**PORT CREATION**

S drive , build the project

Change read only in properties of 10\_configuration

Create p port in swc

Map to runnable of swc

Check

Create R port in swc

Map to runnable of swc

Check

Save

Cfg

We connect p and r

Save

Check c files of swc (to understand the changes after)

In cfg we check the box of obd and comwrap in Contract header settings(screenshot )

Generate in cfg

Now dev

Take comwrap right click and generate ,give path of file

save

Similarly obd2

Save

Change the code in c files vs code

COPY PASTE AND CHANGE THE PORT NAME

compile

**In short**

Port Creation with Interface name given in swc > Connect it to runnable of swc > Check

Do it for P and R port

SAVE in DEV

Open CFG, Connect both port using CFG

SAVE in CFG

Generate in CFG

Open Dev

Component implementation by giving path for 1st Swc files

Save dev

Component implementation by giving path for 2nd Swc files

Save dev

You are free to close both cfg and dev

Now open corresponding changes C files and implement the function logic with new created ports ,Save

Template can be generated even from configurator rather than doing in developer by giving path of c file. It can be done by using settings and generate.

**Testing using CANoe and Winidea**

Dump the code using winidea and canoe

Add the signal drvcyccnd into the analysis window , change its values from inactive to active to inactive

Observe the change in PSU counter value for that (go to diagnostic console type in the ID 22 03 1D and execute )

**XCP in CANoe**

To measure p and r port data inbetween SWC

Add A2l file (20 make bin ) into the canoe to see the port communication signals and analyse