



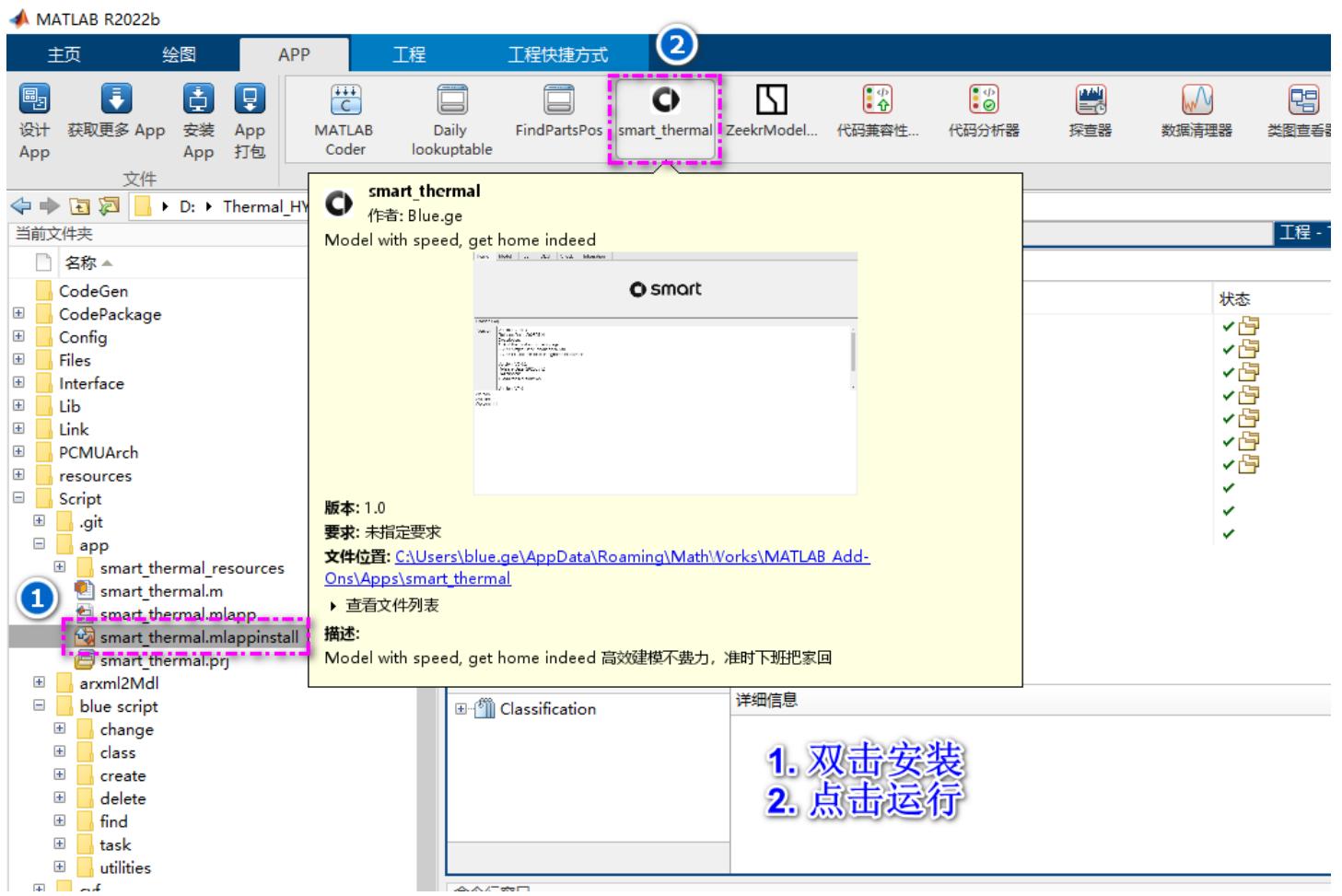
Smart 脚本使用说明

版本信息	更新日志	日期	修改者	备注
V1.0	初步完成了大部分脚本的更新说明	2025.4.22	Blue.ge	
V1.1	更新了初版的上位机界面说明	2025.5.15	Blue.ge	
V1.2	更新了上位机建模界面说明	2025.6.10	Blue.ge	Sldd, 测试, 集成等 GUI界面 操作说明 待完善

上位机使用说明

安装

双击 `smart_thermal.mlappinstall` 进行安装



界面简介

首页

主要包括变更记录

MATLAB App

File Help

功能模块

smart

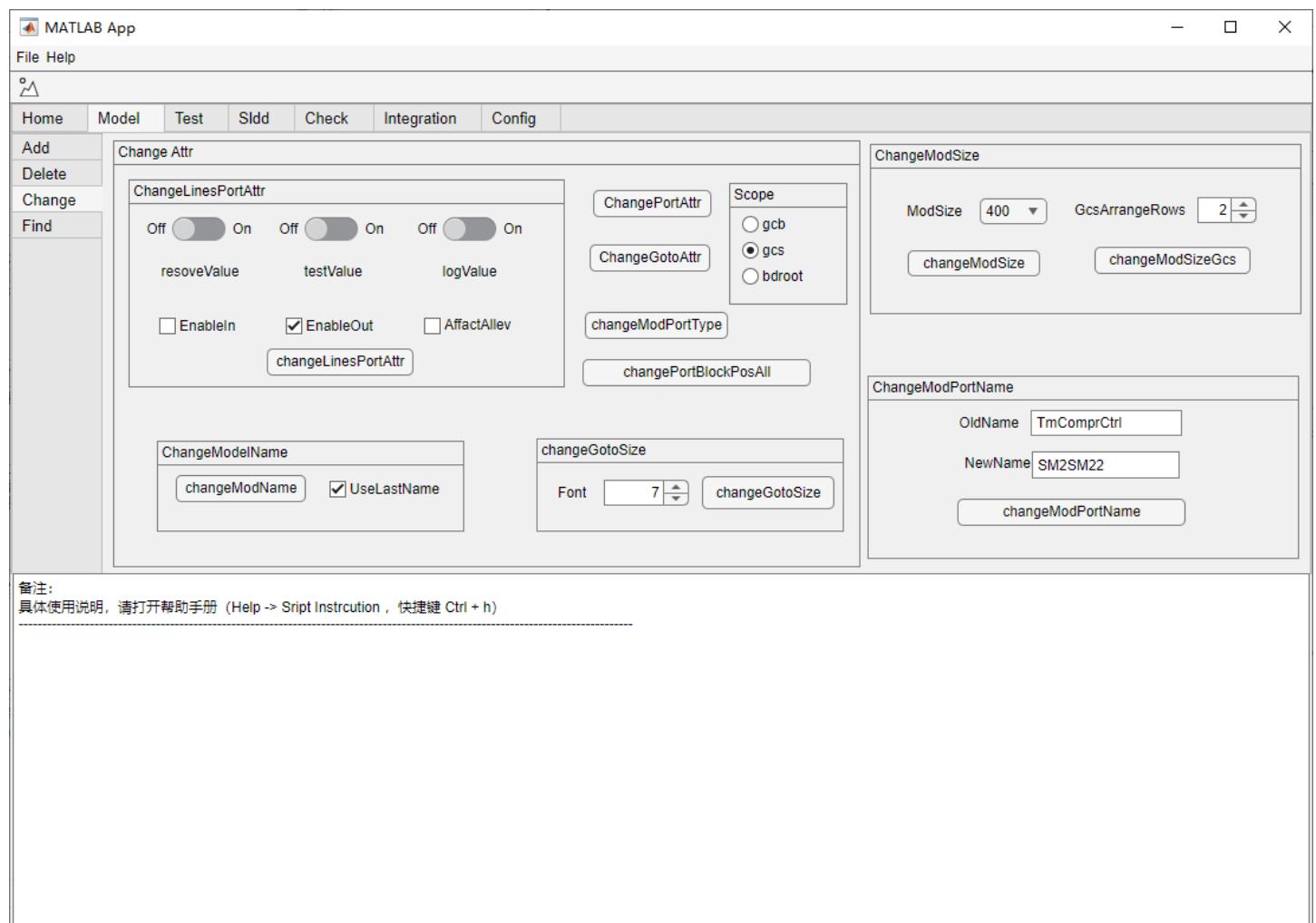
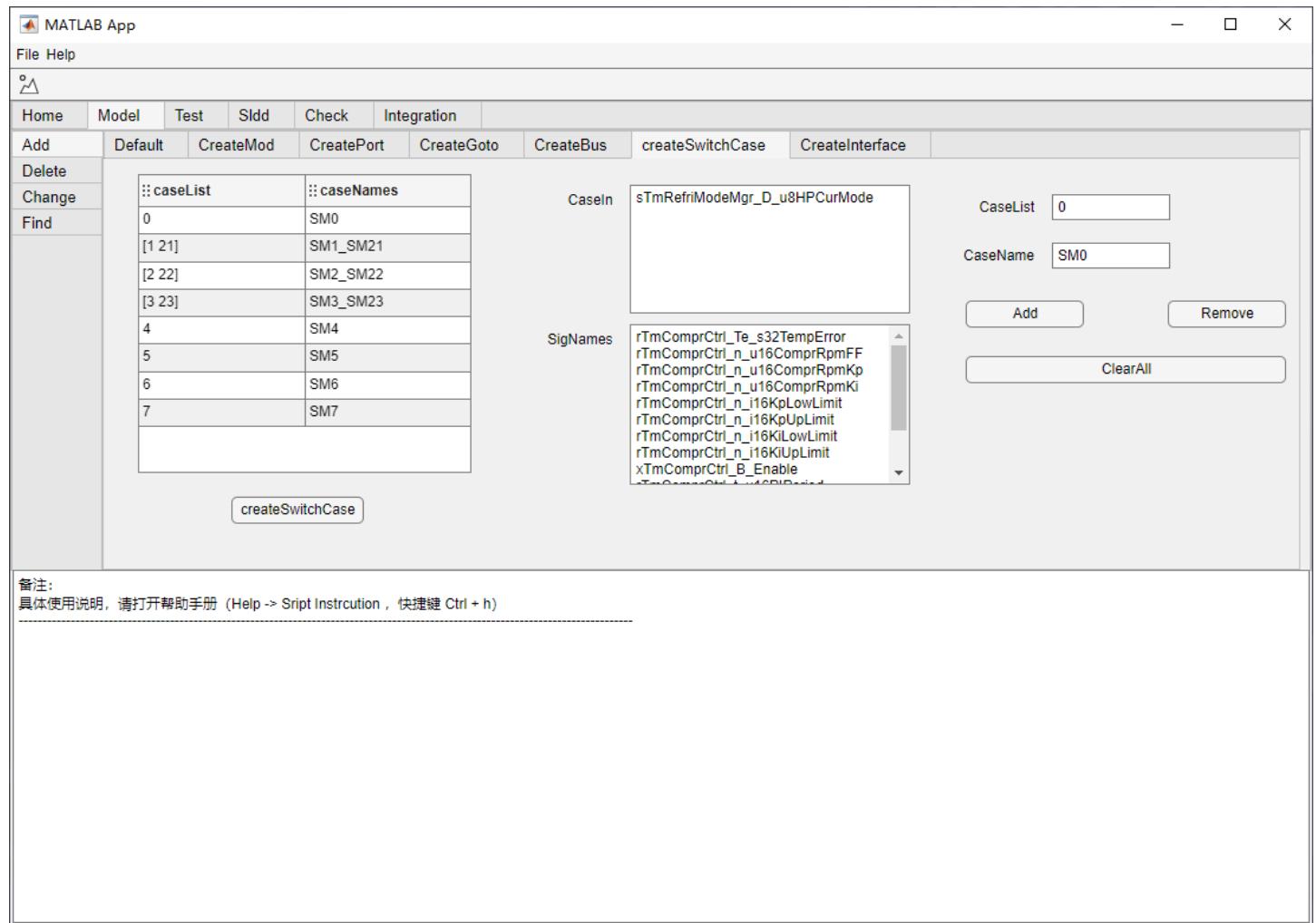
Change Log

Version	Version: V0.1.0 alpha Release Date: 20250609 Description: 1. Add XCU relative working folw, including find the XCU stroage class based on the name, export and import the XCU sldd 2. add the function of block attr change, which include goto from and ports, all the attr have been saved to the config file 3. add the funtion of change model name
	Version: V0.0.7 Release Date: 20250604 Description: 1. add function of createPortsBySheet , so it wil be much easier to create ports by calling this function 2. upgrade createPortsGotoUpdate, which could resolve the output signal 3. fix the bug: createPortsGotoUpdate now created port position error

Welcome !
Welcome !!
Welcome !!!

建模页

这页包括了建模可能会用到的增删改查相关脚本



测试页

高效的进行模型的单元测试

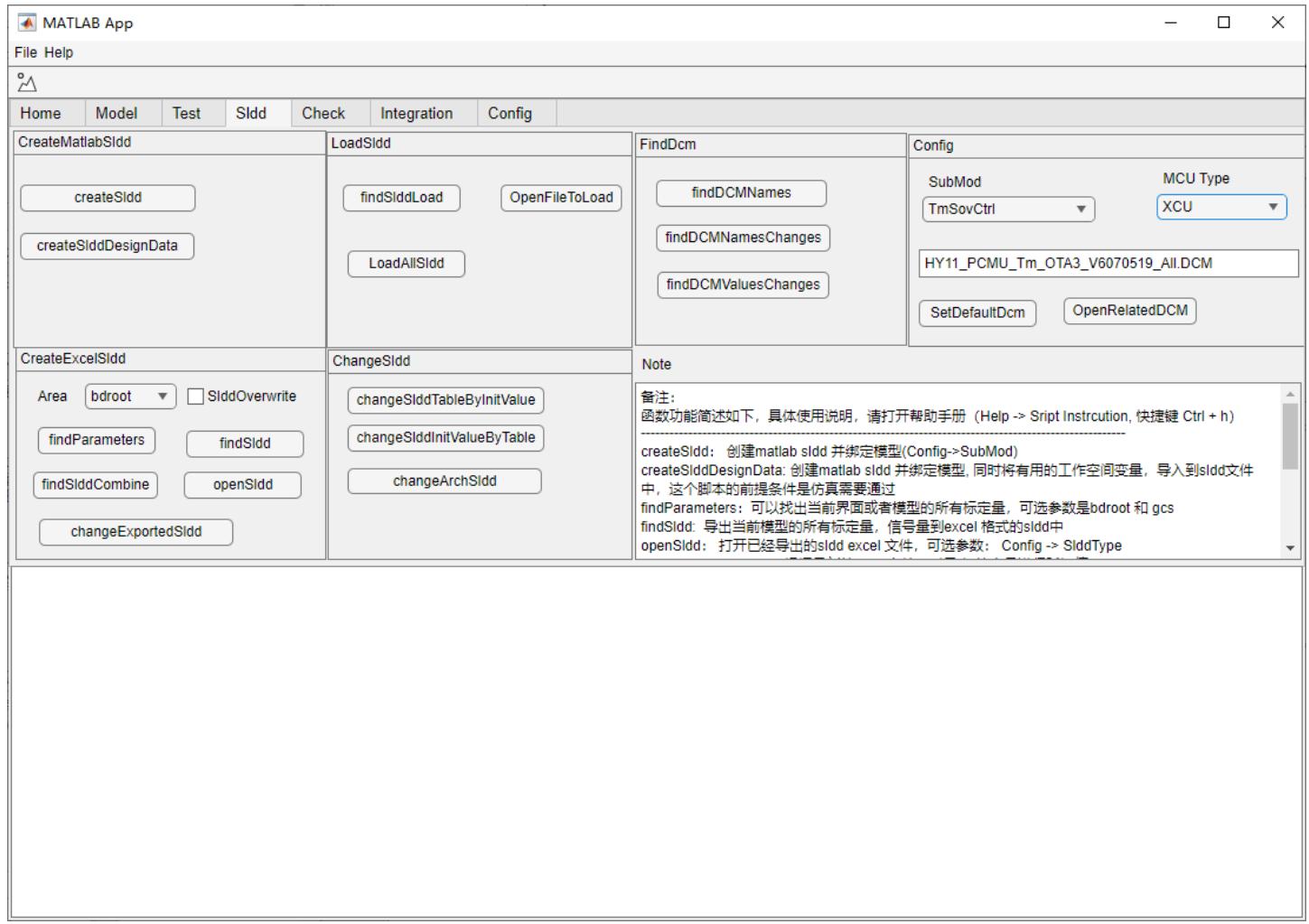
The screenshot shows the MATLAB App interface with the 'Test' tab selected. On the left, there's a 'CaseTree' sidebar with categories like NHP, HP, and Harness History Params. The main area displays a 'TestCase' section with 'LEV1' and 'LEV2' listed. Below this are buttons for 'AddCase', 'DelCase', 'GetPorts', 'createHarness', 'Demo', and 'DelParam'. A large table below shows data for different categories (NHP, HP) across various parameters (Lev1, Lev2, sTmDiag_D_u8CmprDischrgTFItSt, etc.).

备注：
函数功能简述如下，具体使用说明，请打开帮助手册 (Help -> Script Instruction，快捷键 Ctrl + h)

AddCase: 将TestCase中的内容添加到根目录CaseTree中，作为当前harness的测试大类，如果需要添加子类，需要先选中大类，再点此按键
DelCase: 将CaseTree中的测试用例删除
GetPorts: 点击子模型，生成对应的输入输出信号，同时根据CaseTree，生成测试用例个数，然后就可以对其进行填充
createHarness: 根据测试用例，生成harness
Demo: 快速生成demo CaseTree, 需要打开并点击模型: ScriptDemo/createHarness/Demo
DelParam: 删除Harness Histroy Params 选中的参数
在表格上单击右键，可以快速删除全部数据或者选中数据

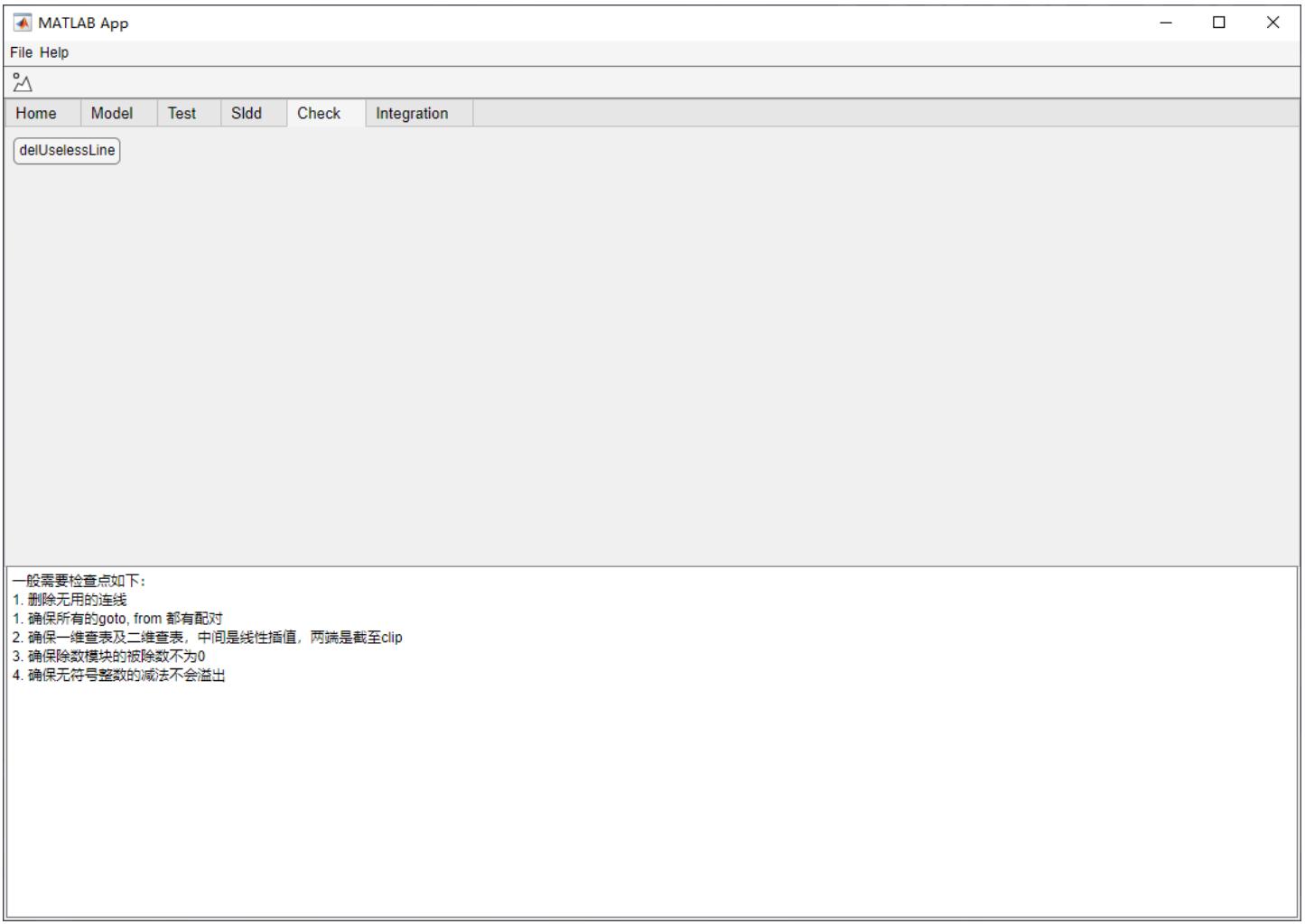
Sldd 页面

主要用于导入，导出，管理信号量和标定量，及DCM相关文件的操作



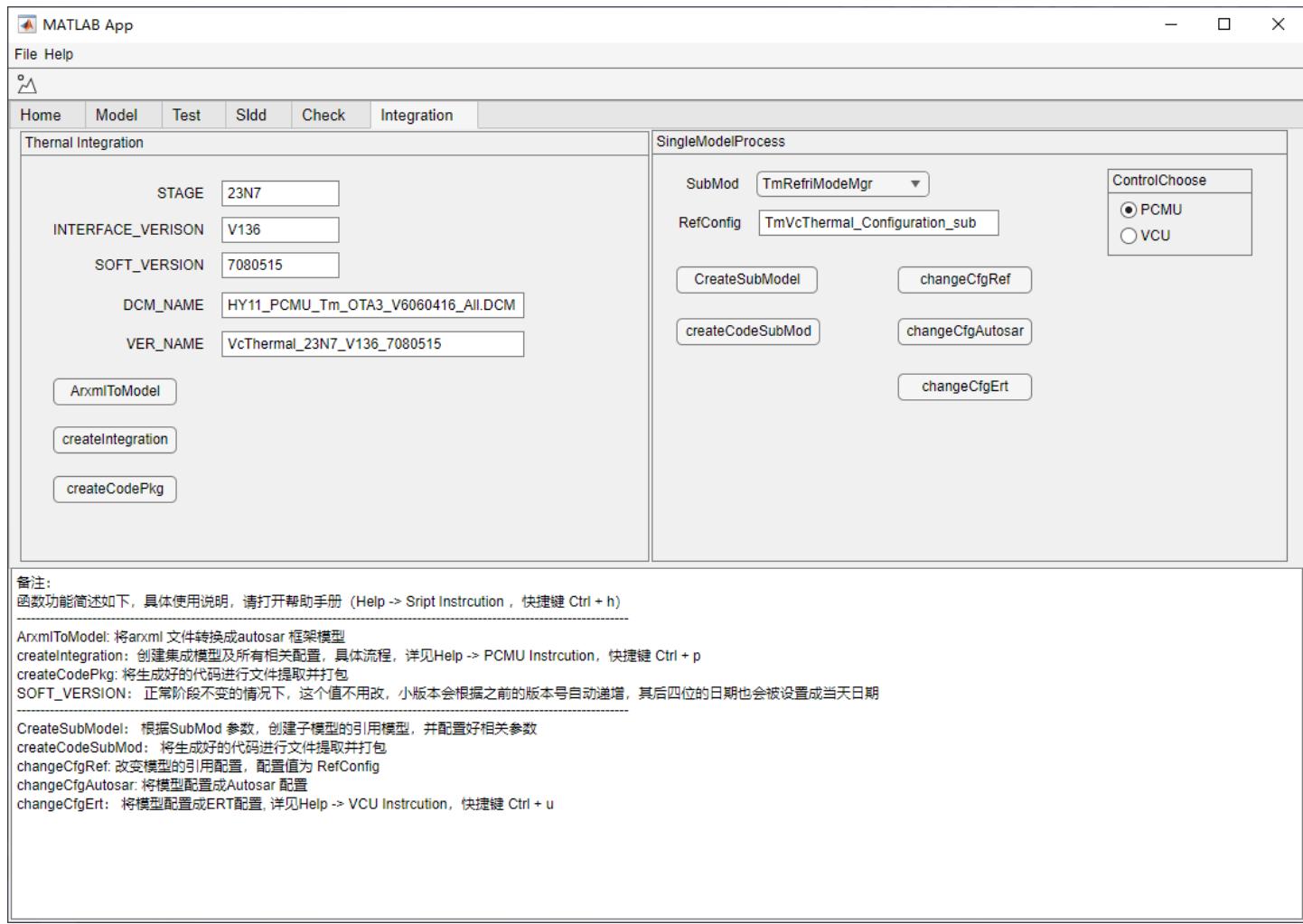
Check 页

主要检查建模中遇到的常见错误，具体功能完善中



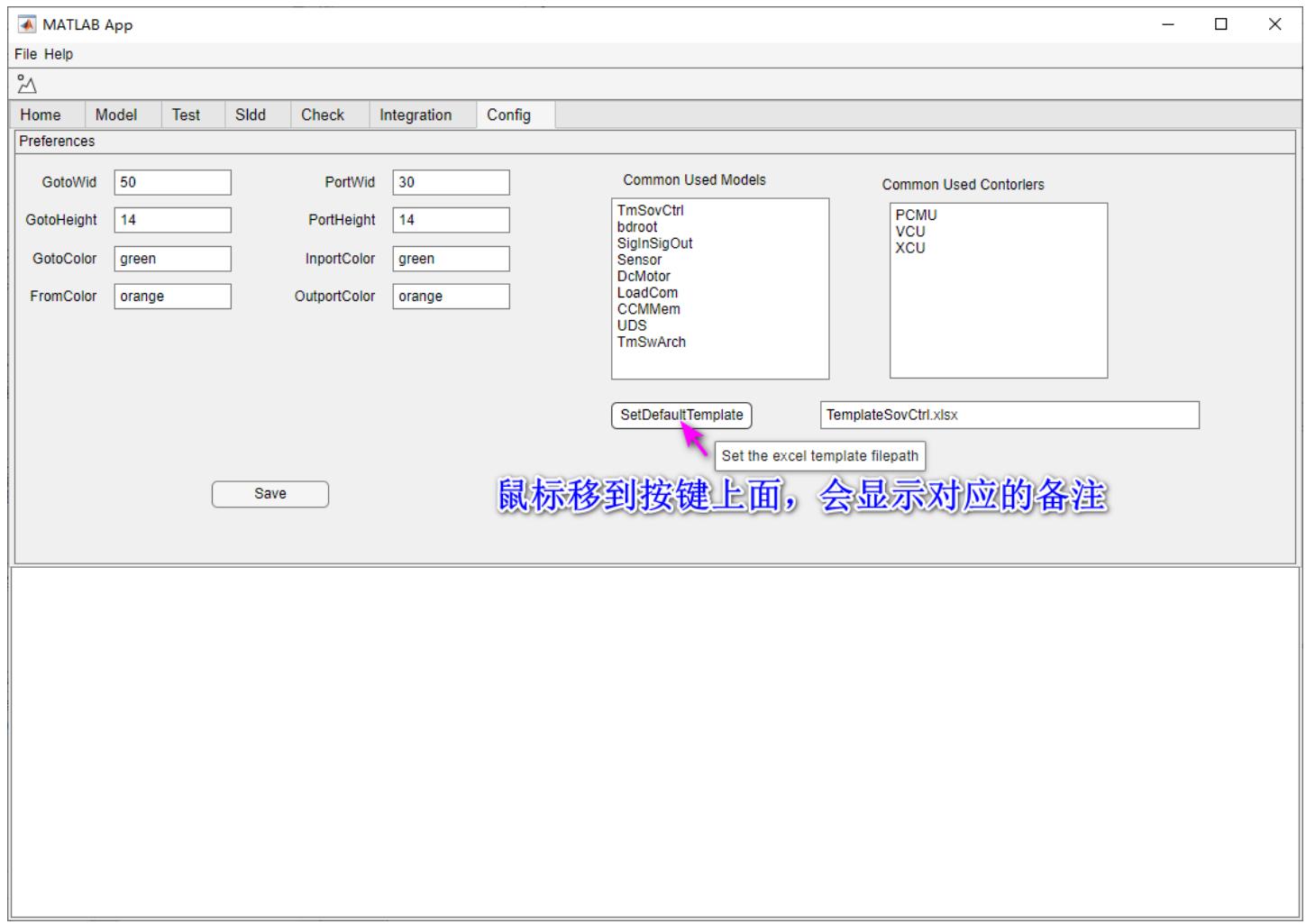
集成页

集成过程中会用到的相关脚本，实现快速集成



配置界面

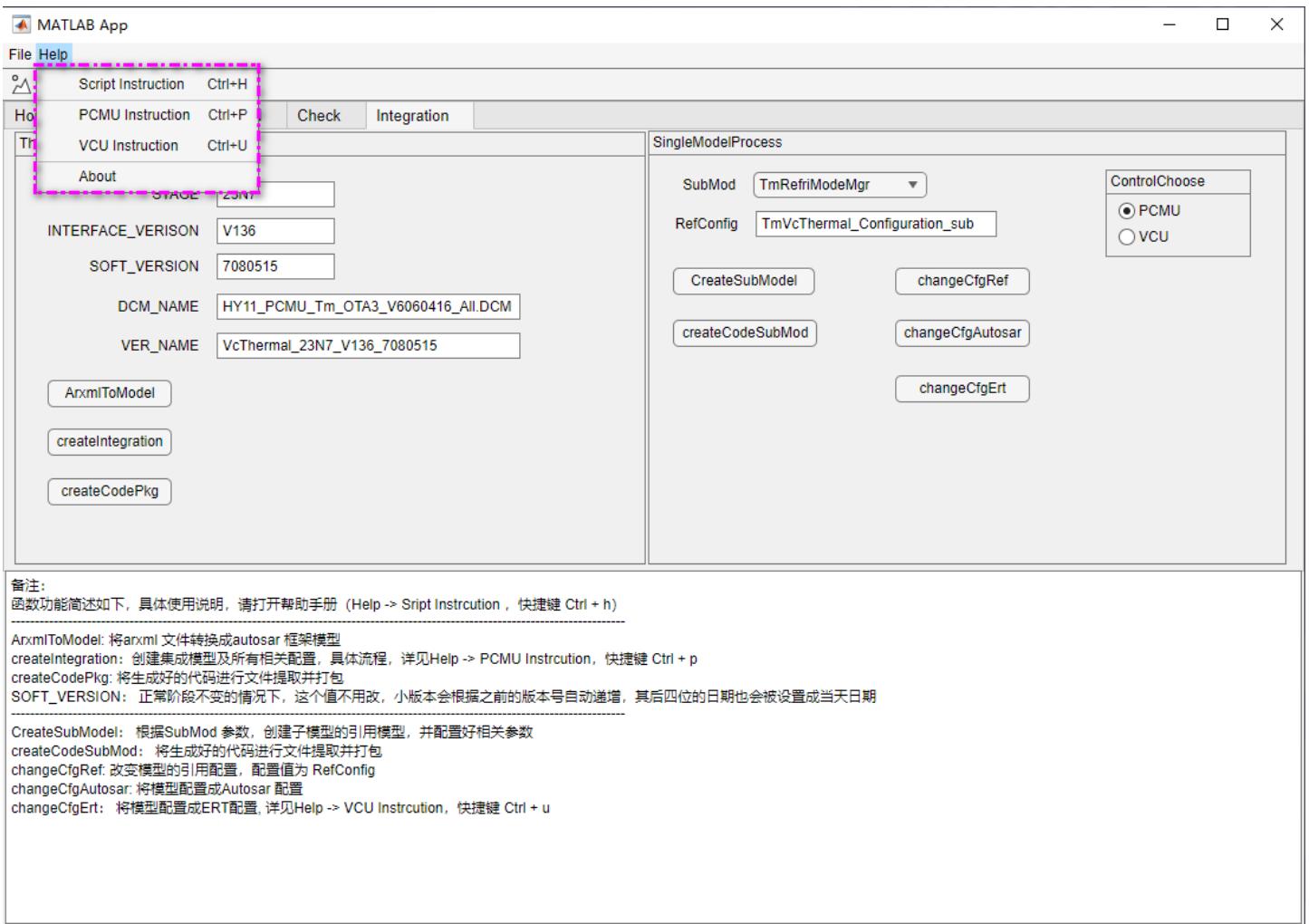
可以进行偏好设置，设置好后，下次打开后也会保存



菜单栏

可以打开详细的帮助文件，目前涉及3个文件：

- 脚本使用说明
- PCMU集成说明
- VCU集成说明



备注：
函数功能简述如下，具体使用说明，请打开帮助手册 (Help -> Script Instruction，快捷键 Ctrl + h)

ArxmlToModel: 将arxml 文件转换成autosar 框架模型

createIntegration: 创建集成模型及所有相关配置，具体流程，详见Help -> PCMU Instruction，快捷键 Ctrl + p

createCodePkg: 将生成好的代码进行文件提取并打包

SOFT_VERSION: 正常阶段不变的情况下，这个值不用改，小版本会根据之前的版本号自动递增，其后四位的日期也会被设置成当天日期

CreateSubModel: 根据SubMod 参数，创建子模型的引用模型，并配置好相关参数

createCodeSubMod: 将生成好的代码进行文件提取并打包

changeCfgRef: 改变模型的引用配置，配置值为 RefConfig

changeCfgAutosar: 将模型配置成Autosar 配置

changeCfgErt: 将模型配置成ERT配置，详见Help -> VCU Instruction，快捷键 Ctrl + u

UI 使用说明

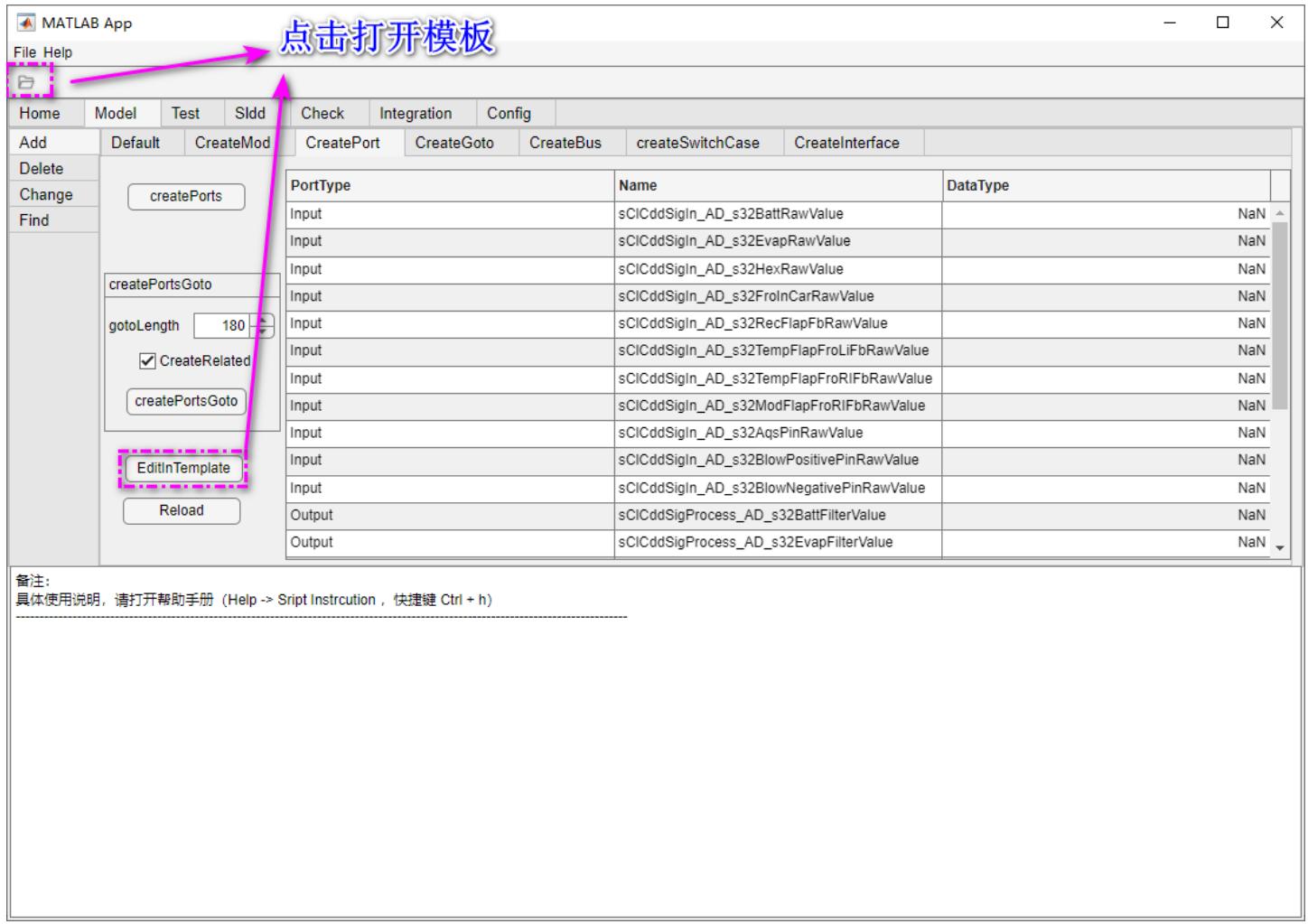
建模篇

建模方式

建模方式一般有两种，

- 从外向内建模，适合一些接口模型，清楚的知道输入输出信号
- 从内向外建模，适用于大部分模型

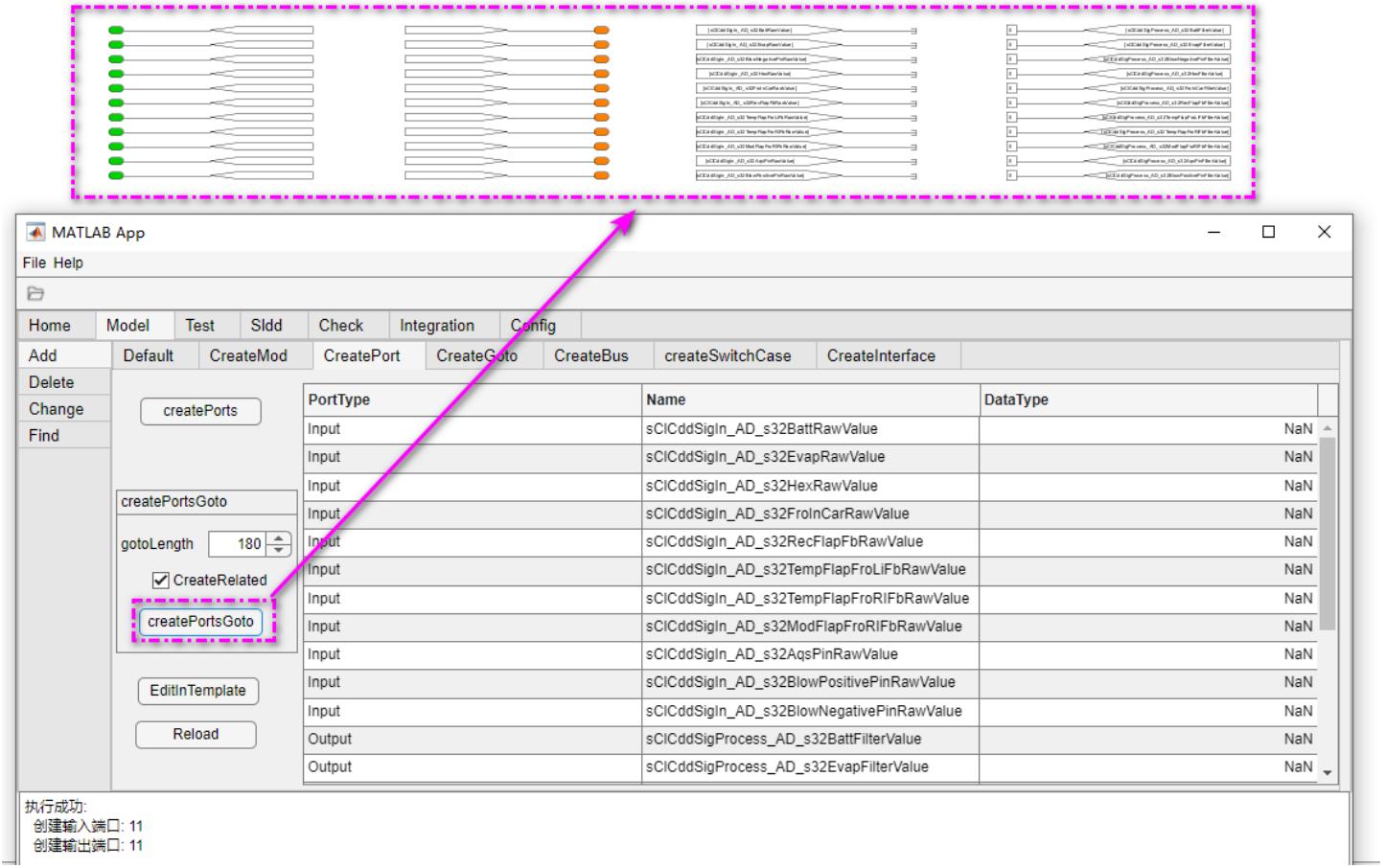
从外向内建模



在Signals页编辑信号

ModelName	PortType	Name	Dimensions	Details	ValueTable	Unit	IniValue	Min	Max
SigProcess	Input	sCICddSigIn_AD_s32BattRawValue		低压电池AD值		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32EvapRawValue		空调箱蒸发器原始AD值		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32HexRawValue		空调箱热交换器原始AD值		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32FroInCarRawValue		前排车内温度传感器AD值		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32RecFlapFbRawValue		内循环风门电机位置反馈		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32TempFlapFroLifbRawValue		前左风门电机位置反馈		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32TempFlapFroRifbRawValue		前右风门电机位置反馈		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32ModFlapFroRifbRawValue		模式风门电机位置反馈		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32AqsPinRawValue		AQSPWM驱动芯片诊断PIN脚AD值		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32BlowPositivePinRawValue		鼓风机调速电机正极反馈		Kg	0	2048	
SigProcess	Input	sCICddSigIn_AD_s32BlowNegativePinRawValue		鼓风机调速电机负极反馈		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32BattFilterValue		低压电池滤波后AD值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32EvapFilterValue		空调箱蒸发器滤波后AD值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32HexFilterValue		空调箱热交换器滤波后AD值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32FroInCarFilterValue		前排车内温度传感器滤波后AD值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32RecFlapFbFilterValue		内循环风门电机位置反馈滤波值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32TempFlapFroLifbFilterValue		前左风门电机位置反馈滤波值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32TempFlapFroRifbFilterValue		前右风门电机位置反馈滤波值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32ModFlapFroRifbFilterValue		模式风门电机位置反馈滤波值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32AqsPinFilterValue		AQSPWM驱动芯片诊断PIN脚滤波值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32BlowPositivePinFilterValue		鼓风机调速电机正极反馈滤波值		Kg	0	2048	
SigProcess	Output	sCICddSigProcess_AD_s32BlowNegativePinFilterValue		鼓风机调速电机负极反馈滤波值		Kg	0	2048	

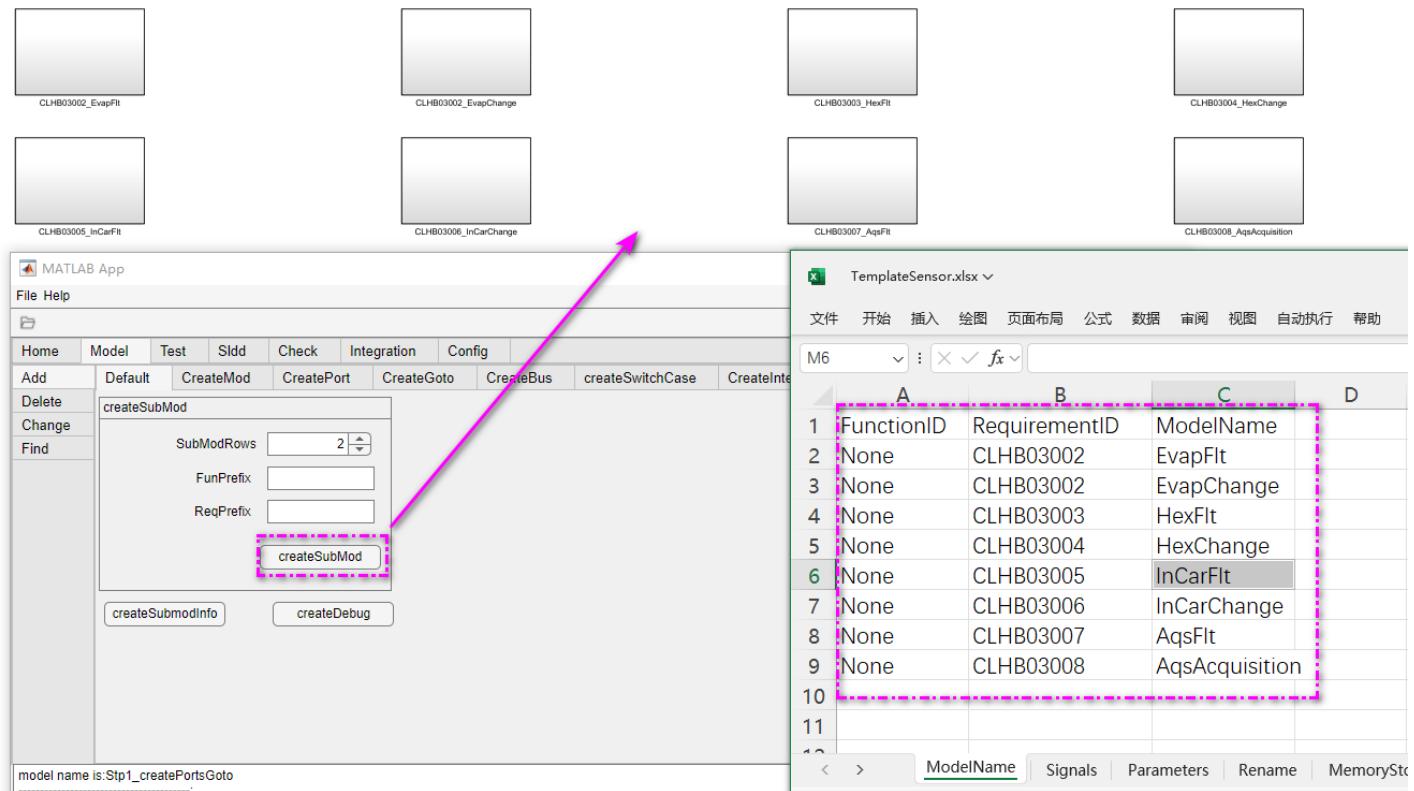
执行脚本



接下来重点介绍下从内到外建模，也是主要的工作方式

根据需求创建子模型

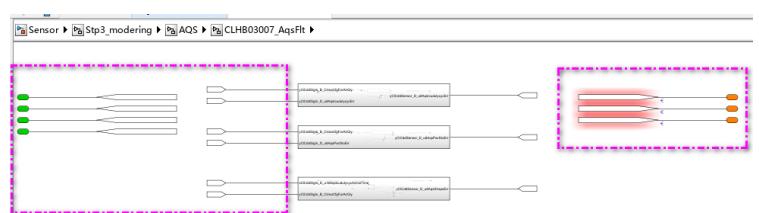
- 将需求清单罗列到模板中，进行子模型批量创建，尽可能一个需求，对应一个子模型，便于更好的测试
- 可以指定软件需求前缀，功能需求前缀，默认为空
- 可以指定初始行数



搭建需求单元子模型

搭建好模型后，执行如下脚本，效果如下：

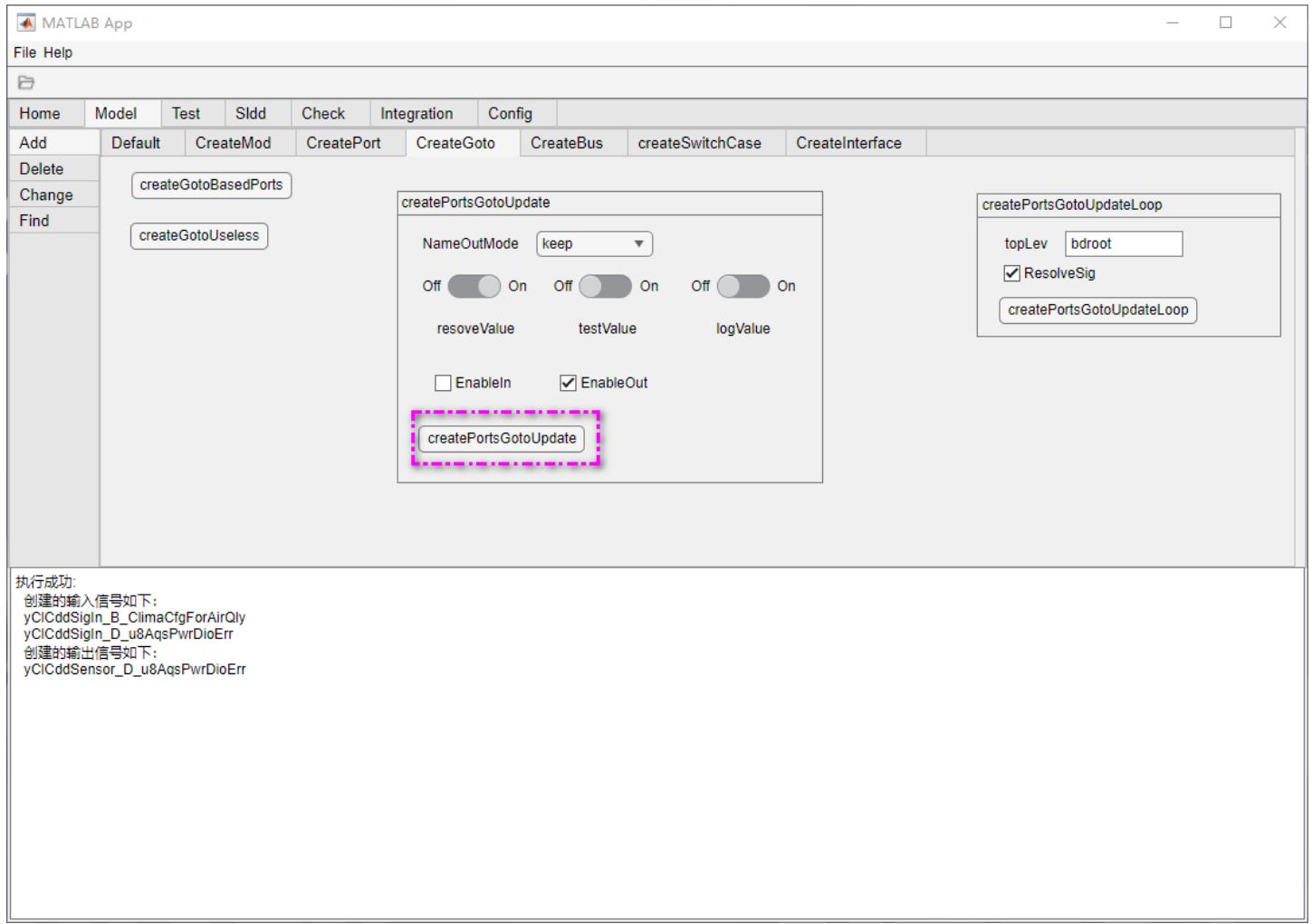
- 为没有接线的模型创建goto, from
- 在当前模型创建匹配的输入，输出port. 同时设定数据类型，颜色等属性
- 可选择是否对输出信号进行解析，或者记录
- 根据端口数量，自动调整模型尺寸
- 在父级目录，创建goto, from



如果父级不是顶层模型，外部也会创建goto fro,

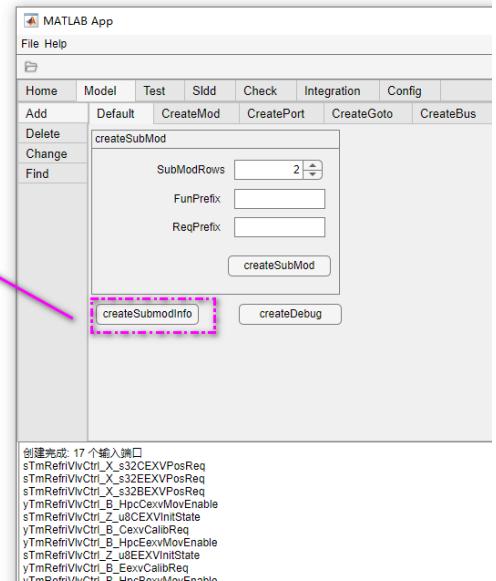
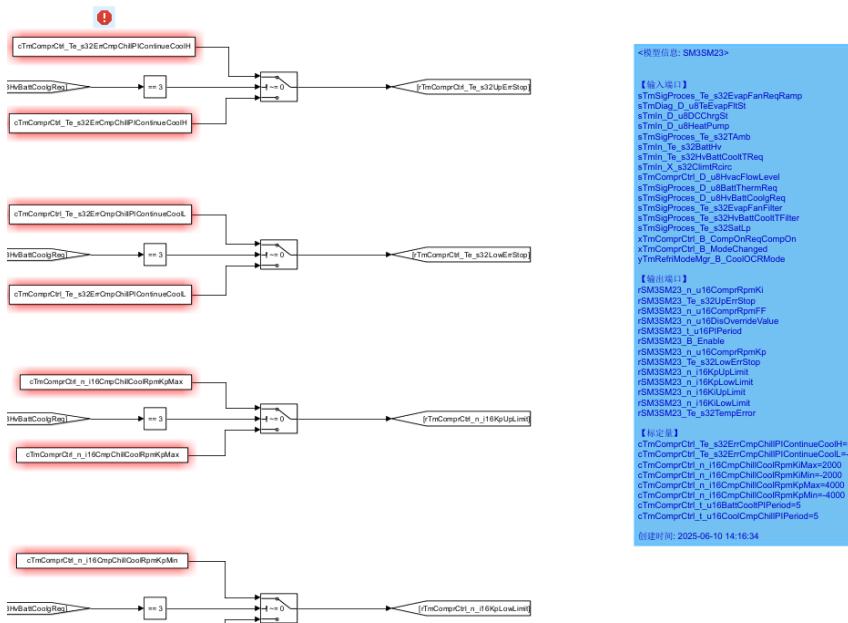


执行如下脚本



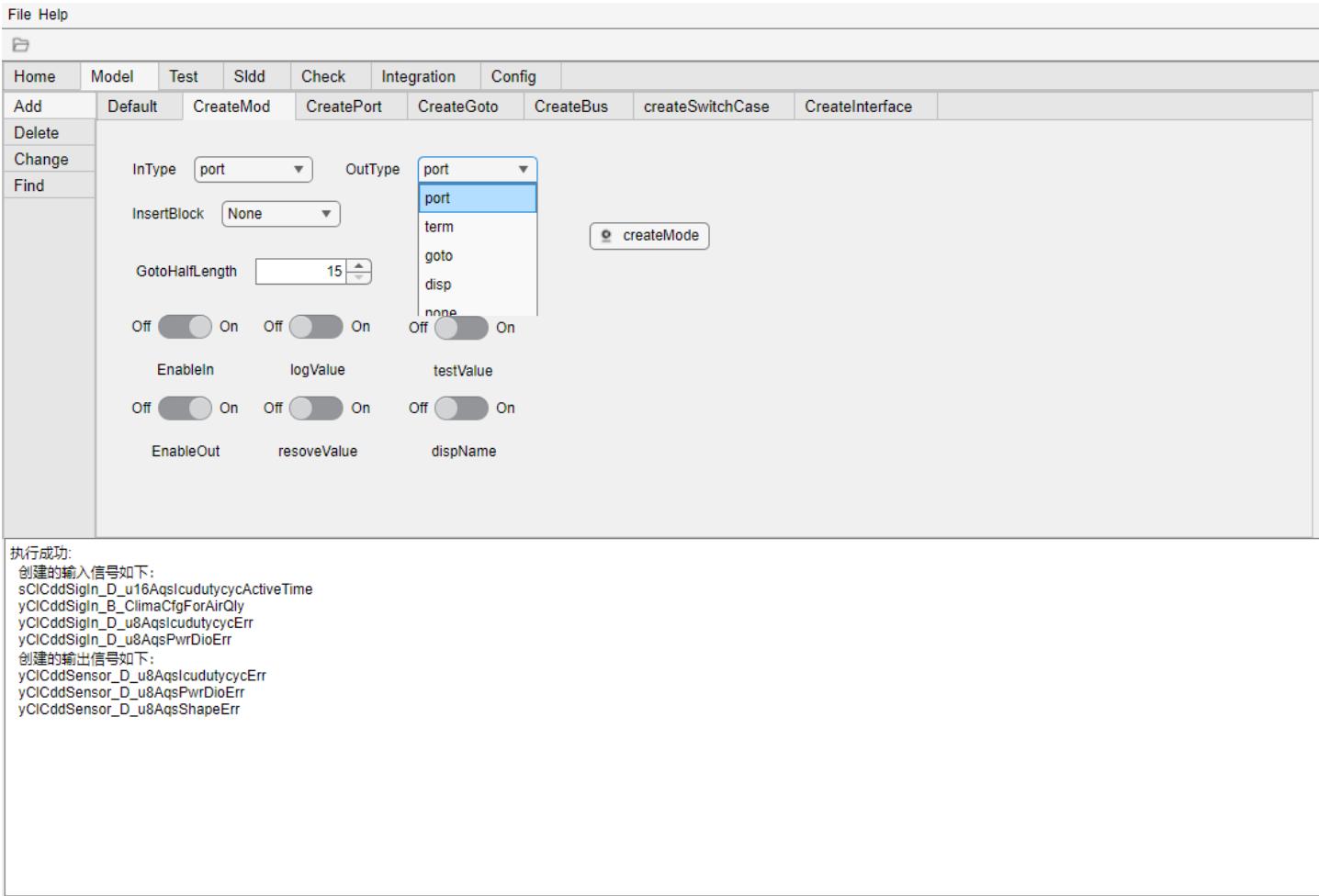
创建模型信息

- 创建输入，输出，及对应的标定量
- 如果非表格类的标定量，会根据DCM更新其值，避免初始值跟标定值不一致造成的混淆



创建跟模型相关的模块

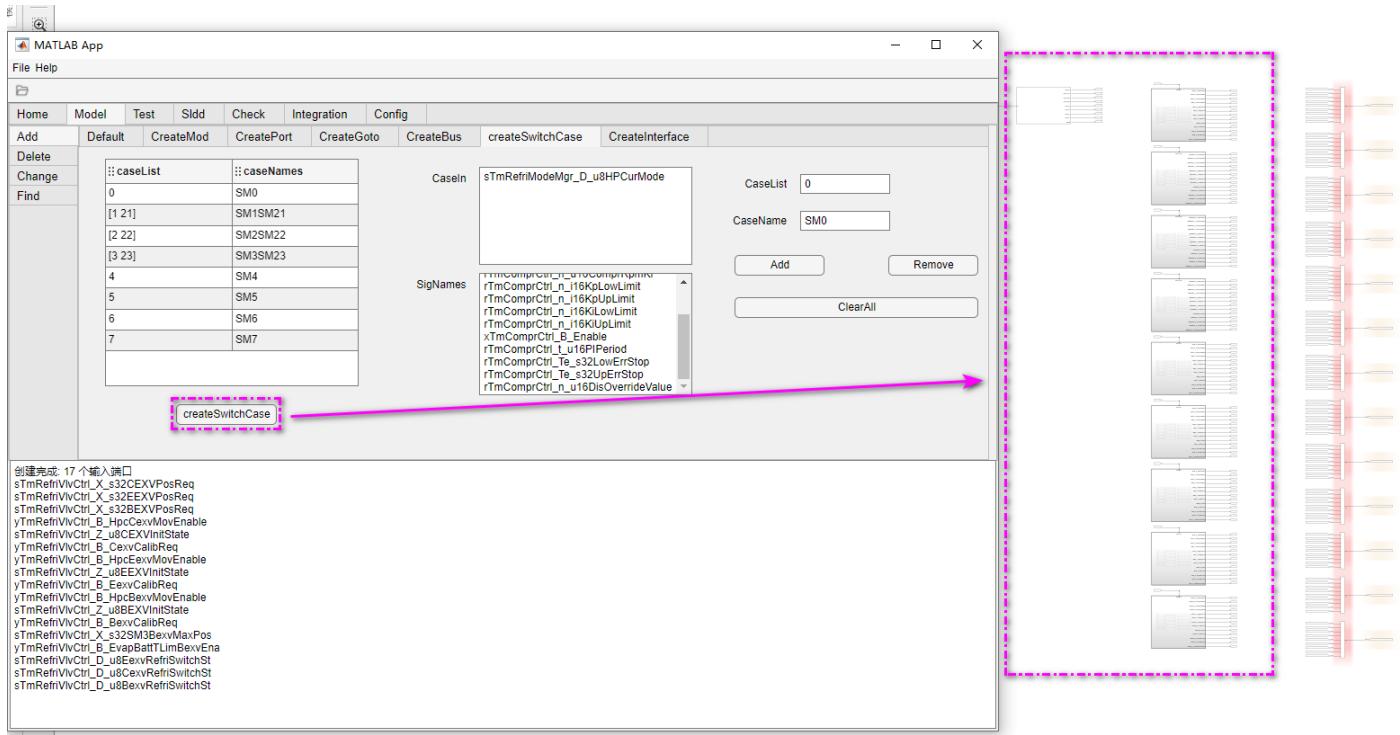
- 可分别选择模型输入输出的创建类型
- 选择是否解析信号
- 如果创建常量，会同步设置其数据类型，便于仿真



效果如下:



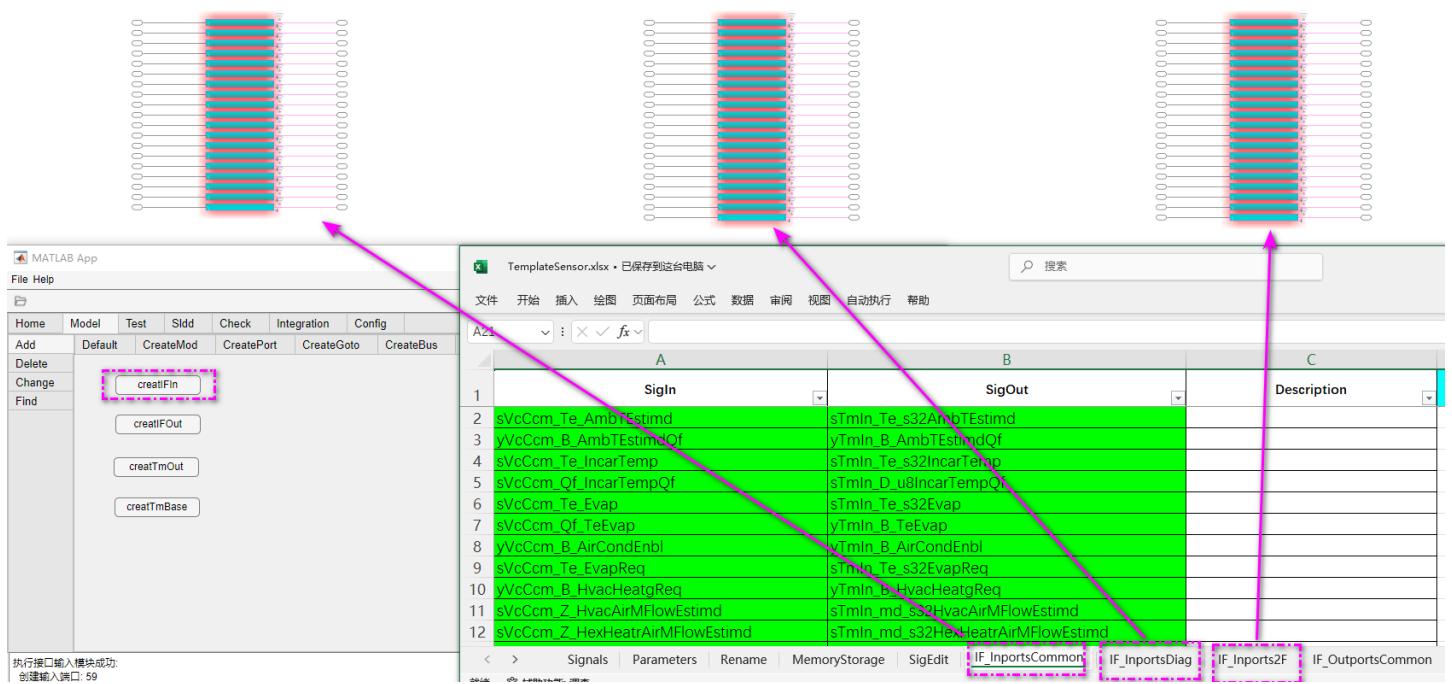
创建switchCase



创建Interface

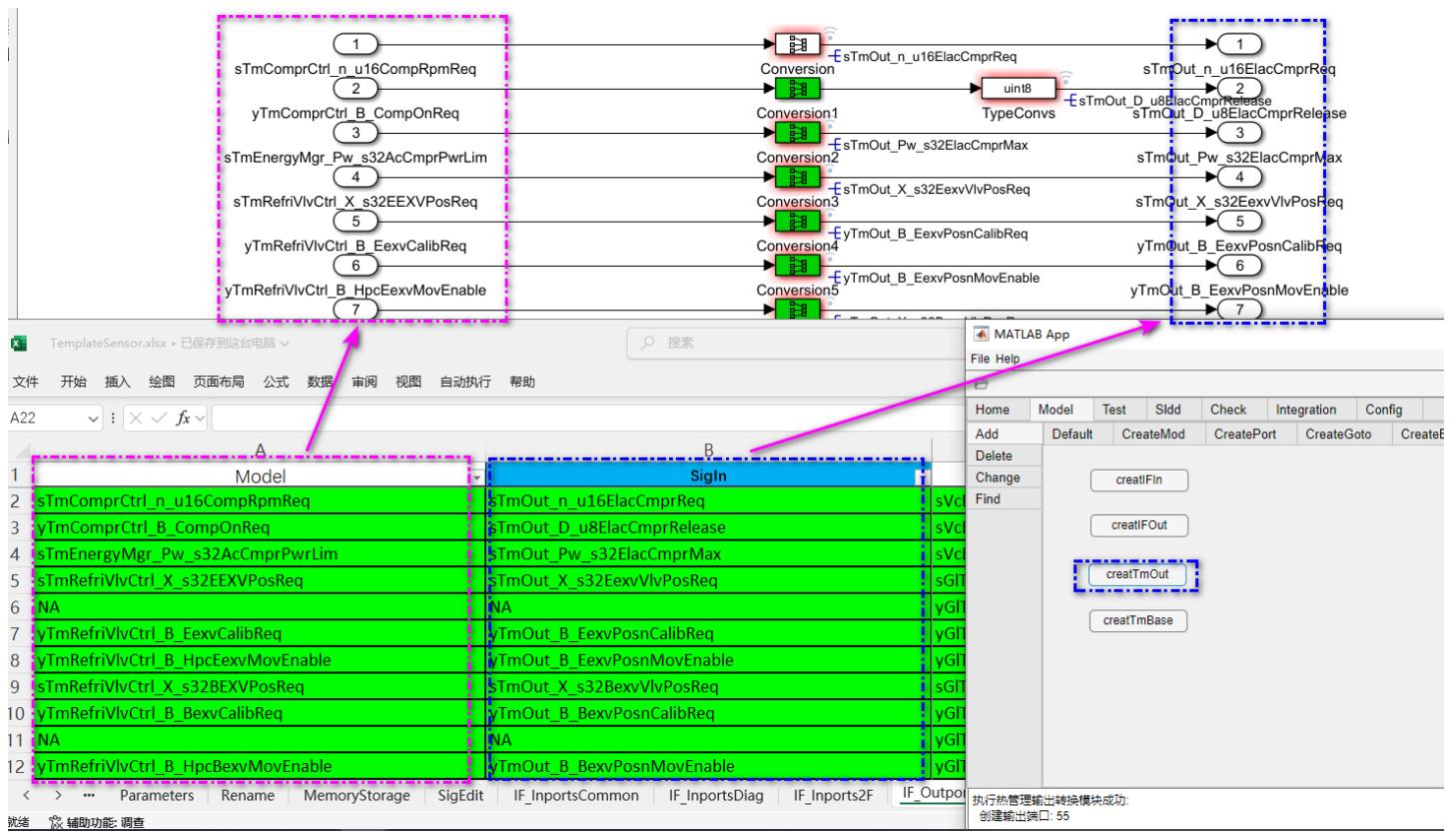
创建接口信号

- 根据输入输出信号，创建输入模块，中间用Debug 隔离
- 识别autosar 接口别名的数据类型，并将其设定到输入信号中
- 如果无法识别输入信号，会添加信号转换模块，使其跟内部信号匹配
- 输入输出类似，这里不再赘述



创建TmOut 架构信号

- 由于内外部信号都解析了，需要进行隔离



创建连线

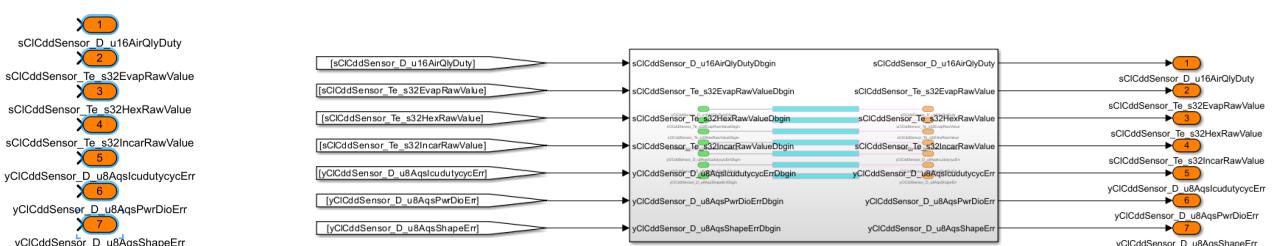
内部模型搭建好后，需要逐层向上连接，如果层次比较深，那工作量巨大

或者模型搭建好后，由于新增了个需求，又得逐层增加信号

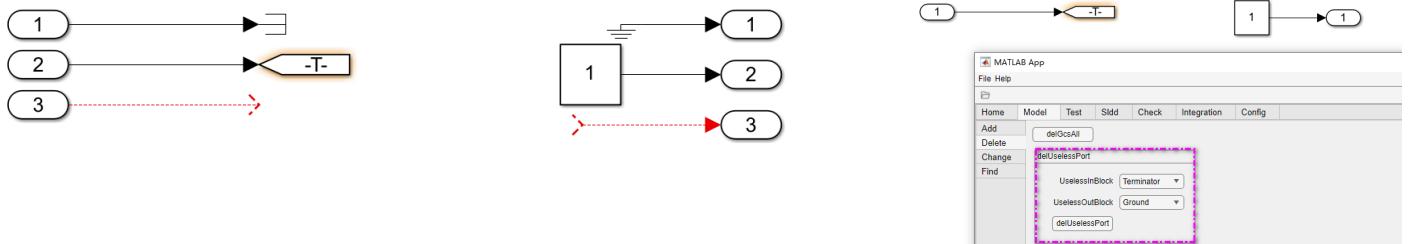
- 递归从内向外创建goto, 端口并设置相关属性，
- 直到指定的层后，会自动停止，并使用端口连接，并解析所有信号

创建Debug模块

为了更好的调试，一般在输出会添加debug 模块

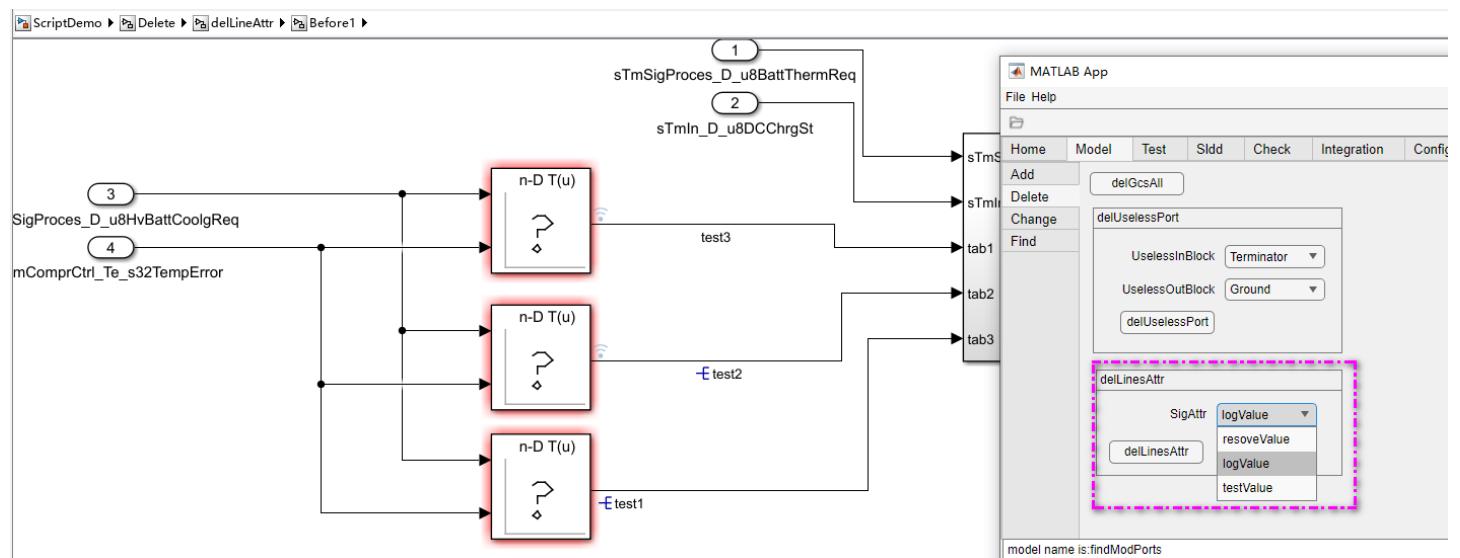


删除无用端口

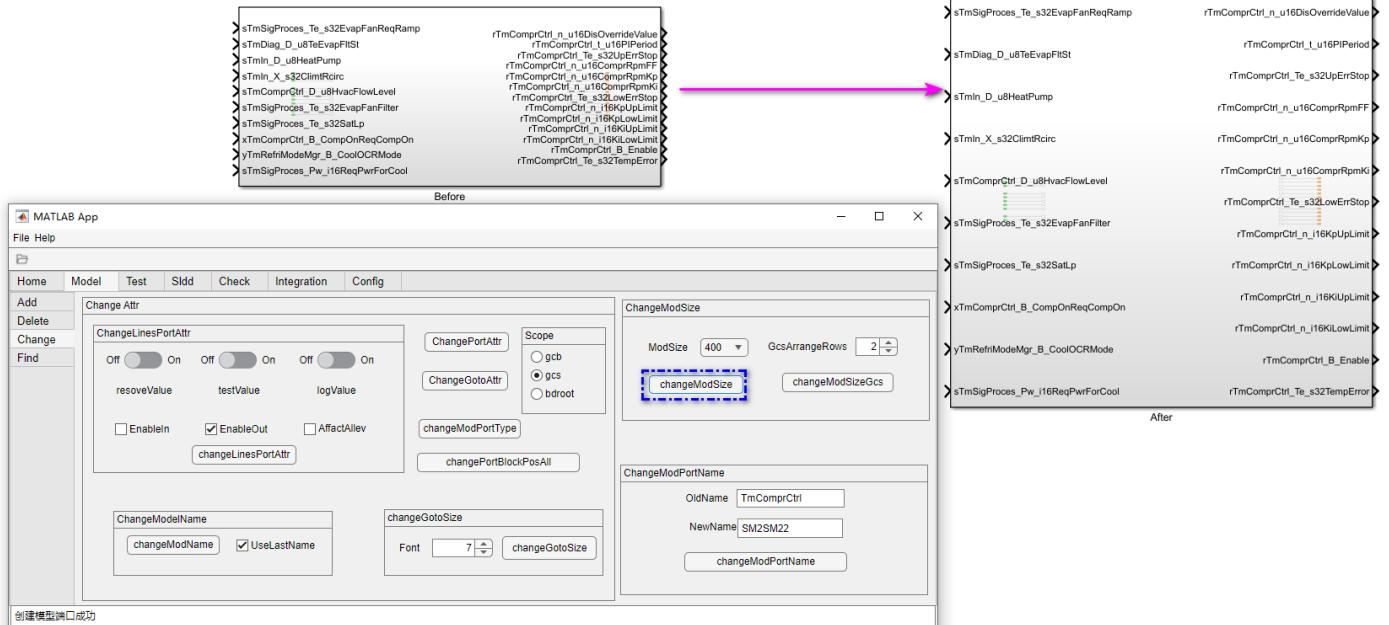


删除无用的线属性

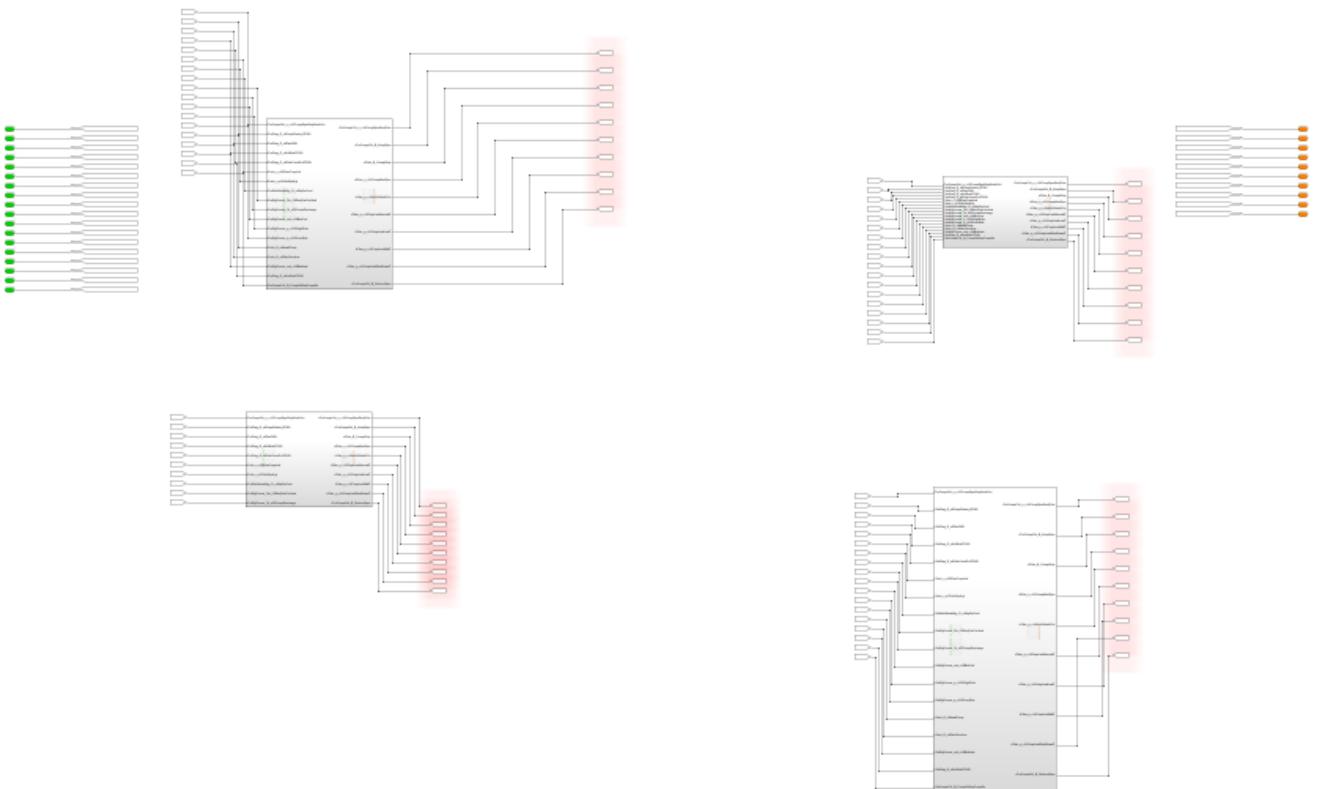
有时候仿真的时候添加了过多的信号线观测属性，仿真完可快捷删掉

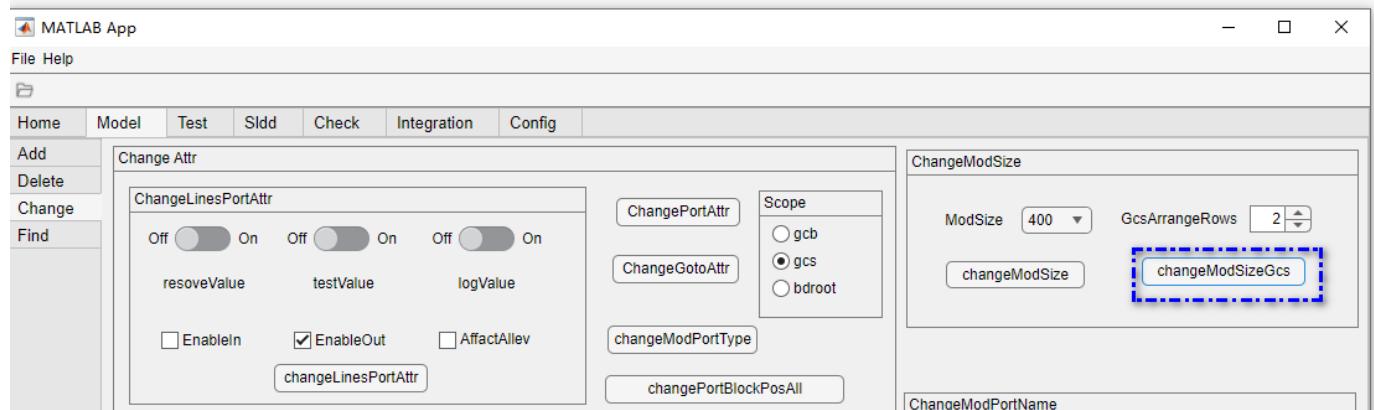


改变模型大小

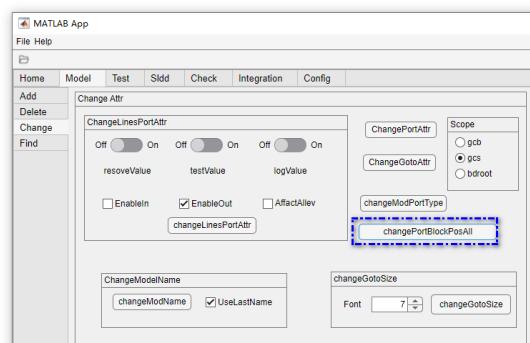
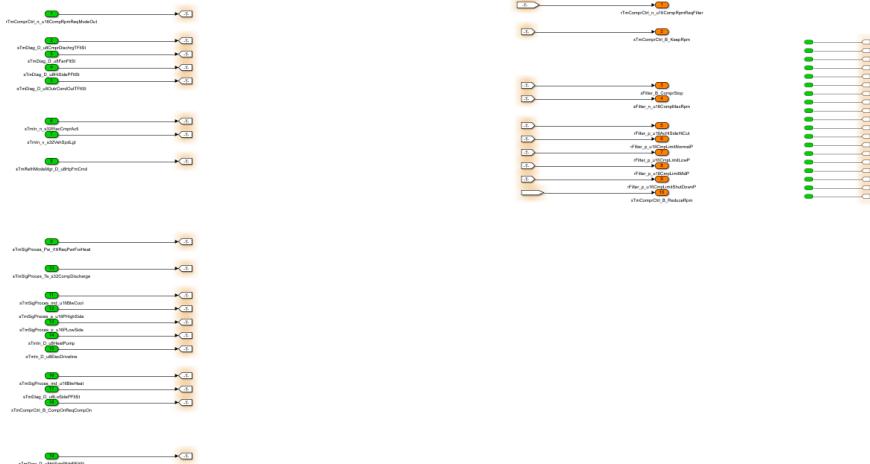


改变当前层的模型布局

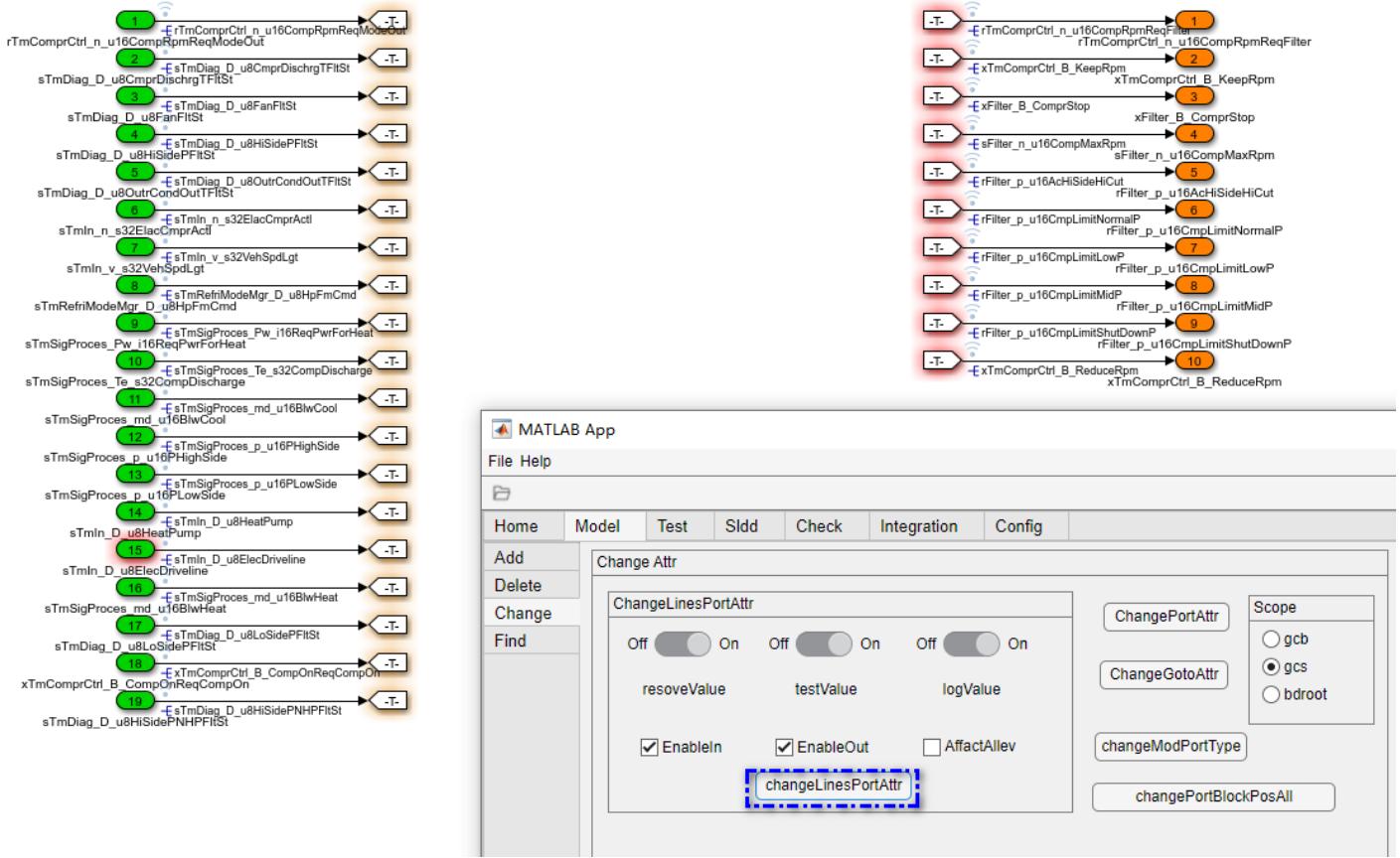




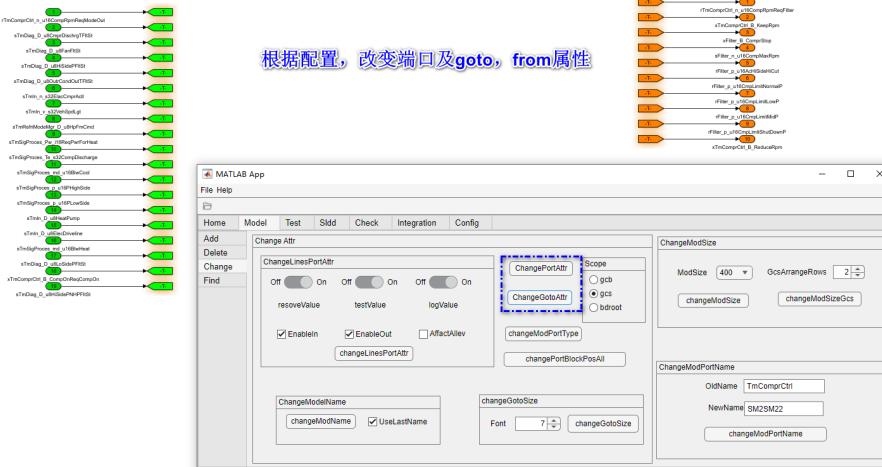
改变输入端口位置



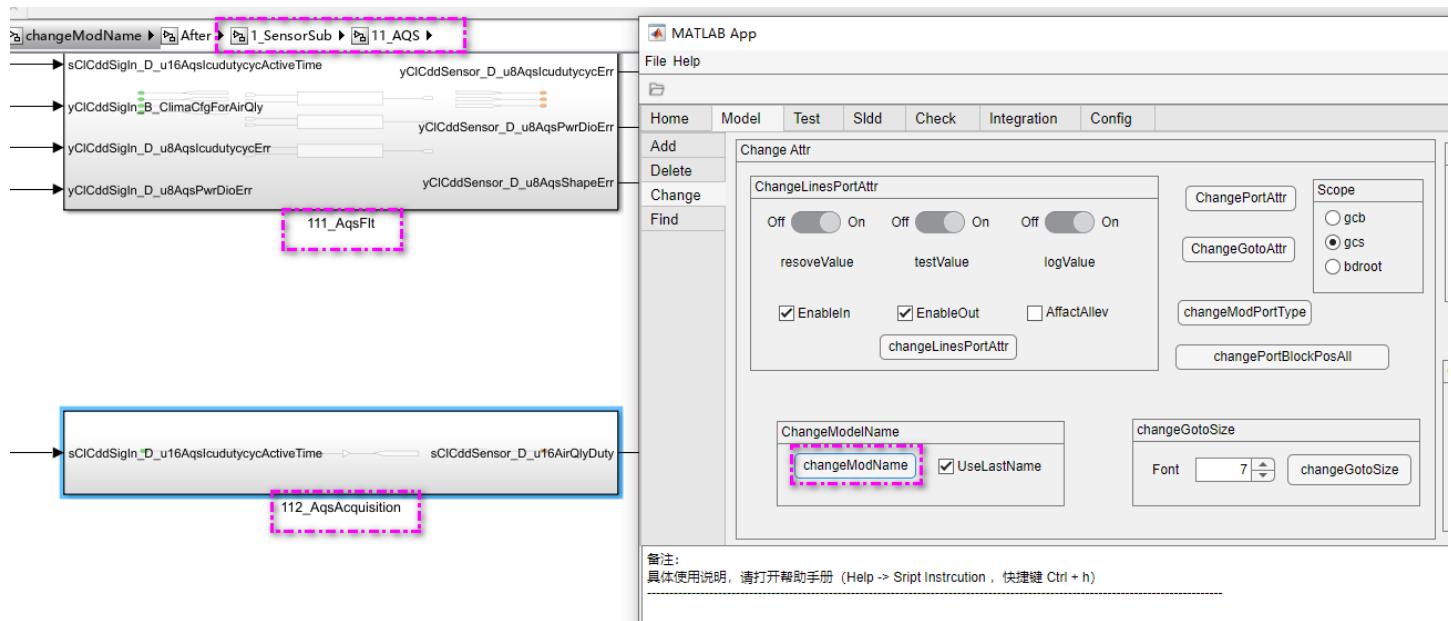
改变输入输出端口信号线属性



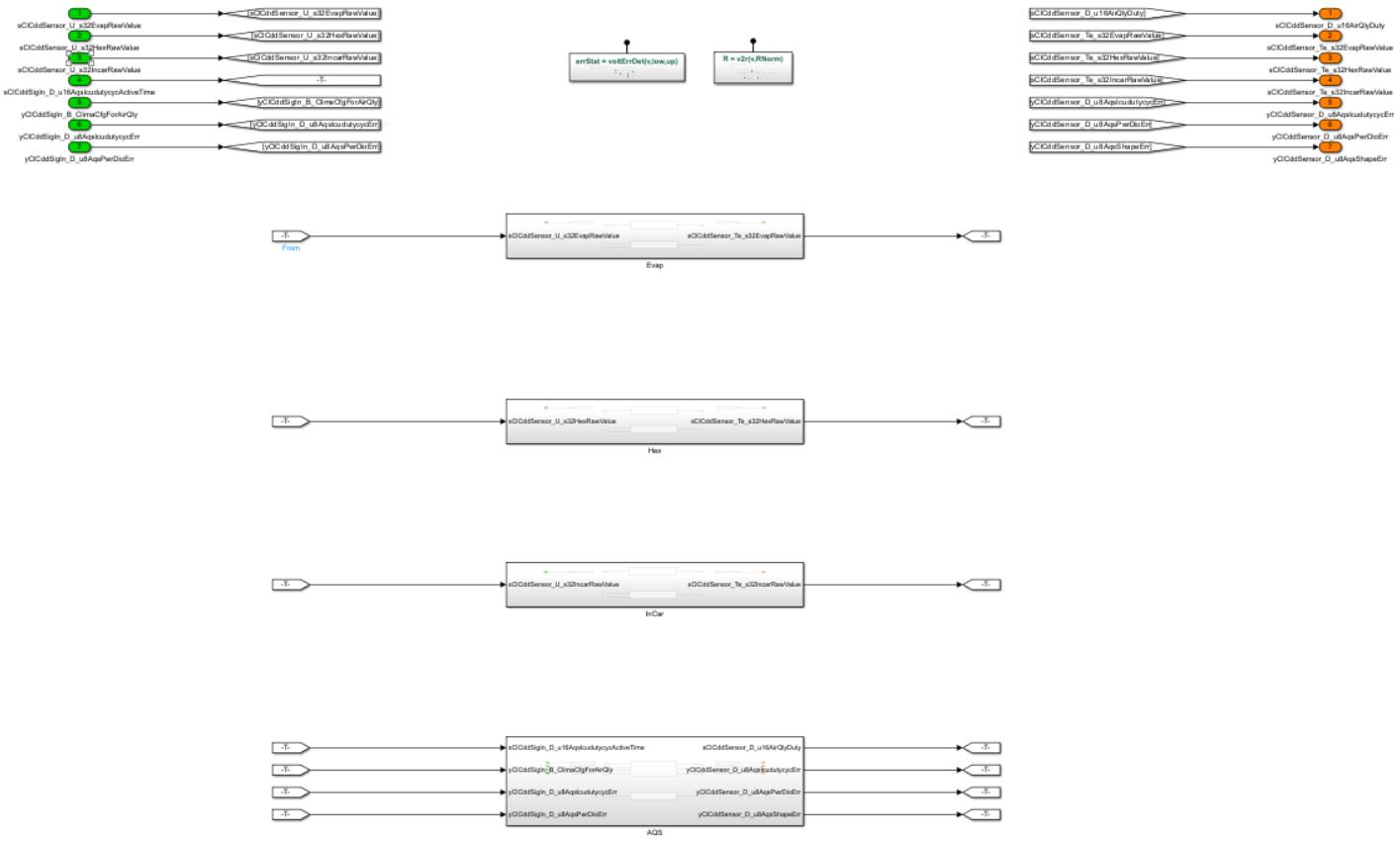
改变端口, goto,from 属性

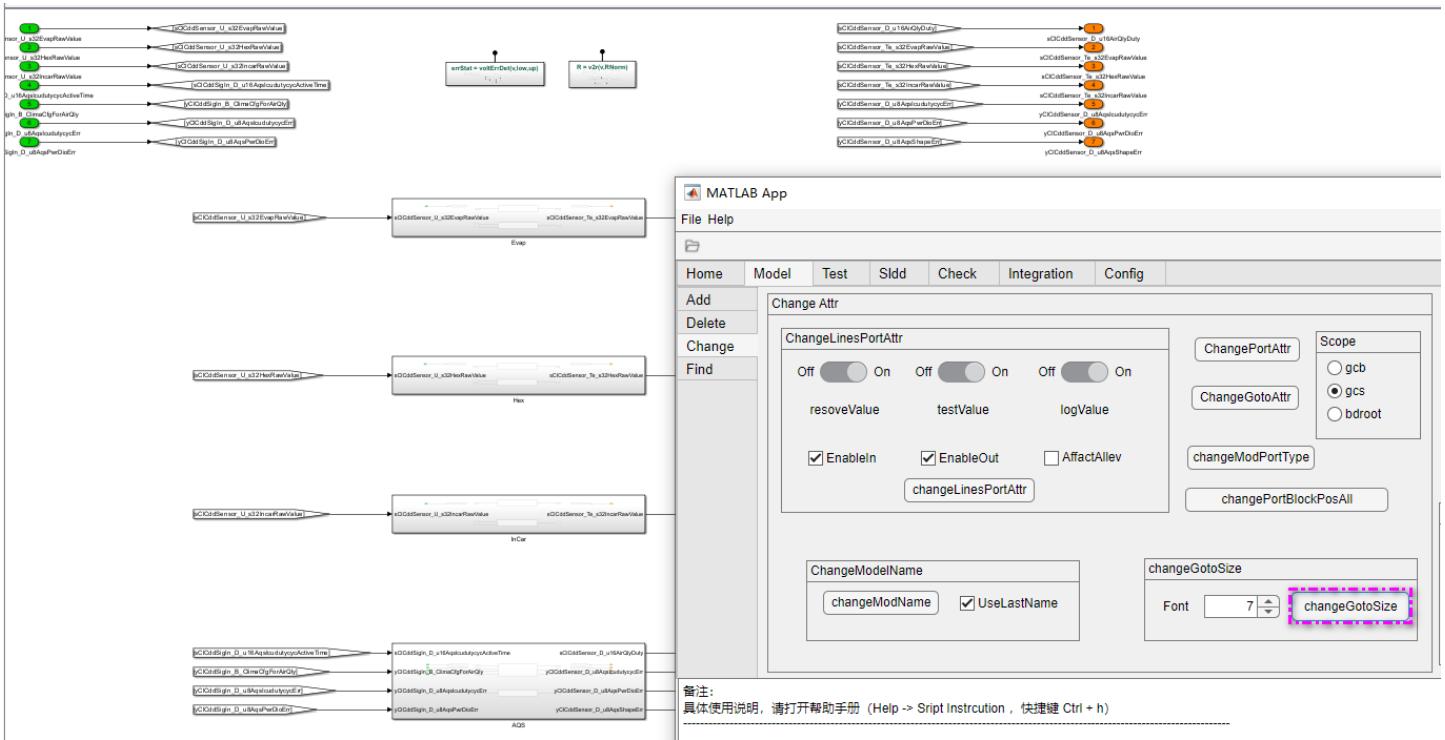


改变模型命名规则



改变goto 长度，根据标签





查找端口信息

MATLAB App

File Help

Home Model Test Sldd Check Integration Config

Add Delete Change Find

findModPorts PortType Path ▾

model name is: findModPorts

输入端口:

```
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_X_s32EEEXVPosReqIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_X_s32BEEXVPosReqIn
ScriptDemo/Find/findModPorts/yTmRefriVlvCtrl_B_HpcCexVmEnableIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_Z_u8CEXVInitStateIn
ScriptDemo/Find/findModPorts/yTmRefriVlvCtrl_B_CexvCalibReqIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_B_HpcCexVmEnableIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_Z_u8EEEXVInitStateIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_Z_u8BEEXVInitStateIn
ScriptDemo/Find/findModPorts/yTmRefriVlvCtrl_B_BevxCalibReqIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_X_s32SM3BexVmMaxPosIn
ScriptDemo/Find/findModPorts/yTmRefriVlvCtrl_B_EvapBattTlmbExvEnaln
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_D_u8ExvRefriSwitchStIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_D_u8CexvRefriSwitchStIn
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_D_u8BevRefriSwitchStIn
```

输出端口:

```
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_X_s32CEXVPosReq
ScriptDemo/Find/findModPorts/sTmRefriVlvCtrl_X_s32FFEXVPosReq
```

MATLAB App

File Help

Home Model Test Sldd Check Integration Config

Add Delete Change Find

findModPorts PortType Name ▾

model name is: findModPorts

输入端口:

```
sTmRefriVlvCtrl_X_s32EEEXVPosReqIn
sTmRefriVlvCtrl_X_s32BEEXVPosReqIn
yTmRefriVlvCtrl_B_HpcCexVmEnableIn
sTmRefriVlvCtrl_Z_u8CEXVInitStateIn
yTmRefriVlvCtrl_B_CexvCalibReqIn
yTmRefriVlvCtrl_B_HpcCexVmEnableIn
sTmRefriVlvCtrl_Z_u8EEEXVInitStateIn
sTmRefriVlvCtrl_Z_u8BEEXVInitStateIn
yTmRefriVlvCtrl_B_BevxCalibReqIn
sTmRefriVlvCtrl_X_s32SM3BexVmMaxPosIn
yTmRefriVlvCtrl_B_EvapBattTlmbExvEnaln
sTmRefriVlvCtrl_D_u8ExvRefriSwitchStIn
sTmRefriVlvCtrl_D_u8CexvRefriSwitchStIn
sTmRefriVlvCtrl_D_u8BevRefriSwitchStIn
```

输出端口:

```
sTmRefriVlvCtrl_X_s32CEXVPosReq
sTmRefriVlvCtrl_X_s32FFEXVPosReq
```

MATLAB App

File Help

Home Model Test Sldd Check Integration Config

Add Delete Change Find

findModPorts PortType OutDataT... ▾

model name is: findModPorts

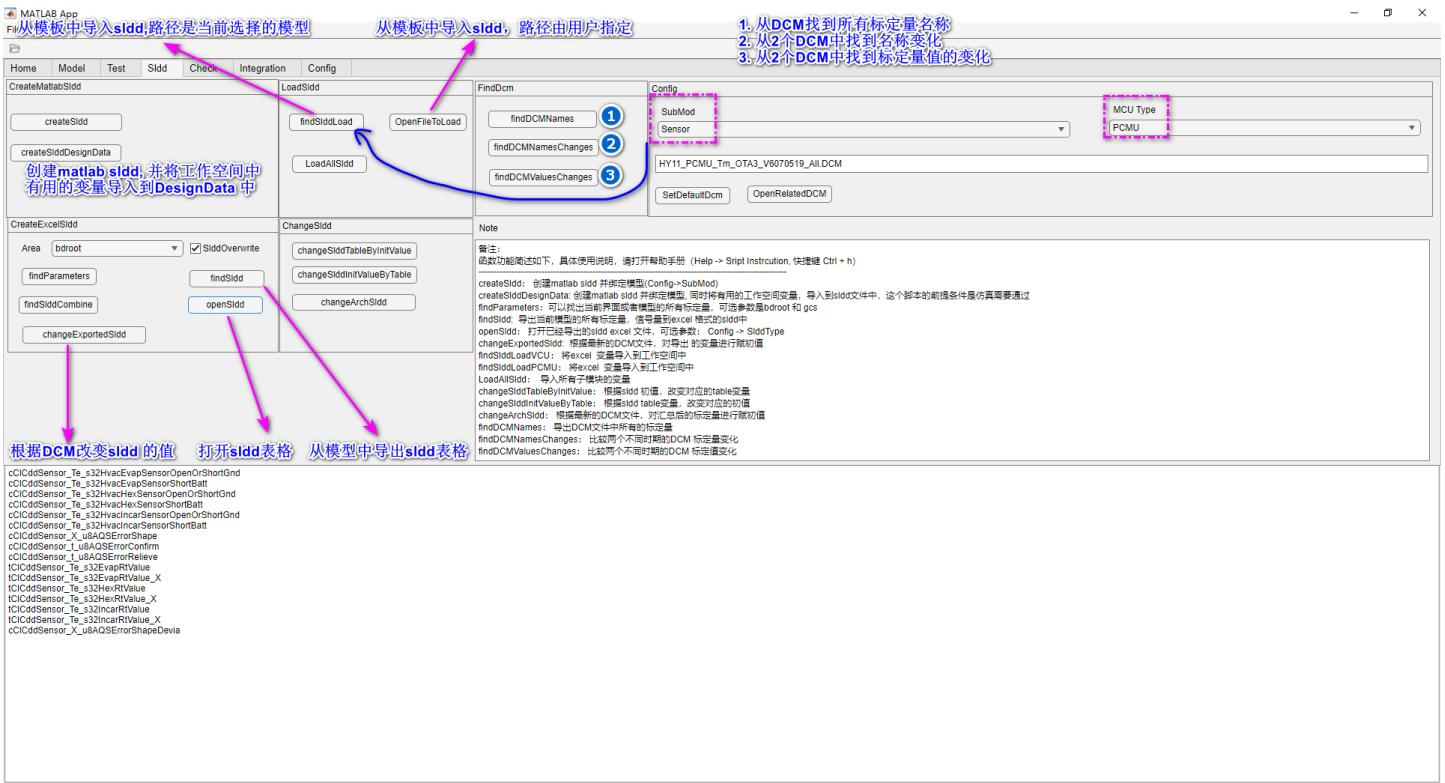
输入端口:

```
single
single
boolean
uint8
boolean
boolean
uint8
boolean
uint8
boolean
single
boolean
uint8
uint8
```

输出端口:

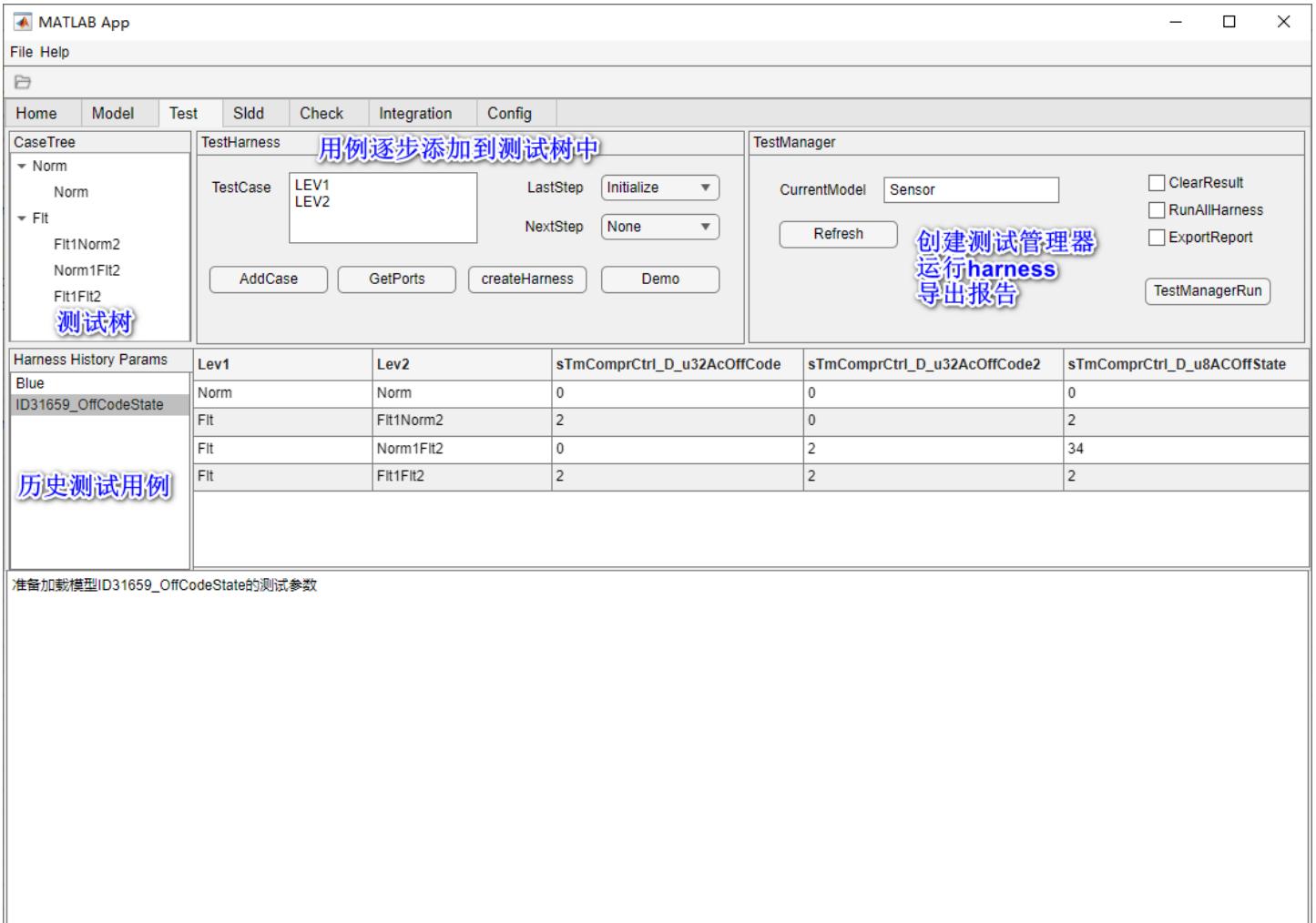
```
single
single
```

Sldd 篇



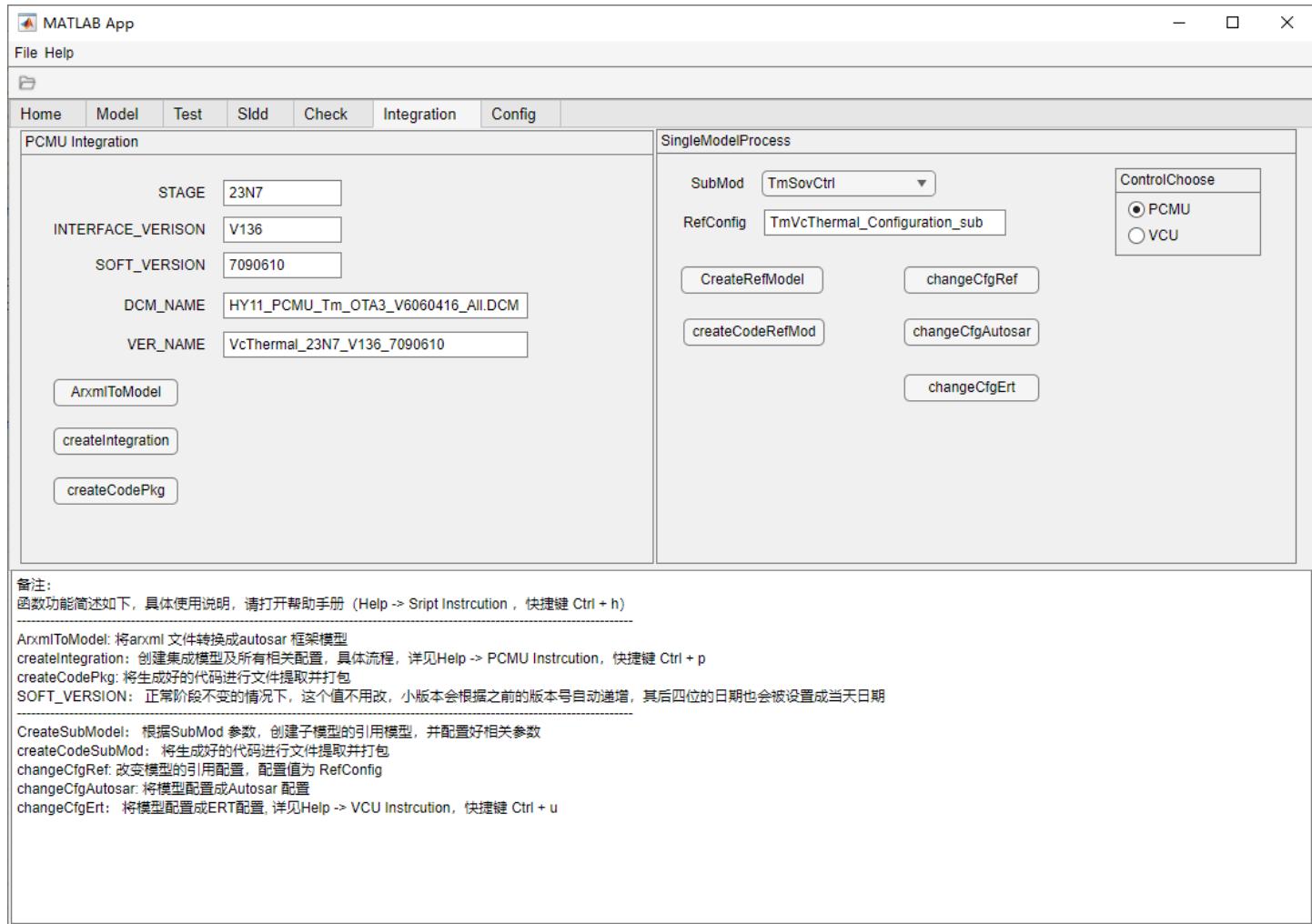
待完善

测试篇



待完善

集成篇



待完善

增

创建模型信息

痛点1: 会经常出现模型需求中的标定量和模型对不上, 导致测试人员很难看懂

痛点2: 随着标定前后的不一致, 默认标定量通常会改变

这个脚本就是基于子模型信息, 自动生成模型名称, 输入输出端口, 标定量等相关信息, 可直接复制到对应的需求上去, 如果这个子模型对应的就是某个需求的话, 从建模规范来说, 强烈推荐一个子模型对应一个需求。另外, 这个脚本可以指定最新的DCM文件, 如果针对非表格类的标定量, 脚本会总DCM查到对应的数据, 同步更新到标定量后面, 便于更好的查阅

脚本说明

```
[inPortNames, outPortNames, params] =  
createSubmodInfo('DCMfileName','HY11_PCMU_Tm_OTA3_V6050327_All.DCM')
```

代码块

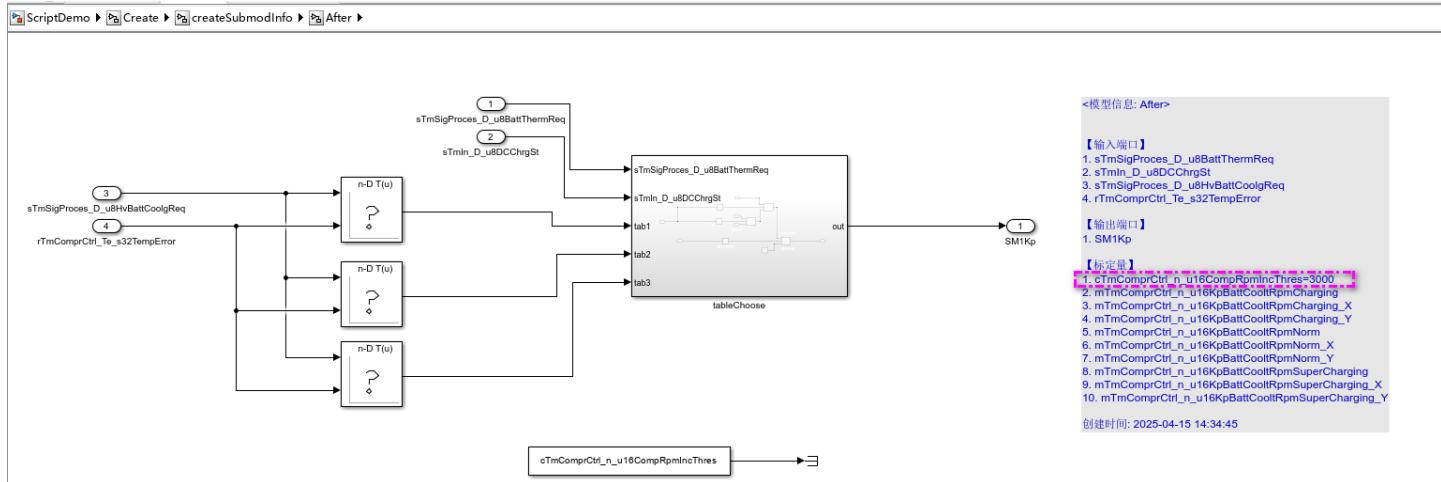
```
1 function [portsInNames, portsOutNames, calibParams] =  
2 %CREATESUBMODINFO 创建包含子模型信息的注释模块并返回收集的信息  
3 % [PORTSINNAMES, PORTSOUTNAMES, CALIBPARAMS] = CREATESUBMODINFO() 在当前子模型  
4 % 中创建信息注释模块  
5 % [PORTSINNAMES, PORTSOUTNAMES, CALIBPARAMS] = CREATESUBMODINFO('Parameter',  
6 % Value, ...) 使用指定参数创建  
7 %  
8 % 输入参数 (名值对) :  
9 % 'path' - 目标系统路径 (字符串), 默认值: gcs (当前系统)  
10 % 'position' - 注释模块位置 [x y width height] (数组), 默认自动计算  
11 % 'fontSize' - 字体大小 (正整数), 默认值: 12  
12 % 'fgColor' - 前景颜色 (字符串), 默认值: 'blue'  
13 % 'bgColor' - 背景颜色 [R G B] (数组), 默认值: [0.9 0.9 0.9]  
14 % 'includeCnt' - 是否包含序号 (逻辑值), 默认值: false  
15 % 'userData' - 用户数据 (任意类型), 默认值: 'modInfo'  
16 % 'DCMfileName' - DCM文件名 (字符串), 默认值:  
17 % 'HY11_PCMU_Tm_OTA3_V6050327_All.DCM'  
18 %  
19 % 输出参数:  
20 % portsInNames - 输入端口名称列表 (元胞数组)  
21 % portsOutNames - 输出端口名称列表 (元胞数组)  
22 % calibParams - 标定量列表 (元胞数组)  
23 %  
24 % 功能描述:  
25 % 1. 收集子模型的输入端口信息  
26 % 2. 收集子模型的输出端口信息  
27 % 3. 收集子模型内部使用的标定量  
28 % 4. 在子模型中创建包含这些信息的注释模块  
29 % 5. 返回收集到的信息, 可用于其他处理  
30 %  
31 % 示例:  
32 % [inPortNames, outPortNames, params] = createSubmodInfo()  
33 % [inPortNames, outPortNames, params] =  
34 % createSubmodInfo('DCMfileName','HY11_PCMU_Tm_OTA3_V6050327_All.DCM')  
35 % [inPortNames, outPortNames, params] = createSubmodInfo('fontSize', 14)  
36 % [~, ~, params] = createSubmodInfo('path', 'myModel/subsystem1',  
% 'includeCnt', true)  
37 %  
38 % 注意事项:  
39 % 1. 使用前需要打开目标Simulink模型  
40 % 2. 如果已存在同名注释模块, 会先删除再创建
```

```

37 % 3. 即使不需要创建注释模块，也可以使用此函数收集模型信息
38 %
39 % 参见: FINDMODPORTS, ADD_BLOCK, GET_PARAM, SET_PARAM, SIMULINK.FINDVARS,
40 % FINDANNOTATION
41 % 作者: Blue.ge
42 % 版本: 1.6
43 % 日期: 20250312

```

执行效果



为bus 总线创建对应的goto

总线信号输出的时候，如果需要转换成goto，便于其它模块调用，手动添加会比较麻烦，因此此脚本可实现批量根据bus 中的名称，创建对应的goto 模块

使用说明

createBusGoto()

代码块

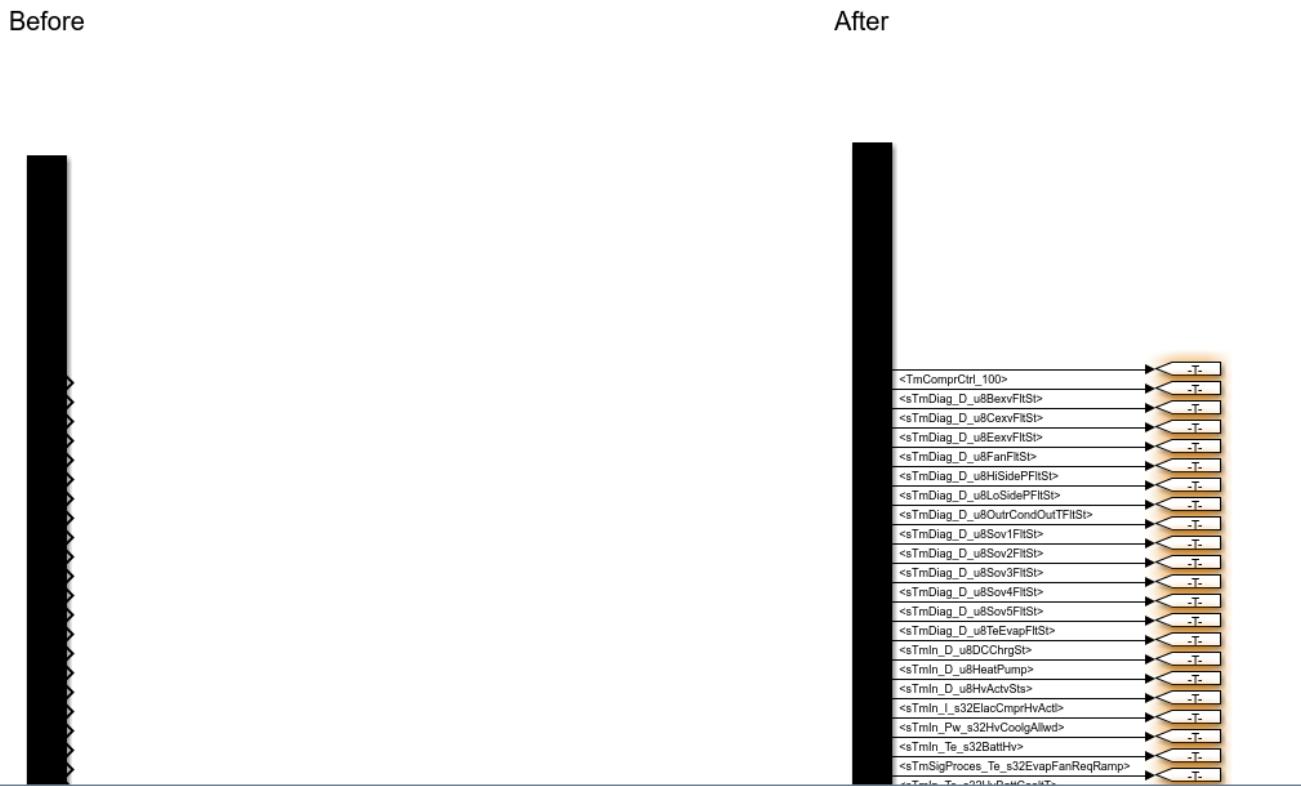
```

1 function result = createBusGoto()
2 %CREATEBUSGOTO 为总线信号创建对应的Goto模块
3 % result = createBusGoto() 为当前选中的总线信号创建对应的Goto模块。
4 % 该函数会自动为总线中的每个未连接的输出信号创建Goto模块，并建立连接。
5 %
6 % 输入参数:
7 % 无
8 %
9 % 输出参数:
10 % result - 操作结果，成功返回true，失败返回false
11 %
12 % 示例:
13 % createBusGoto()

```

```
14 %  
15 % 作者: 葛维冬 (Blue Ge)  
16 % 日期: 2023-10-09  
17 % 版本: 1.1
```

执行效果



为bus 总线创建对应的port

总线信号输出的时候，如果需要转换成port, 便于其它模块调用，手动添加会比较麻烦，因此此脚本可实现批量根据bus 中的名称，创建对应的port模块

使用说明

createBusPort()

代码块

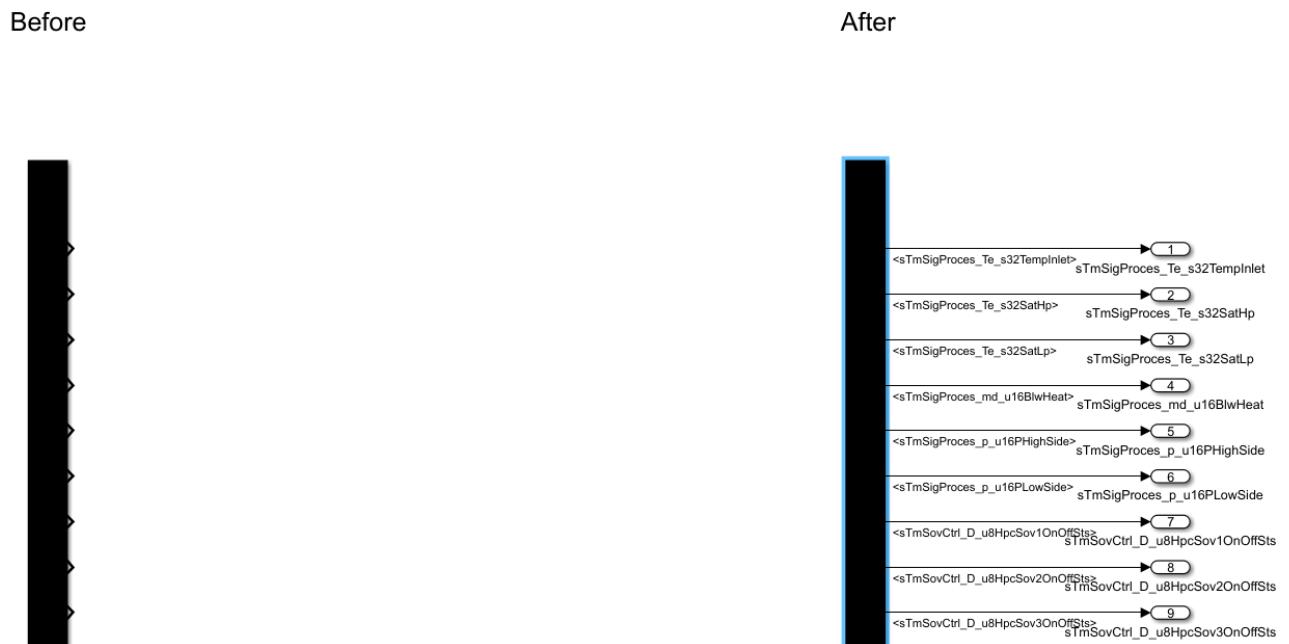
```
1 function result = createBusPort()  
2 %CREATEBUSPORT 为总线信号创建对应的输出端口  
3 % result = createBusPort() 为当前选中的总线信号创建对应的输出端口。  
4 % 该函数会自动为总线中的每个未连接的输出信号创建Outport模块，并建立连接。  
5 %  
6 % 输入参数：  
7 % 无
```

```

8 %
9 %   输出参数:
10 %       result - 操作结果, 成功返回true, 失败返回false
11 %
12 %   示例:
13 %       createBusPort()
14 %
15 %   作者: 葛维冬 (Blue Ge)
16 %   日期: 2023-10-09
17 %   版本: 1.1

```

执行效果



创建autosar 代码包

Autosar 生成代码后，需要格式化打包下，比如软件版本名称：下面的src, h, a2l 文件夹， 需要把生成的代码各个子模块对应的文件，拷贝到其中

使用说明

```
createCodePkg('VcThermal_23N7_V131_6666666')
```

代码块

```

1 function createCodePkg(verName, varargin)
2 %CREATECODEPKG 创建代码包并整理生成的文件
3 % createCodePkg(verName) 根据指定的版本名称创建代码包，并将生成的代码文件(默认根目录
4 % 下的CodeGen)整理到相应的目录中。

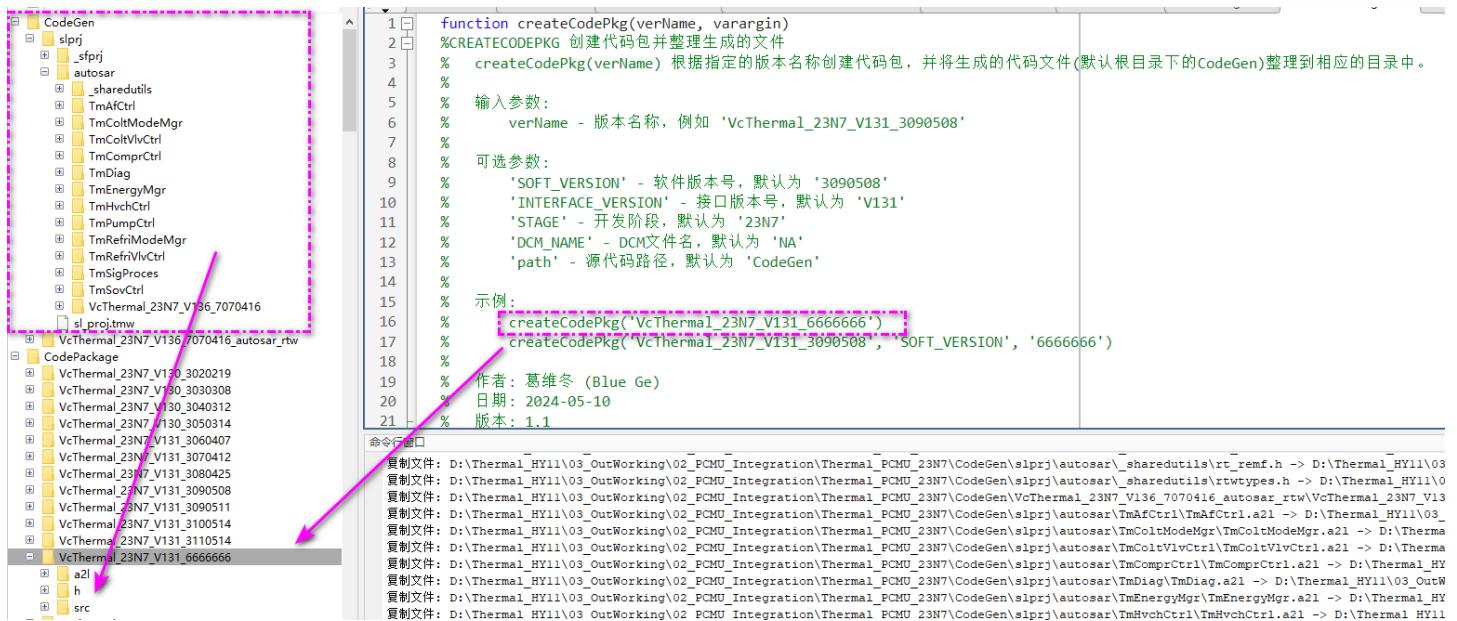
```

```

4 %
5 % 输入参数:
6 %         verName - 版本名称, 例如 'VcThermal_23N7_V131_3090508'
7 %
8 % 可选参数:
9 %         'SOFT_VERSION' - 软件版本号, 默认为 '3090508'
10 %        'INTERFACE_VERSION' - 接口版本号, 默认为 'V131'
11 %        'STAGE' - 开发阶段, 默认为 '23N7'
12 %        'DCM_NAME' - DCM文件名, 默认为 'NA'
13 %        'path' - 源代码路径, 默认为 'CodeGen'
14 %
15 % 示例:
16 %         createCodePkg('VcThermal_23N7_V131_6666666')
17 %
18 % 作者: 葛维冬 (Blue Ge)
19 % 日期: 2024-05-10
20 % 版本: 1.1

```

执行效果



创建autosar 集成工程

- % 1. 配置集成参数
- % 2. 加载SLDD文件
- % 3. 配置软件版本
- % 4. 配置架构模型
- % 5. 加载IDT配置
- % 6. 创建架构

% 7. 将base复制到autosar架构模型中

% 8. 清理代码生成目录

使用说明

```
createIntegration('4070429')
```

代码块

```
1 function createIntegration(softversion,varargin)
2 %CREATEINTEGRATION 创建并配置 Thermal PCMU 集成环境
3 %
4 % createIntegration(SOFTVERSION) 使用指定的软件版本创建集成环境
5 %
6 % createIntegration(SOFTVERSION,Name,Value) 使用指定的软件版本和名称-值对参数创建
7 % 集成环境
8 %
9 % 输入参数:
10 %
11 % 可选名称-值对参数:
12 % STAGE - 项目阶段, 默认为 '23N7'
13 % INTERFACE_VERISON - 接口版本, 默认为 'V136'
14 % CAL_VERSION - 标定版本, 默认为 '666666'
15 % DCM_NAME - DCM文件名, 默认为 'HY11_PCMU_Tm_OTA3_V6060416_All.DCM'
16 % SLDD_NAME - SLDD文件名, 默认为 'PCMU_SLDD_All.xlsx'
17 % ARCH_NAME - 架构模型名称, 默认为 'TmSwArch'
18 %
19 % 功能说明:
20 % 1. 配置集成参数
21 % 2. 加载SLDD文件
22 % 3. 配置软件版本
23 % 4. 配置架构模型
24 % 5. 加载IDT配置
25 % 6. 创建架构
26 % 7. 将base复制到autosar架构模型中
27 % 8. 清理代码生成目录
28 %
29 % 示例:
30 % % 使用默认参数创建集成环境
31 % createIntegration('7070416')
32 %
33 % % 使用自定义参数创建集成环境
34 % createIntegration('7070416', 'STAGE', '23N7', 'INTERFACE_VERISON',
35 % 'V136')
36 % 注意事项:
```

```
37 % - 确保所有必需的文件和目录都存在  
38 % - 确保有足够的权限访问相关文件  
39 % - 建议在执行前备份重要文件  
40 %  
41 % 作者: [Blue.ge]  
42 % 创建日期: [2025.4.28]  
43 % 最后修改: [2025.4.28]
```

执行效果

执行脚本后，就会自动创建好集成工程，然后电机autosar 即可生成代码

创建PCMU子模型引用

模型搭建好后，需要使用引用模型，对其生成代码验证，大概流程包括如下

- 根据模型名称导入对应的sldd
- 打开引用模型，创建模型引用
- 对模型及引用模型进行autorsar 配置
- 清空代码目录CodeGen

之后就可以直接进入autosar app 进行代码生成了

使用说明

```
creatRefPCMU('TmSovCtrl')
```

代码块

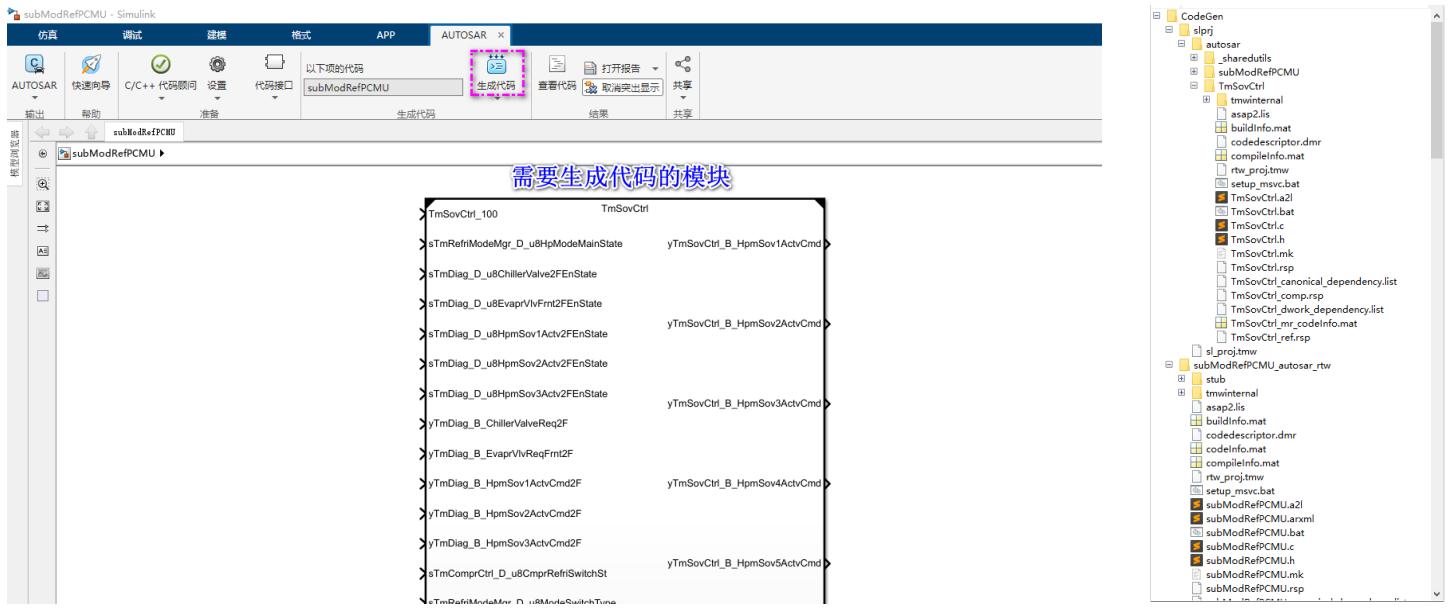
```
1 function creatRefPCMU(modelName, varargin)  
2 %CREATEREFPCMU 创建PCMU配置的模型引用  
3 % creatRefPCMU(modelName) 为指定的模型创建PCMU配置的模型引用。  
4 %  
5 % 输入参数:  
6 % modelName - 模型名称，例如 'TmSovCtrl'  
7 %  
8 % 可选参数:  
9 % 'SLDD' - 自定义SLDD文件路径，默认为 'None'  
10 % 'DCM' - DCM文件路径， 默认为 'None'  
11 %  
12 % 示例:  
13 % creatRefPCMU('TmSovCtrl')  
14 % creatRefPCMU('TmComprCtrl', 'SLDD', 'custom_DD.xlsx')  
15 %  
16 % 作者: 葛维冬 (Blue Ge)
```

17 % 日期: 2024-02-04

18 % 版本: 1.1

执行效果

创建并配置代码生成的所有工作，随后即可直接进行代码生成



创建子模型代码包

软件搭建完成后，需要对单个子模型进行代码生成，同样需要提取代码包中的，c, h, a2l

使用说明

```
createCodeSubMod('TmSovCtrl', 'type', 'PCMU')
```

```
createCodeSubMod('TmSovCtrl', 'type', 'VCU'), 适用于VCU代码生成后的打包
```

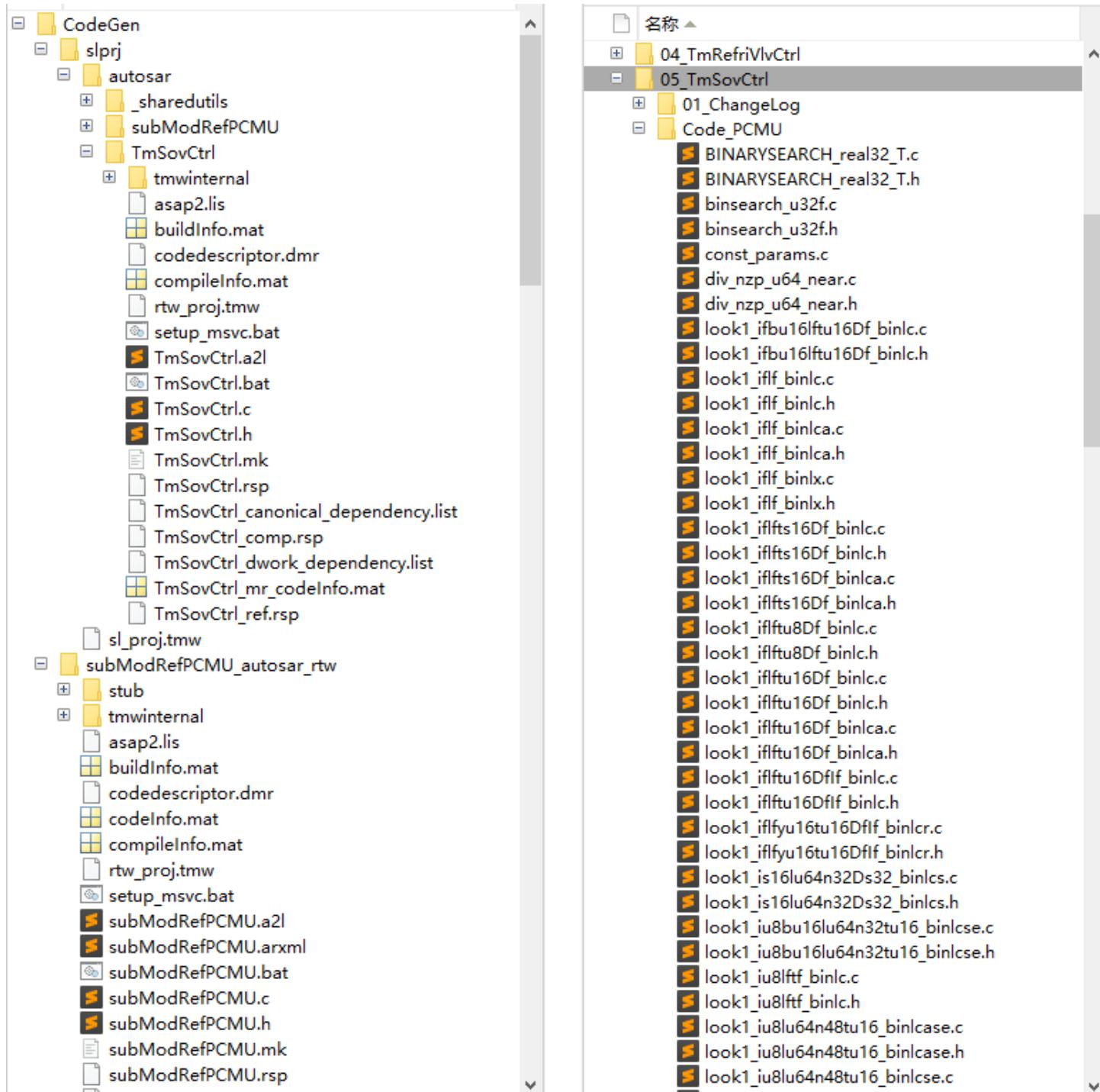
代码块

```
1 function createCodeSubMod(mdName, varargin)
2 %CREATECODESUBMOD 创建子模型代码包
3 % createCodeSubMod(mdName) 为指定的模型创建代码包，并将生成的代码文件整理到相应的目
4 % 录中。
5 %
6 % 输入参数:
7 %
8 % mdName - 模型名称，例如 'TmSovCtrl'
9 %
10 % 可选参数:
11 % 'type' - 控制器类型，可选值为 'VCU' 或 'PCMU'，默认为 'VCU'
12 %
13 % 示例:
14 %     createCodeSubMod('TmSovCtrl')
15 %     createCodeSubMod('TmSovCtrl', 'type', 'PCMU')
```

```
14 %  
15 % 作者: 葛维冬 (Blue Ge)  
16 % 日期: 2024-05-10  
17 % 版本: 1.1
```

执行效果

所有有用的相关代码，会自动整理到对应模型的Code_PCMU 下



- 根据模型名称导入对应的slld
- 打开引用模型，创建模型引用
- 对模型及引用模型进行ERT配置
- 清空代码目录CodeGen

之后就可以直接进入C代码 app 进行代码生成了

使用说明

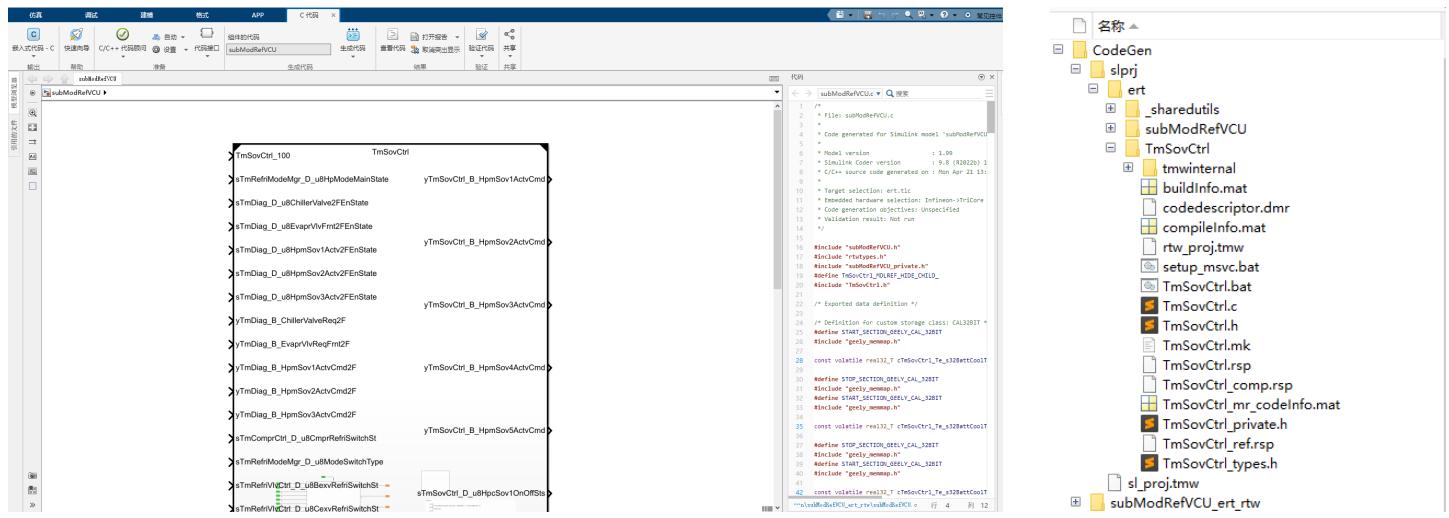
creatRefVCU('TmSovCtrl')

代码块

```
1 function creatRefVCU(modelName, varargin)
2 %CREATEREFCU 创建VCU配置的模型引用
3 % creatRefVCU(modelName) 为指定的模型创建VCU配置的模型引用。
4 %
5 % 输入参数：
6 %     modelName - 模型名称，例如 'TmSovCtrl'
7 %
8 % 可选参数：
9 %     'SLDD' - 自定义SLDD文件路径，默认为 'None'
10 %      'DCM' - DCM文件路径， 默认为 'None'
11 %
12 % 示例：
13 %     creatRefVCU('TmSovCtrl')
14 %     creatRefVCU('TmComprCtrl', 'SLDD', 'custom_DD.xlsx')
15 %
16 % 作者：葛维冬 (Blue Ge)
17 % 日期：2024-02-04
18 % 版本：1.1
```

执行效果

创建并配置代码生成的所有工作，随后即可直接进行代码生成



创建信号输出Debug模块

为了测试方便，一般会在每个模型的输出增加debug 模块，实现输出信号的强标，此脚本的目的就是根据输出信号，自动生成debug 模块。

- 根据输出信号，自动创建debug 模块和输入信号
- 为创建好的子模型，对于输入会自动创建from 信号，输出创建port 信号

使用说明

`cnt = createDebug(gcs)`

`cnt = createDebug(gcs, 'NameInType', 'findLoc')`

代码块

```

1 function cnt = createDebug(path, varargin)
2 %CREATEDEBUG 为输出端口创建debug及对应的输入模块
3 %   cnt = createDebug(path) 为指定路径下的输出端口创建debug模块和对应的输入模块，在独立的子模型中执行脚本。
4 %
5 %   输入参数：
6 %       path - 模型路径，例如 gcs
7 %
8 %   可选参数：
9 %       'step' - 端口间距，默认为30
10 %       'NameInType' - 输入端口命名方式，可选值为 'tailIn' 或 'findLoc'，默认为
11 %       'tailIn'
12 %
13 %   输出参数：
14 %       cnt - 创建的端口数量

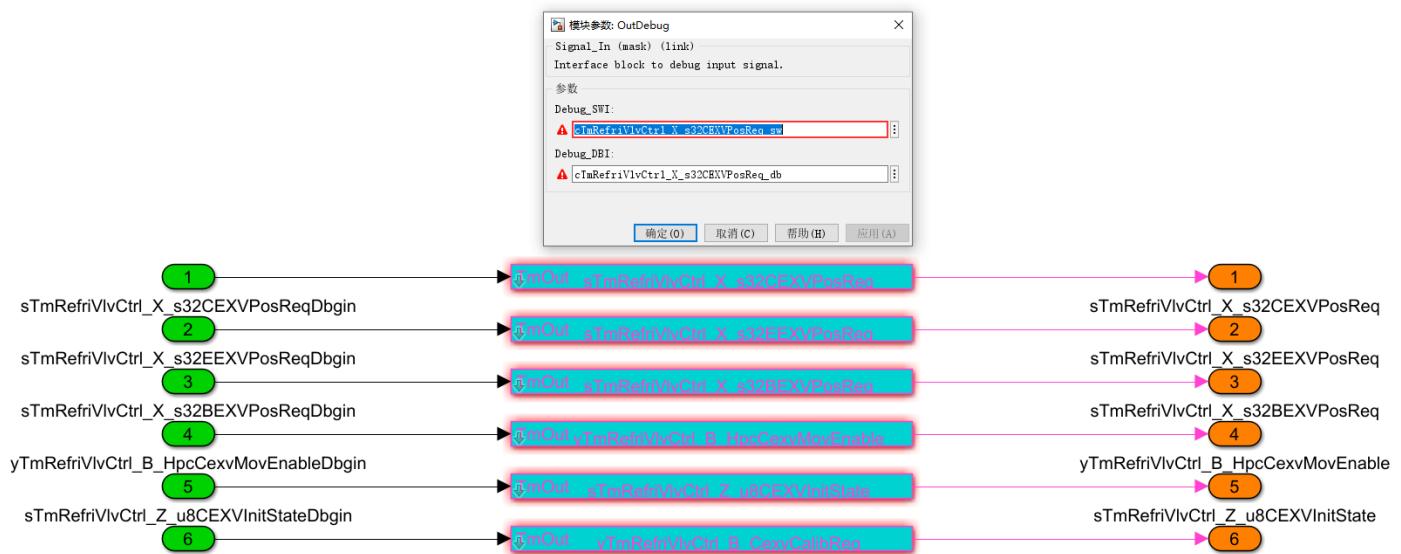
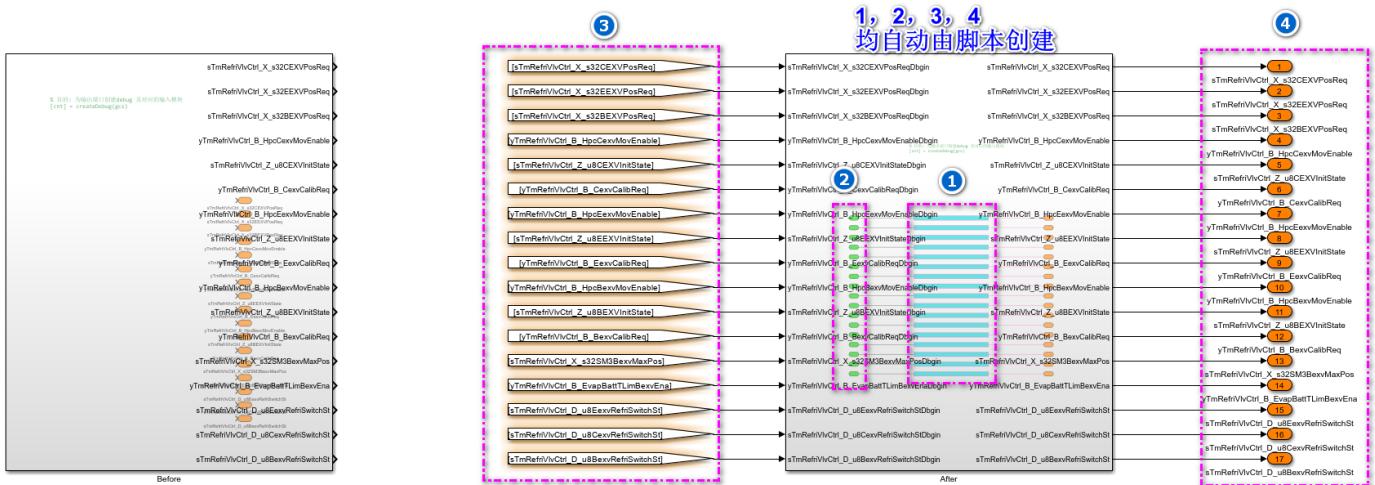
```

```

15 % 示例:
16 % cnt = createDebug(gcs) % 输入端口自动添加后缀
17 % cnt = createDebug(gcs, 'NameInType', 'findLoc') % 根据from 模块, 自动改变
18 % 输入端口名称
19 % 作者: 葛维冬 (Blue Ge)
20 % 日期: 2024-04-29
21 % 版本: 1.1

```

执行效果



根据端口创建对应的goto from

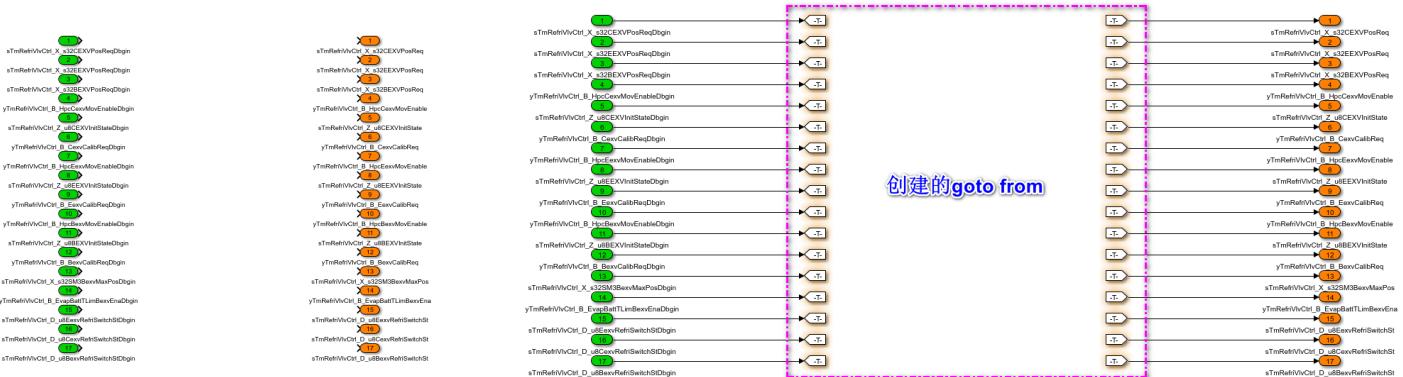
有些场景已经知道了一些输入输出端口，然后需要建模，建议直接使用goto fram 代替连线，那么使用的时候就可以直接用了

使用说明

createGotoBasedPorts(gcs)

```
代码块function [numInputPorts, numOutputPorts] = createGotoBasedPorts(path, varargin)
2 %CREATEGOTOBASEDPORTS 为输入输出端口创建对应的Goto和From模块
3 % [numInputPorts, numOutputPorts] = createGotoBasedPorts(path) 为指定路径下的
4 % 输入输出端口创建对应的Goto和From模块，并进行连线。
5 %
6 % 输入参数：
7 %     path - 模型路径，例如 gcs
8 %
9 % 可选参数：
10 %     'FiltUnconnected' - 是否过滤未连接的端口，默认为true
11 %
12 % 输出参数：
13 %     numInputPorts - 创建的输入端口数量
14 %     numOutputPorts - 创建的输出端口数量
15 %
16 % 示例：
17 %     [numIn, numOut] = createGotoBasedPorts(gcs)
18 %     [numIn, numOut] = createGotoBasedPorts(gcs, 'FiltUnconnected', false)
19 %
20 % 作者：葛维冬 (Blue Ge)
21 % 日期：2023-04-29
22 % 版本：1.1
```

执行效果



创建harness 单元测试用例

模型搭建好后，需要对每个模型单元进行单元测试，这个是非常耗时的工作，因此开发此自动化脚本，来辅助完成单元测试

使用说明

代码块

1 % 运行此脚本的时候，需要确保模型仿真通过

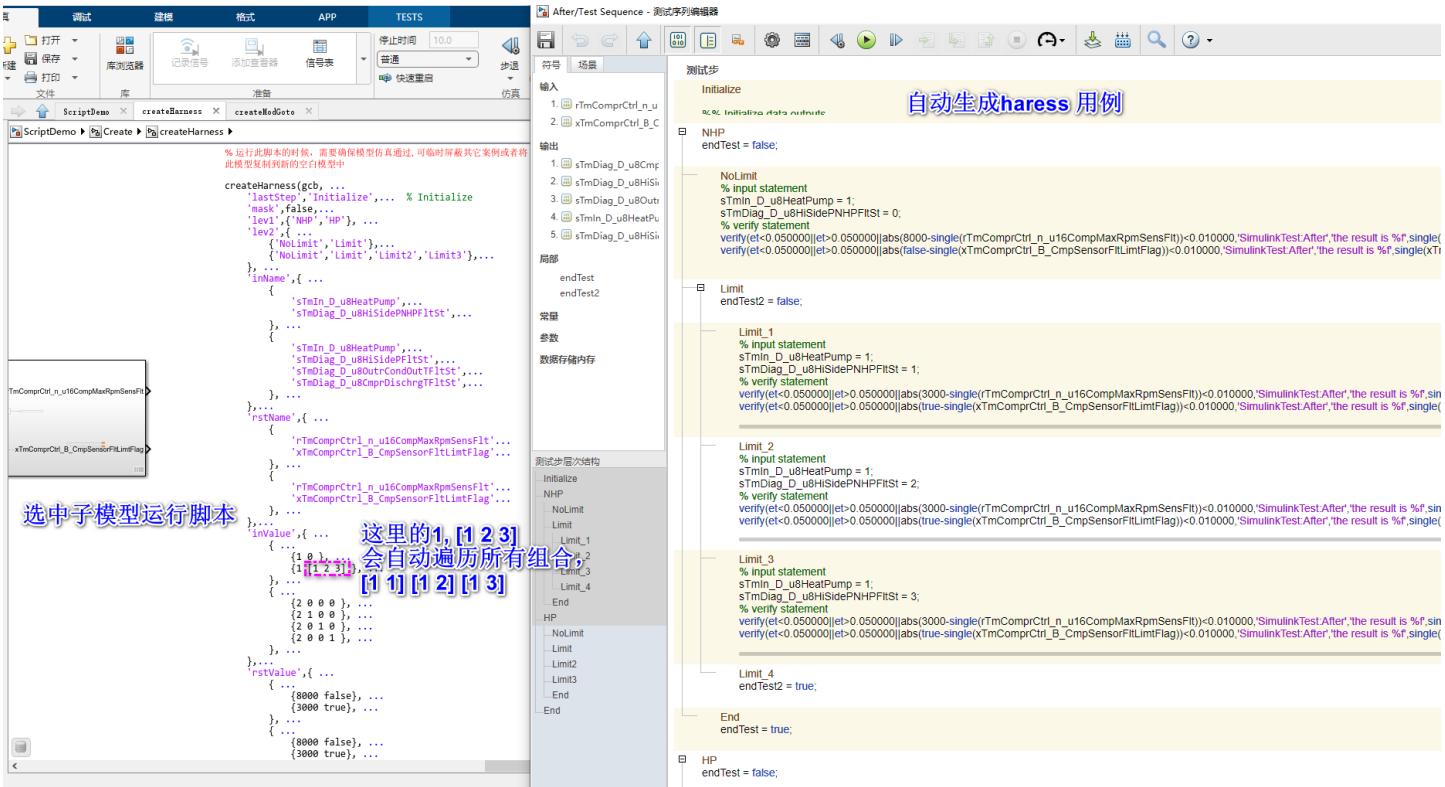
```
2
3  createHarness(gcb, ...
4      'lastStep','Initialize',... % Initialize
5      'nextStep','None',... % 默认为 'None'
6      'mask',false, ...
7      'lev1',{ 'NHP','HP' }, ...
8      'lev2',{ ...
9          {'NoLimit','Limit'},...
10         {'NoLimit','Limit','Limit2','Limit3'},...
11     }, ...
12     'inName',{ ...
13         {
14             'sTmIn_D_u8HeatPump',...
15             'sTmDiag_D_u8HiSidePNHPFltSt',...
16         }, ...
17         {
18             'sTmIn_D_u8HeatPump',...
19             'sTmDiag_D_u8HiSidePFltSt',...
20             'sTmDiag_D_u80utrCondOutTFltSt',...
21             'sTmDiag_D_u8CmprDischrgTFltSt',...
22         }, ...
23     },...
24     'rstName',{ ...
25         {
26             'rTmComprCtrl_n_u16CompMaxRpmSensFlt',...
27             'xTmComprCtrl_B_CmpSensorFltLmtFlag',...
28         }, ...
29         {
30             'rTmComprCtrl_n_u16CompMaxRpmSensFlt',...
31             'xTmComprCtrl_B_CmpSensorFltLmtFlag',...
32         }, ...
33     },...
34     'inValue',{ ...
35         { ...
36             {1 0 }, ...
37             {1 [1 2 3] }, ...
38         }, ...
39         { ...
40             {2 0 0 0 }, ...
41             {2 1 0 0 }, ...
42             {2 0 1 0 }, ...
43             {2 0 0 1 }, ...
44         }, ...
45     },...
46     'rstValue',{ ...
47         { ...
48             {8000 false}, ...
```

```
49             {3000 true}, ...
50         }, ...
51     { ...
52         {8000 false}, ...
53         {3000 true}, ...
54         {3000 true}, ...
55         {3000 true}, ...
56     }, ...
57 }...
58 )
59
```

代码块

```
1 function createHarness(path, varargin)
2 %CREATEHARNESS 为模块创建测试用例
3 %   createHarness(path) 为指定路径下的模块创建测试用例，支持新建、追加和插入三种模式。
4 %
5 %   输入参数：
6 %       path - 模型路径，例如 gcb
7 %
8 %   可选参数：
9 %       'lev1' - 第一级模块名称，默认为 {'SM123', 'SM456'}
10 %      'lev2' - 第二级模块名称，默认为 {'SM1', 'SM2'}, {'SM4', 'SM5', 'SM6'}
11 %      'inName' - 输入端口名称，默认为 {'SM11', 'SM12'}, {'SM14', 'SM15', 'SM16'}
12 %      'inValue' - 输入值，默认为 {[{10, 5}, {8, 3, 2}], {6, 4}, {4, 5, 8}}
13 %      'rstName' - 结果端口名称，默认为 {'SM1', 'SM2'}, {'SM4', 'SM5', 'SM6'}
14 %      'rstValue' - 结果值，默认为 {[{10, 20}, {10, 20, 20}], [{10, 20}, {10, 20, 20}]}
15 %      'mask' - 掩码值，默认为0
16 %      'lastStep' - 上一步操作，默认为'Initialize'
17 %      'nextStep' - 下一步操作，默认为'None'
18 %      'tolerance' - 容差值，默认为0.01
19 %      'waitTime' - 等待时间，默认为0.05
20 %      'usingAllPorts' - 是否使用所有端口，默认为false
21 %      'logValue' - 是否记录值，默认为false
22 %
23 %   示例：
24 %       createHarness(gcb)
25 %       createHarness(gcb, 'lev1', {'SM123'}, 'tolerance', 0.02)
26 %
27 %   作者：葛维冬 (Blue Ge)
28 %   日期：2024-03-29
29 %   版本：1.1
```

执行效果



创建Merge 模块

模型搭建好后，需要对每个模型单元进行单元测试，这个是非常耗时的工作，因此开发此自动化脚本，来辅助完成单元测试

使用说明

代码块

```

1 % 运行次脚本的时候，需要确保有对应的switch case 模块输出后缀信号
2 createMerge( ...
3     'sigList', ...
4     {'rTmComprCtrl_Te_s32LowErrStop', 'rTmComprCtrl_Te_s32TempError', ...
5     'rTmComprCtrl_Te_s32UpErrStop', 'rTmComprCtrl_n_u16ComprRpmFF', ...
6     'rTmComprCtrl_n_u16ComprRpmKi', 'rTmComprCtrl_n_u16ComprRpmKp', ...
7     'rTmComprCtrl_n_u16DisOverrideValue', 'rTmComprCtrl_n_i16KiLowLimit', ...
8     'rTmComprCtrl_n_i16KiUpLimit', 'rTmComprCtrl_n_i16KpLowLimit', ...
9     'rTmComprCtrl_n_i16KpUpLimit', 'rTmComprCtrl_t_u16PIPeriod', ...
10    'xTmComprCtrl_B_Enable',...
11 },...
12     'resovleSig',true)

```

代码块

```

1 function createMerge(varargin)
2 %CREATEMERGE 为当前路径下的使能子系统或switch case输出信号自动生成Merge模块

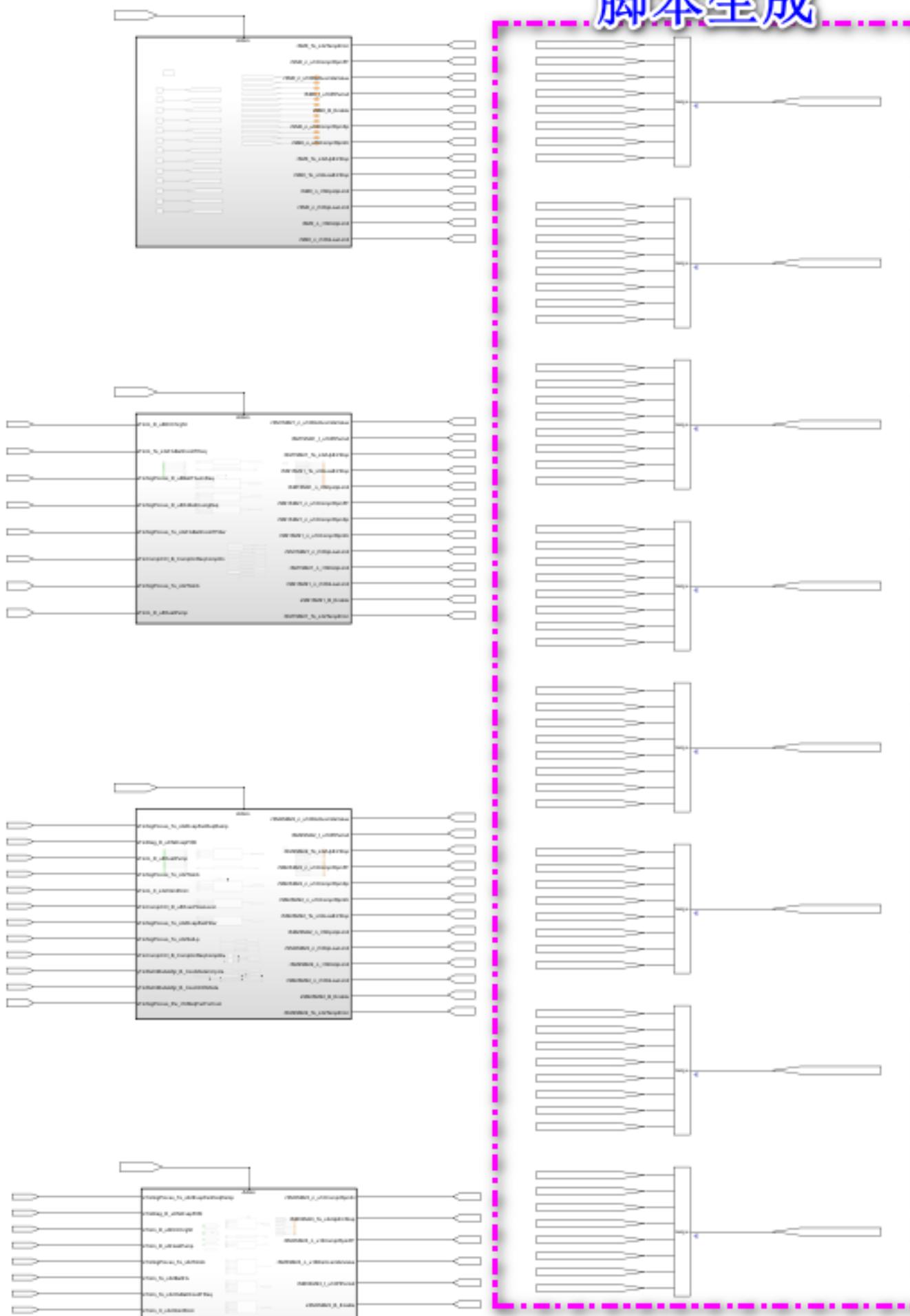
```

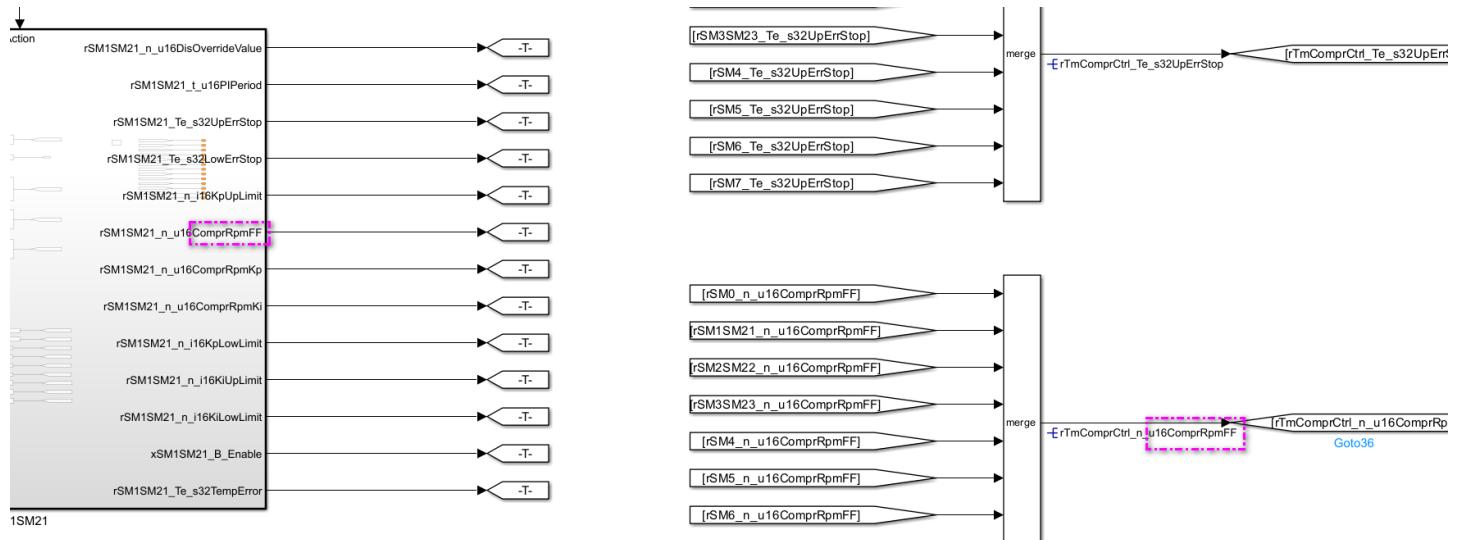
```
3 %     createMerge() 自动为当前模型中的使能子系统或switch case输出信号创建Merge模块。
4 %
5 %     可选参数：
6 %         'sigList' - 需要创建Merge的信号列表，默认为一组压缩机控制信号
7 %         'resovleSig' - 是否需要解析merge的信号，默认为true
8 %
9 %     示例：
10 %         createMerge()
11 %         createMerge('sigList', {'rTmComprCtrl_B_RdyToChange'})
12 %         createMerge('sigList', {'rTmComprCtrl_B_RdyToChange'}, 'resovleSig',
13 %                     false)
14 %
15 %     作者：葛维冬 (Blue Ge)
16 %     日期：2023-10-09
17 %     版本：1.1
```

执行效果

最终的输出信号后缀，需要跟每个switch 模块的输出信号名称，保持一致

脚本生成





创建Switch Case模块

模型搭建好后，需要对每个模型单元进行单元测试，这个是非常耗时的工作，因此开发此自动化脚本，来辅助完成单元测试

使用说明

代码块

```

1  createSwitchCase( ...
2      'caseIn', 'sTmRefriModeMgr_D_u8ModeSwitchType', ...
3      'caseList', '{1,17,[8 9 10 11 12 13 18 19],20,21,22,23}', ...
4      'caseNames',
5      {'Stop','HoldSwitch','ReduceRpm','Test','Test2','Test3','Test4'}, ...
5      'sigNames',{ 'xTmComprCtrl_B_ChangeFinished',
6      'rTmComprCtrl_n_u16ComprRpmReq'})
```

代码块

```

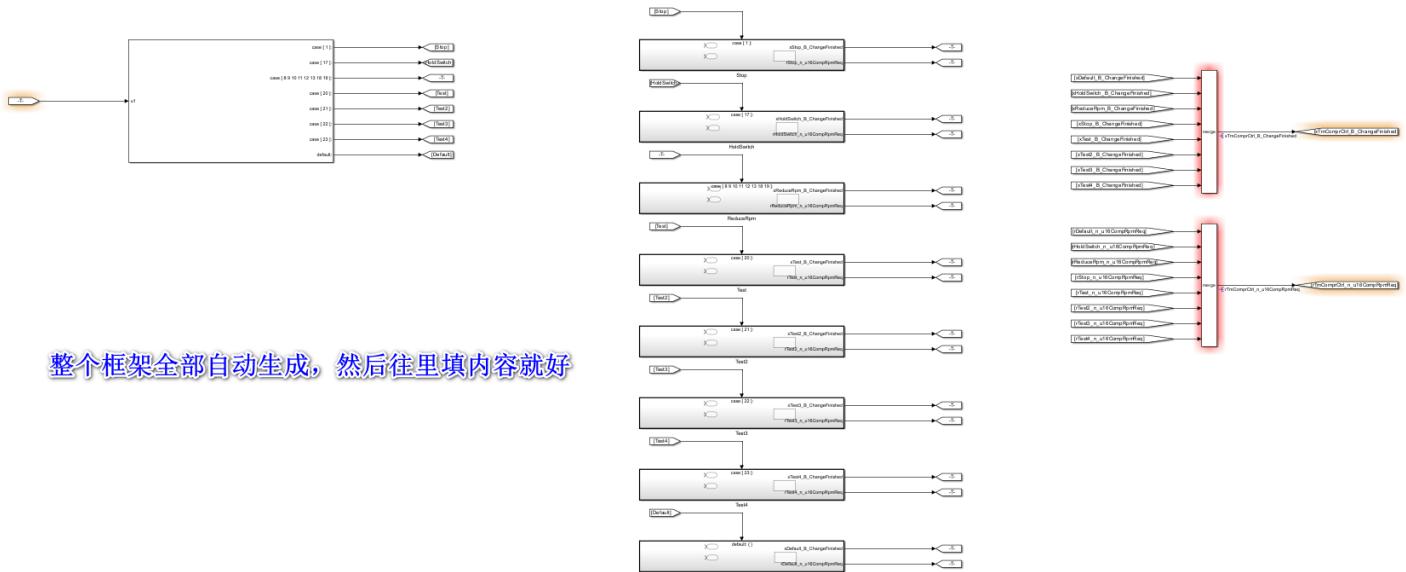
1  %CREATESWITCHCASE 为当前路径下的模块创建Switch Case结构
2  %  createSwitchCase() 自动创建Switch Case模块及其相关的Action子系统。
3  %
4  %  可选参数：
5  %      'caseList' - Case的输入值，默认为'{1,2,3,[4 5 6]}'
6  %      'caseIn' - Case输入信号，默认为'Demo'
7  %      'caseNames' - Case具体名称列表，默认为一组压缩机控制信号
8  %      'sigNames' - Switch的输出信号，默认为{'demo1','demo2'}
9  %      'actionPosYShift' - 每个case的垂直偏移值，默认为50
10 %
11 %      'caseGap' - Switch case的间距，默认为30
11 %      'caseStp' - Switch case间距放大系数，默认为1
```

```

12 %      'resovleSig' - 是否需要解析merge后的信号，默认为true
13 %
14 %    示例：
15 %      createSwitchCase()
16 %      createSwitchCase('caseIn', 'Demo', 'caseList', '{1,2,3,[4 5 6]}', ...
17 %                      'caseNames', {'A','B','C','D'}, 'sigNames', {'demo1','demo2'})
18 %
19 %    作者：葛维冬 (Blue Ge)
20 %    日期：2024-05-17
21 %    版本：1.1

```

执行效果



创建所有子模型对应的goto,from

当一个模块中的多个子模型搭建完成后，可以直接为各个模块创建对应的goto, from, 后续可以根据这些goto from, 创建输入输出端口，当软可以迭代向顶层创建

使用说明

代码块

```
1 createModGotoAll()
```

代码块

```

1 function [createdInputAll, createdOutputAll] = createModGotoAll(varargin)
2 %CREATEMODGOTOALL 为当前模型层级中的所有模块创建输入输出Goto/From模块
3 % [createdInputAll, createdOutputAll] = createModGotoAll() 为当前模型层级中的

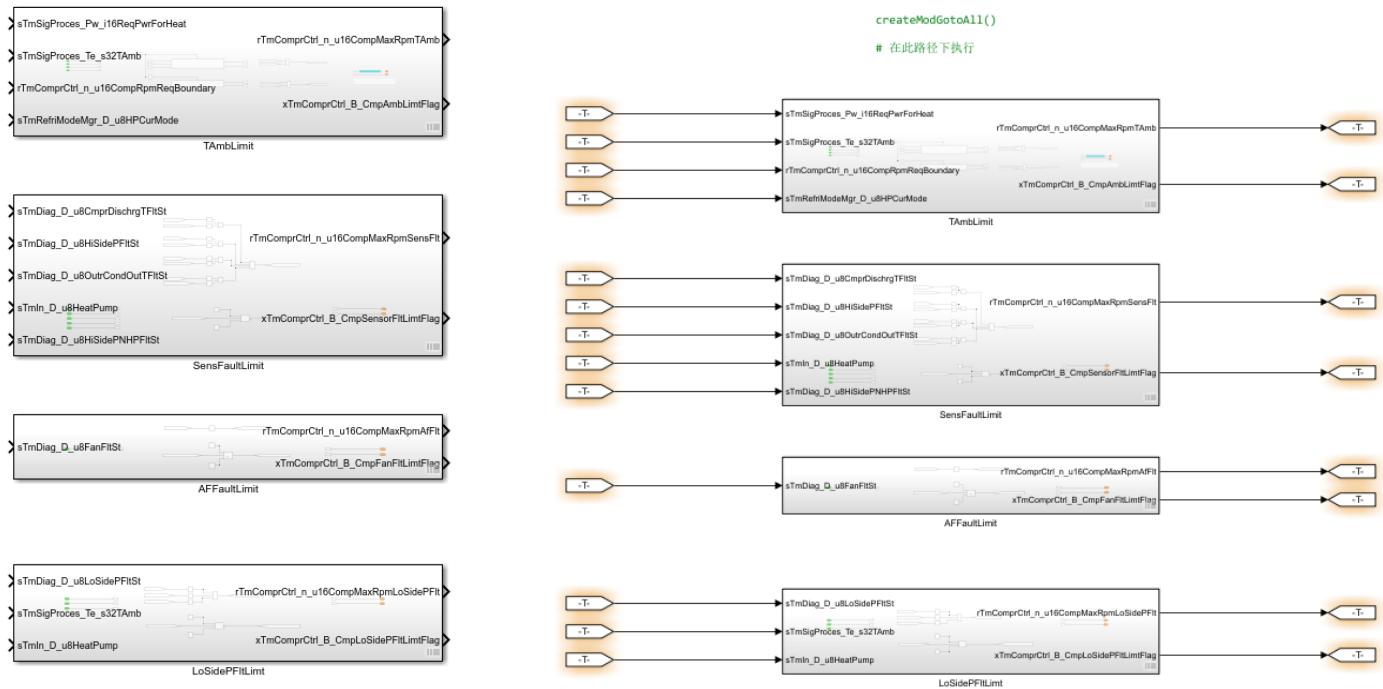
```

```

4 % 所有模块或引用模型创建输入输出Goto/From模块。
5 %
6 % 可选参数：
7 %     'mode' - 创建模式，可选值为'both'（默认）、'input'或'output'
8 %     'inList' - 限制创建的输入信号列表，默认为空
9 %     'outList' - 限制创建的输出信号列表，默认为空
10 %
11 % 输出参数：
12 %         createdInputAll - 成功创建的输入信号列表
13 %         createdOutputAll - 成功创建的输出信号列表
14 %
15 % 示例：
16 % [in, out] = createModGotoAll()
17 % [in, out] = createModGotoAll('mode', 'input', 'inList', {'signal1'})
18 %
19 % 作者：葛维冬 (Blue Ge)
20 % 日期：2024-05-17
21 % 版本：1.1

```

执行效果



创建子模型相关输入输出

当一个子模型搭建完成后，往往需要给输入输出创建一些模块，比如

```

% 'inType' - 输入类型，可选'port'（默认）、'ground'、'from'、'const'、'none'
% 'outType' - 输出类型，可选'port'（默认）、'term'、'goto'、'disp'、'none'

```

这个脚本就是根据不同的配置，创建不同的模块

使用说明

代码块

```
1  createMode(gcb, ...
2      'inType','port', ...           % 可选参数 port, ground, from, const, none
3      'outType','port', ...          % 可选参数 port, term, goto, disp, none
4      ...
5      'add', 'None',...              % 可选参数 blockType, None, etc:
6      SignalConversion
7      ...
8      'bkHalfLength', 15,...        % 创建 goto from 相关配置
9      ...
10     'isEnableIn',true,...
11     'isEnableOut',true,...
12     'resoveValue',false,...
13     'logValue',false,...
14     'testValue',false,...
15     'dispName', false...
16     )
```

代码块

```
1  function [createdInput, createdOutput] = createMode(path, varargin)
2  %CREATemode 统一函数用于模型相关的批量操作
3  % [createdInput, createdOutput] = createMode(path) 对指定模型进行批量操作,
4  % 包括创建端口、Goto/From模块、接地等，同时支持信号解析。
5  %
6  % 输入参数:
7  %     path - 模型路径
8  %
9  % 可选参数:
10 %     基本参数:
11 %         'inType' - 输入类型, 可选'port' (默认)、'ground'、'from'、'const'、'none'
12 %         'outType' - 输出类型, 可选'port' (默认)、'term'、'goto'、'disp'、'none'
13 %         'wid' - 模型宽度, 默认为400
14 %
15 %     createModPorts参数:
16 %         'mode' - 模式, 默认为'both'
17 %         'suffixStr' - 后缀字符串, 默认为''
18 %         'findType' - 查找类型, 默认为'base'
19 %         'add' - 添加类型, 默认为'None'
```

```

20 %           'enFirstTrig' - 是否启用首次触发, 默认为false
21 %
22 %       信号解析参数:
23 %           'skipTrig' - 是否跳过触发, 默认为false
24 %           'isEnableIn' - 是否启用输入, 默认为true
25 %           'isEnableOut' - 是否启用输出, 默认为true
26 %           'resoveValue' - 是否解析值, 默认为true
27 %           'logValue' - 是否记录值, 默认为false
28 %           'testValue' - 是否测试值, 默认为false
29 %           'dispName' - 是否显示名称, 默认为true
30 %
31 %       createModGoto参数:
32 %           'inList' - 输入信号列表, 默认为{}
33 %           'outList' - 输出信号列表, 默认为{}
34 %           'bkHalfLength' - 模块半长, 默认为25
35 %
36 %   输出参数:
37 %       createdInput - 创建的输入信号列表
38 %       createdOutput - 创建的输出信号列表
39 %
40 %   示例:
41 %       [in, out] = createMode(gcb)
42 %       [in, out] = createMode(gcb, 'inType', 'ground', 'outType', 'port') % 可选'port' (默认) 、 'ground'、 'from'、 'const'、 'none'
43 %
44 %   作者: 葛维冬 (Blue Ge)
45 %   日期: 2024-01-26
46 %   版本: 1.1

```

执行效果

代码块

```

1  createMode(gcb, ...
2      'inType', 'port', ...          % 可选参数 port, ground, from, const, none
3      'outType', 'port', ...        % 可选参数 port, term, goto, disp, none
4      ...                          % 创建 ports 相关配置
5      'add', 'None',...            % 可选参数 blockType, None, etc:
SignalConversion
6      ...                          % 创建 goto from 相关配置
7      'bkHalfLength', 15,...       % 创建 信号解析 相关配置
8      ...
9      'isEnableIn', true,...      ...
10     'isEnableOut', true,...     ...
11     'resoveValue', true,...     ...
12     'logValue', false,...      ...

```

```

13     'testValue', false, ...
14     'dispName', false...
15 )

```

代码块

```

1  createMode(gcb, ...
2      'inType', 'port', ...           % 可选参数 port, ground, from, const, none
3      'outType', 'port', ...         % 可选参数 port, term, goto, disp, none
4      ...
5      'add', 'None', ...            % 创建 ports 相关配置
6      ...
7      'bkHalfLength', 15, ...       % 可选参数 blockType, None, etc:
8      ...
9      'isEnableIn', true, ...       % 创建 goto from 相关配置
10     'isEnableOut', true, ...      % 创建 信号解析 相关配置
11     'resoveValue', true, ...
12     'logValue', false, ...
13     'testValue', false, ...
14     'dispName', false...
15 )

```



代码块

```
1  createMode(gcb, ...
```

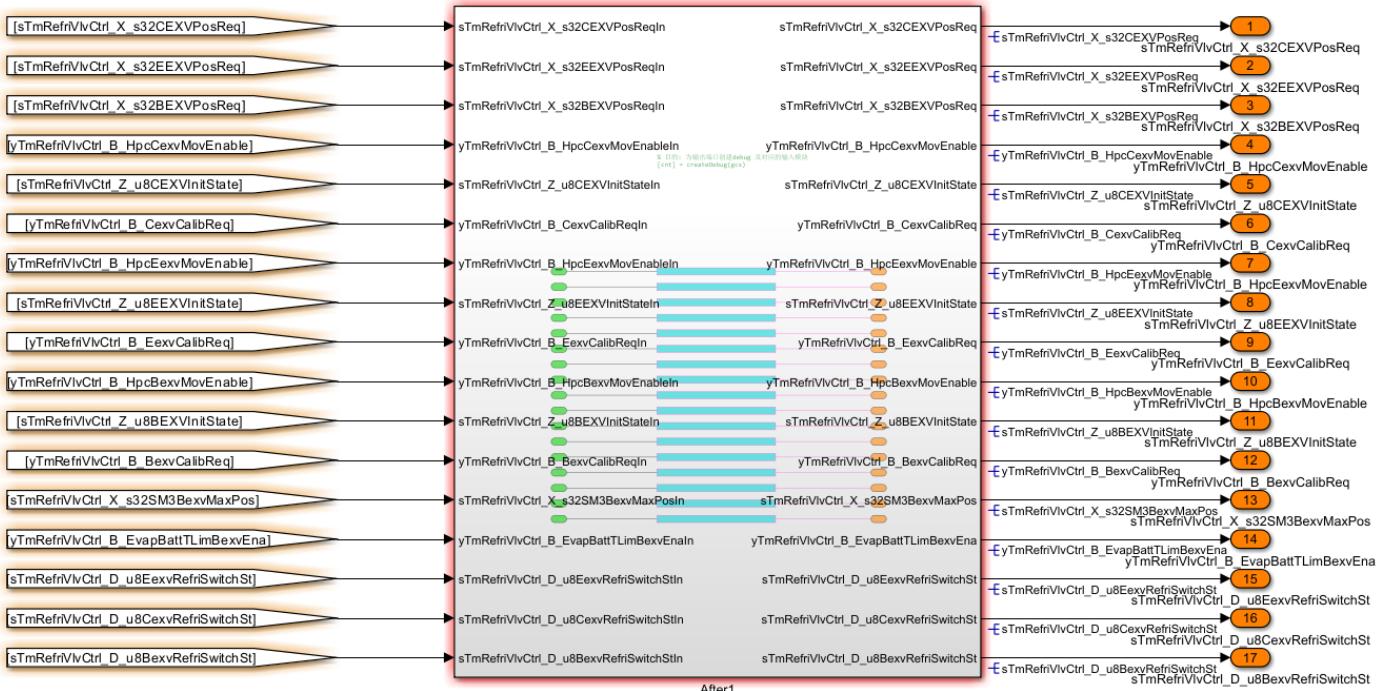
```

2   'inType', 'from', ...           % 可选参数 port, ground, from, const, none
3   'outType', 'port', ...          % 可选参数 port, term, goto, disp, none
4   'suffixStr', 'In', ...          % 创建 ports 相关配置
5   ...                            % 可选参数 blockType, None, etc:
6   'add', 'None', ...              % 可选参数 blockType, None, etc:

SignalConversion

7   ...                            % 创建 goto from 相关配置
8   'bkHalfLength', 125,...       % 创建 信号解析 相关配置
9   ...
10  'isEnableIn', false,...       % 创建 信号解析 相关配置
11  'isEnableOut', true,...      % 创建 信号解析 相关配置
12  'resoveValue', true,...      % 创建 信号解析 相关配置
13  'logValue', false,...        % 创建 信号解析 相关配置
14  'testValue', false,...       % 创建 信号解析 相关配置
15  'dispName', false...         % 创建 信号解析 相关配置
16 )

```



```

1  createMode(gcb, ...
2    'inType', 'const', ...           % 可选参数 port, ground, from, const, none
3    'outType', 'disp', ...          % 可选参数 port, term, goto, disp, none
4    'suffixStr', 'In', ...          % 创建 ports 相关配置
5    ...                            % 可选参数 blockType, None, etc:
6    'add', 'None', ...              % 可选参数 blockType, None, etc:

SignalConversion

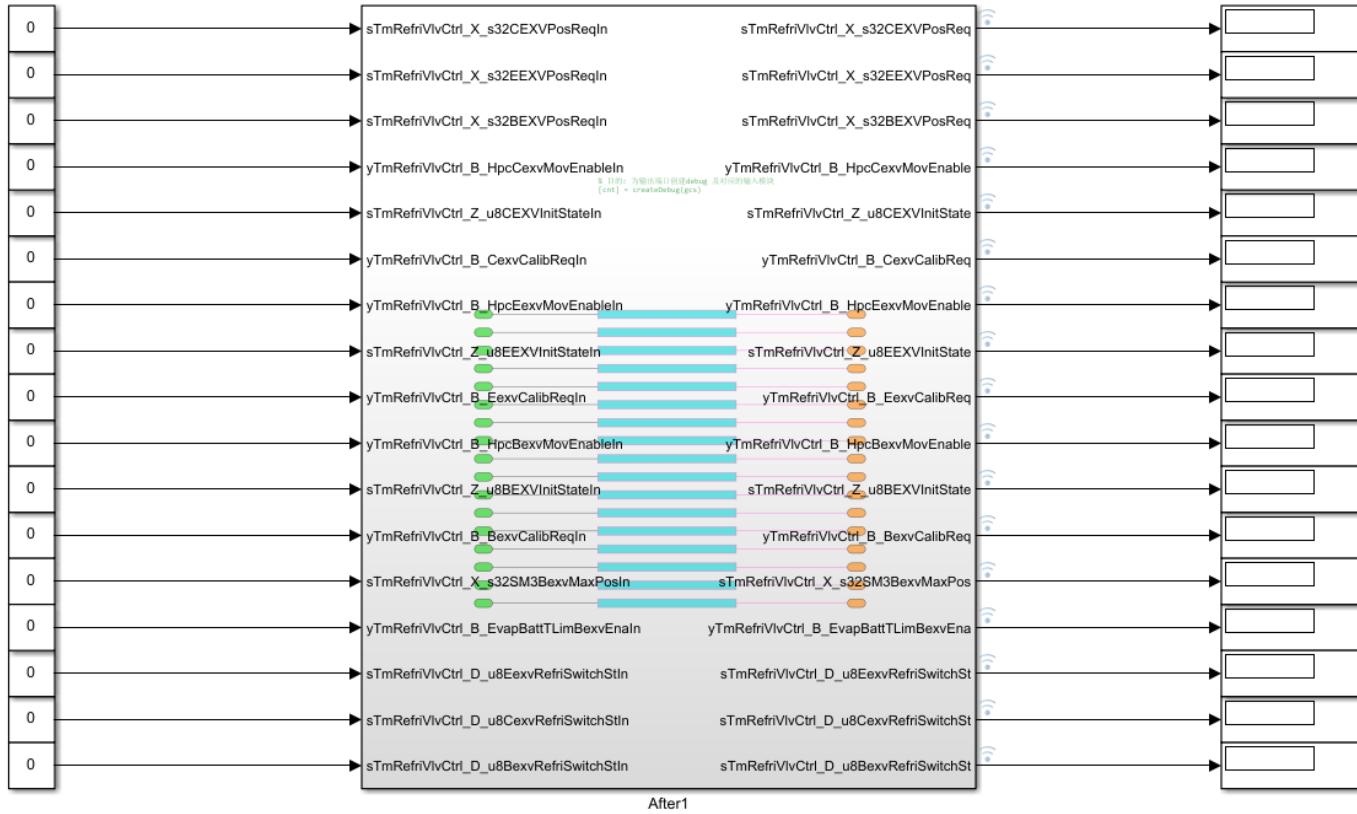
7   ...                            % 创建 goto from 相关配置

```

```

8     'bkHalfLength', 125,...           % 创建 信号解析 相关配置
9     ...
10    'isEnableIn',false,...           % 可选参数 port, ground, from, const, none
11    'isEnableOut',true,...           % 可选参数 port, term, goto, disp, none
12    'resoveValue',false,...           % 创建 ports 相关配置
13    'logValue',true,...             % 可选参数 blockType, None, etc:
14    'testValue',false,...           % 创建 goto from 相关配置
15    'dispName', false...           % 创建 信号解析 相关配置
16 )

```



代码块

```

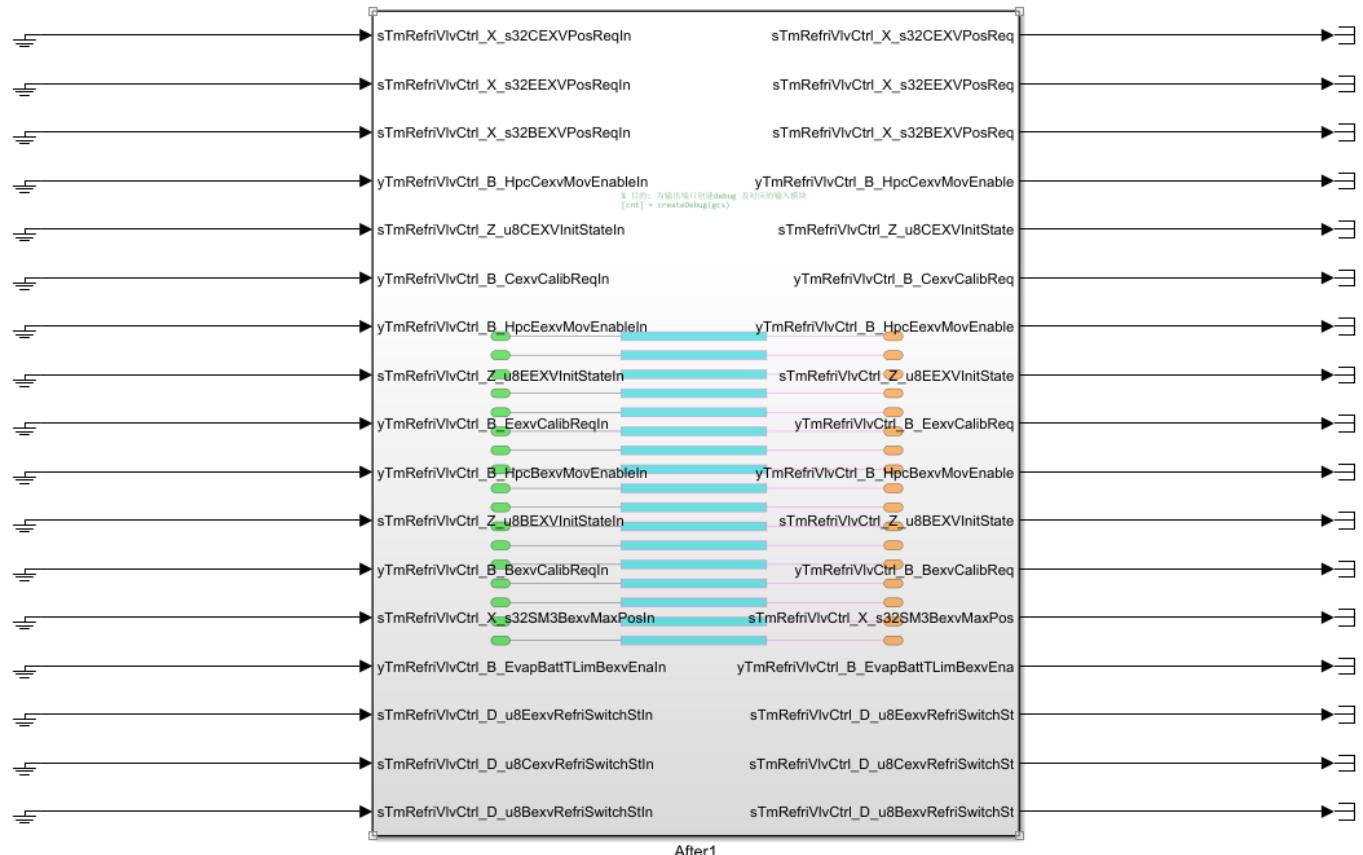
1  createMode(gcb, ...
2      'inType','ground', ...           % 可选参数 port, ground, from, const, none
3      'outType','term', ...           % 可选参数 port, term, goto, disp, none
4      'suffixStr','In',...           % 创建 ports 相关配置
5      ...                            % 可选参数 blockType, None, etc:
6      'add', 'None',...               % 创建 goto from 相关配置
7      SignalConversion
8      ...
9      'bkHalfLength', 125,...         % 创建 信号解析 相关配置
10     ...
11     'isEnableIn',true,...           % 创建 信号解析 相关配置

```

```

11     'isEnableOut', true, ...
12     'resoveValue', false, ...
13     'logValue', false, ...
14     'testValue', false, ...
15     'dispName', false...
16 )

```



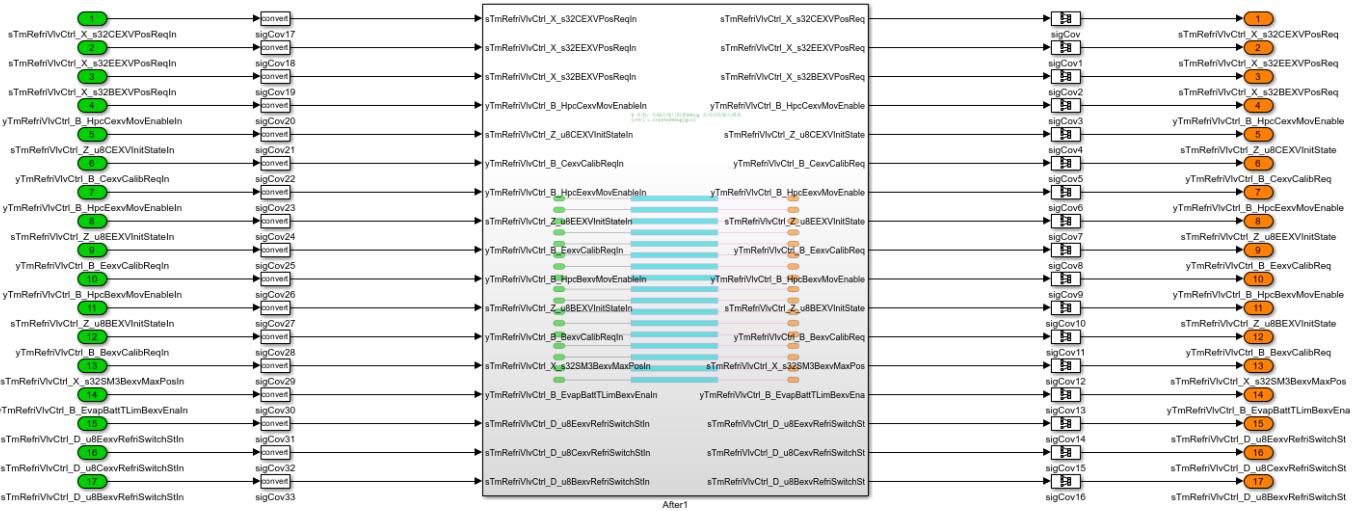
代码块

```

1  createMode(gcb, ...
2      'inType','none', ...           % 可选参数 port, ground, from, const, none
3      'outType','port', ...         % 可选参数 port, term, goto, disp, none
4      'suffixStr','','...',        % 创建 ports 相关配置
5      ...                           % 可选参数 blockType, None, etc:
6      'add', 'SignalConversion',...  % 可选参数 blockType, None, etc:
7          ...                         % 创建 goto from 相关配置
8          'bkHalfLength', 125,...       % 创建 信号解析 相关配置
9          ...
10     'isEnableIn',true,...          ...
11     'isEnableOut',true,...         ...
12     'resoveValue',false,...        ...
13     'logValue',false,...          ...

```

```
14     'testValue',false,...
15     'dispName', false...
16 )
17
18 createMode(gcb, ...
19     'inType','port', ...           % 可选参数 port, ground, from, const, none
20     'outType','none', ...         % 可选参数 port, term, goto, disp, none
21     'suffixStr','',...
22     ...                         % 创建 ports 相关配置
23     'add', 'DataTypeConversion',...          % 可选参数 blockType, None, etc:
24
25     SignalConversion (DataTypeConversion
26
27     ...                         % 创建 goto from 相关配置
28     'bkHalfLength', 125,...
29
30     ...                         % 创建 信号解析 相关配置
31     'isEnableIn',true,...
32     'isEnableOut',true,...
33     'resoveValue',false,...
34     'logValue',false,...
35     'testValue',false,...
36     'dispName', false...
37 )
38 )
```



根据模板，创建输入输出端口

如果清楚一个模型的接口信号，可以将其整理成表格，然后直接导入创建

使用说明

```
1 % 目的：对模型输入输出端口的信号线进行解析
2 [inCnt, outCnt, inCntDel, outCntDel] = createPorts('Template.xlsx', gcs)
```

代码块

```
1 function [inCnt, outCnt, inCntDel, outCntDel] = createPorts(template,
2 portPath, varargin)
3 %CREATEPORTS 从Excel模板中提取信息并创建输入输出端口
4 % 根据Excel模板中的信息在指定路径下创建输入输出端口。
5 % 其中有一列是Action, 如果这列对应的数据是0, 则表示删除该端口, 反之则新建
6 %
7 % 输入参数:
8 %     template - Excel模板文件路径
9 %     portPath - 端口创建路径
10 %
11 % 可选参数:
12 %     'step' - 端口间距, 默认为30
13 %     'posIn' - 输入端口起始位置, 默认为[0, 0]
14 %     'posOut' - 输出端口起始位置, 默认为[1000, 0]
15 %     'inRecognName' - 输入端口识别名称, 默认为'Input'
16 %     'outRecognName' - 输出端口识别名称, 默认为'Output'
17 %     'sheet' - Excel工作表名称, 默认为'Signals'
18 %
19 % 输出参数:
20 %     inCnt - 创建的输入端口数量
21 %     outCnt - 创建的输出端口数量
22 %     inCntDel - 删除的输入端口数量
23 %     outCntDel - 删除的输出端口数量
24 %
25 % 示例:
26 %     [inCnt, outCnt, inCntDel, outCntDel] = createPorts('Template.xlsx',
27 %     gcs)
28 %     [inCnt, outCnt, inCntDel, outCntDel] = createPorts('Template.xlsx',
29 %     gcs, 'step', 40)
30 %
31 % 作者: 葛维冬 (Blue Ge)
32 % 日期: 2024-03-20
33 % 版本: 1.1
```

执行效果

代码块

```
1 [inCnt, outCnt, inCntDel, outCntDel] = createPorts('Template.xlsx', gcs)
```



根据模板，创建输入输出端口及对应的goto

如果清楚一个模型的接口信号，可以将其整理成表格，然后直接导入创建

使用说明

代码块

```

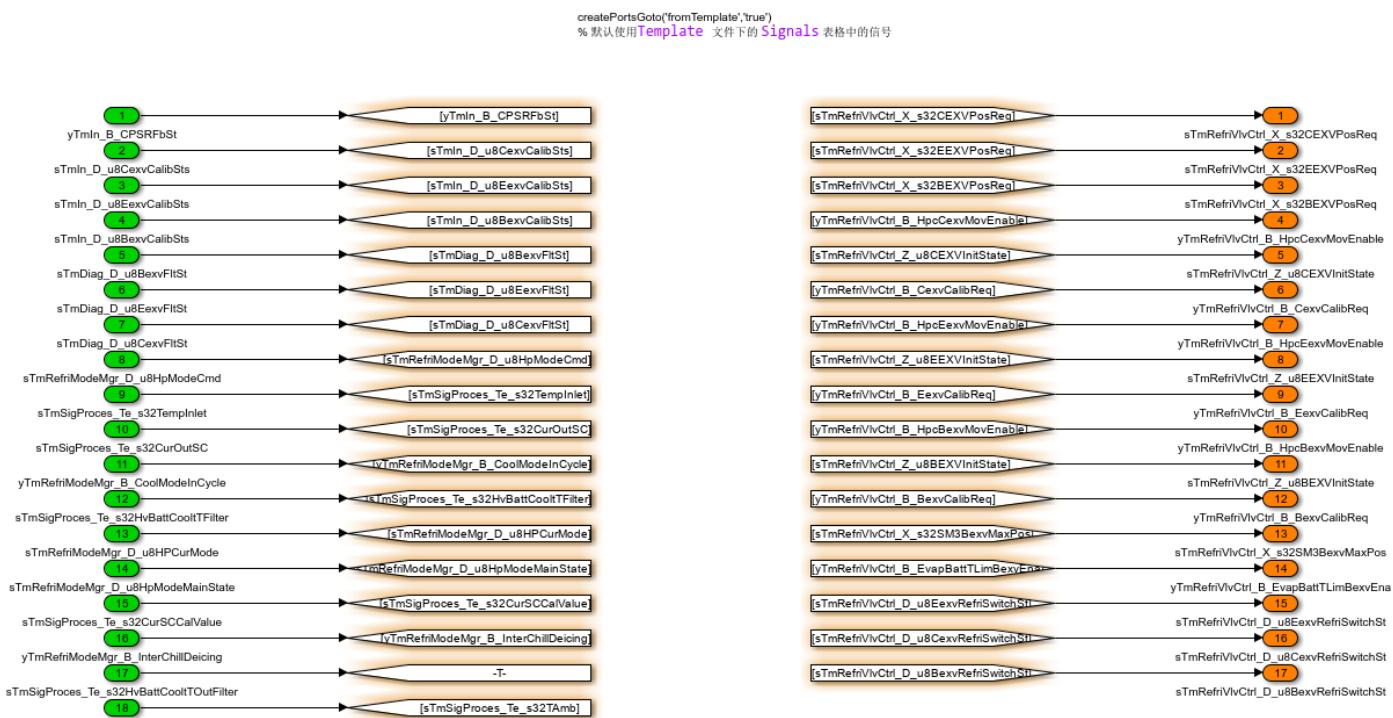
1 function [numInputPorts, numOutputPorts] = createPortsGoto(varargin)
2 %CREATEPORTSGOTO 创建输入输出端口及其相连接的Goto/From模块
3 % [numInputPorts, numOutputPorts] = createPortsGoto() 根据指定的信号列表或模板文
4 % 件
5 %
6 % 可选参数：
7 % 'posIn' - 输入端口起始位置，默认为[-500 0]
8 % 'posOut' - 输出端口起始位置，默认为[500 0]
9 % 'step' - 端口间距，默认为30
10 % 'gotoLength' - Goto/From模块宽度，默认为180
11 % 'inList' - 输入信号列表，默认为{}
12 % 'outList' - 输出信号列表，默认为{}
13 % 'NAstr' - 忽略的字段标识，默认为'NA'
14 % 'mode' - 命名模式，可选'pre'、'tail'、'keep'，默认为'keep'
15 % 'fromTemplate' - 是否使用模板文件，默认为false
16 % 'template' - 模板文件名，默认为'Template'
```

```
17 %      'sheet' - 模板工作表名, 默认为 'Signals'
18 %      'inRecognName' - 输入信号识别字段, 默认为 'Input'
19 %      'outRecognName' - 输出信号识别字段, 默认为 'Output'
20 %
21 %  输出参数:
22 %      numInputPorts - 创建的输入端口数量
23 %      numOutputPorts - 创建的输出端口数量
24 %
25 %  示例:
26 %      [in, out] = createPortsGoto('fromTemplate', true)
27 %      [in, out] = createPortsGoto('inList', {'a', 'b'}, 'outList', {'c', 'd'})
28 %
29 %  作者: 葛维冬 (Blue Ge)
30 %  日期: 2023-09-28
31 %  版本: 1.1
```

执行效果

代码块

```
1 % 目的：对模型输入输出端口的信号线进行解析  
2 [in, out] = createPortsGoto('fromTemplate', true)
```



代码块

```
1 [in, out] = createPortsGoto('inList', {'a','b'}, 'outList', {'c','d'})
```



根据没有匹配的goto, from 创建另外一半

痛点：如果我们使用goto, from完成内部小模型的搭建之后，有些goto,from删除了，但期望先保留使用，因此，需要根据没有匹配的的goto, from 创建对应的goto,from

使用说明

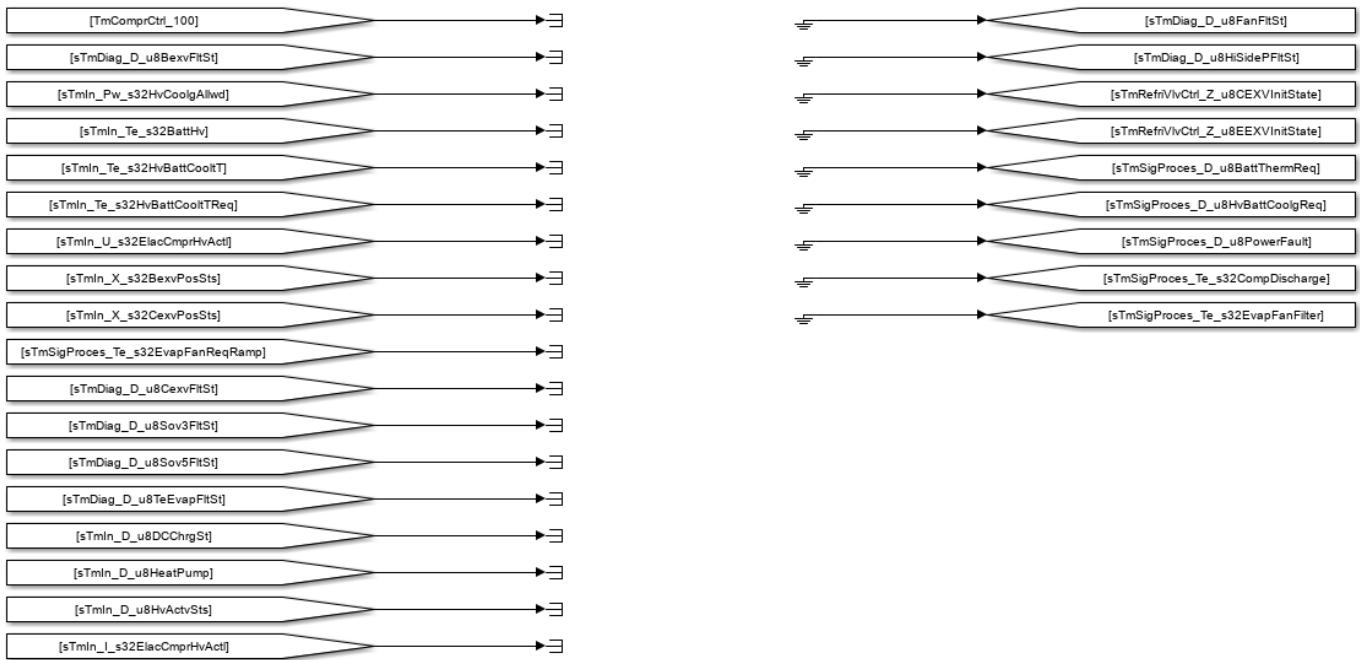
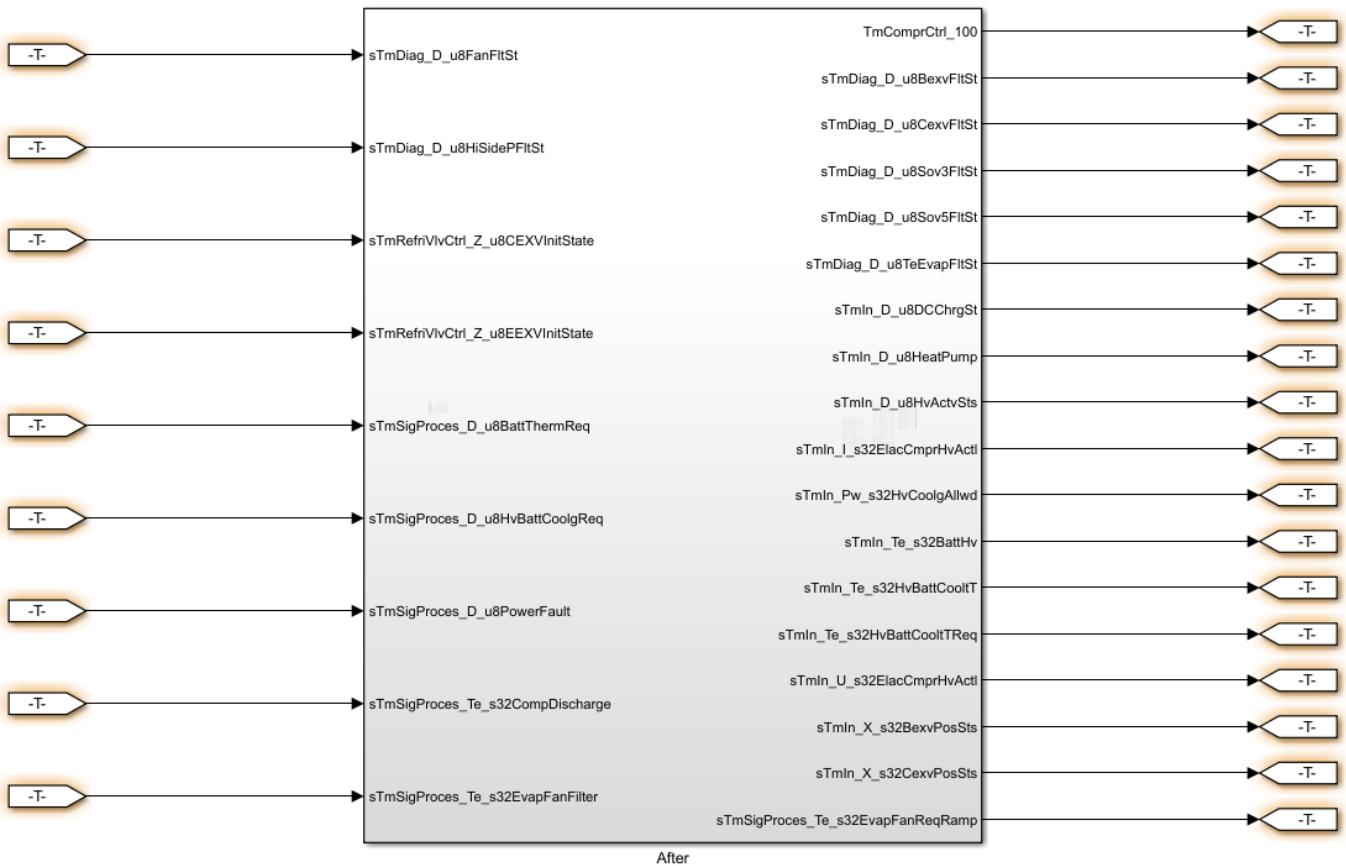
代码块

```

1  function [numGoto, numFrom] = createGotoUseless(varargin)
2  %CREATEGOTOUSELESS 为无用的Goto和From模块创建匹配模块
3  % [numGoto, numFrom] = createGotoUseless() 为模型中未匹配的Goto和From模块
4  % 创建对应的匹配模块，并连接到Ground或Terminator。
5 %
6 % 可选参数：
7 %     'posGotoBase' - Goto模块的起始位置，默认为[12500,0]
8 %     'posFromBase' - From模块的起始位置，默认为[12000,0]
9 %     'step' - 模块之间的垂直间距，默认为30
10 %    'createGoto' - 是否创建Goto模块，默认为true
11 %    'createFrom' - 是否创建From模块，默认为true
12 %
13 % 输出参数：
14 %     numGoto - 创建的Goto模块数量
15 %     numFrom - 创建的From模块数量
16 %
17 % 示例：
18 %     [numGoto, numFrom] = createGotoUseless()
19 %     [numGoto, numFrom] = createGotoUseless('posGotoBase', [14000,0],
20 %     'step', 40)
21 %     [numGoto, numFrom] = createGotoUseless('createGoto', false) % 只创建From
22 %     模块
23 %
24 % 作者: 葛维冬 (Blue Ge)
25 % 日期: 2024-06-28
26 % 版本: 1.0

```

执行效果



根据没有匹配的goto, from 创建输入输出端口

痛点：如果我们使用goto, from完成内部小模型的搭建之后，需要将这些没有匹配的goto,from通过输入输出端口的形式输入输出，因此，需要根据用到的goto, from 创建对应的端口及匹配的goto,from

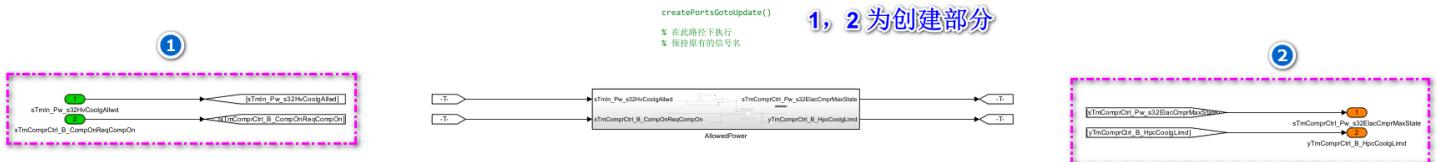
使用说明

[inSignals, outSignals] = createPortsGotoUpdate()

代码块

```
1 function [inList, outList] = createPortsGotoUpdate(varargin)
2 %CREATEPORTSGOTOUPDATE 为无效的From/Goto信号创建对应的端口和信号
3 % [INLIST, OUTLIST] = CREATEPORTSGOTOUPDATE() 使用默认参数创建端口和信号
4 % [INLIST, OUTLIST] = CREATEPORTSGOTOUPDATE('Parameter', Value, ...) 使用指定
5 % 参数创建
6 %
7 % 输入参数 (名值对) :
8 %     'path'          - 目标系统路径 (字符串), 默认值: gcs (当前系统)
9 %     'wid'           - 模型宽度 (正整数), 默认值: 400
10 %    'mode'          - 信号名称处理模式 (字符串), 默认值: 'keep'
11 %                      可选值: 'pre'(保留前缀), 'tail'(保留后缀), 'keep'(保持原样)
12 %
13 % 输出参数:
14 %     inList         - 创建的输入信号名称列表 (元胞数组)
15 %     outList        - 创建的输出信号名称列表 (元胞数组)
16 %
17 % 功能描述:
18 %     1. 查找系统中无效的From块 (没有对应的Goto)
19 %     2. 为这些From块创建对应的输入端口和Goto块
20 %     3. 查找系统中无效的Goto块 (没有对应的From)
21 %     4. 为这些Goto块创建对应的输出端口
22 %     5. 调整模型大小以适应新创建的端口
23 %
24 % 示例:
25 %     [inSignals, outSignals] = createPortsGotoUpdate()
26 %     [inSignals, outSignals] = createPortsGotoUpdate('path',
27 %     'myModel/subsystem1')
28 %
29 % 注意事项:
30 %     1. 使用前需要打开目标Simulink模型
31 %     2. 函数会自动调整模型大小以适应新创建的端口
32 %
33 % 参见: CREATEPORTSGOTO, FINDUSELESSFROM, FINDUSELESSGOTO, CHANGEMODSIZE
34 %
35 % 作者: Blue.ge
36 % 版本: 1.0
37 % 日期: 20231114
```

执行效果



迭代创建新增信号

痛点：当我们模型搭建完成后，如果后续新增了一个信号，如果这个信号处于比较深的模型，那么其所有上层模型都得逐层添加这个信号，特别麻烦

此脚本可以为所有层自动添加对应的信号

使用说明

[inList, outList] = createPortsGotoUpdateLoop('topLev', 'After')

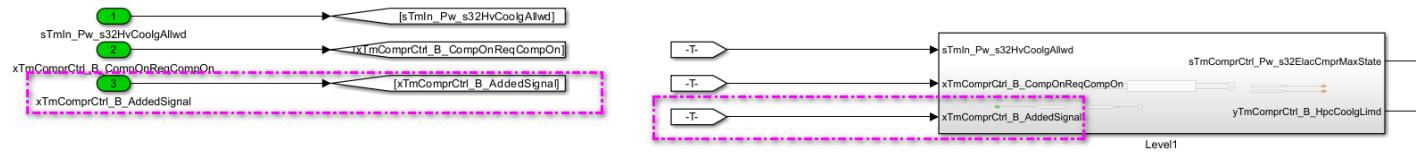
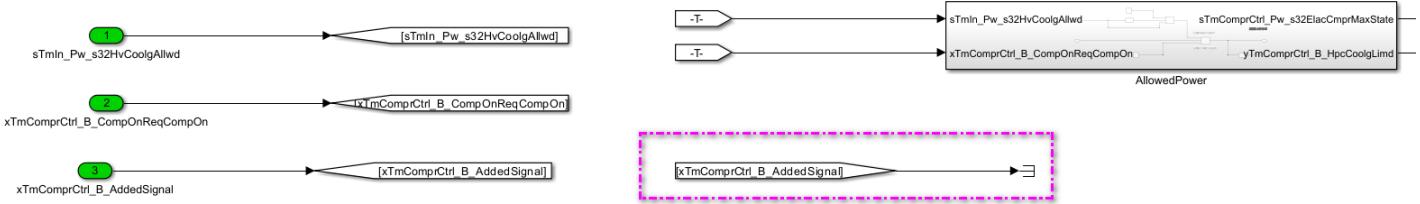
代码块

```

1 function [inList, outList] = createPortsGotoUpdateLoop(varargin)
2 %CREATEPORTSGOTOUPDATELOOP 循环创建From模块对应的Inport和Goto模块
3 % [inList, outList] = createPortsGotoUpdateLoop() 当系统使用From方式新增信号后,
4 % 循环创建对应的Inport和Goto模块, 直到根路径。
5 %
6 % 可选参数:
7 %     'path' - 当前路径, 默认为gcs
8 %     'topLev' - 循环结束的模型层, 默认为bdroot
9 %     'resolveValue' - 是否解析信号, 默认为true
10 %
11 % 输出参数:
12 %     inList - 创建的输入信号列表
13 %     outList - 创建的输出信号列表
14 %
15 % 示例:
16 %     [inList, outList] = createPortsGotoUpdateLoop() % 迭代循环一直到顶层
17 %     [inList, outList] = createPortsGotoUpdateLoop('topLev', 'After')
18 %
19 % 作者: 葛维冬 (Blue Ge)
20 % 日期: 2023-11-14
21 % 版本: 1.1

```

执行效果



创建输出输出端口的信号解析，记录

为了方便测试，我们需要经常在输出输出信号创建解析，便于更好的观察

使用说明

代码块

```

1 function createSigOnLine(pathMd, varargin)
2 %CREATESIGONLINE 在模型输入输出端口的信号线上进行信号设置
3 % createSigOnLine(pathMd) 对指定模型路径下的输入输出端口信号线进行设置,
4 % 包括信号名称、解析设置、数据记录和测试点等。
5 %
6 % 输入参数:
7 %     pathMd - 模型路径
8 %
9 % 可选参数:
10 %     'skipTrig' - 是否跳过触发信号, 默认为false
11 %     'isEnableIn' - 是否处理输入端口, 默认为true
12 %     'isEnableOut' - 是否处理输出端口, 默认为true
13 %     'resovleValue' - 是否解析为Simulink信号对象, 默认为false
14 %     'logValue' - 是否记录数据, 默认为false
15 %     'testValue' - 是否设置为测试点, 默认为false
16 %
17 % 示例:
18 %     createSigOnLine(gcs)

```

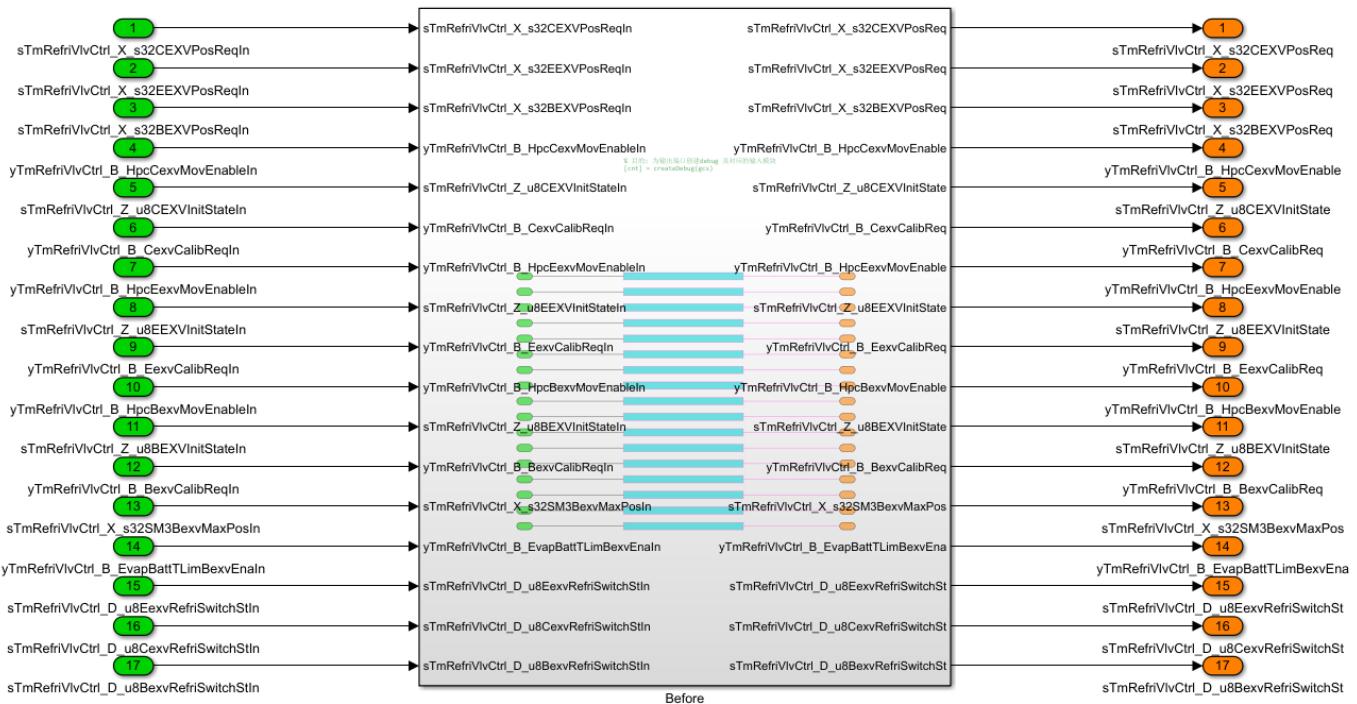
```

19 %      createSigOnLine(gcs, 'skipTrig', true, 'isEnableIn', true,
20 %                           'isEnableOut', true, ...
21 %
22 %   作者: 葛维冬 (Blue Ge)
23 %   日期: 2023-09-05
24 %   版本: 1.1

```

执行效果

执行代码前



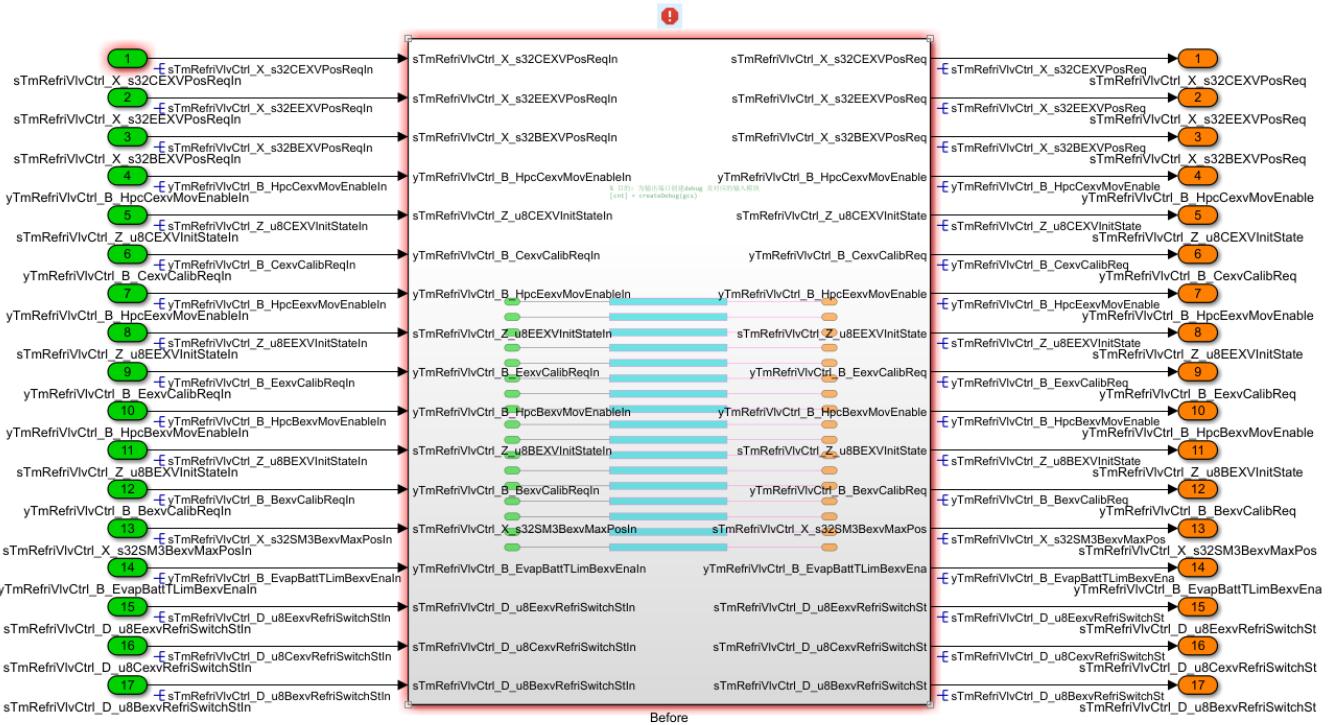
创建解析信号

代码块

```

1 createSigOnLine(gcs)
2 createSigOnLine(gcs, 'resoveValue', true)

```



创建记录信号

代码块

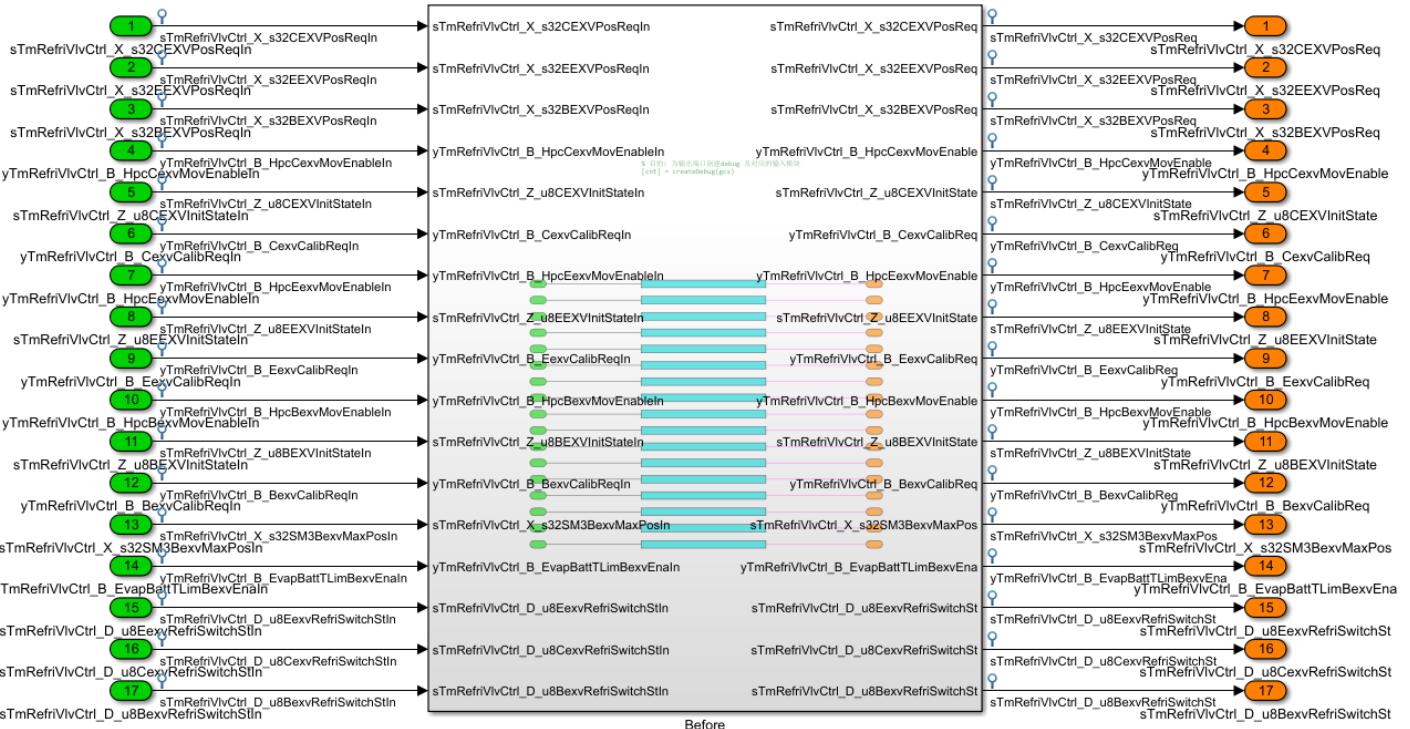
```
1   createSigOnLine(gcs, 'resoveValue', false, 'logValue', true)
```



创建测试点

代码块

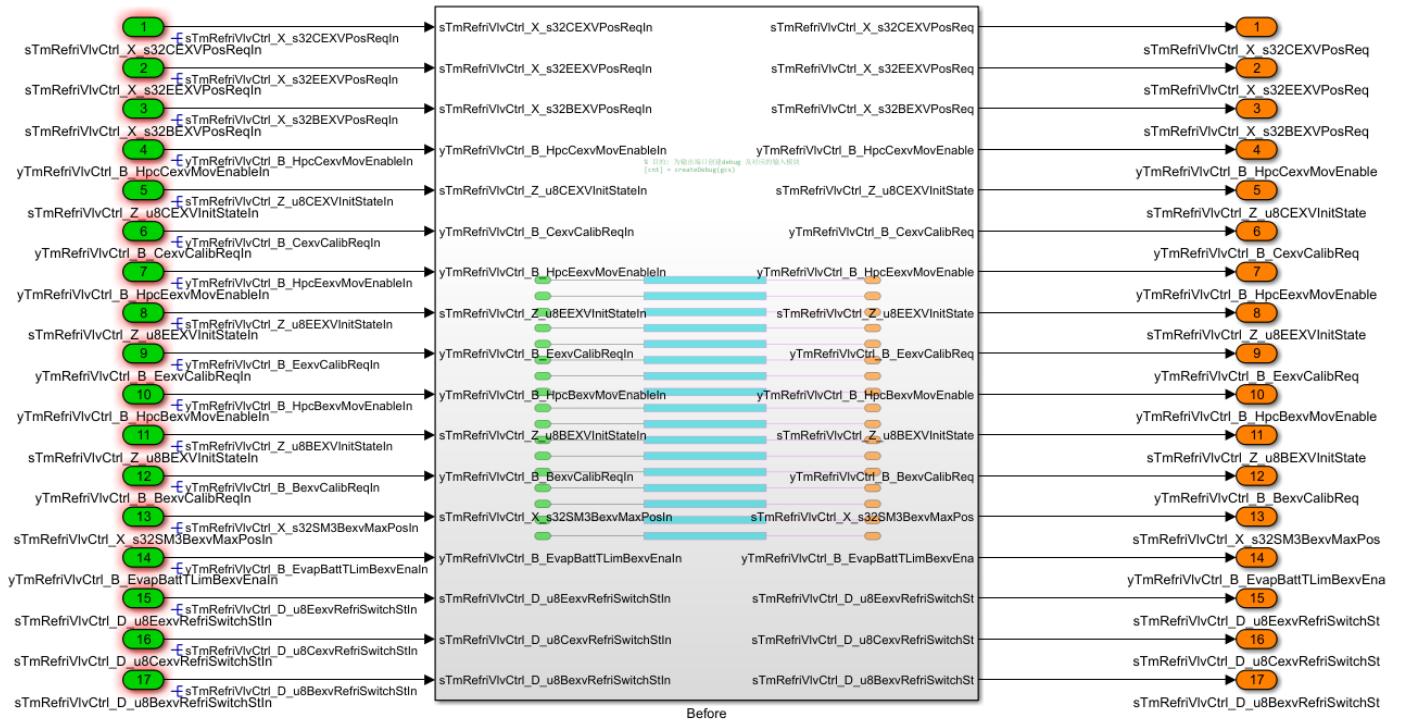
```
1   >> createSigOnLine(gcs, 'resoveValue', false, 'testValue', true)
```



创建单边信号解析

代码块

```
1 createSigOnLine(gcs, 'resovleValue', true, 'isEnableIn', true, 'isEnableOut', false)
```



创建slld并绑定模型

目前热系统是使用excel 表格来维护sldd 的，如果有必要，可以使用matlab 自带的sldd直接绑定模型，那样仿真的时候就不需要导入表格中的变量了。

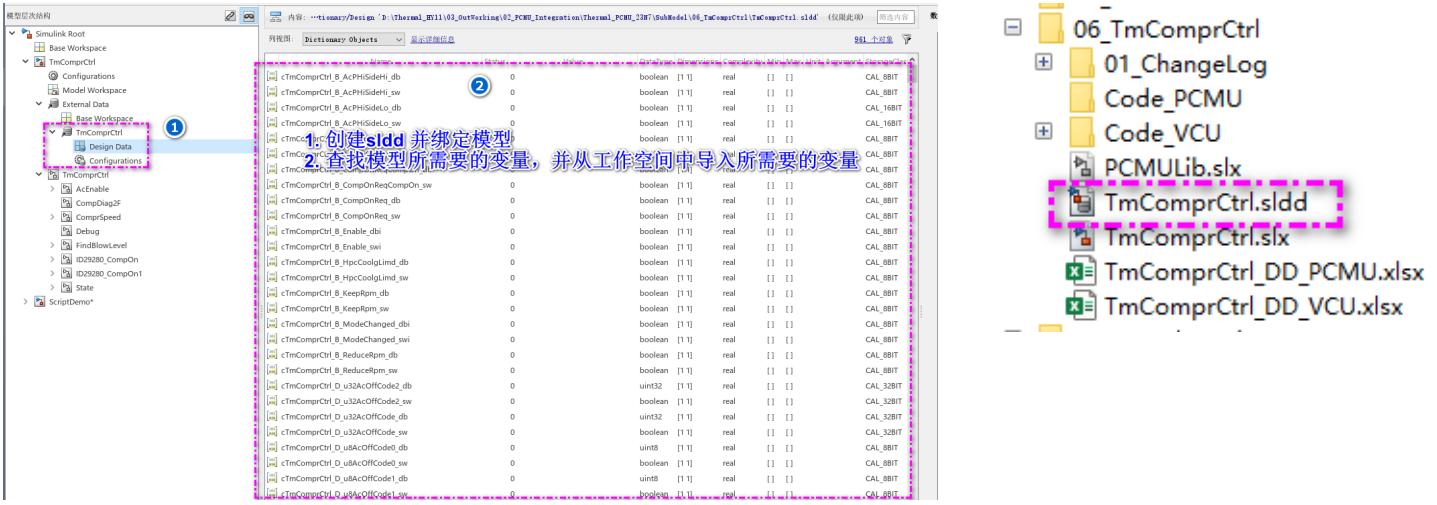
使用说明

代码块

```
1 function dictionaryObj = createSlddDesignData(modelName, varargin)
2 %CREATESLDDDESIGNDATA 创建或更新模型的数据字典
3 %    dictionaryObj = createSlddDesignData(modelName) 根据模型名称创建或更新
4 %    对应的数据字典，将工作空间中的变量导入到数据字典中。
5 %
6 %    输入参数：
7 %        modelName - 模型名称
8 %
9 %    可选参数：
10 %        'slddPath' - 数据字典路径，默认为空
11 %
12 %    输出参数：
13 %        dictionaryObj - 数据字典对象
14 %
15 %    示例：
16 %        dictionaryObj = createSlddDesignData('TmComprCtrl')
17 %        dictionaryObj = createSlddDesignData('TmComprCtrl', 'slddPath',
18 %        'path/to/dictionary.sldd')
19 %
20 %    说明：
21 %        1. 需要先成功编译模型
22 %        2. 所有需要导入的变量必须已加载到工作空间中
23 %
24 %    作者：葛维冬 (Blue Ge)
25 %    日期：2024-06-28
26 %    版本：1.1
```

执行效果

执行后会在目录中生成一个sldd 文件，并绑定模型，导入所需要的变量



如果不需导入数据，可以直接使用如下函数创建并绑定slld，区别就是如下脚本不要求模型必须仿真通过

```
dictionaryObj = createSlld('TmComprCtrl')
```

代码块

```

1 function dictionaryObj = createSlld(modelName, varargin)
2 %CREATESLDD 创建或打开模型的数据字典
3 % dictionaryObj = createSlld(modelName) 根据模型名称创建或打开对应的数据字典,
4 % 并将其绑定到模型。如果未指定数据字典路径，则数据字典将位于模型同目录下。
5 %
6 % 输入参数:
7 %     modelName - 模型名称
8 %
9 % 可选参数:
10 %     'slldPath' - 数据字典路径, 默认为空
11 %
12 % 输出参数:
13 %     dictionaryObj - 数据字典对象
14 %
15 % 示例:
16 %         dictionaryObj = createSlld('TmComprCtrl')
17 %         dictionaryObj = createSlld('TmComprCtrl', 'slldPath',
18 %             'path/to/dictionary.slld')
19 %
20 % 作者: 葛维冬 (Blue Ge)
21 % 日期: 2024-06-28
22 % 版本: 1.1

```

创建Memory 模块

如果系统中需要使用memory 模块，可以使用如下脚本创建，创建是基于excel 模板的

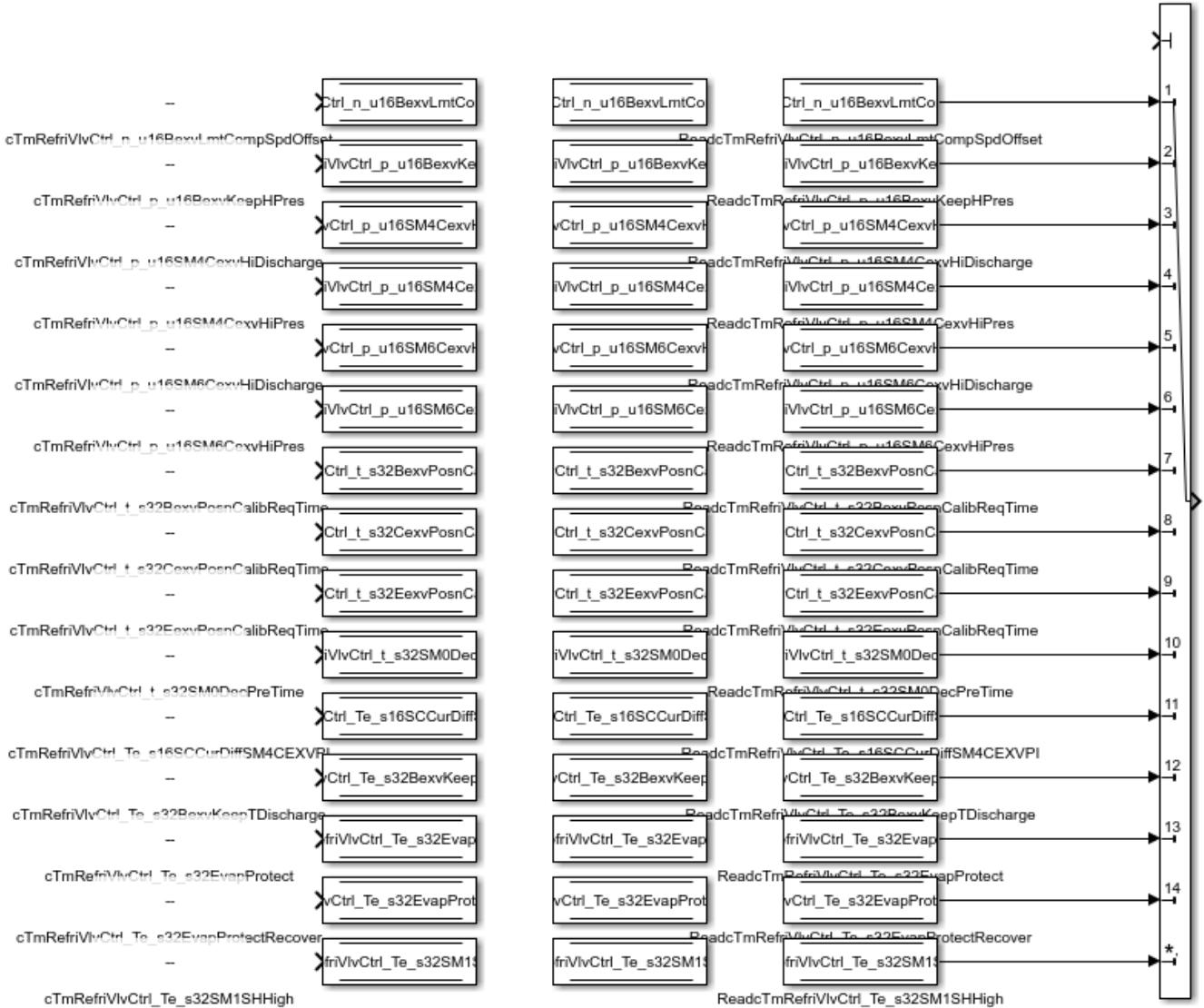
使用说明

代码块

```
1 function cnt = createStoreMem(path, template, varargin)
2 %CREATESTOREMEM 根据Excel模板批量创建Data Store Memory
3 % cnt = createStoreMem(path, template) 根据指定的Excel模板在模型中批量创建
4 % Data Store Memory、Data Store Write和Data Store Read模块。
5 %
6 % 输入参数：
7 %     path - 模型路径
8 %     template - Excel模板文件名
9 %
10 % 可选参数：
11 %     'step' - 模块之间的垂直间距，默认为40
12 %     'posIn' - 起始位置坐标，默认为[0, 0]
13 %     'sheet' - Excel工作表名称，默认为'Parameters'
14 %     'halfLen' - 模块宽度的一半，默认为50
15 %
16 % 输出参数：
17 %     cnt - 创建的模块数量
18 %
19 % 示例：
20 %     cnt = createStoreMem(gcs, 'Template.xlsx')
21 %     cnt = createStoreMem(gcs, 'Template.xlsx', 'step', 30, 'posIn', [100,
22 %     100])
23 %
24 % 作者：葛维冬 (Blue Ge)
25 % 日期：2024-04-29
26 % 版本：1.1
```

Template 模板

A	B	C	D	E
ModelName	PortTyp	Name	DataTyp	CustomStorageClas
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_n_u16BexvLmtCompSpdOffset	uint16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_p_u16BexvKeepHPres	uint16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_p_u16SM4CexvHiDischarge	uint16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_p_u16SM4CexvHiPres	uint16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_p_u16SM6CexvHiDischarge	uint16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_p_u16SM6CexvHiPres	uint16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_t_s32BexvPosnCalibReqTime	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_t_s32CexvPosnCalibReqTime	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_t_s32EexvPosnCalibReqTime	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_t_s32SM0DecPreTime	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s16SCCurDiffSM4CEXVPI	int16	CAL_16BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32BexvKeepTDischarge	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32EvapProtect	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32EvapProtectRecover	single	CAL_32BIT
TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32SM1SHHigh	single	CAL_32BIT



根据子模型列表，批量创建框架子模型

标准化建模，期望每个模块，尽量都绑定功能需求及软件需求，我们可以把系统所有需求使用excel表格来管理

FunctionID	RequirementID	ModelName
666666	26245	DemoSub1
666666	26246	DemoSub2
666666	26247	DemoSub3
666666	26248	DemoSub4
666666	26249	DemoSub5
888888	26250	DemoSub6
888888	26251	DemoSub7
888888	26252	DemoSub8
888888	26253	DemoSub9
888888	26254	DemoSub10

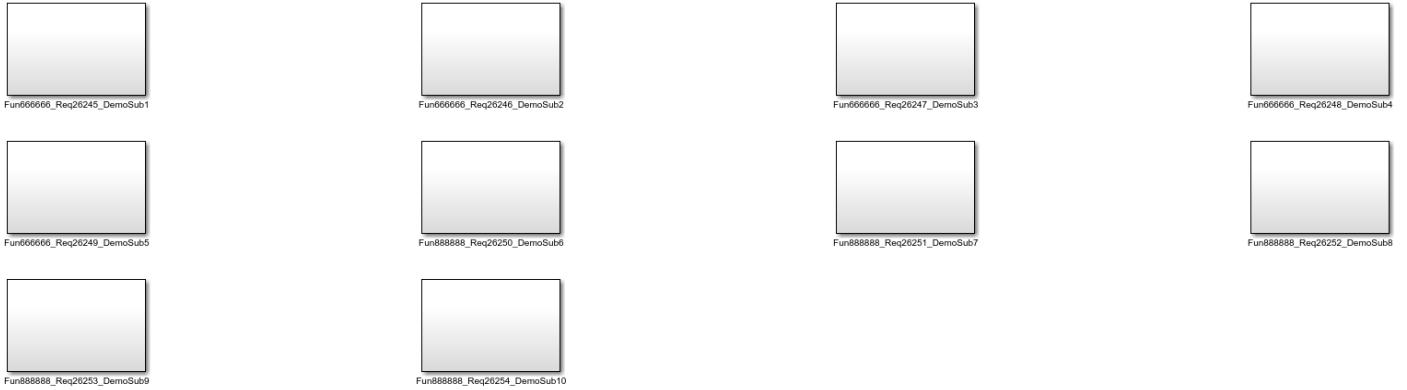
使用说明

```

1 function [modNums] = createSubMod(varargin)
2 %CREATESUBMOD 批量创建Simulink子模型
3 % MODNUMS = CREATESUBMOD() 使用默认参数创建子模型
4 % MODNUMS = CREATESUBMOD('Parameter', Value, ...) 使用指定参数创建子模型
5 %
6 % 输入参数 (名值对) :
7 %     'template'      - Excel文件名 (字符串), 默认值: 'Template.xlsx'
8 %     'sheet'         - Excel工作表名 (字符串), 默认值: 'modelName'
9 %     'FunPrefix'    - 功能ID前缀 (字符串), 默认值: 'Fun'
10 %    'ReqPrefix'    - 需求ID前缀 (字符串), 默认值: 'Req'
11 %    'rows'          - 布局行数 (正整数), 默认值: 1
12 %    'blockWidth'   - 模块宽度 (正整数), 默认值: 150
13 %    'blockHeight'  - 模块高度 (正整数), 默认值: 100
14 %    'hSpacing'     - 水平间距 (正整数), 默认值: 300
15 %    'vSpacing'     - 垂直间距 (正整数), 默认值: 50
16 %
17 % 输出参数:
18 %     modNums        - 创建的子模型数量 (正整数)
19 %
20 % 示例:
21 %     modNums = createSubMod('rows', 3)
22 %     modNums = createSubMod('template', 'myTemplate.xlsx', 'rows', 2)
23 %
24 % 注意事项:
25 %     1. 使用前需要打开目标Simulink模型
26 %     2. Excel模板需包含FunctionID、RequirementID和ModelName三列
27 %     3. 子模型命名格式为: FunXXXXXX_ReqYYYYYY_ModelName
28 %
29 % 参见: ADD_BLOCK, SET_PARAM, GCS
30 %
31 % 作者: Blue.ge
32 % 版本: 1.0
33 % 日期: 20250312

```

执行效果



创建测试管理器

完成各个子模型的单元测试后，我们需要使用一个测试管理器，管理所有的测试用例，导出报告等相关操作。此脚本可以从模型中导出所有的测试用例到测试管理器中进行管理，测试并导出报告。

使用方法

代码块

```
1 function createTestManage(modeFold, varargin)
2 %CREATETESTMANAGE 为模块创建测试管理器
3 %  createTestManage(modeFold) 为指定模块创建测试管理器，包括测试文件、
4 %  测试套件和测试用例的创建，以及测试运行和报告导出功能。
5 %
6 %  输入参数：
7 %      modeFold - 模块文件夹名称，例如'04_TmRefriVlvCtrl'
8 %
9 %  可选参数：
10 %      'run' - 是否运行测试，默认为false
11 %      'clear' - 是否清除测试结果，，默认为false
12 %      'export' - 是否导出测试结果，，默认为false
13 %
14 %  示例：
15 %      createTestManage('04_TmRefriVlvCtrl')
16 %      createTestManage('06_TmComprCtrl', 'run', true, 'export', true)
17 %      createTestManage('06_TmComprCtrl', 'clear', true)
18 %
19 %  作者：葛维冬 (Blue Ge)
20 %  日期：2024-04-18
21 %  版本：1.1
```

执行效果

Test Manager

TESTS

The screenshot shows the Test Manager application window. The top menu bar includes FILE, EDIT, RUN, and RESULTS sections with various icons for operations like New, Open, Save, Cut, Copy, Delete, Test Spec Report, Run, Run with Stepper, Stop, Parallel, Report, Visualize, Import, Export, and Model Data. Below the menu is a toolbar with similar icons. The main area has tabs for 'Test Browser' (selected) and 'Results and Artifacts'. A search bar at the top says 'Filter tests by name or tags, e.g. tags: test'. The 'Test Browser' pane displays a hierarchical tree of test cases under 'TmComprCtrl'. The tree structure is as follows:

- TmComprCtrl
 - TmComprCtrl/TmComprCtrl/AcEnable/0124_Offcode2
 - Offcode2
 - TmComprCtrl/TmComprCtrl/AcEnable/0125_Offcode3
 - Offcode3
 - TmComprCtrl/TmComprCtrl/ID29280_CompOn1
 - CompOn
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM1SM21/ID26245_SM1FF
 - SM1FF
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM1SM21/ID26247_SM1Error
 - SM1Error
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM2SM22/ID26267_SM2FF
 - SM2FF
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM2SM22/ID26265_SM2Ki
 - SM2Ki
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM2SM22/ID26265_SM2Kp
 - SM2Kp
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM2SM22/ID26239_SM2Error
 - SM2Error
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM4/ID26297_SM4FF
 - SM4FF
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM4/ID26259_SM4Error
 - SM4Error
 - TmComprCtrl/TmComprCtrl/ComprSpeed/Norm/SM6/ID26282_SM6FF
 - SM6FF

Below the tree view is a 'PROPERTY' table with the following data:

PROPERTY	VALUE
Name	TmComprCtrl
Location	D:\Thermal_HY11\03_OutWorking\02_PCMU_Integration\Ther...
Enabled	<input checked="" type="checkbox"/>
Tags	Type comma or space separated tags. You can filter tests using t

创建Interface 创建接口调试模块

根据excel模块，自动创建接口模块

- 充分考虑了所有信号链路的数据类型，确保一致
- 解决了输入输出信号不匹配问题，比如有些信号接口信号有，不过我们内部没有输出
- 自动为每个信号创建debug模块

- 对每个输入信号进行解析

使用说明

代码块

```
1 function numCreated = creatInterface(varargin)
2 %CREATINTERFACE 创建接口调试模块
3 % numCreated = creatInterface() 为输入输出接口创建调试模块，支持多种模式。
4 %
5 % 可选参数：
6 %     'excelFileName' - Excel模板文件名，默认为'Template.xlsx'
7 %     'sheetName' - 工作表名称，默认为'IF_InportsCommon'
8 %     'mode' - 模式选择，可选'inport'或'outport'，默认为'inport'
9 %     'sigUse' - 信号使用方式，可选'in'或'out'，默认为'in'
10 %    'posX' - 横坐标位置，默认为0
11 %    'posY' - 纵坐标位置，默认为0
12 %    'gndBlock' - 接地模块类型，可选'Ground'或'Constant'，默认为'Constant'
13 %
14 % 输出参数：
15 %     numCreated - 成功创建的信号数量
16 %
17 % 模式说明：
18 %     mode=0: 正常输入输出，创建Inport Debug Outport
19 %     mode=1: 有输入无输出，创建Inport和Terminator
20 %     mode=2: 有输出无输入，创建Ground和Outport
21 %     mode=3: 一个输入对应多个输出，分别创建并自动连线
22 %
23 % 示例：
24 %     numCreated = creatInterface()
25 %     numCreated = creatInterface('mode', 'outport', 'sheetName',
26 %     'IF_OutportsCommon')
27 %
28 % 作者：葛维冬 (Blue Ge)
29 % 日期：2024-06-28
30 % 版本：1.1
```

执行效果

详见

- 根据接口信号创建Interface输入模块
- 根据接口信号创建Interface输出模块

上述模块都是调用creatInterface来实现的

根据接口信号创建Interface输入模块

作为一个SWC，有好多输入输出信号，我们往往会将这些信号，转换成热系统的内部信号，同时为其增加观测量。此脚本就是批量生成相关模块

- 自动为每个信号创建debug模块
- 对每个输入信号进行解析
- 为所有的输入输出端口添加数据类型，其中interface 信号的数据类型是根据arxml IDT的别名文件判断的，输入信号是根据信号名判断数据类型的，如果两者数据类型不一致，会自动添加转换模块

Arxml 导出模型的时候生成的IDT文件如下

代码块

```
1 % -----
2 % Generated by MATLAB on 16-Jul-2024 09:24:31
3 % MATLAB version: 9.13.0.2320565 (R2022b) Update 6
4 %
5
6 IDT_AD4CoolReqForBkpUD_Ref = Simulink.AliasType;
7 IDT_AD4CoolReqForBkpUD_Ref.Description = '';
8 IDT_AD4CoolReqForBkpUD_Ref.DataScope = 'Auto';
9 IDT_AD4CoolReqForBkpUD_Ref.HeaderFile = 'Rte_Type.h';
10 IDT_AD4CoolReqForBkpUD_Ref.BaseType = 'boolean';
11 IDT_AD4CoolReqForBkp_Ref = Simulink.AliasType;
12 IDT_AD4CoolReqForBkp_Ref.Description = '';
13 IDT_AD4CoolReqForBkp_Ref.DataScope = 'Auto';
14 IDT_AD4CoolReqForBkp_Ref.HeaderFile = 'Rte_Type.h';
15 IDT_AD4CoolReqForBkp_Ref.BaseType = 'boolean';
```

使用说明

代码块

```
1 function numCreated = creatIFIn(varargin)
2 %CREATIFIN 创建接口输入模型
3 % numCreated = creatIFIn() 创建接口输入模型，包括信号创建、解析和端口配置。
4 %
5 % 可选参数：
6 % 'pos' - 模型位置，默认为[0,0]
7 % 'excelFileName' - Excel模板文件名，默认为 'Template.xlsx'
```

```

8    %      'sheetNames' - 工作表名称列表，默认为{'Imports Common', 'Imports
9    %
10   %  输出参数：
11   %      numCreated - 创建的信号总数
12   %
13   %  示例：
14   %      numCreated = creatIFIn()
15   %      numCreated = creatIFIn('pos', [1000,0], 'excelFileName',
16   %          'MyTemplate.xlsx')
17   %      numCreated = creatIFIn('sheetNames',
18   %          {'IF_ImportsCommon', 'IF_ImportsDiag'})
19   %
20   %  作者：葛维冬 (Blue Ge)
21   %  日期：2024-06-28
22   %  版本：1.1

```

执行效果

SigIn	SigOut	Description
sVcCcm_Te_AmbTEstimd	sTmln_Te_s32AmbTEstimd	
yVcCcm_B_AmbTEstimdQf	yTmln_B_AmbTEstimdQf	
sVcCcm_Te_IncarTemp	sTmln_Te_s32IncarTemp	
sVcCcm_Qf_IncarTempQf	sTmln_D_u8IncarTempQf	
sVcCcm_Te_Evap	sTmln_Te_s32Evap	
sVcCcm_Qf_TeEvap	yTmln_B_TeEvap	
yVcCcm_B_AirCondEnbl	yTmln_B_AirCondEnbl	
sVcCcm_Te_EvapReq	sTmln_Te_s32EvapReq	
yVcCcm_B_HvacHeatgReq	yTmln_B_HvacHeatgReq	
sVcCcm_Z_HvacAirMFlowEstimd	sTmln_md_s32HvacAirMFlowEstimd	
sVcCcm_Z_HexHeatrAirMFlowEstimd	sTmln_md_s32HexHeatrAirMFlowEstimd	
sVcCcm_D_HvacFanSts	sTmln_D_u8HvacFanSts	
sVcCcm_X_ClmtRcirc	sTmln_X_s32ClmtRcirc	
sVcCcm_X_HvacTFlpActlPosnFrtRowLe	sTmln_X_s32HvacTFlpActlPosnFrtRowLe	
sVcCcm_X_HvacTFlpActlPosnFrtRowRi	sTmln_X_s32HvacTFlpActlPosnFrtRowRi	
sVcCcm_Te_HeatrAirTReq	sTmln_Te_s32HeatrAirTReq	
sVcCcm_Te_HvacHexAirT	sTmln_Te_s32HvacHexAirT	
yVcCcm_B_HvacHexAirTQf	yTmln_B_HvacHexAirTQf	
yVcVdm_B_EcoClimaSts	yTmln_B_EcoClimaSts	

> ... Parameters | Rename | MemoryStorage | SigEdit | IF_ImportsCommon | IF_ImportsDiag | IF_Imports2F | IF_OutportsCommon | IF_OutportsDiag



根据接口信号创建Interface输出模块

作为一个SWC，有好多输入输出信号，我们往往会将这些信号，转换成热系统的内部信号，同时为其增加观测量。此脚本就是批量生成相关模块

- 自动为每个信号创建debug模块
- 对每个输入信号进行解析
- 为所有的输入输出端口添加数据类型，其中interface信号的数据类型是根据arxml IDT的别名文件判断的，输入信号是根据信号名判断数据类型的，如果两者数据类型不一致，会自动添加转换模块
- 对没有输出的信号进行接0处理，并对常量配置了对应的数据类型

Arxml 导出模型的时候生成的IDT文件如下

代码块

```

1  %
2  % Generated by MATLAB on 16-Jul-2024 09:24:31
3  % MATLAB version: 9.13.0.2320565 (R2022b) Update 6
4  %
5
6  IDT_AD4CoolReqForBkpUD_Ref = Simulink.AliasType;
7  IDT_AD4CoolReqForBkpUD_Ref.Description = '';
8  IDT_AD4CoolReqForBkpUD_Ref.DataScope = 'Auto';
9  IDT_AD4CoolReqForBkpUD_Ref.HeaderFile = 'Rte_Type.h';
10 IDT_AD4CoolReqForBkpUD_Ref.BaseType = 'boolean';
11 IDT_AD4CoolReqForBkp_Ref = Simulink.AliasType;
12 IDT_AD4CoolReqForBkp_Ref.Description = '';

```

```
13 IDT_AD4CoolReqForBkp_Ref.DataScope = 'Auto';
14 IDT_AD4CoolReqForBkp_Ref.HeaderFile = 'Rte_Type.h';
15 IDT_AD4CoolReqForBkp_Ref.BaseType = 'boolean';
```

使用说明

代码块

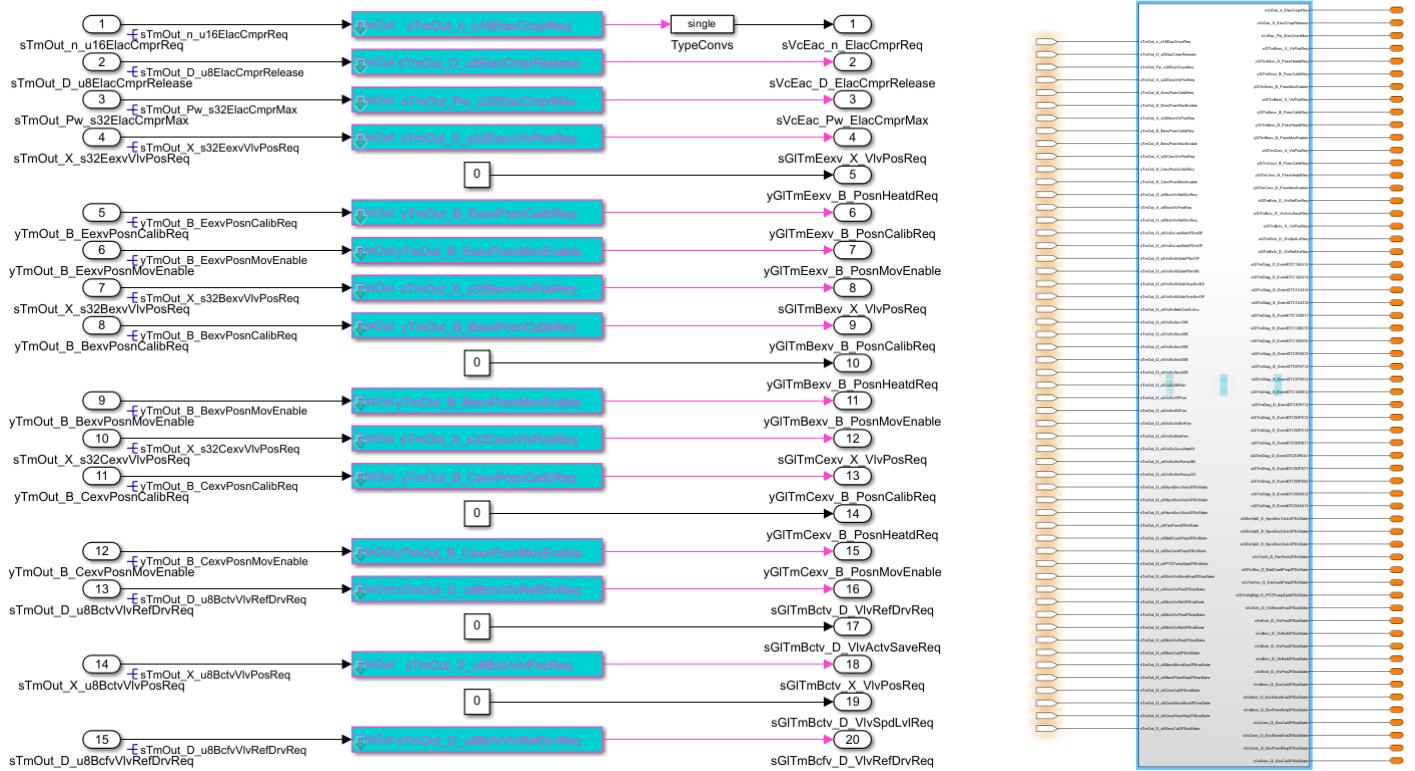
```
1 function numCreated = creatIFOut(varargin)
2 %CREATIFOUT 创建接口输出模型
3 %     numCreated = creatIFOut() 创建接口输出模型，包括信号创建、解析和端口配置。
4 %
5 %     可选参数：
6 %         'pos' - 模型位置，默认为[12000,0]
7 %         'excelFileName' - Excel模板文件名，默认为'Template.xlsx'
8 %         'sheetNames' - 工作表名称列表，默认为
9 %             {'IF_OutportsCommon','IF_OutportsDiag','IF_Outports2F'}
10 %
11 %     输出参数：
12 %
13 %     示例：
14 %         numCreated = creatIFOut()
15 %         numCreated = creatIFOut('pos', [13000,0], 'excelFileName',
16 %             'MyTemplate.xlsx')
17 %         numCreated = creatIFOut('sheetNames',
18 %             {'IF_OutportsCommon','IF_OutportsDiag'})
19 %
20 %     作者：葛维冬 (Blue Ge)
21 %     日期：2024-06-28
22 %     版本：1.1
```

执行效果

表格模板

Model	SigIn	SigOut
sTmComprCtrl_n_u16CompRpmReq	sTmOut_n_u16ElacCmprReq	sVcEac_n_ElacCmprReq
yTmComprCtrl_B_CompOnReq	sTmOut_D_u8ElacCmprRelease	sVcEac_D_ElacCmprRelease
sTmEnergyMgr_Pw_s32AcCmprPwrLim	sTmOut_Pw_s32ElacCmprMax	sVcEac_Pw_ElacCmprMax
sTmRefriVlvCtrl_X_s32EXVPoSReq	sTmOut_X_s32EexvVlvPosReq	sGITmEexv_X_VlvPosReq
NA	NA	yGITmEexy_B_PosnHeatdReq
yTmRefriVlvCtrl_B_EexvCalibReq	yTmOut_B_EexvPosnCalibReq	yGITmEexy_B_PosnCalibReq
yTmRefriVlvCtrl_B_HpcExxvMovEnable	yTmOut_B_ExxvPosnMovEnable	yGITmEexy_B_PosnMovEnable
sTmRefriVlvCtrl_X_s32BEXVPoSReq	sTmOut_X_s32BexvVlvPosReq	sGITmBexv_X_VlvPosReq
yTmRefriVlvCtrl_B_BexvCalibReq	yTmOut_B_BexvPosnCalibReq	yGITmBexv_B_PosnCalibReq
NA	NA	yGITmBexv_B_PosnHeatdReq
yTmRefriVlvCtrl_B_HpcBexvMovEnable	yTmOut_B_BexvPosnMovEnable	yGITmBexv_B_PosnMovEnable
sTmColtVlvCtrl_X_u8HpcBctvRefDrvReq	sTmOut_D_u8BctvVlvRefDrvReq	sGITmBctv_D_VlvRefDrvReq
NA	NA	sGITmBctv_D_VlvActvSaveReq
sTmColtVlvCtrl_X_u8BctvPosReq	sTmOut_X_u8BctvVlvPosReq	sGITmBctv_X_VlvPosReq
NA	NA	sGITmBctv_D_VlvSpdLvlReq
sTmColtVlvCtrl_D_u8HpcBcfvRefDrvReq	sTmOut_D_u8BcfvVlvRefDrvReq	sGITmBcfv_D_VlvRefDrvReq

< > ... Parameters | Rename | MemoryStorage | SigEdit | IF_ImportsCommon | IF_ImportsDiag | IF_Imports2F | IF_OutportsCommon | IF_OutportsDiag | IF_Outports2F



创建TmOut 模块

我们内部有好多子模型，期望都经过TmOut 转换后，再最终输出给Interface. 不过同一根信号线不能解析多个信号，因此需要使用信号转换模块进行转接。这个脚本就是基于excel 表格的信号匹配关系，为每个信号添加信号转接模块

使用说明

代码块

```
1 function nums = creatTmOut(varargin)
2 %CREATTMOUT 创建内部信号到基础模块的输出信号转换
3 %   nums = creatTmOut() 将内部信号转换为基础模块的输出信号，包括信号名转换和信号隔离。
4 %
5 % 可选参数：
6 %     'excelFileName' - Excel模板文件名，默认为'Template.xlsx'
7 %     'sheetNames' - 工作表名称列表，默认为
8 %         {'IF_OutportsCommon', 'IF_OutportsDiag', 'IF_Outports2F'}
9 %     'posBase' - 基础位置坐标，默认为[0,0]
10 %    'posMod' - 模块位置坐标，默认为[10500,0,11000,5000]
11 %    'NAStr' - 无信号标识，默认为'NA'
12 %    'step' - 信号间隔步长，默认为30
13 %    'gndBlock' - 接地模块类型，可选'Ground'或'Constant'，默认为'Constant'
14 %
15 % 输出参数：
16 %     nums - 成功创建的信号数量
17 %
18 % 示例：
19 %     nums = creatTmOut()
20 %     nums = creatTmOut('posBase', [1000,0], 'step', 40)
21 %     nums = creatTmOut('sheetNames',
22 %         {'IF_OutportsCommon', 'IF_OutportsDiag'})
23 %
24 % 作者：葛维冬 (Blue Ge)
25 % 日期：2024-06-28
26 % 版本：1.1
```

执行效果



创建框架引用模型

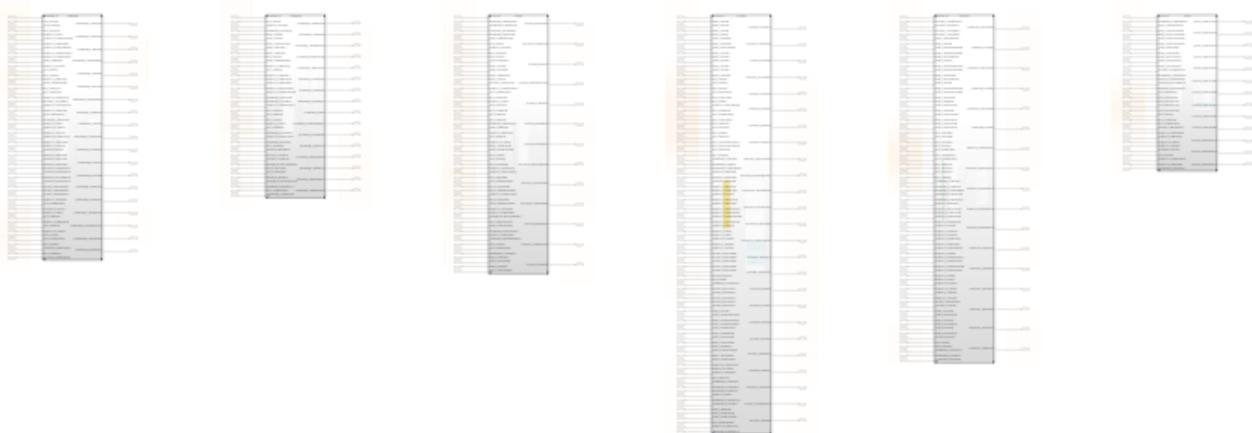
在各个子模型完成后，需要创建框架引用模型，引用所有相关模型，这个功能可以使用此脚本来实现

使用说明

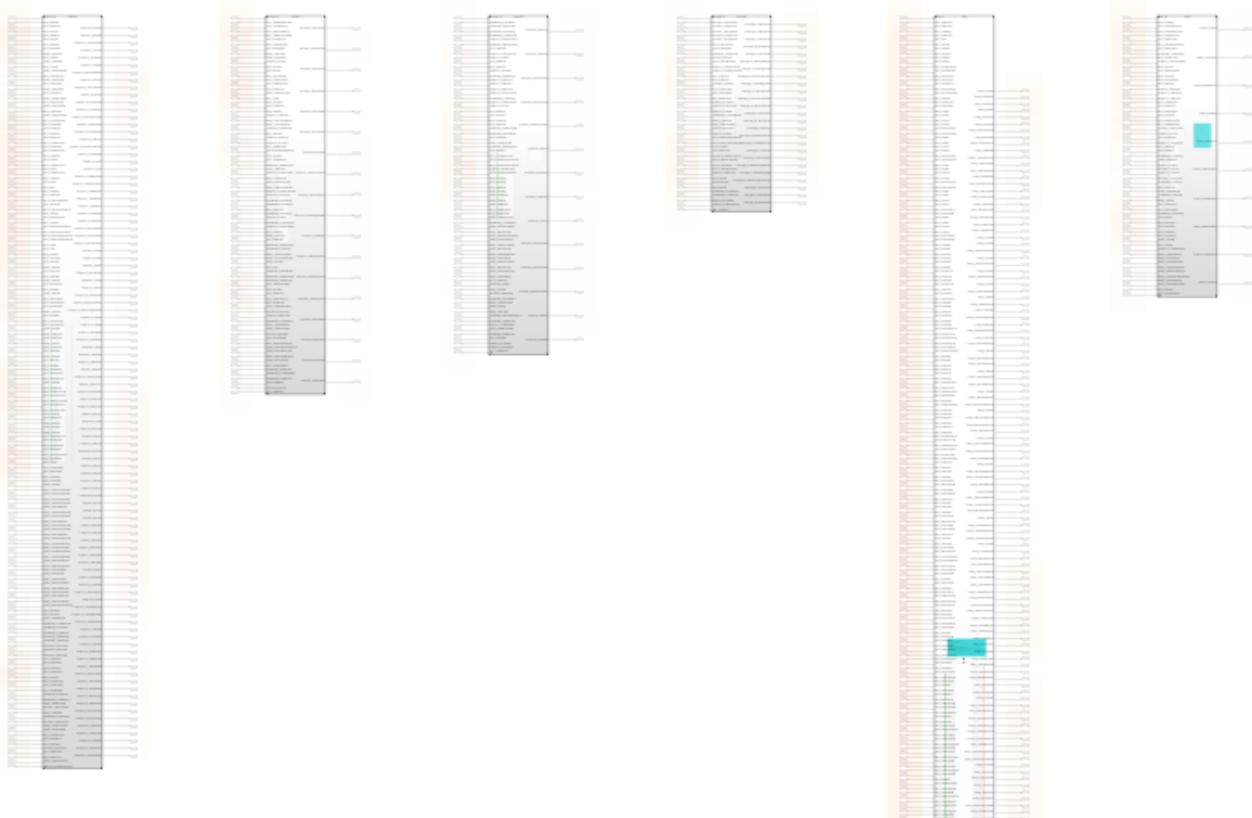
代码块

```
1 function creatTmBase(varargin)
2 %CREATTMBASE 创建基础模块的输入信号转换
3 % creatTmBase() 将外部信号转换为基础模块的输入信号，包括信号名转换和信号隔离。
4 %
5 % 可选参数：
6 %     'stpX' - X轴步长，默认为1500
7 %     'stpY' - Y轴步长，默认为4000
8 %     'posOrg' - 原始位置坐标，默认为[0 0 500 5000]
9 %
10 % 示例：
11 %     creatTmBase()
12 %     creatTmBase('stpX', 2000, 'stpY', 5000)
13 %     creatTmBase('posOrg', [100 100 600 5100])
14 %
15 % 作者：葛维冬 (Blue Ge)
16 % 日期：2024-06-28
17 % 版本：1.1
```

执行效果



根据各个模型自动为其创建引用模型



删

删除当前层以下所有无连接的信号线

脚本说明

代码块

```
1 delUselessLine(gcs)
```

执行效果



删除无用的输入输出端口

模型搭建完成后，有些模型会有多余的信号接地的，或者注释掉的，如果这些信号很多，逐个删除会比较麻烦，此脚本就是用来批量删除无用的端口及其相连接的模块

脚本说明

delUselessPort(gcs)

代码块

```
1 function [validImports, validExports] = delUselessPort(path, varargin)
2 %DELUSELESSPORT 删除模型中无用的输入输出端口及其相连模块
3 % [validImports, validExports] = delUselessPort(path) 删除当前路径下无用的端口
4 % [validImports, validExports] = delUselessPort(path, 'UselessInBlock',
5 % 'Terminator', 'UselessOutBlock', 'Ground')
6 %
7 % 输入参数:
8 %     path - 模型路径, 可以是gcs或bdroot
9 %     'UselessInBlock' - 要删除的输入端口连接的模块类型, 默认为'Terminator'
10 %     'UselessOutBlock' - 要删除的输出端口连接的模块类型, 默认为'Ground'
11 %
12 % 输出参数:
13 %     validImports - 保留的有效输入端口列表
14 %     validExports - 保留的有效输出端口列表
15 %
16 % 示例:
17 %     [validImports, validExports] = delUselessPort(gcs)
18 %     [validImports, validExports] = delUselessPort(gcs, 'UselessInBlock',
19 % 'Terminator')
20 %
21 % 作者: Blue.ge
22 % 日期: 20231020
23 % 修改: 20240321
```

执行效果

代码块

```
1 delUselessPort(gcs)
```



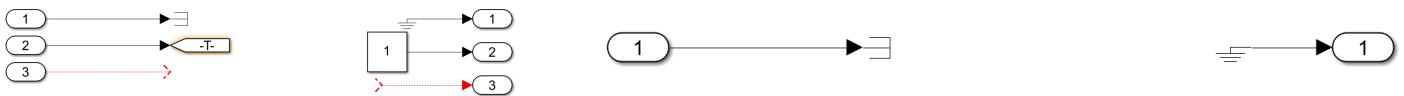
可以指定删除的类型

% 'UselessInBlock' - 要删除的输入端口连接的模块类型, 默认为'Terminator'

% 'UselessOutBlock' - 要删除的输出端口连接的模块类型, 默认为'Ground'

代码块

```
1 delUselessPort(gcs, 'UselessInBlock', 'Goto', 'UselessOutBlock', 'Constant')
```



改

改变输入输出端口位置

脚本说明

```
changePortBlockPosAll(gcs)
```

代码块

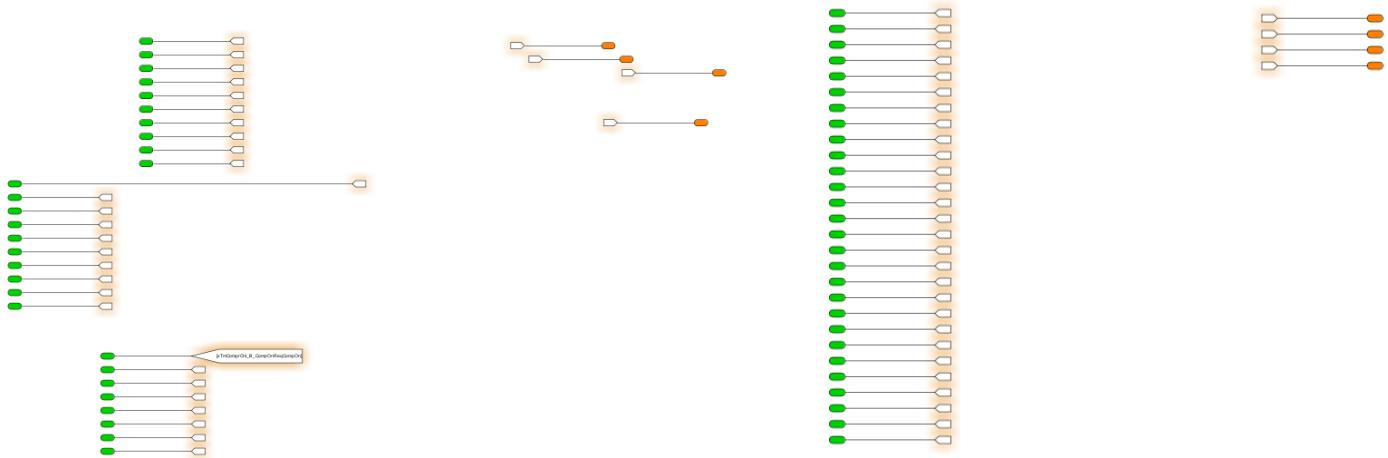
```
1 function changePortBlockPosAll(path,varargin)
2 %CHANGEPORBLOCKPOSALL 重新整理当前模型所有的输入输出端口及其对应连线block的位置
3 % CHANGEPORBLOCKPOSALL(PATH) 使用默认参数重新整理指定路径下模型的端口位置
4 % CHANGEPORBLOCKPOSALL(PATH, 'Parameter', Value, ...) 使用指定参数重新整理端口
5 % 位置
6 % 输入参数:
7 % path          - 目标系统路径 (字符串), 默认值: gcs (当前系统)
8 %
9 % 可选参数 (名值对) :
10 % 'newPosin'   - 新输入端口位置 [x y], 默认值: [0 0]
11 % 'newPosout'  - 新输出端口位置 [x y], 默认值: [1000 0]
```

```

12 % 'portSize' - 端口模块尺寸 [width height], 默认值: [30 14]
13 % 'blockSize' - 连接模块尺寸 [width height], 默认值: [150 14]
14 %
15 % 功能描述:
16 %     1. 查找系统中的所有输入输出端口
17 %     2. 根据指定的新位置重新排列端口
18 %     3. 保持端口之间的间距为固定值
19 %     4. 自动调整端口对应的连线block位置
20 %     5. 可选择性地修改端口和连接模块的尺寸
21 %
22 % 示例:
23 %     % 基本用法
24 %     changePortBlockPosAll(gcs)
25 %
26 %     % 默认端口尺寸和模块尺寸
27 %     changePortBlockPosAll(gcs, 'portSize', [30 14], 'blockSize', [30 14])
28 %
29 %     % 指定位置和尺寸
30 %     changePortBlockPosAll('myModel/subsystem1', ...
31 %         'newPosin', [-730 3600], ...
32 %         'newPosout', [0 3600], ...
33 %         'portSize', [30 14], ...
34 %         'blockSize', [30 14])
35 %
36 % 注意事项:
37 %     1. 使用前需要打开目标Simulink模型
38 %     2. 端口间距固定为30像素
39 %     3. 如果不指定新位置, 将使用第一个端口的位置作为基准
40 %     4. 端口和模块的尺寸可以统一设置
41 %
42 % 参见: CHANGEPORTBLOCKPOS, FINDMODPORTS
43 %
44 % 作者: Blue.ge
45 % 版本: 1.1
46 % 日期: 20231020

```

执行效果



根据DCM改变SLDD初始值

背景

有时候看软件的时候，默认值并非标定后的数据，如果加载slidd之前，基于最新的DCM，改变slidd的初值，就可以解决这个问题

痛点：别人熟悉模型的时候，有时候默认标定量比较简单，很难看出最终标定量的变化趋势
举例：比如上面的Y周，制冷等级1，3，分别对应的Ki值，哪个更大些呢？10是否合理呢？

标定后最终的DCM数据

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	-6	-4	-2	-1	-0.5	0	0.5	1	1.5	2	4	6	8	10
1	1	0	10	10	10	0	10	10	10	10	10	10	10	0
2	2	0	10	10	10	0	10	10	10	10	10	10	10	0
3	3	0	10	10	10	0	10	10	10	10	10	10	10	0

脚本说明

```
outPath = changeArchSlld('HY11_PCMU_Tm_OTA3_V6050327_All.DCM', 'PCMU_SLDD_All.xlsx')
```

```

代码块function outPath = changeArchSlld(dcmPath, slldPath, varargin)
2 %CHANGEARCHSLDD 根据DCM数据更新SLDD文件中的参数值
3 % OUTPATH = CHANGEARCHSLDD(DCMPATH, SLDDPATH) 使用默认参数更新SLDD文件
4 % OUTPATH = CHANGEARCHSLDD(DCMPATH, SLDDPATH, 'Parameter', Value, ...) 使用指定参数更新
5 %
6 % 输入参数:
7 % dcmPath - DCM文件路径 (字符串)
8 % slldPath - SLDD文件路径 (字符串)
9 %
10 % 可选参数 (名值对) :
11 % 'Sheet' - Excel表格名称, 默认值: 'Parameters'
12 %
13 % 输出参数:
14 % outPath - 更新后的SLDD文件路径 (字符串)
15 %
16 % 功能描述:
17 % 1. 读取SLDD文件中的参数表格
18 % 2. 从DCM文件中提取参数值
19 % 3. 更新SLDD文件中的参数初始值
20 % 4. 保存更新后的SLDD文件
21 % 5. 创建1D和2D参数表格
22 %
23 % 示例:
24 % % 基本用法
25 % outPath = changeArchSlld('HY11_PCMU_Tm_OTA3_V6050327_All.DCM',
26 % 'PCMU_SLDD_All.xlsx')
27 %
28 % % 指定表格名称
29 % outPath = changeArchSlld('HY11_PCMU_Tm_OTA3_V6050327_All.DCM', ...
30 % 'PCMU_SLDD_All.xlsx', 'Sheet', 'MyParameters')
31 %
32 % 注意事项:
33 % 1. DCM文件必须包含有效的参数定义
34 % 2. SLDD文件必须包含正确的表格结构
35 % 3. 参数类型支持: const, axis, wert, tab, map
36 %
37 % 参见: FINDDCMPARAM, CHANGESLDDINITVALUE, SAVESLDDTABLEBYINITVALUE
38 %
39 % 作者: Blue.ge
40 % 版本: 1.0
41 % 日期: 20240124

```

执行效果

RTE_Interfa	Dimensio	Detal	Value	Unit	IniValue	Min	Max
	[1, 1]					0	255
	[1, 1]					0	65535
	[1, 1]			s			
	[1, 2]						
	[1, 2]						
	[1, 1]						
	[1, 1]						
	[1, 2]						
	[1, 2]						
	[1, 1]						
	[1, 1]						
	[1, 27]						
	[1, 27]						
	[1, 1]						
	[1, 1]						
	[1, 27]						
	[1, 27]						
	[1, 1]						
	[1, 1]						
	[1, 27]						
	[1, 27]						
	[1, 1]						
	[1, 1]						
	[1, 27]						

Dimensio	Detail	Value	Unit	In/Value	Min	Max
[1,1]				5011225		
[1,1]				0.00000001		
[1,1]				1	0	255
[1,1]		5		1	0	65535
[1,2]				[0.115000002086163 4.75089979171753]		
[1,2]				[0 3400]		
[1,1]				0		
[1,1]				0		
[1,2]				[0.25 4.49749755859375]		
[1,2]				[0 1025]		
[1,1]				0		
[1,1]				0		
[1,27]				[0.0621673129498959 0.0698600858449936 0.0886498689651485]		
[1,27]				[140 135 125 115 105 95 85 80 75 70 65 60 57 54 52 50 48 46 43 40]		
[1,1]				0		
[1,1]				0		
[1,27]				[0.0740372240543365 0.1084936121216545 0.141993638992310.		
[1,27]				[130 115 105 95 85 75 65 55 45 40 35 30 25 20 15 10 8 6 4 2 0 -2 -5		
[1,1]				0		
[1,1]				0		
[1,27]				[0.0758806765079498 0.110646076500416 0.1443110704422 0.15		
[1,27]				[130 115 105 95 85 75 65 55 45 40 35 30 25 20 15 10 8 6 4 2 0 -2 -5		
[1,1]				0		
[1,1]				0		
[1,27]				[0.0621673129498959 0.0785463973879814 0.100296922028065]		
[1,27]				[140 130 120 115 110 105 100 95 80 85 82 80 78 75 70 65 60 55 50]		

将sldd 所有初始值清空

根据DCM，自动生成sIdd的初始值

运行脚本后，可以找到汇总后的slld 重复的标定量

代码块

```
1 there is dump idx: 3059 3067 ----> cTmDiag_t_s32RefcSov2Err2Dhu
2 there is dump idx: 3736 4880 ----> cTmIn_B_CmprPwrEna2FReq_db
3 there is dump idx: 3737 4881 ----> cTmIn_B_CmprPwrEna2FReq_sw
4 there is dump idx: 3740 4884 ----> cTmIn_B_CmprStsEna2FReq_db
5 there is dump idx: 3741 4885 ----> cTmIn_B_CmprStsEna2FReq_sw
6 there is dump idx: 3918 4888 ----> cTmIn_B_PtcFlow2FReq_db
7 there is dump idx: 3919 4889 ----> cTmIn_B_PtcFlow2FReq_sw
8 there is dump idx: 4120 4886 ----> cTmIn_D_u8CmprStsEnaReq2F_db
9 there is dump idx: 4121 4887 ----> cTmIn_D_u8CmprStsEnaReq2F_sw
10 there is dump idx: 4650 4890 ----> cTmIn_L_s32PtcFlowReq2F_db
11 there is dump idx: 4651 4891 ----> cTmIn_L_s32PtcFlowReq2F_sw
12 there is dump idx: 4856 4878 ----> cTmIn_n_s32CmprSpdReq2F_db
13 there is dump idx: 4857 4879 ----> cTmIn_n_s32CmprSpdReq2F_sw
14 there is dump idx: 4654 4882 ----> cTmIn_Pw_s32CmprPwrEnaReq2F_db
15 there is dump idx: 4655 4883 ----> cTmIn_Pw_s32CmprPwrEnaReq2F_sw
16
17 outPath =
18
19
'D:\Thermal_HY11\03_OutWorking\02_PCMU_Integration\Thermal_PCMU_23N7\SubModel\PCMU_SLDD_All.xlsx'
```

可以找到实际模型中，无用的标定量

可以过滤出无用的信号

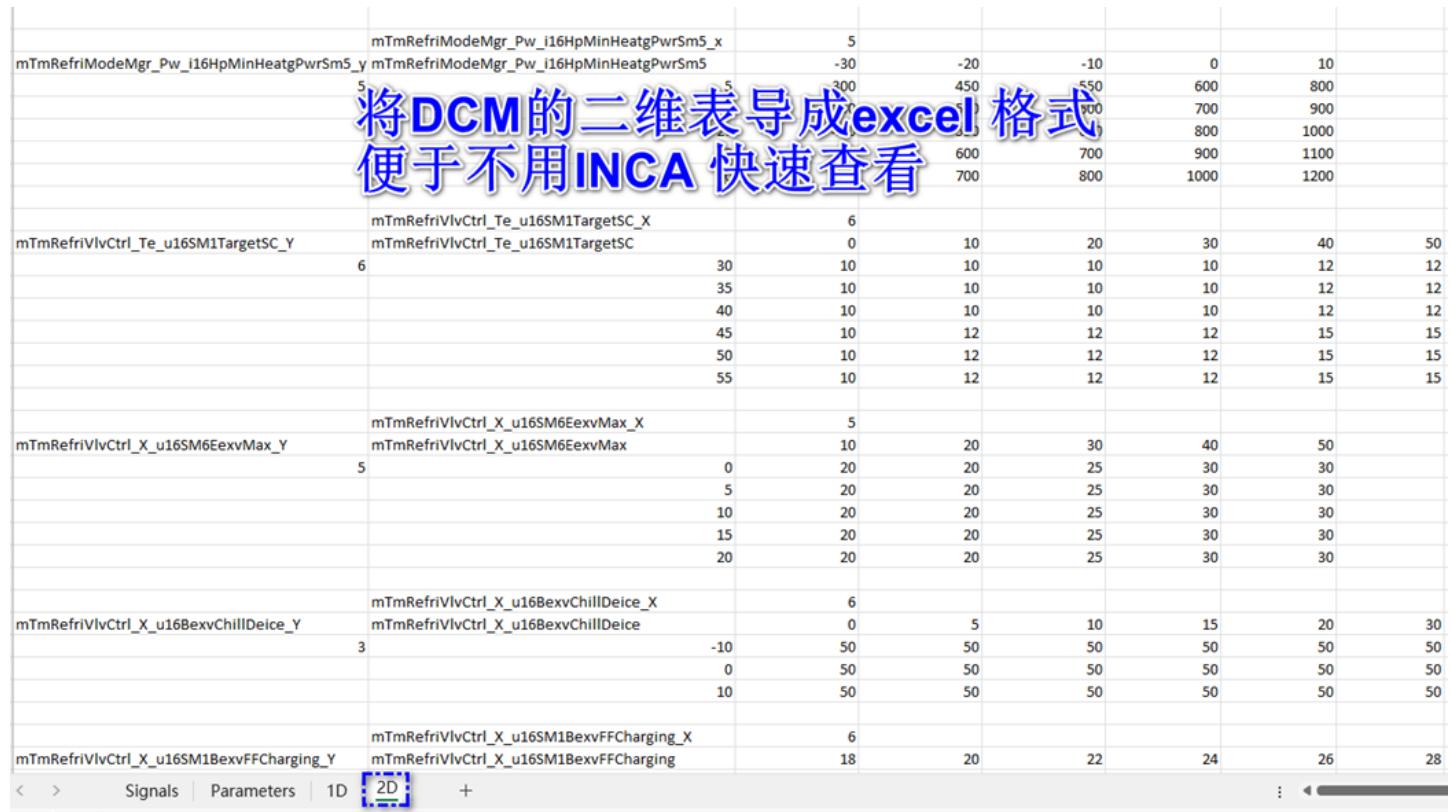
ModelName	PortType	Name	DataType	CustomStorageClass	DefinitionFile	RTE_Interface	Dimension	Detail	ValueTable	Unit	InValue
TmSigProces	Local	mTmSigProces_Te_s32TOutMinMultDehu	single	CAL_32BIT	TmSigProces.c		[3,5]				
TmSigProces	Local	mTmSigProces_Te_s32TOutMinMultDehu_Y	single	CAL_32BIT	TmSigProces.c		[1,3]				
TmSigProces	Local	mTmSigProces_Te_s32TOutMinMultDehu_X	uint16	CAL_16BIT	TmSigProces.c		[1,5]				
TmSigProces	Local	mTmSigProces_Te_s32TOutMinMultDehuECO	single	CAL_32BIT	TmSigProces.c		[3,5]				
TmSigProces	Local	mTmSigProces_Te_s32TOutMinMultDehuECO_Y	single	CAL_32BIT	TmSigProces.c		[1,3]				
TmSigProces	Local	mTmSigProces_Te_s32TOutMinMultDehuECO_X	uint16	CAL_16BIT	TmSigProces.c		[1,5]				
TmSigProces	Local	cTmSigProces_Te_s32FMDentDehuTintMin	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32FMDentDehuTintMid	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32FMDentDehuTintLow	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32FMDentDehuTintHigh	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_T_s32EnterDehuConditionJudgeTime	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_T_s32QuitDehuConditionJudgeTime	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_B_HpEnFlagTamtSw	boolean	CAL_8BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_B_HpEnFlagTamt_db	boolean	CAL_8BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32TDiffMaxPTCTrgBat	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32TBathExOffMinPTCTrgBat	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32TBattRefForDeiceWSM	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32FVVentT	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32FPHHeatT	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32TDehuBattTOffset	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32ExtDehuBattTOffset	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32EntDehuBattTLim	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmSigProces	Local	cTmSigProces_Te_s32OutDehuBattTLim	single	CAL_32BIT	TmSigProces.c		[1,1]				
TmRefrlModeMgr	Local	cTmRefrlModeMgr_Te_s32SupChrgDiffBarMaxBatReqCmpOn	single	CAL_32BIT	TmRefrlModeMgr.c		[1,1]			degC	
TmRefrlModeMgr	Local	cTmRefrlModeMgr_Te_s32SupChrgDiffBarReqBarCurCmpOn	single	CAL_32BIT	TmRefrlModeMgr.c		[1,1]			degC	
TmRefrlModeMgr	Local	cTmRefrlModeMgr_Te_s32SupChrgDiffBarMaxBatReqCmpOff	single	CAL_32BIT	TmRefrlModeMgr.c		[1,1]			degC	
TmRefrlModeMgr	Local	cTmRefrlModeMgr_t_u16DehuToPTCKeepTime	uint16	CAL_16BIT	TmRefrlModeMgr.c		[1,1]			s	
TmRefrlModeMgr	Local	cTmRefrlModeMgr_X_u16EvxToErr	uint16	CAL_16BIT	TmRefrlModeMgr.c		[1,1]				
TmRefrlModeMgr	Local	cTmRefrlModeMgr_t_u16EXVToMaxPosStartCoolPeriod	uint16	CAL_16BIT	TmRefrlModeMgr.c		[1,1]				
TmRefrlModeMgr	Local	cTmRefrlModeMgr_Te_s32HpReqTempSaturoffsetPaDeHu	single	CAL_32BIT	TmRefrlModeMgr.c		[1,1]			degC	

同时，会自动生成1维表和2维表的sheet，便于不用打开INCA或者matlab, 即可快速查看

将DCM的一维表导成excel格式
便于不用打开INCA快速查看

tTmComprCtrl_p_u16CoolAcHiSideHiCut_X	800	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16CoolAcHiSideHiCut	3000	3000	3000	3000	3000	3000
tTmComprCtrl_p_u16HeatActHiSideHiCut_X	200	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16HeatActHiSideHiCut	3000	3000	3000	3000	3000	3000
tTmComprCtrl_p_u16HeatCmpLimitHighP_X	800	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16HeatCmpLimitHighP	2000	2000	2000	2000	2000	2000
tTmComprCtrl_p_u16HeatCmpLimitNormalP_X	800	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16HeatCmpLimitNormalP	1700	1700	1700	1700	1700	1700
tTmComprCtrl_p_u16SM6AcHiSideHiCut_X	800	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16SM6AcHiSideHiCut	3000	3000	3000	3000	3000	3000
tTmComprCtrl_p_u16SM6CmpLimitHighP_X	800	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16SM6CmpLimitHighP	2000	2000	2000	2000	2000	2000
tTmComprCtrl_p_u16SM6CmpLimitNormalP_X	800	1200	2000	3000	6000	8000
tTmComprCtrl_p_u16SM6CmpLimitNormalP	1700	1700	1700	1700	1700	1700
tTmComprCtrl_n_u16HpAwdCmpSpdLAvoidFre	2	1	2			

< > Signals Parameters 1D 2D +



改变配置-autosar 配置

背景

将模型配置成威睿提供的autosar 配置脚本后，`CodeGenAutosarCfg`， 可进一步修改其默认配置

脚本说明

changeCfgAutosar('TmComprCtrl')

代码块

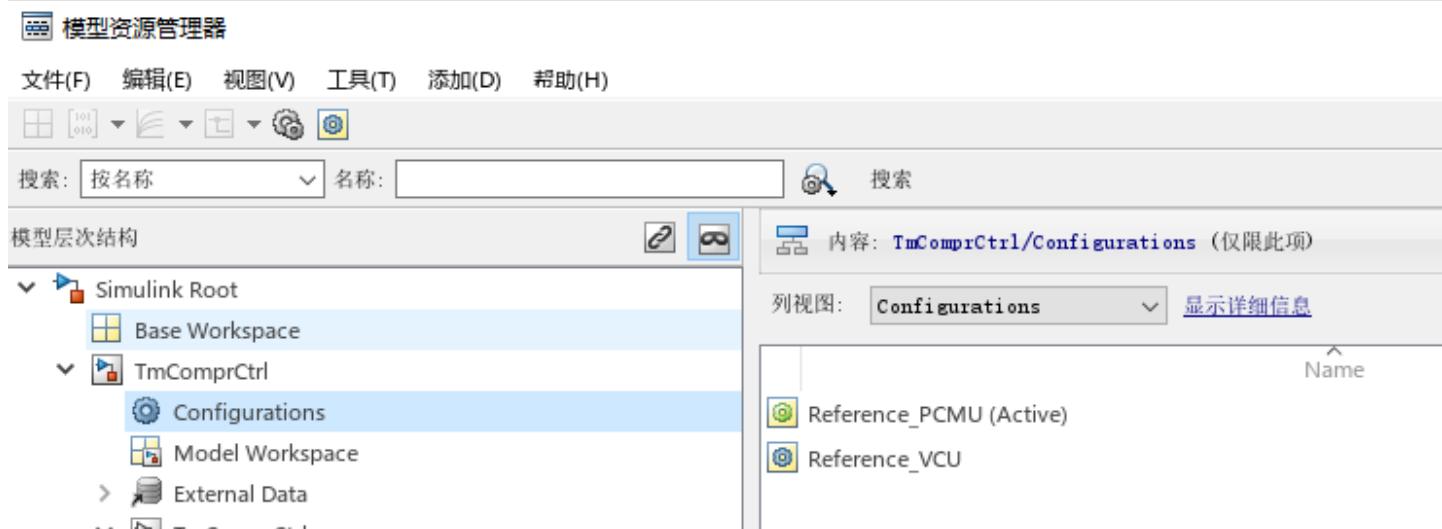
```
1 function changeCfgAutosar(modelName)
2 %CHANGECFGAUTOSAR 配置AUTOSAR模型的参数设置
3 %     CHANGECFGAUTOSAR(MODELNAME) 配置指定AUTOSAR模型的参数设置
4 %
5 %     输入参数:
6 %         modelName      - 模型名称 (字符串)
7 %
8 %     功能描述:
9 %         1. 打开指定的AUTOSAR模型
10 %        2. 配置代码生成参数
11 %        3. 创建AUTOSAR API
12 %        4. 设置模型配置参数
13 %        5. 保存模型更改
14 %
```

```

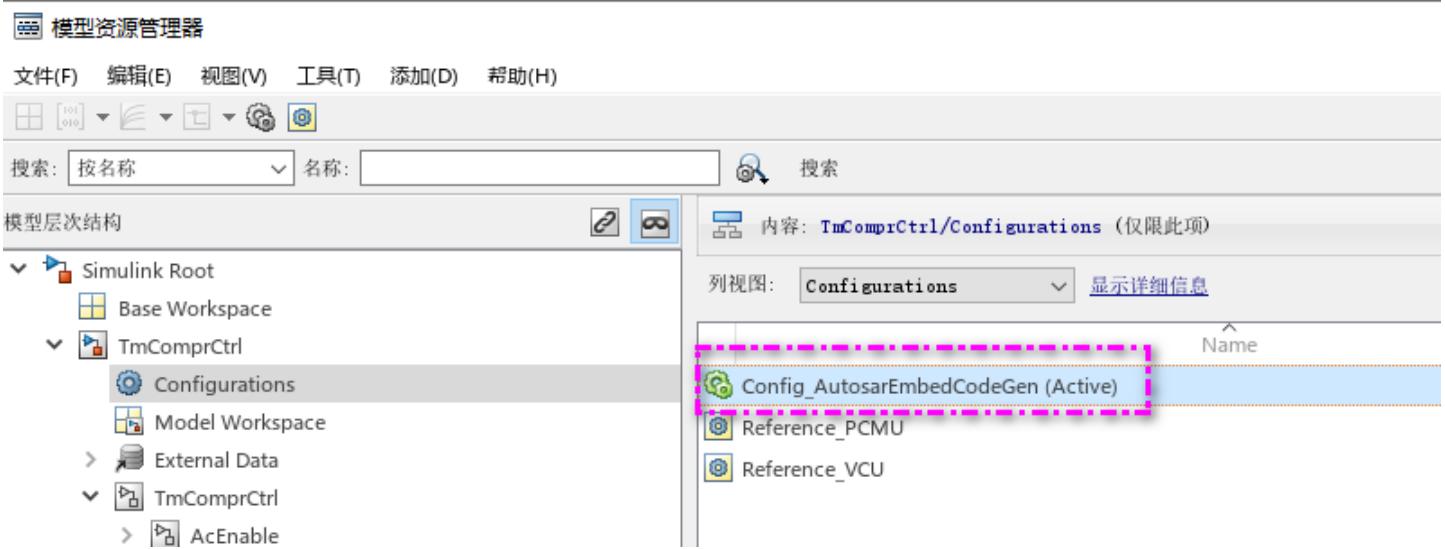
15 % 示例:
16 %     % 基本用法
17 %     changeCfgAutosar('TmSwArch')
18 %
19 % 注意事项:
20 %     1. 使用前需要确保模型文件存在
21 %     2. 配置参数包括:
22 %         - SampleTimeConstraint: 设置为 'Unconstrained'
23 %         - SaveWithDisabledLinksMsg: 设置为 'none'
24 %         - SaveWithParameterizedLinksMsg: 设置为 'none'
25 %         - UnconnectedInputMsg: 设置为 'none'
26 %         - UnconnectedOutputMsg: 设置为 'none'
27 %     3. 函数会自动保存模型更改
28 %
29 % 参见: CODEGENAUTOSARCFG, AUTOSAR.API.CREATE
30 %
31 % 作者: Blue.ge
32 % 版本: 1.0
33 % 日期: 20231201

```

执行效果



运行完成后，模型会新建一个autosar配置，并按个人需求，改变配置中的某些参数



注意事项

运行代码前，请确保模型仿真通过

改变配置-ERT配置

背景

如果要验证VCU配置生成代码，也需要将模型按VCU的配置改变下，包括默认存储类，内存段的配置等。

脚本说明

```
changeCfgErt('TmComprCtrl')
```

代码块

```
1 function changeCfgErt(modelName)
2 %CHANGECFGERT 配置ERT模型的参数设置和代码生成选项
3 % CHANGECONFIGERT(MODELNAME) 配置指定ERT模型的参数设置和代码生成选项
4 %
5 % 输入参数:
6 %     modelName      - 模型名称 (字符串)
7 %
8 % 功能描述:
9 %     1. 打开指定的ERT模型
10 %    2. 加载VCU配置
11 %    3. 配置模型参数设置
12 %    4. 设置代码映射配置
13 %    5. 配置内存段设置
14 %    6. 保存模型更改
15 %
16 % 示例:
17 %     % 基本用法
```

```

18 % changeCfgErt('TmComprCtrl')
19 %
20 % 注意事项:
21 % 1. 使用前需要确保模型文件存在
22 % 2. 需要确保'Configuration_VCU.mat'文件存在
23 % 3. 根据模型类型自动配置不同的内存段设置:
24 %   - TmSwArch: 配置ExportedFunction内存段
25 %   - TEM_VCU1/TmIn/TmOut: 配置Periodic:D1内存段
26 %   - 其他模型: 配置ExportedFunction:modelName_100内存段
27 % 4. 函数会自动保存模型更改
28 %
29 % 参见: CODER.MAPPING, CODER.DICTIONARY
30 %
31 % 作者: Blue.ge
32 % 版本: 1.0
33 % 日期: 20240122

```

执行效果

执行后，配置文件会使用一个引用文件，同时会改变

The screenshot shows two main windows from the MATLAB/Simulink environment:

- Model Catalog Browser (Top Window):** This window displays the 'Configurations' section of the 'TmComprCtrl' model. It lists three configurations: 'Config_AutosarEmbedCodeGen', 'Reference_PCMU', and 'Reference_VCU (Active)'. A configuration reference named 'Configuration_VCU' is highlighted with a pink box.
- Embedded Coder (Bottom Window):** This window shows the 'C 代码' tab of the 'TmComprCtrl' configuration. On the left, the 'Data Types' pane is open, showing memory segments like 'DATAV216_MEMORY' and various data types. On the right, the 'Exported Global' configuration is detailed, including settings for '存储类型' (Non-volatile), '文件位置' (Output), and '头文件' (Header file). Below these panes, the '伪代码预览' (Pseudo-code Preview) section shows generated C code for functions like 'TmComprCtrl_100' and 'TmComprCtrl_initialize'.

相关配置好后，就可以按VCU的要求，通过模型引用生成代码了

改变配置-PCMU模型引用

背景

将所有子模型，统一改成PCMU所需要的引用配置

脚本说明

```
changeCfgRef('TmComprCtrl', 'TmVcThermal_Configuration_sub')
```

代码块

```
1 function changeCfgRef(mdName, cfg)
2 %CHANGECFGREF 更改模型的配置引用设置
3 % CHANGECFGREF(MDNAME, CFG) 为指定模型设置配置引用
4 %
5 % 输入参数:
6 %     mdName          - 模型名称 (字符串)
7 %     cfg              - 配置文件名称 (字符串)
8 %
9 % 功能描述:
10 %      1. 加载指定的配置文件
11 %      2. 为模型创建或更新配置引用
12 %      3. 激活新的配置引用
13 %      4. 保存并关闭模型
14 %
15 % 示例:
16 %     % 基本用法
17 %     changeCfgRef('TmComprCtrl', 'TmVcThermal_Configuration_sub')
18 %
19 % 注意事项:
20 %      1. 使用前需要确保配置文件存在
21 %      2. 配置文件应为.mat格式
22 %      3. 函数会自动保存并关闭模型
23 %      4. 如果配置引用已存在, 将更新其源文件
24 %
25 % 参见: SIMULINK.CONFIGSETREF, SETACTIVECONFIGSET
26 %
27 % 作者: Blue.ge
28 % 版本: 1.0
29 % 日期: 20231031
```

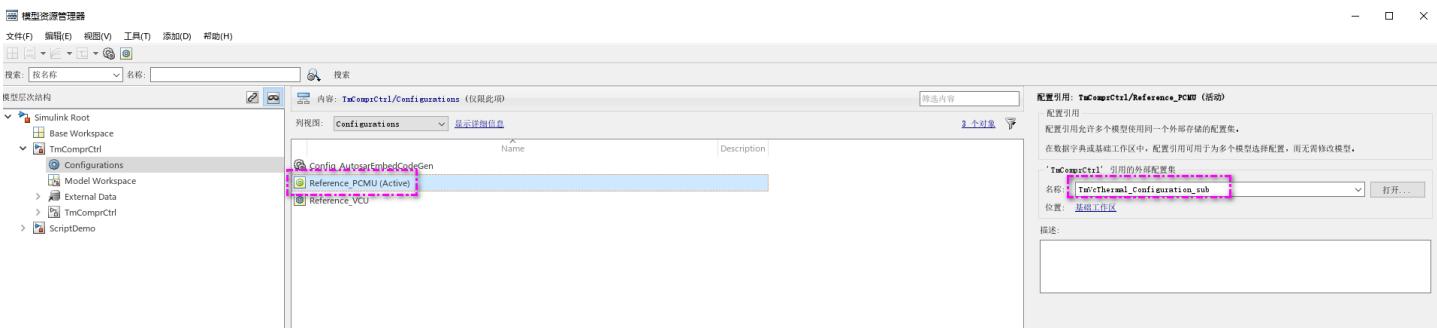
以下是兄弟脚本，可以遍历改变所有模型

```
changeCfgRefAll()
```

代码块

```
1 function changeCfgRefAll()
2 %CHANGECFGREFALL 批量更改所有子模型的配置引用
3 %    CHANGECFGREFALL() 为所有子模型设置相同的配置引用
4 %
5 %    功能描述:
6 %        1. 加载配置文件
7 %        2. 遍历所有子模型
8 %        3. 为每个子模型设置配置引用
9 %        4. 显示处理进度
10 %
11 %    示例:
12 %        % 基本用法
13 %        changeCfgRefAll()
14 %
15 %    注意事项:
16 %        1. 使用前需要确保配置文件存在
17 %        2. 配置文件应为.mat格式
18 %        3. 子模型列表包括:
19 %            - 远飞负责的模块: TmRefriModeMgr, TmColtModeMgr, TmHvchCtrl
20 %            - Blue负责的模块: TmComprCtrl, TmRefriVlvCtrl, TmSovCtrl
21 %            - 江敏负责的模块: TmSigProces, TmPumpCtrl, TmColtVlvCtrl, TmEnergyMgr
22 %            - 冬清负责的模块: TmDiag, TmAfCtrl
23 %        4. 函数会自动保存所有模型的更改
24 %
25 %    参见: CHANGECFGREF
26 %
27 %    作者: Blue.ge
28 %    版本: 1.0
29 %    日期: 20231101
```

执行效果



批量替换输入输出模型端口名

背景

一般的子模型输出端口名称的前缀，都是所负责的模块，不过有时候如果多个case 都输出这些信号，最终通过merge 实现的时候，那输出信号就不能相同，比如SM1输出pid 参数信号，SM2也输出pid 参数信号，因此，每个case 的输出，需要按case 前缀来命名，因此可能会需要批量改下输出名称



Before

脚本说明

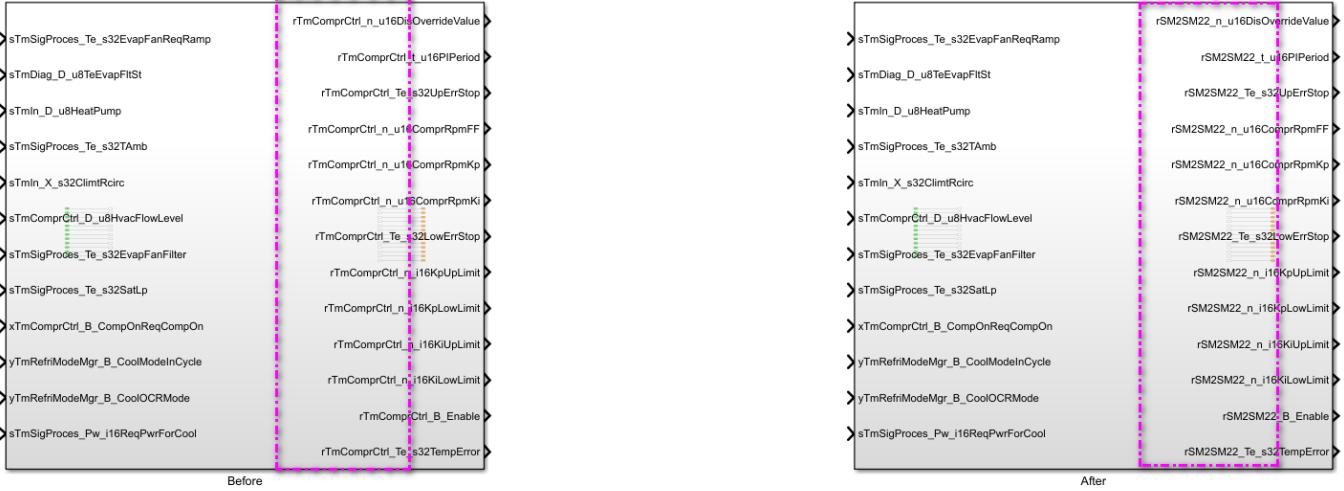
portsChanged = changeModPortName(gcb, 'TmComprCtrl', 'SM2SM22')

代码块

```
1 function portsChanged = changeModPortName(path, old, new, varargin)
2 %CHANGEMODPORTNAME 批量改变模型的端口名称
3 % PORTSCHANGED = CHANGEMODPORTNAME(PATH, OLD, NEW) 使用默认参数改变所有端口名称
4 % PORTSCHANGED = CHANGEMODPORTNAME(PATH, OLD, NEW, 'Parameter', Value, ...)
5 % 使用指定参数
6 %
7 % 输入参数:
8 %     path          - 模型路径 (字符串)
9 %     old           - 需要替换的旧名称 (字符串)
```

```
10 %
11 % 可选参数 (名值对) :
12 %     'changeIn' - 是否改变输入端口名称, 默认值: true
13 %     'changeOut' - 是否改变输出端口名称, 默认值: true
14 %
15 % 输出参数:
16 %     portsChanged - 已更改的端口名称列表 (元胞数组)
17 %
18 % 功能描述:
19 %     1. 查找模型中的所有输入输出端口
20 %     2. 根据指定的旧名称和新名称替换端口名称
21 %     3. 可选择性地只处理输入端口或输出端口
22 %     4. 返回所有更改的端口名称列表
23 %
24 % 示例:
25 %     % 基本用法 (改变所有端口)
26 %     portsChanged = changeModPortName(gcb, 'TmComprCtrl', 'SM123')
27 %
28 %     % 只改变输入端口
29 %     portsChanged = changeModPortName(gcb, 'TmComprCtrl', 'SM123', ...
30 %                                         'changeIn', true, 'changeOut', false)
31 %
32 % 注意事项:
33 %     1. 使用前需要确保模型已打开
34 %     2. 端口名称更改会立即生效
35 %     3. 函数会返回所有更改的端口名称
36 %
37 % 参见: FINDMODPORTS
38 %
39 % 作者: Blue.ge
40 % 版本: 1.1
41 % 日期: 20231207
```

执行效果



根据信号名称设置端口数据类型

有时候我们新建端口的时候，默认数据类型都是继承的，一旦整个链路都没有指定过数据类型，那生成的代码可能就会有问题，因此最好根据信号名称可以自动设置每个端口的数据类型

脚本说明

`portsChanged = changeModPortType(gcs)`

代码块

```

1  function portsChanged = changeModPortType(path)
2  %CHANGEMODPORTTYPE 根据信号名称自动更改端口数据类型
3  % PORTSCHANGED = CHANGEMODPORTTYPE(PATH) 根据信号名称自动更改指定路径下模型的端口
4  % 数据类型
5  %
6  % 输入参数:
7  %     path          - 模型路径 (字符串)
8  %
9  % 输出参数:
10 %
11 % 功能描述:
12 %     1. 查找模型中的所有输入输出端口
13 %     2. 根据端口名称自动判断数据类型
14 %     3. 如果当前数据类型与判断结果不一致，则更新数据类型
15 %     4. 返回所有更改的端口名称列表
16 %
17 % 示例:
18 %     % 基本用法
19 %         portsChanged = changeModPortType(gcs)
20 %
21 % 注意事项:
22 %     1. 使用前需要确保模型已打开

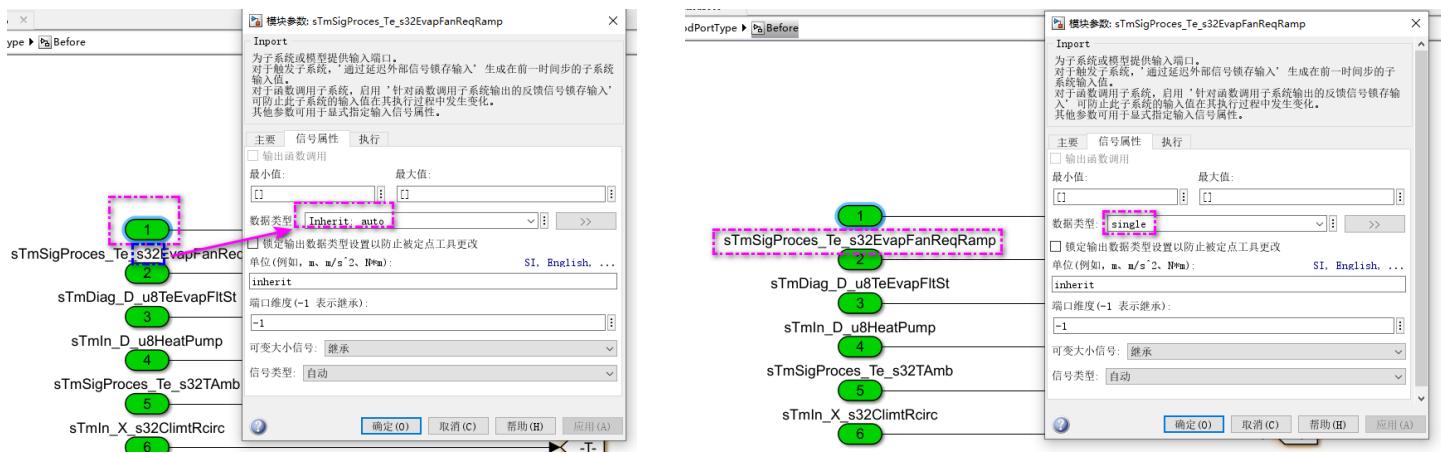
```

```

23 % 2. 数据类型更改会立即生效
24 % 3. 函数会返回所有更改的端口名称
25 % 4. 数据类型判断基于信号名称规则
26 %
27 % 参见: FINDMODPORTS, FINDNAMETYPE
28 %
29 % 作者: Blue.ge
30 % 版本: 1.0
31 % 日期: 20231020

```

执行效果



改变模型大小

创建好一个子模型后，由于端口数量不一样，子模型高度也不一样，因此，如果根据端口数量，自动调整其高度，那就比较方便了

脚本说明

`changeModSize(gcb)`

代码块

```

1 function changeModSize(pathMd, varargin)
2 %CHANGEMODSIZE 根据端口数量调整Simulink模型的大小
3 % CHANGEMODSIZE(PATHMD) 使用默认宽度调整指定模型的大小
4 % CHANGEMODSIZE(PATHMD, 'Parameter', Value, ...) 使用指定参数调整模型大小
5 %
6 % 输入参数:
7 %     pathMd      - 模型路径或句柄 (字符串或数值)
8 %
9 % 可选参数 (名值对) :
10 %     'wid'        - 模型最小宽度 (正整数), 默认值: 400
11 %     'minHeight'  - 模型最小高度 (正整数), 默认值: 60

```

```

12 % 'portStep' - 端口间距 (正整数), 默认值: 30
13 %
14 % 功能描述:
15 % 根据模型的输入输出端口数量, 自动调整模型的大小。
16 % 模型宽度至少为指定的'wid'值, 高度根据端口数量自动计算。
17 % 调整后会自动整理模型内部的连线。
18 %
19 % 示例:
20 % changeModSize(gcb)
21 % changeModSize('myModel/subsystem1')
22 % changeModSize(gcb, 'wid', 600, 'minHeight', 100)
23 %
24 % 注意事项:
25 % 1. 函数会自动跳过MATLAB内置的模块 (如Compare To Constant等)
26 % 2. 对于没有端口的模型, 会使用最小尺寸
27 %
28 % 参见: GET_PARAM, SET_PARAM, FINDMODPORTS, CHANGELINEARRANGE
29 %
30 % 作者: Blue.ge
31 % 版本: 1.1
32 % 日期: 20231020

```

执行效果



对当前模型的子模型进行布局

在一个子模型下, 有些子模型大小, 布局都会比较乱, 因此可以用这个脚本来调整下布局, 同时, 会根据端口数量, 自动调整其高度

脚本说明

`changeModSizeGcs('rows', 2)`

代码块

```

1 function changeModSizeGcs(varargin)
2 %CHANGEMODSIZEGCS 调整当前Simulink页面中子模型的大小和布局

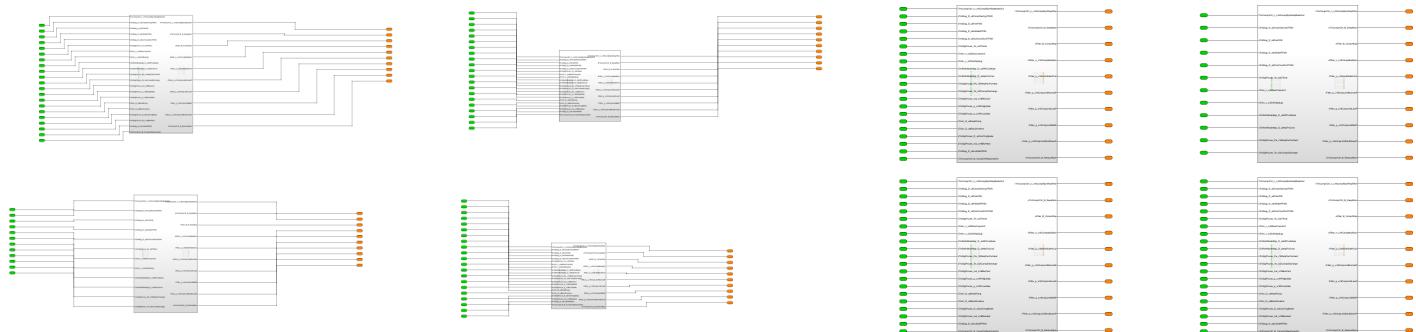
```

```

3 % CHANGEMODSIZEGCS() 使用默认参数调整模型大小和布局
4 % CHANGEMODSIZEGCS('Parameter', Value, ...) 使用指定参数调整模型
5 %
6 % 输入参数 (名值对) :
7 %     'wid'          - 子模型宽度 (正整数), 默认值: 400
8 %     'xStp'         - 水平间距 (正整数), 默认值: 400
9 %     'yStp'         - 垂直间距 (正整数), 默认值: 60
10 %    'rows'         - 布局行数 (正整数), 默认值: 0 (不进行行列布局)
11 %
12 % 示例:
13 % changeModSizeGcs()
14 % changeModSizeGcs('rows', 2)
15 % changeModSizeGcs('wid', 500, 'rows', 2, 'xStp', 300, 'yStp', 100)
16 %
17 % 注意事项:
18 % 1. 使用前需要打开目标Simulink模型
19 % 2. 当rows=0时, 只调整模型大小, 不改变布局
20 % 3. 布局完成后会自动整理连线
21 %
22 % 参见: GET_PARAM, SET_PARAM, GCS
23 %
24 % 作者: Blue.ge
25 % 版本: 1.0
26 % 日期: 20231020

```

执行效果



批量改变信号数据类型

如果一个信号，一开始是u16类型的，只有又改为s32类型，信号名称可以直接通过matlab 替换功能直接换掉，不过对应的输入输出端口的数据类型，也需要同步替换下。

脚本说明

```
% changeSigType('rTmComprCtrl_n_s32CompRpmReqModeOut')
% changeSigType('rTmComprCtrl_n_s32CompRpmReqModeOut', 'type', 'uint16')
```

代码块

```
1 function changeSigType(sigName, varargin)
2 %CHANGESIGTYPE 更改指定信号的端口数据类型
3 % CHANGESIGTYPE(SIGNAME, TYPE) 使用默认参数更改信号的数据类型
4 % CHANGESIGTYPE(SIGNAME, TYPE, 'Parameter', Value, ...) 使用指定参数更改
5 %
6 % 输入参数:
7 %     sigName      - 信号名称 (字符串)
8 %     type         - 目标数据类型 (字符串)
9 %
10 % 可选参数 (名值对) :
11 %     'allLev'     - 是否在全局范围内搜索, 默认值: true
12 %     'fromSigName' - 是否根据信号名称自动判断类型, 默认值: true
13 %
14 % 功能描述:
15 %     1. 根据参数设置搜索指定信号的所有端口
16 %     2. 根据信号名称或指定类型更新端口数据类型
17 %     3. 支持输入端口和输出端口的处理
18 %     4. 可选择在全局或当前系统范围内搜索
19 %
20 % 示例:
21 %     % 基本用法
22 %     changeSigType('rTmComprCtrl_n_s32CompRpmReqModeOut')
23 %     changeSigType('rTmComprCtrl_n_s32CompRpmReqModeOut', 'type', 'uint16')
24 %
25 %     % 根据信号名
26 %     changeSigType('rTmComprCtrl_n_s32CompRpmReqModeOut', 'fromSigName',
27 %                   true)
28 %
29 %     % 在当前系统范围内搜索并更新
30 %     changeSigType('rTmComprCtrl_n_s32CompRpmReqModeOut', 'single',
31 %                   'allLev', true)
32 %
33 % 注意事项:
34 %     1. 使用前需要确保模型已打开
35 %     2. 数据类型更改会立即生效
36 %     3. 如果fromSigName为true, 将忽略type参数
37 %     4. 支持的数据类型包括: single, uint8, int32等
38 %
39 % 参见: FINDNAMETYPE
40 % 作者: Blue.ge
41 % 版本: 1.1
```

执行效果

对应的信号名，改为正确的格式

根据sldd初始值改变sldd的一位表及二维表

如之前脚本介绍，对如果对表格类的标定量，单独使用一位表，二维表进行维护，那更加直观，此脚本就是该表了sldd 初始值后，会同步更新对应的表格中的值

脚本说明

`changeSlldTableByInitValue('TmRefriVlvCtrl_DD_PCMU.xlsx')`

代码块

```

1 function changeSlldTableByInitValue(slldPath)
2 %CHANGESLDDTABLEBYINITVALUE 根据SLDD初始值更新Excel表格
3 % CHANGESLDDTABLEBYINITVALUE(SLDDPATH) 根据SLDD数据字典的初始值更新Excel表格
4 %
5 % 输入参数:
6 % slldPath - SLDD Excel文件路径 (字符串)
7 %
8 % 功能描述:
9 % 1. 读取Excel文件中的Parameters表格
10 % 2. 根据初始值创建1维查找表
11 % 3. 根据初始值创建2维查找表
12 %
13 % 示例:
14 % % 基本用法
15 % changeSlldTableByInitValue('TmRefriVlvCtrl_DD_PCMU.xlsx')
16 %
17 % 注意事项:
18 % 1. Excel文件必须包含Parameters表格
19 % 2. 文件将被直接修改，请确保已备份
20 % 3. 支持1维和2维查找表的创建
21 %
22 % 作者: Blue.ge
23 % 版本: 1.1
24 % 日期: 20240802

```

执行效果

Function	Local	Description	Type	Initial Value
14. TmRefriLvCtrl	Local	cTmRefriLvCtrl_Te_s32ewProtect	single	15
15. TmRefriLvCtrl	Local	cTmRefriLvCtrl_Te_s32ewProtectRecover	single	3
16. TmRefriLvCtrl	Local	cTmRefriLvCtrl_Te_s32SM1SHigh	single	20
17. TmRefriLvCtrl	Local	cTmRefriLvCtrl_Te_s32SM1SHLow	single	10
18. TmRefriLvCtrl	Local	mTmRefriLvCtrl_Te_u16SM1TargetSC	uint16	[100 100 100 100 150] 10 10 10 10 15 15 15 15 20 20 10 15 15 20 20 20 10 15 15 20 20 20 10 15 15 20 20]
19. TmRefriLvCtrl	Local	mTmRefriLvCtrl_Te_u16SM1TargetSC_Y	single	[30 35 40 45 50 55]
20. TmRefriLvCtrl	Local	mTmRefriLvCtrl_Te_u16SM1TargetSC_X	single	[0 10 20 30 40 50]
21. TmRefriLvCtrl	Local	cTmRefriLvCtrl_Te_u16SM4CevsHSlm	uint16	15

27	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_u16SM6CexvSHLim	uint16
28	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_u16SM6CexvSHLimRecov	uint16
29	TmRefriVlvCtrl	Local	tTmRefriVlvCtrl_Te_u16SM4TargetSC	uint16
30	TmRefriVlvCtrl	Local	tTmRefriVlvCtrl_Te_u16SM4TargetSC_X	uint16

		mTmRefnVlvCtrl_Te_u16SM1TargetSC_Y	mTmRefnVlvCtrl_Te_u16SM1TargetSC_X	6
6			mTmRefnVlvCtrl_Te_u16SM1TargetSC	
			30	10 10 20 30 40 50
			35	10 10 10 10 15 15
			40	10 15 15 15 20 20
			45	10 15 15 20 20 20
			50	10 15 15 20 20 20
			55	10 15 15 20 20 20
)		mTmRefnVlvCtrl_X_u16SM6EevxMax_Y	mTmRefnVlvCtrl_X_u16SM6EevxMax_X	5
?)			mTmRefnVlvCtrl_X_u16SM6EevxMax	
5			10	20 30 40 50
			0	20 25 30 30
			5	20 25 30 30
			10	20 25 30 30
			15	20 25 30 30
			20	20 25 30 30
3		mTmRefnVlvCtrl_X_u16BewxChillDeice_Y	mTmRefnVlvCtrl_X_u16BewxChillDeice_X	6
?)			mTmRefnVlvCtrl_X_u16BewxChillDeice	
3			0	5 10 15 20 30
			-10	50 50 50 50 50
			0	50 50 50 50 50
			10	50 50 50 50 50
1		mTmRefnVlvCtrl_X_u16SM1BevxFFCCharging_Y	mTmRefnVlvCtrl_X_u16SM1BevxFFCCharging_X	6
?)			mTmRefnVlvCtrl_X_u16SM1BevxFFCCharging	
7			18	20 22 24 26 28
			15	30 30 30 30 30
			20	30 30 30 30 30
			25	30 30 30 30 30
			30	30 30 30 30 30
			35	30 30 30 30 30
			40	30 30 30 30 30
			45	30 30 30 30 30
1		mTmRefnVlvCtrl_X_u16SM1BevxFFSuperCharging_Y	mTmRefnVlvCtrl_X_u16SM1BevxFFSuperCharging_X	6
?)			mTmRefnVlvCtrl_X_u16SM1BevxFFSuperCharging	
7			18	20 22 24 26 28
			15	40 40 40 40 40
			20	40 40 40 40 40
			25	40 40 40 40 40
			30	40 40 40 40 40

执行脚本之后

	5	10	20	30	40	50
tTmRefnVlCtrl_Te_u16SM4TargetSC_X	30	4	5	6	7	
tTmRefnVlCtrl_Te_u16SM4TargetSC						
	5					
tTmRefnVlCtrl_Te_u16SM6TargetSC_X	10	20	30	40	50	
tTmRefnVlCtrl_Te_u16SM6TargetSC	3	4	5	6	7	
	4					
tTmRefnVlCtrl_X_u16SM2ExvMin_X	15	25	35	45		
tTmRefnVlCtrl_X_u16SM2ExvMin	8	12	18	20		
	9					
tTmRefnVlCtrl_X_u16SM2CycleBexvKi_x	-3	-2	-1	-1	0	0.5
tTmRefnVlCtrl_X_u16SM2CycleBexvKi	0	0	1	1	0	1
	4					
tTmRefnVlCtrl_Te_u16SM2TargetSC_x	1	2	3	4		
tTmRefnVlCtrl_Te_u16SM2TargetSC	4	7	10	15		
	4					
tTmRefnVlCtrl_Te_u16SM2ExvSHLim_x	1	2	3	4		
tTmRefnVlCtrl_Te_u16SM2ExvSHLim	8	13	15	20		
	4					
tTmRefnVlCtrl_Te_u16SM2ExvSHLimRecov_x	1	2	3	4		
tTmRefnVlCtrl_Te_u16SM2ExvSHLimRecov	3	8	15	20		
	5					
tTmRefnVlCtrl_X_u16SM3BexvMinRepPos_x	30	35	40	45	50	
tTmRefnVlCtrl_X_u16SM3BexvMinRepPos	8	10	20	30	40	
	4					
tTmRefnVlCtrl_Te_u16SM3TargetSC_x	1	2	3	4		
tTmRefnVlCtrl_Te_u16SM3TargetSC	4	7	10	15		
	19					
tTmRefnVlCtrl_X_u16SM3ExvKi_x	-20	-15	-10	-8	-6	-4
tTmRefnVlCtrl_X_u16SM3ExvKi	0	0	0	0	0	1
< >	Infor	Standards	Parameters	1D	2D	+

	mTmRefriVlvCtrl_Te.u16SM1TargetSC_Y	mTmRefriVlvCtrl_Te.u16SM1TargetSC_X	6
6		mTmRefriVlvCtrl_Te.u16SM1TargetSC	30 100 100 100 100 100 150
		30	0 10 20 30 40 50
		35	10 10 10 10 15 15
		40	10 15 15 15 20 20
		45	10 15 15 20 20 20
		50	10 15 15 20 20 20
		55	10 15 15 20 20 20
	mTmRefriVlvCtrl_X.u16SM6ExxMax_Y	mTmRefriVlvCtrl_X.u16SM6ExxMax_X	5
5		mTmRefriVlvCtrl_X.u16SM6ExxMax	10 20 30 40 50
		0	20 20 25 30 30
		5	20 20 25 30 30
		10	20 20 25 30 30
		15	20 20 25 30 30
		20	20 20 25 30 30
	mTmRefriVlvCtrl_LX.u16BexvChillDeice_Y	mTmRefriVlvCtrl_LX.u16BexvChillDeice_X	6
3		mTmRefriVlvCtrl_LX.u16BexvChillDeice	0 5 10 15 20 30
		-10	50 50 50 50 50 50
		0	50 50 50 50 50 50
		10	50 50 50 50 50 50
	mTmRefriVlvCtrl_X.u16SM1BexvFFCharging_Y	mTmRefriVlvCtrl_X.u16SM1BexvFFCharging_X	6
7		mTmRefriVlvCtrl_X.u16SM1BexvFFCharging	18 20 22 24 26 28
		15	30 30 30 30 30 30
		20	30 30 30 30 30 30
		25	30 30 30 30 30 30
		30	30 30 30 30 30 30
		35	30 30 30 30 30 30
		40	30 30 30 30 30 30
		45	30 30 30 30 30 30
	mTmRefriVlvCtrl_LX.u16SM1BexvFFSuperCharging_Y	mTmRefriVlvCtrl_LX.u16SM1BexvFFSuperCharging_X	6
7		mTmRefriVlvCtrl_LX.u16SM1BexvFFSuperCharging	18 20 22 24 26 28
		15	40 40 40 40 40 40
		20	40 40 40 40 40 40
		25	40 40 40 40 40 40
		30	40 40 40 40 40 40

根据sldd的一位表及二维表改变sldd初始值

如之前脚本介绍，对如果对表格类的标定量，单独使用一位表，二维表进行维护，那更加直观，不过存在一个通电，就是slld 的初始值在一个单元格中体现，改变初始值不太直观，如果直接在对应的2D 表中更改，再同步回去，那就更加方便了

初始值在一个单元格中体现

14	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32EvapProtect	single		15	
15	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32EvapProtectRecover	single		3	
16	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32SM1SHigh	single		10	初始值扩大了10倍
17	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_s32SM1SLow	single		10	
18	TmRefriVlvCtrl	Local	mTmRefriVlvCtrl_Te_u16SM1TargetSC_X	uint16	[0 100 100 100 100 10 10 10 15 15 10 15 15 20 20 10 15 15 20 20 10 15 15 20 20 10 15 15 20 20 10]	15	
19	TmRefriVlvCtrl	Local	mTmRefriVlvCtrl_Te_u16SM1TargetSC_Y	single	[0 35 40 45 50 55]	15	
20	TmRefriVlvCtrl	Local	mTmRefriVlvCtrl_Te_u16SM1TargetSC_X	single	[0 10 20 30 40 50]	15	
21	TmRefriVlvCtrl	Local	cTmRefriVlvCtrl_Te_u16SM4CexvSHLim	uint16		15	

初始值在对应的2D表中体现

A	B	C	D	E	F	G	H	I
mTmRefriVlvCtrl_Te_u16SM1TargetSC_Y	mTmRefriVlvCtrl_Te_u16SM1TargetSC_X	6						
6	mTmRefriVlvCtrl_Te_u16SM1TargetSC	0	10	20	30	40	50	
		30	100	100	100	100	100	150
		35	10	10	10	10	15	15
		40	10	15	15	15	20	20
		45	10	15	15	20	20	20
		50	10	15	15	20	20	20
		55	10	15	15	20	20	20

脚本说明

changeSlldInitValueByTable('TmRefriVlvCtrl_DD_PCMU.xlsx')

代码块

```

1   function changeSlldInitValueByTable(slldPath)
2   %CHANGESLDDINITVALUEBYTABLE 根据Excel表格更新SLDD数据字典的初始值
3   % CHANGESLDDINITVALUEBYTABLE(SLDDPATH) 根据指定Excel文件更新SLDD数据字典
4   %
5   % 输入参数:
6   %     slldPath - SLDD Excel文件路径 (字符串)
7   %
8   % 功能描述:
9   %     1. 读取Excel文件中的Parameters表格
10  %     2. 处理1维查找表数据
11  %     3. 处理2维查找表数据
12  %     4. 更新并保存修改后的SLDD文件
13  %
14  % 示例:
15  % 基本用法
16  % changeSlldInitValueByTable('TmRefriVlvCtrl_DD_PCMU.xlsx')
17  %
18  % 注意事项:
19  %     1. Excel文件必须包含Parameters表格
20  %     2. 1维查找表数据将更新为[表名]和[表名_X]
21  %     3. 2维查找表数据将更新为[表名]、[表名_X]和[表名_Y]
22  %     4. 文件将被直接覆盖, 请确保已备份
23  %
24  % 作者: Blue.ge
25  % 版本: 1.1
26  % 日期: 20240802

```

执行效果

就是改变1D, 2D sheet 中的表格内容, slld 的初始值中会同步改变

查

查找子模型标准信息

痛点1: 标定后, 我们一般需要记录某一天都改了哪些标定量, 一般会产生一个change 文件, 不过直接在inca 中无法批量复制这些标定量, 此脚本就是根据DCM文件, 反回所有的标定量

脚本说明

```
paramNames = findDCMNames('HY11_PCMU_Tm_OTA3_V6030303_Change.DCM')
```

代码块

```
1 function names = findDCMNames(filepath)
2 %FINDDCMNAMES 提取DCM文件中的所有参数名称
3 % NAMES = FINDDCMNAMES(FILEPATH) 从指定的DCM文件中提取所有参数名称
4 %
5 % 输入参数:
6 %     filepath      - DCM文件的完整路径
7 %
8 % 输出参数:
9 %     names         - 包含所有参数名称的列向量元胞数组
10 %
11 % 示例:
12 %     paramNames = findDCMNames('HY11_PCMU_Tm_OTA3_V6030303_Change.DCM');
13 %     disp(paramNames); % 显示所有参数名称
14 %
15 % 参见: FINDDCMPARAM, FINDDCMNAMESCHANGES
16 %
17 % 作者: Blue.ge
18 % 版本: 2.0
19 % 日期: 20250320
```

执行效果

代码块

```
1 >> paramNames = findDCMNames('HY11_PCMU_Tm_OTA3_V6030303_Change.DCM')
2
3 paramNames =
4
5 10×1 cell 数组
```

```

6
7      {'cTmAfCtrl_t_u16DeicingTime'          }
8      {'cTmColtModeMgr_Te_s32BatHvMaxPtcReqOffDeice'   }
9      {'cTmColtModeMgr_Te_s32BatHvMaxPtcReqOnDeice'   }
10     {'cTmColtModeMgr_Z_u16VSpdLowKeepTimeExitWSM3'   }
11     {'cTmComprCtrl_Te_s32AcCmprDischrgTHiCut'      }
12     {'cTmRefriModeMgr_t_s32DiffLowPTambTooMaxPTCTime' }
13     {'cTmRefriModeMgr_t_s32DiffLowPTambTooMaxTime'    }
14     {'cTmRefriModeMgr_Te_s32DeiceCalTambL'           }
15     {'cTmSwArch_D_u32CalibrationVersion'            }
16     {'tTmSigProces_t_s32ChillTbatSampPeriod'        }
17

```

查找不同阶段DCM名称有变化的标定量

痛点1：INCA根据不同的2个DCM，方便的查看两个DCM的差异，不过有时候如果你想快速查看，打开INCA并不太方便，此脚本可快速打印出不同阶段标定量名称都有哪些新增或者删除

脚本说明

```
diff = findDCMNamesChanges('HY11_PCMU_Tm_OTA3_V6010212_All.DCM',
'HY11_PCMU_Tm_OTA3_V6050327_All.DCM')
```

代码块

```

1  function changes = findDCMNamesChanges(filepath1, filepath2)
2  %FINDDCMNAMESCHANGES 比较两个DCM文件的参数差异
3  %    CHANGES = FINDDCMNAMESCHANGES(FILEPATH1, FILEPATH2) 比较两个DCM文件,
4  %    找出参数的添加、删除和修改
5  %
6  %    输入参数:
7  %        filepath1    - 基准DCM文件的完整路径
8  %        filepath2    - 比较DCM文件的完整路径
9  %
10 %    输出参数:
11 %        changes       - 包含参数变化的结构体, 具有以下字段:
12 %                           .added    - 在文件2中添加的参数列表
13 %                           .removed  - 在文件2中删除的参数列表
14 %                           .common   - 两个文件中共有的参数列表
15 %
16 %    示例:
17 %        diff = findDCMNamesChanges('HY11_PCMU_Tm_OTA3_V6010212_All.DCM',
18 %                                     'HY11_PCMU_Tm_OTA3_V6050327_All.DCM');
19 %        disp('Added parameters:');
20 %        disp(diff.added);

```

```
20 %      disp('Removed parameters:');
21 %      disp(diff.removed);
22 %
23 %    参见: FINDDCMNAMES, FINDDCMPARAM
24 %
25 %    作者: Blue.ge
26 %    版本: 1.0
27 %    日期: 20250320
```

执行效果

代码块

```
1  >> diff = findDCMNamesChanges('HY11_PCMU_Tm_OTA3_V6010212_All.DCM',
2                                 'HY11_PCMU_Tm_OTA3_V6050327_All.DCM')
3  比较结果摘要:
4      共有参数: 5150
5      添加参数: 17
6      删除参数: 10
7  -----
8  具体变化信息:
9      添加参数:
10     cTmAfCtrl_t_u16DeicingTime
11     cTmColtVlvCtrl_B_Bctv2HeatInitCtrl
12     cTmColtVlvCtrl_t_s32BCTV2HeatInitDly
13     cTmColtVlvCtrl_t_s32HeatgModeChangeDly
14     cTmPumpCtrl_L_s32HpumpFlwMin1
15     cTmPumpCtrl_L_s32HpumpFlwMin4
16     cTmPumpCtrl_Te_s32WtrTInOutlFlowLim
17     mTmDiag_Z_s32CopFromCmpPerfmaCool
18     mTmDiag_Z_s32CopFromCmpPerfmaCool_x
19     mTmDiag_Z_s32CopFromCmpPerfmaCool_y
20     mTmDiag_Z_s32CopFromCmpPerfmaHeat
21     mTmDiag_Z_s32CopFromCmpPerfmaHeat_x
22     mTmDiag_Z_s32CopFromCmpPerfmaHeat_y
23     tTmRefriModeMgr_Te_s32IcingLpSatTAmbTDiff
24     tTmRefriModeMgr_Te_s32IcingLpSatTAmbTDiff_x
25     tTmRefriVlvCtrl_Te_u16SM4TargetSC
26     tTmRefriVlvCtrl_Te_u16SM4TargetSC_X
27  -----
28     删除参数:
29     cTmAfCtrl_t_u16FanKeepTimeAfterDeice
30     cTmRefriModeMgr_P_u16HighPExitDeice
31     cTmRefriModeMgr_Te_s32DiffLowPTambHigh
32     cTmRefriVlvCtrl_Te_u16SM4TargetSC
33     mTmDiag_K_s32CopFromCmpPerfmaCool
```

```

33     mTmDiag_K_s32CopFromCmpPerfmaCool_x
34     mTmDiag_K_s32CopFromCmpPerfmaCool_y
35     mTmDiag_K_s32CopFromCmpPerfmaHeat
36     mTmDiag_K_s32CopFromCmpPerfmaHeat_x
37     mTmDiag_K_s32CopFromCmpPerfmaHeat_y
38 -----:

```

查找不同阶段DCM值有变化的标定量

痛点1：INCA根据不同的2个DCM，方便的查看两个DCM的差异，不过有时候如果你想快速查看，打开INCA并不太方便，此脚本可快速打印出不同阶段标定量名称及值都有变化

脚本说明

```
diff = findDCMValuesChanges('HY11_PCMU_Tm_OTA3_V6040303_All.DCM',
'HY11_PCMU_Tm_OTA3_V6050327_All.DCM')
```

代码块

```

1 function changes = findDCMValuesChanges(filepath1, filepath2, varargin)
2 %FINDDCMVALUESCHANGES 比较两个DCM文件中参数值的差异
3 % CHANGES = FINDDCMVALUESCHANGES(FILEPATH1, FILEPATH2) 比较两个DCM文件中
4 % 标定量的具体值，找出发生变化的参数及其变化情况
5 %
6 % CHANGES = FINDDCMVALUESCHANGES(FILEPATH1, FILEPATH2, 'Parameter', Value,
7 % ... )
8 %
9 % 使用指定的参数比较DCM文件
10 %
11 % 输入参数：
12 %
13 % 可选参数（名值对）：
14 % 'Tolerance' - 浮点数比较容差（数值），默认值：1e-6
15 % 'ShowDetails' - 是否显示详细变化信息（逻辑值），默认值：true
16 % 'OnlyCommon' - 是否只比较共有参数（逻辑值），默认值：true
17 % 'IgnoreType' - 要忽略的参数类型（元胞数组），默认值：{}
18 % 可能的值：{'FESTWERT', 'STUETZSTELLENVERTEILUNG',
19 % 'FESTWERTEBLOCK', 'GRUPPENKENNLINIE',
20 % 'GRUPPENKENNFELD'}
21 %
22 % 输出参数：
23 % changes - 包含参数值变化的结构体数组，每个结构体包含以下字段：
24 % .name - 参数名称
25 % .type - 参数类型
26 % .oldValue - 旧值

```

```

26 % .newValue - 新值
27 % .diff - 差值或差异描述
28 %
29 % 示例:
30 % diff = findDCMValuesChanges('HY11_PCMU_Tm_OTA3_V6040303_All.DCM',
31 %                                'HY11_PCMU_Tm_OTA3_V6050327_All.DCM');
32 % diff = findDCMValuesChanges('HY11_VCU_Tm_OTA2_V4100303_All_V2.DCM',
33 %                                'HY11_PCMU_Tm_OTA3_V6040303_All.DCM')
34 % diff = findDCMValuesChanges('HY11_PCMU_Tm_OTA3_V6030217_All.DCM',
35 %                                'HY11_PCMU_Tm_OTA3_V6040303_All.DCM', 'Tolerance', 1e-4, 'ShowDetails', false);
36 %
37 % 参见: FINDDCMNAMES, FINDDCMPARAM, FINDDCMNAMESCHANGES
38 % 作者: Blue.ge
39 % 版本: 1.2
40 % 日期: 20250330

```

执行效果

代码块

```

1 >> diff = findDCMValuesChanges('HY11_PCMU_Tm_OTA3_V6040303_All.DCM',
2                                'HY11_PCMU_Tm_OTA3_V6050327_All.DCM')
3 正在解析基准文件 HY11_PCMU_Tm_OTA3_V6040303_All.DCM...
4 正在解析比较文件 HY11_PCMU_Tm_OTA3_V6050327_All.DCM...
5 开始比较参数值...
6 -----
7 参数名称          类型          变化描述
8 -----
9 cTmSwArch_D_u32CalibrationVersion 常量          6.0403e+06 -> 6.05033e+06
10 (变化: 10024)
11 mTmComprCtrl_n_u16FFCmpCoolRpmNhpOsaFlow5_Val 二维表          值矩阵内容变化
12 mTmComprCtrl_n_u16FFCmpCoolRpmNhpOsaFlow7_Val 二维表          值矩阵内容变化
13 mTmComprCtrl_n_u16FFCmpCoolRpmNhpOsaFlow9_Val 二维表          值矩阵内容变化
14 mTmComprCtrl_n_u16FFCmpCoolRpm_OsaFlow5_Val 二维表          值矩阵内容变化
15 mTmComprCtrl_n_u16FFCmpCoolRpm_OsaFlow7_Val 二维表          值矩阵内容变化
16 mTmComprCtrl_n_u16FFCmpCoolRpm_OsaFlow9_Val 二维表          值矩阵内容变化

```

查找与子模型相连接的输入输出模块

痛点1：有时候如果需要对子模型的输入输出模块进行批量处理，那么第一步就需要找到其对应的句柄，然后就可以遍历所有句柄，实现相关操作，比如更改大小，改变模块相关属性等

脚本说明

[bkIn, bkOut] = findModBkConnected(gcb)

代码块

```
1 function [bkIn, bkOut] = findModBkConnected(pathMd)
2 %FINDMODBKCONNECTED 查找与模型相连的输入输出模块
3 % [BKIN, BKOUT] = FINDMODBKCONNECTED(PATHMD) 查找指定模型路径的输入输出模块
4 %
5 % 输入参数:
6 %     pathMd - 模型路径 (字符串)
7 %
8 % 输出参数:
9 %     bkIn - 输入模块句柄列表 (数组)
10 %    bkOut - 输出模块句柄列表 (数组)
11 %
12 % 功能描述:
13 %     1. 获取模型的端口信息
14 %     2. 查找输入端口连接的模块
15 %     3. 查找输出端口连接的模块
16 %     4. 处理触发和使能端口
17 %
18 % 示例:
19 %     % 基本用法
20 %     [bkIn, bkOut] = findModBkConnected(gcb)
21 %
22 % 注意事项:
23 %     1. 如果端口未连接, 对应位置返回-1
24 %     2. 支持处理触发和使能端口
25 %     3. 输入参数必须是有效的模型路径
26 %
27 % 作者: Blue.ge
28 % 版本: 1.1
29 % 日期: 20231116
```

执行效果

```

1> [bkIn, bkOut] = findModBkConnected(gcb)
2
3   bkIn =
4
5     1.0e+04 *
6
7   列 1 至 14
8
9     8.9677    8.9678    8.9679    8.9680    8.9681    8.9682    8.9683
10    8.9684    8.9685    8.9686    8.9687    8.9688    8.9689    8.9690
11
12   列 15 至 17
13
14   8.9691    8.9692    8.9693
15
16   bkOut =
17
18     1.0e+04 *
19
20   列 1 至 14
21
22    8.9849    8.9850    8.9851    8.9852    8.9853    8.9854    8.9855
23    8.9856    8.9857    8.9858    8.9859    8.9860    8.9861    8.9862
24
25   列 15 至 17
26    8.9863    8.9864    8.9865

```

如果返回值是-1，表示该模块没有连接相关模块

查找与子模型相连接的特殊端口

痛点1：当某个SWC有上千个输入输出端口的时候，如果我们要检查是否有某些端口没有连接，工作量比较大，也可能存在遗落，此脚本就是用来解决这个问题的

脚本说明

[bkIn, bkOut]=findModBkSpecific(gcb, 'condition', 'useless')

[bkIn, bkOut]=findModBkSpecific(gcb, 'condition', 'unconnected')

代码块

```

1 function [bkIn, bkOut] = findModBkSpecific(path, varargin)
2 %FINDMODBKSPECIFIC 查找模型中特定类型的模块
3 % [BKIN, BKOUT] = FINDMODBKSPECIFIC(PATH) 查找指定模型中的特定模块
4 % [BKIN, BKOUT] = FINDMODBKSPECIFIC(PATH, 'condition', CONDITION) 指定查找条件
5 %
6 % 输入参数:
7 %     path      - 模型路径 (字符串)
8 %     condition - 查找条件 (字符串)
9 %                         可选值: 'useless' - 查找接地和终止模块
10 %                      'unconnected' - 查找未连接的端口
11 %
12 % 输出参数:
13 %     bkIn    - 输入端口名称列表 (元胞数组)
14 %     bkOut   - 输出端口名称列表 (元胞数组)
15 %
16 % 功能描述:
17 %     1. 获取模型的所有输入输出模块
18 %     2. 根据条件筛选特定类型的模块
19 %     3. 返回符合条件的端口名称
20 %
21 % 示例:
22 %     % 查找未使用的模块
23 %     [bkIn, bkOut] = findModBkSpecific(gcb, 'condition', 'useless')
24 %
25 %     % 查找未连接的端口
26 %     [bkIn, bkOut] = findModBkSpecific(gcb, 'condition', 'unconnected')
27 %
28 % 注意事项:
29 %     1. 输入参数必须是有效的模型路径
30 %     2. condition参数必须是'useless'或'unconnected'
31 %     3. 返回的是端口名称而不是模块句柄
32 %
33 % 作者: Blue.ge
34 % 版本: 1.1
35 % 日期: 20231117

```

执行效果

代码块

```

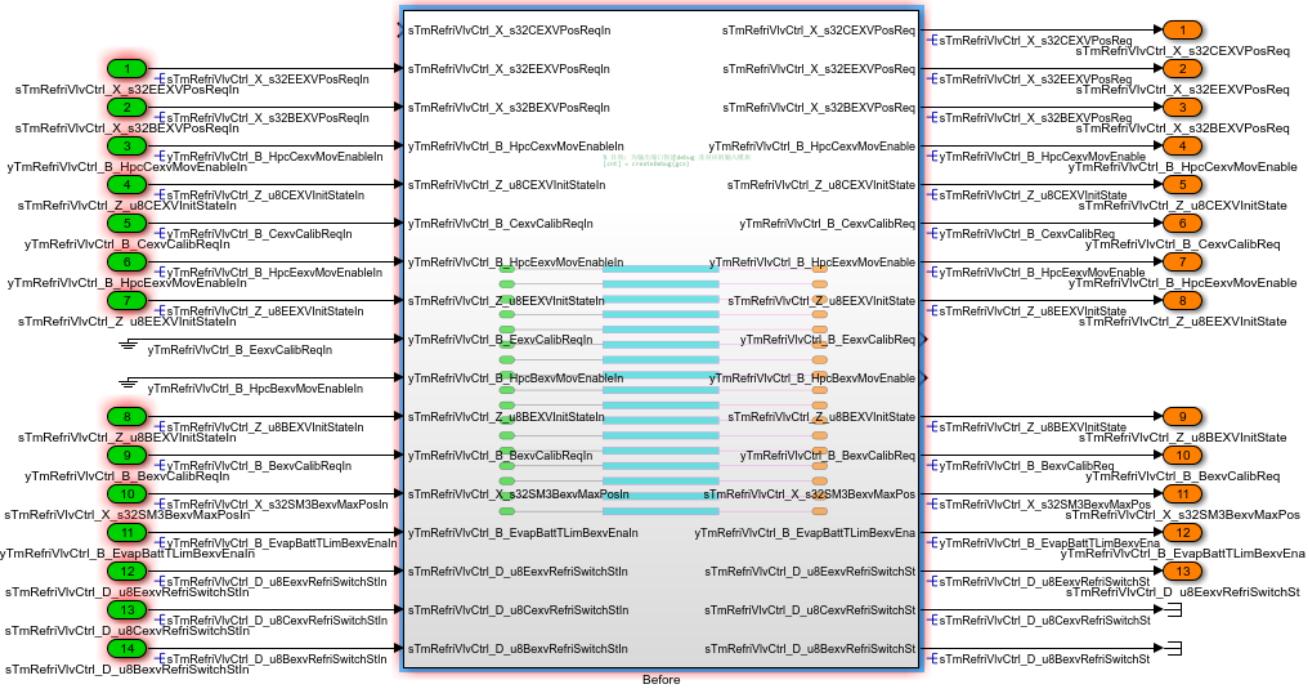
1 >> [bkIn, bkOut] = findModBkSpecific(gcb, 'condition', 'useless')
2 输入端口-useless:
3     yTmRefriVlvCtrl_B_EexvCalibReqIn
4     yTmRefriVlvCtrl_B_HpcBexvMovEnableIn
5 输出端口-useless:
6     sTmRefriVlvCtrl_D_u8CexvRefriSwitchSt

```

```

7      sTmRefriVlvCtrl_D_u8BexvRefriSwitchSt
8
9
10     >> [bkIn, bkOut] = findModBkSpecific(gcb, 'condition', 'unconnected')
11     输入端口-unconnected:
12         sTmRefriVlvCtrl_X_s32CEXVPosReqIn
13     输出端口-unconnected:
14         yTmRefriVlvCtrl_B_EexvCalibReq
15         yTmRefriVlvCtrl_B_HpcBexvMovEnable

```



查找子模型端口属性

痛点1：我们需要经常对模型的输入输出端口进行操作，需要获取端口的句柄，属性等，便于后续进一步分析

脚本说明

```

% [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Path');
% [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Name');
% [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'OutDataTypeStr');

```

代码块

```
1 function [ModelName, PortsIn, PortsOut, PortsSpecial] = findModPorts(pathMd,
```

```

varargin)
2 %FINDMODPORTS 查找模型中的输入输出端口
3 % [MODELNAME, PORTSIN, PORTSOUT, PORTSSPECIAL] = FINDMODPORTS(PATHMD) 使用默认
参数查找端口
4 % [MODELNAME, PORTSIN, PORTSOUT, PORTSSPECIAL] = FINDMODPORTS(PATHMD,
'Parameter', Value, ...) 使用指定参数查找
5 %
6 % 输入参数:
7 % pathMd - 模型路径或句柄 (字符串或数值)
8 %
9 % 可选参数 (名值对) :
10 % 'getType' - 返回端口的属性类型 (字符串), 默认值: 'Path'
11 % 可选值: 'path'(完整路径), 'Name'(端口名称), 'Handle'(句柄)等
12 % 'FiltUnconnected' - 是否只返回未连接的端口 (逻辑值), 默认值: false
13 %
14 % 输出参数:
15 % ModelName - 模型名称 (字符串)
16 % PortsIn - 输入端口列表 (元胞数组)
17 % PortsOut - 输出端口列表 (元胞数组)
18 % 格式同PortsIn
19 % PortsSpecial - 特殊端口信息 (变量)
20 %
21 % 功能描述:
22 % 查找指定模型中的输入输出端口, 可以选择返回端口的不同属性。
23 % 支持过滤未连接的端口和跳过触发端口。
24 %
25 % 示例:
26 % [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Path');
27 % [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Name');
28 % [name, inPorts, outPorts] = findModPorts(gcb, 'getType',
'OutDataTypeStr');
29 % [name, inPorts, outPorts] = findModPorts(gcs, 'FiltUnconnected', true);
30 %
31 % 参见: FIND_SYSTEM, GET_PARAM, BDROOT
32 %
33 % 作者: Blue.ge
34 % 版本: 1.1
35 % 日期: 20250312

```

执行效果

代码块

```

1 >> findModPorts(gcb, 'getType', 'Path');
2 -----
3 输入端口:

```

```
4     ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32CEXVPosReqIn
5     ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32EEXVPosReqIn
6     ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32BEXVPosReqIn
7
8     ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_HpcCexvMovEnableIn
9
10    ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_Z_u8CEXVInitStateIn
11       ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_CexvCalibReqIn
12
13    ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_HpcEexvMovEnableIn
14
15    ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_Z_u8EEXVInitStateIn
16       ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_EexvCalibReqIn
17
18    ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32SM3BexvMaxPosIn
19
20    ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_EvapBattTLimBexvEnaI
n
21
22    ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_D_u8EexvRefriSwitchStI
n
23
24    ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_D_u8CexvRefriSwitchStI
n
25
26    ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_D_u8BexvRefriSwitchStI
n
27
28    -----
29    输出端口：
30
31      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32CEXVPosReq
32      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32EEXVPosReq
33      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32BEXVPosReq
34      ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_HpcCexvMovEnable
35      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_Z_u8CEXVInitState
36      ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_CexvCalibReq
37      ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_HpcEexvMovEnable
38      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_Z_u8EEXVInitState
39      ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_EexvCalibReq
40      ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_HpcBexvMovEnable
41      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_Z_u8BEXVInitState
42      ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_BexvCalibReq
43      ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_X_s32SM3BexvMaxPos
```

```

36     ScriptDemo/Find/findModBkSpecific/Before/yTmRefriVlvCtrl_B_EvapBattTLimBexvEna
37
38     ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_D_u8EexvRefriSwitchSt
39
40     ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_D_u8CexvRefriSwitchSt
41
42     ScriptDemo/Find/findModBkSpecific/Before/sTmRefriVlvCtrl_D_u8BexvRefriSwitchSt
43     -----
44     >> findModPorts(gcb, 'getType', 'Name');
45     -----
46     输入端口:
47
48     sTmRefriVlvCtrl_X_s32CEXVPosReqIn
49     sTmRefriVlvCtrl_X_s32EEXVPosReqIn
50     sTmRefriVlvCtrl_X_s32BEXVPosReqIn
51     yTmRefriVlvCtrl_B_HpcCexvMovEnableIn
52     sTmRefriVlvCtrl_Z_u8CEXVIInitStateIn
53     yTmRefriVlvCtrl_B_CexvCalibReqIn
54     yTmRefriVlvCtrl_B_HpcEexvMovEnableIn
55     sTmRefriVlvCtrl_Z_u8EEXVIInitStateIn
56     yTmRefriVlvCtrl_B_EexvCalibReqIn
57     yTmRefriVlvCtrl_B_HpcBexvMovEnableIn
58     sTmRefriVlvCtrl_Z_u8BEXVIInitStateIn
59     yTmRefriVlvCtrl_B_BexvCalibReqIn
60     sTmRefriVlvCtrl_X_s32SM3BexvMaxPosIn
61     yTmRefriVlvCtrl_B_EvapBattTLimBexvEnaIn
62     sTmRefriVlvCtrl_D_u8EexvRefriSwitchStIn
63     sTmRefriVlvCtrl_D_u8CexvRefriSwitchStIn
64     sTmRefriVlvCtrl_D_u8BexvRefriSwitchStIn
65
66     -----
67     输出端口:
68
69     sTmRefriVlvCtrl_X_s32CEXVPosReq
70     sTmRefriVlvCtrl_X_s32EEXVPosReq
71     sTmRefriVlvCtrl_X_s32BEXVPosReq
72     yTmRefriVlvCtrl_B_HpcCexvMovEnable
73     sTmRefriVlvCtrl_Z_u8CEXVIInitState
74     vTmRefriVlvCtrl_B_CexvCalibReq

```

其中Name属性，可以导出子模型的输入输出信号，OutDataTypeStr属性可以查看是否有些端口的数据类型是为指定的

根据一定的规则，查找子模型名称

为了标准化建模，建议子模型的名称，尽可能绑定对应的软件需求及功能需求，比如
un666666_Req26245_DemoSub1，不过我们真正的模型名称为*DemoSub1*，因此需要从中提取出真

正的模型名称

脚本说明

[modelName, funID, reqID] = findNameMd('Fun666666_Req26245_DemoSub1')

```
function [ModelName, FunctionID, RequirementID] = findNameMd(Name, varargin)%FIND…  
1 function [ModelName, PortsIn, PortsOut, PortsSpecial] = findModPorts(pathMd,  
varargin)  
2 %FINDMODPORTS 查找模型中的输入输出端口  
3 % [MODELNAME, PORTSIN, PORTSOUT, PORTSSPECIAL] = FINDMODPORTS(PATHMD) 使用默认  
参数查找端口  
4 % [MODELNAME, PORTSIN, PORTSOUT, PORTSSPECIAL] = FINDMODPORTS(PATHMD,  
'Parameter', Value, ...) 使用指定参数查找  
5 %  
6 % 输入参数：  
7 % pathMd - 模型路径或句柄 (字符串或数值)  
8 %  
9 % 可选参数 (名值对)：  
10 % 'getType' - 返回端口的属性类型 (字符串), 默认值: 'Path'  
11 % 可选值: 'path'(完整路径), 'Name'(端口名称), 'Handle'(句柄)等  
12 % 'FiltUnconnected' - 是否只返回未连接的端口 (逻辑值), 默认值: false  
13 %  
14 % 输出参数：  
15 % ModelName - 模型名称 (字符串)  
16 % PortsIn - 输入端口列表 (元胞数组)  
17 % PortsOut - 输出端口列表 (元胞数组)  
18 % 格式同PortsIn  
19 % PortsSpecial - 特殊端口信息 (变量)  
20 %  
21 % 功能描述：  
22 % 查找指定模型中的输入输出端口，可以选择返回端口的不同属性。  
23 % 支持过滤未连接的端口和跳过触发端口。  
24 %  
25 % 示例：  
26 % [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Path');  
27 % [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Name');  
28 % [name, inPorts, outPorts] = findModPorts(gcb, 'getType',  
'OutDataTypeStr');  
29 % [name, inPorts, outPorts] = findModPorts(gcs, 'FiltUnconnected', true);  
30 %  
31 % 参见: FIND_SYSTEM, GET_PARAM, BDROOT  
32 %  
33 % 作者: Blue.ge  
34 % 版本: 1.1  
35 % 日期: 20250312
```

执行效果

代码块

```
1 >> [modelName, funID, reqID] = findNameMd('Fun666666_Req26245_DemoSub1')
2
3 modelName =
4
5     'DemoSub1'
6
7 funID =
8
9     666666
10
11 reqID =
12
13     26245
```

查找子模型的输出名称

为了更好的执行建模规范，对输入模型的名称会有所要求，比如SM1，SM2子模型中都会输出Kp值，最终通过merge 到一起变成sTmComprCtrl_D_u32AccmKp，那么每个子模型的输出信号需要区别开来

脚本说明

```
[type, outName] = findNameMdOut('sTmComprCtrl_D_u32AccmKp', 'mode', 'keep')
[type, outName] = findNameMdOut('sTmComprCtrl_D_u32AccmKp', 'mode', 'pre')
[type, outName] = findNameMdOut('sTmComprCtrl_D_u32AccmKp', 'mode', 'tail')
```

```
function [dataType, nameOutPort] = findNameMdOut(Name, varargin)%FINDNAMEMDOUT
1 function [ModelName, PortsIn, PortsOut, PortsSpecial] = findModPorts(pathMd,
varargin)
2 %FINDMODPORTS 查找模型中的输入输出端口
3 % [MODELNAME, PORTSIN, PORTSOUT, PORTSSPECIAL] = FINDMODPORTS(PATHMD) 使用默认
参数查找端口
4 % [MODELNAME, PORTSIN, PORTSOUT, PORTSSPECIAL] = FINDMODPORTS(PATHMD,
'Parameter', Value, ...) 使用指定参数查找
5 %
6 % 输入参数:
7 % pathMd          - 模型路径或句柄 (字符串或数值)
8 %
9 % 可选参数 (名值对) :
10 % 'getType'      - 返回端口的属性类型 (字符串), 默认值: 'Path'
```

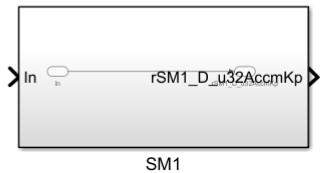
```

11 % 可选值: 'path' (完整路径), 'Name' (端口名称), 'Handle' (句柄)等
12 % 'FiltUnconnected' - 是否只返回未连接的端口 (逻辑值), 默认值: false
13 %
14 % 输出参数:
15 %     ModelName      - 模型名称 (字符串)
16 %     PortsIn        - 输入端口列表 (元胞数组)
17 %     PortsOut       - 输出端口列表 (元胞数组)
18 %                           格式同PortsIn
19 %     PortsSpecial   - 特殊端口信息 (变量)
20 %
21 % 功能描述:
22 %     查找指定模型中的输入输出端口, 可以选择返回端口的不同属性。
23 %     支持过滤未连接的端口和跳过触发端口。
24 %
25 % 示例:
26 %     [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Path');
27 %     [name, inPorts, outPorts] = findModPorts(gcb, 'getType', 'Name');
28 %     [name, inPorts, outPorts] = findModPorts(gcb, 'getType',
29 %         'OutDataTypeStr');
30 %
31 %     [name, inPorts, outPorts] = findModPorts(gcs, 'FiltUnconnected', true);
32 %
33 % 参见: FIND_SYSTEM, GET_PARAM, BDROOT
34 %
35 % 作者: Blue.ge
36 % 版本: 1.1
37 % 日期: 20250312

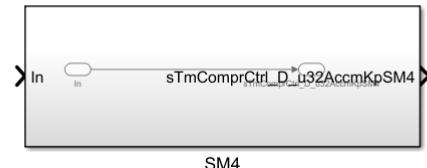
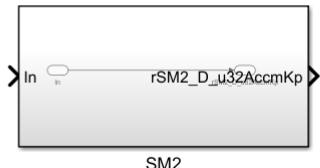
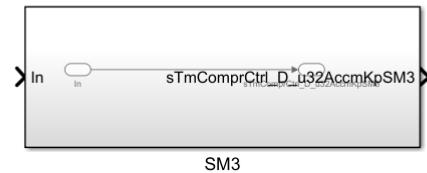
```

执行效果

[type, outName] = findNameMdOut('sTmComprCtrl_D_u32AccmKp', 'mode', 'pre')



[type, outName] = findNameMdOut('sTmComprCtrl_D_u32AccmKp', 'mode', 'tail')



查找模型的所有的标定量

我们经常需要维护slldb表格，不过如果你经常新增一些标定量，得一个个添加，通过此脚本可以批量找出所有的标定量

脚本说明

$[PathAll, ParamAll] = findParameters(gcs)$

findParameters

```
1 function [PathAll, ParamAll] = findParameters(path)
2 %FINDPARAMETERS 查找模型中所有的标定量
3 % [PathAll, ParamAll] = findParameters(path) 查找指定路径下模型中所有的标定量,
4 % 包括常量、1维表、2维表等。
5 %
6 % 输入参数:
7 %     path - 模型路径, 可以是模型名称或路径字符串
8 %
9 % 输出参数:
10 %      PathAll - 所有标定量的路径
11 %      ParamAll - 所有标定量的名称
12 %
13 % 示例:
14 %      [PathAll, ParamAll] = findParameters(bdroot)
15 %      [PathAll, ParamAll] = findParameters(gcs)
16 %
17 % 作者: 葛维冬 (Blue Ge)
18 % 日期: 2023-10-27
19 % 版本: 1.1
```

执行效果

代码块

```
1 PathAll =
2
3     17×1 cell 数组
4
5     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug'  }
6     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug1'  }
7     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug10' }
8     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug11' }
9     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug12' }
10    {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug13' }
11    {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug14' }
12    {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug15' }
13    {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug16' }
14    {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug2'  }
```

```

15     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug3' }
16     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug4' }
17     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug5' }
18     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug6' }
19     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug7' }
20     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug8' }
21     {'ScriptDemo/Find/findModBkSpecific/Before/OutDebug9' }

22
23
24 ParamAll =
25
26 34×1 cell 数组
27
28     {'cTmRefriVlvCtrl_B_BexvCalibReq_db'          }
29     {'cTmRefriVlvCtrl_B_BexvCalibReq_sw'          }
30     {'cTmRefriVlvCtrl_B_CexvCalibReq_db'          }
31     {'cTmRefriVlvCtrl_B_CexvCalibReq_sw'          }
32     {'cTmRefriVlvCtrl_B_EexvCalibReq_db'          }
33     {'cTmRefriVlvCtrl_B_EexvCalibReq_sw'          }
34     {'cTmRefriVlvCtrl_B_EvapBattTLimBexvEna_db'  }
35     {'cTmRefriVlvCtrl_B_EvapBattTLimBexvEna_sw'  }
36     {'cTmRefriVlvCtrl_B_HpcBexvMovEnable_db'      }
37     {'cTmRefriVlvCtrl_B_HpcBexvMovEnable_sw'       }
38     {'cTmRefriVlvCtrl_B_HpcCexvMovEnable_db'       }
39     {'cTmRefriVlvCtrl_B_HpcCexvMovEnable_sw'       }
40     {'cTmRefriVlvCtrl_B_HpcEexvMovEnable_db'       }
41     {'cTmRefriVlvCtrl_B_HpcEexvMovEnable_sw'       }
42     {'cTmRefriVlvCtrl_D_u8BexvRefriSwitchSt_db'   }
43     {'cTmRefriVlvCtrl_D_u8BexvRefriSwitchSt_sw'   }
44     {'cTmRefriVlvCtrl_D_u8CexvRefriSwitchSt_db'   }
45     {'cTmRefriVlvCtrl_D_u8CexvRefriSwitchSt_sw'   }
46     {'cTmRefriVlvCtrl_D_u8EexvRefriSwitchSt_db'   }
47     {'cTmRefriVlvCtrl_D_u8EexvRefriSwitchSt_sw'   }
48     {'cTmRefriVlvCtrl_X_s32BEXVPosReq_db'         }
49     {'cTmRefriVlvCtrl_X_s32BEXVPosReq_sw'         }
50     {'cTmRefriVlvCtrl_X_s32CEXVPosReq_db'         }
51     {'cTmRefriVlvCtrl_X_s32CEXVPosReq_sw'         }
52     {'cTmRefriVlvCtrl_X_s32EEXVPosReq_db'         }
53     {'cTmRefriVlvCtrl_X_s32EEXVPosReq_sw'         }
54     {'cTmRefriVlvCtrl_X_s32SM3BexvMaxPos_db'      }
55     {'cTmRefriVlvCtrl_X_s32SM3BexvMaxPos_sw'      }
56     {'cTmRefriVlvCtrl_Z_u8BEXVInitState_db'        }
57     {'cTmRefriVlvCtrl_Z_u8BEXVInitState_sw'        }
58     {'cTmRefriVlvCtrl_Z_u8CEXVInitState_db'        }
59     {'cTmRefriVlvCtrl_Z_u8CEXVInitState_sw'        }
60     {'cTmRefriVlvCtrl_Z_u8EEXVInitState_db'        }
61     {'cTmRefriVlvCtrl_Z_u8EEXVInitState_sw'        }

```

查找并导出模型的所有的标定量

我们经常需要维护slddb表格，不过如果你经常新增一些标定量，得一个个添加，通过此脚本可以批量找出所有的标定量，最终导出成excel 格式的sldd

脚本说明

```
% [SignalPCMU, SignalVCU, ParamPCMU, ParamVCU] = findSldd(bdroot)
% [SignalPCMU, SignalVCU, ParamPCMU, ParamVCU] = findSldd(gcs)
```

findSldd

```
1 function [SignalPCMU, SignalVCU, ParamPCMU, ParamVCU] = findSldd(pathMd)
2 %FINDSLDD 查找模型中的SLDD数据
3 % [SignalPCMU, SignalVCU, ParamPCMU, ParamVCU] = findSldd(pathMd) 查找指定模型
4 % 中的SLDD数据,
5 %
6 % 输入参数:
7 % pathMd - 模型路径, 可以是模型名称或路径字符串
8 %
9 % 输出参数:
10 % SignalPCMU - PCMU相关的SLDD数据
11 % SignalVCU - VCU相关的SLDD数据
12 % ParamPCMU - PCMU相关的SLDD数据
13 % ParamVCU - VCU相关的SLDD数据
14 %
15 % 示例:
16 % [SignalPCMU, SignalVCU, ParamPCMU, ParamVCU] = findSldd(bdroot)
17 % [SignalPCMU, SignalVCU, ParamPCMU, ParamVCU] = findSldd(gcs)
18 %
19 % 作者: 葛维冬 (Blue Ge)
20 % 日期: 2023-10-26
21 % 版本: 1.1
```

执行效果

会生成findModBkSpecific_DD_PCMU_EXPORT.xlsx 的一个表格

ModelName	PortType	Name	DataType	CustomStorageClass	DefinitionFile	RTE_Interface	Dimensions	Details	ValueTable	Unit	InValue	Min	Max	DataTypeSelect	CustomStorageClassSelect	DefinitionFile
findModBkSpecific	Input	sTmRefriVlvCtrl_X_s32EXVPosReqIn	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_X_s32REXVPosReqIn	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	yTmRefriVlvCtrl_B_HpcCexvMovEnableIn	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_Z_u8CEXVInitStateIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	yTmRefriVlvCtrl_B_CexvCalibReqIn	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	yTmRefriVlvCtrl_B_HpcCexvMovEnableIn	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_Z_u8EEKVInitStateIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_Z_u8BEKVInitStateIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	yTmRefriVlvCtrl_B_BexvCalibReqIn	boolean	Global1	TmModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_X_s32SM3BexvMaxPosIn	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	yTmRefriVlvCtrl_B_EvapBattLimBexvEnaIn	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_D_u8ExvRefriSwitchStIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_D_u8CexvRefriSwitchStIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Input	sTmRefriVlvCtrl_D_u8ExvRefriSwitchStIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_X_s32CEXVPosReq	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_X_s32REXVPosReq	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_X_s32CEXVPosReq	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	yTmRefriVlvCtrl_B_HpcCexvMovEnable	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_Z_u8EEKVInitState	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_Z_u8BEKVInitState	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	yTmRefriVlvCtrl_B_BexvCalibReq	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	yTmRefriVlvCtrl_B_HpcCexvMovEnable	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_X_s32SM3BexvMaxPos	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	yTmRefriVlvCtrl_B_EvapBattLimBexvEna	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Output	sTmRefriVlvCtrl_D_u8ExvRefriSwitchSt	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	yTmRefriVlvCtrl_B_EvapBattLimBexvEnaIn	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	sTmRefriVlvCtrl_Z_u8EEKVInitState	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	sTmRefriVlvCtrl_Z_u8BEKVInitState	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	yTmRefriVlvCtrl_B_BexvCalibReq	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	sTmRefriVlvCtrl_X_s32CEXVPosReqIn	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	sTmRefriVlvCtrl_Z_u8EEKVInitStateIn	uint8	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	sTmRefriVlvCtrl_X_s32REXVPosReqIn	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	sTmRefriVlvCtrl_X_s32CEXVPosReqIn	single	Global1	findModBkSpecific.c	-1			inherit	[]						
findModBkSpecific	Local	yTmRefriVlvCtrl_B_HpcCexvMovEnableIn	boolean	Global1	findModBkSpecific.c	-1			inherit	[]						

查找并导入模型的所有的标定量

模型仿真及生代码之前，都需要导入对应的slld，传统方法是使用上位机导入，而且信号和标定量得分开导入，比较麻烦且费时间，此脚本就是用来解决这个问题

脚本说明

```
% [sigCnts, paramCnts] = findSlldLoad('TmComprCtrl_DD_PCMU.xlsx')
%
% [sigCnts, paramCnts] = findSlldLoad('TmComprCtrl', 'mode', 'PCMU')
%
% [sigCnts, paramCnts] = findSlldLoad('TmComprCtrl', 'mode', 'PCMU', 'exclude', {'Input', 'Output'})
```

```
function [sigCnts, paramCnts] = findSlldLoad(path, varargin)
%FINDSLDDLOAD 导入SLDD数据到工作空间
% [sigCnts, paramCnts] = findSlldLoad(path, varargin) 从Excel文件或模型导入SLDD
% 数据
% 到工作空间，支持PCMU和VCU两种模式。
%
%
% 输入参数：
% path - SLDD文件路径或模型名称
% varargin - 可选参数对，包括：
% 'mode' - 模式选择， 'PCMU'或'VCU'，默认为 'PCMU'
% 'exclude' - 需要排除的模块类型，默认为空
%
% 输出参数：
```

```
13 %      sigCnts - 成功导入的信号数量
14 %      paramCnts - 成功导入的参数数量
15 %
16 %    示例:
17 %      [sigCnts, paramCnts] = findSldLoad('TmComprCtrl_DD_PCMU.xlsx')
18 %      [sigCnts, paramCnts] = findSldLoad('TmComprCtrl', 'mode', 'PCMU')
19 %      [sigCnts, paramCnts] = findSldLoad('TmComprCtrl', 'mode', 'PCMU',
20 %      'exclude', {'Input', 'Output'})
21 %
22 %    作者: 葛维冬 (Blue Ge)
23 %    日期: 2024-02-05
24 %    版本: 1.1
```

执行效果

将表格形式的sldd 导入到工作空间中，便于模型的仿真及代码生成