

# 数据采集(DAQ)基础

基于PC进行进行测量的基础知识

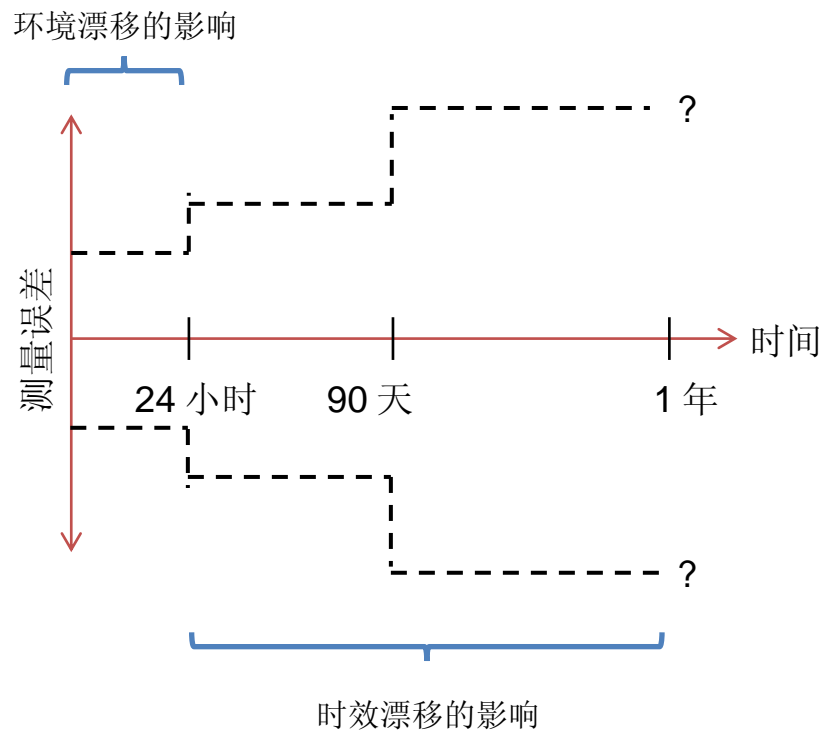
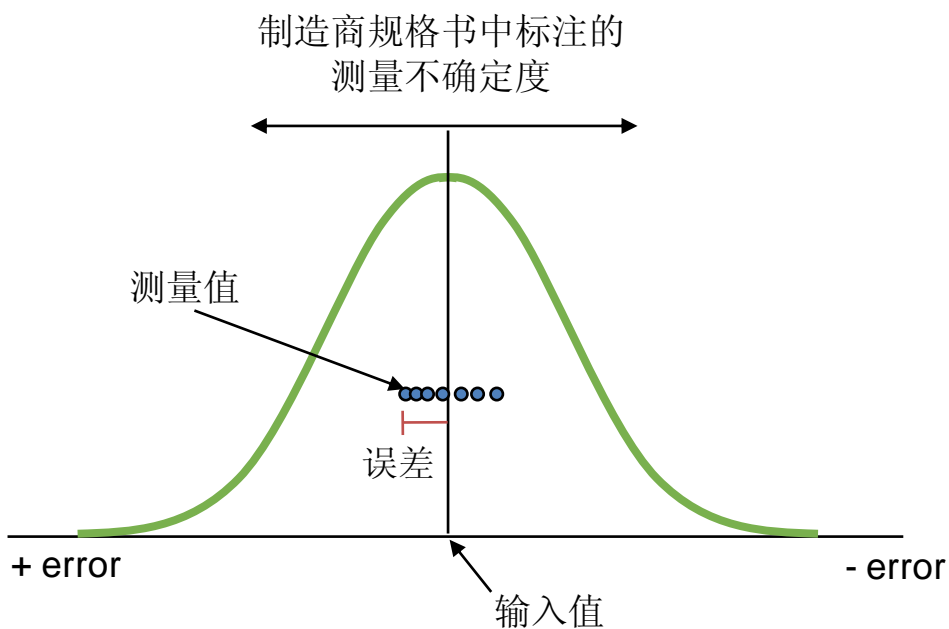
# 什么是数据采集(DAQ)?

数据采集是使用计算机测量电压、电流、温度、压力或声音等电子、物理现象的过程。

与传统的测量系统相比，基于PC的DAQ系统利用行业标准计算机的处理、生产、显示和连通能力，提供更强大、灵活且具有成本效益的测量解决方案。

# 严格来说所有测量都是不准确的

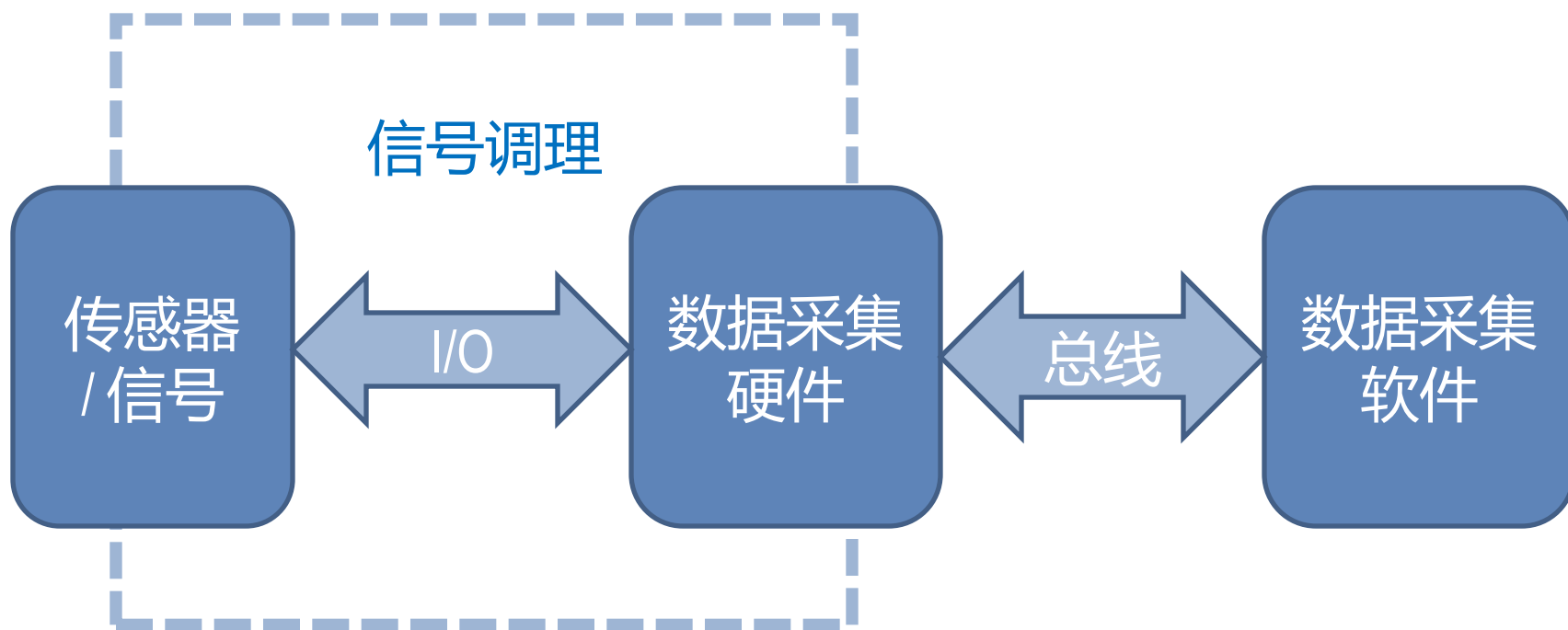
电子元器件随着时间会发生漂移，所以需要校准



# 数据采集的概念及应用

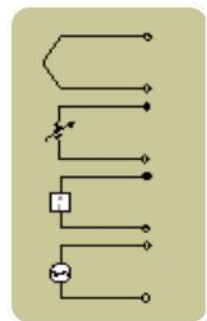
- 数据采集 (Data Acquisition, 简称DAQ)
  - 自动从布置于工厂、实验室、或现场的传感器、仪器、设备等收集获取数据的过程
  - 狭义的数据采集主要是模拟输入(AI)，其目的是为了测量某种电信号或物理信号，如电压、电流、温度、压力、加速度、声强等
  - 广义的数据采集还包括模拟输出、数字I/O等
    - 例如，目前市面上的多功能数据采集设备通常包括模拟输入、模拟输出、数字I/O、计数器/定时器等功能，如NI的M系列多功能DAQ卡
    - 现在一些传感器/变送器已经集成了A/D转换功能，直接通过数字接口读取数据，从而不需要模拟输入采集
  - 数据采集的应用十分广泛，几乎涵盖所有工程专业和科学研究方向
    - 电子、电气、机械、车辆工程、海洋工程、环境、化工、生物医学、土木工程、能源电力、高能物理...

# 数据采集系统概述



# 典型的数据采集系统硬件架构

数据采集硬件可以将PC变为一个自动化系统



传感器

- 任意类型



信号连接

- 直接连接
- 或通过接线端



DAQ设备

- PCI/PXI
- PCIe/PXIe
- USB
- Ethernet



PC

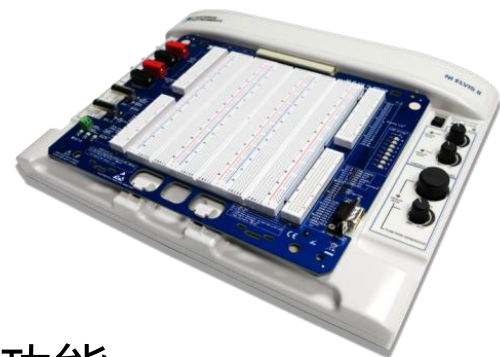
- Windows
- Linux
- Mac

# 部分常用的数据采集设备类型

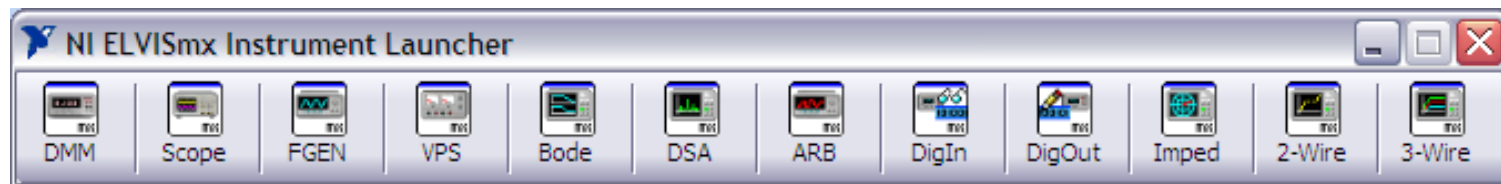
- 实验室、工业环境使用
  - 基于PCI/PXI接口
  - 往往需要外接端子和线缆
- 便携式/远距离
  - USB, Ethernet接口
- 教学实验与学生课外练习
  - 如ELVIS、myDAQ
  - 除了数据采集硬件电路之外还集成了其他一些功能，如数字万用表、可编程电源等



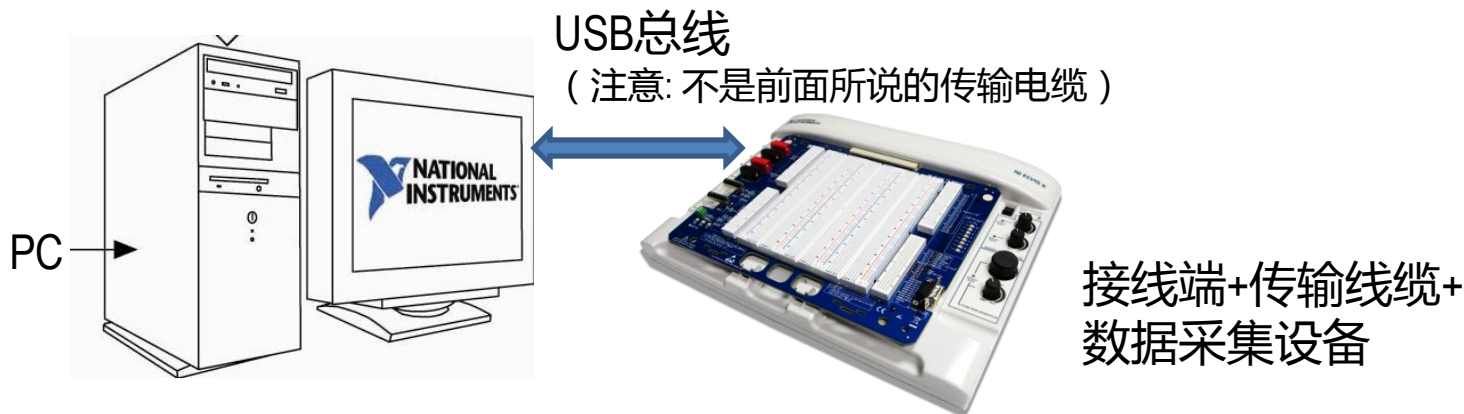
# 举例: 用于教学实验的NI ELVIS



- 一种集成了12种仪器的教学实验设备
  - 安装NI-ELVIS驱动后可通过ELVIS软面板实现这些功能



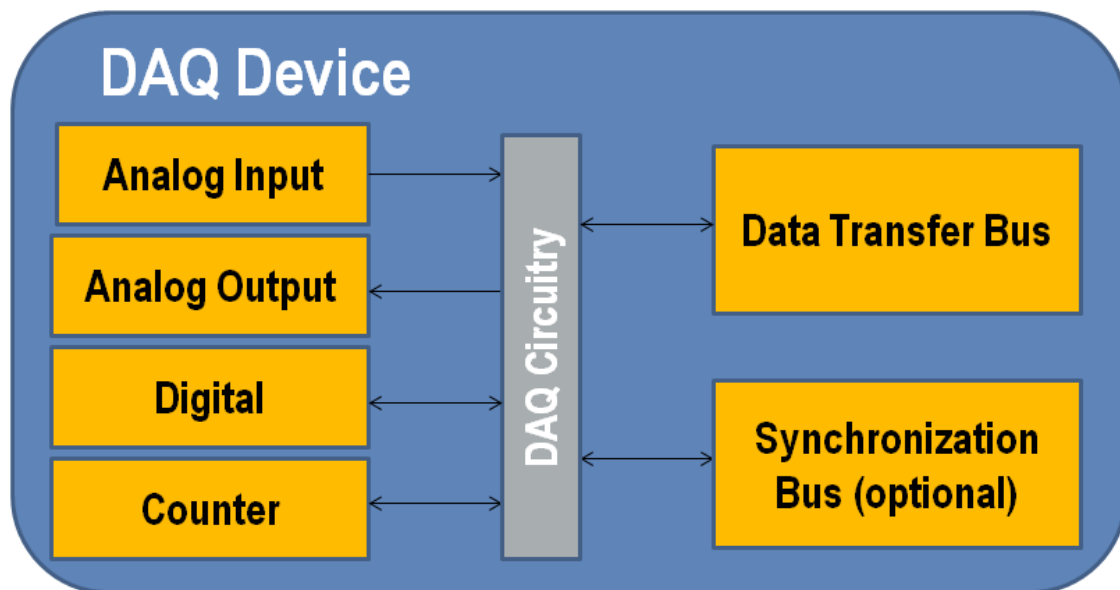
- 同时也可看做是一种基于USB总线的多功能数据采集设备
  - 用于学习虚拟仪器（尤其是数据采集）的相关编程技术





# 数据采集设备的构成

- 数据传输总线
- 同步总线
- DAQ Circuitry
  - 时钟/定时电路
  - 板载FIFO
  - 信号路由
  - 内部校准电路
- 前端电路
  - 模拟输入/模拟输出/数字IO/计数器
  - 注意: 模拟输入可能通过复用器共用一个ADC, 模拟输出通常是每通道专用DAC



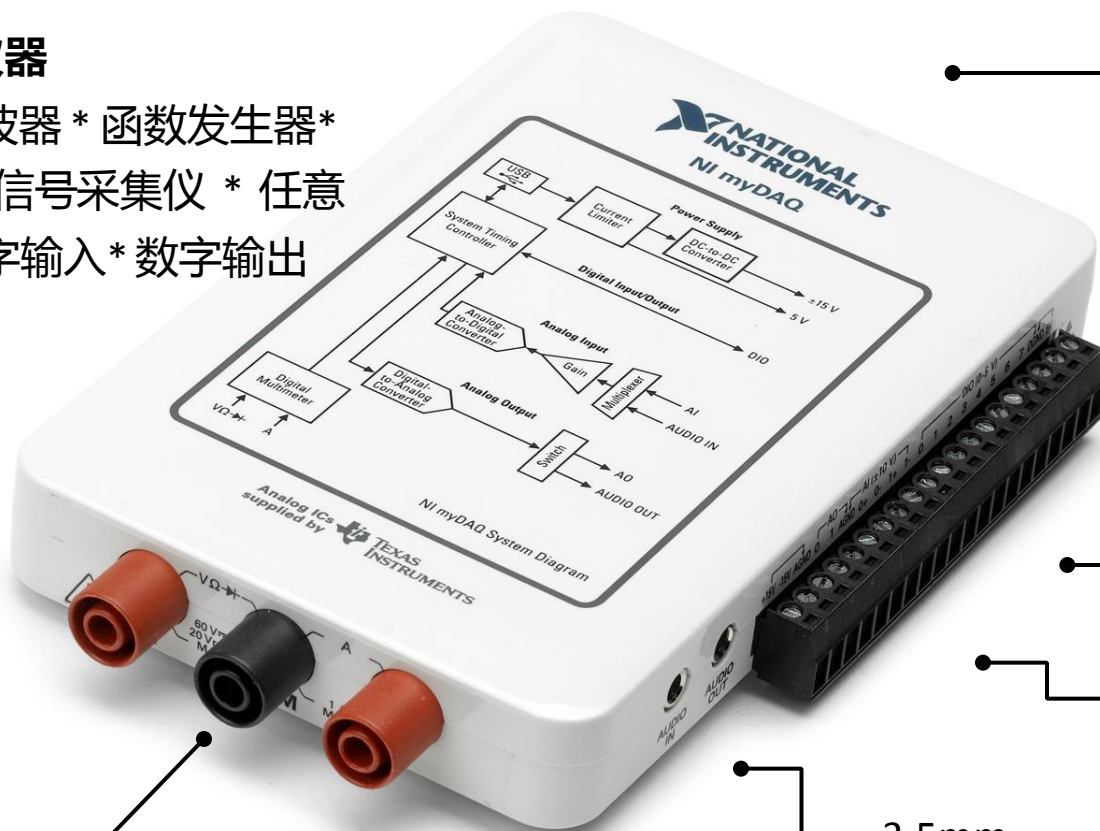
# 学生创新动手实践平台： NI myDAQ —— 随时随地实践工程创新

## 即插即用的8种仪器

数字万用表 \* 示波器 \* 函数发生器 \*  
波特图仪 \* 动态信号采集仪 \* 任意  
信号发生器 \* 数字输入 \* 数字输出

与计算机的接口  
LabVIEW中的  
NI DAQmx 或  
ELVISmx 驱动

集成的数字万用表  
V, I,  $\Omega$ , Diode



USB总线供电

$\pm 15V$  +5V  
电源输出

8通道  
数字输入输出

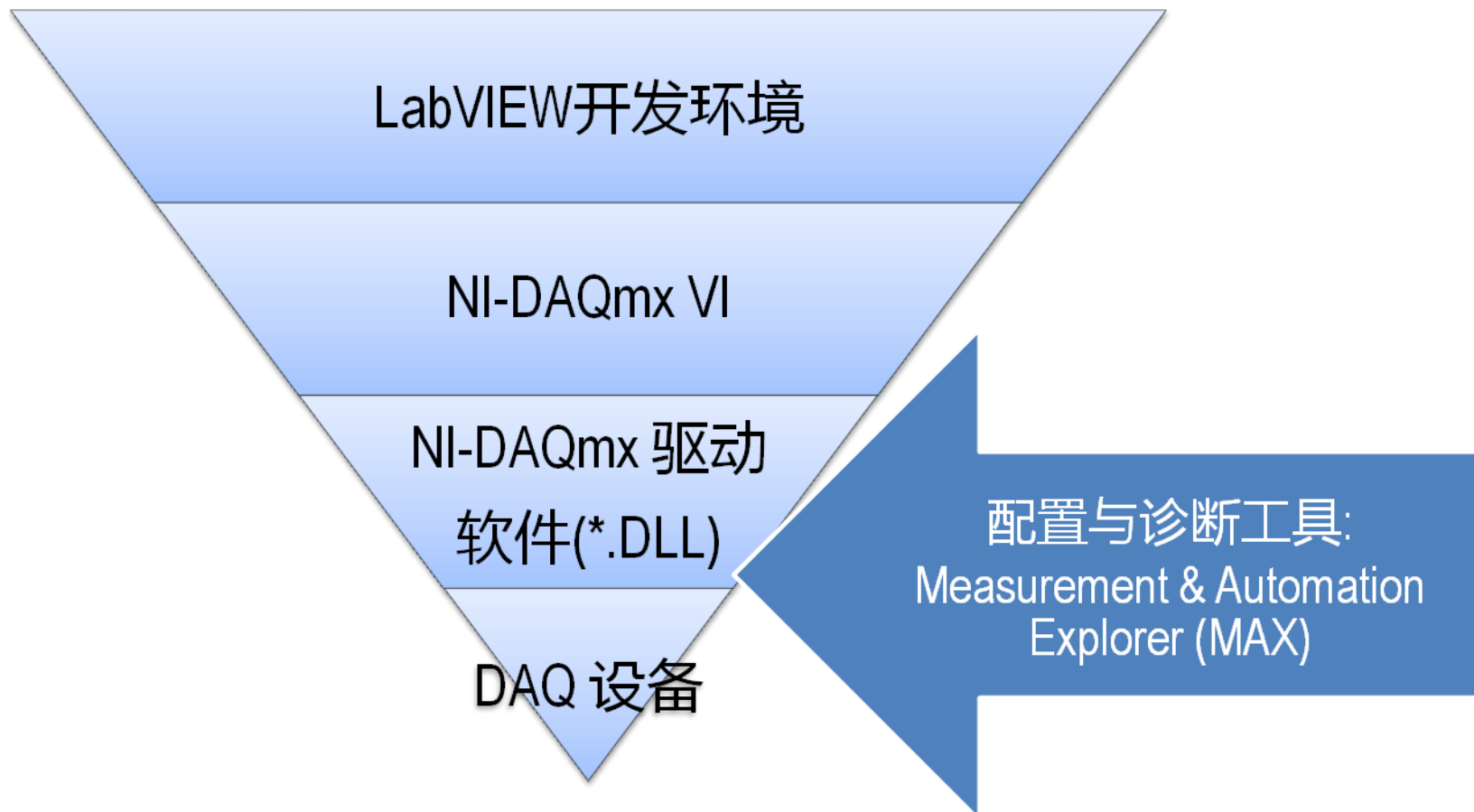
1个计数器

双通道差分模拟  
输入，双通道模  
拟输出，200ks/s

3.5mm  
音频输入与输出

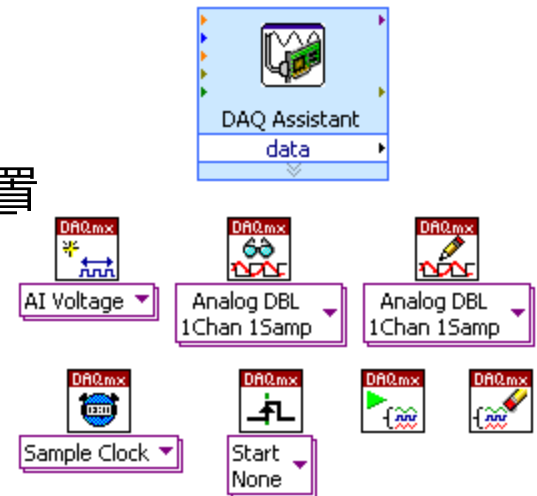
# 数据采集软件

以LabVIEW作为开发环境为例



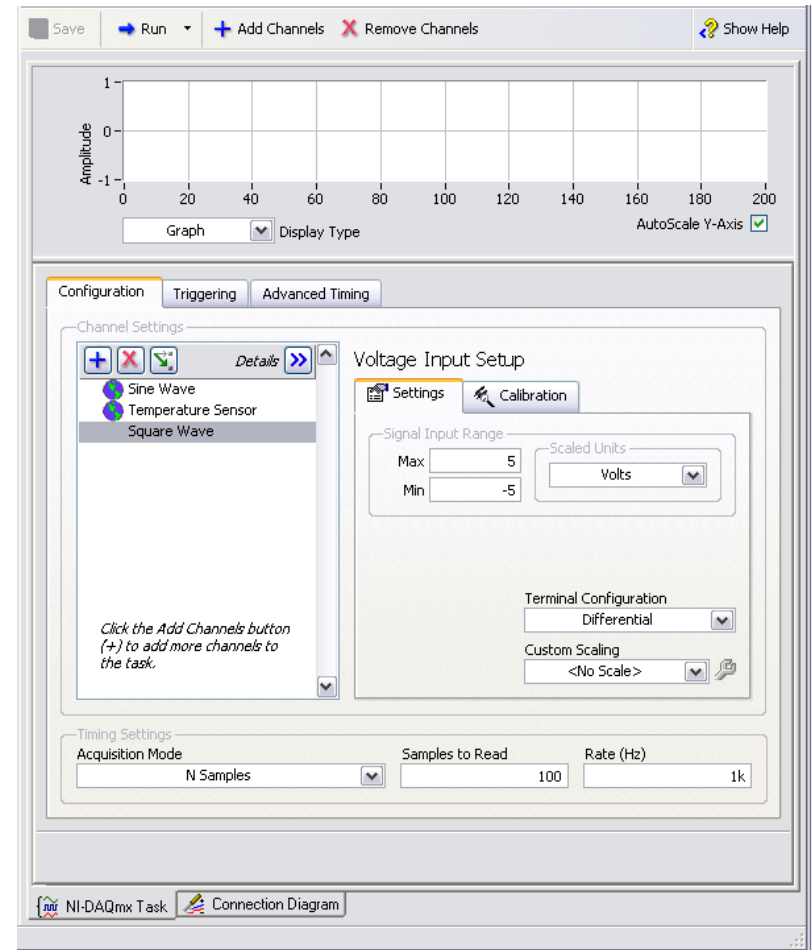
# NI数据采集软件方面的一些术语

- NI-DAQmx
  - 驱动层软件
  - 可以检测到数据采集硬件
  - 在LabVIEW中安装NI-DAQmx函数
- Measurement & Automation Explorer
  - 简称MAX，随NI-DAQmx或任何其他NI驱动软件安装
  - 配置或检测数据采集硬件
- DAQ Assistant (DAQ助手)
  - 用于开发数据采集应用的Express VI，基于配置
- DAQmx API
  - 一系列用于数据采集应用开发的API函数



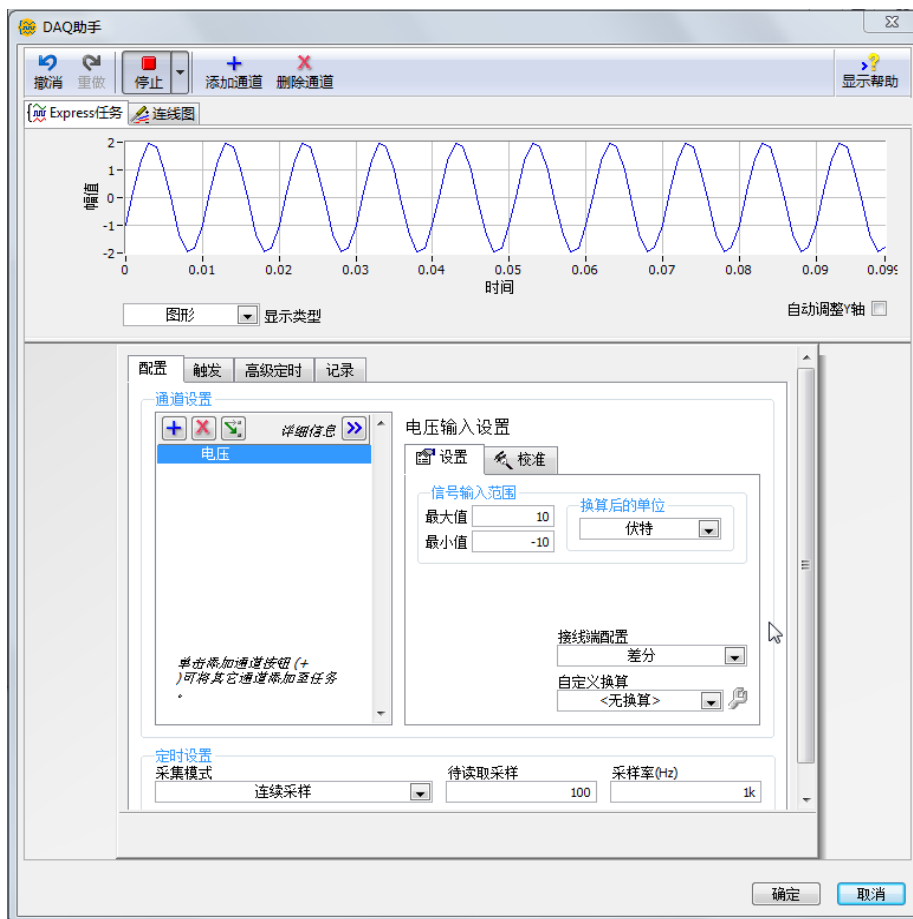
# 数据采集助手（LabVIEW基础培训内容）

- 两种调用方式
  - 在MAX中的“数据邻居”中创建任务
  - 通过LabVIEW中的Express VI
- 基于配置，无需编程
  - 在LabVIEW中可自动转化为调用NI-DAQmx底层API VI的程序，可自定义修改



# MAX 与 DAQ助手

- 该部分为LabVIEW基础培训内容，因此仅作为复习

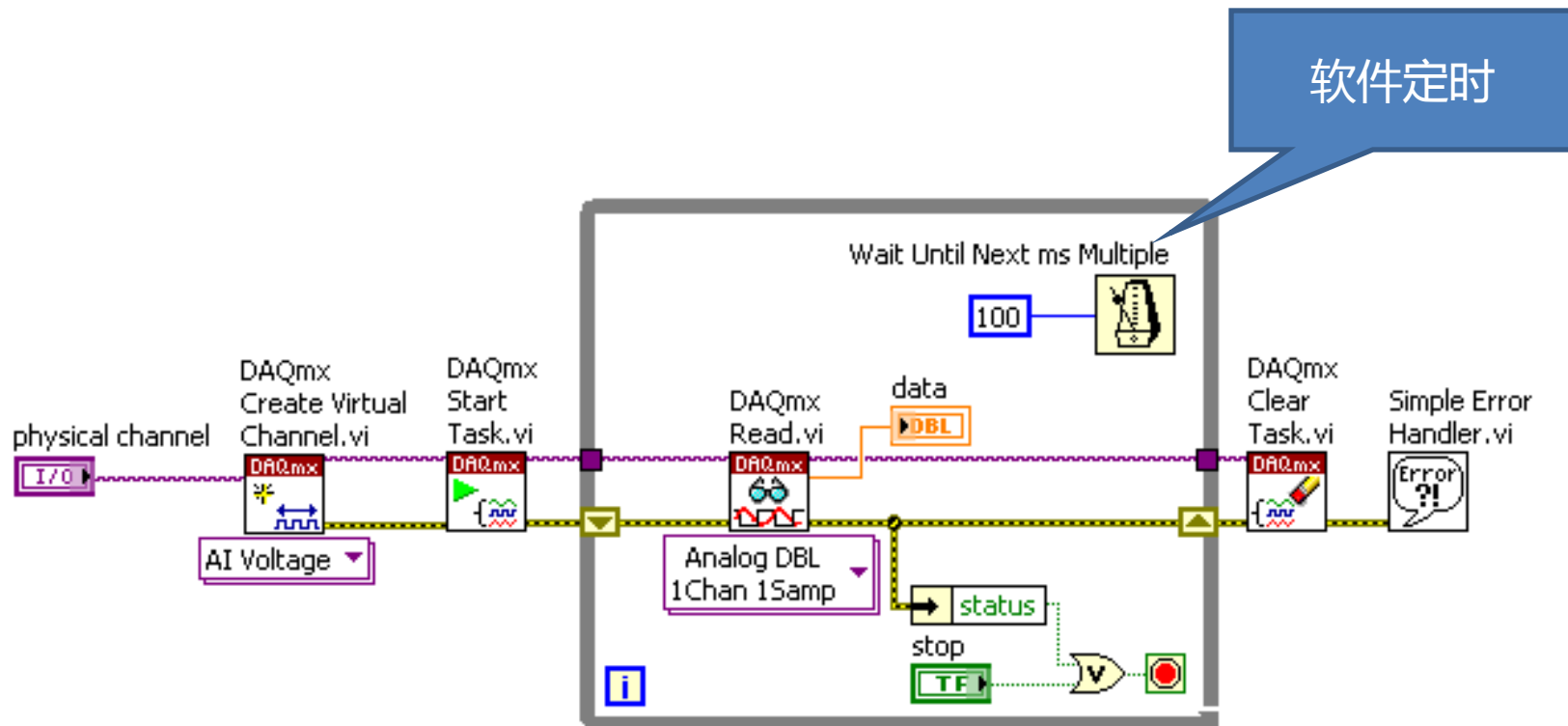


# 模拟输入 (Analog Input)

- 数据采集一词狭义即指模拟输入
  - 即通过A/D转换将模拟信号采样为数字信号，从而可被计算机设备进一步处理
- 常用于实现传感器信号的采集以及电信号的采集

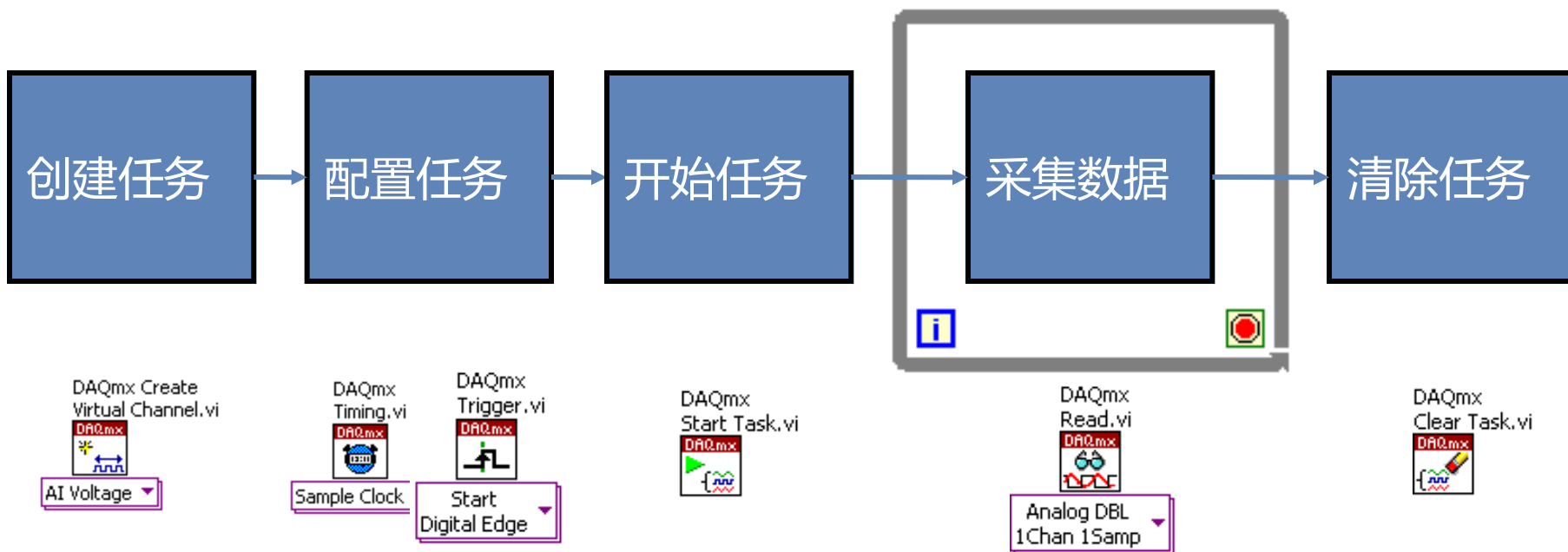
# 一个最简单的模拟输入电压采集程序

- 单点电压采集
- 分别用DAQ助手和DAQmx底层VI实现

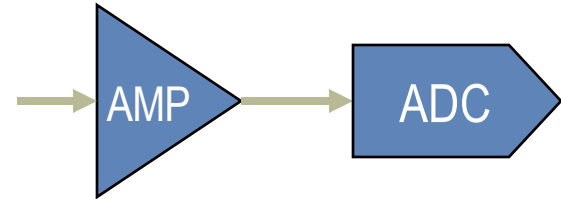




# DAQmx数据采集程序的基本架构

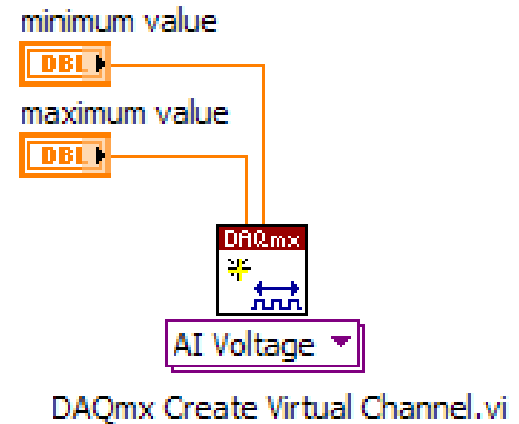
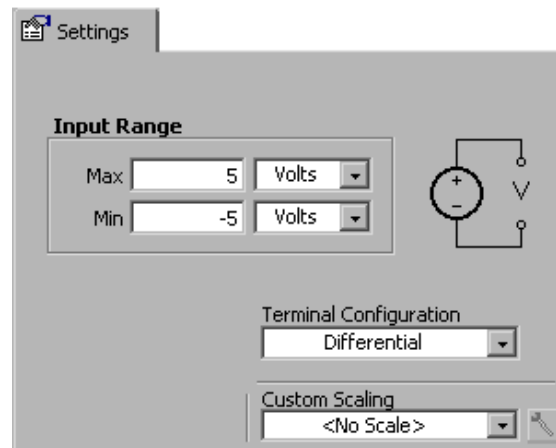


# 输入范围

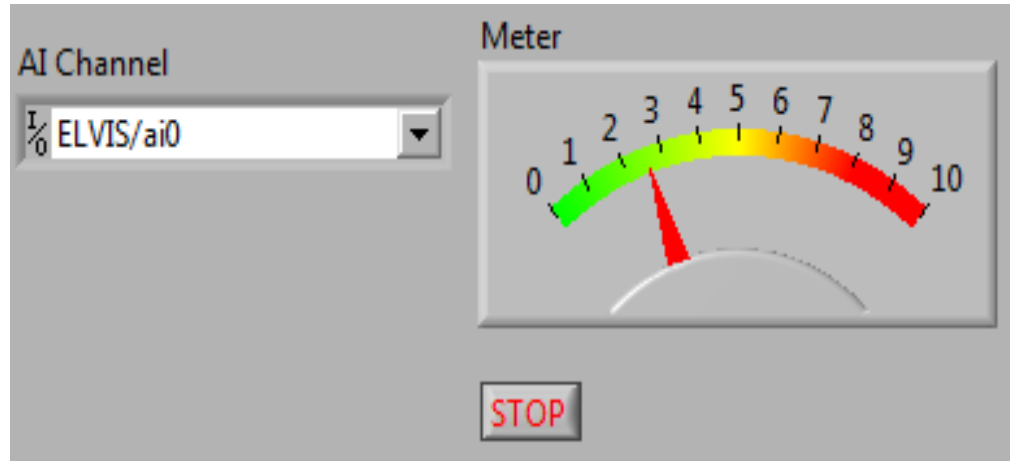


- 仪器放大器

- 为了尽量用满ADC位数，模拟采集通道配有仪器放大器
- 通过合理设置输入最大值和最小值参数，可以使DAQ设备自动配置最合适的仪器放大器增益，从而尽量用满ADC位数



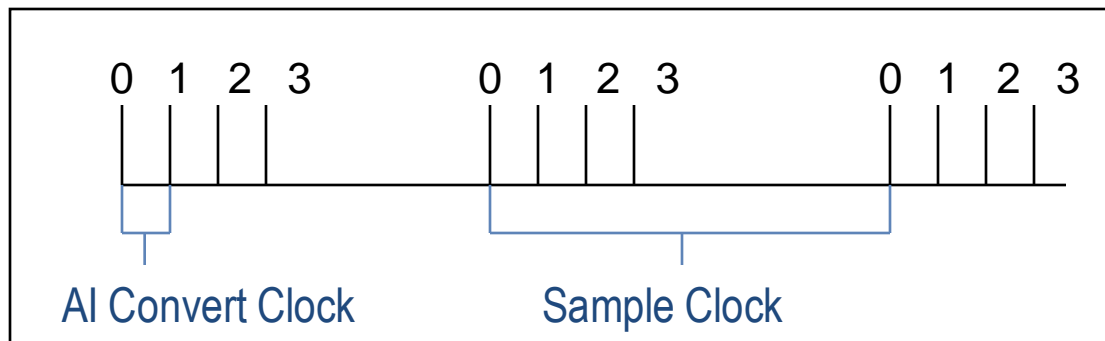
# Lab 3.1 软件定时的单点模拟采集



# 采样术语 (针对硬件定时)

- 采样率

- 单一通道每秒采样点数



- 采样时钟

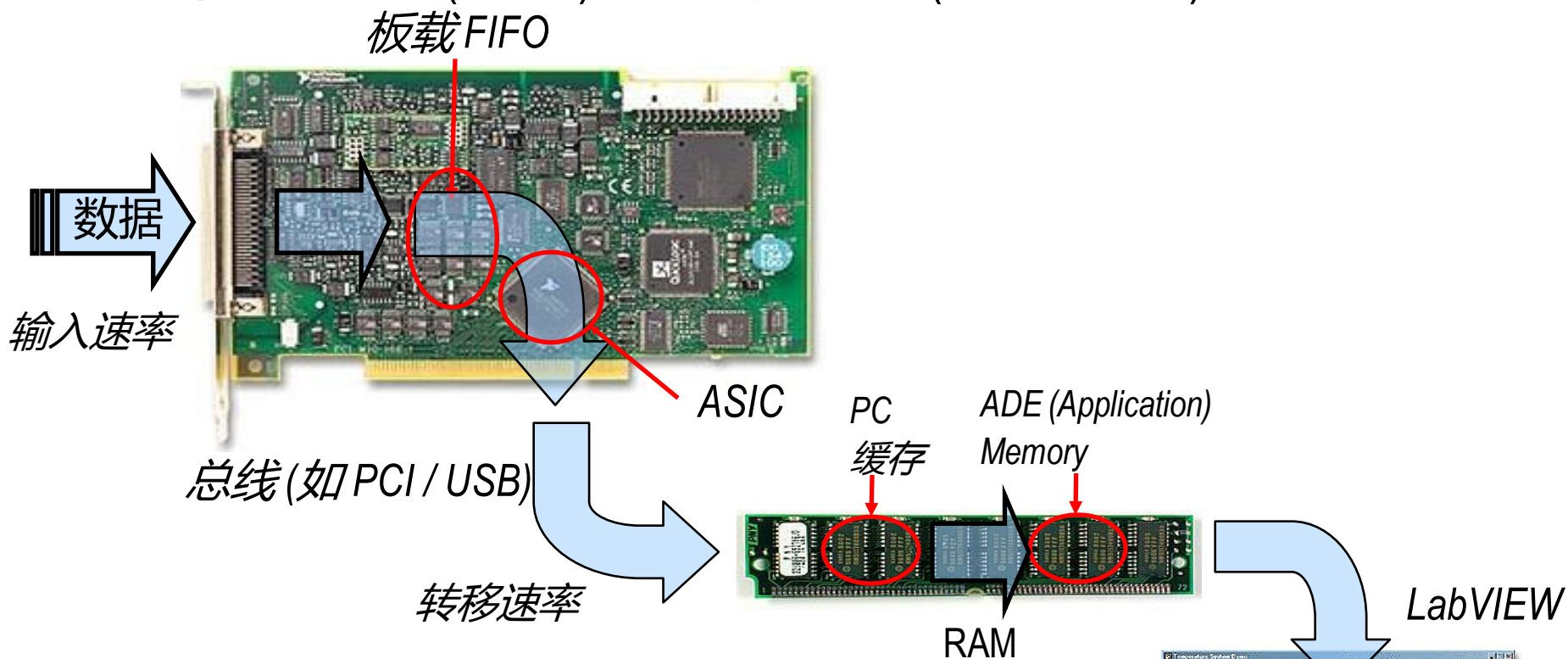
- 速率等于采样率

- AI转换时钟

- 直接产生A/D转换的时钟
- 通道之间会有延时
- 同步采样架构相对多路复用采样架构来说，可认为通道之间的转换时钟是同步的

# 硬件定时的采集需要通过Buffer

- 有限点采集 (Finite) 和 连续采集 (Continuous)



- PC缓存必须够大 (至少需超过一次传递的数据量)
- 连续采集中, 如果要使两处缓存一直不溢出, 必须保证总线的数据转移速率大于数据的输入速率, 同时程序必须尽快读取PC缓存中的数据



# 有限点数据采集的LabVIEW程序

采样点数设置



Samples per Channel

132

sample mode

Finite Samples

Rate

DBL

DAQmx  
Timing.vi

DAQmx  
Start  
Task.vi

timeout  
10.00

DAQmx  
Read.vi

data  
DBL

DAQmx  
Clear  
Task.vi

Simple  
Error  
Handler.vi

minimum value  
DBL

maximum value  
DBL

physical channel  
I/O

DAQmx  
Create  
Virtual  
Channel.vi

AI Voltage

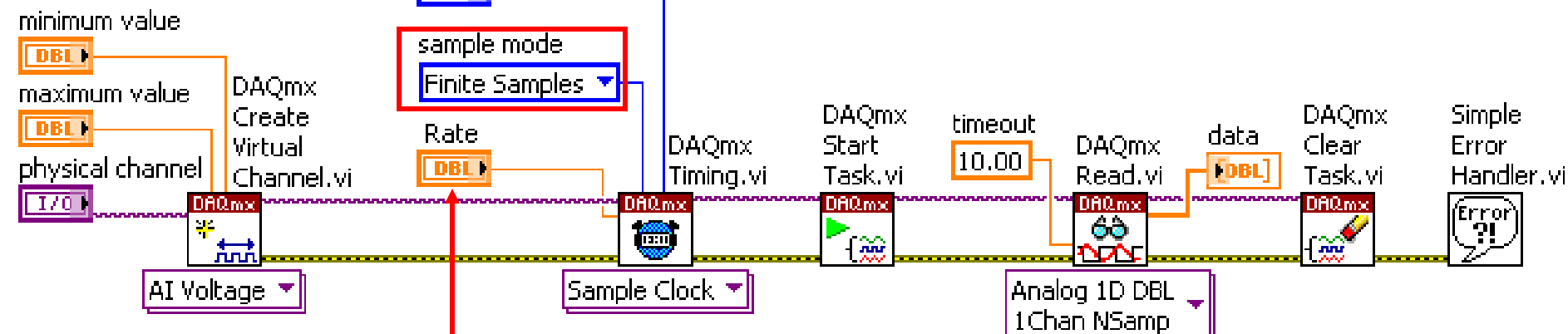
Sample Clock

Analog 1D DBL  
1Chan NSamp

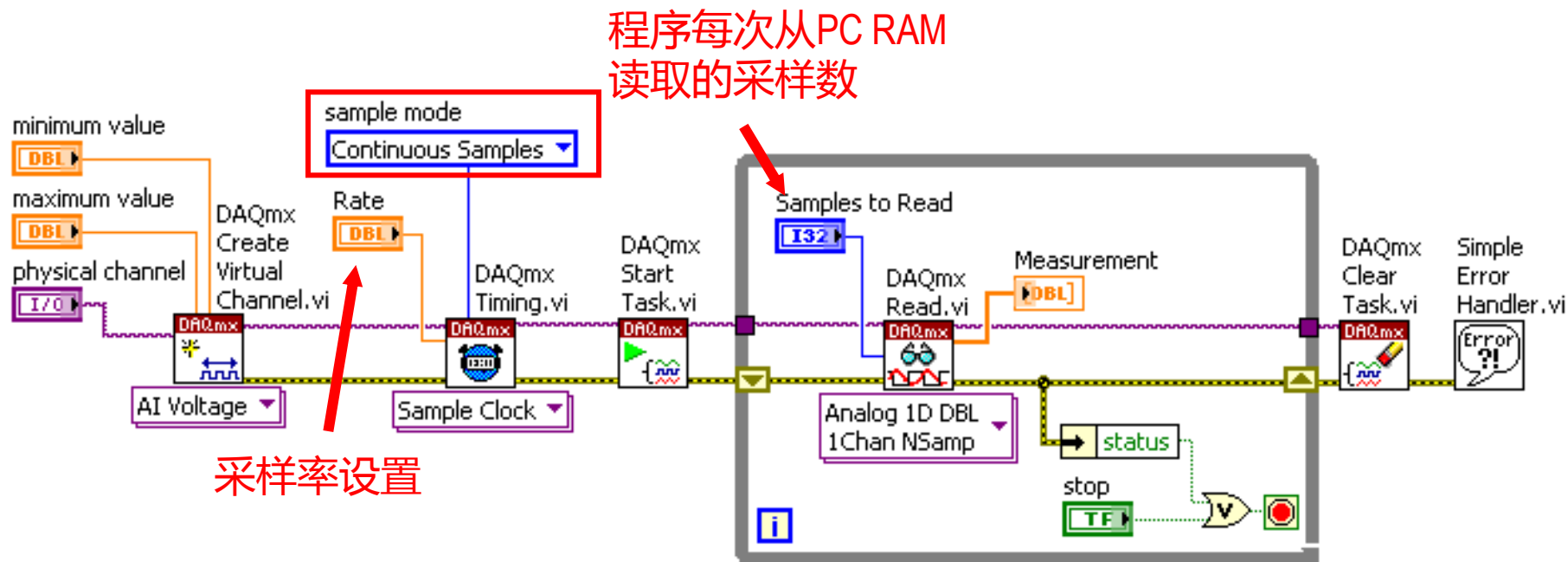
采样率设置



读取多个采样



# 连续数据采集的LabVIEW程序

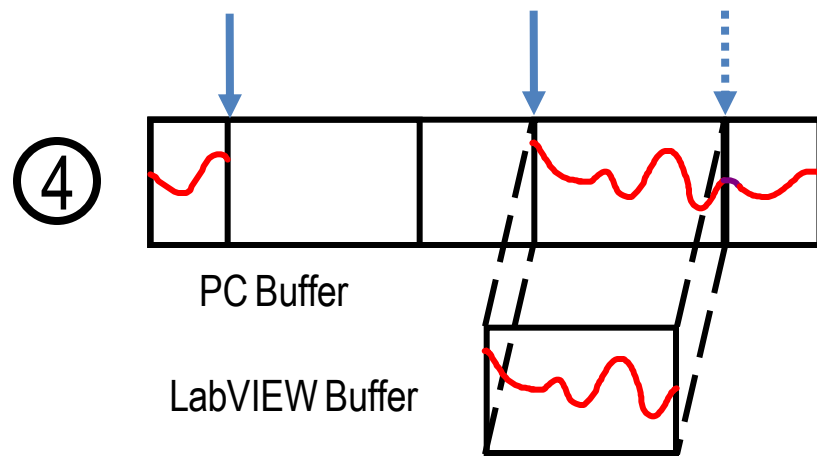
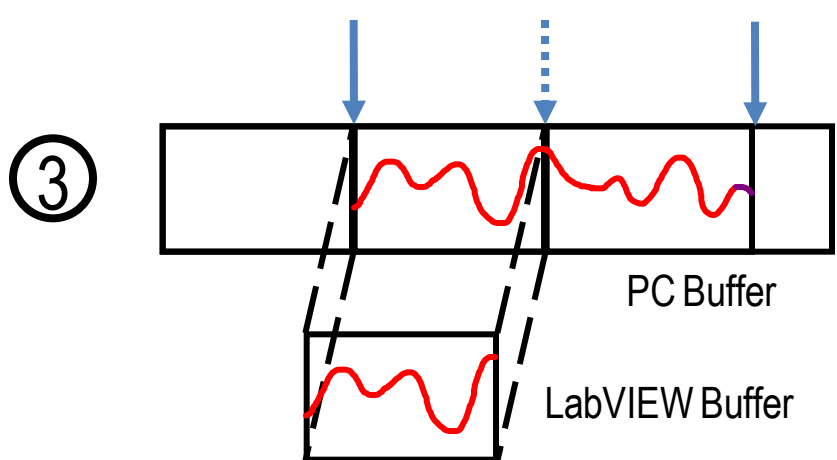
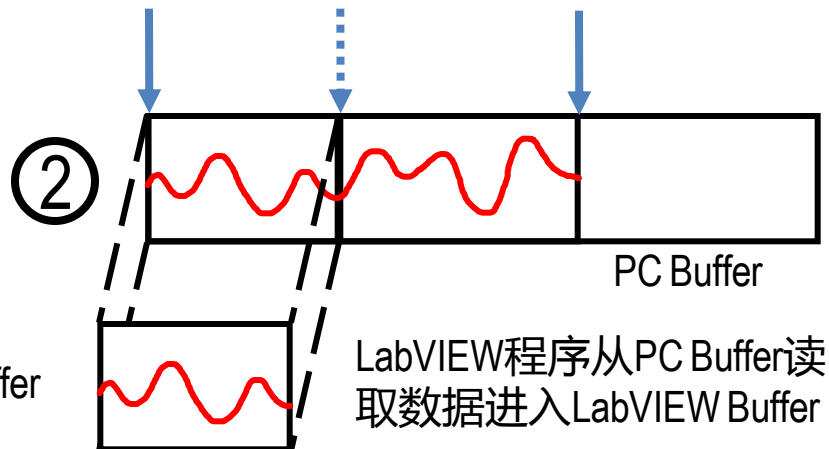
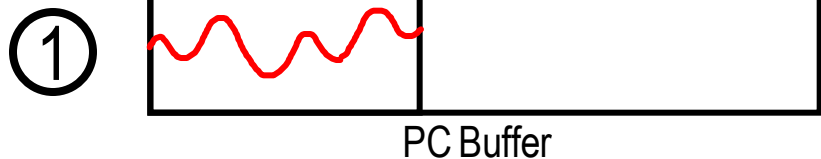


- 驱动会根据采样率设置自动选择合适的RAM Buffer大小

| 采样率                  | 缓存     |
|----------------------|--------|
| 0-100 S/s            | 1 kS   |
| 100-10,000 S/s       | 10 kS  |
| 10,000-1,000,000 S/s | 100 kS |
| > 1,000,000 S/s      | 1 MS   |

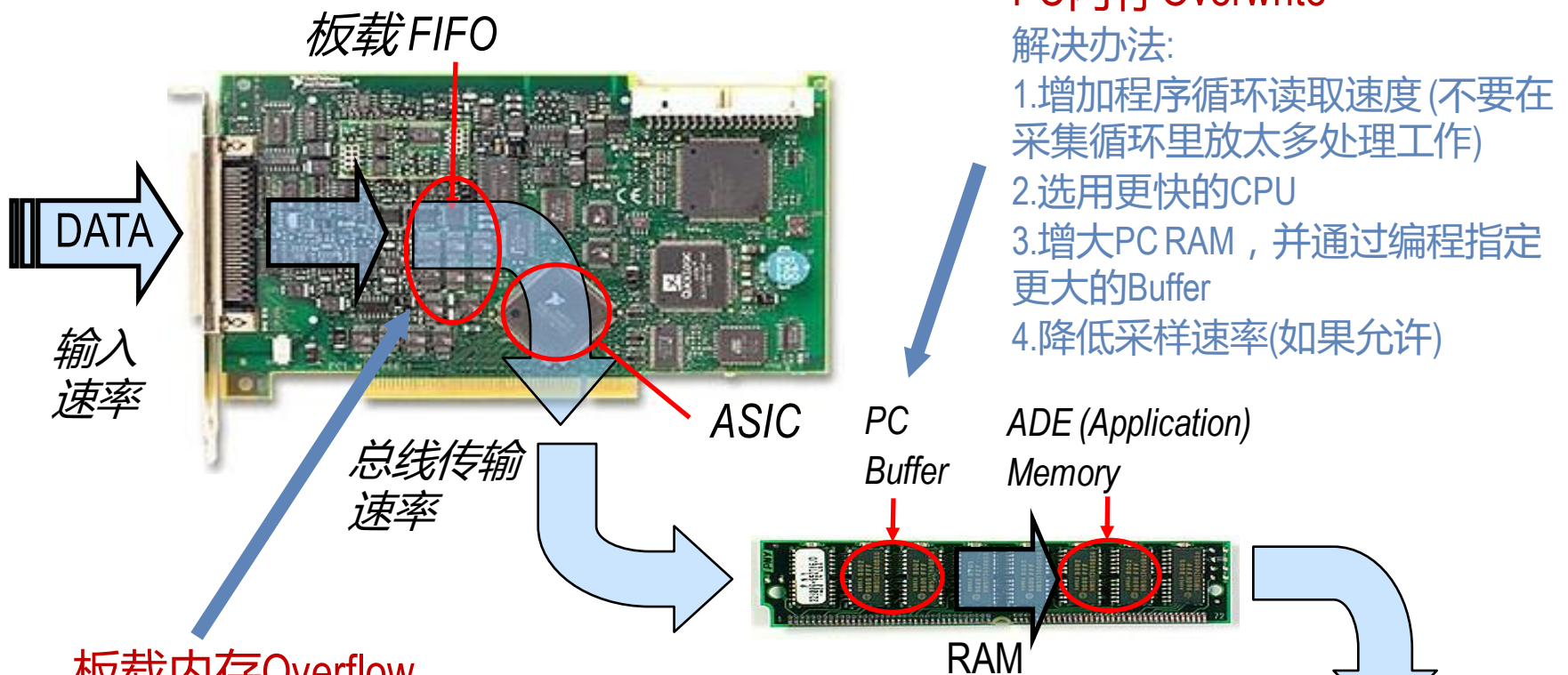
# 理解连续数据采集时PC Buffer

通过总线来自于采集设备的数据





# 连续采集时可能的数据传输异常



## PC内存Overwrite

解决办法:

1. 增加程序循环读取速度 (不要在采集循环里放太多处理工作)
2. 选用更快的CPU
3. 增大PC RAM, 并通过编程指定更大的Buffer
4. 降低采样速率(如果允许)

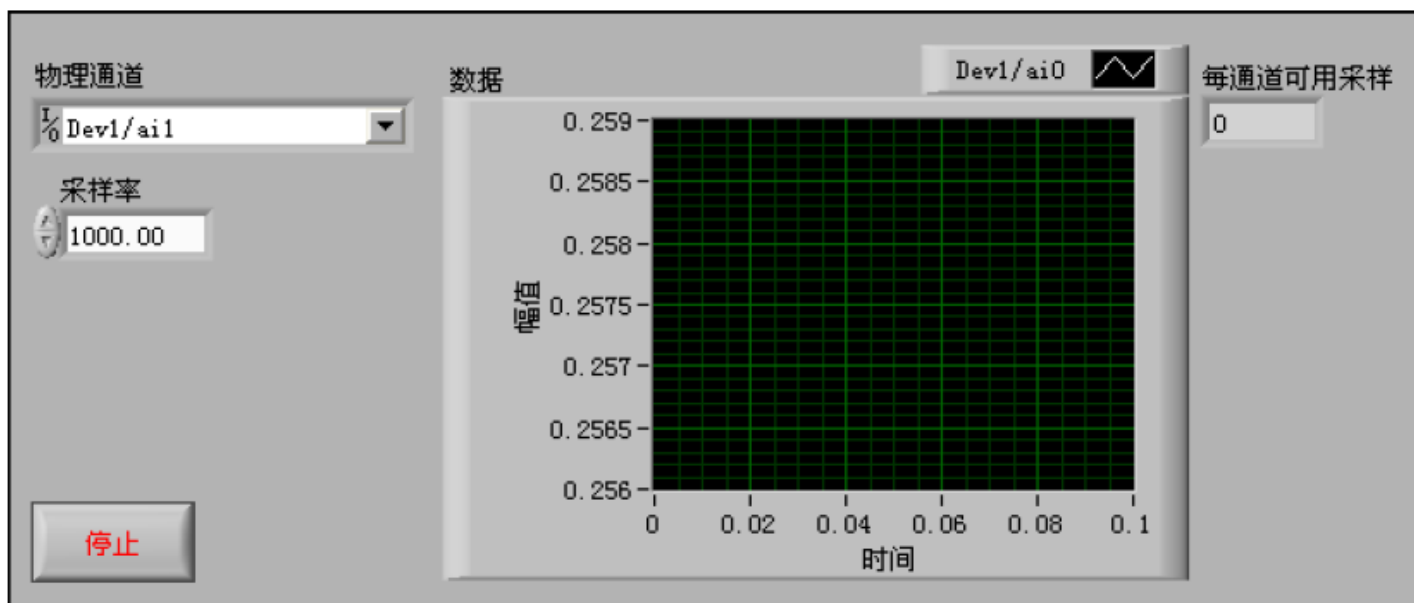
## 板载内存Overflow

解决办法:

1. 提高总线带宽
2. 选择板载FIFO较大的板卡
3. 降低采样速率(如果允许)



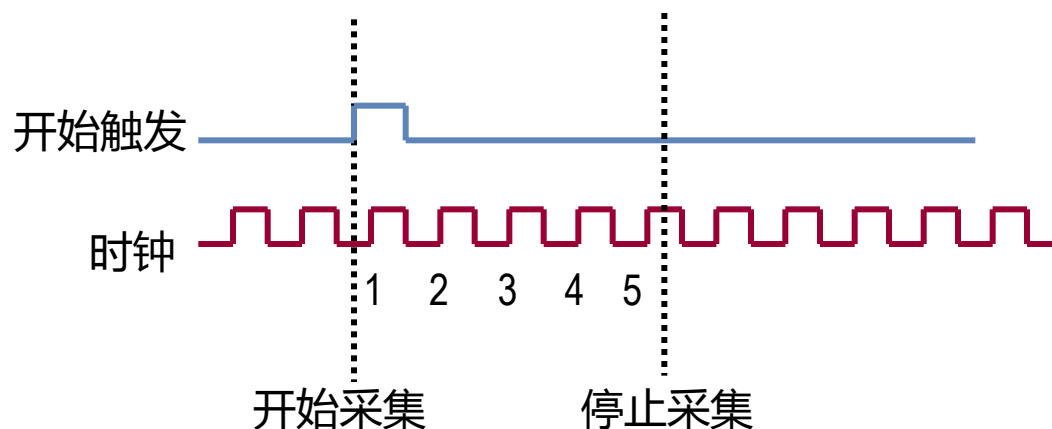
# Lab 3.2 连续信号采集



# 触发 (Trigger)

- 触发的概念

- 每个动作需要一个“激励”或“原因”
- 动作: 比如开始采集信号后产生波形输出

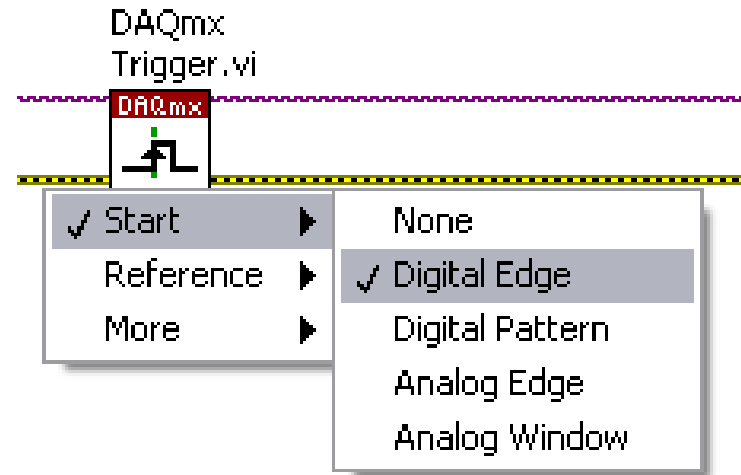


- 触发的分类

- 开始触发、参考触发、停止触发（按动作结果来分）
- 模拟触发、数字触发（依照触发“激励”信号来分）
- 不同的设备不一定支持所有触发方式，可参阅相关手册

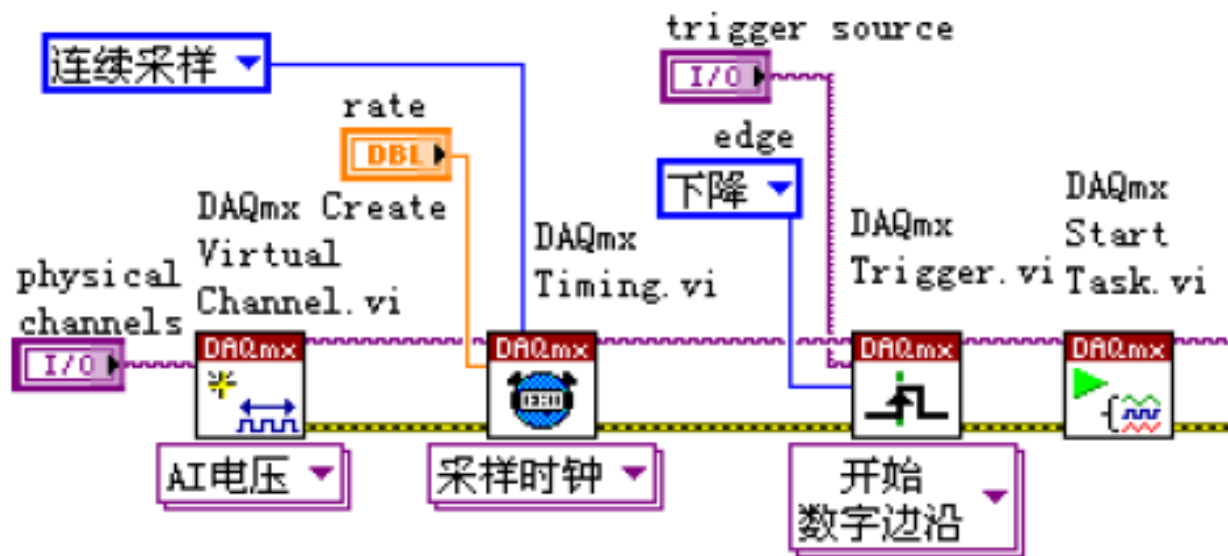
# LabVIEW中对触发的编程

- 在任务种增加DAQmx Trigger.vi并作相应配置

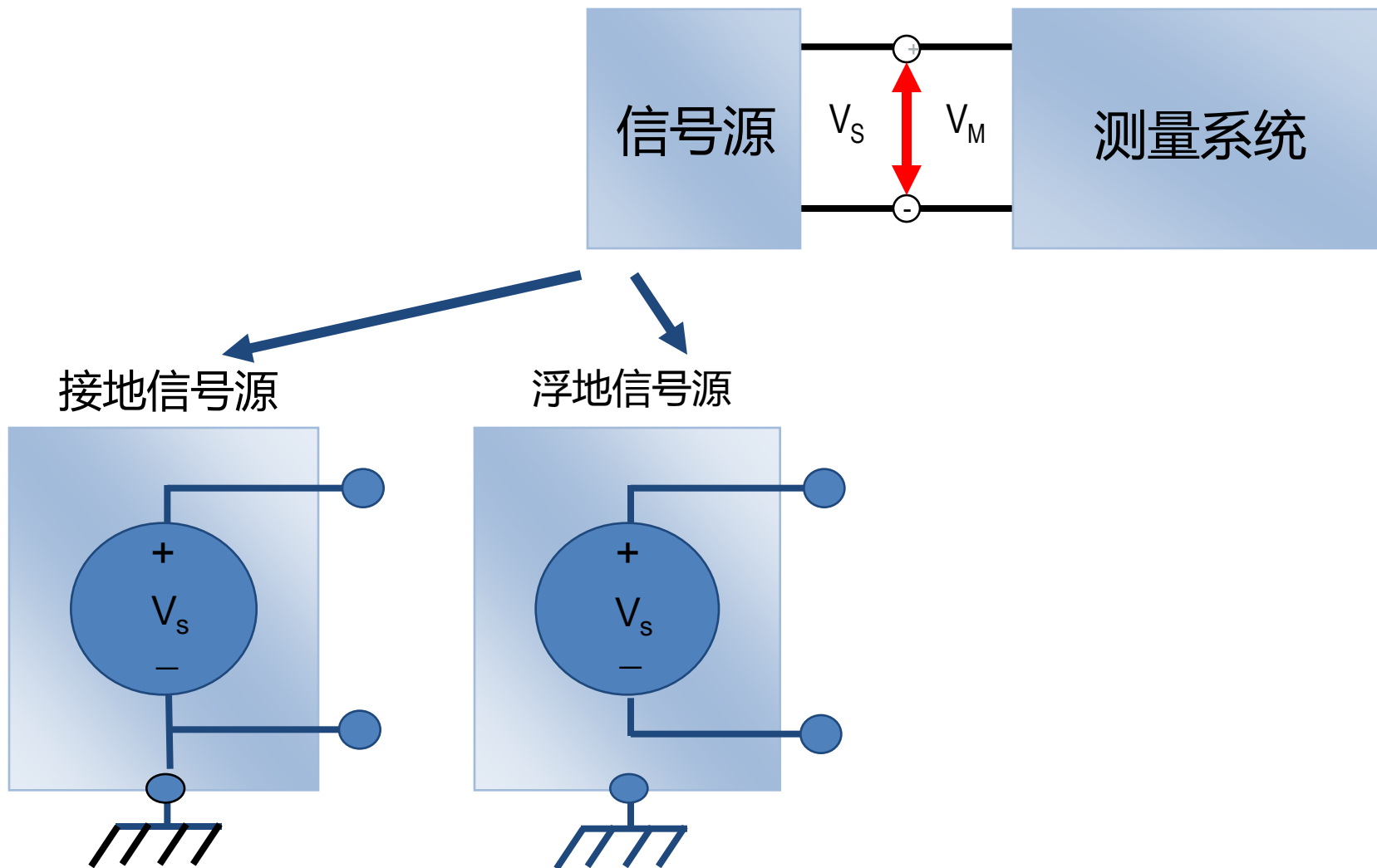


# Lab 3.3 带触发的连续信号采集\*

- \*仅支持ELVIS平台

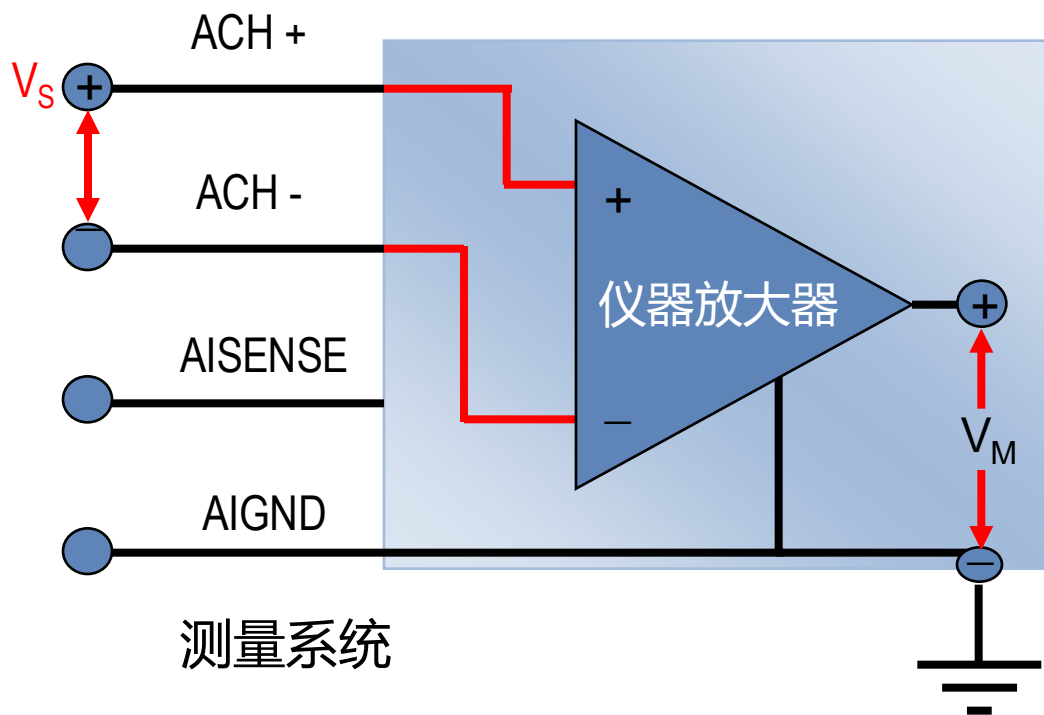


# 数据采集中的接地问题



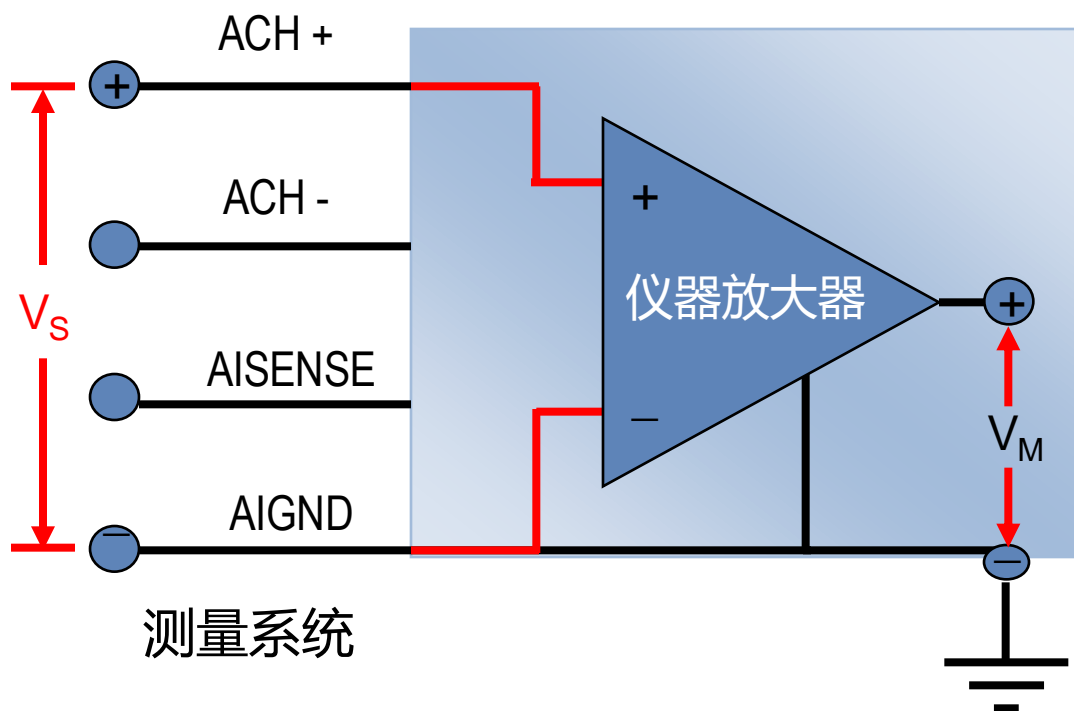
# 接线方式1：差分 (Differential)

- 可以抑制共模电压和共模噪声
- 我们此次培训中用到的主要是差分接线方式



## 接线方式2：参考单端（RSE）

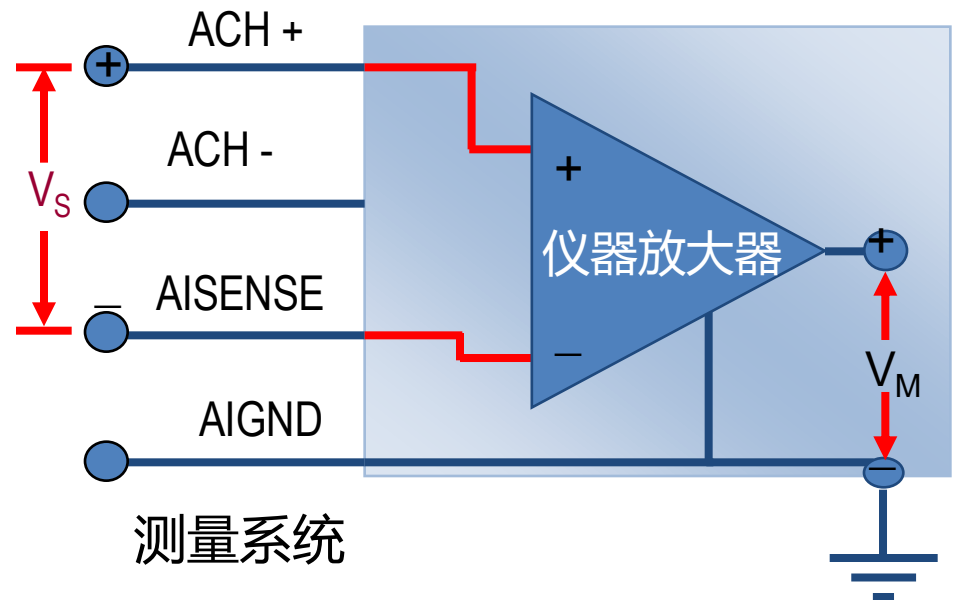
- 测量基于对地参考
- 不能抑制共模噪声



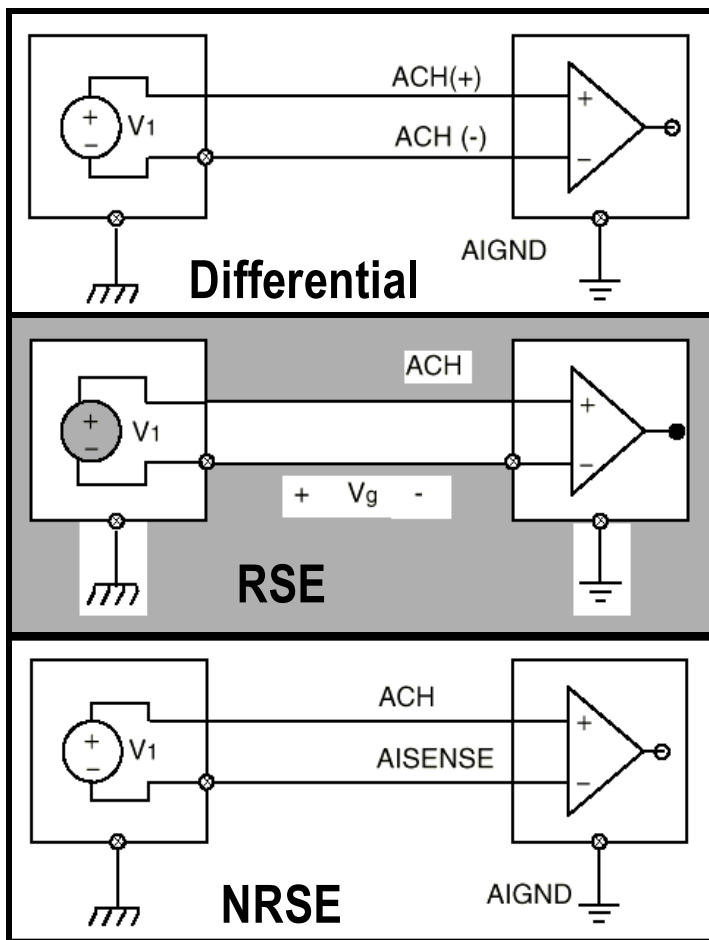
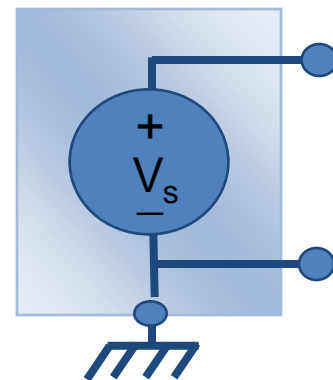


# 接线方式3：非参考单端 (NRSE)

- 测量基于对AISENSE端的参考
- 但是多个通道测量时，AISENSE是共用的
  - 不能抑制共模电压



# 对于接地信号源的测量



最好

- + 抑制共模电压
- 可用通道数减少一半

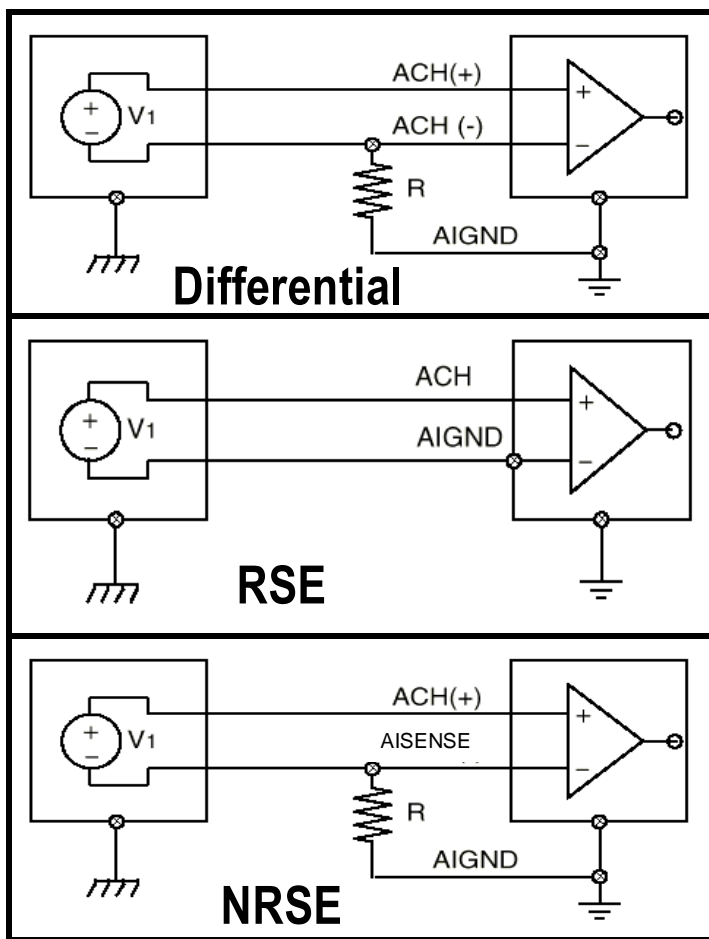
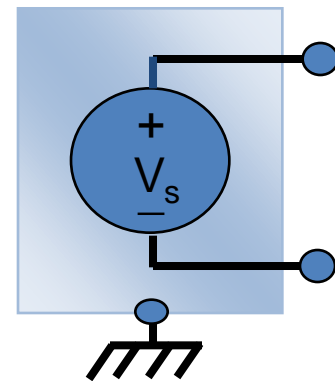
不推荐

- 两个地之间的电压  $V_g$  会产生接地回路, 所产生的电流有可能损坏设备

可以

- + 所有的通道都可以使用
- 不能抑制共模电压

# 对于浮地信号源的测量



最好

- + 抑制共模电压
- 可用通道数减少一半
- 需要偏置电阻

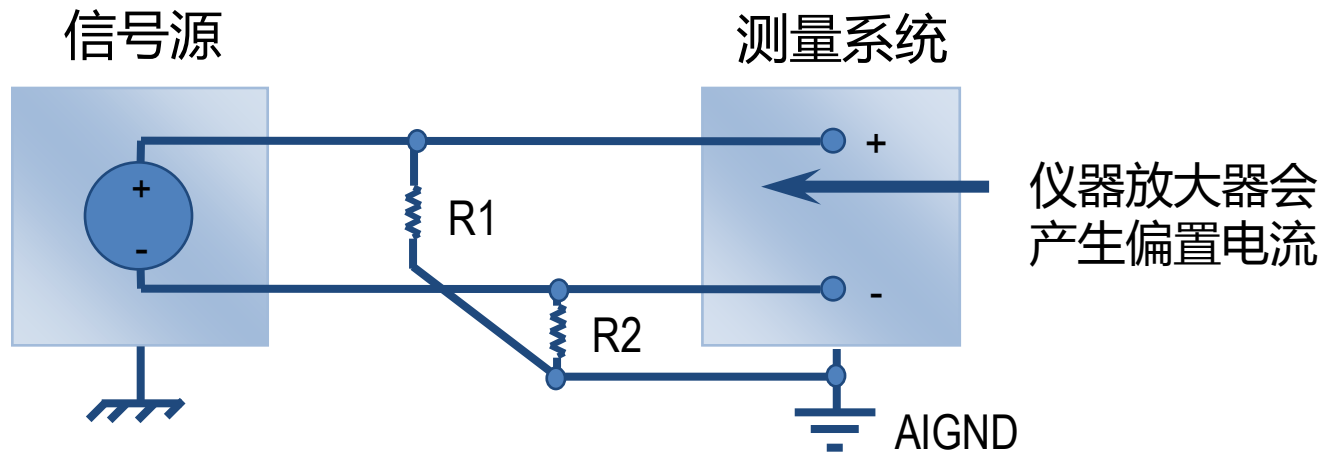
好

- + 所有的通道都可以使用
- + 不需要偏置电阻
- 不能抑制共模电压

可以

- + 所有的通道都可以使用
- 需要偏置电阻
- 不能抑制共模电压

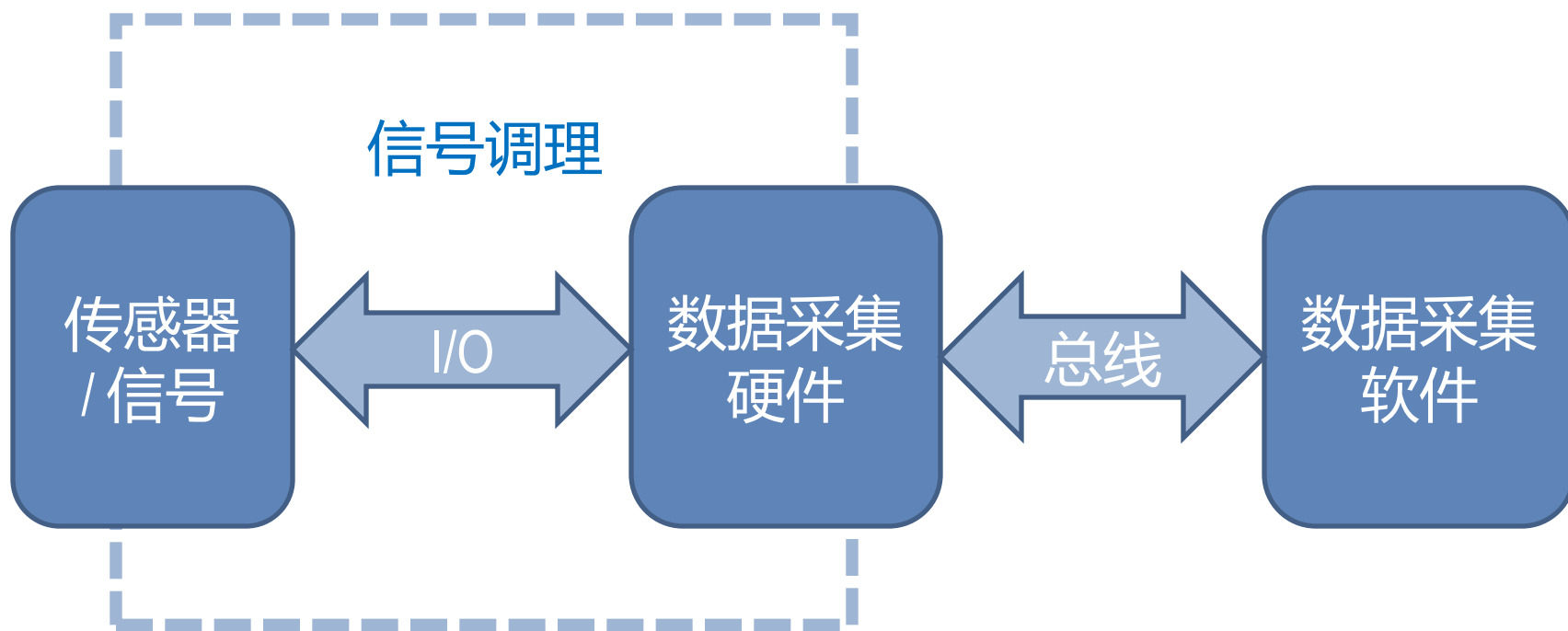
# 偏置电阻



- 浮地信号源采用差分或NRSE方式时需要
- 通过偏置电阻为偏置电流提供入地通道
  - 推荐值10 k $\Omega$  and 100 k $\Omega$

# 信号调理

- 为了正确（或更精确地）测量某些传感器的输出或信号，有时需要信号调理
- 不同的传感器需要不同的信号调理



# 信号调理设备

- 外置式
  - 如NI SCXI
  - 需要再连接数据采集设备
- 与数据采集设备相结合
  - 如NI CompactDAQ平台以及基于PXI Express的SC Express等
  - 模块中已经集成了数据采集（A/D转换）功能



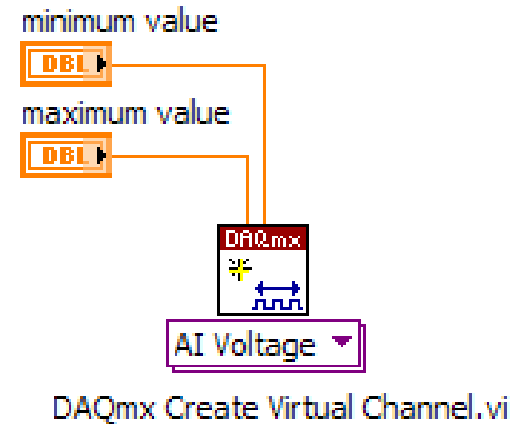
- 在软件方面，使用同样的DAQmx驱动
  - 正确配置系统和参数之后，用户编程时只需关心数据采集部分

# 针对电压信号测量的信号调理

- 放大
  - 针对小信号，为了尽可能用满ADC位数，提高信噪比（SNR）
- 衰减
  - 针对大信号，为了使测量信号范围在模拟输入通道范围之内
- 隔离
  - 通过电磁、光耦等方式使测量信号源与测量仪器没有直接电路相连
  - 可以抑制共模信号、解决接地回路问题、并保护仪器
- 滤波
  - 减小噪声、滤除干扰频率
  - 这里指的是前端硬件滤波，不同于后端数字滤波或软件滤波

# LabVIEW中的电压信号调理设置

- 与普通的数据采集程序基本无异
  - 只需配置并选择相应的信号调理设备通道，并正确设置电压范围以便驱动自动设定合适的增益





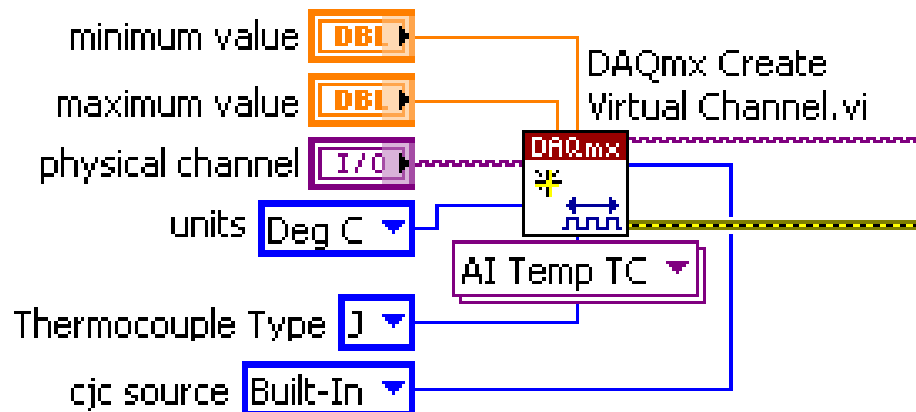
# 热电偶测量

- 热电偶信号调理模块

- 自动冷端补偿功能: 许多热电偶信号调理模块带有直接读取传感器, 可以自动读取冷端补偿温度, 并自动对热电偶测量结果进行补偿计算
- 热电偶模块通常还集成了滤波、放大、隔离等信号调理功能

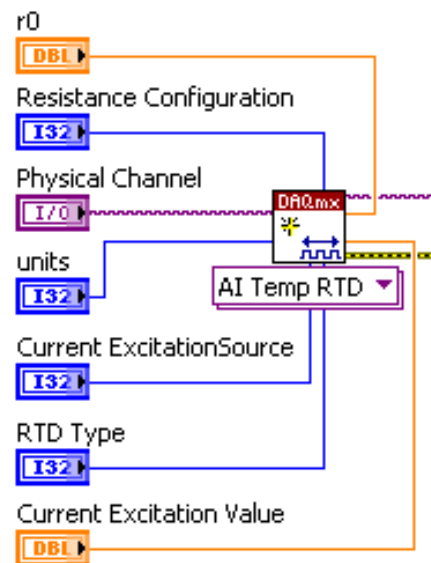
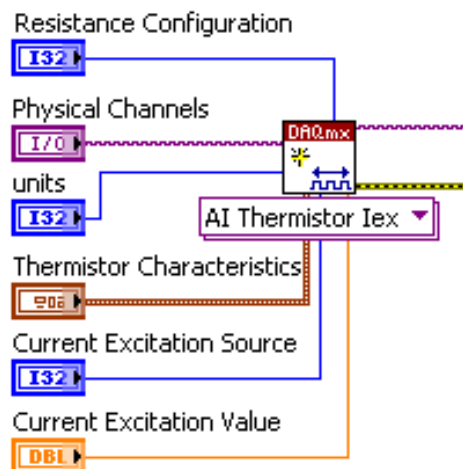
- LabVIEW程序

- 程序中只需在创建通道时做相应设置



# RTD与热敏电阻

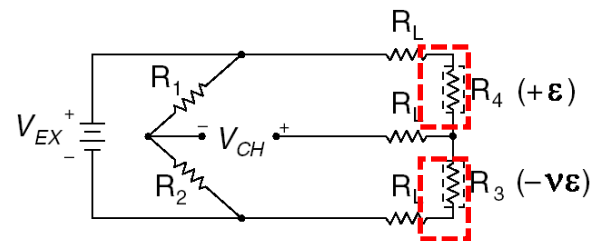
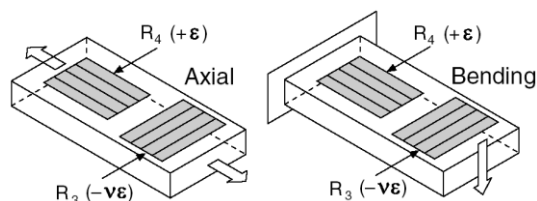
- 被动式的温度测量传感器
  - 需要来自外部源的电流或电压激励
  - 利用金属或金属氧化物的电阻与温度有关的原理
  - 测量时有不同的接线方式: 2-线、3-线、4-线
  - 除了激励之外对RTD与热敏电阻的信号调理还包括滤波和放大



# 应变测量电路

- 应变片接入惠斯通电桥的接法
  - 有1/4桥、半桥、全桥等不同方法
  - 具体参见 <http://zone.ni.com/devzone/cda/tut/p/id/10636> 及相关资料

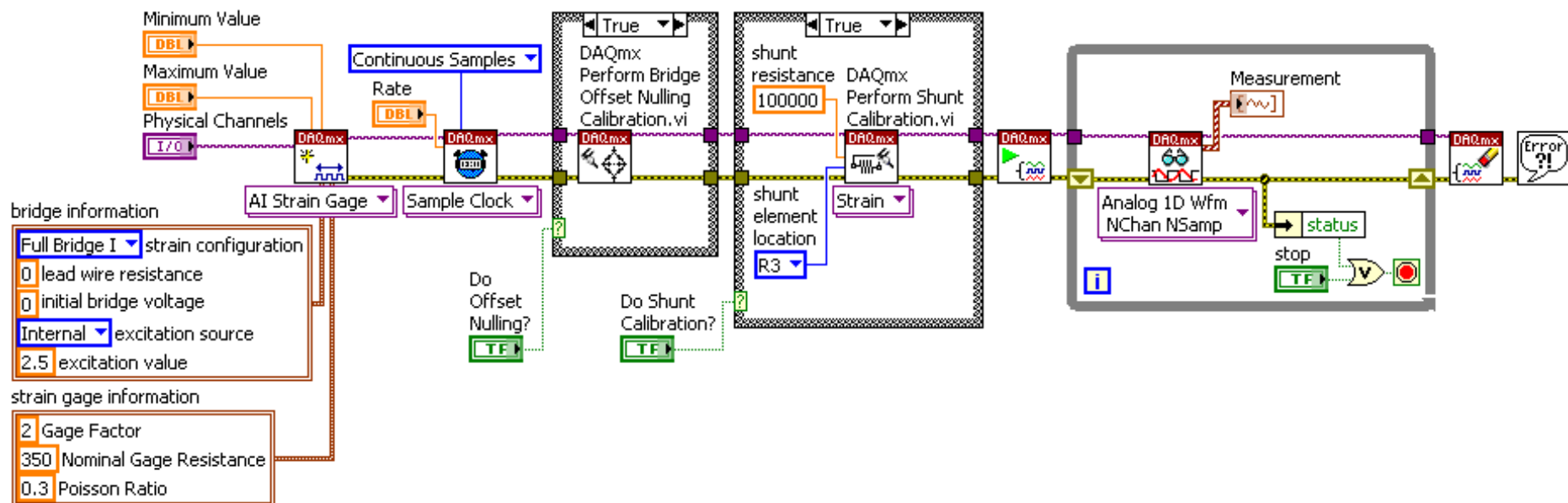
例: 半桥型接法



- 应变测量中的信号调理
  - 桥接电路及激励
  - 负载端电压采样
  - 失调清零
  - 分流校准
  - 放大/滤波

# 应变测量LabVIEW程序举例

- 配置电桥信息和应变计信息
- 可选进行失调清零和分流校准



# 负载、压力和扭矩测量



- 负载（称重）
  - 使用一组应变计测量结构的形变，该形变与力成比例
- 压力传感器
  - 使用固定在隔板的应变计测量隔板的形变，该形变与压力成比例
- 扭矩传感器
  - 使用附加在扭杆的应变计测量扭杆的剪应力，该剪应力与扭矩成比例

# 声音与振动测量

- 典型应用

- 噪声、振动与舒适度检测 ( NVH )

- 例: 用麦克风阵列进行噪声源定位

- 声学测试

- 例: 音箱声音品质测试

- 机器故障监测与诊断

- 例: 旋转机械的故障诊断

- 传感器

- 声压传感器 ( 麦克风 ) : 测量声压

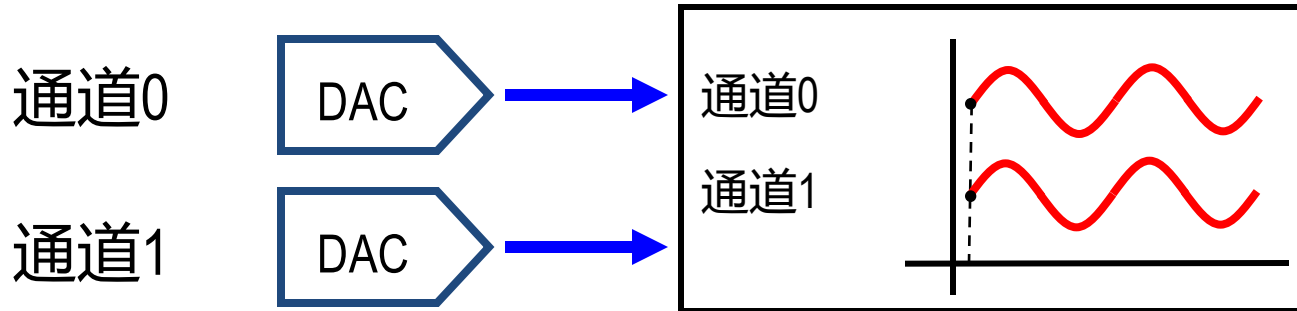
- 加速度传感器 : 根据压电理论测量加速度



# 声音与振动测量中的信号调理及程序

- 信号调理
  - 电流激励（IEPE激励）
  - AC耦合可消除DC偏置
  - 针对不同信号范围的灵活增益设置
    - 声音与振动信号的动态范围较大
  - 低通滤波器消除噪声和防止混叠
- 采集程序与普通输入采集程序几乎无差别
  - 只需对信号调理选项做相应配置（如AC/DC耦合、IEPE电流激励设置等）

# 模拟输出



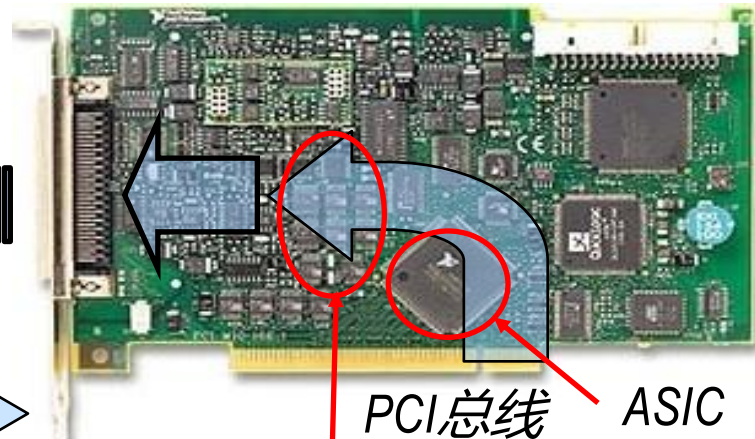
- 大多数多功能DAQ设备的每个模拟输出通道有一个DAC (同步更新)
- 与模拟输入的同步采集类似



# 输出操作的数据传输

输出  
频率

数据

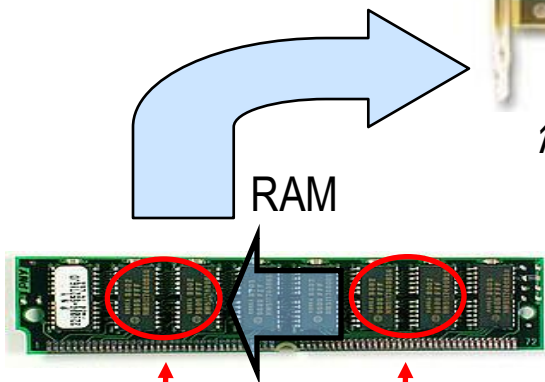


传输速率

PCI总线

ASIC

板载FIFO



RAM

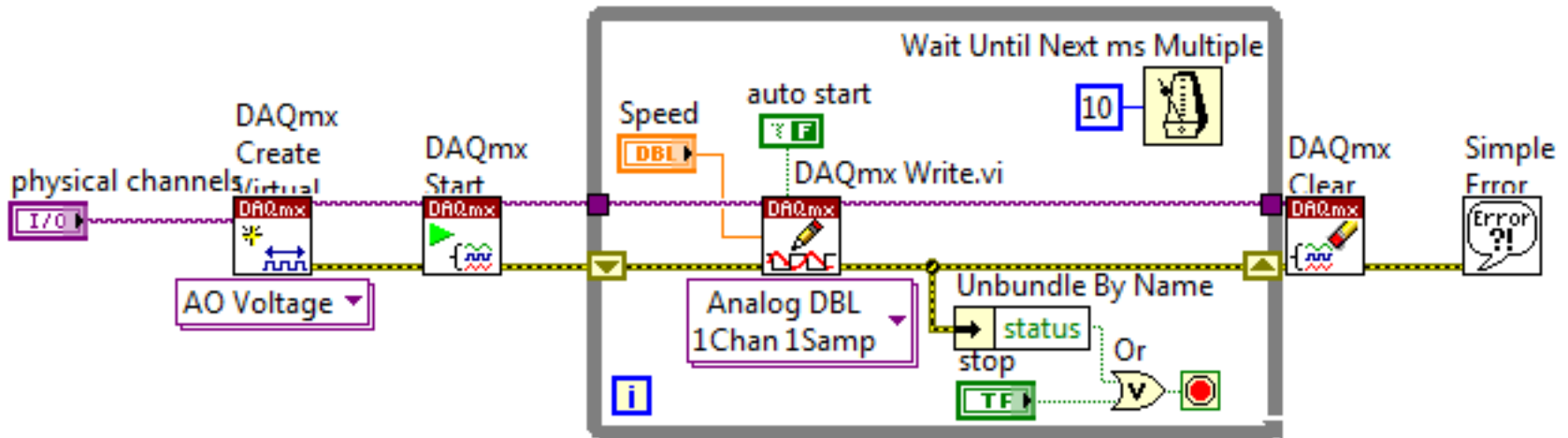
PC  
缓冲区

应用程序  
内存

LabVIEW

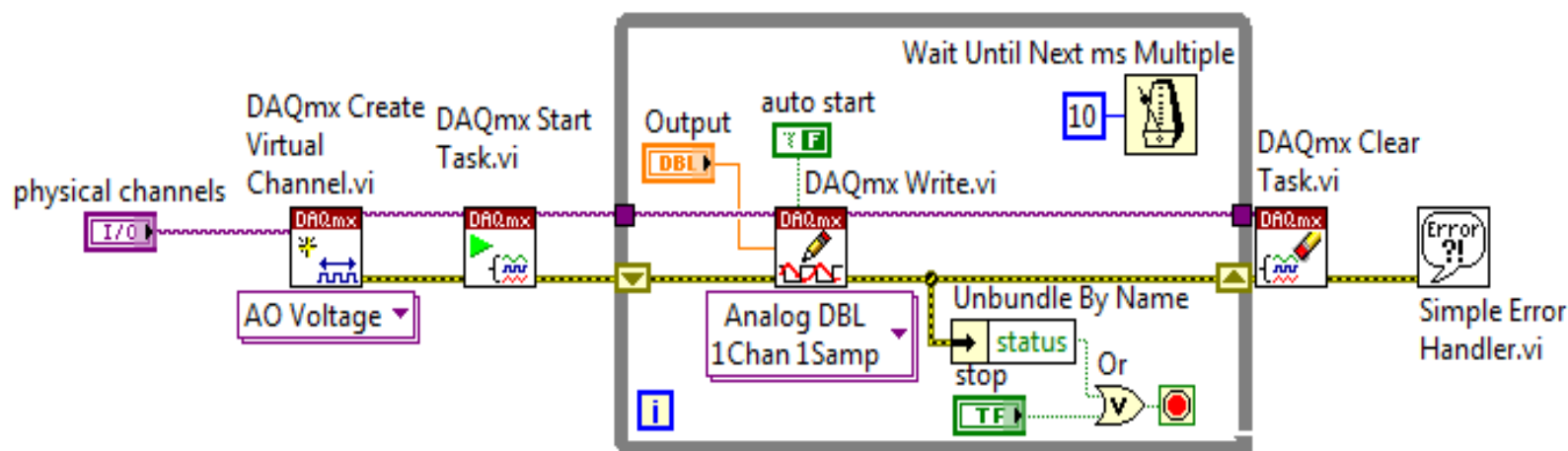
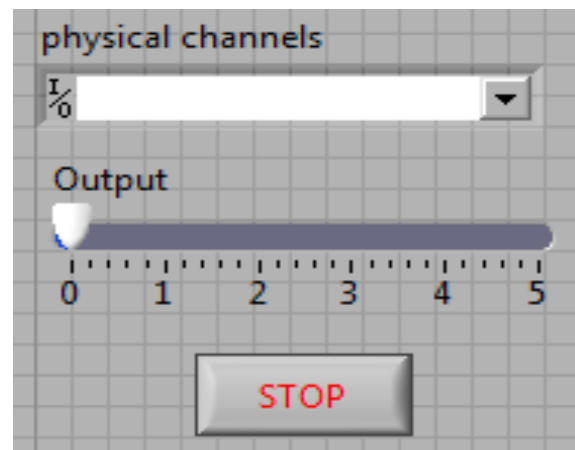


# 软件定时的模拟输出



- 软件定时
  - 速率决定于操作系统或程序（在生成循环中添加延时）
- 硬件定时
  - 设备上的时钟控制定时，比软件定时更快更精确

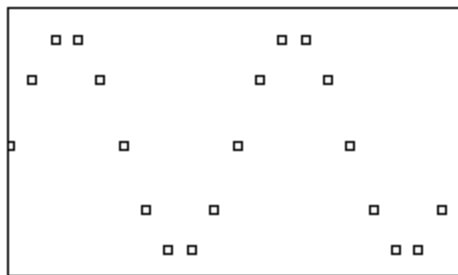
# Lab 4.1 单点模拟输出



# 带缓冲的波形生成

- 生成波形频率取决于下列三个因素
  - 更新率 (每秒多少个更新点)
  - 缓冲区中的数据点
  - 缓冲区中的周期数

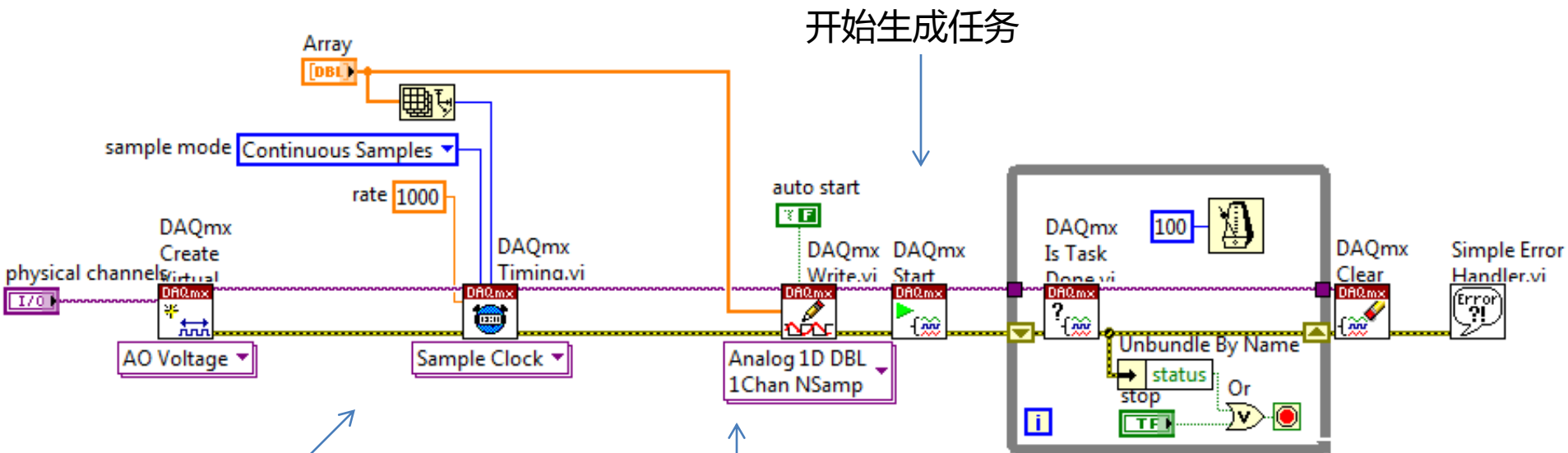
$$\text{信号频率} = \text{缓冲区中的周期数} \times \frac{\text{更新率}}{\text{缓冲区中的数据点}}$$



缓冲区大小 = 1000点  
缓冲区中的周期数 = 2  
更新率 = 1000 点/秒  
则, 信号频率 = 2 Hz

# 使用采样时钟的连续波形生成

- 硬件定时



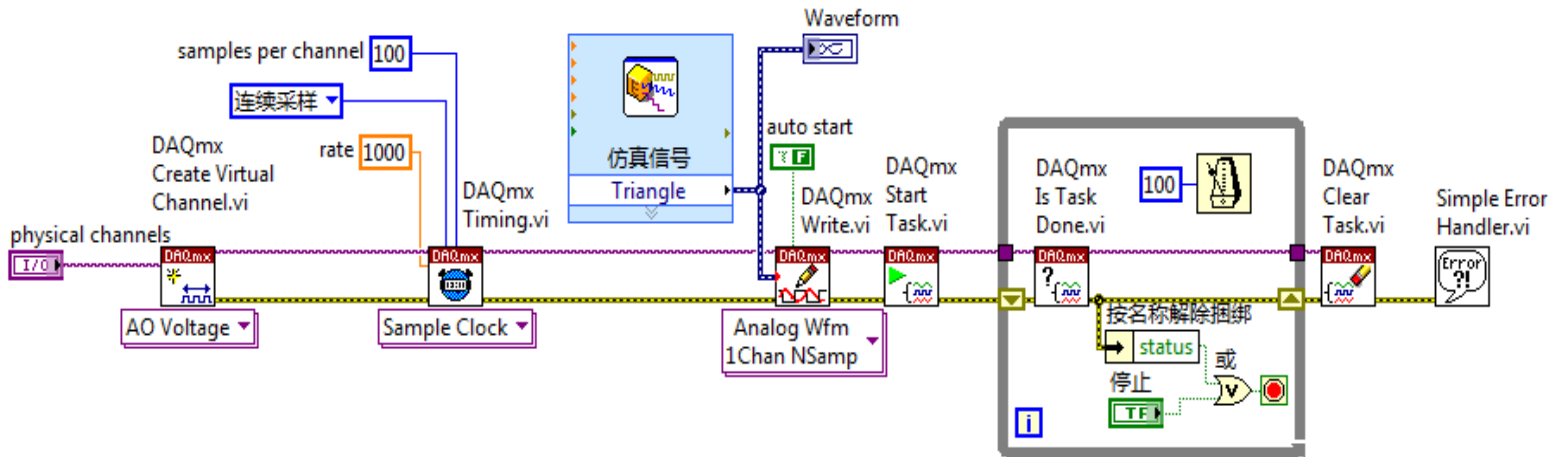
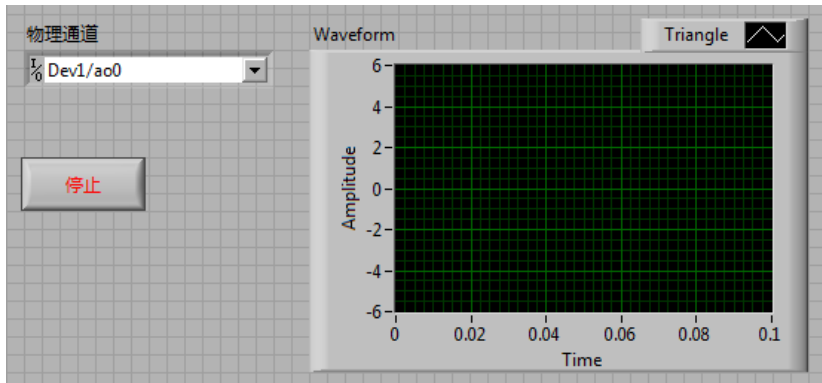
开始生成任务

基于时钟的硬件定时

写入生成数据

与模拟采集不同,循环在这里起的作用仅仅是不断检查任务状态,而非不断写入数据

# Lab 4.2 硬件定时的连续模拟信号输出



# 数字I/O

- 按照电平标准和电流驱动能力分类
  - TTL
  - LVTTTL(低电压)
  - LVDS (利用差分技术)
  - 工业数字I/O (如12V, 24V, 48V等)
- 需根据电平标准、驱动能力、所需速率等因素选用不同的数字I/O板卡
  - 大多数多功能数据采集卡上的数字I/O通道电平与TTL兼容
  - 课程中所用到的NI ELVIS及NI myDAQ上集成的数字I/O通道也是与TTL兼容

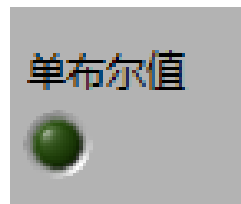
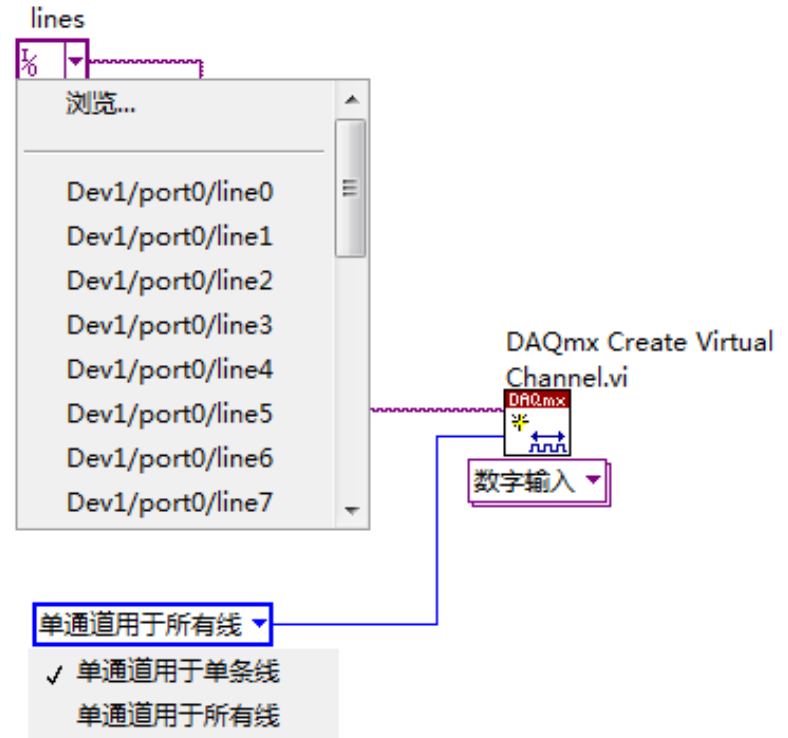
# 数字术语

- 位            数据的最小单位，每一位为1或0
- 字节        包含8位数据的二进制数
- 线            端口中的一路独立信号，位表示传输的数据，线是“位”在硬件上的表示
- 端口        数字线的集合（通常4或8路）
- 端口宽度    端口的数字线数目（通常4或8）

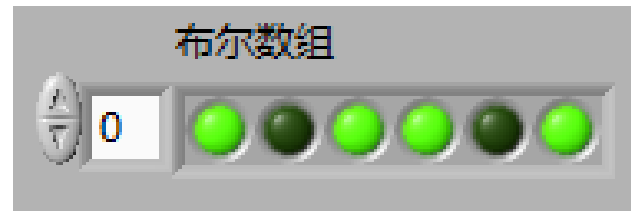


# 通过DAQmx API创建数字虚拟通道

- 创建一个端口、线或线集合的数字通道
- 选择如何将数字线编组为一个或多个虚拟通道
  - 影响DAQmx读取VI的配置



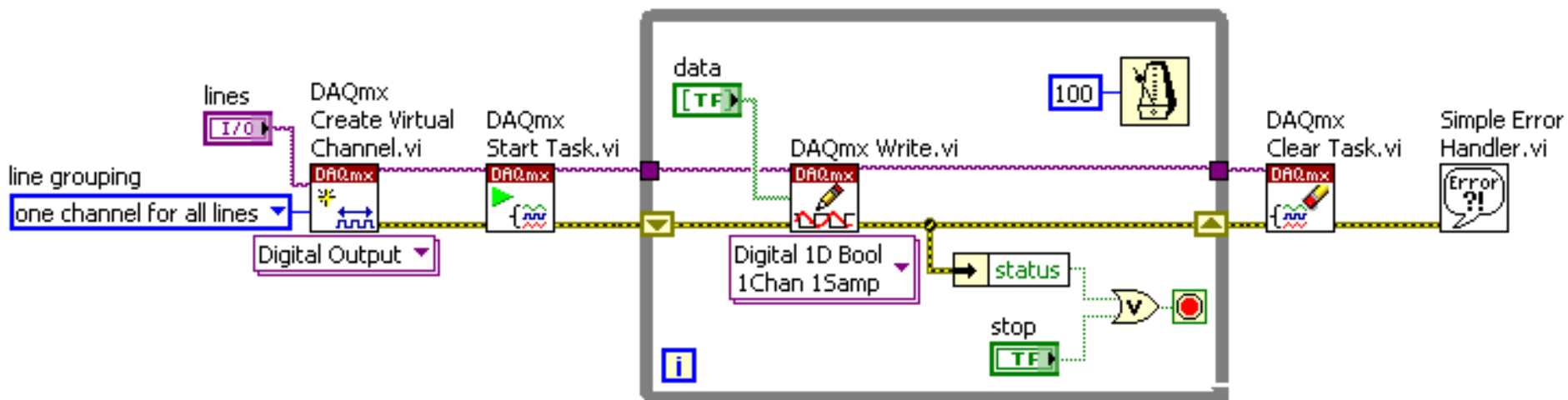
线格式



每个通道多条线

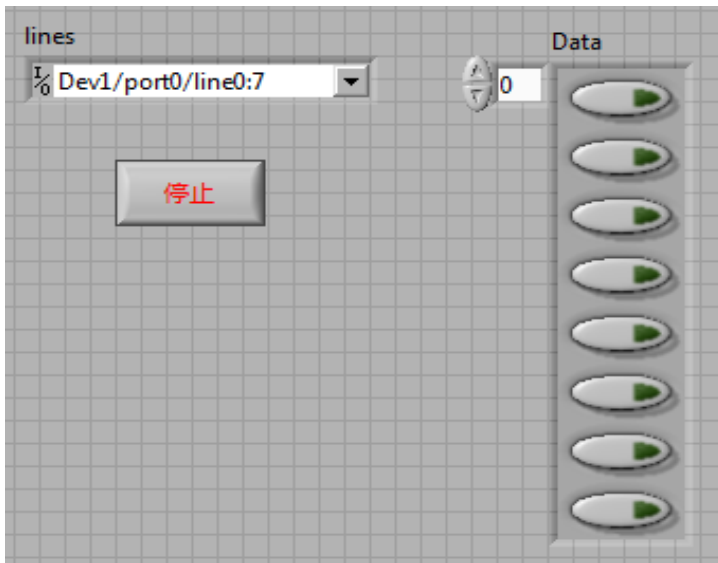
# 数字输出

- 软件定时 (Static Digital I/O)



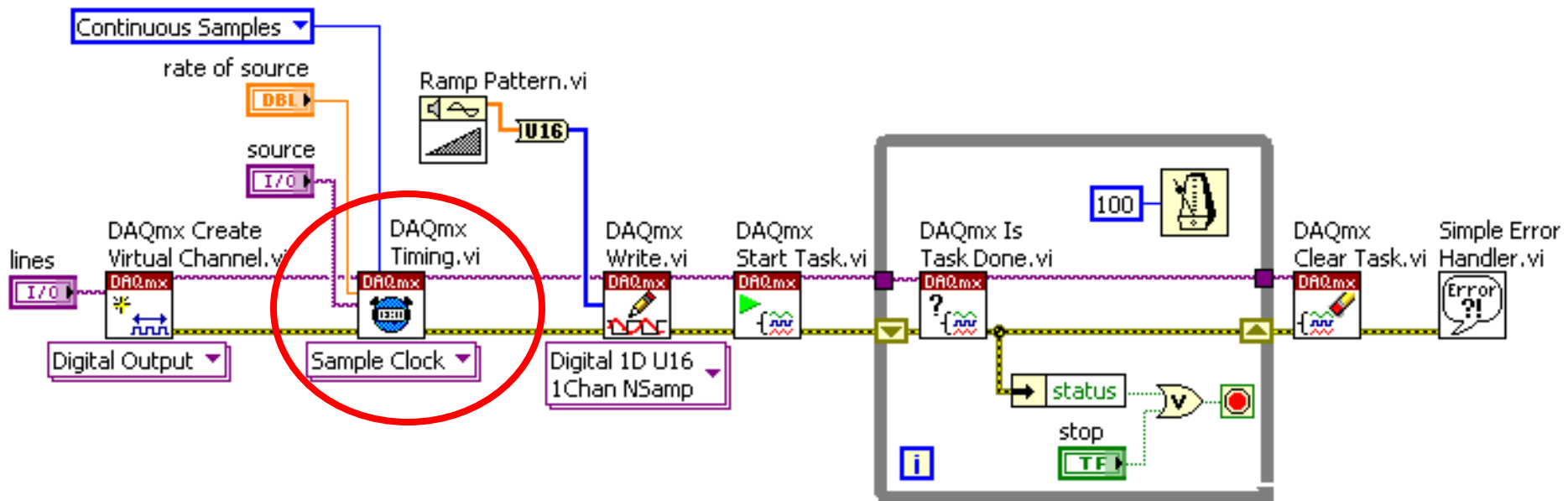
# Lab 5.1 数字输出

# Lab 5.2 数字输入



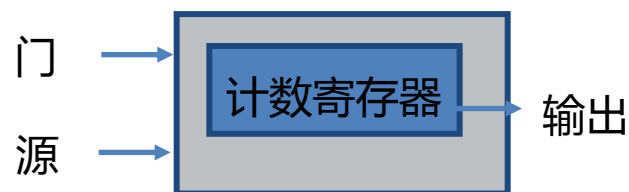
# 硬件定时的数字I/O

- 某些DAQ设备支持硬件定时的数字I/O
  - 与带缓冲的模拟输入输出原理相同
  - 采样时钟用于硬件定时的数字I/O (Correlated Digital I/O)
  - NI ELVIS的数字I/O端口Port1/Port2支持硬件定时，Port0只支持软件定时



# 计数器

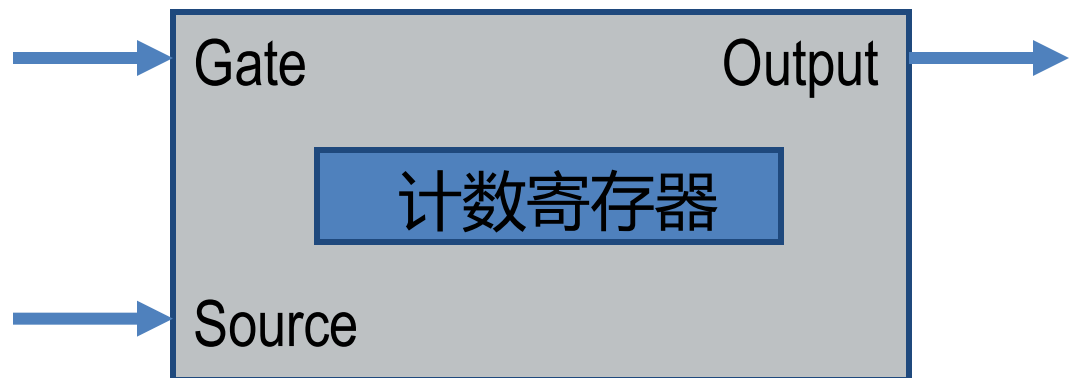
- 两个基本功能
  - 基于输入信号（门和源）的比较，进行计数
  - 基于输入和寄存器值，生成脉冲
- 许多应用由基本计数演变而来
  - 边沿计数，例如简单边沿计数和时间测量
  - 脉冲宽度、半周期和周期测量
  - 频率测量
  - 单脉冲和脉冲序列生成
  - 位置和速度测量



注: 由于课时限制，本课程中主要介绍 脉冲边沿计数、脉冲宽度测量、固定频率连续脉冲串生成 这三种应用，更多计数器的应用可以参考DAQmx帮助及LabVIEW中的范例程序自学

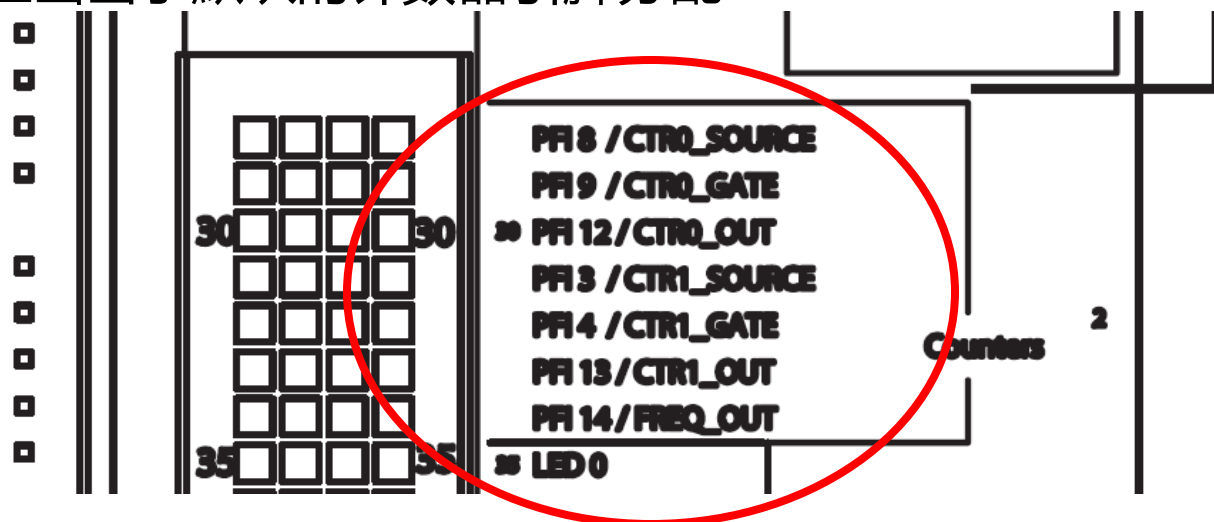
# 计数器的硬件组成

- 计数寄存器
  - 保存当前计数值
  - ELVIS II有两个32位的寄存器
  - 归零前最终计数 =  $2^{32} - 1$
  - 可设置每次计数寄存器加1或减1（向上计数或向下计数）
- 源 (Source)
  - 相当于计数时钟
  - 可设置上升沿或下降沿有效
- 门 (Gate)
  - 相当于使能控制信号
- 输出 (Output)
  - 用于生成脉冲



# 计数器引脚与接线

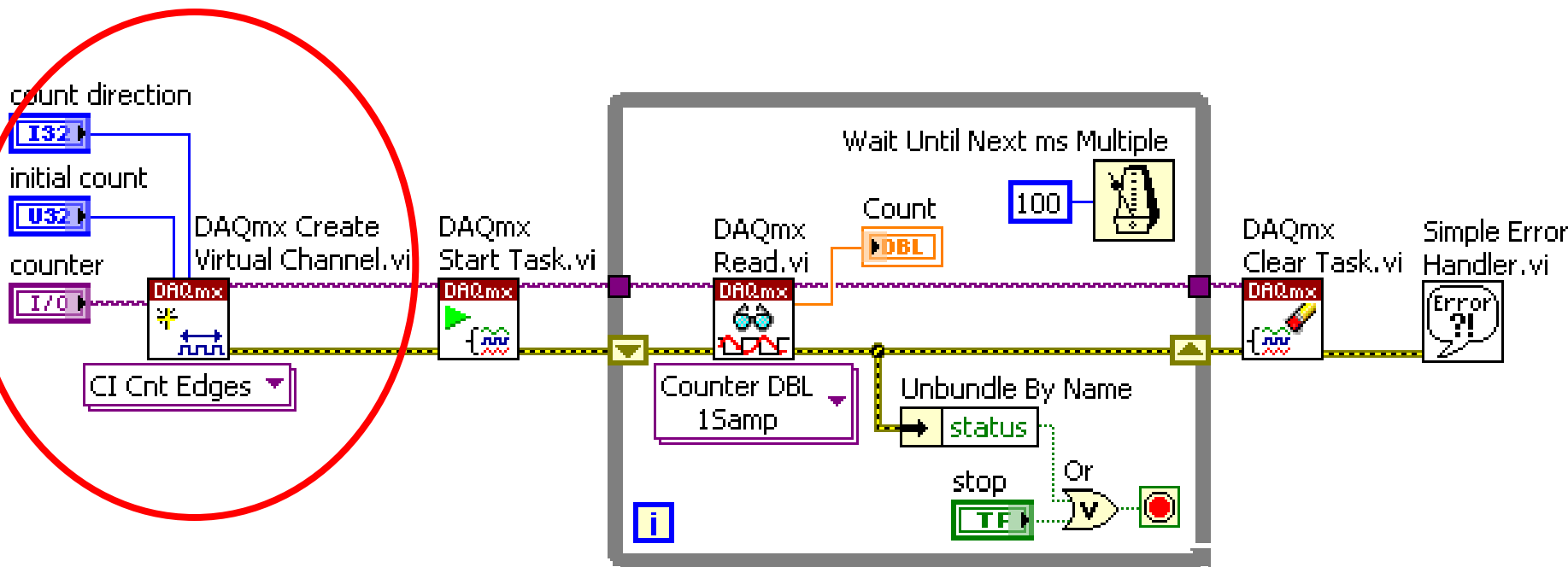
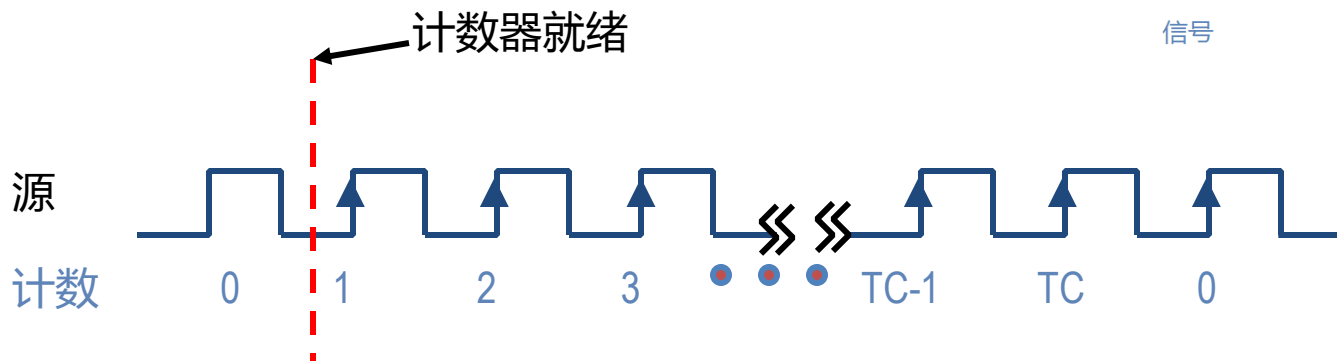
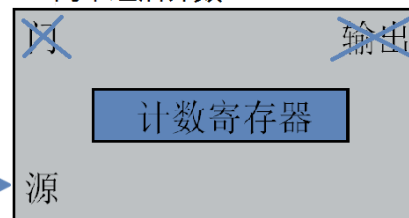
- 所有计数器相关应用的接线都是通过PFI引脚
  - 一个PFI引脚可有多种用途（例如既可以作为计数器的源信号又可以作为模拟采集的触发信号）
  - NI ELVIS的板子上画出了默认的计数器引脚分配



- 可以通过DAQmx帮助中Counter的部分查看针对不同计数器应用的信号连线方式（更方便的方式是在MAX中建立一个任务，DAQ助手会自动显示默认的正确引脚连线方式）

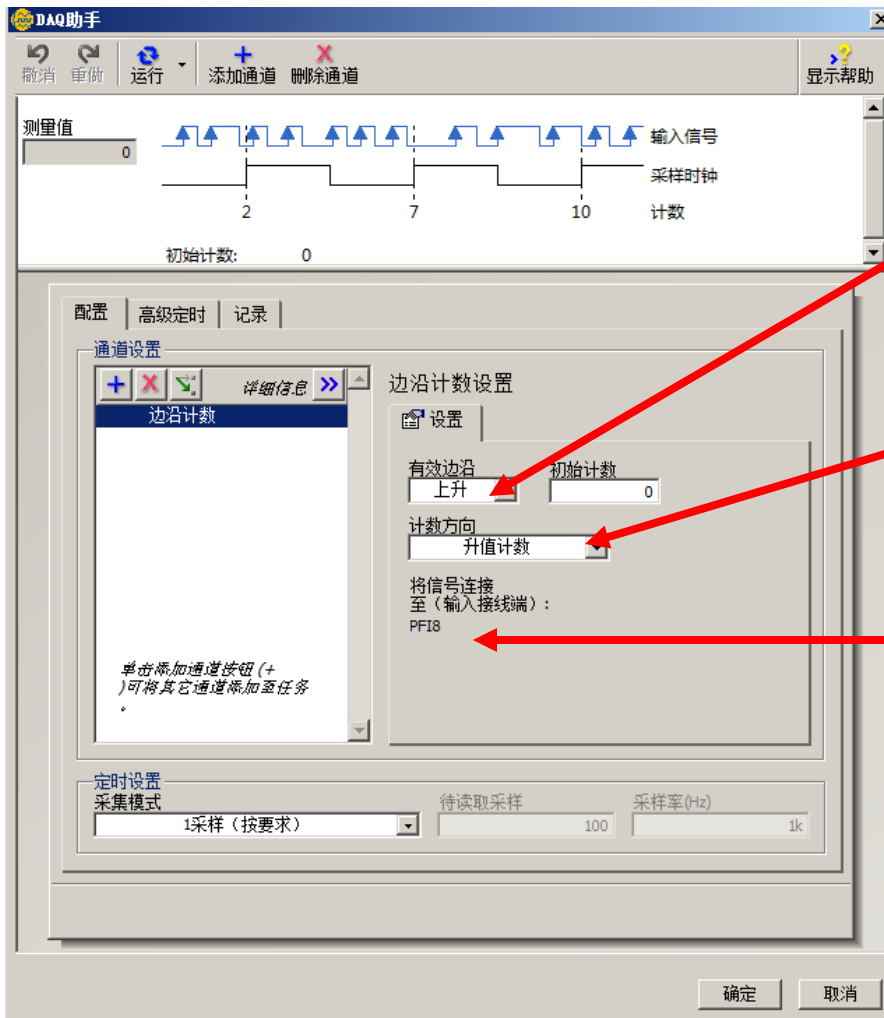
# 最简单的边沿计数

简单边沿计数





# 利用DAQ助手实现最简单的边沿计数



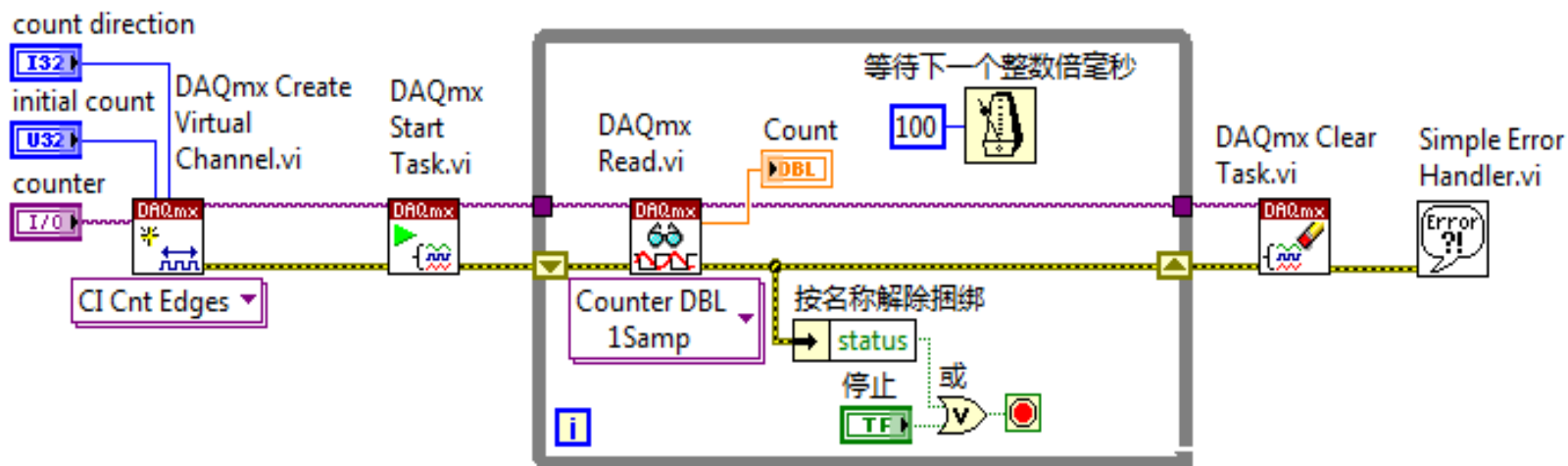
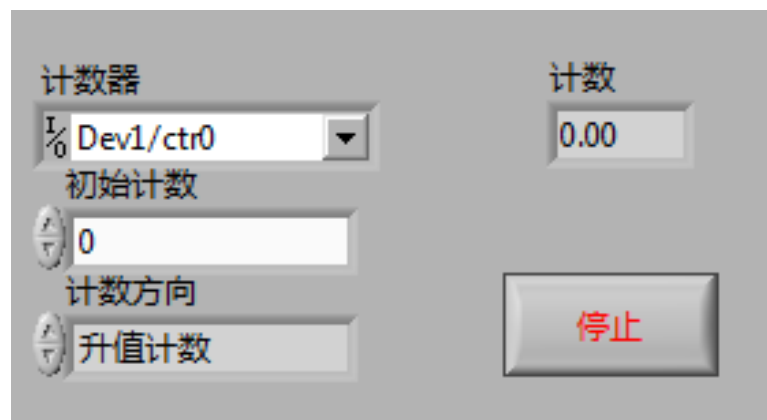
选择上升沿或下降沿

选择计数方向：  
升值计数  
减值计数

自动选择PFI线

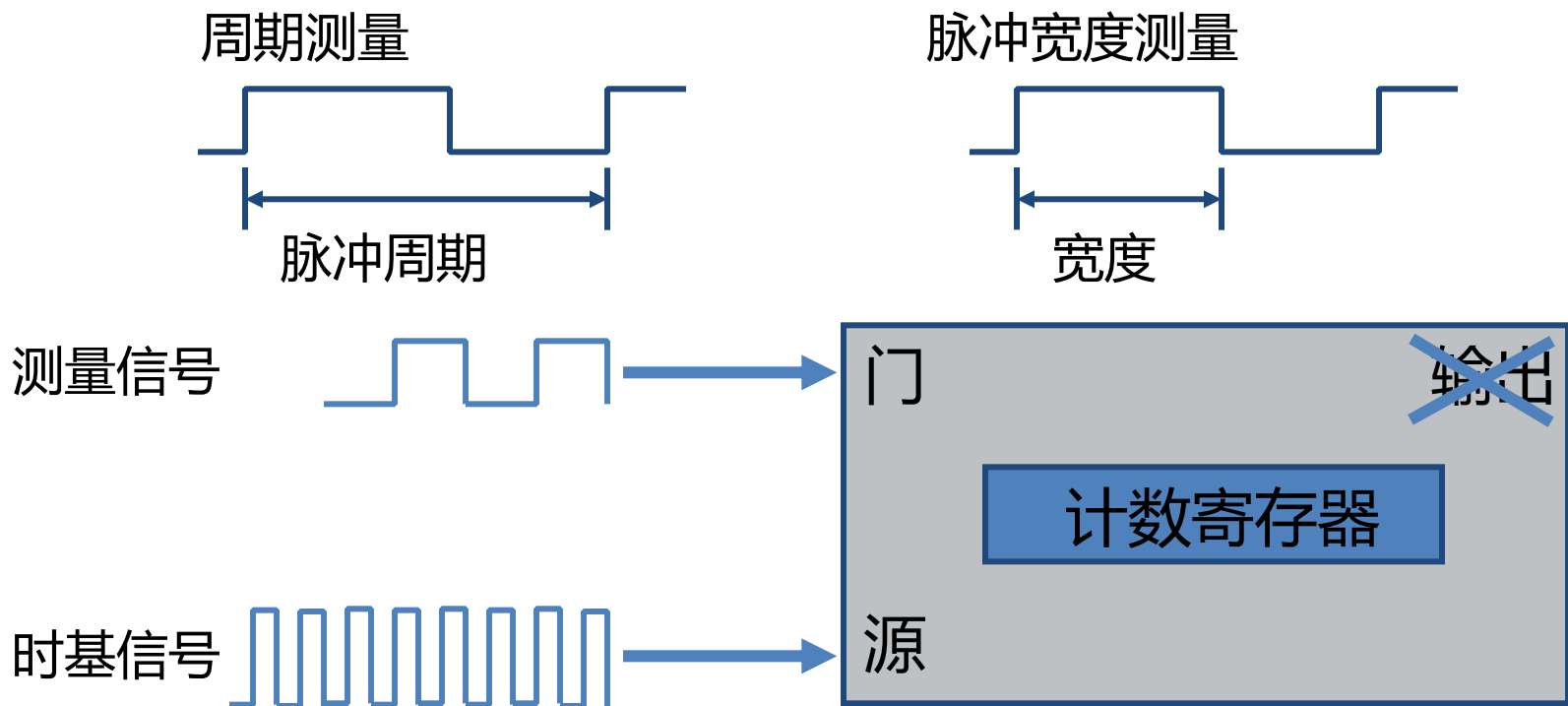
- 利用DAQ助手可以更直观地看到各种基于计数器应用的原理，同时方便地进行相关配置
- 适合作为参考，鼓励仍然基于DAQmx API实现

# Lab 6.1 底层API实现简单边沿计数



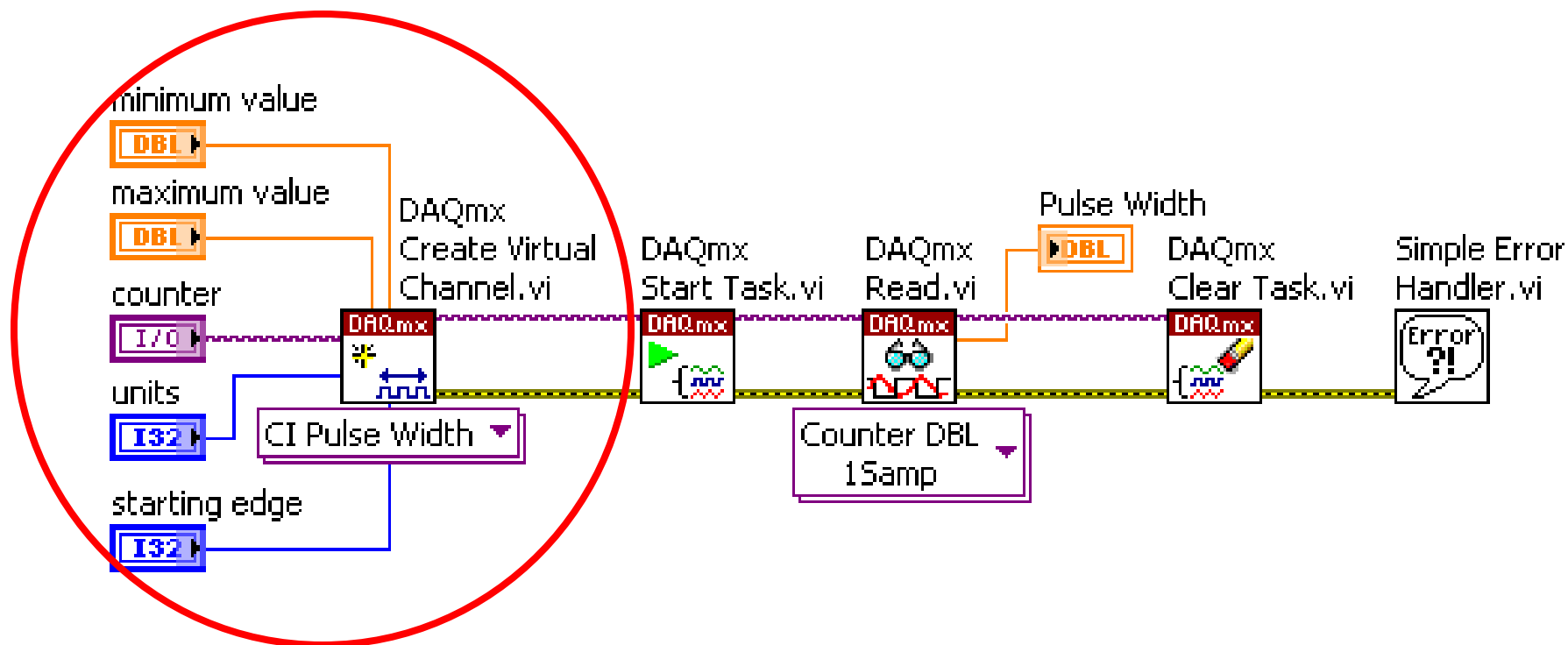
# 脉冲测量

- 使用已知频率的时基测量未知信号的特性
  - ELVIS中计数器自带的时基可选80 MHz, 20 MHz及100 kHz

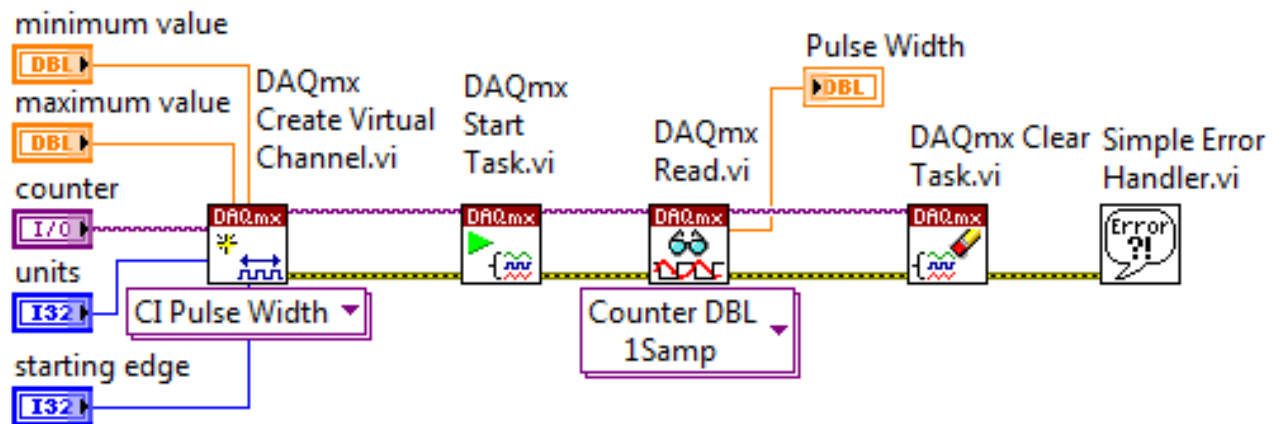
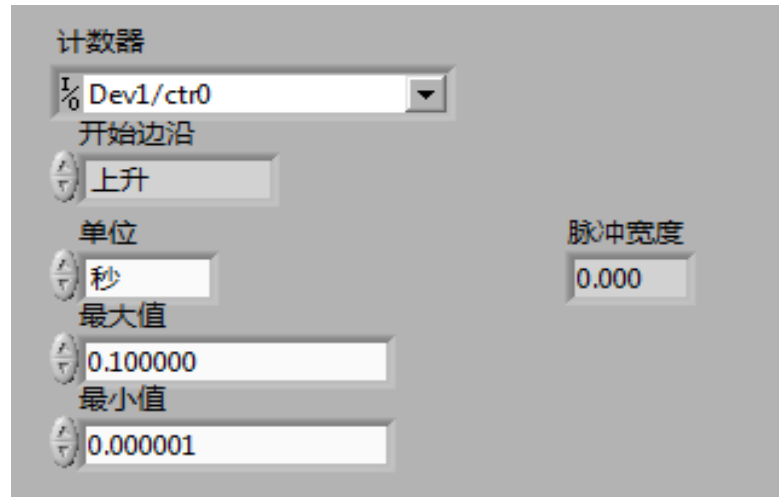


# 例：单脉冲宽度测量

- 理解测量的原理很重要
  - 实际的程序中，DAQmx驱动已经帮我们进行了底层的计算，所以只要选择相应的功能并配置相关参数即可



# Lab 6.2 脉冲宽度测量



# 脉冲生成

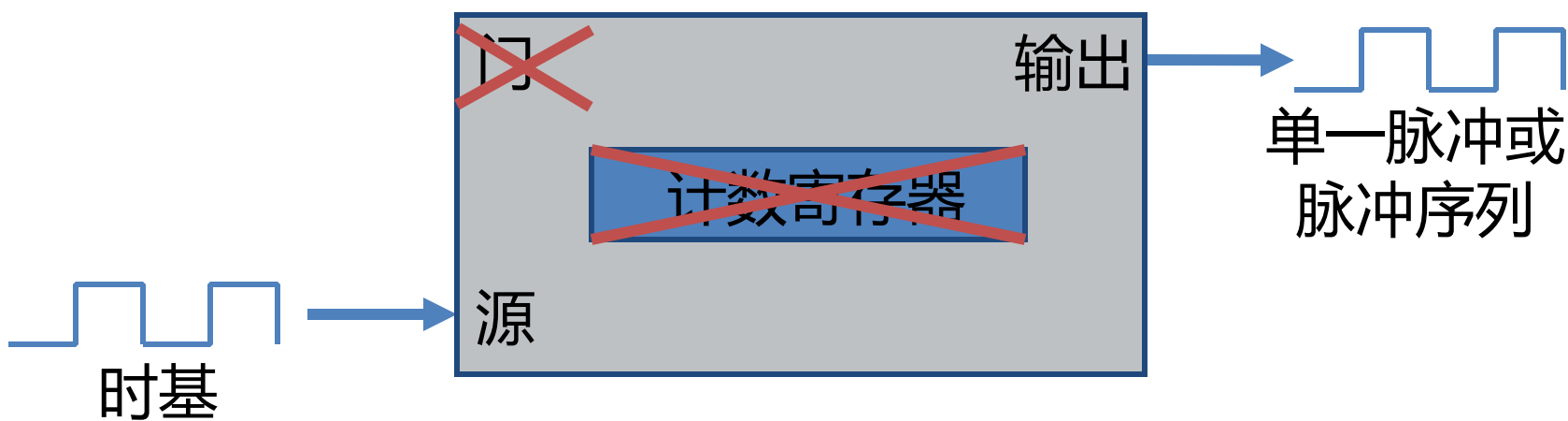
- 在计数器的输出引脚上生成TTL脉冲或脉冲序列



单个脉冲



脉冲序列



# Lab 6.3 连续脉冲序列生成

计数器

Dev1/ctr0

占空比

0.500000

频率

1.000000

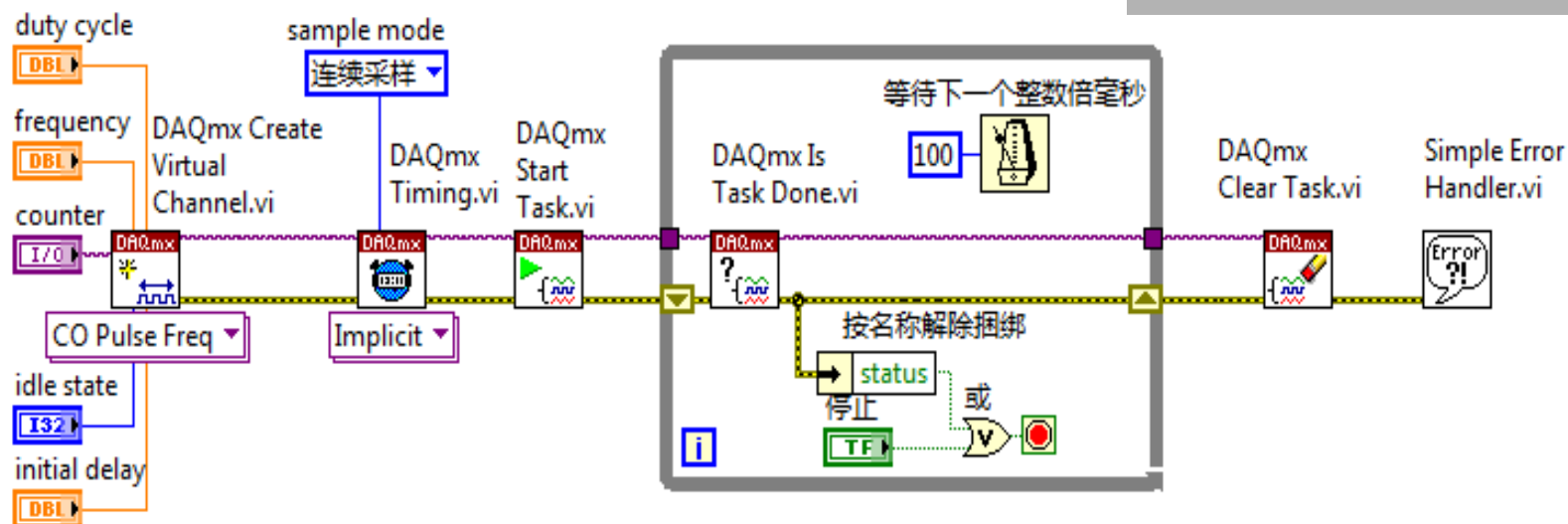
空闲状态

低

初始延迟

0.250000

停止



# 计数器的更多功能

- 频率测量
  - 根据频率范围有3种不同测量方式
- 周期测量
- 正交编码器测量
- 有限脉冲序列生成
  
- 如何应用?
  - 可参考相关LabVIEW范例
  - 利用DAQ助手可以快速了解接线方式并进行验证（演示）

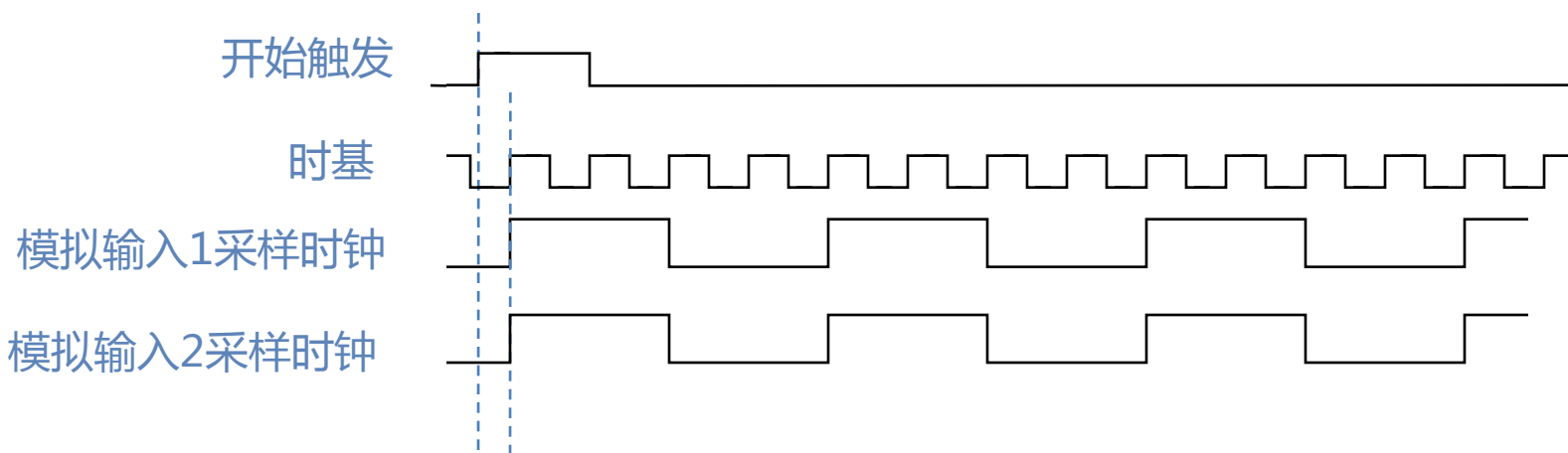


# 同步简介

- 许多数据采集应用中要求同步测量
- 概念
  - 同时测量
    - 操作同时发生，但无须严格同步
    - 无需证明测量在同一时刻发生
    - 如同发令枪响，同时开始跑，但是步伐不一定一致
  - 同步测量
    - 测量是相关的
    - 如同“齐步走”，同时开始，步伐也一致

# 同步原理

- 共享一个主时基和触发信号



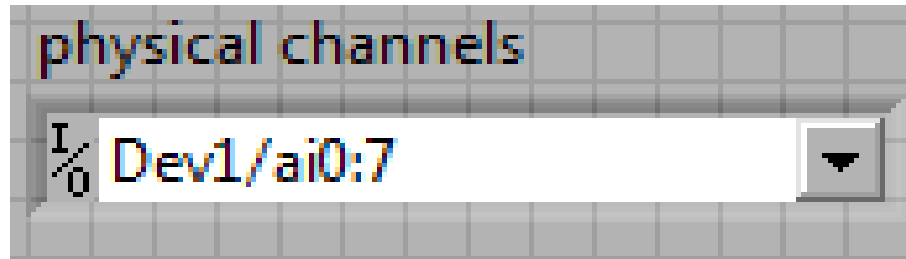
- 采样时钟来自于同一个时基
  - 对于单块板卡，时基总是相同的
  - 多于多块板卡，必须共享时基以消除相位误差

# 同步误差来源

- 抖动 ( Jitter )
  - 抖动是时钟周期之间 ( 两次采样之间 ) 的微小差别
  - 在时钟路径上添加各种元器件均会导致抖动
- 时钟稳定性
  - 易受温度、老化等变化因素影响
  - 振荡器产生的频率不是绝对精确的频率
  - 误差通常用ppm ( 百万分之一 ) 和ppb ( 十亿分之一 ) 为单位
- 偏度 ( Skew )
  - 同一时钟信号到达两路不同位置时产生的传输延迟
  - 受信号路径的距离和阻抗影响

# 同一板卡多通道同步

- 同种通道同步非常容易实现
  - Lab 3.2 中已经提到



- 不同种通道的同步
  - 例如 AI 和 AO、AI 和 Counter
  - 不需要外部连线，可以仅通过软件编程控制内部时钟信号和触发信号的路由
  - 具体可参考LabVIEW相关范例