Images: Sigmatek GmbH & Co KG

Object-oriented programming and graphic representation

Not Future – But Present

With an object-oriented engineering platform machine concepts can be implemented highly structured, modular and quickly. These are major advantages, especially when it gets increasingly complex and must keep up with modular hardware. The all-in-one engineering tool LASAL from SIGMATEK also operates with less code and provides clarity with graphic representation.

Sure: Simple applications with few lines of code don't require an object-oriented approach, as machine manufacturers can also achieve their goal with conventional programming. But when an application is more complex and smarter, object-oriented programming (OOP) is the right choice. Especially for series machines with varying features and the requirement to integrate these machines into an intelligent machine network. The object-oriented programming concept stands for modularity, structure and simple reusability of the application – this allows the software to keep pace with the modularity and scalability of the hardware. Meanwhile, object-oriented programming methods have also made their way into machine building. Particularly with the younger generation of automation engineers, OOP was already a part of the training. There are already numerous software architects, who have experience with this modern programming method and think machine software object-oriented in functions and interfaces.

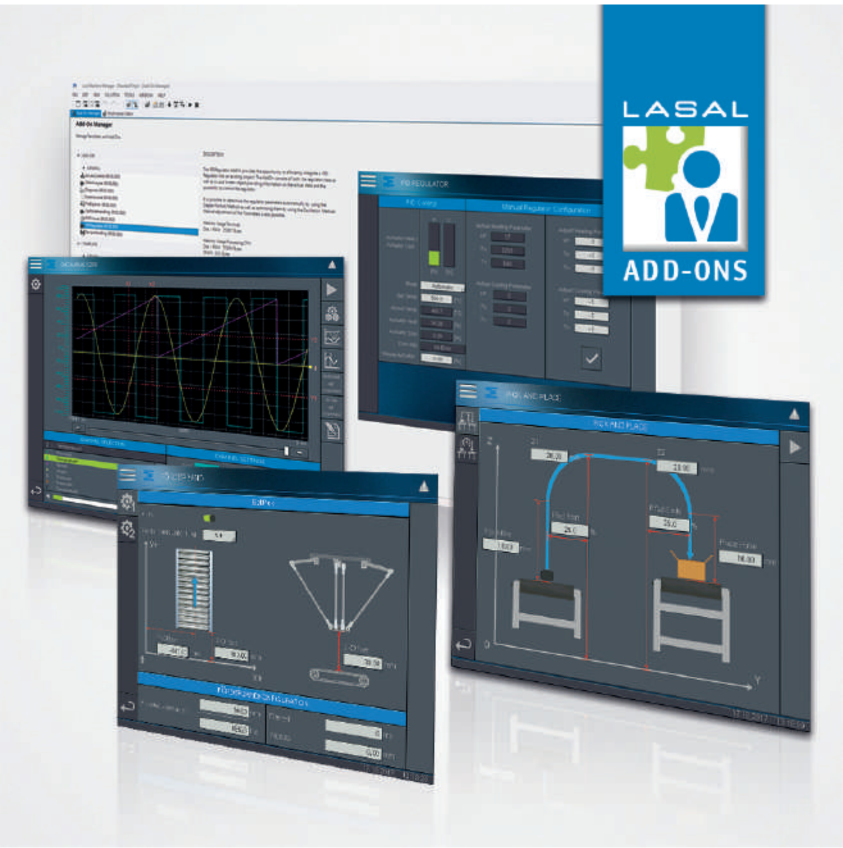
Low Code – Less Programming

For 20 years, SIGMATEK has already been using object orientation. The all-in-one engineering tool LASAL according to IEC 61131-3 is object-oriented from scratch and was also developed as a low-code platform that saves the user programming effort. If, as with LASAL, the object orientation is implemented with graphic representation, the clarity and maintainability of the software project increase enormously. And that is a crucial added value for the machine builder since today it is usually through the software that it generates most of its added value. The application can also be developed consistently with LASAL: control, visualization, motion and Safety, including service tools. OOP opens new possibilities and directions to the machine builder but requires a different approach to application development. Based on the overall project, software developers consider which functions are required in the machine and how these are connected

With the top-down design, the tasks are disassembled and simulated visually: which machine components speak with one another, which exchange data. Objects and interfaces are defined and displayed in the network. "No lines of code are required to understand the function of the application in a LASAL network" explains Franz Aschl, Innovations Manager at SIGMATEK.

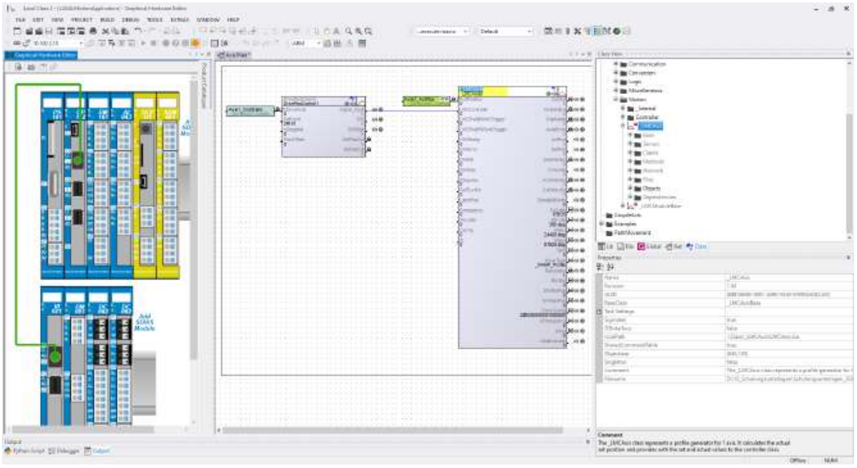
Software in a Toolkit System

When the functions and communication interfaces are defined, the development begins from the bottom-up. With help from the extensive LASAL libraries, the basic functions of a machine can be assembled with a few clicks, provided with parameters and interfaces and the software architect can then already test the application. The diverse LASAL libraries contain over 10,000 classes. Modern templates as well as tested, ready-to-use software components are also available: function-specific Add-Ons and topic-based packages. Examples include controllers, recipe management, alarm system, Safety, data analysis but also complex motion functions such as multi-axis control or delta robot commissioning. "In recent times, the library for communication has grown significantly and extends from TCP/IP through OPC UA Client, as well as Server and MQTT to SSL encryption", says Franz Aschl. There is an increased demand for networking – horizontally, for example, with camera systems or third-party controls and vertically with higher systems such as ERP, MES and cloud. With the use of predefined software components, the development work for basic functions can be reduced up to 70 %. The applications engineer can concentrate on the implementation of machine-specific innovative features.

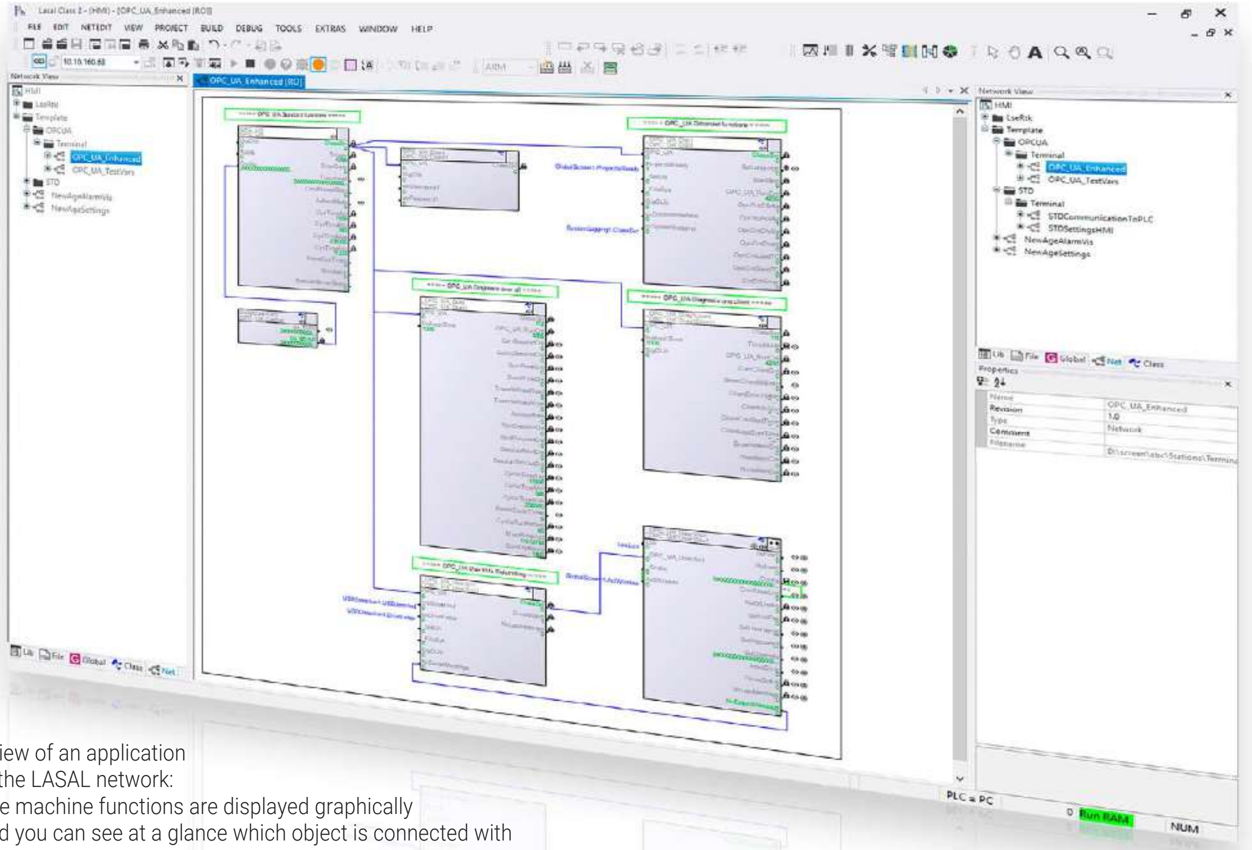


Highest Possible Transparency

LASAL is also ideal for agile development teams, as the work must be very structured. "Every programmer knows the dilemma of sloppily implemented programming, where variables are written spread across the project so that the effects of a program change are practically unforeseeable," knows Franz Aschl. With OOP it's completely different: The objects are isolated from the outside, so that unwanted influences are nearly impossible.



"Variables can only be manipulated via the corresponding methods. Clear interfaces are therewith predefined, which can then be applied," says Franz Aschl. "The visual representation of the encapsulated objects in the network with all communication interfaces is unique and provides maximum transparency, as well as the Low-Code principle." "Through the graphic display of objects, the text editor is only used at the end – unlike other object-oriented tools," explains Franz Aschl. The software engineer can implement the class methods in proven programming languages such as Structured Text (ST), Instruction List (IL), Ladder Diagram (LD) in accordance with IEC 61131-3 or C. By taking over repetitive functions, modern engineering tools such as LASAL simplify the use of object-oriented programming for the user. For example, the code for declaring classes, variables or interfaces is automatically created in the background.



View of an application in the LASAL network: The machine functions are displayed graphically and you can see at a glance which object is connected with others and exchanges data.

"Variables can not only be visualized but also provided across companies via global WAN networks so that it is also determined during development, which interfaces can be operated internally or are visible," says Aschl.

Live Visual Debugging

The realistic behavior of the components in LASAL can be seen in Online mode in real time. In the detail view, optional slots and bus connections are visible, current values and component properties can be read or configured. In addition, the order of the hardware in the LASAL project can be compared with the actual control hardware configuration. Possibly differences are graphically displayed and products can be accepted or removed individually. With comfortable tools for simulation and visualization of all processes, a complete testing environment up to the debugger and including central management of projects and versions, the transparency and quality of the software can be increased. "In LASAL, even live visual debugging is possible. This means that the actual values are not only visible in the program code, but also live in the graphic object image," continues Aschl further. "That saves a lot of time."

Franz Aschl, Sigmatek

Whoever doesn't believe in object orientation is stuck in the past

Evolution Instead of Revolution

"Whoever doesn't believe in object orientation is stuck in the past," says Franz Aschl provocative but with a wink. Objects correspond to machine function and, like proven machine components, can always be reused. "The code is graphically prepared and provides clear interfaces. Via this structure and the Low-Code approach, the code remains readable and easily adaptable after years and possible employee changes. The code becomes sustainable and can be clearly maintained. In addition to costs, it saves time and nerves. "Regarding the application itself, there are mostly only minor changes: evolution instead of revolution. Using modern object orientation, series machine builders can easily acquire software - today and tomorrow," Aschl says in summary.

Ingrid Traintinger,
Manager Marketing Communication,
Sigmatek GmbH & Co KG
www.sigmatek-automation.com/de

Images: Sigmatek GmbH & Co KG

SPS MAGAZIN
Auszug aus dem SPS-MAGAZIN, Fachzeitschrift für Automatisierungstechnik, Ausgabe 7/2020.
Digitales Belegexemplar mit freundlicher Genehmigung des SPS-MAGAZINs. Dieses Dokument ist ausschließlich zur elektronischen Speicherung durch den Autor sowie zur Weitergabe per E-Mail bestimmt. Abweichende Verwendung nur mit schriftlicher Zustimmung des Verlages.