Introduction

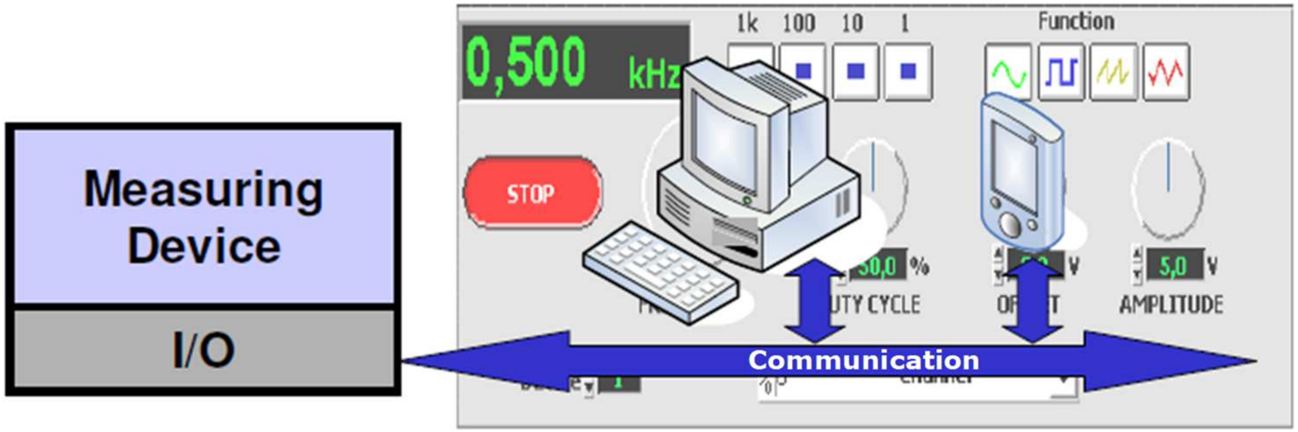
LabVIEW User Interface

Roland Pawliczek, PhD.

The idea of virtual instrumentation

What is the virtual instrument:

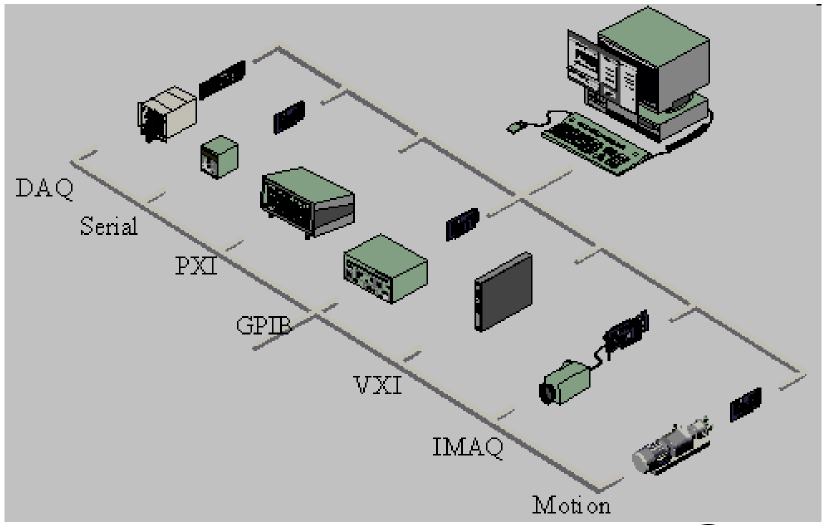
- *A virtual instrument consists of an industry-standard computer or workstation equipped with powerful application software, cost-effective hardware such as plug-in boards, and driver software, which together perform the functions of traditional instruments.*
- *Coupling of the measuring instruments (hardware) witch can be either the stand alone or PC based (e.g. DAQ board), with software for data acquisition and instruments control.*



Communication interfaces

Many types of interfaces can be used to communicate with PC. There are some most popular and widely used in instrumentation including more and more popular USB serial interface.

- *DAQ – Data AcQuizition boards*
- *PXI – PCI eXtensions for Instrumentation established in 1998, defines a compact modular PC-based platform for test, measurement, and control applications*
- *GPIB – General Purpose Interface Bus*
- *and others.*



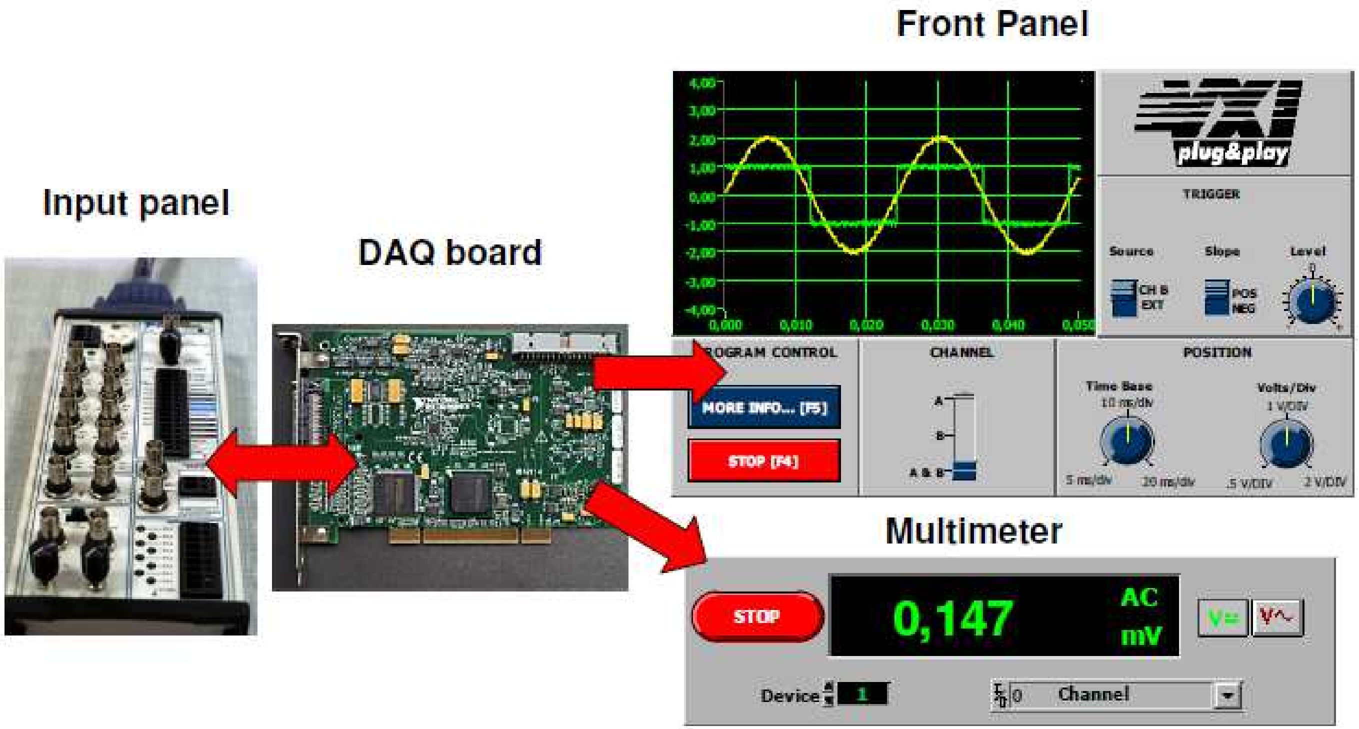
Communication interfaces

- *National Instruments offers many DAQ board. We should consider our requirements to chose the right DAQ board, e.g. with on-board lowpass filter, simultaneous sampling etc.*
- *NI websites can lead you step by step to chose your equipment according to some questions about your needs.*

Compact DAQ (USB connection to PC, but many signal conditioning modules for different sensor types)

The idea of virtual instrumentation

An example of DAQ board and software:

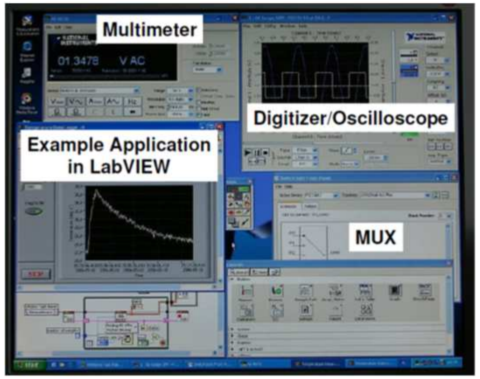


The idea of virtual instrumentation

PXI system: *PXI (PCI eXtensions for Instrumentation) is a rugged PC-based platform for measurement and automation systems. <u>These systems serve applications such as manufacturing test, military and aerospace, machine monitoring, automotive, and industrial test.</u>*

PXI systems are comprised of three basic components – chassis, system controller, and peripheral modules.

Software

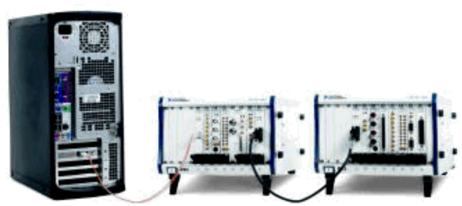


National Instruments PXI-8187 Pentium 4-M 2.5 GHz Embedded Controller. *Notice the familiar PC peripherals such as keyboard/mouse and monitor connections, as well as the hard drive, USB 2.0, Ethernet, serial, and other standard PC peripherals. This controller runs standard Windows 2000/XP OSs, or can be targeted with LabVIEW Real-Time.*

Laptop Control of PXI

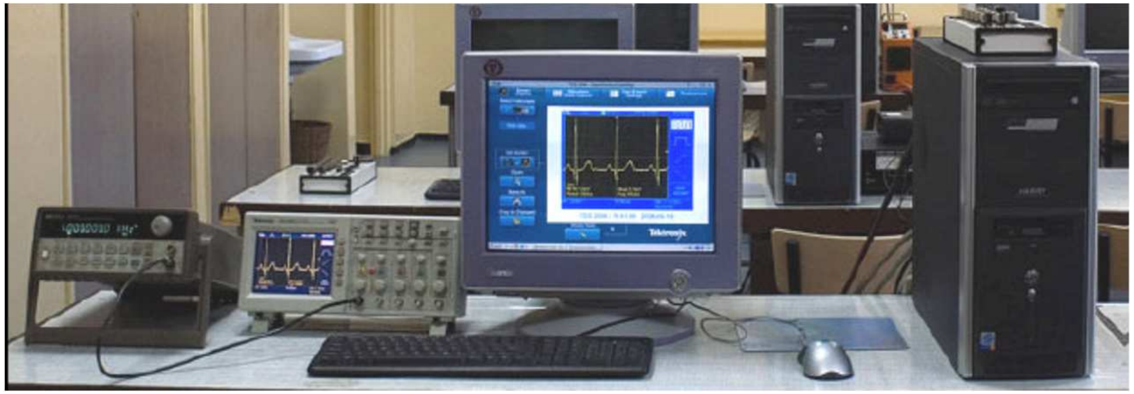
With ExpressCard MXI (Multisystem eXtension Interface) and PCMCIA CardBus interface kits, users can control PXI systems directly from laptop computers. During boot-up, the laptop computer will recognize all peripheral modules in the PXI system as PCI devices.

It is possible to build more complex multichassis system controlled by standard PC.



The idea of virtual instrumentation

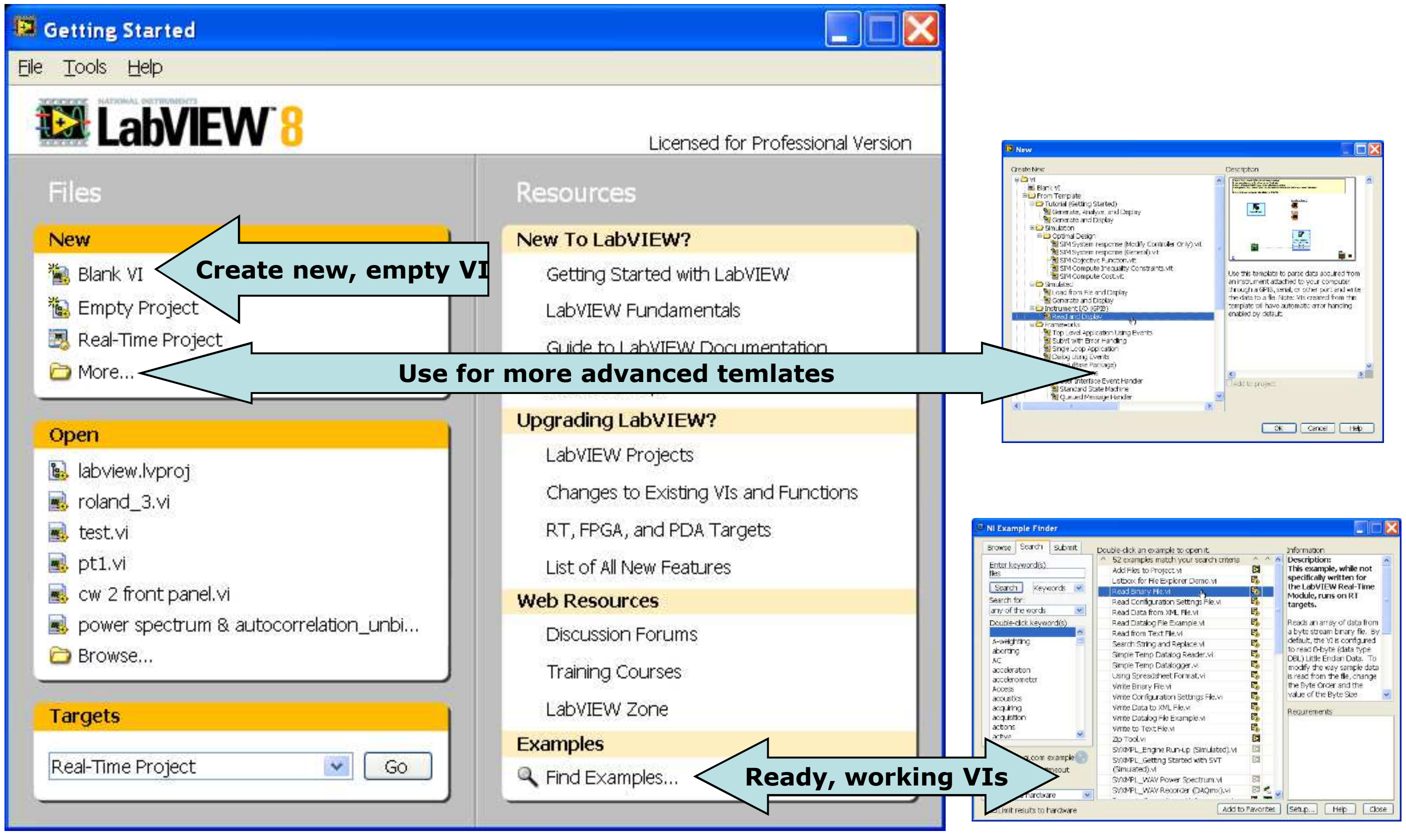
GPIB – General Purpose Interface Bus: *we can integrate different interfaces for using in traditional test and measurement applications.*



Laboratory Virtual Instrument Engineering Workbench

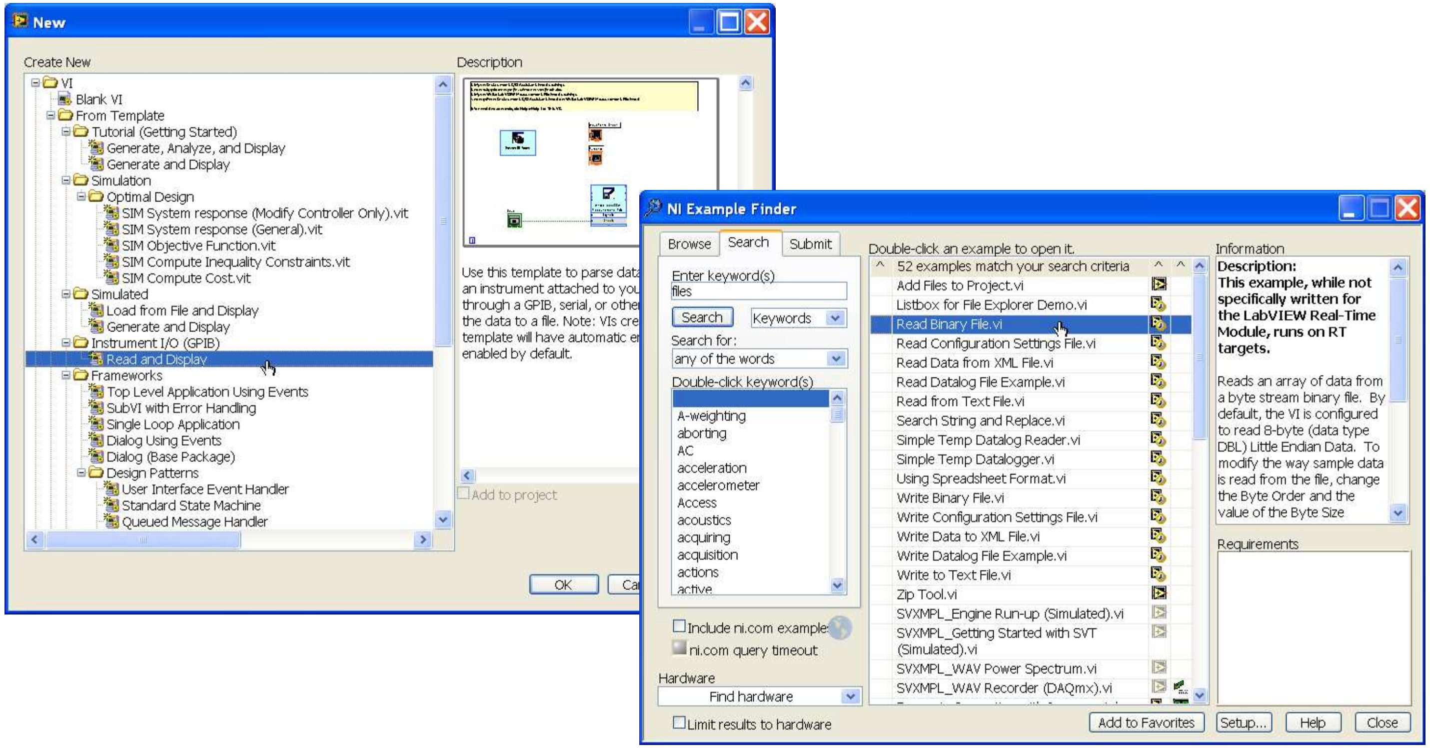
- ***LabVIEW programs*** *are called* ***virtual instruments****, or* ***VIs****, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. LabVIEW contains a comprehensive set of tools for acquiring analyzing, displaying, and storing data, as well as tools to help you troubleshoot your code.*
- *In LabVIEW, you build a user interface (****front panel****) with* ***controls*** *and* ***indicators****. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays.*
- *Use LabVIEW to communicate with hardware such as data acquisition, vision, and motion control devices, and GPIB, PXI, VXI, RS-232, and RS-484 devices. LabVIEW also has built-in features for connecting your application to the Web using the LabVIEW Web Server and software standards such as TCP/IP networking and ActiveX.*
- *Using LabVIEW, you can create test and measurement, data acquisitions, instrument control, measurement analysis, and report generation applications. You also can create stand-alone executables and shared libraries, like DLLs, because LabVIEW is a true 32-bit compiler.*

Starting LabVIEW



Starting LabVIEW

Option ***New/More...*** *- select advanced template for specified application.*



Option ***Find Examples...*** *- displays a few ready VIs for specified keyword*

LabVIEW user interface

Front Panel

(The front panel is the user interface of the VI)

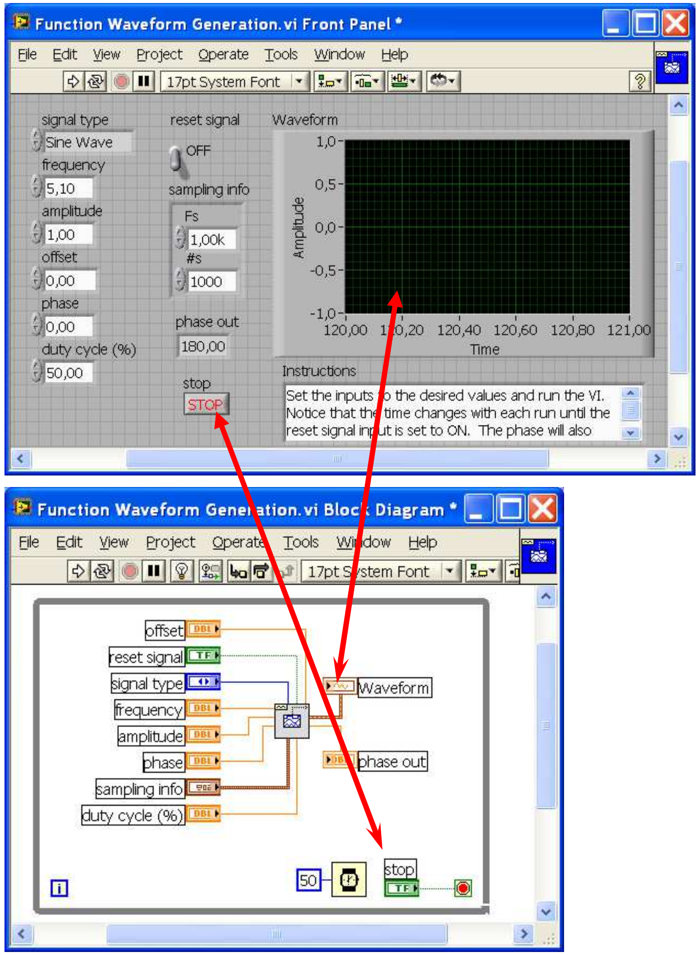
- *Controls (data input)*
- *Indicators (data display)*
- *Other elements (decorations, figures, text)*

Block Diagram

(Graphical code of the VI)

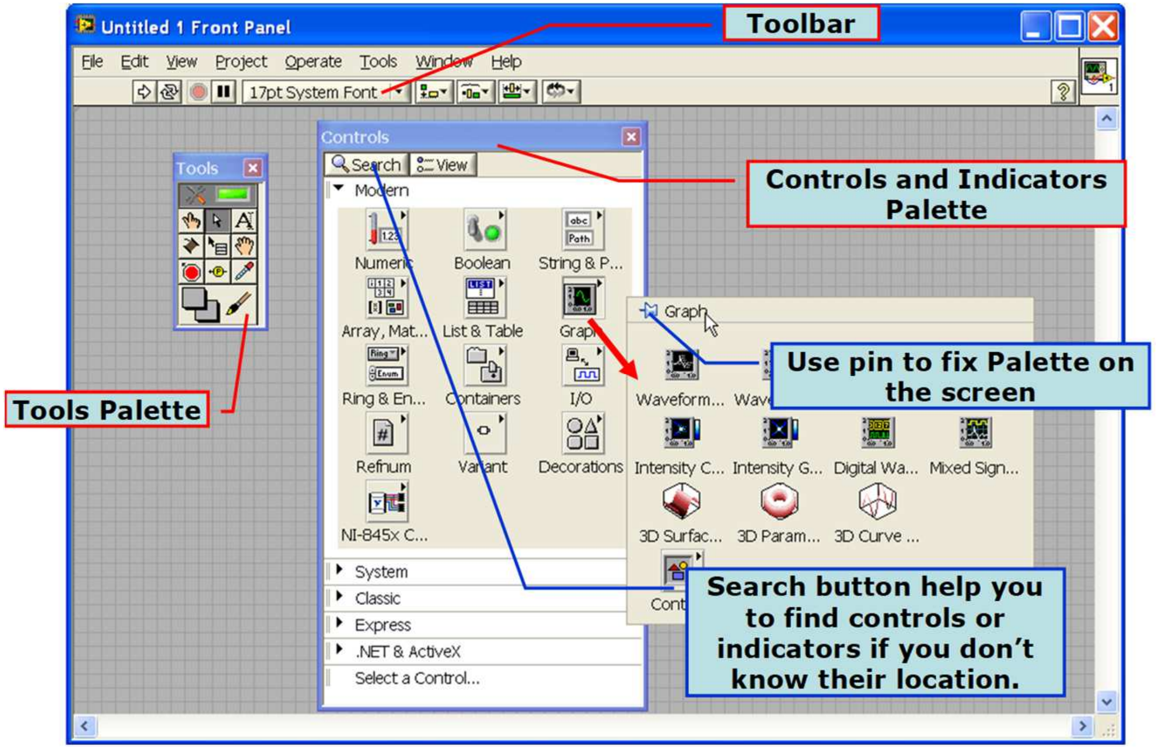
- *Icons connected with virtual wires*
- *Loops, structures, functions, subVI's*
- *Other elements (figures, text)*

Each control and indicator placed on the Front Panel is represented on the Block Diagram. We call it Terminal.



LabVIEW user interface: Front panel – Controls Palette

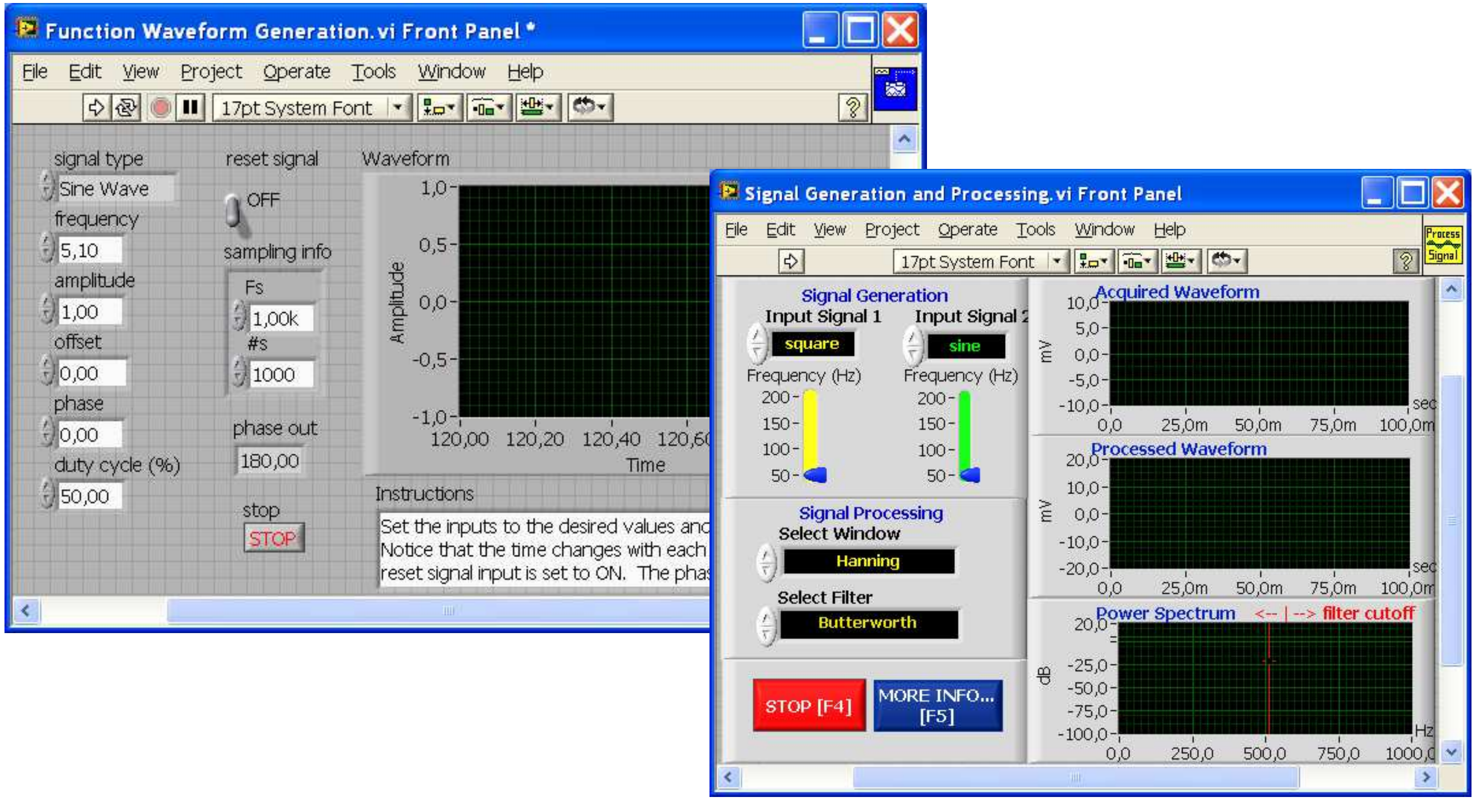
Options: ***View/Controls Palette*** *(or simply RMB on the Front Panel) and* ***View/Tools Palette*** *open the following windows on the Front Panel:*



Use ***drag-and-drop*** *to place selected controls or indicators on the front panel.*

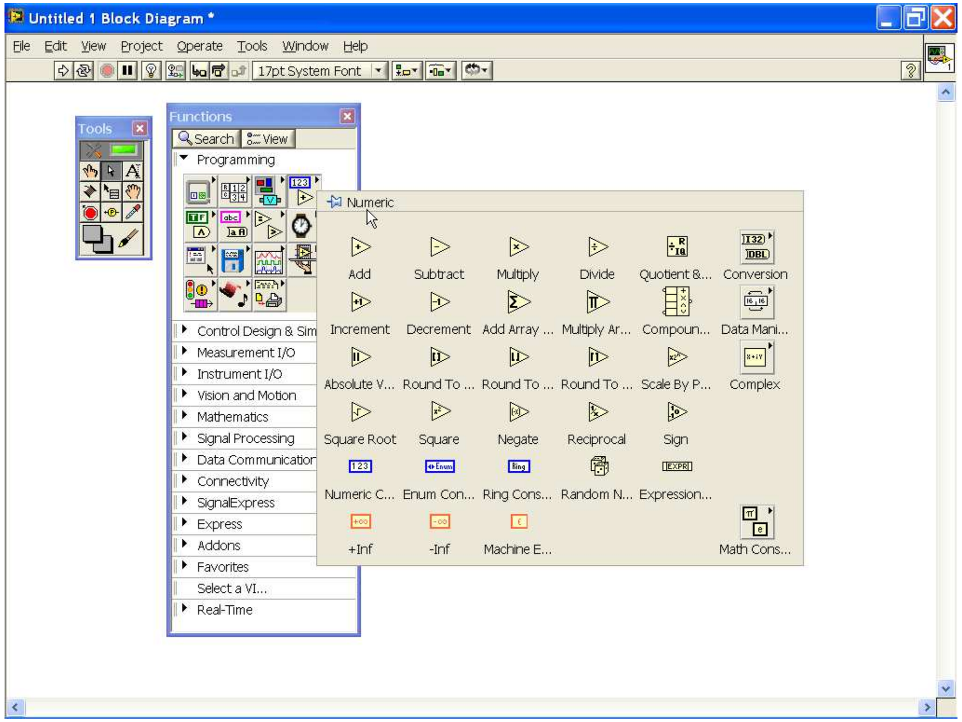
LabVIEW user interface: Front panel

The USER INTERFACE of the created application is designed with the use of the Controls Palette



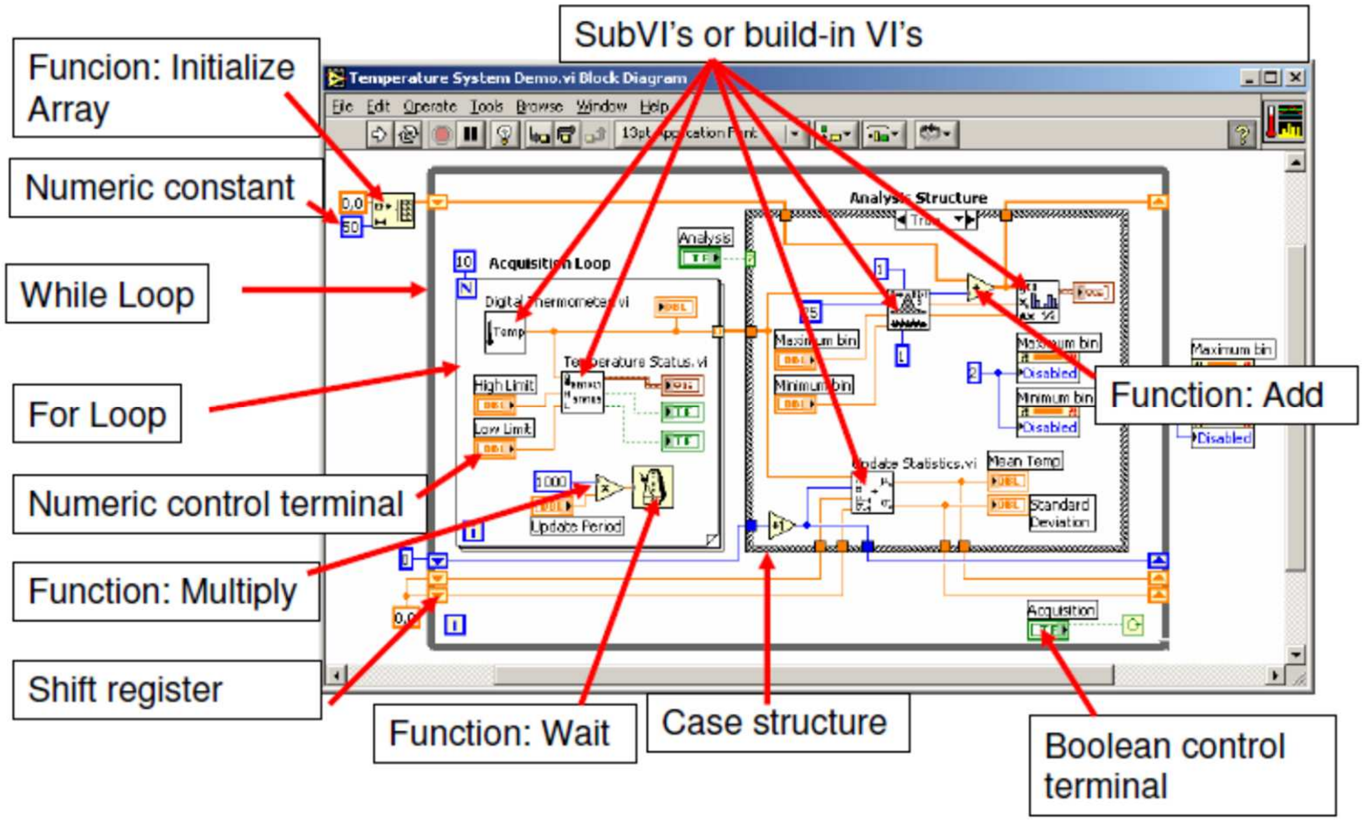
LabVIEW user interface: Block Diagram – Functions Palette

Options: ***View/Functions Palette*** *(or simply RMB on the Block Diagram) and* ***View/Tools Palette*** *open the following windows on the Block Diagram:*



Use ***drag-and-drop*** *to place selected functions on the diagram.*

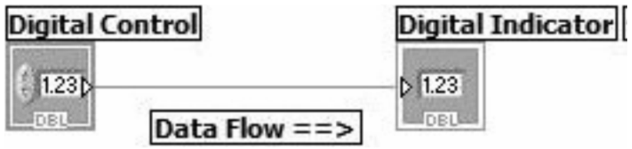
LabVIEW user interface: Block Diagram



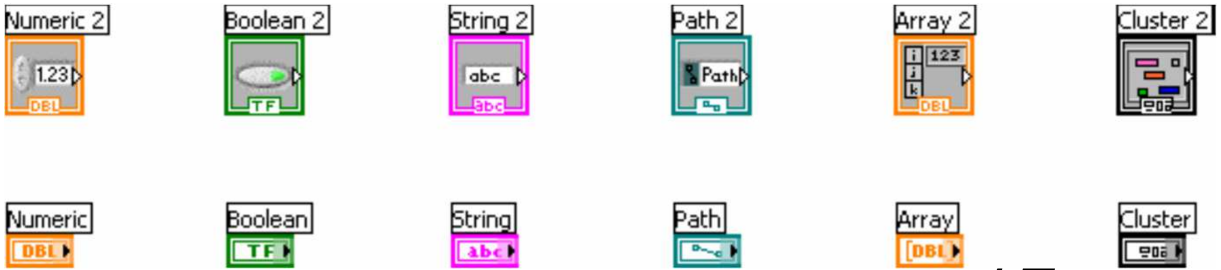
LabVIEW user interface: Block Diagram - components

You construct the block diagram by wiring together objects that perform specific functions. Typical components of a block diagram are: terminals, nodes, and wires.

The terminals represent the data type of the control or indicator. You can configure front panel controls or indicators to appear as icon or data type terminals on the block diagram. The data flow through the wire always from Control to Indicator.



Terminals can be viewed with the ***View As Icon*** *option selected (top row in fig. below) and with the option not selected (bottom row in fig. below). You can select the option from shortcut menu.*

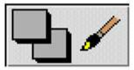


LabVIEW user interface: Tools Palette

- *The Tools palette is available on both the front panel and the block diagram. A tool is a special operating mode of the mouse cursor. The cursor corresponds to the icon of the tool selected in the Tools palette.*
- *Use the tools to operate and modify front panel and block diagram objects.*

- ***Note:** Press the Shift key and right-click to display a temporary version of the Tools palette at the location of the cursor.*

LabVIEW user interface: Tools Palette

	If automatic tool selection is enabled and you move the cursor over objects on the front panel or block diagram, LabVIEW automatically selects the corresponding tool from the Tools palette. You can disable automatic tool selection and select a tool manually by clicking the tool you want on the Tools palette.
	Use the Operating tool to change the values of a control or select the text within a control.
	Use the Positioning tool to select, move, or resize objects.
	Use the Labeling tool to edit text and create free labels.
	Use the Wiring tool to wire objects together on the block diagram.
	Use the Object Shortcut Menu tool to access an object shortcut menu with the left mouse button.
	Use the Scrolling tool to scroll through windows without using scrollbars.
	Use the Breakpoint tool to set breakpoints on VIs, functions, nodes, wires, and structures to pause execution at that location.
	Use the Probe tool to create probes on wires on the block diagram.Use the Probe tool to check intermediate values in a VI that produces questionable or unexpected results.
	Use the Color Copy tool to copy colors for pasting with the Coloring tool.
	Use the Coloring tool to color an object. It also displays the current foreground and background color settings.

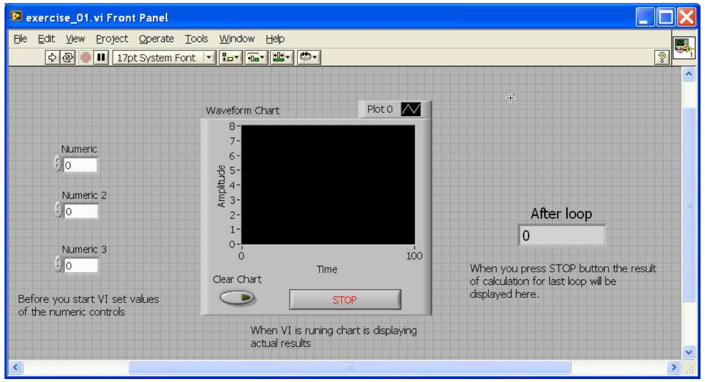
LabVIEW user interface: Front Panel Toolbar

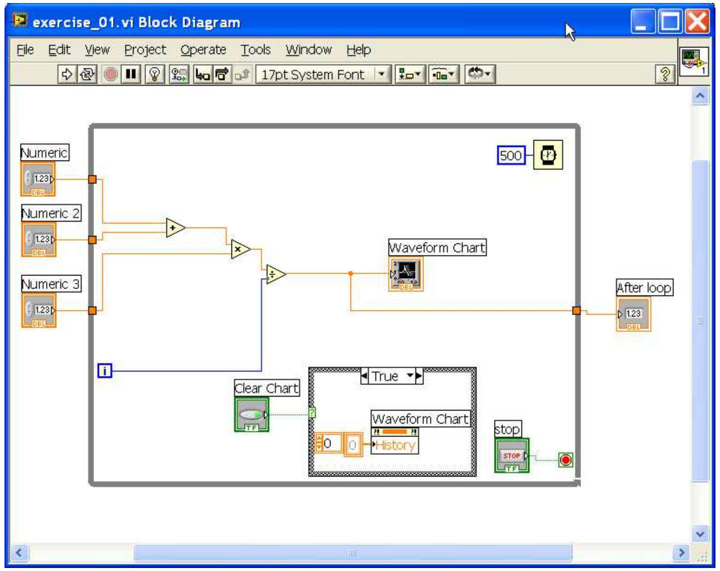
Use the toolbar buttons to run and edit a VI.

	Click the Run button to run a VI. LabVIEW compiles the VI, if necessary. You can run a VI if the Run button appears as a solid white arrow.
	While the VI runs, the Run button appears as shown at left if the VI is a top-level VI, meaning it has no callers and therefore is not a subVI.
	If the VI that is running is a subVI, the Run button appears as shown at left.
	The Run button appears broken when the VI you are creating or editing contains errors Click this button to display the Error list window, which lists all errors and warnings.
	Click the Run Continuously button, shown at left, to run the VI until you abort or pause execution. You also can click the button again to disable continuous running.
	While the VI runs, the Abort Execution button, shown at left, appears. Click this button to stop the VI immediately if there is no other way to stop the VI. Note: Avoid using the Abort Execution button to stop a VI. Either let the VI complete its data flow or design a method to stop the VI programmatically (place a button on the front panel that stops the VI when you click it).
	Click the Pause button to pause a running VI. When you click the Pause button, LabVIEW highlights on the block diagram the location where you paused execution, and the Pause button appears red. Click the button again to continue running VI.
	Select the Text Settings pull-down menu to change the font settings for the selected portions of the VI, including size, style, and color.
	Select the Align Objects pull-down menu to align objects along axes, including vertical, top edge, left, and so on.
	Select the Distribute Objects pull-down menu to space objects evenly, including gaps, compression, and so on.
	Select the Resize Objects pull-down menu to resize multiple front panel objects to the same size.
	Select the Reorder pull-down menu when you have objects that overlap each other and you want to define which one is in front or back of another.
	Select the Show Context Help Window button to toggle the display of the Context Help window.

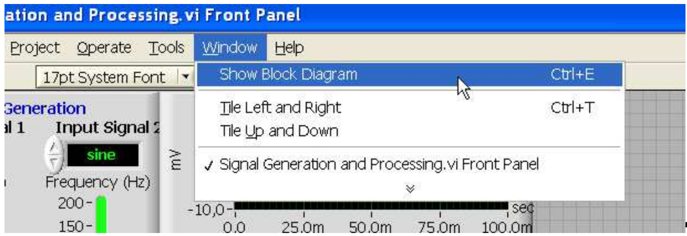
LabVIEW user interface

*Use **Browse...** option on starting screen and open **exercise_01.vi***



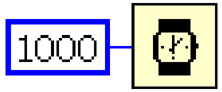


*Open **Block Diagram** using **Window/Show Block Diagram** option (Ctrl+E shortcut).*



LabVIEW Run VI

*Open file **exercise_01.vi**.*

1. *Open Block Diagram. Change time delay to 1000 (1sec.).* 

   Open Front Panel.

2. *Run application. Try to stop application using STOP button* 

   on Front Panel and Abort Execution Button  *on Front Panel Toolbar.*
   Can you explain the difference?

IMPORTANT: ***Avoid using the Abort Execution button to stop a VI. For final application either let the VI complete its data flow or design a method to stop the VI programmatically. By doing so, the VI is at a known state. For example, place a button on the front panel that stops the VI when you click it.***

LabVIEW Run VI

3. Open Block Diagram. Activate Highlight Execution button and observe animation of data flow.

The elements placed on the block diagram are called the <u>nodes</u>.

A block diagram node executes when it receives all required inputs.

When a node executes, it produces output data and passes the data to the next node in the dataflow path.

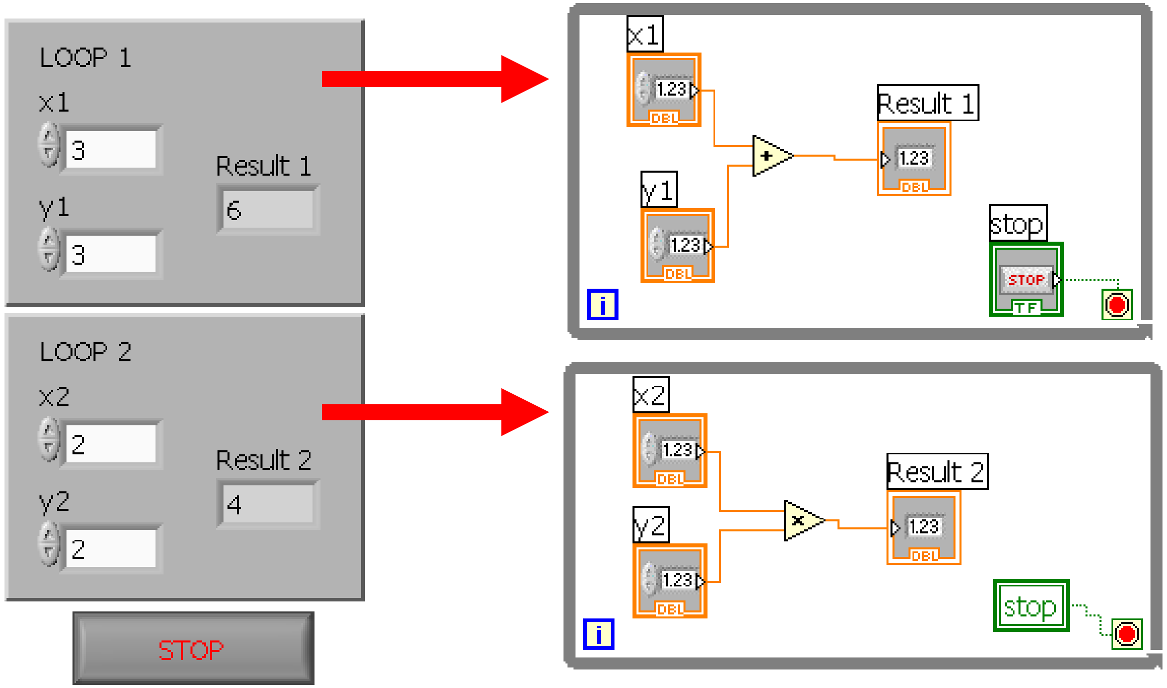
The movement of data through the nodes determines the execution order of the VIs and functions on the block diagram.

<u>Sometimes it is necessary to determine the execution order by arranging its block diagram elements in a certain sequence.</u>

LabVIEW Run VI

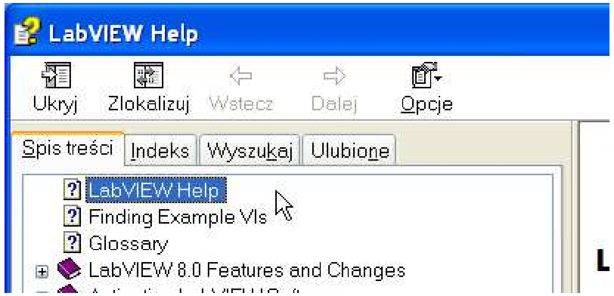
Open another example: ***data flow example.vi****.*

Activate Highlight Execution button  *and observe animation of data flow. It can be visible, that both loops are executed simultaneously.*

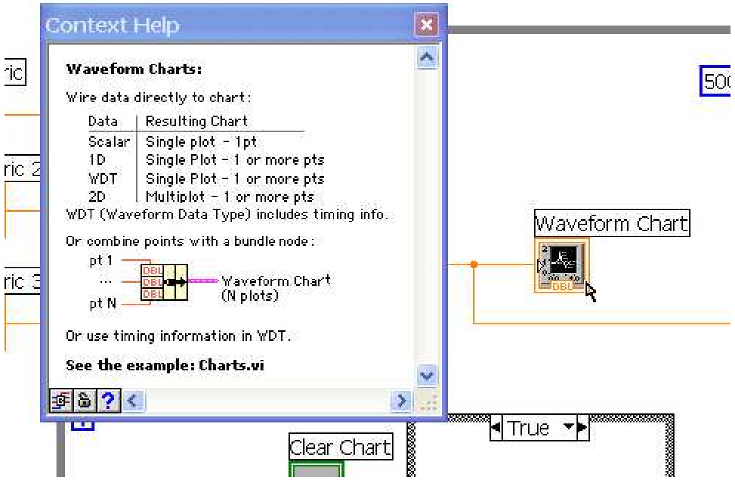


LabVIEW Help

Use ***Help/Search the LabVIEW Help...*** *to open help window, which has got a standard MSWindows looks:*



For immediately, permanent help activate ***Help/Show Context Help (Ctrl+H)*** *to open small window with information about actually selected element.*



LabVIEW Help