

Labview 开发教程

1 软件版本

本例程采用 Labview2014 版本，其它版本操作方法类似。本教程实现了如何通过 Labview 来控制运动控制器，具体运动控制器和其它部件的配合，请灵活运用 Labview 的相关知识。

2 导入共享库

2.1 导入共享库的简介

运动控制器提供标准的共享库（dll）控制方式，Labview 在使用前，需要进行共享库的导入，导入会将每个函数导入成一个 VI，在后续的开发中使用。在导入之前，请获取最新的开发包。只需导入一次，如已经导入，请跳到下一节。

2.2 共享库的调用约定

控制器提供两种版本的 dll，标准调用（stdcall）和 c 调用（cdecl），请根据需要选择，推荐是用标准调用。以下以标准调用为例进行演示，c 调用原理一样。

2.3 导入前的检查

在进行导入之前，请检查一下提供的 h 文件（一般情况下都是修改好的，如不一致，请修改成以下示例情况）

```

29 #define RESOURCE_CIRCLE 0x00000100
30 #define RESOURCE_PLC 0x00000200
31 #define RESOURCE_STREAM 0x00000400
32 #define RESOURCE_ENCODER 0x00000800
33 #define INT32 int
34 MT_API INT32 __stdcall MT_Init(void);
35
36 MT_API INT32 __stdcall MT_DeInit(void);
37
38 MT_API INT32 __stdcall MT_Get_Dll_Version(char** sVer);
39
40 MT_API INT32 __stdcall MT_Open_UART(char* sCOM);
41
42 MT_API INT32 __stdcall MT_Close_UART(void);
43

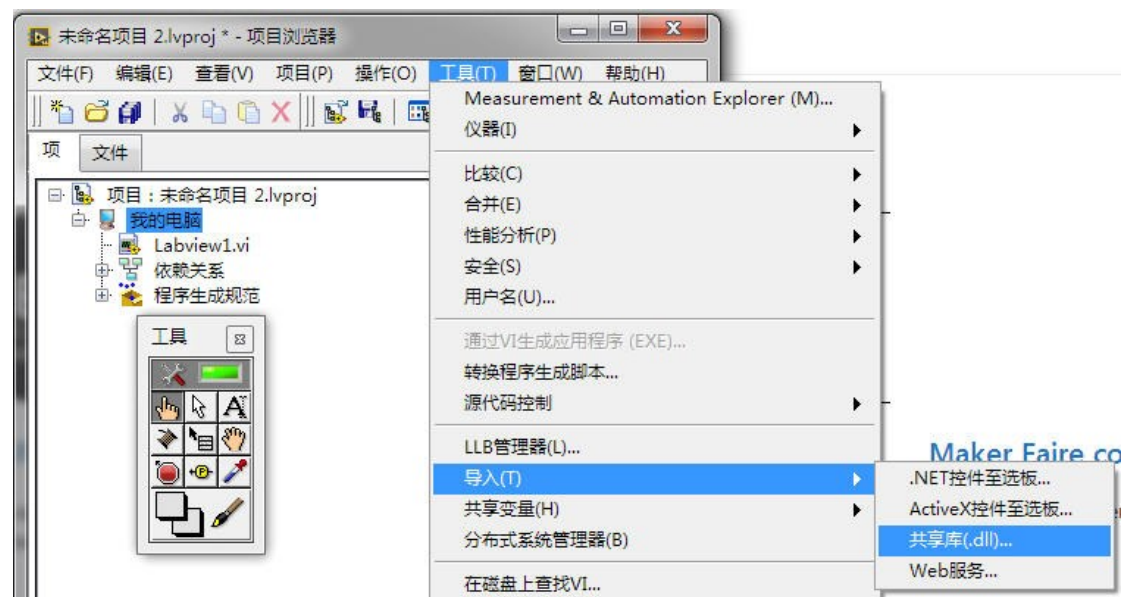
```

Labview 不能识别 INT32 的类型，直接替换 INT32 或者增加如图的定义#define INT32 int

h 文件要和对应的 dll 配套

2.4 导入过程

2.4.1 打开导入向导



2.4.2 导入类型

第一次导入请选择为共享库创建 VI，如果 dll 版本升级第二次更新的话请选择更新，更新可以保留以前的修改。

导入共享库

NATIONAL
INSTRUMENTS

指定创建或更新模式

☐ 为共享库创建VI

基于用户提供的头文件和共享库文件创建VI。

☒ 为共享库更新VI

更新之前导入的VI以用于以下项目库

项目	DLL文件	日期
MT_API.lvlib	MT_API.dll	20:33:53 03/19/2015
MT_API.lvlib	MT_API.dll	13:33:46 02/12/2015

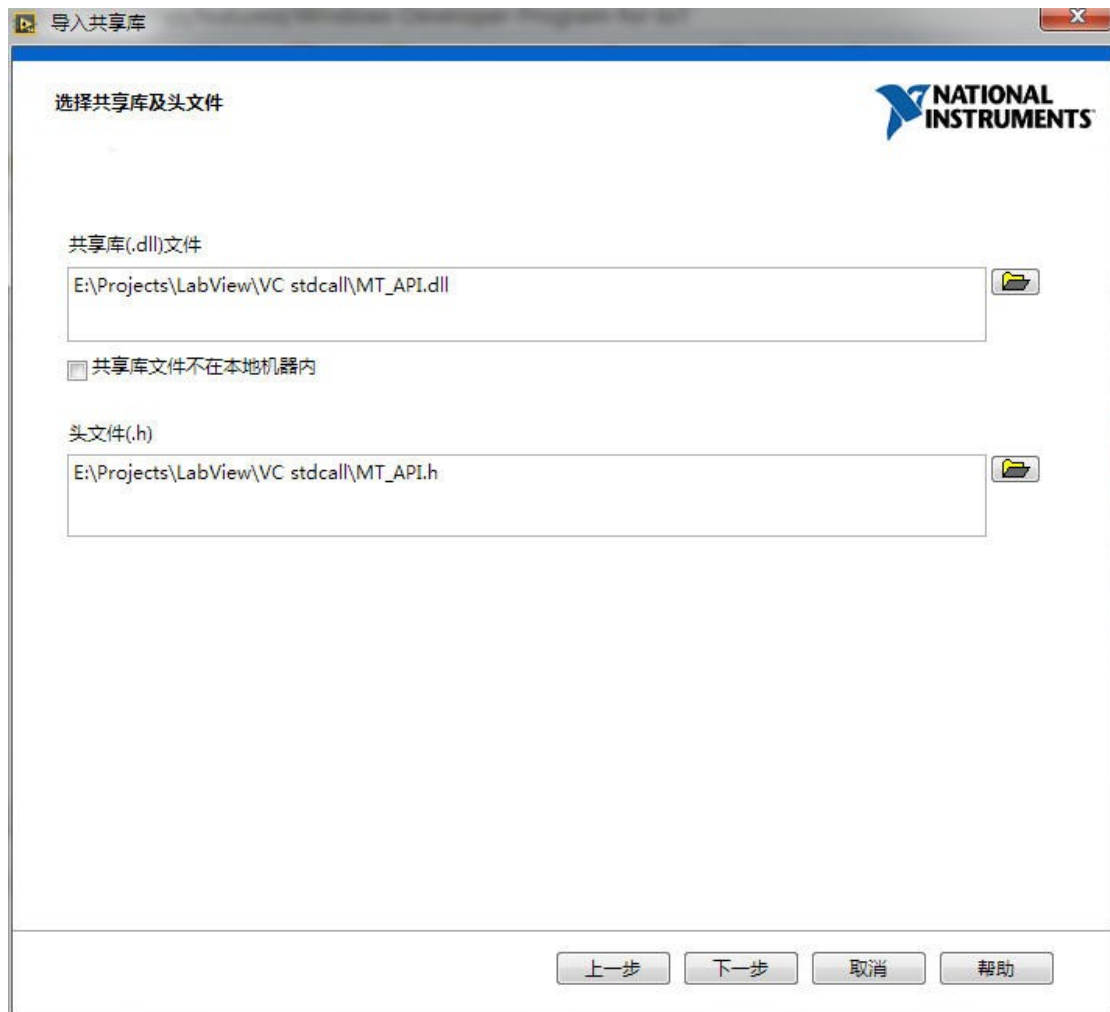
上一步

下一步

取消

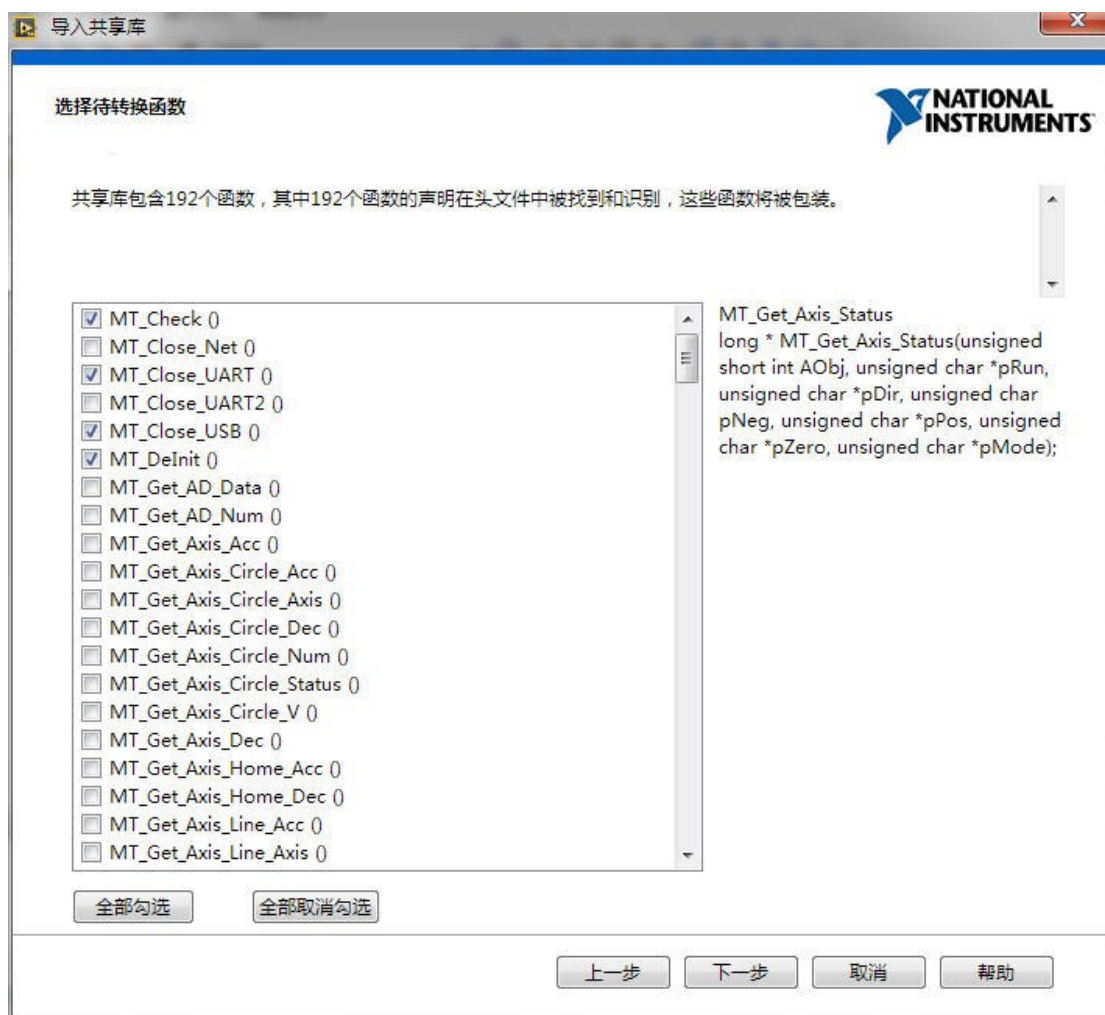
帮助

2.4.3 选择导入文件

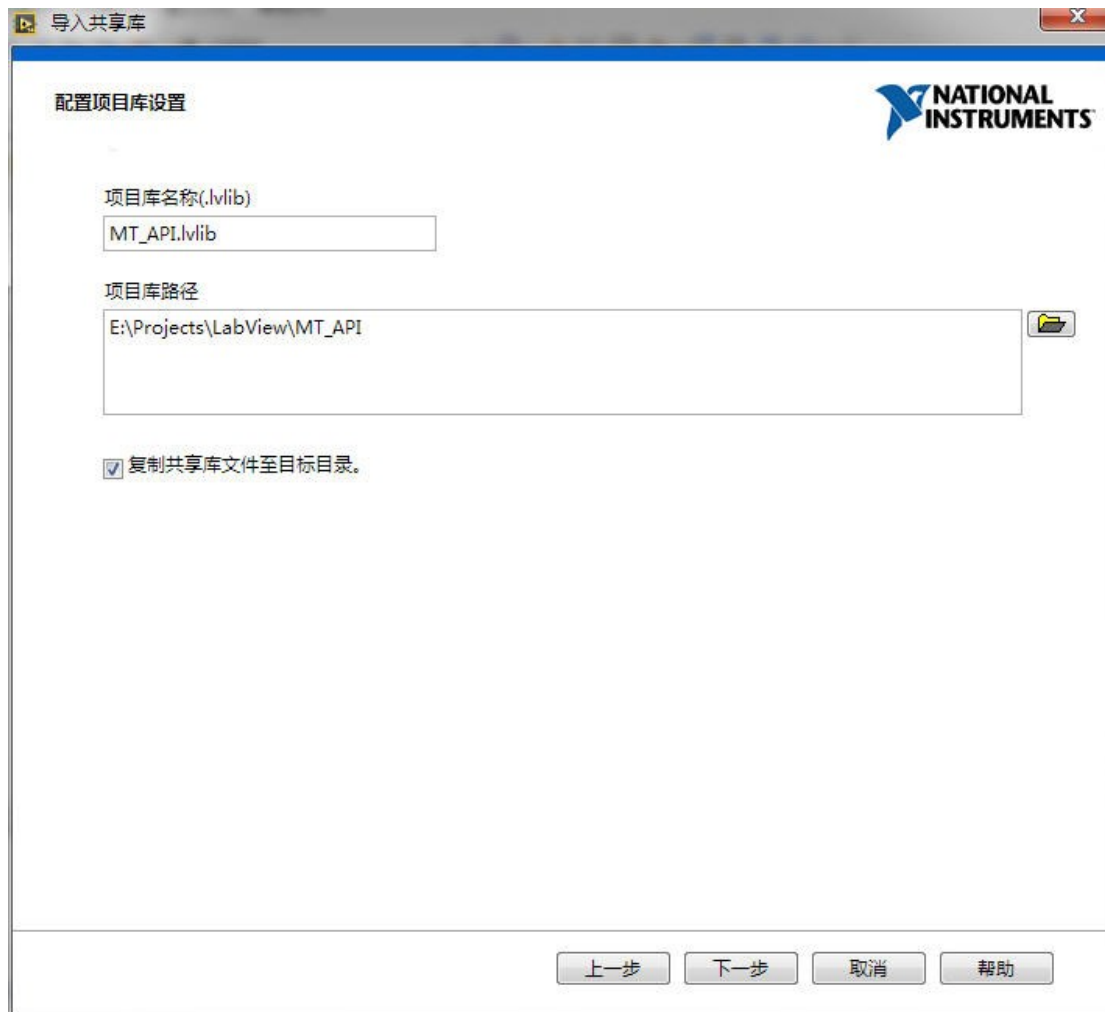


2.4.4 选择需要的函数

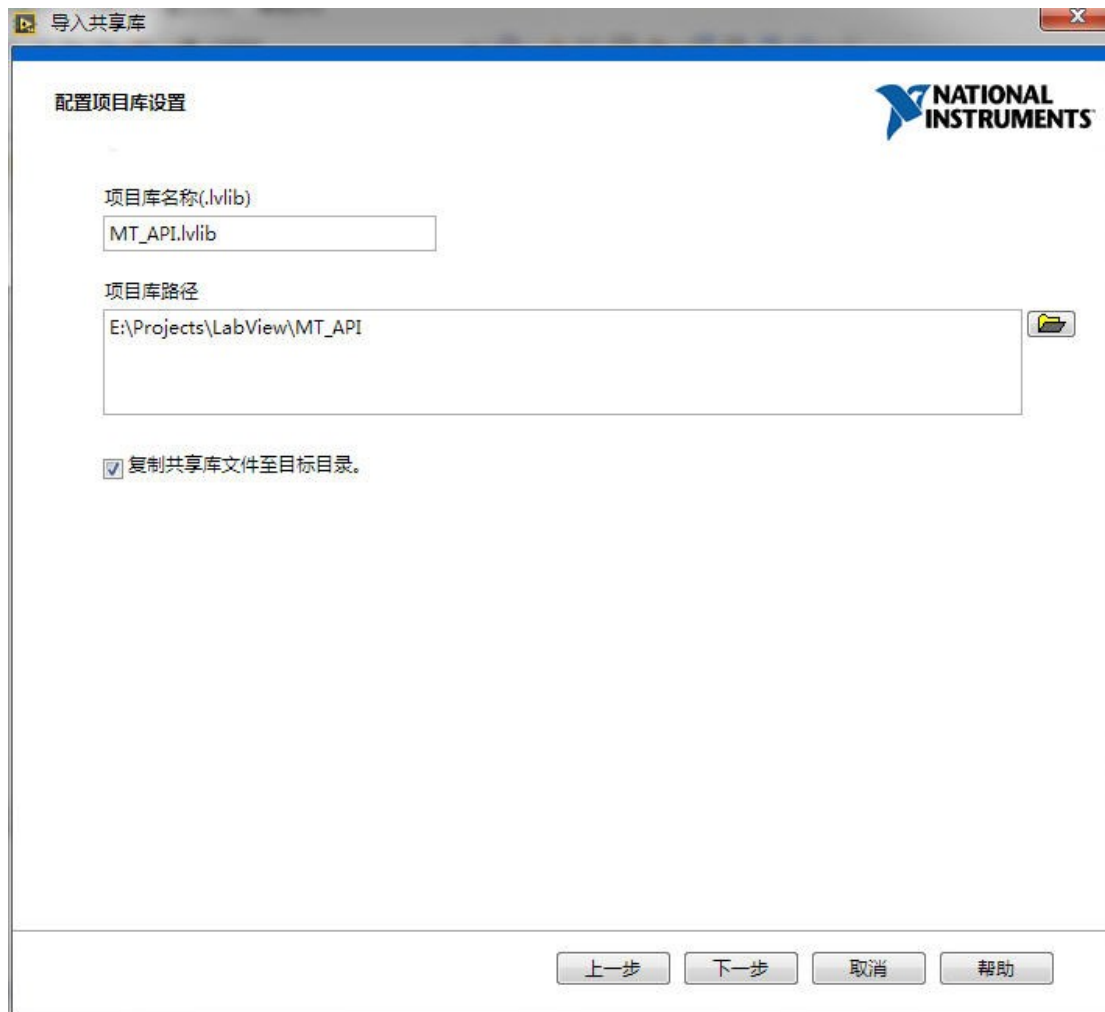
请勾选使用到的函数，函数的功能请参考 SDK 编程手册（MT_SDK_Manual），其中 MT_Init MT_DeInit 必须选择，通信端口函数根据需要选择（串口、USB、网络），工作模式根据需要选择（零位、定速、定位、联动插补、圆弧插补等）也可以都导入。建议根据需要选择，Labview 不能完全自动识别所有函数，有一定的人工修改工作。



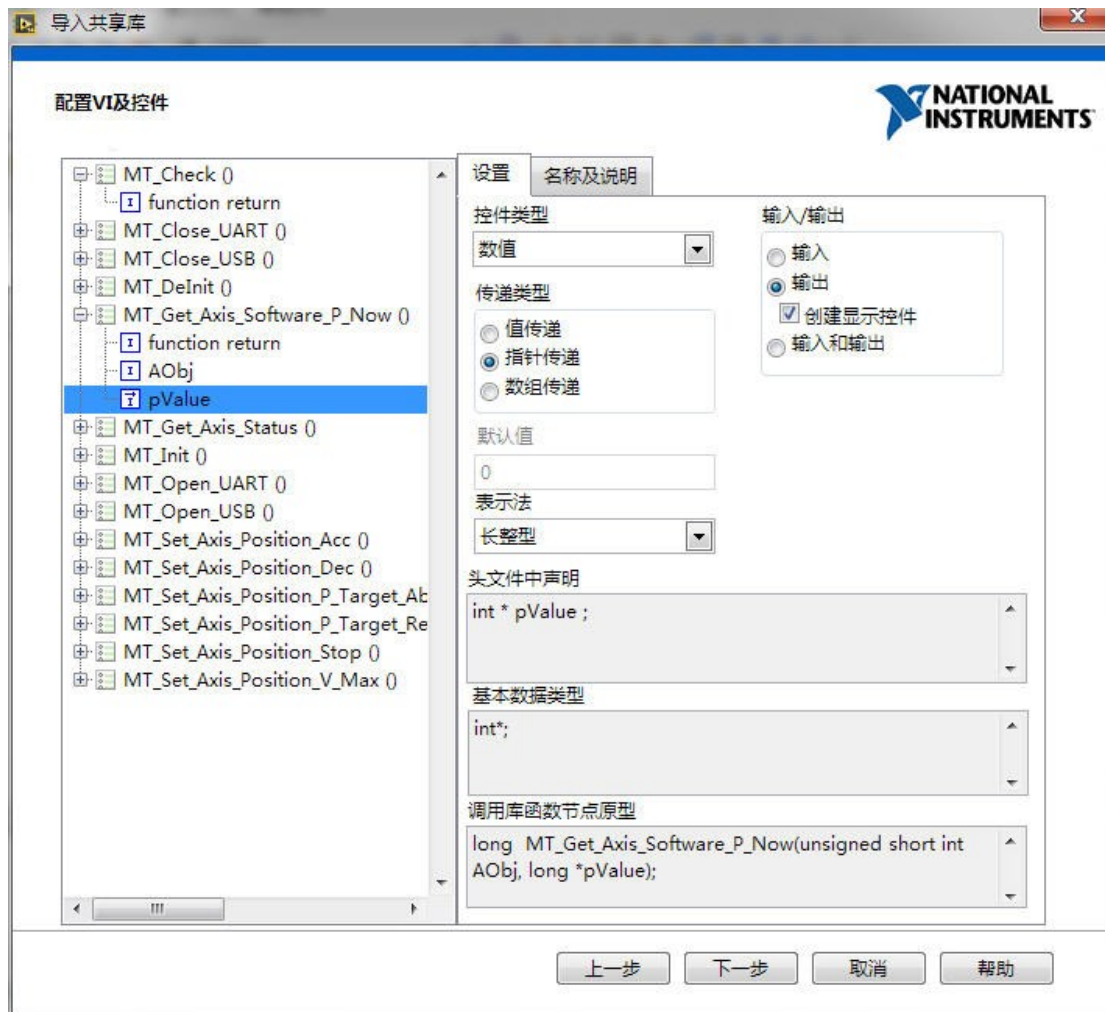
2.4.5 选择库保持的路径

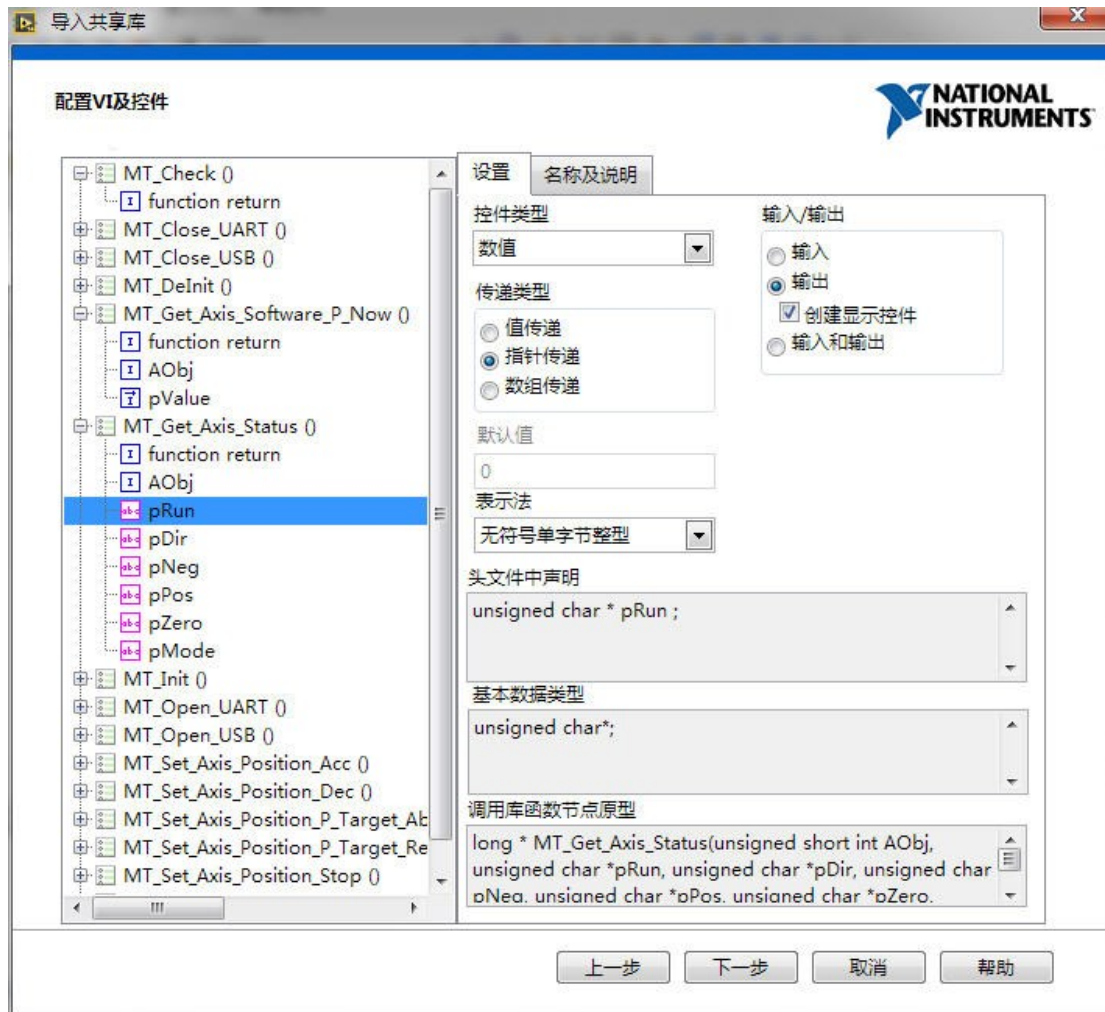


2.4.6 选择错误处理模式



2.4.7 配置 VI 及控件（关键步骤）

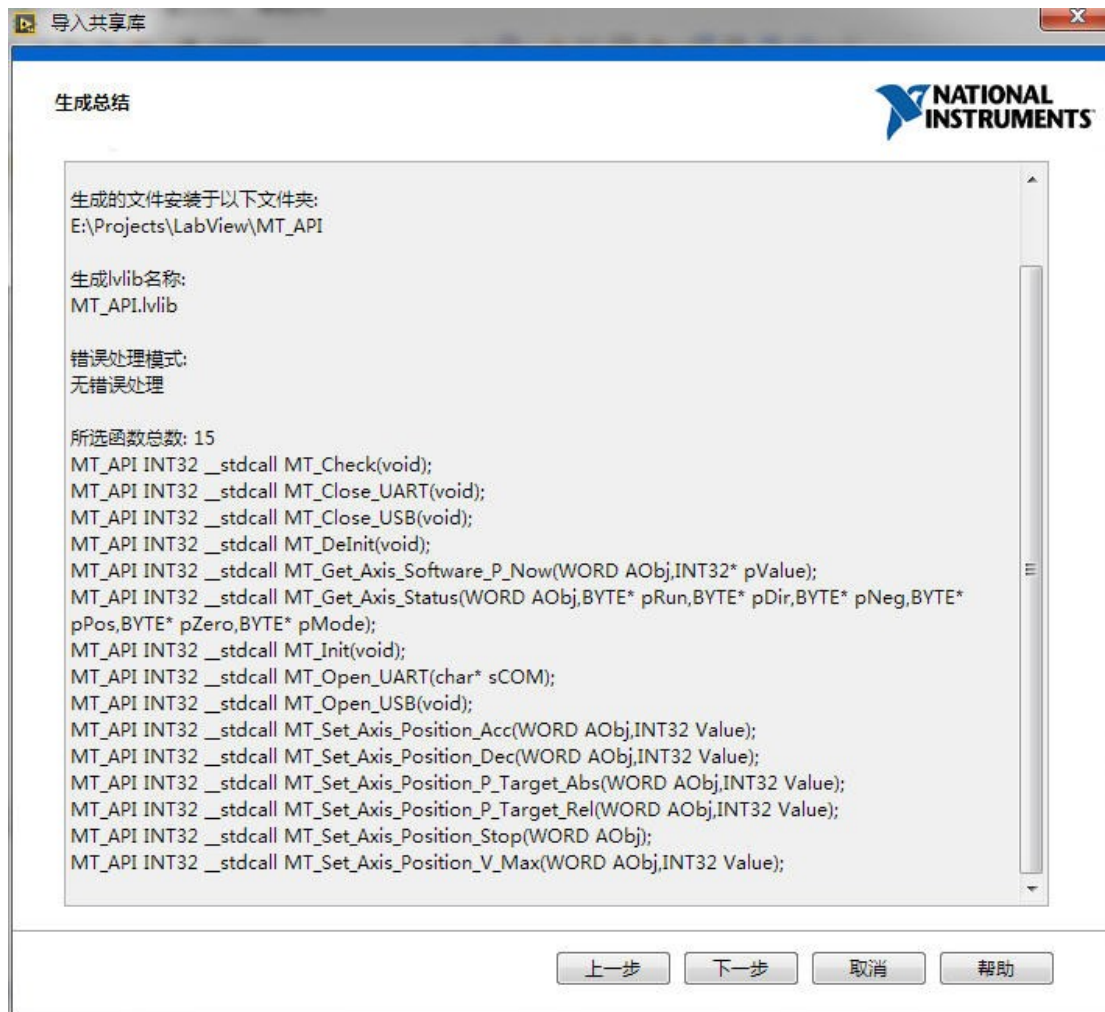




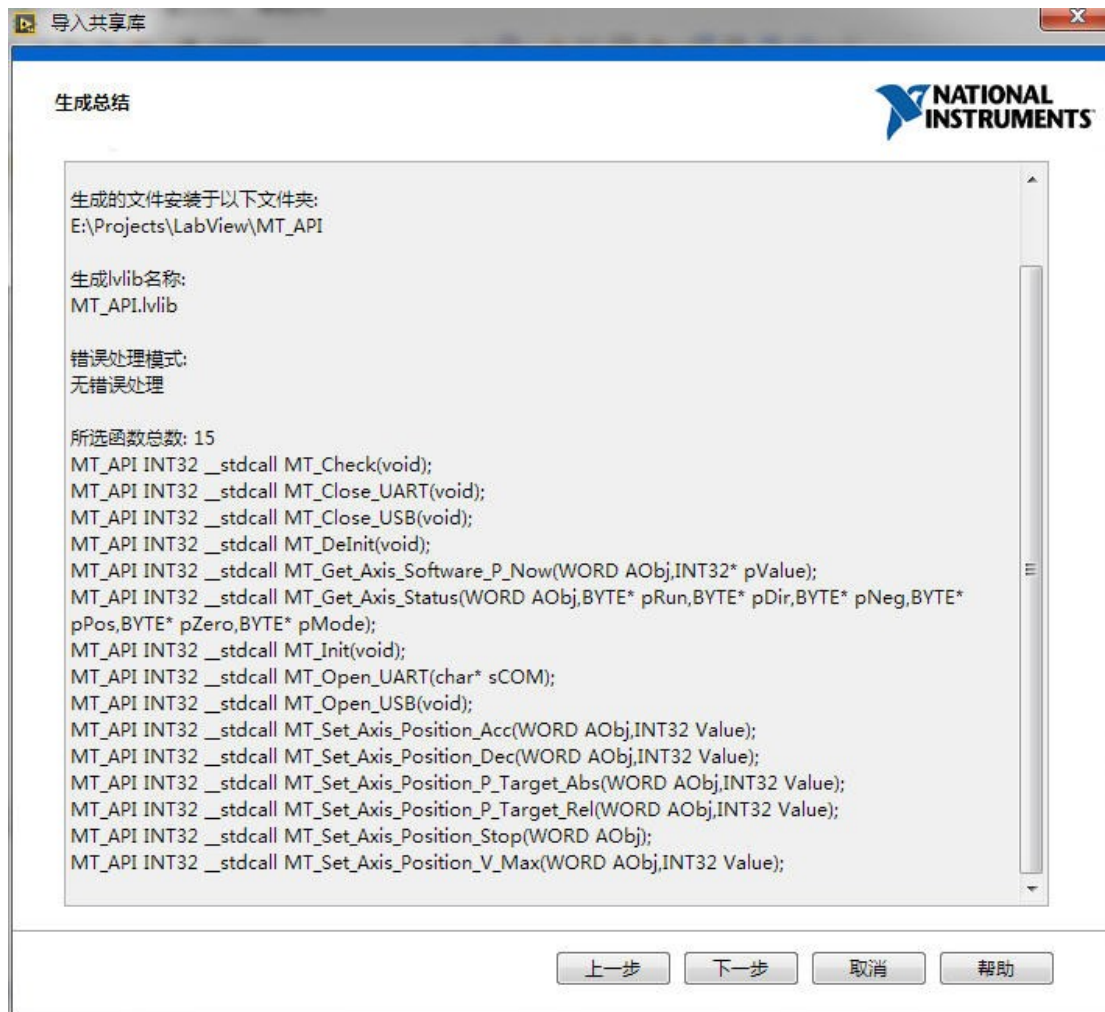
对于 MT_Get 开头的函数中的指针数据，Labview 不能识别，输入输出也不能识别，需要使用的时候进行修改。

如果用联动插补的 MT_Set_Axis_Line_X_Run_Abs 和 MT_Set_Axis_Line_X_Run_Rel，其中的数组指针参数，传递类型请选择数组指针。在更新库类型下，这些设置是可以自动记录保留的。不一致时请查看 SDK 编程手册（MT_SDK_Manual）。

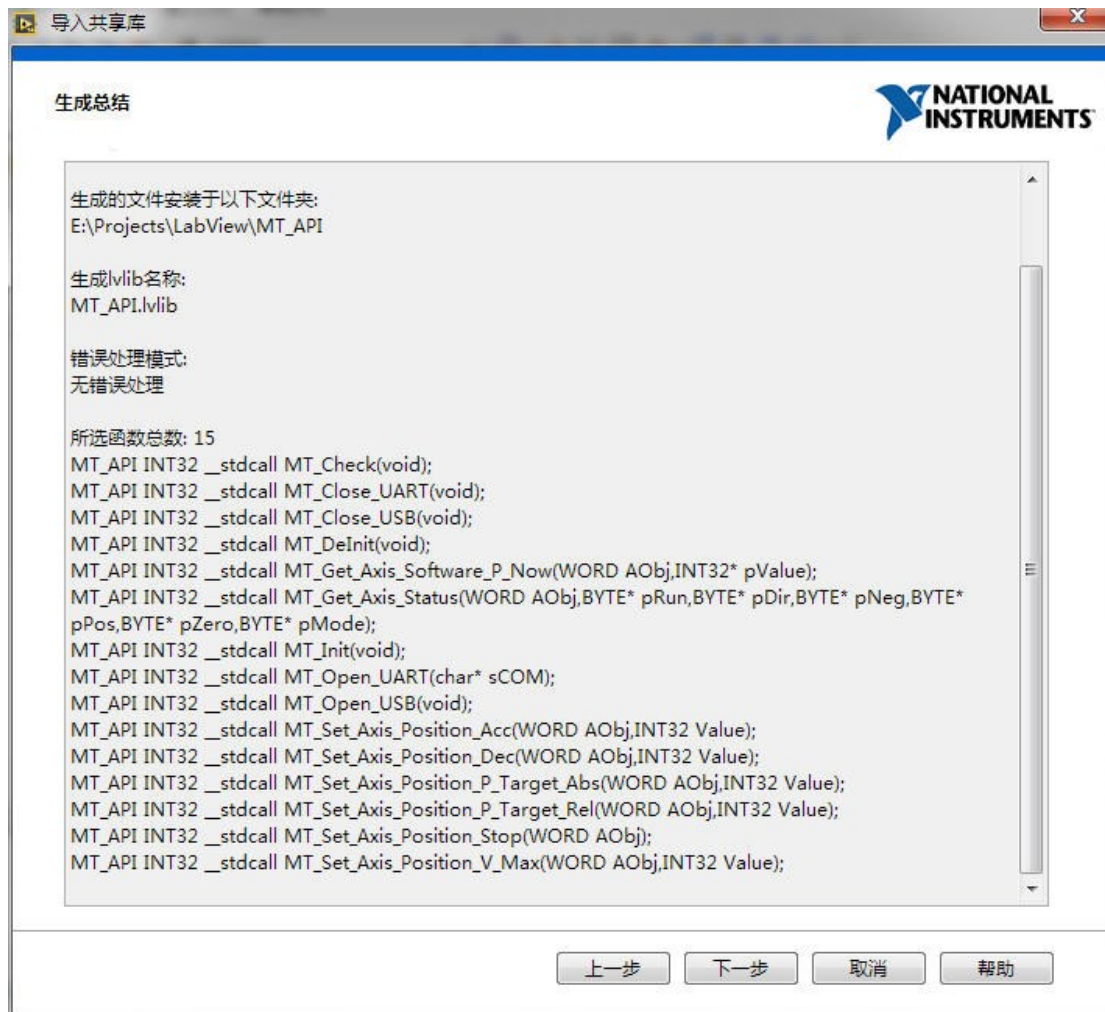
2.4.8 导入总结



2.4.9 生成进度



2. 4. 10 完成

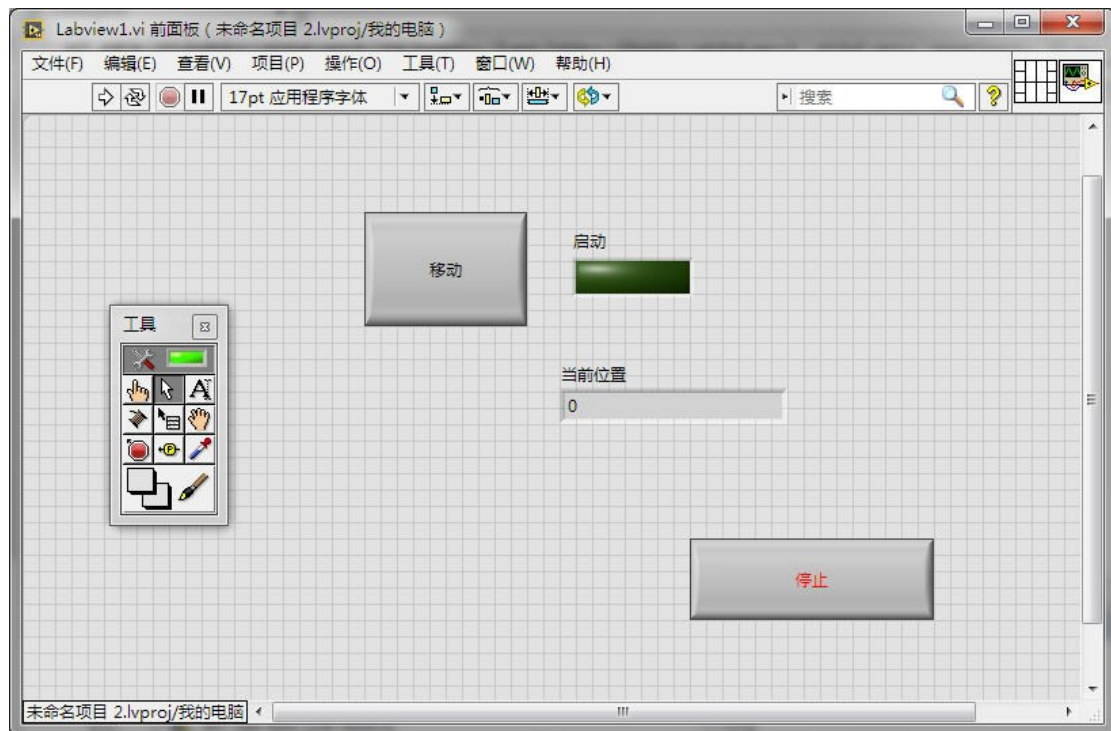


3 例程演示

3. 1 例程的功能

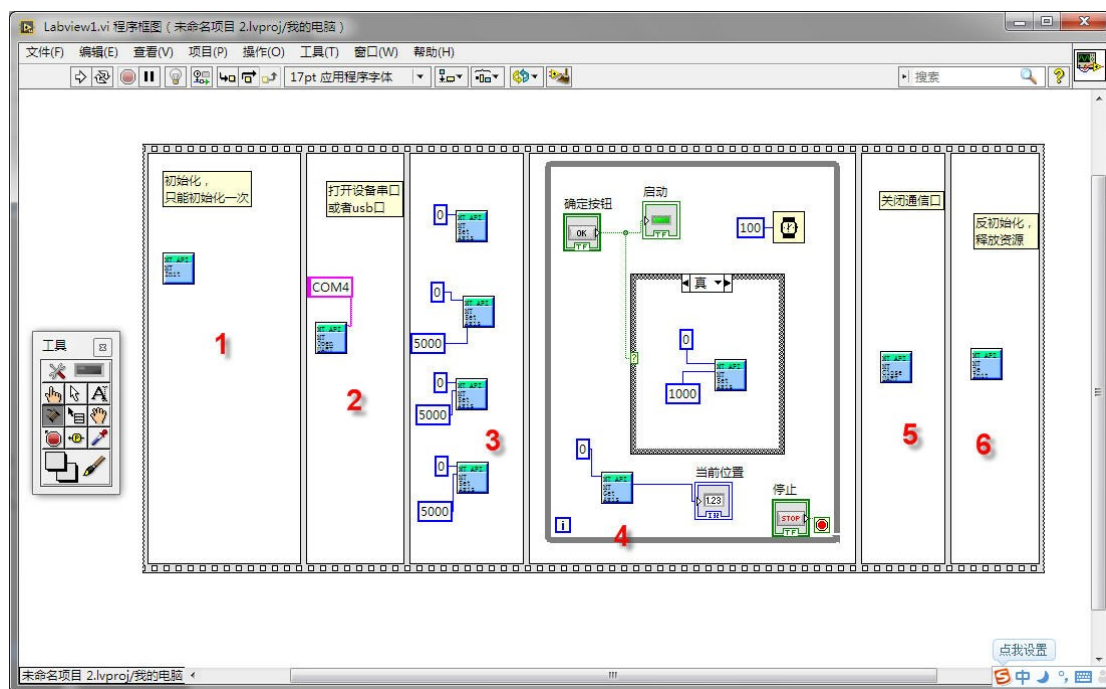
例程完成对第一轴的定位控制，在指令发出的时候，指示灯会亮一下，同时实时显示当前的位置值，其中所有的设置单位都是基于脉冲数，请参考实际情况转换计算。

3.2 例程的前面板



点击移动按钮时，第一轴会移动一个距离（1000 个脉冲，请根据实际情况修改）
启动的指示灯会亮一下，当前位置为显示当前的位置；停止按钮停止程序的运行

3.3 例程的程序框图



程序采用顺序模式，共分为 6 个步骤

步骤 1：初始化 MT_Init，进行控制器的资源申请，只能且必须执行一次

步骤 2：打开设备通信口，请注意修改为对应的通信口（串口号，USB，网络 IP 地址等）本例采用串口通信，串口号为 COM4

步骤 3：需对控制的轴的运动参数进行修改，包括工作模式，加速度，减速度，最大速度等。本例为定位模式。

步骤 4：关键的步骤，用来实现用户的需求。在本例中，做了一个循环结构（带延时），在循环结构中，做了一个判断结构和位置查询。判断结构判断前面板的按钮是否按下，当按下一次则发出一次移动的指令；查询指令用来查询当前的位置，显示在前面板中；当循环检测到停止时，停止循环，执行后续的步骤。

步骤 5：关闭使用的通信口

步骤 6：释放资源。

3.4 注意事项

注意事项 1：在不同的工作模式切换时（零位，定速，定位等），需要发送一次工作模式切换指令

注意事项 2：读取函数都是用指针送数出来，如不能正常工作，请查看前面导入的函数是否有问题

注意事项 3：进行所有的动作前，必须进行资源申请和打开通信口，如果需要，

在导入时选错误处理模式，或者判断返回值，如果返回值不为 0，则发生了错误。
注意事项 4：在循环中注意要延时，避免前面板卡死。