Electronic Design®

WILLIAM WONG | Technology Editor

What's the Difference Between LabVIEW 2017 and LabVIEW NXG?

Find out which of these NI offerings is the right fit for you.

National Instruments' (NI) LabVIEW got its start way back in 1986. A graphical programming language that that uses a dataflow programming model, it has grown significantly since then. It had been a single platform since its inception, but at NI Week 2017 (see "NIWeek: The Best Electronic Conference" on electronicdesign.com), the company took a divergent path—at least for now—by jointly releasing LabVIEW 2017 and LabVIEW NXG.

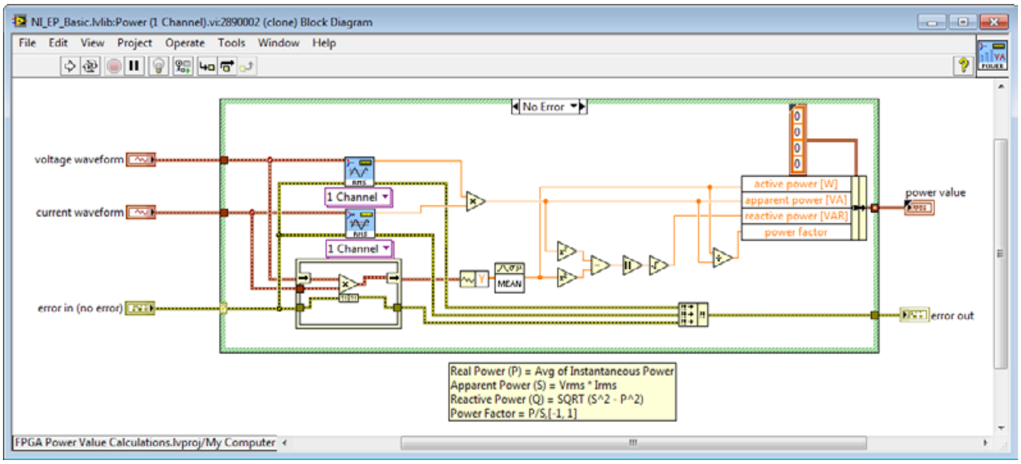
LabVIEW 2017 *(Fig. 1)* is a natural progression of LabVIEW 2016. It contains major user interface (UI) and performance improvements. It maintains upward compatibility with its predecessors and will be the platform of choice for existing developers for existing projects.

LabVIEW NXG *(Fig. 2)* is the next generation of LabVIEW and will eventually wind up leading the charge sometime in the future once it becomes a superset of LabVIEW features. At this point it has a large overlap with LabVIEW 2017, but with a new UI and features designed to improve the user experience.

LabVIEW NXG v1 (or LabVIEW NXG 2017) does not incorporate all of the features of LabVIEW 2017. It was just too difficult to incorporate all the changes that NXG developers came up with into LabVIEW 2017 while maintaining compatibility. LabVIEW NXG will be used by new LabVIEW developers, as well as by existing LabVIEW developers working on new projects. This is a migration path to LabVIEW NXG but developers need to keep in mind that some features are not available in LabVIEW NXG, at this point. The long-term goal is to make this migration totally seamless. At that point, LabVIEW NXG will become the only incarnation of LabVIEW. Migration of projects from LabVIEW NXG to LabVIEW 2017 is not supported.

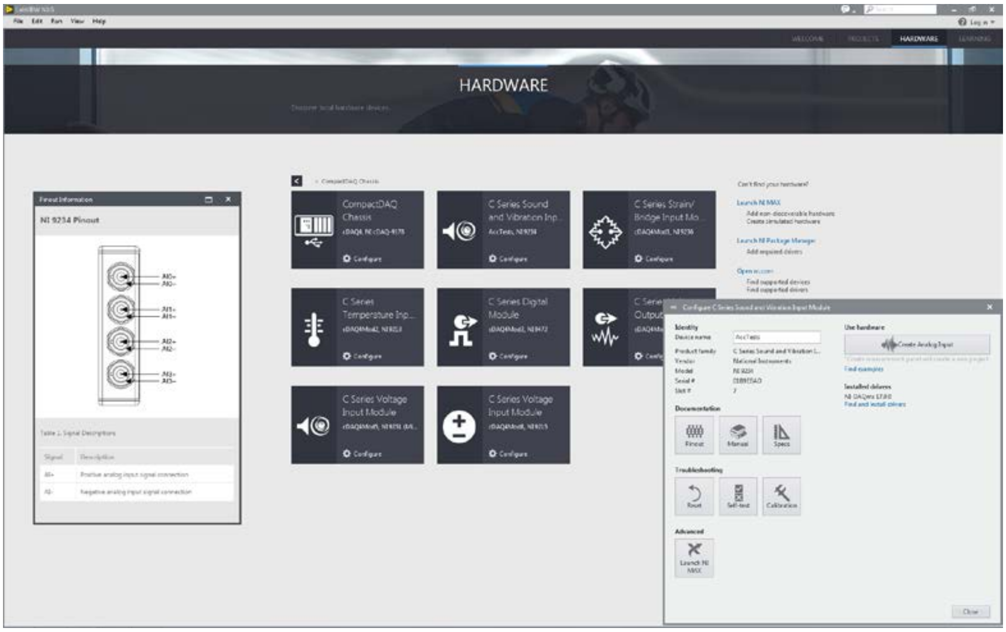
The merging of LabVIEW back to a single platform will probably not occur next year, though it is a priority. In the meantime, look for more frequent updates of LabVIEW NXG, as well as the incorporation of even more new features. Some features will be rolled back into LabVIEW 2017 and its subsequent versions until the platforms reunite.

One of the main reasons for the creation of LabVIEW NXG was to improve the user experience, especially for new users and new projects. The creation of new projects for NI's CompactRIO *(Fig. 3)* highlights this approach. LabVIEW NXG



Real Power (P) = Avg of Instantaneous Power
Apparent Power (S) = Vrms * Irms
Reactive Power (Q) = SQRT (S^2 - P^2)
Power Factor = P/S,[-1, 1]

1. LabVIEW 2017 moves the previous incarnation of LabVIEW forward with new features and improved performance.



2. LabVIEW NXG is the next generation of LabVIEW with a new, improved interface and some new features not found in LabVIEW 2017 designed to improve the user experience, like automatic identification and support for hardware.

will automatically recognize a CompactRIO's configuration, including the various installed modules.

LabVIEW NXG will automatically download and install and needed drivers. In addition, each module can be immediately configured from the LabVIEW NXG user interface . Essentially LabVIEW NXG creates a small program or virtual instrument (VI) for each device, such as an analog-to-digital converter (ADC). A LabVIEW VI can have any number of configuration parameters that can be adjusted immediately. A configured VI can then be copied and pasted into the project, where it can be wired together with other VIs.

The ability to quickly configure a system will save on start-up time, as well as make it simpler for new LabVIEW users to get started. It allows a system to be tested without doing any LabVIEW programming, although that is the long-term intent of a project. This approach also lets a developer know that the hardware configuration matches their expectations and that it is operational.

LabVIEW is typically used in test-and-measurement applications that incorporate hardware—often hardware like NI's CompactRIO—but it is not limited to NI hardware, or any other hardware for that matter. LabVIEW can be used to create software applications that are not controlling hardware other than that needed to run the application and present a user interface.

3. LabVIEW NXG can automatically recognize and support a CompactRIO configuration, making project creation and configuration significantly easier.

LabVIEW's target hardware is typically PC-based or its equivalent like CompactRIO but it can also target embedded hardware like microcontrollers and even FPGAs. In fact, CompactRIO is built around an FPGA. A LabVIEW application can split between the underlying hardware, such as an x86 processor and an FPGA, but this is done as part of project development. This is one area where LabVIEW 2017 exceeds the feature list of the current version of LabVIEW NXG.

The graphical G programming language and matching G compiler is common to both LabVIEW 2017 and LabVIEW NXG. This will make the eventual merging of the two development platforms easier in the future. NI also uses the open-source LLVM compiler as the backend for G.

NI is delivering LabVIEW 2017 and LabVIEW NXG in the same package, so the cost of choosing one platform over the other comes down to which features are desired by the developers. Developers may choose LabVIEW 2017 for supporting their existing projects or migrate some to LabVIEW NXG.

This dichotomy will remain in effect for a few years as the feature set of LabVIEW NXG increases and exclusive LabVIEW NXG features are eventually merged into the mainline version of LabVIEW. Both will be maintained and supported by NI, so LabVIEW NXG should be viewed as an alternate production development platform rather than an experimental instance.