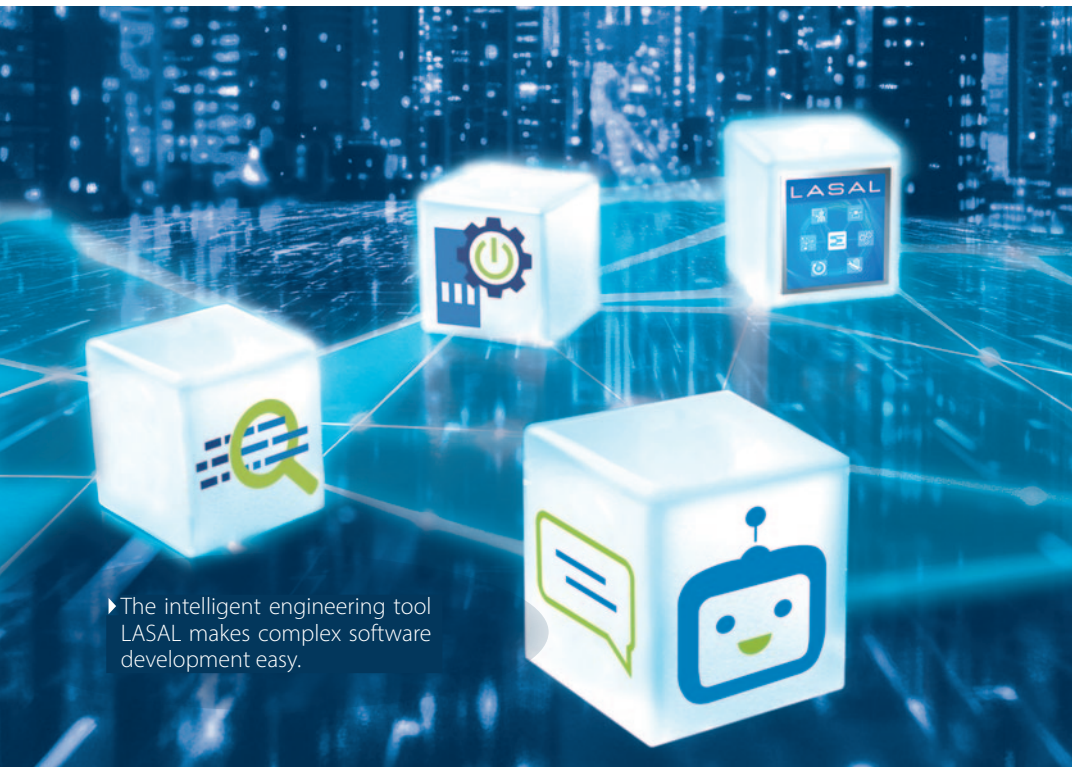


All-in-one engineering tool with quality assurance, auto-tuning, and AI support

Simple coding in a complex world



▶ The intelligent engineering tool LASAL makes complex software development easy.

The daily work of software developers in mechanical engineering is characterized by increasing complexity, scalability, growing dynamics, and vast amounts of data. Modern engineering tools such as LASAL from SIGMATEK aim to reduce the workload of application designers to accelerate engineering processes while maximizing the quality, reusability, and maintainability of code.

LASAL, the object-oriented all-in-one engineering tool for sequence control, visualization, motion and Safety has been successfully used in automation technology for 25 years. Its range of functions is continuously being expanded – as is its user-friendliness. Advanced features and enhancements offer machine builders even greater convenience and accelerate development – without losing the option for experts to dive deep into the system.

“Our goal is to provide our customers with a complete, comfortable development environment that grows with the task. Developers should be relieved of routine tasks and supported in programming

so that they can concentrate on customer-specific functionalities that bring real competitive advantages,” explains SIGMATEK Managing Director Alexander Melkus.

Helpers around the PLC

Over the years, numerous convenience features have been added, including version management, Matlab connector, hardware editor, and Python scripting. Since 2009, the Engineering Suite has included LASAL SafetyDesigner, a seamlessly integrated tool for configuring safety-related applications. Modern visualization technologies such as HTML5 are also integrated into the tool, and the LASAL Machine Manager simplifies the handling of multi-CPU solutions. A central element are the extensive libraries,

which have grown significantly over the years and provide ready-to-use function blocks and packages for a wide range of machine functions. A new addition is the connection of model-based development with Digital Twin. The latest enhancements will be presented at the SPS fair: static code analysis, autotuning for the MDD 2000 servo drives, and the AI assistant .

Identifying errors early on

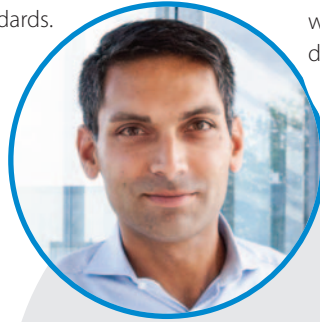
With the growing complexity of automation software, code quality assurance is becoming increasingly important. This is the only way to ensure scalability and maintainability over many years. The later errors are discovered, the more time-consuming and expensive it is to fix them. That's why it's worth identifying and avoiding typical sources of error already in the development phase. Static Code Analysis (SCA) makes it possible to automatically check software code without executing it. This allows potential errors and violations of internal coding standards and best practices – such as those defined by PLCopen – to be identified at an early stage. A uniform coding style also facilitates collaboration in larger development teams and guarantees the readability and



maintainability of the code – an essential aspect, especially for long-lasting machines and systems. In addition, the use of SCA ensures compliance with industry-specific guidelines and standards.

Static code analysis in LASAL CLASS

LASAL CLASS offers an optional extension for control code programmed in Structured Text: static code analysis. It checks the entire code base for possible runtime errors or undefined behavior – such as uninitialized variables, the return of a pointer to a local variable, or implicit conversions, etc. – without having to execute the code. To do this, numerous check rules run in parallel, analyzing different aspects of the control code and highlighting potential quality defects or rule violations.



“Developers should be relieved of routine tasks and supported in their programming work.”

Alexander Melkus,
Sigmatek

Clear results

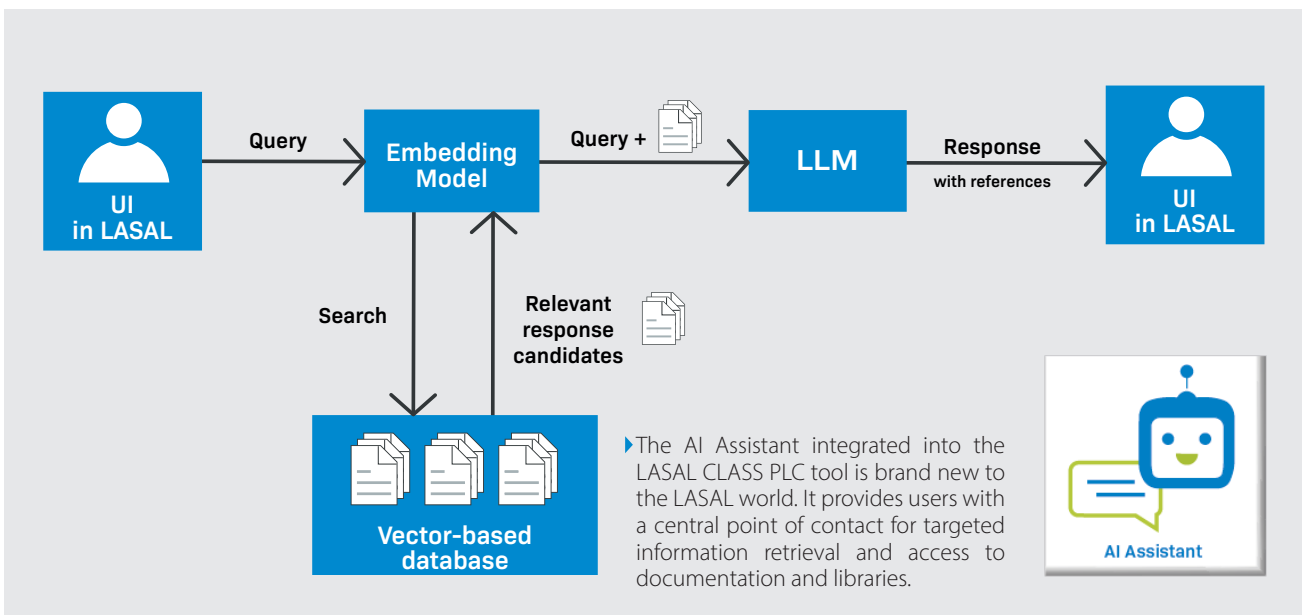
The analysis results are clearly displayed in a separate output window. Similar to the compiler output of LASAL, errors can be double-clicked to jump to the corresponding location in the code. Configurable, color-coded priority levels (trace, info, warning, error) provide additional clarity. It is also possible to suppress individual error messages line by line using code comments. Black, white, and ignore lists for the corresponding regulations allow the check to be even better adapted to individual requirements. It is also possible to exclude specific tables with files and folders from the analysis. “Static code analysis is a kind of safety net: it prevents trivial but hard-to-find errors, reduces the workload on the development team, and increases productivity and software quality – both during creation and maintenance,” says Maximilian Leschanowsky, Head of Software Development at SIGMATEK, summarizing the advantages.

Flexible configuration

Users have flexible configuration options: More than 50 predefined regulations for variables, types, functions, and methods can be individually parameterized. In addition, metrics – i.e., measures for evaluating various aspects of the code base – can be calculated and naming conventions can be fully configured. Rules, metrics, and naming conventions can be activated or deactivated individually or in groups. Automated basic checks allow reviewers to focus on the correctness of the logic and architecture.

Autotuning: Automatic controller optimization

In drive technology, the commissioning of servo axes is greatly accelerated and simplified by autotuning. Instead of time-consuming manual parameterization and repeated readjustment of current, speed, and position controllers, to match motors and axes to the mechanical system, the new autotuning function of the MDD 2000 series servo drives optimizes the controllers fully automatically – at the touch of a button. Parameterization is fully automated in the PLC and in the drive itself and covers the entire



control chain: the parameters for current, speed, and position controllers as well as torque feedforward control. While the MDD 2000 records the necessary measurement data, the PLC performs the mathematical evaluation. This includes discrete Fourier transforms (DFT), the creation of Bode diagrams using sweep frequency response analysis (SFRA), and the automatic adjustment, optimization, and validation of controller parameters based on defined criteria – all without manual intervention.

Fine-tuning of all control stages

For current and speed controllers, start values are calculated from the stored motor parameters. The amplification is then automatically varied until the tuning targets are achieved. For speed controllers, this is done by gradually adjusting the amplification and integration time. The position controller can also be set automatically. Its amplification is derived from the dynamics of the speed controller. Torque feedforward is determined using the same principle, significantly minimizing the amplitude and duration of vibrations in the axis system. In addition, the mass moment of inertia can be determined either via frequency response analysis or by applying a constant acceleration. The autotuning function is conveniently accessible to the user via a visualization object (VOV). After selecting the desired control loop and defining the tuning range, the user starts the fully automatic run. The tuning is performed by the algorithm without any further action by the user. To increase transparency, all measurement data can be exported as CSV files. This allows frequency data to be displayed directly in common analysis tools such as Excel, Matlab, or the LASAL CLASS Data Analyzer. In complex machines, multiple drives often need to be coordinated in parallel. Here, a higher-level class in LASAL CLASS ensures that all axes involved start synchronously, perform their measurements, and then receive the new parameters. Autotuning is a powerful tool that speeds up machine commissioning, improves control quality, and prevents operating errors.

tation, libraries with classes and functions, proven best practices, and FAQs. The user gets a central point of contact for targeted information retrieval and access to documentation and libraries.

SIGMATEK uses Agentic RAG (Retrieval-Augmented Generation) to find the relevant information in the knowledge database and convert it into a useful response for the user with the help of a modern LLM. The focus is on maximum accuracy – the AI assistant does not hallucinate. If it cannot find a result, it will let you know. Its answers include links to the source of the information for more details and suggest library functions. Users can ask questions around the clock and receive answers in their respective language.



“AI-supported tools are driving automation forward. However, protecting company-internal know-how remains crucial.”

Maximilian Leschanowsky,
Sigmatek

AI: Goal and vision

The goal is to provide users with perfect support when implementing basic functions. Typically, LASAL CLASS applications consist of at least 50% library classes that are imported and parameterized to perform the desired function. The AI Assistant can speed up this process by directly recommending the most suitable functionalities for the task at hand. The vision: If a developer asks, for example, “How can I implement an alarm system?”, the AI Assistant directly suggests suitable predefined classes from the library, such as “_AlarmXBuffer.” If this suits the developer, they give the command to load the library and import the corresponding classes (_AlarmXBuffer, _AlarmXPara, and _AlarmDisplayList for visualization). If necessary, the assistant explains the necessary parameterization step by step. “We are at the beginning of an exciting journey: AI-supported tools are driving automation forward – they take over routine tasks, make specific knowledge immediately available, and generate code automatically. The result is shorter development times, higher quality, and cost-efficient, reliable automation. However, it is crucial that confidential company know-how remains protected – because it is precisely this specific knowledge that secures competitive advantage and must not become common property,” summarizes Maximilian Leschanowsky.

AI Assistant for more efficiency

The AI Assistant is brand new to the LASAL world. Using AI-based tools increases development efficiency – a decisive advantage in terms of workflow and iteration cycles. In the first step, the AI Assistant integrated in the LASAL Class PLC tool acts as an intelligent knowledge database that is fed and trained with SIGMATEK-specific know-how: hardware and software documen-

Ingrid Traintinger
Marketing Kommunikation
Sigmatek GmbH & Co KG
www.sigmatek-automation.com