Sviluppa un DataLogger multi-sensore con LabVIEW e CompactDAQ

Luca Gallo – Academic & Research Sales Engineer

National Instruments

Today, We'll Explore:

The Challenges of Making Measurements	Characteristics of Mixed-Measurement Systems
	The National Instruments Approach
	Architecture of a Measurement System
Introduction to LabVIEW	Navigating the LabVIEW Environment
Break	
Fundamentals of Data Acquisition	Essential Data Acquisition Concepts
	The Basics of Signal Conditioning
	The Value of National Instruments Hardware Platforms
Uniting Software and Hardware	Architecture of the NI-DAQmx Driver
	Measurement Services and Utilities
	Exploring and Using the NI-DAQmx API

The Challenges of Making Measurements

Exploring the Traditional Approach to Measurements

The Origin of Automated Measurements

- Traditional pen-and-paper approach
- Redundant circuitry between instruments (e.g., displays)
- Manual data recording and analysis
- Error-prone processes
- Difficult to reproduce or redo

Thermoelectric Voltage in mV											
°C	0	1	2	3	4	5	6	7	8	9	10
0	0.000	0.050	0.101	0.151	0.202	0.253	0.303	0.354	0.405	0.456	0.507
10	0.507	0.558	0.609	0.660	0.711	0.762	0.814	0.865	0.916	0.968	1.019
20	1.019	1.071	1.122	1.174	1.226	1.277	1.329	1.381	1.433	1.485	1.537
30	1.537	1.589	1.641	1.693	1.745	1.797	1.849	1.902	1.954	2.006	2.059
40	2.059	2.111	2.164	2.216	2.269	2.322	2.374	2.427	2.480	2.532	2.585
50	2.585	2.638	2.691	2.744	2.797	2.850	2.903	2.956	3.009	3.062	3.116
60	3.116	3.169	3.222	3.275	3.329	3.382	3.436	3.489	3.543	3.596	3.650
	.650	3.703	3.757	3.810	3.864	3.918	3.971	4.025	4.079	4.133	4.187
		4.240	4.294	4.348	4.402	4.456	4.510	4.564	4.618	4.672	4.726
			4.835	4.889	4.943	4.997	5.052	5.106	5.160	5.215	5.269

Measurement Challenges Are Compounded By:

- Compressed Timelines
- Fixed Software and Hardware
- Conflicting Programming Approaches
- Inadequate Hardware Performance
- Disparate Driver APIs
- Varying Sensors and Connectivity
- Custom Signal Conditioning
- Advanced Visualization
- Changing Application Requirements
- Complex Analysis Algorithms
- Evolving Technology Trends
- Confusing Data Storage
- Differing Sampling Rates

NATIONAL INSTRUMENTS

Mixed-Measurement Applications Are Diverse

Vibration

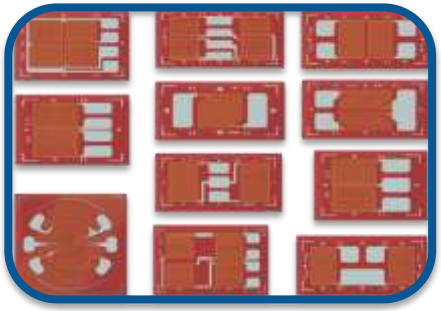
Torque

Displacement

Pressure

Temperature

Force



Strain

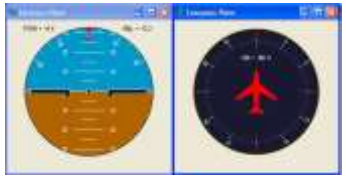
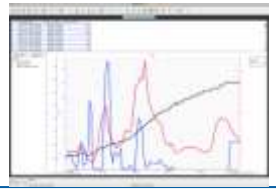
Example Application: Air Quality Measurements

- Potential Sensors Needed:
  - Context
    - GPS
      - Timestamp
      - Position
    - Attitude
    - Altitude
    - Range Finder
  - Environmental
    - Temperature
    - Oxygen
    - Carbon Dioxide
    - Ozone
    - Nitrogen

Sensors, Interfaces, and Signal Conditioning

Sensor		Interface	Conditioning?
GPS		RS232	No
Attitude, Altitude		RS232	No
LiDAR		Ethernet	No
Temperature		Analog Voltage	Required
O_2, CO_2, O_3, NH_3		Analog Voltage	Required

Software Provided With Sensors

Sensor		Software
GPS		
Attitude, Altitude		
LiDAR		
Temperature		
O_2, CO_2, O_3, NH_3		<No Software Provided>

NATIONAL INSTRUMENTS

With a System Like This, How Do You Accommodate…

- …changes in requirements?
- …mixed measurements in a single system?
- …varying connectivity?
- …signal conditioning for sensors?
- …adding or replacing measurements or sensors?
- …incorporating timing, triggering, or synchronization?
- …leveraging emerging technology trends?
- …multiple disparate software environments and APIs?

National Instruments' Strategy: Graphical System Design

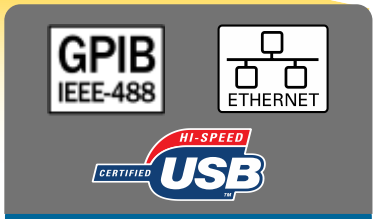
Your Investment in a Platform-Based Approach to Measurements Scales Across…

NATIONAL INSTRUMENTS LabVIEW™

Desktops and PC-Based DAQ

PXI and Modular Instruments

NI CompactRIO



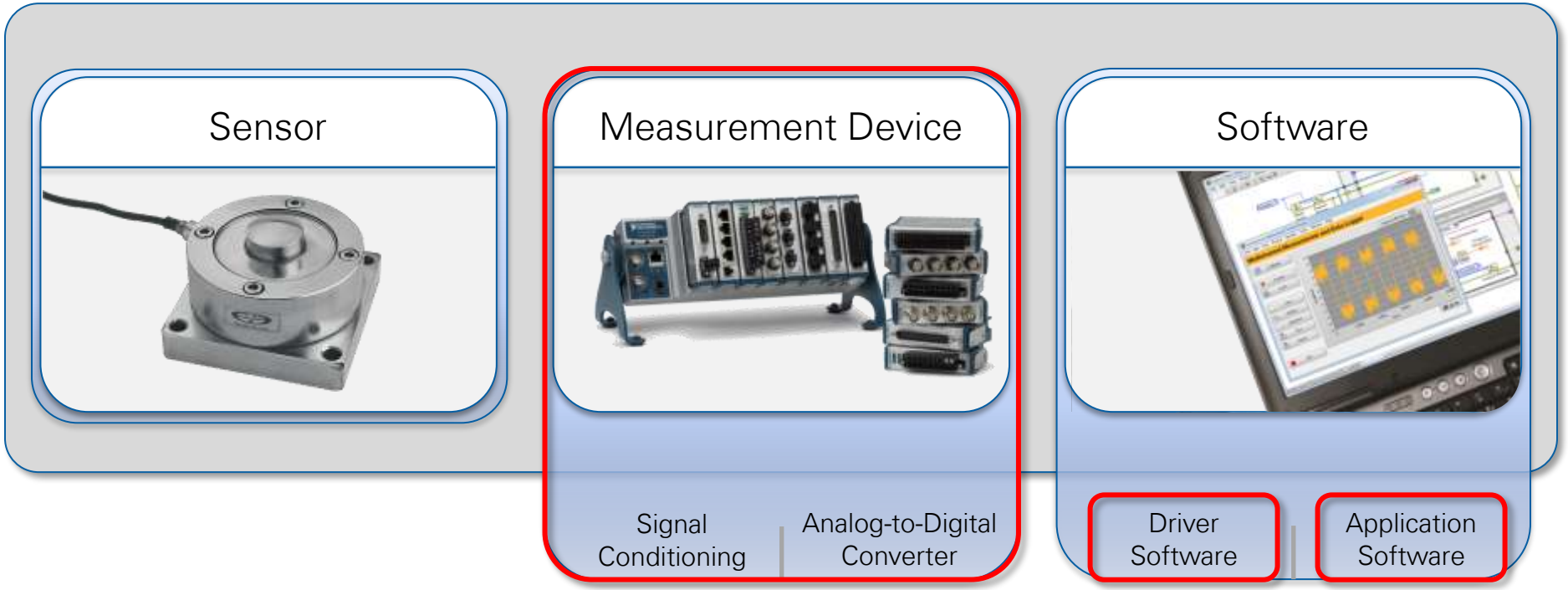
Open Connectivity With Third-Party I/O

Top Benefits of an Integrated Measurement *Platform*

1. Accelerated Productivity
2. Proven Performance and Accuracy
3. Scalability, Adaptability, and Flexibility

Architecture of an Integrated Measurement System

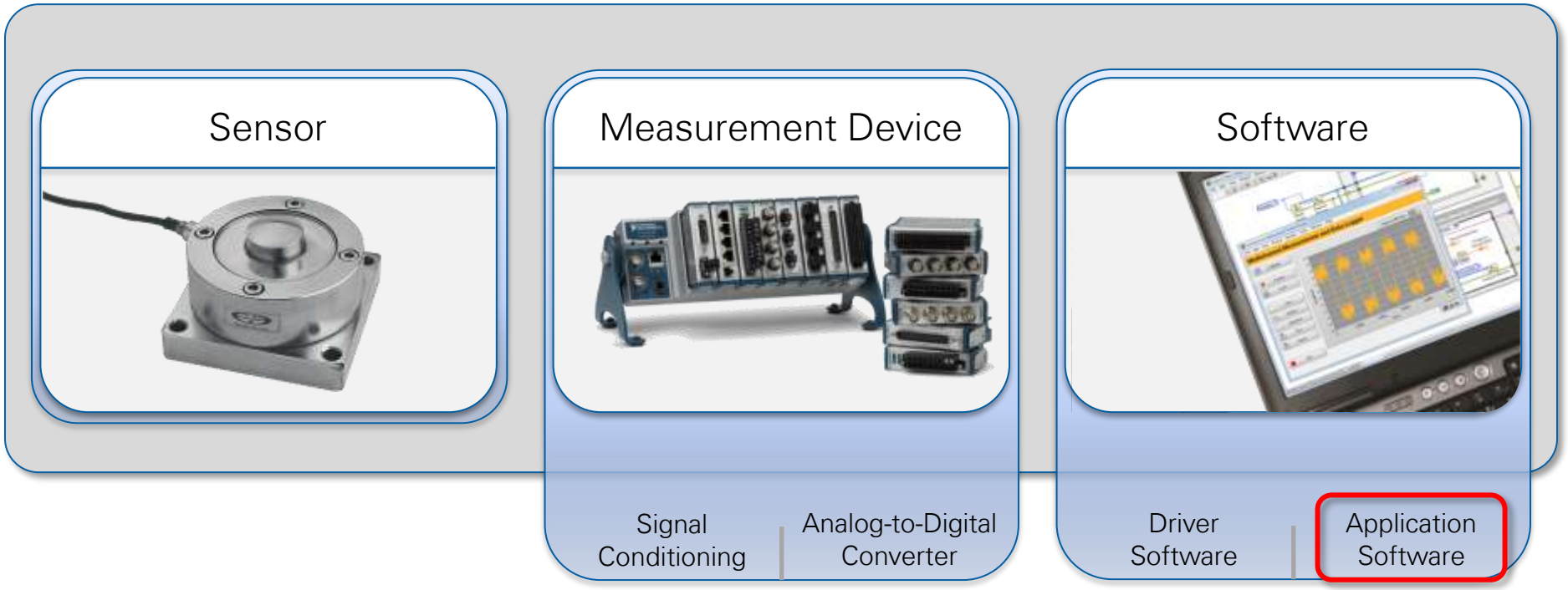
Today, we'll learn about three key differentiating components of a National Instruments data acquisition system:



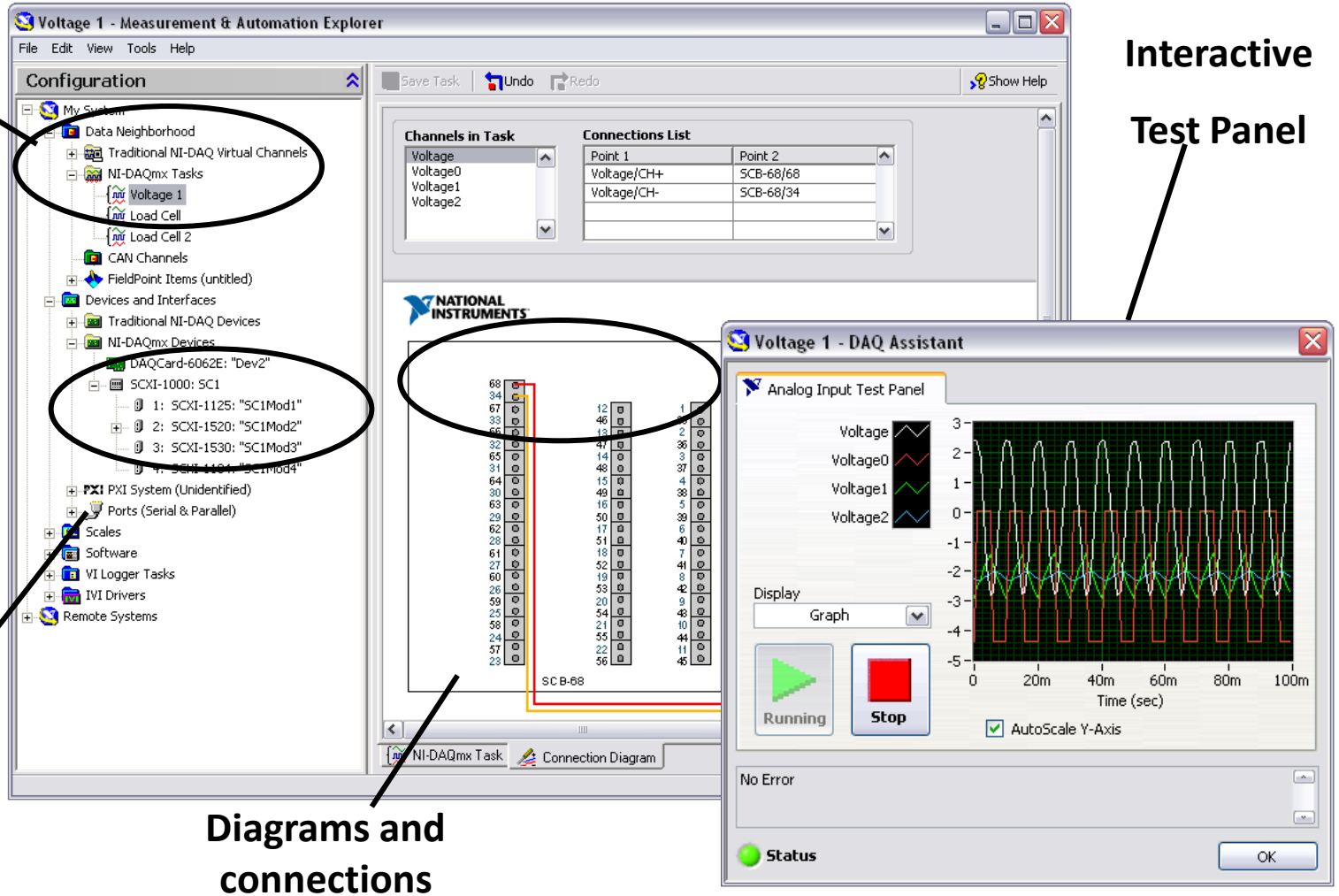
Architecture of an Integrated Measurement System

NATIONAL INSTRUMENTS LabVIEW™

LabVIEW is system design software that provides engineers and scientists with the tools needed to create and deploy measurement and control systems through unprecedented hardware integration.



Exercise 1a: Measurement & Automation Explorer



Introduction to LabVIEW

System Design Software for Any Measurement Application

Unrivaled Hardware Integration in a Single Environment

- NI hardware
  - 200+ data acquisition devices
  - 450+ modular instruments
  - Cameras
  - Motion control
- Third-party hardware
  - Instrument Driver Network
    - 10,000+ instrument drivers
    - 350+ instrument vendors
    - 100+ instrument types
  - Communicate over any bus

The Foundation of LabVIEW: Virtual Instrumentation

Automation through software led to a realization about fixed-functionality instrumentation…

Redundancy: Power Supplies
Each separate instrument requires its own power supply to run measurement circuitry that captures the real-world signal.

Redundancy: Displays
Instrument vendors provide a limited-quality display per instrument, even though monitor technology is far more advanced.

Redundancy: Processors
Chip manufacturers rapidly enhance processors according to Moore's law, but instruments have fixed processing power.

Redundancy: Memory
PCs can quickly capitalize on a performance boost from a memory upgrade from readily available RAM.

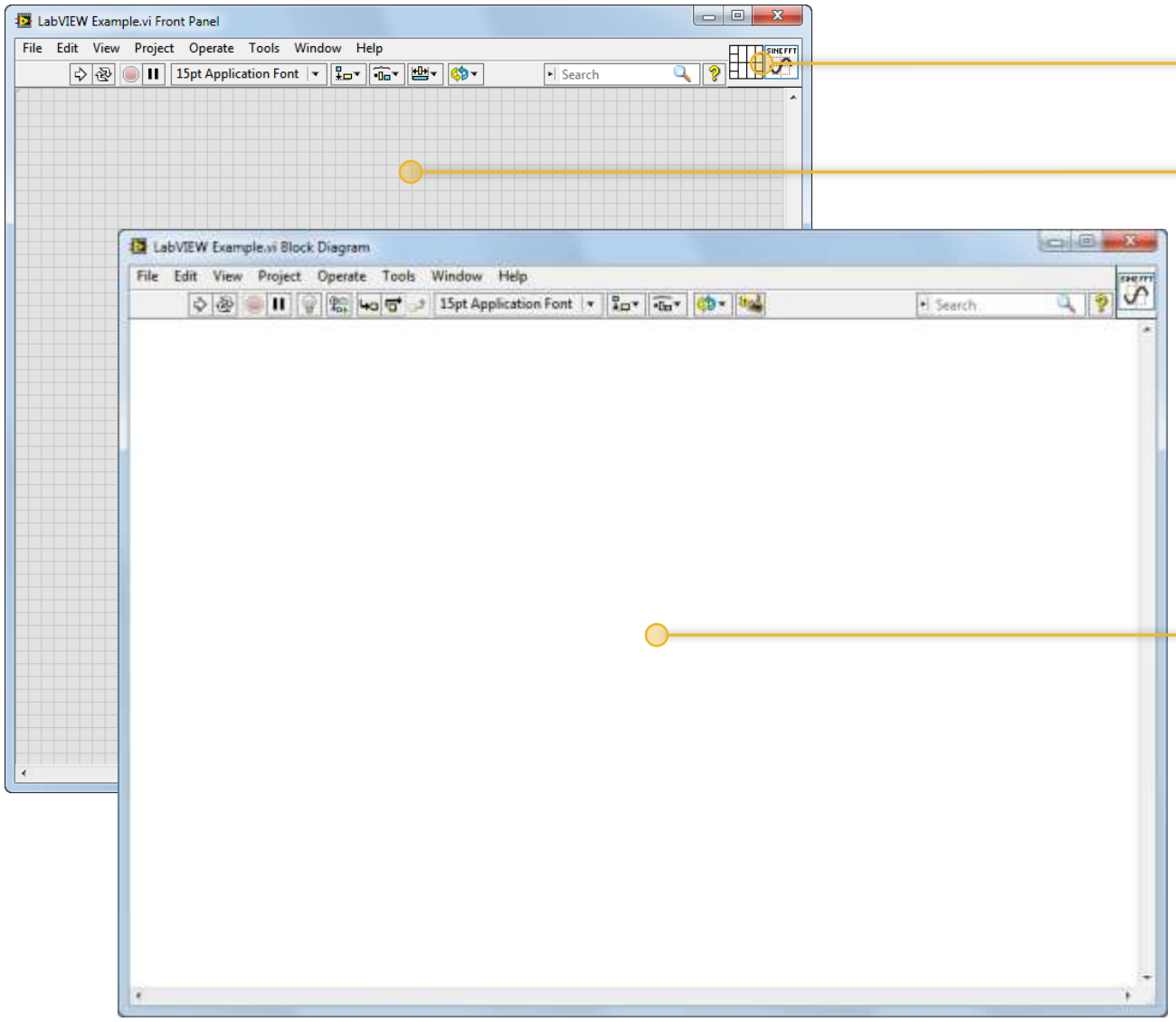
Redundancy: Storage
Each instrument duplicates onboard storage even though PC hard drives are plentiful and cost-effective.

The Foundation of LabVIEW: Virtual Instrumentation

By leveraging COTS PC components, the software becomes the instrument

LabVIEW unlocks the power of instrument and data acquisition hardware by capitalizing on the PC industry and abstracting redundant circuitry.

Therefore, LabVIEW Building Blocks Are Called Virtual Instruments (*.VI)



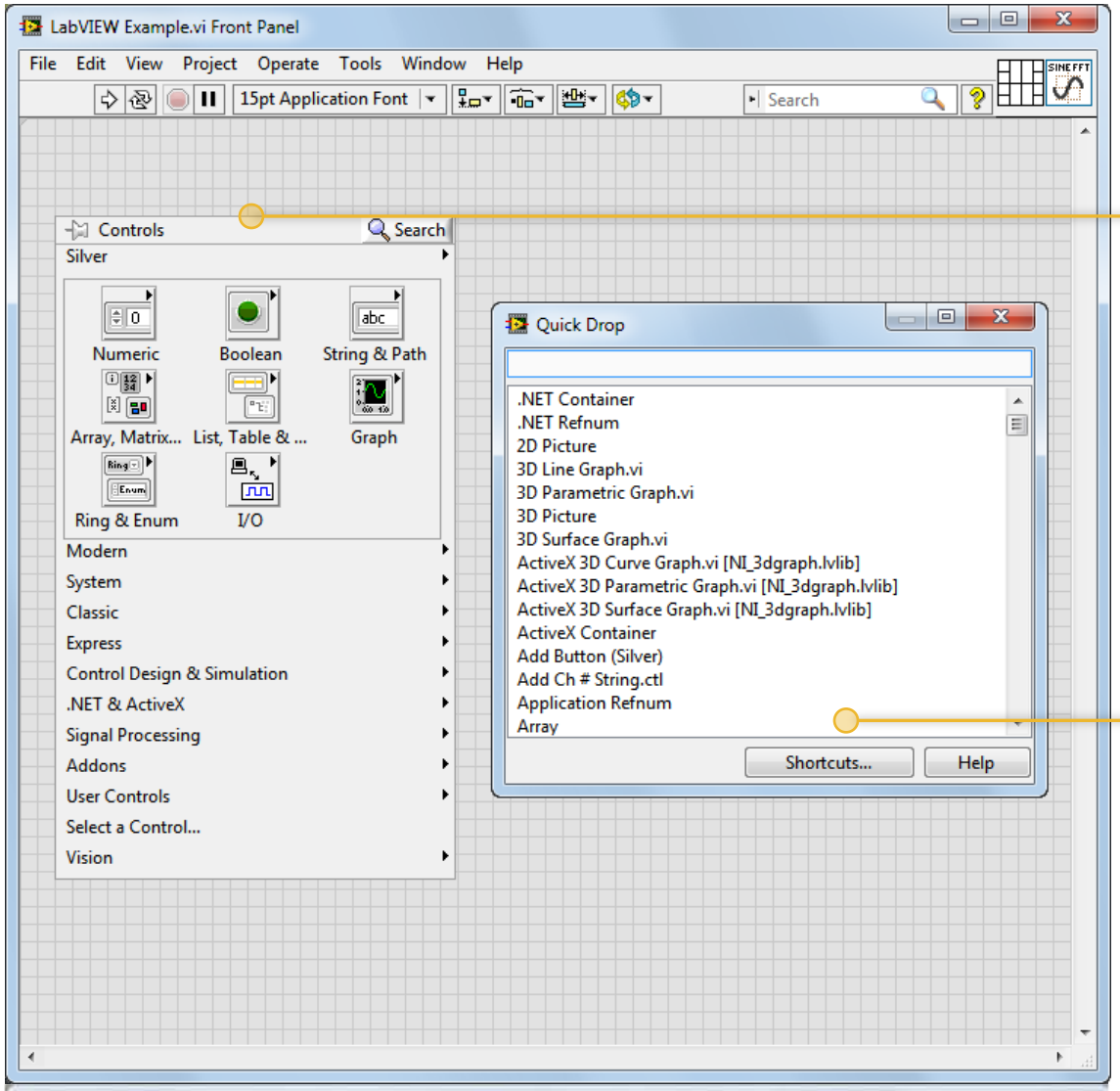
Icon / Connector Pane
Maps inputs and outputs

LabVIEW Front Panel
The user interface of a VI

LabVIEW Block Diagram
The source code of a VI

*Note: A *.vi file encapsulates all three elements*

Creating a LabVIEW Front Panel

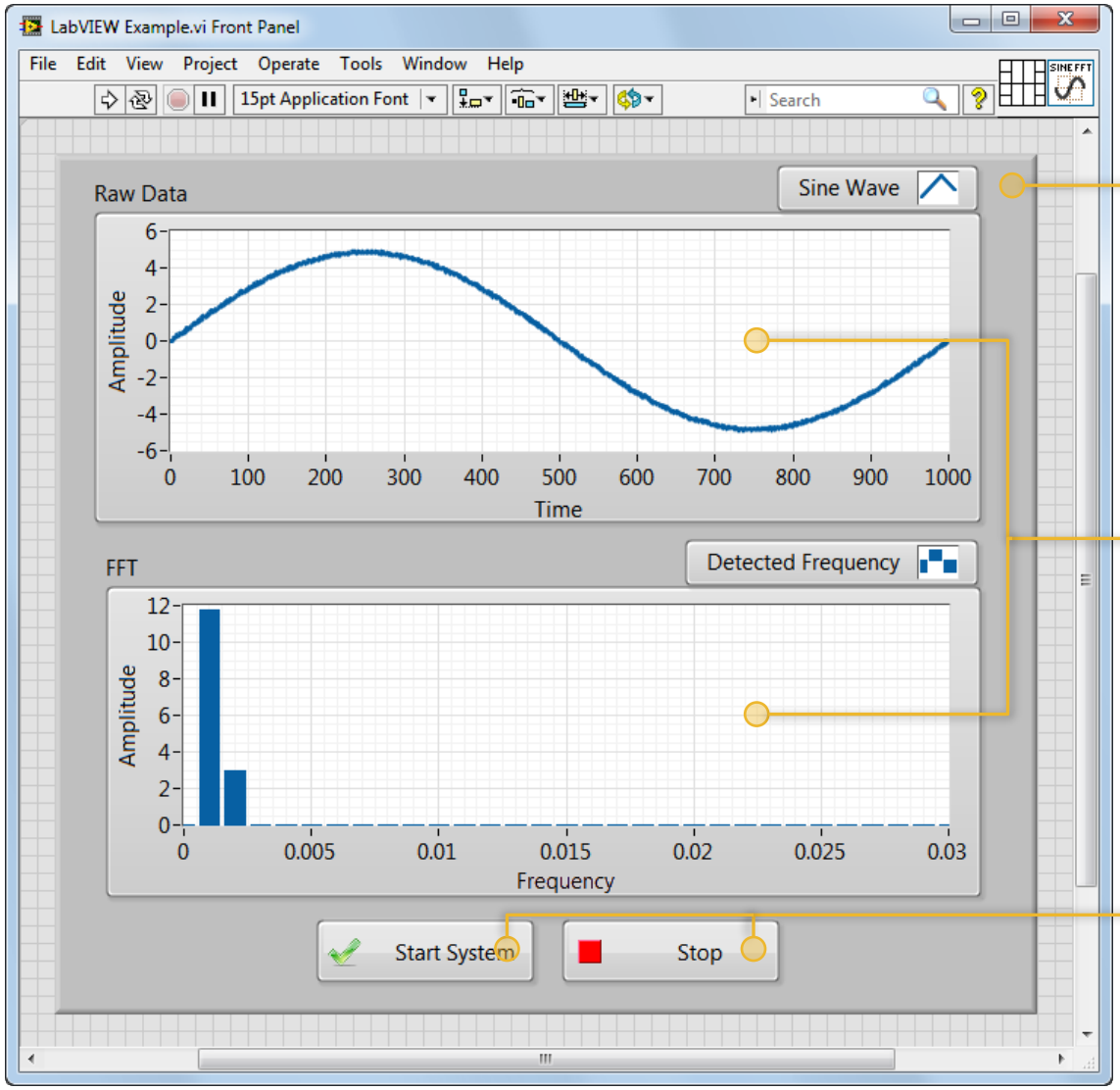


Controls Palette (Right-Click)
Access a hierarchical palette of all front panel elements.

Quick Drop (Ctrl + Space)
Search by object name.

NATIONAL INSTRUMENTS

Front Panel Objects



Decorations

Decorative elements and imagery

- Text
- Arrows
- Callouts
- Lines
- Images
- …and more

Customizable Indicators

Used to convey outputs to a user

- Graphs and Charts
- Progress Bars
- Gauges and Meters
- LEDs
- Numerics
- Strings and Paths
- …and more

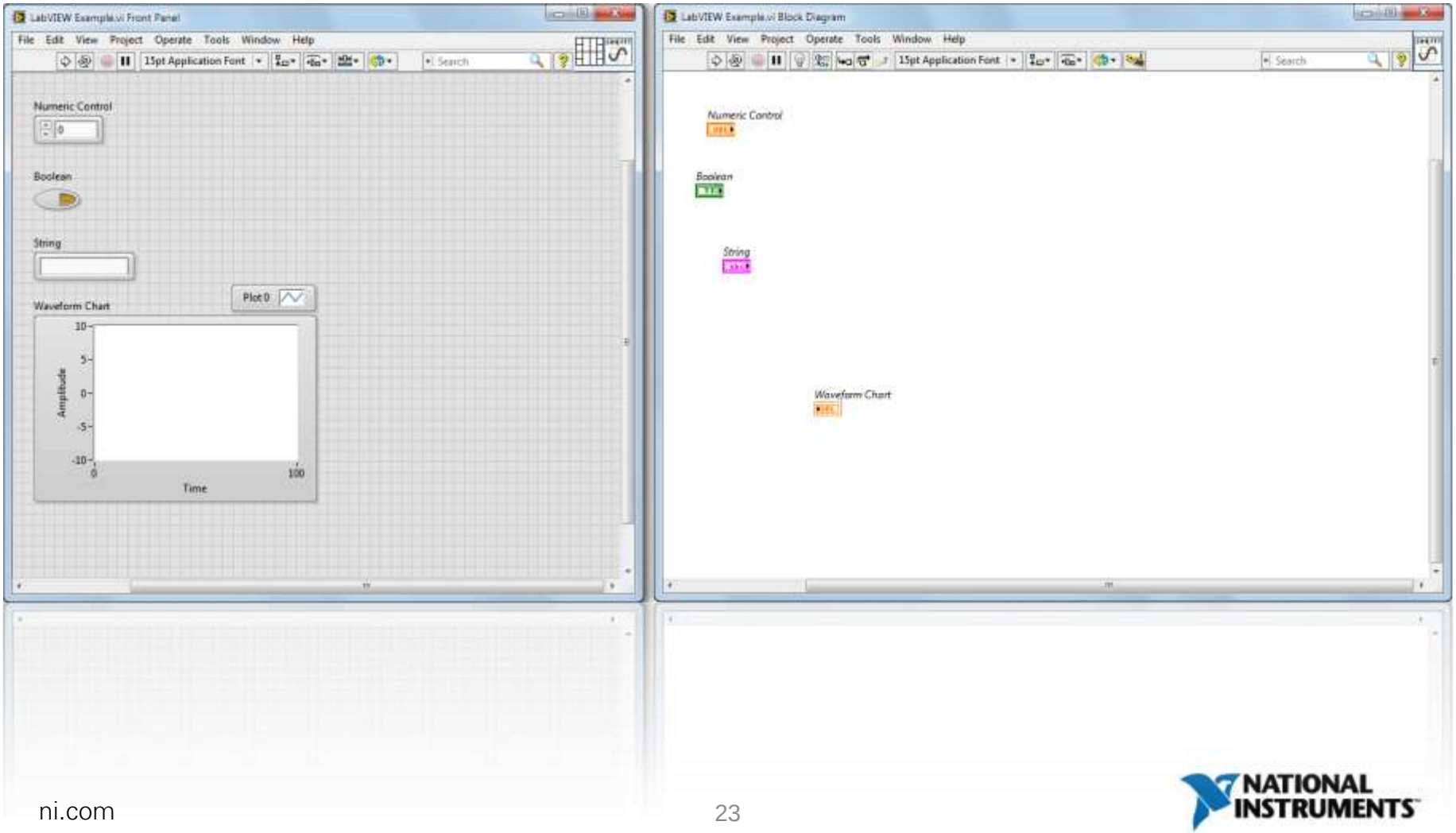
Customizable Controls

Used to receive input from a user

- Knobs and Dials
- Sliders
- Buttons
- Numerics
- Strings and Paths
- …and more

All Front Panel Elements Have Block Diagram Terminals

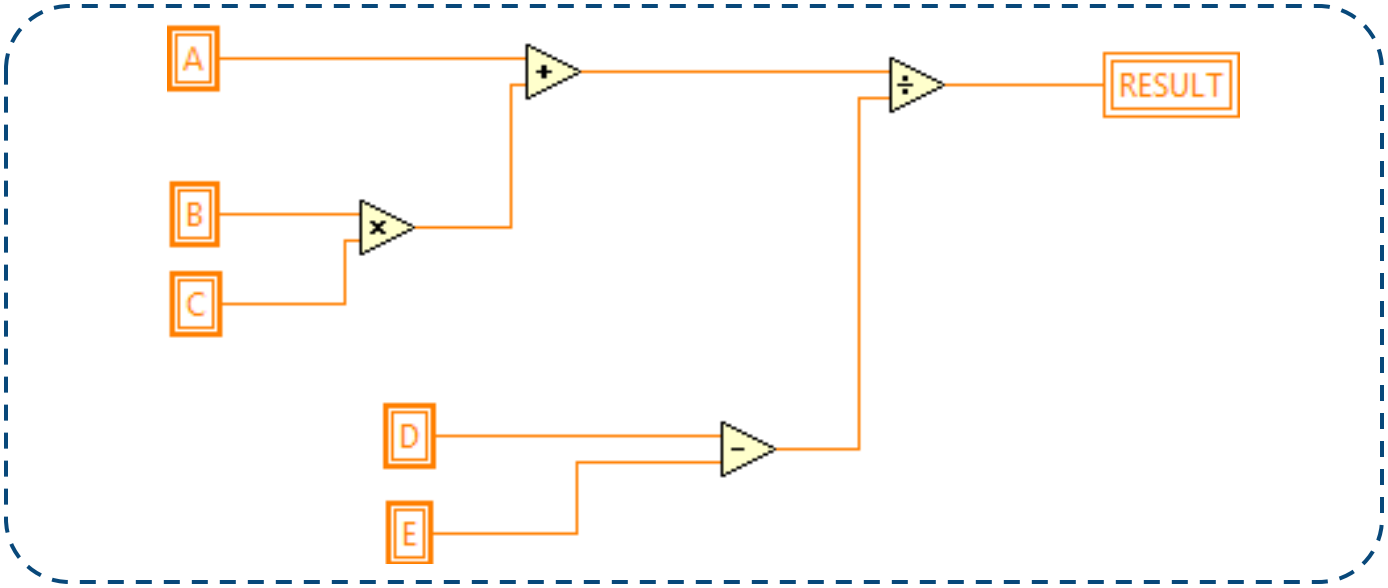
Block diagram terminals provide access to front panel values



NATIONAL INSTRUMENTS

What Is Data Flow?

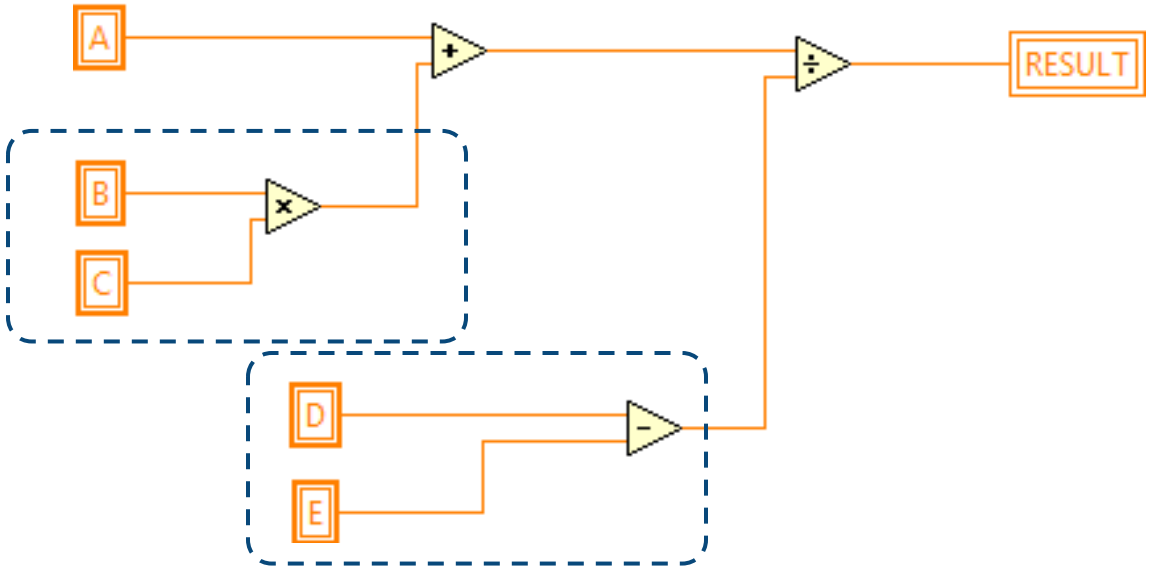
- Each block diagram node executes only when it receives all inputs
- Each node produces output data after execution
- Data flows along a path defined by wires
- The movement of data determines execution order



Formula: Result = (A+B*C) / (D-E)

What Is Data Flow?

- Each block diagram node executes only when it receives all inputs
- Each node produces output data after execution
- Data flows along a path defined by wires
- The movement of data determines execution order

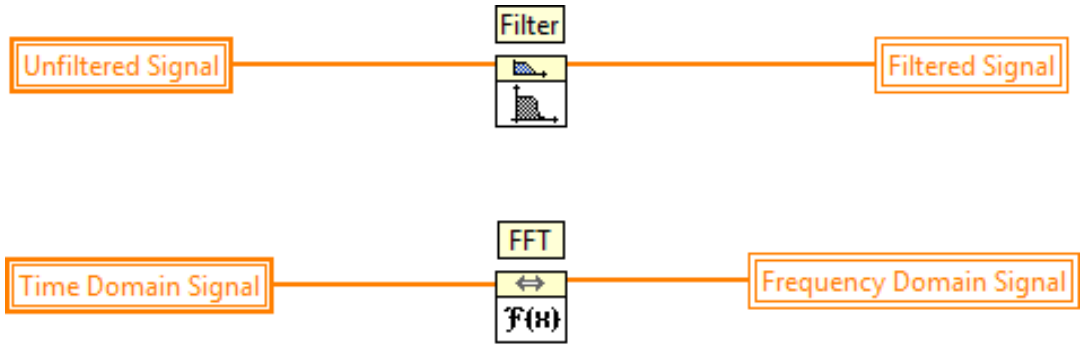


The [Multiply] and [Subtract] operations can execute at the same time since they don't have any data dependencies.

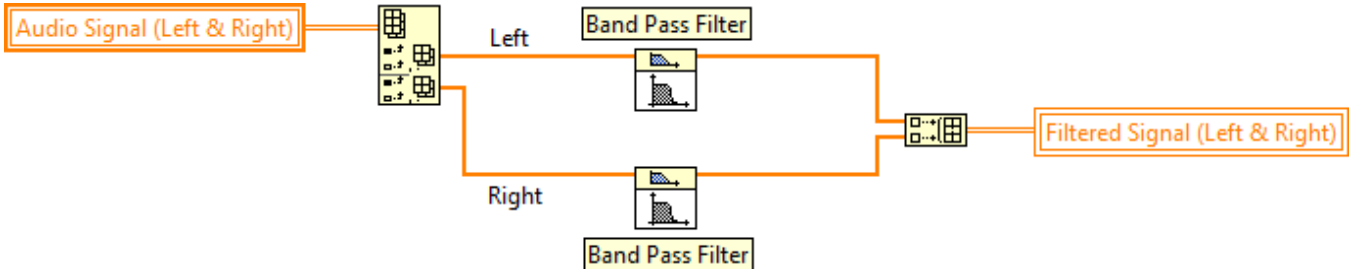
Dataflow Languages Naturally Express Parallelism

The LabVIEW compiler will automatically multithread code expressed in parallel

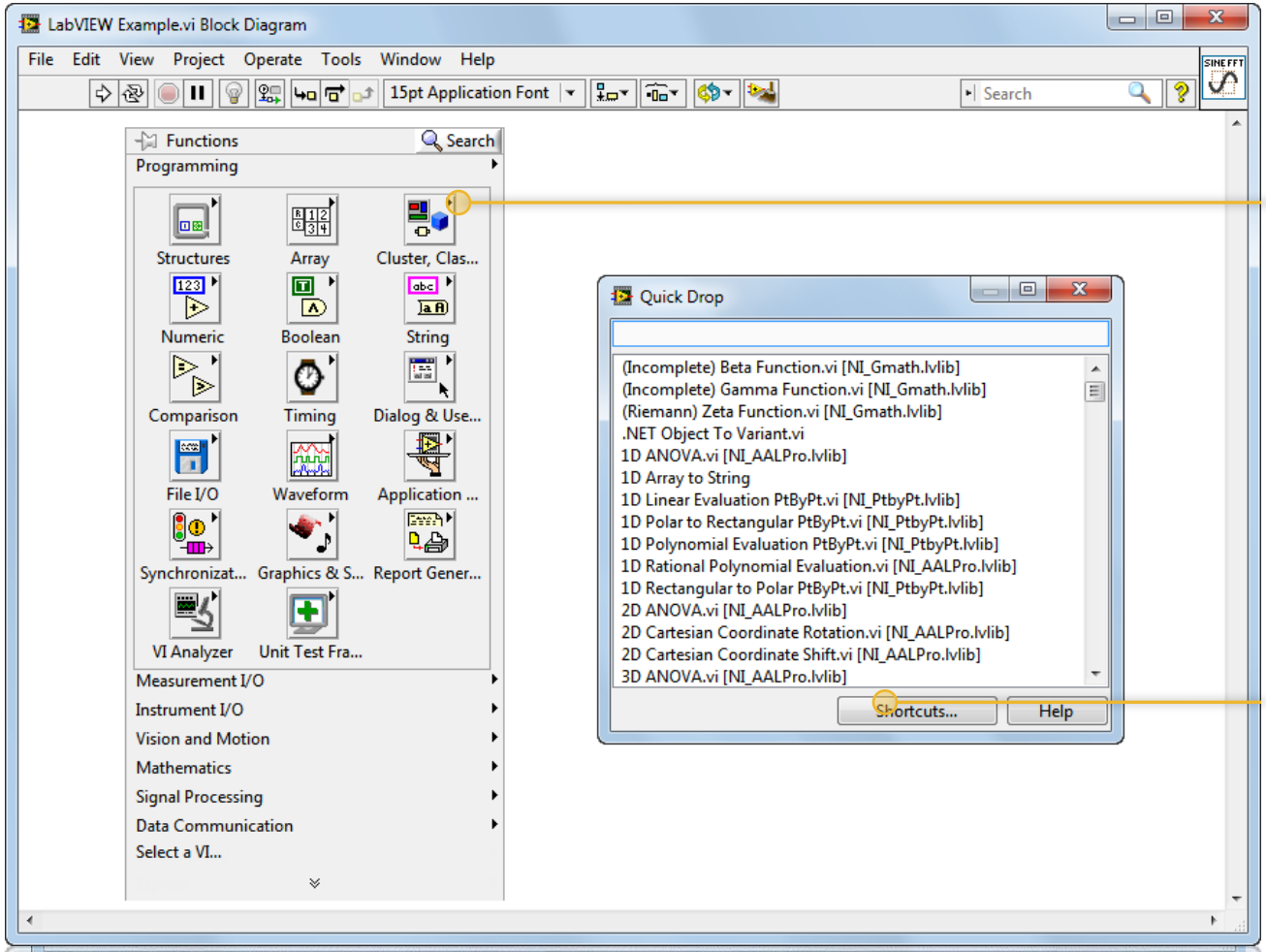
Task Parallelism



Data Parallelism



Creating a LabVIEW Block Diagram



Functions Palette (Right-Click)
Access a hierarchical palette of all block diagram functions.

Quick Drop (Ctrl + Space)
Search by object name.

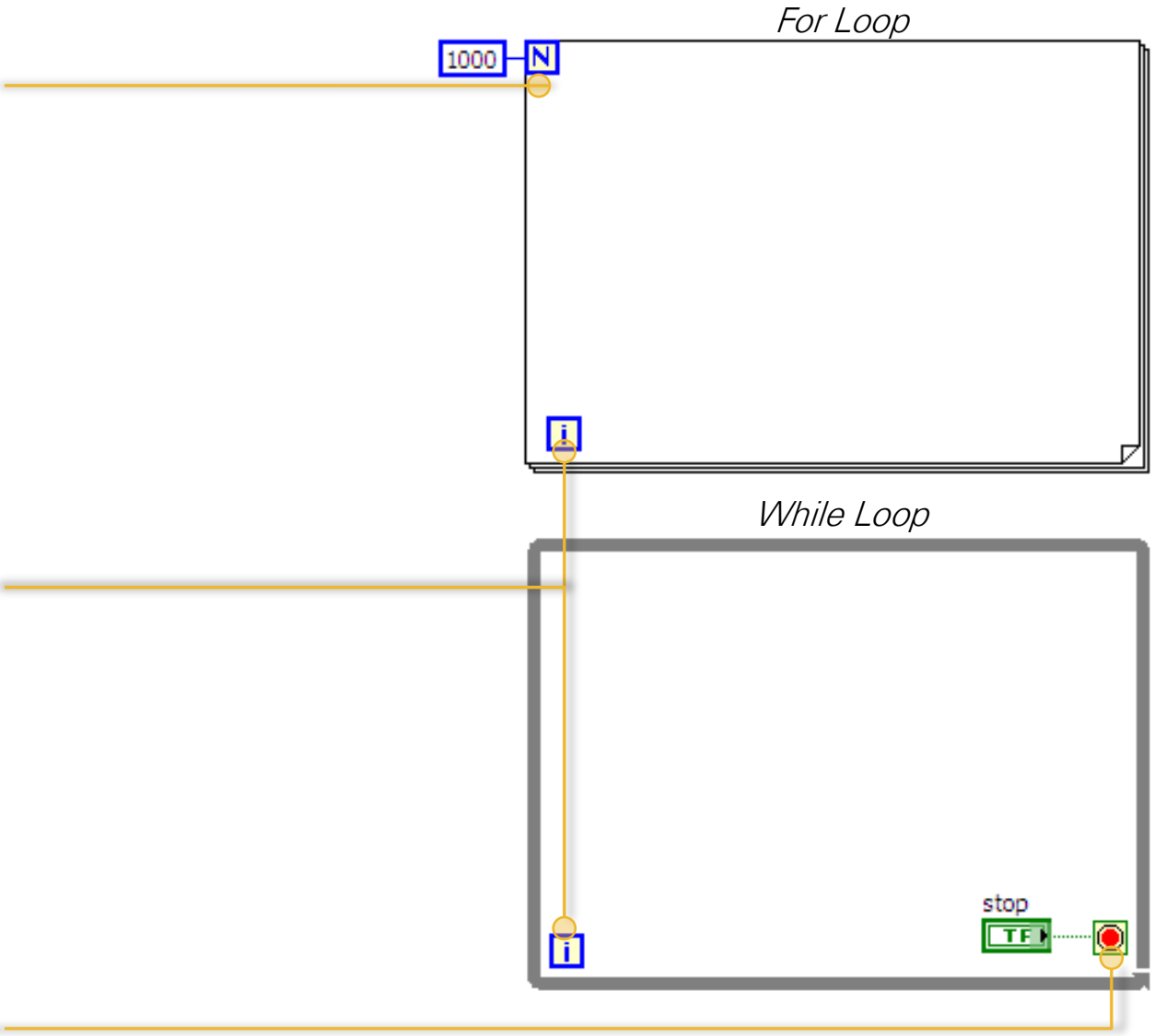
NATIONAL INSTRUMENTS

Execution Control Structures: Loops

Count Terminal
The code contained within this For Loop will execute N times.

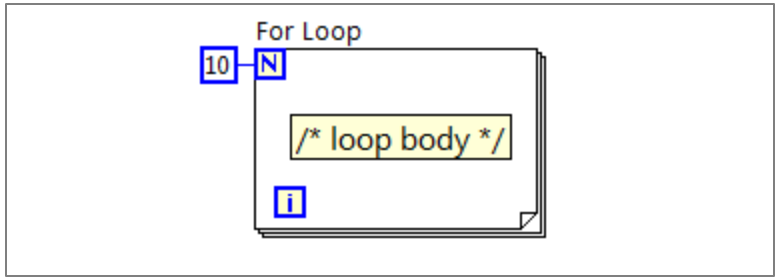
Loop Iteration Terminals
This provides the current loop iteration count, which ranges from 0 to N-1.

Conditional Terminal
The code within this While Loop will run until a True value is evaluated.

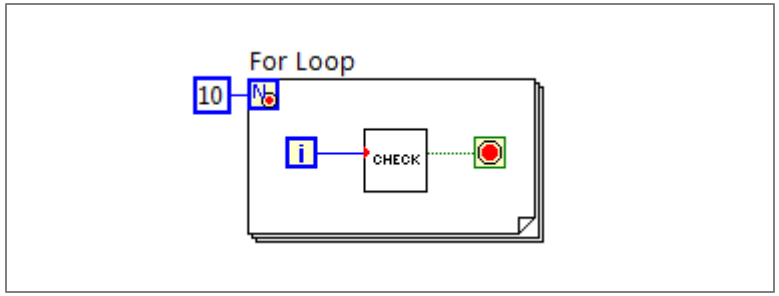


Text Loops and Their LabVIEW Equivalents

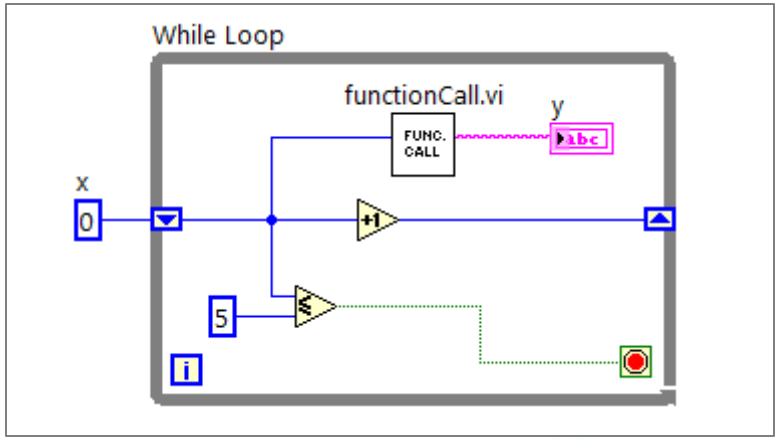
```
for (i = 0; i < 10; i++)
{
    /* loop body */
}
```




```
for (i = 0; i < 10; i++)
{
  if(check(i)) break;
}
```




```
int x = 0;
String y;
while (x < 5)
{
    y = functionCall(x);
    printf(y);
    x++;
}
```

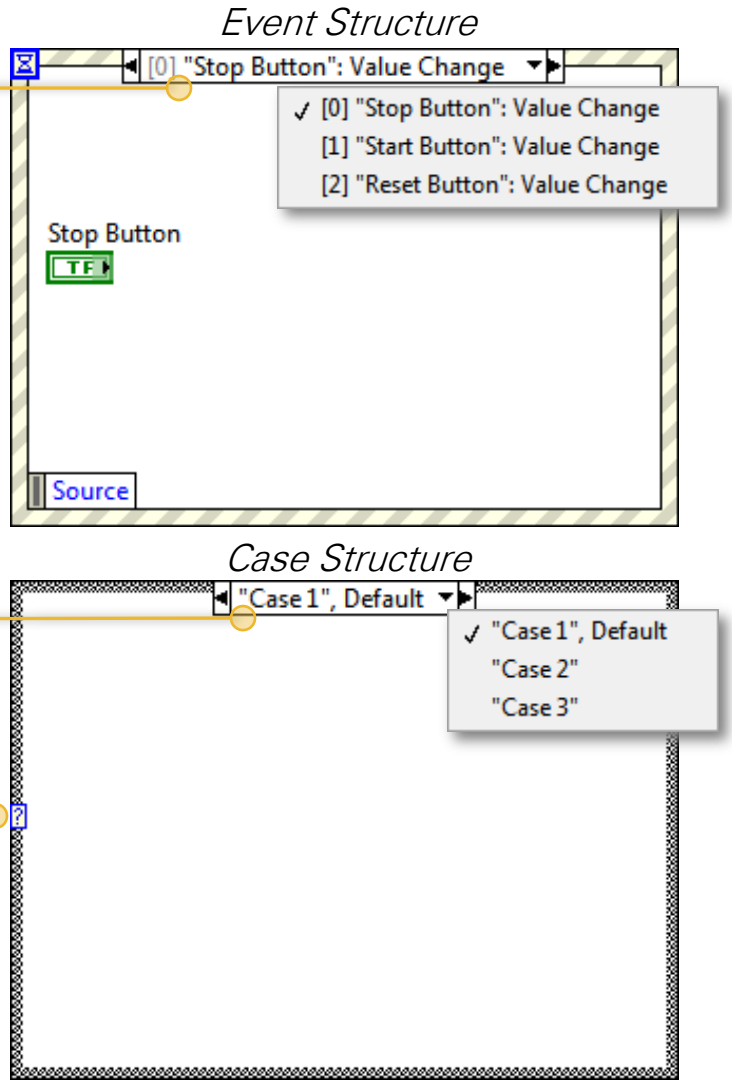


Event and Case Structures

Event Selector Label
This indicates which subdiagram is visible and details the event that the code within the diagram handles.

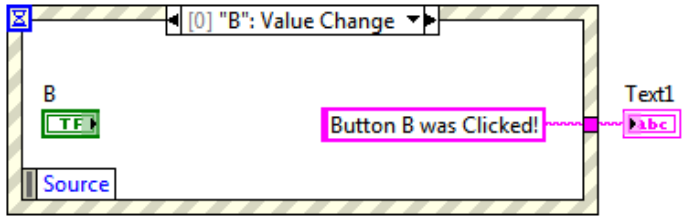
Case Selector Label
This indicates which subdiagram is visible.

Selector Terminal
The value wired to this terminal determines which of the subdiagrams, or cases, will execute.

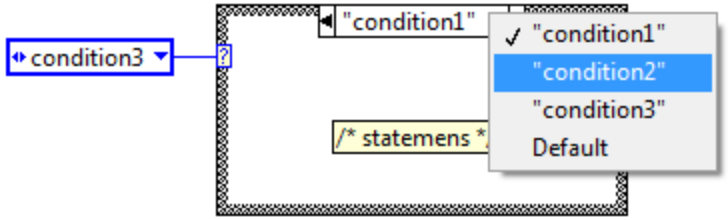


Text Events, Cases, and Their LabVIEW Equivalents

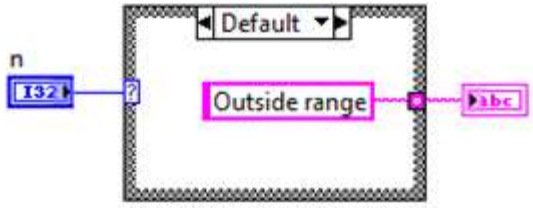
```
Button B = new Button();
B.Click += new RoutedEventHandler(OnBClick);

void OnBClick(object Source)
{
      Text1.Text = "Button B was Clicked!";
}
```



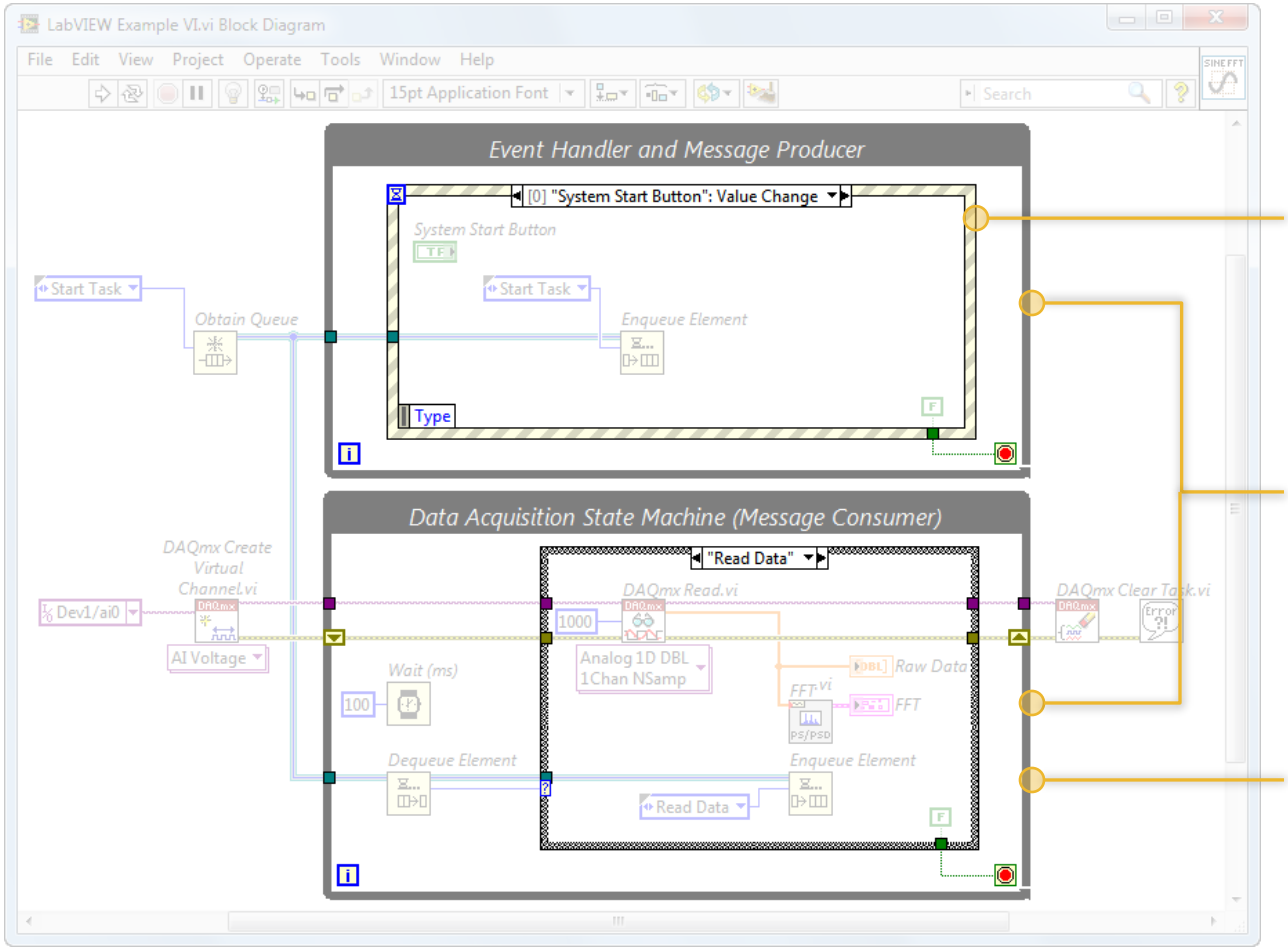

```
if condition1 then
  -- statements;
elseif condition2 then
  -- more statements
elseif condition3 then
  -- more statements;
else
  -- other statements;
end if
```




```
switch (n) {
  case 5:
    printf("Small number.");
    break;
  case 100:
    printf("Large number.");
    break;
default:
    printf("Outside range");
    break;
}
```



Exploring a LabVIEW Block Diagram



Event Structure
Executes different subdiagrams based on events and interrupts

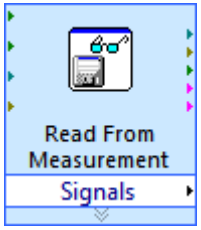
While Loops
Iterate continuously until a true value is passed to the stop terminal

Case Structure
Executes different subdiagrams based on the value of its selector terminal

LabVIEW Functions Are as Complex as You Need

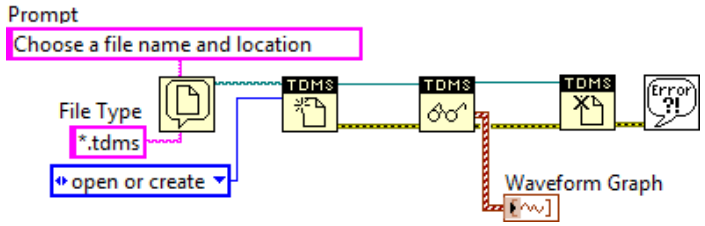
Express VIs

- Quick and Easy
- Configuration-Based
- Limited



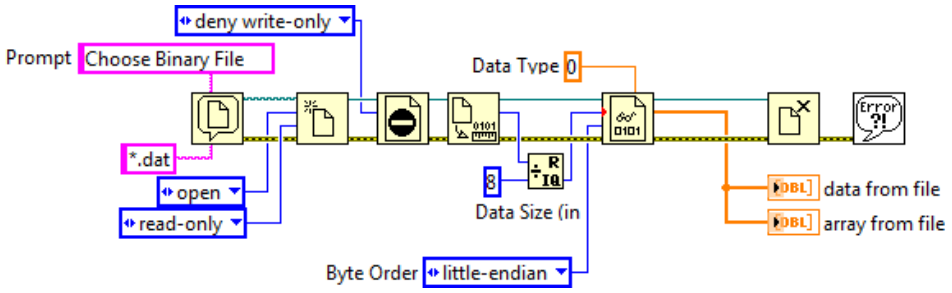
Regular VIs

- Hides Unnecessary Details
- Retains Power and Flexibility



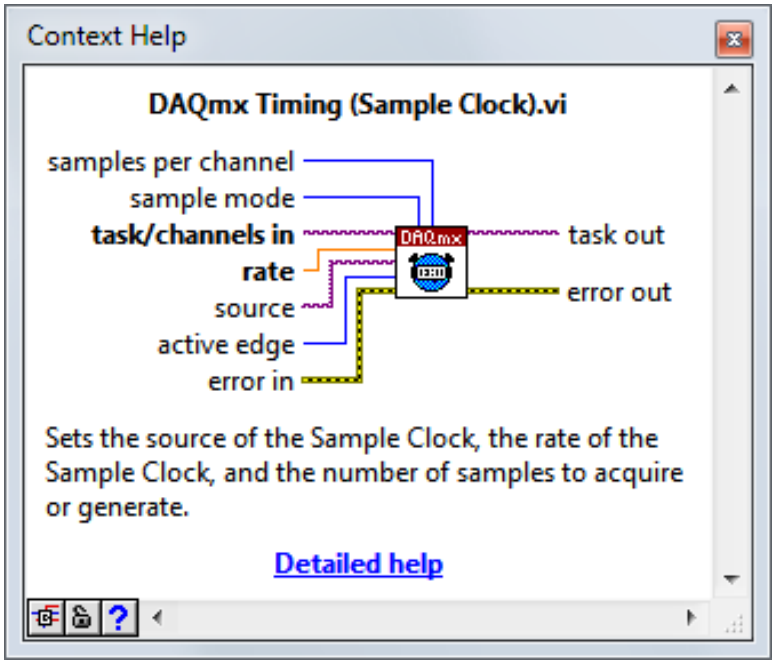
Low-Level VIs

- Powerful, Flexible
- Difficult, Time-Consuming



Understanding SubVI (Function) Behavior

- Code will only compile if required inputs are wired
- Required inputs are **Bold**
- If an optional input is not supplied, a default value will be used for execution

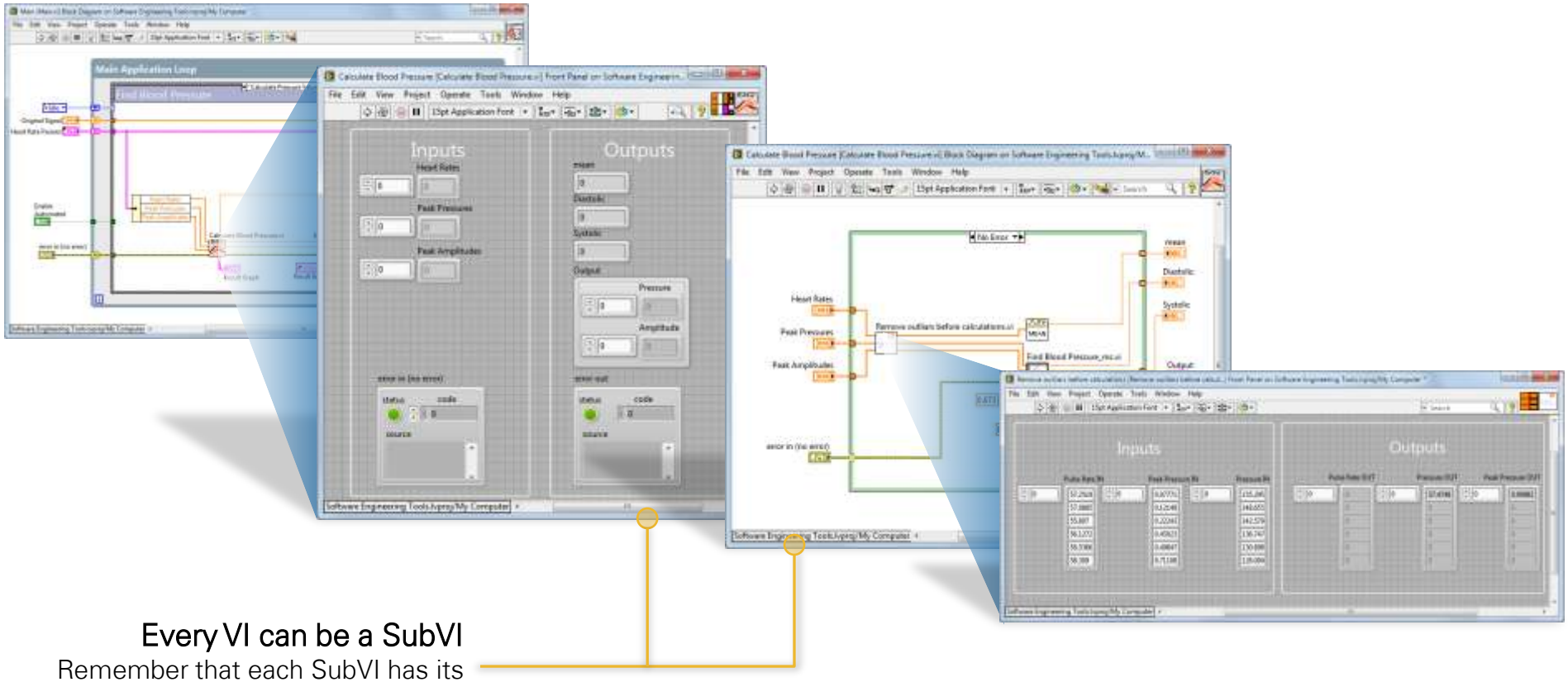


Tip: Access the Context Help using Ctrl+H

NATIONAL INSTRUMENTS

Understanding Application Hierarchy

Double-clicking a nonprimitive SubVI opens the function

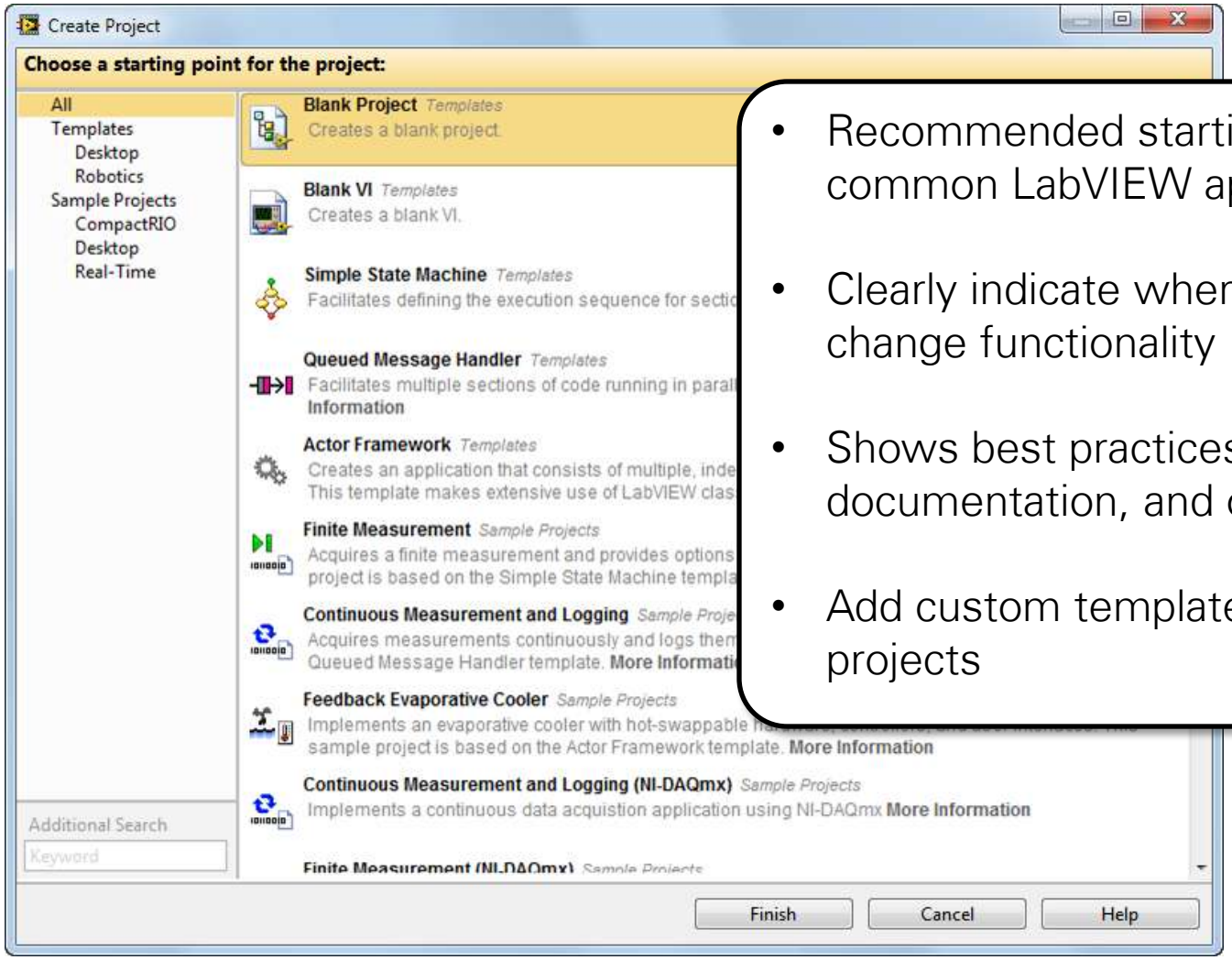


Every VI can be a SubVI
Remember that each SubVI has its own front panel and block diagram.

NATIONAL INSTRUMENTS

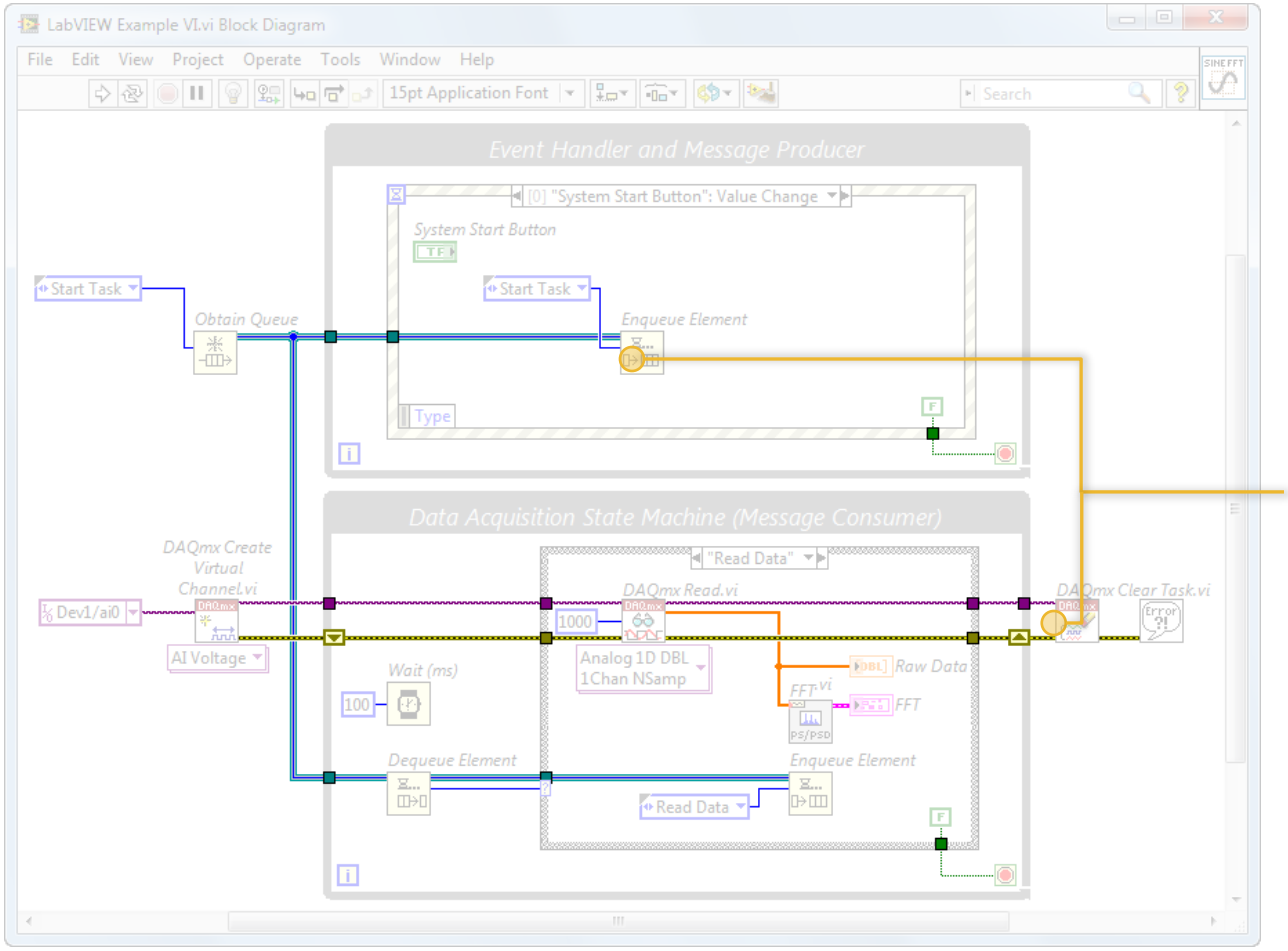
Never Start a LabVIEW Project From Scratch

Abundant sample projects and templates provide a scalable starting point



- Recommended starting points for common LabVIEW applications
- Clearly indicate where to add or change functionality
- Shows best practices for code design, documentation, and organization
- Add custom templates and sample projects

Exploring a LabVIEW Block Diagram

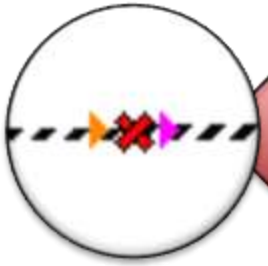


Wires
Data flows on wires between nodes on the block diagram.

The color of the wire indicates its data type, which is strictly enforced at edit-time.

The Color, Style, and Thickness of Common Wires

Wire Type	Scalar	1D Array	2D Array	Color
Floating Point				Orange
Integer				Blue
Boolean				Green
String				Pink
Error				Yellow



A "broken wire" represents a data type conflict that LabVIEW cannot automatically resolve. Fix it, or your code won't run!

The Fundamentals of Data Acquisition (DAQ)

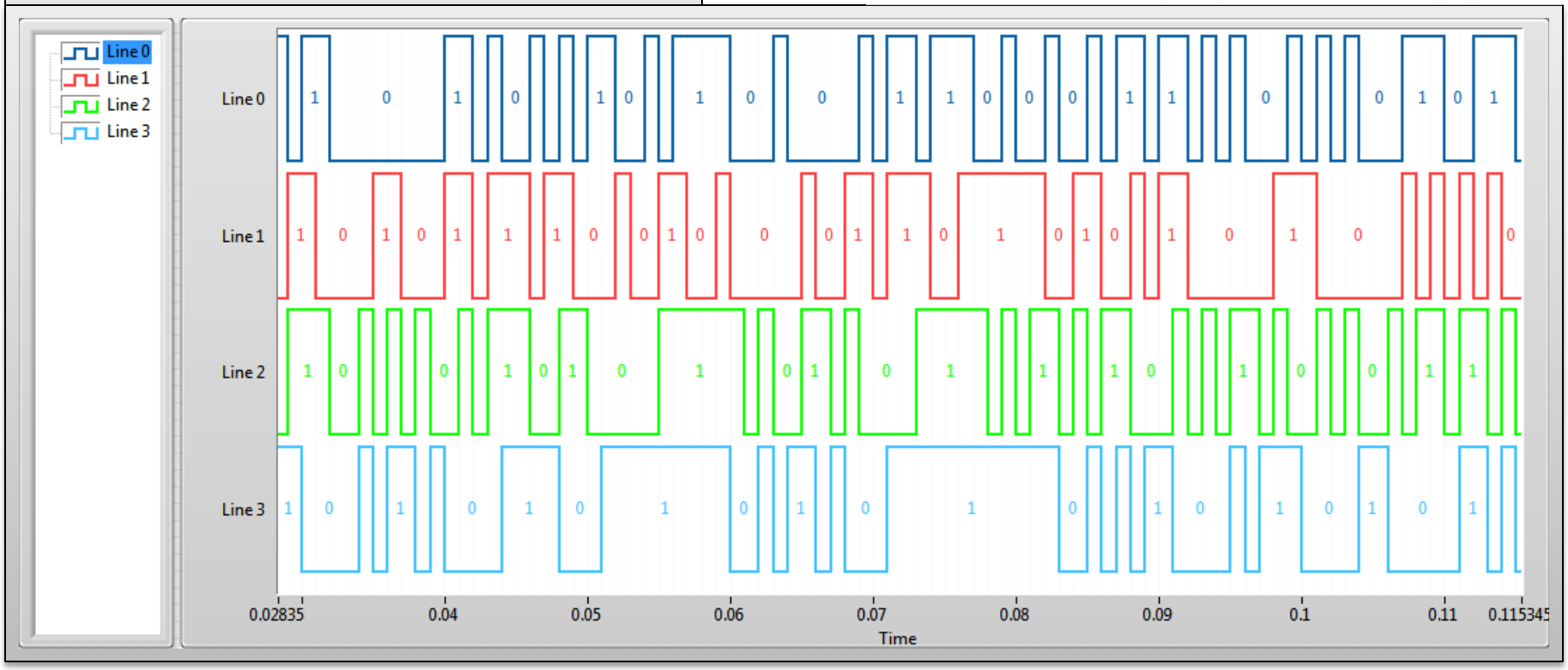
The Basics of Making PC-Based Measurements

NATIONAL INSTRUMENTS

Signals Come in Two Forms: Digital and Analog

Digital

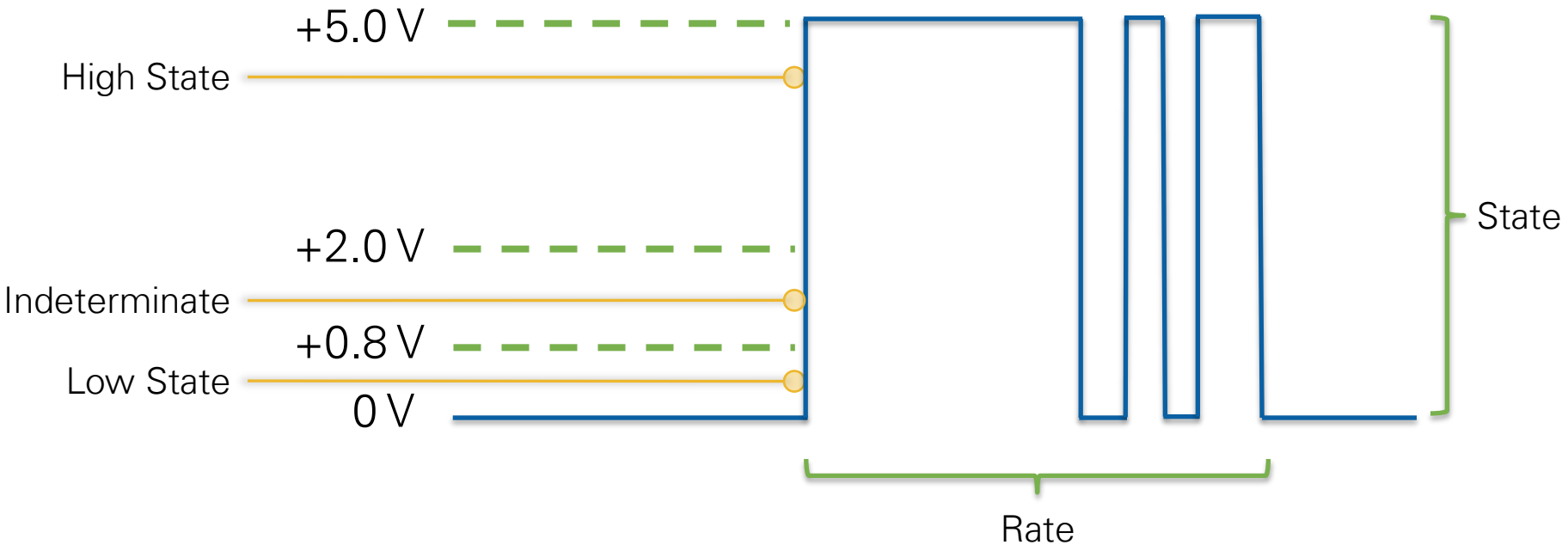
Analog



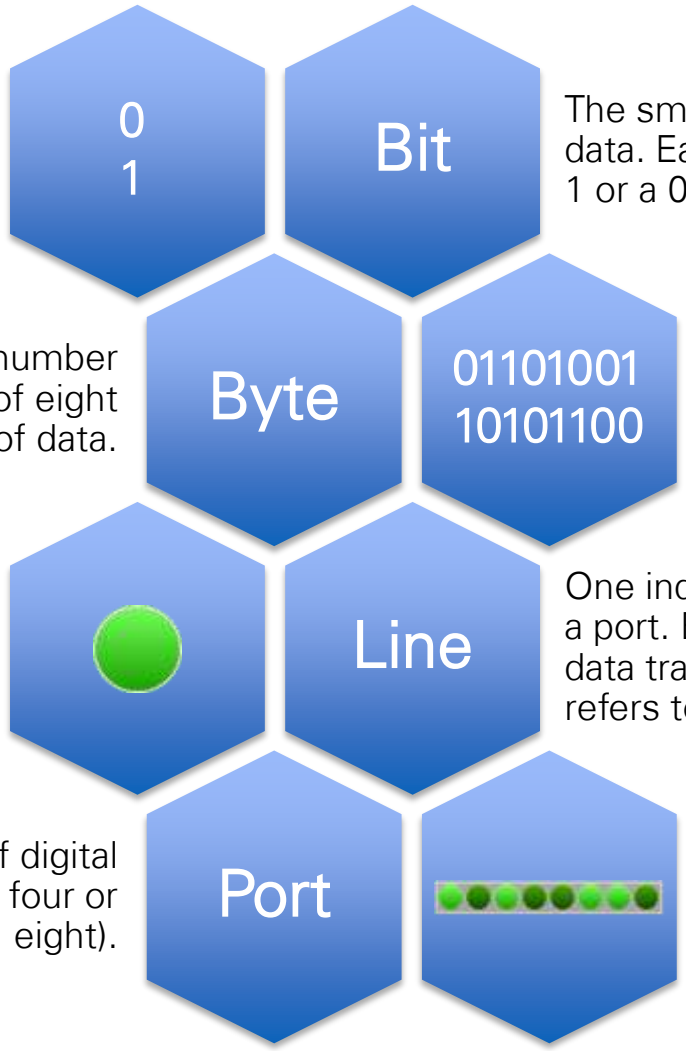
NATIONAL INSTRUMENTS

Digital Signals

- Digital signals have two states: high and low
- Digital lines on a DAQ device accept and generate transistor-transistor logic (TTL) compatible signals



Digital Terminology

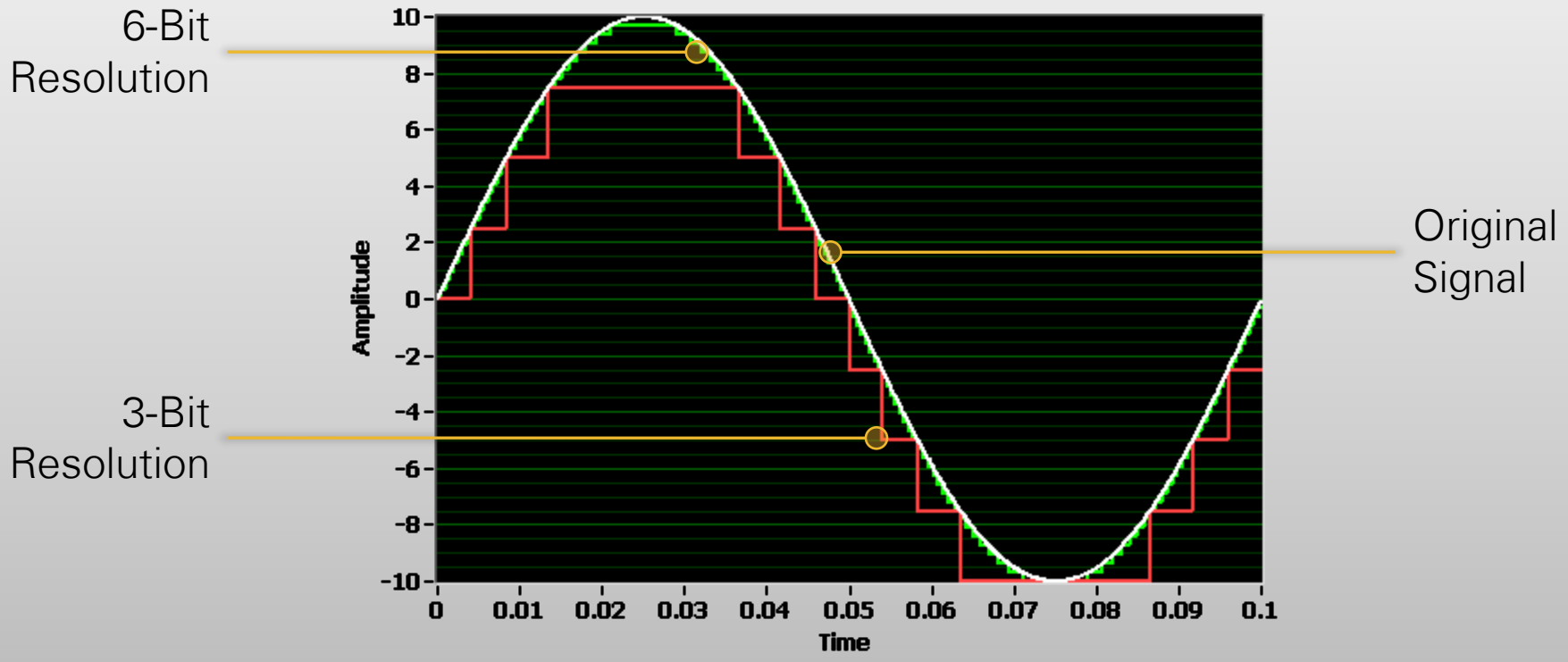


Analog Signals

Analog signals are continuous signals that can be any value with respect to time.

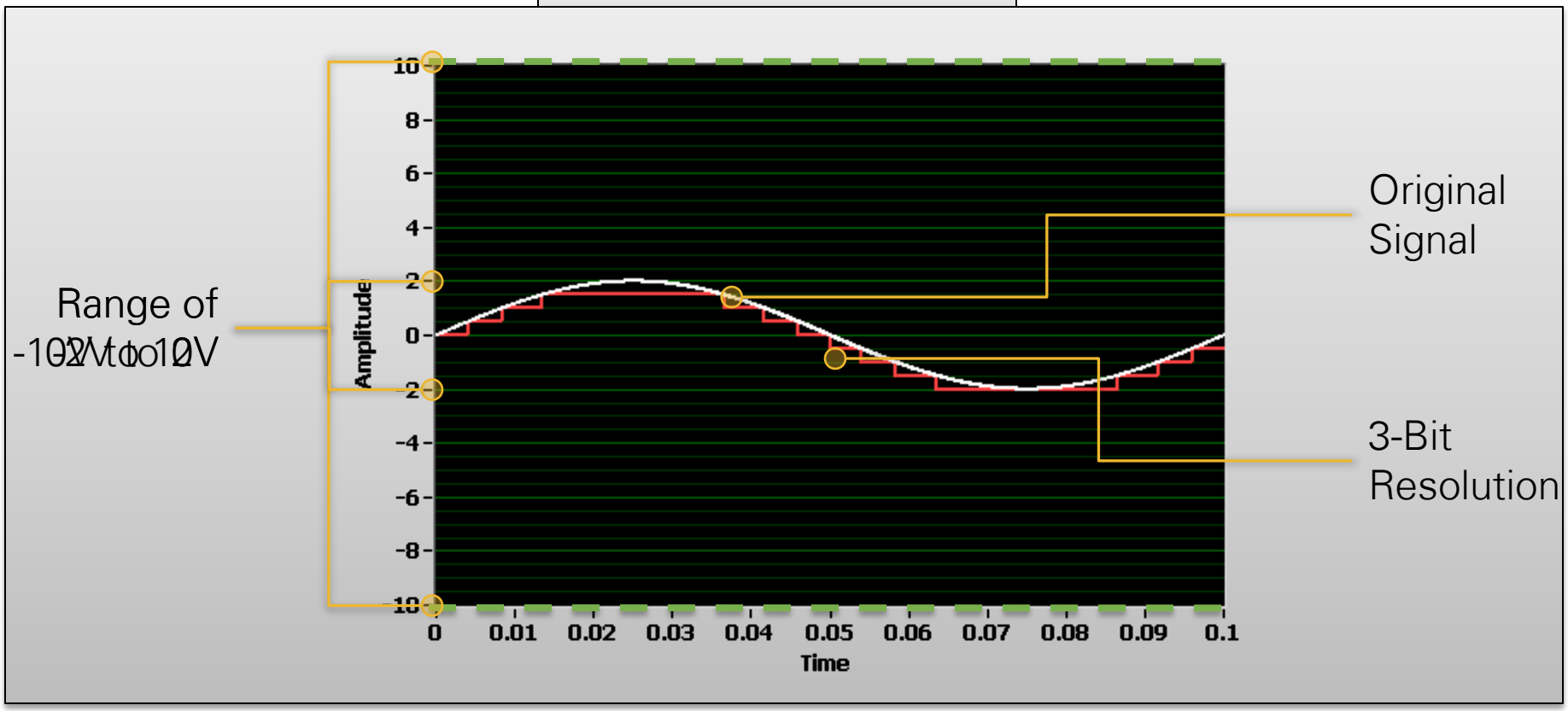
The Three R's of Data Acquisition: Resolution

Resolution | Range | Rate



The Three R's of Data Acquisition: Range

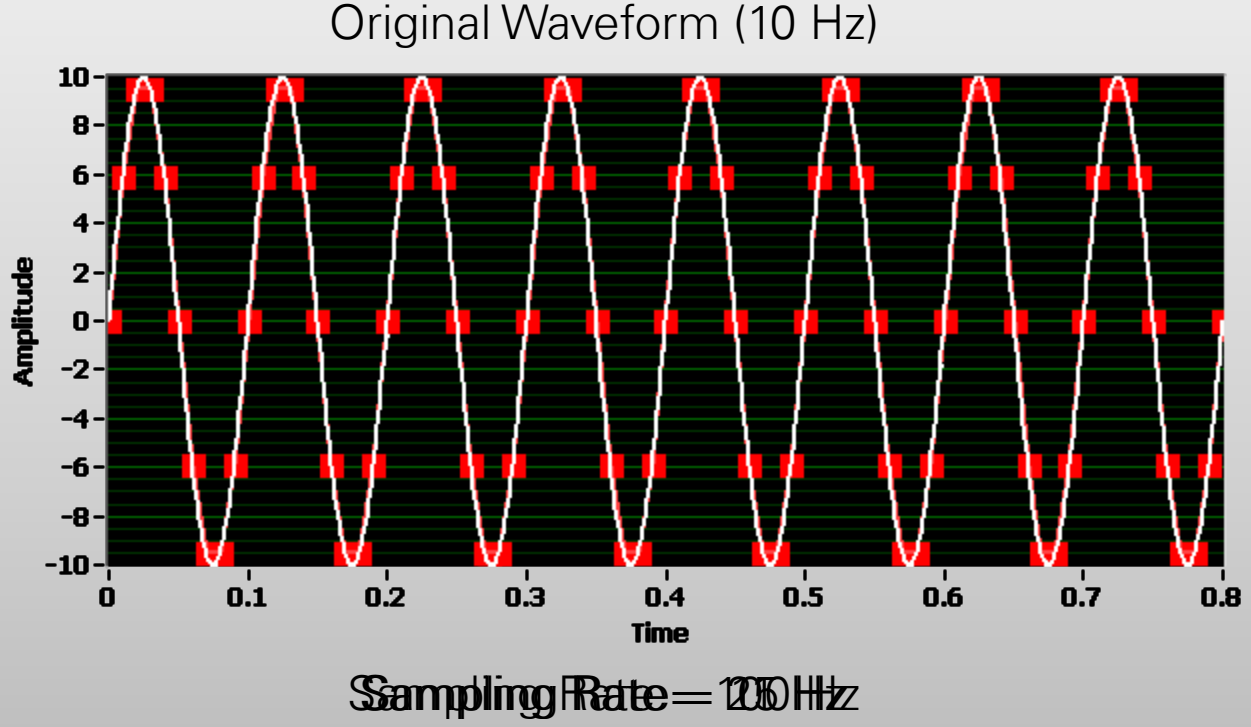
Resolution | **Range** | Rate



NATIONAL INSTRUMENTS

The Three R's of Data Acquisition: Rate

Resolution Range **Rate**



Architecture of an Integrated Measurement System

NI CompactDAQ hardware combines a 1-, 4-, or 8-slot chassis with over 50 measurement-specific NI C Series I/O modules and can operate stand-alone with a built-in controller or connect to a host computer over USB, Ethernet, or 802.11 Wi-Fi.

Sensor

Measurement Device

- Signal Conditioning
- Analog-to-Digital Converter

Software

- Driver Software
- Application Software

NI CompactDAQ Is an Integrated, Modular Solution

Sensors/Signals

- Thermocouple
- Accelerometer
- Strain Gage
- Solar Cell

(etc)

C Series Modules

The NI CompactDAQ Family

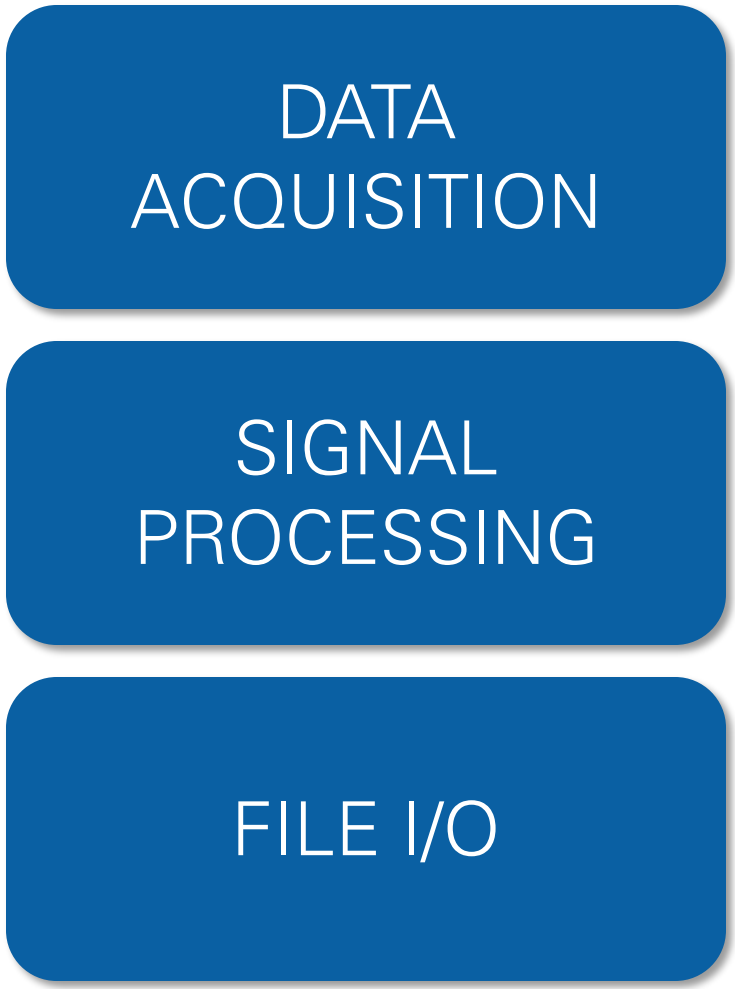
A Custom System for Your Application
Mix and match from the entire family of measurement-specific, auto-detected, hot-swappable C Series modules.

A Module for Any Measurement
Over 50 measurement-specific modules integrate everything you need for a range of signal types, channel counts, and rates.

Same Code, Any Bus
Whether you've chosen to use USB, Ethernet, or Wi-Fi, identical code will run across each bus making scalability simple.

Choose the Right Form Factor for You
Available 1-, 4-, and 8-slot chassis accommodate up to 256 channels per chassis in tethered or stand-alone form.

Datalogger with LabVIEW



National Instruments

Corporate headquarters: *Austin, Texas*

Year established: *1976*

Revenue: $1,1 bilion in 2012

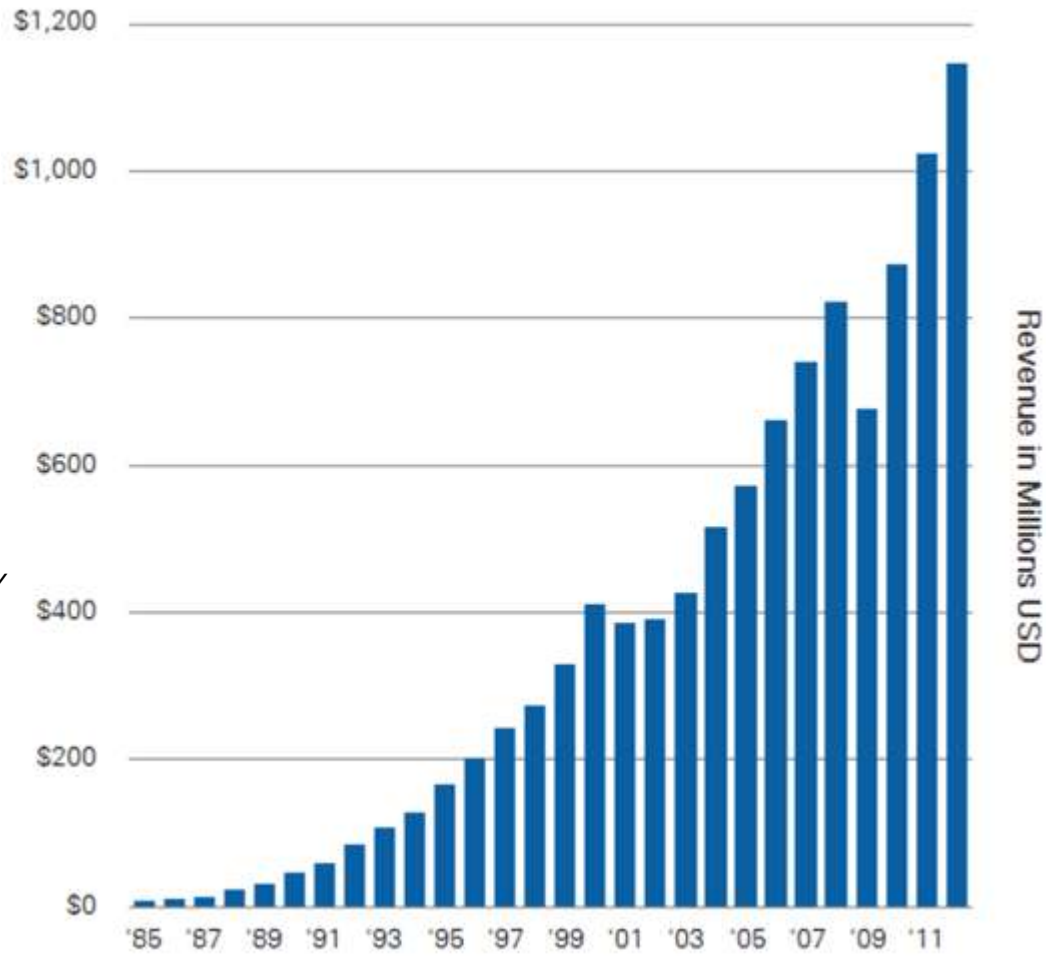
Global operations: *offices in 40 countries*

Investment in R&D: *16% of annual revenue*

Customer base: *30,000 companies annually*

Network: More than 600 Alliance Partners

Diversity: *no industry makes up more than 15% of revenue*

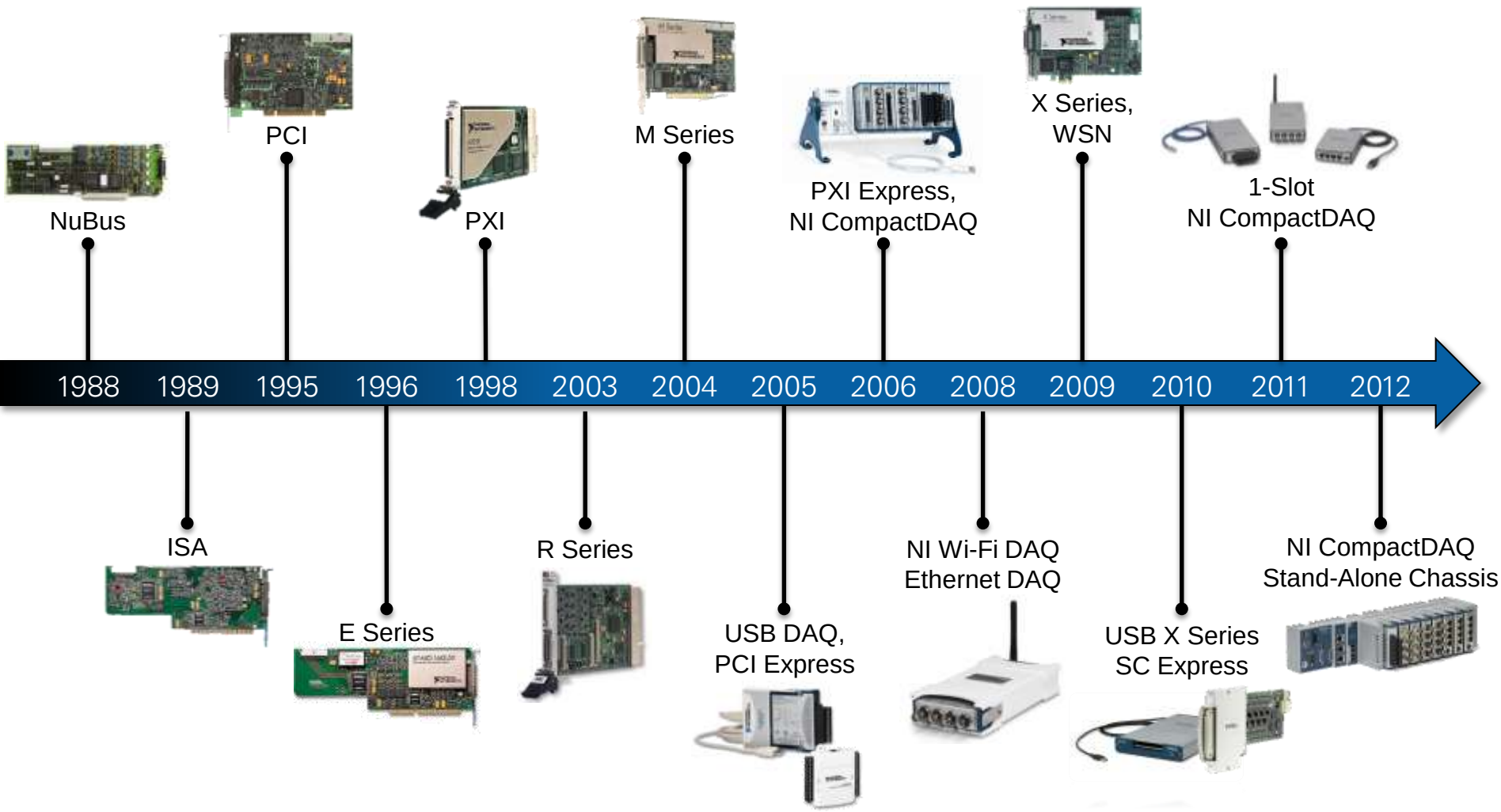


More than 30,000 companies

…including 90% of Fortune 500 manufacturing companies

NI Is the Global Leader in Data Acquisition

With more than 20 years of DAQ hardware history and millions of channels sold



NI Data Acquisition Hardware Families

System

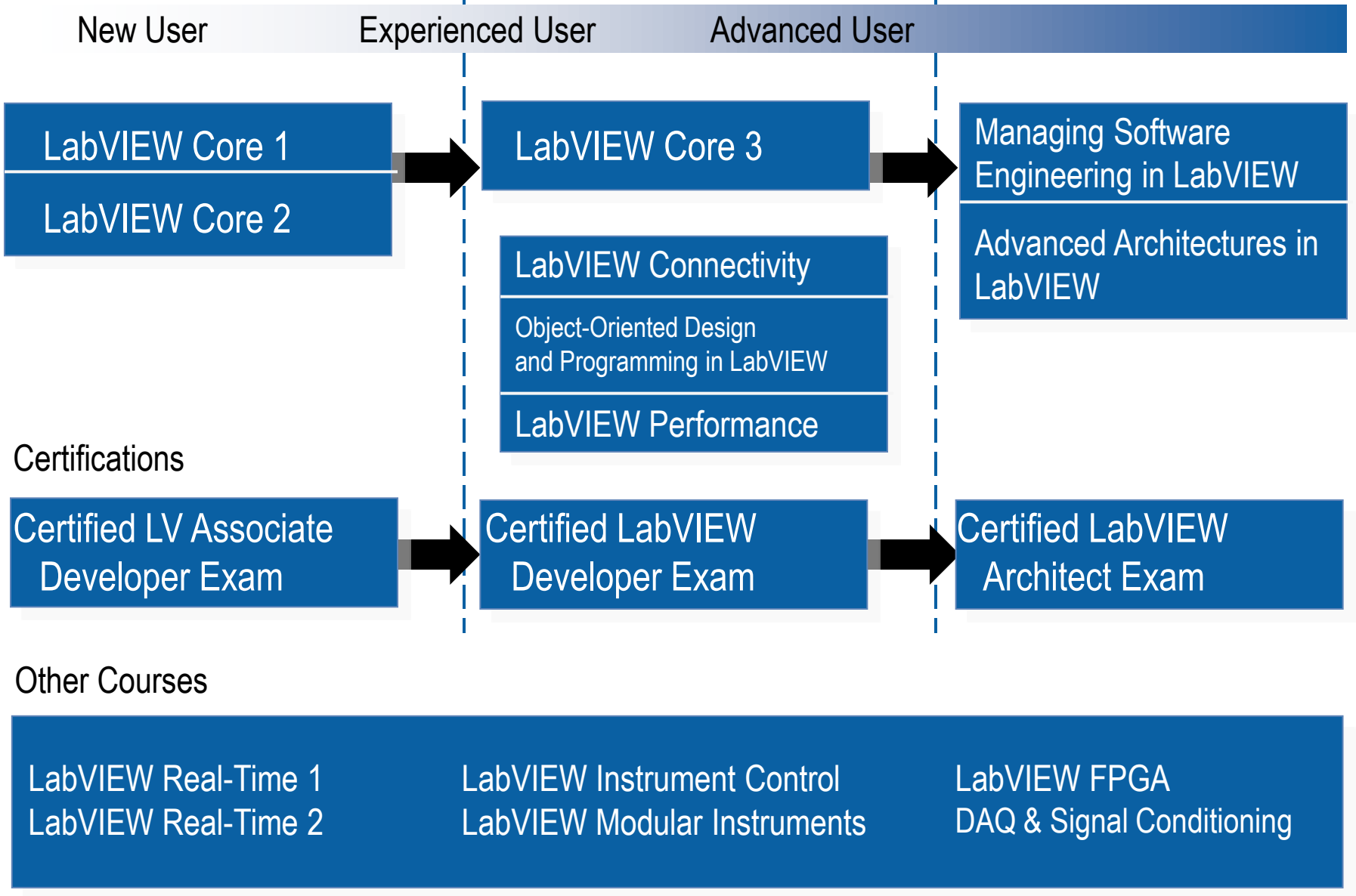
PXI
Optimized for high channel counts and tight synchronization

NI CompactDAQ
Customize with a variety of chassis and module types

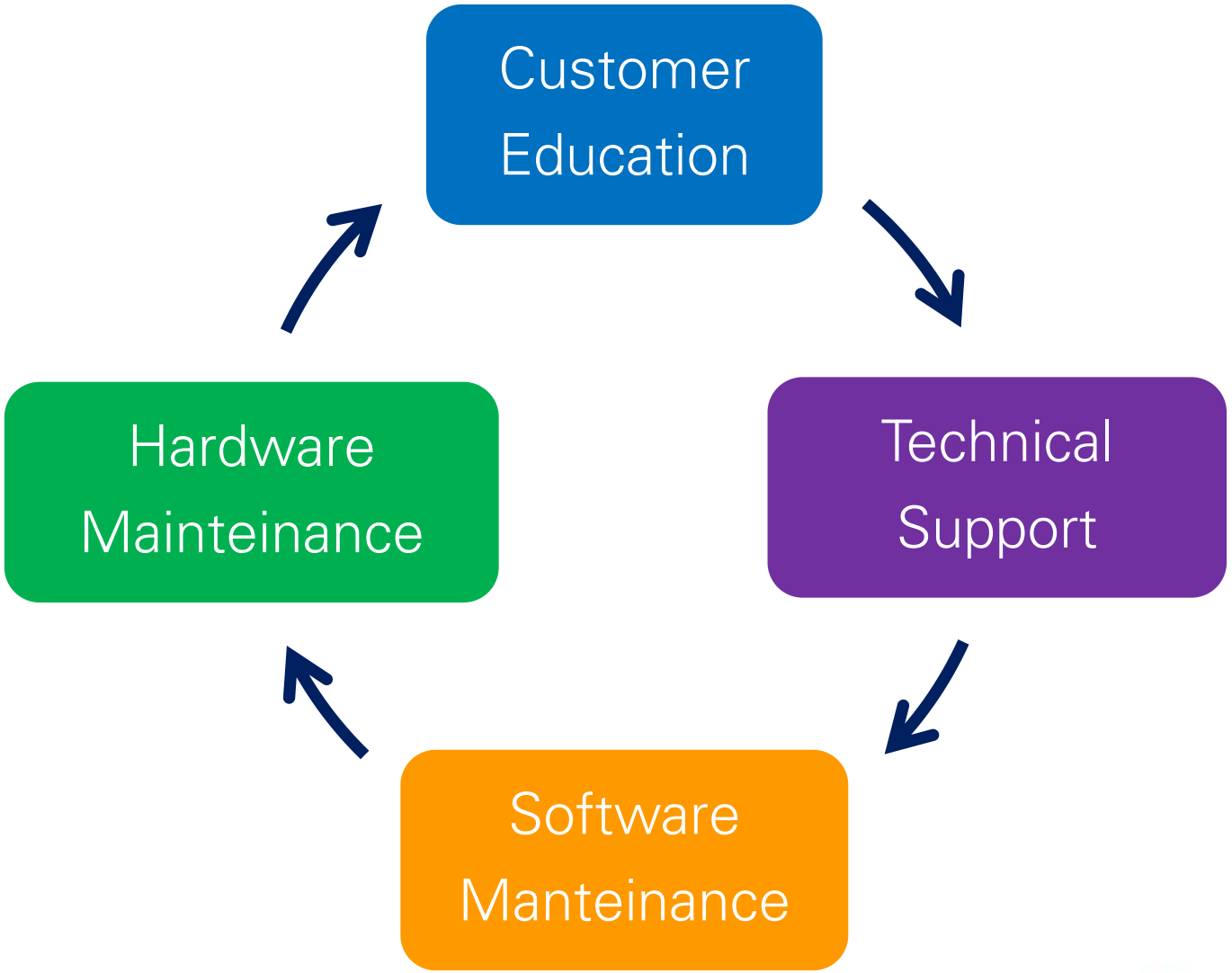
Desktop DAQ
Install in a desktop PC slot for maximum data throughput

Portable DAQ
Easily connect to any laptop or desktop with simple setup

Single Device


New User
Experienced User
Advanced User
LabVIEW Core 1
LabVIEW Core 2
LabVIEW Core 3
Managing Software Engineering in LabVIEW
Advanced Architectures in LabVIEW
LabVIEW Connectivity
Object-Oriented Design and Programming in LabVIEW
LabVIEW Performance
Certifications
Certified LV Associate Developer Exam
Certified LabVIEW Developer Exam
Certified LabVIEW Architect Exam
Other Courses
LabVIEW Real-Time 1
LabVIEW Real-Time 2
LabVIEW Instrument Control
LabVIEW Modular Instruments
LabVIEW FPGA
DAQ & Signal Conditioning

NI Italy Services



Thank you very much!!

QUESTIONS??