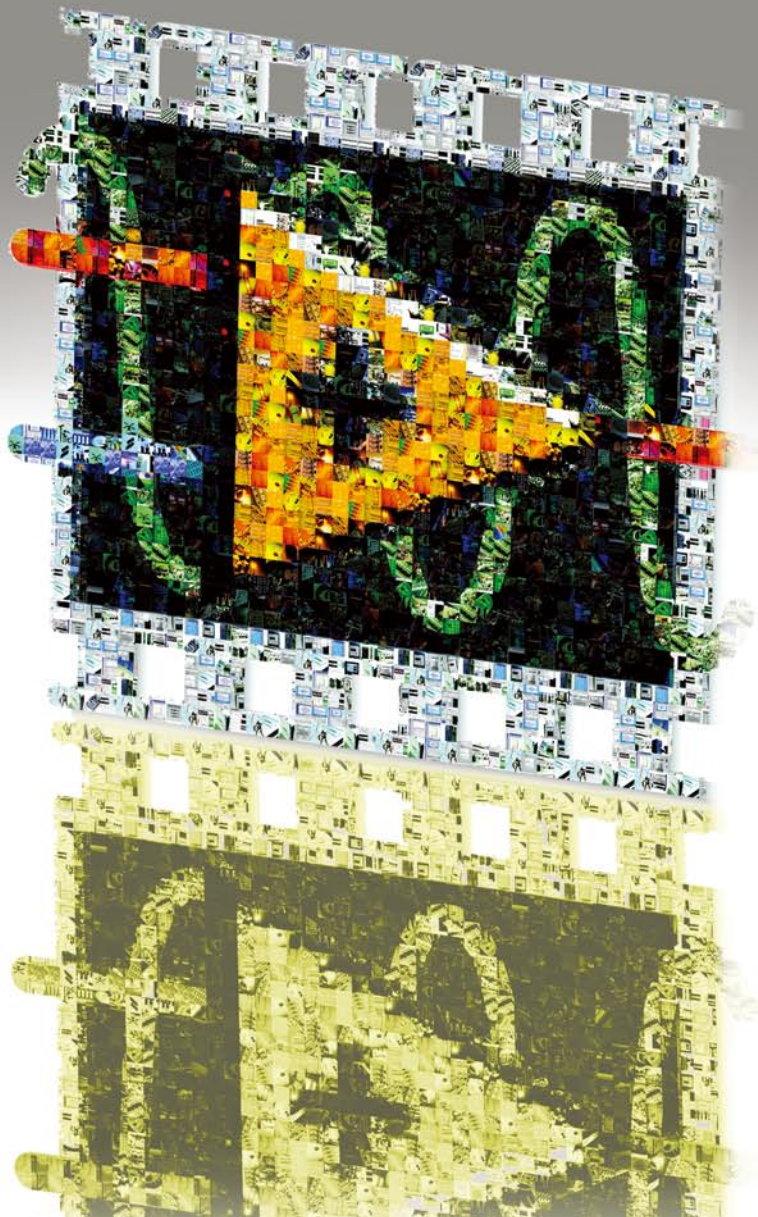


LabVIEW 学习札记——第一卷 上



目 录

第1章 基于计算机的测试和测量

| | |
|----------------|----|
| 第1.1节 测试和测量 | 1 |
| 第1.2节 测量技术的发展 | 4 |
| 第1.3节 基于计算机的测量 | 8 |
| 第1.4节 虚拟仪器 | 10 |
| 第1.5节 虚拟仪器技术 | 15 |
| 第1.6节 本章小结 | 19 |

第2章 我眼中的LabVIEW

| | |
|------------------------|----|
| 第2.1节 LabVIEW到底魅力何在? | 21 |
| 第2.2节 什么是LabVIEW? | 23 |
| 第2.3节 LabVIEW图形化编程语言 | 28 |
| 第2.4节 学习LabVIEW应该看的几本书 | 34 |
| 第2.5节 LabVIEW版本的变迁 | 40 |
| 第2.6节 学习LabVIEW的理由 | 43 |
| 第2.7节 LabVIEW的受众面 | 49 |
| 第2.8节 虚拟仪器开发模式之我见 | 53 |
| 第2.9章 本章小结 | 55 |

第 1 章 基于计算机的测试和测量

第 1.1 节 测试和测量



科学的发展和技术进步，是建立在人类认识客观世界能力提高的基础上，测量与测试就是人们描述对客观事物认识过程的两个基本术语。

测试和测量在使用上还是有一些不同。

1.1.1 测试和测量的基本概念

测量

测量在人类对客观世界进行认识的试验、研究活动中，更强调对测量结果进行“定量”的描述。这种测量活动往往是将被测量与同类已知量进行比较的一个过程，或者说它是有一定的行为规范来约束的。或者说：测量是规范化了的测试过程。测量大多针对的都是物理量（化学量通常使用分析）。

比如：我们通过体温计测量体温来断定某人是否发烧，这就是一个定量的测量过程，因为体温计给出了测量结果的量值，通过测量结果的量值来确定发烧的程度。在“非典”时期，为了排查大量流动人群中可能患“非典”的病人，采用的是使用辐射测温计来进行排查，辐射测温计是无法给出准确的体温数值的，但是它可以准确的分辨出体温有一定差别的人，然后再使用体温计来进一步排查，体温接近于 38 度的，就可能是“非典”疑似病人，需要到医院进一步进行观察。

由于我们单位是依法计量、检定部门，所以通常使用测量这个词可能是更准确、更规范些，通常要避免使用测试这个词。

JJF1001-1998 规范中的定义

“测量——以确定量值为目的的一组操作。”

在该规范中，没有对“测试”做以任何说明解释。

测试

相对于测量，测试通常指：试验研究性的一组测量过程。测试往往是更强调“定性”的作用，或者即包含“定性”又包含“定量”的过程。

测试是测量和试验的简称。

试验是在真实的情况下，或模拟的条件下对被测对象的特征进行测量和度量的研究过程。

测试的概念含义很广泛，一般指生产和科学试验中经常进行的满足一定准确度要求的试验性测量过程。

对一些抽象概念的评价通常使用“测试”这个词，因为抽象的概念很难量化。比如说：心理测试、性格测试等等，此时如果使用“测量”这个词，就不对了。

比如：科研成果鉴定时，鉴定会测试组的任务就是采用一组满足一定准确度要求的试验性测量给出科研成果的正确评估（即定性又定量，没有现成的规范可采用，所以称为“测试组”）。

家里突然停电了，我们使用“试电笔”就可以断定是否是保险丝熔断了。当然，使用手持万用表也可以查出问题所在，但是此时我们并不关心万用表显示的是否是 220V，关注的仅是有没有电。

示波器往往也是被用来察看是否有信号、信号的形状的大致等等。

需要说明的一点是：本《札记》中所涉及的都是基本物理量的测量和分析，不涉及化学量的测量和分析。

1.1.2 测试、测量结果的表述方法

[测量结果的] 精密度 precision [of results of measurements]



基本定义：在规定条件下获得的各个独立观测值之间的一致程度。

精密度仅指由于随机效应使测量结果不能完全重复或复现，表现的是随机误差的大小。

我个人理解：精密度主要用来评定系统的短期或长期稳定性。

比如：交流稳压电源或直流稳压电源它们输出的数值可能不准确，但是很稳定（值的波动量不大），具体讲：一个输出为 5V 的直流稳压电源，其输出电压值可能为 5.1V，但是在 8 小时内其波动值

仅为 10 微伏，所以我们说它是性能非常好的稳压电源。有时这种特性的电源又被称为：精密直流（交流）电源。

值得注意的是：千万不要使用“精密度”来表示“精确度”或“精度”。由于“精确度”或“精度”这两个词容易与“精密度”相混，已经回避使用，取代它们的是：准确度。

在这个概念上经常容易出错的现象是：把 NI 24 位的 DSA 数据采集卡称为：24 位精度（精确度）。显然这是一个概念上的错误，错将 DMM（数字万用表）的“位数”概念用到了这里。而一般 24 位 DSA 的准确度较低，精密度较高。使用 24 位的转换位数是为了获得更大的动态范围。关于动态范围及 DSA 的基本概念，将在数据采集单元详细说明。

图示说明：红色圆点表示测量的数据点，中心的小圆表示测量的约定真值点。图示的测量点说明测量的精密度好，正确度不好。

[测量结果的] 正确度 correctness [of results of measurements]



基本定义：测量结果与真值的接近程度。它反映的是测量结果的系统误差的大小。
该术语基本上也不使用，这里借此为表明一些基本概念。
图示说明：图示的测量点说明测量的精密度不好，正确度好。

[测量结果的] 准确度 accuracy [of results of measurements]



基本定义：测量结果与被测量真值之间的一致程度。

它是精密度和正确度的综合反映。

因为准确度是描述测量结果与被测量真值之间的一致程度，所以通常用准确度等级表示接近的程度。

真值——就是被测量的实际值。实际上，受测量方法、测量能力的限制，我们无法测量到被测量的真值。

替代实际真值的称为：约定真值。这里我的理解就是：由于实际需要可以满足测量要求，就用约定真值来代替实际真值，避免提高测量成本，所以用不同的准确度等级来满足不同的测量需求。比如：体温计上的刻度就是约定真值，因为体温 37.5 度和 37.55 度对医生确诊来讲都是一样的，可是实现二者测量所花费的代价绝对是不同的。

图示的测量点说明测量的精密度好，正确度好，准确度亦好。

[测量] 误差 error [of measurement]

基本定义：测量结果减去被测量的真值。

由于真值不能确定，实际上用的是约定真值。定义给出的是绝对误差。

一个测量结果的误差，若不是正值（正误差）就是负值（负误差），它取决于这个结果是大于还是小于真值。

比如：信号发生器输出频率刻度显示的是 10Hz，而使用数字频率计实测的频率值为：9.98Hz，则误差就是：- 0.02Hz，表示信号发生器的实际输出频率值比约定真值小 0.02Hz。

第 1.2 节 测量技术的发展

1.2.1 利用物质本身的基本特性直接测量

指南针



指南针是用以判别方位的一种简单仪器。指南针的前身是中国古代四大发明之一的司南。主要组成部分是一根装在轴上可以自由转动的磁针。磁针在地磁场作用下能保持在磁子午线的切线方向上。磁针的北极指向地理的南极，利用这一性能可以辨别方向。常用于航海、大地测量、旅行及军事等方面。

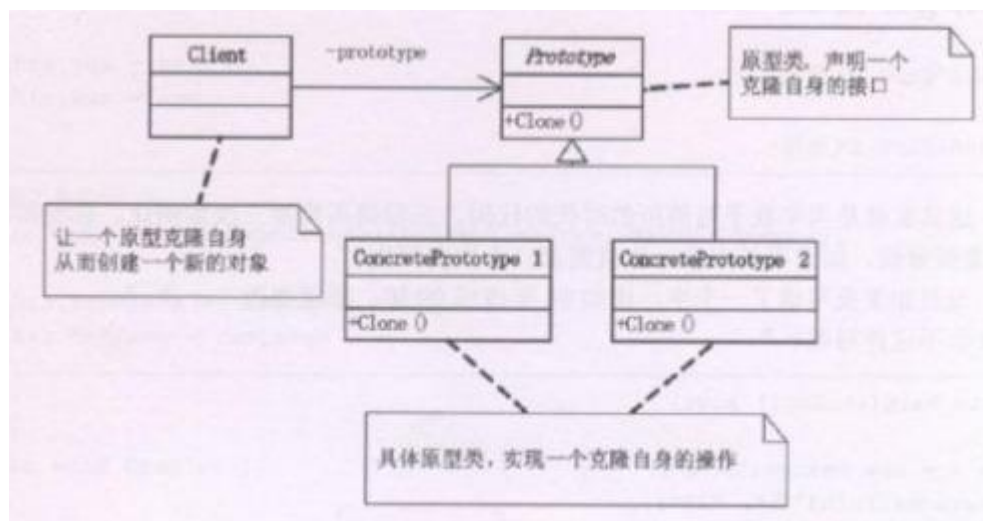
指南针的发明是我国劳动人民，在长期的实践中对物体磁性认识的结果。由于生产劳动，人们接触了磁铁矿，开始了对磁性质的了解。人们首先发现了磁石引铁的性质。后来又发现了磁石的指向性。经过多方的实验和研究，终于发明了可以实用的指南针。

弹簧秤



弹簧具有受力后产生与外力相应的变形的特性。根据虎克定律，弹簧在弹性极限内的变形量与所受力的大小成正比。称重时，弹簧变形所产生的弹性力与被测物的重量(重力)相平衡，故从变形量的大小即可测得被测物的重量,进而确定其质量。

温度计



根据使用目的的不同，已设计制造出多种温度计。其设计的依据有：利用固体、液体、气体受温度的影响而热胀冷缩的现象；在定容条件下，气体（或蒸气压强因不同温度而变化；热辐射的影响）等。

一般地说，任何物质的任一物理属性，只要它随温度的改变而发生单调的、显著的变化，都可用标志温度而制成温度计。

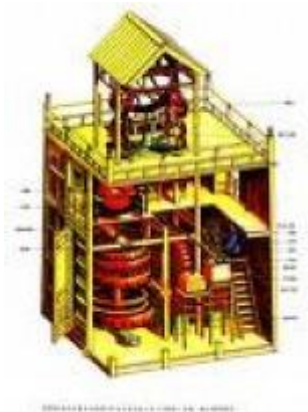
1.2.2 利用物理现象间接测量

日晷计时器



日晷是以太阳影子移动，对应于晷面上的刻度来计时。日晷不用说了，大家应该在北京故宫里和观象台上见过。

滴漏计时器



漏是以滴水为计时，是由四只盛水的铜壶组合，从上而下互相迭放。上三只底下有小孔，最下一只竖放一个箭形浮标，随滴水而水面升高，壶身上有刻度，以为计时。原一昼夜分 100 刻，因不能与十二个时辰整除，又先后改为 96，108，120 刻，到清代正式定为 96 刻；就这样，一个时辰等于八刻。一刻又分成三分，一昼夜共有二十四分，与二十四个节气相对。

1.2.3 电测量技术的发展

热电特性——热电偶



热电偶的测温原理是基于热电效应。将两种不同的导体或半导体连接成闭合回路,当两个接点处的温度不同时,回路中将产生热电势,这种现象称为热电效应,又称为塞贝克效应。闭合回路中产生的热电势有两种电势组成:温差电势和接触电势。温差电势是指同一导体的两端因温度不同而产生的电势,不同的导体具有不同的电子密度,所以他们产生的电势也不相同,而接触电势顾名思义就是指两种不同的导体相接触时,因为他们的电子密度不同所以产生一定的电子扩散,当他们达到一定的平衡后所形成的电势,接触电势的大小取决于两种不同导体的材料性质以及他们接触点的温度。

电磁特性——指示仪表



电磁、磁电系仪表 利用永久磁铁使载流线圈偏转的仪表。它主要用于测量稳恒电流和电压,具有灵敏度高、精度高、功率消耗小、刻度均匀等优点。准确度可达 0.1%。如配以整流元件或变换器,它还可以用于交流电参量和非电参量的测量,是一种使用范围很广的仪表。

数字式仪表



应用数字和模拟电子线路实现电学量的测量,并以数字显示测量结果的电工仪表。数字仪表是随电子技术的进步而发展起来的。第一台数字电压表于 1952 年问世,采用电子管电路控制继电器工作。后来,数字仪表又采用半导体电路。70 年代以来随着集成电路的出现,较简单的数字式面板表、小型多用表中只用几块集成电路芯片。80 年代已出现具有很高计量性能的微机化数字电表。

智能化仪表



微电子技术和计算机技术的不断发展,引起了仪表结构的根本性变革,以微型计算机(单片机)为主体,将计算机技术和检测技术有机结合,组成新一代“智能化仪表”,在测量过程自动化、测量数据处理及功能多样化方面与传统仪表的常规测量电路相比较,取得了巨大进展。智能仪表

不仅能解决传统仪表不易或不能解决的问题，还能简化仪表电路，提高仪表的可靠性，更容易实现高精度、高性能、多功能的目的。

这里需要注意的是：当数字式仪表和智能化仪表出现后，测量原理和测量方法也发生了一些改变，从过去的传统的纯模拟式测量方式转化到模拟－数字化的测量方式。以后还要进一步来说明这个差别的实质（留到数据采集部分来表述）。但它也不同于下面将要介绍的——基于计算机的测量方式。

第 1.3 节 基于计算机的测量

1.3.1 微处理器和计算机

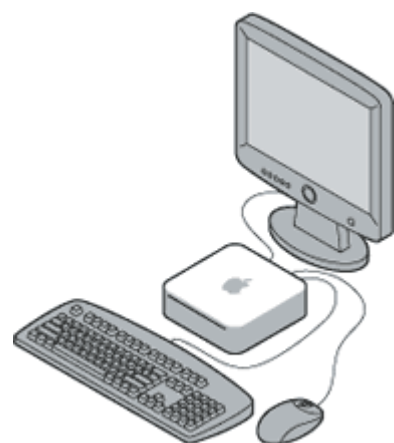


内置微处理器的测量系统一般被称为：嵌入式系统。就象上一节中提到的智能仪表就属于这一类。

也有些精密测量仪器将性能先进的计算机完全置入仪器内部（包括 CPU、硬盘、存储器、LED 显示器等等），这里的计算机仅作为内部测量控制、分析使用，不象用户提供任何计算机功能对外使用，我也将这类仪器列入嵌入式系统中。

我们这里所涉及到的基于计算机的测量是指：商用计算机或者是相类似的专用计算机。

1.3.2 基于计算机的测量



实现基于计算机的测量是微电子技术和计算机科学技术的不断进步的结果，具体体现在一下几个方面：

1、微电子技术发展的非速发展

模数 – 数模转换技术成熟，高性能、低功耗、多通道、高位数、高采样率芯片的出现。包括 FPGA 技术、DSP 技术的发展。

比如：我最早使用的单片 A/D 转换器是 ADC0809，单片 D/A 转换器是：DAC0832。

2、数字信号处理技术发展的结果

若想实现计算机的测量和分析，离散数字信号处理技术必须不断完善和成熟，这样才能充分发挥计算机的作用。

3、计算机本身进步的结果

CPU 的多核技术、大容量内存、海量硬盘存储器、先进的操作系统和高效的应用软件。灵活的接口方式，都给计算机测量的应用带来巨大的好处。

4、计算机的性价比越来越高

计算机在性能不断提高的同时，而价格确不断下降。

5、先进的基于计算机测量的软件开发环境

提供了专业的基于计算机测量、分析的应用软件开发环境日趋成熟。

在基于计算机的测量应用中，目前采用的计算机基本上是 PC、工控机、CompactPC、PXI、VXI。提到基于计算机的测量，就不可能不提到美国的 NI 公司。

1.3.3 致力于计算机测量的美国 NI 公司



美国 NI 公司在基于计算机测量方面创造了这一领域绝对性的辉煌成就，在这个世界上无人能比。它不仅提供基于计算机测量的高性能硬件产品，还创造性的开发出基于计算机测量的应用软件开发平台及“虚拟仪器的概念”，并扩展为：“软件就是仪器”的概念。

二十多年来，他们不断改进、发展了虚拟仪器技术，使之成为事实上的工业标准。使越来越多的用户和使用者成为这项技术的受益者。

我们这个世界无论对动物还是对“产物”来讲：都是一个“适者”生存的世界。而电子产品有着比其它产品更快的淘汰率和更新率。象红及一时的“数字 BP 机”、“汉字 BP 机”、“模拟手机”“都在市场竞争中渐渐地退出了历史舞台。

虚拟仪器技术不但没有衰退的迹象，反而得到更高速的发展和广泛的应用。



2001 年，美国总统小布什在白宫接见 NI 的 CEO。

第 1.4 节 虚拟仪器

我们知道：虚拟仪器的概念是美国 NI 公司首先提出来的。

1.4.1 虚拟仪器基本概念



首先我们先看一个非常实际的例子：

对于整天与 PC（个人计算机）打交道的人来说，在紧张、繁忙的工作一段时间后，通常会选择听一段轻松的音乐或观看一部 DVD 大片，其目的就是来缓解一下因紧张工作带来的压力和烦躁（但此刻要注意主管或太太的心情是否很好）。因为商业化 PC 完全具备了这样的能力，我们为何不借此享受一番呢！我相信，绝大多数经常与 PC 打过交道的人，都曾会有过这样类似的经历。



仔细地想一想：能够获得这样悠闲地享受，我们所做的只不过是 PC 的 DVDROM 中放入了一张 CD 或 DVD 光盘，PC 就会为我们提供视觉和听觉上的享受（为了更好的享受，我准备了一台 17 寸的 iMac Intel，和一对 BOSE 的有源小音箱从而达到了“发烧初级”的效果）。

事实上，之所以能够得到这样的享受是我们利用了 PC 的某些硬件资源和媒体播放软件资源“虚拟”了一部 CD 或 DVD 播放机。还不只这些，如果在 CDROM 中放入不同的游戏光盘我们就可以玩电子游戏，这时 PC 又为我们“虚拟”了一部游戏机（效果可能不如 Xbox 或 PS3，但也无需再次投资）。其实在 PC 机上可“虚拟”东西还很多，比如：“虚拟内存”、“虚拟光驱”等等。怎么又是“虚拟”的呀！是的，“虚拟播放机”、“虚拟游戏机”、“虚拟内存”、“虚拟光驱”与我们所讨论的“虚拟仪器”，从基本概念上看，“虚拟”的含义应该是一样的，只不过是实现的功能不同而已。通过这样简单描述，我相信：对于什么是“虚拟仪器”我们应该有了一个基本上的共识。

所谓“虚拟仪器”，就是在通用计算机上，利用通用接口总线连接硬件数据采集或控制模块，通过软件编程控制硬件模块进行控制或测量，并利用软件实现仪器的测量和分析功能。

其实，“虚拟仪器”这个概念从字面上来简单理解就是：借助于计算机和数据采集模块通过软件设计，能够实现真实仪器的测量功能，但确不是一个实实在在的、有模有样的真实的测量仪器。这话看着似乎别扭，没错，但是这样的表述可能更清晰。

当虚拟仪器出现后，为了区别于虚拟仪器，人们将过去所使用的仪器（示波器、万用表、信号源等）称为：传统仪器。

1.4.2 虚拟仪器构成的三要素



——商业化的 PC 及操作系统（比如：上面所说的 PC 机与操作系统）

商业化的 PC 是指：主流的品牌机或性能较好的组装机、笔记本电脑
操作系统是指：Windows、Linux、Mac OS 等。

——应用软件（比如：上面所说的媒体播放软件）

应用软件是指：能够操作、控制 PC 硬件，实现“虚拟仪器”功能的各种应用软件包。

——能够实现虚拟仪器功能的硬件（比如：上面所说的声卡）

能够实现虚拟仪器功能的硬件是指：一些可实现仪器功能的特殊硬件模块（包括驱动程序在内）。

用声卡来描述能够实现虚拟仪器功能的硬件似乎有些不妥当，但初学者（指初学虚拟仪器技术的人）在没有数据采集模块硬件时完全可以利用声卡来学习和实现一些基本数据采集设计。我曾经看到某大学的老师在指导学生毕业设计时，就是利用声卡来作数据采集试验。利用 PC 上的硬件，不花一分钱，就指导了学生的毕业设计。学生该得到的也都得到了，老师也达到了教学目的，可以说这是多么富有想像力的事情啊！

言归正传，简单的表示就是：虚拟仪器 = 商业化的 PC + 虚拟仪器应用软件 + 数据采集硬件。

需要进一步说明的是：“虚拟仪器”通常是相对于仪器的使用者而言的，它包含可操控的仪器界面和相关硬件配置。这些通常是由系统集成商或专业开发人员设计完成，使用者并不关心它的设计方法，所关心的仅仅是如何使用和在何处使用。

大概了解了什么是虚拟仪器，那它到底会给我们带来什么样的好处呢？下面我们就来谈谈虚拟仪器的特点。

1.4.3 虚拟仪器的特点

虚拟仪器到底有什么特点呢？

1、强大的运行环境

根据虚拟仪器的构成，我们知道虚拟仪器通常是运行在各种商业化计算机上，而商业化计算机在多核技术、运行速度、带宽、存储能力、数据处理能力、网络 LAN 及可利用的各种总线技术 USB、1394 等方面，是任何嵌入式系统所无法比拟的。所以我们说虚拟仪器的运行引擎是最强大的、运行环境是最好的。

注意：尽管运行环境强大，但在实时性还是不如嵌入式系统（解决这个问题可以使用 LabVIEW RT 或使用 cRIO 平台等）。

从虚拟仪器诞生到现在，计算机技术飞速发展也可以说带动或者说是推动了虚拟仪器应用的发展。因为计算机只是虚拟仪器的运行平台，同时它又是由第三方来制造的，所以我们大可不必关心如何提高它的技术性能，这些统统都由计算机设计者和制造商来考虑。面对不断涌现出的多线程技术、多核技术、PCIe 技术以及操作系统的进步，我们无须考虑更多，只需尽情享受和利用它们为我们提供的高性能产品。

同时商业化计算机有着丰富的第三方提供的软件资源如：Word、Excel、MatLab 等等，虚拟仪器也可以充分地利用这些有效的资源。

2、自定义测量功能和多功能性

商品化的测量仪器往往其测量功能较为单一，并且测量功能往往是由制造商定义好的。所以，人们习惯上是以类来区别的它们的测量功能。比如：DMM、示波器、信号发生器等，它们相互间在测量功能上也是不兼容的。比如，使用 DMM 无法产生模拟信号、用信号发生器不可能查看到波形，但虚拟仪器可以完全做到这一点，虚拟仪器的最大的特点就在于：用户可以自定义测

量功能。

在我曾做过一个项目《电动式振动台检定系统》中，就充分地发挥了虚拟仪器的自定义的多功能特性。通常“电动式振动台”检定都是在现场进行的，在此之前，我们去现场检定需要携带 6 件仪器（示波器、电荷放大器、失真度测量仪、动态信号分析仪、数字式频率计、记录仪等还没算上携带的传感器）至少得安排两个人和一辆车。而使用虚拟仪器技术开发的《电动式振动台检定系统》，只需带一个 NI USB 9233（NI USB 9234）数据采集卡（巴掌大小）和几支传感器加上一个笔记本电脑。只需把待测信号通过数据采集卡采集到计算机上，其余所有分析、计算功能就都是通过应用软件来实现。

3、强大的分析处理能力

我们知道虚拟仪器是运行在计算机上，而计算机的数据处理能力及 LabVIEW 软件开发平台提供的强大的分析能力能够解决商品化仪器所做不到的事情。

比如：即便是较好失真度测量仪在低频段也只能测量到 10Hz 频率的失真度，而电动式振动台检定需要检测低至 5Hz 的加速度失真度值，商品仪器可能也买不到，即便是可以买到那么价格和使用的频度都可能是我们承受不起。但利用虚拟仪器技术开发的《电动式振动台检定系统》借助于 LabVIEW 的强大分析功能就可以很容易的解决这个问题。

还有一个应用实例，我们知道：使用数字式频率计来测量频率时，对输入信号的失真度是有一定的要求的。当信号中含有较大的谐波时测试可能是不准确的，道理就是谐波会影响数字式频率计的过零触发的时间点。而电动式振动台检定时希望能够在失真度高达 25% 的情况下测量振动频率值，数字式频率计显然无法做到这一点，但利用虚拟仪器技术开发的《电动式振动台检定系统》借助于 LabVIEW 的强大分析、处理功能解决了这个问题。

4、“再生（复用）”性能

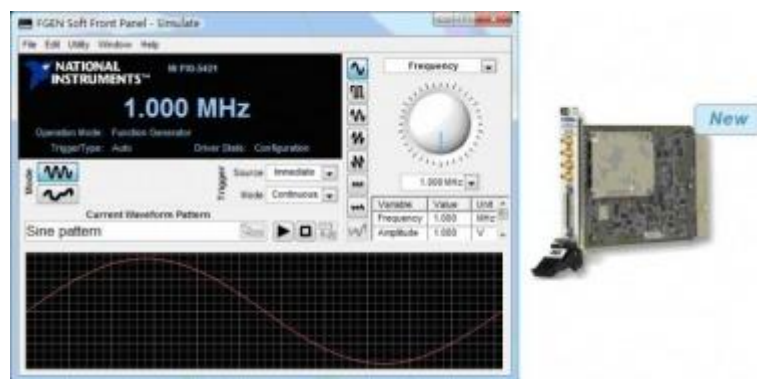
我们知道：“软件就是仪器”是虚拟仪器的最大特点。也就是说：在硬件条件基本不变的情况下，我们可以根据测量需求，通过软件设计可以实现多种方式不同的测量仪器。举个简单的例子：用一块 M 系列数据采集卡，可以实现交流信号的采集、记录、分析；也可以实现直流信号的采集、记录、分析；接入温度变换器还可以测量温度；接入压力变换器还可以测量压力等等等等，我们只需要设计不同测量分析功能的应用软件。从这个特点上看：虚拟仪器硬件具有“再生（复用）”功能。

从这个意义上来讲：虚拟仪器的硬件使用率可以是非常高的。实质上，也就是提高了仪器的使用率。比如：商品示波器，再不需要使用的时候只能将其束之高阁。而虚拟仪器设计的示波器，不使用时，只是应用软件不使用，而硬件还可以设计成其它的仪器使用。这对于实验室、中小企业、教学单位来讲是意义重大的，可以大幅度降低通用设备的投资。

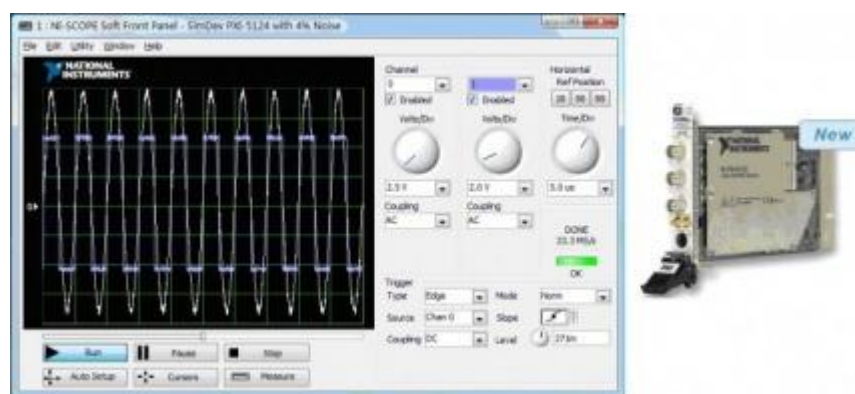
1.4.4 来自 NI 公司的几个虚拟仪器实例



DMM 虚拟仪器前面版和 DMM 卡



信号发生器前面版和板卡



虚拟仪器示波器及板卡

看看这些精美的应用实例,如果我们也希望利用虚拟仪器的形式来满足自定义的测试或控制需求时,或者希望成为这方面的程序设计者时,如何才能达到这个目的呢?那就要涉及到“什么是虚拟仪器技术”了。

第 1.5 节 虚拟仪器技术

我们知道：不仅虚拟仪器的概念是美国 NI 公司提出来的，美国 NI 公司还创造、发明了一整套系统级的虚拟仪器软、硬件开发系统（环境）。

1.5.1 虚拟仪器技术概述



“虚拟仪器”和“虚拟仪器技术”是两个根本不同的概念。“虚拟仪器”往往是偏重于应用性或者说实用性更突出，表现形式通常是虚拟仪器应用的实例。而“虚拟仪器技术”是指虚拟仪器的制造方法和制造平台。这包括一个开放的、灵活的虚拟仪器软件开发环境及与软件平台相适应的硬件体系。

通俗地讲：“虚拟仪器”应该是“虚拟仪器技术”的产物或结果。

事实上，虚拟仪器创始人——美国国家仪器（NI）公司也同样是一个“虚拟仪器技术”的创造者，并在 20 多年来不断完善、提高这项技术。

NI 公司也是全球唯一的一个即可提供虚拟仪器技术软件开发平台，又同时提供专业硬件产品支持的专业制造商。这种软、硬件都来自同一个公司的好处是：软件和硬件之间是无缝链接的，并可以充分发挥和利用软、硬件的全部技术特性。当然，最大的受益者应该还是它的使用者。

“虚拟仪器技术”包括以下几个主要部分：

1.5. 1.1 虚拟仪器应用软件开发环境



NI 公司的 Developer Suite（开发者套件）是一个开放的、灵活的、适应多种需求的虚拟仪器软件开发平台。

它包括：

——LabVIEW 8.5.1：是 NI 公司的旗舰产品，强大、开放、图形化的虚拟仪器软件开发环境。由于 LabVIEW 采用的是图形化的编程方法，所以无论是否有过编程经验的工程师或科学家使用时都可以快速、高效地与测量和控制硬件通信，并进行数据分析及处理。

——LabWindows/CVI 8.5：是一种久经验证的 ANSI C 集成开发环境，为习惯使用 C 语言的工程师和科学家提供用于创建测试和控制应用的全套编程工具。NI LabWindows/CVI 兼有 ANSI C 的耐用性、复用性和特定的工程性能，并适用于仪器控制、数据采集、分析和用户界面的开发。

——Measurement Studio 8.1.2：是一个专为 Visual Studio 2005、Visual Studio .NET 2003 和 Visual Studio 6.0 使用者设计的集成式套件，它包括用于测试、测量和自动化等应用的大量函数类和控件。Measurement Studio 提供专为工程师们所设计的 .NET 和 ActiveX 用户界面控件、先进的科学分析和为测试应用而优化的数据采集(DAQ)和仪器控制类库等，从而大大缩短了应用程序的开发时间。

Developer Suite 中包括的其它组件或工具包：

Signal Express

企业连接工具包

VI 分析器工具包

State Diagram 工具包

Express VI 开发工具包

报告生成工具包

高级信号处理工具包

PID 工具包

数字滤波器设计工具包

关于 NI 的其它专业工具包软件关系见下图，感兴趣的可访问 NI 的网站来了解。

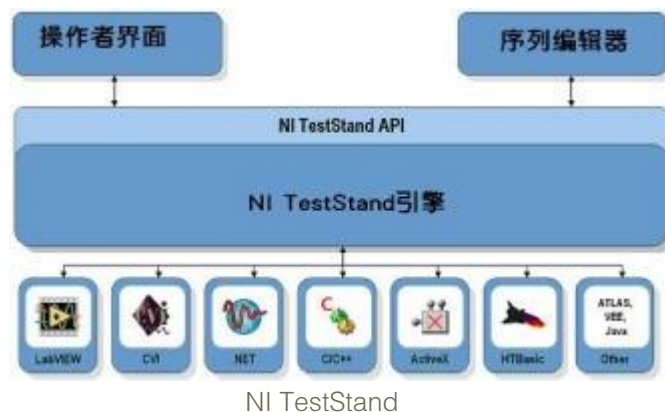


1.5.1.2 虚拟仪器硬件体系结构



“虚拟仪器”硬件体系应该是“虚拟仪器”软件开发环境支持下的所有硬件。NI 就是“虚拟仪器”硬件的专业制造商，因为这部分内容太多了，没法一一列举，NI 测量与自动化产品目录 2008 中包含了全部硬件。根据项目需要去选吧！

1.5.2 虚拟仪器应用 x 管理体系



NI TestStand 是一个简单易用的测试管理环境，用于组织、控制以及运行您的自动化原型开发、验证以及制造测试系统。

使用 NI TestStand, 整合使用您擅长的编程语言开发的测试程序, 快速构建测试序列。NI TestStand 基于一个高速的多线程执行引擎构建, 其强大的性能可以满足您最苛刻的测试吞吐量需求。同时该软件它是可完全自定义的, 因此您可以对它进行修改与增强, 以满足各种特殊需求, 包括自定义操作员界面、生成自定义报告以及修改序列执行需求。使用 NI TestStand, 您可以将精力集中在测试一种特定的产品上, 其他诸如序列化、执行以及报告等任务将由 NI TestStand 为您进行管理。

- 序列开发环境
- 调试
- 多线程并行测试
- 用户管理
- 可自定义的报告
- 可自定义的操作界面
- 源代码控制整合
- 数据库记录

NI DIAdem

NI DIAdem 提供基于配置的技术数据管理、分析以及报告生成工具, 交互式地帮助您对数据进行挖掘和分析。

- 快速进行数据搜索和挖掘, 寻找趋势和相关性
- 载入并分析包含超过 1000 亿个数据点的数据集
- 使用交互式可视化工具研究数据、比较测试运行、寻找异常
- 创建可重用的专业报表, 共享您的结果

1.5.3 虚拟仪器技术小结

从上面介绍可以看到，无论你会哪种计算机语言，都可以从 NI 找到不同的、适合你的虚拟仪器开发环境。

如果你有心想要加入虚拟仪器设计的大家庭,还要具备一些“虚拟仪器技术”所需要的基本知识:

- 应该对计算机的各种总线技术有些了解
- 应该对测试、测量方面的知识有足够的了解
- 应该对传感器技术有一定的了解
- 应该对 A/D、D/A 及其它硬件知识有足够的了解
- 有条件的话，学习一下“数字信号处理”技术

我们今后所介绍的内容则全部是针对 LabVIEW 的，那么我是如何看待 LabVIEW 的呢？



第 1.6 节 本章小结

本章是临时加入的，所以内容不是很十分充足，待以后慢慢来充实。“骨头都有了，还愁肉吗？”届时会在我的博客上发布更新的通知。



我工作后，看到的第一台电脑，HP 85



参加工作后，我在单位见到的第一台电脑，HP-85，是一台测试分析仪配带的。

第 2 章 我眼中的 LabVIEW

第 2.1 节 LabVIEW 到底魅力何在？

从前面的内容中，我们知道：LabVIEW 是美国 NI 公司发明的虚拟仪器应用软件开发平台。大家也完全可以看得出来：我对 LabVIEW 绝对是十分喜爱的。这份热爱来自于获得它的过程是十分不容易，更是来自于它自身所具有的独特魅力，正是由于它在我面前完全展现出了它的独特魅力，从而使我产生了想要得到它、拥有它的梦想，这种梦想多少年来一直挥之不去，直到有一天我真正的拥有它。

那么 LabVIEW 到底魅力何在呢？

2.1.1 LabVIEW 是测试、测量工程师的“玩具”



我们孩提时非常喜欢玩的玩具就是“积木”，用几块大小不同、形状和色彩各异的“积木”就可以拼凑、搭建出各式各样的建筑物（或图案）。

其实，我小的时候最喜欢的“积木”是“铁积木”，一套“铁积木”中有许多零部件，包括：轮子、底盘、拼接条、护板、摇柄等等。

最有意思的是：它是使用螺丝、螺母来构建、组装玩具。开始还是按照标准图谱用螺丝、扳手搭出漂亮的汽车和房子，然后拆了搭，搭了又拆。最后可以按自己（自定义图谱）的想法来搭，依旧是拆了搭，搭了又拆，乐此不疲…，其结果完全取决于你的想像力和创造力是否丰富！



在小学二年级时，我就已经拥有了一套“铁积木”，三年级时因为考试成绩好，就缠着爸爸又买了一套，这样就可以搭建更复杂的、较大型装置了。

回想起那时的我们,可比现在的孩子幸运多了,没有那么多的作业和课外学习(学绘画、学英语、学乐器....),自己一个人在家就可以玩的高高兴兴。但也很不幸的是:我们身上的艺术细胞相对很少!

使用 LabVIEW 来进行程序设计与儿时搭铁积木有很多相似的地方,只不过是:轮子、底盘、拼接条、护板、摇柄等部件换成了函数、结构;螺丝、螺母,被节点和连线所替换;代替标准图谱的是 NI 提供的例程;其它的仍就依赖于你的想像力和创造力是否丰富。

这种十分有趣、好玩的程序设计语言,你能说它没有魅力吗?

2.1.2 LabVIEW 可以快速实现和验证你的想法



一个好的“idea”或设计理念,如果没有切实有效的实施方法和手段,很可能被束之高阁,甚至根本无法与大家见面。

由于 LabVIEW 是一种高效的编程环境,所以它可以迅速地实现你的“idea”或设计理念,甚至是灵光一现的想法也可以很快得到试验与验证,这方面它决不会扼杀你的想像力和创造力。而调试和维护同样是简单有效的。

这种即想即得、快速、实效的程序设计语言,你能说它没有魅力吗?

2.1.3 LabVIEW 可以迅速的提升你的技术能力



提高技术能力除了看书、学习、多实践外,拥有一个快速、灵活、方便的工具也是非常重要的。

在 LabVIEW 开发环境中，内置了大量丰富地程序设计所必须使用的标准函数，包括专业性很强工具包，使用起来非常方便。过去想做而无法做到的事情，现在使用 LabVIEW 则完全可以大显身手了。特别是，利用 NI 提供的例程稍加修改就可以实现你的需求，使你的能力得到大幅度得到提升。关于这方面的例子，以后的章节会不断的介绍给你。

这种大幅度、全方位提升你技术能力的程序设计语言，你能说它没有魅力吗？

2.1.4 LabVIEW 助你成为软件设计大师



每当我为用户完成最终的程序设计、打包发布到目标机后，唯一的感受就是：现在我可以被称为一个软件设计开发人员了。

LabVIEW 提供了系统级的一整套从设计、调试、到最终发布应用程序安装包的软件开发环境。实现这一切的操作，只不过是点击了几下“鼠标”，LabVIEW 助你成为专业的软件开发大师。

这种实现系统级全方位开发设计的程序设计语言，你能说它没有魅力吗？

其实 LabVIEW 最大的魅力在于，它提高了虚拟仪器应用程序设计、开发时的效率。在“效率就是生命”的时代，这一点是尤为重要的。

第 2.2 节 什么是 LabVIEW？



在本《札记》中，从开始到现在大概也数不清有多少次提到了 LabVIEW 这个词，到底什么是 LabVIEW？包括我在内的许多初学者也都曾经提出过类似这样的问题，那么到底什么是

LabVIEW?

其实,关于这个问题,我想无须做更多的解释,仅从它的名称中我们就可以得到非常精准的答案:

LabVIEW 是 Laboratory Virtual Instrument Engineering Workbench(实验室虚拟仪器集成环境) 英文首字母的缩写组合。

它是由美国国家仪器公司 (National Instruments, 简称 NI 公司) 创造、开发的一个系统级、功能强大、而又十分灵活、高效的“虚拟仪器”应用软件开发工具 (环境)。

LabVIEW 核心概念是: 虚拟仪器 (技术); 它的最大特点是: 采用 G (Graphical programming 图形编程) 语言进行虚拟仪器应用程序设计、开发。所以, LabVIEW 的程序也被称作为: VI (Virtual Instrument 虚拟仪器)。

用比较通俗的解释就是: LabVIEW 是一个符合工业标准的系统级虚拟仪器应用软件开发平台, 它包括了采用图形化的虚拟仪器应用程序的设计方法及项目管理、调试、运行、发布等等一整套环节。

纯个人观点



实际上最容易引起混乱的就是:经常用 G 语言来指代 LabVIEW。其实这应该是两个不同的概念。

先看 G 语言: G 语言应该是泛指具有图形化编程能力的所有的编程语言, 而并不是“LabVIEW”的专用代名词。因为安捷伦公司的 VEE Pro 也是一种基于图形化的编程语言 (均属测量领域)。我目前使用的网页设计方法也是基于图形化的设计方法和操作来实现网页设计的。

而 LabVIEW 应该指的是: 一个基于图形化编程方法的**虚拟仪器应用软件开发环境**。它包括了: 基于图形化程序设计的方法、调试、运行、发布等等一系列环节。

所以, 不应该简单的把 LabVIEW 称为 G 语言, 因为它是众多 G 语言中的一种。

但是多年来大家已经习惯了这样叫法, 这里我们还是尊重大家的传统习惯吧! 但至少应该保持概念上的清晰。

虚拟仪器及虚拟仪器技术的缔造者, 美国 NI 公司, 成立于 1976 年。

在 70 年代前后, 美国的 HP (Hewlett-Packard) 公司设计、开发一种用于计算机和仪器通讯的

串行接口系统，简称为：HP-IL（Hewlett-packard Interface Loop）。后来又经过不断的改进成为一种并行通讯接口 HP-IB(Hewlett-Packard Instrument Bus) 或称为 GPIB (General Purpose Interface Bus)。

由于 GPIB 接口总线——有效的解决了计算机和仪器通讯的问题，后来被国际电工委员会 Institute of Electrical and Electronics Engineers，简称：IEEE，批准接纳成为国际标准，也就是人们所熟知的 IEEE-488-1975(IEEE-488.1),及后来修订的 IEEE-488-1987(IEEE-488.2)。实际上，IEEE-488.1 定义了 GPIB 接口的硬件电器标准，而 IEEE-488.2 则定义了 GPIB 的软件语法规则。

后来，NI 公司在此基础上开发出了 488.1 的硬件扩展版，即 HS-488,将 GPIB 总线的传输速率从 1Mbytes/s 提高到 8Mbytes/s，即后来成为国际标准的 IEEE-488-2003。

80 年代初期，NI 公司凭借着在 GPIB 开发上所获得的成功，已经成为基于个人计算机的 GPIB 控制器的稳定开发商和供应商。他们在商务活动中敏感地发现：当时，所有的仪器控制程序都是使用 BASIC 语言设计、开发的，而对于那些精通测试、测量工作的科学家和工程技术人员来讲，使用 BASIC 语言来编制仪器控制程序，可能不是一件很愉快的事，应该是一种负担或者是一种不堪忍受的磨难（肯定不是快乐）！

NI 公司的精英们设想：如果能够发明一种很实用、很方便的仪器控制软件开发工具或软件开发平台，必将彻底改变那些测试、测量科学家和工程技术人员对仪器控制程序设计的态度。可是在当时，NI 公司并非是财大气粗的大公司，搞这样的开发、研究风险是相当大的。应该说：这是每个制定政策的人都十分清楚的问题。即便如此，他们还是下定了决心，于 1983 年 4 月开始，迈出实现这个伟大发明梦想的第一步。

大约是经过了三年多艰苦卓绝的不懈努力，这些天才的发明家，终于实现了他们当初的梦想。LabVIEW 1.0（Mac 版）诞生，那是在 1986 年。20 年后，NI 公司为纪念这一伟大的创举诞生 20 周年，在 2006 年发布了 LabVIEW 8.20 特别纪念版。而 2008 年我们正在使用的版本是：LabVIEW 8.6。

毫无疑问，LabVIEW 的诞生，确实引发了测试、测量仪器领域的一场革命。特别是它以创新的“虚拟仪器”、“软件就是仪器”的概念，以及采用图形化编程的手法，以及所提供的强大的内在分析、处理能力和性能优异的硬件模块支持，逐渐成为测试、测量应用工程师极为有力的帮手。

我对 LabVIEW 的“崇拜”和“热衷”，完全是源于它，使我在项目研究和虚拟仪器应用程序的设计中，体会到了无限的快乐和无比强大的成就感。但是，并非是所有学习过和使用过 LabVIEW 的人，都能够象我一样体会到快乐和成就感，特别是那些刚刚涉足测试、测量领域的年轻人。

事实上，问题出在，LabVIEW 是一个十分强大而又非常复杂的编程环境，即便是一个 LabVIEW 编程高手，如果缺乏对测量知识或硬件电路的充分了解和掌握，也可能无法实现所需的设计工作。换句话说：LabVIEW 很强大，但它还是针对的测试、测量领域，缺乏这方面的基础知识，你仍无法真正地领略到它的强大所在。这也是激发我写此《札记》的原动力，我真诚的希望通过本《札记》所介绍的数据采集基本知识和相关的应用实例能够在这方面对他们有所帮助和提高。

纯个人观点：



我个人一直有一个观点：是 Apple（美国苹果公司）公司不断创新的技术成就和新技术成果地不断涌现，成就了 NI 的发明者。应该说：美洲大陆上两个最伟大的公司，在同一个伟大的时代，各自做出了伟大的发明。直到如今，他们在我心目中，仍旧是世界上最伟大的两个电子公司。

第一，Mac（苹果计算机）电脑的成功设计，的确让 NI 公司的发明家真正的领会到了什么是图形化操作。图形化操作——就是用图标代替代码，用鼠标点击、拖拽图标来实现对电脑的操控。这的确给用户带来了巨大的实用性和灵活性，可以说：图形化操作对电脑的普及和发展起到了相当积极的推动作用。反观当时的 PC 机，还在使用 DOS 操作系统，并且根本就没有鼠标。

第二，Mac 电脑本身强大的内存及图形处理能力，在当时绝对是可以堪称世界第一流的。Mac 电脑本身所具备的先进技术性能，事实上，已经成为实现他们伟大发明的助推器。

第三，Mac 电脑以其灵活、开放的体系架构，提供了丰富的 I/O 扩展能力，这包括：GPIB 控制器、模数转换器和数字 I/O 等等。应该讲，这些能力也恰恰也是“虚拟仪器”本身所需要的。

我的个人主页设计全部是在 iMac 电脑上完成的，而我只使用苹果的 Safari 浏览器和火狐的 Firefox 浏览器。甚至在卧室里，我也可以使用苹果的 iPod Touch 来浏览我的个人主页。

下面给大家公认的、LabVIEW 的一些基本特点：

LabVIEW 是以创新的图形化编程方法，来大大的提高了程序设计的效率。也可以说：使用它完全可以达到“所想”即可以“所得”的程度，这样更能激发人的创造性思维。可以毫不夸张这样讲：就整个自动化测量和自动化控制行业来看，目前，还没有比 LabVIEW 更好、更优秀、更合适的程序设计语言。

LabVIEW 开发的程序是基于数据流的运行方式，它有意、无意的适应了当代计算机的多线程技术和多核技术的发展。目前我们所使用的计算机多为双核，但是当四核、八核、N 核的计算机商业化后，LabVIEW 将更加显现出它强大无比的威力（NI 公司已于 2008 年加入“多核”计算机协会）。

LabVIEW 已经渗透到工业测量的各个领域，已经成为事实上的工业标准。与此同时在嵌入式、FPGA、PDA、DSP、实时控制等领域也发挥着巨大的作用。学习使用它可以使你始终处于技术发展的前沿，并受用终身。

LabVIEW 内置了丰富的数据分析、处理函数，其数量多达数百个。从实际应用的角度出发，它还提供了大量的实例供使用者参考。NI 的网站更是内容丰富，资料齐全。这些都为 LabVIEW 的学习者提供了极大的帮助。

LabVIEW 本身内置了图形代码编译器，可以说：这绝对是世界上最好的动态编译器。它在你在进行程序代码设计地同时，就同步、实时地进行程序代码编译，只有当你的程序出现错误时你才会感觉到它的存在，并引领你到发生错误的地点及提供修改意见指导。

当然，LabVIEW 的优点和特点不止这些，以后我们将会慢慢地涉及、体会到。现在要说的是：鼓起你的勇气，去热烈地拥抱它吧！它带给你的决不是一时的快意，而是从自动化测量工作中获得永恒的快乐！

我必须再次承认：对于 LabVIEW，绝对是存在着我个人偏爱的观点和态度。因为我不想再去设计 A/D、D/A 转换电路。就这些电路而言，它们的更新发展速度实在是太快了，我没有能力和精力去不断的完善和改进电路设计。在我这个年龄段，根本就不会去学习、使用其它任何计算机语言来编写复杂的分析、处理程序，我只想做我应该作好的测试、测量工作。恰恰是 LabVIEW 的出现，满足了我的这个需求：即不用搞新的硬件电路设计，又能够设计编写近乎专业级的测试分析、处理应用程序。

虽然，LabVIEW 是以“虚拟仪器”为核心概念诞生的，并且以图形化编程语言来进行程序设计，但为了照顾其他使用不同语言的“虚拟仪器”设计者，虚拟仪器的缔造者 NI 公司还提供了其它两种不同方式的“虚拟仪器”软件开发环境：

LabWindows/CVI——基于标准 C 的虚拟仪器软件开发环境

Measurement Studio——基于 .net 构架的语言，如：VB、C#、VC++ 语言的虚拟仪器软件开发环境。

而本《札记》的内容完全是基于 LabVIEW 的，并不涉及其它那两种开发环境。

题外话：

Apple 公司和 NI 公司都是具有着极强的一致性和相似性的美国公司，都是即设计软件，又设计硬件；都是在各自的领域，与时俱进不断的发展。这大概也是我长久以来一直钟爱他们产品的一个主要原因吧！





既然 LabVIEW 是一种图形化的编程语言，那么它到底好在那里？它本身会有那些其它编程语言不一样的特点？……。

下面我们就试着解开这些疑问吧！

第 2.3 节 LabVIEW 图形化编程语言

在讨论 LabVIEW 图形化编程语言的特点之前，还是要先回顾这样几个基本概念：

2.3.1 什么是语言？



语言是一种符号系统（文字和语音），是用来表达我们的思想及进行相互交流和沟通的一种手段或着说是：一种工具。

语言是人类创造的，人类也创造了一些没有语音的语言系统，比如：手语、旗语等等。

即使是网络时代的到来，亦被人们称之为：信息化时代或信息化革命（网络起到沟通桥梁的作用）。其本质上还是关注的人际间的交流和沟通，而交流和沟通的关键仍是语言，所有语言最核心的作用就是起着人类间相互交流和沟通的作用。

2.3.2 什么是计算机语言？



当计算机出现后，人类又创造出能够与计算机进行沟通的语言，称为：程序语言。计算机语言的演变和发展，完全是为了更好的指挥计算机为人类工作。

程序语言是用来指挥“冷冰冰的计算机”表达我们的想法（意念）并能够被计算机执行的程序代码。

计算机程序语言历经几代的发展演变：

机器语言(Machine language)

最初的程序语言（直接面对物理层的程序语言，与处理器有关）我们称之为：机器语言。

机器语言，也就是人们通常所说的：机器码。机器码是可被中央处理器（CPU）加载并执行的由 0 和 1 组成的序列。由于硬件处理器的不同，所以机器码序列也会是不一样的。可以讲，机器码是处理器（CPU）可识别的唯一语言，并且是执行速度最高的语言。它也是与计算机的硬件（CPU）有着最为直接的关系的一种语言。

但是，当人们看到机器码时，由 0 和 1 组成的序列（程序），实在是感到毫无任何规律、枯燥无味，并且无法理解和记忆。特别是，当计算机处理器的性能不断提高时（位数增加），机器语言可能被视为世界上最无法读懂和难以记忆的语言。试想一下，由 64 位 0 和 1 组成代码，如何记忆和理解它们。

为了解决这个问题，使更多的人、更方便的使用计算机，并提高编程效率充分发挥计算机的作用，人们对机器语言进行了抽象处理，从而导致了较高一级的汇编语言的诞生。

在处理复杂事物时，采用抽象处理的方法是降低事物复杂度的最好方法。

汇编语言（Assembly language）

汇编语言，是从机器语言中抽象提炼出来的使用缩写或助记符进行编程的低级编程语言，并且提供较少的控制指令和数据类型，其中的每一条语句都对应于一条相应的机器指令（代码）。通过汇编程序开发环境，可以将汇编语言翻译成机器语言，汇编语言同样与给定的处理器相关。

在早期，包括操作系统在内的许多系统软件都是用汇编语言编写的（比如：UNIX 操作系统）。使用汇编语言的好处在于可以提高执行速度，并且程序员还可以直接访问系统硬件。但是，汇

编语言由于其开发环境和汇编程序都依赖于给定的处理器，并且其指令和数据类型很少，同时程序的可读性和可移植性都很差无法广泛使用。于是人们再次对汇编语言进行抽象处理，从而导致了更高级计算机程序语言的诞生。

高级语言

高级语言，是从低级的机器语言抽象而来的更高级计算机程序语言。高级语言通常使用和英文单词类似的关键词，这些关键词可以被翻译成多条机器指令。实际上，比汇编语言高级的计算机语言都是高级语言。

C 语言是目前最为广泛使用的高级语言。它既具有一般高级语言的特性，同时又具备了一些低级语言特性（某些汇编语言的特点）。

当然，高级语言还有很多种比如像我们所熟悉的：面向对象的 C++、Java 及具有可视化编程环境的 VC、VB、Delphi 等等。这些高级编程语言的存在和发展说明，除了各自的特点不相同，应用领域不同外，还有编程方式的不断改进的推动。

比如：Basic 演变到 Quick Basic 然后又演变到 Visual Basic。

C 演变到 C++ 然后又演变到 Java（号称没有指针和头文件的 C++）及后来的 VC。

这些已不在我们讨论的范围内，有兴趣者可以看相关的书籍来了解。这些高级语言的程序设计方式都是基于文本代码的。

还有没有比高级语言更好的计算机程序语言呢？有，那就应该是图形化编程语言。

图形化编程语言

我们前面谈过，人类在解决、处理复杂问题时，往往采用的是抽象处理的方法，比如：从“机器语言”抽象到“汇编语言”，又从“汇编语言”抽象到“高级语言”。

那么现在的问题是：“高级语言”是否还可以进一步进行抽象出“超级编程语言”呢？从计算机语言的发展规律来看，回答应该是肯定的。并且早在二十多年前，这种语言的雏形就已经出现，那就是 LabVIEW 图形化编程语言。

为什么说：图形化编程语言比高级语言还要好呢？那就要从图形本身的特点谈起：

1、因为使用图形，可以直观的表达比较复杂的事情。

比如：路口的交通指示牌，就是用最简单的图形来诠释通行规定。象：禁止通行、禁止停靠、禁止鸣笛、限速 XX 等等....。因为我不大会开车，所以可能会比喻不得当，见谅！

2、图形表现的是一种视觉语言

这种视觉语言具有简单、明了的作用和功能。

比如：十字路口的交通信号灯，只使用三种颜色，依靠不同的状态可以表示出：停止通行、准备通行、正常通行和准备停止通行等多种状态。此时采用语音和文本对于驾驶车辆都不适合。

手语、旗语也是视觉语言。

3、图形因为形象化所以容易记忆和理解

学习过程本身就是：记忆加理解。越简单的表述就越容易记忆，越形象化的东西越容易理解。
比如：禁烟的标识、危险品的标识、卫生间的标识等等...

从高级语言抽象出来的图形化编程语言，基本上也具有上述图形特点。

下面通过几种高级语言的程序代码,来看看基于代码的高级语言与基于图形的编程语言的各自特点。

2.3.3 几种高级语言和图形化语言的代码展示

在许多计算机程序语言的教科书中，都是通过介绍“Hello-world”程序来向读者展示出第一个程序示例作为开场白。我搜集了一些语言的该程序代码示例，但未经编译运行，在这里仅作为展示和比较用。

在如此重多的语言中，你更喜欢哪一种呢？我喜欢 LabVIEW——一种图形化的编程语言。

C 程序代码

```
# include <stdio.h>
int main ()
{
    Printf ( “Hello , World! \n” );
    Return 0;
}
```

C++程序代码

```
# include <iostream>
Int main()
{
    Std::cout<<’ Hello , World !\n’ ;
    return 0;
}
```

C# 程序代码

```
class ConsoleHelloWorld
{
    public static void main ()
    {
        System.Console.WriteLine( “Hello , World !” );
    }
}
```

JAVA 程序代码

```
public class HelloWorldCommandLine{  
    public static void main (String[ ] args){  
        System.out.println( "Hello,World!" );  
    }  
}
```

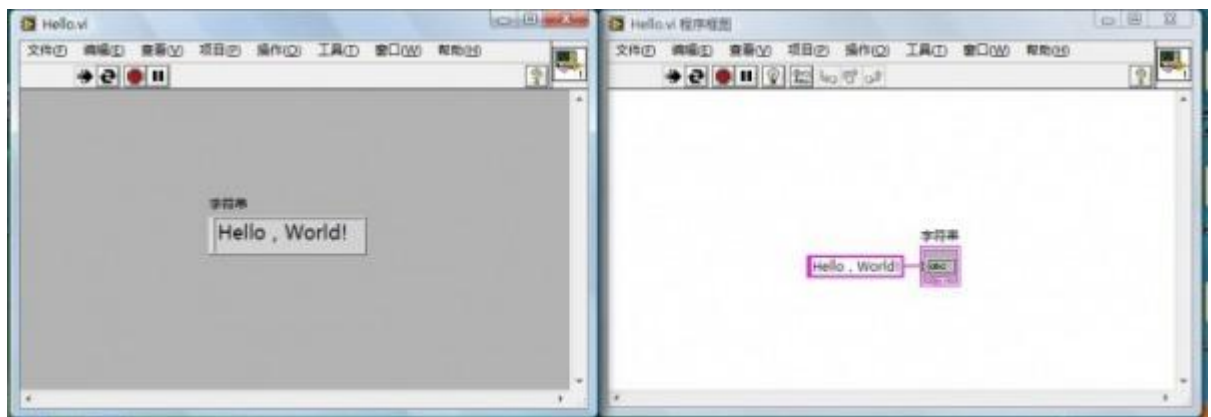
Windows 程序代码

```
# include <windows.h>  
Int WINAPI WinMain (HINSTANCE hinstance,HINSTANCE hPrevInstance,  
PSTR szCmdLine,int iCmdShow)  
{  
    MessageBox(NULL,TEXT( "Hello, World !" ),TEXT( "HelloMsg" ),0);  
    return 0;  
}
```

VB 程序代码

```
Dim ny As Integer  
Private Sub Form_Load ()  
    Ny=0  
    Text1.Text=" Hello , World !"  
    Text1.ForeColor=RGB(255,0,0)  
End Sub
```

LabVIEW(图形化)程序代码



LabVIEW 图形化程序代码见右图,LabVIEW 图形化程序本身还提供了一个漂亮的前面板见左图。一看就清楚,那种语言更简单、更高效。

2.3.4 计算机语言的发展关键是不不断的进行抽象处理

从上一节几种编程语言的简单对比中，我们不难看出：图形化编程语言的确简单、明了。更可贵的是：LabVIEW 还提供了一个前面版，而其它高级语言基本上还是在“控制台”上输出、显示。如果使用基于文本的编程方式，仅一个前面版可能就要编写很多行的程序代码。

下面在通过两个例子来做进一步的说明：

例 1：

小时候，在我们刚刚开始会说话时，大人通常是用“看图说话”的方式来教我们认识什么是苹果？什么是梨？什么是香蕉？显然，利用图形或图像的概念使我们很快认识了这些水果。试想如果用文字或拼音来教我们认识这些水果，效果决不会好的。

换句话说，图形或图像由于简洁明确，的确要比文字或拼音更适合初学者来认识和区分。事实上，编程语言也是具有相同的规律。

对于那些学习某种高级语言的人来讲，高级语言复杂的语法规则、指针、内存、类库等等掌握根本谈不到轻松易学。

而 LabVIEW 确是做到了这一点，使用 LabVIEW 图形化的编程方法，根本无需考虑什么：语法规则、指针、内存、类库等。LabVIEW 这种图形化语言将许多复杂的事情抽象的极为简单明了。

例 2：

如果时光倒退回二十多年前，那时人们对计算机的操作、控制还是基于被称为：DOS 的磁盘操作系统（Diskette Operating System），即便是一个很简单的操作人们也会噼噼啪啪的敲击一阵键盘，对于复杂操作人难免还要认真仔细的查阅 DOS 手册。那时的 MicroSoft 还是一个编写 DOS 的小公司。

后来，Mac（苹果电脑）首先实现了计算机的图形化操作设计，点击鼠标、利用拖拽即可完成对计算机的一些操作控制。LabVIEW 大概也是受到了图形操作系统的启发（不知是否真的如此），从 1983 年开始了 LabVIEW 的设计，并于 1986 年在 Mac 机上完成了 LabVIEW1.0 版的发布。直到 MicroSoft 也设计出图形化操作系统时，LabVIEW 才发布了 Windows 版。这大概是 LabVIEW2.0，到 LabVIEW3.0 发布时它已经全面支持跨平台使用了。

计算机操作系统的图形化，无疑加快了计算机使用的普及和使用的大众化（当然也包含硬件发展的推动作用），这是无可争辩的事实，使计算机由过去的专供专业人员操作使用工具，而真正变成了不分年龄、不分专业的大众工具。其中，真正起核心作用的还是图形化操作系统得简洁、方便和易学。

编程语言的图像化应该与操作系统图形化一样,给那些不善于使用代码编程的人带来了实现复杂程序设计的机会（我就是其中的一个受益者）。

其实许许多多学习过 LabVIEW 的人都应该有这样的体会：图形化的编程方法的确大大降低了程序设计的复杂度，在测试、测量领域 LabVIEW 的确是比那些“高级语言”更好的“超级编程语言”。

当 LabVIEW 真的成为通用编程语言时，它一定可以被称为：超级编程语言。

LabVIEW 是图形化的编程语言。那么，还能对它继续进行抽象处理，进一步提高它的简洁性和方便性吗？

Express VI

DAQ 助手

仪器 I/O 助手

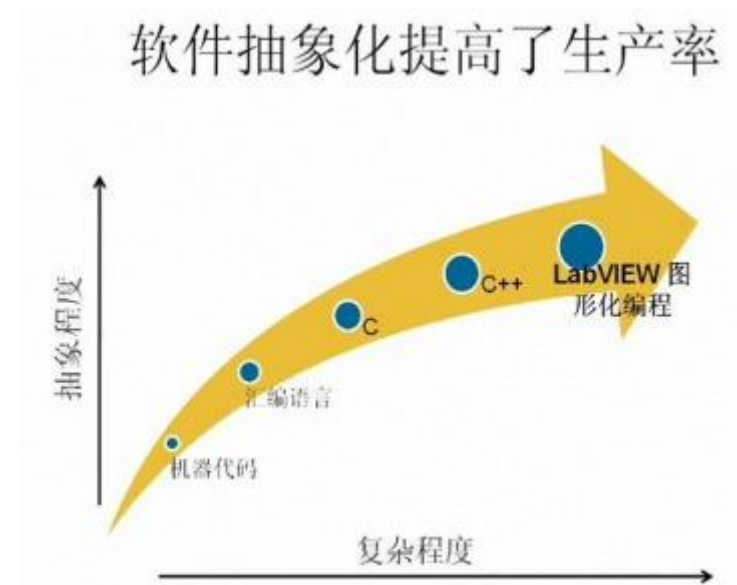
视觉助手

难道这些不是吗？

对上述这些“快速 VI”和“小帮手”(国语)还能再抽象吗？

Signal Express 不是吗？

使用 Signal Express 甚至于无需编写任何代码即可实现测量、分析、处理。



第 2.4 节 学习 LabVIEW 应该看的几本书



上一节，我们在计算机语言程序比较过程中，已经见到了第一个完整的 LabVIEW 程序，对图像化编程语言有了初步的认识。

在 2000 年前后，我刚刚知道 LabVIEW 时，有关 LabVIEW 的中文书籍还不多见，到 2004 年我开始学习 LabVIEW 时，大概有不到五种介绍 LabVIEW 的中文书籍。现在，涉及到介绍 LabVIEW 的中文书籍有数十种之多，那么学习 LabVIEW 看那些书籍比较好呢？关于这一点，从我的博客的日志中也可以看出，只要是介绍、推荐“书”的日志，一般的点击率都很高，说明大家对这方面的内容还是比较关心的。

下面依据我学习的自身体会，介绍几本学习 LabVIEW 的书籍。

2.4.1 中译本书籍

原版：LabVIEW Graphical Programming （3 edition）2001 年

中译本：LabVIEW 图形编程 武嘉澍，陆劲 北京大学出版社, 2002 年



我在 2004 年 8 月购得此书的中文版，它的内容涵盖了 LabVIEW 1 到 LabVIEW 6 的整个历史发展阶段。作者用通俗的语言，像讲故事般的介绍了自动化测试的基础知识及 LabVIEW 的特点。此书绝对堪称 LabVIEW 的经典佳作，百看不厌。不幸的是：好像目前这本书可能不容易买到了。

本书的作者之一，(美) Gary W. Johnson 是一位资深的 LabVIEW 专家，大概从 1994 年开始，先后出版了许多关于 LabVIEW 的书籍，下面是我搜集到的一些书的封面。



[1994 年]出版了书名为：LabVIEW Graphical Programming Practical Applications in Instrumentation and Control（大概对应于 LabVIEW 3.0）By Gary W. Johnson (Author)

[1997 年]出版了本书的第二版：LabVIEW Graphical Programming Practical Applications in Instrumentation and Control（大概对应于 LabVIEW 4.0）By Gary W. Johnson (Author)

[1998 年]又新出版了书名为：LabVIEW Power Programming（大概对应于 LabVIEW 5.0）By Gary W. Johnson (Author)

[2001 年] Gary W. Johnson 又与 Richard Jennings 合作再次出版了：LabVIEW Graphical Programming（3 edition）2001 年（大概对应于 LabVIEW 6.0）。也就是我们上面所看到的中译本的英文原版。

[2006 年] Gary W. Johnson 与 Richard Jennings 合作又再次出版了：LabVIEW Graphical Programming（Fourth edition）对应于 LabVIEW 8.0 版

真心希望此书的译者能够再次翻译第四版的该书，已补中译本（北京出版社 2002 年版）脱销的缺憾。之所以推荐此书，就是看到作者在长达十几年的时间内不断的耕耘写作，把自己积累的经验 and 工作的阅历一次再一次准确的传达给广大读者。

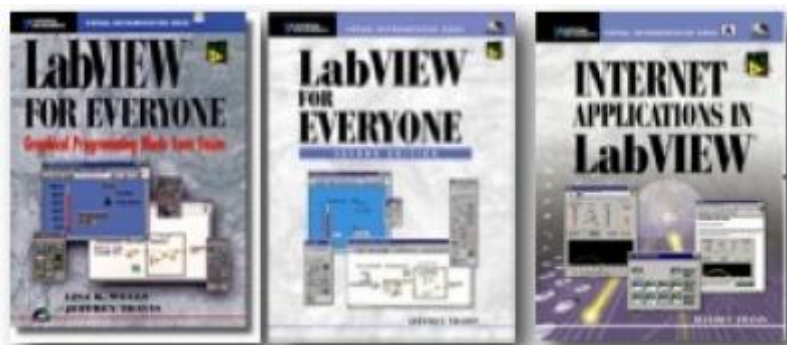
原版：LabVIEW for Everyone (3rd Edition)

中译本：LabVIEW 大学实用教程



这也是此书的第三版,是非常好的 LabVIEW 书籍,堪称 LabVIEW 圣经。应该是每个学习 LabVIEW 的朋友必读之作。其中译本于 2008 年 6 月出版。

本书的作者具有 15 年以上从事 LabVIEW 开发与教学方面的丰富经验, 对 LabVIEW 的原理和内部机制理解深刻, 通过深入浅出、风趣的阐述, 使读者能够快速理解并掌握 LabVIEW 的基础知识和编程技巧。



本书的第一版出版于 1996 年底, 对应的是 LabVIEW 4.0, DAQ4.9。本书的第二版出版于 2001 年, 对应于 LabVIEW 6.1, 我们现在看到的中译本原版书是出版于 2006 年 8 月, 对应于 LabVIEW 8.0。

本书的第一作者, 还出版了 “Internet Applications in LabVIEW” 一书。

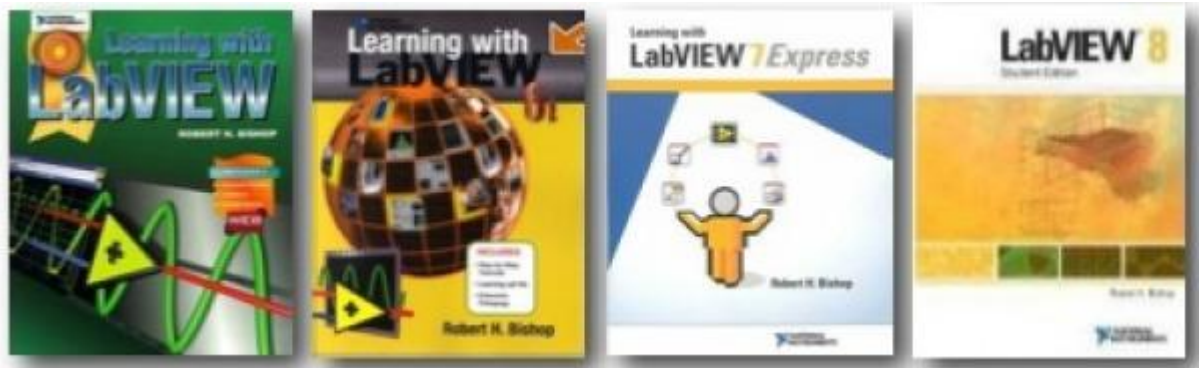
此书的第二作者, 也是 OpenG 的创始者, 多次撰写有关 LabVIEW 的书籍, 也是一名享有盛誉的 LabVIEW 大家。

原版: Labview 8 2007 年 by Robert H. Bishop (Author)

中译本: LabVIEW 8 实用教程 2008 年出版



本书由 Robert H.Bishop 撰写, 是 NI 公司 LabVIEW 8 学生版的正式教材。本书一步一步地指导学生如何构建、调试和运行 VI; 每章包含提示、小结和术语等, 以便指导学生自学; 并配有多幅带有注解的图片; 供用户发布或下载 VI、提问以及访问更学层次的 LabVIEW 信息的配套站点 <http://www.prehall.com/bishop>。



值得一提的是：本书是系列丛书，随着 LabVIEW 版本的变化，本书的作者还相继出版了：

[1999 年]： Learning With LabVIEW （大概对应于 LabVIEW 5.0）

[2001 年]： Learning With LabVIEW 6i（有中译本）

[2003 年]： Learning With LabVIEW 7 Express（有中译本）

[2007 年]： LabVIEW 8 Student Edition（有中译本）

由此可见，作者也是一位资深 LabVIEW 大侠。

原版： A Software Engineering Approach to LabVIEW 2003 年

中译本： 软件工程方法在 LABVIEW 中的应用 清华大学出版社, 2006 年



本书的作者首次从软件工程师的角度来讨论 LabVIEW 的应用，本书适合中、高级开发者。

编写 LabVIEW 软件来完成复杂的任务从来都不足轻松的事，特别是在由于最后提出的功能要求而导致系统产生“复杂性爆炸”的情况下，这会迫使你重新编写许多代码!Jon Conway 和 Steve Watts 对此提出了一种较好的解决方案：LCOD(LabVIEW Component Oriented Design)——LabVIEW 面向组件的设计方法，这是第一次在 LabVIEW 编程中应用软件设计原理。本书内容通俗易懂、风格清新，即使不是计算机 专家，也能够轻松领会。

LCOD 软件工程技术可使软件更健壮，更容易处理复杂性问题，它让软件变得更简单!即使是大型的工业级应用也会变得可管理。

设计首先考虑灵活性，让更改和修正工作更加轻松。

对经过作者试验和测试的技巧进行注重实效的讨论，这些内容足为程序员的实际工作准备的。

总结了设计原则，LCOD 概论、实现过程以及补充技巧，工程学基础，风格问题等。

我看到有的朋友在博客中写道：“此书看了 N 遍，还是有些看不懂，但也大概了解作者的用意”。其实我也是看了 N 遍，也没太看懂。但是还是在继续学习、理解了很多东西，应该说是受益匪浅的。实实在在的讲我还准备再看 N 遍。

2.4.2 中文图书

LabVIEW 程序设计与应用(第 2 版/杨乐平/电子工业出版社)



这套书出版的比较早,在 2004 年可见到的中文书中,应该是有分量的。本书对应于 LabVIEW 6.0 版,是我 2004 年学习 LabVIEW 时的主要参考书之一,所以也将此书列在这里。

其高级程序设计部分只有第一版,第 2 版的是程序设计和应用。

LabVIEW 8.20 程序设计从入门到精通(附盘) 作者:陈锡辉 张银鸿

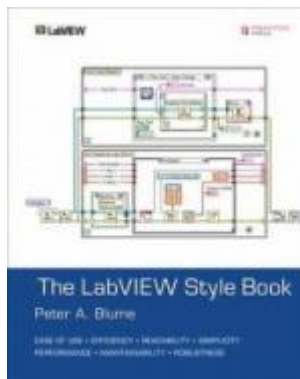


这是一本较为全面的介绍 LabVIEW 的书,相信它的读者应该不在少数。以这本书为主来学习 LabVIEW 应该是一个比较好的选择。

本札记也将此书列为参考书之一。

2.4.3 英文图书

The LabVIEW Style Book (NI Virtual Instrumentation Series)



这是一本关于 LabVIEW 编程风格的书，应该讲是相当好的，十分可惜还没有中译本。

2.4.4 小结

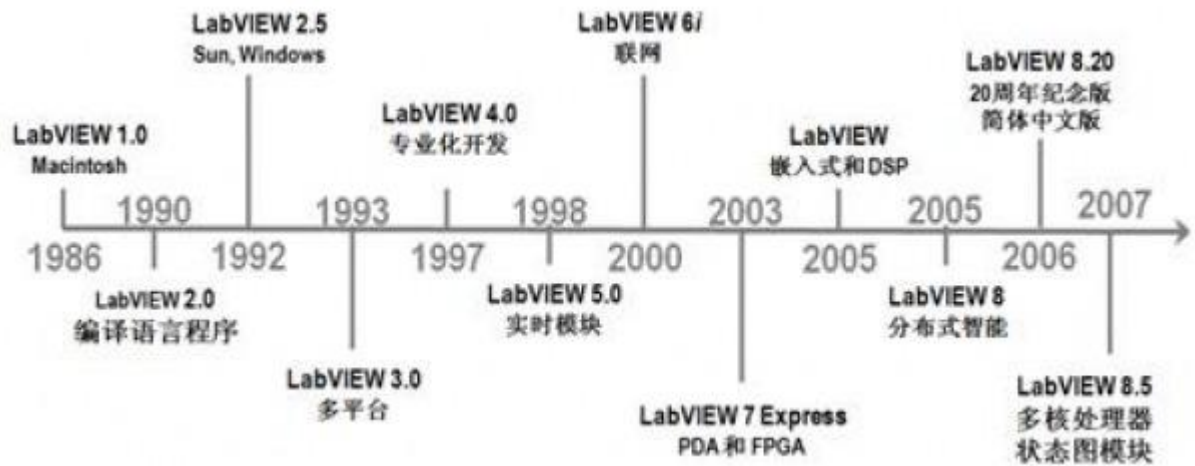
我赞成这样的说法：LabVIEW 的“帮助”文档是最好的 LabVIEW “教科书”。学习主要的问题是学习态度，兴趣也是一个要素。每本书都具有作者自己的观点和特点，捧住一本书看到底就一定能学习的很好。我的办法是：好书要读 N 遍，每次重读都会有新的认识。

书要读，而不是看。看可以一目十行，而读则是要用心去领会、去理解。人们常说的：看报、读书大概就是这个意思。

第 2.5 节 LabVIEW 版本的变迁

在上一节中，谈到了学习 LabVIEW 应该看的几本书。因为每本学习 LabVIEW 书中的内容都与 LabVIEW 的版本有关，所以还是有必要先了解一下 LabVIEW 版本的变迁。下面就简单回顾一下 LabVIEW 最近的发展历史（也仅限于我能够收集到的版本），从这里也可以间接的体会到 LabVIEW 的发展速度有多快。

从 LabVIEW 的软件版本来看（我能收集到的），应该有 LabVIEW 5 系列、LabVIEW 6 系列、LabVIEW 7 系列和 LabVIEW 8 系列。发布年份可能有误，以 NI 为准。



2.5.1 LabVIEW 5 系列：



LabVIEW 5.0 发布于：1998 年
 LabVIEW 5.1 发布于：？
 LabVIEW 5.1.1 发布于：2000 年 3 月

2.5.2 LabVIEW 6 系列：



LabVIEW 6.0 发布于：？
 LabVIEW 6.0.1 发布于：？
 LabVIEW 6.02 发布于：2001 年 2 月
 LabVIEW 6.1 发布于：2002 年 1 月

2.5.3 LabVIEW 7 系列:



LabVIEW 7.0 发布于: 2003 年 5 月

LabVIEW 7.1 发布于: 2004 年 4 月 (同年 8 月, 我开始使用 LabVIEW)

LabVIEW 7.1.1 发布于: 2004 年 11 月

2.5.4 LabVIEW 8 系列:



LabVIEW 8.0 发布于: 2005 年 10 月

LabVIEW 8.0.1 发布于: 2006 年 2 月

LabVIEW 8.20 发布于: 2006 年 8 月

LabVIEW 8.2.1 发布于: 2007 年 3 月

LabVIEW 8.2.1f4 发布于: 2007 年 9 月

LabVIEW 8.5 发布于: 2007 年 8 月

LabVIEW 8.5.1 发布于: 2008 年 4 月

LabVIEW 8.6 发布于: 2008 年 8 月

LabVIEW 8.6.1 发布于: 2009 年 2 月

2.5.5 小结

从 NI 的 LabVIEW 版本号, 可以看出:

1、 系列号: 5、6、7、8 表示新的系列, 软件结构或功能可能有重大改进 (付费升级)

- 2、版本号：5.x、6.x、7.x、8.x 表示软件有新的内容或比较大的改进（付费升级）
- 3、版本号：5.x.x、6.x.x、7.x.x、8.x.x 表示软件较上个版本进行了修补（免费升级）

从上面的情况分析，我个人认为以下版本是最稳定版本。如果不准备马上升级的话，最好使用下面系列中相对应的版本。

LabVIEW 5.1.1

LabVIEW 6.1

LabVIEW 7.1.1

LabVIEW 8.0.1

LabVIEW 8.2.1

LabVIEW 8.5.1

LabVIEW 8.6.1

第 2.6 节 学习 LabVIEW 的理由



爱需要理由！

不爱需要借口！

难道学习还需要什么理由吗？

的确，学习任何知识都不需要找出一个理由。

但是，如果你置身于工业测试、测量领域，我还是建议你学习 LabVIEW。

请关注这样一个事实：那种高级计算机语言可以让 50 多岁的人也能够开始学习计算机应用程序设计，并能够很快达到专业程序员的设计水平？谁能告诉我答案！可是 LabVIEW 确做到了，我就是答案。

不过还是先不要急着下决心，不妨先看看我学习 LabVIEW 的理由。

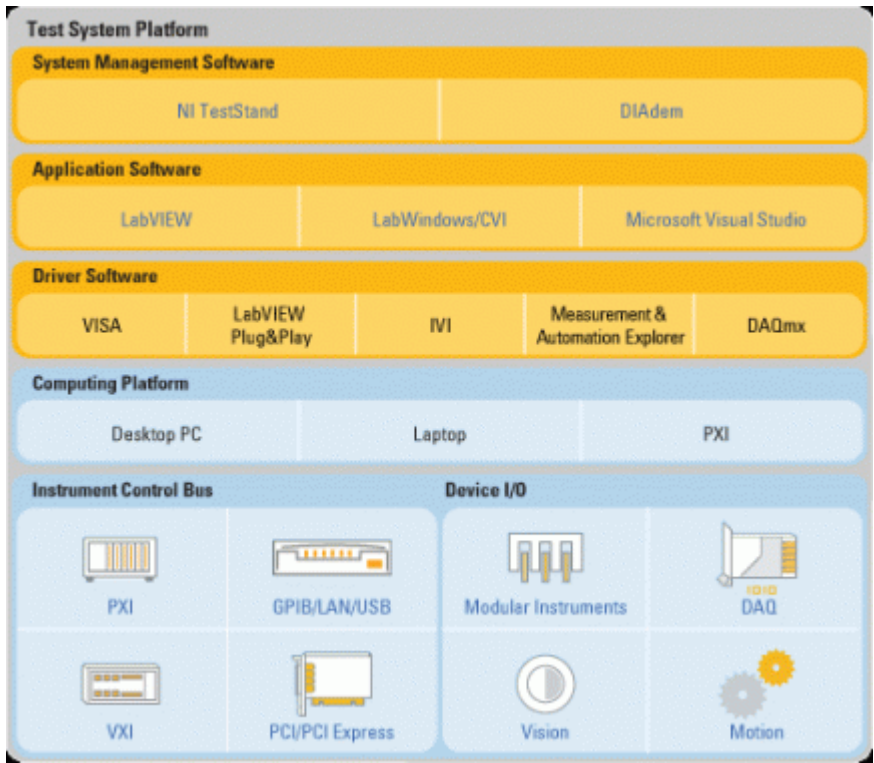
2.6.1 图形化编程的特点使得老少皆易



就工业自动化领域测试、测量方面的应用程序设计而言，LabVIEW 的出现绝对是革命性、创造性的。原因就是它从根本上，改变了人们所习惯的、传统的基于文本撰写代码的编程方式，取而代之的是使用鼠标来点击、拖拽图形、图标、连线等方式来进行程序设计。而这些图形、图标所代表的“控件”或“函数（或方法）”是通过对高级语言进行高度抽象所获得的。所以使得整个编程的过程变得更加简单、方便、有效，从而彻底将编程人员从复杂的语法结构以及众多的数据类型和不停的编写代码、编译、查找错误的过程中解放出来，它为程序员的程序设计工作降低了复杂度，使程序设计者能够更加专注于应用程序的设计，而不用担心语法规则、指针等是否使用的正确。使许多人象我一样没有进行过专门程序设计学习的人，成为高生产力、高效率的程序设计者。

在我退休以后，或许会与我的孙子辈（目前没有）一起共享 LabVIEW 给我们带来的快乐，他玩弄他的机器人（图形化编程的乐高机器人），我仍然继续搞我的虚拟仪器项目学习、开发工作。

2.6.2 系统级的图形化编程环境



之所以说 LabVIEW 是系统级的图形化编程环境，就是因为它提供了一整套基于图形化的软件程序设计方法，包括从编译、调试、发布等等一整套技术环节。

系统级的意义还在于它还提供了相应的硬件体系架构和管理软件体系。

LabVIEW 还支持 Windows、Mac OS X、Linux 等多种计算机操作系统，这种跨平台特性在当今的网络化时代是非常重要的。试想在 Linux 操作系统下设计的 VI，通过网络传递到其它平台上无需改变任何代码，即可使用或调试是一件多么爽快的事情。这大大改善了使用者之间的交流、沟通及评估的灵活性。同时，它还可以充分发挥、利用不同平台自身所具有的优异性能，例如：Windows 系统的广泛性；Mac OS X 系统的美观、时尚；Linux 系统的安全性等等。

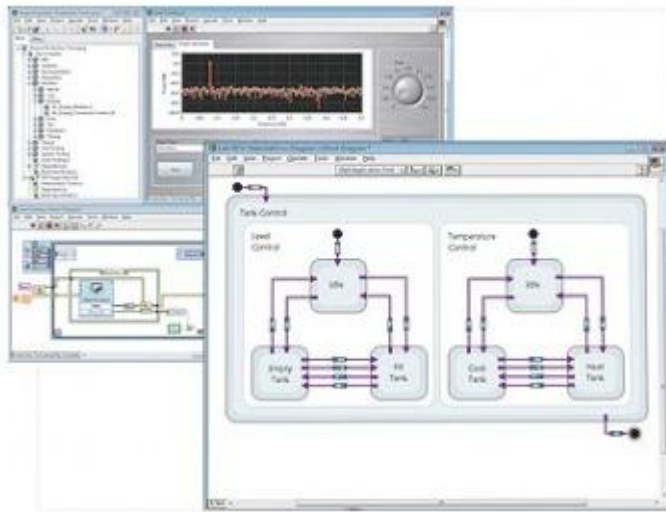
随着，计算机操作系统的不断升级和改进，使 LabVIEW 的开发环境也同样得以不断的改善。举一个简单的例子：Microsoft 最新的操作系统 Vista，界面风格进一步改善，在新的操作系统上使用 LabVIEW，它的 GUI 一定会更加美观、时尚。

尽管 LabVIEW 已是一个独立的图形化软件编程开发环境，但是为了照顾到已习惯使用其它的高级编程语言的编程者，它还提供兼顾其它高级编程语言的开发环境，使已习惯于其它编程语言的使用者也能够充分利用 LabVIEW 的强大的自动化测试、测量及分析、处理能力。

LabWindows/CVI 提供了对 ANSI C 的支持。

Measurement Studio 提供了对 Visual Basic、Visual C# 及 Visual C++的支持。

2.6.3 简洁、高效的程序设计能力



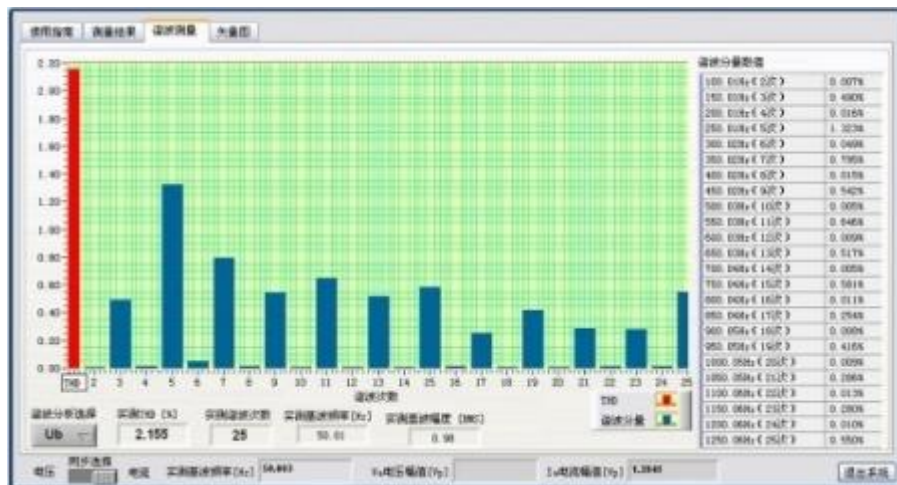
使用 LabVIEW 图形化编程的最大特点就是：编程效率极高。关于使用图形化编程手法可以减少编程时间、缩短开发周期、降低开发成本等说法，已在很多介绍 LabVIEW 的书中进行了充分的表述，我不准备再进行复述。这里，从另外一个角度仅谈谈自己对这方面的体会。

作为应用项目的设计开发者，通常的关注点是设计的合理性和最终结果的正确性。当然，他们也会注意到编程过程的效率。实际上，减少编程时间、缩短开发周期、降低开发成本等大多都是从整体经济利益方面来考虑的。我个人认为，在提高效率这方面，人们往往忽略了“人——设计者”效率得到提高这个要素。由于 LabVIEW 采用的是图形化编程的方法，所以大大降低了编程过程的复杂度，仅仅使用几个快速 VI 就可以非常简单、迅捷的完成程序设计，实现分析和对结果的处理。如果在此基础上还要进行其它分析，假如使用其它的编程语言能够做到这么简单、直截了当吗？

图形化编程可以简单、方便、非常灵活的实现程序设计，立即运行就可以看到分析处理结果。特别是 LabVIEW 的即时编译能力，可以在编程的同时进行检查及时发现错误代码。正是因为编程效率高，改动及替换方便，程序代码直接易懂，所以会更加激发设计者的创造性思维及成就感，自然就会把设计工作重点放在如何更好的分析、处理上，而不是如何编程上。换句话说，当一个新的设计思想出现后，马上就可以得到实验验证或改进，甚至是边设计边改进。而其它的任何高级编程语言都不会如此的快速、方便（因为要写许多代码和进行编译后才能运行）。所以我们说，增强设计者的创新效率和信心是 LabVIEW 不可忽视的内在特点，其价值是无法估量的。

事实上，我们已经看到了 LabVIEW 的发展是迅速的，并且是紧跟时代科学技术的发展，比如：它对嵌入式的支持、对 DSP 的支持以及对 FPGA 的支持及多核技术。所以通过学习 LabVIEW 会同样保证我们技术的持续进步，而付出的代价相对是比较小的。

2.6.4 强大的图形化分析处理能力



LabVIEW 提供了无比强大的分析、处理 VI 库函数及许多专业的工具包，例如：高级信号处理工具包、数字滤波器设计工具包、调制工具包、谱分析工具包、声音振动工具包、阶次分析工具包等（当然都是要花钱购买的），这是任何其它高级编程语言无法提供的。结合 LabVIEW 独特的数据结构（波形数据、簇、动态数据类型等）使得测量数据的分析、处理非常简单、方便、并且实用性很强。很难想象，如果使用文本编程方式进行数字滤波设计或功率谱分析会增加多少工作量，甚至能否设计完成都值得去考虑。

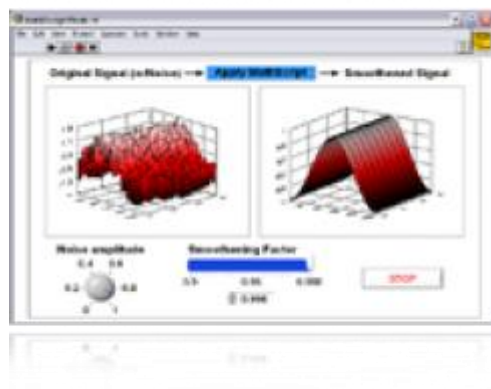
特别是 NI 新近推出的 LabVIEW MathScript，将面向数学的文本编程扩展加入到了图形化的 LabVIEW 中来，提供了除图形化数据流编程以外的另一种自定义开发应用系统的方法，为用户提供获得了获得最佳方案设计选择的机会。

LabVIEW 最大的优势就在于图形化的分析处理方法。从应用角度看，LabVIEW 的分析处理能力绝对是超级的，它使得设计者会更加专注于应用项目的设计，而不是如何进行数据的分析、处理。从而给设计者带来更多的的是工作中的快乐和工作中的成就感。这一点我是绝对认同的。

LabVIEW 的另一个优势就是仿真能力，在设计原型阶段可通过仿真来评估设计的合理性和正确性。由于使用的是图形化的编程方法，这样的工作很快就可以实施并及时得到真实的仿真结果。

我曾经做过一个电力质量分析的项目，要求测试分析电网各相电压、电流中的 25 次全部谐波含量（THD）及各次谐波含量的大小。同时，还要求对各次谐波含量以百分数和柱状图形分别显示出来。使用 LabVIEW 提供的例程，略加修改很快就完成了，真不知道如果使用其它编程语言会搞到什么时候。

2.6.5 灵活、开放的平台



LabVIEW 还是一个开放的开发平台，提供广泛的软件集成工具、运行库和文件格式，可以方便的与第三方设计和仿真连接，例如：

DLL、共享库

ActiveX、COM 和 .NET (微软)

DDE、TCP/IP、UDP、以太网、蓝牙

CAN、DeviceNet、Modbus、OPC

高速 USB、IEEE1394、GPIB、RS232/485

数据库 (ADO、SQL 等)

对于便携式及嵌入式开发：

LabVIEW PDA 支持便携式手持系统 PDA(个人数字处理器) 的开发应用，还支持 Pocket PC OSs 及 Windows CE。使用 LabVIEW 可以创建自定义的便携式测试分系统。

LabVIEW 嵌入式开发模块还支持对 32 位处理器的图形化开发，这应该是一个很了不起的创举。目前支持的目标处理器有：PowerPC、ARM、TI C6xx86 架构；支持的嵌入式操作系统如：VxWorks、eCos、Windows 和嵌入式的 Linux。

LabVIEW DSP 工具包还支持 TI 的 DSP 设计开发。

LabVIEW FPGA 模块还支持 FPGA 设计。丰富了 RIO 系列模块的自定义功能。

这些特点完全可以使你始终保持着在技术上绝对领先于你的竞争者。

不管上述说法能否成为说服你的理由，从我个人的学习经历和体会来看，有机会还是值得一试的。

世界上好多东西都可以给我们带来快乐和享受，比如：汽车、高清电视、美食等等，但是在我的工作中能够为我带来快乐和享受的就是学习 LabVIEW 的过程。



第 2.7 节 LabVIEW 的受众面



在这个世界上，到底会有多少人在学习 LabVIEW，估计没有人能够回答得出。记得是几年前，在 OpenG 的网站看到过一篇报道，是一个普查机构关于计算机语言使用者排行的调查报告，LabVIEW 的排名在 34 位（？）（具体的排位记不太清楚了）。2008 年 4 月看到一份资料，称 LabVIEW 的排名在第 31 位。前三名为：Java、C、VB。

那么，在中国到底有多少人学习过 LabVIEW，估计也不会有人能知道。大学中又有多少学校开设了 LabVIEW 课程？大概也是一个未知数。最近几年的里，在每年的上半年学生开始进行毕业设计时，在与 LabVIEW 有关的网站或论坛上，到处都可以看到什么“跪求”、“救命”、“帮忙”等帖子，询求 LabVIEW 项目的设计资料，说明在大校还是 LabVIEW 学习较为集中的地方。

记得在一次 NI 的招待会上，我看见两个民办中等技术学校的老师，我非常好奇便问到：“你们学校也开 LabVIEW 课程”？他们很坦诚的告诉我：“这样的学生将来比较好就业”。当然，现在学习了，今后就有可能使用到。

2005 年赴上海 NI 学习模块化仪器课程，恰有 20 多年没见面的大学同学在南京，乘车路过南京时约他在南京火车站站台见面，火车停稳后，见他手拿 NI 的白色手提袋站在站台上向我招手，心中很迷惑，因为他不从事技术工作，详细了解后方知：也想了解一下虚拟仪器技术。



在本《札记》的“序”中，曾提到每年帮助过我的 NI AE，基本上都是新人。这也足可以看出 NI 在中国的发展速度是相当快的。

我的“博客”开始是在 EDNChina 的网站上，我深深地知道，由于资金、项目等原因，目前学习、使用 LabVIEW 的人还不是很多；所以，关注这方面内容的人一定很少。令我万万没有想到的是，100 天来我的“博客”竟有了 15200 次的点击；我的日志：“学习 LabVIEW 的理由（1）”和“数据采集系统设计原则（1）”单篇日志的点击率达到 975 和 846。

其实，LabVIEW 的受众面取决于它的应用范围和应用程度，Jeff Kodosky 老先生（我非常尊敬和崇拜他）曾设想 LabVIEW 会成为“通用编程语言”。LabVIEW 真的会成为一种通用的编程语言吗？

我个人认为，LabVIEW 成为一种通用的编程语言还有相当长的一段距离，至少目前是这样。LabVIEW 成为一种通用的编程语言是 Jeff Kodosky 老先生为他的“虚拟仪器”王国的发展或者说未来的发展制定了一个理想。

我认为，LabVIEW 是一种最适合测量和自动化的应用领域的应用程序的开发环境（或者也可以称为：编程语言）。它最大的贡献在于为那些对使用通用编程语言感到陌生和困惑的测试、测量工程师提供了一种简介、快速的图像化编程方法，以此来帮助和提高自动化测量能力。通过多年的学习、使用我深深地感觉到它的方便、快捷能力，使我的工作受益匪浅，套用一句小品的台词：“谁用谁知道”。

使用 LabVIEW 可以创建应用程序（非测试、测量自动化应用程序）吗？回答应该是肯定的：可以。我的同事就使用 LabVIEW 编制了一个很好的游戏程序及做了一个题库自动生成模拟试题等应用程序。

那什么因素导致 LabVIEW 还不能成为一种通用的编程语言呢？

根据我的实践看有以下几个因素：

1、程序的解读性差 我们知道“语言”是用来交流的一种工具。LabVIEW 想成为一种通用的编程语言也不可能例外。而实际使用过程中，特别是在程序较大时，我们很难读懂别人设计的 VI，或者说充分了解设计者的思想。

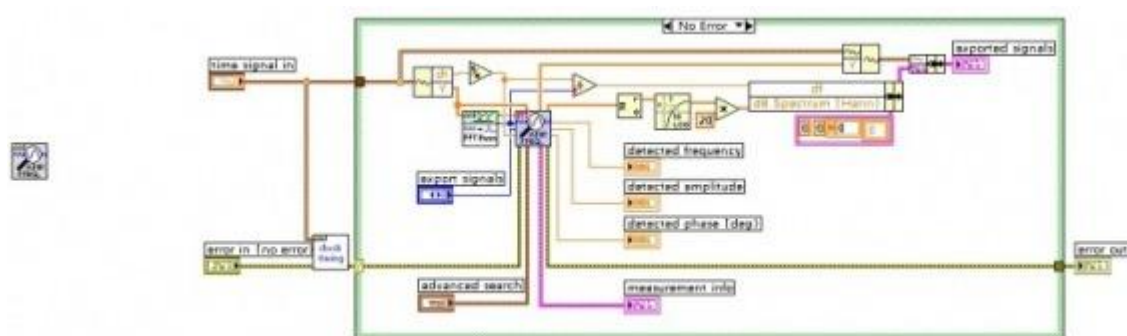
2、程序档案管理能力差 一个应用程序的文档是要给很多人看的，这包括设计者、使用者和维护者，也必须能够很方便的打印、存档。而 LabVIEW 程序框图的打印效果非常不理想，这给程序的维护 and 修改带来了不便。比如一个稍微复杂点的状态机程序框图，打印下来就要有很多分支图，如果是放到《札记》中，必然造成篇幅很大。在写这本《札记》的过程中，我就遇到了这样的麻烦。我认为这是一个致命的弱点（可能是图形化的原因）。

3、使用的广泛性 由于 LabVIEW 设计的初衷就是面对那些测试、测量工程师，可能反而限制了其它使用者的使用，所以应用的广泛性必然会受到一定的影响。另外本地化过程太慢、价格等因素都是影响它成为一种通用的编程语言的障碍。

这些都是我个人的观点，实际上最近几年 LabVIEW 一直在试图扩大它的应用领域和影响面，这包括：与乐高公司合作开发图形化的机器人编程系统（从娃娃抓起？）；ARM 嵌入式处理器的图形化开发；FPGA 的图形化开发、电子仿真、OOP 设计思想的引入等等。

特别值得一提的是，NI 许多应用程序的设计，也来自他们用 LabVIEW 进行设计实现，当然也包括许多 VI 的设计。下面我们举两个例子来说明。

例 1、 Extract Single Tone Information VI



图中的左边是“Extract Single Tone Information VI”的图标，右边是该 VI 的程序框图。显然这个 VI 是用 LabVIEW 语言设计的。在 LabVIEW 提供的函数中，这样的 VI 有很多。

例 2、DMM 应用程序



NI 4070 DMM 的应用程序前面版图，它清晰的写着：Powered by LabVIEW。

Jeff Kodosky 老先生不久将会看到了，由他率领、指挥、驾驶的 LabVIEW 战舰正驶向“通用编程语言”的海港。当越来越多的由 LabVIEW 制造的 LabVIEW 程序出现时，离他的远大目标还会远吗？

Jeff Kodosky 老先生对此现象的言谈被翻译成“自食其力”，到底是动物界的“克隆”，还是生物界的“无性繁殖”，还是人们所说的“自食其力”，这些还是由专家们去定义吧！LabVIEW 快速的发展、功能越来越强大则是不争的事实。

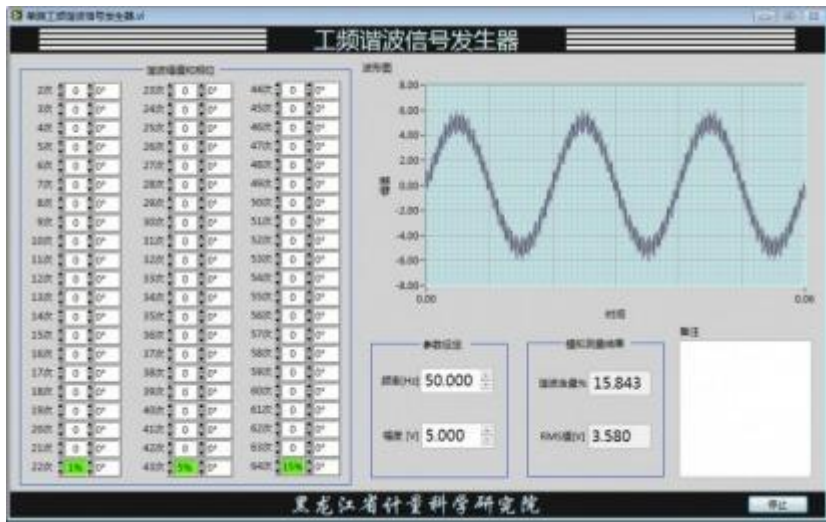
根据 2008 年 11 月份调查 LabVIEW 在语言排行榜已升至第 29 位

| Position | Programming Language | Ratings |
|----------|----------------------|---------|
| 21 | Lisp/Scheme | 0.470% |
| 22 | MATLAB | 0.466% |
| 23 | Ada | 0.410% |
| 24 | Fortran | 0.380% |
| 25 | FoxPro/xBase | 0.320% |
| 26 | Prolog | 0.314% |
| 27 | RPG (OS/400) | 0.298% |
| 28 | Awk | 0.256% |
| 29 | LabVIEW | 0.235% |
| 30 | Tcl/Tk | 0.230% |
| 31 | Erlang | 0.220% |
| 32 | Bourne shell | 0.220% |
| 33 | Caml | 0.196% |
| 34 | Alice | 0.188% |
| 35 | PL/I | 0.188% |
| 36 | Haskell | 0.163% |
| 37 | NXT-G | 0.153% |
| 38 | Objective-C | 0.149% |
| 39 | Smalltalk | 0.148% |
| 40 | PowerShell | 0.143% |
| 41 | Groovy | 0.138% |
| 42 | ML | 0.132% |
| 43 | Scala | 0.120% |
| 44 | REXX | 0.119% |

前三名依旧为：Java、C、VB。

第 2.8 节 虚拟仪器开发模式之我见

2.8.1 商品型的系统集成模式



在此之前，我毫不吝啬地使用大量的篇幅讲述了 LabVIEW 殿堂的美好之处，并大谈什么学习的理由、应该看的几本书等等。下面的内容，我们将探讨一下虚拟仪器的最佳开发模式是什么？

我们知道，虚拟仪器的缔造者 NI 公司只提供各种虚拟仪器开发软件平台和相关的硬件体系模块。大型的应用实例或应用项目都是由虚拟仪器系统集成商来设计开发完成的，而像我这样的虚拟仪器爱好者只能做本公司或本单位的一些小项目。

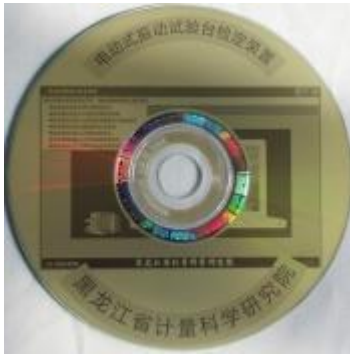
这样的开发方式实际上是制约和阻碍了虚拟仪器技术的应用范围和应用层面，也不利于虚拟仪器应用的整体提高。我们所面对的，实质就是这样的问题：虚拟仪器的产业化方式是什么？虚拟仪器系统集成是否是唯一的方式？商业化或商品化的虚拟仪器开发模式到底应该是什么样的？我们这些爱好者在虚拟仪器开发领域还有多大的发展空间？在中国的企业中，谁最需要、渴望获得这种技术上的帮助来实现企业的发展和技术进步？

其实，最简单的道理就是：市场需求决定开发方式。那么接下来的问题是：虚拟仪器的市场在那里？会是谁？这也是几年来，我一直在苦苦思考的问题之一，并在虚拟仪器应用开发实践中，不断地深入探索、寻求着正确的答案。

我采用虚拟仪器技术设计了基于 USB 接口的《振动试验台检定装置》、《便携式应力测量仪》都已经完成提交给了用户，从用户的反应来看，应该说是不错，也拿到了订单。假如：订单源源不断，我只需订购数据采集卡和笔记本电脑，然后不断的复制光盘，似乎一个人就可以完成这些工作，这种虚拟仪器开发方式你认为“爽”吗？当然“爽”了！我也没有半点不满意的意思，这是一个非常好的虚拟仪器开发方式。但我认为这还不是虚拟仪器的最佳开发模式，为什么？

我们能保证订单源源不断吗？

2.8.2 租赁型的系统集成模式



我还真说不清楚到底是为了什么，只是隐约的感觉到是否又回倒商业化仪器的设计、生产模式，只不过是采用图形化开发和按用户的要求自定义了测试功能。无论如何，我还是感觉没有把虚拟仪器的特点发挥到最佳状态。回顾我们前面提到的虚拟仪器的四个特点，就会发现“强大的运行环境“做到了、”自定义测量功能“做到 了、”强大的分析处理能力“做到了、唯独“再生（复用）”这个特点没有充分发挥出来。如何发挥出这个特点呢？

我还真想出一个好办法，就是：虚拟仪器系统集成租赁业务（但我目前没有资金和技术能力和机会来实践）。

业务实质是：按用户的要求设计虚拟仪器，用户可以一次性买断（买走），也可以是采用租赁的方式以年、月、日为期租赁使用，甚至允许以使用的“次数”来租赁（要有严格的法律保障），租赁期间或租赁期后也可以一次买断。

这样做的好处是：

1、虚拟仪器的特点得到充分发挥、利用

在租赁期满后，硬件只要没有问题，还可以用来为其它用户重新设计使用。提高硬件的利用率的同时，可以降低所有用户成本。

2、降低用户的前期投入和风险

在用户的新产品研发阶段最适合，因为这样可以大大降低用户的前期投入和风险。

3、系统集成商可以大大的增加业务量（特别是像我们这样较小的系统集成商）

在用户投入不多的情况下，一定会吸引更多的用户选择这种业务方式。

4、对于中小企业用户最适合

中小企业用户在市场经济中，面临最多的就是产品落后，资金、技术投入不足，抵抗风险的能力弱。这种业务方式可以降低企业在产品升级中所遇到的各种风险。

5、适合于网络经营

用户只要谈清楚需求，就可以进行设计开发直至最终的验收。这一切都可以通过互联网来进行。

这种业务模式，只要计算好设备折旧、租赁价格及资料管理，合同公证等环节，对虚拟仪器应用

来讲，应该说是很好的一个非常好经营模式，应该说是一个多“赢”的方案。

这个想法源于我了解到的两个项目的投资案例，这两案例的投资人都是较为成功的民营企业，一个是商品经营商，一个是低端电子产品生产商。它们共同的特点就是想在电子仪器领域开辟新的商机，同时又都对电子仪器没有更多的了解。方式都是采用聘用大学的博士或事业单位的工程师来业余设计开发电子类产品。结果因为市场发生变化或合作的不愉快，对继续投入没有信心而导致失败。结果是对方一撤，一个投入三十万元，只剩下一堆自己不认识，也不会使用的电子设备。另一个投入四十万，也只落下一堆没人会使用、也用不上的电子仪器。

我对这两个案例进行了技术分析，得出的结论是：如果采用虚拟仪器租赁方式，损失大致可以降低一半以上。至少是看不到那些卖也卖不出价格的电子仪器，这些设备都占到了整个投资的一半，或许采用这种虚拟仪器租赁的方式，项目合作就可能研发成功。

这是我个人的一些体会，我相信是虚拟仪器市场化非常有效的方案，因为投资和风险并不是很大。可是我因为没有资金和机会来实践，所以也就不会有成功的案例在这里做支持。

如果以后真有人这样做，并获得了成功，千万别忘了分配给我一些股份呀！
我现在有了自己的网页，完全可以尝试着开展这样的业务。



第 2.9 章 本章小结

以我的视角和近一章的篇幅介绍了我眼中的 LabVIEW，其中有一部分内容是来自我的博客，并略加做了整理。这些内容都是来自我真实的体会和感受。随着岁月的流逝，我会不断地整理、补充和充实这些内容，并将最新的体会和感受传递给大家。

毕竟我认识、学习 LabVIEW 才仅仅四年，基础并非很扎实，理解的可能很不准确。LabVIEW 是一种图形化的编程语言，那么就一定有它自身的特点和特性，下一章将针对这方面的内容进行介绍。

