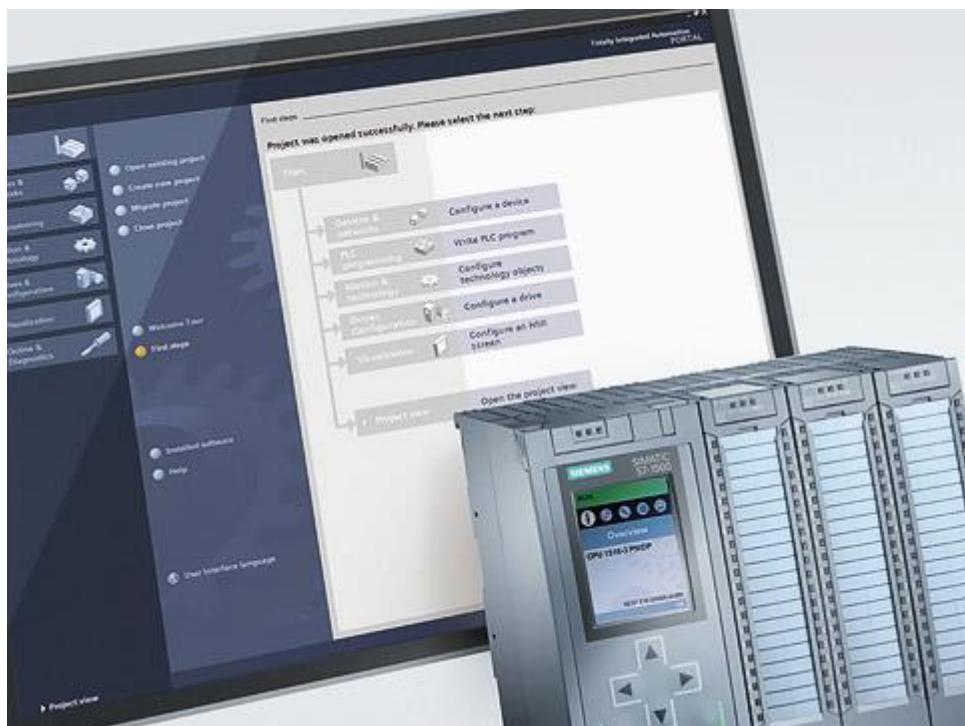


Handhabungsanleitung TIA Portal V15.1



Inhaltsverzeichnis



1 Aufbau TIA Portal	3
1.1 Portalansicht.....	3
1.2 Projektansicht	4
1.3 Grundlegende Einstellungen	5
1.4 Fensteraufteilung des Arbeitsbereichs	6
1.5 Stapelung der Objekte im Arbeitsbereich.....	7
1.6 Task Cards	8
1.7 Inspektor-Fenster	9



2 Projekt10	
2.1 Neues Projekt anlegen	10
2.2 Projekt archivieren / dearchivieren	11
2.2.1 Archivieren.....	11
2.2.2 Dearchivieren	11

3 Hardwarekonfiguration	12
--------------------------------------	-----------

3.1 PLC (Programmable logic controller) einfügen	12
3.2 PLC konfigurieren.....	13
3.3 PLC Netzwerkeinstellungen PROFINET (Ethernet) & PROFIBUS	14
3.3.1 Ethernet	15
3.3.2 Profibus- oder MPI-Schnittstelle	16
3.4 IP-Adresse auf der SPS einstellen	18
3.5 HMI einfügen und Schnittstellen konfigurieren.....	19
3.6 Geräte vernetzen (und Schnittstelle konfigurieren)	20
3.6.1 Netzansicht	21
3.6.2 CPU Verbindungen „Netze“.....	22
3.6.3 HMI Verbindung erstellen.....	22
3.7 Dezentrale Peripherie Einfügen & Konfigurieren	23
3.8 Hardware in Gerät laden	25

4 SPS Schaltprogramm	26
-----------------------------------	-----------

4.1 Variablen.....	26
4.1.1 Globale PLC-Variablen (Gegenüberstellung TIA zu Step 7 classic)	26
4.1.2 PLC Variable erstellen	27
4.1.3 Variable beobachten / steuern	28
4.2 Bausteine erstellen	29
4.3 Baustein in OB1 aufrufen	29
4.4 SPS Schaltprogramm erstellen	30
4.5 Programm in Gerät laden	32

4.6	Programm beobachten	32
5	HMI erstellen	33
5.1	HMI Variable erstellen	33
5.2	Bild erstellen	33
5.3	Bedienelement erstellen	34
5.4	Visueller Indikator erstellen	35
6	Anhang (Grundwissen SPS-Steuerungen).....	36
6.1	Was ist eine SPS und wofür wird sie verwendet?	36
6.1.1	Was bedeutet der Begriff SPS?	36
6.1.2	Wie spricht die SPS einzelne Ein-/Ausgangssignale an?	36
6.1.3	Wie wird in der SPS das Programm bearbeitet?	38
6.1.4	Wie sehen logische Verknüpfungen im SPS-Programm aus?	38
6.1.5	Wie wird das SPS-Programm erstellt?	39
6.1.6	Wie kommt es in den Speicher der SPS?	39
6.2	Programmiersprachen	39
6.2.1	Kontaktplan KOP	40
6.2.2	Funktionsplan FUP	41
6.2.3	Anweisungsliste AWL.....	42
6.2.4	Strukturierter Text mit S7-SCL	43
6.2.5	Ablaufsteuerung mit S7-Graph	44
6.3	Merker.....	45
6.3.1	Remanente Speicherbereiche	45
6.4	Anwenderbausteine für die SIMATIC S7-1200	46
6.5	Beispielaufgabe mit Einschaltverzögerung	47
6.5.1	Referenzaufgabe	47
6.5.2	Zeitglied Einschaltverzögerung	47
6.5.3	Evolution der SPS-Programmierung	48
6.5.4	Umsetzung der Referenzaufgabe	48

1 Aufbau TIA Portal

Das TIA-Portal verfügt über eine Portalansicht und eine Projektansicht. Die Portalansicht dient dazu aufgabenorientierte Projektierungsaufgaben durchzuführen. Die Bedienung des TIA-Portals wird in diesem Dokument allerdings hauptsächlich aus der Projektansicht beschrieben, welche gegenüber der Portalansicht eher als Expertenansicht zu verstehen ist.

Sämtliche im Dokument aufgeführten Grafiken sind sinnbildlich zu verstehen. Die Komponentenauswahl im TIA-Portal muss beim Anwenden dieser Anleitung mit der tatsächlich verwendeten Hardware übereinstimmen!

1.1 Portalansicht



Hauptpunkte

- aufgabenorientierte Arbeitsweise
- schneller Projekteinstieg durch geführte Bedienung

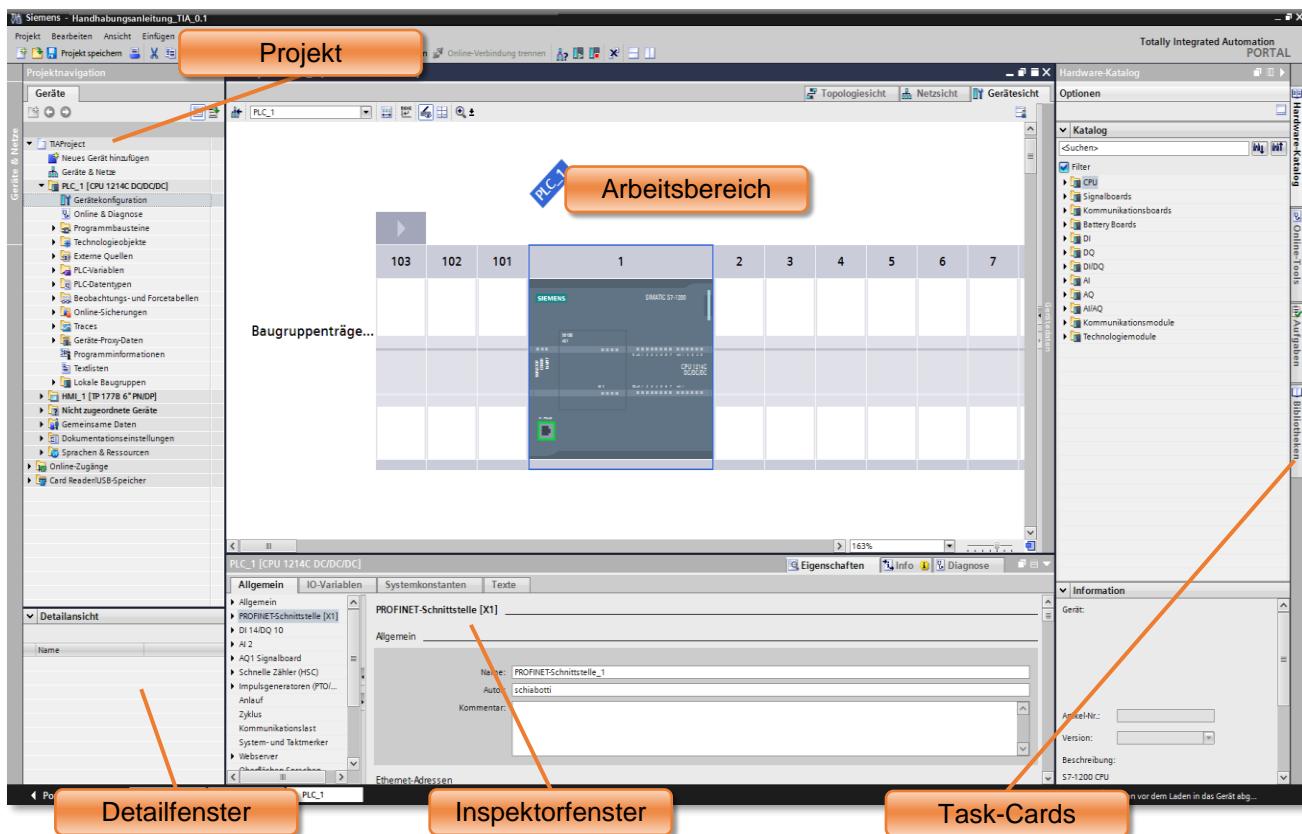
Aktionen

Abhängig vom ausgewählten Portal werden Ihnen hier die Aktionen angeboten, die Sie in diesem Portal ausführen können. Der Aufruf der Hilfe steht Ihnen in jedem Portal kontextsensitiv zur Verfügung.

Auswahlfenster

Das Auswahlfenster steht in allen Portalen zur Verfügung. Der Inhalt des Fensters passt sich der aktuellen Auswahl an.

1.2 Projektansicht



Hauptpunkte

- Projekt ist hierarchisch aufgegliedert
- die benötigten Editoren öffnen sich aufgabenorientiert
- alle Editoren, Parameter und Daten befinden sich in einer Ansicht

Projekt

Ein Projekt beinhaltet alle Komponenten und Projektdaten einer Automatisierungslösung. In der Projekt-Ansicht können z.B. neue Komponenten hinzugefügt oder bereits bestehende Komponenten bearbeitet abgefragt / verändert werden. Die Projekt-Struktur ist wie in einem Dateiexplorer in Ordner, Unterordner und Objekte gegliedert.

Arbeitsbereich

Innerhalb des Arbeitsbereichs werden die Objekte angezeigt, die zur Bearbeitung bereits geöffnet sind. Diese Objekte sind z. B. Hardware-Komponenten, Bausteine, PLC-Variablen-Tabelle, Bilder von HMI-Geräten usw. Sind mehrere Objekte gleichzeitig geöffnet, werden diese in der Taskleiste des Editors als Register angezeigt.

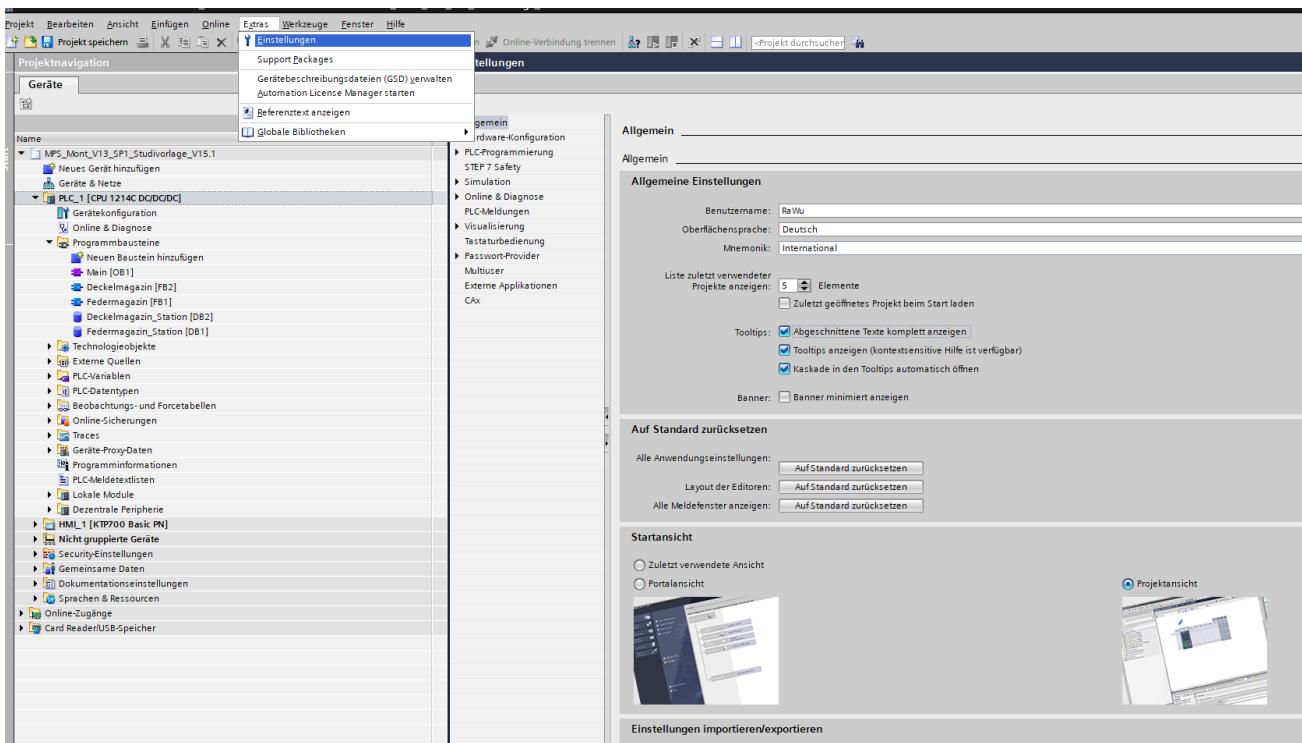
Inspektorenfenster

Im Inspektorenfenster werden zusätzliche Informationen zu einem selektierten Objekt oder zur ausgeführten Aktionen angezeigt.

Task Cards

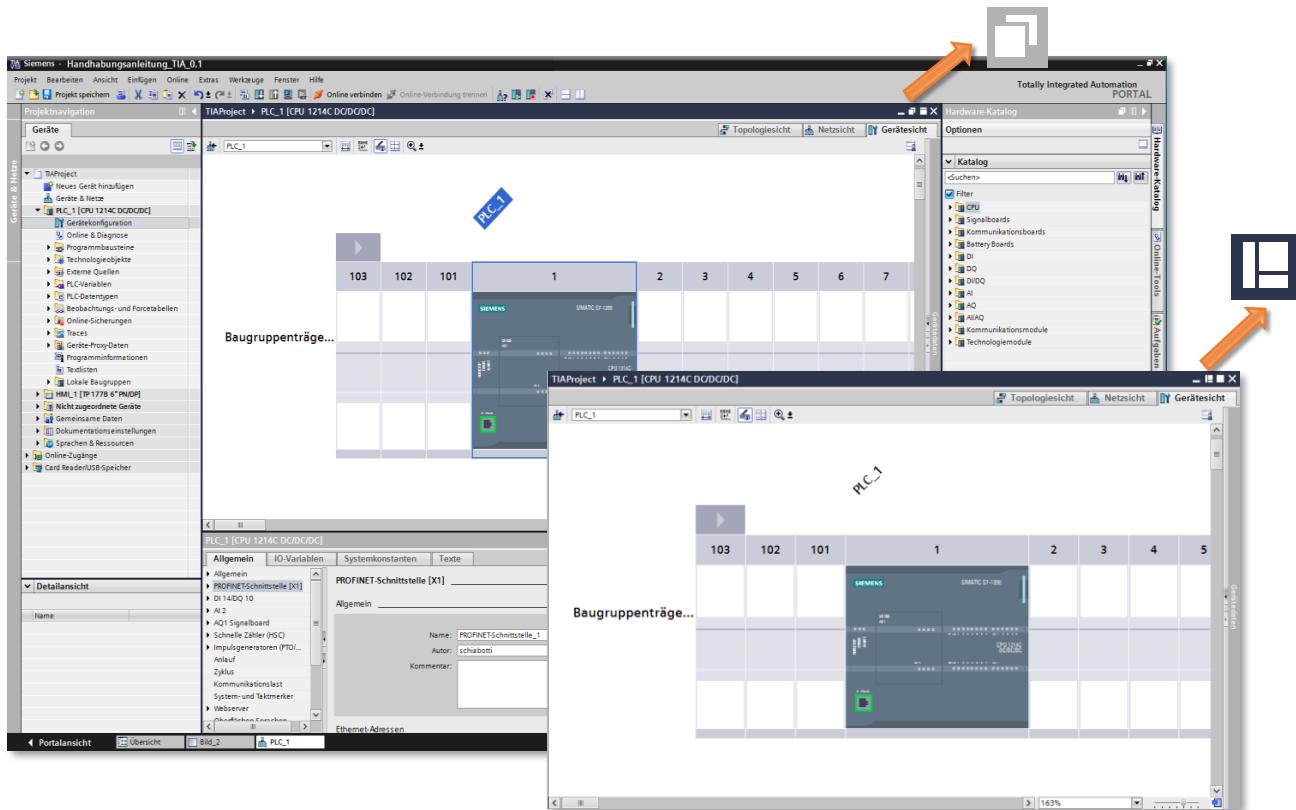
Der Inhalt der Task Cards ist abhängig vom Objekt, das im Arbeitsbereich angezeigt wird. Ist eine Hardware-Station geöffnet, steht z.B. der Hardware-Katalog als Task Card zur Verfügung, ist ein Programmsteckblatt geöffnet, eine Task Card mit Anweisungen.

1.3 Grundlegende Einstellungen



Damit die Sprach- und Programmansichtseinstellungen am Computer mit denjenigen der Handhabungsanleitung und den Praktika übereinstimmen, müssen beim erstmaligen Starten der Software folgende Einstellungen getätigt werden. Klicken Sie in der Projektansicht im Menü «Extras» auf «Einstellungen» und gehen zum Eintrag «Allgemein». Hier wechseln Sie die Oberflächensprache (User interface language) auf Deutsch und wählen als Startansicht die Projektansicht. Die „Mnemonik“ können Sie auf „international“ belassen. Damit werden im Projekt SPS-Ein- und Ausgänge mit „I“ bzw. „Q“ bezeichnet. Bei deutscher Mnemonik heißen sie „E“ bzw. „A“.

1.4 Fensteraufteilung des Arbeitsbereichs



Arbeitsbereich

Die Fenster des Arbeitsbereichs können wie folgt angeordnet werden:



Maximieren eines Fensters



Ablösen bzw. Herauslösen eines Fensters aus dem Arbeitsbereich



Wieder-Einbetten eines Fensters in den Arbeitsbereich



Arbeitsbereich horizontal in zwei Fenster teilen

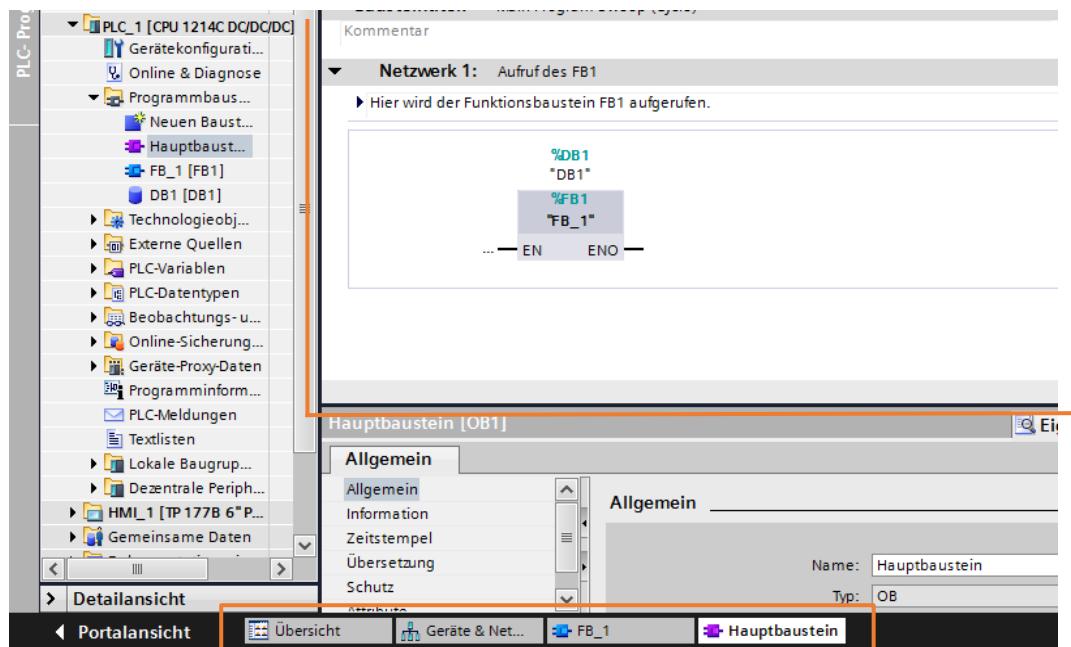


Arbeitsbereich vertikal in zwei Fenster teilen



Fenster Automatisch reduzieren

1.5 Stapelung der Objekte im Arbeitsbereich

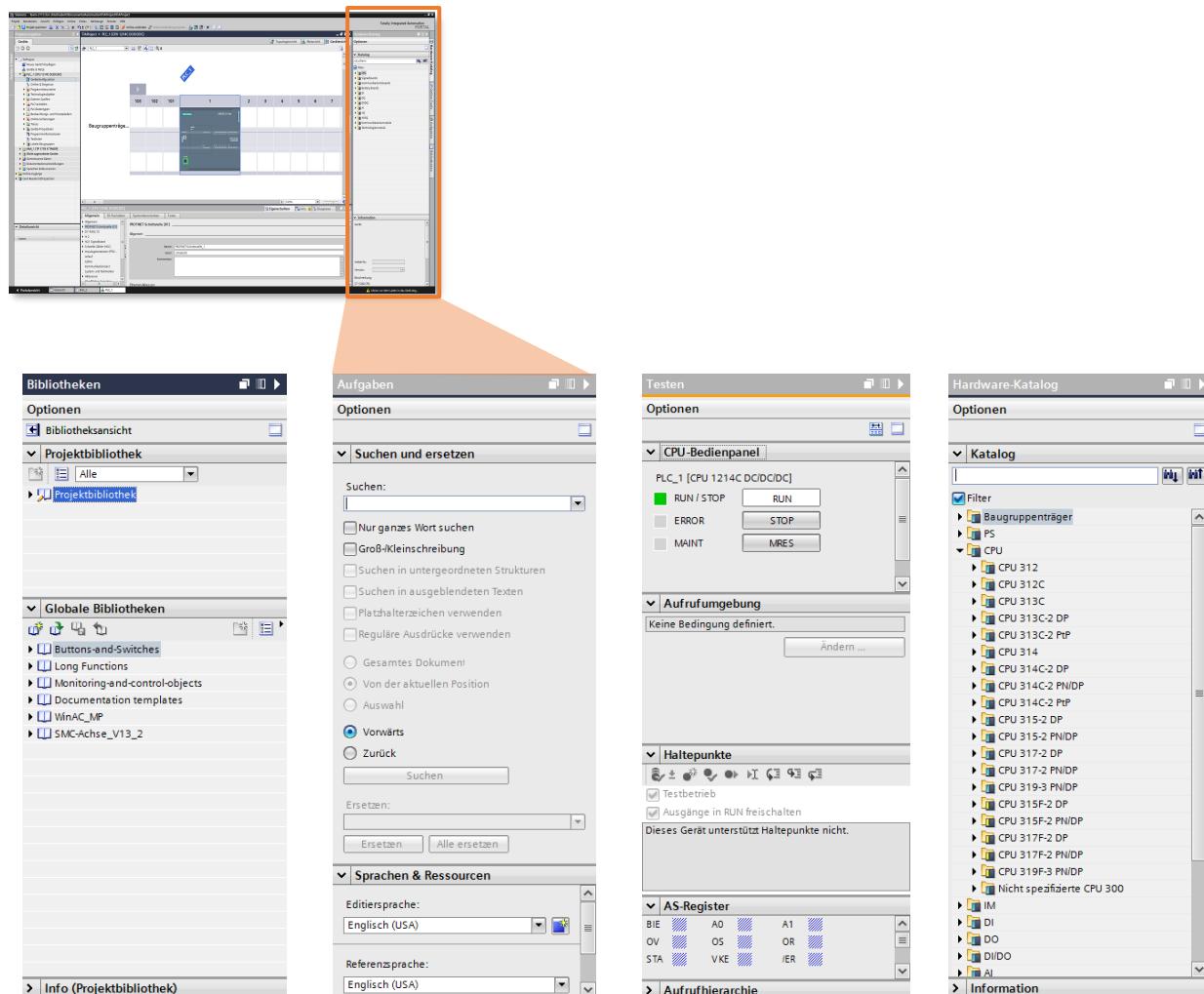


Im Arbeitsbereich ist grundsätzlich immer nur ein Bearbeitungsfester zu sehen, selbst wenn mehrere Bearbeitungsfenster geöffnet sind. Die Bearbeitungsfenster liegen demnach im Arbeitsbereich gestapelt hintereinander.

An der untersten Kante der Software sind die geöffneten Bearbeitungsfenster aufgeführt. Gemäss der obigen Abbildung hat der Anwender ein Fenster für das Einrichten der „Geräte & Netze“, ein Fenster für die Bearbeitung des Funktionsbausteins „FB_1“ und eines für die Bearbeitung des „Hauptbausteins“ geöffnet.

Durch Anwählen der Fensterüberschrift an der untersten Softwarekante kann effizient zwischen den verschiedenen Fenstern gewechselt werden.

1.6 Task Cards



Die verfügbaren Task Cards stehen am rechten Bildschirmrand zur Verfügung und sind jederzeit auf- und zuklappbar. Welche Task Cards zur Verfügung stehen, hängt von den installierten Produkten und vom gerade bearbeiteten bzw. im Arbeitsbereich geöffneten Objekt ab. Über die Task Cards können weitere Aktionen ausgeführt werden:

- **Bibliotheken**

Zum Speichern von Objekten, die wiederverwendet werden sollen, stehen lokale (zum Projekt gehörende) und globale Bibliotheken zur Verfügung. Hier können Objekte jeder Art (ganze Stationen, einzelne Bausteine, Variabellisten usw.) gespeichert und entnommen (kopiert) werden.

- **Aufgaben:**

Hier stehen klassische Editor-Funktionen zur Verfügung wie z.B. Suchen und Ersetzen von Variablen, Anweisungen usw.

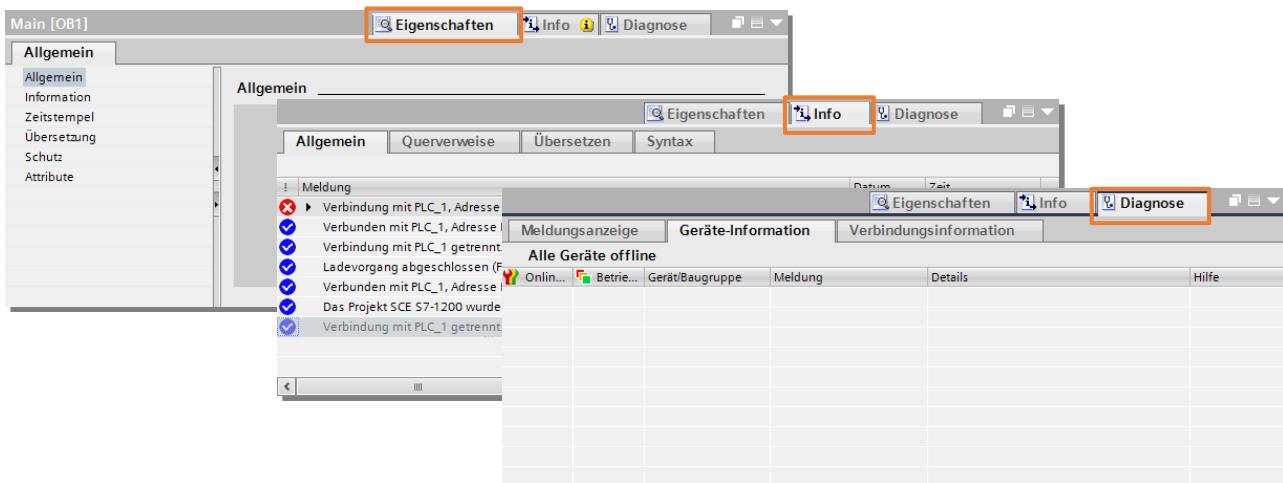
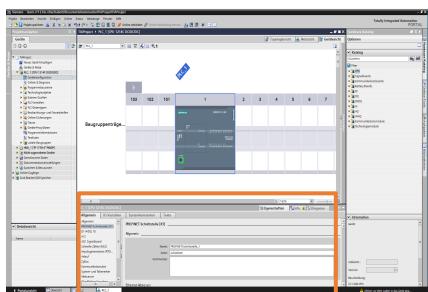
- **Online/Testen:**

Bei bestehender Online-Verbindung sind Diagnose- und Online-Informationen abrufbar wie die aktuelle Zykluszeit der CPU und der Ausbau des Lade- und Arbeitsspeichers der CPU. Außerdem kann die CPU in den STOP- und RUN-Modus geschaltet werden.

- **Hardware-Katalog:**

Hier sind alle zur Verfügung stehenden Hardware-Komponenten wie CPU's, Baugruppen usw. auswählbar

1.7 Inspektor-Fenster



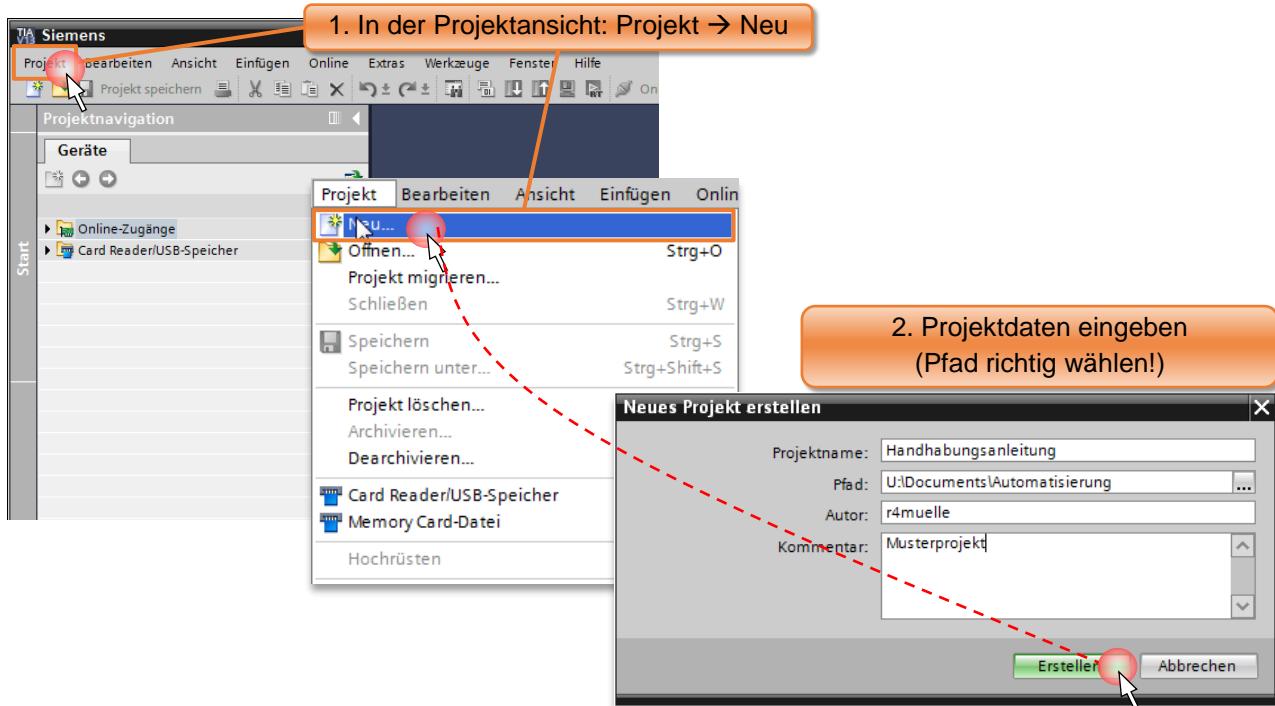
Inspektorfenster

Im Inspektorfenster werden zusätzliche Informationen zu einem selektierten Objekt oder zu ausgeführten Aktionen angezeigt. Das Inspektorfenster besteht aus folgenden Registern:

- Eigenschaften**
 In diesem Register werden die Eigenschaften des selektierten Objekts angezeigt und es können editierbare Eigenschaften verändert werden.
- Info**
 In diesem Register werden weitere Informationen zum selektierten Objekt angezeigt. Außerdem werden Meldungen zu ausgeführten Aktionen z. B. Übersetzen und Laden von Bausteinen in die CPU ausgegeben.
- Diagnose**
 In diesem Register werden Informationen zur Systemdiagnose und projektierten Meldungseignisse angezeigt

2 Projekt

2.1 Neues Projekt anlegen



In der Projektansicht unter Projekt, Neu anwählen. Anschliessend können die Projektdaten eingegeben werden. Dabei ist es wichtig den korrekten Pfad zu wählen.

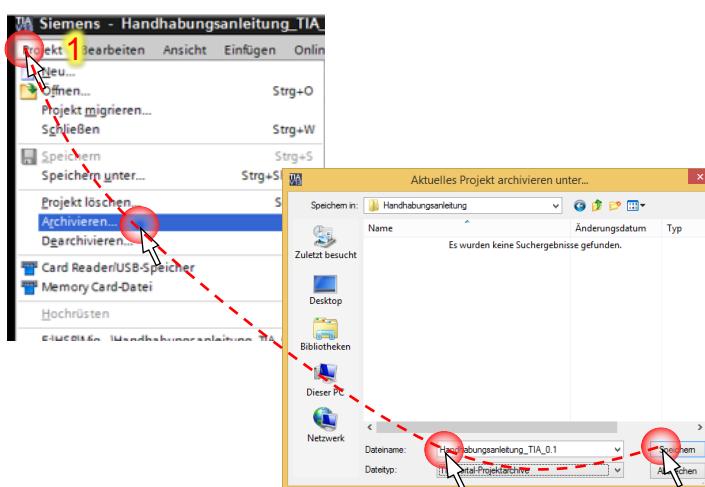
Das nun erstellte Projekt beinhaltet erst die Grundstruktur und noch keinerlei Geräte/Konfigurationen. Diese müssen in einem nächsten Schritt definiert werden.

2.2 Projekt archivieren / dearchivieren

Ein Projekt besteht nicht aus einer einzelnen Datei, sondern aus einer Projektdatei und einer dazugehörigen Ordnerstruktur. Um die Handhabung eines Projekts zu vereinfachen, beispielsweise wenn ein Projekt via Email verschickt werden soll, besteht die Möglichkeit die Projektdatei in einem Archiv zu bündeln. Ein archiviertes Projekt zu einer einzigen Datei zusammengeschnürt „beispiel_projekt.zap15_1“.

2.2.1 Archivieren

Bei der Projektarchivierung werden alle Projektdateien zusammengeschnürt.

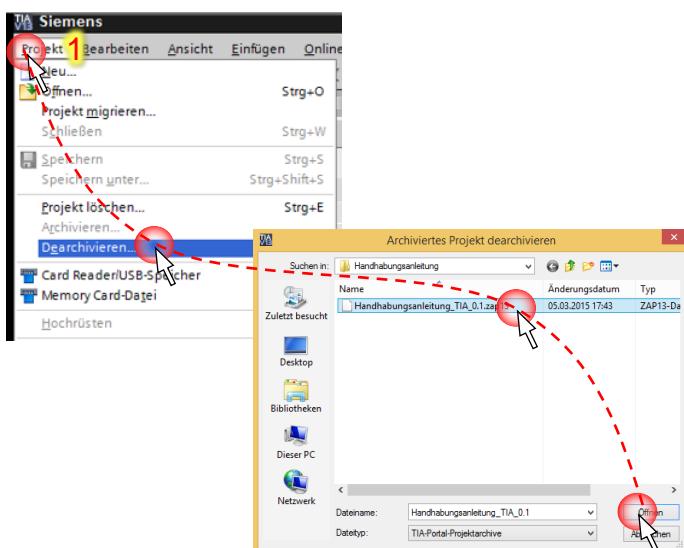


Um ein Projekt zu Archivieren öffnet man das Register *Projekt* und drückt *Archivieren*.

Im Fenster „*Aktuelles Projekt archivieren unter...*“ Dateiname eingeben, Ordner wählen und speichern.

2.2.2 Dearchivieren

Bei der Projektdearchivierung wird ein archiviertes Projekt entpackt.



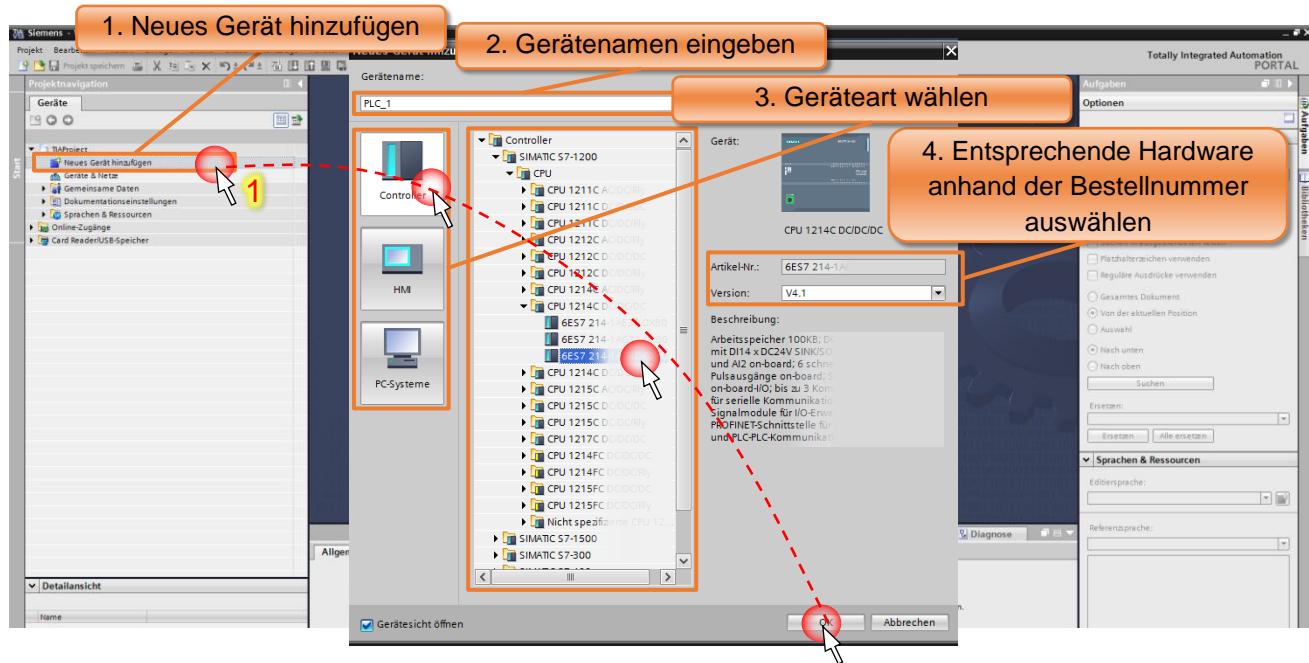
Um ein Projekt zu dearchivieren öffnet man das Register *Projekt* und drückt *Dearchivieren*.

Im Fenster „*Archiviertes Projekt dearchivieren*“ Datei wählen und öffnen.

3 Hardwarekonfiguration

Bevor die eigentliche Programmlogik geschrieben werden kann, muss die verwendete Hardwarekonfiguration im TIA-Portal implementiert werden. Dies umfasst das Einfügen der exakt verwendeten Hardwarekomponenten, deren Namensgebung und Adressierung. Je nach ausgewählter Gerätschaft werden im TIA gerätebezogene Bibliotheken freigeschaltet.

3.1 PLC (Programmable logic controller) einfügen



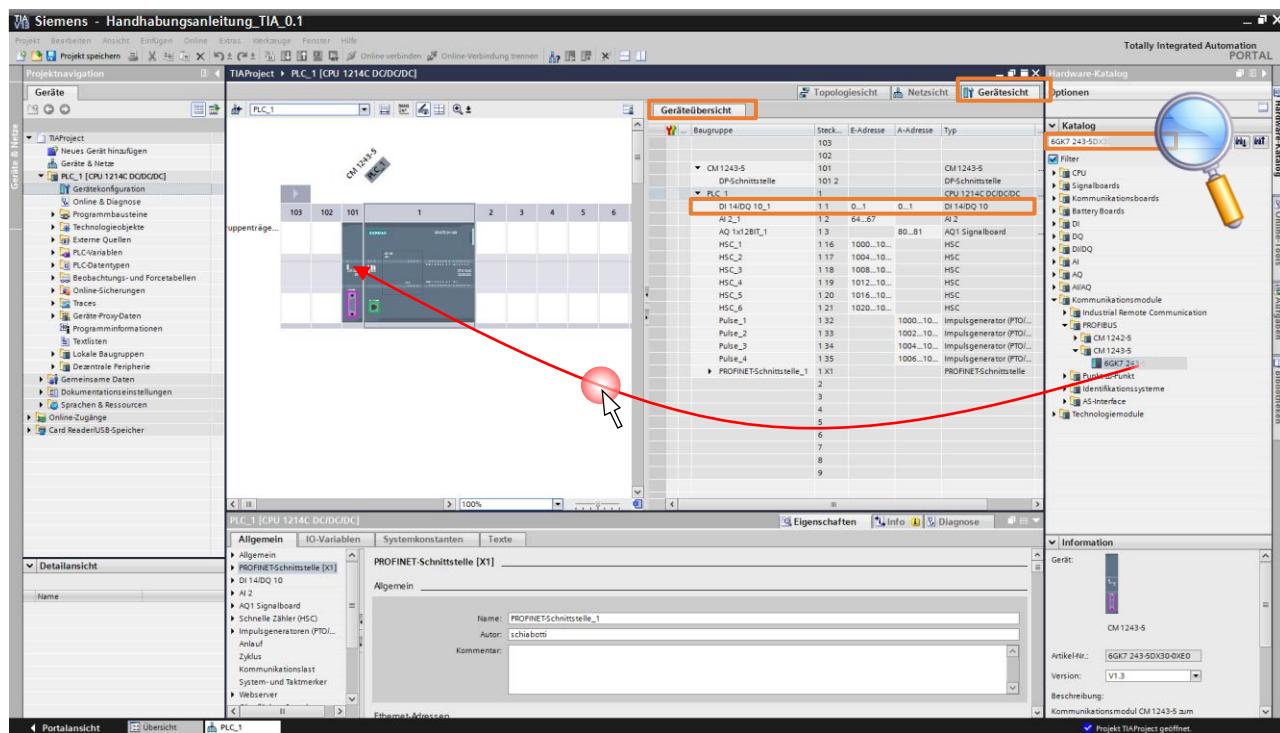
Mit einem Doppelklick auf „Neues Gerät hinzufügen“ erscheint ein Auswahlfenster, aus welchem die gewünschte Hauptgerätschaft angewählt werden kann. Hierbei wird zwischen Controller, HMI oder PC-Systeme unterschieden.

Bei dieser grundlegenden Geräteauswahl kann die Wunschkomponente nicht über eine Eingabe (wie beispielsweise die Bestellnummer oder die Gerätebezeichnung) in eine Suchfunktion gefunden werden. Bei diesem Arbeitsschritt muss der Anwender sich zwingend durch den Katalog klicken.

Im obigen Beispiel wird ein Controller ausgewählt.

Nach der Bestätigung durch OK wird die ausgewählte Komponente, wie im Kapitel 3.2 dargestellt, auf eine sinnbildliche Profilschiene abgebildet. Diese Profilschiene besitzt verschiedene Steckplätze. Komponenten wie beispielsweise gewisse CPUs werden automatisch am Steckplatz 2 eingefügt. Die dazu passenden Stromversorgungen sind so konstruiert, dass diese nur auf der linken Seite der CPU verschraubt werden kann. Folglich muss diese in der Software im Steckplatz 1 platziert werden, wenn die CPU für den Steckplatz 2 prädestiniert ist.

3.2 PLC konfigurieren

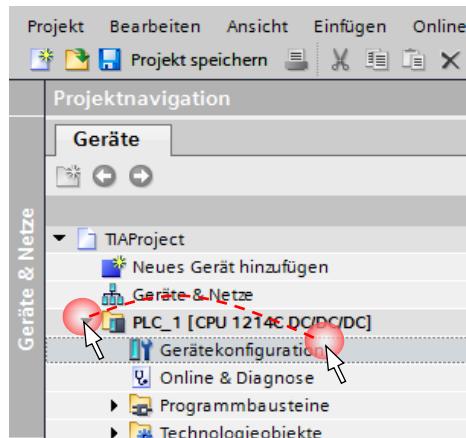


Nachdem ein Gerät aus dem Katalog eingefügt wurde (gemäß vorhergehendem Kapitel), öffnet die Software automatisch die Gerätekonfiguration (Gerätesicht) des soeben eingefügten Geräts. Hier können nun weitere Module (Zubehör) aus dem Hardware-Katalog (Task-Card rechts) hinzugefügt werden.

Über die „Suche“  im Katalog lassen sich nun auch einfach Komponenten mittels einer Bestellnummer finden.

In der Geräteübersicht sind gewisse Komponenteneigenschaften ersichtlich. Beispielsweise sind hier die CPU integrierten Eingänge mit den Adressen %I0 - %I1 und die integrierten Ausgänge mit den Adressen %Q0 - %Q1 konfiguriert. Weitere Informationen betreffend der Hardware Eigenschaften und Parameter Konfiguration können der Simatic Hilfe (Taste „F1“) entnommen werden.

Eine Gerätekonfiguration kann auch unabhängig/nachträglich über die Gerätekonfiguration in der Projektnavigation aufgerufen werden.

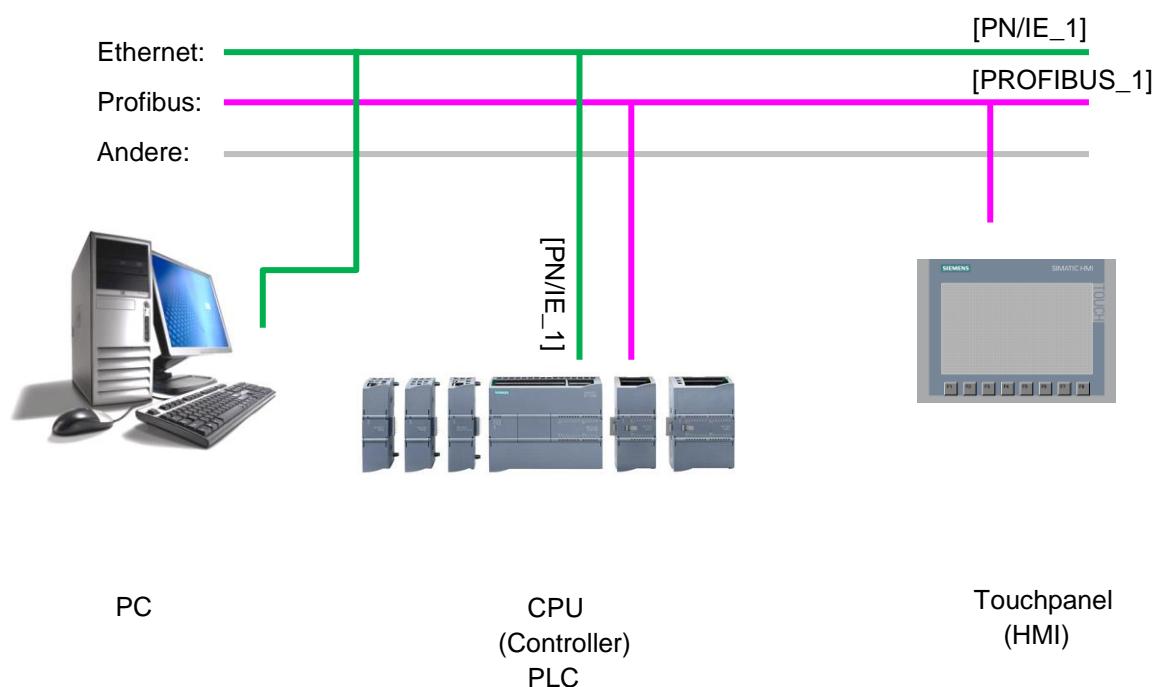


3.3 PLC Netzwerkeinstellungen PROFINET (Ethernet) & PROFIBUS

Damit der PC oder das Programmiergerät mit der CPU übers Ethernet kommunizieren kann, muss der CPU eine IP-Adresse zugeordnet werden und ein Ethernet-Subnetz als Kommunikationskanal (PN/IE_1) angelegt werden. Falls noch kein Subnetz aus einer vorgängigen Aktion vorhanden ist, muss der Anwender explizit ein Ethernet-Subnetz (beispielsweise PN/IE_1 gemäss automatischer Namensgebung) erstellen.

Bei nachgehenden Einstellungen/Konfiguration oder Geräte-Anbindungen, wird der Anwender immer wieder auf das zuvor erstellte Ethernet-Subnetz Bezug nehmen müssen. Bei grösseren Anlagen können auch mehrere Subnetze angelegt werden/vorliegen.

Das Gleiche gilt für das Profibusnetz. Auch dieses muss bei Bedarf explizit angelegt werden, falls dies nicht schon implizit durch das Hinzufügen von Profibusgerätschaften passiert ist.

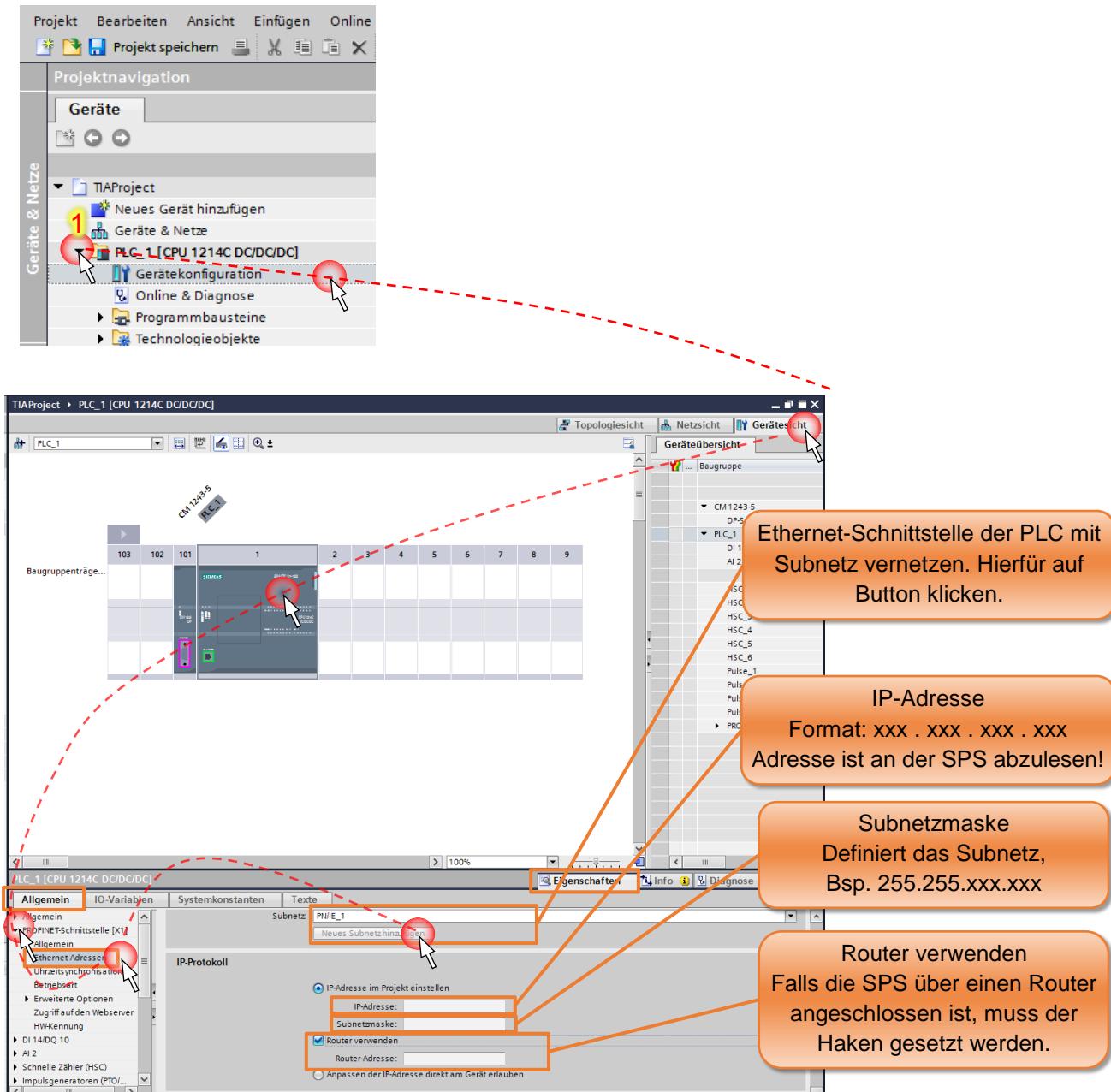


PC

CPU
(Controller)
PLC

Touchpanel
(HMI)

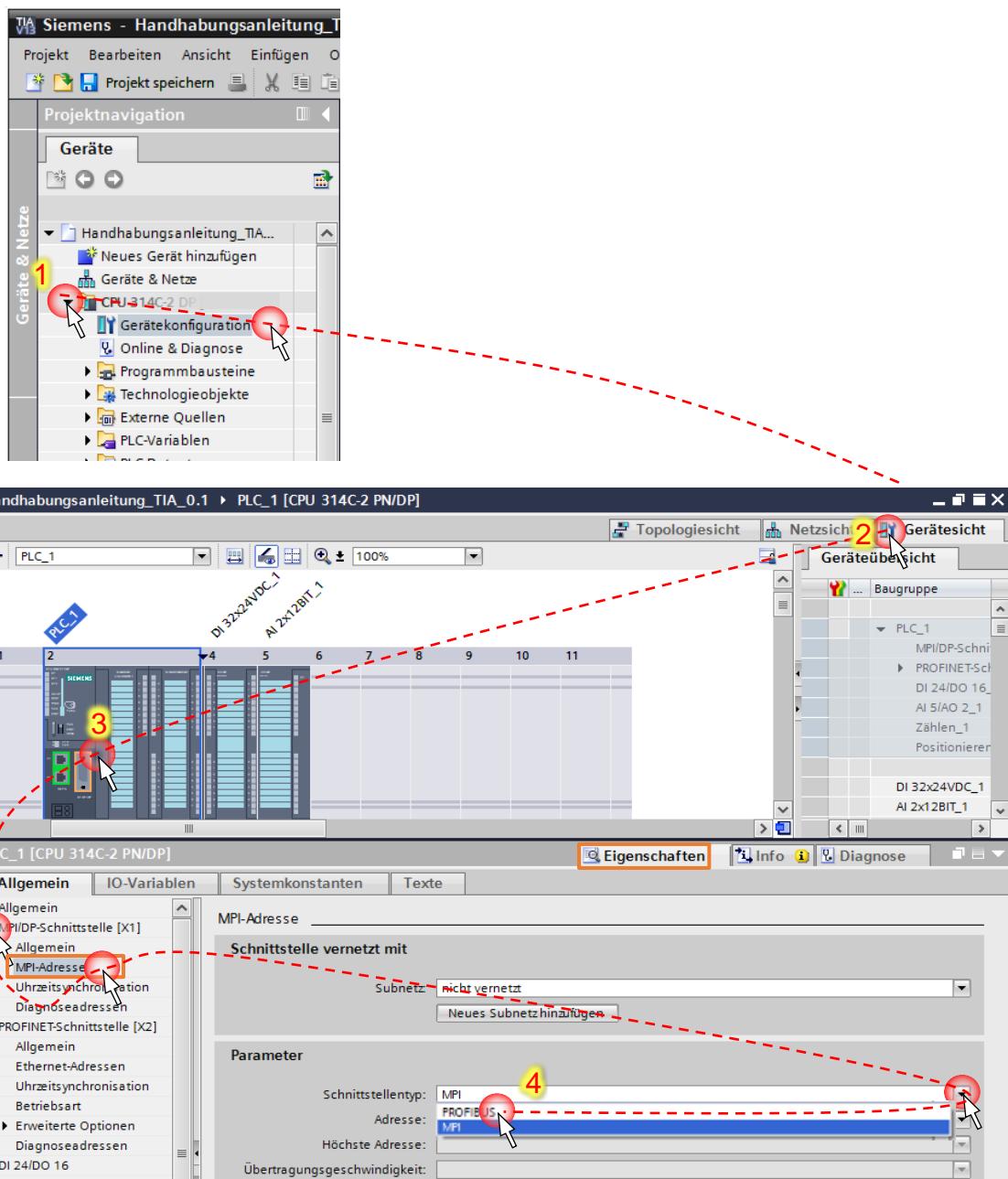
3.3.1 Ethernet



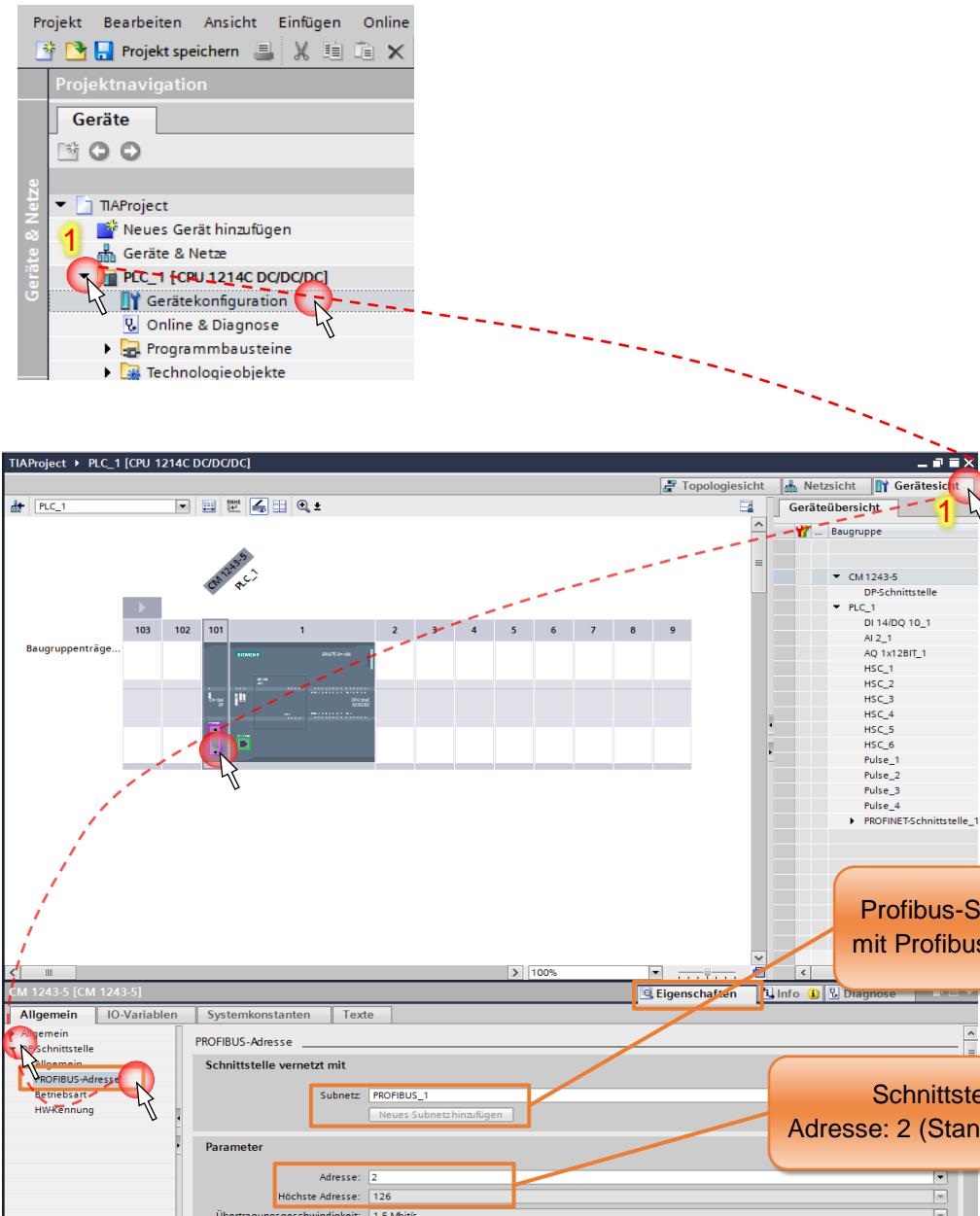
- Damit das TIA-Portal später auf die richtige PLC zugreift, muss deren IP-Adresse und die Subnetzmaske im TIA-Portal eingestellt werden. In der Geräteansicht (Gerätekonfiguration) die CPU anwählen → Eigenschaften → Allgemein → Profinet-Schnittstelle → Ethernet-Adressen.
- Um mit anderen Geräten „Slaves“ zu kommunizieren, die mit der CPU vernetzt sind und über den PC Programmiert werden sollen, muss ein Ethernet Netz (PN/IE_1) projektiert werden, auch wenn vorerst kein Slave im Netz vorhanden ist. Um das Subnetz zu Projektieren muss auf die Schaltfläche „Neues Subnetz hinzufügen“ geklickt werden.

3.3.2 Profibus- oder MPI-Schnittstelle

Die MPI-Schnittstelle ist eine spezifische Siemens-Schnittstelle, welche mit der standardisierten Profibus-Schnittstelle verwandt ist. Die MPI-Schnittstelle ist nur noch bei älteren CPU-Ausführungen zu finden. Je nach Artikel verfügen die CPUs über einen fest eingebauten Steckanschluss (3), welcher wahlweise (4) als Profibus- oder MPI-Schnittstelle konfiguriert werden kann. Verfügt die CPU nicht über einen fest eingebauten Anschluss, wird die Profibusfunktionalität über ein Zusatzmodul ermöglicht. Bei letzterer Variante muss im TIA, für die Konfiguration der Schnittstelle, sinngemäss der entsprechende Steckanschluss am Zusatzmodul (3) angeklickt werden.



Die PLC ist, meistens, der Master am PROFIBUS Netz und daher muss die PROFIBUS Adresse und das zugehörige Netz einrichtet werden. Zuerst wird die Schnittstelle ausgewählt (MPI oder PROFIBUS), danach wird die gewünschte Adresse der PLC eingestellt. Um ein PROFIBUS Netz einzurichten muss auf die Schaltfläche „Neues Subnetz hinzufügen“ geklickt werden.



Wenn die PROFIBUS Schnittstelle schon vorgängig eingerichtet wurde, liegt der Registername **PROFIBUS-Adresse** statt **MPI-Adresse** vor.

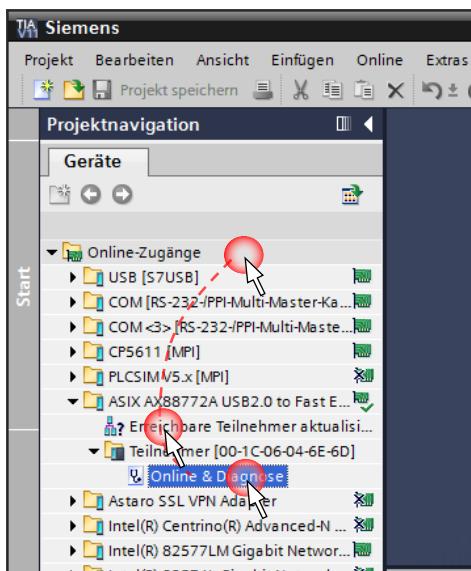
3.4 IP-Adresse auf der SPS einstellen

Wenn eine CPU oder eine sonstige netzwerkfähige Komponente in Betrieb genommen wird, so ist diese ab Fabrik mit einer Default-IP-Adresse konfiguriert. Sobald die neue Komponente in ein bestehendes Netzwerk integriert werden soll, muss ihre IP-Adresse auf die vorliegenden Netzwerkgegebenheiten angepasst werden.

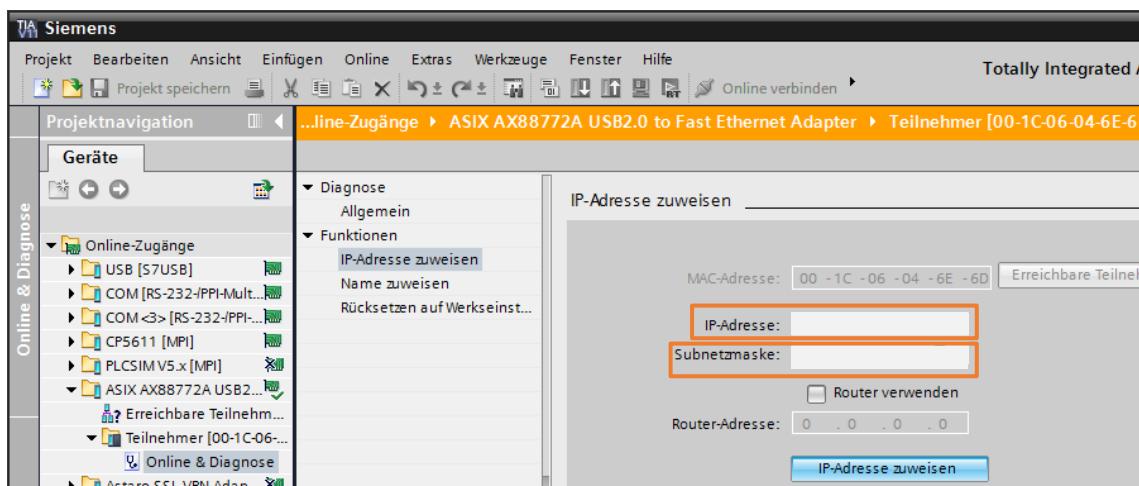
1. Gehen Sie in die Projektansicht

2. In der Projektnavigation wählen Sie dann unter „Online-Zugängen“, die Netzwerkkarte die bereits eingestellt wurde. Wenn Sie hier auf „Erreichbare Teilnehmer aktualisieren“ klicken, dann sehen Sie die MAC-Adresse der angeschlossenen SIMATIC S7-SPS. Wählen Sie hier „Online & Diagnose“.

Hinweis: Wurde bei der CPU bereits vorher eine IP-Adresse eingestellt, so sehen Sie diese anstelle der MAC-Adresse.



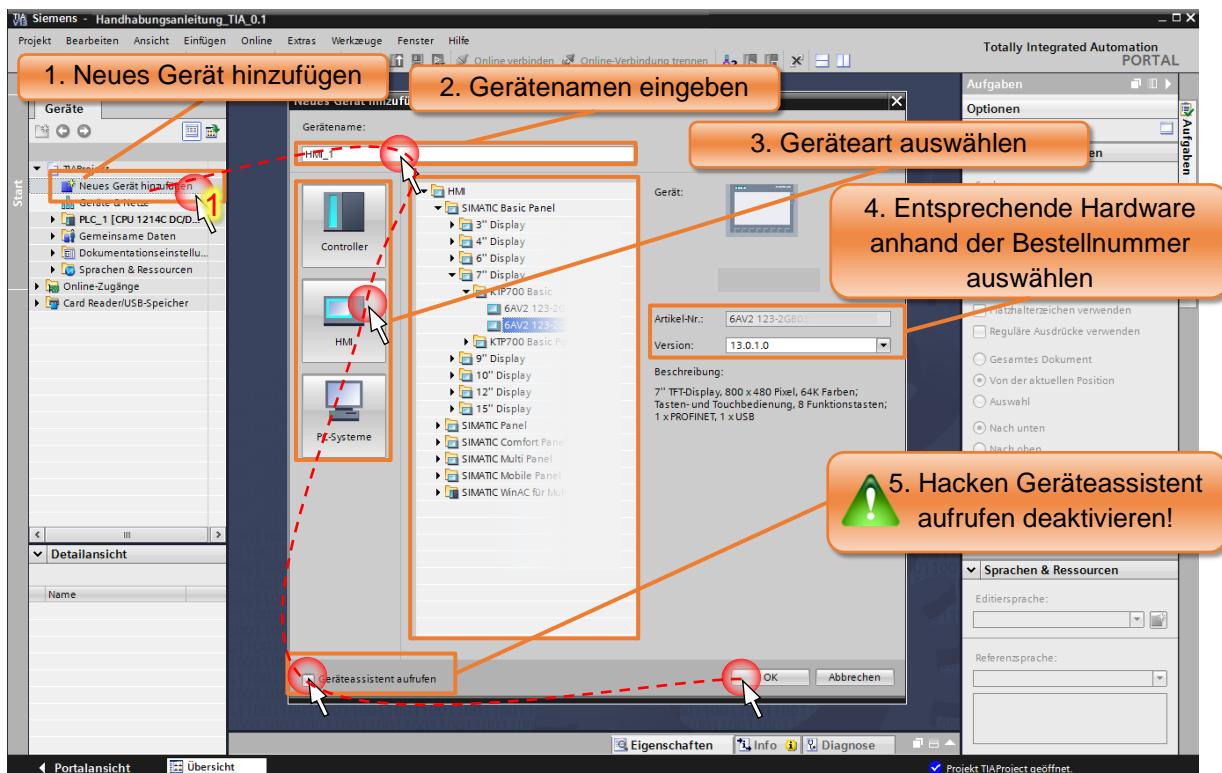
3. Unter „Funktionen“ finden Sie den Punkt „IP-Adresse zuweisen“. Geben Sie hier „IP-Adresse“ und „Subnetz-Maske“ ein. Klicken Sie dann auf „IP-Adresse zuweisen“ und Ihrer SIMATIC S7-SPS wird diese neue Adresse zugewiesen.



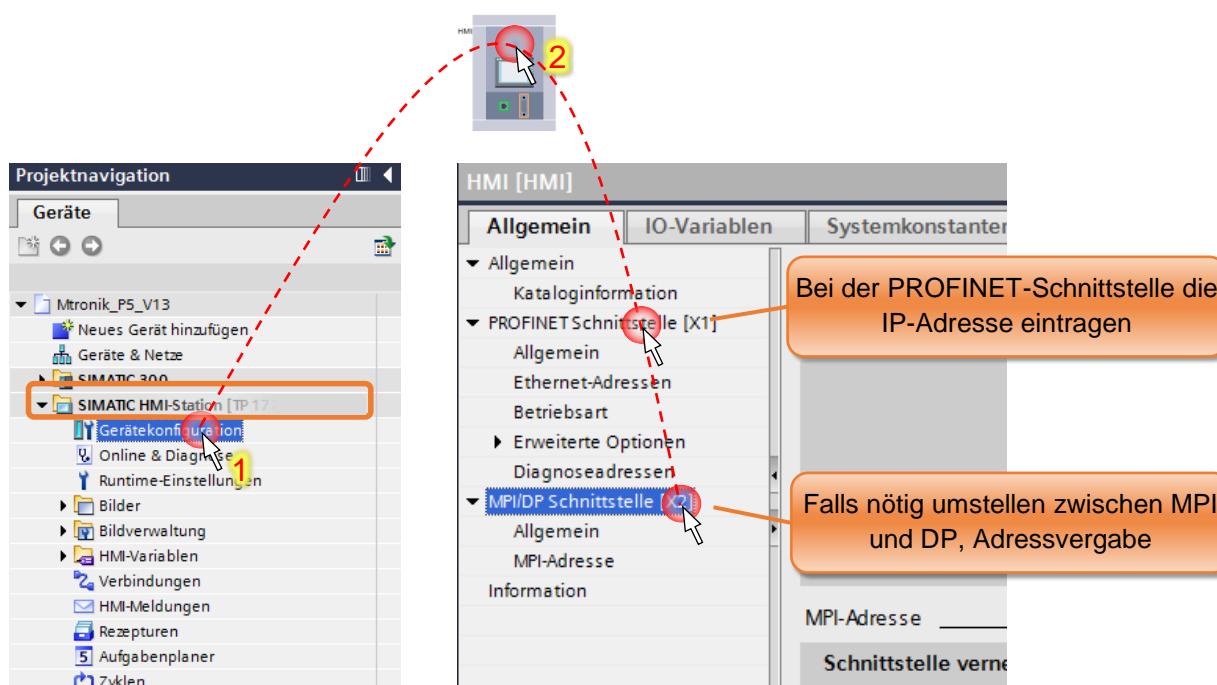
4. Kontrolle, wenn sie nun wieder in der Projektnavigation unter „Online-Zugängen“ auf „Erreichbare Teilnehmer aktualisieren“ doppelklicken, dann sehen Sie die CPU mit der vorhin zugewiesenen IP-Adresse.

3.5 HMI einfügen und Schnittstellen konfigurieren

Ein HMI-Gerät (Touchpanel) wird analog wie ein Controller mit einem Doppelklick auf „Neues Gerät hinzufügen“ eingefügt.



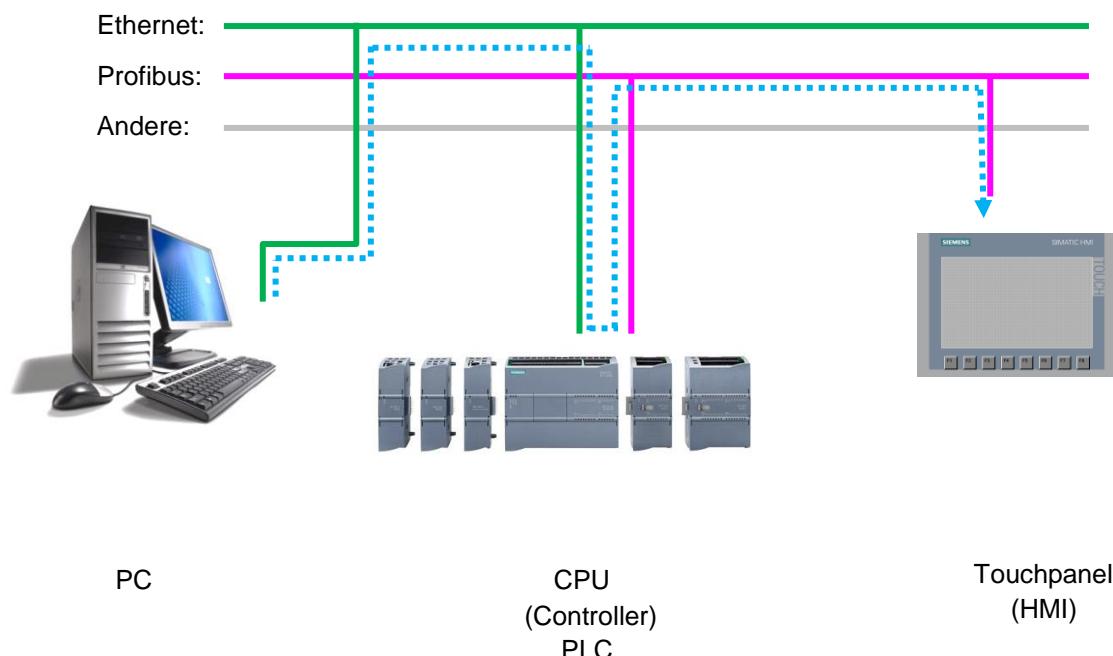
Nach der Bestätigung durch „OK“ erscheint in der Projektnavigation die HMI-Station. Mit einem Doppelklick auf die Gerätekonfiguration (1) und einem Klick auf das HMI-Gerät (2), können im Detailfenster die Ethernet und Profibus Parameter eingetragen werden.



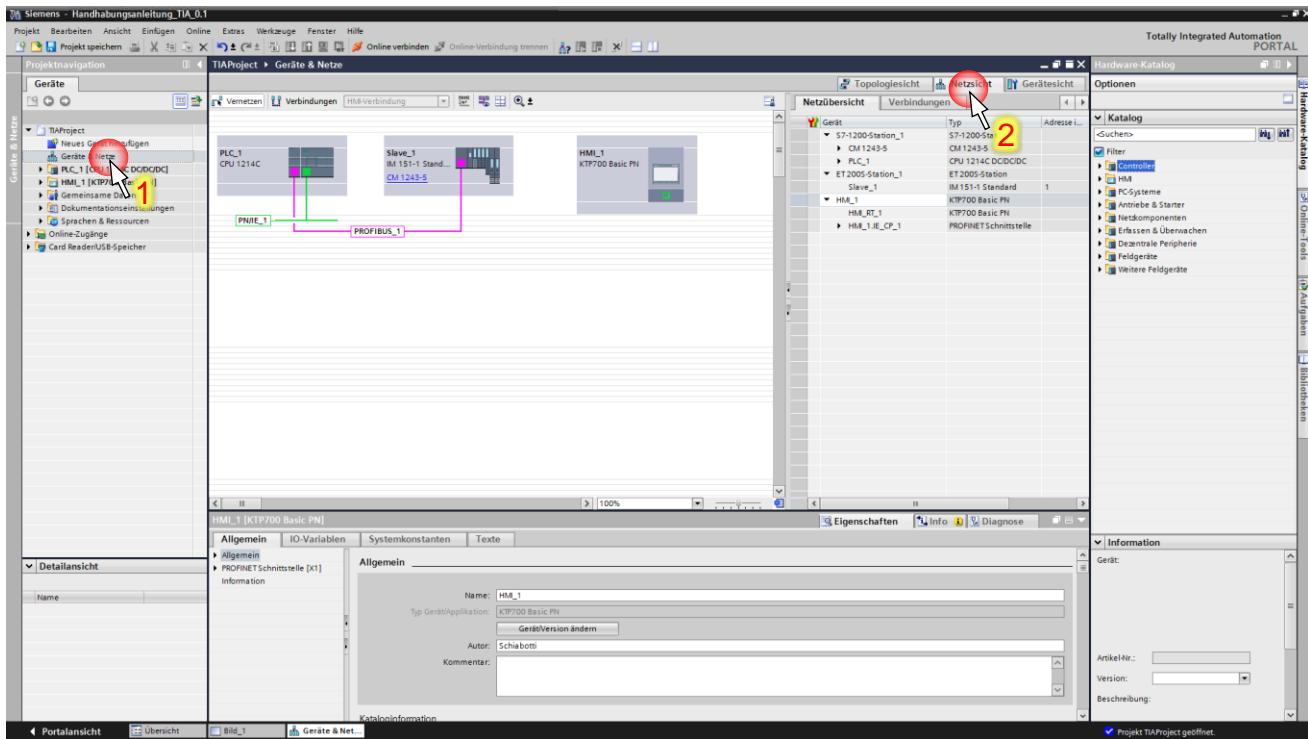
3.6 Geräte vernetzen (und Schnittstelle konfigurieren)

Einleitung:

Damit sämtliche Nachrichten/Befehle korrekt adressiert werden, müssen alle vernetzten Gerätschaften mit einer eindeutigen Adresse versehen werden. Wenn der Anwender beispielsweise eine grafische Oberfläche auf ein Touchpanel laden will, welches nicht direkt mit dem PC (TIA-Portal) vernetzt ist, bedarf es einer konsistenten Vernetzung aller Teilnehmer (Gerätschaften). Nur so weiss die CPU, dass sie für die Erschliessung von PC zu Touchpanel-Nachrichten, das Ethernet-Netz mit dem Profibus-Netz routen muss.



3.6.1 Netzansicht



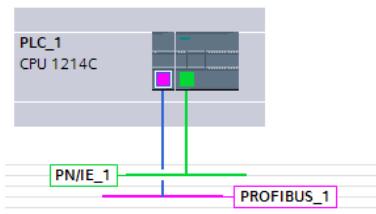
Generell werden Gerätevernetzungen in der Netzansicht getätigt. (1) Diese kann im Projektfenster mit einem Doppelklick auf „Geräte und Netze“ aufgerufen werden, (2) oder auch durch umwählen der Ansicht im Arbeitsfenster.

In der Netzansicht sind alle Geräte und deren Vernetzung aufgeführt.

3.6.2 CPU Verbindungen „Netze“

Die Subnetze, PROFIBUS und Ethernet, der CPU sind im Kapitel 3.2 eingerichtet worden. Daher sollten die beiden Netze wie nebenstehendes Bild dargestellt sein.

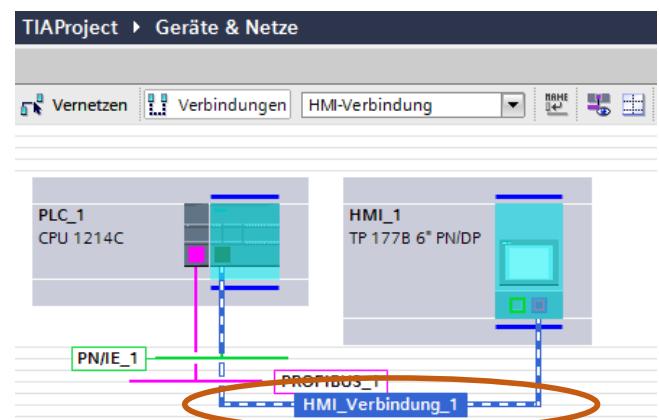
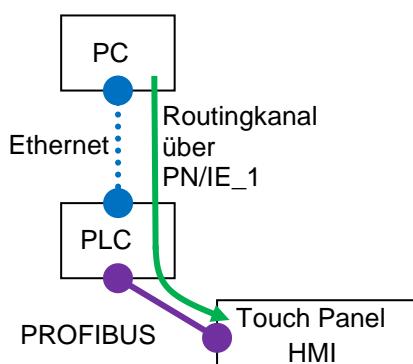
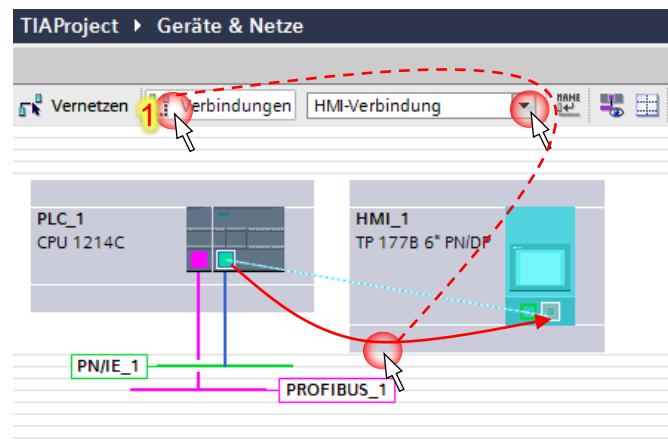
! In Verbindung mit der Kommunikation über Ethernet, zum Beispiel mit einem HMI, ist es notwendig die Ethernet Schnittstelle an der CPU zu projektieren und nicht nur einzurichten.



3.6.3 HMI Verbindung erstellen

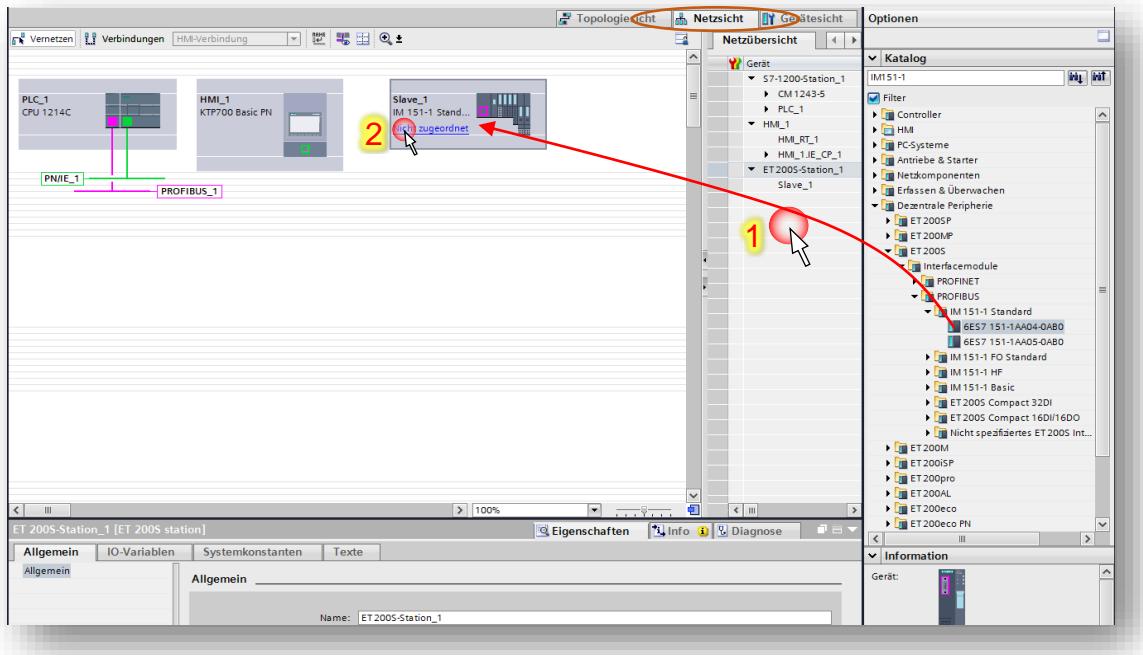
In der HMI Verbindung zu der PLC ist die PROFIBUS Verbindung integriert, daher ist sie nicht Einzel sichtbar.

Das HMI braucht diese Verbindung nicht nur um mit der PLC zu kommunizieren, sondern auch zur Kommunikation mit dem PC. Die PLC dient als Router zwischen dem PN/IE_1 (Ethernet) Netz und dem PROFIBUS_1 Netz. Um die zwei Geräte mit der HMI-Verbindung zu vernetzen, wird eine Verbindung zwischen den Violetten PROFIBUS Buchsen der CPU und des HMI's gezogen.

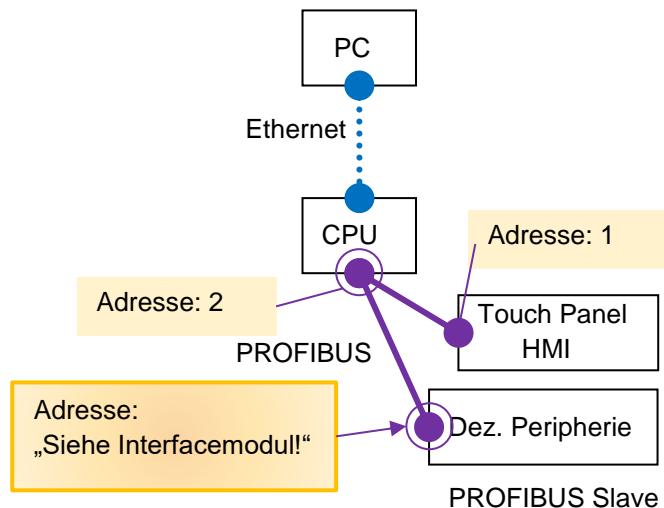


3.7 Dezentrale Peripherie Einfügen & Konfigurieren

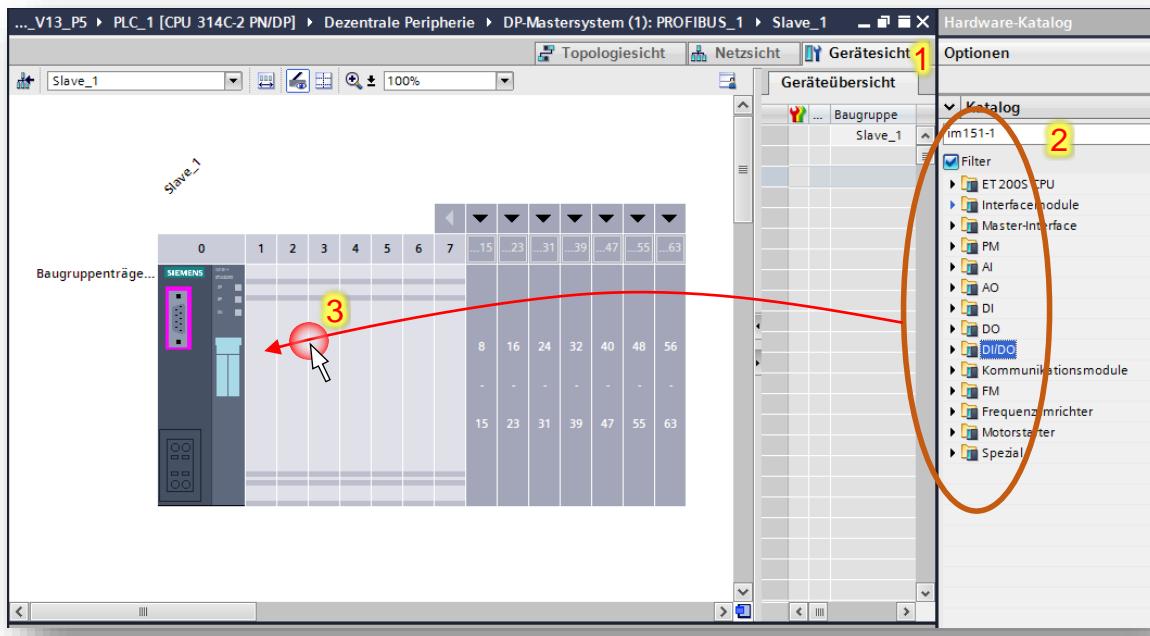
Wählen Sie im Hauptfenster die *Netzsicht*.



1. Ziehen Sie das Interfacemodul IM151-1 in das Fenster.
2. Klicken Sie, beim Interfacemodul IM151-1, auf *Nicht zugeordnet* und wählen sie den Master aus. So wird das Modul mit der CPU verbunden.
3. Mit einem Doppelklick auf das IM151-1 Modul kommen Sie in die *Gerätesicht* des Interfacemoduls.
4. Die untere Grafik zeigt die momentane Profibus-Konfiguration auf, wonach die CPU mit der Adresse Nr. 2 und das HMI mit der Adresse Nr. 1 konfiguriert ist. Definieren Sie im Register *Eigenschaften*, *Allgemein*, *PROFIBUS-Adresse* eine PROFIBUS-Adresse für die dezentrale Peripherie. Diese zu verwendende Adressnummer wurde am Interfacemodul voreingestellt und kann dort abgelesen werden.

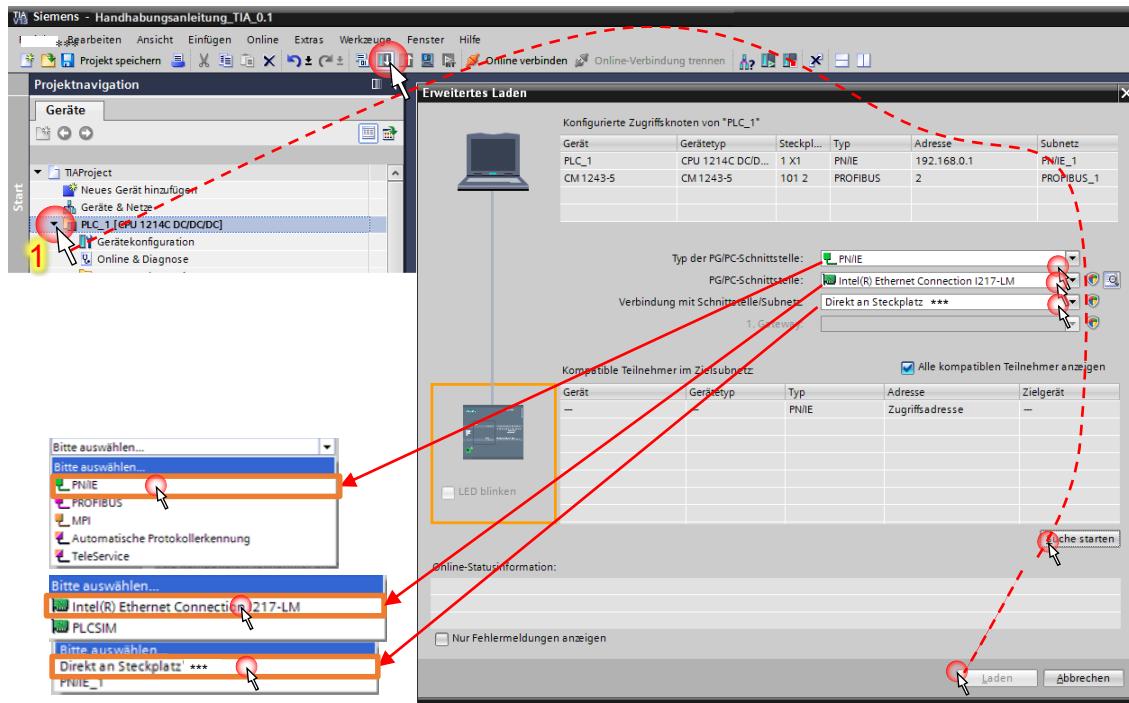


5. Rüsten Sie die dezentrale Peripherie noch mit dem Powermodul sowie den Digitaleingabe- und Digitalausgabemodulen aus. Die Module können per Drag & Drop beim korrekten Steckplatz eingefügt werden.



Speichern und übersetzen Sie Ihre Hardware-Konfiguration ().

3.8 Hardware in Gerät laden



Um die Hardwarekonfiguration in das Gerät zu laden klicken Sie auf das *PLC* Register und auf „Laden in Gerät“ .

Danach öffnet sich das Fenster „Erweitertes Laden“, indem man die die Schnittstelle von dem PG/PC einstellen muss. Als erstes wählen Sie der *Typ der PG/PC Schnittstelle* PN/IE, danach die *PG/PC-Schnittstelle* Netzwerkkarte, zuletzt die *Verbindung mit Schnittstelle/Subnetz* Direkt am vorgegebenen Steckplatz.

Ein Klick auf „Suche starten“ zeigt einige Sekunden die erreichbaren Steuerungen im Netzwerk an. Markieren Sie das richtige Gerät und drücken Sie auf „Laden“.

Wenn Sie zum ersten Mal ein Projekt auf diese Steuerung laden, muss in einem darauf folgenden Fenster noch das Überschreiben der vorhandenen Daten etc. bestätigt werden. Ansonsten wird das Laden nicht ausgeführt.

4 SPS Schaltprogramm

4.1 Variablen

4.1.1 Globale PLC-Variablen (Gegenüberstellung TIA zu Step 7 classic)

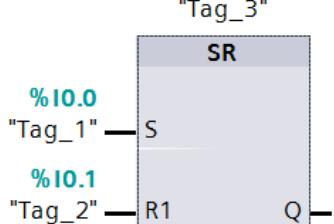
Die erste TIA-Entwicklungsumgebung ist im Jahre 2011 als TIA11 erschienen. Zuvor wurden die Siemensprodukte mit der Software „SIMATIC Step7 Classic“ programmiert. Beim Wechsel zum TIA hat ein Technologieschnitt stattgefunden.

In der vorgängigen Entwicklungsumgebung hat der Anwender selber entscheiden müssen, welchen Merker/Variable (bsp. M1.2) für welchen Zweck eingesetzt wurde. Diesem Merker kann ein Name zugeordnet werden, jedoch wird diese Variable nur über die Merkeradresse identifiziert. Tauscht demnach der Anwender bei der Programmierung eine Merkeradresse (von M1.2 auf M1.1), so muss er sich selber darum kümmern diese Adressänderung bei jeder einzelnen Verwendung im ganzen Projekt sinnvollerweise nachzuführen.

Adresse	Name
M1.0	...
M1.1	Alarmvariable
M1.2	Meldespeicher
M1.3	...

Name	Adresse
...	M1.0
Alarmvariable	M1.1
Meldespeicher	M1.2
...	M1.3

Seit der TIA-Generation, werden Variablen mit einer sprechenden Namensbezeichnung (wie bei der textlichen Programmierung, C/C++) angelegt. Dabei spielt die dafür verwendete Adresse eine sekundäre Rolle und wird grundsätzlich vom System verwaltet. Die Programmierung ist in diesem Fall namensgebunden. Tauscht in diesem Fall der Anwender die Variablen-Adresse, so wird diese Änderung durchgängig für jede einzelne Verwendung im Projekt nachgeführt/umverdrahtet.

Beispiel absoluter Programmierung:	Beispiel namensgebundene Programmierung:
<p>%Q0.0</p> 	<p>%Q0.0 "Tag_3"</p> 

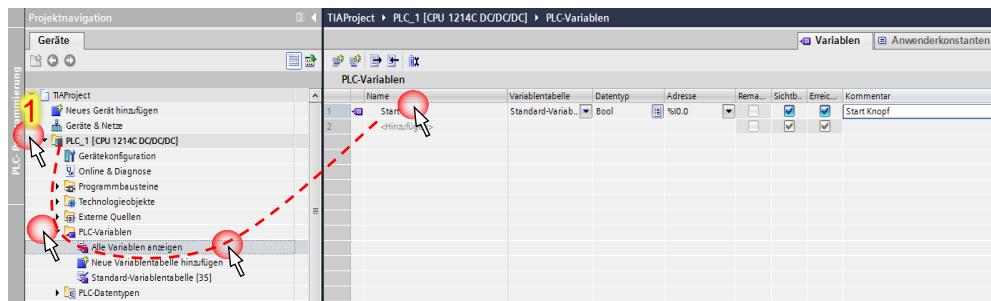
Das TIA-Portal ist prinzipiell nicht für die Programmierung mit absoluter Adressierung ausgelegt. Deshalb wird bei Eingabe eines absoluten Operanden (Eingang, Ausgang, Merker etc.) automatisch ein symbolischer Name (Tag_1, Tag_2, Tag_3) zugewiesen:

4.1.2 PLC Variable erstellen

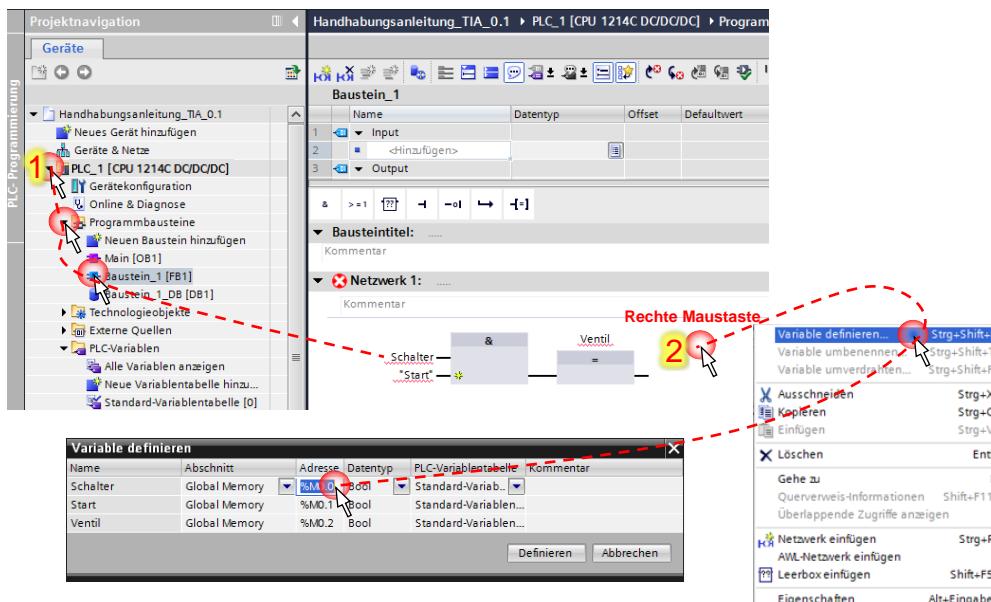
Die globalen PLC-Variablen sind beschreibende Namen mit Kommentar für jene Eingänge, Ausgänge und Merker die im Programm Verwendung finden. Später kann bei der Programmierung über diesen Namen auf die globalen PLC-Variablen zugegriffen werden.

Diese globalen Variablen sind im gesamten Programm in allen Bausteinen verwendbar.

PLC Variablen werden in der Variabellabelle oder direkt im Programm erstellt.

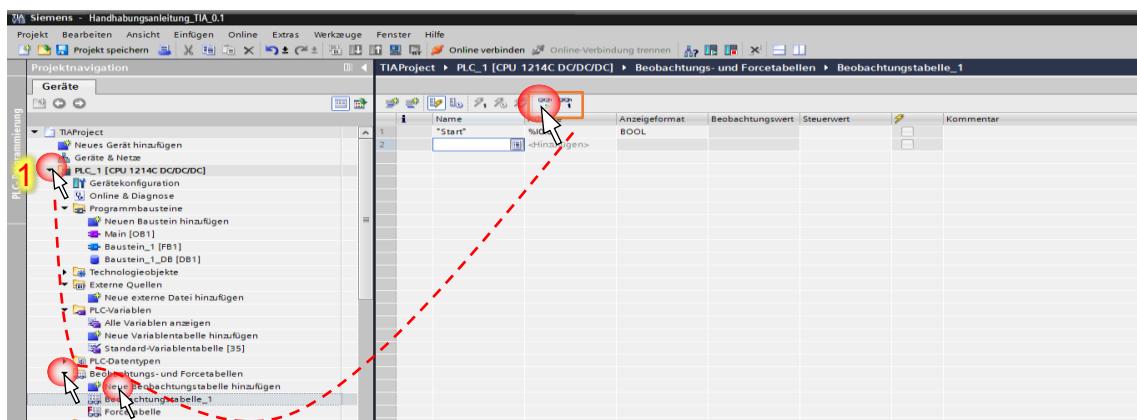


Um die PLC Variablen in der Variabellabelle zu erstellen gehen Sie im *PLC Register*, *PLC Variablen* in die Tabelle „Alle Variablen Anzeigen“. Geben Sie den Namen der Variable ein, Dateityp und falls gewünscht eine Adresse ein.

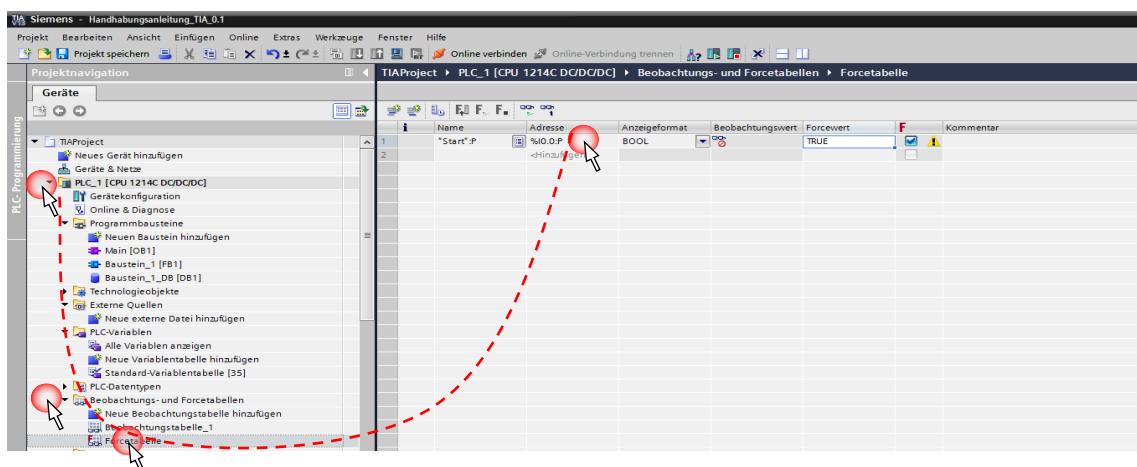


Die Variablen können direkt beim Programmieren definiert werden. Wichtig ist, dass bei den Anschlüssen der Blöcke sprechende Namen (wie Schalter, Start, Ventil, ...) gewählt werden. Vorerst werden diese noch nicht definierten Variablen mit einer roten Linie unterstrichen. Um diese endgültig definieren zu können, klicken Sie mit der rechten Maustaste auf eine leere Fläche im Netzwerk und klicken Sie anschliessend auf „Variable definieren“. Danach öffnet sich das Fenster „Variable definieren“, welches die ausstehenden Definitionen im Netzwerk anzeigt. Zum Schluss können mit der Schaltfläche „Definieren“ alle Variablen bestätigt werden. Auch hier besteht die Möglichkeit von der automatisch generierten Merkvariable abzuweichen.

4.1.3 Variable beobachten / steuern



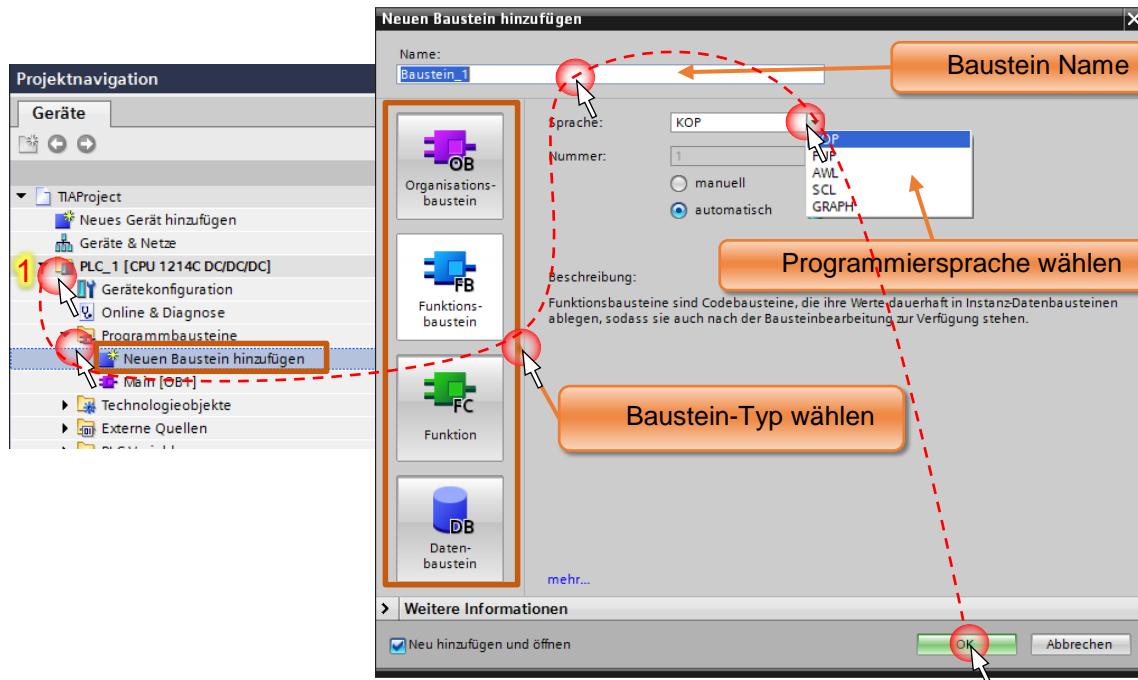
Um die Variablen zu beobachten, gehen Sie im *PLC Register* auf das Register *Beobachtungs- und Forcetabellen*, fügen eine *Neue Beobachtungstabelle hinzufügen*. Fügen Sie über den Namen oder Adresse die Variablen zum Beobachten hinzu. Nach der Eingabe der variablen Drücken Sie auf „Variablen Beobachten“ .



Um eine oder mehrere variablen zu Steuern/Forcen öffnen Sie die Forcetabelle und tragen Sie die Variablen ein. Stellen Sie den gewünschten Forcewert ein und klicken auf Forcen .

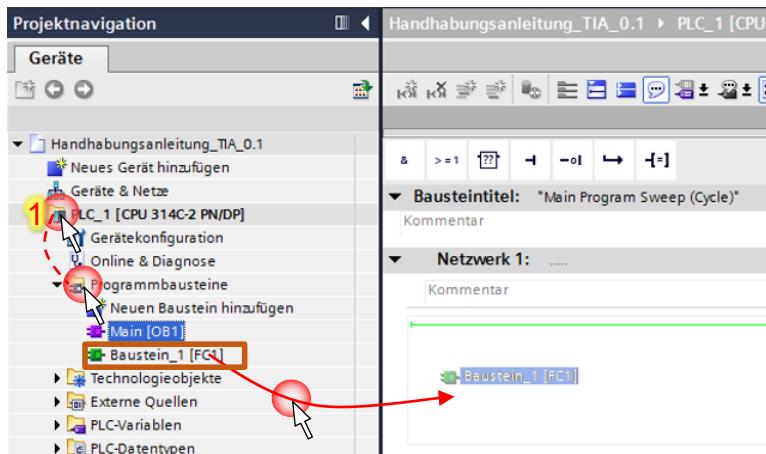
4.2 Bausteine erstellen

Um Bausteine zu erstellen, gehen Sie in der Projektnavigation Links im Fenster, in das Register der PLC (*PLC_1, Programmbausteine*) und doppelklicken Sie auf *Neuen Baustein hinzufügen*.



Geben Sie den gewünschten Namen für den Baustein ein, wählen Sie den Baustein Typ und die gewünschte Programmiersprache aus, danach bestätigen Sie mit OK.

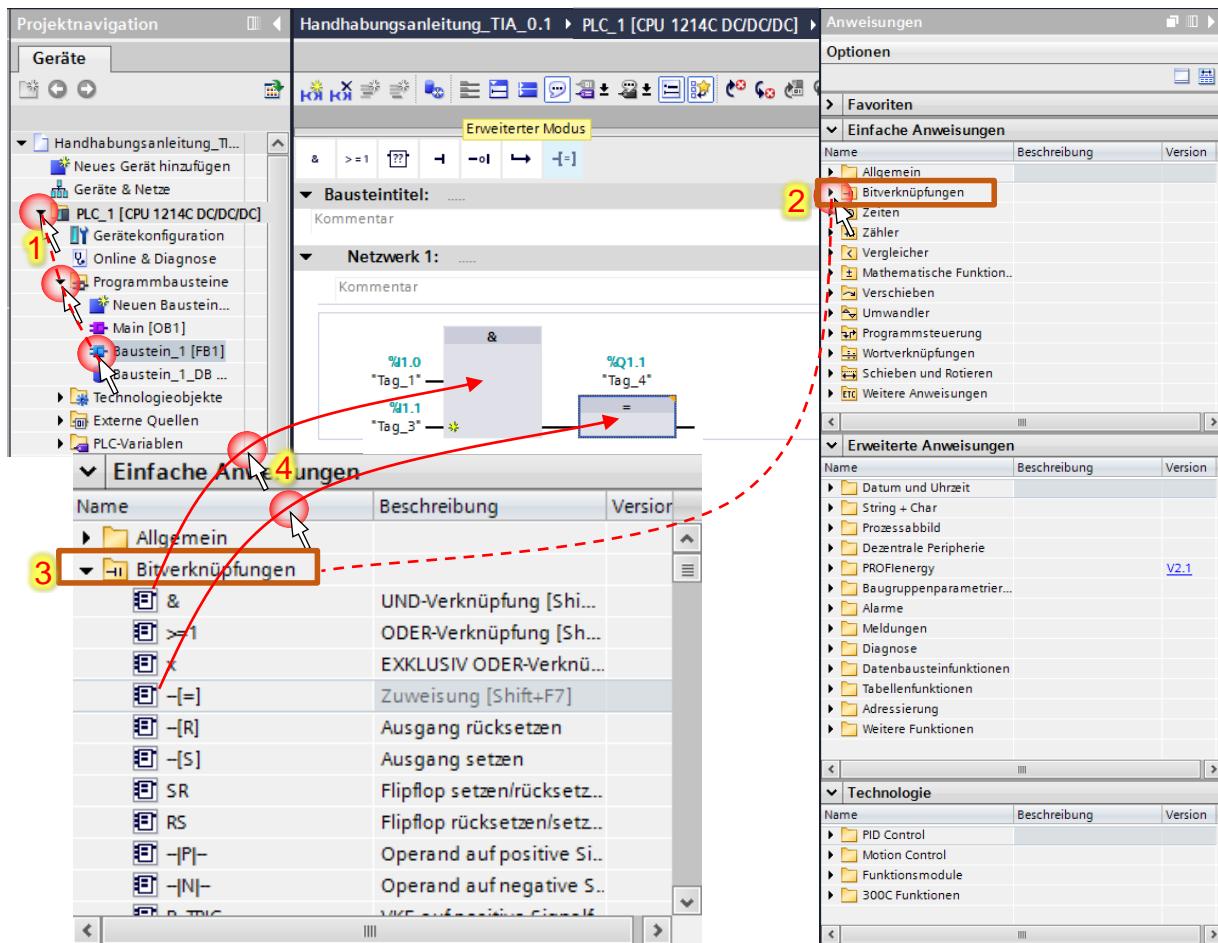
4.3 Baustein in OB1 aufrufen



Öffnen Sie den OB1 und ziehen Sie den gewünschten Baustein, vom Register *Programmbausteine*, in den OB1 hinein.

Speichern Sie das Projekt, , und schliessen Sie den OB1.

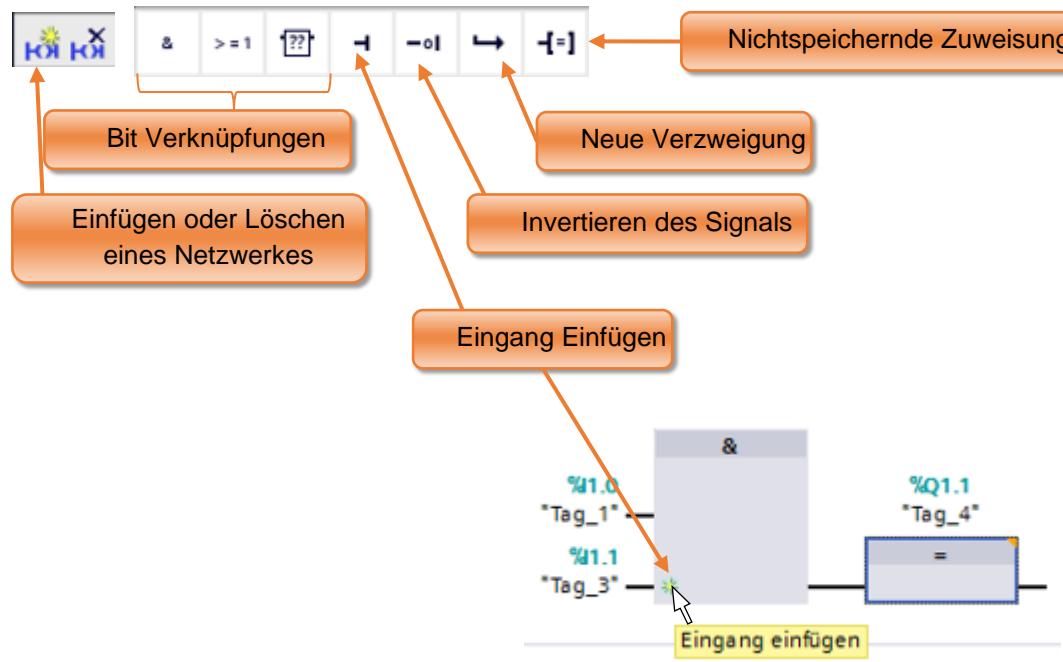
4.4 SPS Schaltprogramm erstellen



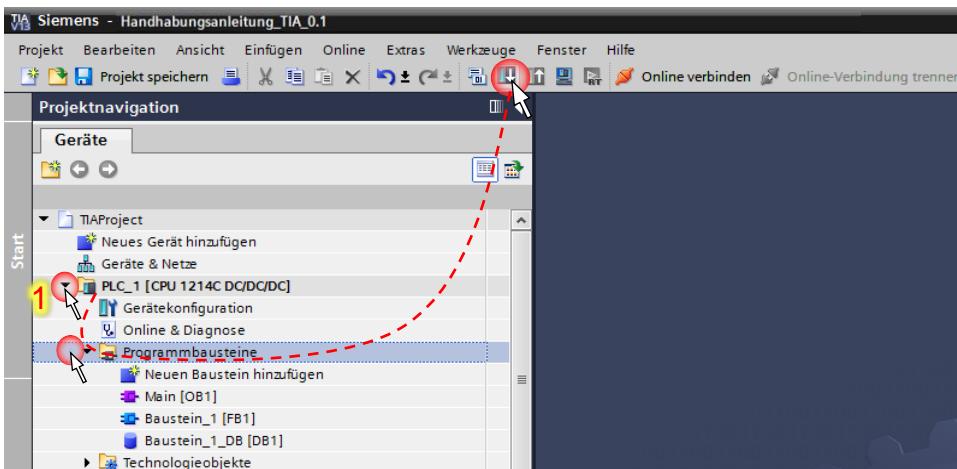
- (1) Um ein Programm zu schreiben öffnen Sie den erstellten Baustein.
- (2) Auf der rechten Seite öffnet sich das Fenster Anweisungen, darunter finden Sie Einfache Anweisungen (Bitverknüpfungen, Zähler, Zeiten,...), Erweiterte Anweisungen (Datenbausteinfunktionen, Datum und Uhrzeit, String + Char,...) und Technologie (PID Control, Motion Control, Funktionsmodule,...)
- (3) Klicken Sie auf das Register *Bitverknüpfungen* und ziehen Sie die „UND-Verknüpfung“ ins *Netzwerk 1*.
- (4) Ziehen Sie dann die „--[=] Zuweisung“ ins *Netzwerk 1*.

Geben Sie die Variablen in die dazugehörigen Felder ein, zum Beispiel I1.0 und I1.1 bei der UND-Verknüpfung und Q1.1 bei der Zuweisung. Speichern Sie danach das Projekt.

Ergänzen Sie das Programm gemäss der Aufstellung. Das nächste Netzwerk wird Automatisch erstellt sobald man im aktuellen Netzwerk programmiert.



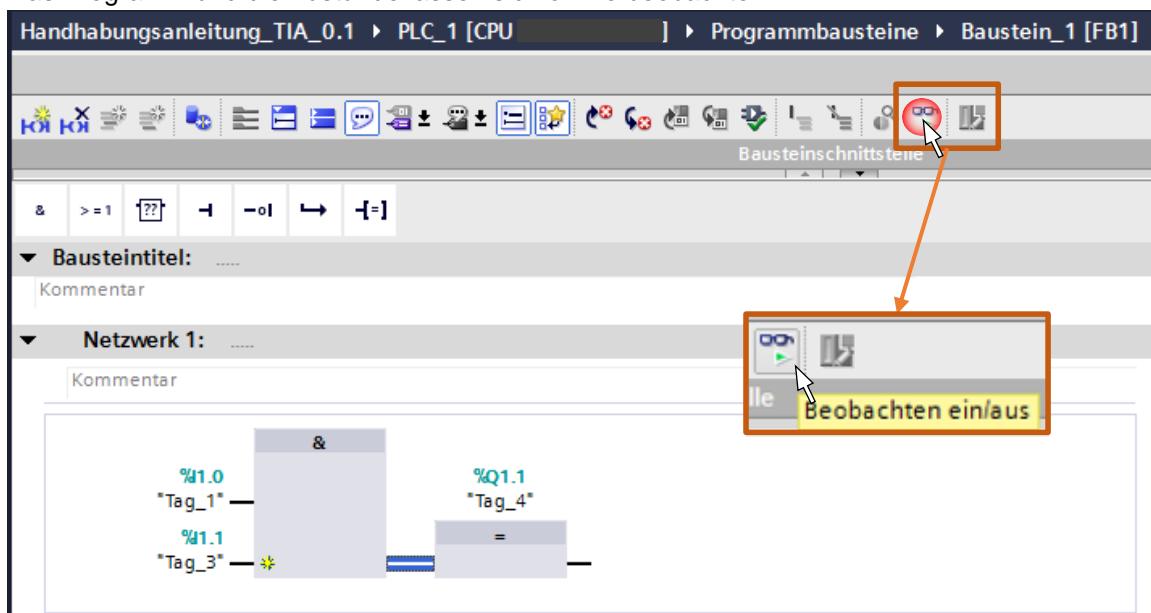
4.5 Programm in Gerät laden



Um das Programm in das Gerät zu laden gehen Sie auf das Register *Programmbausteine* und klicken Sie auf „Laden in Gerät“ .

4.6 Programm beobachten

Das Programm und die Zustände lassen sich online beobachten.



Falls das Programm sich nicht beobachten lässt, kontrollieren Sie ob der Baustein im OB1 aufgerufen wird und ob die Verbindung zur PLC richtig eingestellt ist. Um die Beobachtung zu starten oder zu beenden drücken Sie auf die „Brille - Beobachten ein/aus“.

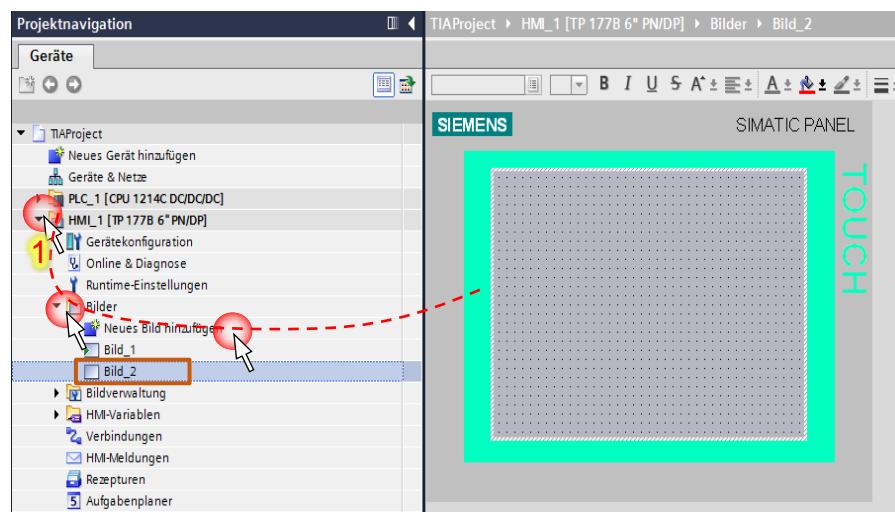
5 HMI erstellen

5.1 HMI Variable erstellen

Die HMI Variabelltabelle wird automatisch ausgefüllt sobald man eine PLC Variable am HMI Programmiert.

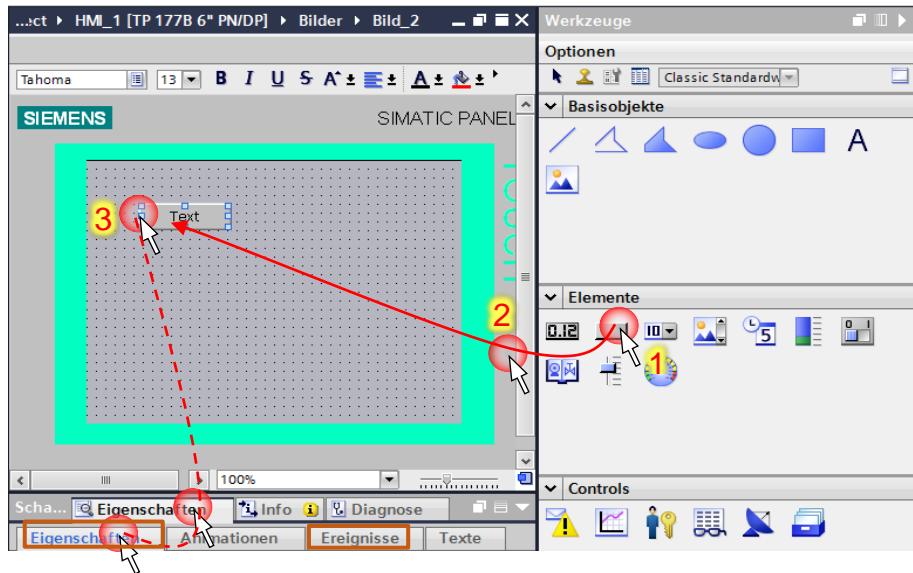
Handhabungsanleitung_TIA_0.1 > HMI [TP 177B 6" PN/DP] > HMI-Variablen										
Name	Variabelltabelle	Datentyp	Verbindung	PLC-Name	PLC-Variable	Adresse	Zugriffsart	Erfassungszyklus	Archiv...	Kommentar
Start	Standard-Variabelltabelle	Bool	HMI_Verbindung_1	PLC_1	Start	%E0.0	<Absoluter Zugriff>	1 s		
<Hinzufügen>										

5.2 Bild erstellen



Um ein neues Bild zu erstellen öffnen Sie das Register *HMI, Bilder* und klicken Sie auf „Neues Bild hinzufügen“, danach öffnet sich das neu erstellte Bild.

5.3 Bedienelement erstellen



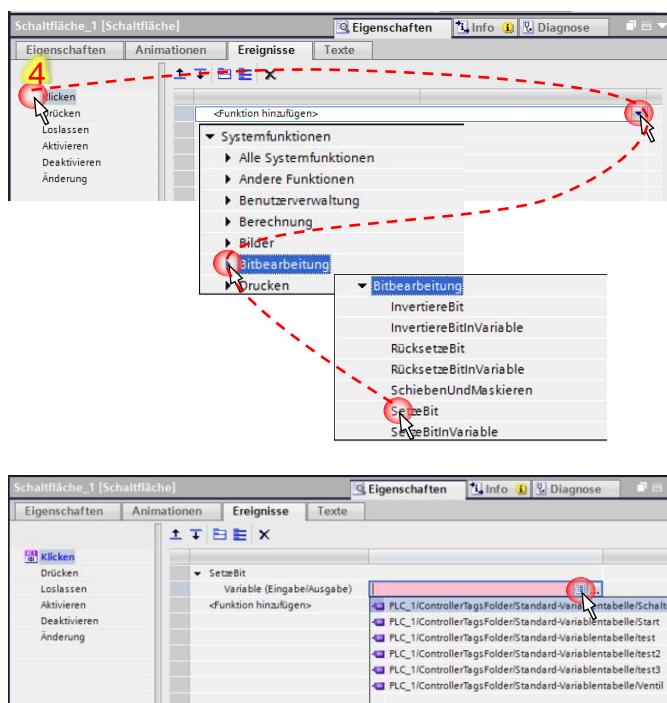
Um ein Bedienelement zu erstellen müssen Sie das gewünschte Bild öffnen.

Auf der rechten Seite erscheint das Fenster „Werkzeuge“, darunter befinden sich die *Basisobjekte*, *Elemente* und *Controls*.

(1) Zum Einfügen einer Schaltfläche gehen Sie unter „Elemente“ und ziehen das Symbol der Schaltfläche an den gewünschten Platz im Bild.

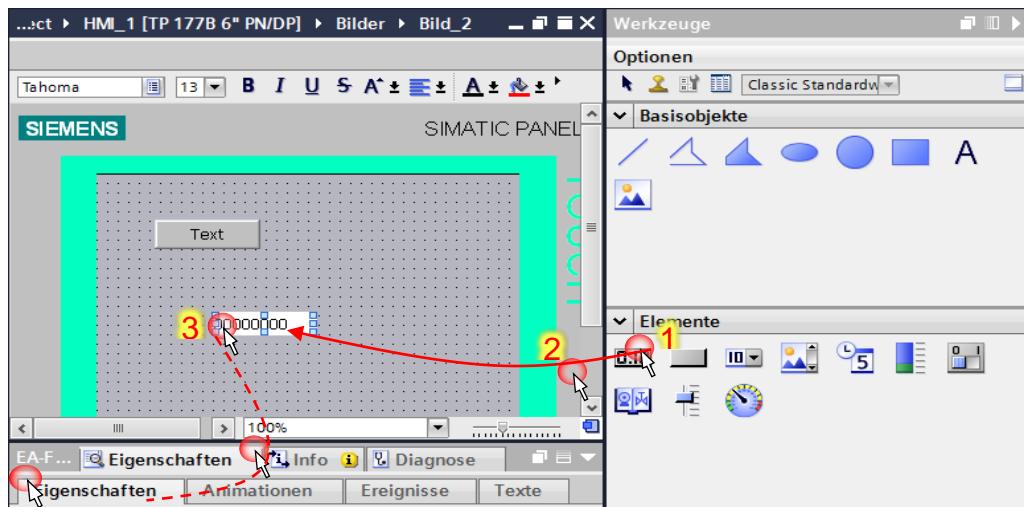
(3) Nach dem Klicken auf das neue Element werden unten die Eigenschaften angezeigt. Unter dem Register *Eigenschaften* ändert man die Beschriftung des Elements

(4) Ausgehend von der Tastenaktivität (Klicken, Drücken, Loslassen,...), wird unter dem Register *Ereignisse* dem Element eine Funktion zugewiesen. Unter *Systemfunktionen*, *Bitbearbeitung* kann beispielsweise das *SetzeBit* ausgewählt werden.



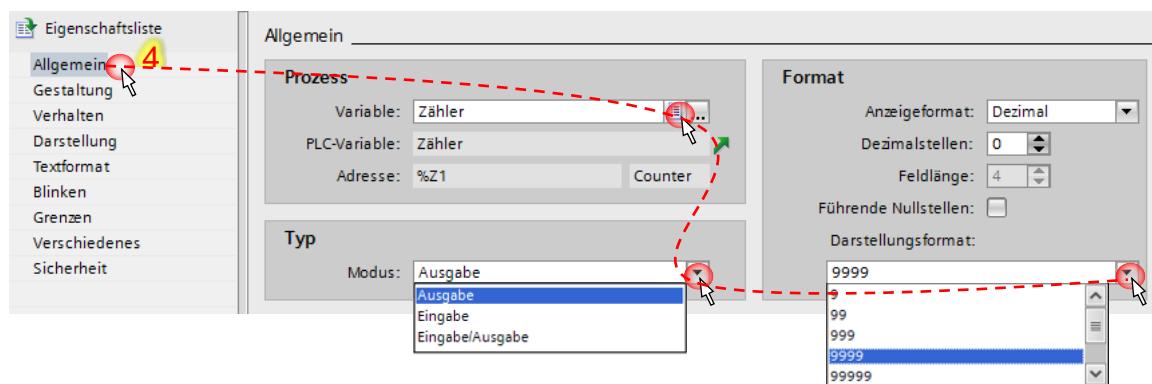
Das zu Setzende Bit kann aus der PLC Variabellabelle  ausgewählt werden.

5.4 Visueller Indikator erstellen



Um einen Visuellen Indikator (Zähler) zu erstellen müssen Sie das gewünschte Bild öffnen. Auf der rechten Seite erscheint das Fenster „Werkzeuge“, darunter befinden sich die *Basisobjekte*, *Elemente* und *Controls*.

- (1) Zum Einfügen eines E/A-Feld gehen Sie unter „Elemente“ und ziehen das Symbol der Schaltfläche an den gewünschten Platz im Bild.
- (3) Nach dem Klicken auf das neue Element werden unten die Eigenschaften angezeigt.
- (4) Unter „Allgemein“ stellt man die Prozess Variable ein die man aus der PLC Variabelltabelle  ausgewählt, zum Beispiel den Zähler Z1. Der Typ des Visuellen Indikators wird je nach Anwendung auf Eingabe/Ausgabe oder nur Eingabe oder Ausgabe eingestellt, im Beispiel wird nur der Zählerstand angezeigt also Ausgabe. Das „Anzeigeformat“ wird Automatisch von der PLC Variabelltabelle eingestellt, nur das „Darstellungsformat“ muss man noch Einstellen.



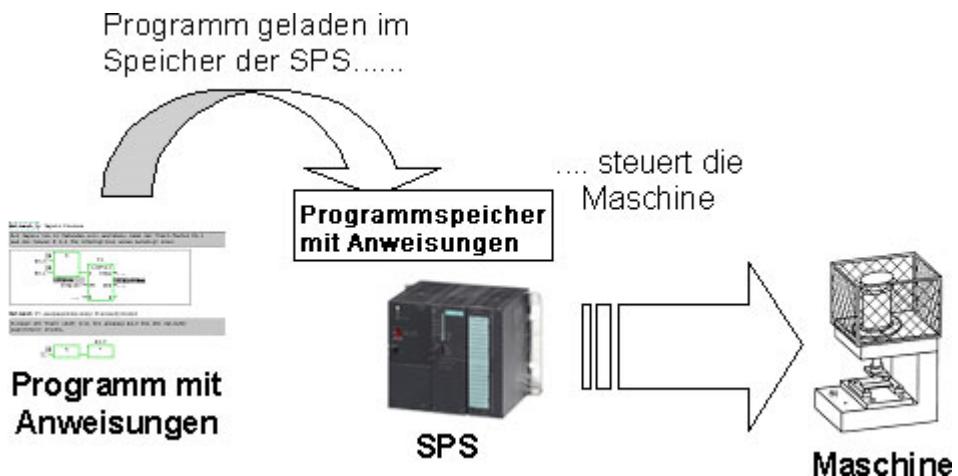
6 Anhang (Grundwissen SPS-Steuerungen)

6.1 Was ist eine SPS und wofür wird sie verwendet?

6.1.1 Was bedeutet der Begriff SPS?

SPS ist die Abkürzung für Speicherprogrammierbare Steuerung. Dies beschreibt ein standardisiertes Automatisierungsgerät, das Arbeitsvorgänge selbstständig ablaufen lässt und beherrscht. Dabei kann es sich um fertigungstechnische oder verfahrenstechnische Prozesse handeln. Beispiele: Eine Druckmaschine zum Drucken von Zeitungen, eine Abfüllanlage zum Abfüllen von Zement oder eine Presse zum Pressen von Kunststoffformteilen.

Die Steuerung geschieht entsprechend den Anweisungen eines Programms, das in einem Programmspeicher der SPS steht.

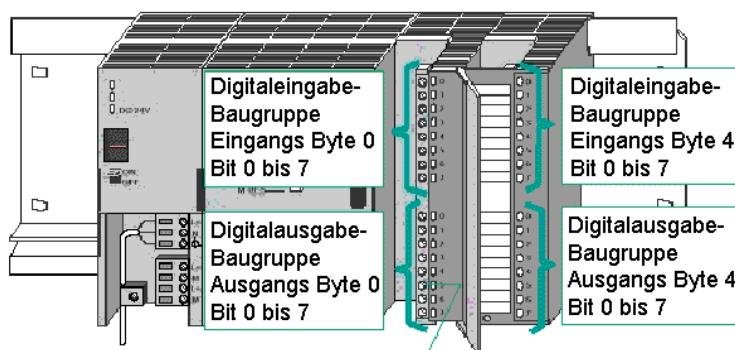


6.1.2 Wie spricht die SPS einzelne Ein-/Ausgangssignale an?

Die Angabe eines bestimmten Ein- oder Ausgangs innerhalb des Programms bezeichnet man als Adressierung.

Die Ein- und Ausgänge der SPS sind zumeist in 8er-Gruppen, also Byteweise, auf Digitaleingabe- bzw. Digitalausgabebaugruppen zusammengefasst. Jede Gruppe erhält eine Nummer als sogenannte Byteadresse. Pro Baugruppe gibt es maximal 4 Byte. Erster Baugruppenplatz Byte 0 bis 3 und zweiter Platz Byte 4 bis 7.

Um nun einen einzelnen Ein- oder Ausgang innerhalb eines Bytes anzusprechen, wird jedem Eingang / Ausgang eine Bitadresse zugeordnet. Die Bitadressen werden von Bit 0 bis Bit 7 durchnummeriert. Daraus ergeben sich für das Beispiel hier mit einer Mischbaugruppe mit 8 Eingängen und 8 Ausgängen die Eingangsadressen E 0.0 bis E 0.7 beziehungsweise für die zweite Baugruppe E4.0 bis E 4.7 und für die Ausgänge entsprechend A0.0 bis A0.7, resp. A4.0 bis 4.7 für die zweite Baugruppe.



Beispiel mit zwei Digital-Baugruppe 6ES7323-1BH01-0AA0 mit je 8 Eingänge und 8 Ausgängen.

Um nun den z. B. fünften Eingang der Byteadresse 0 anzusprechen muss die folgende Adresse angegeben werden:

E 0.4

E kennzeichnet hier den Adressentyp als Eingang, **0** die Byteadresse und **4** die Bitadresse.
Byteadressen und Bitadressen sind immer durch einen Punkt getrennt.

Hinweis: Für die Bitadresse steht hier beim fünften Eingang eine 4, weil bei 0 begonnen wird zu zählen.
Um nun den z. B. untersten Ausgang anzusprechen gibt man die folgende Adresse an:

A 4.7

A kennzeichnet hier den Adressentyp als Ausgang, **4** die Byteadresse und **7** die Bitadresse.
Byteadressen und Bitadressen sind immer durch einen Punkt getrennt.

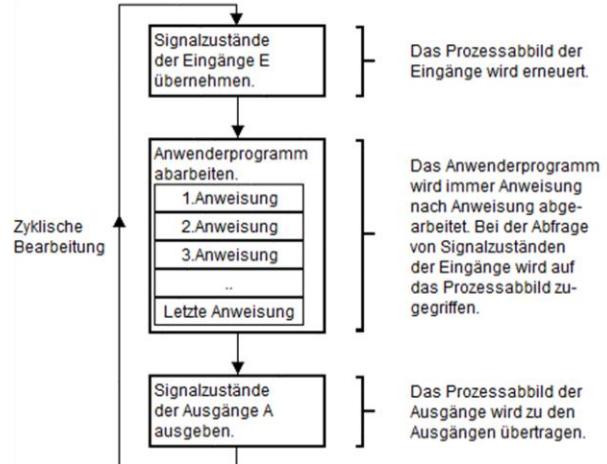
Hinweis: Für die Bitadresse steht hier beim untersten Ausgang eine 7, weil bei 0 begonnen wird zu zählen.

Hinweis: Die Byte-Start-Adressen für die Baugruppen können vom Anwender bei der PLC-Konfiguration frei gewählt werden. Als Default kann nicht bei jeder Steuerung davon ausgegangen werden, dass die Startadressen bei NULL beginnen. Es ist möglich, dass das System der ersten Baugruppe, die eingefügt wird, beispielsweise automatisch die Byte-Adresse 134 zuordnet.

6.1.3 Wie wird in der SPS das Programm bearbeitet?

Die Programmbearbeitung in einer SPS geschieht zyklisch mit folgendem Ablauf:

1. Nachdem die SPS eingeschaltet wurde fragt der Prozessor, der praktisch das Gehirn der SPS darstellt, ab ob die einzelnen Eingänge Spannung führen, also ein Signal anliegt oder nicht. Dieser Status der Eingänge wird in dem Prozessabbild der Eingänge (PAE) gespeichert. Dabei wird für die Spannung führenden Eingänge die Information 1 oder "High", für die keine Spannung führenden die Information 0 oder "Low" hinterlegt.
2. Dieser Prozessor arbeitet dann das im Programmspeicher hinterlegte Programm ab. Dies besteht aus einer Liste von logischen Verknüpfungen und Anweisungen, die nacheinander abgearbeitet werden. Dabei wird für die benötigte Eingangsinformation auf das bereits vorher eingelesene PAE zugegriffen. Die Verknüpfungsergebnisse werden in ein sogenanntes Prozessabbild der Ausgänge (PAA) geschrieben. Auch auf andere Speicherbereiche für Zähler, Zeiten und Merker wird während der Programmbearbeitung gegebenenfalls vom Prozessor zugegriffen.
3. Im dritten Schritt wird nach Abarbeitung des Anwenderprogramms der Status aus dem PAA an die Ausgänge übertragen und diese ein- bzw. ausgeschaltet. Danach beginnt die zyklische Programmbearbeitung wieder an.



Hinweis: Die Zeit die der Prozessor für diesen Ablauf benötigt nennt man Zykluszeit. Diese ist wiederum abhängig von Anzahl und Art der Anweisungen.

6.1.4 Wie sehen logische Verknüpfungen im SPS-Programm aus?

Logische Verknüpfungen werden dazu verwendet um je nach Signalzustand der Eingänge Bedingungen für das Schalten der Ausgänge festzulegen.

Im SPS- Programm können diese in den Programmiersprachen Kontaktplan (KOP), Funktionsplan (FUP) oder Anweisungsliste (AWL) erstellt werden. Der Anschaulichkeit halber wollen wir uns hier auf FUP beschränken. Es gibt eine Vielzahl verschiedener logischer Verknüpfungen die in SPS- Programmen angewendet werden können. UND- sowie ODER- Verknüpfung und die NEGATION eines Eingangs sind Standardverknüpfungen und sollen deshalb hier kurz anhand von Beispielen erläutert werden.

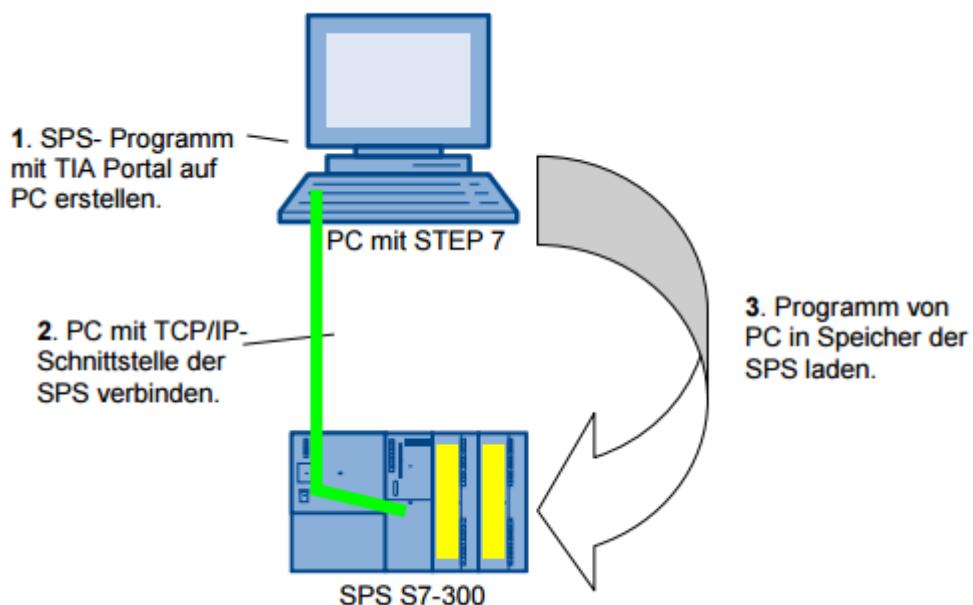
Hinweis: Informationen zu weiteren logischen Verknüpfungen finden Sie in der TIA Portal Online-Hilfe bzw. aus dem SCE-Modul [Grundlegende Programmierbefehle](#) im Bereich [Anhang](#).

6.1.5 Wie wird das SPS-Programm erstellt?

Das SPS- Programm wird mit der Software TIA Portal auf einem PC erstellt und dort zwischengespeichert.

6.1.6 Wie kommt es in den Speicher der SPS?

Nachdem der PC mit der PROFIBUS-, MPI- oder Ethernet- Schnittstelle der SPS verbunden wurde, kann das Programm in den Speicher der SPS geladen werden.



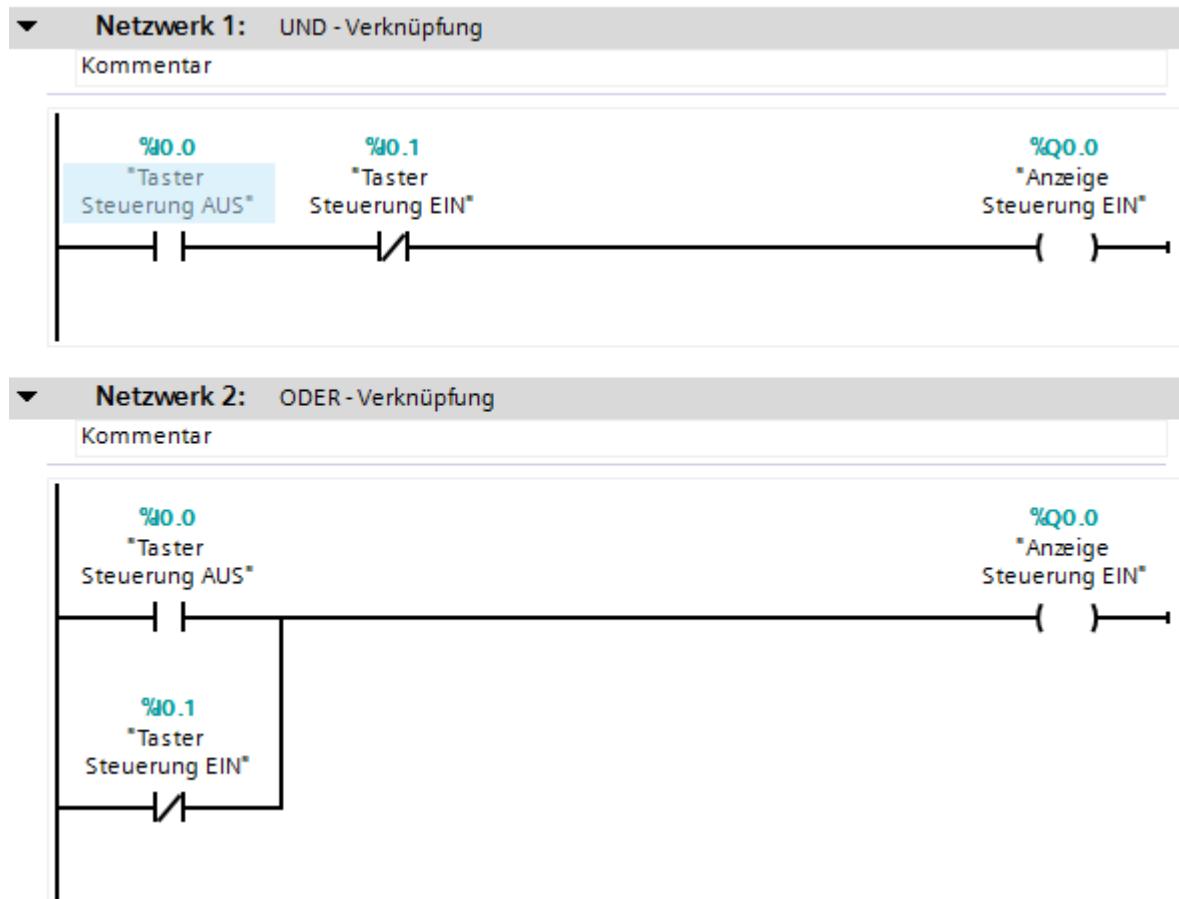
6.2 Programmiersprachen

In STEP 7 lässt sich ein Programm in verschiedenen Arten darstellen und programmieren:

- Kontaktplan KOP
- Funktionsplan FUP
- Anweisungsliste AWL
- Schrittkettenprogrammierung S7-Graph (IEC1131-3: AS Ablaufsprache)
- Hochsprachenprogrammierung S7-SCL (IEC1131-3: ST Strukturierter Text)

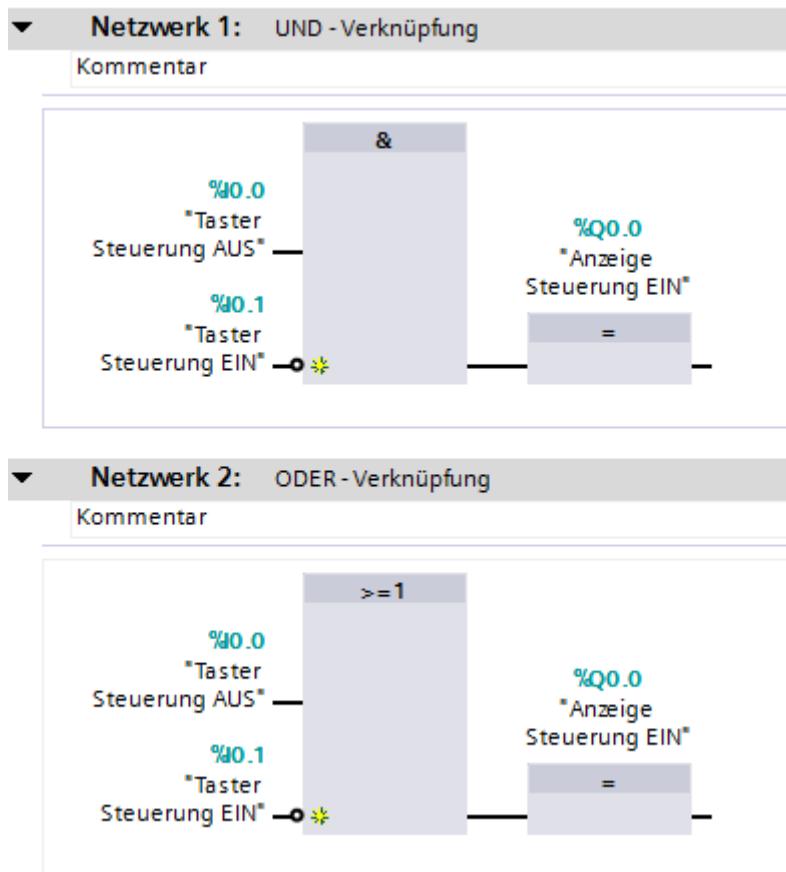
6.2.1 Kontaktplan KOP

Der Kontaktplan ist die bildliche Darstellung der Steuerungsaufgabe mit Symbolen nach DIN 19 239, die auch in den USA üblich sind. Er hat viel Ähnlichkeit mit dem herkömmlichen Stromlaufplan, jedoch sind mit Rücksicht für die Darstellung auf einem Bildschirm die einzelnen Strompfade nicht senkrecht, sondern waagrecht angeordnet.



6.2.2 Funktionsplan FUP

Der Funktionsplan ist die bildliche Darstellung der Steuerungsaufgabe mit Symbolen nach DIN 40 700 und DIN 19 239. Die einzelnen Funktionen werden durch ein Symbol mit Funktionskennzeichen dargestellt. Auf der linken Seite des Symbols werden die Eingänge, auf der rechten Seite die Ausgänge angeordnet.



6.2.3 Anweisungsliste AWL

In der Anweisungsliste wird die Steuerungsaufgabe mit einzelnen Steuerungsanweisungen beschrieben. Die Steuerungsanweisung (Operation und Operand) stellt die Aufgabe mit mnemotechnischen (sinnfälligen) Abkürzungen der Funktionsbezeichnung dar. (Nach DIN 19 239)

Netzwerk 1: UND - Verknüpfung			
1	A	"Tag_1"	%I2.0
2	AN	"Tag_2"	%I2.1
3	=	"Tag_3"	%Q2.0
Netzwerk 2: ODER - Verknüpfung			
1	O	"Tag_1"	%I2.0
2	O	"Tag_2"	%I2.1
3	=	"Tag_3"	%Q2.0

Jede Darstellungsart beinhaltet spezielle Eigenschaften und bestimmte Grenzen. Wenn bei der Programmierung bestimmte Regeln eingehalten werden, ist ein Übersetzen in alle drei Darstellungsarten problemlos möglich. Steuerungsprogramme die in KOP oder in FUP programmiert wurden, können grundsätzlich immer in Anweisungsliste AWL übersetzt werden.



Hinweis: AWL wird von der Steuerung S7-1200 nicht unterstützt.

6.2.4 Strukturierter Text mit S7-SCL

S7-SCL (Structured Control Language) ist eine höhere Programmiersprache, die sich an PASCAL orientiert und eine strukturierte Programmierung ermöglicht. Die Sprache entspricht der in der Norm DIN EN-61131-3 (IEC 61131-3). S7-SCL enthält neben Hochsprachenelementen auch typische Elemente der SPS wie Eingänge, Ausgänge, Zeiten, Merker, Bausteinaufrufe usw. als Sprachelemente. Sie unterstützt das Bausteinkonzept von STEP 7 und ermöglicht daher neben AWL, KOP und FUP die normkonforme Programmierung von Bausteinen. D.h. S7-SCL ergänzt und erweitert die Programmiersoftware STEP 7 mit ihren Programmiersprachen KOP, FUP und AWL.

Bausteine, die mit S7-SCL programmiert sind, können Sie mit AWL-, KOP- und FUP- Bausteinen mischen. Das bedeutet, dass ein mit S7-SCL programmiert Baustein einen anderen Baustein, der in AWL, KOP oder FUP programmiert ist, aufrufen kann. Entsprechend können S7-SCL Bausteine auch in AWL-, KOP- und FUP-Programmen aufgerufen werden.

Die Testfunktionen von S7-SCL ermöglichen die Suche nach logischen Programmierfehlern in einer fehlerfreien Übersetzung. Die Fehlersuche erfolgt dabei in der Quellsprache.

```
1
2 IF #i_Start AND NOT #s_FP_Start AND NOT (#i_PNR = 0) THEN
3
4     #o_X_Achse:= 0;
5     #o_Y_Achse:= 0;
6
7 FOR #t_X_Index := 1 TO 7 DO
8     FOR #t_Y_Index := 1 TO 12 DO
9         IF #Fach[#t_X_Index,#t_Y_Index] = 0 THEN
10            #Fach[#t_X_Index,#t_Y_Index]:= #i_PNR;
11            #OK:= true;
12            #o_X_Achse:= #t_X_Index;
13            #o_Y_Achse:= #t_Y_Index;
14            EXIT;
15        END_IF;
16    END_FOR;
17    IF #OK THEN
18        EXIT;
19    END_IF;
20 END_FOR;
21 END_IF;
22
23 #s_FP_Start:= #i_Start;
```

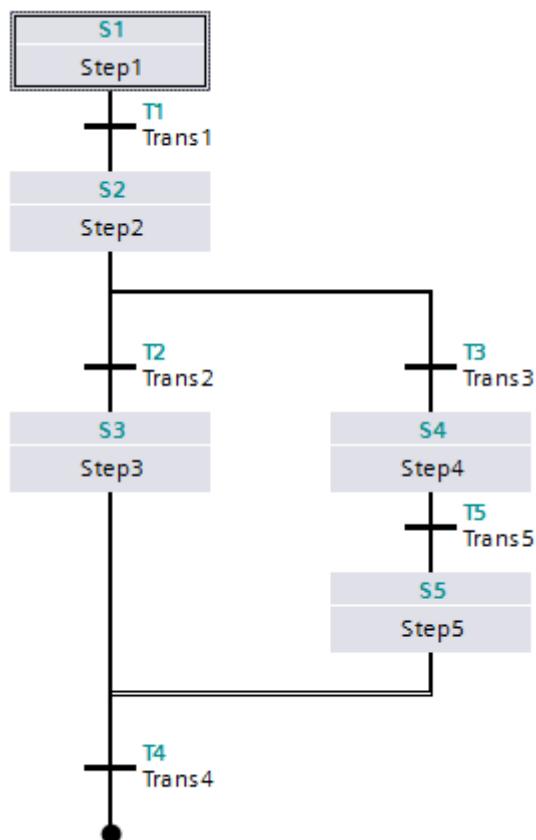
6.2.5 Ablaufsteuerung mit S7-Graph

Mit der Programmiersprache S7-GRAF wird der Funktionsumfang von STEP 7 um eine grafische Programmiermöglichkeit für Ablaufsteuerungen erweitert.

Mit S7-GRAF können Sie Ablaufsteuerungen übersichtlich und schnell programmieren. Der Prozess wird dabei in Einzelschritte zerlegt und der Ablauf graphisch dargestellt.

In den Einzelschritten werden die auszuführenden Aktionen festgelegt.

Die Weiterschaltbedingungen zu den jeweiligen nächsten Schritt (Transitionen) können in der Programmiersprache KOP oder FUP erstellt werden.



6.3 Merker

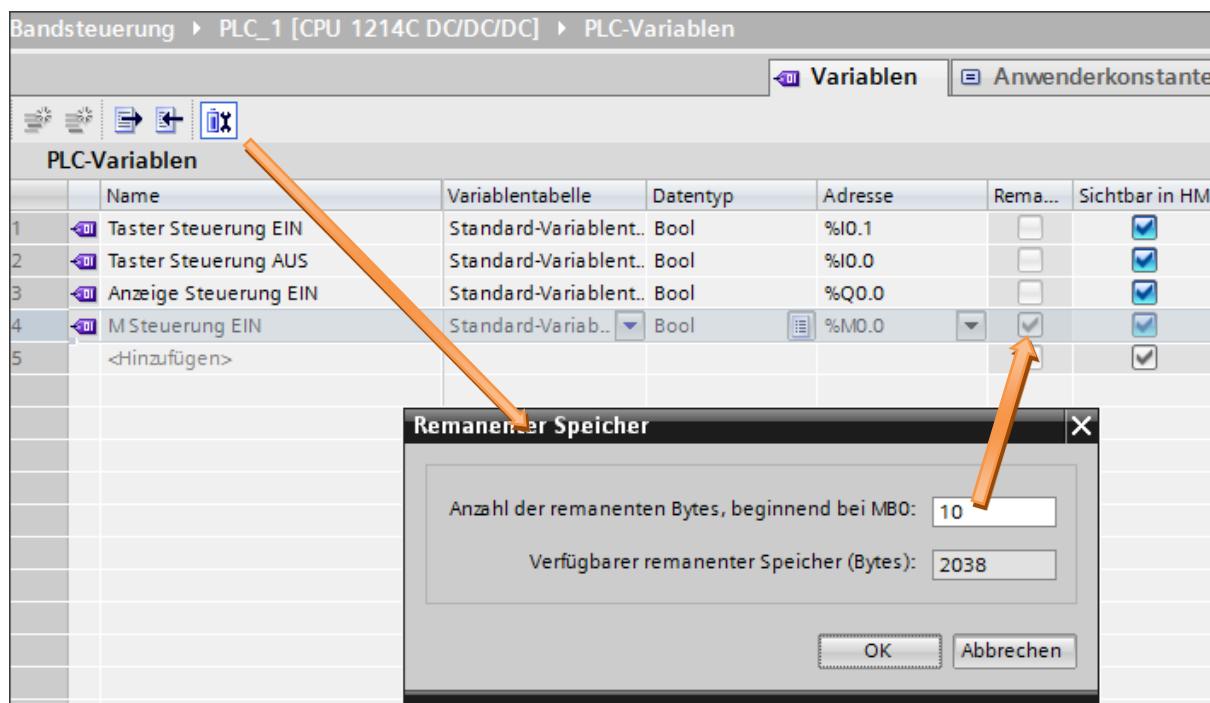
Ein Merker ist ein Speicherbereich im Systemspeicher einer CPU. Auf ihn kann schreibend und lesend zugegriffen werden (Bit-, byte-, wort- und doppelwortweise). Bei der klassischen SPS-Programmierung muss der Anwender den Speicherbereich eigenhändig auswählen. Bei der Programmierung im TIA-Portal, wählt der Anwender ein Merkername und das TIA-Portal ordnet automatisch diesen Merker einem Speicherbereich zu.

6.3.1 Remanente Speicherbereiche

Um Datenverluste bei Spannungsauftakt zu vermeiden, können Sie bestimmte Daten als remanent kennzeichnen. Diese werden in einem remanenten Speicherbereich abgelegt. Ein remanenter Speicherbereich ist ein Bereich, dessen Inhalt bei einem Neustart (Warmstart), d. h. nach Ausschalten der Versorgungsspannung und nach Einschalten bei einem Übergang von STOP nach RUN erhalten bleibt.

Als remanent können Sie die folgenden Daten definieren:

- Merker: Die genaue Breite des remanenten Speicherbereichs können Sie für Merker in der PLC-Variabelliste oder im Belegungsplan festlegen.
- Variablen eines Funktionsbausteins (FB): In der Schnittstelle eines FBs können Sie einzelne Variablen als remanent definieren, wenn für diesen Baustein der optimierte Bausteinzugriff aktiviert ist. Wenn der optimierte Bausteinzugriff für einen FB nicht aktiviert ist, können Remanenzeinstellungen nur im zugeordneten Instanz-Datenbaustein vorgenommen werden.
- Variablen eines globalen Datenbausteins: In einem globalen Datenbaustein können Sie abhängig von der Einstellung für den Zugriff entweder einzelne oder alle Variablen des Bausteins als remanent definieren:
 - Baustein mit optimierter Zugriff: Remanenz ist für jede einzelne Variable einstellbar.
 - Baustein mit Standardzugriff: Remanenzeinstellung gilt für alle Variablen des DBs; entweder sind alle Variablen remanent oder keine Variable ist remanent.



6.4 Anwenderbausteine für die SIMATIC S7-1200

Für die strukturierte Programmierung stehen folgende Anwenderbausteine zur Verfügung:

- OB (Organisationsbaustein):



Ein OB wird vom Betriebssystem zyklisch aufgerufen und bildet somit die Schnittstelle zwischen Anwenderprogramm und Betriebssystem. In diesem OB wird dem Steuerwerk über Bausteinaufrufbefehle mitgeteilt, welche Programmabsteine es bearbeiten soll.

- FB (Funktionsbaustein):



Ein FB benötigt zu jedem Aufruf(Instanz) einen zugeordneten Speicherbereich. Wenn ein FB aufgerufen wird kann ihm z.B. ein Datenbaustein (DB) als Instanz-DB zugeordnet werden. Auf die Daten in diesem Instanz-DB wird dann über die Variablen des FB zugegriffen. Einem FB müssen verschiedene Speicherbereiche zugeordnet werden wenn er mehrmals aufgerufen wird. In einem Funktionsbaustein können auch wieder weitere FBs und FCs aufgerufen werden.

- FC (Funktion):



Eine FC besitzt keinen zugeordneten Speicherbereich. Die lokalen Daten einer Funktion sind nach der Bearbeitung der Funktion verloren. In einer Funktion können auch wieder weitere FBs und FCs aufgerufen werden.

- DB (Datenbaustein):



DBs werden verwendet um Speicherplatz für Datenvariablen bereitzustellen. Es gibt zwei Arten von Datenbausteinen. Globale DBs, wo alle OBs, FBs und FCs die gespeicherten Daten lesen oder schreiben können und Instanz-DBs, die einem bestimmten FB zugeordnet sind.

6.5 Beispielaufgabe mit Einschaltverzögerung

In Anlehnung an eine überschaubare Schaltfunktion, wird nachfolgend aufgezeigt, wie die gleiche Aufgabe verschiedenartig mittels SPS-Programmierung umgesetzt werden kann. Es werden implizit auch grundlegende Aspekte thematisiert, welche das allgemeine Programmierverständnis im Wandel von der klassischen SPS-Programmierung hin zur modernen objektorientierten Programmier-Art veranschaulichen.

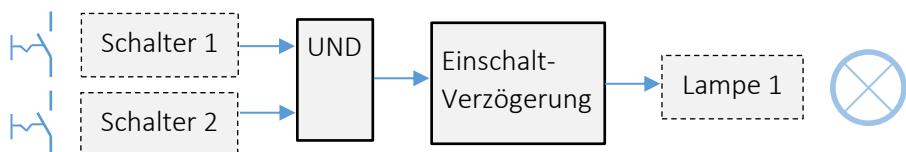
6.5.1 Referenzaufgabe

Als Ausgangslage für die weiteren Erläuterungen wird eine Schaltfunktion thematisiert, welche eine zeitliche Abhängigkeit aufweist.

Die Aufgabe lautet in Worte wie folgt:

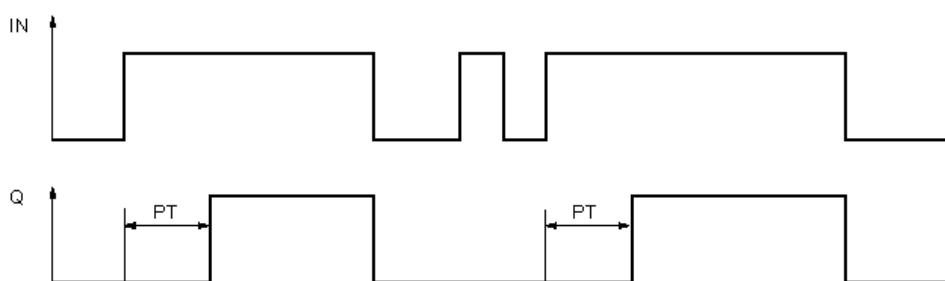
Sobald der Schalter 1 und Schalter 2 dauerhaft betätigt werden, soll die Lampe 1 nach 10s einschalten. Werden danach einer der beiden Schalter oder beide zusammen entriegelt, schaltet die Lampe 1 ohne zeitliche Verzögerung aus.

Die Aufgabe lässt sich als Schaltfunktion wie folgt abbilden:



6.5.2 Zeitglied Einschaltverzögerung

Ein weitverbreitetes zeitliches Verhalten in der Steuerungstechnik ist die Einschaltverzögerung. Wie der Name schon vermuten lässt, wird ein eingehendes Signal am Block-Eingang (IN) erst nach einer definierten Zeit am Block-Ausgang (Q) freigeschaltet. Achtung, es handelt sich hierbei nicht um eine zeitliche Verschiebung des Signals. Das folgende Impulsdiagramm zeigt das Schaltverhalten auf.



6.5.3 Evolution der SPS-Programmierung

Das TIA-Portal ermöglicht eine zeitgemäße SPS-Programmierung und rückt strukturell betrachtet im Vergleich zur klassischen SPS-Programmierung (eher adressorientiert) näher zur variablenorientierten Programmierung. Neulanierte Siemens-Produkte (z.B. S7-1200 und S7-1500) setzen zwingend die Programmierung mit dem TIA-Portal voraus und es ist nur in Einzelfällen möglich auf klassische Entwicklungsumgebungen auszuweichen. Diese werden im Siemens-Umfeld als «Classic» bezeichnet (z.B. Step7 Classic für die Steuerungen S7-300 und S7-400).

S7-300er Steuerungen können ab einem gewissen Firmware-Stand als Übergangsmodell zwischen der klassischen und zeitgemäßen SPS-Programmierung betrachtet werden. Wird eine CPU-300 mit dem TIA-Portal programmiert, so findet der Anwender in den Bibliotheken sowohl neuartige wie auch klassische Programmierelemente. Bezogen auf vorhandene Zeitglieder ist aus der untenstehenden Auflistung (Auszug aus TIA-Portal) zu erkennen, dass bei dieser CPU viele SIMATIC-Timer (klassisch adressorientiert) zur Auswahl stehen und drei IEC-Timer (neuartig objektorientiert) vorhanden sind.



IEC-Timer:
Objektorientierter Timer-Typ, welcher in aktuellen CPU-Generationen verwendet wird und eine zeitgemäße Programmierung ermöglicht.

SIMATIC-Timer:
Adressorientierter Timer-Typ aus der klassischen SPS-Programmierung. Ist im Umgang mit älteren CPU-Generationen vorzufinden.

Bitverknüpfungen	
Zeiten	
IEC-Zeiten	
■ TP	Impuls erzeugen
■ TON	Einschaltverzögerung erzeugen
■ TOF	Ausschaltverzögerung erzeugen
SIMATIC-Zeiten	
■ S_PULSE	Zeit als Impuls parametrieren und starten
■ S_PEXT	Zeit als verlängerten Impuls parametrieren und starten
■ S_ODT	Zeit als Einschaltverzögerung parametrieren und starten
■ S_ODTS	Zeit als speichernde Einschaltverzögerung parametrieren und starten
■ S_OFFDT	Zeit als Ausschaltverzögerung parametrieren und starten
■ -[SP]	Zeit als Impuls starten
■ -[SE]	Zeit als verlängerten Impuls starten
■ -[SD]	Zeit als Einschaltverzögerung starten
■ -[SS]	Zeit als speichernde Einschaltverzögerung starten
■ -[SF]	Zeit als Ausschaltverzögerung starten
+1 Zähler	

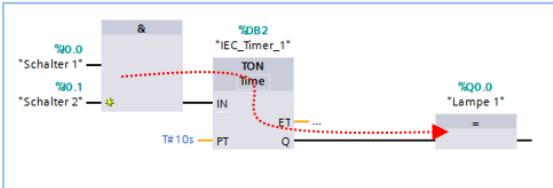
6.5.4 Umsetzung der Referenzaufgabe

Die nachfolgende Auflistung zeigt auf, wie die Referenzaufgabe auf vier verschiedene Arten implementiert werden kann.

Die Lösungsvarianten eins, zwei und drei wurden mit IEC-Timer realisiert. Hierbei wird unter anderem gezeigt, wie unterschiedlich sich der Datenaustausch zwischen Programm-Netzwerken gestalten lässt. Der Leitgedanke liegt darin, den Datenaustausch mittels Variablen (Merker) zu bewerkstelligen.

Die Lösungsvariante vier wurde mit einem SIMATIC-Timer umgesetzt und zeigt auf, wie der Datenaustausch in diesem Fall (mittels einem Zeitmerker) zu realisieren ist.

Netzwerk 1

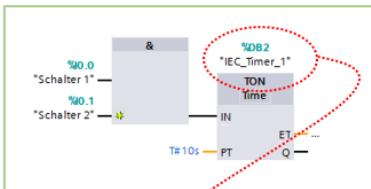


Bei dieser Lösungsvariante sind alle Blöcke mittels Signallinien innerhalb eines Netzwerkes verbunden. Diese Lösung ist sehr stark mit der in der

Referenzaufgabe abgebildeten Schaltfunktion verwandt. Das binäre Ausgangsverhalten des IEC-Timer-Blocks wird direkt über den eigenen Ausgang Q an die Lampe 1 weitergeleitet.

Lösungsvariante 1

Netzwerk 1

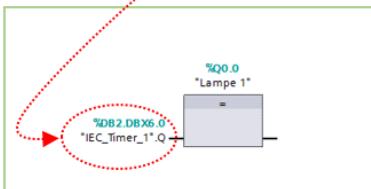


Die gleiche Funktion kann als Lösung auf zwei Netzwerke unterteilt werden. Der Datenaustausch zwischen Netzwerken findet generell über Variablen statt.

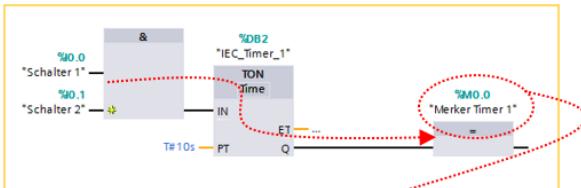
Das binäre Ausgangsverhalten des IEC-Timers wird im zweiten Netzwerk mittels der strukturierten Schreibweise „IEC_Timer_1.Q“ der Lampe 1 übermittelt.

Lösungsvariante 2

Netzwerk 2



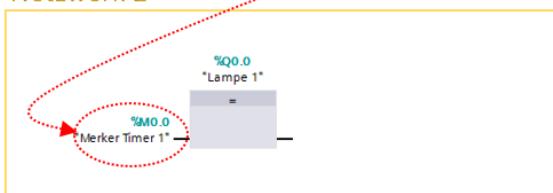
Netzwerk 1



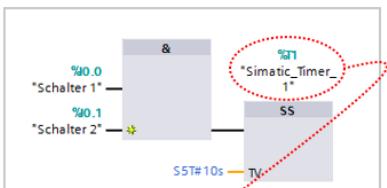
Diese Lösung ist auch auf zwei Netzwerke unterteilt. Das binäre Ausgangsverhalten des IEC-Timers wird in diesem Fall zuerst in eine binäre Merker-Variable „M0.0“ abgelegt. Im zweiten Netzwerk wird der Zustand mittels derselben Hilfsvariable der Lampe 1 übermittelt.

Lösungsvariante 3

Netzwerk 2



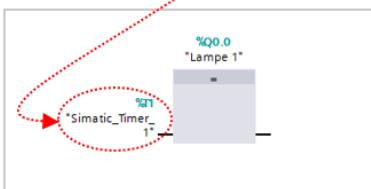
Netzwerk 1



Diese Variante unterscheidet sich in der Wahl des Timer-Typs. Das gleiche Zeitverhalten kann anstatt mit einem IEC-Timer auch mit einem SIMATIC-Timer erfolgen.

Lösungsvariante 4

Netzwerk 2



Das binäre Ausgangsverhalten des SIMATIC-Timers ist in der Zeit-Merker-Variable „T1“ enthalten und wird im zweiten Netzwerk mittels derselben der Lampe 1 übermittelt.