

## Serien XC50, XC55 und XC75

Digitalanzeigen

mit PROFINET IO RT-Schnittstelle

Quickstart für TIA Portal

Bedienungsanleitung

---

## Inhaltsverzeichnis

---

<b>1 Kontakt</b>	<b>3</b>
<b>2 Rechtlicher Hinweis</b>	<b>4</b>
<b>3 Konzept und Anwendungsfälle</b>	<b>5</b>
Aufbau .....	5
<b>4 Datenverbindung</b>	<b>6</b>
Trennung der Datenbindung von PROFINET und Ethernet .....	6
PROFINET und Ethernet über ein Kabel .....	6
<b>5 PROFINET Module und XC50/XC55/XC75 Datenmodell</b>	<b>7</b>
Baugruppenparameter .....	7
Plain Text Module .....	8
Extended Module .....	8
Extended Text Module .....	8
Extended Numeric Module .....	8
Extended Index Module .....	9
Extended JSON Module .....	9
<b>Quick-Start</b>	<b>10</b>
Schritt 1: Projekt öffnen .....	10
Schritt 2: GSDML-Datei der Anzeige installieren und Anzeige hinzufügen .....	11
Schritt 3: Zur Projektansicht wechseln und Gerät hinzufügen .....	11
Schritt 4: Verbindung zur Anzeige herstellen .....	11
Schritt 5: Sub-Modul der Anzeige hinzufügen und Adressbereich definieren .....	12
Schritt 6: Der Anzeige einen PROFINET-Gerätenamen zuweisen .....	13
Schritt 7: Text definieren .....	14
Schritt 8: Hardwarekonfiguration und Programm in die Steuerung laden .....	14
<b>6 Beispiel SCL</b>	<b>15</b>
SCL 1: Datenblock R_TRIG_DB definieren .....	15
SCL 2: Datentyp für die Texte definieren .....	15
SCL 3: Funktion definieren .....	16
SCL 4: SCL Code in das TIA Portal importieren .....	18
SCL 5: Ausgangsvariablen definieren .....	18
SCL 6: Programm auf die SPS laden .....	19
<b>7 Meldungen der Anzeige</b>	<b>21</b>

---

**1 Kontakt**

---

**www.siebert-group.com**

**DEUTSCHLAND**

Siebert Industrieelektronik GmbH  
Siebertstrasse, D-66571 Eppelborn  
Postfach 11 30, D-66565 Eppelborn  
Telefon +49 (0)6806 980-0, Fax +49 (0)6806 980-999  
Email info.de@siebert-group.com

**FRANKREICH**

Siebert France Sarl  
4 rue de l'Abbé Louis Verdet, F-57200 Sarreguemines  
BP 90 334, F-57203 Sarreguemines Cédex  
Telefon +33 (0)3 87 98 63 68, Fax +33 (0)3 87 98 63 94  
Email info.fr@siebert-group.com

**ITALIEN**

Siebert Italia Srl  
Via Galileo Galilei 2A, I-39100 Bolzano (BZ)  
Telefon +39 (0)471 053753, Fax +39 (0)471 053754  
Email info.it@siebert-group.com

**NIEDERLANDE**

Siebert Nederland B.V.  
Jadedreef 26, NL-7828 BH Emmen  
Telefon +31 (0)591 633444, Fax +31 (0)591 633125  
Email info.nl@siebert-group.com

**ÖSTERREICH**

Siebert Österreich GmbH  
Simmeringer Hauptstrasse 24, A-1110 Wien  
Telefon +43 (0)1 890 63 86-0, Fax +43 (0)1 890 63 86-99  
Email info.at@siebert-group.com

**SCHWEIZ**

Siebert AG  
Bützbergstrasse 2, CH-4912 Aarwangen  
Telefon +41 (0)62 922 18 70, Fax +41 (0)62 922 33 37  
Email info.ch@siebert-group.com

---

## **2 Rechtlicher Hinweis**

---

© Siebert Industrieelektronik GmbH

Diese Bedienungsanleitung wurde mit grösster Sorgfalt erstellt. Für eventuelle Fehler können wir jedoch keine Haftung übernehmen. Berichtigungen, Verbesserungsvorschläge, Kritik und Anregungen sind jederzeit willkommen. Bitte schreiben Sie an: [redaktion@siebert-group.com](mailto:redaktion@siebert-group.com)

Siebert<sup>®</sup>, LRD<sup>®</sup> und XC-Board<sup>®</sup> sind eingetragene Marken der Siebert Industrieelektronik GmbH. Soweit andere Produkt- oder Firmennamen in dieser Dokumentation erwähnt sind, können sie Marken oder Handelsnamen ihrer jeweiligen Inhaber sein.

Technische Änderungen und Liefermöglichkeiten vorbehalten. Alle Rechte, auch die der Übersetzung, vorbehalten. Kein Teil dieses Dokuments darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne unsere schriftliche Genehmigung reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

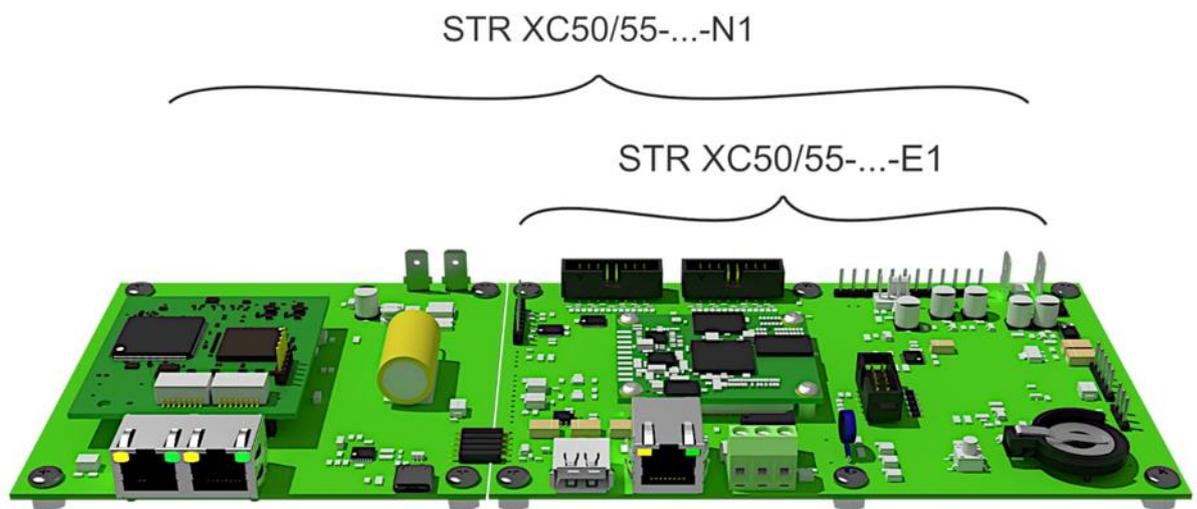
### 3 Konzept und Anwendungsfälle

Durch das Erweiterungsmodul N1 werden die Anzeigen der Serie XC50, XC55 und XC75 PROFINET-RT-fähig.

Diese Anleitung ergänzt die Bedienungsanleitung der Standardanzeigen und beschreibt den erweiterten Funktionsumfang der Anzeigen mit eingebautem Erweiterungsmodul gegenüber der Standardausführung, die weiterhin unverändert zur Verfügung steht.

#### Aufbau

In der Anzeige befindet sich links neben dem Steuerrechner das Erweiterungsmodul N1 mit PROFINET-Interface. Dieses verfügt über zwei RJ45-Buchsen.



Die RJ45-Buchse auf dem Steuerrechner XC50/XC55-...-E1 dient der Ethernet-Ankopplung zur Parametrierung und Bedienung mithilfe des integrierten Webservers. Diese Funktionalität entspricht dem Standard-Funktionsumfang, der in einer separaten Bedienungsanleitung beschrieben ist. Darüber hinaus kann die Anzeige über diese Buchse mit JSON-Datenstrings oder dem PLC Connector angesteuert werden.

Die RJ45-Doppelbuchse dient zur PROFINET-Ansteuerung. Das Interface ist Class-C fähig und kann somit als Switch für eine Linienverdrahtung von PROFINET-Geräten verwendet werden.

Der Datenaustausch zwischen dem PROFINET-Erweiterungsmodul N1 und dem Steuerrechner XC50/XC55-...-E1 erfolgt über ein internes Bussystem. Die Kopplung zwischen den PROFINET-Modulen im TIA Portal und den Layout-Elementen der Anzeige erfolgt über eine Namenskonvention, die später erläutert wird.

---

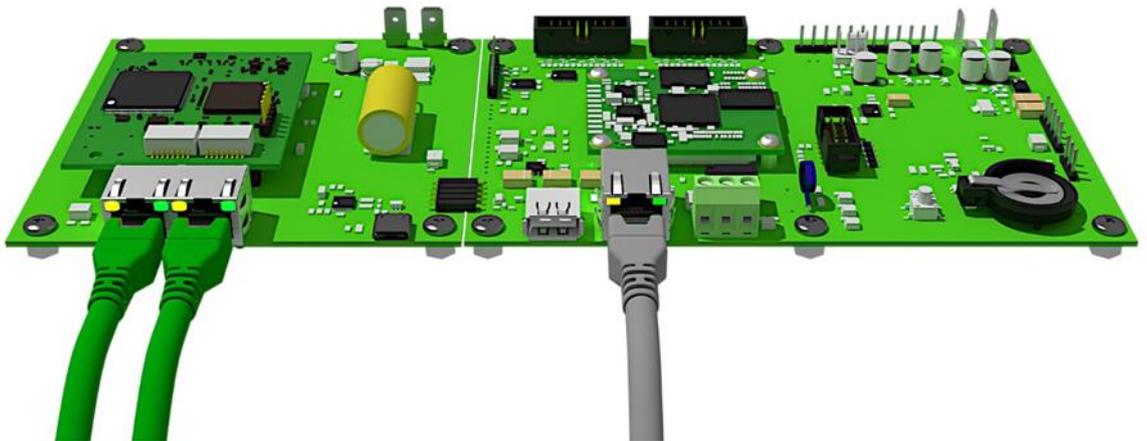
## 4 Datenverbindung

---

### Trennung der Datenbindung von PROFINET und Ethernet

Durch eine getrennte Verkabelung von PROFINET –N1 und Ethernet –E1 kann die Anzeige völlig unabhängig voneinander über das jeweilige Interface betreiben werden. Somit lässt sich die Anzeige auch über physikalisch getrennte Netzwerke ansteuern. In dieser Konfiguration werden bis zu drei Netzkabel in die Anzeige geführt: eines für PROFINET (bzw. zwei für die PROFINET Linienverdrahtung) und eines für Ethernet.

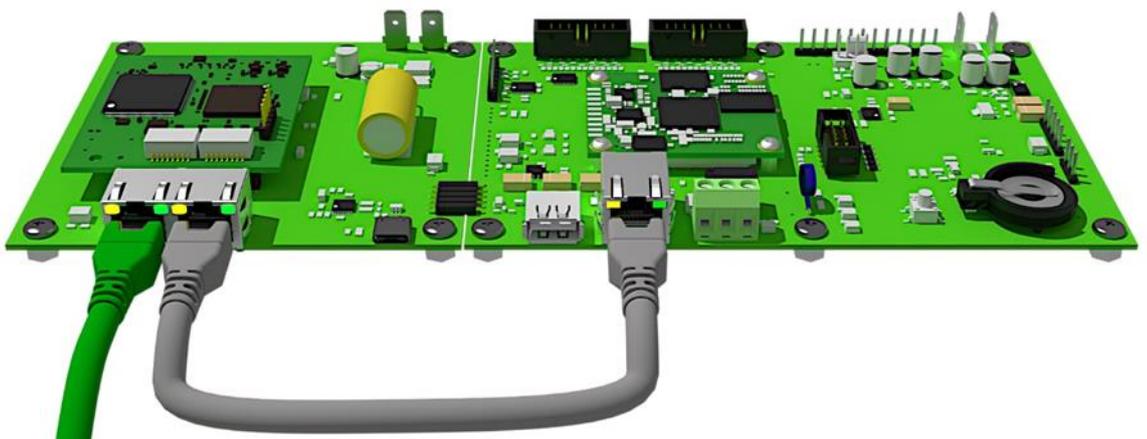
Über die Ethernet-Anbindung wird die Anzeige parametrieren und das Layout konfiguriert. Sind diese Einstellungen abgeschlossen, kann die Ethernet-Verbindung vom Netzwerk getrennt werden. Der Empfang der dazustellenden Informationen erfolgt über PROFINET.



### PROFINET und Ethernet über ein Kabel

Der interne Switch auf dem PROFINET-Erweiterungsmodul ermöglicht es, Ethernet-Daten über die Datenzuleitung auf das PROFINET-Modul an den Steuerrechner XC50/XC55-...-E1 weiterzugeben. Dadurch sinkt der Verdrahtungsaufwand und die Webseite der Anzeige bleibt erreichbar.

Weil nun ein Kabel verwendet wird sind PROFINET und Ethernet nicht mehr physikalisch getrennt.



## 5 PROFINET Module und XC50/XC55/XC75 Datenmodell

Im PROFINET Engineering Tool stehen nach dem Import der GSDML Datei verschiedene Module zur Verfügung. Durch die Verwendung eines oder mehrerer Module werden die PROFINET Adressbereiche und die Interpretation der darüber gesendeten Daten in der Anzeige definiert.

Bei den Anzeigen der Serien XC50/55/75 können maximal vier Module pro Anzeige gesteckt werden.

Jedes gesteckte Modul benötigt im PROFINET Ausgangsbereich 32 aufeinanderfolgende Bytes. Die Lage des Moduls im PROFINET Adressraum wird vom Anwender beim Stecken des Moduls definiert. Durch einen Baugruppenparameter erfolgt die Zuordnung der PROFINET Moduldaten zu den Elementen des XC50/55/75 Datenmodells.

Die Aktualisierungsrate der Anzeige beträgt eine Sekunde. Wenn Daten in der Steuerung schneller aktualisiert werden, dann können Werte auf der Anzeige möglicherweise übersprungen werden.

Die 32 Bytes eines Moduls werden von der Anzeige als String interpretiert und die Bedeutung wird durch den Modultyp festgelegt. Strings müssen grundsätzlich mit einem Nullbyte (\$00) abgeschlossen werden.

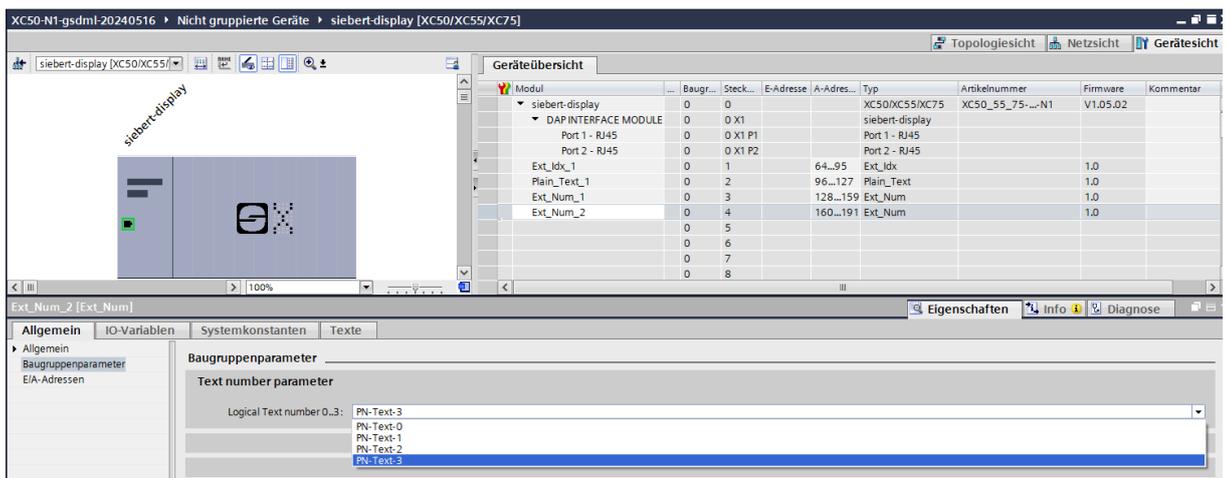
In der GSDML Datei sind die folgenden Module definiert:

- Plain Text
- Extended Text
- Extended Numeric
- Extended Index
- Extended JSON
- 

### Baugruppenparameter

Jedes Modul hat den Baugruppenparameter „Logical Text Number“.

Beim Stecken eines Moduls im Engineering Tool wird mit diesem Parameter die Textnummer festgelegt, unter der die Daten dieses Moduls im Datenmodell der Anzeige erscheinen. Die Textnummer muss für jedes gesteckte Modul eindeutig sein.



Modul	Baugr...	Steck...	E-Adresse	A-Adres...	Typ	Artikelnummer	Firmware	Kommentar
siebert-display	0	0			XC50/XC55/XC75	XC50_55_75-...N1	V1.05.02	
DAP INTERFACE MODULE	0	0	X1		siebert-display			
Port 1 - RJ45	0	0	X1	P1	Port 1 - RJ45			
Port 2 - RJ45	0	0	X1	P2	Port 2 - RJ45			
Ext_idx_1	0	1		64...95	Ext_idx		1.0	
Plain_Text_1	0	2		96...127	Plain_Text		1.0	
Ext_Num_1	0	3		128...159	Ext_Num		1.0	
Ext_Num_2	0	4		160...191	Ext_Num		1.0	
	0	5						
	0	6						
	0	7						
	0	8						

## Plain Text Module

Mit diesem Modul kann ein Text Element der XC50/55/75 Anzeige mit einem maximal 32 Bytes langen Text beschrieben werden.

Der Name des Textelementes im Layout der Anzeige und der Wert des Baugruppenparameters müssen sich dabei entsprechen. Maximal vier Textelemente mit den Namen PN-Text-0, PN-Text-1, PN-Text-2 und PN-Text-3 sind möglich.

Mit diesem Modul lässt sich die Anzeige am einfachsten über PROFINET kommandieren. Im folgenden QuickStart wird das ausführlich beschrieben.

## Extended Module

Extended Module werden verwendet, wenn mehr als vier Layoutelemente über PROFINET angesprochen werden. Dabei müssen die Informationen nacheinander übertragen werden. Die Aktualisierungsrate der Anzeige nimmt mit der Anzahl der Layoutelemente ab.

Bei den Extended Modulen wird ebenfalls ein maximal 32 Byte langer String übertragen. Dabei wird der vollständige Name des Layout Elementes, das kommandiert wird, an den Anfang des Strings gestellt. Damit wird der eigentliche Inhalt um die Länge des vollständigen Namens und des Trennzeichens „|“ reduziert. Bei allen Extended Modulen hat der String den folgenden Aufbau:

```
Elementname|Text$00
```

Sinnvollerweise werden kurze Elementnamen gewählt, z.B. T1, T2 für Textelemente oder N1, N2 für numerische Elemente.

Bei hierarchisch angeordneten Elementen muss der vollständige Name angegeben werden, z.B. C1.T3 für das Textelement T3 das in dem Container C1 platziert ist. Dadurch ist es möglich, in unterschiedlichen Containern C1, C2, ... Elemente gleich zu benennen. Durch die vollständigen Namen C1.T3 bzw. C2.T3 sind die beiden Textelement eindeutig anzusprechen.

Der Typ des Extended Moduls legt fest wie der Text nach dem „|“ zu interpretieren ist.

Der Baugruppenparameter muss bei den Extended Modulen ebenfalls eindeutig sein.

## Extended Text Module

Bei Extended Text wird die Information an ein Textelement weiter gereicht und als Text dargestellt. Dieses Modul kann mit jedem Layoutelement verwendet werden das über eine „Text“ Property verfügt, z.B. das „Text“ oder das „Textarea“ Element.

Mit dem folgenden String wird auf das Textelement C1.T1 der Text Hallo geschrieben. Mit dem Nullbyte wird der String abgeschlossen.

```
C1.T1|Hallo$00
```

## Extended Numeric Module

Bei Extended Numeric wird die Information an ein numerisches Element weiter gereicht und als Zahl (FLOAT) interpretiert. Dieses Modul kann mit jedem Layoutelement verwendet werden das über eine „Value“ Property verfügt, z.B. ein „Numeric“ oder ein „Bargraph“ Element.

Mit dem folgenden String wird auf das numerische Element C1.N1 die Zahl 1234 geschrieben. Durch das Layoutelement wird dann bestimmt, in welcher Form diese Zahl auf der Anzeige erscheint. Mit dem Nullbyte wird der String abgeschlossen.

```
C1.N1|1234$00
```

### Extended Index Module

Bei Extended Index wird die Information an ein Element weiter gereicht und als positive ganze Zahl (UINT) interpretiert. Dieses Modul kann mit jedem Layoutelement verwendet werden das über eine „Index“ Property verfügt, z.B. das „Multitext“ oder das „Multiimage“ Element.

Mit dem folgenden String wird bei dem Index Element C1.I1 der zum Index 4 gehörende Eintrag aufgerufen. Beim einem Multitext Element ist das ein Text aus der Textliste, bei einem Multiimage ein Bild aus der Bildliste. Mit dem Nullbyte wird der String abgeschlossen.

```
C1.I1|4$00
```

### Extended JSON Module

Bei den vorhergehenden Extended Modulen wurde über den Typ festgelegt, welche Property des Elementes kommandiert wird.

Mit dem Extended JSON Module wird ein JSON String an das Element gesendet. Damit sind alle Properties dieses Elementes ansprechbar, vorausgesetzt der komplette String hat weniger als 32 Bytes. Dieses Modul kann mit jedem Layoutelement verwendet werden. Welche Properties für ein bestimmtes Element definiert sind lässt sich z.B. auf der Webseite der Anzeige im Layout Editor feststellen.

Das vollständige JSON Protokoll ist in der BAL der Anzeige beschrieben:

```
{  
  "meta":{"request":"setData", . . .},  
  "data":{" . . . }}  
}
```

Beim Ext\_JSON Modul wird nur der Teilstring nach "data": einschließlich der geschweiften Klammern übertragen.

Mit dem folgenden String wird z.B. bei dem Element C1.T1 die Farbe auf blau geändert. Mit dem Nullbyte wird der String abgeschlossen.

```
C1.T1|{"color":"blue"}$00
```

## Quick-Start

Dieser Quick Start gilt für alle alphanumerische Anzeigen der Serien XC50-...-N1, XC55-...-N1 und XC75-...-N1 mit der Firmware ab V1.0.19 und der Device ID 0x0050. Die Firmware eines Gerätes kann, z.B. mit der Software PRONETA Basic der Firma SIEMENS, vorab geprüft werden:

Hersteller-ID	Geräte-ID	Firmwareversion	Hersteller-Name	Bestellnummer
0x0161	0x0050	V1.0.12	Siebert Industrieelektronik GmbH	XC50_55_75-...-N1

Anhand der folgenden Schritte wird die Anzeige in Betrieb genommen. Anschliessend zeigt die Anzeige die über PROFINET gesendeten Werte an.

Die Screenshots wurden mit der in folgender Tabelle aufgeführten Hard- und Software erstellt. Bei anderen Engineering-Frameworks können die Darstellungen abweichen.

Anzeige	XC50-096.032-...-N1
Engineering-Framework	Siemens TIA Portal V17, Update 4
SPS	Siemens S7-1214C DC/DC/Rly, V4.4, 6ES7 214-1HG40-0XB0
Betriebssystem	Microsoft Windows 10 Professional, 64 Bit

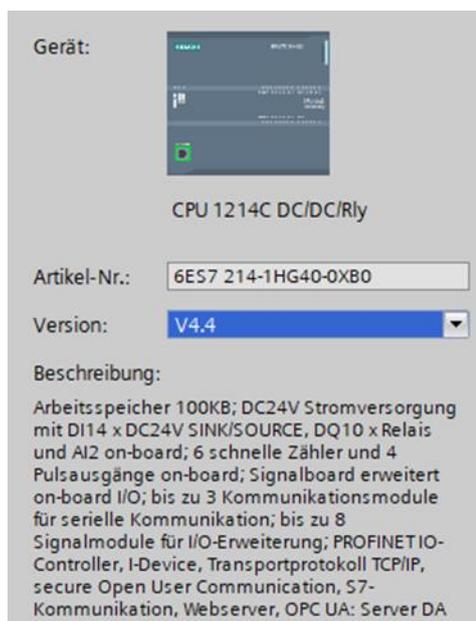
Der Quick-Start beschreibt die Parametrierung der Anzeige. Die Bedienung des TIA Portals wird vorausgesetzt.

Die Geräte werden ohne PROFINET-Namen ausgeliefert.

Das Beispielprojekt aus diesem Quick Start steht auf der Webseite [www.siebert-group.com](http://www.siebert-group.com) zum Download zur Verfügung.

### Schritt 1: Projekt öffnen

Öffnen Sie ein neues Projekt im Engineering-Framework und definieren Sie die CPU. Beim Erstellen dieses Quick Starts wurde die Siemens S7-1214C DC/DC/Rly mit der Version 4.4 verwendet.



Die Einstellungen für die IP Adresse, den Namen, die Schutzstufe usw. einstellen.

## Schritt 2: GSDML-Datei der Anzeige installieren und Anzeige hinzufügen

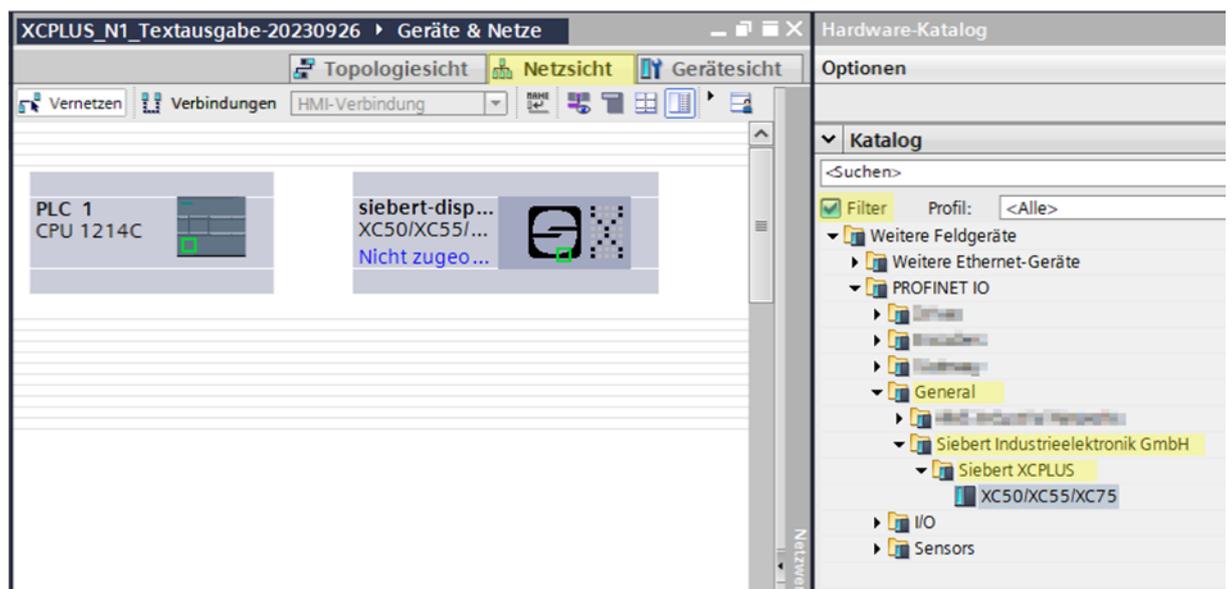
Installieren Sie die GSDML-Datei „GSDML-V2.43-Siebert-XC50\_55\_75-XX-...“ der Anzeige. Sie finden diese auf dem im Lieferumfang enthaltenen Datenträger oder auf [www.siebert-group.com](http://www.siebert-group.com). Nach der Installation wird die Anzeige im Hardwarekatalog aufgeführt.

Aktivieren Sie im Gerätekatalog die Checkbox 'Filter' und navigieren Sie zu dem Eintrag: 'General / Siebert Industrieelektronik GmbH / Siebert XCPLUS'.

## Schritt 3: Zur Projektansicht wechseln und Gerät hinzufügen

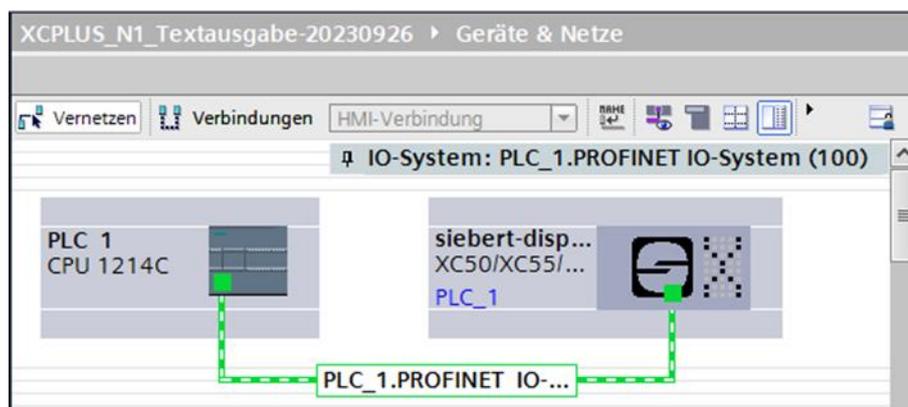
Wechseln Sie zur Ansicht 'Geräte & Netze / Netzsicht'.

Ziehen Sie das Kopfmodul 'XC50/XC55/XC75' vom Katalog auf die Fläche 'Geräte & Netze / Netzsicht'.



## Schritt 4: Verbindung zur Anzeige herstellen

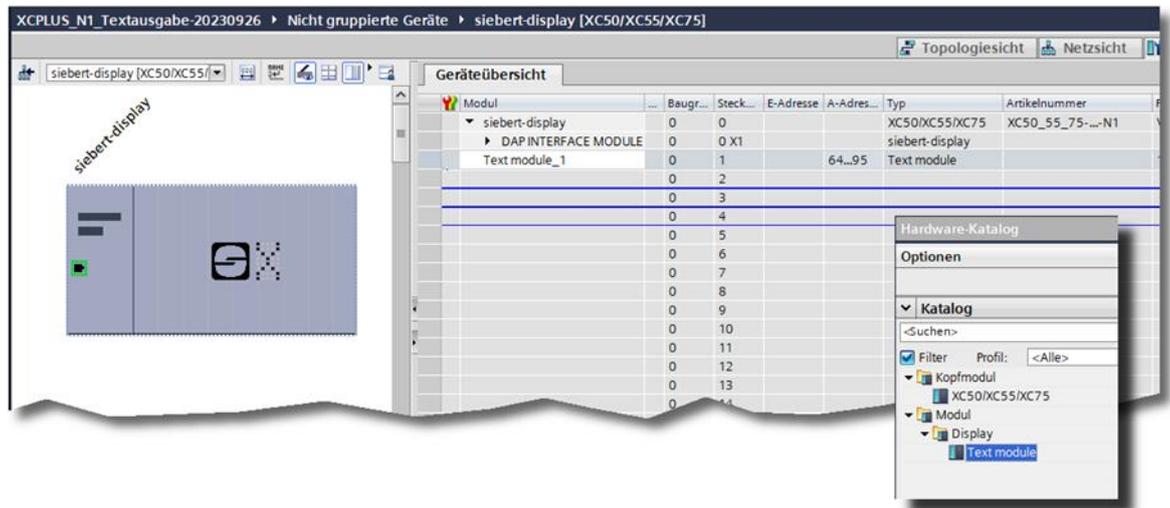
Ordnen Sie in der Netzsicht die Anzeige der gewünschten Steuerung zu. Danach wird die PROFINET-Verbindung als gestrichelte grüne Linie angezeigt.



### Schritt 5: Sub-Modul der Anzeige hinzufügen und Adressbereich definieren

Fügen Sie nun das Modul 'Text module' aus dem Hardware-Katalog in die Geräteübersicht hinzu.

Hierdurch wird festgelegt, dass die Anzeige die darzustellenden Werte als ASCII String erwartet. Das Modul belegt 32 Bytes im IO-Bereich (in diesem Beispiel die Adressen 64 bis 95).

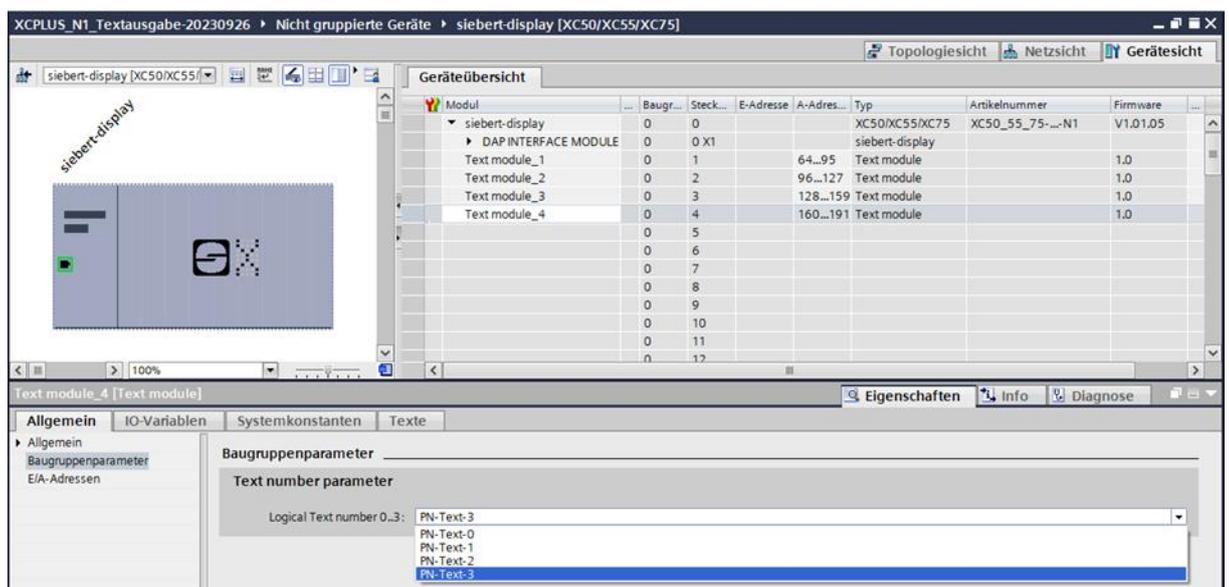


Den vom TIA Portal vorgegebenen Namen können Sie individuell ändern.

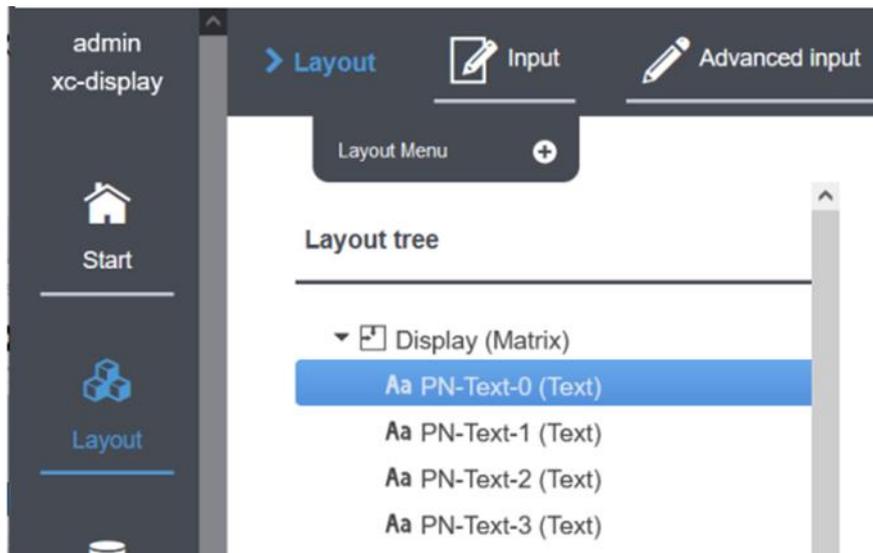
Maximal können vier 'Text module' hinzugefügt werden.

Der Inhalt der PROFINET Text Module wird den Layout-Elementen PN-Text-0, PN-Text-1, PN-Text-2 und PN-Text-3 der XC50, XC55 und XC75 Anzeigen zugeordnet. Dazu ist für jedes Modul der Initialisierungsparameter 'Logical Text number' einzustellen.

**ACHTUNG:** Jedes hinzugefügte 'Text module' muss eindeutig einer 'Logical Text number' zugeordnet werden. Wird eine 'Logical Text number' mehrfach verwendet, dann führt das zu Fehlermeldungen beim Programmstart.



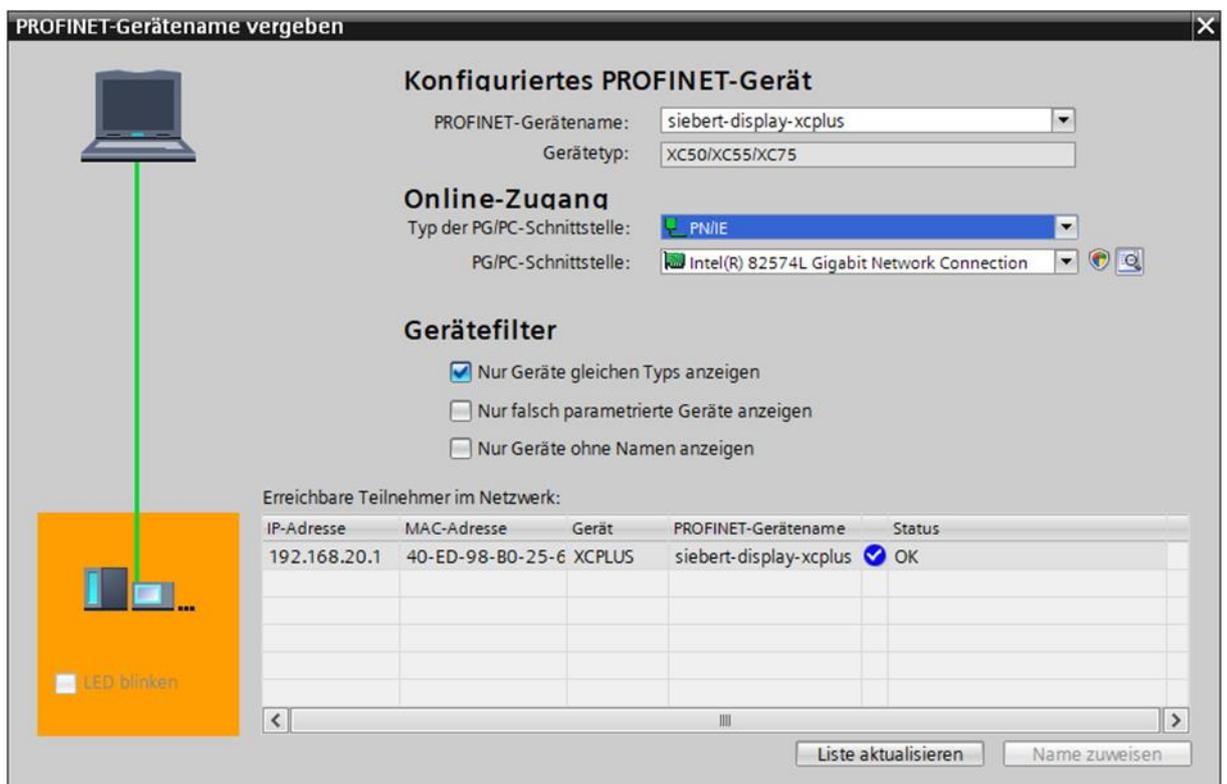
**ACHTUNG:** Zu jedem hinzugefügten 'Text module' muss im Layout der Anzeige ein gleichnamiges Textelement definiert werden. Die Daten werden von der SPS über PROFINET zu den Textmodulen gesendet und von dort aus weiter zu den Textelementen im Layout der Anzeige. Wenn im Layout die Textelemente nicht vorhanden sind, dann können die Texte nicht auf der Anzeige dargestellt werden.



### Schritt 6: Der Anzeige einen PROFINET-Gerätenamen zuweisen

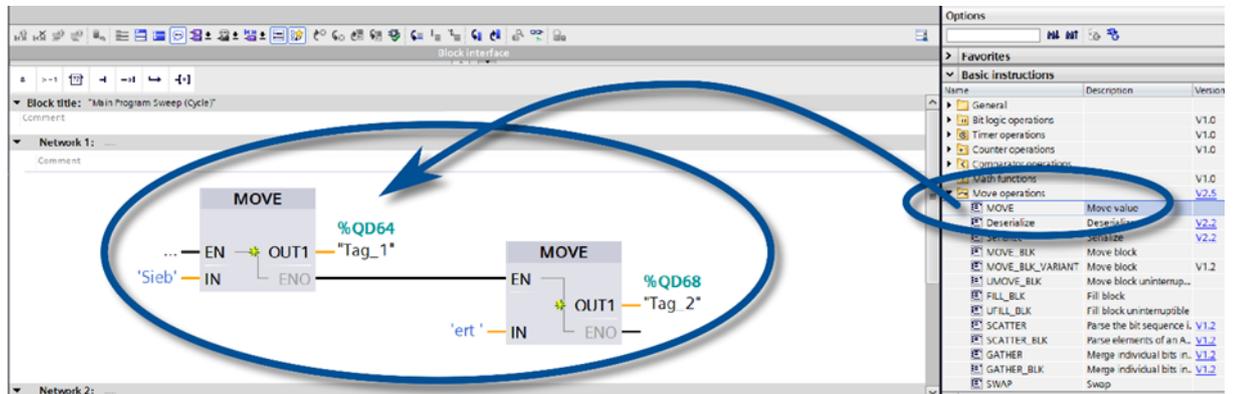
Teilen Sie nun der Anzeige eine IP-Adresse und einen Gerätenamen zu. Rufen Sie dazu z.B. das Kontextmenu der Siebert Anzeige in der Gerätesicht auf und klicken Sie den Punkt 'Gerätename zuweisen' an.

Nachdem die Daten erfolgreich zur Anzeige übertragen wurden, wird die Konfiguration wie folgt im Engineering-Tool dargestellt.



### Schritt 7: Text definieren

Um einen Wert zur Anzeige zu senden, können Sie beispielsweise einen Funktionsplan mit 'MOVE' Anweisungen erstellen. Im folgenden Screenshot wird der Text 'Siebert' gesendet. Dazu werden zwei 'MOVE' Bausteine benötigt.



Ein umfangreicheres Beispiel, welches SCL verwendet, wird im folgenden Kapitel beschrieben.

### Schritt 8: Hardwarekonfiguration und Programm in die Steuerung laden

Nach dem Einschalten der Anzeige wird das voreingestellte Layout angezeigt.

Sobald die Konfiguration und der Programmbaustein in die Steuerung geladen ist verbindet sich die Steuerung mit der Anzeige über PROFINET und die Anzeige zeigt auf dem Layout-Element 'PN-Text-0' den Text 'Siebert' an.

---

## 6 Beispiel SCL

---

Alternativ zur Verwendung der Move Blöcke kann die Anzeige beispielsweise auch mit SCL beschrieben werden.

Hierbei ändert sich Schritt 7 des Quickstarts.

Sie können die folgenden SCL Programmteile aus diesem Dokument kopieren, diese als Textdatei auf dem Rechner speichern und über den Punkt 'Externe Quellen / Neue externe Datei hinzufügen' in Ihr Projekt importieren.

ToDo: Taktmerkerbits hinzufügen

### SCL 1: Datenblock R\_TRIG\_DB definieren

Dieser Datenblock dient dazu, dass das eigentliche Programm einmal pro Sekunde aufgerufen wird.

```
DATA_BLOCK "R_TRIG_DB"
{InstructionName := 'R_TRIG';
 LibVersion := '1.0';
 S7_Optimized_Access := 'TRUE' }
AUTHOR : Siebert
FAMILY : BIT
NAME : R_TRIG
VERSION : 1.0
NON_RETAIN
R_TRIG

BEGIN
END_DATA_BLOCK
```

### SCL 2: Datentyp für die Texte definieren

Über den eigenen Datentyp lassen sich die Ausgangsdaten einfach zuweisen.

```
TYPE "XC50_55_75_Texte"
VERSION : 1.1
STRUCT
Text : Array[0..31] of Byte;
END_STRUCT;

END_TYPE
```

### SCL 3: Funktion definieren

In der Funktion SCL\_Test werden zunächst die Hilfsvariablen definiert. Danach wird, gesteuert über R\_TRIG\_DB, einmal pro Sekunde die Berechnung der vier Texte durchgeführt. Jeder Text besteht aus einem Zähler der sich sekundlich ändert und einem festen Teil, der den Text kennzeichnet.

Jeder Text muss mit einem Null-Byte abgeschlossen werden. Ohne dieses Null-Byte würden möglicherweise Teile von früher gesendeten längeren Texten weiterhin angezeigt.

```
FUNCTION "SCL_Test" : Void
TITLE = Strings für XC50 XC55 und XC75
{ S7_Optimized_Access := 'TRUE' }
VERSION : 1.1

VAR_TEMP
    chrcnt : UInt;
    cnt : Byte;
    str5bytes : String[5];
    Text_0 : String[30];
    Text_1 : String[30];
    Text_2 : String[30];
    Text_3 : String[30];
    Trig_1Hz : Bool;
END_VAR

BEGIN

    "R_TRIG_DB" ( CLK := "Clock_1Hz",
                 Q  => #Trig_1Hz );

    IF (#Trig_1Hz = true)
    THEN
        "counterwert" := "counterwert" + 1;

        // Die PLC Variable counterwert, die automatisch hochgezählt wird,
        // ist vom Typ Word. Für die einzelnen Stellen der XC50 werden Bytes
        // benötigt, deshalb die Konvertierungsfunktion INT_TO_BYTE.

        #cnt := INT_TO_BYTE("counterwert");
        #str5bytes := INT_TO_STRING(#cnt);
```

```
// Texte mit $00 abschliessen

#Text_0 := CONCAT(IN1 := #str5bytes, IN2 := ' T0$00');
#Text_1 := CONCAT(IN1 := #str5bytes, IN2 := ' T1$00');
#Text_2 := CONCAT(IN1 := #str5bytes, IN2 := ' T2$00');
#Text_3 := CONCAT(IN1 := #str5bytes, IN2 := ' T3$00');

Strg_TO_Chars(Strg := #Text_0,
              pChars := 0,
              Cnt => #chrCnt,
              Chars := "xc50_t0".Text);

Strg_TO_Chars(Strg := #Text_1,
              pChars := 0,
              Cnt => #chrCnt,
              Chars := "xc50_t1".Text);

Strg_TO_Chars(Strg := #Text_2,
              pChars := 0,
              Cnt => #chrCnt,
              Chars := "xc50_t2".Text);

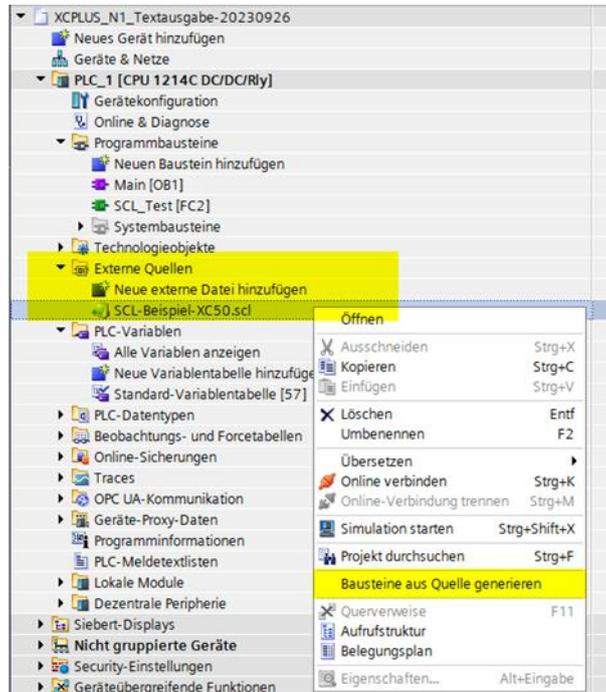
Strg_TO_Chars(Strg := #Text_3,
              pChars := 0,
              Cnt => #chrCnt,
              Chars := "xc50_t3".Text);

END_IF;
END_FUNCTION
```

#### SCL 4: SCL Code in das TIA Portal importieren

Im TIA Portal können Sie unter dem Punkt 'Externe Quellen' durch einen Doppelklick auf 'Neue externe Quelle hinzufügen' die Windows Dateiauswahl öffnen und die Textdatei SCL-Beispiel-XC50.scl, die aus den Codeteilen SCL 1 bis SCL 3 besteht, importieren.

Danach kann im Kontextmenu der Punkt 'Bausteine aus Quelle generieren' aufrufen werden.



#### SCL 5: Ausgangsvariablen definieren

In der SCL Datei werden bei den `Strg_TO_Chars` Funktionsaufrufen die Variablen

"xc50\_t0".Text

"xc50\_t1".Text

"xc50\_t2".Text

"xc50\_t3".Text

beschrieben. Diese Variablen müssen nun noch in der Standard Variablen-tabelle angelegt werden. Durch den Anfangs definierten Datentyp 'XC50\_55\_75\_Texte' werden die 32 Bytes automatisch angelegt. Als Startadresse muss %Qnn.0 angegeben werden.

xc50_t0	"XC50_55_75_Texte"	%Q64.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
xc50_t1	"XC50_55_75_Texte"	%Q96.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
xc50_t2	"XC50_55_75_Texte"	%Q128.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
xc50_t3	"XC50_55_75_Texte"	%Q160.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

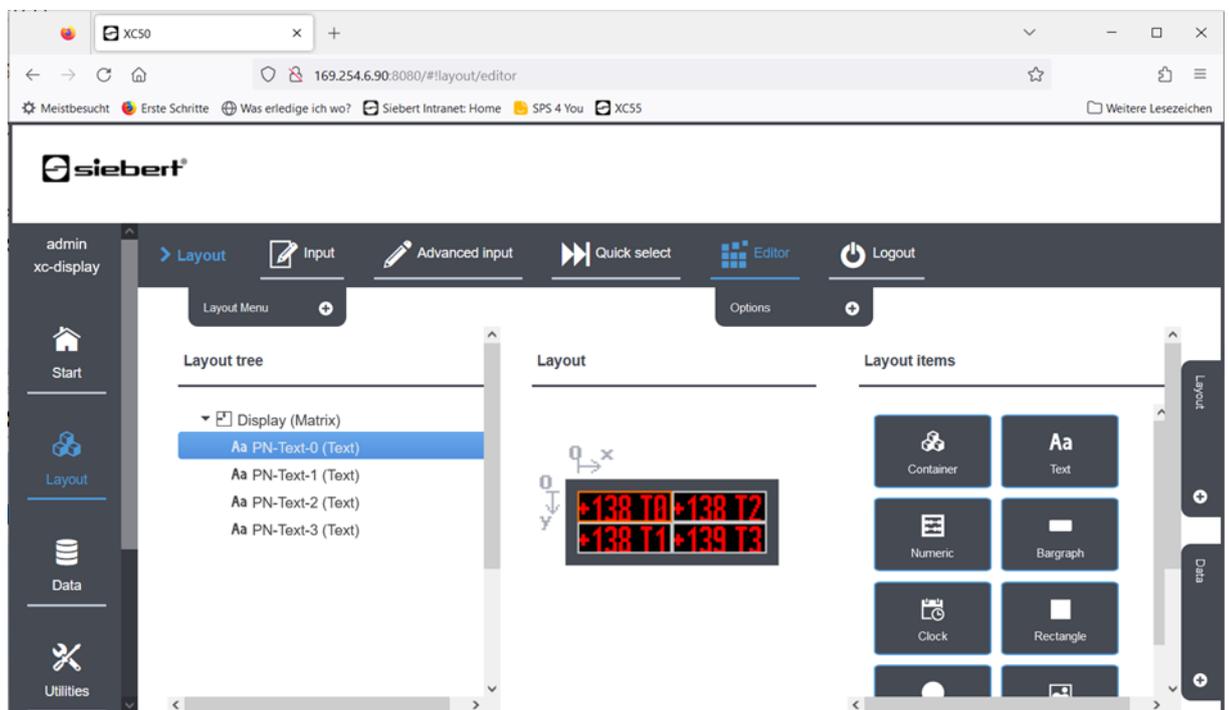
ToDo: DB in Main beschreiben

## SCL 6: Programm auf die SPS laden

Nach dem kompilieren und hochladen in die Steuerung sollte die Anzeige die folgenden Informationen anzeigen. Die Zahlen ändern sich einmal pro Sekunde und die Texte T0, T1, T2 und T3 sind konstant und zeigen an, um welchen Text es sich handelt.



Auf der Anzeige ist im Auslieferungszustand ein Layout definiert, das aus vier Textelementen mit den passenden Namen besteht.



Die Lage, Grösse, Schriftart, Farbe, Laufschrift und alle weiteren Eigenschaften der Textelemente werden über die Webseite der Anzeige im Layout-Editor definiert. Sie lassen sich über PROFINET nicht ändern.

Die Textelemente müssen sich auf der obersten Ebene befinden. Elemente in Containern oder Pagern sind über PROFINET nicht erreichbar.

Die interne Aktualisierung der Anzeige erfolgt einmal pro Sekunde. Es ist möglich auch einen Text über die Webseite vorzugeben. Dieser wird aber spätestens nach einer Sekunde durch den Inhalt der PROFINET Variablen überschrieben.

Auch ist es möglich, die Daten über PROFINET schneller zu senden. In diesem Fall werden nicht alle Texte dargestellt, sondern genau nur die Daten, die bei der sekundlichen Abfrage gerade aktuell sind.

Das Layout kann zusätzlich beliebige weitere Layout-Element enthalten, z.B. Bilder, weitere Texte, Datum und Uhrzeit. Diese werden entweder von der Anzeige automatisch aktualisiert, z.B. Datum, Uhrzeit, Temperatur, oder müssen vom Anwender entweder über die Webseite, über JSON oder eine der anderen Möglichkeiten aktualisiert werden.

---

## 7 Meldungen der Anzeige

---

Der Datenverkehr über PROFINET wird mit Status-LEDs an beiden RJ45-Buchsen signalisiert. Die Bedeutung ist bei beiden Ports gleich.

LED grün	LED gelb	Bedeutung
aus	aus	<ul style="list-style-type: none"><li>keine Spannungsversorgung</li><li>keine Netzwerkverbindung</li></ul>
ein	aus	<ul style="list-style-type: none"><li>Netzwerkverbindung vorhanden</li></ul>
ein	blinkt	<ul style="list-style-type: none"><li>Initialisierungsphase</li><li>Application Relation wird aufgebaut</li></ul>
ein	ein	<ul style="list-style-type: none"><li>Application Relation etabliert</li></ul>

Kann die Verbindung nicht aufgebaut werden oder treten Störungen im Betrieb auf, zeigt die Anzeige die Fehlermeldungen an. Mögliche Ursachen sind IP-Probleme durch falsche Netzwerkparameter, mehrfach vergebene Gerätenamen, Kommandierung nicht definierter Properties oder sonstige Feldbus-Störungen.