

测试原理与技术基础实验-Labview 基础学习

【注】：本实验可以到实验室参加现场实验，也可以在自己电脑上学习完成要求的内容并提交报告。（两种情况都需要进入选课系统预约实验，否则线上参加的同学无法提交报告！）

（一） 实验目的

1. 通过本次实验，熟悉测试原理与技术实验软件平台，为后续实验做好准备。
2. 通过本实验让同学们了解 Labview 的基本操作及其特点，并能使用 Labview 进行最简单的虚拟仪器开发。重点理解图形化编程语言特点及数据流驱动的含义。
3. 熟悉信号采样过程，并通过本实验观察欠采样时信号频谱的混迭现象。
4. 了解采样前后信号频谱的变化，加深对采样定理的理解，掌握采样频率的确定方法。

（二） 虚拟仪器与开发平台 labVIEW 简介

1、 虚拟仪器简介

虚拟仪器技术就是利用高性能的模块化硬件，结合高效灵活的软件来完成各种测试、测量和自动化的应用。灵活高效的软件是虚拟仪器的核心，模块化的硬件是虚拟仪器的基础。虚拟仪器概念是由美国的国家仪器公司在上世纪 80 年代提出的，它提出了软件就是仪器的口号。虚拟仪器同时拥有高效的软件、模块化 I/O 硬件和用于集成的软硬件平台这三大组成部分，充分发挥了虚拟仪器技术性能高、扩展性强、开发时间少，以及出色的集成这四大优势。

虚拟仪器技术的两大组成部分：

1) 高效的软件

软件是虚拟仪器技术中最重要的部份。使用正确的软件工具并通过调用特定的程序模块，工程师和科学家们可以高效地创建自己的应用以及友好的人机交互界面。由 NI 公司提供的行业标准的图形化编程软件——NI LabVIEW，不仅能轻松方便地完成与各种软硬件的连接，更能提供强大的数据处理能力，并将分析结果有效地显示给用户。此外，还有许多其它交互式的测量工具和系统管理软件工具，例如连接设计与测试的交互式软件 SignalExpress、基于 ANSI-C 语言的 LabWindows/CVI、支持微软 Visual Studio 的 Measurement Studio 等等。拥有了功能强大的软件，我们就可以灵活的设计需要的测试系统或数据采集系统。

2) 模块化的 I/O 硬件

面对如今日益复杂的测试测量应用，模块化的硬件设计成为主流。借助于总线技术的发展，比如 PCI, PXI, PCMCIA, USB 或者是 IEEE 1394 总线，我们可以将各类的模块化硬件产品进行互联。产品种类从数据采集及信号调理、模块化仪器、机器视觉、运动控制、仪器控制、分布式 I/O 到 CAN 接口等工业通讯等。高性能的硬件产品结合灵活的开发软件，可以为使我们创建完全自定义的测量系统，满足各种灵活独特的应用需求。

虚拟仪器技术的四大优势：

1) 性能高

虚拟仪器技术是在 PC 技术的基础上发展起来的, 所以完全“继承”了以现成即用的 PC 技术为主导的最新商业技术的优点, 包括功能超卓的处理器和文件 I/O, 使您在数据高速导入磁盘的同时就能实时地进行复杂的分析。此外, 当前正蓬勃发展的一些新兴技术(如多核、PCI Express 等)也成为推动虚拟仪器技术发展的新动力, 使其展现出更强大的优势。

2) 扩展性强

虚拟仪器的软硬件平台使得工程师和科学家们不再局限于固有的、封闭的技术之中。得益于软件的灵活性, 只需更新计算机或测量硬件, 就能以最少的硬件投资和极少、甚至无需软件上的升级即可改进的整个测试系统。

3) 开发时间少

在驱动和应用两个层面上, 虚拟仪器高效的软件构架能与计算机、仪器仪表和通讯方面的最新技术结合在一起。

4) 出色的集成

虚拟仪器技术从本质上说是一个集成的软硬件概念。随着产品在功能上不断地趋于复杂, 工程师们通常需要集成多个测量设备来满足完整的测试需求, 而连接和集成这些不同设备总是要耗费大量的时间。虚拟仪器软件平台为 I/O 设备提供了标准的接口, 帮助用户轻松地将多个测量设备集成到一个系统之中, 减少了任务的复杂性。

2、 LabVIEW 与虚拟仪器

目前广泛应用的虚拟仪器开发工具中按技术类型可分为两类: 面向对象的编程技术和图形编程技术。两者在虚拟仪器开发中都有应用, 各有所长。

可视化编程语言环境 Visual C++、Visual Basic 等属于前者, 均可以用来开发虚拟仪器的配套软件, 但与图形编程语言相比, 编程难度较大, 开发周期较长且不易进行更改、升级和维护等。

而图形编程语言具有容易入门、编程简单、开发周期短等特点, 开发出的应用程序界面美观, 功能强大, 正日益成为主流。较为流行的有 NI 公司的 LabVIEW 和 HP 公司的 VEE 等软件, 此外还有 Lookout、BridgeVIEW 和 LabWINDOWS/CVI 等。其中最为常用的就是 LabVIEW。

(1) LabVIEW 简介

LabVIEW 的发展历史

在 80 年代初, 几乎所有的仪器控制程序都是用 BASIC 语言开发的, 几乎所有使用可编程仪器的实验室在搭建测试系统时, 仪器控制器的主导语言都是 BASIC。所有使用仪器的工程师和技术人员都得做编程工作, 与所有的文本编程语言一样, 使用 BASIC 语言进行仪器编程的过程是单调、繁琐而乏味的。

National Instruments 公司的编程团队注意到了这些现象, 它们试图开发一种用于开发仪器控制程序的新工具, 减轻工程师和科学家们的负担。National Instruments 公司的创始人 Jim Truchard 和 Jeff Kodosky 博士, 以及 Jack MacCrisken 顾问便着手开发这种软件工具。

LabVIEW 最初的概念来源于一个大型测试系统, 该系统用于测试海军的声纳定位仪传感器, 该系统的主要缺点是需要投入极长的编程时间(超过 18 个工作年), 使用者想做任何改动都得懂得面板上的复杂方法。Kodosky 重新定义了该测试系统的概念, 提出了虚拟仪器

的仪器软件分层体系的概念，即一个虚拟仪器可由若干较低层的虚拟仪器组成，低层虚拟仪器代表了最基本的软件结构模块，负责计算和输入输出操作。虚拟仪器的概念是核心概念，而且这个概念最终包含在这个产品的名称中，该产品最终命名为 LabVIEW，即 Laboratory Virtual Instrument Engineering Workbench（实验室虚拟仪器工程工作平台）的首字母缩写组合。工程师在设计系统时常常要绘制框图，而 LabVIEW 所体现的一大革新就是将框图转化为可被计算机识别和编译的程序，使用图形化的工作平台作为一种程序设计语言来开发仪器软件。这样可以帮助工程师将问题形象化，轻松完成系统设计，减轻编程负担。

在突破了种种技术上的难关后，1986 年 10 月，LabVIEW Version 1.0 for Macintosh 面世。作为第一种可行的图形化语言，它引起了全世界的巨大反响。它所带来的全新的虚拟仪器概念和图形化编程环境为业界带来了一场革命，而且赢得了人们的赞叹。1990 年 1 月，LabVIEW 2.0 发售。随后为使 LabVIEW 具有可移植性，能够在不同平台上运行，开始向 Windows 和 Sun 上移植系统，1992 年 8 月，LabVIEW 2.5 for Windows 发布，1993 年 1 月，LabVIEW 2.5.2 for Sun 发布，1993 年 10 月，LabVIEW 3.0 for Windows NT 发布。直至 2004 年 5 月发布的最新版本 7.1，LabVIEW 经过不断完善，已经成为一套划时代的图形化编程系统，在数据采集与控制、数据分析、数据表达方面，有着全新的概念和独特的优势，几乎已成为业界标准。

(2) LabVIEW 编程语言的特点

LabVIEW 是一个基于 G (Graphic) 语言的图形编程开发环境，在工业界和学术界中广泛用作开发数据采集系统、仪器控制软件和分析软件的标准语言，对于科学研究和工程应用来说是很理想的语言。它含有种类丰富的函数库，科学家和工程师们利用它可以方便灵活地搭建功能强大的测试系统。LabVIEW 编程语言最主要的两个特点是图形化编程和数据流驱动：

LabVIEW 与 Visual C++、Visual Basic、LabWindows/CVI 等编程语言不同，后几种都是基于文本的语言，而 LabVIEW 则是使用图形化程序设计语言 G 语言，用框图代替了传统的程序代码，编程的过程即是使用图形符号表达程序行为的过程，源代码不是文本而是框图。一个 VI 有三个主要部分组成：框图、前面板和图标 / 连接器。框图是程序代码的图形表示，一个典型的 VI 框图如图 1 所示：

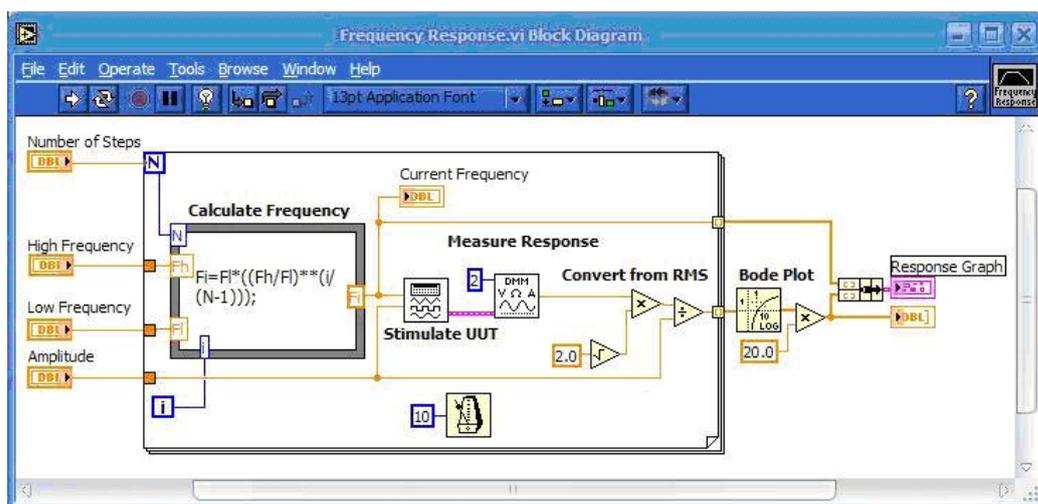


图 1 Labview 框图的图形化设计

LabVIEW 的框图中使用了丰富的设备和模块图标，与科学家、工程师们习惯的大部分图标基本一致，这使得编程过程和思维过程非常的相似。多样化的图标和丰富的色彩也给用

户带来不一样的体验和乐趣。

前面板是 VI 的交互式用户界面，外观和功能都类似于传统仪器面板，用户的输入数据通过前面板传递给框图，计算和分析结果也在前面板上以数字、图形、表格等各种不同方式显示出来。一个典型的 VI 前面板如图 2 所示：

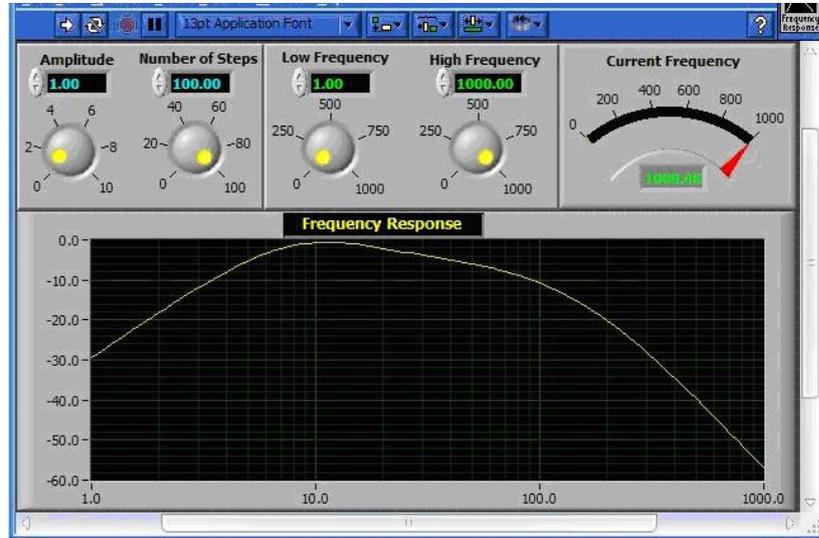


图 2 labview 设计虚拟仪器前面板

数据流驱动：

宏观上讲，LabVIEW 的运行机制已不再是传统上的冯·诺伊曼式计算机体系结构的执行方式了。传统计算机语言（如 C 语言）中的顺序执行结构在 LabVIEW 中被并行机制所代替。本质上讲它是一种带有图形控制流结构的数据流模式，程序中的每一个函数节点只有在获得它的全部输入数据后才能够被执行。既然 LabVIEW 程序是数据流驱动的，数据程序设计规定，一个目标只有当它的所有输入有效时才能够被执行；而目标的输出只有当它的功能完全时才是有效的。于是 LabVIEW 中被连接的函数节点之间的数据流控制着程序的执行次序，而不像文本程序那样受到行顺序执行的约束。我们可以通过相互连接函数节点简洁高效地开发应用程序，还可以有多个数据通道同步运行，即所谓的多线程，如图 3 所示。

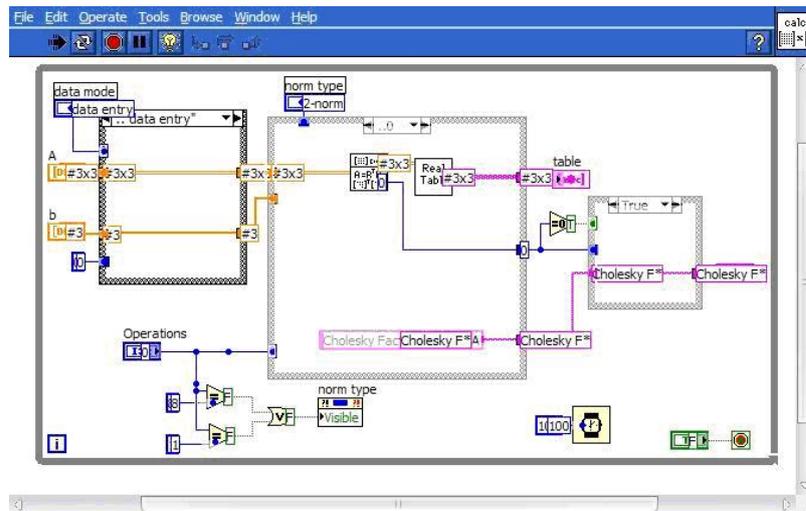


图 3 数据流驱动

(三) 频率混叠与采样定理的实验原理

模拟信号经过 A/D 变换转换为数字信号的过程称之为采样，信号采样后其频谱产生了周期延拓，每隔一个采样频率 ω_s ，重复出现一次。

1. 频混现象

频混现象又称为频谱混叠效应，它是由于采样信号频谱发生变化，而出现高、低频成分发生混淆的一种现象，如图 1 所示。信号 $x(t)$ 的傅里叶变换为 $X(\omega)$ ，其频带范围为 $-\omega_m \sim +\omega_m$ ；采样信号 $x(t)$ 的傅里叶变换是一个周期谱图，其周期为 ω_s ，并且：

$$\omega_s = 2\pi / T_s$$

T_s 为时域采样周期。当采样周期 T_s 较小时， $\omega_s > 2\omega_m$ ，周期谱图相互分离如图 4 中 (b) 所示；当 T_s 较大时， $\omega_s < 2\omega_m$ ，周期谱图相互重叠，即谱图之间高频与低频部分发生重叠，如图 1 中(c)所示，此即为频混现象，这将使信号复原时丢失原始信号中的高频信息。

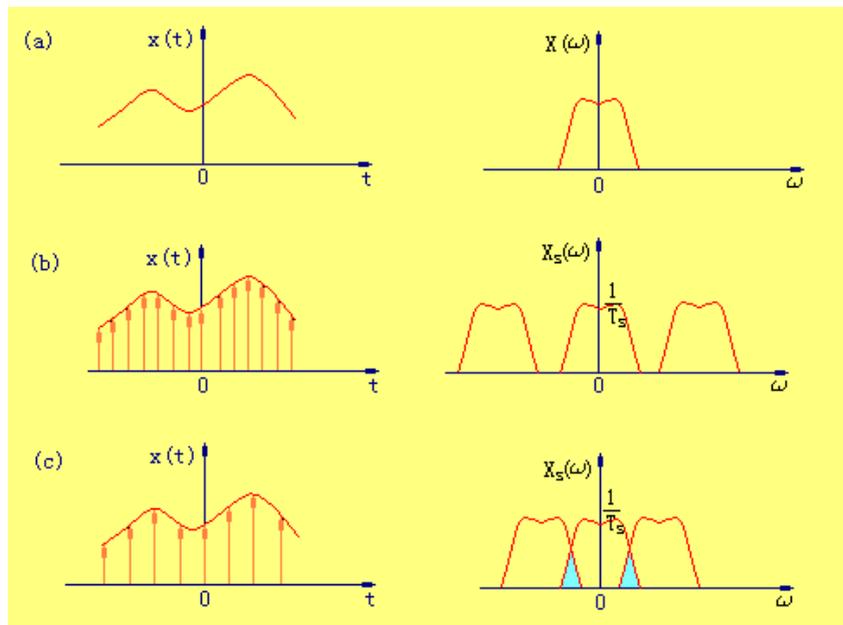


图 4 采样信号的频混现象

下面从时域信号波形来看这种情况。图 5(a)是频率正确的情况，以及其复原信号；(b)是采样频率过低的情况，复原的是一个虚假的低频信号。

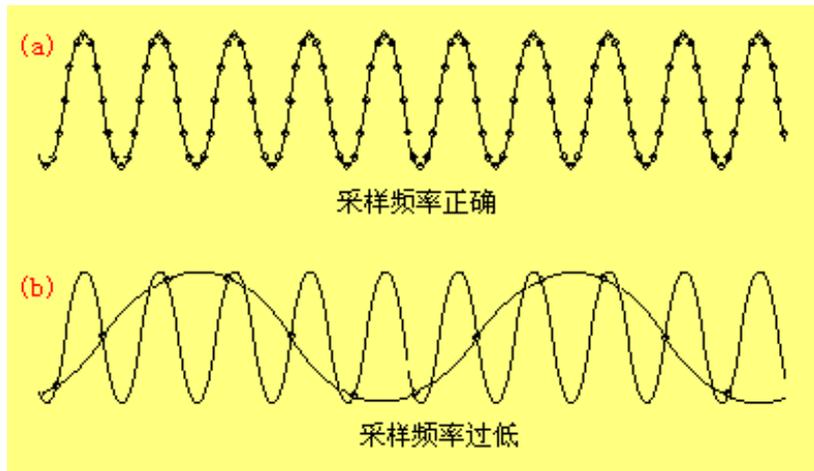


图 5 发生频混现象的时域信号波形

当采样信号的频率低于被采样信号的最高频率时，采样所得的信号中混入了虚假的低频分量，这种现象叫做频率混叠。

2. 采样定理

上述情况表明，如果 $\omega_s > 2\omega_m$ ，就不发生频混现象，因此对采样脉冲序列的间隔 T_s 须加以限制，即采样频率 ω_s ($2\pi / T_s$) 或 f_s ($1 / T_s$) 必须大于或等于信号 $x(t)$ 中的最高频率 ω_m 的两倍，即： $\omega_s > 2\omega_m$ ，或 $f_s > 2f_m$ 。

为了保证采样后的信号能真实地保留原始模拟信号的信息，采样信号的频率必须至少为原信号中最高频率成分的 2 倍。这是采样的基本法则，称为采样定理。

需要注意的是，在对信号进行采样时，满足了采样定理，只能保证不发生频率混叠，对信号的频谱作逆傅立叶变换时，可以完全变换为原时域采样信号，而不能保证此时的采样信号能真实地反映原信号。工程实际中采样频率通常大于信号中最高频率成分的 3 到 5 倍。

(四) 实验内容

1. 设计一个以随机信号为信号源的单通道模拟示波器
2. 设计一个以正弦波为信号源的单通道模拟示波器
3. 设计一个可以同时显示 2 路波形的多通道模拟示波器
4. 频率混叠与采样定理

(五) 实验报告要求

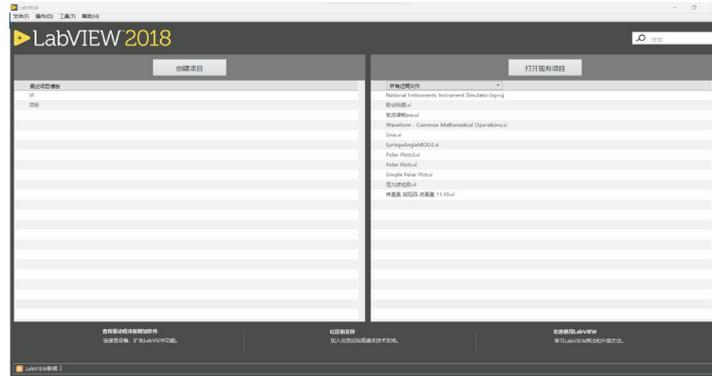
完成实验后，登录中心实验选课系统网站 https://lcme.sjtu.edu.cn/openlab/prepare_login，以 pdf 文件格式提交 VI 的前面板和程序面板截图，**前面板中必须包含学生姓名、学号、完成日期。**

(六) 实验设备

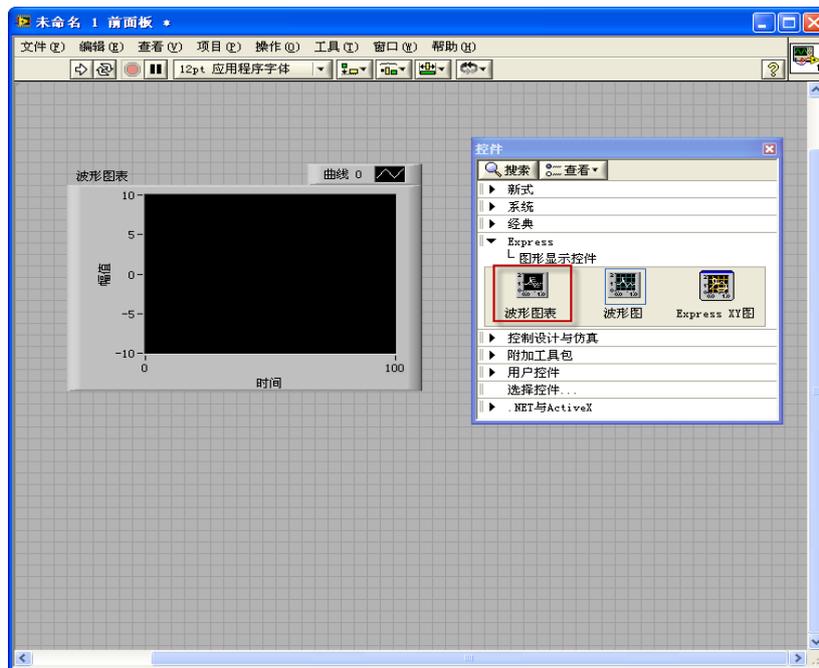
1. 电脑 1 台
2. LabVIEW 中文版 1 套：<http://lic.si.sjtu.edu.cn/Default/soft> 中的 labview，按要求安装激活。

(七) 实验步骤

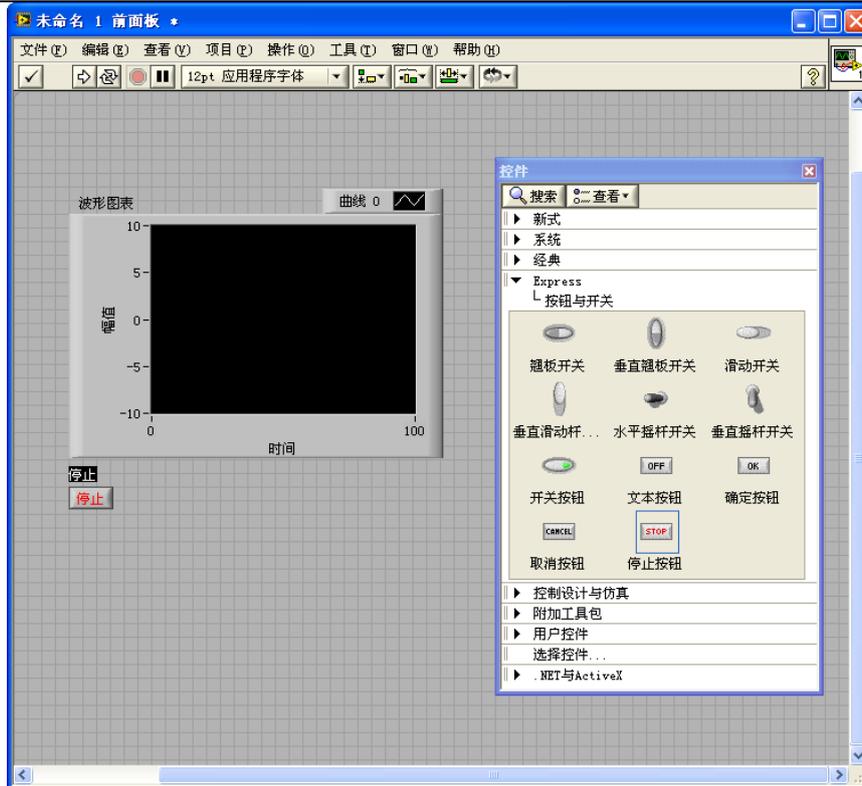
1. 内容一：设计一个以随机信号为信号源的单通道模拟示波器



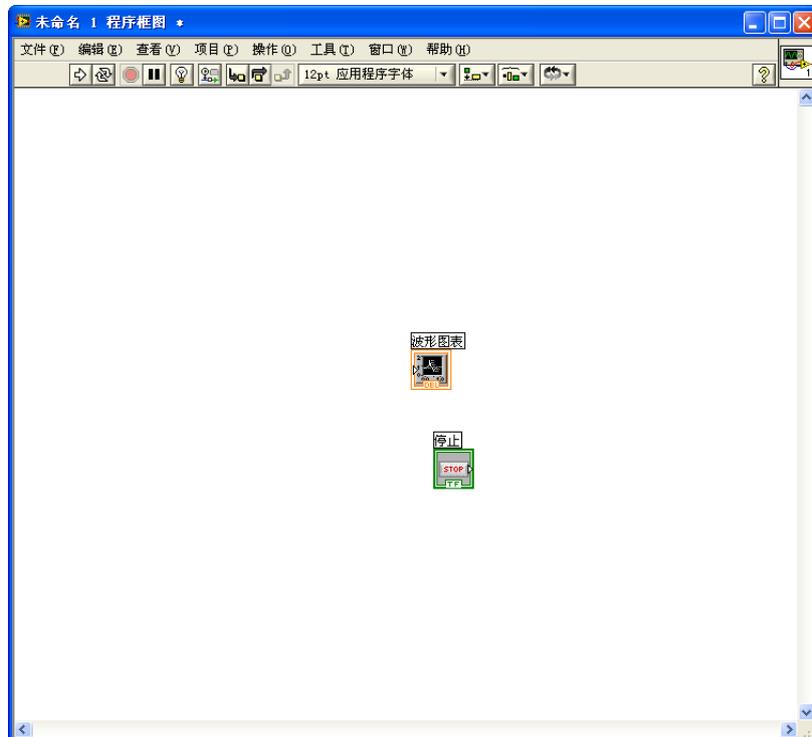
步骤 1：打开 LabVIEW，点击文件，点击新建 VI。



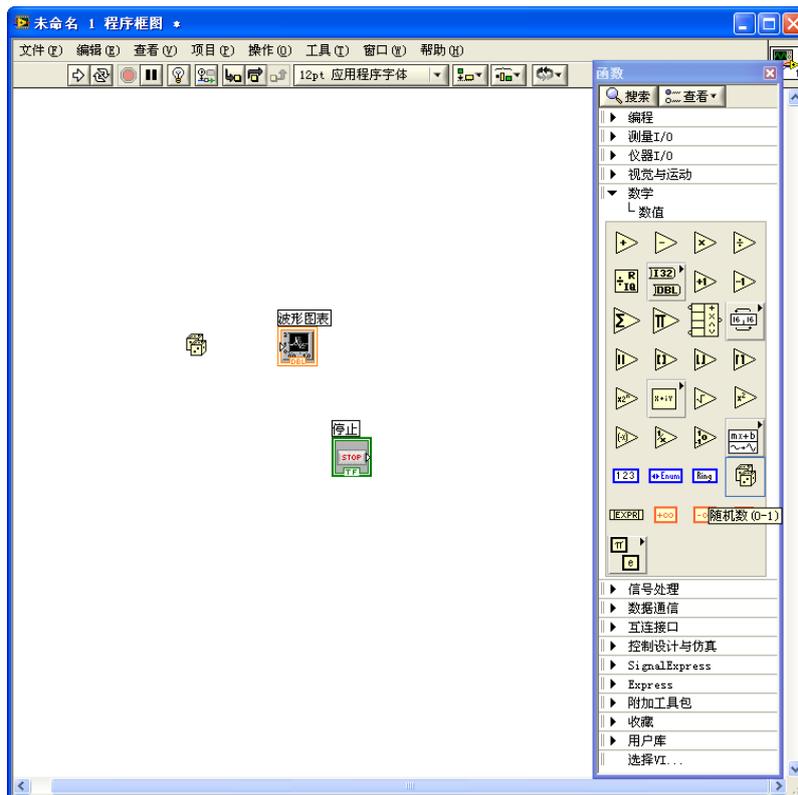
步骤 2：在控件中选“Express” — “图像显示控件” — “波形图表”控件，放置于前面板



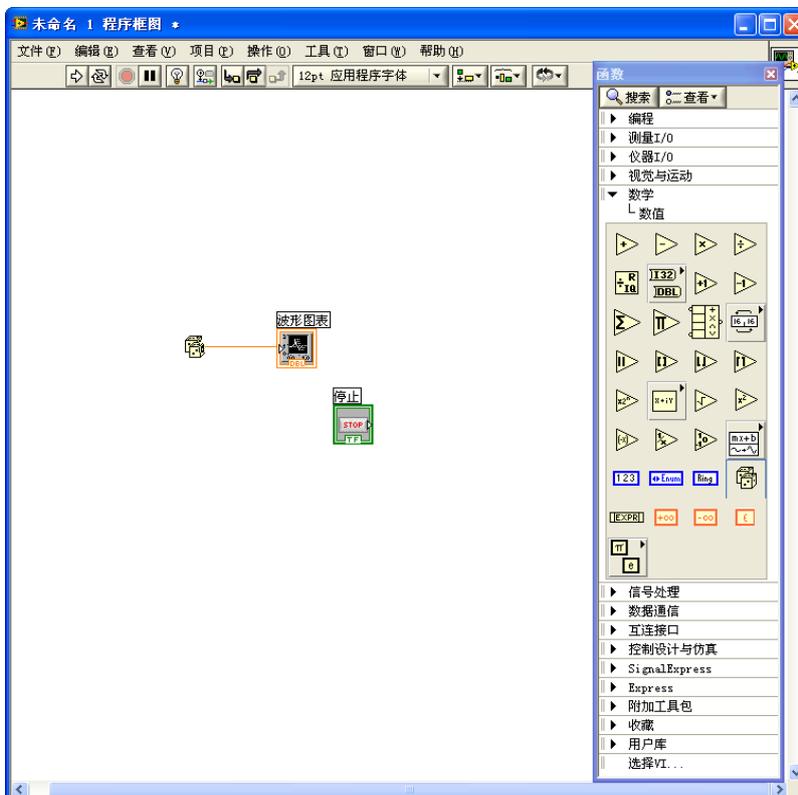
步骤 3: 在控件中选“Express” — “按钮与开关” — “停止按钮”控件，放置前面板



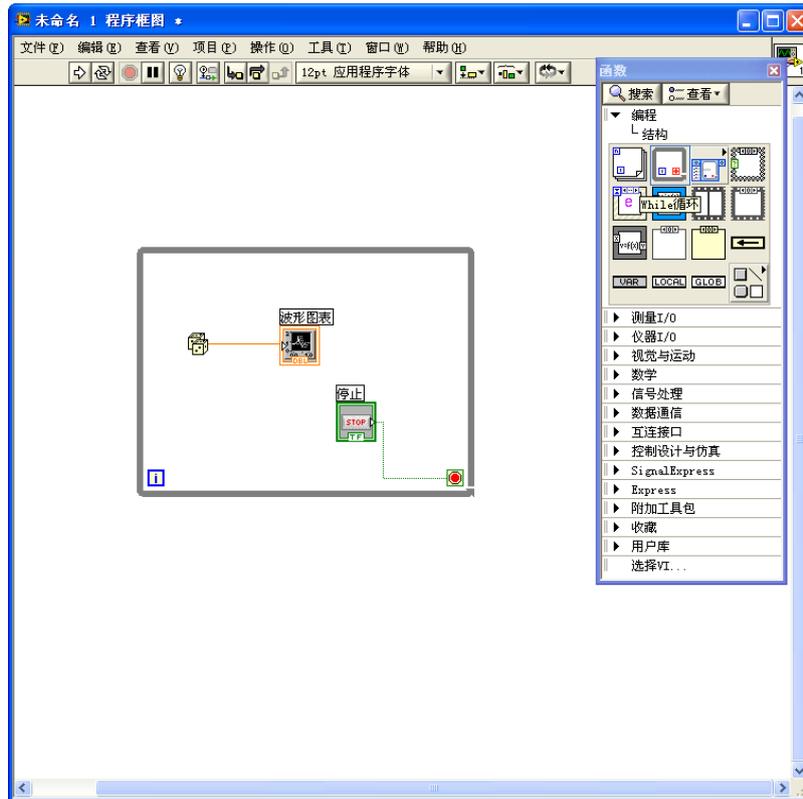
步骤 4: 切换到程序框图，可以发现框图中已经存在对应“波形图表”和“停止”控件



步骤 5: 在“函数”窗口中选择“数学”——“数值”——“随机数(0-1)”，放置到程序框图

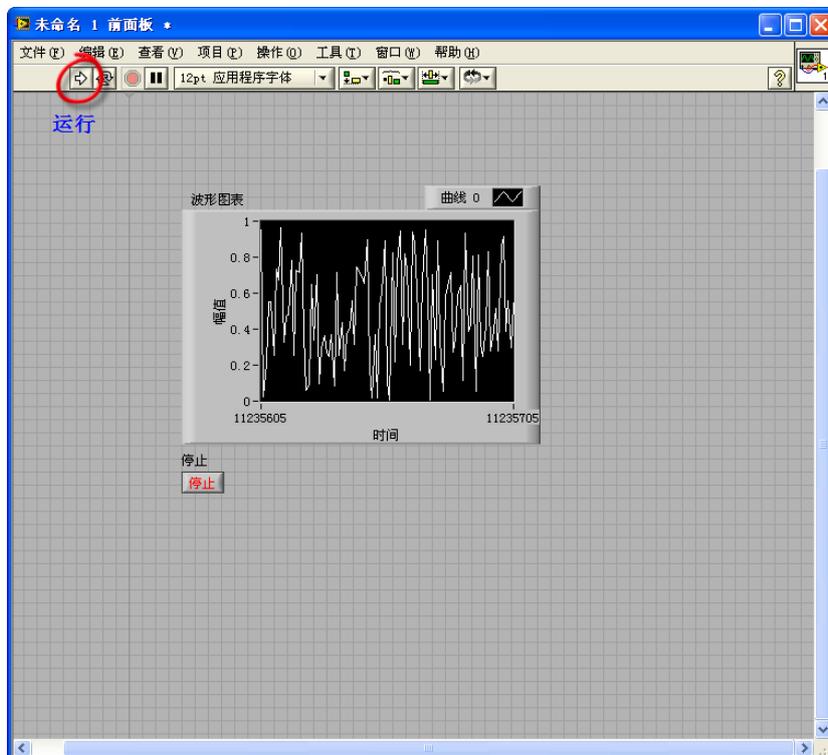


步骤 6: 利用连线工具讲“随机数 (0-1)”与“波形图表”相连。

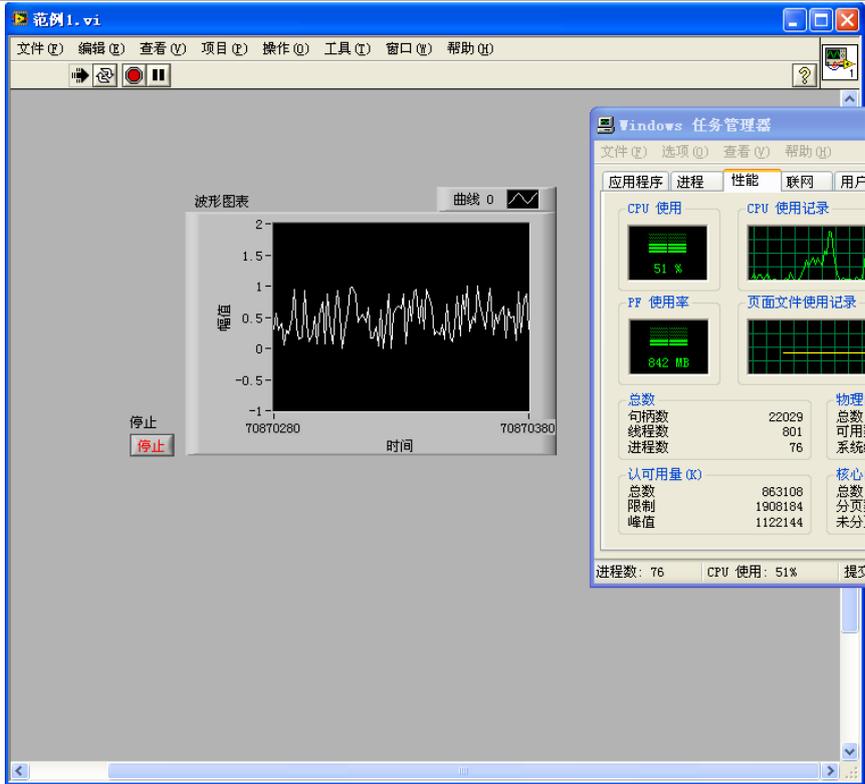


步骤 7: 选择“函数”——“编程”——“结构”——“While 循环”，将图中元素包含其中。

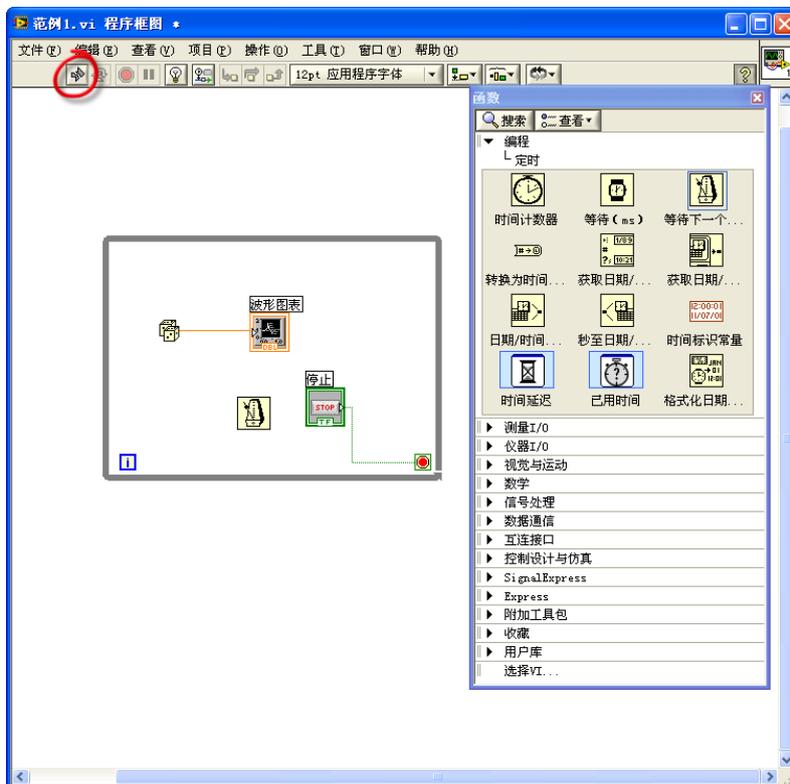
步骤 8: 用连线工具将“停止”连线到“While 循环”的红色标志上。



步骤 9: 点击前面板的“运行”，观察波形图表；点击“停止”，观察波形图表。

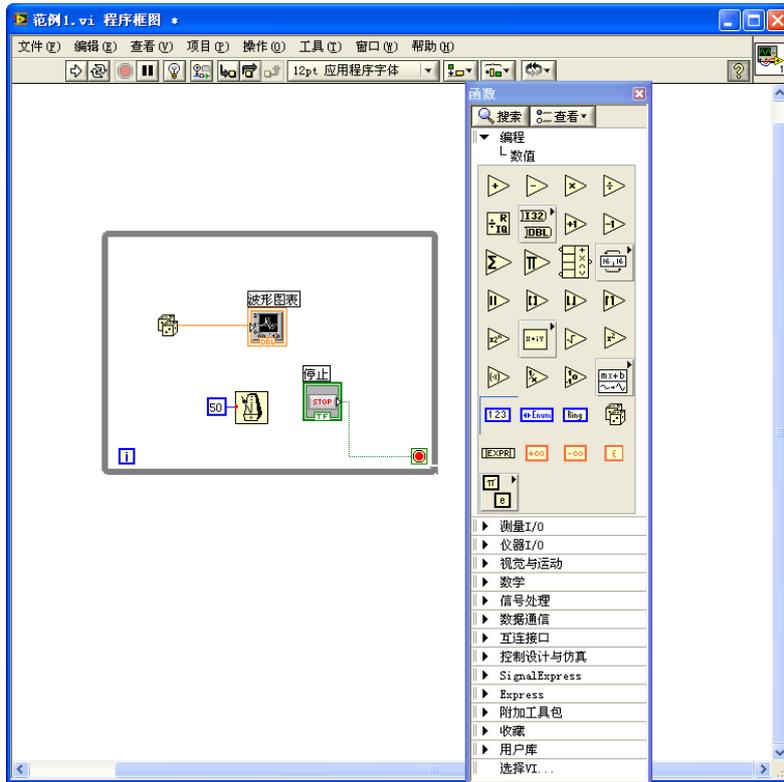


步骤 10: 打开 windows 任务管理器，运行程序，观察 CPU 使用负荷。



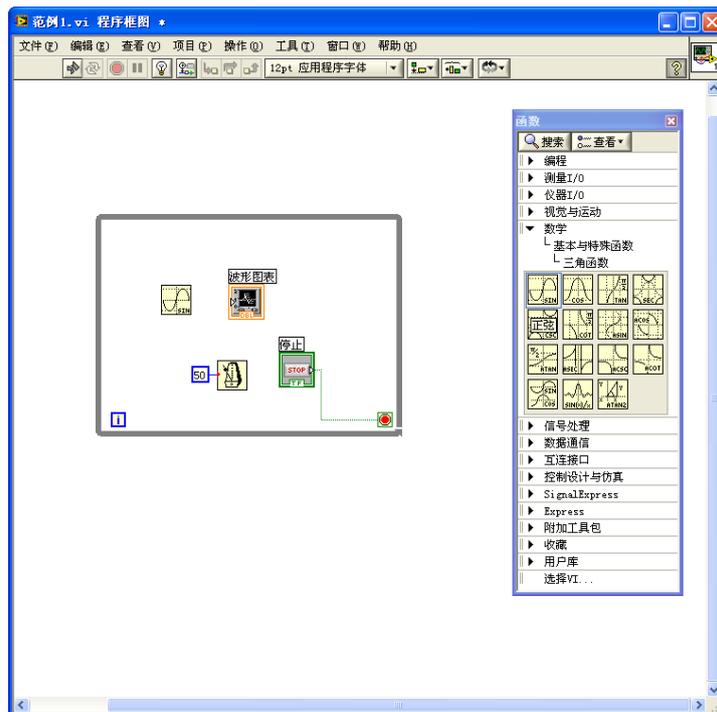
步骤 11: 在“函数”面板中选择“编程”—“定时”—“等待下一个整数倍毫秒”定时

器，放于程序框图 While 循环中。注意图中红圈中“运行”图标的变化。

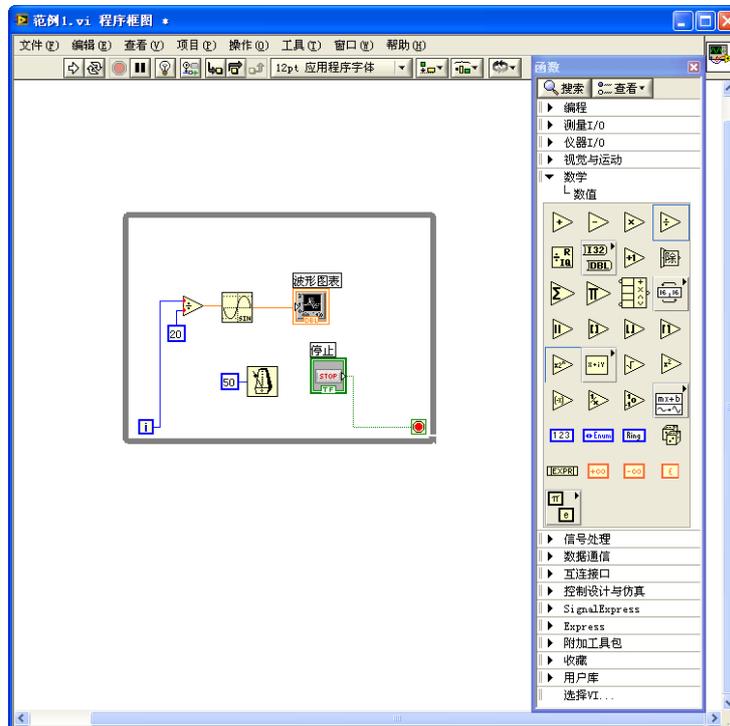


步骤 12: 在“函数”面板中选择“编程”——“数值”——“数值常量”，放于 while 循环中，输入数值 50，并与定时器相连。运行程序，观察变化，同时观察“任务管理器”中 CPU 负荷变化，理解定时器作用。调整定时器的数值，观察变化。

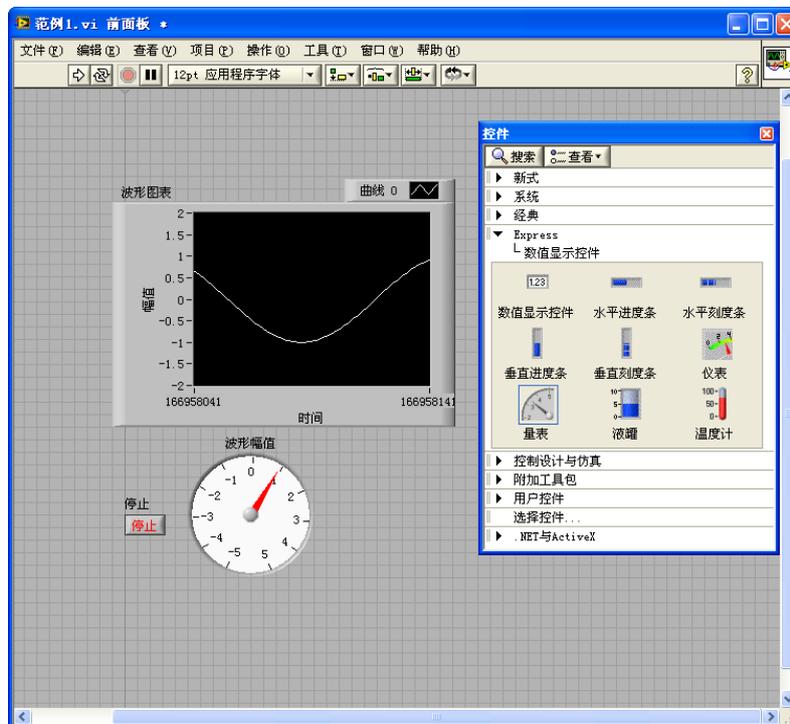
2. 内容二: 设计一个以正弦波为信号源的单通道模拟示波器
在内容 1 的基础上，对程序框图进行修改。



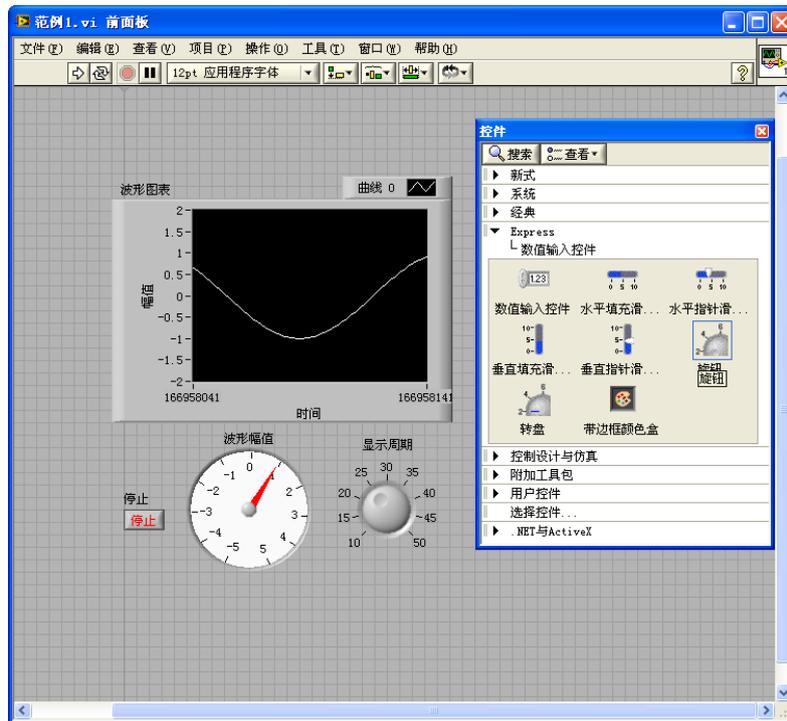
步骤 1: 删除程序框图中的“随机函数(0-1)” (选择后, 点击键盘“delete”), 清理多余连线 (CTRL+B)。选择“函数”模版中“数学” — “基本与特殊函数” — “三角函数” — “正弦”函数, 放于 while 循环中。



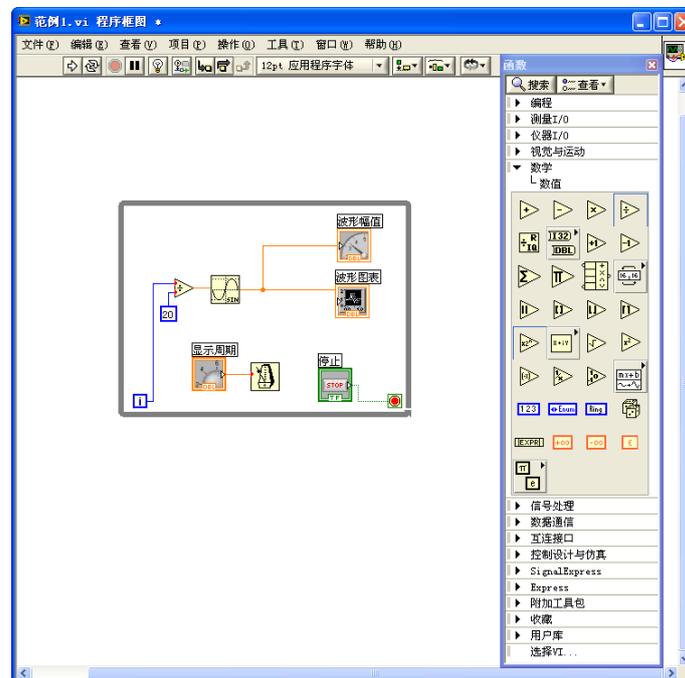
步骤 2: “函数”模版中选择“数学” — “数值” — “除”, 放于 while 循环中, 并按照图示连线。运行程序, 观察效果。



步骤 3: 在前面板的“控件”中选择“Express” — “数值显示控件” — “量表”并放置。修改“量表”的显示范围 (-5, +5), 并改名为“波形幅值”。



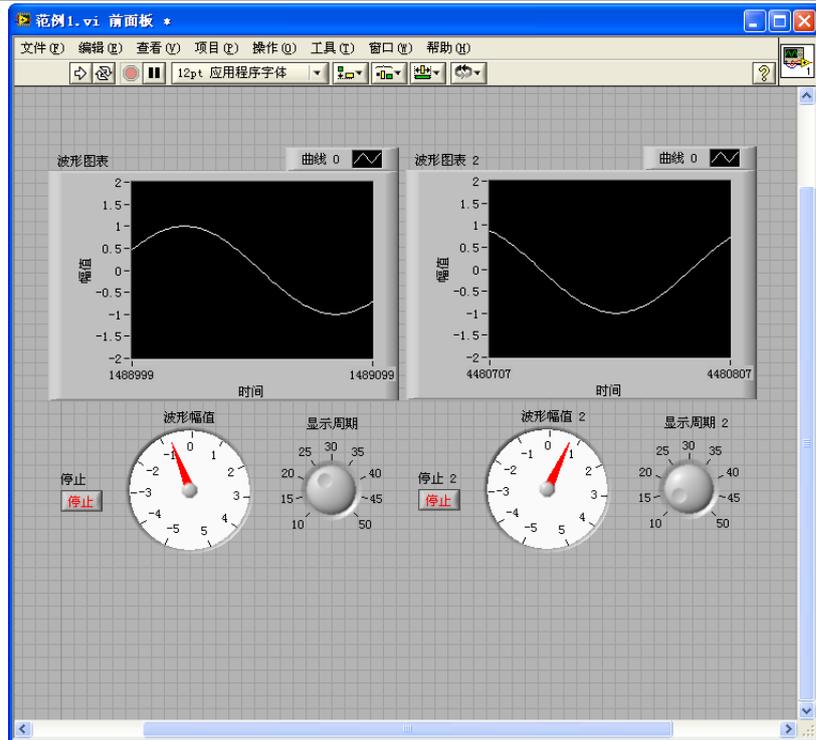
步骤 4: 在前面板的“控件”中选择“Express”——“数值输入控件”——“旋钮”并放置。修改“旋钮”的输入范围（10，50），并改名为“显示周期”。



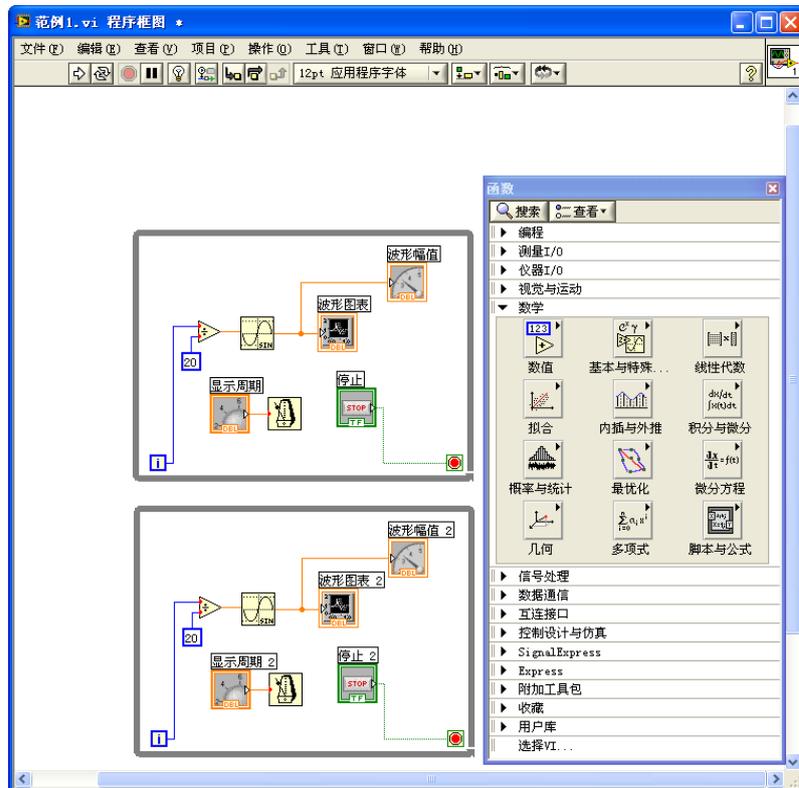
步骤 5: 切换到“程序框图”，按照图示调整并连线。完成连线后，运行程序，观察结果。并可以自行改变显示和输入控件，设计前面板。

3. 内容三：设计一个可以同时显示 2 路波形的多通道模拟示波器

通过添加示波器面板，同时显示两路波形，理解 LabVIEW 并行多线程概念。



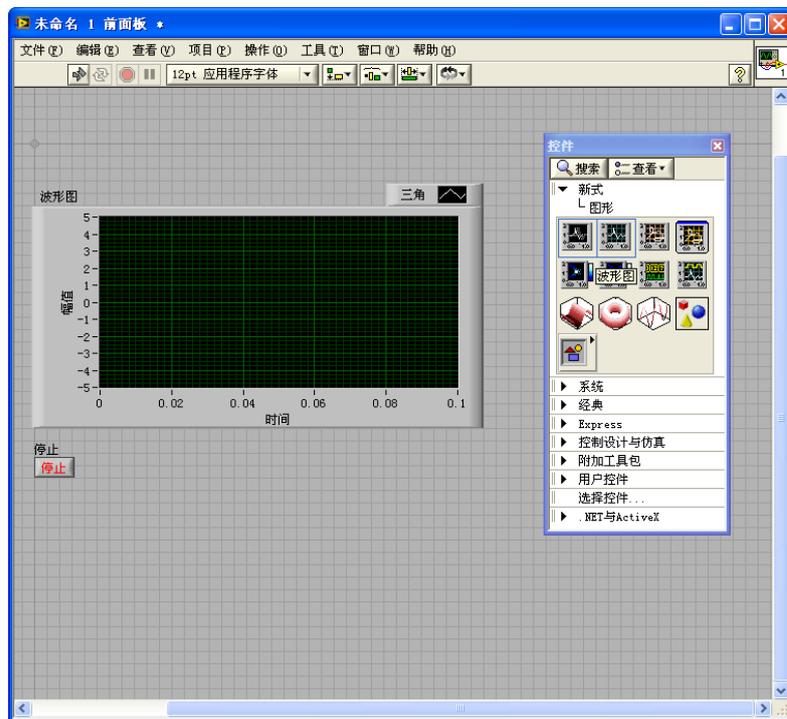
步骤 1：在实验 2 的基础上添加相同的前面板元素，设计 2 路示波器显示面板。



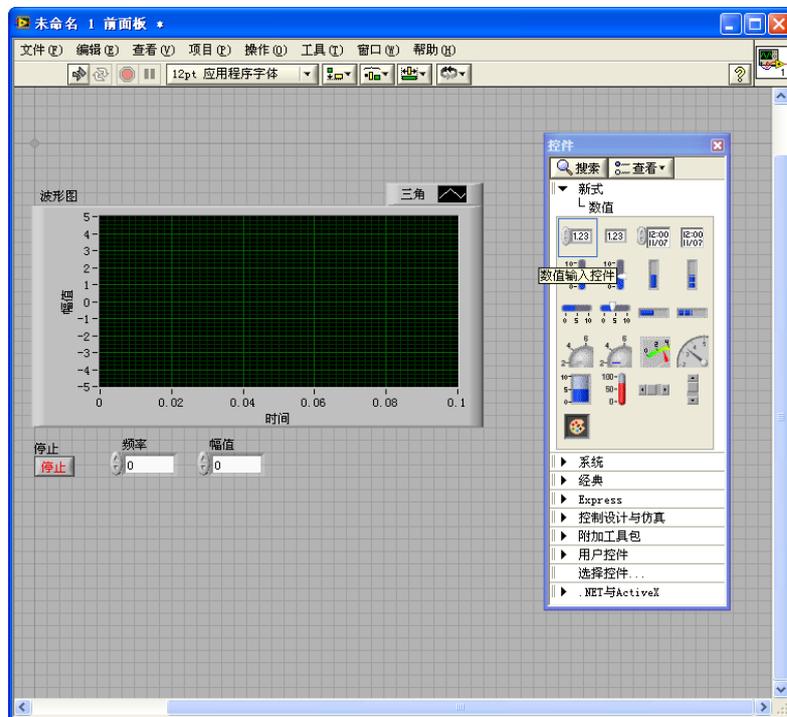
步骤 2：如图完成程序框图中函数连线，并运行程序。理解多 while 循环运行原理，思考 labVIEW 多线程并行运行的概念。

4. 内容四：设计一模拟信号， $x(t)=80\sin(2\pi*50*t)$ 。在 100Hz - 1000 Hz 的范围内用

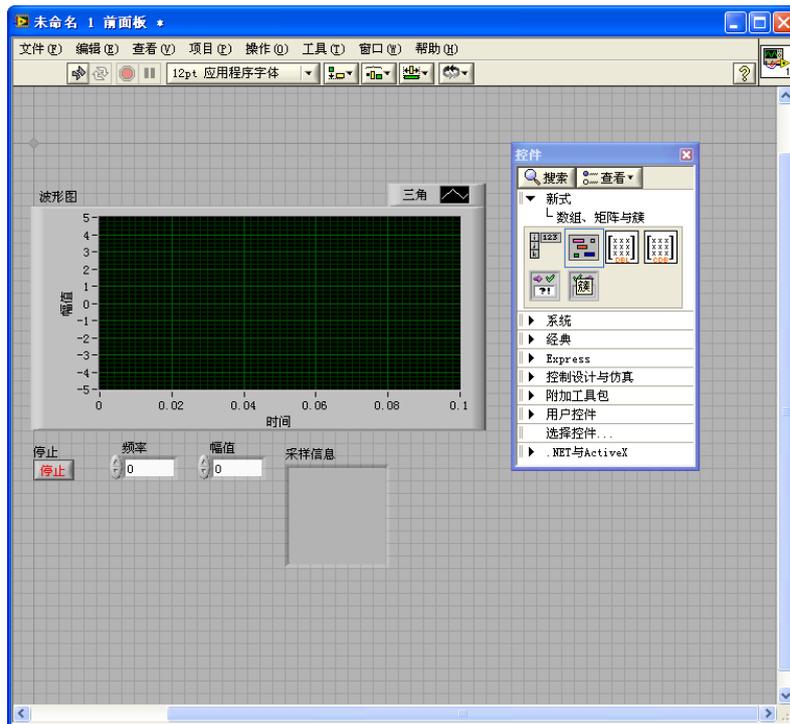
不同的采样频率对信号进行采样，观察采样信号频谱的频率混迭现象。



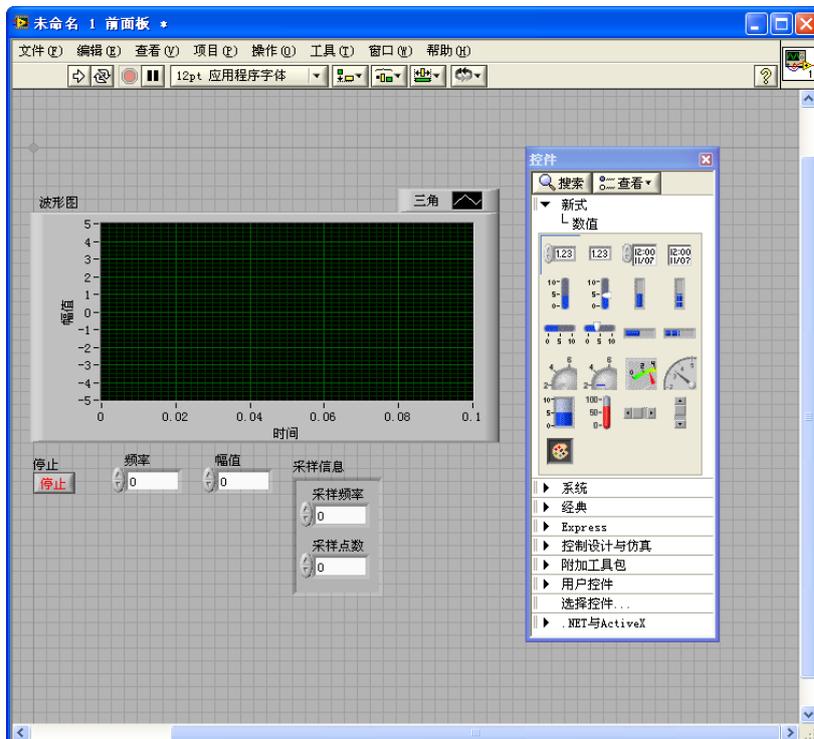
步骤 1：在控件面板上选择“新式”——“图形”——“波形图”控件，放于前面板上；同时放置“停止”按钮。



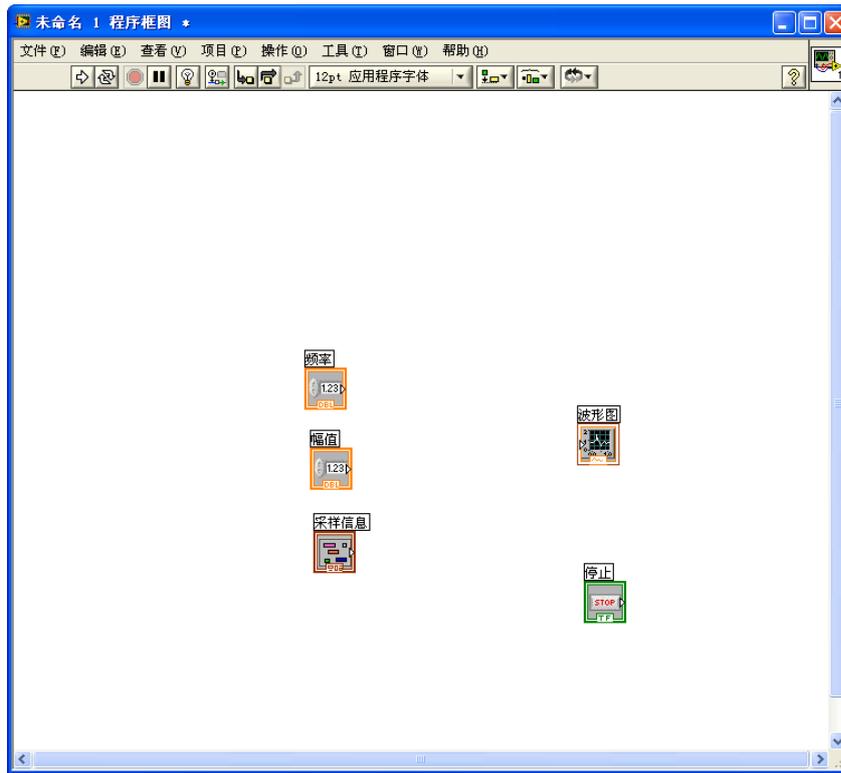
步骤 2：在控件面板上选择“新式”——“数值”——“数值输入控件”，放于前面板上，并改名“频率”；重复操作，添加另一个“数值输入控件”，改名为“幅值”。



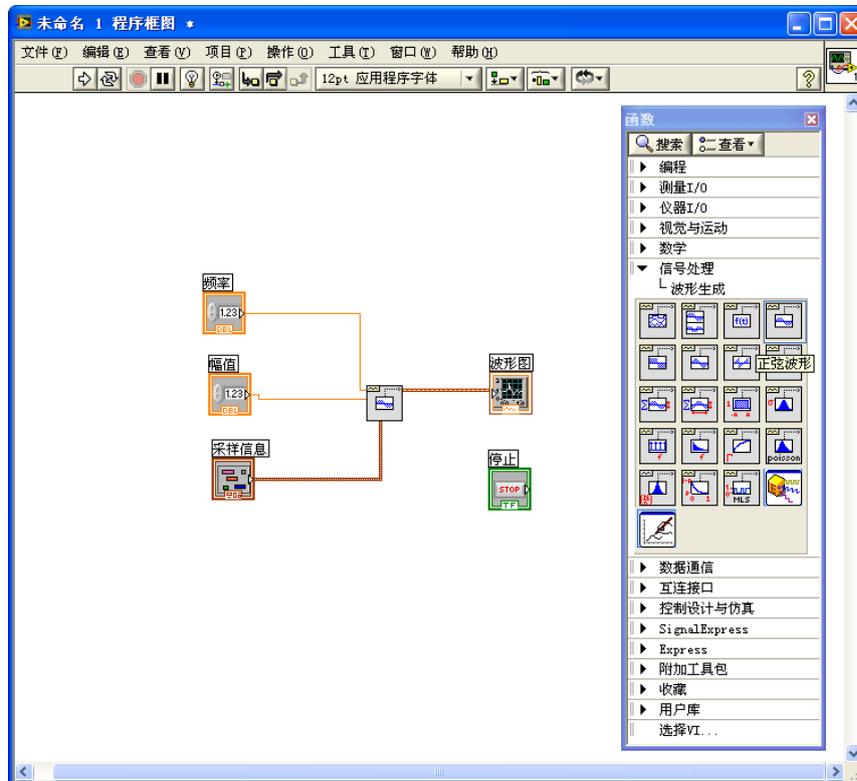
步骤 3: 在控件面板上选择“新式”——“数组、矩阵与簇”——“簇”，放置于前面板上，并改名为“采样信息”。



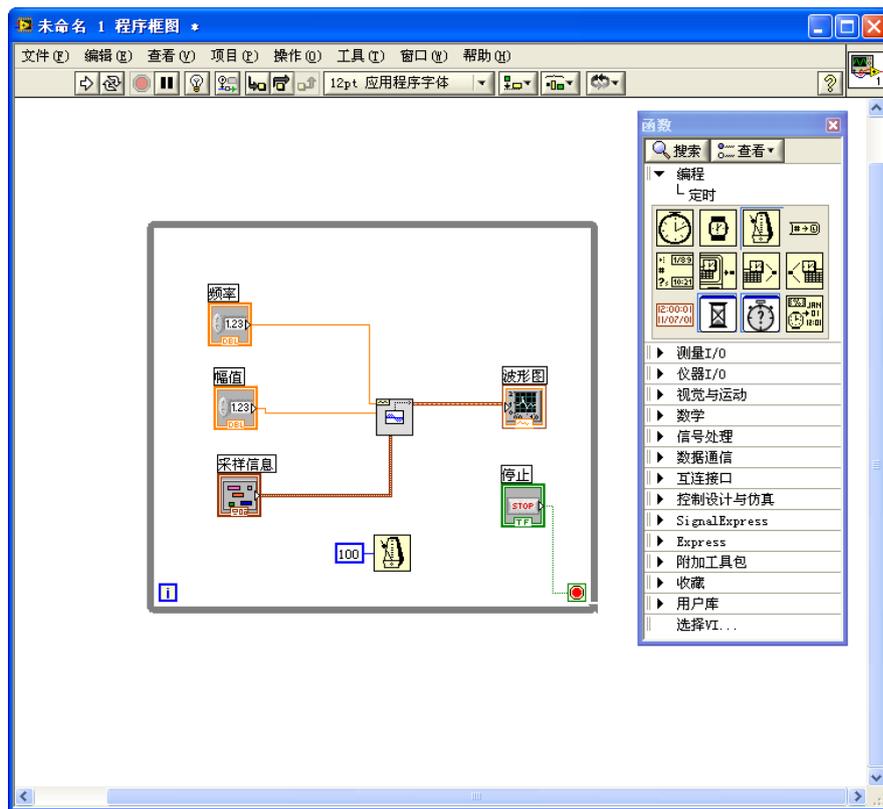
步骤 4: 在控件面板上选择“新式”——“数值”——“输入数值”控件 2 次，放于前面板的“采样信息”簇中（拖拉到簇的容器范围内），并分别改名“采样频率”和“采样点数”。特别注意放置顺序，首先放入的“输入数值”控件为“采样频率”，后放入的为“采样点数”。



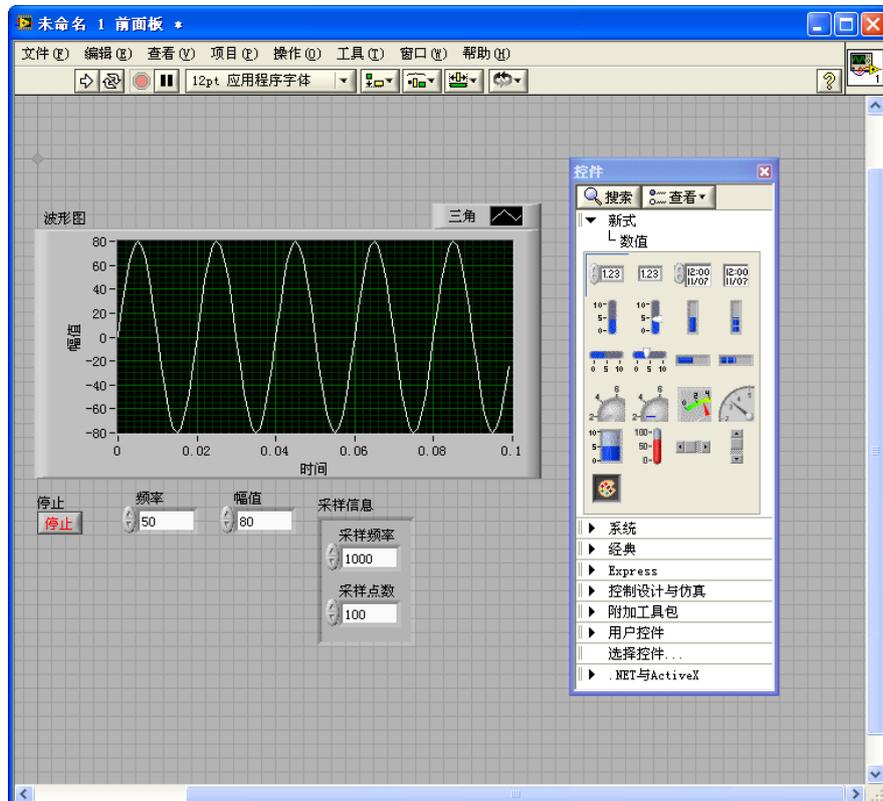
步骤 5:切换到程序框图, 此时有 5 个图标分别对应前面板的控件。



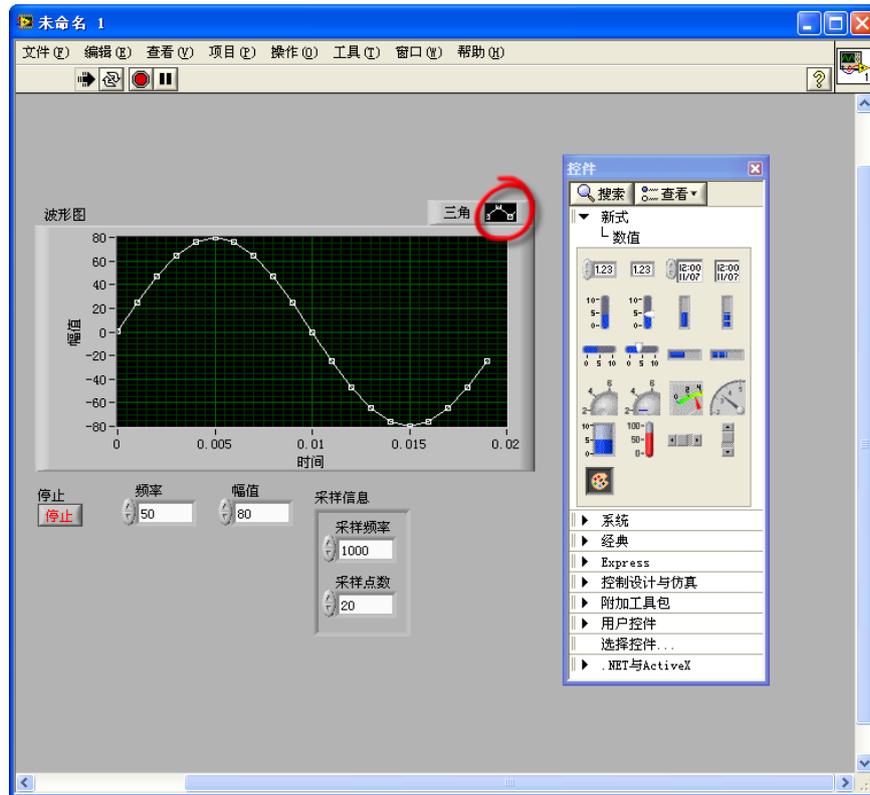
步骤 6: 添加“信号处理”—“波形生成”—“正弦波形”函数, 并按照图示完成连线。



步骤 7: 添加“while”循环和“定时器”，完成如图所示连线。



步骤 8: 按照图示分别设置前面板上控件的数值: 频率 50, 幅值 80, 采样频率 1000, 采样点数 100。运行程序, 观察结果。然后逐步调整采样频率, 范围为 (101-1000)。



步骤 9: 改变波形图的显示方式, 如图所示。逐步修改采样点数, 观察结果。理解采样频率和采样点数的含义。修改采样频率低于 101, 观察运行结果, 并解释原因。

(八) 思考题

- 什么是虚拟仪器, 虚拟仪器与传统仪器比有何优点?
- labVIEW 开发平台有何特点?
- 简述数据流与传统文本编程程序执行顺序上的区别。
- 若信号频率为 5000Hz, 采样频率为 800Hz, 请问模拟信号采样后的混迭频率是多少 Hz?
- 为什么在实际测量中采样频率通常要大于信号中最高频率成分的 3 到 5 倍?

【注】: 思考题不要求在报告中回答

附加内容 - 波形的合成与分解

【注】：此实验内容不作为指定要求，供有兴趣的同学参考学习。

(一) 实验目的

1. 加深了解信号分析手段之一的傅立叶变换的基本思想和物理意义。
2. 观察和分析由多个频率、幅值和相位成一定关系的正弦波叠加的合成波形。
3. 观察和分析频率、幅值相同，相位角不同的正弦波叠加的合成波形。
4. 通过本实验熟悉信号的合成、分解原理，了解信号频谱的含义。

(二) 实验原理

按傅立叶分析的原理，任何周期信号都可以用一组三角函数 $\{\sin(2\pi f_0 t), \cos(2\pi f_0 t)\}$ 的组合表示：

$$\begin{aligned} x(t) = & a_0/2 \\ & + a_1 \sin(2\pi f_0 t) + b_1 \cos(2\pi f_0 t) \\ & + a_2 \sin(4\pi f_0 t) + b_2 \cos(4\pi f_0 t) \\ & + \dots \end{aligned} \quad (1)$$

也就是说，我们可以用一组正弦波和余弦波来合成任意形状的周期信号。

例如对于典型的方波，其时域表达式为：

$$x(t) = \begin{cases} -A & (-T/2 < t < 0) \\ A & (0 < t < T/2) \end{cases} \quad (2)$$

根据傅立叶变换，其三角函数展开式为：

$$\begin{aligned} X(t) &= \frac{4A}{\pi} (\sin \omega_0 t + \frac{1}{3} \sin 3\omega_0 t + \frac{1}{5} \sin 5\omega_0 t + \dots) \\ &= \frac{4A}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \sin n\omega_0 t \\ &= \frac{4A}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \cos(n\omega_0 t - \frac{\pi}{2}) \quad n = 1, 3, 5, 7, 9, \dots \end{aligned} \quad (3)$$

由此可见，周期方波是由一系列频率成分成谐波关系，幅值成一定比例，相位角为 0 的正弦波叠加合成的。

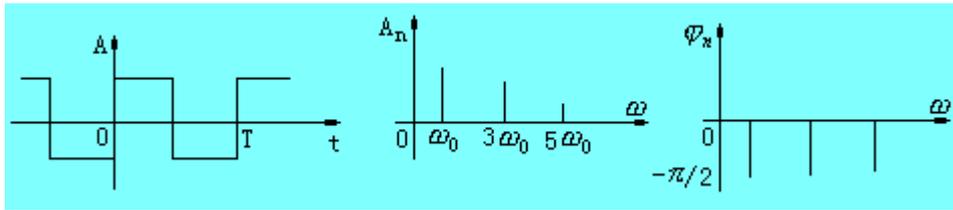


图1 方波信号的波形、幅值谱和相位谱

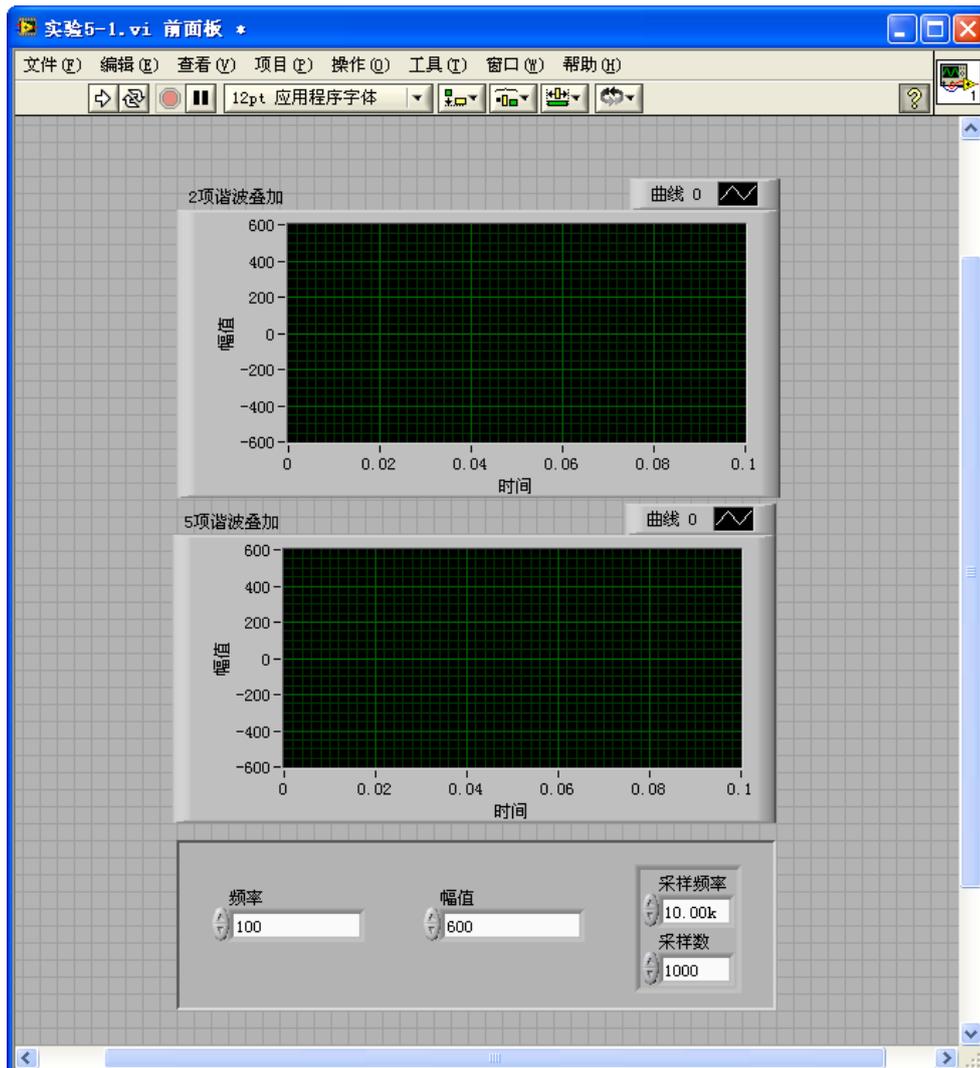
那么,我们在实验过程中就可以通过设计一组奇次正弦波来完成方波信号的合成,同理,对三角波、锯齿波等周期信号也可以用一组正弦波和余弦波信号来合成。

(三) 实验内容

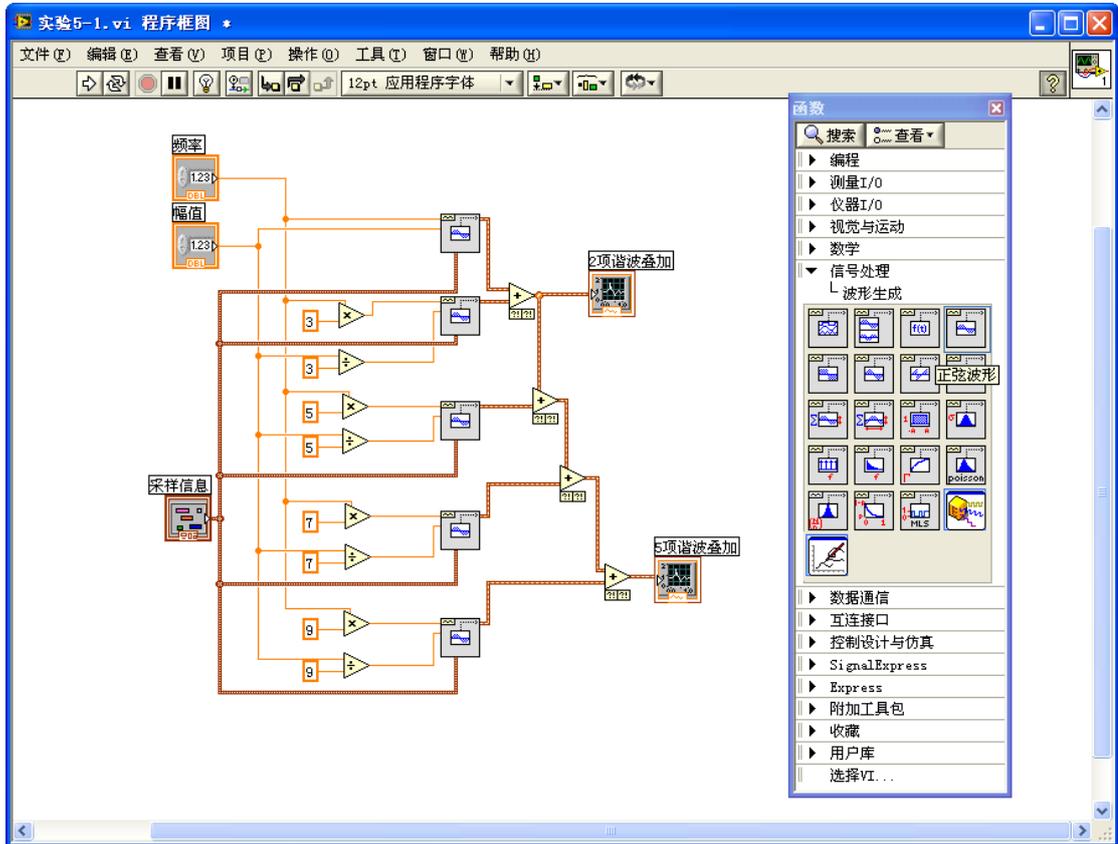
- 1、用前5项谐波叠加近似合成一个频率为100,幅值为600的方波信号;
- 2、在内容1的基础上设计一个多阶可调谐波叠加器,叠加合成一个频率为100,幅值600的方波信号。

(四) 实验步骤

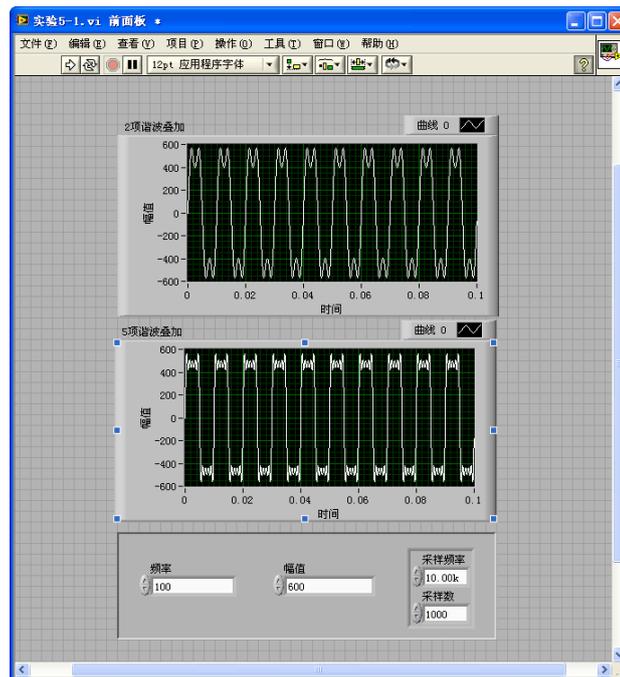
内容1:用前5项谐波叠加近似合成一个频率为100,幅值为600的方波信号;



步骤 1：设计前面板，添加 2 个波形图控件，分别取名“2 项谐波叠加”和“5 项谐波叠加”；添加 2 个数值输入控件，分别取名“频率”“幅值”，分别设置为 100 和 600；添加一个簇，包含 2 个数值输入控件，分别为“采样频率”“采样数”，设置为 10000 和 1000。

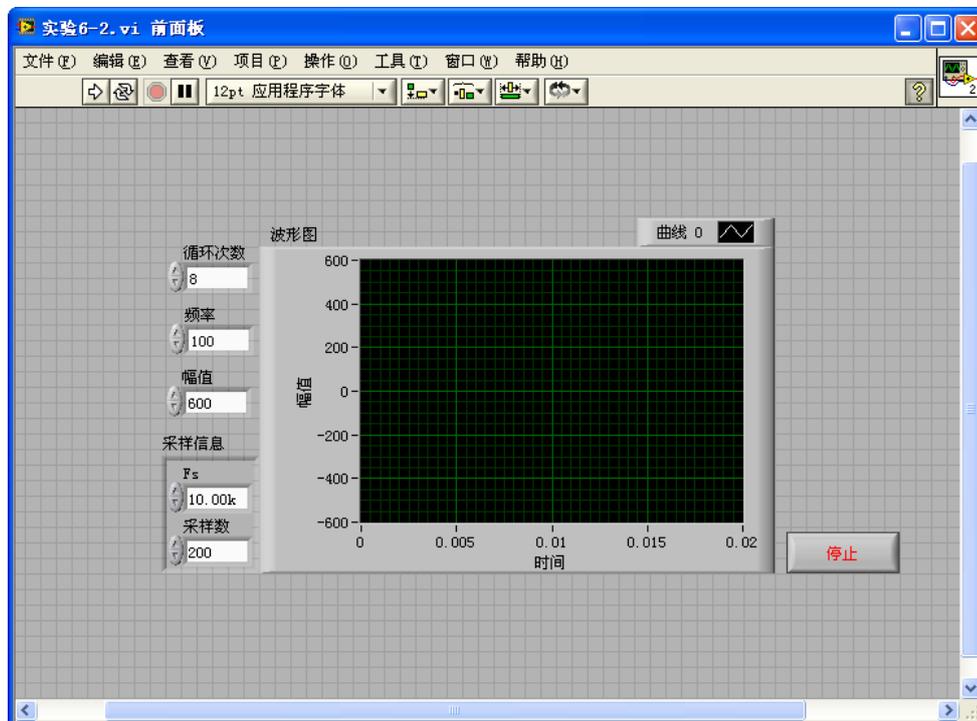


步骤 2：添加 5 个“信号处理”—“波形生成”—“正弦波形”函数，完成如图连线。将“频率”“幅值”“采样信息”分别连接到“正弦波形”对应端口，将 5 项波形叠加。其中，前 2 项谐波叠加后的波形输出到“2 项谐波叠加”、前 5 项谐波叠加后的波形输出到“5 项谐波叠加”。

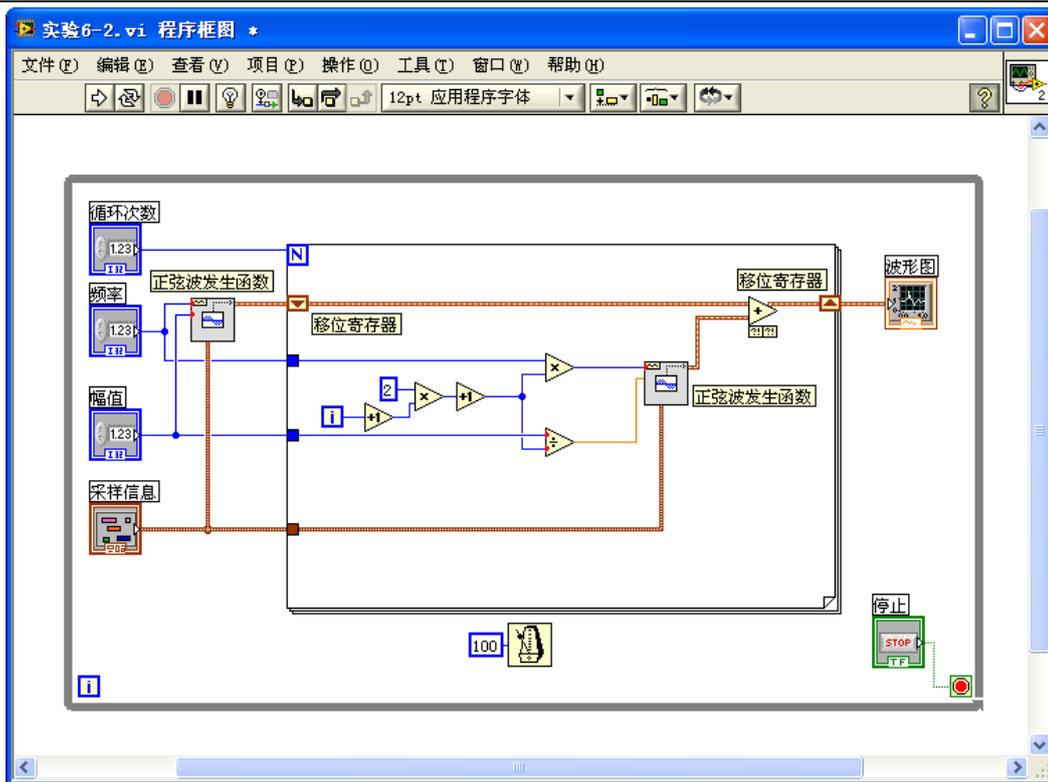


运行程序，观察波形，可以发现随着叠加项数的增加，越来越接近方波。

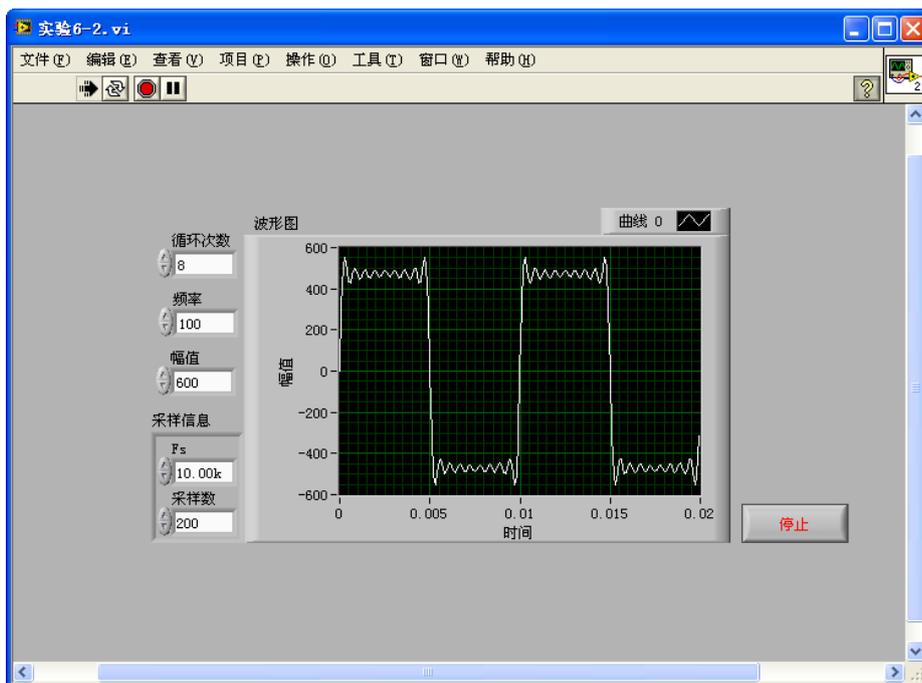
内容 2：通过前面进行 5 项谐波叠加成近似方波的尝试，为了能够更加接近于真正的方波，叠加的项数越多越好。然而在内容 1 中我们可以发现，多项谐波的叠加必然带来更复杂的程序框图，是否有更好的办法来实现方便的谐波叠加设计呢？在内容 2 中，我们将实现一个更高级的谐波叠加器。



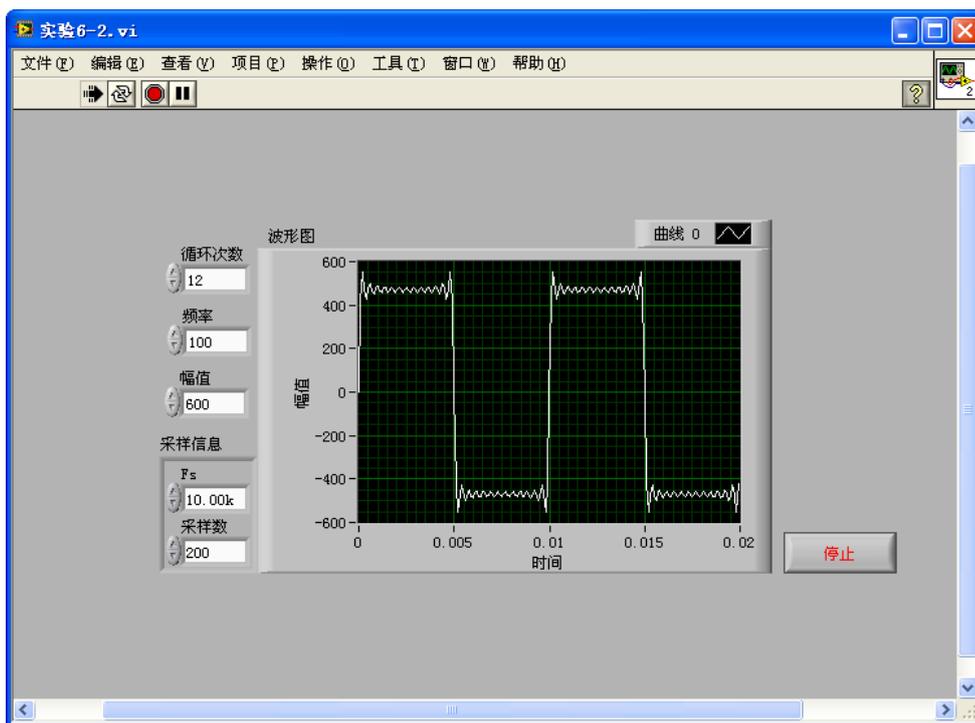
步骤 1：设计前面板如图所示，添加相应控件并完成数值输入。



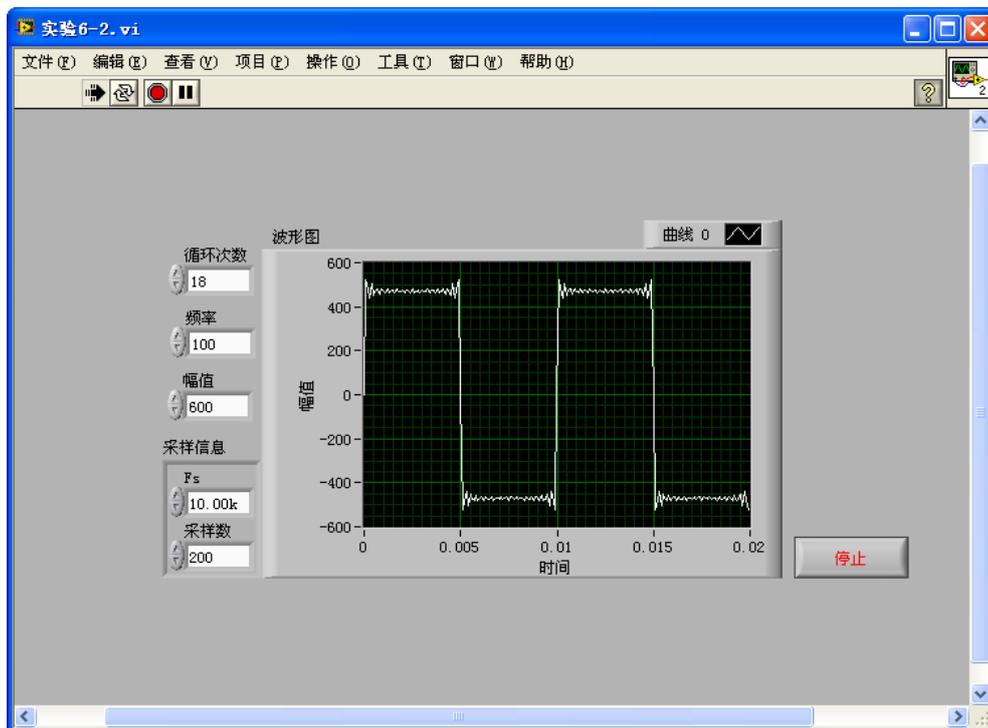
步骤 2: 程序框图如图所示。整个程序框图中包含了 While 循环和 For 循环。其中 For 循环内的程序框图用于叠加波形, While 循环用于保持程序连续执行。程序中有 2 个“正弦波发生”函数, 一个位于 For 循环外, 用于生成第一项谐波; 一个位于 For 循环内部, 用于叠加其余的谐波。在 For 循环内部, 有一对“移位寄存器”, 用于完成 For 循环内部的谐波叠加 (移位寄存器是用于储存变量在上一次循环结束后的输出值, 通过右击 For 循环边界, 可以选择添加移位寄存器)。运行程序, 调节叠加项数, 观察结果。



9 项谐波叠加



13 项谐波叠加



19 项谐波叠加

通过调整循环次数，从而方便的实现了波形叠加。通过观察更高的叠加项数，我们可以更好的理解波形叠加的原理。

(五) 实验报告

可与前面实验一起截图提交。