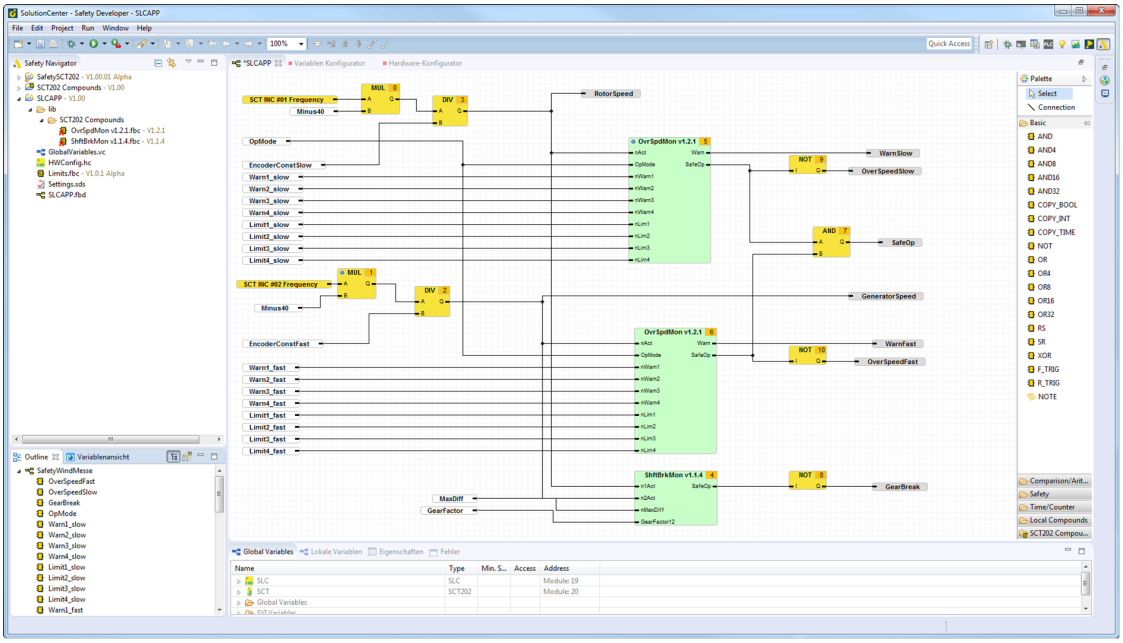
Safety Developer Engineering Tool

For the security relevant engineering steps the SolutionCenter contains the Safety Developer that includes all tools required for safety-related programming in accordance with IEC 61508 and PLCopen. Safety Developer has been developed and certified in close collaboration with TÜV.

All the methods required for logging and the machine manufacturer´s duty of verification are integral: password management, fail-safe program transmission, tamper-proof logging on the target device, documentation of the safety program and all software components used, unique identification of the safety modules and the programming itself.

Features

- Integrated in the SolutionCenter all-in-one engineering tool
- Full-graphic Function Block Diagram editor with autorouting
- Certified safety modules in accordance with PLCopen Safety
- Standard modules for logical links, timer and arithmetic operations acc. to IEC 61131-3
- Color coding of secure and insecure signal flow
- Grouping of circuit parts for repeat use (compounds)
- Adjustable test depth for the project translation
- Variable monitoring, value simulation and break points
- Bidirectional exchange of values between secure and insecure controller
- Open application programming interfaces (PLC, C, C++) for online monitoring, evaluation and connection of user-defined tools
- Unique identification of the safe hardware
- Configuration of the clocked self-monitoring of inputs/outputs
- Communication to safe hardware via Ethernet (M200) or serial
- Certified redundant program download
- Logging of the acceptance state in PDF format
- Direct connection to version management
- Online monitoring of all I/Os also in safe operating mode



Programming acc. to PLCopen Safety

The safety application is programmed in a free-graphic function block diagram editor (Safety Editor) acc. to IEC 61131-3. The module set includes a library of safety modules that have been strictly implemented, tested, and certified in accordance with the PLCopen Safety standard. For the additional logic required, standard modules such as timers, arithmetic and logical operations are available.

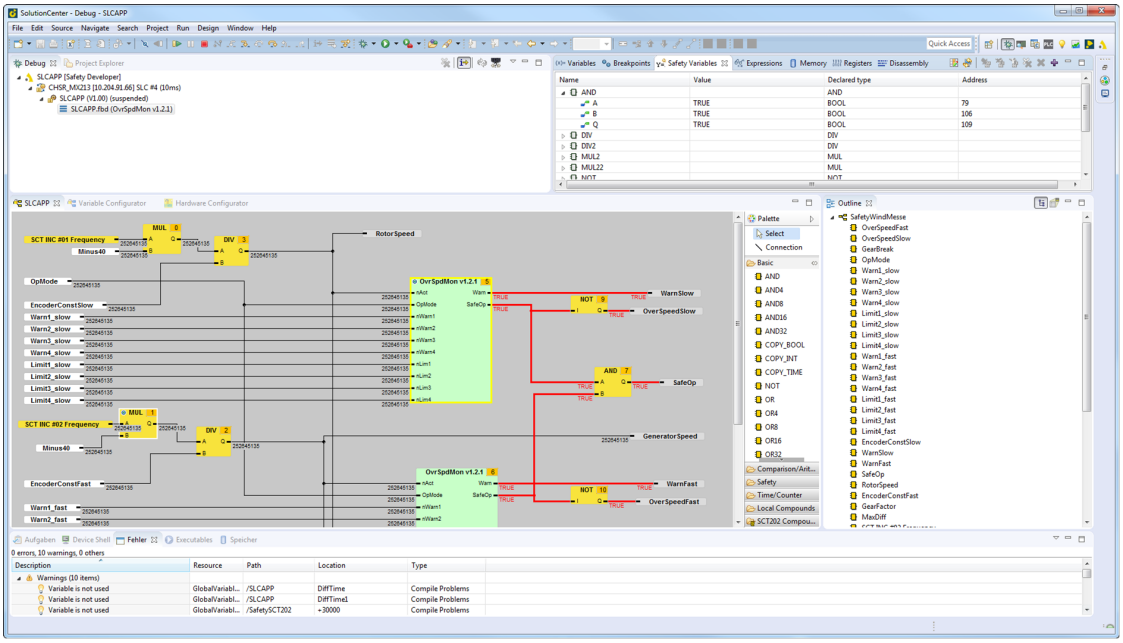
The application program can be organized in separated functional units and sub programs in order to structure the program. The execution sequence of the modules is presented graphically and can be corrected by the user. Redundant hardware inputs are summarized via equivalence or antivalence blocks and subsequently presented as a safe signal in the program. The data types bool, integer and time, are supported.

Visibility in the standard program

The transparent exchange of signals functions in both directions. In the Safety Developer it is configured which values – irrespective of the states of the safe I/Os – should also be visible in the unsafe world. Thus intermediate results in networks and the status of modules can be presented in a visualization, recorded with the Scope or evaluated in a PLC program. Thus there are extensive diagnostic possibilities and a high level of operating convenience.

Safe and unsafe paths

Signals from the standard sequential program and from the I/O modules of the control system can be used as unsafe input and output signals in the safety program. The classification of a signal as safe or unsafe is presented with color coding.



User-specific templates

A logical circuit that is structured from a group of basic modules can be put together and given an interface in the project for a function group. Thus separate user-specific function blocks (templates) can be created and used in the project multiple times. These groups or even complete networks can also be transferred from one project into the next project.

Change tracking

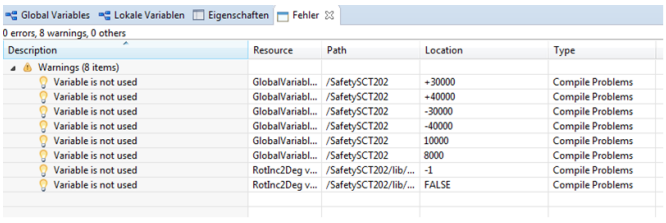
In the Safety Developer version management is an integral component – regardless of the logging and verification obligation. The version management database is operated directly from the project navigator. A local history is always kept automatically, which enables reversal of changes even without a genuine version management system. Undo/ redo functions are of course also available.

Modularity

The Safety Developer takes the flexible requirements in today's industry into account through its modular project management. Subprojects, FBD networks and also additional safe I/O modules of the project can be activated depending on the expansion stage, signals can be connected differently depending on the expansion stage. In this way each expansion stage up to the maximum machine expansion can be completely created and tested within an overall project. Adaptation to the actual degree of expansion is achieved by combining the required parts. Commissioning of individual machine parts is also possible in this manner.

Program verification

The program for execution on the target system is not compiled, but rather is translated into a script that is checked and executed redundantly by the firmware of the target system. The Safety Developer recognizes possible sources of error in the code when the script is generated and points them out.



Logging

The verification obligation is supported in different ways. A project report can be generated for logging the acceptance, which graphically displays the entire program code, exports all project settings, the safety scripts and its own acceptance tables for on-site commissioning. The tamper-proof logbook of the Safety Controller logs each security relevant change in the system, such as the download of a changed program. Thus every access is traceable with user name, date and time.

Optionally, the entire project can be stored on the Safety Controller and can be opened and further processed from the there. Additional customized information, such as author, version history and additional comments can be stored for each network.

Hardware configuration

In addition to the tools for variable selection, programming, and logging, the safe hardware can also be directly configured in the Safety Developer.

This includes not only assignment of unique channel names, but particularly also allocation of the controller to the project, the adding of additional safe I/O modules, and specification of test intervals for clocked lines, which then are tested automatically by the hardware for short circuit, cross-connection and external voltage. Safety-relevant, required unique module identification that excludes the possibility of swapping modules after a service deployment is also executed directly in the Safety Developer.

Communication between Safety Developer and the control system for program download, diagnostics and configuration is executed conveniently via the Ethernet Interface of the M200 control system. Alternatively, communication can also be executed directly with the Safety Controller via a serial RS232 interface, which also enables the use of the Safety Controller as a stand-alone solution without a surrounding control system.