



Programmierwerkzeug

SILworX[®]

Release-Notes V12.28



Alle in diesem Handbuch genannten HIMA Produkte sind mit dem Warenzeichen geschützt. Dies gilt ebenfalls, soweit nicht anders vermerkt, für weitere genannte Hersteller und deren Produkte.

HIQuad®, HIQuad®X, HIMax®, HIMatrix®, SILworX®, XMR®, HICore® und FlexSILon® sind eingetragene Warenzeichen der HIMA Paul Hildebrandt GmbH.

Alle technischen Angaben und Hinweise in diesem Handbuch wurden mit größter Sorgfalt erarbeitet und unter Einschaltung wirksamer Kontrollmaßnahmen zusammengestellt. Bei Fragen bitte direkt an HIMA wenden. Für Anregungen, z. B. welche Informationen noch in das Handbuch aufgenommen werden sollen, ist HIMA dankbar.

Technische Änderungen vorbehalten. Ferner behält sich HIMA vor, Aktualisierungen des schriftlichen Materials ohne vorherige Ankündigungen vorzunehmen.

Alle aktuellen Handbücher können über die E-Mail-Adresse documentation@hima.com angefragt werden.

© Copyright 2020, HIMA Paul Hildebrandt GmbH

Alle Rechte vorbehalten.

Kontakt

HIMA Paul Hildebrandt GmbH

Postfach 1261

68777 Brühl

Tel.: +49 6202 709-0

Fax: +49 6202 709-107

E-Mail: info@hima.com

Revisions- index	Änderungen	Art der Änderung	
		technisch	redaktionell
1.02	Erstausgabe des Dokuments		

Inhaltsverzeichnis

1	SILworX V12.28	5
1.1	Kompatibilitäten	5
1.1.1	PADT-Betriebssystem	5
1.1.2	Unterstützte HIMA Systemfamilien	5
1.1.3	Nicht unterstützte HIMA Systemfamilien	5
1.1.4	Kompatibilität zu älteren Projekten	5
1.1.5	Kompatibilität zur PADT-Hardware	6
1.1.6	Windows-Rechte für die Installation und den Betrieb von SILworX	6
2	Neue Funktionen	7
3	Verbesserungen der Version	9
4	Einschränkungen	10
4.1	FBS-Editor	10
4.2	Structured Text	12
4.3	Hardware	12
4.4	Anwenderprogramm	13
4.5	Versionsvergleich	16
4.6	Codegenerierung	17
4.7	Reload	18
4.8	Protokolle	19
4.9	Projekt	20
4.10	Dokumentation	21
4.11	Benutzerverwaltung	21
4.12	Lizenzen	21
4.13	SILworX API	22
5	Besonderheiten	23
6	Upgrade von einer Vorversion	26
6.1	Referenzen	26

1 SILworX V12.28

Dieses Kapitel beschreibt die Verbesserungen, neuen Funktionen und Einschränkungen von SILworX V12.28 gegenüber Vorversionen.

1.1 Kompatibilitäten

Das Kapitel beschreibt die Kompatibilität der einzelnen Software-Versionen zu den Hardware-Versionen und zu Windows-Betriebssystemen.

1.1.1 PADT-Betriebssystem

SILworX ist ab der Version V11 eine 64-Bit-Anwendung und wird nur von Windows 10 (64-Bit) unterstützt.

1.1.2 Unterstützte HIMA Systemfamilien

SILworX V12.28 ist für folgende HIMA Systemfamilien einsetzbar:

- HIMax.
- HIMatrix F-Systeme.
- HIQuad X.

Generell sind die TÜV-Versionslisten der jeweiligen Systemfamilien zu beachten.

1.1.3 Nicht unterstützte HIMA Systemfamilien

Ab SILworX V9.36 werden nachfolgende Steuerungen nicht mehr unterstützt:

- HIMatrix F10 PCI 03.
- HIMatrix F20 01.
- HIMatrix F30 01.
- HIMatrix F31 02.
- HIMatrix F31 03.
- HIMatrix F35 01.
- HIMatrix F60 01.

Diese Steuerungen dürfen zusammen mit SILworX Versionen ab V9.36 nicht mehr benutzt werden (Haftungsausschluss). Dies betrifft vor allem die Codegenerierung, den Download und die Online-Dienste.

D. h. Projekte, welche die aufgeführten Steuerungen enthalten, dürfen in SILworX geöffnet werden z. B. für Updates/Upgrades-Aktionen (Ersatz durch Nachfolgeprodukte). Eine Verbindung mit den aufgeführten Steuerungen aufzubauen, ist mit den SILworX Versionen ab V9.36 nicht mehr zulässig.

1.1.4 Kompatibilität zu älteren Projekten

SILworX V12.28 kann Projekte konvertieren und bearbeiten, die mit einer älteren Version erstellt wurden.

HIMA achtet bei der Weiterentwicklung von SILworX darauf, dass Codegenerierungen mit einer neuen SILworX Version nur in Ausnahmefällen zu geänderten Konfigurations-CRCs führen. Diese Fälle werden in den Release-Notes jeder SILworX Version dokumentiert.

1.1.5 Kompatibilität zur PADT-Hardware

HIMA empfiehlt, aktuelle Rechner-Hardware einzusetzen. Speziell bei sehr großen Projekten können ältere Rechner lange Verarbeitungszeiten aufweisen und dadurch ungeeignet sein. Die Rechner-Hardware sollte möglichst dem Stand der Technik entsprechen. Bessere Hardware-Eigenschaften wie Rechenleistung (Anzahl CPU-Kerne) und Speicherausbau (RAM) führen zu verbesserter Performance.

1.1.6 Windows-Rechte für die Installation und den Betrieb von SILworX

Für die Installation und den Betrieb von SILworX unter Windows ist folgendes zu beachten:

- Für die Installation sind «Administrator» -Rechte nötig.
- Für den Betrieb oder Start sind «Benutzer» -Rechte ausreichend.

2 Neue Funktionen

- OPC UA wird als neue Funktionalität in dieser Version eingeführt.
- Parameter «Minimale Konfigurationsversion» wurde um SILworX V12 erweitert, um z. B. OPC UA-Server zu unterstützen. Standardeinstellung neu: SILworX V12.
- Das SILworX Application Programming Interface (SILworX API) bietet zahlreiche Funktionen zum Steuern von SILworX. Mittels Bibliotheken (z. B. für Python und C#) können Anwender API-Aufrufe in ihre Automatisierungsumgebung einbinden.
Unter anderem stehen folgende Funktionen zur Verfügung:
 - Projekt erzeugen.
 - Projekt öffnen, auch mit Benutzerverwaltung.
 - Projekt schließen.
 - SILworX schließen.
 - SILworX Status auslesen (z. B. Version und Lizenzen).
 - Knoten im «Projekt» Strukturbaum erstellen.
 - Strukturbaum auslesen.
 - Validierung im Strukturbaum.
 - Archivieren (Projekt, Strukturbaum-Knoten).
 - Wiederherstellen (Projekt, Strukturbaum-Knoten).
 - Code generieren (für Download/für Reload).
 - Verbinden und Trennen von Ressourcen.
 - Modul-Login.
 - Starten und Stoppen von Ressourcen.
 - Starten und Stoppen von Systemen und deren Module.
 - Downloads von Konfigurationen in Ressourcen.
 - Online-Werte lesen (System und Modulebene / einzeln oder gruppiert).
 - Diagnoseeinträge von Modulen abfragen.
 - Downloads von Betriebssystemen.
 - Starten und Stoppen des globalen Forcens.
 - Globale Force Variablen lesen und schreiben.

Allgemeines zur API:

- Die Benutzung der SILworX API erfordert eine Lizenz.
- Die SILworX API muss explizit in der «settings.ini» aktiviert werden.
- Zugriffe auf die SILworX API sind ausschließlich über SSL (TLS 1.2) möglich.
Hierzu ist die Installation von OpenSSL und ein gültiges Zertifikat nötig.
- Zugriffe über die SILworX API auf Projekte benötigen die gleichen Benutzerrechte wie beim manuellen Arbeiten.
- Konfigurierbare Timeouts bei der Benutzung der SILworX API-Zugriffe sorgen dafür, dass Projekte automatisch geschlossen werden, wenn bis zum Timeout keine weiteren API-Anfragen gesendet werden.
- API-Aktivität wird in der SILworX Statusleiste angezeigt.
- Alle Aktionen werden im SILworX Logbuch protokolliert. Dies gilt sowohl für das manuelle Arbeiten, als auch für API-Zugriffe.

Wichtig:

Der Anwender muss für seine SILworX API-Anwendung eine Tool-Klassifikation durchführen und entsprechend qualifizieren.

Im Unterordner ...\\c3\\openapi des SILworX Installationsverzeichnis befindet sich die API-Dokumentation in HTML-Format und ein C# Anwendungsbeispiel.

- Keine weitere Unterstützung folgender Ressource-Typen:
 - HIMatrix F10 PCI 03
 - HIMatrix F20 01
 - HIMatrix F30 01
 - HIMatrix F31 02
 - HIMatrix F31 03
 - HIMatrix F35 01
 - HIMatrix F60 01

Projekte, die diese Ressource-Typen enthalten, können weiterhin geöffnet werden. Für diese Ressource-Typen kann keine neue Hardware mehr angelegt werden. Für diese Ressource-Typen kann kein Code mehr generiert werden. Online-Verbindungen zu Steuerungen mit diesen Ressource-Typen sind nicht mehr möglich. Beim Öffnen von Projekten, die diese Ressource-Typen enthalten, wird für jede betroffene Ressource eine Warnung ausgegeben.

- Im FBS-Editor stehen die Funktionen «Rückgängig» und «Wiederherstellen» zur Verfügung.
- Wenn das Aktualisieren aller Instanzen oder Wertfelder mit einem Fehler abbricht, werden alle zu aktualisierenden Elemente in den Zustand vor der Aktion zurückversetzt. In früheren SILworX Versionen wurden Elemente, die sich fehlerfrei aktualisieren ließen, trotz Abbruch nicht zurückversetzt.

3 Verbesserungen der Version

- Die Codegenerierung für X-OTS zu X-OPC-Set mit zwei Servern scheitert, wenn Systemvariablen der Verbindung zum zweiten Server benutzt werden.
Dieses Problem wurde gelöst.
- Wenn die COM-Konfiguration einer HIMatrix Steuerung geändert wird und danach eine Reload-Codegenerierung durchgeführt wird, gibt SILworX eine Reload-Warnung aus.
- Inhalt des OLT-Felds hielt sich nicht an seine Feldgröße. Aus den OLT-Feldern optisch herausragende Texte werden gekürzt und dem Anwender mit «...» am Ende des Feldes angezeigt. Wenn der Anwender den vollständigen Text sehen möchte, kann er das OLT-Feld größer ziehen. Alternativ kann der Text auch per Tooltip angezeigt werden.
- SILworX ermöglicht die Ablage von GSDML-Bibliotheken zusätzlich unterhalb von Bibliotheksknoten. Bisher konnten GSDML-Dateien nur in genau einer fixen GSDML-Bibliothek unterhalb eines PROFINET-Controllers abgelegt werden. Jetzt ist die Controller- bzw. Ressourcen-übergreifende gemeinsame Verwendung von GSDML-Dateien möglich, dadurch können Redundanzen vermieden werden.
- Offline-Simulation: Der Start eines Anwenderprogramms oder eines Zyklusschritts ist wieder möglich, wenn folgende Randbedingungen gelten:
 - a) Das Anwenderprogramm enthält die Verwendung der SQRT-Funktion mit REAL-Typisierung ausschließlich in benutzerdefinierten Funktions-POEs.
 - b) Bei der Ausführung der SQRT-Funktion liegt am Eingang «IN» ein negativer REAL-Wert an, z. B. -0.5.
D. h. entsprechende SQRT-Verwendungen liefern als Ergebnis am «OUT» den REAL-Ausnahmewert NaN bzw. setzen den Ausgang «ENO» auf den Wert FALSE.
- Englische und deutsche Texte, die Einheiten beinhalten, werden korrigiert, um nach DIN 5008 konform zu sein:
 - 1mA -> 1 mA
 - 1°C -> 1 °C
 - 1mV -> 1 mV
 - 24V -> 24 V

Diese Änderung führt dazu, dass sich die Spalten von neu exportierten CSV-Dateien, von denen in bereits bestehenden Dateien unterscheiden können. Bestehende CSV-Dateien sind möglicherweise erst nach dem Ersetzen von Texten mit «Suchen/Ersetzen» wieder importierfähig.

- Der neue Parameter «LB-Austastung (Anzahl LS/LB-Intervalle)» bei den HIMax Modulen X-DO 24 01 und X-DO 24 02 definiert die Anzahl von Testintervallen (Parameter «LS/LB-Intervall [µs]»). Weitere Einzelheiten siehe HIMax Release-Notes HI 801 547 D.

4 Einschränkungen

Die Verwendung von SILworX unterliegt Einschränkungen. Wenn Sie die nachfolgenden Hinweise beachten, haben die Einschränkungen keine Auswirkungen auf die Sicherheit und die Verfügbarkeit des Codes, der für eine Steuerung generiert wird.

4.1 FBS-Editor

- **Auswirkung:**
Im FBS-Editor werden Kommentartexte in zugeordneten Kommentarfeldern nicht mehr angezeigt, nachdem neue Blätter eingefügt wurden.

Bedingung:

Folgende Bedingungen müssen dabei zutreffen:

- Ein Blatt mit einem Kommentar im zugeordneten Kommentarfeld befindet sich neben einem zweiten Blatt, welches mindestens ein Logikelement enthält und
- auf dem zweiten Blatt wird die Aktion «Leere Blätter einfügen -> Spalte oder Zeile einfügen» so ausgeführt, dass neue Blätter zwischen dem Blatt mit dem Kommentarfeld und dem zweiten Blatt eingefügt werden.

Der Text im Kommentarfeld wird temporär nicht mehr angezeigt.

Abhilfe:

Der Text erscheint wieder, wenn das zugeordnete Kommentarfeld bewegt oder der Editor geschlossen und neu geöffnet wird.

- **Auswirkung:**
Im FBS-Editor werden Blattinformationen nach einem Projektimport aus ELOP II nicht korrekt platziert.

Bedingung:

Folgende Bedingungen müssen dabei zusammentreffen:

Ein zugeordnetes Kommentar- oder OLT-Feld liegt auf einem leeren Blatt ohne weitere Logikelemente, während sich das damit verbundene Element auf einem anderen Blatt befindet.

Abhilfe:

Zugeordnete Kommentar- oder OLT-Felder sollten sich auf dem gleichen Blatt wie die damit verbundenen Elemente befinden.

- **Auswirkung:**
Konflikt-Icon bei Variablen bleibt auch nach Behebung des Konflikts sichtbar.

Bedingung:

In folgenden Fällen bleibt das Konflikt-Icon auch nach Ablehnung einer ungültigen Aktion und Anzeige des korrekten Werts sichtbar:

- Eingabe eines ungültigen Namens im Feld einer lokalen Variable.
- Vergabe einer bereits existierenden Reihenfolge-Nummer für eine Interface-Variable.

Abhilfe:

Verifizieren oder Aktualisieren.

- **Auswirkung:**
Anzeige von globalen Variablen, die als VAR_EXTERNAL verwendet werden.
Werden globale Variable mit Datentyp «Struct» oder «Array» als VAR_EXTERNAL verwendet, zeigt der FBS-Editor für die Unterelemente die Einträge in den Spalten «Initialwert», «Beschreibung», «Zusatzkommentar» und «Technische Einheit» nicht an.

Bedingung:

Globale Variable vom Datentyp «Struct» erstellen und in einem Struct-Element eine Beschreibung eintragen. Diese globale Variable in einem Programm als Wertfeld verwenden. Anschließend die erzeugte VAR_EXTERNAL im Register «Lokale Variablen» anschauen.

Abhilfe:

Attribut-Eigenschaften der VAR_EXTERNAL in der entsprechenden globalen Variable anschauen.

- **Auswirkung:**

Leere Blätter im Zeichenfeld des FBS-Editors sind nicht immer löschar.

Bedingung:

Der Menüpunkt «Leere Blätter löschen» im Kontextmenü ist nicht aktiv, wenn folgende Bedingungen zusammentreffen:

- Eine Linie verläuft über zwei oder mehrere Nachbarblätter der leeren Blätter.
- Die Linie verläuft nicht durch das leere Blatt.

Das leere Blatt kann somit nicht gelöscht werden.

Abhilfe:

Keine.

- **Auswirkung:**

Ausführen von «Rückgängig» oder «Wiederherstellen» kann in seltenen Fällen dazu führen, dass eine POE-Instanz in einem FBS-Baustein ungültig wird.

Bedingung:

- Im FBS-Editor aus der Objektauswahl mit Drag&Drop einen Baustein ins Zeichenfeld kopieren. Dies erzeugt eine POE-Instanz, die auf den Baustein referenziert.
- An den Eigenschaften der POE-Instanz eine Änderung durchführen (z.B. Retain setzen) und den Inhalt des FBS-Editors speichern. Das Sternchen (*) in der Kopfzeile des FBS-Editors verschwindet.

Hinweis: Der Fehler tritt nur auf, nachdem der Inhalt des FBS-Editors gespeichert wurde. Wird der Inhalt nicht gespeichert und das Sternchen in der Kopfzeile des FBS-Editors weiterhin angezeigt, bleibt die Referenz der POE-Instanz gültig.

- In der Objektauswahl den verwendeten (referenzierten) Baustein umbenennen. Der neue Bausteinname wird sofort wirksam.
- «Rückgängig» ausführen. Dadurch wird die Referenz der POE-Instanz wieder auf den vorherigen Bausteinnamen gesetzt. Das Umbenennen des Bausteins wird jedoch nicht rückgängig gemacht, so dass die POE-Instanz ungültig wird.

Abhilfe:

Den Namen des referenzierten Bausteins manuell auf den vorherigen Namen ändern. Dadurch kann die Referenz wieder hergestellt werden.

- **Auswirkung:**

Für «Rückgängig» von Instanznamen und deren Kommentare sind 2 Rückgängig-Wiederherstellen-Schritte notwendig.

Bedingung:

Im Register «Instanzen» die Instanznamen und die Kommentare so ändern, dass sie teilweise identische Textabschnitte enthalten. Mit «Suchen und Ersetzen» die identischen Textabschnitte durch anderen Text ersetzen. Um die Änderungen in den Instanznamen und den Kommentaren zurückzunehmen muss «Rückgängig» zweimal ausgeführt werden.

Abhilfe:
«Rückgängig» zweimal ausführen.

4.2 Structured Text

- **Auswirkung:**
Im ST-Editor sind 2700 aufeinanderfolgende Kommentarzeilen nicht möglich.
- Bedingung:**
Beim Auskommentieren mit 2700 aufeinanderfolgenden Zeilen im ST-Editor terminiert SILworX.
- Abhilfe:**
Längere Kommentare unterteilen, z. B. jeweils 1000 Zeilen zu einem Kommentar zusammenfassen.

4.3 Hardware

- **Auswirkung:**
Im Hardware-Editor ist die Schaltfläche «Detailansicht» nicht aktiv.
- Bedingung:**
Hardware-Online-Ansicht starten und die Detailansicht eines Moduls öffnen: Die Schaltfläche «DetailView» ist nicht aktiv.
- Abhilfe:**
Nicht notwendig, da eine Schaltfläche «Schließen» in der Detailansicht existiert.
- **Auswirkung:**
Ab SILworX V6 werden bei der Codegenerierung die Lizenzen nach Namen und nicht mehr nach der Reihenfolge des Eintragens sortiert. Dadurch kann es bei der Projektkonvertierung aus früheren Versionen zu CRC-Änderungen kommen.
- Bedingung:**
Lizenznamen in nicht alphabetischer Reihenfolge in V5 eintragen und Code generieren. Danach in V6 Code generieren.
- Abhilfe:**
Geeignete Lizenznamen vergeben, HIMA Support hinzuziehen.
- **Auswirkung:**
Die globale Fehlerstatistik wird nicht in den Remote I/Os angezeigt, sondern nur in der übergeordneten Steuerung. Die Systemvariablen für das Forcen sowie CPU-Autostart-Freigabe, CPU-Start-Freigabe, CPU-Haupt-Freigabe, ReadOnlyInRun und Start-Cycle werden von den Remote I/Os nicht bedient. Die Systemvariablen einer Remote I/O werden nicht automatisch von der übergeordneten Steuerung ausgewertet.
- Bedingung:**
Systemvariablen globalen Variablen zuweisen und entsprechende Fehler simulieren.
- Abhilfe:**
Die Fehlerstatistik der übergeordneten Steuerung verwenden. Die Systemvariablen der Remote I/O-Fehlerstatistik mit globalen Variablen belegen und in der Logik auswerten.

4.4 Anwenderprogramm

- **Auswirkung:**
Folgendes Verhalten der Funktion EXPT weicht auf den PES von der IEEE-754-Norm ab.
1.0 ** NaN := 1.0 erwartet: NaN
EXPT.ENO := TRUE erwartet: FALSE

NaN ** 0.0 := 1.0 erwartet: NaN
EXPT.ENO := TRUE erwartet: FALSE

Mit OTS und Offline-Simulation funktioniert EXPT gemäß IEEE-754.

Bedingung:
Siehe oben.

Abhilfe:
Bei ENO-Bedarf, den Wert NaN an beiden Eingängen programmtechnisch abfangen / vermeiden.

- **Auswirkung:**
Die Funktion DIV_TIME aus der Standardbibliothek setzt fälschlich den Fehlerausgang ENO auf FALSE und meldet somit einen Fehler unter folgenden Bedingungen:
 - Der Eingang IN2 (Divisor) ist vom Typ REAL.
 - Der Wert von IN2 ist +/-INF.

Bedingung:
DIV_TIME mit EN/ENO verwenden und als Divisor +/- INF anlegen. (INF ist z. B. das Ergebnis aus 1.0 / 0.0).

Abhilfe:
ENO in diesem Fall ignorieren.

- **Auswirkung:**
Die Funktion EXPT liefert bei Offline-Simulation und OTS für die Basis IN1 = 1.0 und Exponent IN2 = -INF den Wert NaN statt 1.0.

Bedingung:
EXPT mit IN1 = 1.0 und IN2 = -INF (oder andere große negative Zahl aufrufen) und Ergebnis in der Offline-Simulation oder OTS ansehen.

Abhilfe:
Wenn der Sonderfall für die Anwendung relevant ist, muss dies im Anwenderprogramm abgehandelt werden.

- **Auswirkung:**
Bei der Abarbeitung einer POE werden zuerst die Schrittketten bearbeitet, danach die AS-Aktionen und dann die FBS-Logik. Das führt dazu, dass die Eingangswerte von AS-Transitionen und AS-Aktionen, die in der FBS-Logik beschrieben werden, immer aus dem vorherigen Zyklus stammen. Bei der konkreten Auswertung der Eingangswerte bestehen allerdings kleine Unterschiede:
Der Eingangswert einer AS-Transition wird während der FBS-Bearbeitung in den Speicher der AS-Transition geschrieben und dort gemerkt und erst im nächsten Zyklus in der Schrittkette bearbeitet. Dies führt dazu, dass nach einem Kaltstart Schrittketten generell erst ab dem zweiten Zyklus weitergeschaltet werden.
Der Eingangswert einer AS-Aktion wird bei der Abarbeitung der AS-Aktion aus der Quelle gelesen. Falls dies eine Funktion ist, wird der Initialwert gelesen, da Funktionen zu Beginn der POE-Ausführung initialisiert werden und erst nach den AS-Aktionen ausgeführt werden.

Bedingung:
Siehe oben.

Abhilfe:

AS-Transition:

Bei der Programmierung der Schrittkette muss berücksichtigt werden, dass frühestens im zweiten Zyklus eine AS-Transition durchgeführt wird.

AS-Aktion:

Um ein Funktionsergebnis als Eingangswert einer AS-Aktion zu verwenden, muss zwischen Ausgang der Funktion und Eingang der AS-Aktion eine Variable geschaltet werden.

- **Auswirkung:**

Wenn die Ketten einer Auswahlverzweigung mit AS-Schritten enden, führt dies zu einem Deadlock. Wenn die Ketten einer Simultanverzweigung mit Transitionen enden, führt dies zu mehreren aktiven Schritten außerhalb der Simultanverzweigung.

Bedingung:
Siehe oben.

Abhilfe:

Der Anwender muss durch geeignete Maßnahmen dafür sorgen, dass derart fehlerhafte Schrittketten nicht verwendet werden.

- **Auswirkung:**

Beim Zugriff auf ein Array-Element mit einem Index außerhalb des Wertebereiches des Arrays wird nach einem definierten und performanten Verfahren auf ein Element innerhalb des Arrays zugegriffen, um willkürliche Zugriffe auf Speicherbereiche zu vermeiden.

Bedingung:
Siehe oben.

Abhilfe:

Der Anwender muss durch geeignete Programmierung dafür sorgen, dass Array-Zugriffe nur mit Indizes innerhalb des Bereichs des Arrays erfolgen.

- **Auswirkung:**

Verschiedene Elemente einer Strukturvariable sind nicht von unterschiedlichen Quellen beschreibbar.

Es ist nicht möglich, dass sowohl das Anwenderprogramm, als auch die Hardware oder die Kommunikation unterschiedliche Elemente derselben Strukturvariable beschreiben.

Bedingung:
Siehe oben.

Abhilfe:

Unterschiedliche Struktur-Variablen für Elemente, in die das Anwenderprogramm schreibt, und für Elemente, in die Hardware oder in die Kommunikation schreiben.

- **Auswirkung:**

Elemente von Variablen eines benutzerdefinierten Datentyps können nicht als Indizes für Arrays verwendet werden.

Bedingung:
Siehe oben.

Abhilfe:

Den Wert der gewünschten Variable des benutzerdefinierten Datentyps in eine Standardvariable kopieren und die Standardvariable als Index verwenden.

- **Auswirkung:**

Umwandlung von Gleitkommazahlen geändert.

Bei der Umwandlung von Gleitkommazahlen aus der Dezimalform in die dem Datentyp (REAL, LREAL) entsprechende Binärform kommt SILworX V11 zu einem anderen Ergebnis als SILworX V10. Grund für dieses Verhalten ist ein modernerer Umwandlungsalgorithmus, der schneller und exakter arbeiten soll.

Die neuen Ergebnisse fließen in jede mit SILworX ab V11 generierte Konfiguration ein (in Programm, ke.config usw.) und werden mit dieser auch wirksam.

Der Versionsvergleich zwischen einer bis V10 und einer ab V11 generierten Konfiguration zeigt für jede davon betroffene Datei und POE eine CRC-Änderung an.

Für ein betroffenes Wertfeld meldet die Detailansicht der POE eine Änderung der damit verbundenen Instanz oder Zuweisung.

Für eine betroffene lokale Variable meldet die Detailansicht der POE eine Änderung des Initialwert-CRCs der POE.

Für eine betroffene globale Variable erscheint keine Meldung ihres geänderten Initialwerts (in der Detailansicht der ke.config).

Der Dialog «Force-Daten bearbeiten» zeigt für einen Force-Wert sofort das Ergebnis der Umwandlung aus der Dezimalform in Binärform und zurück in Dezimalform an, so dass der Anwender das Ergebnis noch vor dem Absenden prüfen kann.

Bisher wurde nur ein konkreter Fall beobachtet:

-SILworX V10 wandelt das Literal 2.2250738585072012e-308 in den LREAL-Wert mit dem Bitmuster 16#000F_FFFF_FFFF_FFFF um, dessen Dezimaldarstellung in SILworX ist 2.2250738585072009e-308.

-SILworX V11 wandelt das Literal 2.2250738585072012e-308 in den LREAL-Wert mit dem Bitmuster 16#0010_0000_0000_0000 um, dessen Dezimaldarstellung in SILworX ist 2.2250738585072014e-308.

Die beiden Ergebnisse unterscheiden sich nur um eine Einheit in der letzten Stelle (unit in the last place, ulp). Der exakte Literalwert ist ca. 0.37 ulp vom neuen, ca. 0.63 ulp vom alten Ergebnis entfernt. Die Norm IEC 559 : 1989 erlaubt bei der dezimalen Umwandlung in bestimmten Wertebereichen, in die auch dieses Literal fällt, dass das Ergebnis geringfügig von der optimalen Näherung abweicht, so dass beide Ergebnisse zulässig erscheinen ($0.63 \text{ ulp} < (0.37 \text{ ulp} + 0.47 \text{ ulp})$).

Bedingung:

Beispiel für den beobachteten Fall: Das Literal 2.2250738585072012e-308 in ein Wertfeld eingeben oder als Initialwert einer lokalen oder einer globalen Variable mit dem Datentyp LREAL eingeben. Beim Wertfeld den Ausgang mit einem anderen Wertfeld oder einer POE-Instanz passend verbinden.

Für dieses Beispiel in SILworX V10 und V11 jeweils Code generieren. In SILworX V11 einen Versionsvergleich zwischen der V10- und der V11-Konfiguration durchführen.

Wahlweise die jeweilige Konfiguration ins PES laden oder in SILworX V10/V11 die Offline-Simulation starten und die wirksamen Werte im Force-Editor betrachten.

Oder online dasselbe Literal als Force-Wert einer Variable eingeben und das danach angezeigte Ergebnis betrachten.

Abhilfe:

Wenn es auf jedes Bit ankommt, ein REAL- bzw. LREAL-Literal so eingeben, dass es der reellen Zahl möglichst nahe kommt, die dem geforderten Bitmuster zugeordnet¹ ist. Für eine eindeutige Umwandlung sind bei REAL bis zu 9, bei LREAL bis zu 17 signifikante Dezimalstellen erforderlich (Gesamtanzahl vor + nach dem Komma, ohne Betrachtung des

Exponenten).

¹ Dem Bitmuster einer binären Gleitkommazahl ist mathematisch eine ganz bestimmte reelle Zahl zugeordnet. Gleichzeitig ist die Gleitkommazahl Stellvertreter für alle reellen Zahlen in einem Intervall, das die ihr zugeordnete Zahl umgibt (in vielen, aber nicht in allen Fällen symmetrisch).

4.5 Versionsvergleich

- **Auswirkung:**

Die Detailansicht einer POE zeigt beim Versionsvergleich fälschlicherweise die Änderung einer POE-Instanz an, wenn die beiden folgenden Punkte zutreffen:

1. Die Vergleichsbasis wurde mit einer Version vor V4.116 erzeugt.
2. Der aufgerufene POE-Typ enthält Umlaute im Namen.

Bedingung:

Siehe oben.

Abhilfe:

In Bausteinnamen nur Leerzeichen und Zeichen aus der folgenden Liste verwenden:

- 0 1 2 3 4 5 6 7 8 9
 - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 - a b c d e f g h i j k l m n o p q r s t u v w x y z
 - \$ % & () * + - / : ; < = > ? \ ^ _ ` { | }

- **Auswirkung:**

Der Versionsvergleich einer Konfiguration, die mit SILworX vor V9 generiert wurde und einer Konfiguration, die mit SILworX ab V9 generiert wurde zeigt die Meldung «Die Reihenfolge der Variablen- und Instanzendeklaration hat sich geändert.» nicht mehr an:

1. Für alle Funktionen und Funktionsbausteine
2. Für Programme, die in der ab SILworX V9 generierten Konfiguration mindestens eine der folgenden Standardfunktionen verwenden: ADD, SUB, MUL, DIV, MOD, MOVE, AND, OR, XOR, NOT, SHL, SHR, alle ANY-to-Funktionen, ADD_TIME, SUB_TIME, MAX, MIN, SEL, MUX, GT, LT, GE, LE, EQ, NE, PACK.

Bedingung:

In SILworX vor V9 eine Ressource mit zwei Konfigurationen (erste Konfiguration: geladen oder importiert, zweite Konfiguration: generiert) erstellen, deren Versionsvergleich (zu Recht) eine Änderung der Deklarationsreihenfolge in einer POE meldet.

Der Projektstand muss dem der generierten Konfiguration entsprechen.

Das Projekt nach SILworX V9 konvertieren und Code generieren.

Im Versionsvergleich fehlt die Reihenfolge-Meldung, obwohl die betreffende POE den gleichen CRC-Unterschied aufweist wie zuvor.

Abhilfe:

Wenn der zur Vergleichsbasis (z. B. zur geladenen Konfiguration) passende Projektstand noch verfügbar ist:

1. Eine Kopie dieses alten Projektstandes nach SILworX V9 bzw. in die gewünschte, neue Version konvertieren. Dort Code generieren und das Ergebnis über den Startdialog des Versionsvergleichs exportieren.
2. Das andere Projekt, in dem der Versionsvergleich stattfinden soll, in der neuen SILworX Version öffnen, die eben exportierte Konfiguration im Startdialog des Versionsvergleichs importieren und als Vergleichsbasis verwenden.

4.6 Codegenerierung

- **Auswirkung:**
Ändern des CONST-Attributs globaler Variablen nach deren Verwendung im Anwenderprogramm führt zu einem Konflikt.
Wird eine globale Variable als VAR_EXTERNAL verwendet und danach das Attribut CONST gesetzt oder zurückgesetzt, resultiert die Codegenerierung in einem Konflikt, wenn die VAR_EXTERNAL beschrieben wird, aber CONST der globalen Variable TRUE ist.

Bedingung:

Eine globale Variable als VAR_EXTERNAL in der Logik verwenden und das CONST-Attribut der globalen Variable ändern.

Abhilfe:

Die globale Variable an allen Verwendungsstellen löschen. Dadurch wird die zugehörige VAR_EXTERNAL entfernt. Anschließend die globale Variable an allen Verwendungsstellen erneut einfügen.

- **Auswirkung:**
In sehr seltenen Fällen liefert die Codegenerierung ab SILworX V5.30.0 für betroffene Programme einen anderen CRC als die Codegenerierung mit früheren SILworX Versionen.

Der Versionsvergleich zwischen den generierten Konfigurationen aus beiden SILworX Versionen zeigt den CRC-Unterschied der jeweiligen Programm-Binärdatei (*.ldb oder *.dll), sowie davon abhängiger Dateien (Programm-Parameter, ls.config, root.config). Die untergeordneten CRCs der verwendeten Datentypen, POE-Typen und globalen Variablen sind jedoch gleich. In den Detailansichten der Programmdatei und der POEs erscheinen keine Meldungen, die den Unterschied erklären würden.

An diesen Symptomen ist nicht zu erkennen, welche Ursache und Auswirkungen der Unterschied hat.

Bedingung:

Ein Projekt mit einer SILworX-Version vor V5.30.0 erstellen, Code für die enthaltenen Ressourcen generieren. Eine Kopie des Projekts in eine SILworX-Version ab V5.30.0 konvertieren und erneut Code generieren.

Die in beiden SILworX-Versionen generierten Konfigurationen vergleichen.

Abhilfe:

Keine.

- **Auswirkung:**
Im Speicherlayout eines Anwenderprogramms (AP) sind die globalen Variablen (GVs), die das AP lesend und schreibend verwendet, in vier benachbarten Gruppen angeordnet:
(a) GVs ohne Retain.
(b) Multisource-GVs¹ ohne Retain.
(c) Multisource-GVs mit Retain.
(d) GVs mit Retain.

An den Grenzen zwischen diesen Gruppen kann es Padding (ungenutzte Bytes) geben. Die Reload-Codegenerierung gibt fälschlicherweise eine Reload-Warnung aus, wenn beim Reload das Padding zwischen (a) und (b) oder zwischen (c) und (d) wegfällt. Die Reload-Warnung bezeichnet das gelöschte (Padding-)Element als "x.10" oder "x.20", wobei x für den Namen des APs steht.

¹ Multisource-GVs haben neben dem AP noch mindestens eine weitere Quelle in einem Protokoll (in SILworX V9: Modbus-Slave V2).

Bedingung:

Beispiel: Ressource mit einem Protokoll des Typs "Modbus-Slave-Set V2" konfigurieren. Globale Variablen (GVs) A mit Typ BOOL und B mit Typ DINT anlegen.

Die vordefinierte Dataview des Modbus-Slave bearbeiten und die GV B (DINT) in deren "Register-Eingänge" ziehen. Speichern.

Das Programm bearbeiten, die GV A und B in die Logik ziehen und bei jedem der damit angelegten Wertfelder eine Linie vom Ausgang zum Eingang ziehen (Rückführungsschleifen). Speichern.

Code dafür generieren und das Ergebnis laden (oder im Versionsvergleich exportieren und wieder importieren).

Das Wertfeld für die GV B mitsamt seinen Linien aus der Logik des Programms löschen. Speichern.

Code für Reload mit der geladenen (bzw. importierten) Konfiguration als Basis generieren.

Abhilfe:

Warnung kann in diesem Fall ignoriert werden.

4.7 Reload

- Auswirkung:
 1. Wenn beim Zuweisen einer Vergleichskonfiguration «Abbruch» gewählt wurde, kann eine unverständliche Fehlermeldung angezeigt werden.
 2. Die Benutzerführung bei der Reload-Codegenerierung mehrerer Ressourcen und dem Import einer Config-Datei macht erst nach der Bestätigung der Config-Datei mit der Reload-Codegenerierung weiter.

Bedingung:

Die Reload-Codegenerierung für eine Konfiguration mit mehreren Ressourcen starten, die eine Backup- oder Import-Config enthalten und im Auswahldialog abbrechen.

Abhilfe:

Keine.

- Auswirkung:

Wenn eine neu eingefügte POE-Instanz den Namen einer zuvor gelöschten POE-Instanz bekommt, hat dies - zum Teil kritische - Auswirkungen auf den Reload-Prozess:

Fall 1: Die neu eingefügte POE-Instanz startet nach dem Reload mit den letzten Prozesswerten (und Force-Daten) der gelöschten POE-Instanz gleichen Namens. Dies kann zu unerwarteten Ergebnissen und zu unbeabsichtigtem Verhalten der Anlage führen.

Fall 2: Die Übernahme der Prozesswerte in die eingefügte Funktionsinstanz bleibt folgenlos, da eine Funktionsinstanz immer rechtzeitig mit ihren Initialwerten belegt wird, bevor sie zur Ausführung kommt. (Auch das Lesen ihrer Ausgänge vor der Ausführung liefert immer die Initialwerte.) Funktionsinstanzen haben außerdem keine Force-Daten.

Fall 3: Die Reload-Codegenerierung meldet einen Fehler.

Bedingung:

Fall 1:

- A ist in dieser Beschreibung ein benutzerdefinierter Funktionsbaustein-Typ oder ein Programm.

- B ist in dieser Beschreibung ein Standard- oder benutzerdefinierter Funktionsbaustein-Typ. Der Anwender löscht eine Instanz von B aus der Logik von A und fügt später an derselben oder einer anderen Stelle eine neue Instanz von B in die Logik von A ein. Das Problem entsteht, wenn die eingefügte Instanz den gleichen Namen erhält wie die gelöschte und wenn der Anwender diese Namensgleichheit und deren Folgen nicht beabsichtigt hatte. Der Instanzname wird zunächst vom PADT automatisch vergeben und kann vom Anwender nachträglich verändert werden, wobei auch der Name der gelöschten Instanz eingegeben werden kann.
- Zwischen der gelöschten und eingefügten Instanz bestehen keine Unterschiede, die beim Reload als verbotene Änderung gelten würden.
- Der Anwender führt für diese Logikänderung die Reload-Codegenerierung und den Reload durch.

Fall 2:

- A ist in dieser Beschreibung ein benutzerdefinierter POE-Typ oder ein Programm.
 - B ist in dieser Beschreibung ein Funktionstyp.
- Sonst wie Fall 1.

Fall 3:

Wie Fall 1 oder 2, aber die Instanzen unterscheiden sich auf eine beim Reload verbotene Weise.

Abhilfe:

Die Fälle 1 und 2 sind nach der Codegenerierung wie folgt zu erkennen: In der Detailansicht für den POE-Typ A im Versionsvergleich erscheint die eingefügte Instanz nicht als «neu», sondern als «geändert», «verschoben» (bzgl. Abarbeitungsreihenfolge) oder gar nicht.

4.8 Protokolle

- Auswirkung:
Falscher Standardwert für Timing-Master nach Projektkonvertierung auf V6.

Bedingung:

In einem Projekt vor V6 eine safeethernet Verbindung mit zwei Ressourcen (Res1, ID = 1 und Res2, ID = 2) erzeugen und in einer aktuellen Version die Verbindung auf «Ab V6» einstellen. Das kann zur Folge haben, dass Res2 zum Timing-Master ernannt wird. Aus Anwendersicht ist keine Vorhersage möglich, welche der beiden Ressourcen beim Übergang zu «Ab V6» zum Timing-Master ernannt wird.

Abhilfe:

Timing-Master explizit festlegen.

- Auswirkung:
Nach der Konvertierung auf SILworX ab V10 sind safeethernet Verbindungen im Projekt gestört.
Anstatt der System-ID wird nur "?" angezeigt.
Beispiel: «?.0.1 (192.168.1.206:6010)».

Bedingung:

Das Problem entsteht, wenn bei der Konvertierung von Projekten vor SILworX V10 konfigurationsübergreifende Verbindungen mit «/» im Ressource-Namen von X-OPC-Server existieren.

Den Dialog «Eigenschaften» der safeethernet Verbindung öffnen und im Register «Partner» die Felder «IF Kanal 1» und «IF Kanal 2» ansehen.

Abhilfe:

Die safeethernet Verbindungen können korrigiert werden, wenn in der safeethernet

Verbindungsübersicht zuerst der Partner auf «keine» und danach der Partner auf «neu» umgestellt wird.

4.9 Projekt

- **Auswirkung:**

Projekte können durch die Windows-interne Synchronisierung mit Netzlaufwerken auf dem Netzlaufwerk verloren gehen. Die Projekte sind dann, unter Umständen, nur noch lokal verfügbar.

Bedingung:

Ein Projekt in einem synchronisierten Ordner öffnen, bearbeiten und schließen.
Anschließend ist das Projekt vom Netzlaufwerk gelöscht.

Abhilfe:

Projekte nicht öffnen, wenn sie synchronisiert werden und eine Netzwerkverbindung besteht.
Durch das Öffnen eines Projekts wird dessen Netzversion von Windows gelöscht. In diesem Fall das lokal gespeicherte Projekt wieder auf das Netzlaufwerk kopieren.

- **Auswirkung:**

SILworX kann terminieren, wenn im Programmname ein geschütztes Trennzeichen und Leerzeichen in Kombination verwendet werden.

Bedingung:

Im Strukturbaum für eine Ressource ein Programm «P 1» anlegen. Für dieselbe Ressource ein weiteres Programm «P- 1» anlegen, wobei bei diesem Namen vor dem Leerzeichen ein durch Alt-0173 eingebautes geschütztes Trennzeichen steht. Dann stürzt SILworX ab.

Abhilfe:

Keine.

- **Auswirkung:**

In SILworX V4 konnte es bei Löschaktionen dazu kommen, dass in der Datenbank Objekte übrig blieben, die nicht mehr bearbeitet werden konnten. Diese Objekte hatten keinen Einfluss auf das Projekt, wurden aber bei «Prüfe Projektintegrität» gemeldet.

Bedingung:

Projekte der SILworX Versionen V4 und V5, die solche übriggebliebenen Objekte enthalten, sind mit großer Wahrscheinlichkeit nicht in die Versionen V6 und V7 konvertierbar. Die Wahrscheinlichkeit ist besonders groß, wenn die Projekte benutzerdefinierte Datentypen enthalten.

Abhilfe:

Die mit «Prüfe Projektintegrität» gefundenen Objekte sind vor der Konvertierung zu entfernen. Dazu in der alten SILworX Version die folgenden Schritte durchführen:

1. Alle Knoten im Strukturbaum, die sich unterhalb des «Projekt»-Knotens befinden, archivieren, mit Ausnahme des Knotens «Programmiergerät».
2. In der alten SILworX Version ein neues Projekt erstellen.
3. Im neuen Projekt den Knoten «Konfiguration» löschen.
4. Die unter 1. archivierte Konfiguration und - falls vorhanden - weitere Knoten im neuen Projekt unter dem «Projekt»-Knoten wiederherstellen.

Das so erzeugte Projekt sollte in die aktuelle SILworX Version konvertierbar sein.

- **Auswirkung:**

Nach der Konvertierung auf SILworX ab V10 sind mono safeethernet Verbindungen nicht korrekt.

Fehlermeldung aufgrund nicht auflösbarer Ziele auf Objekte (z. B. zweiter safeethernet «IF-Kanal»), die in einer Version vor V10 gelöscht wurden. Diese Objekte werden nach der Konvertierung wieder sichtbar und müssen manuell entfernt werden.

Das kann z. B. ein gelöscht Prozessormodul oder Kommunikationsmodul sein, das in einem safeethernet «IF-Kanal» genutzt wurde. Dieser Kanal zeigt in V10 den Verweis «?» auf das nicht mehr verfügbare Modul. In anderen Fällen wird ein Pfad auf das verwiesene Objekt angezeigt.

Bedingung:

In einer Version vor SILworX V10 (z. B. V9.36.0) ein neues Projekt mit zwei Ressourcen anlegen. Jeder Ressource ein Prozessormodul und ein Kommunikationsmodul hinzufügen. Zwischen den Ressourcen eine safeethernet Verbindung erzeugen und beide Module verwenden. Dann Kommunikationsmodul oder Prozessormodul löschen.

Der Verweis auf den Kanal scheint verschwunden zu sein. Danach das Projekt nach V10 konvertieren. Der Verweis wird mit einem «?» als System-ID trotzdem angezeigt, da das Ziel nicht auflösbar ist.

Abhilfe:

Den Verweis manuell korrigieren oder auf «Keine» setzen. Im Fall eines zweiten safeethernet «IF-Kanal», den zweiten «IF-Kanal» manuell deaktivieren.

4.10 Dokumentation

- Auswirkung:
Die Querverweise von Struct- oder Array-Elementen werden nicht ausgedruckt.

Bedingung:

Struct- oder Array-Element verwenden, Querverweise in der Dokumentation prüfen und mit den Querverweisen in SILworX vergleichen.

Abhilfe:

Keine.

4.11 Benutzerverwaltung

- Auswirkung:
Beim Wiederherstellen eines Benutzerverwaltungs-Archivs, das mit V9 erstellt wurde, wird ein enthaltener Standardbenutzer nicht berücksichtigt. Der Anwender muss sich explizit mit Benutzername und Passwort anmelden.

Bedingung:

- Neues Projekt mit Benutzerverwaltung anlegen.
- Standardbenutzer festlegen.
- Benutzerverwaltung archivieren.
- Vorhandene Benutzerverwaltung im Projekt entfernen.
- Archivierte Benutzerverwaltung wiederherstellen.

Abhilfe:

Keine.

4.12 Lizenzen

- Auswirkung:
Unter Umständen werden beim Wechsel auf eine neuere Windows-Version oder ein Update innerhalb einer Windows-Version (vornehmlich langlaufende Windows-Versionen wie z. B.

Win10) Lizenz-Dongles auf U3ID-Basis ungültig.

Bedingung:

SILworX wird mit einer Hardlock-Lizenz auf Basis einer U3ID betrieben. Beim Update einer Windows-Version oder dem Wechsel auf eine neuere Windows-Version, z. B. von Win7 nach Win10 kann es vorkommen, dass die Hardlock-Lizenz ungültig wird, da die auf U3ID-Basis erstellte Hardlock-Lizenz plötzlich mit einem anderen Wert ausgelesen wird.

Abhilfe:

Es muss eine neue Hardlock-Lizenz auf Basis U3ID erzeugt werden, speziell für neuere Windows-Versionen.

- **Auswirkung:**

Die Meldung im Logbuch beim Anfordern einer Lizenz ist irreführend.

Bedingung:

Lizenz anfordern.

Abhilfe:

Laden Sie die Datei

«C:/Users/Administrator/Downloads/silworx_lic_request_3467898754.olcml» in das Extranet auf der HIMA Website. Bei Problemen kontaktieren Sie support@hima.com.

4.13 SILworX API

- **Auswirkung:**

Über die SILworX API kann der Knoten «PADT-Benutzerverwaltung» im Projektbaum nicht archiviert werden, auch wenn man als «Sicherheitsadministrator» eingeloggt ist.

Bedingung:

Anwender, die Projekte über die SILworX API mit Zugriffsart «Sicherheitsadministrator» geöffnet haben, können den Knoten «PADT-Benutzerverwaltung» des Projektbaums nicht archivieren.

Abhilfe:

Keine.

- **Auswirkung:**

Die Binärstabilität von Archiven ist in den folgenden Fällen nicht gewährleistet:

- Nach dem Konvertieren von Projekten zwischen unterschiedlichen SILworX Versionen.
- Wenn Verweisziele so geändert werden, dass die verweisenden Stellen angepasst werden müssen.
- Wenn in einem Editor Änderungen gespeichert und diese wieder rückgängig gemacht werden.

Bedingung:

- Projekte zwischen unterschiedlichen SILworX Versionen konvertieren.
- Verweisziele ändern, z. B. ein Datentyp umbenennen. Alle globalen Variablen, die diesen Datentyp verwenden, werden damit implizit geändert.
- Änderungen speichern und rückgängig machen.

Abhilfe:

Keine.

5 Besonderheiten

Bei der Benutzung von SILworX sind die beschriebenen Besonderheiten zu beachten.

- **Auswirkung:**
Im Hardware-Editor werden Skalierungseinstellungen eines Analogwerts als REAL gelesen. SILworX liest die eingegebenen Werte für die Stützpunkte eines Analogwerts (bei 4 mA und 20 mA) als REAL. Die Weiterverarbeitung erfolgt dagegen als LREAL. Auch im Anwenderprogramm kann LREAL benutzt werden. Die Einschränkung hat aber nur bei sehr großen oder sehr kleinen Stützpunkt-Werten Auswirkungen.

Bedingung:

Durch die Verwendung von extrem kleinen oder großen Stützpunkt-Werten kann die Genauigkeit des Prozesswertes schlechter werden.

Abhilfe:

Rohwerte im Anwendungsprogramm verarbeiten.

- **Auswirkung:**
Logische Verknüpfungen von BOOL-Variablen, deren Werte von Fremdsystemen stammen, können andere als die erwarteten Ergebnisse liefern.

Bedingung:

Die Ursache ist die Codierung der BOOL-Werte im externen System, die von der im HIMA System abweicht.

Abhilfe:

Es gibt zwei Möglichkeiten der Abhilfe:

- Das externe System liefert definiert nur die Werte 0 für FALSE und 1 für TRUE.
- Ins Anwenderprogramm wird für alle entsprechenden BOOL-Variablen eine Korrekturschaltung eingefügt, die den Wert auf 0 oder 1 normiert:
unnormierte Variable -> Baustein AtoByte -> Baustein AtoBOOL -> normierte Variable.

- **Auswirkung:**
Bei Berechnungen mit Variablen vom Datentyp REAL oder LREAL können die Zykluszeiten insbesondere bei Verwendung von trigonometrischen Funktionen extrem schwanken.

Bedingung:

Siehe oben.

Abhilfe:

Für die Bemessung der Watchdog-Zeit ist es notwendig, die Zykluszeit unter realistischen Bedingungen zu bestimmen, siehe Kapitel «Watchdog-Zeit durch Test ermitteln» im jeweiligen Sicherheitshandbuch.

- **Auswirkung:**
Online-Test und Offline-Simulation zeigen den Wert von Systemvariablen des Anwenderprogramms nicht an:
 - OLT-Feld ist leer.
 - Der Wert von digitalen Systemvariablen ist nicht durch die Farbe der entsprechenden Linien dargestellt.
 - Die Spalte «Prozess-Wert» im Register «System-Variablen» der Objektauswahl ist leer.
 - Der Force-Editor enthält keine Systemvariablen.

Bedingung:

Siehe oben.

Abhilfe:

Die meisten Informationen sind an anderer Stelle, z. B. im Control Panel, ersichtlich. Zur Anzeige im OLT-Feld die Systemvariable mit einer Variablen verbinden und an die Variable ein OLT-Feld anschließen.

Forcen ist nur möglich, wenn die Systemvariable mit einer Variablen verbunden ist.

▪ **Auswirkung:**

SILworX behandelt Wertänderungen von VAR_INPUT-Variablen bei benutzerdefinierten Funktionsbausteinen unterschiedlich:

- Wenn die Eingänge mit Variablen von Standard-Datentypen verbunden werden, wird der Wert der Variablen an eine baustein-lokale Kopie übergeben (Call-by-Value).
- Wenn die Eingänge mit Variablen von benutzerdefinierten Datentypen verbunden werden, wird die Referenz auf die Variable übergeben (Call-by-Reference).

Bedingung:

Dieses Verhalten kann zu Fehlern führen, wenn alle folgenden Bedingungen zutreffen:

- Die Quelle einer VAR_INPUT-Variable ist eine VAR_EXTERNAL-Variable.
- Die selbe Quelle der VAR_INPUT-Variable wird im aufgerufenen Funktionsbaustein gleichzeitig als VAR_EXTERNAL-Variable verwendet.

Wenn der Wert der VAR_EXTERNAL-Variablen im Funktionsbaustein verändert wird, hat anschließendes Lesen der entsprechenden VAR_INPUT-Variablen im Funktionsbaustein folgende Auswirkungen:

- Bei benutzerdefiniertem Datentyp werden die aktuellen Werte gelesen.
- Bei elementarem Datentyp werden die alten Werte gelesen, die bei Beginn der Abarbeitung der Bausteininstanz gültig waren.

Abhilfe:

VAR_EXTERNAL-Variablen sollten nicht gleichzeitig als Quelle einer VAR_INPUT von Instanzen dieser POE verwendet werden.

▪ **Auswirkung:**

Das Dokumentenmanagement kann den Inhalt der Online-Hilfe einer benutzerdefinierten POE nicht ausdrucken.

Bedingung:

Keine.

Abhilfe:

Anzeigen der Online-Hilfe und Ausdrucken der einzelnen Themen aus Windows.

▪ **Auswirkung:**

Es kann nicht ausgeschlossen werden, dass sich zwischen SILworX Versionen Schlüsselbegriffe in den Export-/Import-Dateien (.CSV, .XML) ändern. Dadurch importiert SILworX die entsprechenden Daten als Standardwert und gibt eine Fehlermeldung aus.

Bedingung:

Beispiel: In Versionen vor V5 ist der Datentyp für die Spracheinstellung English als «Data Type» gekennzeichnet, ab V5 als «Data type». Beim Import einer Export-Datei einer Version vor V5 legt SILworX alle Variablen mit dem Standard-Datentyp BOOL an.

Abhilfe:

In den zu importierenden Dateien die entsprechenden Schlüsselwörter anpassen.

▪ **Auswirkung:**

Wenn bei bestehendem System-Login die Diagnoseansicht geöffnet und die Verbindung getrennt wird, dann bietet SILworX beim Versuch, die Verbindung neu aufzubauen, den Modul-Login an.

Bedingung:

Den Hardware-Login durchführen.

Die Detailansicht zum Modul öffnen.

Die Diagnose zu diesem Modul in einem zweiten Fenster öffnen.

Anschließend die Verbindung trennen.

Die Diagnose bietet nach Verbindungsverlust nur den Modul-Login an.

Die Detailansicht bietet nur den System-Login an.

Abhilfe:

Wenn der Modul-Login-Dialog für die Diagnose einmal geöffnet wurde, müssen die Diagnose und die Modul-Online-Ansicht geschlossen und wieder geöffnet werden. Erst danach werden sie wieder über das System ausgelesen.

6 Upgrade von einer Vorversion

Projektdaten aus vorherigen SILworX Versionen können in V12.28 weiterverwendet werden.

HIMA achtet bei der Weiterentwicklung von SILworX darauf, dass Codegenerierungen mit einer neuen SILworX Version nur in Ausnahmefällen zu geänderten Konfigurations-CRCs führen. Diese Fälle werden in den Release-Notes jeder SILworX Version dokumentiert.

HIMA unterstützt Konvertierungen ab V6.114. Versionen vor V6.114 werden von HIMA nicht mehr supported.

Das Upgrade von einer Version ab V6.114 auf V12.28 ist folgendermaßen durchzuführen:

1. Eine Sicherung (Archiv) des vorhandenen Projekts erstellen.
2. Für alle Ressourcen Code generieren.
3. Alle Codes exportieren und im Versionsvergleich importieren. Durch die Konvertierung entstandene Abweichungen können im Versionsvergleich festgestellt werden.
4. Weitere Sicherung (Archiv) erstellen.
5. Projekt in V12.28 öffnen und konvertieren.
6. Nach der Konvertierung die Projektintegrität prüfen.
7. In V12.28 eine Codegenerierung durchführen und feststellen, ob Fehler auftreten. Im Versionsvergleich mit der importierten Konfiguration prüfen, ob sich CRCs verändert haben.
8. Erkannte Fehler beseitigen und erneut Code generieren, um CRC-Änderungen festzustellen.
9. Liegen keine CRC-Änderungen vor, ist die Konvertierung erfolgreich beendet.
10. Liegen CRC-Änderungen vor, prüfen, ob diese akzeptabel sind. Sind die Änderungen akzeptabel, ist die Konvertierung erfolgreich beendet.
11. Sind die Änderungen inakzeptabel, mit der entsprechenden Vorversion weiterarbeiten.

Hinweise zur Konvertierung:

- Die Konvertierung kann bei sehr großen Projekten länger dauern.

6.1 Referenzen

- SILworX Online-Hilfe.
- SILworX Erste-Schritte-Handbuch, HI 801 102 D.
- Kommunikationshandbuch, HI 801 100 D.

Release-Notes

HI 801 544 D

Für weitere Informationen kontaktieren Sie:

HIMA Paul Hildebrandt GmbH

Albert-Bassermann-Str. 28
68782 Brühl, Germany

Telefon: +49 6202 709-0

Fax +49 6202 709-107

E-Mail: info@hima.com

Erfahren Sie online mehr über **SILworX**:

 www.hima.com/de/produkte-services