Structured software development

Codesys has extended IEC 61131-3 application engineering by reducing the development effort that is caused by recurring procedures: The Codesys Development System provides automated methods.

For certain mobile machines, it is mandatory to fulfill standard compliances according to IEC 61508 for SIL-2 or SIL-3 safety controllers and their software components, such as applications and communication-stacks. Traditionally, engineering requires many manual tasks to comply with the standard. Modern engineering already benefits from certified communication stacks, for example CANopen Safety.

For safety development, specific methods are recommended for the development of life-cycle models. To reduce the risk of expensive field service operations, the main focus of the methods must be on coding, debugging, and testing, followed by concept work and administrative tasks, such as code management. With automated methods fully integrated into the Codesys Development System, the following recommended IEC 61508 methods can be completed more efficiently and reliably — regardless of the chosen development model.

The Unified Modeling Language (UML) was invented to graphically describe different aspects of the application software. In Codesys, it is possible to use UML in each IEC 61131-3 project as an additional integrated programming language, supporting two types of engineering. The class diagram is a type that shows the internal software structure by means of interfaces and software classes. The state chart is used to describe the behavior of the software during different states. UML state charts are similar to SFC diagrams and can be used to structure the different logical application steps.

Codesys UML is an ideal add-on for object-oriented application programming. Furthermore, it provides an efficient way for communicating with customers and developers by means of a single tool. It supports continuous engineering in IEC 61131-3 languages based on information that is part of the UML diagram by reusing UML models in the application.

The code administration is a non-functional method for improving the code quality of safety software by means of prevention. It documents the meta-information. Codesys SVN extends the capabilities of the Codesys Development System by interfacing an Apache Subversion server, so that the single objects that are part of the project file can be administrated. Thus different developers can work on dedicated objects of the same project at the same time. Furthermore, branches of the developed project can be administered and versioned.

The execution of checklists and formal inspections are a minimum recommendation for SIL-1 to SIL-4 safety levels, and formal inspections are even highly recommended for SIL-3 and SIL-4. PLCopen recognized these requirements and provides a technical paper with coding guidelines for IEC 61131-3 applications as of July 2015.

The primary aim of Codesys Static Analysis is the improvement of the quality of the application code by automatic detection of code weaknesses and anomalies. Using this tool, the manual effort of code reviews and metrics calculations can be reduced. Codesys Static
The WAGO-I/O-SYSTEM 750 XTR – TAKING IT TO THE EXTREME

- eXTReme temperature ... from -40°C to +70°C
- eXTReme vibration ... up to 5g acceleration
- eXTReme isolation ... up to 5 kV impulse voltage
- eXTReme dimensions ... as compact as 750 Series standard

www.wago.com
Analysis checks the application code against numerous integrated coding guidelines and naming conventions. Furthermore, it calculates metrics of the IEC 61131-3 program. For a maximum benefit, the tool is integrated seamlessly into the IEC 61131-3 IDE so that application code import or export is not required.

Time measurement on the controller can be provided by event-based measurement tools, such as Codesys Profiler. The runtime between two predefined code points is measured by instrumentation of the code in order to fulfill the recommendations for SIL-2 to SIL-4. Codesys Profiler automatically adds and removes extra measurement code at the calling and exit points of the POU (Program Organization Unit). As a result, the following information can be retrieved from the controller:

- Total time spent in call,
- Average time of all POU calls of a single cycle,
- Minimum and maximum processing time over multiple cycles,
- Number of calls,
- Standard deviation of average measured time.

After profiling is switched off, the extra measurement code is deleted automatically, thus no unused code exists in the final application.

Safety-compliant software testing has to include independent test specifications, confirmation of intended functions, and test documentation. The Codesys Test Manager provides two options for performing tests and fulfilling IEC 61508 requirements. Tests can be written in IEC 61131-3 for execution in real time on the controller or they can be defined to operate via the monitoring interface. The tests can be developed in parallel to the application. The planned test cases can be described directly from the integrated editor and referenced to the appropriate requirement. When the tests have been performed on the device, the results are displayed in a result window and can be saved to a file or directly within the Codesys Development System. With continuous model-based testing, the functional quality is always transparent and can be documented properly. Furthermore, the effort for recurring tests is reduced considerably.

Development tools which provide methodical development support make engineering life easier. They reduce the manual effort for safety certifications and improve the quality of IEC 61131-3 applications in automation technology. In addition, non-safety applications also benefit from conventional methods known throughout the IT world. Although they are not mandatory, these methods help reduce unwanted application behavior, field service missions and increase application quality right away.

After profiling is switched off, the extra measurement code is deleted automatically, thus no unused code exists in the final application.

Safety-compliant software testing has to include independent test specifications, confirmation of intended functions, and test documentation. The Codesys Test Manager provides two options for performing tests and fulfilling IEC 61508 requirements. Tests can be written in IEC 61131-3 for execution in real time on the controller or they can be defined to operate via the monitoring interface. The tests can be developed in parallel to the application. The planned test cases can be described directly from the integrated editor and referenced to the appropriate requirement. When the tests have been performed on the device, the results are displayed in a result window and can be saved to a file or directly within the Codesys Development System. With continuous model-based testing, the functional quality is always transparent and can be documented properly. Furthermore, the effort for recurring tests is reduced considerably.

Development tools which provide methodical development support make engineering life easier. They reduce the manual effort for safety certifications and improve the quality of IEC 61131-3 applications in automation technology. In addition, non-safety applications also benefit from conventional methods known throughout the IT world. Although they are not mandatory, these methods help reduce unwanted application behavior, field service missions and increase application quality right away.

After profiling is switched off, the extra measurement code is deleted automatically, thus no unused code exists in the final application.

Safety-compliant software testing has to include independent test specifications, confirmation of intended functions, and test documentation. The Codesys Test Manager provides two options for performing tests and fulfilling IEC 61508 requirements. Tests can be written in IEC 61131-3 for execution in real time on the controller or they can be defined to operate via the monitoring interface. The tests can be developed in parallel to the application. The planned test cases can be described directly from the integrated editor and referenced to the appropriate requirement. When the tests have been performed on the device, the results are displayed in a result window and can be saved to a file or directly within the Codesys Development System. With continuous model-based testing, the functional quality is always transparent and can be documented properly. Furthermore, the effort for recurring tests is reduced considerably.

Development tools which provide methodical development support make engineering life easier. They reduce the manual effort for safety certifications and improve the quality of IEC 61131-3 applications in automation technology. In addition, non-safety applications also benefit from conventional methods known throughout the IT world. Although they are not mandatory, these methods help reduce unwanted application behavior, field service missions and increase application quality right away.
Lift-Report takes part regularly in all major fairs and trade events – worldwide.

Take the advantage of the magazine and make new contacts in the lift industry.