

ETHERNET   
**POWERLINK**  
certified product

open   
**SAFETY**  
certified product

D

Seite 2 - 36

GB

Page 37 - 72

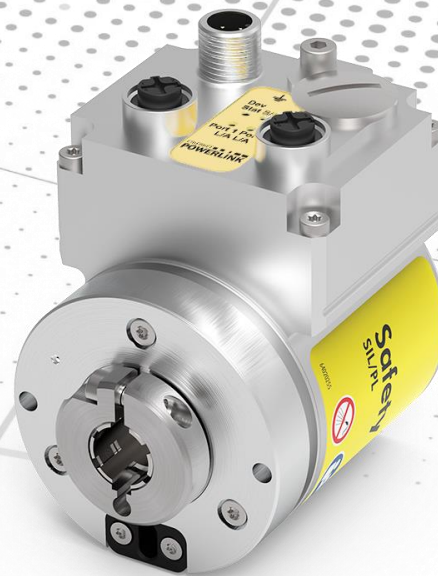
# Absolut Encoder CD\_582-EPL POWERLINK/openSAFETY

Parametrierung mit B&R Steuerungssystem  
X20 CPU

**CDV582**



**CDH582**



Abbildungen ähnlich

**DIN EN 61508:** SIL CL2 / SIL CL3  
**DIN EN ISO 13849:** PL d / PL e

- \_ Sicherheitsprogramm
- \_ Konfiguration
- \_ Zugriff auf Datenkanäle
- \_ Parametrierung

**Technische  
Information**

### **TR-Electronic GmbH**

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

---

#### **Urheberrechtsschutz**

Dieses Handbuch, einschließlich den darin enthaltenen Abbildungen, ist urheberrechtlich geschützt. Drittenwendungen dieses Handbuchs, welche von den urheberrechtlichen Bestimmungen abweichen, sind verboten. Die Reproduktion, Übersetzung sowie die elektronische und fotografische Archivierung und Veränderung bedarf der schriftlichen Genehmigung durch den Hersteller. Zuwiderhandlungen verpflichten zu Schadenersatz.

#### **Änderungsvorbehalt**

Jegliche Änderungen, die dem technischen Fortschritt dienen, vorbehalten.

#### **Dokumenteninformation**

Ausgabe-/Rev.-Datum: 08/02/2022  
Dokument-/Rev.-Nr.: TR-ECE-TI-DGB-0388 v01  
Dateiname: TR-ECE-TI-DGB-0388-01.docx  
Verfasser: MÜJ, DIR

#### **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

<b>Inhaltsverzeichnis .....</b>	<b>3</b>
<b>Änderungs-Index .....</b>	<b>4</b>
<b>1 Allgemeines .....</b>	<b>5</b>
1.1 Geltungsbereich.....	5
1.2 Verwendete Abkürzungen und Begriffe.....	6
<b>2 Sicherheitshinweise .....</b>	<b>7</b>
2.1 Symbol- und Hinweis-Definition.....	7
2.2 Organisatorische Maßnahmen .....	8
2.3 Personalqualifikation.....	8
2.4 Nutzungsbedingungen der Softwarebeispiele .....	8
<b>3 Sicherheitsprogramm erstellen .....</b>	<b>9</b>
3.1 Hinweise zum Projekt .....	9
3.1.1 Zugriffschutz .....	9
3.1.2 Voraussetzungen .....	10
3.2 Installation der Gerätebeschreibung.....	11
3.3 Hardware-Konfiguration.....	12
3.4 Parametereinstellung graue Istwerte .....	15
3.5 Zweitschnittstelle (optional) .....	16
3.6 openSAFETY-Projekt anlegen (TR-Profile).....	17
3.6.1 Sichere Parametrierung .....	18
3.6.2 Erstellen des Sicherheitsprogramms .....	19
3.6.3 Generieren des Sicherheitsprogramms .....	20
3.6.4 Sicherheitsprogramm laden .....	20
3.6.5 Sicherheitsprogramm testen .....	22
3.7 openSAFETY-Projekt anlegen (TR-Legacy) .....	23
<b>4 Sicherheitsprogramm erweitern - Anwendungsbeispiele .....</b>	<b>25</b>
4.1 Preset-Durchführung .....	26
4.2 Sichere Geschwindigkeit (SLS) .....	30
4.2.1 Codebeispiel SLS .....	30
4.3 Sichere Stillstandserfassung .....	31
4.3.1 Codebeispiel sichere Stillstandserfassung .....	31
4.4 Sichere Positionserfassung .....	32
4.4.1 Codebeispiel für die Erfassung eines sicheren Positionsbereiches .....	33
4.5 Sichere Richtungserfassung.....	34
4.5.1 Codebeispiel für die sichere Richtungserfassung .....	34
4.6 Typumwandlung der Mess-System - Istwerte im SafeDESIGNER .....	35
<b>5 Download - Softwarebeispiele .....</b>	<b>36</b>
5.1 Legacy-Beispiel-Projekte .....	36

## Änderungs-Index

Änderung	Datum	Index
Erstausgabe für CD_582M-EPL, vorläufige Version	10.06.2022	00
Dokument finalisiert und Übersetzung ergänzt	02.08.2022	01

---

## 1 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.

### 1.1 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:

- CD\_582M-EPL

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 MASAFETY-X20SLXxxx-GER\_V1141
- anlagenspezifische Betriebsanleitungen des Betreibers
- Sicherheitshandbuch TR-ECE-BA-D-0142
- schnittstellenspezifische Benutzerhandbuch TR-ECE-BA-D-0169
- und diese optionale „Technische Information“

## 1.2 Verwendete Abkürzungen und Begriffe

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
OSDD	openSAFETY Device Description
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 Sicherheitshinweise

### 2.1 Symbol- und Hinweis-Definition



bedeutet, dass Tod oder schwere Körperverletzung eintreten wird, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



bedeutet, dass Tod oder schwere Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



bezeichnet wichtige Informationen bzw. Merkmale und Anwendungstipps des verwendeten Produkts.

---

### 2.2 Organisatorische Maßnahmen

Das mit Tätigkeiten am Mess-System beauftragte Personal muss vor Arbeitsbeginn das Sicherheitshandbuch [TR-ECE-BA-D-0142](#), insbesondere das Kapitel „Grundlegende Sicherheitshinweise“, gelesen und verstanden haben.

### 2.3 Personalqualifikation

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

### 2.4 Nutzungsbedingungen der Softwarebeispiele

---

#### **WARNUNG**

Für das fehlerfreie Funktionieren des Sicherheitsprogrammes und der Anwendungsbeispiele 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.

---

---

## 3 Sicherheitsprogramm erstellen

Im Folgenden wird das Beispielprojekt von TR-Electronic erläutert. Dazu gehören das Installieren der Gerätebeschreibung, das Projektieren des modularen TR-Devices und die Ausführung von Beispielcode für Preset und weitere sichere Funktionen.



Für den Betrieb der Drehgeberfamilie wird mindestens das Automation Studio V4.5 oder höher benötigt, da ältere Versionen die Installation einer sicheren Gerätebeschreibung (osdd) nicht unterstützen.

---

### 3.1 Hinweise zum Projekt

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

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-X (X20SLX806)`. Die SLX ist direkt an den POWERLINK MN angeschlossen und ist selbst der openSAFETY SCM.

Als MN bei POWERLINK kommt eine `X20CP482` zum Einsatz. Sie enthält eine POWERLINK-Anwendung, die ebenfalls die Mess-System - Informationen (u.a. die Istwerte „grauer Kanal“) auswerten kann.

#### 3.1.1 Zugriffsschutz

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 `SLX806`, z.B. zur Programmierung, ist ebenfalls durch ein Passwort geschützt.



Das Passwort lautet: `abc123`

---

### 3.1.2 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 Systemdokumentation 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 ein Beispiel. Der Anwender ist daher verpflichtet, die Verwendbarkeit der Projektierung für seine Applikation zu überprüfen und anzupassen. Dazu gehören auch die Auswahl der geeigneten sicherheitsgerichteten Hardwarekomponenten, sowie die notwendigen Softwarevoraussetzungen.

---

#### **Für das Konfigurationsbeispiel verwendete Software-Komponenten:**

- Automation Studio V4.10.2.37
- SafeDESIGNER V4.3.3.7



Die Verwendung nachfolgender Versionen ist ebenfalls möglich.

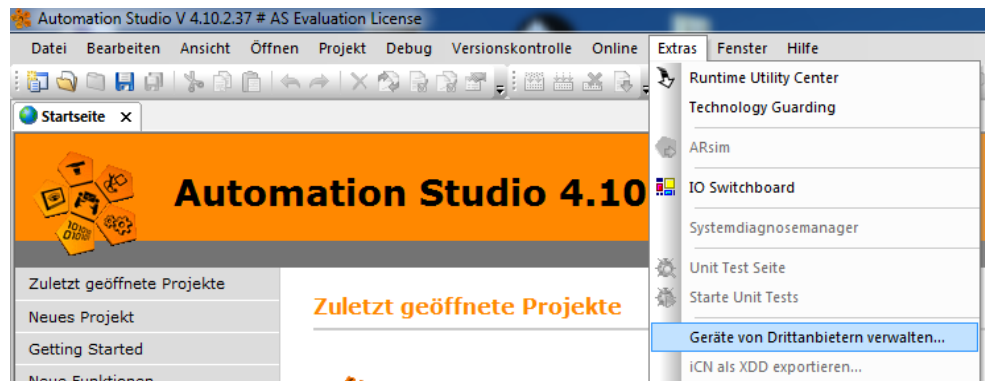
---

#### **Für das Konfigurationsbeispiel benutzte Hardware-Komponenten:**

- POWERLINK-MN: X20CP482 mit X20PS9600
- openSAFETY SCM: X20SLX806

## 3.2 Installation der Gerätebeschreibung

Um das modulare Mess-System zu projektieren, müssen zunächst die zugehörigen Gerätebeschreibungsdateien installiert werden. Dazu gehören eine Kopfstation und steckbare Module. Die Verwaltung der Gerätebeschreibungen findet man im Menü unter dem Punkt „Geräte von Drittanbietern verwalten...“.



Im sogenannten „Fremdgerätemanager“ werden nun die Gerätebeschreibungsdateien in folgender Reihenfolge installiert („Feldbus Gerät(e) importieren“):

1. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_Head.xdd
2. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_TR-Profile.xosdd  
0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_TR-Profile.xdd
3. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_Legacy.xosdd  
0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_Legacy.xdd
4. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_ITF2\_SSI.xdd
5. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_ITF2\_Incr.xdd



Bei den sicheren Modulen muss als Erstes die xosdd-Datei und dann erst die entsprechende xdd-Datei ausgewählt werden.

Nach der Installation der Gerätebeschreibungsdateien erscheinen die Kopfstation und die Module in der Liste des Fremdgerätemanagers:

Fremdgerätemanager						
Automation Studio Fremdgerätemanager						
Dieser Dialog erlaubt es Ihnen Feldbus- und DTM Fremdgeräte zu verwalten						
Suchen...						
Name	Version	Hersteller	DTM	Protokoll(e)	Beschreibung	
TR-CD582 Slot ITF2 Incremental	1.00	TR Electronic GmbH		module_interface	TR-Electro	
TR-CD582 Slot ITF2 SSI	1.00	TR Electronic GmbH		module_interface	TR-Electro	
TR-CD582 Slot Legacy	1.00	TR Electronic GmbH		module_interface	TR-Electro	
TR-CD582 Slot TR-Profile	1.00	TR Electronic GmbH		module_interface	TR-Electro	
TR-CD582-EPL	1.00	TR Electronic GmbH		Powerlink	TR-Electro	
TR-CD75	1.02	TR Electronic GmbH		Powerlink	TR-Electro	

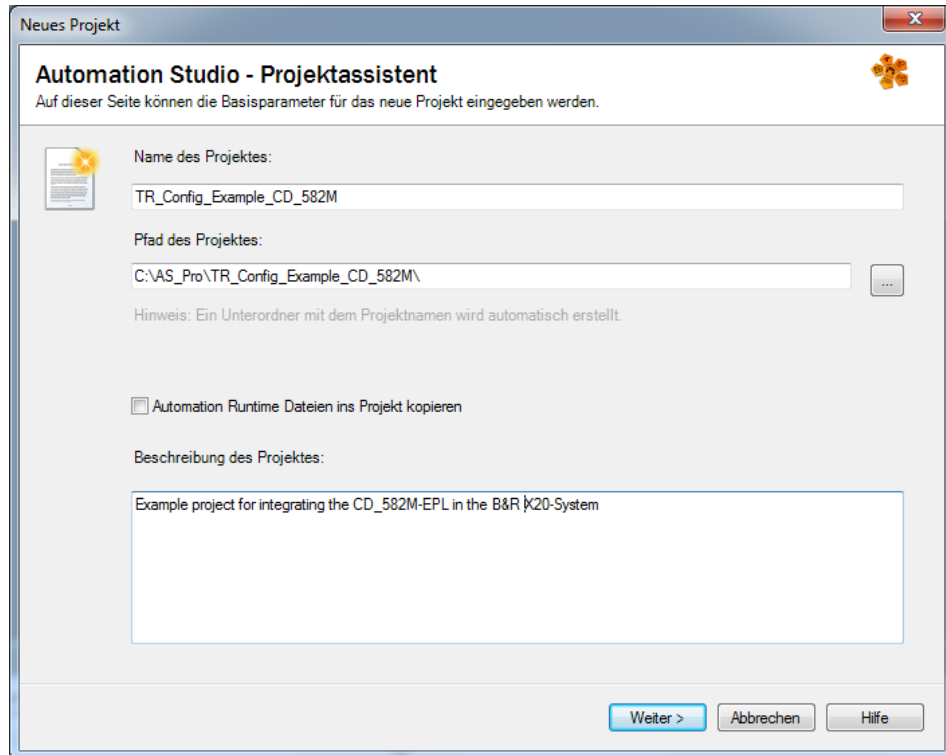
Im Anschluss kann mit der Erstellung des Projektes begonnen werden.



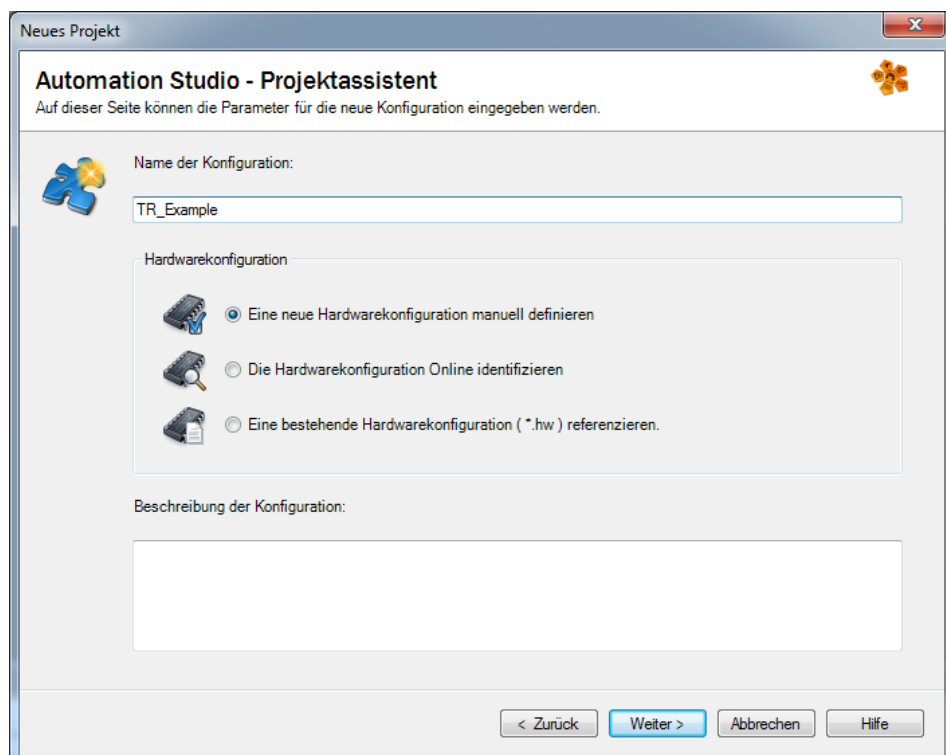
Um neu installierte Fremdgerätebeschreibungen zu projektieren, kann es notwendig sein, das aktuell geöffnete Projekt erneut zu öffnen oder das Automation Studio neu zu starten.

## 3.3 Hardware-Konfiguration

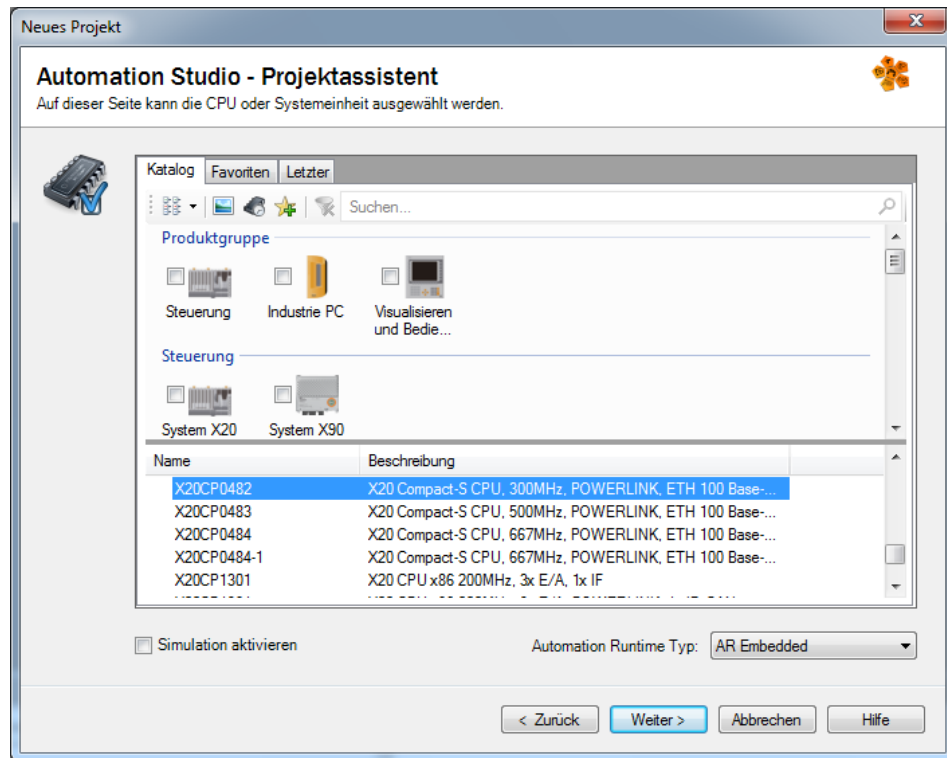
- Automation Studio starten und ein neues Projekt anlegen.



- Es wird eine neue Hardwarekonfiguration erstellt:



- Abschließend ist die korrekte CPU auszuwählen: X20CP0482



- 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.

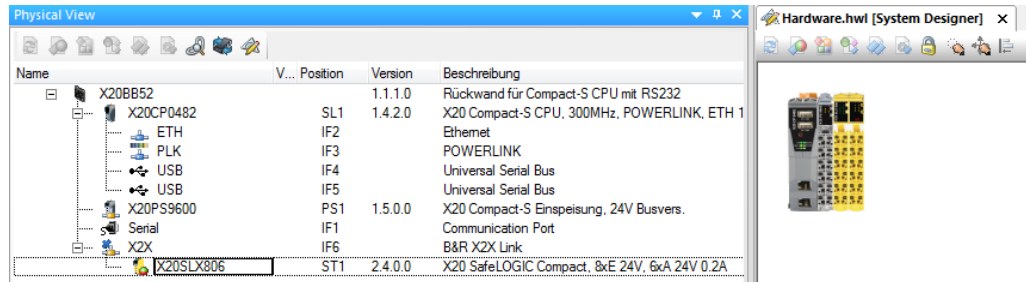


Für das Mess-System wird im Beispiel das Safety Release 1.10 verwendet. Alternativ kann die sichere Applikation via mappSafety erstellt werden.

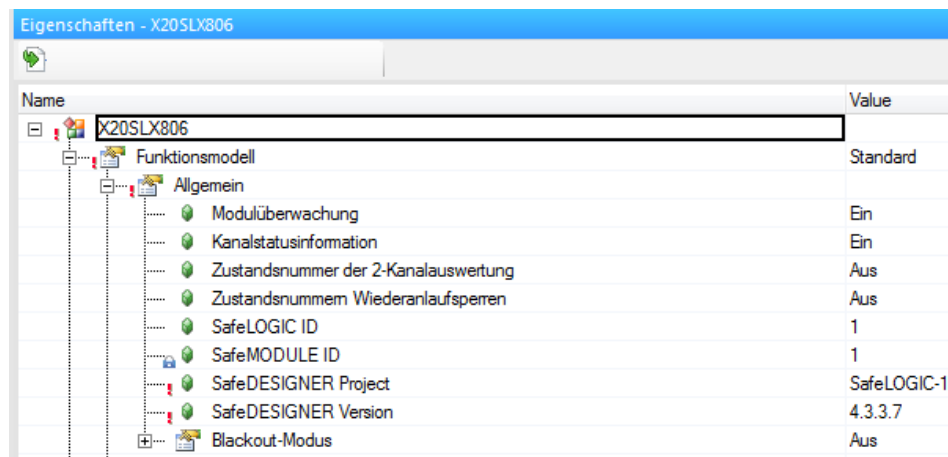
X20CP0482

Komponente	Bevorzugt	In Verwendung	Scope
Automation Runtime	D4.52	D4.52	
Visual Components	nicht definiert	nicht definiert	
mapp Motion	nicht definiert	nicht definiert	
ACP10 ARNC0 (Motion)	nicht definiert	nicht definiert	
mapp Safety	nicht definiert	nicht definiert	
mapp Vision	nicht definiert	nicht definiert	
mapp Services	nicht definiert	nicht definiert	
mapp View	nicht definiert	nicht definiert	
mapp Control	nicht definiert	nicht definiert	
mapp Cockpit	nicht definiert	nicht definiert	
Safety Release	1.10	1.10	

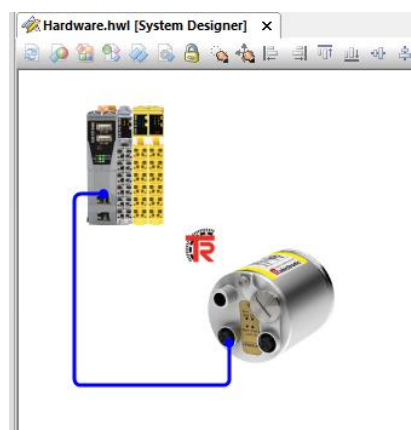
- SCM zum EPL-Netz hinzufügen, indem man das entsprechende Gerät aus dem Hardware Katalog auf die POWERLINK-Schnittstelle im Physical View zieht:



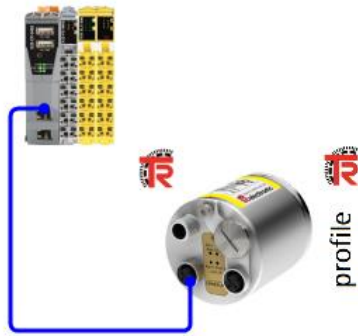
- Die SafeDESIGNER-Version kann unter der Konfiguration der SafeLOGIC ausgewählt werden:



- Nun kann das TR-Mess-System hinzugefügt werden. Wir beginnen mit der nicht sicheren Kopfstation (nur sichtbar, wenn „Safety“-Filter im Hardware Katalog nicht eingeschaltet ist) „TR-CD582-EPL“ und verbinden die POWERLINK-Schnittstelle mit der Steuerung:



- Im Anschluss kann ein Modul mit dem Mess-System verknüpft werden. Im Beispiel wird das TR-Profil „TR-CD582 Slot TR-Profile“ ausgewählt:



- Als POWERLINK Node-ID wird für das Mess-System im Beispiel die 18 verwendet. Somit ist die Hardware-Konfiguration abgeschlossen. Das Projekt stellt sich in Physical View folgendermaßen dar:

Physical View				
Name	V...	Position	Version	Beschreibung
X20BB52			1.1.1.0	Rückwand für Comp...
X20CP0482		SL1	1.4.2.0	X20 Compact-S CPI
ETH		IF2		Ethernet
PLK		IF3		POWERLINK
TR-CD582-EPL		ST18	1.00	TR-Electronic CD_5
module_interface		IF1		
TR-CD582_Slot_TR-Profile		ST1	1.00	TR-Electronic CD_5
USB		IF4		Universal Serial Bus
USB		IF5		Universal Serial Bus
X20PS9600		PS1	1.5.0.0	X20 Compact-S Eins
Serial		IF1		Communication Port
X2X		IF6		B&R X2X Link
X20SLX806		ST1	1.10.12.1	X20 SafeLOGIC Cor

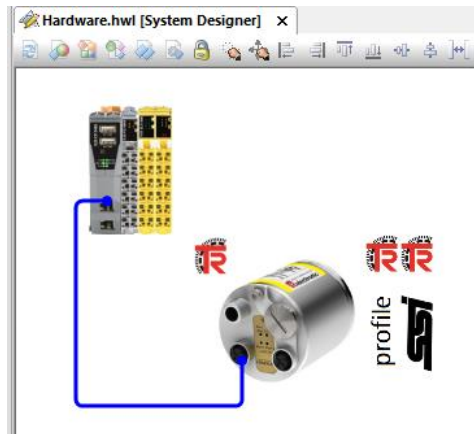
### 3.4 Parametereinstellung graue Istwerte

Die Parameter für den einkanaligen grauen Istwert können beim TR-Profil komplett unabhängig von den Safety-Parametern eingestellt werden. Die Einstellung erfolgt in den Eigenschaften des TR-Profile-Slots:

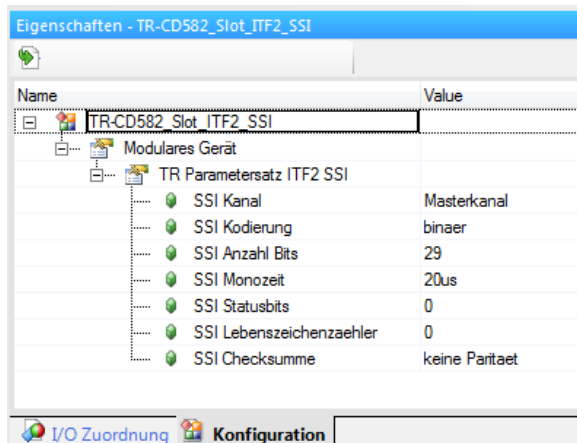
Eigenschaften - TR-CD582_Slot_TR-Profile	
Name	Value
TR-CD582_Slot_TR-Profile	
Allgemein	
SafeDOMAIN ID	1
SafeNODE ID	2
Modulares Gerät	
TR Parametersatz Profil	
Drehrichtung	vorwaerts
Messlaenge	536870912
Umdrehungen Zaehler	65536
Umdrehungen Nenner	1
Geschwindigkeit Einheit	(U/min) * Faktor
Geschwindigkeit Faktor	1
Geschwindigkeit Zeit	100
Geschwindigkeit Filtertiefe	0
Geschwindigkeit Filterdynamik	Filter statisch
Remote Ersatzwert	grauer Ersatzwert
Gekoppelt an sicheren Sensor	nicht gekoppelt

## 3.5 Zweitschnittstelle (optional)

Bei Mess-Systemen mit einer Zweitschnittstelle kann diese ebenfalls im Automation Studio eingestellt werden. Dafür muss die entsprechende Zweitschnittstelle zusätzlich gesteckt werden (hier beispielhaft „TR-CD582 Slot ITF2 SSI“):



Im Anschluss können die Zweitschnittstellenparameter unter den Eigenschaften des SSI-Slots eingestellt werden:



Die Zweitschnittstelle besitzt keine zyklischen Feldbus-Daten. Die Steuerung überträgt im Hochlauf lediglich die Parameter für die Zweitschnittstelle.



Die Zweitschnittstellen-Parameter werden vom Mess-System gespeichert. Wurde das Mess-System einmal parametrierungsläufig, läuft die Zweitschnittstelle auch ohne Steuerung mit den korrekten Parametern an.

### 3.6 openSAFETY-Projekt anlegen (TR-Profile)

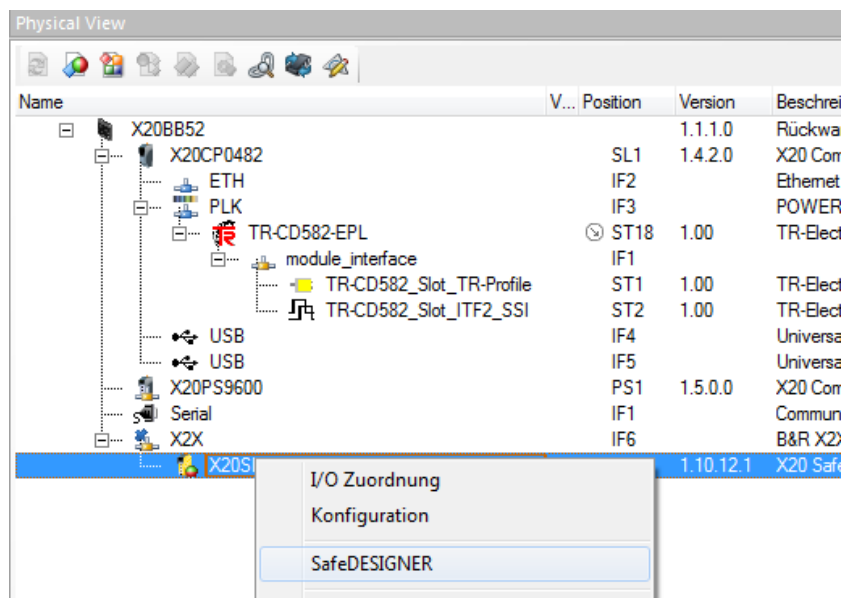
Im vorliegenden Beispiel wurde für das TR-Mess-System (Slot TR-Profile) die SafeMODULE ID = 2 im openSAFETY-Netzwerk vergeben.

Eigenschaften - TR-CD582_Slot_TR-Profile	
Name	Value
TR-CD582_Slot_TR-Profile	
Allgemein	
SafeDOMAIN ID	1
SafeNODE ID	2
Modulares Gerät	
TR Parametersatz Profil	
Drehrichtung	vorwaerts
Messlaenge	536870912



Das Safety-Projekt wurde vom Automation-Studio automatisch angelegt und mit der SLX verknüpft.

- 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. Unter dem Mess-System sind alle im SafeDESIGNER verfügbaren Ein- und Ausgangsdaten des Mess-Systems aufgelistet:

ID	Name	Variable	Typ	CPU Variable	CPU Adresse
SL1					
SL1.SM1	X20SLX806				IF6.ST1
SL1.SM2	TR-CD582_Slot_TR-Profile				IF3.ST18.IF1.ST1
General					
SafeModuleOk			SAFEBOOL		
InputSafeChannels					
SafeSpeedError			SAFEBOOL		
SafeErrorAckRequest			SAFEBOOL		
SafePresetOK			SAFEBOOL		
SafePresetError			SAFEBOOL		
SafeState			SAFEBOOL		
SafePresetActive			SAFEBOOL		
SafePresetLocked			SAFEBOOL		
SafeScalingError			SAFEBOOL		
SafeTRStatus2			SAFEUSINT		
SafeTRInputPos			SAFEDWORD		
SafeTRInputVel			SAFEDINT		
OutputSafeChannels					
SafePresetPreparation			SAFEBOOL		
SafePresetRequest			SAFEBOOL		
SafeTRPresetValue			SAFEDWORD		

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

## 3.6.1 Sichere Parametrierung

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

Parameter	Wert
<b>GenericParameters</b>	
SIL	SIL2
VIT Rotary Direction	Forward
Measuring Length	536870912
Rotations Numerator	65536
Rotations Denominator	1
Speed Format	rot per min * factor
Speed Factor	1
Speed Time	100
Speed Filter Type	static
Speed Filter Depth	0
Window Increments	1000

Die einzelnen Werte sind im Benutzerhandbuch beschrieben. Die sicheren Parameter werden beim Hochlauf automatisch vom SCM an das Mess-System (SN) übertragen.

### 3.6.2 Erstellen des Sicherheitsprogramms

Eine sichere Applikation, die auf dem SCM läuft, wird im Editor des SafeDESIGNERS erstellt. Folgende Schritte zeigen den Aufbau eines Programms, welches zunächst über zwei digitale Ausgänge der SLX806 die Zustandsbits `SafeModuleOk` und `SafeState` ausgibt.

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

14		SM2_SafeState	SAFEBOOL		SL1.SM2.SafeState
15		SM2_SafeTRInputVel	SAFEDINT		SL1.SM2.SafeTRInputVel
16		SM2_SafeModuleOk	SAFEBOOL		SL1.SM2.SafeModuleOk
17		SM2_SafePresetOK	SAFEBOOL		SL1.SM2.SafePresetOK
18		SM2_SafePresetError	SAFEBOOL		SL1.SM2.SafePresetError
19		SM2_SafeSpeedError	SAFEBOOL		SL1.SM2.SafeSpeedError
20		SM2_SafeTRInputPos	SAFEDWORD		SL1.SM2.SafeTRInputPos
21		SM2_PresetPreparation	SAFEBOOL		SL1.SM2.SafePresetPreparation
22		SM2_PresetRequest	SAFEBOOL		SL1.SM2.SafePresetRequest
23		SM2_PresetValue	SAFEDWORD		SL1.SM2.SafeTRPresetValue

- Variablen mit den Ausgängen der SLX verknüpfen:

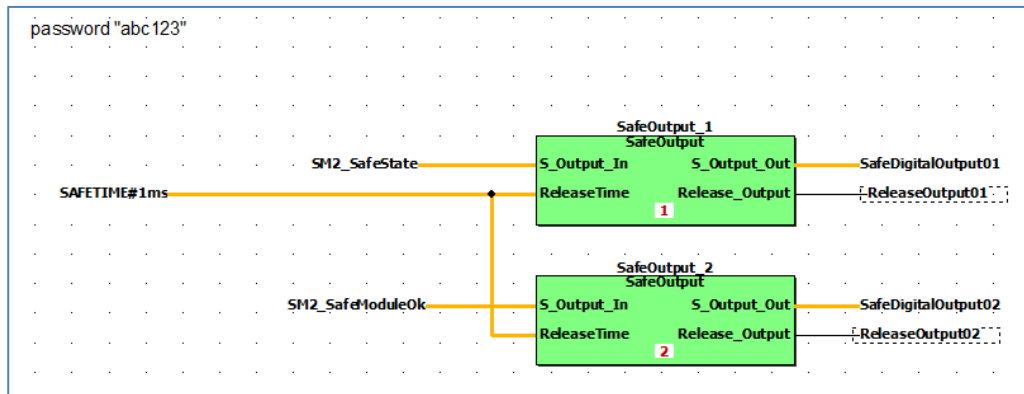
1		NewGroup			
2		SafeDigitalOutput01	SAFEBOOL		SL1.SM1.SafeDigitalOutput01
3		SafeDigitalOutput02	SAFEBOOL		SL1.SM1.SafeDigitalOutput02



Die digitalen Ausgänge der SLX müssen im Automation Studio unter den Einstellungen der SLX auf „über SafeLogic“ gesetzt werden.

Eigenschaften - X20SLX806	
Name	Value
X20SLX806	
Funktionsmodell	Standard
Allgemein	
Ausgangssignalpfad	
DigitalOutput01	über SafeLOGIC
DigitalOutput02	über SafeLOGIC
DigitalOutput03	über SafeLOGIC
DigitalOutput04	über SafeLOGIC
DigitalOutput05	über SafeLOGIC
DigitalOutput06	über SafeLOGIC
Kommunikation von SafeDESIGNER zur...	
SPROXY aktivieren	Ein

- Zum Reiter Code: Main wechseln. Hier wird nun die eigentliche Applikation grafisch programmiert:



Die Erstellung der Beispielapplikation ist damit abgeschlossen. Die digitalen Ausgänge 1 und 2 der SLX806 werden entsprechend der Mess-Systemen-Statusbits gesetzt.

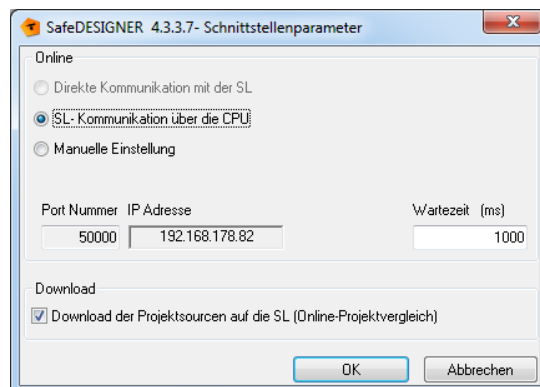
### 3.6.3 Generieren des Sicherheitsprogramms

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

### 3.6.4 Sicherheitsprogramm laden

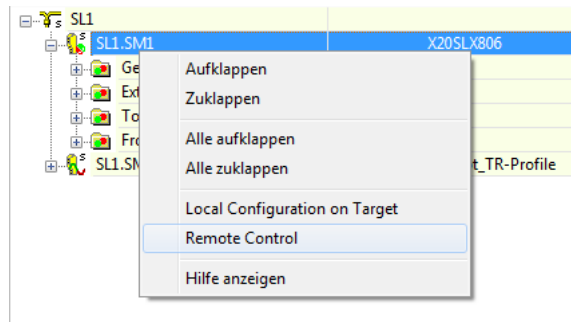
Um die folgenden Schritte durchzuführen, muss zunächst das Projekt aus dem Automation Studio kompiliert und in die X20CP0482 programmiert werden (=POWERLINK-Projekt).

Nachdem das Sicherheitsprogramm generiert wurde, 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, sollte Einstellung SL - communication through BR-CPU verwendet werden:



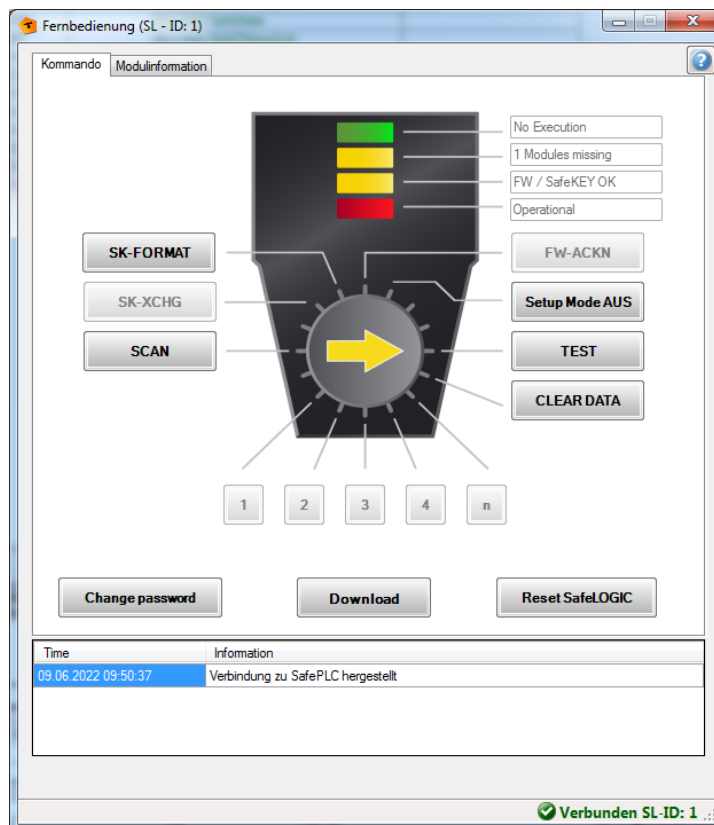
Der SafeDESIGNER verwendet die Portnummer sowie die aktuelle Onlinekonfiguration aus dem Automation Studio als Basis für die Onlinekommunikation.

Für das Laden der Software kann die „Remote Control“ der SLX verwendet werden. Sie wird über den Safety View gestartet:



Das Projekt wird über die Schaltfläche „Download“ in die SLX geladen.

Nach dem Download startet die SafeLogic neu. Nun müssen Firmware und neue Teilnehmer an der SLX quittiert werden. Für die einzelnen Quittierungen wird ebenfalls die Remote-Bedienung der Steuerung verwendet:



Nachdem die neue Firmware (FW-ACKN) und die neuen Teilnehmer quittiert wurden (1...n) läuft das System an und das Mess-System wechselt in den sicheren Betrieb. Nun leuchten die zwei Status-LEDs des Mess-Systems grün: EPL operational und openSAFETY operational.

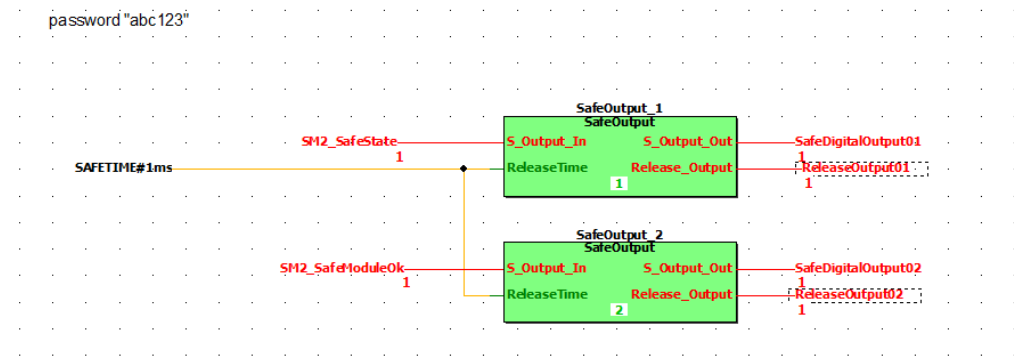


Da sich die Blinkmuster und Hinweise je nach Vorgehensweise und vorinstallierter Software leicht unterscheiden können, muss für diesen Vorgang zusätzlich das Handbuch der SL herangezogen werden.

## 3.6.5 Sicherheitsprogramm testen

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

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



Erst wenn der Hochlauf über openSAFETY und die sichere Parametrierung erfolgt sind, gibt das Mess-System die aktuellen Istwerte aus. Bis dahin sind alle Safe-Variablen auf den Wert 0 gesetzt.

Alle Istwerte (grau und safe) aller Mess-Systemen - IO-Daten sind auch im Online-Mode des Automation Studios sichtbar, sie werden unter der I/O Zuordnung des Mess-Systems angezeigt:

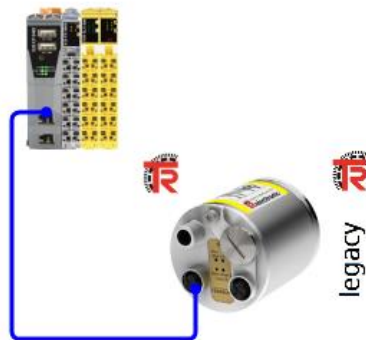
Eigenschaften - TR-CD582_Slot_TR-Profil	
Kanalname	Physikalischer Wert
TR_Grey_Input_I0000_S1E_Cyc_In_Status	2
TR_Grey_Input_I0000_S1E_Cyc_In_Position	268467057
TR_Grey_Input_I0000_S1E_Cyc_In_Velocity	0
TR_Grey_Output_I0000_S20_Cyc_Out_Control	0
TR_Grey_Output_I0000_S20_Cyc_Out_Preset	0
SafeSpeedError	FALSE
SafeErrorAckRequest	FALSE
SafePresetOK	FALSE
SafePresetError	FALSE
SafeState	TRUE
SafePresetActive	FALSE
SafePresetLocked	FALSE
SafeScalingError	FALSE
SafeTRStatus2	0
SafeTRInputPos	7426
SafeTRInputVel	0

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 im POWERLINK-Projekt genutzt werden. Sie sind statisch gemappt und immer verfügbar.

### 3.7 openSAFETY-Projekt anlegen (TR-Legacy)

Wenn die Funktionalitäten des TR-Profiles (Geschwindigkeitseinheiten, Mittelung, Skalierung, Preset-on-the-fly etc.) nicht benötigt werden und der Anwender Wert darauf legt, dass sich das CD\_582M-EPL – Mess-System so verhält wie das CD\_75M-EPL – Mess-System so ist ein Betrieb im sogenannten Legacy-Modus möglich. Die Parametrierung im Legacy-Format erfolgt auf die gleiche Art und Weise wie beim TR-Profil im vorhergehenden Kapitel. Es wird lediglich ein anderes Modul gesteckt („TR-CD582 Slot Legacy“ statt „TR-CD582 Slot TR-Profil“):

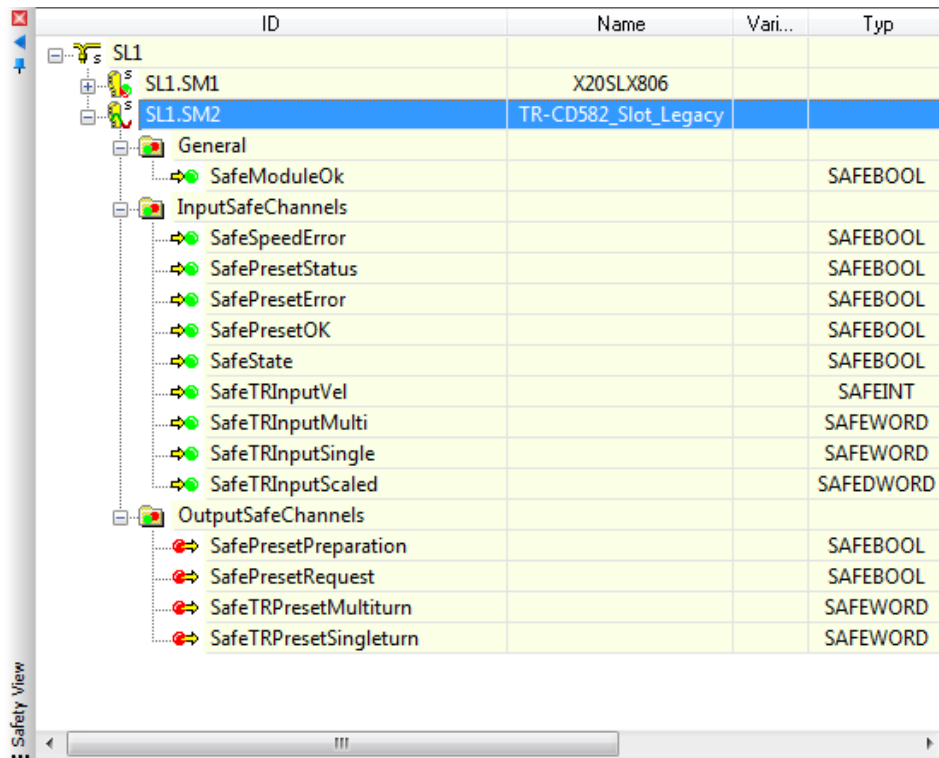


Dadurch werden die einzustellenden Parameter und die zyklischen Daten im Legacy-Format dargestellt.

Graue zyklische Daten im Automation Studio:

Eigenschaften - TR-CD582_Slot_Legacy		
Kanalname	Prozessvariable	Datentyp
TR_Grey_Input_I0000_S1E_Cyc_In_Velo_Overflow		USINT
TR_Grey_Input_I0000_S1E_Cyc_In_Velo_16		INT
TR_Grey_Input_I0000_S1E_Cyc_In_Multitum		UINT
TR_Grey_Input_I0000_S1E_Cyc_In_Singletum		UINT
TR_Grey_Input_I0000_S1E_Cyc_In_Scaled		UDINT
SafeSpeedError		BOOL
SafePresetStatus		BOOL
SafePresetError		BOOL
SafePresetOK		BOOL
SafeState		BOOL
SafeTRInputVel		INT
SafeTRInputPosMulti		UINT
SafeTRInputPosSingle		UINT
SafeTRInputPosScaled		UDINT

Safety View im Safe Designer für den Legacy-Mode:



Sichere Parameter im Safe Designer für den Legacy Mode:

Materialnummer:	<b>TR-Electronic_CD_582_-EPL_Legacy</b>		
Beschreibung:	<b>TR-Electronic CD_582M - POWERLINK / openSAFETY (TR-Child)</b>		
SafeMODULE ID:	2		
Importdatei:	-		
Parameter	Wert	Einheit	
<b>GenericParameters</b>			
Integration Time	2	none	
Window Increments	1000	none	
Idleness Preset Tolerance	1	none	
VIT Rotary Direction	Forward		
<span>Basic</span> / <span>Safety Response Time</span> / <span>Basic</span> / <b><span>GenericParameters</span></b> / <span>ALL</span>			

Die hier vorhandenen Datentypen und Dateninhalte entsprechen den Daten der CD\_75M-EPL-Familie. Dadurch ist es möglich, schon vorhandene Steuerungs-Software-Teile, die mit der CD\_75M-EPL-Familie entstanden sind, wiederzuverwenden.

## 4 Sicherheitsprogramm erweitern - Anwendungsbeispiele

Das unter Kapitel 3 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 einigen der 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!



Die hier vorgestellten Beispiele beziehen sich auf das TR-Profil. Beispiele für das Legacy-Format werden hier nicht gesondert aufgeführt, da diese Beispiele bereits in der Inbetriebnahme-Dokumentation der CD\_75M-EPL-Familie enthalten sind.

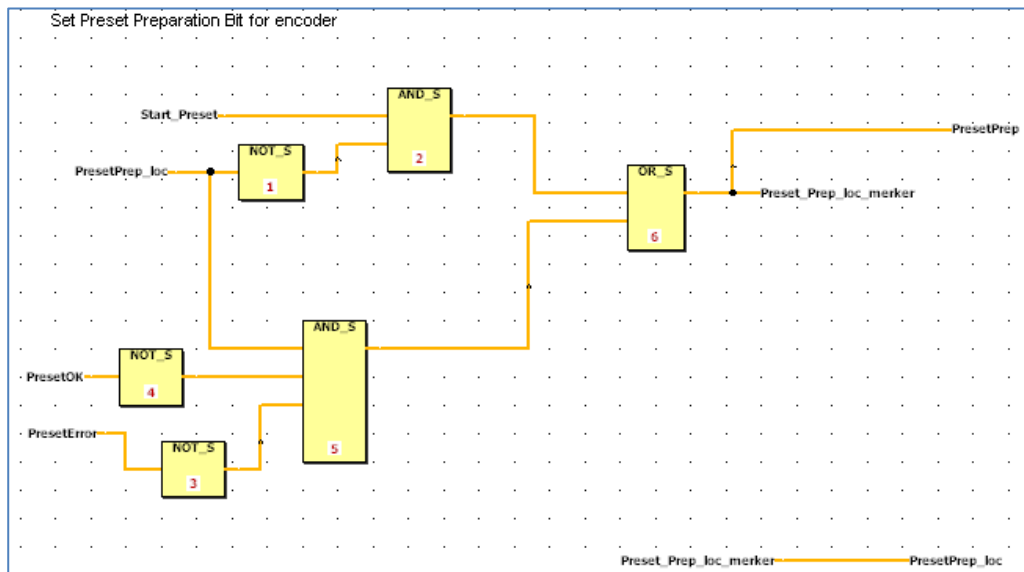
### 4.1 Preset-Durchführung

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		<input type="checkbox"/>
3	PresetReq	SAFEBOOL	VAR_OUTPUT	Set Preset Request	SAFEBOOL#0	<input type="checkbox"/>	<input type="checkbox"/>
4	PresetPrep_loc	SAFEBOOL	VAR	Set Preset Prep local	SAFEBOOL#0		<input checked="" type="checkbox"/>
5	PresetOK	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		<input type="checkbox"/>
6	PresetError	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		<input type="checkbox"/>
7	PresetFinish	SAFEBOOL	VAR_OUTPUT	Preset finished	SAFEBOOL#0	<input type="checkbox"/>	<input type="checkbox"/>
8	PresetFinish_loc	SAFEBOOL	VAR		SAFEBOOL#0		<input checked="" type="checkbox"/>
9	PresetFinish_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		<input type="checkbox"/>
10	PresetPrep	SAFEBOOL	VAR_OUTPUT	Set Preset Preparation	SAFEBOOL#0	<input type="checkbox"/>	<input type="checkbox"/>
11	Preset_Prep_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		<input type="checkbox"/>
12	PresetReq_loc	SAFEBOOL	VAR		SAFEBOOL#0		<input checked="" type="checkbox"/>
13	Reset_Lock	SAFEBOOL	VAR_INPUT	Lock Reset	SAFEBOOL#0		<input type="checkbox"/>
14	PresetReq_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		<input type="checkbox"/>

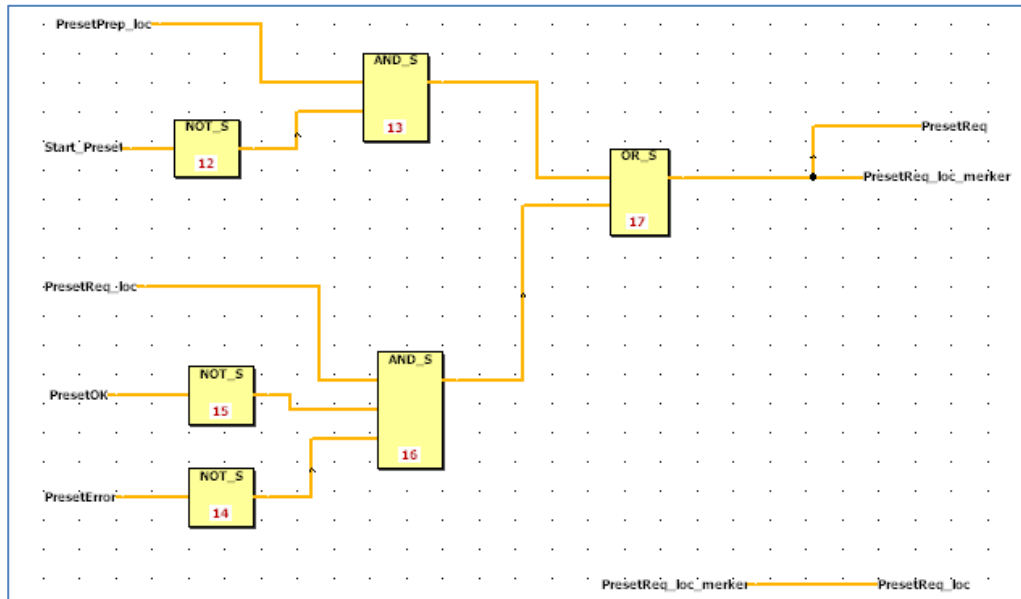
Folgende Darstellungen zeigen den Aufbau des neuen Funktions-Blocks TR\_PRESET\_Example:

#### 1. Setzen von Preset Preparation



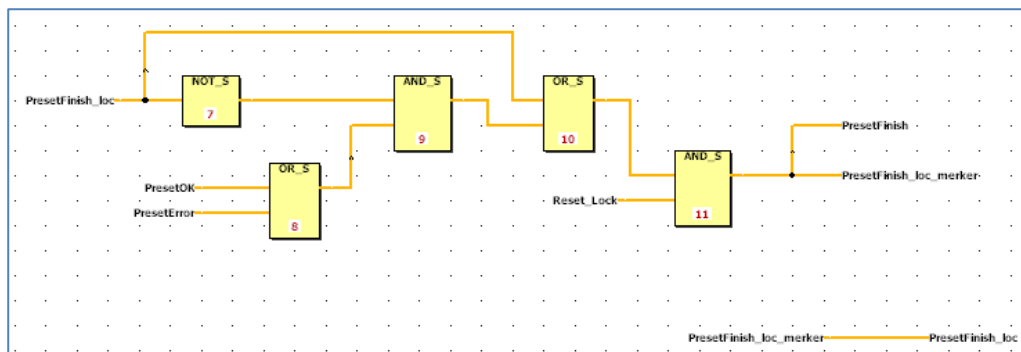
Im ersten Abschnitt wird durch das Setzen des Start\_Preset-Eingangs das Mess-System Ausgangssignal PresetPrep-Bit gesetzt. Das Signal bleibt gesetzt, bis der Preset-Ablauf im Mess-System beendet ist. Wenn der Preset-Ablauf ohne Fehler ausgeführt werden konnte wird vom Mess-System PresetOK gesetzt und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

## 2. Setzen von Preset Request



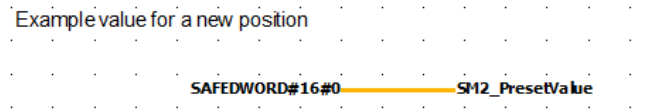
Wenn das Ausgangssignal PresetPrep zum Mess-System gesetzt ist und das Start\_Preset-Signal wieder auf FALSE gesetzt wird, so wird das Signal PresetReq gesetzt. Das Signal bleibt gesetzt, bis der Preset-Ablauf im Mess-System beendet ist. Wenn der Preset-Ablauf ohne Fehler ausgeführt werden konnte wird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

## 3. Setzen von Preset Finish

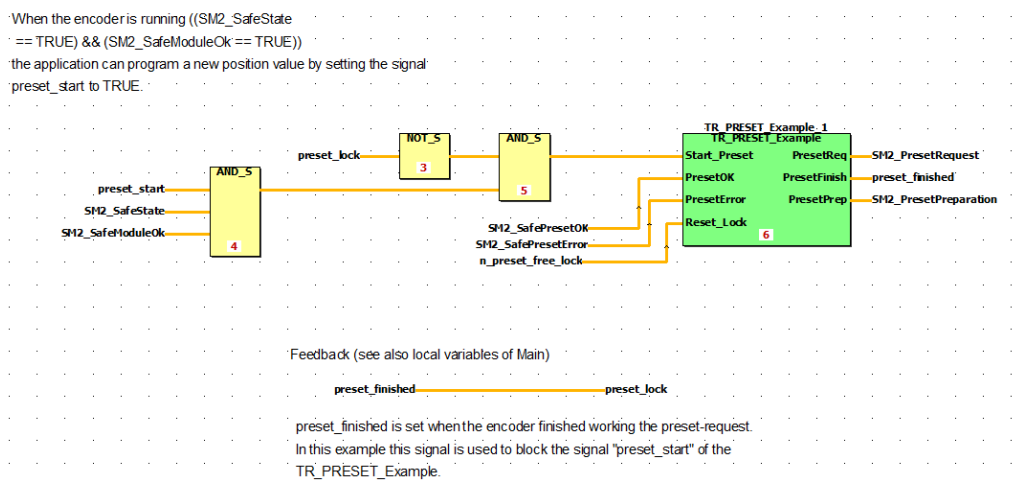


In diesem Abschnitt wird die Information gespeichert, dass die Durchführung beendet wurde. Reset\_Lock muss hierfür immer auf „1“ gesetzt sein, da die Variable low-aktiv ist.

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:



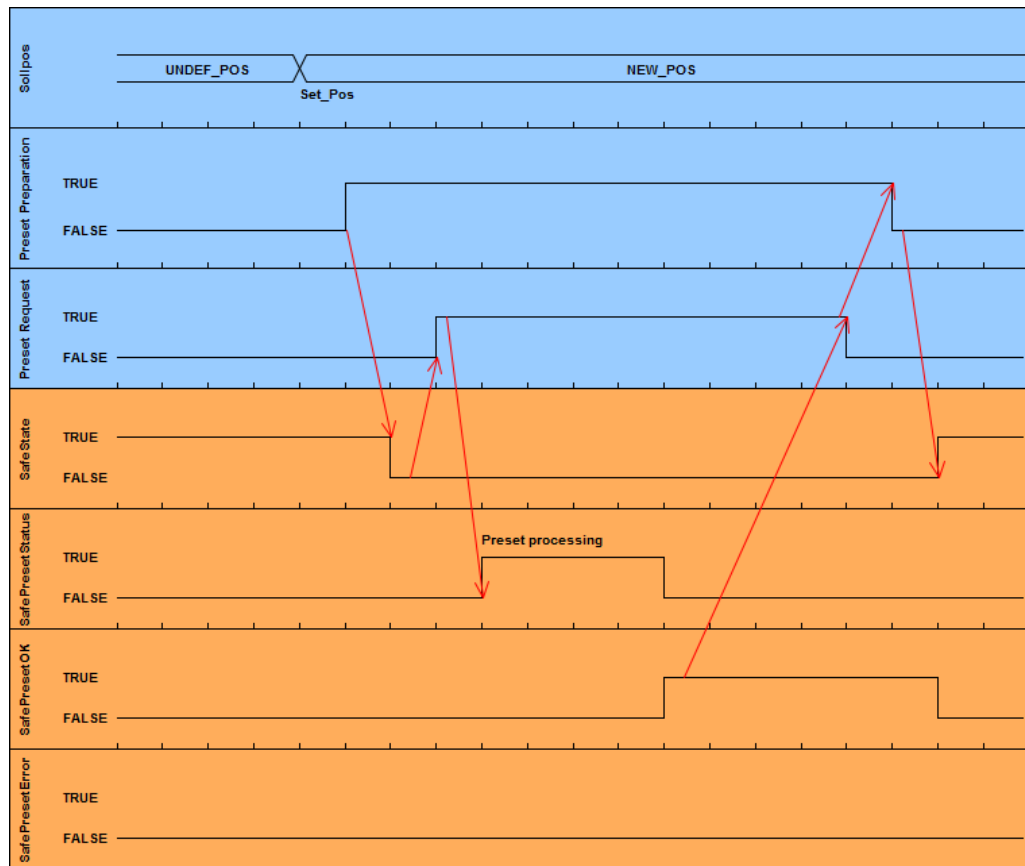
Die Ausgänge `PresetPrep` und `PresetReq` zum Mess-System hin werden mit den entsprechenden Ausgängen des `TR_PRESET_Example`-Funktionsblocks verbunden. Der Preset wird mit Hilfe eines Lock-Signals nur einmal je `Start_Preset`-Signal ausgeführt: Wenn das Signal `preset_lock` noch nicht gesetzt ist und das Signal `preset_start` 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` wieder auf `TRUE` gesetzt werden.

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

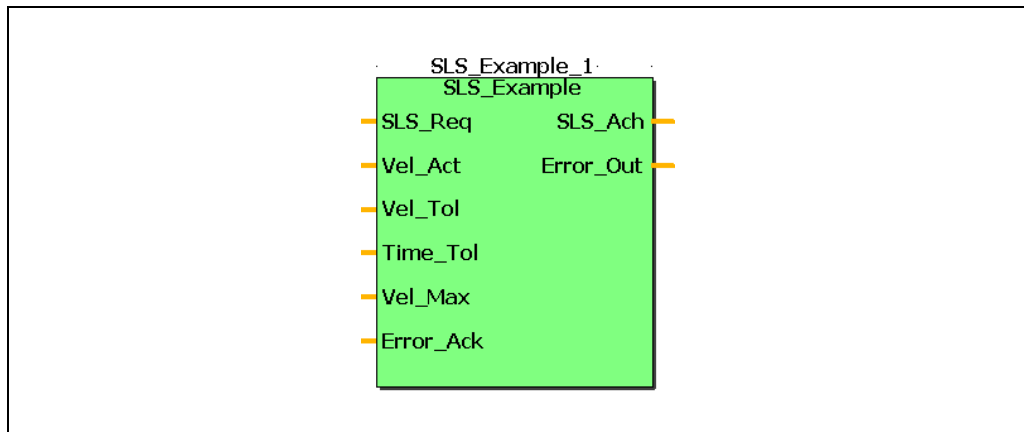
blauer Bereich: Ausgangssignale Steuerung -> Mess-System  
 oranger Bereich: Eingangssignale Mess-System -> Steuerung



## 4.2 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 parametrisierten Toleranzzeit `Time_Tol` unterhalb der parametrisierten sicheren Geschwindigkeit `Vel_Tol` sein. Ist dies der Fall, wird der Ausgang `SLS_Ach` gesetzt. Im Fehlerfall wird der Ausgang `Error_Out` gesetzt.

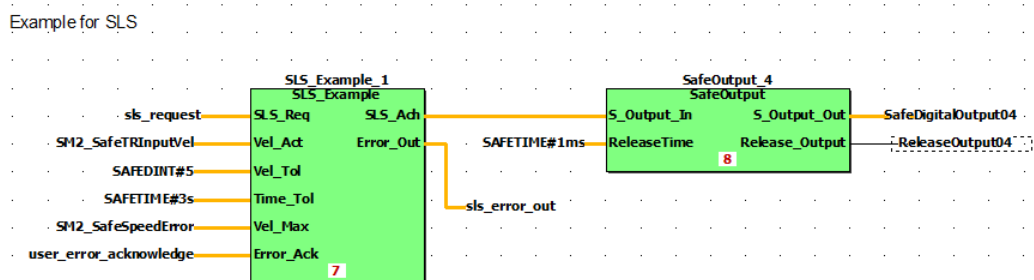
Ein Fehler kann mit `Error_Ack` quittiert werden.



Signal	Typ	Richtung	Beschreibung
SLS_Req	SAFEBOOL	Eingang	Anforderung der sicheren Geschwindigkeit mit <code>SLS_Req = TRUE</code>
Vel_Act	SAFEDINT	Eingang	Sichere Geschwindigkeit des Mess-Systems
Vel_Tol	SAFEDINT	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

### 4.2.1 Codebeispiel SLS

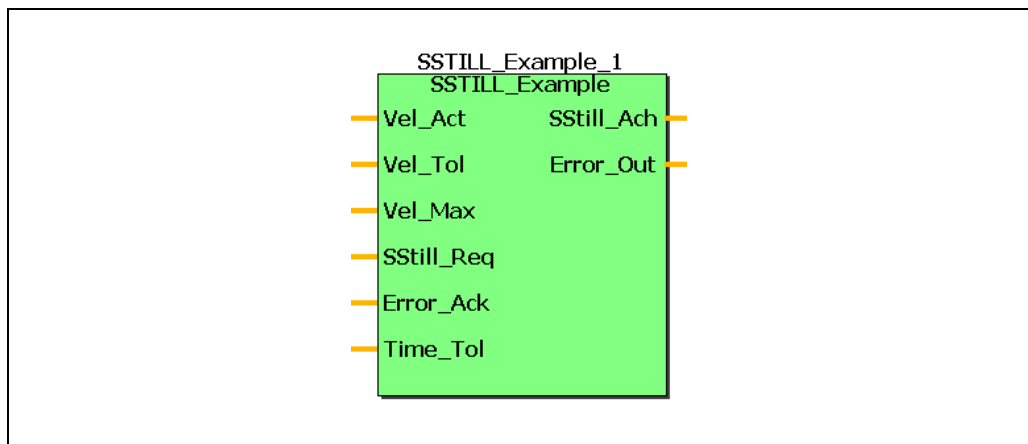
Im Beispielprojekt wird das angeforderte Ergebnis via Digitalausgang 4 bereitgestellt. Eine Fehlerauswertung erfolgt im Beispiel nicht.



### 4.3 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 parametrisierte Maximalgeschwindigkeit `Vel_Tol`, wird der Ausgang `Sstill_Ach` gesetzt. Nach der Anforderung auf Stillstand `Sstill_Req` muss `Vel_Act` innerhalb der parametrisierten Toleranzzeit `Time_Tol` unterhalb der parametrisierten Maximalgeschwindigkeit `Vel_Tol` für einen Stillstand sein. Ist dies nicht der Fall, wird der Ausgang `Error_Out` gesetzt.

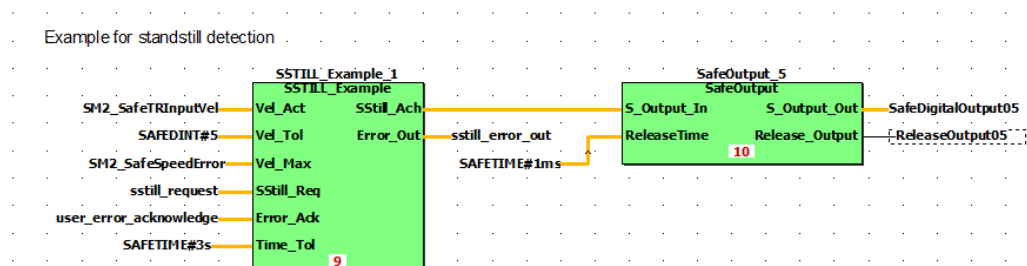
Ein Fehler kann mit `Error_Ack` quittiert werden.



Signal	Typ	Richtung	Beschreibung
Vel_Act	SAFEDINT	Eingang	Sichere Geschwindigkeit des Mess-Systems
Vel_Tol	SAFEDINT	Eingang	Grenze für Stillstand (Toleranz)
Vel_Max	SAFEBOOL	Eingang	Mess-System Geschwindigkeitsüberlauf
Sstill_Req	SAFEBOOL	Eingang	Anforderung des sicheren Stillstands mit <code>Sstill_Req = TRUE</code>
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

#### 4.3.1 Codebeispiel sichere Stillstandserfassung

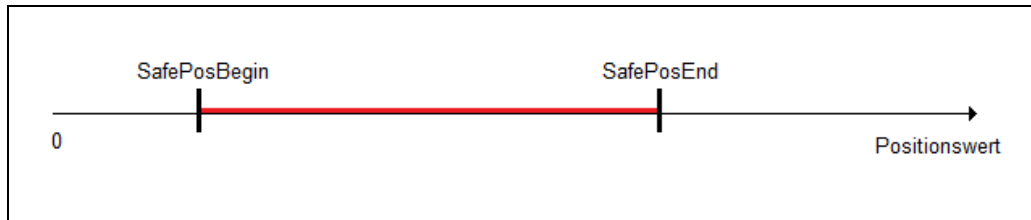
Im Beispielprojekt wird das angeforderte Ergebnis via Digitalausgang 5 bereitgestellt. Eine Fehlerauswertung erfolgt im Beispiel nicht.



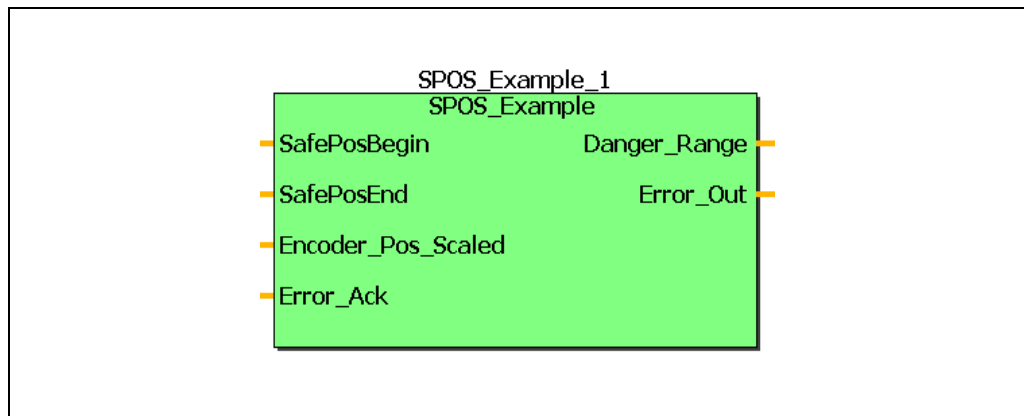
## 4.4 Sichere Positionserfassung

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  $\text{SafePosBegin} < \text{SafePosEnd}$  ist. Ein Überlauf ist nicht erlaubt und führt zu einer Fehlerausgabe.

Ein Fehler kann mit Error\_Ack quittiert werden.



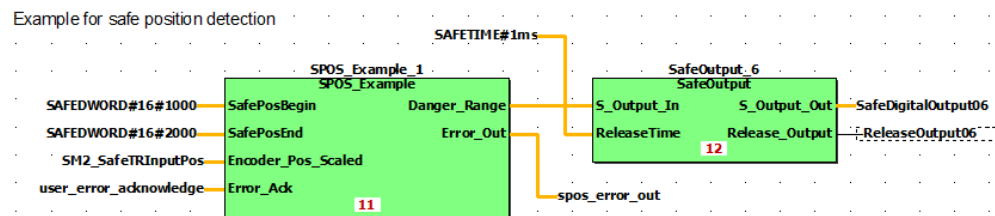
Im roten Bereich wird der Ausgang Danger\_Range gesetzt.



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: $\text{SafePosBegin} \leq \text{Encoder\_Pos\_Scaled} \leq \text{SafePosEnd}$
Error_Out	SAFEBOOL	Ausgang	Fehler liegt an, z.B. Zeitüberschreitung

#### 4.4.1 Codebeispiel für die Erfassung eines sicheren Positionsbereiches

Im Beispielprojekt wird das angeforderte Ergebnis via Digitalausgang 6 bereitgestellt. Eine Fehlerrauswertung erfolgt im Beispiel nicht.



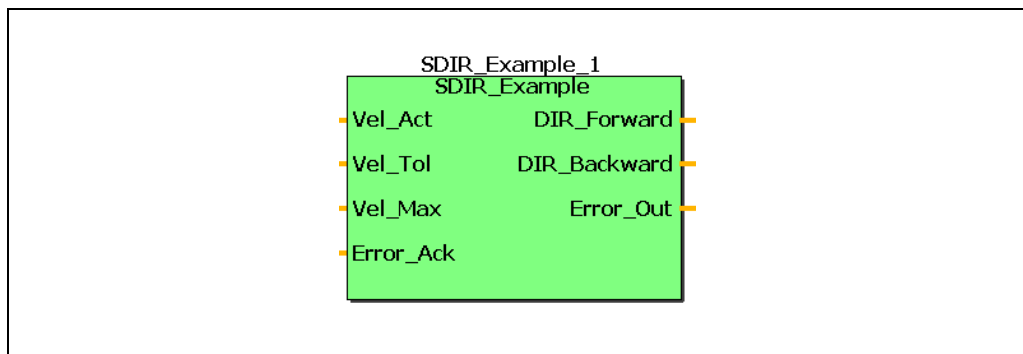
## 4.5 Sichere Richtungserfassung

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`.

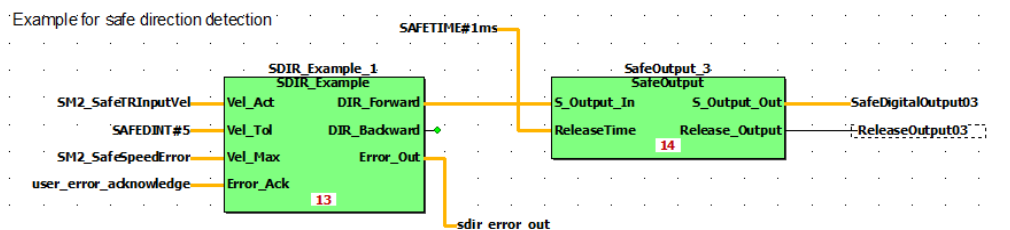
Ein Fehler kann mit `Error_Ack` quittiert werden.



Signal	Typ	Richtung	Beschreibung
Vel_Act	SAFEDINT	Eingang	Sichere Geschwindigkeit des Mess-Systems
Vel_Tol	SAFEDINT	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-System Istwert fallend
Error_Out	SAFEBOOL	Ausgang	Fehler liegt an, z.B. Geschwindigkeitsüberlauf

### 4.5.1 Codebeispiel für die sichere Richtungserfassung

Im Beispielprojekt wird eine erkannte Vorwärts-Bewegung via Digitalausgang 3 bereitgestellt. Eine Fehlerauswertung erfolgt im Beispiel nicht.



## 4.6 Typumwandlung der Mess-System - Istwerte im SafeDESIGNER

Die Mess-System-Istwerte für Singleturn, Multiturn und skaliertes Gesamtwert werden von TR-Electronic bewusst als vorzeichenlose Werte bereitgestellt.

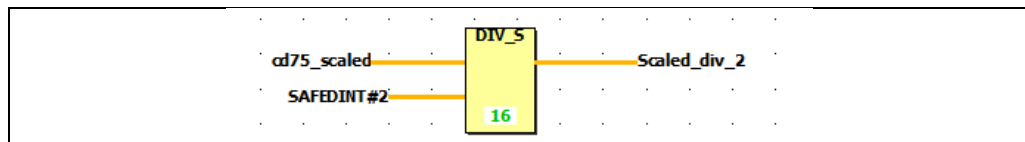
Im SafeDESIGNER können einige Funktionen nicht mit vorzeichenlosen Werten durchgeführt werden. Dazu gehört beispielsweise die Division. Wenn der Anwender eine Division mit einem vorzeichenlosen Wert im SafeDESIGNER durchführen will, muss er vorzeichenlose Werte auf einen vorzeichenbehafteten Wert casten.

10		cd75_multi	SAFEDWORD		SL1.SM5.SafeTRInputMulti
11		cd75_scaled	SAFEDWORD		SL1.SM5.SafeTRInputScaled
12		cd75_speederror	SAFEBOOL		SL1.SM5.SafeSpeedError
13		cd75_pres_status	SAFEBOOL		SL1.SM5.SafePresetStatus
14		cd75_pres_error	SAFEBOOL		SL1.SM5.SafePresetError
15		cd75_preset_ok	SAFEBOOL		SL1.SM5.SafePresetOK
16		cd75_pres_out_single	SAFEDWORD		SL1.SM5.SafeTRPresetSingleturn
17		cd75_pres_out_multi	SAFEDWORD		SL1.SM5.SafeTRPresetMultiturn
18		cd75_pres_out_prep	SAFEBOOL		SL1.SM5.SafePresetPreparation
19		cd75_pres_out_req	SAFEBOOL		SL1.SM5.SafePresetRequest

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 Variablen-tabelle auf `SAFEDINT` umgestellt werden:

10		cd75_multi	SAFEDWORD		SL1.SM5.SafeTRInputMulti
11		cd75_scaled	SAFEDINT		SL1.SM5.SafeTRInputScaled
12		cd75_speederror	SAFEBOOL		SL1.SM5.SafeSpeedError

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 im Legacy-Modus gelten unabhängig davon, ob der Wert vorzeichenbehaftet interpretiert wird oder nicht:

Positionswert 32 Bit unskaliert: 0, 1, 2,... 268435455 → Überlauf auf 0, 1, 2....  
 Singleturn Legacy: 0, 1, 2,...8189, 8190, 8191 → Überlauf auf 0, 1, 2....  
 Multiturn Legacy: 0, 1, 2,...32766, 32767 → 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.

## 5 Download - Softwarebeispiele

Das in diesem Dokument beschriebene Beispielprojekt kann hier heruntergeladen werden:

<https://www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0383>

### 5.1 Legacy-Beispiel-Projekte

Die Beispiel-Projekte der CD\_75M-Familie können hilfreich für die Verwendung des Legacy-Modes sein. Sie können nach wie vor heruntergeladen werden.

Dokumentation zur Inbetriebnahme eines CD\_75M-EPL:

<https://www.tr-electronic.de/f/TR-ECE-TI-DGB-0264>

Beispiel CD\_75M-EPL mit XDD-/OSDD-Gerätebeschreibung ab Automation Studio Versionen V4.5:

[www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0351](http://www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0351)

ETHERNET 

**POWERLINK**  
certified product

open 

**SAFETY**  
certified product

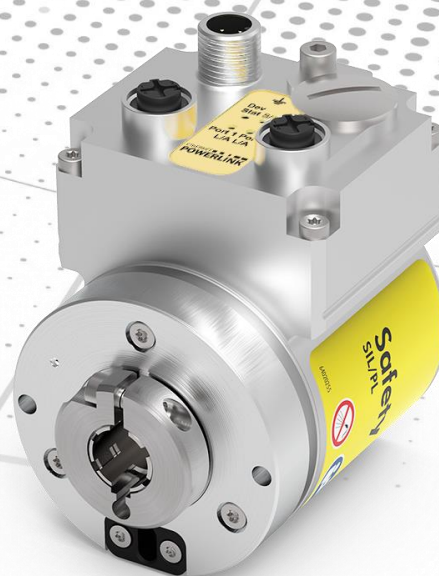
# Absolute Encoder CD\_582-EPL POWERLINK/openSAFETY

Parameterization with B&R control system  
X20 CPU

**CDV582**



**CDH582**



Figures similar

**DIN EN 61508:** SIL CL2 / SIL CL3  
**DIN EN ISO 13849:** PL d / PL e

- Safety Program
- Configuration
- Access to data channels
- Parameterization

**Technical  
Information**

---

## **TR-Electronic GmbH**

D-78647 Trossingen

Eglisshalde 6

Tel.: (0049) 07425/228-0

Fax: (0049) 07425/228-33

Email: [info@tr-electronic.de](mailto:info@tr-electronic.de)

[www.tr-electronic.de](http://www.tr-electronic.de)

---

### **Copyright protection**

This Manual, including the illustrations contained therein, is subject to copyright protection. Use of this Manual by third parties in contravention of copyright regulations is not permitted. Reproduction, translation as well as electronic and photographic archiving and modification require the written consent of the manufacturer. Violations shall be subject to claims for damages.

### **Subject to modifications**

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

### **Document information**

Release date / Rev. date:	08/02/2022
Document rev. no.:	TR-ECE-TI-DGB-0388 v01
File name:	TR-ECE-TI-DGB-0388-01.docx
Author:	MÜJ, DIR

### **Font styles**

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

`Courier` font displays text, which is visible on the display or screen and 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

<b>Contents .....</b>	<b>39</b>
<b>Revision index .....</b>	<b>40</b>
<b>1 General information .....</b>	<b>41</b>
1.1 Applicability .....	41
1.2 Abbreviations and terms used .....	42
<b>2 Safety instructions .....</b>	<b>43</b>
2.1 Definition of symbols and notes .....	43
2.2 Organizational measures .....	44
2.3 Personnel qualification .....	44
2.4 Terms of use of the software examples .....	44
<b>3 Create safety program .....</b>	<b>45</b>
3.1 Notes on the project .....	45
3.1.1 Access protection .....	45
3.1.2 Organizational measures .....	46
3.2 Installing the device description .....	47
3.3 Hardware configuration .....	48
3.4 Parameter setting of grey actual values .....	51
3.5 Secondary interface (optional) .....	52
3.6 Create openSAFETY project (TR profiles) .....	53
3.6.1 Safe parameterization .....	54
3.6.2 Creating the safety program .....	55
3.6.3 Generating the safety program .....	56
3.6.4 Loading the safety program .....	56
3.6.5 Testing the safety program .....	58
3.7 Create openSAFETY project (TR legacy) .....	59
<b>4 Extending the safety program - application examples .....</b>	<b>61</b>
4.1 Preset execution .....	62
4.2 Safe speed (SLS) .....	66
4.2.1 Code example SLS .....	66
4.3 Safe standstill detection .....	67
4.3.1 Code example safe standstill detection .....	67
4.4 Safe position detection .....	68
4.4.1 Code example for the acquisition of a safe position range .....	69
4.5 Safe direction detection .....	70
4.5.1 Code example for safe direction detection .....	70
4.6 Type conversion of the measuring system - actual values in SafeDESIGNER .....	71
<b>5 Download - Software examples .....</b>	<b>72</b>
5.1 Legacy sample projects .....	72

## Revision index

<b>Modification</b>	<b>Date</b>	<b>Index</b>
First edition for CD_582M-EPL	06/10/2022	00
Document finalized and translation added	08/02/2022	01

---

## 1 General information

This "Technical Information" contains the following topics:

- Creating a safety program
- Setting the safe parameters
- Using the safety-related data channel.

The "Technical Information" can be requested separately.

### 1.1 Applicability

This "Technical Information" applies exclusively to the following measuring system series with **POWERLINK** interface and **openSAFETY** profile in connection with a B&R X20 controller:

- CD\_582M-EPL

The products are identified by affixed nameplates and are part of a system.

The following documentation therefore applies together:

- B&R: X20 System User Manual,  
Order no.: MAX20-GER
- B&R: Integrated Safety Technology user manual,  
Order no.: MASAFETY-GER
- B&R: X20 data sheet MASAFETY-X20SLXxxx-GER\_V1141
- Plant-specific operating instructions of the operator
- Safety manual [TR-ECE-BA-GB-0142](#)
- interface-specific user manual [TR-ECE-BA-GB-0169](#)
- and this optional "Technical Information"

### 1.2 Abbreviations and terms used

Name	Description
AS	Automation Studio from B&R: Development environment for POWERLINK projects
B&R	Bernecker+Rainer Industrie-Elektronik GmbH
EPL	Ethernet PowerLink
FB(s)	Function block(s)
FBL	Programming language, function block language
FU(s)	Function(s)
gay data	single-channel actual values via POWERLINK, not safety-related
KOP	programming language, ladder diagram
openSAFETY	open and bus-independent safety standard for all Industrial Ethernet solutions
OSDD	openSAFETY Device Description
POWERLINK CN	Controlled Node: Slave device in a POWERLINK network
POWERLINK MN	Managing Node: Initiates the POWERLINK communication and parameterizes the CN
SCM	Safety Configuration Manager: Initializes the SNs in an openSAFETY project via the underlying fieldbus
SL	SafeLogic CPU from B&R
SN	Safety Node: Slave participant in an openSAFETY project
XML	EXtensible Markup Language

## 2 Safety instructions

### 2.1 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 can occur if the required precautions are not met.

---



**CAUTION** means that minor injuries can occur if the required precautions are not met.

---

---

**NOTICE**

---

means that damage to property can occur if the required precautions are not met.

---



indicates important information or features and application hints for the product used.

---

### 2.2 Organizational measures

Personnel handling the measuring system must have read and understood the Safety Manual [TR-ECE-BA-GB-0142](#), in particular chapter "Basic safety instructions" prior to commencing work.

### 2.3 Personnel qualification

The configuration of the measuring system may only be performed by qualified personnel, see B&R user manual.

### 2.4 Terms of use of the software examples

---

#### **WARNING**

TR-Electronic GmbH assumes no liability and no warranty for the error-free functioning of the safety program and the application examples.

#### **NOTICE**

The software examples offered for download are for demonstration purposes only, the usage by the operator is at his own risk.

---



The password for the examples is always "abc123".  
This also applies to the zip archive of the software examples.

---

---

## 3 Create safety program

In the following the sample project of TR-Electronic is explained. This includes installing the device description, projecting the modular TR device and executing sample code for preset and other safe functions.



At least Automation Studio V4.5 or higher is required to operate the 582 encoder family, since older versions do not support the installation of a safe device description (osdd).

---

### 3.1 Notes on the project

This chapter describes the procedure for creating a sample safety program using B&R's `Automation Studio` configuration software and the `SafeDESIGNER` option package.

The safety program is created in `SafeDESIGNER`, which contains a code editor for developing the program for the safety controller. This is done using the FBS and KOP graphical programming languages.

The safety program is executed on an `X20 SafeLOGIC-X (X20SLX806)`. The SLX is directly connected to the POWERLINK MN and is itself the openSAFETY SCM.

An `X20CP482` is used as the MN at POWERLINK. It contains a POWERLINK application that can also evaluate the measuring system information (including the actual values "grey channel").

#### 3.1.1 Access protection

Access to the `SafeDESIGNER` project is secured by a password query. There is one password each for the development and for the commissioning. The access to the `SLX806`, e.g. for programming, is also protected by a password.



The password is: `abc123`

---

### 3.1.2 Organizational measures

---

#### **⚠ WARNING**

***Danger of overriding the fail-safe function due to improper project planning of the safety program!***

- The safety program may only be created in conjunction with the system documentation supplied by B&R for the software or hardware.
- The following descriptions refer to the pure process, without taking into account all instructions from the B&R manuals.

The information and notes given in the B&R manuals, especially the safety instructions and warnings, must therefore be observed and complied with.

- The shown project planning is an example. The user is therefore obliged to check and adapt the usability of the project planning for his application. This also includes the selection of suitable safety-related hardware components and the necessary software requirements.

---

#### **Software components used for the configuration example:**

- Automation Studio V4.10.2.37
- SafeDESIGNER V4.3.3.7



The use of subsequent versions is also possible.

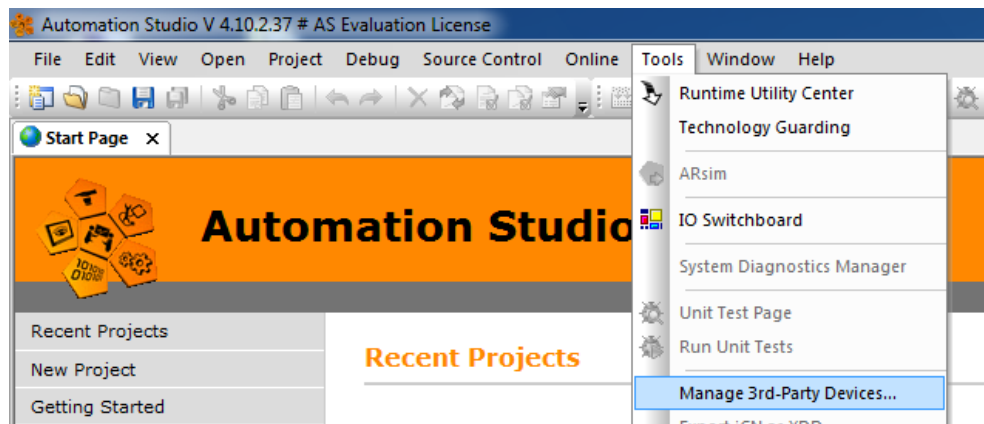
---

#### **Hardware components used for the configuration example:**

- POWERLINK-MN: X20CP482 with X20PS9600
- openSAFETY SCM: X20SLX806

## 3.2 Installing the device description

To project the modular measuring system, the corresponding device descriptions have to be installed first. This includes a head station and pluggable modules. The management of the device descriptions can be found in the menu under the item "Manage 3rd-Party Devices...".



In the so-called "external device manager", the device description files are now installed in the following order ("Import fieldbus device(s)"):

1. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_Head.xdd
2. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_TR-Profile.xosdd  
0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_TR-Profile.xdd
3. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_Legacy.xosdd  
0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_Legacy.xdd
4. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_ITF2\_SSI.xdd
5. 0x0000025C\_TR-Electronic\_CD\_582\_-EPL\_ITF2\_Incr.xdd



For the safety-related modules, the xosdd file is always selected first, followed by the corresponding xdd file.

After installing the device write files, the head-end station and the modules appear in the list of the third-party device manager:

Automation Studio 3rd-Party Device Manager						
This dialog allows you to manage 3rd-party fieldbus- and DTM devices.						
Name	Version	Vendor	DTM	Protocol(s)	Descript	
TR-CD582 Slot ITF2 Incremental	1.00	TR Electronic GmbH		module_interface	TR-Elec	
TR-CD582 Slot ITF2 SSI	1.00	TR Electronic GmbH		module_interface	TR-Elec	
TR-CD582 Slot Legacy	1.00	TR Electronic GmbH		module_interface	TR-Elec	
TR-CD582 Slot TR-Profile	1.00	TR Electronic GmbH		module_interface	TR-Elec	
TR-CD582-EPL	1.00	TR Electronic GmbH		Powerlink	TR-Elec	

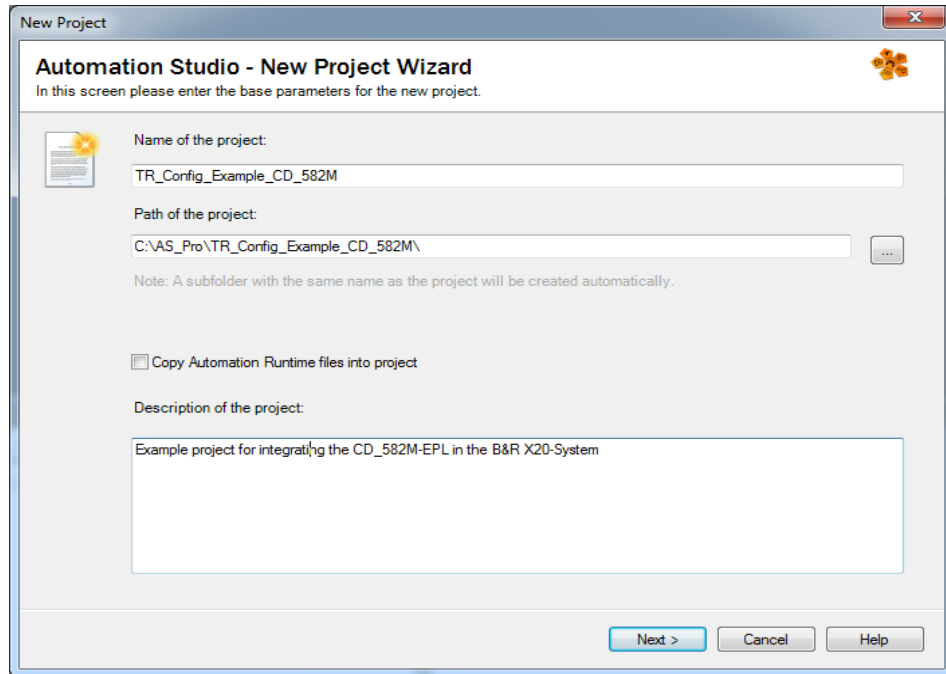
Subsequently, the creation of the project can be started.



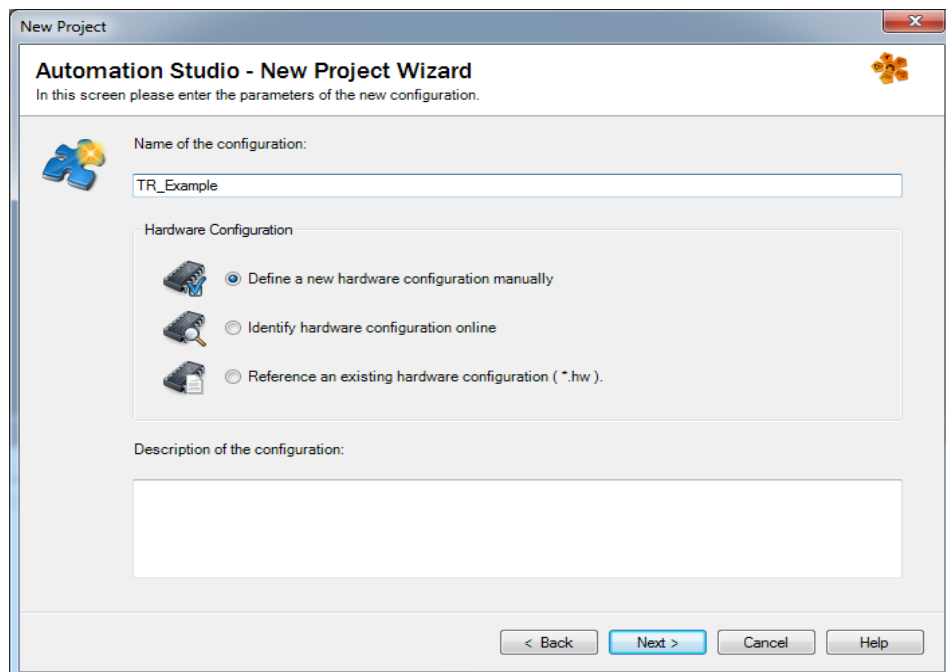
To configure newly installed foreign device descriptions, it may be necessary to reopen the currently opened project or to restart Automation Studio.

## 3.3 Hardware configuration

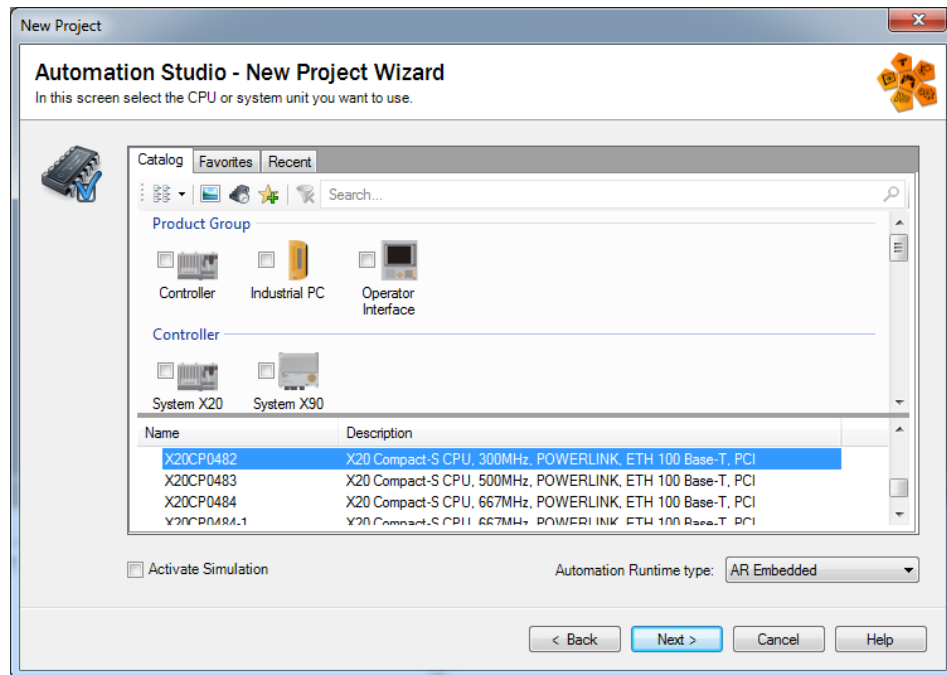
- Start Automation Studio and create a new project.



- A new hardware configuration will be created:



- Finally, the correct CPU must be selected: X20CP0482



- Select Safety Release 1.10 under the menu item -> Project -> Change Runtime Versions.... In this example, version D4.52 is used as Automation Runtime.



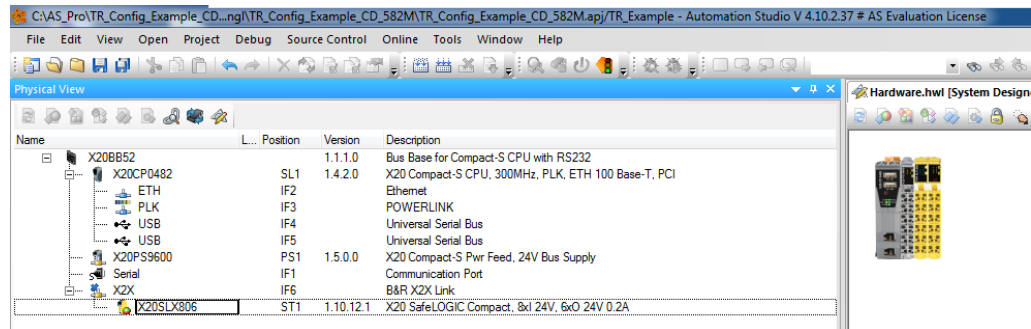
Safety Release 1.10 is used for the measuring system in the example. Alternatively, the safe application can be created via mappSafety.

X20CP0482

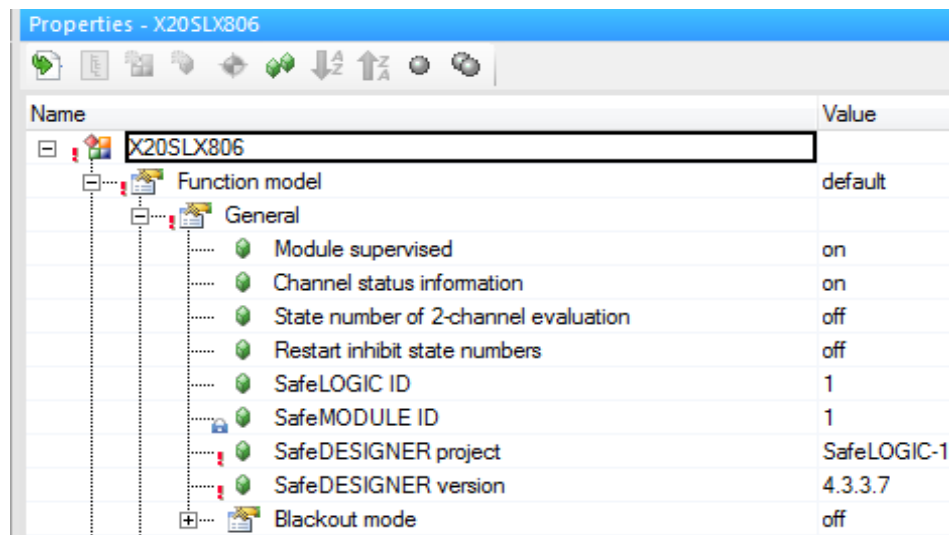
Component	Preferred	In use	Scope
Automation Runtime	D4.52	D4.52	
Visual Components	not defined	not defined	
mapp Motion	not defined	not defined	
ACP10 ARNC0 (Motion)	not defined	not defined	
mapp Safety	not defined	not defined	
mapp Vision	not defined	not defined	
mapp Services	not defined	not defined	
mapp View	not defined	not defined	
mapp Control	not defined	not defined	
mapp Cockpit	not defined	not defined	
Safety Release	1.10	1.10	

## Create safety program

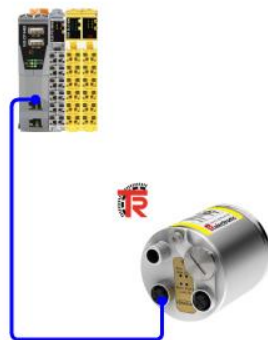
- In the Physical View: Add the SCM to the EPL network by dragging the corresponding device from the hardware catalog to the POWERLINK interface:



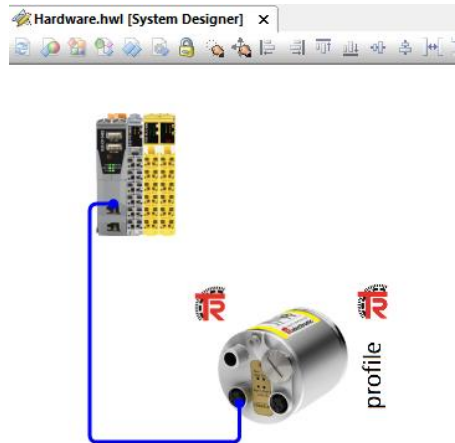
- The SafeDESIGNER version can be selected under the configuration of the SafeLOGIC:



- Now the TR measuring system can be added. We start with the non-safe head station (only visible if "Safety" filter is not switched on in the hardware catalog) "TR-CD582-EPL" and connect the POWERLINK interface to the controller:



- Subsequently, a module can be linked to the measuring system. In the example, the TR profile "TR-CD582 Slot TR-Profile" is selected:



- In the example the POWERLINK Node-ID 18 is used for the measuring system. This completes the hardware configuration. The project is displayed in the Physical View as follows:

Name	Location	Position	Version	Description
X20BB52			1.1.1.0	Bus Base for Compact-S CPU with RS232
X20CP0482		SL1	1.4.2.0	X20 Compact-S CPU, 300MHz, PLK, ETH 100 Base-T, PC
ETH		IF2		Ethernet
PLK		IF3		POWERLINK
TR-CD582-EPL		ST1	1.00	TR-Electronic CD_582M - POWERLINK / openSAFETY H
module_interface		IF1		
TR-CD582_Slot_TR-Profile		ST1	1.00	TR-Electronic CD_582M - POWERLINK / openSAFETY (T
USB		IF4		Universal Serial Bus
USB		IF5		Universal Serial Bus
X20PS9600		PS1	1.5.0.0	X20 Compact-S Pwr Feed, 24V Bus Supply
Serial		IF1		Communication Port
X2X		IF6		B&R X2X Link
X20SLX806		ST1	1.10.12.1	X20 SafeLOGIC Compact, 8x1 24V, 6x0 24V 0.2A

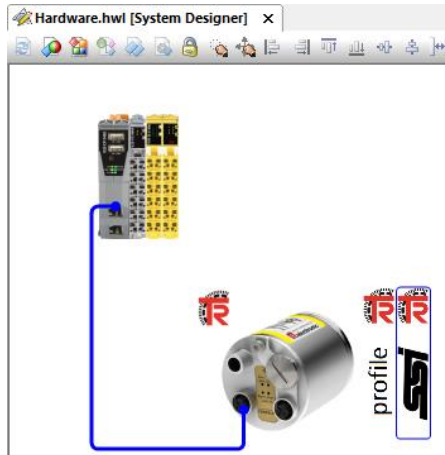
### 3.4 Parameter setting of grey actual values

The parameters for the single-channel grey actual value can be set completely independently of the safety parameters for the TR profile. The setting is done in the properties of the TR profile slot:

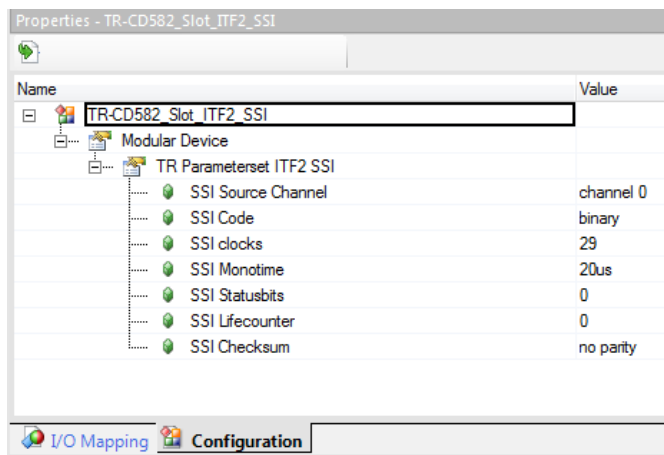
Name	Value
<b>TR-CD582_Slot_TR-Profile</b>	
General	
SafeDOMAIN ID	1
SafeNODE ID	2
Modular Device	
TR Parameterset Profile	
Counting direction	forward
Measuring length	536870912
Rotations Numerator	65536
Rotations Denominator	1
Speed unit	rpm * factor
Speed factor	1
Speed time	100
Speed filter depth	0
Speed filter dynamic	Filter static
remote substitute	grey substitute
coupled to savesensor	not coupled to safe

### 3.5 Secondary interface (optional)

For measuring systems with a secondary interface, this can also be parametrized in Automation Studio. To do this, the corresponding secondary interface must also be plugged in (in this example, "TR-CD582 Slot ITF2 SSI"):



Subsequently, the secondary interface parameters can be set under the properties of the SSI slot:



The secondary interface does not have any cyclic fieldbus data. The controller only transfers the parameters for the secondary interface during startup.



The secondary interface parameters are stored by the measuring system. Once the measuring system has been parameterized, the second interface starts with the correct parameters even without POWERLINK communication.

### 3.6 Create openSAFETY project (TR profiles)

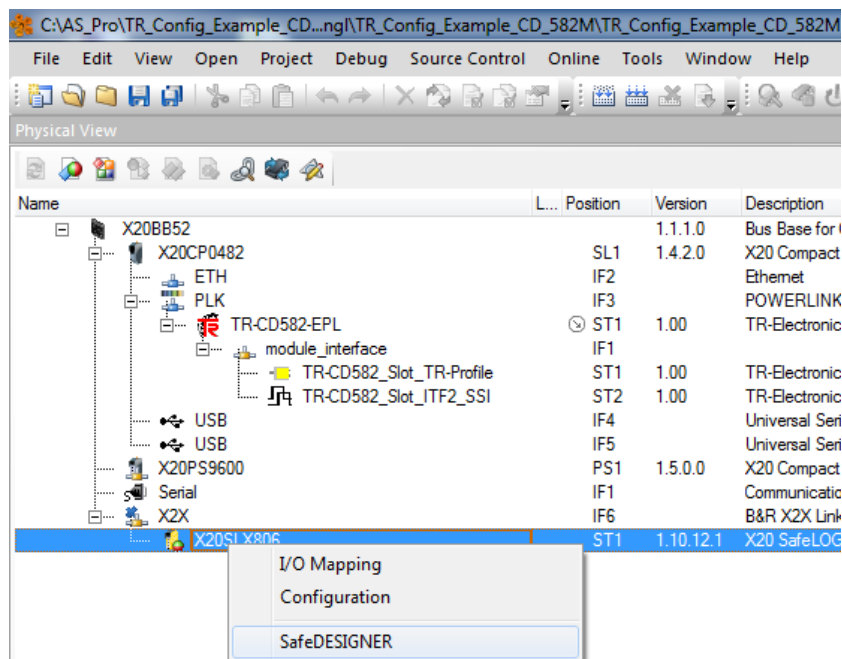
In this example in the openSAFETY network, the SafeMODULE ID = 2 was assigned for the TR measuring system (Slot TR-Profile).

Properties - TR-CD582_Slot_TR-Profile	
Name	Value
TR-CD582_Slot_TR-Profile	
General	
SafeDOMAIN ID	1
SafeNODE ID	2
Modular Device	
TR Parameterset Profile	
Counting direction	forward
Measuring length	536870912



The Safety project was automatically created by Automation Studio and linked to the SLX.

- Start SafeDESIGNER by right-clicking on the SafeLOGIC:



- Assign and confirm passwords for the SafeDESIGNER project. Here in the example "abc123".

In the Safety View (bottom left by default) the openSAFETY participants are listed. The SCM (SL1.SM1) and the measuring system (SL1.SM2) are displayed. The "2" corresponds to the SafeMODULE ID. Under the measuring system all input and output data of the measuring system available in the SafeDESIGNER are listed:

ID	Name	Variable	Type	CPU Variable	Slot
SL1					
SL1.SM1	X20SLX806				IF6.ST1
SL1.SM2	TR-CD582_Slot_TR-Profile				IF3.ST1.IF1.ST1
General					
SafeModuleOk			SAFEBOOL		
InputSafeChannels					
SafeSpeedError			SAFEBOOL		
SafeErrorAckRequest			SAFEBOOL		
SafePresetOK			SAFEBOOL		
SafePresetError			SAFEBOOL		
SafeState			SAFEBOOL		
SafePresetActive			SAFEBOOL		
SafePresetLocked			SAFEBOOL		
SafeScalingError			SAFEBOOL		
SafeTRStatus2			SAFEUSINT		
SafeTRInputPos			SAFEDWORD		
SafeTRInputVel			SAFEDINT		
OutputSafeChannels					
SafePresetPreparation			SAFEBOOL		
SafePresetRequest			SAFEBOOL		
SafeTRPresetValue			SAFEDWORD		

The green dots mark input data from the controller's point of view, the red dots mark output data from the controller's point of view.

### 3.6.1 Safe parameterization

The safe parameters can be set via the SafeDESIGNER user interface. The measuring system - specific parameters are located under the Generic Parameters tab:

ID	Name	Variable	Type
SL1			
SL1.SM1	X20SLX806		
SL1.SM2	TR-CD582_Slot_TR-Profile		
General			SAFEBOOL
SafeModuleOk			SAFEBOOL
InputSafeChannels			SAFEBOOL
SafeSpeedError			SAFEBOOL
SafeErrorAckRequest			SAFEBOOL
SafePresetOK			SAFEBOOL
SafePresetError			SAFEBOOL
SafeState			SAFEBOOL
SafePresetActive			SAFEBOOL
SafePresetLocked			SAFEBOOL
SafeScalingError			SAFEBOOL
SafeTRStatus2			SAFEUSINT
SafeTRInputPos			SAFEDWORD
SafeTRInputVel			SAFEDINT
OutputSafeChannels			SAFEBOOL
SafePresetPreparation			SAFEBOOL
SafePresetRequest			SAFEBOOL

Parameter	Value	Unit
<b>GenericParameters</b>		
SIL	SIL2	
VIT Rotary Direction	Forward	
Measuring Length	536870912	none
Rotations Numerator	65536	none
Rotations Denominator	1	none
Speed Format	rot per min * factor	
Speed Factor	1	none
Speed Time	100	none
Speed Filter Type	static	
Speed Filter Depth	0	none
Window Increments	1000	none

The individual values are described in the user manual. The safe parameters are automatically transferred from the SCM to the measuring system (SN) during startup.

### 3.6.2 Creating the safety program

A safe application that runs on the SCM is created in the editor of the SafeDESIGNER. The following steps show the structure of a program that first outputs the `SafeModuleOk` and `SafeState` status bits via two digital outputs of the SLX806.

- Declare the required variables. Use the toolbar or <Ctrl>+<G> to open the Global Declarations interface in SafeDESIGNER. Create the new variables with a right click via the context menu -> New variable.
- Link the created variables with the corresponding values from the measuring system. Drag the corresponding variable with the help of the mouse from the Safety View to the terminal field of the corresponding variable.

14		SM2_SafeState	SAFEBOOL		SL1.SM2.SafeState
15		SM2_SafeTRInputVel	SAFEDINT		SL1.SM2.SafeTRInputVel
16		SM2_SafeModuleOk	SAFEBOOL		SL1.SM2.SafeModuleOk
17		SM2_SafePresetOK	SAFEBOOL		SL1.SM2.SafePresetOK
18		SM2_SafePresetError	SAFEBOOL		SL1.SM2.SafePresetError
19		SM2_SafeSpeedError	SAFEBOOL		SL1.SM2.SafeSpeedError
20		SM2_SafeTRInputPos	SAFEDWORD		SL1.SM2.SafeTRInputPos
21		SM2_PresetPreparation	SAFEBOOL		SL1.SM2.SafePresetPreparation
22		SM2_PresetRequest	SAFEBOOL		SL1.SM2.SafePresetRequest
23		SM2_PresetValue	SAFEDWORD		SL1.SM2.SafeTRPresetValue

- Link variables to the outputs of the SLX:

1		NewGroup			
2		SafeDigitalOutput01	SAFEBOOL		SL1.SM1.SafeDigitalOutput01
3		SafeDigitalOutput02	SAFEBOOL		SL1.SM1.SafeDigitalOutput02

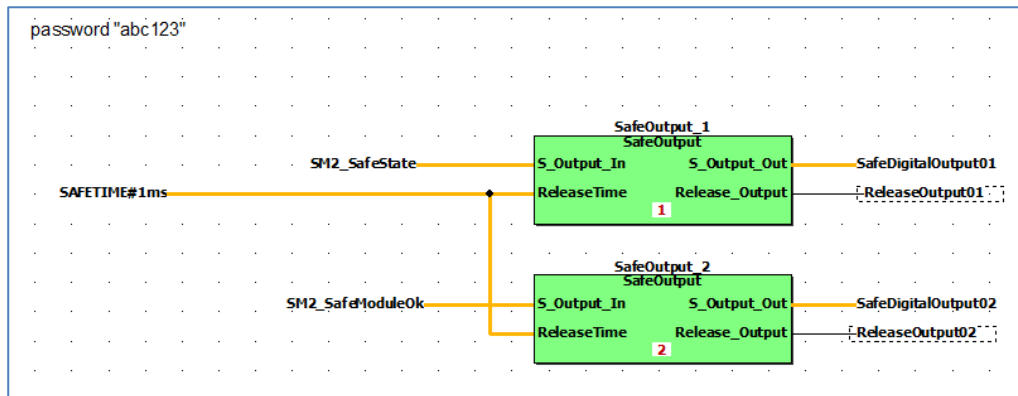


The digital outputs of the SLX must be set to "via SafeLogic" in Automation Studio under the SLX settings.

Properties - X20SLX806	
Name	Value
X20SLX806	
Function model	default
General	
Output signal path	
DigitalOutput01	via SafeLOGIC
DigitalOutput02	via SafeLOGIC
DigitalOutput03	via SafeLOGIC
DigitalOutput04	via SafeLOGIC
DigitalOutput05	via SafeLOGIC
DigitalOutput06	via SafeLOGIC
SafeDESIGNER to SafeLOGIC communication	
Activate SPROXY	no

## Create safety program

- Go to the Code: Main. Here the actual application is programmed graphically:



The creation of the sample application is now completed. The digital outputs 1 and 2 of the SLX806 are set according to the measuring system status bits.

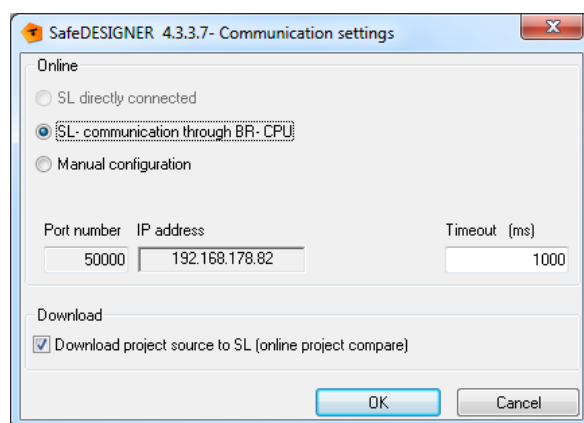
### 3.6.3 Generating the safety program

To create the safety program, the sample application just created must be saved and compiled. To do this, the Compile button in SafeDESIGNER or the <F9> key must be pressed.

### 3.6.4 Loading the safety program

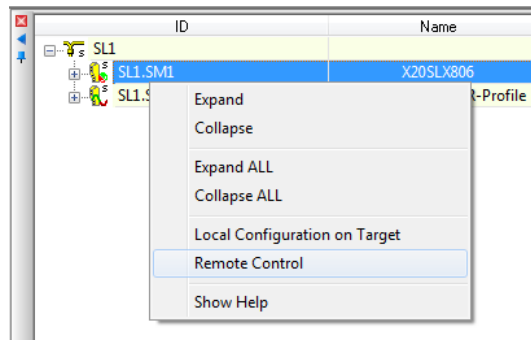
To perform the following steps, the project must first be compiled from Automation Studio and programmed in the X20CP0482 (=POWERLINK project).

After the safety program has been generated, it can be loaded into the SCM. For this, a connection to the SCM must exist. The connection type can be set under the Online menu item in SafeDESIGNER. If Ethernet communication to the POWERLINK CPU was set up in advance and the POWERLINK CPU can be reached via Ethernet, the setting SL - communication through BR-CPU should be used:



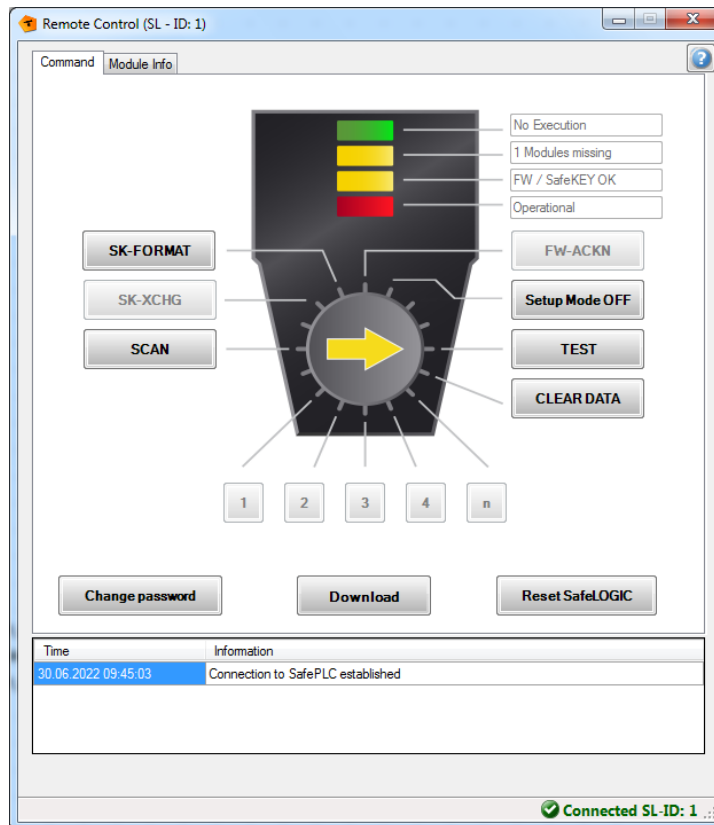
SafeDESIGNER uses the port number and the current online configuration from Automation Studio as the basis for online communication.

The "Remote Control" of the SLX can be used for loading the software. It is started via the Safety View:



The project is loaded into the SLX via the "Download" button.

After the download the SafeLogic restarts. Now firmware and new stations must be acknowledged at the SLX. For the single acknowledgements the remote control is also used for these acknowledgements:



After acknowledging the new firmware (FW-ACKN) and the new stations (1...n), the system starts up and the measuring system switches to safe operation. Now the two status LEDs of the measuring system light up green: EPL operational and openSAFETY operational.

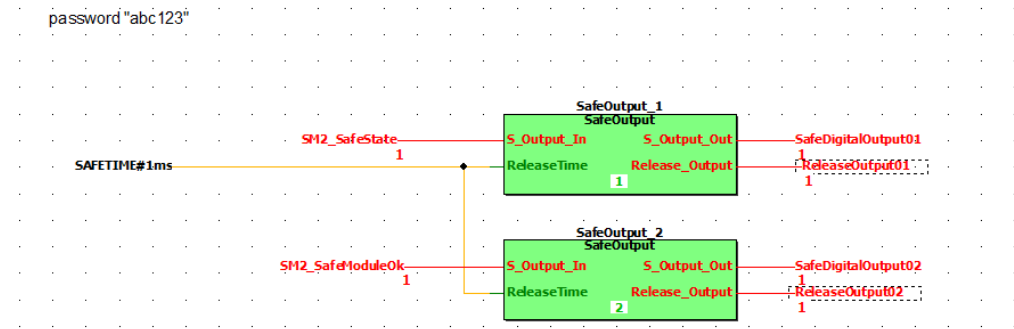


Since the flashing patterns and indications may differ slightly depending on the procedure and preinstalled software, the SL manual must also be consulted for this procedure.

### 3.6.5 Testing the safety program

After creating the safety program, a complete function test must be performed according to the automation task.

In SafeDESIGNER an online connection can be established with the Variable status button or by pressing the <F10> key, during which the current variable values are displayed in SafeDESIGNER.



The measuring system does not output the current actual values until the startup via openSAFETY and the safe parameterization have taken place. Until then all safe variables are set to the value 0.

All actual values (grey and safe) of all measuring systems - IO data are also visible in Automation Studio's online mode; they are displayed under the I/O assignment of the measuring system:

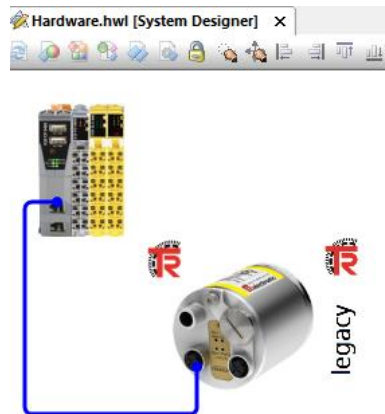
Properties - TR-CD582_Slot_TR-Profile	
Channel Name	Physical Value
TR_Grey_Input_I0000_S1E_Cyc_In_Status	0
TR_Grey_Input_I0000_S1E_Cyc_In_Position	0
TR_Grey_Input_I0000_S1E_Cyc_In_Velocity	0
TR_Grey_Output_I0000_S20_Cyc_Out_Control	0
TR_Grey_Output_I0000_S20_Cyc_Out_Preset	0
SafeSpeedError	FALSE
SafeErrorAckRequest	FALSE
SafePresetOK	FALSE
SafePresetError	FALSE
SafeState	FALSE
SafePresetActive	FALSE
SafePresetLocked	FALSE
SafeScalingError	FALSE
SafeTRStatus2	0
SafeTRInputPos	0
SafeTRInputVel	0

In the figure above, the variable names of the two-channel data start with the word Safe and can also be used in the "grey" area.

The "grey" single-channel values can be used in the POWERLINK project. They are statically mapped and always available.

### 3.7 Create openSAFETY project (TR legacy)

If the functionalities of the TR profile (velocity units, averaging, scaling, preset-on-the-fly etc.) are not required and the user wants the CD\_582M-EPL measuring system to behave like the CD\_75M-EPL measuring system, operation in the so-called legacy mode is possible. The parameterization in the legacy format is done in the same way as for the TR profile in the previous chapter. Only another module is plugged in ("TR-CD582 Slot Legacy" instead of "TR-CD582 Slot TR-Profile"):



If the legacy module is plugged in the cyclic data are shown in legacy format.

Grey cyclic data in Automation Studio:

Properties - TR-CD582_Slot_Legacy		
Channel Name	Process Variable	Data Type
TR_Grey_Input_I0000_S1E_Cyc_In_Velo_Overflow		USINT
TR_Grey_Input_I0000_S1E_Cyc_In_Velo_16		INT
TR_Grey_Input_I0000_S1E_Cyc_In_Multitum		UINT
TR_Grey_Input_I0000_S1E_Cyc_In_Singletum		UINT
TR_Grey_Input_I0000_S1E_Cyc_In_Scaled		UDINT
SafeSpeedError		BOOL
SafePresetStatus		BOOL
SafePresetError		BOOL
SafePresetOK		BOOL
SafeState		BOOL
SafeTRInputVel		INT
SafeTRInputPosMulti		UINT
SafeTRInputPosSingle		UINT
SafeTRInputPosScaled		UDINT

Safety View in Safe Designer for legacy mode:

ID	Name	Vari...	Typ
SL1			
SL1.SM1	X20SLX806		
SL1.SM2	TR-CD582_Slot_Legacy		
General			
SafeModuleOk			SAFEBOOL
InputSafeChannels			
SafeSpeedError			SAFEBOOL
SafePresetStatus			SAFEBOOL
SafePresetError			SAFEBOOL
SafePresetOK			SAFEBOOL
SafeState			SAFEBOOL
SafeTRInputVel			SAFEINT
SafeTRInputMulti			SAFEWORD
SafeTRInputSingle			SAFEWORD
SafeTRInputScaled			SAFEWORD
OutputSafeChannels			
SafePresetPreparation			SAFEBOOL
SafePresetRequest			SAFEBOOL
SafeTRPresetMultiturn			SAFEWORD
SafeTRPresetSingleturn			SAFEWORD

Safe parameters in the Safe Designer for Legacy Mode:

Model no.:	<b>TR-Electronic_CD_582_-EPL_Legacy</b>	
Description:	<b>TR-Electronic CD_582M - POWERLINK / openSAFETY (TR-Child)</b>	
SafeMODULE ID:	2	
Import file:	-	
Parameter	Value	
<b>GenericParameters</b>		
Integration Time	2	n
Window Increments	1000	n
Idleness Preset Tolerance	1	n
VIT Rotary Direction	Forward	
<span>Basic</span> / <span>Safety Response Time</span> / <span>Basic</span> / <b><span>GenericParameters</span></b> / <span>ALL</span>		

The data types and data contents available here correspond to the data of the CD\_75M-EPL family. This makes it possible to reuse existing control software parts that were created with the CD\_75M-EPL family.

## 4 Extending the safety program - application examples

The safety program created in chapter 3 is extended in the following sections by functional examples for the use of the measuring system actual values in SafeDESIGNER.

However, the examples do not represent customer-specific solutions, but are only intended to provide assistance for different automation tasks.

With the help of the presented function blocks the integration of the measuring system into an application should be simplified.

For some of the following application examples

- Safe speed (SLS)
- Safe standstill detection
- Safe position detection
- Safe direction detection

the error states are output by the function blocks presented here. The associated error handling is not part of the examples and must be implemented by the user.

In the application example for “Preset execution”, the function block does not output a direct error state, since the error information is generated by the measuring system itself.



Observe the “Terms of use of the software examples” on page 44!



The examples presented here refer to the TR profile. Examples for the legacy format are not listed separately here, as these examples are already included in the commissioning documentation for the CD\_75M-EPL family.

---

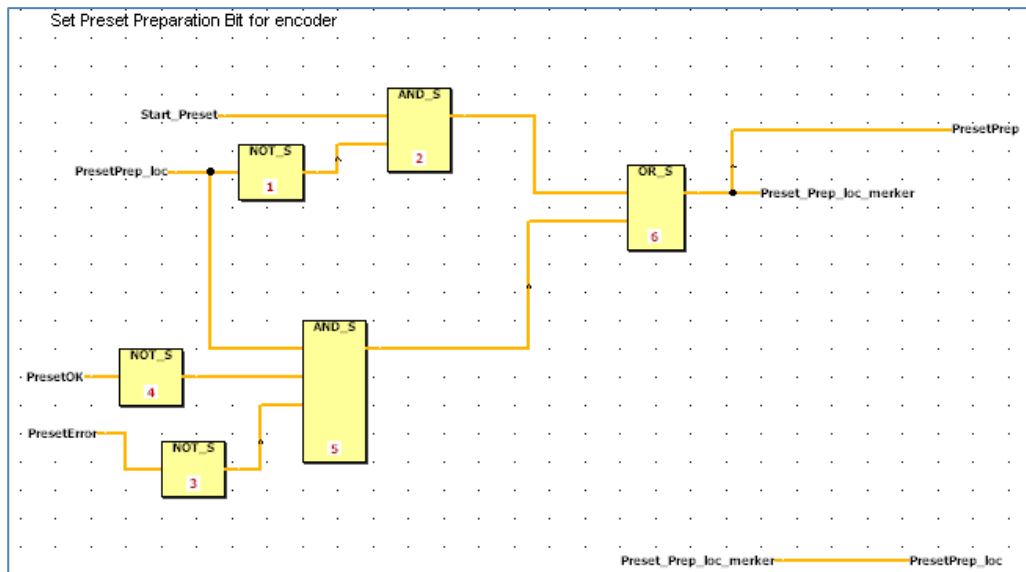
### 4.1 Preset execution

First a new function block is created in SafeDESIGNER (New Function Block after right-clicking on Logical POU's in the Project Tree Window). To the new function block the name TR\_PRESET\_Example is assigned. Then the following local variables are created for the function block:

	Name	Data type	Usage	Description	Init	Diag	Feedback
1	[-] NewGroup						
2	Start_Preset	SAFEBOOL	VAR_INPUT	Preset Start	SAFEBOOL#0		<input type="checkbox"/>
3	PresetReq	SAFEBOOL	VAR_OUTPUT	Set Preset Request	SAFEBOOL#0	<input type="checkbox"/>	<input type="checkbox"/>
4	PresetPrep_loc	SAFEBOOL	VAR	Set Preset Prep local	SAFEBOOL#0		<input checked="" type="checkbox"/>
5	PresetOK	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		<input type="checkbox"/>
6	PresetError	SAFEBOOL	VAR_INPUT		SAFEBOOL#0		<input type="checkbox"/>
7	PresetFinish	SAFEBOOL	VAR_OUTPUT	Preset finished	SAFEBOOL#0	<input type="checkbox"/>	<input type="checkbox"/>
8	PresetFinish_loc	SAFEBOOL	VAR		SAFEBOOL#0		<input checked="" type="checkbox"/>
9	PresetFinish_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		<input type="checkbox"/>
10	PresetPrep	SAFEBOOL	VAR_OUTPUT	Set Preset Preparation	SAFEBOOL#0	<input type="checkbox"/>	<input type="checkbox"/>
11	Preset_Prep_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		<input type="checkbox"/>
12	PresetReq_loc	SAFEBOOL	VAR		SAFEBOOL#0		<input checked="" type="checkbox"/>
13	Reset_Lock	SAFEBOOL	VAR_INPUT	Lock Reset	SAFEBOOL#0		<input type="checkbox"/>
14	PresetReq_loc_merker	SAFEBOOL	VAR		SAFEBOOL#0		<input type="checkbox"/>

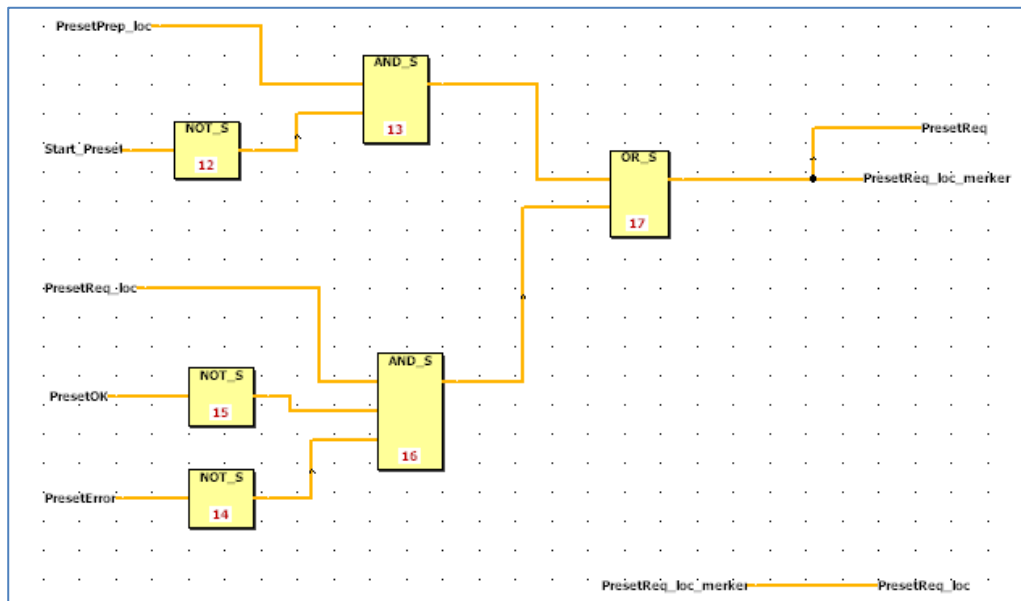
The following illustrations show the structure of the new function block TR\_PRESET\_Example:

#### 1. Setting of Preset Preparation



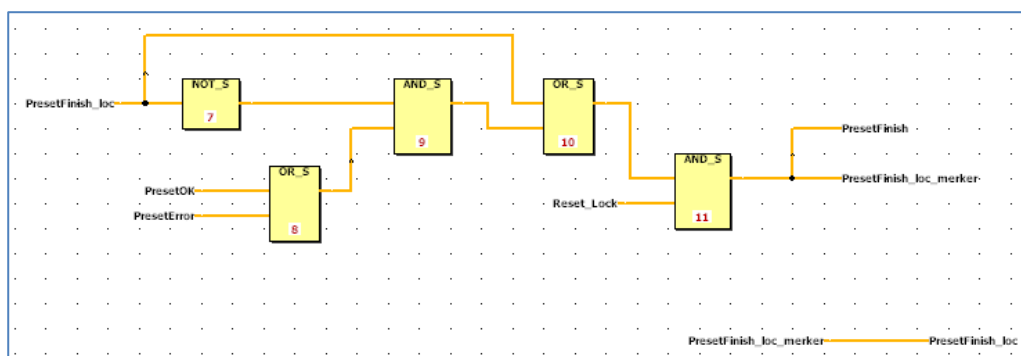
In the first section, setting the Start\_Preset input sets the measuring system output signal PresetPrep bit. The signal remains set until the preset process in the measuring system is completed. If the preset process is executed without error, the measuring system sets PresetOK and in case of error PresetError.

## 2. Setting of Preset Request



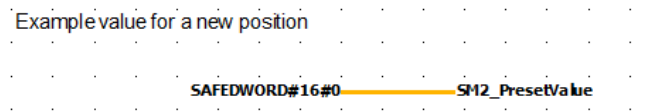
If the PresetPrep output signal to the measuring system is set and the Start\_Preset signal is set to FALSE again, the PresetReq signal is set. The signal remains set until the preset process in the measuring system is completed. If the preset process could be executed without error, the measuring system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

## 3. Setting of Preset Finish

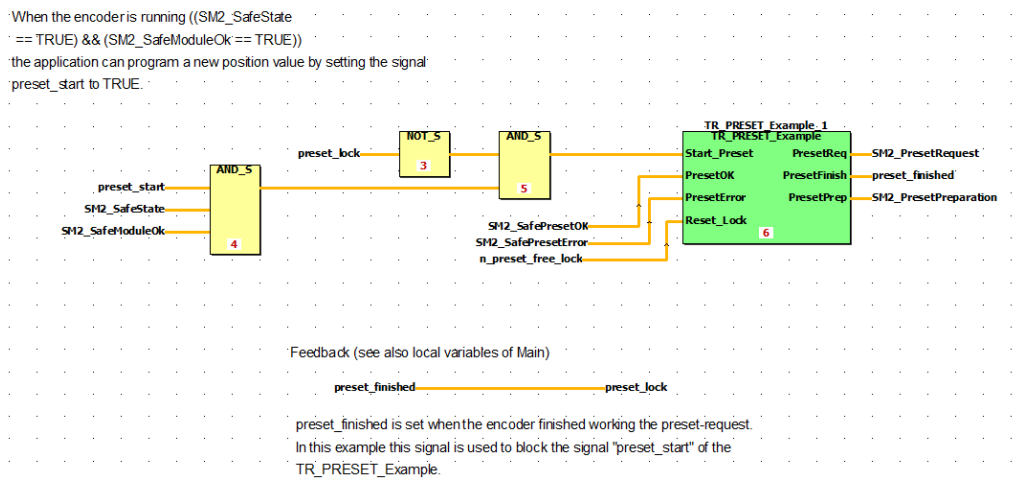


In this section the information is stored that the execution was terminated. Reset\_Lock must always be set to "1" for this, since the variable is low-active.

To start the preset, a new signal with the name `preset_start` is created, which can be connected to a digital input for example. The main function block is now supplemented so that the value 0 is sent to the measuring system as a new fixed preset:



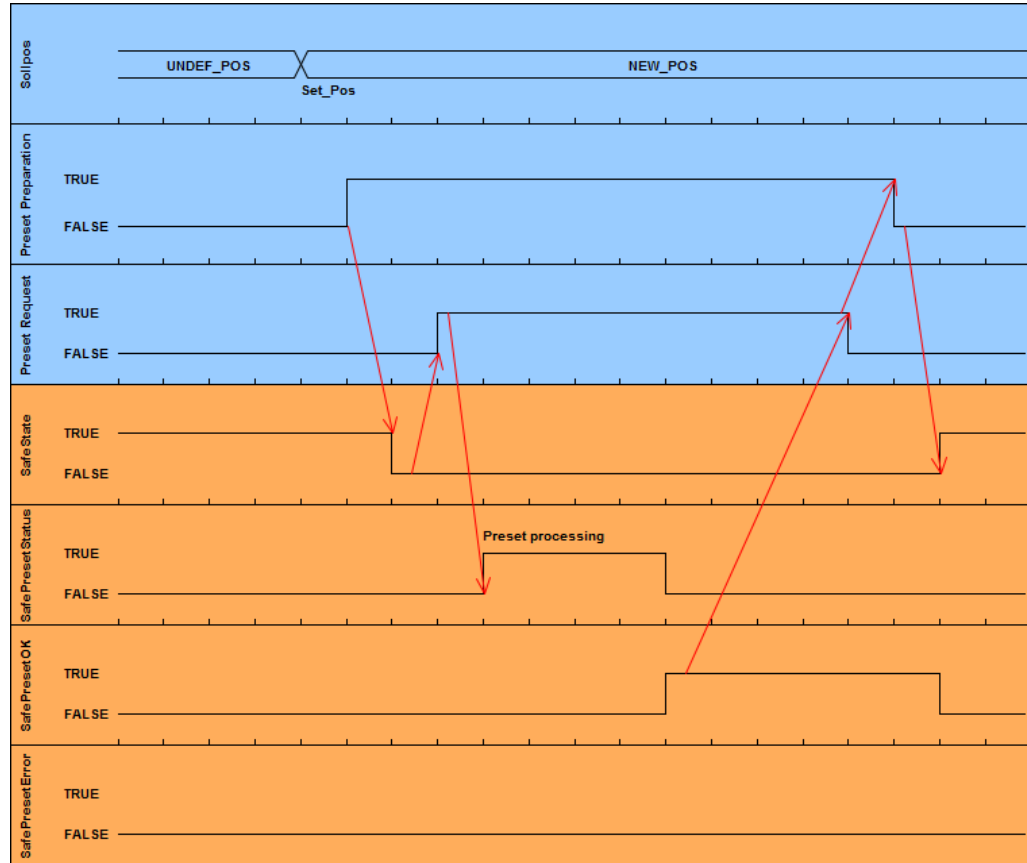
The `PresetPrep` and `PresetReq` outputs towards the measuring system are connected to the corresponding outputs of the `TR_PRESET_Example` function block. With the help of a lock signal the preset is executed only once per `Start_Preset` signal: When the `preset_lock` signal is not yet set and the `preset_start` signal is set. Thus with the present program a preset is executed after the run-up to the new setpoint 0 as soon as the signal `preset_start` is set to `TRUE` by the user.



In order to execute another preset, the signal `n_preset_free_lock` must first be set to `FALSE` again in this example (start value = `TRUE`). This causes the output signal `preset_finished` to be reset. For a further PRESET, `preset_start` can now be set to `TRUE` again.

The following timing diagram shows the error-free sequence of the preset function.

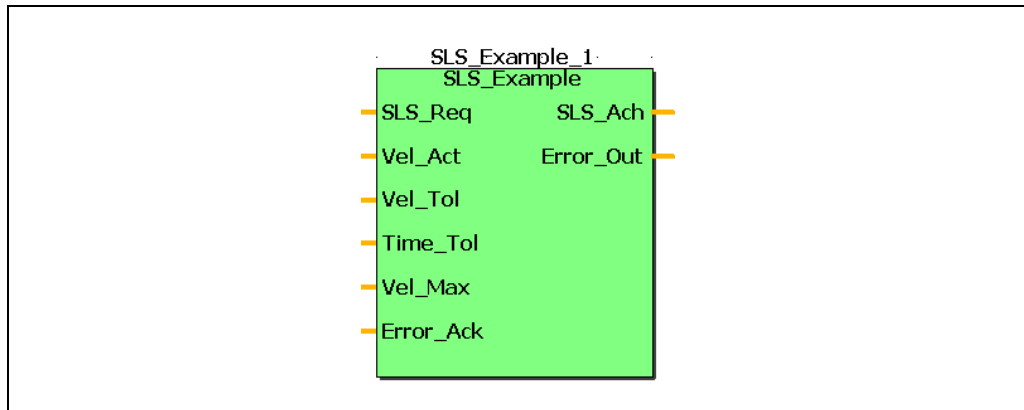
Blue area: Output signals control -> measuring system  
 Orange area: Input signals measuring system -> control system



## 4.2 Safe speed (SLS)

The function block SLS\_Example provides a monitoring for safe speed (SLS = Safely Limited Speed) on request. The amount of the current measuring system speed `Vel_Act` must be within the parameterized tolerance time `Time_Tol` below the parameterized safe speed `Vel_Tol` after the request `SLS_Req`. If this is the case, the output `SLS_Ach` is set. In case of an error, the output `Error_Out` is set.

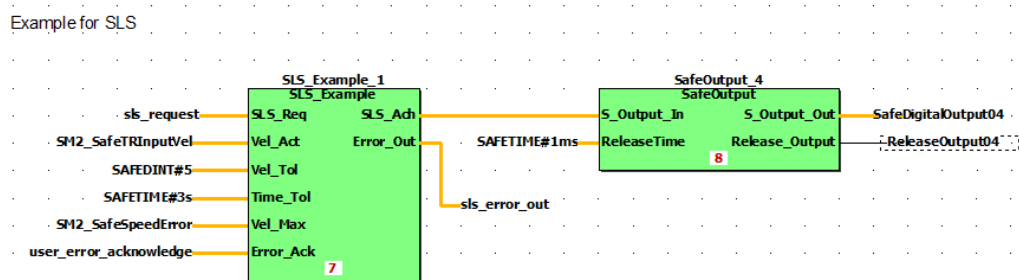
An error can be acknowledged with `Error_Ack`.



Signal	Type	Direction	Description
SLS_Req	SAFEBOOL	Input	Request for safe speed with SLS_Req = TRUE
Vel_Act	SAFEDINT	Input	Safe speed of the measuring system
Vel_Tol	SAFEDINT	Input	Limit for safe 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 acknowledgement
SLS_Ach	SAFEBOOL	Output	SLS reached
Error_Out	SAFEBOOL	Output	Error present, e.g. timeout

### 4.2.1 Code example SLS

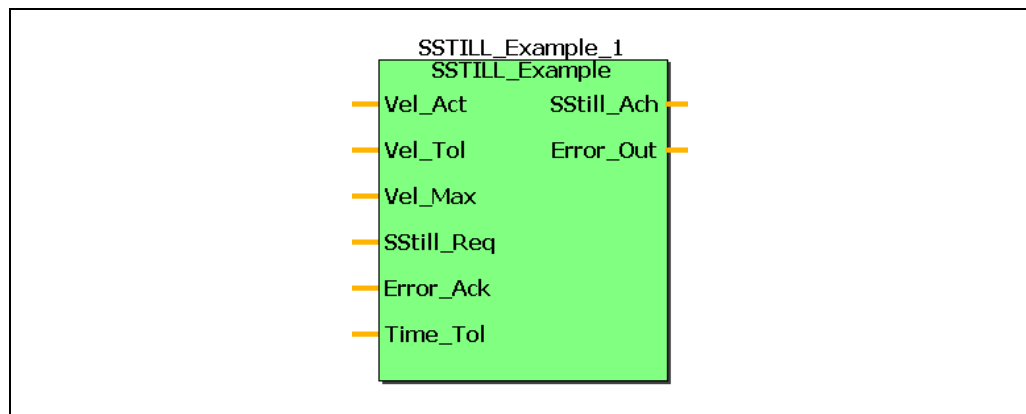
In the example project, the requested result is provided via digital output 4. Error evaluation is not performed in the example.



### 4.3 Safe standstill detection

The function block `SSTILL_Example` provides a monitoring for a safe standstill. As soon as the amount of the current measuring system velocity `Vel_Act` is smaller than the parameterized maximum velocity `Vel_Tol`, the output `SStill_Ach` is set. After the request for standstill `SStill_Req` the `Vel_Act` must be within the parameterized tolerance time `Time_Tol` below the parameterized maximum velocity `Vel_Tol` for a standstill. If this is not the case, the output `Error_Out` is set.

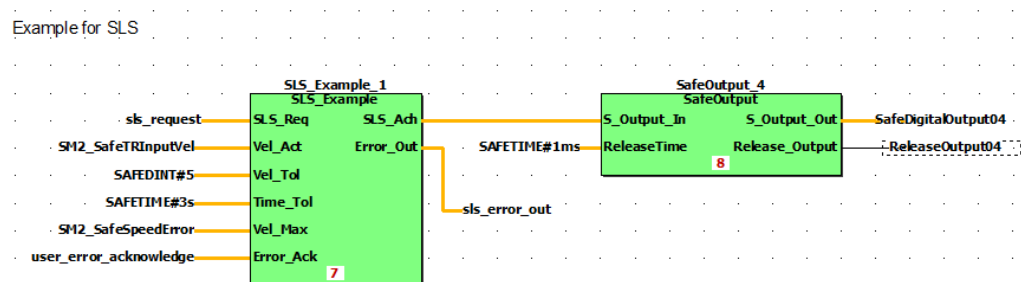
An error can be acknowledged with `Error_Ack`.



Signal	Type	Direction	Description
<code>Vel_Act</code>	SAFEDINT	Input	Safe speed of the measuring system
<code>Vel_Tol</code>	SAFEDINT	Input	Limit for standstill (tolerance)
<code>Vel_Max</code>	SAFEBOOL	Input	Measuring system speed overflow
<code>SStill_Req</code>	SAFEBOOL	Input	Request for safe standstill with <code>SStill_Req = TRUE</code>
<code>Error_Ack</code>	SAFEBOOL	Input	Error acknowledgement
<code>Time_Tol</code>	SAFETIME	Input	Maximum time from request to standstill
<code>SStill_Ach</code>	SAFEBOOL	Output	TRUE: Standstill
<code>Error_Out</code>	SAFEBOOL	Output	Error present, e.g. timeout

#### 4.3.1 Code example safe standstill detection

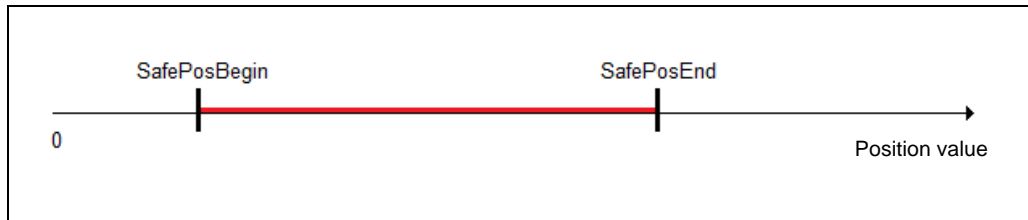
In the example project, the requested result is provided via digital output 5. An error evaluation does not take place in the example.



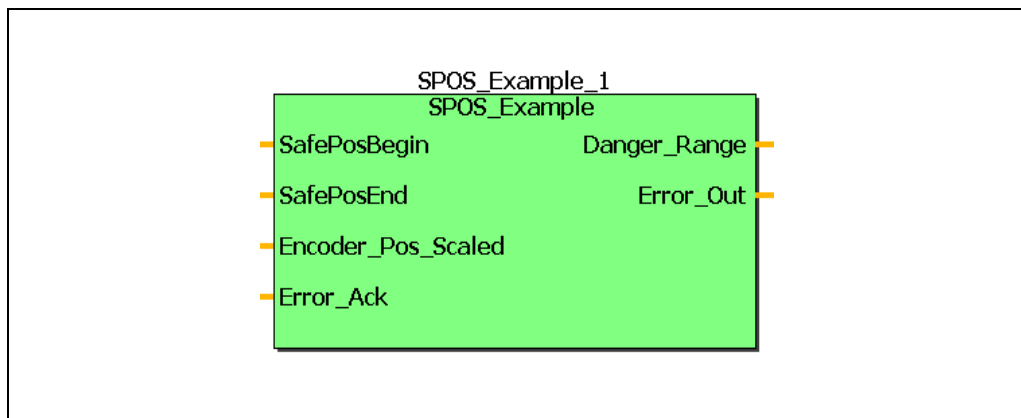
### 4.4 Safe position detection

The SPOS\_Example function block can be used to define a danger range. With the help of SafePosBegin and SafePosEnd an actual value range can be defined in which the Danger\_Range output is set. The range must always be selected so that SafePosBegin < SafePosEnd. An overflow is not allowed and leads to an error output.

An error can be acknowledged with Error\_Ack.



In the red area the output Danger\_Range is set.



Signal	Type	Direction	Description
SafePosBegin	SAFEDWORD	Input	Start of the dangerous area
SafePosEnd	SAFEDWORD	Input	End of the dangerous area
Encoder_Pos_Scaled	SAFEDWORD	Input	Current measuring system actual value
Error_Ack	SAFEBOOL	Input	Error acknowledgement
Danger_Range	SAFEBOOL	Output	TRUE: Measuring system actual value is in the dangerous range: $SafePosBegin \leq Encoder\_Pos\_Scaled \leq SafePosEnd$
Error_Out	SAFEBOOL	Output	Error is present, e.g. timeout



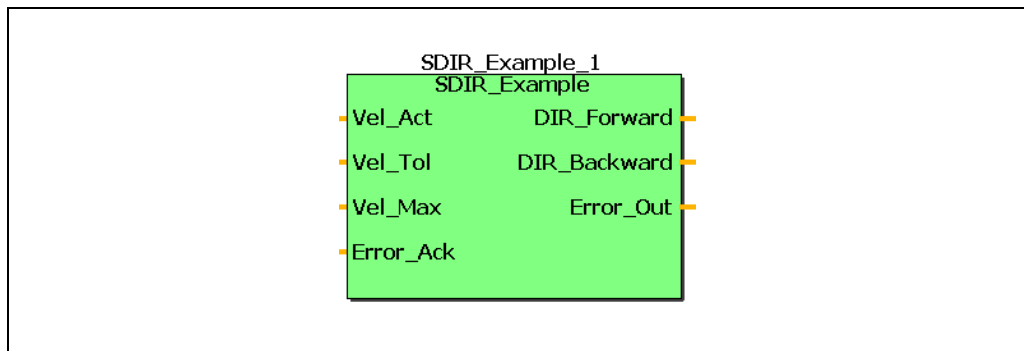
### 4.5 Safe direction detection

The `SDIR_Example` function block can be used to determine the counting direction of the measuring system:

- positive direction = increasing actual values
- negative direction = falling actual values

To avoid uncontrolled output due to slight movements at standstill, a velocity can be defined from which the direction detection is active. Under this velocity `Vel_Tol` the outputs `DIR_Forward` and `DIR_Backward` are always `FALSE`.

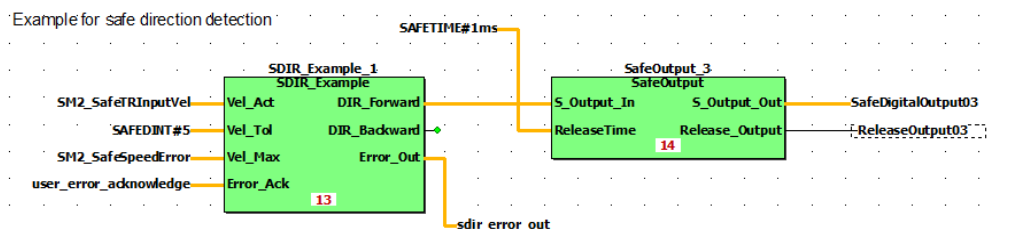
An error can be acknowledged with `Error_Ack`.



Signal	Type	Direction	Description
Vel_Act	SAFEDINT	Input	Safe speed of the measuring system
Vel_Tol	SAFEDINT	Input	Limit for direction measurement
Vel_Max	SAFEBOOL	Input	Measuring system speed overflow
Error_Ack	SAFEBOOL	Input	Error acknowledgement
DIR_Forward	SAFEBOOL	Output	TRUE: Measuring system actual value increasing
DIR_Backward	SAFEBOOL	Output	TRUE: Measuring system actual value decreasing
Error_Out	SAFEBOOL	Output	Error present, e.g. speed overflow

#### 4.5.1 Code example for safe direction detection

In the example project, a detected forward movement is provided via digital output 3. An error evaluation does not take place in the example.



## 4.6 Type conversion of the measuring system - actual values in SafeDESIGNER

By TR-Electronic, the actual values of the measuring system for singleturn, multiturn and scaled total value are provided as unsigned values.

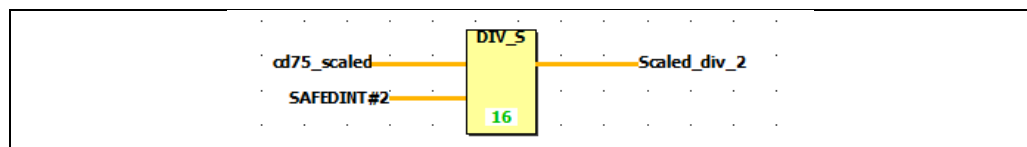
In SafeDESIGNER some functions cannot be performed with unsigned values. This includes, for example, division. If the user wants to perform a division with an unsigned value in SafeDESIGNER, he must cast unsigned values to a signed value.

10		cd75_multi	SAFEWORD		SL1.SM5.SafeTRInputMulti
11		cd75_scaled	SAFEWORD		SL1.SM5.SafeTRInputScaled
12		cd75_speederror	SAFEBOOL		SL1.SM5.SafeSpeedError
13		cd75_pres_status	SAFEBOOL		SL1.SM5.SafePresetStatus
14		cd75_pres_error	SAFEBOOL		SL1.SM5.SafePresetError
15		cd75_preset_ok	SAFEBOOL		SL1.SM5.SafePresetOK
16		cd75_pres_out_single	SAFEWORD		SL1.SM5.SafeTRPresetSingleturn
17		cd75_pres_out_multi	SAFEWORD		SL1.SM5.SafeTRPresetMultiturn
18		cd75_pres_out_prep	SAFEBOOL		SL1.SM5.SafePresetPreparation
19		cd75_pres_out_req	SAFEBOOL		SL1.SM5.SafePresetRequest

In the above figure it can be seen as an example that the measuring system initially has the `SAFEWORD` data type for the 32-bit total value. This is an unsigned 32-bit value. To use the scaled total value for a division, the value in the variable table can be changed to `SAFEDINT`:

10		cd75_multi	SAFEWORD		SL1.SM5.SafeTRInputMulti
11		cd75_scaled	SAFEDINT		SL1.SM5.SafeTRInputScaled
12		cd75_speederror	SAFEBOOL		SL1.SM5.SafeSpeedError

Now the value can be used in SafeDESIGNER as `SAFEDINT`:



What is to be noted for the user is that the value actually never becomes negative, because the uppermost bit is never set. The following value ranges in the Legacy-Mode apply regardless of whether the value is interpreted as signed or not:

Position value 32 bit unscaled: 0, 1, 2, ... 268435455 → Overflow to 0,1, 2....  
 Singleturn Legacy: 0, 1, 2, ... 8189, 8190, 8191 → Overflow to 0,1, 2....  
 Multiturn Legacy: 0, 1, 2, ... 32766, 32767 → Overflow to 0,1, 2....

So the values are always in the positive range.

**Risk of bodily injury and damage to property due to an actual value jump between 0 and maximum value!**

**⚠ WARNING**

**NOTICE**

- The value jump of the actual values between 0 and maximum value must be considered accordingly by the user. This means that when turning backwards in position 0, the next actual value is not -1 but the maximum value (e.g. 8191 for singleturn). This behavior is independent of whether the SafeDESIGNER value is interpreted as `SAFEWORD` or `SAFEDINT`.

## 5 Download - Software examples

The sample project described in this document can be downloaded here:

[www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0383](http://www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0383)

### 5.1 Legacy sample projects

The sample projects of the CD\_75M family can be helpful for using the legacy mode and can still be downloaded.

Documentation for commissioning a CD\_75M-EPL:

[www.tr-electronic.de/f/TR-ECE-TI-DGB-0264](http://www.tr-electronic.de/f/TR-ECE-TI-DGB-0264)

Example CD\_75M-EPL with XDD/OSDD device description from Automation Studio versions V4.5:

[www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0351](http://www.tr-electronic.de/f/zip/TR-ECE-TI-DGB-0351)