



Handbuch

---

**HIMax<sup>®</sup>**

---

Systemhandbuch

---



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](mailto: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](mailto:info@hima.com)

Revisions-index	Änderungen	Art der Änderung	
		technisch	redaktionell
9.00	Neu: Typische Reaktionszeit, Fehlerbehandl. E/A-Module, Trennung v. Protokollen, Reservezeiten, Geändert: Benutzerverwaltung	X	X
10.00	Geändert: Berechnung der Max. Dauer der Konfigurationsverbindungen [ms]. Standardwert für Maximale Systembus-Latenzzeit [µs] auf <i>System-Standardwerte</i> gesetzt.	X	X
11.00	Geändert: Kapitel Benutzerverwaltung, Kapitel Forcen Neu: Kapitel MultiForcen	X	X
12.00	Geändert: Kapitel Systembus (Max. Systembus-Latenzzeit) Neu: LB-Austastung Neu: Application Programming Interface (API)	X	X

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
1.1	Aufbau und Gebrauch der Dokumentation	9
1.2	Zielgruppe	9
1.3	Darstellungskonventionen	10
1.3.1	Sicherheitshinweise	10
1.3.2	Gebrauchshinweise	11
1.4	Safety Lifecycle Services	12
<b>2</b>	<b>Sicherheit</b>	<b>13</b>
2.1	Bestimmungsgemäße Verwendung	13
2.1.1	Anwendung im Ruhestromprinzip	13
2.1.2	Anwendung im Arbeitsstromprinzip	13
2.1.3	Einsatz in Brandmelderzentralen	13
2.1.4	Explosionsschutz	13
2.2	ESD-Schutzmaßnahmen	14
2.3	Restrisiken	14
2.4	Sicherheitsvorkehrungen	14
2.5	Notfallinformationen	14
2.6	Automation Security	15
2.6.1	Produkteigenschaften	15
2.6.2	Risikoanalyse und Planung	16
<b>3</b>	<b>Produktbeschreibung</b>	<b>17</b>
3.1	Basisträger und Basisträger-Typen	19
3.1.1	Aufbau eines Basisträgers	20
3.1.2	Belüftung	21
3.1.3	Überwachung der Temperatur	21
3.1.4	Spannungsversorgung	21
3.1.4.1	Abschätzung der benötigten Leistung	22
3.2	Systembus	22
3.2.1	Systembus mit Linienstruktur	24
3.2.2	Systembus mit Netzstruktur	25
3.2.3	Erweiterung des Systembusses, Systembus-Latenzzeit	28
3.2.3.1	Standardwerte der maximalen Systembus-Latenzzeit	29
3.2.3.2	Systembus-Ausdehnung bei Standardeinstellung der maximalen Latenzzeit	30
3.2.3.3	Maximaler Abstand zwischen Prozessormodulen	31
3.2.3.4	Berechnung einer anwenderspezifischen maximalen Systembus-Latenzzeit	32
3.2.3.5	Berechnungsbeispiel für eine anwenderspezifische maximale Latenzzeit	34
3.3	Module und Connector Boards	37
3.3.1	Identifikation der Module über SRS	38
3.3.2	Zulässige Belegung von Steckplätzen	38
3.3.2.1	Zulässige Steckplätze für Prozessormodule	38
3.3.3	Fehlerbehandlung bei E/A-Modulen	40
3.4	Prozessormodul	41
3.4.1	Betriebssystem	41
3.4.1.1	Genereller Ablauf des Zyklus	41
3.4.1.2	Zustände des Betriebssystems	41

3.4.2	Verhalten bei Fehlern	43
3.4.3	Prozessormodul X-CPU 31	43
<b>3.5</b>	<b>Störaustastung</b>	<b>44</b>
3.5.1	Wirkung der Störaustastung	44
3.5.2	Einstellung der Störaustastung	44
3.5.3	Ablauf der Störaustastung	45
3.5.4	Wirkrichtung der Störaustastung	47
3.5.4.1	Wirkrichtung vom Eingangsmodul zum Prozessormodul (■ 2)	47
3.5.4.2	Wirkrichtung vom Prozessormodul zum Ausgangsmodul (■ 3)	47
3.5.4.3	Wirkrichtung vom Ausgangsmodul zum Prozessormodul (■ 6)	47
3.5.4.4	Ausgangs-Störaustastung (■ 5)	48
3.5.4.5	LB-Austastung	48
<b>3.6</b>	<b>Alarm- und Ereignisaufzeichnung</b>	<b>49</b>
3.6.1	Alarmer und Ereignisse	49
3.6.2	Bildung von Ereignissen	49
3.6.2.1	Ereignisbildung auf dem Prozessormodul	49
3.6.2.2	Ereignisbildung auf SOE-Modulen	49
3.6.2.3	Systemereignisse	50
3.6.2.4	Statusvariable	50
3.6.3	Aufzeichnung von Ereignissen	50
3.6.4	Weitergabe von Ereignissen	50
<b>3.7</b>	<b>Kommunikation</b>	<b>51</b>
3.7.1	ComUserTask (CUT)	51
3.7.2	Lizensierung Protokolle	51
<b>3.8</b>	<b>Kommunikation mit Programmiergeräten</b>	<b>51</b>
<b>3.9</b>	<b>Lizensierung</b>	<b>52</b>
<b>4</b>	<b>Redundanz</b>	<b>53</b>
<b>4.1</b>	<b>Prozessormodul</b>	<b>53</b>
4.1.1	Verminderung der Redundanz	53
4.1.2	Redundanzausrüstung	53
4.1.3	Prozessormodul X-CPU 31	53
<b>4.2</b>	<b>Redundanz von E/A-Modulen</b>	<b>54</b>
4.2.1	Modulredundanz	54
4.2.1.1	Reserve-Module	54
4.2.2	Kanalredundanz	54
4.2.3	Connector Boards für redundante Module	54
<b>4.3</b>	<b>Systembus-Redundanz</b>	<b>54</b>
<b>4.4</b>	<b>Kommunikations-Redundanz</b>	<b>55</b>
4.4.1	safeethernet	55
4.4.2	Standardprotokolle	55
<b>4.5</b>	<b>Spannungsversorgung</b>	<b>55</b>
<b>4.6</b>	<b>Mono-Betrieb</b>	<b>55</b>
<b>5</b>	<b>Programmierung</b>	<b>56</b>
<b>5.1</b>	<b>Anschluss des Programmierwerkzeugs</b>	<b>56</b>
5.1.1	Verwendung der Ethernet-Schnittstellen	56
<b>5.2</b>	<b>Verwendung von Variablen in einem Projekt</b>	<b>56</b>

5.2.1	Variablentypen	57
5.2.2	Initialwert	58
5.2.3	Systemvariablen und Systemparameter	58
5.2.4	Systemparameter der Ressource	59
5.2.4.1	Hinweise zum Parameter <i>Minimale Konfigurationsversion</i>	62
5.2.4.2	Verwendung der Parameter <i>Sollzykluszeit</i> und <i>Sollzykluszeit-Modus</i>	63
5.2.4.3	Maximale Kommunikationszeitscheibe	64
5.2.4.4	Ermitteln der maximalen Dauer der Kommunikationszeitscheibe	64
5.2.4.5	Berechnung der <i>Max. Dauer Konfigurationsverbindungen [ms]</i> $t_{\text{Konfig}}$	65
5.2.5	Systemvariablen eines Racks	67
5.2.5.1	Eingangsvariablen zum Auslesen von Parametern	67
5.2.5.2	Abschließen und Aufschließen der Ressource	72
5.2.6	Systemparameter der Anwenderprogramme	73
5.2.7	Hinweise zum Parameter <i>Codegenerierung Kompatibilität</i>	74
5.2.8	Die lokalen Systemvariablen des Anwenderprogramms	75
5.2.9	Zuordnung zu E/A-Kanälen	76
5.2.9.1	Verwendung digitaler Eingänge	76
5.2.9.2	Verwendung analoger Eingänge	77
5.2.9.3	Verwendung sicherheitsbezogener Zählereingänge	77
5.2.9.4	Verwendung digitaler Ausgänge	78
5.2.9.5	Verwendung analoger Ausgänge	78
5.2.10	Zuordnung zu Kommunikationsverbindungen	78
5.2.11	Ereignisaufzeichnung konfigurieren	79
5.2.11.1	Register <b>Event-Definition BOOL</b>	80
5.2.11.2	Register <b>Event-Definition Skalar</b>	81
5.2.11.3	Register <b>Eigenschaften</b>	83
5.2.11.4	Status von LL, L, N, H, HH bei X-AI 32 01 und X-AI 32 02	83
<b>5.3</b>	<b>Forcen</b>	<b>83</b>
5.3.1	Verwendung von Forcen	84
5.3.2	Per Reload geänderte Zuweisung einer Datenquelle	84
5.3.3	Zeitbegrenzung	85
5.3.4	Einschränkung des Forcens	85
5.3.5	Force-Editor	86
5.3.6	Automatisches Zurücksetzen des Forcens	86
5.3.7	Forcen und skalare Ereignisse	87
5.3.8	MultiForcen	87
5.3.8.1	Ziele von MultiForcen	88
5.3.8.2	Globales MultiForcen	88
<b>5.4</b>	<b>Ablauf des Zyklus</b>	<b>88</b>
5.4.1	Reservezeit zur Watchdog-Zeit und zur Sollzykluszeit	89
5.4.2	Multitasking	91
5.4.3	Multitasking-Mode	94
5.4.4	Typische Reaktionszeit	97
<b>5.5</b>	<b>Laden von Anwenderprogrammen</b>	<b>99</b>
5.5.1	Download	99
5.5.2	Reload	99
5.5.2.1	Bedingungen für die Verwendung von Reload	101
5.5.2.2	Cold Reload	103
5.5.2.3	Grenzen für Reload	104
<b>5.6</b>	<b>Benutzerverwaltung</b>	<b>104</b>
5.6.1	Standard-Zugriffsrechte	106

5.6.1.1	Benutzergruppe «Sicherheitsadministratoren»	106
5.6.2	Zugriffsarten und Berechtigungen	106
5.6.2.1	Zugriffsarten der PADT-Benutzerverwaltung	107
5.6.2.2	Zugriffsarten der PES-Benutzerverwaltung	108
5.6.3	PADT-Benutzerverwaltung anlegen	109
5.6.4	PES-Benutzerverwaltung anlegen	109
<b>5.7</b>	<b>Application Programming Interface (API)</b>	<b>110</b>
<b>6</b>	<b>Diagnose</b>	<b>111</b>
<b>6.1</b>	<b>Leuchtdioden (LEDs)</b>	<b>111</b>
<b>6.2</b>	<b>Diagnosehistorie</b>	<b>111</b>
6.2.1	Diagnosemeldung von E/A-Modulen	112
<b>6.3</b>	<b>Online-Diagnose</b>	<b>113</b>
<b>7</b>	<b>Produktdaten und Dimensionierung</b>	<b>115</b>
<b>7.1</b>	<b>Umgebungsbedingungen</b>	<b>115</b>
<b>7.2</b>	<b>Dimensionierung</b>	<b>115</b>
<b>8</b>	<b>Lebenszyklus</b>	<b>117</b>
<b>8.1</b>	<b>Installation</b>	<b>117</b>
8.1.1	Mechanischer Aufbau	117
8.1.2	Anschluss der Feldebene an E/A-Module	117
8.1.2.1	Beschaltung 1	118
8.1.2.2	Beschaltung 2	119
8.1.2.3	Beschaltung 3	119
8.1.2.4	Beschaltung 4	121
8.1.3	Erdung	122
8.1.3.1	Erdfreier Betrieb	122
8.1.3.2	Geerdeter Betrieb	122
8.1.3.3	CE-konformer Schaltschrankaufbau	122
8.1.3.4	Erdung in der HIMA Steuerung	122
8.1.3.5	HIMax Montage im Gestell	123
8.1.3.6	HIMax Montage im Schwenkrahmen	125
8.1.3.7	Erdungsverbindungen	125
8.1.3.8	Zusammenschaltung der Erdanschlüsse mehrerer Schaltschränke	126
8.1.4	Elektrische Anschlüsse	126
8.1.4.1	Schirmung im Ein-/Ausgangsbereich	126
8.1.4.2	Blitzschutz für Datenleitungen in HIMA Kommunikationssystemen	127
8.1.4.3	Kabelfarben	127
8.1.4.4	Anschluss der Versorgungsspannung	127
8.1.4.5	Anschluss der Feldgeräte und Anschluss der Abschirmung	127
8.1.4.6	Verbinden der Racks	127
8.1.5	Montage eines Connector Boards	128
8.1.6	Montage von FTAs im Schaltschrank	130
8.1.7	Wärmebetrachtung	132
8.1.7.1	Wärmeabfuhr	132
8.1.7.2	Definitionen	132
8.1.7.3	Aufstellungsart	132
8.1.7.4	Eigenkonvektion	133
8.1.7.5	Normhinweis	133
8.1.7.6	Temperaturüberwachung	133

<b>8.2</b>	<b>Inbetriebnahme</b>	<b>134</b>
8.2.1	Schaltschrank in Betrieb nehmen	134
8.2.1.1	Alle Eingänge und Ausgänge prüfen	134
8.2.1.2	Spannungszuschaltung	134
8.2.2	Inbetriebnahme der Steuerung mit X-CPU 01	135
8.2.2.1	Fehlerfälle	136
8.2.3	Inbetriebnahme einer Steuerung mit X-CPU 31	137
8.2.3.1	Fehlerfälle	138
8.2.4	Rack-ID zuweisen	139
8.2.5	Umschalten zwischen Linienstruktur und Netzstruktur	140
8.2.5.1	Umschaltung auf Netzstruktur	140
8.2.5.2	Umschaltung auf Linienstruktur	140
<b>8.3</b>	<b>Wartung und Instandhaltung</b>	<b>141</b>
8.3.1	Störungen	141
8.3.2	Zuschalten der Spannungsversorgung nach Betriebsunterbrechung	141
8.3.3	Anschließen einer redundanten Spannungsversorgung	141
8.3.4	Laden von Betriebssystemen	143
8.3.4.1	Upgrade/Downgrade von Betriebssystemen	144
8.3.5	Reparaturen	145
<b>8.4</b>	<b>Besondere Betriebszustände</b>	<b>146</b>
8.4.1	Mono-Betrieb	146
8.4.2	Start mit nur einem responsible Systembus- oder Prozessormodul	146
8.4.3	Auf die Racks 0 und 1 verteilte Prozessormodule X-CPU 01	147
8.4.4	Prozessormodule mit unterschiedlichen Projektkonfigurationen	147
8.4.5	Autostart bei gestopptem System	147
8.4.6	Kommunikation zwischen zwei Prozessormodulen X-CPU 31	148
<b>9</b>	<b>Dokumentation</b>	<b>149</b>
<b>9.1</b>	<b>Dokumentation HIMax System</b>	<b>149</b>
	<b>Anhang</b>	<b>151</b>
	<b>Applikationsbeispiele</b>	<b>151</b>
	<b>Glossar</b>	<b>153</b>
	<b>Abbildungsverzeichnis</b>	<b>154</b>
	<b>Tabellenverzeichnis</b>	<b>155</b>
	<b>Index</b>	<b>157</b>





# 1 Einleitung

Das Systemhandbuch beschreibt den Aufbau und die Wirkungsweise des sicherheitsbezogenen programmierbaren elektronischen Systems HIMax. Das System ist für sicherheitsbezogene Anwendungen bis SIL 3 (IEC 61508), PL e (EN 13849) und für höchste Verfügbarkeit ausgelegt.

HIMax ist für unterschiedliche Steuerungsaufgaben in der Prozess- und Fabrikautomatisierung einsetzbar.

## 1.1 Aufbau und Gebrauch der Dokumentation

Dieses Systemhandbuch enthält folgende Hauptkapitel:

Einleitung	Führt in das Handbuch ein.
Sicherheit	Informationen zum sicheren Einsatz des HIMax Systems.
Produktbeschreibung	Aufbau des HIMax Systems.
Redundanz	Möglichkeiten zur Erhöhung der Verfügbarkeit.
Programmierung	Wichtige Hinweise zur Erstellung eines Anwenderprogramms.
Diagnose	Zusammenfassende Darstellung der Diagnosemöglichkeiten.
Produktdaten und Dimensionierung	Daten, die das gesamte System betreffen. Daten einzelner Komponenten im jeweiligen Handbuch.
Lebenszyklus	Phasen des Lebenszyklus eines HIMax Systems: <ul style="list-style-type: none"><li>▪ Installation.</li><li>▪ Inbetriebnahme.</li><li>▪ Wartung und Instandhaltung.</li></ul>
Dokumentation	Überblick über die Dokumentation
Anhang	<ul style="list-style-type: none"><li>▪ Konfigurationsbeispiele für HIMax Systeme</li><li>▪ Glossar.</li><li>▪ Bild- und Tabellenverzeichnis.</li><li>▪ Index.</li></ul>

## 1.2 Zielgruppe

Dieses Dokument wendet sich an Planer, Projekteure, Programmierer und Personen, die zur Inbetriebnahme, zur Wartung und zum Betreiben von Automatisierungsanlagen berechtigt sind. Vorausgesetzt werden spezielle Kenntnisse auf dem Gebiet der sicherheitsbezogenen Automatisierungssysteme.

Jedes Fachpersonal (Planung, Montage, Inbetriebnahme) muss über die Risiken und deren mögliche Folgen unterrichtet sein, die im Falle einer Manipulation von einem sicherheitsbezogenen Automatisierungssystem ausgehen können.

Planer und Projekteure müssen zusätzlich Kenntnisse in Auswahl und Verwendung elektrischer und elektronischer Sicherheitssysteme in Anlagen der Automatisierungstechnik haben, um z. B. falsche Anschlüsse oder falsche Programmierung zu vermeiden.

Der Anlagenbetreiber ist für die Qualifikation und Sicherheitseinweisung des Bedien- und Wartungspersonals verantwortlich.

Änderungen oder Erweiterungen an der Verdrahtung des Systems ist nur durch Personal zulässig, das Kenntnisse von Steuer- und Regelungstechnik, Elektrotechnik, Elektronik, Einsatz von PES und ESD-Schutzmaßnahmen hat.

## 1.3 Darstellungskonventionen

Zur besseren Lesbarkeit und zur Verdeutlichung gelten in diesem Dokument folgende Schreibweisen:

<b>Fett</b>	Hervorhebung wichtiger Textteile. Bezeichnungen von Schaltflächen, Menüpunkten und Registern im Programmierwerkzeug, die angeklickt werden können.
<i>Kursiv</i>	Parameter und Systemvariablen, Referenzen.
<i>Courier</i>	Wörtliche Benutzereingaben.
<b>RUN</b>	Bezeichnungen von Betriebszuständen (Großbuchstaben).
Kap. 1.2.3	Querverweise sind Hyperlinks, auch wenn sie nicht besonders gekennzeichnet sind. Im elektronischen Dokument (PDF): Wird der Mauszeiger auf einen Hyperlink positioniert, verändert er seine Gestalt. Bei einem Klick springt das Dokument zur betreffenden Stelle.

Sicherheits- und Gebrauchshinweise sind besonders gekennzeichnet.

### 1.3.1 Sicherheitshinweise

Um ein möglichst geringes Risiko zu gewährleisten, sind die Sicherheitshinweise unbedingt zu befolgen.

Die Sicherheitshinweise im Dokument sind wie folgt dargestellt.

- Signalwort: Warnung, Vorsicht, Hinweis.
- Art und Quelle des Risikos.
- Folgen bei Nichtbeachtung.
- Vermeidung des Risikos.

Die Bedeutung der Signalworte ist:

- Warnung: Bei Missachtung droht schwere Körperverletzung bis Tod.
- Vorsicht: Bei Missachtung droht leichte Körperverletzung.
- Hinweis: Bei Missachtung droht Sachschaden.

#### **SIGNALWORT**



**Art und Quelle des Risikos!**  
**Folgen bei Nichtbeachtung.**  
**Vermeidung des Risikos.**

#### **HINWEIS**



**Art und Quelle des Schadens!**  
**Vermeidung des Schadens.**

### 1.3.2 Gebrauchshinweise

Zusatzinformationen sind nach folgendem Beispiel aufgebaut:

---

**i**

An dieser Stelle steht der Text der Zusatzinformation.

---

Nützliche Tipps und Tricks erscheinen in der Form:

---

**TIPP**

An dieser Stelle steht der Text des Tipps.

---

## 1.4 Safety Lifecycle Services

HIMA unterstützt Sie in allen Phasen des Sicherheitslebenszyklus einer Anlage: Von der Planung, der Projektierung, über die Inbetriebnahme, bis zur Aufrechterhaltung der Sicherheit.

Für Informationen und Fragen zu unseren Produkten, zu Funktionaler Sicherheit und zu Automation Security stehen Ihnen die Experten des HIMA Support zur Verfügung.

Für die geforderte Qualifizierung gemäß Sicherheitsstandards führt HIMA produkt- oder kundenspezifische Seminare in eigenen Trainingszentren oder bei Ihnen vor Ort durch. Das aktuelle Seminarangebot zu Funktionaler Sicherheit, Automation Security und zu HIMA Produkten finden Sie auf der HIMA Webseite.

### Safety Lifecycle Services:

<b>Onsite+ / Vor-Ort-Engineering</b>	In enger Abstimmung mit Ihnen führt HIMA vor Ort Änderungen oder Erweiterungen durch.
<b>Startup+ / Vorbeugende Wartung</b>	HIMA ist verantwortlich für die Planung und Durchführung der vorbeugenden Wartung. Wartungsarbeiten erfolgen gemäß der Herstellervorgabe und werden für den Kunden dokumentiert.
<b>Lifecycle+ / Lifecycle-Management</b>	Im Rahmen des Lifecycle-Managements analysiert HIMA den aktuellen Status aller installierten Systeme und erstellt konkrete Empfehlungen für Wartung, Upgrade und Migration.
<b>Hotline+ / 24-h-Hotline</b>	HIMA Sicherheitsingenieure stehen Ihnen für Problemlösung rund um die Uhr telefonisch zur Verfügung.
<b>Standby+ / 24-h-Rufbereitschaft</b>	Fehler, die nicht telefonisch gelöst werden können, werden von HIMA Spezialisten innerhalb vertraglich festgelegter Zeitfenster bearbeitet.
<b>Logistic+/ 24-h-Ersatzteilservice</b>	HIMA hält notwendige Ersatzteile vor und garantiert eine schnelle und langfristige Verfügbarkeit.

### Ansprechpartner:

<b>Safety Lifecycle Services</b>	<a href="https://www.hima.com/de/unternehmen/ansprechpartner-weltweit/">https://www.hima.com/de/unternehmen/ansprechpartner-weltweit/</a>
<b>Technischer Support</b>	<a href="https://www.hima.com/de/produkte-services/support/">https://www.hima.com/de/produkte-services/support/</a>
<b>Seminarangebot</b>	<a href="https://www.hima.com/de/produkte-services/seminarangebot/">https://www.hima.com/de/produkte-services/seminarangebot/</a>

## 2 Sicherheit

Die Sicherheitsinformationen, Hinweise und Anweisungen in diesem Dokument unbedingt lesen. Das Produkt nur unter Beachtung aller Richtlinien und Sicherheitsrichtlinien einsetzen.

Zum Thema Sicherheit muss das HIMax Sicherheitshandbuch HI 801 002 D beachtet werden.

Dieses Produkt wird mit SELV oder PELV betrieben. Mit dem HIMax Relaismodul X-DO 12 01 können externe Spannungen bis 250 VDC/VAC geschaltet werden. Von dem Produkt selbst geht kein Risiko aus. Der Einsatz im Ex-Bereich ist nur mit zusätzlichen Maßnahmen erlaubt.

### 2.1 Bestimmungsgemäße Verwendung

Das Kapitel beschreibt die bestimmungsgemäße Verwendung des sicherheitsbezogenen Automatisierungssystems HIMax.

Das Automatisierungssystem ist ausgelegt für den Prozessmarkt zum Steuern und Regeln von Prozessen, Schutzsystemen, Brennersteuerungen, Maschinensteuerungen und verfahrenstechnischen Anlagen, sowie für die Fabrikautomatisierung. Für die Programmierung, Konfiguration, Überwachung, Bedienung und Dokumentation des Systems HIMax wird das HIMA Programmierwerkzeug SILworX eingesetzt.

Der redundante Betrieb von HIMax Modulen schließt den gleichzeitigen nicht-redundanten Betrieb anderer Module nicht aus.

#### 2.1.1 Anwendung im Ruhestromprinzip

Das HIMax System ist für das Ruhestromprinzip konzipiert.

Ein System, das nach dem Ruhestromprinzip funktioniert, schaltet z. B. einen Aktor aus, um seine Sicherheitsfunktion auszuführen (de-energize to trip).

#### 2.1.2 Anwendung im Arbeitsstromprinzip

Das HIMax System kann in Arbeitsstrom-Anwendungen eingesetzt werden.

Ein System, das nach dem Arbeitsstromprinzip funktioniert, schaltet z. B. einen Aktor ein, um seine Sicherheitsfunktion auszuführen (energize to trip).

Bei der Konzeption des Automatisierungssystems sind die Anforderungen aus den Anwendungsnormen zu beachten, z. B. kann eine Leitungsüberwachung (LS/LB) der Eingänge und Ausgänge oder eine Rückmeldung der ausgelösten Sicherheitsfunktion erforderlich sein.

#### 2.1.3 Einsatz in Brandmelderzentralen

HIMax Systeme mit analogen Eingängen sind für Brandmelderzentralen nach EN 54-2 und NFPA 72 geprüft und zertifiziert.

Die im Sicherheitshandbuch HI 801 002 D aufgeführten Verwendungsbedingungen sind zu beachten.

#### 2.1.4 Explosionsschutz

Das Automatisierungssystem HIMax ist geeignet zum Einbau in die Zone 2.



Die im HIMax Sicherheitshandbuch HI 801 002 D aufgeführten besonderen Bedingungen sind zu beachten!

## 2.2 ESD-Schutzmaßnahmen

Arbeiten am HIMax System muss von Personal durchgeführt werden, das Kenntnisse von ESD-Schutzmaßnahmen besitzt.

### HINWEIS



**Schäden am HIMax System durch elektrostatische Entladung!**

- Für die Arbeiten einen antistatisch gesicherten Arbeitsplatz benutzen und ein Erdungsband tragen.
- Module bei Nichtbenutzung elektrostatisch geschützt aufbewahren, z. B. in der Verpackung.

## 2.3 Restrisiken

Von einem HIMA System selbst geht kein Risiko aus.

Restrisiken können ausgehen von:

- Fehlern in der Projektierung.
- Fehlern im Anwenderprogramm.
- Fehlern in der Verdrahtung.

## 2.4 Sicherheitsvorkehrungen

Am Einsatzort geltende Sicherheitsbestimmungen beachten und vorgeschriebene Schutzausrüstung tragen.

## 2.5 Notfallinformationen

Ein HIMA System ist Teil der Sicherheitstechnik einer Anlage. Der Ausfall einer Steuerung bringt die Anlage in den sicheren Zustand.

Im Notfall ist jeder Eingriff, der die Sicherheitsfunktion des HIMA Systems verhindert, verboten.

## 2.6 Automation Security

HIMA unterscheidet zwischen den Begriffen *Safety* im Sinne der funktionalen Sicherheit und *Security* im Sinne von Schutz eines Systems vor Manipulationen.

Industrielle Steuerungen (PES) müssen gegen IT-typische Problemquellen geschützt werden, z. B.:

- Unzureichender Schutz von IT-Einrichtungen (z. B. offenes WLAN, veraltete Betriebssysteme).
- Fehlendes Bewusstsein für den richtigen Umgang mit Betriebsmitteln (z. B. USB-Stick).
- Direkte Zugänge zu schützenswerten Bereichen.
- Angreifer innerhalb von Betriebsgeländen.
- Angreifer über Kommunikations-Netzwerke innerhalb und außerhalb von Betriebsgeländen.

HIMA Safety-Systeme bestehen aus folgenden zu schützenden Teilen:

- Sicherheitsbezogenes Automatisierungssystem.
- PADT.
- Optionale X-OPC Server (auf einem Host-PC).
- Optionale Kommunikationsverbindungen zu externen Systemen.

### 2.6.1 Produkteigenschaften

HIMax Steuerungen erfüllen bereits in den Grundeinstellungen Anforderungen an Automation Security.

In Steuerungen und im Programmierwerkzeug sind Schutzmechanismen integriert, die versehentliche oder nicht genehmigte Veränderungen verhindern:

- Jede Änderung am Anwenderprogramm oder an der Konfiguration einer Steuerung führt zu einem neuen Konfigurations-CRC.
- In der Steuerung können Online-Änderungen der Sicherheitsparameter deaktiviert werden. Dadurch sind Änderungen der Sicherheitsparameter nur durch Download oder Reload möglich.
- Der Anwender kann eine Benutzerverwaltung einrichten, um die Security zu erhöhen. Hier werden Benutzergruppen, Benutzerkonten, Zugriffsrechte für das PADT und für die Steuerungen (PES) projektbezogen festgelegt. In einer Benutzerverwaltung kann der Anwender definieren, ob für das Öffnen des Projekts und für den Login in eine Steuerung eine Autorisierung erforderlich ist.
- Der Zugang zu Daten einer Steuerung ist nur dann möglich, wenn im PADT das gleiche Anwenderprojekt geladen wurde wie in der Steuerung. Die CRCs müssen identisch sein (Archiv-Pflege!).
- Eine physikalische Verbindung zwischen einem PADT und einer Steuerung (PES) ist im Betrieb nicht notwendig und muss aus Gründen der Security getrennt werden. Das PADT kann für Diagnose- und Wartungszwecke erneut mit der Steuerung verbunden werden.

Die Anforderungen der Normen für Safety und Security sind zu beachten. Die Autorisierung von Personal und die notwendigen Schutzmaßnahmen unterliegen der Verantwortung des Betreibers.

**⚠ WARNUNG**

**Personenschaden durch unbefugte Manipulationen an Steuerungen möglich!**

**Steuerungen sind gegen unbefugte Zugriffe zu schützen:**

- **Standardeinstellungen für Logins und Passworte sind zu ändern.**
- **Zugänge zu Steuerungen und PADTs sind zu kontrollieren!**
- **Weitere Schutzmaßnahmen entnehmen Sie dem Automation Security Handbuch (HI 801 372 D).**

## 2.6.2 Risikoanalyse und Planung

Security ist kein Produkt, sondern ein Prozess. So helfen z. B. gepflegte Netzwerkpläne sicherzustellen, dass sichere Netzwerke dauerhaft von öffentlichen Netzwerken getrennt sind. Sinnvollerweise sollte nur ein definierter Übergang über eine Firewall oder ein eigenständiges Subnetz bestehen.

Eine sorgfältige Planung nennt die erforderlichen Maßnahmen. Nach erfolgter Risikoanalyse sind die benötigten Maßnahmen zu ergreifen, wie z. B.:

- Zugriffsrechte für Benutzergruppen und Benutzerkonten gemäß den vorgesehenen Aufgaben zuweisen.
- Passwörter verwenden, die den Anforderungen an die Security entsprechen.

Ein regelmäßiges Review (z. B. jährlich) der Security-Maßnahmen ist erforderlich.

### **i**

**Die für eine Anlage geeignete Umsetzung der benötigten Maßnahmen liegt in der Verantwortung des Betreibers!**

Weitere Informationen finden Sie im HIMA Automation Security Handbuch HI 801 372 D.



### 3 Produktbeschreibung

HIMax ist ein sicherheitsbezogenes Steuerungssystem, das für Dauerbetrieb und maximale Verfügbarkeit ausgelegt ist.

HIMax ist ein modulares System. Die Funktionen der Verarbeitung, der Kommunikation und der Ein- und Ausgänge sind auf steckbare Module verteilt. Diese werden in einem oder mehreren Basisträgern eingefügt. Durch die Auswahl geeigneter Module lässt sich die HIMax Steuerung auf eine konkrete Anwendung zuschneiden. Die Basisträger werden über Patchkabel (Systembus) miteinander verbunden.

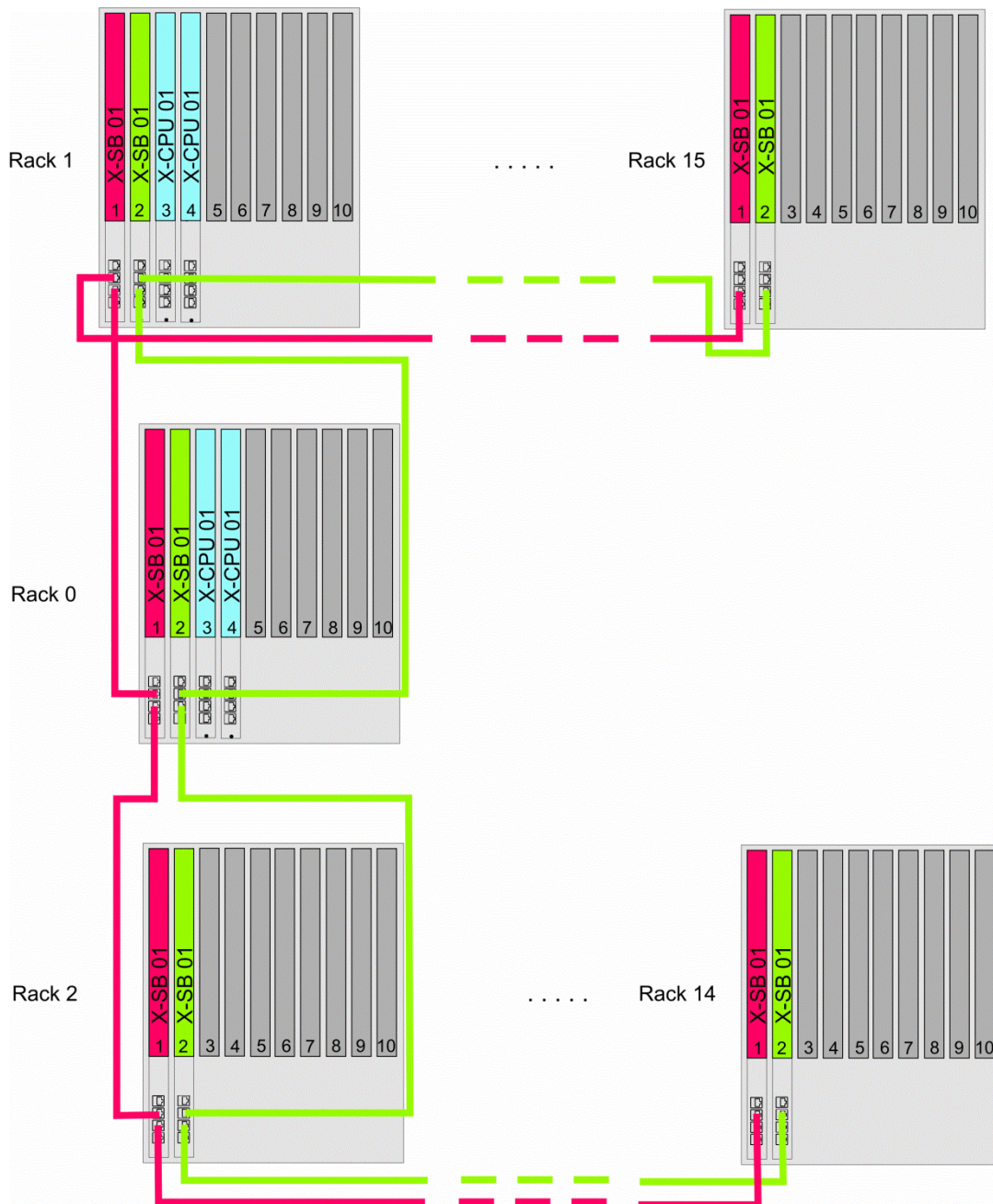
Die Steuerung lässt sich für spätere Erweiterungen des zu steuernden Prozesses leicht anpassen, z. B. durch Hinzufügen von einzelnen Modulen oder von einem oder mehreren Basisträgern mit Modulen.

Den Aufbau des HIMax Systems zeigt Bild 1. Im Bild sind die Basisträger, die beiden Systembusse, die Systembusmodule, die Prozessormodule und die Connector Boards der Module dargestellt.

Zur Erhöhung der Verfügbarkeit ist HIMax für den redundanten Betrieb konzipiert. Näheres siehe Kapitel 3.9.

Das System ist auch als Mono-System ohne Redundanz einsetzbar, siehe Kapitel 3.3.2, Variante 1, und Anhang.

In jedem Fall ist mit den entsprechenden Modultypen ein sicherheitsbezogener Betrieb bis SIL 3 möglich.



Rack 0: Basisträger 0

Rack 1: Basisträger 1

Rack 2: Basisträger 2

Rack 14: Basisträger 14

Rack 15: Basisträger 15

Bild 1: Systemübersicht

Ein HIMax System besteht aus mindestens einem Basisträger, dem Rack 0. Er trägt die Rack-ID (Nummer) 0 und enthält mindestens ein Prozessormodul. Alle weiteren Racks sind Erweiterungs-Racks. Von diesen darf zusätzlich nur Rack 1 maximal zwei Prozessormodule enthalten. Die übrigen Erweiterungs-Racks dürfen kein Prozessormodul enthalten.

Rack 0 ist mit bis zu 15 Erweiterungs-Racks erweiterbar. Die beiden Systembusse A und B aller Racks sind durch Patchkabel (Kapitel 3.2) miteinander verbunden.

**i**

Die HIMax Dokumentation verwendet die Bezeichnungen *Basisträger* und *Rack* in folgenden Fällen:

- *Basisträger* bei der Beschreibung der Hardware eines einzelnen Basisträgers.
- *Rack* bei der Benennung eines Basisträgers innerhalb des Systems.

### 3.1 Basisträger und Basisträger-Typen

HIMax Basisträger unterscheiden sich in der Anzahl ihrer Steckplätze.

Jeder der Basisträger, aus denen eine HIMax Steuerung aufgebaut ist, enthält entweder 10, 15, oder 18 Steckplätze.

Basisträger-Typen:

- Mit 10 Steckplätzen: X-BASE PLATE 10 01, X-BASE PLATE 10 31 für Rückwandmontage, z. B. auf einer Montageplatte.
- Mit 15 Steckplätzen: X-BASE PLATE 15 01, X-BASE PLATE 15 31 für Rückwandmontage.
- Mit 15 Steckplätzen: X-BASE PLATE 15 02, X-BASE PLATE 15 32 für 19-Zoll-Montage.
- Mit 18 Steckplätzen: X-BASE PLATE 18 01, X-BASE PLATE 18 31 für Rückwandmontage.

Jeder der Steckplätze ist mit je einem Modul und einem Connector Board bestückbar.

Die Verbindung zwischen den Basisträgern wird durch Patchkabel (Systembus) hergestellt.

### 3.1.1 Aufbau eines Basisträgers

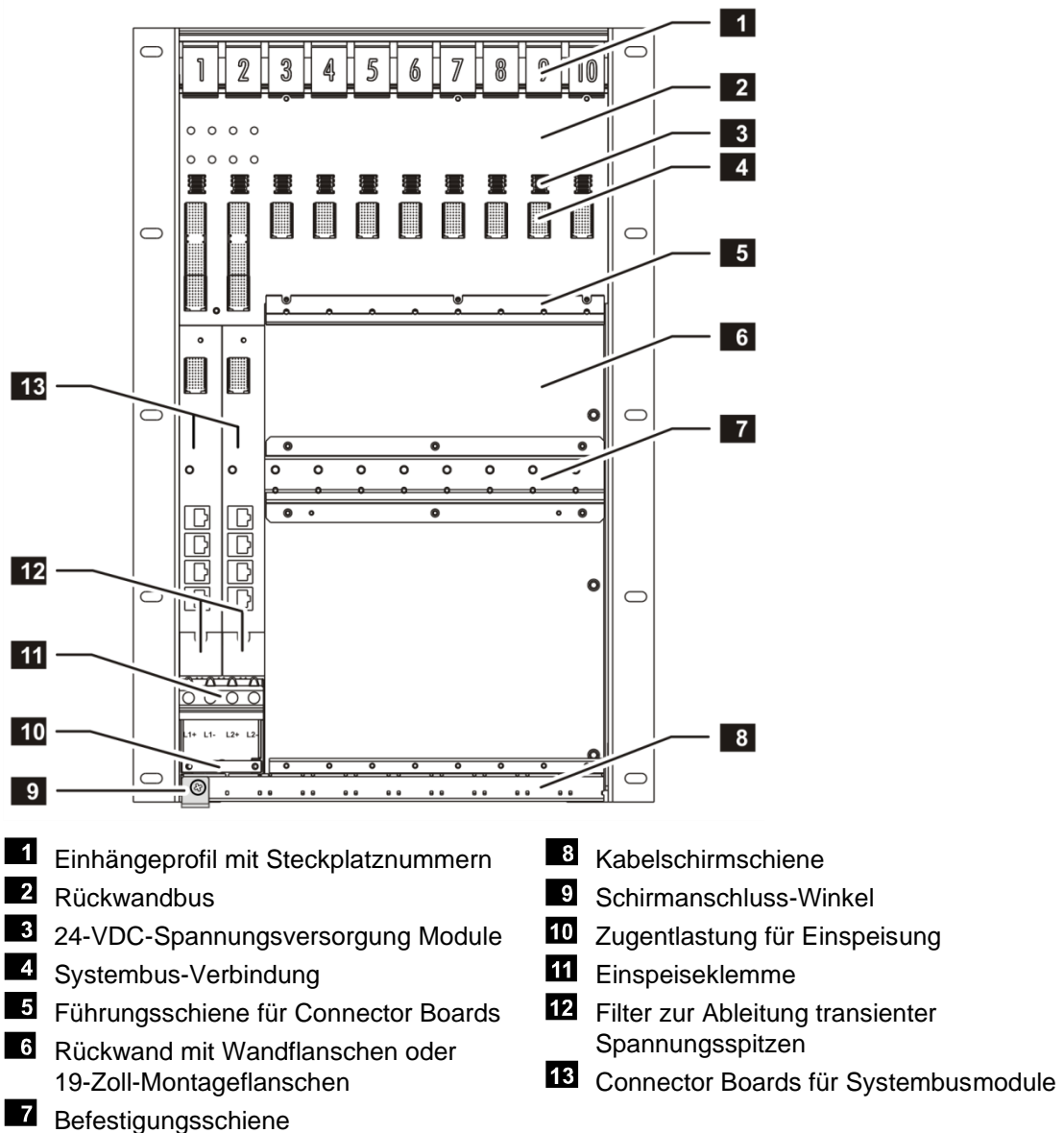


Bild 2: Aufbau eines Basisträgers

Die Steckplätze 1 und 2 sind für Systembusmodule reserviert. Bei den Basisträgern X-BASE PLATE 10 31, X-BASE PLATE 15 31 oder X-BASE PLATE 18 31 sind die Steckplätze 1 und 2 für X-CPU 31 Prozessormodule reserviert. Die übrigen Steckplätze können andere Module aufnehmen, wobei Einschränkungen für die Position von Prozessormodulen zu beachten sind, siehe Kapitel 3.3.2.

Zu jedem Modul gibt es ein Connector Board, an das externe Geräte wie Sensoren, Aktoren und andere Steuerungen angeschlossen werden. Die beiden Connector Boards für die Systembusmodule sind im Lieferumfang des Basisträgers enthalten.

Die Einspeiseklemmen des Basisträgers dienen zum Anschluss der Stromversorgung. Zwei redundante 24-VDC-Stromversorgungen sind anschließbar.

### 3.1.2 Belüftung

Die Belüftung der Module erfolgt durch einen passenden Systemlüfter oberhalb des Racks.

Die Luft strömt von unten durch den Anschlussraum vor den Connector Boards und durch die Module zum Systemlüfter. Für eine einwandfreie Luftführung müssen alle nicht belegten Steckplätze des Racks mit Leermusername ausgerüstet werden.

**HIMA empfiehlt, das HIMax System mit den entsprechenden Systemlüftern (X-FAN) zu betreiben. HIMA Systemlüfter gewährleisten eine hohe Verfügbarkeit des HIMax Systems!**

Für jeden Basisträgertyp gibt es einen entsprechenden Systemlüfter in passender Breite. Die Systemlüfter sind je nach Breite mit 2 bis 4 Lüftern ausgestattet. Näheres im Handbuch X-FAN, HI 801 032 D.

In einem Schaltschrank muss die im Schrank entstehende Verlustwärme beachtet werden. Gegebenenfalls müssen Maßnahmen zur Reduzierung der Verlustwärme getroffen werden z. B. durch Dachlüfter. Eine geringere Umgebungstemperatur erhöht die Lebensdauer und die Zuverlässigkeit der eingebauten Komponenten, siehe Kapitel 8.1.7.1.

### 3.1.3 Überwachung der Temperatur

Die Module überwachen ihre Temperatur selbst. Es ist möglich, den Temperaturzustand im Programmierwerkzeug SILworX anzeigen zu lassen und für die Programmierung von Reaktionen auszuwerten, siehe Kapitel 8.1.7.6.

### 3.1.4 Spannungsversorgung

Für das HIMax System wird eine Spannungsversorgung von 24 VDC vorausgesetzt.

Die sichere elektrische Trennung muss in der 24-V-Spannungsversorgung erfolgen. Nur Spannungsversorgungen in den Ausführungen PELV oder SELV einsetzen. Beim Einsatz gemäß UL-Bestimmungen ist auch ein regelbares Netzgerät mit einer maximalen Spannung von 150 V und einer maximalen Leistung von 10 kVA zulässig.

Die verwendete Spannungsversorgung muss Netzspannungs-Ausfälle bis 20 ms überbrücken. HIMA Spannungsversorgungsgeräte sind geeignet ausgestattet. Vor dem Einsatz von Spannungsversorgungsgeräten anderer Hersteller sind diese entsprechend zu prüfen.

Es ist möglich, zwei redundante Spannungsversorgungen anzuschließen.

**Zur Gewährleistung einer hohen Verfügbarkeit HIMax Systeme mit einer Spannungsversorgung betreiben, für die folgendes gilt:**

- **Die maximale Überspannung am Ausgang der Spannungsversorgung im Fehlerfall beträgt 35 V.**
- **Jeder Basisträger ist durch Versicherungen gegen Ströme von mehr als 63 A abgesichert.**

Die Module überwachen beide Versorgungsspannungen. Der Spannungszustand lässt sich im Programmierwerkzeug SILworX anzeigen und für die Programmierung von Reaktionen auswerten.

### 3.1.4.1 Abschätzung der benötigten Leistung

Die Leistung, für die die Spannungsversorgung ausgelegt werden muss, lässt sich durch folgende Faustformel abschätzen.

$$P_{\text{gesamt}} = n_{\text{CPU}} * 35 + n_{\text{Module}} * 20 + n_{\text{Lüfter}} * 20 + P_{\text{extern}}$$

$P_{\text{gesamt}}$ : Gesamte benötigte Leistung.

$n_{\text{CPU}}$ : Anzahl der eingesetzten Prozessormodule.

$n_{\text{Module}}$ : Anzahl der eingesetzten Module ohne die Prozessormodule.

$n_{\text{Lüfter}}$ : Anzahl der verwendeten Lüfter. Ein Systemlüfter enthält 2 ... 4 Lüfter.

$P_{\text{extern}}$ : Leistung, die von Ausgangsmodulen an die angeschlossenen Aktoren abgegeben wird.

Die Formel enthält folgende typischen Leistungen:

- Leistungsaufnahme eines HIMax Prozessormoduls ca. 35 W.
- Leistungsaufnahme weiterer HIMax Module ca. 20 W (außer Prozessormodul).
- Leistungsaufnahme eines Lüfters ca. 20 W.
- Leistungsaufnahme der an die Ausgangsmodule angeschlossenen Aktoren, die durch die Ausgangsmodule versorgt werden.

Der errechnete Wert ergibt eine grobe Abschätzung der benötigten Leistung eines HIMax Systems in Watt.

Für eine exakte Bestimmung der benötigten Leistung sind die Werte der Leistungsaufnahme der einzelnen Module den entsprechenden Handbüchern zu entnehmen. Die Leistungsaufnahme der anderen Verbraucher ist deren Handbüchern zu entnehmen.

## 3.2 Systembus

Das HIMax System arbeitet mit zwei redundanten Systembussen, Systembus A und Systembus B.

Die Systembusse verlaufen innerhalb eines Basisträgers auf der Rückwandbus-Leiterplatte. Das Einstecken eines Moduls in den Basisträger verbindet es mit den Systembussen. Die Systembusse A und B verbinden die Module untereinander über die Systembusmodule. Bei Ausfall eines Moduls bleiben die Systembus-Verbindungen zu den übrigen Modulen intakt.

Die Systembus-Anschlüsse der Module haben eine galvanische Trennung zum Basisträger. Eine Isolationsspannung von mindestens 1500 V zwischen Prozessormodul und jedem Eingangs- und Ausgangsmodul ist gewährleistet.

Zur Verwaltung eines Systembusses ist ein Systembusmodul notwendig. Das Systembusmodul in Steckplatz 1 betreibt den Systembus A und das Systembusmodul in Steckplatz 2 betreibt den Systembus B.



Ist nur ein Systembusmodul in den Basisträger eingefügt, steht nur ein Systembus zur Verfügung!

---

Bei Betrieb des HIMax Systems über beide Systembusmodule läuft die Kommunikation gleichzeitig über beide Systembusse.

Bei einem HIMax System aus mehreren Basisträgern sind die Systembusse der Basisträger mit Patchkabeln zu verbinden. Diese sind in die RJ-45-Buchsen in den Connector Boards der Systembusmodule zu stecken. Die Systembusse A und B dürfen dabei nicht gekreuzt oder verbunden werden.

Es ist unzulässig, die Systembusse mehrerer unterschiedlicher HIMax Systeme miteinander zu verbinden!

**HINWEIS****Betriebsstörungen möglich!**

Die Anschlüsse UP, DOWN und DIAG der Systembusse sind ausschließlich zur Verbindung von HIMax Basisträgern zu verwenden. Die Systembusse können nicht als normale Ethernet-Verbindungen verwendet werden.

Die Anschlüsse UP, DOWN und DIAG dürfen nicht mit lokalen Netzwerken und sonstigen Geräten mit LAN-Anschluss, z. B. PADT, verbunden werden!

Systembus A und Systembus B dürfen auf keinen Fall miteinander verbunden oder gekreuzt werden!

Bei Linienstruktur müssen alle Basisträger entweder redundant oder nicht-redundant betrieben werden!

Basisträger mit Prozessormodulen oder *responsible* Systembusmodulen müssen unabhängig von der Netz- oder Linienstruktur mit redundantem Systembus aufgebaut werden.

Die Patchkabel für die Systembusse müssen folgende Anforderungen erfüllen:

- Mindestens Cat 5e (gemäß IEEE 802.3) für 1 Gbit/s für industrielle Anwendung.
- Industrietaugliche RJ-45-Stecker an beiden Seiten.
- Die Schirmung des Kabels muss mindestens Class D gemäß ISO/IEC 11801 erfüllen.
- Durch Auto-Crossover sind sowohl Kabel mit gekreuzten, als auch mit durchverbundenen Adern zulässig.

Geeignete Patchkabel mit Industriestecker sind bei HIMA in Standardlängen erhältlich.

**HINWEIS****Kommunikationsstörung möglich!**

**Patchkabel nach Industriestandard Cat 5e oder besser verwenden!**

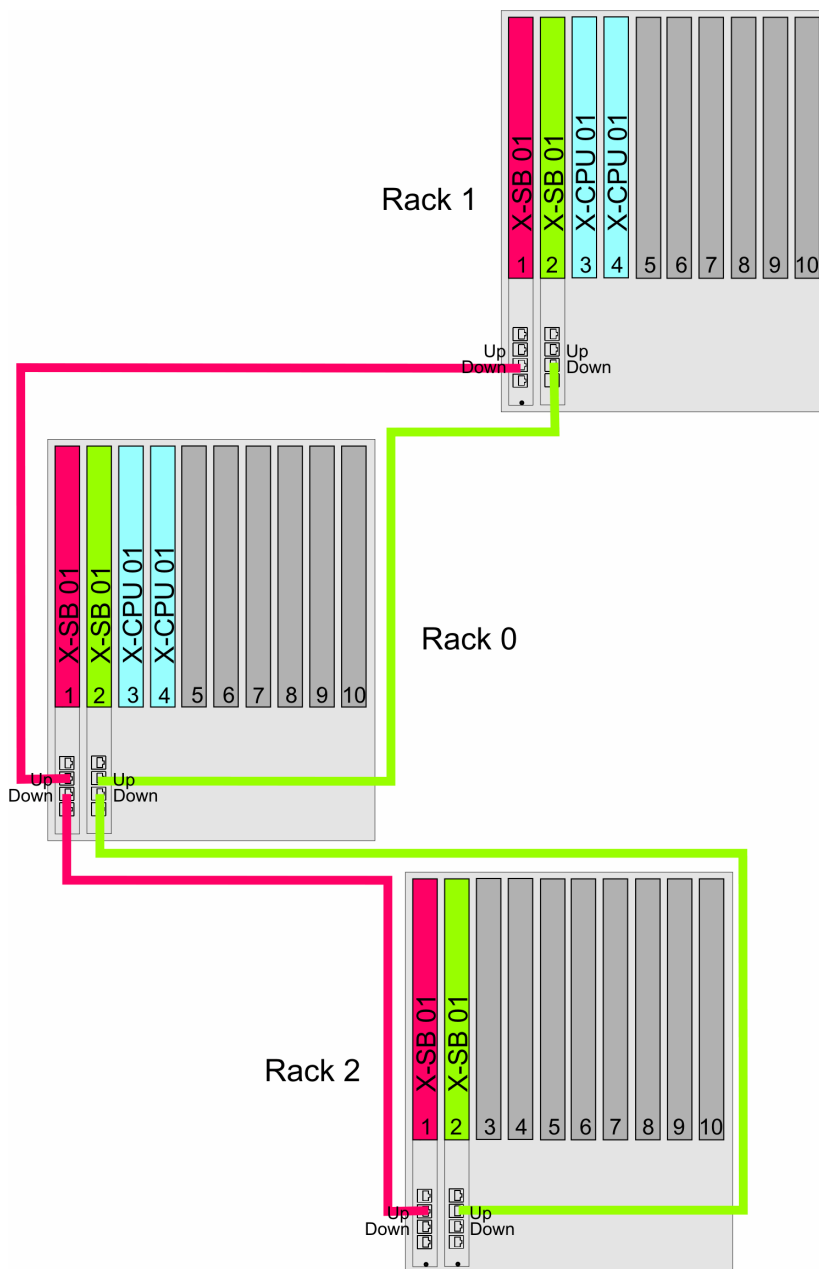
**Qualitativ minderwertige Patchkabel können in rauen Umgebungsbedingungen (z. B. Temperaturwechsel, EMV-Belastungen) zu Ausfällen bei der Kommunikation führen!**

Es gibt zwei Möglichkeiten, den Systembus aufzubauen:

- Linienstruktur  
Die Linienstruktur ist die Standard-Struktur.
- Netzstruktur  
Wird Netzstruktur verwendet, können Racks im laufenden Betrieb abgeschaltet und ausgetauscht werden, ohne die Verbindung zu weiteren Racks zu unterbrechen.

### 3.2.1 Systembus mit Linienstruktur

An einem Rack sind zwei Nachbar-Racks anschließbar.



Rack 0: Basisträger 0

Rack 2: Basisträger 2

Rack 1: Basisträger 1

Bild 3: Reihenfolge der Basisträger am Systembus

Durch das Zusammenschalten entsteht eine Reihenfolge der Racks, siehe Bild 3.

- Beginn beim Rack mit der Rack-ID 0.
- Das Erweiterungs-Rack an der *UP*-Buchse des Racks 0 hat die Rack-ID 1.
  - Alle weiteren Racks, die über das Rack 1 an das Rack 0 angeschlossen sind, haben ungerade Rack-IDs bis Rack-ID 15.
- Das Erweiterungs-Rack an der *DOWN*-Buchse des Racks 0 hat die Rack-ID 2.
  - Alle weiteren Racks, die über Rack 2 an das Rack 0 angeschlossen sind, haben gerade Rack-IDs bis Rack-ID 14.



### 3.2.2 Systembus mit Netzstruktur

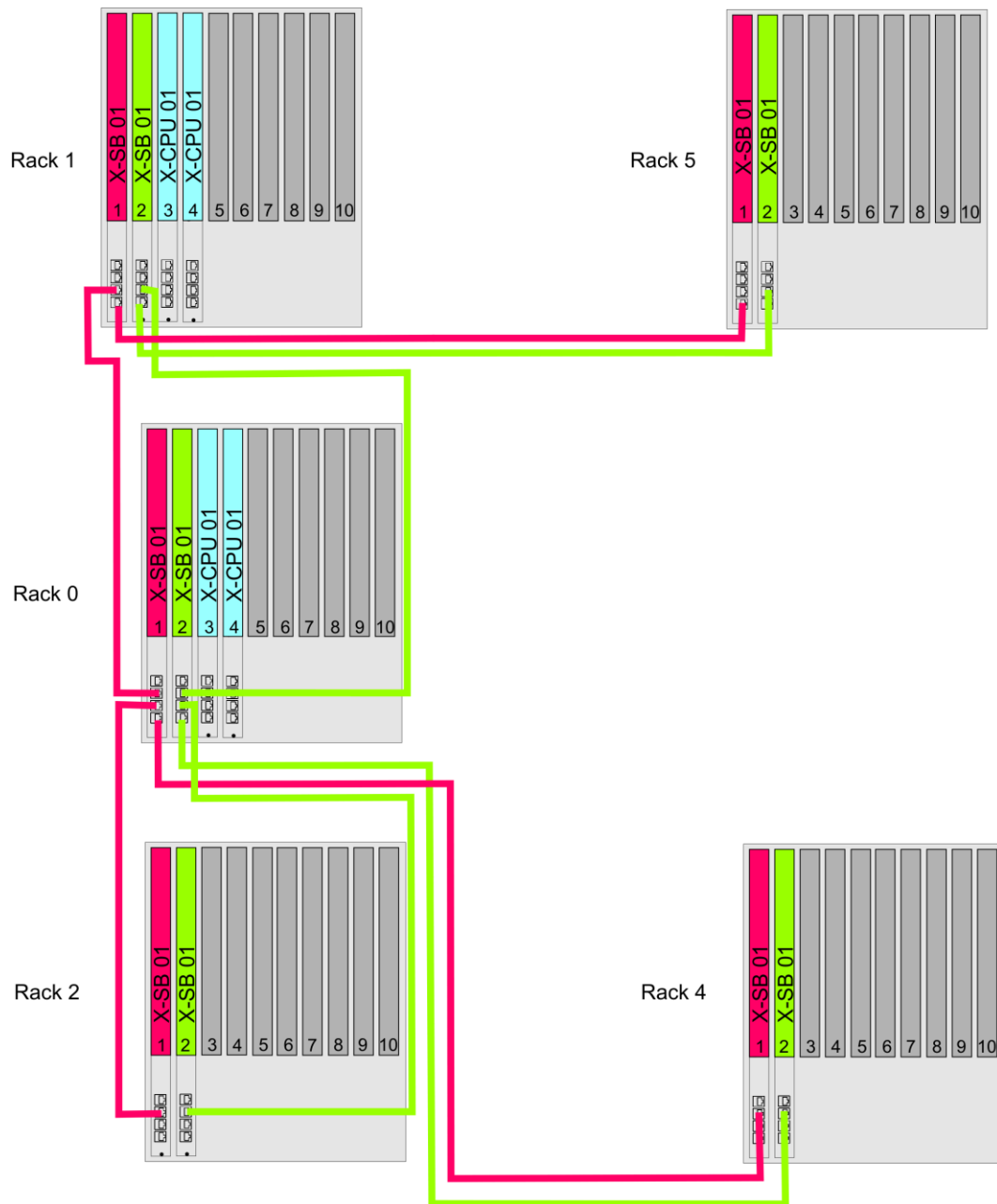
Beim Betrieb des Systembusses in Netzstruktur sind die Anschlüsse UP, DOWN und DIAG des Systembusmoduls gleichwertig, so dass beliebige Strukturen aufbaubar sind. Dazu sind weitere Netzwerkkomponenten wie Switches einsetzbar. Die Netzwerkkomponenten müssen folgende Eigenschaften haben:

- Unterstützung von 1 Gbit/s und Ethernet Flow Control.
- Ausreichend Speicherplatz, so dass alle Nachrichten weitergesendet werden und kein Überlauf auftritt. Das Verwerfen von Nachrichten ist an Systembusdiagnose-Fehlern in der Diagnoseanzeige des Prozessormoduls erkennbar.

Anders als bei der Linienstruktur ist es bei der Netzstruktur möglich, die Rack-ID weitgehend frei zu vergeben. Es ist aber erforderlich, dass die (redundanten) Prozessormodule in die Racks 0 und 1 eingefügt sind. Die Racks 0 und 1 müssen direkt miteinander verbunden sein, d. h. nur über Patchkabel, oder über Patchkabel und Medienkonverter, wenn in beiden Racks Prozessormodule oder responsible Systembusmodule enthalten sind. Die Verbindung darf eine zusätzliche Latenz von bis zu 10 µs haben.

HIMA empfiehlt, die Racks 0 und 1 auch dann direkt zu verbinden, wenn nur Rack 0 Prozessormodule enthält. Dadurch ist eine spätere Erweiterung mit Prozessormodulen in Rack 1 möglich.

Der Aufbau von Ethernet-Ringen ist für den Systembus nicht zulässig. Der Netzwerkpfad von einem beliebigen Modul zu einem Prozessormodul muss immer eindeutig sein, d. h. Datenpakete dürfen nur auf einem Weg zu einer Steuerung gelangen.



Rack 0: Basisträger 0

Rack 1: Basisträger 1

Rack 2: Basisträger 2

Rack 4: Basisträger 4

Rack 5: Basisträger 5

Bild 4: Systembus in Netzstruktur

**HINWEIS**

**Fehlerhafte Funktion des HIMax Systems möglich!**

Die Rack-IDs aller Racks, die mittelbar oder unmittelbar mit dem Systembus verbunden sind, müssen eindeutig sein! Bei Netzstruktur ist das HIMax System nicht in allen Fällen in der Lage, mehrdeutige Rack-IDs zu erkennen.

Es dürfen nur Racks eines HIMax Systems miteinander verbunden werden. Niemals dürfen Racks von unterschiedlichen HIMax Systemen miteinander über einen Systembus verbunden werden.

**Nichtbeachtung kann zu Sicherheitsproblemen führen.**

- Vor Aufnahme des sicherheitsbezogenen Betriebs durch Planung und Überprüfung sicherstellen, dass die Rack-IDs eindeutig sind.
- Die Verantwortung liegt beim Betreiber.

---

Das Systembusmodul kann bei Netzstruktur das Auftreten von Ethernet-Ringen nicht verhindern.

---

**i**

Ein fehlerhafter Aufbau einer Netzstruktur kann dazu führen, dass ein Teil oder das gesamte HIMax System abgeschaltet wird.

---

### 3.2.3 Erweiterung des Systembusses, Systembus-Latenzzeit

Der Systembus basiert auf Ethernet-Technologie. Deshalb ist es möglich, den Systembus mit Ethernet-Komponenten zu erweitern. Das HIMax System kann sich so z. B. über eine weitläufige Produktionsanlage oder die Länge einer Pipeline erstrecken. Dabei müssen alle verwendeten Komponenten eine Datenübertragungsrate von 1 Gbit/s zulassen.

Für größere Entfernungen eignen sich Lichtwellenleiter (LWL) zur Verlängerung des Systembusses.

Größere Entfernungen und ein umfangreicher Ausbau des Systems haben eine Verzögerung der Nachrichten auf dem Systembus zur Folge. Alle Verzögerungen, die durch den räumlichen Aufbau eines Systems entstehen, werden in der Systembus-Latenzzeit zusammengefasst.

Die Systembus-Latenzzeit ist die Verzögerung, die eine Nachricht auf dem Weg zwischen einem Basisrack mit Prozessormodulen und einem E/A-Rack erfährt.

Die maximale Systembus-Latenzzeit ist die größte zulässige Verzögerung, die eine Nachricht von einem E/A-Rack in Richtung Basisrack mit Prozessormodulen erfährt. Die maximale Systembus-Latenzzeit ist abhängig von der Anzahl der verzögernden Netzwerkkomponenten zwischen den Racks. Für die Ermittlung der maximalen Systembus-Latenzzeit muss die Verbindung mit den meisten verzögernden Netzwerkkomponenten herangezogen werden.

Folgende verzögernde Netzwerkkomponenten gibt es:

- Rack mit den Switches der Systembusmodule.
- Anwenderseitige Switches und Medienkonverter für Lichtwellenleiter.
- Lange Patchkabel.
- Lange Lichtwellenleiter.

---

#### i

Von HIMA zur Erweiterung des Systembusses freigegebene Switches und Medienkonverter sind beim Support zu erfragen!

---

Die maximale Systembus-Latenzzeit lässt sich mit dem Systemparameter *Maximale Systembus-Latenzzeit [µs]* in den Eigenschaften der Ressource im Bereich 100 ... 50 000 µs einstellen. Bei der Standard-Einstellung auf *System-Standardwerte* bestimmt das System die maximale Systembus-Latenzzeit. Für die Einstellung auf einen Wert  $\neq$  *System-Standardwerte* ist eine Lizenz erforderlich.

Die maximale Systembus-Latenzzeit ist mit Lizenz auch online einstellbar.

Das HIMax System misst die tatsächliche Systembus-Latenzzeit während des Betriebs und zeigt sie im Control Panel von SILworX an.

### 3.2.3.1 Standardwerte der maximalen Systembus-Latenzzeit

Das HIMax System benutzt in folgenden Fällen die Standardwerte der maximalen Systembus-Latenzzeit:

- Der Parameter *Maximale Systembus-Latenzzeit* [ $\mu\text{s}$ ] ist auf *System-Standardwerte* gesetzt.
- Das Projekt wurde mit einer SILworX Version  $\leq V4$  erstellt.

Für ein HIMax System, das ausschließlich aus HIMax Komponenten und maximal 100 m langen Patchkabel je Verbindung zweier Racks aufgebaut ist, gelten die Standardwerte der maximalen Systembus-Latenzzeit entsprechend der Spalten *Min* der Tabelle 1.

Bei der Benutzung zusätzlicher Netzwerk-Infrastruktur wie Lichtwellenleiter, Medienkonverter, Switches darf die dadurch verursachte **zusätzliche** Verzögerungszeit maximal 50  $\mu\text{s}$  betragen. Die Standardwerte der maximalen Systembus-Latenzzeit einschließlich der zusätzlichen Verzögerung zeigt die Tabelle 1 in den Spalten *Max*.

Maximaler Rack-Abstand	Maximale Systembus-Latenzzeit in $\mu\text{s}$				Beispiele: System ist aufgebaut aus den genannten Racks
	X-CPU 01		X-CPU 31		
	Min	Max <sup>1)</sup>	Min	Max <sup>1)</sup>	
0	49,1	-	665,2	-	Nur Rack 0
1	105,5	155,5	721,6	771,6	Racks 0 und 1
2	161,9	211,9	778,0	828,0	Racks 0, 1, 3
3	218,4	268,4	834,4	884,4	Racks 0, 1, 3, 5
4	274,8	324,8	890,8	940,8	Racks 0, 1, 3,5, 7
5	331,2	381,2	947,2	997,2	Racks 0, 1, 3, 5, 7, 9
6	387,6	437,6	1003,6	1053,6	Racks 0, 1, 3, 5, 7, 9, 11
7	444,0	494,0	1060,9	1110,9	Racks 0, 1, 3, 5 ,7, 9, 11, 13,
8	500,4	550,4	1116,4	1166,4	Racks 1, 0, 2, 4, 6, 8, 10, 12, 14
1) Maximale Systembus-Latenzzeit einschließlich maximaler zusätzlicher Verzögerung durch die Netzwerk-Infrastruktur					

Tabelle 1: Standardwerte der maximalen Systembus-Latenzzeit

HIMax verwendet diese Standardwerte der maximalen Systembus-Latenzzeit unabhängig davon, ob die Linien- oder die Netzstruktur eingestellt ist.

Bei der Berechnung des maximalen Rack-Abstands für **mehr als ein** Rack geht das System vom ungünstigsten Fall aus. Das bedeutet, dass das Rack 1, das Prozessormodule X-CPU 01 enthalten könnte, immer mit berücksichtigt ist, auch dann, wenn es weder parametrisiert wurde noch existiert. Aus diesem Grund nimmt HIMax für ein System, das aus den Racks 0, 2 und 4, besteht, als maximalen Rack-Abstand 3 an und nicht 2!

Ein System mit Prozessormodulen vom Typ X-CPU 31, das aus den Racks 0, 2 und 4 besteht, hat dagegen einen maximalen Rack-Abstand von 2, da die Prozessormodule vom Typ X-CPU 31 nur im Basisrack 1 eingesetzt werden können.

### 3.2.3.2 Systembus-Ausdehnung bei Standardeinstellung der maximalen Latenzzeit

Bereits bei der Standard-Einstellung der maximalen Systembus-Latenzzeit, und damit ohne zusätzliche Lizenz, ist es möglich, den Systembus mittels Lichtwellenleiter über eine große Entfernung auszudehnen. Dabei wird die Leitungslänge durch die zusätzliche Verzögerung des Signals im Lichtwellenleiter und in den Medienkonvertern zwischen Patchkabel und Lichtwellenleiter begrenzt.

HIMax erlaubt bei Standard-Latenzzeit folgende maximale **zusätzliche** Verzögerungszeiten zwischen Modulen:

- Maximal 10 µs zwischen zwei redundanten Prozessormodulen.
- Maximal 50 µs zwischen einem Prozessormodul und dem am weitesten entfernten E/A-Modul.

Die Verwendung eines Lichtwellenleiters bedingt folgende Verzögerungen:

- Innerhalb des Lichtwellenleiters abhängig von Typ z. B. 5 µs/km.
- 1 µs durch die Medienkonverter, Lichtwellenleiter und zwei Patchkabel.

Die Verzögerung durch die kurzen Patchkabel zwischen Systembusmodulen und Konvertern entspricht der des Lichtwellenleiters. Die Länge der Patchkabel geht in die Gesamtlänge mit ein.

Wenn alle Prozessormodule in unmittelbarer Nähe zueinander angeordnet sind, entweder im Rack 0 oder verteilt auf die Racks 0 und 1, dann dürfen sich die beiden am weitesten entfernten Racks mit E/A-Modulen bis zu je 9,8 km von den Prozessormodulen befinden.

Das HIMax System kann eine Ausdehnung von maximal 19,6 km haben (Bild 5).

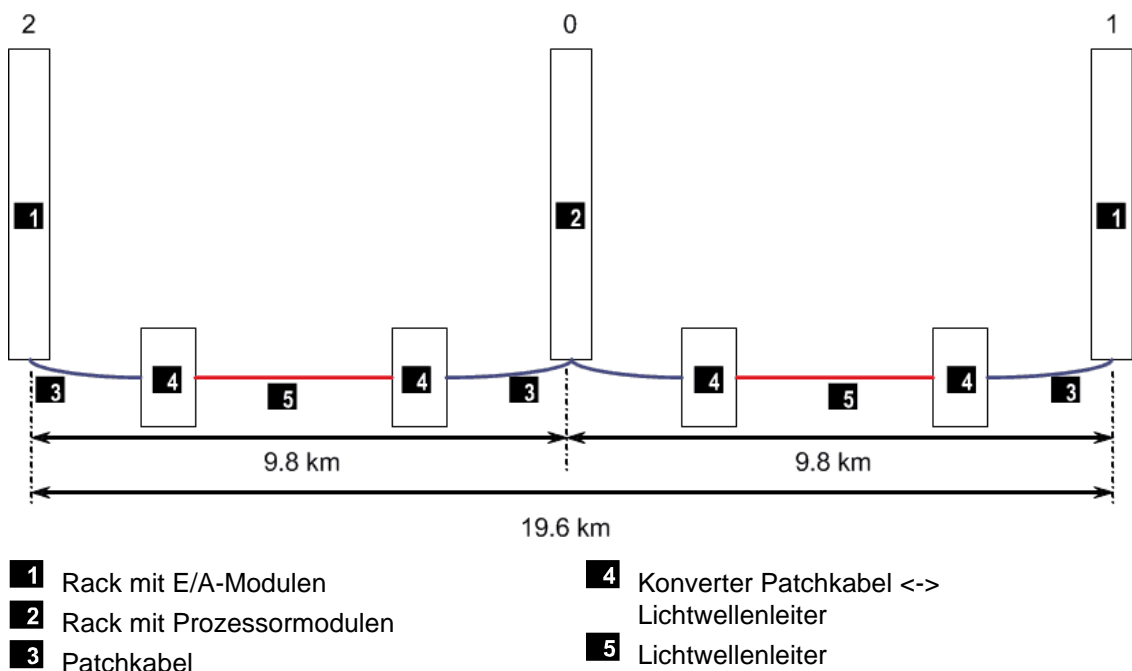


Bild 5: Maximale Ausdehnung bei Standardwert der Latenzzeit

Die Verzögerungszeit zwischen den Prozessormodulen im Rack 0 und z. B. dem linken Rack (Rack-ID 2) mit E/A-Modulen setzt sich zusammen aus der Verzögerungszeit durch die Konverter (1 µs) und der Verzögerungszeit durch die Länge des Lichtwellenleiters (maximal 50 µs - 1 µs). Für die Verzögerungszeit des Lichtwellenleiters und dessen Länge gilt:

$$49 \mu s \geq \text{Länge} * 5 \mu s/km, \text{ d. h., Länge} \leq 9800 \text{ m}$$

Für die Länge zwischen Prozessormodulen im Rack 0 und dem rechten Rack (Rack-ID 1) mit E/A-Modulen gilt dasselbe, die maximale Länge des Lichtwellenleiters ist ebenfalls 9800 m.

### 3.2.3.3 Maximaler Abstand zwischen Prozessormodulen

Sind die Prozessormodule auf die Racks 0 und 1 verteilt, ist es möglich, diese Racks entfernt voneinander aufzustellen, und sie durch Lichtwellenleiter zu verbinden (Bild 6).

Die beiden Racks mit Prozessormodulen dürfen sich in einem Abstand von bis zu 1,8 km voneinander befinden.

Das HIMax System kann in diesem Fall eine Ausdehnung von bis zu 17,4 km haben.

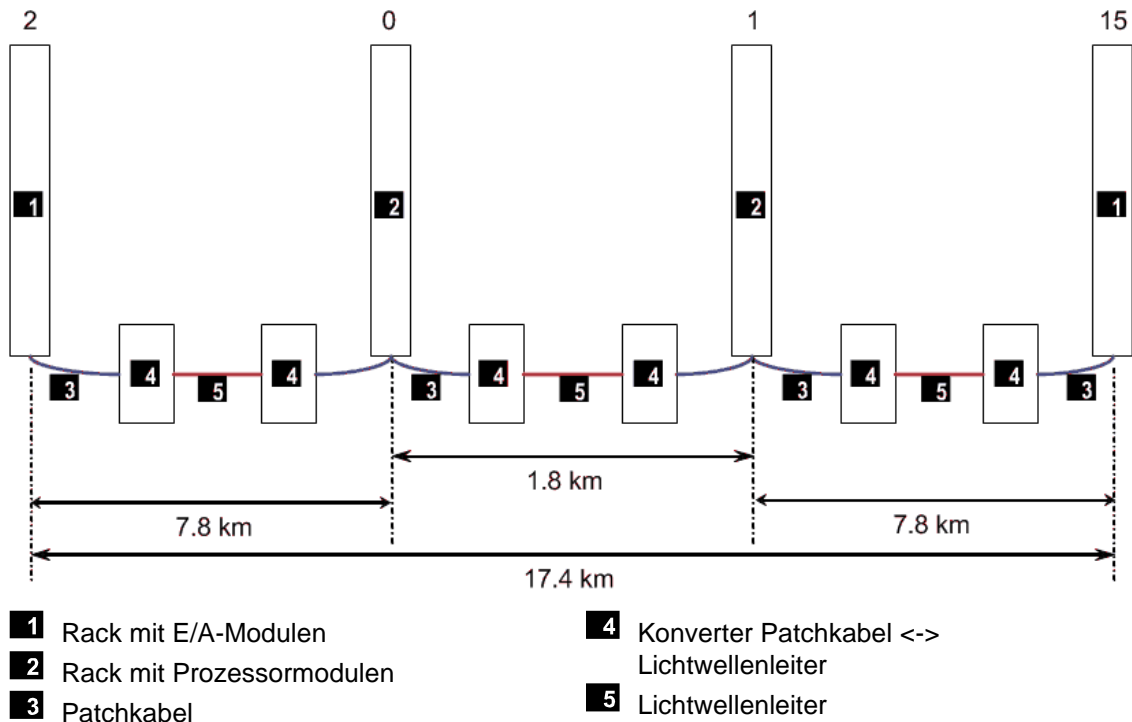


Bild 6: Maximale Entfernung zwischen Prozessormodulen bei Standardwert der Latenzzeit

Die Verzögerungszeit zwischen den Racks 0 und 1 setzt sich zusammen aus der Verzögerungszeit der beiden Konverter (1  $\mu$ s) und der Verzögerungszeit des Lichtwellenleiters (maximal 10  $\mu$ s - 1  $\mu$ s). Für die Verzögerungszeit des Lichtwellenleiters und dessen Länge gilt:

$$9 \mu\text{s} \geq \text{Länge} * 5 \mu\text{s/km, d. h., Länge} \leq 1800 \text{ m}$$

Die Verzögerungszeit zwischen dem linken Rack (Rack-ID 2) mit E/A-Modulen und dem Rack (Rack-ID 1) mit Prozessormodulen setzt sich zusammen aus:

- Der Verzögerungszeit der Strecke zwischen den beiden Racks mit Prozessormodulen (Rack-ID 0 und Rack-ID 1) (siehe Bild 6).
- Der Verzögerungszeit der Strecke zwischen dem linken Rack (Rack-ID 2) mit E/A-Modulen und dem Rack (Rack-ID 0) mit den Prozessormodulen. Diese darf maximal 50  $\mu$ s - 10  $\mu$ s = 40  $\mu$ s sein.  
Sie setzt sich zusammen aus der Verzögerungszeit der Konverter (1  $\mu$ s) und der Verzögerungszeit des Lichtwellenleiters (maximal 39  $\mu$ s). Für die Verzögerungszeit des Lichtwellenleiters und dessen Länge gilt:

$$39 \mu\text{s} \geq \text{Länge} * 5 \mu\text{s/km, d. h., Länge} \leq 7800 \text{ m}$$

Für die Länge des Lichtwellenleiters zwischen den Racks 1 und 15 gilt dasselbe, die maximale Länge des Lichtwellenleiters ist ebenfalls 7800 m.

### 3.2.3.4 Berechnung einer anwenderspezifischen maximalen Systembus-Latenzzeit

Bei der Berechnung der maximalen Systembus-Latenzzeit sind folgende Einflüsse zu berücksichtigen:

- Die Latenz von zusätzlichen Netzwerkkomponenten z. B. Switches.
- Die Staueffekte an einem Linienpunkt (Knoten) oder Linien-Endpunkt eines Racks.

Zur Bestimmung der maximalen Systembus-Latenzzeit sind alle Verbindungen zwischen den Racks mit Prozessormodulen und den E/A-Racks zu betrachten.

Die Summe, die sich aus den Latenzzeiten aller Netzwerkkomponenten zwischen Racks mit einem Prozessormodul und einem betrachteten Rack ergibt, ist die maximale Latenzzeit.

Ein Beispiel für die Verbindung zweier Racks, Rack A und Rack B, durch eine Lichtwellenleiter-Strecke zeigt Bild 7.

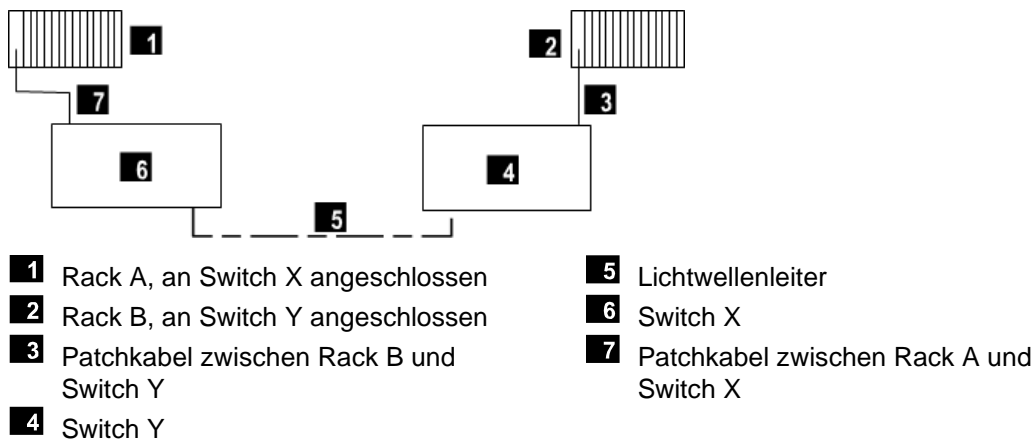


Bild 7: Verbindung zweier Racks durch einen Lichtwellenleiter

Für eine Lichtwellenleiter-Verbindung mit zugelassenen Switches ist die Latenzzeit für die Strecke zwischen dem Anschluss am Systembusmodul in Rack A und dem Anschluss am Systembusmodul in Rack B gemäß folgender Formel zu berechnen:

$$t_{\text{Latenz}} = t_{\text{Patch1}} + t_{\text{Nachricht}} + t_{\text{Switch X}} + t_{\text{LWL}} + t_{\text{Nachricht}} + t_{\text{Switch Y}} + t_{\text{Patch2}} + t_{\text{Nachricht}}$$

$t_{\text{Latenz}}$	Latenzzeit der Verbindung	
$t_{\text{Patch1}}$	Latenzzeit des Patchkabels zwischen Rack A und Switch X	s. u.
$t_{\text{Switch X}}$	Latenzzeit von Switch X	5 $\mu\text{s}$
$t_{\text{LWL}}$	Latenzzeit des Lichtwellenleiters	s. u.
$t_{\text{Switch Y}}$	Latenzzeit von Switch Y	5 $\mu\text{s}$
$t_{\text{Patch2}}$	Latenzzeit des Patchkabels zwischen Rack B und Switch Y	s. u.
$t_{\text{Nachricht}}$	Dauer einer Nachricht bei 1 Gbit/s. Muss für jede Teilstrecke einmal berücksichtigt werden.	6,592 $\mu\text{s}$



Die Latenzzeiten der Patchkabel **3** und **7**, und des Lichtwellenleiters **5** sind jeweils wie folgt zu berechnen:

$$t = \text{Dämpfung} * l/c$$

t	Latenzzeit des Patchkabels oder des Lichtwellenleiters	t <sub>Patch1</sub> oder t <sub>Patch2</sub> oder t <sub>LWL</sub>
l	Länge des Patchkabels oder des Lichtwellenleiters	l <sub>Patch1</sub> oder l <sub>Patch2</sub> oder l <sub>LWL</sub>
c	Lichtgeschwindigkeit	ca. 300 000 km/s
Dämpfung	Dämpfung des Patchkabels oder des Lichtwellenleiters	2 (für beide angenommener Wert)

Beim Aufbau des Systembusses sind die folgenden Hinweise zu beachten:

- Die maximale Latenzzeit zwischen Prozessormodulen, Kommunikationsmodulen und Systembusmodulen, an denen das PADT angeschlossen werden darf, berechnet sich ausschließlich gemäß der Tabelle 1 nach dem Abstand zu Basisracks mit Prozessormodulen. Die berechnete maximale Latenzzeit ist unabhängig vom Wert des Systemparameters *Maximale Systembus-Latenzzeit [µs]* in SiLworX.
- Die maximale Latenzzeit zwischen den beiden Racks mit Prozessormodulen oder mit responsible Systembusmodulen darf gegenüber einer Standardverkabelung, d. h. direkter Verbindung über maximal 100 m langen Patchkabel, maximal um 10 µs zusätzliche Latenz erhöht sein.

Die Netzwerkparameter, wie Latenzzeit von Switches oder der Dämpfung, sind den technischen Daten zu entnehmen oder durch Messung zu ermitteln und in die Berechnung zu übernehmen.

---

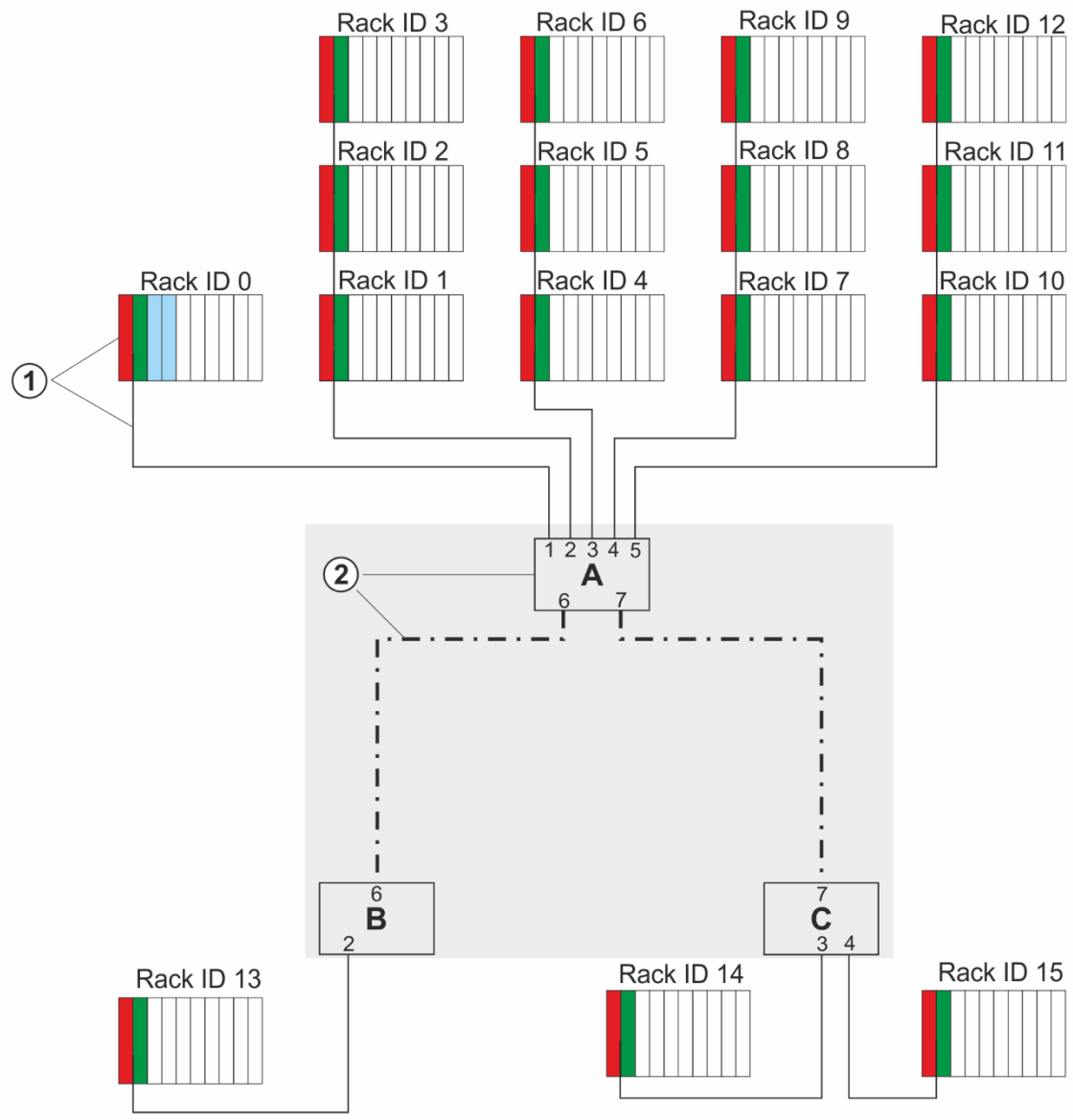
**i**

HIMA empfiehlt, mit der Auslegung der Netzwerkstrukturen und der Berechnung der maximalen Latenzzeit einen Netzwerkfachmann zu beauftragen.

---

### 3.2.3.5 Berechnungsbeispiel für eine anwenderspezifische maximale Latenzzeit

Für die Berechnung einer anwenderspezifischen maximalen Latenzzeit wird die maximale Anzahl aller Racks betrachtet, die über Patchkabel oder über Switches miteinander verbunden sind, siehe Beispiel in Bild 8. Während für die Verbindungen mit Patchkabel je 100 m angenommen werden, werden für die Verbindungen mit Lichtwellenleiter je 10 km angenommen. Für die Berechnung wird ein System mit Prozessormodulen X-CPU 01 und mit Switches der Fa. Hirschmann SPIDER II Giga 5T/2S EEC Rail Switch betrachtet.



- 1** Racks untereinander und mit Switches mit je 100 m Patchkabel verbunden.
- 2** Switches A, B und C untereinander mit je 10 km Lichtwellenleiter verbunden.

Bild 8: Beispiel zur Berechnung der Systembus-Latenzzeit

Bei der Berechnung der maximalen Systembus-Latenzzeit für dieses Beispiel wird von folgenden Werten ausgegangen:

Term	Wert	Beschreibung
$t_{\text{Switch}}$ :	5 $\mu\text{s}$	Interne Latenz eines Switches
$c$ :	300 000 km/s	Lichtgeschwindigkeit
$D_{\text{LWL}}$ :	1 angenommen	Dämpfung des Lichtwellenleiters
$D_{\text{Patch}}$ :	2 angenommen	Dämpfung des Patchkabels
$l_{\text{Patch}}$ :	100 m	Länge des Patchkabels, hier für alle gleich
$l_{\text{LWL}}$ :	10 km	Länge der Lichtwellenleiter, hier für alle gleich
$t_{\text{Nachricht}}$ :	6,592 $\mu\text{s}$	Dauer einer Nachricht bei 1 Gbit/s, für jede Teilstrecke einmal zu berücksichtigen
$t_{\text{LWL}}$ :	$= (l_{\text{LWL}} * D_{\text{LWL}} / c) + t_{\text{Nachricht}}$ $= 33,33 \mu\text{s} + 6,592 \mu\text{s}$ $= 40 \mu\text{s} (39,92 \mu\text{s})$	Nachrichten-Laufzeit über 10 km Lichtwellenleiter mit einer Dämpfung = 1. Dämpfung in Anwendungen berücksichtigen.
$t_{\text{Patch}}$ :	$= 8 \mu\text{s} (7,259 \mu\text{s})$	Nachrichten-Laufzeit über 100 m Patchkabel
$t_{\text{JamSwitch}}$ :	$= x * t_{\text{Nachricht}}$ $= x * 6,592 \mu\text{s}$	Staueffekte an einem Switch je Anzahl der konkurrierenden Ports (3, 4, 5, 6 und 7).
$t_{\text{LineEP}}$ :	35,96 $\mu\text{s}$	Staueffekte an einem Linien-Endpunkt
$t_{\text{LineP}}$ :	42,55 $\mu\text{s}$	Staueffekte an einem Linienpunkt (Knoten)

Tabelle 2: Werte für die Berechnung der Systembus-Latenzzeit

Für die Berechnung der maximalen Systembus-Latenzzeit eines HIMax Systems müssen die Verbindungen mit den meisten Netzwerkkomponenten betrachtet werden.

In Bild 8 ist es erstens die Verbindung zwischen Rack 3 (Rack ID 3) und Rack 0 (Rack ID 0). Die Latenzzeit wird wie folgt berechnet:

$$\begin{aligned}
 t_{\text{Latenz}} &= 4 * t_{\text{Patch}} + t_{\text{Switch}} + t_{\text{JamSwitch}} + 2 * t_{\text{LineP}} + 2 * t_{\text{LineEP}} \\
 t_{\text{Latenz}} &= 4 * 8 \mu\text{s} + 5 \mu\text{s} + 5 * 6,592 \mu\text{s} + 2 * 42,55 \mu\text{s} + 2 * 35,96 \mu\text{s} \\
 t_{\text{Latenz}} &= 32 \mu\text{s} + 5 \mu\text{s} + 32,96 \mu\text{s} + 85,1 \mu\text{s} + 71,92 \mu\text{s} \\
 t_{\text{Latenz}} &= 226,98 \mu\text{s}
 \end{aligned}$$

$4 * t_{\text{Patch}}$ :	Insgesamt 4 Patchkabel zwischen den Racks 0 und Rack 3 über Switch.
$t_{\text{Switch}}$ :	Interne Latenz des Switchs A.
$t_{\text{JamSwitch}}$ :	$= x * t_{\text{Nachricht}}$ , maximalen Latenzzeit durch 5 konkurrierenden Ports (3, 4, 5, 6 und 7).
$2 * t_{\text{LineP}}$ :	Staueffekte an den Linienpunkten (Knoten) Rack 1 und 2.
$2 * t_{\text{LineEP}}$ :	Staueffekte an den Linien-Endpunkten Rack 0 und 3.

Die Berechnung gilt übereinstimmend für die Verbindungen zwischen Rack 0 und den Racks 6, 9 oder 12.

Zweitens die Verbindung zwischen Rack 15 (Rack ID 15) und Rack 0 (Rack ID 0). Die Latenzzeit wird wie folgt berechnet:

$$\begin{aligned}
 t_{\text{Latenz}} &= 2 * t_{\text{Patch}} + 2 * t_{\text{Switch}} + t_{\text{LWL}} + t_{\text{JamSwitch}} + 2 * t_{\text{LineEP}} \\
 t_{\text{Latenz}} &= 2 * 8 \mu\text{s} + 2 * 5 \mu\text{s} + 40 \mu\text{s} + 6 * 6,592 \mu\text{s} + 2 * 35,96 \mu\text{s} \\
 t_{\text{Latenz}} &= 16 \mu\text{s} + 10 \mu\text{s} + 40 \mu\text{s} + 39,55 \mu\text{s} + 71,92 \mu\text{s} \\
 t_{\text{Latenz}} &= 177,47 \mu\text{s}
 \end{aligned}$$

**Rack 15 -> Rack 0:**

- $2 * t_{\text{Patch}}$ : Insgesamt 2 Patchkabel zwischen den Racks 0 und Rack 15 über Switch A und C.
- $2 * t_{\text{Switch}}$ : Interne Latenz Switch A und C.
- $t_{\text{LWL}}$ : Nachrichten-Laufzeit über 10 km Lichtwellenleiter
- $t_{\text{JamSwitch}} (A, C)$ :  $= x * t_{\text{Nachricht}}$ , maximalen Latenzzeit durch 6 konkurrierenden Ports (A: 2, 3, 4, 5 und 6) (C: 3).
- $2 * t_{\text{LineEP}}$ : Staueffekte an den Linien-Endpunkten Rack 0 und 15.

Für das HIMax System in Bild 8 muss der größere Wert (226,98  $\mu\text{s}$ ) der beiden Berechnungen für den Parameter *Maximale Systembus-Latenzzeit* [ $\mu\text{s}$ ] eingesetzt werden. Die Eingabe eines noch größeren Wertes ist jederzeit möglich.

Wenn im HIMax System in Bild 8 zusätzlich ein Kommunikationsmodul in Rack 8 eingesetzt werden soll, dann muss die maximal zulässige Latenzzeit zwischen Basisrack und dem Rack 8 gemäß der Tabelle 1 mit der nachfolgenden Berechnung verglichen werden.

Ist die Latenzzeit der Berechnung größer als die in der Tabelle 1 angegebene maximal zulässige Latenzzeit, so ist der Einsatz eines Kommunikationsmodul im Rack 8 nicht zu empfehlen.

$$\begin{aligned}
 t_{\text{Latenz}} &= 3 * t_{\text{Patch}} + t_{\text{Switch}} + t_{\text{JamSwitch}} + 1 * t_{\text{LineP}} + 2 * t_{\text{LineEP}} \\
 t_{\text{Latenz}} &= 3 * 8 \mu\text{s} + 5 \mu\text{s} + 5 * 6,592 \mu\text{s} + 1 * 42,55 \mu\text{s} + 2 * 35,96 \mu\text{s} \\
 t_{\text{Latenz}} &= 24 \mu\text{s} + 5 \mu\text{s} + 32,96 \mu\text{s} + 24,55 \mu\text{s} + 71,92 \mu\text{s} \\
 t_{\text{Latenz}} &= 158,43 \mu\text{s}
 \end{aligned}$$

**Rack 8 -> Rack 0:**

- $3 * t_{\text{Patch}}$ : Insgesamt 3 Patchkabel zwischen den Racks 0 und Rack 8 über Switch A.
- $t_{\text{Switch}}$ : Interne Latenz des Switchs A.
- $t_{\text{JamSwitch}}$ :  $= x * t_{\text{Nachricht}}$ , maximalen Latenzzeit durch 5 konkurrierenden Ports (2, 3, 5, 6 und 7).
- $1 * t_{\text{LineP}}$ : Staueffekte an den Linienpunkten (Knoten) Rack 7.
- $2 * t_{\text{LineEP}}$ : Staueffekte an den Linien-Endpunkten Rack 0 und 8.

Ergebnis: Die errechnete maximale Latenzzeit 158,43  $\mu\text{s}$  ist kleiner als die in Tabelle 1 angegebene Latenzzeit von 324,8 (X-CPU 01) und 940,8 (X-CPU 31). Aus diesem Grund ist es zulässig in das Rack 8 ein Kommunikationsmodul einzusetzen.

### 3.3 Module und Connector Boards

Das HIMax System ist ein modulares System mit unterschiedlichen Modultypen. Folgende Modultypen werden unterschieden:

- Prozessormodule  
zur Abarbeitung der Anwenderprogramme.
- Das Prozessormodul X-CPU 31 vereinigt die Funktionen eines Prozessormoduls mit der eines Systembusmoduls.
- Systembusmodule  
zur Verwaltung der Systembusse.
- Eingangsmodule  
zur Messung und Vorverarbeitung der Prozesswerte.
- Ausgangsmodule  
zur Umsetzung der Ergebnisse des Anwenderprogramms in Steuerbefehle für Aktoren.
- Spezielle E/A-Module, wie z. B. Überdrehzahl-Schutzmodul (X-MIO 7/6 01)  
HART-Kommunikationsmodul (X-HART 32 01).
- Kommunikationsmodule
  - zur Kommunikation mit externen Geräten oder Systemen, die mit Standard-Datenübertragungsprotokollen arbeiten (z. B. Modbus, PROFIBUS).
  - physikalische Schnittstellen für **saferethernet** zur Verbindung mit weiteren HIMA Steuerungen.

Die elektronischen Komponenten der Module sind mit einem Schutzlack gegen Korrosion und Staub überzogen.

Jedes Modul bildet mit einem Connector Board eine funktionale Einheit. Ein Connector Board stellt die Verbindung zwischen Modul und der Feldebene her oder die Kommunikationsverbindung zu anderen Steuerungen oder Geräten. Beim Austausch eines Moduls bleibt das Connector Board im Rack. Dadurch ist es nicht nötig, die an das Connector Board angeschlossenen Kabel oder Leitungen zu lösen und neu anzuschließen.

Zu jedem Modultyp gehören ein oder mehrere Typen von Connector Boards.

Die Steckverbinder zwischen den E/A-Modulen und ihren Connector Boards sind mechanisch codiert. Dadurch ist ein Modul eines bestimmten Typs nur auf das zugehörige Connector Board steckbar, was eine Bestückung mit unpassenden Modulen verhindert. Die Codierung erfolgt durch Keile auf der Federleiste des Connector Boards, siehe auch die Handbücher der E/A-Module.

Bei den Connector Boards für die E/A-Module gibt es zwei Typen:

- Connector Boards zum direkten Anschluss der Zuleitungen zu den Feldgeräten.
- Connector Boards zum Anschluss von Field Termination Assemblies (FTAs).

Die FTAs dienen dem Anschluss der Feldgeräte und können getrennt von der Steuerung untergebracht werden, z. B. in einem eigenen Rangierverteiler.

Weitere Einzelheiten zu Connector Boards und Field Termination Assemblies finden Sie in den Modulhandbüchern und in den Handbüchern der Field Termination Assemblies.

### 3.3.1 Identifikation der Module über SRS

Das HIMax System identifiziert die Module mit den Angaben **System.Rack.Slot** (SRS):

Bezeichnung	Wertebereich	Beschreibung
System	1 ... 65 535	Identifikation der Ressource
Rack	0 ... 15	Identifikation des Racks
Slot	1 ... 18	Identifikation des Steckplatzes

Tabelle 3: Identifikation eines Moduls durch System.Rack.Slot

i

Jedem im Netzwerk erreichbaren Gerät, z. B. Remote I/O, ist eine eindeutige SRS zuzuweisen.

### 3.3.2 Zulässige Belegung von Steckplätzen

Die Belegung der Steckplätze ist folgendermaßen festgelegt:

1. Die Steckplätze 1 und 2 jedes Racks sind für Systembusmodule reserviert. In Basisträgern vom Typ X-BASE PLATE 1x 31 sind die Steckplätze 1 und 2 für X-CPU 31 Prozessormodule reserviert.  
Keine anderen Module auf diese Steckplätze stecken!
2. Prozessormodule sind nur auf Steckplätzen entsprechend den Regeln im nächsten Abschnitt zugelassen.
3. E/A-Module und Kommunikationsmodule sind auf alle verbleibenden Steckplätze steckbar, nachdem die Steckplätze für Prozessormodule festgelegt sind.

#### 3.3.2.1 Zulässige Steckplätze für Prozessormodule

Für die Belegung von Steckplätzen mit Prozessormodulen, auch im Hardware-Editor, gelten folgende Regeln:

1. Für ein HIMax System sind maximal 4 Prozessormodule vom Typ X-CPU 01 **oder** 2 Prozessormodule vom Typ X-CPU 31 auf X-BASE PLATE 1x 31 möglich.
2. Prozessormodule X-CPU 01 sind nur auf folgenden Steckplätzen zulässig:
  - Steckplätze 3 bis 6 im Rack 0.
  - Steckplätze 3 bis 4 in Rack 1.
3. Steckplatz 5 auf Rack 0 und Steckplatz 4 auf Rack 1 dürfen nicht gleichzeitig Prozessormodule enthalten.
4. Steckplatz 6 auf Rack 0 und Steckplatz 3 auf Rack 1 dürfen nicht gleichzeitig Prozessormodule enthalten.
5. Prozessormodule X-CPU 31 sind nur auf den Steckplätzen 1 und 2 von Rack 0 zulässig.  
In diesem Fall dürfen keine Prozessormodule auf anderen Steckplätzen vorhanden sein, auch nicht in Rack 1!

#### HINWEIS



**Betriebsstörung möglich!**

**Steckplätze für Prozessormodule nur entsprechend diesen Regeln planen!**

Die Tabelle zeigt die Vorzugsvarianten entsprechend den Regeln:

Variante	Basisrack 0 Prozessormodul(e) in Steckplatz:	Rack 1 Prozessormodul(e) in Steckplatz:	Erforderliche Systembusse
1	3 bei Mono-Betrieb <sup>1)</sup>	---	A
2	3	---	A + B
3	3, 4	---	A + B
4	3, 4, 5	---	A + B
5	3, 4, 5, 6	---	A + B
6	3	3	A + B
7	3, 4	3	A + B
8	3, 4	3, 4	A + B
9	3, 4, 5	3	A + B
10	1 bei Mono-Betrieb (X-CPU 31) <sup>2)</sup>	---	A
11	1, 2 (X-CPU 31)	---	A + B
<sup>1)</sup> Mono-Betrieb: Das Projekt ist in SILworX für Mono-Betrieb konfiguriert und hat nur ein Prozessormodul in Steckplatz 3, mindestens ein Systembusmodul in Steckplatz 1, sowie E/A-Module und evtl. Kommunikationsmodule. In SILworX muss der Schalter für Mono-Startup gesetzt sein. Redundante Systembusmodule sind immer möglich und empfohlen! <sup>2)</sup> Mono-Betrieb: Das Projekt ist in SILworX für Mono-Betrieb konfiguriert und hat nur ein Prozessormodul X-CPU 31 in Steckplatz 1, sowie E/A-Module und evtl. Kommunikationsmodule. In SILworX muss der Schalter für Mono-Startup gesetzt sein. Redundante X-CPU 31 sind immer möglich und empfohlen!			

Tabelle 4: Empfohlene Steckplatzpositionen von Prozessormodulen

HIMA empfiehlt, die Variante 3 auch dann einzusetzen, wenn die Variante 1 möglich wäre. Dann ist ein Tausch des Prozessormoduls möglich, ohne den Betrieb zu unterbrechen.

Entsprechend ist bei Verwendung von X-CPU 31 die Variante 11 der Variante 10 vorzuziehen.

Da das Betriebssystem auf maximale Verfügbarkeit ausgelegt ist, ermöglicht es den Betrieb anderer, jedoch nicht empfohlener, Kombinationen. Damit bietet HIMax bei Maßnahmen wie Modultauch oder Umbau mehr Flexibilität. Nach Abschluss der Maßnahmen muss das System jedoch so aufgebaut sein, dass es einer der in der Tabelle 4 empfohlenen Kombinationen entspricht.

Bei Verwendung von Prozessormodulen X-CPU 01 ist für Rack 0 einer der Typen X-BASE PLATE 10 01, X-BASE PLATE 15 01, X-BASE PLATE 15 02, X-BASE PLATE 18 01 einzusetzen.

Bei Verwendung von Prozessormodulen X-CPU 31 ist für Rack 0 einer der Typen X-BASE PLATE 10 31, X-BASE PLATE 15 31, X-BASE PLATE 15 32, X-BASE PLATE 18 31 einzusetzen.

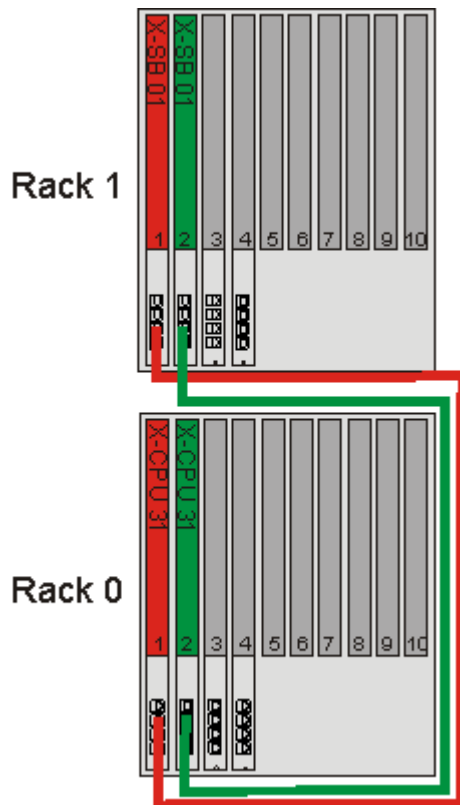


Bild 9: Einsatz von Prozessormodulen X-CPU 31

### 3.3.3 Fehlerbehandlung bei E/A-Modulen

Entdecken die Selbst-Tests Fehler, verhält sich das Modul wie folgt:

- Bei Fehlern einzelner Kanäle schaltet das Modul die einzelnen Kanäle ab und setzt die Systemvariable *Kanal OK* für die fehlerhaften Kanäle auf FALSE.
- Bei Fehlern aller Kanäle schaltet sich das Modul vollständig ab. Das Prozessormodul setzt die Systemvariable *Modul OK* für das fehlerhafte Modul auf FALSE.
- In E/A-Modulen entdeckte Fehler signalisiert das System durch LEDs und Diagnosemeldungen.

HIMA empfiehlt, beim Auftreten von Fehlern die angeschlossene Verkabelung einschließlich Sensoren oder Aktoren zu überprüfen. Kann ein Fehler in der Verkabelung und bei den Sensoren oder Aktoren ausgeschlossen werden, ist das Modul zu tauschen.

Einzelheiten zu den Fehlercodes der einzelnen Module im jeweiligen Modulhandbuch.



### 3.4 Prozessormodul

Im Prozessormodul laufen Anwenderprogramme unter der Kontrolle des CPU-Betriebssystems ab.

#### 3.4.1 Betriebssystem

Aufgaben:

- Steuert den zyklischen Ablauf der Anwenderprogramme.
- Führt Selbst-Tests des Moduls durch.
- Steuert die sicherheitsbezogene Kommunikation über **safeethernet**.
- Verwaltet die Redundanz in Zusammenarbeit mit den anderen Prozessormodulen (Synchronisierung).

##### 3.4.1.1 Genereller Ablauf des Zyklus

Phasen:

1. Lesen der Eingangsdaten.
2. Bearbeiten der Anwenderprogramme.
3. Schreiben der Ausgangsdaten.
4. Sonstige Aktivitäten, z. B. Bearbeitung von Reload.

##### 3.4.1.2 Zustände des Betriebssystems

Für den Anwender erkennbare Zustände:

- LOCKED
- STOPP/GÜLTIGE KONFIGURATION
- STOPP/FEHLERHAFTE KONFIGURATION
- STOPP/BS WIRD GELADEN
- RUN
- RUN/AP STOPP

Die Zustände der Module sind an Hand der Leuchtdioden erkennbar, dabei ist es notwendig, alle LEDs zu betrachten. Weitere Einzelheiten finden Sie in den Modulhandbüchern. In SILworX werden die Zustände in der Online-Ansicht angezeigt.

Die Tabelle 5 gibt einen Überblick über die Zustände des Betriebssystems und nennt die Bedingungen, unter denen sie erreicht werden.

Zustand	Beschreibung	Zustand wird erreicht durch
LOCKED	Prozessormodul nimmt Fabrikeinstellungen an z. B. SRS Netzwerkeinstellungen.	Anlegen der Versorgungsspannung an das Prozessormodul bei Stellung des Mode-Schalters auf <i>Init</i> .
STOPP/GÜLTIGE KONFIGURATION	Prozessormodul gestoppt, gültige Konfiguration im Speicher.	Stoppen des Prozessormoduls mittels SILworX.
		Anlegen der Versorgungsspannung: <ul style="list-style-type: none"> <li>Autostart durch Projektkonfiguration verboten.</li> <li>Mode-Schalter in Stellung <i>Stop</i> und Prozessormodul startet allein.</li> </ul>
		Aus dem Zustand LOCKED: Drehen des Mode-Schalters auf <i>Stop</i> , wenn nur ein einziges Prozessormodul existiert.
		Auftreten eines Fehlers.
STOPP/FEHLERHAFTE KONFIGURATION	Prozessormodul gestoppt, keine gültige Konfiguration im Speicher.	Laden mit Fehler.
		Aus dem Zustand LOCKED: Drehen des Mode-Schalters auf <i>Stop</i> , wenn nur ein einziges Prozessormodul existiert
STOPP/BS WIRD GELADEN	Prozessormodul gestoppt, Betriebssystem wird in den nichtflüchtigen Speicher geladen.	Laden des Betriebssystems mittels SILworX.
RUN	Anwenderprogramm läuft.	Aus dem Zustand STOPP/GÜLTIGE KONFIGURATION: Kommando von SILworX.
		Anlegen der Versorgungsspannung, die folgenden Voraussetzungen müssen erfüllt sein: <ul style="list-style-type: none"> <li>Gültige Projektkonfiguration geladen.</li> <li>Autostart durch Projektkonfiguration erlaubt.</li> <li>Mode-Schalter nicht in Stellung <i>Init</i>.</li> <li>Mode-Schalter in Stellung <i>Run</i>, wenn das Prozessormodul allein startet.</li> </ul>
		Aus dem Zustand LOCKED: Drehen des Mode-Schalters von <i>Init</i> nach <i>Stop</i> oder <i>Run</i> , wenn ein weiteres Prozessormodul im Zustand RUN existiert.
RUN/AP STOPP	Das Anwenderprogramm läuft nicht. Dieser Zustand dient zum Test der Ein- und Ausgänge und der Kommunikation.	Aus dem Zustand STOPP/GÜLTIGE KONFIGURATION durch Kommando von SILworX.

Tabelle 5: Zustände des Betriebssystems, Erreichen der Zustände

Die Tabelle 6 nennt die Eingriffsmöglichkeiten des Anwenders im jeweiligen Zustand.

Zustand	Mögliche Eingriffe durch den Anwender:
LOCKED	<ul style="list-style-type: none"> <li>Fabrikeinstellungen ändern.</li> <li>Durch Drehen des Mode-Schalters in STOPP <sup>1)</sup> gehen.</li> <li>Durch Drehen des Mode-Schalters in RUN gehen.</li> <li>Durch PADT-Kommando stoppen (Zustand STOPP).</li> <li>Durch PADT-Kommando starten (Zustand RUN).</li> </ul>
STOPP/GÜLTIGE KONFIGURATION	<ul style="list-style-type: none"> <li>Anwenderprogramm laden.</li> <li>Anwenderprogramm starten.</li> <li>Betriebssystem laden.</li> <li>Forcen von Variablen vorbereiten.</li> </ul>
STOPP/FEHLERHAFT KONFIGURATION	<ul style="list-style-type: none"> <li>Anwenderprogramm laden.</li> <li>Betriebssystem laden.</li> </ul>
STOPP/BS WIRD GELADEN	Keine. Nach Ende des Ladevorgangs wird das Prozessormodul gestoppt (Zustand STOPP).
RUN	<ul style="list-style-type: none"> <li>Anwenderprogramm stoppen.</li> <li>Variablen forcen.</li> <li>Online-Test durchführen.</li> </ul>
RUN/AP STOPP	Durch PADT-Kommando stoppen (Zustand STOPP).
<sup>1)</sup> STOPP/GÜLTIGE KONFIGURATION oder STOPP/FEHLERHAFT KONFIGURATION, abhängig davon, ob das Prozessormodul eine gültige Konfiguration enthält.	

Tabelle 6: Zustände des Betriebssystems, mögliche Eingriffe durch den Anwender

### i

Die Zykluszeit erhöht sich mit der Anzahl der im System vorhandenen Module. Dies gilt unabhängig davon, ob die Module in der Konfiguration enthalten sind oder nicht.

**Dies kann zum Überschreiten der Watchdog-Zeit führen, wenn im RUN-Betrieb zusätzliche Racks mit mehreren Modulen angeschlossen werden!**

## 3.4.2 Verhalten bei Fehlern

Bei Fehlern geht das Prozessormodul in den Fehlerstopp und versucht einen Neustart. Dabei macht es einen ausführlichen Selbst-Test, der erneut einen Fehlerstopp zur Folge haben kann.

Liegt immer noch ein Fehler an, findet ein Neustart mit eingeschränkter Funktionalität statt, um Dauer-Reboot zu verhindern.

Wenn das Prozessormodul eine Minute normal gelaufen ist, gilt ein folgender Fehlerstopp wieder als erster Fehlerstopp, bei dem ein Neustart versucht wird.

### i

Zur Fehlersuche und Beseitigung der Fehlerursache das PADT benutzen!

## 3.4.3 Prozessormodul X-CPU 31

Die X-CPU 31 erfüllt die Funktionen eines Systembusmoduls (X-SB 01) und eines Prozessormoduls (X-CPU 01). Aus diesem Grund ist es ausschließlich in Rack 0 auf den Steckplätzen 1 und 2 einzusetzen. Der Einsatz auf anderen Racks und Steckplätzen ist nicht möglich.

Eine gemischte Bestückung des Racks 0 mit einem Modul X-CPU 31 und einem X-SB 01 ist nicht möglich. Prozessormodule X-CPU 01 auf den Steckplätzen 3 ... 6 können nicht redundant zu X-CPU 31 auf den Steckplätzen 1 oder 2 arbeiten.

Das Prozessormodul X-CPU 31 hat wegen seiner Doppelfunktion eine geringere Performance für Anwenderprogramme als X-CPU 01. Daher ist es nur für ein System mit bis zu 64 E/A-Modulen geeignet.

### 3.5 Störaustastung

Dieses Kapitel beschreibt die Funktionsweise der Störaustastung von E/A-Modulen im HIMax System.

#### 3.5.1 Wirkung der Störaustastung

Die Störaustastung unterdrückt transiente Störungen, um die Verfügbarkeit des Systems zu erhöhen. Dabei ist sichergestellt, dass das System auf anstehende Störungen innerhalb der parametrisierten Sicherheitszeit sicherheitsbezogen reagiert.

Die Störaustastung ist für jedes E/A-Modul aktivierbar. Die Standardeinstellung ist *aktiv* bei allen Typen von E/A-Modulen, mit Ausnahme der Zählermodule.

Wird eine Störung ausgetastet, verarbeitet das System automatisch die letzten gültigen Ein- und Ausgangswerte anstatt der aktuell gestörten Werte. Die Zeit, in der Störungen ausgetastet werden können, wird begrenzt durch die Sicherheitszeit, Watchdog-Zeit und die Zykluszeit.

Die maximale Störaustastzeit kann mit folgender Formel berechnet werden:

**Maximale Störaustastzeit = Sicherheitszeit - (2 x Watchdog-Zeit)**

Je länger die Störaustastzeit ist, desto länger kann eine Störung ausgetastet werden. Da eine Störung bis zu einem Zyklus anstehen kann, bis sie beim Einlesen erkannt wird, muss zur Berechnung der minimalen Störaustastzeit vom Maximalwert ein Zyklus abgezogen werden.

**Minimale Störaustastzeit = Maximale Störaustastzeit - Zykluszeit**

Die Störaustastung ist wirksam, wenn die Zykluszeit kleiner als die Störaustastzeit ist.

#### 3.5.2 Einstellung der Störaustastung

Um eine möglichst große Anzahl an Zyklen austasten zu können, muss die Sicherheitszeit unter Berücksichtigung der Prozess-Sicherheitszeit möglichst groß sein. Gleichzeitig sollte für die Watchdog-Zeit ein möglichst kleiner Wert eingestellt werden. Dieser muss aber ausreichend groß sein, um Reload und die Synchronisation eines weiteren Prozessormoduls zu zulassen. Details zu den unterschiedlichen Zeiten und deren Anwendung sind im Sicherheitshandbuch (HI 801 002 D) beschrieben.

Einstellung der Störaustastung an folgenden Beispielen:

Beispiel	1 <sup>1)</sup>	2	3 <sup>2)</sup>
Sicherheitszeit [ms]	600	2000	1000
Watchdog-Zeit [ms]	200	500	500
Zykluszeit [ms]	100	200	200
Maximale Störaustastzeit [ms]	200	1000	0
Minimale Störaustastzeit [ms]	100	800	0
<sup>1)</sup> Standardeinstellung in SILworX			
<sup>2)</sup> Bei Beispiel 3 ist keine Störaustastung möglich, da die Störaustastzeit < Zykluszeit ist.			

Tabelle 7: Beispiele zur Berechnung von minimaler und maximaler Störaustastzeit

### 3.5.3 Ablauf der Störaustastung

Der Ablauf der Störaustastung wird verdeutlicht durch die Beispiele:

- Eine transiente Störung wird erfolgreich ausgetastet.
- Eine länger als die maximale Störaustastzeit anstehende Störung führt zur sicheren Reaktion.

#### Beispiel 1: Transiente Störung wird erfolgreich unterdrückt

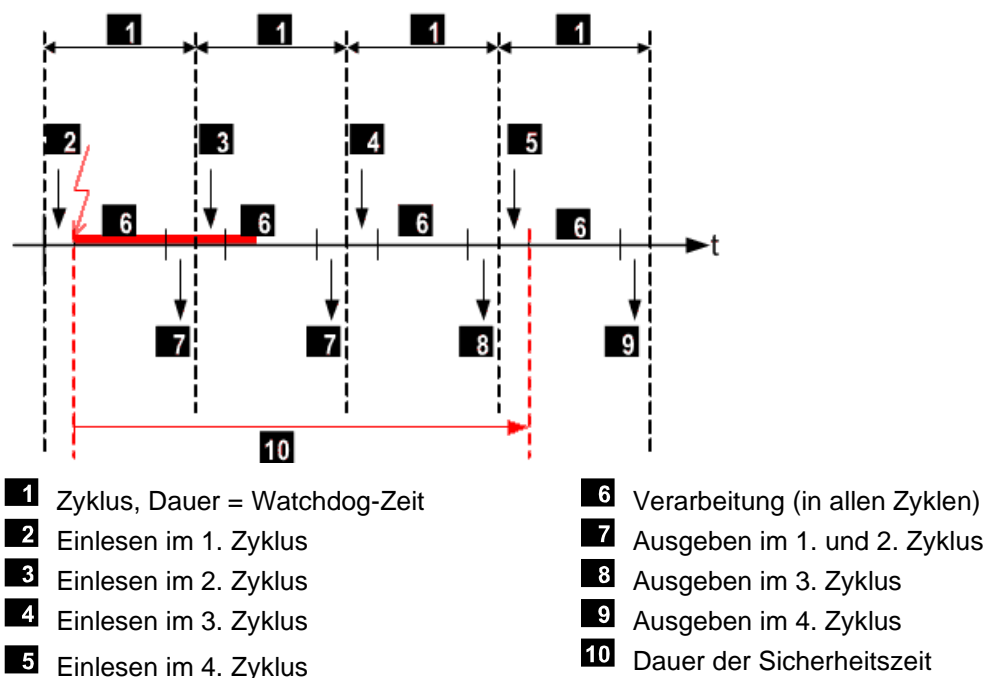


Bild 10: Transiente Störung

Im Beispiel 1 werden in einem Zyklus gültige Eingangswerte bei **2** eingelesen. Das System verarbeitet für diesen Zyklus die gültigen Eingangswerte, obwohl direkt nach Abschluss des Einlesevorgangs eine Störung erfolgt.

Steht die Störung während des Einlesens **3** im folgenden Zyklus noch an, wird die Störung vom Modul erkannt und das System entscheidet nach folgender Regel, ob eine Störaustastung zu diesem Zeitpunkt möglich ist:

**Sicherheitszeit - abgelaufene Zeit - (2 x Watchdog-Zeit) > 0**

Abgelaufene Zeit = Zeit zwischen Einlesen letzter gültiger Werte und Erkennung der Störung.

In diesem Beispiel ist die Störaustastung möglich, da die Störung weniger als ein Zyklus (= abgelaufene Zeit) ansteht und noch 2 weitere Zyklen (2 x Watchdog-Zeit) für eine sichere Reaktion zur Verfügung stehen. Das System verarbeitet für diesen Zyklus die letzten gültigen Eingangswerte vom Zeitpunkt **2**, ohne dass Fehlerreaktionen ausgelöst werden. Die transiente Störung wurde erfolgreich unterdrückt.

Steht die Störung bei **4** nicht mehr an, werden neue gültige Werte eingelesen und verarbeitet.

Bei deaktivierter Störaustastung löst das System beim Einlesen **3** sofort die definierten Fehlerreaktionen aus.

## Beispiel 2: Auslösen einer sicheren Reaktion bei anstehender Störung

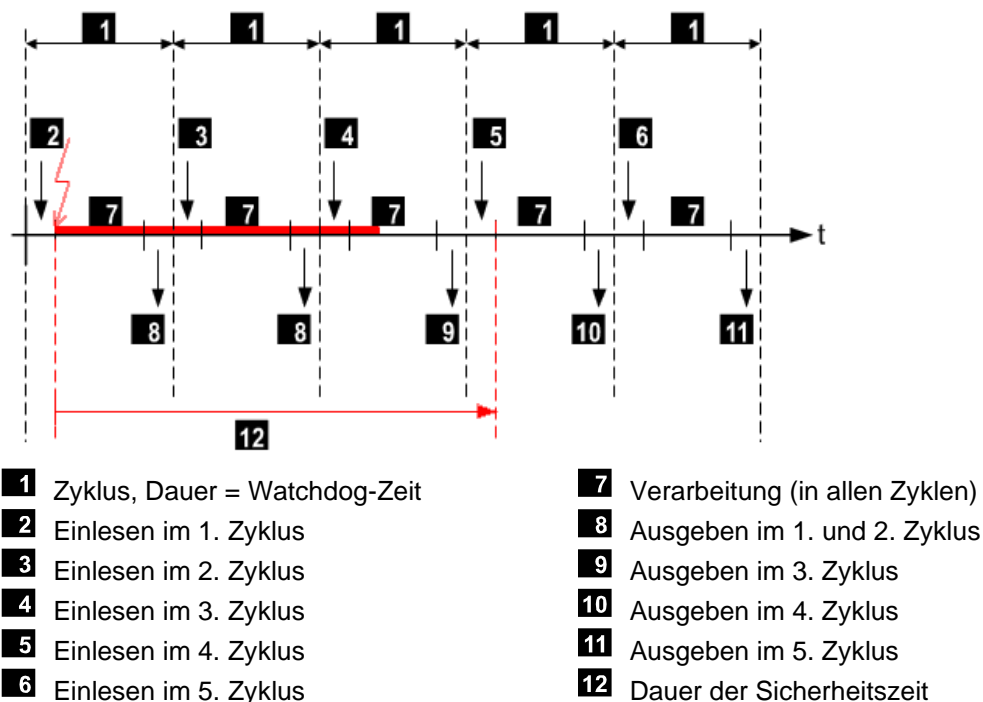


Bild 11: Anstehende Störung führt zu sicherer Reaktion

Im Beispiel 2 werden in einem Zyklus gültige Eingangswerte bei **2** eingelesen. Das System verarbeitet für diesen Zyklus die gültigen Eingangswerte, obwohl direkt nach Abschluss des Einlesevorgangs eine Störung erfolgt.

Steht die Störung während des Einlesens **3** im folgenden Zyklus noch an, wird die Störung vom Modul erkannt und das System entscheidet nach folgender Regel, ob eine Störaustastung zu diesem Zeitpunkt möglich ist:

#### Sicherheitszeit - abgelaufene Zeit - (2 x Watchdog-Zeit) > 0

Die Störaustastung ist im 1. und 2. Zyklus möglich, da die Störung weniger als ein Zyklus (= abgelaufene Zeit) ansteht und noch 2 (ein) weitere Zyklen (2 x Watchdog-Zeit) für eine sichere Reaktion zur Verfügung stehen. Das System verarbeitet für diesen Zyklus die letzten gültigen Eingangswerte von **2**, ohne dass definierte Fehlerreaktionen ausgelöst werden. Die transiente Störung wurde erfolgreich unterdrückt.

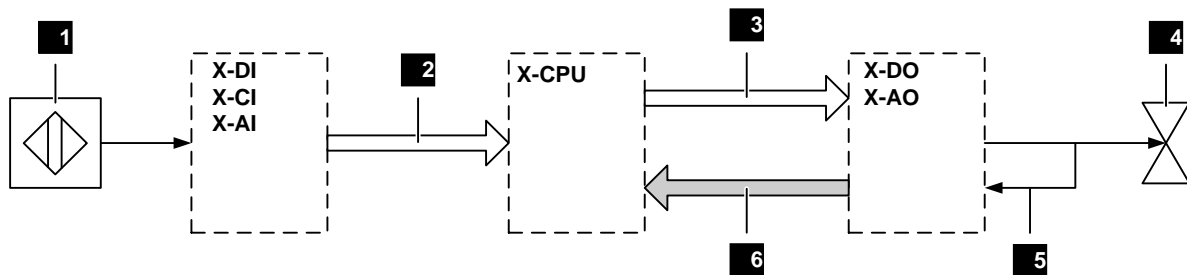
Bei einem Verhältnis von Sicherheitszeit/Watchdog-Zeit = 3/1 wie im Beispiel 2, stehen für die sichere Reaktion noch 2 Zyklen zur Verfügung.

Steht beim nächsten Einlesen **4** die Störung weiterhin an, muss die Fehlerreaktion in diesem Zyklus erfolgen. Der letztmögliche Zeitpunkt der Fehlerreaktion ist das Schreiben der Ausgänge **9**. Zum nächsten Ausgabe-Zeitpunkt **10** ist die Sicherheitszeit abgelaufen.

Bei deaktivierter Störaustastung löst das System beim Einlesen **3** sofort die definierten Fehlerreaktionen aus.

### 3.5.4 Wirkrichtung der Störaustattung

Für die Betrachtung der Störaustattung und der Ausgangs-Störaustattung ist deren Wirkrichtung zu beachten, siehe Bild 12 und folgende Kapitel.



- |  |  |
|--|--|
| <b>1</b> Sensor  | <b>4</b> Aktor   |
| <b>2</b> Wirkrichtung Eingangsmodul zum Prozessormodul     | <b>5</b> Ausgangs-Störaustattung                       |
| <b>3</b> Wirkrichtung vom Prozessormodul zum Ausgangsmodul | <b>6</b> Wirkrichtung Ausgangsmodul zum Prozessormodul |

Bild 12: Wirkrichtungen bei der Störaustattung und der Ausgangs-Störaustattung

#### 3.5.4.1 Wirkrichtung vom Eingangsmodul zum Prozessormodul (**2**)

Die Störaustattung mit der Wirkrichtung vom Eingangsmodul zum Prozessormodul wird vom Prozessormodul ausgeführt. Die Störaustattung unterdrückt dabei die transienten Störungen auf dem Eingangsmodul und auf dem Systembus. Die Störaustattung auf dem Eingangsmodul kann in den Eigenschaften (SILworX) deaktiviert werden (Standard = aktiviert), siehe Handbücher der Eingangsmodule. Die Unterdrückung von transienten Störungen auf dem Systembus ist immer aktiv und kann in SILworX nicht deaktiviert werden.

#### 3.5.4.2 Wirkrichtung vom Prozessormodul zum Ausgangsmodul (**3**)

Die Störaustattung mit der Wirkrichtung vom Prozessormodul zum Ausgangsmodul wird vom Ausgangsmodul ausgeführt und ist immer aktiv. Die Störaustattung unterdrückt dabei die transienten Störungen auf dem Systembus.

#### 3.5.4.3 Wirkrichtung vom Ausgangsmodul zum Prozessormodul (**6**)

Die Störaustattung mit der Wirkrichtung vom Ausgangsmodul zum Prozessormodul auf dem Systembus wird vom Prozessormodul ausgeführt. Die Störaustattung unterdrückt dabei Status-Rückmeldungen des Ausgangsmoduls, wie z. B. die LS/LB-Erkennung. Die Störaustattung auf dem Ausgangsmodul kann in den Eigenschaften (SILworX) deaktiviert werden (Standard = aktiviert), siehe Handbücher der Ausgangsmodule.

#### 3.5.4.4 Ausgangs-Störaustastung (■5)

Die Ausgangs-Störaustastung wird vom Ausgangsmodul selbst ausgeführt. Diese unterdrückt die Abschaltreaktion eines Kanals auf eine Abweichung zwischen Vorgabewert und Rücklesewert des Ausgangskanals. Die Ausgangs-Störaustastung kann für jedes Ausgangsmodul aktiviert werden (Standardeinstellung: Deaktiviert), siehe Handbücher der Ausgangsmodule.

Wenn die Ausgangs-Störaustastung aktiviert ist, kann sich die Reaktionszeit bis auf den Wert *Sicherheitszeit – Watchdog-Zeit* verlängern.

#### 3.5.4.5 LB-Austastung

Die LB-Austastung kann für die Module X-DO 24 01 und X-DO 24 02 parametrierbar werden und wird vom Ausgangsmodul selbst ausgeführt.

Der Parameter *LB-Austastung (Anzahl LS/LB-Intervalle)* definiert die Anzahl von Testintervallen (Parameter *LS/LB-Intervall [μs]*), die ablaufen müssen, bis ein erkannter Feldfehler als Leitungsbruch an das Prozessormodul (X-CPU) übermittelt wird. Bis zur Fehlerreaktion werden transiente Störungen unterdrückt. Die Einstellung von *LB-Austastung (Anzahl LS/LB-Intervalle)* wird für alle Kanäle übernommen.

Mit der Einstellung von *LB-Austastung (Anzahl LS/LB-Intervalle) > 1* verlängert sich die Reaktionszeit. Dies ist bei der Parametrierung der Sicherheitszeit und der Watchdog-Zeit zu beachten.



## 3.6 Alarm- und Ereignisaufzeichnung

Das HIMax System verfügt über die Fähigkeit, Alarmer und Ereignisse aufzuzeichnen (Sequence of Events, SOE).

### 3.6.1 Alarmer und Ereignisse

Ereignisse sind Änderungen des Zustands von Anlage oder Steuerung, die mit einem Zeitstempel versehen sind.

Alarmer sind solche Ereignisse, die eine Erhöhung des Risikopotenzials signalisieren.

Das HIMax System zeichnet als Ereignisse die Zustandsänderungen zusammen mit dem Zeitpunkt ihres Auftretens auf. Der X-OPC-Server kann die Ereignisse auf andere Systeme wie Leitsysteme übertragen, die die Ereignisse darstellen oder auswerten.

HIMax unterscheidet boolesche und skalare Ereignisse.

Boolesche Ereignisse:

- Änderungen von booleschen Variablen, z. B. von digitalen Eingängen.
- Alarm- und Normalzustand, diese sind den Zuständen der Variablen beliebig zu zuordnen.

Skalare Ereignisse:

- Übergänge über Grenzwerte, die für eine skalare Variable definiert sind.
- Skalare Variable haben einen numerischen Datentyp, z. B. INT, REAL.
- Es sind zwei obere und zwei untere Grenzen möglich.
- Für die Grenzwerte muss gelten:  
Oberste Grenze  $\geq$  obere Grenze  $\geq$  Normalbereich  $\geq$  untere Grenze  $\geq$  unterste Grenze.
- Eine Hysterese kann in folgenden Fällen wirken:
  - Bei Unterschreitung einer oberen Grenze.
  - Bei Überschreitung einer unteren Grenze.

Die Angabe einer Hysterese vermeidet eine unnötig große Menge an Ereignissen, wenn die globale Variable stark um einen Grenzwert schwankt.

HIMax kann nur dann Ereignisse bilden, wenn diese in SILworX definiert sind, siehe Kapitel 5.2.11. Die maximale Anzahl der definierbaren Ereignisse beträgt 20 000.

### 3.6.2 Bildung von Ereignissen

Sowohl das Prozessormodul als auch bestimmte Typen von E/A-Modulen sind in der Lage, Ereignisse zu bilden. Diese E/A-Module werden im Folgenden als SOE-Module bezeichnet.

#### 3.6.2.1 Ereignisbildung auf dem Prozessormodul

Das Prozessormodul bildet die Ereignisse aus globalen Variablen und legt sie im Puffer ab, siehe Kapitel 3.6.3. Die Ereignisbildung findet im Zyklus des Anwenderprogramms statt.

#### 3.6.2.2 Ereignisbildung auf SOE-Modulen

SOE-Module können Ereignisse aus den Zuständen von Eingängen bilden. Die Ereignisbildung findet im Zyklus des SOE-Moduls statt.

Das SOE-Modul legt die Ereignisse im Zwischenpuffer ab, aus dem die Prozessormodule lesen. Der Zwischenpuffer ist im flüchtigen Speicher angelegt, so dass beim Abschalten der Versorgungsspannung die Ereignisse verloren gehen.

Jedes gelesene Ereignis kann durch ein neu aufgetretenes Ereignis überschrieben werden.

### 3.6.2.3 Systemereignisse

Außer den Ereignissen, die Änderungen von globalen Variablen oder Eingangssignalen registrieren, bilden die Prozessor- und SOE-Module folgende Arten von Systemereignissen:

- Überlauf: Es sind infolge von Pufferüberlauf Ereignisse nicht gespeichert worden. Der Zeitstempel des Überlauf-Ereignisses entspricht dem des Ereignisses, das den Überlauf erzeugt hat.
- Init: der Ereignispuffer wurde initialisiert.
- Betriebsmodus Stopp: ein SOE-Modul ist in den Zustand STOPP gegangen.
- Betriebsmodus Run: ein SOE-Modul ist in den Zustand RUN gegangen.
- Kommunikationsaufnahme: die Kommunikation zwischen Prozessormodul und SOE-Modul beginnt.
- Kommunikationsverlust: Die Kommunikation zwischen Prozessormodul und SOE-Modul wurde beendet.

Systemereignisse enthalten die SRS-Identifikation des Moduls, das sie ausgelöst hat.

### 3.6.2.4 Statusvariable

Statusvariable stellen dem Anwenderprogramm den Ereigniszustand skalarer Ereignisse zur Verfügung. Jedem der folgenden Zustände kann als Statusvariable eine globale Variable vom Typ BOOL zugeordnet sein:

- Normal.
- Untere Grenze unterschritten.
- Unterste Grenze unterschritten.
- Obere Grenze überschritten.
- Oberste Grenze überschritten.

Die zugeordnete Statusvariable wird TRUE, wenn der betreffende Zustand erreicht ist.

### 3.6.3 Aufzeichnung von Ereignissen

Das Prozessormodul sammelt folgende Ereignisse:

- Von E/A-Modulen gebildete Ereignisse.
- Vom Prozessormodul selbst gebildete Ereignisse.

Das Prozessormodul speichert alle Ereignisse in seinem Puffer. Der Puffer ist im nichtflüchtigen Speicher angelegt und fasst 5000 Ereignisse.

Das Prozessormodul führt die Ereignisse aus verschiedenen Quellen nach Eintreffen zusammen und sortiert sie nicht nach ihrem Zeitstempel.

Ist der Puffer voll, werden keine neuen Ereignisse gespeichert, bis weitere Ereignisse gelesen und dadurch zum Überschreiben markiert wurden.

Zum Forcen im Zusammenhang mit skalaren Ereignissen siehe Kapitel 5.3.7.

### 3.6.4 Weitergabe von Ereignissen

Der X-OPC-Server liest die Ereignisse aus dem Puffer aus und gibt sie an Fremdsysteme zur Darstellung oder Auswertung weiter. 4 OPC-Server können gleichzeitig aus einem Prozessormodul Ereignisse auslesen.

### 3.7 Kommunikation

Die Kommunikation mit anderen HIMA Systemen oder mit Fremdsystemen erfolgt über Kommunikationsmodule. HIMax unterstützt folgende Kommunikationsprotokolle:

- **safeethernet** (sicherheitsbezogen).
- HIPRO-S V2 (sicherheitsbezogen) für die Verbindung mit HIQuad PES.
- Standardprotokolle.

**safeethernet** Verbindungen auch mit Hilfe der Ethernet-Anschlüsse des Prozessormoduls möglich.

Weitere Einzelheiten zur Kommunikation und zu unterstützten Standardprotokollen finden Sie im Kommunikationshandbuch HI 801 100 D und im HIPRO-S V2 Handbuch HI 800 722 D.

#### 3.7.1 ComUserTask (CUT)

Es ist möglich, in der Programmiersprache C Programme zu schreiben, die zyklisch auf dem Kommunikationsmodul ablaufen. Auf diese Weise sind z. B. eigene Kommunikationsprotokolle realisierbar. Die C-Programme sind nicht sicherheitsbezogen.

#### 3.7.2 Lizenzierung Protokolle

Die Standardprotokolle und ComUserTask sind nur dann dauerhaft ablauffähig, wenn eine gültige Lizenz besteht. Einige Protokolle benötigen eine Aktivierung durch einen Software-Freischaltcode. Zur Freischaltung siehe Kapitel 3.9.

---

**i**

Rechtzeitig den Software-Freischaltcode bestellen!

Nach Ablauf von 5000 Betriebsstunden läuft die Kommunikation weiter, bis die Steuerung gestoppt wird. Danach lässt sich das Anwenderprogramm ohne gültigen Software-Freischaltcode für die projektierten Protokolle nicht mehr starten (fehlerhafte Konfiguration).

---

### 3.8 Kommunikation mit Programmiergeräten

Die Kommunikation einer HIMax Steuerung mit einem PADT erfolgt über Ethernet. Ein PADT ist ein Rechner, auf dem das Programmierwerkzeug SILworX installiert ist.

Der Rechner muss die Steuerung über das Ethernet erreichen können.

Das Ethernet zum PADT ist anschließbar an folgende Schnittstellen des HIMax Systems:

- Die mit PADT gekennzeichnete RJ-45-Buchse eines Systembusmoduls.
- Eine RJ-45-Buchse eines Kommunikationsmoduls.
- Eine RJ-45-Buchse eines Prozessormoduls.

Es ist möglich, dass ein HIMax System gleichzeitig mit bis zu 5 PADTs kommuniziert. Dabei kann jedoch nur ein Programmierwerkzeug schreibend auf die Steuerung zugreifen. Alle übrigen PADTs können nur Informationen auslesen. Bei jedem weiteren Versuch, eine schreibende Verbindung aufzubauen, erteilt die Steuerung nur einen lesenden Zugriff.

### 3.9 Lizenzierung

Einige Funktionen des HIMax Systems benötigen eine Lizenz:

- Remote Rack  
Diese Lizenz schaltet folgende Funktionen frei:
  - Aufbau des Systems in Netzstruktur.
  - Eintrag einer maximalen Systembus-Latenzzeit  $\geq 100 \mu\text{s}$ .
- Einige Kommunikationsprotokolle, weitere Einzelheiten finden Sie im Kommunikationshandbuch HI 801 100 D.

Die Lizenzen können von HIMA kostenpflichtig bezogen werden. Zur Aktivierung der Funktion ist ein Freischalt-Code von HIMA nötig, der dann mit dem PADT in die Konfiguration eingetragen werden kann. Der Freischalt-Code ist an die System-ID des PES gebunden.

Die Generierung des Freischalt-Codes erfolgt auf der HIMA Webseite [www.hima.com/de](http://www.hima.com/de) unter Produkte & Services -> Produkt-Registrierung. Die Einzelheiten sind auf der entsprechenden Unterseite beschrieben.

#### Eine Funktion durch einen Freischaltcode aktivieren

1. Den Software-Freischaltcode auf der HIMA Webseite [www.hima.com/de](http://www.hima.com/de) mit der System-ID (z. B. 10 000) der Steuerung und den von HIMA erhaltenen Lizenznummern generieren. Dazu den Anweisungen auf der HIMA Webseite folgen!

---

**i**

Der Software-Freischaltcode ist untrennbar mit dieser System-ID verbunden. Eine Lizenz kann nur einmalig für eine bestimmte System-ID genutzt werden. Deshalb sollte die Freischaltung erst durchgeführt werden, wenn die System-ID eindeutig feststeht.

---

2. In SILworX für die Ressource eine Lizenzverwaltung anlegen, falls noch keine existiert.
  3. In der Lizenzverwaltung einen Lizenzschlüssel anlegen und den Freischaltcode eintragen.
  4. Das Projekt kompilieren und auf die Steuerung laden.
- Die Funktion ist freigeschaltet.

## 4 Redundanz

Das HIMax System ist als hochverfügbares System konzipiert. Zu diesem Zweck lassen sich alle Systembestandteile redundant betreiben. Dieses Kapitel beschreibt die Redundanz für die verschiedenen Systembestandteile.

---

### i

Ein redundantes System erhöht ausschließlich die Verfügbarkeit, nicht den Sicherheitsintegritätslevel (SIL)!

---

### 4.1 Prozessormodul

Ein HIMax System kann als Mono-System mit nur einem Prozessormodul oder als hochverfügbares System mit bis zu 4 redundanten Prozessormodulen aufgebaut werden.

Ein System mit redundanten Prozessormodulen benötigt immer auch einen redundanten Systembus.

Die Redundanz der Prozessormodule muss im Hardware-Editor (SILworX) als digitales Abbild der Steuerung angelegt werden. Danach muss die kompilierte Konfiguration auf die Prozessormodule übertragen werden. Nach der Validierung auf den Prozessormodulen kann der redundante Betrieb aufgenommen werden.

#### 4.1.1 Verminderung der Redundanz

Bei einem HIMax System mit zwei- bis vierfacher Redundanz der Prozessormodule läuft der sicherheitsbezogene Betrieb auch dann weiter, wenn eines der Prozessormodule nicht mehr verfügbar ist, z. B. durch Ausfall oder Ziehen des Moduls. Auch bei Ausfall mehrerer Prozessormodule bleibt der sicherheitsbezogene Betrieb gewährleistet.

#### 4.1.2 Redundanzaufrüstung

Ein zu einem laufenden HIMax System hinzugefügtes Prozessormodul synchronisiert sich automatisch mit der Konfiguration der vorhandenen Prozessormodule. Der sicherheitsbezogene Betrieb bleibt gewährleistet.

Voraussetzungen:

- Das vom Prozessormodul ausgeführte Anwenderprogramm ist redundant parametrisiert (Standard-Einstellung).
- Einer der Steckplätze 4, 5, 6 auf dem Rack 0 oder 3, 4 auf dem Rack 1 ist noch frei.
- Mindestens ein Systembus ist in Betrieb.
- Der Mode-Schalter des hinzugefügten Prozessormoduls ist in Stellung *Stop* oder *Run*.
- Die Betriebssystem-Version des hinzugefügten Prozessormoduls ist entweder dieselbe Version wie bei den vorhandenen Prozessormodulen oder eine höhere Version.

#### 4.1.3 Prozessormodul X-CPU 31

Beim Prozessormodul X-CPU 31 ist nur eine zweifache Redundanz möglich, da der Einsatz auf die Steckplätze 1 und 2 des Racks 0 beschränkt ist. Ein Prozessormodul X-CPU 01 kann nicht redundant zu einem Prozessormodul X-CPU 31 arbeiten.

## 4.2 Redundanz von E/A-Modulen

Bei der Redundanz von Eingangs- und Ausgangsmodulen sind die beiden folgenden Fälle zu unterscheiden:

- Modulredundanz.
- Kanalredundanz.

HIMA empfiehlt die Modulredundanz vor der Kanalredundanz festzulegen. HIMax unterstützt die zweifache Redundanz von Eingang- und Ausgangsmodulen. Höhere Redundanzstufen sind mit Hilfe entsprechender Programmlogik möglich. Bei der Redundanz-Verschaltung eines Sensors oder Aktors auf mehrere E/A-Module sind die zulässigen Eingangs- und Ausgangswerte der E/A-Module zu beachten.

### 4.2.1 Modulredundanz

2 E/A-Module gleichen Typs sind im Programmiersystem als zueinander redundant definiert. Sie bilden eine Redundanzgruppe.

#### 4.2.1.1 Reserve-Module

Zueinander redundante Module können in SILworX das Attribut *Reserve-Modul* erhalten. Das verhindert eine Fehlermeldung beim Fehlen oder Ausfall des redundanten Moduls.

### 4.2.2 Kanalredundanz

Voraussetzung: Zwei Module sind redundant zueinander definiert.

Kanäle mit derselben Kanalnummer sind als redundant zueinander definierbar.

In diesem Fall ordnet das Programmierwerkzeug eine globale Variable, die einem Kanal (einer Kanalnummer) zugeordnet ist, automatisch beiden Kanälen der redundanten Module zu. Näheres in der Online-Hilfe zum Hardware-Editor von SILworX.

Bei Eingangskanälen kann festgelegt werden, auf welche Weise die Steuerung die Signale der beiden redundanten Kanäle zu einem resultierenden Wert verknüpft. Diesen Wert nimmt die globale Variable an.

Es müssen nicht alle Kanäle zweier redundanter Module redundant zugewiesen sein.

### 4.2.3 Connector Boards für redundante Module

In vielen Anwendungsfällen sind bei zwei redundanten Modulen alle Kanäle redundant, aber die angeschlossenen Transmitter oder Aktoren sind nicht redundant.

In diesen Fällen lässt sich auf folgende Weise Verdrahtungsaufwand sparen:

- Ein für diesen Zweck vorgesehenes Connector Board benutzen, das zwei Steckplätze belegt.
- Die beiden redundanten Module auf benachbarte Steckplätze stecken.
- Es ist nur einmal notwendig, die Verbindungen zum Feld auf dem Connector Board herzustellen.

Weitere Einzelheiten zu den Connector Boards finden Sie in den Modulhandbüchern.

## 4.3 Systembus-Redundanz

Das HIMax System kann mit redundanten Systembussen A und B betrieben werden.

Voraussetzungen für einen redundanten Betrieb:

- Einsatz von zwei Systembusmodulen je Rack.
- Geeignete Konfiguration der Systembusmodule.
- Verbindung der Racks einer Steuerung, siehe Kapitel 3.2.

HIMA empfiehlt, die Systembusse A und B auch dann redundant zu betreiben, wenn ein nicht-redundanter Betrieb möglich ist, siehe Variante 1 in Kapitel 3.3.2.

## 4.4 Kommunikations-Redundanz

Weitere Einzelheiten finden Sie in der SILworX Online-Hilfe oder im Kommunikationshandbuch HI 801 100 D.

### 4.4.1 safeethernet

Die Redundanz ist im **safeethernet** Editor in SILworX zu konfigurieren. Eine Kommunikationsverbindung ist redundant, wenn zwei gleiche physikalische Übertragungswege existieren.

### 4.4.2 Standardprotokolle

Bei Standardprotokollen muss das Anwenderprogramm die Redundanz verwalten, ausgenommen bei Modbus-Slave.

## 4.5 Spannungsversorgung

Das HIMax System lässt sich mit einer redundanten Spannungsversorgung betreiben. Der Anschluss der Spannungsversorgungen erfolgt am Klemmenblock an den Klemmen L1+/L1- für die erste Spannungsversorgung und L2+/L2- für die redundante Spannungsversorgung. Jedes Modul enthält intern eine Entkopplung der beiden Anschlüsse für die Versorgungsspannung.

Bei Connector Boards mit externer Speisung ist eine redundante Speisung außerhalb des HIMax Systems bereitzustellen.

Weitere Einzelheiten finden Sie in den Modulhandbüchern.

## 4.6 Mono-Betrieb

Für Anwendungen, bei denen auf einen redundanten Betrieb verzichtet werden kann, ist der Mono-Betrieb angemessen.

Bedingungen für Mono-Betrieb:

- Das Projekt enthält nur ein Prozessormodul in Rack 0, entweder vom Typ X-CPU 01 auf Steckplatz 3 oder vom Typ X-CPU 31 auf Steckplatz 1.
- Der Datenverkehr zwischen den Modulen wird nur über Systembus A abgewickelt.
- Das Projekt ist in SILworX entsprechend konfiguriert.

Folgende Konfigurationen in Rack 0 sind für Mono-Betrieb geeignet:

- Ein Prozessormodul X-CPU 01 in Steckplatz 3, ein Systembusmodul in Steckplatz 1.
- Ein Prozessormodul X-CPU 31 in Steckplatz 1.

Dazu gehören E/A-Module und evtl. Kommunikationsmodule nach Projektanforderungen.

Nach Verbindungsaufnahme mit der Steuerung ist der Schalter für Mono-Startup zu setzen. Dieser Schalter bleibt bei Spannungsausfall gesetzt. Das Laden einer redundanten Projektkonfiguration in die Steuerung (PES) setzt diesen Schalter zurück. Zum Setzen des Schalters für Mono-Startup siehe Kapitel 8.4.1.

HIMA empfiehlt den Einsatz redundanter Systembusse und Systembusmodule!

## 5 Programmierung

Die Anwenderprogramme für das HIMax System sind mit Hilfe eines PADTs zu erstellen, das aus einem PC mit dem Programmierwerkzeug SILworX besteht. Ein Anwenderprogramm besteht aus Standard-Funktionsbausteinen gemäß IEC 61131-3, aus benutzerdefinierten Funktionsbausteinen und aus Variablen und Konnektoren. Der FBS-Editor von SILworX dient dazu, die Elemente zu platzieren und sie grafisch miteinander zu verbinden. Aus dieser grafischen Darstellung erzeugt SILworX ein ablauffähiges Programm, das in die Steuerung geladen werden kann.

Einzelheiten des Programmierwerkzeugs in der Online-Hilfe von SILworX.

Bis zu 32 Anwenderprogramme lassen sich in eine Steuerung laden. Die Steuerung arbeitet die Anwenderprogramme gleichzeitig ab. Die Programme können mit einstellbaren Prioritäten ablaufen.

### 5.1 Anschluss des Programmierwerkzeugs

Der PC mit dem Programmierwerkzeug SILworX (PADT) wird über Ethernet mit dem HIMax System verbunden. Folgende Schnittstellen stehen zur Verfügung:

- Die Ethernet-Schnittstellen **PADT** der Systembusmodule (X-SB). An diesen Schnittstellen sind nur Patchkabel mit gekreuzten Adern zulässig.
- Die Ethernet-Schnittstellen der Kommunikationsmodule (X-COM).
- Die Ethernet-Schnittstellen der Prozessormodule (X-CPU).

#### 5.1.1 Verwendung der Ethernet-Schnittstellen

Aus Gründen der Automation Security sind die Ethernet-Schnittstellen, in Abhängigkeit vom Systemaufbau, in folgender Reihenfolge zu verwenden:

1. **PADT**-Schnittstelle einer X-SB, die nicht *Responsible* ist.
2. Ethernet-Schnittstelle einer X-COM.
3. Ethernet-Schnittstelle einer X-CPU.
4. **PADT**-Schnittstelle einer X-SB, die *Responsible* ist.

### 5.2 Verwendung von Variablen in einem Projekt

Eine Variable ist ein Platzhalter für einen Wert innerhalb der Programmlogik. Über den Variablennamen wird der Speicherplatz mit dem gespeicherten Wert symbolisch adressiert.

Die Verwendung von symbolischen Namen an Stelle der physikalischen Adresse hat zwei wesentliche Vorteile:

- Im Anwenderprogramm sind die im Prozess verwendeten Bezeichnungen von Eingängen und Ausgängen verwendbar.
- Änderungen der Zuordnung der Variable zu den Eingangs- und Ausgangskanälen haben keinen Einfluss auf das Anwenderprogramm.

Es gibt lokale und globale Variable. Die lokalen Variablen gelten nur in einem eng abgegrenzten Bereich des Projekts, einem Anwenderprogramm oder Funktionsbaustein. Die globalen Variablen gelten in mehreren Bausteinen oder Programmen und können Daten zwischen den Bausteinen austauschen.

Globale Variable können auf verschiedenen Ebenen des Projektbaums erstellt werden. Die globalen Variablen gelten für alle zum Geltungsbereich gehörenden untergeordneten Ebenen.

Beispiel: Ist ein Projekt aus mehreren Ressourcen aufgebaut, so gelten die unter einer Ressource angelegten globalen Variablen ausschließlich unterhalb dieser Ressource.



Hierarchie der Ebenen, auf denen globale Variable definierbar sind:

1. Projekt.
2. Konfiguration.
3. Ressource.

Globale Daten dürfen nur an einer Stelle im Programm mit Werten beschrieben werden! Die möglichen Quellen sind:

- Logik in einem Anwenderprogramm.
- Eingänge.
- Systemvariable.
- Kommunikationsprotokolle.

Das Beschreiben globaler Variablen an mehreren Stellen im Programm kann zu unbeabsichtigten Resultaten führen!

Mit der Funktion *Querverweis in Spalte* im Globale-Variable-Editor ist die Verwendung globaler Daten zu prüfen.

## 5.2.1 Variablentypen

SILworX unterstützt folgende Variablentypen:

- VAR, eine Variable innerhalb einer Logik (lesen und schreiben).
- VAR mit Attribut CONST, eine Variable wurde als Konstante definiert und kann nicht geändert werden.
- VAR mit Attribut RETAIN, eine Variable verliert ihren Wert nach einem Stromausfall nicht.
- VAR\_EXTERNAL, Verweis auf Globale Variablen (lesen und schreiben).
- VAR\_GLOBAL, Globale Variable (lesen und schreiben) zum Austausch von Werten zwischen Programmen und untergeordneten Funktionen und Funktionsbausteinen.
- VAR\_INPUT, Eingangsvariable (lesen) einer POE. Wird auch in der Schnittstellenanzeige angezeigt.
- VAR\_OUTPUT, Ausgangsvariable (schreiben) einer POE. Wird auch in der Schnittstellenanzeige angezeigt.
- VAR\_TEMP, temporäre Variable (lesen und schreiben).
- VAR\_ACTION, Aktionsdeklaration (lesen und schreiben).

Der einer Variable zuweisbare Variablentyp ist abhängig von der Hierarchie der Variable im Strukturbaum. Die nachfolgende Tabelle zeigt die erlaubten Variablentypen in Abhängigkeit der Strukturbaum-Knoten.

Typ	Projekt	Konfiguration	Ressource	Programm-Typ	FB-Typ	Funktions-Typ
VAR				X		
VAR_EXTERNAL				X		
VAR_GLOBAL	X	X	X	(1)		
VAR_INPUT				(1)	X	X
VAR_OUTPUT				(1)	X	X
VAR_TEMP				(2)	X	
VAR_ACTION				(2)	X	
(1) Abweichend von der Norm wird diese Funktion nicht unterstützt.						
(2) VAR_ACTION wird abweichend von der Norm unterstützt.						

Tabelle 8: Unterstützte Variablentypen

### 5.2.2 Initialwert

Jeder Variable lässt sich ein Initialwert zuweisen. Diesen Wert nimmt die Variable in den Fällen an, in denen das Programm ihr keinen Wert zugewiesen hat:

- Beim Programmstart.
- Bei einem Fehler der Quelle, von der die Variable ihren Wert übernimmt z. B.:
  - Physikalischer Eingang.
  - Kommunikationsschnittstelle.
  - Anwenderprogramm im Zustand STOP.

Initialwerte werden wie folgt angegeben:

- Es gilt der Initialwert gemäß der Datentypdeklaration. Für elementare Datentypen gelten die Initialwerte gemäß IEC 61131-3, Tabelle *Wertebereiche und Initialisierungswerte* der Datentypen. Dieser Initialwert hat die niedrigste Priorität.
- Es gilt der Initialwert, der vom Anwender für einen abgeleiteten Datentyp angegeben wurde. Es wird nicht mehr der Initialwert des Basistyps geerbt.
- Es gilt der Initialwert, der vom Anwender bei der Variablendeklaration explizit für eine Variable angegeben wurde.
- Es gilt der instanzspezifische Initialwert, welcher zum Zeitpunkt der Initialisierung bereits definiert ist. Dieser Initialwert hat die höchste Priorität.

Bei **safeethernet** und bei Kommunikationsprotokollen ist einstellbar, welchen Wert die angeschlossene Variable bei Fehlern annehmen soll.



In sicheren Anwendungen muss für alle Variablen, die ihren Wert von einem physikalischen Eingang oder aus der Kommunikation erhalten, ein sicherer Wert als Initialwert angegeben werden!

---

### 5.2.3 Systemvariablen und Systemparameter

*Systemvariable* sind vordefinierte Variable zur Verarbeitung von Eigenschaften und Zuständen des HIMax Systems im Anwenderprogramm. Zu diesem Zweck sind den Systemvariablen globale Variable zuzuweisen, die im Anwenderprogramm verwendet werden.

Über *Systemparameter* sind die Eigenschaften der Steuerung einstellbar. Systemparameter, die nur die Werte TRUE und FALSE haben können, werden auch als Schalter bezeichnet.

Systemvariable und Systemparameter sind auf verschiedenen Ebenen des Projekts definiert. Das Einstellen der Systemvariablen und Systemparameter erfolgt in SILworX, entweder im Eigenschaftsdialog des entsprechenden Zweigs im Strukturbaum oder in einer Detailansicht im Hardware-Editor.

Projektebene	Beschreibung der Systemvariablen und Systemparameter
Ressource	Siehe Tabelle 10.
Hardware allgemein	<ul style="list-style-type: none"> <li>▪ Systemvariable zum Einstellen der Steuerung siehe Tabelle 10.</li> <li>▪ Systemvariable, die Informationen liefern, siehe Tabelle 13 und Tabelle 14.</li> </ul>
Hardware: Module	Einzelheiten finden Sie in den Modulhandbüchern. Einstellung der Systemvariablen und Systemparameter im Hardware-Editor, in der Detailansicht des Moduls.
Anwenderprogramm	Siehe Kapitel 5.2.6.

Tabelle 9: Systemvariable auf unterschiedlichen Projektebenen

## 5.2.4 Systemparameter der Ressource

Die Systemparameter der Ressource legen das Verhalten der Steuerung während des Betriebs fest. Die Systemparameter sind in SILworX im Dialog *Eigenschaften* der Ressource einstellbar.

Systemparameter	S <sup>1)</sup>	Beschreibung	Einstellung für sicheren Betrieb
Name	N	Name der Ressource	Beliebig
System-ID [SRS]	J	System-ID der Ressource Wertebereich: 1 ... 65 535 Standardwert: 60 000 Es ist notwendig, der System-ID einen anderen Wert als den Standardwert zu zuweisen, sonst ist das Projekt nicht ablauffähig!	Eindeutiger Wert innerhalb des Netzwerks der Steuerungen. Das sind alle Steuerungen, die potenziell miteinander verbunden sind.
Sicherheitszeit [ms]	J	Sicherheitszeit der Ressource in Millisekunden, siehe Kapitel <i>Sicherheitsrelevante Zeiten</i> im Sicherheitshandbuch. Wertebereich: 20 ... 22 500 ms. Standardwert: 600 ms (online änderbar)	Applikations-spezifisch
Watchdog-Zeit [ms]	J	Watchdog-Zeit in Millisekunden, siehe Kapitel <i>Sicherheitsrelevante Zeiten</i> im Sicherheitshandbuch. Wertebereich: 6 ... 7500 ms Standardwert: 200 ms (online änderbar)	Applikations-spezifisch
Sollzykluszeit [ms]	N	Gewünschte oder maximale Zykluszeit, siehe <i>Sollzykluszeit-Modus</i> . Wertebereich 0 ... 7500 ms Standardwert: 0 ms (online änderbar) Die Sollzykluszeit darf höchstens so groß sein wie die eingestellte <i>Watchdog-Zeit [ms]</i> abzüglich des kleinsten einstellbaren Werts der <i>Watchdog-Zeit [ms]</i> (6 ms, s. o.), andernfalls wird die Eingabe abgelehnt. Ist der Standardwert 0 ms eingestellt, so wird die Sollzykluszeit nicht beachtet. Weitere Details siehe nachfolgende Kapitel.	Applikations-spezifisch
Sollzykluszeit-Modus	N	Verwendung der <i>Sollzykluszeit [ms]</i> , siehe nachfolgende Kapitel. Die Standardeinstellung ist fest-tolerant (online änderbar).	Applikations-spezifisch
Multitasking-Modus	N	Mode 1 Die Länge eines Zyklus der CPU richtet sich nach der benötigten Ausführungsdauer aller Anwenderprogramme.	Applikations-spezifisch
		Mode 2 Prozessor stellt von Anwenderprogrammen niederer Priorität nicht benötigte Ausführungszeit den Anwenderprogrammen hoher Priorität zur Verfügung. Betriebsart für hohe Verfügbarkeit.	
		Mode 3 Prozessor wartet nicht benötigte Ausführungszeit von Anwenderprogrammen ab und verlängert so den Zyklus.	
		Standardwert: Mode 1	
Max. Kom.-Zeitscheibe [ms]	N	Höchstwert in ms der Zeitscheibe, die innerhalb des Zyklus der Ressource für Kommunikation verwendet wird, siehe Kommunikationshandbuch HI 801 100 D. Wertebereich: 2 ... 5000 ms Standardwert: 60 ms	---

Systemparameter	S <sup>1)</sup>	Beschreibung	Einstellung für sicheren Betrieb
Optimierte Nutzung der Kom.-Zeitscheibe	N	<p>Der Systemparameter verkürzt die Antwortzeiten für die Kommunikation über das oder die Prozessormodule.</p> <hr/> <p><b>i</b> Es kann sich die zeitliche Ausnutzung der <i>Max. Kom.-Zeitscheibe [ms]</i> und somit der Systemparameter <i>Max. Dauer Konfigurationsverbindungen [ms]</i> ändern, so dass diese stärker beansprucht werden können, z. B. beim Reload.</p> <hr/>	---
Max. Dauer Konfigurationsverbindungen [ms]	N	<p>Definiert, wie viel Zeit innerhalb eines CPU-Zyklus für die Konfigurationsverbindungen zur Verfügung steht:  Wertebereich: 2 ... 3500 ms  Standardwert: 20 ms  Weitere Details siehe nachfolgende Kapitel.</p>	Applikations-spezifisch
Maximale Systembus-Latenzzeit [µs]	N	<p>Maximale Verzögerung einer Nachricht zwischen einem E/A-Modul und einem Prozessormodul. 100 ... 50 000 µs,  Standardwert: <i>System-Standardwerte</i></p> <hr/> <p><b>i</b> Für die Einstellung der maximalen Systembuslatenz auf einen Wert <math>\neq</math> <i>System-Standardwerte</i> ist eine Lizenz erforderlich.</p> <hr/>	Applikations-spezifisch
Online-Einstellungen erlauben	J	<p>TRUE: <b>Alle</b> unter FALSE genannten Schalter/Parameter sind online mit dem PADT änderbar. Dies gilt nur, wenn die Systemvariable <i>Read-only in RUN</i> den Wert FALSE hat.  Standardwert: TRUE.</p>	HIMA empfiehlt die Einstellung FALSE.
		<p>FALSE: Folgende Parameter sind <b>nicht</b> online änderbar:</p> <ul style="list-style-type: none"> <li>▪ <i>System-ID</i></li> <li>▪ <i>Autostart</i></li> <li>▪ <i>Globales Forcen erlaubt</i></li> <li>▪ <i>Globales MultiForcen erlaubt</i></li> <li>▪ <i>Globale Force-Timeout-Reaktion</i></li> <li>▪ <i>Laden erlaubt</i></li> <li>▪ <i>Reload erlaubt</i></li> <li>▪ <i>Start erlaubt</i></li> </ul> <p>Wenn <i>Reload erlaubt</i> = TRUE ist, sind folgende Parameter online änderbar:</p> <ul style="list-style-type: none"> <li>▪ <i>Watchdog-Zeit (der Ressource)</i></li> <li>▪ <i>Sicherheitszeit</i></li> <li>▪ <i>Sollzykluszeit</i></li> <li>▪ <i>Sollzykluszeit-Modus</i></li> </ul>	
		<p>Bei gestoppter Steuerung und durch einen Reload ist es möglich, <i>Online-Einstellungen erlauben</i> = TRUE zu setzen.</p>	

Systemparameter	S <sup>1)</sup>	Beschreibung		Einstellung für sicheren Betrieb
Autostart	J	TRUE:	Wenn die Steuerung an die Versorgungsspannung angeschlossen wird, starten die Anwenderprogramme automatisch. Standardwert: TRUE.	Applikations-spezifisch
		FALSE:	Kein automatischer Start nach Zuschalten der Versorgungsspannung.	
		Einstellungen in den Programm-Eigenschaften der Ressource beachten!		
Start erlaubt	J	TRUE:	Kaltstart oder Warmstart durch PADT im Zustand RUN oder STOPP erlaubt. Standardwert: TRUE.	Applikations-spezifisch
		FALSE:	Kein Start erlaubt.	
Laden erlaubt	J	TRUE:	Download der Konfiguration erlaubt. Standardwert: TRUE.	Applikations-spezifisch
		FALSE:	Kein Start erlaubt.	
Reload erlaubt	J	TRUE:	Reload der Konfiguration erlaubt. Standardwert: TRUE.	Applikations-spezifisch
		FALSE:	Reload der Konfiguration nicht erlaubt. Ein laufender Reload-Prozess wird beim Umschalten auf FALSE nicht abgebrochen.	
Globales Forcen erlaubt	J	TRUE:	Globales Forcen für diese Ressource erlaubt. Standardwert: TRUE.	Applikations-spezifisch
		FALSE:	Globales Forcen für diese Ressource nicht erlaubt.	
Globale Force-Timeout-Reaktion	N	Legt fest, wie sich die Ressource beim Ablauf des globalen Force-Timeout verhält: <ul style="list-style-type: none"><li>Nur Forcen beenden.</li><li>Forcen beenden und Ressource stoppen.</li></ul> Standardwert: Nur Forcen beenden.		Applikations-spezifisch
Globales MultiForcen erlaubt	J	TRUE:	Anwender mit MultiForcen-Zugriff können in einer Ressource Force-Daten (Force-Werte und Force-Einzelschalter) für globale Variablen schreiben, wenn die dafür erforderlichen, übergeordneten Bedingungen gegeben und Force-Freigaben erfolgt sind.	Applikations-spezifisch
		FALSE:	Anwender mit MultiForcen-Zugriff können keine globale Variablen forcen. Standardwert: FALSE (online änderbar).	
Minimale Konfigurations-version	N	Mit dieser Einstellung ist es möglich, Code zu generieren, der entsprechend den Projektanforderungen zu alten oder zu neuen Versionen des HIMax Betriebssystems kompatibel ist. Als Standardwert wird die installierte SILworX Version angezeigt.		Applikations-spezifisch
Schneller Hochlauf	N	Für HIMax nicht anwendbar.		---

<sup>1)</sup> Systemparameter wird vom Betriebssystem sicherheitsbezogen behandelt, ja (J) oder nein (N).

<sup>1)</sup> Systemparameter wird vom Betriebssystem sicherheitsbezogen behandelt, ja (J) oder nein (N).

Tabelle 10: Die Systemparameter der Ressource

#### 5.2.4.1 Hinweise zum Parameter *Minimale Konfigurationsversion*

- Bei einem neu angelegten Projekt wird jeweils die neueste *Minimale Konfigurationsversion* ausgewählt. Ob diese Einstellung zur verwendeten Betriebssystem-Version passt, ist zu prüfen.
- Bei einem Projekt, das von einer früheren SILworX Version konvertiert wurde, bleibt der in der Vorversion eingestellte Wert für die *Minimale Konfigurationsversion* erhalten. Dadurch ist gewährleistet, dass bei der Codegenerierung derselbe Konfigurations-CRC erzeugt wird wie in der Vorversion, und die generierte Konfiguration kompatibel zu den Betriebssystemen der Module bleibt.

Bei konvertierten Projekten sollte deshalb die *Minimale Konfigurationsversion* nur im Zusammenhang mit anderen Änderungen an der betroffenen Ressource verändert werden.

- SILworX erzeugt automatisch eine höhere Konfigurationsversion als die eingestellte *Minimale Konfigurationsversion*, wenn im Projekt Fähigkeiten benutzt werden, die nur eine höhere Konfigurationsversion zur Verfügung stellt. Dies zeigt SILworX im Ergebnis der Codegenerierung an. Die Module lehnen das Laden höherer Konfigurationsversionen ab, die nicht zu ihren Betriebssystemen passen.

Für die Beseitigung von solchen Inkompatibilitäten kann es nützlich sein, den vom Versionsvergleich gelieferten Informationen die Moduldaten-Übersicht gegenüber zu stellen.

- Bei Verwendung von Prozessormodulen X-CPU 31 ist *Minimale Konfigurationsversion* auf *SILworX V6* oder höher einzustellen.

#### 5.2.4.2 Verwendung der Parameter *Sollzykluszeit* und *Sollzykluszeit-Modus*

Mit den Einstellungen im Systemparameter *Sollzykluszeit-Modus* kann die Zykluszeit möglichst konstant auf dem Wert der *Sollzykluszeit [ms]* gehalten werden. Dazu muss der Systemparameter auf einen Wert > 0 eingestellt sein.

HIMax begrenzt dabei den Reload und die Synchronisierung redundanter Prozessormodule soweit, dass die *Sollzykluszeit* eingehalten wird.

Die folgende Tabelle beschreibt die Einstellungen im Systemparameter *Sollzykluszeit-Modus*:

Einstellung	Beschreibung
fest	<p>Ist ein CPU-Zyklus kürzer als die definierte <i>Sollzykluszeit</i>, wird der CPU-Zyklus bis zur <i>Sollzykluszeit</i> verlängert.</p> <p>Ist der CPU-Zyklus länger als die <i>Sollzykluszeit</i>, setzt die CPU den Zyklus ohne Verzögerung fort.</p> <hr/> <p><b>i</b> Ein Reload oder eine Aufsynchronisation wird abgelehnt, wenn die Reservezeit (<i>Sollzykluszeit</i> minus tatsächliche Zykluszeit) nicht ausreicht.</p>
fest-tolerant	<p>Wie <i>fest</i>, jedoch mit den folgenden Unterschieden:</p> <ol style="list-style-type: none"> <li>1. Wenn erforderlich wird bei der Aufsynchronisation die <i>Sollzykluszeit</i> für einen CPU-Zyklus nicht eingehalten, um die Aufsynchronisation erfolgreich durchführen zu können.</li> <li>2. Wenn erforderlich wird beim Reload die <i>Sollzykluszeit</i> für 1 bis n (n = Anzahl der geänderten Anwenderprogramme) CPU-Zyklen nicht eingehalten, um den Reload erfolgreich durchführen zu können.</li> </ol> <p>Die Standardeinstellung ist <i>fest-tolerant</i>!</p> <hr/> <p><b>i</b> Nach dem 1. Reload-Aktivierungszyklus gelten Watchdog-Zeit, <i>Sollzykluszeit</i> und <i>Sollzykluszeit-Modus</i> gemäß der neuen Konfiguration. Höchstens jeder fünfte Zyklus kann während des Reload verlängert werden. Ein einziger Zyklus kann während der Synchronisation verlängert werden.</p>
dynamisch	<p>Die CPU führt jeden CPU-Zyklus so schnell wie möglich aus. Dies entspricht einer eingestellten <i>Sollzykluszeit</i> von 0 ms.</p> <hr/> <p><b>i</b> Ein Reload oder eine Aufsynchronisation wird abgelehnt, wenn die Reservezeit (<i>Sollzykluszeit</i> minus tatsächliche Zykluszeit) nicht ausreicht. Höchstens jeder fünfte Zyklus kann während des Reload verlängert werden. Ein einziger Zyklus kann während der Synchronisation verlängert werden.</p>
dynamisch-tolerant	<p>Wie <i>dynamisch</i>, jedoch mit den folgenden Unterschieden:</p> <ol style="list-style-type: none"> <li>1. Wenn erforderlich wird bei der Aufsynchronisation die <i>Sollzykluszeit</i> für einen CPU-Zyklus automatisch erhöht, um die Aufsynchronisation erfolgreich durchführen zu können.</li> <li>2. Wenn erforderlich wird beim Reload die <i>Sollzykluszeit</i> für 1 bis n (n = Anzahl der geänderten Anwenderprogramme) CPU-Zyklen automatisch erhöht, um den Reload erfolgreich durchführen zu können.</li> </ol> <hr/> <p><b>i</b> Nach dem 1. Reload-Aktivierungszyklus gelten Watchdog-Zeit, <i>Sollzykluszeit</i> und <i>Sollzykluszeit-Modus</i> gemäß der neuen Konfiguration. Ein Reload oder eine Aufsynchronisation wird abgelehnt, wenn die Reservezeit (<i>Sollzykluszeit</i> minus tatsächliche Zykluszeit) nicht ausreicht.</p>

Tabelle 11: Einstellungen *Sollzykluszeit-Modus*

#### 5.2.4.3 Maximale Kommunikationszeitscheibe

Die maximale Kommunikationszeitscheibe ist die zugeteilte Zeit in Millisekunden (ms) pro CPU-Zyklus, innerhalb welcher das Prozessormodul die Kommunikationsaufgaben abarbeitet. Auch wenn die Protokollverarbeitung innerhalb der Dauer einer Kommunikationszeitscheibe nicht beendet werden konnte, führt das Prozessormodul dennoch die sicherheitsrelevanten Überwachungen für alle Protokolle in einem CPU-Zyklus aus.

**i**

Können nicht alle in einem CPU-Zyklus anstehenden Kommunikationsaufgaben ausgeführt werden, erfolgt die komplette Übertragung der Kommunikationsdaten über mehrere CPU-Zyklen. Die Anzahl der Kommunikationszeitscheiben ist dann größer 1.

Für die Berechnungen der zulässigen maximalen Reaktionszeiten gilt die Bedingung, dass die Anzahl der Kommunikationszeitscheiben genau 1 ist.

#### 5.2.4.4 Ermitteln der maximalen Dauer der Kommunikationszeitscheibe

Für eine erste Abschätzung der maximalen Dauer der Kommunikationszeitscheibe müssen die folgenden Zeiten aufsummiert und das Ergebnis in den Systemparameter *Max. Kom.-Zeitscheibe [ms]* in den Eigenschaften der Ressource eingetragen werden:

- Pro Kommunikationsmodul (X-COM 01) 3 ms.
- Pro redundante **safeethernet** Verbindung 1 ms.
- Pro nicht redundante **safeethernet** Verbindung 0,5 ms.
- Pro kByte Nutzdaten bei nichtsicheren Protokollen (z. B. Modbus) 1 ms.

HIMA empfiehlt, den abgeschätzten Wert *Max. Kom.-Zeitscheibe [ms]* mit dem im Control Panel angezeigten Wert zu vergleichen und gegebenenfalls in den Eigenschaften der Ressource zu korrigieren. Dies kann z. B. in einem FAT (Factory Acceptance Test) oder SAT (Site Acceptance Test) durchgeführt werden.

##### Ermitteln der tatsächlichen Dauer der maximalen Kommunikationszeitscheibe

1. Das HIMax System unter voller Last betreiben (FAT, SAT):  
Alle Kommunikationsprotokolle sind in Betrieb (**safeethernet** und Standardprotokolle).
2. Das **Control Panel** öffnen und im Strukturbaum das Verzeichnis **Kom.-Zeitscheibe** wählen.
3. Anzeige *Maximale Kom.-Zeitscheibe Dauer pro Zyklus [ms]* auslesen.
4. Anzeige *Maximale Anzahl benötigter Kom.-Zeitscheibe Zyklen* auslesen.

Die Dauer der Kommunikationszeitscheibe ist so hoch einzustellen, dass der CPU-Zyklus die vom Prozess vorgegebene Watchdog-Zeit nicht überschreiten kann, wenn er die eingestellte Kommunikationszeitscheibe ausnutzt.



#### 5.2.4.5 Berechnung der *Max. Dauer Konfigurationsverbindungen [ms]* $t_{\text{Konfig}}$

Der Systemparameter *Max. Dauer Konfigurationsverbindungen [ms]* entspricht dem erforderlichen Zeitbudget  $t_{\text{Konfig}}$  für die systeminternen Kommunikationsverbindungen (Tasks):

- PADT Online Verbindungen (z. B. Download/Reload, BS-Update, Online-Test, Diagnose).
- Remote I/O Status-Verbindungen (Start, Stopp und Diagnose).
- Konfiguration von Modulen (z. B. Laden ausgetauschter Module).

Können diese Tasks nicht in einem CPU-Zyklus abgeschlossen werden, werden die verbleibenden Tasks im nächsten CPU-Zyklus abgearbeitet. Dadurch können unerwartete Verzögerungen für diese Tasks entstehen.

### i

HIMA empfiehlt  $t_{\text{Konfig}}$  so zu dimensionieren, dass alle Tasks in einem CPU-Zyklus abgearbeitet werden können.

Für die Betriebssysteme HIMax CPU ≤ V3 wird  $t_{\text{Konfig}}$  von SILworX mit 6 ms vorgegeben. Jedoch darf die Verarbeitungsdauer der genannten Tasks in einem CPU-Zyklus die Vorgabe überschreiten.

Für die Betriebssysteme HIMax CPU ≥ V4 wird  $t_{\text{Konfig}}$  wie folgt berechnet:

**X-CPU 01:**  $t_{\text{Konfig}} = (n_{\text{Com}} + n_{\text{PADT}} + n_{\text{RIO}}) * 0,25 \text{ ms} + 4 \text{ ms} + 4 * (t_{\text{Latenz}} * 2 + 0,31 \text{ ms})$

**X-CPU 31:**  $t_{\text{Konfig}} = (n_{\text{Com}} + n_{\text{PADT}}) * 1 \text{ ms} + n_{\text{RIO}} * 0,25 \text{ ms} + 4 \text{ ms} + 4 * (t_{\text{Latenz}} * 2 + 0,8 \text{ ms})$

$t_{\text{Konfig}}$ :	Systemparameter <i>Max. Dauer Konfigurationsverbindungen [ms]</i>
$n_{\text{COM}}$ :	Anzahl Module mit Ethernet-Schnittstellen (X-SB, X-CPU, X-COM)
$n_{\text{PADT}}$ :	5, maximale Anzahl PADT-Verbindungen
$n_{\text{RIO}}$ :	Anzahl konfigurierter Remote I/Os
$t_{\text{Latenz}}$ :	Aktive <i>maximale Systembus-Latenzzeit einsetzen, siehe nachfolgende Beschreibungen.</i> Wenn der Wert der maximalen Systembus-Latenzzeit in $\mu\text{s}$ angegeben ist, dann muss dieser vor der Berechnung durch 1000 dividiert werden, um den Wert in ms zu erhalten.

Je nachdem welche Systembusstruktur für das HIMax System gewählt wurde, muss für die Systembus-Latenzzeit folgender Wert eingesetzt werden:

Netzwerkstruktur:	Wenn für <b>den Parameter Maximale Systembus-Latenzzeit [<math>\mu\text{s}</math>]</b> ein Wert von 100 ... 50 000 $\mu\text{s}$ manuell eingetragen wurde, dann ist dieser Wert als $t_{\text{Latenz}}$ in die Formel einzusetzen.
Linierstruktur:	Wenn der Parameter <i>Maximale Systembus-Latenzzeit [<math>\mu\text{s}</math>]</i> auf <i>System-Standardwerte</i> eingestellt ist, dann ist der entsprechende Standardwert der maximalen Systembus-Latenzzeit für $t_{\text{Latenz}}$ aus der nachfolgenden Tabelle zu entnehmen und in die Formel einzusetzen. Alternativ zu dem Wert der Tabelle kann zunächst der mögliche Maximalwert eingesetzt werden für X-CPU 01 = 550,4 $\mu\text{s}$ und für X-CPU 31 = 1166,4 $\mu\text{s}$ .

Bei der Codegenerierung und bei der Projektkonvertierung wird im Logbuch des PADTs ein Hinweis ausgegeben, wenn  $t_{\text{Konfig}}$  kleiner ist, als nach obiger Formel errechnet.

**i**

Wenn  $t_{\text{Konfig}}$  zu klein eingestellt wurde, kann sich die Performance von PADT Online Verbindungen (Tasks) extrem verschlechtern und die Verbindung zu Remote I/Os abgebrochen werden.

HIMA empfiehlt den berechneten Wert  $t_{\text{Konfig}}$  mit dem im Control Panel angezeigten Wert zu vergleichen und gegebenenfalls in den Eigenschaften der Ressource zu korrigieren. Dies kann z. B. in einem SAT (Site Acceptance Test) durchgeführt werden.

Zu Testzwecken kann  $t_{\text{Konfig}}$  im Control Panel auch online eingestellt werden.

Der eingestellte Wert von  $t_{\text{Konfig}}$  muss für die Dimensionierung der erforderlichen Watchdog-Zeit berücksichtigt werden, siehe Kapitel *Sicherheitsrelevante Zeiten*.

Maximaler Rack-Abstand	Maximale Systembus-Latenzzeit in $\mu\text{s}$				Beispiele: System ist aufgebaut aus den genannten Racks
	X-CPU 01		X-CPU 31		
	Min	Max <sup>1)</sup>	Min	Max <sup>1)</sup>	
0	49,1	-	665,2	-	Nur Rack 0
1	105,5	155,5	721,6	771,6	Racks 0 und 1
2	161,9	211,9	778,0	828,0	Racks 0, 1, 3
3	218,4	268,4	834,4	884,4	Racks 0, 1, 3, 5
4	274,8	324,8	890,8	940,8	Racks 0, 1, 3,5, 7
5	331,2	381,2	947,2	997,2	Racks 0, 1, 3, 5, 7, 9
6	387,6	437,6	1003,6	1053,6	Racks 0, 1, 3, 5, 7, 9, 11
7	444,0	494,0	1060,9	1110,9	Racks 0, 1, 3, 5 ,7, 9, 11, 13,
8	500,4	550,4	1116,4	1166,4	Racks 1, 0, 2, 4, 6, 8, 10, 12, 14
1) Maximale Systembus-Latenzzeit einschließlich maximaler zusätzlicher Verzögerung durch die Netzwerk-Infrastruktur					

Tabelle 12: Standardwerte der maximalen Systembus-Latenzzeit

### 5.2.5 Systemvariablen eines Racks

Diese Systemvariablen dienen dazu, das Verhalten der Steuerung im laufenden Betrieb bei bestimmten Zuständen zu verändern. Diese Variablen sind einstellbar im Hardware-Editor von SILworX, in der Detailansicht des Racks, Register *System*.

Systemvariable	S <sup>1)</sup>	Funktion	Einstellung für sicheren Betrieb
Force-Deaktivierung	J	Verhindert das Starten des Forcen-Vorgangs und beendet einen laufenden Force-Vorgang. Die Standardeinstellung ist FALSE.	Applikations-spezifisch
Leer 0 ... Leer 16	J	Keine Funktion!	---
MultiForcen gesperrt	J	MultiForcen kann per Systemvariable MultiForcen gesperrt aktiviert und deaktiviert werden, so dass die damit verbundenen Funktionen vom Anwenderprogramm gesteuert werden können. Für globales MultiForcen muss die Systemvariable FALSE sein. Die Standardeinstellung ist FALSE.	Applikations-spezifisch
Notaus 1 ... Notaus 4	J	Schaltet die Steuerung in vom Anwenderprogramm erkannten Störfällen ab. Die Standardeinstellung ist FALSE.	Applikations-spezifisch
Read-only in RUN	J	Nach dem Starten der Steuerung sind die Zugriffsrechte auf die Zugriffsart <i>Lesen</i> herabgestuft. Ausnahmen sind Forcen und Reload. Die Standardeinstellung ist FALSE.	Applikations-spezifisch
Reload-Deaktivierung	J	Sperrt die Durchführung von Reload. Die Standardeinstellung ist FALSE.	Applikations-spezifisch
<sup>1)</sup> Systemparameter wird vom Betriebssystem sicherheitsbezogen behandelt, ja (J) oder nein (N).			

Tabelle 13: Systemvariablen des Racks

Diesen Systemvariablen lassen sich globale Variablen zuweisen, deren Wert durch einen physikalischen Eingang oder die Logik des Anwenderprogramms verändert wird.

#### 5.2.5.1 Eingangsvariablen zum Auslesen von Parametern

Die Eingangsvariablen können im Hardware-Editor von SILworX, in der Detailansicht des Racks, Register *System* ausgelesen und einer globalen Variable zugewiesen werden.

In der Spalte *Eingangsvariable* sind die Eingangsvariablen dadurch gekennzeichnet, dass ein Häkchen gesetzt ist.

Systemvariable	S <sup>1)</sup>	Beschreibung	Datentyp
Anzahl Feldfehler	N	Anzahl aktueller Feldfehler	UDINT
Anzahl Feldfehler historisch	N	Aufsummierte Anzahl Feldfehler (Zähler rücksetzbar)	UDINT
Anzahl Feldwarnungen	N	Anzahl aktueller Feldwarnungen	UDINT
Anzahl Feldwarnungen historisch	N	Aufsummierte Anzahl Feldwarnungen (Zähler rücksetzbar)	UDINT
Anzahl Kommunikationsfehler	N	Anzahl aktueller Kommunikationsfehler	UDINT
Anzahl Kommunikationsfehler historisch	N	Aufsummierte Anzahl Kommunikationsfehler (Zähler rücksetzbar)	UDINT
Anzahl Kommunikationswarnungen	N	Anzahl aktueller Kommunikationswarnungen	UDINT
Anzahl Kommunikationswarnungen historisch	N	Aufsummierte Anzahl Kommunikationswarnungen (Zähler rücksetzbar)	UDINT
Anzahl Systemfehler	N	Anzahl aktueller Systemfehler	UDINT

Systemvariable	S <sup>1)</sup>	Beschreibung		Datentyp
Anzahl Systemfehler historisch	N	Aufsummierte Anzahl Systemfehler (Zähler rücksetzbar)		UDINT
Anzahl Systemwarnungen	N	Anzahl aktueller Systemwarnungen		UDINT
Anzahl Systemwarnungen historisch	N	Aufsummierte Anzahl Systemwarnungen (Zähler rücksetzbar)		UDINT
Autostart	J	TRUE	Das Prozessormodul startet beim Anlegen der Versorgungsspannung das Anwenderprogramm automatisch.	BOOL
		FALSE	Das Prozessormodul geht beim Anlegen der Versorgungsspannung in den Zustand STOP.	
BS Major [1] ... [4] BS Minor [1] ... [4]	J	Version des Betriebssystems für jedes Prozessormodul. Die Anzahl redundanter Prozessormodule und die Werte sind abhängig vom Steuerungstyp.		UINT
CRC	J	Prüfsumme der Ressource-Konfiguration		UDINT
Datum/Uhrzeit [ms-Anteil] Datum/Uhrzeit [Sek.-Anteil]	N	Systemdatum und Systemzeit in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Force-Deaktivierung	J	TRUE	Forcen ist deaktiviert.	BOOL
		FALSE	Forcen ist möglich.	
Forcen aktiv	J	TRUE	Globales oder lokales Forcen ist aktiv.	BOOL
		FALSE	Globales und lokales Forcen sind nicht aktiv.	
Force-Schalterzustand	N	Information über gesetzte Force-Schalter:		UDINT
		0xFFFF FFFE	Kein Force-Schalter gesetzt.	
		0xFFFF FFFF	Mindestens ein Force-Schalter gesetzt.	
Globales Forcen gestartet	J	TRUE	Globales Forcen ist aktiv.	BOOL
		FALSE	Globales Forcen ist nicht aktiv.	
Leer 0 ... 16		Reserviert, <b>nicht</b> verwenden!		UDINT
Leer 17		Reserviert, <b>nicht</b> verwenden!		BOOL
Letzte Feldwarnung [ms] Letzte Feldwarnung [s]	N	Datum und Uhrzeit der letzten Feldwarnung in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Letzte Kommunikationswarnung [ms] Letzte Kommunikationswarnung [s]	N	Datum und Uhrzeit der letzten Kommunikationswarnung in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Letzte Systemwarnung [ms] Letzte Systemwarnung [s]	N	Datum und Uhrzeit der letzten Systemwarnung in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Letzter Feldfehler [ms] Letzter Feldfehler [s]	N	Datum und Uhrzeit des letzten Feldfehlers in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Letzter Kommunikationsfehler [ms] Letzter Kommunikationsfehler [s]	N	Datum und Uhrzeit des letzten Kommunikationsfehlers in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Letzter Systemfehler [ms] Letzter Systemfehler [s]	N	Datum und Uhrzeit des letzten Systemfehlers in Millisekunden und Sekunden seit 01.01.1970: 0 ... 999 ms, 0 ... 4 294 967 295 s.		UDINT
Lüfterzustand	N	Abhängig vom Steuerungstyp, siehe Dokumentation. 0xFF = Nicht vorhanden		BYTE

Systemvariable	S <sup>1)</sup>	Beschreibung		Datentyp
Mono Startup Release	J	Freigabe für nicht-redundanten Betrieb.		BOOL
		TRUE	Ein einzelnes Prozessormodul in Rack 0, Steckplatz 3 darf mit nur einem Systembus starten.	
		FALSE	Auch für ein einzelnes Prozessormodul sind beide Systembusse erforderlich.	
MultiForcen gesperrt	J	TRUE	MultiForcen ist gesperrt.	BOOL
		FALSE	MultiForcen ist möglich.	
Online-Einstellungen erlauben	J	Haupt-Freigabeschalter des Prozessormoduls.		BOOL
		TRUE	Die untergeordneten Freigabeschalter können verändert werden.	
		FALSE	Die untergeordneten Freigabeschalter können nicht verändert werden.	
Read-only in RUN	J	TRUE	Die Bedienfunktionen Stopp, Start, Download sind gesperrt.	BOOL
		FALSE	Die Bedienfunktionen Stopp, Start, Download sind nicht gesperrt.	
Redundanz-Info	J	Bitcodierter Redundanzzustand der Prozessormodule.		
		Bit Nr.	Prozessormodule	
		0	X-CPU 01 in Steckplatz 3 oder X-CPU 31 in Steckplatz 1	
		1	X-CPU 01 in Steckplatz 4 oder X-CPU 31 in Steckplatz 2	
		2	X-CPU 01 in Steckplatz 5 oder X-CPU 01 in Steckplatz 4 (Rack 1)	
		3	X-CPU 01 in Steckplatz 6 oder X-CPU 01 in Steckplatz 3 (Rack 1)	
		4 ... 31	Nicht belegt!	
		Bit = 0: Prozessormodul nicht in Redundanz Bit = 1: Prozessormodul in Redundanz Alle übrigen Bits haben den Wert = 0.		
Reload erlaubt	J	TRUE	Laden der Steuerung mittels Reload ist möglich.	BOOL
		FALSE	Laden der Steuerung mittels Reload ist gesperrt.	
Reload-Deaktivierung	J	TRUE	Laden mittels Reload ist gesperrt.	BOOL
		FALSE	Laden mittels Reload ist möglich.	
Reload-Zyklus	J	TRUE	Der aktuelle Zyklus ist der erste Zyklus nach einem Reload.	BOOL
		FALSE	Sonst.	

Systemvariable	S <sup>1)</sup>	Beschreibung		Datentyp
Responsible Module Essential	J	Essential-Zustand redundanter Prozessormodule. Die Variable existiert abhängig von der Steuerungsfamilie.		BYTE
		Bit Nr.	Zustand bei Bit = 1	
		0	Kein Systembusmodul oder X-CPU 31 mit Status <i>Responsible</i> ist Essential d. h., Bus A und B funktionieren. Eines der Module darf entfernt werden.	
		1	Das Systembusmodul oder X-CPU 31 mit Status <i>Responsible</i> für den Systembus A ist Essential. Das Systembusmodul darf nicht entfernt werden, da Systembus B nicht aktiv.	
		2	Das Systembusmodul oder X-CPU 31 mit Status <i>Responsible</i> für den Systembus B ist Essential. Das Systembusmodul darf nicht entfernt werden, da Systembus A nicht aktiv.	
		3	Gilt nur für X-CPU 31: Die Organisation des Systembetriebs ist gestört und eine weitere Diagnose des Problem ist nicht möglich. Nach ca. 10 Minuten in diesem Zustand geht HIMax zu einem der Zustände 1 oder 2 über.	
		Falls einer der Zustände 1, 2 oder 3 einen transienten Fehler als Ursache hatte, kehrt HIMax nach Zustand 0 zurück, wenn dieser Fehler nicht mehr ansteht.		
Sicherheitszeit [ms]	J	Für die Ressource eingestellte Sicherheitszeit in ms.		UDINT
Start erlaubt	J	TRUE	Das Starten des Prozessormoduls durch das PADT ist erlaubt.	BOOL
		FALSE	Das Starten des Prozessormoduls durch das PADT ist nicht erlaubt.	
Start Zyklus	J	TRUE	Der aktuelle Zyklus ist der erste Zyklus nach dem Start.	BOOL
		FALSE	Sonst.	
Stromversorgungszustand [1] ... [4]	N	Bitcodierter Zustand der Spannungsversorgung der Prozessormodule 1 ... 4.		BYTE
		Bit Nr.	Zustand bei gesetztem Bit	
		0	Versorgungsspannung L1 fehlerhaft	
		1	Versorgungsspannung L2 fehlerhaft	
		2	Über-/Unterspannung bei intern erzeugter Spannung	
		3	Ungültige Abgleichdaten der intern erzeugten Spannungen	
		4	Spannungsquelle für VCC hat Unterspannung	
5	Spannungsquelle für VCC hat Unterspannung			
System-ID [SRS]	J	System-ID der Steuerung, 1 ... 65 535.		UINT
Systemtick HIGH Systemtick LOW	J	Umlaufender Millisekundenzähler (64 bit).		UDINT
Temperaturzustand [1] ... [4]	N	Bitcodierter Temperaturzustand der Prozessormodule 1 ... 4:		BYTE
		Bit Nr.	Zustand bei Bit = 1	
		0	Temperaturschwelle 1 überschritten.	
		1	Temperaturschwelle 2 überschritten.	
		2	Fehlerhafter Temperaturwert.	
		Der Wert ist unabhängig von der Einstellung der Temperaturüberwachung für das zugehörige Rack.		

Systemvariable	S <sup>1)</sup>	Beschreibung	Datentyp
Verbleibende globale Force-Dauer [ms]	J	Zeit in Millisekunden bis zum Ablauf der globalen Force-Zeitbegrenzung.	DINT
Watchdog-Zeit [ms]	J	Höchste zulässige Dauer eines RUN-Zyklus in Millisekunden	UDINT
Zykluszeit, letzte [ms]	J	Aktuelle Zykluszeit.	UDINT
Zykluszeit, max [ms]	N	Maximale Zykluszeit in Millisekunden.	UDINT
Zykluszeit, min [ms]	N	Minimale Zykluszeit in Millisekunden.	UDINT
Zykluszeit, mittlere [ms]	N	Mittlere Zykluszeit in Millisekunden.	UDINT
<sup>1)</sup> Systemvariable wird vom Betriebssystem sicherheitsbezogen behandelt, ja (J) oder nein (N).			

Tabelle 14: Eingangsvariablen

Die folgenden Systemvariablen aus der Tabelle 14 sind Felder, deren Index die Nummer des Prozessormoduls ist:

- BS Major, BS Minor.
- Redundanz-Info (Bit-Leiste).
- Stromversorgungszustand.
- Temperaturzustand.

Der in diesen Feldern verwendete Index des Prozessormoduls X-CPU 01 wird auf folgende Weise auf die Steckplätze der Prozessormodule in den Racks abgebildet:

1. Im Rack 0 ist der Index ab Steckplatz 3 aufsteigend zu zählen.
2. Im Rack 1 ist der Index bis zum Steckplatz 3 abfallend zu zählen.

Daher gilt die folgende Zuordnung:

	Steckplätze			
	3	4	5	6
Rack 1	4	3		
Rack 0	1	2	3	4

Tabelle 15: Zuordnung des Index zu Steckplätzen der Prozessormodule X-CPU 01

Die Prozessormodule X-CPU 01 mit den Indizes 3 und 4 dürfen sich entweder im Rack 0 oder im Rack 1 befinden!

Für Prozessormodule X-CPU 31 gilt:

- Das Prozessormodul auf Steckplatz 1 hat den Index 1.
- Das Prozessormodul auf Steckplatz 2 hat den Index 2.

### 5.2.5.2 Abschließen und Aufschließen der Ressource

**Abschließen** der Steuerung bedeutet das Verriegeln von Eingriffsmöglichkeiten des Anwenders während des Betriebs. Eine unbefugte Manipulation des Anwenderprogramms wird damit verhindert.

**Aufschließen** der Steuerung bedeutet das Entfernen der aktiven Verriegelung, zum Beispiel zur Durchführung von Maßnahmen an der Steuerung.

Zum Verriegeln dienen die Systemvariablen *Read only in RUN*, *Reload-Deaktivierung*, *Force-Deaktivierung* und *MultiForcen gesperrt*, siehe Tabelle 13.

Wenn alle der oben genannten Systemvariablen TRUE sind, dann ist kein Zugriff auf die Steuerung mehr möglich. In diesem Fall kann die Steuerung nur durch Neustart aller Prozessormodule mit dem Mode-Schalter in Stellung *Init* wieder in den Zustand STOP versetzt werden. Dann ist ein Neuladen eines Anwenderprogramms möglich. Das Beispiel beschreibt den einfachen Fall, dass mit einem Schlüsselschalter alle Eingriffe in die Ressource gesperrt oder zugelassen werden.

#### Steuerung abschließbar machen

1. Globale Variablen vom Typ BOOL definieren, Initialwerte auf FALSE setzen.
  2. Globale Variable den oben genannten Systemvariablen als Ausgangsvariable zuweisen.
  3. Globale Variablen dem Kanalwert eines digitalen Eingangs zuweisen.
  4. Schlüsselschalter an den digitalen Eingang anschließen.
  5. Programm kompilieren, auf die Steuerung laden und starten.
- Der Besitzer eines passenden Schlüsselschalters kann die Steuerung ab- und aufschließen. Bei einem Fehler im entsprechenden digitalen Eingangsmodul ist die Steuerung aufgeschlossen.

Dieses einfache Beispiel lässt sich durch die Verwendung von mehreren globalen Variablen, digitalen Eingängen und Schlüsselschaltern abwandeln. Damit können Berechtigungen für Forcen, MultiForcen, Reload und weiteren Bedienfunktionen auf unterschiedliche Schlüssel und Personen verteilt werden.



### 5.2.6 Systemparameter der Anwenderprogramme

Die folgenden Parameter von Anwenderprogrammen lassen sich im Dialogfenster *Eigenschaften* des Anwenderprogramms einstellen:

Systemparameter	S <sup>1)</sup>	Beschreibung	Einstellung für sicheren Betrieb
Name	N	Name des Anwenderprogramms. Der Name muss innerhalb der Ressource eindeutig sein.	Beliebig
Programm ID	J	ID für die Identifizierung des Programms bei der Anzeige in SILworX. Wertebereich: 0 ... 4 294 967 295 Standardwert: 0 Bei Einstellung von <i>Codegenerierung Kompatibilität</i> auf <i>SILworX V2</i> ist nur der Wert 1 zulässig.	Applikations-spezifisch
Priorität	J	Priorität des Anwenderprogramms. Wertebereich: 0 ... 31 Standardwert: 0 (maximale Priorität) Einstellung nur nötig bei Verwendung mehrerer APs!	Applikations-spezifisch
Maximale CPU-Zyklen Programm	J	Maximale Anzahl an CPU-Zyklen, die ein Zyklus des Anwenderprogramms dauern darf. Wertebereich: 1 ... 4 294 967 295 Standardwert: 1 Einstellung nur nötig bei Verwendung mehrerer APs!	Applikations-spezifisch
Max. Dauer pro Zyklus [µs]	N	Maximale Ausführungsdauer pro Zyklus des Prozessormoduls für ein Anwenderprogramm. Wertebereich: 0 ... 4 294 967 295 Standardwert: 0 (keine Begrenzung) Die sicherheitsbezogene Reaktion wird über den Watchdog gewährleistet. Einstellung nur nötig bei Verwendung mehrerer APs!	Applikations-spezifisch
Watchdog-Zeit [ms] (berechnet)	---	Überwachungszeit des Anwenderprogramms, berechnet aus dem Produkt der Watchdog-Zeit der Ressource und der parametrisierten maximaler Anzahl von CPU-Zyklen. Nicht änderbar!	
Klassifikation	N	Einstufung des Anwenderprogramms in <i>sicherheitsgerichtet</i> oder <i>standard</i> , dient nur zur Dokumentation und hat keinen Einfluss auf die Funktion des Programms. Die Standardeinstellung ist sicherheitsgerichtet	Applikations-spezifisch
Online-Einstellungen erlauben	J	Wenn <i>Online-Einstellungen erlauben</i> ausgeschaltet ist, können die Einstellungen der anderen Programmschalter nicht per Online-Zugriff (Control Panel) verändert werden. Wirkt nur, wenn <i>Online-Einstellungen erlauben</i> der Ressource TRUE ist! Standardwert: TRUE.	
Autostart	J	Freigegebene Art des Autostarts: Kaltstart, Warmstart, Aus. Die Standardeinstellung ist Warmstart.	Applikations-spezifisch
Start erlaubt	J	TRUE: Start des Anwenderprogramms durch das PADT erlaubt. Standardwert: TRUE.	Applikations-spezifisch
		FALSE: Start des Anwenderprogramms durch das PADT nicht erlaubt.	

Systemparameter	S <sup>1)</sup>	Beschreibung		Einstellung für sicheren Betrieb
Testmodus erlaubt	J	TRUE:	Testmodus für das Anwenderprogramm ist erlaubt.	Applikations-spezifisch <sup>2)</sup>
		FALSE:	Testmodus für das Anwenderprogramm ist nicht erlaubt. Standardwert: FALSE.	
Reload erlaubt	J	TRUE:	Reload des Anwenderprogramms ist erlaubt. Standardwert: TRUE.	Applikations-spezifisch
		FALSE:	Reload des Anwenderprogramms ist nicht erlaubt.	
		Einstellungen in den Ressource-Eigenschaften beachten!		
Lokales Forcen erlaubt	J	TRUE:	Forcen auf Programmebene erlaubt.	FALSE empfohlen
		FALSE:	Forcen auf Programmebene nicht erlaubt. Standardwert: FALSE.	
Lokale Force-Timeout-Reaktion	J	Verhalten des Anwenderprogramms nach Ablauf der Force-Zeit: <ul style="list-style-type: none"><li>Nur Forcen beenden.</li><li>Programm stoppen.</li></ul> Die Standardeinstellung ist <i>Nur Forcen beenden</i> .		
Codegenerierung Kompatibilität	-	Die Codegenerierung arbeitet kompatibel zu früheren Versionen von SILworX.		Applikations-spezifisch
		SILworX V2	Codegenerierung arbeitet kompatibel zu SILworX V2.	
		SILworX V3	Codegenerierung arbeitet kompatibel zu SILworX V3.	
		SILworX V4 – V6b	Codegenerierung arbeitet kompatibel zu SILworX V4 bis SILworX V6b.	
		ab SILworX V7	Codegenerierung arbeitet kompatibel zu SILworX V7.	
		Die Standardeinstellung ist bei allen neuen Projekten: <i>ab SILworX V7</i> .		

1) Systemparameter wird vom Betriebssystem sicherheitsbezogen behandelt, ja (J) oder nein (N)

2) Nach Ende des Testbetriebs muss ein Kaltstart des Programms durchgeführt werden, bevor ein sicherheitsbezogen Betrieb aufgenommen wird!

Tabelle 16: Systemparameter des Anwenderprogramms

### 5.2.7 Hinweise zum Parameter *Codegenerierung Kompatibilität*

Für den Parameter *Codegenerierung Kompatibilität* folgende Punkte beachten:

- Bei einem neu angelegten Projekt wählt SILworX die aktuellste Einstellung für *Codegenerierung Kompatibilität* aus. Damit werden die aktuellen, optimierten Einstellungen aktiviert und die aktuellsten Versionen von Modulen und Betriebssystemen unterstützt. Prüfen Sie, ob diese Einstellung zur verwendeten Hardware passt!
- Bei einem älteren Projekt, das in die aktuelle SILworX Version konvertiert wurde, bleibt die ursprünglichen *Codegenerierung Kompatibilität* erhalten. Dadurch ändert sich bei der Codegenerierung der Konfigurations-CRC gegenüber der Vorversion nicht, und die Konfiguration bleibt kompatibel zu den Betriebssystemen der Module. Bei konvertierten Projekten muss die *Codegenerierung Kompatibilität* nur dann geändert werden, wenn Sie zusätzliche Funktionen einer Steuerung nutzen möchten.
- Wenn in der Eigenschaft der Ressource eine *Minimale Konfigurationsversion* von *SILworX V4* oder höher eingestellt ist, dann muss in jedem Anwenderprogramm der Parameter *Codegenerierung Kompatibilität* auf *ab SILworX V7* eingestellt werden.

### 5.2.8 Die lokalen Systemvariablen des Anwenderprogramms

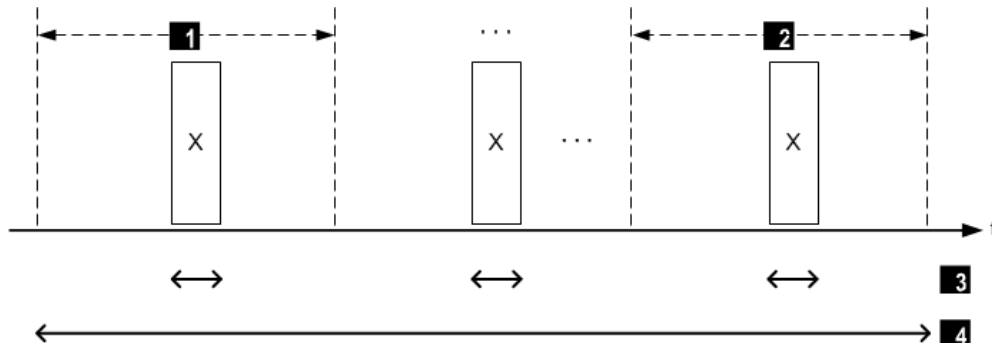
Die lokalen Systemvariablen liefern zur Laufzeit Informationen zu den Betriebsbedingungen des Anwenderprogramms. Nicht alle lokalen Systemvariablen dürfen für die Programmierung sicherheitsbezogener Reaktionen verwendet werden.

Variable	S <sup>1)</sup>	Beschreibung	Datentyp
Program_CRC	X	Prüfsumme über das Anwenderprogramm, durch die Verfälschungen erkennbar werden.	LWORD
Program_CycleDuration		Dauer aller Zyklen des Prozessormoduls, die für die Ausführung eines Anwenderprogramm-Zyklus benötigt wurden, in µs, siehe Bild 13. Gemessen wurde die Dauer des vorherigen Anwenderprogramm-Zyklus. Im ersten Zyklus nach dem Start des Anwenderprogramms hat <i>Program_CycleDuration</i> den Wert 0.	UDINT
Program_ExecutionCycles		Anzahl Zyklen des Prozessormoduls, die für die vollständige Ausführung eines Zyklus des Anwenderprogramms benötigt werden. Gemessen wurde die Anzahl Prozessormodul-Zyklen des vorherigen Anwenderprogramm-Zyklus.	UDINT
Program_ExecutionDuration		Verarbeitungszeit eines Zyklus des Anwenderprogramms in µs, siehe Bild 13. Gemessen wurde die Verarbeitungszeit des vorherigen Zyklus. Im ersten Zyklus nach dem Start des Anwenderprogramms hat <i>Program_ExecutionDuration</i> den Wert 0.	UDINT
Program_ForceSwitch	X	TRUE Die Bedingungen für das lokale Forcen sind erfüllt. FALSE Die Bedingungen für das lokale Forcen sind nicht erfüllt.	BOOL
Program_ID		Vom Anwender vergebene Identifikation des Programms. Die <i>Program_ID</i> identifiziert das Anwenderprogramm in Anzeigefunktionen wie im Control Panel.	UDINT
Program_ReloadCycle	X	TRUE Im ersten Zyklus nach einem Reload Ausnahme: Der Reload ändert nur Parameter des Anwenderprogramms FALSE In allen anderen Zyklen, auch bei Reload, der nur Parameter von Anwenderprogrammen ändert	BOOL
Program_StartCycle	X	TRUE Im ersten Zyklus nach dem Start des Anwenderprogramms FALSE In allen anderen Zyklen	BOOL
<sup>1)</sup> Nur die in Spalte S markierten Systemvariablen sind für Sicherheitsfunktionen verwendbar! Die anderen Systemvariablen dürfen <b>nicht</b> zur Programmierung von Sicherheitsfunktionen verwendet werden!			

Tabelle 17: Die lokalen Systemvariablen des Anwenderprogramms

### *Program\_CycleDuration* und *Program\_ExecutionDuration*

Das folgende Bild zeigt den Ablauf eines Zyklus des Anwenderprogramms X, der mehrere Zyklen des Prozessormoduls dauert. Der erste betrachtete Zyklus des Prozessormoduls enthält den Anfang des Anwenderprogramm-Zyklus, der letzte betrachtete Zyklus des Prozessormoduls das Ende.



- 1** Erster betrachteter Zyklus des Prozessormoduls
- 2** Letzter betrachteter Zyklus des Prozessormoduls

- 3** *Program\_ExecutionDuration*
- 4** *Program\_CycleDuration*

Bild 13: *Program\_CycleDuration* und *Program\_ExecutionDuration*

*Program\_CycleDuration* ist die Dauer aller Zyklen des Prozessormoduls für einen Zyklus des Anwenderprogramms, entspricht im Bild 13 dem großen Doppelpfeil **4**.

*Program\_ExecutionDuration* ist der Anteil des Anwenderprogramms X an der Verarbeitungszeit in allen betrachteten Zyklen des Prozessormoduls. *Program\_ExecutionDuration* entspricht im Bild 13 der Summe aller kleinen Doppelpfeile **3**.

*Program\_ExecutionCycles* ist die Anzahl der Zyklen vom ersten **1** bis zum letzten **2** betrachteten Zyklus des Prozessormoduls.

## 5.2.9 Zuordnung zu E/A-Kanälen

Im Hardware-Editor von SILworX ist es möglich, eine globale Variable einem E/A-Kanal zu zuweisen. Dazu lässt sich in der Detailansicht eines E/A-Moduls die globale Variable per Drag&Drop aus der Objektauswahl in die Kanalliste des E/A-Moduls ziehen.

Dadurch lassen sich der Wert und die Statusinformationen des Kanals im Anwenderprogramm verfügbar machen.

### 5.2.9.1 Verwendung digitaler Eingänge

**Folgende Schritte sind notwendig, um den Wert eines digitalen Eingangs im Anwenderprogramm zu verwenden**

1. Eine globale Variable vom Typ BOOL definieren.
2. Bei der Definition als Initialwert den sicheren Wert angeben.
3. Die globale Variable dem Kanalwert des Eingangs zuweisen.
- Die globale Variable liefert einen sicheren Wert ins Anwenderprogramm.

Für digitale Initiator-Eingangsmodule, die intern analog arbeiten, ist es auch möglich, den Rohwert zu benutzen und im Anwenderprogramm den Wert zu berechnen. Näheres siehe unten.

Durch Zuweisen globaler Variable auf *Kanal OK* und auf weitere Diagnosestatus bieten sich zusätzliche Möglichkeiten, die externe Beschaltung zu diagnostizieren und Fehlerreaktionen im

Anwenderprogramm zu programmieren. Weitere Einzelheiten zu den Diagnosestatus wie Leitungsschluss und Leitungsbruch finden Sie in den Modulhandbüchern.

#### 5.2.9.2 Verwendung analoger Eingänge

Analoge Eingangskanäle wandeln die gemessenen Eingangsströme in einen Wert vom Typ DINT (double integer) um. Dieser Wert steht dann dem Anwenderprogramm als *Rohwert* zur Verfügung. Dabei entspricht 1 mA einem Wert von 10 000, der Wertebereich ist 0 ... 240 000.

In vielen Fällen ist es einfacher, anstelle des Rohwertes den *Prozesswert* vom Datentyp REAL zu verwenden. Diesen berechnet HIMax aus dem Rohwert und der Skalierung auf 4 und 20 mA. Näheres siehe Handbuch des Moduls.

Es gibt zwei Möglichkeiten, die Werte analoger Eingänge im Anwenderprogramm zu verwenden:

- Verwendung des Prozesswerts  
Der Prozesswert eines analogen Eingangs liefert den Wert einschließlich der sicheren Fehlerreaktion, sofern er korrekt konfiguriert ist.
- Verwendung des Rohwerts  
Der Rohwert ist der Messwert ohne die sichere Fehlerreaktion. Es ist notwendig, diese projektspezifisch zu programmieren.

**Folgende Schritte sind notwendig, um den Prozesswert zu verwenden:**

1. Globale Variable vom Typ REAL definieren.
  2. Bei der Definition als Initialwert den sicheren Wert angeben.
  3. Die globale Variable dem Prozesswert des Eingangs zuweisen.
  4. Den Messbereich des Kanals durch Angabe je eines REAL-Werts für 4 mA und für 20 mA festlegen.
- Die globale Variable liefert einen sicheren Wert ins Anwenderprogramm.

**Folgende Schritte sind notwendig, um den Rohwert zu verwenden:**

1. Globale Variable vom Typ DINT definieren.
  2. Globale Variable von einem im Anwenderprogramm benötigten Typ definieren.
  3. Im Anwenderprogramm eine geeignete Umrechnungsfunktion programmieren, um den Rohwert in einen dort verwendeten Typ umzuwandeln. Dabei den Messbereich des Datentyps berücksichtigen.
  4. Im Anwenderprogramm eine sicherheitsbezogene Fehlerreaktion unter Verwendung der Status *Kanal OK*, *LS*, *LB* (ggfs. weiterer) programmieren.
- Das Anwenderprogramm kann den Messwert sicher verarbeiten.

Wenn bei einem Kanal der Wert 0 im gültigen Messbereich liegt, muss das Anwenderprogramm zusätzlich zum Prozesswert mindestens den Parameter *Kanal OK* auswerten.

Durch Zuweisen globaler Variable auf *Kanal OK*, *Submodul OK*, *Modul OK* und auf weitere Diagnosestatus bieten sich zusätzliche Möglichkeiten, die externe Beschaltung zu diagnostizieren und Fehlerreaktionen im Anwenderprogramm zu programmieren. Weitere Einzelheiten zu den Diagnosestatus wie Leitungsschluss und Leitungsbruch finden Sie in den Modulhandbüchern.

#### 5.2.9.3 Verwendung sicherheitsbezogener Zählereingänge

Es ist möglich, den Zählerstand oder die Drehzahl/Frequenz als ganzzahligen Wert oder als skalierten Gleitkommawert zu verwenden.

**Folgende Schritte sind notwendig, um den ganzzahligen Wert zu verwenden:**

1. Globale Variable vom Typ UDINT definieren.
2. Bei der Definition als Initialwert den sicheren Wert angeben.

3. Die globale Variable dem ganzzahligen Wert des Eingangs zuweisen.
- Die globale Variable liefert einen sicheren Wert ins Anwenderprogramm.

**Folgende Schritte sind notwendig, um den skalierten Gleitkommawert zu verwenden:**

1. Globale Variable vom Typ REAL definieren.
2. Bei der Definition als Initialwert den sicheren Wert angeben.
3. Die globale Variable dem skalierten Gleitkommawert des Eingangs zuweisen.
4. Den Skalierungswert des Kanals durch Angabe eines REAL-Werts festlegen.
- Die globale Variable liefert einen sicheren Wert ins Anwenderprogramm.

#### 5.2.9.4 Verwendung digitaler Ausgänge

**Folgende Schritte sind notwendig, um einen Wert im Anwenderprogramm auf einen digitalen Ausgang zu schreiben:**

1. Eine globale Variable vom Typ BOOL definieren.
2. Bei der Definition als Initialwert den sicheren Wert angeben.
3. Die globale Variable dem Kanalwert des Ausgangs zuweisen.
- Die globale Variable liefert einen sicheren Wert auf den digitalen Ausgang.

Durch Zuweisen globaler Variable auf *Kanal OK* und auf weitere Diagnosestatus bieten sich zusätzliche Möglichkeiten, die externe Beschaltung zu diagnostizieren und Fehlerreaktionen im Anwenderprogramm zu programmieren. Weitere Einzelheiten zu den Diagnosestatus wie Leitungsschluss und Leitungsbruch finden Sie in den Modulhandbüchern.

#### 5.2.9.5 Verwendung analoger Ausgänge

**Folgende Schritte sind notwendig, um einen Wert im Anwenderprogramm auf einen analogen Ausgang zu schreiben:**

1. Eine globale Variable vom Typ REAL definieren.
2. Bei der Definition als Initialwert den sicheren Wert angeben.
3. Globale Variable dem Kanalwert des Ausgangs zuweisen.
4. Bei den Parametern *4 mA* und *20 mA* des Ausgangskanals die zugehörigen REAL-Werte entsprechend dem genutzten Bereich der globalen Variable angeben.
- Die globale Variable liefert einen sicheren Wert auf den analogen Ausgang.

---

**i**

Bei nicht (mehr) genutzten Ausgangskanälen müssen die Parameter *4 mA* und *20 mA* auf die Standardeinstellungen 4.0 und 20.0 eingestellt sein.

---

Durch Zuweisen globaler Variable auf *Kanal OK* und auf weitere Diagnosestatus bieten sich zusätzliche Möglichkeiten, die externe Beschaltung zu diagnostizieren und Fehlerreaktionen im Anwenderprogramm zu programmieren. Weitere Einzelheiten zu den Diagnosestatus wie Leitungsschluss und Leitungsbruch finden Sie in den Modulhandbüchern.

#### 5.2.10 Zuordnung zu Kommunikationsverbindungen

Die Werte globaler Variable lassen sich über Kommunikationsverbindungen versenden oder von diesen empfangen. Dazu den Editor des verwendeten Kommunikationsprotokolls öffnen und die globale Variable per Drag&Drop aus der Objektauswahl in den Arbeitsbereich ziehen.

Weitere Einzelheiten zu Kommunikationsprotokollen finden Sie im Kommunikationshandbuch HI 801 100 D und die Bedienung der Editoren für die Kommunikationsprotokolle finden Sie in der SILworX Online-Hilfe.

### 5.2.11 Ereignisaufzeichnung konfigurieren

Der Alarm&Events-Editor beinhaltet folgende Register:

- Event-Definition BOOL.
- Event-Definition Skalar.
- Eigenschaften.

#### Definition von Ereignissen

1. Für jedes Ereignis eine globale Variable definieren. In der Regel globale Variablen verwenden, die bereits für das Programm definiert sind.
  2. Unter der Ressource einen neuen Unterzweig **Alarm&Events** erzeugen, falls dieser noch nicht existiert.
  3. Im Alarm&Events-Editor Ereignisse definieren
    - Globale Variablen ins Ereignisfenster für boolesche oder skalare Ereignisse ziehen.
    - Die Einzelheiten der Ereignisse festlegen, siehe nachfolgende Tabellen.
- Ereignisse sind definiert.

Weitere Informationen finden Sie im Handbuch X-OPC-Server HI 801 479 D.

### 5.2.11.1 Register **Event-Definition BOOL**

Für jedes der booleschen Ereignisse sind die folgenden Parameter im Alarm&Events-Editor der Ressource einzustellen:

Parameter	Beschreibung								
Name	Name des Ereignisses, muss in der Ressource eindeutig sein. Text: maximal 63 Zeichen.								
Globale Variable	Name der zugewiesenen globalen Variable (eingefügt z. B. durch Drag&Drop).								
Datentyp	Datentyp der zugewiesenen globalen Variable, nicht änderbar.								
Event-Quelle	<p>Quelle des Ereignisses:</p> <table border="1"> <thead> <tr> <th>Quelle</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>CPU</td><td>Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch.</td></tr> <tr> <td>E/A</td><td>Ein geeignetes E/A-Modul bildet den Zeitstempel.</td></tr> <tr> <td>Auto</td><td>Falls ein geeignetes E/A-Modul zugewiesen ist, bildet dieses den Zeitstempel, falls nicht, das Prozessormodul.</td></tr> </tbody> </table> <p>Bei E/A-Events wird der Name zusätzlich mit der Position als Source im OPC Client angezeigt, z. B. «Name_0_10_3». Standardeinstellung: Auto</p>	Quelle	Beschreibung	CPU	Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch.	E/A	Ein geeignetes E/A-Modul bildet den Zeitstempel.	Auto	Falls ein geeignetes E/A-Modul zugewiesen ist, bildet dieses den Zeitstempel, falls nicht, das Prozessormodul.
Quelle	Beschreibung								
CPU	Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch.								
E/A	Ein geeignetes E/A-Modul bildet den Zeitstempel.								
Auto	Falls ein geeignetes E/A-Modul zugewiesen ist, bildet dieses den Zeitstempel, falls nicht, das Prozessormodul.								
Alarm bei FALSE	<p>Aktiviert: Die Wertänderung TRUE-&gt;FALSE der globalen Variablen löst ein Ereignis aus.</p> <p>Deaktiviert: Die Wertänderung FALSE-&gt;TRUE der globalen Variablen löst ein Ereignis aus</p> <p>Standardeinstellung: Deaktiviert</p>								
Text für Alarm	Text, der den Alarmzustand benennt.								
Priorität für Alarm	<p>Priorität des Alarmzustands.</p> <p>Je höher der eingestellte Wert, umso niedriger die Priorität.</p> <p>Wertebereich: 0 ... 1000</p> <p>Standardwert: 500</p>								
Bestätigung für Alarm erforderlich	<p>Bestätigung des Alarms durch den Bediener (Quittierung).</p> <p>Aktiviert: Bestätigung erforderlich.</p> <p>Deaktiviert: Bestätigung nicht erforderlich.</p> <p>Standardeinstellung: Deaktiviert</p>								
Text für Rückkehr zu Normalzustand	Text, der den Normalzustand benennt.								
Priorität für Rückkehr zu Normalzustand	<p>Priorität für die Rückkehr in den Normalzustand.</p> <p>Je höher der eingestellte Wert, umso niedriger ist die Priorität.</p> <p>Wertebereich: 0 ... 1000</p> <p>Standardwert: 500</p>								
Bestätigung für Rückkehr zu Normalzustand erforderlich	<p>Bestätigung der Rückkehr in den Normalzustand durch den Bediener (Quittierung)</p> <p>Aktiviert: Bestätigung erforderlich.</p> <p>Deaktiviert: Bestätigung nicht erforderlich.</p> <p>Standardeinstellung: Deaktiviert</p>								

Tabelle 18: Parameter für boolesche Ereignisse



### 5.2.11.2 Register **Event-Definition Skalar**

Für jedes der skalaren Ereignisse sind die folgenden Parameter im Alarm&Events-Editor der Ressource einzustellen:

Parameter	Beschreibung								
Name	Name des Ereignisses, muss in der Ressource eindeutig sein. Text: maximal 63 Zeichen.								
Globale Variable	Name der zugewiesenen globalen Variable (Eingefügt z. B. durch Drag&Drop).								
Datentyp	Datentyp der zugewiesenen globalen Variable, nicht änderbar.								
Event-Quelle	<p>Quelle des Ereignisses:</p> <table border="1"> <thead> <tr> <th>Quelle</th><th>Beschreibung</th></tr> </thead> <tbody> <tr> <td>CPU</td><td>Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch.</td></tr> <tr> <td>E/A</td><td>Ein geeignetes E/A-Modul bildet den Zeitstempel.</td></tr> <tr> <td>Auto</td><td>Falls ein geeignetes E/A-Modul zugewiesen ist, bildet dieses den Zeitstempel, falls nicht, das Prozessormodul.</td></tr> </tbody> </table> <p>Standardeinstellung: Auto</p>	Quelle	Beschreibung	CPU	Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch.	E/A	Ein geeignetes E/A-Modul bildet den Zeitstempel.	Auto	Falls ein geeignetes E/A-Modul zugewiesen ist, bildet dieses den Zeitstempel, falls nicht, das Prozessormodul.
Quelle	Beschreibung								
CPU	Das Prozessormodul bildet den Zeitstempel. Es führt die Ereignisbildung komplett in jedem seiner Zyklen durch.								
E/A	Ein geeignetes E/A-Modul bildet den Zeitstempel.								
Auto	Falls ein geeignetes E/A-Modul zugewiesen ist, bildet dieses den Zeitstempel, falls nicht, das Prozessormodul.								
Alarm bei FALSE	<p>Aktiviert: Die Wertänderung TRUE-&gt;FALSE der globalen Variablen löst ein Ereignis aus.</p> <p>Deaktiviert: Die Wertänderung FALSE-&gt;TRUE der globalen Variablen löst ein Ereignis aus</p> <p>Standardeinstellung: Deaktiviert</p>								
Text für HH-Alarm	Text, der den Alarmzustand des obersten Grenzwerts benennt.								
Grenzwert für HH-Alarm	<p>Oberster Grenzwert, der ein Ereignis auslöst. Bedingung: (HH Alarm Value - Hysterese) &gt; H Alarm Value oder HH Alarm Value = H Alarm Value</p> <p>Standardwert: 0, Darstellung abhängig vom Typ der globalen Variablen</p>								
Priorität für HH-Alarm	<p>Priorität des obersten Grenzwerts.</p> <p>Je höher der eingestellte Wert, umso niedriger ist die Priorität.</p> <p>Wertebereich: 0 ... 1000</p> <p>Standardwert: 500</p>								
Bestätigung für HH-Alarm erforderlich	<p>Bestätigung des HH-Alarms durch den Bediener (Quittierung).</p> <p>Aktiviert: Bediener muss Überschreitung des obersten Grenzwerts bestätigen (Quittierung).</p> <p>Deaktiviert: Bediener muss Überschreitung des obersten Grenzwerts nicht bestätigen.</p> <p>Standardeinstellung: Deaktiviert</p>								
Text für H-Alarm	Text, der den Alarmzustand des oberen Grenzwerts benennt.								
Grenzwert für H-Alarm	<p>Oberer Grenzwert, der ein Ereignis auslöst. Bedingung: (H Alarm Value - Hysterese) &gt; (L Alarm Value + Hysterese) oder H Alarm Value = L Alarm Value</p> <p>Standardwert: 0, Darstellung abhängig vom Typ der globalen Variablen</p>								
Priorität für H-Alarm	<p>Priorität des oberen Grenzwerts.</p> <p>Je höher der eingestellte Wert, umso niedriger ist die Priorität.</p> <p>Wertebereich: 0 ... 1000</p> <p>Standardwert: 500</p>								

Parameter	Beschreibung
Bestätigung für H-Alarm erforderlich	Bestätigung des H-Alarms durch den Bediener (Quittierung). Aktiviert: Bediener muss Überschreitung des oberen Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Überschreitung des oberen Grenzwerts nicht bestätigen. Standardeinstellung: Deaktiviert
Text für Rückkehr zu Normalzustand	Text, der den Normalzustand benennt.
Priorität für Rückkehr zu Normalzustand	Priorität für die Rückkehr in den Normalzustand. Je höher der eingestellte Wert, umso niedriger ist die Priorität. Wertebereich: 0 ... 1000 Standardwert: 500
Bestätigung für Rückkehr zu Normalzustand	Bestätigung der Rückkehr in den Normalzustand durch den Bediener (Quittierung). Aktiviert: Bestätigung erforderlich. Deaktiviert: Bestätigung nicht erforderlich. Standardwert: Deaktiviert
Text für L-Alarm	Text, der den Alarmzustand des unteren Grenzwerts benennt.
Grenzwert für L-Alarm	Unterer Grenzwert, der ein Ereignis auslöst. Bedingung: $(L \text{ Alarm Value} + \text{Hysterese}) < (H \text{ Alarm Value} - \text{Hysterese})$ oder $L \text{ Alarm Value} = H \text{ Alarm Value}$ Standardwert: 0, Darstellung abhängig vom Typ der globalen Variablen
Priorität für L-Alarm	Priorität des unteren Grenzwerts. Je höher der eingestellte Wert, umso niedriger ist die Priorität. Wertebereich: 0 ... 1000 Standardwert: 500
Bestätigung für L-Alarm erforderlich	Bestätigung des L-Alarms durch den Bediener (Quittierung). Aktiviert: Bediener muss Unterschreitung des unteren Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Unterschreitung des unteren Grenzwerts nicht bestätigen. Standardeinstellung: Deaktiviert
Text für LL-Alarm	Text, der den Alarmzustand des untersten Grenzwerts benennt.
Grenzwert für LL-Alarm	Unterster Grenzwert, der ein Ereignis auslöst. Bedingung: $(LL \text{ Alarm Value} + \text{Hysterese}) < (L \text{ Alarm Value})$ oder $LL \text{ Alarm Value} = L \text{ Alarm Value}$ Standardwert: 0, Darstellung ist abhängig vom Typ der globalen Variablen.
Priorität für LL-Alarm	Priorität des untersten Grenzwerts. Je höher der eingestellte Wert, umso niedriger ist die Priorität. Wertebereich: 0 ... 1000 Standardwert: 500
Bestätigung für LL-Alarm erforderlich	Bestätigung des LL-Alarms durch den Bediener (Quittierung). Aktiviert: Bediener muss Unterschreitung des untersten Grenzwerts bestätigen (Quittierung). Deaktiviert: Bediener muss Unterschreitung des untersten Grenzwerts nicht bestätigen. Standardeinstellung: Deaktiviert

Parameter	Beschreibung
Alarm-Hysterese	Die Hysterese verhindert ein ständiges Erzeugen von vielen Ereignissen, wenn der Prozesswert häufig um einen Grenzwert schwankt. Standardwert: 0, Darstellung ist abhängig vom Typ der globalen Variablen.

Tabelle 19: Parameter für skalare Ereignisse

**HINWEIS**

**Fehlerhafte Ereignisbildung durch Parametrierungsfehler möglich!**

Setzen der Parameter *Wert für L-Alarm* und *Wert für H-Alarm* auf denselben Wert kann zu unerwünschtem Verhalten der Ereignisbildung führen, da in diesem Fall kein Normalbereich existiert.

Deshalb sicherstellen, dass *Wert für L-Alarm* und *Wert für H-Alarm* unterschiedliche Werte haben.

5.2.11.3 Register **Eigenschaften**

Das Register **Eigenschaften** enthält die folgenden Parameter:

Bezeichnung	Beschreibung
Typ	Alarm&Events
Name	Name des Alarm&Events-Editors, nicht änderbar.
Alarm&Events-ID	Ein A&E-Cookie ist ein eindeutiger 32-bit-Wert und wird für jeden einzelnen Alarm generiert. Ein A&E-Cookie dient dazu, dass ein OPC Client einen Alarm eindeutig identifizieren kann. Nicht verwenden: Das A&E-Cookie wird aus dem Namen und der System-ID des Servers generiert. Alarm&Events-ID: Das A&E-Cookie wird aus dem Namen und der Alarm&Events-ID generiert. Wertebereich: 1 ... 511 Standardeinstellung: Nicht verwenden

Tabelle 20: Register **Eigenschaften**

## 5.2.11.4 Status von LL, L, N, H, HH bei X-AI 32 01 und X-AI 32 02

Sind für einen Kanal eines analogen Eingangsmoduls X-AI 32 01 oder X-AI 32 02 skalare Ereignisse für Grenzwerte definiert, dann sind die Zustandsvariablen -> *Zustand LL*, -> *Zustand L*, -> *Zustand N*, -> *Zustand H*, -> *Zustand HH* verfügbar.

**Diese Zustandsvariablen müssen für sicherheitsbezogene Anwendungen mit der Variablen *Kanal OK* verknüpft werden!**

**5.3 Forcen**

Unter Forcen versteht man das manuelle Beschreiben von Variablen mit Werten, die sich nicht aus dem Prozess ergeben, sondern vom Anwender vorgegeben werden, während die Steuerung das Anwenderprogramm abarbeitet.

In einem System existieren verschiedene Arten von global force-baren Datenquellen:

- Alle Eingangs und Statusinformationen von Modulen (z. B. E/A-Module) und Kommunikationsprotokollen.
- Alle nicht beschriebenen, aber gelesenen globalen Variablen (VAR\_EXTERNAL).
- Alle von einem Anwenderprogramm beschriebenen globalen Variablen (VAR\_EXTERNAL).

Neben den global force-baren Datenquellen existieren in einem System auch verschiedene Arten von lokal (im Anwenderprogramm) force-baren Datenquellen:

- Alle nicht beschriebenen, aber gelesenen Anwenderprogramm-Variablen (VAR).
- Alle von einem Anwenderprogramm beschriebenen Variablen (VAR).

i

Beim Forcen einer Variable wird immer ihre Datenquelle geforct! Eine geforcte Variable ist vom Prozess unabhängig, da der Wert vom Anwender vorgegeben wird.

### 5.3.1 Verwendung von Forcen

Forcen unterstützt den Anwender bei folgenden Aufgaben, z. B:

- Zum Testen des Anwenderprogramms für Fälle, die im Normalbetrieb nicht oder nur selten eintreten und somit nur bedingt prüfbar sind.
- Zur Simulation von Sensorwerten, z. B. nicht verbundener Sensoren.
- Zu Service- und Reparaturarbeiten.
- Zur allgemeinen Fehlersuche.

#### **WARNUNG**



**Personenschäden durch geforcte Werte möglich!**

- **Werte nur nach Absprache mit dem Anlagenverantwortlichen und bei Inbetriebnahme mit der Prüfstelle forcen.**
- **Einschränkungen des Forcens nur nach Absprache mit Anlagenverantwortlichen und bei Inbetriebnahme mit der Prüfstelle aufheben.**

Während des Forcens muss der Verantwortliche die sicherheitstechnisch ausreichende Überwachung des Prozesses durch andere technische und organisatorische Maßnahmen gewährleisten. Es wird empfohlen, das Forcen zeitlich zu begrenzen, siehe Kapitel 5.3.3.

#### **WARNUNG**



**Störung des sicherheitsbezogenen Betriebs durch geforcte Werte möglich!**

- **Geforcte Werte können zu unerwarteten Ausgangswerten führen.**
- **Forcen verlängert die Zykluszeit. Dadurch kann die Watchdog-Zeit überschritten werden.**

Forcen kann in zwei Geltungsbereichen erfolgen:

- Globales Forcen: Globale Variable werden für alle Verwendungen geforct.
- Lokales Forcen: Lokalen Variablen werden innerhalb eines Anwenderprogramms geforct.

### 5.3.2 Per Reload geänderte Zuweisung einer Datenquelle

Das Ändern von Zuweisungen von Variablen zu einer anderen Datenquelle per Reload kann bei folgenden Eingängen zu einem unerwarteten Ergebnis führen:

- Hardware.
- Kommunikationsprotokolle.
- Systemvariablen.

Folgende per Reload durchgeführte Änderungen führen zu geänderten Force-Zuständen:

1. Eine globale Variable A ist einer geforcten Datenquelle zugewiesen und ist damit geforct.
2. Die Zuweisung der globalen Variable A wird per Reload entfernt. Die Datenquelle behält die Eigenschaft *geforct*. Die globale Variable A ist jetzt nicht mehr geforct.
3. Die geforcte Datenquelle wird einer anderen globalen Variable B zugeordnet.
4. Beim nächsten Reload ist dann die globale Variable B geforct, obwohl dies nicht beabsichtigt war.

### Konsequenz

Um dies zu vermeiden, beenden Sie zuerst das Forcen einer Variable, bevor die Datenquelle geändert wird. Dazu den Force-Einzelschalter deaktivieren.

Welche Kanäle geforct sind, ist im Register *Eingänge* des Force-Editors erkennbar.

---

## i

Globale Variablen, deren Datenquelle das Anwenderprogramm ist, behalten die Eigenschaft *geforct* auch dann bei, wenn die Zuweisung geändert wird.

---

### 5.3.3 Zeitbegrenzung

Für das globale wie für das lokale Forcen sind unterschiedliche Zeitbegrenzungen einstellbar. Nach Ablauf der eingestellten Zeit beendet die Steuerung das Forcen.

Das Verhalten des HIMax Systems nach dem Ablauf der Zeitbegrenzung ist einstellbar:

- Beim globalen Forcen sind folgende Einstellungen wählbar:
  - *Ressource stoppen*.
  - *Nur Forcen beenden*, d. h. die Ressource läuft weiter.
- Beim lokalen Forcen sind folgende Einstellungen wählbar:
  - *Programm stoppen*.
  - *Nur Forcen beenden*, d. h. das Anwenderprogramm läuft weiter.

Forcen ist auch ohne Zeitbegrenzung möglich. In diesem Fall ist das Forcen manuell zu beenden.

Der für das Forcen Verantwortliche muss klären, welche Auswirkungen das Beenden des Forcens auf die Gesamtanlage hat!

### 5.3.4 Einschränkung des Forcens

Der Anwender hat die Möglichkeit die Benutzung des Forcens einzuschränken, eventuelle Störungen des Betriebs durch das Forcen sind zu vermeiden. In der Konfiguration können folgende Maßnahmen dafür getroffen werden:

- Die Einrichtung unterschiedlicher Benutzerkonten mit und ohne Force-Rechten.
- Das Forcen für eine Ressource (PES) explizit erlauben.
- Die Einrichtung von MultiForce-Benutzerkonten in der PES-Benutzerverwaltung.
- Das lokale Forcen für ein Anwenderprogramm explizit erlauben.
- Die Wirkung des Forcens kann über die Systemvariable *Force-Deaktivierung* per Schlüsselschalter unmittelbar abgeschaltet werden.
- Zusätzlich kann über die Systemvariable *MultiForcen gesperrt* MultiForcen unterbunden werden.

### 5.3.5 Force-Editor

Der Force-Editor von SILworX zeigt alle Variablen an, getrennt nach globalen und lokalen Variablen.

Für jede Variable ist folgendes einstellbar:

- Force-Werte.
- Ein Force-Einzelschalter, um das Forcen der Variable vorzubereiten.

Das Forcen lässt sich jeweils für globale und lokale Variable starten und stoppen.

Das Forcen erfolgt für eine festgelegte Zeitdauer oder für unbegrenzte Zeit. Nach dem Starten des Forcens, werden alle Variablen, deren Force-Einzelschalter eingeschaltet sind, auf ihren Force-Wert gesetzt.

Beim Stopp des Forcens - manuell oder durch die Zeitbegrenzung - erhalten die Variablen wieder die Werte vom Prozess.

Bei einem erneuten Start des Forcens ersetzen die vorher eingestellten Force-Werte wieder die Werte aus dem Prozess!

Weitere Einzelheiten zum Forcen und zum Force-Editor finden Sie in der Online-Hilfe von SILworX.

Das Kopieren aktueller Daten aus dem Force-Editor in die Zwischenablage erfasst nur den sichtbaren Bereich im Force-Editor. Daten, die sich außerhalb des sichtbaren Bereichs befinden, werden nicht regelmäßig aktualisiert und können einen Wert haben, der vor Minuten oder Stunden von der Ressource gelesen wurde! Die Tastenkombination **STRG-A** wählt auch Daten außerhalb des sichtbaren Bereichs zum Kopieren aus.

### 5.3.6 Automatisches Zurücksetzen des Forcens

Das Betriebssystem setzt in folgenden Fällen das Forcen zurück:

- Bei Neustart der Ressource, z. B. nach Zuschalten der Versorgungsspannung.
- Beim Stoppen der Ressource.
- Beim Laden einer neuen Konfiguration per Download.
- Beim Stoppen eines Anwenderprogramms: Rücksetzen des lokalen Forcens für dieses Anwenderprogramm

Beim Reload bleiben die lokalen und globalen Force-Werte und Force-Einzelschalter weiterhin gültig, ebenso die Force-Zeiten und Force-Timeout-Reaktionen.

#### **WARNUNG**



**Störung des sicherheitsbezogenen Betriebs durch geforcete Werte möglich!**

- Bei gestoppter Ressource ist es möglich, globale Force-Werte und Force-Einzelschalter einzustellen. Diese werden nach dem Start der Ressource und des Forcens gültig.
- Bei gestopptem Anwenderprogramm ist es möglich, lokale Force-Werte und Force-Einzelschalter einzustellen. Diese werden nach dem Start des Anwenderprogramms und des Forcens gültig.

### 5.3.7 Forcen und skalare Ereignisse

Beim Forcen einer globalen Variable, die zur Bildung von skalaren Ereignissen verwendet wird, ist folgendes zu beachten:

- Ereignisse werden entsprechend dem Force-Wert gebildet.
- Die Werte der von dieser Variable abhängigen Statusvariablen werden dem Force-Wert nicht nachgeführt!

In einem solchen Fall auch die abhängigen Statusvariablen forcen!

### 5.3.8 MultiForcen

Anwender mit MultiForcen-Zugriff können in einer Ressource Force-Daten (Force-Werte und Force-Einzelschalter) für globale Variablen schreiben, wenn die dafür erforderlichen, übergeordneten Bedingungen gegeben und Force-Freigaben erfolgt sind. Auf alle anderen Funktionen einer Ressource kann der Anwender nur lesend zugreifen. Das Starten, Stoppen oder Zurücksetzen eines Force-Vorgangs ist nicht möglich.

Das MultiForcen ist auf bis zu 5 Benutzer gleichzeitig begrenzt. Die Benutzer können räumlich voneinander entfernt sein und auch zeitlich unabhängig voneinander arbeiten. Die Abgrenzung der Aufgaben der einzelnen Benutzer ist durch organisatorische Maßnahmen des Betreibers sicherzustellen.

#### WARNUNG



**Nicht steuerbares Verhalten durch den Anwender möglich!**

Der Betreiber muss dafür sorgen, dass verschiedene Force-User nicht gleichzeitig dieselben Variablen forcen und es nicht zu zeitlichen Überschneidungen kommt. Schreiben mehrere Force-User auf dieselben Variablen, setzen sich diejenigen Force-Werte und Force-Einzelschalter durch, die von der Firmware zuletzt geschrieben wurden. Da Force-Daten in mehreren Blöcken übertragen werden, können auf einer einzelnen Steuerung anderenfalls auch Einstellungen unterschiedlicher Force-User wirksam werden. Dieses Verhalten ist für den Anwender nicht steuerbar!

#### WARNUNG



***MultiForcen gesperrt* = TRUE, bestehende Force-Daten werden nicht deaktiviert!**

Wenn *MultiForcen gesperrt* = TRUE ist, können Anwender mit MultiForcen-Zugriff keine Veränderungen an den Force-Werten und den Force-Einzelschaltern vornehmen. Bestehende Force-Daten werden nicht deaktiviert, wenn *MultiForcen gesperrt* = TRUE ist! Globales Forcen ist, wenn erlaubt, dann nur für einen einzigen Benutzer mit mindestens Bedienerrechten möglich.

#### 5.3.8.1 Ziele von MultiForcen

Für die Inbetriebnahme sind im Rahmen der Site Acceptance Tests normativ und funktional Loop-Tests vorgeschrieben, wobei ein Loop den Weg vom Sensor zum Aktor darstellt. MultiForcen ermöglicht es, die anfallenden Aufgaben auf bis zu 5 PADTs zu verteilen und damit effizient abzuarbeiten.

Anhand von Loop-Tests wird der nominale Betriebsbereich geprüft, ebenso wie die Reaktionen bei Leitungsbruch und Leitungsschluss. Da häufig zahlreiche Loops getestet werden müssen, ist die Dauer von Site Acceptance Tests ein wesentlicher Kostenfaktor. MultiForcen kann helfen, diese Aufgaben zu optimieren.

- Das Verhalten von Aktoren und verknüpften Informationen (z. B. Endlagenrückmeldung) wird durch Forcen getestet. Die Ausgangssignale werden direkt geforct. Dadurch wird die Verdrahtung und externe Schaltung geprüft.
- In einer Anlage, die sich im Teilbetrieb befindet, werden Sensoren durch Forcen so getestet, dass die Tests keine Auswirkung auf die Aktoren haben. Diese Variante kann auch bei der Fehlersuche im Zusammenhang mit Sensoren zur Anwendung kommen.

#### 5.3.8.2 Globales MultiForcen

Globales MultiForcen ist das gleichzeitige Schreiben von Force-Daten (Force-Werte und Force-Einzelschalter) für globale Variablen durch mehr als einen Benutzer (Force-User).

Ein Force-User ist eine Person, die entweder mit MultiForcen-Rechten, Bedienerrechten, Schreibrechten oder mit Administratorrechten in einer Steuerung eingeloggt ist. Jeder Force-User kann neben dem Lesen von Daten mindestens auch Force-Daten schreiben. Pro Steuerung können maximal 5 Force-User eingeloggt sein. Die Anzahl der aktuellen Force-User wird in der SILworX -Statuszeile angezeigt.

Um die durch Force-User mit MultiForcen-Zugriff eingestellten Force-Werte und Force-Einzelschalter wirksam werden zu lassen ist ein Anwender erforderlich, der mit mindestens Bedienerrechten in der Steuerung eingeloggt ist. Nur dieser Anwender kann Forcen starten und stoppen.



Um globales MultiForcen durchführen zu können, muss auch globales Forcen erlaubt sein! Die Einstellungen werden online angezeigt.

---

### 5.4 Ablauf des Zyklus

Der Zyklus eines Prozessormoduls (CPU-Zyklus) für nur ein Anwenderprogramm besteht - vereinfacht dargestellt - aus folgenden Phasen:

1. Verarbeitung der Eingabedaten.
2. Abarbeitung des Anwenderprogramms.
3. Bereitstellung der Ausgabedaten.

Nicht dargestellt sind besondere Aufgaben, die gegebenenfalls innerhalb des CPU-Zyklus durchgeführt werden, etwa Reload.

Die erste Phase stellt die globalen Variablen, Ergebnisse von Funktionsbausteinen und andere Daten als Eingabedaten für die zweite Phase bereit. Die erste Phase muss nicht mit dem Anfang des Zyklus beginnen, sondern kann sich verschieben. Daher führt der Versuch, die Zykluszeit im Anwenderprogramm mit Hilfe von Timer-Funktionsbausteinen zu bestimmen, zu ungenauen Ergebnissen, bis hin zu Zykluszeiten, die größer als die Watchdog-Zeit sind.

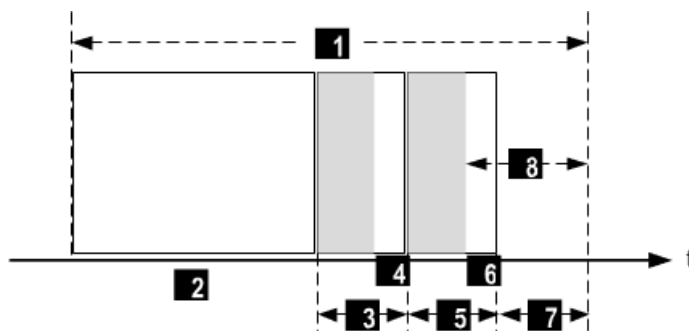
Die dritte Phase gibt die Ergebnisse des Anwenderprogramms für die Verarbeitung in folgenden Zyklen und für die Ausgangskanäle weiter.



### 5.4.1 Reservezeit zur Watchdog-Zeit und zur Sollzykluszeit

Das Control Panel von SILworX zeigt die verfügbaren Reservezeiten an, *Reserve zur Watchdogzeit [ms]* und *Reserve zur Sollzykluszeit [ms]*.

Die Reservezeiten geben die Zeit innerhalb des Zyklus an, die der Steuerung für zusätzliche Aufgaben wie Reload zur Verfügung steht, wenn die konfigurierten Zeiten im Zyklus ausgeschöpft würden.



- |  |   |
|--|---|
| <b>1</b> Watchdog-Zeit oder Sollzykluszeit   | <b>5</b> Konfigurierter Wert <i>Max. Kom.-Zeitscheibe ASYNC [ms]</i>        |
| <b>2</b> Von den Programmen benötigte Zeit   | <b>6</b> Nicht benötigter Anteil an <i>Max. Kom.-Zeitscheibe ASYNC [ms]</i> |
| <b>3</b> Konfigurierter Wert<br><i>Max. Dauer Konfigurationsverbindungen [ms]</i>        | <b>7</b> Angezeigte Reservezeit bei Betriebssystemen ab V6                  |
| <b>4</b> Nicht benötigter Anteil an<br><i>Max. Dauer Konfigurationsverbindungen [ms]</i> | <b>8</b> Angezeigte Reservezeit bei Betriebssystemen vor V6                 |

Bild 14: Reservezeiten innerhalb des Zyklus

Hinweis zum Bild 14: die Größenverhältnisse und die Reihenfolge der Zeiten entsprechen nicht unbedingt den tatsächlichen Gegebenheiten!

Die konfigurierten Zeiten gehen als  $t_{\text{CycleAdjust}}$  in die Berechnungen der Reservezeiten ein. Es gilt:

Betriebssystem-Version	Wert von $t_{\text{CycleAdjust}}$
Ab V6	Konfigurierte Werte <i>Max. Dauer Konfigurationsverbindungen [ms]</i> + $t_{\text{EA-Phasen-Timeout}}$ (siehe unten) + <i>Max. Kom.-Zeitscheibe ASYNC [ms]</i>
Vor V6	Konfigurierter Wert <i>Max. Dauer Konfigurationsverbindungen [ms]</i> + $t_{\text{EA-Phasen-Timeout}}$ (siehe unten)

Tabelle 21:  $t_{\text{CycleAdjust}}$  abhängig von der Version des CPU-Betriebssystems

Der Wert von  $t_{\text{EA-Phasen-Timeout}}$  ist mit folgender Tabelle abhängig vom Parameter *Maximale Systembus-Latenzzeit [μs]* und dem Typ des Prozessormoduls bestimmbar:

<i>Maximale Systembus-Latenzzeit [μs]</i>	Prozessormodul-Typ	$t_{\text{EA-Phasen-Timeout}} =$
≥ 100	X-CPU 01	$2 * \text{Maximale Systembus-Latenzzeit } [\mu\text{s}] + 0,169 \mu\text{s}$
	X-CPU 31	$2 * \text{Maximale Systembus-Latenzzeit } [\mu\text{s}] + 0,533 \mu\text{s}$
≠ System-Standardwerte	X-CPU 01	$1,232 \text{ ms} + 5,5 \text{ ms} = 6,732 \text{ ms}$
	X-CPU 31	$1,844 \text{ ms} + 5,5 \text{ ms} = 7,344 \text{ ms}$

Tabelle 22: Bestimmung von  $t_{\text{EA-Phasen-Timeout}}$

Reload nutzt die Zeiten wie folgt:

- Der Reload eines Anwenderprogramms, dessen Zyklus **mehrere** Zyklen des Prozessormoduls benötigt, nutzt die angezeigte Reservezeit (**7** oder **8**), Ein derartiger Reload wird abgelehnt, wenn die Reservezeit < 6 ms ist!
- Der Reload eines Anwenderprogramms, dessen Zyklus nur **einen** Zyklus des Prozessormoduls benötigt, nutzt folgende Zeiten:
  - Die angezeigte Reservezeit (**7** oder **8**)
  - Den Anteil an den konfigurierten Zeiten ( $t_{\text{CycleAdjust}}$ ), der nicht ausgeschöpft wird (**4** und, ab V6, **6**).

Daher ist in diesem Fall ein Reload auch möglich, wenn die angezeigte Reservezeit = 0 ist.

Folgende Aktivitäten nutzen die angezeigte Reservezeit, um etwas schneller abzulaufen:

- Aktivitäten beim Starten eines Anwenderprogramms im Zustand RUN der PES
- Spannungsausfallsicheres Speichern der Projektkonfiguration im Rahmen von Reload

### Reservezeit zur Watchdog-Zeit

Die *Reserve zur Watchdogzeit [ms]* errechnet sich aus:

$$t_{\text{Reserve}} \leq t_{\text{WDZ}} - t_z - t_{\text{VarAdjust}} - t_{\text{CycleAdjust}}$$

Dabei bedeuten:

$t_{\text{Reserve}}$	<i>Reserve zur Watchdogzeit [ms]</i>
$t_{\text{WDZ}}$	Konfigurierte Watchdog-Zeit
$t_z$	Tatsächlich benötigte Zykluszeit
$t_{\text{VarAdjust}}$	10 ms bei HIMax
$t_{\text{CycleAdjust}}$	Siehe Tabelle 21

### Reservezeit zur Sollzyklus-Zeit

Ist eine Sollzykluszeit > 0 konfiguriert, zeigt SILworX im Control Panel einen Wert > 0 für die *Reserve zur Sollzykluszeit* an. Für die *Reserve zur Sollzykluszeit* gilt eine folgende Berechnung:

$$t_{\text{Reserve}} \leq t_{\text{Soll}} - t_z - t_{\text{VarAdjust}} - t_{\text{CycleAdjust}}$$

Dabei bedeuten:

$t_{\text{Reserve}}$	<i>Reserve zur Sollzykluszeit [ms]</i>
$t_{\text{Soll}}$	Konfigurierte Sollzykluszeit
$t_z$	Tatsächlich benötigte Zykluszeit
$t_{\text{VarAdjust}}$	2 ms
$t_{\text{CycleAdjust}}$	Siehe Tabelle 21

## i

Eine gewisse Reservezeit zur Sollzykluszeit muss existieren, sonst kann HIMax die Synchronisierung eines Prozessormoduls ablehnen. Die Ablehnung erfolgt auch dann, wenn der *Sollzykluszeit-Modus* auf *fest-tolerant* oder *dynamisch-tolerant* eingestellt ist.

Der Wert der Reservezeit zur Sollzykluszeit ist abhängig von der Einstellung folgender Systemparameter (siehe vorhergehende Formeln und Tabellen):

- *Max. Dauer Konfigurationsverbindungen [ms]*
- *Max. Kom.-Zeitscheibe ASYNC [ms]*
- *Maximale Systembus-Latenzzeit [μs]*

**Abhilfe:** Die Systemparameter so einstellen, dass eine ausreichend große Reservezeit zur Sollzykluszeit existiert.

### 5.4.2 Multitasking

Multitasking bezeichnet die Fähigkeit des HIMax Systems, bis zu 32 Anwenderprogramme innerhalb des Prozessormoduls abzuarbeiten.

Dadurch lassen sich Teilfunktionen eines Projekts voneinander trennen. Die einzelnen Anwenderprogramme lassen sich unabhängig voneinander starten und stoppen. SILworX zeigt im Control Panel die Zustände der einzelnen Anwenderprogramme an und ermöglicht deren Bedienung.

Bei Multitasking ändert sich die zweite Phase, so dass ein CPU-Zyklus folgendermaßen abläuft:

1. Verarbeitung der Eingabedaten.
2. Abarbeitung aller Anwenderprogramme.
3. Bereitstellung der Ausgabedaten.

In der zweiten Phase kann HIMax bis zu 32 Anwenderprogramme abarbeiten. Dabei sind für jedes Anwenderprogramm zwei Fälle möglich:

- Innerhalb eines CPU-Zyklus wird ein vollständiger Zyklus des Anwenderprogramms abgearbeitet.
- Ein vollständiger Zyklus des Anwenderprogramms benötigt mehrere CPU-Zyklen zur Abarbeitung.

Innerhalb eines CPU-Zyklus ist eine Übergabe von globalen Daten zwischen Anwenderprogrammen nicht möglich. Die von einem Anwenderprogramm geschriebenen Daten werden nach der vollständigen Ausführung des Anwenderprogramms unmittelbar vor Phase 3 verfügbar gemacht. Damit können diese Daten erst beim nächsten Start eines anderen Anwenderprogramms als Eingangswerte genutzt werden.

Das Beispiel in Bild 15 zeigt beide Fälle in einem Projekt, das zwei Anwenderprogramme, AP 1 und AP 2 enthält.

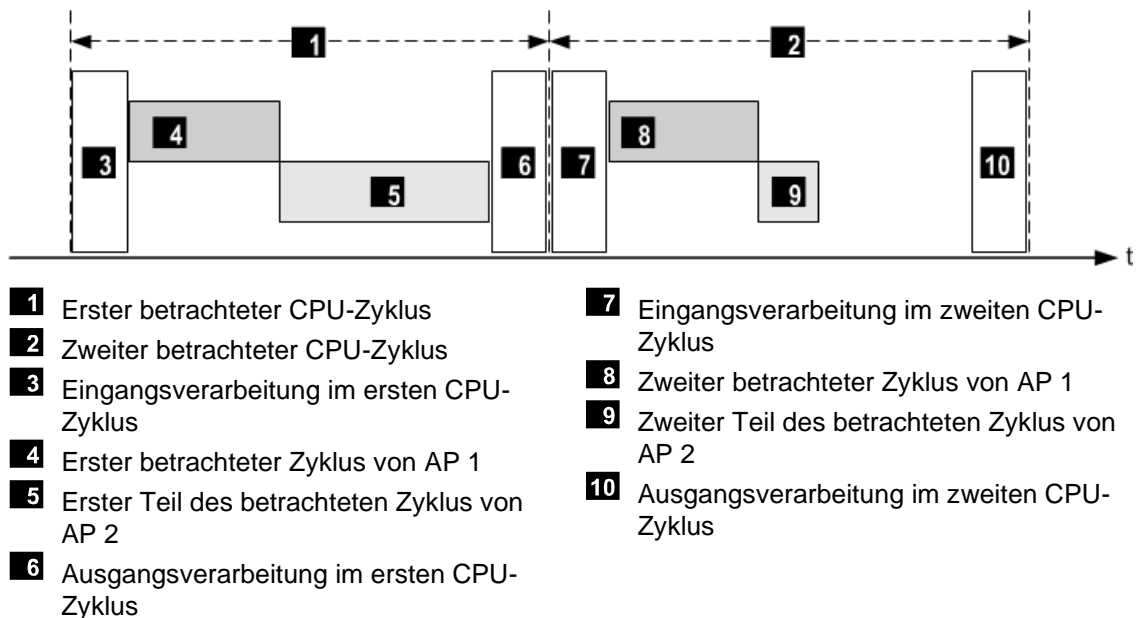


Bild 15: Ablauf des CPU-Zyklus bei Multitasking

Jeder Zyklus des Anwenderprogramms AP 1 wird in jedem CPU-Zyklus vollständig abgearbeitet. AP 1 verarbeitet eine Eingabeänderung, die das System am Anfang des CPU-Zyklus **1** registriert hat, und liefert eine Reaktion am Ende dieses Zyklus.

Ein Zyklus des Anwenderprogramms AP 2 benötigt zu seiner Abarbeitung zwei CPU-Zyklen. AP 2 benötigt zur Verarbeitung einer Eingabeänderung, die das System am Anfang des CPU-Zyklus **1** registriert hat, auch noch CPU-Zyklus **2**. Aus diesem Grund steht die Reaktion auf

diese Eingabeänderung erst am Ende von CPU-Zyklus **2** zur Verfügung.  
Die Reaktionszeit von AP 2 ist doppelt so groß wie die von AP 1.

Am Ende des ersten Teils **5** des betrachteten Zyklus von AP 2 wird die Abarbeitung von AP 2 **vollständig** unterbrochen und erst mit dem Beginn von **9** fortgesetzt. AP 2 verarbeitet während seines Zyklus die Daten, die das System zum Zeitpunkt **3** bereitgestellt hat. Die Ergebnisse von AP 2 stehen dem System zum Zeitpunkt **10** zur Verfügung (z. B. zur Ausgabe zum Prozess). Die Daten, die das Anwenderprogramm mit dem System austauscht, sind immer konsistent.

Die Verarbeitung der Programme ist durch eine Priorität steuerbar, die angibt, wie wichtig das jeweilige Anwenderprogramm im Verhältnis zu anderen ist (siehe Multitasking Mode 2).

Die Abarbeitung der Anwenderprogramme ist durch folgende Parameter bei Ressource und Programmen oder im Multitasking Editor festlegbar:

Parameter	Beschreibung	Einstellbar bei
Watchdog-Zeit	Watchdog-Zeit der Ressource.	Ressource, Multitasking Editor
Sollzykluszeit [ms]	Gewünschte oder maximale Zykluszeit.	Ressource, Multitasking Editor
Multitasking Mode	Verwendung der von Anwenderprogrammen nicht benötigten Ausführungsdauer, d. h. der Differenz zwischen der tatsächlichen Ausführungsdauer in einem CPU-Zyklus und der eingestellten <i>Max. Dauer pro Zyklus [µs]</i> .	Ressource, Multitasking Editor
	Mode 1 Die Länge eines Zyklus der CPU richtet sich nach der benötigten Ausführungsdauer aller Anwenderprogramme.	
	Mode 2 Prozessor stellt von Anwenderprogrammen niederer Priorität nicht benötigte Ausführungszeit den Anwenderprogrammen hoher Priorität zur Verfügung. Betriebsart für hohe Verfügbarkeit.	
	Mode 3 Prozessor wartet nicht benötigte Ausführungszeit von Anwenderprogrammen ab und verlängert so den Zyklus.	
Sollzykluszeit-Mode	Verwendung der <i>Sollzykluszeit [ms]</i> .	Ressource, Multitasking Editor
Programm ID	ID für die Identifizierung des Programms bei der Anzeige in SILworX.	Anwenderprogramm
Priorität	Wichtigkeit eines Anwenderprogramms, höchste Priorität: 0.	Anwenderprogramm
Maximale CPU-Zyklen Programm	Maximale Anzahl CPU-Zyklen zur Abarbeitung eines Zyklus des Anwenderprogramms.	Anwenderprogramm
Max. Dauer pro Zyklus [µs]	Zulässige Ausführungsdauer für ein Anwenderprogramm innerhalb eines CPU-Zyklus.	Anwenderprogramm

Tabelle 23: Für Multitasking einstellbare Parameter

Bei der Festlegung der Parameter folgende Regeln beachten:

- Ist die *Max. Dauer pro Zyklus [µs]* auf 0 gesetzt, so ist die Ausführungszeit des Anwenderprogramms nicht begrenzt, d. h., es wird immer vollständig ausgeführt. Daher darf *Maximale CPU-Zyklen Programm* in diesem Fall nur 1 sein.

- Die Summe der Parameter *Max. Dauer pro Zyklus [μs]* aller Anwenderprogramme darf nicht größer als die Watchdog-Zeit der Ressource sein. Dabei ist auf eine ausreichende Reserve zur Bearbeitung der restlichen Aufgaben des Systems zu achten, siehe Kapitel 5.4.1.
- Die Summe der Parameter *Max. Dauer pro Zyklus [μs]* aller Anwenderprogramme muss so groß sein, dass noch eine Reserve für die Einhaltung der Sollzykluszeit bleibt.
- Die *Programm IDs* aller Anwenderprogramme müssen eindeutig sein.

SILworX überwacht die Einhaltung dieser Regeln bei der Verifizierung und Codegenerierung. Bei der Änderung von Parametern durch Online-Änderung oder Reload sind diese Regeln ebenfalls einzuhalten. Außerdem ist zu beachten, dass sich das System bei der Änderung von *Maximale CPU-Zyklen Programm* durch Online-Änderung oder Reload folgendermaßen verhält:

- Wird *Maximale CPU-Zyklen Programm* gegenüber dem aktuell gültigen Wert **verringert**, so wird die Änderung erst gültig, wenn das Anwenderprogramm nicht mehr ausgeführt wird. Der neue Wert gilt erst für die Ausführung des nächsten Anwenderprogramm-Zyklus. Während der Ausführung des Anwenderprogramm-Zyklus wird *Maximale CPU-Zyklen Programm* nicht verringert. Dadurch ist sichergestellt, dass ein über mehrere CPU-Zyklen hinweg ausgeführtes Anwenderprogramm nicht aufgrund einer Überschreitung von *Maximale CPU-Zyklen Programm* in FEHLER\_STOPP geht.
- Wird *Maximale CPU-Zyklen Programm* online gegenüber dem aktuell gültigen Wert **erhöht**, so wird die Änderung sofort gültig. Nach einem Reload wird die Änderung gültig, sobald die Reload-Konfiguration in der Ressource aktiv wird. Die Erhöhung von *Maximale CPU-Zyklen Programm* ist auch während der Ausführung des Anwenderprogramm-Zyklus möglich.

Aus diesen Parametern errechnet SILworX die Watchdog-Zeit des Anwenderprogramms zu:

Watchdog-Zeit des Anwenderprogramms = *Watchdog-Zeit* \* *Maximale CPU-Zyklen Programm*

---

## i

Die Ablaufsteuerung zur Ausführung der Anwenderprogramme arbeitet in Schritten zu 250 μs. Aus diesem Grund können die parametrisierten Werte für *Max. Dauer pro Zyklus [μs]* um bis zu 250 μs über- oder unterschritten werden.

---

Die einzelnen Anwenderprogramme laufen generell rückwirkungsfrei voneinander ab. Gegenseitige Beeinflussung ist jedoch möglich durch:

- Verwendung derselben globalen Variablen in mehreren Anwenderprogrammen.
  - Unvorhersehbar lange Laufzeiten bei einzelnen Anwenderprogrammen, falls keine parametrisierte Limitierung durch *Max. Dauer pro Zyklus [μs]* erfolgt.
  - Die Verteilung der Anwenderprogramm-Zyklen auf Prozessormodul-Zyklen beeinflusst die Reaktionszeit des Anwenderprogramms und der von ihm beschriebenen Variablen stark!
  - Ein Anwenderprogramm wertet globale Variable, die ein anderes Anwenderprogramm beschrieben hat, mindestens einen Zyklus des Prozessormoduls später aus. Abhängig von der Einstellung *Maximale CPU-Zyklen Programm* bei den Programmen kann das Auslesen um viele Zyklen des Prozessormoduls verzögert geschehen. Die Reaktion auf Änderungen derartiger globaler Variablen ist somit entsprechend verzögert!
- 

## i

HIMA empfiehlt, den Parameter *Max. Dauer pro Zyklus [μs]* auf einen geeigneten Wert ≠ 0 einzustellen. Dadurch wird das jeweilige Anwenderprogramm bei zu langer Laufzeit im aktuellen CPU-Zyklus beendet und im nächsten fortgesetzt, ohne die anderen Anwenderprogramme zu beeinträchtigen.

Andernfalls ist es möglich, dass eine ungewöhnlich lange Laufzeit eines oder mehrerer Anwenderprogramme zu einem Überschreiten der Sollzykluszeit oder gar der Watchdog-Zeit der Ressource und damit zum Fehlerstopp der Steuerung führt.

---

Das Betriebssystem legt die Ausführungsreihenfolge der Anwenderprogramme folgendermaßen fest:

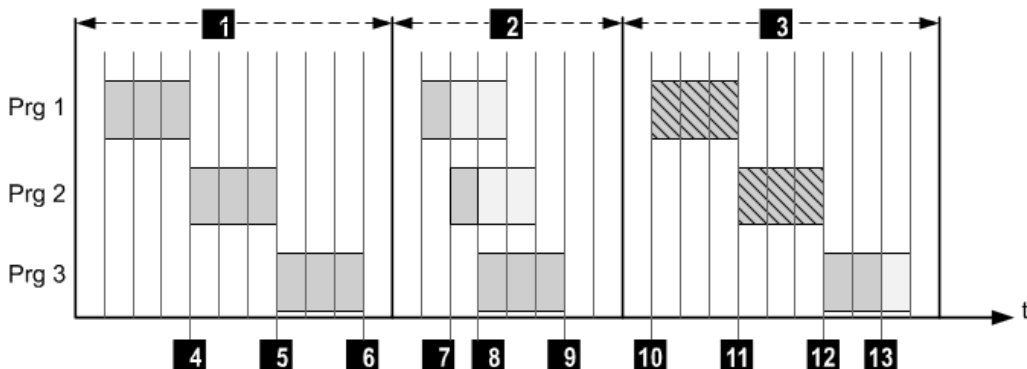
- Das System arbeitet Anwenderprogramme mit niedrigerer Priorität vor Anwenderprogrammen mit höherer Priorität ab.
- Haben Anwenderprogramme die gleiche Priorität, arbeitet das System diese nach aufsteigenden *Programm IDs* ab.

Diese Reihenfolge gilt auch für das Starten und Stoppen der Anwenderprogramme beim Starten bzw. Stoppen des PES.

### 5.4.3 Multitasking-Mode

Es gibt drei Arbeitsweisen des Multitasking, Multitasking Mode 1 ... Multitasking Mode 3. Die Arbeitsweisen unterscheiden sich durch die Nutzung nicht benötigter Zeit der Ausführungsauern pro CPU-Zyklus der Anwenderprogramme. Für jede Ressource ist eine Arbeitsweise wählbar:

1. **Multitasking Mode 1** nutzt nicht benötigte Dauer zur Verringerung des CPU-Zyklus. Ist die Bearbeitung eines Anwenderprogramms abgeschlossen, wird sofort die Bearbeitung des nächsten Anwenderprogramms gestartet. Insgesamt ergibt sich dadurch ein kürzerer Zyklus. Beispiel: 3 Anwenderprogramme mit den Namen *Prg 1*, *Prg 2* und *Prg 3*, bei denen ein Zyklus des Anwenderprogramms bis zu 3 CPU-Zyklen dauern darf.



- 1 Erster betrachteter CPU-Zyklus. Weitere Nummern siehe Text.
- 2 Zweiter betrachteter CPU-Zyklus.
- 3 Dritter betrachteter CPU-Zyklus.

Bild 16: Multitasking Mode 1

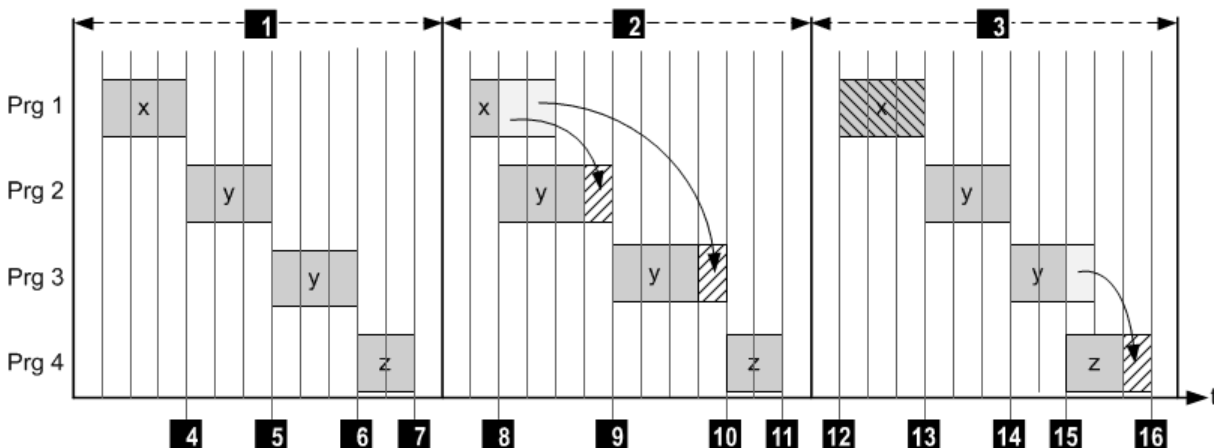
Ablauf der drei betrachteten Zyklen:

- 4 Max. Dauer pro Zyklus [ $\mu$ s] von *Prg 1* abgelaufen, *Prg 2* startet.
- 5 Max. Dauer pro Zyklus [ $\mu$ s] von *Prg 2* abgelaufen, *Prg 3* startet.
- 6 Max. Dauer pro Zyklus [ $\mu$ s] von *Prg 3* abgelaufen, Ende des ersten CPU-Zyklus.
- 7 Anwenderprogramm-Zyklus von *Prg 1* beendet, *Prg 2* wird fortgesetzt.
- 8 Anwenderprogramm-Zyklus von *Prg 2* beendet, *Prg 3* wird fortgesetzt.
- 9 Max. Dauer pro Zyklus [ $\mu$ s] von *Prg 3* abgelaufen, Ende des zweiten CPU-Zyklus.
- 10 Nächster Anwenderprogramm-Zyklus von *Prg 1* beginnt.
- 11 Max. Dauer pro Zyklus [ $\mu$ s] von *Prg 1* abgelaufen. Nächster Anwenderprogramm-Zyklus von *Prg 2* beginnt.
- 12 Max. Dauer pro Zyklus [ $\mu$ s] von *Prg 2* abgelaufen, *Prg 3* startet.
- 13 Anwenderprogramm-Zyklus von *Prg 3* beendet.

2. **Multitasking Mode 2** verteilt nicht benötigte Dauer von Anwenderprogrammen niedriger Priorität auf Anwenderprogramme höherer Priorität. Dadurch stehen diesen außer ihrer eingestellten *Max. Dauer pro Zyklus [μs]* noch die Anteile an der nicht benötigten Dauer zur Verfügung. Diese Arbeitsweise sorgt für hohe Verfügbarkeit.

Im folgenden Beispiel gibt es vier Anwenderprogramme mit den Namen *Prg 1 ... Prg 4*. Den Anwenderprogrammen sind folgende Prioritäten zugewiesen:

- *Prg 1* hat die niedrigste Priorität *x*
- *Prg 2* und *Prg 3* haben die mittlere Priorität *y*
- *Prg 4* hat die höchste Priorität *z*



**1** Erster betrachteter CPU-Zyklus.

Weitere Nummern siehe Text.

**2** Zweiter betrachteter CPU-Zyklus.

**3** Dritter betrachteter CPU-Zyklus.

Bild 17: Multitasking Mode 2

Ablauf der Zyklen:

- 4** *Max. Dauer pro Zyklus [μs]* von *Prg 1* abgelaufen, *Prg 2* startet.
- 5** *Max. Dauer pro Zyklus [μs]* von *Prg 2* abgelaufen, *Prg 3* startet.
- 6** *Max. Dauer pro Zyklus [μs]* von *Prg 3* abgelaufen, *Prg 4* startet.
- 7** *Max. Dauer pro Zyklus [μs]* von *Prg 4* abgelaufen, erster CPU-Zyklus beendet.
- 8** Anwenderprogramm-Zyklus von *Prg 1* beendet, *Prg 2* wird fortgesetzt. Restliche Dauer wird auf die *Max. Dauer pro Zyklus [μs]* von *Prg 2* und *Prg 3* (höhere Priorität *y*) verteilt (Pfeile).
- 9** *Max. Dauer pro Zyklus [μs]* von *Prg 2*+ anteilige Restdauer von *Prg 1* abgelaufen, *Prg 3* wird fortgesetzt.
- 10** *Max. Dauer pro Zyklus [μs]* von *Prg 3*+ anteilige Restdauer von *Prg 1* abgelaufen, *Prg 4* startet.
- 11** *Max. Dauer pro Zyklus [μs]* von *Prg 4* abgelaufen, zweiter CPU-Zyklus beendet.
- 12** Nächster Anwenderprogramm-Zyklus von *Prg 1* beginnt.
- 13** *Max. Dauer pro Zyklus [μs]* von *Prg 1* abgelaufen, *Prg 2* wird fortgesetzt.
- 14** *Max. Dauer pro Zyklus [μs]* von *Prg 2* beendet, *Prg 3* wird fortgesetzt.
- 15** Anwenderprogramm-Zyklus von *Prg 3* beendet, *Prg 4* wird fortgesetzt. Restliche Dauer wird *Prg 4* (höhere Priorität *z*) zugeschlagen.
- 16** *Max. Dauer pro Zyklus [μs]* von *Prg 4*+ Restdauer von *Prg 3* abgelaufen, dritter Zyklus beendet.

i

Die nicht verwendete Ausführungsdauer von Anwenderprogrammen, die nicht ausgeführt werden, steht nicht als Restzeit für andere Anwenderprogramme zur Verfügung. Anwenderprogramme werden nicht ausgeführt, wenn sie sich in einem der Zustände befinden:

- STOPP
- ERROR
- TEST\_MODE

Dies kann dazu führen, dass sich die Anzahl der CPU-Zyklen erhöht, die zur Abarbeitung des Zyklus eines anderen Anwenderprogramms benötigt werden.

**In diesem Fall kann zu niedrige Parametrierung der *Maximalen Zyklusanzahl* zur Überschreitung der maximalen Verarbeitungsdauer des Anwenderprogramms und zum FEHLER\_STOPP führen!**

**Maximale Verarbeitungsdauer = Max. Dauer pro Zyklus[μs] \* Maximale Zyklusanzahl**

Zur Prüfung der Parametrierung Multitasking Mode 3 verwenden!

3. **Multitasking Mode 3** nutzt die nicht benötigte Dauer nicht für die Ausführung von Anwenderprogrammen, sondern wartet die Zeit bis zum Erreichen der *Max. Dauer pro Zyklus [μs]* des Anwenderprogramms und startet die Bearbeitung des nächsten Anwenderprogramms. Dieses Verhalten führt zu gleicher Dauer der CPU-Zyklen.

Multitasking Mode 3 ist zur Prüfung gedacht, ob Multitasking Mode 2 auch im ungünstigsten Fall eine ordnungsgemäße Programmausführung gewährleisten kann.

Das Beispiel betrachtet Anwenderprogramme mit den Namen *Prg 1*, *Prg 2* und *Prg 3*:

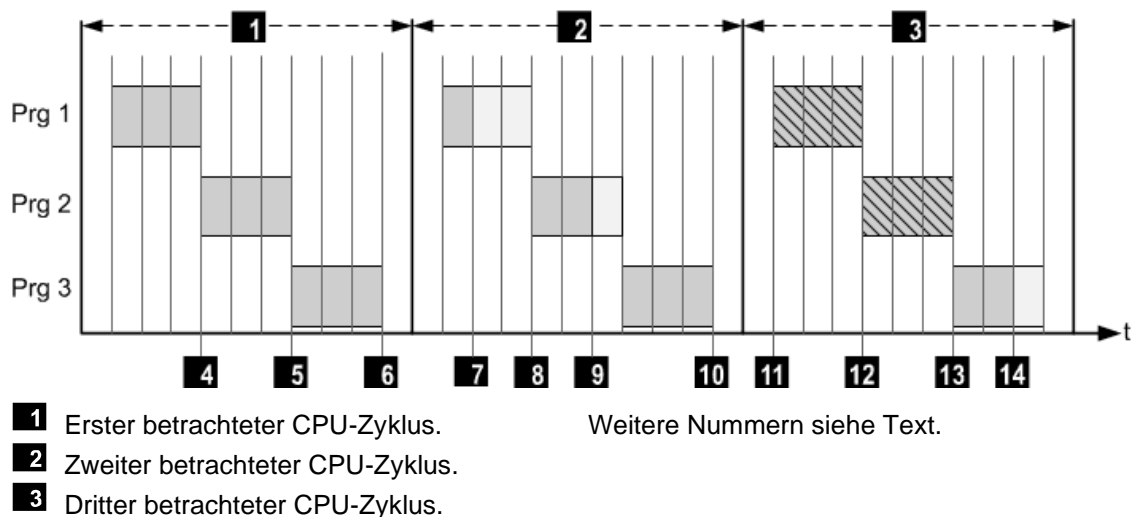


Bild 18: Multitasking Mode 3

Ablauf der Zyklen:

- 4 *Max. Dauer pro Zyklus [μs]* von *Prg 1* abgelaufen, *Prg 2* startet.
- 5 *Max. Dauer pro Zyklus [μs]* von *Prg 2* abgelaufen, *Prg 3* startet.
- 6 *Max. Dauer pro Zyklus [μs]* von *Prg 3* abgelaufen, erster CPU-Zyklus beendet. *Prg 1* wird fortgesetzt.
- 7 Anwenderprogramm-Zyklus von *Prg 1* beendet. Restliche Dauer wird gewartet.
- 8 *Max. Dauer pro Zyklus [μs]* von *Prg 1* abgelaufen, *Prg 2* wird fortgesetzt.
- 9 Anwenderprogramm-Zyklus von *Prg 2* beendet. Restliche Dauer wird gewartet.
- 10 *Max. Dauer pro Zyklus [μs]* von *Prg 3* beendet. Zweiter CPU-Zyklus beendet.
- 11 Nächster Anwenderprogramm-Zyklus von *Prg 1* beginnt.
- 12 *Max. Dauer pro Zyklus [μs]* von *Prg 1* abgelaufen, nächster Anwenderprogramm-Zyklus von *Prg 2* startet.
- 13 *Max. Dauer pro Zyklus [μs]* von *Prg 2* abgelaufen. *Prg 3* wird fortgesetzt.



- 14** Anwenderprogramm-Zyklus von *Prg 3* beendet. Wartezeit bis zum Ende von *Max. Dauer pro Zyklus [μs]* von *Prg 3*. Dritter CPU-Zyklus beendet.

**i**

In den Beispielen für die Multitasking Modes sind die Eingabe- und Ausgabeverarbeitung durch leere Bereiche am Anfang und Ende jedes CPU-Zyklus angedeutet.

Der Multitasking-Modus ist einstellbar durch den Parameter *Multitasking Mode* der Ressource, siehe Tabelle 10.

#### 5.4.4 Typische Reaktionszeit

Eine sicherheitsbezogene Reaktion des HIMax Systems muss in einer Zeit erfolgen, die kürzer ist als die vom Prozess vorgegebene Prozess-Sicherheitszeit. Die Prozess-Sicherheitszeit muss im Rahmen einer vom Betreiber durchgeführten Gefährdungs- und Risikoanalyse ermittelt worden sein. Die Prozess-Sicherheitszeit ist üblicherweise in der Sicherheitsanforderungsspezifikation dokumentiert.

Die hier betrachtete Reaktionszeit bezieht sich ausschließlich auf die HIMax Ressource!

Verzögerungszeiten wie Sensor- und Aktor-Schaltzeiten, die für die Reaktionszeit der kompletten Sicherheitsfunktion (SIF) zu beachten sind, aber außerhalb des HIMax Systems wirken, werden hier nicht betrachtet.

Die *typische Reaktionszeit* setzt sich aus den folgenden Zeiten zusammen:

- Zeitbedarf im Eingangsmodul:
  - E/A-Zykluszeit 2 ms.
  - Eingestellte Verzögerungszeit (TON, TOFF).
  - Zusätzliche Verzögerungen der Hardware (z. B. Einschwingzeit bei analogen Eingängen), siehe entsprechende Handbücher.
- Zeitbedarf im Prozessormodul:
  - Maximal 2 \* Zykluszeit
- Zeitbedarf im Ausgangsmodul:
  - 2 \* E/A-Zykluszeit

Der Zyklus eines Prozessormoduls umfasst in zeitlicher Reihenfolge die folgenden Aktivitäten:

Phase	Aktivität	Zeitbedarf hängt ab von
1	Selbsttests	
2	Abarbeitung der Konfigurationsverbindungen	Systembus-Aufbau, Anzahl Remote I/Os
3	Eingänge über Kommunikation lesen	Anzahl und Datenmenge der Kommunikationsverbindungen
4	Physikalische Eingänge lesen	Anzahl physikalischer Eingänge
5	Abarbeitung des Anwenderprogramms	Aufbau Anwenderprogramm und Größe.
6	Physikalische Ausgänge schreiben	Anzahl physikalischer Ausgänge
7	Ausgänge über Kommunikation schreiben	Anzahl und Datenmenge der Kommunikationsverbindungen
8	(Watchdog-Reservezeit: Restzeit bis zum Ablauf der Watchdog-Zeit)	(kein Zeitbedarf, Reserve für Reload, Synchronisierung)

Tabelle 24: Aufbau eines Zyklus des Prozessormoduls

Der Zeitbedarf für die Phase 2 ist durch die Einstellung des Parameters *Max. Dauer Konfigurationsverbindungen [ms]* vorgegeben, siehe Kapitel 5.2.4.5.

Der Zeitbedarf für die Phasen 3 und 7 ist durch die Einstellung des Parameters *Max. Kom.Zeitscheibe ASYNC [ms]* vorgegeben.

Die Voraussetzungen für die typische Reaktionszeit sind:

- Das System arbeitet störungsfrei.
- Keine Störaustastung ist wirksam.

Einzelheiten zu den Zeiten, die für eine HIMax Ressource zu betrachten sind, finden Sie im HIMax Sicherheitshandbuch HI 801 002 D.

Abhängig von den Bedingungen kann die Reaktionszeit entweder die doppelte CPU-Zykluszeit oder die Sicherheitszeit sein.

### Doppelte CPU-Zykluszeit als Reaktionszeit

In Anlagen, die im Low Demand Mode betrieben werden, kann die **doppelte CPU-Zykluszeit** als **Reaktionszeit** angenommen werden, wenn alle folgenden Bedingungen erfüllt werden.

Low Demand Mode bedeutet, dass die Sicherheitsfunktion einer Anlage höchstens einmal im Jahr angefordert wird.

Folgende Bedingungen müssen erfüllt sein:

- Die Anlage ist frei von regelmäßig auftretenden Fehlern, die zum Ansprechen der Störaustastung führen (Die Störaustastung kann die Reaktion auf einen erkannten Fehler bis zum Wert der Sicherheitszeit verzögern). Dazu muss eine wirksame Diagnose installiert sein und eine Reparatur innerhalb der in der PFD-Berechnung angenommenen Reparaturzeit gewährleistet sein.
- Das HIMax System ist frei von häufig auftretenden Lastspitzen, ausgelöst z. B. durch:
  - Anwenderprogramme: z. B. Verwendung von Enable-Eingängen (EN) bei Funktionsbausteinen oder von Programmschleifen.
  - Kommunikation (statische Kommunikationslast).
- Das HIMax System ist frei von Verzögerungen:
  - Programmierte Verzögerungen im Anwenderprogramm, z. B. TON.
  - Die Systembuslatenzzeit ist auf den Standardwert 0 eingestellt.
  - Die Abarbeitungsreihenfolge der Programmlogik ist so festgelegt, dass innerhalb eines Zyklus jeder Eingang gelesen wird, bevor Ausgänge beschrieben werden.
- Durch Anwendereingriffe ausgelöste Lastspitzen sind sehr selten:
  - Reload.
  - Synchronisierung von Prozessormodulen.
- Es gibt nur ein Anwenderprogramm (kein Multitasking).

Hinweis: Wenn das tatsächliche Zeitverhalten nicht bekannt ist oder häufige Lastspitzen (z. B. durch Reloads, Anwenderprogramme oder Kommunikation) nicht sicher ausgeschlossen werden können, ist von der **Watchdog-Zeit** als maximaler Zykluszeit auszugehen!

Ansonsten kann die tatsächliche, im Control Panel über eine längere Zeit beobachtete, Zykluszeit (Vor Beobachtung Statistik zurücksetzen!) angenommen werden.

Die Bedingung eines fehlerfreien Systems bleibt in jedem Falle bestehen!

### Sicherheitszeit als Reaktionszeit

Bei Anlagen, die im High Demand Mode oder Continuous Mode betrieben werden oder bei denen eine der oben genannten Bedingungen nicht erfüllt ist, kann die Reaktionszeit maximal die Sicherheitszeit sein.

High Demand Mode bedeutet, dass die Sicherheitsfunktion einer Anlage mehr als einmal im Jahr angefordert wird.

Für High Demand Mode sind weitere Betrachtungen notwendig. Diese Betrachtungen sind zusammen mit der hinzugezogenen Prüfstelle durchzuführen.

## Fazit

Von den genannten Randbedingungen hängt ab, welche der folgenden Zeiten für die Berechnung der Reaktionszeit angenommen werden kann:

- Die **tatsächlich beobachtete Zykluszeit**.
- Die **Watchdog-Zeit**.
- Die **Sicherheitszeit**.

Im Zweifel ist ein konservativer Ansatz zu wählen.

Die Entscheidung muss vor der für die Anlagenabnahme zuständigen Behörde gerechtfertigt werden.

## 5.5 Laden von Anwenderprogrammen

Mit SILworX lässt sich die Projektkonfiguration mit den Anwenderprogrammen in die Steuerung laden. Es gibt zwei Varianten des Ladens:

- **Download**  
Laden einer neuen Projektkonfiguration mit Unterbrechung des sicherheitsbezogenen Betriebs.
- **Reload**  
Laden einer geänderten Projektkonfiguration ohne Unterbrechung des sicherheitsbezogenen Betriebs.

---

### i

HIMA empfiehlt, nach jedem Laden eines Anwenderprogramms in die Steuerung die Projektkonfiguration zu sichern, z. B. auf einen Wechseldatenträger.

Damit soll gewährleistet werden, dass die zur Konfiguration auf der Steuerung passenden Projektdaten weiterhin verfügbar sind, auch wenn das PADT ausfällt.

HIMA empfiehlt eine regelmäßige Datensicherung auch unabhängig vom Laden des Anwenderprogramms.

---

### 5.5.1 Download

Voraussetzungen für Download:

- Steuerung im Zustand STOPP.
- Ressource-Freigabeschalter „Laden erlaubt“ gesetzt.

Nach dem Download das Anwenderprogramm durch SILworX starten, um den sicherheitsbezogenen Betrieb aufzunehmen.

Die Funktion Download ist zu benutzen, um ein neues Programm in eine Steuerung zu laden, oder wenn eine der im nächsten Abschnitt genannten Bedingungen die Verwendung des Reload ausschließt.

### 5.5.2 Reload

Voraussetzungen für Reload:

- Die Steuerung im Zustand RUN.
- Freigabeschalter „Reload erlaubt“ ist TRUE.
- Systemvariable „Reload-Deaktivierung“ ist FALSE.

i

Reload kann mit einem oder mehreren Prozessormodulen durchgeführt werden. Während eines Reloads ist kein weiterer Zugriff vom PADT auf die Steuerung möglich! Ausnahmen:

Ein Abbruch des Reloads ist möglich, ebenso die Änderung der Watchdog-Zeit und der Sollzykluszeit, um Reload zu ermöglichen.

HIMax erlaubt es, nach dem Ändern eines Anwenderprogramms, das bereits in einer Steuerung läuft, die geänderte Version mit Reload in die Steuerung zu laden. Während die alte Version des Anwenderprogramms noch läuft, wird die neue Version in den Speicher der Steuerung übertragen, geprüft und mit den Variablenwerten versorgt. Sind diese Vorbereitungen abgeschlossen, schaltet die Steuerung auf die neue Version des Anwenderprogramms um und führt den sicherheitsbezogenen Betrieb nahtlos fort.

Die globalen und lokalen Variablen erhalten beim Reload jeweils die Werte der gleichnamigen Variablen des vorhergehenden Projektstands. Die Namen lokaler Variablen enthalten den Instanznamen der POE.

Diese Vorgehensweise hat folgende Auswirkungen, wenn Namen geändert und mittels Reload auf die Steuerung geladen werden:

- Umbenennen einer Variablen wirkt wie Löschen und neu Einfügen, d. h. führt zum Initialisieren, auch bei Retain-Variablen. Dadurch verlieren diese ihren aktuellen Wert.
- Umbenennen einer Funktionsbaustein-Instanz führt zum Initialisieren aller Variablen, auch der Retain-Variablen, und aller enthaltenen Funktionsbaustein-Instanzen.
- Umbenennen eines Programms führt zum Initialisieren aller enthaltenen Variablen und Funktionsbaustein-Instanzen.

Ebenso wirkt sich das Verschieben von Teilen der Programmlogik von einem Anwenderprogramm in ein anderes wie ein Löschen und Einfügen aus.

**Dieses Verhalten kann unbeabsichtigte Auswirkungen auf ein oder mehrere Anwenderprogramme und damit auf die zu steuernde Anlage haben!**

Folgende Faktoren beschränken die Möglichkeit, ein geändertes Programm mit Reload auf die Steuerung zu laden:

- Die in Kapitel 5.5.2.1 beschriebenen Einschränkungen.
- Der Zeitbedarf für die Ausführung des Reload.

Da die zusätzlichen Aufgaben beim Reload Zeit benötigen, verlängert sich der Zyklus. Um zu verhindern, dass der Watchdog anspricht und die Steuerung in FEHLERSTOPP geht, prüfen sowohl SILworX als auch die Steuerung vor einem Reload den zusätzlichen Zeitbedarf. Ist dieser zu hoch, dann wird der Reload abgelehnt.

i

Bei Watchdog-Zeit und Sollzykluszeit ausreichende Zeitreserve für Reload einplanen. HIMA empfiehlt die im Sicherheitshandbuch HI 801 002 D angegebene Vorgehensweise zur Bestimmung der Watchdog-Zeit.

Es ist möglich, die Watchdog-Zeit und die Sollzykluszeit für die Dauer des Reloads zu erhöhen, näheres siehe die Online-Hilfe von SILworX. Dies kann dann erforderlich sein, wenn die Zeitreserve zu gering bemessen ist, so dass Reload in der Phase "Cleanup" blockiert.

Mit der Online-Funktion ist es nur möglich, die Watchdog-Zeit und Sollzykluszeit zu erhöhen, aber nicht, sie unter den im Projekt eingestellten Wert zu verkleinern.

---

**i****Beim Reload von Schrittketten zu beachten:**

Die Reload-Information für Schrittketten berücksichtigt nicht den aktuellen Status der Kette. Daher ist es möglich, durch Reload einer entsprechenden Änderung der Schrittkette diese in einen undefinierten Zustand zu versetzen. Die Verantwortung hierfür liegt beim Anwender.

Beispiele:

- Löschen des aktiven Schritts. Danach hat kein Schritt der Schrittkette den Zustand *aktiv*.
  - Umbenennen des Initialschritts, während ein anderer Schritt aktiv ist.  
Dies führt zu einer Schrittkette mit zwei aktiven Schritten!
- 

---

**i****Beim Reload von Actions zu beachten:**

Reload lädt Actions mit ihren kompletten Daten. Die Konsequenzen daraus sind vor dem Reload sorgfältig zu überdenken.

Beispiele:

- Entfernen eines Timer-Bestimmungszeichens durch den Reload führt dazu, dass der Timer sofort abgelaufen ist. Dadurch kann der Ausgang Q in Abhängigkeit von der restlichen Belegung auf TRUE gehen.
  - Entfernen des Bestimmungszeichens bei haftenden Elementen (z. B. Bestimmungszeichen S), die gesetzt waren, führt dazu, dass die Elemente gesetzt bleiben.
  - Entfernen eines Bestimmungszeichens *P0*, das TRUE gesetzt war, löst den Trigger aus.
- 

### 5.5.2.1 Bedingungen für die Verwendung von Reload

Die folgenden Projekt-Änderungen sind mit Reload in die Steuerung übertragbar:

- Änderungen an den Parametern des Anwenderprogramms.
- Änderungen an der Logik in Programm, Funktionsbausteinen, Funktionen.
- Änderungen, bei denen gemäß Tabelle 25 Reload möglich ist.

Änderungen bei	Art der Änderung			
	Hinzufügen	Löschen	Initialwert ändern	Andere Variable zuweisen
Zuweisungen globaler Variablen zu				
Anwenderprogrammen	•	•	•	•
Systemvariablen	•	•	•	•
E/A-Kanälen	•	•	•	•
Kommunikationsprotokollen <sup>3)</sup>	•	•	•	•
safeethernet <sup>1)</sup>	•	•	•	•
Rack mit Systembus- und E/A-Modulen	•	•	n. a.	n. a.
Modulen (E/A-, Systembus-, Prozessormodule)	•	•*	n. a.	n. a.
Kommunikationsprotokollen <sup>3)</sup>	•	•	n. a.	n. a.
Anwenderprogrammen	•	•**	n. a.	n. a.
Ereignisdefinitionen <sup>2)</sup>	•	•	n. a.	• (Ereignis-zustände)
Änderungen bei	Änderungen			
Namen von Racks	• <sup>3)</sup>			
Namen von Modulen	•, Systembusmodule und Kommunikationsmodule: • <sup>3)</sup>			
System-ID, Rack-ID	-			
safeethernet-Zieladressen (IP-Adressen)	• <sup>1)</sup>			
Benutzerkonten und Lizenzen	•			
Grenzen und Hysterese bei skalaren Ereignisdefinitionen	•			
Prozessormodulen: ▪ IP-Konfiguration ▪ Routings ▪ Switch-Konfiguration ▪ Kommunikationsprotokolle hinzufügen, ändern, löschen	• <sup>3)</sup>			
Systembusmodulen: ▪ IP-Konfiguration ▪ Routings ▪ Spannungsversorgung und Temperaturüberwachung ▪ Einstellung <i>Minimale Konfigurationsversion</i> ▪ Modulname	• <sup>3)</sup>			
Kommunikationsmodulen	Siehe Kommunikationshandbuch HI 801 100 D			
<div>• Reload möglich</div> <div>- Reload nicht möglich</div> <div>* Reload möglich, außer bei Systembusmodulen mit gesetztem Attribut <i>Responsible</i></div> <div>** Reload möglich, aber in der Steuerung muss mindestens ein Anwenderprogramm verbleiben.</div> <div>n. a. nicht anwendbar</div> <div>1) Einzelheiten zum Reload von Änderungen bei safeethernet im Kommunikationshandbuch HI 801 100 D</div> <div>2) Die Ereignisquelle einer Ereignisdefinition ist nicht per Reload änderbar, d. h., die Identifikationsnummer kann nicht durch Reload neu verwendet werden.</div> <div>3) Durch Cold Reload, d. h. mit Neustart des Moduls</div>				

Tabelle 25: Reload nach Änderungen

Reload ist nur nach Änderungen gemäß obigen Bedingungen möglich, andernfalls die Steuerung stoppen und Download verwenden.

### 5.5.2.2 Cold Reload

In bestimmten Fällen ist ein Reload für ein einzelnes Modul nicht durchführbar:

- Die in der Tabelle 25 mit «<sup>3)</sup>» gekennzeichneten Bedingungen.
- Konfigurationsänderungen von Kommunikationsmodulen und Systembusmodulen.
- Ein Kommunikationsmodul arbeitet mit nicht reloadfähigen Standardprotokollen.

In diesen Fällen wird Cold Reload angewendet. Das betreffende Modul muss während des Reloads in den Zustand STOPP gebracht werden («cold») und danach neu gestartet werden:

- Kommunikationsmodule und Systembusmodule kann der Reload-Vorgang selbstständig stoppen und neu starten.
- Bei Prozessormodulen zeigt der Reload-Vorgang eine Aufforderung an den Anwender an, das Modul zu stoppen und zu starten.

Der Anwender behält in jedem Fall die Kontrolle über den Reload-Vorgang und wird über dessen Ablauf informiert. Der Anwender kann den Vorgang gegebenenfalls abbrechen.

Vor der Durchführung eines Cold Reload ist folgendes zu bedenken:

- Welche Module werden gestoppt?
- Gibt es dafür redundante Module oder kann auf die von den Modulen erfüllten Aufgaben vorübergehend verzichtet werden?

---

**TIPP**

Auf folgende Weise lässt sich Cold Reload in Fällen vermeiden, in denen Zuweisungen globaler Variablen hinzugefügt werden:

- Bereits beim Erstellen der Konfiguration den Standardprotokollen unbenutzte globale Variable zuweisen.
- Den unbenutzten globalen Variablen einen sicheren Wert als Initialwert zuweisen.

Auf diese Weise ist es später nur notwendig, die Namen der Variablen zu ändern, und keine Variablen hinzuzufügen, so dass ein Reload möglich ist.

---

### 5.5.2.3 Grenzen für Reload

Die folgenden Bedingungen können dazu führen, dass Reload eines Anwenderprogramms nicht möglich ist:

- Die Anzahl der Instanzen von Funktionen und Funktionsbausteinen (POEs) in SILworX ist auf 21 845 beschränkt. Ein Anwenderprogramm, das mehr Instanzen enthält, kann nicht durch Reload geladen werden.
- Große Arrays, deren Elemente benutzerdefinierte Strukturen sind, können bei Änderung des Aufbaus der Struktur möglicherweise nicht mehr durch Reload geladen werden.

Beispiel: Bei einem Struktur-Datentyp ein Strukturelement hinzufügen oder löschen.

In beiden Fällen zeigt SILworX an, dass die maximale Anzahl Transferoperationen (65 536) überschritten wird.

#### Abhilfe

Es gibt folgende Möglichkeiten, die Anzahl Transferoperationen beim Reload zu verringern:

- Vermeiden von zu starker Strukturierung des Programms, d. h. Verringerung der Anzahl POEs.
- Vermeiden von großen Struktur-Datentypen und von großen Arrays von Strukturen.
- Auf einfache Weise kann dadurch Abhilfe geschaffen werden, dass in den POEs mit vielen Instanzen auf VAR verzichtet wird.

Ersetzen aller VAR einer POE durch VAR\_OUTPUT reduziert die Anzahl der Transferoperationen, hat aber folgende Auswirkungen:

- Die Anzahl der Ausgänge der POE erhöht sich.
- Forcen der Variablen ist nicht möglich.

## 5.6 Benutzerverwaltung

Die Benutzerverwaltung dient der projektbezogenen Verwaltung von Benutzergruppen, Benutzerkonten, Zugriffsrechten für das PADT und Zugriffsrechten für die Steuerungen (PES).

Eine Benutzergruppe besteht aus ein oder mehreren Benutzerkonten. Das Öffnen von Projekten in SILworX kann eine Anmeldung mit den Daten eines Benutzerkontos erfordern. Zusätzlich kann der Benutzergruppe der Zugriff auf Ressourcen erlaubt werden.

Sie können in einem Projekt genau eine Benutzerverwaltung anlegen. Die Benutzerverwaltung gilt für dieses Projekt und wird im Projekt verschlüsselt gespeichert. Der Name *Benutzerverwaltung* wird von SILworX vorgegeben und ist sprachabhängig.

Eine Benutzergruppe ist eine Sammlung von Benutzern mit identischen Zugriffsrechten für die PADT-Bedienung oder für die Zugriffe auf Steuerungen.

PADT-Benutzerverwaltung: Regelt den Zugang zum SILworX Projekt.

PES-Benutzerverwaltung: Verwaltet die PES-bezogenen Zugriffsrechte. Die PES-Benutzerverwaltung basiert auf den Daten der PADT-Benutzerverwaltung.



**⚠ WARNUNG**

Bei nicht eingerichteter Benutzerverwaltung sind Manipulationen jeglicher Art am Projekt oder der Ressource möglich!

Für den sicheren Betrieb einer Anlage muss ein Projekt eine Benutzerverwaltung mit mehreren Benutzergruppen und unterschiedlichen Zugriffsarten enthalten. In den Benutzergruppen müssen mehrere Benutzerkonten definiert sein. Anderenfalls kann jeder das Projekt in SILworX öffnen und sich als Standardbenutzer in Ressourcen einloggen.

Ein Projekt wird ohne Login-Dialog geöffnet, wenn keine Benutzerverwaltung vorhanden ist. Ansonsten wird ein Dialogfenster zur PADT-Benutzerabfrage angezeigt, und es erfolgt eine Aufforderung einen Benutzernamen und ein Passwort einzugeben.

Falls das Projekt mit einer früheren Version von SILworX erstellt wurde und eine Projektkonvertierung erforderlich ist, wird diese nur ausgeführt, wenn der angemeldete Benutzer mindestens über Schreibrechte verfügt. Andernfalls wird das Projekt nicht konvertiert und nicht geöffnet.

Mit dem Anlegen der Benutzerverwaltung wird automatisch auch ein erster Sicherheitsadministrator zur PADT-Benutzerverwaltung hinzugefügt und parametrisiert. Danach können Änderungen an der Benutzerverwaltung nur noch von diesem oder einem anderen Sicherheitsadministrator durchgeführt werden.

### 5.6.1 Standard-Zugriffsrechte

Solange für ein Projekt keine Benutzerverwaltung angelegt wurde, wird im Strukturbaum kein Element *Benutzerverwaltung* angezeigt. Sowohl für das in SILworX geöffnete Projekt, als auch für die Ressourcen gelten die werkseitig festgelegten Zugriffsrechte.

	PADT	Ressource (Steuerung)
Anzahl der Benutzer	1	1
Benutzerkennung	Sicherheitsadministrator	Administrator
Passwort	Ohne	Ohne
Zugriffsrecht	Sicherheitsadministrator	Administrator

Tabelle 26: Werkseitige Zugriffsrechte für PADT und PES

---

**i** Die werkseitig festgelegten Zugriffsrechte werden außer Kraft gesetzt, wenn im Projekt eine Benutzerverwaltung angelegt wird.

---

#### 5.6.1.1 Benutzergruppe «Sicherheitsadministratoren»

Sicherheitsadministratoren verfügen über alle Zugriffsrechte. Sobald eine Benutzerverwaltung angelegt wird, erstellt SILworX automatisch eine Benutzergruppe mit der Zugriffsart *Sicherheitsadministrator*. Diese Benutzergruppe enthält standardmäßig ein (1) Benutzerkonto, weitere Benutzerkonten können hinzugefügt werden.

Damit ist sichergestellt, dass in der Benutzerverwaltung immer mindestens ein Sicherheitsadministrator existiert.

Sowohl das standardmäßig erstellte Benutzerkonto, als auch alle später hinzugefügten Benutzerkonten dieser Benutzergruppe, verfügen über die Zugriffsart *Sicherheitsadministrator*.

Diese Zugriffsart gilt nur für die Bearbeitung des Projekts in SILworX. In der *PES-Benutzerverwaltung* kann die Benutzergruppe der Sicherheitsadministratoren auch eine andere Zugriffsart mit reduzierten Privilegien haben.

---

**i** Sicherheitsadministratoren haben vollen Zugriff auf alle Benutzerkonten. Sie können z. B. auch deren Passworte ändern, ohne das aktuelle Passwort des Benutzerkontos zu kennen. Dies kann erforderlich werden, wenn ein Benutzer sein Passwort vergessen hat.

---

### 5.6.2 Zugriffsarten und Berechtigungen

Abhängig von der definierten Zugriffsart und den vorhandenen Lizenzen hat ein Anwender mehr oder weniger Berechtigungen, Funktionen in SILworX oder auf einer Ressource auszuführen.

Unabhängig vom angemeldeten Benutzer können die Zugriffsrechte auch über Lizenzen eingeschränkt werden. Hierzu gibt es die Lizenz-Option *Wartung*, welche die wirksamen Rechte für jeden Benutzer auf *Lesen* reduziert.

### 5.6.2.1 Zugriffsarten der PADT-Benutzerverwaltung

In der nachfolgenden Tabelle sind die Zugriffsarten zusammengefasst, die in der PADT-Benutzerverwaltung verwendet werden können.

Zugriffsart	Beschreibung
Sicherheitsadministrator	Mit der Zugriffsart <i>Sicherheitsadministrator</i> kann der Anwender alle Funktionen in SILworX ausführen, die von den vorhandenen Lizenzen abgedeckt sind.
Lesen + Schreiben	Mit der Zugriffsart <i>Lesen + Schreiben</i> kann der Anwender alle Funktionen in SILworX ausführen, die nicht explizit an das Recht <i>Sicherheitsadministrator</i> gebunden und von den vorhandenen Lizenzen abgedeckt sind.
Lesen	Anwender mit dieser Zugriffsart können auf Funktionen nur lesend zugreifen. So ist z. B. auch Archivieren nicht erlaubt. Manche Informationen werden nicht angezeigt (z. B. Details der Benutzerverwaltung).

Tabelle 27: Zugriffsarten der PADT-Benutzerverwaltung

### 5.6.2.2 Zugriffsarten der PES-Benutzerverwaltung

In der nachfolgenden Tabelle sind die Zugriffsarten zusammengefasst, die in der PES-Benutzerverwaltung verwendet werden können. An einer Steuerung können maximal 5 PES-Benutzer gleichzeitig eingeloggt sein.

Zugriffsart	Beschreibung
Administrator	<p><b>Tätigkeit:</b> Neues System in Betrieb nehmen.</p> <p>Administratoren verfügen im Geltungsbereich der Ressourcen über alle Berechtigungen, die von den vorhandenen Lizenzen abgedeckt sind.</p> <ul style="list-style-type: none"> <li>- Aktionen, welche erforderlich sind, um Module oder Systeme in Betrieb zu nehmen.</li> <li>- Einrichten einer ersetzten Steuerung.</li> <li>- Inbetriebnahme von Sicherheitseinrichtungen.</li> </ul>
Lesen + Schreiben	<p><b>Tätigkeit:</b> Ein bestehendes System ändern.</p> <p>Mit der Zugriffsart <i>Lesen + Schreiben</i> kann der Anwender alle Funktionen auf einer Ressource ausführen, die erforderlich sind, um Änderungen durchzuführen und die von den vorhandenen Lizenzen abgedeckt sind.</p> <p>Anwender können die zum Verändern freigegebenen Online-Einstellungen anpassen, jedoch nicht die Einstellung des Parameter <i>Online-Einstellungen erlauben</i> selbst.</p>
Lesen + Bediener	<p><b>Tätigkeit:</b> Ein laufendes System in Betrieb halten.</p> <p>Mit der Zugriffsart <i>Lesen + Bediener</i> kann der Anwender alle Aktionen durchführen, die erforderlich sind, um ein laufendes System zu warten. Änderungen an den programmierten Funktionen sind nicht möglich.</p> <p>Die Berechtigungen sind gegenüber <i>Lesen + Schreiben</i> nochmals reduziert und beschränken sich im Wesentlichen auf folgende Tätigkeiten:</p> <ul style="list-style-type: none"> <li>- Alle Aktionen, welche zur Diagnose erforderlich sind.</li> <li>- Aktionen, die erforderlich sind, um ein System zu starten und zu forcen.</li> </ul>
MultiForcen	<p><b>Tätigkeit:</b> Forcen durch mehrere Benutzer gleichzeitig.</p> <p>Anwender mit <i>MultiForcen</i>-Zugriff können in einer Ressource Force-Daten (Force-Werte und Force-Einzelschalter) für globale Variablen schreiben, wenn die dafür erforderlichen, übergeordneten Bedingungen gegeben und Force-Freigaben erfolgt sind. Auch das Verwenden vorkonfigurierter Watchpages ist möglich. Auf alle anderen Funktionen einer Ressource können Anwender nur lesend zugreifen. Das Starten, Stoppen oder Zurücksetzen eines Force-Vorgangs ist nicht möglich.</p> <p>Damit Anwender Watchpages verändern und im Projekt speichern können, benötigen sie im PADT die Zugriffsrechte (<i>Lesen + Schreiben</i> oder <i>Sicherheitsadministrator</i>).</p>
Lesen	<p><b>Tätigkeit:</b> Lesen der Diagnose.</p> <p>Anwender mit dieser Zugriffsart können ausschließlich lesend auf Funktionen zugreifen.</p> <p>Die Zugriffsart <i>Lesen</i> wird bei einem System-Login in eine Ressource automatisch vergeben, wenn bereits ein anderer Benutzer mit Schreibrechten eingeloggt ist.</p>
Kein Zugriff	<p><b>Tätigkeit:</b> Keine, das Einloggen ist gesperrt.</p> <p>Dies ist die Standardeinstellung für alle Benutzergruppen, für die in der Zuordnungstabelle nicht explizit eine Zugriffsart eingestellt wurde.</p> <p>Für <i>Kein Zugriff</i> wird eine leere Tabellenzelle angezeigt.</p>

Tabelle 28: Zugriffsarten der PES-Benutzerverwaltung

### 5.6.3 PADT-Benutzerverwaltung anlegen

Das Anlegen der PADT-Benutzerverwaltung ist in der SILworX Online-Hilfe beschrieben.

### 5.6.4 PES-Benutzerverwaltung anlegen

Die PES-Benutzerverwaltung zeigt die Benutzergruppen aus der PADT-Benutzerverwaltung automatisch an. Den Ressourcen können in der PES-Benutzerverwaltung Zugriffsrechte für die angelegten Benutzergruppen zugewiesen werden, siehe SILworX Online-Hilfe.

Die PES-Benutzerverwaltung kann Benutzerkonten für maximal 10 Benutzergruppen pro Steuerung verwalten. Sobald das Projekt per Download auf die Steuerung übertragen wurde, stehen diese Informationen für Logins zur Verfügung. Die Benutzerkonten sind in der Steuerung abgelegt und bleiben auch nach dem Ausschalten der Betriebsspannung erhalten. Die Benutzerkonten einer Steuerung gelten auch für angeschlossenen Remote I/Os.

Die Benutzer identifizieren sich beim Login auf der Steuerung mit der Benutzergruppe und dem Passwort der Benutzergruppe. Für die PES-Benutzergruppe und das Passwort sind die mit dem letzten Laden auf das PES übertragenen Werte gültig, nicht die aktuell im Projekt eingetragenen Werte!

## 5.7 Application Programming Interface (API)

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:

- Projekte erzeugen.
- Projekte öffnen, auch solche mit Benutzerverwaltung.
- Projekte schließen.
- SILworX schließen.
- SILworX Status auslesen, z. B. die Version und die Lizenzen.
- Knoten im *Projekt*-Strukturbaum erstellen.
- Strukturbaum auslesen.
- Im Strukturbaum Validierungen vornehmen.
- Projekte oder individuelle Strukturbaum-Knoten archivieren.
- Projekte oder individuelle Strukturbaum-Knoten wiederherstellen.
- Code generieren, für Download oder Reload).
- Ressourcen verbinden und trennen.
- Modul-Login durchführen.
- Ressourcen starten und stoppen.
- CPU-Systembetrieb starten und stoppen, Module starten und stoppen.
- Konfigurationen per Download in Ressourcen laden.
- Online-Werte lesen auf System- und Modulebene, einzeln oder gruppiert.
- Diagnoseeinträge von Modulen abfragen.
- Betriebssysteme per Downloads laden.
- Globales Forcen starten und stoppen.
- Globale Force-Variablen lesen und schreiben.

---

**i**

### **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 HTLM-Format und ein C# Anwendungsbeispiel.

## 6 Diagnose

Diagnose-LEDs verschaffen einen ersten, schnellen Überblick über den Systemzustand. Detailliertere Informationen beim Auslesen der Diagnosehistorie mit SILworX.

### 6.1 Leuchtdioden (LEDs)

LEDs auf der Frontseite zeigen den Zustand eines Moduls an. Dabei sind alle LEDs im Zusammenhang zu betrachten. Eine einzelne LED reicht für die Beurteilung des Modulzustandes nicht unbedingt aus.

Die Bedeutung der LEDs ist in den Modulhandbüchern beschrieben.

Nach dem Zuschalten der Versorgungsspannung erfolgt immer ein LED-Test, bei dem alle LEDs für mindestens 2 s leuchten. Bei zweifarbigen LEDs erfolgt während des Tests einmalig ein Farbwechsel.

#### Definition der Blinkfrequenzen

In der folgenden Tabelle sind die Blinkfrequenzen definiert:

Definition	Blinkfrequenz
Blinken1	Lang (600 ms) an, lang (600 ms) aus.
Blinken2	Kurz (200 ms) an, kurz (200 ms) aus, kurz (200 ms) an, lang (600 ms) aus.
Blinken-x	Ethernet-Kommunikation: Aufblitzen im Takt der Datenübertragung.

Tabelle 29: Blinkfrequenzen der LEDs

Einige LEDs signalisieren Warnungen (Ein) und Fehler (Blinken1). Die Anzeige von Fehlern hat Priorität gegenüber der Anzeige von Warnungen. Bei der Anzeige von Fehlern können Warnungen nicht angezeigt werden.

### 6.2 Diagnosehistorie

Jedes Modul des HIMax Systems führt über die aufgetretenen Störungen und Ereignisse eine Historie in chronologischer Reihenfolge. Die Historie ist als Ringspeicher organisiert.

Die Diagnosehistorie besteht aus einer Kurzzeit- und einer Langzeitdiagnose:

- Kurzzeitdiagnose:  
Wenn die maximale Anzahl der Einträge erreicht ist, wird für jeden neuen Eintrag der älteste Eintrag gelöscht.
- Langzeitdiagnose:  
Die Langzeitdiagnose speichert hauptsächlich Aktionen und Konfigurationsänderungen des Anwenders.  
Wenn die maximale Anzahl der Einträge erreicht ist, wird für jeden neuen Eintrag der älteste Eintrag nur dann gelöscht, wenn er älter ist als drei Tage.  
Gibt es nur Einträge, die jünger als drei Tage sind, dann wird der neue Eintrag verworfen.  
Ein besonderer Eintrag kennzeichnet das Verwerfen.

Die Anzahl der Ereignisse, die gespeichert werden können, hängt vom Typ des Moduls ab:

Modultyp	Max. Anzahl Ereignisse Langzeitdiagnose	Max. Anzahl Ereignisse Kurzzeitdiagnose
X-CPU 01 und X-CPU 31	2500	1500
X-COM 01	300	700
E/A-Module	400	500
X-SB 01	400	500

Tabelle 30: In der Diagnosehistorie maximal gespeicherte Einträge pro Modultyp

## i

Bei Stromausfall kann es vorkommen, dass Diagnoseeinträge verloren gehen, wenn sie gerade noch nicht im nicht-flüchtigen Speicher abgelegt sind.

Es ist möglich, die Historien der einzelnen Module mit SILworX auszulesen und so darzustellen, dass die zur Analyse eines Problems notwendigen Informationen zur Verfügung stehen:

Beispiel:

- Mischen der Historien aus verschiedenen Quellen.
- Filtern nach Zeitbereich.
- Ausdrucken der bearbeiteten Historie.
- Abspeichern der bearbeiteten Historie.

Weitere Funktionen sind in der Online-Hilfe von SILworX beschrieben.

### 6.2.1 Diagnosemeldung von E/A-Modulen

Eine Diagnosemeldung für ein E/A-Modul ist folgendermaßen aufgebaut:

IO-Fehler >> Steckplatz S E/A-Modultyp MMMM Status[Bg: mm Aus: AAAA Ein: EEEE]  
Kanal[Aus:aaaa Ein:eeee] <<

Die folgende Tabelle beschreibt die Bedeutung der Datenfelder innerhalb der Meldung.

Datenfeld	Format	Beschreibung
S	Dezimal	Steckplatznummer des E/A-Moduls
MMMM	Hexadezimal	Modultyp
mm	Hexadezimal	Status des Moduls
AAAA	Hexadezimal	Fehlercode für Fehler bei Ausgängen des Moduls
EEEE	Hexadezimal	Fehlercode für Fehler bei Eingängen des Moduls
aaaa	Hexadezimal	Fehlercode für Kanalfehler bei Ausgangskanälen
eeee	Hexadezimal	Fehlercode für Kanalfehler bei Eingangskanälen

Tabelle 31: Datenfelder einer Diagnosemeldung

Die Einzelheiten der Fehlercodes sind den jeweiligen Handbüchern zu entnehmen. Sind mehrere Kanäle fehlerhaft, enthält das Datenfeld aaaa/eeee eine ODER-Verknüpfung mit 0x8000, d. h., zusätzlich zum Fehlercode ist das höchstwertige Bit auf 1 gesetzt.

Der Modultyp kann im Hardware-Editor ermittelt werden.



### 6.3 Online-Diagnose

Die Online-Ansicht des SILworX Hardware-Editors dient zur Diagnose von Störungen der HIMax Module. Gestörte Module sind durch einen Farbumschlag gekennzeichnet:

- Rot kennzeichnet schwere Störungen, z. B. Modul nicht gesteckt.
- Gelb kennzeichnet weniger schwere Störungen, z. B. Temperaturgrenze überschritten.

Beim Positionieren der Maus auf einem Modul zeigt SILworX einen Tooltip an, der die folgenden Zustandsinformationen über das Modul enthält:

Information	Darstellung	Wertebereich	Beschreibung										
SRS:	Dezimal- zahlen	System: 0 ... 65 535 Rack: 0 ... 16 Slot: 1 ... 18	Identifikation des Moduls (System, Rack, Slot).										
Name:	Text		Bezeichnung der Information, hier immer : <i>Online-MODUL-Information</i> .										
Zustand des Moduls:	Text	RUN, STOPP, NICHT VERBUNDEN, Unknown, ...	Betriebszustand in dem sich das Modul befindet.										
Gesteckter Modultyp:	Text	Zulässige Modultypen	Typ des Moduls, welches im Rack gesteckt ist.										
Konfigurierter Modultyp:	Text	Zulässige Modultypen	Typ des Moduls, welches in der Steuerung konfiguriert und geladenen ist.										
Modultyp im Projekt:	Text	Zulässige Modultypen	Typ des Moduls, welches als digitales Abbild projiziert wurde.										
Verbindungsstatus des Protokolls:	Hexadezimal-Wert	16#00 ... 0F	Status der Verbindung zwischen jedem der maximal 4 Prozessormodule. Jedes der Bits 0 ... 3 zeigt die Verbindung zum Prozessormodul mit dem korrespondierenden Index an. Bit x = 0: Nicht verbunden. Bit x = 1: Verbunden.										
Sendestatus der Schnittstelle:	Hexadezimal-Wert	16#0000 ... FF FF	Je zwei Bits stellen den Zustand einer Schnittstelle dar, die durch einen Index 0 ... 16 identifiziert ist. Bits 0 und 1 gelten für Schnittstelle 0, usw.										
			<table><tr><th>Wert</th><th>Beschreibung</th></tr><tr><td>00</td><td>Noch keine Meldung empfangen/gesendet, Status unbekannt.</td></tr><tr><td>01</td><td>OK, keine Fehler.</td></tr><tr><td>10</td><td>Letzte/r Empfang/Sendung war fehlerhaft.</td></tr><tr><td>11</td><td>Letzte/r Empfang/Sendung war fehlerfrei, davor ist ein Fehler aufgetreten.</td></tr></table>	Wert	Beschreibung	00	Noch keine Meldung empfangen/gesendet, Status unbekannt.	01	OK, keine Fehler.	10	Letzte/r Empfang/Sendung war fehlerhaft.	11	Letzte/r Empfang/Sendung war fehlerfrei, davor ist ein Fehler aufgetreten.
			Wert	Beschreibung									
			00	Noch keine Meldung empfangen/gesendet, Status unbekannt.									
			01	OK, keine Fehler.									
			10	Letzte/r Empfang/Sendung war fehlerhaft.									
11	Letzte/r Empfang/Sendung war fehlerfrei, davor ist ein Fehler aufgetreten.												
Empfangsstatus der Schnittstelle:	Hexadezimal-Wert	16#0000 ... FF FF	Je zwei Bits stellen den Zustand einer Schnittstelle dar, die durch einen Index 0 ... 16 identifiziert ist. Bits 0 und 1 gelten für Schnittstelle 0, usw.										
			<table><tr><th>Wert</th><th>Beschreibung</th></tr><tr><td>00</td><td>Noch keine Meldung empfangen/gesendet, Status unbekannt.</td></tr><tr><td>01</td><td>OK, keine Fehler.</td></tr><tr><td>10</td><td>Letzte/r Empfang/Sendung war fehlerhaft.</td></tr><tr><td>11</td><td>Letzte/r Empfang/Sendung war fehlerfrei, davor ist ein Fehler aufgetreten.</td></tr></table>	Wert	Beschreibung	00	Noch keine Meldung empfangen/gesendet, Status unbekannt.	01	OK, keine Fehler.	10	Letzte/r Empfang/Sendung war fehlerhaft.	11	Letzte/r Empfang/Sendung war fehlerfrei, davor ist ein Fehler aufgetreten.
			Wert	Beschreibung									
			00	Noch keine Meldung empfangen/gesendet, Status unbekannt.									
			01	OK, keine Fehler.									
			10	Letzte/r Empfang/Sendung war fehlerhaft.									
11	Letzte/r Empfang/Sendung war fehlerfrei, davor ist ein Fehler aufgetreten.												

Information	Darstellung	Wertebereich	Beschreibung	
Fehlerstatus des Moduls	Hexadezimal-Wert	16#00 ... 3F	Bitcodierter Status des Moduls:	
			Bit	Beschreibung bei Wert = 1
			0	Warnung bei externer Kommunikation.
			1	Warnung bei Feldanschluss.
			2	Systemwarnung.
			3	Fehler bei externer Kommunikation.
			4	Fehler bei Feldanschluss.
			5	Systemfehler.
			6	Nicht benutzt.
7				
SBA-Verbindungsstatus	Hexadezimal-Wert	16#0 ... 3	Status der Schnittstelle zu Systembus A:	
			Wert	Beschreibung
			0	Die Schnittstelle ist OK
			1	Die Schnittstelle hat beim letzten Empfang einen Fehler erkannt, ist jetzt OK
			2	Die Schnittstelle hat einen Fehler
			3	Die Schnittstelle ist abgeschaltet
SBB-Verbindungsstatus	Hexadezimal-Wert	16#0 ... 3	Status der Schnittstelle zu Systembus B:	
			Wert	Beschreibung
			0	Die Schnittstelle ist OK
			1	Die Schnittstelle hat beim letzten Empfang einen Fehler erkannt, ist jetzt OK
			2	Die Schnittstelle hat einen Fehler
			3	Die Schnittstelle ist abgeschaltet

Tabelle 32: Diagnoseinformationen in der Online-Ansicht des Hardware-Editors

Das Zeitverhalten des Betriebssystems kann dazu führen, dass die Diagnose keine Sicherheitsparameter anzeigt. Die folgende Vorgehensweise beim Öffnen der Diagnose führt dazu, dass die Diagnose die Sicherheitsparameter mit anzeigt:

1. Control Panel öffnen und abwarten, bis darin alle Anzeigefelder aktualisiert sind.
2. Die Diagnose über das Kontextmenü in der Online-Ansicht des Hardware-Editors öffnen, nicht über eine Detailansicht!

Vor dem Öffnen der Diagnose keine Detailansicht und möglichst wenige Online-Ansichten (z. B. Force-Editor, Online-Test) öffnen!

## 7 Produktdaten und Dimensionierung

Das Kapitel gibt die Umgebungsbedingungen und die Dimensionierung an.

### 7.1 Umgebungsbedingungen

Andere als die genannten Umgebungsbedingungen können zu Betriebsstörungen des HIMax Systems führen. Zusätzlich sind die Angaben in den Modulhandbüchern zu beachten:

Allgemein	
Schutzklasse	Schutzklasse II nach IEC/EN 61131-2
Umgebungstemperatur	0 ... +60 °C
Transport- und Lagertemperatur	-40 ... +70 °C
Verschmutzung	Verschmutzungsgrad II nach IEC/EN 60664-1
Aufstellhöhe	< 2000 m
Gehäuse	Standard: IP20 Falls es die zutreffenden Applikationsnormen (z. B. EN 60204) fordern, muss das HIMax System in ein Gehäuse der geforderten Schutzart (z. B. IP54) eingebaut werden.
Eingangsspannung Netzteil	24 VDC, -15 ... +20 %, $w_s \leq 5\%$ , SELV, PELV

Tabelle 33: Umgebungsbedingungen

Andere als die genannten Umgebungsbedingungen können zu Betriebsstörungen des HIMax Systems führen.

### 7.2 Dimensionierung

Detaillierte technische Daten sind den Handbüchern der einzelnen Komponenten und dem Kommunikationshandbuch HI 801 100 D zu entnehmen.

Je Ressource:	Wertebereich von ... bis
Anzahl Racks	1 ... 16
Anzahl E/A-Module	
Bei X-CPU 01	0 ... 200
Bei X-CPU 31	0 ... 64
Anzahl E/A-Punkte (Sensoren, Aktoren)	Abhängig vom Modultyp, hier für Module mit 32 Eingängen oder Ausgängen:
Bei X-CPU 01	0 ... 6400
Bei X-CPU 31	0 ... 2048
Maximale Länge des Systemkabels zum FTA	30 m
Anzahl Prozessormodule	
X-CPU 01	1 ... 4
X-CPU 31	1 ... 2
Gesamter Programm- und Datenspeicher für alle Anwenderprogramme	
X-CPU 01	10 MB, abzügl. 4 kB für CRCs
X-CPU 31	5 MB, abzügl. 4 kB für CRCs
Speicher für Retain-Variable	32 kB
Für globale Prozessdaten verfügbarer Speicher	512 kB
Anzahl Systembusmodule je Rack	1 ... 2

Je Ressource:	Wertebereich von ... bis
Maximale Länge der Systembusse bei Standard-Einstellung der Latenzzeit	1500 m
unter Verwendung von Lichtwellenleitern (siehe Kapitel 3.2)	19,6 km
Bei erhöhter Latenzzeit sind größere Längen möglich, siehe Kap. 3.2.3	
Anzahl Kommunikationsmodule	
Bei X-CPU 01	0 ... 20
Bei X-CPU 31	0 ... 4
Anzahl <b>safeethernet</b> Verbindungen	0 ... 255
Anzahl <b>safeethernet</b> Verbindungen zwischen zwei Ressourcen folgender Typen: <ul style="list-style-type: none"> <li>▪ HIMax</li> <li>▪ HIMatrix F30 03, F35 03 oder F60 mit CPU 03 mit CPU-Betriebssystem ab V10 und COM-Betriebssystem ab V15</li> </ul>	0 ... 64
<b>safeethernet</b> Puffergröße pro Verbindung <ul style="list-style-type: none"> <li>▪ Verbindung mit anderer HIMax Steuerung</li> <li>▪ Mit HIMatrix Steuerungen F30 03, F35 03, F60 CPU 03</li> </ul>	1100 Bytes je Richtung
Verbindung mit sonstiger HIMatrix Steuerung	900 Bytes je Richtung
Puffergröße für Verbindung zum OPC-Server	128 kBytes
Anzahl PES-Benutzergruppen	1 ... 10
Anzahl Anwenderprogramme	1 ... 32
Anzahl Ereignisdefinitionen	0 ... 20 000
Größe des nichtflüchtigen Ereignispuffers	5000 Ereignisse

Tabelle 34: Dimensionierung einer HIMax Steuerung

Detaillierte technische Daten in den Handbüchern der einzelnen Komponenten und im Kommunikationshandbuch HI 801 100 D.

## 8 Lebenszyklus

Dieses Kapitel beschreibt die folgenden Phasen des Lebenszyklus:

- Installation.
- Inbetriebnahme.
- Wartung und Instandhaltung.

Hinweise zu Außerbetriebnahme und Entsorgung in den Handbüchern der einzelnen Komponenten.

### 8.1 Installation

Dieses Kapitel beschreibt den Einbau und den Anschluss der HIMax Steuerungen.

#### 8.1.1 Mechanischer Aufbau

Bei der Wahl des Montageplatzes für das HIMax System Verwendungsbedingungen beachten (siehe Kapitel 2.1), damit ein störungsfreier Betrieb sichergestellt werden kann.

Hinweise zur Montage von Basisträgern und anderen Komponenten in den jeweiligen Handbüchern beachten.

#### 8.1.2 Anschluss der Feldebene an E/A-Module

Das HIMax System ist ein flexibles und auf Dauerbetrieb ausgelegtes System. Es erlaubt folgende Möglichkeiten, die Feldebene an die E/A-Module anzuschließen:

- Direkt an das Connector Board.
- Indirekt über Field Termination Assemblies.

Nachfolgend werden die vier empfohlenen Beschaltungen beschrieben:

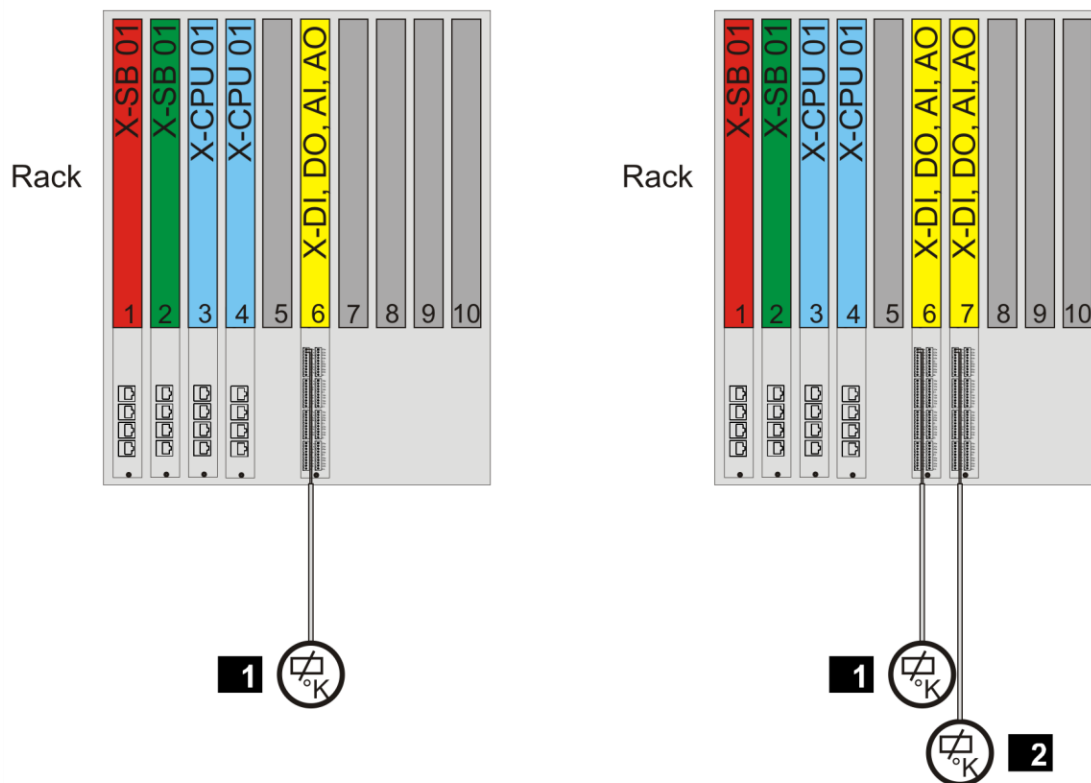
1. Anschluss an Mono-Connector Boards mit Schraubklemmen.
2. Anschluss an redundante Connector Boards mit Schraubklemmen.
3. Anschluss über Field Termination Assembly und Systemkabel an Mono-Connector Boards.
4. Anschluss über Field Termination Assembly und Systemkabel an redundante Connector Boards.

Andere Beschaltungen sind mit einem höheren Planungsaufwand realisierbar und nicht in den Handbüchern beschrieben. HIMA empfiehlt, sich im Bedarfsfall mit HIMA, Abteilung Global Project Management and Engineering, in Verbindung zu setzen.

## 8.1.2.1 Beschaltung 1

Sensoren/Aktoren an ein Mono-Connector Board mit Schraubklemmen anschließen für ein einzelnes E/A-Modul.

- Einzelne Sensoren/Aktoren kanalweise an ein einzelnes E/A-Modul (nicht redundant) anschließen.
- Zwei oder mehr redundante Sensoren/Aktoren kanalweise an zwei oder mehr redundante Module anschließen. Die Anzahl der redundanten Sensoren/Aktoren muss gleich der Anzahl der redundanten Module sein (z. B. 2 Sensoren/2 Module).



**1** Sensor oder Aktor

**2** Redundanter Sensor oder Aktor

Bild 19: Beschaltung 1 - einfaches Connector Board mit Schraubklemmen

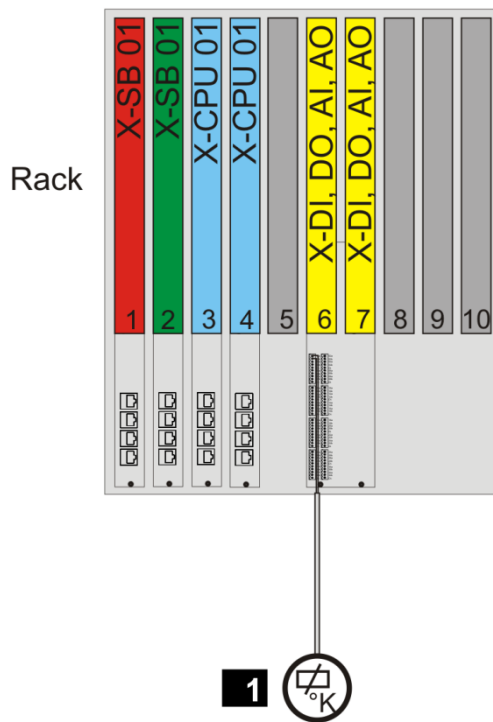
Bei Beschaltung 1 werden die Connector Boards Typ 01 (z. B. X-CB 008 01) im Basisträger benötigt.

### 8.1.2.2 Beschaltung 2

Sensoren/Aktoren an ein redundantes Connector Board mit Schraubklemmen anschließen. Das Connector Board verteilt die Signale eines Sensors an zwei redundante Module oder führt die Signale zweier redundanter Module an einem Aktor zusammen.

Für diese Beschaltung müssen der redundante Systembus und die redundante Spannungsversorgung gewährleistet sein.

- Einzelne Sensoren/Aktoren kanalweise an ein redundantes Connector Board anschließen, bei dem die E/A-Module direkt nebeneinander eingebaut sind.



**1** Sensor oder Aktor

Bild 20: Beschaltung 2 - redundantes Connector Board mit Schraubklemmen

Bei Beschaltung 2 werden die Connector Boards Typ 02 (z. B. X-CB 008 02) im Basisträger benötigt.

### 8.1.2.3 Beschaltung 3

Sensoren/Aktoren über Field Termination Assembly und Systemkabel an ein Mono-Connector Board mit Kabelstecker anschließen:

- Einzelne Sensoren/Aktoren kanalweise an ein Field Termination Assembly anschließen.
- Zwei oder mehr redundante Sensoren/Aktoren kanalweise an zwei oder mehr redundante Field Termination Assembly anschließen. Jedes Field Termination Assembly über ein Systemkabel an ein Mono-Connector Board anschließen. Die Anzahl der redundanten Sensoren/Aktoren muss gleich der Anzahl der redundanten Module sein (z. B. 2 Sensoren/2 Module)

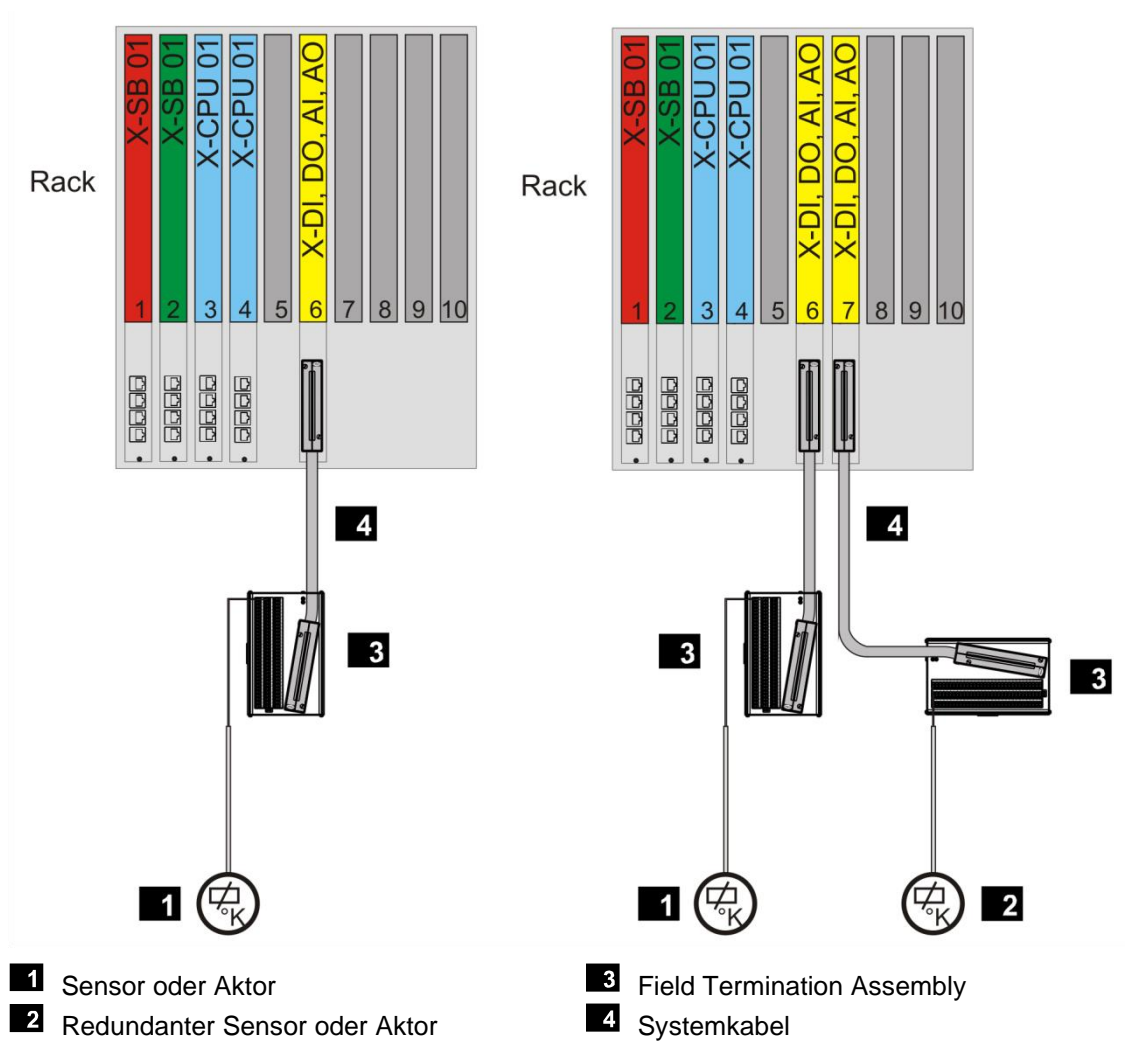


Bild 21: Beschaltung 3 - Mono-Connector Board mit Systemkabel

Bei Beschaltung 3 werden die Connector Boards Typ 03 (z. B. X-CB 008 03) im Basisträger benötigt.

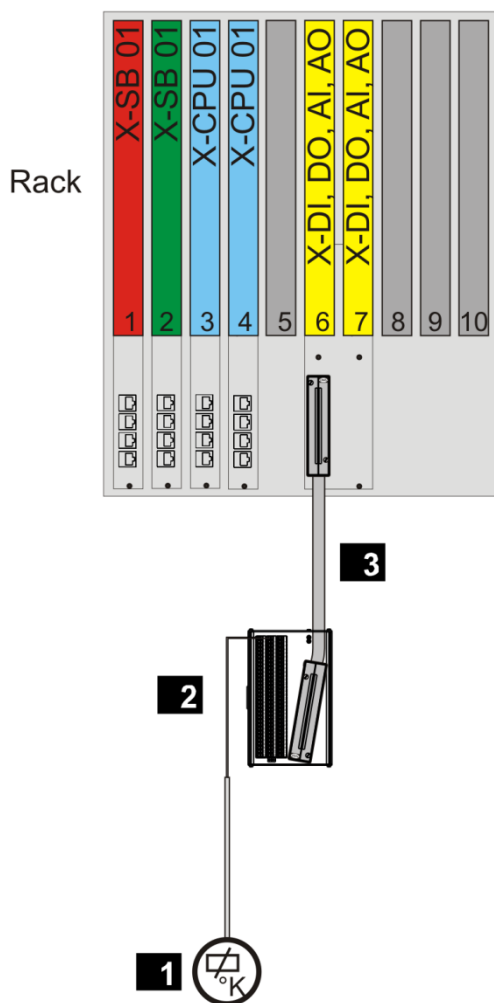


### 8.1.2.4 Beschaltung 4

Sensoren/Aktoren über Field Termination Assembly über Systemkabel an ein redundantes Connector Board mit Kabelstecker anschließen. Das Connector Board verteilt die Signale eines Sensors an zwei redundante Module oder führt das Signal zweier redundanter Module an einem Aktor zusammen.

Für diese Beschaltung müssen der redundante Systembus und die redundante Spannungsversorgung gewährleistet sein.

Einzelne Sensoren/Aktoren kanalweise an ein redundantes Connector Board über ein Field Termination Assembly anschließen. Dabei die E/A-Module auf benachbarten Steckplätzen einbauen.



**1** Sensor oder Aktor

**2** Field Termination Assembly

**3** Systemkabel

Bild 22: Beschaltung 4 - redundantes Connector Board mit Systemkabel

Bei Beschaltung 4 werden die Connector Boards mit Typ 04 (z. B. X-CB 008 04) im Basisträger benötigt.

### 8.1.3 Erdung

Die Bestimmungen der Niederspannungsrichtlinie SELV (Safety Extra Low Voltage) oder PELV (Protective Extra Low Voltage) sind zu beachten.

Zur Verbesserung der elektromagnetischen Verträglichkeit (EMV) ist eine Funktionserde vorzusehen. Diese Funktionserde im Schaltschrank so ausführen, dass sie den Anforderungen einer Schutzerde genügt.

Es ist möglich, alle HIMax Systeme mit geerdetem L- oder auch ungeerdet zu betreiben.

#### 8.1.3.1 Erdfreier Betrieb

Beim erdfreien Betrieb hat ein einziger Erdschluss keine Auswirkungen auf die Sicherheit und Verfügbarkeit der Steuerung.

Bei mehreren unentdeckten Erdschlüssen können fehlerhafte Steuersignale ausgelöst werden, deshalb empfiehlt es sich, bei erdfreiem Betrieb eine Erdschlussüberwachung einzusetzen. Bei einigen Anwendungsnormen ist eine Erdschlussüberwachung vorgeschrieben, z. B.

DIN EN 50156-1:2005. Nur von HIMA zugelassene Erdschlussüberwachungsgeräte einsetzen.

#### 8.1.3.2 Geerdeter Betrieb

Es wird vorausgesetzt, dass einwandfreie Erdungsverhältnisse vorhanden sind und eine möglichst separate Erdverbindung besteht, über die keine Fremdströme fließen. Es ist nur die Erdung des Minuspols L- zulässig. Die Erdung des Pluspols L+ ist unzulässig, da ein eventuell auftretender Erdschluss auf der Geberleitung den betreffenden Geber überbrückt.

Die Erdung von L- darf nur an einer Stelle innerhalb des Systems erfolgen. Üblicherweise wird L- direkt hinter dem Netzgerät geerdet (z. B. auf der Sammelschiene). Die Erdung soll gut zugänglich und trennbar sein. Der Erdungswiderstand muss  $\leq 2 \Omega$  sein.

#### 8.1.3.3 CE-konformer Schaltschrankaufbau

Gemäß der EU-Ratsrichtlinie 89/336/EWG, umgesetzt in das EMV-Gesetz für die Bundesrepublik Deutschland, müssen seit dem 1. Januar 1996 elektrische Betriebsmittel innerhalb der Europäischen Union das CE-Kennzeichen für elektromagnetische Verträglichkeit (EMV) tragen.

Alle Module der HIMA Systemfamilie HIMax tragen das CE-Kennzeichen.

Um beim Aufbau von Steuerungen in Schaltschränken und Gestellen EMV-Probleme zu vermeiden, ist eine sachgerechte und störungsarme Elektroinstallation in der Umgebung der Steuerungen erforderlich, z. B. keine Starkstromleitungen zusammen mit den 24-V-Speiseleitungen verlegen.

#### 8.1.3.4 Erdung in der HIMA Steuerung

Zur Gewährleistung der sicheren Funktion von HIMA Steuerungen, auch unter EMV-Gesichtspunkten, sind die in den folgenden Abschnitten ausgeführten Erdungsmaßnahmen durchzuführen.

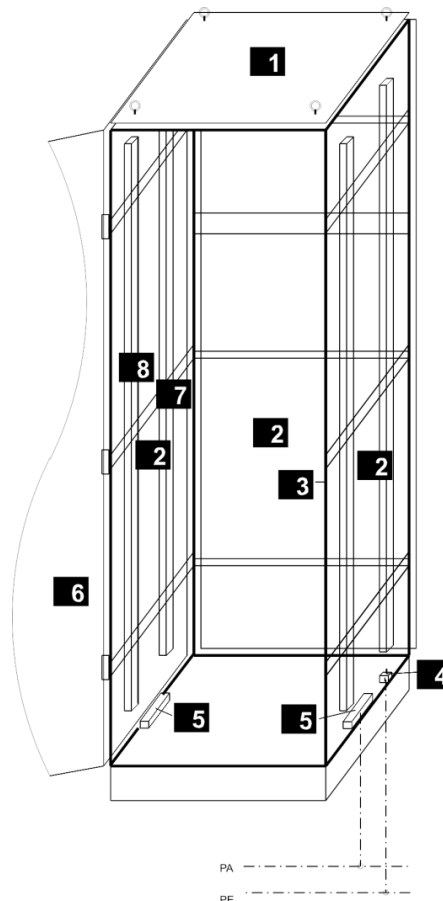
Alle berührbaren Flächen der Komponenten von HIMax, z. B. Basisträger, mit Ausnahme der steckbaren Module, sind elektrisch leitfähig (ESD-Schutz, ESD = Elektrostatische Entladung). Käfigmuttern mit Krallen stellen die sichere elektrische Verbindung zwischen Einbauteilen, wie Basisträgern und dem Schaltschrank, her. Die Krallen durchdringen die Oberfläche der Bauteile und gewährleisten eine sichere Kontaktgabe. Die dabei verwendeten Schrauben und Unterlegscheiben sind zur Vermeidung einer elektrischen Korrosion in Edelstahl ausgeführt.

### 8.1.3.5 HIMax Montage im Gestell

Das Dachblech ist über die vier Trageösen (siehe Bild 23) mit dem Schrankgerüst verschraubt. Seitenwände, Rückwand und Bodenblech sind über Erdungskralen mit dem Schrankgerüst leitend verbunden.

Zwei Sammelschienen M 2500 **5** sind standardmäßig im Schrank eingebaut und über Rundleiter 25 mm<sup>2</sup> mit dem Schrankgerüst verbunden. Nach Ausbau dieser Verbindung ist es möglich, die Sammelschienen **5** für ein von Erde getrenntes Potenzial (z. B. für den Anschluss der Abschirmung von Feldkabeln) zu verwenden.

Zum kundenseitigen Anschluss des Schutzleiters befindet sich ein Schraubbolzen M 8 **4** am Schrankgerüst.



- |  |   |
|--|---|
| <p><b>1</b> Abschirmung am Dachblech durch Standardbefestigungen am Schrankprofilrahmen</p> <p><b>2</b> Abschirmung und Erdung der Seitenbleche, Rückwand, Bodenbleche und Sockel durch Standardbefestigungen am Schrankprofilrahmen</p> <p><b>3</b> Schrankprofilrahmen bildet die Bezugserde für den Schrank</p> <p><b>4</b> Zentraler Erdungspunkt zur Erdung des Schrankprofilrahmens (M8-Gewindebolzen)</p> <p><b>5</b> Potenzialschienen M 2500 von der Schrankerde isoliert auf Schrankprofilrahmen montiert. Diese dienen zur Aufnahme des Potenzialausgleichs von externer Einspeisung und den E/A-Feldkabeln</p> | <p><b>6</b> Abschirmung und Erdung von beweglichen Schrankteilen durch Flachbänder der am Schrankprofilrahmen</p> <p><b>7</b> Erdung von mechanischen Teilen wie Chassis durch Standardbefestigungen. Die Teile sind miteinander und mit dem Schrankprofilrahmen verbunden. Die Erdung der Montageplatte erfolgt durch einen Flachbänder der 25 mm<sup>2</sup>.</p> <p><b>8</b> Potenzialausgleich über Tragschienen oder Schirmschienen. Standardfall: Potenzialausgleich über Bezugserde. Die Schienen sind leitend auf dem Chassis oder der Montageplatte befestigt.</p> |
|--|---|

Bild 23: Erdungsverbindungen im Schaltschrank

Beim Einbau von Geräten mit einer Spannung von  $\geq 60$  VDC oder  $\geq 42$  VAC ist ein Erdungsband 25 mm zu verwenden.

Das Bild 24 zeigt das Konzept der Erdung und Schirmung des 19-Zoll-Schaltschranks.

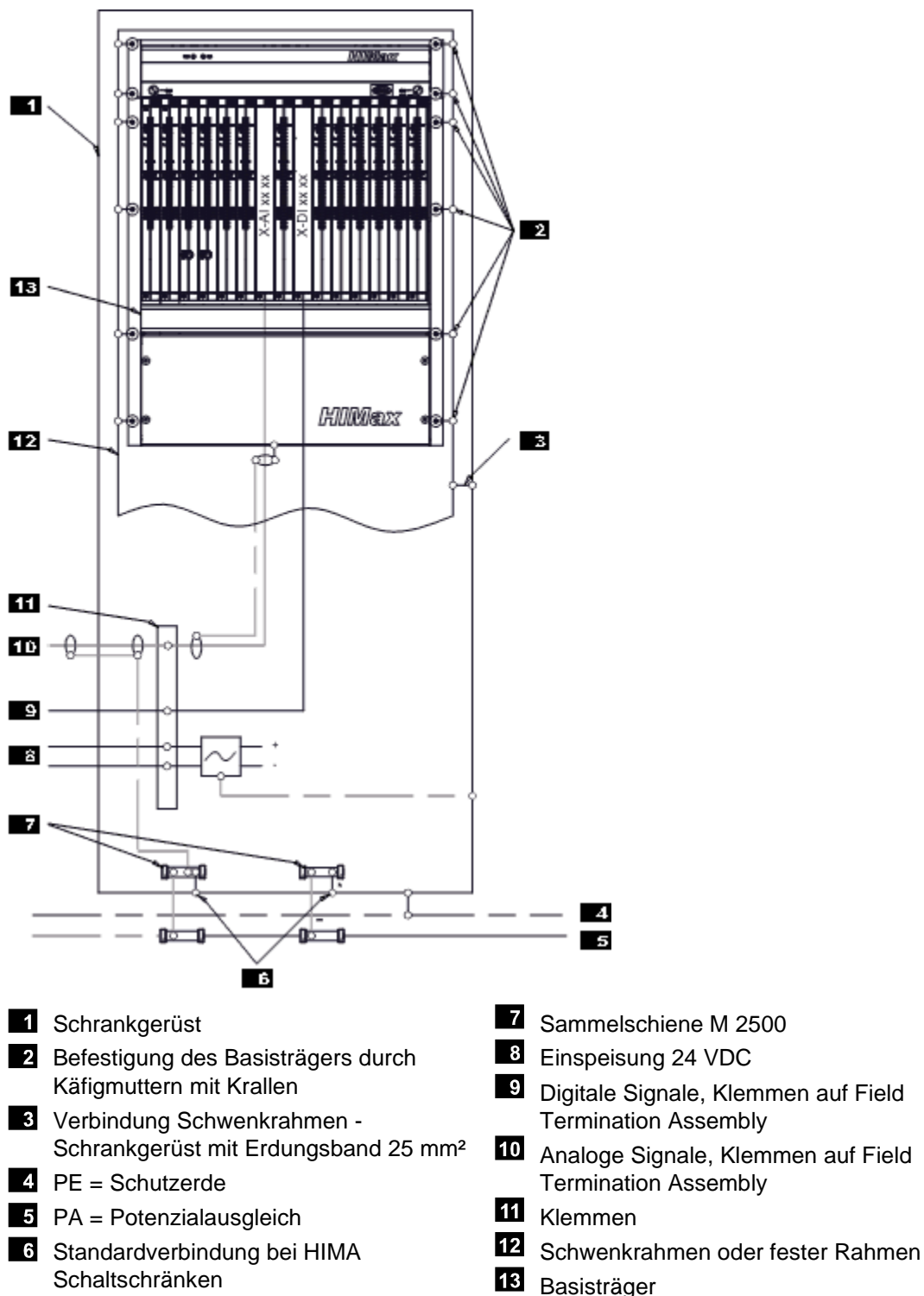
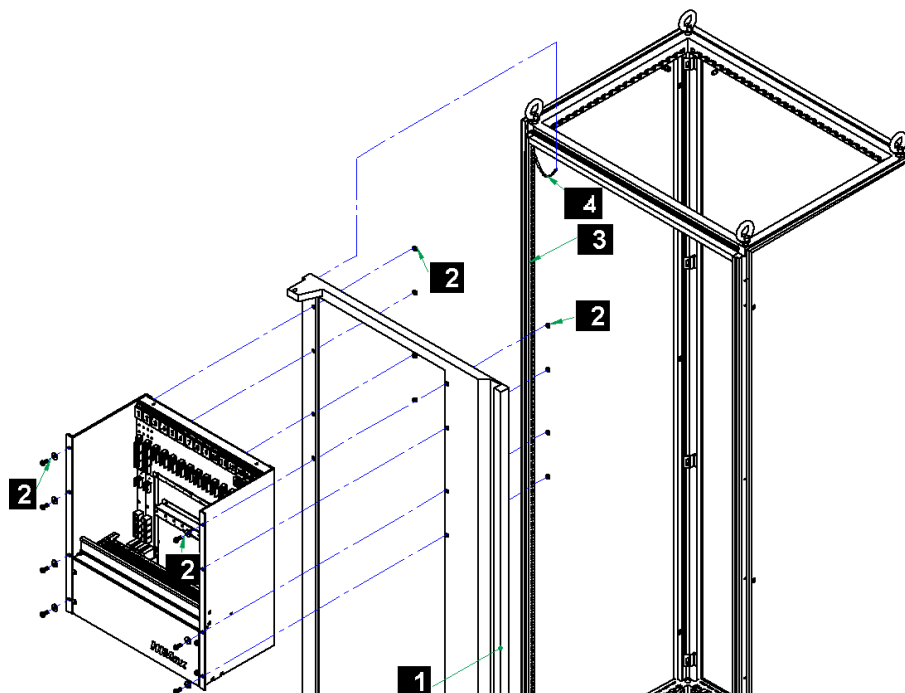


Bild 24: Erdung und Schirmung des 19"-Schaltschranks

### 8.1.3.6 HIMax Montage im Schwenkrahmen

Die Teile des Schrankgerüsts **3** sind miteinander verschweißt und damit ein elektrisch leitfähiges Konstruktionsteil. Kurze Erdungsbänder mit Querschnitten von 16 mm<sup>2</sup> oder 25 mm<sup>2</sup> verbinden Schwenkrahmen, Tür und evtl. Montageplatten leitfähig mit dem Schrankgerüst.



**1** Schwenkrahmen

**2** Schrauben und Unterlegscheiben

**3** Schrankgerüst

**4** Erdungsband 25 mm

Bild 25: Erdungsverbindungen für den Basisträger

### 8.1.3.7 Erdungsverbindungen

Nachfolgende Tabelle gibt eine Übersicht über die Größe der Erdungsverbindungen:

Einbauort	Querschnitt	Länge
Tür	16 mm <sup>2</sup>	300 mm
Schwenkrahmen (in Bild 25)	25 mm <sup>2</sup>	300 mm
Sammelschiene M 2500 (Verbindung mit Rundleiter GN/YE)	25 mm <sup>2</sup>	300 mm

Tabelle 35: Erdungsverbindungen

Für die Erdung von Bedeutung sind:

- Krallenklammern an Seitenwänden, Rückwand, Bodenblech verwendet
- Zentraler Erdpunkt (Position **4** in Bild 23)
- Trageösen  
Das Dachblech ist über vier Trageösen mit dem Schrankrahmen verbunden. Die elektrische Verbindung erfolgt über Kontaktscheiben.

Auf ordnungsgemäße Montage der Erdungsverbindungen achten!

### 8.1.3.8 Zusammenschaltung der Erdanschlüsse mehrerer Schaltschränke

Die zentrale Erde sollte möglichst störspannungsarm sein. Ist dies nicht der Fall, eine eigene Erde für die Steuerung einrichten.

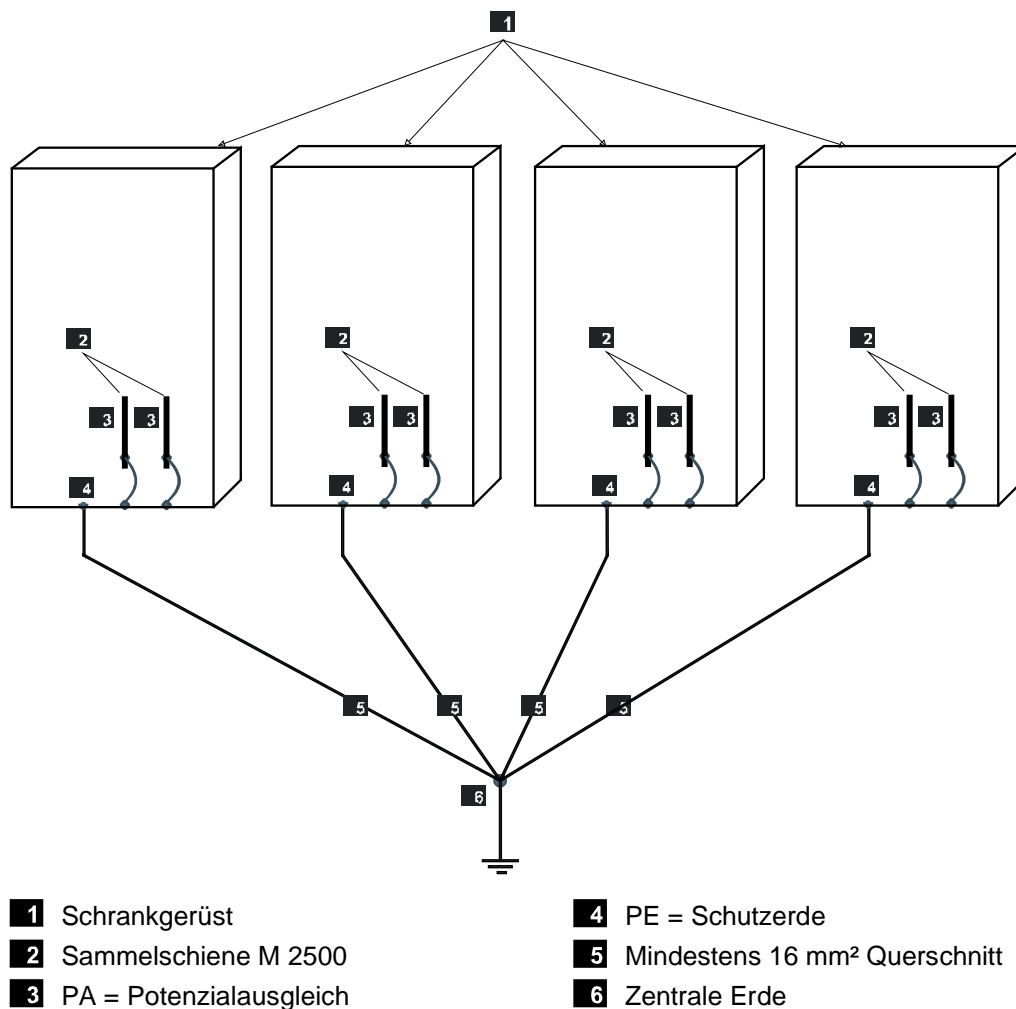


Bild 26: Erdanschlüsse mehrerer Schaltschränke

## 8.1.4 Elektrische Anschlüsse

### 8.1.4.1 Schirmung im Ein-/Ausgangsbereich

Die Feldkabel für Sensoren und Aktoren sind getrennt von Stromversorgungsleitungen und in ausreichender Entfernung von elektromagnetisch aktiven Geräten (Elektromotoren, Transformatoren) zu verlegen.

Zur Vermeidung von Störeinflüssen ist beim Anschluss von Feldkabeln auf eine durchgehende Schirmung zu achten. Dazu ist der Schirm von Feldkabeln grundsätzlich an beiden Enden aufzulegen. Dies gilt insbesondere für Feldkabel analoger Eingänge und Initiatoren.

In Fällen, bei denen hohe Ausgleichsströme erwartet werden, ist der Schirm mindestens auf einer Seite aufzulegen. Zusätzlich sind weitere Maßnahmen zu ergreifen, um Ausgleichsströme zu vermeiden z. B. durch galvanische Trennung.

Weitere Informationen für die Anforderungen an Schirmung und Erdung sind in den Handbüchern der Module zu finden.

#### 8.1.4.2 Blitzschutz für Datenleitungen in HIMA Kommunikationssystemen

Um mögliche Probleme durch Blitzschlag zu minimieren, können folgende Maßnahmen durchgeführt werden:

- Feldverdrahtung von HIMA Kommunikationssystemen komplett abschirmen.
- System korrekt erden.

In exponierten Lagen außerhalb von Gebäuden kann es sinnvoll sein, Blitzschutzgeräte zu installieren.

#### 8.1.4.3 Kabelfarben

Die Kabelfarben bei HIMax Geräten folgen international üblichen Normen.

Es ist möglich, abweichend vom HIMA Standard auf Grund nationaler normativer Anforderungen auch andere Kabelfarben bei der Verdrahtung zu verwenden. In diesem Fall sind die Abweichungen zu dokumentieren und zu verifizieren.

#### 8.1.4.4 Anschluss der Versorgungsspannung

Die Zuleitungen für die Versorgungsspannung ist an die Einspeiseklemmen (L1+, L2+, L1-, L2-) der Racks anzuschließen.

Wenn ein HIMax Modul über ein Connector Board von einer externen Spannungsquelle versorgt wird, muss dem Modul ein Netzfilter vorgeschaltet werden. Die Netzfilter sind möglichst nahe an den Connector Boards zu installieren. Es sind die Netzfilter H 7034 für 24 V und H 7035 für 48 V zu verwenden.

Die Zuleitungen für die Versorgungsspannung des Systemlüfters an dessen Schraubklemmen befestigen.

Beim Anziehen der Schrauben darauf achten, dass zur Erfüllung der Anforderungen nach UL das im Handbuch X-BASE PLATE, HI 801 024 D, genannte maximale Anzugsmoment nicht überschritten wird.

#### 8.1.4.5 Anschluss der Feldgeräte und Anschluss der Abschirmung

Bei E/A-Modulen die Zuleitungen für die Feldgeräte in den Schraubklemmen entweder der Connector Boards oder der FTAs befestigen. Dabei zur Erfüllung der Anforderungen nach UL die Anzugsmomente der Schrauben in den Modulhandbüchern beachten.

Zum Anschluss von Feldgeräten über FTAs die dafür vorgesehenen Systemkabel benutzen. Mit den Systemkabeln die FTAs und die entsprechenden Connector Boards verbinden.

Bei Verwendung von geschirmten Kabeln, ist der Schirm grundsätzlich an beiden Enden aufzulegen.

---

### i

Die korrekte Verdrahtung hängt von der Anwendung ab. Bei der Verdrahtung sind folgende Punkte zu beachten:

- Korrekte Leitungsführung
  - Biegeradius der Kabel/Leitungen
  - Zugentlastung
  - Belastbarkeit der Kabel/Leitungen
- 

#### 8.1.4.6 Verbinden der Racks

##### **Eine – redundante – Verbindung der Systembusse zweier Racks herstellen**

1. Den einen RJ-45-Stecker eines Patchkabels in die Buchse „UP“ im Connector Board des linken Systembusmoduls im ersten Rack stecken.
2. Den zweiten RJ-45-Stecker desselben Patchkabels in die Buchse „DOWN“ im Connector Board des linken Systembusmoduls im zweiten Rack stecken.

- ☒ Eine nicht-redundante Verbindung ist hergestellt
- 3. Den einen RJ-45-Stecker eines zweiten Patchkabels in die Buchse „UP“ im Connector Board des rechten Systembusmoduls im ersten Rack stecken.
- 4. Den zweiten RJ-45-Stecker desselben Patchkabels in die Buchse „DOWN“ im Connector Board des rechten Systembusmoduls im zweiten Rack stecken.
- Die beiden Racks sind redundant verbunden.

**i**

Farbige oder auf andere Weise markierte Patchkabel helfen, Verwechslungen der Kabel zu vermeiden, z. B. rote Patchkabel für Systembus A, grüne Patchkabel für Systembus B

### 8.1.5 Montage eines Connector Boards

#### **Werkzeuge und Hilfsmittel:**

- Schraubendreher Kreuz PH 1 oder Schlitz 0,8 x 4,0 mm.
- Passendes Connector Board.

#### **Connector Board einbauen:**

1. Connector Board mit der Nut nach oben in die Führungsschiene einsetzen (siehe hierzu nachfolgende Zeichnung). Die Nut am Stift der Führungsschiene einpassen.
2. Connector Board auf der Kabelschirmschiene auflegen.
3. Mit den unverlierbaren Schrauben am Basisträger festschrauben. Zuerst die unteren, dann die oberen Schrauben eindrehen.

#### **Connector Board ausbauen:**

1. Unverlierbare Schrauben vom Basisträger losschrauben.
2. Connector Board unten von der Kabelschirmschiene vorsichtig anheben.
3. Connector Board aus der Führungsschiene herausziehen.

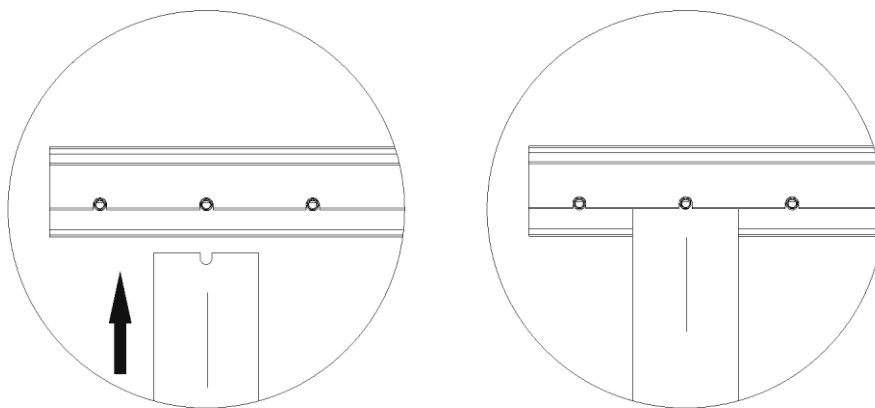


Bild 27: Einsetzen des Mono Connector Boards, exemplarisch



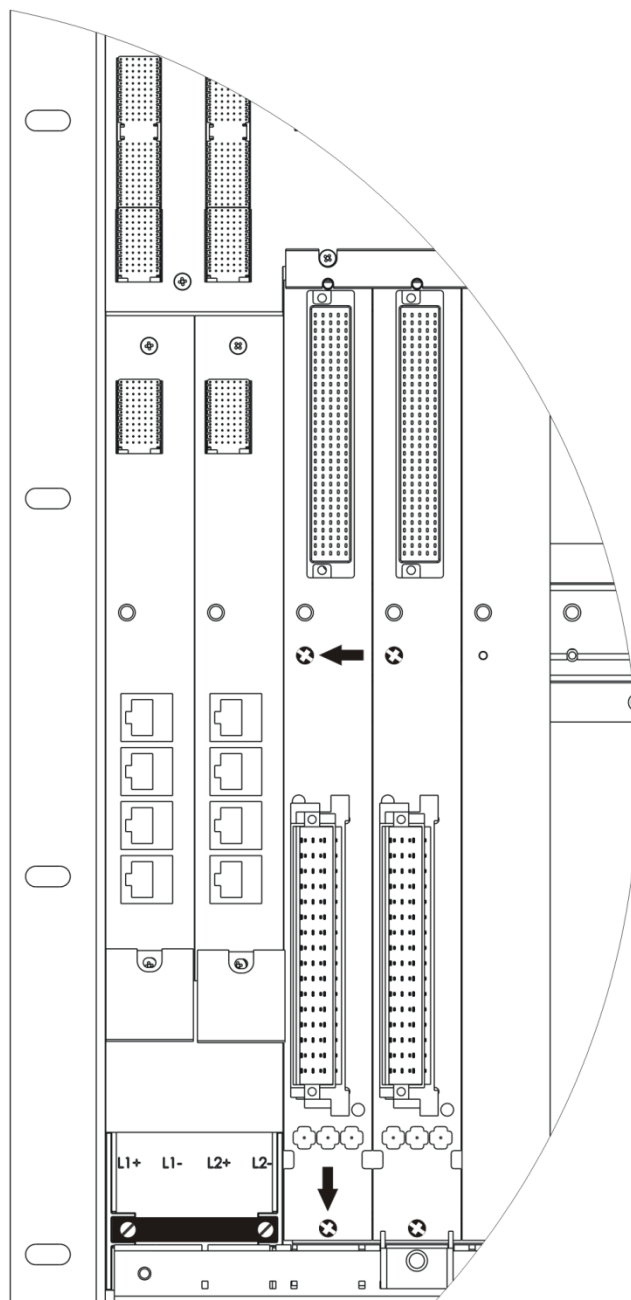


Bild 28: Festschrauben des Mono Connector Boards, exemplarisch

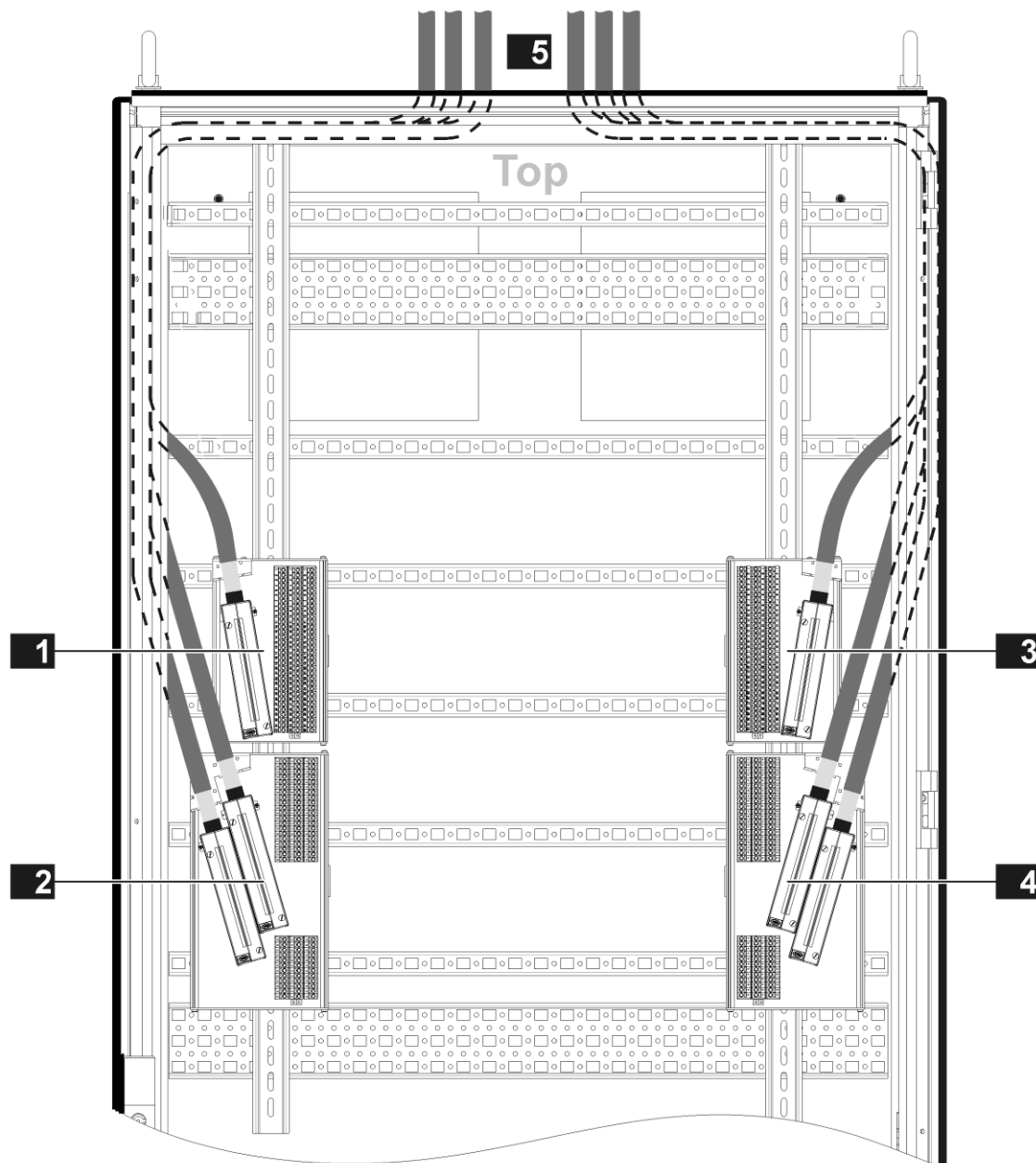
**i** Montageanleitung gilt ebenso für redundante Connector Boards. Je nach Typ des Connector Boards wird eine entsprechende Anzahl von Steckplätzen belegt. Die Anzahl der unverlierbaren Schrauben ist vom Typ des Connector Boards abhängig.

### 8.1.6 Montage von FTAs im Schaltschrank

FTAs sind in einer rechten und linken Variante verfügbar. Die linken FTAs werden mit den Systemkabeln von links und die rechten FTAs von rechts angeschlossen. Bei vertikaler Montage der FTAs ist zu unterscheiden, ob diese von oben oder unten angeschlossen werden und wie die FTAs im Schaltschrank angeordnet sind.

**i**

Bei der Montage der Systemkabel ist darauf zu achten, dass die Systemkabel keiner dauerhaften Zugbelastung ausgesetzt werden.



**1** Rechtes FTA

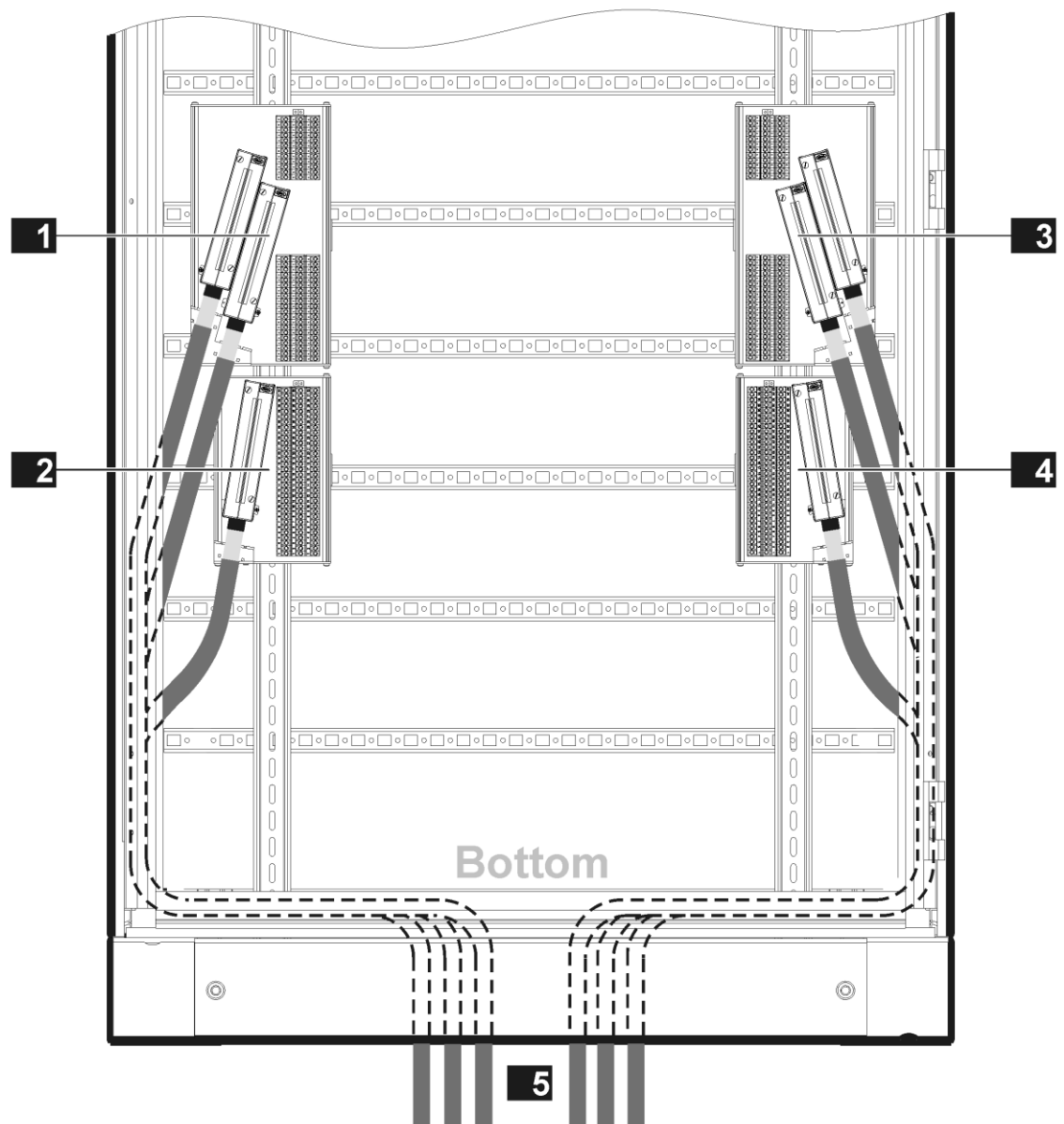
**2** Rechtes redundantes FTA

**3** Linkes FTA

**4** Linkes redundantes FTA

**5** Systemkabel-Einführung

Bild 29: Anordnung von FTAs im Schaltschrank bei Systemkabel-Einführung von oben



- |                                  |                                 |
|----------------------------------|---------------------------------|
| <b>1</b> Linkes redundantes FTA  | <b>4</b> Rechtes FTA            |
| <b>2</b> Linkes FTA              | <b>5</b> Systemkabel-Einführung |
| <b>3</b> Rechtes redundantes FTA |                                 |

Bild 30: Anordnung von FTAs im Schaltschrank bei Systemkabel-Einführung von unten

### 8.1.7 Wärmebetrachtung

Der hohe Integrationsgrad elektronischer Bauelemente verursacht Verlustwärme, die zudem von der externen Belastung der HIMax Module abhängig ist. Daher sind, je nach Aufbau, die Montage und die Luftverteilung innerhalb des Schaltschranks von Bedeutung. Eine geringere Umgebungstemperatur erhöht die Lebensdauer und die Zuverlässigkeit der eingebauten Komponenten.

#### i

Bei der Montage der Geräte müssen die zulässige Umgebungsbedingungen eingehalten werden.

#### 8.1.7.1 Wärmeabfuhr

Ein geschlossenes Gehäuse oder ein geschlossener Schrank muss so beschaffen sein, dass die im Innenraum auftretende Wärme über die Oberfläche abgeführt werden kann.

Montageart und -ort so wählen, dass die Wärmeabfuhr gewährleistet bleibt.

Zur Bestimmung der Lüftungskomponenten sind die Verlustleistungen der Einbauten maßgebend. Es wird von einer gleichmäßigen Verteilung der Wärmebelastung und einer ungestörten Eigenkonvektion ausgegangen.

#### 8.1.7.2 Definitionen

Größe	Beschreibung	Einheit
$P_V$	Verlustleistung (Wärmeleistung) der im Gehäuse eingebauten elektronischen Komponenten	W
A	Effektive Gehäuseoberfläche (siehe unten)	m <sup>2</sup>
B	Gehäusebreite	m
H	Gehäusehöhe	m
T	Gehäusetiefe	m
k	Wärmedurchgangskoeffizient des Gehäuses	W/m <sup>2</sup> K
	Beispiel Stahlblech	Ca. 5,5 W/m <sup>2</sup> K

Tabelle 36: Definitionen zur Berechnung der Verlustleistung

#### 8.1.7.3 Aufstellungsart

Die effektive Gehäuseoberfläche A in Abhängigkeit von der Montage oder Aufstellungsart wie folgt ermitteln:






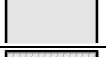

Gehäuseaufstellung nach VDE 0660 Teil 5	Berechnung der Gehäuseoberfläche A
 Einzelgehäuse allseitig freistehend	$A = 1,8 \times H \times (B + T) + 1,4 \times B \times T$
 Einzelgehäuse für Wandanbau	$A = 1,4 \times B \times (H + T) + 1,8 \times H \times T$
 Endgehäuse freistehend	$A = 1,4 \times T \times (B + H) + 1,8 \times B \times H$
 Endgehäuse für Wandanbau	$A = 1,4 \times H \times (B + T) + 1,4 \times B \times T$
 Mittelgehäuse freistehend	$A = 1,8 \times B \times H + 1,4 \times B \times T + H \times T$
 Mittelgehäuse für Wandanbau	$A = 1,4 \times B \times (H + T) + H \times T$
 Mittelgehäuse für Wandanbau, Dachfläche abgedeckt	$A = 1,4 \times B \times H + 0,7 \times B \times T + H \times T$

Tabelle 37: Aufstellungsarten

#### 8.1.7.4 Eigenkonvektion

Bei der Eigenkonvektion wird die Verlustwärme über die Wände des Gehäuses nach außen abgeführt. Voraussetzung: Umgebungstemperatur niedriger als die Temperatur innerhalb des Gehäuses.

Die maximale Temperaturerhöhung  $(\Delta T)_{\max}$  aller elektronischen Geräte im Gehäuse berechnet sich wie folgt:

$$(\Delta T)_{\max} = \frac{P_V}{k \cdot A}$$

Die Verlustleistung  $P_V$  lässt sich aus den elektrischen Leistungen der Steuerung sowie deren Eingängen und Ausgängen anhand der technischen Daten berechnen.

#### 8.1.7.5 Normhinweis

Die Berechnung der Temperatur in einem Gehäuse kann auch nach VDE 0660 Teil 507 (HD 528 S2) erfolgen.

### i

Bei der Wärmebetrachtung alle Komponenten in einem Schrank oder Gehäuse berücksichtigen, auch solche, die nicht Teil des HIMax Systems sind!

#### 8.1.7.6 Temperaturüberwachung

Die Steuerungen sind für den Betrieb bis zu einer maximalen Umgebungstemperatur von 60 °C ausgelegt. Die Temperaturzustände der einzelnen Module werden zentral vom Prozessormodul ausgewertet.

Ein Temperatursensor an einer besonders temperaturrelevanten Stelle erfasst und überwacht den Temperaturzustand auf dem jeweiligen Modul selbsttätig und kontinuierlich.

Die Temperaturzustände lassen sich mit SILworX über die Systemvariable *Temperaturzustand [X]* für jedes Prozessormodul  $X = 1 \dots 4$  auswerten (Zuordnung der Indizes zu Steckplatznummern siehe Kapitel 5.2.5).

Die Systemvariable *Temperaturzustand[X]* signalisiert die gemessenen Umgebungstemperaturen in den folgenden Temperaturbereichen:

Temperaturbereich	Temperaturzustand	Wert der Systemvariablen <i>Temperaturzustand[X]</i> [BYTE]
< 40 °C	Normal	0x00
≥ 40 °C	Schwelle 1 überschritten	0x01
> 60 °C	Schwelle 2 überschritten	0x03
Rückkehr auf 60 °C ... 40 °C	Schwelle 1 überschritten	0x01
Rückkehr auf < 40 °C	Normal	0x00

Tabelle 38: Temperaturzustände

Über- oder unterschreitet die Umgebungstemperatur eine Temperaturschwelle, wechselt der Temperaturzustand.

Die Tabelle 38 gilt für den normalen Betrieb von HIMax zusammen mit dem Systemlüfter X-FAN. Abhängig vom Steckplatz eines Moduls im Rack und der eigenen Verlustleistung eines Moduls kann der Temperaturzustand schon bei Umgebungstemperaturen unterhalb der angegebenen Grenzen der Temperaturbereiche wechseln.

Bei abnormalem Betrieb, z. B. ohne Lüfter, kann der Temperaturzustand schon bei Umgebungstemperaturen unterhalb der angegebenen Grenzen der Temperaturbereiche wechseln.

Der Übergang in den Zustand *Schwelle 1 überschritten* oder *Schwelle 2 überschritten* bedeutet **nicht**, dass die Sicherheit des Systems beeinträchtigt ist.

Für jedes Rack ist einstellbar, welche Temperaturschwelle bei einer Überschreitung zu einer Meldung führt. Die Parametrierung erfolgt im SILworX Hardware-Editor, in der Detailansicht des Racks.

## 8.2 Inbetriebnahme

Das System erst nach vollständigem Aufbau der Hardware und Anschluss aller Kabel hochfahren. Zunächst ist der Schaltschrank, danach das PES selbst in Betrieb zu nehmen.

### HINWEIS



**Anlagenschaden möglich!**

**Anlagenschaden durch falsch angeschlossene oder falsch programmierte sicherheitsbezogene Automatisierungssysteme.**

**Anschlüsse vor Inbetriebnahme prüfen und Gesamtanlage testen!**

### 8.2.1 Schaltschrank in Betrieb nehmen

Vor dem Zuschalten der Versorgungsspannung ist zu prüfen, ob alle Kabel korrekt angeschlossen sind und somit kein Risiko für Steuerung und Anlage besteht.

#### 8.2.1.1 Alle Eingänge und Ausgänge prüfen

Unzulässige Fremdspannungen (insbesondere z. B. 230 VAC gegen Erde oder L-) lassen sich mit einem Universal-Messinstrument messen.

HIMA empfiehlt, jeden einzelnen Anschluss auf unzulässige Fremdspannung zu prüfen.

Bei der Prüfung der externen Kabel auf Isolationswiderstand, Schluss und Bruch dürfen die Kabel beidseitig nicht angeschlossen sein, um Defekte oder Zerstörungen der Module durch zu hohe Spannungen zu vermeiden.

Zur Prüfung auf Erdschluss Spannungsanschlüsse der Kabelstecker auf den Potenzialverteiler abziehen, Speisespannungen für die Sensoren und Minuspol an den Aktoren auftrennen.

Ist der Minuspol während des Betriebs geerdet, ist die Erdverbindung während der Dauer der Überprüfung auf Erdschluss zu unterbrechen. Dies gilt auch für die Erdverbindung einer eventuell vorhandenen Erdschlussmesseinrichtung.

Zur Prüfung jedes Anschlusses gegen Erde ist ein Widerstandsmesser oder eine spezielle Messeinrichtung zu verwenden.

#### 8.2.1.2 Spannungszuschaltung

Voraussetzung: Die HIMax Module müssen gesteckt und zugehörige Kabel müssen angeschlossen sein. Die Versorgungsspannung 24 VDC ist vor Anschluss auf richtige Polarität, Höhe und Welligkeit zu prüfen.

## 8.2.2 Inbetriebnahme der Steuerung mit X-CPU 01

Voraussetzungen für die Inbetriebnahme:

- Hardware installiert.
- Racks sind untereinander nicht verbunden.
- Mode-Schalter aller Prozessormodule befinden sich in Stellung *Init*.
- Alle übrigen Module im Zustand STOPP.
- Der Netzwerkanschluss des PADT ist so konfiguriert, dass die Module des HIMax Basisträgers erreichbar sind. Falls nötig, für die benutzte Schnittstellenkarte ein Routing eintragen.
- Ein geeignetes Projekt mit der Konfiguration der Rack-IDs, IP-Adressen und System-ID ist vorhanden.

### Steuerung mit X-CPU 01 in Betrieb nehmen

1. Versorgungsspannung einschalten.
2. SRS, IP-Adresse, Attribut *responsible* und Systembusmodus auf dem Systembusmodul in Rack 0, Steckplatz 1 einstellen:
  - Direkte physikalische Verbindung zwischen PADT und Systembusmodul herstellen.

---

**i**

Die Ethernet-Schnittstelle *PADT* des Systembusmoduls ist nicht in der Lage, Auto-Cross-Over durchzuführen.

Daher ist für die Verbindung zum Systembusmodul ein Cross-Over-Kabel zu benutzen.

---

- Im Strukturbaum Ressource, **Hardware** anklicken, dann **Online** in der Aktionsleiste klicken.  
Register *Online Hardware* und Fenster *System-Login* öffnen sich.
  - Schaltfläche **Zum Modul-Login** klicken.
  - In *Online Hardware* auf dem Systembusmodul einloggen (Doppelklick auf das Systembusmodul, Modul-Login-Fenster öffnet sich).  
MAC-Adresse (siehe Aufkleber auf dem Modul) benutzen, um die IP-Adresse und die SRS auszulesen (Schaltfläche **Suchen ...** im Login-Fenster).
  - Über die Schaltfläche **Ändern** im Fenster *Suchen per MAC* das Fenster *Schreiben per MAC* öffnen. In diesem Fenster können SRS, IP-Adresse, Attribut *responsible* und Systembusmodus auf dem Systembusmodul eingestellt werden.  
Das Systembusmodul in Rack 0, Steckplatz 1 ist immer *responsible* für Systembus A. Für Systembus B kann zwischen den Systembusmodulen auf Rack 0, Steckplatz 2 und Rack 1, Steckplatz 2 gewählt werden.
3. Punkt 2 wiederholen für alle Systembusmodule auf allen vorhandenen Racks.
  4. Systembusse aller Racks miteinander verbinden. Dazu eine Ethernet-Verbindung herstellen zwischen:
    - Rack 0, Steckplatz 1, Anschluss *DOWN* und Rack 1, Steckplatz 1 Anschluss *UP*.
    - Rack 0, Steckplatz 2, Anschluss *DOWN* und Rack 1, Steckplatz 2, Anschluss *UP*.Falls Rack 2 verwendet wird, eine Ethernet-Verbindung herstellen zwischen:
    - Rack 0, Steckplatz 1, Anschluss *UP* und Rack 2, Steckplatz 1 Anschluss *DOWN*.
    - Rack 0, Steckplatz 2, Anschluss *UP* und Rack 2, Steckplatz 2, Anschluss *DOWN*.Die Verbindungen zu den weiteren Racks erfolgen entsprechend.
- ☒ Die LEDs *UP* und *DOWN* und die LEDs *Red* der entsprechenden Systembusmodule leuchten.
5. Prozessormodul auf Rack 0 in Steckplatz 3 vorbereiten:
    - Auf Prozessormodul einloggen: Doppelklick auf die Darstellung des Prozessormoduls im Online-Abbild.

**i**

Wenn eine gültige Konfiguration auf einem Prozessormodul geladen ist und die Bedingungen für den Systembetrieb erfüllt sind, werden alle Einstellungen wie SRS und IP-Adressen aus der gültigen Konfiguration wirksam. Besonders beachten bei Verwendung eines Prozessormoduls mit Vorgeschichte bei einer Erstinbetriebnahme.

HIMA empfiehlt, Prozessormodule mit unbekannter Vorgeschichte auf Werkseinstellungen zurückzusetzen (Urlöschen).

- Prozessormodul auf Werkseinstellungen zurücksetzen (Urlöschen).
  - SRS auf dem Prozessormodul einstellen.  
Bei Mono-System (ein Prozessormodul, ein Systembusmodul) Mono-Betrieb einstellen. Dazu im Menü *Online->Inbetriebnahme* den Punkt **Mono-/Redundanzbetrieb einstellen** wählen.  
Die Einstellung ist nur wirksam, wenn ein Mono-Projekt geladen wird. Andernfalls setzt das System den Schalter automatisch zurück.
  - Den Mode-Schalter des Prozessormoduls auf *Stop* drehen.
  - ☒ Nach einiger Zeit zeigt das Prozessormodul an, Systembetrieb erreicht zu haben: LED *Stop* leuchtet oder blinkt, LED *Init* ist dunkel.
6. Ins System einloggen.
7. Mode-Schalter aller übrigen Prozessormodule nacheinander in Stellung *Stop* bringen.
- ☒ Prozessormodule nehmen am Systembetrieb teil (Redundanz). Die LEDs *Stop* leuchten oder blinken, die LEDs *Init* sind dunkel.
8. Vorhandene Konfiguration auf die Prozessormodule laden (Menü **Online -> Ressource Download**)
- ☒ Alle Prozessormodule erreichen den Zustand STOPP / GÜLTIGE KONFIGURATION.
9. Mode-Schalter aller Prozessormodule in Stellung *Run* bringen.
10. Kaltstart der Ressource durchführen.
- Das System, das heißt, alle Module, sind im Zustand RUN (oder im Zustand RUN / AP STOPP, falls das Anwenderprogramm nicht gestartet wurde).

Eine detaillierte Beschreibung der Inbetriebnahme befindet sich im Erste-Schritte-Handbuch HI 801 102 D.

### 8.2.2.1 Fehlerfälle

- Ein Prozessormodul nimmt nicht den redundanten Betrieb auf oder verlässt diesen wieder, wenn es gestört ist.
- Das System geht in den Zustand STOPP / FEHLERHAFTE KONFIGURATION, wenn das Projekt in SILworX nicht zur Hardware passt.



### 8.2.3 Inbetriebnahme einer Steuerung mit X-CPU 31

Prozessormodule vom Typ X-CPU 31 sind nur in Rack 0 einsetzbar. Sie arbeiten gleichzeitig als Prozessormodul und als Systembusmodul. Daher sind sie auf die Steckplätze 1 und 2 zu stecken.

Vorbedingungen herstellen:

- Die Versorgungsspannung ist nicht zugeschaltet.
- Racks sind untereinander nicht verbunden.
- Prozessormodule X-CPU 31 sind in den Steckplätzen 1 und 2 vorhanden. Kommunikationsmodule und E/A- Module können ebenfalls gesteckt sein.
- Die Mode-Schalter der Prozessormodule befinden sich in Stellung *Init*.
- Alle übrigen Module im Zustand STOPP.
- Der Netzwerkanschluss des PADT ist so konfiguriert, dass die Module des HIMax Basisträgers erreichbar sind. Falls nötig, für die benutzte Schnittstellenkarte ein Routing eintragen.
- Ein passendes SILworX Projekt mit der Konfiguration der Rack-IDs, IP-Adressen und System-ID ist vorhanden.

#### Steuerung mit X-CPU 31 in Betrieb nehmen

1. Versorgungsspannung zuschalten.
2. SRS, IP-Adresse, Attribut *responsible* und Systembusmodus des linken Prozessormoduls X-CPU 31 einstellen:
  - Direkte physikalische Verbindung zwischen PADT und Prozessormodul herstellen.
  - Im Strukturbaum Ressource, **Hardware** anklicken, dann **Online** in der Aktionsleiste klicken.  
Register *Online Hardware* und Fenster *System-Login* öffnen sich.
  - Schaltfläche **Zum Modul-Login** klicken.
  - In *Online Hardware* auf dem Prozessormodul einloggen (Doppelklick auf das Prozessormodul, Modul-Login-Fenster öffnet sich).  
MAC-Adresse (siehe Aufkleber auf dem Modul) benutzen, um die IP-Adresse und die SRS auszulesen (Schaltfläche **Suchen** ... im Login-Fenster).
  - Über die Schaltfläche **Ändern** im Fenster *Suchen per MAC* das Fenster *Schreiben per MAC* öffnen. In diesem Fenster können SRS, IP-Adresse, (Attribut *responsible*) und Systembusmodus auf dem Prozessormodul eingestellt werden.  
Beide Prozessormodule X-CPU 31 sind immer *responsible*.
3. Punkt 2 wiederholen für das rechte Prozessormodul X-CPU 31 und für alle Systembusmodule auf allen vorhandenen Racks. Die Systembusmodule können nicht *responsible* sein!
4. Systembusse aller Racks miteinander verbinden. Dazu eine Ethernet-Verbindung herstellen zwischen:
  - Rack 0, Steckplatz 1, Anschluss *DOWN* und Rack 1, Steckplatz 1 Anschluss *UP*.
  - Rack 0, Steckplatz 2, Anschluss *DOWN* und Rack 1, Steckplatz 2, Anschluss *UP*.Falls Rack 2 verwendet wird, eine Ethernet-Verbindung herstellen zwischen:
  - Rack 0, Steckplatz 1, Anschluss *UP* und Rack 2, Steckplatz 1 Anschluss *DOWN*.
  - Rack 0, Steckplatz 2, Anschluss *UP* und Rack 2, Steckplatz 2, Anschluss *DOWN*.Die Verbindungen zu den weiteren Racks erfolgen entsprechend.
- ☒ Die LEDs *UP* und *DOWN* und die LEDs *Red* der entsprechenden Prozessormodule und Systembusmodule leuchten.
5. Prozessormodul auf Rack 0 in Steckplatz 1 vorbereiten:
  - Auf Prozessormodul einloggen: Doppelklick auf die Darstellung des Prozessormoduls im Online-Abbild.

**i**

Wenn eine gültige Konfiguration auf einem Prozessormodul geladen ist und die Bedingungen für den Systembetrieb erfüllt sind, werden alle Einstellungen wie SRS und IP- Adressen aus der gültigen Konfiguration wirksam. Besonders beachten bei Verwendung eines Prozessormoduls mit Vorgeschichte bei einer Erstinbetriebnahme.

HIMA empfiehlt, Prozessormodule mit unbekannter Vorgeschichte auf Werkseinstellungen zurückzusetzen (Urlöschen).

- Prozessormodul auf Werkseinstellungen zurücksetzen (Urlöschen).
- Bei Mono-System (ein Prozessormodul) Mono-Betrieb einstellen. Dazu im Menü *Online->Inbetriebnahme* den Punkt **Mono-/Redundanzbetrieb einstellen** wählen.  
Die Einstellung ist nur wirksam, wenn ein Mono-Projekt geladen wird. Andernfalls setzt das System den Schalter automatisch zurück.
- 6. Mode-Schalter des linken Prozessormoduls auf *Stop* einstellen und warten, bis das Prozessormodul anzeigt, Systembetrieb erreicht zu haben.  
☒ Die LED *Stop* leuchtet oder blinkt, die LED *Init* ist aus.
- 7. Ins System einloggen.
- 8. Mode-Schalter des rechten Prozessormoduls in Stellung *Stop* bringen.  
☒ Das rechte Prozessormodul geht in den redundanten Betrieb. Die LED *Stop* leuchtet und die LED *Init* ist aus.
- 9. Vorhandene Konfiguration mittels **Download** auf die Prozessormodule laden (Menü **Online -> Ressource Download**).  
☒ Die Prozessormodule gehen in STOPP / GÜLTIGE KONFIGURATION.
- 10. Mode-Schalter aller Prozessormodule auf *Run* einstellen.
- 11. Kaltstart der Ressource durchführen.  
► Das System, das heißt, alle Module, sind im Zustand RUN (oder im Zustand RUN / AP STOPP, falls das Anwenderprogramm nicht gestartet wurde).

Eine detaillierte Beschreibung der Inbetriebnahme befindet sich im Erste-Schritte-Handbuch HI 801 102 D.

### 8.2.3.1 Fehlerfälle

- Ein Prozessormodul nimmt nicht den redundanten Betrieb auf oder verlässt diesen wieder, wenn es gestört ist.
- Das System geht in den Zustand STOPP / FEHLERHAFTE KONFIGURATION, wenn das Projekt in SILworX nicht zur Hardware passt.

#### 8.2.4 Rack-ID zuweisen

Beim Aufbau und bei Erweiterungen der Hardware ist den Basisträgern ihre Identifikationsnummer zu zuweisen oder eine bestehende Zuweisung zu ändern.

Die Rack-ID ist im Connector Board des Systembusmoduls gespeichert und muss über das Systembusmodul geändert werden. Das Systembusmodul verteilt die Rack-ID an die übrigen Module eines Racks.

Von der Rack-ID hängt die eindeutige Identifizierung eines Racks und der darin befindlichen Module ab. Davon wiederum hängt die Identifizierung der Ein- und Ausgänge ab.

Die Rack-ID immer über einen direkten Anschluss des PADT an das betreffende Systembusmodul einstellen, um auszuschließen, dass die Rack-ID des falschen Systembusmoduls verändert wird.

**Vorgehensweise einhalten, Rack-ID ist ein sicherheitskritischer Parameter**

##### **Rack-ID zuweisen**

1. Vorbedingungen herstellen:

- ☒ Alle Module des Racks im Zustand STOPP (damit keine alte Rack-ID zwischen den Modulen ausgetauscht wird).
- ☒ Keine Verbindung zwischen PADT und Prozessormodul.
- ☒ Direkte Verbindung zwischen PADT und Systembusmodul.

2. Rack-ID ändern:

- Über die Direktverbindung die Rack-ID des einen Systembusmoduls ändern.
- Die Rack-ID des zweiten Systembusmoduls ändern (falls vorhanden), ebenfalls über eine Direktverbindung.

► Die neue Rack-ID ist gültig. Die Konfiguration ist konsistent.

#### **HINWEIS**



**Fehlfunktion der Steuerung durch inkonsistente Rack-ID(s)!**

**Rack-ID ist ein sicherheitskritischer Parameter, deshalb die Rack-ID nur auf die beschriebene Weise ändern!**

## 8.2.5 Umschalten zwischen Linienstruktur und Netzstruktur

Die Umschaltung des HIMax Systems zwischen Linien- und Netzstruktur ist nur durch Umschalten der Systembusmodule möglich.

### 8.2.5.1 Umschaltung auf Netzstruktur

Voraussetzungen zum Umschalten des Systembusmodus auf Netzstruktur:

- Die Racks sind in Linienstruktur verbunden.
- Alle Racks sind redundant angeschlossen.
- Das System ist fehlerfrei und korrekt parametrier.
- Die Prozessormodule sind in STOPP.
- Das PADT ist mit dem System am Rack 0 verbunden. Ein System-Login wurde durchgeführt.

#### Umschalten auf Netzstruktur

1. Zunächst Systembus A umschalten, dazu die folgenden Schritte 2. und 3. jeweils für das **linke** Systembusmodul oder Prozessormodul X-CPU 31 eines Racks ausführen:
  2. Den Systembusmodus des Systembusmoduls auf **Netz** einstellen, das vom Rack mit der Rack-ID 0 am weitesten entfernt ist. Am weitesten entfernt bedeutet, dass die Verbindung zu diesem Rack über die meisten anderen Racks oder Ethernet-Segmente verläuft.
  3. Den Schritt 2 nacheinander für das dann am weitesten entfernten Rack durchführen, bis zu Rack-ID 0.
  4. Nach dem Umschalten des Systembusmoduls oder Prozessormoduls X-CPU 31 in Rack 0 verbindet sich der Systembus A neu. Dies kann einige Zeit dauern.
  5. Wenn Systembus A auf Netzbetrieb umgestellt und verbunden ist, den Systembus B umschalten. Dazu die Schritte 2. und 3. jeweils für das **rechte** Systembusmodul (in Rack 0 evtl. X-CPU 31) durchführen.
- Das HIMax System arbeitet in Netzstruktur. Die Racks können in der gewünschten Struktur neu verbunden werden.

### 8.2.5.2 Umschaltung auf Linienstruktur

Voraussetzungen zum Umschalten des Systembusmodus auf Linienstruktur:

- Die Racks sind in Netzstruktur aufgebaut.
- Das System ist fehlerfrei und korrekt parametrier.
- Die Prozessormodule sind in STOPP.
- Das PADT ist mit dem System am Rack 0 verbunden. Ein System-Login wurde durchgeführt.

#### Umschalten auf Linienstruktur

1. Zunächst Systembus A umschalten, dazu die folgenden Schritte 2. und 3. jeweils für das **linke** Systembusmodul eines Racks (in Rack 0 evtl. X-CPU 31) ausführen:
  2. Den Systembusmodus des Systembusmoduls auf **Linie** einstellen, das vom Rack mit der Rack-ID 0 am weitesten entfernt ist. Am weitesten entfernt bedeutet, dass die Verbindung zu diesem Rack über die meisten anderen Racks oder Ethernet-Segmente verläuft.
  3. Den Schritt 2 nacheinander für das dann am weitesten entfernten Rack durchführen, bis zu Rack-ID 0.
  4. Nach dem Umschalten von Systembus A die Systembus-Verkabelung in Linienstruktur umbauen. Dabei die Racks so verbinden, dass die Reihenfolge der Rack-IDs einer korrekten Linienstruktur entspricht.
  5. Nach dem erfolgreichen Umbau der Verkabelung von Systembus A den Systembus B umschalten und umbauen. Dazu die Schritte 2. und 3. jeweils für das **rechte** Systembusmodul (in Rack 0 evtl. X-CPU 31) durchführen.
- Das HIMax System arbeitet in Linienstruktur.

## 8.3 Wartung und Instandhaltung

HIMA empfiehlt, die Lüfter der Steuerung in regelmäßigen Abständen zu wechseln.

i

Bei einer sicherheitsbezogenen Anwendung muss die Steuerung in regelmäßigen Abständen einer Wiederholungsprüfung unterzogen werden. Näheres siehe Sicherheitshandbuch HI 801 002 D.

### HINWEIS



#### **Fehlfunktionen durch Elektrostatische Entladung!**

**Schaden an der Steuerung oder an angeschlossenen elektronischen Geräten!**

**Wartungsarbeiten an Versorgungs-, Signal- und Datenleitungen nur durch qualifiziertes Personal. ESD-Schutzmaßnahmen beachten. Vor jedem Kontakt mit den Versorgungs- oder Signalleitungen muss das Personal elektrostatisch entladen sein!**

### 8.3.1 Störungen

Störungen im Prozessormodul führen dazu, dass ein redundantes Prozessormodul die Steuerungsaufgabe übernimmt. Wenn kein redundantes Prozessormodul existiert, erfolgt Abschalten der gesamten Steuerung.

Die LED *Error* auf dem Prozessormodul zeigt Störungen an.

Mögliche Ursachen für die Anzeige *Error* siehe Handbuch des verwendeten Prozessormoduls.

Alle Module erkennen während des Betriebs automatisch Störungen und zeigen sie durch die LED *Error* auf der Frontplatte an.

Über SILworX sind Fehler auch im Zustand STOPP diagnostizierbar (ausgenommen Kommunikationsfehler).

Einige der Modul-Selbsttests laufen im Hintergrund ab. Besteht ein Modul den Hintergrundtest nicht, erfolgt ein Eintrag in die Diagnosehistorie. Besteht das Modul nicht innerhalb von 24 h nach dem Diagnoseeintrag den Hintergrundtest, schaltet es sich als defekt ab.

Vor dem Wechsel eines E/A-Moduls prüfen, ob eine externe Leitungsstörung vorliegt und der entsprechende Sensor oder Aktor in Ordnung ist.

Nachdem eine Störung beseitigt ist (z. B. durch Reparatur der extern angeschlossenen Leitungen, durch Tausch eines Moduls) geht das HIMax System selbsttätig in den fehlerfreien Zustand zurück und löscht die entsprechenden LEDs. Eine Quittierung durch den Bediener ist nicht erforderlich.

Wird für eine Anwendung eine Wiederanlaufsperrung benötigt, dann ist diese im Anwenderprogramm zu realisieren.

### 8.3.2 Zuschalten der Spannungsversorgung nach Betriebsunterbrechung

Nach Spannungszuschaltung starten die Module des HIMax Systems in zufälliger Reihenfolge. Dies gilt sowohl für die HIMax Module als auch für angeschlossenen Remote I/Os.

### 8.3.3 Anschließen einer redundanten Spannungsversorgung

Beim Anschließen einer redundanten Spannungsversorgung während des Betriebs ist wegen der möglichen hohen Stromstärken besondere Vorsicht geboten.

**⚠ WARNUNG**

**Personenschaden durch Überhitzung beim Anschluss einer redundanten Spannungsquelle möglich!**

**Vor dem Anschließen einer redundanten Spannungsquelle während des Betriebs unbedingt die Polarität prüfen!**

### 8.3.4 Laden von Betriebssystemen

Das Laden von Betriebssystemen ist in den Release-Notes zur jeweiligen Betriebssystem-Version beschrieben.

Alle Module des HIMax Systems enthalten Prozessoren und Betriebssysteme, welche die Module steuern. Die Betriebssysteme werden zusammen mit den Modulen geliefert. Im Zuge der Produktpflege entwickelt und verbessert HIMA die Betriebssysteme, welche mit Hilfe von SILworX in die Module geladen werden können.

Nachfolgende Tabelle listet alle HIMax Module und die zugehörigen Betriebssystem-Versionen (z. B.: **HIMAXIO\_HA1\_OS\_Vx.xx.ldb**) in Abhängigkeit des Hardware-Revisionsindex (HW-Rev.) auf.

E/A-Module	HW-Rev. xx <sup>1)</sup>	Dateiname des Betriebssystems beginnt mit:	PADT anschließen an:
X-AI 16 51	Alle	HIMaxIO_HA2 ...	Kommunikationsmodul Prozessormodul Systembusmodul, siehe Kapitel 5.1.1 Verwendung der Ethernet-Schnittstellen
X-AI 32 01	01, 10, 11, 12, 13	HIMaxIO_HA1 ...	
	14	HIMaxIO_HA3 ...	
X-AI 32 02	Alle	HIMaxIO_HA3 ...	
X-AI 32 51	Alle	HIMaxIO_HA2 ...	
X-AO 16 01	Alle	HIMaxIO_HA1 ...	
X-AO 16 51	Alle	HIMaxIO_HA2 ...	
X-CI 24 01	Alle	HIMaxIO_HA3 ...	
X-CI 24 51	Alle	HIMaxIO_HA2 ...	
X-DI 16 01	Alle	HIMaxIO_HA1 ...	
X-DI 32 01	01, 02, 10, 11	HIMaxIO_HA1 ...	
	12	HIMaxIO_HA3 ...	
X-DI 32 02	01, 02, 10, 11	HIMaxIO_HA1 ...	
	12	HIMaxIO_HA3 ...	
X-DI 32 03	Alle	HIMaxIO_HA1 ...	
X-DI 32 04	Alle	HIMaxIO_HA3 ...	
X-DI 32 05	Alle	HIMaxIO_HA3 ...	
X-DI 32 51	Alle	HIMaxIO_HA2 ...	
X-DI 32 52	Alle	HIMaxIO_HA2 ...	
X-DI 64 01	10	HIMaxIO_HA1 ...	
	11	HIMaxIO_HA3 ...	
X-DI 64 51	Alle	HIMaxIO_HA2 ...	
X-DO 12 01	Alle	HIMaxIO_HA1 ...	
X-DO 12 51	Alle	HIMaxIO_HA2 ...	
X-DO 12 02	Alle	HIMaxIO_HA1 ...	
X-DO 24 01	01, 02, 10, 11, 12, 13	HIMaxIO_HA1 ...	
	14	HIMaxIO_HA3 ...	
X-DO 24 02	Alle	HIMaxIO_HA1 ...	

E/A-Module	HW-Rev. xx <sup>1)</sup> )	Dateiname des Betriebssystems beginnt mit:	PADT anschließen an:
X-DO 32 01	10, 11	<b>HIMaxIO_HA1 ...</b>	Kommunikationsmodul Prozessormodul Systembusmodul, siehe Kapitel 5.1.1
	12	<b>HIMaxIO_HA3 ...</b>	
X-DO 32 51	Alle	<b>HIMaxIO_HA2 ...</b>	
X-HART 32 01	Alle	<b>HIMaxIO_HA3 ...</b>	
X-MIO 7/6 01	Alle	<b>HIMaxIO_HA3 ...</b>	
Prozessormodule	HW-Rev. xx <sup>1)</sup> )	Dateiname des Betriebssystems beginnt mit:	PADT anschließen an:
X-CPU 01	Alle	<b>HIMaxCPU_ ...</b>	Kommunikationsmodul Prozessormodul Systembusmodul, siehe Kapitel 5.1.1
X-CPU 31	Alle	<b>HIMaxCPU3x_ ...</b>	
<div>Hinweise:<ul style="list-style-type: none"><li>▪ Gemeinsamer Betrieb von Prozessormodulen mit unterschiedlichen Betriebssystem-Versionen ist nur während der Dauer der Aktualisierung zulässig!</li><li>▪ Bei Benutzung von <b>safeethernet</b> sind die Prozessormodule unmittelbar nacheinander zu aktualisieren, ohne dass dazwischen andere Aktionen erfolgen!</li></ul>In die Prozessormodule X-CPU 31 dürfen nur Betriebssystem-Versionen ≥ V6 geladen werden!</div>			
Systembusmodule	HW-Rev. xx <sup>1)</sup> )	Dateiname des Betriebssystems beginnt mit:	PADT anschließen an:
X-SB 01	Alle	<b>HIMaxSB_HA2 ...</b>	Kommunikationsmodul Prozessormodul Systembusmodul, siehe Kapitel 5.1.1
Kommunikationsmodul	HW-Rev. xx <sup>1)</sup> )	Dateiname des Betriebssystems beginnt mit:	PADT anschließen an:
X-COM 01	Alle	<b>HIMaxCOM_HA2 ...</b>	Prozessormodul
<sup>1)</sup> Der Hardware-Revisionsindex jedes Moduls ist mit der <b>Moduldaten-Übersicht</b> in SILworX auslesbar.			

Tabelle 39: In die Module zu ladende Betriebssysteme

### i

Vor der Aktualisierung von Betriebssystemen muss ein HiMax System in einem fehlerfreien Zustand sein!

Während des gesamten Aktualisierungsvorgangs dürfen keine anderen Aktionen am System erfolgen!

**Betriebsunterbrechung durch Ladevorgang möglich!**

**Betrieb eines funktionsfähigen, redundanten Moduls sicherstellen! Dieses erhält während des Ladens den Betrieb aufrecht.**

#### 8.3.4.1 Upgrade/Downgrade von Betriebssystemen

In sehr seltenen Fällen kann es sinnvoll sein, eine ältere Version eines Betriebssystems in ein Modul zu laden.

Wenn eine Steuerung längere Zeit unverändert gelaufen ist, und ein einzelnes Modul ausgetauscht werden muss, kann es angebracht sein, in das Ersatzmodul die ursprünglich verwandte Betriebssystem-Version zu laden. Diese passt möglicherweise besser zu den Versionen, mit der die übrigen Module arbeiten.



Das Laden einer älteren Betriebssystem-Version in ein Modul kann dazu führen, dass Daten im nicht-flüchtigen Speicher gelöscht werden, z. B. die Diagnosehistorie.

---

**i**

**Ein Upgrade der Prozessormodule auf BS-Version V10.X aus BS-Versionen < V6.30 muss über ein Zwischen-Upgrade V6.30, V7 oder V8 erfolgen!**

**Das Laden einer Betriebssystem-Version vor V3 ist zu vermeiden.**

---

### 8.3.5 Reparaturen

#### **HINWEIS**



**Fehlfunktion der Steuerung durch unzureichend durchgeführte Reparaturen!**

**Reparatur eines sicherheitsbezogenen HIMax Systems oder der darin enthaltenen Module nur durch HIMA.**

**Bei Eingriffen in das Gerät ist die funktionale Sicherheit nicht mehr gegeben und die Garantie sowie die Zertifizierung erlöschen.**

---

## 8.4 Besondere Betriebszustände

Das HIMax System kann die hier aufgeführten Betriebszustände erreichen, wenn Komponenten ausgefallen sind oder der Anwender die Betriebszustände absichtlich eingestellt hat.

In diesen Fällen sorgen die Betriebssysteme der Module dafür, dass das HIMax System in einen definierten Zustand gelangt.

### 8.4.1 Mono-Betrieb

Für den Start eines HIMax Systems mit nur Systembus A und einem Prozessormodul X-CPU 01 in Rack 0, Steckplatz 3 oder X-CPU 31 in Steckplatz 1 ist es notwendig, dass der Mono-Betrieb eingestellt ist.

#### Mono-Betrieb einstellen

1. Voraussetzungen erfüllen:
  - Nur ein Systembus aktiv.
  - Mono-Ressourcenkonfiguration geladen.
  - PES-Anwender mit Schreib-Berechtigung ist angemeldet.
  - *Online-Einstellungen erlaubt* ist ON.
2. Im Menu **Online->Inbetriebnahme** den Menüpunkt **Mono-/Redundanz-Betrieb einstellen** auswählen.
  - ☒ Das Dialogfenster *Mono-/Redundanz-Betrieb einstellen* öffnet sich.
3. Im Dialogfenster die Einstellung von *Redundanz* auf *Mono* ändern und mit **OK** abschließen.
  - Das HIMax System ist auf Mono-Betrieb eingestellt und kann mit nur einem Systembus und einem Prozessormodul starten.

Die Einstellung bleibt spannungsausfallsicher gespeichert.

Ist Mono-Betrieb eingestellt und nur ein Systembus zugänglich, dann startet das Prozessormodul in Rack 0, Steckplatz 3 allein. Dies gilt auch dann, wenn weitere Prozessormodule existieren und diese aktuelle Retain-Daten enthalten!

Ebenso startet das Modul X-CPU 31 auf Steckplatz 1 bei Mono-Betrieb allein, wenn auf Steckplatz 2 ein zweites X-CPU 31 vorhanden ist, zu dem keine Verbindung besteht.

Laden einer redundanten Ressource-Konfiguration in ein auf Mono-Betrieb eingestelltes System hat zur Folge, dass sich die Einstellung automatisch auf Redundanzbetrieb ändert.

HIMA empfiehlt, während der Inbetriebnahme eines Mono-Systems ein Systembusmodul vorübergehend in den Steckplatz 2 des Racks 0 zu stecken und auf *responsible* zu setzen. Dadurch wird Folgendes ermöglicht:

- Einfacher Ersatz des Systembusmoduls bei Defekt, auch bei Defekt von Steckplatz 1
- Erweiterung auf redundanten Betrieb bei laufendem System

### 8.4.2 Start mit nur einem responsible Systembus- oder Prozessormodul

Ein redundant konfiguriertes System läuft nach dem Einschalten der Versorgungsspannung nicht an, wenn nicht beide responsible Systembusmodule oder Prozessormodule (X-CPU 31) erreichbar und funktionsfähig sind.

Vor dem Start muss diese Einschränkung der Verfügbarkeit beseitigt werden, z. B. durch Austausch des defekten Moduls.

*Notfall-Mono-Systembetrieb* starten dient dazu, ein System im Ausnahmefall mit nur einem responsible Modul zu starten. Den Mono-Systembetrieb durch das Setzen des *Mono-Startup-Schalters* oder durch *Notfall-Mono-Systembetrieb starten* für ein Prozessormodul zu starten ist nur erlaubt, wenn sich kein Prozessormodul im Systembetrieb befindet und wenn kein anderes als das ausgewählte Prozessormodul bereit ist, Systembetrieb aufzunehmen. Im anderen Fall ist nicht garantiert, dass das ausgewählte Prozessormodul den Systembetrieb zuerst startet.

Weitere Details siehe SILworX Online-Hilfe.

#### 8.4.3 Auf die Racks 0 und 1 verteilte Prozessormodule X-CPU 01

Ein in hohem Maß redundanter Systemaufbau sieht folgendermaßen aus:

- Die Prozessormodule X-CPU 01 sind auf die Racks 0 und 1 verteilt.  
Diese Konfiguration ist nur mit X-CPU 01 möglich, nicht mit X-CPU 31!
- Die responsible Systembusmodule sind auf Rack 0, Steckplatz 1, und Rack 1, Steckplatz 2 verteilt.
- Die Art der Verkabelung der beiden Systembusse zwischen Rack 0 und Rack 1 schließt aus, dass eine Störung oder Unterbrechung innerhalb der Watchdog-Zeit zugleich auf beiden Systembussen auftreten kann.

Werden bei dieser Konfiguration die Verbindungen **beider** Systembusse zwischen den Racks 0 und 1 zugleich getrennt, dann könnten dadurch zwei HIMax Teilsysteme entstehen. Jedes Teilsystem wäre unabhängig vom anderen lauffähig und könnte unabhängig vom anderen den Prozess steuern. Dadurch könnten im Prozess riskante Zustände entstehen.

Um ein solches Szenario zu vermeiden, verhalten sich die Betriebssysteme folgendermaßen:

- Wird Systembus A vor Systembus B getrennt, geht Rack 0 in STOPP. Rack 1 bleibt in RUN.
- Wird Systembus B vor Systembus A getrennt, geht Rack 1 in STOPP. Rack 0 bleibt in RUN.

Das beschriebene Verhalten funktioniert nur, wenn weder ein Systembusmodul (X-SB 01) noch ein Prozessormodul (X-CPU 01) im Status *essential* ist. Ist ein Systembusmodul (X-SB 01) oder ein Prozessormodul (X-CPU 01) bereits im Status *essential*, dann wird das gesamte HIMax System in den sicheren Zustand überführt.

Für Wartungsarbeiten kann eine gezielte Abschaltung der Systembusse wie oben beschrieben genutzt werden, um jeweils ein Rack in RUN und das andere in STOP zu versetzen.

#### 8.4.4 Prozessormodule mit unterschiedlichen Projektkonfigurationen

Bei der Verwendung von Prozessormodulen, die bereits in anderen Anlagen benutzt waren, sollte sorgfältig vorgegangen werden.

Beim Neustart eines Systems mit Prozessormodulen, die unterschiedliche Projektkonfigurationen enthalten, hat der Anwender dafür zu sorgen, dass dasjenige Prozessormodul zuerst startet, das die zur Anlage passende Projektkonfiguration enthält.

Dazu sind die Prozessormodule, die möglicherweise eine andere Projektkonfiguration enthalten, erst nach dem Start des ersten Prozessormoduls in das Rack einzufügen. Sie synchronisieren sich auf und übernehmen die Projektkonfiguration vom ersten Prozessormodul.

Alternative: Mode-Schalter bei den Prozessormodulen, die möglicherweise eine andere Projektkonfiguration enthalten, auf *Init* einstellen. Nach dem Start des ersten Prozessormoduls den Mode-Schalter bei den übrigen auf *Run* einstellen.

#### 8.4.5 Autostart bei gestopptem System

Ist der Systemparameter *Autostart* gesetzt und das System in STOPP, führen folgende Aktionen dazu, dass das System – unbeabsichtigt - in RUN geht:

- Entfernen und erneutes Stecken der beiden Systembusmodule mit dem Attribut *responsible*.
- Abschalten und wieder Einschalten der Versorgungsspannung.

Der Übergang in RUN kann vermieden werden, indem der Systemparameter *Autostart* vor der Aktion online zurückgesetzt wird.

#### 8.4.6 Kommunikation zwischen zwei Prozessormodulen X-CPU 31

Besteht eine Störung in der Kommunikation zwischen zwei redundanten Prozessormodulen X-CPU 31, dann kann das Prozessormodul getauscht werden, das nicht *Essential* anzeigt.

Bleibt dabei die Kommunikation weiterhin gestört, dann kann das Prozessormodul, das *Essential* anzeigt, durch ein Kommando vom PADT gezielt aus dem Systembetrieb genommen werden. Das neu eingefügte Prozessormodul übernimmt den Systembetrieb und wird *Essential*.

Vor dem Stoppen des Prozessormoduls ist zu prüfen, ob das andere Prozessormodul den Systembetrieb allein aufrechterhalten kann. Dazu muss das neu eingefügte Prozessormodul intakte Verbindungen zu allen anderen ungestörten Modulen und zu den **safeethernet** Kommunikationspartnern haben.

## 9 Dokumentation

Das Kapitel Dokumentation listet die verfügbaren HIMax Dokumente in einer Tabelle auf.

### 9.1 Dokumentation HIMax System

Folgende Dokumente stehen zur Verfügung:

Dokument	Inhalt	Dokumenten-Nr.
Systemhandbuch	Hardware-Beschreibung des HIMax Systems	HI 801 000 D
Sicherheitshandbuch	Sicherheitsfunktionen des HIMax Systems	HI 801 002 D
Wartungshandbuch	Wartungsmaßnahmen des HIMax Systems	HI 801 170 D
X-BASE PLATE	Basisträger für X-CPU 01	HI 801 024 D
X-BASE PLATE	Basisträger für X-CPU 31	HI 801 370 D
X-FAN	Systemlüfter	HI 801 032 D
X-CPU 01	Prozessormodul, SIL 3	HI 801 008 D
X-CPU 31	Prozessormodul mit Systembus, SIL 3	HI 801 354 D
X-COM 01	Kommunikationsmodul	HI 801 010 D
X-SB 01	Systembusmodul, SIL 3	HI 801 006 D
X-AI 16 51	Analoges Eingangsmodul, 16 Kanäle, SIL 1	HI 801 178 D
X-AI 32 01	Analoges Eingangsmodul, 32 Kanäle, SIL 3	HI 801 020 D
X-AI 32 02 SOE	Analoges Eingangsmodul, 32 Kanäle, Ereigniserfassung, SIL 3	HI 801 054 D
X-AI 32 51	Analoges Eingangsmodul, 32 Kanäle	HI 801 180 D
X-AO 16 01	Analoges Ausgangsmodul, 16 Kanäle, SIL 3	HI 801 110 D
X-CI 24 01	Zähler-Eingangsmodul, 24 Kanäle, SIL 3	HI 801 112 D
X-CI 24 51	Zähler-Eingangsmodul, 24 Kanäle	HI 801 188 D
X-DI 16 01	Digitales Eingangsmodul, 16 Kanäle; SIL 3	HI 801 056 D
X-DI 32 01	Digitales Eingangsmodul, 32 Kanäle, SIL 3	HI 801 014 D
X-DI 32 02	Digitales Eingangsmodul, 32 Kanäle für Initiatoren, SIL 3	HI 801 016 D
X-DI 32 03	Digitales Eingangsmodul, 32 Kanäle, SIL 3	HI 801 058 D
X-DI 32 04 SOE	Digitales Eingangsmodul, 32 Kanäle, Ereigniserfassung, SIL 3	HI 801 050 D
X-DI 32 05 SOE	Digitales Eingangsmodul, 32 Kanäle für Initiatoren, Ereigniserfassung, SIL 3	HI 801 052 D
X-DI 32 51	Digitales Eingangsmodul, 32 Kanäle, SIL 3	HI 801 172 D
X-DI 32 52	Digitales Eingangsmodul, 32 Kanäle für Initiatoren, SIL 3	HI 801 174 D
X-DI 64 01	Digitales Eingangsmodul, 32 Kanäle, SIL 3	HI 801 092 D
X-DI 64 02	Digitales Eingangsmodul, 32 Kanäle, SIL 3	HI 801 176 D
X-DO 12 01	Digitales Relais-Ausgangsmodul, 12 Kanäle, SIL 3	HI 801 022 D
X-DO 12 02	Digitales Ausgangsmodul, 12 Kanäle, SIL 3	HI 801 098 D
X-DO 12 51	Digitales Relais-Ausgangsmodul, 12 Kanäle	HI 801 184 D
X-DO 24 01	Digitales Ausgangsmodul, 24 Kanäle, SIL 3	HI 801 018 D
X-DO 24 02	Digitales Ausgangsmodul, 24 Kanäle, SIL 3	HI 801 094 D
X-DO 32 01	Digitales Ausgangsmodul, 32 Kanäle, SIL 3	HI 801 096 D
X-DO 32 51	Digitales Ausgangsmodul, 32 Kanäle	HI 801 182 D
X-HART 32 01	HART-Modul	HI 801 014 D
X-MIO 6/7 01	Überdrehzahl-Schutzmodul	HI 801 304 D

Dokument	Inhalt	Dokumenten-Nr.
X-FTA 001 01	Field Termination Assemblies für die verschiedenen Module	HI 801 114 D
X-FTA 001 02		HI 801 130 D
X-FTA 002 01		HI 801 116 D
X-FTA 002 02		HI 801 118 D
X-FTA 003 02		HI 801 120 D
X-FTA 005 02		HI 801 124 D
X-FTA 006 01		HI 801 126 D
X-FTA 006 02		HI 801 128 D
X-FTA 007 02		HI 801 132 D
X-FTA 008 02		HI 801 134 D
X-FTA 009 02		HI 801 136 D
SILworX Erste-Schritte-Handbuch	Einführung in SILworX	HI 801 102 D
SILworX Online-Hilfe (OLH)	SILworX Bedienung	
Kommunikationshandbuch	Beschreibung der <b>safeethernet</b> Kommunikation und Auflistung der verfügbaren Protokolle.	HI 801 100 D

Tabelle 40: Übersicht HIMax Dokumente

Die aktuellen Handbücher können über die E-Mail-Adresse [documentation@hima.com](mailto:documentation@hima.com) angefragt werden. Für registrierte Kunden stehen die Produktdokumentationen im HIMA Extranet als Download zur Verfügung.

## Anhang

### Applikationsbeispiele

Dieses Kapitel zeigt Beispiele für den Aufbau von HIMax Systemen. Die E/A- und Kommunikationsmodule sind nicht betrachtet. Sie werden nach Bedarf auf die verbleibenden Steckplätze gesteckt.

Anstelle der Basisträger mit 10 Steckplätzen in den Beispielen sind – je nach Bedarf – Basisträger mit 15 oder 18 Steckplätzen verwendbar.

#### Kleines System

Dieses redundante System besteht aus einem Basisträger mit zwei Prozessormodulen. Der Basisträger hat die Rack-ID 0.

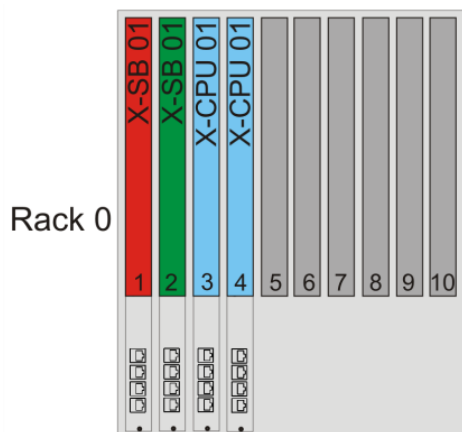


Bild 31: Kleines HIMax System: ein Basisträger, zwei Prozessormodule

#### Minimalsystem

Dieses System ohne Redundanz stellt das absolute Minimum dar: der Basisträger 0, ein Prozessormodul, ein Systembusmodul. Nur Systembus A wird verwendet.

Der Steckplatz 2 muss wegen der Strömung der Kühlluft ein Leermodul enthalten. Es ist nicht möglich, auf den Steckplatz 2 ein E/A- oder Kommunikationsmodul zu stecken.

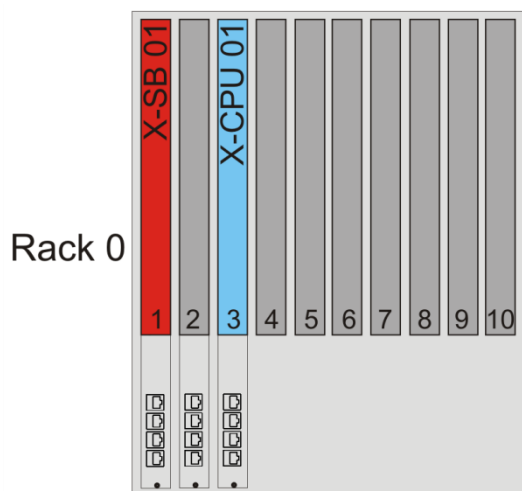


Bild 32: Minimalsystem ohne Redundanz

**i**

HIMA empfiehlt, beide Systembusse zu verwenden.

### Verteilte Redundanz

Dieses System enthält vier redundante Prozessormodule, die auf die zwei Basisträger 0 und 1 verteilt sind.

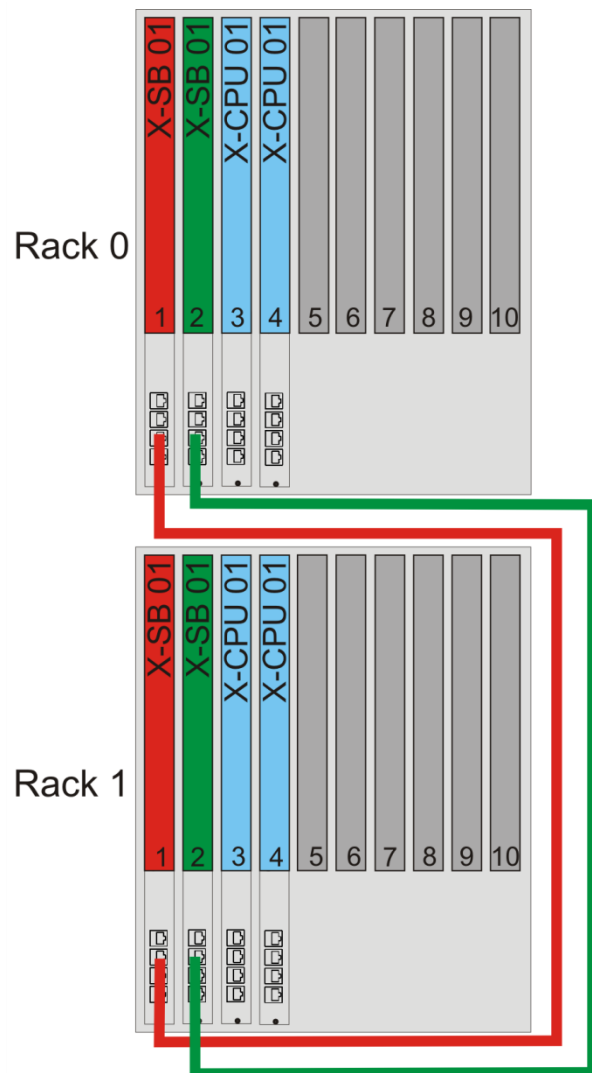


Bild 33: HIMax System mit verteilter Redundanz



**Glossar**

Begriff	Beschreibung
AI	Analog Input: Analoger Eingang
AO	Analog Output: Analoger Ausgang
ARP	Address Resolution Protocol: Netzwerkprotokoll zur Zuordnung von Netzwerkadressen zu Hardware-Adressen
COM	Kommunikation (Modul)
CRC	Cyclic Redundancy Check: Prüfsumme
DI	Digital Input: Digitaler Eingang
DO	Digital Output: Digitaler Ausgang
EMV	Elektromagnetische Verträglichkeit
EN	Europäische Normen
ESD	Electrostatic Discharge: Elektrostatische Entladung
FB	Feldbus
FBS	Funktionsbausteinsprache
HW	Hardware
ICMP	Internet Control Message Protocol: Netzwerkprotokoll für Status- und Fehlermeldungen
IEC	Internationale Normen für die Elektrotechnik
LS/LB	Leitungsschluss/Leitungsbruch
MAC	Media Access Control: Hardware-Adresse eines Netzwerkanschlusses
PADT	Programming and Debugging Tool (nach IEC 61131-3): PC mit SILworX
PELV	Protective Extra Low Voltage: Funktionskleinspannung mit sicherer Trennung
PES	Programmable Electronic System: Programmierbares Elektronisches System
R	Read: Auslesen einer Variablen
Rack-ID	Identifikation eines Basisträgers (Nummer)
rückwirkungsfrei	Eingänge sind für rückwirkungsfreien Betrieb ausgelegt und können in Schaltungen mit Sicherheitsfunktionen eingesetzt werden.
R/W	Read/Write: Spaltenüberschrift für Art von Systemvariable
SB	Systembus (-modul)
SELV	Safety Extra Low Voltage: Schutzkleinspannung
SFF	Safe Failure Fraction: Anteil der sicher beherrschbaren Fehler
SIL	Safety Integrity Level (nach IEC 61508)
SILworX	Programmierwerkzeug
SNTP	Simple Network Time Protocol (RFC 1769)
SRS	System.Rack.Slot: Adressierung eines Moduls
SSL	Secure Sockets Layer, siehe TLS
SW	Software
TLS	Transport Layer Security: Hybrides Verschlüsselungsprotokoll
TMO	Timeout
W	Write: Variable wird mit Wert versorgt, z. B. vom Anwenderprogramm
WD	Watchdog: Funktionsüberwachung für Systeme. Signal für fehlerfreien Prozess
WDZ	Watchdog-Zeit
ws	Scheitelwert der Gesamt-Wechselspannungskomponente

**Abbildungsverzeichnis**

<b>Bild 1:</b>	<b>Systemübersicht</b>	<b>18</b>
<b>Bild 2:</b>	<b>Aufbau eines Basisträgers</b>	<b>20</b>
<b>Bild 3:</b>	<b>Reihenfolge der Basisträger am Systembus</b>	<b>24</b>
<b>Bild 4:</b>	<b>Systembus in Netzstruktur</b>	<b>26</b>
<b>Bild 5:</b>	<b>Maximale Ausdehnung bei Standardwert der Latenzzeit</b>	<b>30</b>
<b>Bild 6:</b>	<b>Maximale Entfernung zwischen Prozessormodulen bei Standardwert der Latenzzeit</b>	<b>31</b>
<b>Bild 7:</b>	<b>Verbindung zweier Racks durch einen Lichtwellenleiter</b>	<b>32</b>
<b>Bild 8:</b>	<b>Beispiel zur Berechnung der Systembus-Latenzzeit</b>	<b>34</b>
<b>Bild 9:</b>	<b>Einsatz von Prozessormodulen X-CPU 31</b>	<b>40</b>
<b>Bild 10:</b>	<b>Transiente Störung</b>	<b>45</b>
<b>Bild 11:</b>	<b>Anstehende Störung führt zu sicherer Reaktion</b>	<b>46</b>
<b>Bild 12:</b>	<b>Wirkrichtungen bei der Störaustastung und der Ausgangs-Störaustastung</b>	<b>47</b>
<b>Bild 13:</b>	<b><i>Program_CycleDuration</i> und <i>Program_ExecutionDuration</i></b>	<b>76</b>
<b>Bild 14:</b>	<b>Reservezeiten innerhalb des Zyklus</b>	<b>89</b>
<b>Bild 15:</b>	<b>Ablauf des CPU-Zyklus bei Multitasking</b>	<b>91</b>
<b>Bild 16:</b>	<b>Multitasking Mode 1</b>	<b>94</b>
<b>Bild 17:</b>	<b>Multitasking Mode 2</b>	<b>95</b>
<b>Bild 18:</b>	<b>Multitasking Mode 3</b>	<b>96</b>
<b>Bild 19:</b>	<b>Beschaltung 1 - einfaches Connector Board mit Schraubklemmen</b>	<b>118</b>
<b>Bild 20:</b>	<b>Beschaltung 2 - redundantes Connector Board mit Schraubklemmen</b>	<b>119</b>
<b>Bild 21:</b>	<b>Beschaltung 3 - Mono-Connector Board mit Systemkabel</b>	<b>120</b>
<b>Bild 22:</b>	<b>Beschaltung 4 - redundantes Connector Board mit Systemkabel</b>	<b>121</b>
<b>Bild 23:</b>	<b>Erdungsverbindungen im Schaltschrank</b>	<b>123</b>
<b>Bild 24:</b>	<b>Erdung und Schirmung des 19"-Schaltschranks</b>	<b>124</b>
<b>Bild 25:</b>	<b>Erdungsverbindungen für den Basisträger</b>	<b>125</b>
<b>Bild 26:</b>	<b>Erdanschlüsse mehrerer Schaltschränke</b>	<b>126</b>
<b>Bild 27:</b>	<b>Einsetzen des Mono Connector Boards, exemplarisch</b>	<b>128</b>
<b>Bild 28:</b>	<b>Festschrauben des Mono Connector Boards, exemplarisch</b>	<b>129</b>
<b>Bild 29:</b>	<b>Anordnung von FTAs im Schaltschrank bei Systemkabel-Einführung von oben</b>	<b>130</b>
<b>Bild 30:</b>	<b>Anordnung von FTAs im Schaltschrank bei Systemkabel-Einführung von unten</b>	<b>131</b>
<b>Bild 31:</b>	<b>Kleines HIMax System: ein Basisträger, zwei Prozessormodule</b>	<b>151</b>
<b>Bild 32:</b>	<b>Minimalsystem ohne Redundanz</b>	<b>151</b>
<b>Bild 33:</b>	<b>HIMax System mit verteilter Redundanz</b>	<b>152</b>

**Tabellenverzeichnis**

<b>Tabelle 1:</b>	<b>Standardwerte der maximalen Systembus-Latenzzeit</b>	<b>29</b>
<b>Tabelle 2:</b>	<b>Werte für die Berechnung der Systembus-Latenzzeit</b>	<b>35</b>
<b>Tabelle 3:</b>	<b>Identifikation eines Moduls durch System.Rack.Slot</b>	<b>38</b>
<b>Tabelle 4:</b>	<b>Empfohlene Steckplatzpositionen von Prozessormodulen</b>	<b>39</b>
<b>Tabelle 5:</b>	<b>Zustände des Betriebssystems, Erreichen der Zustände</b>	<b>42</b>
<b>Tabelle 6:</b>	<b>Zustände des Betriebssystems, mögliche Eingriffe durch den Anwender</b>	<b>43</b>
<b>Tabelle 7:</b>	<b>Beispiele zur Berechnung von minimaler und maximaler Störaustastzeit</b>	<b>44</b>
<b>Tabelle 8:</b>	<b>Unterstützte Variablentypen</b>	<b>57</b>
<b>Tabelle 9:</b>	<b>Systemvariable auf unterschiedlichen Projektebenen</b>	<b>58</b>
<b>Tabelle 10:</b>	<b>Die Systemparameter der Ressource</b>	<b>61</b>
<b>Tabelle 11:</b>	<b>Einstellungen Sollzykluszeit-Modus</b>	<b>63</b>
<b>Tabelle 12:</b>	<b>Standardwerte der maximalen Systembus-Latenzzeit</b>	<b>66</b>
<b>Tabelle 13:</b>	<b>Systemvariablen des Racks</b>	<b>67</b>
<b>Tabelle 14:</b>	<b>Eingangsvariablen</b>	<b>71</b>
<b>Tabelle 15:</b>	<b>Zuordnung des Index zu Steckplätzen der Prozessormodule X-CPU 01</b>	<b>71</b>
<b>Tabelle 16:</b>	<b>Systemparameter des Anwenderprogramms</b>	<b>74</b>
<b>Tabelle 17:</b>	<b>Die lokalen Systemvariablen des Anwenderprogramms</b>	<b>75</b>
<b>Tabelle 18:</b>	<b>Parameter für boolesche Ereignisse</b>	<b>80</b>
<b>Tabelle 19:</b>	<b>Parameter für skalare Ereignisse</b>	<b>83</b>
<b>Tabelle 20:</b>	<b>Register Eigenschaften</b>	<b>83</b>
<b>Tabelle 21:</b>	<b>t<sub>CycleAdjust</sub> abhängig von der Version des CPU-Betriebssystems</b>	<b>89</b>
<b>Tabelle 22:</b>	<b>Bestimmung von t<sub>EA-Phasen-Timeout</sub></b>	<b>89</b>
<b>Tabelle 23:</b>	<b>Für Multitasking einstellbare Parameter</b>	<b>92</b>
<b>Tabelle 24:</b>	<b>Aufbau eines Zyklus des Prozessormoduls</b>	<b>97</b>
<b>Tabelle 25:</b>	<b>Reload nach Änderungen</b>	<b>102</b>
<b>Tabelle 26:</b>	<b>Werkseitige Zugriffsrechte für PADT und PES</b>	<b>106</b>
<b>Tabelle 27:</b>	<b>Zugriffsarten der PADT-Benutzerverwaltung</b>	<b>107</b>
<b>Tabelle 28:</b>	<b>Zugriffsarten der PES-Benutzerverwaltung</b>	<b>108</b>
<b>Tabelle 29:</b>	<b>Blinkfrequenzen der LEDs</b>	<b>111</b>
<b>Tabelle 30:</b>	<b>In der Diagnosehistorie maximal gespeicherte Einträge pro Modultyp</b>	<b>111</b>
<b>Tabelle 31:</b>	<b>Datenfelder einer Diagnosemeldung</b>	<b>112</b>
<b>Tabelle 32:</b>	<b>Diagnoseinformationen in der Online-Ansicht des Hardware-Editors</b>	<b>114</b>
<b>Tabelle 33:</b>	<b>Umgebungsbedingungen</b>	<b>115</b>
<b>Tabelle 34:</b>	<b>Dimensionierung einer HIMax Steuerung</b>	<b>116</b>
<b>Tabelle 35:</b>	<b>Erdungsverbindungen</b>	<b>125</b>
<b>Tabelle 36:</b>	<b>Definitionen zur Berechnung der Verlustleistung</b>	<b>132</b>
<b>Tabelle 37:</b>	<b>Aufstellungsarten</b>	<b>132</b>
<b>Tabelle 38:</b>	<b>Temperaturzustände</b>	<b>133</b>
<b>Tabelle 39:</b>	<b>In die Module zu ladende Betriebssysteme</b>	<b>144</b>



## Index

Alarm (siehe Ereignis).....	49	Lizensierung	
analoge Ausgänge		Protokolle .....	51
Verwendung .....	78	Low Demand Mode.....	98
analoge Eingänge		Programmierung .....	56
Verwendung .....	77	Rack-ID	
Arbeitsstromprinzip .....	13	zuweisen .....	139
Automation Security.....	15	Reaktionszeit .....	97
Basisträger-Typen.....	19	Redundanz .....	53
Betriebssystem laden.....	143	Ein-/Ausgangsmodule .....	54
Blitzschutz.....	127	Kommunikation.....	55
Diagnose .....	111	Prozessormodul .....	53
Historie .....	111	Spannungsversorgung .....	55
Diagnosemeldung		Systembus .....	54
E/A-Modul .....	112	Reserve-Modul .....	54
digitale Ausgänge		Reservezeit.....	89
Verwendung .....	78	Ruhestromprinzip.....	13
digitale Eingänge		SILworX .....	56
Verwendung .....	76	Steuerung abschließbar machen .....	72
Erdung.....	122	Störungen .....	141
Ereignis		Systembus .....	22
allgemein.....	49	Ausdehnung .....	28
Aufzeichnung .....	50	Ausdehnung Standard.....	30
Bildung .....	49	Systembus-Latenzzeit.....	28
Definition .....	79	Systembus-Latenzzeit, maximale	
ESD-Schutz .....	14	Berechnung.....	32
Hardware-Editor.....	67	Standardwerte .....	29
High Demand Mode .....	98	Technische Daten .....	115
Inbetriebnahme		Temperaturüberwachung .....	133
Schaltschrank.....	134	Temperaturzustand.....	133
Installation.....	117	Umgebungsbedingungen .....	115
Kommunikationszeitscheibe.....	64	Wärmeabfuhr .....	132
Konfiguration laden		Wartung .....	141
Download .....	99	Zählereingänge	
Reload.....	99	Verwendung .....	77
Leermodul.....	21		


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](mailto:info@hima.com)

Erfahren Sie online mehr über HIMax:

 [www.hima.com/de/produkte-services/himax/](http://www.hima.com/de/produkte-services/himax/)