

Custom Devices 生成与编程 操作手册

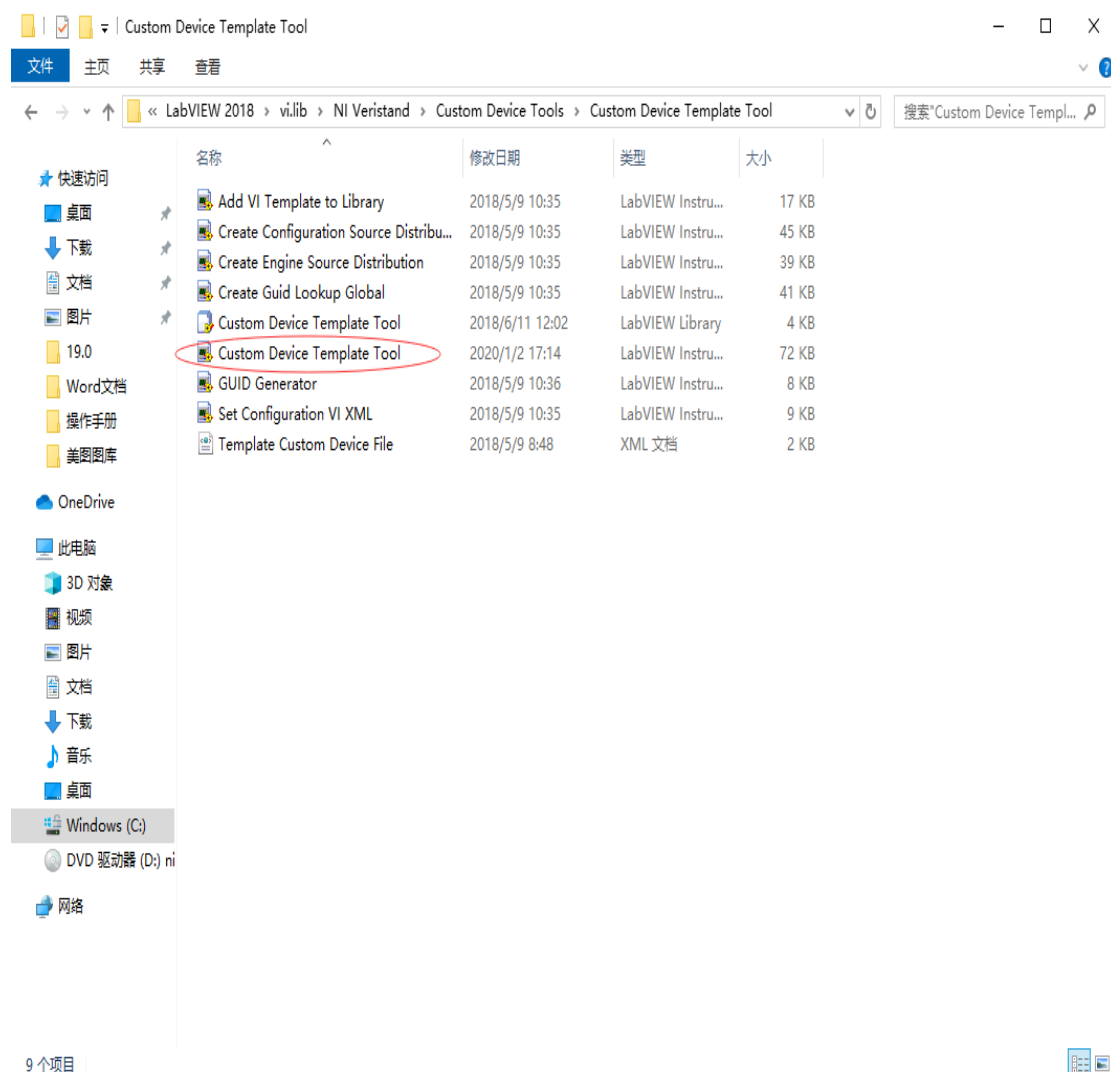
目录

创建 Custom Devices 项目模板	3
Custom Devices 项目编程.....	5
创建 Custom Devices 在 CompactRIO 平台运行文件	10

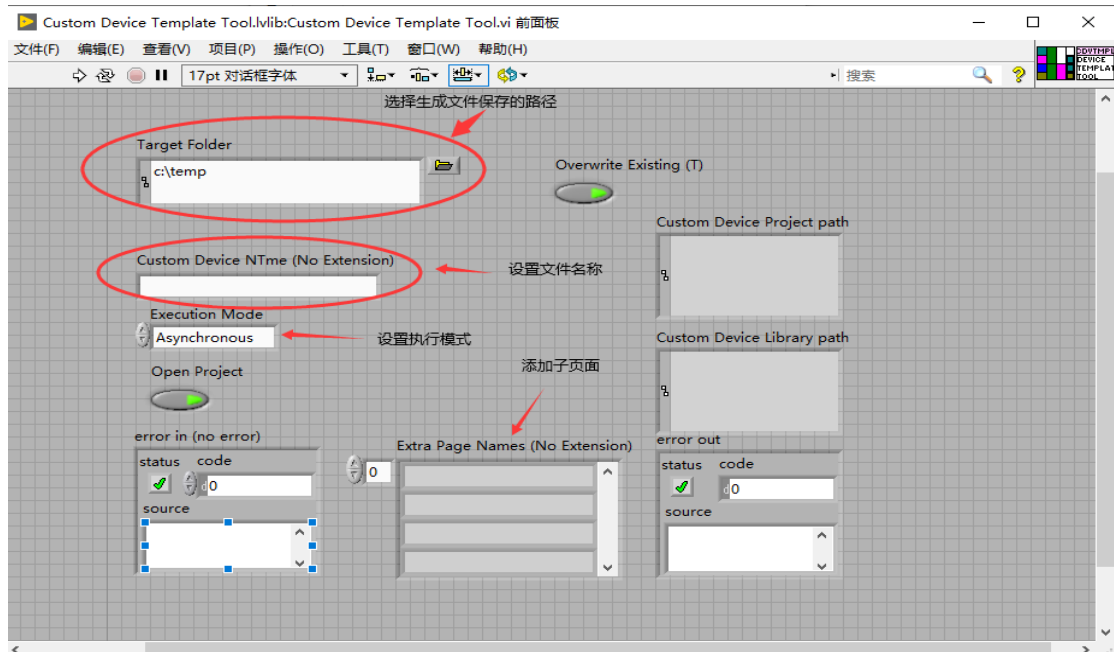
创建 Custom Devices 项目模板

1. 电脑中已安装 Labview 软件
2. 安装了 Labview 软件后，本地磁盘中会生成 Custom Devices 模板生成工具，点击进入模板生成工具，

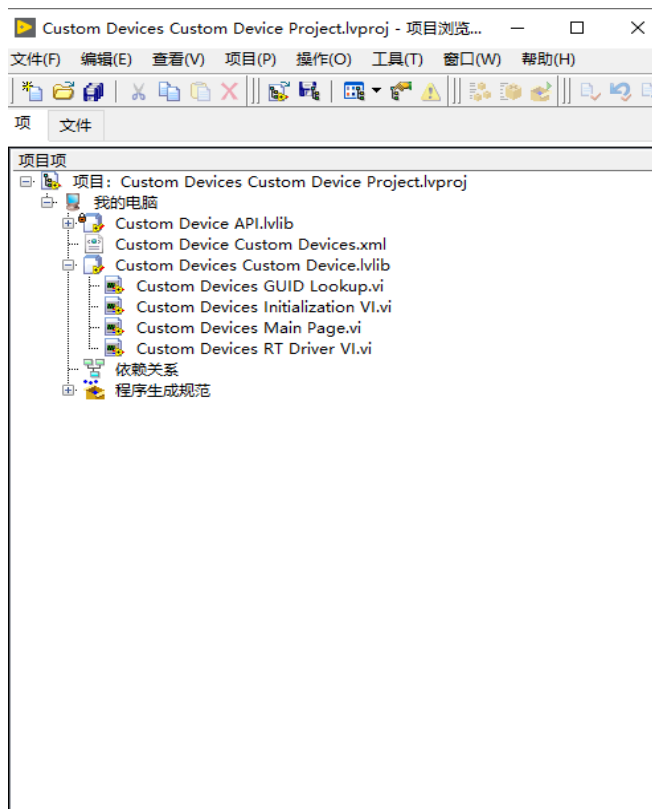
地 址：C:\Program Files (x86)\National Instruments\LabVIEW 2018\vi.lib\NI Veristand\Custom Device Tools\Custom Device Template Tool\ Custom Device Template Tool.vi



- 配置生成工具参数，选择文件生成路径，输入 Custom Devices 项目名称，选择执行模式，有需要时可以添加子页面，点击左上角运行按钮，就可以自动生成 Custom Devices 模板

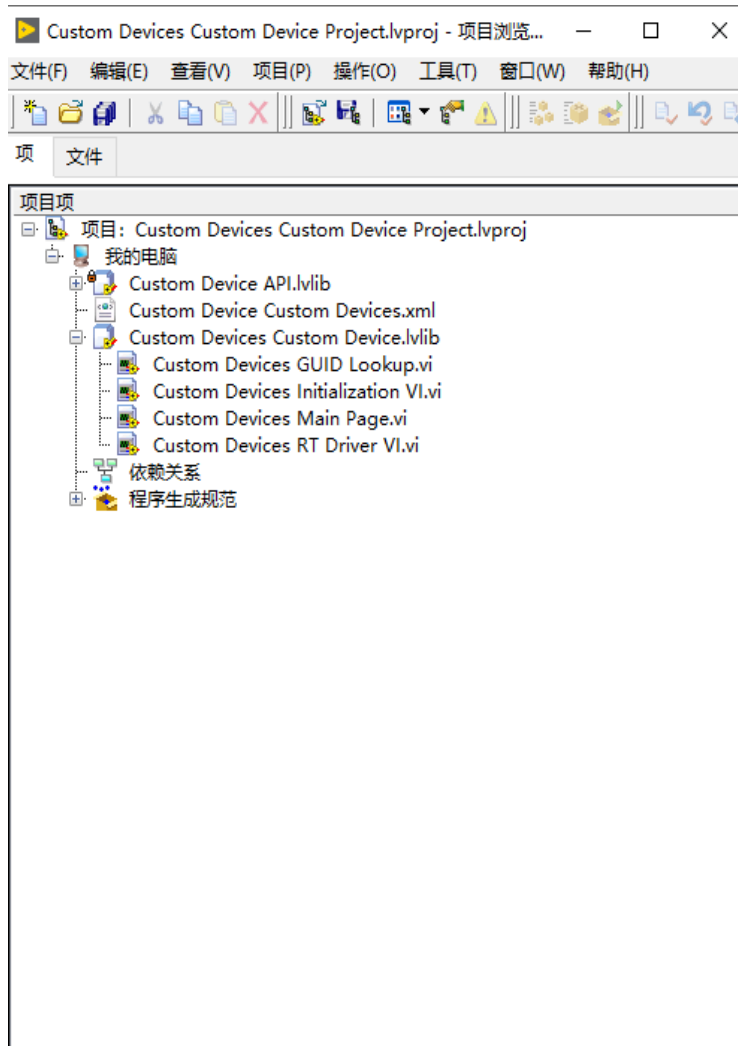


- 查看生成的 Custom Devices 模板

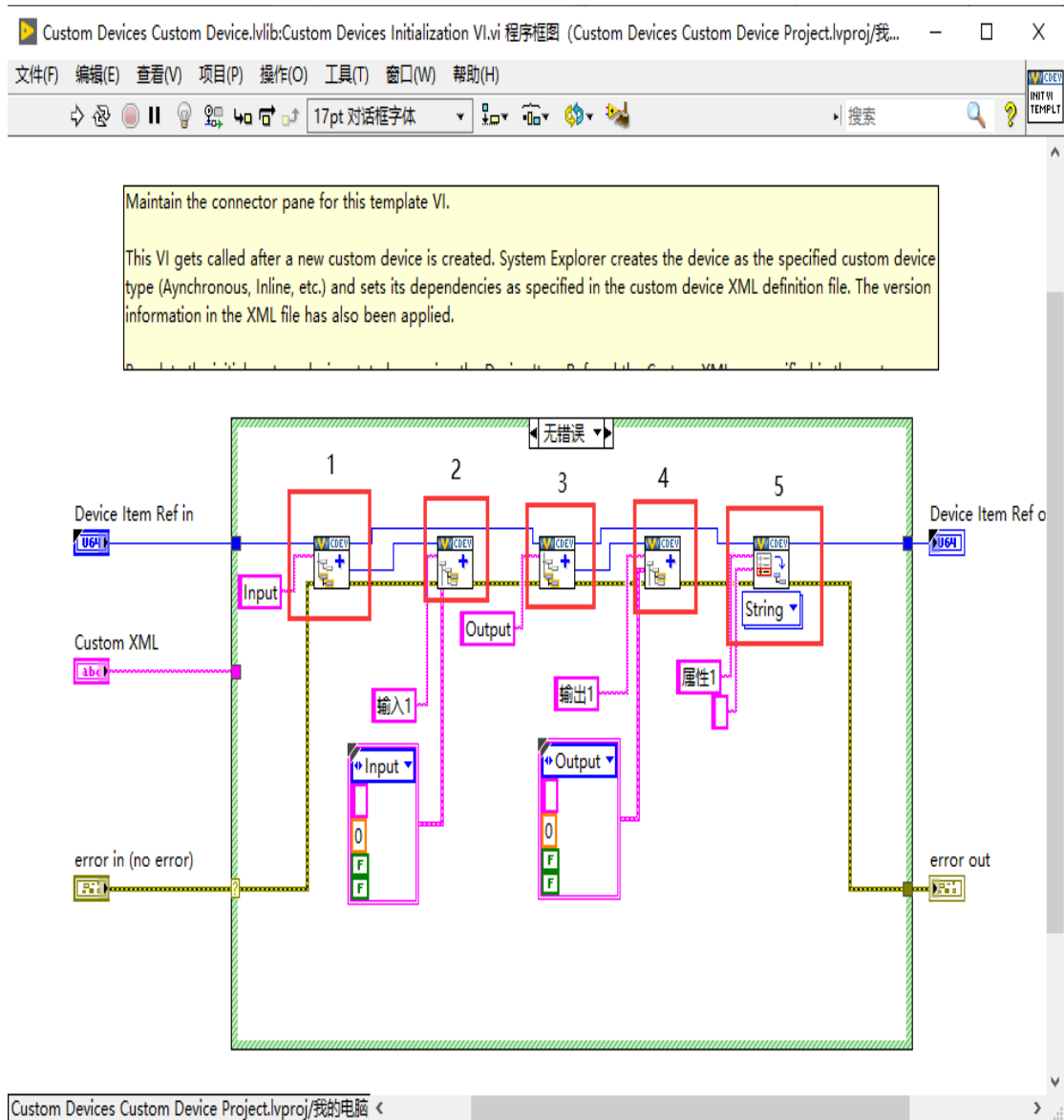


Custom Devices 项目编程

1. 打开 Custom Devices 项目，
Initialization.vi 实现的功能是当 Veristand 中创建该 Custom Devices 时，为 Custom Devices 创建相对应的属性和通道；
Main Page.vi 实现的功能是在 Veristand 中显示 Custom Devices 的属性和通道的信息；
RT Driver.vi 是 Custom Devices 的执行核心，也是最终在 RT 环境下运行的程序，Custom Devices 调用外设和数据处理的功能就是在这个 vi 中实

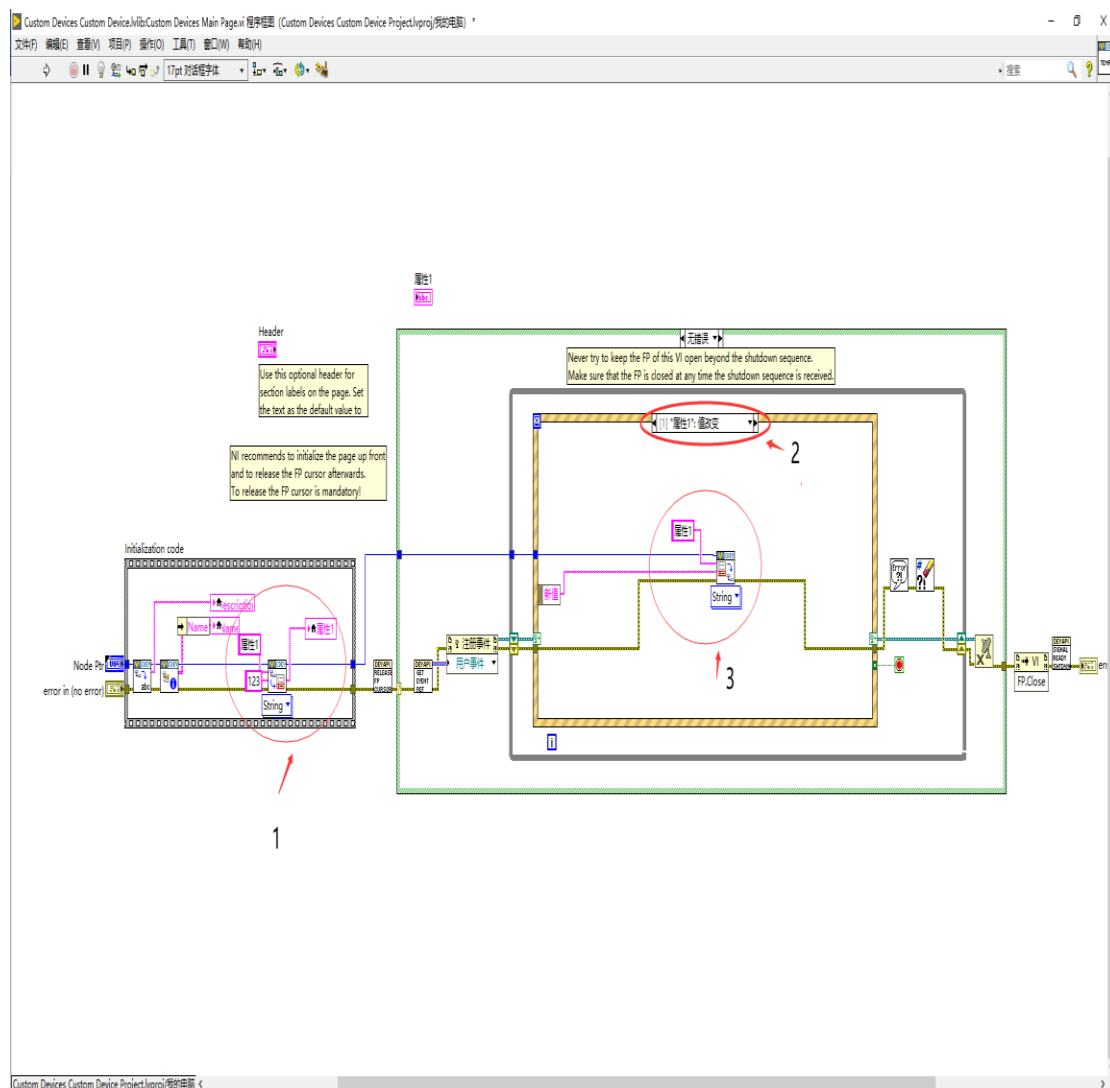


2. 点击打开 Initialization VI.vi, 1 和 3 中使用的 vi 是 Add Custom Device Section, 实现逻辑上的分组; 2 和 4 中使用的 vi 是 Add Custom Device Channel, 为各分组添加通道; 5 中使用的 vi 是 Set Item Property, 定义设备属性, 各 vi 位置: 右键-NI Veristand-Custom Devices API-Configuration

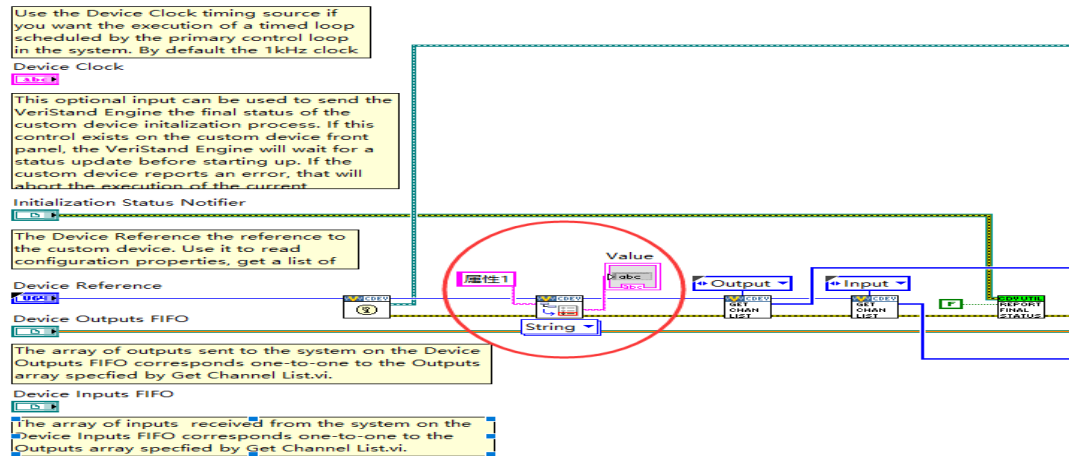


3. 打开 Main Page.vi, , Main page 中采用了一个顺序帧结构和一个事件结构。

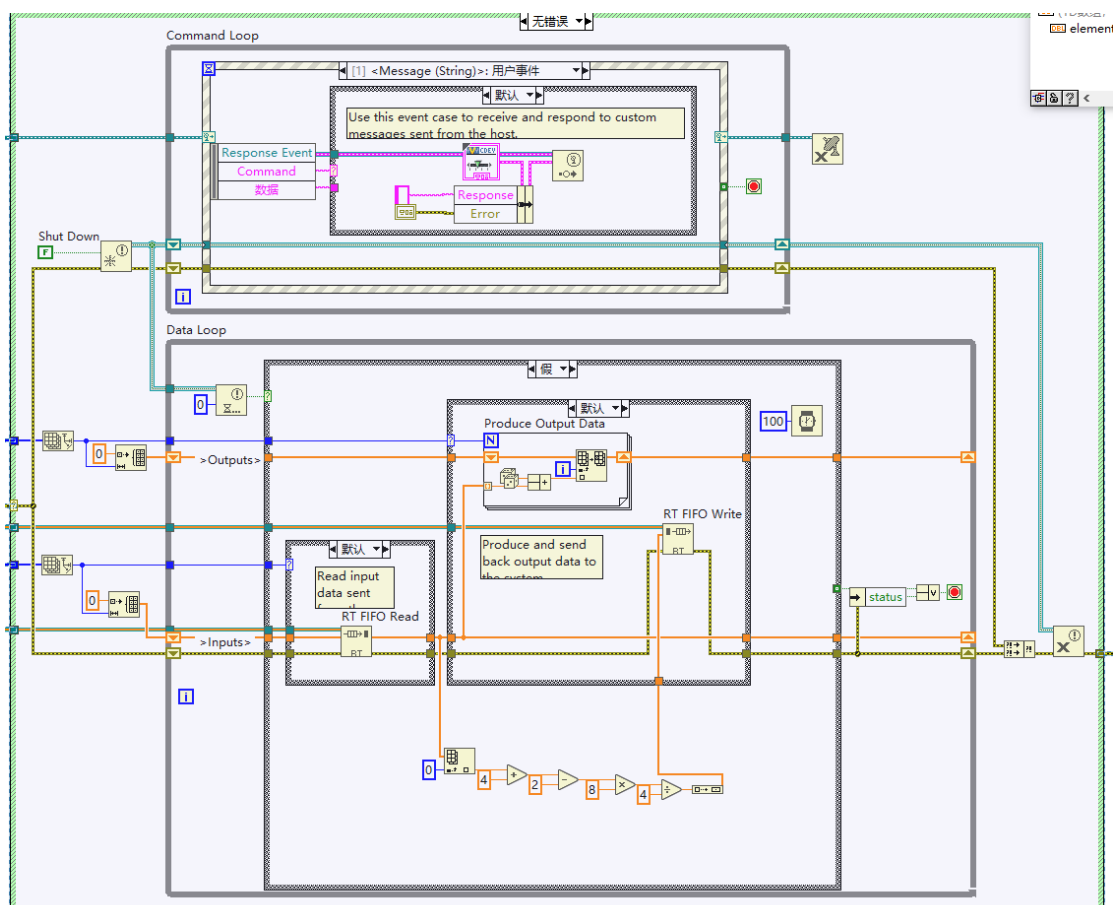
在顺序帧结构中所做的工作是读取默认的属性, 当属性在 Veristand 中的配置, 通过事件结构将新值赋给 Custom Devices 属性; 1 中所用 vi 是 Get Item property, 第二步是添加事件结构, 3 是 Set Item property



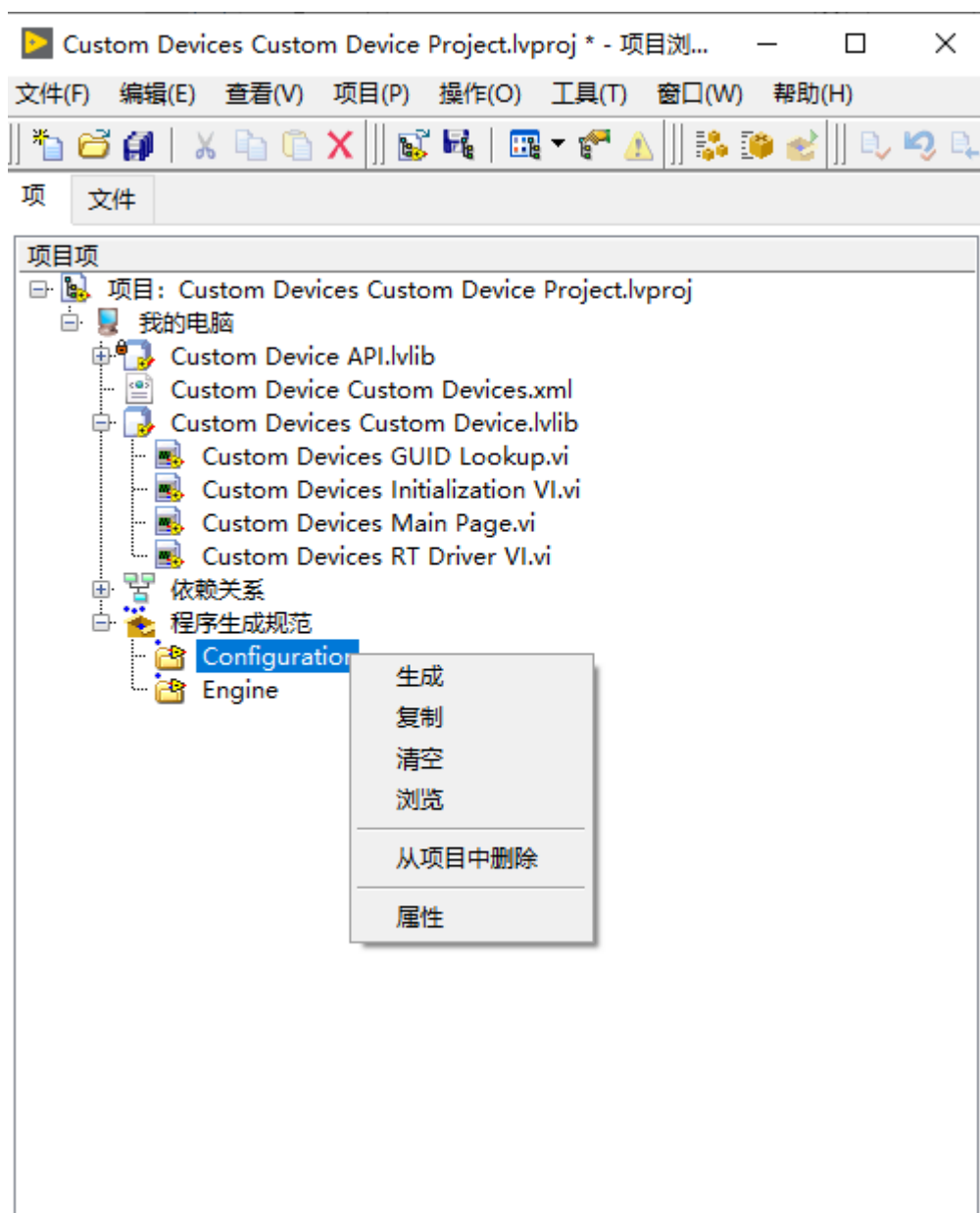
4. 打开 RT Driver VI.vi,添加代码读取 Main Page.vi 传入的属性值



5. 利用 RT FIFO Read 获取模型产生的数据，输入的数据为一个数组的形式，需要通过数组索引把单个数据提取出来，通过编写的代码运算后，输出，输出接收端是接收的是一维数组

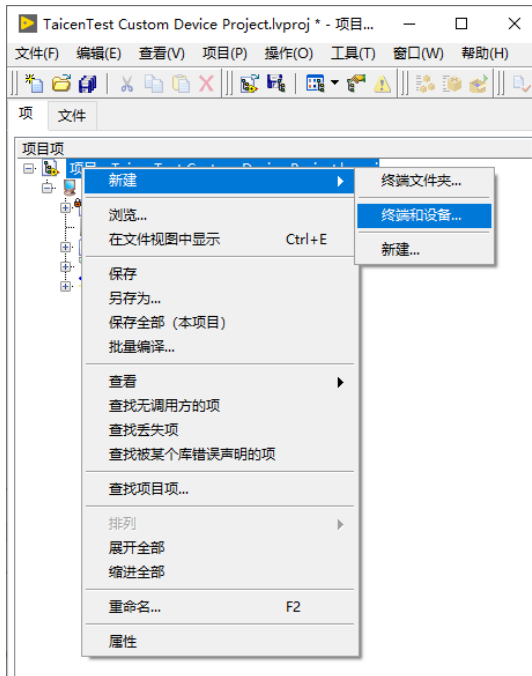


6. 点击程序生成规范，Configuration 和 Engine 分别生成一次，可以在 C:\Users\Public\Documents\National Instruments\NI VeriStand 2018\Custom Devices\项目名称，中找到生成的 Custom Devices 文件

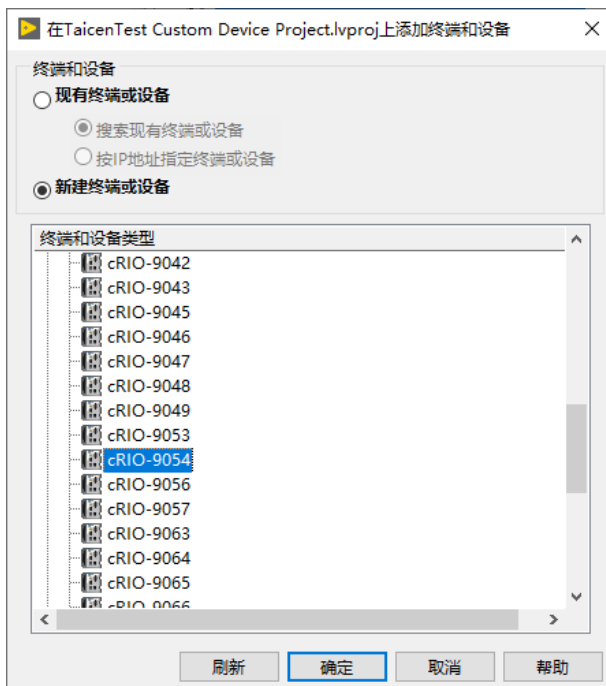


创建 Custom Devices 在 CompactRIO 平台运行文件

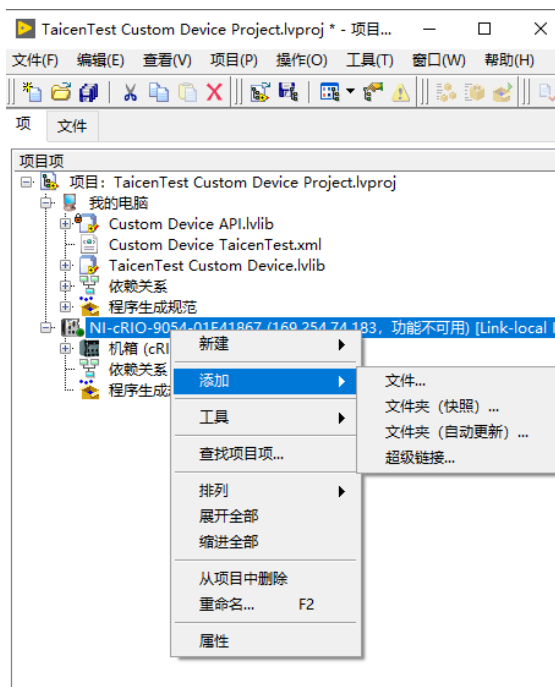
1. 点击 Custom Devices 项目右键-新建-设备和终端



2. 点击新建终端和设备-Real-Time CompactRIO-设备型号为 cRIO-9054-确定

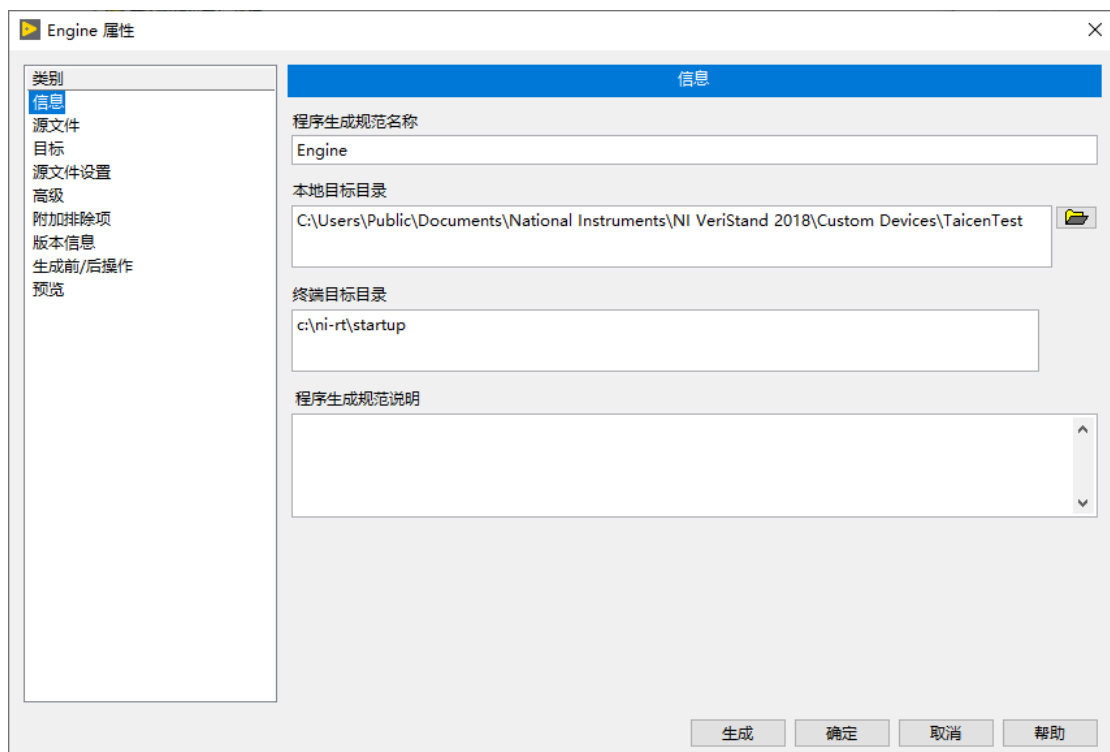


3. 点击添加的设备右键-添加-文件，将 RT.vi 添加至设备中

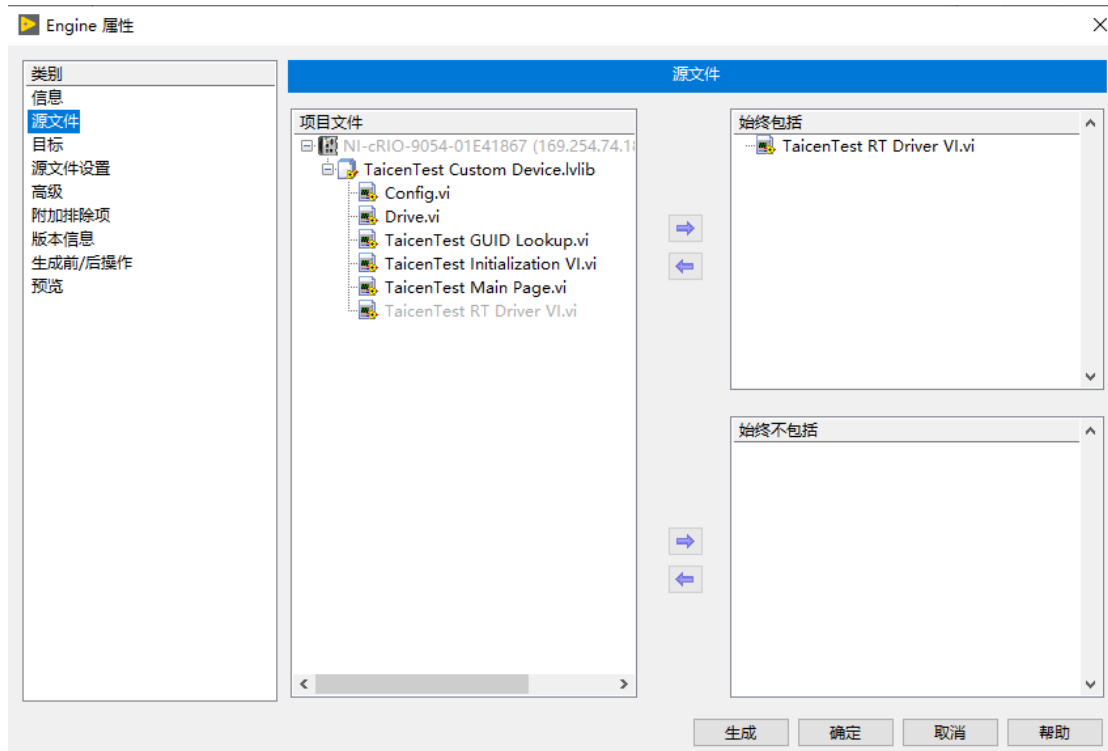


4. 点击程序生成规范，新建-源码发布，在信息一栏中
程序生成规范名称为：Engine

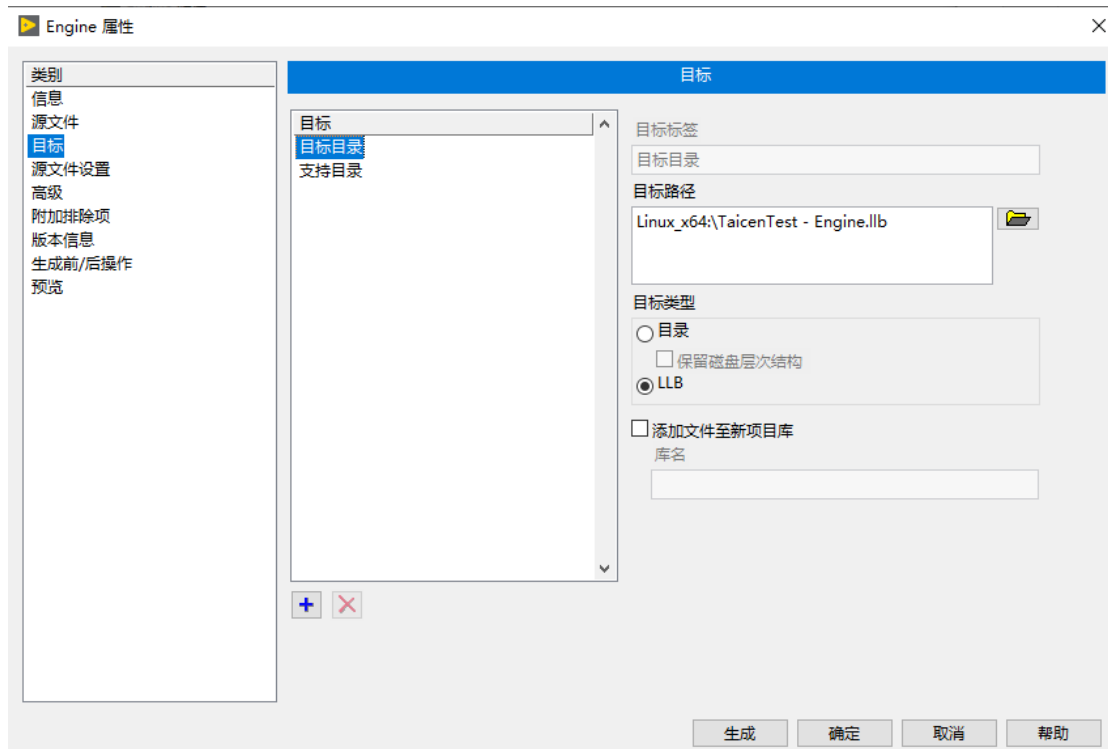
本地目标目录为：C:\Users\Public\Documents\National Instruments\NI VeriStand 2018\Custom Devices\项目名称



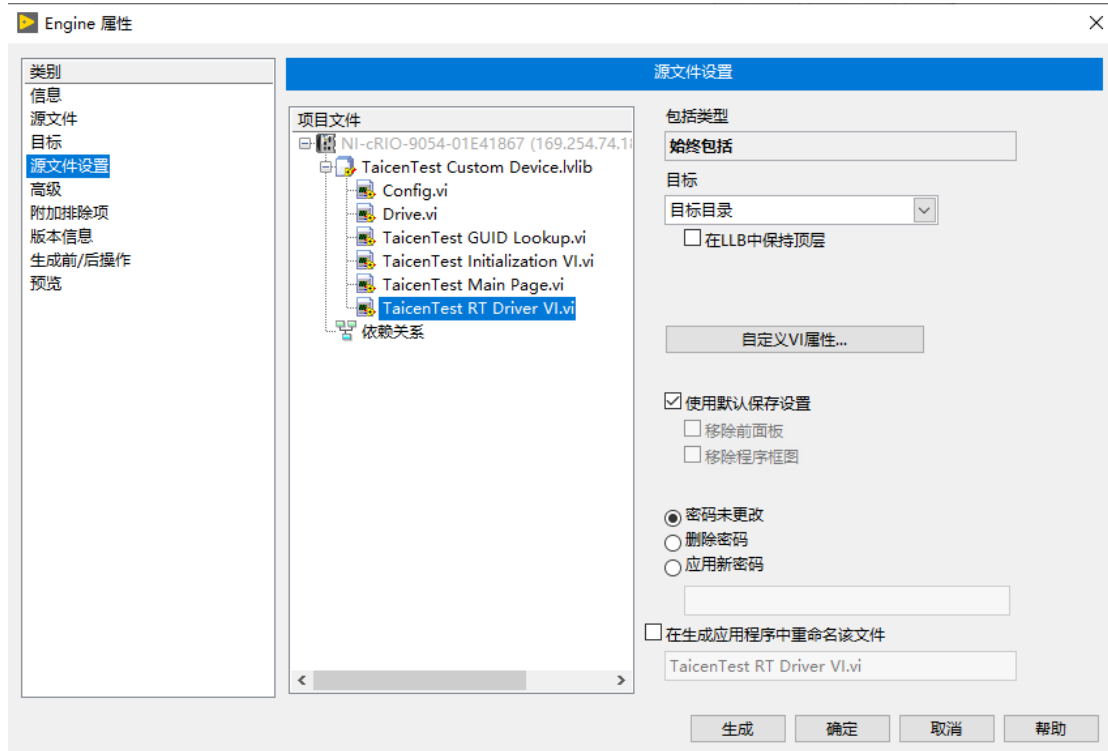
5. 在源文件一栏中，将 RT Driver.vi 添加到始终包括



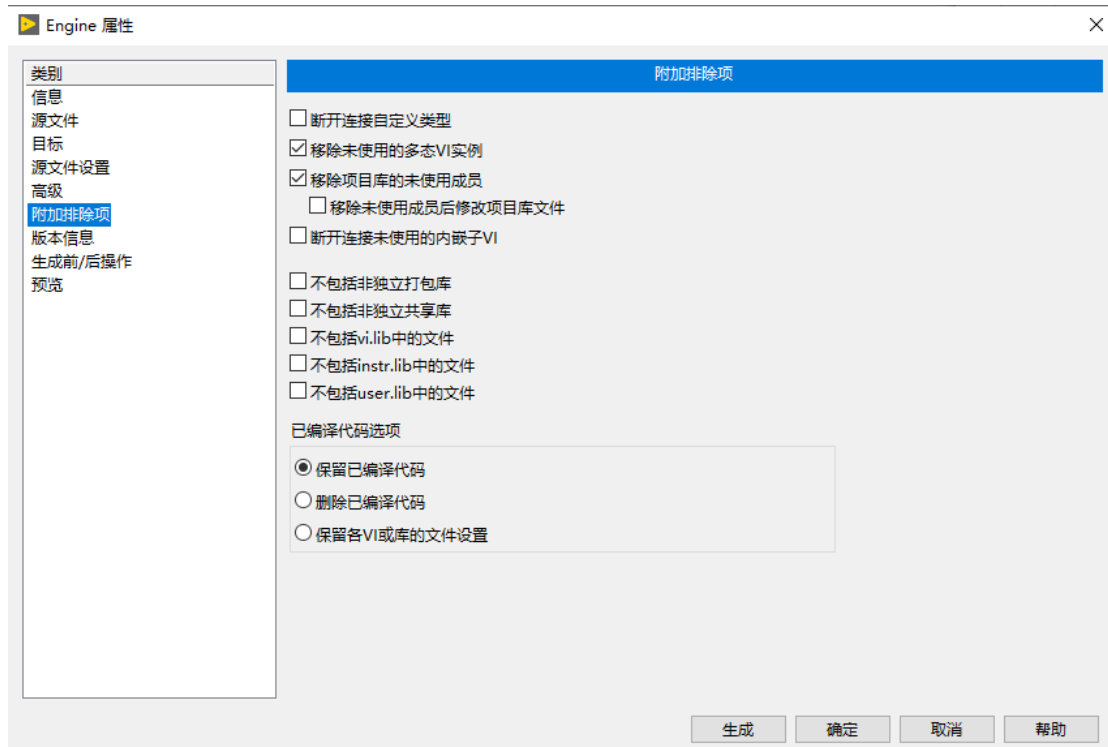
6. 在目标一栏中
目标路径: Linux_x64:\Custom Devices Engine.llb
支持目录: Linux_x64:\data



7. 在源文件设置一栏中，将 RT.vi 目标改为目标目录



8. 在附加排除项一栏中，勾选“移除未使用的多态 VI 实例”和“移除项目库的未使用成员”，点击确定



9. 修改 XML 配置文件，将 Linux_x64 调用路径添加至配置文件中

名称	修改日期	类型	大小
Lib	2020/1/8 17:52	文件夹	
Custom Device TaicenSW	2020/3/13 18:07	XML 文档	3 KB

```

- <CustomDeviceVI>
- <SourceDistribution>
- <Source>
  <SupportedTarget>PharlapWindows</SupportedTarget>
- <Source>
  <Type>To Common Doc Dir</Type>
  <Path>Custom Devices\TaicenSW\TaicenSW Engine.llb\TaicenSW RT Driver VI.vi</Path>
</Source>
  <RealTimeSystemDestination>c:\ni-rt\VeriStand\Custom Devices\TaicenSW\TaicenSW Engine.llb\TaicenSW RT Driver VI.vi</RealTimeSystemDestination>
</Source>
- <Source>
  <SupportedTarget>Linux_x64</SupportedTarget>
- <Source>
  <Type>To Common Doc Dir</Type>
  <Path>Custom Devices\TaicenSW\Linux_x64\TaicenSW - Engine.llb\TaicenSW RT Driver VI.vi</Path>
</Source>
  <RealTimeSystemDestination>c:\ni-rt\VeriStand\Custom Devices\TaicenSW\Linux_x64\TaicenSW - Engine.llb\TaicenSW RT Driver VI.vi</RealTimeSystemDestination>
</Source>
</SourceDistribution>
</CustomDeviceVI>
  
```

10. 进入 Labview 项目中，点击程序生成规范，重新生成文件

共享 查看

公共 > 公用文档 > National Instruments > NI VeriStand 2018 > Custom Devices > TaicenTest >

名称	修改日期	类型	大小
Data	2020/3/16 18:32	文件夹	
Linux_x64	2020/3/16 18:33	文件夹	
Custom Device TaicenSW	2020/3/15 10:34	XML 文档	2 KB
Engine_depinstr.bin	2020/3/16 18:33	BIN 文件	1 KB
TaicenTest Configu	2020/3/16 18:32	LabVIEW LLB	1,188 KB
TaicenTest Engine	2020/3/16 18:33	LabVIEW LLB	911 KB

创建日期: 2020/3/16 18:33
大小: 923 KB
文件夹: data
文件: TaicenTest - Engine

共享 查看

公共 > National Instruments > NI VeriStand 2018 > Custom Devices > TaicenTest > Linux_x64

名称	修改日期	类型	大小
data	2020/3/16 18:33	文件夹	
TaicenTest - Engine	2020/3/16 18:33	LabVIEW LLB	911 KB

搜索"Linux_x64"