

## 目 录

1	概述.....	1
2	课题简介及意义.....	2
2.1	设计时钟意义.....	2
2.2	时钟简介.....	2
3	虚拟仪器概述.....	3
4	LabVIEW 简介.....	4
4.1	LabVIEW 的运行机制.....	4
4.1.1	LabVIEW 应用程序的构成.....	4
4.1.2	LabVIEW 的操作模板.....	6
4.2	LabVIEW 的具体操作.....	11
4.2.1	显示对象 (Indicator)、控制对象 (Control) 和数值常数对象....	11
4.2.2	关于连线.....	11
5	虚拟时钟系统设计的实现.....	12
5.1	总体设计.....	12
5.2	子 vi 的相关介绍.....	14
5.3	功能及实现.....	15
5.3.1	获得系统时间.....	15
5.3.2	时、分、秒的获取.....	16
5.3.3	数据的运算.....	16
5.3.4	记录坐标变换.....	18
5.3.5	图像的绘制.....	19
5.3.6	While 循环实现秒针的跳变.....	20
5.3.7	程序结构介绍.....	21
5.3.8	图像采集与图像处理.....	22
5.3.9	程序设计总体.....	24
6.	结束语.....	25
	参考文献.....	26
	致 谢.....	27

## 1 概述

随着科学技术的快速发展，各种功能的软件的都得到迅速的开发与应用。虚拟仪器成为计算机技术和仪器科学领域完美结合的产物，代表了仪器仪表的发展方向。LabVIEW作为虚拟仪器开发的平台，是一个具有革命性的图形化开发环境，在工业测量和控制领域中掀起了一场变革。它具有功能强大、编程灵活、人机界面友好的特点，在测量技术与仪器工程科学领域中得到了非常广泛的应用。

本文基于 LabVIEW 软件，设计一个虚拟的时钟程序，使之在桌面可以直接显示，形象、直观、方便。通过对本设计的研究应该能够比较熟练的掌握 Labview 软件的使用，并能在此平台上进行应用程序的开发。下面我将由时钟开始，对虚拟仪器、LabVIEW 以及整个设计做详细的介绍。

## 2 课题简介及意义

### 2.1 设计时钟意义

时钟，自从它被发明的那天起，就成为人们生活中必不可少的一种工具，尤其是在现在这个讲究效率的年代，时钟更是在人类生产、生活、学习等多个领域得到广泛的应用。然而随着时间的推移，人们不仅对于时钟精度的要求越来越高，而且对于时钟功能的要求也越来越多，时钟已不仅仅是一种用来显示时间的工具，在很多实际应用中它还需要能够实现更多其它的功能。诸如闹钟功能、日历显示功能、温度测量功能、湿度测量功能、电压测量功能、频率测量功能、过欠压报警功能等。钟表的数字化给人们的生产生活带来了极大的方便，而且大大地扩展了钟表原先的报时功能。诸如定时自动报警、按时自动打铃、时间程序自动控制、定时广播、自动起闭路灯、定时开关烘箱、通断动力设备、甚至各种定时电气的自动启用等，所有这些，都是以钟表数字化为基础的。可以说，设计多功能数字时钟的意义已不只在数字时钟本身，更大的意义在于多功能数字时钟在许多实时控制系统中的应用。在很多实际应用中，只要对数字时钟的程序和硬件电路加以一定的修改，便可以得到实时控制的实用系统，从而应用到实际工作与生产中去。因此，研究数字时钟及扩大其应用，有着非常现实的意义。

怎样让时钟更好的为我们服务？怎样让时钟更符合实际应用的需求？这就要求人们不断设计出新型时钟，不断设计出适合实际应用的多功能时钟。

### 2.2 时钟简介

随着人类科技文明的发展，人们对于时钟的要求在不断地提高。时钟已不仅仅被看成一种用来显示时间的工具，在很多实际应用中它还需要能够实现更多其它的功能。高精度、多功能、小体积、低功耗，是现代时钟发展的趋势。在这种趋势下，时钟的数字化、多功能化已经成为现代时钟生产研究的主导设计方向。

设计过程采用系统设计的方法，先分析任务，得到系统要求，然后进行总体设计，划分子系统，然后进行详细设计，决定各个功能子系统内部的电路，最后进行测试。

时钟的设计有很多种实现形式。它可以通过单片机接口来实现，也可以通过硬件电路，运用各种芯片实现。当然还可以用许多硬件语言 VHDL、软件语言 C、JAVA 等编程来实现。虚拟仪器 LabVIEW 软件，是一个具有革命性的图形化开发环境，内置信号采集、测量分析与数据显示功能，摒弃了传统开发工具的复杂性，从简单的仪器控制、数据采集到过程控制和工业自动化系统，它都得到了广泛的应用。出于它功能的强大性，本文通过 LabVIEW 开发的程序来完成虚拟时钟的设计。

后文将先对虚拟仪器及 LabVIEW 软件进行介绍。

### 3 虚拟仪器概述

虚拟仪器（virtual instrumentation）是基于计算机的仪器。计算机和仪器的密切结合是目前仪器发展的一个重要方向。粗略地说这种结合有两种方式，一种是将计算机装入仪器，其典型的例子就是所谓智能化的仪器。随着计算机功能的日益强大以及其体积的日趋缩小，这类仪器功能也越来越强大，目前已经出现含嵌入式系统的仪器。另一种方式是将仪器装入计算机。以通用的计算机硬件及操作系统为依托，实现各种仪器功能。虚拟仪器主要是指这种方式。下面的框图反映了常见的虚拟仪器方案。

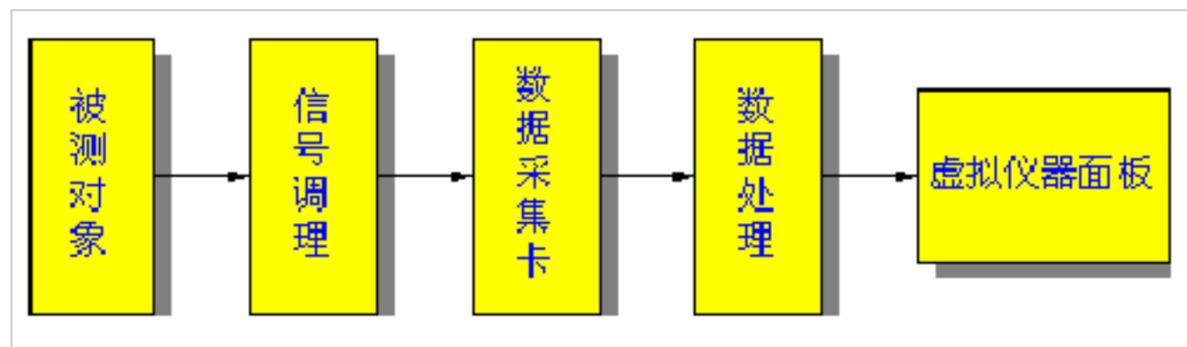


图 3.1

虚拟仪器的主要特点有：

尽可能采用了通用的硬件，各种仪器的差异主要是软件。

可充分发挥计算机的能力，有强大的数据处理功能，可创造功能更强的仪器。

用户可以根据自己的需要定义和制造各种仪器。

虚拟仪器实际上是一个按照仪器需求组织的数据采集系统。虚拟仪器的研究中涉及的基础理论主要有计算机数据采集和数字信号处理。目前在这一领域内，使用较为广泛的计算机语言是美国 NI 公司的 LabVIEW。虚拟仪器的起源可以追溯到 20 世纪 70 年代，那时计算机测控系统在国防、航天等领域已经有了相当的发展。PC 机出现以后，仪器级的计算机化成为可能，甚至在 Microsoft 公司的 Windows 诞生之前，NI 公司已经在 Macintosh 计算机上推出了 LabVIEW2.0 以前的版本。对虚拟仪器和 LabVIEW 长期、系统、有效的研究开发使得该公司成为业界公认的权威。

普通的 PC 有一些不可避免的弱点。用它构建的虚拟仪器或计算机测试系统性能不可能太高。目前作为计算机化仪器的一个重要发展方向是制定了 VXI 标准，这是一种插卡式的仪器。每一种仪器是一个插卡，为了保证仪器的性能，又采用了较多的硬件，但这些卡式仪器本身都没有面板，其面板仍然用虚拟的方式在计算机屏幕上出现。这些卡插入标准的 VXI 机箱，再与计算机相连，就组成了一个测试系统。VXI 仪器价格昂贵，目前又推出了一种较为便宜的 PXI 标准仪器。

虚拟仪器研究的另一个问题是各种标准仪器的互连及与计算机的连接。目前使用较多的是 IEEE 488 或 GPIB 协议。未来的仪器也应当是网络化的。

## 4 LabVIEW 简介

LabVIEW (Laboratory Virtual instrument Engineering) 是一种图形化的编程语言，它广泛地被工业界、学术界和研究实验室所接受，视为一个标准的数据采集和仪器控制软件。LabVIEW 集成了与满足 GPIB、VXI、RS-232 和 RS-485 协议的硬件及数据采集卡通讯的全部功能。它还内置了便于应用 TCP/IP、ActiveX 等软件标准的库函数。这是一个功能强大且灵活的软件。利用它可以方便地建立自己的虚拟仪器，其图形化的界面使得编程及使用过程都生动有趣。

图形化的程序语言，又称为“G”语言。使用这种语言编程时，基本上不写程序代码，取而代之的是流程图或流程图。它尽可能利用了技术人员、科学家、工程师所熟悉的术语、图标和概念，因此，LabVIEW 是一个面向最终用户的工具。它可以增强你构建自己的科学和工程系统的能力，提供了实现仪器编程和数据采集系统的便捷途径。使用它进行原理研究、设计、测试并实现仪器系统时，可以大大提高工作效率。

利用 LabVIEW，可产生独立运行的可执行文件，它是一个真正的 32 位编译器。像许多重要的软件一样，LabVIEW 提供了 Windows、UNIX、Linux、Macintosh 的多种版本。

### 4.1 LabVIEW 的运行机制

#### 4.1.1 LabVIEW 应用程序的构成

所有的 LabVIEW 应用程序，即虚拟仪器 (VI)，它包括前面板 (front panel)、流程图 (block diagram) 以及图标/连接器 (icon/connector) 三部分。

##### 1. 前面板

使用输入控制和输出显示来构成。控制是用户输入数据到程序的接口。而显示是输出程序产生的数据接口。控制和显示有许多种类，可以从控制模板的各个子模板中选取。

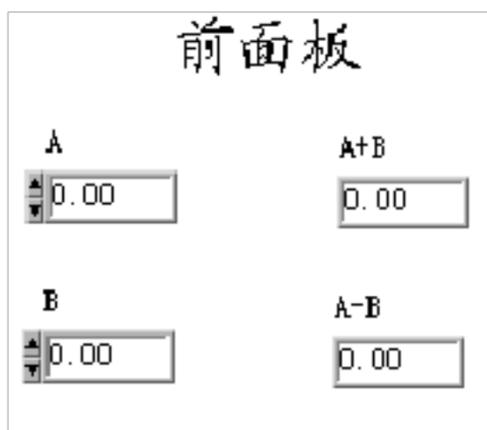


图 4.1 前面板原理图

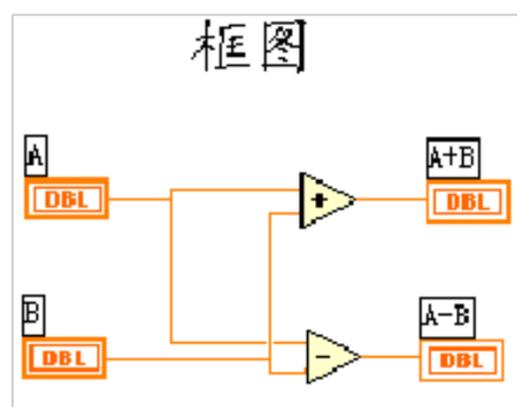


图 4.2 框图结构

两种最常用的前面板对象是数字控制和数字显示。若想要在数字控制中输入或修改数值，你只需要用操作工具（见工具模板）点击控制部件和增减按钮，或者用操作工具或标签工具双击数值栏进行输入数值修。

##### 2. 框图程序

框图程序是由节点、端点、图框和连线四种元素构成的。

节点类似于文本语言程序的语句、函数或者子程序。LabVIEW 有二种节点类型——函数节点和子 VI 节点。两者的区别在于：函数节点是 LabVIEW 以编译好了的机器代码供用户使用的，而子 VI 节点是以图形语言形式提供给用户的。用户可以访问和修改任一子 VI 节点的代码，但无法对函数节点进行修改。上面的框图程序所示的 VI 程序有两个功能函数节点，一个函数使两个数值相加，另一个函数使两数相减。

端点是只有一路输入/输出，且方向固定的节点。LabVIEW 有三类端点——前面板对象端点、全局与局部变量端点和常量端点。对象端点是数据在框图程序部分和前面板之间传输的接口。一般来说，一个 VI 的前面板上的对象（控制或显示）都在框图中有一个对象端点与之一一对应。当在前面板创建或删除面板对象时，可以自动创建或删除相应的对象端点。控制对象对应的端点在框图中是用粗框框住的，如例子中的 A 和 B 端点。它们只能在 VI 程序框图中作为数据流源点。显示对象对应的端点在框图中是用细框框住的。如例子中的 A+B 和 A-B 端点。它们只能在 VI 程序框图中作为数据流终点。常量端点永远只能在 VI 程序框图中作为数据流源点。

图框是 LabVIEW 实现程序结构控制命令的图形表示。如循环控制、条件分支控制和顺序控制等，编程人员可以使用它们控制 VI 程序的执行方式。代码接口节点（CIN）是框图程序与用户提供的 C 语言文本程序的接口。

连线是端口间的数据通道。它们类似于普通程序中的变量。数据是单向流动的，从源端口向一个或多个目的端口流动。不同的线型代表不同的数据类型。在彩显上，每种数据类型还以不同的颜色予以强调。

当需要连接两个端点时，在第一个端点上点击连线工具（从工具模板栏调用），然后移动到另一个端点，再点击第二个端点。端点的先后次序不影响数据流动的方向。

当把连线工具放在端点上时，该端点区域将会闪烁，表示连线将会接通该端点。当把连线工具从一个端口接到另一个端口时，不需要按住鼠标键。当需要连线转弯时，点击一次鼠标键，即可以正交垂直方向地弯曲连线，按空格键可以改变转角的方向。

快速提示：

接线头是为了帮助正确连接端口的连线。当把连线工具放到端口

### 3. 从框图程序窗口创建前面板对象

用选择和连线工具，你都可以用鼠标右键点击任一节点和端点，然后从弹出菜单中选择“创建常数”，“创建控制”，或“创建显示”等命令。LabVIEW 会自动地在被创建的端点与所点击对象之间接好连线。

### 4. 数据流编程

控制 VI 程序的运行方式叫做“数据流”。对一个节点而言，只有当它的所有输入端口上的数据都成为有效数据时，它才能被执行。当节点程序运行完毕后，它把结果数据送给所有的输出端口，使之成为有效数据。并且数据很快从源送到目的端口。

如图 4.3 所示，这个 VI 程序把两个输入数值相乘，再把乘积减去 50.0。这个程序中，框图程序从左往右执行，这个执行次序不是由于对象的摆放位置，而是由于相减运算函数的一个输入量是相乘函数的运算结果，它只有当相乘运算完成并把结果送到减运算的输入口后才能继续下去。请记住，一个节点（函数）只有当它所有的输入端的数据都成为有效数据后才能被执行，而且只有当它执行完成后，它的所有输出端口上的数据才成为有效。

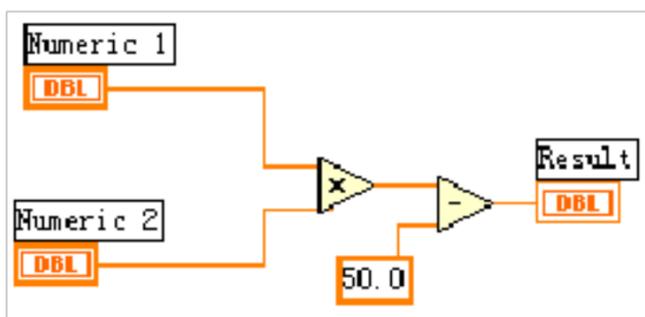


图 4.3

再看另一个程序(如图 4.4 所示)，哪一个节点函数将先执行，是乘法还是除法，在这个例子中，我们无法知道哪一个节点函数首先执行，因为所有输入量几乎同时到达。对于这样一种相互独立的数据流程，如果又必须明确指定节点执行的先后次序，就必须使用顺序（Sequence）结构来明确执行次序。

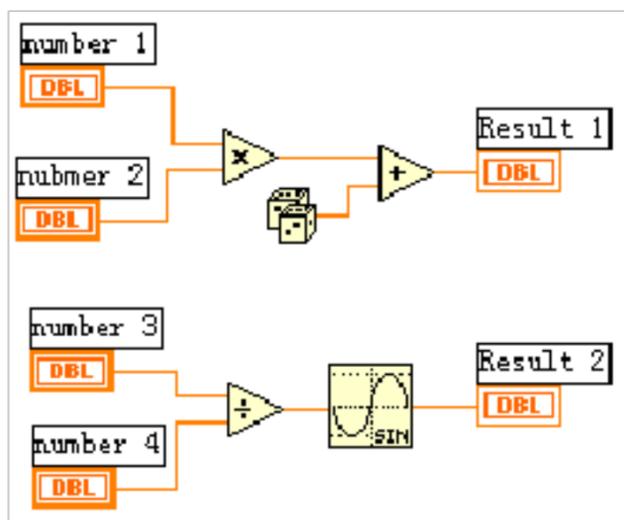


图 4.4

## 4.1.2 LabVIEW 的操作模板

在 LabVIEW 的用户界面上，应特别注意它提供的操作模板，包括工具（Tools）模板、控制（Controls）模板和函数（Functions）模板。这些模板集中反映了该软件的功能与特征。下面我们来大致浏览一下。

### 1. 工具模板（Tools Palette）

该模板（如图 4.5 所示）提供了各种用于创建、修改和调试 VI 程序的工具。如果

## 武汉科技大学本科毕业设计

该模板没有出现，则可以在 Windows 菜单下选择 Show Tools Palette 命令以显示该模板。当从模板内选择了任一种工具后，鼠标箭头就会变成该工具相应的形状。当从 Windows 菜单下选择了 Show Help Window 功能后，把工具模板内选定的任一种工具光标放在流程图程序的子程序 (Sub VI) 或图标上，就会显示相应的帮助信息。



图 4.5

下述工具中注意 1 和 2 的区别，2 用于编程时，1 用于运行程序时。4 是一个特有的工具，它并不是一个简单的画线工具，而是一个符合 LabVIEW 语言规定的对象连接工具。

工具图标有如下几种：

表 4.1

图标	名称	功能
	值)	控制中键入值时，工具会变成标签工具
	Position/Size/Select (选择)	用于选择、移动或改变对象的大小。当它用于改变对象的连框大小时，会变成相应形状。
	Edit Text (编辑文本)	用于输入标签文本或者创建自由标签。当创建自由标签时它会变成相应形状。
	Connect Wire (连线)	用于在流程图程序上连接对象。如果联机帮助的窗口被打开时，把该工具放在任一条连线上，就会显示相应的数据类型。
	Object Shortcut Menu (对象菜单)	用鼠标左键可以弹出对象的弹出式菜单。
	Scroll Windows (窗口漫游)	使用该工具就可以不需要使用滚动条而在窗口中漫游。
	Set/Clear Breakpoint (断点设置 / 清除)	使用该工具在 VI 的流程图对象上设置断点。
	Probe Data (数据探针)	可在框图程序内的数据流线上设置探针。通过控针窗口来观察该数据流线上的数据变化状况。
	Get Color (颜色提取)	使用该工具来提取颜色用于编辑其他的对象。
	Set Color (颜色设置)	用来给对象定义颜色。它也显示出对象的前景色和背景色。

下面的两个模板是多层的，其中每一个子模板下还包括多个对象。

## 2. 控制模板 (Control Palette)

注意：只有打开前面板时才能调用该模板。

该模板用来给前面板设置各种所需的输出显示对象和输入控制对象。每个图标代表一类子模板。如果控制模板不显示，可以用 Windows 菜单的 Show Controls Palette 功能打开它，也可以在前面板的空白处，点击鼠标右键，以弹出控制模板。

控制模板如图 4.6 所示，它包括如下所示的一些子模板。子模板中包括的对象，我们在功能中用文字简要介绍。



表 4.2

图 4.6

图标	子模板名称	功能
	Boolean (布尔量)	逻辑数值的控制和显示。包含各种布尔开关、按钮以及指示灯等。
	String & Path (字符串和路径)	字符串和路径的控制和显示。
	Array & Cluster (数组和簇)	数组和簇的控制和显示。
	List & Table (列表和表格)	列表和表格的控制和显示
	Graph (图形显示)	显示数据结果的趋势图和曲线图。
	Ring & Enum (环与枚举)	环与枚举的控制和显示。
	I/O (输入/输出功能)	输入/输出功能。于操作 OLE、ActiveX 等功能。
	Refnum	参考数
	Digilog Controls (数字控制)	数字控制
	Classic Controls (经典控制)	经典控制，指以前版本软件的面板图标。
	Activex	用于 ActiveX 等功能。

# 武汉科技大学本科毕业设计

13		Decorations (装饰)	用于给前面板进行装饰的各种图形对象。
14		Select a Controls (控制选择)	调用存储在文件中的控制和显示的接口。
15		User Controls (用户控制)	用户自定义的控制和显示。

### 3. 功能模板 (Functions Palette)

注：只有打开了流程图程序窗口，才能出现功能模板。

功能模板是创建流程图程序的工具。该模板上的每一个顶层图标都表示一个子模板。若功能模板不出现，则可以用 Windows 菜单下的 Show Functions Palette 功能打开它，也可以在流程图程序窗口的空白处点击鼠标右键以弹出功能模板。

功能模板如图 4.7 所示，其子模块如下所示。个别不常用的子模块未包含。

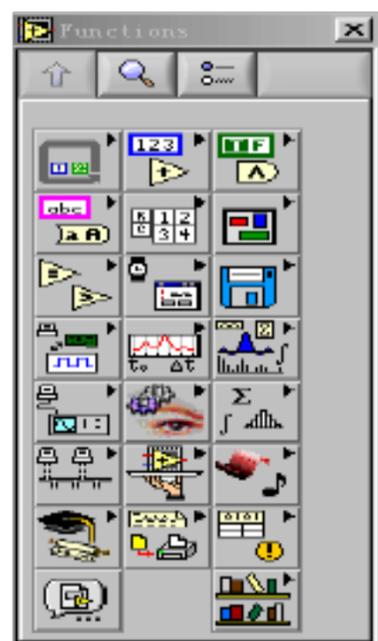


图 4.7

表 4.3

图标	子模板名称	功能
	Numeric (数值运算)	包括各种常用的数值运算，还包括数制转换、三角函数、对数、复数等运算，以及各种数值常数。
	Boolean (布尔运算)	包括各种逻辑运算符以及布尔常数。
	String (字符串运算)	包含各种字符串操作函数、数值与字符串之间的转换函数，以及字符(串)常数等。
	Array (数组)	包括数组运算函数、数组转换函数，以及常数数组等。
	Cluster (簇)	包括簇的处理函数，以及群常数等。这里的群相当于 C 语言中的结构。
	Comparison (比较)	包括各种比较运算函数，如大于、小于、等于。
	Time & Dialog (时间和对话框)	包括对话框窗口、时间和出错处理函数等。

# 武汉科技大学本科毕业设计

LabVIEW 用户子 VI 库			
1 0		Data Acquisition (数据采集)	包括数据采集卡的驱动, 以及信号调理所需的各种功能模块。
1 1		Waveform (波形)	各种波形处理工具
1 2		Analyze (分析)	信号发生、时域及频域分析功能模块及数学工具。
1 3		Instrument I/O (仪器输入/输出)	包括 GPIB(488、488.2)、串行、VXI 仪器控制的程序和函数, 以及 VISA 的操作功能函数。
1 4		Motion & Vision (运动与景像)	
1 5		Mathematics (数学)	包括统计、曲线拟合、公式框节点等功能模块, 以及数值微分、积分等数值计算工具模块。
1 6		Communication (通讯)	包括 TCP、DDE、ActiveX 和 OLE 等功能的处理模块。
1 7		Application Control (应用控制)	包括动态调用 VI、标准可执行程序的功能函数。
1 8		Graphics & Sound (图形与声音)	包括 3D、OpenGL、声音播放等功能模块。包括调用动态连接库和 CIN 节点等功能的处理模块。
1 9		Tutorial (示教课程)	包括 LabVIEW 示教程序。
2 0		Report Generation (文档生成)	
2 1		Advanced (高级功能)	
2 2		Select a VI (选择子 VI)	
2 3		User Library (用户子 VI 库)	

## 4.2 LabVIEW 的具体操作

### 4.2.1 显示对象 (Indicator)、控制对象 (Control) 和数值常数对象

显示对象和控制对象都是前面板上的控件，前者有输入端子而无输出端子，后者正好相反，它们分别相当于普通编程语言中的输出参数和输入参数。数值常数对象可以看成是控制对象的一个特例。

在前面板中创建新的控制对象或显示对象时，LabVIEW 都会在流程图中创建对应的端子。端子的符号反映该对象的数据类型。例如，DBL 符号表示对象数据类型是双精度数；TF 符号表示布尔数；I16 符号表示 16 位整型数；ABC 符号表示对象数据类型是字符串。

一个对象应当是显示对象还是控制对象必须弄清楚，否则无法正确连线。有时他们的图标是相似或相同的，可以根据需要明确规定它是显示对象还是控制对象。方法是将鼠标移到图标上，然后点右键，可出现快速菜单（例见右图）。如果菜单中的第一项是 Chang to Control,说明这是一个显示对象，你可以根据需要，将其变为控制对象。如果菜单中的第一项是 Chang to Indicator ,说明这是一个控制对象，你也可以根据需要，将其变为显示对象。

控制对象和显示对象都不能在流程图中删除，只能从前面板上删除。

### 4.2.2 关于连线

连线是程序设计中较为复杂的问题。流程图上的每一个对象都带有自己的连线端子，连线将构成对象之间的数据通道。因为这不是几何意义上的连线，因此并非任意两个端子间都可连线，连线类似于普通程序中的变量。数据单向流动，从源端口向一个或多个目的端口流动。不同的线型代表不同的数据类型。下面是一些常用数据类型所对应的线型和颜色：

类型	颜色	标量	一维数组	二维数组
整形数	蓝色			
浮点数	橙色			
逻辑量	绿色			
字符串	粉色			
文件路径	青色			

当需要连接两个端点时，在第一个端点上点击连线工具（从工具模板栏调用），然后移动到另一个端点，再点击第二个端点。端点的先后次序不影响数据流动的方向。

当把连线工具放在端点上时，该端点区域将会闪烁，表示连线将会接通该端点。当把连线工具从一个端口接到另一个端口时，不需要按住鼠标键。当需要连线转弯时，点击一次鼠标键，即可以正交垂直方向地弯曲连线，按空格键可以改变转角的方向。接线头是为了帮助正确连接端口的连线。当把连线工具放到端口上，接线头就会弹出。接线头还有一个黄色小标识框，显示该端口的名字。

线型为波折号的连线表示坏线。出现坏线的原因有很多，例如：连接了两个控制对象；源端子和终点端子的数据类型不匹配（例如一个是数字型，而另一个是布尔型）。可以通过使用定位工具点击坏线再按下 <Delete> 来删除它。选择 Edit»Remove Bad Wires 或者按下 <Ctrl-B> 可以一次删除流程图中的所有坏线。当 VI 无法运行，或者显示 Signal has Loose Ends（信号丢失终端）的错误信息时，这是一个快捷的调试方法。

## 5 虚拟时钟系统设计的实现

### 5.1 总体设计

- 一. 先从系统中获得一个初始时间，直接从软件用调用一个控件。
- 二. 将获得的秒转换成所需要的时、分、秒，此处用一个子 vi 实现。
- 三. 对时、分、秒进行函数运算，得到时钟指针的坐标值。这里用公式节点来实现。
- 四. 将所得的坐标值输入给绘图的子 vi，再用移动画笔的子 vi 进行移动指针。
- 五. 在第四步之前要有个图片数据处理的过程。通过一个绘制平化像素图的子 vi 来进行处理与运算。
- 六. 程序的循环实现。调用一个 while 循环。
- 七. 程序的流程与结构要采用顺序结构实现，while 循环要嵌套在顺序结构中，再调用库函数来采集图片的数据信息。

整体流程图如图 5.1 所示。

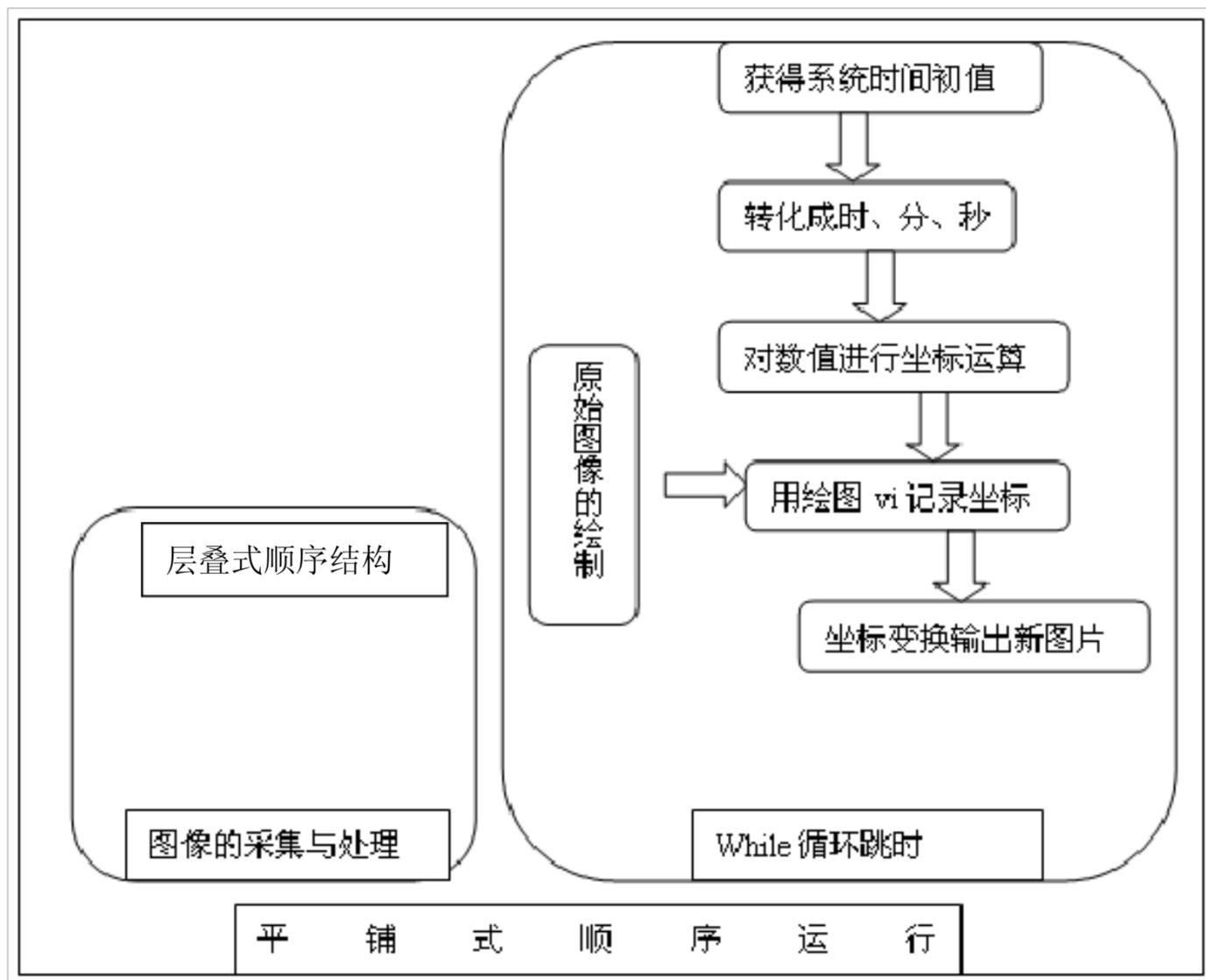


图 5.1 整体流程图

以下即为总体设计出来的时钟前面板及程序框图



图 5.2 时钟演示（前面板）

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/976132154011011005>