

Lab View: GateWay To Graphical Programming

by Elaine Vanessa - Sunday, February 23, 2020

<https://gonitsora.com/lab-view-gateway-to-graphical-programming/>

Are you tired of using traditional text-based programming languages like [Python](#), Java, C, or C++? Consider trying LabVIEW if you haven't.

LabVIEW is an acronym for Laboratory Virtual Instrument Engineering Workbench, a graphical-based programming language that has taken the industry by storm. What makes the most-hyped LabVIEW different from other languages is its general-purpose graphical programming language (G) and friendly user interface. In the LabVIEW environment, programmers can create amazing programs using tools, ready-built templates, debugging features, and simple 'drag and drop' techniques.

It merely means that you can play around different elements while joining them, which give a perfect data flow structure.

Why LabVIEW?

If you are a programmer looking for versatile software to analyze and measure data with automation, then LabVIEW is for you. It is easy-going and recommended by most experts simply because its Graphical User Interface (GUI) is so easy to program with.

By using one of the most powerful visual programming languages that most LabVIEW programmers lovingly call 'G', you can multiply your productivity. With LabVIEW, You can complete programs in hours that usually require weeks or months to be written in common programming languages.

It will make your life hassle-free as possible. It has enormous libraries that help you to go with the routine tasks, without worrying about memory allocation and pointers.

How Does LabVIEW 'G' Programming Will Benefit You?

- **Everything Graphical**

The major strength of LabVIEW comes from its responsive graphical interface and simplicity that makes it stand out in the crowd. The G programming is done by wiring different graphical icons on a map and compiled to machine code for computer processing. The interesting thing is that while G contains almost similar concepts found in most of the traditional languages, everything represents graphically instead of a script version. This increases the flexibility empowering scientists and programmers to learn it quickly.

Fewer Details

Being a visual language, LabVIEW is free from all the syntactical details associated with text programming languages. You are finally free from the trouble of remembering where to put curly braces and semicolons. If you are not aware of how text-based programming languages work. Relax! You don't need to know now. With LabVIEW this need has been eliminated successfully.

- **An Intuitive Graphical Programming Language**

Commonly, many people learn quickly by seeing images. Most programmers are also characterized as 'visual thinkers' and accustomed at visual processing of information. Graphical programming with G is much more fun and easy than most [general-purpose programming languages](#) because they require you to spend huge time learning the text-based syntax linked with the language.

LabVIEW makes the lives of scientists and engineers much easier as they are habitual visualizers and can even imagine the diagrammatical model in their mind. In real life, the graphical dataflow and block diagram approach represent the data in its most natural form and map user interface controls data extremely intuitively. So, you can easily view and modify the data input.

- **Combinability With Other Languages**

Even though G-code offers an excellent opportunity for parallel programming and eliminates the need of developers to understand and organize computer memory, this is not the ideal case in every situation. Mainly, mathematical equations can be neatly expressed with text. For this purpose, you can combine graphical programming with other text-based programming approaches for advanced numerical analysis and mathematical processes.

For example, the MathScript Node in LabVIEW allows users to insert the textual mathematical expressions and algorithms into VI. After this step, you have to use the LabVIEW graphical environment to equip the scripts with slides, buttons, and knobs, and graphics. In layman's terms, you can choose a textual approach, a graphical approach, or a mixture of two. The choice is yours.

- **Interactive Debugging Tools**

Since the G-code is easy to understand, LabVIEW is inherent with some [debugging techniques](#) that you can use to fix errors in the programs interactively without a slight delay. It comes with unique debugging tools such as correcting broken virtual instruments (Vis), identifying common reasons behind broken VIs, and fixing irregular behaviour. You can also have a probe tool to check intermediate values on a wire as a VI runs.

- **Automatic Parallelism and Performance**

Like most graphical programming languages, LabVIEW also comes with the option of automatic parallelization. Contrary to sequential programming languages like C and C++, visual programs contain built-in information about which fragment of the code should be executed in parallel. For example, both consumer and producer loops operate in parallel. However, most programmers perceive LabVIEW like any other sequential program.

G-code design can offer the greatest parallel execution in its most simple form. This nature of LabVIEW has profited the test and measurement state because the data is created and consumed simultaneously at the same time.

- **Improved Problem Solving**

If you are interested in pursuing your career in the science field, such as a [heuristic programmer](#), and solving complicated problems is part of your profession, then you must learn LabVIEW. Surprisingly, LabVIEW applications are commonly seen in almost all industries from engineering to farming, and chemistry to teaching and everything in between. As compared to the traditional lower-level alternatives, this visual programming language provides us with improved options to solve problems.

- **Coding Made Easy**

Unlike languages that use statements and lengthy codes, the repetitive actions are controlled in graphical languages like LabVIEW. For instance, a loop will be created following a combination of statements multiple numbers of times. Generally, a loop in C++ looks like a combination of text, symbols, and alphanumeric digits.

However, in LabVIEW, a loop is created and all the elements are placed at their respective positions. The code embedded within the loop is executed only for a specific number of times. Still confused? [Australian Master](#) provides helpful content to students and programmers helping them to learn coding faster.

LabVIEW Does Have Some Cons Too

- **Lack of Standardization**

Using a product that is not standardized by the entire industry is quite a risky task for most companies.

- **High Cost of Ownership**

Cost of ownership should be an essential consideration when navigating build or buy decisions. For

industrialized projects with highly optimized specifications, most commercial off-the-shelf vendors fail to meet the requirements.

- **Difficult for Text Programmers**

If you are an experienced text-based programmer and have worked a lot with popular text-based languages like then, you will need some familiarization time to get used to LabVIEW, which may result in productivity loss.

- **Lack of Portability**

The only thing that makes LabVIEW not everyone's favorite is portability issues. It has been noted that LabVIEW is not at all portable to iOS and Android platforms. Hence, it is not the best choice for the development of Single Page Application (SPA).

Conclusion

Considering the above-mentioned discussion and analysis, it has been concluded that LabVIEW's graphical data flow programming approach provides programmers a simple-to-understand and implement a way to solve complex problems. With all the options to write, read, measure, and control, LabVIEW proves to be a flexible tool that makes a programmer's life more comfortable day by day. The intuitive graphical coding and the ability to combine other languages are some of the most exciting features of LabVIEW that you cannot find in its counterpart text-based programming languages.

PDF generated from <https://gonitsora.com/lab-view-gateway-to-graphical-programming/>.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.