Programmierleitfaden Safety für SIMATIC S7-1200/1500

SIMATIC Safety Integrated

https://support.industry.siemens.com/cs/ww/de/view/109750255

Siemens Industry Online Support

Gewährleistung und Haftung

Hinweis

Die Anwendungsbeispiele sind unverbindlich und erheben keinen Anspruch auf Vollständigkeit hinsichtlich Konfiguration und Ausstattung sowie jeglicher Eventualitäten. Die Anwendungsbeispiele stellen keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bei typischen Aufgabenstellungen bieten. Sie sind für den sachgemäßen Betrieb der beschriebenen Produkte selbst verantwortlich. Dieses Anwendungsbeispiel enthebt Sie nicht der Verpflichtung zu sicherem Umgang bei Anwendung, Installation, Betrieb und Wartung. Durch Nutzung dieses Anwendungsbeispiels erkennen Sie an, dass wir über die beschriebene Haftungsregelung hinaus nicht für etwaige Schäden haftbar gemacht werden können. Wir behalten uns das Recht vor, Änderungen an diesem Anwendungsbeispiel jederzeit ohne Ankündigung durchzuführen. Bei Abweichungen zwischen den Vorschlägen in diesem Anwendungsbeispiel und anderen Siemens Publikationen, wie z. B. Katalogen, hat der Inhalt der anderen Dokumentation Vorrang.

Für die in diesem Dokument enthaltenen Informationen übernehmen wir keine Gewähr.
Unsere Haftung, gleich aus welchem Rechtsgrund, für durch die Verwendung der in diesem Anwendungsbeispiel beschriebenen Beispiele, Hinweise, Programme, Projektierungs- und Leistungsdaten usw. verursachte Schäden ist ausgeschlossen, soweit nicht z. B. nach dem Produkthaftungsgesetz in Fällen des Vorsatzes, der groben Fahrlässigkeit, wegen der Verletzung des Lebens, des Körpers oder der Gesundheit, wegen einer Übernahme der Garantie für die Beschaffenheit einer Sache, wegen des arglistigen Verschweigens eines Mangels oder wegen Verletzung wesentlicher Vertragspflichten zwingend gehaftet wird. Der Schadensersatz wegen Verletzung wesentlicher Vertragspflichten ist jedoch auf den vertragstypischen, vorhersehbaren Schaden begrenzt, soweit nicht Vorsatz oder grobe Fahrlässigkeit vorliegt oder wegen der Verletzung des Lebens, des Körpers oder der Gesundheit zwingend gehaftet wird. Eine Änderung der Beweislast zu Ihrem Nachteil ist hiermit nicht verbunden.

Security-hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen nur einen Bestandteil eines solchen Konzepts.

Der Kunde ist dafür verantwortlich, unbefugten Zugriff auf seine Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Systeme, Maschinen und Komponenten sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und entsprechende Schutzmaßnahmen (z. B. Nutzung von Firewalls und Netzwerksegmentierung) ergriffen wurden.

Zusätzlich sollten die Empfehlungen von Siemens zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Industrial Security finden Sie unter http://www.siemens.com/industrialsecurity.

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Aktualisierungen durchzuführen, sobald die entsprechenden Updates zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter http://www.siemens.com/industrialsecurity.

Inhaltsverzeichnis

1 Einleitung

Die neue Steuerungsgeneration SIMATIC S7-1200 und S7-1500 weist eine zeitgemäße Systemarchitektur auf und bietet zusammen mit dem TIA Portal neue und effiziente Möglichkeiten der Programmierung und Projektierung.

Die vielen Möglichkeiten, die STEP 7 bietet, können auch zu negativen Ergebnissen führen, wenn unsauber programmiert wird:

- CPU-Stopps
- Lange Übersetzungsprozesse
- Zusätzliche, umfangreiche Abnahmen

Dieses Dokument gibt Ihnen viele Empfehlungen und Hinweise zur optimalen Projektierung und Programmierung von S7-1200/1500 Steuerungen. Dies hilft Ihnen, eine standardisierte und optimale Programmierung Ihrer Automatisierungslösungen zu erstellen.

Die beschriebenen Beispiele können universell auf den Steuerungen S7-1200 und S7-1500 eingesetzt werden.

Vorteile

Mit der Einhaltung der hier genannten Empfehlungen erzielen Sie viele Vorteile:

- Wiederverwendbarkeit von Programmteilen
- Einfachere Abnahme (Code-Review, Fehlererkennung und -korrektur)
- Höhere Flexibilität bei Programmänderungen
- Reduzierung von Programmierfehlern
- Erhöhte Anlagenverfügbarkeit durch Vermeidung von CPU-Stopps
- Leichtere Lesbarkeit für Dritte
- Verringerte Laufzeit des Sicherheitsprogramms

Hinweis Nicht alle Empfehlungen dieses Dokuments können gleichzeitig angewandt werden. In diesen Fällen müssen Sie als Anwender entscheiden, welcher Empfehlung Sie eine höhere Priorität geben (z. B. Standardisierung oder Laufzeitoptimierung des Sicherheitsprogramms).

Programmierleitfaden und -styleguide

Bei der Programmierung von Sicherheitsprogrammen gelten grundsätzlich dieselben Empfehlungen wie aus dem Programmierleitfaden und dem Programmierstyleguide.

Programmierleitfaden für SIMATIC S7-1200/1500:

https://support.industry.siemens.com/cs/ww/de/view/90885040

Programmierstyleguide für SIMATIC S7-1200/1500:

https://support.industry.siemens.com/cs/ww/de/view/109478084

Leitfaden zur Bibliothekshandhabung im TIA Portal:

https://support.industry.siemens.com/cs/ww/de/view/109747503

Dieses Dokument dient als Ergänzung zu den genannten Dokumenten und behandelt die Besonderheiten bei der Programmierung von Sicherheitsprogrammen mit STEP 7.

2 Fehlersichere Steuerungen projektieren

2.1 Die geeignete F-CPU auswählen

Die Auswahl der F-CPU ist von folgenden Faktoren abhängig:

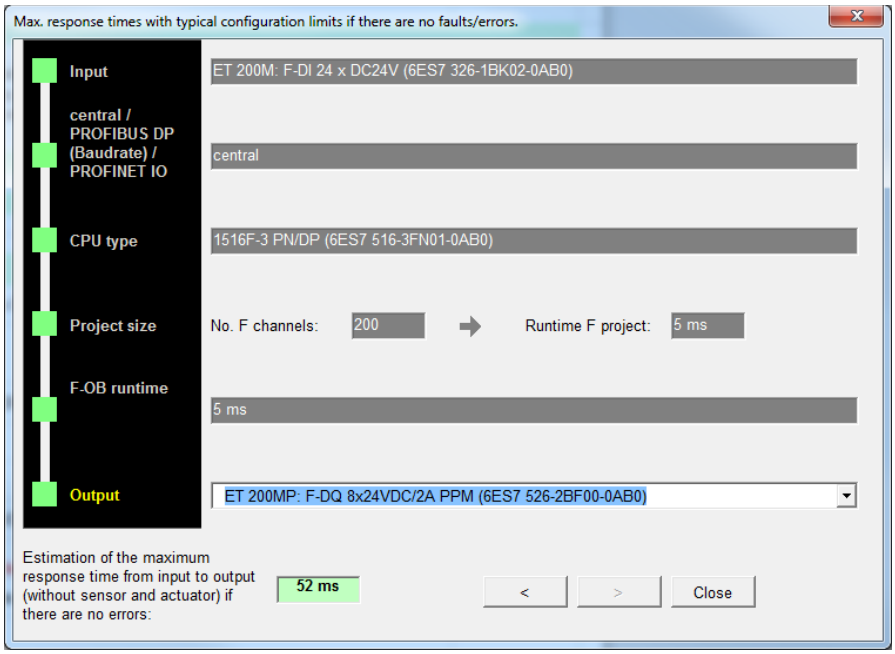
- Laufzeit des Sicherheitsprogramms
- PROFIsafe-Kommunikationszeit
- Reaktionszeit der Sicherheitsfunktion
- Anzahl der benötigten Ein- und Ausgänge
- Anzahl der angebundenen Peripherie

Abschätzung der Reaktionszeit

Wenn Sie bereits eine grobe Vorstellung haben, welches Automatisierungssystem Sie einsetzen möchten, können Sie die Reaktionszeit Ihres Sicherheitsprogramms mit der SIMATIC STEP 7 Reaktionszeittabelle abschätzen oder verschiedene Szenarien durchspielen und so die geeignete F-CPU auswählen:

https://support.industry.siemens.com/cs/ww/de/view/93839056

Abbildung 2-1: Reaktionszeit-Assistent der SIMATIC STEP 7 Reaktionszeittabelle



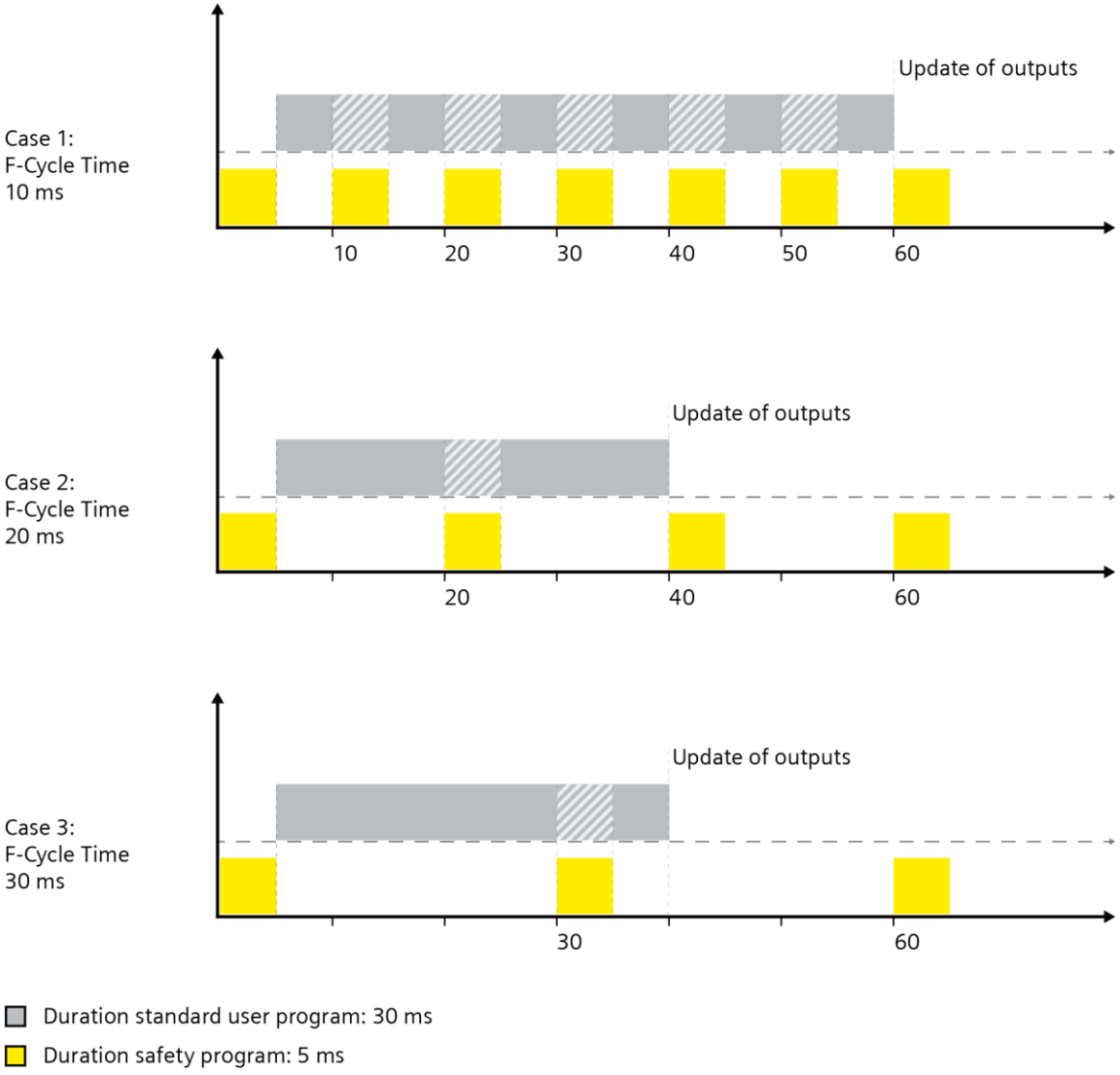
Einfluss der Zykluszeit des Sicherheitsprogramms auf das Standard-Anwenderprogramm

Eine hohe Zykluszeit des Sicherheitsprogramms verlangsamt die Reaktionszeit Ihrer Sicherheitsfunktionen, lässt dafür aber mehr Zeit für die Bearbeitung des Standard-Anwenderprogramms zu.

Eine kurze Zykluszeit des Sicherheitsprogramms erhöht die Reaktionszeit Ihrer Sicherheitsfunktionen, lässt dafür aber weniger Zeit für die Bearbeitung des Standard-Anwenderprogramms zu.

Die folgende Abbildung zeigt den Einfluss der Zykluszeit des Sicherheitsprogramms auf die Zeit, die für die Bearbeitung des Standard-Anwenderprogramms zur Verfügung steht.

Abbildung 2-2: Einfluss der Zykluszeit des Sicherheitsprogramms auf das Standard-Anwenderprogramm



Hinweis

Beachten Sie, dass höherpriore Organisationsbausteine (z. B. Weckalarm-OBs oder Motion Control-OBs) auf dieselbe Weise, wie in Abbildung 2-2 gezeigt, auch das Sicherheitsprogramm unterbrechen können.

Um sicherzustellen, dass das Sicherheitsprogramm nicht unterbrochen wird, können Sie die Prioritäten in den Eigenschaften der jeweiligen OBs anpassen.

ACHTUNG

Die Zykluszeit muss höher als die Bearbeitungsdauer des Sicherheitsprogramms sein.

2.2 PROFIsafe-Adresstypen

Die PROFIsafe-Adresse dient zur eindeutigen Adressierung von F-Peripherie und der Absicherung von Standard-Adressierungsmechanismen wie z. B. IP-Adressen. Dabei wird die Eindeutigkeit bei F-Peripherie des PROFIsafe-Adresstyps 1 und F-Peripherie des PROFIsafe-Adresstyps 2 unterschiedlich definiert.

Tabelle 2-1: Unterschiede PROFIsafe-Adresstypen

PROFIsafe-Adresstyp 1	PROFIsafe-Adresstyp 2
• Die Eindeutigkeit der PROFIsafe-Adresse wird nur durch die F-Zieladresse sichergestellt • Die F-Zieladresse muss netz- und CPU-weit eindeutig sein. • Im Sicherheitsausdruck ist jede F-Zieladresse auf netz- und CPU-weite Eindeutigkeit zu prüfen, indem geprüft wird, dass sich die F-Zieladressbereiche aller F-CPUs nicht überschneiden. • F-Zieladresse und F-Quelladresse gehen in den CRC des Sicherheitsprogramms ein.	• Die Eindeutigkeit der PROFIsafe-Adresse wird durch die Kombination von F-Quelladresse und F-Zieladresse sichergestellt. • Die F-Zieladresse muss CPU-weit eindeutig sein und sich von allen F-Zieladressen des PROFIsafe-Adresstyps 1 im selben Netz unterscheiden. • Die F-Quelladresse, die für die F-Peripherie einer F-CPU verwendet wird, muss netzweit eindeutig sein. • F-Zieladresse und F-Quelladresse gehen in den CRC des Sicherheitsprogramms ein.

Sie müssen sicherstellen, dass jede PROFIsafe-Adresse eindeutig ist.

Durch die immer stärker werdende Vernetzung von Analgen- und Analgenteilen – besonders wenn diese getrennt projektiert werden – müssen Sie die Vergabe der PROFIsafe-Adresse umso genauer planen.

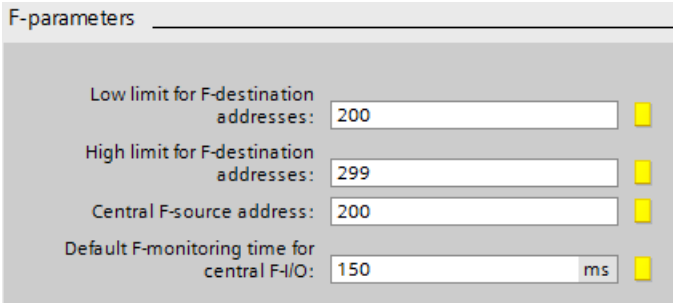
Dabei vereinfacht die Verwendung von F-Peripherie des PROFIsafe-Adresstyps 2 die Handhabung der PROFIsafe-Adressen. Bei gemischten Konstellationen oder reinen Adresstyp 1-Konstellationen ist jedoch mehr Vorsicht walten zu lassen.

Empfehlung

- Betrachten Sie bereits zu Projektbeginn mögliche Kommunikationsbeziehungen und Netzwerktopologien. Leiten Sie mit den Beteiligten Maßnahmen zur Vergabe der PROFIsafe-Adressen ab.
- Vergeben Sie jeweils eigene Adressbereiche für die PROFIsafe-Adresstypen 1 und 2:
  - Weisen Sie F-Peripherie des PROFIsafe-Adresstyps 1 ein niedriges Nummernband zu [1)].
  - Weisen Sie F-Peripherie des PROFIsafe-Adresstyps 2 ein hohes Nummernband zu.
- Definieren Sie die F-Quelladressen für alle F-CPUs grundsätzlich eindeutig. Dies erleichtert die projektübergreifende Arbeit wie auch spätere Erweiterungen.

[1] Den zulässigen Bereich für F-Zieladressen des PROFIsafe-Adresstyps 1 können Sie in den Eigenschaften der CPU definieren.

Abbildung 2-3: Adressbereich für F-Zieladressen festlegen



Weitere Informationen

Weitere Informationen zu PROFIsafe-Adresstypen finden Sie im Siemens Industry Online Support:

Worin unterscheiden sich die PROFIsafe-Adresstypen 1 und 2 in Bezug auf die Eindeutigkeit der PROFIsafe-Adresse?

https://support.industry.siemens.com/cs/ww/de/view/109479905

Wie sind PROFIsafe-Adressen zu vergeben, damit diese netz- und CPU-weit eindeutig sind?

https://support.industry.siemens.com/cs/ww/de/view/109740240

2.3 F-CPU vor unberechtigten Zugriffen schützen

Um unberechtigte Änderungen oder Manipulation eines Sicherheitsprogramms zu verhindern, müssen Sie einen entsprechenden Zugriffsschutz implementieren.

Dies können Sie z. B. durch organisatorische Maßnahmen (z. B. Absperren des Schaltschranks) erreichen.

Ein einfacherer und effektiverer Zugriffschutz lässt sich jedoch durch die Vergabe von Passwörtern erreichen.

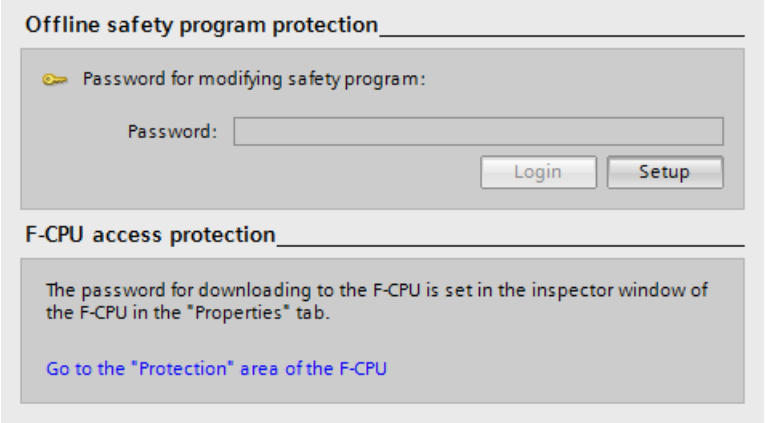
Einen Zugriffsschutz können Sie getrennt für das Sicherheitsprogramm und die F-CPU einrichten.

Zugriffsschutz für das Sicherheitsprogramm

Der Zugriffsschutz für das Sicherheitsprogramm stellt sicher, dass das F-Programm nur von berechtigten Personen geändert wird.

Das Passwort für den Zugriffsschutz des Sicherheitsprogramms legen Sie in der Safety Administration im TIA Portal fest.

Abbildung 2-4: Passwort für das Sicherheitsprogramm festlegen



Nachdem Sie sich mit dem Passwort für das Sicherheitsprogramm angemeldet haben, können Sie die Zugriffsrechte auf das Sicherheitsprogramm folgendermaßen zurücknehmen:

- In der Safety Administration ausloggen
- In der Menüleiste unter "Online > Zugriffsrechte löschen"
- TIA Portal schließen

Hinweis

Zusammenarbeit von Programmierern mit und ohne Berechtigung für das Sicherheitsprogramm

Änderungen an Standard-DBs, auf die vom Sicherheitsprogramm lesend oder schreibend zugegriffen wird, erfordern ein erneutes Übersetzen des Sicherheitsprogramms. Diese Standard-DBs unterliegen nicht der Zugriffsberechtigung für das Sicherheitsprogramm. Deshalb ist für den Datenaustausch zwischen F-Programm und Standardprogramm eine definierte Schnittstelle notwendig, die der Programmierer des Standard-Anwenderprogramms bei seiner Arbeit nicht verändern muss.

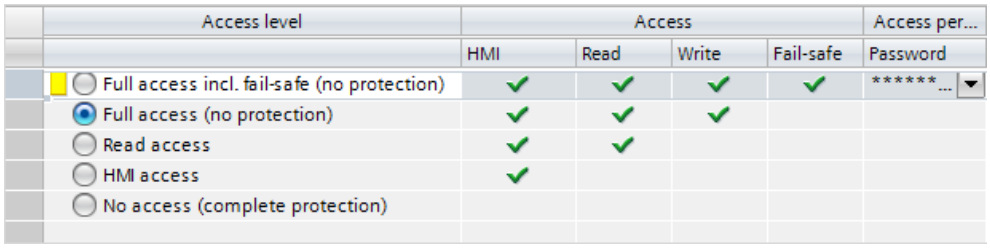
Weitere Informationen zum Datenaustausch finden Sie im Kapitel 3.9.

Zugriffsschutz für die F-CPU

Der Zugriffsschutz für die F-CPU stellt sicher, dass nur berechtigte Personen ein Sicherheitsprogramm ins Gerät laden oder den Sicherheitsbetrieb deaktivieren können.

Das Passwort für die F-CPU legen Sie in den Eigenschaften der CPU fest.

Abbildung 2-5: Passwort für F-CPU festlegen



Der Zugriffsschutz gilt nur für die jeweilige F-CPU. Dieses Passwort dient auch der Identifikation der F-CPU und muss daher netzweit eindeutig sein.

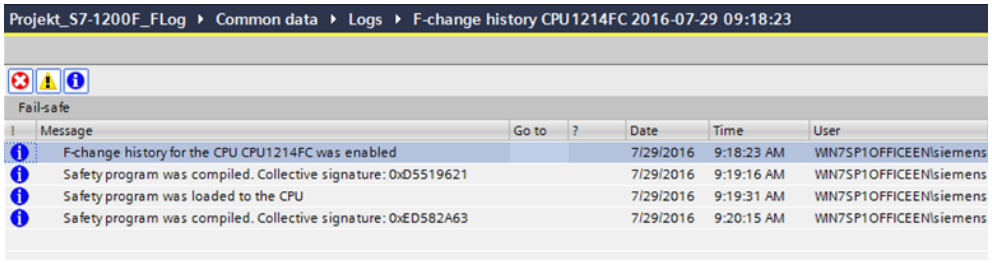
2.4 F-Änderungshistorie

Die F-Änderungshistorie verhält sich wie die Änderungshistorie des Standard-Anwenderprogramms. In der Projektnavigation wird unter "Gemeinsame Daten > Protokolle" ("Common data > Logs") für jede F-CPU eine F-Änderungshistorie angelegt.

Folgendes wird in der F-Änderungshistorie protokolliert:

- F-Gesamtsignatur
- Benutzername
- Compile-Zeitstempel
- Download des Sicherheitsprogramms mit Zeitstempel
- Übersetzte F-Bausteine mit Signatur und Zeitstempel

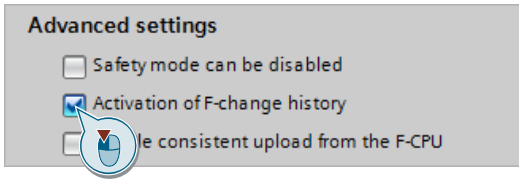
Abbildung 2-6: F-Änderungshistorie



Empfehlung

Aktivieren Sie die Änderungshistorie zu Beginn der Projektierung oder spätestens nach endgültiger Festlegung des projektspezifischen CPU-Namens, da die Änderungshistorie an den CPU-Namen gekoppelt ist.

Abbildung 2-7: F-Änderungshistorie aktivieren



Vorteile

- Sicherstellen, dass die letzte Änderung geladen wurde durch Vergleich von Online- und Offline-Stand des CRC
- Nachverfolgung in Multiuser-Projekten welcher Anwender das Sicherheitsprogramm geändert oder geladen hat.
- Abgleich von Online- und Offline-Stand ohne Online-Verbindung zwischen CPU und PG/PC.

ACHTUNG	Die F-Änderungshistorie dürfen Sie nicht für das Erkennen von Änderungen im Sicherheitsprogramm oder in der Projektierung der F-Peripherie bei der Abnahme von Änderungen verwenden.

2.5 Konsistenter Upload von F-CPUs

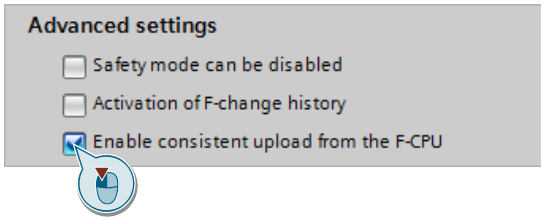
Mit TIA Portal V14 SP1 und höher können Sie fehlersichere SIMATIC S7-1500 CPUs konsistent aus dem Automatisierungssystem ins TIA Portal hochladen.

Empfehlung

Ein Upload aus dem Automatisierungssystem ist nur möglich, wenn das Projekt dafür freigegeben ist.

Aktivieren Sie bei Beginn der Projektierung die Option "Konsistenter Upload" in der Safety Administration im TIA Portal.

Abbildung 2-8: Konsistenten Upload aktivieren



Vorteile

Durch Verzicht auf komplexe "Offline"-Projektverwaltungen, können Sie Fehler vermeiden und den Service-Aufwand weiter reduzieren.

2.6 Know-how-Schutz

Ab STEP 7 Safety V14 können Sie den Know-how-Schutz für fehlersichere Bausteine (FCs und FBs) aktivieren.

Der Know-how-Schutz schützt vor dem Zugriff durch unberechtigte Personen auf bestimmte Programmteile, unabhängig vom Zugriffschutz der F-CPU und des Sicherheitsprogramms. Der Inhalt eines FC oder FB kann ohne Passwort nicht eingesehen oder verändert werden.

Empfehlung

Prüfen Sie während der Projektphase inwieweit es sinnvoll ist, Bausteine eines Sicherheitsprogramms vor dem Zugriff Dritter zu schützen.

Vorteile

- Schutz Ihres Know-hows über den Inhalt der Programmteile.
- Abgenommene Bausteine können nicht verändert werden.

Weitere Informationen

Die nachfolgende Dokumentation bietet eine Anleitung zum Umgang mit dem Know-how-Schutz für unterschiedliche Szenarien:

https://support.industry.siemens.com/cs/ww/de/view/109742314

3 Methoden für die Safety-Programmierung

3.1 Programmstrukturen

3.1.1 Programmstruktur definieren

Empfehlung

- Teilen Sie den Programmcode modular auf, z. B.
  - in Teilbereiche für Erfassen, Auswerten, Reagieren oder
  - nach Anlagenteilen
- Erstellen Sie im Vorfeld eine Spezifikation für jedes Modul (basierend auf den Anforderungen der Risikobeurteilung).
- Vermeiden Sie komplexe Signalpfade.

Vorteile

- Komplexität wird minimiert.
- Programmierfehler werden reduziert.
- Erlaubt den Programmcode ohne Ausführung des Programms (z. B. Code-Review oder PLCSIM) zu analysieren/testen.
- Leichtere Erweiterbarkeit und Vereinfachung der erneuten Abnahme.
- Wiederverwendbarkeit von Programmteilen ohne erneute Abnahme.
- Fertige Programmteile können vorab getestet und abgenommen werden.

Beispiel

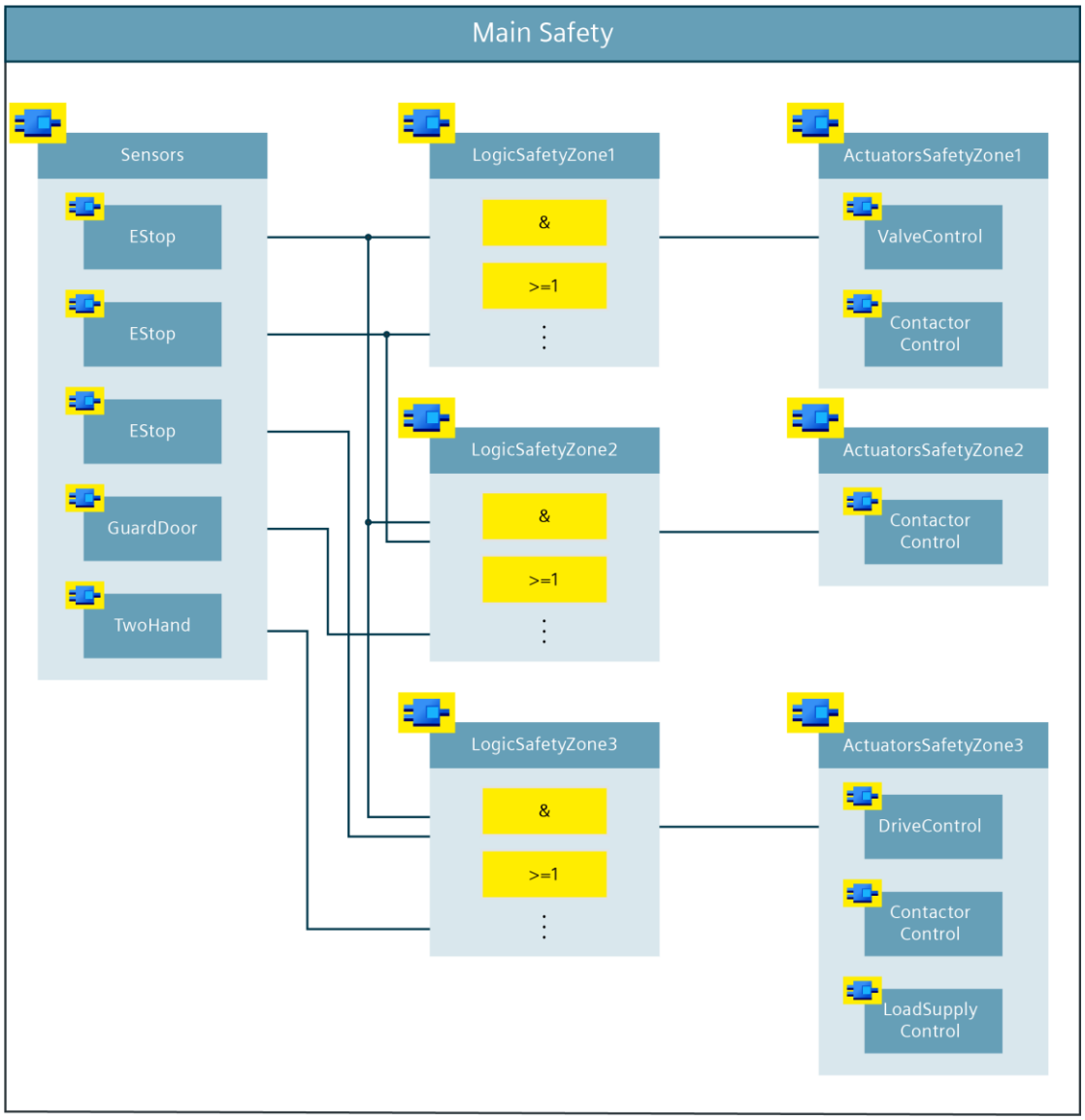
Die folgende Abbildung zeigt eine Sicherheitsapplikation, die in drei Maschinenbereiche (Safety Zones) aufgeteilt ist.

Da die Sensorsignale teilweise bereichsübergreifend verschaltet werden (z. B. global-wirkende Not-Halt-Funktionen), werden sie einem FB "Sensors" gruppiert (eine Aufteilung in physikalische oder logische Bereiche wäre ebenso möglich). Die Auswertung der jeweiligen Sensoren erfolgt über standardisierte Funktionsbausteine (z. B. "GuardDoor").

Auch die Bausteine der Mobile Panel werden hier aufgerufen.

Für jeden Maschinenbereich werden eigene Logik- und Aktor-FBs erstellt. Die Ansteuerung der Aktoren erfolgt über standardisierte Funktionsbausteine (z. B. "ContactorControl").

Abbildung 3-1: Beispiel einer Programmstruktur



Hinweis Die hier dargestellte Strukturierung ist beispielhaft. Je nach Größe und Komplexität des Sicherheitsprogramms kann auch eine andere Aufteilung gewählt werden. In kleineren Applikationen wäre es z. B. auch möglich, die Logik und Aktoransteuerung in einem gemeinsamen Funktionsbaustein zu realisieren.

3.1.2 Aufrufebenen von F-FBs/F-FCs

Die Anzahl der Aufrufebenen für Standard-Anwenderprogramme sind abhängig von der CPU begrenzt. Bei Sicherheitsprogrammen können Sie maximal acht Aufrufebenen verwenden. Ab dieser Grenze erscheint eine Warnung und bei reinen FC- und Multiinstanzaufrufketten eine Fehlermeldung.

Systemanweisungen ("ESTOP1", "SF_DOOR" usw.) zählen nicht zur Anzahl der Aufrufebenen dazu.

Hinweis Funktionen werden auf Systemseite im Absicherungsprogramm als FBs mit Multiinstanzaufruf abgebildet, weshalb auch für FC-Aufrufketten ab acht Aufrufebenen eine Fehlermeldung erscheint.

Die in Abbildung 3-1 dargestellte Programmstruktur zeigt eine Möglichkeit auf, wie die Aufrufebenen relativ flach gehalten werden können, sodass das Sicherheitsprogramm innerhalb der hier spezifizierten Grenzen bleibt.

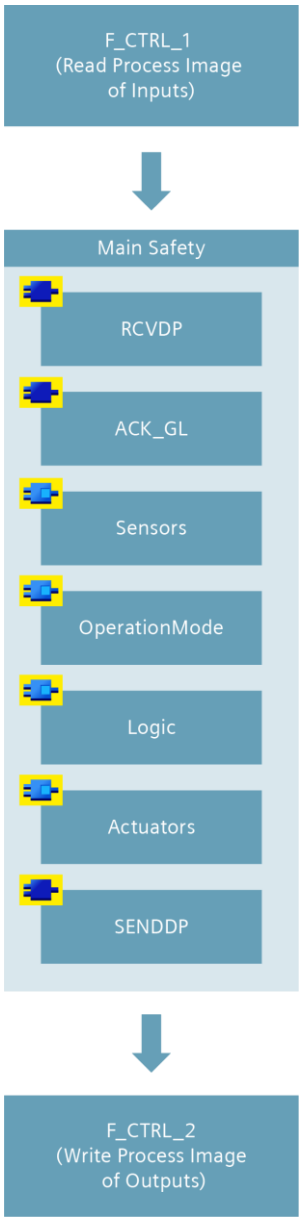
3.1.3 Aufrufreihenfolge der Bausteine im Main Safety

Empfehlung

Rufen Sie Bausteine innerhalb vom Main Safety in folgender Reihenfolge auf:

1. Empfangs-Bausteine von anderen CPUs (F-CPU-F-CPU-Kommunikation)
2. Fehlerquittierung/Wiedereingliederung von F-Modulen/-Kanälen
3. Auswertebaustein der Sensoren
4. Betriebsartenauswertung
5. Logische Verknüpfungen, Berechnungen, Auswertungen usw.
6. Ansteuerbausteine für sichere Aktoren
7. Sende-Bausteine zu anderen CPUs (F-CPU-F-CPU-Kommunikation)

Abbildung 3-2: Aufrufreihenfolge im Main Safety



Vorteile

- CPU arbeitet immer mit den aktuellsten Werten
- Erleichtert die Orientierung im Main Safety

3.1.4 F-geeignete PLC-Datentyp

Auch bei Sicherheitsprogrammen ist es möglich, Daten optimal mit PLC-Datentypen zu strukturieren.

F-geeignete PLC-Datentypen haben folgende Eigenschaften:

- F-geeignete PLC-Datentypen werden genauso wie PLC-Datentypen deklariert und verwendet.
- In F-geeigneten PLC-Datentypen können alle Datentypen verwendet werden, die im Sicherheitsprogramm erlaubt sind.
- Die Verschachtelung von F-geeigneten PLC-Datentypen innerhalb anderer F-geeigneter PLC-Datentypen wird nicht unterstützt.
- F-geeignete PLC-Datentypen können sowohl im Sicherheitsprogramm als auch im Standard-Anwenderprogramm eingesetzt werden.

Empfehlung

- Legen Sie F-geeignete PLC-Datentypen an, um auch im Sicherheitsprogramm Daten zu strukturieren.
- Verwenden Sie F-geeignete PLC-Datentypen, um große Anzahlen an Variablen an Bausteine zu übergeben.
- Nutzen Sie für den Zugriff auf E/A-Bereiche F-geeignete PLC-Datentypen.

  Folgende Regeln müssen dabei beachtet werden:

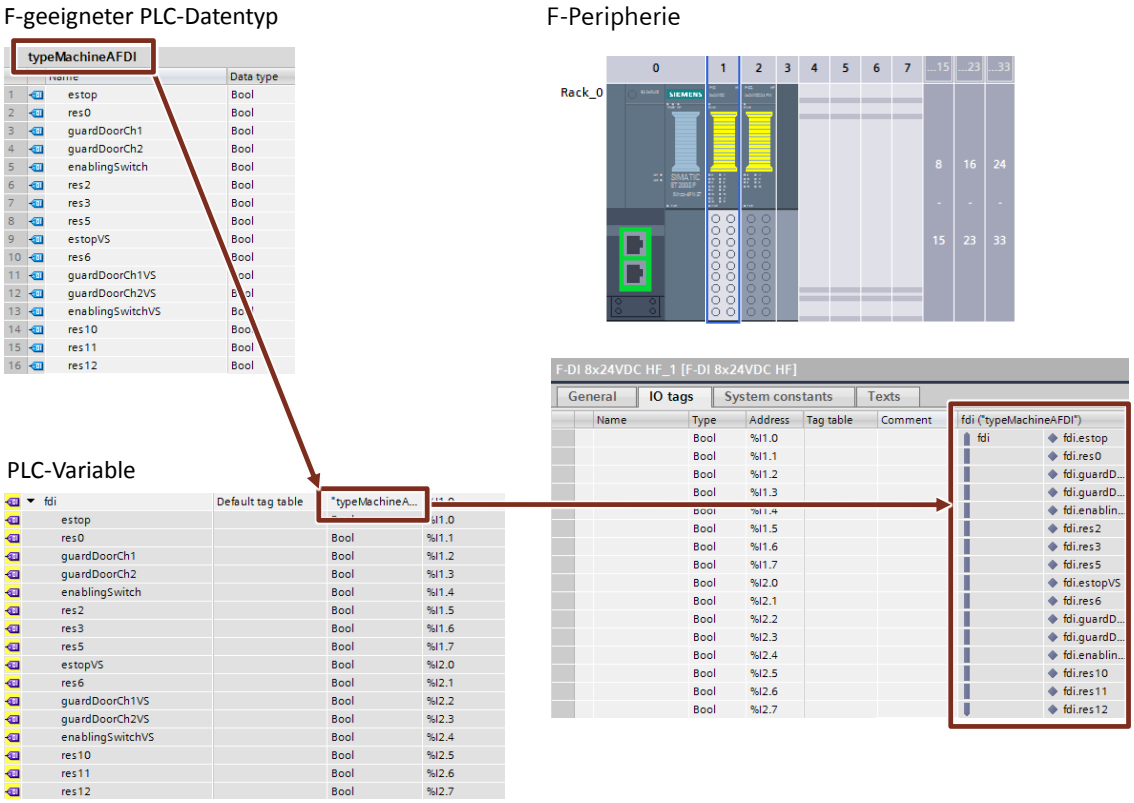
  - Die Struktur der Variablen des F-geeigneten PLC-Datentyps muss mit der Kanalstruktur der F-Peripherie übereinstimmen.
  - Ein F-geeignete PLC-Datentyp für eine F-Peripherie mit 8 Kanälen ist z. B.:
    - 8 BOOL-Variablen (Kanalwert) oder
    - 16 BOOL-Variablen (Kanalwert + Wertstatus)
  - Zugriffe auf F-Peripherie sind nur für aktivierte Kanäle erlaubt. Bei der Parametrierung einer 1oo2 (2v2)-Auswertung wird immer der höherwertige Kanal deaktiviert.

Vorteile

Eine Änderung in einem PLC-Datentyp wird an allen Verwendungsstellen im Anwenderprogramm automatisch aktualisiert.

Beispiel

Abbildung 3-3: Zugriff auf E/A-Bereiche mit F-geeigneten PLC-Datentypen



3.2 Bausteininformationen und Kommentare

Allgemein

In SIMATIC Safety stehen Ihnen die Programmiersprachen Funktionsplan (FUP) und Kontaktplan (KOP) zur Verfügung. Beide Sprachen bieten die Möglichkeit, Block- und Netzwerkkommentare zu hinterlegen.

Kommentare haben keinen Einfluss auf die Signatur von F-FBs/F-FCs und können daher auch nach der Abnahme noch bearbeitet werden.

Empfehlung

Tragen Sie formale Informationen zum Baustein mithilfe der folgenden Schablone in den Blockkommentar Ihres Bausteins ein.

Wenn Sie in einem F-FB Diagnosefunktionen umsetzen, die für den PL bzw. SILCL eines anderen Teilsystems (Erfassen oder Auswerten) relevant sind, tragen Sie die normativen Parameter wie PL bzw. SILCL und Kategorie (nach ISO 13849-1), DC-Maßnahmen, CCF-Maßnahmen usw. ebenfalls in den Blockkommentar ein.

Tragen Sie nach erfolgreicher Abnahme des Bausteins die Signatur ebenfalls in den Blockkommentar ein. Dies erleichtert die Nachverfolgbarkeit bei funktionalen Änderungen des Bausteins.


```
//===========================================================================
// Company
//---------------------------------------------------------------------------
// Library: (that the source is dedicated to)
// Tested with: (test system with FW version)
// Engineering: TIA Portal (SW version)
// Restrictions: (OB types, etc.)
// Requirements: (hardware, technological package, memory needed, etc.)
// Functionality: (that is implemented in the block)
//---------------------------------------------------------------------------
// Reference to Safety Requirement Specification:
// Safety related information: (SIL/PL (Cat.), DC, methods against CFF for connected
subsystems)
//---------------------------------------------------------------------------
// Change log table:
// Version   Date          Signature     Expert in charge   Changes applied
// 01.00.00  (dd.mm.yyyy)  (Block CRC)   (Name of expert)   First released version
//===========================================================================
```

3.3 Funktionale Bezeichner von Variablen

Bei Safety wird sprachlich oftmals von Abschaltungen oder Abschaltsignalen gesprochen. In der Praxis wird auch die Beschreibung einer Sicherheitsfunktion in diesem Wortlaut ausgeführt:

"Wird eine Schutztür geöffnet, muss der Antrieb XY sicher abgeschaltet werden."

Bei der technischen Realisierung als Sicherheitsprogramm werden in der Regel jedoch Freigabesignale programmiert. Dies liegt daran, dass Sicherheitsverschaltungen nach dem Ruhestrom-Prinzip ausgelegt werden.

Wenn zum Beispiel eine Schutztür geschlossen ist, gibt sie die Freigabe, einen sicheren Aktor einzuschalten.

Empfehlung

Legen Sie vor Projektbeginn eine einheitliche Bezeichnung der Variablen mit entsprechenden Suffixen fest. Der Bezeichner gibt den Sinn und Zweck der Variablen im Kontext des Quellcodes wieder.

Wählen Sie den Bezeichner der Variablen so, dass er den logischen "1"-Zustand ("true") wiederspiegelt.

Zum Beispiel "maintDoorEnable" oder "conveyorSafetyRelease".

Hinweis Beachten Sie, dass die standardisierten Bezeichnungen der Antriebsfunktionen (z. B. STO und SLS) nach IEC61800-5-2 nicht der obigen Empfehlung entsprechen.

3.4 True & False

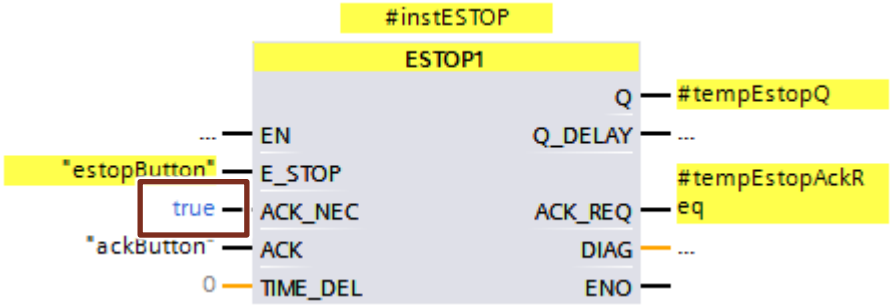
Die Verwendung von "TRUE"- und "FALSE"-Signalen in Sicherheitsprogrammen, kann in zwei Anwendungsfälle unterschieden werden:

- Aktualparameter an Bausteinen
- Zuweisungen an Operationen

Aktualparameter an Bausteinen

Bei S7-1200/1500 Steuerungen können Sie zur Verschaltung von Formalparametern bei Bausteinaufrufen im Sicherheitsprogramm die boolschen Konstanten "FALSE" für "0" und "TRUE" für "1" als Aktualparameter verwenden. Es wird an den Formalparameter nur das Stichwort "FALSE" oder "TRUE" geschrieben.

Abbildung 3-4: "TRUE" bzw. "FALSE"-Signale als Aktualparameter



Zuweisungen an Operationen

Um "TRUE" oder "FALSE"-Signale für Operationen zu erzeugen, gehen Sie diese wie folgt vor:

1. Erstellen Sie zwei statische Variablen "statTrue" und "statFalse" vom Datentyp BOOL.
2. Geben Sie der Variable "statTrue" den Defaultwert "true”.
3. Geben Sie der Variable "statFalse" den Defaultwert "false”.

Die Variablen können Sie im kompletten Funktionsbaustein als "TRUE"- und "FALSE"-Signale lesend einsetzen.

Abbildung 3-5: Deklaration von "TRUE"- und "FALSE"-Signalen

	Name	Data type	Default value	Retain
	Static			
	statTrue	Bool	true	Non-retain
	statFalse	Bool	false	Non-retain

3.5 Bausteine standardisieren

Neben der eigentlichen Auswertung eines Sensors bzw. der Ansteuerung eines Aktors sind häufig dieselben Aufbereitungen von Ein- und Ausgangsparametern notwendig (z. B. Flankenauswertung, Zeitfunktionen, Quittierung usw.).

Hierfür eignet es sich, modulare Bausteine zu erstellen und wiederzuverwenden.

Siemens bietet dafür im Industry Online Support Bausteinbibliotheken an, die Sie in Ihrem Projekt einsetzen können, z. B. "LDrvSafe":

https://support.industry.siemens.com/cs/ww/de/view/109485794

Empfehlung

Erstellen Sie modulare Bausteinen, die Sie wiederverwenden können:

- Bausteine für typische fehlersichere Sensoren
- Bausteine für typische fehlersichere Aktoren
- Bausteine für oft genutzte Funktionen (z. B. Reintegration, Betriebsart)

Vorteile

- Wiederverwendete Bausteine müssen nur einmal abgenommen werden
- Schnellere Programmierung weiterer Funktionen und Projekte
- Versionierung mit dem TIA Portal-Bibliothekskonzept möglich
- Standardisierung von Formalparametern über Projekte und Programmierer hinweg und dadurch leichte Lesbarkeit und Prüfbarkeit

Hinweis Nachfolgende Programmierung der Bausteine sind Beispiele. Die tatsächliche Funktion ist abhängig von der Risikobeurteilung der Applikation bzw. den Anforderungen des Projekts.

3.5.1 Sensorauswertung standardisieren

Empfehlung

Erstellen Sie einen eigenen Funktionsbaustein für jeden Sensortyp (z. B. Not-Halt-Befehlsgerät, Schutztür, Lichtvorhang usw.), der die Auswertung des Sensors und notwendige Hilfsfunktionen bündelt. Verwenden Sie diesen Sensorbaustein für weitere Sensoren desselben Typs.

Erstellen Sie F-Datentypen für komplexe Sensoren.

Hilfsfunktionen für einen sicheren Sensor umfassen zum Beispiel folgendes:

- Rückstellen
- Wiederanlaufsperre
- Zeitfunktionen
- Flankenauswertung
- Anlauftest
- Bereitstellung von Diagnoseinformationen

Abbildung 3-6: Sensorauswertung standardisieren

3.5.2 Aktoransteuerung standardisieren

Empfehlung

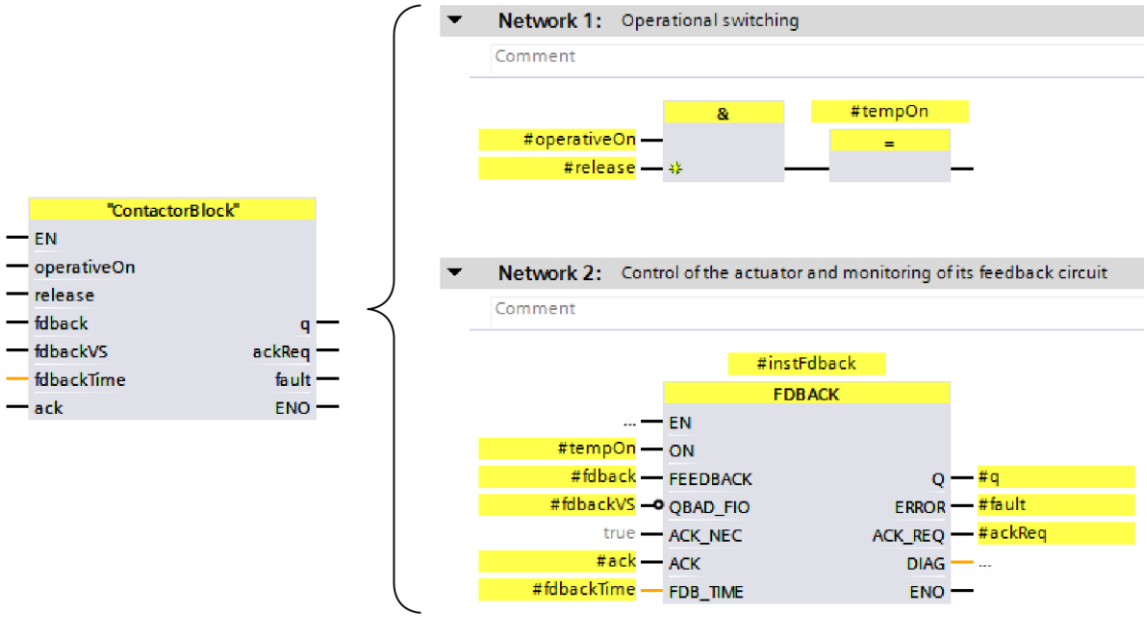
Erstellen Sie einen eigenen Funktionsbaustein für jeden Aktortyp (z. B. Schütze, Ventile, Antriebe usw.), der die Ansteuerung des Aktors und die notwendigen Hilfsfunktionen bündelt. Verwenden Sie diesen Aktorbaustein für weitere Aktoren desselben Typs.

Erstellen Sie F-Datentypen für komplexe Aktoren.

Hilfsfunktionen für einen sicheren Aktor umfassen zum Beispiel folgendes:

- Rückführkreisüberwachung
- Fehlerquittierung
- Flankenauswertung
- Zeitfunktionen
- Betriebsmäßiges Schalten
- Bereitstellung von Diagnoseinformationen

Abbildung 3-7: Aktoransteuerung standardisieren



3.6 Logische Verknüpfungen programmieren

Aufgabe der Bausteine

- Erzeugung von Freigabesignalen zur Ansteuerung der sicherheitsgerichteten Aktoren basierend auf den relevanten Sicherheitsfunktionen
- Verknüpfung der Sensorfreigaben, Betriebsartenfreigaben usw. mit den Ansteuersignalen der Aktoren

Empfehlung

- Verwenden Sie vorrangig UND- und ODER-Logikelemente
- Reduzieren Sie den Einsatz von SR-Bausteinen auf ein Minimum
- Vermeiden Sie Sprünge in binärer Logik

3.7 Betriebsartenabhängige Sicherheitsfunktionen programmieren

Empfehlung

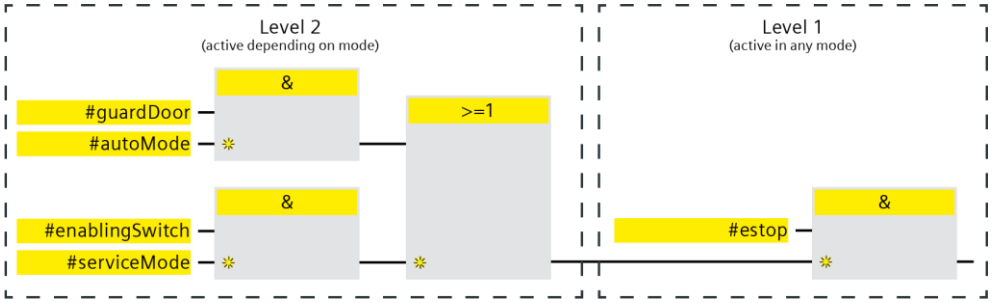
Teilen Sie die Logik in unterschiedliche Level auf (vgl. IEC 62061):

- Level 1: alle Sicherheitsfunktionen, die von Betriebsarten bzw. Anlagenzuständen unabhängig sind.
  - Logische UND-Verknüpfungen aller Sicherheitsfunktionen, die immer aktiv sind.
  - Typischerweise Not-Halt-Einrichtungen.
- Level 2: alle Sicherheitsfunktionen, die betriebsartenabhängig sind.
  - Logische ODER-Verknüpfung der Sicherheitsfunktionen, die nur in bestimmten Betriebsarten wirken.
  - Z. B. Schutztüren im Automatikbetrieb, abwechselnd mit Zustimmtastern im Servicebetrieb.

Beispiel

An einer Maschine sind drei Sicherheitsfunktionen realisiert, wobei die Not-Halt-Funktion "estop" in jeder Betriebsart wirkt und die Schutztürüberwachung "guardDoor" und die Zustimmfunktion "enablingSwitch" wirken jeweils nur in einer Betriebsart.

Abbildung 3-8: Betriebsarten programmieren



3.8 Anbindung von Global-Daten

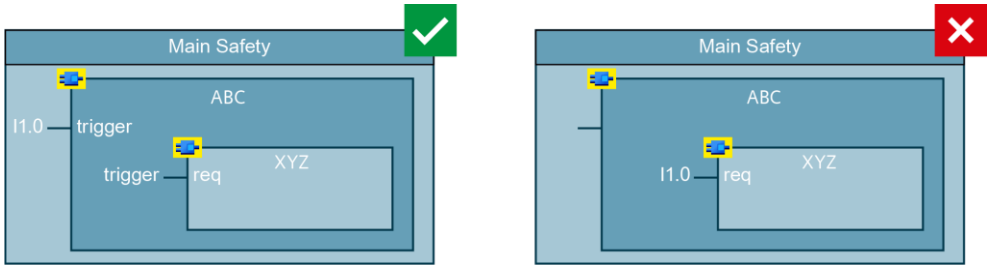
Empfehlung

- Verbinden Sie Global-Daten (Eingänge, Ausgänge, Datenbausteine) in der höchsten Ebene der Baustein-Hierarchie (Main Safety).
- Verwenden Sie die Bausteinschnittstellen, um Signale an unterlagerte Ebenen weiterzugeben.

Vorteile

- Modulares Bausteinkonzept
- Programmteile können ohne Anpassungen in anderen Projekten wiederverwendet werden
- Programmierfehler werden reduziert
- Das Gesamtprogramm wird leichter lesbar, da die generelle Funktion eines Bausteins bereits anhand der Schnittstellen abgeschätzt werden kann.

Abbildung 3-9: Anbindung von Global-Daten



3.9 Datenaustausch zwischen Standard-Anwenderprogramm und Sicherheitsprogramm

Prinzipiell hat das Sicherheitsprogramm die Aufgabe alle Funktionen auszuführen, die eine risikomindernde Maßnahme darstellen. Alle anderen betrieblichen Funktionen, wie auch Funktionen zur Bedienung und Wartung, gehören in das Standard-Anwenderprogramm.

Da in der Praxis auch im Sicherheitsprogramm Informationen für das Diagnose- und Meldekonzept anfallen und auch betriebliche Informationen für das Sicherheitsprogramm relevant sind, können beide Programmteile nicht komplett getrennt werden.

Um nicht-sicherheitsrelevante Funktionen ins Standard-Anwenderprogramm auslagern zu können, müssen Sie eine Schnittstelle definieren. Globale Datenbausteine sind dafür am besten geeignet.

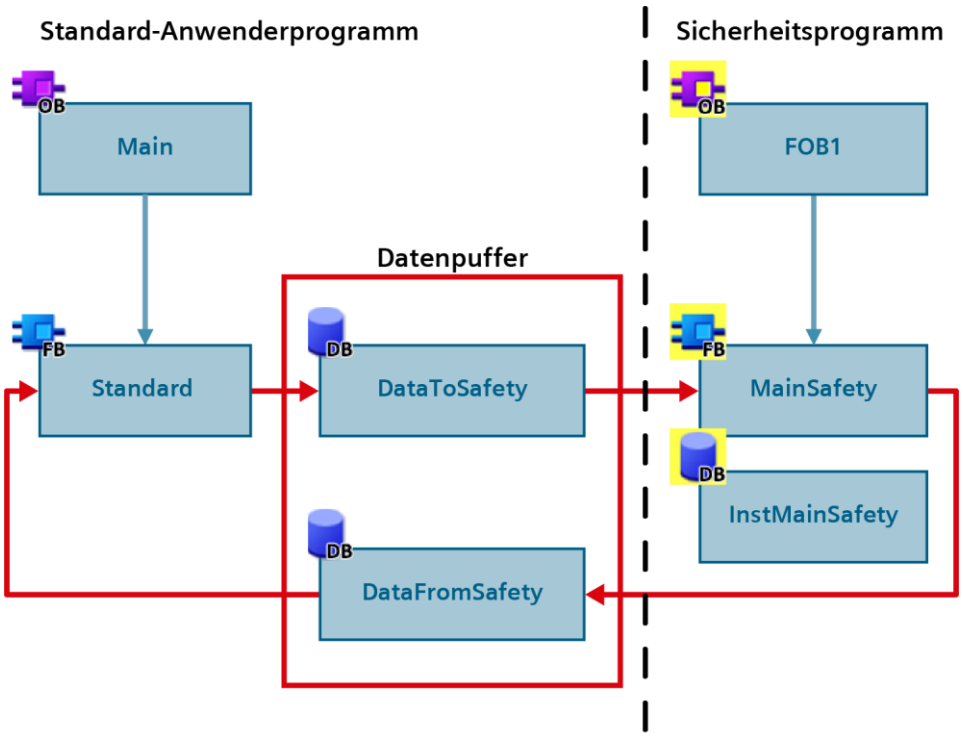
Empfehlung

Verwenden Sie globale Standard-Datenbausteine, um Daten zwischen dem Standard-Anwenderprogramm und dem Sicherheitsprogramm auszutauschen.

Um eine gute Übersicht zu bewahren, welcher Programmteil liest und welcher schreibt, empfiehlt es sich, zwei Datenbausteine für die beiden Richtungen anzulegen.

In den Datenbausteinen sollten Sie keine weiteren Informationen (z. B. Diagnosedaten aus dem Standard-Anwenderprogramm) abgelegen, da jede Änderung des Datenbausteins eine Änderung des Sicherheitsprogramms nach sich zieht.

Abbildung 3-10: Datenaustausch zwischen Standard-Anwenderprogramm und Sicherheitsprogramm



Vorteile

- Schlanke F-Ablaufgruppe
- Bessere Übersicht über die ausgetauschten Daten
- Änderungen des Diagnose- oder Meldekonzepts im Standard-Anwenderprogramm haben keine Auswirkung auf die Signatur des Sicherheitsprogramms
- Minimiertes Risiko von Stillständen wegen Datenverfälschung aufgrund schreibender Zugriffe ins Sicherheitsprogramm
- Vereinfachte Typisierung von F-Bausteinen
- Änderungen am Standard-Anwenderprogramm können ohne CPU-Stopp geladen werden
- Standard-Anwenderprogramm und Sicherheitsprogramm können unabhängig voneinander erstellt werden, wenn Schnittstellen vorher definiert wurden

3.9.1 Diagnose- und Meldeinformationen aus dem Sicherheitsprogramm lesen

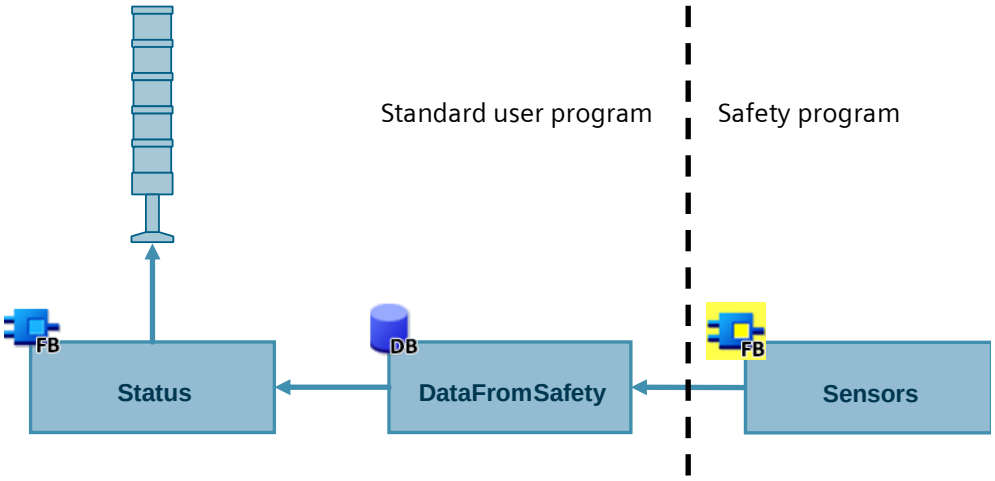
Eine häufige Anwendung für den Datenaustausch zwischen Standard-Anwenderprogramm und Sicherheitsprogramm ist die Visualisierung von Diagnose- und Meldeinformationen, z. B.:

- Quittieranforderungen von Fehlern
- Rückstellanforderungen von Sicherheitsfunktionen
- Fehlermeldungen
- Zustände von Sicherheitsfunktionen

Übergeben Sie die "Rohdaten" aus dem Sicherheitsprogramm. Die logische Verknüpfung erfolgt dann im Standard-Anwenderprogramm. Dies hat den Vorteil, dass das Sicherheitsprogramm schlank gehalten wird und unabhängig von Änderungen im Standard-Anwenderprogramm ist. Kleine Änderungen im Nachhinein (z. B. Änderungen in der Ansteuerung eines Leuchtmelders) werden im Standard-Anwenderprogramm durchgeführt. Abgenommene F-Bausteine werden dadurch nicht verändert.

Wenn Sie eine große Anzahl an Diagnosedaten aus dem Sicherheitsprogramm übergeben, erstellen Sie dafür einen F-Datentyp. Eine Variable mit selbst definiertem Datentyp hält die Bausteinschnittstelle kompakt und übersichtlich. Bei immer ähnlich zu übergebenden Daten ist eine Standarisierung dieser F-Datentypen über alle F-Funktionsbausteine empfehlenswert.

Abbildung 3-11: Diagnose- und Meldeinformationen aus dem Sicherheitsprogramm lesen



3.9.2 Betriebliche Informationen an Sicherheitsprogramm übergeben

In vielen Applikationen ist es unerlässlich bestimmte nicht-sicherheitsgerichtete Verknüpfungsergebnisse vom Standard-Anwenderprogramm an das Sicherheitsprogramm zu übergeben. Typischerweise sind dies betriebliche Einschaltbedingungen (z. B. wenn ein Motorstarter betrieblich und fehlersicher geschaltet wird) oder Maschinenstati zur Betriebsartenvorwahl.

Bereiten Sie dabei die Daten soweit wie möglich im Standard-Anwenderprogramm vor. Je mehr nicht-sicherheitsrelevante Logik im Standard-Anwenderprogramm realisiert wird, desto einfacher lassen sich Änderungen an der prozessrelevanten Logik realisieren.

3.9.3 Nicht-sicheren Eingänge im Sicherheitsprogramm verwenden

Das Einlesen von Standard-Eingängen, die direkt im Sicherheitsprogramm benötigt werden, müssen direkt im Sicherheitsprogramm gelesen werden. Ein "Umweg" über das Standard-Anwenderprogramm ist zu vermeiden.

Hintergrund dafür ist, dass auch nicht-sicherheitsgerichtete Signale in die systematische Integrität der Applikation eingehen. Typische Beispiel dafür sind Quittier-/Rückstelltaster oder Betriebsartenwahlschalter. Welcher Taster welche Sicherheitsfunktion rückstellen darf, ist ein direktes Ergebnis der Risikobewertung. Daher muss eine Änderung der Befehlsgeräte einen Einfluss auf die Signatur haben und darf nur in Begleitung einer Neubeurteilung und Änderungsabnahme einhergehen.

ACHTUNG	Die Bewertung, welche Signale Einfluss auf die systematische Integrität einer Applikation haben und abhängig davon in Standard-Anwenderprogramm oder Sicherheitsprogramm ausgewertet werden, ist abhängig von der Risikobeurteilung einer Applikation.

3.9.4 HMI-Signale ans Sicherheitsprogramm übergeben

Human Machine Interfaces (HMI) sind komfortable und unverzichtbare Komponenten im Alltag eines Maschinenbetreibers. Um diesen Komfort beim Bedienen und Beobachten von Prozessen und Anlagen auch in sicherheitsgerichteten Anwendungen zu nutzen, sind zusätzliche Maßnahmen notwendig.

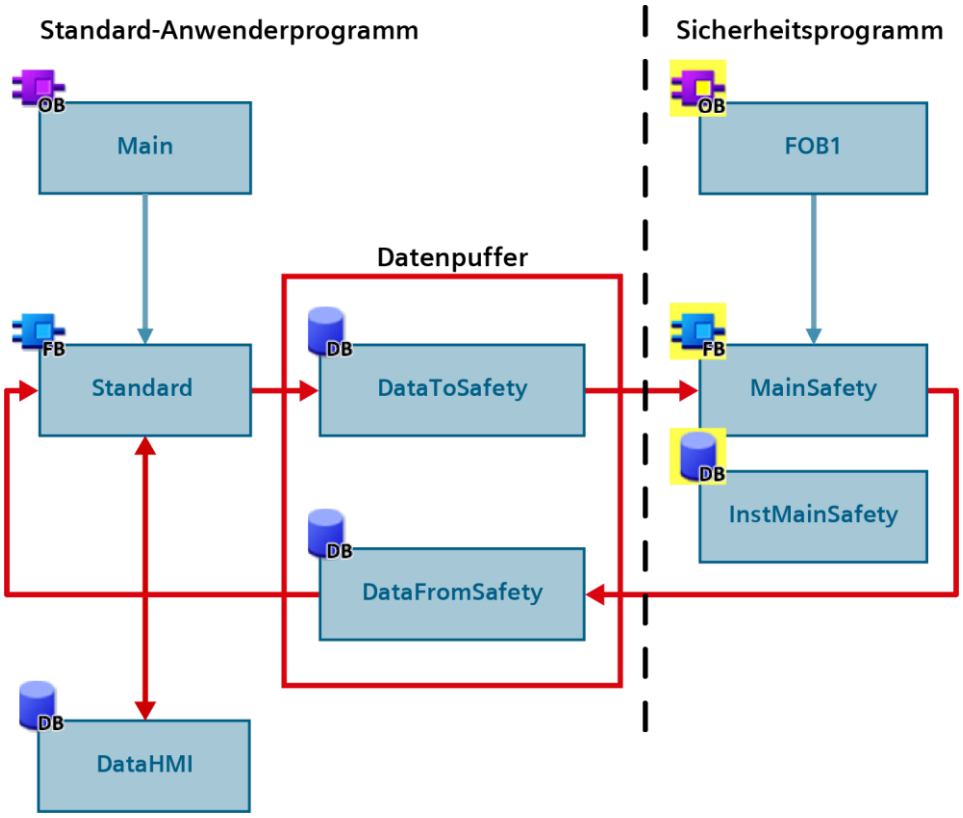
Das Schreiben von Variablen aus dem HMI in das Sicherheitsprogramm ist aus folgenden Gründen problematisch:

- Signale aus dem HMI Panel sind nicht sicherheitsgerichtet und werden nicht überwacht. Ein Fehler kann dazu führen, dass sicherheitsgerichtete Werte unzulässig verändert werden und es dadurch zu einer Risikoerhöhung kommt.
- Die Kommunikation zwischen HMI und CPU erfolgt azyklisch. Daher kann es vorkommen, dass der Schreibzugriff des HMI während der Abarbeitung des Sicherheitsprogramms erfolgt.
  Der erste Programmdurchlauf arbeitet dann noch mit dem ursprünglichen Wert. Das codierte Anwenderprogramm nutzt den zwischenzeitlich aktualisierten Wert. Das führt zu einer Datenverfälschung im Sicherheitsprogramm und folglich zu einem Stopp der CPU (siehe Kapitel 5).

Empfehlung

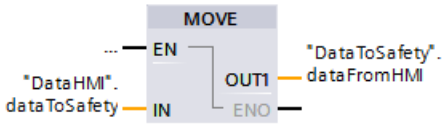
Verwenden Sie einen weiteren Datenbaustein für die Kommunikation zum HMI und kopieren Sie die sicherheitsrelevanten Daten im Standard-Anwenderprogramm in den Datenpuffer.

Abbildung 3-12: Datenaustausch zwischen HMI und Sicherheitsprogramm



Legen Sie einen Datentyp für die Daten vom HMI zum Sicherheitsprogramm an. Diesen Datentyp verwenden Sie in den HMI-Variablen, im Datenpuffer zum Sicherheitsprogramm und im Standard-Anwenderprogramm, wo die Daten umkopiert werden.
Um weitere Variablen hinzuzufügen, die vom HMI ins Sicherheitsprogramm geschrieben werden sollen, passen Sie lediglich den Datentyp an.

Abbildung 3-13: Kopiervorgang der Daten vom HMI zum Sicherheitsprogramm im Standard-Anwenderprogramm



Signale sicher übertragen

Die Kommunikation zwischen HMI und CPU ist nicht sicher. Um sicherheitsgerichtete Daten zu übertragen, sind Maßnahmen notwendig, die die sichere Übertragung gewährleisten.

Dieses Anwendungsbeispiel zeigt Ihnen ein geeignetes Sicherheitskonzept:

https://support.industry.siemens.com/cs/ww/de/view/67634251

Sicherheitsfunktionen rückstellen

Für das Rückstellen von Sicherheitsfunktionen oder das Quittieren von Fehlern über ein HMI stellt TIA Portal den Systembaustein "ACK_OP" bereit.

Eine Quittierung besteht aus zwei Schritten:

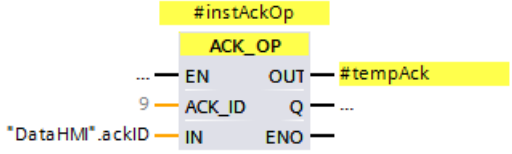
1. Wechsel des Durchgangs IN für genau einen Zyklus auf den Wert "6".
2. Wechsel des Durchgangs IN für genau einen Zyklus auf den Wert am Eingang "ACK_ID" innerhalb einer Minute.

Dieser Systembaustein stellt eine Ausnahme zum empfohlenen Datenaustausch dar.

Der Systembaustein setzt in jedem Zyklus den InOut-Parameter "IN" auf "0" zurück. Werden die Daten vom HMI im Standard-Anwenderprogramm umkopiert, wird die "0" in jedem Zyklus mit dem Wert aus dem HMI überschrieben und die Bedingung, dass die Werte für genau einen Zyklus anstehen, ist nicht erfüllt.

Beschreiben Sie die Variable am Eingang "IN" daher direkt vom HMI aus und setzen Sie die Priorität des Sicherheitsprogramms höher als die der Kommunikation, um eine mögliche Datenverfälschung zu vermeiden.

Abbildung 3-14: Systembaustein "ACK_OP"



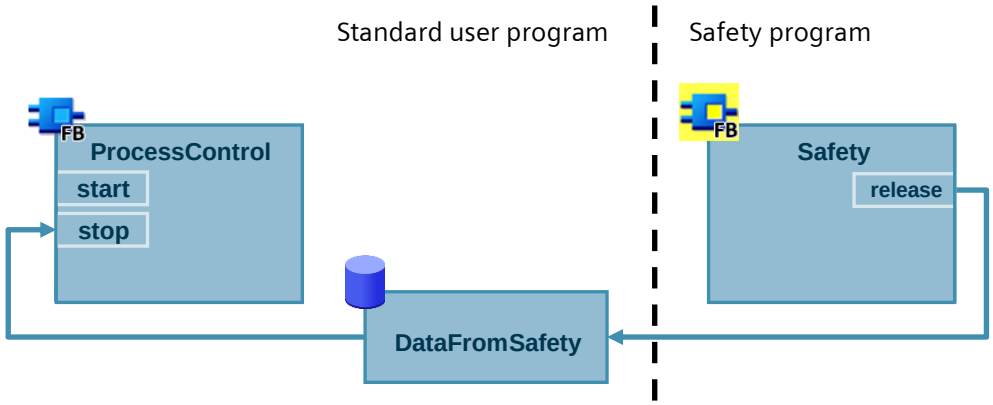
3.10 Betriebsmäßiges Schalten zurücksetzen

Sichere Aktoren werden oft auch für betriebsmäßiges Schalten verwendet. Die einschlägigen Sicherheitsnormen fordern, dass ein Rückstellen der Sicherheitsfunktion keinen Wiederanlauf der Maschine auslöst. Beim Auslösen der Sicherheitsfunktion muss daher das betriebsmäßige Schalten zurückgesetzt und ein erneutes Einschaltsignal erforderlich werden.

Empfehlung

- Verriegeln Sie die Prozesssteuerung im Standard-Anwenderprogramm mit dem Freigabesignal aus dem Sicherheitsprogramm. Eine sichere Abschaltung setzt dadurch auch die Prozesssteuerung zurück.
- Übergeben Sie das Freigabesignal aus dem Sicherheitsprogramm über einen globalen Datenbaustein (siehe auch Kapitel 3.9).

Abbildung 3-15: Prozesssteuerung mit dem Freigabesignal verriegeln



3.11 Wiedereingliederung von fehlersicheren Peripheriemodulen/-kanälen

Erkennt die F-CPU einen sicherheitsrelevanten Fehler, passiviert sie den betroffenen fehlersicheren Kanal bzw. das gesamte Modul. Nachdem der Fehler behoben wurde, muss der passivierte Kanal wiedereingegliedert (depassiviert) werden.

Solange ein Kanal passiviert ist, arbeitet er mit Ersatzwerten. Ein Eingang liefert den Ersatzwert "0" an das Prozessabbild. Ein Ausgang wird mit dem Ersatzwert "0" beschaltet, unabhängig davon ob das Programm den Ausgang ansteuert oder nicht.

3.11.1 Passivierte Module/Kanäle auswerten

Allgemein

Ob ein Kanal passiviert ist, können Sie folgendermaßen auswerten:

- Wertstatus des Kanals ist "false"
- Variable "QBAD" des F-Peripherie-Datenbausteins des Moduls ist "true"
- LEDs des Kanals und des Moduls leuchten rot
- Eintrag im Diagnosepuffer

Die Wiedereingliederung kann entweder manuell oder automatisch erfolgen. Abhängig von der Risikobeurteilung legen Sie das Quittierverhalten fest.

Nachdem ein Fehler behoben wurde, wird Ihnen die Quittierbereitschaft folgendermaßen angezeigt:

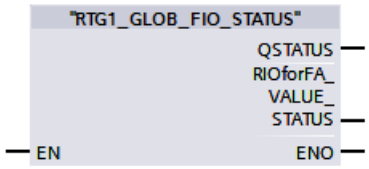
- Variable "ACK_REQ" des F-Peripherie-Datenbausteins des Moduls ist "true"
- LEDs des Kanals und des Moduls blinken abwechselnd rot und gün

Status der F-Peripherien/-Kanäle global auswerten

Ab STEP 7 V14 SP1 können Sie einen Baustein vom System generieren lassen, um den Status aller F-Peripherien/-Kanäle einer F-Ablaufgruppe global auszuwerten.

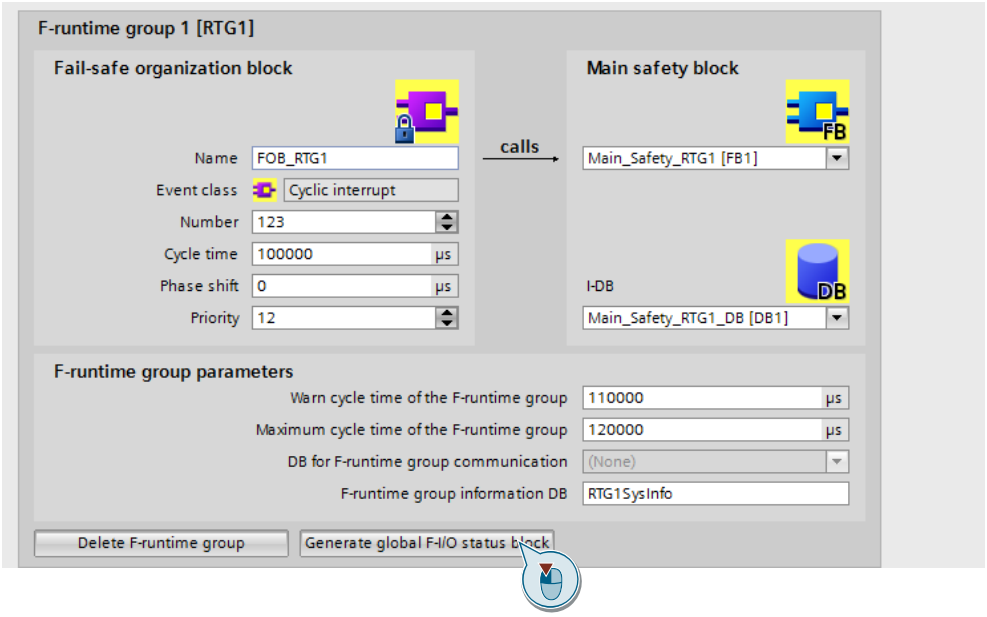
Dieser Baustein wertet aus, ob für mindestens eine F-Peripherie oder mindestens einen Kanal einer F-Peripherie einer F-Ablaufgruppe statt der Prozesswerte Ersatzwerte ausgegeben werden. Das Ergebnis der Auswertung steht am Ausgang "QSTATUS" an. Dabei bleiben F-Peripherien, die Sie mit der Variable DISABLE im F-Peripherie-DB deaktiviert haben, unberücksichtigt.

Abbildung 3-16: Systemgenerierter Baustein zur globalen Auswertung der F-Peripherien



Den Baustein generieren Sie in der Safety Administration in den Einstellungen der jeweiligen F-Ablaufgruppe.

Abbildung 3-17: Baustein zur globalen Auswertung der F-Peripherien generieren



3.11.2 Automatische Wiedereingliederung

Abhängig davon, ob das jeweilige Modul den Standard "RIOforFA" (siehe Kapitel 5) unterstützt, können Sie die automatische Wiedereingliederung auf unterschiedliche Weise realisieren.

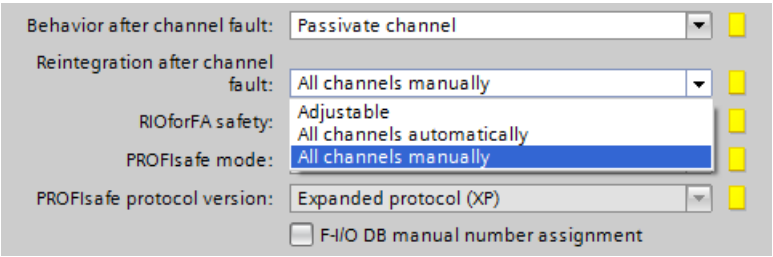
WARNUNG	**Automatische Wiedereingliederung kann eine gefährliche Situation einleiten** Ob eine automatische Wiedereingliederung für betreffenden Prozess sicherheitstechnisch zulässig ist, ist abhängig von der Risikobeurteilung.

Hinweis Die automatische Wiedereingliederung bezieht sich auf F-Peripherie-/Kanalfehler (z. B. Diskrepanzfehler, Kurzschluss). Kommunikationsfehler müssen weiterhin manuell quittiert werden (siehe Kapitel 3.11.3).

Module, die "RIOforFA" unterstützen

Bei Modulen, die "RIOforFA" unterstützen, können Sie eine automatische Wiedereingliederung entweder für das gesamte Modul oder auch nur für einzelne Kanäle parametrieren.

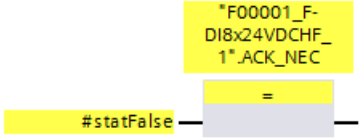
Abbildung 3-18: Automatische Wiedereingeliederung parametrieren



Module, die "RIOforFA" nicht unterstützen

Bei Modulen, die "RIOforFA" nicht unterstützen, programmieren Sie die automatische Wiedereingliederung im Sicherheitsprogramm. Setzen Sie dazu die Variable "ACK_NEC" des jeweiligen F-Peripherie-Datenbausteins auf "false":

Abbildung 3-19: Automatische Wiedereingliederung programmieren

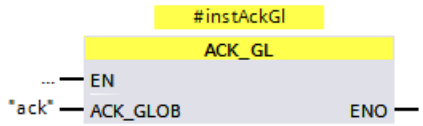


3.11.3 Manuelle Wiedereingliederung

Globale Reintegration aller passivierten F-Module

Um alle passivierten F-Module bzw. -Kanäle einer F-Ablaufgruppe wiedereinzugliedern, verwenden Sie die Anweisung "ACK_GL":

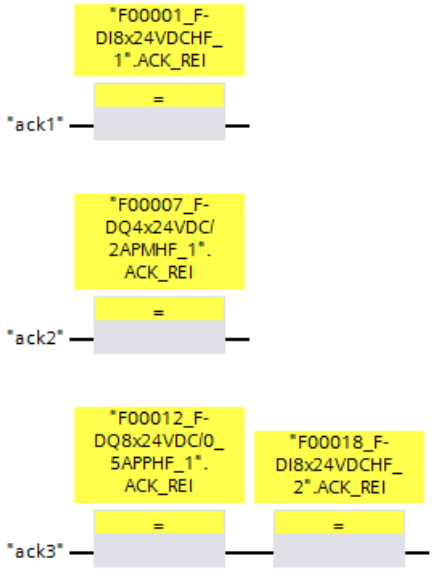
Abbildung 3-20: Anweisung "ACK_GL"



Separate Wiedereingliederung von Modulen (oder einer Gruppe von Modulen)

Bei verteilten Anlagen kann es erforderlich sein, dass nur lokal wiedereingegliedert werden darf (z. B. separate Befehlsgeräte am Schaltschrank). Verschalten Sie dazu die Variablen "ACK_REI" der jeweiligen F-Peripherie-Datenbausteine:

Abbildung 3-21: Separate Wiedereingliederung von Modulen



4 Sicherheitsprogramme optimieren

4.1 Übersetzungsdauer und Laufzeit optimieren

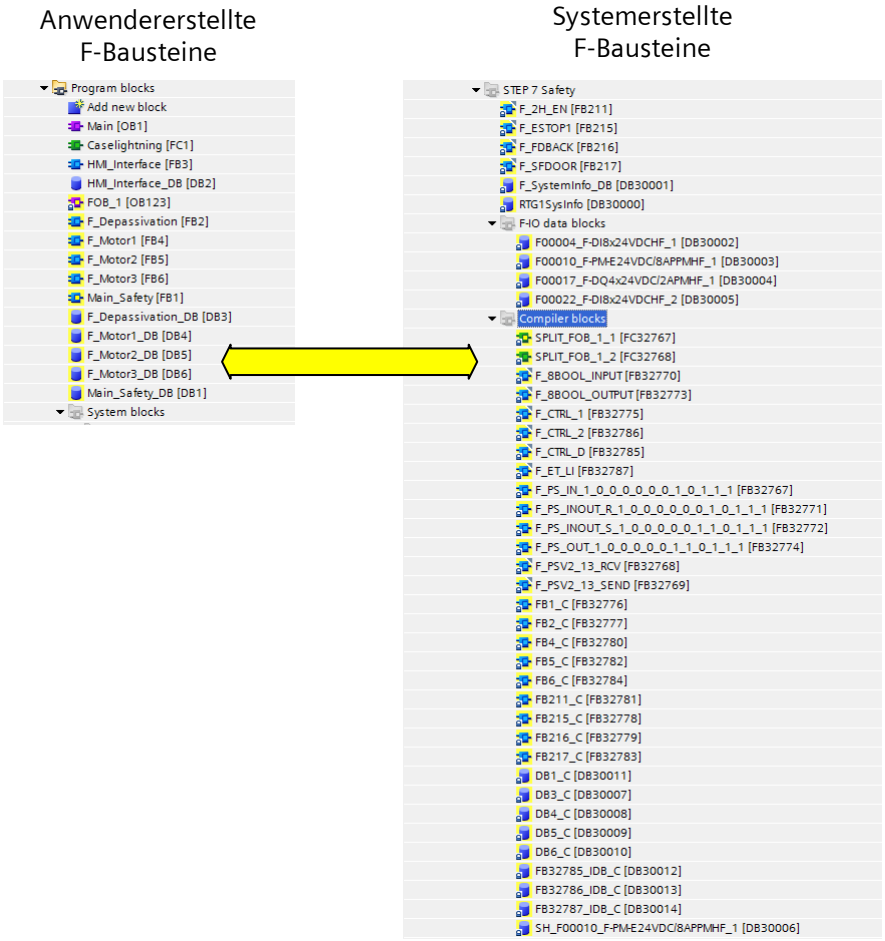
Einleitung

Ein wichtiger Bestandteil eines Sicherheitsprogramms ist die Absicherung der Anwenderprogrammierung durch das Coded Processing (siehe Kapitel 5). Ziel ist es, jegliche Datenverfälschung im Sicherheitsprogramm aufzudecken und damit unsichere Zustände zu verhindern.

Dieses Absicherungsprogramm wird während der Übersetzung erzeugt und verlängert so die Übersetzungsdauer. Auch die Laufzeit der F-CPU wird durch das Absicherungsprogramm verlängert, da die F-CPU dieses zusätzlich bearbeitet und die Ergebnisse mit dem Anwenderprogramm vergleicht.

Das Absicherungsprogramm, das automatisch vom System generiert wird, finden Sie im Systembausteinordner Ihrer F-CPU.

Abbildung 4-1: Absicherungsprogramm



Dabei haben manche Anweisungen, die im Sicherheitsprogramm verwendet werden können, stärkeren Einfluss auf die Performance einer fehlersicheren Steuerung als andere.

In diesem Kapitel werden Ihnen verschiedene Möglichkeiten zur Verkürzung der Übersetzungs- und Programmlaufzeit aufgezeigt.

Hinweis Es ist je nach Anwendung nicht immer möglich, alle Vorschläge zu nutzen. Sie geben aber Aufschluss, warum bestimme Programmiermethoden kürzere Übersetzungs- und Programmlaufzeiten als ein nicht-optimiertes Programm verursachen.

Laufzeit ermitteln

TIA Portal erstellt für jede F-Ablaufgruppe automatisch einen Datenbaustein "RTGxSysInfo", der unter anderem die aktuelle sowie die längste Laufzeit dieser F-Ablaufgruppe enthält.

Diesen systemgenerierten Baustein finden Sie in der Projektnavigation unter ("Program blocks > System blocks > STEP 7 Safety").

Abbildung 4-2: Systemgenerierter DB "RTGxSysInfo"

RTG1SysInfo

	Name	Data type	Start value	Monitor value
1	Input			
2	Output			
3	MODE	Bool	false	FALSE
4	F_SYSINFO	F_SYSINFO		
5	MODE	Bool	false	FALSE
6	TCYC_CURR	DInt	0	100
7	TCYC_LONG	DInt	0	101
8	TRTG_CURR	DInt	0	0
9	TRTG_LONG	DInt	0	2
10	T1RTG_CURR	DInt	0	0
11	T1RTG_LONG	DInt	0	0
12	F_PROG_SIG	DWord	DW#16#103E2...	16#103E_261A
13	F_PROG_DAT	DTL	DTL#2017-9-1...	DTL#2017-09-19-1...
14	F_RTG_SIG	DWord	DW#16#8A587...	16#8A58_7EBD
15	F_RTG_DAT	DTL	DTL#2017-9-1...	DTL#2017-09-19-1...
16	VERS_S7SAF	DWord	DW#16#14000...	16#1400_0100
17	InOut			
18	Static			

4.1.1 Sprünge im Sicherheitsprogramm

In einem Standard-Anwenderprogramm ist ein Sprung von einem Netzwerk in ein anderes (Jump auf Label) oder aus dem Baustein heraus (Return) eine einfache Programmverzweigung, die für jeden Zyklus neu berechnet, aber nicht extra abgesichert wird. Es wird also nicht geprüft, ob z. B. durch einen durch EMV erzeugten Speicherfehler ein Sprung trotz Bedingung "false" springt oder nicht.

In einem fehlersicheren Programm ist das nicht zulässig, da zu jeder Zeit garantiert werden muss, dass sich das Programm im korrekten Ablaufzweig befindet.

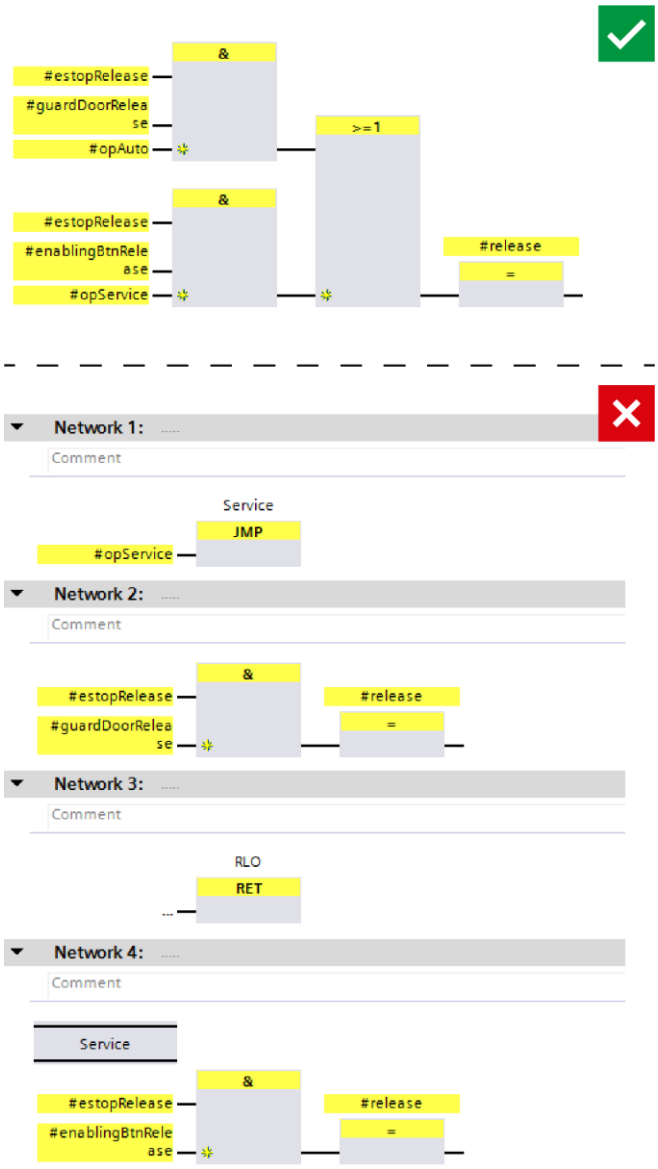
Dafür müssen im Absicherungsprogramm beide Alternativen (Jump auf Label ist "true" oder "false") vollständig berechnet werden.

Je mehr Sprünge Sie in einem Sicherheitsprogramm verwenden, desto stärker wird der Einfluss auf die Leistung der Steuerung.

Empfehlung

- Wenn möglich, vermeiden Sie Sprünge im Sicherheitsprogramm.
- Verwenden Sie Zustandsautomaten statt Sprüngen in FBs mit binärer Logik.

Abbildung 4-3: Sprünge vermeiden



4.1.2 Timer-Bausteine

Timer sind für ein Sicherheitsprogramm ein zentraler Bestandteil, da auch viele der Systemfunktionen wie "ESTOP1" intern diese Timer verwenden. Trotz dessen ist der Aufwand zur Erzeugung eines fehlersicheren Zeitwerts äußerst umfangreich und muss für jeden einzelnen Timer-Baustein erneut generiert werden.

Empfehlung

Reduzieren Sie die Anzahl der Timer-Bausteine auf ein Minimum.

4.1.3 Multiinstanzen

Empfehlung

Verwenden Sie Multiinstanzen für fehlersichere Funktionsbausteine. Das bedeutet, dass die Baustein-internen Variablen in die Bausteinschnittstelle des aufrufenden Bausteins integriert werden.

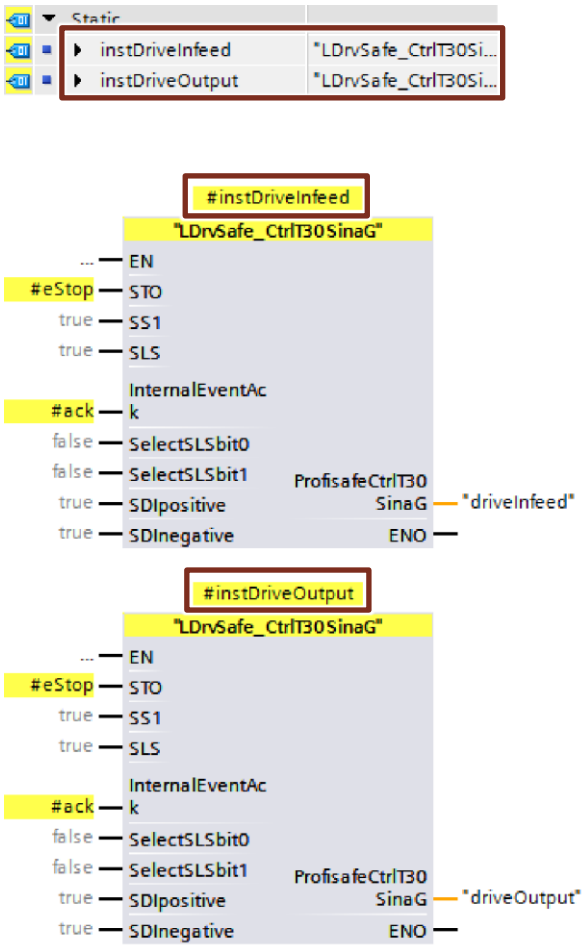
Vorteile

- Standarisierung von Sicherheitsprogrammen:
  Es werden keine Globaldaten für die Bausteinvariablen verwendet. Somit kann der aufrufende Baustein (inklusive der integrierten Bausteine) wiederverwendet werden.
- Optimierung der Compile-Performance:
  - Die Verwendung von Multiinstanzen benötigt weniger Absicherungscode als die Verwendung von Instanz-Datenbausteinen.
  - Durch die Verwendung von Multiinstanzen wird der Umfang an systemseitigen F-Bausteinen reduziert. Die Übersetzung des Sicherheitsprogramms wird dadurch optimiert.

Beispiel

Zwei Antriebe werden mit demselben Funktionsbaustein "LDrvSafe_CtrlT30SinaS" sicher angesteuert. Die Datenablage erfolgt in Multiinstanzen mit eindeutigen Namen.

Abbildung 4-4: Multiinstanzen



Die Bibliothek "LDrvSafe" zur Ansteuerung der Sicherheitsfunktionen von SINAMICS Antrieben finden Sie im Industry Online Support:

https://support.industry.siemens.com/cs/ww/de/view/109485794

4.2 Datenverfälschung vermeiden

Die Absicherungsmechanismen im Rahmen des Coded Processing (siehe Kapitel 5) analysieren den Programmablauf zyklisch auf Datenverfälschungen. Im Falle einer solchen Verfälschung löst ein spezieller Systemfunktionsbaustein einen F-STOPP der CPU aus.

Dieser Mechanismus bezweckt, Einflüsse wie EMV, defekte Bauteile und ähnliches aufzudecken und das System in einen sicheren Zustand zu bringen, bevor die Maschine zu einer Gefahr für Mensch und Umgebung wird.

Neben äußeren Einflüssen kann auch eine falsche Programmierung Datenverfälschung verursachen. Die häufigste Ursache für Datenverfälschung ist, dass das Standard-Anwenderprogramm oder ein externes Gerät (z. B. HMI) Daten beschreibt, während das Sicherheitsprogramm diese liest.

Dies kann in folgenden Situationen auftreten:

- Schreibender Zugriff durch höherpriore Alarme
- Schreibender Zugriff durch HMI/Kommunikation
- Verwendung von Taktmerkern
- Aktualisierung eines Teil-PAE durch höherpriore Alarme

Wie Sie Zugriffe vom Standard-Anwenderprogramm auf das Sicherheitsprogramm korrekt programmieren, finden Sie im Kapitel 3.9.

Checkliste

Mit der folgenden Checkliste können Sie anwendererzeugte STOPP-Ursachen identifizieren und beheben.

Tabelle 4-1: Checkliste

Mögliche Ursachen	Checked
Überlauf Arithmetische Funktionen können Unter- bzw. Überlaufen, was der Anwender im Programm abfangen muss. Eine Bibliothek mit arithmetischen Funktionen für das Sicherheitsprogramm finden Sie in SIOS: https://support.industry.siemens.com/cs/ww/de/view/109482083	
Division durch 0 Kommt es im Sicherheitsprogramm zu einer Division durch 0 geht die F-CPU in STOP. Der Divisor muss vor der Division auf 0 geprüft und die Division bedingt übersprungen werden.	
Zugriff über HMI Über ein HMI werden schreibend Daten (Merker, DBs) verändert, die im Sicherheitsprogramm lesend verwendet werden. Da die Kommunikation voreingestellt eine höhere Priorität als Safety hat, kann dadurch eine Datenverfälschung entstehen. Mögliche Lösungen finden Sie im Kapitel 3.9.	
Standardzugriff auf F-Daten Das Standard-Anwenderprogramm ändert Daten von fehlersicheren Variablen oder Teile von deren Absicherungen. Der schreibende Zugriff auf F-Daten ist ausschließlich im Sicherheitsprogramm zulässig.	
Pointerzugriff auf F-Daten Identisch zum Standardzugriff, kann der Zugriff zur Laufzeit bei ungünstigen Standardwerten zur Bildung eines Pointers auf F-Bereiche (Eingänge, Ausgänge, Datenbausteine usw.) auftreten.	
Verändern von Startwerten in F-Instanzdatenbausteinen Veränderungen von Startwerten dürfen nur in den Interfaces von F-FBs vorgenommen werden.	
Lesen und Schreiben von Standarddaten Ein lesender und schreibender Zugriff auf das gleiche Standarddatum ist aus dem Sicherheitsprogramm nicht zulässig.	

Weitere Informationen

Weitere Informationen und Ursachen für Datenverfälschung finden Sie im Siemens Industry Online Support:

https://support.industry.siemens.com/cs/ww/de/view/19183712

5 Glossar

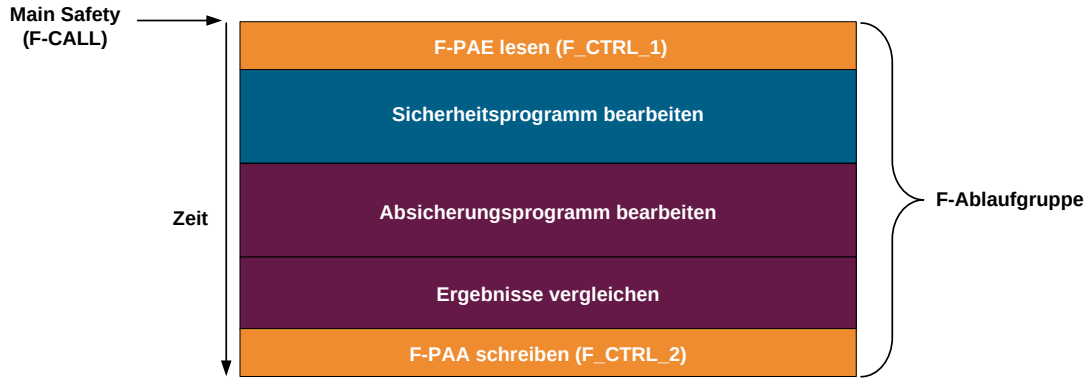
Coded Processing

Zur Erfüllung der normativen Anforderungen bezüglich Redundanz und Diversität nutzen alle SIMATIC F-CPUs das Prinzip des "Coded Processing". Bei diesem Prinzip wird das Sicherheitsprogramm von einem einzelnen Prozessor zweimal bearbeitet.

Dafür erzeugt der Compiler beim Übersetzen ein diversitäres (kodiertes) Sicherheitsprogramm, das als Absicherungsprogramm bezeichnet wird.

Im ersten Programmlauf wird das unveränderte Sicherheitsprogramm des Anwenders bearbeitet. Danach erfolgt die Bearbeitung des Absicherungsprogramms. Anschließend vergleicht die F-CPU die Ergebnisse. Bei korrekter Abarbeitung werden die sicheren Ausgänge geschrieben. Sollte es zu Unterschieden zwischen beiden Programmteilen kommen (z. B. aufgrund von Datenverfälschung), geht die F-CPU in den Stopp- Zustand und erstellt einen Eintrag im Diagnosepuffer.

Abbildung 5-1: Ablauf der Bearbeitung des Sicherheitsprogramms



Datenverfälschung

Datenverfälschung bedeutet, dass Daten des Sicherheitsprogramms durch äußere Einflüsse (z. B. EMV-Einflüsse) oder unzulässige, schreibende Zugriffe verfälscht werden.

F-CPU

Eine F-CPU ist eine Steuerung, die für sicherheitsgerichtete Aufgaben geeignet ist.

PROFIsafe

PROFIsafe ist ein Protokoll für die fehlersichere Kommunikation über PROFINET.

Querschluss

Die Querschlusserkennung ist eine Diagnosefunktion eines Auswertegerätes, wodurch Kurz- bzw. Querschlüsse zwischen zwei Eingangskanälen (Sensorkreisen) erkannt werden.

Ein Querschluss kann beispielsweise durch das Quetschen einer Mantelleitung entstehen. Ohne Querschlusserkennung würde dies zur Folge haben, dass z. B. eine zweikanalige Not-Halt-Schaltung auch bei nur einem fehlerhaften Öffnerkontakt (Zweitfehler) keine Abschaltung auslöst.

RIOforFA

RIOforFA (Remote IO for Factory Automation) ist ein Standard der PROFIBUS & PROFINET International-Organisation und beschreibt unter anderem folgende Funktionen:

- Synchrone Bereitstellung kanalgranularer Diagnose vom Remote IOs für eine hohe Performance
- Kanalgranulare Passivierung und Wiedereingliederung von PROFIsafe Remote IOs

Rückführkreis

Ein Rückführkreis dient der Überwachung angesteuerter Aktoren (z. B. Relais oder Lastschütze) mit zwangsgeführten Kontakten bzw. Spiegelkontakten. Die Ausgänge können nur bei geschlossenem Rückführkreis aktiviert werden. Bei Verwendung eines redundanten Abschaltpfades muss der Rückführkreis beider Aktoren ausgewertet werden. Diese dürfen dafür auch in Reihe geschaltet werden.

Rückstellfunktion/rückstellen

Nach dem Auslösen einer Sicherheitsfunktion muss der Stoppzustand aufrechterhalten bleiben, bis ein sicherer Zustand für den Wiederanlauf gegeben ist.

Als Rückstellfunktion bzw. Rückstellen wird das Wiederherstellen der Sicherheitsfunktion und Aufheben des Stoppbefehls bezeichnet.

Oft wird hier auch vom "Quittieren der Sicherheitsfunktion" gesprochen.

Sicherheitsprogramm

Der Teil des Anwenderprogramms, in dem sicherheitsgerichtete Aufgaben bearbeitet werden.

STEP 7 Safety Basic/Advanced

STEP 7 Safety Basic und Advanced sind Optionspakete für STEP 7, mit denen F-CPUs projektiert und Sicherheitsprogramm erstellt werden können.

- Mit STEP 7 Safety Basic können Sie die fehlersicheren Steuerungen SIMATIC S7-1200 projektieren.
- Mit STEP 7 Safety Advanced können Sie alle fehlersicheren SIMATIC-Steuerungen projektieren.

6 Anhang

6.1 Service und Support

Industry Online Support

Sie haben Fragen oder brauchen Unterstützung?

Über den Industry Online Support greifen Sie rund um die Uhr auf das gesamte Service und Support Know-how sowie auf unsere Dienstleistungen zu.

Der Industry Online Support ist die zentrale Adresse für Informationen zu unseren Produkten, Lösungen und Services.

Produktinformationen, Handbücher, Downloads, FAQs und Anwendungsbeispiele – alle Informationen sind mit wenigen Mausklicks erreichbar:
https://support.industry.siemens.com

Technical Support

Der Technical Support von Siemens Industry unterstützt Sie schnell und kompetent bei allen technischen Anfragen mit einer Vielzahl maßgeschneiderter Angebote – von der Basisunterstützung bis hin zu individuellen Supportverträgen.

Anfragen an den Technical Support stellen Sie per Web-Formular:
www.siemens.de/industry/supportrequest

Serviceangebot

Unser Serviceangebot umfasst, unter anderem, folgende Services:

- Produkttrainings
- Plant Data Services
- Ersatzteilservices
- Reparaturservices
- Vor-Ort und Instandhaltungsservices
- Retrofit- und Modernisierungsservices
- Serviceprogramme und Verträge

Ausführliche Informationen zu unserem Serviceangebot finden Sie im Servicekatalog:
https://support.industry.siemens.com/cs/sc

Industry Online Support App

Mit der App "Siemens Industry Online Support" erhalten Sie auch unterwegs die optimale Unterstützung. Die App ist für Apple iOS, Android und Windows Phone verfügbar:
https://support.industry.siemens.com/cs/ww/de/sc/2067

6.2 Links und Literatur

Tabelle 6-1: Links und Literatur

Nr.	Thema
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link auf die Beitragsseite des Anwendungsbeispiels https://support.industry.siemens.com/cs/ww/de/view/109750255
\3\	Programmierleitfaden für SIMATIC S7-1200/1500 https://support.industry.siemens.com/cs/ww/de/view/90885040
\4\	Programmierstyleguide für SIMATIC S7-1200/1500 https://support.industry.siemens.com/cs/ww/de/view/109478084
\5\	SIMATIC Industrie Software SIMATIC Safety - Projektieren und Programmieren https://support.industry.siemens.com/cs/ww/de/view/54110126
\6\	Themenseite "Safety Integrated – Sicherheitstechnik in der Fertigungsautomatisierung" https://support.industry.siemens.com/cs/ww/de/view/109747812

6.3 Änderungsdokumentation

Tabelle 6-2: Änderungsdokumentation

Version	Datum	Änderung
V1.0	10/2017	Erste Ausgabe