位置
网络
打开设置
卸载或更改程序
系统属性
管理
系统
下载
文档
图片
Bin
demo-off
Doc
test
OneDrive
此电脑
网络
图片
文档
下载
音乐
桌面
设备和驱动器 (4)
本地磁盘 (C:)
26.9 GB 可用，共 100 GB
本地磁盘 (D:)
10.6 GB 可用，共 284 GB
本地磁盘 (E:)
249 GB 可用，共 283 GB
本地磁盘 (F:)
52.5 GB 可用，共 264 GB
网络位置 (4)
15 个项目



映射网络驱动器

要映射的网络文件夹:

请为连接指定驱动器号，以及你要连接的文件夹:

驱动器(D): W:

文件夹(O): \\192.168.100.225\ssss

示例: \\server\share

☑ 登录时重新连接(R)

□ 使用其他凭据连接(C)

连接到可用于存储文档和图片的网站。

Then input ssss's password to create network keyboard for visiting Samba server.