UiO : University of Oslo

FYS3240- 4240
Data acquisition & control

PC-based data acquisition I

Spring 2019 – Lecture #8

Reading: RWI page 198 – 202, 395 – 405, 431 – 436 and 349 - 351.

Recommended additional reading: Essick Ch5.

Bekkeng, 26.02.2019

General-purpose computer

- With a Personal Computer (PC) we mean a **general-purpose computer**. Such systems are easy to expand with more memory, and more I/O ports etc.
- A PC is designed to be able to run all kind of application programs that you can buy or intend to develop. A general-purpose computer need to be ready for new device drivers and software to run hardware it doesn't know about yet, like new printers or hard drives, and it need to run different application programs.
- The PC usually need to run several programs at the same time on the CPU by sharing CPU time between the different applications (multitasking), or by running different applications in parallel on different CPUs or different CPU cores. The typical PC today have four CPU cores, but can have up to 16 cores (for instance AMD Ryzen Threadripper).

Workstation vs. Desktop PC

- A **workstation** is a high-end computer designed for technical or scientific applications, running numeric- and graphic-intensive applications
  - high-performance computing (HPC)
- **Desktops PCs** are all about flexibility – they are designed to meet a much wider variety of computing needs
- As desktop PCs become faster and stronger, the lines between a high-powered desktop and true workstations become more blurred
- Workstations typically offered higher performance than personal computers, especially with respect to **CPU** and **graphics**, **memory capacity** and multitasking capability
- In addition, workstations typically have more slots for PCI and PCI Express

Workstation vs. PC (2011) examples

HP Z800 Workstation

HP Compaq dc7900

- 2 x Intel® Xeon® six Core processors (5600 series)
- 192 GB RAM (maximum)
- Rack-mountable (19" rack)
- 1 PCI
- 1 PCI Express Gen1 (x8 mechanical, x4 electrical)
- 1 PCI Express Gen2 (x8 mechanical, x4 electrical)
- 2 PCI Express Gen2 (x16 mechanical, x8 electrical)
- 2 PCI Express Gen2 x16 (for graphics card)

Update 2012: HP Z820: **16 cores**, up to **512 GB DDR3 RAM**, four internal HDDs

- Intel® Core™2 Quad processor
- 16 GB RAM (maximum)
- 3 PCI
- 1 PCI Express x16
- 2 PCI Express x1

Physical memory (RAM) limits for 64-bit Windows(X64).

- Windows XP : 128 GB
- Windows 7 : 192 GB
- Windows 8 : 512 GB
- Windows 10 : 2000 GB (= 2 TB)

Intel Architecture example

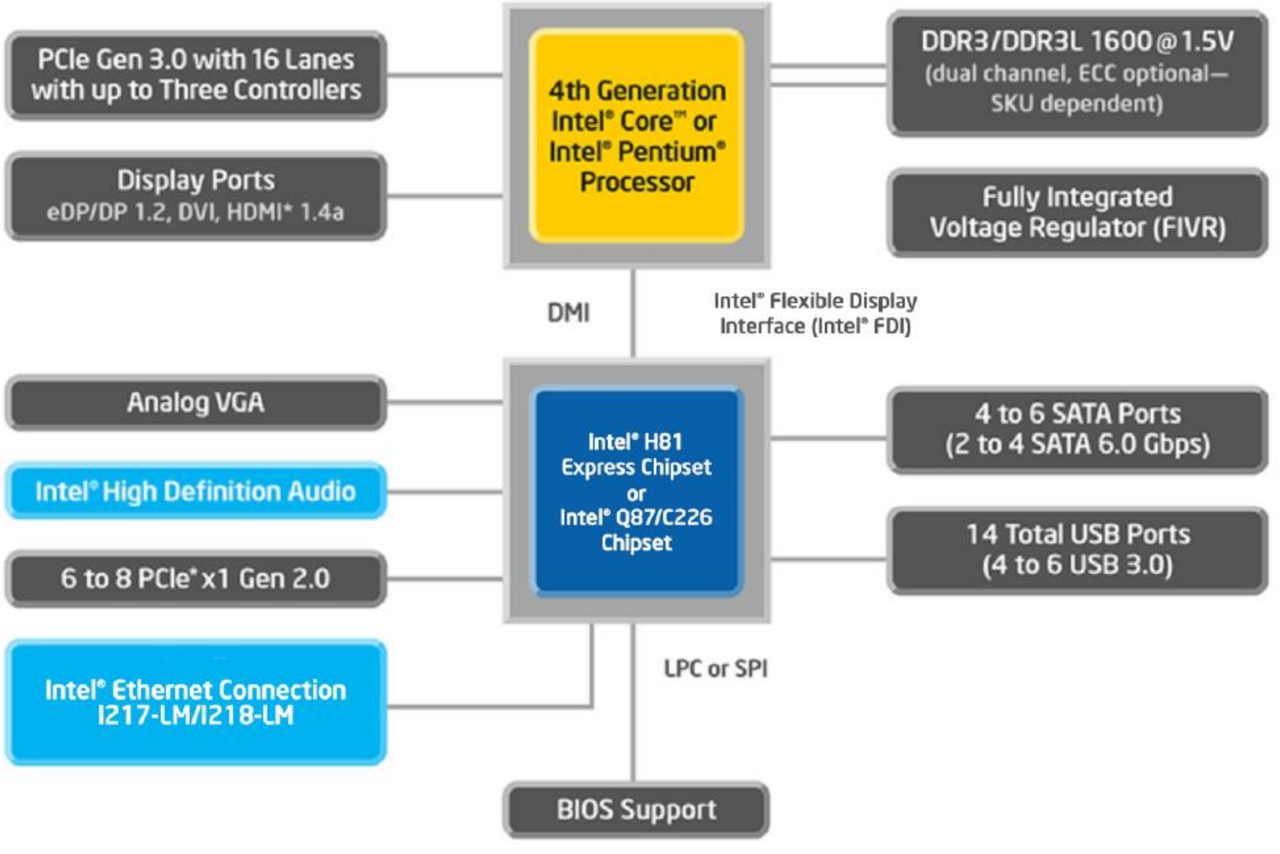


Figure 1: Typical system based on the Intel® Core™ i7 processor

http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ia-introduction-basics-paper.pdf

General Purpose Operating Systems (OS)

- Windows, Linux, MacOS, Unix
  - Processor time shared between programs
  - OS can preempt high priority threads
  - Service interrupts –keyboard, mouse, Ethernet…
  - <u>Cannot ensure that code finish within specified time limits!</u>

Data acquisition (DAQ)

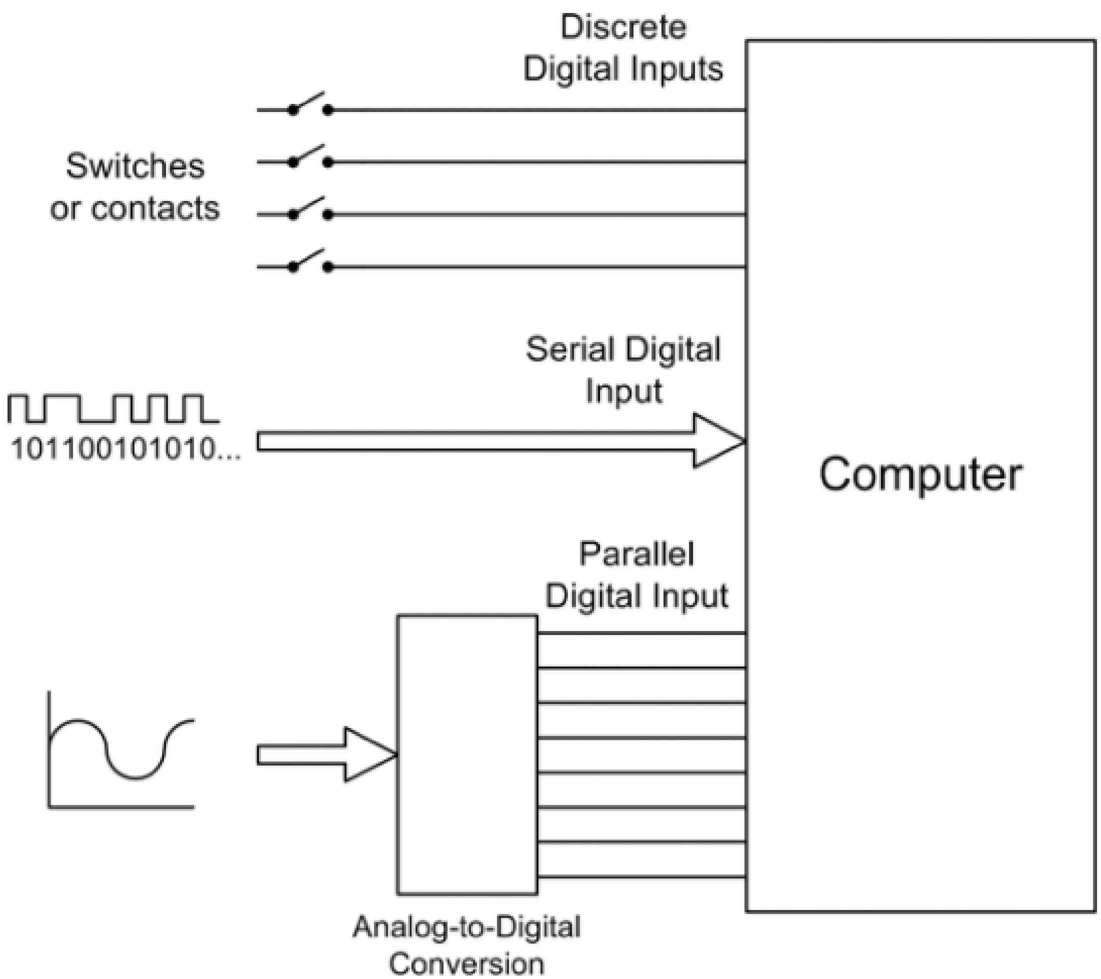


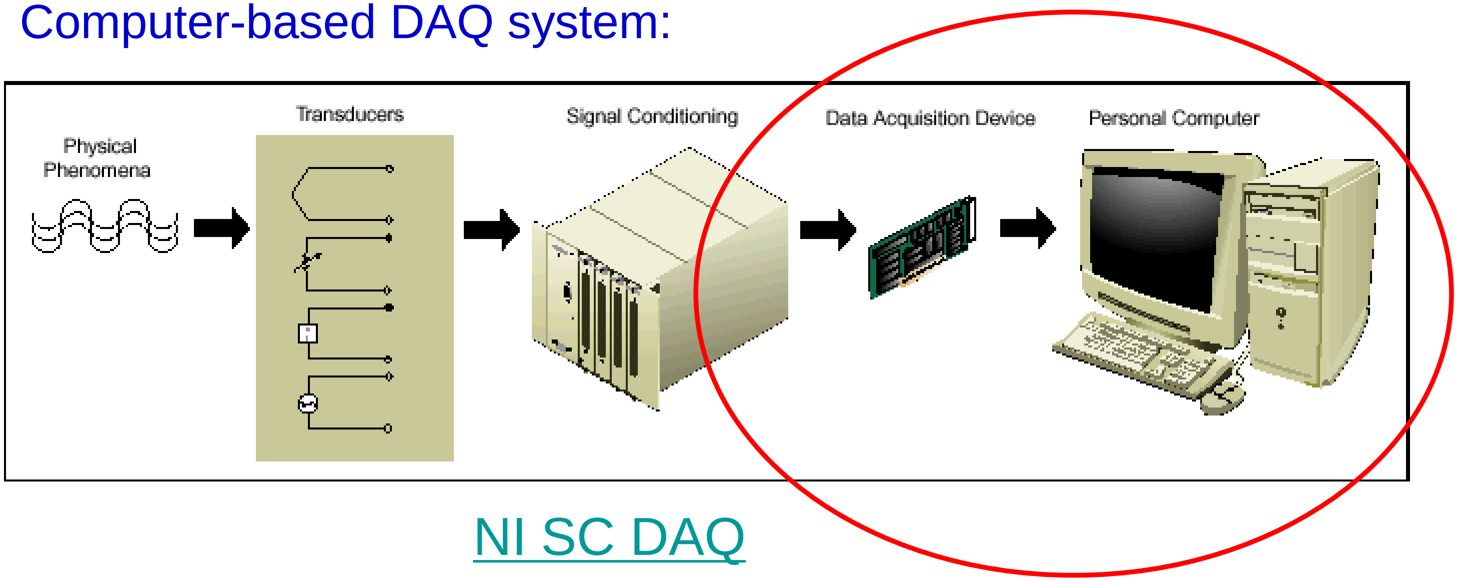
Figure 1-2. Digital and analog data inputs

Data acquisition (DAQ)

- **Data acquisition involves measuring signals (from a real-world physical system) from different sensors, and digitizing the signals for storage, analysis and presentation.**

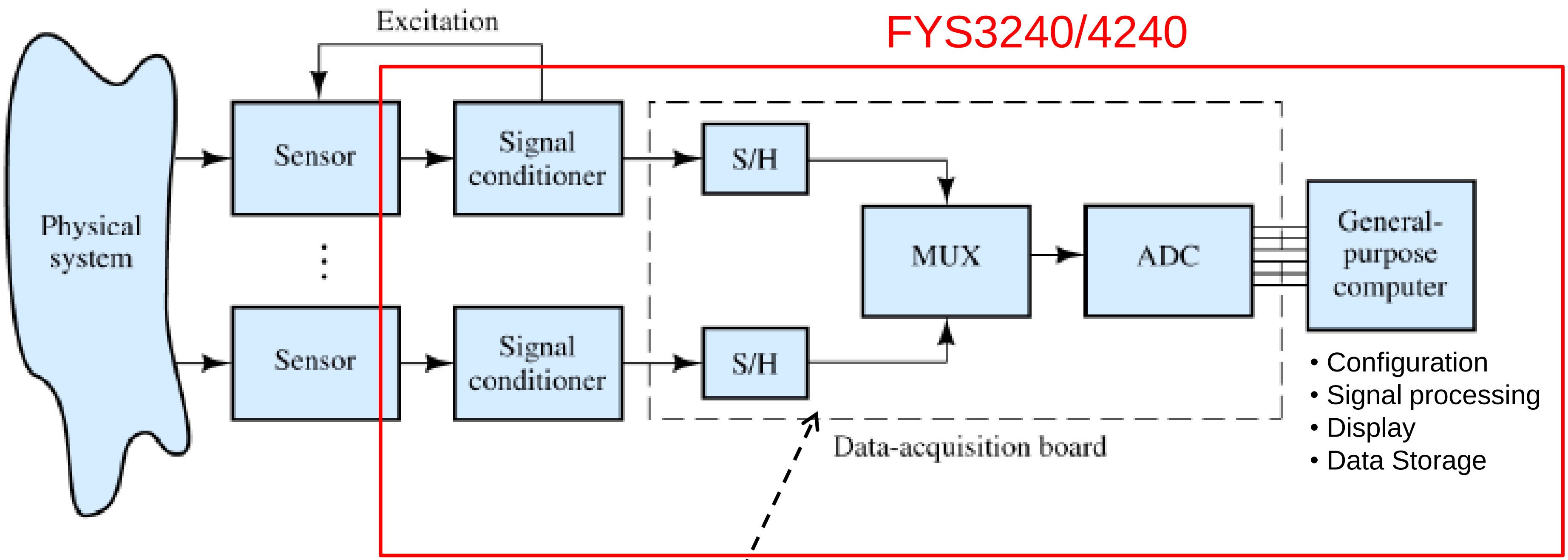
- Analog input channels can vary in number from one to several hundred or even thousands

Computer-based DAQ system:

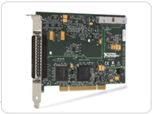


NI SC DAQ

Computer-based DAQ system

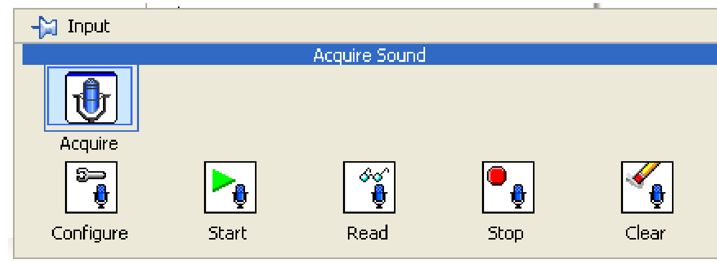


- PCI DAQ card
- USB DAQ box
- PXI DAQ system
- …..

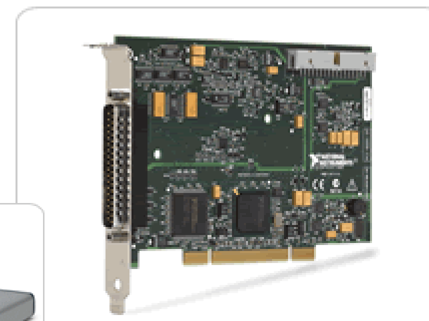


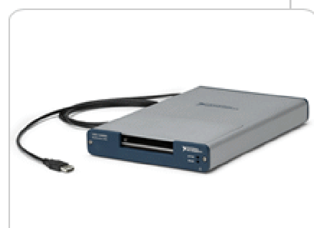
From Simple to advanced PC-based DAQ systems ….

- DAQ using the PC sound card
  - AC, low frequencies (10 – 20 kHz)

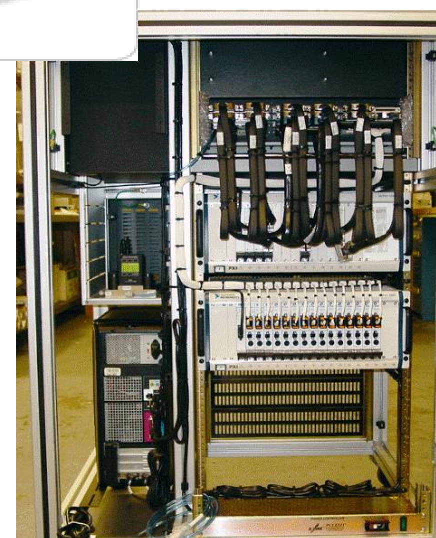


- PC with plug-in PCI DAQ card(s)
- PC with a USB DAQ device





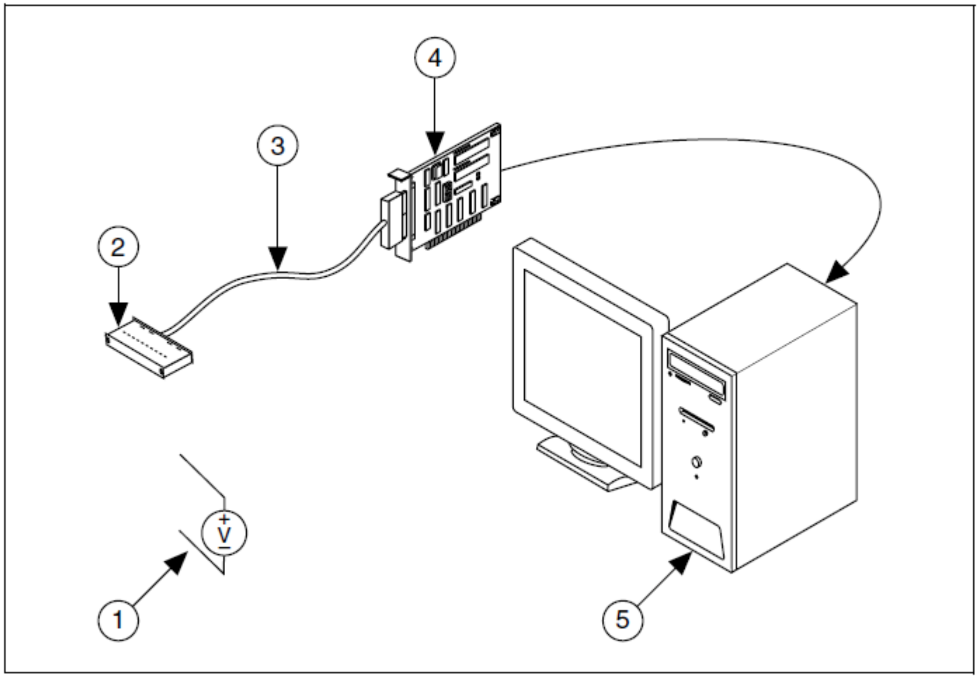
- DAQ system with hundreds of measurement channels
  - multiple connected PXI systems



Overview of a PC-based Data acquisition (DAQ) system

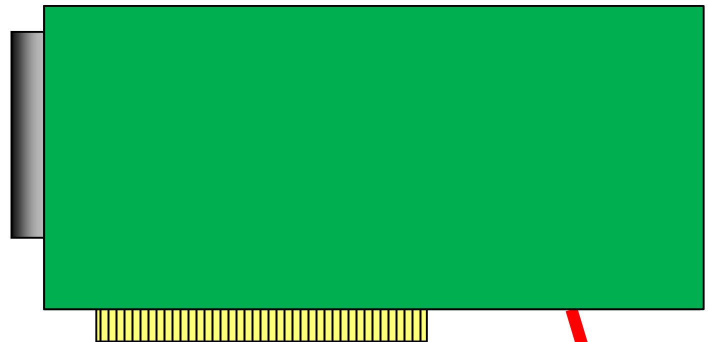
A DAQ system consists of:

- Sensors (transducers)
- Signal Conditioning
- Cables
- DAQ hardware
- Drivers
- Software

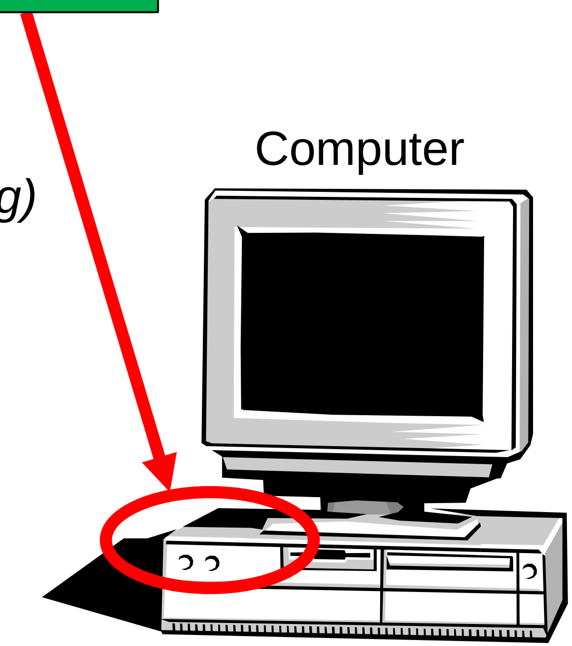


1	Sensors and Transducers	4	DAQ Hardware
2	Signal Conditioning	5	Personal Computer and DAQ Software
3	Cable Assembly		

Multifunction DAQ-units



- These DAQ-units have:
  - Analog Input
  - Analog Output
  - Digital I/O
  - Counters
    - Frequency measurements (*digital edge counting)*
    - Angular measurements from <u>angular encoders</u> (*quadrature encoders)*

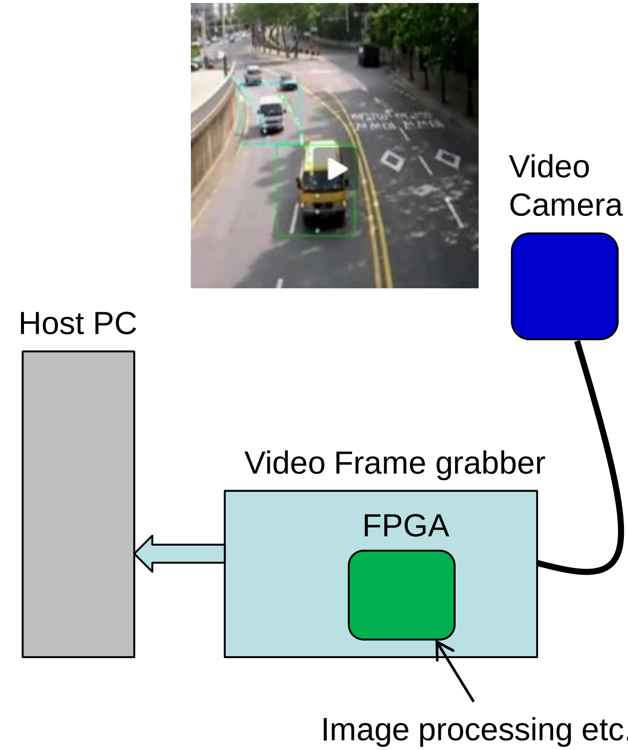


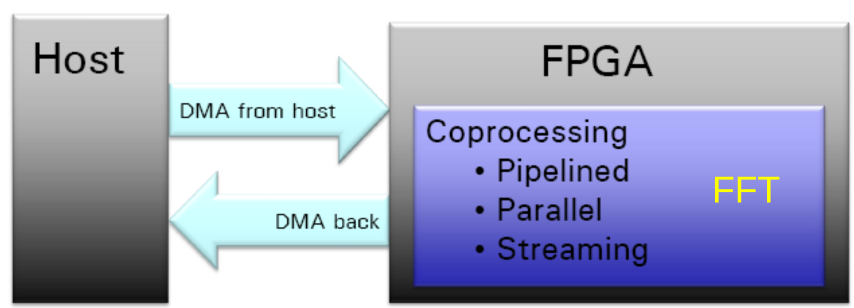
- Connects to the bus of your computer
  - PCI, PXI, PCIe, PXIe, or USB

DAQ-cards/units (usually) do <u>not</u> include an anti-aliasing low-pass filter

FPGAs in DAQ and control systems

- High-speed control
- Hardware timing/synchronization
- Hardware programmable DAQ-cards
- Onboard processing and data reduction
  - e.g. video processing
- Co-processing
  - offload the CPU



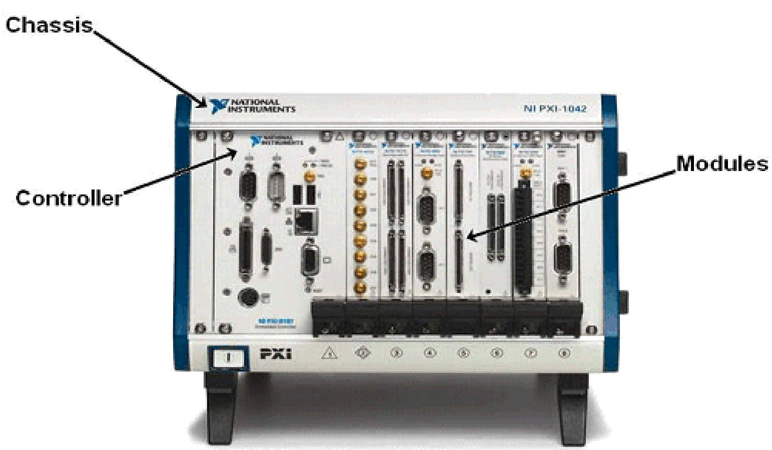


PXI

- PXI = PCI eXtensions for Instrumentation.
- PXI is a high-performance PC-based platform for measurement and automation systems.
- PXI was developed in 1997 and launched in 1998.
- Today, PXI is governed by the PXI Systems Alliance (PXISA), a group of more than 70 companies chartered to promote the PXI standard, ensure interoperability, and maintain the PXI specification.

PXI

- PXI systems are composed of three basic components:
  - Chassis
  - Controller
  - Peripheral modules



PXI chassis

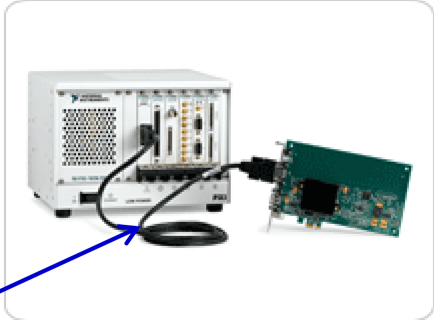
- The PXI chassis contains the backplane for the plug-in DAQ cards
- The chassis provides power, cooling, and communication buses for the PXI controller and modules.
- Chassis are available both with PCI and PCI Express
- 4 – 18 slots chassis are common

PXI controllers

- PXI Embedded Controller
  - Can run Windows or/and real-time OS

- Laptop Control of PXI
  - Using e.g. ExpressCard serial bus

- Desktop PC Control of PXI



> 800 MB/s possible (MXI bus)

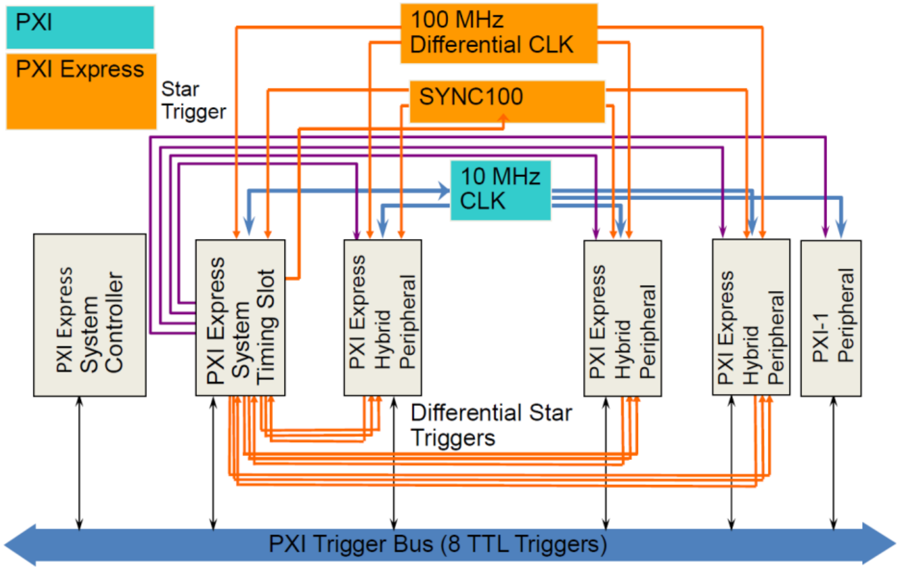
PXI-based DAQ systems

- PXI-based data acquisition systems include a more rugged packaging suitable for industrial applications.

- PXI systems offer a modular architecture
  - Possible to expand the DAQ system far beyond the capacity of a desktop computer.

PXI triggering and timing

- One of the key advantages of a PXI system is the integrated timing and synchronization.
- The PXI chassis includes reference clocks, triggering buses and slot-to-slot local bus.
  - Any module in the system can set a trigger that can be seen from any other module.
  - The local bus provides a means to establish dedicated communication between adjacent modules.



VISA

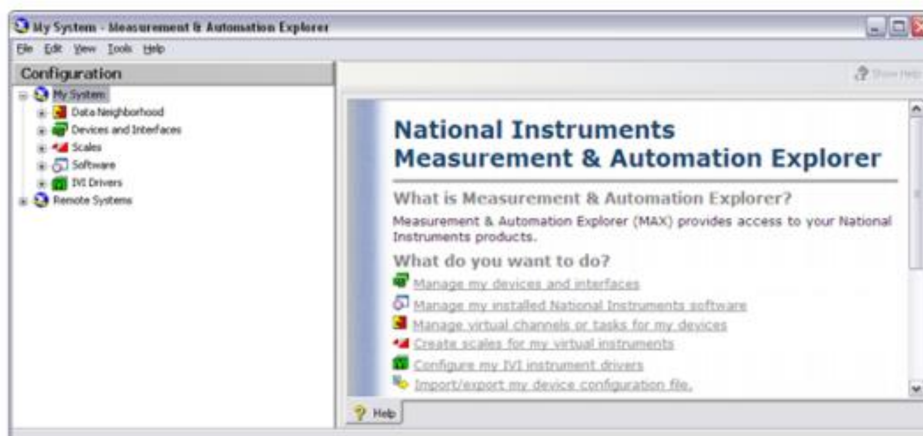
- VISA = Virtual Instrument Software Architecture.
- NI-VISA is the NI implementation of the VISA standard.
- LabVIEW instrument drivers are based on the VISA standard, which makes them bus- and platform-independent.
- Supports communication with instruments via:
  - GPIB
  - Serial
  - Ethernet
  - USB
  - PXI

NI-DAQmx

- NI-DAQmx (multithreaded) driver software provides ease of use, flexibility, and performance in multiple programming environments
- Driver level software
  - DLL that makes direct calls to your DAQ device
- Supports the following software:
  - NI LabVIEW
  - NI LabWindows CVI
  - C/C++
  - C#
  - Visual Basic .NET.

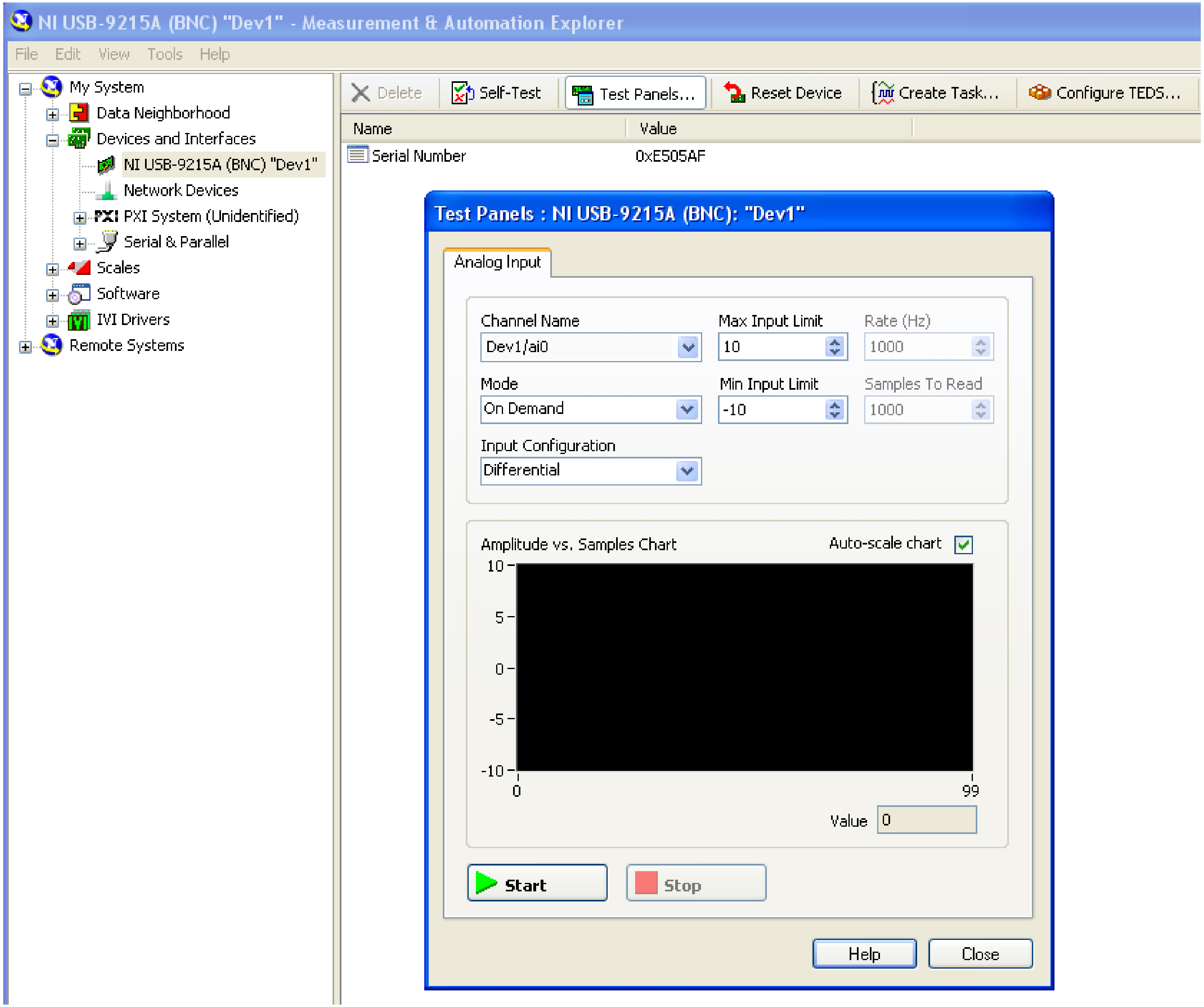
NI Measurement & Automation Explorer (MAX)

- All NI-DAQmx devices include MAX, a configuration and test utility
- You can use MAX to
  - Configure and test NI-DAQmx hardware with interactive test panels
  - Perform self-test sequences
  - Create simulated devices
  - Reference wiring diagrams and documentation
  - Save, import, and export configuration files
  - Create NI-DAQmx virtual channels that can be referenced in any programming language

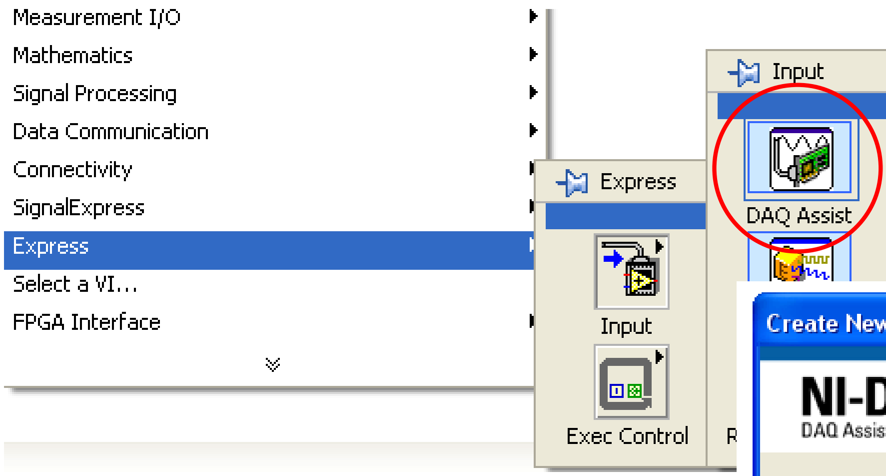


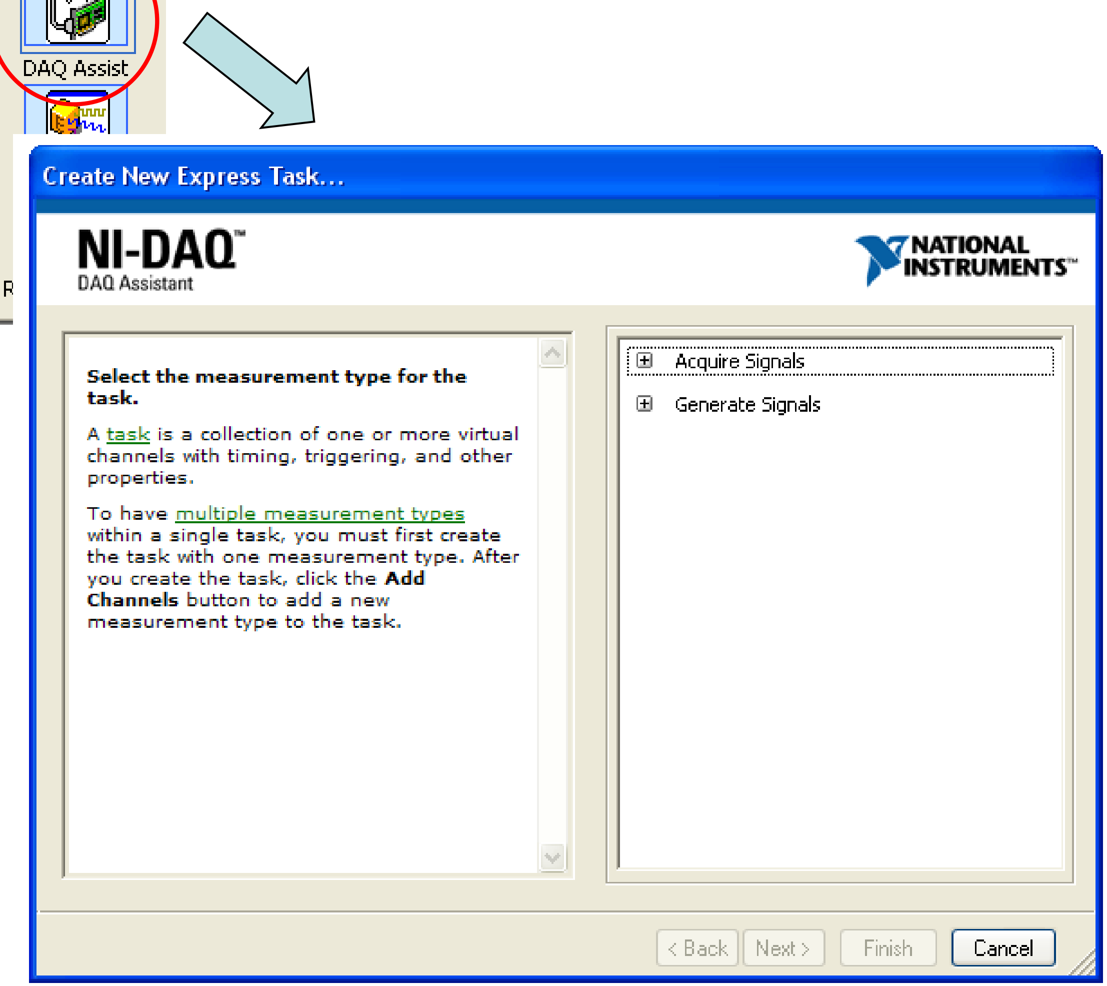
UiO : University of Oslo

MAX Example

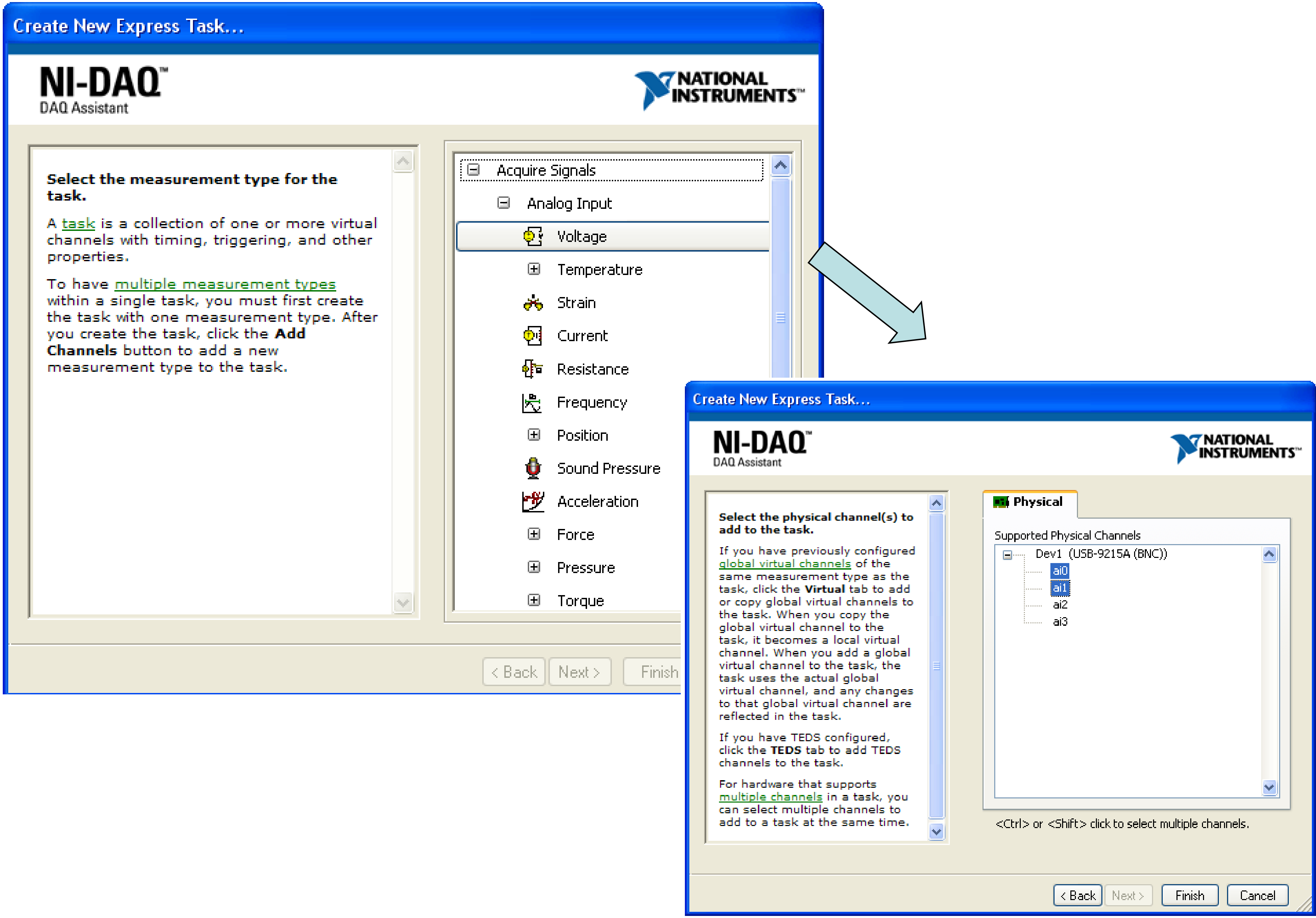


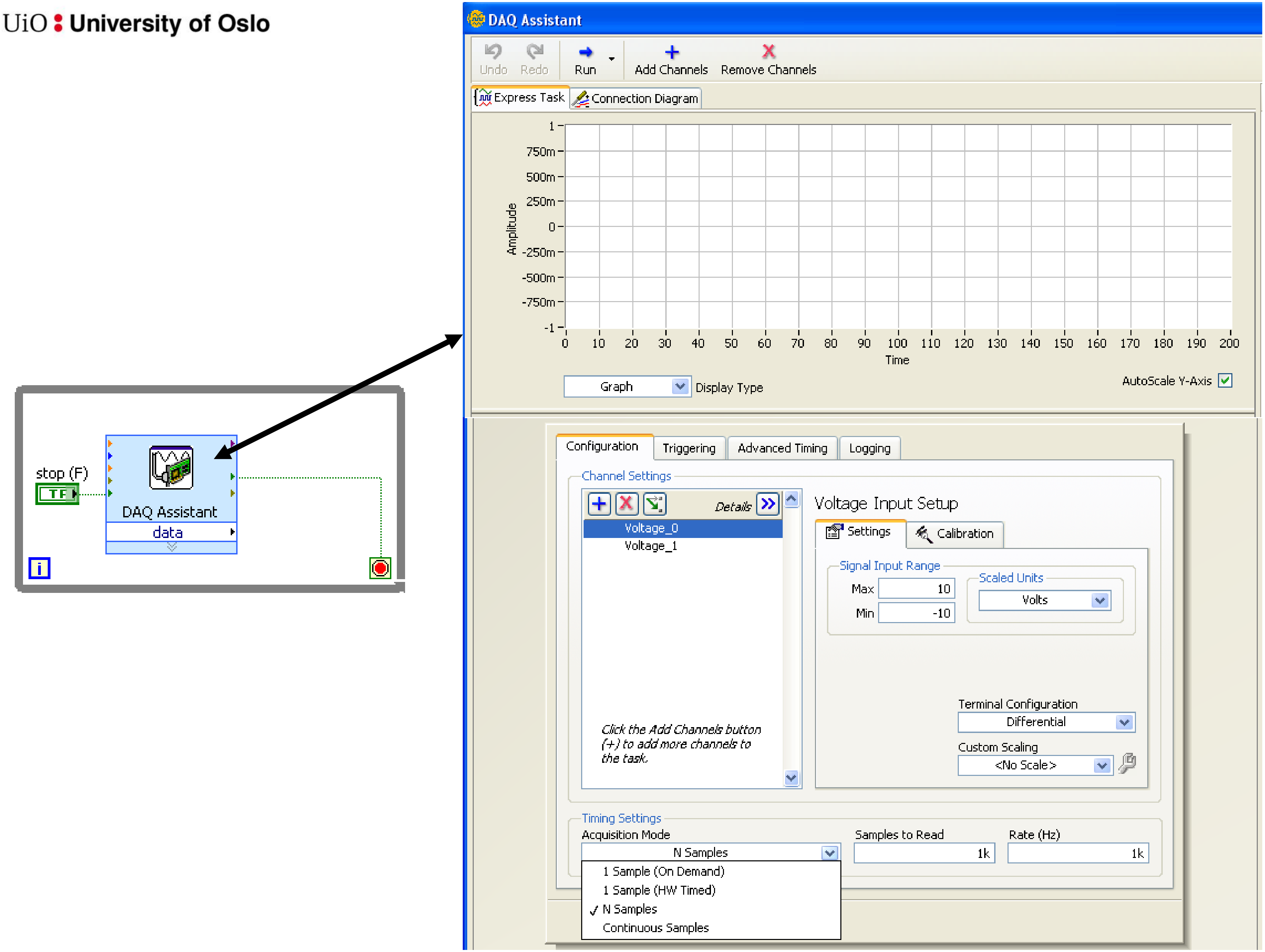
LabVIEW Express: DAQ assistant





Using the the <u>DAQ assistant</u> is the easy way to configure and read from a DAQ card!


Create New Express Task...
NI-DAQ
DAQ Assistant
NATIONAL INSTRUMENTS
Select the measurement type for the task.
A task is a collection of one or more virtual channels with timing, triggering, and other properties.
To have multiple measurement types within a single task, you must first create the task with one measurement type. After you create the task, click the Add Channels button to add a new measurement type to the task.
Acquire Signals
Analog Input
Voltage
Temperature
Strain
Current
Resistance
Frequency
Position
Sound Pressure
Acceleration
Force
Pressure
Torque
< Back
Next >
Finish
Create New Express Task...
NI-DAQ
DAQ Assistant
NATIONAL INSTRUMENTS
Select the physical channel(s) to add to the task.
If you have previously configured global virtual channels of the same measurement type as the task, click the Virtual tab to add or copy global virtual channels to the task. When you copy the global virtual channel to the task, it becomes a local virtual channel. When you add a global virtual channel to the task, the task uses the actual global virtual channel, and any changes to that global virtual channel are reflected in the task.
If you have TEDS configured, click the TEDS tab to add TEDS channels to the task.
For hardware that supports multiple channels in a task, you can select multiple channels to add to a task at the same time.
Physical
Supported Physical Channels
Dev1 (USB-9215A (BNC))
ai0
ai1
ai2
ai3
<Ctrl> or <Shift> click to select multiple channels.
< Back
Next >
Finish
Cancel


stop (F)
TF
DAQ Assistant
data
DAQ Assistant
Undo
Redo
Run
Add Channels
Remove Channels
Express Task
Connection Diagram
Amplitude
1
750m
500m
250m
0
-250m
-500m
-750m
-1
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 200
Time
Graph
Display Type
AutoScale Y-Axis
Configuration
Triggering
Advanced Timing
Logging
Channel Settings
Details
Voltage_0
Voltage_1
Click the Add Channels button (+) to add more channels to the task.
Voltage Input Setup
Settings
Calibration
Signal Input Range
Max 10
Min -10
Scaled Units
Volts
Terminal Configuration
Differential
Custom Scaling
<No Scale>
Timing Settings
Acquisition Mode
N Samples
1 Sample (On Demand)
1 Sample (HW Timed)
N Samples
Continuous Samples
Samples to Read
1k
Rate (Hz)
1k

LabVIEW - Sequential DAQ design

- Configure
- Acquire data
- Analyze data
- Visualize data
- Save data

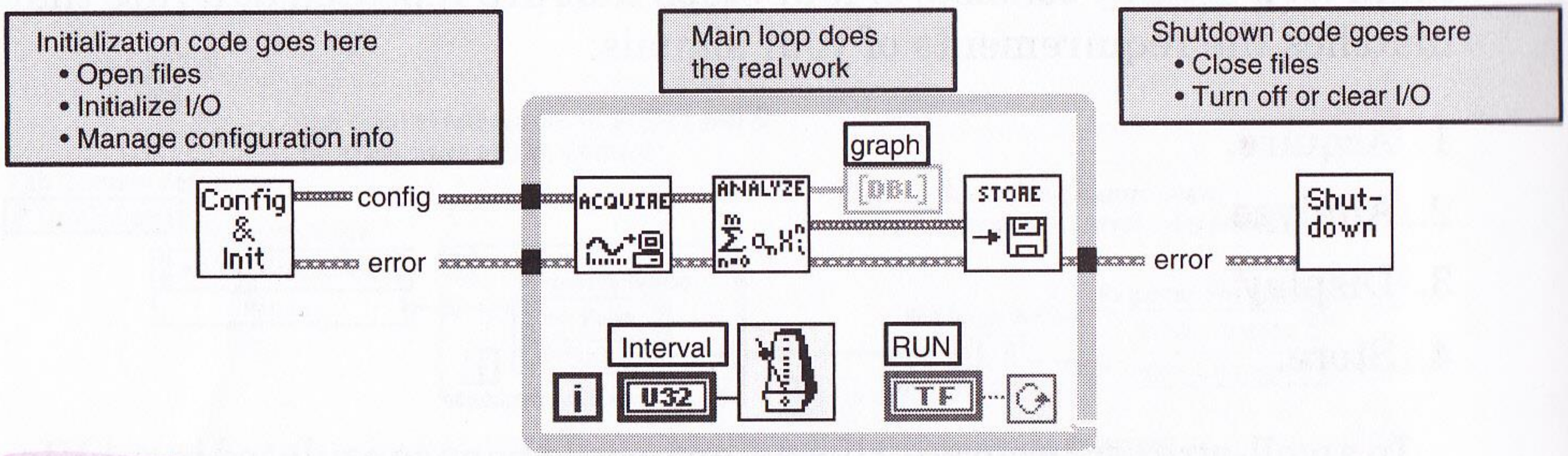
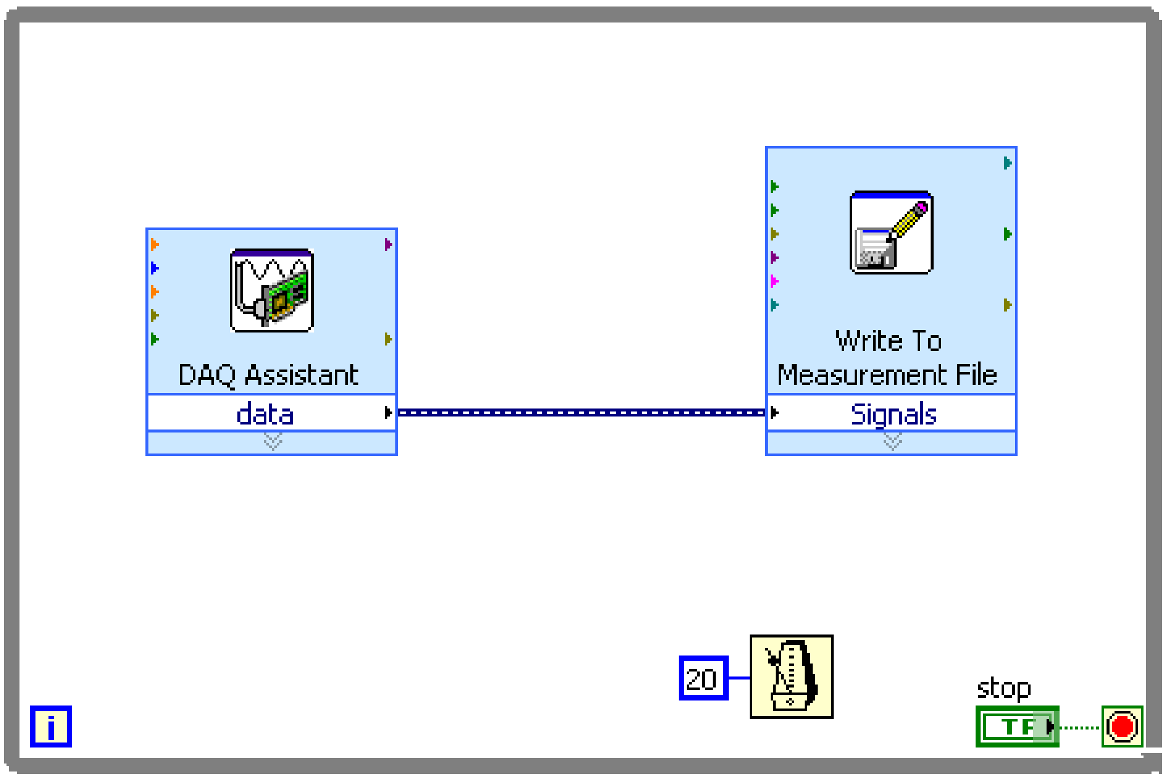


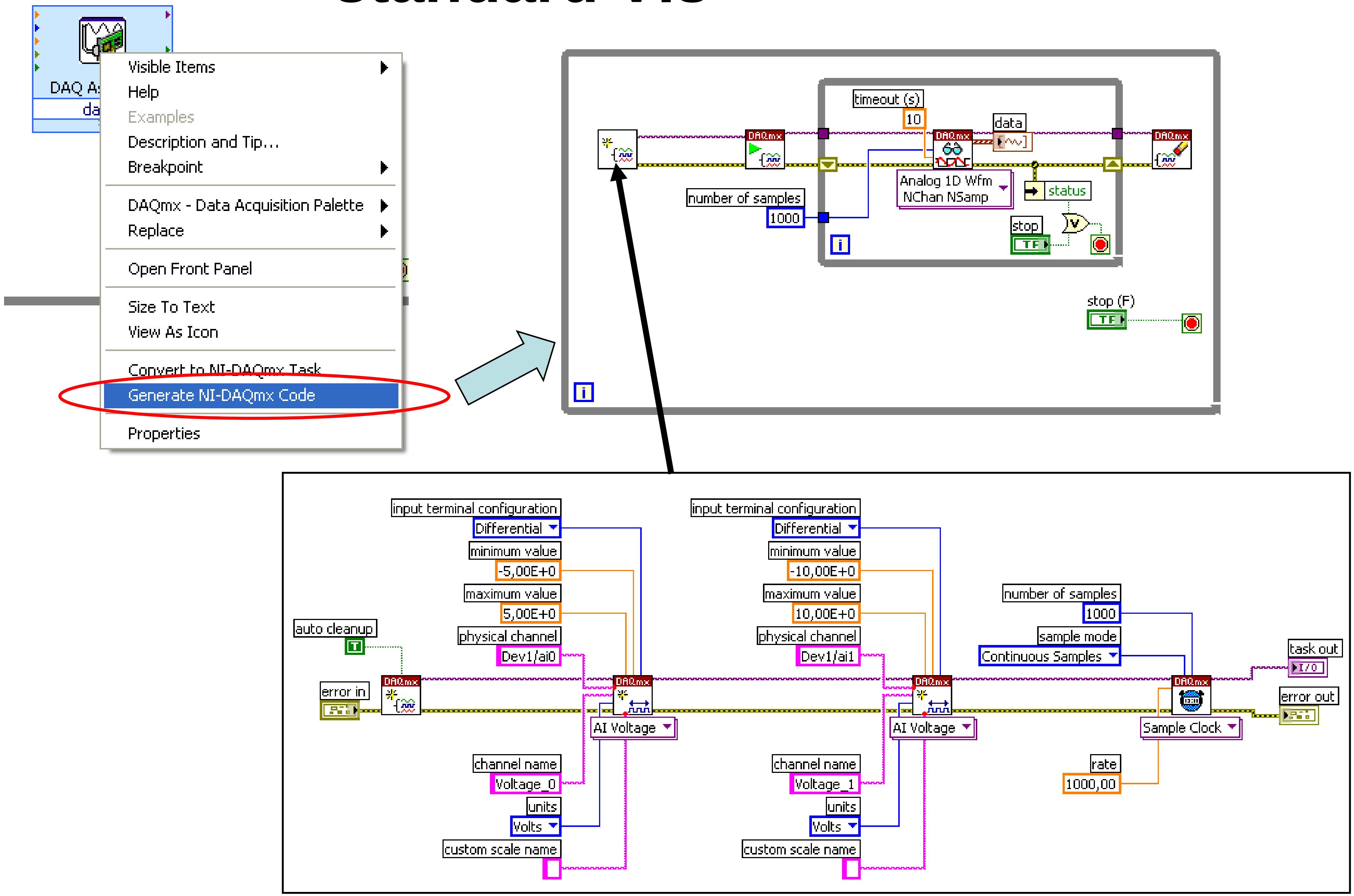
Figure 3.25 Dataflow instead of Sequence structure enhances readability. It has the same functionality as Figure 3.24. Note that the connections between tasks (subVIs) are not optional: they force the order of execution.

LabVIEW: Low-speed DAQ

- Sequential architecture
- ***DAQ assistant Express VI*** used in the block diagram
- Data written to file using the ***Write to Measurement File* Express VI**

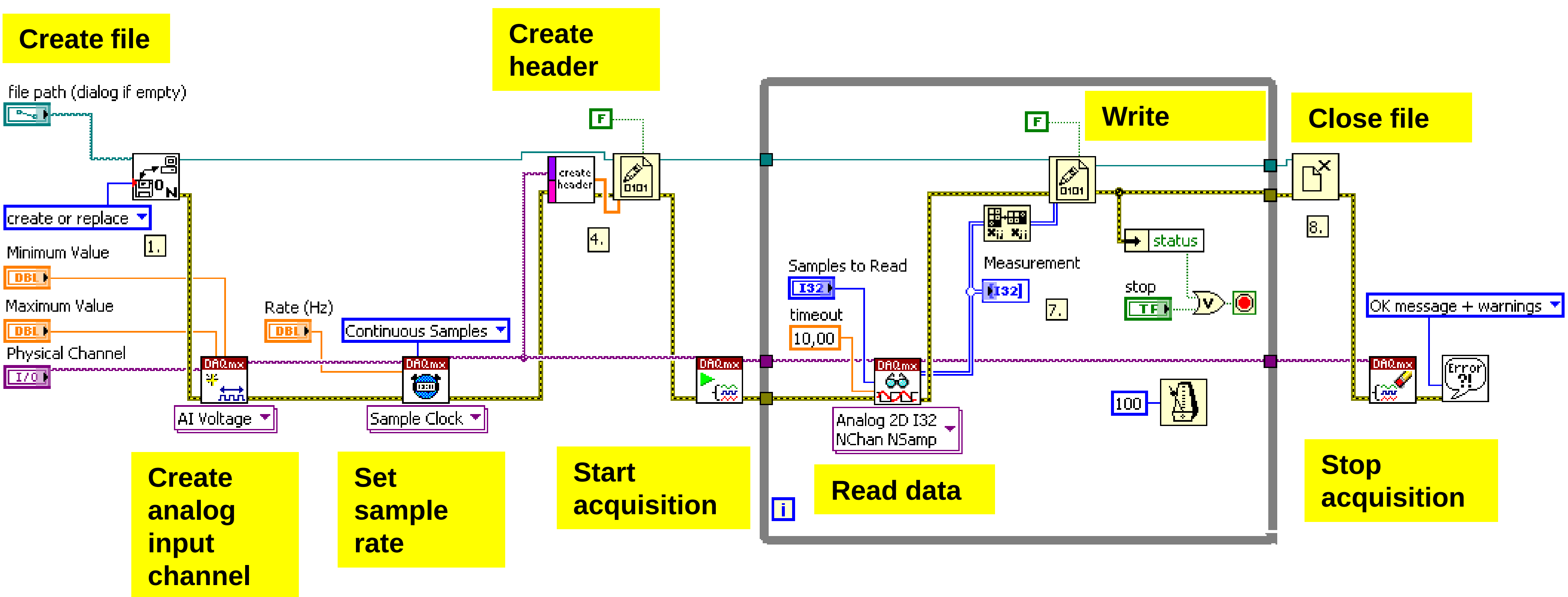


DAQ Assistant Express VI to standard VIs



LabVIEW: Medium-speed DAQ

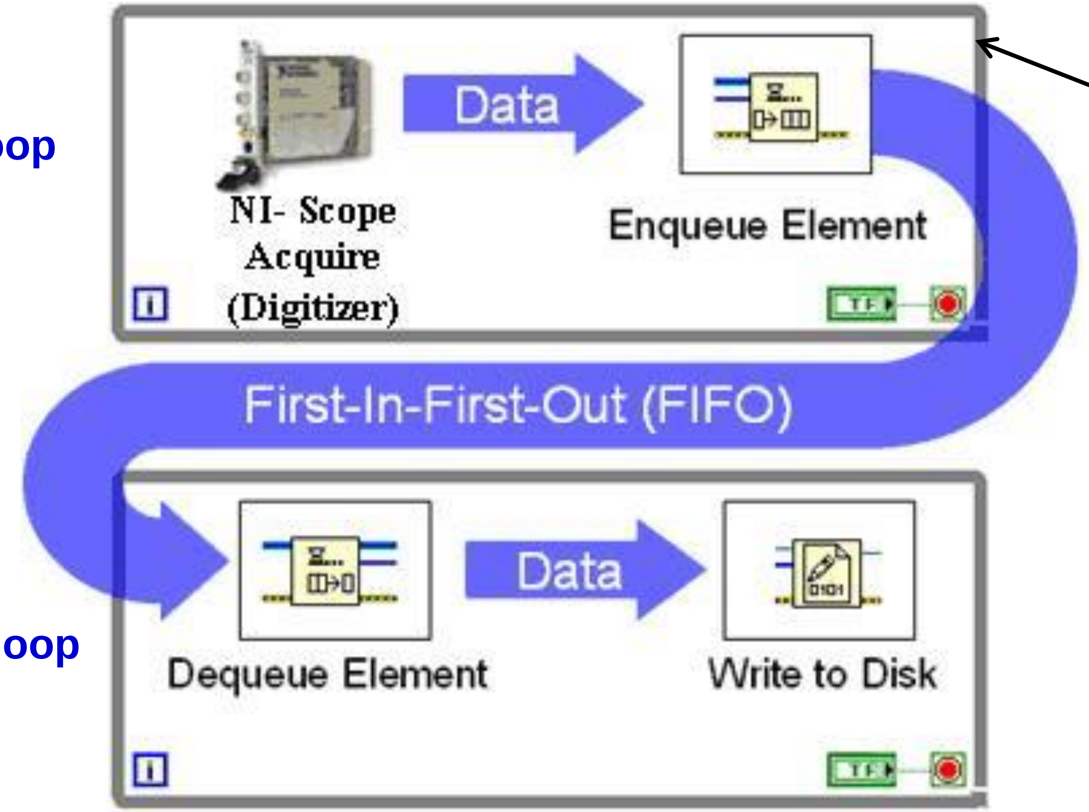
- Example: Cont Acq&Graph Voltage -To File (Binary).vi
- Sequential architecture
- **Standard VIs used, and data written to a binary file**



High-speed DAQ

- Based on the [producer-consumer](#) architecture
  - Parallel programming architecture

Producer loop



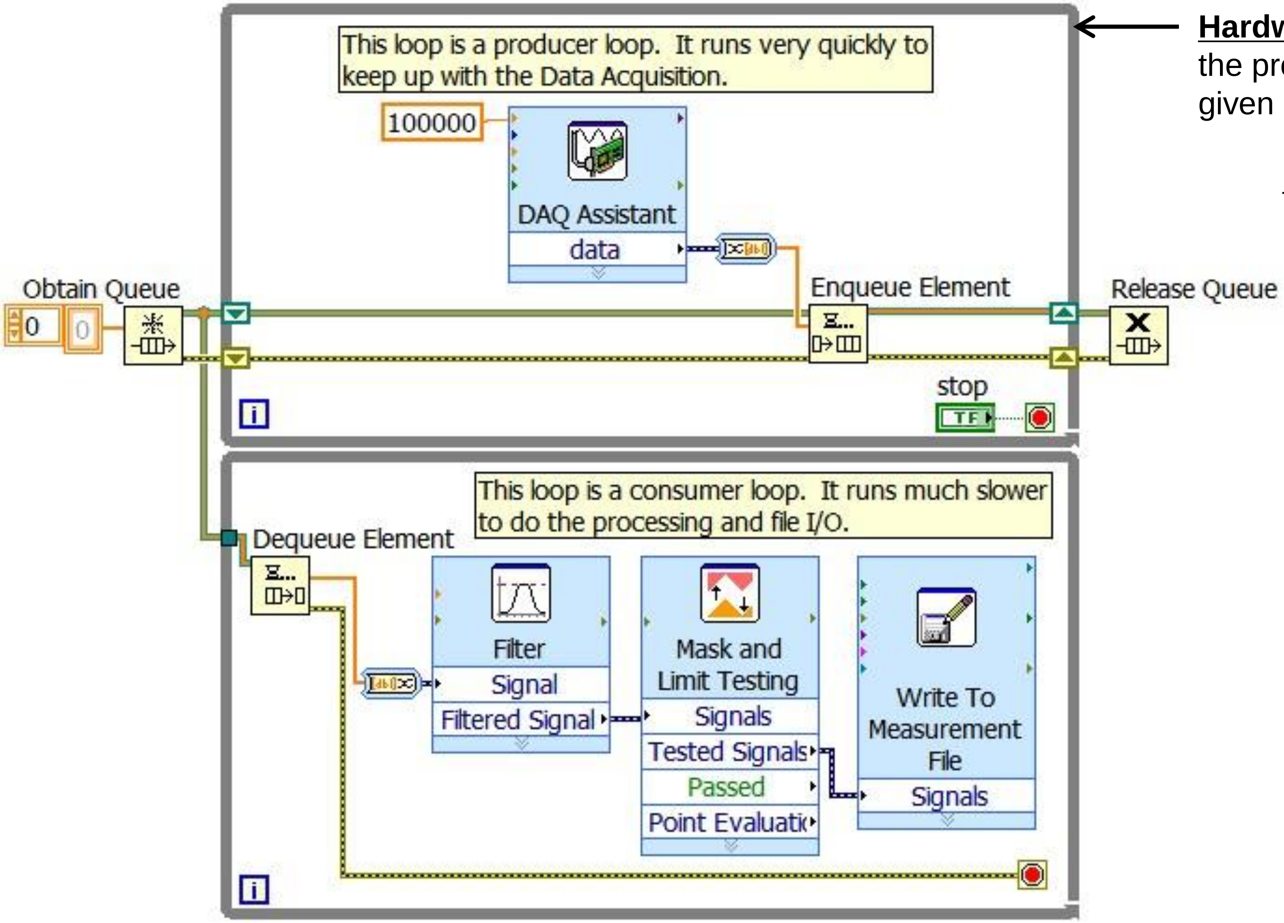
Consumer loop

Hardware timing, since no Wait function is used in the producer loop. The producer loop rate is given by the DAQ-card setup

$$f_{producer} = \frac{sample\ rate\ (Hz)}{DAQ\ card\ buffer\ size}$$

Producer – consumer DAQ Example

When we have multiple tasks that run at different speeds and cannot afford to be slowed down.



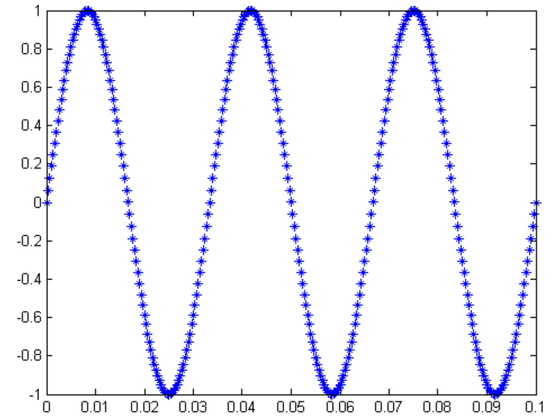
Hardware timing; no Wait function is used in the producer loop. The producer loop rate is given by the DAQ-card setup:

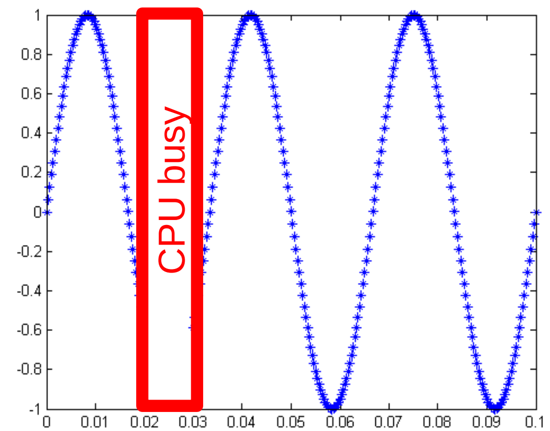
$$f_{producer} = \frac{sample\ rate\ (Hz)}{DAQ\ card\ buffer\ size}$$

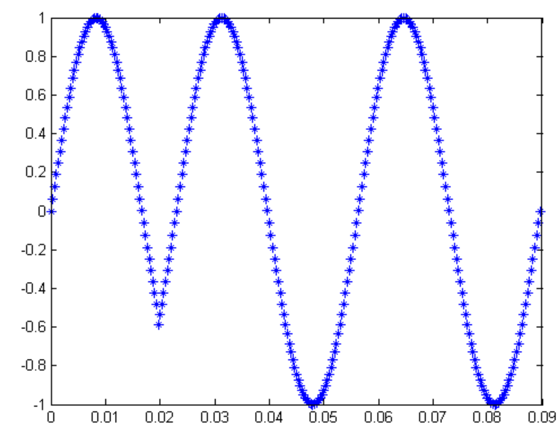
CPU busy example

- 30 Hz sinus signal sampled at 3 kHz (top figure)
- Assume a non real-time system used, without DMA and FIFO.
  - If the CPU get busy with something else between 0.02 seconds and 0.03 seconds, this section of the sine wave does not get sampled (middle figure)
- The computer will then interpret the sine wave as shown in the bottom figure, unaware that the <u>samples are not evenly spaced in time</u>
  - This will give a wrong result if we do a frequency analysis of the signal.

Therefore, DMA and FIFO buffers are used to compensate for non real-time properties of the operating system

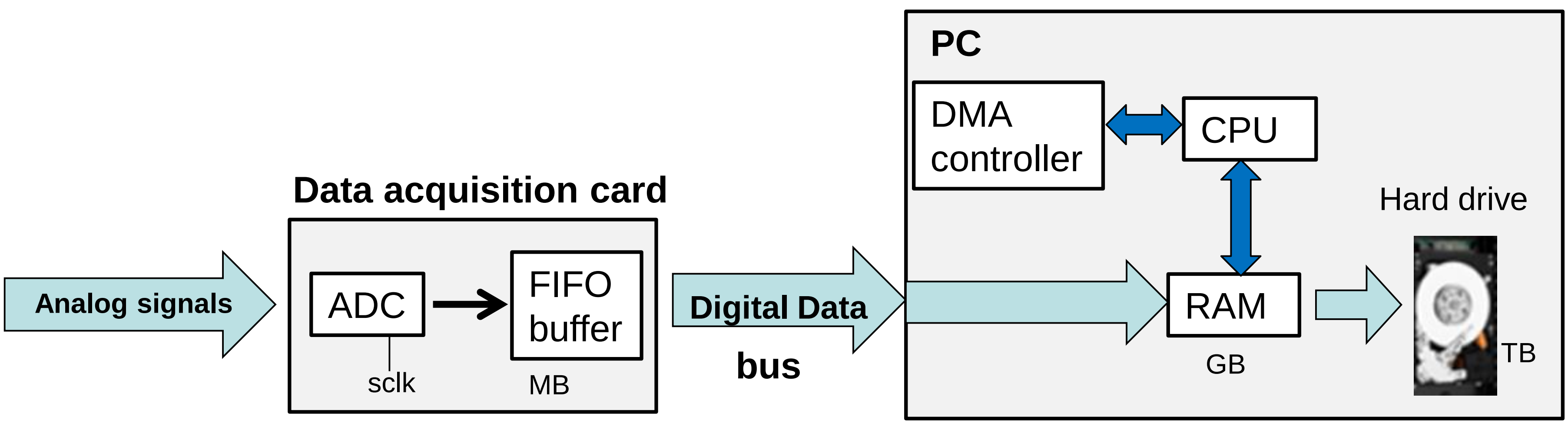






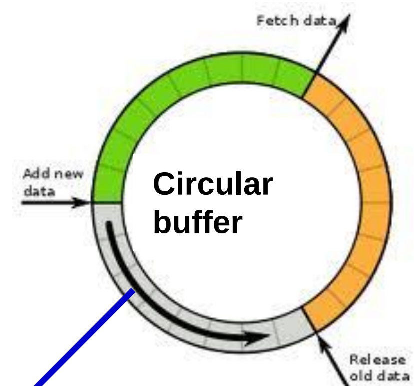
Transferring Data from DAQ-card to hard drive

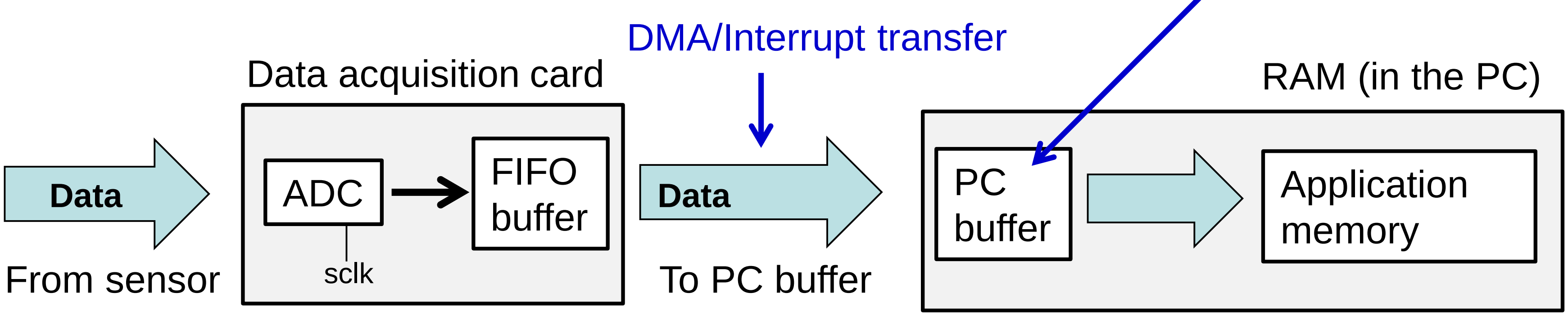
- Acquired data are stored in the hardware's first-in first-out (FIFO) buffer.
- Data is transferred from the DAQ-card FIFO buffer (of fixed size) to PC RAM using interrupts or DMA, across e.g. the PCI/PCI Express bus and the computer I/O bus.
- The samples are then transferred from RAM to hard drive via the computer I/O bus.



Continuous data acquisition

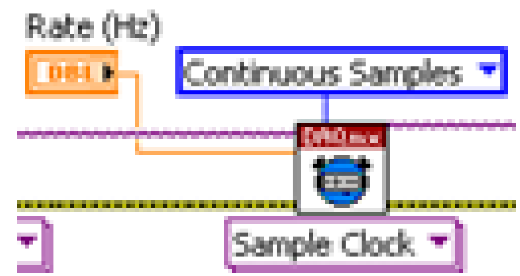
- To implement a continuous data acquisition on a non real-time system a **PC buffer** is needed in addition to the FIFO buffer on the DAQ card
- The PC buffer is a **circular buffer** in the computer RAM
- When we perform a DAQ-read in our application software we read the values out of the circular buffer and into a “variable” in our application program

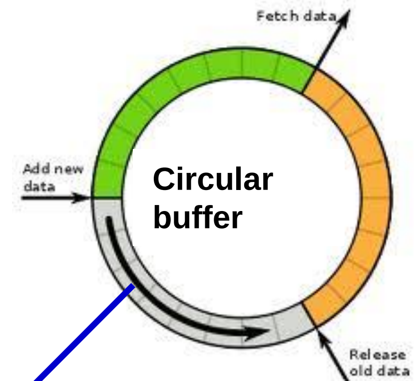


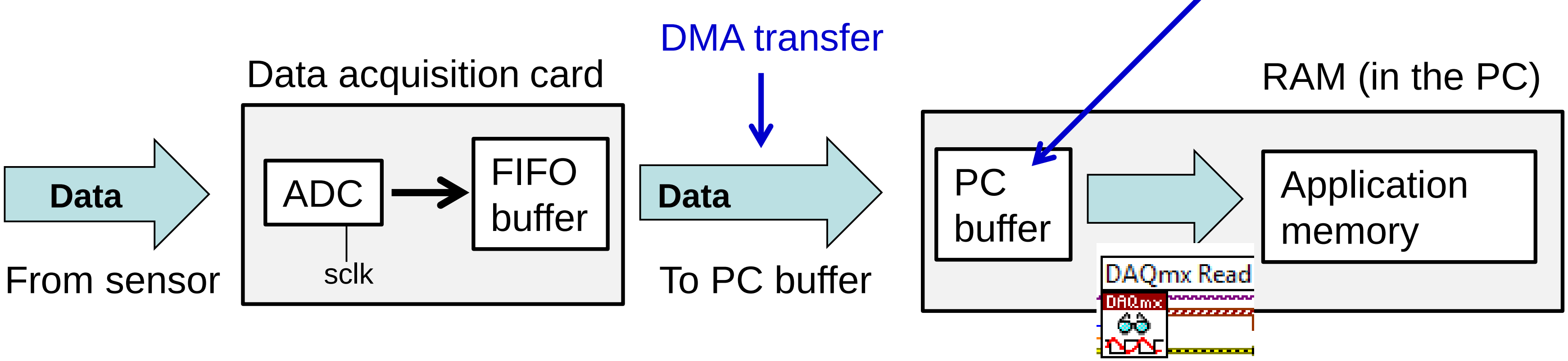


LabVIEW DAQ - hardware setup

- When the **sample clock** (DAQmx Timing.vi) is configured, DAQmx configures the board for hardwared-timed I/O
  - *DAQ card sample clock or external sample clock*
- By enabling **continuous sampling** DAQmx automatically sets up a circular buffer in RAM
- **DMA** is the default method of data transfer for DAQ devices that support DMA







DAQ data overwrite and overflow

- An **overwrite error** indicates that information is lost and occurs when the software program does not read data from the PC buffer quickly enough. Samples that are written to the circular PC buffer are overwritten before they are read into the application memory.
  - Solution: use **Producer-Consumer** architecture.
- An **overflow error** indicate that information has been lost earlier in the data acquisition process. Overflow errors indicate that the First In First Out (FIFO) memory buffer onboard the data acquisition card has reached its maximum capacity for storing acquired samples and can no longer accept new samples. An overflow error is symptomatic of a bus transfer rate that falls short of the requested data input rate.
  - Solution: use a Direct Memory Access (**DMA**) transfer mechanism.

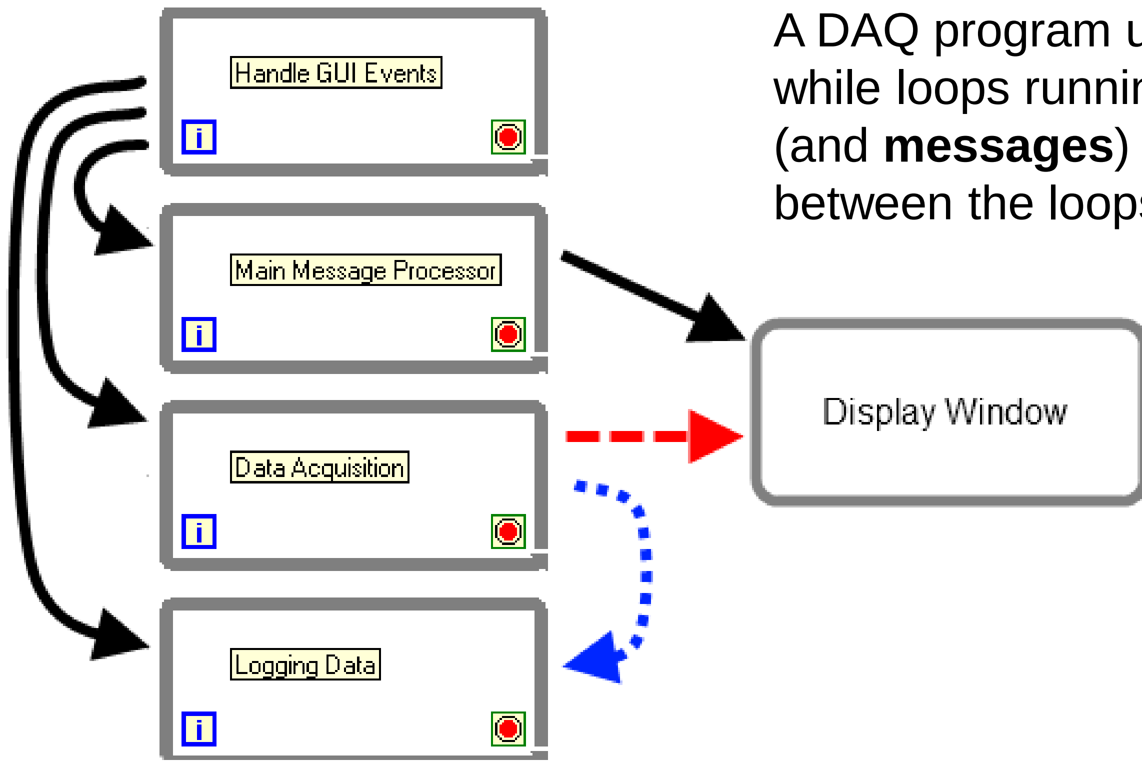
How Is Buffer Size Determined in LabVIEW DAQmx ?

- If the acquisition is **continuous** (sample mode in DAQmx Timing.vi set to Continuous Samples), NI-DAQmx allocates a PC buffer equal in size to the value of the **samples per channel** (gives the number of samples to acquire) property, unless that value is less than the value listed in the following table. If the value of the **samples per channel** property is less than the value in the table below, NI-DAQmx uses the value in the table.

Sample Rate	PC Buffer Size
No rate specified	10 kS
0–100 S/s	1 kS
101–10,000 S/s	10 kS
10,001–1,000,000 S/s	100 kS
>1,000,000 S/s	1 MS

- You can override the default buffer size using the function *DAQmx Configure Input Buffer.vi*

Advanced DAQ with multiple while loops



A DAQ program usually have several while loops running in parallel, and **data** (and **messages**) should be distributed between the loops using **queues**

Black : Messages transfer (using queue)
Blue : Data transfer using **Queue** (NB: queues have memory – no data is lost)
Red : Data transfer using **Notifier** (NB: notifiers do not have memory/FIFO)

Sample project in LabVIEW – queued message handler

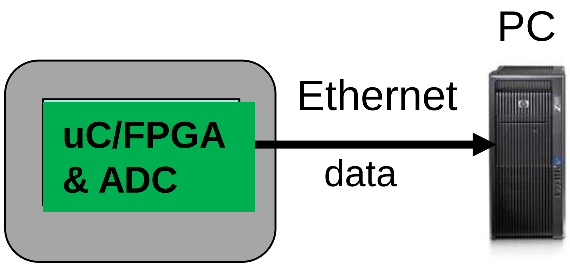
Remote control and data distribution

- **Remote Control**
  - Enabling another computer to connect to the experiment and control that experiment remotely.

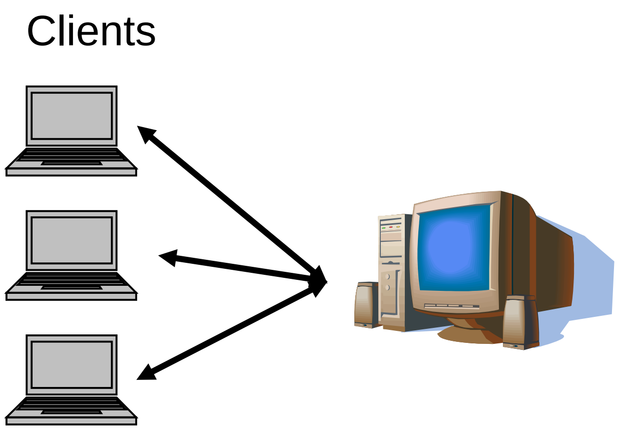
- **Distributed Execution**
  - A system architecture that shares the acquisition and analysis of the test among several computers.

Distributed DAQ examples

- **Remote DAQ**
  - Transfer data from a remote DAQ device to a single PC for display and storage



- **Networked (distributed) DAQ**
  - Distribute measurement data to several clients connected to a network
  - Enable a central computer to acquire all of the data from several machines and then process or store that data



How to increase the signal-to-noise-ratio

- **Use an amplifier** (as close to the sensor as possible, to amplify the signal before the noise enter e.g. the transmission cable.
- **Use an ADC with more bits per sample** (The SNR of an ideal N-bit ADC is SNR(dB) = 6.02*N + 1.76 (for sinusoidal signals).
- **Use oversampling** (followed by digital Low pass filtering and down-sampling).
- **Filtering** (to remove noise and limit the signal bandwidth), in hardware or software
  - remember from basic electronics that <u>thermal noise</u> (Johnson noise) in a resistors is proportional to the square root of the signal bandwidth.
- **Averaging** of n samples : $\sigma_{avg} = \sigma\sqrt{\frac{1}{n}}$

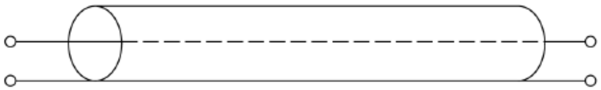
$$S/N_n = S/N\sqrt{n}$$

How to increase the signal-to-noise-ratio II

- Position noise sources away from data acquisition device, cable, and sensor.
- Place data acquisition device as close to sensor as possible to prevent noise from entering the system.
- Use **twisted pairs** and **shielding**, or **coax cables**.
- Use **differential signals**
- **Avoid ground loops**.

- Topics not covered in lectures:
  - Lock-in amplifier

(a) Twisted-pair cable



Coax cable

Savitzky-Golay smoothing filters

- Savitzky-Golay filters are optimal in the sense that they **minimize the least-squares error in fitting a polynomial to frames of noisy data**.
- Savitzky-Golay smoothing filters are typically used to "smooth out" a noisy signal whose frequency span (without noise) is large. In this type of application, Savitzky-Golay smoothing filters perform much better than standard averaging filters, which tend to filter out a significant portion of the signal's high frequency content along with the noise.
- Although Savitzky-Golay filters are more effective at preserving the pertinent high frequency components of the signal, they are less successful than standard averaging filters at rejecting noise.
- This filter **preserve the amplitude of a time-varying signal much better than a sliding-average smoothing.**