

# 编译环境说明

## 账号申请

入职第二天管理员开通邮箱和OA账号，可以发邮件申请开发相关的账号权限，邮件格式参考：

**发件人：**刘波 [mailto:liubo@raisecom.com]  
**发送时间：**2023年4月26日 15:59  
**收件人：**乔强国  
**抄送：**刘爱芳；骆震江  
**主题：**武汉新员工刘波代码权限+虚拟机+编译服务器申请

乔总您好，

我是武汉软件一部新入职员工刘波，工号010888，需要开通代码库权限，麻烦您有空批一下

主要涉及代码如下：

```
git@192.168.34.231:soft3/os.git
```

```
git@192.168.34.231:soft3/ROS6.2.git
```

```
git@192.168.34.231:soft3/product.git
```

```
git@192.168.34.231:soft3_ctc/product_esw_ritp.git
```

```
git@192.168.34.231:iTN8800A/product_ritp.git
```

另外麻烦刘工 @刘爱芳 开通武汉xp虚拟机+编译服务器权限，谢谢！

涉及的代码路径根据需要开通，一般下面四个工程：

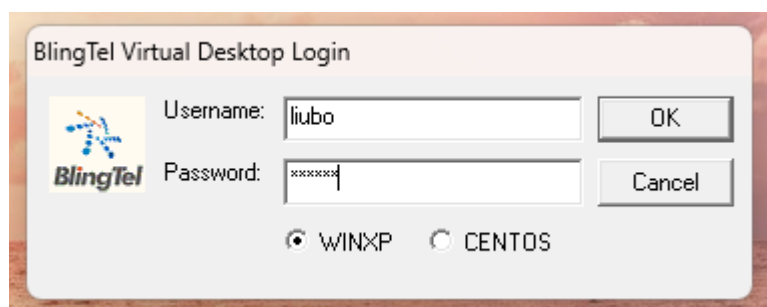
```
git@192.168.34.231:soft3/os.git  
git@192.168.34.231:soft3/ROS6.2.git  
git@192.168.34.231:soft3/product.git  
git@192.168.34.231:soft3_ctc/product_esw_ritp.git
```

## 开发环境准备

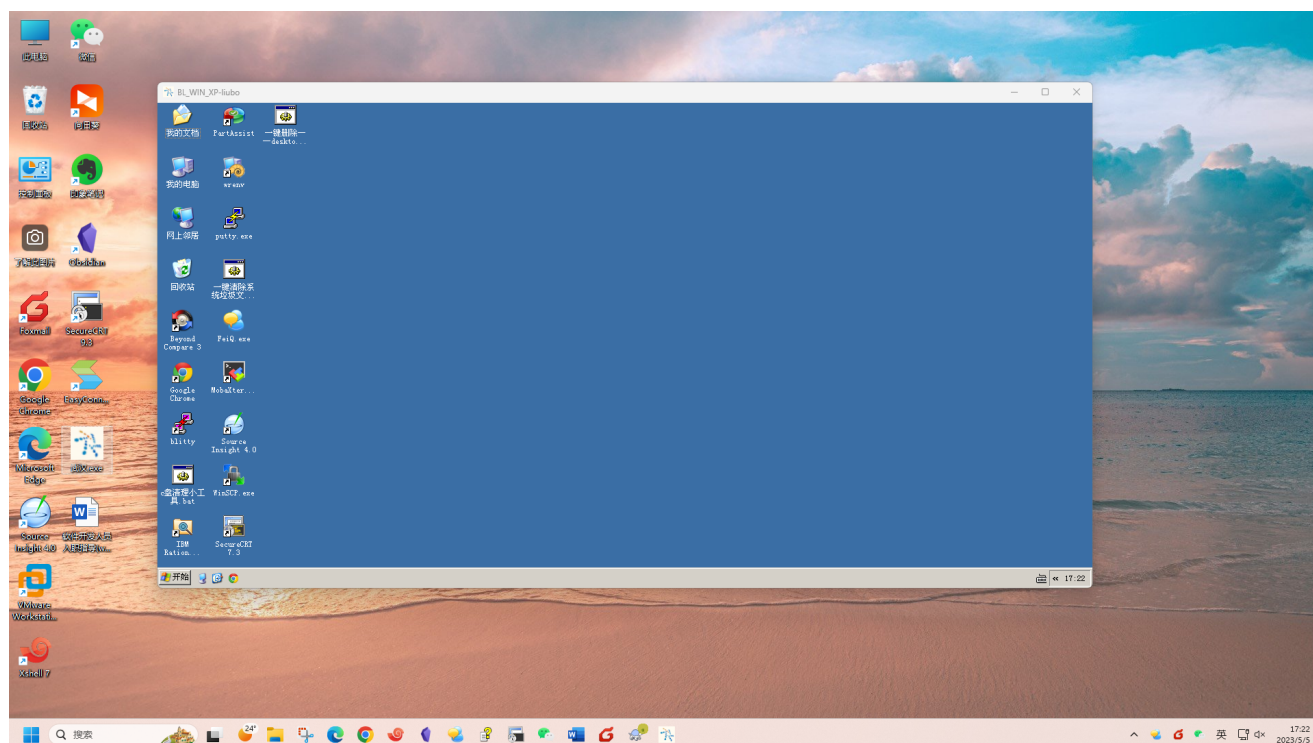
开发需要虚拟机账号、编译机账号和git账号。

## 虚拟机登录

代码阅读、修改和编写都在虚拟机上操作，所以需要通过BlingTelVD\_wuhan.exe这个工具登录自己的虚拟机（XP系统）工作，用户名为姓名全拼，密码为：123456。



登录后进入虚拟机环境，示例图本地是Windows 11系统，分辨率1080P，所以虚拟机界面显示较小，在虚拟机内设置分辨率为1920x1080即可。



双击上图中虚拟机上部显示 BL\_WIN\_XP-liubo 的白色区域，进入虚拟机全屏显示，鼠标左键单击右上角推出虚拟机全屏显示，回到本地Windows11系统。

虚拟机和本地系统可以互传文件，详细说明参考文档《软件开发人员入职指导(wuhan).docx》。该文档和BlingTelVD\_wuhan.exe工具找老同事帮忙传一下。

## 编译机

代码只能在编译机上编译，编译机IP地址在邮件申请的答复邮件中有说明，武汉软件一部编译机IP地址为 10.18.3.202，账号为姓名全拼，密码为 raisecom。在虚拟机上通过SecureCRT登录编译机。

## gitlab密钥导入

在虚拟机中通过 ssh 登录编译机后，配置 git 下载代码。参考文档《GIT快速上手文档.docx》，这里简单说明一下。

ssh 登录后默认进入 `/home/username/` 目录，例如：员工名为 `liubo`，登录后 `pwd` 查看当前目录即为 `/home/liubo`。编译机上生成 ssh 密钥：

```
ssh-keygen -t rsa -b 4096
```

在 `/home/username` 目录下生成一个 `.ssh` 目录，包括两个密钥文件：

```
liubo@wuhan-soft3-202:~$ ls -l .ssh/
total 12
-rw----- 1 liubo liubo 3243  4月 27 16:00 id_rsa
-rw-r--r-- 1 liubo liubo  747  4月 27 16:00 id_rsa.pub
-rw-r--r-- 1 liubo liubo  222  4月 27 15:58 known_hosts
liubo@wuhan-soft3-202:~$
```

用户必须先将 ssh public key 导入 gitlab 才可以进行代码的上传与下载，所以需要复制本地 `id_rsa.pub` 里的内容。打开浏览器，地址栏输入 gitlab 的 IP 地址：`192.168.34.231`。

在网页右上角个人账户下，依次点击 `Settings`、`SSH Keys`，将 `/home/username/.ssh/id_rsa.pub` 里面的内容粘贴过来，然后点击 `Add key`，完成密钥导入。

## 代码下载

在 `/home/username` 目录下创建自己的代码存放目录，例：

```
$ pwd
/home/liubo
$ mkdir work
```

开始下载代码：

```
git clone git@192.168.34.231:soft3/os.git
git clone git@192.168.34.231:soft3/ROS6.2.git ros
git clone git@192.168.34.231:soft3/product.git
git clone git@192.168.34.231:soft3_ctc/product_esw_ritp.git
```

注意，平台的代码 `ROS6.2` 下载到本地时需要修改文件夹名为 `ros`。

## 目录说明

编译代码需要前面下载四个工程目录：`os`、`product`、`product_esw_ritp` 和 `ros`。这四个工程需要放在同一个目录下，`ROS6.2` 下载到本地时需要修改文件夹名为 `ros`。

```

liubo@wuhan-soft3-202:~$ cd work/
liubo@wuhan-soft3-202:~/work$ ls -l
total 20
drwxrwxr-x 12 liubo liubo 4096 4月 28 09:35 os
drwxrwxr-x 35 liubo liubo 4096 4月 27 18:00 product
drwxrwxr-x 27 liubo liubo 4096 4月 27 16:51 product_esw_ritp
drwxrwxr-x 26 liubo liubo 4096 4月 28 11:00 product_ritp
drwxrwxr-x 74 liubo liubo 4096 4月 27 17:23 ros
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$

```

四个工程目录分别对应不同的功能模块，其中，`os` 目录主要实现 BSP 部分的功能，包括：各个设备使用的 boot、不同版本的 linux 内核、不同类型的文件系统、编译链工具、SDK 代码和一些公用代码模块。

```

liubo@wuhan-soft3-202:~/work$ ls os/ -l
total 36
drwxrwxr-x 14 liubo liubo 4096 4月 28 09:33 boot
drwxrwxr-x 15 liubo liubo 4096 4月 28 09:33 build
drwxrwxr-x 5 liubo liubo 4096 4月 28 09:33 drv
drwxrwxr-x 11 liubo liubo 4096 4月 28 09:34 kernel
drwxrwxr-x 5 liubo liubo 4096 5月 4 10:28 rc_common
drwxrwxr-x 10 liubo liubo 4096 4月 28 09:34 rootfs
drwxrwxr-x 3 liubo liubo 4096 4月 28 09:34 sdk
drwxrwxr-x 6 liubo liubo 4096 4月 28 09:35 toolchains
drwxrwxr-x 12 liubo liubo 4096 4月 28 09:35 tools
liubo@wuhan-soft3-202:~/work$

```

`product` 和 `product_esw_ritp` 在目录结构上类似，但是功能不同。`product` 主要包括各个产品编译时需要的静态库文件和平台编译需要的脚本，编译时也会生成一些目标文件，而 `product_esw_ritp` 除了各个产品编译需要的静态库文件，还有编译脚本、各个产品的 SDK 接口和适配层部分代码。

以 `106_iTN8800-A-NXU` 产品目录为例，下图是执行 `product_esw_ritp/106_iTN8800-A-NXU` 目录下的编译脚本后的目录结构，`product` 下的 `bin`、`lib` 和 `obj` 三个目录是编译生成的链接文件和目标文件。

```

liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$ ls product/106_iTN8800-A-NXU/ -l
total 28
drwxrwxr-x 2 liubo liubo 4096 5月 6 10:52 bin
drwxrwxr-x 1 liubo liubo 49 4月 27 17:59 devdef.h -> ../../product_esw_ritp/106_iTN8800-A-NXU/devdef.h
drwxrwxr-x 2 liubo liubo 4096 5月 6 10:52 lib
drwxrwxr-x 2 liubo liubo 4096 4月 27 17:59 make
drwxrwxr-x 1 liubo liubo 49 4月 27 17:59 moddef.h -> ../../product_esw_ritp/106_iTN8800-A-NXU/moddef.h
-rwxrwxr-x 1 liubo liubo 3237 4月 27 17:59 model_cardif.c
drwxrwxr-x 8 liubo liubo 4096 5月 6 10:52 obj
drwxrwxr-x 4 liubo liubo 4096 4月 27 17:59 obj_src
-rwxrwxr-x 1 liubo liubo 1604 4月 27 17:59 plat.sh
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$ ls product_esw_ritp/106_iTN8800-A-NXU/ -l
total 280
drwxrwxr-x 2 liubo liubo 4096 4月 27 16:51 burn
drwxrwxr-x 4 liubo liubo 4096 4月 27 16:51 dev
-rwxrwxr-x 1 liubo liubo 49458 5月 4 11:27 devdef.h
drwxrwxr-x 3 liubo liubo 4096 5月 4 11:08 make
-rwxrwxr-x 1 liubo liubo 2932 4月 27 16:51 Makefile
-rwxrwxr-x 1 liubo liubo 34489 4月 27 16:51 mitn8800a_nxub.sh
-rwxrwxr-x 1 liubo liubo 34509 5月 4 13:50 mitn8800a_nxuc.sh
-rwxrwxr-x 1 liubo liubo 34489 4月 27 16:51 mitn8800a.sh
-rwxrwxr-x 1 liubo liubo 63318 5月 4 11:27 moddef.h
-rwxrwxr-x 1 liubo liubo 3237 4月 27 16:51 model_cardif.c
drwxrwxr-x 5 liubo liubo 4096 4月 27 16:51 mytools
drwxrwxr-x 7 liubo liubo 4096 5月 4 11:17 obj
drwxrwxr-x 3 liubo liubo 4096 4月 27 16:51 obj_src
drwxrwxr-x 2 liubo liubo 4096 5月 4 11:13 paf
drwxrwxr-x 2 liubo liubo 4096 5月 4 11:16 script
drwxrwxr-x 2 liubo liubo 4096 4月 27 16:51 version
-rwxrwxr-x 1 liubo liubo 116 4月 27 16:51 version.c
-rw-rw-r-- 1 liubo liubo 1248 5月 4 11:13 version.o
drwxrwxr-x 4 liubo liubo 4096 4月 27 16:51 weblib
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$

```

最终构建版本时使用的是 `product_esw_ritp/106_iTN8800-A-NXU` 目录下编译生成的链接文件，而不是 `product` 目录。`product` 目录只是编译时使用，日常开发一般在 `product_esw_ritp` 目录，因为 `product_esw_ritp` 目录不仅包括各个产品子目录，还包括 SDK 接口和适配层代码：

```
liubo@wuhan-soft3-202:~/work$  
liubo@wuhan-soft3-202:~/work$ ls product_esw_ritp/ -l  
total 96  
drwxrwxr-x 8 liubo liubo 4096 4月 27 16:51 077_iTN8800-A-RK4  
drwxrwxr-x 6 liubo liubo 4096 4月 27 16:51 085_iTN8800-A-RC1  
drwxrwxr-x 8 liubo liubo 4096 4月 27 16:51 087_iTN8800-E-DEMO  
drwxrwxr-x 9 liubo liubo 4096 4月 27 16:51 088_SLH6300I  
drwxrwxr-x 11 liubo liubo 4096 5月 4 11:27 090_iTN8800-E-NXU  
drwxrwxr-x 12 liubo liubo 4096 5月 4 11:27 101_721A  
drwxrwxr-x 10 liubo liubo 4096 4月 27 16:51 103_SLH63000i-48GE  
drwxrwxr-x 9 liubo liubo 4096 4月 27 16:51 104_711C  
drwxrwxr-x 10 liubo liubo 4096 4月 27 16:51 105_iTN331-E  
drwxrwxr-x 12 liubo liubo 4096 5月 4 13:50 106_iTN8800-A-NXU  
drwxrwxr-x 10 liubo liubo 4096 4月 27 16:51 108_721A_T  
drwxrwxr-x 5 liubo liubo 4096 4月 27 16:51 109_iTN8800-A-RG10  
drwxrwxr-x 6 liubo liubo 4096 4月 27 16:51 110_iTN8800-A-RX2S  
drwxrwxr-x 10 liubo liubo 4096 4月 27 16:51 111_721C  
drwxrwxr-x 5 liubo liubo 4096 4月 27 16:51 112_iTN8800-A-RL2  
drwxrwxr-x 5 liubo liubo 4096 4月 27 16:51 113_iTN8800-A-RS2  
drwxrwxr-x 6 liubo liubo 4096 4月 27 16:51 114_iTN8800-A-RC2  
drwxrwxr-x 10 liubo liubo 4096 5月 5 14:33 115_RQX721-C-XC  
drwxrwxr-x 17 liubo liubo 4096 4月 27 16:51 drv  
drwxrwxr-x 4 liubo liubo 4096 4月 27 16:51 fdi_ctc  
drwxrwxr-x 11 liubo liubo 4096 4月 27 16:51 module  
drwxrwxr-x 6 liubo liubo 4096 4月 27 16:51 sdk  
drwxrwxr-x 52 liubo liubo 4096 4月 27 16:51 ssp_ctc  
drwxrwxr-x 5 liubo liubo 4096 4月 27 16:51 tools
```

其中，

- `drv`：底层驱动部分代码，包括 `fpga`、`phy` 以及其他定制功能；
- `fdi_ctc`：主要是封装 SDK 接口，供适配层 `ssp_ctc` 下的各个功能模块调用；
- `sdk`：不同版本的 SDK 代码；
- `ssp_ctc`：适配层代码，对上供平台配置查询调用，对下调用 `fdi_ctc` 提供的封装接口，但是目前也直接调用 SDK 的接口；

对不同芯片做适配，或者实现定制功能时，一般在上面几个目录进行开发。

## 编译

### 编译脚本执行

`product_esw_ritp` 下各个产品目录都有自己的编译脚本，以 `106_iTN8800-A-NXU` 产品目录为例：

```

liubo@wuhan-soft3-202:~/work$ ls product/106_iTN8800-A-NXU/ -l
total 28
drwxrwxr-x 2 liubo liubo 4096 5月 6 10:52 bin
lrwxrwxrwx 1 liubo liubo 49 4月 27 17:59 devdef.h -> ../../product_esw_rtp/106_iTN8800-A-NXU/devdef.h
drwxrwxr-x 2 liubo liubo 4096 5月 6 10:52 lib
drwxrwxr-x 2 liubo liubo 4096 4月 27 17:59 make
lrwxrwxrwx 1 liubo liubo 49 4月 27 17:59 moddef.h -> ../../product_esw_rtp/106_iTN8800-A-NXU/moddef.h
-rwxrwxr-x 1 liubo liubo 3237 4月 27 17:59 model_cardif.c
drwxrwxr-x 8 liubo liubo 4096 5月 6 10:52 obj
drwxrwxr-x 4 liubo liubo 4096 4月 27 17:59 obj_src
-rwxrwxr-x 1 liubo liubo 1604 4月 27 17:59 plat.sh
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$ ls product_esw_rtp/106_iTN8800-A-NXU/ -l
total 280
drwxrwxr-x 2 liubo liubo 4096 4月 27 16:51 burn
drwxrwxr-x 4 liubo liubo 4096 4月 27 16:51 dev
-rwxrwxr-x 1 liubo liubo 49458 5月 4 11:27 devdef.h
drwxrwxr-x 3 liubo liubo 4096 5月 4 11:08 make
-rwxrwxr-x 1 liubo liubo 2932 4月 27 16:51 Makefile
-rwxrwxr-x 1 liubo liubo 34489 4月 27 16:51 mitn8800a_nxub.sh
-rwxrwxr-x 1 liubo liubo 34509 5月 4 13:50 mitn8800a_nxuc.sh
-rwxrwxr-x 1 liubo liubo 34489 4月 27 16:51 mitn8800a.sh
-rwxrwxr-x 1 liubo liubo 63318 5月 4 11:27 moddef.h
-rwxrwxr-x 1 liubo liubo 3237 4月 27 16:51 model_cardif.c
drwxrwxr-x 5 liubo liubo 4096 4月 27 16:51 mytools
drwxrwxr-x 7 liubo liubo 4096 5月 4 11:17 obj
drwxrwxr-x 3 liubo liubo 4096 4月 27 16:51 obj_src
drwxrwxr-x 2 liubo liubo 4096 5月 4 11:13 paf
drwxrwxr-x 2 liubo liubo 4096 5月 4 11:16 script
drwxrwxr-x 2 liubo liubo 4096 4月 27 16:51 version
-rwxrwxr-x 1 liubo liubo 116 4月 27 16:51 version.c
-rwxrwxr-x 1 liubo liubo 1248 5月 4 11:13 version.o
drwxrwxr-x 4 liubo liubo 4096 4月 27 16:51 weblib
liubo@wuhan-soft3-202:~/work$
liubo@wuhan-soft3-202:~/work$

```

mitn8800a\*.sh 这三个脚本都可以进行编译，可以根据需求在编译选项上有所不同。执行脚本 mitn8800a\_nxuc.sh，不带参数后会显示可供选择的编译模块：

```

liubo@wuhan-soft3-202:~/work/product_esw_rtp/106_iTN8800-A-NXU$
liubo@wuhan-soft3-202:~/work/product_esw_rtp/106_iTN8800-A-NXU$ ./mitn8800a_nxuc.sh
/home/liubo/work
/home/liubo/work/os
/home/liubo/work/os/toolchains/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu
Help:sh mSLH6300I,sh (all|kernel|sdk|dev|dms|drv|ssp|fdi|module|tdmop|mytools|product|ros|ritp|boot|rbbsp_drv|rbbsp_linux|rbbsp_rc_comm
on|rbbsp) [clean]
all          编译所有模块，包括ros,ritp下所有模块
kernel      编译内核及根文件系统
sdk          编译CTC SDK
dev          编译dev模块
dms          编译dms模块
drv          编译drv模块
ssp          编译ssp模块
fdi          编译fdi模块
module      编译module模块，不包含dms、tdmop模块
tdmop        编译tdmop模块
mytools      编译mytools模块
product      编译ritp目录下所有模块,适用于重新编译ritp下所有修改
ros          编译ros目录下所有模块
ritp         只连接ritp生成文件，一般编译各自模块后，使用此命令连接ritp
boot        编译boot启动引导系统
rbbsp_drv   编译rbbsp用户驱动模块
rbbsp_linux 编译rbbsp内核模块
paf          制作paf文件
liubo@wuhan-soft3-202:~/work/product_esw_rtp/106_iTN8800-A-NXU$

```

代码下载后第一次编译选择 all 参数，完成后会构建一个 system.bin 版本，并有邮件提示。后续开发时改动哪个模块的代码选择对应的模块参数即可。

修改 product\_esw\_rtp/fdi\_ctc 目录下的文件单独编译时指定参数 fdi，修改 product\_esw\_rtp/ssp\_ctc 目录下的文件单独编译时指定参数 ssp。如果修改的是 product\_esw\_rtp/fdi\_ctc 目录下的文件，指定编译参数为 ssp 时，fdi\_ctc 下的修改不会被编译。

单个模块编译成功会有如下提示，指定参数 all，或者 ros 时会编译多个模块，所以编译过程中某个模块编译完成时也会有对应模块编译成功提示。

```
make[2]: Nothing to be done for `ritplib'.
make[2]: Leaving directory `/home/liubo/work/product_esw_ritp/ssp_ctc/ssp_synce/src'
make[1]: Leaving directory `/home/liubo/work/product_esw_ritp/ssp_ctc/ssp_synce'
/home/liubo/work/os/toolchains/gcc-linaro-7.5.0-2019.12-x86_64_aarch64-linux-gnu/bin/aarch64-linux-gnu-ar crus /home/liubo/work/product_esw_ritp/106_iTN8800-A-NXU/obj/ssp/libssp.a /home/liubo/work/product_esw_ritp/106_iTN8800-A-NXU/obj/ssp/*.o
mv -f /home/liubo/work/product_esw_ritp/106_iTN8800-A-NXU/obj/ssp/libssp.a /home/liubo/work/product_esw_ritp/106_iTN8800-A-NXU/obj/ritp/libssp.a
make ssp success
liubo@wuhan-soft3-202:~/work/product_esw_ritp/106_iTN8800-A-NXU$
liubo@wuhan-soft3-202:~/work/product_esw_ritp/106_iTN8800-A-NXU$
```

## 编译文件打包

执行编译脚本指定参数 `all` 时，会在 `product_esw_ritp/106_iTN8800-A-NXU/obj/` 目录下生成一个可以用于升级的 `system.bin` 文件。

名称	大小	类型	修改日期
drv		文件夹	2023-5-4 11:11
module		文件夹	2023-5-4 11:13
ritp		文件夹	2023-5-6 14:16
ros		文件夹	2023-5-6 10:52
ssp		文件夹	2023-5-6 14:16
charon.gz	3,461 KB	WinRAR 压缩文件	2023-5-4 11:17
fsl-ls1043a-rc-product-sdk.dtb	33 KB	DTB 文件	2023-5-4 11:17
Image.gz	39,329 KB	WinRAR 压缩文件	2023-5-4 11:17
itn8800_a_nxub.itb	39,363 KB	ITB 文件	2023-5-4 11:17
paf.ini	5 KB	配置设置	2023-5-4 11:13
ritp.gz	49,184 KB	WinRAR 压缩文件	2023-5-4 11:17
rnet.gz	14,242 KB	WinRAR 压缩文件	2023-5-4 11:17
ros.gz	76,023 KB	WinRAR 压缩文件	2023-5-4 11:17
system.bin	182,270 KB	BIN 文件	2023-5-4 11:18

上图是编译机映射到 Windows 的目录，显示更清楚一点。编译生成的 `.d`、`.o`、`.a` 等文件都在 `obj` 目录下的各个文件夹中。

比如，指定编译参数 `ssp` 时，会在 `obj/ssp` 下生成各个适配层模块的 `.d`、`.o` 文件，在 `obj/ritp` 下生成 `libssp.a` 文件。指定编译参数 `fdi` 时，则会在 `obj/ritp/fdi` 下生成 `.d`、`.o` 文件，在 `obj/ritp` 下生成 `libfdi.a` 文件。其他模块如 `drv`、`rbsp`、`module` 或者 `mytools` 编译时都会 `obj/ritp` 或者 `obj/ritp` 下的子目录生成对应的 `.a` 文件。

编译参数指定 `ros` 时，则会在 `obj/ros/lib` 下生成 `.a` 文件，在 `obj/ros/bin` 下生成可执行文件：

名称	大小	类型	修改日期
charon	8,189 KB	文件	2023-5-6 10:52
rnet	31,662 KB	文件	2023-5-6 10:34
ros	254,348 KB	文件	2023-5-6 10:52
ssp	6,002 KB	文件	2023-5-6 10:52

1. 打包 `obj/ritp` 下所有文件，生成 `ritp.gz` 文件，并将打包的文件移动到 `obj/` 目录。



地址 (D) Z:\product_esw_ritp\106_iTN8800-A-NXU\obj\ritp			
名称	大小	类型	修改日期
dev		文件夹	2023-5-5 16:14
dms		文件夹	2023-5-5 16:14
fdi		文件夹	2023-5-6 14:04
mytools		文件夹	2023-5-4 11:13
sdk		文件夹	2023-5-4 11:08
tdmop		文件夹	2023-5-4 11:13
bcmshell	19 KB	文件	2023-5-4 11:13
bcmshell.dbg	24 KB	DBG 文件	2023-5-4 11:13
libdrv.a	1,279 KB	A 文件	2023-5-6 14:39
libfdi.a	8,331 KB	A 文件	2023-5-6 14:04
libmodule.a	7,158 KB	A 文件	2023-5-4 11:13
librbsp.a	637 KB	A 文件	2023-5-4 10:28
librbsp_rc_common.a	44 KB	A 文件	2023-5-4 10:28
libssp.a	49,440 KB	A 文件	2023-5-6 14:16
libtdmop.a	8,054 KB	A 文件	2023-5-4 11:13
rcdb	158 KB	文件	2023-5-4 11:13
rcdb.dbg	281 KB	DBG 文件	2023-5-4 11:13
ritp.gz	49,184 KB	WinRAR 压缩文件	2023-5-4 11:13

2. 分别打包 obj/ros/bin 下的 charon、rnet、ros，生成 charon.gz、rnet.gz、ros.gz 文件，并将打包的文件移动到 obj/ 目录。
3. 前两步生成的四个 .gz 文件都移动到 obj/ 目录后，使用工具加上校验头打包生成 system.bin 文件。