

10.2 利用 LabVIEW 进行图像采集与处理

利用 LabVIEW 进行图像处理是一个非常重要的应用。在许多行业中采用图像的采集和识别来进行判断、控制，使操作更加精确，具有可信度、人性化、智能化。本节将讲解利用 LabVIEW 进行图像采集和处理的实例。

10.2.1 图像处理介绍

图像处理也可以称作视觉处理。LabVIEW 提供了多种图像处理的方法。其中 NI 公司的视觉采集软件提供的驱动和函数，既能够从数千种连接到 NI 帧接收器上的不同相机上采集图像，也能够从连接在 PC、PXI 系统或笔记本电脑上标准端口的 IEEE 1394 和千兆位以太网视觉相机采集图像。

LabVIEW 中的视觉开发模块作为强大的机器视觉处理库，配有各类函数，其中包括：边缘检测、颗粒分析、光学字符识别和验证、一维和二维代码支持、几何与模式匹配、颜色工具。该模块可与 NI 公司的所有软件、C++、Microsoft Visual

Basic、Microsoft .NET 相互调用，为用户提供了相当便利的操作。用户可通过视觉开发模块的同步功能，实现与运动或数据采集测量的同步。

NI 公司提供的图像处理软件包 Vision 8.5.1 Acquisition Software，是专门为 LabVIEW 8.5 服务的。它可以在 LabVIEW 8.5 中完成各种关于图像处理、视觉运行的控制。

10.2.2 实例内容说明

本实例主要完成通过 USB 摄像头采集图像，并经过一些运算对图像进行数据分析。在实例中用采集到的图片作样本，让系统认识一个像素，然后开始自动查找图像中的相同像素，查找时还要对图片进行翻转，以全面找到相同的像素，最后再标注出这些点的中心位置和点数。

10.2.3 Vision 安装与介绍

本例主要通过 Vision 8.5.1 Acquisition Software 软件包来实现。Vision 8.5.1 Acquisition Software 软件包是一种专门的图像处理软件，需要单独安装。此软件一般可以通过供应商购买，也可以通过 NI 公司网站下载。

1 · Vision 安装

Vision 8.5.1 Acquisition Software 安装步骤如下：

(1) 把光盘放入计算机光驱，系统会自动识别，并显示出安装自检界面，如图10-34 所示。

(2) 当安装程序自检完成后, “Next” 按钮显示为可操作状态。此时单击 “Next” 按钮进入下一个界面, 选择安装路径, 如图 10-35 所示。



图 10-34 安装自检



图 10-35 选择安装路径

(3) 一般此路径就是 LabVIEW 软件安装的默认路径。如果在安装 LabVIEW 软件时没有改动路径, 则不需要修改此路径。单击 “Next” 按钮, 进入下一步选择安装驱动的界面, 如图 10-36 所示。

(4) 在驱动选择界面中, 可以选择要安装的驱动类型。这里可以看到有 4 种驱动可安装。当不需要某种驱动时, 在其选项上打叉即可。当选择完要安装的驱动后, 单击 “Next” 按钮, 进入下一步的安装确认界面, 如图 10-37 所示。

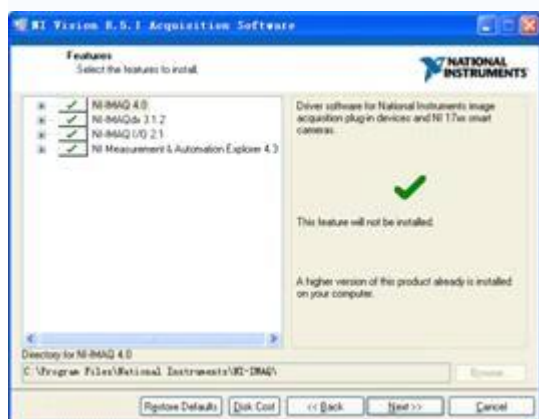


图 10-36 驱动选择

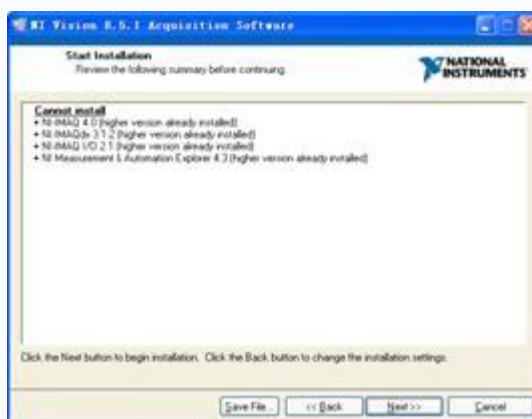


图 10-37 信息确认

(5) 对显示的信息确认无误后, 单击 “Next” 按钮开始安装, 如图 10-38 所示。

(6) 当安装完上面的所有软件包后, 系统会显示安装完成界面, 并提示是否选择激活此软件。根据自己的需要选择后, 单击 “Next” 按钮进入下一步骤。最后安装完成的界面如图 10-39 所示。

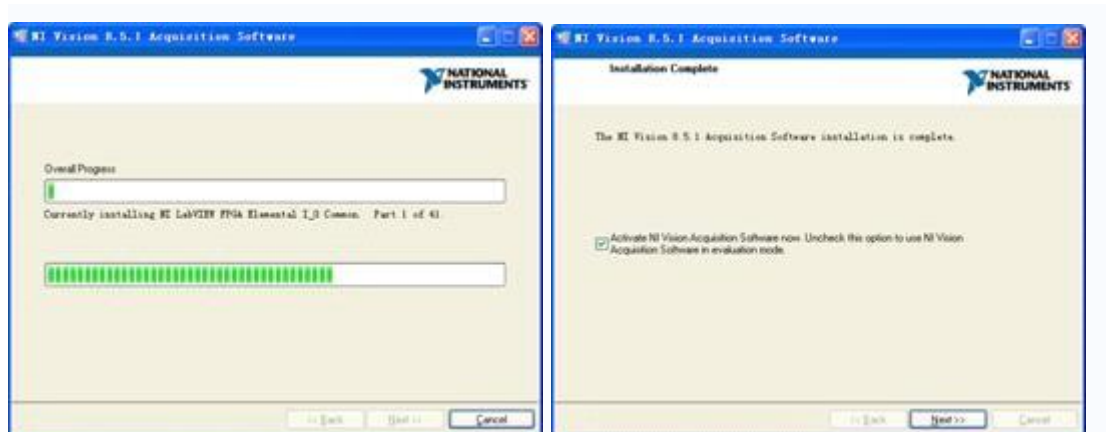


图 10-38 正在安装

图 10-39 安装完成

2 · Vision 介绍

Vision 是一个图像处理的工具包，当安装完成后，启动 LabVIEW 软件，可以在前面板与后面板上看到相应的控件和函数。

(1) 前面板控件

安装完 Vision 后，前面板中会自动出现一个 Vision 选项列表，打开它后可以看到有三个显示控件，如图 10-40 所示。

— IMAQ Image.ctl: 对图像进行分析和处理时用到的一些控件，可用于对图像的类型。图像处理的方式、不同的形态算子，以及颜色的类型的选择等。

— Image Display 用于设置图片显示方式，包括放大、移动、选择等。

— Image Display (Classic) 也用于对图像显示的设置。它只是设置以经典的方式显示图像。

(2) 后面板函数

在后面板中，打开函数选板，可以看到“视觉与运动”下拉列表。打开此列表将显示 Vision 安装后所有的函数，其中包括运动控制方面的函数，如图 10-41 所示。



图 10-40 Vision 显示控件



图 10-41 “视觉与运动”函数

在后面板的函数选板中，可以看到有 5 大类的“视觉与运动”函数。它们主要是一些图像采集和图像处理的函数。

— **NI-IMAQ 列表**：是图像采集函数列表，主要用于通过 NI 的系列图像采集板卡来获得图像。其中包括任务的建立、设备的初始化以及硬件参数的设定等函数，如图 10-42 所示。

— **Vision Utilities 列表**：是视觉应用函数列表，用于对图像进行一些初步的整体操作。它包括的函数节点如图 10-43 所示。

其中，Image Management 是图像管理模块，包括建立和清除图像任务，获取图像的各类信息，图像的类型转换等函数节点，如图 10-44 所示。Files 是图像文件模块，完成对图像文件的读写，以及图像附加信息的读写操作，如图 10-45 所示。



图 10-42 NI-IMAQ 函数列表



图 10-43 Vision Utilities 函数列表



图 10-44 Image Management 函数列表



图 10-45 Files 函数列表

External Display 是图像的外部显示模块，如选中某一光标区域等。Pixel Manipulation 是图像像素处理模块。此模块中的函数节点对图像的像素直接进行操作，包括图像上的点、线、面像素值的获取和设定，以及在图像中插入文本。Overlay 是图像覆盖模块，可以对图像上的某一点、线、面（多边形、矩形和圆）进行覆盖。此种覆盖为非破坏性的覆盖，即不破坏原有的图像，覆盖信息可以另外和图像一起保存。Color Utilities 是颜色应用模块，用于彩色图像中色彩的提取，图像中某点、线、面中色彩的设定或获取，以及不同色彩模型的转换。

10.2.4 图像采集

要处理图像，必须先要获取一张图像。获取图像是通过调用 Vision 中的 IMAQ USB 函数来实现的。它的最大优势在于可以自动识别 USB 摄像头，并读取数据。

选择“函数”|“Vision and Motion”|“IMAQ USB”命令，打开 IMAQ USB 函数下拉列表，如图 10-46 所示。

在进行图像采集时，先要用 IMAQ Create 函数创建一个图像任务，此函数的接线如图 10-47 所示。



图 10-46 图像采集函数

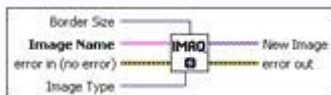


图 10-47 IMAQ Create 函数图标

这里通过此函数对图像的名称和类型大小等进行定义，否则后面的工作没有办法进行。然后再通过一个 USB 启动函数来调用 USB 摄像头设备。此函数是 NI 公司自己研发专门读取 USB 设备的函数，其内部通过调用 ImaqDirectShowDll.dll 文件来实现。

打开 USB 设备后，就要从此设备上获取数据，所以用了一个 IMAQ USB Grab Acquire 函数来实现。整个图像采集程序框图如图 10-48 所示。

10.2.5 图像处理过程

此处的图像处理包括对图像进行采样，找出与采样点相同的图像。为了找出各种角度放置的采样点，在查找的同时对图像进行了 360° 的翻转，这样可以找出图像上所有相同点。

1 · 整体设计

在本实例设计中，对一张图片先设置好要查找的内容，然后开始自动查找，最后对找出的内容进行标记和显示。

由于软件的运行比较复杂，数据的采集又是实时的，要求处理速度比较快，所以要对其进行整体设计，合理安排控件的调用和执行顺序。本程序中采用了一个大循环，保持程序的持续运行。在内部再调用一个顺序结构来控制程序的执行顺序，这样可以保证程序按编程者的思路进行。具体的工作流程图如图 10-49 所示。

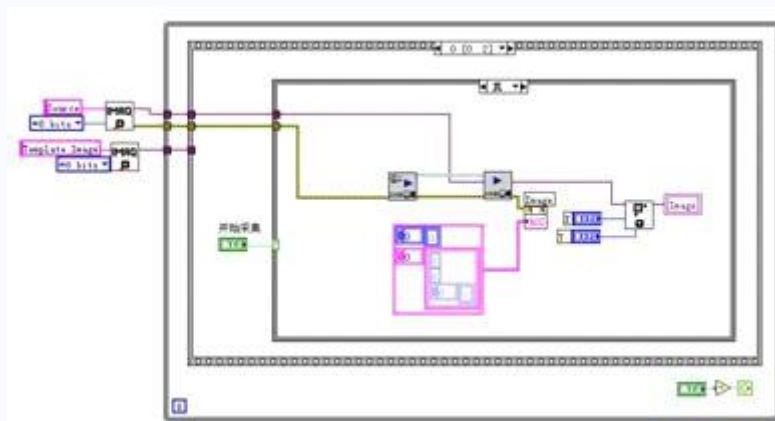


图 10-48 图像采集

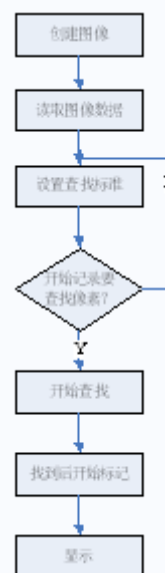


图 10-49 整个程序流程图

2 · 设置查找像素

读取了图像数据后，还要设置查找的像素。这里通过一个光标选择函数来实现。先用函数 IMAQ Setup Learn Pattern 2 来设置需要记录的各项，然后再用 IMAQ Extract 函数进行光标设置。这样就记录了此光标区域的图像数据。它的程序框图如图 10-50 所示。

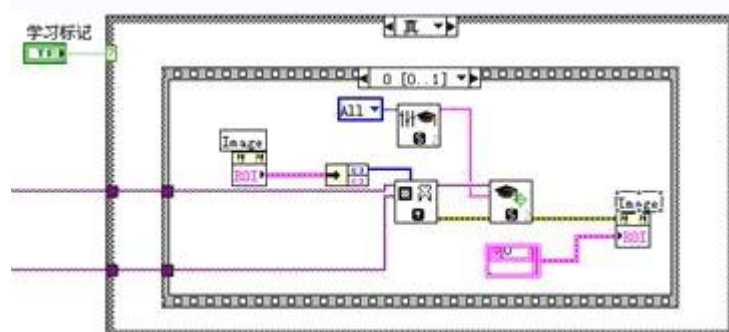


图 10-50 设置查找像素

这里用一个条件结构来控制是否进入记录像素的程序。也就是当选择了要记录的像素后，才进入此分支程序。在这一分支程序中，又利用了一个顺序结构，这样提高了程序运行的效率。它的程序如图 10-51 所示。

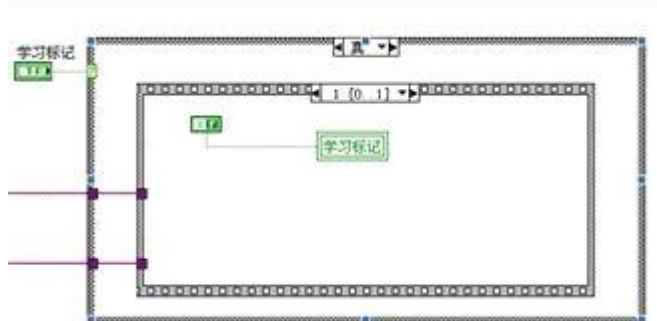


图 10-51 复位记录按钮

3 · 开始查找

当设置完以上要查找的像素后，就可以在需要的图片中查找此像素。为了查找有用的像素，在选择了“开始查找”后，要先读取上面标记的像素，再进行查找。此处程序的设计中，也是先运行一个条件结构，再运行顺序结构，按顺序执行程序。读取光标选择像素的程序如图 10-52 所示。

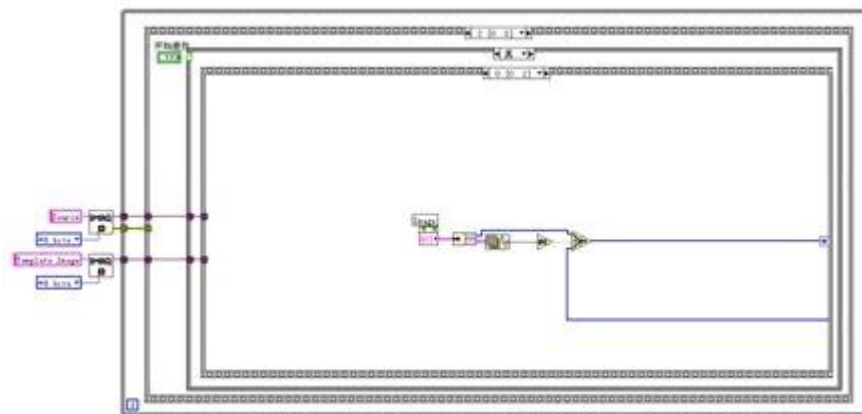


图 10-52 读取选择的像素

当读取像素后，利用顺序结构在第二帧的图像中继续查找。在这一帧中放置了一个循环，并限制循环次数为 4。此时先用一个 IMAQ Rotate 对图像进行翻转，每次翻转 90°。这样就可以在循环 4 次时翻转一周，对图像上各个角度的像素进行查找。再把图像送到 IMAQ Match Pattern 2 函数，对其进行查找。通过此函数直接输出找到的像素信息的数组。为了对找到的信息进行处理，又用一个 For 循环对此数据和簇进行拆分。这里的 For 循环次数直接由数组大小来控制。把数

组拆开后再找到像素的中心位置，并按设置好的字体和格式显示出来。对字体的设置，前面板如图 10-53 所示。

从图 10-52 中可以看出，其中包括字体名称、字体大小、下画线、加粗等标记方式。标记字体程序如图 10-54 所示。



图 10-53 标记字体设置

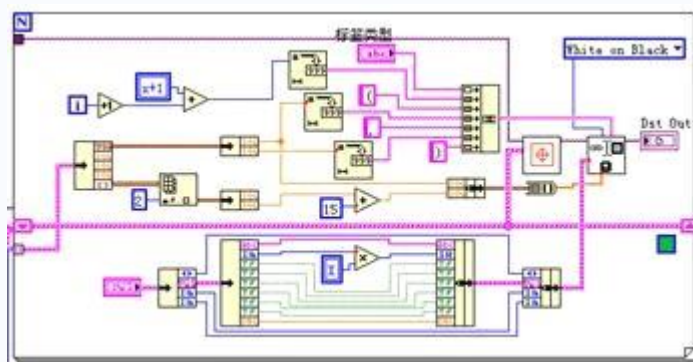


图 10-54 标记字体程序

可以看到，最复杂的设置是对标记内容的组合与写入。这里先由 Draw Pattern Matches Position 函数读取各个记录点的坐标值，然后用一个字符串组合函数把标记和坐标值组合成一个标记字符串。最后由 IMAQ Draw Text 函数把组合好的标记信息直接写入图像界面中。整个标记程序如图 10-55 所示。

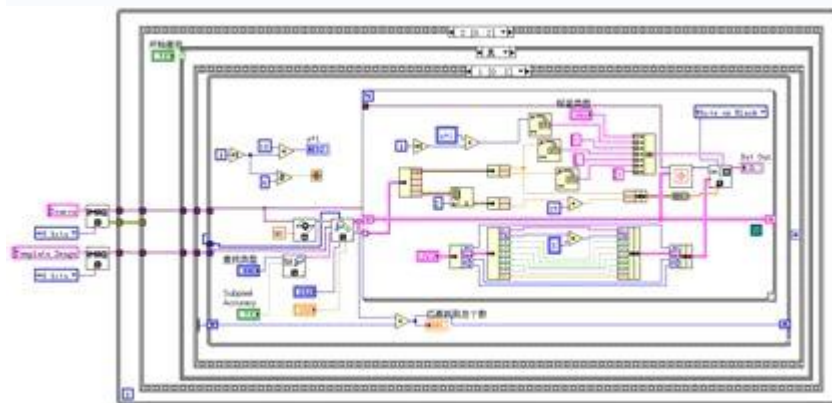


图 10-55 完整标记程序

当第二帧执行完后，进入第三帧的图像大小设置。因为在前面查找的过程中，为了查找得更加精确，对图像进行了像素的放大，这里为了显示的方便，就需要对

图像进行还原，这样才可以看得清楚。此处用了 IMAQ Extract 函数，对显示的图像进行放大和缩小，但是没有改变真正的图像。它的程序框图如图 10-56 所示。

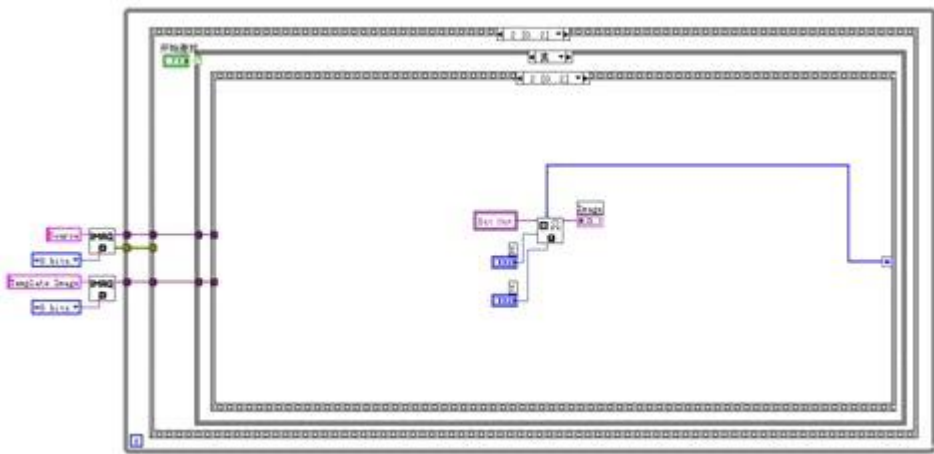


图 10-56 图像还原

4 · 程序前面板

当所有后面板程序编写完后，就要进行前面板控件布局。由于此程序中涉及的控件不多，所以很好设置整体布局。程序界面如图 10-57 所示。其中，“标签类型”是为了给标签进行不同的编号，在当前坐标前面加的一个前缀，这样更好区分不同的像素点。“最大数目”是查找出目标的最大限制。如果设置为 40，则最多只能找出 40 个目标。当大于此值时，则将多出来的其他目标量忽略。

下面对查找时要放大的倍数进行设置，这里只是对二维的数据放大，所以在界面上设置两个值，一个是 X，另一个是 Y。前面板布局如图 10-58 所示。



图 10-57 控制项设置



图 10-58 放大设置

为了对系统的工作进行控制，需要设置 4 个按钮。它们分别如下：

- 开始采集：用于启动采集，完成整个任务的第一个动作，采集一张图片。
- 记录标记：用于对光标选择的区域进行记录，以识别设置像素。

— 开始查找：启动整个系统，对采集到的图片进行查找，找出相同像素并进行标记。

— 停止工作：系统停止工作。

界面布局如图 10-59 所示。

接下来对查找后的图像进行显示，这是非常关键的一个步骤。这里还对查找到的总个数进行了统计，这样可以让人对查找结果一目了然，完整的前面板界面如图 10-60 所示。

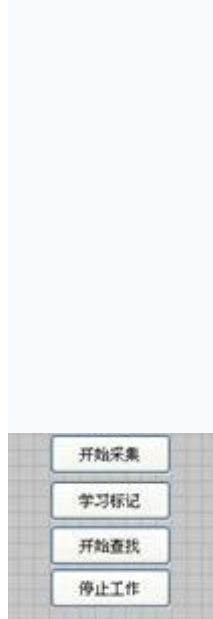


图 10-59 按钮设置

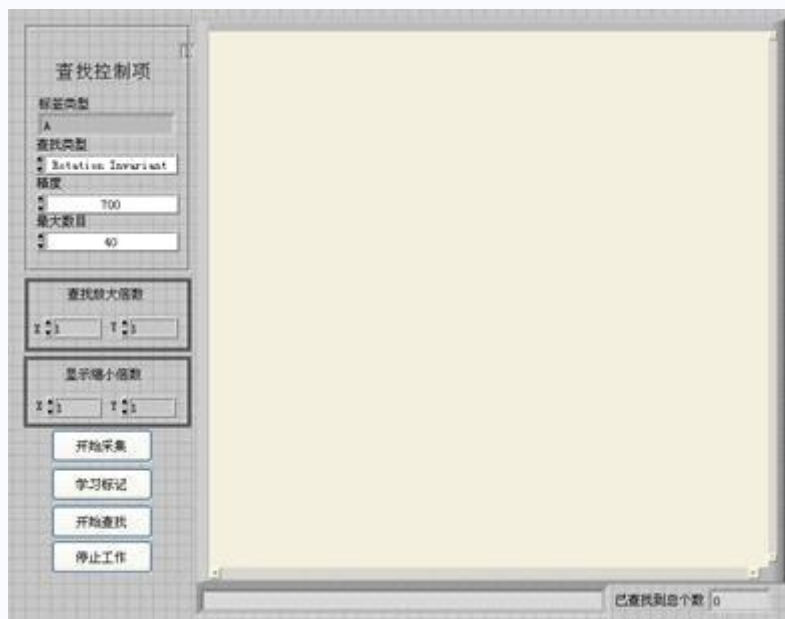


图 10-60 完整界面

10.2.6 程序调试

程序编写完成后，要对系统进行软硬件的联机调试。这里把 NI 公司的 USB 摄像头和计算机连接，并在计算机上安装此软件工具包。具体操作如下：

(1) 运行本程序，在摄像头下放置好一块电路板，并对其设置好焦距和亮度。单击“开始采集”按钮，对其进行采集，并显示实时采集到的图像，如图 10-61 所示。

(2) 当采集完一次后，在界面上可以看到清晰的采集结果。此时我们用光标在需要进行识别的地方画出一个区域。此时光标变为绿色，表示用光标选中了要记录的像素。然后单击“学习标记”按钮，此光标消失，表示已经记录（学习）完成，如图 10-62 所示。

从图上可以看出，这里选中的是电路板上字母 C，让程序学习记录，并找出界面上所有的字母 C。

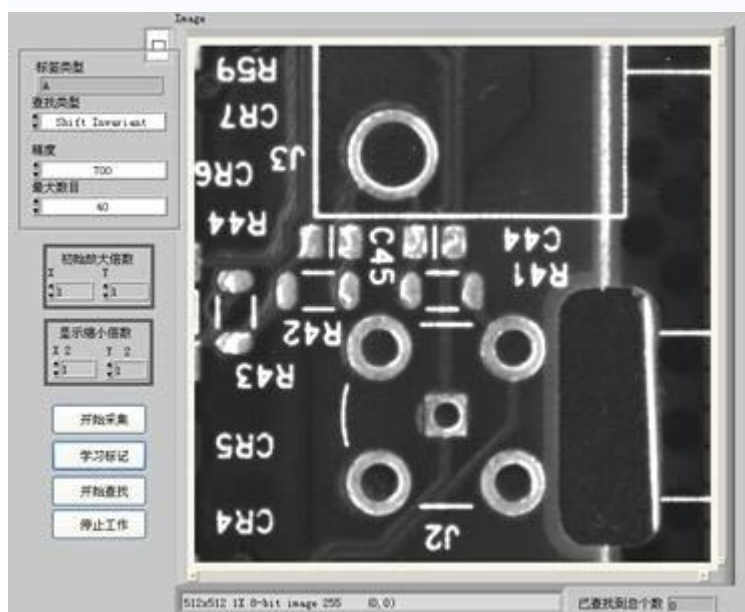


图 10-61 图像采集

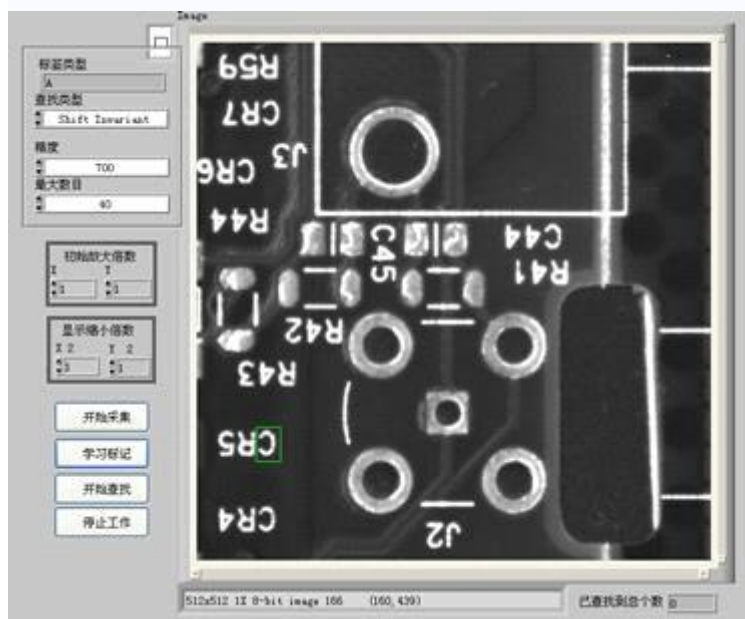


图 10-62 设置学习区域

(3) 记录学习完成后，就可以查找相同像素的点了。单击“开始查找”按钮后，程序开始在此图上查找记录的像素点，并以此像素点为标准，进行比对，找出相同的点。查找的过程中标出了各个点的坐标和编号。这里把标签类型设置为“A”，以 Shift Invariant 的方式查找，精度设置为 700，最大数目设置为 40。这样就可找出所有相同的元素。程序运行时如图 10-63 所示。

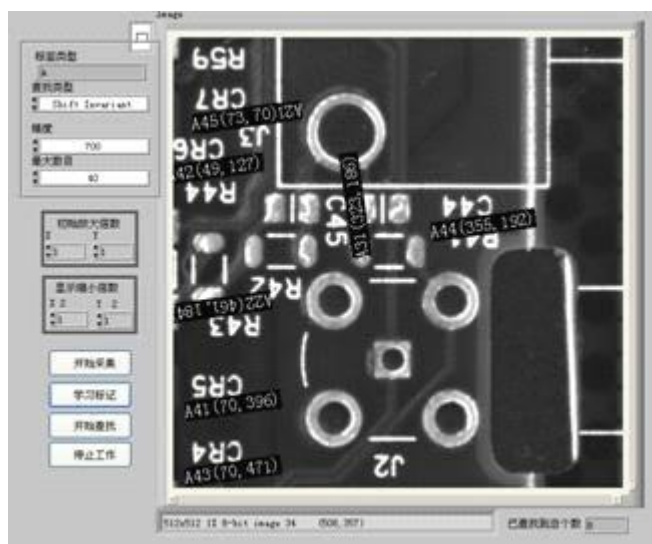


图 10-63 查找结果

从图中可以看出，查找过程中，程序找出了所有字母 C，并标注了它们的坐标和标签。此处的标签有些不是端正的，主要是由于在对图片进行翻转查找时，按相应的角度找到后直接就标记了。如标签 A43（70，471）表示的意义如下：字母 A 是对程序进行查找前设置的标签类型。4 表示此图片翻转到第 4 次找到的，即第 4 个角度。3 表示是此角度下的第 3 个点。（70，471）表示当前查找到像素的中心坐标，即此目标的 $X=70$ ， $Y=471$ ，坐标的值是从图像开始的左上角界面算起的。同时可以看到界面的下面显示了图片的属性，如图片大小、位数等。最后还可以看到一个“已查找到总个数”本框，此处显示的是 8 个目标。这和图上标注的一样，也和实际完全相符。

（4）完成查找后，单击“停止工作”按钮中断此程序的运行。系统退出图像采集和处理环节。

10.2.7 实例总结

本例讲解了 LabVIEW 中图像处理方面的具体应用，主要对如何利用 Vision 工具包采集图像进行了详细的讲解。同时采集到图像后进行了各类复杂的运算，其中编程的思路也是笔者的经验展示，读者需要用心体会，掌握设置不同事件的合理运行顺序。

10.3 通过点阵控制显示图形

在 LabVIEW 中利用点阵的排列可以显示出完美的图形，就像是现实中的各种点阵显示屏一样，可以用来绘制各类数字、文字和图形。本例将介绍如何实现汉字解码显示。

10.3.1 点阵介绍

在 LabVIEW 中显示点阵是一种比较新颖的技术，也是 LabVIEW 程序设计中一个重要的显示技巧。所谓点阵，就是用灯的亮灭来显示各类图形。

10.3.2 前面板

本程序中，要对输入的字符进行点阵的显示，就要先制作一个点阵显示屏。此处用一个圆形指示灯和数组来实现。具体步骤如下：

(1) 在前面板上先通过“控件”|“新式”|“数组、矩阵与簇”|“数组”命令放置一个空的数组，如图 10-64 所示。

(2) 在数组控件内通过“控件”|“新式”|“布尔”|“圆形指示灯”命令放置一个图形指示灯，如图 10-65 所示。



图 10-64 放置数组的方法



图 10-65 放置图形指示灯的方法

把这个布尔量放入数组后，数组会自动变化为圆形指示灯的状态，如图 10-66 所示。此时可以根据数组的边界大小框来设置。在数组边界将光标变化为一个 T

字状态，此时拖动鼠标会改变数组的大小，然后再拖动指示灯的边界，可以设置指示灯在数组中的大小。

（3）显示输入的字符。这里通过一个文本输入控件来实现对输入的控制，如图10-67 所示。

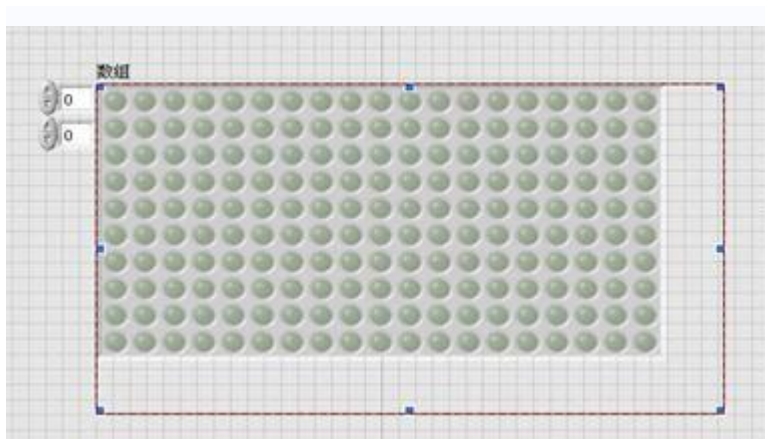


图 10-66 设置点阵大小

(4) 选择输入的字体类型，默认设置为系统字体和类型。此部分的设置界面如图 10-68 所示。在“字体名称”输入框，用户可以选择系统中已安装的一种字体。

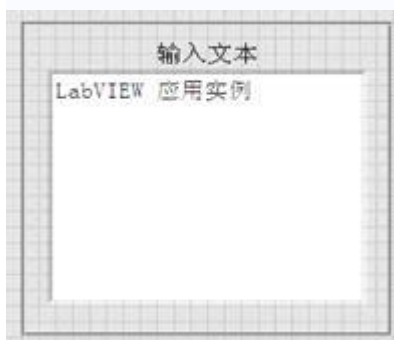


图 10-67 输入文本



图 10-68 用户定义字体

(5) 为了对文字显示的动作进行控制，这里用到了两个按钮：一个用来设置字体滚动的方式是水平还是垂直，另一个用来设置滚动的方向是向上、向下或者向左、向右。它们的界面如图 10-69 所示。同时在此界面中还对字体滚动的速度进行了设置，当改变“移动速度调节”中的数字时，点阵的显示速度就随之改变。



图 10-69 设置显示动作

(6) 当整个前面板设置完成后，对其进行整体布局。它的完整界面如图 10-70 所示。



图 10-70 完整程序界面

10.3.3 程序框图

当前面板设置完毕后，就要编写后面板的程序，下面具体讲解编程步骤。

（1）先对输入的文本进行区别，得出每一行的具体内容。这里通过一个“匹配模式”函数来实现。它的图标如图 10-71 所示。

读取字符的程序如图 10-72 所示。在程序中，对输入的字符进行判断，当输入为空时此循环中断，否则一直循环直至读出最后一个逗号后的值。

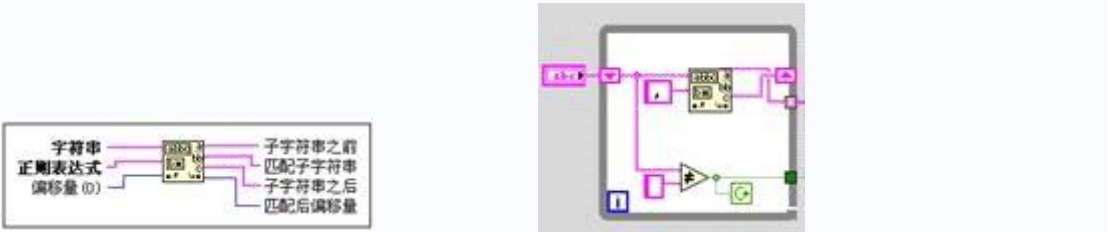


图 10-71 “匹配模式”函数节点图标

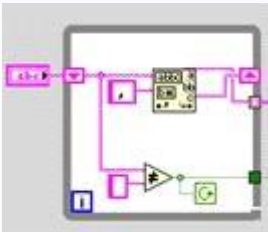


图 10-72 读取字符

（2）接下来要进行字符到图片的设置。这里用“在矩形中绘制文本”函数来实现，它的主要作用是在图片中绘制字符串。此函数的图标如图 10-73 所示。

同时这里在输入前还对输入字体进行了区别。当选择了“用户自定义”时，程序 用一个条件结构选择“用户自定义”并进入其中处理。程序框图如图 10-74 所示。

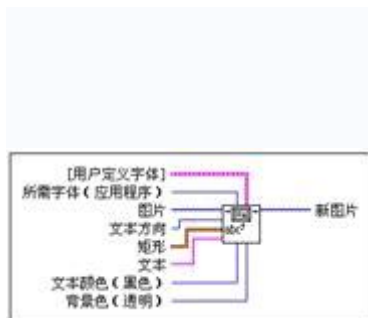


图 10-73 “在矩形中绘制文本”函数节点图标

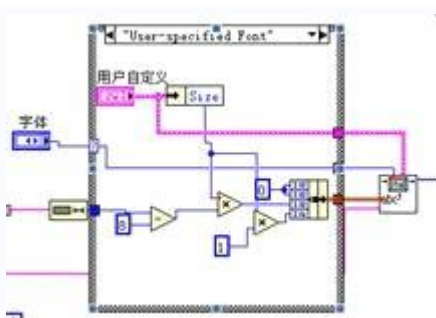


图 10-74 用户自定义

(3) 用户选择了系统字体后，程序如图 10-75 所示。此部分程序的功能是对显示屏的大小进行设置，其他的都没有改变，全是自动调用系统默认值。

(4) 当把字符转换成为图片后，此部分处理已经成功一半，接着还要把图片转换成像素图，这样才能对写入的字符进行控制。此处用的是“图片至像素图转换”函数，它的作用就是将图片转换为图像数据的簇，用于执行与图片相关的任务。它的图标如图 10-76 所示。

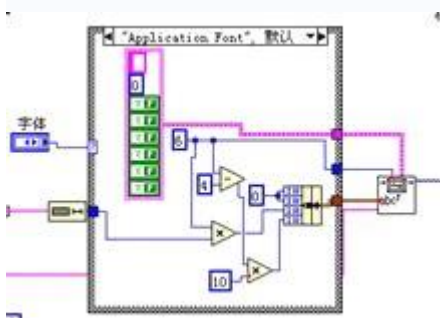


图 10-75 系统自定义

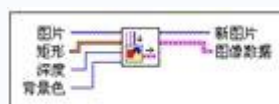


图 10-76 “图片至像素图转换”函数节点图标

标

(5) 当转换为图像数据后，就要对其进行从图像数据簇到二维数组的转换，此处用的是“还原像素图”函数。它的图标如图 10-77 所示。此函数可以把图像数据转换为 4 位、8 位、24 位等几种数组。

(6) 对于上面的步骤中得到的二维数组还要进行转换，得出布尔量，这样才能在布尔指示灯上显示。这里用一个 For 循环来控制所有图片数据。此方法是把图片的二维数组和数字 0 进行比较，当结果为 0 时，输出布尔真，这样就得到了所谓图片对应的布尔量，如图 10-78 所示。

(7) 把图形转换为布尔量后，可以直接送到点阵显示屏显示。这里为了让图形可以移动和变化，需要编程对其进行控制。用一个开关量来控制条件结构，在其

中采用不同的移动方式。当图形为垂直移动方式时，直接运用一个一维数组移位函数，对其进行移动。这样就能完成上下移动。当图形为水平移动方式时，先把图形的二维数组转置，再用一个一维数组移位函数实现移动，这样就形成左右移动的效果。用一个选择函数控制移动的方向，这样就可以控制各个方向的移动显示。程序如图 10-79 所示。

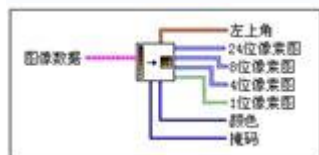


图 10-77 “还原像素图”函数图标

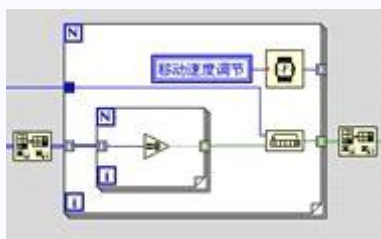


图 10-78 获取布尔量

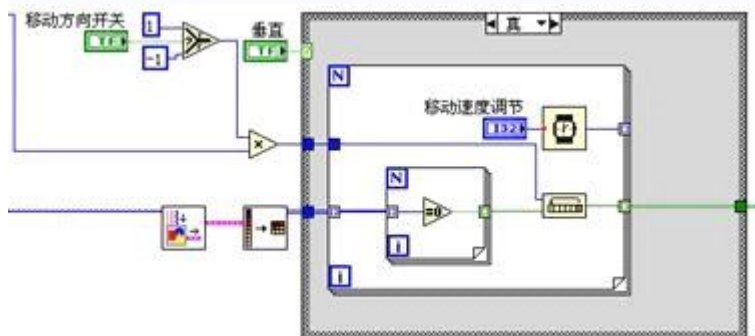


图 10-79 移动方向控制

(8) 为了程序能持续运行, 在后面板中还要添加一个循环结构。它主要保证文字连续移动。完整程序的框图如图 10-80 所示。

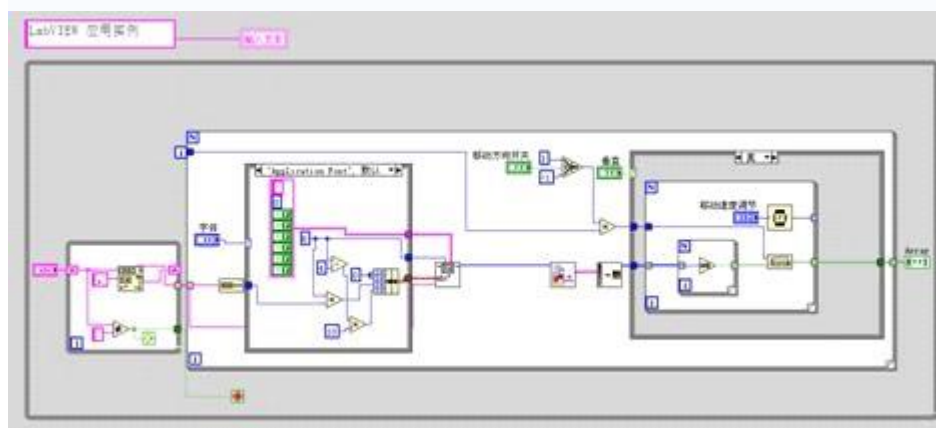


图 10-80 完整程序

从上图中可以看出，本程序利用一个局部变量来控制显示的字体。这是为了让程序在一开始运行时显示默认的值，比如这里显示“LabVIEW 应用实例”几个图形。

10.3.4 程序调试

程序编写完成后，要进行其功能和各方面的测试。运行此程序，可以看到开始时，在点阵显示屏上显示“LabVIEW 应用实例”几个字，如图 10-81 所示。对上下移动和左右移动按钮进行操作，可以看到相应的运行效果。



图 10-81 运行效果

当在“所需字体（应用程序）”文本框中选择“User-specified Font”时，在“字体名称”中写入“黑体”，字号设置为 18。方向设置为水平向左运行，可以看到相应效果如图 10-82 所示。



图 10-82 自定义效果

10.3.5 实例总结

本例讲解了 LabVIEW 中点阵显示和控制的方法，主要对点阵显示的思路和方法进行了介绍。读者通过学习可以了解点阵显示与控制的原理，掌握运用数组转换来实现点阵显示的关键技术。

随着通信与测试技术的发展，LabVIEW 的优势越来越得到体现，LabVIEW 也越来越多地被用来进行网络通信与自动测试开发。本章将具体介绍 4 个典型实例。

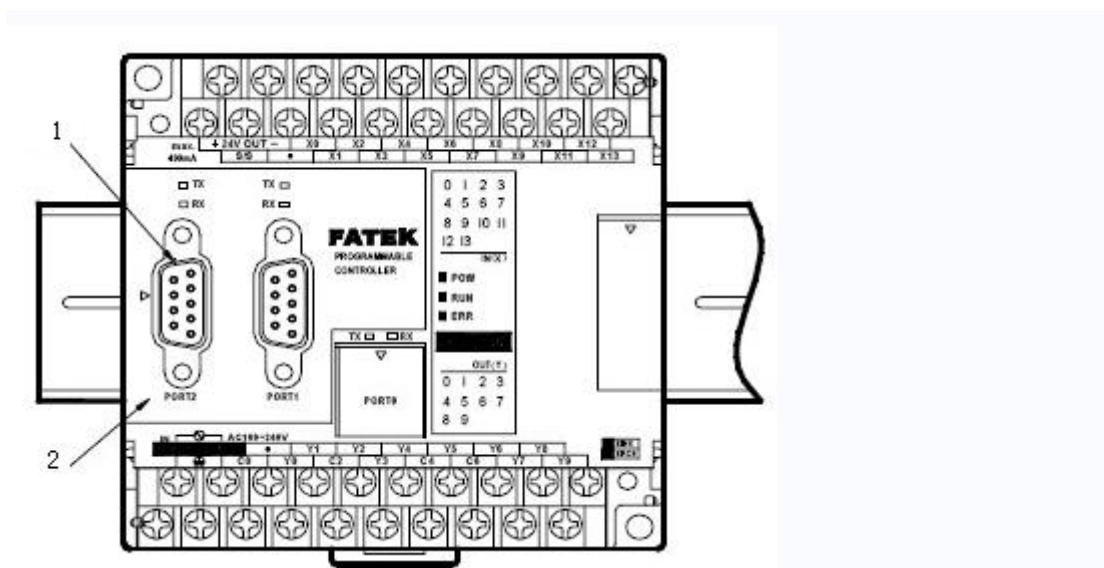
11.1 LabVIEW 与永宏 PLC 通信

PLC 即可编程控制器（Programmable Logic Controller），是一种以计算机技术为基础的新型工业控制装置，它是专门为工业环境应用而设计的数字运算操作的电子装置。它采用可以编制程序的存储器，在其内部存储和执行逻辑运算、顺序运算、计时、计数和算术运算等操作的指令，并能通过数字式或模拟式的输入和输出，控制各种类型的机械设备或生产过程。PLC 及其有关的外围设备都应该按易于与工业控制系统形成一个整体，易于扩展功能的原则而设计。

PLC 通信包括 PLC 间的通信及 PLC 与其他智能设备间的通信。随着计算机控制技术的发展，工厂自动化网络发展得很快，各 PLC 厂商都十分重视 PLC 的通信功能，纷纷推出各自的网络系统。新近生产的 PLC 都具有通信接口，通信非常方便。本章具体介绍 LabVIEW 与永宏 PLC 通信的设计过程。

11.1.1 通信硬件连接

图 11-1 即为永宏 PLC 的硬件图，其中的硬件通信板可以选择，图 11-2 所示为选择的 RS485 与 RS232 的通信接口。



1. FBS-CB22 通信板 (Communication Board 简称 CB)
2. FBS-CB22 通信板对应的盖板 (每一种通信板都有其对应的盖板)

图 11-1 永宏FBS 系列 PLC 硬件

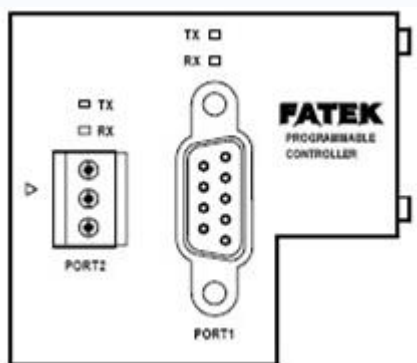


图 11-2 FBS-CB22 RS232 和 RS485 接口通信板

11.1.2 PLC 串口通信原理

PLC 各型主机均内建 2 个通信接口的标准配置，即一个 RS232 和一个 RS485 通信接口，其RS232 接口主要用于上下载程序或用来与上位机、触摸屏通信，而RS485 接口主要用于组建使用 RS485 协议的网络，实现通信控制。

1 · RS232 接口

RS232-C 接口连接器一般使用型号为 DB-9 的 9 芯插头座，只需 3 条接口线，即“发送数据”、“接收数据”和“信号地”即可传输数据，其 9 个引脚的定义如图 11-3 所示。



图 11-3 RS232-C 接口连接器定义

在 RS232 的规范中, 电压值在 $+3V \sim +15V$ (一般使用 $+6V$) 称为“0”或“ON”。电压在 $-3V \sim -15V$ (一般使用 $-6V$) 称为“1”或“OFF”; 计算机上的 RS232 “高电位”约为 $9V$, 而“低电位”则约为 $-9V$ 。

RS232 为全双工工作模式, 其信号的电压是参考地线而得到的, 可以同时进行数据的传送和接收。在实际应用中采用 RS232 接口, 信号的传输距离可以达到 $15m$ 。不过 RS232 只具有单站功能, 即一对一通信。

2 · RS485 接口

RS485 采用正负两根信号线作为传输线路。两线间的电压差为 $+2V \sim 6V$ 表示逻辑“1”; 两线间的电压差为 $-2V \sim 6V$ 表示逻辑“0”。

RS485 为半双工工作模式, 其信号由正负两条线路信号准位相减而得, 是差分输入方式, 抗共模干扰能力强, 即抗噪声干扰性好; 实际应用中其传输距离可达 1200 米。RS485 具有多站能力, 即一对多的主从通信。

在串行通信中, 数据通常是在两个站之间传送, 按照数据在通信线路上的传送方向可分为 3 种基本的传送方式: 单工、半双工和全双工, 如图 11-4 所示。

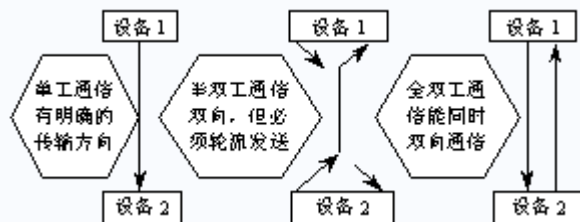


图 11-4 单工、半双工和全双工通信

单工通信使用一根导线, 信号的传送方和接收方有明确的方向性。也就是说, 通信只在一个方向上进行。

若使用同一根传输线既作为接收线路又作为发送线路，虽然数据可以在两个方向上传送，但通信双方不能同时收发数据，这样的传送方式称为半双工。采用半双工方式时，通信系统每一端的发送器和接收器，通过收发开关分时转接到通信线上，进行方向的切换。

当数据的发送和接收，分别由两根不同的传输线传送时，通信双方都能在同一时刻进行发送和接收操作，这样的传送方式就是全双工。在全双工方式下，通信系统的每一端都设置了发送器和接收器，因此，能控制数据同时在两个方向上传输。

全双工方式无须进行方向的切换。

串行通信可分为两种类型，一种是同步通信，另一种是异步通信。采用同步通信时，将所有字符组成一个组，这样，字符可以一个接一个地传输，但是，在每组信息的开始要加上同步字符，在没有信息要传输时，填上空字符，因为同步传输不允许有空隙。采用异步通信时，两个字符之间的传输间隔是任意的，所以，每个字符的前后都要用一些数据位来作为分隔位。比较起来，在传输率相同时，同步通信方式下的信息有效率要比异步方式高，因为同步方式的非数据信息比例比较小。但是，从另一方面看，同步方式要求进行信息传输的双方必须用同一个时钟进行协调，正是这个时钟确定了同步串行传输过程中每一个信息位的位置。这样一来，如果采用同步方式，那么，在传输数据的同时，还必须传输时钟信号。而在异步方式下，接收方的时钟频率和发送方的时钟频率不必完全一样，而只要比较相近，即不超过一定的允许范围就行了。在数据传输中，较为广泛采用的是异步通信，异步通信的标准数据格式如图 11-5 所示。

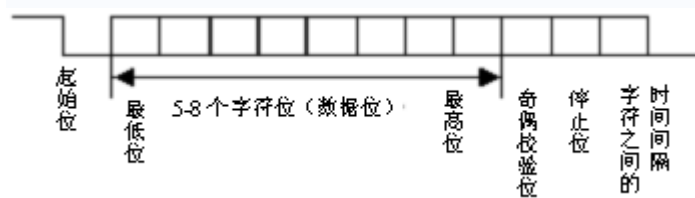


图 11-5 异步通信数据格式

从图 11-5 所列格式可以看出，异步通信的特点是一个字符一个字符地传输，并且每个字符的传送总是以起始位开始，以停止位结束，字符之间没有固定的时间间隔要求。每一次有一个起始位，紧接着是 5~8 个的数据位，再后为校验位，可以是奇检验，也可以是偶校验，也可不设置，最后是 1 比特，或 1 比特半，或 2

比特的停止位，停止位后面是不定长度的空闲位。停止位和空闲位都规定为高电平，这样就保证起始位开始处一定有一个下降沿，以此标识开始传送数据。

11.1.3 永宏 PLC 通信协议

永宏通信协议可以简单通过串口来实现，上位机的具体参数如图 11-6 所示。永宏 PLC 除了拥有自己的永宏标准通信协议以外，还支持 Modbus 通信协议（Porto 除外），具体的通信步骤如下：

通信界面	Port0(RS232 或USB)	通信速率 4.8 kbps ~ 921.6 kbps(9.6 kbps)	
	Port1~Port4 (RS232、RS485 或Ethernet)	通信速率 4.8 kbps ~ 921.6 kbps(9.6 kbps)	Port1~4 可提供永宏或 Modbus RTU Master/Slave 通信协议
	最大联机站数	254	

图 11-6 永宏PLC 通信参数

- (1) 设定外围设备的参数。
- 控制方式（如启动、停止）：通信控制。

— 频率给定方式：通信设置。

— 设置通信参数：波特率、数据位、校验方式、停止位、RTU 或者ASCII 方式。

— 站号。
- (2) PLC 同该设备的 RS485 接线。
- (3) 改变设备的控制地址。找到所需要的控制地址，然后变成永宏 Modbus 地址。
- (4) 设定 PLC 通信接口的通信参数和通信方式，PLC 和该设备上配置的参数要一致。
- (5) 在 PLC 内写 M-BUS 指令，并填写所需要的相应的通信表格。
- (6) 程序写好后，接通电源，进行调试。

11.1.4 程序结构

本例程序采用循序结构控制，程序流程框图如图 11-7 所示，主要由打开串口、设置串口、向串口写命令、读返回值、关闭串口 4 部分组成。

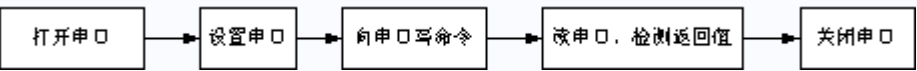


图 11-7 程序流程

程序中对串口的操作使用的是 VISA。

VISA 是虚拟仪器软件结构体系 (Virtual Instrument Software Architecture) 的简称。VISA 是在所有 LabVIEW 工作平台上控制 VXI、GPIB、RS232 以及其他种类仪器的单接口程序库，是组成 VXI Plug & Play 系统联盟的 35 家最大的仪器仪表公司所统一采用的标准。采用 VISA 标准后，就可以不考虑时间及仪器 I/O 选择项，驱动软件可以相互兼容。VISA 的功能模块包含在 LabVIEW Instrument I/O 的 VISA 子模板中。

VISA 驻留于计算机系统中，是计算机与仪器之间的软件连接层，用以实现对仪器的控制。VISA 对于测试软件开发者来说是一个可调用的操作函数集，它本身不提供仪器编程能力，只是一个高层 API（应用程序接口），通过调用低层的驱动程序来控制仪器。NI-VISA 的层次如图 11-8 所示。

VISA 的内部结构是一个先进的、面向对象的结构，这一结构使得 VISA 和在它之前的 I/O 控制软件相比，在接口无关性、可扩展性和功能上都有很大提高。VISA 主要由 Config 模块、Write 模块、Read 模块、Close 模块组成，下面对这些模块做详细介绍。

VISA Config 模块对 VISA 资源信息如波特率、通信端口名称、数据校验、数据位等进行配置，并通过 VISA resource Name 端口将打开的 VISA 资源名称传递给下一个节点。

VISA Write 模块把 write buffer 中的字符串写入 VISA session 指定的设备中。dup VISA session 向下传送相同的 session 值。在 UNIX 工作平台上，数据同步写入；在其他工作平台上，数据异步写入。return count 返回实际传送的字节数。error in 和 error out 字符串用于说明出错状况。

VISA Read 模块读取由 VISA session 指定设备中的数据。byte count 指明读入 read buffer 中的字节数，当收到的字节数小于 read buffer 中所指定的字节数时，VISA read 将会一直等待，直到收到的字节数等于 read buffer 中所指定的字节数。VISA session 向下传送相同的 session 值。同样，在 UNIX 工作平台上，数据同步读入；其他平台上的数据异步读入。

VISA Close 模块关闭由 VISA session 指定设备的通信过程，释放系统资源。VISA 仪器控制流程见图 11-9。图 11-10 为串口读写程序示意图。

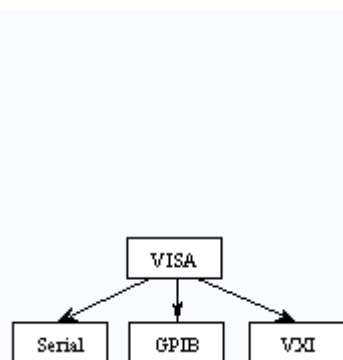


图 11-8 VISA 结构层次



图 11-9 VISA 仪 器

控制流程

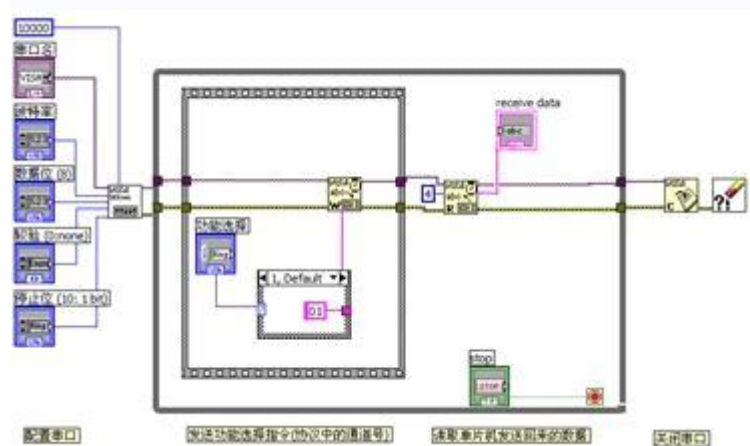


图 11-10 串口读写程序

11.1.5 程序编写

基于 LabVIEW 的 PC 与 PLC 串口通信程序的面板如图 11-11 所示。

根据永宏 PLC 的通信协议及数据操作流程，本程序采用顺序结构。顺序结构可实现数据流的顺序流动，首先打开串口和设置串口参数（如图 11-12 所示），波特率为 9600 Bps，串口号为 COM1，偶校验（Even），7 位数据位，数据停止位为 1 0（如图 11-13 所示）。



图 11-11 程序面板

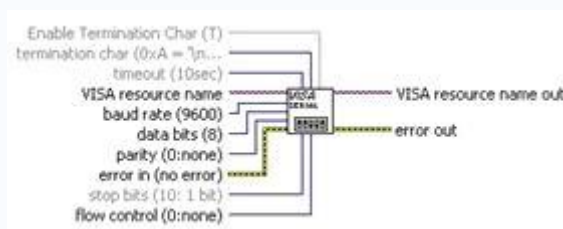


图 11-12 VISA 串口配置

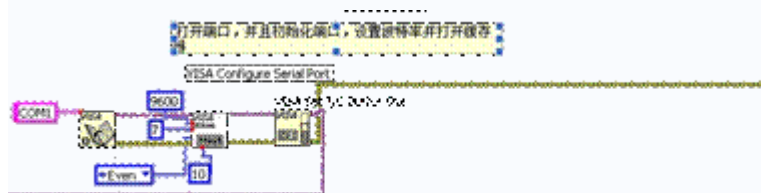


图 11-13 开串口和设置串口参数

第二步写入命令。先要对命令进行判断，看其是否合法，后面板程序如图 11-14 所示。然后，判断写入的命令是否超长，并将合法的数据组合成可写入的命令，后面板程序如图 11-15 所示。

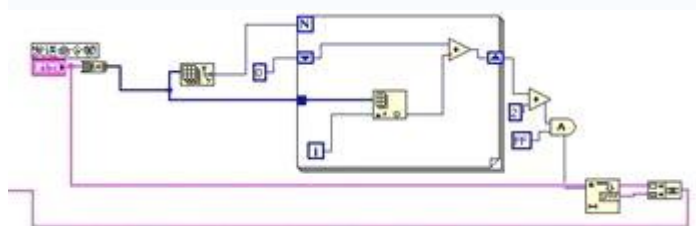


图 11-14 检查输入的命令合法性

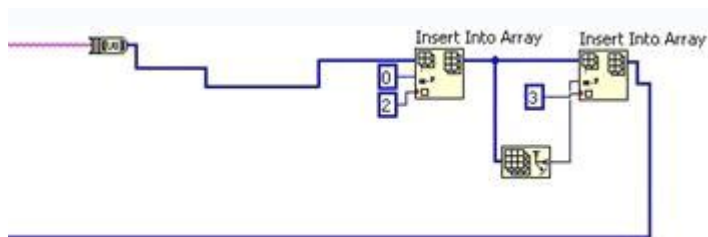


图 11-15 根据通信协议进行数据组合

串口写入过程需要时间，所以在这一步之后需要延时 250ms，如图 11-16 所示。

接下来就是读串口返回值。通过 VISA 读取返回的数值（如图 11-17 所示），然后进行分析检验，判断所接收的数据是否正确。

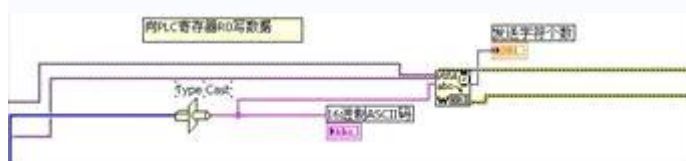


图 11-16 将数据写入PLC 的 R0 寄存器

检验数据之后这个操作流程就结束了，可以关闭 VISA 串口（如图 11-18 所示）。

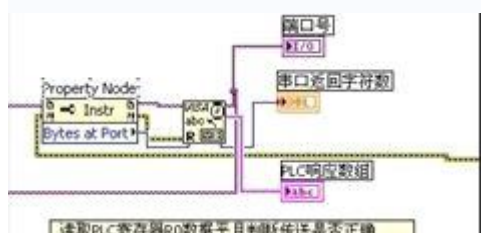


图 11-17 读串口返回值

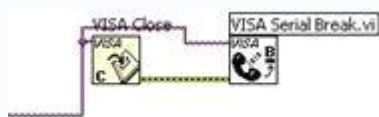


图 11-18 关闭

串口

11.1.6 实例小结

本例运用LabVIEW 驱动配置方法通过VISA 建立了上位机与永宏PLC 通信的程序，对于其他品牌和型号的 PLC 通信，同样可以使用该方法。同样，Modbus 协议的设备间通信也可以使用此类方法。

10.4 仪器串行控制智能小车

仪器串行控制就是用串行通信的方式控制系统。串行通信是一种重要通信方式，它广泛应用于各类场合。下面通过具体的实例来讲解。

10.4.1 实例内容说明

本例所讲解的例子为智能玩具小车在 LabVIEW 程序的控制和监视下自动运行的系统。首先在系统中实现对小车的监视，再对小车进行控制，让其在控制下按要求运动。这样就实现了系统的自动化运行。

1 · 仪器的控制

仪器的控制就是通过上位的控制，让下位能按要求执行有关的动作。在 LabVIEW 中经常用的是对下位的监视、数据采集分析等。这里不但对下位进行了监视，还进行了一些控制。这也是对 LabVIEW 应用的一种创新。

用 LabVIEW 实现仪器控制有许多种方式，这里用到的是 PID 控制。这种控制的最大好处是能对系统的情况作出较为稳定的决策。本系统采用单闭环、双控制器的控制策略，其中双控制器为模糊控制器和 PID 控制器，分别控制小车的速度和舵机角度，另外有一个异常处理器，用来处理小车可能出现的异常情况。其中只有 PID 控制是在 LabVIEW 中应用的，其他控制算法是在下位上用单片机控制实现的。本节给出的是将单片机和 LabVIEW 通信结合实现的实例，这里侧重讲解的是上位 LabVIEW 编程及对仪器的控制这一部分内容。

2 · USB 串行通信

USB 是英文 Universal Serial Bus 的缩写，含义是“通用串行总线”。它不是一种新的总线标准，而是应用在计算机领域的接口技术。USB 用一个 4 针插头作为标准插头，采用菊花链形式可以把所有的外设连接起来，最多可以连接 127 个外部设备，并且不会损失带宽。USB 需要主机硬件、操作系统和外设三个方面的支持才能工作。目前的主板一般都采用支持 USB 功能的控制芯片组，主板上也安装有 USB 接口插座，而且除了背板的插座之外，主板上还预留有 USB 插针，可以通过连线接到机箱前面作为前置 USB 接口以方便使用。

本例中用 USB 的好处是操作方便，在任何计算机上都可以实现。不像 RS232 在一些台式机和笔记本上就没有，会产生很多的不便。

3 · 智能小车

在本例设计中，小车是利用电动玩具车改装得来的。在下位装了一个单片机，用来控制小车的运行速度及方向。同时也连接了一个无线接收模块，是和上位机的 USB 无线模块配套的。这样就实现了小车的无线监控。

下位由蓄电池供电，提供动力和单片机等电子器件的能量。上位只在计算机的 USB 上连接一个无线收发模块，并在计算机上装 LabVIEW 软件。这样就构成了一个完整的仪器串行控制智能小车系统。

10.4.2 开发思路分析

在本例中只讲解上位的开发过程，也就是 USB 通信和 LabVIEW 软件编程这一块。在开发过程中，先进行 USB 设置，包括 USB 驱动安装，再进行整个系统的编程，包括系统管理界面、监视界面以及最后的系统登录界面的设计。

USB 设置要用到 VISA 控件，所以要安装 VISA 组件包。整个程序控制界面中，用到了 PID 控制算法，因此还要安装 PID 组件包。在编程前先安装所有用到的这些组件工具，再从最小的模块开始编程。

10.4.3 具体开发过程

下面将详细介绍 VISA 驱动的安装及编程过程。本例的界面开发框架如图 10-83 所示。

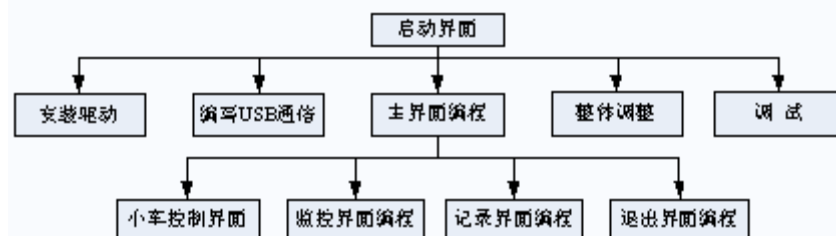


图 10-83 程序开发框架

1 · VISA 驱动安装

VISA 是 Virtual Instrument Software Architecture 的简称。它是一个用来与各种仪器总线进行通信的高级应用编程接口（API），不受平台、总线和环境的限制。USB 是一种串行通信方式，每个 USB 设备都有各自的指令集，可以使用 NI-VISA 的读写功能向仪器发送这些指令，并读取仪器的反馈。用户可向仪器制造商索要仪器的有效指令清单。

VISA 的安装及 USB 的配置步骤如下：

（1）将 VISA 驱动光盘放入计算机光驱，系统将自动打开安装界面，并自行加载一些光盘内容，如图 10-84 所示。

（2）加载完后，单击“Next”按钮，进入安装路径选择界面，如图 10-85 所示。

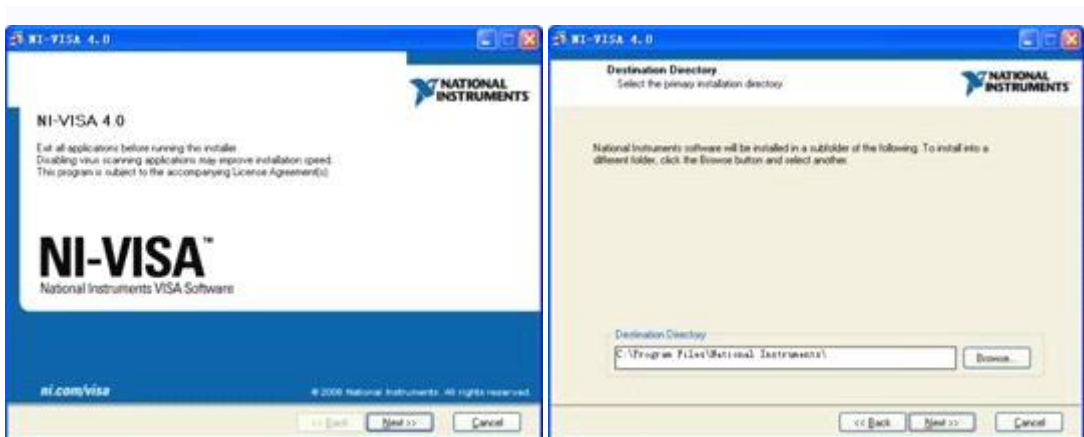


图 10-84 VISA 启动

图 10-85 选择 VISA 安装路径

(3) 默认安装路径是在 C 盘，用户自己可以按需要调整，然后单击“Next”按钮，进入组件选择界面，如图 10-86 所示。

(4) 按需要选择要安装的组件，单击“Next”按钮，进入软件许可协议界面，如图 10-87 所示。

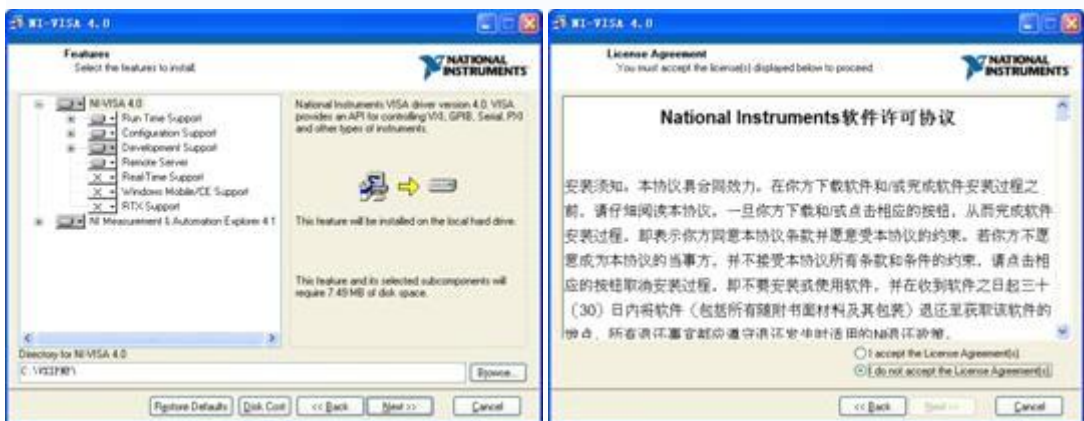


图 10-86 选择组件

图 10-87 许可协议

(5) 选择“I accept the license Agreement(s)”，此时“Next”按钮变为可用，并单击此按钮进入安装提示界面，如图 10-88 所示。

(6) 单击“Next”按钮，出现安装界面，显示安装进度条，如图 10-89 所示。

(7) 安装完成后，出现安装成功提示界面，如图 10-90 所示。

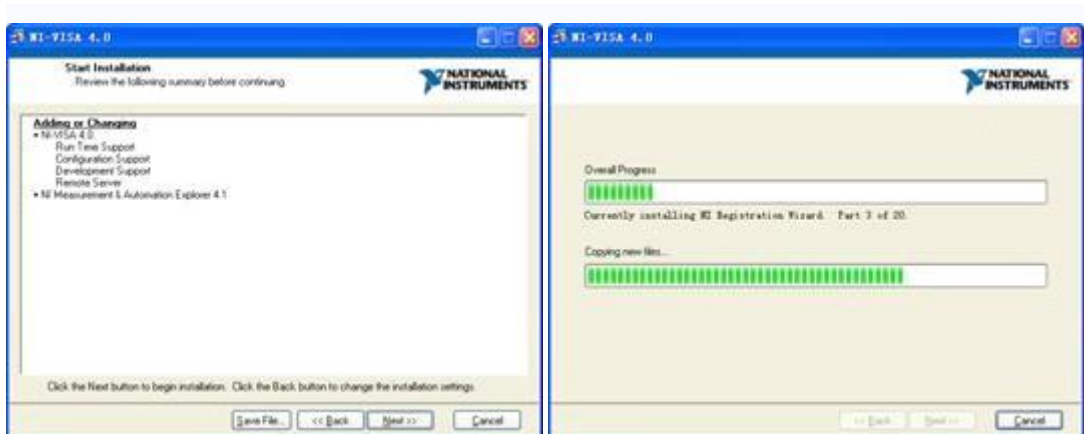


图 10-88 安装提示

图 10-89 安装进度

(8) 单击“Finish”按钮后，出现对计算机的操作选择对话框，一般选择第一项“Restart”，重启计算机即可，如图 10-91 所示。



图 10-90 安装成功

图 10-91 选择重启计算机

2 · VISA 配置

安装好 VISA 后，必须对 VISA 进行配置，才能在 LabVIEW 中调用 USB 通信模块。下面讲解 VISA 和 USB 配置的具体过程。

(1) 单击“开始”|“所有程序”|“National Instruments”|“VISA”|“VISA Driver Development Wizard”命令，弹出“VISA Driver Development Wizard”设置界面，如图 10-92 所示。

(2) 在单选框中选择“USB”，单击“下一步”按钮，进入 VISA DDW 基本设备信息设置窗口，如图 10-93 所示。

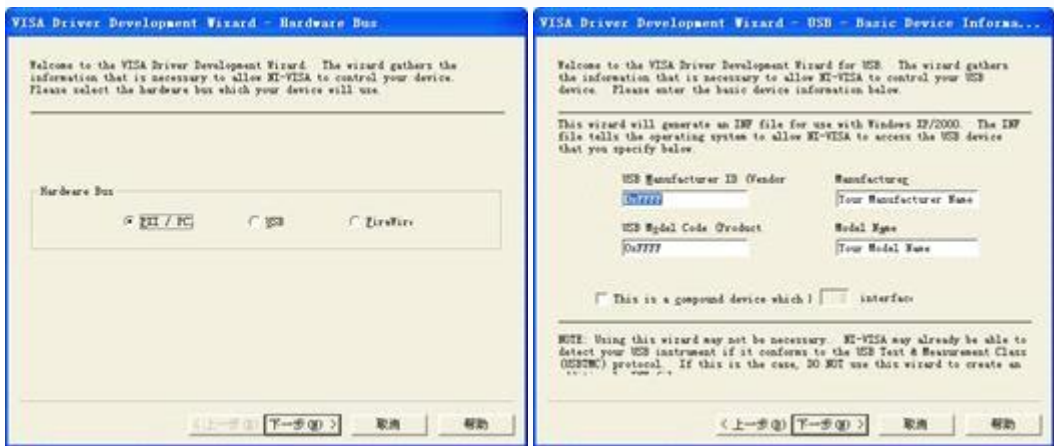


图 10-92 VISA 驱动设置

图 10-93 VISA USB 信息设置

(3) 在上图所示的界面中，主要是对 USB ID 和 Code 进行设置，这是通信中对 USB 的标识。这两组数据可以从要应用的 USB 器件上查找到，若上面没有标注，也可以向厂家询问。设置好以后单击“下一步”按钮，进入 INF 信息生成界面，如图 10-94 所示。

(4) 可以在图 10-93 所示的界面中修改 INF 文件的名称和地址，修改完以后单击“下一步”按钮，出现确认对话框，如图 10-95 所示。



图 10-94 生成 INF 信息



图 10-95 信息确认

(5) 查看生成信息是正确的后，单击“是”按钮，出现组件选择界面，单击“确定”按钮即可完成 USB 的所有设置工作。

(6) 下面就可以安装生成的 INF 文件，选择生成的 INF 文件夹，将其复制到操作系统的 INF 文件夹中。在 Windows XP 操作系统中，这个文件夹的位置通常是：C:\WINDOWS\INF。由于 INF 文件夹可能是隐藏的，因此需要将文件夹选项改成

“显示隐藏文件”。复制到系统 INF 文件夹后，单击鼠标右键，选择“安装”命令即可开始安装。

3 · PID 安装

PID 模块在安装 LabVIEW 时并没有被一起安装，需要用户自行安装，下面对 PID 的安装进行详细叙述。

(1) 插入 PID 安装盘，系统将自动运行安装程序并弹出安装界面，如图 10-96 所示。

(2) 单击“Next”按钮，进入安装组件选择界面，如图 10-97 所示。

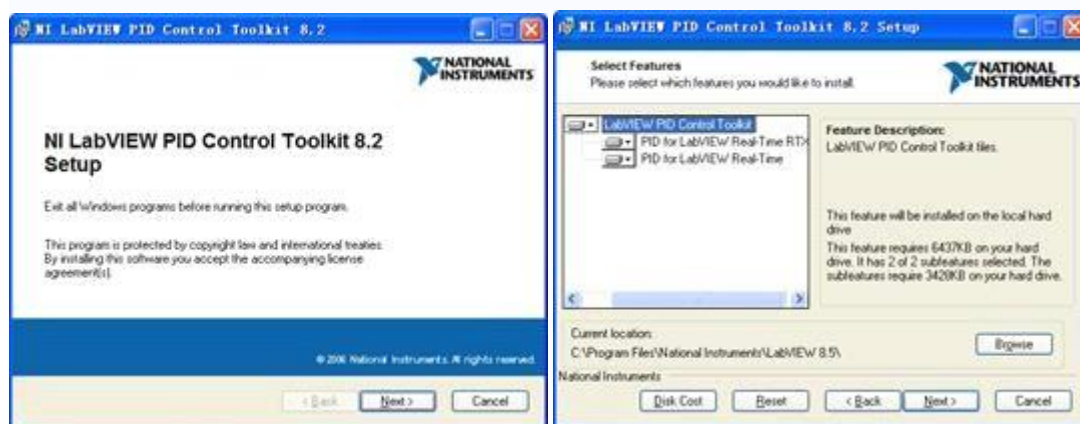


图 10-96 PID 安装界面

图 10-97 组件选择

(3) 在图 10-96 中可以看到有两个组件可以安装。通过下拉选项执行安装选择，完成后单击“Next”按钮，进入信息确认对话框，如图 10-98 所示。

(4) 单击“Next”按钮，进入安装进度界面，如图 10-99 所示。

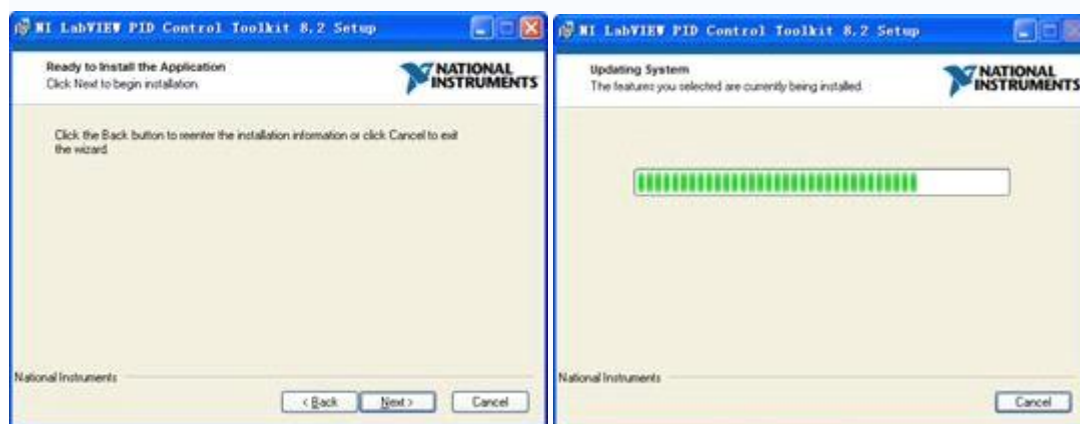


图 10-98 信息确认框

图 10-99 安装进度

(5) 安装完成后，单击“Finish”按钮，安装完成，如图 10-100 所示。



图 10-100 安装完成

4 · 采集模块编程

将采集到的数据传送到 USB 后，要从 USB 上读取数据以便分析和处理。下面将讲解从 USB 读取数据以及对端口界面进行设置的编辑。

(1) 从端口读取数据时, 要进行 VISA 设置, 所以要放置一个 VISA 对 USB 的操作控件, 主要用于打开 VISA, 写入 USB, 读出 USB 等。下面先配置 VISA 端口, 打开 VISA 端口, 如图 10-101 所示。

端口的配置包括设置 VISA 资源名称、数据位数、停止位等，并设置对前面板的装饰，将对各控件放置整齐，并加一个底框着色。通信端口设置完成后前面板界面如图 10-102 所示。

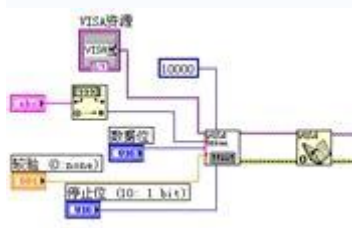


图 10-101 VISA 端口配置



图 10-102 “通信端口设置”界面

(2) 然后进行端口的读写操作。本例中对端口的读写是通过一个层叠式顺序结构来控制的。当 VISA 端口打开后, 先读取初始化值, 并通过“字节数组至字符串转换”函数对初始化值转换后传送到分析处理模块, 再显示数据, 在进行分析决策后发出指令。

这一步只设置 VISA 对 USB 的读写转换及操作。在顺序结构的第 0 帧中设置为对 USB 端口的读取，如图 10-103 所示。

在第 1 帧中实现对 USB 端口的写入，它主要是把要发送的一些内容写入端口。VISA 函数读写操作完成后，要关闭 VISA 函数，以节省系统资源。如图 10-104 所示。

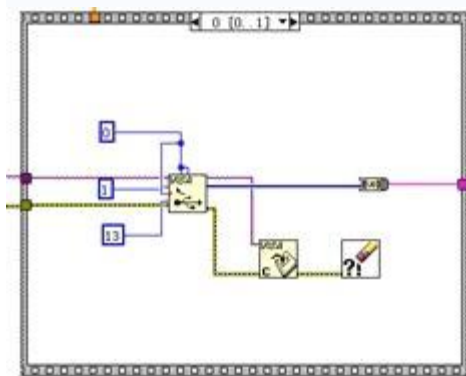


图 10-103 VISA 读取 USB 数据

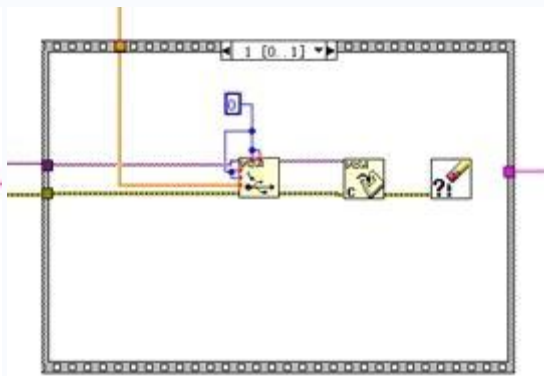


图 10-104 VISA 写入程序

5 · 主界面编程

下面详细讲解主体界面编程，主要涉及一些常用控件的应用，以及一些装饰控件的布局安排技巧。

(1) 编写小车参数设置界面。按控制需要创建一个界面，系统运行后在此界面中用户按要求写入参数后就可以控制下位小车运行。先在前面板上放置一些要用的控件，再对其进行装饰和布局安排。本例用到 6 个数值控件，用于输入各类信息。中间有一个旋钮，用来控制小车报警的音量。两个按钮分别用来控制小车的夜间行驶和是否需要报警，还有一个布尔控件用来整体控制小车的运行状态。前面板界面如图 10-105 所示。

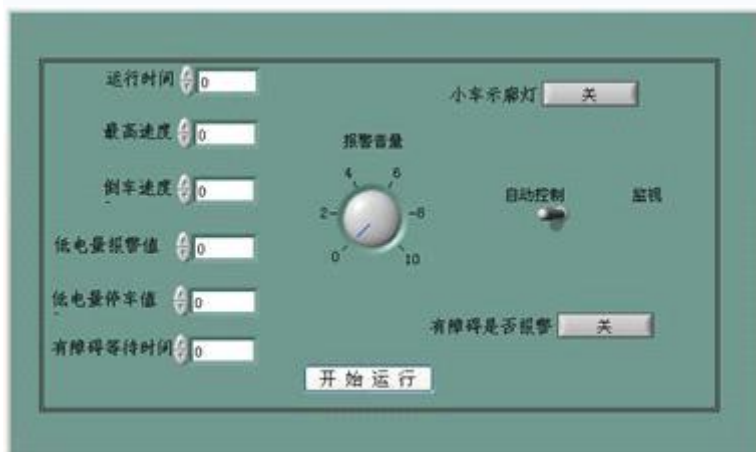


图 10-105 小车控制界面

(2) 编写监控界面。为了能对小车的运行情况进行监视，要从下位读取数据并处理，让界面上显示出小车的基本运行情况。我们先放置好控件，再进行布局和

装饰。监视界面上最多的是指示灯，用来指示各运行状态；还有一个速度转盘，用来形象地显示速度；最后用一个波形图显示小车运行时偏离正常轨道情况。监控界面如图 10-106 所示。

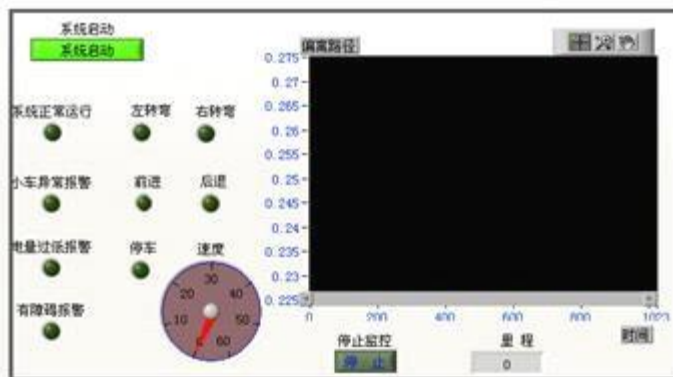


图 10-106 监控界面

(3) 查看历史记录。这个界面主要用来查看系统运行的记录。系统运行时单击“读取”按钮，就可读出记录，如图 10-107 所示。

(4) 系统退出，这是操作完后本系统的退出界面，如图 10-108 所示。



图 10-107 读取记录

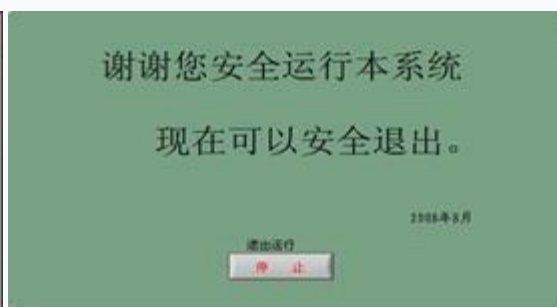


图 10-108 退出系统

6 · 监视程序

监视程序是本系统的首要功能。做好前面板后，就要对其后面板进行编程。在前面板找到连线控件，双击以转到程序框图界面，并显示接线端。

(1) 先对从 USB 上读取的数据进行分解，不同的字段代表不同的控制对象。对字符的识别用“截取字符串”函数来实现。此函数通过“函数”|“编程”|“字符串”|“截取字符串”命令调用。截取特定长度的字符串后，调用一个把字符串转换为数字的子程序，如图 10-109 所示。

(2) 在小车的参数设置中, 包括一个小车电量最低限设置。它用来控制小车在电量低于此值时执行相关动作, 在监控界面有一个小车的电量指示灯。这一块的程序框图如图 10-110 所示。低电量设置值在这里设置为局部变量, 主要是考虑到这个值在后面要输出到 USB。

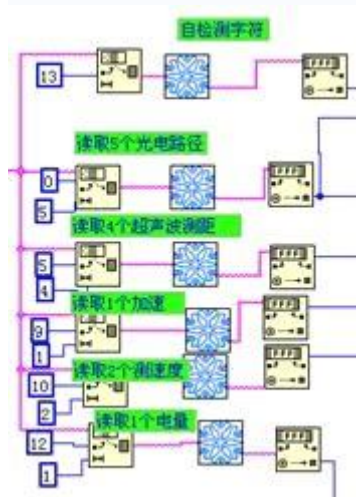


图 10-109 字符分节及转换

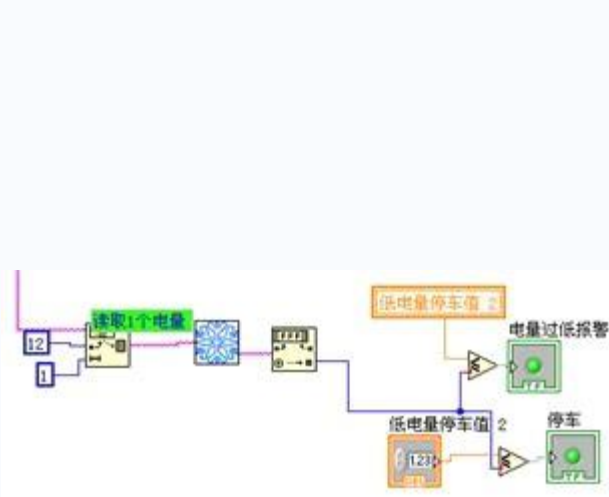


图 10-110 电量设置模块

(3) 对小车速度的控制及监视, 是一个比较重要的模块。先对速度进行比较, 判断小车是前进还是后退, 再显示速度和里程。里程通过速度与运行时间相乘而得到, 这部分的程序如图 10-111 所示。

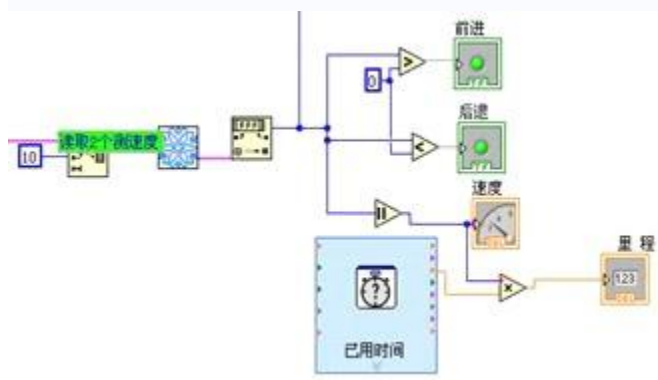


图 10-111 速度控制

(4) 设置公式节点, 这里主要是处理对小车进行控制的逻辑。它可以通过“函数”|“编程”|“结构”|“公式节点”命令打开。在公式节点中编程要符合 C 语言结构和语法。你可以理解为是在 C 界面编程, 只不过它的变量是在公式节点的边框上定义。输入与输出变量通过右键单击公式节点的边框, 选择“添加输入”或“添加输出”添加, 如图 10-112 所示。

添加好输入和输出变量后，就可以编程了，公式节点的定义及编程如图 10-113 所示。

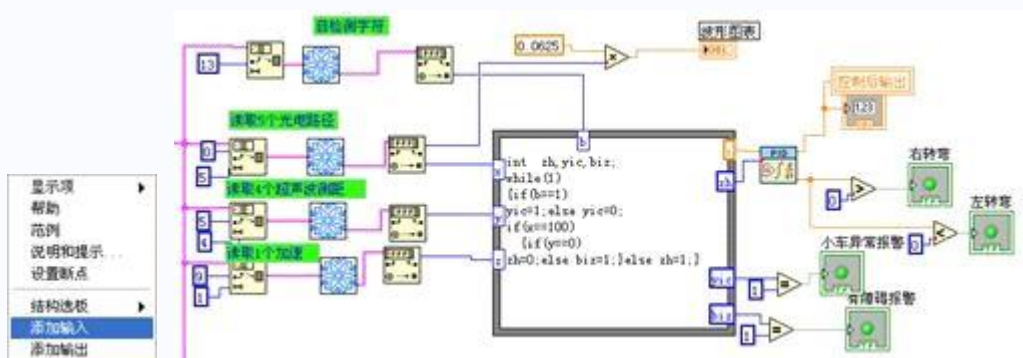


图 10-112 添加输入输出

图 10-113 公式节点的应用

从图 10-113 中可以看到，读取的字符串被转换后送到公式节点作为输入，在公式节点内处理后再输出，进行显示和 PID 控制。同时程序还通过路径传感器传来的数据判断小车偏离正常轨道的情况，并通过图表显示。

(5) 为了让系统在运行出错后能自动停止，这里加一个条件结构来与停止按钮并列工作，如图 10-114 所示。

(6) 对端口的写入操作，这里主要是用一个数组把各个命令参数组合在一起。同时这部分还用用到了一些局部变量，如图 10-115 所示。

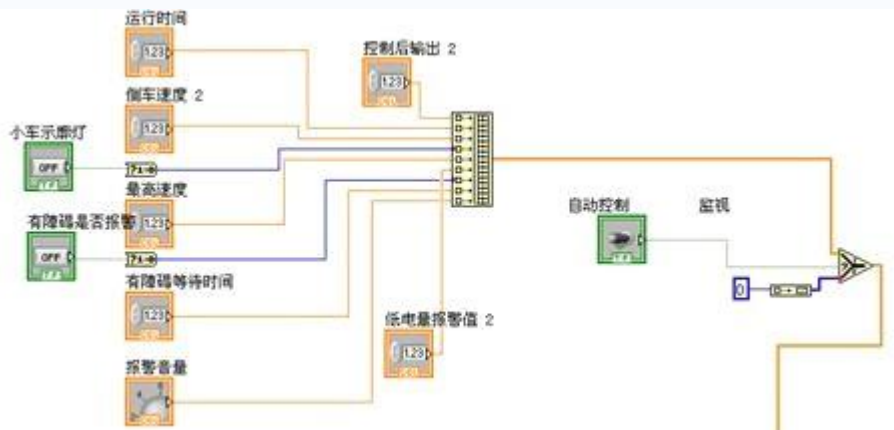


图 10-115 USB 输出量

在图中有一个布尔按钮，用来选择系统的工作状态。此按钮拨到左边时，系统工作在自动控制模式，系统会输出这些设置量；拨到右边时，系统输出全是 0，小车由下位的单片机控制，上位只作监视。

7 · 文件操作

本程序中，要对小车运行中出现地异常情况或故障情况进行记录，这样方便对系统进行维护和修理；同时也要读取记录，以便随时查看。这里采用文本文件进行记录。文本文件最大的好处是能方便地用多种方式打开，存储容量大，占用空间小。下面分别讲解文件的写和读取过程。

（1）文件写操作

写文本文件前，先要打开一个文本。因为此文本文件和程序放置在同一个文件夹中，所以先要通过“函数”|“编程”|“文件 I/O”|“文件常量”|“当前路径”命令调用“当前路径”函数；再通过“函数”|“编程”|“文件 I/O”|“拆分路径”命令调用“拆分路径”函数。

应用“当前路径”函数得到当前执行程序的路径，并在此路径上添加一个当前时间作为文件名，然后以此文件名创建一个新的文件，写入相应的数据即可。文本写入函数通过“函数”|“编程”|“文件 I/O”|“写入文件”命令调用。最后要关闭文件。这部分程序如图 10-116 所示。

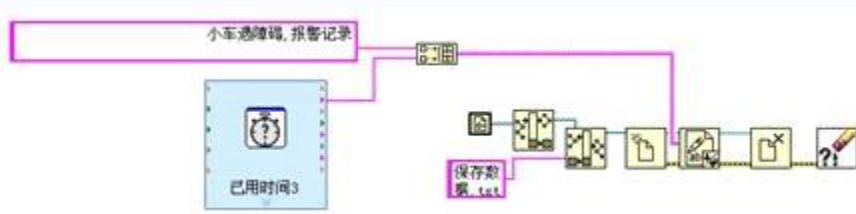


图 10-116 写入文本

（2）文件读取

在查看记录时，要读取文本文件中的值。这与前面的写操作类似，只不过写文本是把数据送到写控件；读文本是把读控件得到的数据送到文本框显示，其他的完全一样。文件读取部分的程序框图如图 10-117 所示。

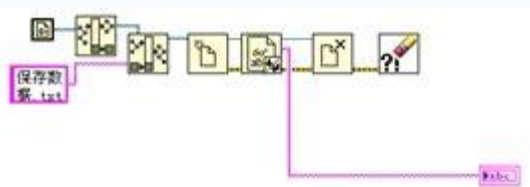


图 10-117 读文本文件

8 · 子 VI 操作

子 VI 操作就是在 LabVIEW 中调用另一个 VI，被调用的那个 VI 称为子 VI。在本程序中，调用了一个子 VI 处理字符串。

(1) 先创建一个字符串处理 VI，这个 VI 就是被调用的 VI，它实现对字符串的转换。这个子 VI 在前面板的控件较少，只有三个字符串输入和输出框，如图 10-118 所示。

(2) 进入后面板进行程序设计。先把输入的字符串通过“字符串至字节数组转换”函数转换成数组，然后通过“索引数组”函数获取数组的第 1 个值，最后通过“十六进制数字字符串至数值转换”函数和“十进制数字字符串至数值转换”函数将此值转换为十六进制字符串和十进制字符串。后面板程序如图 10-119 所示。

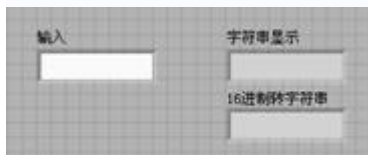


图 10-118 ASCII 码转换

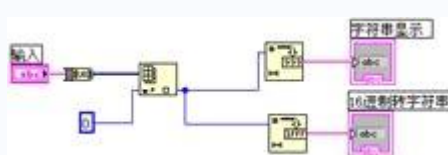


图 10-119 子 VI 的程序框图

(3) 在前面板设置子 VI 被调用时的端口。先在前面板右上角的程序图标上右键单击鼠标，此时会弹出下拉列表，如图 10-120 所示。

然后在下拉列表中选择“显示连线板”命令，图标自动变化为连线板，而且连线板上会出现不同的格子，代表可以用的几个端口。这种端口的数量可以通过右键单击增加或减少，具体方法是右键单击连线板，选择“添加接线端”或“删除接线端”命令。

设置好端口个数后就可以进行连线了。将光标放到接线板上某一小格中，光标将自动变为接线状态，单击后当前格会变黑，此时再把光标移动到某一控件上单击即可。完成后右上角接线板上对应的小格就会变为粉色，表示这些端口设置成功，以后就直接可以在这些端口上接线。

对子 VI 调用前要进行属性设置，通过“文件”|“VI 属性”|“窗口外观”|“自定义”命令来实现，如图 10-121 所示。同时在其他程序中调用子 VI 时，也要以子 VI 端口对应的属性进行。如果子 VI 中的输入是一个数值型的量，那在其他程序中调用时，在此端口输入的量也只能是数值型。



图 10-120 图标右键单击菜单

图 10-121 子 VI 的属性设置

子 VI 属性比较多，这里关键是左下角的“调用时显示前面板”这一项。选择此项后在调用子 VI 时会自动打开子 VI 并显示执行情况。若不想在调用子 VI 时显示前面板，就不必设置这一项。

9 · 整体调整

为了使整个操作界面更便于控制，最后需要添加一个选项卡控件，该控件可通过“控件”|“系统”|“系统选项卡控件”命令打开。这样设置的最大好处是单击相应的选项卡，对应的界面就会产生变化。将光标放在选项卡上面右键单击会出现下拉列表，选择其中的“在后面添加选项卡”命令，把前面编写的界面拖入刚才添加的选项卡界面。在界面中插入一个纯蓝色图片，然后从工具栏中选择排序按钮, 弹出下拉列表如图 10-122 所示。

选择“移至后面”命令，就可以把上面所布局好的选项卡控件全部放到图片之上。整个界面最后如图 10-123 所示。单击上面的选项卡可以在几个界面之间转换。



图 10-122 控件组合

图 10-123 整个界面

10 · 系统调试

完成所有的编程后，还要对其进行测试，以检验结果和编程目的是否一致，程序运行是否正常等。这项工作比较简单，限于篇幅，这里不再赘叙。

10.4.4 实例总结

本节详细讲解了 LabVIEW 中的 USB 控制技术。首先从编程的原理入手，讲解通信及控制的技术，然后结合应用实例，讲解了 USB 控制的 LabVIEW 程序实现。本例中用到的重要方法有局部变量、事件结构、对文件的读写操作、选项卡的操作等，希望读者学习时候认真领会。

LabVIEW 在数据采集与仪器控制中有着十分广泛的应用，本章将通过几个典型实例来介绍 LabVIEW 在数据采集与仪器控制领域的编程方法。

10.1 虚拟万用表

万用表是一种电子测量工具，它能对电压、电流、电阻等进行度量。它在现实中应用相当广泛，是电子测试不可缺少的工具。用 LabVIEW 中编写的万用表使用起来方便、灵活，改变了传统观念上万用表的外形，增强了其测量功能。这种方式现在越来越多地被人们应用。本节将以板卡作数据采集硬件进行编程，实现虚拟数字万用表。

10.1.1 实例内容说明

板卡在工业中应用相当广泛，尤其近年来在数据采集行业中被人们认可。它最大的好处是安装方便，操作简单，编程容易。本节中将利用 PCI8335 接口板卡作为数据采集卡，对信号进行采集并处理，开发一款虚拟 DT9205 万用表。

DT9205 型数字万用表是一个功能强大的工具。本例中将以它为实物参考，用 LabVIEW 实现一款虚拟的万用表，基本功能与 DT9205 一样，同时还增加了数据分析、在计算机上存储数据等功能。

1 · 板卡说明

PCI8335 接口板卡是一种基于 32bit PCI 总线的多功能数据采集卡，它可以应用于现场数据采集、控制、小型实验和教学等多种场合。它在硬件安装上也非常简

单，使用时只需将接口卡插入机内任何一个 PCI 总线插槽，并用螺丝固定，将信号电缆从机箱外部直接接入。它的详细功能参数如下：

- 32 bit PCI 总线，即插即用
- 输入范围：0~10V, -5~+5V
- 分辨率：12 bit
- 32路单端/16路差分模拟量输入
- 12 bit A/D 转换
- 4路 12 bit D/A 输出
- 2倍、10倍、100倍硬件增益选择
- 8路 TTL 开入（其中 4路可中断）、8路 TTL 开出
- A/D 单通道采样速率 100 KHz，N 个通道时，每一个通道的采样率为 100/N KHz
- 3路 8 bit 计数器输入，可级联为 1路 24 bit 计数器
- A/D 工作方式：软件触发、定时启动、FIFO 半满中断
- 8 KB FIFO
- 自动通道切换和单通道设置
- 通过率：单通道 100 KHz
- 超压范围：-12~+12V
- 输入阻抗：10 MΩ
- A/D 触发方式：定时
- 8 通道输入输出
- 提供 DLL 作为用户程序的接口

PCI8335 板卡的功能如图 10-1 所示。

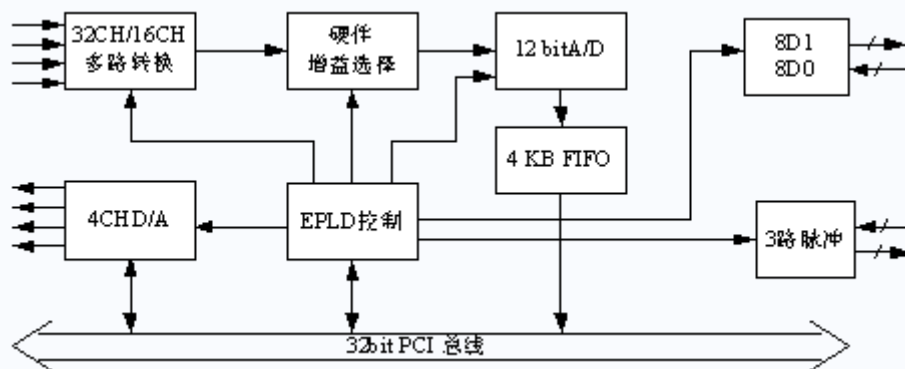


图 10-1 PCI8335 板卡功能图

2 · DT9205 万用表说明

DT9205 数字万用表具有多精度、多范围的优点。它可以直接把测量得到的数据显示出来，并按要求保存。它的具体参数如下：

— 直流电压：200mV~1000V

— 交流电压：200mV~750V

— 直流电流：2mA~10A

— 交流电流：2mA~20A

— 电阻：200Ω~200MΩ

— 电容：2nF~200μF

— 二极管测试

— 导通蜂鸣

— 数据保存

— 精度调节

— 工作指示灯

10.1.2 开发思路分析

在 LabVIEW 中开发虚拟仪器，必须先配置好硬件电路。这里先介绍 PCI8335 板卡的安装，这是整个工作的前期准备，再从 PCI 数据采集开始详细讲解说明，最后阐述信号在计算机中的分析、运算、显示等的编程。

10.1.3 板卡开发过程

板卡开发过程就是以相应板卡为硬件进行编程的过程。这里先用 PCI8335 板卡进行数据采集，然后对其调试，并在 LabVIEW 中编程，完成一个虚拟万用表。编程是本例的重点，也是本例的难点，尤其是整个程序的组合编程，涉及许多数据的融合和控件的配合。下面分步进行介绍。

1 · 板卡安装

板卡的安装是整个工作的第一步，PCI8335 在硬件安装完成后还需要进行软件安装，软件安装是非常重要的一个环节，它关系到采集板能否正常使用。具体安装步骤如下：

（1）检查硬件安装是否正确，然后通电正常启动计算机，在系统启动完成后，会自动检测新安装的硬件，如图 10-2 所示。



图 10-2 发现 PCI 硬件

（2）等到检测完毕后，弹出“找到新的硬件向导”对话框，如图 10-3 所示。

（3）单击“下一步”按钮，进入硬件连接向导界面，如图 10-4 所示。



图 10-3 硬件安装向导



图 10-4 PCI 桥连接

(4) 单击“下一步”按钮，进入硬件类型选择界面，如图 10-5 所示。

(5) 单击“下一步”按钮，进入驱动选择界面，如图 10-6 所示。



图 10-5 硬件类型选择



图 10-6 选择驱动程序

(6) 单击“从磁盘安装”按钮，将出现选择驱动程序地址的对话框，如图 10-7 所示。



图 10-7 选择驱动程序地址

(7) 单击“浏览”按钮，选择“PCI8335.inf”所在的目录（光盘：\ZTPCI8KA），单击“确定”按钮。系统将会按要求自动安装驱动。安装完成后重启计算机即可使用。你也可以继续安装第二块 PCI 卡。

安装第二块 PCI8335 卡时，系统会自动检测安装，不再经过上述步骤，只是卡的索引值不同。索引值是使用 PCI 板卡时非常重要的参数，相同的 ISA 卡是靠地址拨号区分的，而相同的 PCI 卡是靠索引值来区分的。一般情况下，第一块安装的 PCI8335 卡的索引值为 0，以后安装的 PCI8335 卡索引值自动依次加 1。调用动态链接库（ZTPCI8KA.dll）中的 OpenDevice 函数，将 PCI 卡的索引值赋给此函数，即可操作相应的 PCI 卡。

2. 板卡测试

在板卡安装好后，必须进行测试，这样才能保证以后的正常使用。测试是利用 PCI8335 数据采集卡驱动光盘自带的测试程序进行的。它一般位于光盘：\ZTPCI8KA\8335\测试程序\8335test_vb\8335test.exe 中。此程序给硬件测试带来方便，也给后面的工作提供保障。下面将对测试程序的使用方法进行讲解，测试程序的整体界面如图 10-8 所示。



图 10-8 测试界面

运行“PCI8335 数据采集卡测试程序”，从菜单栏中单击“选择板卡”命令，弹出“选择板卡”对话框，如图 10-9 所示。先输入所要操作的板卡的索引值，即可在程序的界面中对选中的 PCI8335 卡进行测试。若计算机中只安装了一块 PCI8335 卡，可略过这一步操作（本程序默认对索引值为 0 的板卡进行测试）。设置完后，就可以进行各项测试操作。此测试程序根据 PCI8335 的各种功能，分为以下几个部分：

(1) 开关量测试

该部分程序既可测试开关量输出，也可以测试开关量输入，界面如图 10-10 所示。在测试界面的左上角“开关量测试”功能块中，单击“输出测试”后面的灯，可进行开关量的输出测试。这 8 个灯，从左至右依次对应着通道 1 至通道 8，8 个开关量输出通道。单击某个灯，其状态就会在亮灭之间切换，同时输出到对应的开关量输出通道，进行相应的输出通断状态的切换。

“输入状态”后面的 8 个灯，从左至右依次为通道 1 至通道 8，分别指示了开关量输入通道的通断状态。该功能是自动的，无须手动操作。

(2) 模拟量输出测试

此部分程序用于测试模拟量输出，界面如图 10-11 所示。其中 4 个文本下拉框对应着 4 个通道模拟量输出。在某一文本框中输入的值会输出到对应的通道。这几个文本下拉框的值一般是固定的。可以通过旁边的下拉按钮选择，也可以输入任意值。

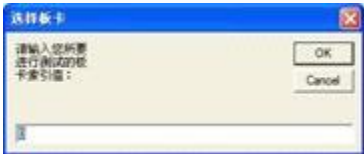


图 10-9 选择板卡



图 10-10 开关量测试



图 10-11 模拟量输出测试

(3) 模拟量输入测试

此部分程序用于模拟量输入测试，界面如图 10-12 所示，32 个文本框对应着 32 个通道的模拟量输入。单击“模拟量输入”按钮，32 个模拟输入电压值就会通过相应的文本框显示出来。

在“采集方式”单选框中，可选择模拟信号采集方式。

- 单路：用来对模拟量进行实时采集。
- 单路滤波：用来对采集到的模拟量进行滤波处理，这也是程序默认方式。
- 多路循环：用来对 32 路模拟信号通道同时采集。它的采集速度较前两者快，多用于快速变化量的测量过程测试中。

在“显示设置”单选框中，可进行输入模拟量的显示设置。

- 原码值（十六进制）：文本框中显示模拟量的十六进制原码值（ 000~FFF）。
- 0-10000mV：显示 0 到 10000mV 的单极性输入信号的电压值。

— -5000~+5000mV：显示-5000mV到+5000mV的双极性输入信号的电压值。

以上设置都可在单击“模拟量输入”按钮后进行动态地进行。这样方便在系统测试过程中对参数进行设置。

(4) 计数器测试

计数器测试主要是测试脉冲输入，对脉冲计数，如图 10-13 所示。它可进行 PCI 8335 的三个计数器的测试，分为三组，分别对三个计数器进行测试。在每组中的第一个文本框显示计数器寄存器中的当前值，第二个文本框显示计数累加值。单击“累加清零”按钮，将清空第二列文本框中的值，重新进行累加计数。测试程序运行后，软件就自动读取计数器寄存器中的当前值，并进行累加，无须手动操作。



图 10-12 模拟量输入测试



图 10-13 计数器测试

(5) 中断测试

中断测试就是对采集卡的中断进行一定的测试。它能对 PCI8335 采集卡上的三种中断进行测试。单击“打开中断总允许”按钮，可打开 PCI8335 中断功能，其界面如图 10-14 所示。

- “FIFO 半满中断”区域：可对 FIFO 半满产生的中断进行测试。单击“FIFO 中断测试”按钮，打开 FIFO 中断允许，并启动 A/D 转换。当中断到来时，“中断次数”文本框显示由于 FIFO 半满产生的中断次数，“中断个数”文本框显示本次中断把 FIFO 中的数据读空，所读取的数据个数。
- “计数器中断”区域：对计数器产生的中断进行测试。单击“计数器中断测试”按钮，即打开计数器中断允许。“中断次数”文本框中显示计数中断的累计次数。
- “开关量中断”区域：可对开关量中断进行测试。单击“开关量中断测试”按钮，打开开关量中断允许。“中断次数”文本框中显示开关量中断的累计次数。

单击“关闭所有中断”按钮，将关闭所有中断，禁止中断进入。

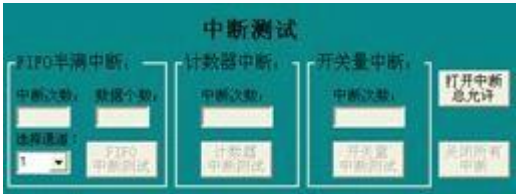


图 10-14 中断测试

3 · 参数设置

在 PCI8335 数据采集板卡中，为了简化操作。动态链接库（PCI8KA.dll）中几乎所有的函数的参数均通过一个结构体（ZT_PCIBOARD）

来传递。该结构体及 PCI8000 的有关函数说明如下：

```
struct ZT_PCIBOARD
{
    long lIndex;    /* PCI 卡索引值，该值在安装时被系统分配*/
    HANDLE hHandle; /* PCI 卡的操作句柄（只有涉及中断时才用）*/
    short nCh;     /* 通道号*/
    long lData;    /*输入输出数据变量*/
    long* plData;  /*输入输出数据指针*/
    long lCode;    /*设备控制字，具体含义见函数说明*/
};
```

下面对各函数功能进行说明。

(1) 打开设备

函数	long APIENTRY OpenDeviceAmcc(ZT_PCIBOARD* bs)
功能	打开某一块PCI8335 卡
入口有效参数	lIndex: PCI 卡索引值
出口返回参数	hHandle: PCI 卡的操作句柄（只有涉及中断时，返回才有意义）
返回值	0：打开设备成功 1：打开设备失败

(2) 关闭设备

函数	long APIENTRY CloseDeviceAmcc(ZT_PCIBOARD* bs)
功能	关闭某一块PCI8335 卡
入口有效参数	lIndex: PCI 卡索引值
出口返回参数	无
返回值	0：关闭设备成功 1：关闭设备失败

(3) 开关量输入

函数	long APIENTRY ZT8335DI (ZT_PCIBOARD* bs)
功能	采集PCI8335 某一通道开关量输入信号的状态
入口有效参数	lIndex: PCI 卡索引值

	ICode: 设备控制码, 0 表示单通道操作; 1 表示多通道操作 nCh: 通道号, 1~8 (单通道模式使用)
出口返回参数	IData: 某一通道读取值, 0 或 1 (单通道操作) 该卡所有开关量输入状态值 (多通道操作)
返回值	0: 采集成功 1: 采集失败

(4) 开关量输出

函数	long APIENTRY ZT8335DO (ZT_PCIBOARD* bs)
功能	进行某一个通道的开关量数据输出操作
入口有效参数	lIndex: PCI 卡索引值 ICode: 设备控制码, 0 表示单通道操作; 1 表示多通道操作 nCh: 通道号, 1~8 (单通道模式使用) IData: 某一通道输出设定值, 0 或 1 (单通道操作); 该卡所有开关量输入设定值 (多通道操作)
出口返回参数	无
返回值	0: 开关量数据输出成功 1: 开关量数据输出失败

(5) 模拟量输出

函数	long APIENTRY ZT8335AO(ZT_PCIBOARD* bs)
功能	进行某一个通道的模拟量数据输出操作
入口有效参数	lIndex: PCI 卡索引值 nCh: 通道号, 1~4 (单通道模式使用) IData: 某一通道输出原码值 (0 ~ 4095)
出口返回参数	无
返回值	0: 模拟量数据输出成功 1: 模拟量数据输出失败

(6) 初始化模拟量采集

函数	long APIENTRY ZT8335AIInt(ZT_PCIBOARD* bs)
功能	初始化模拟量数据采集
入口有效参数	lIndex: PCI 卡索引值 nCh: 通道号 单路采集通道号: 1~32 多路循环采集停止通道号: 1~32 ICode: 设备控制字。 停止: 0 启动: 1 多路循环: 2 (默认单路采集) 单端: 4 (默认差分方式) 单极性 (0~10V): 256 (默认原码值) 双极性 (-5V ~ +5V): 512 (默认原码值)
出口返回参数	无

返回值	0: 初始化成功
	1: 初始化失败

(7) 单通道采集

函数	long APIENTRY ZT8335AI (ZT_PCIBOARD* bs)
功能	采集某一通道模拟量数据
入口有效参数	lIndex: PCI 卡索引值
	nCh: 通道号, 1~32
出口返回参数	lData: 某一通道模拟量的数值
返回值	0: 采集成功
	1: 采集失败

注：该函数无须调用 ZT8335AIInit 初始化。

(8) 单通道采集（滤波）

函数	long APIENTRY ZT8335AIA (ZT_PCIBOARD* bs)
功能	采集一组某一通道模拟量数据数据，并进行滤波处理
入口有效参数	lIndex: PCI 卡索引值
	nCh: 通道号, 1~32
出口返回参数	lData: 某一通道模拟量的数值
返回值	0: 采集成功
	1: 采集失败

注：该函数无须调用 ZT8335AIInit 初始化。

(9) 设置 FIFO 数据地址

函数	long APIENTRY GetAddress(short nIndex,long* lAdd)
功能	获取用户存放FIFO 数据的地址
参数	nIndex: PCI 卡索引值
	lAdd: 存放FIFO 数据的地址（建议用户开 8192 个 32 位的长整型数据缓冲区，接收 FIFO 中的数据）
返回值	0: 设置成功
	1: 设置失败

(10) FIFO 数据批量采集

函数	long APIENTRY ZT8335AIFifo (ZT_PCIBOARD* bs)
功能	采集某一通道或多路循环通道的一组模拟量数据
入口有效参数	lIndex: PCI 卡索引值
	plData: 存放 FIFO 数据的指针
出口返回参数	lData: 从FIFO 中读取的数据个数
返回值	0: 采集成功
	1: 采集失败

注：1. 使用前，需调用 ZT8335AIInit 初始化。
2. 初始化后，用户需调用 GetAddress 设置存放数据的数组地址（该函数是为不支持指针的计算机语言设置的，比如：Visual Basic）。

3. 使用支持指针的计算机语言，可不用 GetAddress 函数，直接将存放 FIFO 数据的地址赋给 pData 即可。

(11) 脉冲输入

函数	long APIENTRY ZT8335CT (ZT_PCIBOARD* bs)
功能	读取某个计数器的寄存器值
入口有效参数	lIndex: PCI 卡索引值 nCh: 通道号 (计数器编号) , 1~3
出口返回参数	lData: 某一计数器的寄存器值
返回值	0: 读取成功 1: 读取失败

(12) 打开中断

函数	long APIENTRY OpenIRQ(HANDLE* hList,ULONG hDeviceNum,void CALLBACK IrqCallBack (ZT_PCIBOARD* bs))
功能	开一出PCI 卡的中断总允许，等待中断到来
入口有效参数	hList: 开中断允许列表 hDeviceNum: 开中断允许数量 IrqCallBack: 这是一个回调函数的入口指针
出口返回参数	无
返回值	0: 打开中断成功 1: 打开中断失败 2: 没有要打开的中断
回调函数	void CALLBACK IrqCallBack(ZT_PCIBOARD* bs)
功能	当 OpenIRQ 调用后，PCI8335 产生中断时，系统会调用此函数，用户自己编写函数IrqCallBack 进行中断处理，系统会提供以下参数，供用户使用：nIndex (引起中断的索引值) , 用户可调用板卡操作函数，对板卡进行相应的操作。

(13) 关闭中断

函数	long APIENTRY CloseIRQ()
功能	禁止本机上所有已开PCI8335 卡的中断总允许
返回值	0: 禁止中断成功 1: 禁止中断失败

(14) 设置中断允许寄存器

函数	long APIENTRY SetIE(ZT_PCIBOARD* bs)
功能	设置中断允许寄存器值
入口有效参数	lIndex: PCI 卡索引值 lCode: 设备控制字 (中断类型: FIFO 半满中断、计数器半满中断、开关量中断)
返回值	0: 设置成功 1: 设置失败

(15) 读中断状态寄存器

函数	long APIENTRY ReadIS (ZT_PCIBOARD* bs)
----	--

功能	读取中断状态寄存器值
入口有效参数	lIndex: PCI 卡索引值
出口返回参数	lCode: 设备控制字（产生中断的类型：FIFO 丰满中断、计数器丰满中断、开变量中断）
返回值	0：读取成功 1：读取失败

（16）DLL 编译

在 LabVIEW 中利用 PCI8335 数据采集卡编程时，直接利用 DLL 文件，对其进行数据采集处理，所以先要进行 DLL 程序的编译。在 VC++中编写程序，并生成 DLL 文件即可。C 源程序如下：

```
#include "stdafx.h"

#include <windows.h>
#include <stdio.h>
#include "PCI8KA.h"
ZT_PCIBOARD* bs;
ZT_PCIBOARD ztbs;
void main()
{
    long i;
    ULONG sData[4096*16];
    ULONG mOutput=0;
    bs=&ztbs;
    //打开一个 PCI8335 卡
    ztbs.lIndex=0;
    OpenDeviceAmcc(bs);
    ztbs.lCode=7;    //多路循环(4+2+1)
    ztbs.nCh=4;    //终止通道为 4，即 4 个通道循环采集！
    ZT8335AIIInit(bs);
    /*以下用两种方法实现 FIFO 的读取!*/

    //*****函数 ReadFifo 示例*****
    ReadFifo((short)bs->lIndex, sData, &mOutput);

    for (i=0;i<25;i++)
    {
        printf("%x\n",sData[i]);
    }

    printf("%u\n",mOutput/4); // mOutput/4 为 FIFO 被读空所返回的数据*
}
```

10.1.4 设计万用表界面

用 LabVIEW 编写虚拟万用表的界面是本例重点要讲解的内容，下面具体讲解此部分内容。

1 · 采集设置

(1) 采集设置界面主要用来对采集卡参数进行调节。这里先要在前面板放置几个数值输入控件，并依次按要求为其命名，如图 10-15 所示。

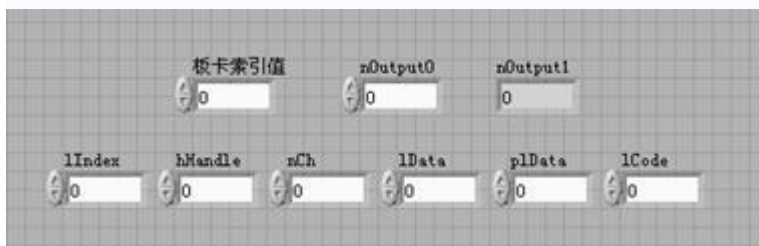


图 10-15 采集设置界面

(2) 创建一个簇控件，并重命名为 BS，并把图 10-15 中下面一排控件放入其中，如图 10-16 所示。

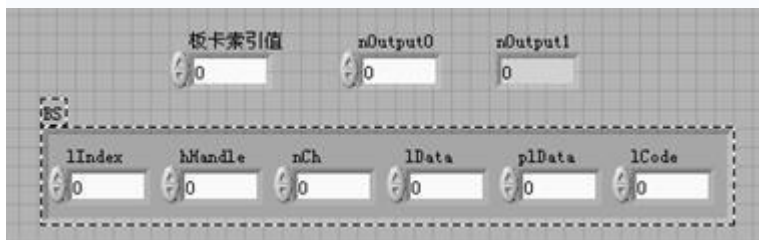


图 10-16 “簇”的创建

簇控件可以通过“控件”|“新式”|“数组、矩阵、簇”|“簇”命令调用。设置簇时，只要把想设置为簇的元素全选中，拖动到簇框中就可以了。

(3) 在后面板中放置一个“层叠式顺序结构”图标，可通过“函数”|“编程”|“结构”|“层叠式顺序结构”命令调用，如图 10-17 所示。

(4) 在“层叠式顺序结构”的边框上右键单击，选择“在后面添加帧”就会在结构体上出现两个帧。在帧下拉列表上单击，可两个帧之间转换。在第 0 帧中，要把簇和索引号捆绑到一起，这样才便于后面的采集过程处理。捆绑函数可用“函数”|“编程”|“簇、变体、类”|“按名称捆绑”命令来调用。



图 10-17 层叠式顺序结构

将捆绑后的数据通过调用库函数来处理。调用库函数可通过“函数”|“互连接口”|“库与可执行文件”命令来调用。处理完后要对捆绑的数据解除捆绑，以方便下次处理。解除捆绑函数通过“函数”|“编程”|“簇、变体、类”|“按名称解除捆绑”命令来调用。具体程序如图 10-18 所示。

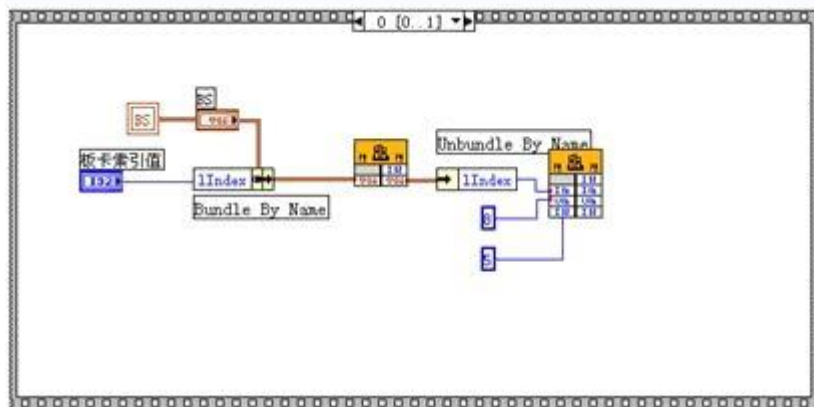


图 10-18 第 0 帧程序框图

(5) 对第 1 帧进行编程时，只需要调用库函数操作即可。这里实现的操作是读取数据采集卡的数据，并将其数据输出，输出端口的函数名为 Noutput，把此端口连接到顺序结构的外面用于数据输出，如图 10-19 所示。

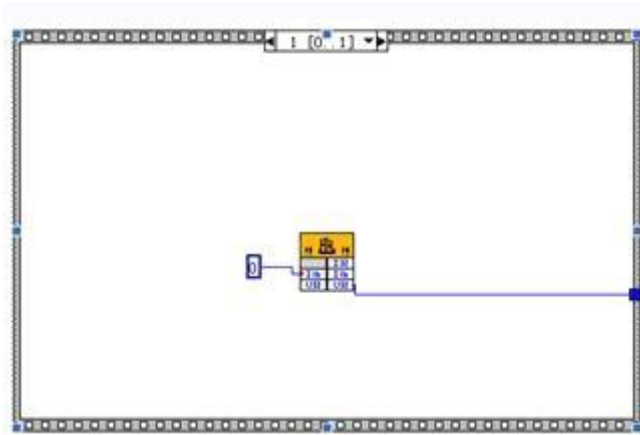


图 10-19 第 1 帧程序框图

2 · 虚拟万用表旋钮

虚拟万用表的旋钮是按照现实中的 DT9205 模式进行设计的，下面将分步进行介绍。

(1) 通过“控件”|“新式”|“数值”|“转盘”命令，在前面板放置一个转盘控件。对其选中后，在边角处拖动，放大控件，并在右键菜单中选择“文本标签”命令，再进行设置属性，其界面如图 10-20 所示。把数据类型设置为“无符号长整型”。在“文本标签”选项卡双击“文本标签”栏的选项，写入旋钮对应的名称，再单击“插入”按钮，重复多次，写入每一个项的名称。注意这些名称的先后顺序不能乱。

旋钮标签设置完以后前面板界面如图 10-21 所示。



图 10-20 旋钮标签设置

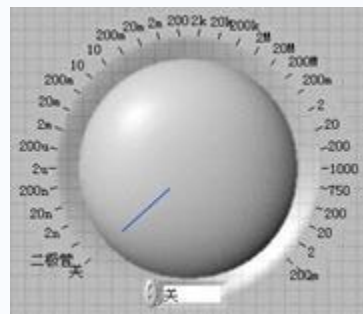


图 10-21 旋钮界面

(2) 在前面板放一个字符串，用于显示测量结果。如果结果只是一个数值的话，可以用数值控件显示。这里为了在更加直观、完美地显示数值的同时，还显示测量种类和单位等一些信息，因此选择了字符串控件。前面板界面如图 10-22 所示。

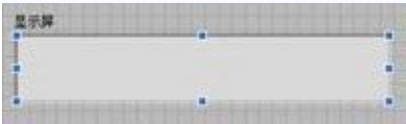


图 10-22 结果显示屏

(3) 放置一个数值输入控件，用于控制测量精度。同时放置一个工作指示灯，用于指示仪器工作状态。这些操作比较简单，这里不再详细说明。

3 · 程序框图结构

在后面板中，先要放置一个条件结构，用于指示万用表对旋钮不同刻度执行不同动作。条件结构通过“函数” | “编程” | “结构” | “条件结构”命令调用。在条件结构上，右键单击下拉框，选择“在后面添加分支”命令，添加条件分支，如图 10-23 所示。

(1) 在条件结构中添加 32 个分支，这样每一个分支都对应一个条件结构，执行不同的动作，并把旋钮和条件结构的“分支选择器”连接起来。0~31 分支与旋钮的文本标签值是对应的，如图 10-24 所示。

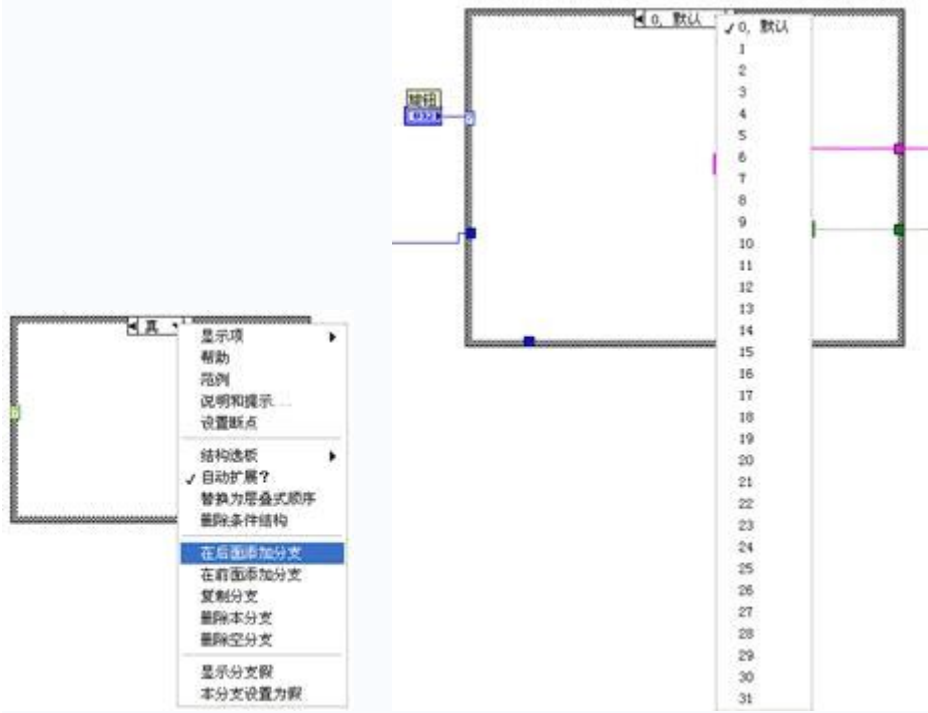


图 10-23 添加分支

图 10-24 条件结构设置

(2) 对每一个分支进行设置。因为不同的分支执行不同动作，所以每一个分支的程序要根据情况来写。

分支 0 负责显示系统的开闭状态，默认为关机，指示灯灭，显示屏显示“关”。这里通过“函数”|“编程”|“字符串”|“字符串常量”命令写入一个字符串常量“关”，而程序中指示灯的状态是由一个布尔控件控制的。此时要灯灭，就通过“函数”|“编程”|“布尔”|“假常量”命令放置一个假常量布尔控件。分支 1 负责检测二极管。当二极管红表笔接正极时，系统中电路接通，显示“此时红表笔为接二极管正极”。接反时会显示“此时红表笔接为二极管负极”。当表笔没有接到二极管上时，显示“没有检测到任何二极管”。程序框图如图 10-25 所示。

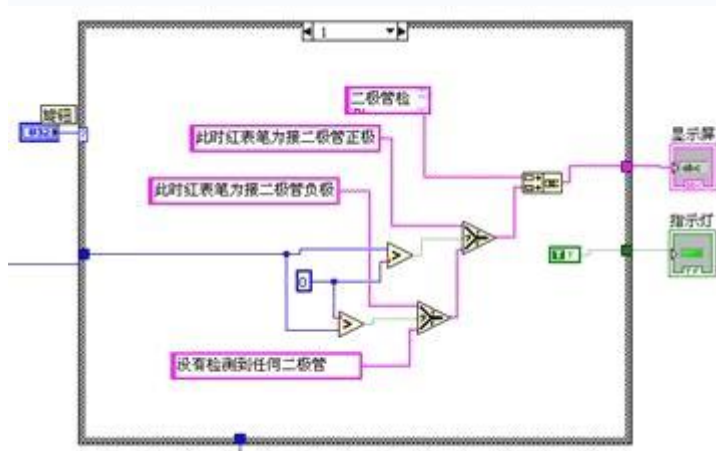


图 10-25 二极管检测程序

分支 2 到 6 负责电容的检测，只是量程不同。测量中先对数据的大小进行检测，判断测量值是否超过量程。若没有超过量程，就读取测量精度，对测量值进行相应精度的转换。然后将测量值转换为字符串，通过一个字符串连接控件，并添加上单位，再送到文本框显示。若超过量程，则在文本框中显示“超出测量范围，请选择高档位”，提示用户换用高档位。程序框图如图 10-26 所示。

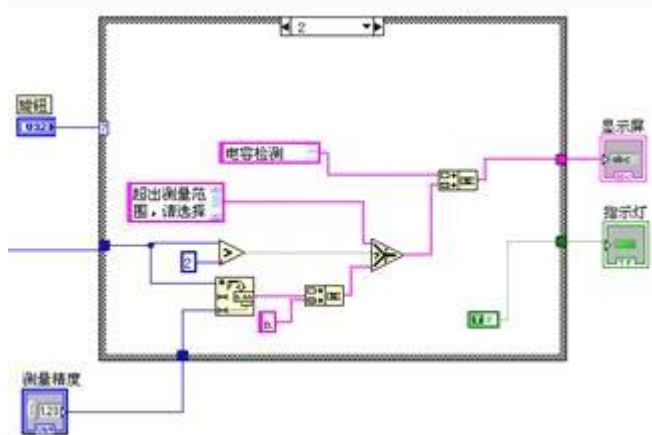


图 10-26 电容测量程序

分支 7 到 10 负责交流电流测量，其中分支 11 到 14 负责测量直流电流。它们的测量原理相同，只是大小量程不同。还是先判断测量值的大小，再进行相应精度的字符串转换显示。整个程序框图如图 10-27 所示。

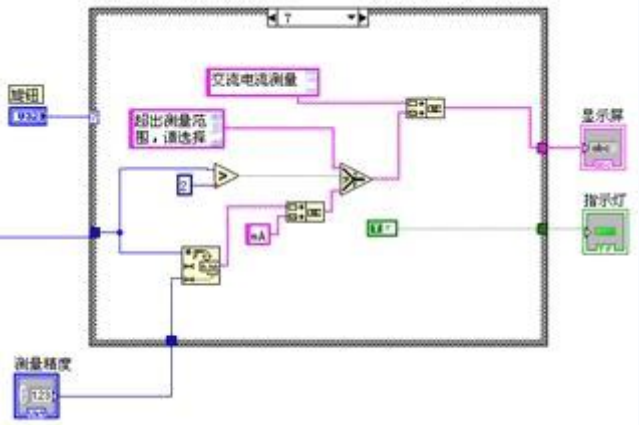


图 10-27 交流电流测量程序

分支 15 到 21 负责电阻测量。这一个测量范围相当广泛，也是一个数值量的比较。在测量范围内万用表才会测量，如果超过范围，则显示“超出测量范围，请选择高档位”。程序框图如图 10-28 所示。

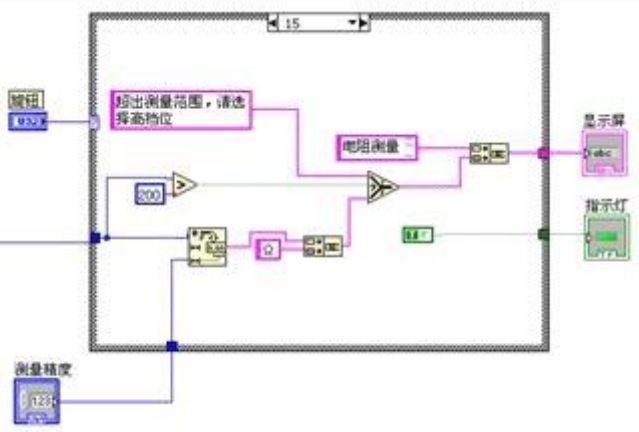


图 10-28 电阻测量程序

分支 22 到 31 负责电压测量。测量原理和上面的相同，只是单位和大小不同。整个程序框图如图 10-29 所示。

4 · 设置时间和信息

下面讲解如何显示时间和万用表名称。先在前面板放置一个文本显示框，用来显示时间和信息，再到后面板中放置一个时间发生函数（通过“函数” | “编程” |

“定时” | “已用时间” 命令来调用），然后用一个字符串连接控件把要显示的信息和时间字符连接到一起显示，程序框图如图 10-30 所示。

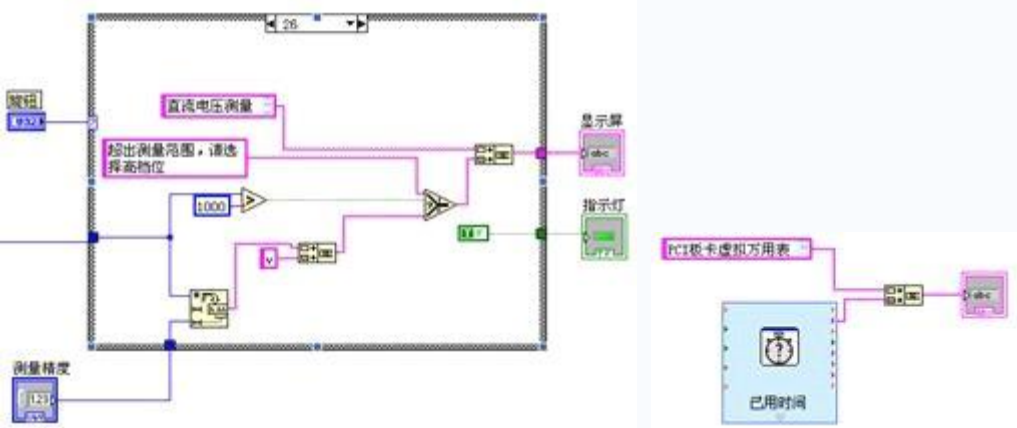


图 10-29 电压测量程序

图 10-30 显示时间

5 · 整体程序

在程序编写完成后，还必须添加一个循环让万用表能连续工作。这里用 While 循环控制程序的运行。整个程序框图如图 10-31 所示。

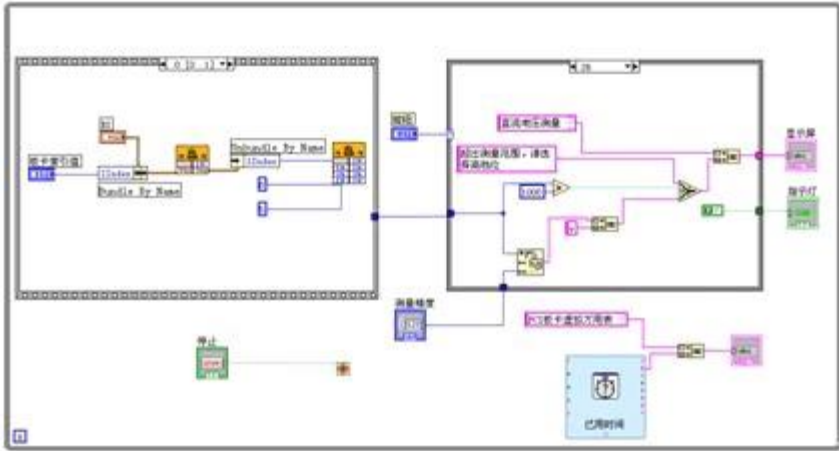


图 10-31 完整程序

6 · 前面板装饰

当程序写完后，就要对其进行界面布局，让之看上去更整齐美观。所谓的装饰前面板即给一些控件进行着色布局。这里先对旋钮进行着色，打开“查看” | “工具选板”面板，单击最下面的颜色设置控件，就可以对旋钮进行着色。为了区别各种测量对象刻度，将它们着色为不同的颜色，如图 10-32 所示。接着对其整个界面进行布局安排，最后的前面板如图 10-33 所示。

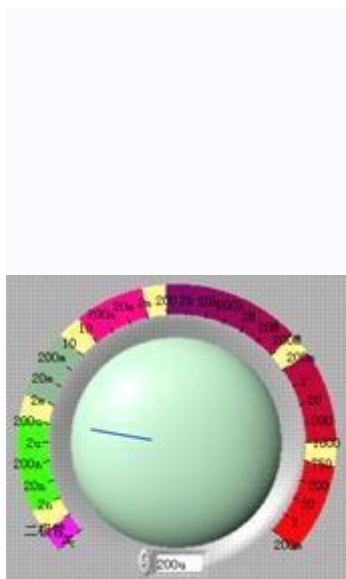


图 10-32 旋钮装饰



图 10-33 完整前面板界面

10.1.5 实例总结

10.1 节利用 PCI 板卡和 LabVIEW 设计了虚拟万用表，通过计算机和硬件的配合实现了对各种电路相关物理量的测量。PCI 板卡是现在应用比较广泛的一种采集卡。学习完本节后，相信读者将对板卡的应用以及虚拟仪器编程有了一定的了解，并掌握了其具体的编程的方法和技巧。

11.2 鱼体复阻抗测量系统设计

现代智能化仪器仪表的设计，在很大程度上是软件设计。对于同一个硬件电路，配以不同的软件，实现的功能也就不同，而且有些硬件电路的功能可用软件来实现。虚拟仪器的主要特点就是体现了“软件就是仪器”这一思想，它以特定的软件取代相应的硬件电路，充分利用计算机软硬件资源，用计算机完成传统仪器硬件的部分以至于全部功能，它是传统仪器功能和外形的模块化和软件化。利用虚拟仪器可以快速组建测控系统，将原来示波器、信号发生器、电压表等人们所熟悉的常规仪器，通过模块化的功能软件，集成到一个系统中，完成特定的分析任务。

利用虚拟仪器采用“软件就是仪器”的思想，只需要开发不同功能的程序代替硬件，就可以节约大量的硬件成本。本节具体介绍鱼体复阻抗测量系统的设计过程。

11.2.1 实例内容说明

1 · 系统组成结构

典型的虚拟仪器测试系统的工作流程为：传感器检测到被测信号，将其转化为电量信号；信号处理电路将电量信号进行放大、滤波、检波处理后，变成标准信号；数据采集卡采集到信号处理电路的电量信号，转换为计算机所能处理的数字信号；通过驱动程序将数字信号输入计算机；基于虚拟仪器平台，调用数字信号处理程序，得到测试结果。

本节采用虚拟仪器 LabVIEW 来组建鱼体复阻抗测量系统，系统的示意图如图 11-19 所示。该测量系统主要包括基于虚拟仪器的信号发生器、信号采集模块、复阻抗信息提取模块和测量数据保存模块。

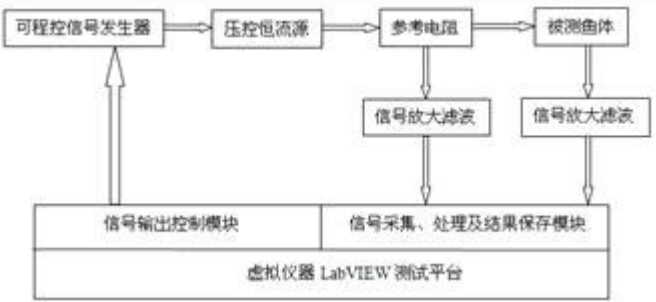


图 11-19 鱼体复阻抗测量系统示意图

2 · 系统功能

该系统能快速实现鱼体复阻抗信号的采集和复阻抗特征参数的提取，并对相位角提取算法进行研究分析和自动保存文件。该测量系统能在 2s 内快速检测和分析单频率的鱼体复阻抗信息，在 100Hz~100 kHz 内采用每倍频 10 点扫频激励，并能在 30~50s 内实现鱼体复阻抗谱的测量和记录。

11.2.2 软件设计流程

本例通过 NI 公司的 LabVIEW 软件开发阻抗测量系统，软件设计流程如图 11-20 所示，测量系统界面如图 11-21 所示，

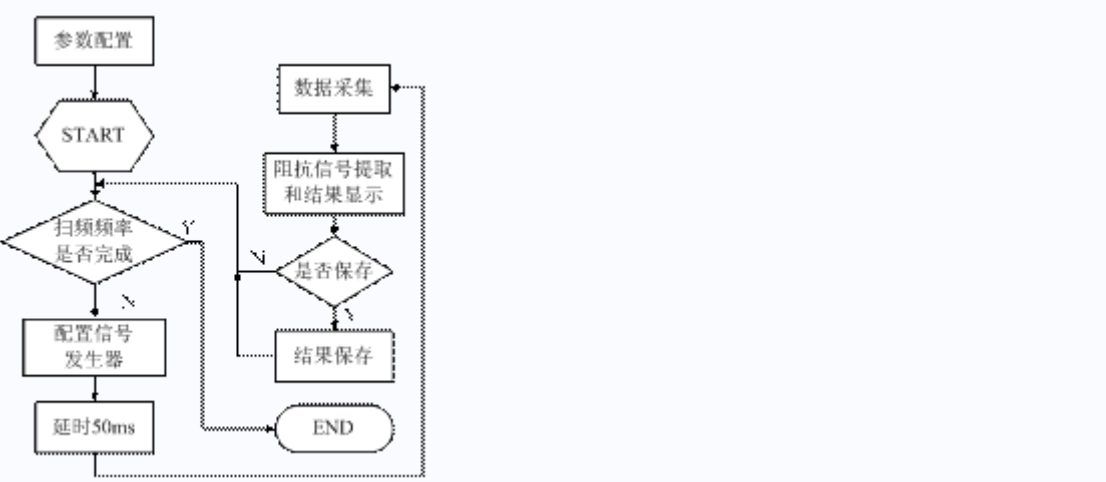


图 11-20 测量系统流程图

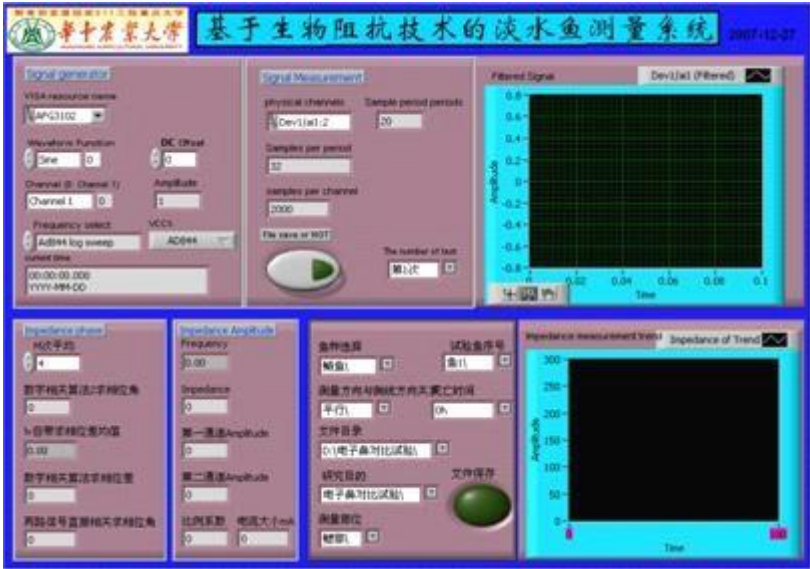


图 11-21 鱼体复阻抗测量系统界面

11.2.3 基于虚拟仪器的信号发生器

生物阻抗具有频率特性，Schwan 在研究生物组织频率特性时发现生物组织内存在三个不同的频率散射，分别称为 α 、 β 和 γ 散射，其对应的频段为音频频段（几 Hz~几十 kHz）、射频频段（几十 kHz~几十 MHz）及微波频段（几十 MHz 以上），其中在音频频段和射频频段的生物组织介电特性可以同时反映出细胞内、外液体的特征，可以将这两个频段作为生物阻抗测量的激励频率范围来分析生物体特性。为了研究鱼体复阻抗的变化规律，需要在一个较宽泛的频率范围内进行研究，并且信号参数必须可调，为此开发了基于虚拟仪器的程控信号发生器，可实现自动扫频测量，以研究鱼体复阻抗特性变化规律。

1 · 程控信号发生器的选择和设计

为了保证信号的波形失真小，幅值稳定，可以方便调节信号的频率和幅值，并能实现快速在不同频率、幅值下研究鱼体复阻抗信息，本例采用了泰克公司的 AFG 3022 信号发生器，该信号发生器标准接口为 USB 1.1 GPIB、LAN 10Base-T/100 Base-TX、USB 1.1。本实例通过 USB 接口或网线与计算机相连，采用 VISA 可视化仪器软件结构（该软件结构可到 NI 和 Tektronix 公司网站下载），在 LabVIEW 平台下编写自定义的扫频程序。基于虚拟仪器的程控信号发生器前后面板如图 11-22 和图 11-23 所示，将它作为一个子 VI，可集成在鱼体复阻抗测量系统中，实现输出信号可调的信号发生器。



图 11-22 程控信号发生器的前面板

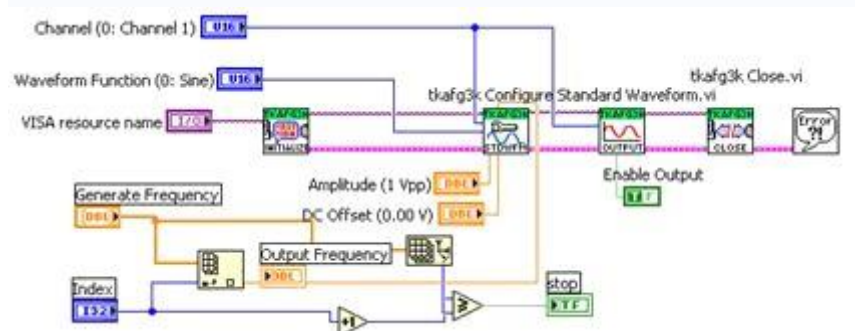


图 11-23 基于 AFG 3022 的程控信号发生器后面板

2 · 扫频频率的设计

根据 Cole-Cole 理论模型，对生物组织的阻抗特性的研究需要在 $0 \sim \infty \text{ Hz}$ 的频率范围内进行，但在实际测量中，由于鱼体组织会随死亡时间的延长而发生一定变化，如僵硬、解僵、腐败等，这要求对鱼体复阻抗的一次扫频测量必须在足够短的时间内完成，因此对鱼体复阻抗特性的研究也只能在有限个频率点上进行。由于该研究项目的目的是利用鱼体的复阻抗特性的变化规律来评价鱼体新鲜度，只需要知道鱼体复阻抗与鱼体腐败评价的化学指标的关系，为此采用了对数等距扫

频法和准线性扫频法来研究鱼体复阻抗特性，并将这两种方法作为子 VI 来控制泰克信号发生器产生指定的信号，实现自动扫频。这两种扫频方法具体实现过程如下：

(1) 准线性扫频法 (Quasi-linear Frequency Method)，是在每十倍频程内等几何间距的原则设置 N (试验时取 $N=10$) 个频率，如在 $10\sim 100\text{Hz}$ 的十倍频程内，扫频频率设为 10, 19, 28, 37, 46, 55, 64, 73, 82, 91Hz；在其他十倍频程内，如 $10\text{kHz}\sim 100\text{kHz}$ ，扫频频率为上述频率值乘以 10^i ($i=1\sim 3$)。基于准线性扫频法的虚拟仪器子 VI 如图 11-24 所示。

(2) 对数等间距扫频法 (Logarithmic Frequency Method)，是在对数频率轴上频率点等几何间距分布，如在 $10\sim 100\text{Hz}$ 的十倍频程内，扫频频率设为 10, 12.6, 16.0, 20.3, 25.7, 32.6, 41.2, 52.2, 66.1, 83.8Hz；在其他十倍频程内，如 $10\text{kHz}\sim 100\text{kHz}$ ，扫频频率为上述频率值乘以 10^i ($i=1\sim 3$)，基于对数准线性扫频法的虚拟仪器子 VI 如图 11-25 所示。

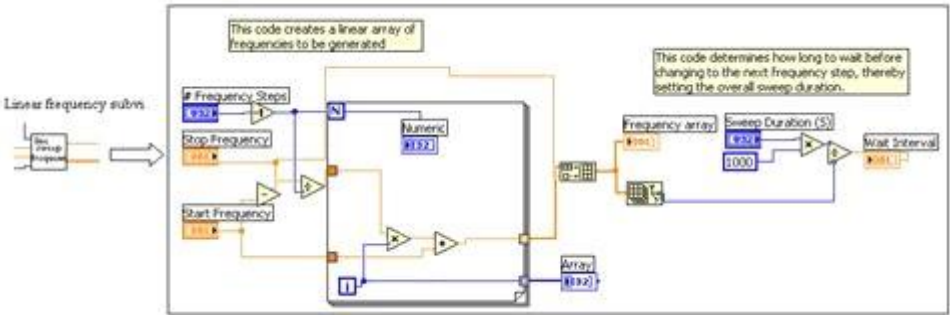


图 11-24 基于准线性扫频法的频率输出程序

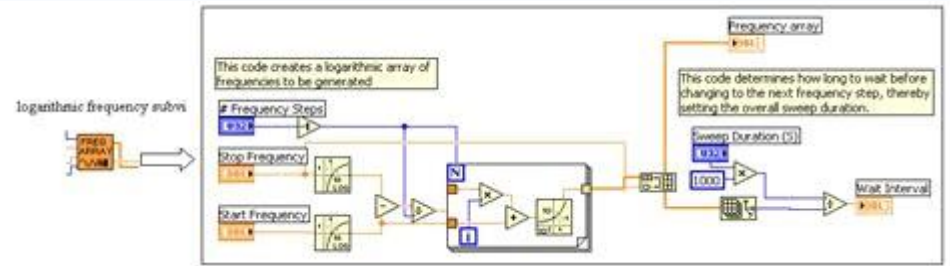


图 11-25 基于对数扫频法的频率输出程序

11.2.4 数据采集模块

图 11-26 所示为数据采集模块的结构。在采集数据之前，程序要对采集板卡初始化，板卡上和内存中的 Buffer 是数据采集存储的中间环节。需要注意的两个问题是：是否使用 Buffer；是否使用外触发启动、停止或同步一个操作。

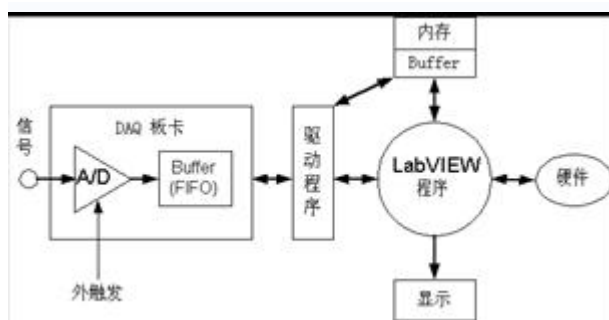


图 11-26 数据采集系统结构

这里的 Buffer（缓冲）指的是计算机内存的一个区域（不是数据采集卡上的 FIFO 缓冲），它用来临时存放数据。例如，你需要每秒采集几千个数据，而在 1s 内显示或图形化所有数据是困难的。但是如果将采集卡的数据送到 Buffer，你就可以先将它们快速存储起来，稍后再重新读出它们而进行显示或分析。需要注意的是 Buffer 与采集操作的速度及容量有关。如果你的卡有 DMA 功能，模拟输入操作就有一个通向计算机内存的高速硬件通道，这就意味着所采集的数据可以直接送到计算机的内存。

不使用 Buffer 意味着对所采集的每一个数据都必须及时处理（图形化、分析等），因为没有地方可以保存那些你着手处理的数据之前的若干数据点。

下列情况需要使用 Buffer I/O:

- 需要采集或产生许多样本，其速率超过了实际显示、存储到硬件，或实时分析的速度。
- 需要连续采集或产生 AC 数据（>10 样本 / 秒），并且要同时分析或显示某些数据。
- 采样周期必须准确、均匀地通过数据样本。

下列情况可以不使用 Buffer I/O:

- 数据组短小，例如每秒只从两个通道之一采集一个数据点。
- 需要缩减存储器的开支。

触发（Triggering）涉及初始化、终止或同步采集事件的任何方法。触发器通常是一个数字或模拟信号，其状态可确定动作的发生。软件触发最容易，你可以直接用软件，例如使用布尔面板去启动/停止数据采集。硬件触发则是由板卡上的电路管理触发器，控制采集事件的时间分配，有很高的精确度。硬件触发可进一步分为外部触发和内部触发。当某一输入通道发生一个指定的电压电平时，让板卡输出一个数字脉冲，这是内部触发。采集卡等待一个外部仪器发出的数字脉冲

到来后初始化采集卡，这是外部触发。许多仪器提供数字输出（常称为“trigger out”）用于触发特定的装置或仪器，在这里，要触发的装置就是数据采集卡。

在以下情况下使用软件触发：

— 用户需要对所有采集操作有明确的控制，并且事件定时不需要非常准确。

下列情况使用硬件触发：

— 采集事件定时需要非常准确。

— 用户需要削减软件开支。

— 采集事件需要与外部装置同步。

1 · 差分测量系统 (DEF)

鱼体复阻抗特征参数（阻抗幅值和相位）的准确提取，需要同步采集参考电阻与鱼体电压电极上的信号，这样才能准确测量鱼体复阻抗特征参数。为了抑制共模信号，采用差动输入接法，如图 11-27 所示。

一个理想的差分测量系统仅能测出（+）和（-）输入端口之间的电位差，完全不会测量到共模电压。然而，实际应用的板卡却限制了差分测量系统抵抗共模电压的能力，数据采集卡的共模电压的范围限制了相对于测量系统地的输入电压的波动范围。共模电压的范围关系到一个数据采集卡的性能，因此必须采用一些方法来消除共模电压的影响。如果系统共模电压超过允许范围，需要限制信号地与数据采集卡的地之间的浮地电压，以避免测量数据错误。

2 · 参考地单端测量系统 (RSE)

一个 RSE 测量系统，也叫做接地测量系统，被测信号的一端接模拟输入通道，另一端接系统地 AIGND。图 11-28 描绘了一个 16 通道的 RSE 测量系统。

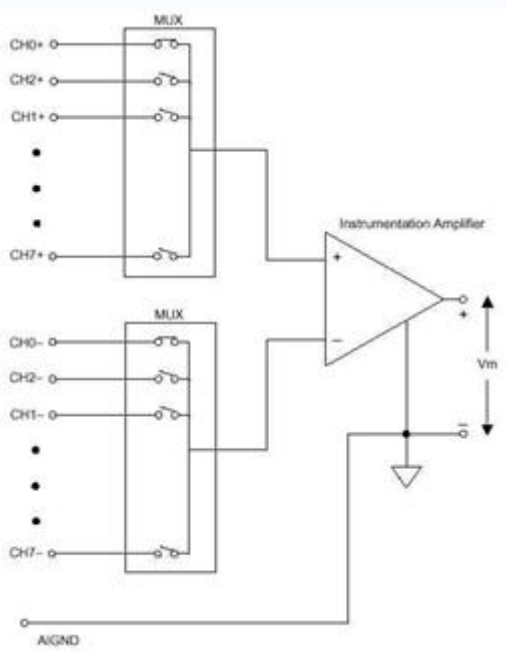


图 11-27 差分测量系统

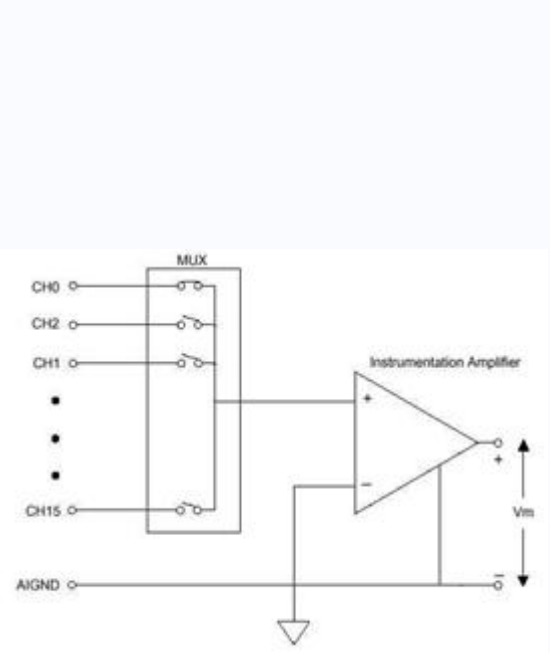


图 11-28 参考地单端测量系统

3 · 无参考地单端测量系统（NRSE）

在 NRSE 测量系统中，信号的一端接模拟输入通道，另一端接一个公用参考端，但这个参考端电压相对于测量系统的地来说是不断变化的。图 11-29 说明了一个 NRSE 测量系统，其中 AISENSE 是测量系统的公共参考端，AIGND 是系统的地。

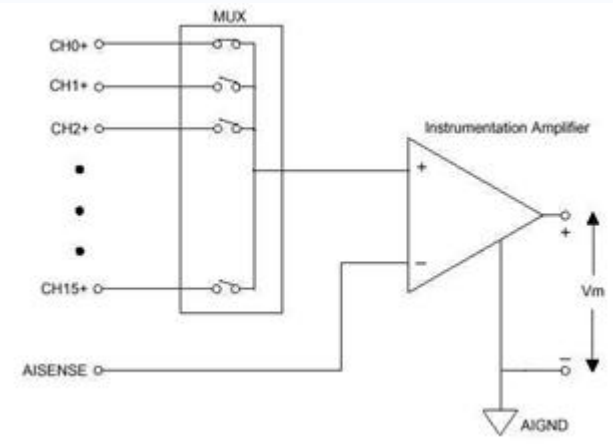


图 11-29 无参考地单端测量系统

4 · 选择合适的测量系统

两种信号源和三种测量系统一共可以组成六种连接方式：

	接地信号	浮动信号
差分	*	*
RSE		**
NRSE	*	*

其中，不带*号的方式不推荐使用。一般说来，浮动信号和差动连接方式可能较好。但实际测量时还要看情况而定。

（1）测量接地信号

测量接地信号最好采用差分或 NRSE 测量系统。如果采用 RSE 测量系统，将会给测量结果带来较大的误差。图 11-30 展示了用一个 RSE 测量系统去测量一个接地信号源的弊端。在本例中，测量电压 V_m 是测量信号电压 V_s 和电位差 DV_g 之和，其中 DV_g 是信号地和测量地之间的电位差，这个电位差来自于接地回路电阻上的压降，可能会造成数据错误。接地回路通常会在测量数据中引入频率为电源频率的交流和偏置直流干扰。一种避免接地回路形成的办法就是在测量信号前使用隔离方法，测量隔离之后的信号。

如果信号电压很高，并且信号源和数据采集卡之间的连接阻抗很小，也可以采用 RSE 系统，因为此时接地回路电压相对于信号电压来说很小，信号源电压的测量值受接地回路的影响可以忽略。

（2）测量浮动信号

浮动信号可以用差分、RSE、NRSE 方式测量。在差分测量系统中，应该保证相对于测量地的信号的共模电压在测量系统设备允许的范围之内。如果采用差分或 NRSE 测量系统，放大器输入偏置电流会导致浮动信号电压偏离数据采集卡的有效范围。为了稳住信号电压，需要在每个测量端与测量地之间连接偏置电阻，如图 11-31 所示。这样就为放大器输入到放大器的地提供了一个直流通路。这些偏置电阻的阻值应该足够大，使得信号源可以相对于测量地浮动。对低阻抗信号源来说，10kW 到 100kW 的电阻比较合适。

如果输入是直流信号，就只需要用一个电阻将“-”端与测量系统的地连接起来。

然而如果信号源的阻抗相对较高，从免除干扰的角度而言，这种连接方式会导致系统不平衡。在信号源的阻抗足够高的时候，应该选取两个等值电阻，一个连接信号高电平（+）到地，一个连接信号低电平（-）到地。如果输入是交流信号，就需要两个偏置电阻，以达到放大器的直流偏置通路的要求。

总的来说，不论测量接地还是测量浮动信号，差分测量系统是很好的选择，因为它不但避免了接地回路干扰，还避免了环境干扰。相反的，RSE 系统却允许两种

干扰的存在，在所有输入信号都满足以下指标时，可以采用 RSE 测量方式：输入信号是高电平（一般要超过 1V）；连线比较短（一般小于 5m）并且环境干扰很小或屏蔽良好；所有输入信号都与信号源共地。当有一项不满足要求时，就要考虑使用差分测量方式。

本例采用 NI 公司的 PCI-6133S 型数据采集卡以便于系统的集成，该数据采集卡具有 8 路模拟同步差分输入功能，可以从硬件上保证计算精度，其主要技术指标如表 11-1 所示。数据采集卡与信号之间的接口采用 NI 公司的 SCB-68、SCB-68 是具有屏蔽作用的 I/O 接线盒，具有较好的抗干扰能力。

表 11-1 PCI-6133s 的主要技术指标

模拟输入	最高采样率/Hz	输入分辨率/位	输入电压范围/V	数字I/O/位
8 路同步	3M	14	-10~+10	8

5 · 数据采集系统电路接线图

差分输入具有抑制共模信号、环境抗干扰能力强的优点，为此测量电压电极采集信号和参考电阻的信号都采用差分输入，鱼体复阻抗数据采集系统的电路接线图如图 11-32 所示。

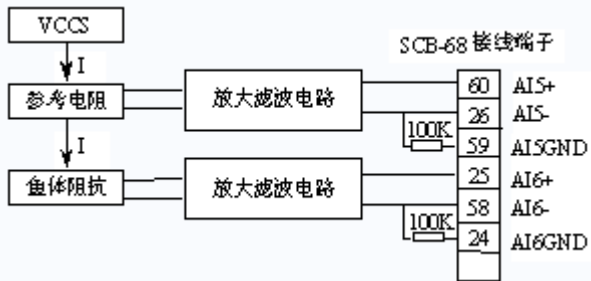


图 11-32 鱼体复阻抗数据采集系统的接线图

6 · 数据采集程序的设计

采用虚拟仪器构建测控平台，其最方便之处是数据采集程序简单，它提供了硬件标准的程序接口，囊括了PCI、GPIB、PXI、VXI、RS232/485、USB 等各种仪器通信总线标准的所有功能函数，使得不懂得总线标准的开发者也能够驱动不同总线标准接口设备与仪器，工程师不需要懂得太多的编程技巧也可完成任务。由于虚拟仪器数据采集驱动已经发展到第三代，数据采集系统直接采用 DAQmx 驱动，更能缩短开发时间。

数据采集设置模块主要由 DAQmx Create Virtual Channel、DAQmx Timing、DAQmx Start Task、DAQmx Read、DAQmx Clear Task 这五个 VI 组成，简单的数据采集程序如图 11-33 所示，需要配置数据采集卡模拟通道、测量电压范围、采样频率和采样点。

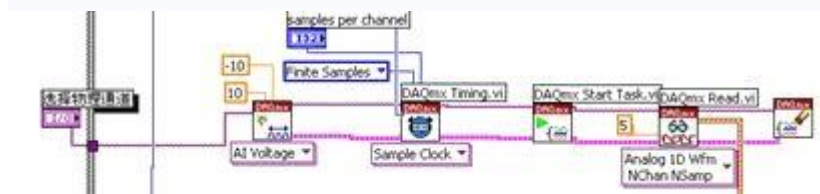


图 11-33 数据采集程序

尽管基于虚拟仪器的信号发生器已经产生了较精确的信号，但为了能更准确地提取鱼体复阻抗的信息，需要实现信号的精确整周期采样，就必须预采样一次以获取信号更精确的频率。本例使用 LabVIEW 自带的信号分析工具 Tone Measurement 提取频率，参数配置如图 11-34 所示，根据此频率调整采样率，实际使用的数据采集系统流程图如图 11-35 所示。

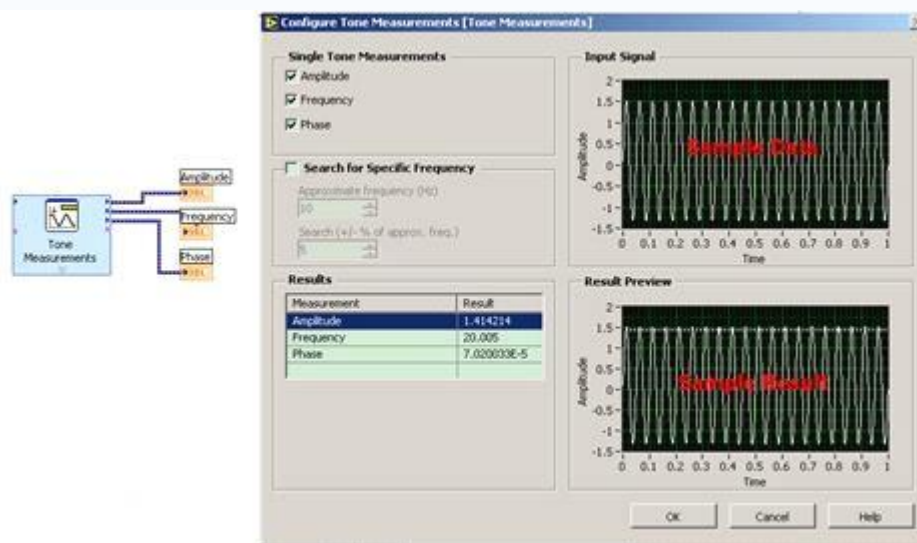


图 11-34 Tone Measurement 的参数配置

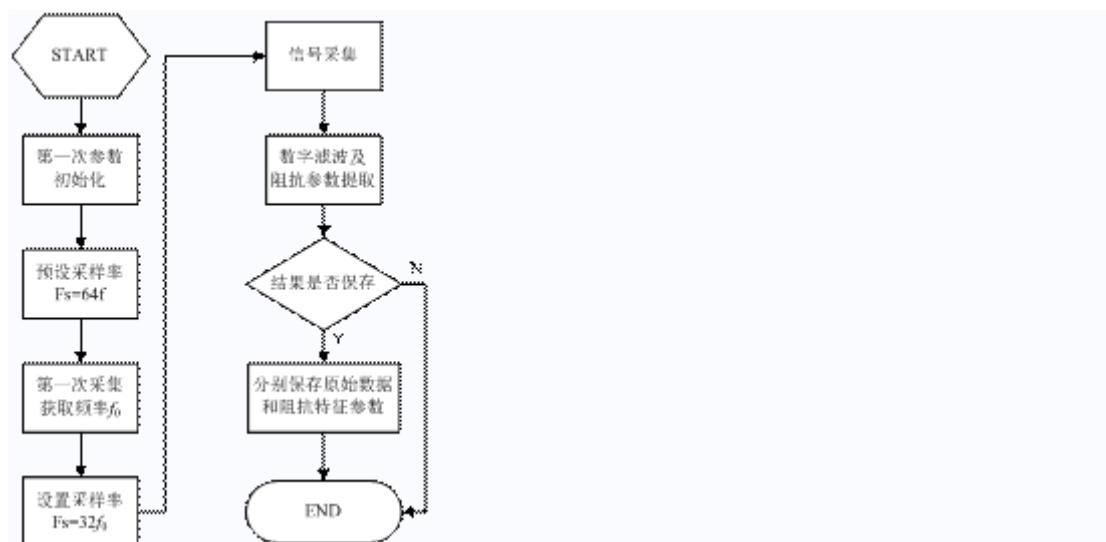
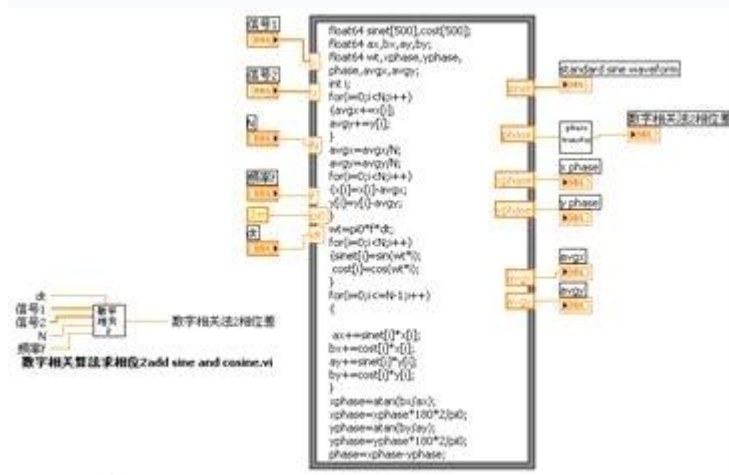


图 11-35 数据采集流程图

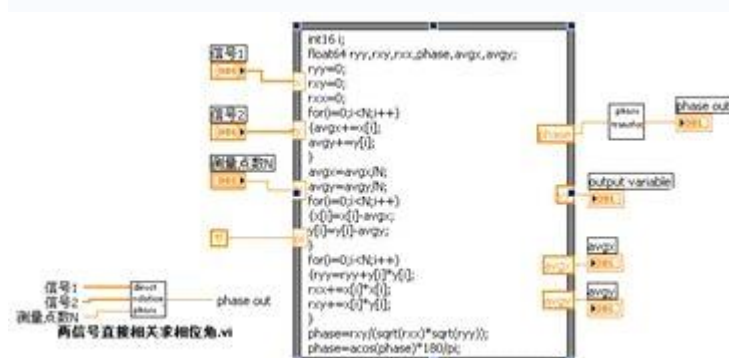
11.2.5 信号分析模块

生物阻抗是一个复阻抗特性，可以表示成 $Z=|Z|\angle\varphi_Z$ 形式。信号幅值可以利用 Tone Measurement.vi 获取；求取相位角的方法比较多，前面的数字解调算法是一种比较常规的生物阻抗提取方法，很多学者还提出了很多其他数字解调算法，利用 LabVIEW 中的公式节点实现算法，而其内部使用 C 语言实现。

研究鱼体阻抗的相位角变化规律与鱼体贮藏品质的关系，需要准确获取相位角，因此本例除了采用前面介绍的数字解调方式获得相位差（程序中称为加标准正弦和余弦法，方法a），还根据自相关和互相关函数能求信号相位差特点采用了两种方法：利用阻抗信号和参考电阻直接求相位差（程序中称为直接相关法，方法b）和对测量信号乘以标准同频正弦信号进行相关求相位差（程序中称为加标准正弦法，方法c），还采用了 LabVIEW 自带的函数 Tone Measurement.vi 求相位角（程序中称为 LabVIEW 自带相位角法，方法d）以及利用 Fourier 级数将信号展开求基波系数来求相位差（程序中称为FFT 法，方法e），各算法求相位差的程序框图如图 11-36 所示。

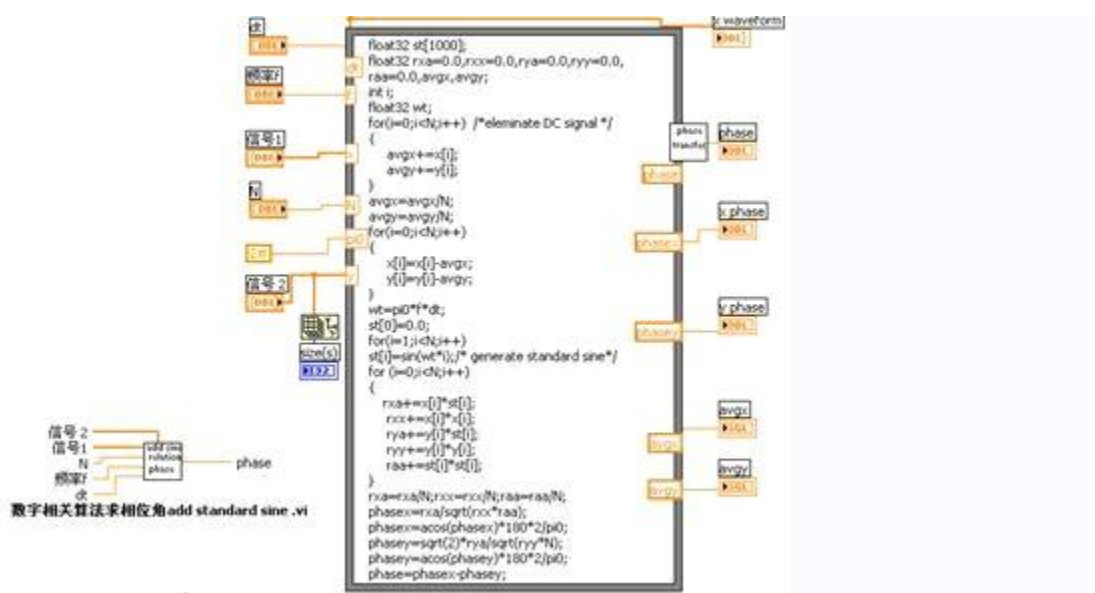


(a) 加标准正弦和余弦法程序框图

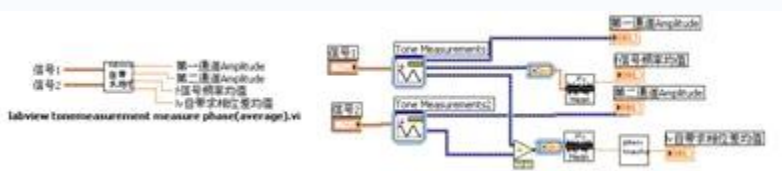


(b) 直接相关法程序框图

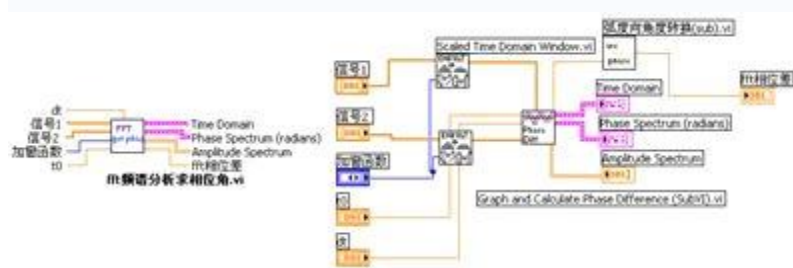
图 11-36 相位差求取方法



(c) 加标准正弦程序框图



(d) LabVIEW 自带相位角法程序框图



(e) FFT 法程序框图

图 11-36 相位差求取方法 (续)

11.2.6 文件保存模块

根据 Cole-Cole 理论, 需要扫频测量才能研究生物体的电特性, 如果从 10Hz~100kHz 每十倍频扫频 10 点, 则每次测量将产生 40 个文件, 为了方便进行离线数据处理和分析, 测量文件必须按照一定格式自动存储, 我们研究设计了自动文件保存模块, 可根据试验参数、试验次数、测量频率、测量时间自动为文件命名并生成通用的 Excel 文档, 实现智能化检测, 以原始数据保存为例, 其 LabVIEW 程序如图 11-37 所示。

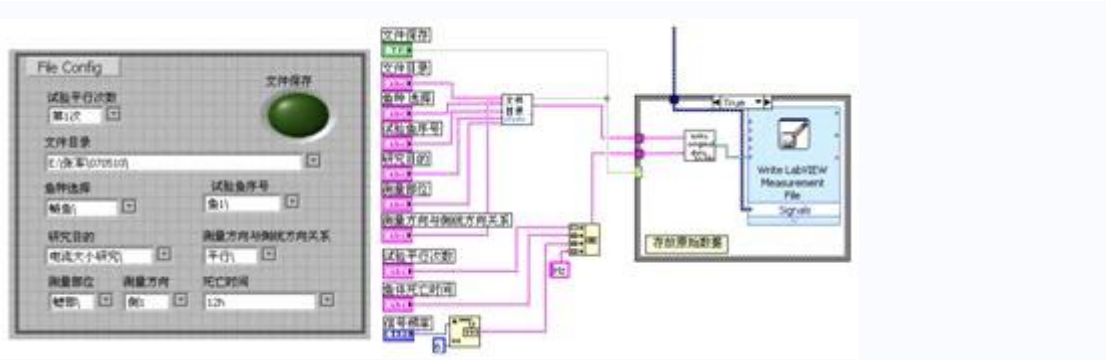


图 11-37 文件自动保存模块

整个写文件的子程序主要由创建文件目录与生成文件名称两个模块组成，其中创建文件目录的子 VI 程序框图如图 11-38 所示。

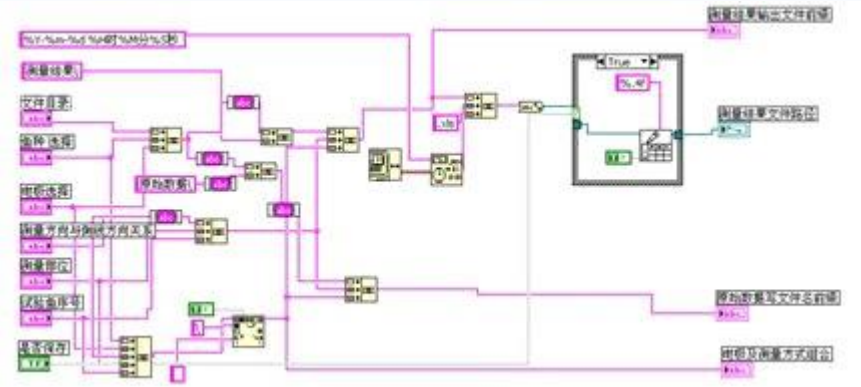


图 11-38 创建文件目录的子 VI 程序框图

11.2.7 实例小结

本例介绍了鱼体复阻抗测量系统的设计过程，主要利用虚拟仪器采用“软件就是仪器”的思想，通过开发不同功能的程序来代替硬件功能，以节约大量的硬件成本。本例的重点在于数据采集和信号分析模块的设计。

11.3 多功能形位误差检测仪设计

本例所述的多功能形位误差测量系统是基于 LabVIEW 软件平台设计的，可对轴类零件的直线度、圆度、同轴度等形位误差进行测量，并且可以采用不同的方法自动评定误差，是一种精度高、功能多、性能稳定、测量速度快、操作简单和使用方便的形位误差测量系统。

11.3.1 实例内容说明

目前工件的直线度、圆度多数是先采用自准直仪和圆度仪测量，再采用人工计算和作图的方法，最后得出结论。这种方法存在两方面的不足：一是测量误差大、精度低，二是劳动强度大、效率低。

本例采用计算机来进行数据处理，以 LabVIEW 为软件开发工具和设计平台，利用软件模拟仪器部分硬件功能，并结合通用的数据采集设备实现真实仪器的功能。本检测系统连续自动地对工件进行测量，并且可以用不同的方法评定误差，绘制出各项误差曲线图，实现各项误差的自动评定。

本检测系统以计算机为核心，用位移传感测量位移，被测工件与电机 1 同轴安装，测头与位移传感器安装在可升降调整的丝杠螺母上，在径向测量力的作用下与被测工件保持接触，两路信号经 A/D 数模转换送进计算机内，再由软件处理计算得到各项误差。其总体设计方案示意图如图 11-39 所示。

11.3.2 硬件设计

本检测系统选用的是北京阿尔泰科技的 PCI2003 数据采集卡，这是一种基于 PCI 总线的数据采集卡，可以直接插在 IBM-PC/AT 或与之兼容的计算机内任一 PCI 插槽中，构成实验室、产品质量检测中心等各种领域的数据采集、波形分析和处理系统，也可以构成工业生产过程控制监控系统。此外，PCI2003 有完备的函数模块，极大地方便了应用程序的开发，而不需要用户对硬件有过深了解，真正做到即插即用。

PCI2003 数据采集卡是一个具有 16 路单端和 8 路双端模拟输入通道的标准的 PCI 总线插卡，卡上集成的 12 位 ADC，采样通过频率为 100KHz，为了获得良好的信号，减小外界因素对信号的影响，在系统中选用了一块与 PCI2003 配套使用的调理端子板—BH5614A，将它置于信号和数据采集卡之间，以实现信号的调理。

11.3.3 软件设计

本节详细讲解开发过程。下面将分别对各部分程序单元进行说明。

1 · 软件设计流程

程序的流程图如图 11-40 所示。

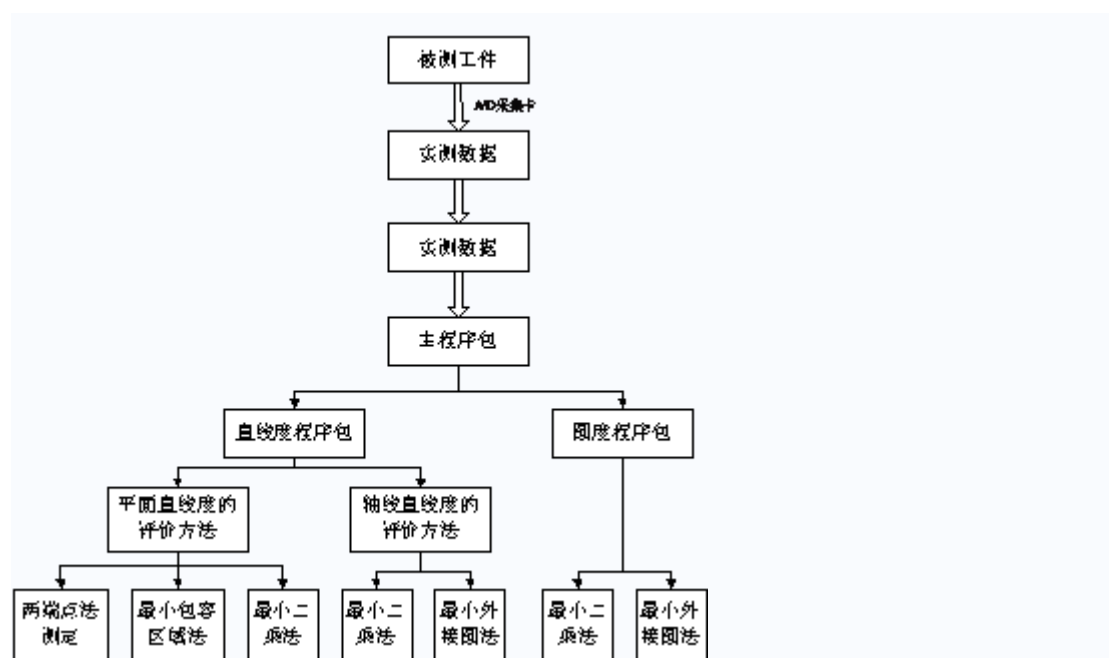


图 11-40 程序流程图

主程序的前面板界面如图 11-41 所示，后面板如图 11-42 所示。

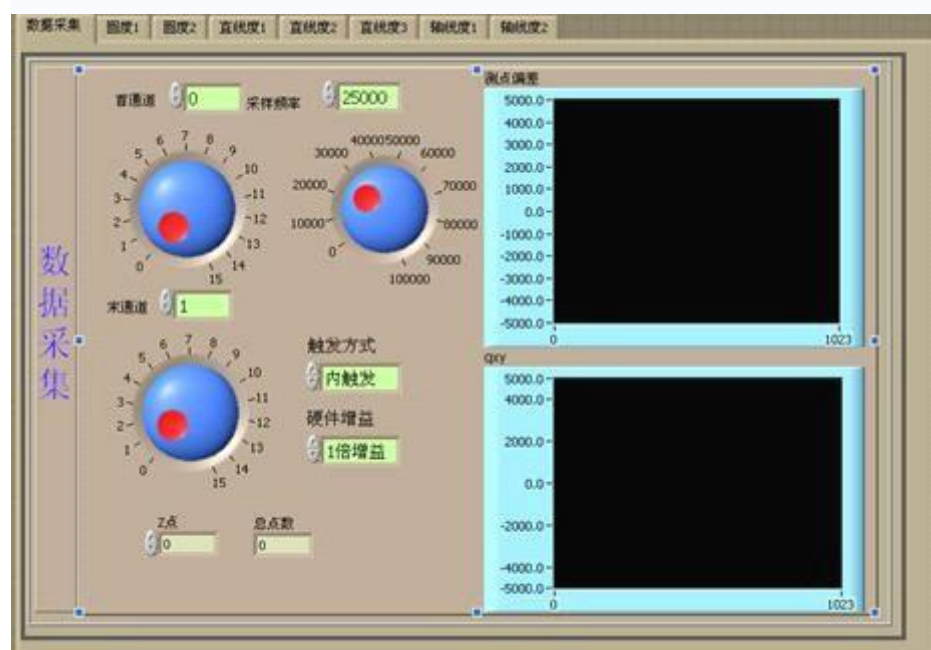


图 11-41 主程序前面板

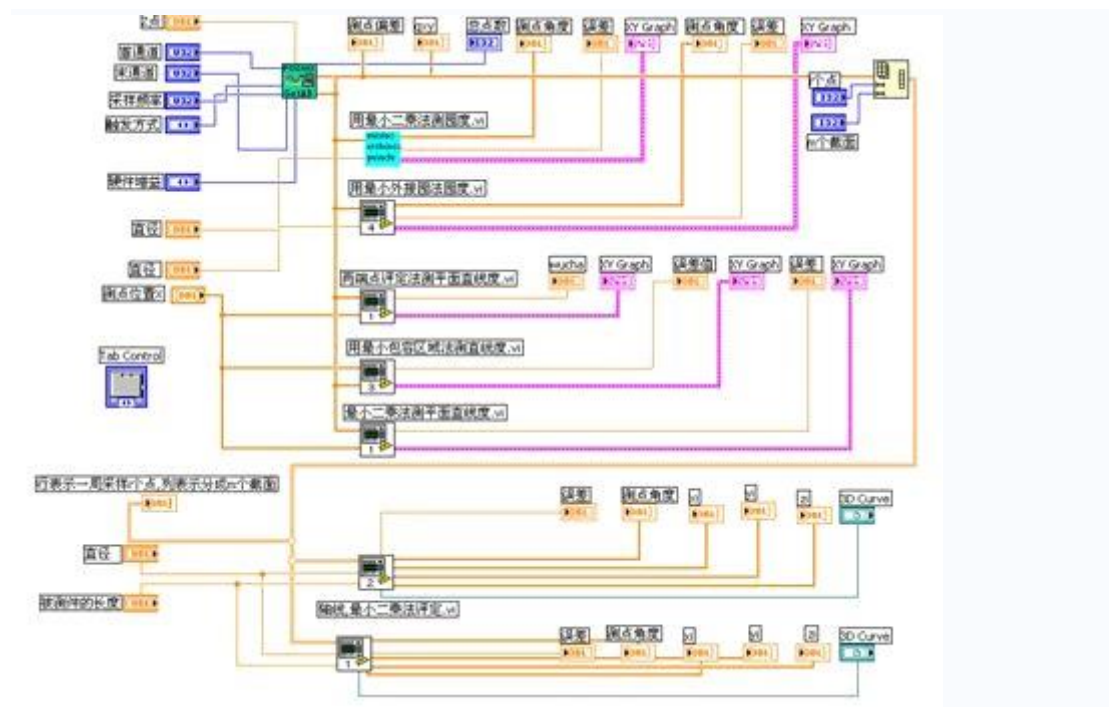


图 11-42 主程序后面板

2 · PCI2003 数据采集卡的操作

PCI2003 并非 NI 公司生产的硬件, 故不能采用 DAQmax 来配置数据采集参数。不过可以使用 Call Library Function 节点进行配置。在 LabVIEW 中调用 win32

API GetTickCount 函数，获取系统的开机时间。具体步骤为：

- (1) 在 LabVIEW 前面板放置一个数字指示框 (Numeric Indicator) 与一个按钮 (Button) 控件, 实现按下按钮后, 在数字指示框中显示系统的开机时间。
- (2) 在后面板中, 放置一个 While 框与一个 Case 框、用 While 循环检测按钮是否被按下, 如果按下, 则停止循环, 并向 Case 框输出真值。
- (3) 右键单击 Case 框, 编写代码, 在弹出的 Functions 面板中选择 “Advanced” | “Call Library Function Node”, 并将 Call Library Function Node 放置在 Case 框 True 条件中。
- (4) 右键单击选择的 Call Library Function Node, 在出现的快捷菜单中选择 “Configure”, 出现设置对话框, 如图 11-43 所示。
- (5) 在 “Library name or path” 栏中选择所调用 API 函数的动态连接库 (.dll) 的位置与文件名。
- (6) 在 “Function name” 栏中选择欲调用的函数。
- (7) 在 Calling conventions 中选择 “c” 或 “Stdcall” (Win API)。

(8) 在“Parameter”选项卡页面选择输入输出参数的类型，单击“OK”按钮完成设置。

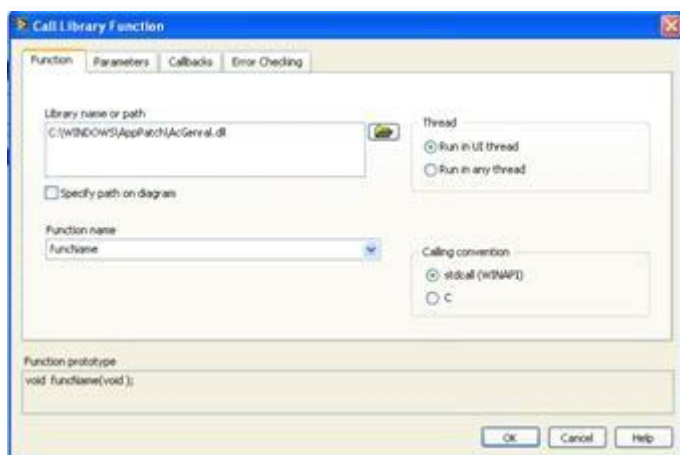


图 11-43 Call Library Function 节点配置

(9) 将 Call Function Node 的输出连接到数字指示框，即可按照 PCI 数据采集卡的操作时序完成具体操作，具体实现如图 11-44 所示。

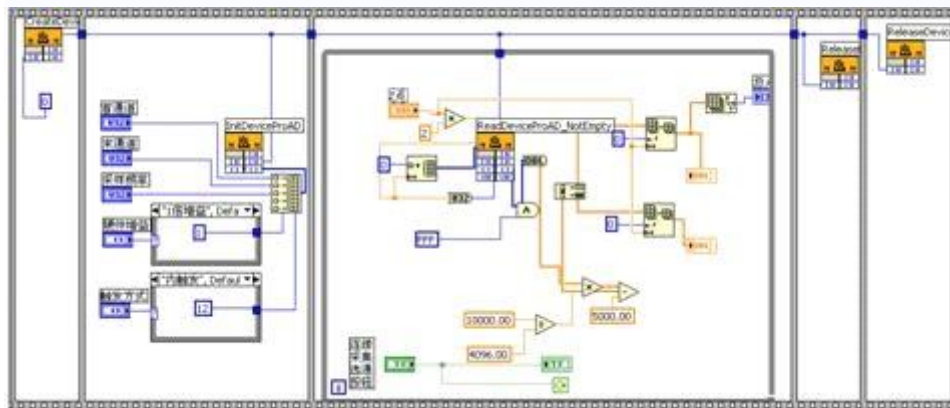


图 11-44 配置 PCI2003 数据采集卡

3 · 数据分析模块

(1) 圆度测量

圆度公差带是垂直于被测工件轴线的任意正截面上半径为公差值 t 的两个同心圆之间的区域，圆度测量的常用的算法有最小二乘法和最小外接圆法。

一 用最小二乘法测圆度

用最小二乘法测圆度即求最小二乘圆。最小二乘圆是一个穿过实际被测轮廓的圆，它所处的位置使实际被测轮廓上各测点至它的距离的平方之和为最小。具体程序如图 11-45 所示。

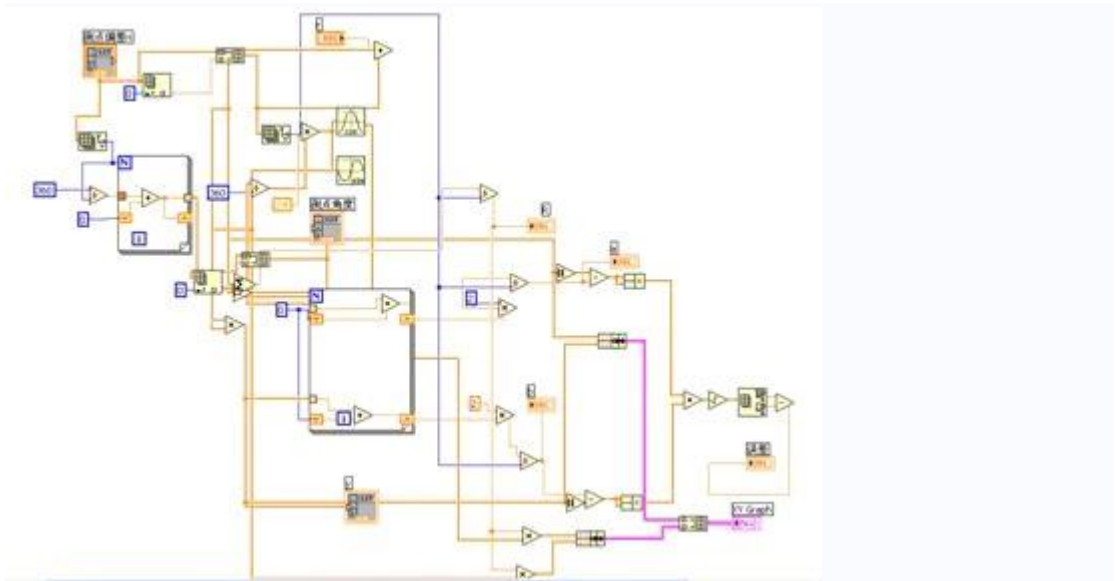


图 11-45 最小二乘法测圆度程序

一 用最小外接圆法测圆度

最小外接圆是指外接于轴的实际被测轮廓的可能最小圆，最小外接圆测圆度的原理如图 11-46 所示。具体程序如图 11-47 所示。

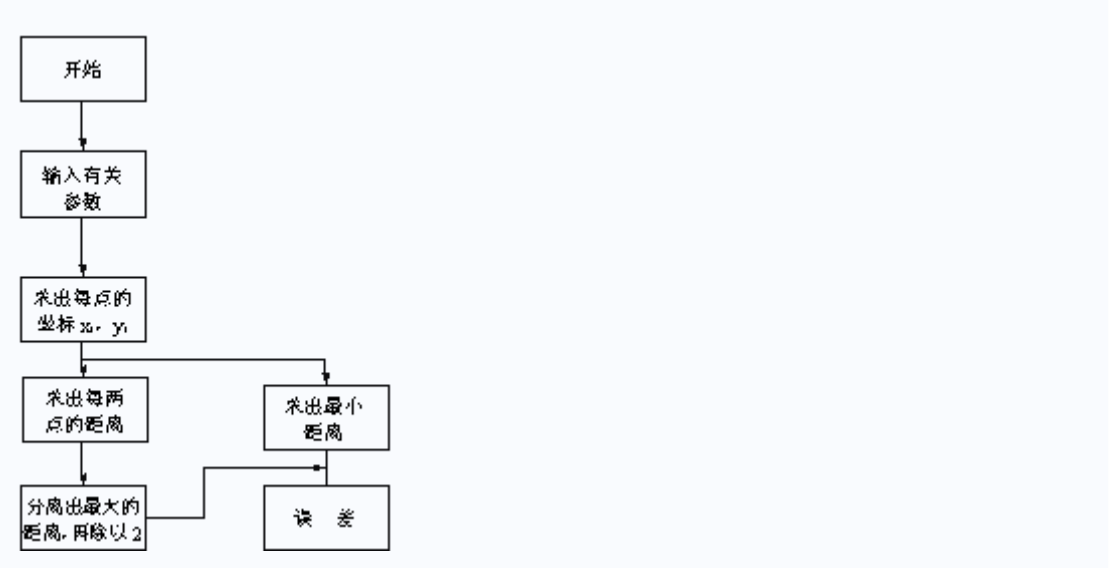


图 11-46 最小外接圆法测圆度原理

(2) 直线度测量

直线度测量方法有 3 种：两端点测平面直线度、最小包容法测直线度、最小二乘法测直线度。下面分别介绍。

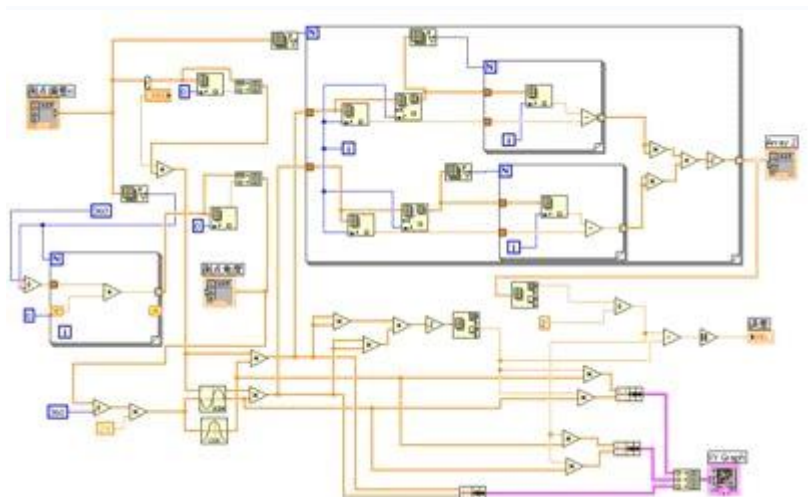


图 11-47 最小外接圆法测圆度程序

— 两端点测平面直线度

如图 11-48 所示，以实际被测直线 S 的首、末两端点 B 和 E 的连线 l_{BE} 作为评定基准， $f_{BE}=h_{\max}-h_{\min}$ 。其程序实现如图 11-49 所示。

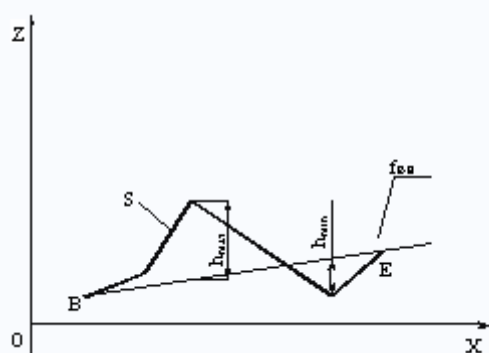


图 11-48 两端点测平面直线度原理

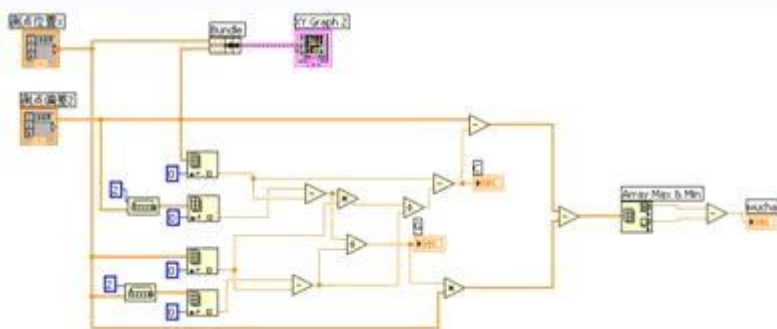


图 11-49 两端点测平面直线度的程序

— 最小包容法测直线度

如图 11-50 所示，根据给定平面内直线度公差带的形状，由两条平行直线包容实际被测直线 S 时，成“高一低一高”三极点相间接触，则这两条平行直线之间的区域就是最小包容区域 U （简称最小区域），直线度的误差就是 f_{MZ} 。程序实现如图 11-51 所示。

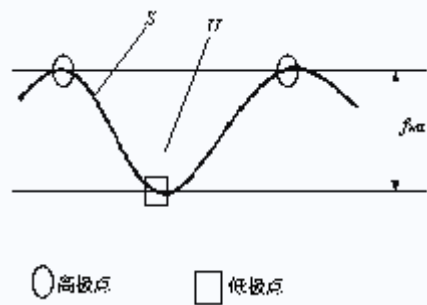


图 11-50 最小包容法测直线度

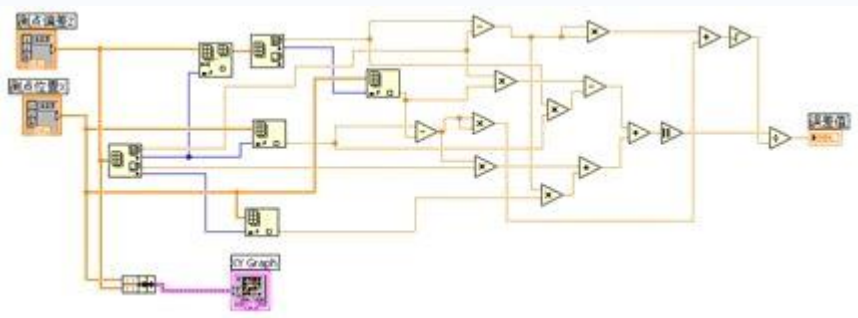


图 11-51 最小包容法测平面直线度的程序

— 最小二乘法测直线度

如图 11-52 所示，最小二乘中线 l_{BE} 是一条穿过被测直线 S 的理想直线，它所处的位置使被测直线上各点至它的平方之和为最小， $f_{LS}=h_{max}-h_{min}$ ， f_{LS} 为直线度误差。程序实现如图 11-53 所示。

11.3.4 实例小结

本例介绍了多功能形位误差测量系统设计的过程，该系统可对轴类零件的直线度、圆度、同轴度等形位误差进行测量，并且可以采用不同的方法自动评定误差。本系统主要通过 PCI2003 数据采集卡来实现，所以读者熟悉 PCI2003 数据采集卡的操作很有必要。

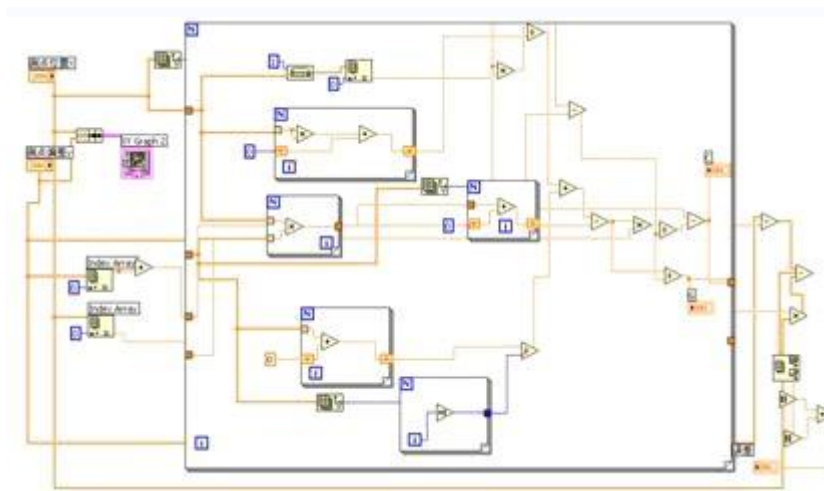


图 11-53 最小二乘法测直线度的程序实现

11.4 无线设备自动测试系统设计

无线通信网络设备的技术指标都需要用专用的仪表进行测试，如果是多台仪表的互动测试，则需要人工随时操作，不仅费时费力，而且容易出现误差，这种情况最适合采用自动测试系统 ATS 完成。随着现代测试测量仪器功能的不断提高和完善，各种各样的智能化仪器随处可见，而且它们都具有 GPIB 接口或串口，能很方便地与计算机组建自动测试系统 ATS。

自动测试系统一般是指针对某类特定的测试项目而组建的智能测试系统，当前比较流行的组建方案是基于 NI 公司的 GPIB 产品和 LabVIEW 软件开发平台。

本例将介绍无线自动测试系统的设计和开发方法。该系统是笔者基于 NI 公司 GPIB 产品，利用其图形化软件开发工具 LabVIEW 8.5 和 VC++ 6.0 开发工具联合开发的。

11.4.1 自动测试系统简介

自动测试系统（Auto Test System）就是以控制计算机为核心，以标准接口总线为平台，以程控仪表为被控对象，用计算机测控软件设计的自动化系统。常用的自动测试系统一般都是采用虚拟仪器技术，通过 LabVIEW 等图形化的编程语言来控制仪器，以完成数据采集、数据分析和显示测试的数据结果。

自动测试系统是相对传统测试而言的，传统测试方法，人工干预多、操作易出错、测试速度和稳定性难以保证，而自动测试系统利用虚拟仪器技术把仪器的部分功能软件化，用户可以通过编写应用软件来实现不同的测试功能，减少了人工干预，

因而提高了测试速度和可靠性。普通测试设备一般需要人工记录测试数据，而采用自动测试技术，测试设备得到的数据将被送到计算机，计算机根据实际需要调用 LabVIEW 等软件自带的分析处理函数，实时、直接地对测试数据进行分析处理和数据库存储等工作，并可将测试和分析结果显示在界面上，方便观测。

随着计算机技术、通信技术和仪器仪表的智能化技术的发展，自动测试技术越来越多地在测试测量领域被采用，特别是在人工无法完成的测试任务中被广泛采用，尤其适合生产性企业。

11.4.2 自动测试系统结构

采用虚拟仪器技术实现的自动测试系统结构可分为如下四个层面：

1 · 测试管理层

采用虚拟仪器生产厂商开发的应用程序，组成测试仪器。

2 · 应用开发层

这一层是用生产商提供的软件开发工具，如 NI 公司的 LabVIEW 软件，LabWindows/CVI 软件。根据测试项目的需要，开发应用和管理程序，以扩展仪器原有的功能。

3 · 仪器驱动层

这一层是由仪器生产商开发，针对不同类型的仪器有不同的驱动程序接口，一般采用 DLL 形式提供。为了给用户提供方便、易用的仪器驱动程序，泰克、惠普和 NI 公司等仪器公司成立了 VXI Plug & Play 系统联盟，并于推出 VISA (Virtual Instrument Software Architecture) 标准。

4 · I/O 总线驱动层

这一层由生产商开发，用于将不同类型的实际仪器通过相同标准的总线连接起来形成一套完整的测试系统，比如广泛应用的 VXI 总线系统。

GPIB 技术可用计算机实现对仪器的操作和控制，替代传统的人工操作方式，可以很方便地把多台仪器组合起来，形成自动测量系统。一般程控仪器都提供 GPIB 接口，在自动测试系统的设计过程中，首选的方式就是利用 GPIB 接口将仪器与计算机连接起来，然后通过软件实现自动测试。

一般组建基于虚拟仪器的自动测试系统可分为以下 5 个步骤：

(1) 选择操作系统和软件开发平台。组建虚拟仪器时，绝大多数人会采用 Windows 系列操作系统，开发软件平台可选择 LabVIEW、LabWindows/CVI、VC 或 VB 等。

(2) 选择总线形式和总线机箱。总线形式可选择 PXI、VXI、PCI、USB 或 GPIB 等。

(3) 选择机箱内嵌式或外置式计算机。内嵌式计算机使虚拟仪器系统结构紧凑，外置式计算机成本较低。

(4) 选择仪器模块。根据实际需要，从模块化仪器供应商处采购各种仪器模块。

(5) 选择信号调理模块和自制专用模块。组建一个虚拟仪器系统时通常还会碰到信号调理模块的选择问题，当信号特殊时，还需自制部分信号调理模块。本例介绍的无线设备自动测试系统就是根据以上系统结构设计的，利用计算机连接仪器的 GPIB 接口，然后根据设备测试需要编制相应软件程序，实现无线设备的自动化测试任务。

11.4.3 系统功能说明

本自动测试系统主要分为两个模块：前台仪器控制和数据采集模块、后台数据管理和分析模块。前台仪器控制和数据采集模块是本系统的核心模块，主要完成对仪表的控制数据和测量结果的收集，以及实时预警分析。系统利用 NI 公司 GPIB 产品和其 LabVIEW 8.5 开发平台，建立一个标准的虚拟仪器平台，实现了对干放、直放站和基站等设备进行自动测试所需要的仪表的自动化控制。系统功能如图 11-54 所示。

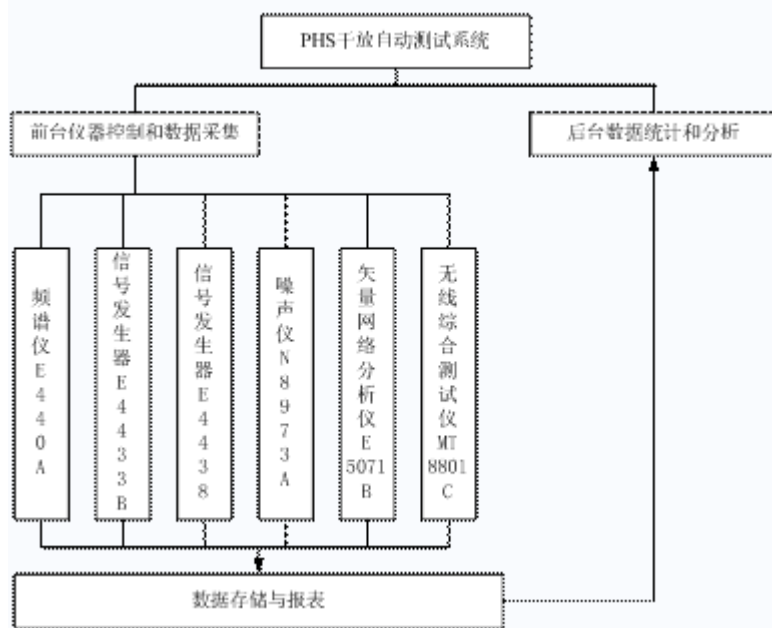


图 11-54 系统功能图

系统结构如图 11-55 所示。该系统主要控制了多台仪表，包括频谱仪 E440A、信号发生器 E4433B、信号发生器 E4438、噪声仪 N8973A、矢量网络分析仪 E5071B、无线综合测试仪 MT8801C 等。这些仪表分别完成设备的信道功率、邻道功率泄漏 ACP、自动电平控制 ALC、三阶交调、杂散发射、带外抑制、饱和功率、噪声系

数、驻波比、传输时延、平坦度、RMS、峰值误差和占用带宽等的测试，并且自动对采集的数据进行报表输出，同时将数据自动存储以供后台进行数据分析。

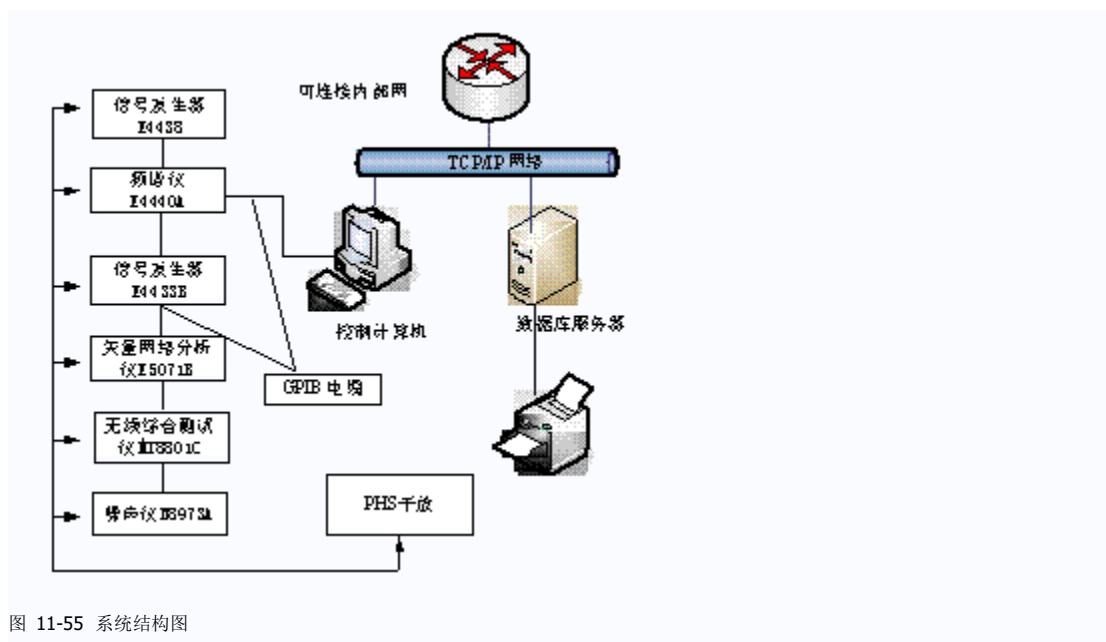


图 11-55 系统结构图

程控仪器软件的编写分为两种方式：一种是利用仪器厂商提供的函数包，另一种是直接利用 VISA 语句编写程控仪器指令。前者使用方便，函数是现成的，可直接调用，但这种方式可移植性差，函数包需要厂商提供的仪器驱动程序支持，不同类型仪器需要安装不同的设备驱动。后者需要开发者掌握 VISA 语言和底层的 GPIB 指令，但可以脱离不同厂家的仪器驱动程序，实现对任意 GPIB 设备的远程控制。本例中对 E4440A、E4433B、E4438 和 E5071 仪器的控制采用前者方式编写，N8973A 和 MT8801C 采用后者方式开发。

前台仪器控制与数据采集模块主要是由计算机通过 GPIB 接口对各仪器发送不同的指令来实现，在本系统中，频谱仪、信号发生器和矢量网络分析仪使用了 LabVIEW 8.5 的仪表驱动程序进行编程，由于其他仪表（噪声仪、无线综合测试仪）没有合适的驱动程序可以调用，就利用标准 VISA 接口发送 SCPI 指令直接对仪表进行控制。

对于数据存储与报表，由于购买的 NI 公司开发者套件开发包中不包含数据库操作工具包（需另行购买），为了节省开发成本，本例采用 Microsoft Visual C++ 6.0 开发与 Office Access 和 Excel 交互操作的动态连接库（DLL），然后在 LabVIEW 中调用 DLL。这样，本系统采用了 LabVIEW 和 VC++ 混合编程的方式实现，实践证明此方法可行。

对于采集到的数据的处理，可将采集的测试数据直接存入 Access 表，人工导入到后台数据服务器进行分析，也可通过前台软件设置使采集数据自动存储在服务器上，供后台软件进行分析。

开发人员可选择不同的软硬件平台方案。由于 LabVIEW 是虚拟仪器的主流软件开发平台，建议选用它作为软件平台；硬件部分选择余地比较大，目前很多厂家都可以提供 GPIB 硬件产品，不过考虑到兼容性，建议在价格允许的情况下，优先选择 NI 公司 GPIB 产品，本项目中就选择了 NI 公司的 PCI-GPIB 卡和 USB-GPIB 线缆，这样的方案可以得到 NI 公司的良好售后服务和技术支持。

具体系统开发方案如图 11-56 所示。

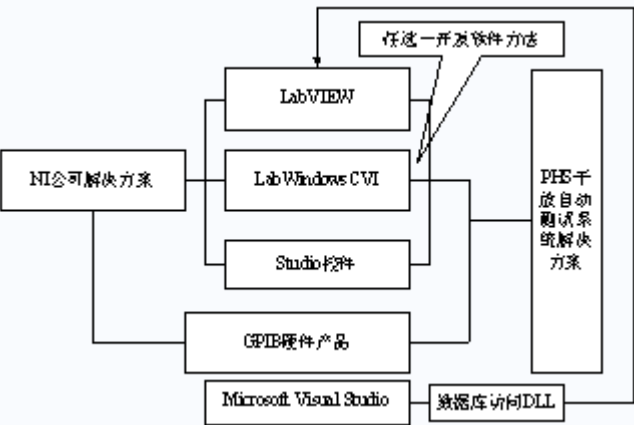


图 11-56 系统功能图

通过 LabVIEW 开发平台，可以对每个模块进行分类，并使每个模块被主程序调用，最后利用相关仪器的驱动程序模块或通过 SCPI 编写的程序模块调用底层。系统中每台仪器有一个 0~30 的 GPIB 地址。由一个控者控制总线，在总线上传送仪器命令和数据，控者寻址一个讲者，一个或多个听者，数据串在总线上从讲者向听者传送。GPIB 软件包自动处理寻址和其他的总线管理功能。

本例中各仪器的 GPIB 地址如图 11-57 所示。

11.4.4 项目方案选择

无线通信产品的质量控制，是运营商室内分布系统建设以及移动通信网络建设的重要环节。无线通信产品的测试和评估的大部分工作量都集中在对仪表进行操作的实验室测试以及测试数据的人工判别与处理上。目前，这种繁重的实验室测试、分析工作以人工操作为主，存在着明显的不足：

- 测试人员读数可能带来误差，导致测试结果出现一定的偏差，难以保证高精度测试，需反复核对，费时费力。
- 一些测试指标难以通过人工测试完成，比如功率稳定度，使得测试能力受到限制，难以保证无线设备测试评估的全面性。
- 测试过程需要大量的重复性工作，并产生大量的测试数据，需要烦琐的后期数据处理，人工操作力所难及。
- 依靠人工手段的数据处理耗时耗力，分析手段和分析方式也受到较大限制，给测试数据的处理带来一定难度，且缺乏统一、灵活的综合评分配置模板，影响了对无线设备测试数据的深度挖掘工作。

这些因素都在较大程度上制约了工作效率和测试精度的提高，影响着对无线设备的深度分析与评估工作。

为适应无线网络建设对无线设备测试提出的高要求，适时建设一套符合企业发展要求的自动化综合测试分析平台是十分必要的。这将有利于保证测试工作的自动化、系统化和规模化，保持检测技术的先进性，对打造产品的质量监控品牌，树立优良的产品质量监控和评估形象，具有重要的意义。

本自动测试系统中有源器件包括 PHS 干放、CDMA 干放、CDMA 直放站、CDMA 基站等，无源器件包括功分器、耦合器、电桥、衰减器、合路器和负载等。

11.4.5 系统平台选择

考虑到通用性，本自动测试系统采用了 Windows XP 操作系统，系统运行具体要求如下：

1 · 运行平台

本系统中的测控计算机需要满足如下软硬件要求：

（1）软件要求

- 操作系统 Windows XP
- LabVIEW 8.5 或以上版本
- Microsoft Visual Studio 6.0 或以上版本
- Microsoft Office 2000 或以上版本
- ICP25 和 VISA3.6 驱动程序
- NI 488.5 V2.4 驱动程序
- E4440A、E4433B、E5071B 仪器的 LabVIEW 8.5 驱动程序

（2）硬件要求

最低配置：

- 处理器：>1.2GHz
- 内存：256 MB
- 硬盘：1GB
- 显示器：1024×768，16 位

推荐配置：

- 处理器：>2.6GHz
- 内存：1GB
- 硬盘：1GB
- 显示器：1024×768，32 位

2 · 开发平台

本系统中的测控计算机开发环境需要配置以下环境：

- 操作系统 Windows XP
- LabVIEW 8.5
- Microsoft Visual Studio 6
- Microsoft Office 2000 或以上版本
- ICP25 和 VISA3.6 驱动程序
- NI488.5 V2.4 驱动程序
- E4440A、E4433B、E5071B 仪器的 LabVIEW 8.5 驱动程序

11.4.6 数据管理

在本系统中数据在本地以 Access 表方式进行存储，后台可将其测试好的结果表导入软件进行统计和分析等。本例中设计的数据库表结构如下所述。

1 · 厂家信息表

测试系统针对不同厂家的测试设备进行分类，并且在测试前设置测试厂家信息，这些信息保存至厂家信息表中，厂家信息表结构如表 11-2 所示。

表 11-2 厂家信息表

属 性	数据类型	数据库对应属性名	备 注
厂家	文本	C_Manufacturer	Primary key

地址	文本	C_Address	
联系电话	文本	C_TelNum	
联系人	文本	C_LinkMan	
EMAIL	文本	C_EMAIL	
厂家代号	文本	C_ManuCode	

2 · 器件信息表

测试系统针对不同厂家的测试设备进行分类，并且在测试前对测试设备和器件进行设置，并将这些设置信息存入器件信息表中，器件信息表的结构如表 11-3 所示。

表 11-3 器件信息表

属 性	数据类型	数据库对应属性名
设备号	文本	C_Manufacturer
厂家	文本	C_InsNum
年份	文本	D_Year
批次	整型	I_Batch

续表

属 性	数据类型	数据库对应属性名
设备型号	文本	C_InsMod
设备类型	文本	C_InsType
设备代号	文本	C_InsCode
主要参数	文本	C_Parameter
设备ID	文本	C_InsID

3 · 测试指标表

测试系统将测试的设备指标存入计算机中，以便对数据进行后续处理。针对每种测试指标，程序都规范地定义了指标的类别、号码标识、测试的指标范围等。测试的指标号、名称和范围值存入测试指标表中，测试指标表结构如表 11-4 所示。

表 11-4 测试指标表

属 性	数据类型	数据库对应属性名	备 注
测试指标号	整型	I_MeaNum	Primary key
测试指标名	文本	C_TarNam	
优先级别	整型	I_Priority	
指标上限	实数单精度浮点型	F_TargetUpLimit	
指标下限	实数单精度浮点型	F_TargetLowLimit	
指标版本	整型	I_Edition	

4 · 测试指标配置表

本自动测试系统需要对信号源和频谱仪进行参数的配置后才能执行相应指标的测试任务，为了实现自动配置的功能，将每种指标测试中需要的配置参数事先存入计算机的数据库中，在需要时调用出来设置仪表。根据需要定义配置参数类别和索引号，并存入测试指标配置表中，此表结构如表 11-5 所示。

表 11-5 测试指标配置表

属 性	数据类型	数据库对应属性名
测试号	整型	ID
测试项目	整型	N_MeasureType
开始频率	实数单精度浮点型	F_FreqStart
终止频率	实数单精度浮点型	F_FreqStop
中心频率	实数单精度浮点型	F_FreqCenter
频率带宽	实数单精度浮点型	F_FreqSpan
分辨率带宽	实数单精度浮点型	F_ResBW
视频带宽	实数单精度浮点型	F_VideoBW
扫描时间	实数单精度浮点型	F_SweepTime

续表

属 性	数据类型	数据库对应属性名
参考电平	实数单精度浮点型	F_RefLevel
衰减	实数单精度浮点型	F_Attenuaton
偏置参考电平	实数单精度浮点型	F_RefLvIOffset
积分带宽	实数单精度浮点型	F_IntegBW
信道功率带宽	实数单精度浮点型	F_ChanPwrSpan
参考带宽	实数单精度浮点型	F_RefBW
频率偏置A	实数单精度浮点型	F_OffsetFreqA
频率偏置B	实数单精度浮点型	F_OffsetFreqB

5 · 器件测试表

器件测试表保存的是仪表的测试结果，根据事先定义好的指标号将这些测试结果存入对应的记录中，以便后面进行数据统计和分析，器件测试表结构如表 11-6 所示。

表 11-6 器件测试表

属 性	数据类型	数据库对应属性名	备 注
设备ID	文本	C_InsID	Primary key
测试指标号	整型	I_MeaNum	
测试结果	单精度浮点型	F_Result	
测试时间	时间	T_DateTime	
测试人员	文本	C_MeaMan	

6 · 测试指标与测试指标号

针对无线设备的特点，本例中定义了待测试的测试指标和指标号码，这些属性描述了测试指标的上下限，也就是值的范围，通过这个范围程序可以知道测得的数据是否有效或合格，可以自动进行指标的判断，还可以判断是否需要重新测试等。

测试指标与测试指标号表的结构如表 11-7 所示。

表 11-7 测试指标与测试指标号表

测试指标名	测试指标号	上 限	下 限	优 先 级
工作频率	1	1910	1793	1
标称输出功率 2C2W（下行）	2	38	36	1
标称输出功率 2C1W（下行）	271	35	33	1
标称输出功率 2C0.5W（下行）	272	32	30	1
输入功率（下行）	3	10	0	3
饱和输出 2C2W（下行）	4	9999	42	3
饱和输出 2C1W（下行）	281	9999	39	3
饱和输出 2C0.5W（下行）	282	9999	36	3

续表

测试指标名	测试指标号	上 限	下 限	优 先 级
增益可调范围 15（下行）	5	16	14	3
增益可调范围 23（下行）	6	6	8	3
增益可调范围 30（下行）	7	0	0	3
增益调节步长（下行）	8			2
干线放大器具有自动增益控制（ALC）功能（下行）	9	2	-1	3
占用带宽（下行）	10	288	-9999	3
RMS 矢量误差（下行）	11	12.50%	-9999	3
峰值矢量误差（下行）	12	25%	-9999	3
带外余数（1920~1980 MHz）（下行）	13	-71	-9999	1
带外余数（1710~1750 MHz）（下行）	14	-60	-9999	2
带外余数（885~960 MHz）（下行）	15	-47	-9999	3
带外余数（825~880 MHz）（下行）	16	-47	-9999	3
带外余数（9 KHz~1GHz）（下行）	17	-36	-9999	3
带外余数（1 GHz~12.75 GHz）（下行）	17	-30	-9999	3
邻道功率泄漏（-600 k）（下行）	19	-31	-9999	2
邻道功率泄漏（+600 k）（下行）	20	-31	-9999	2
邻道功率泄漏（-900 k）（下行）	21	-36	-9999	2
邻道功率泄漏（+900 k）（下行）	22	-36	-9999	2
带内互调衰减（下行）	23	-44	-9999	1
带外抑制（1.710 GHz~1.883 GHz）（下行）	24	9999	45	1
带外抑制（1.92 GHz~1.98 GHz）（下行）	25	9999	45	1

驻波比（下行）	26	1.5	-9999	3
带内平坦度（下行）	27	1	-9999	3
传输时延（下行）	28	1	-9999	3
输入输出阻抗（下行）	291			3
温度特性（工作环境温度：-10℃~55℃）（下行）	292			3
噪声系数（上行）	29	4	-9999	3
标称输出功率（上行）	30	0	-9999	3
输入功率（上行）	31	暂缺	暂缺	3
增益可调范围 15（上行）	32	16	14	3
增益可调范围 23（上行）	33	6	8	3
增益可调范围 30（上行）	34	0	0	3
增益调节步长（上行）	35			3
饱和输出（上行）	36			3
带内平坦度（上行）	38	1	-9999	3
传输时延（上行）	39	1	-9999	3

续表

测试指标名	测试指标号	上 限	下 限	优 先 级
驻波比（上行）	37	1.5	-9999	3
输入输出阻抗（上行）	293			3
温度特性（工作环境温度：-10℃~55℃）（上行）	294			3

7 · 数据库模块实现

鉴于 LabVIEW 数据库操作的不方便和工具包需要付费使用等问题，本系统的数据库模块采用混合编程的方式实现，利用 Microsoft Visual C++ 6.0 开发通过 AD O 与 Office Access 和 SQL Sever 数据库操作的 DLL，然后在 LabVIEW 中调用。

（1）利用 C++生成 DLL

此方法适合那些对 C++语言熟悉的开发人员。对于熟练使用 LabVIEW 或已经购买过 NI 公司相关数据库产品的开发人员可直接采用 LabVIEW 操控数据库的方法，具体方法见第 10 章。

本例提出了利用文本语言作为中介进行数据库操作的混合编程方法，利用 C++编写的 DLL 可以在 LabVIEW 中调用，在 LabVIEW 框图中输入标准的 SQL 语句即可灵活地对数据库进行操作。

Microsoft Visual C++6.0 实现 DLL 导出的代码如下：

```
extern "C"  __declspec(dllexport) bool SaveResultToDB(char cSql[])
{
    CString csSql;
    csSql=(CString)cSql;
    CDBModule QM;
    QM.QConnect();
```

```
QM.QSaveTestResult(csSql);  
QM.QClose();  
return true;  
}
```

（2）LabVIEW 调用 DLL

LabVIEW 中实现调用 DLL 方法如图 11-58 所示。

在图 11-58 所示的“Libraries & Executables”模板中选中“Call Library Function”，并拖到后面板中，会弹出如图 11-59 所示的对话框。在此对话框中选择要调用的 DLL 文件，并指定需要使用的函数。

8 · 数据报表

对于自动测试软件采集的数据可通过标准报表提供给测试工程师，无线设备标准测试结果格式如表 11-8 所示。

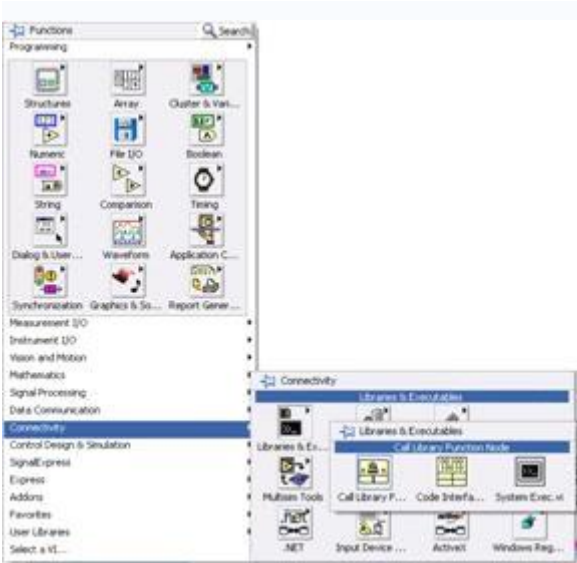


图 11-58 在 LabVIEW 中调用 DLL



图 11-59 LabVIEW 调用的 DLL 参数

表 11-8 测试指标标准表

测试表格			
主要参数		测试厂家	
		设备型号	
		设备号（S/N）	
批 次		测试人员	
		测试时间	
链 路	序 号	项目和内容	指标要求
下行	1	工作频率	1793 MHz~1910 MHz
	2	标称输出功率	0.5W/2 载波、4 载波（27dBm）
链 路	序 号	项目和内容	指标要求
下行	3		1W/1 载波、2 载波、4 载波（30dBm）
	4		2W/1 载波、2 载波（33dBm）
	5	输入功率	每载波输入电平在 0~+10dBm 内变化，可使干放达到标称输出
	6	饱和输出（1dB 压缩点输出功率）	饱和输出 2C2W≥42

	7		饱和输出 2C1W≥39
	8		饱和输出 2C0.5W≥36
	9	增益可调范围	增益在 15 dB
	10		增益在 23 dB
	11		增益在 30 dB
	12	增益调节步长	增益调节步长: 1dB; 增益控制精度: ±1dB
	13	干线放大器应具有自动增益控制 (ALC) 功能	达到标称输出功率时, 当输入信号电平增加 10dB, 输出电平的变化应保持在 2dB 之内
	14	占用带宽	≤288 kHz
	15	矢量幅度误差 (EVM) 指标 (下行)	RMS 矢量误差: ≤12.5%
	16		峰值矢量误差: ≤25%
	17	带外杂散 (下行)	≤-60 dBm/3.84 MHz (1920~1980 MHz) 或 ≤-71 dBm/300kHz (1920~1980MHz)
	18		≤-60 dBm/100 kHz (1710~1750 MHz) ;
	19		≤-47 dBm/100 kHz (885~960 MHz) , 与GSM900 共用空内分布
	20		≤-47 dBm/100 kHz (825~880 MHz) , 与CDMA800 共用空内分布
	21		≤-36 dBm/30 KHz (9 KHz~1 GHz)
	22		≤-30 dBm/30 KHz (1GHz~12.75 GHz)
	23	邻道功率漏漏 (PATTEN 模式)	≤-31dBm (-600k)
	24		≤-31dBm (+600k)
	25		≤-36 dBm (-900k)
	26		≤-36 dBm (+900k)
	27	带内互调衰减	≤-44 dBc (标称功率)
	28	带外抑制 (扫频范围 1773~1930 MHz)	≥45 dBc (1.710GHz~1.883 GHz/30k)
	29		≥45 dBc (1.92GHz~1.98 GHz/30k)
	30	驻波比	≤1.5
	31	带内平坦度	≤1dB
	32	传输时延	<1us
	33	输入输出阻抗	干线放大器输入输出阻抗为 50Ω
	34	温度特性 (工作环境温度: -10℃~55℃)	在整个温度范围内, 满功率输出情况下连续工作 4 小时后, 测试功率稳定度: 标称值+2dB、带外抑制: 标称值+5dB
上行	35	标称输出功率	≤0
	36	输入功率	

链 路	序 号	项 目 和 内 容	指 标 要 求
上行	37	增益可调范围	增益在 15 dB
	38		增益在 23 dB
	39		增益在 30 dB
	40	增益调节步长	增益调节步长: 1dB; 增益控制精度: ±1dB
	41	饱和输出 (上行)	
	42	驻波比	≤1.5
	43	带内平坦度	≤1dB
	44	传输时延	<1us
	45	噪声系数 (上行)	<4dB
	46	输入输出阻抗	干线放大器输入输出阻抗为 50Ω

	47	温度特性（工作环境温度：-10℃~55℃）	在整个温度范围内，满功率输出情况下连续工作 4 小时后，测试功率稳定度：标称值+2dB，带外抑制：标称值+5dB
--	----	-----------------------	--

11.4.7 界面设计

前台采集系统界面主要分为 5 个部分，分别是系统管理模块、E4440A/E4433B/E4438 仪器控制模块、E5071B 仪器控制模块、N8973A 仪器控制模块和 MT8801C 仪器控制模块。

1 · 系统管理

系统管理模块主要是对所连接控制的仪表进行管理，通过此模块可以测试仪器是否处于连接状态，也可对仪器进行复位和自检等操作。

系统管理模块的界面如图 11-60 所示。



图 11-60 系统管理模块界面

2 · E4440A、E4433B 和 E4438 仪器控制

本模块的功能是对 E4440A、E4433B、E4438 的三台仪器进行控制，通过对信号源和频谱仪的控制来测试无线设备的相关指标。E4440A、E4433B 和 E4438 仪器控制模块界面如图 11-61 所示。

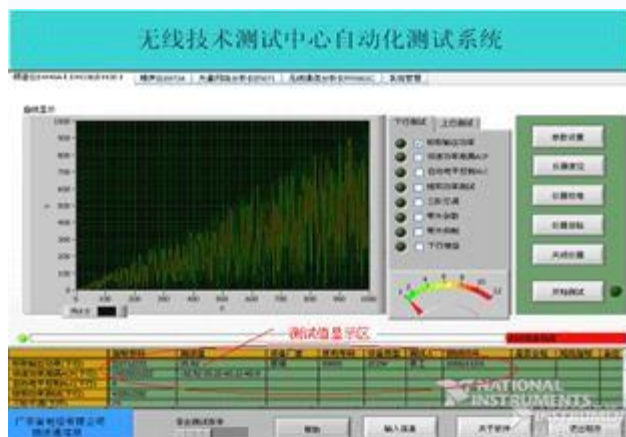


图 11-61 E4440A、E4433B 和 E4438 仪器控制模块界面

3 · E5071B 仪器控制

本模块的功能是对矢量网络分析仪 E5071B 进行控制，通过对 E5071B 的控制来测试无线设备的相关指标，主要是驻波比、传输时延和平坦度等。

E5071B 仪器控制模块的界面如图 11-62 所示。



图 11-62 E5071B 仪器控制模块的界面

4 · N8973A 仪器控制

本模块的功能是对噪声分析仪 N8973A 进行控制，通过对 N8973A 的控制来测试无线设备的相关指标，主要是噪声系数等。N8973A 仪器控制模块的界面如图 11-63 所示。



图 11-63 N8973A 仪器控制模块界面

5 · MT8801C 仪器控制

本模块的功能是对无线通信综合分析仪 MT8801C 进行控制，通过对 MT8801C 的控制来测试无线设备的相关指标，主要是 RMS 和占用带宽等。

MT8801C 仪器控制模块界面如图 11-64 所示。



图 11-64 MT8801C 仪器控制模块界面

11.4.8 指标测试

本自动测试系统能够对仪表进行自动控制，仪表分别完成无线通信设备（包括上、下行）的信道功率、邻道功率泄漏 ACP、自动电平控制 ALC、三阶交调、杂散发射、带外抑制、饱和功率、噪声系数、驻波比、传输时延、平坦度、RMS、峰值误差和占用带宽等指标的测试。

例如 CDMA 直放站测试项目所需的仪表设备如表 11-9 所示。

表 11-9 CDMA 直放站测试项目及对应仪表型号表

测试项目	仪表 1		仪表 2	
	仪 表	型号 (厂家)	仪 表	型号 (厂家)
输入/输出功率	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
频率误差	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
自动增益控制 (ALC) 功能	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
最大增益	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
增益调节步长	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
互调衰减	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
带外抑制	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
杂散发射	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
波形质量因素	频谱仪	FSP (Rohde & Schwarz)	综合信号发生器	Agilent E4433B
带内波动	矢量分析仪	Agilent E5071C		
驻波比 (VSWR)	矢量分析仪	Agilent E5071C		
传输时延	矢量分析仪	Agilent E5071C		
噪声系数	噪声分析仪	Agilent N8973A		
输出光功率				
接收光功率				

由于不同仪表和上/下行测试的电缆连接方式不同，所以自动测试应分成若干个阶段，每个阶段完成接线方式相同的指标测试。

无线设备的测试流程图如图 11-65 所示。

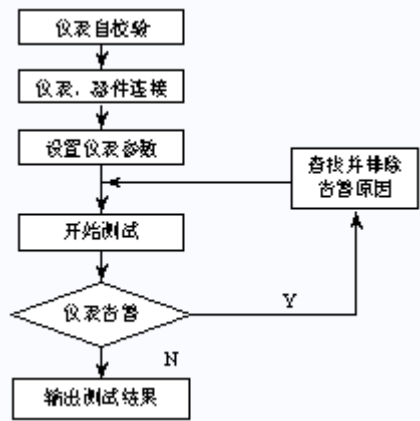


图 11-65 测试流程图

测试新的指标之前，本模块根据操作用户选择的测试产品和项目，从参数配置文件中调出原先已配置或新配置的参数，然后自动完成各个项目的测试。软件在自动测试过程中能够判断测试参数与测试数据是否异常。

干放测试的技术指标中，测试标称功率是最重要的环节，因为此后的多项指标测试均在此状态下回退 3dB 情况下进行测试。图 11-66 为干放输出曲线图。

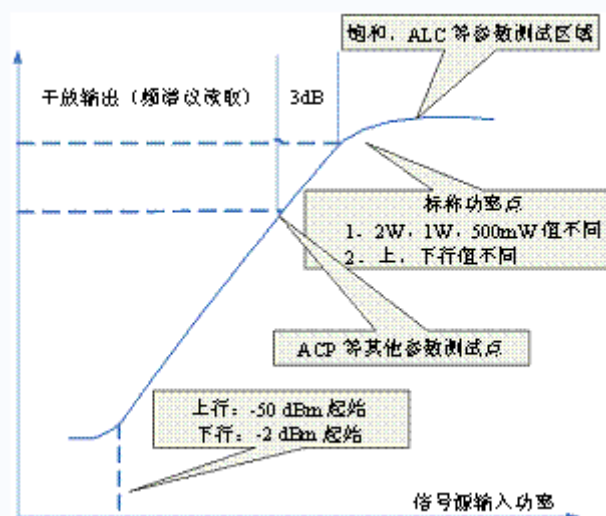


图 11-66 系统功能图

下面是图 11-66 中所指出的标称功率点测试的算法实现步骤。

- ① 逐级递加法。逐步递加信号源功率，一般步长为 1dB。注意：步长大，可能越过饱和点；步长短，测试速度慢。
- ② 逐级递加法修正。针对步骤 1（可理解为粗调），当越过饱和点时，进行微调，变步长为 0.1dB，在 1dB 粗调误差范围内进行递减。步长可根据测试精度要求适当选择。
- ③ 针对步骤 ②，在线性区测试过程中，如果存在野值，程序退出测试可能出现错误结果，此种情况下进行全程采集，然后利用中值滤波法剔除野值，再采用 3 点（也可用 5 点）平滑算法处理，得到平滑曲线，最后利用梯度法找到标称功率点（也可参考使用 sobel 算子或拉普拉斯算子法进行分析）。
- ④ 对于步骤 ③ 中所求得的标称功率点，如果满足精度要求，可不进行步骤 4。如果要在节省测试时间的情况下，尽可能地提高测试精度，需要利用插值法得到拟合曲线，然后运用边界搜索等方法对此数据进行分析，此时根据测试精度选取适当的域值，就能准确定位到干放标称功率点。

图 11-67 为三阶交调指标测试算法示意图。一般设备测试规范要求，A 点和 B 点中功率比较低的信号与 C 点和 D 点中功率比较低的信号功率之差一定要大于某

值，否则认为设备不合格。一般要求功率之差大于 44dBc，测试过程中，计算机控制 E4433 输出 1900 MHz 信号，E4438 输出 1902 MHz 信号，同时计算机通过收敛算法在尽量使 A 点和 B 点信号功率相等且达到标称输出状态下（此时需要根据给定的误差来确定收敛速度和精度，本系统中取 0.05），此时在频谱仪 E4440 中会发现最大的干扰信号三阶交调信号 C 和 D。最后计算机控制频谱仪 E4440 读取 Min(A, B) 和 Max(C, D)，并计算两者差值，将结果存入后台数据库。

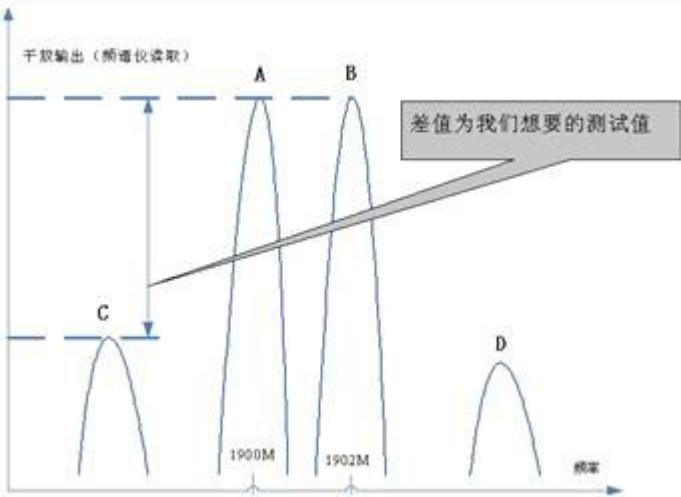


图 11-67 三阶交调指标测试算法示意图

表 11-10 列出了 2 项 2C2W PHS 干放采用自动测试系统前后的测试数据。

表 11-10 PHS 干放测试数据对比

测试目标	手动测试		自动测试
标称输出	35.5 dBm		35.62 dBm
邻道泄露	-600 k	-46.5 dBm	-46.62 dBm
	+600 k	-46.8 dBm	-46.91 dBm
	-900 k	-48.5 dBm	-48.32 dBm
	+900 k	-48.4 dBm	-48.62 dBm

通过表 11-10 可以看出，手动测试值的精度比自动测试低，主要是因为人工读取数据的不确定性比较大，同时仪表数据读取时数字会跳动，不能保证读取高精度的测试数据。通过自动测试系统可以将测试结果进行多次平均（本系统取 30 次平均），同时根据需要截取小数点位数（本系统取 2 位有效数字）。这样保证了自动测试精度，同时测试标准也得到了统一。

以上是干放标称功率点和三阶交调确定方法，带外杂散等指标测试算法，在此不叙述。

11.4.9 实例小结

本例给出了无线设备自动测试的整体实现方法和思路，也给出了测试指标中最重要标称功率点和三阶交调的测试算法，由于篇幅所限，各模块的具体实现过程做了省略，同时未具体叙述带外杂散和其他参数等技术指标的自动测试实现算法。通过实际测试表明，该系统明显提高了测试效率。

本系统具有如下特点：

（1）操作的简易性

前台数据采集系统需要用户配置的各种测试参数没有涉及设备底层，用户只要设置测试仪表对每项指标测试时的参数即可，而且常用的数据库参数配置已经能满足 PHS 干放的自动测试任务，不需要任何更改就可进行测试。这种操作的简易性降低了对现场测试人员的专业水平的要求，可以节省人力资源。在这种情况下，可以在测试前把设备接好，设置好参数，然后让计算机自动测试，在没有专业人员的情况下完成测试任务。

（2）系统的便携性

如果想节省成本，可以不采用 PXI 平台，测控前端只需要一台笔记本电脑或台式机、一个 GPIB 卡和与所需仪器测量数目相同的 GPIB 电缆等，并不需要其他专用测试仪表就可以完成所有的测试工作。

（3）数据的准确性

本系统由于采用了计算机自动控制以及自动读取数据，因此可以避免传统用人工抄写仪器测量数据而带来的人为误差。

（4）数据采集的高效性

本系统采用 GPIB 电缆连接仪器，可以同时接入多台仪器进行不同项目指标的测试，并且某些指标的测试过程需要较长时间，这时候测试工程师就可以从事其他工作，等到测试完毕后软件会自动提示下一步如何操作，这样明显提高了工作效率。