

WM_W60X_SDK_GCC 编译指南

V1.2

北京联盛德微电子有限责任公司 (winner micro)

地址：北京市海淀区阜成路 67 号银都大厦 18 层

电话：+86-10-62161900

公司网址：www.winnermicro.com

文档历史

版本	完成日期	修订记录	作者	审核
V0.1	2018/9/20	创建	Leihy	
V1.0	2018/9/28	添加 linux 下交叉编译环境的搭建	Huangll	
V1.1	2018/11/2	更新作者信息	Cuiych	
V1.2	2018/12/13	文件更名为 W60X	Cuiych	

目录

1	交叉编译环境搭建	2
1.1	Linux 篇（二进制包方式）	2
1.1.1	在 CentOS 7 系统上的环境处理	2
1.1.2	环境验证	3
2	使用 xxx-gcc 编译项目	5
2.1.1	默认条件	5
2.1.2	Makefile 支持的功能	5
2.1.3	编译	5
3	附录	6
3.1	附录 A Samba 服务器安装和应用	6
3.1.1	CentOS 7 下 Samba 服务器安装	6
3.1.2	Windows 下访问 Linux 的 Samba 服务器	7

1 交叉编译环境搭建

在第一次编写该章节的时候,不论是 Cygwin 还是 MSYS,编译的速度均大幅慢于 Linux 环境,所以,在此建议选择 Linux 环境进行交叉编译环境的部署。

1.1 Linux 篇(二进制包方式)

由于 Ubuntu 系统在安装时,默认情况下,每一次系统安装的数据包不太确定,为了达到可控、明确的要求, Linux 系统环境选择 CentOS,本文档使用的具体的版本为 CentOS 7.5.1804,该系统镜像可从 CentOS 官网免费获得。

1.1.1 在 CentOS 7 系统上的环境处理

在安装该系统的时候,以下一点需要注意:

- ◆ 在【软件选择】部分,建议选择【**开发及生成工作站**】类型的基本环境,并且**勾选所有**该基本环境的附加选项。

1.1.1.1 准备工作

由于选择的交叉编译工具链为 32 位 ELF,而目前大多数设备的字长为 64 位,故需要增加一些软件包用于支持在 64 位系统上运行 32 位程序。具体在 root 用户下做的所有操作如下(注意账户权限):

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

其中, YOU-ACCOUNT 为您创建的普通账户,方便解压 toolchain 使用。

1.1.1.2 交叉编译工具获得与处理

该部分需要执行的命令如下:

```
$ 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 修改环境变量

为了方便使用,需要在 PATH 变量中增加 toolchain 的相关信息,具体命令如下(注意账户权限):

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

1.1.2 环境验证

a) 查看 arm-gcc 的绝对路径

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

b) 验证 arm-gcc 版本

使用如下的命令验证 arm-gcc 的版本，由此可以知道，当前 arm-gcc 的版本为 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-libstdc++-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-libstdc++='-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)

\$

北京联盛德微电子有限责任公司

2 使用 xxx-gcc 编译项目

2.1.1 默认条件

默认时，使用 Linux 环境进行编译，另外，lwip 代码使用的是 2.0.3 版本。

2.1.2 Makefile 支持的功能

除了基本的默认目标和 clean 目标之外，当前项目的 Makefile 还支持两个功能：

- ◆ 不带参数：静默编译
- ◆ V=s 参数：输出所有信息

2.1.3 编译

2.1.3.1 Linux 下的编译

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

```
$ make
```

或

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

```
$ make V=s
```

编译成功之后，目标文件存放在项目根目录中的 Bin 目录下。

二次开发客户，如果增加文件，把新增的文件放在 app 目录下，同时修改 Tools/GNU 下的 Makefile。

3 附录

3.1 附录 A Samba 服务器安装和应用

3.1.1 CentOS 7 下 Samba 服务器安装

3.1.1.1 samba 配置

根据以下内容修改/etc/samba/smb.conf 文件中的内容：

```
# 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

#
```

添加 samba 账户密码，使用的命令如下所示：

```
# smbpasswd -a ssss
```

New SMB password:

Retype new SMB password:

```
#
```

为了安全起见，输入的密码，linux 不会回显出来。在这里，建议此时设置的密码与 ssss 账户的登录密码一致。

3.1.1.2 防火墙及 SELinux

根据以下内容修改/etc/selinux/config 文件中的内容：

```
# 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.
#
```

上述文件修改之后，不会立即生效，在下次启动时会生效，修改当前运行时的 SELinux 状态，可以使用下面的命令：

setenforce 0

对于防火墙的操作，生产环境下，需要添加 samba 的开放端口，在这里，仅仅关闭防火墙，命令如下：

systemctl stop firewalld

systemctl disable firewalld

3.1.1.3 Samba 服务启动

完成 samba 的启动目标，需要借助以下四条命令实现：

systemctl enable smb

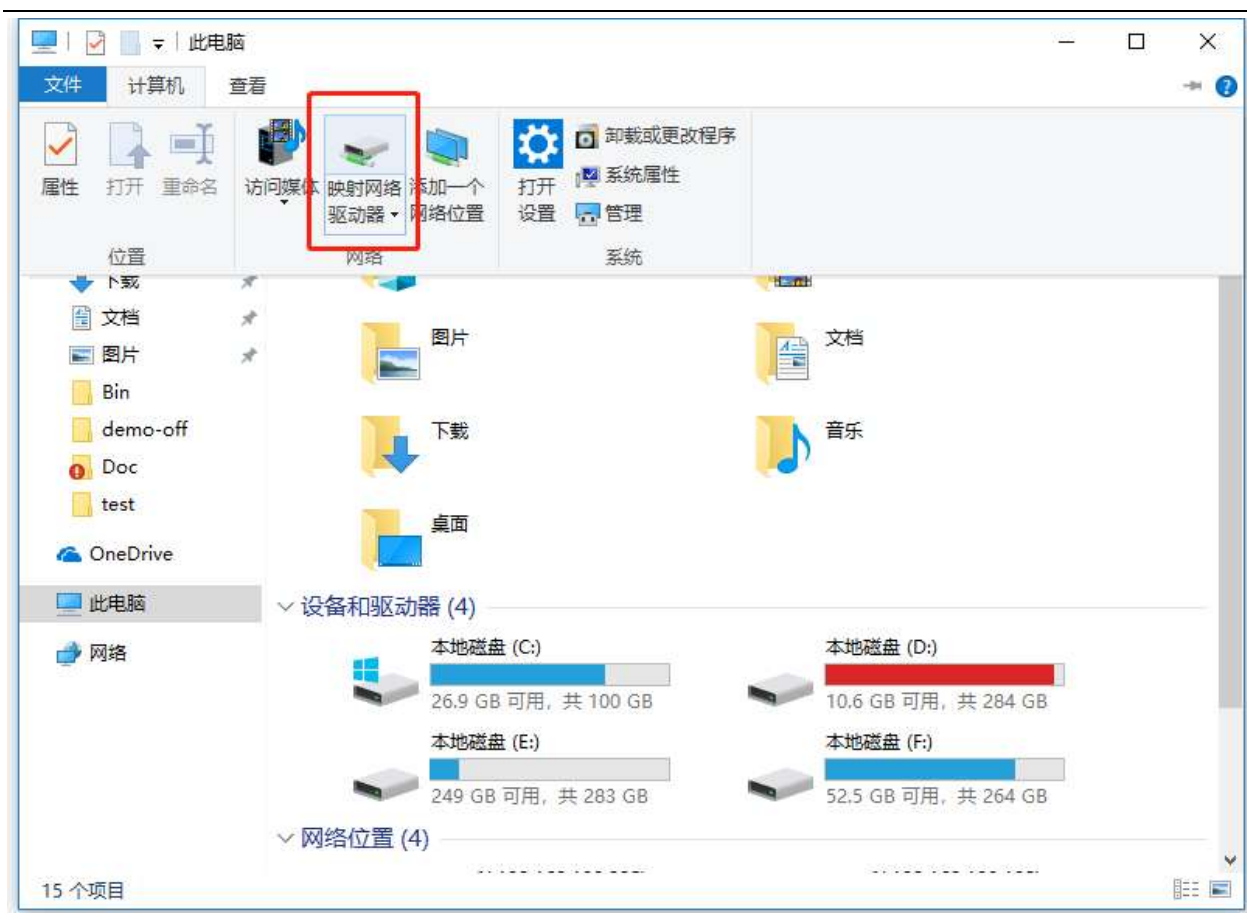
systemctl enable nmb

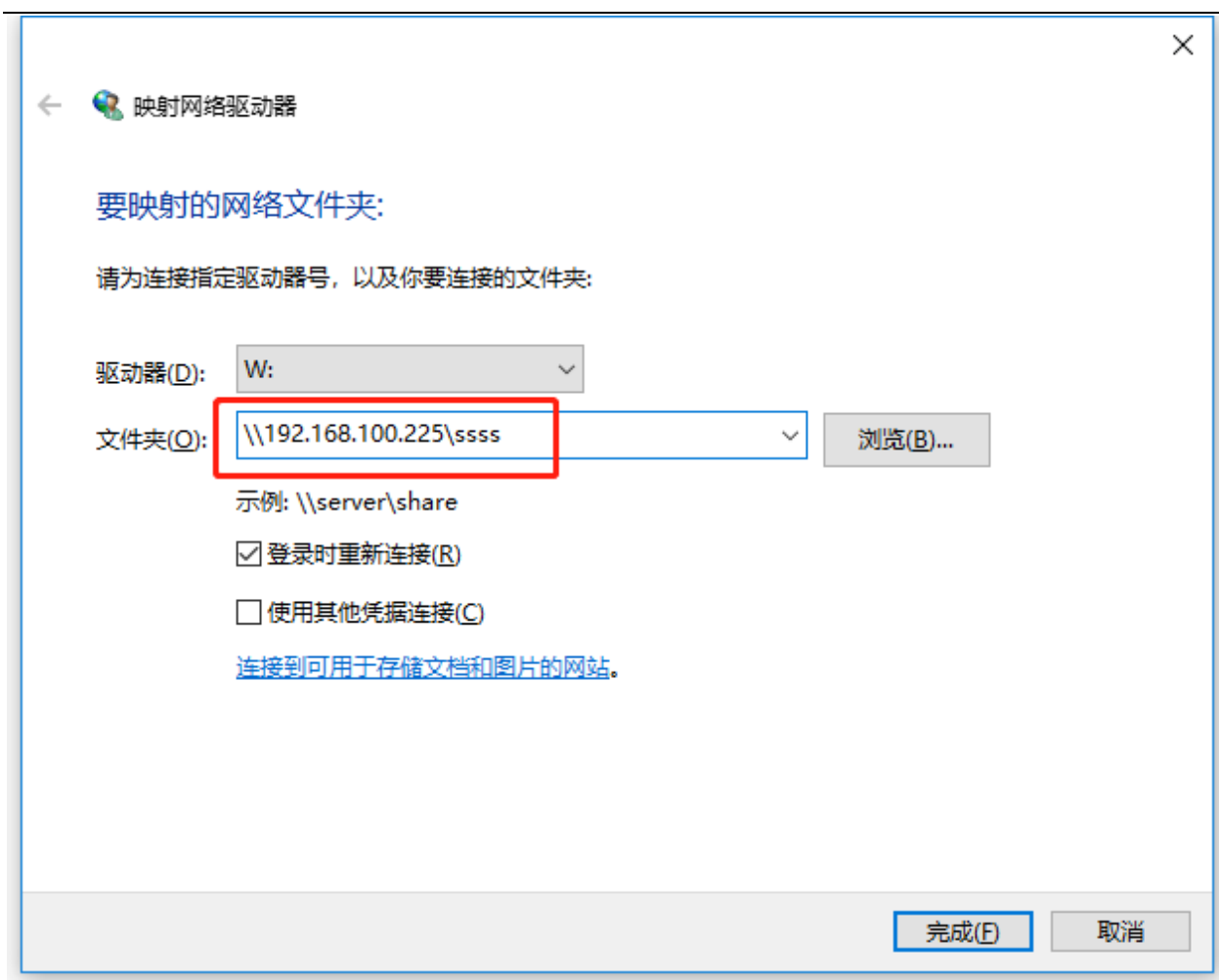
systemctl restart nmb.service

systemctl restart smb.service

3.1.2 Windows 下访问 Linux 的 Samba 服务器

以下步骤为在 Win10 系统下访问 Linux Samba 服务的一种过程（使用 Samba 账户 ssss 访问）：





接下来, 输入 ssss 账户的密码, 即可创建出访问 Samba 服务器的网络磁盘。