

边干边学仪器控制



目 录

序言	1
识 LabVIEW 仪器控制	2-7
仪器控制实战	8-12
后记	13

序言

在自动化测试领域，仪器控制是每一个自动化程序开发工程师的基本功。好多刚刚进入自动化测试领域的工程师都会问我同一个问题：“仪器控制到底难不难，有没有什么书可以推荐啊？”

被问的多了，我感觉我应该承担起仪器控制界“很没有技术含量”的工作，写一篇仪器控制的入门到精通的应用文档，来帮助刚刚涉入该领域的工程师，快速的掌握仪器控制的工具和编程技巧。

之所以说“很没有技术含量”，是因为，这个世界上，如果某项技术没有什么书籍去讲述，只有两个原因：1 是这个技术太简单了，书不够写；2 是这个技术太难了，还没有研究透彻，不能写。基于 LabVIEW 的仪器控制，就是属于“技术太简单了，书不够写”。

使用 LabVIEW 的原因

很多工程师给我写 Email，提及，可否介绍 VB/VC 下的仪器控制技术，不要因为是 NI 公司的销售工程师，就一味推 NI 公司的 LabVIEW 嘛。

其实，我并不是一味推 NI 公司的 LabVIEW，而是纯粹从技术角度上考虑，LabVIEW 是最适合开发自动化测试软件的平台。

我多次在客户那里碰到一些仪器销售人员，我总是在人群中静静的倾听他们天花乱坠的介绍，并不时的发出诸如“哇，你们用 LabVIEW 啊，这么不用 VB 呢，VB 不要钱啊~”——现在 VB 的盗版很多，在这个地区“不要钱”，不代表在世界其它地区“不要钱”；在这个“不要钱”的地区用 VB，不一定能将工作共享到世界其它地区的分公司。

另外，当提起仪器是否可以程控时，销售人员拍着胸脯说：“我用 VB 写过全部控制代码，当然可以啊。”随即，我们请销售人员马上试几条读测试值的指令，销售人员立马一头汗水，马上说“指令怎么可以记得住啊”...无语...

我常常对我身边的测试工程师讲，VB/VC 的开发公司是 Microsoft，所以在 Microsoft 内部使用 VB/VC，是既正规，又支持本公司产品。但 Microsoft 在开发自己产品 Xbox 的测试程序时，选用了 LabVIEW，而不是 VB/VC，为什么呢？——真正的智慧，不被所谓的盗版和“不要钱”蒙蔽了双眼，把精力集中于如何提高工作效率和质量，创造更多的价值。

本篇文章的读者是：

勤奋且愿意在业余时间专研新技术的工程师；

开关电源行业的研发及测试工程师；

刚刚接触 LabVIEW，想用 LabVIEW 尝试自动化的工程师；

想深入了解如何控制泰克示波器、Chroma 电子负载、功率表、程控电源、安捷伦万用表的工程师。

初识 LabVIEW 仪器控制

伟大的军事指挥官，把 LabVIEW 仪器控制技术总结为“二二二”，即二个辅助开发工具(Scan for Instrument and communicate with Instrument)，二个开发函数(VISA Read and Write)和二一个调试工具(Troubleshooting tool and NI-Spy)。

二个辅助开发工具

在进行仪器控制前，首要事情是要找到仪器，并能保证与仪器正常通讯。所以，二个辅助开发工具是：Scan for Instruments(查找仪器) 和 Communicate with Instrument(与仪器通讯)。

Scan for Instruments(查找仪器)

装好 GPIB 卡后，请双击击桌面上 Measurement & Automation(简称 MAX)图标，如图 2.1 所示。Measurement & Automation 是 NI 公司的设备管理器，类似于 Windows 的设备管理器。



图 2.1 Measurement & Automation

在 MAX 中，先选中设备与接口(Device and Interface)下的 GPIB 设备，然后单击 Scan for Instruments 按钮，如果软硬件正常，就可以在 Connected Instruments 窗口中看到已连接到 GPIB 卡上的仪器信息，如图 2.2 所示。

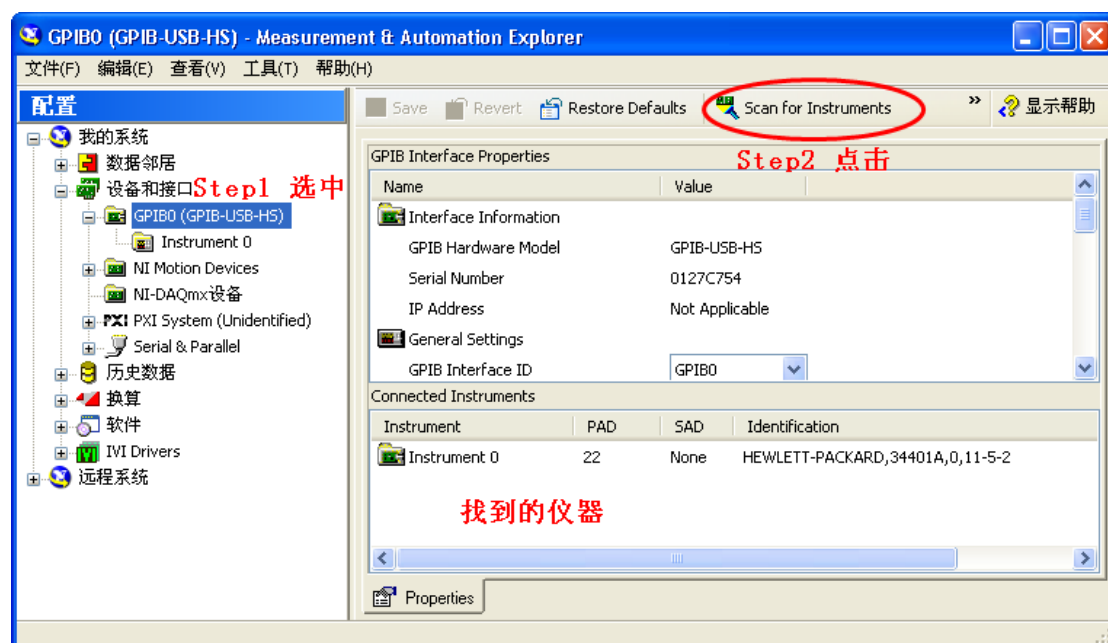


图 2.2 Scan for Instruments

该工具保证了仪器连接正常，可以进行下一步工作了。

Communicate with Instrument(与仪器通讯)

由于有些仪器的控制器可能有问题,所以,与仪器成功连接上,并不代表仪器能正确响应控制指令,所以还需要用第二个辅助工具来测试仪器响应关键指令的能力,这个工具就是 Communicate with Instrument(与仪器通讯)。

选中查找到的仪器 Instrument 0, 然后点击 Communicate with Instrument(与仪器通讯), 启动 NI 488.2 Communicator(通讯器), 如图 2.3 所示。

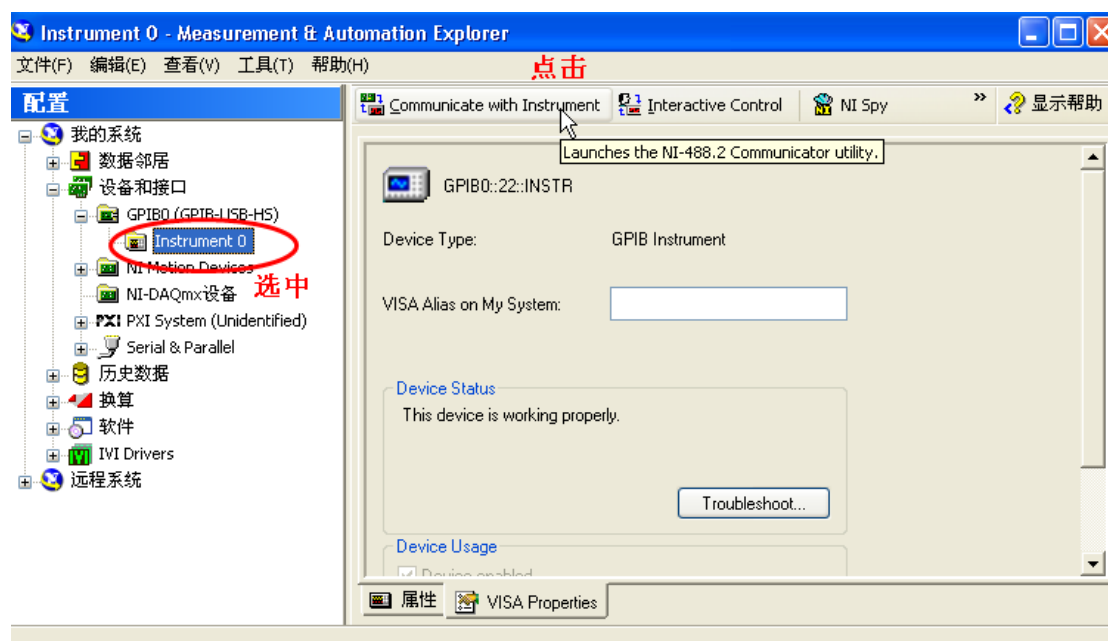


图 2.3 启动 NI 488.2 Communicator(通讯器)

在 NI 488.2 Communicator(通讯器)中, 可以对一起写入命令, 然后再读取结果, 以测试仪器对命令响应是否成功, 如图 2.4 所示。

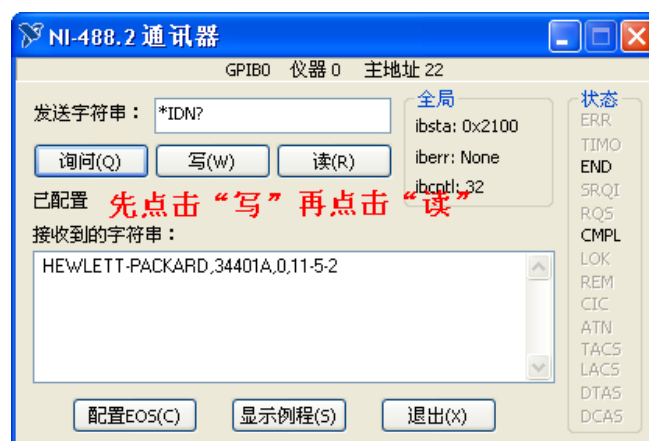


图 2.4 NI 488.2 Communicator(通讯器)

完成以上两步测试并验证仪器通讯没有问题后, 就可以开始 LabVIEW 编程了。

小贴士：很多有经验的开发人员，在开发自动化程序前，会使用 NI 488.2 Communicator(通讯器)来尝试将要使用的仪器控制命令，尝试成功，则开发的时候就心底无忧了。

二个开发函数(VISA Read and Write)

在 LabVIEW 环境下，开发 GPIB 仪器控制程序非常简单，这也是众多工程师喜欢在 LabVIEW 环境下进行自动化程序开发的原因。总的来说，只需要用 NI-VISA Write.vi 来向仪器发送命令，用 NI-VISA Read.vi 来从仪器读取数据即可，如图 2.5 所示。

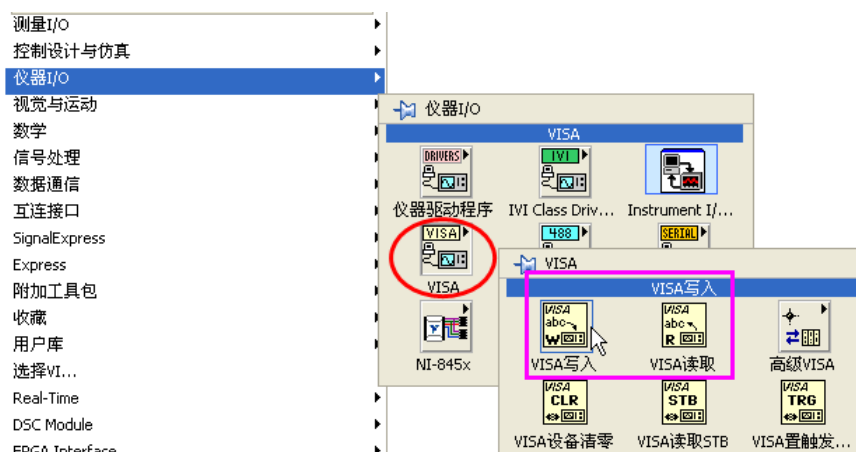


图 2.5 NI-VISA Write&Read

从 Agilent 34401A 的使用手册中 115 页可以查知，测量 DC 值的指令为“MEAS:DC?”，如图 2.6 所示。

MEASure? 范例： 下面一程序段示出如何使用 MEASure? 命令来作测量。这个范例将万用电表配置为测量 DC 电压，并自动将万用电表设定至“等待触发”状态，内部触发万用电表取一个读数，然后将读数送到输出缓冲器上。

```
MEAS:VOLT:DC? 10,0.003
```

```
bus enter statement
```

这是读取读数最简单的方法。不过，您没有任何机会使用 MEASure 来设定触发计数、取样计数和触发延迟等。除了功能，量程和分辨率之外，所有测量参数都会自动预设(请参阅第 112 页上的表)。

图 2.6 Agilent 34401A 使用手册

所以，用 NI-VISA Write 向仪器发送指令：MEAS:VOLT:DC?，然后用 NI-VISA Read.vi 读回 DC 电压测量值，如图 2.7 所示。

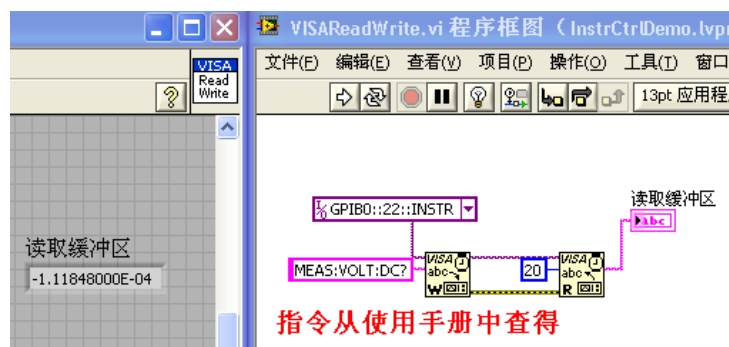


图 2.7 读取 DC 范例程序

【总结】仪器控制程序开发，只有三个步骤：

- 查阅仪器使用手册，找到所需的仪器指令；
- 用 NI-VISA Write.vi 向仪器发送该指令；
- 用 NI-VISA Read.vi 从仪器中读回数据；

NI 仪器驱动网

刚刚开始学习 GPIB 控制仪器的工程师，会为成功控制仪器而欣喜。可是，当写过几个仪器控制程序后，再去重复查阅手册->发指令->读数据的工作，就会感觉乏味和痛苦，感觉一点长进也没有，感觉自己的时间全浪费在看仪器的指令手册上了。

早在二十多年前(1986 年)，NI 的工程师已经意识到这个问题了，为了解决工程师的乏味问题，NI 的工程师把写过的常用的仪器驱动函数加以总结和整理，免费发布给客户，之后，各大仪器公司也参照 NI 的做法，为自己的仪器配上了 LabVIEW 仪器驱动程序。NI 把这些前人总结好的仪器驱动程序放到了网上，成就了今天的 NI 仪器驱动网。

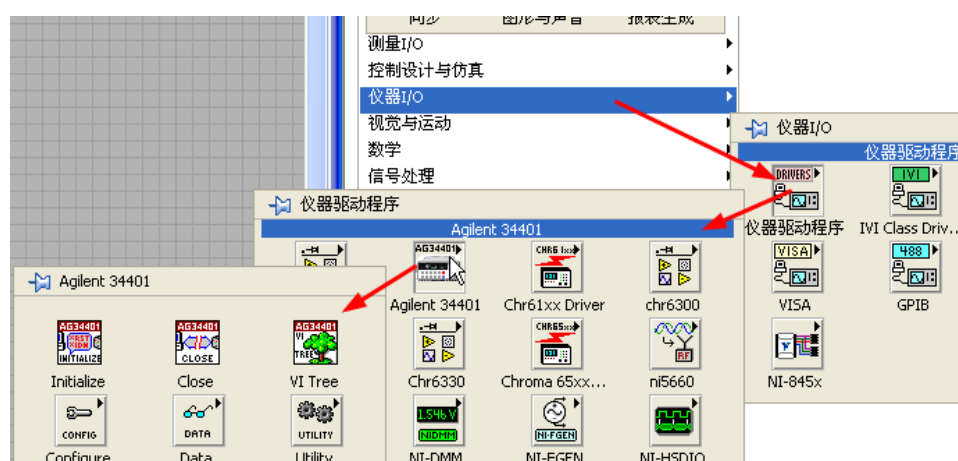
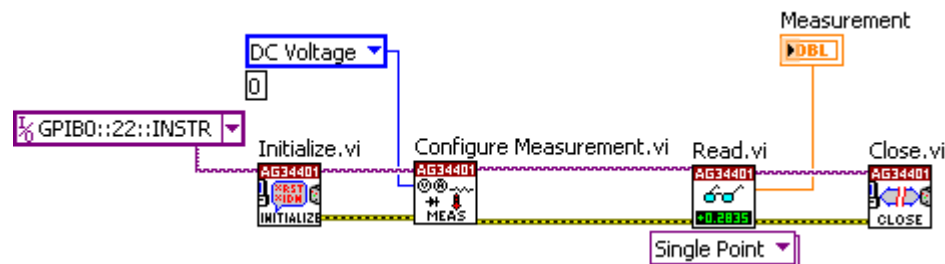


图 2.8 Agilent34401 驱动程序



二个调试工具(Troubleshooting tool and NI-Spy)

当程序出错的时候，需要用两个调试工具来诊断：一个是 Troubleshooting tool，诊断 GPIB 硬件和 GPIB 驱动程序的问题；另一个是 NI-SPY，诊断 NI-VISA 指令收发的问题。

Troubleshooting tool

在 MAX 中选中 GPIB 卡，然后点击右键菜单中的 NI-488.2 Troubleshooting Utility，如图 2.9 所示。

如果软件(NI-488.2)出错，可以重装 NI-488.2 软件；如果 GPIB 出错，则寄回 NI 公司修理，如图 2.10 所示。

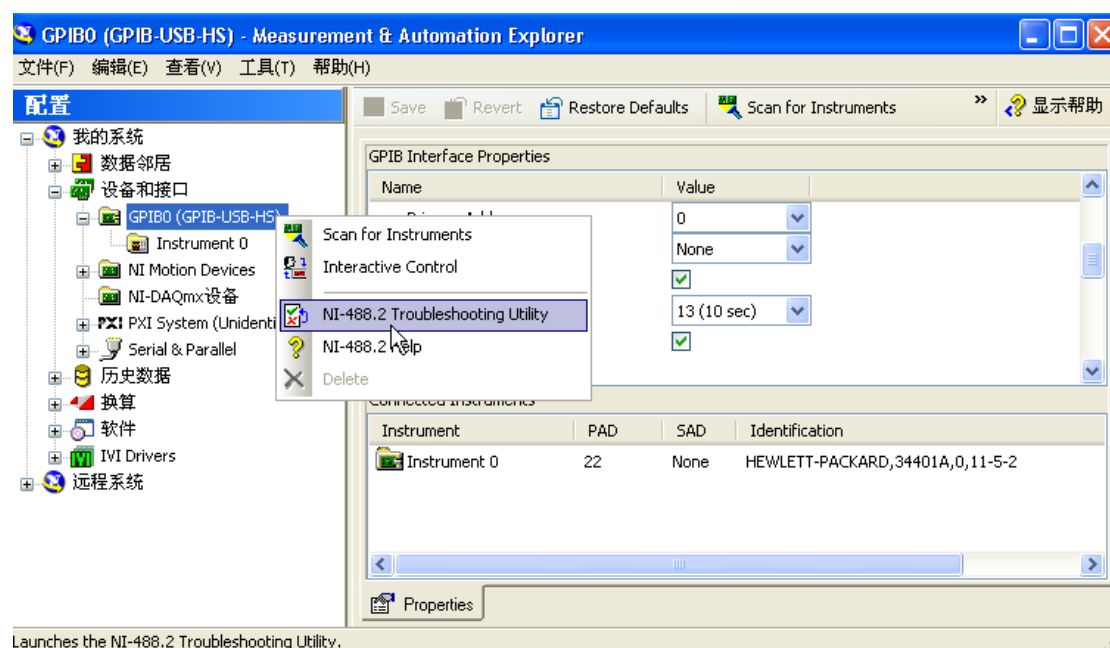


图 2.9 启动 NI-488.2 Troubleshooting Utility



图 2.10 GPIB 软硬件检测结果

仪器控制实战

工程师常向我述说，其实，在控制仪器的过程中，也只用一小部分功能，大部分指令都用不到，不想花太多时间去研究，能不能给点范例程序，只展示这一小部分常用功能就好了。

本章主要总结安捷伦 34401 万用表，泰克示波器 DPO7054、Chroma 电子负载 63303、功率表 66202 和程控电源 6560 的常用功能的程控技术。

安捷伦 34401 万用表

安捷伦 34401 万用表的程控功能比较简单，可以读万用表的测量值即可。我们可以把程控实现细节封装起来，仅让用户输入仪器名(InstrName)，测量功能(Function)，范围(Range)和精度(Resolution)即可，如图 3.1 所示，具体程序实现参见范例 Agilent34401Read.vi。

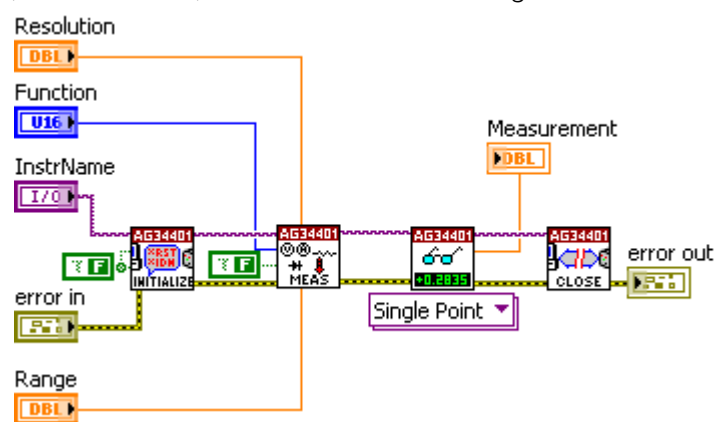


图 3.1 Agilent34401Read.vi

泰克示波器 DPO7054

准备工作：下载

泰克示波器的常用程控功能有：设置通道参数，设置触发，设置光标，读取测量值，拷贝波形到主机。

设置通道参数

设置触发

设置光标

读取测量值

拷贝波形到主机

程控电源 Chroma 6530

对于程控电源来说，最主要的是要掌握以下三个功能：

静态输出设置——用于电源多数常规测试；

List 模式输出设置——用于电源 Cycle Drop 测试；

读取测量值——用于 ATE 测试。

静态输出设置

静态输出设置最简单，只需要告诉 AC Source 输出电压和输出频率即可，请参考范例程序 ACOutputDemo.vi，如图 3.2 所示。

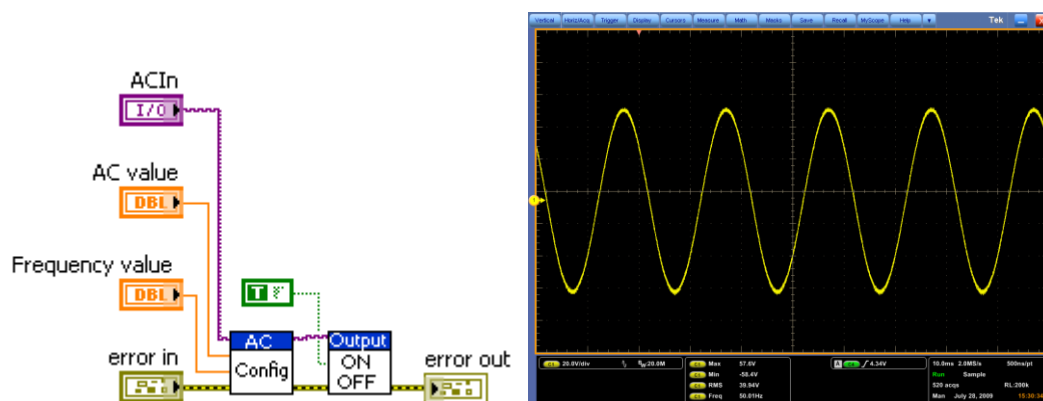


图 3.2 ACOutputDemo.vi 和示波器测试结果图片

List 模式输出设置

List 模式相比静态输出，主要区别是要配置一组输出的电压和频率值，而不是一对电压和频率值。具体实现代码请参考范例程序 ListConfDemo.vi，如图 3.3 所示。

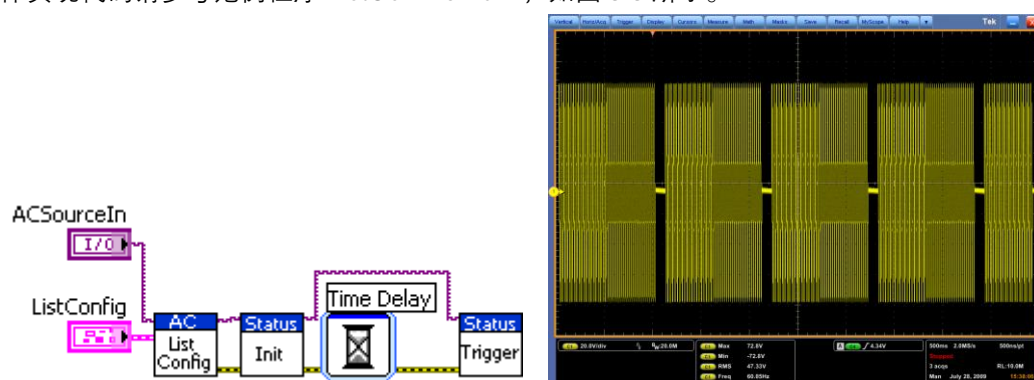


图 3.3 ListConfDemo.vi 和示波器测试结果图片

读测量值

装了 Chroma 65xx 的仪器驱动后，可以找许多测量 VI，如图 3.4 所示。使用这些 VI，可以直接读取输出电压、频率、电流、功率因素、电流峰值因素、Inrush 电流等等。

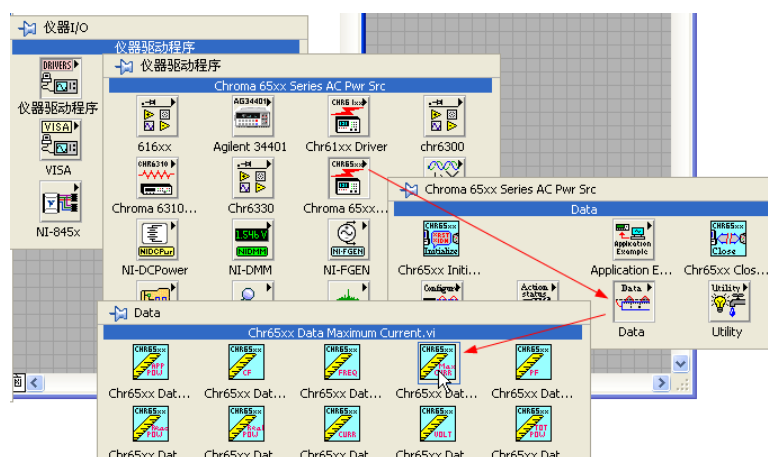


图 3.4 Chroma 65xx 测量 VI

电子负载 Chroma 6330

恒电流和动态电流设置

电子负载的控制主要包括恒电流设置、动态电流设置和多模块同步。恒定电流设置和动态电流设置，大家可以参考 LoadConfig.vi，如图 3.5 所示：

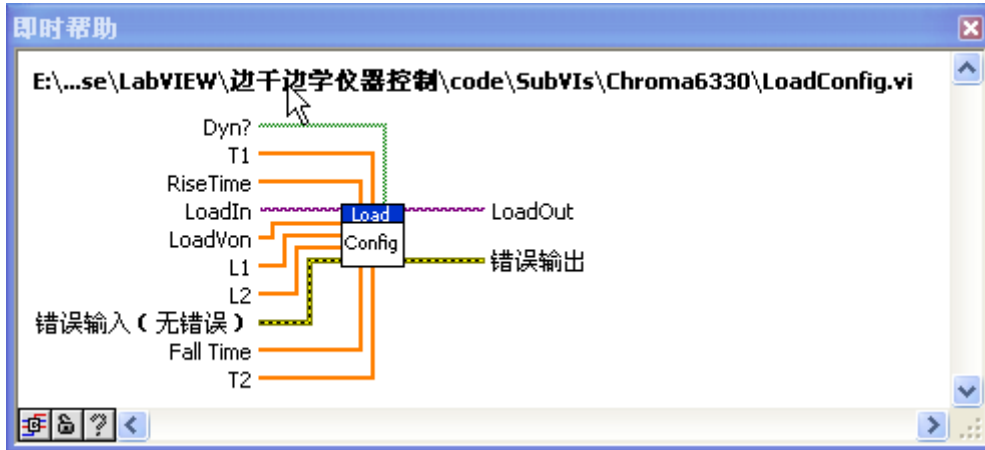
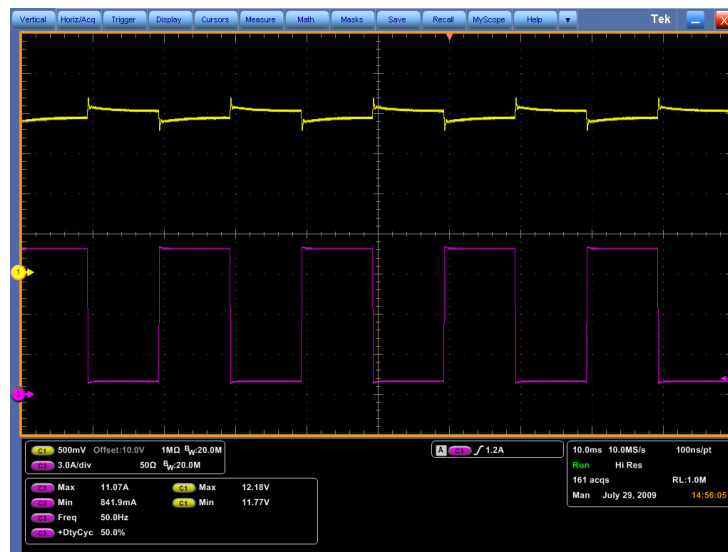


图 3.5 LoadConfig.vi

参数：Dyn? 是告诉电子负载工作在动态(Dynamic)模式还是静态模式(Static)；
T1, T2 是电子负载工作在动态模式下，负载电流 L1 和 L2 持续的时间；
L1, L2 是电子负载工作在动态模式下的负载电流，在静态模式下，仅 L1 有效；
RiseTime 和 FallTime 是电子负载工作在动态模式下的上升时间和下降时间；
LoadVon，是电子负载的 Von 点。

下图是负载工作在动态模式下的示波器截图，紫色的波形是电流，黄色的波形是电压。



多模块同步

当一个负载模块功率不够的时候，可以把多个模块并联起来，以提高负载功率。比较难编写的程序是动态模式下多模块的同步，大家可以参考 LoadConfigSync.vi，如图 3.6 所示。

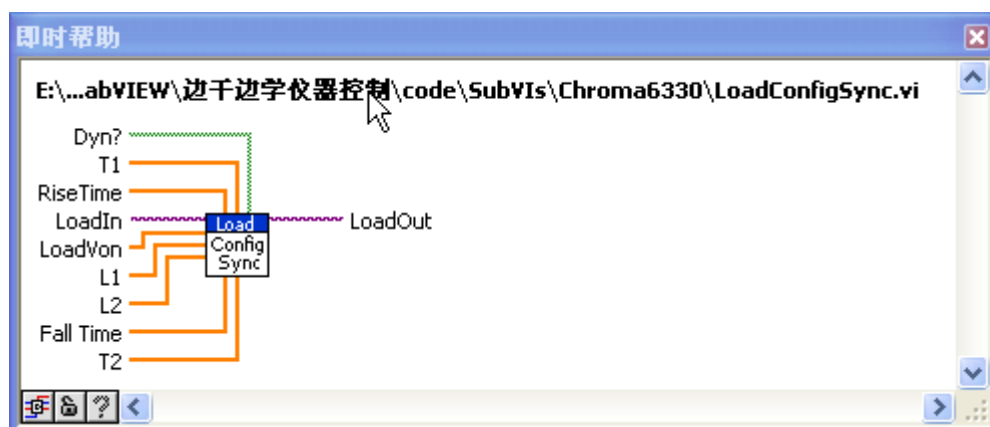


图 3.6 LoadConfigSync.vi

其参数的意义与 LoadConfig.vi 一致。图 3.7 是多模块同步的示波器截图，黄色是电压波形，紫色和绿色是不同模块的动态电流波形。

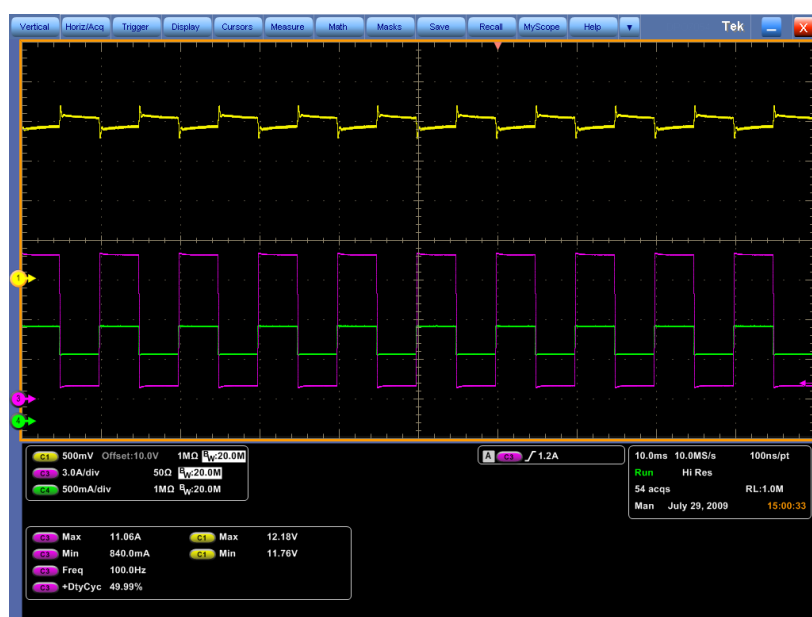


图 3.7 多模块同步

功率计 Chroma66202

功率计的程控思想和万用表 34401 一样，先告诉功率计要读什么值，然后把值读出来即可，如图 3.8 所示。Reading? 是告诉功率计要读取什么类型的值，Value 即为返回值。

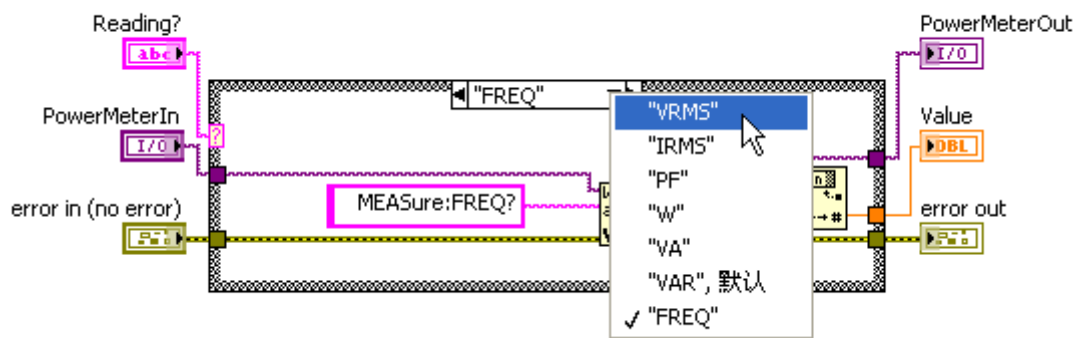


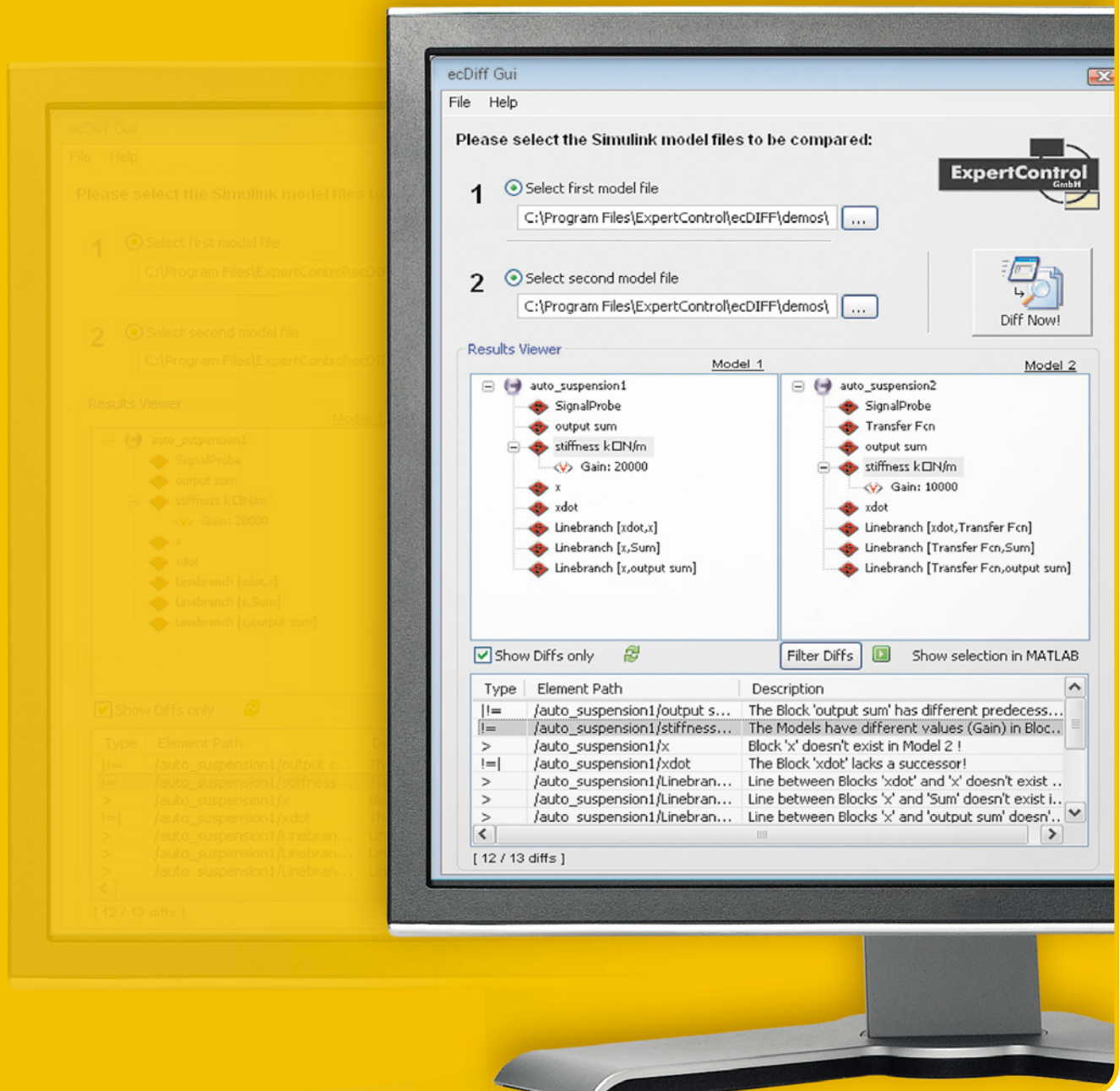
图 3.8 Read66202.vi

后记

"I hear and I forget;
I see and I remember;
I do and I understand."

眼见为实，耳听为虚，任何事都要亲自付以行动才能真真正正的理解。在 LabVIEW 这样一个非常适合工程师和科学家的平台上，多尝试，多实践是工程能力增长的不二法门。

边干边学报表生成



目 录

序言	1
初识 LabVIEW Office 报告生成工具包	2-10
再识 LabVIEW Office 报告生成工具包	11-17
后记	18

序言

在自动化测试领域，生成的 Office 报表(Word,Excel)几乎是每个专业的自动化测试程序的标配。不具备自动报表生成的自动化测试程序通常被视作“入门级”程序，就像汽车里面的“奔奔”或者“QQ”。先前，生成专业的 office 报表几乎被少数几个 VB 或 VC 高手“垄断”，因为你除了要熟悉 VB 或 VC 外，还要熟悉 Windows 下的 ActiveX 机制以及 Word 和 Excel 导出的属性和方法。

NI Office 报告生成工具包的出现打破了上述的技术壁垒，使得任何一个非计算机专业毕业且并不精通 Windows ActiveX 机制的工程师都能做出一份能与 VB、VC 高手媲美的专业的 Office 报表。我向给大家介绍这个工具包，并不是想要“剥夺”大家学习 ActiveX 的机会，成为某些偏执程序员所宣称的“LabVIEW 依赖者”。相反，我内心所考虑却是希望尽我所能，帮助大家把宝贵的时间从繁琐的技术细节中解脱出来，投入到核心价值的创造中去。另外，NI Office 报告生成工具包并不是以 dll 的形式给出，而是以源代码的形式给出——只要你愿意，你可以在完整的商业源代码的基础上随时研究 office 报告生成的所有细节。

本篇教程适合的读者是：

- 勤奋且愿意在业余时间钻研新技术的工程师；
- 正在使用 LabVIEW 编写自动化测试代码的工程师；
- 想学习如何生成 Office 报表的工程师；
- 不想阅读 office 报告生成工具包英文帮助文档的工程师。

初识 LabVIEW Office 报告生成工具包

LabVIEW Office 报告生成工具包升级到 1.1.2 版本后就变得非常可爱了，因为它提供了一个基于交互式配置的 Express VI——MS Office Report, 如图 2.1 所示。本章就从这个 Express VI 开始，与大家一起进入 LabVIEW Office 报告生成的世界。

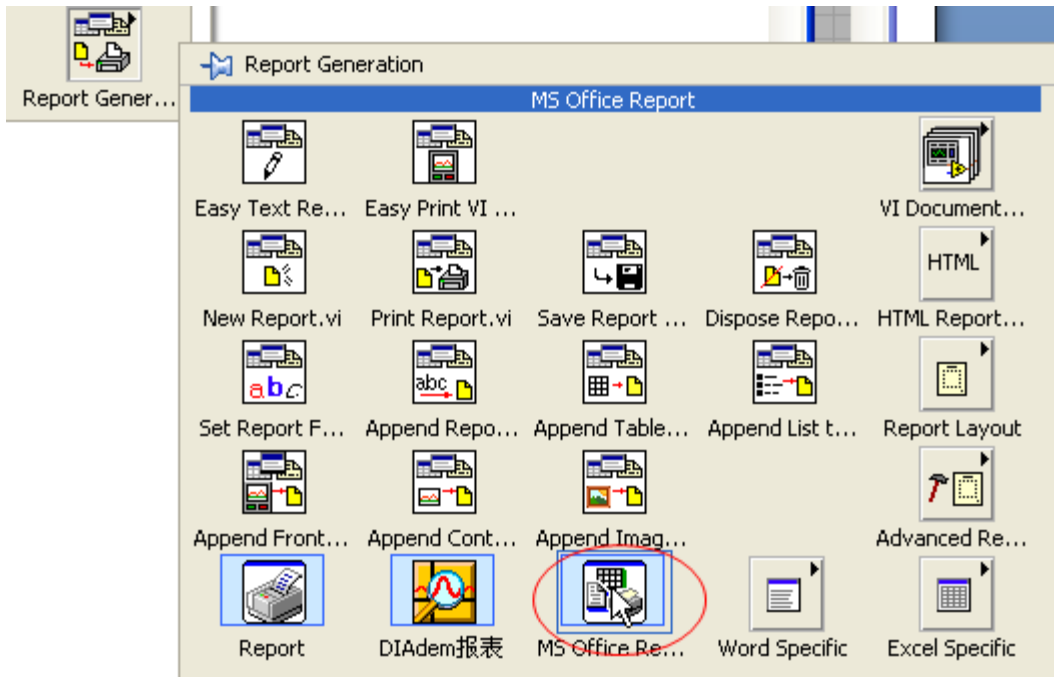


图 2.1 MS office Report

毛主席说过，要在“战略上藐视敌人，在战术上重视敌人”，对于刚接触 LabVIEW office 报告生成工具包的工程师来说，由于陌生而产生一点畏难情绪是非常正常的，就像台儿庄战役以前感觉日本鬼子很难打一样——其实打过了就觉得不难打了。LabVIEW office 报告生成工具包也是一样，用过了就觉得很好用了。

总的来说，要用好 LabVIEW office 报告生成工具包需要做好两件事——“Where”和“What”，即告诉 LabVIEW office 报告生成工具包，在 Office 文档的哪个位置，放上什么内容即可。

请告诉我内容放在哪里？——Excel 版

请大家打开 Excel，在 Excel 的左上角有一个 name box，如图 2.2 所示。Name box 相当于给单元格起了一个名字，方便开发人员记忆和在程序中使用。比如，我们给 A2 单元格起个名字叫“Operator Name”，那么在 LabVIEW 里面告诉 MS office report.vi，“刘邦”的位置是“Operator Name”，则 MS office report.vi 就会把“刘邦”写入 A2 单元格了。使用 Namebox 还有一个好处是，当你想把更改“刘邦”的写入位置时，只需要把对应的单元格命名为“Operator Name”即可，而不需要更改 LabVIEW 程序。

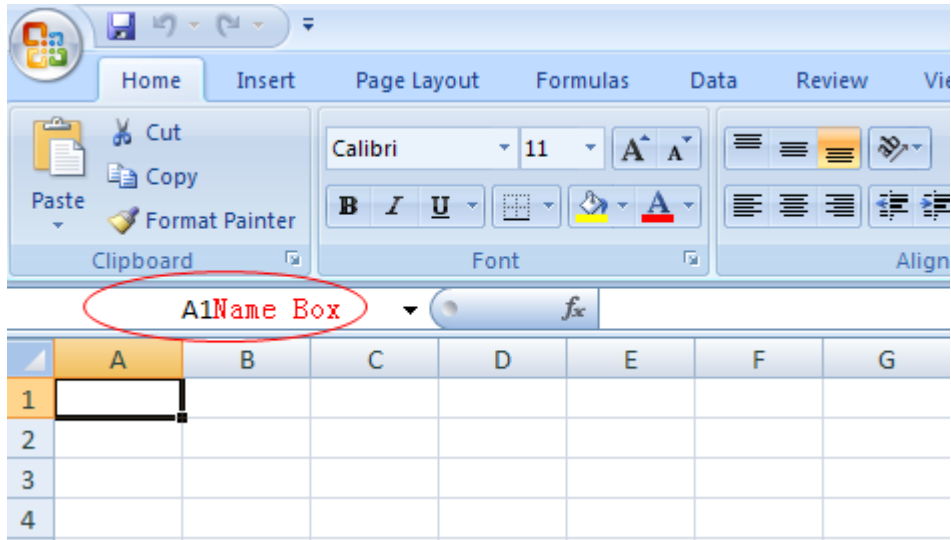


图 2.2 Excel 里面的 Name box

为了使我们的第一个报告看起来“专业”一点，我们在 A1 单元格中输入报表的名字“美国国家仪器测试报告”，然后把 A3 单元格命名为“OperatorName”，如图 2.3 所示。Excel 的 Name box 输入还有点儿小小的顺序性技巧：

1. 单击选中 Excel 单元格
2. 在 Name box 里面输入名字并敲回车

点击 Name box 右边的下拉箭头(如图 2.4 所示)，会发现名字已经在 Name box 里面了。Name box 里面的命名规则请见“Google”。

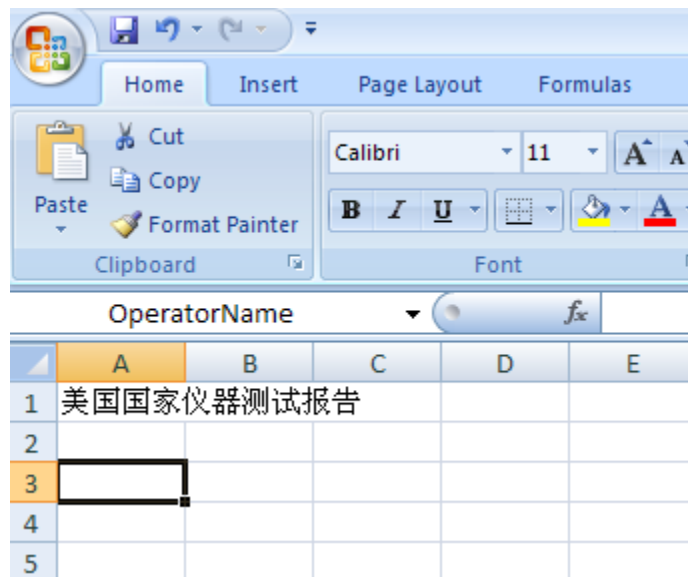


图 2.3 把 A3 命名为“OperatorName”

接着把 A4 单元格命名为“Time”，把 A5 单元格命名为“Value”，如图 2.4 所示。

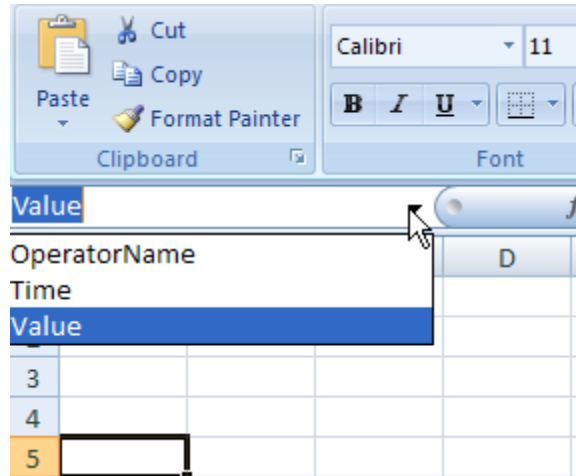


图 2.4 单元格的别名

完成上述步骤后，请把该文件以 Excel97–2003 模板的形式保存，如图 2.5 所示。

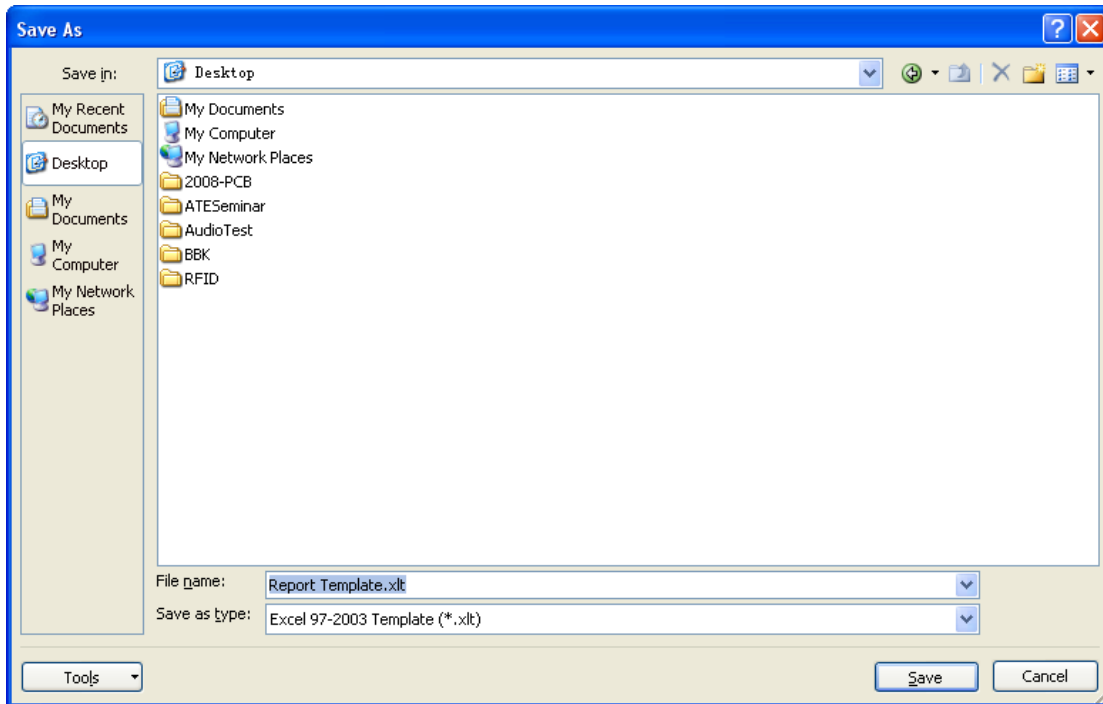


图 2.5 保存成 Excel97–2003 模板

成功完成上述步骤之后，恭喜大家，已经完成了整个报告生成工作量的 90%，以后大家会体会到，**基于 LabVIEW 的报告生成**，大量的工作是在设计报告模板上。

请打开 LabVIEW，并在程序框图中放入 MS Office Report.vi，这时会弹出配置对话框，如图 2.6 所示，然后在第一项中选择“Custom Report for Excel”。

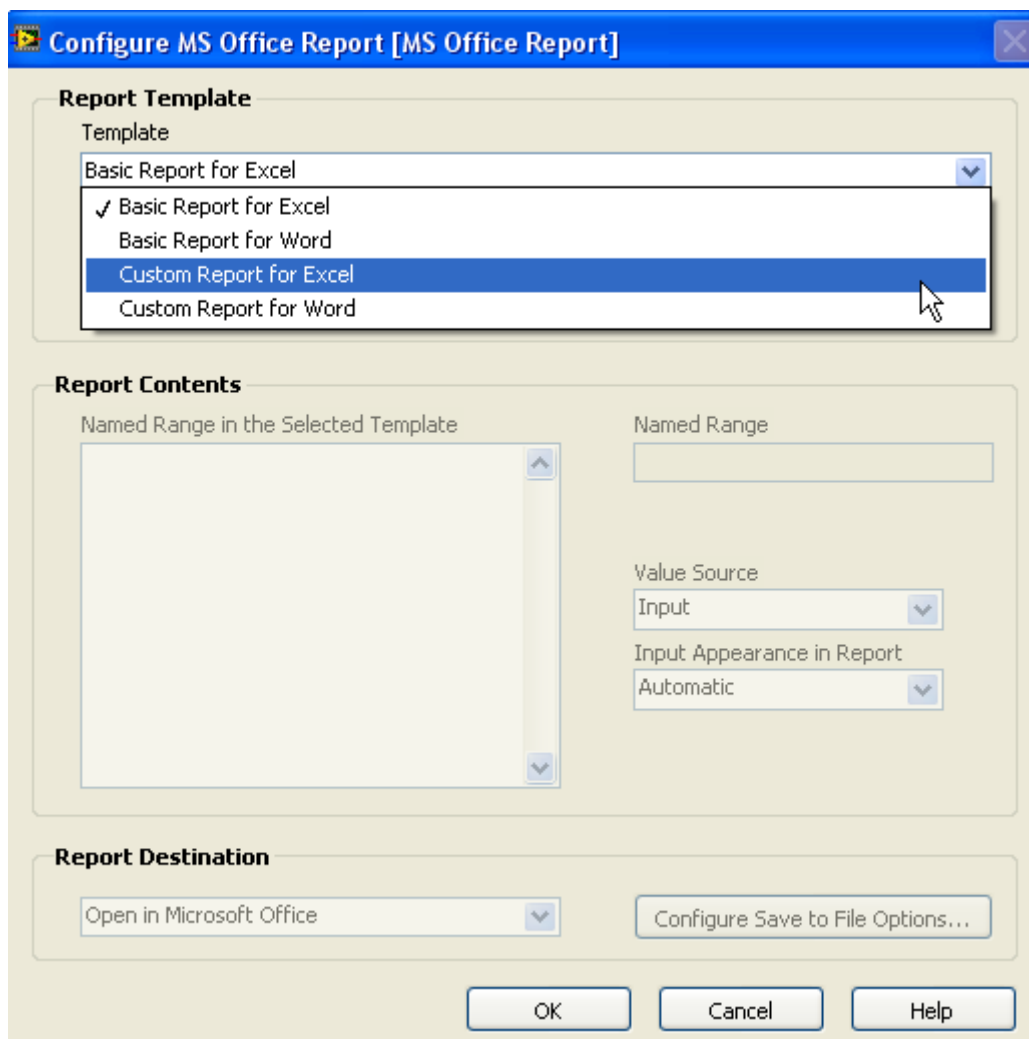


图 2.6 MS Office Report.vi 配置对话框

接着在 Path to template 里面选中刚才保存的模板，如图 2.7 所示，大家可以发现，MS office report.vi 会自动找到命名过的单元格。点击 OK 按钮，完成配置。

到这里，“Where”就完成了，即完成了告诉 LabVIEW 在**哪里**放置你想插入的内容。

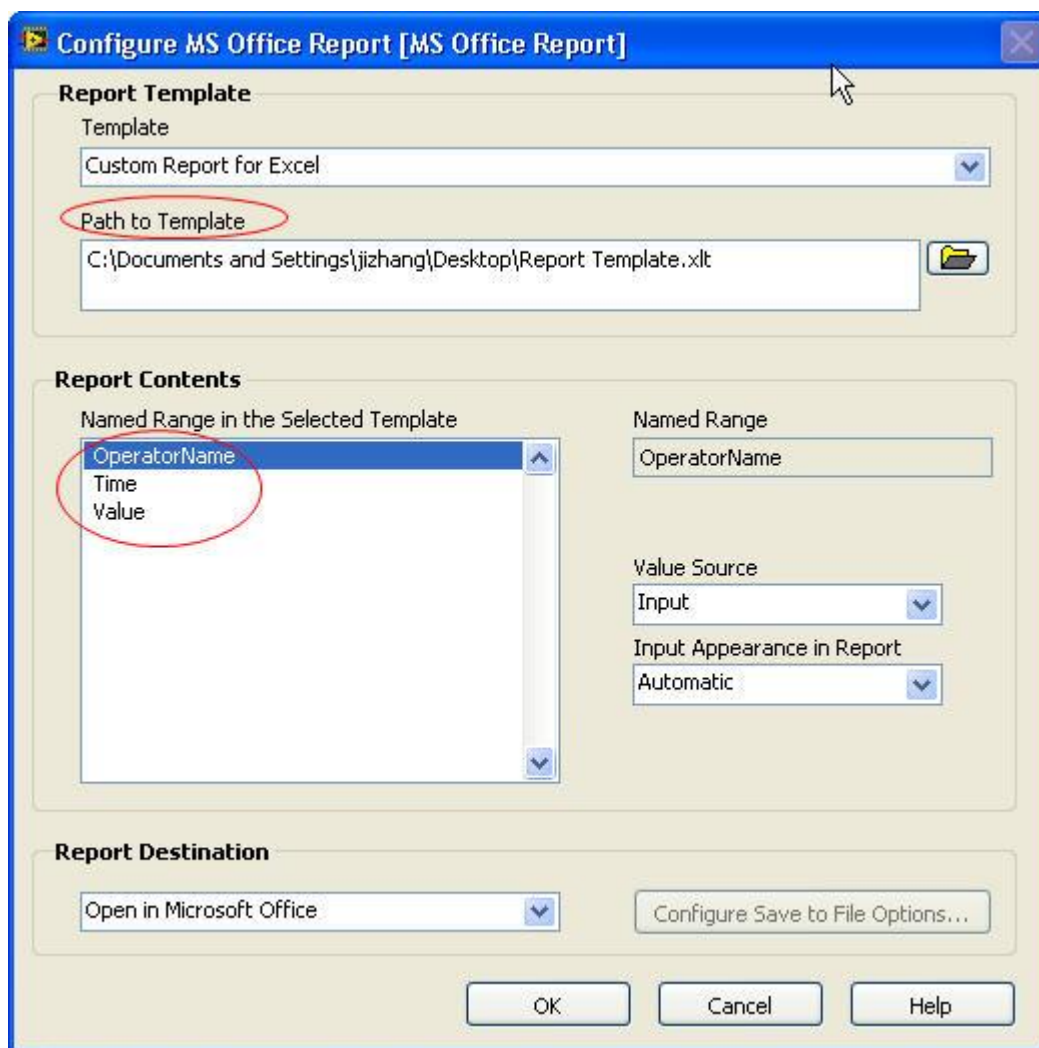


图 2.7 选择报告模板

在 LabVIEW 程序框图中，我们为“OperatorName”输入“刘邦”；为“Time”输入当前日期；为“Value”输入一个随机数，如图 2.8 所示。MS office Report.vi 可以接受各种类型的输入，大大方便了我们编程。

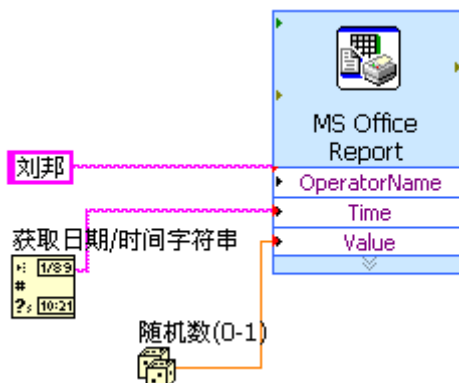
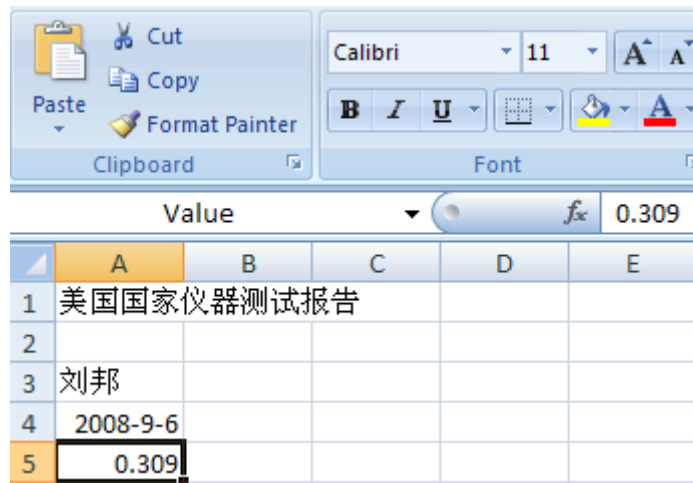


图 2.8 报告生成程序

运行程序，结果如图 2.9 所示，可见 MS office Report.vi 已经把数据插入了我们指定的地方，第一次尝试，圆满成功！~☺



	A	B	C	D	E
1	美国国家仪器测试报告				
2					
3	刘邦				
4	2008-9-6				
5	0.309				

图 2.9 运行结果

对于简单的报告应用，大家学到这里就可以了。想做出格式精美专业的报告，请继续阅读第三章，这时候，除了“Where”和“What”外，还需要告诉 LabVIEW，“How”，即以什么格式把内容插到指定位置去。

请告诉我内容放在哪里？——Word 版

与 Excel 中的 Namebox 作用相同，Word 中用 Bookmark 来为一个位置命名，如图 2.10 所示。MS office report.vi 可以找到 Word 模板中有 Bookmark 的位置。

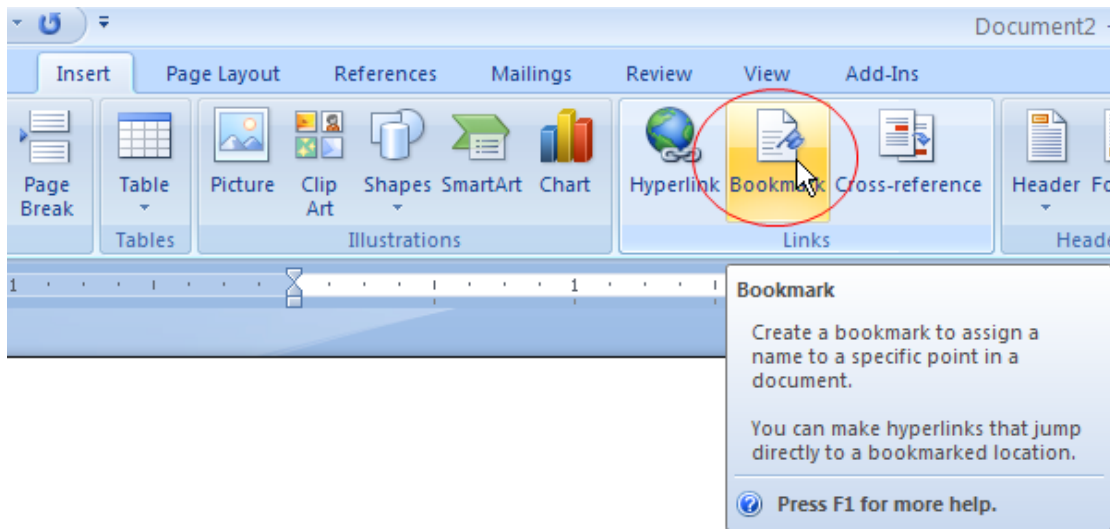


图 2.10 Word Bookmark

与上节相同，我们开始做一个简单的 Word 模板，第一行键入“美国国家仪器测试报告”，第二行键入“操作员姓名”，这时，光标停留在“操作员姓名”，我们为此位置添加一个 Bookmark，如图 2.11 所示。

美国国家仪器测试报告

操作员姓名: |

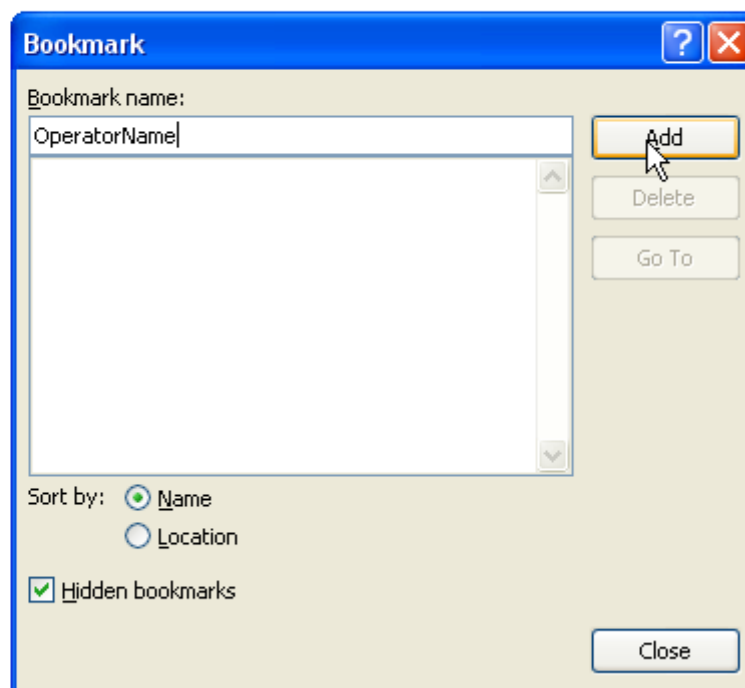


图 2.11 插入 Bookmark

同 Excel，为 Word 创建“测试时间”，“测试值”的 Bookmark，如图 2.12 所示：

美国国家仪器测试报告

操作员姓名:

测试时间:

测试值: |

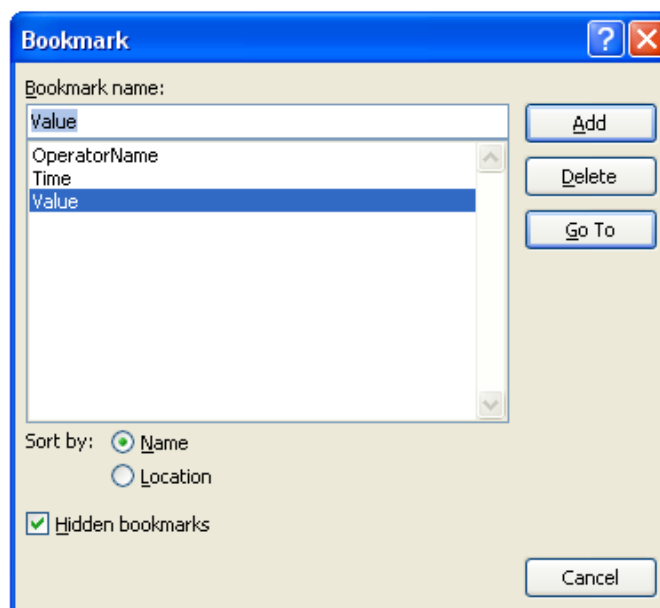


图 2.12 Word 报告模板

制作好 Word 模板后，请保存为 Word97–2003 template 格式，如图 2.13 所示。

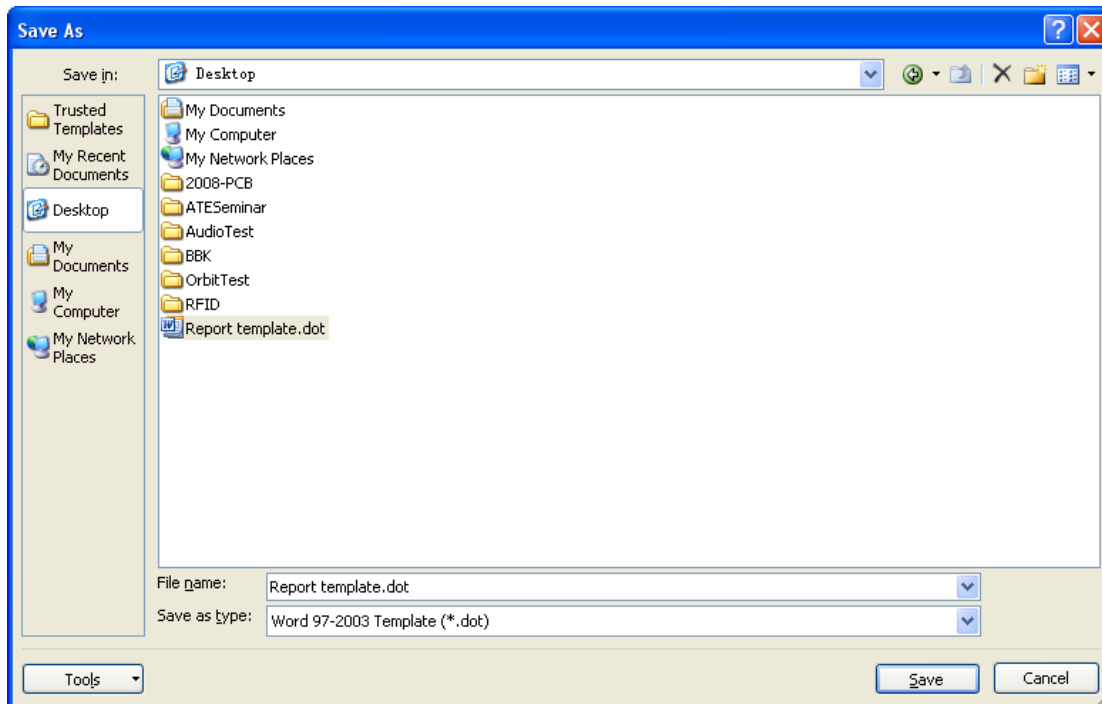


图 2.13 保存为 Word97–2003 template 格式

当 Word 模板做好后，我们可以像上节中那样，使用 MS Office Report.vi 向模板插入内容。在 MS Office Report.vi 的配置窗口中，可以看到我们创建的 Bookmark，如图 2.15 所示。

同样，我们可以为 MS Office Report.vi 输入参数，如图 2.14 所示。

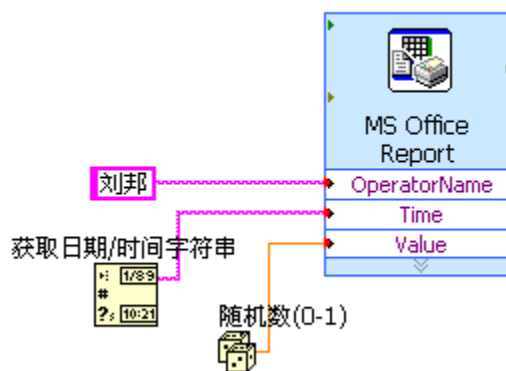


图 2.14 LabVIEW 报告生成程序

运行程序，可以看到生成的报告如图 2.16 所示。第二次尝试，再次成功！☺

通过上面的实战，我们可以看到，LabVIEW Office 报告生成工具包的精髓就是“Where”和“What”，Excel 中通过 Namebox 来定位，Word 中通过 Bookmark 来定位。准备好内容，并告诉 MS Office Report.vi 位置在哪里，MS Office Report.vi 就会把内容精准的插到您指定的位置中去

了。

在下节中，我们将再识 Office 报告生成工具包，我们会使用 Office 报告生成工具包中的普通 VI 来创建更加漂亮的文档。

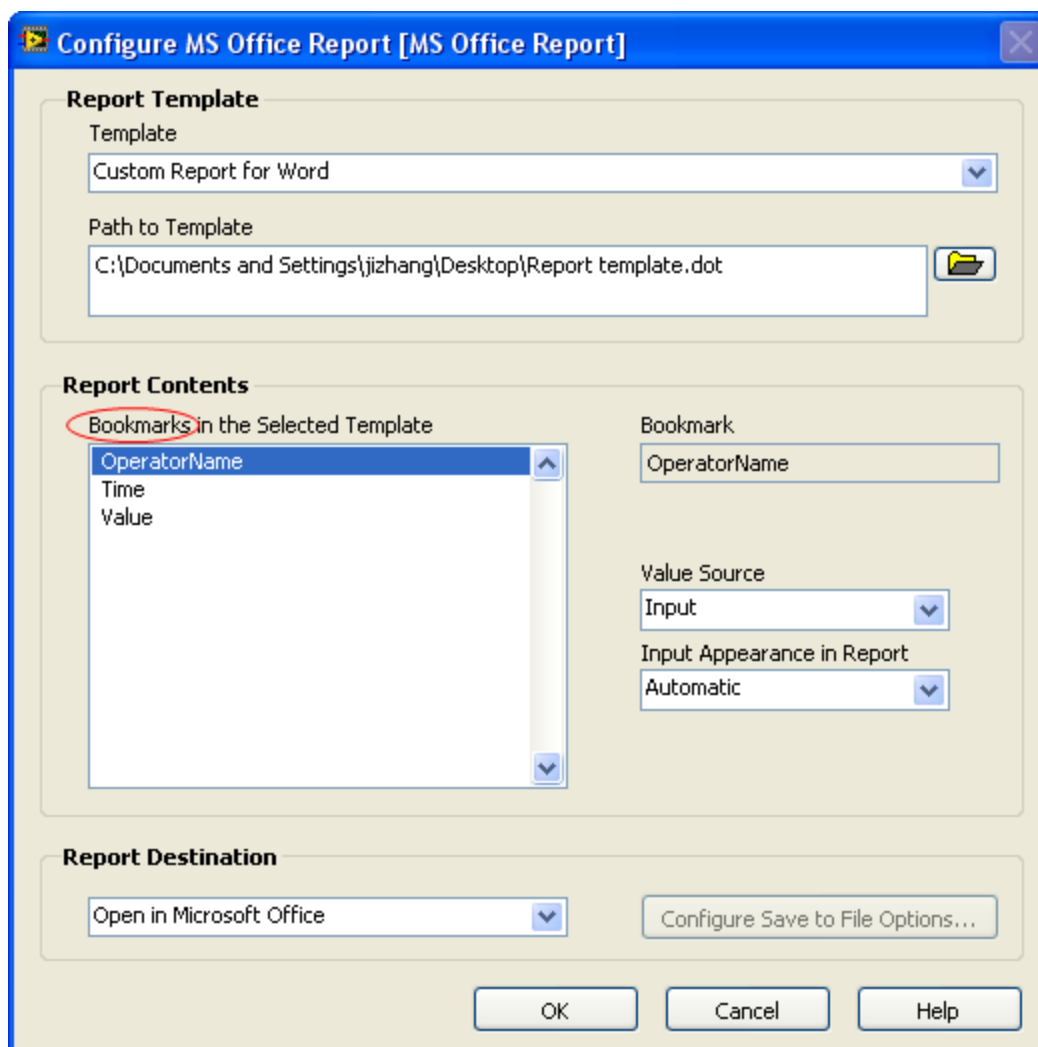


图 2.15 从 MS Office Report 配置窗口看到的 Bookmark

美国国家仪器测试报告

操作员姓名: 刘邦

测试时间: 2008-9-6

测试值: 0.084

图 2.16 自动生成的报告

再识 LabVIEW Office 报告生成工具包

俗话说“做事做细”，当我们能利用 MS Office Report.vi 成功生成报告之后，接下来的想法肯定是如何把这个报告做的更漂亮。

本章主要与大家分享如何编辑单元格/字体的格式，下一章会与大家分享如何插入图片，表格...通过编辑单元格/字体的格式和插入图片和表格，可以使报告变得更加漂亮，更加专业。MS Office Report.vi 这个 Express VI 中没有格式设置选项，所以格式的编辑都将用普通 VI 实现。本章使用的开发环境是 LabVIEW8.6，MS Office report generation toolkit 1.1.3。

使用普通 VI 生成报告

在使用普通 VI 前，先看看 MS Office report generation toolkit 函数面板的构成，如图 3.1 所示。

Part I 是一些基础性的报表生成 VI，最常用的是“新建报表”——如同打开文件一样，“保存报表至...”——保存报表，如同保存文件一样，和“处置报表”——释放资源，如同关闭文件一样。



图 3.1 MS Office report generation toolkit 函数面板的构成

Part II 是与 Word 和 Excel 操作相关的 VI，向 Word/Excel 插入文本，图片，表格以及编辑格式等都在这里。

Part III 是插入 VI 说明信息以及设置打印格式的 VI。

了解这些 VI 的功能后，我们可以使用“新建报表”，“Excel Easy Text.vi”，“保存报表至...”和“处置报表”来实现先前 MS Office Report.vi 实现的功能，如图 3.2 所示。



图 3.2 生成简单的 Excel 报告

LabVIEW8.6 中文版有一个非常好的特性——帮助文档都是中文，这就非常方便初学者学习。

图 3.3 是“新建报表”VI 的帮助文档截图。通过中文帮助文档，我们可以很容易理解上面的程序的意思。我们通过 New Report.vi 告诉程序按照 Report Template.xlt 模板生成一个新的 Excel 格式的报表，通过 Excel Easy Text.vi 告诉程序在 OperatorName 的位置插入内容刘邦，通过 Save Report to File.vi 告诉程序把生成的报告保存在 C:\Test.xls，最后释放所有占用的资源。

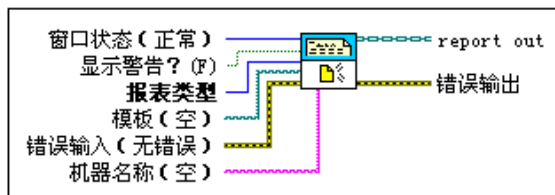
新建报表 (VI)

所属选板：报表生成VI

安装：基础软件包

创建新报表。

如不使用简易文本报表VI，必须使用该VI创建新报表。



■ 放置在程序框图上 ■ 在函数选板中定位

窗口状态 (Report Generation工具包) 设置Microsoft Word或Excel窗口正常显示、最小化或最大化。

0	maximized
1	minimized
2	normal (默认)
3	no change

图 3.3 新建报表帮助文档

到此为止，我们具备了用普通 VI 来生成报告的能力。感兴趣的读者请利用普通 VI 把 Time 和 Value 也插入到报告中去。

在工程实践过程中，我发现绝大部分用户都倾向于使用 Excel 来制作产品测试报告，理由有：Excel 里面有丰富的数学和图表工具；Excel 方便做统计；Excel 形式的报表符合自动化测试结果的表达习惯...

为减少冗余的文字，本文从现在开始仅以 Excel 报表为范例来讲述各种报告生成技术，Word 形式的报告生成技术与 Excel 大致相同，大家可以参考第二章的模式，自行研究，当做完学完本文之后的作业。

设置字体

设置 Excel 单元格的字体，最常用也是最容易的方法是调用报告生成工具包中，Excel Format 子集下面的 Excel Set Cell Font.vi，如图 3.4 所示。

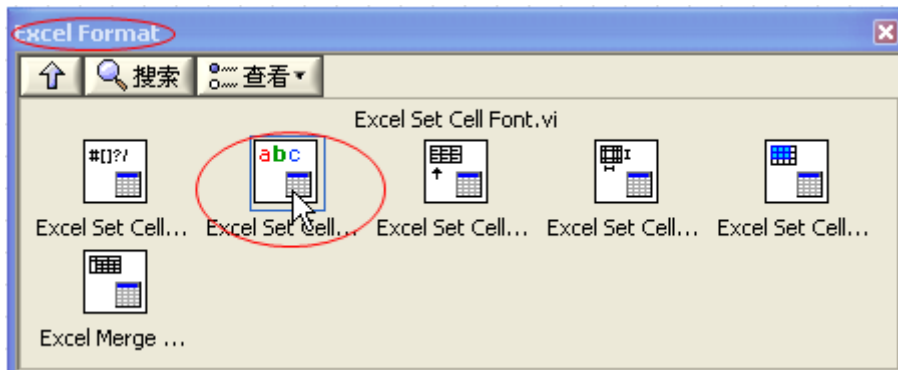


图 3.4 Excel Set Cell Font.vi

在 Excel Set Cell Font.vi 的即时帮助窗口(图 3.5)中可以看到，Excel Set Cell Font.vi 用法很简单，只需要告诉 Excel Set Cell Font.vi 以什么样的字体（font）格式化哪个单元格(name/start)即可，用法与 Excel Easy Text.vi 很相似。

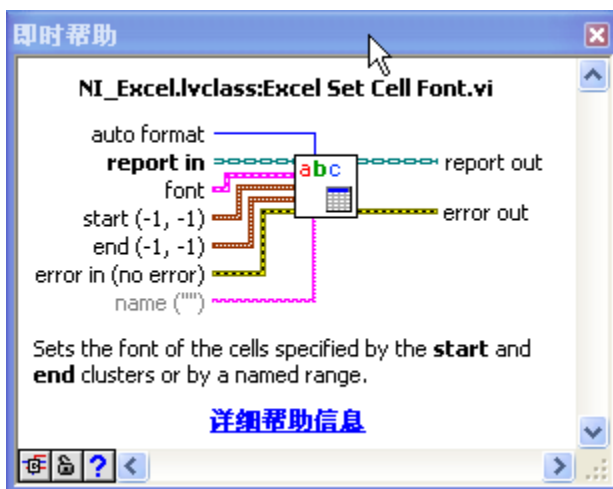


图 3.5 Excel Set Cell Font.vi

我们在图 3.2 的程序的 Excel Easy Text.vi 前面加入 Excel Set Cell Font.vi，即可以控制该单元格的字体了，如图 3.6 所示。

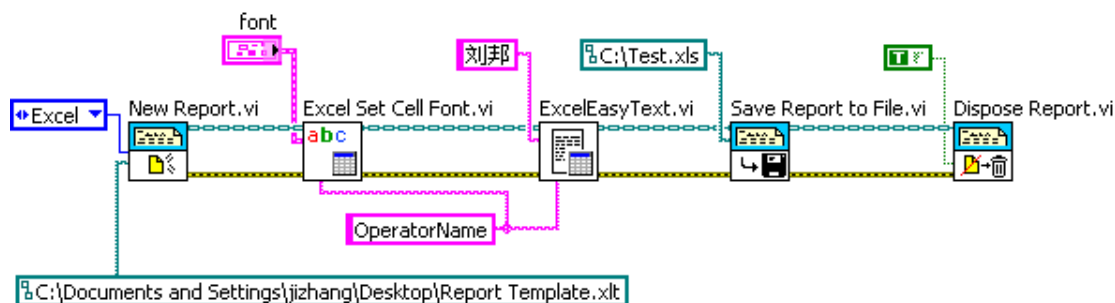


图 3.6 设置单元格字体范例

Excel Easy Text.vi 中的字体设置

每一个刚接触报告生成工具包的工程师都会问同一个问题“为什么 Excel Easy Text.vi 中有 font 参数输入(如所示), 还要另外使用 Excel Set Cell Font.vi 设置字体? 为什么不能直接给 Excel Easy Text.vi 的 font 参数直接赋值来实现字体设置?”

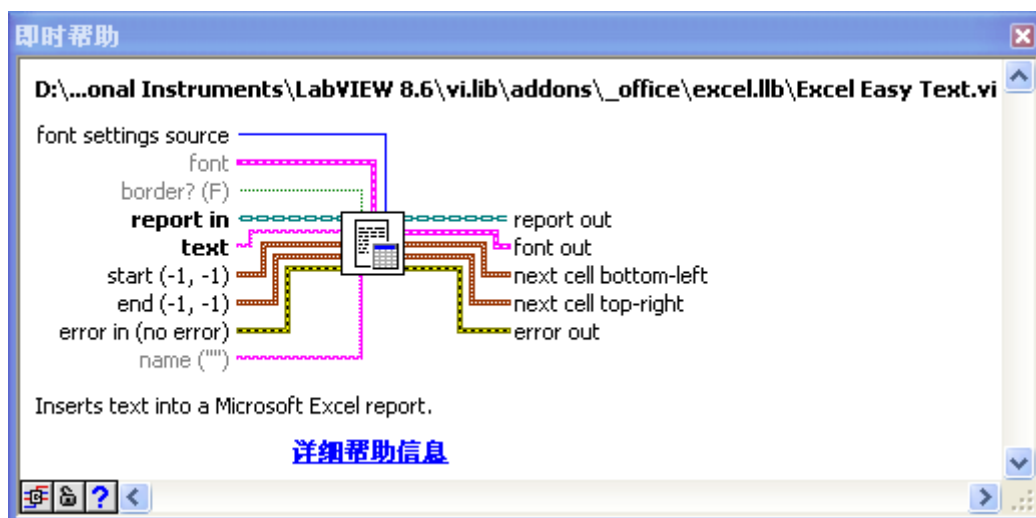


图 3.7 Excel Easy Text.vi 中的 font 参数

这是因为 Excel Easy Text.vi 中的 font 参数不能影响从模板创建的 Excel 文档中的单元格, 只能对新建的 Excel 文档起作用, 如图 3.8 所示。

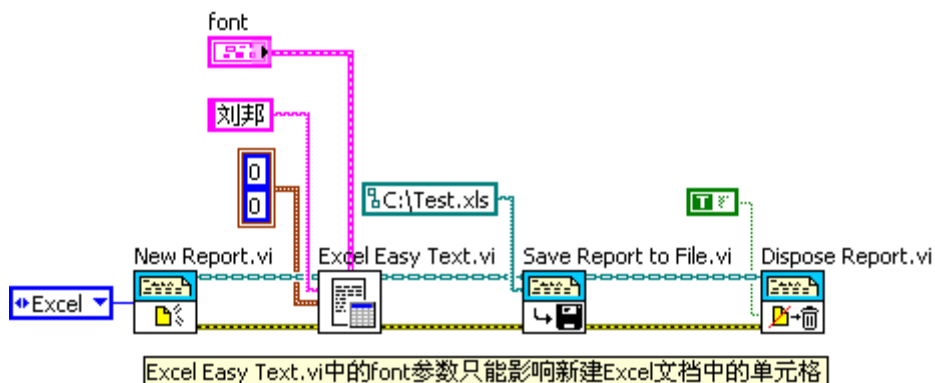


图 3.8 使用 font 参数设置字体格式

设置单元格对齐

设置完字体后，下一个编辑操作就是单元格对齐，同上，借助报告生成工具包中 Set Excel Format 选板下的 Excel Set Cell Alignment.vi 即可完成，如图 3.9 所示。

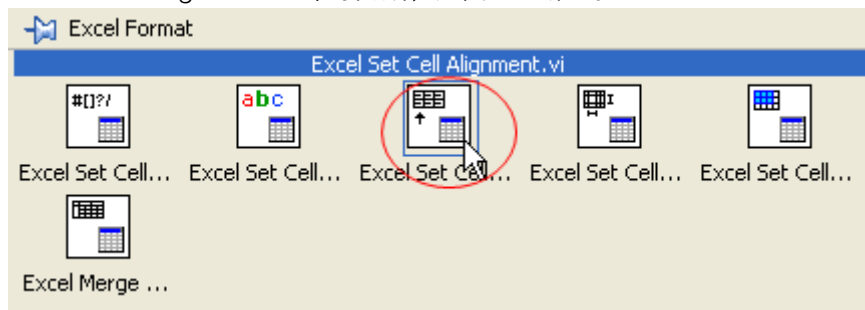


图 3.9 Excel Set Cell Alignment.vi

在使用 Excel Set Cell Alignment.vi 前，应该先把内容插入单元格，然后在对齐，即 Excel Set Cell Alignment.vi 应该在 Excel Easy Text.vi 后面使用，如图 3.10 所示。

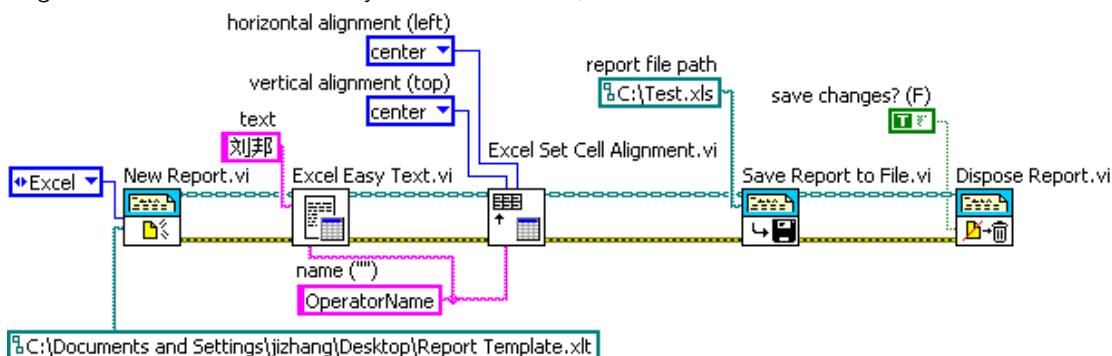


图 3.10 设置单元格对齐

设置单元格边框和背景色

按图索骥，在报告生成工具包中 Set Excel Format 选板下，我们可以找到设置单元格边框和背景色的 VI，Excel Set Cell Color and Border.vi，如图 3.11 所示。

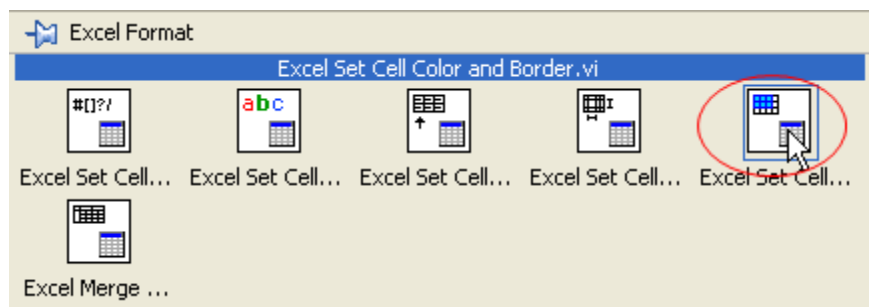


图 3.11 Excel Set Cell Color and Border.vi

使用 Excel Set Cell Color and Border.vi 与 Excel Set Cell Alignment.vi 类似，如图 3.12 所示。

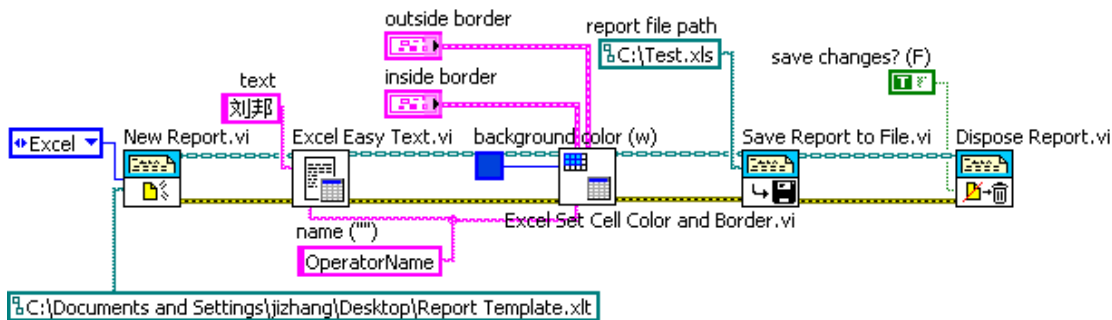


图 3.12 设置单元格背景

精通 LabVIEW 报告生成工具包

通过上面的练习，我们对报告生成工具包的陌生感应该大大降低了，另外，我们也发现，报告生成工具包的使用并不难，它也遵从“打开->操作->关闭”的文件编程模式。

我们还发现，在 LabVIEW 这样一个“所编即所得”的友好的开发环境中，很容易掌握一个全新的功能——只需要借助帮助文档多尝试。这也印证了一个 LabVIEW 大牛的话“学好 LabVIEW 仅仅需要大胆去尝试”。

俗话说“观众器者为良将，观千方者为良医”，精通 LabVIEW 报告生成工具包的不二法门则是看完 LabVIEW 报告生成工具包自带的范例程序，如图 3.13 所示。

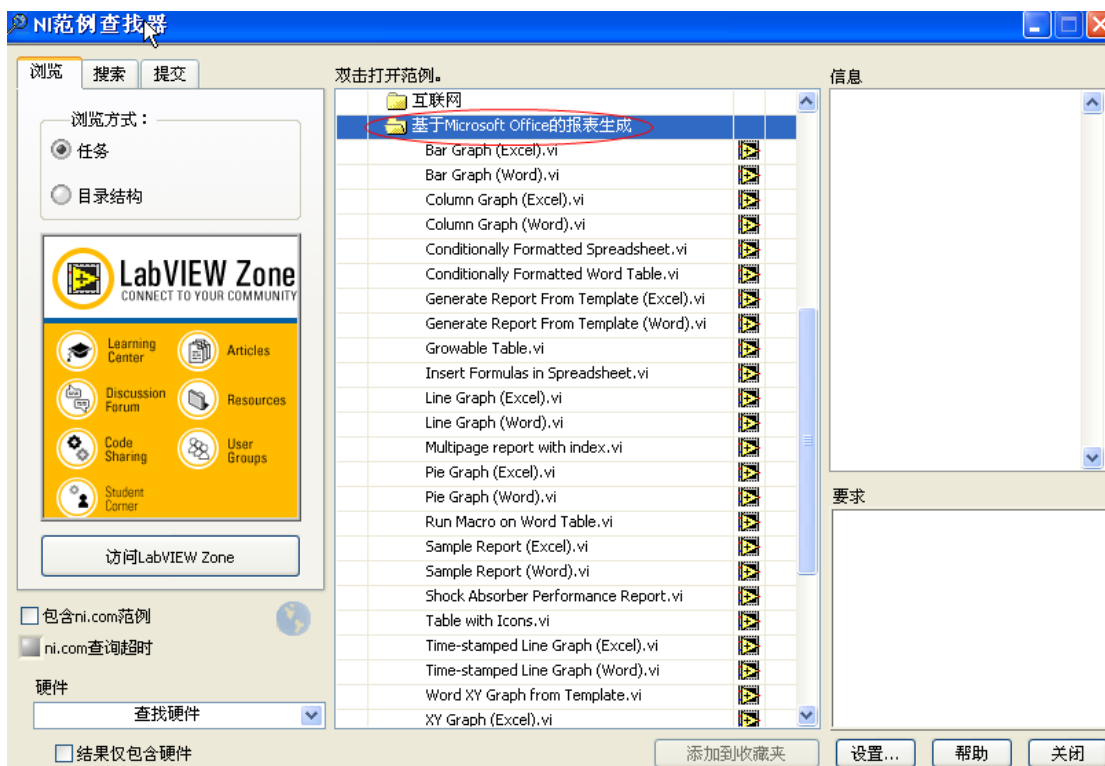


图 3.13 LabVIEW 报告生成工具包经典范例

另外一个值得学习的范例程序就是由 NI 公司方慧敏女士写的范例程序。



图 3.14 报告生成经典范例

后记

"I hear and I forget;
I see and I remember;
I do and I understand."

眼见为实，耳听为虚，任何事都要亲自付以行动才能真真正正的理解。在 LabVIEW 这样一个非常适合工程师和科学家的平台上，多尝试，多实践是工程能力增长的不二法门。

拙作肯定存在不少问题，有任何问题，可以给我 Email: jing.zhang.zju@gmail.com，真心希望能与大家一起分享一起讨论。

边干边学界面本地化



目 录

背景	1
实现	2-5
总结	6

背景

LabVIEW 功能越来越强大，用 LabVIEW 开发的程序也越来越复杂。很多国内的工程师用 LabVIEW 开发的程序也开始冲出亚洲，走向世界。但是为了让 LabVIEW 编写的程序可以远销海内外，大家遇到的一个棘手的问题就是本地化的问题。如何让界面上的文字能够按照我们的需要随时切换呢？

实现本地化的方法有很多。我们姑且分为静态和动态的两种。

所谓静态的实现，其实就是在开发完成之后，根据我们需要发布的语言种类，将源代码直接做本地化。比如我们开发好的 VI 前面板都是英文显示的，需要中文版本的时候，直接在源代码的前面板上把各种文字改成中文的再打包发布。这样做显然费时费力，而且很不灵活。

动态的实现方法，也是很多高级编程语言所使用的方法。这种方法把程序的源代码和界面文字分开。界面文字保存在一个资源文件里面和程序一起发布。所有的界面文字都是从资源文件里面动态读取出来然后显示的。这样做的一个最大好处就是，很容易实现语言的本地化。即使程序已经发布成 EXE 了，也可以通过修改资源文件方便地实现界面语言的本地化。

以前在 LabVIEW 中要实现本地化，大都只能静态的实现。LabVIEW 在生成安装包的时候也可以选择语言，但是那些也是针对一些系统自带的菜单和对话框。

有一种动态实现本地化的方法是用 ini 文件来做资源文件。从而通过 LabVIEW 程序动态读取 ini 文件中事先设定好的各种语言的项目来正确显示对应的文字。这种方法是可行的，但是需要编写大量的 VI 来管理这些 ini 文件中的项目。

许多 LabVIEW 的开发者在为希望能够实现界面上的语言随时按需切换而呕心沥血，煞费苦心。现在，基于 XML 技术的本地化配置编辑器 LCE 出现啦！有了它，我们就可以非常方便地实现随心所欲本地化的梦想。图 1 中是最终程序效果，用左边的下拉列表控件可以实现语言的动态切换。

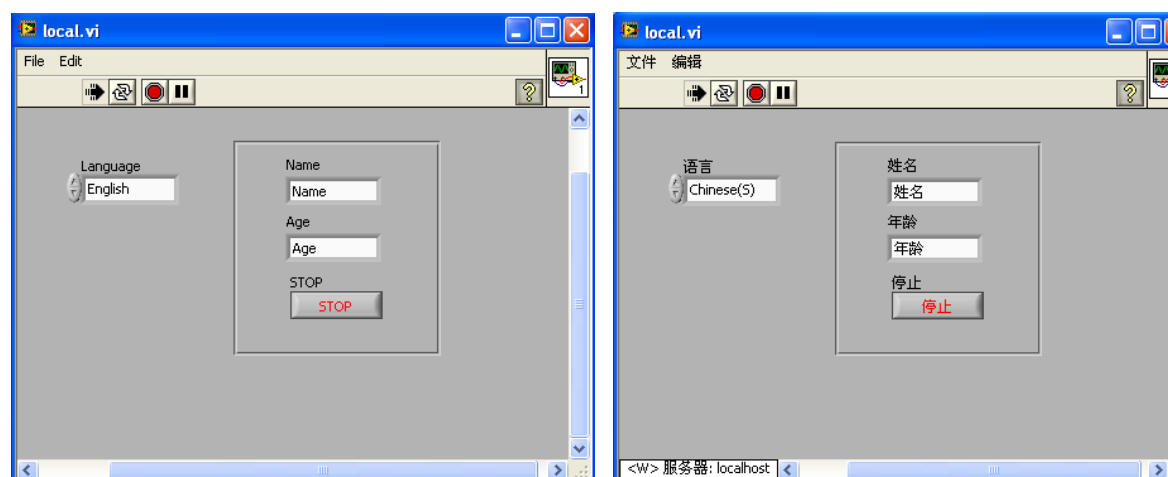


图 1. 最终效果（左）英文界面；（右）中文界面

(LCE 的介绍请参见: <http://zone.ni.com/devzone/cda/epd/p/id/6257>)

图 2 显示了基于 LCE (Localization Configuration Editor)来实现本地化的结构图。LCE 是在 NI 的 Devzone 上新推出的一个可以免费下载使用的工具包。它有两个部分组成:LCE 编辑器和 LCE 函数。

LCE 编辑器: 它其实是用 LabVIEW 2009 开发出来的应用程序。可以创建并修改 XML 的资源文件。资源文件中包括控件和菜单在各种语言下的名称。

LCE 库文件: 在 LabVIEW 中提供 LCE 的一系列 VI 来实现动态调用各种语言下的对应项目的字符。然后通过属性节点赋给各个菜单和控件。

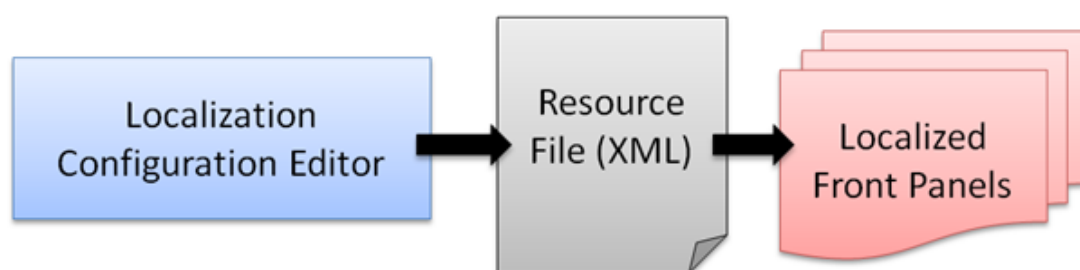


图 2. LCE 实现本地化的结构

安装 LCE

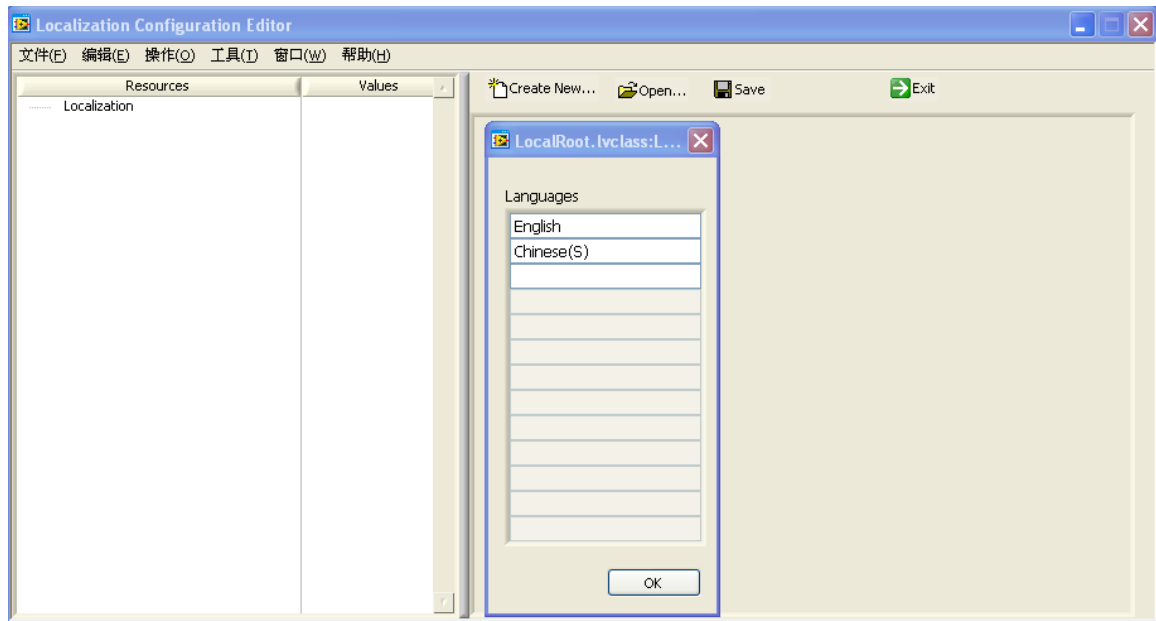
解压并安装 lce_installer_101.zip 即可。由于 LCE 是 LabVIEW2009 编写的, 所以需要 LabVIEW 2009 的 RTE (下载链接: <http://joule.ni.com/nidu/cds/view/p/id/1383/lang/zhs>)。

然后安装 lce_lib_installer_b.zip。这个是调用资源文件的那一系列 VI。这套 VI 支持 LabVIEW 8.6.1 以及以后版本。安装这些文件就会在 LabVIEW 的函数选版中的用户库里面添加 LCE 的 VI。

使用介绍

创建资源文件

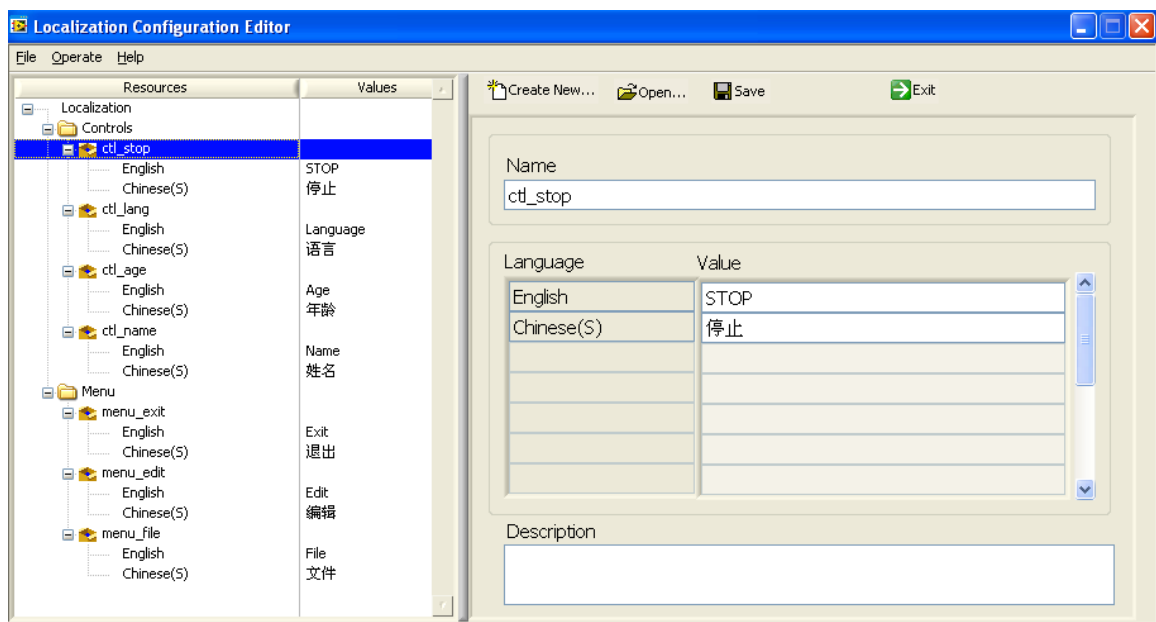
从开始菜单»程序» Localization Configuration Editor v1»LCE 就可以打开 LCE 来创建资源文件了。



首先填入需要的语言

然后在左边的树形结构下右键创建 resource，每个 resource 都有一个唯一的名称，用以以后在程序中调用。每个 resource 下都可以定义每种语言对应的字符串。

按照这个方法创建需要本地化的 resource 列表。也可以把相关的 resource 创建在一个文件夹。



最后点击保存即可。

可以通过 LCE 打开文件夹中 example 目录下的 Localization.xml 来看我已经创建好的例子。

从资源文件获取语言信息

在 LabVIEW 的函数选版»用户库»LCE 里面可以找到 LCE 的几个 VI。主要就是用下面 2 个 VI 就够了。在程序最开始，要通过 Load LCE Resource List.vi 来指定资源文件的路径。

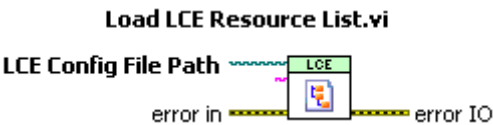


图 3. Load LCE Resource List.vi

然后通过 LCE_Get Resource Value.vi 指定 resource 的名称和语言的种类来得到需要显示的字符串。

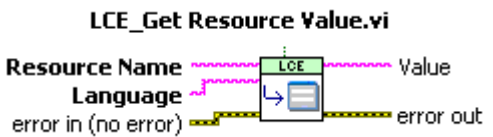


图 4. LCE_Get Resource Value.vi

例程

LCE 提供了一个创建和读取资源文件的方法来管理本地化的语言信息。那么如何通过程序将所有的菜单和控件的名称改变过来呢？为了帮助大家理解，我编写了一个简单的例程。
运行效果

请参考\example 目录下的 local.vi 这个程序。在 VI 运行的过程中，点击 language 下拉框选择不同的语言之后相应的字符都会变成对应的语言。参见图 1。

程序实现

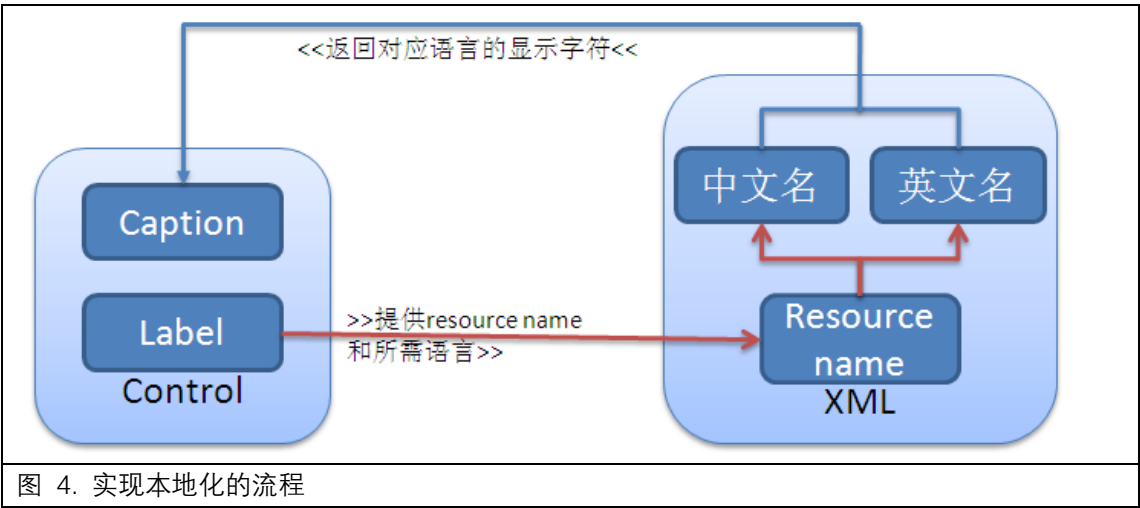


图 4. 实现本地化的流程

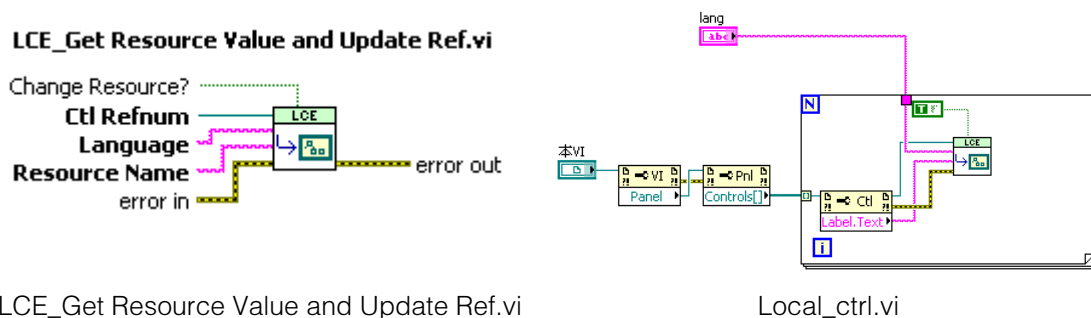
通过事件结构来侦测当下拉框选择不同语言的时候调用 local_ctrl.vi 和 local_menu.vi 来分别对所有的控件和菜单项进行更新。

由于控件的标签(label)是不能在运行时更改的，而标题(caption)是可以的。所以我用 label 作为控件的唯一标识。在面板上让控件都显示出 caption，不要显示 label。改变语言实际上就是改

变控件的 caption。(注：如果控件没有显示过 caption 而去改变 caption 的时候 LabVIEW 是会报错的，所以创建控件的时候务必确保 caption 都已经显示出来了)

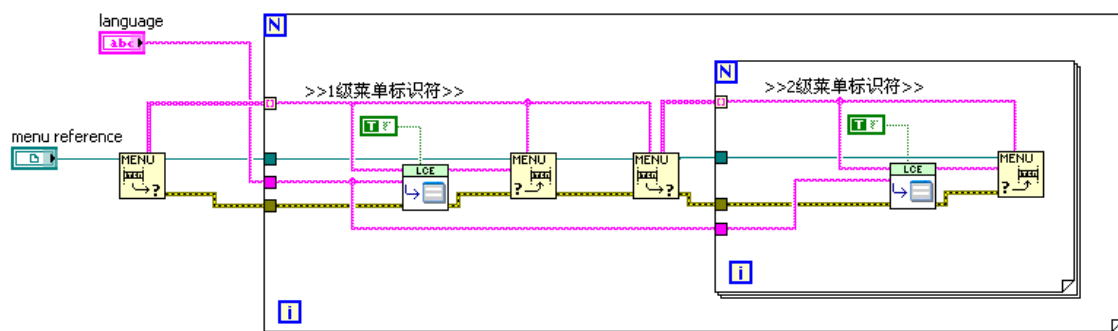
为了将控件和资源文件中的 resource 一一对应起来。我将 resource 的 name 和控件的 label 对应起来。比如 STOP 按钮的 label 叫做 `ctl_stop`，那么在资源文件里面编辑的时候这项 resource 的 name 也叫 `ctl_stop`。在 VI 中就可以通过控件的 `label.text` 属性来查到资源文件中对应的 resource，从而取出需要的语言的显示值。

这个显示值可以通过前面提到的 LCE_Get Resource Value.vi 来得到再用属性节点赋给控件的 Caption 属性。在 LCE 的 VI 里面还有一个叫做 LCE_Get Resource Value and Update Ref.vi。通过它，可以直接把控件引用对应的控件标题自动更改掉。使用起来更加方便。所以在本例程中，我是用的这个方法来的。



菜单的更改是类似的。每个菜单项对应一个菜单标识符和菜单项名称。这两项分别和控件的 label 以及 caption 类似。只要把菜单的项标识符和 resource name 对应。然后用对应的语言字符更新项名称即可。

与控件不同的是，菜单项的访问是有一系列 VI 的，而不是通过属性节点。而且菜单是有层次的。每次读取菜单项都只能读取到一层的。我在这里在例程中只编写了两层的菜单。所以要遍历所有的菜单项可能需要迭代来实现。



总结

本文通过一个实例来介绍了基于资源文件来实现 LabVIEW 前面板本地化的方法。通过 LCE 这个工具可以方便地实现资源文件的创建、编辑和读取。在使用这个工具包的时候，也发现有几个需要注意的地方。

创建和编辑资源文件的 LCE 程序需要 LabVIEW 2009 运行库引擎。这个给该方法的使用造成了一定的限制。不过网上有 LCE 源代码，也许可以重写到 LabVIEW 8.x 的版本从而可以使其适用面更广。

LCE 的高级 VI 是可以直接把连在控件引用上的控件的 caption 自动改变的功能。这点固然方便，但是有时候也改变了一些不必要的东西。比如，例程中文本输入控件的 caption 和内容都被改变了。所以在实际使用的时候，这个 VI 也许需要根据具体情况做相应的修改。

目前发现 LCE 如果给定的 resource name 不存在也会返回一个其它 resource name 的名称。这一点应该算个 bug。因为这样如果 前面出现控件的 label 没有被定义的情况，其名字可能被改为和其它的控件一样。所以使用的时候需要这点。

如果大家在使用的时候有什么心得或者发现本文有什么问题，欢迎和我联系！

韩翼

Yi.han@ni.com