

Schnelleinstieg BASICS CODESYS V3.5 für den CM211

CM211_Basics.project* - CODESYS

Geräte

- CM211_Basics
 - Device (elrest CM211)
 - SPS-Logik
 - Application
 - Bibliotheksverwalter
 - PLC_PRG (PRG)
 - Taskkonfiguration
 - MainTask
 - PLC_PRG

Kommunikationseinstellungen

Gateway-1 192.168.5.228 (aktiv)

IP-Address: localhost Gerätename: CM211_v35130

Port: 1217 Geräteadresse: 0000.5460.9000.2DDC.C0A8.05E4

Geräte-IP-Adresse: 192.168.5.228

Zielsystem-ID: 1018 2000

Zielsystemtyp: 4096

Zielsystemhersteller: elrest Automationssysteme GmbH

Zielsystemversion: 3.5.1.30

Meldungen - Gesamt 0 Fehler, 0 Warnung(en), 4 Meldung(en)

Übersetzen 0 Fehler 0 Warnung(en) 4 Meldung(en)

Beschreibung	Projekt	Objekt	Position
-----Übersetzungslauf gestartet: Applikation: Device,Application-----			
Die Applikation ist aktuell			
Größe des erzeugten Codes: 11384 Bytes	CM211_Basics		
Größe der globalen Daten: 3774 Bytes	CM211_Basics		
Gesamter allozierter Speicherumfang für Code und Daten: 17208 bytes	CM211_Basics		
Speicherbereich 0 enthält: Daten, Eingang, Ausgang, Speicher und Code: Größe: 1048576, höchste verwendete Adresse: 1048576, größte zusammenhängende Spei...	CM211_Basics		
Kompilierung abgeschlossen -- 0 Fehler, 0 Warnungen			

Precompile: OK

elrest Automationssysteme GmbH
Leibnizstraße 10
73230 Kirchheim unter Teck
Germany
Telefon: +49 (0) 7021 / 92025-0
www.elrest.de

Inhaltsverzeichnis

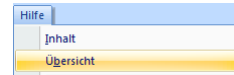
Inhaltsverzeichnis	2
1 Willkommen bei CODESYS V3.....	3
1.1 Installation	4
1.1.1 CODESYS V3 Installation	4
1.2 CODESYS V3	6
1.2.1 CODESYS - Marktstandard für Steuerungsprogrammierung nach IEC 61131-3.....	6
2 Beispielprojekte	7
2.1 Das erste Programm <code>a:=a+1;</code>	11
2.1.1 Die Variable a direkt auf die Digitalausgänge ausgeben.....	12
2.1.2 Ein-und Ausgabe Teil der Programmierung.....	13
2.1.3 Digitale Ein- und Ausgänge	13
2.1.4 Einloggen und Starten.....	16
2.1.5 Schreiben von Werte	16
2.1.6 Abschluss der Programmierung	32
3 Support.....	33
4 Historie	33

1 Willkommen bei CODESYS V3

Wir beglückwünschen Sie zu Ihrer Entscheidung für CODESYS V3.

Mit CODESYS V3 haben Sie ein professionelles Programmierwerkzeug zur graphischen Programmierung von MMI- (Mensch Maschine Interface), MSR- (Messen, Steuern, Regeln) und MOTION- (CNC, Softmotion) Anwendung erworben. Wir sind überzeugt, dass CODESYS V3 Sie bei Ihrer Arbeit in Zukunft entscheidend unterstützen wird und Ihnen eine rationelle und kostensensitive Umsetzung Ihrer Aufgaben auf höchstem Niveau ermöglicht. Unsere Produkte können mit einer Vielzahl von Gerätekonfigurationen arbeiten, darüber hinaus erweitern und verbessern wir dahingehend ständig die Produktpalette.

Für eine Übersicht der Soft- und Hardware Produkte sowie Hinweise und Unterstützung öffnen Sie in der Projektverwaltung „Hilfe→Übersicht“.



Folgende Schnittstellen werden von elrest unterstützt:



Interface zu allen Peripheriegeräten wie Stick, Tastatur, Maus, Drucker, u.v.m.



Serielle Schnittstellen RS232 und/oder RS485. Diese können mit beliebigen UART Protokollen oder dem vorbereiteten Protokoll RTU-Modbus betrieben werden.



Offenes Feldbusinterface zu beliebigen CAN Protokollen, wie beispielsweise Truck-Norm J1939-based.



CANopen ist ein verbreitetes Layer7 Protokoll für die Automatisierung.



EtherCAT ist ein verbreitetes Realtime Ethernet Protokoll für die Automatisierung.



Unter Ethernet kann TCP-Modbus als UDP oder TCP verwendet werden. Weiter können auf Basis von socket Funktionen weitere Protokolle ergänzt werden.



Folgende Icon's werden verwendet:



Klick auf den linken Maus-Knopf



Klick auf den rechten Maus-Knopf



Doppelklick auf den linken Maus-Knopf



Das CODESYS Framework ist das zentrale Startprogramm.

1.1 Installation

1.1.1 CODESYS V3 Installation

Der einfachste Weg, um CODESYS zu installieren, ist der Download von unserer [elrest Homepage](#).

Über Service → Kunden Login
kommen Sie auf den öffentlichen Downloadbereich.
Das Passwort erhalten Sie erstmalig und
unmittelbar nach einer Anfrage an

support@elrest.de

Service
Startseite >> Service >> Kunden Login

Kunden Login

- Kunden Login
- Einkauf
- Rücklieferungen
- Info / Konditionen
- Datenschutz

Downloadbereich
Bitte geben Sie Ihren Benutzernamen und Passwort ein um sich an unserem Downloadbereich anzumelden!

Benutzername:

Passwort:

Dateiname	Geändert
Bilder	12-10-11 08:18
Company	17-05-10 15:49
FAQ	02-09-11 11:40
Hardware	19-07-10 10:46
Software	15-06-12 10:06
kundenuploads	30-11-11 14:38
produktuebersicht	18-05-10 16:30

Wechseln Sie nach: **Öffentlich → Software → CODESYS → V3 → CODESYS_Install**

Um die neueste Version zu installieren wählen Sie die höchste Versionsnummer aus. Kopieren Sie die Datei auf Ihr lokales Laufwerk und installieren Sie indem Sie

[CODESYSV35SP4 complete installation](#)

aufrufen.

Dateiname	Geändert
..	
CODESYS 3.5SP4 Release.zip	
CODESYS OPC-Server V3.5SP3Patch6 Setup.exe	
CODESYSV3_complete_installation.zip	



Eine „complete installation“ sollte vor einem Patch installiert werden.

Je nach installierter Version können folgende Bezeichnungen/Menüführungen variieren.

Die Datei hat eine Größe von ca. 0,8 GB.

Sowohl der Download, als auch die Installation dauert einige Zeit.

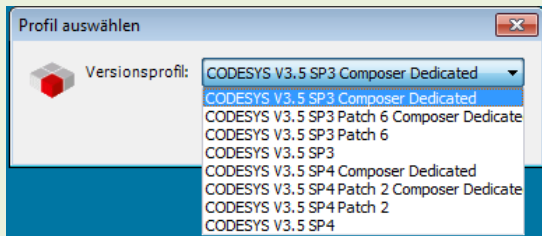
Sollten Sie Windows XP verwenden, installieren Sie zuvor das Service Pack 3.



Entfernen Sie im Link des Desktopitem die Aufrufparameter.



Das Entfernen der Parameter bewirkt, daß sie künftig beim Start ein Profil wählen können.

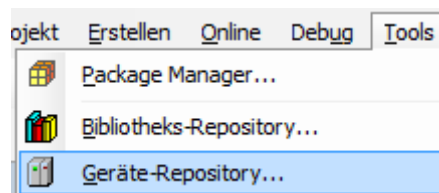



Wechseln Sie nach: **Öffentlich** → **Software** → **CODESYS** → **V3** → **Device**

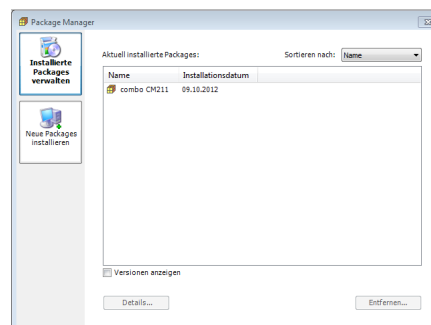
Hier sind für jede Geräteausführung die *.devdesc.xml Dateien abgelegt. Beispielsweise:

[CM211](#)

Innerhalb von CODESYS starten Sie den Device Manager.



Mit dem  können Sie die gewünschten Geräte installieren.



Nun sind Sie am Ziel angekommen. Die Installation ist hiermit abgeschlossen und sie können mit der Programmierung des ersten Gerätes beginnen.

1.2 CODESYS V3

elrest ist seit mehr als 10 Jahren Partner der Fa. 3S Software (www.3s-software.com). CODESYS ist ein SPS Programmierwerkzeug mit verschiedenen Sprachelementen. CODESYS ist mittlerweile das führende SPS Entwicklungssystem. Das bedeutet, dass Tausende von Anwendern täglich CODESYS nutzen. Somit setzt elrest auf den Standard CODESYS.

1.2.1 CODESYS - Marktstandard für Steuerungsprogrammierung nach IEC 61131-3

CODESYS (Controller Development System) ist eine SPS-Programmiersoftware, mit der Sie Steuerungen, Embedded und PC basierte Geräte nach dem internationalen Standard IEC 61131-3 programmieren können. Es besteht aus dem Programmiersystem **CODESYS** sowie dem Laufzeitsystem **CODESYS Control**. Mit dem Laufzeitsystem wird Ihr Gerät zu einer Steuerung, die ganz einfach mit CODESYS programmiert werden kann. Integrierte native Code-Generatoren (Compiler) sorgen dafür, dass der Programmcode mit optimaler Geschwindigkeit abgearbeitet wird.

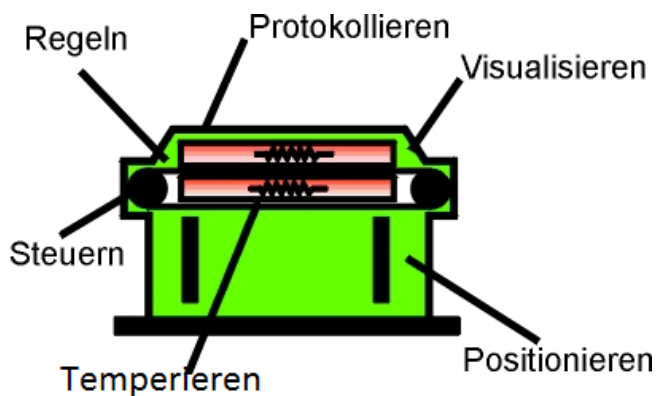
Über 250 namhafte Gerätehersteller aus unterschiedlichen Branchen setzen das Software-Tool CODESYS als Programmierwerkzeug für ihre Automatisierungsgeräte ein. Damit nutzen Tausende von Endanwendern aus dem Maschinen- und Anlagenbau und weiteren Industriezweigen CODESYS bei ihrer täglichen Arbeit. Das sind mehr als bei jedem anderen hardwareunabhängigen IEC 61131-3 Entwicklungssystem. Damit ist CODESYS de facto Marktstandard.

Und das Beste daran: Sie können sich CODESYS nach der [Registrierung](#) kostenlos [herunterladen](#)! Oder Sie sehen sich das [Quick Start Projekt \(Tutorial\)](#) bzw. die [Product Tour](#) an, um einen ersten Eindruck von CODESYS zu bekommen.

2 Beispielprojekte

Was ist zu tun?


- Geschwindigkeiten - Regeln!
- Übergänge - Steuern!
- Werkzeuge - Temperieren!
- Transportgut - Positionieren!
- Vorgänge - Visualisieren!
- Ereignisse – Protokollieren!



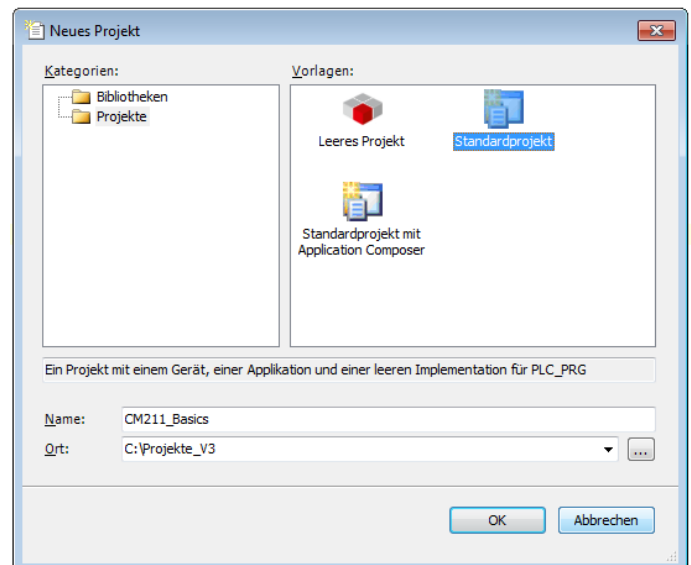
Meine Vorüberlegungen, was will ich?

Die mechanische Konstruktion der Anlage/Maschine ist abgeschlossen. Jetzt werden die Details ausgearbeitet. Bewegungen, Geschwindigkeiten, Kräfte und Prozesse sollen in definierten Größen gesteuert oder geregelt werden. Dazu sind verschiedene Sensorik- und Aktorikprodukte (SuA) erforderlich. Hierzu bietet der Markt eine Vielzahl an SuA mit elektronischen Schnittstellen. Für diese Schnittstellen bietet elrest eine Vielzahl an frei programmierbaren Geräten. Nun können anhand der Hardwareanforderungen die geeigneten Geräte (Module) gewählt werden. Die Funktion wird in späteren Schritten frei programmiert.

Starten Sie die  CODESYS V3.5, nach einer Neuinstallation erscheint ein leeres Projektfenster.

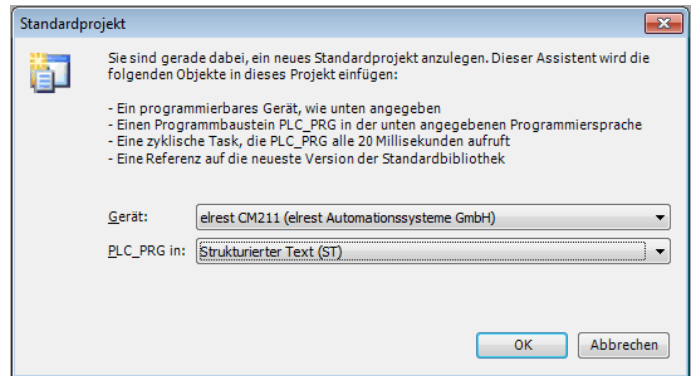
Mit dem  Neu Icon kann ein neues Projekt erstellt werden.

Wählen Sie Standardprojekt, den Namen ihres Projektes z. B. „CM211_Basics“ und den Speicherort des Projektes aus.



Es folgt ein Dialog zur Auswahl des Gerätes und in welcher Programmiersprache Sie entwickeln möchten:

- Elrest CM211
- ST – Strukturierter Text

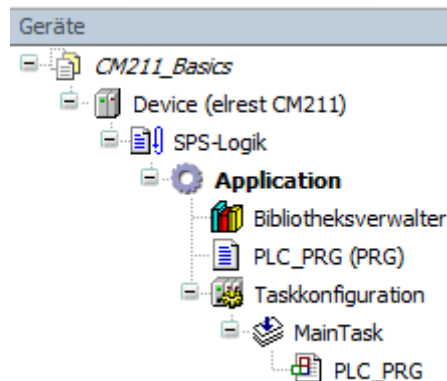
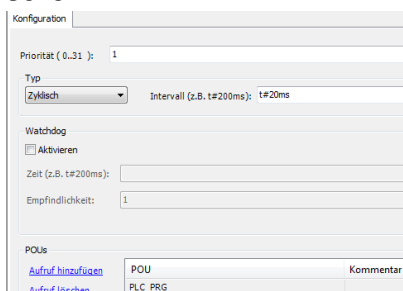


Der Gerätebau wurde wie folgt aufgebaut:

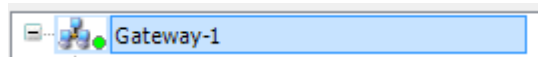
- Device : elrest CM211
- SPS-Logik : bedeutet die Steuerungsebene
- Application : Ist der Name der zu erstellenden Applikation. Diesen Namen können Sie **nicht** ändern. Das Bootprojekt auf der Steuerung trägt den Namen der Applikation.
- Bibliotheksverwalter
Hier werden alle im Projekt verwendeten Bibliotheken mit Versionsnummer gelistet:

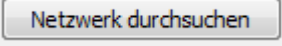
Name	Namensraum	Effektive Version
Standard, 3.5.1.0 (System)	Standard	3.5.1.0
IoStandard, 3.5.1.0 (System)	IoStandard	3.5.1.0

- PLC_PRG : Der erste geschriebene Programmcode.
- Taskkonfiguration : Einstellung welche Programmcodes in welchem Thread laufen sollen.



Aktivieren Sie das



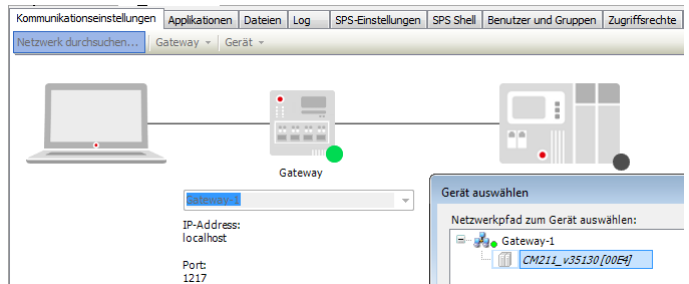
und verbinden Sie das Gerät über Ethernet mit dem Netzwerk. Mit dem Knopf  starten Sie den Autoscans.

Über **Gerät** → **Gerät hinzufügen** kann das Gerät direkt über seine IP-Adresse **192.168.5.228** angesprochen werden.

Es erscheint eine Liste vom gefundenen Geräte.

Selektieren Sie das gewünschte Gerät und

aktivieren es mit dem .

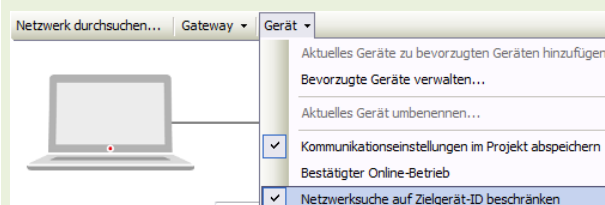


Möchten Sie die Netzwerkeinstellungen ändern, so geht dies nur im verbundenen und eingeloggten Zustand.

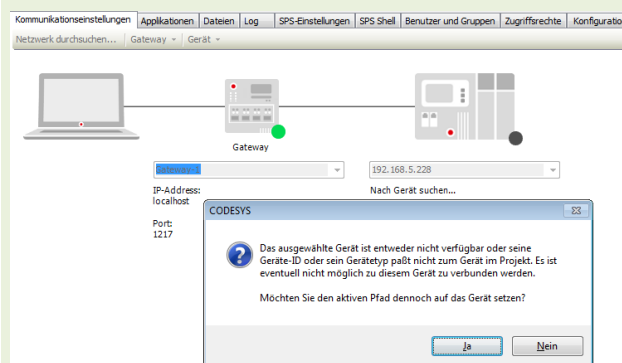
Dazu gehen Sie auf **Online** → **Start**. Anschließend auf **Device** → **Konfiguration**. In dem Reiter Konfiguration können Sie in die Spalte „**vorbereiteter Wert**“ die gewünschten Netzwerkeinstellungen eintragen. Zuletzt schreiben Sie die Werte mit dem Button „**Parameter schreiben**“. Ein Neustart des Gerätes ist notwendig um die Werte zu übernehmen.



Kann das Gerät im Autoscans nicht gefunden werden aktivieren oder deaktivieren Sie die Einstellung „Netzwerksuche auf Zielgeräte-ID beschränken“.




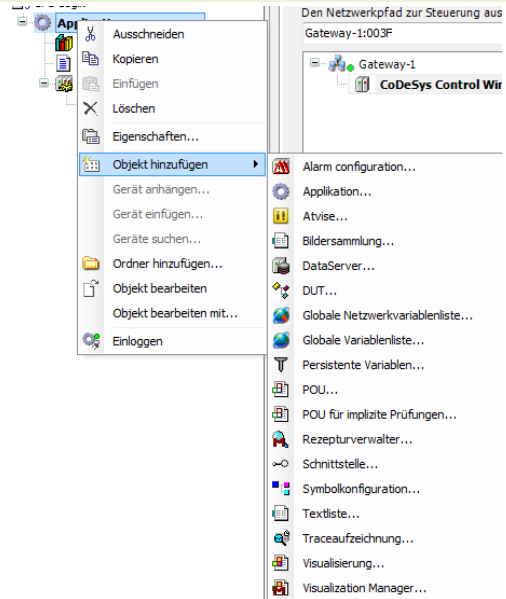
In Fällen in denen der Autoscans nicht funktioniert, z.B. bei einer VPN-Verbindung zum Gerät, kann die IP-Adresse des Gerätes manuell angegeben werden und der aktive Pfad gesetzt werden.




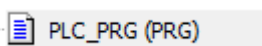


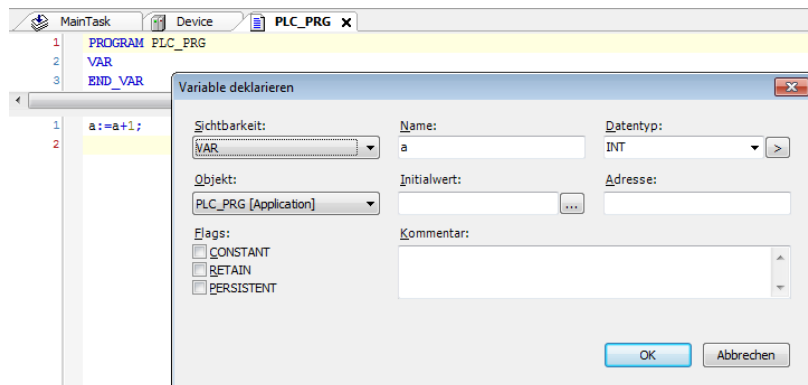
Sie können den Knotennamen nicht ändern.

Selektieren Sie unter dem Gerätenamen die Applikation und mit  können Sie ein Objekt hinzufügen.



2.1 Das erste Programm a:=a+1;

Mit  auf  erscheint der Editor und wir schreiben die Zuweisung a:=a+1; Nach Bestätigung der Zeile werden Sie aufgefordert die unbekannte Variable a zu deklarieren.



Nach der Declaration sieht Ihr Programm wie folgt aus:

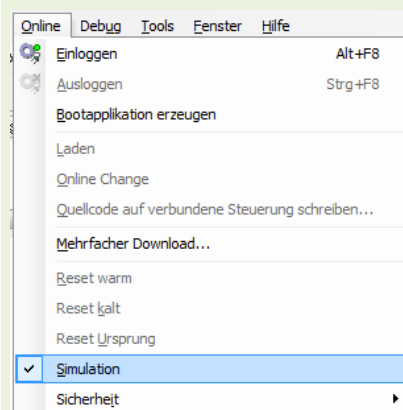


ST

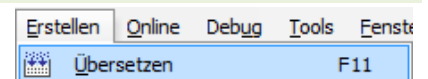


```
PROGRAM PLC_PRG
VAR
  a: INT;
END_VAR
a:=a+1;
```

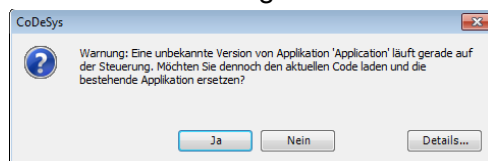
Sie können diese Übung auch ohne Gerät durchführen in der Simulation durchführen, in dem sie Online → Simulation wählen.



Nun muss nur noch das Projekt „Alles bereinigen“, anschließend „Übersetzen“ und mit „Online→Einloggen“ das erstellte Programm downloaden und schon fertig.

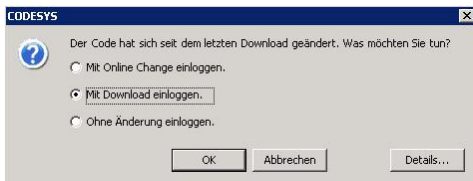


Falls schon ein Programm auf dem Gerät läuft erscheint eine Warnmeldung:



welche mit „Ja“ zu bestätigen ist. Mit **Online→Start** starten Sie die Abarbeitung dieses Programmes.
Mit F5 starten Sie das Programm.

Falls das Programm nicht mehr mit dem auf dem Gerät übereinstimmt erscheint beim Einloggen folgende Meldung:



welche mit „Mit Download einloggen“ und „Ja“ zu bestätigen ist.



Das Programm kann ohne Simulation nicht gestartet/ausgeführt werden, ohne vorher auf das Gerät geladen worden zu sein.

Je nach Version startet die Abarbeitung auch über **Online→Einloggen**.

2.1.1 Die Variable a direkt auf die Digitalausgänge ausgeben.

Ändern Sie Ihre Variable a wie folgt:



ST

```
PROGRAM PLC_PRG
VAR
  a AT %QW0 :INT;
END_VAR
a:=a+1;
```



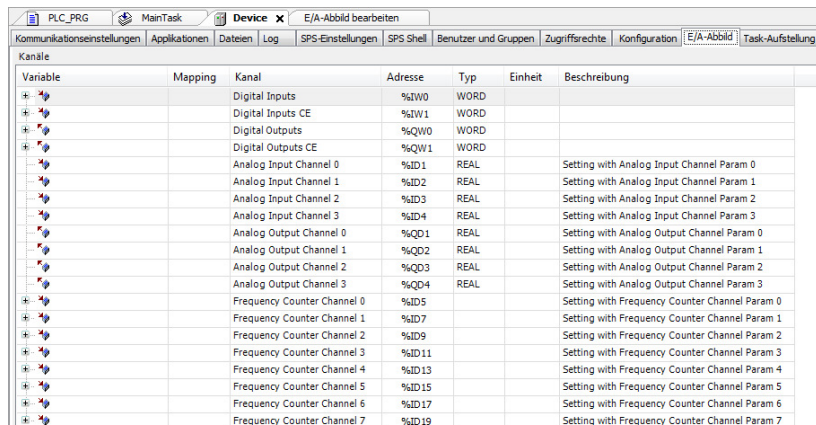
2.1.2 Ein- und Ausgabe Teil der Programmierung

In diesem Abschnitt wird die Bearbeitung von Ein- und Ausgängen besprochen.

Im Tabreiter „E/A Abbild“
können Sie jeden:

- Digitale
- Analoge

Ein- und Ausgänge eine
CODESYS Variablen zuweisen.



Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
		Digital Inputs	%IW0	WORD		
		Digital Inputs CE	%IW1	WORD		
		Digital Outputs	%QW0	WORD		
		Digital Outputs CE	%QW1	WORD		
		Analog Input Channel 0	%ID1	REAL		Setting with Analog Input Channel Param 0
		Analog Input Channel 1	%ID2	REAL		Setting with Analog Input Channel Param 1
		Analog Input Channel 2	%ID3	REAL		Setting with Analog Input Channel Param 2
		Analog Input Channel 3	%ID4	REAL		Setting with Analog Input Channel Param 3
		Analog Output Channel 0	%QD1	REAL		Setting with Analog Output Channel Param 0
		Analog Output Channel 1	%QD2	REAL		Setting with Analog Output Channel Param 1
		Analog Output Channel 2	%QD3	REAL		Setting with Analog Output Channel Param 2
		Analog Output Channel 3	%QD4	REAL		Setting with Analog Output Channel Param 3
		Frequency Counter Channel 0	%ID5			Setting with Frequency Counter Channel Param 0
		Frequency Counter Channel 1	%ID7			Setting with Frequency Counter Channel Param 1
		Frequency Counter Channel 2	%ID9			Setting with Frequency Counter Channel Param 2
		Frequency Counter Channel 3	%ID11			Setting with Frequency Counter Channel Param 3
		Frequency Counter Channel 4	%ID13			Setting with Frequency Counter Channel Param 4
		Frequency Counter Channel 5	%ID15			Setting with Frequency Counter Channel Param 5
		Frequency Counter Channel 6	%ID17			Setting with Frequency Counter Channel Param 6
		Frequency Counter Channel 7	%ID19			Setting with Frequency Counter Channel Param 7

2.1.3 Digitale Ein- und Ausgänge

Mapping- Möglichkeiten:

Dies kann für jenen Ein- oder Ausgang einzeln erfolgen oder mittels einer Variable vom Typ Word für den gesamten Bereich.

boolescher Zugriff z.B.

Digital_Input_00_00 → %IX0.0

Word- Zugriff

Digital_Inputs_M00 → %IW0

Während bei Digitaleingängen ein gemischer Zugriff zwischen Bool und Word Variablen möglich ist, ist die bei Ausgängen zum Schutz gegen konkurrierenden Zugriff nicht zulässig.

Wählen Sie für unsere weiteres Beispiel den booleschen Zugriff.

Bitte übersetzten sie nach Anlegen der Variablen das Projekt durch drücken der Taste <F11>.

Anschließend werden in der Eingabehilfe <F2>:

Name	Typ	Adresse	Herkunft
a	INT	%QW0	
IoConfig_Globals	VAR_GLOBAL		
IoConfig_Globals_Mapping	VAR_GLOBAL		
Digital_Input_00_00	BOOL	%DX0.0	Device
Digital_Input_00_01	BOOL	%DX0.1	Device
Digital_Input_00_02	BOOL	%DX0.2	Device
Digital_Input_00_03	BOOL	%DX0.3	Device
Digital_Inputs_M00	WORD	%IW0	

ihre definierten Ein- und Ausgänge angezeigt und können an beliebiger Stelle im Programmcode verwendet werden.



Digitale Ein –und Ausgänge lassen sich im E/A- Abbild auf verschiedene Arten ansprechen.

- 1.) Einzeln
- 2.) Alle gemeinsam



Im Editorfenster lassen sich mittels der Eingabehilfe (Taste F2) elegant vorhandenen Variablen einführen.

Digitale EA's als boolsche Einzelvariablen..


Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
Digital_Input_00_00		Digital Inputs	%IW0	WORD		Alle Digitaleingänge des Mastermodul
Digital_Input_00_01		Bit0	%IX0.0	BOOL		1. Digitaleingang
Digital_Input_00_02		Bit1	%IX0.1	BOOL		2. Digitaleingang
Digital_Input_00_03		Bit2	%IX0.2	BOOL		3. Digitaleingang
Digital_Input_00_04		Bit3	%IX0.3	BOOL		..
Digital_Input_00_05		Bit4	%IX0.4	BOOL		
Digital_Input_00_06		Bit5	%IX0.5	BOOL		
Digital_Input_00_07		Bit6	%IX0.6	BOOL		
Digital_Input_00_08		Bit7	%IX0.7	BOOL		
Digital_Input_00_09		Bit8	%IX1.0	BOOL		
Digital_Input_00_10		Bit9	%IX1.1	BOOL		
Digital_Input_00_11		Bit10	%IX1.2	BOOL		
Digital_Input_00_12		Bit11	%IX1.3	BOOL		
Digital_Input_00_13		Bit12	%IX1.4	BOOL		
Digital_Input_00_14		Bit13	%IX1.5	BOOL		
Digital_Input_00_15		Bit14	%IX1.6	BOOL		
Digital_Input_00_15		Bit15	%IX1.7	BOOL		
Digital_Inputs_CE		Digital Inputs CE	%IW1	WORD		Alle Digitaleingänge des Aufsteckmodul
Digital_Outputs		Digital Outputs	%QW0	WORD		Alle Digitalausgänge des Mastermodul
Digital_Output_00_00		Bit0	%QX0.0	BOOL		1. Digitalausgang
Digital_Output_00_01		Bit1	%QX0.1	BOOL		2. Digitalausgang
Digital_Output_00_02		Bit2	%QX0.2	BOOL		3. Digitalausgang
Digital_Output_00_03		Bit3	%QX0.3	BOOL		..
Digital_Output_00_04		Bit4	%QX0.4	BOOL		
Digital_Output_00_05		Bit5	%QX0.5	BOOL		
Digital_Output_00_06		Bit6	%QX0.6	BOOL		
Digital_Output_00_07		Bit7	%QX0.7	BOOL		
Digital_Output_00_08		Bit8	%QX1.0	BOOL		
Digital_Output_00_09		Bit9	%QX1.1	BOOL		
Digital_Output_00_10		Bit10	%QX1.2	BOOL		
Digital_Output_00_11		Bit11	%QX1.3	BOOL		
Digital_Output_00_12		Bit12	%QX1.4	BOOL		
Digital_Output_00_13		Bit13	%QX1.5	BOOL		
Digital_Output_00_14		Bit14	%QX1.6	BOOL		
Digital_Output_00_15		Bit15	%QX1.7	BOOL		

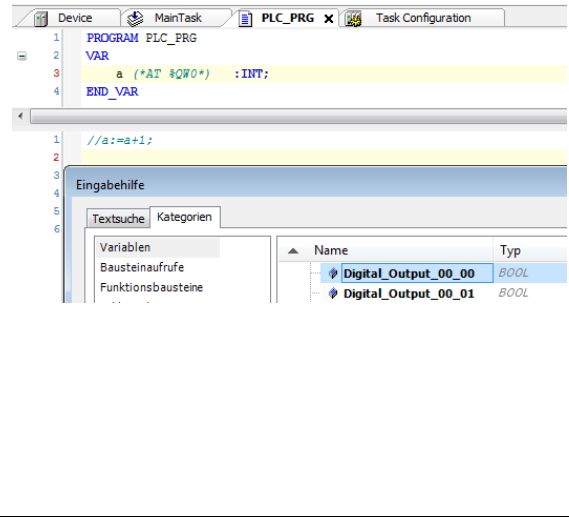
Digitale EA's als Gesamtvariable vom Typ Word.

Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
Digital_Inputs_M00		Digital Inputs	%IW0	WORD		Alle Digitaleingänge des Mastermodul
Digital_Inputs_C00		Digital Inputs CE	%IW1	WORD		Alle Digitaleingänge des Aufsteckmodul
Digital_Outputs_M00		Digital Outputs	%QW0	WORD		Alle Digitalausgänge des Mastermodul
Digital_Outputs_C00		Digital Outputs CE	%QW1	WORD		Alle Digitalausgänge des Aufsteckmodul

Bitte wechseln sie in das Editorfenster und kommentieren Ihren bisherigen Code
`a:=a+1;` aus.
 Er wird nicht mehr benötigt.
 Dies erfolgt durch ein vorhergehendes `//`
 Ebenso das AT `%QW0` durch umklammern mit `(* *)`

Gehen sie in Zeile 2 und drücken die Taste F2.
 Es öffnet sich die Eingabehilfe.

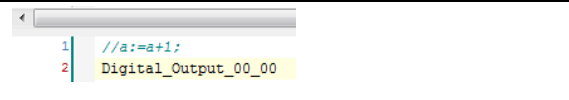
Durch  auf die Variable `Digital_Output_00_00`



```

1 PROGRAM PLC_PRG
2 VAR
3   a (*AT %QW0*) :INT;
4 END_VAR
5
6 //a:=a+1;
7
8 Eingabehilfe
9
10 Textsuche Kategorien
11 Variablen
12 Bausteinaufrufe
13 Funktionsbausteine
14
15 Name Typ
16 Digital_Output_00_00 BOOL
17 Digital_Output_00_01 BOOL
  
```

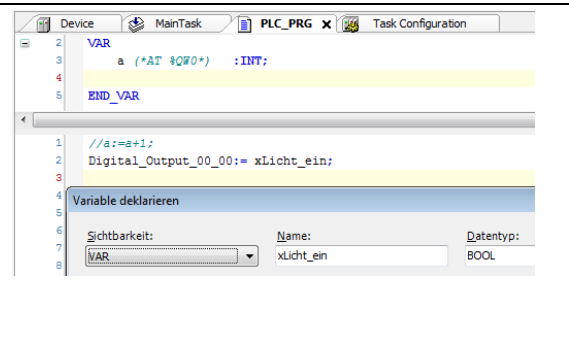
.... wird diese in das Editorfenster Zeile 2 übernommen.



```


1 //a:=a+1;
2 Digital_Output_00_00
  
```

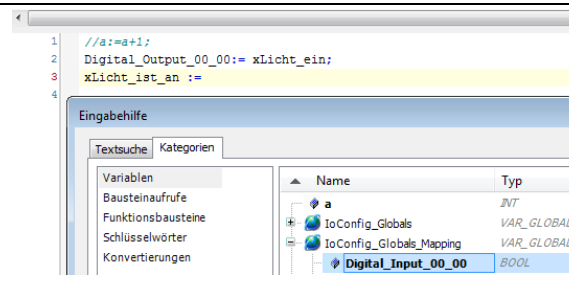
Vervollständigen sie die Zeile mit der Zuweisung
`„:= xLicht_ein;“` und drücken sie [Enter].
 Da die Variable `xLicht_ein` dem System bisher nicht bekannt ist, öffnet sich das Variablendeklarationsfenster.
 Bestätigen sie mit dem vorgeschlagenen Datentyp `BOOL`.



```

1 //a:=a+1;
2 Digital_Output_00_00:= xLicht_ein;
3
4 Variable deklarieren
5
6 Sichtbarkeit: Name: Datentyp:
7 VAR xLicht_ein BOOL
8
  
```

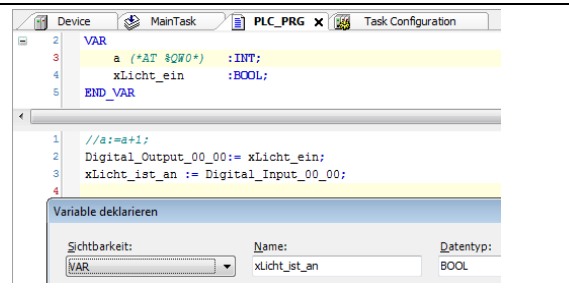
Tippen Sie in Zeile 3
`„xLicht_ist_an := „` und drücken die Taste F2
 (Eingabeaufforderung)
 Wählen Sie durch  die Variable `Digital_Input_00_00`
 Schliessen sie die Zeile mit `„;“` ab.



```

1 //a:=a+1;
2 Digital_Output_00_00:= xLicht_ein;
3 xLicht_ist_an :=
4
5 Eingabehilfe
6
7 Textsuche Kategorien
8 Variablen
9 Bausteinaufrufe
10 Funktionsbausteine
11 Schlüsselwörter
12 Konvertierungen
13
14 Name Typ
15 a INT
16 IoConfig_Globals VAR_GLOBAL
17 IoConfig_Globals_Mapping VAR_GLOBAL
18 Digital_Input_00_00 BOOL
  
```

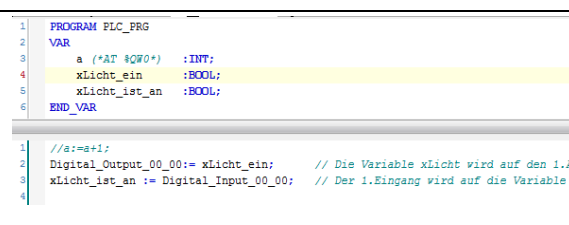
Durch betätigen der Enter-Taste gelangen Sie erneut in das Variablendeklarationsfenster.
 Bestätigen Sie den Deklarationsvorschlag.



```

1 //a:=a+1;
2 Digital_Output_00_00:= xLicht_ein;
3 xLicht_ist_an := Digital_Input_00_00;
4
5 Variable deklarieren
6
7 Sichtbarkeit: Name: Datentyp:
8 VAR xLicht_ist_an BOOL
  
```

Erweitern Sie Zeile 2 und 3 durch Kommentare



```

1 PROGRAM PLC_PRG
2 VAR
3   a (*AT %QW0*) :INT;
4   xLicht_ein :BOOL;
5   xLicht_ist_an :BOOL;
6 END_VAR
7
8 //a:=a+1;
9 Digital_Output_00_00:= xLicht_ein; // Die Variable xLicht wird auf den 1.Ausgang gegeben.
10 xLicht_ist_an := Digital_Input_00_00; // Der 1.Eingang wird auf die Variable xLicht_ist_an gegeben.
  
```

Das Programm zum Lesen eines Eingang und Schreiben eines Ausgang ist fertig.
 Nachfolgend wird mit Online-Debugging im Detail auf jede Zeile eingegangen.



Zuweisungen im Strukturieren Text (ST) erfolgen von Rechts nach Links.
 Digital_Output_00_00 := xLicht_Ein;

Digital_Output_00_00 ← xLicht_Ein



Dem System unbekannte Variablen müssen dekariert werden.
 Das System unterstützt sie hier mit dem Variablendekarationsfenster.




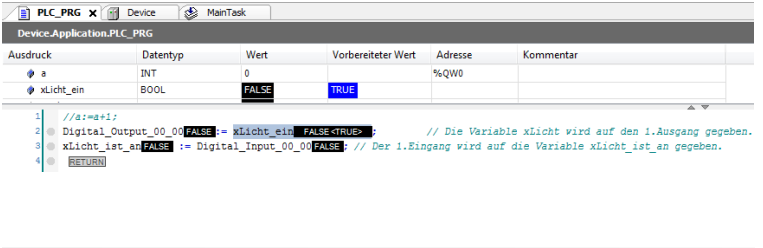
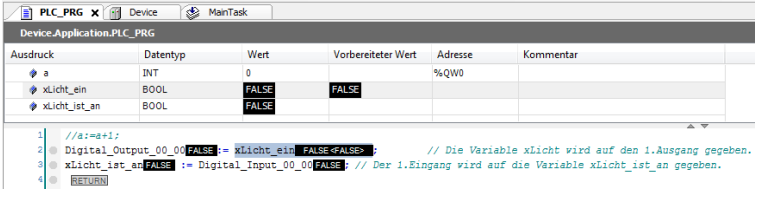

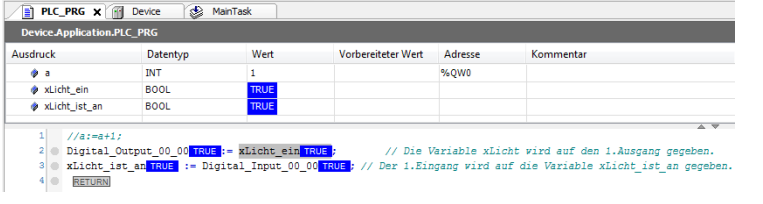


Kommentare können Zeilenweise mit // erfolgen. Hierbei wird alles rechts kommentiert.
 Kommentare können vom Komentaranfang (* bis zum Kommentarabschluss *) erfolgen.
 Komentare werden in grüner Farbe dargestellt.

2.1.4 Einloggen und Starten

<p>F11</p>	<p>Wählen Sie „Erstellen“ und „Übersetzten“</p> <p>Etwaige Fehler werden im Meldungsfenster angezeigt.</p>	
<p>Alt+F8</p>	<p>Wählen Sie „Online“ und „Einloggen“. Sie werden gefragt, wie der Code zur Steuerung übertragen werden soll.</p> <p>Wählen Sie für unsere Beispiele „Mit Download einloggen“</p>	
<p>F5</p>	<p>Starten sie das Programm mit „Debug“ und „Start“</p>	

2.1.5 Schreiben von Werte

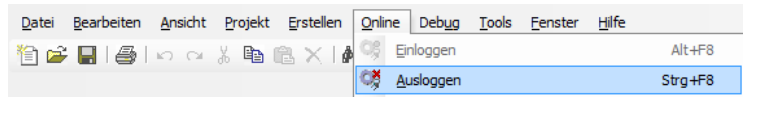
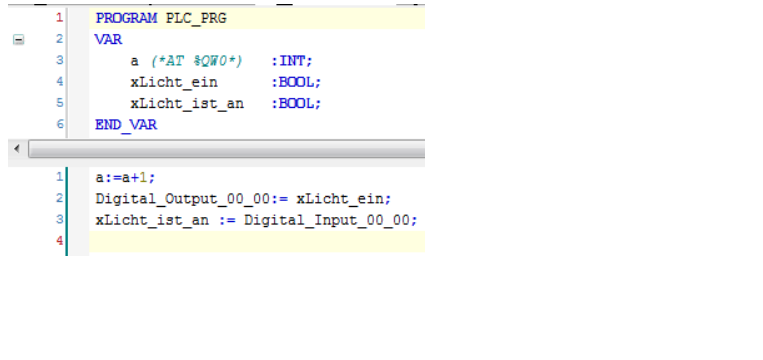
	<p>Tätigen Sie einen  auf die Variable xLicht_ein in Zeile 2.</p> <p>Durch den  wird die Variable mit dem Wert TRUE <u>vorbereitet</u>.</p> <p>Durch einen weiteren  mit den Wert FALSE</p>	 
Alt+F7	<p>Durch einen weiteren  mit „Aufheben“</p>	
Strg+F7		

Das Programm läuft nun.

Durch ändern der Variable xLicht_an schält Ausgang Digital_Output_00_00.

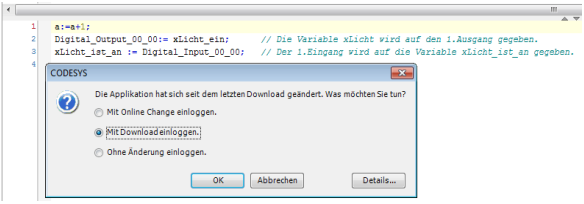
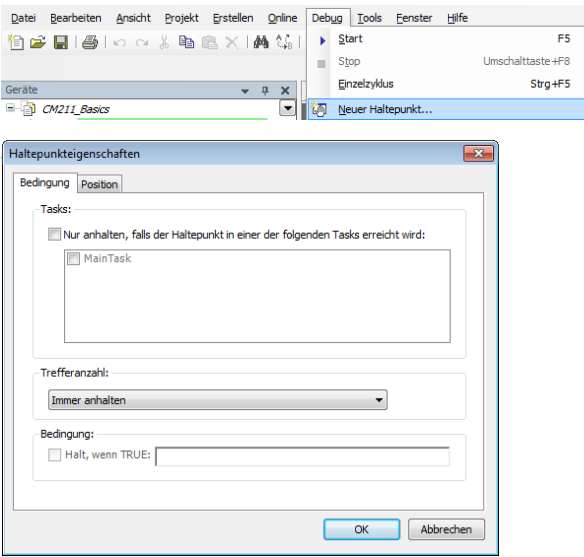
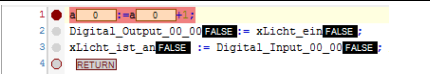
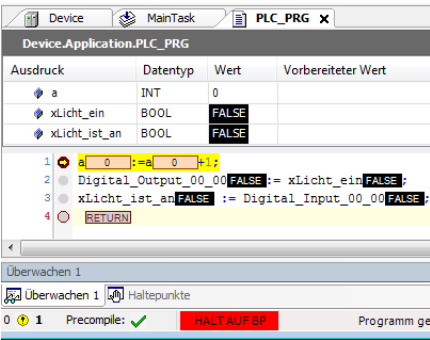
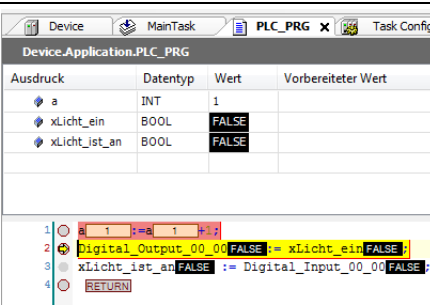
Da es sich bei CM211 um DIO handelt, wird der DigitalOutput_00_00 als Digital_input_00_00 zurückgelesen. (xLicht_ist_an)

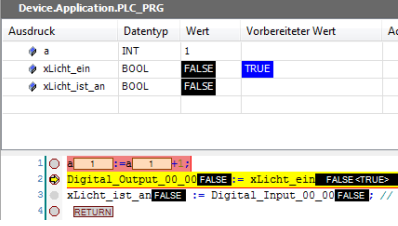
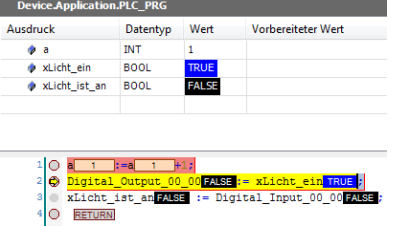

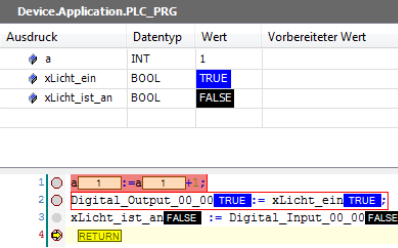
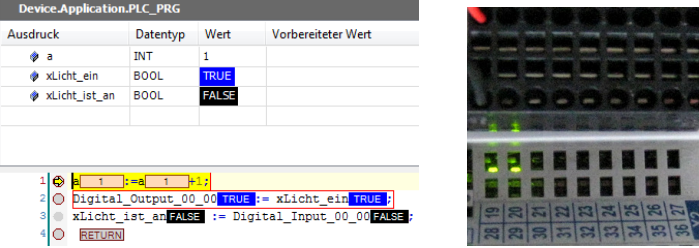
Ausloggen.

Strg+F8	<p>Loggen Sie sich für die folgende Übung aus. „Online“ und „Ausloggen“</p>	
	<p>Entfernen Sie in Zeile 1 // , damit a:=a+1; wieder als Code wirksam ist.</p> <p>Diese Zeile ist für die nachfolgende Übung zwar technisch nicht notwendig, unterstützt jedoch das bessere Verständnis für Breakpointing (setzen von Haltepunkten)</p>	

Breakpoint (Haltepunkt)

Zur Codeanalyse oder Fehlersuche ist das schrittweise durchlaufen von Codezeile von Vorteil. Es kann genau beobachtet werden wie sich Zustände ändern und wohin Sprünge erfolgen.

<p>Alt+F8</p> <p>Einloggen:</p> <p>Loggen Sie sich erneut ein. „Online“ und „Einloggen“</p> <p>Wählen Sie entweder Download oder OnlineChange.</p>	
<p>Breakpoint aktivieren:</p> <p>Wählen Sie „Debug“ und „Neuer Haltepunkt“</p> <p>Bestätigen Sie den nachfolgenden Dialog.</p>	
<p>Das das Programm noch nicht gestartet ist, stehen sie noch auf Zeile 1</p>	
<p>F5</p> <p>Programm starten:</p> <p>Starten Sie das Programm mit „Debug“ und „Start“.</p>	
<p>F8</p> <p>Einzelschritt:</p> <p>.. in Zeile 2.</p> <p>Durch die Anwahl „Debug“ und „Einzelschritt“ gelangen Sie in die nächste Zeile;</p>	

	<p>Beobachtung: Vom Wechsel von Zeile 1 auf Zeile 2 wurde der Variablenwert erwartungsgemäß um 1 erhöht.</p>	
	<p>Wert vorbereiten: Tätigen Sie einen Doppelklick auf den das Zustandsfeld der Variable xLicht_ein</p> <p>Beobachtung: Der Vorbereitete Wert ändert seinen Zustand auf TRUE.</p>	
<p>Strg+F7</p>	<p>Wert schreiben: Wählen Sie „Debug“ und „Werte schreiben“</p> <p>Beobachtung: Die Variable xLicht_ein nimmt den vorbereiteten Wert an.</p>	
<p>F8</p>	<p>Einzelschritt: .. in Zeile 3.</p> <p>Beobachtung: 1.) Die Ausgangsvariable Digital_Output_00_00 wird gesetzt. 2.) Der Digitalausgang an X2-20 bleibt jedoch weiterhin aus!</p>	
<p>F8</p>	<p>Einzelschritt: .. in Zeile 4. (Bausteinende)</p>	
<p>F8</p>	<p>Einzelschritt: .. in Zeile 1 (Bausteinanfang)</p> <p>Beobachtung: Der Digitalausgang an X2-20 wird gesetzt.</p>	
<p>F8</p>	<p>Einzelschritt(e): Bewegen Sie sich mit F8 zum</p>	

	<p>Bausteinende.</p> <p>Beobachtung: Die Variablen Digital_Input_00_00 und xLicht_ist_ein in Zeile 3 bleiben unverändert auf FALSE;</p>																	
F8	<p>Einzelschritt</p> <p>.. in Zeile 1</p> <p>Beobachtung: Jetzt erst nimmt die Variable Digital_Input_00_00 den Werte TRUE an.</p>	<table border="1"> <thead> <tr> <th>Ausdruck</th> <th>Datentyp</th> <th>Wert</th> <th>Vorbereiteter Wert</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>INT</td> <td>2</td> <td></td> </tr> <tr> <td>xLicht_ein</td> <td>BOOL</td> <td>TRUE</td> <td></td> </tr> <tr> <td>xLicht_ist_an</td> <td>BOOL</td> <td>FALSE</td> <td></td> </tr> </tbody> </table> <pre> 1 a_2 :=a_2 -1; 2 Digital_Output_00_00 TRUE := xLicht_ein TRUE; 3 xLicht_ist_an FALSE := Digital_Input_00_00 TRUE; 4 RETURN; </pre>	Ausdruck	Datentyp	Wert	Vorbereiteter Wert	a	INT	2		xLicht_ein	BOOL	TRUE		xLicht_ist_an	BOOL	FALSE	
Ausdruck	Datentyp	Wert	Vorbereiteter Wert															
a	INT	2																
xLicht_ein	BOOL	TRUE																
xLicht_ist_an	BOOL	FALSE																
Strg+F5	<p>Einzelzyklus</p> <p>Wählen Sie „Debug“ und „Einzelzyklus“</p> <p>Durch den Einzelzyklus durchlaufen Sie den gesamten Baustein und gelangen erneut in den Breakpoint in Zeile 1.</p> <p>Beobachtung: Indem Zeile 3 durchlaufen wurde Wird der Zustand vom Digitaleingang erst auf die Variable xLicht_ist_an übernommen.</p>	<table border="1"> <thead> <tr> <th>Ausdruck</th> <th>Datentyp</th> <th>Wert</th> <th>Vorbereiteter Wert</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>INT</td> <td>3</td> <td></td> </tr> <tr> <td>xLicht_ein</td> <td>BOOL</td> <td>TRUE</td> <td></td> </tr> <tr> <td>xLicht_ist_an</td> <td>BOOL</td> <td>TRUE</td> <td></td> </tr> </tbody> </table> <pre> 1 a_3 :=a_3 -1; 2 Digital_Output_00_00 TRUE := xLicht_ein TRUE; 3 xLicht_ist_an TRUE := Digital_Input_00_00 TRUE; 4 RETURN; </pre>	Ausdruck	Datentyp	Wert	Vorbereiteter Wert	a	INT	3		xLicht_ein	BOOL	TRUE		xLicht_ist_an	BOOL	TRUE	
Ausdruck	Datentyp	Wert	Vorbereiteter Wert															
a	INT	3																
xLicht_ein	BOOL	TRUE																
xLicht_ist_an	BOOL	TRUE																

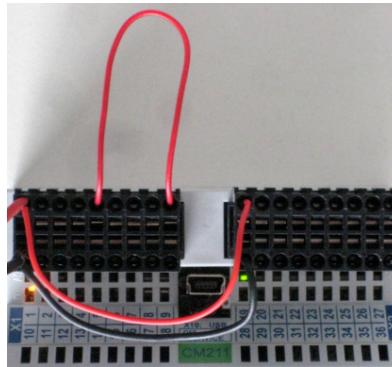


Die Ausgänge werden erst am Bausteinende gesetzt, bzw die Eingänge gelesen. Bitte beachten Sie dies z.B. bei der Programmierung von Schleifen.

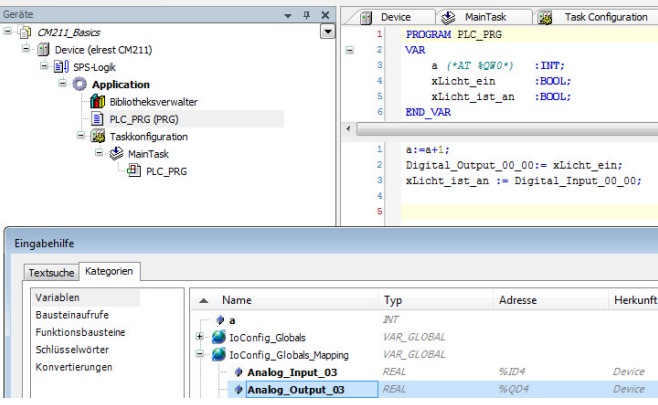
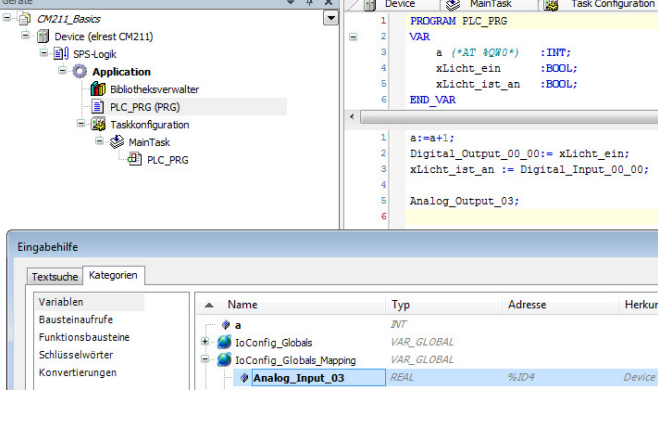


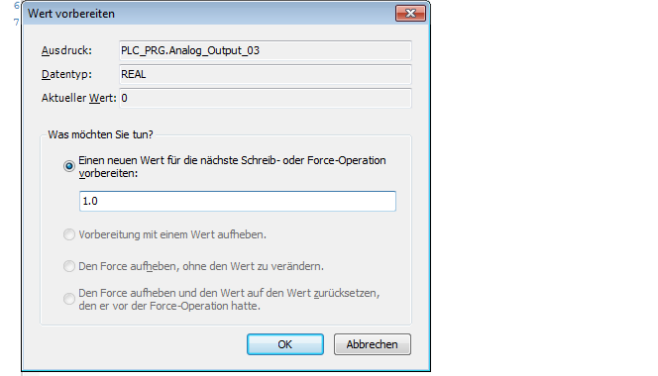
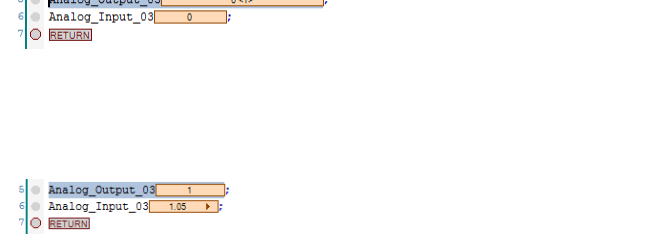
2.1.5.1 Analoge Ein- und Ausgänge

Verbinden Sie für das nachfolgende Beispiel den Analogeingang 3 (X1-5) mit dem Analogausgang 3 (X1-9)

Aufgabe der folgenden Übung wird sein eine Analogspannung über den Ausgang auszugeben und über einen Analogeingang wieder einzulesen.

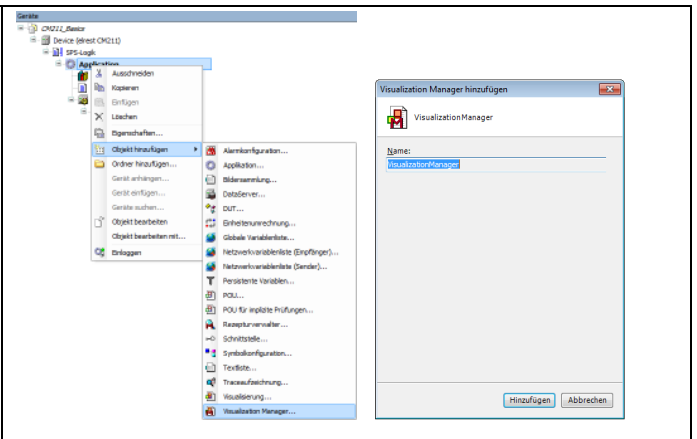
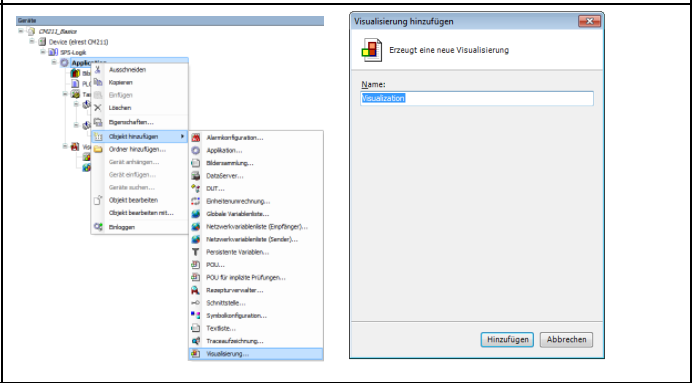
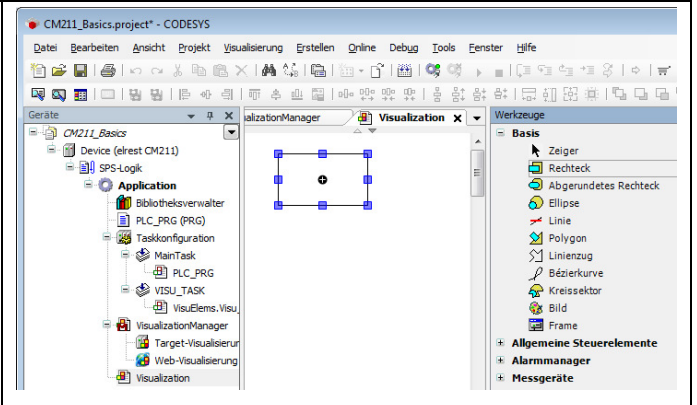





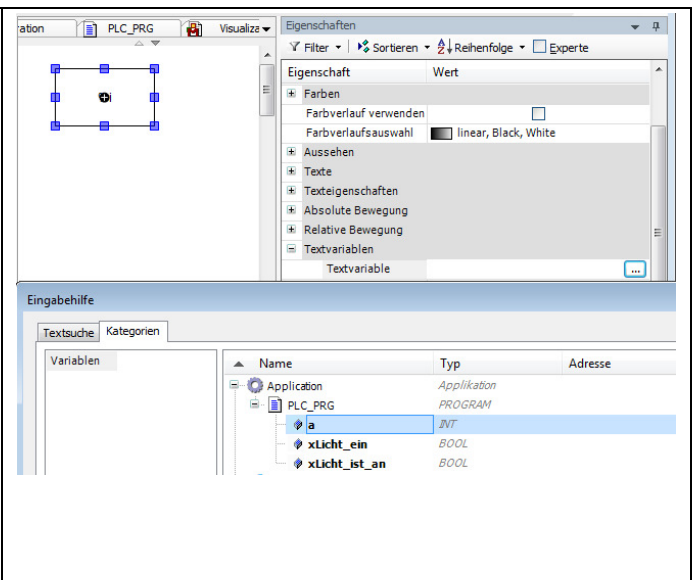
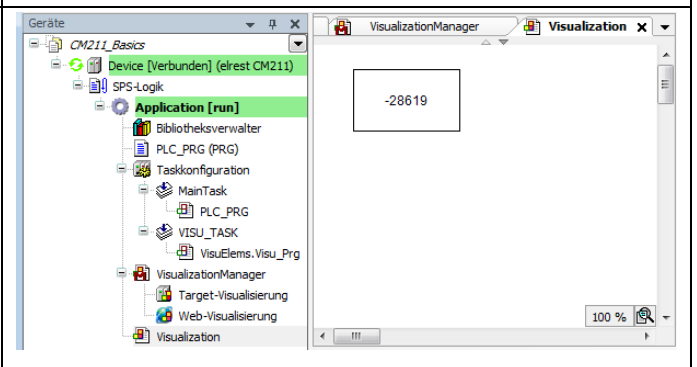
	Wählen Sie Device - Konfiguration	
	<p>Klappen Sie unter dem Parameter <code>Analog_Input_Channel_Param_3</code> über das [+] – Symbol das Untermenü auf.</p> <p>Wählen Sie unter Sensortyp den Wert <code>0V - +10 Volt</code></p>	
	Wählen Sie unter <code>Analog_Output_Channel_Param_3</code> den Wert <code>0V -+10 Volt</code> .	
	<p>Weisen Sie den Kanälen Variablenamen zu. in diesem Fall:</p> <p><code>Analog Input_Channel 3</code> → <code>Analog_Input_03</code></p> <p><code>Analog Output_Channel 3</code> → <code>Analog_Output_03</code></p>	

<p>F2</p>	<p>Wechseln Sie wieder in ihm PLC_PRG(PRG) gehen Sie in Zeile 5 und drücken die Taste F2. Es öffnet sich die Eingabehilfe. Wählen Sie Analog Output. Schliessen sie die Zeile mit Semicolon ab.</p>	
<p>Wiederholen Sie den Vorgang in Zeile 6 mit Analog_Input_03</p>	<p>Wiederholen Sie den Vorgang in Zeile 6 mit Analog_Input_03</p>	
<p>In Zeile 5 und 6 sollten jetzt beide Variablen stehen, von einem Semicolon abgeschlossen.</p>	<p>In Zeile 5 und 6 sollten jetzt beide Variablen stehen, von einem Semicolon abgeschlossen.</p>	
<p>Strg+F8 F5</p>	<p>Loggen Sie sich ein Starten sie das Programm.</p>	
<p>Klicken Sie auf das Variablenfenster von Analog_Output_03 um einen Wert vorzubereiten. 0..10 Volt entsprechen hierbei dem Wert 0.0 .. 1.0 REAL</p>	<p>Klicken Sie auf das Variablenfenster von Analog_Output_03 um einen Wert vorzubereiten. 0..10 Volt entsprechen hierbei dem Wert 0.0 .. 1.0 REAL</p>	
<p>Strg+F7</p>	<p>Schreiben Sie den soeben vorbereiteten Wert zur Steuerung. Am Analog_Input_03 liegt nun der Wert 1.05 an, was einer Spannung in</p>	

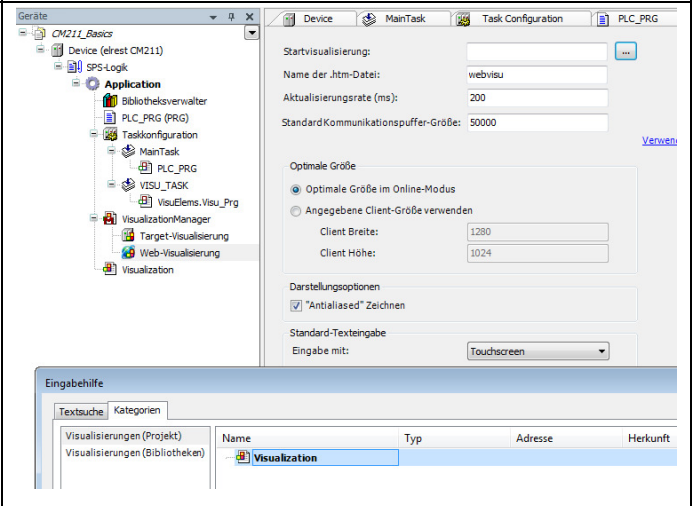
	Höhe von 10,5 Volt entspricht.	

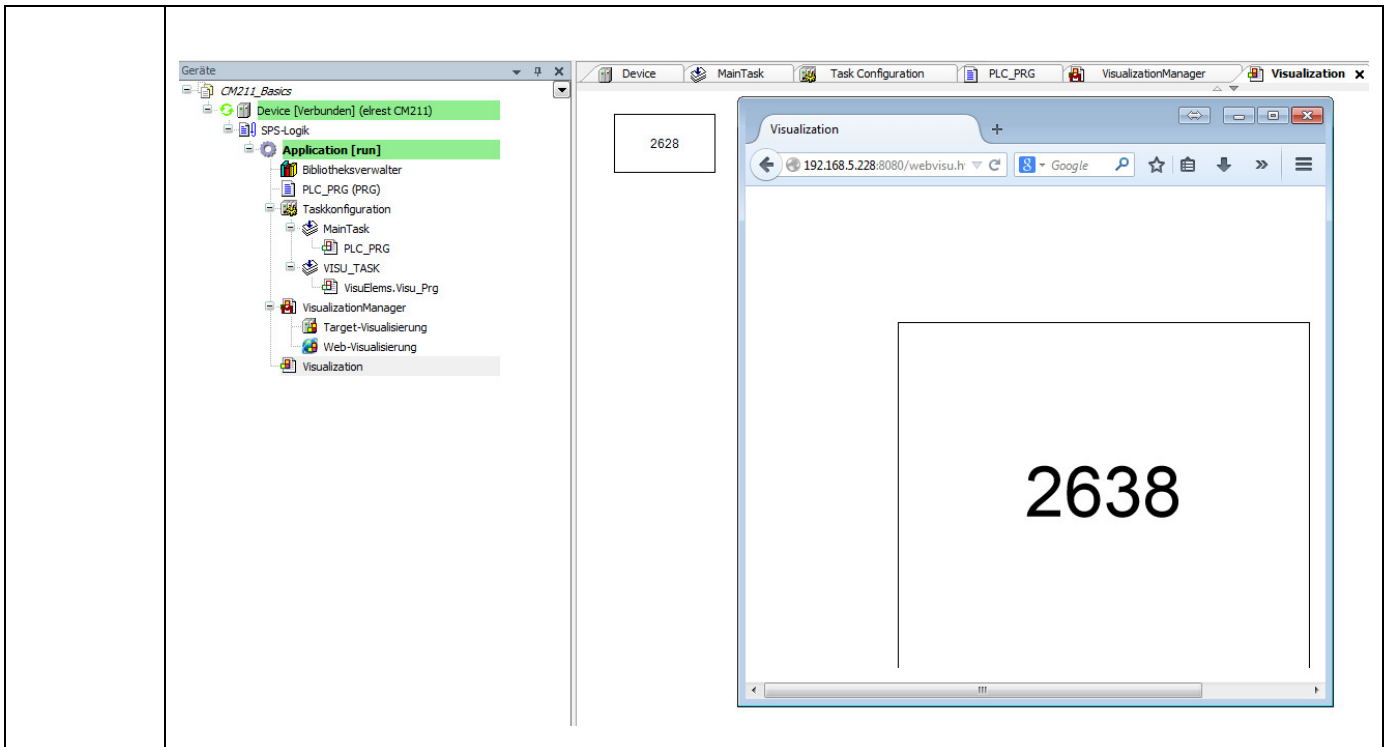
2.1.5.2 Verbinden der Datenpunkte mit einer HMI

	<p>Binden Sie unter Applikation – Objekte hinzufügen Den Visualizations Manager ein.</p> <p>Vergeben Sie einen Namen oder bestätigen sie den Vorgeschlagenen.</p>	
	<p>Binden Sie unter Applikation – Objekte hinzufügen Eine Visualisierung ein.</p> <p>Vergeben Sie einen Namen oder bestätigen sie den Vorgeschlagenen.</p>	
	<p>Wechseln Sie in den Bereich „Visualization“ wählen als Werkzeug ein Rechteck und zeichnen Sie dieses.</p>	
	<p>Klicken Sie mittig in das Rechteck. Es erscheint ein Eingabefeld Geben Sie %i als Platzhalter für einen Integer-Variable ein.</p>	

	<p>Klicken Sie nicht mittig in das Rechteck. Das Tool „Werkzeuge“ am Rand wechselt zum Tool „Eigenschaften“</p> <p>Öffnen sie durch Anklicken des [+] – Symbol das Untermenü Textvariablen und  Sie in das Feld Textvariable. Es scheint eine Taste mit 3 Punkten. [...] Betätigen Sie diese und es öffnet sich die Eingabehilfe.</p> <p>Wählen sie durch  die Variable „a“</p>	
<p>Alt+F8 F5</p>	<p>Loggen Sie sich ein und Starten sie das Programm</p>	

2.1.5.3 WebVisu (Webbrowser)

<p>Strg+F8</p>	<p>Um das vorangegangene Targevisu auf Webvisu zu erweitern, loggen sie sich bitte aus.</p>	
	<p>Gehen sie im VisualisierungsManager zu Web-Visualisierung.</p> <p>Das Feld Startvisualisierung ist leer. Tragen sie hier bitte mit Hilfe der Eingabehilfe (Taste [...]) die Visualization ein.</p>	
<p>Alt+F8 F5</p>	<p>Laden und Starten sie die Applikation erneut.</p>	
	<p>Verbinden sich sich mit Ihrem Webbrowser zu http://<geräteip>:8080/webvisu.htm</p>	

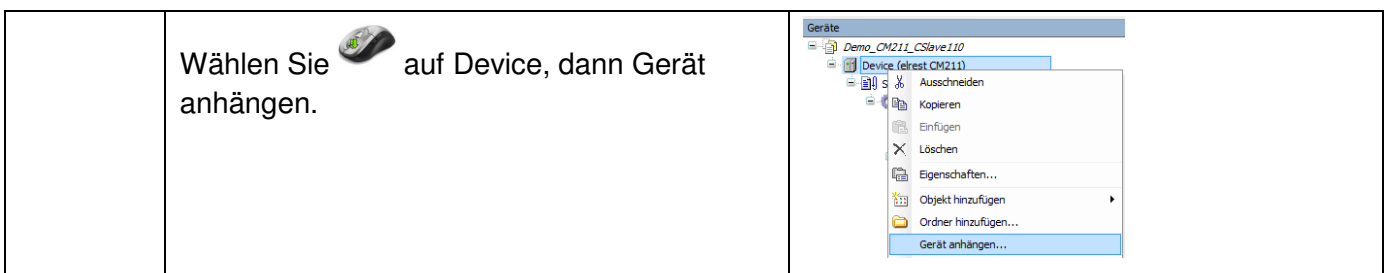



2.2 Einbindes eines CanOpen- Slaves (CS110)

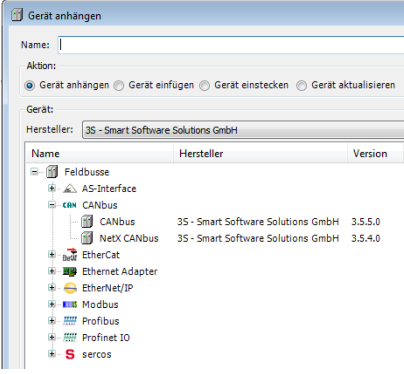
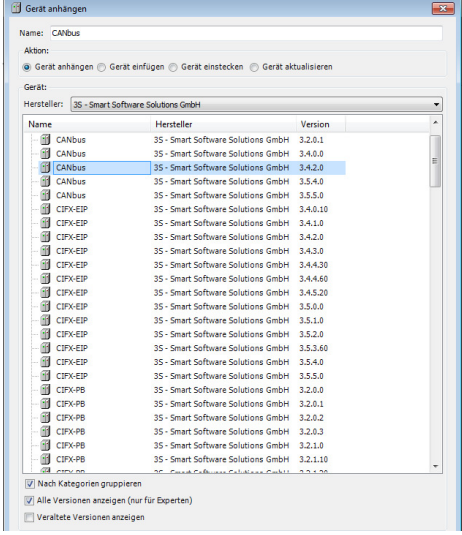

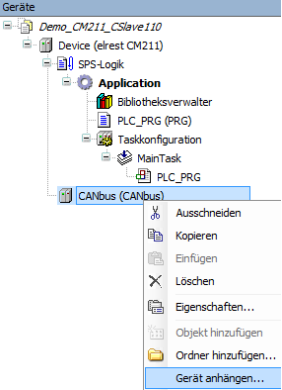
Ihre Mastersteuerung kann durch eine Vielzahl von Slavegeräten erweitert werden. Die generelle Vorgehensweise zum Einbinden eines CANOPEN- Slaves wird hier in Folge erläutert.

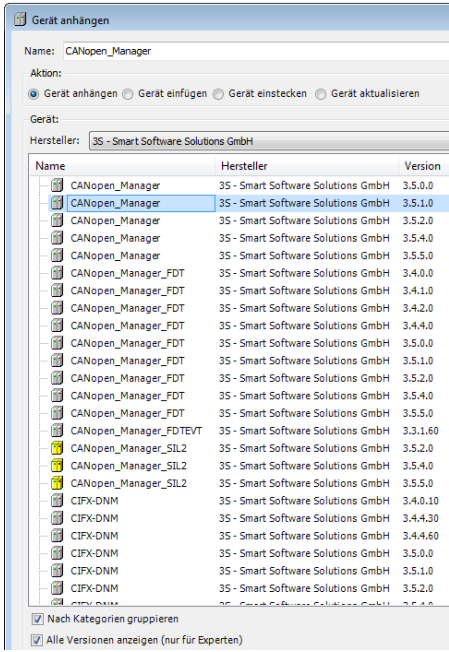

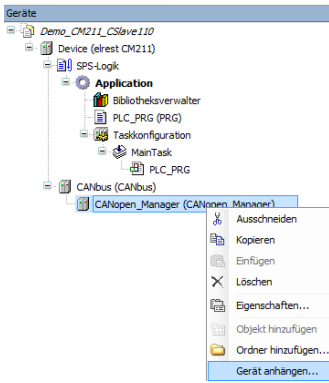
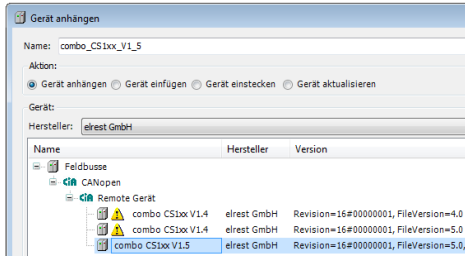
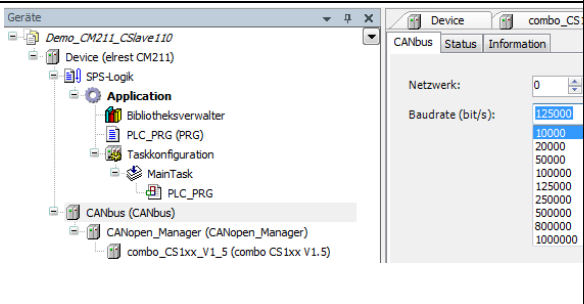


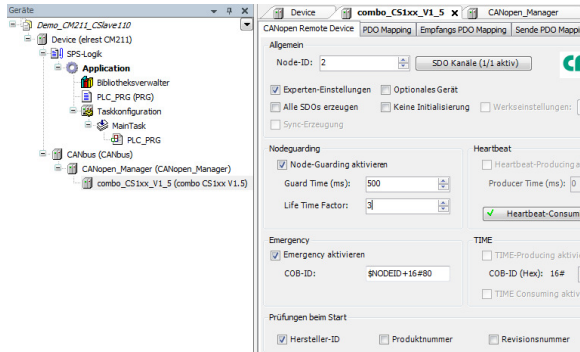
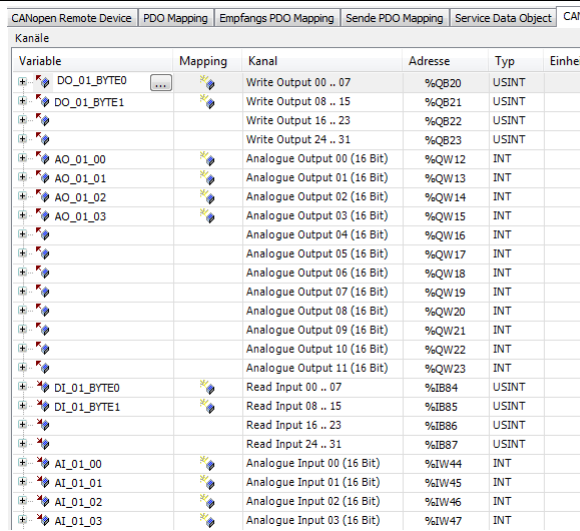
Durch einmaliges importieren dieser Übung als Archivfile, werden allen notwendigen Versionsstände in ihre Entwicklungsumgebung geladen.

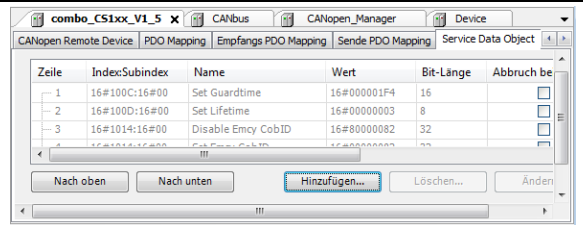
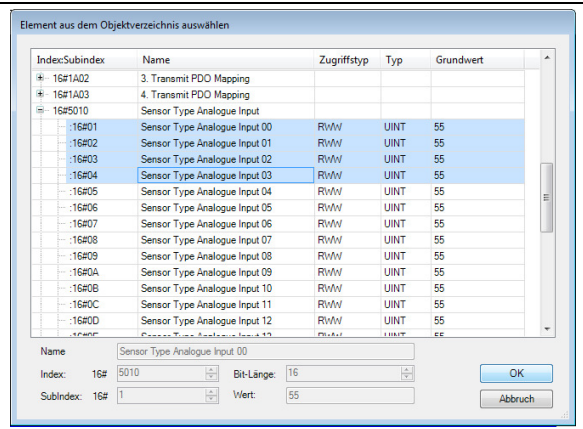
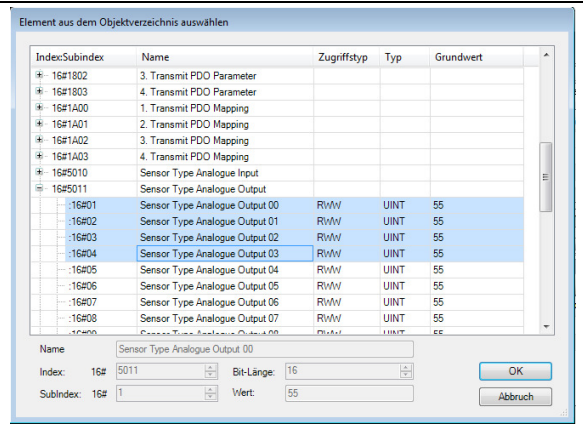
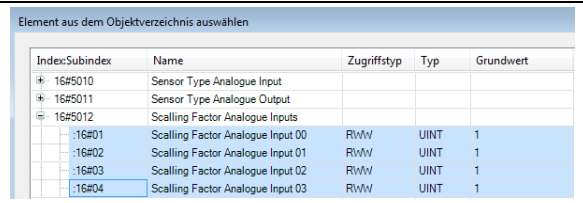
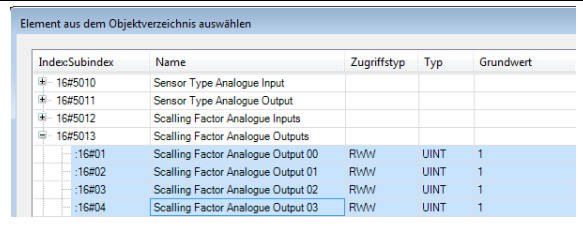
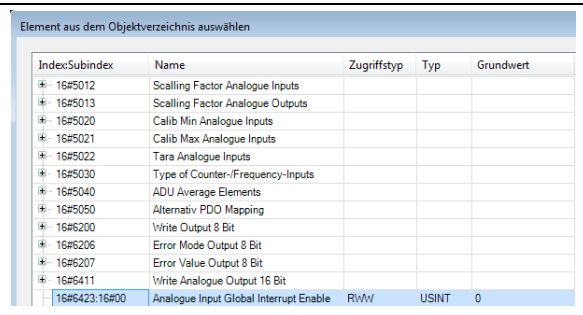


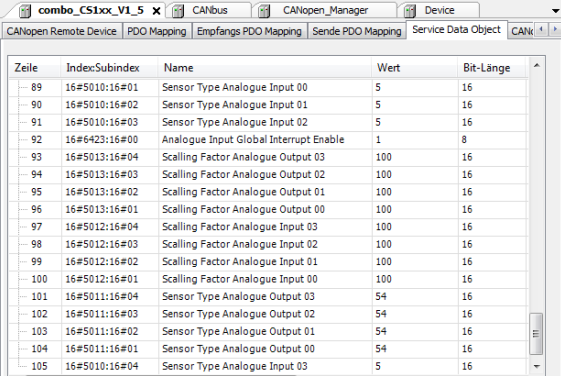
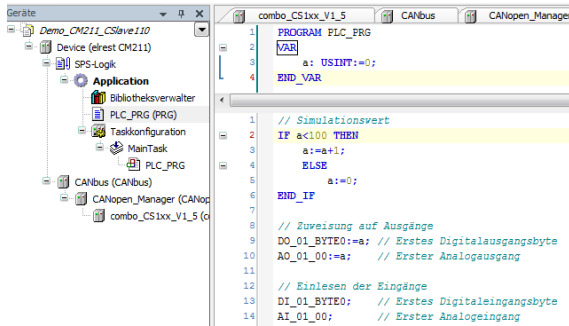
Wählen Sie  auf Device, dann Gerät anhängen.

	<p>Es erscheint eine Auswahl der Feldbusse mit den aktuellsten Versionen.</p>	
	<p>Wählen Sie „Alle Versionen anzeigen“ (nur für Experten“. Sie erhalten eine Übersicht aller Canbus-Versionen.</p> <p>Wählen sie jetzt die Version 3.4.2.0 Dann Taste „Gerät einhängen“</p>	
	<p>Wählen Sie  auf Canbus, dann Gerät anhängen.</p>	

<p>Wählen Sie „Alle Versionen anzeigen“. (nur für Experten) Sie erhalten eine Übersicht aller CanOpen_Manager- Versionen.</p> <p>Wählen sie jetzt die Version 3.5.1.0 Dann Taste „Gerät einhängen“</p>		
<p>Wählen Sie  auf CanOpenManager, dann Gerät anhängen.</p>		
<p>Wählen sie unter Hersteller: Elrest GmbH Die EDS combo_CS1xx_V1.5</p>		
<p>Nun kann die Konfiguration erfolgen</p>		
<p>Canbus – Baudrate Doppelklick auf den Canbus, dann unter der Registerkarte CANBus die Baudrate 125000 wählen.</p>		

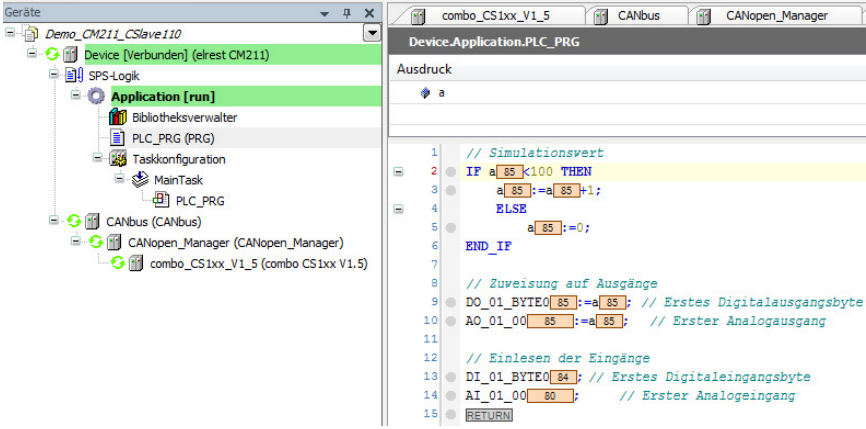
	<p>Slave-Node ID: Doppelklick auf den Slave, dann Wahl der Registerkarte „CANopen Remote Device“.</p> <p>Tragen Sie hier die Knotennummer (Node-ID) ein, welche sie am Hex-Schalter des Slaves eingestellt haben. Im Beispiel = 2.</p> <p>Node- Guarding (Slaveüberwachung) 1.) Wählen sie Experten-Einstellung 2.) Deaktivieren Sie Heartbeat 3.) Aktivieren Sie Node-Guarding 4.) Tragen Sie die Werte 500 und 3 ein.</p>	 <p>Es findet nun eine beidseitige Kommunikationsüberwachung alle 500ms statt. Wird diese 3x hintereinander nicht bestätigt, gilt die Kommunikation als unterbrochen.</p>																																																																																																																																																						
	<p>Gehen sie innerhalb der Slavekonfiguration in die Registerkarte „CanOpen E/A- Abbild“. Durch Doppelklick in das Feld Variable, lassen sich diese Namen zuweisen.</p> <p>Vergeben Sie für die 16 digitalen Ein- und Ausgänge Namen. Ebenso für die 4 analogen Ein- und Ausgänge.</p>	 <table border="1"> <thead> <tr> <th>Variable</th> <th>Mapping</th> <th>Kanal</th> <th>Adresse</th> <th>Typ</th> <th>Einheit</th> </tr> </thead> <tbody> <tr><td>* DO_01_BYTE0</td><td></td><td>Write Output 00 .. 07</td><td>%QB20</td><td>USINT</td><td></td></tr> <tr><td>* DO_01_BYTE1</td><td></td><td>Write Output 08 .. 15</td><td>%QB21</td><td>USINT</td><td></td></tr> <tr><td>* DO_01_BYTE2</td><td></td><td>Write Output 16 .. 23</td><td>%QB22</td><td>USINT</td><td></td></tr> <tr><td>* DO_01_BYTE3</td><td></td><td>Write Output 24 .. 31</td><td>%QB23</td><td>USINT</td><td></td></tr> <tr><td>* AO_01_00</td><td></td><td>Analogue Output 00 (16 Bit)</td><td>%QW12</td><td>INT</td><td></td></tr> <tr><td>* AO_01_01</td><td></td><td>Analogue Output 01 (16 Bit)</td><td>%QW13</td><td>INT</td><td></td></tr> <tr><td>* AO_01_02</td><td></td><td>Analogue Output 02 (16 Bit)</td><td>%QW14</td><td>INT</td><td></td></tr> <tr><td>* AO_01_03</td><td></td><td>Analogue Output 03 (16 Bit)</td><td>%QW15</td><td>INT</td><td></td></tr> <tr><td>* AO_01_04</td><td></td><td>Analogue Output 04 (16 Bit)</td><td>%QW16</td><td>INT</td><td></td></tr> <tr><td>* AO_01_05</td><td></td><td>Analogue Output 05 (16 Bit)</td><td>%QW17</td><td>INT</td><td></td></tr> <tr><td>* AO_01_06</td><td></td><td>Analogue Output 06 (16 Bit)</td><td>%QW18</td><td>INT</td><td></td></tr> <tr><td>* AO_01_07</td><td></td><td>Analogue Output 07 (16 Bit)</td><td>%QW19</td><td>INT</td><td></td></tr> <tr><td>* AO_01_08</td><td></td><td>Analogue Output 08 (16 Bit)</td><td>%QW20</td><td>INT</td><td></td></tr> <tr><td>* AO_01_09</td><td></td><td>Analogue Output 09 (16 Bit)</td><td>%QW21</td><td>INT</td><td></td></tr> <tr><td>* AO_01_10</td><td></td><td>Analogue Output 10 (16 Bit)</td><td>%QW22</td><td>INT</td><td></td></tr> <tr><td>* AO_01_11</td><td></td><td>Analogue Output 11 (16 Bit)</td><td>%QW23</td><td>INT</td><td></td></tr> <tr><td>* DI_01_BYTE0</td><td></td><td>Read Input 00 .. 07</td><td>%IB84</td><td>USINT</td><td></td></tr> <tr><td>* DI_01_BYTE1</td><td></td><td>Read Input 08 .. 15</td><td>%IB85</td><td>USINT</td><td></td></tr> <tr><td>* DI_01_BYTE2</td><td></td><td>Read Input 16 .. 23</td><td>%IB86</td><td>USINT</td><td></td></tr> <tr><td>* DI_01_BYTE3</td><td></td><td>Read Input 24 .. 31</td><td>%IB87</td><td>USINT</td><td></td></tr> <tr><td>* AI_01_00</td><td></td><td>Analogue Input 00 (16 Bit)</td><td>%IW44</td><td>INT</td><td></td></tr> <tr><td>* AI_01_01</td><td></td><td>Analogue Input 01 (16 Bit)</td><td>%IW45</td><td>INT</td><td></td></tr> <tr><td>* AI_01_02</td><td></td><td>Analogue Input 02 (16 Bit)</td><td>%IW46</td><td>INT</td><td></td></tr> <tr><td>* AI_01_03</td><td></td><td>Analogue Input 03 (16 Bit)</td><td>%IW47</td><td>INT</td><td></td></tr> </tbody> </table>	Variable	Mapping	Kanal	Adresse	Typ	Einheit	* DO_01_BYTE0		Write Output 00 .. 07	%QB20	USINT		* DO_01_BYTE1		Write Output 08 .. 15	%QB21	USINT		* DO_01_BYTE2		Write Output 16 .. 23	%QB22	USINT		* DO_01_BYTE3		Write Output 24 .. 31	%QB23	USINT		* AO_01_00		Analogue Output 00 (16 Bit)	%QW12	INT		* AO_01_01		Analogue Output 01 (16 Bit)	%QW13	INT		* AO_01_02		Analogue Output 02 (16 Bit)	%QW14	INT		* AO_01_03		Analogue Output 03 (16 Bit)	%QW15	INT		* AO_01_04		Analogue Output 04 (16 Bit)	%QW16	INT		* AO_01_05		Analogue Output 05 (16 Bit)	%QW17	INT		* AO_01_06		Analogue Output 06 (16 Bit)	%QW18	INT		* AO_01_07		Analogue Output 07 (16 Bit)	%QW19	INT		* AO_01_08		Analogue Output 08 (16 Bit)	%QW20	INT		* AO_01_09		Analogue Output 09 (16 Bit)	%QW21	INT		* AO_01_10		Analogue Output 10 (16 Bit)	%QW22	INT		* AO_01_11		Analogue Output 11 (16 Bit)	%QW23	INT		* DI_01_BYTE0		Read Input 00 .. 07	%IB84	USINT		* DI_01_BYTE1		Read Input 08 .. 15	%IB85	USINT		* DI_01_BYTE2		Read Input 16 .. 23	%IB86	USINT		* DI_01_BYTE3		Read Input 24 .. 31	%IB87	USINT		* AI_01_00		Analogue Input 00 (16 Bit)	%IW44	INT		* AI_01_01		Analogue Input 01 (16 Bit)	%IW45	INT		* AI_01_02		Analogue Input 02 (16 Bit)	%IW46	INT		* AI_01_03		Analogue Input 03 (16 Bit)	%IW47	INT	
Variable	Mapping	Kanal	Adresse	Typ	Einheit																																																																																																																																																			
* DO_01_BYTE0		Write Output 00 .. 07	%QB20	USINT																																																																																																																																																				
* DO_01_BYTE1		Write Output 08 .. 15	%QB21	USINT																																																																																																																																																				
* DO_01_BYTE2		Write Output 16 .. 23	%QB22	USINT																																																																																																																																																				
* DO_01_BYTE3		Write Output 24 .. 31	%QB23	USINT																																																																																																																																																				
* AO_01_00		Analogue Output 00 (16 Bit)	%QW12	INT																																																																																																																																																				
* AO_01_01		Analogue Output 01 (16 Bit)	%QW13	INT																																																																																																																																																				
* AO_01_02		Analogue Output 02 (16 Bit)	%QW14	INT																																																																																																																																																				
* AO_01_03		Analogue Output 03 (16 Bit)	%QW15	INT																																																																																																																																																				
* AO_01_04		Analogue Output 04 (16 Bit)	%QW16	INT																																																																																																																																																				
* AO_01_05		Analogue Output 05 (16 Bit)	%QW17	INT																																																																																																																																																				
* AO_01_06		Analogue Output 06 (16 Bit)	%QW18	INT																																																																																																																																																				
* AO_01_07		Analogue Output 07 (16 Bit)	%QW19	INT																																																																																																																																																				
* AO_01_08		Analogue Output 08 (16 Bit)	%QW20	INT																																																																																																																																																				
* AO_01_09		Analogue Output 09 (16 Bit)	%QW21	INT																																																																																																																																																				
* AO_01_10		Analogue Output 10 (16 Bit)	%QW22	INT																																																																																																																																																				
* AO_01_11		Analogue Output 11 (16 Bit)	%QW23	INT																																																																																																																																																				
* DI_01_BYTE0		Read Input 00 .. 07	%IB84	USINT																																																																																																																																																				
* DI_01_BYTE1		Read Input 08 .. 15	%IB85	USINT																																																																																																																																																				
* DI_01_BYTE2		Read Input 16 .. 23	%IB86	USINT																																																																																																																																																				
* DI_01_BYTE3		Read Input 24 .. 31	%IB87	USINT																																																																																																																																																				
* AI_01_00		Analogue Input 00 (16 Bit)	%IW44	INT																																																																																																																																																				
* AI_01_01		Analogue Input 01 (16 Bit)	%IW45	INT																																																																																																																																																				
* AI_01_02		Analogue Input 02 (16 Bit)	%IW46	INT																																																																																																																																																				
* AI_01_03		Analogue Input 03 (16 Bit)	%IW47	INT																																																																																																																																																				
	<p>Wählen sie für dieses Demoprojekt abschließend „Variablen Immer aktualisieren“</p>	<p>Hinweis: Im CanOpenstack werden nur Variablen bearbeitet, die im Code auch tatsächlich verwendet werden. Benutzen Sie diese Option daher für diese Demo, deaktivieren sie diese für Ihre künftigen Projekte.</p>																																																																																																																																																						
	<p>Konfigurieren der Analogkanäle:</p> <p>Während der Startphase (PRE-OPERATIONAL) müssen mittels SDO Werte im Register gesetzt werden.</p> <p>Hierzu werden nachfolgende die Register ausgewählt und Werte gesetzt.</p>																																																																																																																																																							

	<p>Betätigen Sie die Taste [Hinzufügen] unter Combo_CS1xx_V1.5 → Service Data Object</p>	
	<p>Analogeingänge – Sensor Type</p> <p>Wählen Sie die ersten 4 Kanäle des Objekt 16#5010 aus und bestätigen mit [OK]</p>	
	<p>Analogausgänge – Sensor Type</p> <p>Wählen Sie die ersten 4 Kanäle des Objekt 16#5011 aus und bestätigen mit [OK]</p>	
	<p>Analogeingänge – Scalling Factor</p> <p>Wählen Sie die ersten 4 Kanäle des Objekt 16#5012 aus und bestätigen mit [OK]</p>	
	<p>Analogausgänge – Scalling Factor</p> <p>Wählen Sie die ersten 4 Kanäle des Objekt 16#5012 aus und bestätigen mit [OK]</p>	
	<p>Analogkanäle aktivieren</p> <p>Wählen Sie das Objekt 16#6423 aus und bestätigen mit [OK]</p>	

	<p>Alle notwendigen Register zum Parametrieren der Analogen sind gewählt.</p>																																																																																											
	<p>Registereinträge:</p> <p>Sensor Type Analogue Input 5 = 0_10_Volt_IN 6 = 0_20_mA_IN 21 = PT100</p> <p>Sensor Type Analogue Output 54 = 0_10_Volt_OUT</p> <p>Scaling Faktor 100 = Mit diesem Wert wird bei der Übertragung multipliziert. Da es sich bei den Registereinträgen um 16Bit Ganzzahlen handelt, erlaubt dies Nachkommastellen z.B. 53 → 5,3 Volt da der Werte 0,0-1,0 der Spannung 0 – 10 Volt entspricht.</p> <p>Analogue Input Global Interrupt Enable 1 = Die Analogverarbeitung wird eingeschalten</p>	 <table border="1"> <thead> <tr> <th>Zeile</th> <th>Index/Subindex</th> <th>Name</th> <th>Wert</th> <th>Bit-Länge</th> </tr> </thead> <tbody> <tr><td>89</td><td>16#5010:16#01</td><td>Sensor Type Analogue Input 00</td><td>5</td><td>16</td></tr> <tr><td>90</td><td>16#5010:16#02</td><td>Sensor Type Analogue Input 01</td><td>5</td><td>16</td></tr> <tr><td>91</td><td>16#5010:16#03</td><td>Sensor Type Analogue Input 02</td><td>5</td><td>16</td></tr> <tr><td>92</td><td>16#6423:16#00</td><td>Analogue Input Global Interrupt Enable</td><td>1</td><td>8</td></tr> <tr><td>93</td><td>16#5013:16#04</td><td>Scaling Factor Analogue Output 03</td><td>100</td><td>16</td></tr> <tr><td>94</td><td>16#5013:16#03</td><td>Scaling Factor Analogue Output 02</td><td>100</td><td>16</td></tr> <tr><td>95</td><td>16#5013:16#02</td><td>Scaling Factor Analogue Output 01</td><td>100</td><td>16</td></tr> <tr><td>96</td><td>16#5013:16#01</td><td>Scaling Factor Analogue Output 00</td><td>100</td><td>16</td></tr> <tr><td>97</td><td>16#5012:16#04</td><td>Scaling Factor Analogue Input 03</td><td>100</td><td>16</td></tr> <tr><td>98</td><td>16#5012:16#03</td><td>Scaling Factor Analogue Input 02</td><td>100</td><td>16</td></tr> <tr><td>99</td><td>16#5012:16#02</td><td>Scaling Factor Analogue Input 01</td><td>100</td><td>16</td></tr> <tr><td>100</td><td>16#5012:16#01</td><td>Scaling Factor Analogue Input 00</td><td>100</td><td>16</td></tr> <tr><td>101</td><td>16#5011:16#04</td><td>Sensor Type Analogue Output 03</td><td>54</td><td>16</td></tr> <tr><td>102</td><td>16#5011:16#03</td><td>Sensor Type Analogue Output 02</td><td>54</td><td>16</td></tr> <tr><td>103</td><td>16#5011:16#02</td><td>Sensor Type Analogue Output 01</td><td>54</td><td>16</td></tr> <tr><td>104</td><td>16#5011:16#01</td><td>Sensor Type Analogue Output 00</td><td>54</td><td>16</td></tr> <tr><td>105</td><td>16#5010:16#04</td><td>Sensor Type Analogue Input 03</td><td>5</td><td>16</td></tr> </tbody> </table>	Zeile	Index/Subindex	Name	Wert	Bit-Länge	89	16#5010:16#01	Sensor Type Analogue Input 00	5	16	90	16#5010:16#02	Sensor Type Analogue Input 01	5	16	91	16#5010:16#03	Sensor Type Analogue Input 02	5	16	92	16#6423:16#00	Analogue Input Global Interrupt Enable	1	8	93	16#5013:16#04	Scaling Factor Analogue Output 03	100	16	94	16#5013:16#03	Scaling Factor Analogue Output 02	100	16	95	16#5013:16#02	Scaling Factor Analogue Output 01	100	16	96	16#5013:16#01	Scaling Factor Analogue Output 00	100	16	97	16#5012:16#04	Scaling Factor Analogue Input 03	100	16	98	16#5012:16#03	Scaling Factor Analogue Input 02	100	16	99	16#5012:16#02	Scaling Factor Analogue Input 01	100	16	100	16#5012:16#01	Scaling Factor Analogue Input 00	100	16	101	16#5011:16#04	Sensor Type Analogue Output 03	54	16	102	16#5011:16#03	Sensor Type Analogue Output 02	54	16	103	16#5011:16#02	Sensor Type Analogue Output 01	54	16	104	16#5011:16#01	Sensor Type Analogue Output 00	54	16	105	16#5010:16#04	Sensor Type Analogue Input 03	5	16
Zeile	Index/Subindex	Name	Wert	Bit-Länge																																																																																								
89	16#5010:16#01	Sensor Type Analogue Input 00	5	16																																																																																								
90	16#5010:16#02	Sensor Type Analogue Input 01	5	16																																																																																								
91	16#5010:16#03	Sensor Type Analogue Input 02	5	16																																																																																								
92	16#6423:16#00	Analogue Input Global Interrupt Enable	1	8																																																																																								
93	16#5013:16#04	Scaling Factor Analogue Output 03	100	16																																																																																								
94	16#5013:16#03	Scaling Factor Analogue Output 02	100	16																																																																																								
95	16#5013:16#02	Scaling Factor Analogue Output 01	100	16																																																																																								
96	16#5013:16#01	Scaling Factor Analogue Output 00	100	16																																																																																								
97	16#5012:16#04	Scaling Factor Analogue Input 03	100	16																																																																																								
98	16#5012:16#03	Scaling Factor Analogue Input 02	100	16																																																																																								
99	16#5012:16#02	Scaling Factor Analogue Input 01	100	16																																																																																								
100	16#5012:16#01	Scaling Factor Analogue Input 00	100	16																																																																																								
101	16#5011:16#04	Sensor Type Analogue Output 03	54	16																																																																																								
102	16#5011:16#03	Sensor Type Analogue Output 02	54	16																																																																																								
103	16#5011:16#02	Sensor Type Analogue Output 01	54	16																																																																																								
104	16#5011:16#01	Sensor Type Analogue Output 00	54	16																																																																																								
105	16#5010:16#04	Sensor Type Analogue Input 03	5	16																																																																																								
	<p>Das Einstellen des Slaves ist somit abgeschlossen.</p>																																																																																											
<p>T1</p>	<p>Funktionstest</p> <p>Zur Erprobung der soeben eingegebenen Konfiguration benutzen wir folgenden Funktionstest in PLC_PRG()</p> <pre> PROGRAM PLC_PRG VAR a: USINT:=0; END_VAR // Simulationswert IF a<100 THEN a:=a+1; ELSE a:=0; END_IF // Zuweisung auf Ausgänge DO_01_BYTE0:=a; // Erstes Digitalausgangsbyte AO_01_00:=a; // Erster Analogausgang // Einlesen der Eingänge DI_01_BYTE0; // Erstes Digitaleingangsbyte AI_01_00; // Erster Analogeingang </pre>	 <pre> 1 PROGRAM PLC_PRG 2 VAR 3 a: USINT:=0; 4 END_VAR 5 6 // Simulationswert 7 IF a<100 THEN 8 a:=a+1; 9 ELSE 10 a:=0; 11 END_IF 12 13 // Zuweisung auf Ausgänge 14 DO_01_BYTE0:=a; // Erstes Digitalausgangsbyte 15 AO_01_00:=a; // Erster Analogausgang 16 17 // Einlesen der Eingänge 18 DI_01_BYTE0; // Erstes Digitaleingangsbyte 19 AI_01_00; // Erster Analogeingang </pre>																																																																																										

T2	Übersetzen Sie das Programm und Starten es.	

T3



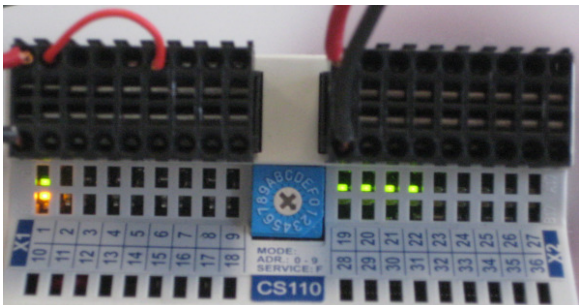
Erklärung des Screenshots:
 Grüne Pfeile → Teilnehmer arbeitet korrekt

Variable a → wird zwischen Zeile 2 und 6 mit jedem Zyklus inkrementiert.
 Bei Erreichen des Wert > 100 wird a wieder 0.

in Zeile 9 wird VAR a auf die Digitalausgänge gelegt.
 in Zeile 10 wird VAR a auf den ersten Analogausgang gelegt.
 0..100 entsprechen hier 0-10 Volt.

in Zeile 13 wird das gesetzte digitale 1.Byte rückgelesen,
 da es sich um DIO handelt wird jeder gesetzte Ausgang auch als Eingang rückgelesen.

in Zeile 14 wird die gesetzte Analogspannung über den Analogeingang zurückgelesen, da zwischen AI_01_00 auf AO_01_00 eine Drahtbrücke angebracht wurde.



2.2.1 *Abschluss der Programmierung*



Hoffentlich konnten wir Ihr Interesse an diesem Programmierool wecken und freuen uns Sie bei elrest begrüßen zu dürfen.

3 Support

Hotline

Für zusätzliche Unterstützung und Informationen, können Sie unsere Hotline zu folgenden Zeiten:

Mo-Fr: von 8.00- 12.00 und 13.00 - 16.30

Außerhalb dieser Zeiten, können Sie uns per e-mail oder fax erreichen:

Telefon: +49 (0) 7021/92025-33
Fax: +49 (0) 7021/92025-29
E-mail: hotline@elrest.de

Training und Workshops

Wir bieten Ausbildung oder projekt basierte Workshops zu elrest Produkte an.

Für weitere Informationen, kontaktieren Sie bitte unsere Vertriebsabteilung:

Telefon: +49 (0) 7021/92025-0
Fax: +49 (0) 7021/92025-29
E-mail: vertrieb@elrest.de

4 Historie

Datum	Name	Kapitel	Änderung
26.02.2014	GS	erstellt	
23.07.2014	Ne	Basics	Komplett Beispiele für CM211 erstellt
14.08.2014	Ne	CS110	Einbindes eines CanOpen Combo-Slave

© 2014 elrest Automationssysteme GmbH. Alle Rechte vorbehalten.

Die in diesem Dokument enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens elrest Automationssysteme GmbH dar. Die Software und/oder Datenbanken, die in diesem Dokument beschrieben sind, werden unter einer Lizenzvereinbarung und einer Geheimhaltungsvereinbarung zur Verfügung gestellt. Die Software und/oder Datenbanken dürfen nur nach Maßgabe der Bedingungen der Vereinbarung benutzt oder kopiert werden. Es ist rechtswidrig, die Software auf ein anderes Medium zu kopieren, soweit das nicht ausdrücklich in der Lizenz- oder Geheimhaltungsvereinbarung erlaubt wird. Ohne ausdrückliche schriftliche Erlaubnis der elrest Automationssysteme GmbH dürfen weder dieses Handbuch noch Teile davon für irgendwelche Zwecke in irgendeiner Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie oder Aufzeichnung reproduziert oder übertragen werden. Abbildungen und Beschreibungen sowie Abmessungen und technische Daten entsprechen den Gegebenheiten oder Absichten zum Zeitpunkt des Druckes dieses Prospektes. Änderungen jeder Art, insbesondere soweit sie sich aus technischem Fortschritt, wirtschaftlicher Ausführung oder ähnlichem ergeben, bleiben vorbehalten. Die externe Verschaltung der Geräte erfolgt in Eigenverantwortung.