1. 封FMU

·VS C/C++代码

·环境配置 头文件包含路径 bin文件目录 生成FMU目录

·输入输出 同一个类型的输入输出数量定义到一个宏上 可以定义大一些 避免不够

·逻辑处理 从输入获得数据 处理后通过输出对外体现

·Rhapsody 流程图等封装

·环境配置 需要修改文件msvcmake.bat中vs的路径，修改绝对路径，到最外层目录就可以，或者配置环境变量

·流程图 体现处理逻辑和过程 有固定的输入参数switcher(bool类型)和事件chSwitcher 才能被wrp识别

·面板图 调试

·生成FMU

·matlab 封装

·环境配置

·封装

1. WRP使用

静态建模 架构

动态建模 调度

时序仿真：

双V脚本：数据输入的一种手段，在没有与下位机连接，无数据输入情况下，构造数据输入，测试系统准确性；也可以构造异常，检测系统在异常情况下的处理。可以提升易用性

·可以多个双V脚本在同一个节拍修改同一个变量的值，这种情况下不确定哪个修改是有效的

·当前存在保持的效果 比如先给变量a 从0~99999拍一个随机值；在100拍的时候使用set命令将a赋值为100，持续1拍，那在之后a的值会一直是100。这样的设计会导致我每次构造完异常后还得想办法给他恢复到正常的运行中，还得增加其它的命令。我认为比较合理的效果是：我先设置一系列的数据输入，比如这里的从0~99999拍每一拍都生成一个随机值，这是供系统正常运行的一个数据；然后我在系统正常运行的基础上去构造异常，比如就是在100拍的时候将a赋值为100，持续一拍，然后异常构造完后，我期望系统可以按照我最开始设计的可以继续正常运行。直到遇到我设置的下一个异常数据或者一直到结束。这个实现的效果就是一个变量在某一拍时候的值由几个值一起控制，如果不好确定应该是什么值，可以通过设置优先级来控制，正常运行的值就是最低的优先级，如果有其它类型的值就可以覆盖正常运行的值，

·