

数据采集编程指南 下篇



目 录

数据存储与文件 I/O	1-8
同步（上）	9-14
同步（下）	14-21
特别篇：模块化仪器	22-26

数据存储与文件 I/O

简介

本期内容将介绍如何使用 NI 数据采集板卡来实现数据的存储和文件 I/O 操作。

在一个典型的测试测量系统当中，除了进行信号调理，信号采集，信号处理，信号显示之外，我们常常还会需要将采集到的数据存储到磁盘上，用于做后续离线处理或是作为数据日志。在另外一种情况下，我们可能会将事先存储好的数据文件加载到数采系统中，通过我们的板卡进行信号的输出，这就不得不使用文件 I/O 的操作。

几种常用文件格式比较

说到文件 I/O。我们简要地回顾一下几种常用的数据存储格式。

将文件写入计算机硬盘时，文件在最底层是以一串二进制位表示的。当然还有很多格式可用于组织和表示文件中的数据。

LabVIEW 中最常用的三种数据存储格式是：

- ASCII（美国标准信息交互码）文件格式
- 直接二进制存储
- TDMS（技术数据管理流）文件格式

下面我们简单地比较一下三种常用文件格式的优缺点，总结如表 8-1 所示。

	ASCII	TDMS	直接二进制
数值精度	好	最优	最优
共享数据	最优 (任何文本程序容易进行读取)	次优 (NI 程序容易进行读取)	好 (仅有元数据)
效率	好	最优	最优
理想适用范围	在磁盘空间和精度不重要时，用于与其他程序共享数据	将简单数组数据和元数据与编程者进行共享	紧凑的存储数值数据，并提供随机访问功能

表 8-1 三种常用文件格式的优缺点

ASCII 文件格式具有方便其他用户或应用程序访问数据的特性但是仅当磁盘空间和文件 I/O 速度都不重要且不需要对文件进行随机读写时 我们才会选择使用 ASCII 类型的文件作为存

储。另外 ASCII 文件的数值精度相对其他两者来说没有优势。

ASCII 文件的缺点即是直接二进制存储的优点：当数值精度很重要，需要随机访问存储数据且效率需要得到考量的场合，我们常会使用二进制直接存储。

对于 TDMS 我们会在后面做详细的介绍

LabVIEW 中的文件 I/O

LabVIEW 中提供了高层文件和底层文件 I/O 函数来进行文件 I/O 的操作，图 8-1 中红

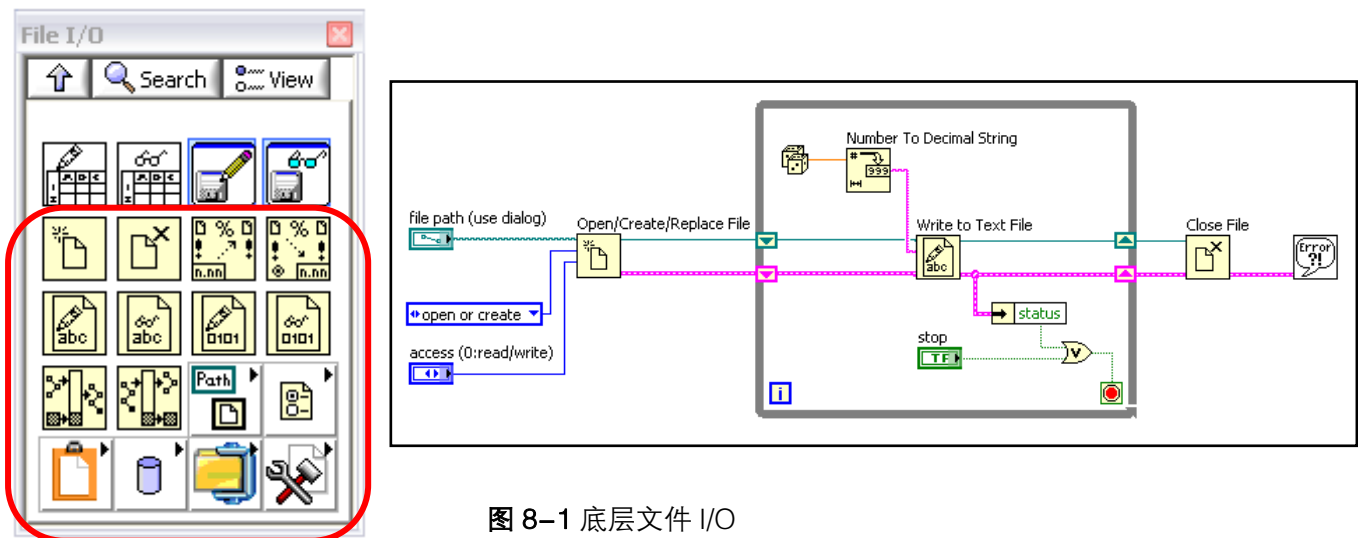


图 8-1 底层文件 I/O

色圈内的部分为底层的文件 I/O 函数，每个函数完成相对独立的功能，如打开文件，写入文件，读取文件，关闭文件等等。通过分立地进行文件操作可以提高连续文件写入或读取的效率，一个典型的底层文件 I/O 完成文件写入的例子如图 8-1 右图所示，分别在 WHILE 循环外面打开和关闭文件资源，在循环内进行数据写入。

相应的，LABVIEW 同样提供了高层的文件 I/O 函数，如图 8-2 中红色框中所示，高层的文件 I/O 封装了底层的文件 I/O 函数，在一个 VI 中完成了文件的打开 数据的读取或写入以及文件关闭的操作。完成一站式的文件操作，但是应当避免把高层文件 I/O 函数放入一个循环结构中，因为反复打开，关闭文件会大大降低文件操作的效率。

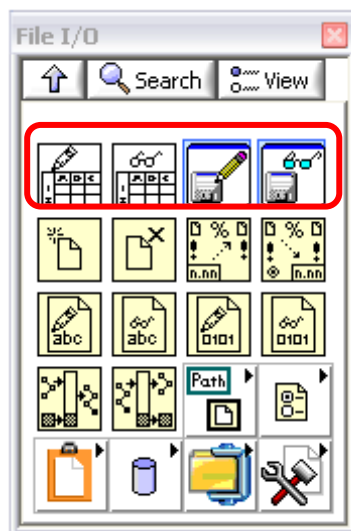


图 8-2 高层文件 I/O

采集波形并存储至 ASCII 文件 DEMO 演示

了解了 LV 中提供的高层和底层文件 I/O 函数，我们就能快速地进行数据的存储操作了，下面我们来看一个典型的 ASCII 文件存储范例。如图 8-3 所示。

这个例子配置了 AI 模拟输入通道进行连续的数据采集，对于 WRITE TO TEXT FILE 进行 ASCII 文件操作，使用了典型的底层 VI 操作流程，首先打开文件，设置文本文件属性，之后再 WHILE 循环内进行数据写入，跳出循环之后关闭文件资源。

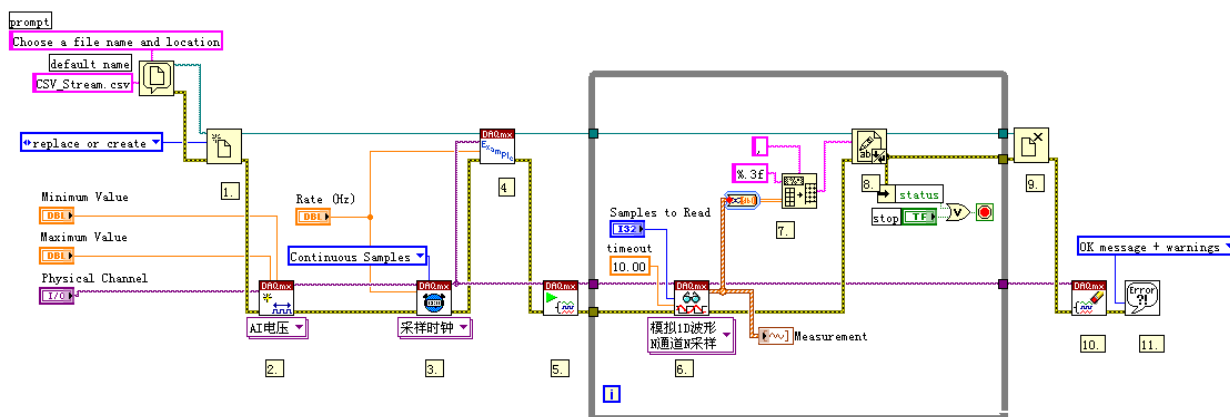


图 8-3 采集波形并存储至 ASCII 文件

我们运行一下这个 VI，选择相应的文件存储的路径，之后就开始数据采集过程。可以在前面板上观察采集到的连续模拟波形，停止采集之后，文件被关闭。

由于写入的是 ASCII 文件，所以我们可以使用 MICROSOFT EXCEL 来打开存储的文件，观察到具体的数据。

四. 回放磁盘上的二进制波形文件 DEMO 演示

如果您的磁盘上已经存储了二进制数据文件，您可以将该数据文件通过 NI 数据板卡的 AO 通道进行输出，下面我们来看一个信号生成的范例。如图 8-4 所示。

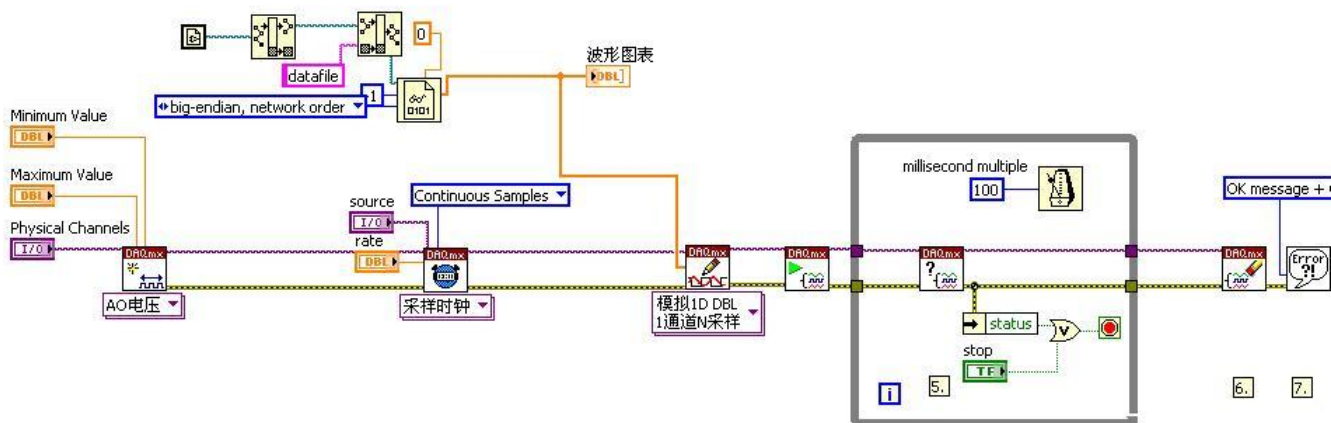


图 8-4 AO 输出二进制波形文件

在这个例子中，我们先前就在硬盘上存储了一个二进制的三角波形文件，该文件与这个模拟输出程序位于同一目录下，程序中我们首先建立 AO 输出通道，设定为连续波形输出模式。将从文件中读取到的二进制数据写入模拟输出通道，开始任务后，进行波形的循环输出。

为了演示这个程序，我们将二进制信号波形通过 M 系列板卡的 AO0 输出，在 BNC2120 上将 AO0 通道与 AI1 通道连接起来，如图 8-5 所示。可以简单地在 MAX 中通过测试面板在 AI1 上观察 AO0 的信号输出。



图 8-5 通过 BNC2120 上将 AO0 通道与 AI1 通道连接

TDMS 初探

除了普通的 ASCII 文件和二进制文件，NI 提出了一种针对测试测量应用的高效数据存储格式，我们称它为 TDMS，下面让我们了解一下什么是 TDMS，以及如何使用这一类型的文件格式。

为了简化设计和维护自己定制的数据文件格式，NI 提出了一种灵活的数据模型称为 TDM，他可以在 NI labview CVI 以及 DIAdem 中进行访问，如果需要从第三方软件中访问 TDM 数据模型，只需要使用我们提供的相应 TDM DLL 即可。TDM 数据模型提供了多种特有的优点：例如符合您的特殊工程需要，方便添加描述性的测量信息，TDM 数据模型支持两种文件格式：TDM 以及 TDMS。

位于文件 I/O-》TDM 流子选板下面的一组 API 用来访问 TDMS 文件类型并针对数据流盘

进行了优化。另外一种数据类型 即 TDM 文件 使用文件 I/O-》存储子选版下的函数来进行访问，同样提供了类似的性能，但并没有对数据流盘进行优化，这里我们着重介绍如何操作 TDMS 文件。

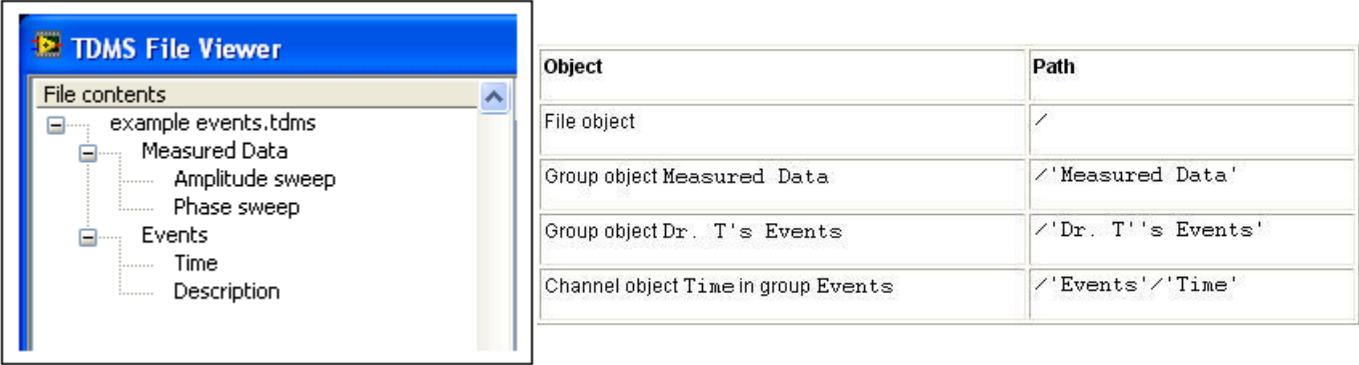


图 8-6 TDMS 中不同层次的对象和结构

当您需要存储测试或测量数据， 为数据分组创建新的数据结构，存储定制数据信息 并进行高速数据读写时，应当考虑使用 TDMS 文件类型。

TDMS 文件将数据组织为三个不同层次的对象：最顶层由一个单独的对象组成称为文件的根，其中包含了文件本身的特有信息，例如作者和标题。每个 TDMS 文件都能包括无限个组，而每个组可以包含无限个通道。

图 8-6 中给出的 example events.tdms 文件包含了两个组， 每个组中又各包含了两个通道。简而言之 我们需要记住其三层结构对象为：

- 文件的根
- 组
- 通道

每一个 TDMS 对象都由一个路径来唯一地标识， 每个路径则是一个包含对象名称以及其所有者名称的字符串，并以正斜杠作为分隔符，每个名称均被包含在一对单引号中，如果对象名称中就含有单引号那么他们将被直接替换为两个单引号符。图 8-6 中的列表给出了 TDMS 中不同对象的路径格式实例。

LabVIEW 中 TDMS 文件 API

常用的 TDMS 文件 API 如图 8-7 所示。

每个 TDMS 对象同时可以拥有无限个属性。每个 TDMS 属性则由一个字符名称，类型标识符以及一个二进制值组成。属性的典型数据类型包括数值类型，时间标志，字符串等等。TDMS 属性不支持数组或复杂数据类型。

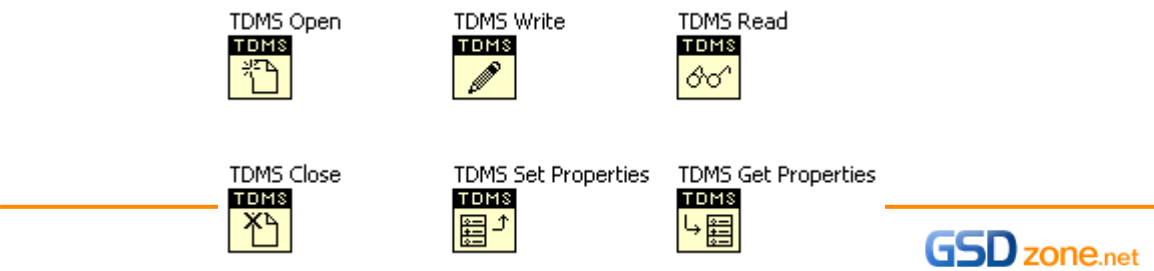


图 8-7 LabVIEW 中常用 TDMS 文件 API

TDMS 具有一个以 TDMS_Index 作为扩展名的二进制索引文件。这个索引文件包含了块数据文件中包含的属性以及指针，以加快数据读取时的访问速度。如果索引文件丢失了它可以自动得到重建。TDM 数据模型文件包含多个组，每个组又包含了多个通道，用户可以在任一层中插入定制的属性。

TDMS 文件中的描述性信息在无需设计一个定制文件头结构的情况下，提供了一种方便的存档方式，当我们的文档需求变得更为复杂时，无需重新设计我们的应用程序，只需要扩展 TDMS 数据模型来满足我们的特定需求即可。

与其他文件 I/O 函数相类似，使用 TDMS API 时同样需要打开 写入以及关闭文件。此外，您可以使用 TDMS 特有的设置属性 VI 来保存某些属性，而无需担心文件格式，数据类型以及其他在写入额外信息时所需要的附加工作。

使用 TDMS 读写

我们来看一个最为简单的写入 TDMS 文件的例子，图 8-8 左半部分的程序中 DAQ 助手产生的数据被送到 WRITE DATA vi 当中。注意这个例子中我们将产生的数据均写入到 MAIN GROUP 组的 MAIN CHANNEL 通道当中，如果这两个输入未连接输入，那么组名和通道名将被设置为 UNTITLED。

右半部分的程序则完成了读取 MAIN GROUP 组中 TDMS 数据的读取，只要设置相应的数据组名就能轻松地读取组中的全部数据。

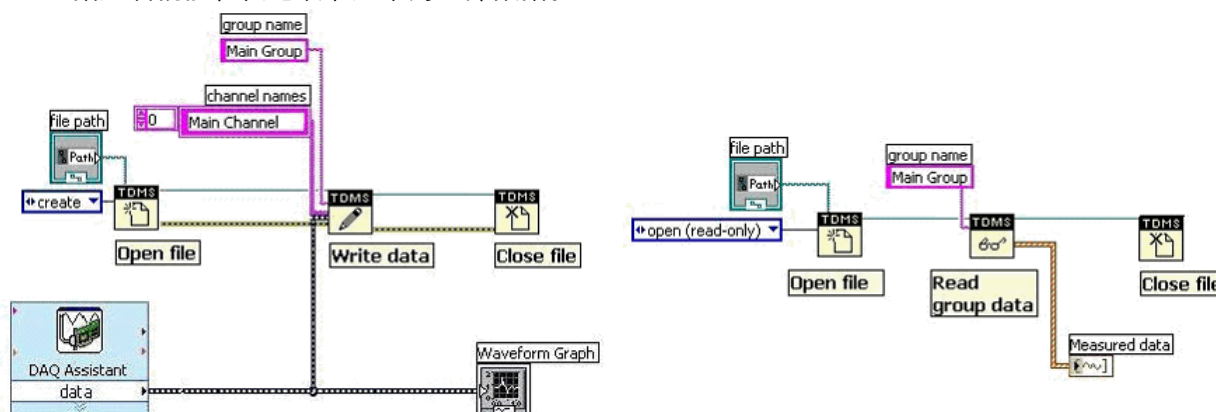


图 8-8 使用 TDMS 读写

我们可以对文件、组以及通道这三个层次进行任意的属性定制，如果需要设置文件层的属性，则无需连接组名称和通道名输入端。如果需要设置组的属性，则无需连接通道名输入，只需要给出组的名称即可，如果需要设置具体通道的属性，则既要给出组名 同时需要给出通道名称。我们注意到属性的值不仅可以是字符串，同样可以是数值类型，如图 8-9 所示。

在进行 TDMS 属性读取时的操作也相类似，当读取某个层的属性时只需要给出相应的组名

以及通道名即可，如图 8-10 所示。

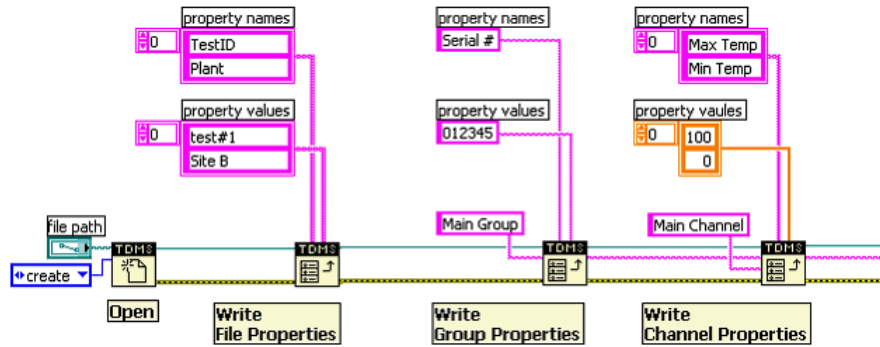


图 8-9 TDMS 的任意的属性定制

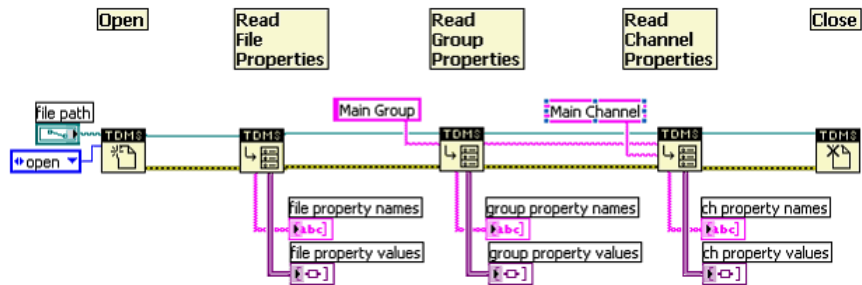


图 8-10 TDMS 的任意的属性读取

采集数据并录入 TDMS 文件的 DEMO 展示

下面我们就来看一个典型的将采集到的数据写入 TDMS 文件的例子，在采集结束后我们将使用上图中的 TDM 流文件查看器来浏览数据的详细信息。

这个例子配置了 AI 模拟输入通道进行连续的数据采集，对于 TDMS 文件操作，使用了典型的底层 VI 操作流程，如图 8-11 所示：

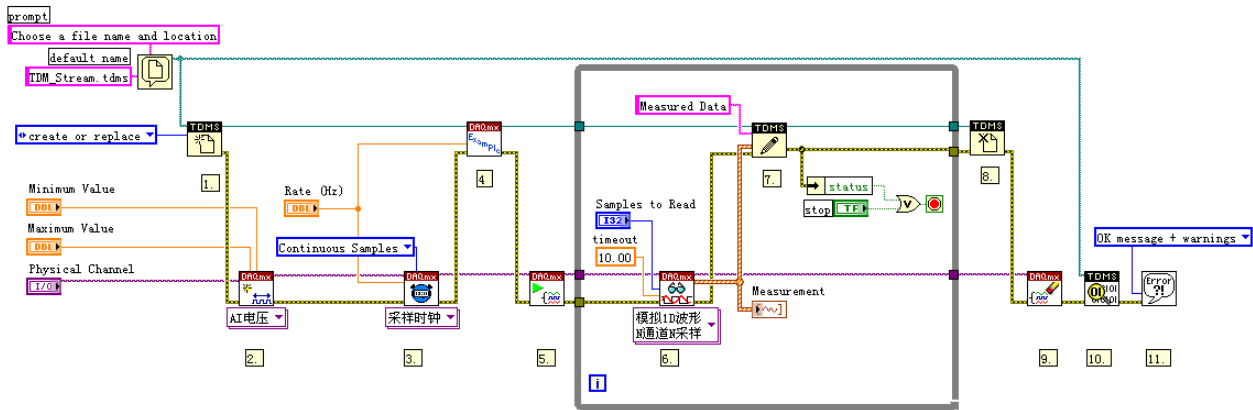


图 8-11 采集数据并录入 TDMS

首先打开 TDMS 文件，设置 TDMS 属性，之后再 WHILE 循环内进行数据写入，跳出循环之后关闭文件资源，并自动调用 TDM 流文件查看器来浏览数据。

我们运行一下这个 VI，选择 TDMS 文件存储的路径，之后就开始数据采集过程。可以在前面板上观察采集到的连续模拟波形，停止采集之后，文件被关闭，并自动打开 TDM 流文件查看器，在文件查看器中可以浏览具体的对象属性，详细的数据，以及直观的数据波形。如图 8-12 所示。可见使用 TDMS 文件格式来存储测试测量数据，可以有效地进行数据的组织与合并，创建新的数据结构，并存储定制数据信息。

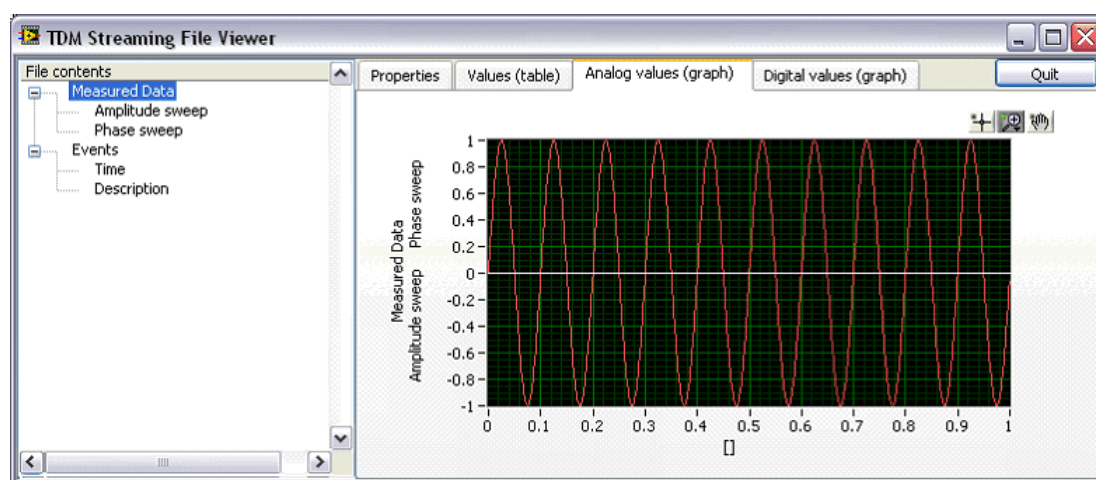


图 8-12 TDM 流文件查看器

同步（上）

简介

本期节目介绍：如何实现 NI 数据采集板卡的多功能同步功能。

在许多应用中需要在同一时间段内进行多种不同信号的测量，同步信号采集大致可以分为两大类：

- 一类我们称之为同时测量，即不同的操作开始于同一时刻：
比如说在一个输入通道上采集数据，同时在一个输出通道上产生信号 然而这两者完全可能是并不相关的，也就是说即使两者在同一时刻开始，但他们可能具有各自独立的采样率和更新率。

- 另一类则称为同步测量，所有的测量通道会共用一个时钟信号并在同一时刻开始：
例如同步测量汽车的速度以及轮胎上的温度号。在同步测量当中，又可以分为多功能同步测量以及多设备同步测量。这一期中，我们会着重讨论同时测量以及多功能同步测量的内容。

下面我们详细了解一下怎样实现一个同时测量的例子。

同时模拟输入与模拟输出

要实现两个任务的同时开始，最简单的方式就是使用同一个 START TRIGGER，如图 9-1 所示：

首先，我们分别建立了模拟电压输入和模拟电压输出通道，在 DAQmx 定时 VI 中，他们均被设置为连续的采样模式，并独立设置各自的采样和更新率。在上半部分的 AI 通道中，通过调用获取带有设备前缀的终端名称这样一个 VI 来获取 AI 通道开始触发信号的资源名称，并将这个输出连接到模拟输出通道的触发输入源上，这样，我们就完成了模拟输出通道共享模拟输入开始触发信号的配置。

在开始任务部分，需要注意的是，我们必须事先在模拟输出通道上调用 DAQMX 开始 VI，以保证输出通道早于输入通道进入运行就绪状态，之后才调用输入通道上的 DAQMX 开始 VI。

因为开始触发信号是从输入通道上发出的，所以仅当输出通道首先就绪之后，输入通道开始触发才能保证两者同时开始。这就是一个典型的进行同时测量的例子。

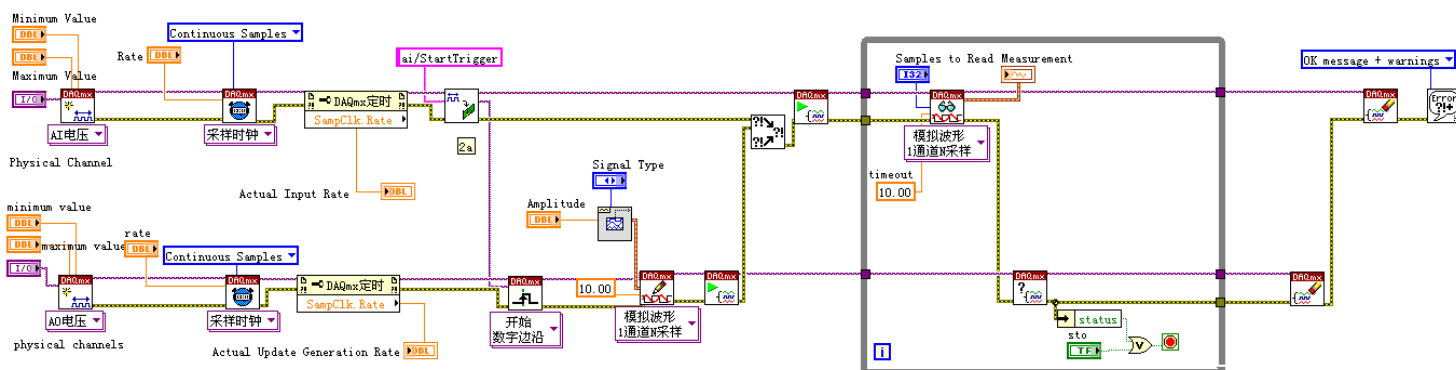


图 9-1 同时模拟输入与模拟输出

多功能同步测量

在刚刚的例子当中我们看到了使用同一个开始触发可以使得两个任务同时开始，那么如果要进行完全意义上的同步测量，应该怎么做呢？

下面我们来看一个例子。在这个例子中我们将会看到如何实现一个多功能同步测量，所谓多功能同步测量，是相对于多设备同步测量而言的，也就是说在同一个设备上完成不同功能模块间的同步。如果要做到完全意义上的模拟输入和模拟输出同步，我们有两种方法，第一种方法：我们需要让多个模块共享同一个时基，一旦时基共享了，那么由时基产生的采样时钟，更新时钟，转换时钟都能够达到严格的同相，除此之外我们还需要共享一个开始触发信号，来使不同的任务同时开始。第二种方法则是直接共享同一个采样时钟。

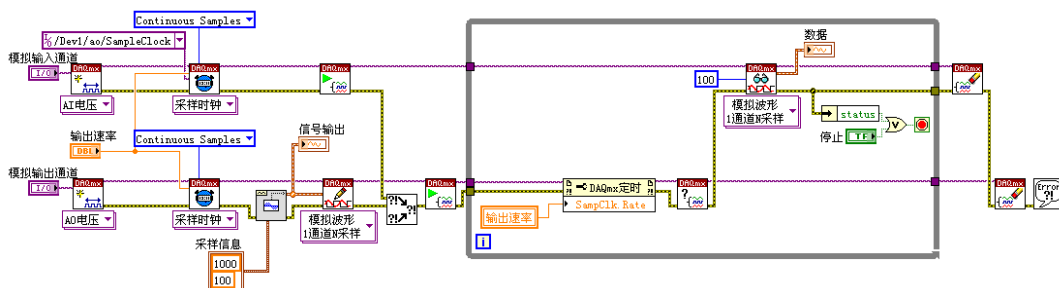


图 9-2 AI AO 多功能同步测量

从图 9-2 的程序中，我们看到在模拟输入通道的 DAQmx 定时部分，我们将采样时钟源设置为模拟输出的采样时钟，如此一来两者使用的是同一个采样时钟源，因此就可以做到完全意义上的同步操作了。

AI 与 DI 同步测量

与模拟输入输出相类似的，我们再看一个数字输入共享模拟输入采样时钟进行同步的例子：

如图 9-3 所示

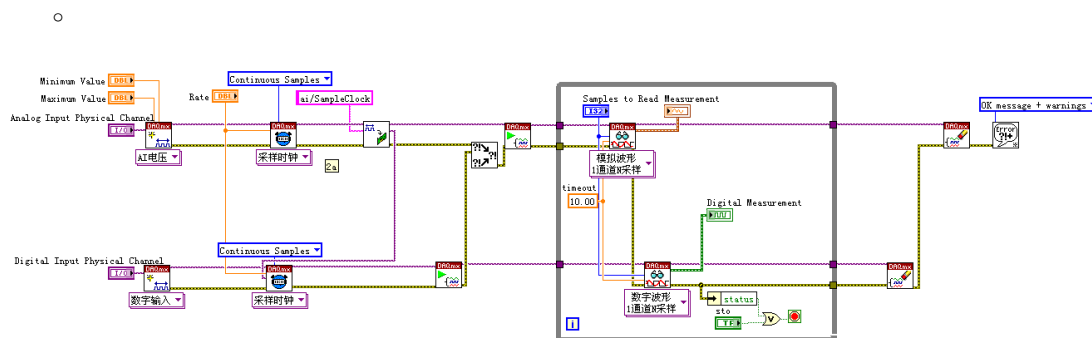


图 9-3 AI 与 DI 同步测量

首先我们创建一个模拟输入通道，同时创建一个数字输入通道

其次，为模拟和数字输入设置相同的采样速率，并设置采样模式为连续采样。之后我们同样调用了获取带有设备前缀的终端名称 VI 来获取我们模拟输入通道采样时钟信号的终端名称，并将这个输出连接到数字输入通道的采样时钟源的输入端口上，这样，我们就完成了数字输入通道共享模拟输入采样时钟的配置。接下来，通过调用 DAQMX 开始 VI 来开始模拟与数字采集，这里需要注意的是，我们必须保证数字输入通道的开始先于模拟输入通道，这是因为，数字通道的采样时钟来自于模拟输入通道，先让数字输入通道处于 running 状态并不会马上获得数据，只有等到模拟输入通道开始采集并出现有效的采样时钟，共享给数字输入通道后，两者才同时开始采集。试想一下，如果我们调用模拟通道的 DAQMX 开始 VI 先于调用数字通道的开始 VI，那么这两个输入通道就无法达到同步了。

DI,DO 与 CO 同步测量

除了模拟输入输出以及数字输入功能之外，我们如何使用多功能数据采集板卡上的计数器来完成同步功能呢？下面我们再来看几个有趣的范例。

在这个例子中，我们将完成数字输入，数字输出通道共享采样时钟与计数器输出通道同步的功能，如图 9-4 所示。

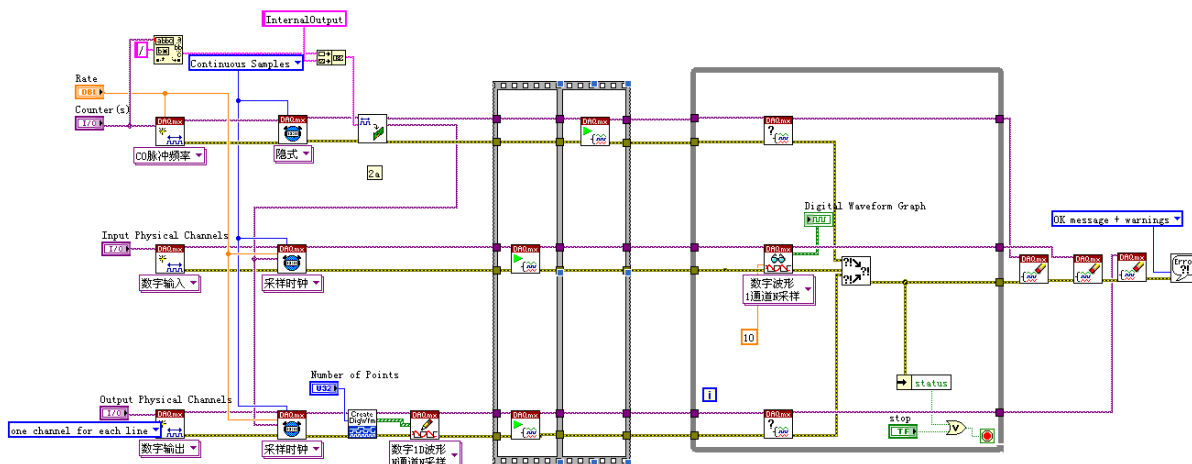


图 9-4 数字输入，数字输出通道共享采样时钟与计数器输出通道同步

首先我们创建一个数字输入通道，一个数字输出通道，以及一个计数器频率脉冲输出通道。与先前的例子相类似，我们通过调用获取带有设备前缀的终端名称 VI 来获取计数器频率脉冲输出通道内部输出信号的资源名称。这个计数器的内部输出将被路由到数字输入和数字输出通道的采样时钟上，我们看到 在数字输入和数字输出通道中，通过调用 DAQmx 定时 VI 为数字通道配置了采样时钟源，该采样时钟源即为我们刚刚配置好的计数器频率脉冲输出信号。之后为了保证三者能够保持同步，我们使用了顺序结构来保证数字输入和数字输出通道先于计数器频率脉冲输出通道运行，这样一个典型的多功能数字同步任务就完成了。

AI 与 CO 同步测量

相类似的，这是一个计数器输出连续脉冲 作为 AI 的 sample CLK 完成 CO 与 AI 同步的例子，如图 9-5 所示：

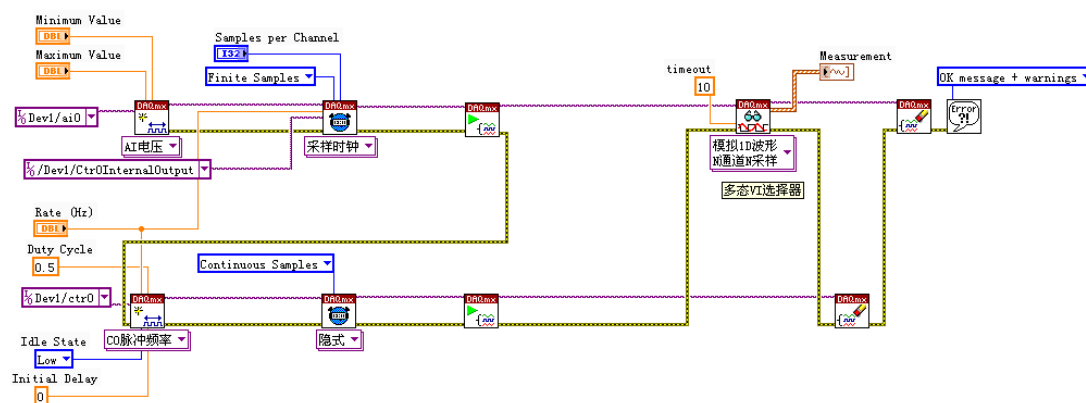


图 9-5 AI 与 CO 同步测量

首先我们在 AIO 创建了一个模拟输入电压通道，同时并行地建立一个计数器输出通道来产生脉冲序列输出，将空闲状态设置为低电平，也就是说脉冲输出的第一个边沿是由低到高的电平转换。

其次，将 AI 通道的采样时钟源设置为计数器的内部输出。同时定义了 AI 通道为有限点采集，计数器输出通道为连续脉冲序列输出。

之后，调用 DAQmx 开始 VI 来开始 AI 采集，需要注意的是，在这个阶段 AI 通道上并没有数据会被采集进来，原因是它还没有得到有效的采样时钟输入，直到使用了 DAQmx 开始 VI 使得计数器开始输出脉冲序列之后，AI 通道才得到有效的采样时钟并进行有限点的数据采集，实现 AI 与计数器输出的同步。

实现 AI 的可重触发

在某些情况下，我们可能会需要完成带有可重触发功能的模拟信号采集，但是我们的模拟

输入通道并不具有可重触发的功能，怎样才能实现这一功能呢，借助计数器，我们就能轻松解决这样的应用。

在这个例子中，我们使用计数器输出作为模拟输入采样时钟来完成可重触发模拟输入采集的功能，如图 9-6 所示：

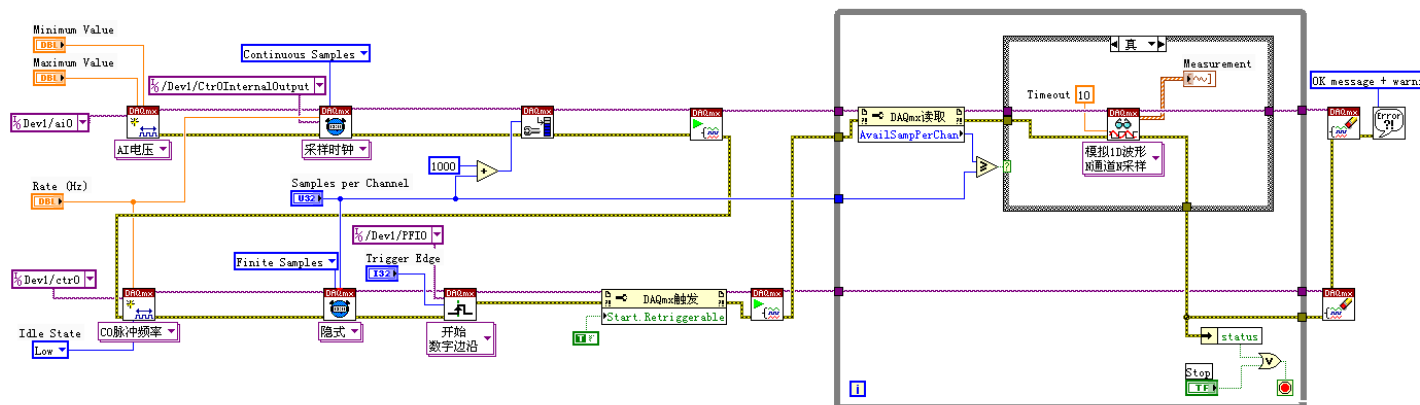


图 9-6 实现 AI 的可重触发

首先 我们在 AI0 上创建一个模拟输入电压通道， 同时并行地在计数器 0 上创建一个计数器脉冲频率输出通道。

其次，我们为计数器 0 脉冲输出配置定时，因为是计数器脉冲输出，所以我们只需要设置定时为隐式即可，同时，这个例子中我们希望每触发一次就采集 N 个采样点，于是将计数器输出脉冲定时设置为有限点模式，并在每通道采样输入端给出每次触发需要采集的采样点数， 之后为了启用计数器独有的可重触发功能，我们将 DAQmx 触发属性中的可重触发输入设置为真！ 在配置完计数器通道后，我们对 AI0 模拟输入通道进行相应的 DAQmx 定时配置，将刚刚配置好计数器 0 内部输出作为模拟输入 AI0 的外部采样时钟源，这样每当计数器输出脉冲时，模拟输入通道 0 就会得到有效的采样时钟输入，进行 N 个点的数据采集。在计数器的触发源配置部分，我们选择 PFI0 引脚来作为外部触发信号源。

在上半部分，程序对模拟输入的缓冲区大小进行设置，这里设置为每通道采样数加上 1000 个采样点，其原因在于通过 DMA 传送数据的时候，在硬件上需要一个略大于每通道采样点数的缓冲区来进行有效的数据传输。

之后我们通过首先调用 DAQMX 开始 VI 来开始 AI 采集，与之前的例子相类似，此时因为没有有效的采样时钟出现，所以 AI 通道上并不会采集数据，只有当有效的触发出现在 PFI0 上之后，计数器 0 开始产生有限脉冲输出 提供给 AI 通道作为采样时钟，AI0 才进行每次 N 个点的有限点采集。

完成了同步配置之后，在 while 循环当中，我们使用 DAQ 读取属性节点中的每通道可用采样属性来获取当前缓冲中可用而未被读取的采样数，一旦缓冲当中的可用采样点数等于或超过之前设定的每通道采样数，则使用 DAQMX 读取 VI 将有效数据读取出来，否则则继续 WHILE 循环。

在程序的最后，通过调用清除任务 VI 来释放资源，并使用简易错误处理器来显示可能出现的错误。这样，一个可重触发模拟输入应用就完成了。

同步（下）

简介

本期内容介绍：如何实现 NI 数据采集板卡的多设备同步功能。

在许多应用中需要在同一时间段内进行多种不同信号的测量，同步信号采集大致可以分为两大类：

- 一类我们称之为同时测量，即不同的操作开始于同一时刻：
比如说在一个输入通道上采集数据，同时在一个输出通道上产生信号 然而这两者完全可能是并不相关的，也就是说即使两者在同一时刻开始，但他们可能具有各自独立的采样率和更新率。上一期中我们曾举过一个同时进行 AIAO 的例子。

- 另一类则称为同步测量，所有的测量通道会共用一个时钟信号并在同一时刻开始：
例如同步测量汽车的速度以及轮胎上的温度号。 在同步测量当中，又可以分为多功能同步测量以及多设备同步测量。上一期中，我们着重讨论了同时测量以及多功能同步测量的内容。

这一讲中我们将着重介绍多设备同步的应用。

多设备同步基本要素：

下面我们就来看一看进行多设备间同步时所需要的基本要素：

如果要做到不同对象间的同步，

- 第一，我们可以通过共享主时基（或者称之为参考时钟）并共享一个开始触发来实现同步。
- 第二，则可以通过直接共享采样时钟来完成同步。

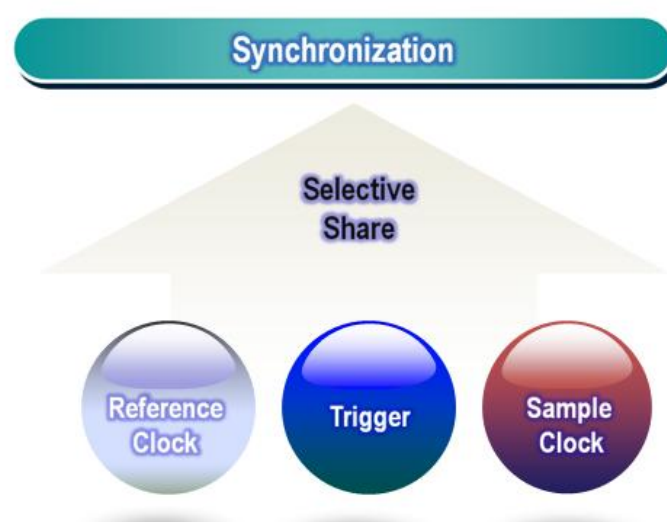


图 10-1 不同对象间的同步的要素

在共享一个主时基的情况下多个时钟都来自同一个时基。对于多个板卡来说,这样就避免了相位误差。因为共享了时基 那么不同的采样时钟就可以设置成不同的采样率,而且所有任务都有开始触发,如果触发信号未被显示的创建,则它会在任务开始后被创建。

而对于直接共享采样时钟进行同步的情况,则多个对象总是使用相同的采样率来完成同步。

M 系列板卡定时引擎回顾

在回顾了同步的规则之后,我们再简要地浏览一下第七讲中提到过的 M 系列定时引擎,如图 10-2 所示:

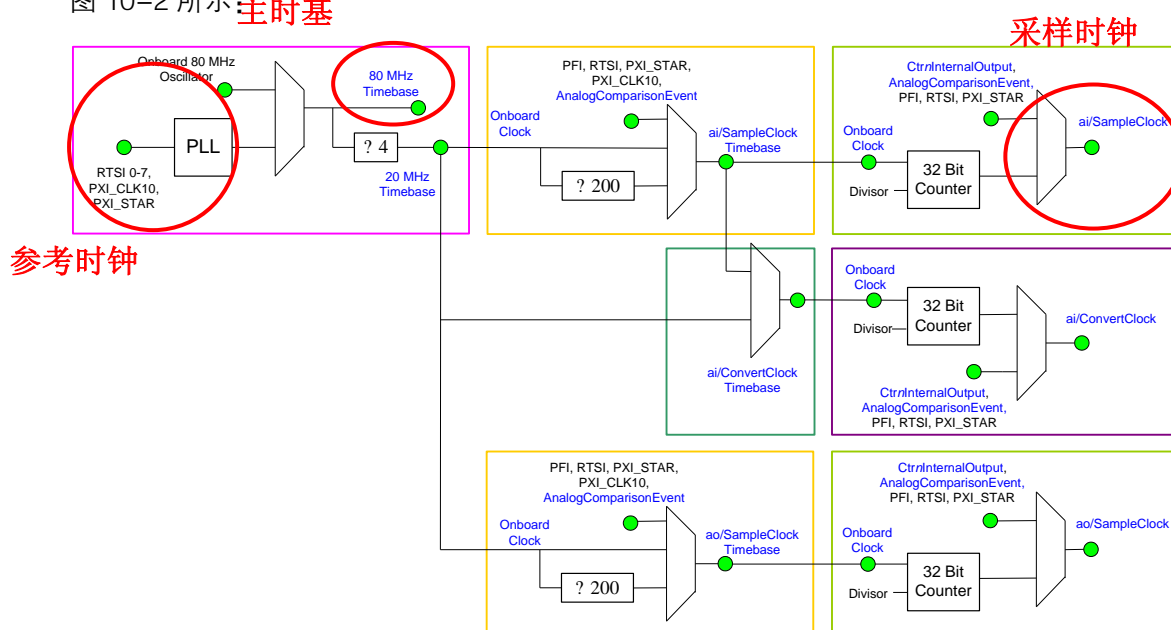


图 10-2 M 系列定时引擎回顾

在这里我们可以看到参考时钟主时基以及采样时钟在整个定时引擎中所处的位置,采样时钟可以通过 PFI 可编程多功能接口, PXI-STAR 或者是 RTSI 引入,主时基参考时钟则可以是 PXI 背板上的 10MHZ 时钟, PXI-STAR 或者是 RTSI 之一。

关于 RTSI

这里我们反复提到 RTSI 接口,那么到底什么是 RTSI 呢?

RTSI 我们称之为 实习系统集成总线。用来在不同设备之间传递触发以及控制信号,对于 PCI 平台来说,RTSI 被放置在板卡后部的上端,以 34 针引脚引出,为了让两块板卡同步,我们需要使用一根 34 针 RTSI 电缆来连接两者。

而对于 PXI 平台来说,所有的信号都可以从 PXI 背板上进行路由,于是就不需要额外的连线了。下面我们来了解一下 RTSI 总线的具体配置:

实时系统集成总线(RTSI)，可以传输高达 20MHZ 的时钟信号，这一特性使得它非常适合于高速的应用。在 DAQmx 编程中，RTSI 总线的管理是不对用户公开的，DAQmx 驱动在底层帮助用户完成了大部分的信号路由工作，然而用户必须告诉驱动程序，各个设备之间是怎样通过内部总线相连接的。对于 PCI 系统，我们必须在 MAX 当中手动注册用于连接不同设备的 RTSI 线缆，如图 10-3 所示。

对于 PXI 系统，我们必须告诉驱动程序当前正在使用的 PXI 机箱型号。如图 10-4 所示。

在配置完 RTSI 之后，我们需要明确 DAQmx 中进行多设备同步编程的规则。首先，我们会建立一个主任务进行输入类型，定时信息，时基参考，开始触发等设置，但不立即开始主任务。其次建立必要个数的从任务，类似地进行相应的配置，之后先开始从任务，再开始主任务。对于开始任务先后次序的配置与上一讲中多功能同步相类似，这里就不加赘述了。

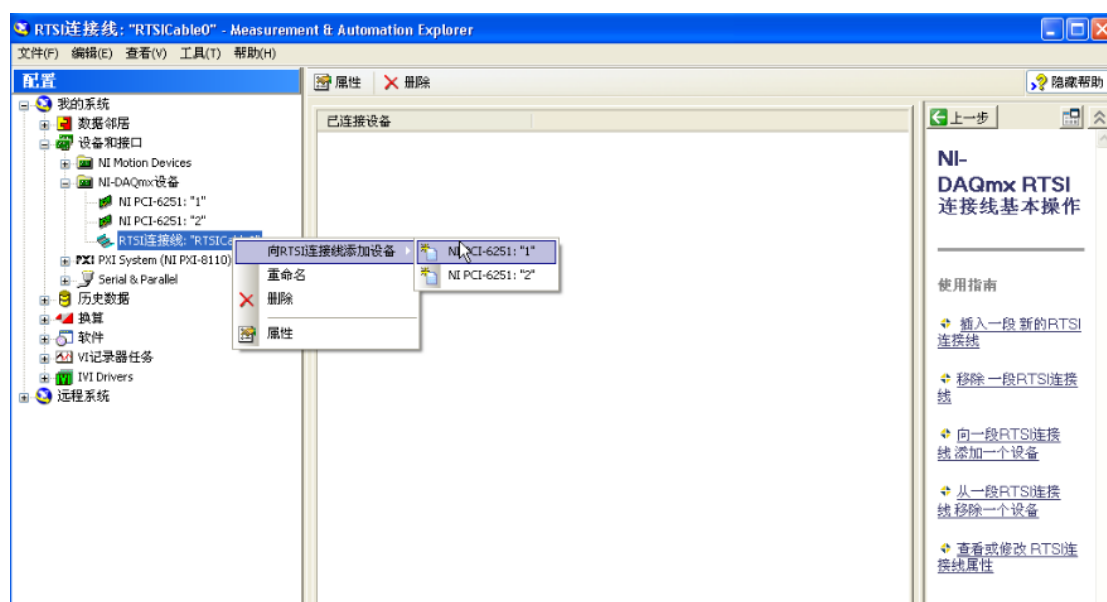


图 10-3 在 MAX 当中手动注册用于连接不同设备的 RTSI 线缆

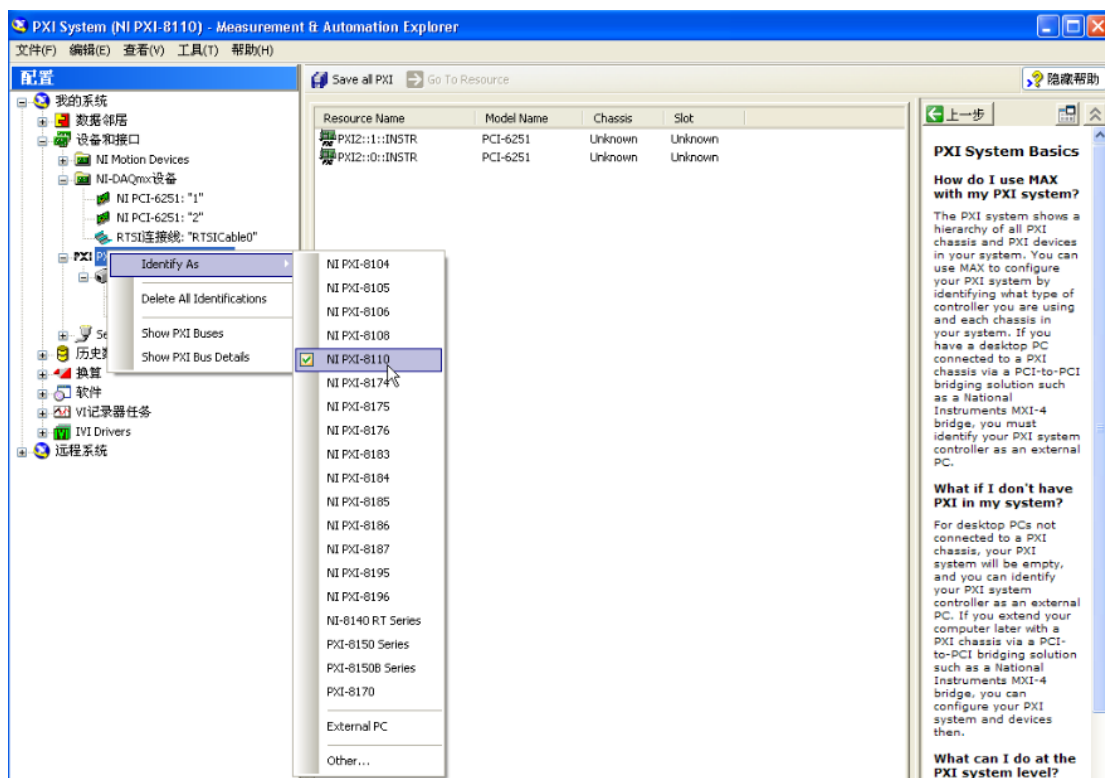


图 10-4 配置当前正在使用的 PXI 机箱及控制器

下面我们就来讨论一下几种常见的多设备同步案例

M 系列板卡共享参考时钟同步

图 10-5 的下方是使用 DAQmx 驱动编写的，使用两块 M 系列板卡进行同步的范例程序，图 10-5 的上方则是相应的定时引擎路由情况。

我们看到程序中首先分别建立了两组模拟输入通道，上面的通道为主设备，下面的则为从设备，并将他们都设置为连续采样模式，使用相同的采样率配置。之后通过使用 DAQmx 定时属性节点来共享 PXI 背板上的 10MHz 参考时钟，然后将从设备的开始触发输入配置为主设备的 AI START TRIGGER，并按照之前我们所说的规则先开始从任务再开始主任务，以完成整个同步。

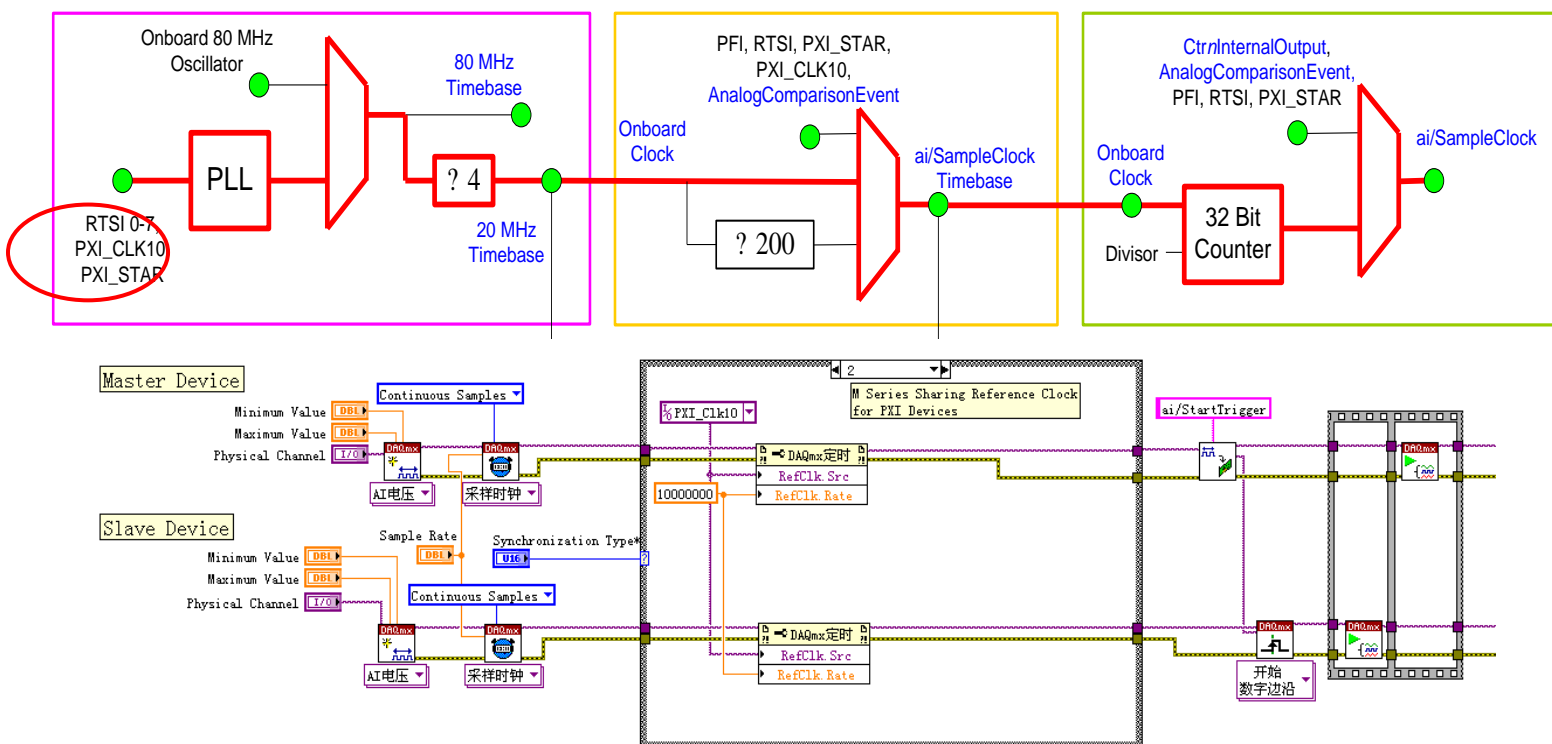
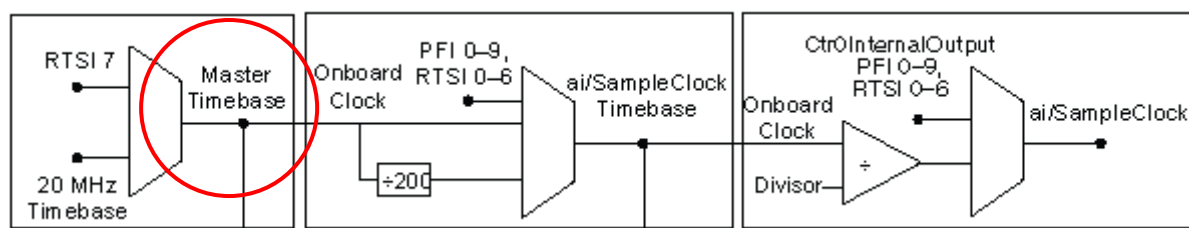


图 10-5 M 系列板卡共享参考时钟同步

从定时引擎路由情况来看，模拟输入的采样时钟，由 PXI 背板 10M 时钟作为参考源，输入锁相环倍频后再分频为模拟输入采样时钟的 20M 时基，并经过内部分频器转化为所设定的采样率，整个路由情况一目了然。

E/S 系列 共享主时基同步

在了解了 M 系列多功能数据采集板卡的定时引擎以及同步机制之后，要进行 E 系列和 S 系列板卡的同步也就变得非常简单易懂了，图 10-6 的例子通过共享主时基，完成了两个设备之间的模拟输入同步。



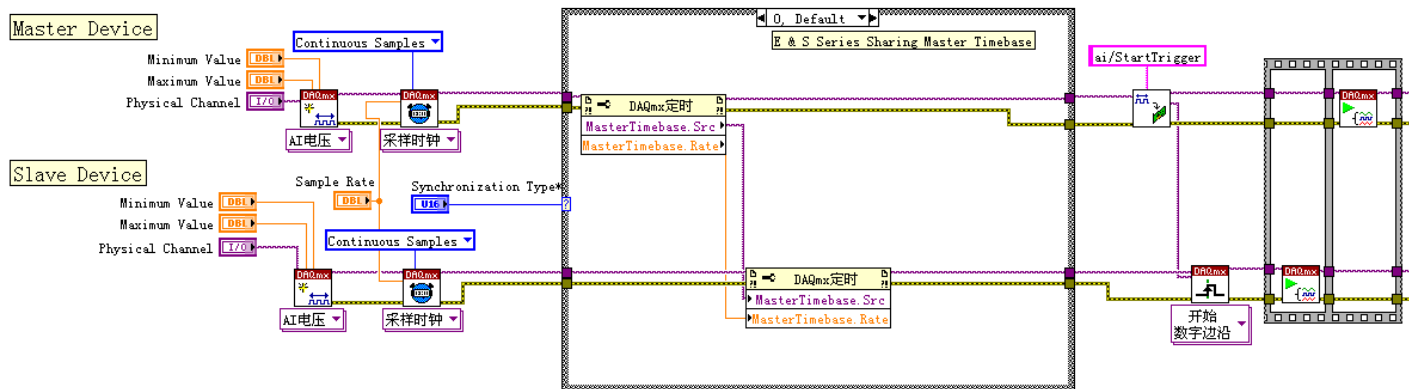


图 10-6 E/S 系列 共享主时基同步

M/E/S 系列共享采样时钟

图 10-7 的例子则给出了通过共享采样时钟来同步 NI M 系列 E 系列或 S 系列的数据采集板卡的模拟输入通道。同样，程序的上半部分是主设备，下半部分是从设备，从程序上来看，主设备 DAQmx 定时设置部分的采样时钟源没有连接任何输入，于是驱动就默认使用板上的 80M 时基作为参考时钟，注意图中上半部分定时引擎的路由情况，从板上 80MHZ 时基出发，经过一系列分频后作为主设备的采样时钟，而从设备的 DAQmx 定时设置部分选择了主设备的 AI SAMPLE CLK 作为自己的采样时钟源输入，其相应的定时引擎路由见图 10-7 底部的红色部分，从而共享了主设备的采样时钟，之后通过先开始从任务，在开始主任务来完成不同设备间的同步。

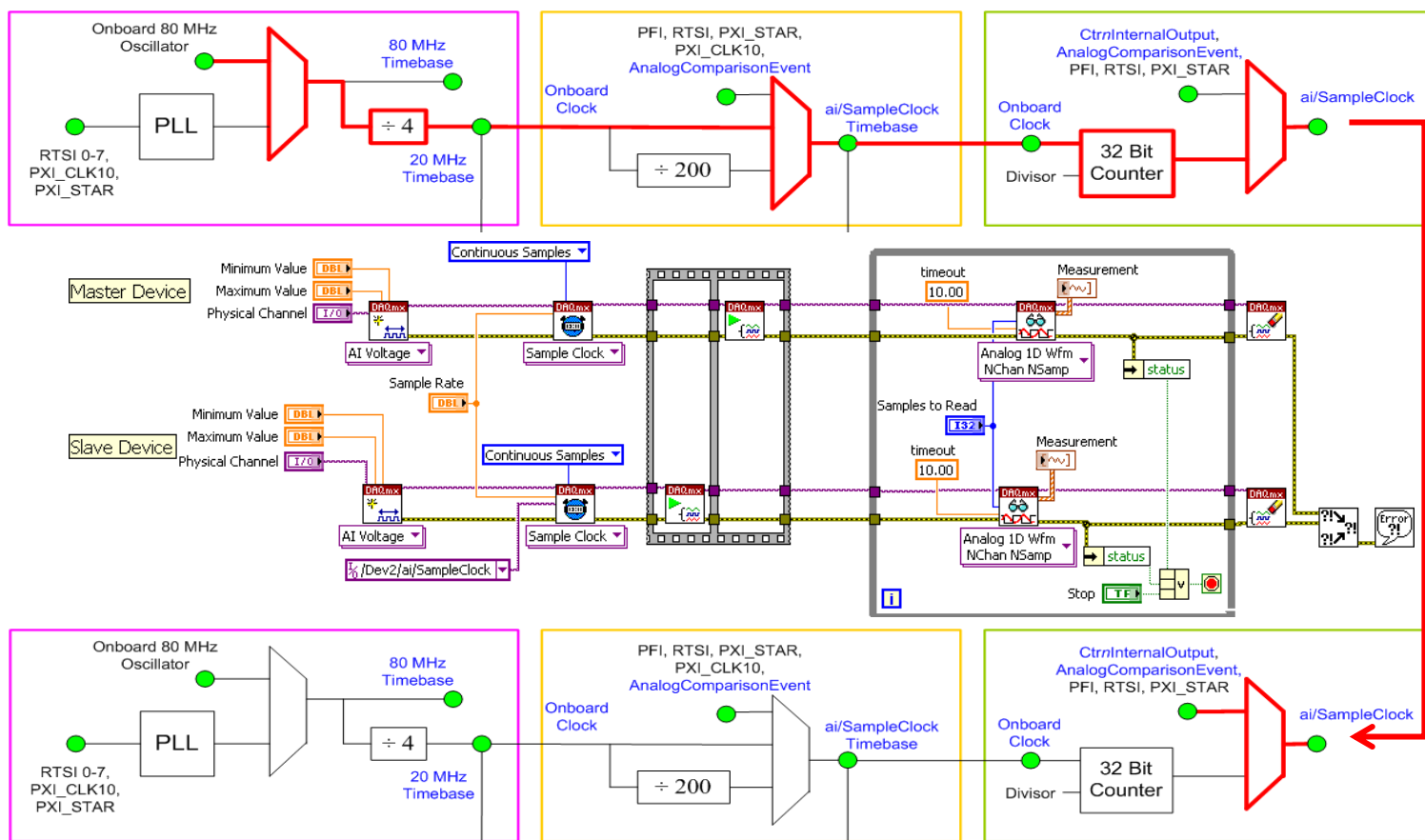


图 10-7 M/E/S 系列共享采样时钟同步

E/S 系列板卡 与 M 系列板卡间同步

相对于新推出的 M 系列数据采集板卡，E 系列板卡在业内早已具有良好的口碑，并占据着市场的主导地位，所以同步使用 E 系列和 M 系列板卡变得非常重要，这里我们给出一个 NI 推荐您使用的同步 E 系列和 M 系列板卡的范例，如图 10-8 所示。

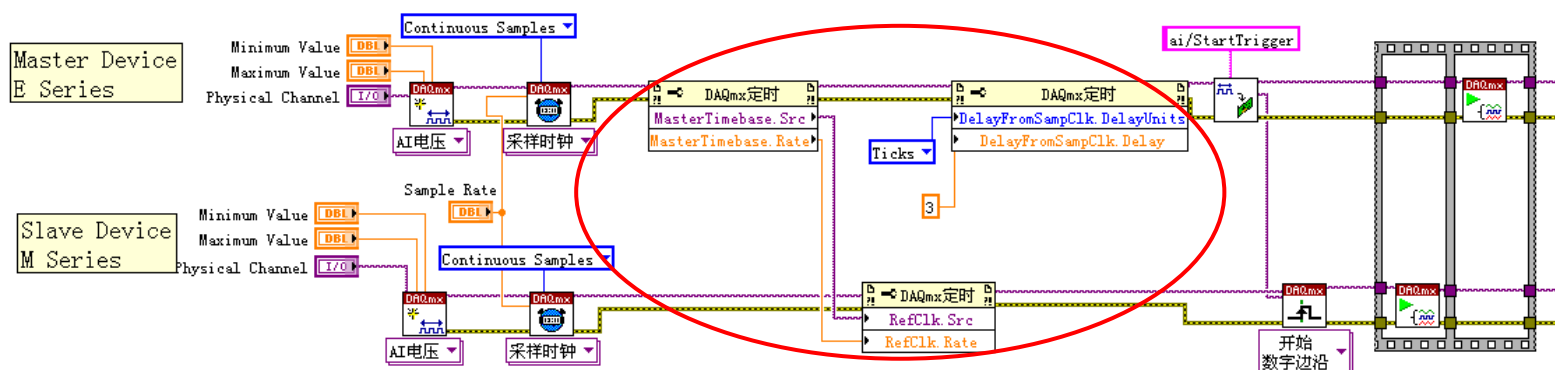


图 10-8 E/S 系列板卡 与 M 系列板卡间同步

E 系列板卡和 M 系列板卡在同步方面使用的技术有所不同。当需要同步多个 E 系列板卡是，一个设备会将自己的 20MHZ 主时基信号导出，供给给其他设备来进行同步，这一方法正是我们刚刚所介绍的第二种情况。 尽管一块 E 系列板卡可以导入一个更低频率的信号作为主时基来进行同步，例如 导入 10Mhz 的背板信号，但是它无法倍频导入的时基来重建一个 20MHZ 的时基，因此，如果使用 10Mhz 作为导入时基的话，那么板卡内部基于这个时基的其他采样时钟的分辨率就下降了。

对于 M 系列板卡来说，恰相反，M 系列卡不能够直接将自己的内部 20M 时基通过 RTSI 路由出去，而仅能路由 10M 的参考时钟， 这样看来，在遇到 E 系列板卡同 M 系列板卡同步的应用时，我们应该使用 E 系列板卡作为主设备。 这样 M 系列板卡作为从设备可以使用由 E 系列板卡路由出来的 20Mhz 主时基作为自己的参考时钟输入。于是 M 系列板卡上的时基将与作为主设备的 E 系列板卡上的 20M 主时基同相。在程序中只要将 E 系列板卡的主时基源连接到 M 系列板卡的参考时钟源上即可。

除了主从设备的选择之外，另外一个需要注意到的区别是，M 系列和 E 系列板卡的默认采样时钟延迟有所不同。 采样时钟延迟就是 AI 采样时钟与第一个 AI 转换时钟脉冲之间的间隔。对于 E 系列板卡来说，这个默认间隔为主时基的两个时钟滴答， 对于 M 系列板卡来说这个值为当前主时基的 3 个时钟滴答。这个差别可以从图 10-9 中清楚地看到。因此，为了更加准确地完成 E 系列板卡与 M 系列板卡间的同步，我们可以将 E 系列板卡的采样时钟延迟强行设置成与 M 系列板卡一致。 这一配置在程序中通过设置相应的 DAQmx 定时属性节点来实现。

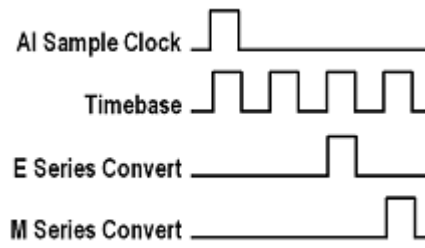


图 10-9 采样时钟延迟

程序的后续部分依旧遵循先开始从任务，后开始主任务的顺序完成同步。

特别篇：模块化仪器

简介

之前我们通过 NI 数据采集技术十讲，为大家全面介绍了 NI 的数据采集硬件产品，以及如何使用 LabVIEW 编程，加速数据采集项目的开发。节目播出后，我们收到许多工程师的反馈，其中有工程师提到：他们有一些参数要求非常高的数据采集应用，希望得到产品选型方面的建议。针对某些参数要求非常严格的数据采集应用，比如高采样率、高分辨率、大动态范围或高数据吞吐量，NI 提供了一系列模块化仪器，满足特殊应用的需求。所以，我们特别增加了本期节目，为大家介绍一下 NI 的模块化仪器平台。

什么是模块化仪器

在演示视频中的 PXI 机箱，里面的每一块板卡，都是一部仪器。常用的传统仪器，有信号发生器、示波器、数字万用表等等，模块化仪器可以实现跟传统仪器同样的功能。

比如在这个 PXI 机箱中，5422 就是一个信号发生器，5124 是一个示波器，4070 是一个数字万用表；除此以外，还有 5652、5661 射频信号发送和测量仪器，6552 高速数字信号发生器和分析仪，4461 动态信号采集卡，4130 源测量仪器，2593 开关矩阵模块等等。

模块化仪器的特点

与传统仪器相比，模块化仪器的体积更小，成本更低，还可以方便地将不同的模块化仪器集成在一起，并实现多通道多设备间的同步。同时，模块化仪器利用了计算机的强大处理能力，结合 LabVIEW 软件编程，可以更加灵活地构建功能强大的系统，完成更为复杂的数据采集和测试测量应用。

与普通的数据采集卡相比，模块化仪器的专用型更强，性能参数更优越。比如，高速数字化仪可以提供高达 2G 的采样率，数字万用表可以提供 26 位的分辨率等等。所以针对特殊的数据采集应用，如果普通数采卡不能满足需求，就需要使用模块化仪器来实现。

NI 模块化仪器类别

NI 提供了八大类模块化仪器产品，以满足从直流到射频信号的数据采集和测试测量应用。这八类产品分别为：

- 信号发生器

它可以生成标准函数或自定义的任意波形，最高采样率 400M Samples/s，最高分辨率 16 比特，并配备有最高 512M 的板载内存，用于大数据量波形序列的输出。

- 高速数字化仪

它是模块化的示波器，也是采样率更高的同步数据采集卡。

它提供最高 2G 采样率，1GHz 带宽，以及 24 bit 的分辨率；与此同时，多通道大板载内存的特性，可确保较长时间的高速同步采集。

- 高速数字 I/O 模块

单个模块上最多 32 个 DIO 通道，可通过软件设置为输入或输出方向；最大 200MHz 时钟频率，工作在双倍数据率 (double data rate) 模式下时，每通道最大传输速率更可达到 400 Mbit/s；支持 TTL 和 LVDS 标准逻辑电平，同时也可通过软件进行不同的逻辑电平的设置。

- 数字万用表 (DMM)

提供最高七位半的精度，以及 10 至 26 比特的可变分辨率，能够精确地测量电压、电流、电阻、电容、电感和温度。特殊的 FlexDMM 架构，还使其具有 1000V， 1.8 MS/s 隔离数字化仪相同的功能，即可以最高 1.8 MS/s 的采样率，对幅度范围 1000V 的模拟输入信号进行数据采集。通过将数字万用表与开关模块集成使用，还可以灵活搭建高通道数的测试测量系统。

- 动态信号采集卡 (DSA)

提供 24 比特的高分辨率和 118 dB 大动态范围，最高采样率 204.8k Samples/s，适用于对声音和振动信号进行多通道同步采集与分析。LabVIEW 还为此类应用提供了专业的声音与振动工具包，该工具包包含了大量常用的声音与振动信号分析和处理函数，可加速实现应用项目的开发。

- 射频模块

射频信号发生器和分析仪，分别配备了高达 6.6 G 的频率范围， 100MHz 和 50MHz 的最大瞬时带宽，以及 512M 和 256M 的板载内存。同时借助 LabVIEW 的调制解调工具包，可以灵活方便地实现各类标准的调制解调应用程序的开发。

- 源测量单元 (SMU)

既可以作为高精度的恒流源和恒压源输出，同时也可以进行高精度的电流、电压和功率的测量。

- 开关

开关模块包括通用继电器、多路复用器以及开关矩阵。配合 NI 其它模块化仪器使用，可满足高精度、高通道数、高切换速率的具体应用需求。对于大型的复杂开关系统，NI 还提供了 Switch Executive 软件，方便进行配置、管理和维护。

模块化仪器相关软件

那么，模块化仪器如何与软件交互实现自动化的数据采集与测试测量任务呢？NI 提供了如下几类工具，如图 1-1 所示。

应用软件
NI LabVIEW

软面板

模块化仪器驱动

模块化仪器硬件

图 1-1 模块化仪器软件体系

首先是驱动软件，就像普通数据采集板卡有 DAQmx 驱动一样，NI 也为所有的模块化仪器提供了仪器驱动，方便应用软件和硬件进行交互，缩短软件的开发周期。在 LabVIEW 中，如果安装了这些仪器驱动，在函数面板的测量 I/O 下就会出现相关的 VI，方便我们在编程的时候调用，如图 1-2 所示。同时，模块化仪器驱动也提供对 VC、VB 等常用编程语言的支持，方便与其它代码进行集成。



图 1-2 模块化仪器驱动

除此以外，为了更加符合工程师们使用传统仪器的习惯，NI 也提供了操作界面与传统仪器非常相似的软面板，无需编程就能配置模块化仪器完成指定的功能。如果需要实现相对复杂的数据采集和测试测量应用，我们推荐使用 LabVIEW 进行编程。与普通数据采集卡一样，LabVIEW 中不但有模块化仪器相关的底层 VI 方便我们调用，同样也提供了一系列范例程序供我们参考，可以大大缩短软件的开发周期。

Demo 演示

视频演示了在软面板和 LabVIEW 两种工具中，使用信号发生器板卡输出信号，然后用示波器板卡采集和分析信号的过程。

信号发生器对应的驱动是 NI-FGEN，示波器对应的驱动是 NI-SCOPE，安装了这两个驱动以后，我们就可以使用它们的软面板，同时在 LabVIEW 中会包含相关的 VI 和范例程序供我们编程使用。

分别打开信号发生器和示波器的软面板。在信号发生器的软面板中对输出波形进行配置，如图 1-3 所示。比如波形的形状（正弦、方波、三角波、锯齿波、随机噪声），频率、幅度、相位、直流偏置等等。除了标准函数以外，信号发生器还可以输出一个在文件中编辑好的任意波形。

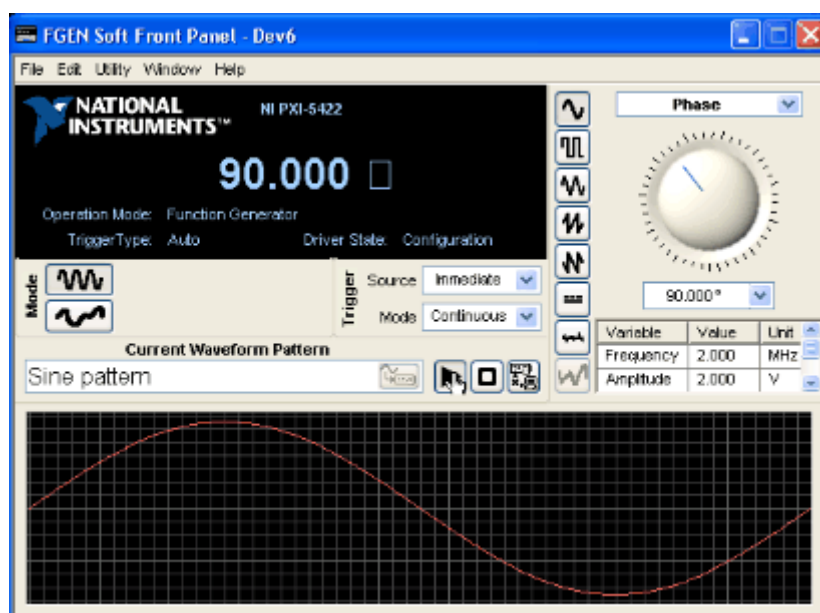


图 1-3 信号发生器软面板

示波器的软面板，跟传统仪器的操作界面也非常相似，如图 1-4 所示。选择输入通道、刻度值、耦合方式。然后点击运行，观察采集到的波形。如果需要获取波形的更多信息，点击测量按钮，添加不同的测量类型，比如频率，峰峰值，这些参数都可以在表格中实时地显示和更新。

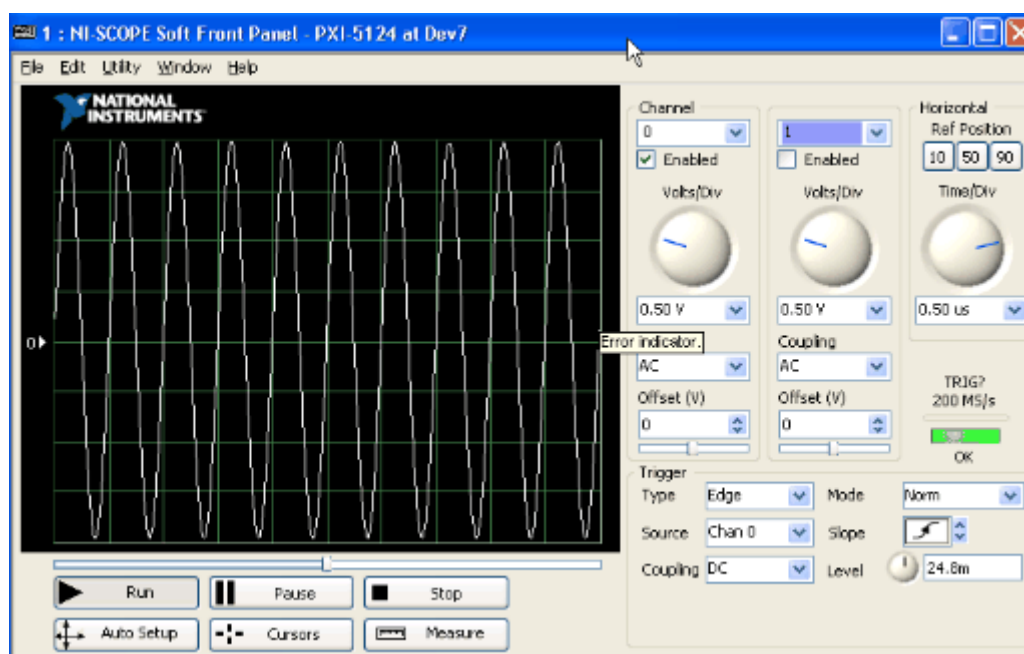


图 1-4 示波器软面板

下面再来看 LabVIEW 中的范例程序，在硬件输入输出》模块化仪器的文件夹下，有现成的范例供我们参考。在 NI-FGEN 选择 Fgen Basic Standard Function.vi，在 NI-SCOPE 下选择 niScope EX Measurement Library.vi 做为演示程序，演示结果详见视频。

打开程序框图，可以看到，与普通数据采集卡用 DAQmx 底层 VI 编程类似，模块化仪器相关的 LabVIEW 程序也是由仪器驱动提供的一系列底层 VI 搭建而成，如图 1-5 所示。编程简单方便，可大大缩短软件开发的周期。

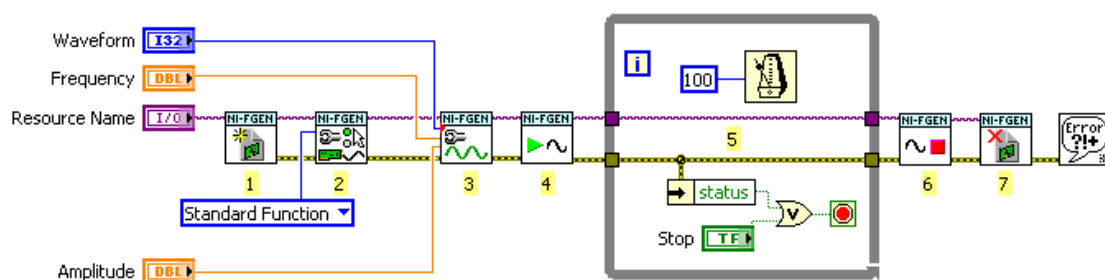


图 1-5 模块化仪器底层驱动 VI 构建的程序框图