TR electronic

ETHERNET POWERLINK certified product

open SAFETY certified product

Absolute Encoder CDx-75 POWERLINK/openSAFETY

Parametrierung mit B&R Steuerungssystem /

Parameterization with B&R control system

X20 CPU

CDH 75 M

CDV 75 M

DIN EN 61508: **SIL CL3**
DIN EN ISO 13849: **PL e**

_Sicherheitsprogramm
_Konfiguration
_Zugriff auf Datenkanäle
_Parametrierung

_Safety Program
_Configuration
_Access to data channels
_Parameterization

Technical Information

TR - ECE - TI - DGB - 0264 - 07 06/18/2019

TR-Electronic GmbH
D-78647 Trossingen
Eglishalde 6
Tel.: (0049) 07425/228-0
Fax: (0049) 07425/228-33
E-mail: info@tr-electronic.de
http://www.tr-electronic.de

Urheberrechtsschutz

Änderungsvorbehalt

Jegliche Änderungen, die dem technischen Fortschritt dienen, vorbehalten.

Dokumenteninformation

Ausgabe-/Rev.-Datum:	06/18/2019
Dokument-/Rev.-Nr.:	TR - ECE - TI - DGB - 0264 - 07
Dateiname:	TR-ECE-TI-DGB-0264-07.docx
Verfasser:	MÜJ

Schreibweisen

Kursive oder **fette** Schreibweise steht für den Titel eines Dokuments oder wird zur Hervorhebung benutzt.

`Courier`-Schrift zeigt Text an, der auf dem Bildschirm sichtbar ist und Software bzw. Menüauswahlen von Software.

″ < > ″ weist auf Tasten der Tastatur Ihres Computers hin (wie etwa <RETURN>).

Marken

Genannte Produkte, Namen und Logos dienen ausschließlich Informationszwecken und können Warenzeichen ihrer jeweiligen Eigentümer sein, ohne dass eine besondere Kennzeichnung erfolgt.

Inhaltsverzeichnis

Änderungs-Index

Änderung	Datum	Index
Erstausgabe	16.12.14	00
Beispiel für Preset ergänzt	22.01.15	01
Hinweise für den Betrieb mit einer SLX CPU von B&R	09.04.15	02
Entfernt: Hinweise für den Betrieb mit einer SLX CPU von B&R	25.06.15	03
Anpassungen unter Kapitel „Preset-Durchführung“ Neue Funktionsbaustein – Beispiele: - Sichere Geschwindigkeit (SLS) - Sichere Stillstandserfassung - Sichere Positionserfassung - Sichere Richtungserfassung	02.10.15	04
Kapitel „Typumwandlung der Mess-System - Istwerte im SafeDESIGNER“ ergänzt	18.10.16	05
Paßwortangabe unter Kap. 4.1.1 Zugriffschutz	09.03.18	06
Erweiterung für Update Drehgeber mit openSAFETY-Stack V1.5; Verwendung von XDD-/OSDD-Gerätebeschreibung im Automation Studio V4.5	18.06.19	07

1 Glossar

Name	Beschreibung
AS	Automation Studio von B&R: Entwicklungsumgebung für POWERLINK-Projekte
B&R	Bernecker + Rainer Industrie-Elektronik GmbH
EPL	Ethernet PowerLink
FB(s)	Funktionsbaustein(e)
FBS	Programmiersprache, Funktionsbaustein-Sprache
FU(s)	Funktion(en)
graue Daten	einkanalige Istwerte über POWERLINK, nicht sicherheitsgerichtet
KOP	Programmiersprache, Kontaktplan
openSAFETY	offener und busunabhängiger Sicherheitsstandard für alle Industrial-Ethernet-Lösungen
POWERLINK CN	Controlled Node: Slave-Teilnehmer in einem POWERLINK-Netzwerk
POWERLINK MN	Managing Node: Initiiert die POWERLINK-Kommunikation und parametriert die CN
SCM	Safety Configuration Manager: Initialisiert die SNs bei einem openSAFETY-Projekt via dem zugrunde liegenden Feldbus.
SL	SafeLogic CPU von B&R
SN	Safety Node: Slave-Teilnehmer in einem openSAFETY-Projekt
XML	EXtensible Markup Language

2 Allgemeines

Die vorliegende „Technische Information“ beinhaltet folgende Themen:

- Sicherheitsprogramm erstellen
- Festlegen der sicheren Parameter
- Verwendung des sicherheitsgerichteten Datenkanals

Die „Technische Information“ kann separat angefordert werden.

2.1 Geltungsbereich

Diese „Technische Information“ gilt ausschließlich für folgende Mess-System-Baureihen mit ***POWERLINK*** Schnittstelle und ***openSAFETY*** Profil in Verbindung mit einer B&R X20 Steuerung:

- CDV-75
- CDH-75

Die Produkte sind durch aufgeklebte Typenschilder gekennzeichnet und sind Bestandteil einer Anlage.

Es gelten somit zusammen folgende Dokumentationen:

- B&R: X20 System Anwenderhandbuch,
  Bestell-Nr.: MAX20-GER
- B&R: Integrated Safety Technology Anwenderhandbuch,
  Bestell-Nr.: MASAFETY-GER
- B&R: X20 Datenblatt V1.51 X20SL80xx
- anlagenspezifische Betriebsanleitungen des Betreibers
- Sicherheitshandbuch TR-ECE-BA-D-0107
- schnittstellenspezifische Benutzerhandbuch TR-ECE-BA-D-0110
- und diese optionale „Technische Information“

3 Sicherheitshinweise

3.1 Symbol- und Hinweis-Definition

⚠ GEFAHR	bedeutet, dass Tod oder schwere Körperverletzung eintreten wird, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
⚠WARNUNG	bedeutet, dass Tod oder schwere Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
⚠VORSICHT	bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
ACHTUNG	bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.
	bezeichnet wichtige Informationen bzw. Merkmale und Anwendungstipps des verwendeten Produkts.

3.2 Organisatorische Maßnahmen

Das mit Tätigkeiten am Mess-System beauftragte Personal muss vor Arbeitsbeginn das Sicherheitshandbuch TR-ECE-BA-D-0107, insbesondere das Kapitel „Grundlegende Sicherheitshinweise“, gelesen und verstanden haben.

3.3 Personalqualifikation

Die Konfiguration des Mess-Systems darf nur von qualifiziertem Fachpersonal durchgeführt werden, siehe B&R Anwenderhandbuch.

3.4 Nutzungsbedingungen der Softwarebeispiele

⚠WARNUNG

Für das fehlerfreie Funktionieren des Sicherheitsprogrammes und der Anwendungsbeipiele übernimmt die TR-Electronic GmbH keine Haftung und keine Gewährleistung.

ACHTUNG

Die zum Download angebotenen Softwarebeispiele dienen ausschließlich zu Demonstrationszwecken, der Einsatz durch den Anwender erfolgt auf eigene Gefahr.

Das Kennwort für die Beispiele ist immer „abc123“.
Dies gilt auch für das zip-Archiv der Softwarebeispiele.

4 Sicherheitsprogramm erstellen - Konfigurationsbeispiele

Im folgenden werden zwei Beispiele aufgeführt. Im ersten Beispiel wird die Legacy-Gerätebeschreibung verwendet, die als Update im AS installiert werden kann. Erst ab der Automation Studio Version 4.5 kann die aktuelle Geräteschreibung in Form einer XDD-/OSDD-Kombination verwendet werden. Dies stellt das zweite Beispiel dar.

4.1 Legacy Gerätebeschreibung (Update)

Dieses Kapitel beschreibt die Vorgehensweise bei der Erstellung eines Beispiel-Sicherheitsprogramms mit Verwendung der B&R Projektierungssoftware `Automation Studio` (V4.0.18.71) und dem Optionspaket `SafeDESIGNER` (V3.0.16).

Das Sicherheitsprogramm wird im `SafeDESIGNER` erstellt, welcher einen Code-Editor zur Entwicklung des Programms für die Sicherheitssteuerung enthält. Dies erfolgt mit Hilfe der graphischen Programmiersprachen FBS und KOP.

Die Ausführung des Sicherheitsprogramms erfolgt auf einer `X20 SafeLOGIC` (`SL8010`). Diese nimmt als normaler POWERLINK-CN an der Feldbuskommunikation teil und ist selbst wiederrum der openSAFETY SCM.

Als MN bei POWERLINK kommt eine `X20 CP1584` zum Einsatz. Sie enthält eine POWERLINK-Anwendung, die ebenfalls die Mess-System - Informationen (u.a. die Istwerte „grauer Kanal“) auswerten kann.

4.1.1 Zugriffschutz

Der Zugang zum `SafeDESIGNER`-Projekt ist durch eine Paßwortabfrage gesichert. Es existiert jeweils ein Paßwort für die Entwicklung und für die Inbetriebnahme. Der Zugang zur `SL8010`, z.B. zur Programmierung, ist ebenfalls durch ein Paßwort geschützt.

Das Paßwort heißt: `abc123`

4.1.2 Voraussetzungen

⚠WARNUNG ***Gefahr der Außerkraftsetzung der fehlersicheren Funktion durch unsachgemäße Projektierung des Sicherheitsprogramms!***

- Die Erstellung des Sicherheitsprogramms darf nur in Verbindung mit der von B&R zur Software bzw. Hardware mitgelieferten System-dokumentation erfolgen.
- Nachfolgende Beschreibungen beziehen sich auf den reinen Ablauf, ohne dabei alle Hinweise aus den B&R-Handbüchern mit zu berücksichtigen.
  Die in den B&R-Handbüchern gegebenen Informationen, Hinweise, insbesondere die Sicherheitshinweise und Warnungen, sind daher zwingend zu beachten und einzuhalten.
- Die aufgezeigte Projektierung ist als Beispiel aufzufassen. Der Anwender ist daher verpflichtet, die Verwendbarkeit der Projektierung für seine Applikation zu überprüfen und anzupassen. Dazu gehören auch die Auswahl der geeigneten sicherheitsgerichteten Hard-warekomponenten, sowie die notwendigen Softwarevoraussetzungen.

Für das Konfigurationsbeispiel benutzte Software-Komponenten:

- `Automation Studio` V4.0.18.71
- `SafeDESIGNER` V3.0.16.262

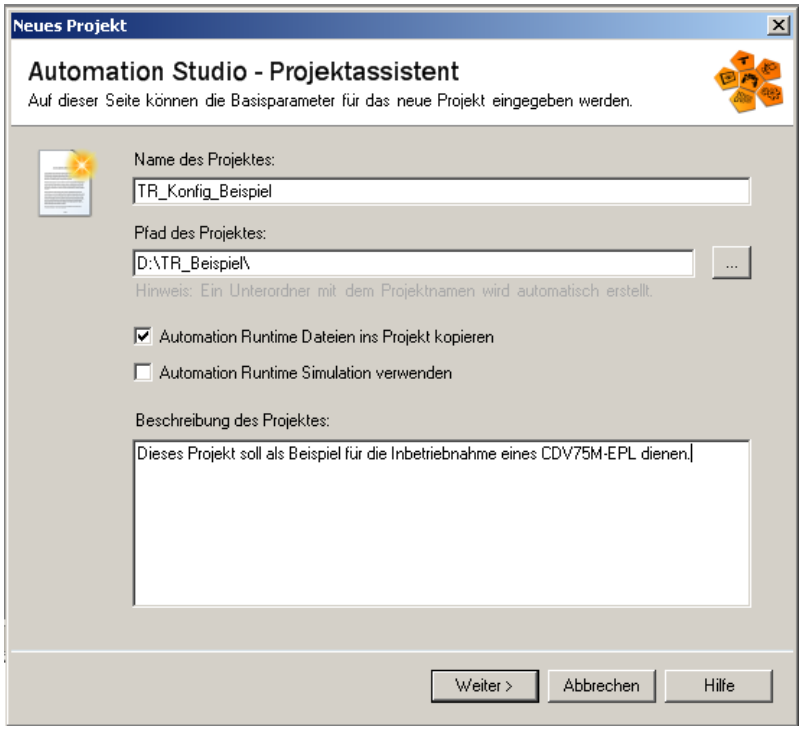
Die Verwendung nachfolgender Versionen ist ebenfalls möglich. Der Aufbau wurde auch mit einer `Automation Studio` Version 4.1.4.401 und einer `SafeDESIGNER` Version 4.1.0.320 in Betrieb genommen.

Für das Konfigurationsbeispiel benutzte Hardware-Komponenten:

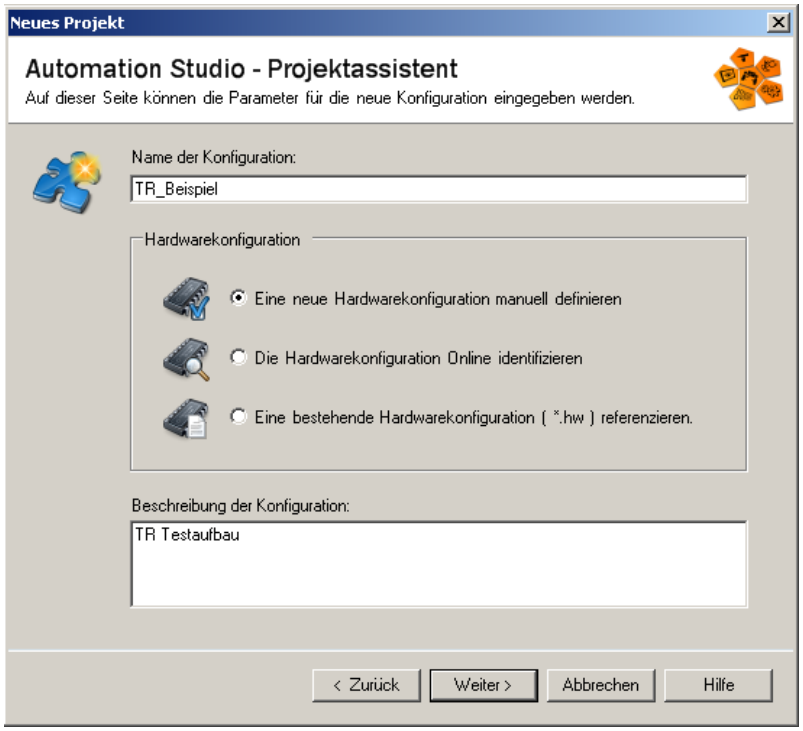
- POWERLINK-MN: `X20CP1584` mit `X20IF1082-2`
- openSAFETY SCM: `X20SL8010`

4.1.3 Hardware-Konfiguration

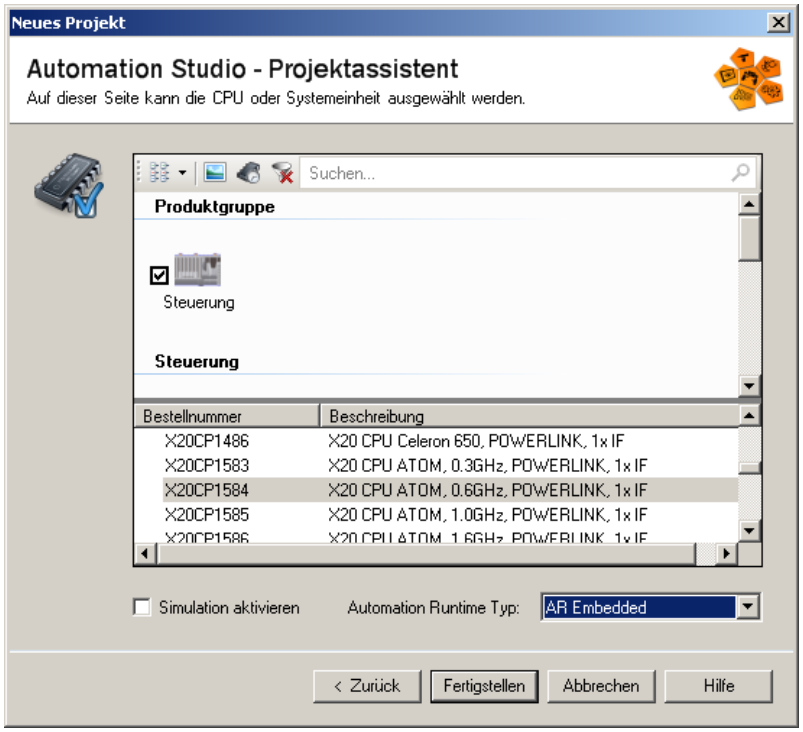
➢ `Automation Studio` starten und ein neues Projekt anlegen.



➢ Es wird eine neue Hardwarekonfiguration erstellt:



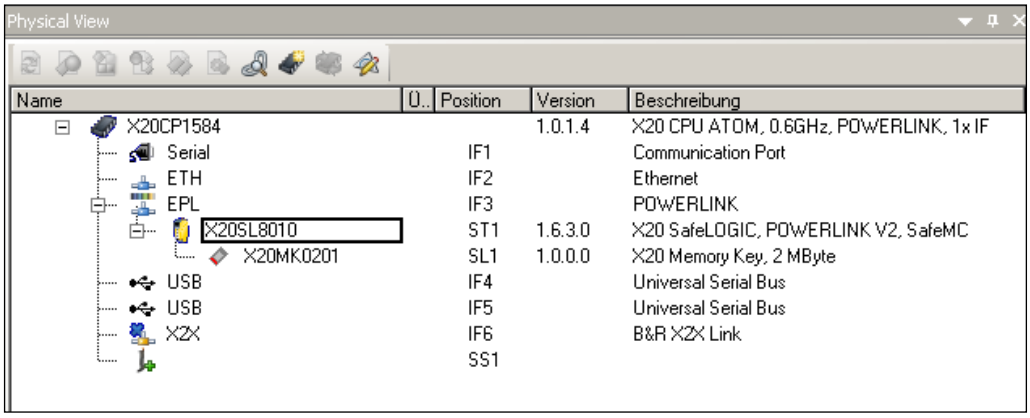
- Abschließend korrekte CPU auswählen: `X20CP1584`



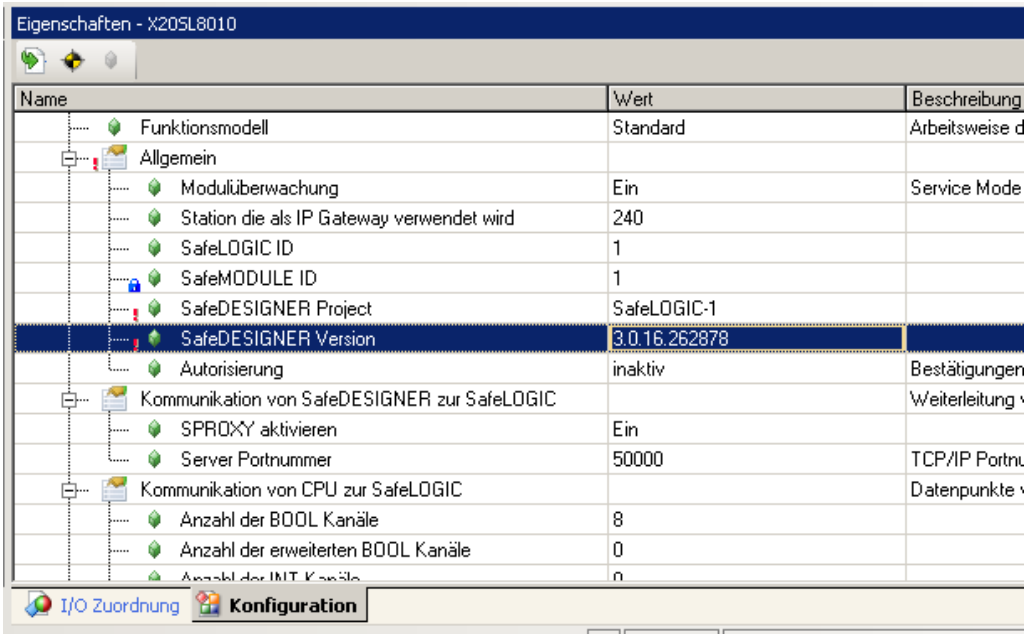
- Unter dem Menüpunkt `-> Projekt -> Runtime Versionen ändern...` das Safety Release 1.6 auswählen.

Das Mess-System benötigt das Safety Release 1.6 oder höher.

- SCM zum EPL-Netz hinzufügen, indem man das entsprechende Gerät aus dem Hardware Katalog auf die POWERLINK-Schnittstelle im `Physical View` zieht:

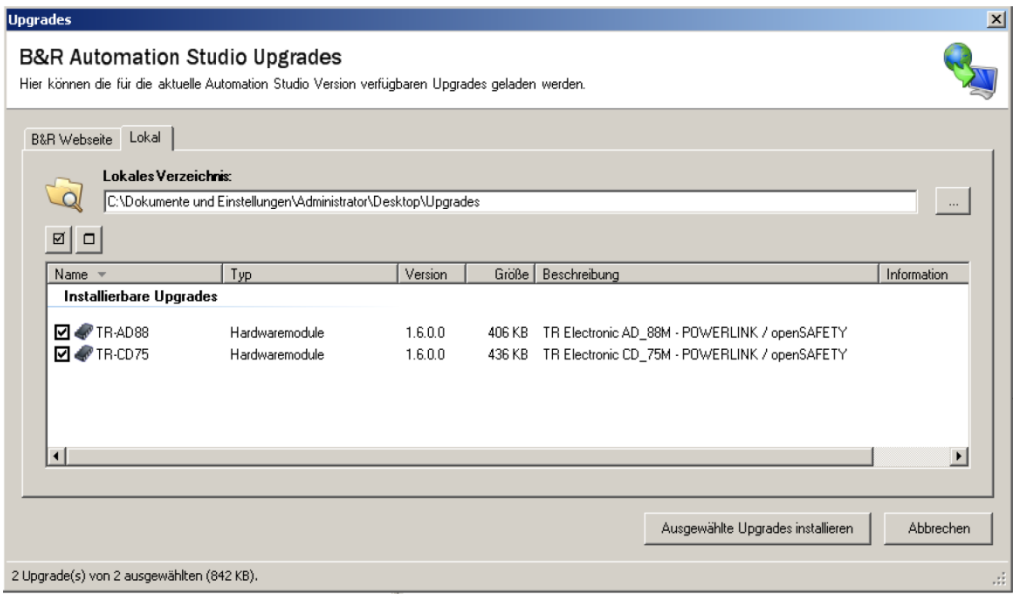


➢ `SafeDESIGNER`-Version unter der Konfiguration der `SafeLOGIC` auswählen:

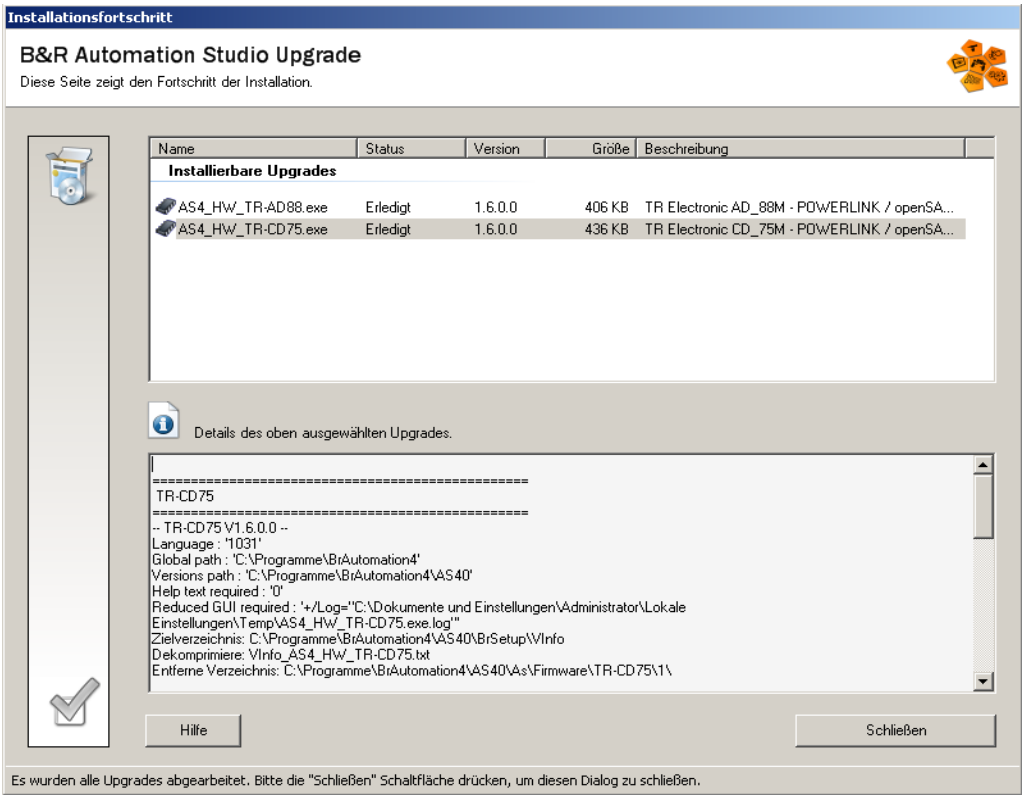


Bevor das Mess-System hinzugefügt werden kann, muss die Gerätebeschreibung installiert werden. Weil der Import einer xml-basierten Gerätebeschreibung für openSAFETY-Geräte zum Zeitpunkt der Erstellung dieses Dokumentes noch nicht existiert, wird die Gerätebeschreibung über ein AS-Update eingelesen. Dieses wird von TR-Electronic bereitgestellt. Das AS sollte nach der Installation neu gestartet werden.

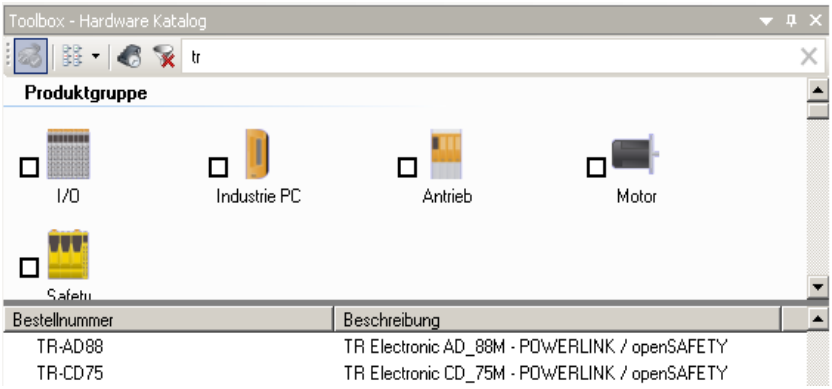
➢ AS-Update starten über Menü `Extras -> Upgrades...`:



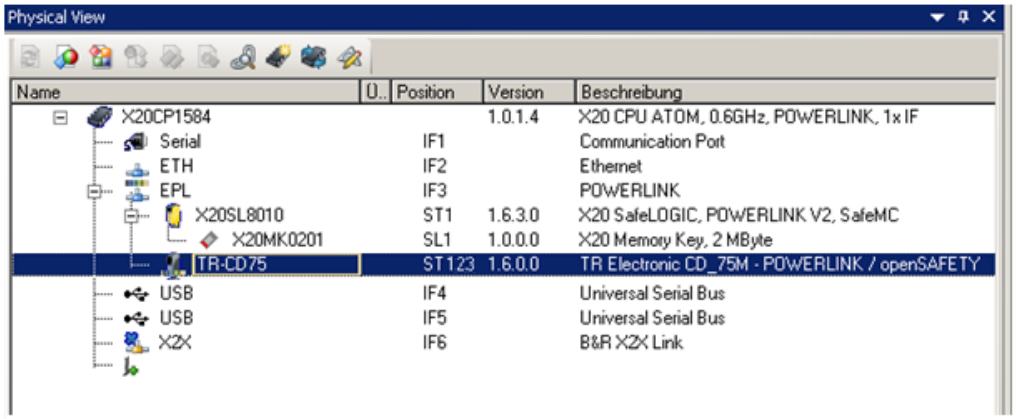
➢ Upgrades auswählen und Button `Ausgewählte Upgrades installieren` klicken:



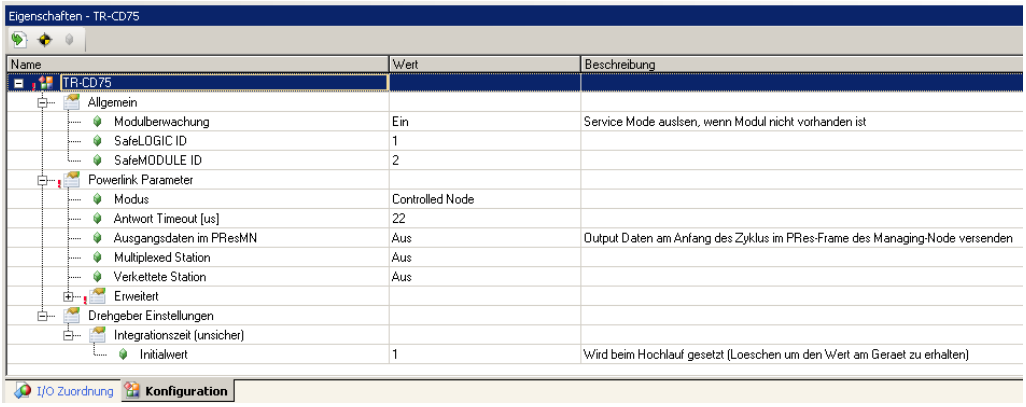
Anschließend erscheint die Auswahl im Hardwarekatalog:



Das Mess-System kann nach der Installation ebenfalls noch im `Physical View` hinzugefügt werden. Es muss die korrekte POWERLINK Node-ID vergeben werden. Hier im Beispiel die Node-ID 123 (0x7B), diese muss dann für das Mess-System eingestellt werden.

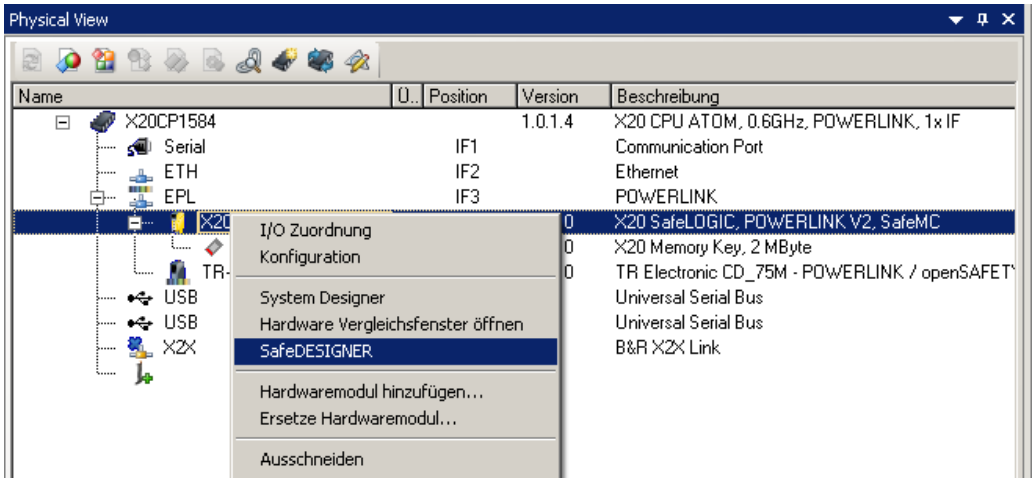


Die Mess-System - Konfiguration im `Automation Studio` wird zunächst nicht angepasst. Im vorliegenden Fall wurde für das Mess-System die `SafeMODULE ID` = 2 im openSAFETY-Netzwerk vergeben.



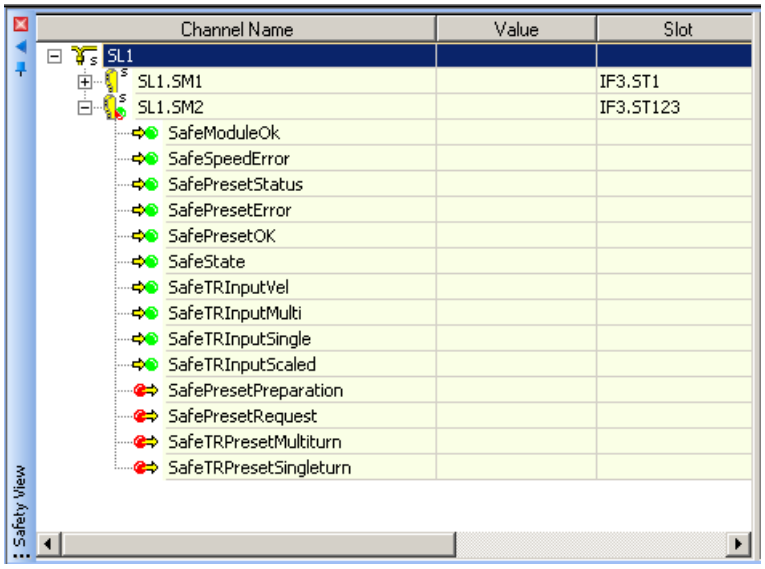
Nun kann das Projekt gespeichert und mit <Strg>+<F7> kompiliert werden. Bei der nachfolgenden Abfrage kann, falls eine Verbindung zur EPL-Steuerung besteht, das Projekt direkt auf die X20CP1584 übertragen werden. Dieser Punkt kann aber auch zu einem späteren Zeitpunkt nachgeholt werden. Wurde das Projekt übertragen, erhält der Anwender nach dem Download folgende Meldung:

- SafeDESIGNER durch einen Rechtsklick auf die SafeLOGIC starten:



- Kennwörter für das SafeDESIGNER-Projekt vergeben und bestätigen. Hier im Beispiel „abc123“.

Im `Safety View`, standardmäßig unten links, werden die openSAFETY-Teilnehmer aufgelistet. Es wird der SCM (SL1.SM1) und das Mess-System (SL1.SM2) angezeigt. Die „2“ entspricht der `SafeMODULE ID`. In der Spalte `Slot` wird die Node-ID des Mess-Systems angezeigt. Unter dem Mess-System sind alle im `SafeDESIGNER` verfügbaren Daten des Mess-Systems aufgelistet:

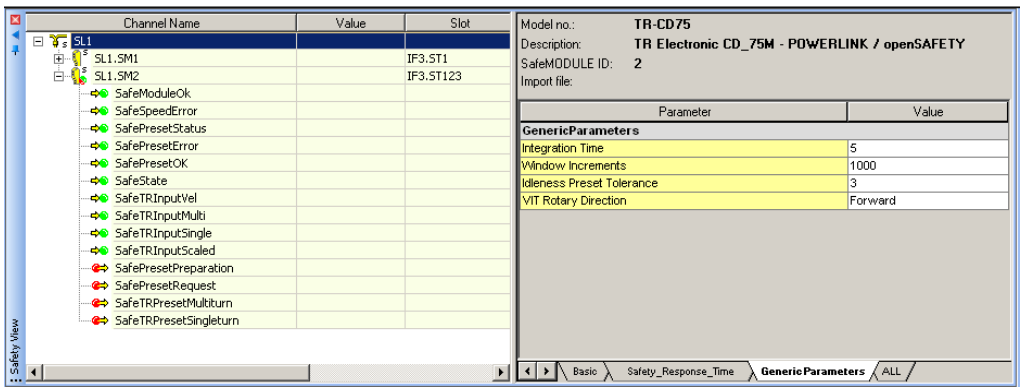


Die grünen Punkte markieren Eingangsdaten aus Sicht der Steuerung, die roten Punkte markieren Ausgangsdaten aus Sicht der Steuerung.

Im folgenden Kapitel werden die sicheren Parameter des Mess-Systems festgelegt.

4.1.4 Sichere Parametrierung

Die Parameter können über die Benutzeroberfläche des SafeDESIGNERS eingestellt werden. Die Mess-System - spezifischen Parameter befinden sich unter dem Reiter Generic Parameters:



In obigem Beispiel wurden folgende Werte vergeben:

Integration Time:	5
Window Increments	1000
Idleness Preset Tolerance	3
VIT Rotary Direction	Forward

Die einzelnen Werte sind wie folgt definiert:

Der Parameter Integration Time dient zur Berechnung der **sicheren Geschwindigkeit**, welche über openSAFETY ausgegeben wird. Über den Wertebereich von 1…10 (Zeitbasis 50 ms) kann die Zeit definiert werden, über welche die Geschwindigkeit gemessen wird:

- Hohe Integrationszeit = hohe Auflösung bei kleinen Drehzahlen
- Kleine Integrationszeit = schnelle Änderung für hohe Drehzahlen

Der Parameter Window Increments definiert die maximal zulässige Positionsabweichung in Inkrementen vom Master / Slave-Abtastsystem. Das zulässige Toleranzfenster ist abhängig von der maximalen Drehzahl und muss vom Anwender ermittelt werden. Hohe Drehzahlen erfordern ein hohes Toleranzfenster.
Je größer das Toleranzfenster, desto größer der Winkel, bis ein Fehler erkannt wird.
Der Standardwert ist 1000.

Der Parameter Idleness Preset Tolerance definiert die maximale Geschwindigkeit zur Durchführung eines Presetvorgangs. Der Wertebereich 1..5 ist direkt abhängig von der Integrationszeit (safe). Der Anwender muss diesen Wert in Abhängigkeit zur Applikation einstellen.

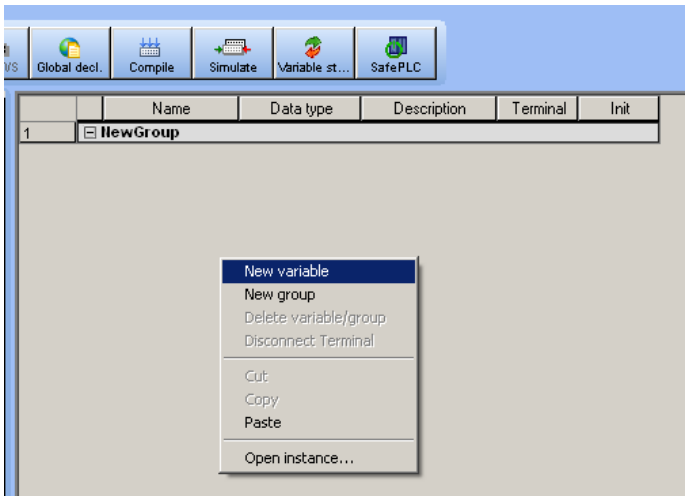
Der Parameter VIT Rotary Direction gibt an, ob der Istwert im Vorlauf ansteigt (forward) oder kleiner wird (backward).

Die sicheren Parameter werden beim Hochlauf automatisch vom SCM an das Mess-System (SN) übertragen.

4.1.5 Erstellen des Sicherheitsprogramms

Eine sichere Applikation, die auf dem SCM läuft, wird im Editor des SafeDESIGNERS erstellt. Folgende Schritte zeigen den Aufbau eines Dummy-Programms, welches den Singleturn-Wert und den Multiturn-Wert vergleicht und in Abhängigkeit dazu eine globale Variable setzt. Die Variable wird aber nur gesetzt, wenn das Mess-System einen gültigen Istwert ausgibt. Dafür müssen die Zustandsvariablen SafeModuleOk und SafeState gesetzt sein.

- Benötigte Variablen deklarieren. Mittels Toolleiste bzw. über <Strg>+<G> die Oberfläche Global Declarations im SafeDESIGNER öffnen. Mit einem Rechtsklick über das Kontextmenü -> New variable die neue Variable erstellen:



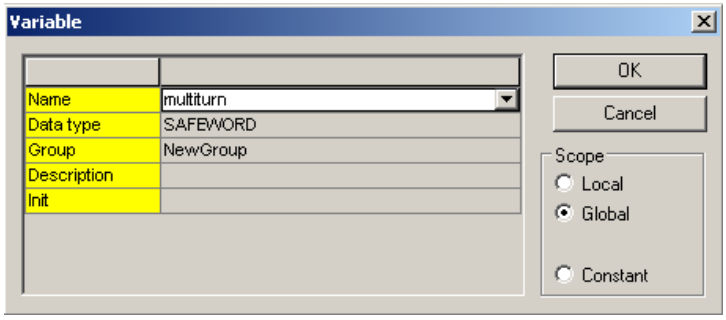
- Es werden insgesamt 5 Variablen mit den entsprechenden Datentypen erstellt:

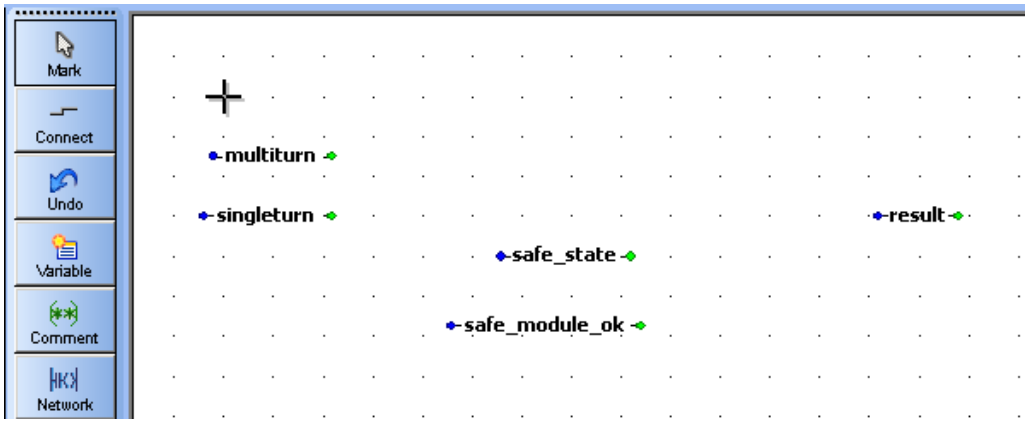
	Name	Data type	Description	Terminal	Init
1	NewGroup				
2	singleturn	SAFEWORD			
3	multiturn	SAFEWORD			
4	safe_module_ok	SAFEBOOL			
5	safe_state	SAFEBOOL			
6	result	SAFEBOOL			

- Erstellte Variablen mit den entsprechenden Werten vom Mess-System verknüpfen. Dafür wird mit der Maus die entsprechende Variable aus dem Safety View auf das Terminal-Feld der entsprechenden Variable gezogen. Es werden die vier Mess-System - Variablen SafeModuleOk, SafeState, SafeTRInputSingle und SafeTRInputMulti verknüpft:

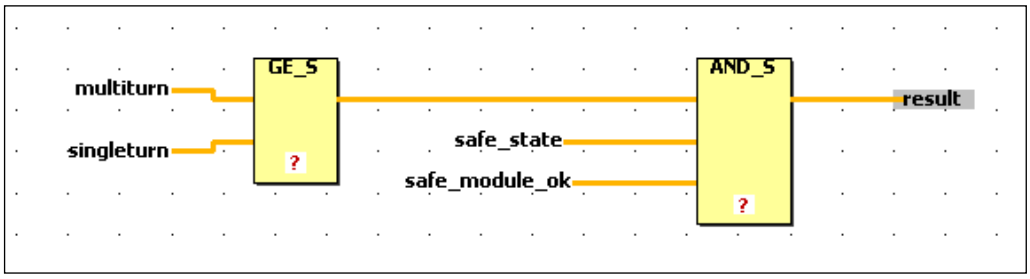
	Name	Data type	Description	Terminal	Init
1	NewGroup				
2	singleturn	SAFEWORD		SL1.SM2.SafeTRInputSingle	
3	multiturn	SAFEWORD		SL1.SM2.SafeTRInputMulti	
4	safe_module_ok	SAFEBOOL		SL1.SM2.SafeModuleOk	
5	safe_state	SAFEBOOL		SL1.SM2.SafeState	
6	result	SAFEBOOL			

- Wieder zum Reiter `Code: Main` überwechseln. Hier wird nun die eigentliche Applikation grafisch programmiert. Dazu können mit dem Button `Variable` am linken Rand des Worksheets die benötigten Variablen ausgewählt und auf die Oberfläche kopiert werden:





- Variablen mit Hilfe der FUs und FBs miteinander verknüpfen:



Die Erstellung der Beispielapplikation ist damit abgeschlossen. Die Variable `result` wird `TRUE`, sobald der Multiturn-Wert des Mess-Systems größer oder gleich (greater or equal) als der Singleturn-Wert ist und die Variablen `safe_state` und `safe_module_ok` gesetzt sind.

4.1.6 Generieren des Sicherheitsprogramms

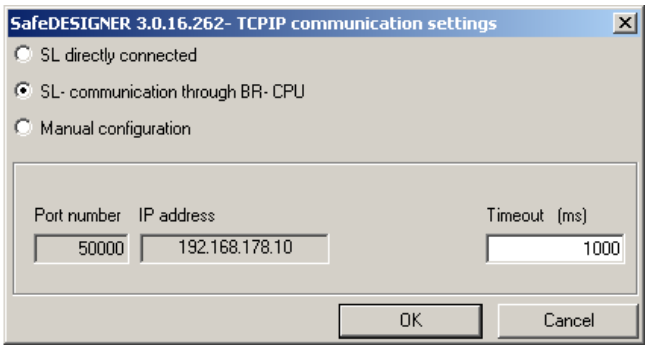
Zur Erstellung des Sicherheitsprogramms muss die gerade erstellte Beispielapplikation gespeichert und übersetzt werden. Dafür wird der `Compile`-Button im `SafeDESIGNER` oder die Taste <F9> gedrückt.

Da die `result`-Variable nur geschrieben wird und nicht weiter verwendet wird, werden zwei Warnungen erzeugt. Diese können an dieser Stelle ignoriert werden.

4.1.7 Sicherheitsprogramm laden

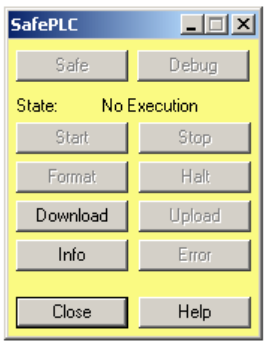
Um die folgenden Schritte durchzuführen, muss zunächst das Projekt aus dem `Automation Studio` in die `X20CP1584` eingelesen werden (=POWERLINK-Projekt).

Nachdem das Sicherheitsprogramm generiert worden ist, kann es in den SCM geladen werden. Dafür muss eine Verbindung zum SCM bestehen. Unter dem Menüpunkt `Online` im `SafeDESIGNER` kann die Verbindungsart eingestellt werden. Wurde im Vorfeld eine Ethernet-Kommunikation zur POWERLINK-CPU eingerichtet und die POWERLINK-CPU ist über Ethernet zu erreichen, empfiehlt sich die Einstellung `SL - communication through BR-CPU`:

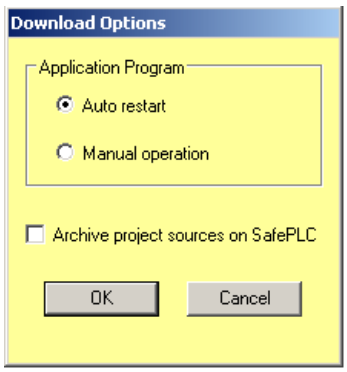


Der `SafeDESIGNER` verwendet die Portnummer sowie die aktuelle Onlinekonfiguration aus dem `Automation Studio` als Basis für die Onlinekommunikation.

Für das Laden der Software muss über den Button `SafePLC` oder über <Strg>+<F10> eine Verbindung zum SCM aufgebaut werden. Wurde eine Verbindung hergestellt, muss zunächst ein Paßwort vergeben oder eingegeben werden. Anschließend erscheint ein kleines Bedienfenster, mit welchem der Download durchgeführt werden kann:



Um einen Download durchzuführen, muss der `Download`-Button aktiviert sein. Dafür muss unter Umständen in den Debug-Mode gewechselt werden. Dann wird, falls vorhanden, die aktuelle Applikation mit `Stop` auf dem SCM angehalten. Anschließend wird der `Download`-Button aktiviert und der Download der eigenen Applikation kann durchgeführt werden. Unter `Download Options` wird `Auto restart` ausgewählt, damit die Applikation automatisch anläuft:



Nach dem Download startet die `SafeLogic` neu. Nun müssen Firmware und neue Teilnehmer an der SL quittiert werden. Für die einzelnen Quittierungen wird jeweils ein Blinkmuster oder eine entsprechende LED gesetzt.

1. Die SL läuft neu hoch. Dabei blinkt die R/E-LED und die drei darunterliegenden LEDs (ENTER, MXCHG, FW_ACKN) blinken nacheinander von oben nach unten.

2. Am Ende des Hochlaufs muss zunächst die neue Firmware quittiert werden .Dies wird durch ein Blinken der R/E-LED angezeigt und durch ein Blinkmuster der FW-ACKN-LED von 80% an und 20% aus -> Die Quittierung wird durchgeführt, indem der Codierschalter auf FW_ACKN („12 Uhr“) gestellt wird und anschließend die Taste <ENTER> gedrückt wird. Die SL startet wiederum neu.

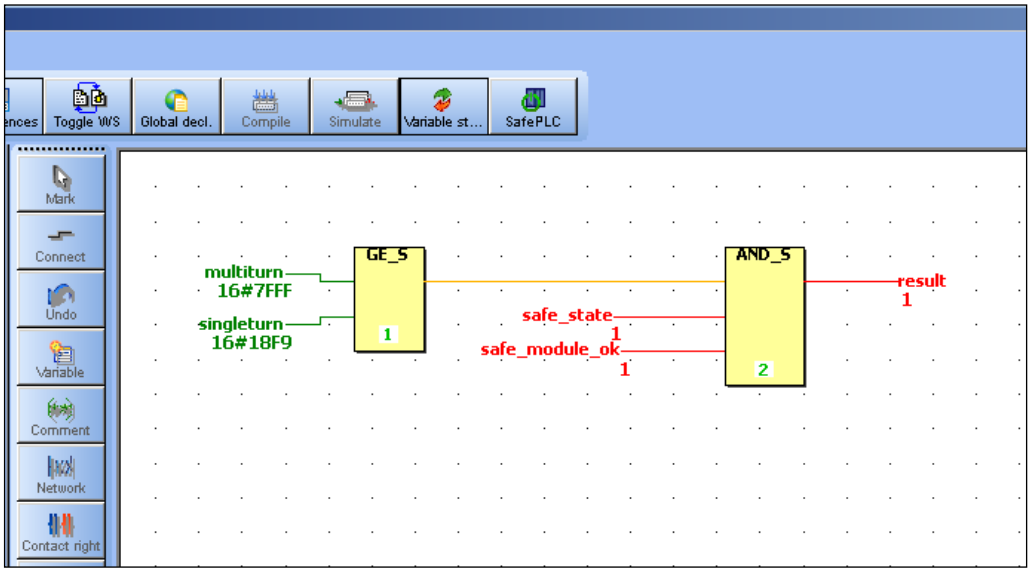
3. Am Ende des Hochlaufs müssen die neuen Teilnehmer quittiert werden. Dies wird durch ein Blinken der R/E-LED angezeigt. Zusätzlich wird über die MXCHG-LED die Anzahl zu quittierender Teilnehmer ausgegeben: 1xBlinken=1 neuer Teilnehmer; 2xBlinken=2 neue Teilnehmer; ... -> für die Quittierung wird der Codierschalter auf die Anzahl zu quittierender Teilnehmer gestellt (für >4 wird die Stellung „n“ gewählt) und anschließend die Taste <ENTER> gedrückt. Die SL fährt anschließend den neuen Teilnehmer hoch.

Bitte für diesen Vorgang zusätzlich das Handbuch der SL heranziehen, da sich die Blinkmuster je nach Vorgehensweise und vorinstallierter Software leicht unterscheiden können.

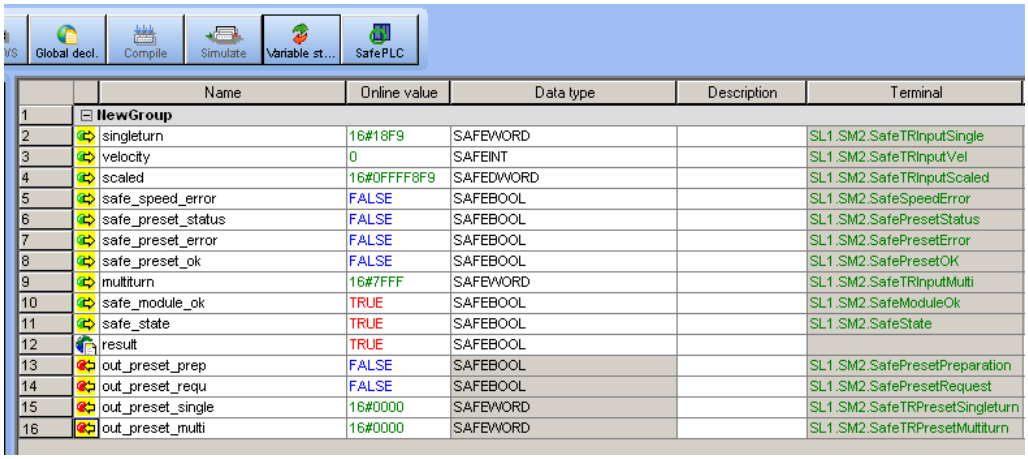
4.1.8 Sicherheitsprogramm testen

Nach Erstellung des Sicherheitsprogramms muss ein vollständiger Funktionstest entsprechend der Automatisierungsaufgabe durchgeführt werden.

Im SafeDESIGNER kann mit dem Button Variable status oder durch die Taste <F10> eine Online-Verbindung aufgebaut werden, bei welcher die aktuellen Variablenwerte im SafeDESIGNER angezeigt werden.



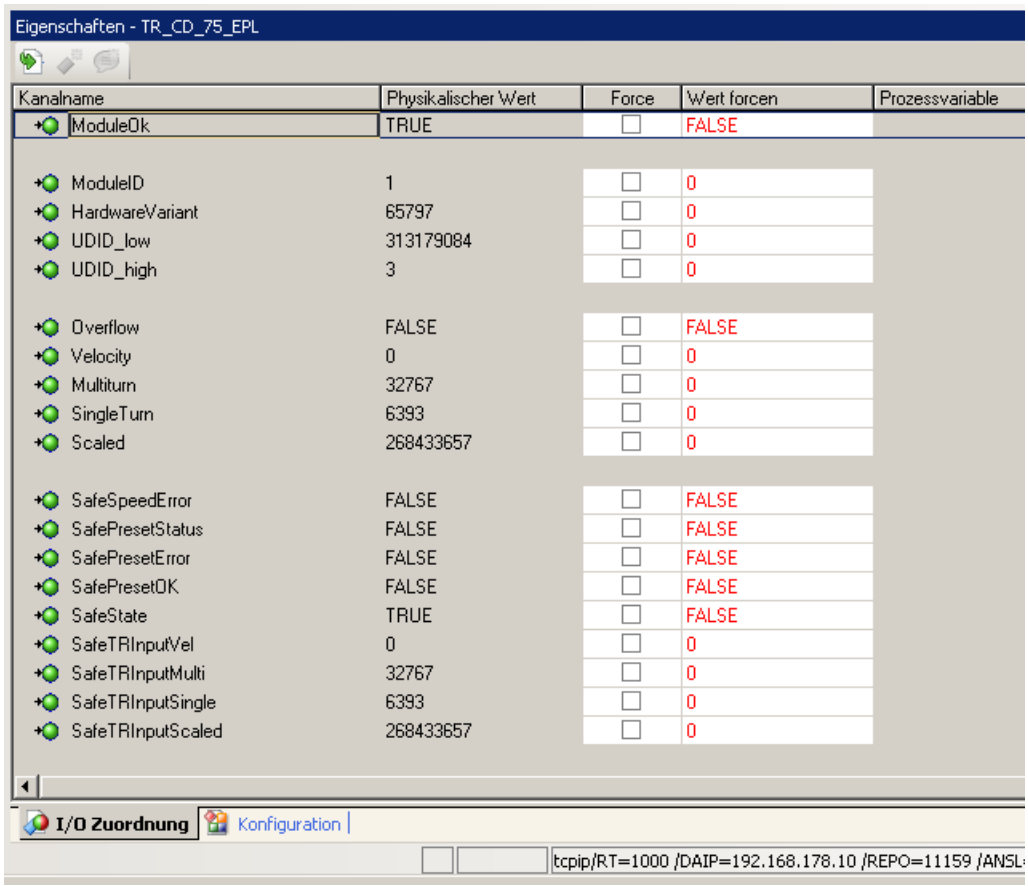
Auch zu den globalen Variablen werden alle Werte angezeigt. Im folgenden Bild wurden noch alle restlichen Mess-System - Werte zu Variablen verbunden:



	Name	Online value	Data type	Description	Terminal
1	NewGroup				
2	singleturn	16#18F9	SAFEWORD		SL1.SM2.SafeTRInputSingle
3	velocity	0	SAFEINT		SL1.SM2.SafeTRInputVel
4	scaled	16#0FFFF8F9	SAFEDWORD		SL1.SM2.SafeTRInputScaled
5	safe_speed_error	FALSE	SAFEBOOL		SL1.SM2.SafeSpeedError
6	safe_preset_status	FALSE	SAFEBOOL		SL1.SM2.SafePresetStatus
7	safe_preset_error	FALSE	SAFEBOOL		SL1.SM2.SafePresetError
8	safe_preset_ok	FALSE	SAFEBOOL		SL1.SM2.SafePresetOK
9	multiturn	16#7FFF	SAFEWORD		SL1.SM2.SafeTRInputMulti
10	safe_module_ok	TRUE	SAFEBOOL		SL1.SM2.SafeModuleOk
11	safe_state	TRUE	SAFEBOOL		SL1.SM2.SafeState
12	result	TRUE	SAFEBOOL		
13	out_preset_prep	FALSE	SAFEBOOL		SL1.SM2.SafePresetPreparation
14	out_preset_requ	FALSE	SAFEBOOL		SL1.SM2.SafePresetRequest
15	out_preset_single	16#0000	SAFEWORD		SL1.SM2.SafeTRPresetSingleturn
16	out_preset_multi	16#0000	SAFEWORD		SL1.SM2.SafeTRPresetMultiturn

Das Mess-System gibt erst aktuelle Istwerte aus, wenn der Hochlauf über openSAFETY und die sichere Parametrierung erfolgt sind. Bis dahin sind alle Safe-Variablen auf dem Wert 0 und alle Bits der „grauen Daten“ auf 1 gesetzt.

Alle Istwerte der Variablen sind auch im Online-Mode des `Automation Studios` sichtbar, sie werden unter der I/O Zuordnung des Mess-Systems angezeigt:



In obiger Abbildung beginnen die Variablen-Namen der zweikanaligen Daten mit dem Wort `Safe` und können auch im „grauen“ Bereich genutzt werden.

Die „grauen“ einkanaligen Werte werden darüber angegeben:

- `Scaled`
- `SingleTurn`
- `Multiturn`
- ...

4.2 XDD/OSDD Gerätebeschreibung (ab AS V4.5)

Dieses Kapitel beschreibt die Vorgehensweise bei der Erstellung eines Beispiel-Sicherheitsprogramms mit Verwendung der B&R Projektierungssoftware Automation Studio (V4.5.2.102) und dem Optionspaket SafeDESIGNER (V4.3.3.7).

Das Sicherheitsprogramm wird im SafeDESIGNER erstellt, welcher einen Code-Editor zur Entwicklung des Programms für die Sicherheitssteuerung enthält. Dies erfolgt mit Hilfe der graphischen Programmiersprachen FBS und KOP.

Die Ausführung des Sicherheitsprogramms erfolgt auf einer X20 SafeLOGIC (X20SL8100). Diese nimmt als normaler POWERLINK-CN an der Feldbuskommunikation teil und ist selbst wiederrum der openSAFETY SCM.

Als MN bei POWERLINK kommt eine X20 CP1584 zum Einsatz. Sie enthält eine POWERLINK-Anwendung, die ebenfalls die Mess-System - Informationen (u.a. die Istwerte „grauer Kanal“) auswerten kann.

4.2.1 Zugriffschutz

Der Zugang zum SafeDESIGNER-Projekt ist durch eine Paßwortabfrage gesichert. Es existiert jeweils ein Paßwort für die Entwicklung und für die Inbetriebnahme. Der Zugang zur SL8100, z.B. zur Programmierung, ist ebenfalls durch ein Paßwort geschützt.

Das Paßwort heißt: abc123

4.2.2 Voraussetzungen

⚠WARNUNG ***Gefahr der Außerkraftsetzung der fehlersicheren Funktion durch unsachgemäße Projektierung des Sicherheitsprogramms!***

- Die Erstellung des Sicherheitsprogramms darf nur in Verbindung mit der von B&R zur Software bzw. Hardware mitgelieferten System-dokumentation erfolgen.
- Nachfolgende Beschreibungen beziehen sich auf den reinen Ablauf, ohne dabei alle Hinweise aus den B&R-Handbüchern mit zu berücksichtigen.
  Die in den B&R-Handbüchern gegebenen Informationen, Hinweise, insbesondere die Sicherheitshinweise und Warnungen, sind daher zwingend zu beachten und einzuhalten.
- Die aufgezeigte Projektierung ist als Beispiel aufzufassen. Der Anwender ist daher verpflichtet, die Verwendbarkeit der Projektierung für seine Applikation zu überprüfen und anzupassen. Dazu gehören auch die Auswahl der geeigneten sicherheitsgerichteten Hard-warekomponenten, sowie die notwendigen Softwarevoraussetzungen.

Für das Konfigurationsbeispiel benutzte Software-Komponenten:

- `Automation Studio` V4.5.2.102
- `SafeDESIGNER` V4.3.3.7

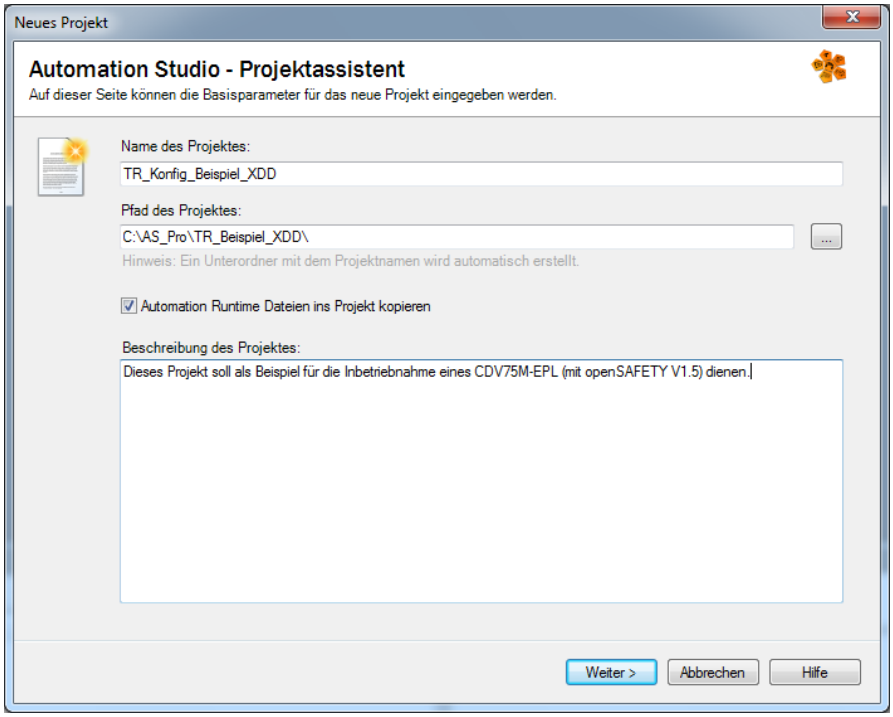
Die Verwendung nachfolgender Versionen ist ebenfalls möglich.

Für das Konfigurationsbeispiel benutzte Hardware-Komponenten:

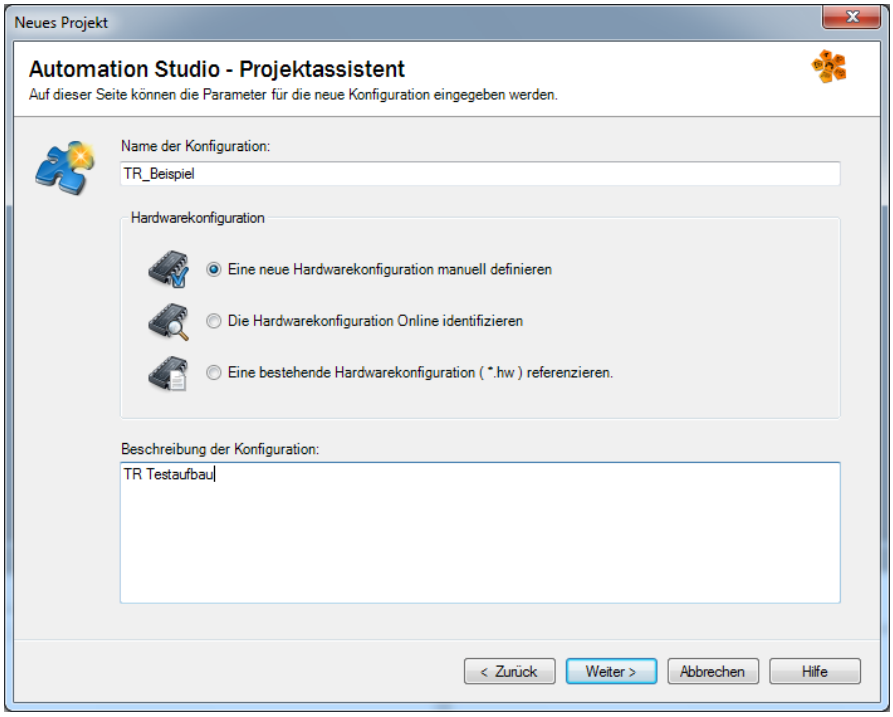
- POWERLINK-MN: `X20CP1584` mit `X20IF1082-2`
- openSAFETY SCM: `X20SL8100`

4.2.3 Hardware-Konfiguration

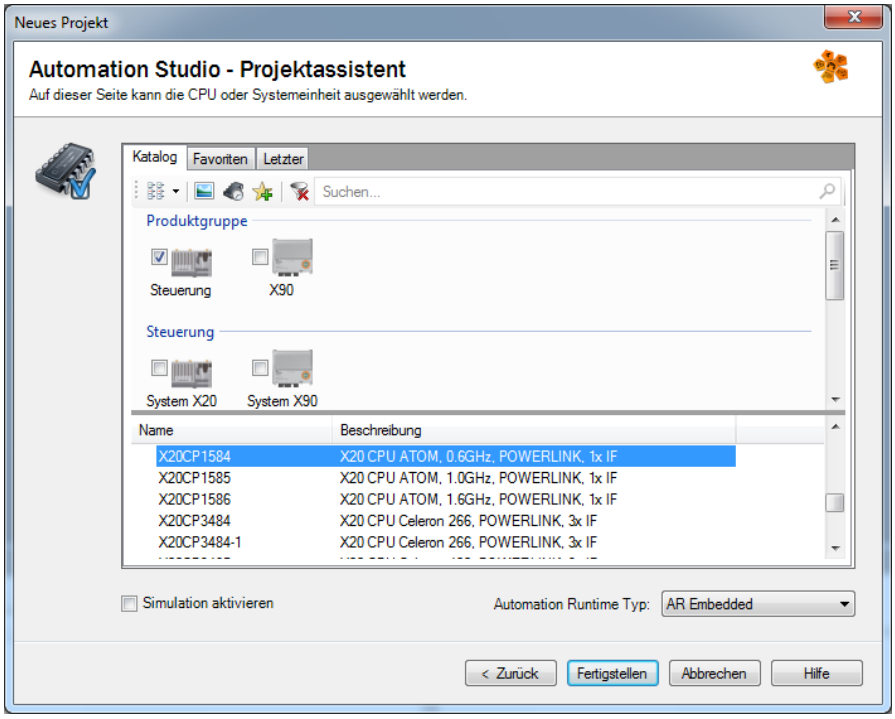
➢ `Automation Studio` starten und ein neues Projekt anlegen.



➢ Es wird eine neue Hardwarekonfiguration erstellt:



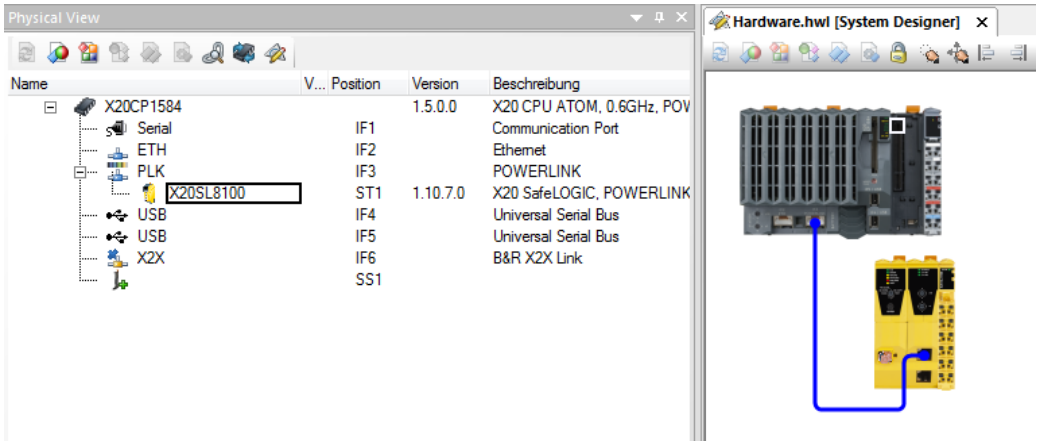
➢ Abschließend korrekte CPU auswählen: `X20CP1584`



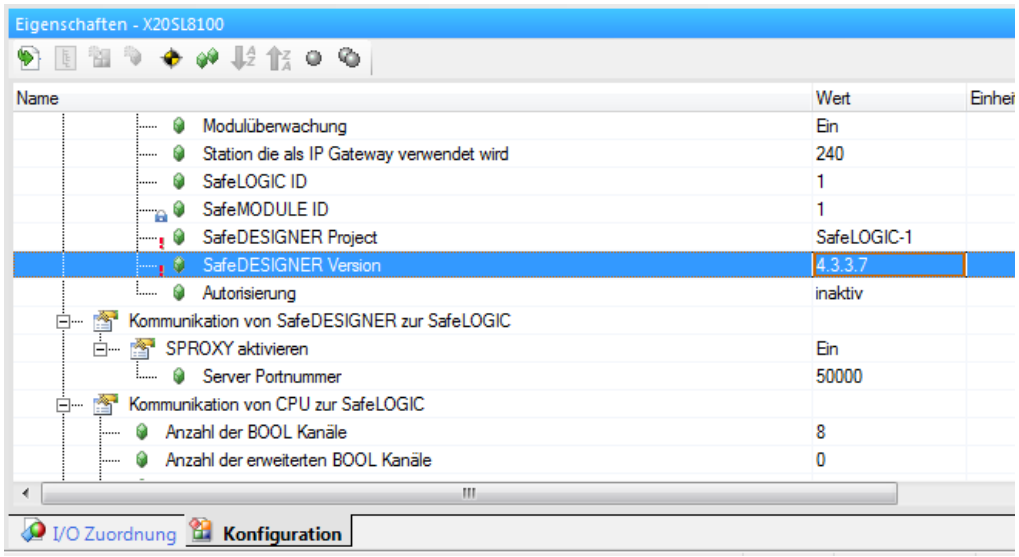
➢ Unter dem Menüpunkt `-> Projekt -> Runtime Versionen ändern...` das Safety Release 1.10 auswählen. Als Automation Runtime wird im vorliegenden Beispiel die Version D4.52 verwendet.

Für das Mess-System wird das Safety Release 1.10 oder höher empfohlen.

➢ SCM zum EPL-Netz hinzufügen, indem man das entsprechende Gerät aus dem Hardware Katalog auf die POWERLINK-Schnittstelle im `Physical View` zieht:



➢ `SafeDESIGNER`-Version unter der Konfiguration der `SafeLOGIC` auswählen:

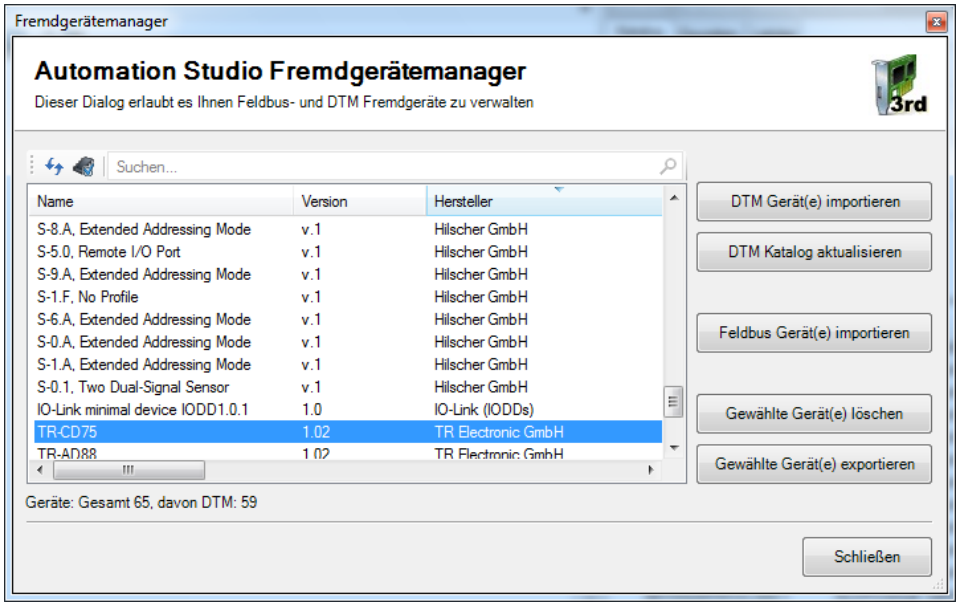


Bevor das Mess-System hinzugefügt werden kann, muss die Gerätebeschreibung installiert werden. Der Import einer xml-basierten Gerätebeschreibung für openSAFETY-Geräte ist hierfür notwendig. Die XDD-/OSDD-Gerätebeschreibungen werden von TR-Electronic bereitgestellt. Das AS sollte nach der Installation neu gestartet werden.

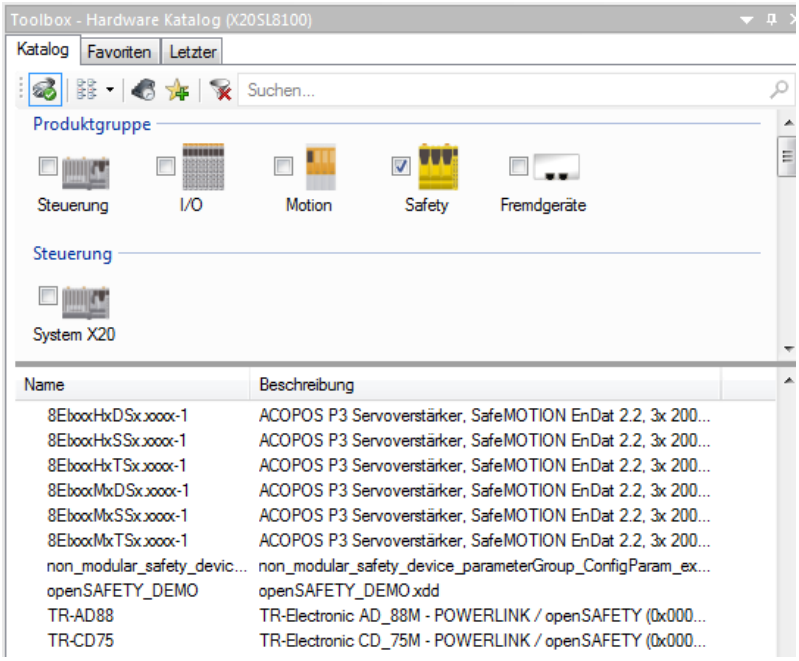
➢ Über Menü `Extras -> Fremdgeräte verwalten` den Button `Feldbus Gerät(e) importieren` drücken.

➢ Zuerst die OSDD-Datei auswählen, dann die XDD-Datei

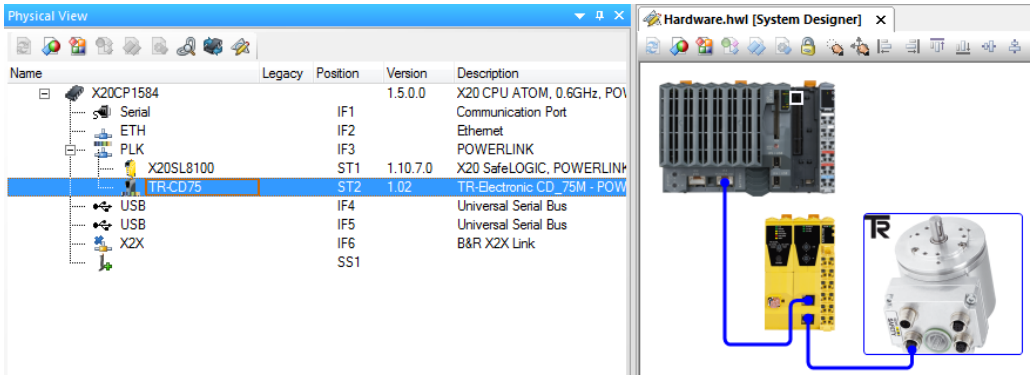
➢ Die installierten Geräte erscheinen nach der Installation in der Liste:



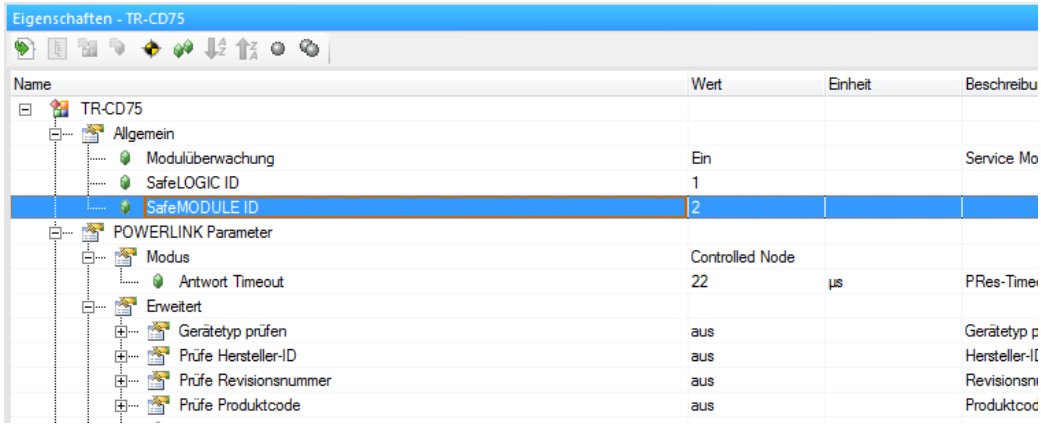
Anschließend erscheint die Auswahl im Hardwarekatalog:



Das Mess-System kann nach der Installation ebenfalls noch im `Physical View` hinzugefügt werden. Es muss die korrekte POWERLINK Node-ID vergeben werden. Hier im Beispiel die Node-ID 123 (0x7B), diese muss dann für das Mess-System eingestellt werden.

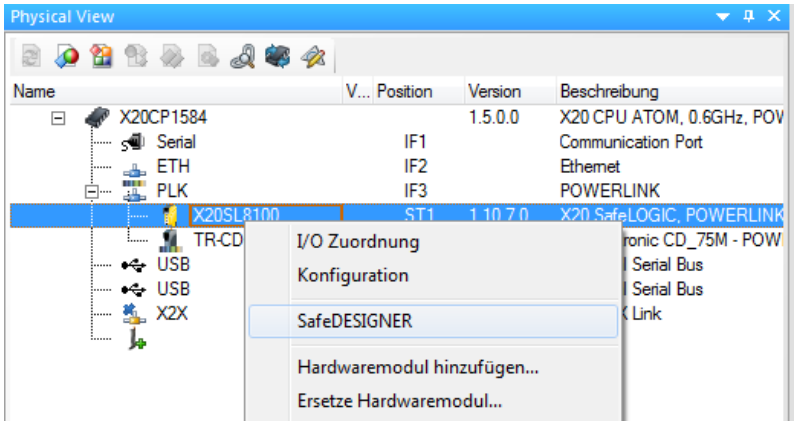


Die Mess-System - Konfiguration im `Automation Studio` wird zunächst nicht angepasst. Im vorliegenden Fall wurde für das Mess-System die `SafeMODULE ID` = 2 im openSAFETY-Netzwerk vergeben.



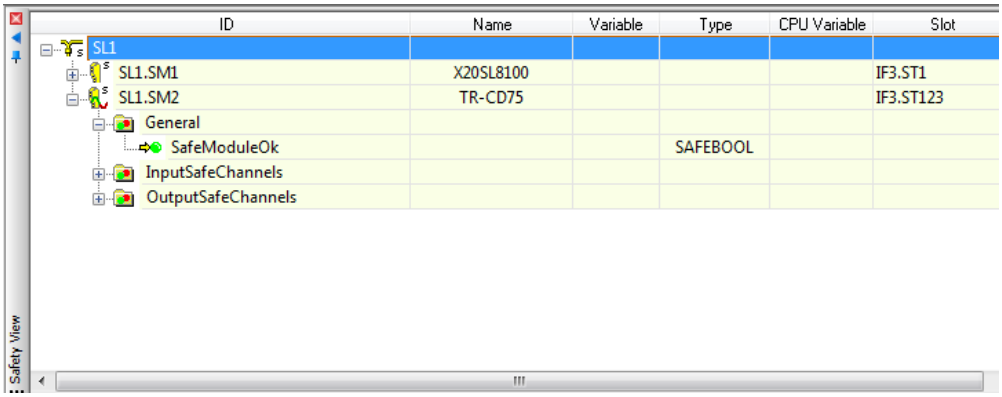
Nun kann das Projekt gespeichert und mit <Strg>+<F7> kompiliert werden. Bei der nachfolgenden Abfrage kann, falls eine Verbindung zur EPL-Steuerung besteht, das Projekt direkt auf die X20CP1584 übertragen werden. Dieser Punkt kann aber auch zu einem späteren Zeitpunkt nachgeholt werden.

- SafeDESIGNER durch einen Rechtsklick auf die SafeLOGIC starten:



- Kennwörter für das SafeDESIGNER-Projekt vergeben und bestätigen. Hier im Beispiel „abc123“.

Im `Safety View`, standardmäßig unten links, werden die openSAFETY-Teilnehmer aufgelistet. Es wird der SCM (SL1.SM1) und das Mess-System (SL1.SM2) angezeigt. Die „2“ entspricht der `SafeMODULE ID`. In der Spalte `Slot` wird die Node-ID des Mess-Systems angezeigt. Unter dem Mess-System sind alle im `SafeDESIGNER` verfügbaren Daten des Mess-Systems aufgelistet:



Diese sind:

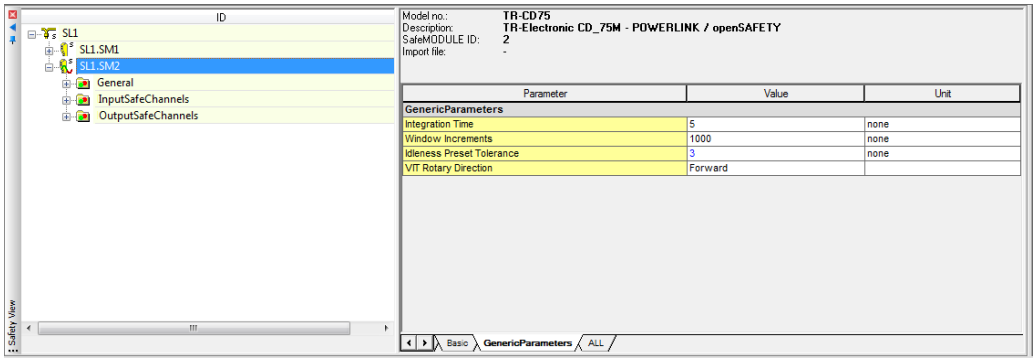
- InputSafeChannels:
  - SafeSpeedError
  - SafePresetStatus
  - SafePresetError
  - SafePresetOK
  - SafeState
  - SafeTRInputVel
  - SafeTRInputMulti
  - SafeTRInputSingle
  - SafeTRInputScaled
- OutputSafeChannels
  - SafePresetPreparation
  - SafePresetRequest
  - SafeTRPresetMultiturn
  - SafeTRPresetSingleturn

Die grünen Punkte markieren Eingangsdaten aus Sicht der Steuerung, die roten Punkte markieren Ausgangsdaten aus Sicht der Steuerung.

Im folgenden Kapitel werden die sicheren Parameter des Mess-Systems festgelegt.

4.2.4 Sichere Parametrierung

Die Parameter können über die Benutzeroberfläche des SafeDESIGNERS eingestellt werden. Die Mess-System - spezifischen Parameter befinden sich unter dem Reiter Generic Parameters:



In obigem Beispiel wurden folgende Werte vergeben:

Integration Time:	5
Window Increments	1000
Idleness Preset Tolerance	3
VIT Rotary Direction	Forward

Die einzelnen Werte sind wie folgt definiert:

Der Parameter Integration Time dient zur Berechnung der **sicheren Geschwindigkeit**, welche über openSAFETY ausgegeben wird. Über den Wertebereich von 1…10 (Zeitbasis 50 ms) kann die Zeit definiert werden, über welche die Geschwindigkeit gemessen wird:

- Hohe Integrationszeit = hohe Auflösung bei kleinen Drehzahlen
- Kleine Integrationszeit = schnelle Änderung für hohe Drehzahlen

Der Parameter Window Increments definiert die maximal zulässige Positionsabweichung in Inkrementen vom Master / Slave-Abtastsystem. Das zulässige Toleranzfenster ist abhängig von der maximalen Drehzahl und muss vom Anwender ermittelt werden. Hohe Drehzahlen erfordern ein hohes Toleranzfenster.
Je größer das Toleranzfenster, desto größer der Winkel, bis ein Fehler erkannt wird.
Der Standardwert ist 1000.

Der Parameter Idleness Preset Tolerance definiert die maximale Geschwindigkeit zur Durchführung eines Presetvorgangs. Der Wertebereich 1..5 ist direkt abhängig von der Integrationszeit (safe). Der Anwender muss diesen Wert in Abhängigkeit zur Applikation einstellen.

Der Parameter VIT Rotary Direction gibt an, ob der Istwert im Vorlauf ansteigt (forward) oder kleiner wird (backward).

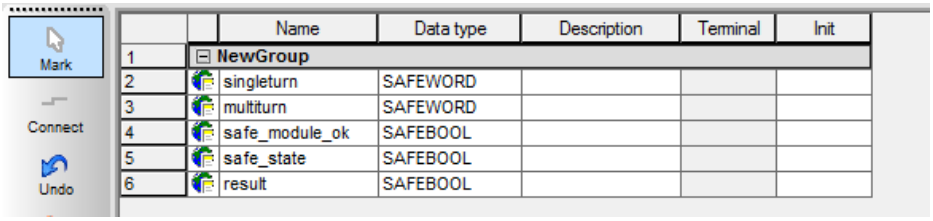
Die sicheren Parameter werden beim Hochlauf automatisch vom SCM an das Mess-System (SN) übertragen.

4.2.5 Erstellen des Sicherheitsprogramms

Eine sichere Applikation, die auf dem SCM läuft, wird im Editor des SafeDESIGNERS erstellt. Folgende Schritte zeigen den Aufbau eines Dummy-Programms, welches den Singleturn-Wert und den Multiturn-Wert vergleicht und in Abhängigkeit dazu eine globale Variable setzt. Die Variable wird aber nur gesetzt, wenn das Mess-System einen gültigen Istwert ausgibt. Dafür müssen die Zustandsvariablen SafeModuleOk und SafeState gesetzt sein.

- Benötigte Variablen deklarieren. Mittels Toolleiste bzw. über <Strg>+<G> die Oberfläche Global Declarations im SafeDESIGNER öffnen. Mit einem Rechtsklick über das Kontextmenü -> New variable die neuen Variablen erstellen.

- Es werden insgesamt 5 Variablen mit den entsprechenden Datentypen erstellt:

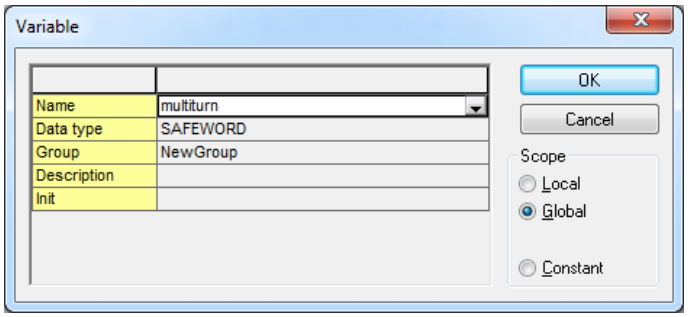


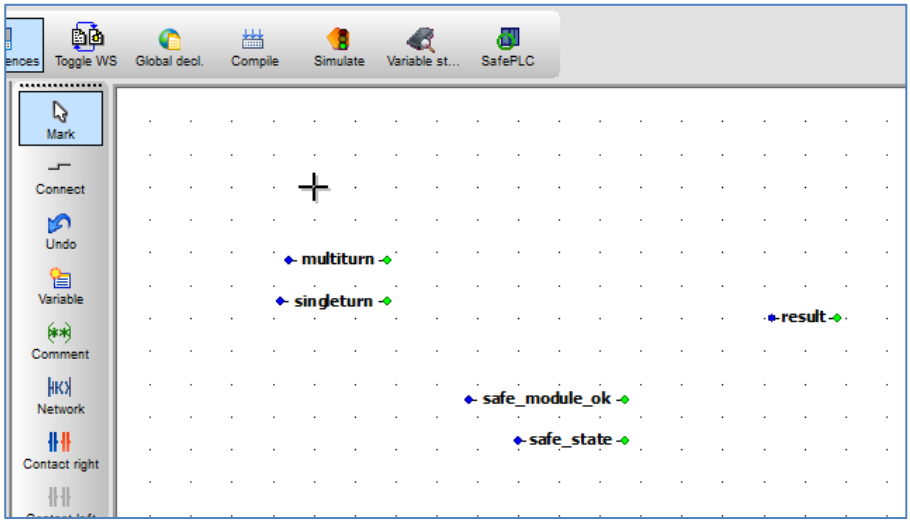
		Name	Data type	Description	Terminal	Init
1		NewGroup				
2		singleturn	SAFEWORD			
3		multiturn	SAFEWORD			
4		safe_module_ok	SAFEBOOL			
5		safe_state	SAFEBOOL			
6		result	SAFEBOOL			

- Erstellte Variablen mit den entsprechenden Werten vom Mess-System verknüpfen. Dafür wird mit der Maus die entsprechende Variable aus dem Safety View auf das Terminal-Feld der entsprechenden Variable gezogen. Es werden die vier Mess-System - Variablen SafeModuleOk, SafeState, SafeTRInputSingle und SafeTRInputMulti verknüpft:

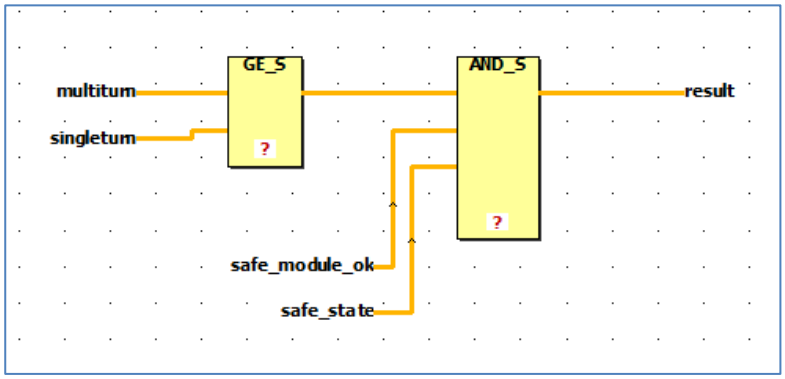
		Name	Data type	Description	Terminal	Init
1		NewGroup				
2		singleturn	SAFEWORD		SL1.SM2.SafeTRInputSingle	
3		multiturn	SAFEWORD		SL1.SM2.SafeTRInputMulti	
4		safe_module_ok	SAFEBOOL		SL1.SM2.SafeModuleOk	
5		safe_state	SAFEBOOL		SL1.SM2.SafeState	
6		result	SAFEBOOL			

➢ Wieder zum Reiter `Code: Main` überwechseln. Hier wird nun die eigentliche Applikation grafisch programmiert. Dazu können mit dem Button `Variable` am linken Rand des Worksheets die benötigten Variablen ausgewählt und auf die Oberfläche kopiert werden:





➢ Variablen mit Hilfe der FUs und FBs miteinander verknüpfen:



Die Erstellung der Beispielapplikation ist damit abgeschlossen. Die Variable `result` wird `TRUE`, sobald der Multiturn-Wert des Mess-Systems größer oder gleich (greater or equal) als der Singleturn-Wert ist und die Variablen `safe_state` und `safe_module_ok` gesetzt sind.

4.2.6 Generieren des Sicherheitsprogramms

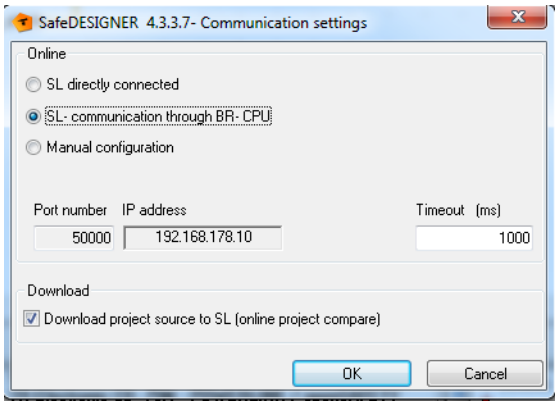
Zur Erstellung des Sicherheitsprogramms muss die gerade erstellte Beispielapplikation gespeichert und übersetzt werden. Dafür wird der `Compile`-Button im `SafeDESIGNER` oder die Taste <F9> gedrückt.

Da die `result`-Variable nur geschrieben wird und nicht weiter verwendet wird, werden zwei Warnungen erzeugt. Diese können an dieser Stelle ignoriert werden.

4.2.7 Sicherheitsprogramm laden

Um die folgenden Schritte durchzuführen, muss zunächst das Projekt aus dem `Automation Studio` in die `X20CP1584` programmiert werden (=POWERLINK-Projekt).

Nachdem das Sicherheitsprogramm generiert worden ist, kann es in den SCM geladen werden. Dafür muss eine Verbindung zum SCM bestehen. Unter dem Menüpunkt `Online` im `SafeDESIGNER` kann die Verbindungsart eingestellt werden. Wurde im Vorfeld eine Ethernet-Kommunikation zur POWERLINK-CPU eingerichtet und die POWERLINK-CPU ist über Ethernet zu erreichen, empfiehlt sich die Einstellung `SL - communication through BR-CPU`:

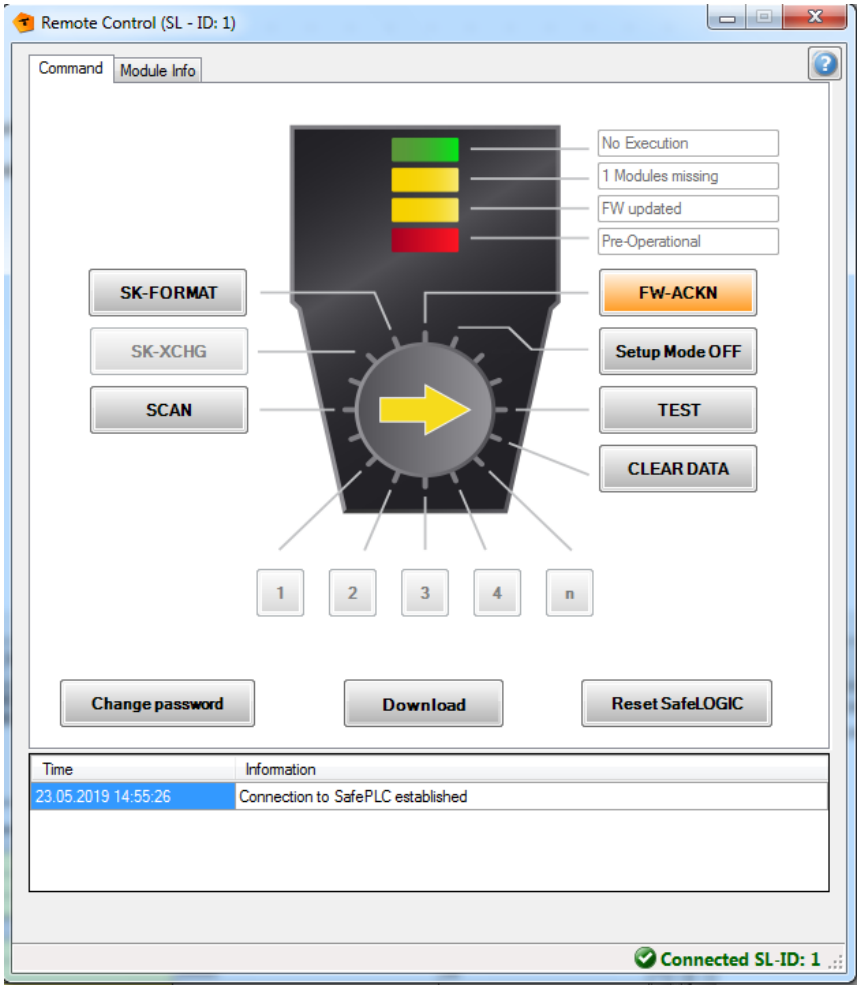


Der `SafeDESIGNER` verwendet die Portnummer sowie die aktuelle Onlinekonfiguration aus dem `Automation Studio` als Basis für die Onlinekommunikation.

Für das Laden der Software muss über den Button `SafePLC` oder über <Strg>+<F10> eine Verbindung zum SCM aufgebaut werden. Wurde eine Verbindung hergestellt, muss zunächst ein Paßwort vergeben oder eingegeben werden. Anschließend erscheint ein kleines Bedienfenster, mit welchem der Download durchgeführt werden kann:

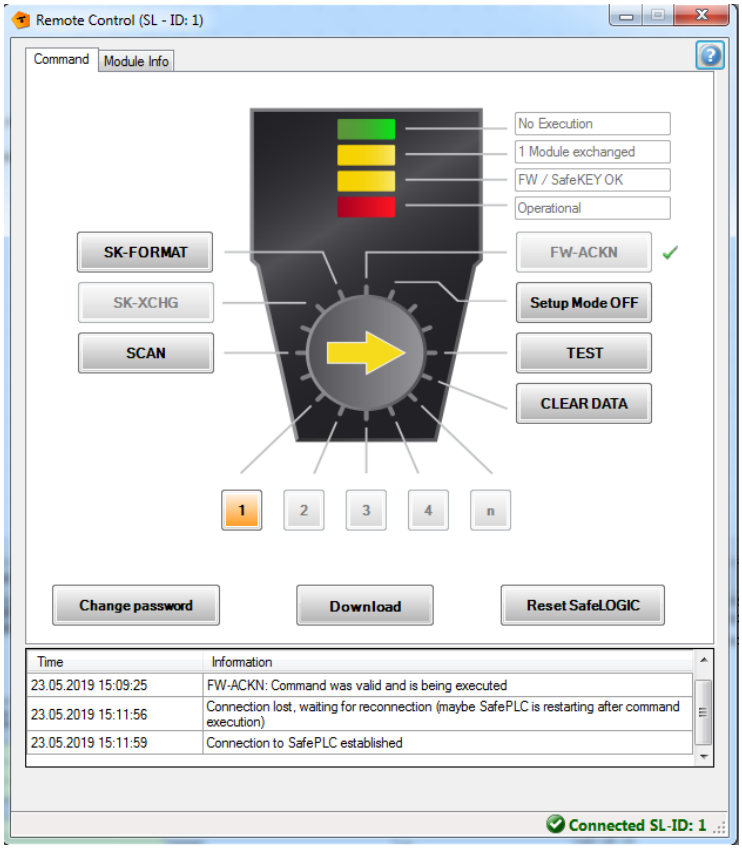
Um einen Download durchzuführen, muss der `Download`-Button aktiviert sein. Dafür muss unter Umständen in den Debug-Mode gewechselt werden. Dann wird, falls vorhanden, die aktuelle Applikation mit `Stop` auf dem SCM angehalten. Anschließend wird der `Download`-Button aktiviert und der Download der eigenen Applikation kann durchgeführt werden. Unter `Download Options` wird `Auto restart` ausgewählt, damit die Applikation automatisch anläuft.

Nach dem Download startet die `SafeLogic` neu. Nun müssen Firmware und neue Teilnehmer an der SL quittiert werden. Für die einzelnen Quittierungen wird die Remote-Bedienung der Steuerung verwendet. Sie wird mit einem Rechtsklick auf die SL1.SM1 im SafetyView gestartet:



Nun muss die neue Firmware quittiert werden (`FW-ACKN`). Die SL startet hierbei neu.

Nach dem Neuanlauf zeigt die SL u.U. an, dass ein neuer Teilnehmer (der Drehgeber) quittiert werden muss:



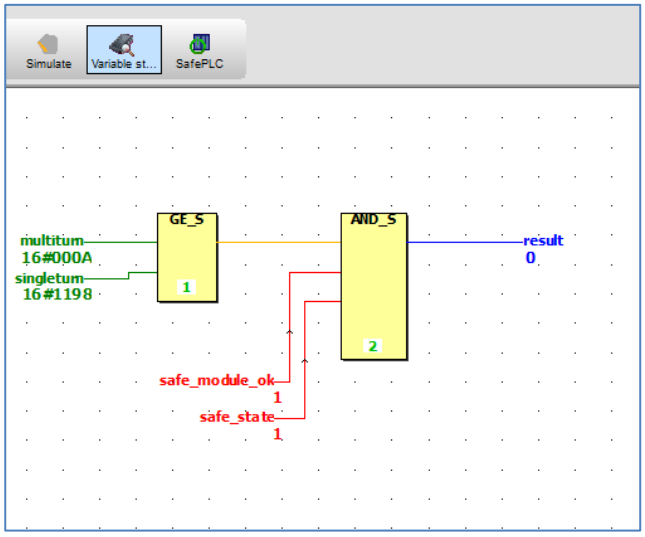
Nachdem die 1 gedrückt wurde, geht der Drehgeber in der sicheren Betrieb. Nun leuchten die Drehgeber die zwei Status-LEDs grün: EPL operational und openSAFETY operational.

Bitte für diesen Vorgang zusätzlich das Handbuch der SL heranziehen, da sich die Blinkmuster und Hinweise je nach Vorgehensweise und vorinstallierter Software leicht unterscheiden können.

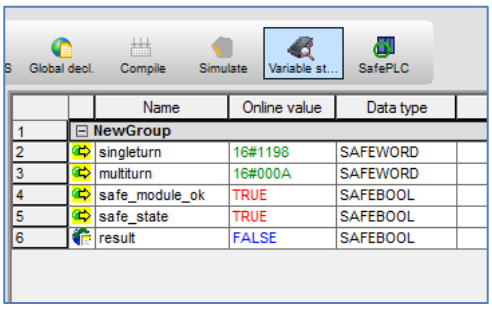
4.2.8 Sicherheitsprogramm testen

Nach Erstellung des Sicherheitsprogramms muss ein vollständiger Funktionstest entsprechend der Automatisierungsaufgabe durchgeführt werden.

Im SafeDESIGNER kann mit dem Button Variable status oder durch die Taste <F10> eine Online-Verbindung aufgebaut werden, bei welcher die aktuellen Variablenwerte im SafeDESIGNER angezeigt werden.



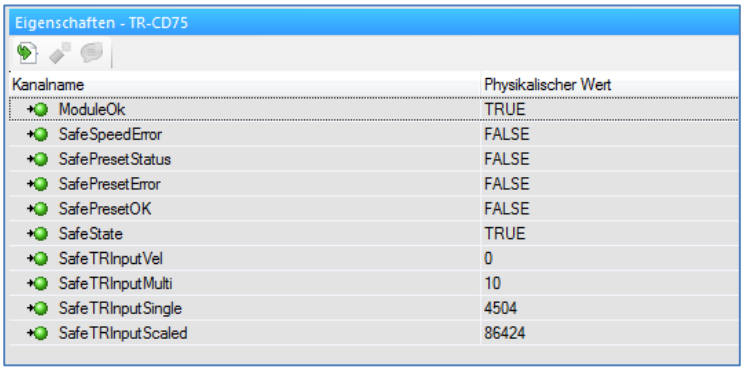
Auch zu den globalen Variablen werden alle Werte angezeigt:



		Name	Online value	Data type
1		NewGroup		
2		singleturn	16#1198	SAFEWORD
3		multiturn	16#000A	SAFEWORD
4		safe_module_ok	TRUE	SAFEBOOL
5		safe_state	TRUE	SAFEBOOL
6		result	FALSE	SAFEBOOL

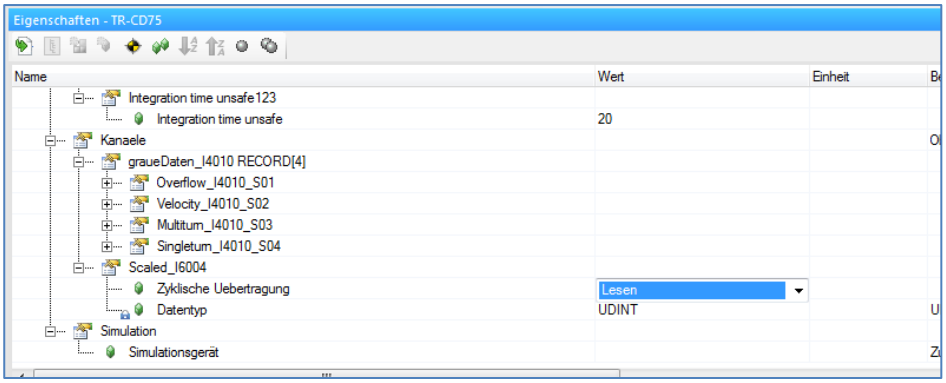
Das Mess-System gibt erst aktuelle Istwerte aus, wenn der Hochlauf über openSAFETY und die sichere Parametrierung erfolgt sind. Bis dahin sind alle Safe-Variablen auf dem Wert 0 und alle Bits der „grauen Daten“ auf 1 gesetzt.

Alle Istwerte der Variablen sind auch im Online-Mode des `Automation Studios` sichtbar, sie werden unter der I/O Zuordnung des Mess-Systems angezeigt:

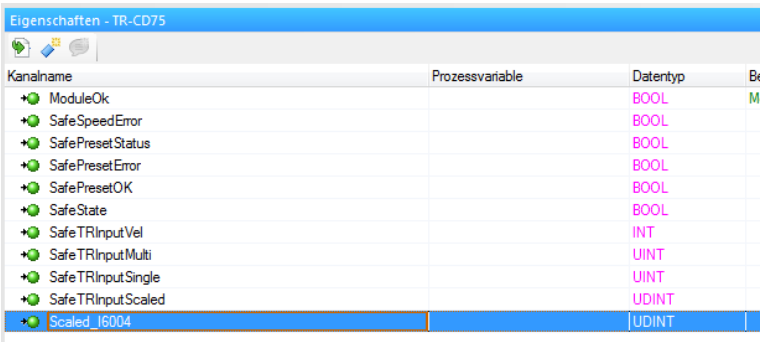


In obiger Abbildung beginnen die Variablen-Namen der zweikanaligen Daten mit dem Wort `Safe` und können auch im „grauen“ Bereich genutzt werden.

Die „grauen“ einkanaligen Werte können ebenfalls gemappt und genutzt werden. Dafür muss für die einzelnen Werte die zyklische Übertragung aktiviert werden. Im folgenden Beispiel wird die zyklische Übertragung für den grauen skalierten Istwert aktiviert:



Dieser wird dann ebenfalls nach dem erneuten Kompilieren des Projekts sowie einem Download im I/O-Bereich angezeigt:



Des Weiteren können noch folgende graue Daten eingeblendet werden:

- Overflow
- Velocity
- Multiturn
- Singleturn

5 Sicherheitsprogramm erweitern - Anwendungsbeispiele

Das unter Kapitel 4.1 erstellte Sicherheitsprogramm wird in den nachfolgenden Abschnitten um Funktionsbeispiele für die Verwendung der Mess-System – Istwerte im `SafeDesigner` erweitert.

Die Beispiele stellen jedoch keine kundenspezifischen Lösungen dar, sondern sollen lediglich Hilfestellung bei unterschiedlichen Automatisierungsaufgaben leisten.

Mit Hilfe der vorgestellten Funktionsbausteine soll die Integration des Mess-Systems in eine Applikation vereinfacht werden.

Bei den nachfolgenden Anwendungsbeispielen
- Sichere Geschwindigkeit (SLS)
- Sichere Stillstandserfassung
- Sichere Positionserfassung
- Sichere Richtungserfassung

werden die Fehlerzustände von den hier vorgestellten Funktionsbausteinen ausgegeben. Die zugehörige Fehlerbehandlung ist nicht Teil der Beispiele und muss vom Anwender umgesetzt werden.

Bei dem Anwendungsbeispiel für die Preset-Durchführung gibt der Funktionsbaustein keinen direkten Fehlerzustand aus, da die Fehlerinformation vom Mess-System selbst generiert wird.

Nutzungsbedingungen der Softwarebeispiele auf Seite 8 beachten!

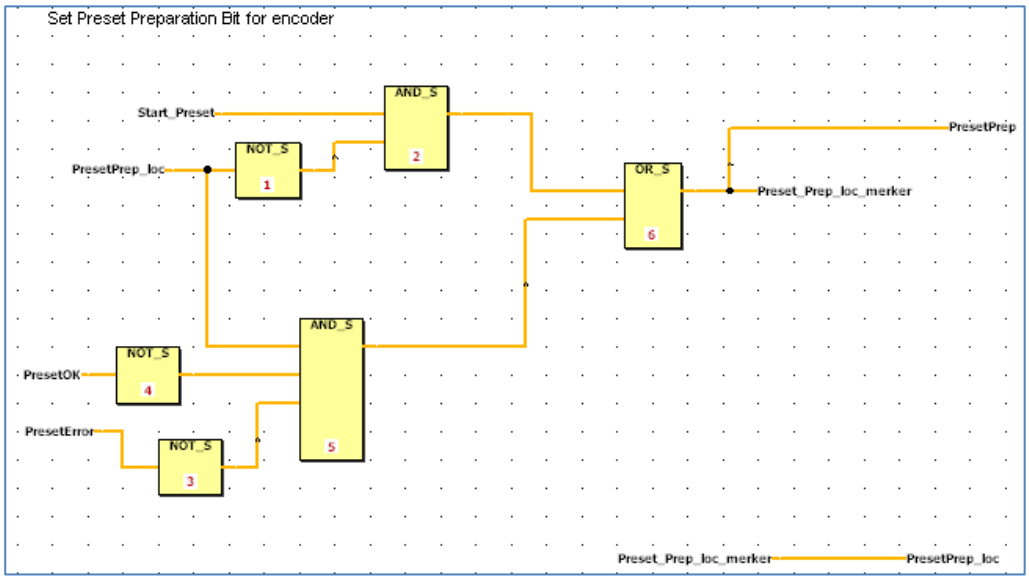
5.1 Preset-Durchführung

Im SafeDESIGNER wird zunächst ein neuer Funktionsblock erstellt (New Function Block nach Rechtsklick auf Logical POUs im Project Tree Window). Der Name wird hier als TR_PRESET_Example vergeben. Anschließend werden folgende lokale Variablen für den Funktionsblock erstellt:

	Name	Data type	Usage	Description	Init	Diag	Feedback
1	**NewGroup**						
2	Start_Preset	SAFEBOOL	VAR_INPUT	Preset Start	SAFEBOOL#0		☐
3	PresetReq	SAFEBOOL	VAR_OUTPUT	Set Preset Request	SAFEBOOL#0	☐	☐
4	PresetPrep_loc	SAFEBOOL	VAR	Set Preset Prep local	SAFEBOOL#0		☑
5	PresetOK	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		☐
6	PresetError	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		☐
7	PresetFinish	SAFEBOOL	VAR_OUTPUT	Preset finished	SAFEBOOL#0	☐	☐
8	PresetFinish_loc	SAFEBOOL	VAR		SAFEBOOL#0		☑
9	PresetFinish_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		☐
10	PresetPrep	SAFEBOOL	VAR_OUTPUT	Set Preset Preparation	SAFEBOOL#0	☐	☐
11	Preset_Prep_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		☐
12	PresetReq_loc	SAFEBOOL	VAR		SAFEBOOL#0		☑
13	Reset_Lock	SAFEBOOL	VAR_INPUT	Lock Reset	SAFEBOOL#0		☐
14	PresetReq_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		☐

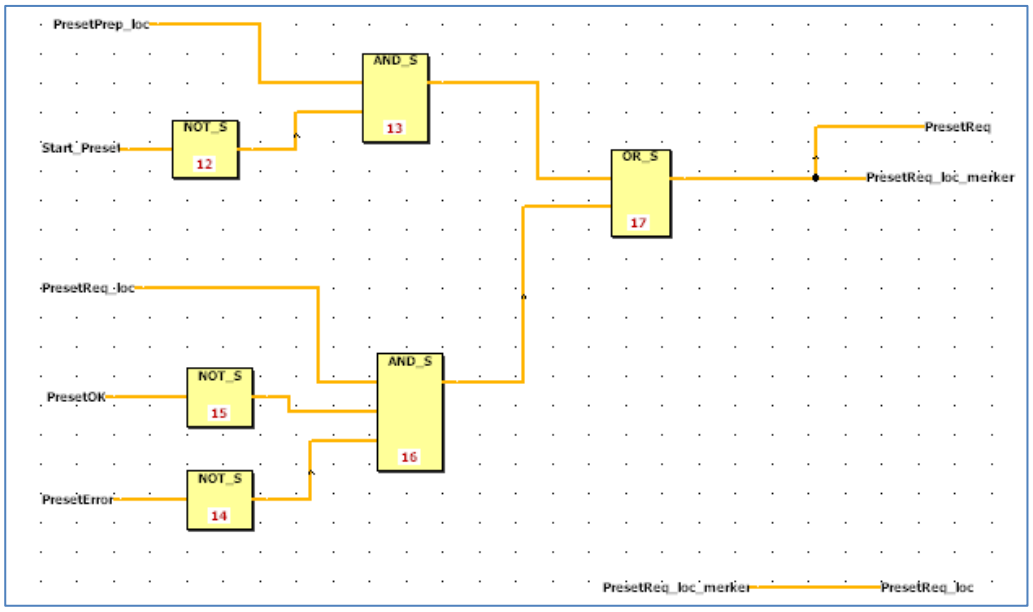
Folgende Darstellungen zeigen den Aufbau des neuen Funktions-Blocks TR_PRESET_Example:

1. Setzen von Preset Preparation



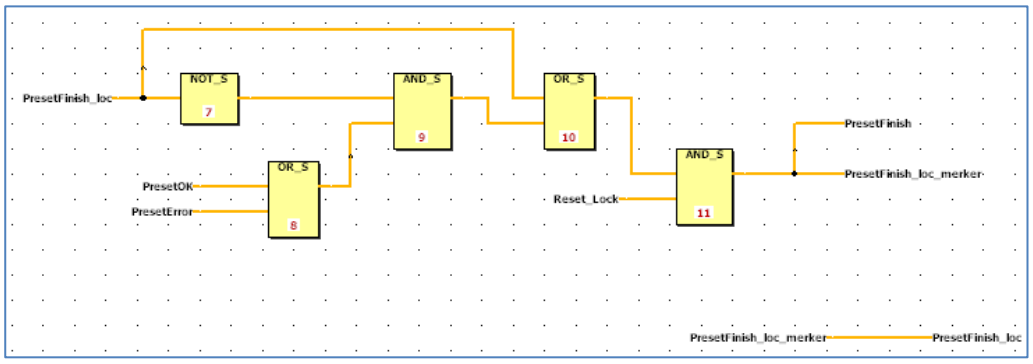
Im ersten Abschnitt wird durch das Setzen des Start_Preset-Eingangs das Mess-System Ausgangssignal PresetPrep-Bit gesetzt. Das Signal bleibt gesetzt, bis der Preset-Ablauf im Mess-System beendet ist. Wenn der Preset-Ablauf ohne Fehler ausgeführt werden konnte wird vom Mess-System PresetOK gesetzt und im Fehlerfall PresetError.

2. Setzen von `Preset Request`



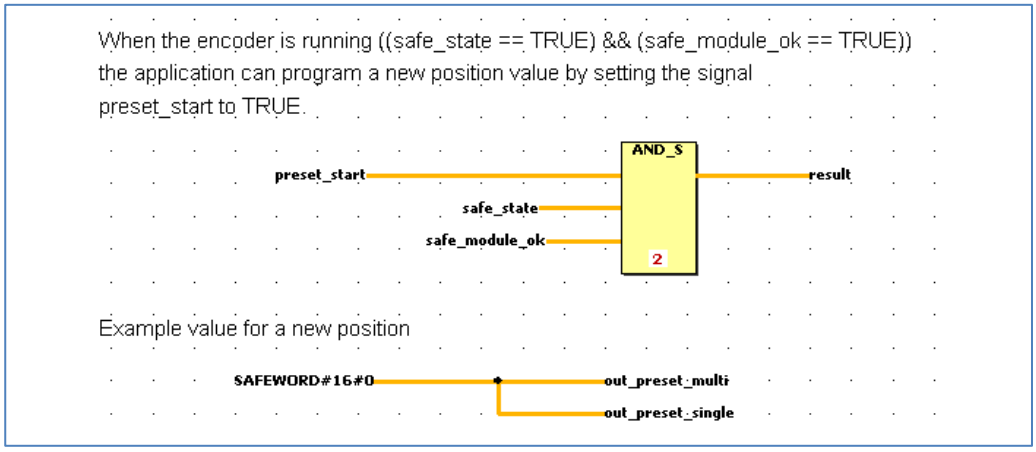
Wenn das Ausgangssignal `PresetPrep` zum Mess-System gesetzt ist und das `Start_Preset`-Signal wieder auf `FALSE` gesetzt wird, so wird das Signal `PresetReq` gesetzt. Das Signal bleibt gesetzt, bis der Preset-Ablauf im Mess-System beendet ist. Wenn der Preset-Ablauf ohne Fehler ausgeführt werden konnte wird vom Mess-System `PresetOK` gesetzt und im Fehlerfall `PresetError`.

3. Setzen von `Preset Finish`

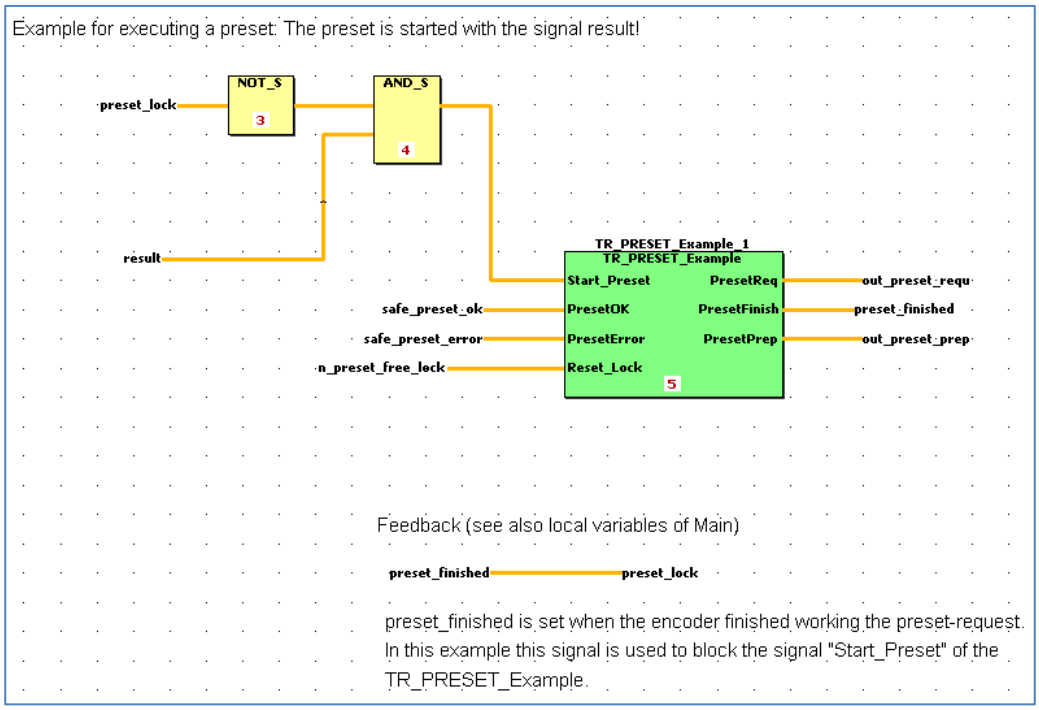


In diesem Abschnitt wird die Information gespeichert, dass die Durchführung beendet wurde. `Reset_Lock` muss hierfür immer auf „1“ gesetzt sein, da die Variable low-aktiv ist.

Um den Preset zu starten wird ein neues Signal mit dem Namen `preset_start` erstellt, welches zum Beispiel mit einem digitalen Eingang verbunden werden kann. Der Main-Funktions-Block wird nun so ergänzt, dass als neue Vorgabe fix der Wert 0 an das Mess-System gesendet wird:



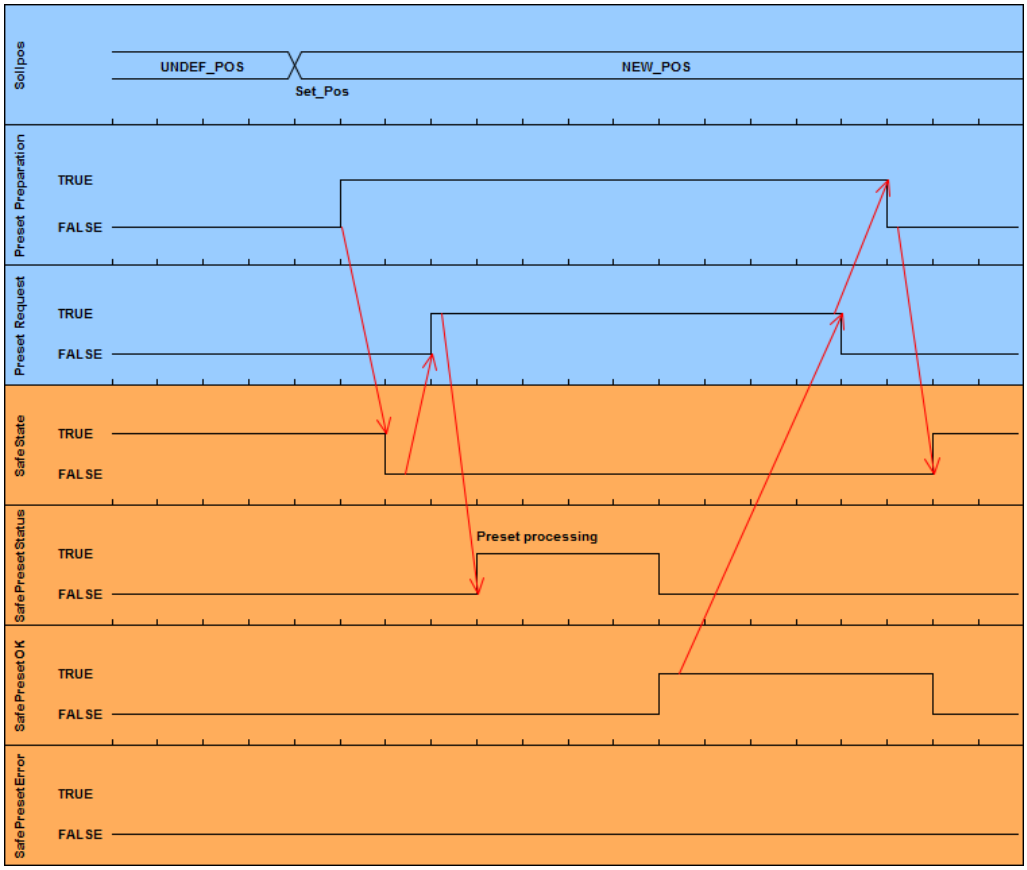
Die Ausgänge `PresetPrep` und `PresetReq` zum Mess-System hin werden mit den entsprechenden Ausgängen des `TR_PRESET_Example`-Funktionsblocks verbunden. Der Preset wird mit Hilfe eines `Lock`-Signals nur einmal je `Start_Preset`-Signal ausgeführt: Wenn das Signal `preset_lock` noch nicht gesetzt ist und das Signal `result` gesetzt wird. Somit wird mit dem vorliegenden Programm ein Preset nach dem Hochlauf auf den neuen Sollwert 0 durchgeführt, sobald das Signal `preset_start` vom Anwender auf `TRUE` gesetzt wird.



Um einen weiteren Preset auszuführen muss im vorliegenden Beispiel das Signal `n_preset_free_lock` zunächst wieder auf `FALSE` gesetzt werden (Startwert = `TRUE`). Dies bewirkt, dass das Ausgangssignal `preset_finished` wieder zurückgesetzt wird. Für einen weiteren PRESET kann nun `preset_start` und somit `result` wieder auf `TRUE` gesetzt werden.

Im nachfolgenden Timing-Diagramm wird der fehlerfreie Ablauf der Preset-Funktion dargestellt.

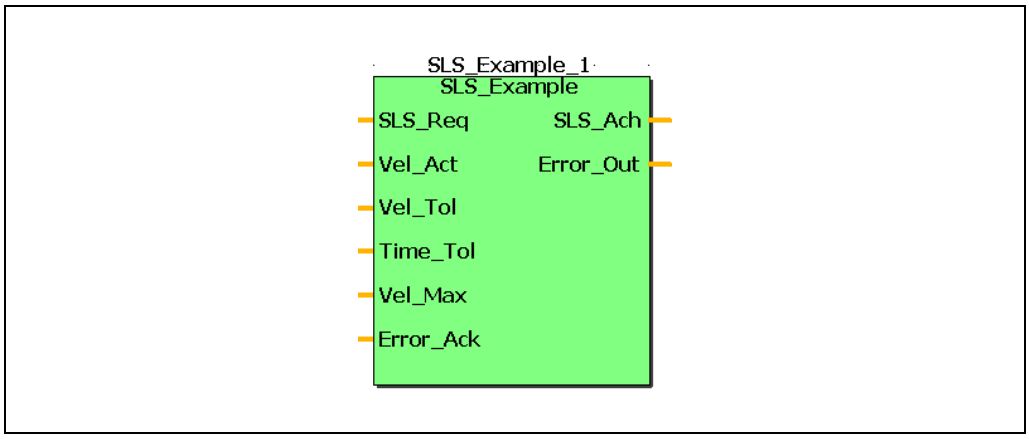
blauer Bereich: Ausgangssignale Steuerung -> Mess-System
oranger Bereich: Eingangssignale Mess-System -> Steuerung



5.2 Sichere Geschwindigkeit (SLS)

Der Funktionsbaustein `SLS_Example` liefert auf Anforderung eine Überwachung auf sichere Geschwindigkeit (SLS = Safely Limited Speed). Der Betrag der aktuellen Mess-System – Geschwindigkeit `Vel_Act` muss nach der Anforderung `SLS_Req` innerhalb der parametrierten Toleranzzeit `Time_Tol` unterhalb der parametrierten sicheren Geschwindigkeit `Vel_Tol` sein. Ist dies der Fall, wird der Ausgang `SLS_Ach` gesetzt. Im Fehlerfall wird der Ausgang `Error_Out` gesetzt.

Ein Fehler kann mit `Error_Ack` quittiert werden.

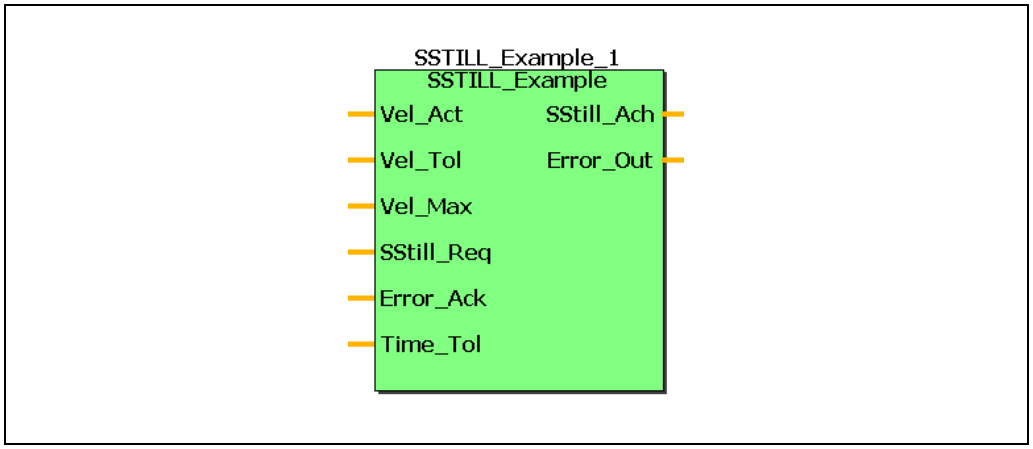


Signal	Typ	Richtung	Beschreibung
`SLS_Req`	SAFEBOOL	Eingang	Anforderung der sicheren Geschwindigkeit mit `SLS_Req` = `TRUE`
`Vel_Act`	SAFEINT	Eingang	Sichere Geschwindigkeit des Mess-Systems
`Vel_Tol`	SAFEINT	Eingang	Grenze für sichere Geschwindigkeit SLS
`Time_Tol`	SAFETIME	Eingang	Maximale Zeit von Anforderung bis SLS
`Vel_Max`	SAFEBOOL	Eingang	Mess-System Geschwindigkeitsüberlauf
`Error_Ack`	SAFEBOOL	Eingang	Fehlerquittierung
`SLS_Ach`	SAFEBOOL	Ausgang	SLS erreicht
`Error_Out`	SAFEBOOL	Ausgang	Fehler liegt an, z.B. Zeitüberschreitung

5.3 Sichere Stillstandserfassung

Der Funktionsbaustein `SSTILL_Example` liefert eine Überwachung auf einen sicheren Stillstand. Sobald der Betrag der aktuellen Mess-System – Geschwindigkeit `Vel_Act` kleiner ist als die parametrierte Maximalgeschwindigkeit `Vel_Tol`, wird der Ausgang `SStill_Ach` gesetzt. Nach der Anforderung auf Stillstand `SStill_Req` muss `Vel_Act` innerhalb der parametrierten Toleranzzeit `Time_Tol` unterhalb der parametrierten Maximalgeschwindigkeit `Vel_Tol` für einen Stillstand sein. Ist dies nicht der Fall, wird der Ausgang `Error_Out` gesetzt.

Ein Fehler kann mit `Error_Ack` quittiert werden.

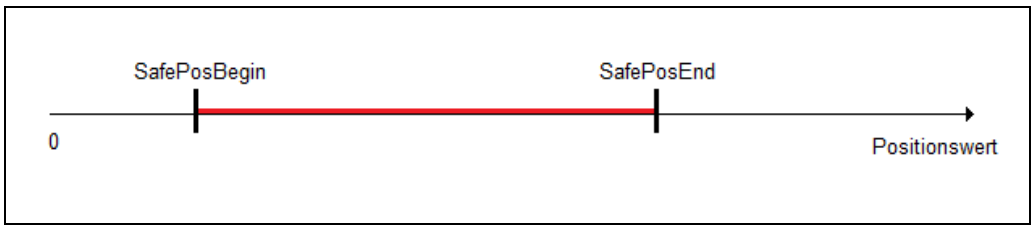


Signal	Typ	Richtung	Beschreibung
`Vel_Act`	SAFEINT	Eingang	Sichere Geschwindigkeit des Mess-Systems
`Vel_Tol`	SAFEINT	Eingang	Grenze für Stillstand (Toleranz)
`Vel_Max`	SAFEBOOL	Eingang	Mess-System Geschwindigkeitsüberlauf
`SStill_Req`	SAFEBOOL	Eingang	Anforderung des sicheren Stillstands mit `SStill_Req` = `TRUE`
`Error_Ack`	SAFEBOOL	Eingang	Fehlerquittierung
`Time_Tol`	SAFETIME	Eingang	Maximale Zeit von Anforderung bis Stillstand
`SStill_Ach`	SAFEBOOL	Ausgang	`TRUE`: Stillstand
`Error_Out`	SAFEBOOL	Ausgang	Fehler liegt an, z.B. Zeitüberschreitung

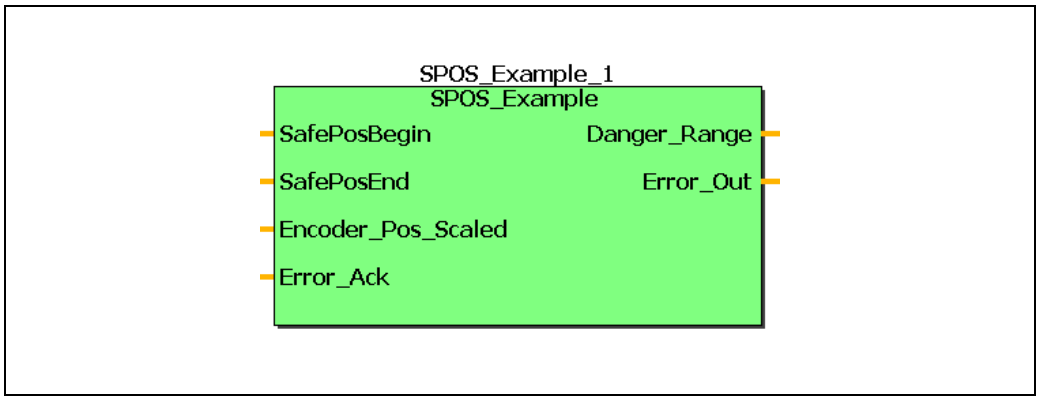
5.4 Sichere Positionserfassung

Mit dem Funktionsbaustein `SPOS_Example` kann ein Gefahrenbereich festgelegt werden. Mit Hilfe von `SafePosBegin` und `SafePosEnd` kann ein Istwert-Bereich definiert werden, in welchem der Ausgang `Danger_Range` gesetzt wird. Der Bereich muss hierbei immer so gewählt werden, dass `SafePosBegin` < `SafePosEnd` ist. Ein Überlauf ist nicht erlaubt und führt zu einer Fehlerausgabe.

Ein Fehler kann mit `Error_Ack` quittiert werden.



Im roten Bereich wird der Ausgang `Danger_Range` gesetzt.



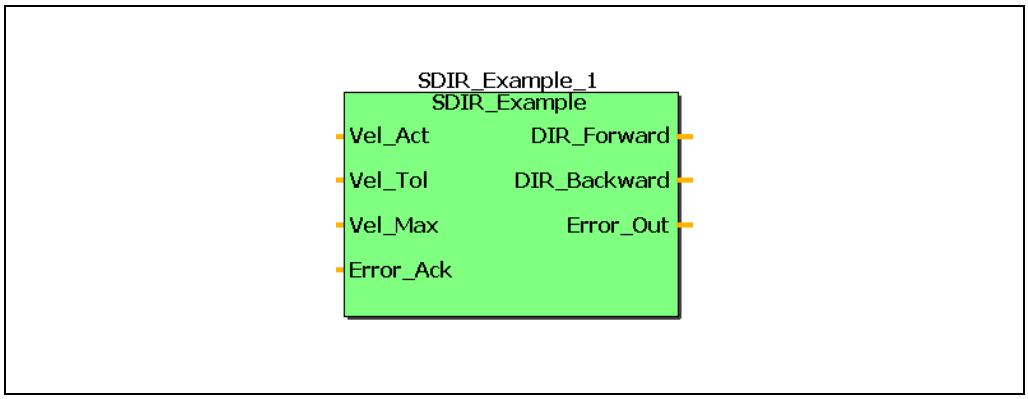
Signal	Typ	Richtung	Beschreibung
`SafePosBegin`	SAFEDWORD	Eingang	Start des gefährlichen Bereichs
`SafePosEnd`	SAFEDWORD	Eingang	Ende des gefährlichen Bereichs
`Encoder_Pos_Scaled`	SAFEDWORD	Eingang	Aktueller Mess-System Istwert
`Error_Ack`	SAFEBOOL	Eingang	Fehlerquittierung
`Danger_Range`	SAFEBOOL	Ausgang	`TRUE`: Mess-System Istwert befindet sich im gefährlichen Bereich: `SafePosBegin` ≤ `Encoder_Pos_Scaled` ≤ `SafePosEnd`
`Error_Out`	SAFEBOOL	Ausgang	Fehler liegt an, z.B. Zeitüberschreitung

5.5 Sichere Richtungserfassung

Mit dem Funktionsbaustein `SDIR_Example` kann die Zählrichtung des Mess-Systems ermittelt werden:
- positive Richtung = steigende Istwerte
- negative Richtung = fallende Istwerte

Um eine unkontrollierte Ausgabe durch leichte Bewegungen im Stillstand zu umgehen, kann eine Geschwindigkeit definiert werden, ab der die Richtungserkennung aktiv ist. Unter dieser Geschwindigkeit `Vel_Tol` sind die Ausgaben `DIR_Forward` und `DIR_Backward` immer `FALSE`.

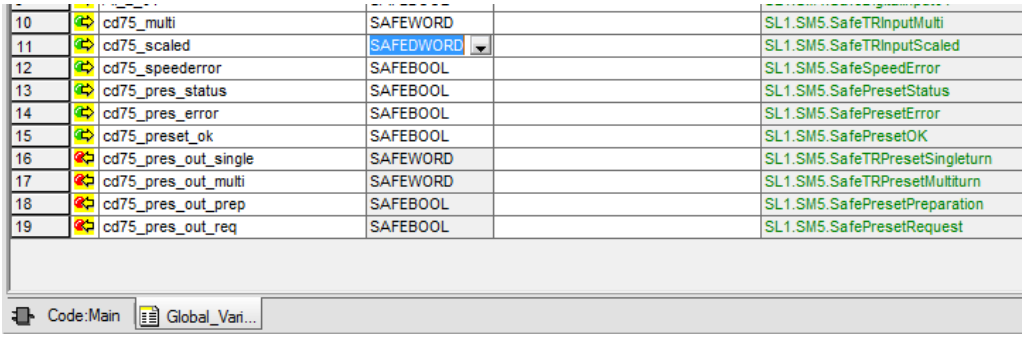
Ein Fehler kann mit `Error_Ack` quittiert werden.



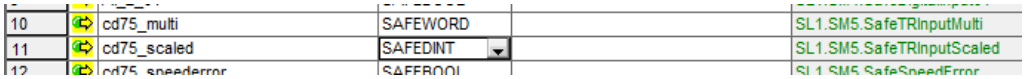
Signal	Typ	Richtung	Beschreibung
`Vel_Act`	SAFEINT	Eingang	Sichere Geschwindigkeit des Mess-Systems
`Vel_Tol`	SAFEINT	Eingang	Grenze für Richtungsmessung
`Vel_Max`	SAFEBOOL	Eingang	Mess-System Geschwindigkeitsüberlauf
`Error_Ack`	SAFEBOOL	Eingang	Fehlerquittierung
`DIR_Forward`	SAFEBOOL	Ausgang	`TRUE`: Mess-System Istwert steigend
`DIR_Backward`	SAFEBOOL	Ausgang	`TRUE`: Mess-Sysem Istwert fallend
`Error_Out`	SAFEBOOL	Ausgang	Fehler liegt an, z.B. Geschwindigkeitsüberlauf

5.6 Typumwandlung der Mess-System - Istwerte im SafeDESIGNER

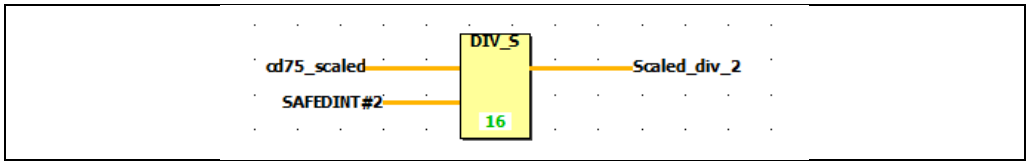
Die Mess-System-Istwerte für Singleturn, Multiturn und skalierter Gesamtwert werden von TR-Electronic bewusst als vorzeichenlose Werte bereitgestellt.
Im SafeDESIGNER können einige Funktionen nicht mit vorzeichenlosen Werten durchgeführt werden. Dazu gehört beispielsweise die Division. Wenn der Anwender eine Division mit einem vorzeichenlosen Wert im SafeDESIGNER durchführen will, muss er vorzeichenlose Werte auf einen vorzeichenbehafteten Wert casten.



In obiger Abbildung ist beispielhaft zu erkennen, dass das Mess-System für den 32-Bit-Gesamtwert zunächst den Datentyp SAFEDWORD besitzt. Dabei handelt es sich um einen vorzeichenlosen 32-Bit-Wert. Um den skalierten Gesamtwert für eine Division zu nutzen, kann der Wert in der Variablentabelle auf SAFEDINT umgestellt werden:



Nun kann der Wert im SafeDESIGNER als SAFEDINT verwendet werden:



Was für den Anwender zu beachten ist, ist das der Wert tatsächlich niemals negativ wird, weil das oberste Bit niemals gesetzt wird. Folgende Wertebereiche gelten unabhängig davon, ob der Wert vorzeichenbehaftet interpretiert wird oder nicht:

Singleturn: 0, 1, 2,...8189, 8190, 8191 → Überlauf auf 0 ,1, 2....
Multiturn: 0, 1, 2,...32766, 32767 → Überlauf auf 0,1, 2....
Gesamtwert: 0, 1, 2,... 268435455 → Überlauf auf 0 ,1, 2....

Die Werte bewegen sich also immer im positiven Bereich.

Gefahr von Körperverletzung und Sachschaden durch einen Istwertsprung zwischen 0 und Maximalwert!

⚠WARNUNG

ACHTUNG

- Der Wertesprung der Istwerte zwischen 0 und Maximalwert müssen vom Anwender entsprechend berücksichtigt werden. Das bedeutet, dass wenn in Position 0 rückwärts gedreht wird, der nächste Istwert nicht -1 sondern der Maximalwert (z.B. 8191 für Singleturn) ist. Dieses Verhalten ist unabhängig davon, ob der Wert SafeDESIGNER als SAFEDWORD oder SAFEDINT interpretiert wird.

6 Download - Softwarebeispiele

Legacy-Beispiel für Automation Studio Versionen <V4.5:

www.tr-electronic.de/f/zip/TR-ECE-TI-MUL-0268

Beispiel mit XDD-/OSDD-Gerätebeschreibung ab Automation Studio Versionen V4.5:

www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0351

Absolute Encoder CDx-75 POWERLINK/openSAFETY

Parameterization with B&R control system X20 CPU

TR-Electronic GmbH
D-78647 Trossingen
Eglishalde 6
Tel.: (0049) 07425/228-0
Fax: (0049) 07425/228-33
E-mail: info@tr-electronic.de
http://www.tr-electronic.de

Copyright protection

Subject to modifications

The right to make any modifications in the interest of technical progress is reserved.

Document information

Release date / Rev. date:	06/18/2019
Document / Rev. no.:	TR - ECE - TI - DGB - 0264 - 07
File name:	TR-ECE-TI-DGB-0264-07.docx
Author:	MÜJ

Font styles

Italic or **bold** font styles are used for the title of a document or are used for highlighting.

`Courier` font displays text that is visible on the screen and software/software menu selections.

″ < > ″ indicates keys on your computer keyboard (such as <RETURN>).

Brand names

Products, names and logos in this Manual are only mentioned for information purposes and may be brands of their owners without this fact being expressly declared.

Contents

Revision index

Revision	Date	Index
First edition	02/26/2015	01
Notes for operation with a SLX CPU of B&R	04/09/2015	02
Removed: Notes for operation with a SLX CPU of B&R	06/25/2015	03
Modifications under Chapter “Preset implementation” New function module - examples: - Safely Limited Speed (SLS) - Safe standstill detection - Safe position detection - Safe direction detection	10/02/2015	04
Chapter “Typecast of input values in the SafeDESIGNER“ added	10/18/2016	05
Specification of the password, chapter 4.1.1 Access protection	03/09/2018	06
Update for encoder with openSAFETY-Stack V1.5; usage of XDD-/OSDD-device description in the Automation Studio V4.5	06/18/2019	07

1 Glossary

Name	Description
AS	Automation Studio by B&R: development environment for POWERLINK projects
B&R	Bernecker + Rainer Industrie-Elektronik GmbH
EPL	Ethernet PowerLink
FB(s)	Function block(s)
FBD	Programming language, function block diagram
FU(s)	Function(s)
Gray data	Single-channel actual values via POWERLINK, not safety instrumented
LD	Programming language, ladder diagram
openSAFETY	Open and bus-independent safety standard for all Industrial Ethernet solutions
POWERLINK CN	Controlled Node: slave user in a POWERLINK network
POWERLINK MN	Managing Node: initiates POWERLINK communication and parameterizes the CNs
SCM	Safety Configuration Manager: initializes the SNs in an openSAFETY project via the underlying field bus
SL	SafeLogic CPU by B&R
SN	Safety Node: slave user in an openSAFETY project
XML	EXtensible Markup Language

2 General

The present “Technical Information” addresses the following topics:

- Creating the safety program
- Determining the safe parameters
- Using the safety instrumented data channel

The “Technical Information” is available as a separate document.

2.1 Scope

This “Technical Information” is only applicable to measuring system model ranges featuring a ***POWERLINK*** interface and an ***openSAFETY*** profile in connection with a B&R X20 control:

- CDV-75
- CDH-75

The products are labeled with affixed nameplates and are components of a system.

This means that, all in all, the following documentations are applicable:

- B&R X20 System User Manual,
  Order no.: MAX20-ENG
- B&R Integrated Safety Technology User Manual,
  Order no.: MASAFETY-ENG
- B&R X20 data sheet V1.51 X20SL80xx
- The responsible organization’s system-specific operating instructions
- Safety Manual TR-ECE-BA-GB-0107
- Interface-specific User Manual TR-ECE-BA-GB-0110
- This optional “Technical Information”

3 Safety instructions

3.1 Definition of symbols and notes

⚠ DANGER	means that death or serious injury will occur if the user fails to take the respective precautionary measures.
⚠ WARNING	means that death or serious injury may occur if the user fails to take the respective precautionary measures.
⚠ CAUTION	means that minor injuries may occur if the user fails to take the respective precautionary measures.
NOTICE	means that damage to property may occur if the user fails to take the respective precautionary measures.
	indicates important information or features and application tips for the product used.

3.2 Organizational measures

Prior to commencing work, personnel handling the measuring system must have read and understood the Safety Manual (TR-ECE-BA-GB-0107), in particular chapter “Basic safety instructions”.

3.3 Personnel qualification

The measuring system may only be configured by qualified specialist personnel; see B&R User Manual.

3.4 Terms of use of the software examples

⚠ WARNING TR-Electronic GmbH assumes no liability and no warranty for the flawless operation of the safety program and the application examples.

NOTICE The software examples offered for download are solely for demonstration purposes and used by the user at its own risk.

The password for the examples and for the zip-archive of the software examples is always “abc123”.

4 Creating the safety program - example configuration

The following two chapters are showing two example configurations. The first one uses the legacy device description which is installed as update in the B&R Automation Studio. The second example uses the current device description. It is a combination of XDD- and OSDD-file. This device description can be used with the Automation Studio V4.5 and newer.

4.1 Legacy device description (Update)

This chapter describes the procedure to be followed while creating an example safety program using the B&R projecting software Automation Studio (V4.0.18.71) and the optional package SafeDESIGNER (V3.0.16).

The safety program is created in SafeDESIGNER which features a code editor for developing the program for the safety control. The programming languages used to do this are FBD and LD.

The safety program is executed on an X20 SafeLOGIC (SL8010). This control is a normal POWERLINK CN participating in field bus communication and, in turn, is itself the openSAFETY SCM.

An X20 CP1584 is used as POWERLINK MN. It features a POWERLINK application which is also able to evaluate information from the measuring system (e.g., “gray channel” actual values, and others).

4.1.1 Access protection

Access to the SafeDESIGNER project is protected by a password prompt. The development and commissioning phases each have their own password. Access to the SL8010, for example for programming purposes, is also protected by a password.

The password is: abc123

4.1.2 Requirements

⚠ WARNING ***If the safety program is projected improperly, there is the risk that the fail-safe function might be deactivated!***

- The safety program may only be created based on the system documentation B&R supplies with its software and hardware.
- The descriptions below only refer to the steps required, without taking all instructions from the B&R manuals into account. It is essential to observe and comply with the information and instructions provided in the B&R manuals, particularly the safety instructions and warnings.
- The projecting procedure shown is an example only. Users are therefore obliged to check whether project planning is appropriate for their application and adjust it if necessary. This also includes selecting the appropriate safety instrumented hardware components and defining the software requirements.

Software components used for the example configuration:

- `Automation Studio` V4.0.18.71
- `SafeDESIGNER` V3.0.16.262

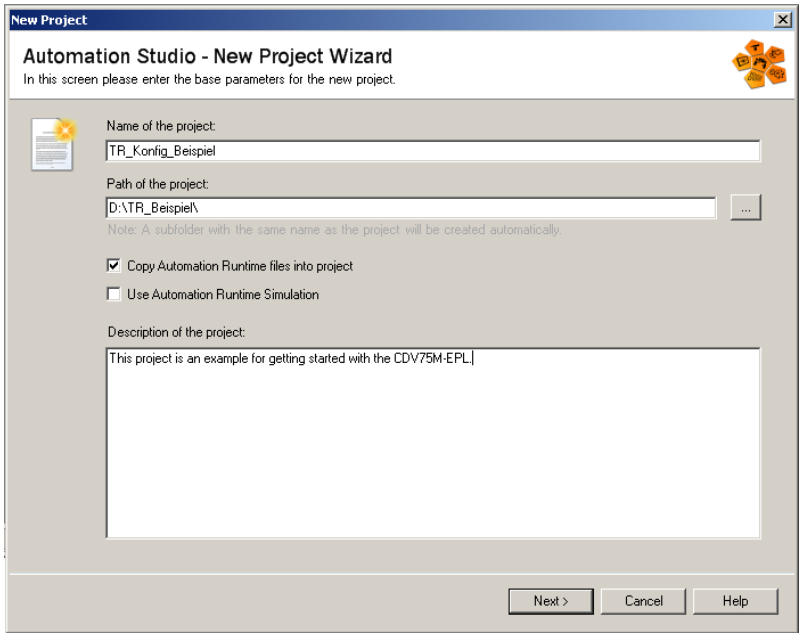
The following versions can also be used. The setup was commissioned with `Automation Studio` version 4.1.4.401 and `SafeDESIGNER` version 4.1.0.320.

Hardware components used for the example configuration:

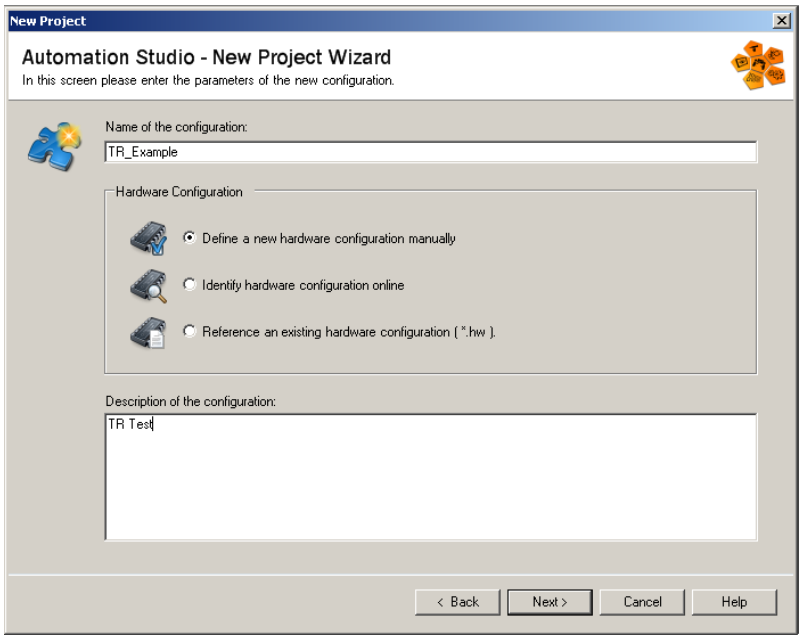
- POWERLINK-MN: `X20CP1584` with `X20IF1082-2`
- openSAFETY SCM: `X20SL8010`

4.1.3 Hardware configuration

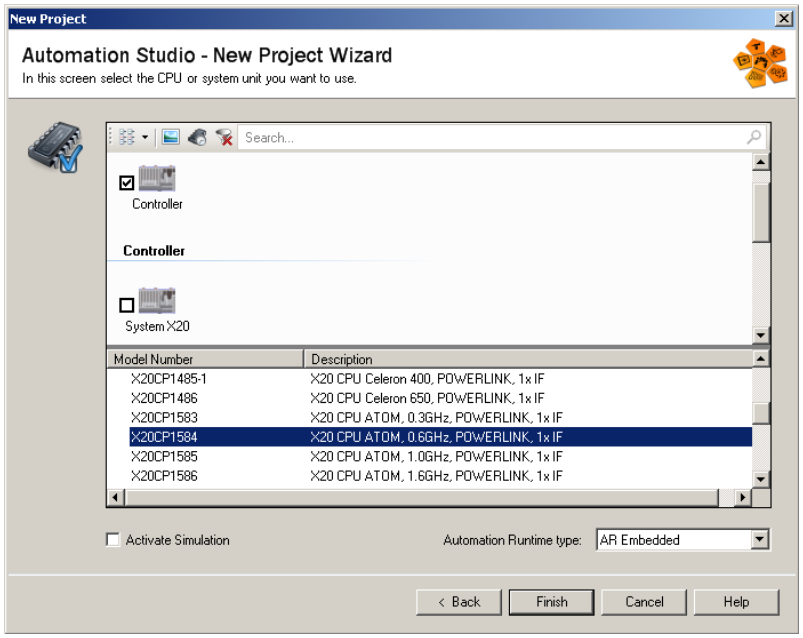
➢ Start `Automation Studio` and create a new project.



➢ This creates a new hardware configuration:



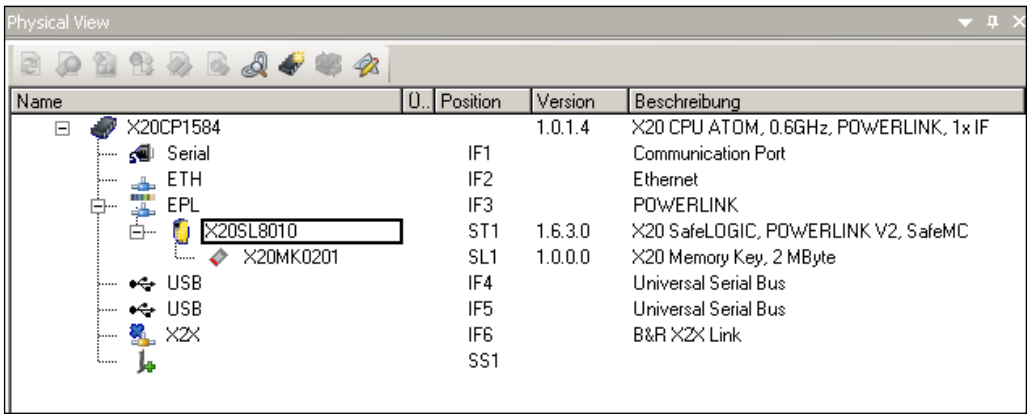
- Select the correct CPU in your final step: `X20CP1584`



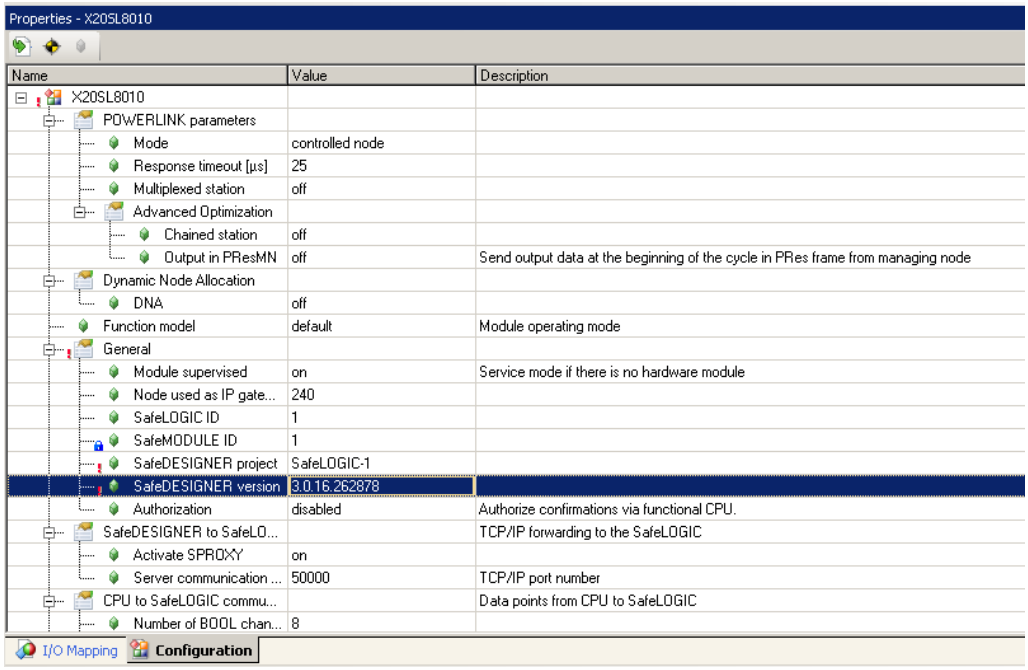
- Go to `-> Project -> Change Runtime Versions...` and select Safety Release 1.6.

The measuring system requires Safety Release 1.6 or higher.

- Add the SCM to the EPL net by dragging the corresponding device from the hardware catalog to the POWERLINK interface in `Physical View`:

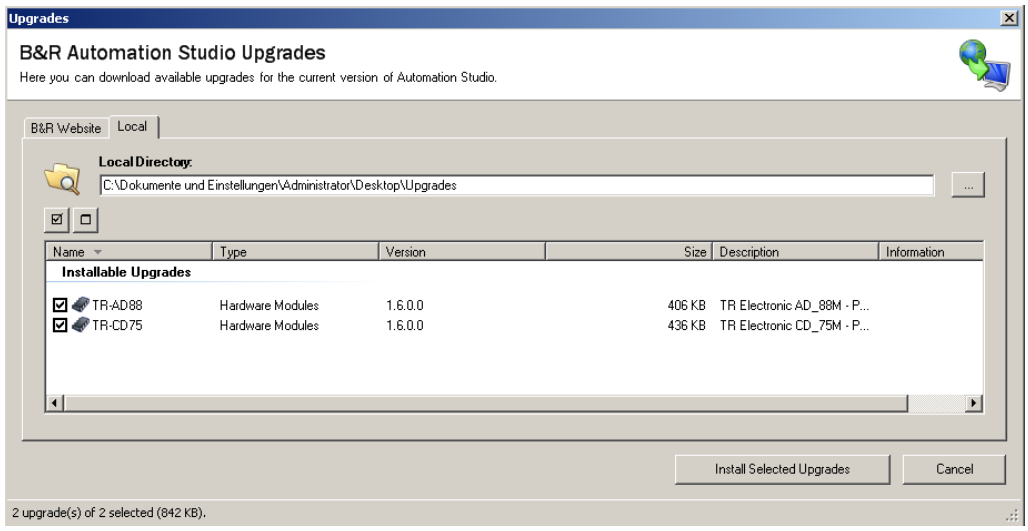


➢ Select the `SafeDESIGNER` version from the `SafeLOGIC` configuration:

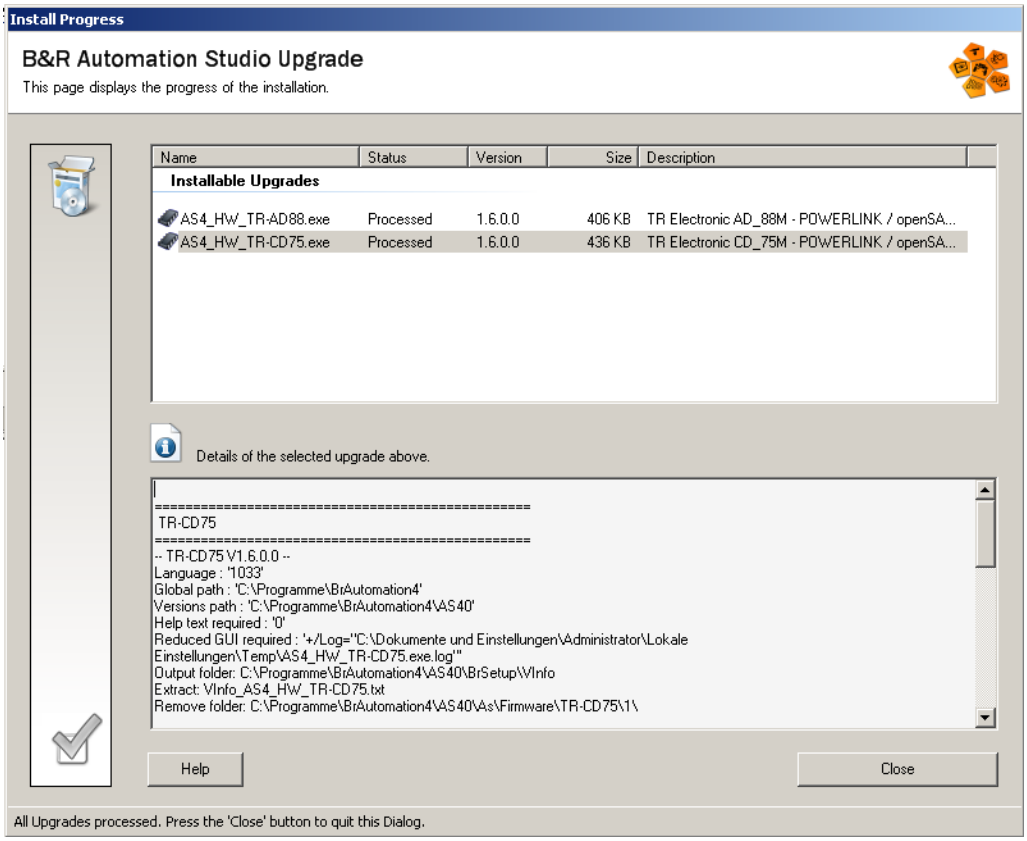


Before the measuring system can be added, the device description must be installed. Since the function of importing an xml-based device description for openSAFETY devices has not been available yet when this document was under preparation, the device description is imported through an AS update. This AS update can be obtained from TR-Electronic. The AS should be restarted after installation.

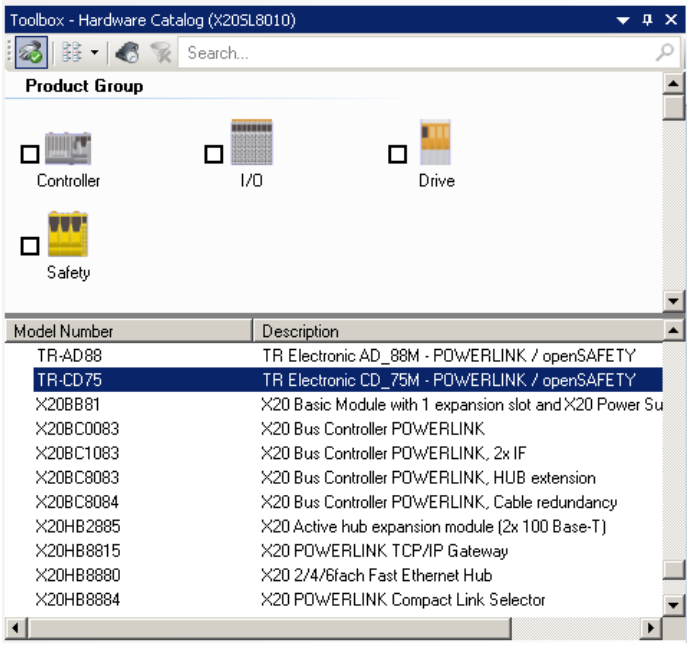
➢ Select the `Tools -> Upgrades...` menu and start the AS update:



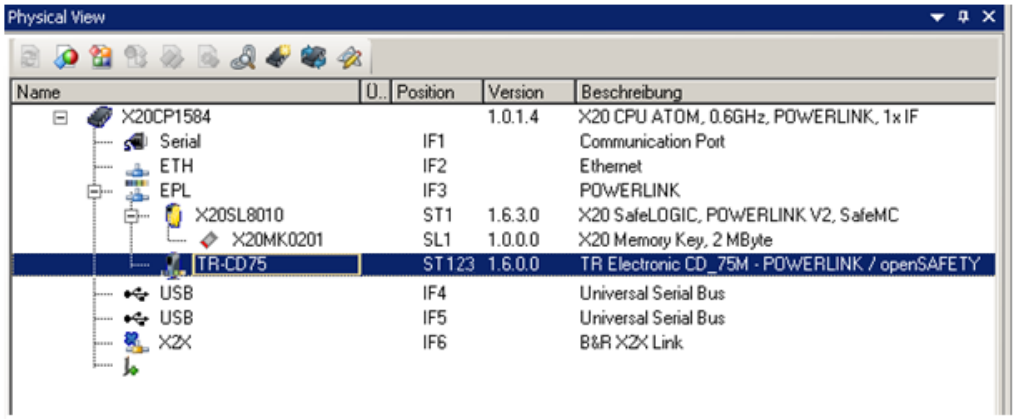
➢ Select the upgrades and click on `Install Selected Upgrades`:



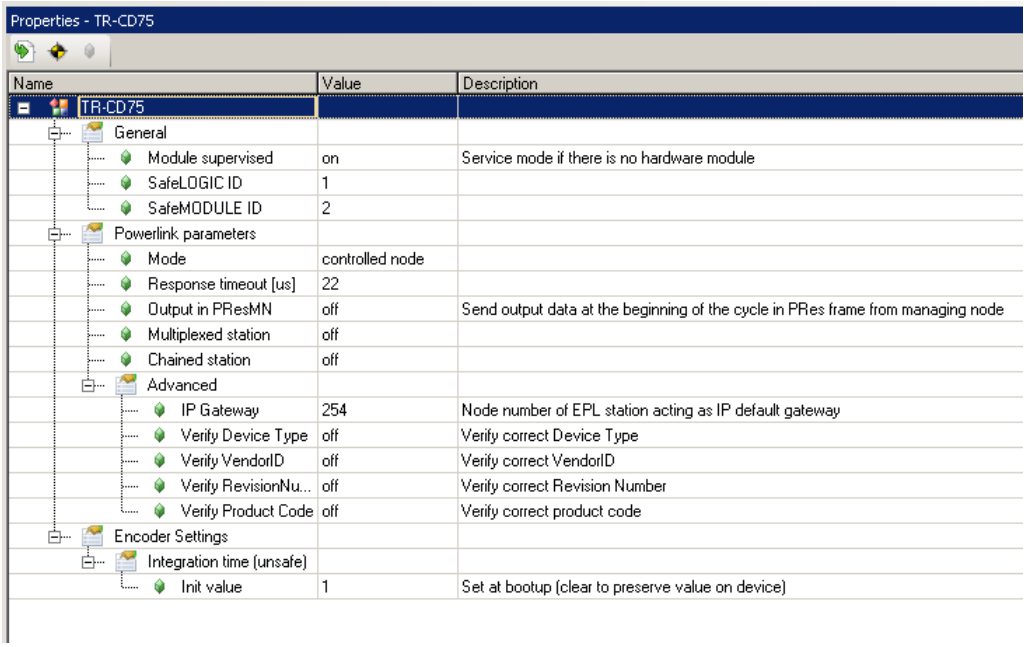
The hardware catalog will display the upgrades selected:



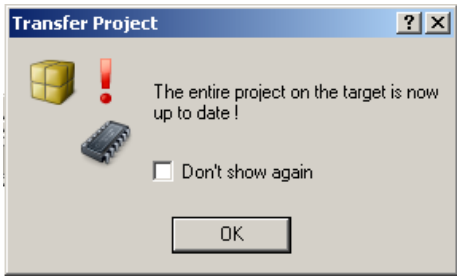
After completed installation, the measuring system can also be added in `Physical View`. The correct POWERLINK node ID must be assigned. In the illustrated instance, this is node ID 123 (0x7B). This node ID must then be set for the measuring system.



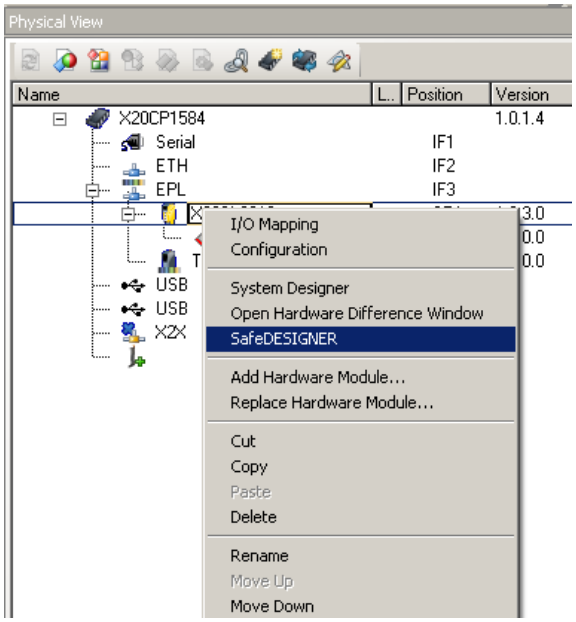
Initially, the measuring system configuration is not adjusted in `Automation Studio`. In the illustrated instance, `SafeMODULE ID` = 2 is assigned to the measuring system in the openSAFETY network.



The project can now be saved and compiled with <Ctrl>+<F7>. Provided there is a connection to the EPL control, you can directly transfer the project to the X20CP1584 when prompted to do so on the next display. If you wish, you can also do this at a later point. Once the project has been transferred, the following message is displayed after the download:

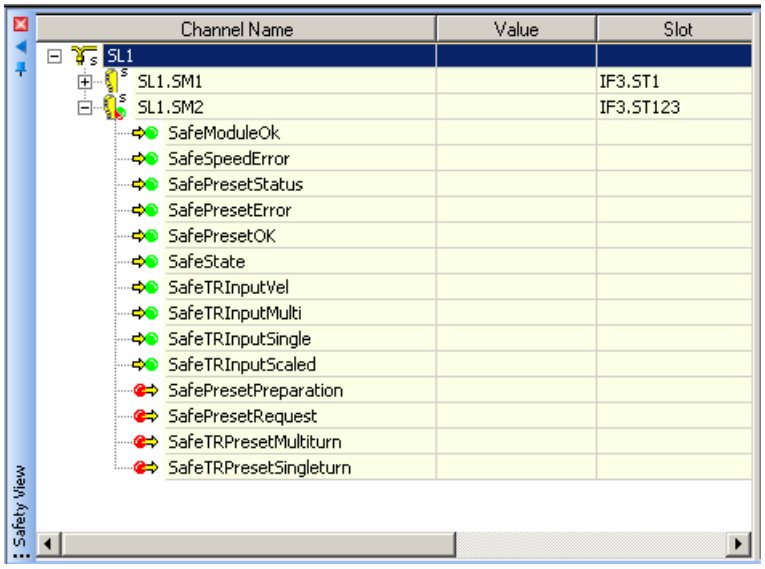


➢ Start SafeDESIGNER by right-clicking on SafeLOGIC:



➢ Assign and confirm the passwords for the SafeDESIGNER project. In the example, this is “abc123”.

In `Safety View`, the display lists the openSAFETY users at the bottom left by default. It shows the SCM (SL1.SM1) and the measuring system (SL1.SM2). The number “2” corresponds to the `SafeMODULE ID`. The `Slot` column shows the node ID of the measuring system. The list below the measuring system shows all data relating to the measuring system that are available in `SafeDESIGNER`:

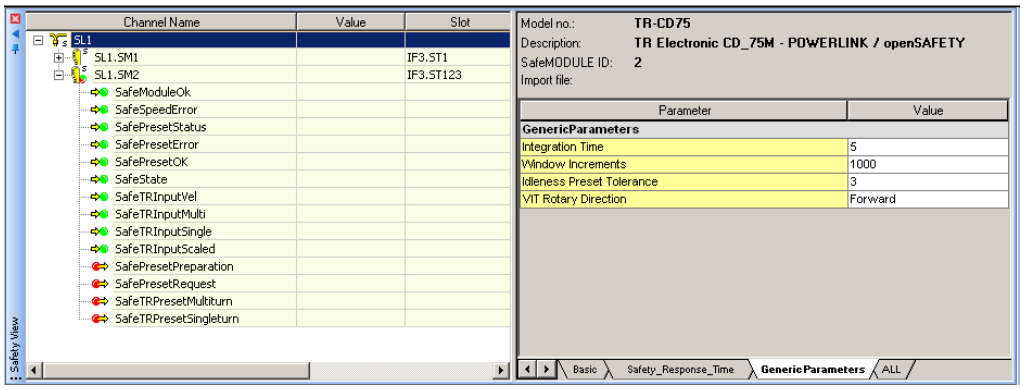


The green dots refer to input data as seen by the control while the red dots refer to output data as seen by the control.

The safe parameters of the measuring system are defined in the following chapter.

4.1.4 Safe parameterization

The parameters can be set on the user interface of `SafeDESIGNER`. The measuring-system-specific parameters are listed under `Generic Parameters`:



The following values have been assigned in the above example:

`Integration Time`:	5
`Window Increments`	1000
`Idleness Preset Tolerance`	3
`VIT Rotary Direction`	Forward

The individual values are defined as follows:

The `Integration Time` parameter is used to calculate the **safe velocity** which is output via openSAFETY. The time against which the velocity is measured can be defined within the value range from 1...10 (50 ms time base):

- Long integration time = high resolution at low speeds
- Short integration time = fast change for high speeds

The `Window Increments` parameter defines the maximum allowed position deviation in increments from the master/slave scanning system. The permissible tolerance window depends on the maximum speed and must be determined by the user. High speeds require a large tolerance window.
The larger the tolerance window, the larger the angle until an error will be detected.
The default value is 1000.

The `Idleness Preset Tolerance` parameter defines the maximum velocity for the preset step. The value range 1...5 is directly dependent on the integration time (safe). The user has to set this value subject to the application.

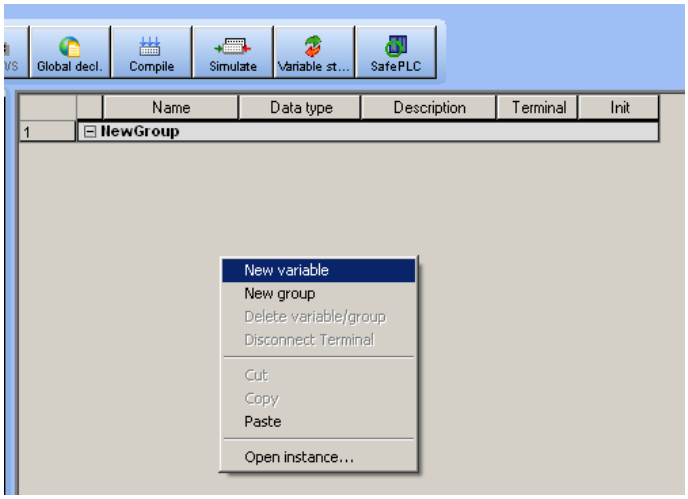
The `VIT Rotary Direction` parameter specifies whether the actual value during the approach increases (forward) or decreases (backward).

During start-up, the SCM automatically transmits the safe parameters to the measuring system (SN).

4.1.5 Creating the safety program

Safe applications running on the SCM are created in the `SafeDESIGNER` editor. The following steps show the structure of a dummy program which compares the single-turn value with the multi-turn value and sets a global variable in relation thereto. However, the variable is only set if the measuring system outputs a valid actual value. This requires that the `SafeModuleOk` and `SafeState` state variables should be set.

- The variables required must be declared. Use the toolbar or push <Ctrl>+<G> to open the `Global Declarations` screen in `SafeDESIGNER`. Right-click to open the context menu and select `New variable` to create the new variable:



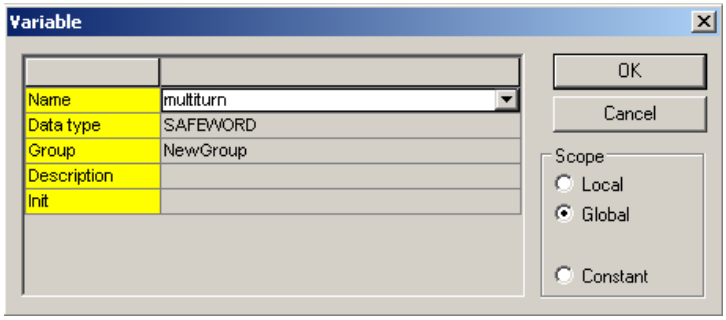
- All in all, 5 variables are created with the corresponding data types:

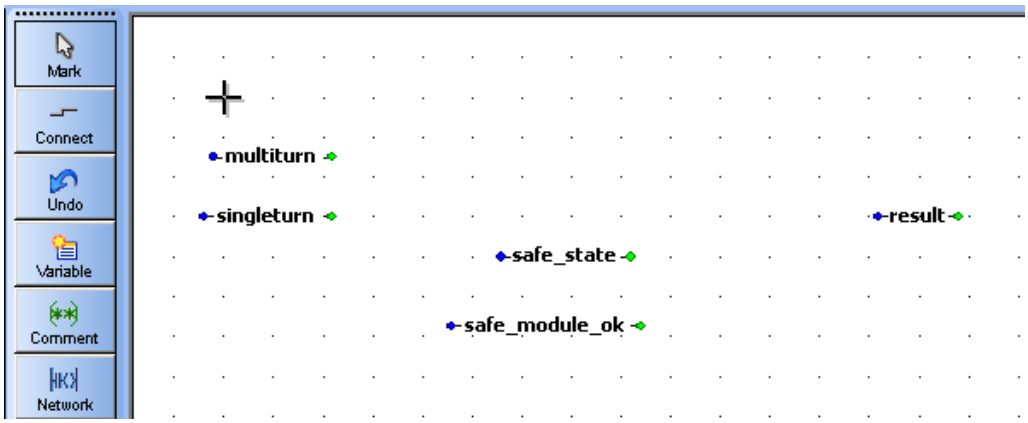
	Name	Data type	Description	Terminal	Init
1	NewGroup				
2	singleturn	SAFEWORD			
3	multiturn	SAFEWORD			
4	safe_module_ok	SAFEBOOL			
5	safe_state	SAFEBOOL			
6	result	SAFEBOOL			

- Link the variables you created to the corresponding values of the measuring system. To do this, move the corresponding variable from `Safety View` to the terminal field of that variable by drag-and-drop. This links the four `SafeModuleOk`, `SafeState`, `SafeTRInputSingle` and `SafeTRInputMulti` variables of the measuring system:

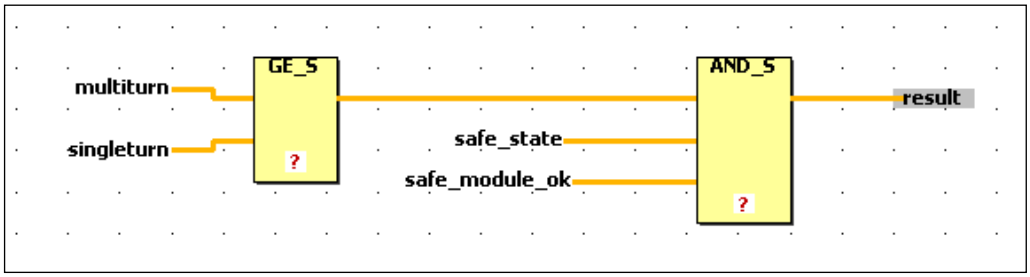
	Name	Data type	Description	Terminal	Init
1	NewGroup				
2	singleturn	SAFEWORD		SL1.SM2.SafeTRInputSingle	
3	multiturn	SAFEWORD		SL1.SM2.SafeTRInputMulti	
4	safe_module_ok	SAFEBOOL		SL1.SM2.SafeModuleOk	
5	safe_state	SAFEBOOL		SL1.SM2.SafeState	
6	result	SAFEBOOL			

➢ Return to the `Code: Main` tab where you can now program the graphics of the actual application. To do this, click the `Variable` button on the left border of the worksheet, select the variables required and copy them to the graphical interface:





➢ Link the variables to each other using the FUs and FBs.



This completes the creation of the example application. The `result` variable becomes `TRUE` as soon as the multi-turn value of the measuring system is greater than or equal to the single-turn value and the `safe_state` and `safe_module_ok` variables are set.

4.1.6 Generating the safety program

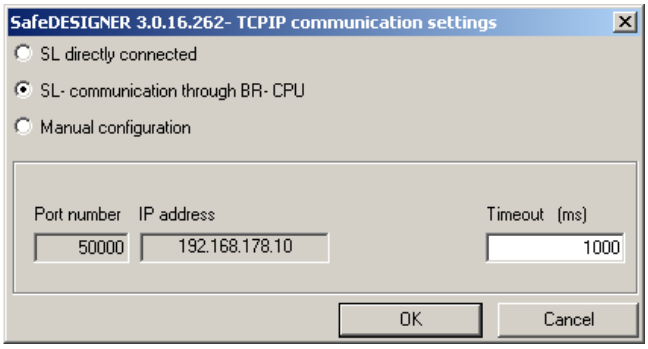
To generate the safety program, save and compile the example application you have just created. To do this, click the Compile button in SafeDESIGNER or push the <F9> key.

Since the result variable is only written and will no longer be used, two warning messages are generated. These warnings can be ignored at this point.

4.1.7 Loading the safety program

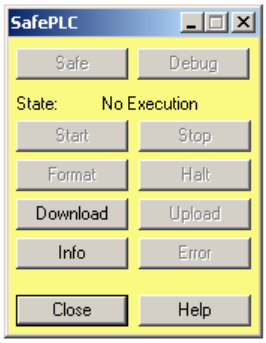
Before carrying out the steps described below, import the project from Automation Studio to the X20CP1584 (= POWERLINK project).

After having been generated, the safety program can be loaded to the SCM. This requires that a connection to the SCM is established. Go to the Online menu item in SafeDESIGNER and set the connection type. If you set up Ethernet communication with the POWERLINK CPU beforehand and the POWERLINK CPU can be reached via Ethernet, we recommend that you select the SL – communication through BR-CPU setting:

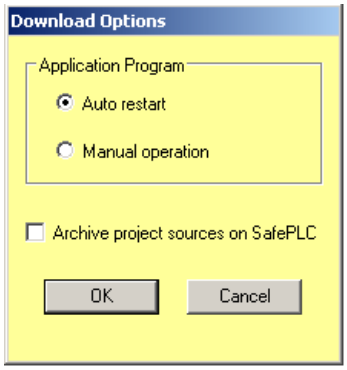


SafeDESIGNER uses the port number and current online configuration from Automation Studio as basis for online communication.

To load the software, click the SafePLC button or push <Ctrl>+<F10> and establish the connection to the SCM. Thereafter, assign or enter a password. This opens a small operating window where you can download the software.



This can only be done when the Download button is activated. If necessary, go to debug mode. If opened, click Stop to stop the current application on the SCM. This activates the Download button and you can download your own application. In the Download Options window, select Auto restart to start the application automatically:



After the download has been completed, the SafeLogic restarts. At this point, firmware and new users must be acknowledged at the SL. A flashing sequence or an appropriate LED must be set for each acknowledgment.

1. The SL is running up. This is indicated by a flashing R/E LED while the three LEDs (ENTER, MXCHG, FW_ACKN) below that LED are successively flashing from top to bottom.

2. Once run-up is completed, the new firmware must be acknowledged. This is indicated as follows: the R/E LED flashes and the FW_ACKN LED flashes 80% on and 20% off -> acknowledge by setting the coding switch to FW_ACKN (“12 hours”) and then pushing the <ENTER> key. The SL restarts once more.

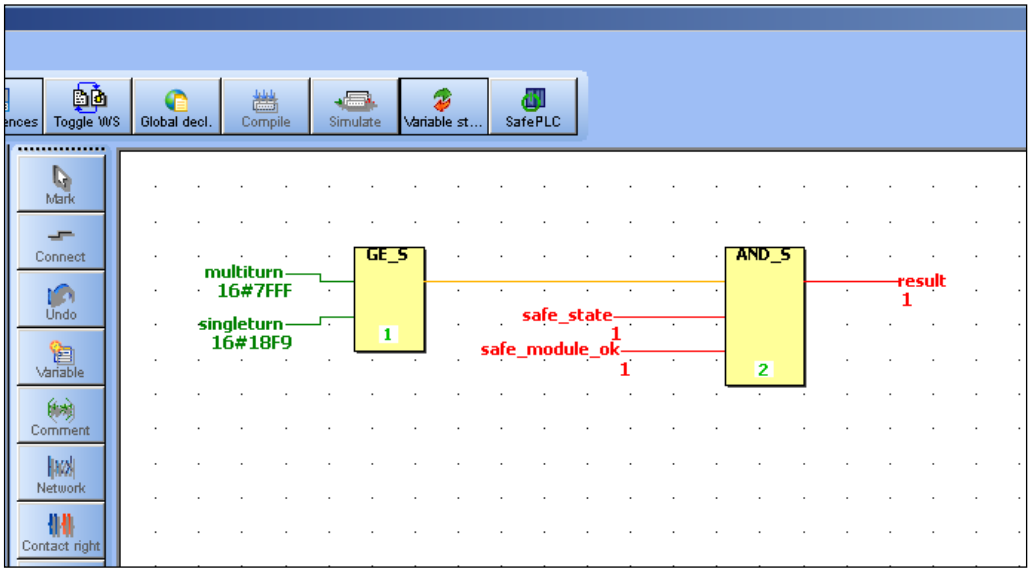
3. Once run-up is completed, the new users must be acknowledged. This is indicated by the R/E LED flashing. In addition, the MXCHG LED indicates the number of users to be acknowledged: flashing once = 1 new user; flashing twice = 2 new users; ... -> acknowledge by setting the coding switch to the number of users to be acknowledged (select “n” if this number is higher than 4) and then pushing the <ENTER> key. The SL will then run up the new user.

Please also consult the SL manual for these acknowledgments because the flashing sequences may vary depending on your course of action and on the pre-installed software.

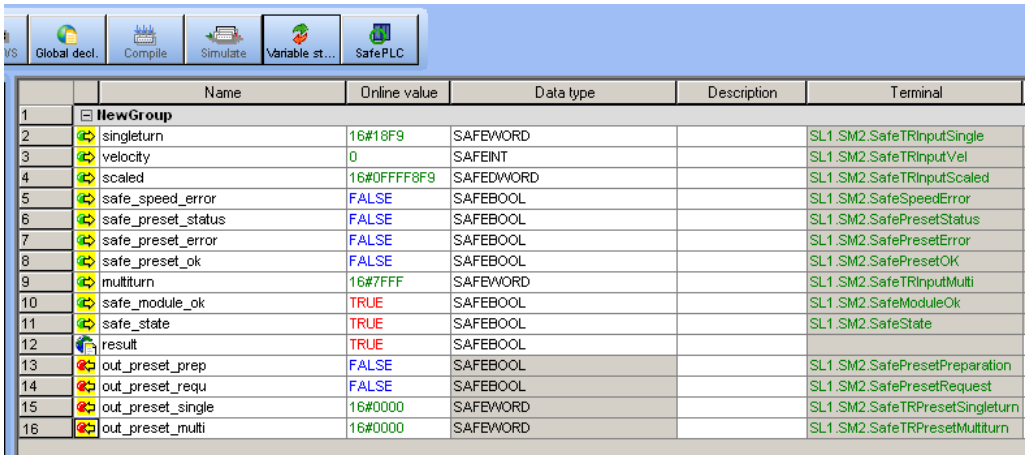
4.1.8 Testing the safety program

After having created the safety program, carry out a complete functional test of the automation task.

Click the `Variable status` button in `SafeDESIGNER` or push the <F10> key to establish the online connection which allows `SafeDESIGNER` to display the current variable values.



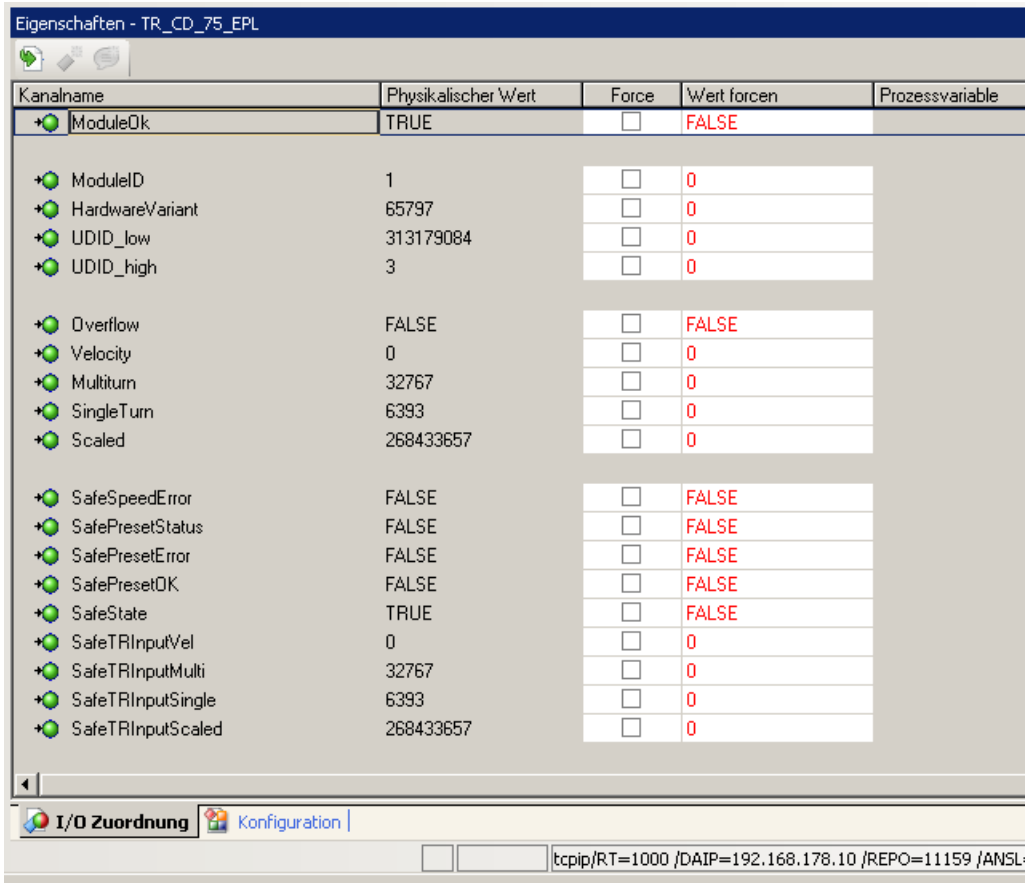
The global variables are displayed as well. The following screen shows how all remaining measuring system values are connected to variables.



	Name	Online value	Data type	Description	Terminal
1	NewGroup				
2	singleturn	16#18F9	SAFEWORD		SL1.SM2.SafeTRInputSingle
3	velocity	0	SAFEINT		SL1.SM2.SafeTRInputVel
4	scaled	16#0FFFF8F9	SAFEDWORD		SL1.SM2.SafeTRInputScaled
5	safe_speed_error	FALSE	SAFEBOOL		SL1.SM2.SafeSpeedError
6	safe_preset_status	FALSE	SAFEBOOL		SL1.SM2.SafePresetStatus
7	safe_preset_error	FALSE	SAFEBOOL		SL1.SM2.SafePresetError
8	safe_preset_ok	FALSE	SAFEBOOL		SL1.SM2.SafePresetOK
9	multiturn	16#7FFF	SAFEWORD		SL1.SM2.SafeTRInputMulti
10	safe_module_ok	TRUE	SAFEBOOL		SL1.SM2.SafeModuleOk
11	safe_state	TRUE	SAFEBOOL		SL1.SM2.SafeState
12	result	TRUE	SAFEBOOL		
13	out_preset_prep	FALSE	SAFEBOOL		SL1.SM2.SafePresetPreparation
14	out_preset_requ	FALSE	SAFEBOOL		SL1.SM2.SafePresetRequest
15	out_preset_single	16#0000	SAFEWORD		SL1.SM2.SafeTRPresetSingleturn
16	out_preset_multi	16#0000	SAFEWORD		SL1.SM2.SafeTRPresetMultiturn

The measuring system does not output any current actual values before the run-up via openSAFETY and safe parameterization have been completed. Until that point, all `Safe` variables are set to 0 and all bits of the “grey data” are set to 1.

All of the actual values of the variables are also visible in online mode of Automation Studio where they are displayed under I/O assignment of the measuring system.



In the above screen, variable names beginning with Safe refer to dual-channel data which can also be used in the “grey” range.

The “grey” single-channel values are displayed above the dual-channel data:

- Scaled
- SingleTurn
- Multiturn
- ...

4.2 XDD/OSDD device description (AS V4.5 and newer)

This chapter describes the procedure to be followed while creating an example safety program using the B&R projecting software Automation Studio (V4.5.2.102) and the optional package SafeDESIGNER (V4.3.3.7).

The safety program is created in SafeDESIGNER which features a code editor for developing the program for the safety control. The programming languages used to do this are FBD and LD.

The safety program is executed on an X20 SafeLOGIC (X20SL8100). This control is a normal POWERLINK CN participating in field bus communication and, in turn, is itself the openSAFETY SCM.

An X20 CP1584 is used as POWERLINK MN. It features a POWERLINK application which is also able to evaluate information from the measuring system (e.g., “gray channel” actual values, and others).

4.2.1 Access protection

Access to the SafeDESIGNER project is protected by a password prompt. The development and commissioning phases each have their own password. Access to the SL8100, for example for programming purposes, is also protected by a password.

The password is: abc123

4.2.2 Requirements

⚠ WARNING ***If the safety program is projected improperly, there is the risk that the fail-safe function might be deactivated!***

- The safety program may only be created based on the system documentation B&R supplies with its software and hardware.
- The descriptions below only refer to the steps required, without taking all instructions from the B&R manuals into account. It is essential to observe and comply with the information and instructions provided in the B&R manuals, particularly the safety instructions and warnings.
- The projecting procedure shown is an example only. Users are therefore obliged to check whether project planning is appropriate for their application and adjust it if necessary. This also includes selecting the appropriate safety instrumented hardware components and defining the software requirements.

Software components used for the example configuration:

- `Automation Studio` V4.5.2.102
- `SafeDESIGNER` V4.3.3.7

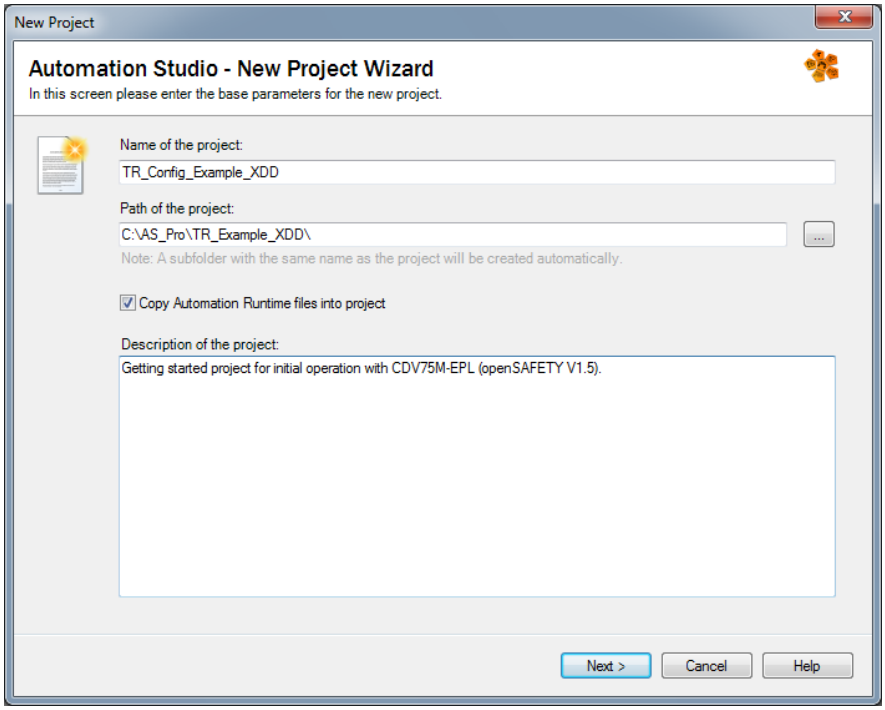
Usage of newer versions of the Automation Studio is also possible.

Hardware components used for the example configuration:

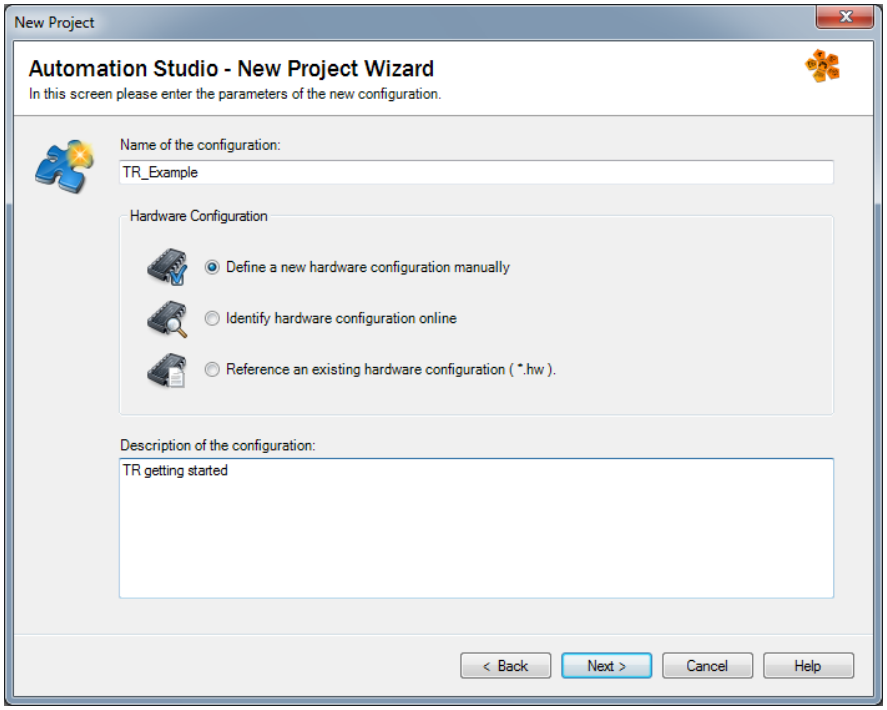
- POWERLINK-MN: `X20CP1584` mit `X20IF1082-2`
- openSAFETY SCM: `X20SL8100`

4.2.3 Hardware configuration

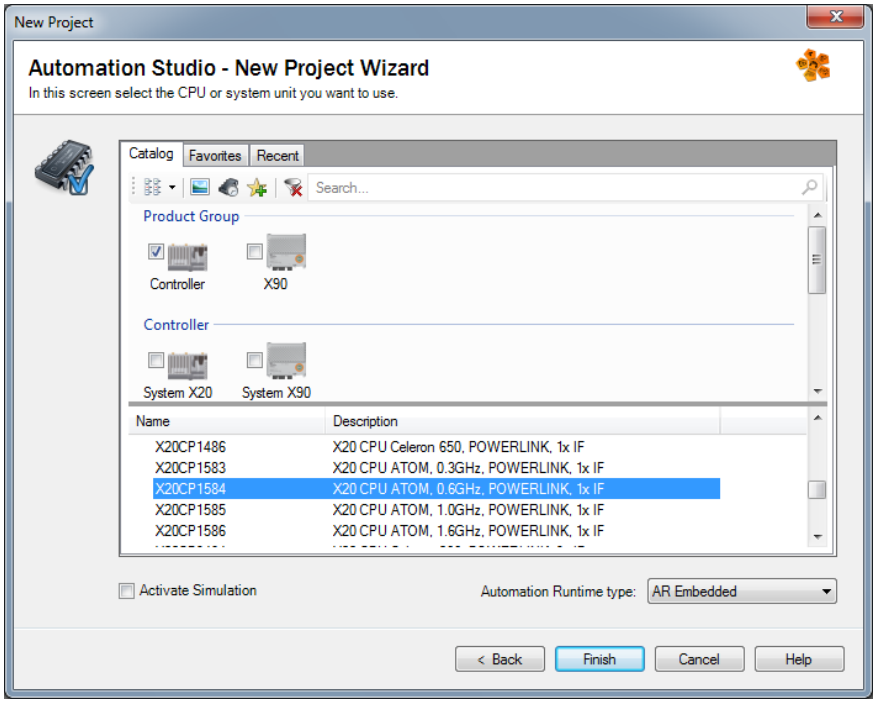
➢ Create a new project with `Automation Studio`:



➢ This creates a new hardware configuration:



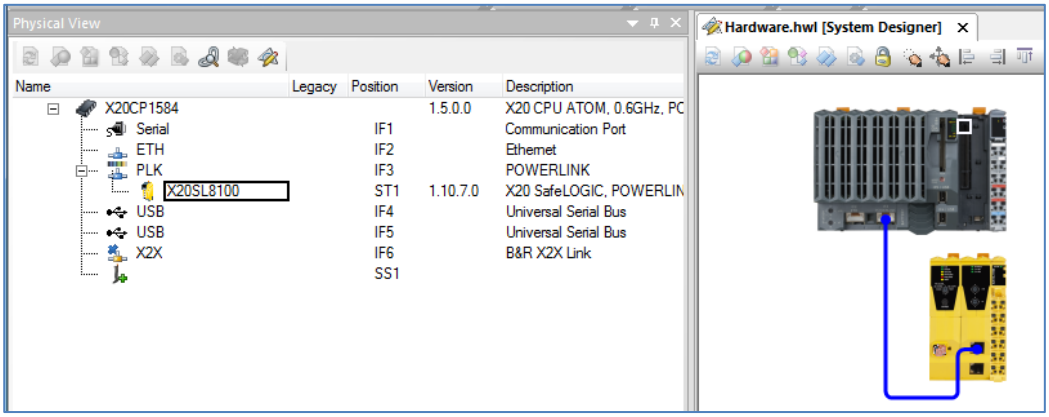
➢ Select the correct CPU in your final step: `X20CP1584`



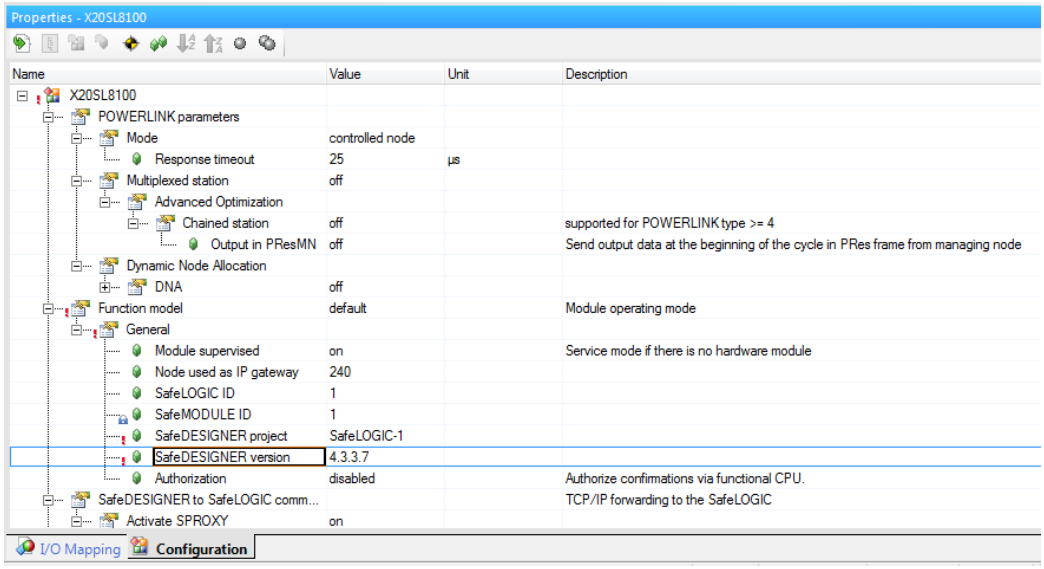
➢ Go to `-> Project -> Change Runtime Versions...` and select Safety Release 1.10. Automation Runtime version D4.52 is used in this example.

Safety Release 1.10 or newer is recommended for the encoders with openSAFETY V1.5.

➢ Add the SCM to the EPL net by dragging the corresponding device from the hardware catalog to the POWERLINK interface in `Physical View`:



➢ Select the SafeDESIGNER version from the SafeLOGIC configuration:

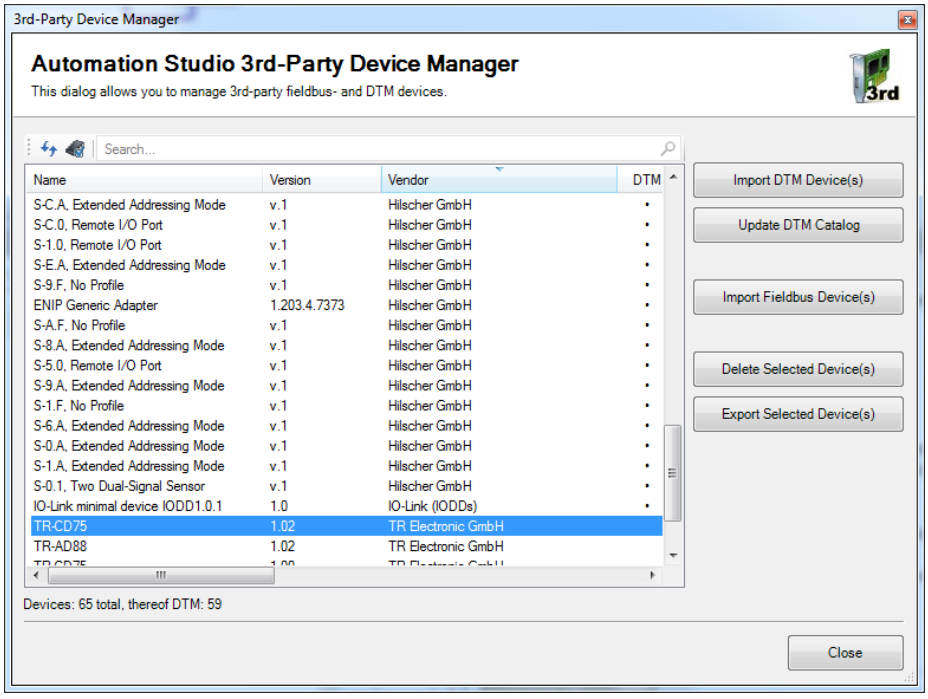


The import of a xml-based openSAFETY-device-description is necessary. The XDD-/OSDD-device descriptions are provided by TR-Electronic. It is recommended to restart the Automation Studio after installing new device descriptions.

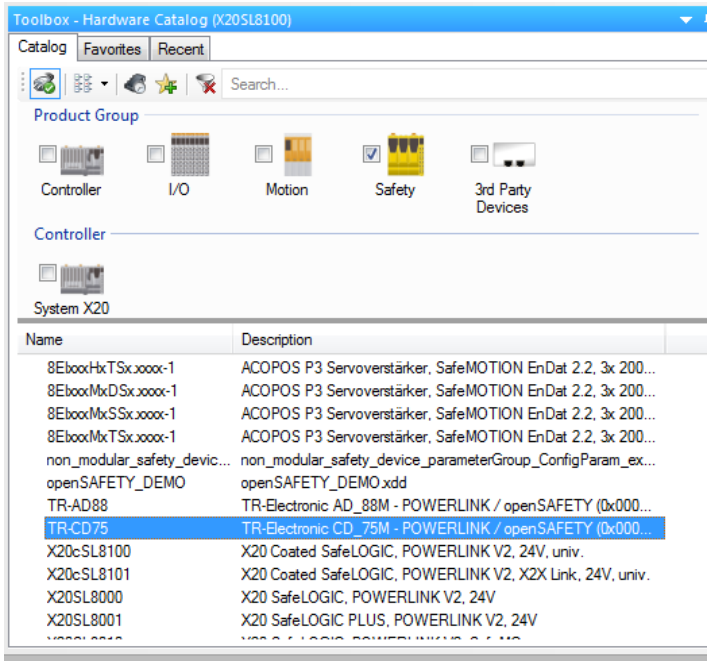
➢ Goto menu Tools -> Manage 3rd-Party Devices and hit the button Import Fieldbus Device(s).

➢ First choose the OSDD-file, after that the XDD-file

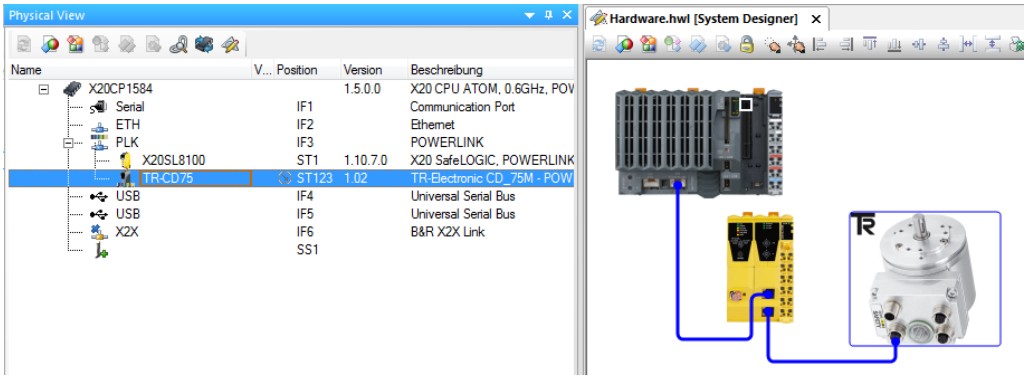
➢ The installed devices are shown in the device list:



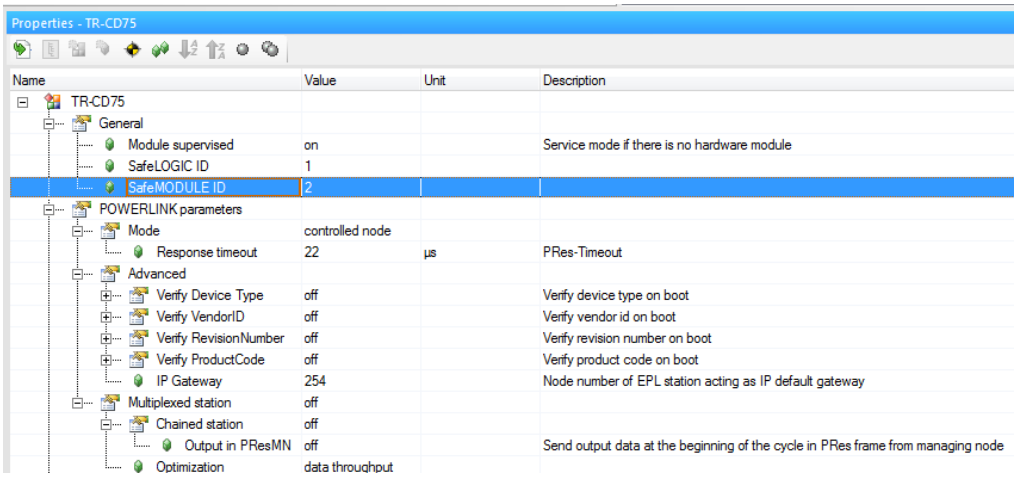
The hardware catalog will also display the devices:



After completed installation, the measuring system can also be added in `Physical View`. The correct POWERLINK node ID must be assigned. In the illustrated instance, this is node ID 123 (0x7B). This node ID must then be set for the measuring system.

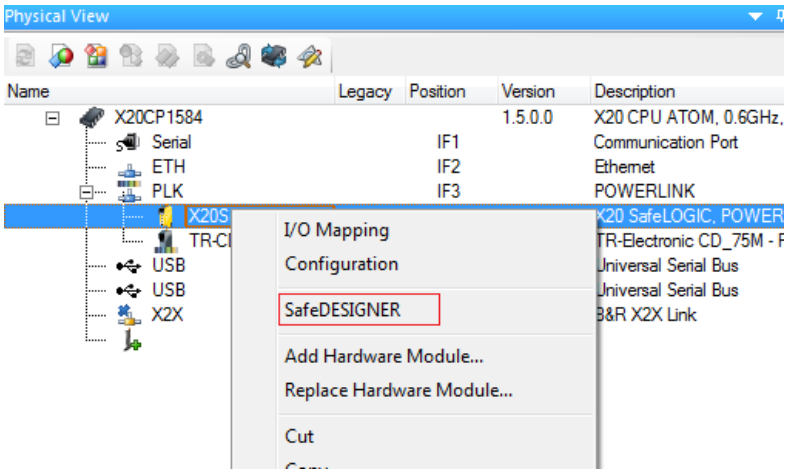


Initially, the measuring system configuration is not adjusted in `Automation Studio`. In the illustrated instance, `SafeMODULE ID` = 2 is assigned to the measuring system in the openSAFETY network.



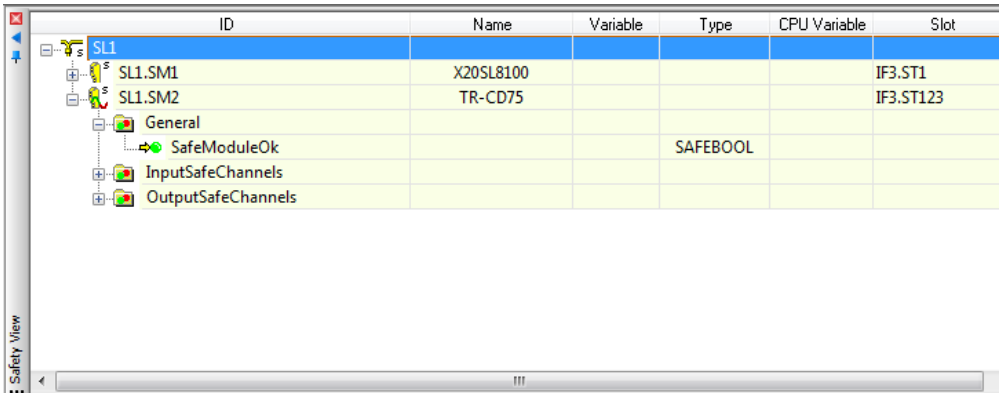
The project can now be saved and compiled with <Ctrl>+<F7>. Provided there is a connection to the EPL control, you can directly transfer the project to the X20CP1584 when prompted to do so on the next display. If you wish, you can also do this at a later point.

➢ Start the SafeDESIGNER (e.g. right-click on the SL):



➢ Assign and confirm the passwords for the SafeDESIGNER project. In the example, this is “abc123”.

In `Safety View`, the display lists the openSAFETY users at the bottom left by default. It shows the SCM (SL1.SM1) and the measuring system (SL1.SM2). The number “2” corresponds to the `SafeMODULE ID`. The `Slot` column shows the node ID of the measuring system. The list below the measuring system shows all data relating to the measuring system that are available in `SafeDESIGNER`:



These are:

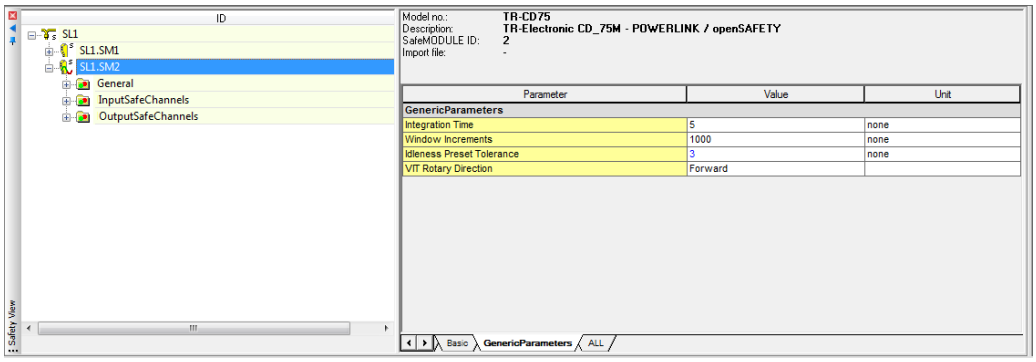
- InputSafeChannels:
  - SafeSpeedError
  - SafePresetStatus
  - SafePresetError
  - SafePresetOK
  - SafeState
  - SafeTRInputVel
  - SafeTRInputMulti
  - SafeTRInputSingle
  - SafeTRInputScaled
- OutputSafeChannels
  - SafePresetPreparation
  - SafePresetRequest
  - SafeTRPresetMultiturn
  - SafeTRPresetSingleturn

The green dots refer to input data as seen by the control while the red dots refer to output data as seen by the control.

The safe parameters of the measuring system are defined in the following chapter.

4.2.4 Safe parameterization

The parameters can be set on the user interface of SafeDESIGNER. The measuring-system-specific parameters are listed under Generic Parameters:



The following values have been assigned in the above example:

Integration Time:	5
Window Increments	1000
Idleness Preset Tolerance	3
VIT Rotary Direction	Forward

The individual values are defined as follows:

The Integration Time parameter is used to calculate the **safe velocity** which is output via openSAFETY. The time against which the velocity is measured can be defined within the value range from 1...10 (50 ms time base):

- Long integration time = high resolution at low speeds
- Short integration time = fast change for high speeds

The Window Increments parameter defines the maximum allowed position deviation in increments from the master/slave scanning system. The permissible tolerance window depends on the maximum speed and must be determined by the user. High speeds require a large tolerance window.
The larger the tolerance window, the larger the angle until an error will be detected.
The default value is 1000.

The Idleness Preset Tolerance parameter defines the maximum velocity for the preset step. The value range 1...5 is directly dependent on the integration time (safe). The user has to set this value subject to the application.

The VIT Rotary Direction parameter specifies whether the actual value during the approach increases (forward) or decreases (backward).

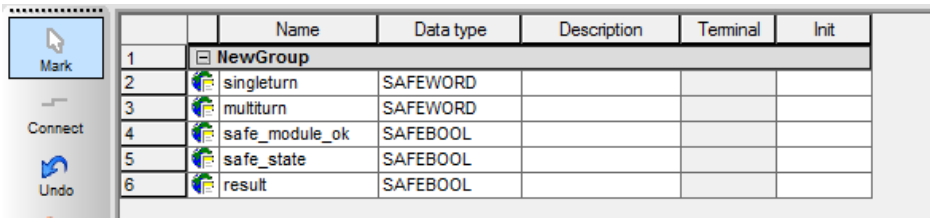
During start-up, the SCM automatically transmits the safe parameters to the measuring system (SN).

4.2.5 Creating the safety program

Safe applications running on the SCM are created in the SafeDESIGNER editor. The following steps show the structure of a dummy program which compares the single-turn value with the multi-turn value and sets a global variable in relation thereto. However, the variable is only set if the measuring system outputs a valid actual value. This requires that the SafeModuleOk and SafeState state variables should be set.

- The variables required must be declared. Use the toolbar or push <Ctrl>+<G> to open the Global Declarations screen in SafeDESIGNER. Right-click to open the context menu and select New variable to create the new variable:

- 5 variables are created:

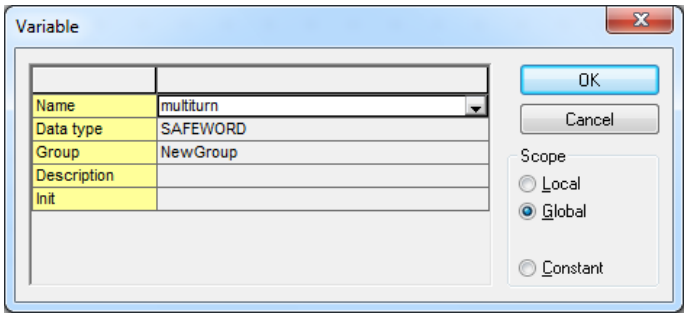


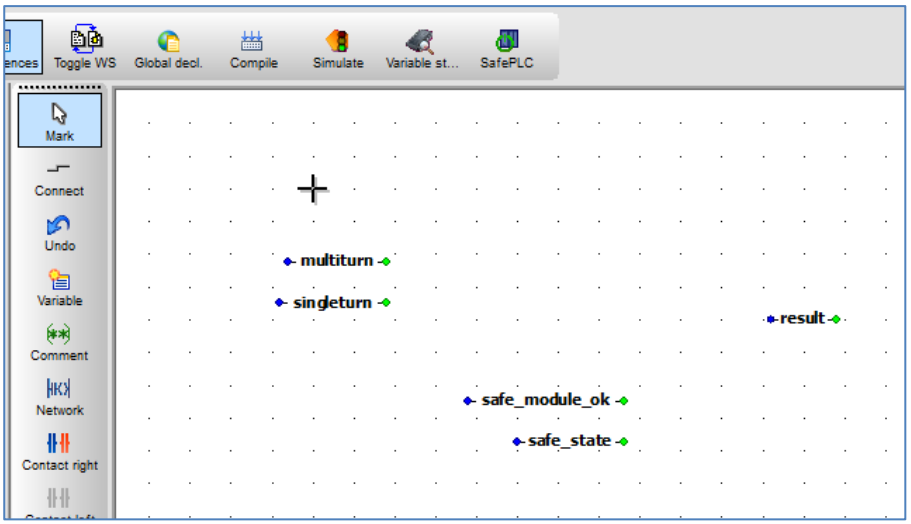
		Name	Data type	Description	Terminal	Init
1		NewGroup				
2		singleturn	SAFEWORD			
3		multiturn	SAFEWORD			
4		safe_module_ok	SAFEBOOL			
5		safe_state	SAFEBOOL			
6		result	SAFEBOOL			

- Link the variables you created to the corresponding values of the measuring system. To do this, move the corresponding variable from Safety View to the terminal field of that variable by drag-and-drop. This links the four SafeModuleOk, SafeState, SafeTRInputSingle and SafeTRInputMulti variables of the measuring system:

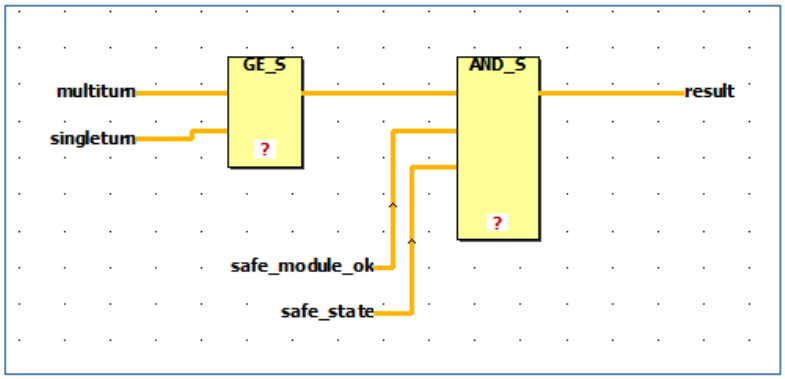
		Name	Data type	Description	Terminal	Init
1		NewGroup				
2		singleturn	SAFEWORD		SL1.SM2.SafeTRInputSingle	
3		multiturn	SAFEWORD		SL1.SM2.SafeTRInputMulti	
4		safe_module_ok	SAFEBOOL		SL1.SM2.SafeModuleOk	
5		safe_state	SAFEBOOL		SL1.SM2.SafeState	
6		result	SAFEBOOL			

➢ Return to the `Code: Main` tab where you can now program the graphics of the actual application. To do this, click the `Variable` button on the left border of the worksheet, select the variables required and copy them to the graphical interface:





➢ Link the variables to each other using the FUs and FBs:



This completes the creation of the example application. The `result` variable becomes `TRUE` as soon as the multi-turn value of the measuring system is greater than or equal to the single-turn value and the `safe_state` and `safe_module_ok` variables are set.

4.2.6 Generating the safety program

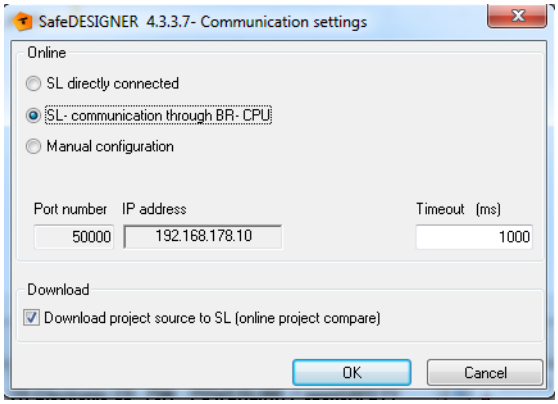
To generate the safety program, save and compile the example application you have just created. To do this, click the `Compile` button in `SafeDESIGNER` or push the <F9> key.

Since the `result` variable is only written and will no longer be used, two warning messages are generated. These warnings can be ignored at this point.

4.2.7 Loading the safety program

Before carrying out the steps described below, import the project from `Automation Studio` to the `X20CP1584` (= POWERLINK project).

After having been generated, the safety program can be loaded to the SCM. This requires that a connection to the SCM is established. Go to the `Online` menu item in `SafeDESIGNER` and set the connection type. If you set up Ethernet communication with the POWERLINK CPU beforehand and the POWERLINK CPU can be reached via Ethernet, we recommend that you select the `SL – communication through BR-CPU` setting:



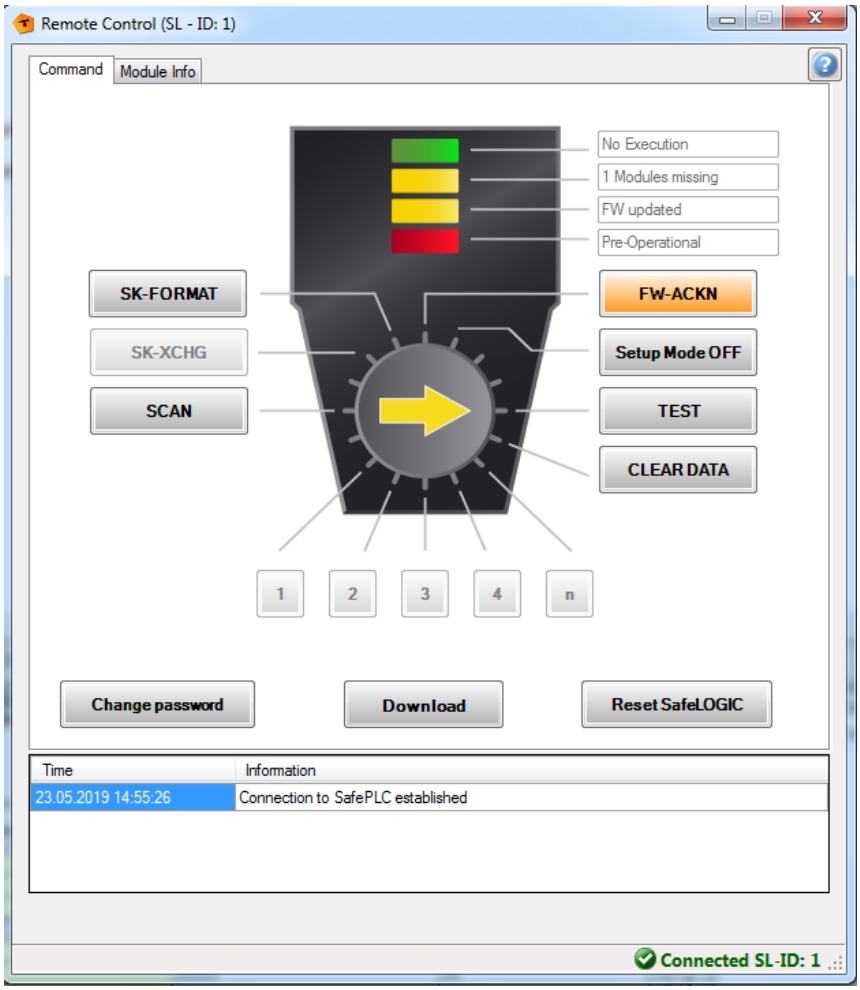
`SafeDESIGNER` uses the port number and current online configuration from `Automation Studio` as basis for online communication.

To load the software, click the `SafePLC` button or push <Ctrl>+<F10> and establish the connection to the SCM. Thereafter, assign or enter a password. This opens a small operating window where you can download the software.

This can only be done when the Download button is activated. If necessary, go to debug mode. If opened, click Stop to stop the current application on the SCM. This activates the Download button and you can download your own application. In the Download Options window, select Auto restart to start the application automatically.

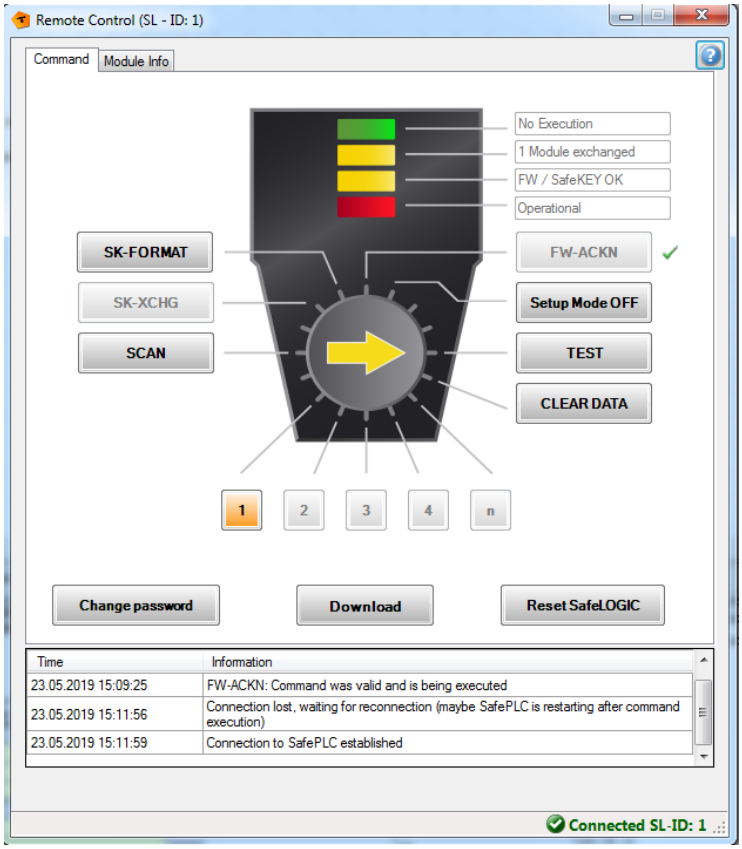
After the download has been completed, the SafeLogic restarts. At this point, firmware and new users must be acknowledged at the SL. A flashing sequence or an appropriate LED must be set for each acknowledgment.

The remote control can be used for the acknowledgements. It is started with a right-click on the SL1.SM1 in the safety view:



If the firmware needs to be acknowledged the button is displayed colored (here: FW-ACKN).

If a new user (e.g. the TR-encoder) has to be acknowledged one of the buttons 1, 2, 3, 4 or n is displayed colored (here: 1 for one new user):



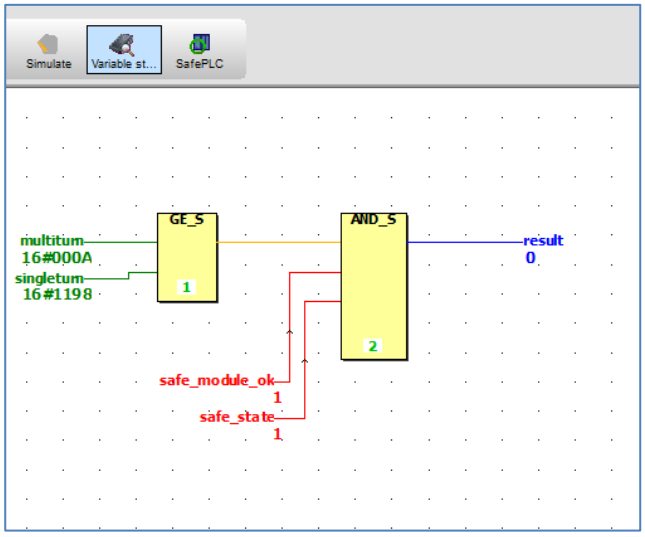
After acknowledgement of the new user the encoder is running „operational“. Both status-LEDs indicate green light which means: EPL operational and openSAFETY operational.

Please also consult the SL manual for these acknowledgments because the flashing sequences may vary depending on your course of action and on the pre-installed software.

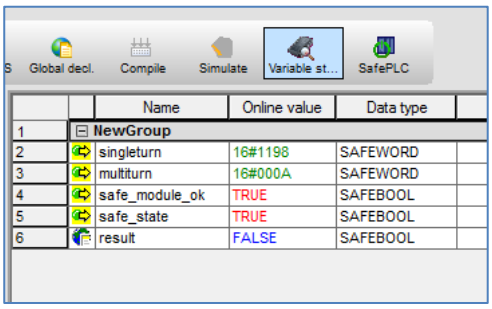
4.2.8 Testing the safety program

After having created the safety program, carry out a complete functional test of the automation task.

Click the `Variable status` button in `SafeDESIGNER` or push the <F10> key to establish the online connection which allows `SafeDESIGNER` to display the current variable values.



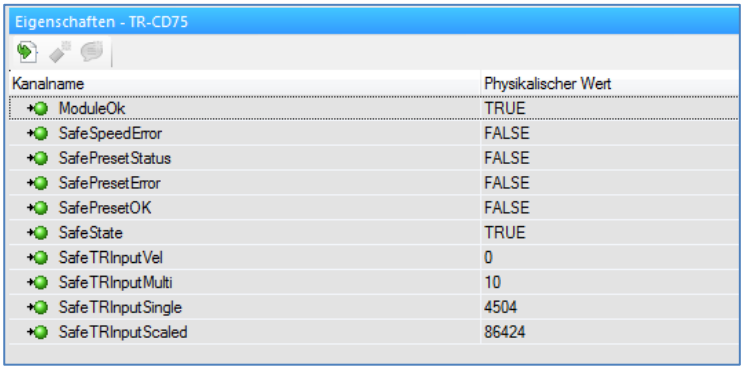
The global variables are displayed as well:



	Name	Online value	Data type
1	NewGroup		
2	singleturn	16#1198	SAFEWORD
3	multiturn	16#000A	SAFEWORD
4	safe_module_ok	TRUE	SAFEBOOL
5	safe_state	TRUE	SAFEBOOL
6	result	FALSE	SAFEBOOL

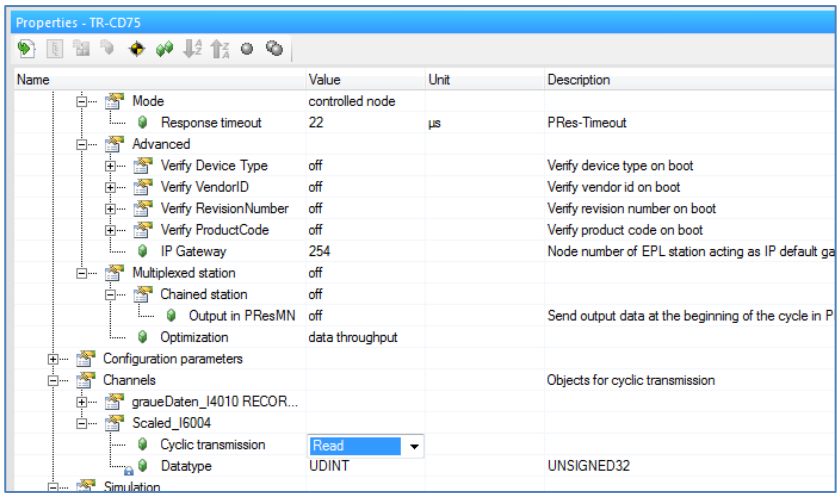
The measuring system does not output any current actual values before the run-up via openSAFETY and safe parameterization have been completed. Until that point, all `Safe` variables are set to 0 and all bits of the “grey data” are set to 1.

All of the actual values of the variables are also visible in online mode of `Automation Studio` where they are displayed under I/O assignment of the measuring system:

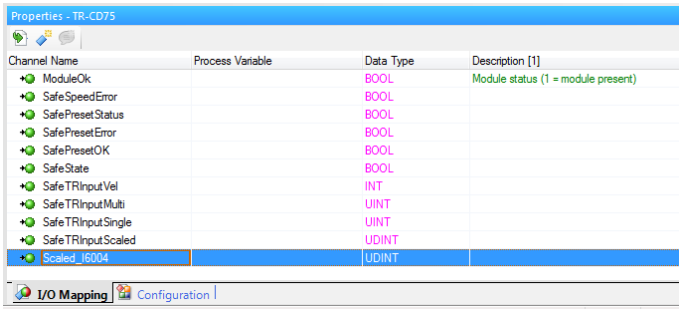


In the above screen, variable names beginning with `Safe` refer to dual-channel data which can also be used in the “grey” range.

The „grey“ values can also be mapped and used. Therefore the cyclic transmission has to be activated. In the following example the 32Bit grey position value is activated.



The new value is also shown in the I/O-mapping of the encoder:



The following grey values can also be mapped:

- Overflow
- Velocity
- Multiturn
- Singleturn

5 Expanding the safety program - application examples

The safety program created in Chapter 4 is expanded in the following sections by function examples for the use of actual values of the measuring system in `SafeDesigner`.

However, these examples are not customer-specific solutions and only intended to provide assistance in various automation tasks .

The integration of the measurement system in an application is intended to be simplified by using the featured function modules.

In the following application examples
- Safely Limited Speed (SLS)
- Safe standstill detection
- Safe position detection
- Safe direction detection

the featured function modules issue the fault states. The associated error handling is not part of the examples and must be implemented by the user.

The function module in the application example for Preset implementation does not direct output a fault state because the error information is generated by the measuring system itself.

Please note the “Terms of use of the software examples” on page 60!

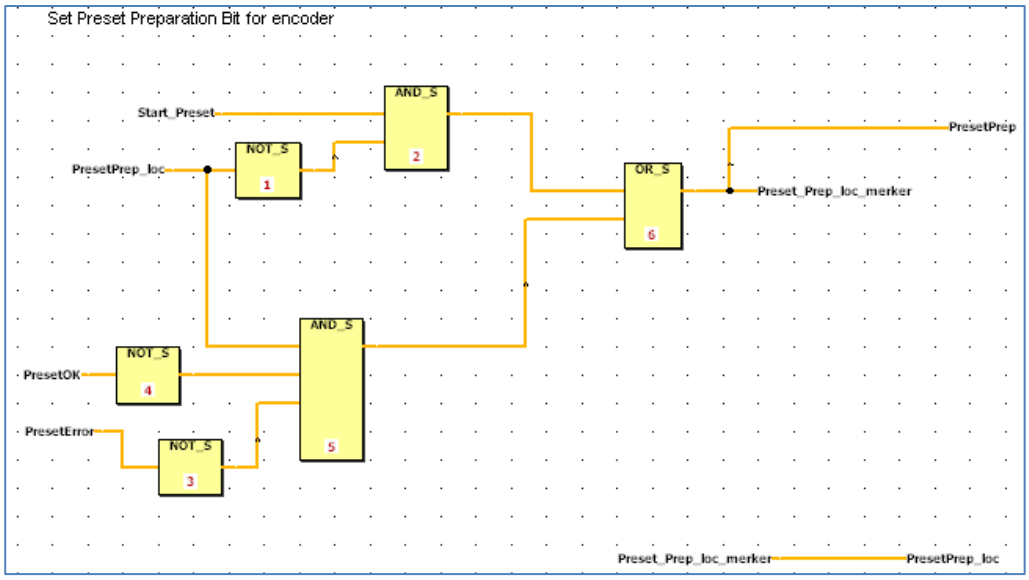
5.1 Preset implementation

At first, create a function block in SafeDESIGNER (New Function Block after having right-clicked the Logical POUs in the Project Tree Window). Assign the name as TR_PRESET_Example. Then create the following local variables for the function block:

	Name	Data type	Usage	Description	Init	Diag	Feedback
1	⊟ NewGroup						
2	Start_Preset	SAFEBOOL	VAR_INPUT	Preset Start	SAFEBOOL#0		☐
3	PresetReq	SAFEBOOL	VAR_OUTPUT	Set Preset Request	SAFEBOOL#0	☐	☐
4	PresetPrep_loc	SAFEBOOL	VAR	Set Preset Prep local	SAFEBOOL#0		☑
5	PresetOK	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		☐
6	PresetError	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		☐
7	PresetFinish	SAFEBOOL	VAR_OUTPUT	Preset finished	SAFEBOOL#0	☐	☐
8	PresetFinish_loc	SAFEBOOL	VAR		SAFEBOOL#0		☑
9	PresetFinish_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		☐
10	PresetPrep	SAFEBOOL	VAR_OUTPUT	Set Preset Preparation	SAFEBOOL#0	☐	☐
11	Preset_Prep_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		☐
12	PresetReq_loc	SAFEBOOL	VAR		SAFEBOOL#0		☑
13	Reset_Lock	SAFEBOOL	VAR_INPUT	Lock Reset	SAFEBOOL#0		☐
14	PresetReq_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		☐

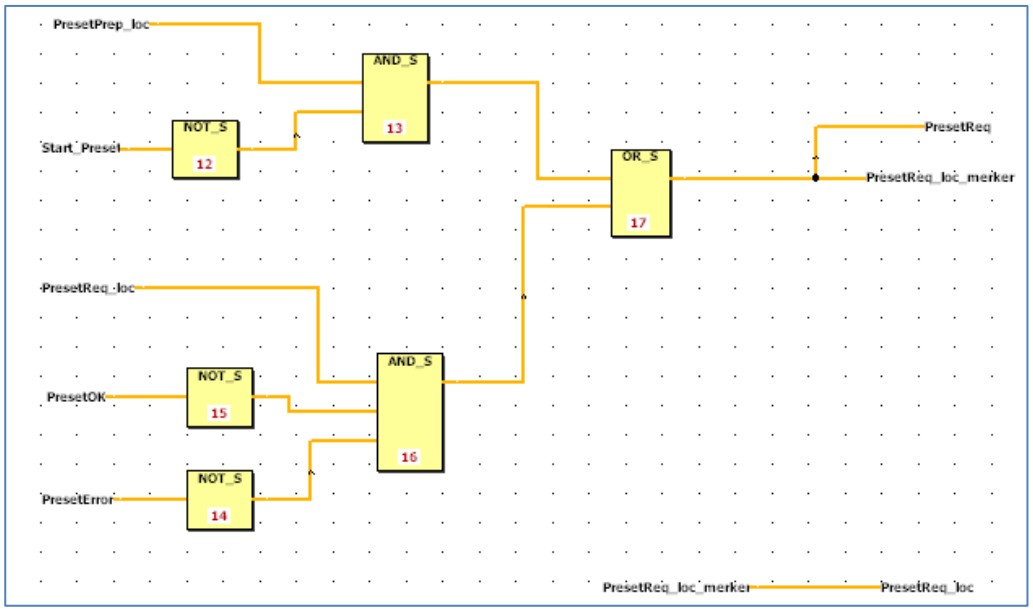
The following pictures show the structure of the new TR_PRESET_Example function block:

1. Set Preset Preparation



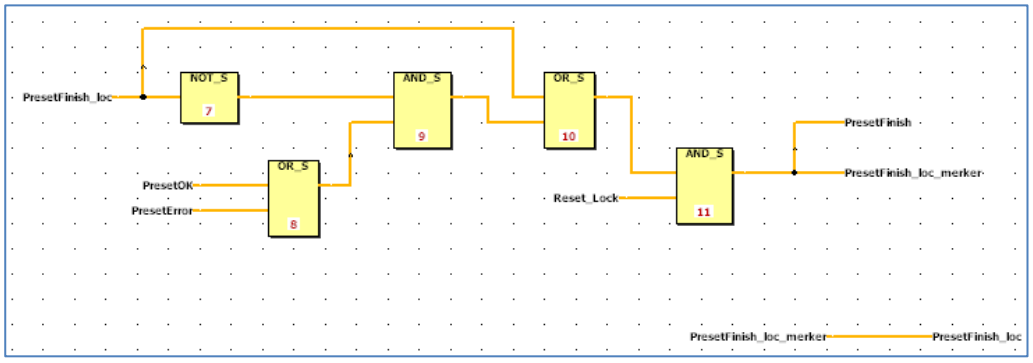
Setting the Start_Preset input in the first section sets the measuring system output signal PresetPrep bit. The signal remains set until the measuring system completed the preset process. The measuring system sets PresetOK if the preset process could be executed without error and PresetError in case of failure.

2. Set `Preset Request`



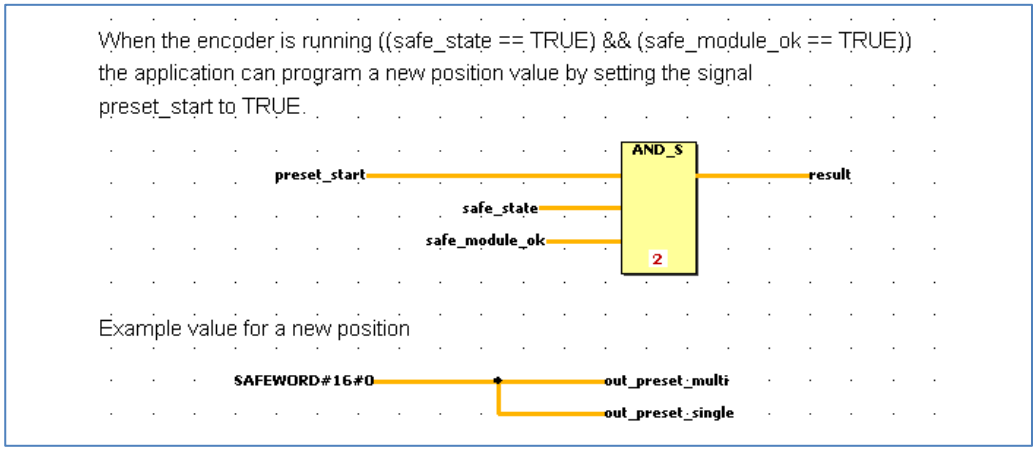
If the output signal `PresetPrep` is set for the measuring system and the `Start_Preset` signal is again set to `FALSE`, the `PresetReq` signal is set. The signal remains set until the measuring system completed the preset process. The measuring system sets `PresetOK` if the preset process could be executed without error and `PresetError` in case of failure.

3. Set `Preset Finish`

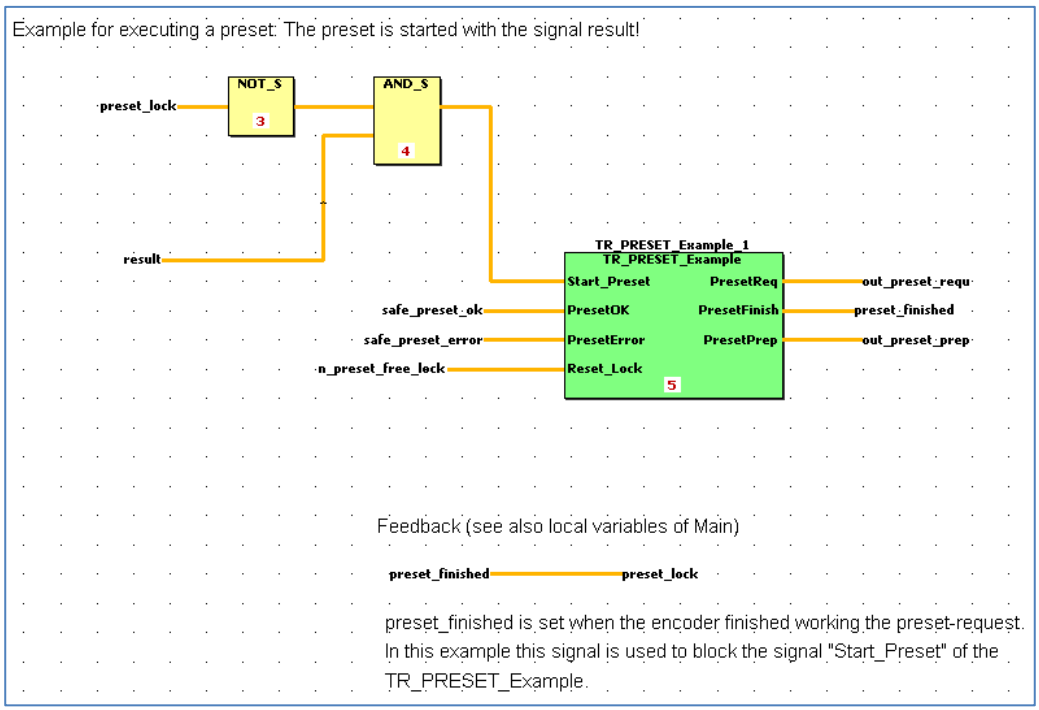


This section saves the information that the implementation has been completed. `Reset_Lock` must always be set to “1”, as the variable is low-active.

A new signal named `preset_start` is created to start preset, which could be connected with a digital input, for example. Now, the main function block is complemented such that a fixed value of 0 is the new specification which is transmitted to the measuring system.



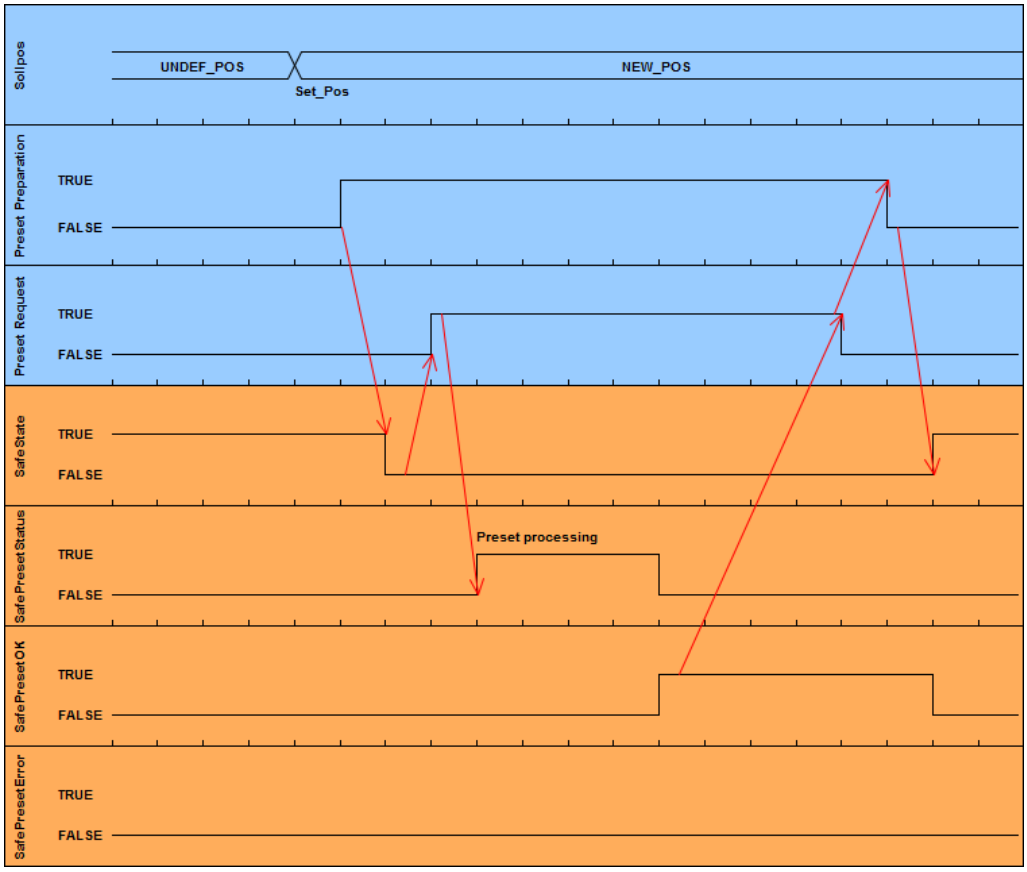
The `PresetPrep` and `PresetRequ` outputs to the measuring system are connected to the corresponding outputs of the `TR_PRESET_Example` function block. The preset step is carried out only once for each `Start_Preset` signal using a `Start_Preset` signal, If the `preset_lock` signal has not been set yet and the `result` signal is being set. The preset function in this program is executed after a run-up to the new desired value of 0, once the user sets the `preset_start` signal to `TRUE`.



The signal `n_preset_free_lock` must first be re-set to `FALSE` (starting value = `TRUE`) to run another Preset in the present example. This causes the output signal `preset_finished` to be reset. For another preset, `preset_start` and thus `result` can again be set to `TRUE`.

The following timing diagram shows a faultless execution of the preset function.

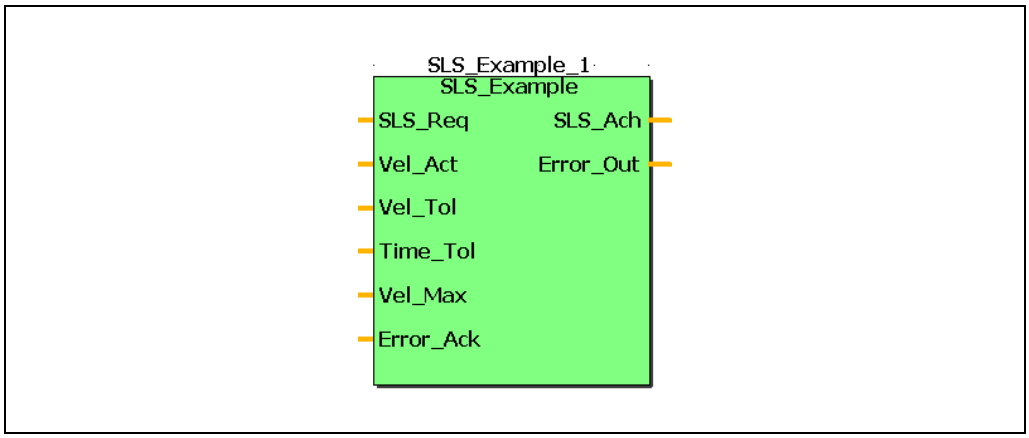
blue area: Output signal control -> measuring system
orange area: Input signals measuring system -> control system



5.2 Safely Limited Speed (SLS)

The function module `SLS_Example` provides monitoring to Safely Limited Speed (SLS) on demand. The value of the measuring system speed `Vel_Act` must be, according to the requirement `SLS_Req`, within the parameterized tolerance time `Time_Tol` below the parameterized safely limited speed `Vel_Tol`. The output `SLS_Ach` is set if this is the case. The output `ERROR_OUT` is set if an error occurs.

An error can be acknowledged with `Error_Ack`.

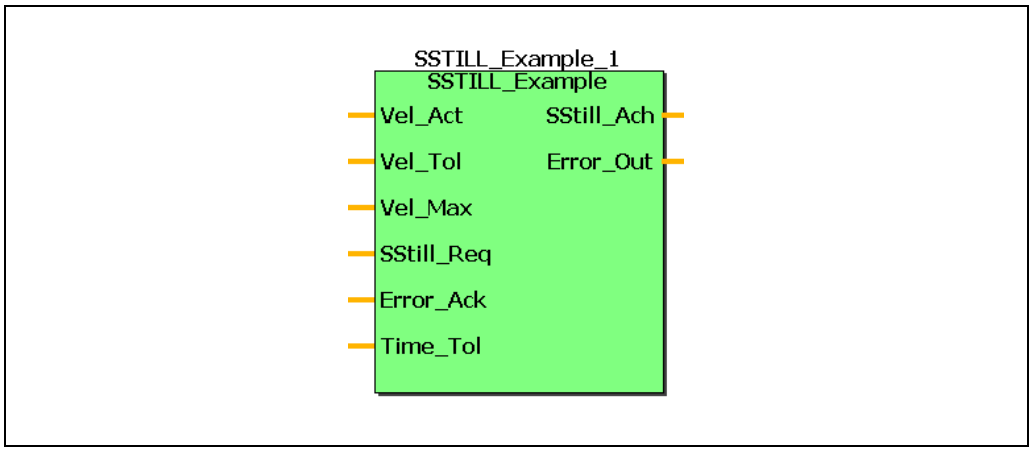


Signal	Type	Direction	Description
`SLS_Req`	SAFEBOOL	Input	Requesting the safely limited speed with `SLS_Req` = `TRUE`
`Vel_Act`	SAFEINT	Input	Safely limited speed of the measuring system
`Vel_Tol`	SAFEINT	Input	Limit for safely limited speed (SLS)
`Time_Tol`	SAFETIME	Input	Maximum time from request to SLS
`Vel_Max`	SAFEBOOL	Input	Measuring system speed overflow
`Error_Ack`	SAFEBOOL	Input	Error acknowledgment
`SLS_Ach`	SAFEBOOL	Output	SLS reached
`Error_Out`	SAFEBOOL	Output	Error exists, e.g. timeout

5.3 Safe standstill detection

The function module `SSTILL_Example` provides monitoring to a safe standstill. Once the value of the current measuring system speed `Vel_Act` is smaller than the parameterized maximum speed `Vel_Tol`, the output `SStill_Ach` is set. After the request of a standstill `SStill_Req`, `Vel_Act` must be within the parameterized tolerance time `Time_Tol` below the parameterized maximum speed `Vel_Tol` for a standstill. The output `Error_Out` is set if this is not the case.

An error can be acknowledged with `Error_Ack`.

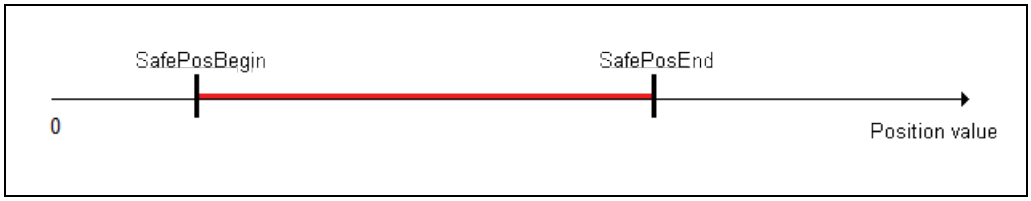


Signal	Type	Direction	Description
`Vel_Act`	SAFEINT	Input	Safely limited speed of the measuring system
`Vel_Tol`	SAFEINT	Input	Limit for standstill (tolerance)
`Vel_Max`	SAFEBOOL	Input	Measuring system speed overflow
`SStill_Req`	SAFEBOOL	Input	Request of a safe standstill with `SStill_Req` = `TRUE`
`Error_Ack`	SAFEBOOL	Input	Error acknowledgment
`Time_Tol`	SAFETIME	Input	Maximum time from request to standstill
`SStill_Ach`	SAFEBOOL	Output	`TRUE`: Standstill
`Error_Out`	SAFEBOOL	Output	Error exists, e.g. timeout

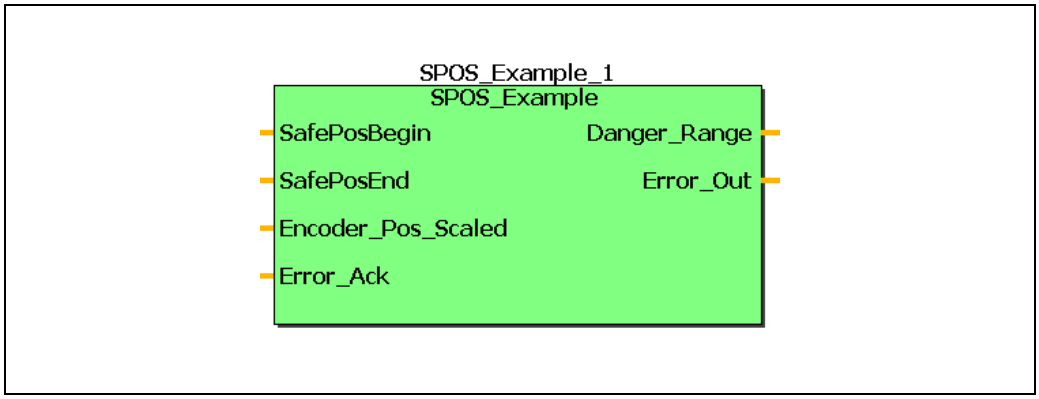
5.4 Safe position detection

A danger zone can be specified with the function module `SPOS_Example`. An actual value range can be defined using `SafePosBegin` and `Safe PosEnd`, within which the output `Danger_Range` is set. The area must always be selected that `SafePosBegin` < `Safe PosEnd`. An overflow is not permitted and results in an error.

An error can be acknowledged with `Error_Ack`.



The output `Danger_Range` is set in the red zone.



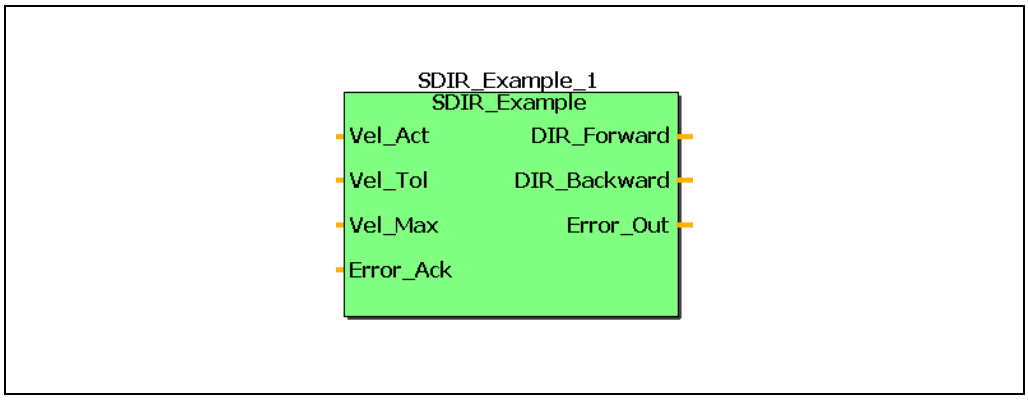
Signal	Type	Direction	Description
`SafePosBegin`	SAFEDWORD	Input	Start of dangerous area
`SafePosEnd`	SAFEDWORD	Input	End of dangerous area
`Encoder_Pos_Scaled`	SAFEDWORD	Input	Current actual value of the measuring system
`Error_Ack`	SAFEBOOL	Input	Error acknowledgment
`Danger_Range`	SAFEBOOL	Output	`TRUE`: The actual value of the measuring system is set within the dangerous area: `SafePosBegin` ≤ `Encoder_Pos_Scaled` ≤ `SafePosEnd`
`Error_Out`	SAFEBOOL	Output	Error exists, e.g. timeout

5.5 Safe direction detection

The counting direction of the measuring system can be determined with the function module `SDIR_Example`:
- positive direction = ascending actual values
- negative direction = falling actual values

A speed can be defined, from which the direction detection becomes active, to avoid an uncontrolled output caused by slight movements at standstill. Below the speed `Vel_Tol`, the outputs `DIR_Forward` and `DIR_Backward` are always `FALSE`.

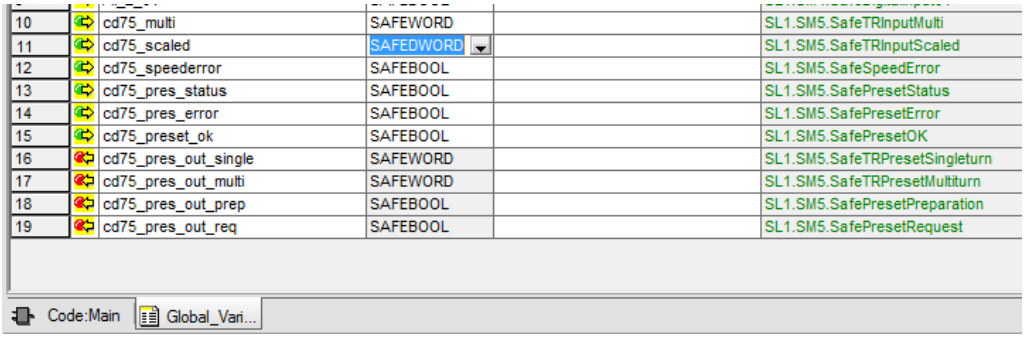
An error can be acknowledged with `Error_Ack`.



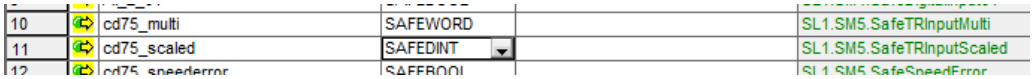
Signal	Type	Direction	Description
`Vel_Act`	SAFEINT	Input	Safely limited speed of the measuring system
`Vel_Tol`	SAFEINT	Input	Limit for direction measurement
`Vel_Max`	SAFEBOOL	Input	Measuring system speed overflow
`Error_Ack`	SAFEBOOL	Input	Error acknowledgment
`DIR_Forward`	SAFEBOOL	Output	`TRUE`: Actual value of the measuring system increasing
`DIR_Backward`	SAFEBOOL	Output	`TRUE`: Actual value of the measuring system falling
`Error_Out`	SAFEBOOL	Output	Error exists, e.g. speed overflow

5.6 Typecast of input values in the SafeDESIGNER

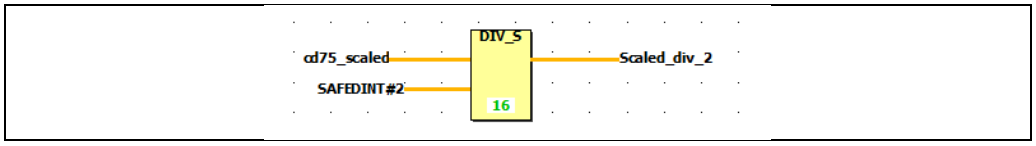
The input values for singleturn, multiturn und scaled position are unsigned integers. Due to the fact that some functions in the SafeDESIGNER cannot handle unsigned values (e.g. division) it can be necessary to treat the unsigned values as signed values.



In the image above it is shown that the type of the actual position value (scaled) is SAFEDWORD. This is an unsigned 32-Bit value. To use this value for a normal division the type can be set to SAFEDINT:



Now the value can be used as SAFEDINT in the SafeDESIGNER and a division is possible:



The user has to keep in mind, that the value never becomes negative. This is because the highest bit in the position value is always 0:

singleturn: 0, 1, 2,...8189, 8190, 8191 → overflow to 0, 1, 2....
multiturn: 0, 1, 2,...32766, 32767 → overflow to 0, 1, 2....
scaled_pos: 0, 1, 2,... 268435455 → overflow to 0, 1, 2....

The actual values are always "positive".

Risk of injury and damage to property by an actual value jump between the value 0 and the maximum value!

⚠ WARNING

NOTICE

- The user has to keep in mind that the value changes from 0 to the maximum value if the shaft is turning backwards. This means if the position is 0 and the position is decreasing, the next value is not -1 but the maximum value (e.g. 8191 for singleturn). This behaviour is independant of the value interpretation (SAFEDWORD or SAFEDINT) in the SafeDESIGNER.

6 Download - software examples

Legacy example for Automation Studio versions <V4.5:

www.tr-electronic.de/f/zip/TR-ECE-TI-MUL-0268

Example with XDD-/OSDD device description as from Automation Studio versions V4.5:

www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0351