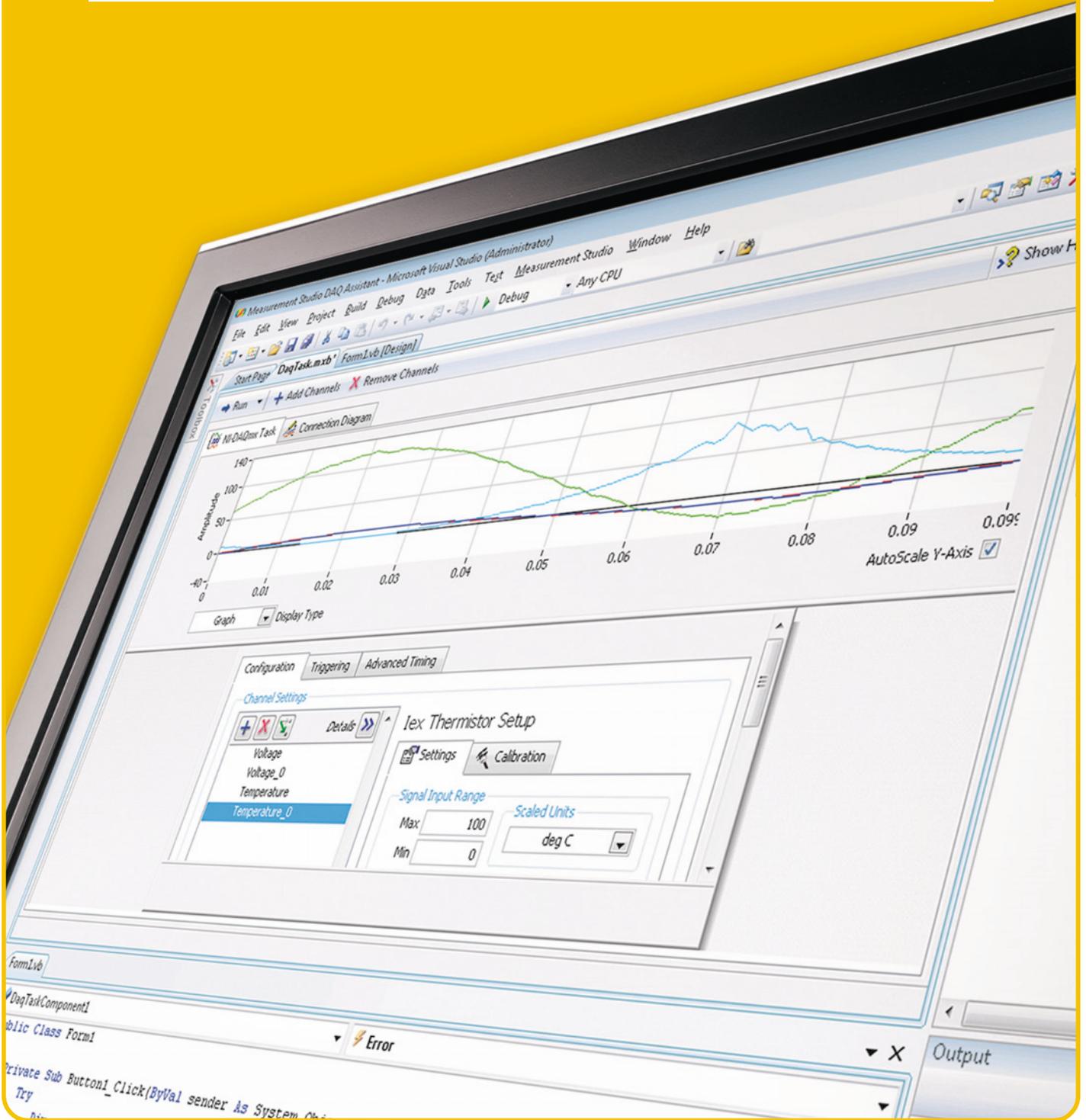


数据采集编程指南 中篇



目 目 录 录

模拟 I/O 与数字 I/O	1-12
计数器应用 (上)	13-18
计数器应用 (下)	18-23
定时与触发	23-30

简介

本期节目介绍测试测量的接线方式，如何使用 NI 数据采集板卡及底层的 DAQmx VI 来完成模拟输入输出以及数字输入输出功能。

在介绍具体的模拟 I/O，数字 I/O 之前，首先介绍接线方式。

接线方式

对于不同的信号，需要采用不同的接线方式，如图 4-1 所示。对于接地信号和浮地信号，不同的接线方式将带来不同的测量效果。为了得到正确的测量结果，需要使用正确的连线方式。

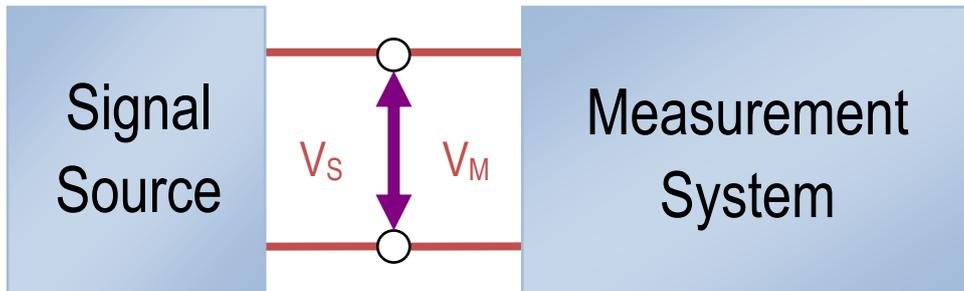


图 4-1 信号源与测量系统的接线

确定正确连线方式的步骤分为两步：

1. 首先要确定信号源种类
2. 其次来选择测量系统提供的合适的终端模式

NI 数采卡上提供了三种不同的终端模式：

- 差分模式：在一个差分测量系统中，仪表放大器的任何一个输入都不是以系统地作为参考的，如图 4-2 所示，AIGND 引脚以及放大器本身是以系统地作为参考的，但两个输入端均不以地作为参考。这里需要注意的是，当我们使用差分方式时，对于一个输入信号 需要使用两个模拟输入通道，于是整个可用通道数就减半了，对于一个 16 通道的数据采集设备，处于差分模式下的时候，只能采集 8 路输入信号了，输入信号的配对规则如图所示，ACH(N)与 ACH(N+8) 组成一对差分输入通道。

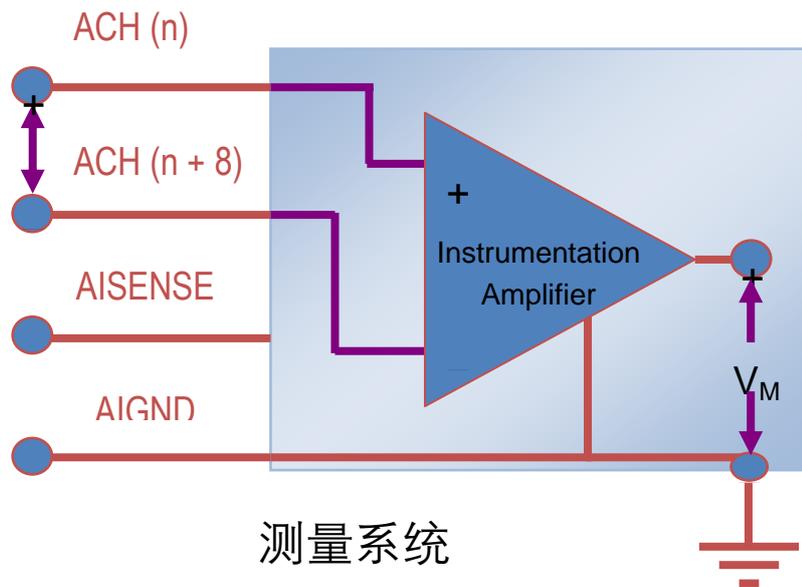


图 4-2 差分模式

举例来说，如果我们想要在通道 5 上测量模拟输入，那么需要将信号的正端连接到 ACH5 并将负端连接到 ACH 13 上，如图 4-3 所示。既然使用差分的模式会使可用通道数减半，为什么我们有时仍需要使用这样的测量方式呢？答案是为了获得更好的测量效果。因为差分模式可以使得放大器有效地抑制共模电压，以及任何与信号混杂在一起的共模形式噪声，有效提高测量质量。

ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7

图 4-3 差分输入通道对

• 参考单端模式(RSE):

一个参考单端测量系统以系统地作为参考，信号源的负端是被连接到 AIGND 上的，也就是说它是被连到系统地上。这种连接的方式使得我们在测量时，对于每个信号只需

要使用一个模拟输入通道，所以，一个 16 通道的数据采集设备在使用 RSE 模式时，可以测量 16 路信号。如果我们想要在模拟输入通道 10 上测量一个信号，那么只需要将信号的正端连接到 ACH10，负端连接到 AIGND 上。如图 4-4 所示。

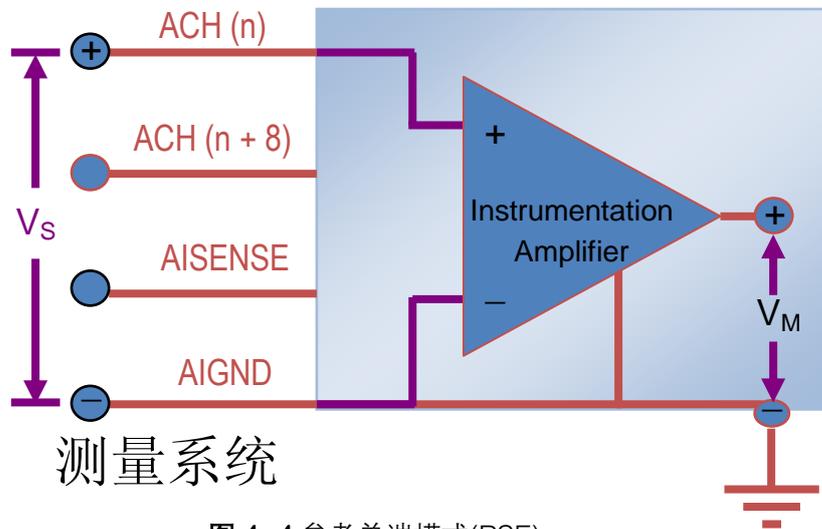


图 4-4 参考单端模式(RSE)

此外，我们的板卡上提供了许多 AIGND 引脚来防止由于输入连线搭接所造成的信号间串扰，如图 4-5 所示。尽管 RSE 的连接模式能够保证通道数的使用效率，但是它无法抑制共模电压。在某些应用当中，过大的共模电压会造成测量误差甚至毁坏您的设备。

ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7

图 4-5 多个 AIGND 防止由于输入连线搭接所造成的信号间串扰

- 非参考单端模式(NRSE):

NI 的数据采集板卡上还提供了一种不同于 RSE 参考单端的模式，我们称它为 NRSE，非参考单端模式，在 NRSE 模式下，所有的测量同 RSE 相类似都参考同一个参考点，但与 RSE 模式不同的是该参考点的电压值可以调整 and 变化。如图 4-6 所示，信号的

负端被连接到 AISENSE 引脚上，而 AISENSE 并不是以地作为参考的。于是 AISENSE 上的电压是浮地的。

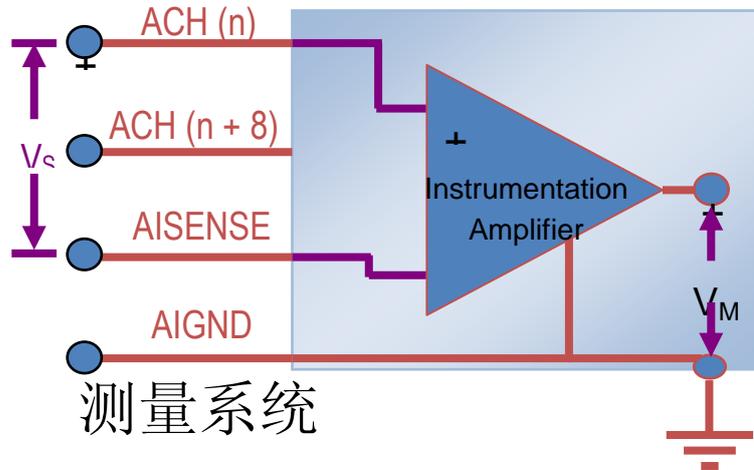


图 4-6 非参考单端模式(NRSE)

图 4-7 的引脚图中给出了板上只有一个 AISENSE 引脚可供信号连接，因为我们需要确认每个信号均使用了同样的参考点。与 RSE 模式相类的是，NRSE 模式最大程度地保留了可用的模拟通道数，但同样无法抑制共模电压。那为什么要使用 NRSE 呢？我们之后会看到需要根据信号源的不同来选择不同的终端模式。

ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7

图 4-7 AISENSE 引脚图

介绍了 NI 产品提供的三种不同终端模式后，我们根据不同的信号源来分析一下应该使用哪一种接线方式。对于接地信号源来说，图 4-8 中列出了三种模式的优点和缺点。

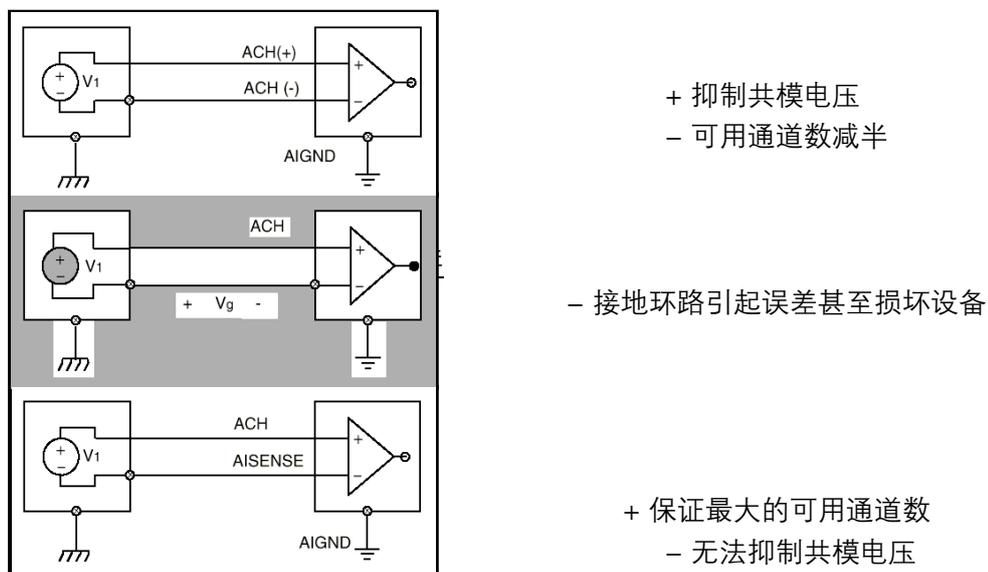


图 4-8 对于接地信号 三种模式的优点和缺点

1. 差分模式 (Differential): 虽然该模式会使可用通道数减半, 但是它具有非常好的共模电压和共模噪声抑制能力, 是不错的选择。
2. 其次是参考单端(RSE), 对于接地信号, 参考单端是不推荐使用的终端模式, 因为接地环路的电势差会造成测量误差, 并将交流噪声以及直流偏移量引入到测量系统当中。除此之外, 当信号源正端不小心接到 RSE 测量系统的 AIGND 上时, 还会造成信号源短路以至于损坏。
3. 第三是非参考单端(NRSE), 由于测量系统的负端以 AISENSE 为参考而不是直接以地作为参考, 对于接地信号, NRSE 模式可以保证最大的可用通道数, 然而它无法像差分模式那样抑制共模信号。

也就是说对于接地信号的情况, 我们只有差分和 NRSE 两种模式可选, 如果您的剩余可用通道数足够多的话, 首先推荐使用差分模式, 如果您想尽可能多地使用模拟输入通道, 那么可以选择 NRSE 模式。

对于浮地信号, 三种终端模式均可以选择, 他们的优缺点如图 4-9 所示, 首选推荐差分模式, 在牺牲了通道数的情况下能够提高测量的质量。其次可以使用 RSE 模式, 因为该方式下, 不需要连接偏置电阻。最后才选择 NRSE 模式。

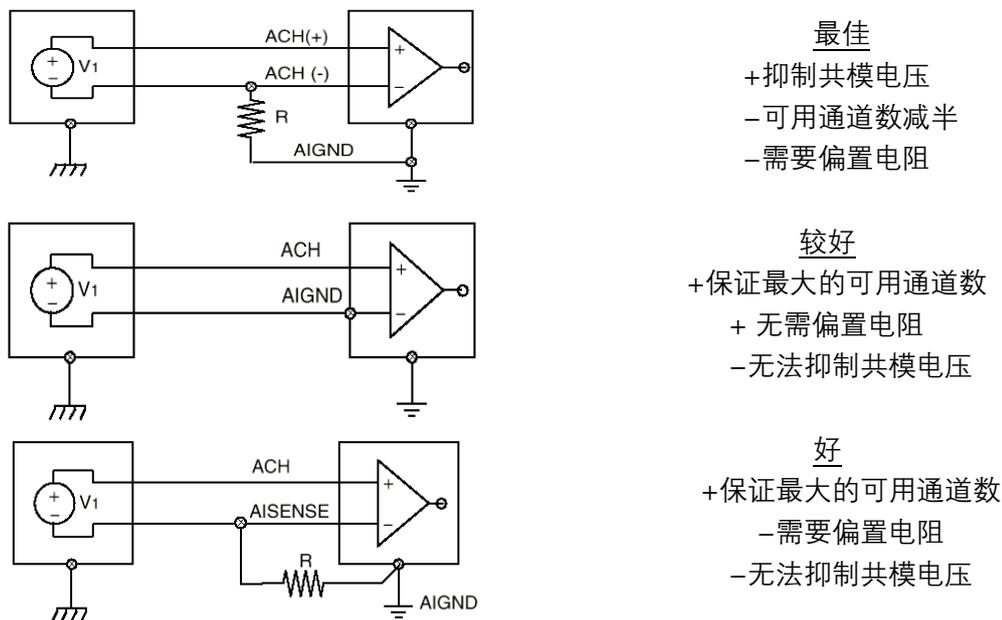


图 4-9 对于浮地信号 三种模式的优点和缺点

在差分 and NRSE 模式下，需要为仪表放大器连接对地回路的偏置电阻，对于 DC 信号只需要连接负端到地，而对于 AC 信号则需要分别在信号输入端各连接一个偏置电阻，偏置电阻的大小取决于信号源的阻抗大小，典型值在 10 k 到 100 k 欧姆之间。

二. 模拟输入

使用 DAQmx 底层 VI 进行数据采集：

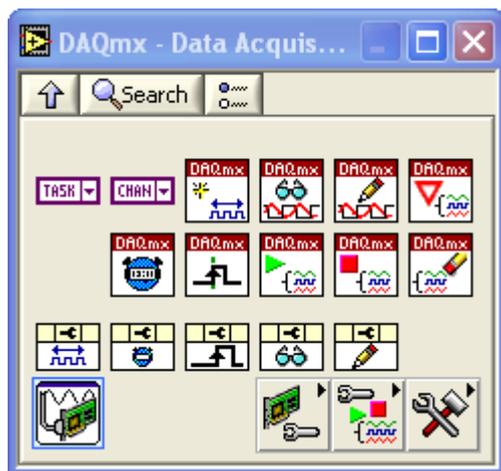


图 4-10 DAQmx 数据采集子选板

我们看到的所有的 DAQmx 底层驱动 VI 都能在测量 I/O 选版下的 DAQmx 子选版下找到包括了 I/O 端口，创建通道，读取，写入，定时，触发等等。如图 4-10 所示。

对于通道， I/O， 定时， 触发等底层设置都有各自的属性节点， 在您的数据采集编程当中所需要的绝大多数功能组件都位于函数图标下， 由于这些函数都是多态的， 普通 DAQmx 函数的接线端无法一次性容纳所有可能的输入输出设置。 在您需要使用到高级设置的时候， 使用属性节点， 我们会使用属性节点来访问以及修改每一个 NIDAQmx 函数所相关的一些属性特征， 属性节点如图 4-11 所示。

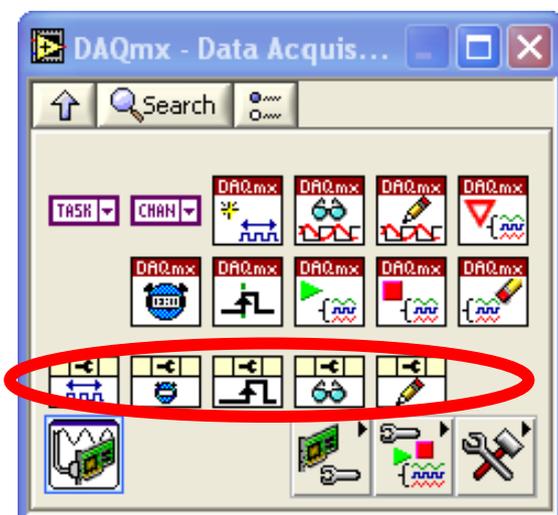


图 4-11 属性节点

下面我们依次了解一下各个底层 DAQmx VI 的详细功能

1. 创建虚拟通道函数：

通过给出所需的目标通道名称以及物理通道连接， 用来在程序中创建一个通道。

图 4-12 中选择了创建一个热电偶输入通道。

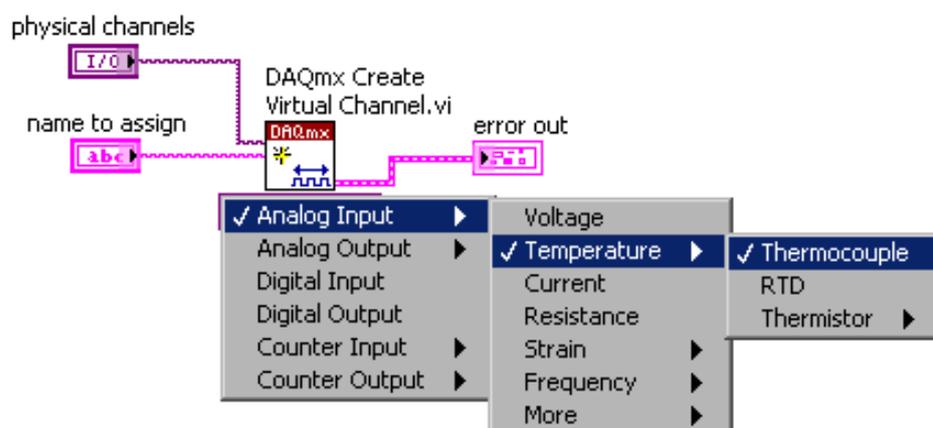


图 4-12 创建虚拟通道

您在 MAX 当中创建通道时进行的相同的设置在这个函数中均会得到设置。

当程序操作员需要经常更换物理通道连接设置而非其他诸如终端配置或自定义缩放设置的时候， 这个创建虚拟通道 VI 就非常有用了。

物理通道下拉菜单被用来指定 DAQ 板卡的设备号以及实际连接信号的物理通道。
通道属性节点是创建虚拟通道函数的功能扩展，允许您在程序当中动态改变虚拟通道的设置。举例来说，对于一组测试我们可用通过它来对一个通道设置一个自定义缩放之后在对另一组进行测试时可以通过属性节点改变自定义缩放的值。

2. 定时设定 VI

DAQmx 定时 VI 配置了任务、通道的采样定时以及采样模式，并在必要时自动创建相应的缓存。如图 4-13 所示。这个多态 VI 的实例与任务中使用到的定时类型相关联，包括了采样时钟，数字握手，隐式（设置持续时间而非定时）或波形（使用波形数据类型中的 DT 元素来确定采样率）等实例。类似的定时属性节点允许您进行高级的定时属性配置。

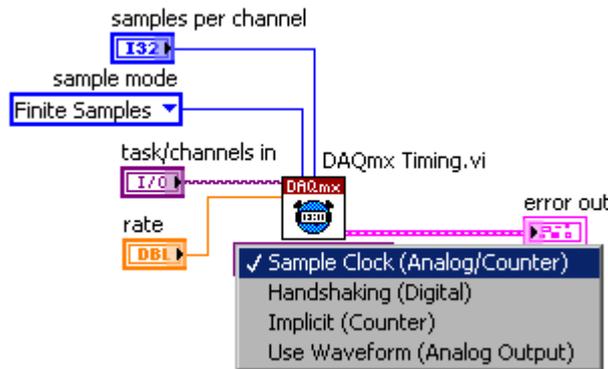


图 4-13 DAQmx 定时 VI

3. DAQmx 触发设定 VI

DAQmx 触发 VI 配置了任务、通道的触发设置。如图 4-14 所示。这个多态 VI 的实例包括了触发类型的设置，数字边沿开始触发，模拟边沿开始触发，模拟窗开始触发，数字边沿参考触发，模拟边沿参考触发或是模拟窗口参考触发等等。同样的我们会使用触发属性节点来配置更多高级的触发设置

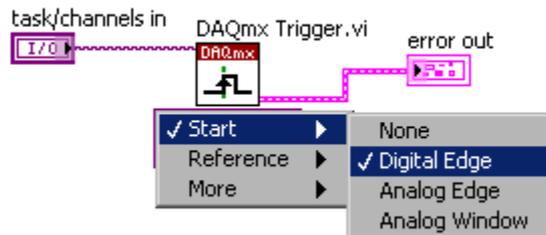


图 4-14 触发设定 VI

4. DAQmx 读取 VI

DAQmx 读取 VI 从特定的任务或者通道当中读取数据，如图 4-15 所示，这个 VI 的多态实例会指出 VI 所返回的数据类型，包括一次读取一个单点采样还是读取多点采样，以及从单通道读取还是从多通道中读取数据
其相应的属性节点可以设置 偏置 波形属性 以及获取当前可用采样数等数据

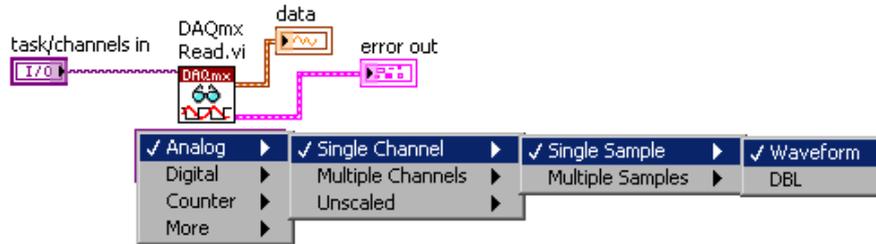


图 4-15 DAQmx 读取 VI

模拟输入

图 4-16 程序完成了模拟信号的连续采集, 与上一讲中使用 DAQ 助手快速 VI 不同, 这里我们使用的都是 DAQmx 的底层驱动 VI,

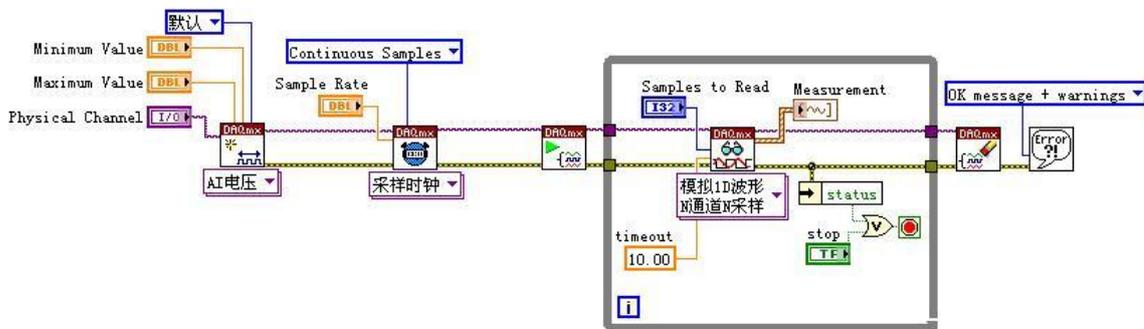


图 4-16 模拟信号的连续采集

连续采集的流程图如图 4-17 所示, 首先创建虚拟通道, 设置缓存大小, 设置定时, (必要时可以设置触发), 开始任务, 开始读取。由于我们是连续采集信号, 于是我们需要连续地读取采集到的信号。因此我们将 DAQmx 读取 VI 放置在循环当中, 一旦有错误发生或者用户在前面板上手动停止采集时 程序会跳出 while 循环。

之后使用 DAQmx 停止任务来释放相应的资源并进行简单错误处理。

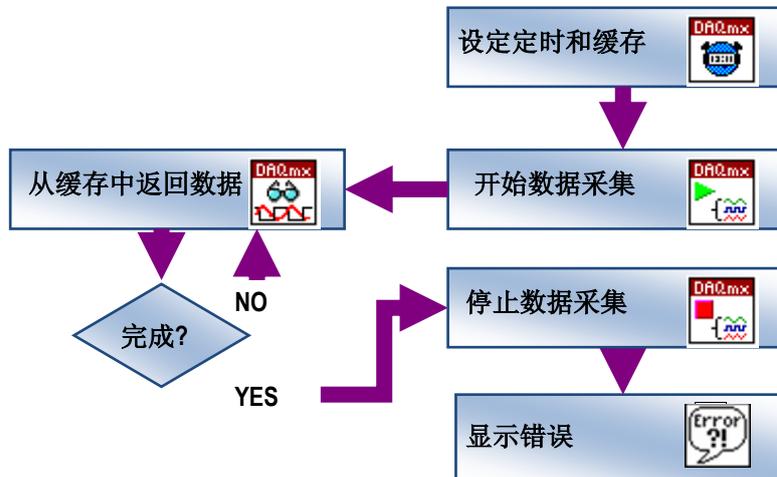


图 4-17 模拟信号的连续采集流程

在连续采集当中，我们会使用一个环形缓冲区，这个缓冲区的大小由 DAQmx 定时 VI 中的 SAMPLES PER CHANNEL 每通道采样来确定。

如果该输入端未进行连接或者设置的数值过小，那么 NI DAQmx 驱动会根据当前的采样率来分配相应大小的缓冲区，其具体的映射关系可以参考 DAQmx 帮助。

同时，在 WHILE 循环中 DAQmx 读取的输入参数 SAMPLES TO READ (每通道采样数) 表示了，每次循环，我们从缓冲中读取多少个点数的数据。

为了防止缓冲区溢出，我们必须保证读取的速率足够快。一般我们建议 SAMPLES TO READ 的值为 PC 缓冲大小的 1/4。

三. 模拟输出

对于 AO，我们需要知道输出波形的频率，输出波形的频率取决于两个因素，更新率以及缓冲中波形的周期数。我们可以用以下等式来计算我们输出信号的频率：

$$\text{信号频率} = \text{周期数} \times \text{更新率} \div \text{缓冲中的点数}$$

举例来说，我们有一个 1000 点的缓冲放置了一个周期的波形，如果要以 1kHz 的更新率来产生信号的话，那么，1 个周期乘以每秒一千个点更新率除以总共一千个点 等于 1HZ。如果我们使用 2 倍的更新率。那么，一个周期乘以每秒 2000 个点除以总共 1000 个点，得到 2HZ 的输出。

如果我们在缓冲中放入两个周期的波形，那么，两个周期 乘以 1000 个点每秒的更新率除以总共 1000 个点，得到输出频率为 2HZ。

也就是说 我们可以通过增加更新率 或者 缓冲中的周期数来 提高输出信号的频率。DAQmx 中产生连续模拟波形的流程如图 4-18 所示

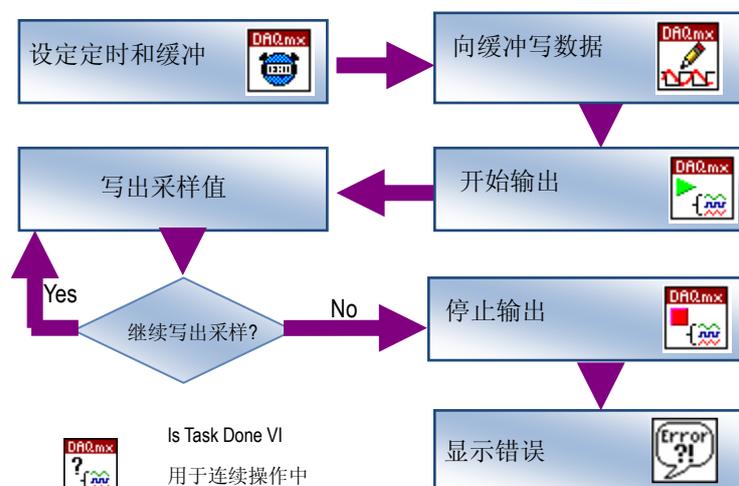


图 4-18 模拟信号的连续产生流程

图 4-19 中的例子使用 DAQmx 定时 VI 设定一个给定的 44100S/S 输出更新率，并在 while 循环中使用 DAQmx 任务完成 VI 来检测任何可能出现的错误。

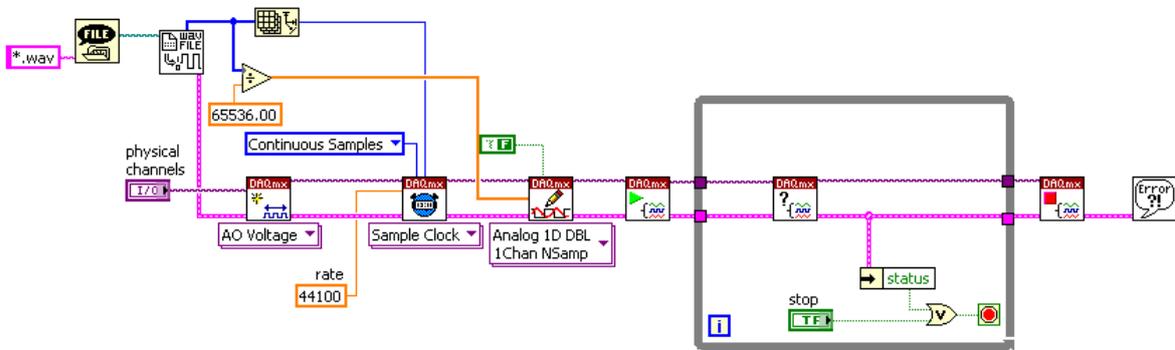


图 4-19 使用采样时钟定时的连续数据输出

四. 数字 I/O

在 DAQmx 当中，物理通道是由设备名 I/O 类型以及物理通道号组成的字符串名称。如果在 NI DAQmx 名字中省略了线号，该端口中的所有线将被包含进来。

当某根线的线号出现在 NI DAQmx 名称中时，仅有那根线处于被使用状态，需要注意的是同往常一样，您可以使用 Dev x/Port y/Line a:b or Dev x/Port y/Line a,b,c 的格式来指定多根连线。

一个数字虚拟通道可以由一个数字口组成，可以由一根数字线组成，也可以由一组线组成。

当创建一个数字输入或输出虚拟通道的时候，用户需要指定该通道是为多条线创建的还是为单独一根线所创建的。如图 4-20 所示。

需要注意的是，当在 LabVIEW 中编程创建虚拟通道的时候，所有可用的线均会出现在他们相应的端口下面。如果要让端口出现在通道数中，那么您需要改变 I/O 过滤属性。

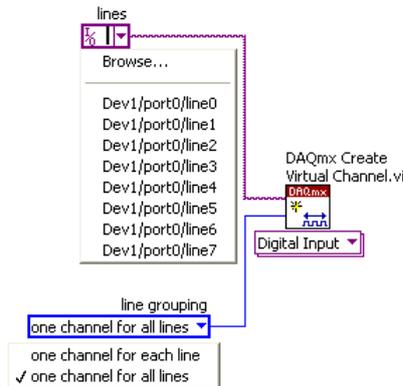


图 4-20 数字 I/O 虚拟通道设置

在大多数情况下，您会使用“one channel for all lines”来创建一个单通道。当使用较老版本 NI-DAQmx 驱动的时候，您不能修改多线通道中单线的属性(当使用“one channel for all lines”来创建的通道)。这种情况下，您可以创建“one channel for each line”并

对于每个单线通道进行相应的属性设置。

图 4-21 的例子用来读取单跟数字线的通道采样数据：

首先创建了一个单线的虚拟通道，之后开始这个数字输入任务，在 DAQmx 读取中读取外部 PORT0 Line0 上的数字信号，最后停止整个任务。

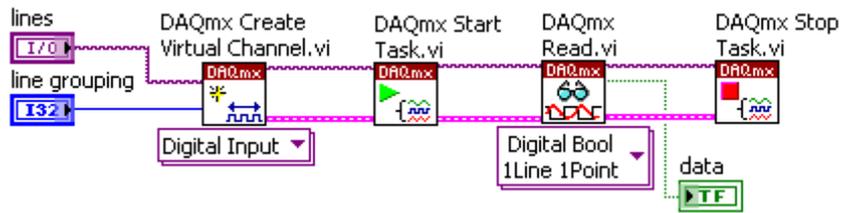


图 4-21 读取单数字线通道采样数据

如果要从多数字线通道读取采样数据，那么为多跟线创建一个虚拟通道，之后可以使用 DAQmx 读取 VI 来讲多根线的数据同时读取回来，如图 4-22 所示。

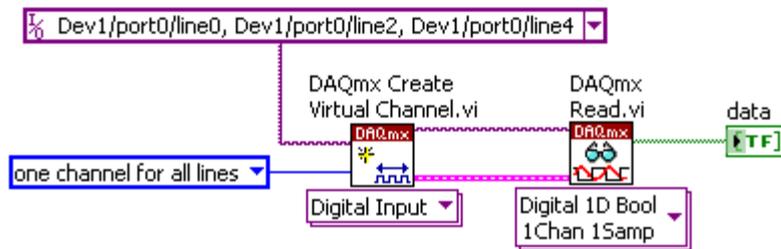


图 4-22 从多数字线通道读取采样数据

数字信号的输出也非常简单，首先我们创建数字输出通道，之后开始任务，并将数据写到相应的数字线上 最后停止任务即可，如图 4-23 所示。

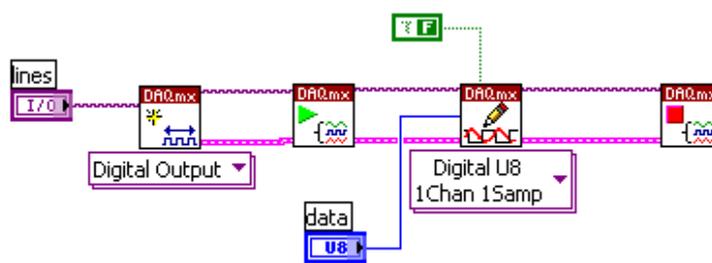


图 4-23 数字端口输出

计数器应用（上）

简介

计数器应用上、下两期节目为介绍计数器的常见应用。包括边沿计数、脉冲生成、脉冲测量和位置测量。

硬件

PXI-6259 多功能数据采集卡、BNC-2120 屏蔽接线盒、以及 68 引脚 的屏蔽电缆。6259 上有 2 个计数器，可以通过电缆和 BNC-2120 接线盒将 6259 上的引脚引出，方便进行接线。同时，2120 上的 TTL 方波信号发生器和正交编码器也会在我们的 Demo 中被使用到。

基本知识点

1. 计数器结构

计数器由四个部分组成：

- **Count register**—计数寄存器，该寄存器用来存储当前的计数值。它的存储范围跟计数器的分辨率有关，对于 PXI-6259 来说，计数器的分辨率是 32 bit，所以寄存器的计数范围是 $0 \sim 2^{32}-1$ ，到达最大值后，又从 0 开始计数。
- **Source**—被计数的信号从 Source 端引入。
- **Gate**—确定计数是否启动的门控信号。
- **Output**—用于输出单个脉冲或脉冲序列。

2. 计数器应用领域

计数器通常被应用在以下任务中：

1. 对数字脉冲信号进行边沿计数；
2. 生成单个数字脉冲或脉冲串；
3. 对脉冲的高低电平宽度、周期、频率等特性进行测量；
4. 对编码器返回的旋转角度、线性位置等信息进行测量。

下面，我们结合 LabVIEW 程序来进一步了解计数器的以上四种应用具体如何实现。

Demo 演示

演示程序全部来自 LabVIEW 范例查找器，在硬件输入与输出 >> DAQmx >> 计数器测量、生成数字脉冲 两个文件夹的下面，如图 1-1 所示。这些演示程序被整合在附件的 Counter Demos.lvproj 项目中。

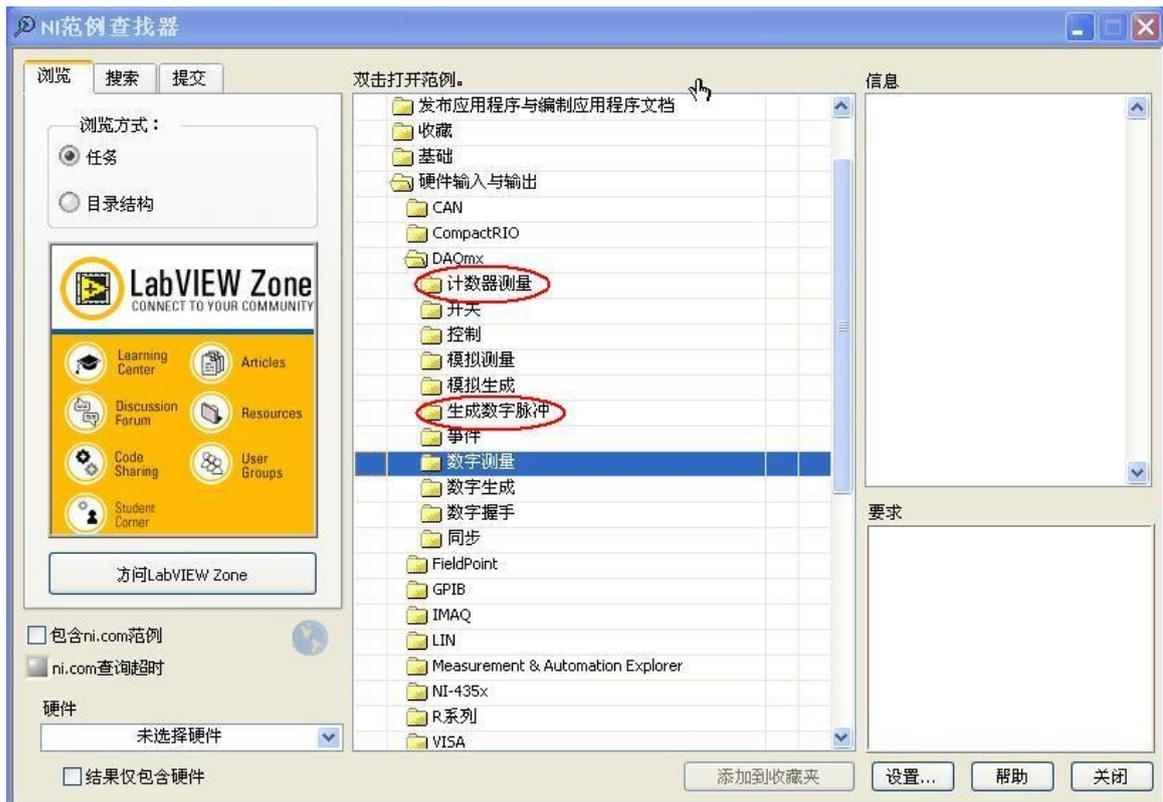


图 1-1 计数器相关范例程序的位置

1. 边沿计数

在边沿计数应用中，物理连接上只需将待计数的信号连入 Source 端即可。程序中可以设定为对信号的上升沿或是下降沿进行计数。如果待计数的信号源是频率已知的标准时基信号，我们还可以将计数值转换为时间值从而实现对时间的精确测量。如图 1-2 所示。

物理连线做法：在 MAX 中鼠标右键点击 6259，得到它的引脚定义图。由于我们准备使用 6259 中的计数器 0，它的 Source 端为 37pin，PFI8 引脚。所以，我们利用 BNC-2120 上的波形发生器生成 TTL 脉冲，并将它与 PFI8 引脚相连。

接下来，看一下范例程序“简单边沿计数.vi”。计数器通道选择 6259/ctr0，设置计数方向（向上递增还是向下减小），待计数的脉冲边沿（对上升沿计数还是下降沿计数），初始计数值（一般设为 0），然后运行程序即可。

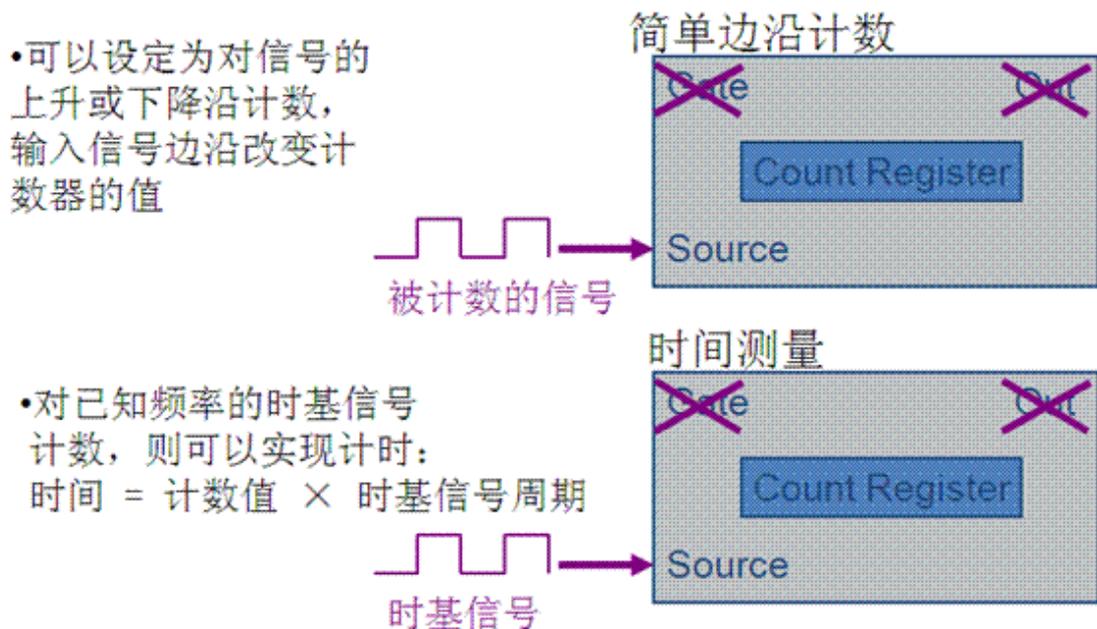


图 1-2 边沿计数原理图

有时在计数应用中会增加一个门控信号，当门控信号有效时，才对脉冲边沿进行计数，门控信号无效，则计数值不改变。参考范例“门控边沿计数.vi”，它使用了“DAQmx 触发”属性节点，如图 1-3 所示。在该属性节点中设置门控信号的输入引脚（演示程序中为 6259/PFI1），以及停止计数的门控信号状态（演示程序中为高电平）。经过这样的配置后，当 PFI 引脚上的信号为低电平时，计数器 0 正常计数；如果 PFI 引脚上的信号为高电平，则计数器暂停计数。

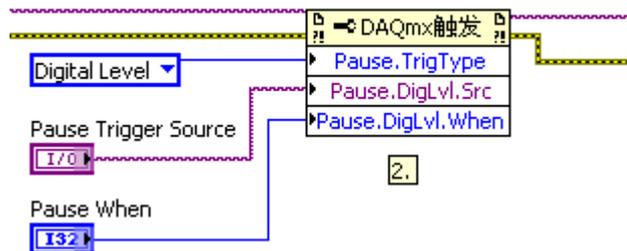


图 1-3 门控边沿计数通过属性节点设置

2. 脉冲生成

计数器生成单个数字脉冲和脉冲序列，并且可以通过软件设置脉冲的频率和占空比。在物理连接上，输入端不需要任何连线，DAQmx 驱动底层会选择计数器中适合的时基信号做为 Source 端的输入。对于 6259 来说，它的内部有三个时基信号，频率分别为 80M、20M 和 100kHz，计数器会自动对这些时基信号进行分频处理，从而得到指定频率的脉冲序列，并通过 OUT 端输出。如图 1-4 所示

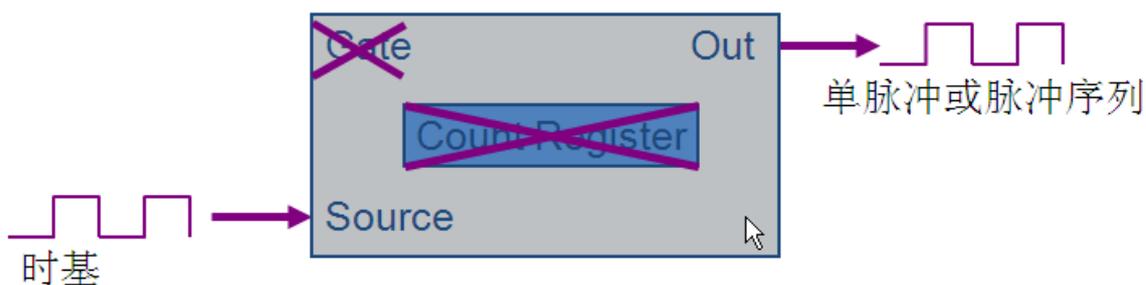


图 1-4 脉冲生成原理图

在程序设计上，参考范例“连续脉冲生成.vi”。程序中，DAQmx 创建通道多态 VI 选择“计数器输出》脉冲生成》频率”，并设置输出脉冲的频率和占空比，DAQmx 时钟 VI 选择“隐式》计数器”模式。运行程序，将计数器 0 的 OUT 端输入的信号接入数字线 p0.7，通过 MAX 中的测试面板观察输出脉冲序列的状态。

3. 脉冲测量

脉冲测量又可以细分为脉冲宽度测量、周期半周期测量、以及频率测量。在本期的节目中，我们先介绍前两种应用。

脉冲测量是使用已知频率的时基信号对未知信号进行测量，在物理连接上，频率较高的时基信号接入 Source 端，而频率较低的待测信号接入 Gate 端。如图 1-5 所示。

使用已知频率的时基信号对未知信号进行测量

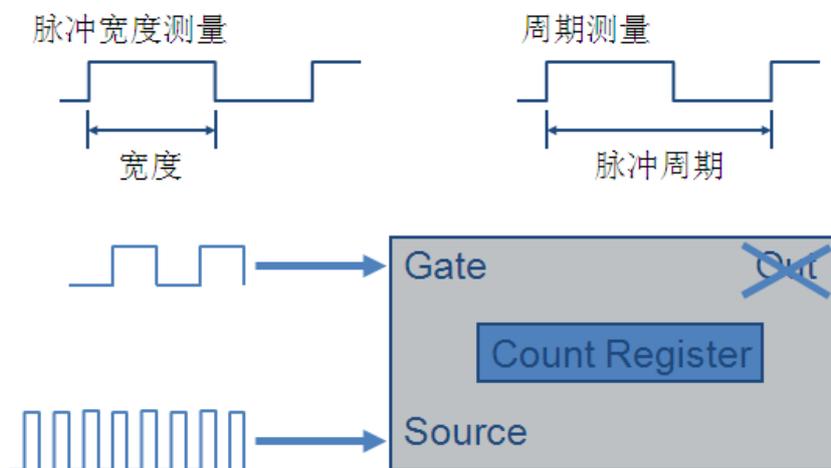


图 1-5 脉冲测量原理图

- 脉冲宽度测量

首先来看脉冲宽度测量，它的原理是将待测脉冲的高电平或低电平置为有效的门控信号，在这段时间内对 Source 端的时基信号进行计数，将计数值乘以时基信号的周期，就得到 Gate 端信号的脉冲宽度。如图 1-6 所示。

Gate脉冲宽度 = 计数值 x 时基信号周期

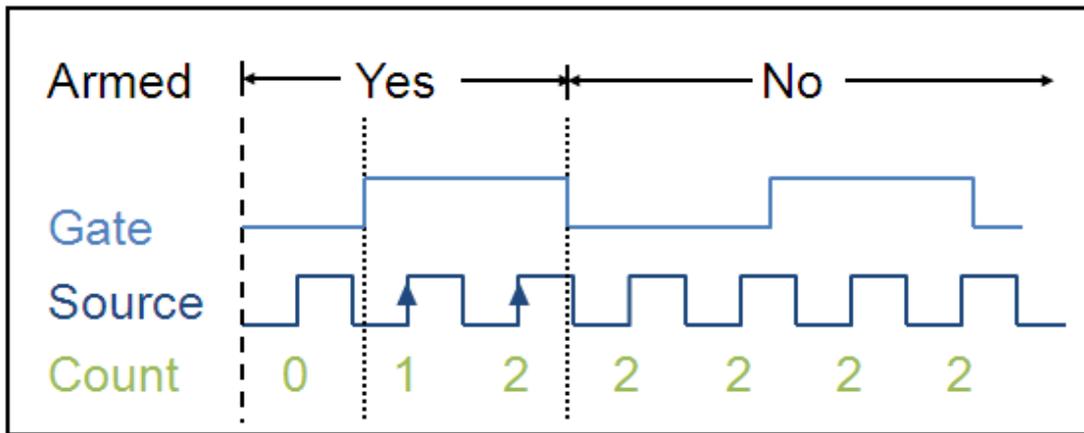


图 1-6 脉冲宽度测量原理图

程序设计上，参考范例“脉冲宽度测量.vi”。对于 DAQmx 创建通道多态 VI，选择“计数器输入》脉冲宽度”，同时需要设置开始边沿，如果选择上升沿，则对脉冲的高电平宽度进行测量，如果选择下降沿，则对低电平宽度进行测量。同时，如果知道待测脉冲宽度的大致范围，还可以设置一个最小和最大阈值，驱动底层会根据这个值去自动选择合适的时基信号进行更为精确的测量（范例设置下，驱动将选择 6259 的 20M 时基信号做为 Source 端的输入信号）。

- 周期/半周期测量

周期/半周期测量与脉冲宽度测量的物理连接和基本原理其实是一样的。尤其半周期测量，它返回的是数字信号两次状态转换的时间间隔，本质就是脉冲宽度测量。如图 1-7 所示。

Gate脉冲周期（半周期） = 计数值 x 时基信号周期

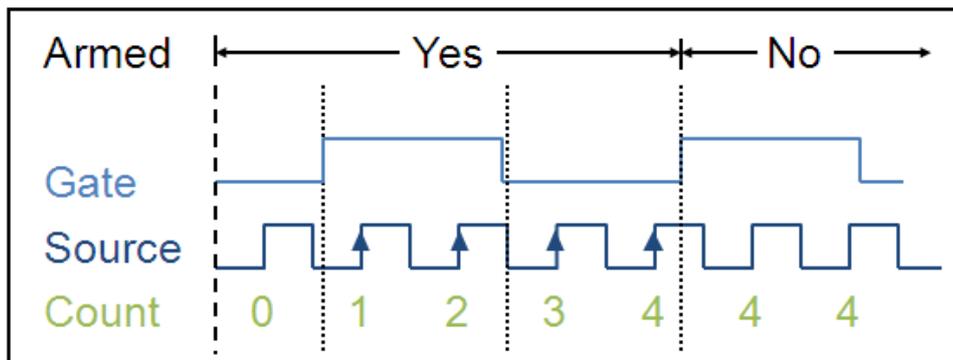


图 1-7 周期/半周期测量原理图

它们在程序设计上也非常相似，参考范例“周期测量.vi”、“半周期测量.vi”。主要的区别就是“DAQmx 创建通道”多态 VI 的选择不同。物理连接上只需要将待测信号连入计数器的 Gate 端，其余的工作 LabVIEW 程序和 DAQmx 驱动会自动完成，非常方便。运行程序，

得到周期和半周期数值，跟前面脉冲宽度测量的结果对比，说明测量值是准确的。

计数器应用（下）

简介

本期节目继续“计数器应用（上）”中未完的内容，介绍如何使用计数器实现频率测量和位置测量。

硬件

用到的硬件设备跟上期节目相同，依然是 PXI-6259 多功能数据采集卡、BNC-2120 屏蔽接线盒、以及 68 引脚的屏蔽电缆。

Demo 演示

1. 频率测量

频率测量有三种方法：周期取反法、平均法和分频法。它们的适用情况不同，以下分别介绍。

- 周期取反法

首先是周期取反法，它的原理和物理连接跟上期节目中介绍的脉冲周期测量是相同的。即将内部时基信号或已知频率的标准信号接入 Source 端，将待测信号接入 Gate 端。待测信号周期等于计数值乘以 Source 端信号周期。对周期取倒数，就得到待测信号的频率。如图 1-1 所示。

$$\text{待测信号周期} = \text{计数值} \times \text{时基信号周期}$$

$$\text{频率} = 1 / \text{周期}$$

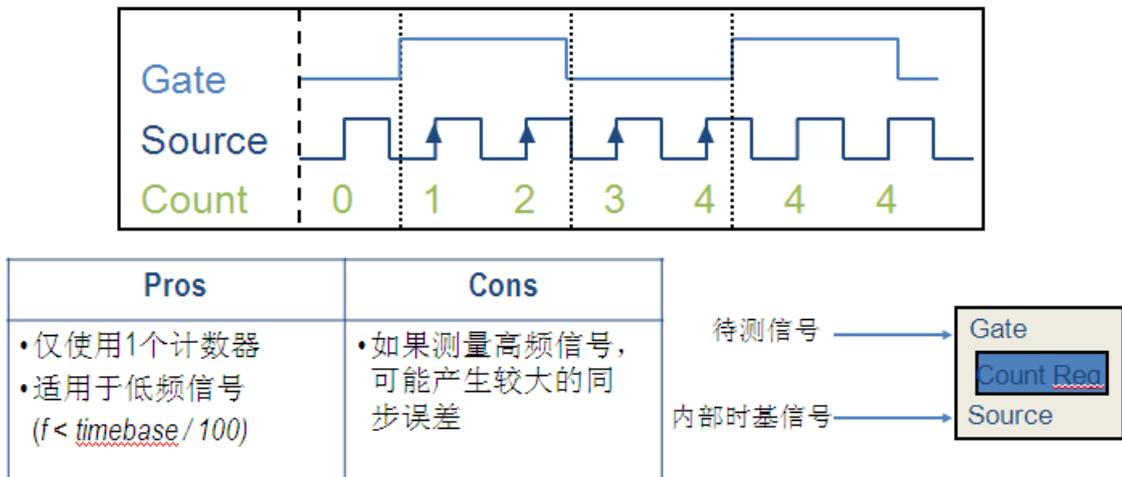


图 1-1 周期取反测频率原理图

程序参考范例“频率测量_1 Ctr.vi”。跟周期测量的程序非常类似，唯一的区别就是将“DAQmx 创建通道”多态 VI 设置为“计数器输入》频率”。并且设置待测频率的范围，以便驱动底层去自动选择适合的内部时基信号做为 Source 信号。

周期取反这种测量方法的特点是简单，且仅使用一个计数器，它适用于低频信号的测量，即待测信号频率应该低于 Source 端时基信号频率的百分之一。如果待测信号频率较高，将产生较大的同步误差，导致测量结果不准确。

那么什么是同步误差呢？我们通过图 1-2 来说明一下。图中的 Gate 信号高电平宽度约为 Source 端信号的 4 个周期，但由于 Gate 信号的上升沿与 Source 信号第一个脉冲的上升

沿不能完全同步，可能会造成 Source 端第一个脉冲上升沿被漏计数，而第五个脉冲上升沿被误计数的结果，所以测得的 Gate 信号脉冲宽度可能有 ± 1 个 source 周期的误差存在。这种误差就叫做同步误差。

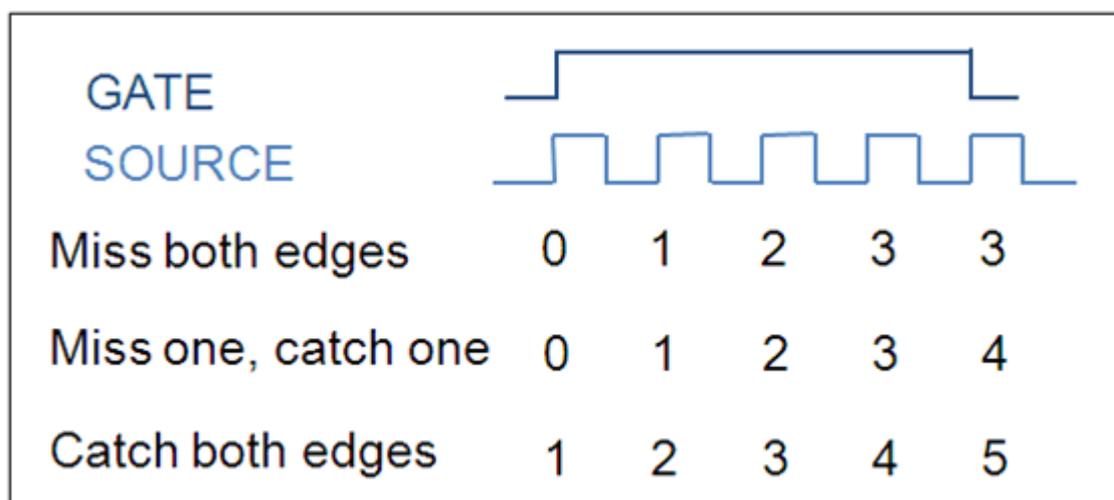


图 1-2 同步误差产生原理图

在时基信号已定的情况下，待测信号频率越高，越接近时基信号频率，产生的测量误差将越大。如图 1-3 所示，如果时基信号频率为 20M，待测信号频率为 50k，则在 Gate 信号的一个周期内，应该得到的时基信号脉冲计数值为 400。但由于同步误差的存在会产生 ± 1 的计数偏差，所以实际得到的测量值为 49.88k 或 50.13k。误差相对比较小。但如果待测信号频率为 5M，则在 Gate 信号的一个周期内，应该得到的时基信号脉冲计数值为 4，此时 ± 1 的计数偏差将产生很大的误差，实际测量的频率值将为 4M 或 6.67M。所以这种情况下，我们需要考虑使用另外的方法来进行频率测量。

时基信号频率为 20M Hz

Actual Frequency	Number of 50 ns Cycles	Measurement Error of +1 Cycle	Measurement Error of -1 Cycle	Frequency with Error of +1 Cycle	Frequency with Error of -1 Cycle
50 kHz	400	401	399	49.88 kHz	50.13 kHz
5 MHz	4	5	3	4 MHz	6.67 MHz

图 1-3 同步误差对比分析

- 平均法

第一种适用于高频信号测量的方法叫平均法。

根据同步误差产生的原理，在 Gate 信号有效的时间内，Source 信号的脉冲数量越多，同步误差就越小。平均法就是根据这样的原理，如图 1-4 所示。在 Source 端信号频率已定的情况下，延长 Gate 端的有效时间来提高测量精度。与周期取反法不同的是，平均法需要 2 个计数器，将频率较高的待测信号接入其中一个计数器的 Source 端，然后用另一个

计数器，通过内部时基信号生成指定时间长度的 Gate 信号。物理连接上，我们只需将待测信号接入其中一个计数器的 Source 端，在程序中，这个计数器也就是我们应该选择的计数器通道。其它的配置和物理连接，DAQmx 驱动会自动帮我们完成。



$$\text{待测信号频率} = \text{计数值} / \text{计数时间}$$

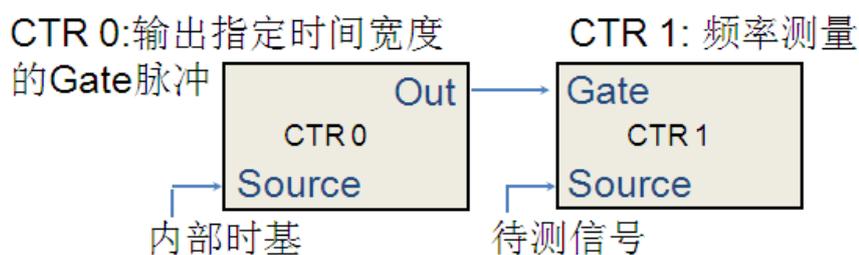


图 1-4 平均法测频率原理图

程序参考范例“频率测量_2 Ctr_一段时间内.vi”。“DAQmx 创建通道”多态 VI 选择“计数器输入》频率”，由于使用了两个计数器，所以测量方法接线端选择“High Frequency with 2 Counters”。但在“计数器”通道接线端，只需选择 Source 端与待测信号相连的那一个计数器，另一个计数器驱动会自动地帮我们选择。同时，需要指定测量时间。DAQmx 定时 VI 选择隐式，表示采样率由待测信号本身的频率决定。

- 分频法

另一种适用于高频信号测量的方法叫分频法。

在周期取反法中，当待测信号，也就是 Gate 端信号频率接近 Source 端的时基信号频率时，会产生较大的同步误差。所以分频法中，先将待测信号做分频处理，降低它的频率，以保证测量结果的准确性，然后将得到的频率值乘以分频系数，就可以还原出待测信号的真实频率。所以，分频法也需要使用 2 个计数器才能实现。物理连接上，同样只需将待测信号接入一个计数器的 Source 端即可。其它的连接，驱动会自动帮我们完成。如图 15-所示。

Gate信号频率= 时基频率/计数值

待测信号频率=Gate信号频率x 分频系数

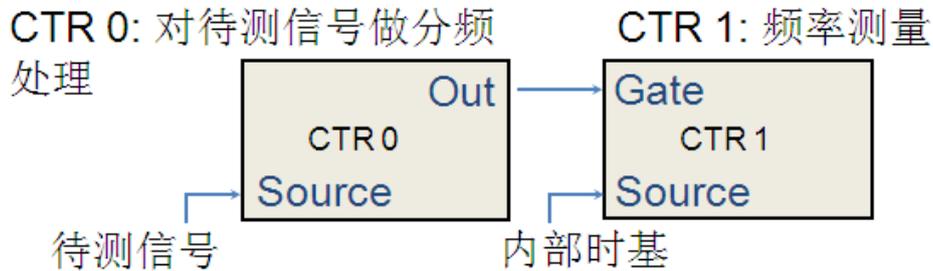


图 1-5 分频法测频率原理图

程序参考范例“频率测量_2 Ctr_分频.vi”。 “DAQmx 创建通道”多态 VI 同样选择“计数器输入》频率”，测量方法选择“Large Range with 2 Counters”。设置待测频率的范围，以便 DAQmx 选择合适的时基信号。设置分频系数，使得待测信号分频后的频率低于时基信号频率的百分之一。

2. 位置测量

介绍完频率测量的三种方法，我们接下来看位置测量。位置测量又分为线性位置测量和角度测量，它们分别配合线性编码器和旋转编码器使用。这两类编码器的工作原理类似，都使用光学传感器来提供两路脉冲序列形式的电信号，分别为序列 A 和序列 B。由于两路脉冲信号之间有 90 度的相位延迟，即有正交的关系，所以这些编码器又被叫做正交编码器。如图 1-6 所示。

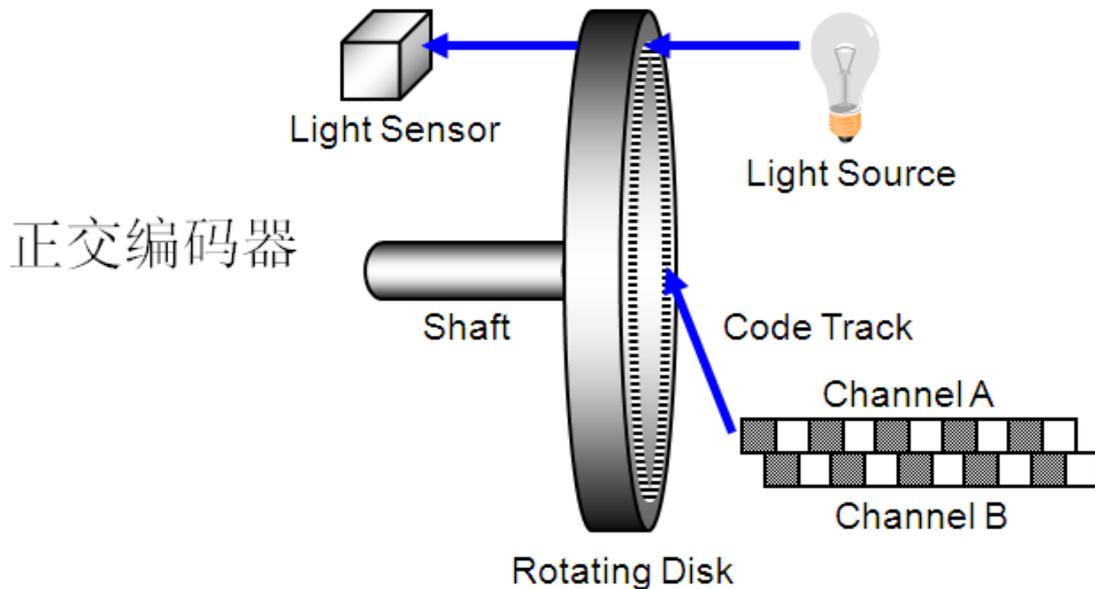


图 1-6 正交编码器结构图

当编码器转动时，产生两路数字脉冲信号 A 和 B。如果 A 相位超前 90 度，说明编码器以顺时针方向旋转；如果 B 相位超前 90 度，说明编码器以逆时针方向旋转。同时我们知道，编码器每旋转一周会产生多少个脉冲。对于 BNC 2120 上的编码器来说，旋转一周将产生 96 个脉冲。所以，通过计数器监控脉冲数目和信号 A、B 之间的相对相位信息，就可以获

得旋转角度和旋转方向的信息。

在物理连接上，将信号 A 接入计数器的 Source 端，并在程序中设置对序列 A 的脉冲下降沿进行计数。将信号 B 接入计数器的 AUX 端，AUX 端决定了计数方向，当它的输入为高电平时，计数器向上计数，当它的输入为低电平时，计数器向下计数。如果正交编码器顺时针旋转，则序列 A 超前序列 B 90 度，所以 A 脉冲的下降沿对应序列 B 的高电平，计数值总是向上增加的；如果正交编码器逆时针旋转，则序列 B 超前序列 A 90 度，A 脉冲的下降沿对应序列 B 的低电平，计数值总是向下减少。所以，根据计数值增加还是减少可以判断出编码器的旋转方向。同时通过计算还可以将计数值转换为位置信息。如图 1-7 所示。

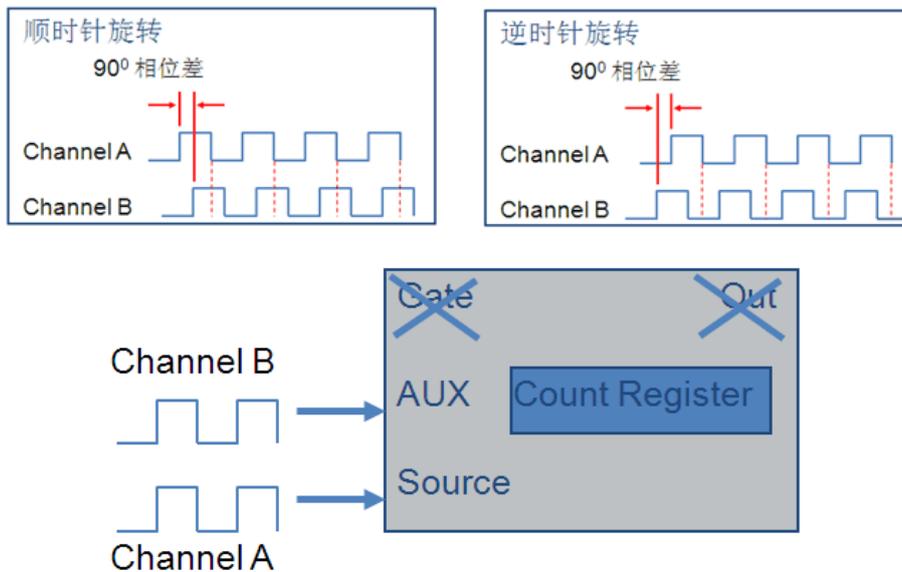


图 1-7 位置测量原理图

对于角度测量，转换关系为：计数值除以编码类型再除以旋转一周产生的脉冲数目，然后乘以 360 度。常用的编码类型有 $\times 1$ ， $\times 2$ 和 $\times 4$ 三种，由所选编码器决定。对于线性位置测量，转换关系为：计数值除以编码类型、除以旋转一周产生的脉冲数目，然后再除以每英寸对应脉冲数目 PPI，PPI 的具体值也由所选编码器决定。计算公式如图 1-8 所示。

角度测量

$$(^{\circ}) = \frac{Edge_Count}{xN} \cdot 360^{\circ}$$

线性位置测量

$$(\text{英寸}) = \frac{Edge_Count}{xN} \cdot \left(\frac{1}{PPI} \right)$$

程序参考范例“角度测量.vi”。在“DAQmx 创建通道”多态 VI 处选择“计数器输入》位置》角度编码器”。物理连接上，将编码器输出的序列 A 接入计数器 0 的 Source 端，序列 B 接入计数器 0 的 AUX 端（由于在 BNC2120 接线盒上，计数器 0 的 AUX 端口没有被引出，所以我们把该端口与 PFI2 引脚相连，在程序中使用 DAQmx 通道属性节点，选择“计数器输入》位置》B 输入》接线端”，将 PFI2 做为该属性节点的输入，然后将序列 B 连接到 BNC2120 的 PFI2 引脚上）。计数器通道选择 Counter0。解码类型根据所选编码器的不同，可选择 $\times 1$ ， $\times 2$ ， $\times 4$ ，以及 Two Pulse Counting。旋转一周脉冲数目也根据所选编码器决定，对于 BNC 2120 的编码器，它的值为 96。

除此之外，有些正交编码器还包含被称为零信号或者参考信号的第三个输出通道，这个信号通常被称为 Z 轴或者 Z 索引。这个通道每旋转一圈输出一个单脉冲。可以使用这个单脉

冲来精确计算某个参考位置。所以如果使用到 Z 索引，就按下 Enable 布尔控件，并对 Z 索引的值和相位进行配置。由于 BNC2120 上没有 Z 所以功能，所以无需额外的配置。

定时与触发

简介

本期节目主要介绍 NI 多功能数据采集板卡的定时与触发架构，以及如何使用属性节点来进行相应的配置。

PXI 总线不仅提供了 PCI 规范高达 132M 字节每秒的数据吞吐量，还增加了定时和触发功能，所以我们可以非常方便地在 PXI 背板上传递时钟和触发信号，以适合我们的应用。下面介绍一下多功能 M 系列板卡的定时架构。

M 系列定时引擎

图 7-1 所示是 NI 的 M 系列多功能数据采集卡定时引擎简图。注意到，这里的 AI 和 AO Sample Clock, AI Convert Clock 就是在我们的板卡上控制采集和波形发生的时钟，而所有这些时钟都从同一个参考时钟得到。

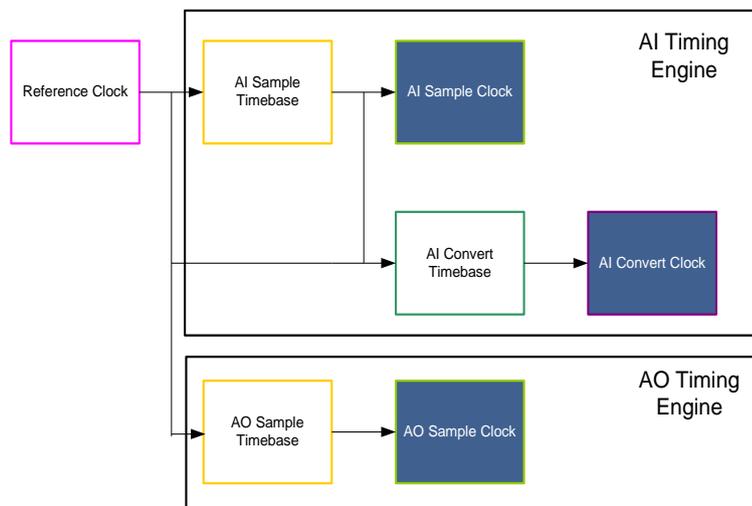


图 7-1 M 系列定时引擎简图

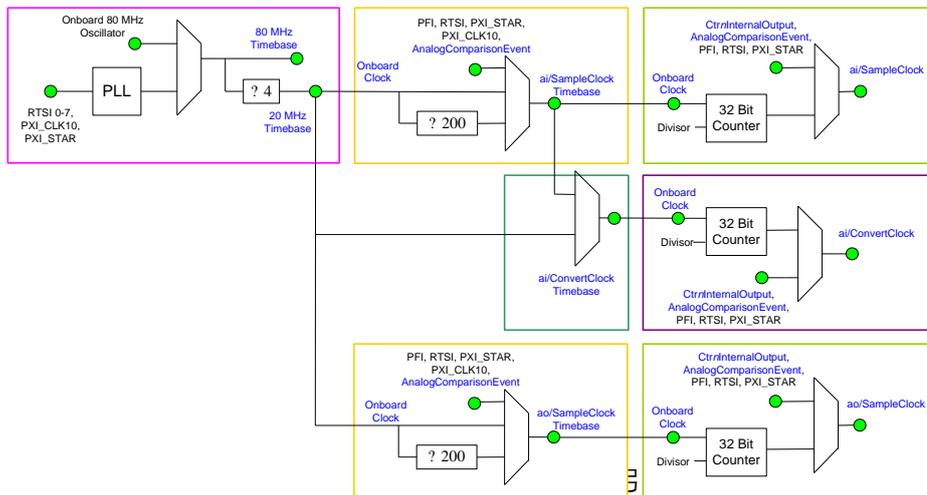
以下是对这几个时钟信号的简单解释：

- AI sample clk 决定了多长时间内每个通道能拿到一个采样点，
- AI convert CLK 决定了 AD 转换器每次实际转换的间隔

- AO SAMPLE CLK 决定了每个通道产生一个采样点的时间间隔

定时引擎的详细结构

将 M 系列定时引擎简图展开，图 7-2 给出了详细的定时拓扑。根据不同的应用，这三个时钟可以从多种途径得到，以下，以 AI Sample Clock 为例——介绍如何从不同的时基获取相应的模拟输入采样时钟。



- 情况一，使用板上时钟：

注意图 7-3 中的红色通路部分，我们可以通过板上的 80MHZ 时基，进行四分频后，降频到 20MHZ，得到 AI 采样时基，再通过一个 32 位的内部时钟进行分频，得到您设定的采样率。这个过程，包括通路选择和降频系数的计算都已经由 DAQmx 驱动完成，在 LV 中你所要做的只是指定时基的来源和给定采样率。如图 7-3 中左下角所示。

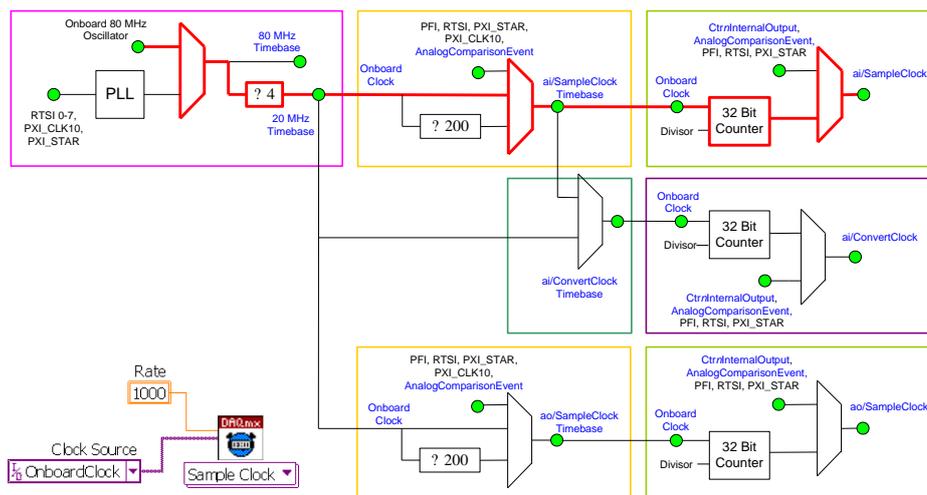


图 7-3 使用板上时钟

- 情况二，从外部输入时基以与其他设备同步：

如果需要与外部信号或外部设备同步，您就会选择这条通路。从图 7-4 中可以看到在左上角，从外部输入时钟经过锁相环得到基本时基，接下来的过程与图 7-3 相同。同样，DAQmx 为您作了绝大多数工作，您只需要通过相应的属性结点进行设置。这里的例子选择了 PXI 机箱的背板 10M 时钟作为参考时钟，关于同步的应用，我们将在第九和第十集中向您详细介绍。

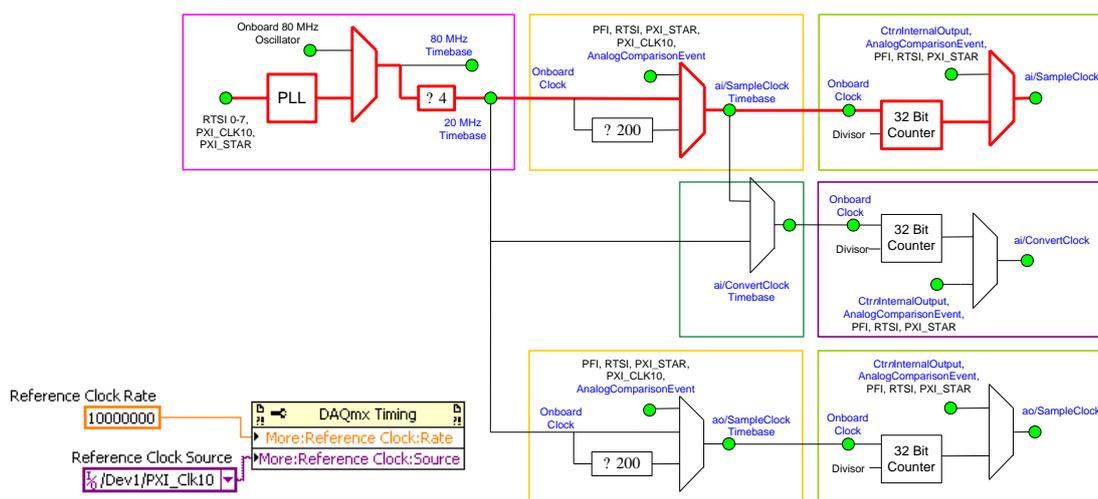


图 7-4 从外部输入时钟以与其他设备同步

- 情况三，直接使用外部采样时钟：

如果您对采样时钟有特殊要求，需要设定为一个很特别的频率，您可能会自己通过专用仪器产生一个专用信号来作为我们板卡上的采样时钟，NI 数据采集卡的定时引擎就为您提供了这样的灵活性，如图 7-5 所示，直接右上角的 PFI/RTSI/PXI_STAR 输入您的信号。配置方法和第一种方法类似。这个例子中，我们使用了设备的 PFI7 脚作为专用信号输入端口，如图 7-5 左下角所示。

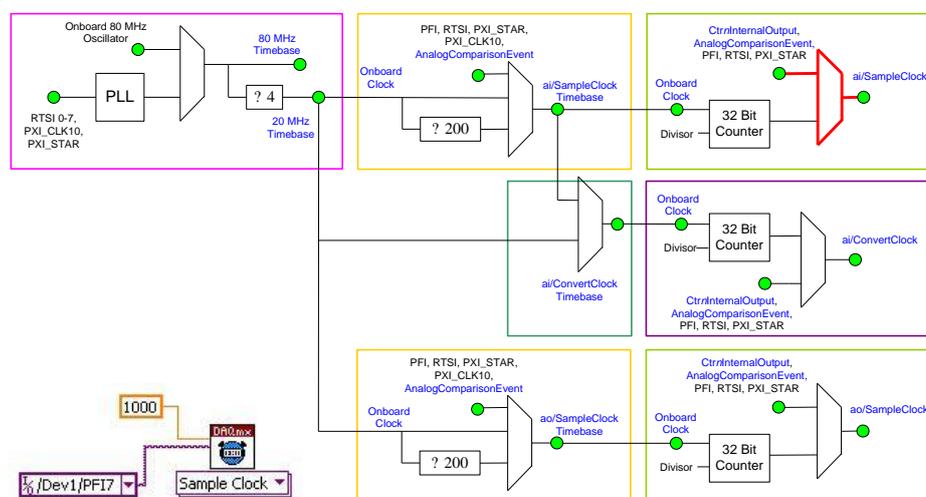


图 7-5 直接使用外部采样时钟

- 情况四，外部采样时钟作为采样时基

如果您希望既使用外部时钟又能获得多种采样率，您可以选择这种方法。把外部时钟作为采样时基输入，利用内部计数器作分频，这样既满足了您的要求，又无须增加额外的设备。如图 7-6 所示。

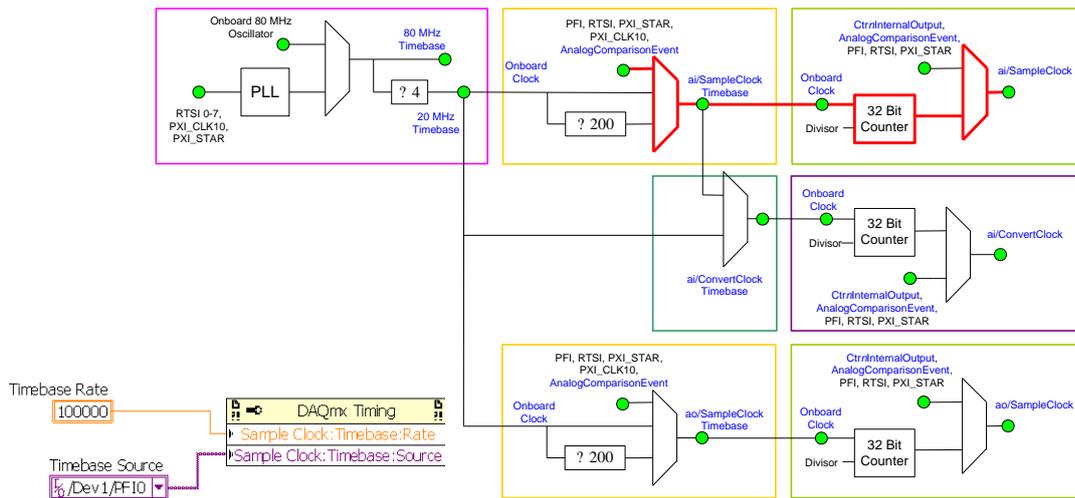


图 7-6 外部采样时钟作为采样时基

接下来我们就通过实际的应用来看看 LV 中的程序实现。

定时 Demo 演示：

在这个 Demo 中我们所要做的是，把 PXI 的 10M 背板时钟作为板卡的时基通过定时引擎获得采样时钟。整个通路如图 7-7 中红色部分所示。

相应的 LV 程序如图 7-8 所示。

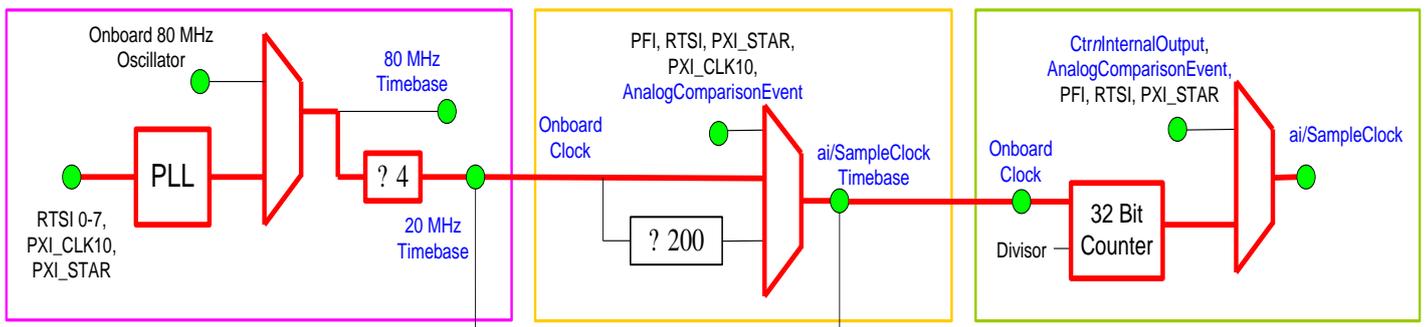


图 7-7 10M 背板时钟作为板卡的时基

在这个例子中我们分别配置了 AI 和 AO 模拟输入输出通道均设置为连续采样模式,通过 DAQmx 定时属性节点将 PXI 的 10M 背板时钟作为板卡的时基。DAQmx 驱动会根据设置的输入输出采样和更新率来得到相应的采样时钟 两者通过 PLL 共享 10M 时基。

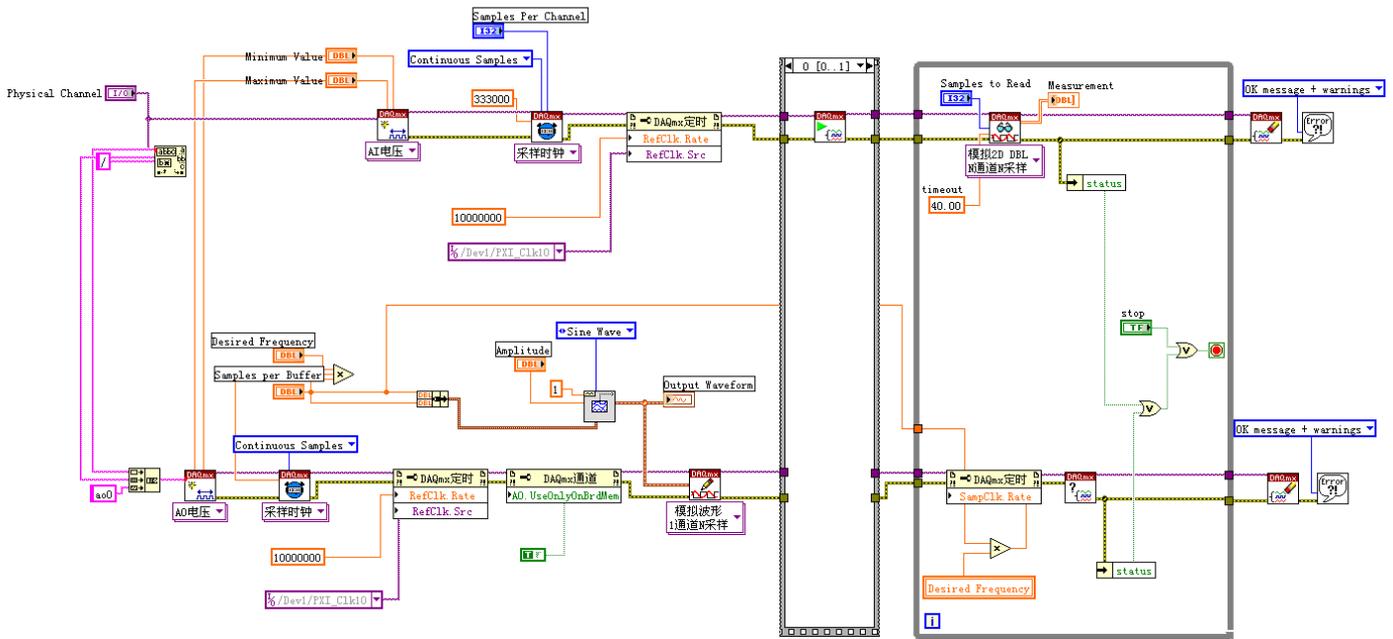


图 7-8 使用 PLL 共享外部时基 LabVIEW 程序

触发

除了可靠的定时架构之外，我们还需要一类信号来控制数据采集或是信号生成，NI 的数据采集系统提供了丰富的触发机制，包括了开始触发，停止触发，参考触发等等，不同的触发方式可以被灵活地应用在不同的场合，以下详细介绍几种常用的触发机制。

- 第一种称为 Start Trigger，即开始触发，通过它来开始一个数据采集过程。
- 其次是 Pause Trigger，它根据触发信号的电平来决定是否执行采集，也就是说可以在采集开始后暂停或继续采集。
- Reference Trigger 即参考触发是经常被混淆的一种触发，它并不等同于停止触发，通过 Reference Trigger，我们可以获得触发点前后指定数量的采集点。

图 7-9 很清楚地说明了 Start Trigger 和 Reference Trigger 的用途。在收到 Start Trigger 后，每个采样脉冲来到都进行一次数据采集，而在收到 Reference Clock 后，保存 Pretrigger 的点数，并在经过指定的 Posttrigger 点数后停止采集。这两个数字在 DAQmx 编程里分别由 pretrigger Samples per channel 和 Samples per channel 决定。Samples per channel 的数值减去 PRETRIGGER SAMPLES 的数值后便得到了

POSTTRIGGERsample 的个数。

这个例子中，我们将 Samples per channel 设置为 7，PRETRIGGER SAMPLES per channel 设置为 5 于是 POSTTRIGGERsample 的个数即为 2，如图 7-9 的下半部分所示。

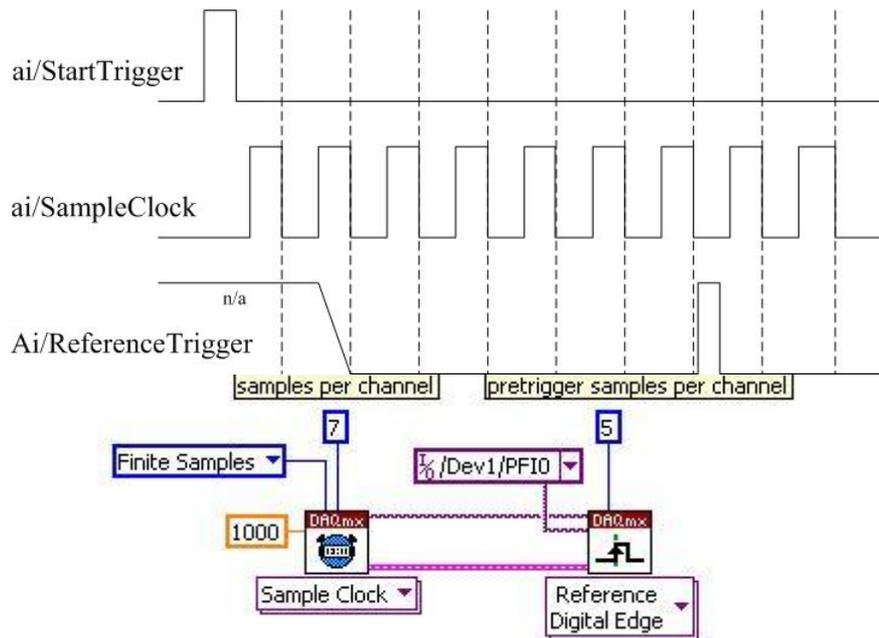


图 7-9 开始触发和参考触发

图 7-10 是对 pause trigger 很好的解释。当 ai/Pause Trigger 为低电平时，ai/Sample Clock 暂停，图中虚线部分的脉冲表示被屏蔽的采样时钟，而当 Pause Trigger 再次抬高时，采样时钟再次出现，进行采样操作。如此便能达到用一个门信号来控制采样过程的目的。

这个例子的 LV 程序实现需要使用 DAQmx TRIGGER 属性节点，通过配置相应的触发源，触发类型以及有效电平来完成相应的功能，如图 7-10 的下半部分所示。

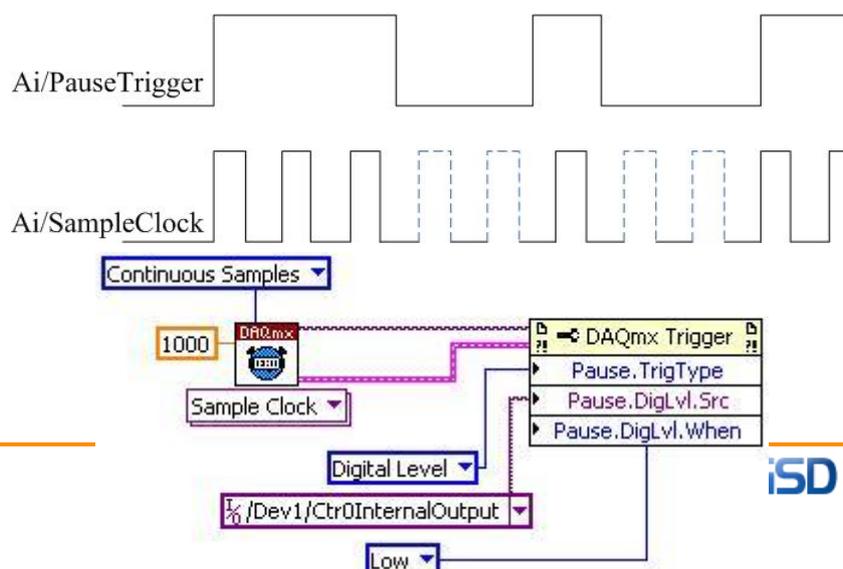


图 7-10 停止触发

对于模拟触发，有两种类型：沿触发和窗触发，如果触发信号带有较大噪声，为了避免噪声使系统误动作，我们通常使用带滞后的沿触发，在 LV 中也是经过简单的配置来实现，DAQmx TRIGGER 属性节点下提供了相应的滞后边沿触发设置，如图 7-11 所示。实际上底层的很多工作都由 DAQmx 驱动来完成了。

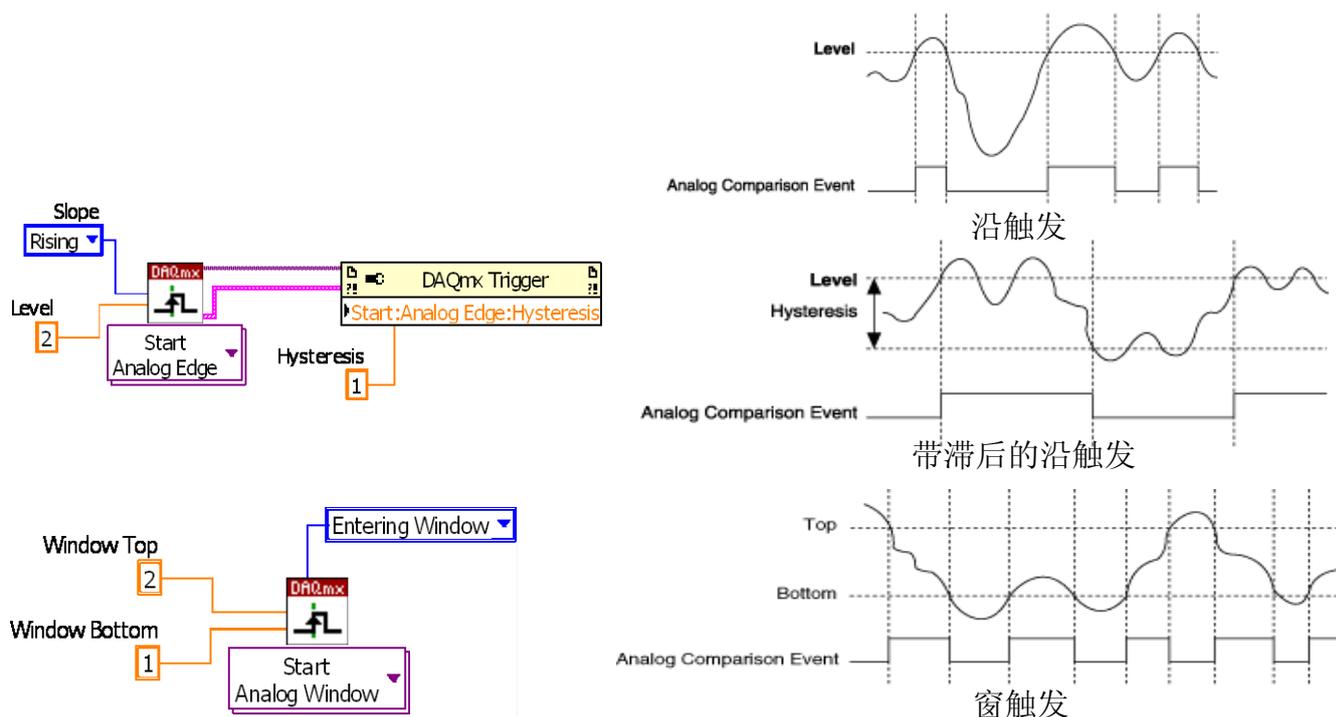


图 7-11 模拟触发类型

刚才我们是从应用层和驱动层的角度来讨论触发，现在我们再进一步，看看硬件上触发信号是怎么产生的。注意图 7-12 中的多路开关，它有两个输入，一个来自模拟触发线 APFI0, 1，另一个来自模拟输入通道。这两种通路都可以作为触发信号的输入通道，但各有利弊。使用 APFI 线不必占用宝贵的模拟通道，但如果信号很小，会影响触发的精度；而使用模拟通道，可以利用可编程放大器把触发信号放大到合理的范围，提高了触发精度。您可以根据您的具体应用选择。并且我们注意到，模拟触发比较器的输出可以路由到不同的子系统，扩大了触发信号的应用范围。

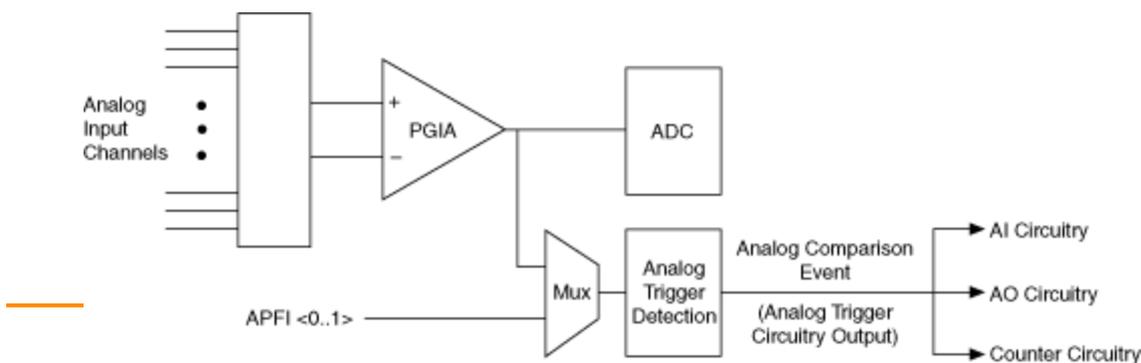


图 7-12 硬件触发信号拓扑

触发 DEMO 演示:

现在我们来查看一个使用 Reference Trigger 的 Demo，在这里我们使用 Reference Trigger 来停止一个采集过程，如图 7-13 所示。程序中使用的 AO 通道用来产生一个供 AI 采集的信号。

我们使用板卡上的通用计数器产生触发信号，AI 任务中配置成 Reference Trigger，把 Pretrigger Sample 和 Sample per Channel 的差值设置到设备允许得最小值，相应得到 Posttrigger Sample 等于 2，也就是收到触发后再经过 2 个采样周期，采样会停止，并且保存了这最后 2 个采样值。如果您的目的只是使采样停止，完全可以忽略这两个值。理解了不同的触发类型之后，我们发现编写基于 DAQmx 的触发程序变得简单快捷。

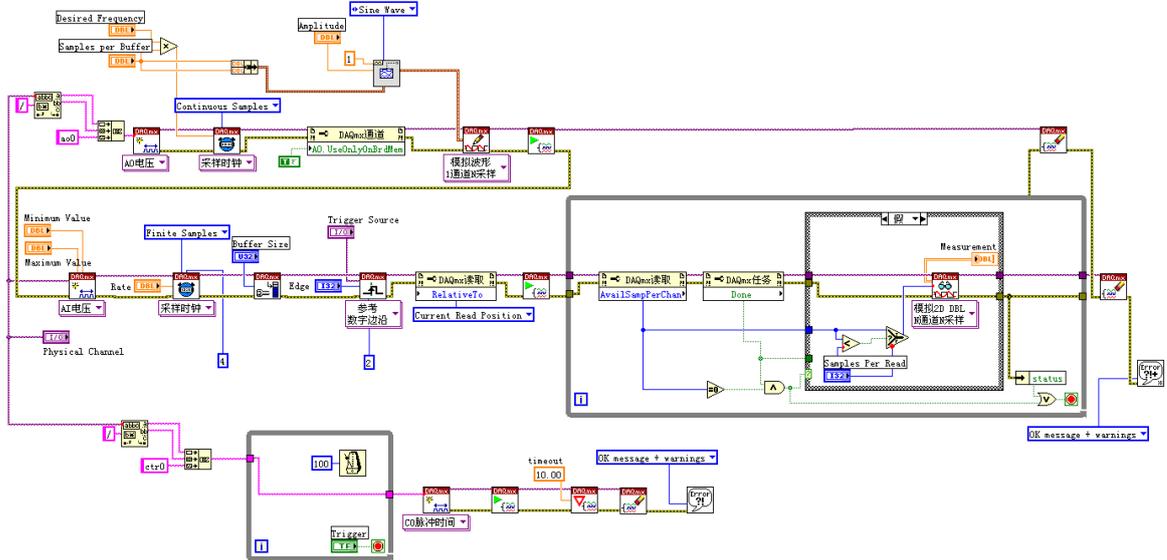


图 7-13 触发 DEMO 演示