Techniques for Efficient Measurement Automation

Goals of Today

1. Learn about the benefits of standardizing on a scalable hardware platform.

2. Know the options for automating measurement hardware.

3. Understand the importance of improving your proficiency with whatever tool you choose.

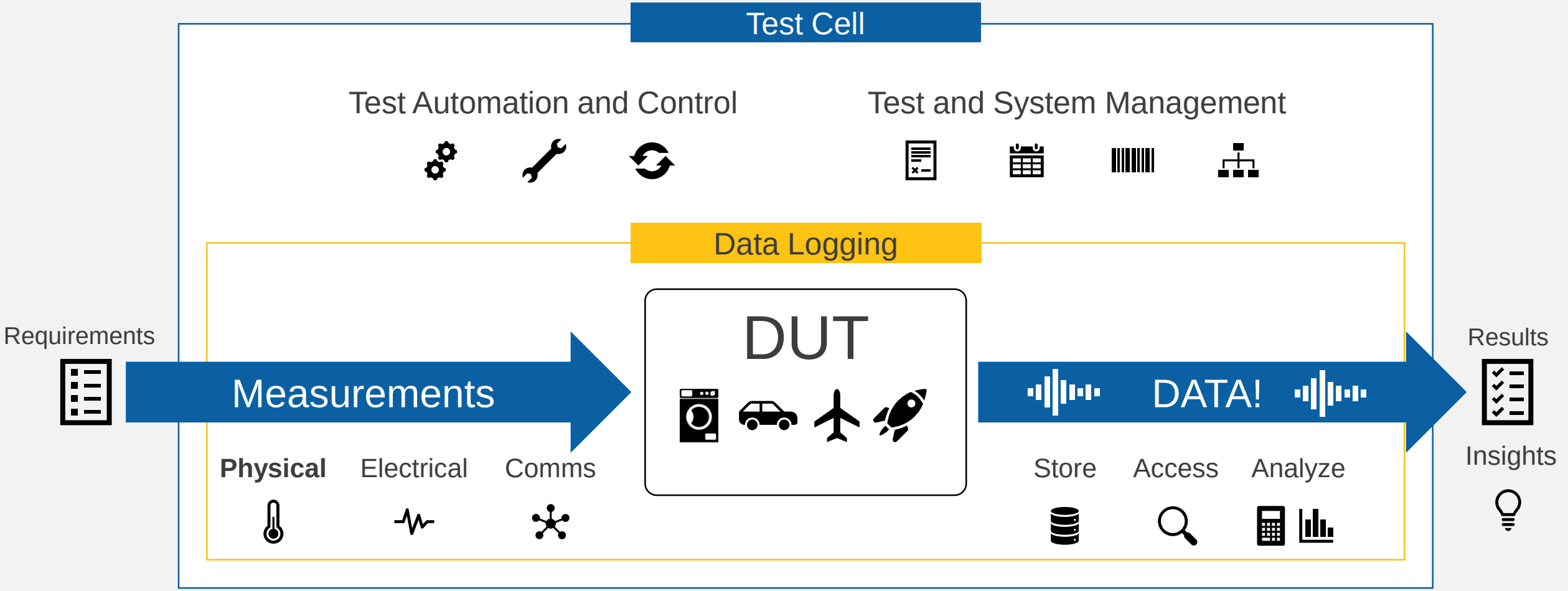
4. Get hands-on with different software

Agenda

- Why automate?
- Challenges with test and measurement automation
- Why is standardisation important?
- NI DAQ platform
- Different ways to work with NI hardware
  - LabVIEW
  - Python
  - C, .NET
  - FlexLogger

Test Labs Are Becoming More Automated

What Do We Mean By Automated Measurements?



NATIONAL INSTRUMENTS

Benefit of Automation For Product Design Teams

- Tests run independently
- Reuse tests
- Can speed up setup with configuration files

Improve Repeatability

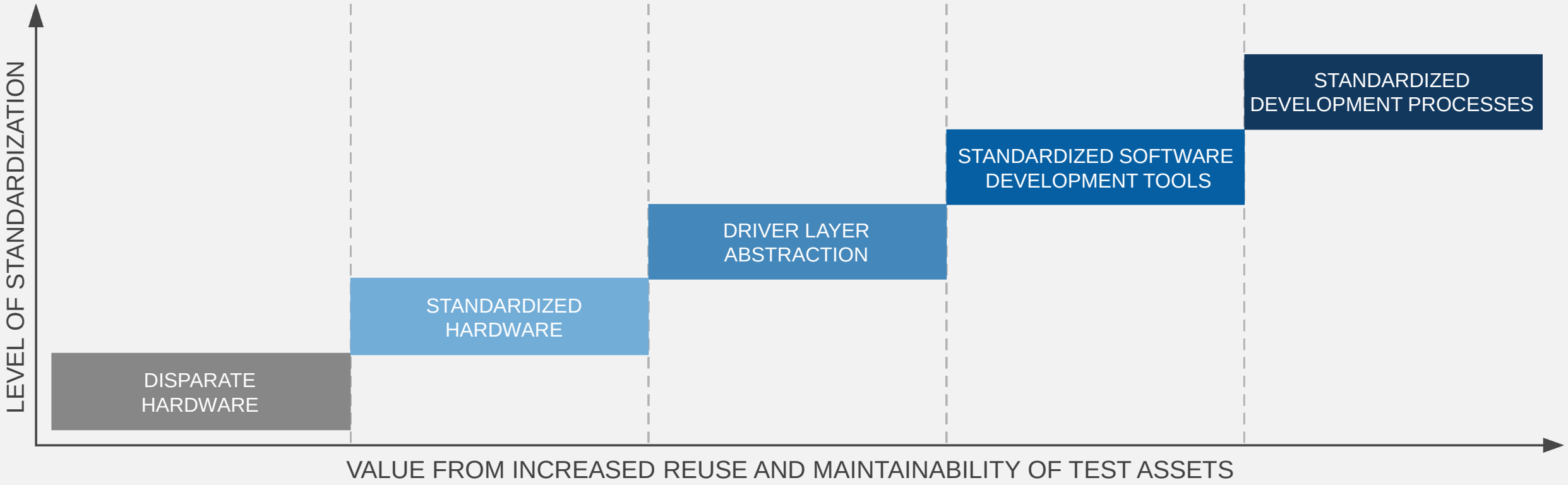
Spend More Time Improving
Device Under Test

NATIONAL INSTRUMENTS

Challenges With Automated Measurement Systems

- Hard to combine mixed-signals from disparate hardware
- Fixturing
- Different data and config file formats
- Duplication of automation code
- Passing work between colleagues
- Programming skills of engineers

Increased Value of Standardization



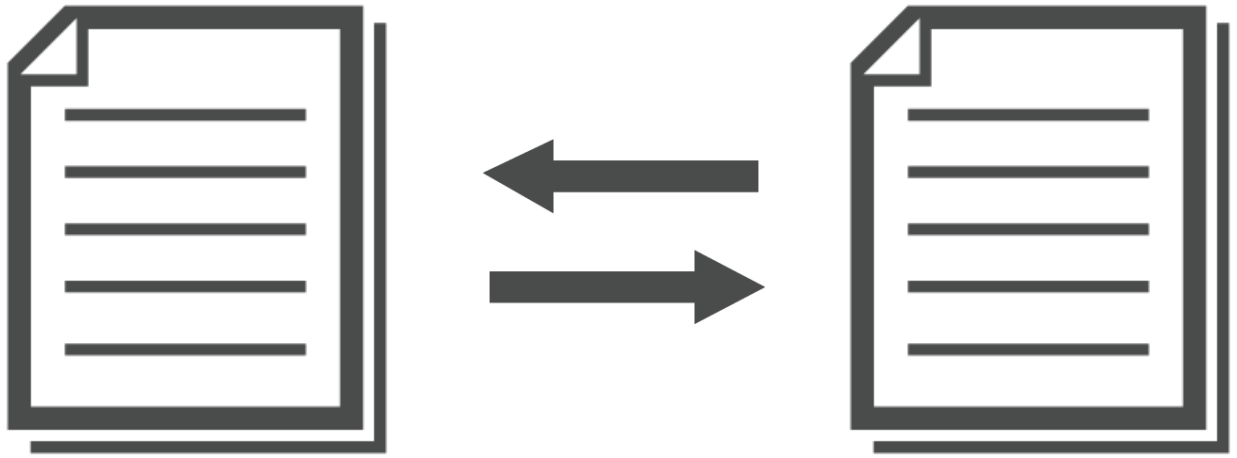
NATIONAL INSTRUMENTS

Benefits of Hardware Standardization

Improve communication between teams with a common language

Benefits of Hardware Standardization

Increase code reuse with shared software libraries



Benefits of Hardware Standardization

Save time and increase repeatability with a standardized device fixture

Improve Repeatability

Common Test Points
For Debugging

Spend More Time Improving
Device Under Test

NATIONAL INSTRUMENTS

Benefits of Hardware Standardization

Reduced Number of
Data File Types

Reduced Training Time

Lower Support &
Maintenance Costs

NATIONAL INSTRUMENTS

Challenges With Automated Measurement Systems

- Hard to combine mixed-signals from disparate hardware
- Fixturing
- Different data and config file formats
- Duplication of automation code
- Passing work between colleagues
- Programming skills of engineers

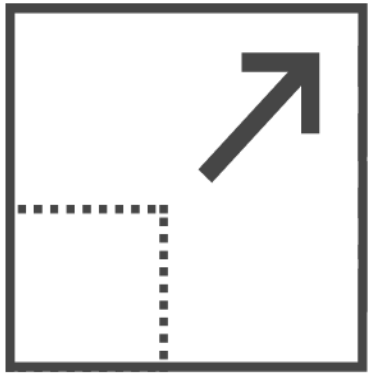
NATIONAL INSTRUMENTS

Standardising on a Modular System

What is a Modular System?

Flexible

User-Defined Software



Scalable

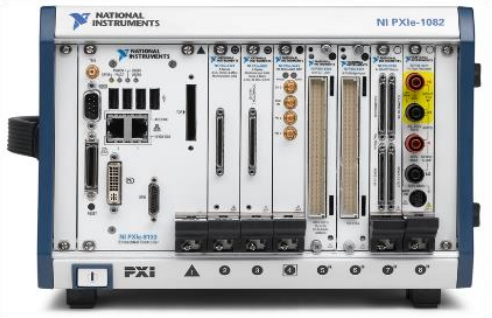
NATIONAL INSTRUMENTS

One Platform to Scale Throughout Product Development

Design			Prototyping				Deployment	
Fundamental Research	Formulation of Technology Concept	Feasibility of Concept & Technologies	Prototype Lab Validation	Prototype Field Test	Pre-Production Product	Pilot Production	Fully Tested, Validated and Qualified	Production & Final Deployment

NATIONAL INSTRUMENTS

NI Modular Test & Measurement Platform



NATIONAL INSTRUMENTS

One Platform to Scale Throughout Product Development

Fundamental Research	Formulation of Technology Concept	Feasibility of Concept & Technologies	Prototype Lab Validation	Prototype Field Test	Pre-Production Product	Pilot Production	Fully Tested, Validated and Qualified	Production & Final Deployment
Design			Prototyping				Deployment	

NATIONAL INSTRUMENTS

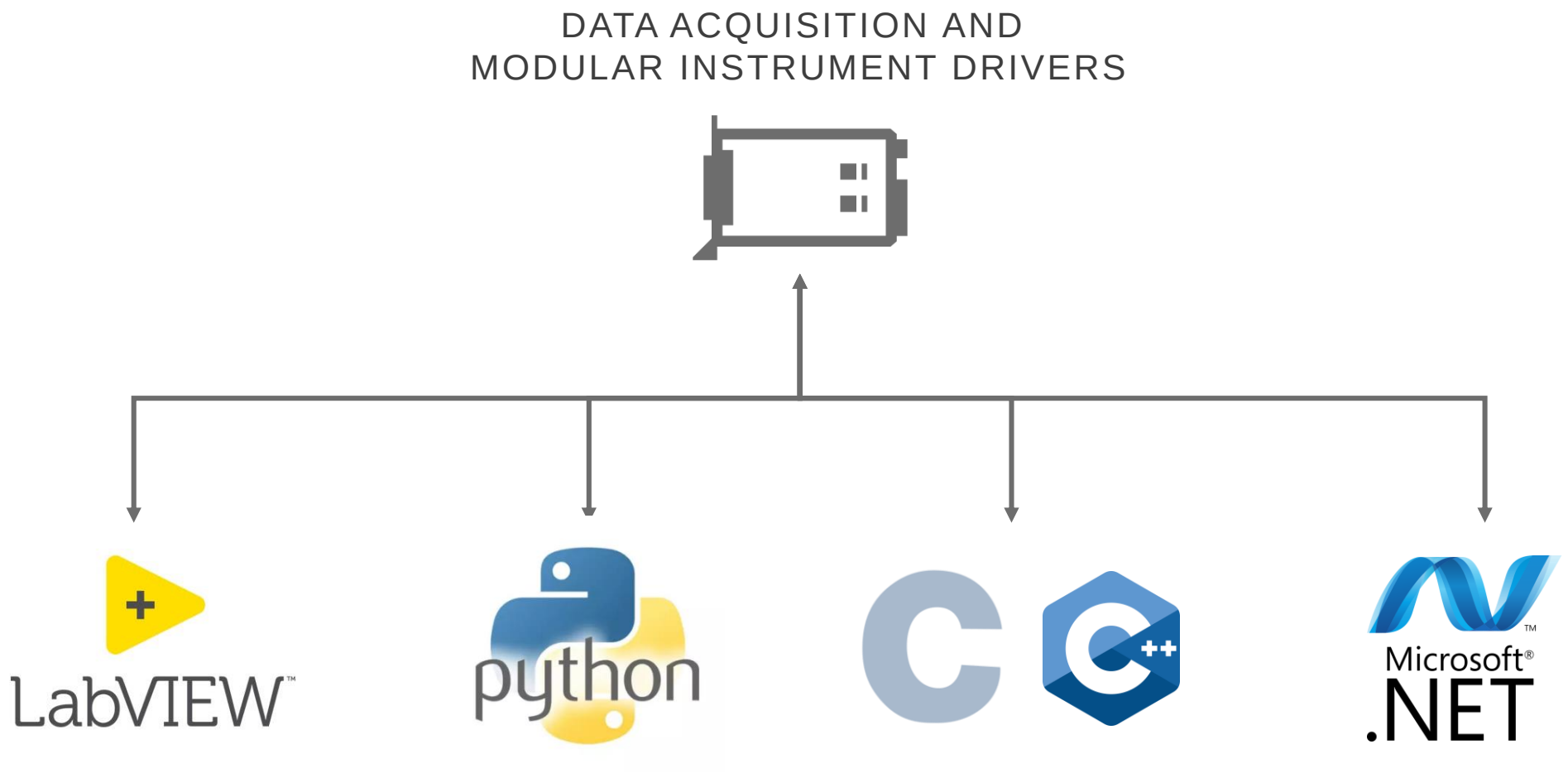
How to Choose the Right Hardware

- Step-by-step guides
- Online configuration advisors
- Speak to Alex Floor

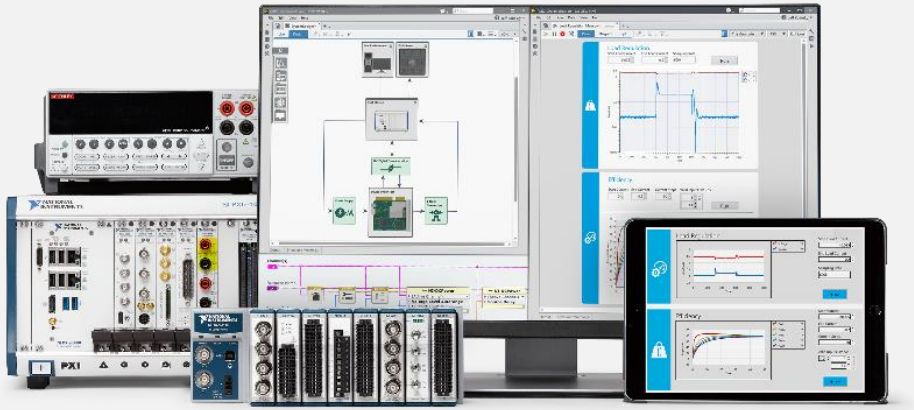
NATIONAL INSTRUMENTS

Different Options for Automating NI Hardware

What Should I Use?



LabVIEW™



LabVIEW NXG

LabVIEW 2018

Two Versions. One Price.

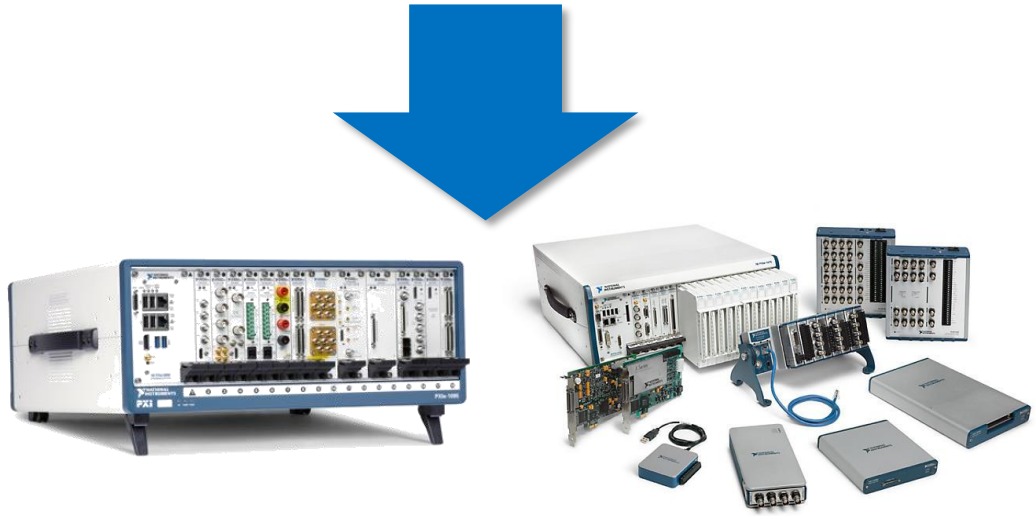
NATIONAL INSTRUMENTS™

Python APIs for NI Hardware

- Automate and control NI hardware using the Python
- Interface with the Python drivers through a concise, object-oriented Python API
- Easy access to properties, device-specific parameters, exception handling, and more
- Get dynamic, real-time device feedback through interactive scripting

```
import nidaqmx


task = nidaqmx.Task()
task.ai_channels.create_voltage_channel('cDAQ1Mod1/ai0')
print task.read(number_of_samples_per_channel=10)
task.clear()
```



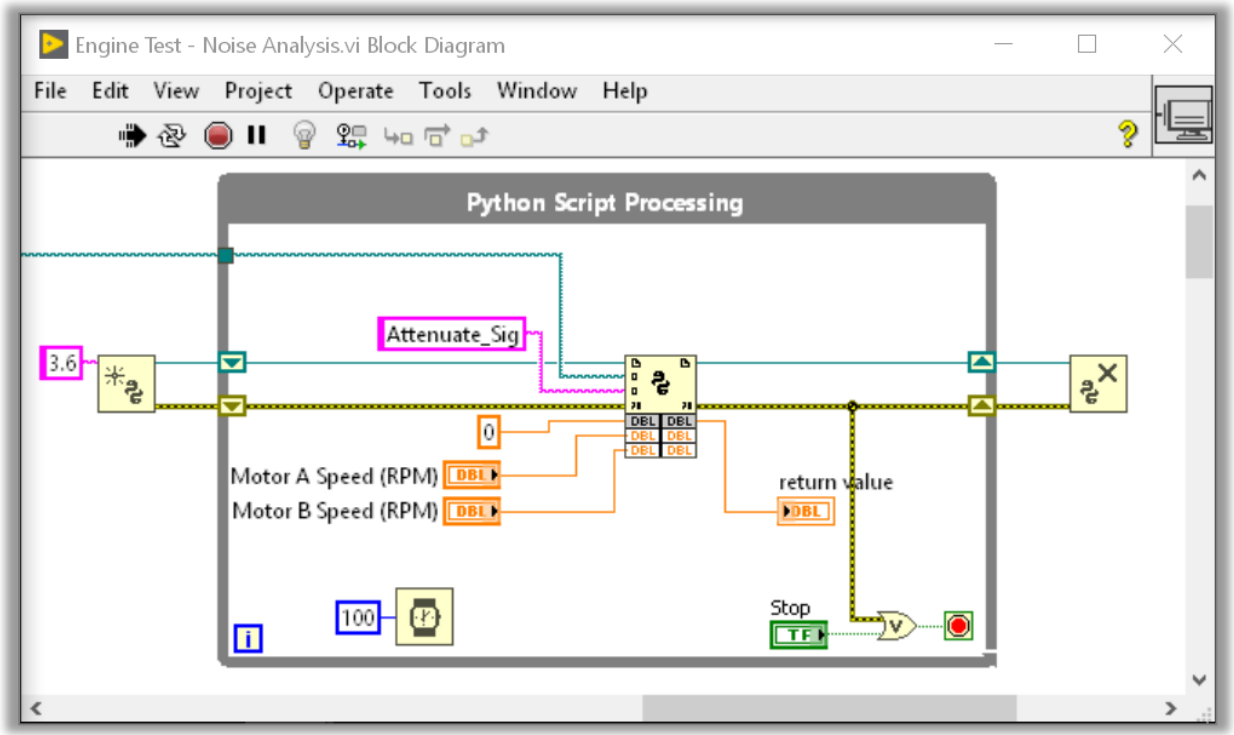
NATIONAL INSTRUMENTS

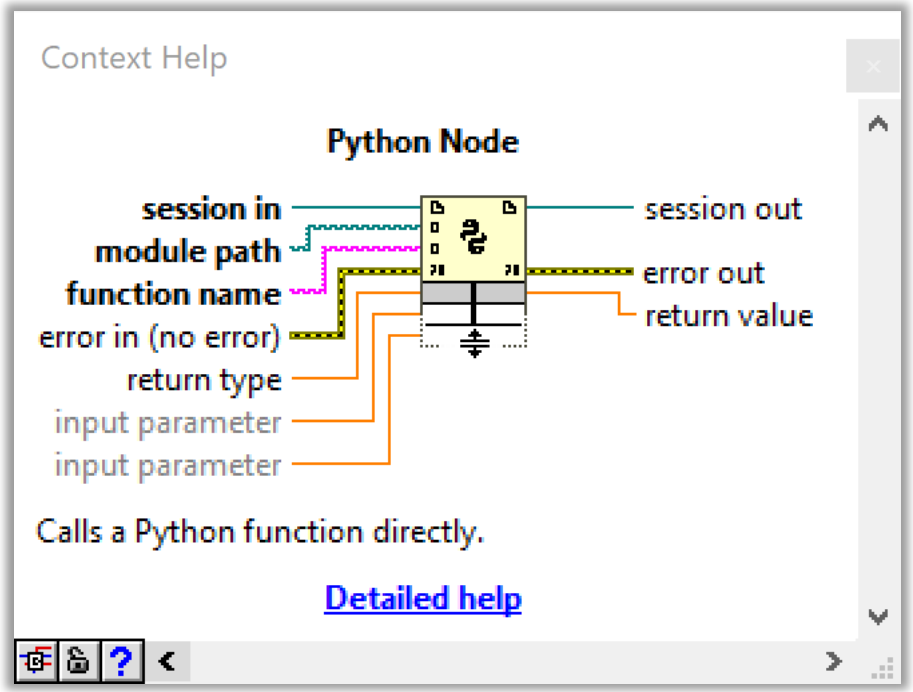
Python APIs for NI Hardware

- NI Maintained APIs (Open Source)
  - NI-VISA (General Instrument Control)
  - NI-DAQmx (Data Acquisition - Analog In/Out and Digital)
  - NI-RIO (User-programmable FPGA Devices)
  - NI-XNET (Automotive Busses)
  - NI-SCOPE (Oscilloscopes and Digitizers)
  - NI-FGEN (Waveform Generators)
  - NI-DMM (Digital Multimeters)
  - NI-SWITCH (Switches)
  - NI-DCPower (Programmable Power Supplies and Source Measure Units)
- Community Developed APIs (Open Source)
  - NI-VISION (Frame Grabbers and Camera Interfaces)
  - NI-VirtualBench (All in one instrument)

Python Node

Natively call Python functions and pass parameters within the LabVIEW environment





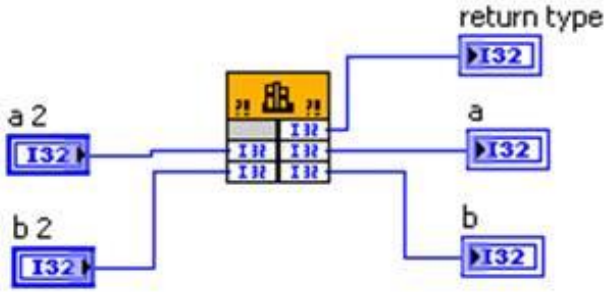
Available in LabVIEW 2018
Coming for LabVIEW NXG soon

NATIONAL INSTRUMENTS

Working with C/C++

- Vast majority of drivers have a C version
- LabWindows/CVI - dedicated test and measurement environment
- Natively call DLLs from LabVIEW

```
#include <stdio.h>
#include <NIDAQmx.h>
#define DAQmxErrChk(functionCall) if( DAQmxFailed(error=(functionCall)) ) goto Error; else
int main(void)
{
	int32		error=0;
	TaskHandle	taskHandle=0;
	int32		read;
	float64		data[1000];
	char		errBuff[2048]={'\0'};
// DAQmx analog voltage channel and timing parameters
  DAQmxErrChk (DAQmxCreateTask("", &taskHandle));
  DAQmxErrChk(DAQmxCreateAIVoltageChan(taskHandle, "Dev1/ai0", "", DAQmx_Val_Cfg_Default, -10.0,
10.0, DAQmx_Val_Volts, NULL));
DAQmxErrChk(DAQmxCfgSampClkTiming(taskHandle, "", 10000.0, DAQmx_Val_Rising,
DAQmx_Val_FiniteSamps, 1000));
// DAQmx Start Code
 DAQmxErrChk(DAQmxStartTask(taskHandle));
// DAQmx Read Code
 DAQmxErrChk(DAQmxReadAnalogF64(taskHandle, 1000, 10.0, DAQmx_Val_GroupByChannel, data, 1000,
&read, NULL));
// Stop and clear task
Error:
	if( DAQmxFailed(error) )
		DAQmxGetExtendedErrorInfo(errBuff,2048);
	if( taskHandle!=0 )  {
		DAQmxStopTask(taskHandle);
		DAQmxClearTask(taskHandle);
	}
	if( DAQmxFailed(error) )
		printf("DAQmx Error: %s\n",errBuff);
		return 0;
}
```

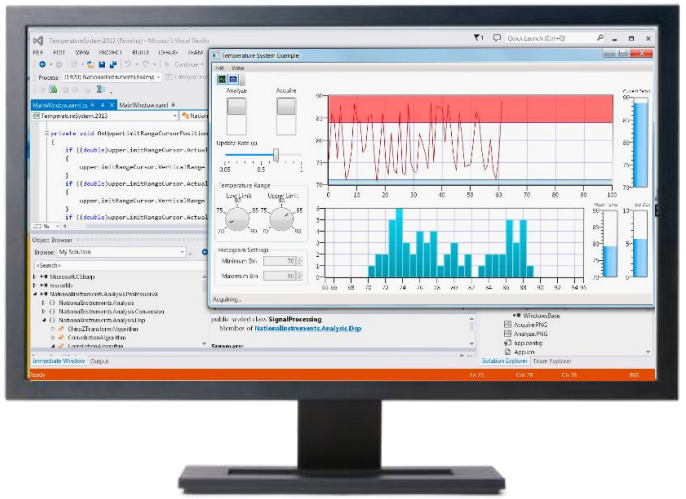


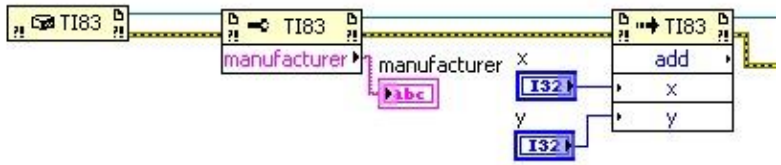
NATIONAL INSTRUMENTS

Working with .NET

- Majority of drivers have a .NET Framework library for VB and C#
- Measurement Studio - dedicated test and measurement add-on for Visual Studio
- Various ways to call .NET from LabVIEW

```
// Create a channel
  myTask.AIChannels.CreateVoltageChannel(physicalChannelComboBox.Text, "",
(AITerminalConfiguration)(-1), rangeMinimum, rangeMaximum, AIVoltageUnits.Volts);
// Configure timing specs
myTask.Timing.ConfigureSampleClock("", sampleRate, SampleClockActiveEdge.Rising,
SampleQuantityMode.FiniteSamples, samplesPerChannel);
// Verify the task
  myTask.Control(TaskAction.Verify);
// Prepare the table for data
InitializeDataTable(myTask.AIChannels, ref dataTable);
acquisitionDataGrid.DataSource = dataTable;
// Read the data
  reader = new AnalogMultiChannelReader(myTask.Stream);
// clear task
  myTask.Dispose();
```





MeasurementStudio™

FlexLogger™

Data Logging Software for Characterization and Durability V&V

Reduce test setup time with configuration-based workflows
Quickly find and share results across your organisation with traceable data

Product Features:

- Wide breadth of I/O with NI DAQ hardware
- Automatically synchronized, distributed measurements
- Comprehensive metadata capture for traceability
- Interactive data review in integrated TDMS Viewer

NATIONAL INSTRUMENTS™

Philips Innovation Services

Getting Hands-On