

HOW NO-CODE DESIGN COULD ACCELERATE INDUSTRIAL IOT DEVELOPMENT

By Philip Ling | January 14, 2022

No-code and low-code tools are the latest development in automatic code generation. The advantage of automatic code generation became apparent after the first program was hand-written. It makes a lot of sense to automate anything that follows well-defined rules. The syntax and semantics of a programming language are good examples of rules. Follow the rules and you end up with executable code.

This is the theory behind a no-code or low-code approach to software development. Raise the abstraction layer and hide the complexity using a graphical interface. Web developers harnessed this with the “what you see is what you get” (WYSIWYG) revolution, making it faster and simpler to create websites.

Those rules also mean that automation must have a restricted view of the world by necessity. Limiting the possible actions is the only way automation can work. Limited flexibility was a characteristic of early WYSIWYG web tools.

This can also lead to code that compiles but is inelegant. However, the productivity gains that come from auto-code generation should offset the lack of fine-grained control over the code produced. For many hardened developers, auto-code generation is only used under the right conditions. Often, at an embedded level where code size and/or performance can be critical, engineers will still write most of the code, or at least optimize it by hand.

Low-code platforms are already used for developing simple desktop or mobile applications. In the enterprise domain, the use of low-code or even no-code platforms is gaining popularity. It allows anyone with a need and an idea to create an application that runs either in the cloud or on a desktop computer. This makes it much easier for non-software engineers to act on their inspiration and develop solutions that are stable enough for customers to use.

These enterprise-focused solutions use a drag-and-drop approach to design. Construction of complex functions and processes uses predefined building blocks. This isn't unlike using scripting languages to automate other desktop applications, but the implementation is graphical rather textual.

What is important here is that the hardware platform is generally uniform and well defined. Mobile phones use one of two dominant operating systems, desktops and servers tend to follow the same pattern. Knowing what the underlying platform can and can't do makes a no-code approach effective. But can it transfer to the embedded domain, where the underlying platforms are much more varied?

No-code hardware in the Industrial IoT

In the IoT, the idea of no-code configuration is also beginning to take hold. In this case, the need is less about enabling non-engineers and more about tackling the size of the challenge involved with developing a total IoT solution. Connecting one sensor to the internet may not be too difficult, but developing a complex, connected system involves more than point-to-point connectivity. Deploying and commissioning multiple sensors, actuators and control elements across an entire organization becomes a huge task. If every one of those devices also requires low-level design and configuration it increases the potential for error, as well as increasing the time and effort required.

Moving to a no-code strategy can increase productivity and reduce design complexity. It is also easier to verify the functionality of your design before deploying it using a no-code approach, because the tool has an implicit understanding of the functionality. This makes it easier to check the results and verify the implementation. Testing low-level software, even with hardware in the loop, is not so straightforward because the intent is not documented in a machine-readable way.

But as all engineers know, embedded systems are not like an enterprise platform, or even a conventional computer. Each one will have low-level differences. Any tool that hopes to automate embedded software generation must know and understand this.

Despite this obvious challenge, there are solutions. One is to focus on standard hardware, such as the development kits and evaluation modules provided by semiconductor manufacturers to support their components. This does require a lot of up-front effort from the code creation tool provider, meaning commercial restrictions will limit the number of boards that can be supported. This is particularly true if the code generation platform is open source or free to use, which is the case for many of the early examples.

A variant of this approach is to focus solely on one type of hardware. In the embedded domain, Arduino is one of the best known and most adopted platforms. There are several examples of low-code/no-code solutions that target the Arduino platform.



A no-code approach to IoT design means your entire team can be part of the design process.

HOW NO-CODE DESIGN COULD ACCELERATE INDUSTRIAL IOT DEVELOPMENT

By Philip Ling | January 14, 2022

While this approach is viable, it does present an issue. The hardware in each example is not optimized for industrial applications. In a typical workflow, development kits and evaluation modules are a steppingstone toward a production release. It is usually possible for a customer to use the same circuit design in their own product, which could therefore run the same code. But it still requires some design effort to optimize the evaluation module for production applications.

A more viable approach is to standardize your designs on a portfolio of hardware solutions for industrial applications. This is the right approach to take when using a low-code or no-code strategy toward scaling up in the Industrial IoT. Avnet developed the IoT Express Connect tool to support this approach. It includes a library of components, including hundreds of sensors, that connect through the tool and use a custom IoTConnect® dashboard.

The software behind a no-code design strategy

With no obvious access to the code, users need some other way to interact with the software they are developing. Often, this takes the form of a visual programming language, or VPL, and flow-based programming (FBP).

In traditional FBP, process blocks have inputs and outputs. Depending on the function, the block may include parameters that can be set or adjusted. The inputs and outputs are “wired” together, graphically, to describe the flow of data. Node-RED is a well-known example of an FBP based low-code platform for IoT development.

Transferring a flow to a hardware platform involves interpreting the design in a way that the hardware understands. With Node-RED, and others, an interpretive layer handles this, such as Firmata. This comprises a protocol which works with a corresponding client. Each platform needs a hardware-specific port of the protocol, such as the Arduino port. There is also a Firmata client for Python. Avnet’s IoT Express Connect tool also uses FBP and VPL to provide the no-code experience for users.

As a no-code approach to design, the majority of the configuration needed to make it happen hides from view. In fact, every no-code platform includes a huge amount of code under the hood. To the user, of course, this is unseen.

Getting to market with no-code development

Using a no-code/low-code strategy can deliver huge productivity gains when deploying large IoT systems in an industrial environment. The main challenges relate to the scale of development and amount of design diversity involved. While no-code/low-code reduces design effort, it can come at the cost of support for diversity.

By creating a tool that includes hundreds of sensors and other components, configurable in an almost endless number of combinations, IoT solutions can scale. The IoT Express Connect tool also includes built-in security and support for cloud deployment and device management. This means the global solution updates over the air, adding new features in the field. The platform supports the use of AI and edge computing, as well.

Using no-code solutions developers can create large scale connected systems in much less time than it takes to achieve the same result using conventional development approaches. The simplified approach empowers more team members to contribute to the system’s development, not just engineers with specialist knowledge. Avnet’s approach uses libraries that can also include technologies from third party providers.

Contact your Avnet representative to find out how you could be using no-code/low-code to accelerate your industrial IoT development, visit our [IoT Express Connect](#) page to learn more.