Safety Developer

Safety Developer Engineering-Tool

Für die sicherheitsrelevanten Schritte des Engineerings enthält das SolutionCenter den Safety Developer, der alle erforderlichen Werkzeuge zur sicherheitsgerichteten Programmierung nach IEC 61508 und PLCopen enthält. Der Safety Developer wurde in enger Zusammenarbeit mit dem TÜV entwickelt und zertifiziert. Für die Protokollierung und Beweispflicht des Maschinenherstellers erforderliche Mittel sind integraler Bestandteil: Passwortverwaltung, fehlersichere Programmübertragung, fälschungssichere Protokollierung auf dem Zielgerät, Dokumentation des Sicherheitsprogrammes und aller verwendeter Softwarekomponenten, eindeutige Identifizierung der Safety-Module und die Programmierung selbst.

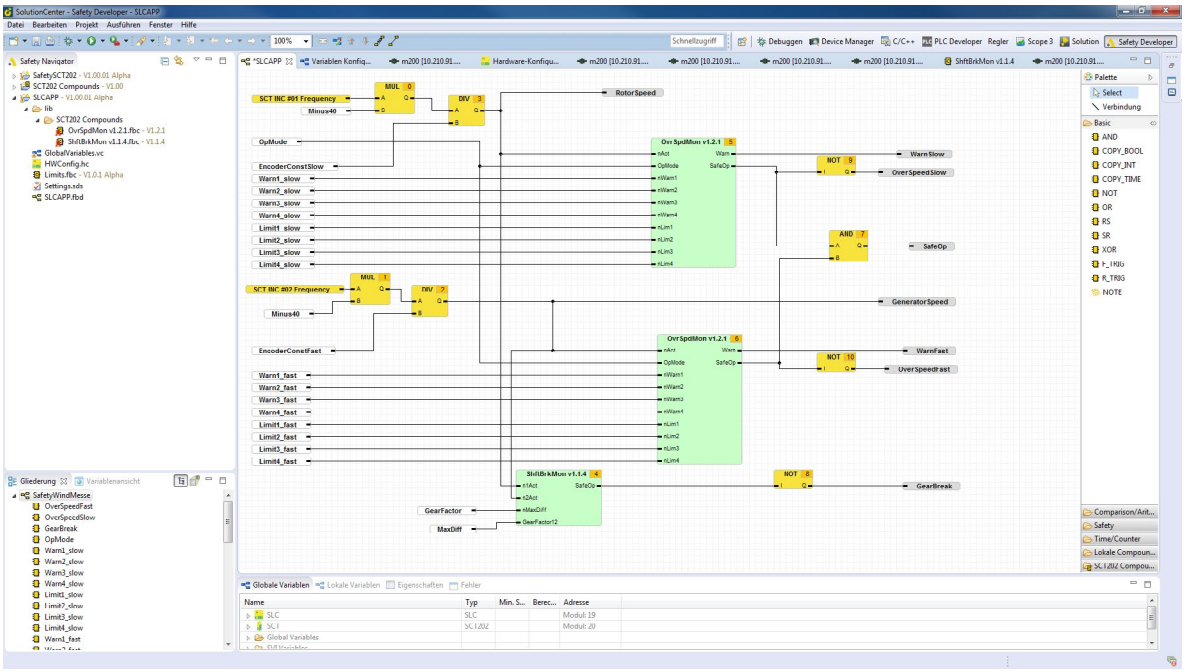
bachmann. SolutionCenter

Safety

PLCopen safety

TÜV SÜD

- Integriert ins All-in-One Engineering-Tool SolutionCenter
- Vollgrafischer Funktionsplan-Editor mit Autorouting
- Zertifizierte Safety-Bausteine nach PLCopen Safety
- Standard-Bausteine für logische Verknüpfungen, Timer und arithmetische Operationen nach IEC 61131-3
- Farbliche Kennzeichnung von sicherem und nicht sicherem Signalfluss
- Gruppieren von Schaltungsteilen zur wiederholten Verwendung (Compounds)
- Einstellbare Prüftiefe bei Projektübersetzung
- Variablenüberwachung, Wertesimulation und Break-Points
- Offene Programmierschnittstellen (PLC, C, C++) zur Online-Überwachung, Auswertung und Verbindung von benutzerdefinierten Tools
- Bidirektionaler Austausch von Werten zwischen sicherer und unsicherer Steuerung
- Eindeutige Identifikation der sicheren Hardware
- Konfiguration der getakteten Selbstüberwachung von Ein-/Ausgängen
- Kommunikation zur sicheren Hardware über Ethernet (M1) oder seriell
- Zertifizierter redundanter Programm-Download
- Protokollierung des Abnahmestandes im PDF-Format
- Direkte Anbindung an Versionsverwaltung
- Online-Monitoring aller I/Os auch im sicheren Betriebsmodus



Programmierung nach PLCopen Safety

Die Programmierung der Sicherheitsapplikation erfolgt in einem freigrafischen Funktionsplan-Editor (Safety Editor) nach IEC 61131-3. Der Bausteinvorrat umfasst eine Bibliothek von Sicherheitsbausteinen, die strikt nach dem PLCopen Safety Standard implementiert, getestet und zertifiziert wurden. Für die zusätzlich erforderliche Logik stehen Standardbausteine wie Timer, arithmetische und logische Operationen zur Verfügung. Die Applikation kann in getrennte Funktionseinheiten und Unterprogramme gegliedert werden, um das Programm zu strukturieren. Die Abarbeitungsreihenfolge der Bausteine wird grafisch dargestellt und kann vom Anwender konfiguriert werden. Redundante Hardware-Eingänge werden über Äquivalenz- bzw. Antivalenz-Bausteine zusammengefasst und anschließend im Programm als ein sicheres Signal dargestellt. Die Datentypen Bool, Integer und Time werden unterstützt.

Sichtbarkeit im Standardprogramm
Der transparente Austausch von Signalen funktioniert in beide Richtungen. Im Safety Developer wird konfiguriert, welche Werte – abgesehen von den Zuständen der sicheren I/Os – zusätzlich in der unsicheren Welt sichtbar sein sollen. So können Zwischenergebnisse in Netzwerken und Status von Bausteinen in einer Visualisierung dargestellt, mit dem Scope aufgezeichnet oder in einem SPS-Programm ausgewertet werden. Dadurch entstehen umfangreiche Diagnosemöglichkeiten und ein hoher Bedienkomfort.

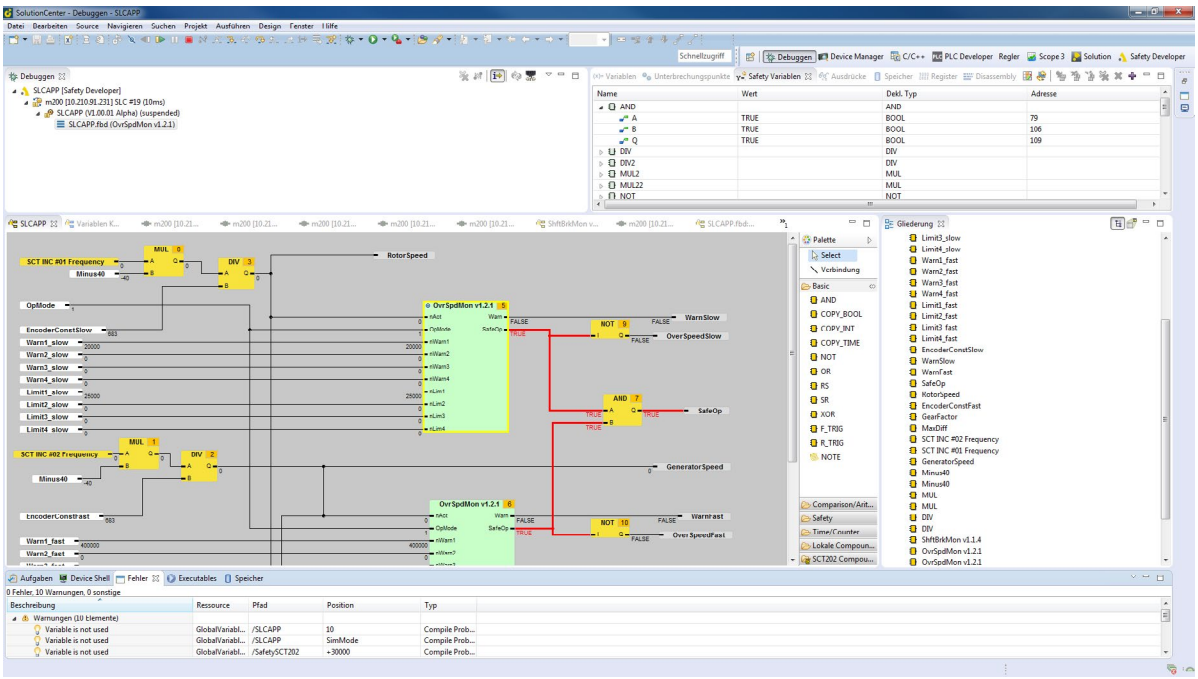
Sichere und unsichere Pfade
Signale aus dem Standard-Ablaufprogramm und von den I/O-Modulen des Steuerungssystems können im Safety-Programm als unsichere Ein- und Ausgangssignale verwendet werden. Die Einteilung eines Signales in sicher oder unsicher wird farblich dargestellt.

Safety Developer

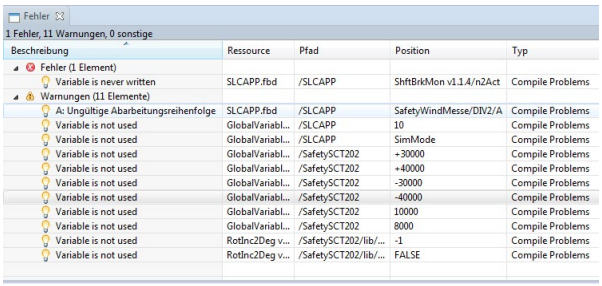
Benutzerspezifische Templates
Eine logische Schaltung, die aus einer Gruppe von Grundbausteinen aufgebaut wird, kann im Projekt zu einer Funktionsgruppe zusammengestellt und mit Schnittstellen versehen werden. Somit lassen sich eigene benutzerspezifische Funktionsbausteine (Templates) erstellen und im Projekt mehrfach verwenden. Diese Gruppen oder auch vollständige Netzwerke können darüber hinaus von einem Projekt ins nächste übernommen werden.

Änderungsverfolgung
Im Safety Developer ist die Versionsverwaltung ein integraler Bestandteil – unabhängig von der Protokoll- und Nachweispflicht. Die Bedienung der Versionsverwaltungs-Datenbank erfolgt direkt aus dem Projektnavigator. Eine lokale Historie wird automatisch immer mitgeführt, was die Rücknahme von Änderungen auch ohne netzwerkbasiertes Versionsverwaltungssystem ermöglicht. Zusätzlich stehen natürlich Undo/Redo-Funktionen zur Verfügung.

Modularität
Der Safety Developer nimmt durch seine modulare Projektverwaltung Rücksicht auf die flexiblen Anforderungen in der heutigen Industrie. Teilprojekte, FBD-Netzwerke und auch zusätzliche sichere I/O-Module des Projektes können je nach Ausbaustufe aktiviert werden, Signale können abhängig von der Ausbaustufe unterschiedlich verbunden werden. So lässt sich innerhalb eines Gesamtprojektes jede Ausbaustufe bis zum maximalen Maschinenausbau komplett erstellen und testen. Die Anpassung an den realen Ausbaugrad wird durch die Zusammenstellung der gewünschten Teile erreicht. Eine Inbetriebnahme von einzelnen Maschinenteilen ist auf diesem Weg ebenfalls möglich.



Programmüberprüfung
Das Programm für die Ausführung auf dem Zielsystem wird nicht kompiliert, sondern in ein Script übersetzt, das von der Firmware des Zielsystems redundant geprüft und ausgeführt wird. Der Safety Developer erkennt bei der Erzeugung des Scripts mögliche Fehlerquellen im Code und weist darauf hin.


Fehler
1 Fehler, 11 Warnungen, 0 sonstige

Beschreibung	Ressource	Pfad	Position	Typ
Fehler (1 Element)				
Variable is never written	SLCAPP.fbd	/SLCAPP	ShftBrkMon v1.1.4/n2Act	Compile Problems
Warnungen (11 Elemente)				
A: Ungültige Abarbeitungsreihenfolge	SLCAPP.fbd	/SLCAPP	SafetyWindMesse/DIV2/A	Compile Problems
Variable is not used	GlobalVariabl...	/SLCAPP	10	Compile Problems
Variable is not used	GlobalVariabl...	/SLCAPP	SimMode	Compile Problems
Variable is not used	GlobalVariabl...	/SafetySCT202	+30000	Compile Problems
Variable is not used	GlobalVariabl...	/SafetySCT202	+40000	Compile Problems
Variable is not used	GlobalVariabl...	/SafetySCT202	-30000	Compile Problems
Variable is not used	GlobalVariabl...	/SafetySCT202	-40000	Compile Problems
Variable is not used	GlobalVariabl...	/SafetySCT202	10000	Compile Problems
Variable is not used	GlobalVariabl...	/SafetySCT202	8000	Compile Problems
Variable is not used	RotInc2Deg v...	/SafetySCT202/lib/...	-1	Compile Problems
Variable is not used	RotInc2Deg v...	/SafetySCT202/lib/...	FALSE	Compile Problems

Protokollierung

Die Nachweispflicht wird auf verschiedene Arten unterstützt. Für die Protokollierung der Abnahme kann ein Projektreport generiert werden, der den gesamten Programmcode grafisch darstellt, alle Projekteinstellungen, die Sicherheitsskripte und eigene Abnahmetabellen zur Vor-Ort Inbetriebnahme exportiert. Das fälschungssichere Logbuch des Safety-Controllers protokolliert jede sicherheitsrelevante Änderung am System, wie z.B. den Download eines geänderten Programms. Jeder Zugriff ist somit mit Benutzername, Datum und Uhrzeit nachvollziehbar.

Optional kann das gesamte Projekt auf dem Safety Controller abgelegt, von dort geöffnet und weiterbearbeitet werden. Zu jedem Netzwerk können anwenderspezifische Informationen wie Autor, Versionsgeschichte und weitere Kommentare gespeichert werden.

Hardware-Konfiguration

Zusätzlich zu den Werkzeugen für Variablenauswahl, Programmierung und Protokollierung kann auch die sichere Hardware direkt im Safety Developer konfiguriert werden. Dazu zählt nicht nur die Vergabe von eindeutigen Kanalnamen, sondern vor allem auch die Zuordnung des Controllers zum Projekt, das Hinzufügen von weiteren sicheren I/O-Modulen und die Festlegung von Prüfintervallen für getaktete Leitungen, die dann automatisch von der Hardware auf Kurzschluss, Querschluss und Fremdspannung geprüft werden. Auch die sicherheitstechnisch geforderte, eindeutige Modulidentifizierung, die ein Verwechseln von Modulen nach einem Serviceeinsatz ausschließt, erfolgt direkt im Safety Developer.
Die Kommunikation zwischen Safety Developer und dem Steuerungssystem für Programmdownload, Diagnose und Konfiguration erfolgt komfortabel über die Ethernet-Schnittstelle des M1-Steuerungssystems. Alternativ kann auch über eine serielle RS232-Schnittstelle direkt mit dem Safety Controller kommuniziert werden, was den Einsatz des Safety Controllers auch als Einzellösung ohne umgebendes Steuerungssystem ermöglicht.