
上海格西信息科技有限公司

使用 **NI-DAQmx** 进行数据采集

版本 **0.1**

目录

1. 概述	3
2. 创建项目	4
3. 运行项目	8
3.1 打开项目	8
3.2 运行采集用例	8

1. 概述

NI-DAQmx 是 National Instruments 现阶段的数据采集驱动，NI-DAQmx 集成了全新的驱动架构和 API，用于控制 National Instruments DAQ 设备。

本项目演示如何通过 NI-DAQmx 提供的 API 来控制 National Instruments DAQ 设备，实现数据采集任务。

本例子文件位于：<软件安装目录>\Examples\Solutions\DAQ。

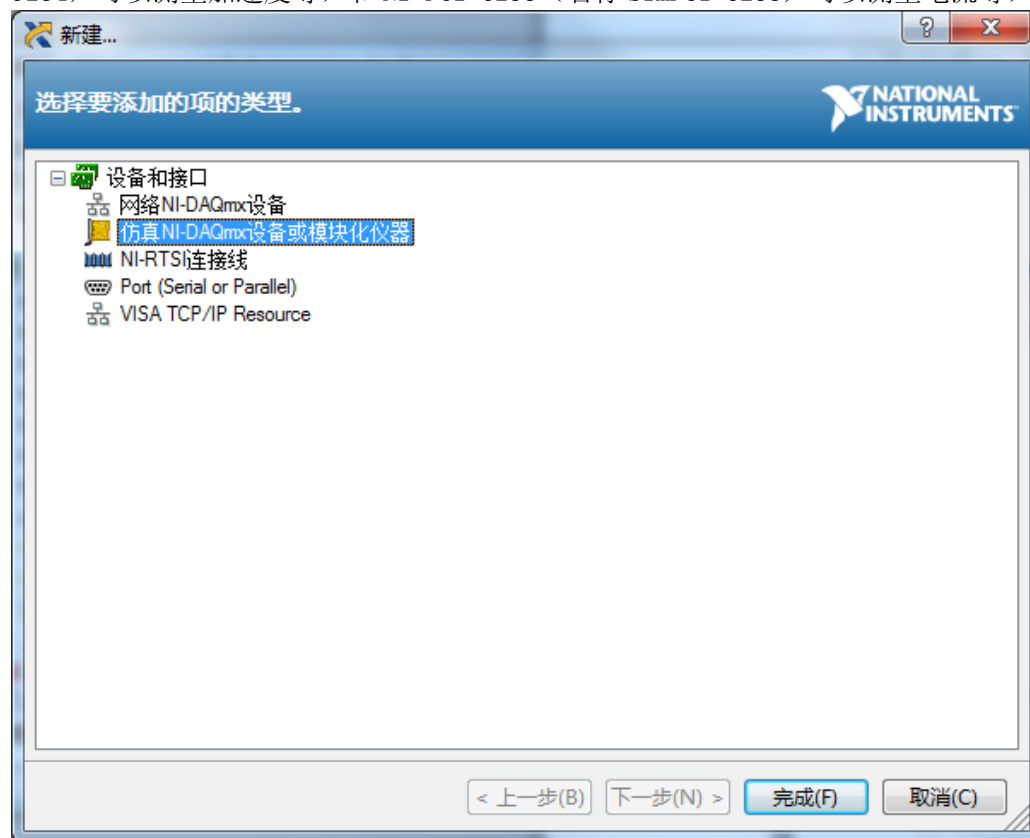
文件说明：

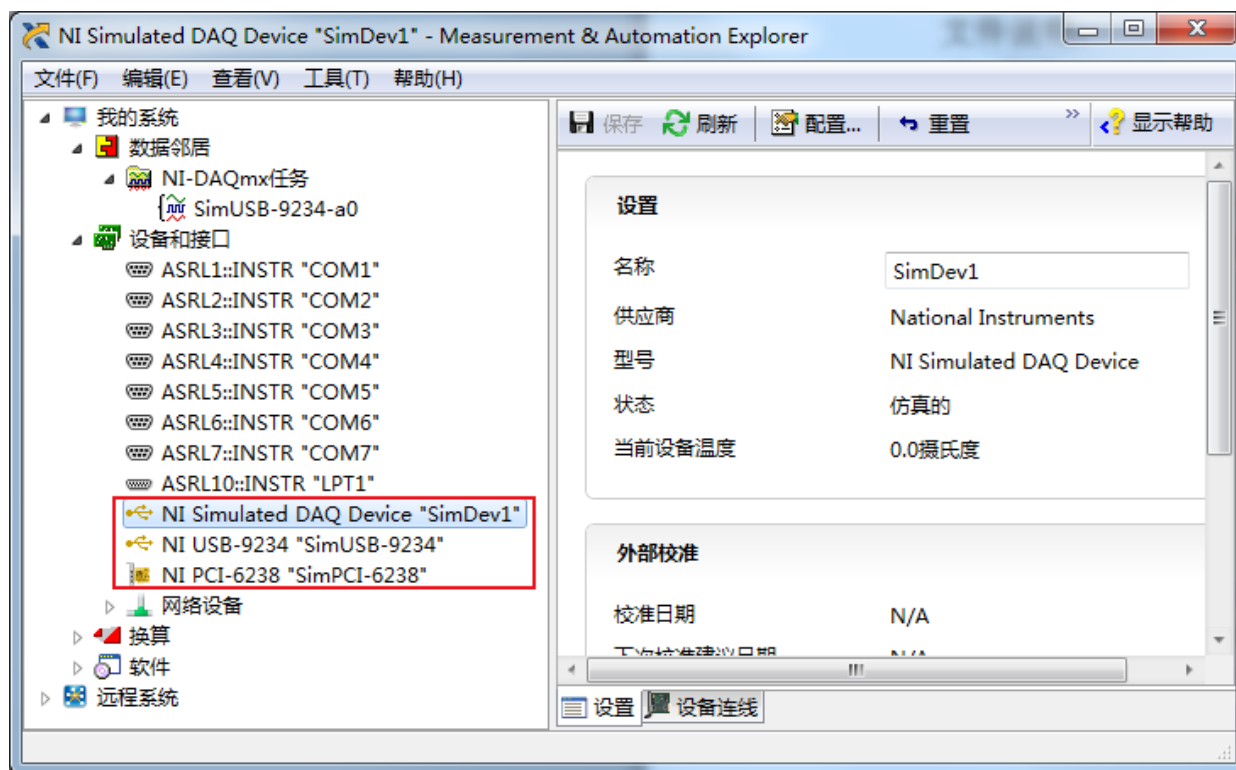
✓ DAQmx.gpj - 演示使用 NI 的 DAQmx 驱动库来进行数据采集 - 中文

运行例子前提条件：

1) 从 National Instruments 官网 www.ni.com 下载 NI-DAQmx 20.0 版本软件并安装；安装完毕 NI-DAQmx 软件驱动包，在目录 C:\Users\Public\Documents\National Instruments\NI-DAQ\Examples\DotNET4.5 下，存放 NI-DAQmx 的常用例子，本例子项目的脚本基本上是参考这些例子编写的。

2) 运行 NI MAX 设备管理软件，通过设备和接口节点，新建仿真设备 NI USB9234（名称 SimUSB-9234，可以测量加速度等）和 NI PCI-6238（名称 SimPCI-6238，可以测量电流等）。



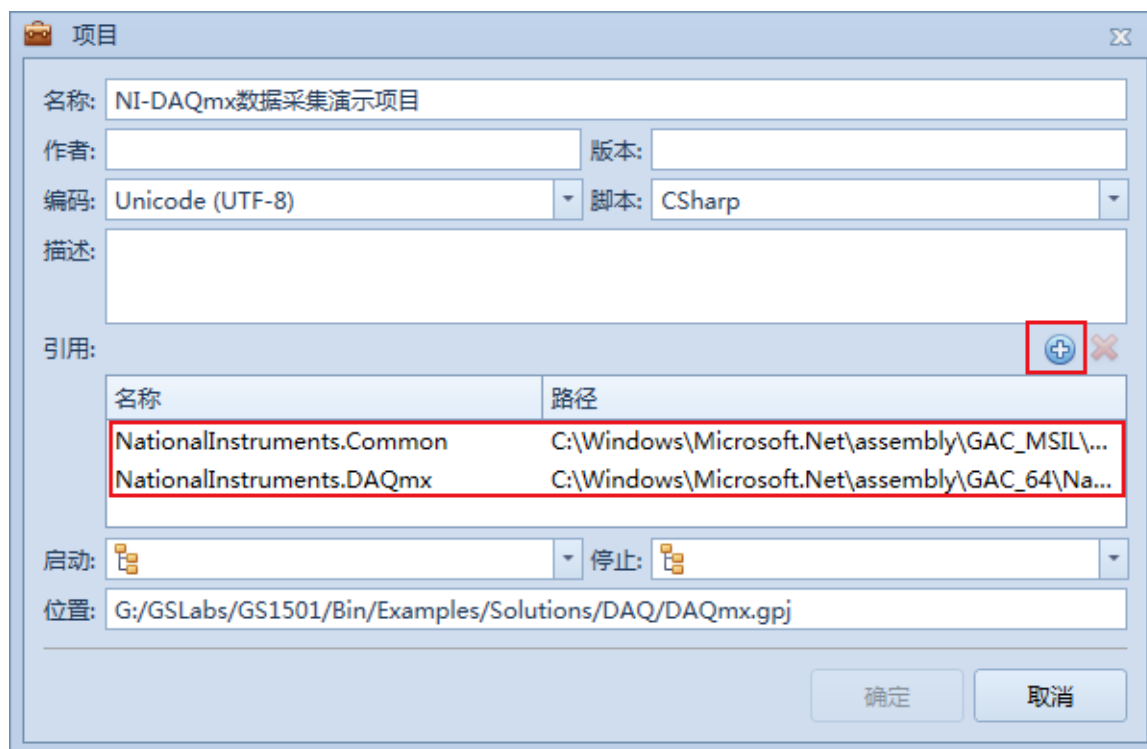


2. 创建项目

NI-DAQmx 对于 Microsoft .NetFramework 平台，提供了 NationalInstruments.Common.dll 和 NationalInstruments.DAQmx.dll 两个动态链接库，作为第三方使用 NI-DAQmx 控制 National Instruments DAQ 设备的应用程序接口。

本例子所带的 NationalInstruments.DAQmx.dll 是 64 位版本，如果使用的是 32 位操作系统，则需要在 C:\Program Files (x86)\National Instruments\MeasurementStudioVS2012\DotNET\Assemblies\Current 目录下把 32 位版本的 NationalInstruments.DAQmx.dll 拷贝到<软件安装目录>\Examples\Solutions\DAQ 目录下方可进行测试。

格西测控大师通过项目属性来引用第三方托管 DLL，如下图所示。



本项目的采集步骤采用 Value 型步骤，用脚本实现。下面是采集步骤 AnalogIn\MeasureVoltage\AcqMultVoltageSamples 的脚本。

```
using System;
using Genesis;
using Genesis.Scripting;
using Genesis.Sequence;
using Genesis.Workbench;
using NationalInstruments; // 使用 Command 库的命名空间
using NationalInstruments.DAQmx; // 使用 DAQmx 库的命名空间

public class Step_F427B2EB5AF54498884357A099AFEE39
{
    public ScriptContext Context { get; set; }

    //
    public Int32 BeginExecute(IStepContext context, IStep step)
    {
        try
        {
            string[] paramNames = new string[] { "物理通道", "最大值(V)", "最小值(V)", "采样数" };
            string[] paramDefaults = new string[] { "SimDev1/ai0", "10.0", "-10.0", "1000" };
            string[] parameterOptions = new string[] {}
            {DaqSystem.Local.GetPhysicalChannels(PhysicalChannelTypes.AI,
```

```
PhysicalChannelAccess.External), null, null, null});

    string[] paramResults = this.Context.ShowParameterDialog("DAQmx 参数设置", paramNames, paramDefaults, parameterOptions);
    if (paramResults == null)
    {
        return 0;
    }
    // Create a new task
    using(Task myTask= new Task())
    {
        // Create a virtual channel
        myTask.AIChannels.CreateVoltageChannel(paramResults[0], "",
            (AITerminalConfiguration)(-1), Convert.ToDouble(paramResults[2]),
            Convert.ToDouble(paramResults[1]),
            AIVoltageUnits.Volts);

        // Verify the Task
        myTask.Control(TaskAction.Verify);

        AnalogMultiChannelReader myAnalogReader = new
        AnalogMultiChannelReader(myTask.Stream);

        int totalSamples = Convert.ToInt32(paramResults[3]);
        for(int i=0; i<totalSamples; )
        {
            //Read the data from the channels
            double [] data = myAnalogReader.ReadSingleSample();
            if (data.Length > 0)
            {
                this.Context.Variants["AnalogIn/MeasureVoltage/Voltage"].Value = data[0];
                i++;
            }
        }
    }
    catch(DaqException exception)
    {
        // Display Errors
        this.Context.ShowMessageBox("DAQmx", exception.Message, System.Windows.MessageBoxButtons.OK, System.Windows.MessageBoxImage.Error);
    }
    return 0;
}

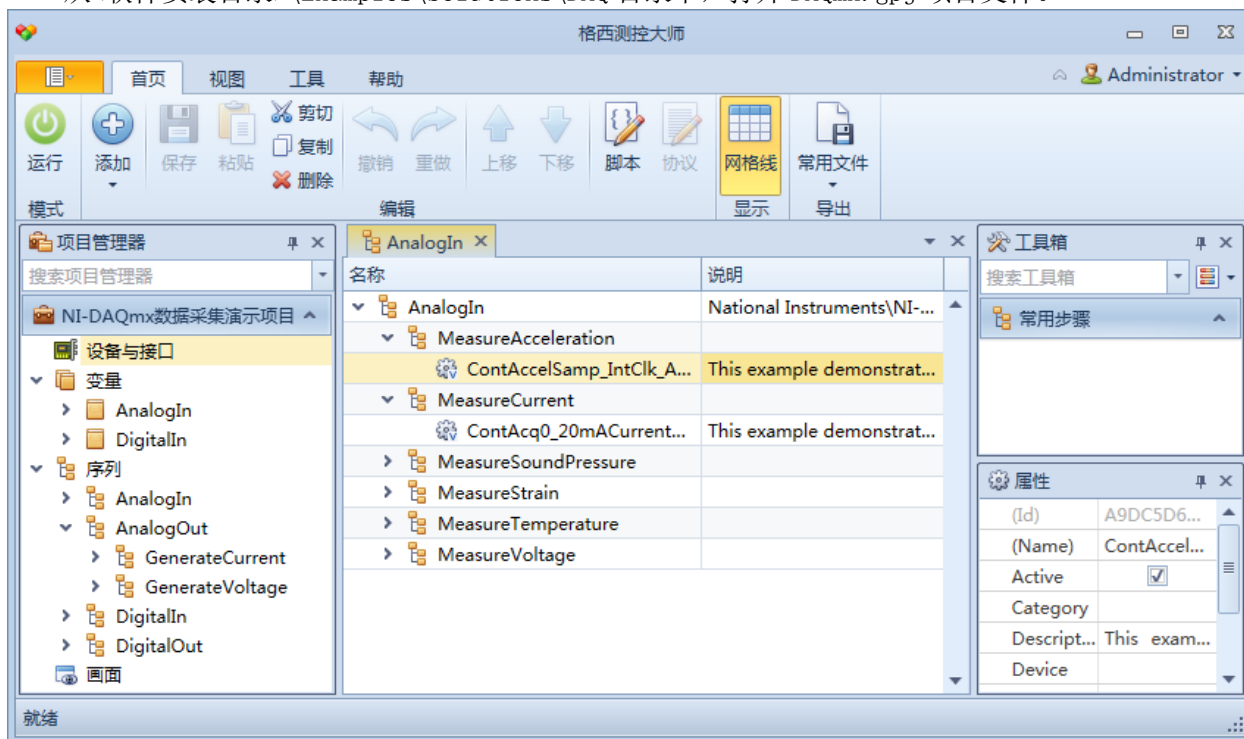
//
public Int32 EndExecute(IStepContext context, IStep step)
```

```
{  
    return 0;  
}
```

3. 运行项目

3.1 打开项目

从<软件安装目录>\Examples\Solutions\DAQ 目录中，打开 DAQmx.gpj 项目文件。

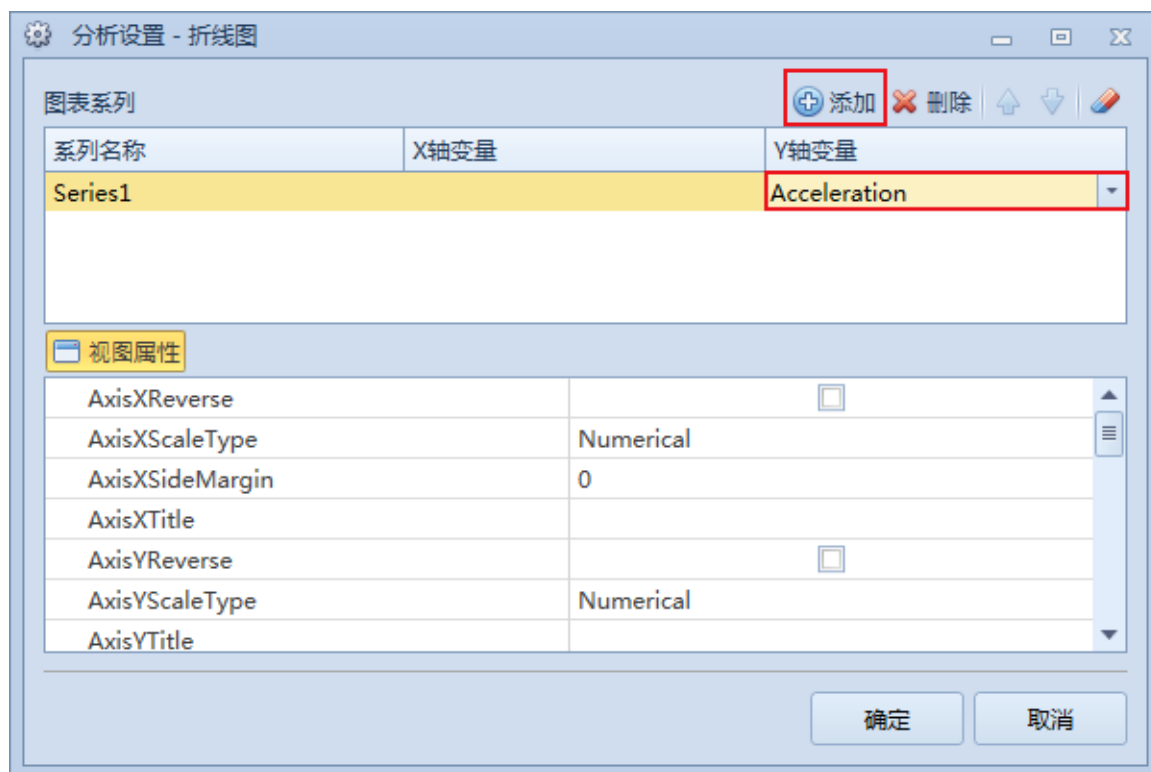


3.2 运行采集用例

点击工具栏的“运行”按钮，切换到运行模式。

1) 选择“变量->AnalogIn-> MeasureAcceleration”，点击工具栏“变量数据->变量数据”菜单，打开变量数据面板；

2) 点击工具栏“新建区域->右”菜单，新建一个区域，并通过“基本->折线图”菜单，建立变量数据的折线图。配置如下图所示，点击确定。



3) 在项目管理器中选择“序列->AnalogIn->MeasureAcceleration->”步骤，点击工具栏的“运行”按钮，运行前弹出参数设置对话框，使用默认参数执行即可。

