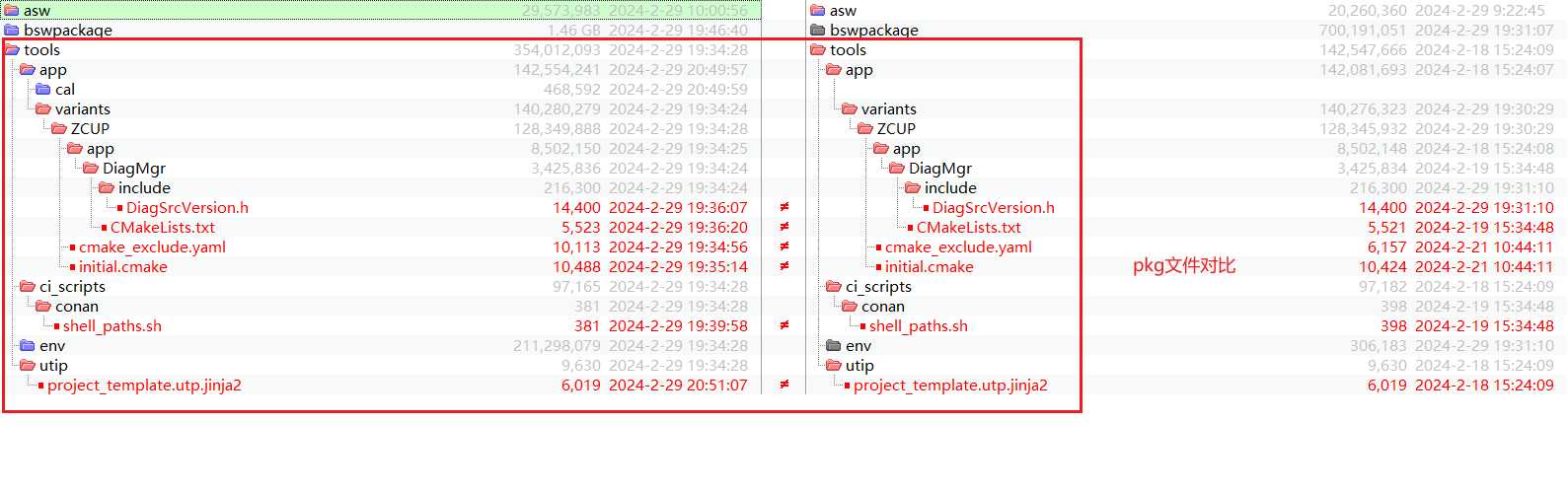
**总集成环境流程手册**

|  |  |  |  |
| --- | --- | --- | --- |
| 文件修订记录 REVISION HISTORY OF THIS DOCUMENT | | | |
| 版本号 Version no | 创建者/修订者 Create/Changed by | 修订日期 Data | 修订记录 Description of Changes |
| V1.0 | 蔡宗阳 | 2024/02/27 | 1、ZCUP总集成环境流程手册初版[已评审] |

为了确保新人快速上手总集成的生成和释放，本指导手册会详细描述，在收到conti提交的代码后，具体做哪些操作。（以ZCUP为例）

千万注意！！！！！！！！！！！！



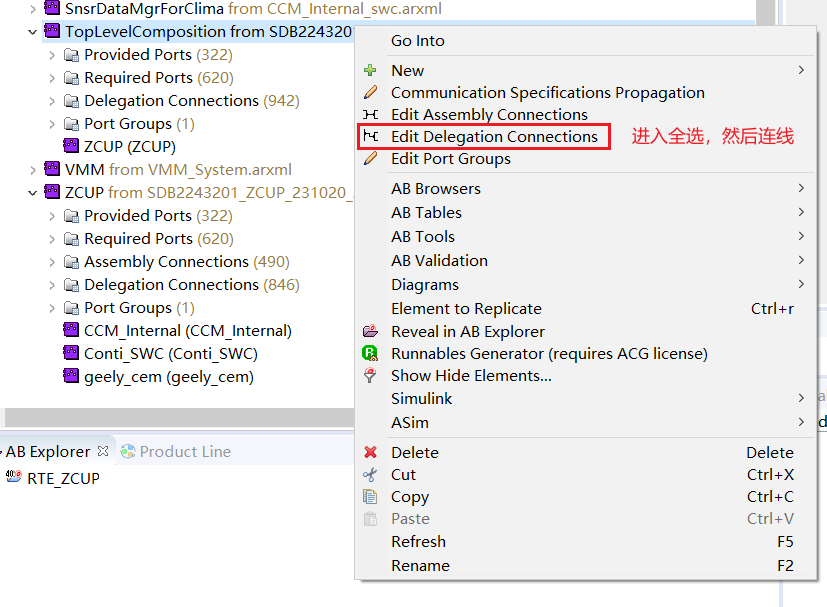
# 预编译环境释放：

## 一、SDB Merge

SDB修正合并自动化流水线以及使用手册如下：



ZCU层 delegation到Top层



EB消错，消错完，可以释放SDB

【首先确定我们是总集成还是Conti是总集成】

## 二、接口配置

应用层接口表：[接口表汇总\_ZCU\_E245 - Files - CASE Share (geely-test.com)](http://caseshare.geely-test.com/apps/files/?dir=/%E7%94%B5%E5%AD%90%E7%94%B5%E5%99%A8-%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91/ZCU%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91/%E6%8E%A5%E5%8F%A3%E8%A1%A8%E6%B1%87%E6%80%BB_ZCU_E245&fileid=382389)

接口配置文档：[AB软件操作 - Files - CASE Share (geely-test.com)](http://caseshare.geely-test.com/apps/files/?dir=/%E7%94%B5%E5%AD%90%E7%94%B5%E5%99%A8-%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91/%E5%9F%BA%E7%A1%80%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E4%B8%8E%E9%9B%86%E6%88%90%20-%20BSW%20and%20integration/%E9%9B%86%E6%88%90%E9%83%A8%E6%96%B0%E5%91%98%E5%B7%A5%E5%85%A5%E8%81%8C/%E5%B7%A5%E5%85%B7%E4%BD%BF%E7%94%A8/AB_EB/AB%E8%BD%AF%E4%BB%B6%E6%93%8D%E4%BD%9C&fileid=401752)

选择：人工配置和流水线配置



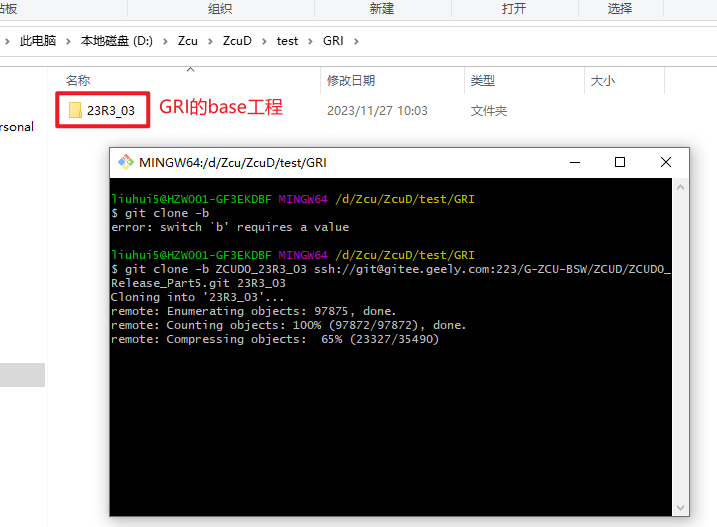
配置完成后，进行接口验证，使用脚本，生成接口表检查报告，命名-V1，并上传

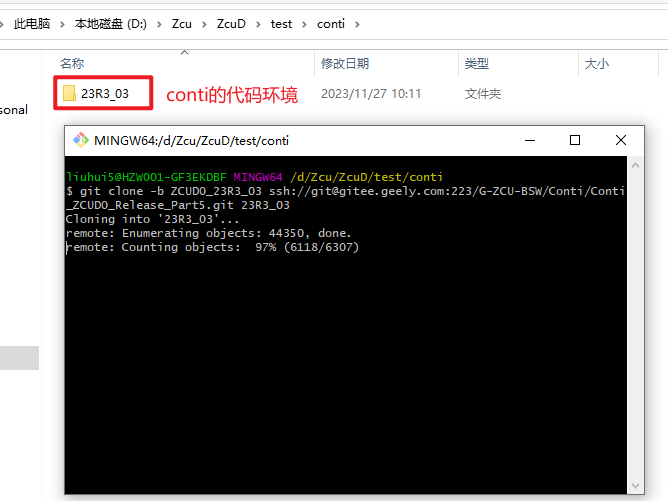
http://caseshare.geely-test.com/apps/files/?dir=/%E7%94%B5%E5%AD%90%E7%94%B5%E5%99%A8-%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91/%E5%9F%BA%E7%A1%80%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E4%B8%8E%E9%9B%86%E6%88%90%20-%20BSW%20and%20integration/ZCU/SWE5/ZCU-%E9%9B%86%E6%88%90%E6%B5%8B%E8%AF%95%E6%8A%A5%E5%91%8A/%E6%B5%8B%E8%AF%95log/ZCUP/23R3%20U1\_02\_01-Interfaces&fileid=887350

准备合并大陆和CCM的工程，准备释放预编译环境。

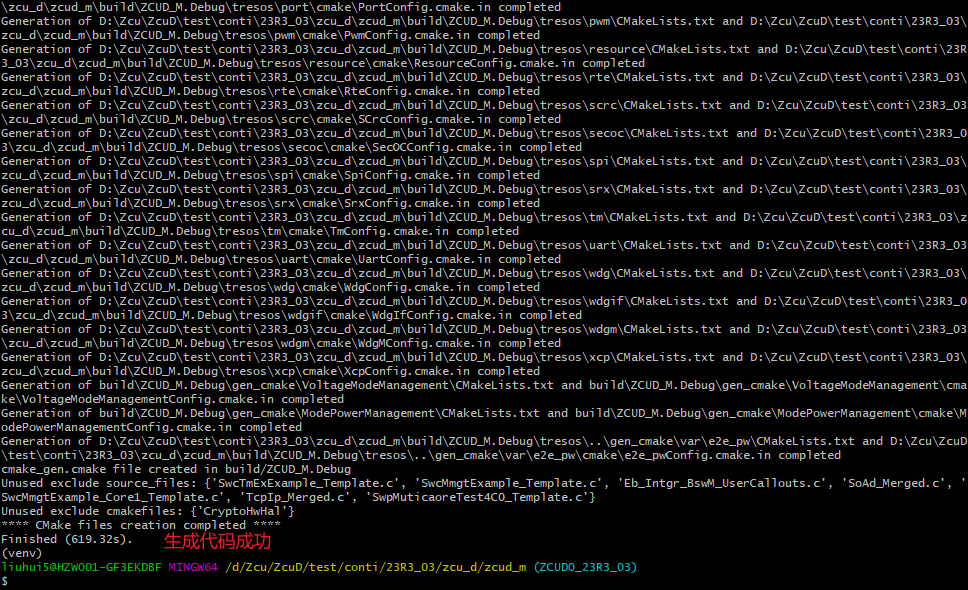
## 三、明确要操作的工程对象

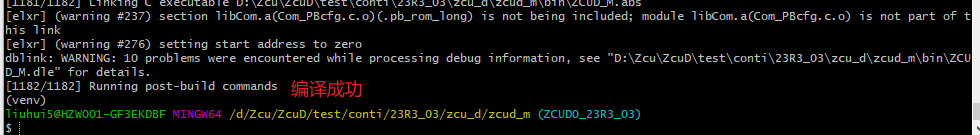
总集成环境主要是将conti的代码工程合并到GRI的工程中，首先需要在本地确认好GRI的工程和conti的工程。

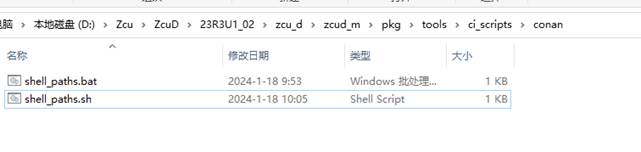
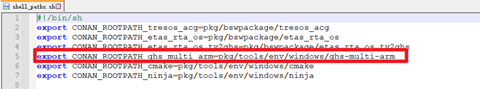
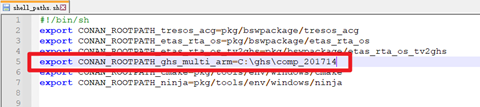




## 四、验证conti的工程是否可以生成代码及编译通过





如果编译失败，执行以下操作  
1.修改编译大陆编译文件，路径是 …\zcu\_d\zcud\_m\pkg\tools\ci\_scripts\conan\shell\_paths.sh2.将下图设置为set CONAN\_ROOTPATH\_ghs\_multi\_arm=C:\ghs\comp\_201714 

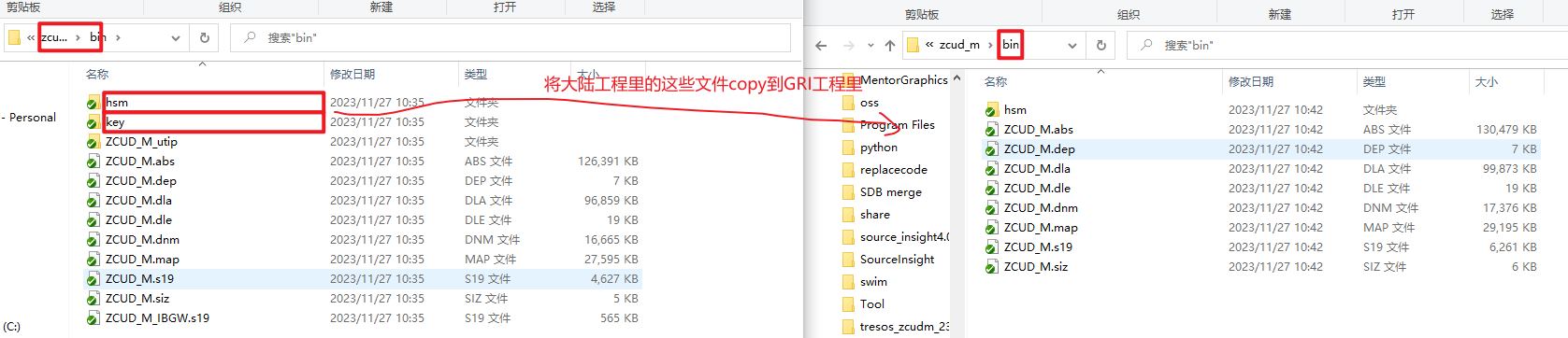
3.然后重复编译操作

## 五、工程替换

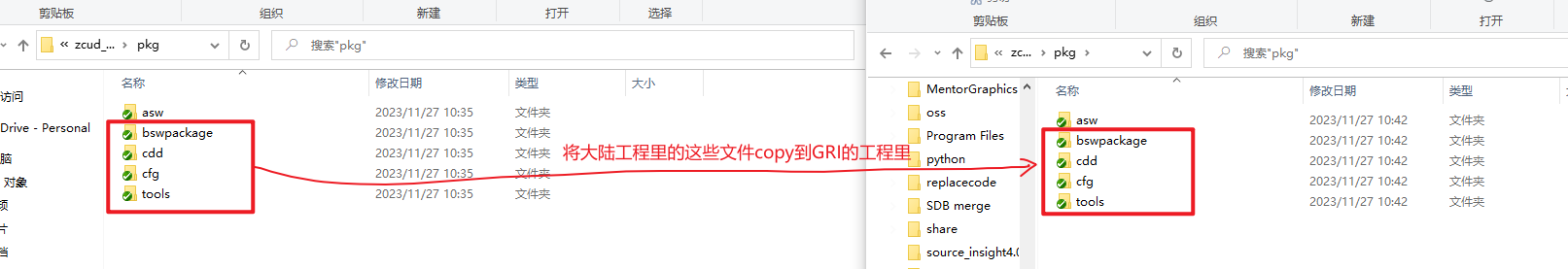
将conti工程里的部分文件，copy到GRI的过程中，以GRI的工程为主。

（1）、bin文件夹下

大陆工程包含key文件夹则替换，否则替换hsm文件夹



（2）、pkg文件夹下 删除原工程的四个文件夹，然后复制

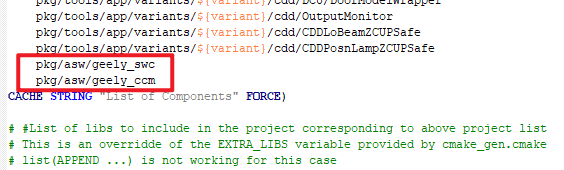


替换完文件后，需要修改cmake\_exclude.yaml和initial.cmake，具体如下：

cmake\_exclude.yaml：把所有GRI的SWC的Template.c信息写出来。确保RTE生成代码时不生成SWC\_Template.c文件。

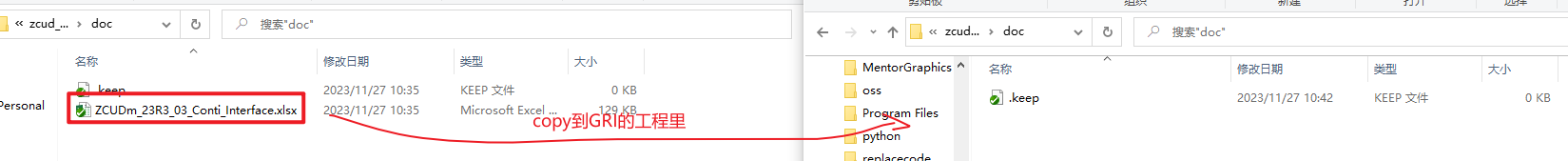


initial.cmake：修改两点，添加GRI的代码

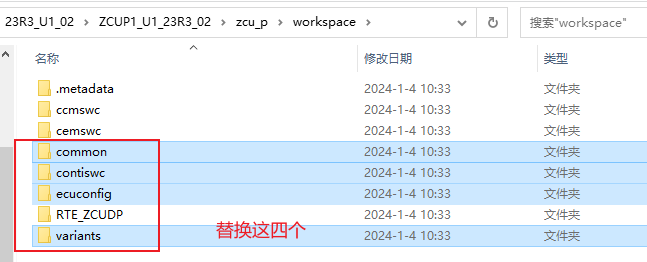




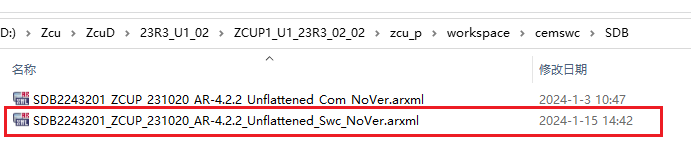
（3）、doc文件夹下



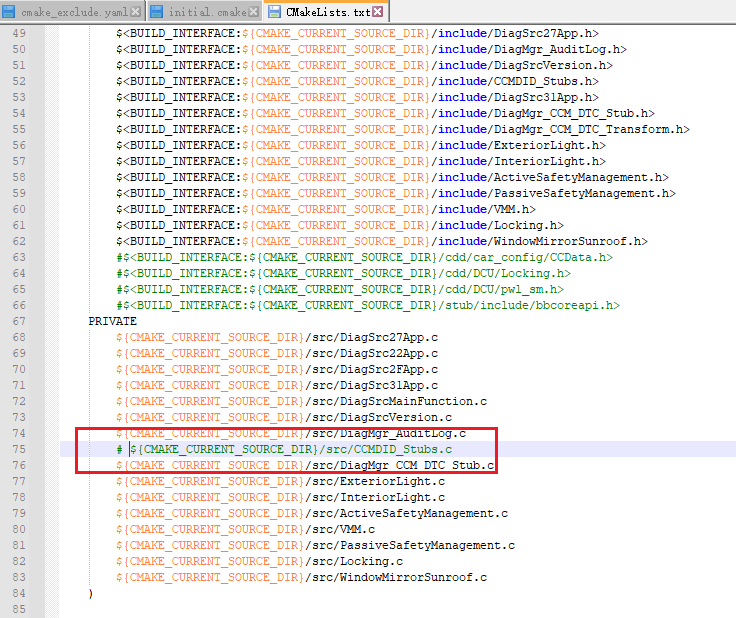
（4）、workspace文件夹下



【注意：首先确定我们是总集成还是Conti是总集成，Conti总集成的话，需要替换Cemswc的swc\_SDB】

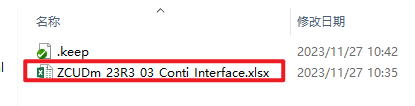


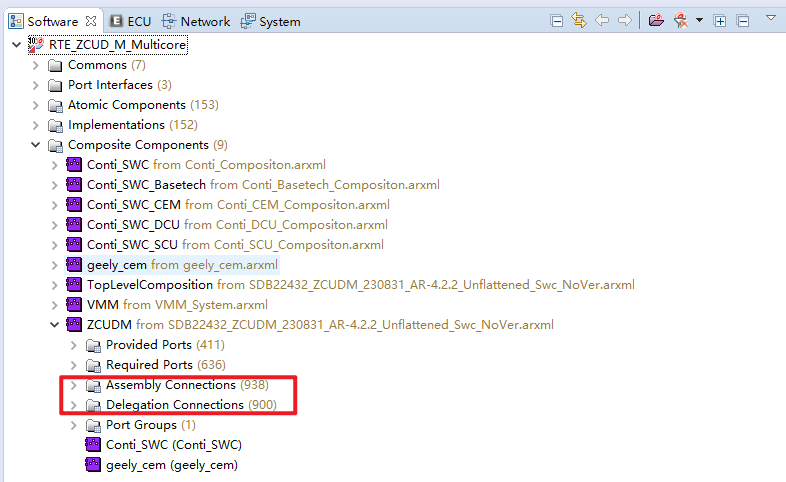
在全部替换完毕后，注释该路径下文件的CCMDID\_Stubs.c这一行，避免CCM 与Conti的代码冲突。D:\Zcu\ZcuD\23R3\_U1\_02\ZCUP1\_U1\_23R3\_02\_02\zcu\_p\pkg\tools\app\variants\ZCUP\app\DiagMgr Cmakelist.txt中的CCMDID



## 六、使用AB工具进行连线（GRI&conti&CCM、Network）

根据大陆的接口表信息进行内部连线及网络连线





配置完成后，进行接口验证，使用脚本，生成接口表检查报告，命名-V2，并上传

[23R3 U1\_02\_01-Interfaces - Files - CASE Share (geely-test.com)](http://caseshare.geely-test.com/apps/files/?dir=/%E7%94%B5%E5%AD%90%E7%94%B5%E5%99%A8-%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91/%E5%9F%BA%E7%A1%80%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E4%B8%8E%E9%9B%86%E6%88%90%20-%20BSW%20and%20integration/ZCU/SWE5/ZCU-%E9%9B%86%E6%88%90%E6%B5%8B%E8%AF%95%E6%8A%A5%E5%91%8A/%E6%B5%8B%E8%AF%95log/ZCUP/23R3%20U1_02_01-Interfaces&openfile=893041&scrollto=%E6%8E%A5%E5%8F%A3%E8%A1%A8%E6%B1%87%E6%80%BB_ZCUP_E245_23R3_U1_02-V2.xlsx)

预编译做到这一步即可，接下来去执行预编译释放流水线：

[ZCU\_Conti\_BSW\_PreCompile [ZCU\_Conti\_BSW\_Compile] [Jenkins]](http://10.24.24.176/view/ZCU/job/ZCU_BSW_FullCompile/job/ZCU_Conti_BSW_PreCompile/)

git commit -m “[preCompile]:ZCUP#PreCompile Relsease:ZCUP1\_23R3\_02\_01”

# 总编译环境释放

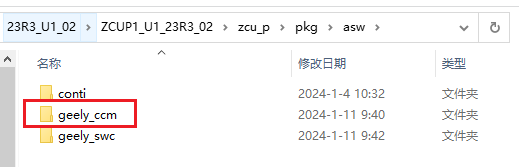
**总编译环境释放前，Conti会再次发送环境，需要重复步骤三-五，**

## 七、代码编译

GRI替换



CCM替换

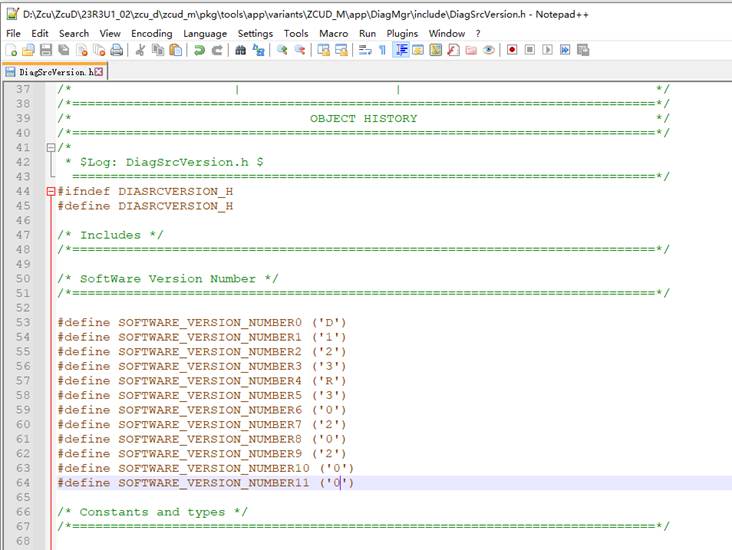


**然后再替换应用层相关file\_c,file\_h,**



**然后继续执行步骤七及其下面步骤**

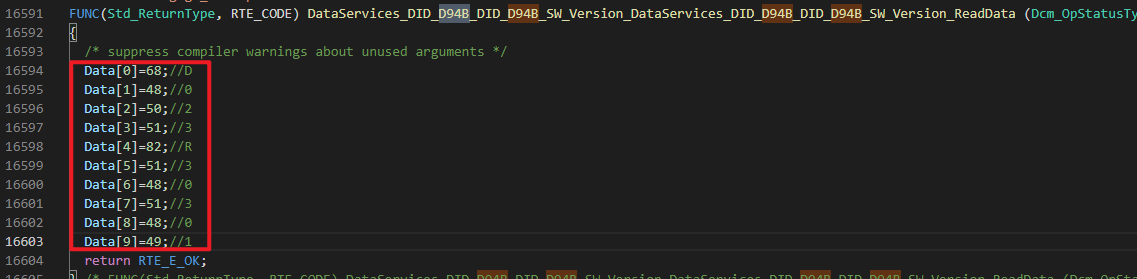
## 八、填写DID（D94B）



修改文件  
D:\Zcu\ZcuD\23R3U1\_02\zcu\_d\zcud\_m\pkg\tools\app\variants\ZCUD\_M\app\DiagMgr\include\DiagSrcVersion.h

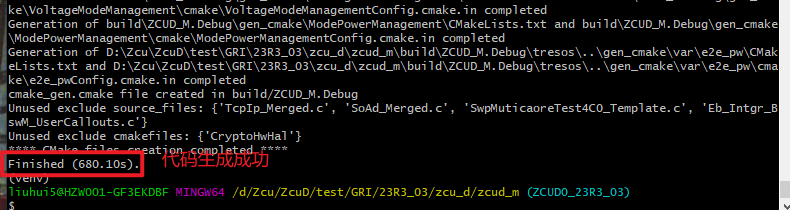
脚本暂时不使用，下面是脚本流程  
1. 先执行GenVersion.py脚本  
python geely\_utils/CheckInfo/GenVersion.py  
-v D123R30201





## 九、RTE验证总集成环境（指令生成）

按照相关指令运行：

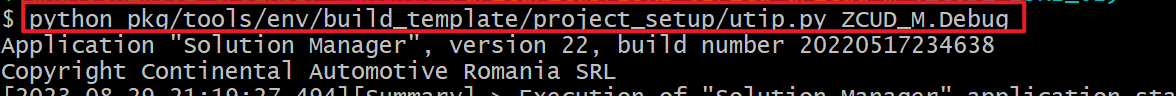


## 十、build总编译

python pkg/tools/env/zcu\_builder.py -p ZCUP -b Debug -c true

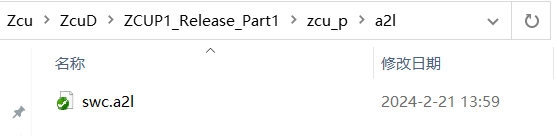
## 十一、生成utip

运行以下指令，生成utip文件



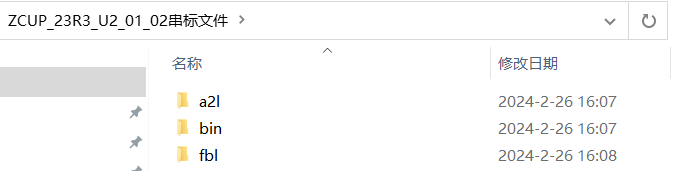
## 十二、生成A2L

拷贝到a2l文件夹下



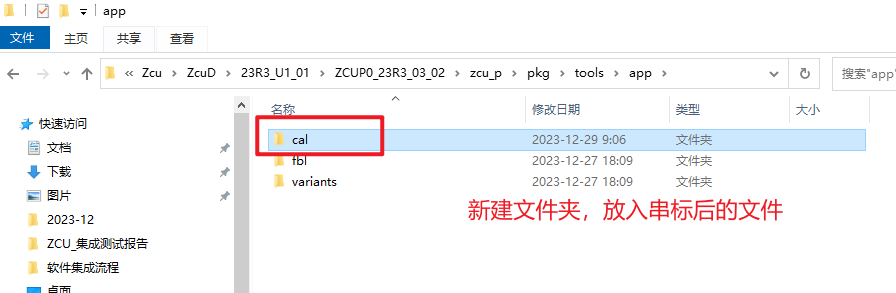
## 十三、串标

CRI这边向CCM提供以下文件



串标完成后，CCM提供两个S19文件，命名要跟下面一致





**后续环节都将替换，注释大陆该文件的两行：**

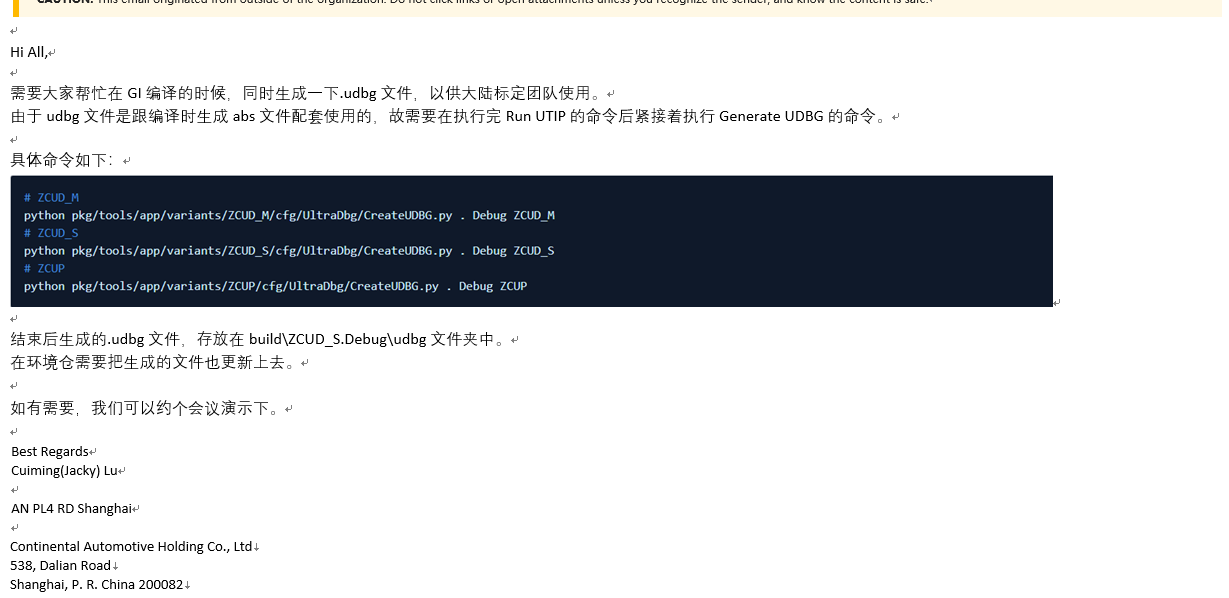


**换完后，直接运行utip，生成s19。将CCM中的串标文件串入S19**

## 十四、生成udbg

python pkg/tools/app/variants/ZCUP/cfg/UltraDbg/CreateUDBG.py . Debug ZCUP

拷贝到bin/udbg文件夹（自己新建）



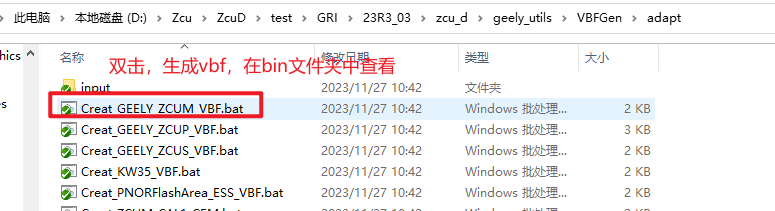
**改零件号，改完零件号，生成VBF**

## 十五、生成VBF

根据零件号表格、大陆输入的VBB地址信息（更新频率低）后，跑脚本生成：





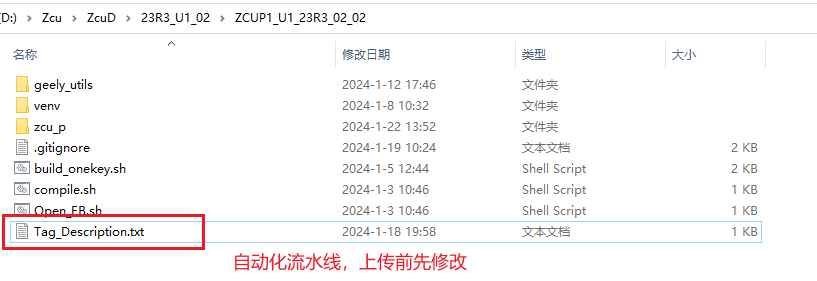


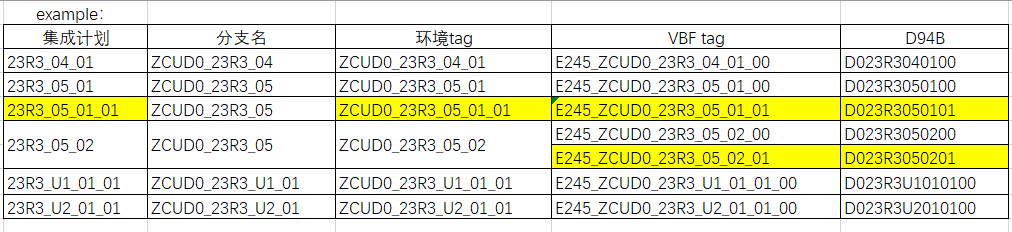
## 十六、填写CheckList、Changelog



## 十七、全编译释放流水线

【释放前记得修改】







[ZCUP1\_CICT [ZCU\_Conti\_BSW\_Compile] [Jenkins]](http://10.24.24.176/job/ZCU_BSW_FullCompile/job/ZCUP1_CICT/)

**附录、相关指令**

rm ./venv -fr  -- 删除venv  
python -m venv ./venv  
source ./venv/Scripts/activate  -- 激活  
pip install metayaml lxml jsonschema  -- 安装  
pip install pywin32    -- 安装  
python pkg/tools/env/build\_template/project\_setup/tresos\_generate.py ZCUD\_M.Debug -- 生成tresos  
python pkg/tools/env/zcu\_builder.py -p ZCUD\_M -b Debug -c true  -- 编译

python pkg/tools/env/build\_template/project\_setup/tresos\_start.py ZCUD\_M.Debug -- 打开EB工具