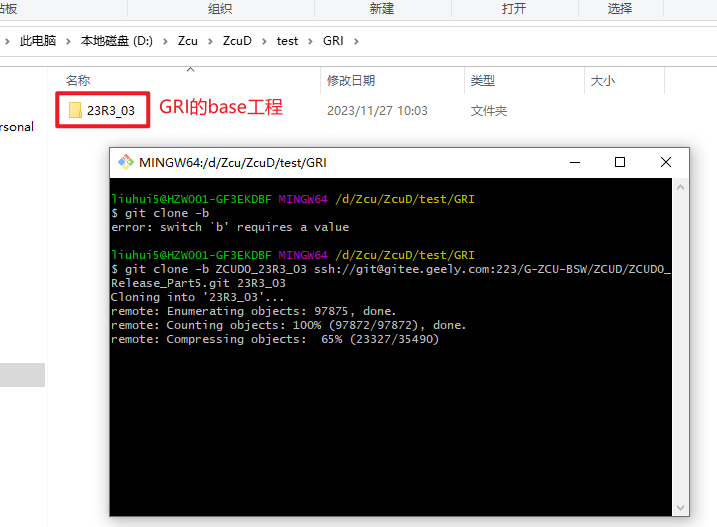
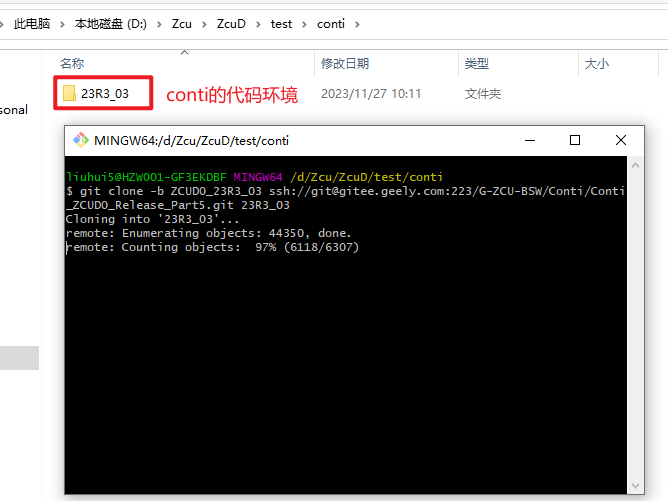
**总集成环境流程手册**

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ZCUDM为例）

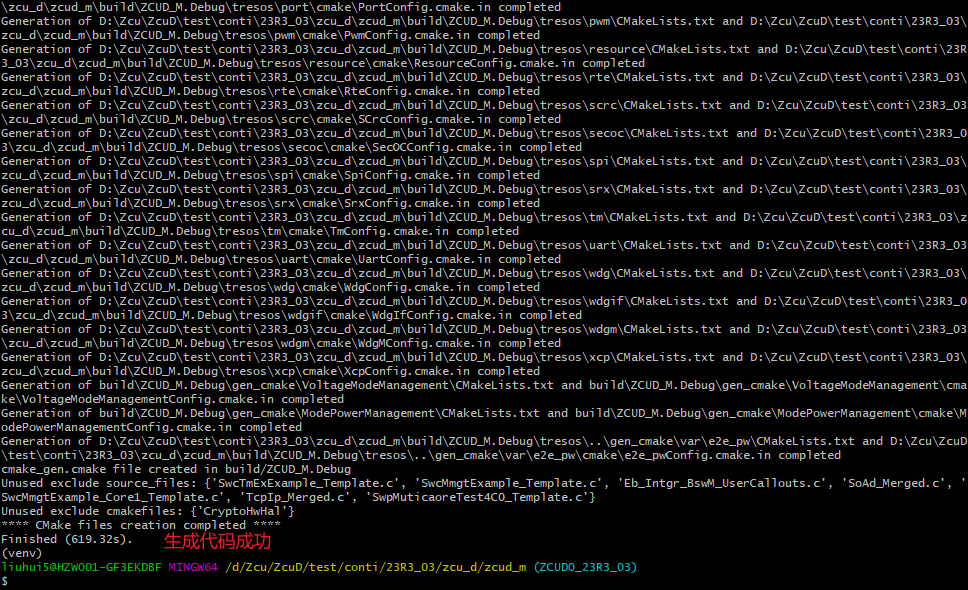
**一、明确要操作的工程对象**

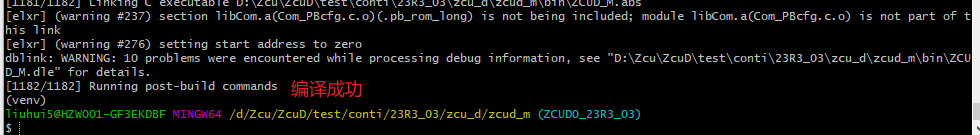
总集成环境主要是将conti的代码工程合并到GRI的工程中，首先需要在本地确认好GRI的工程和conti的工程。





**二、验证conti的工程是否可以生成代码及编译通过**

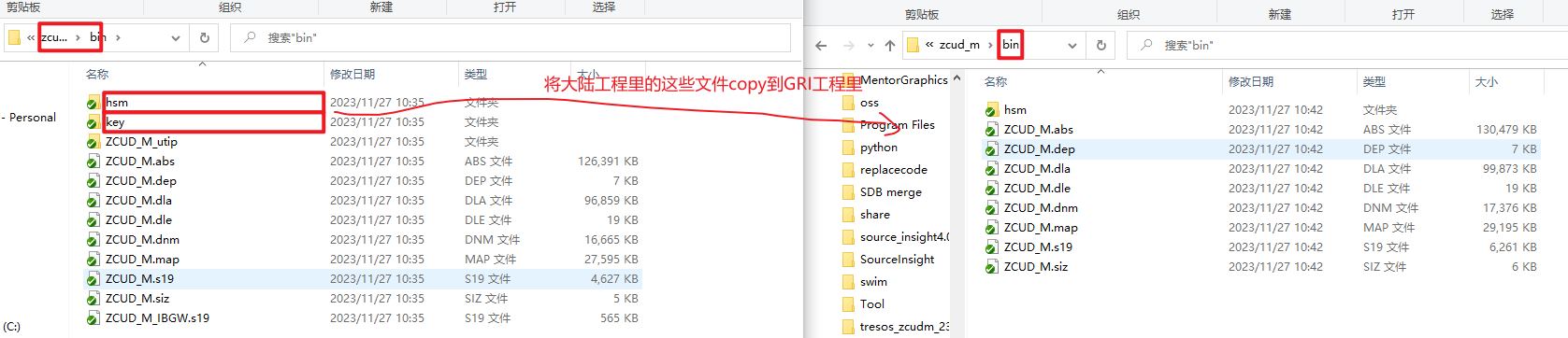




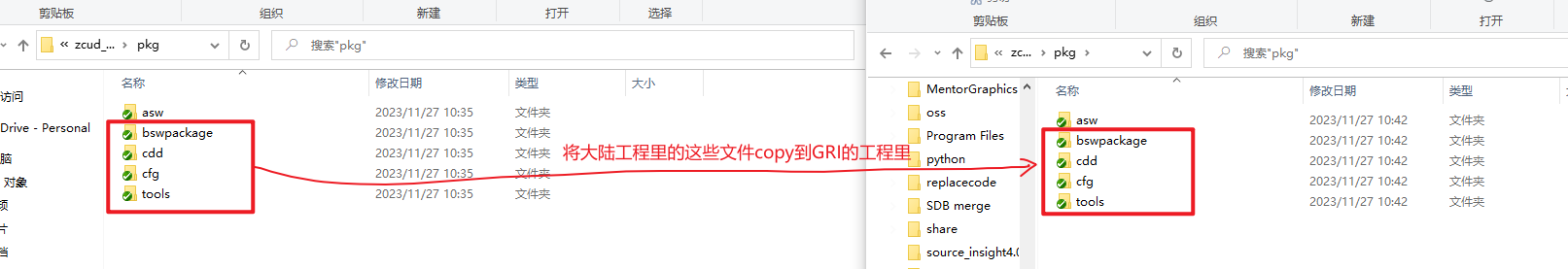
**三、工程替换**

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



（2）、pkg文件夹下

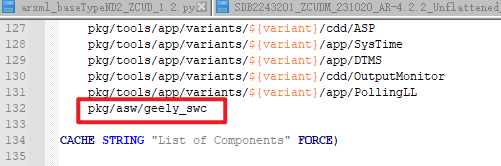


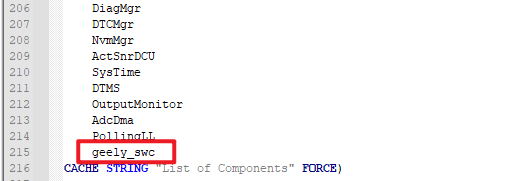
替换完文件后，需要修改cmake\_exclude.yaml和initial.cmake，具体如下：

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。

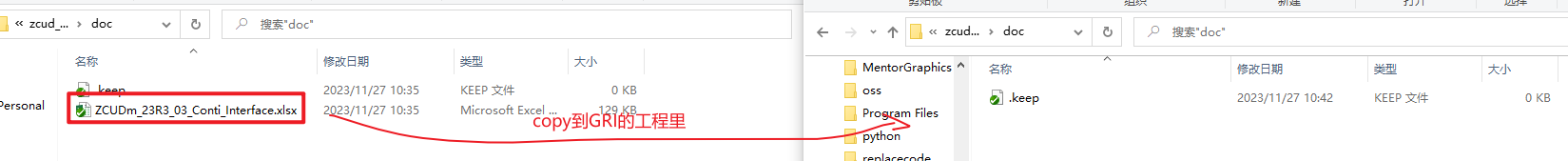


initial.cmake：修改两点，添加GRI的代码

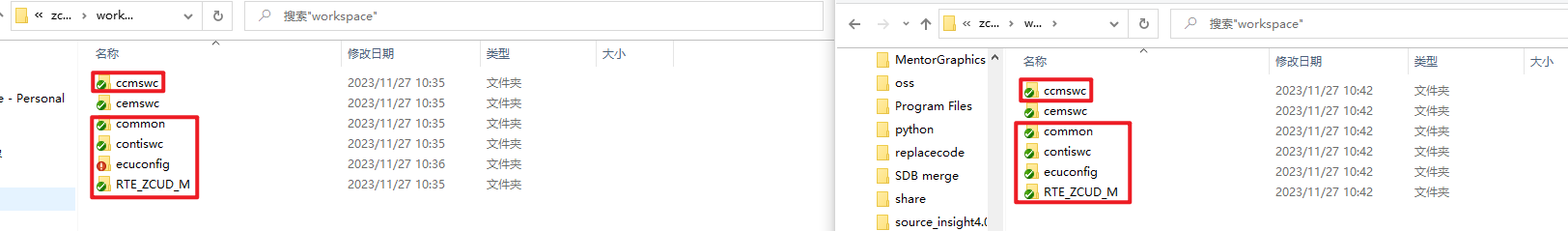




（3）、doc文件夹下

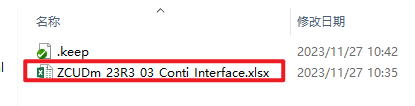


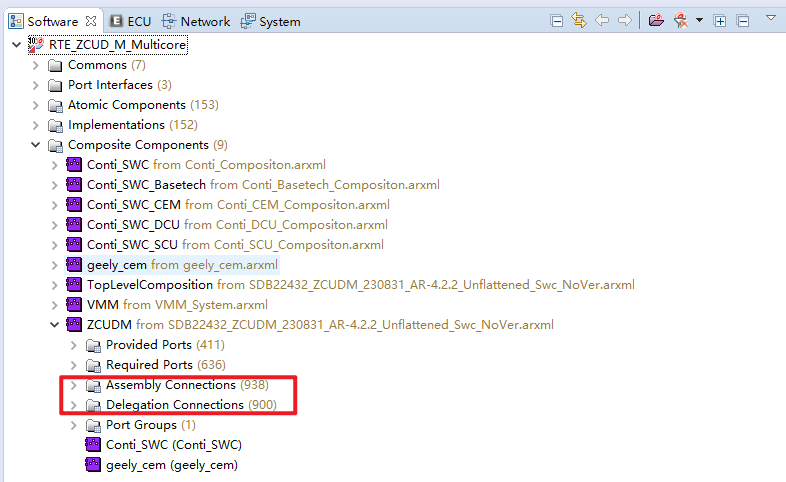
（4）、workspace文件夹下



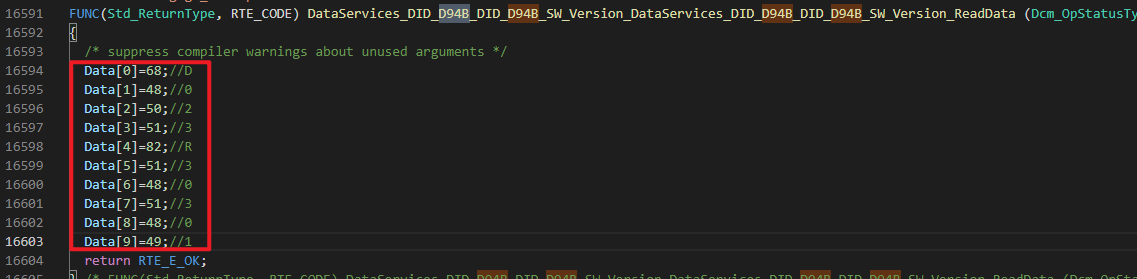
**四、使用AB工具进行连线（GRI&conti、Network）**

根据大陆的接口表信息进行内部连线及网络连线



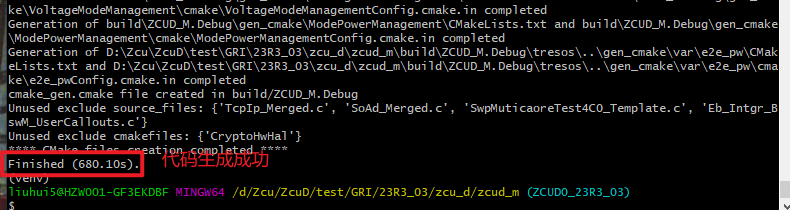


**四、填写DID**



**六、RTE验证总集成环境（可选择指令生成也可EB生成）**

按照相关指令运行：

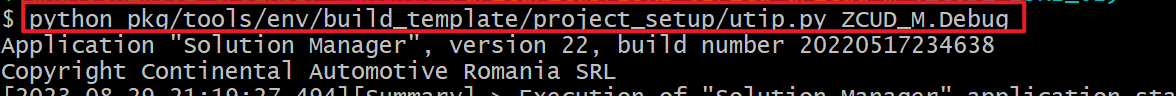


**七、代码编译**

代码编译通过

**八、生成utip**

运行以下指令，生成utp文件

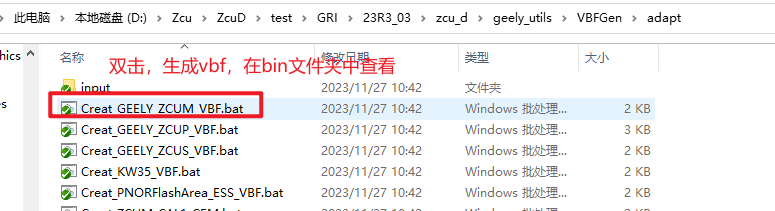


**九、生成VBF**

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、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