

# Modbus Kommunikation zwischen zwei Maschinen

Deveci, Fentler 5CHIT

## Vorraussetzungen

- Codesys V3.5 SP13 Patch1 (wurde verwendet)
- 2 Rechner
- 2 Applikationen, nicht die selbe (Müssen **64 Bit** sein, sonst funktioniert nicht)
- aktive Internetverbindung

## Firewall Konfiguration

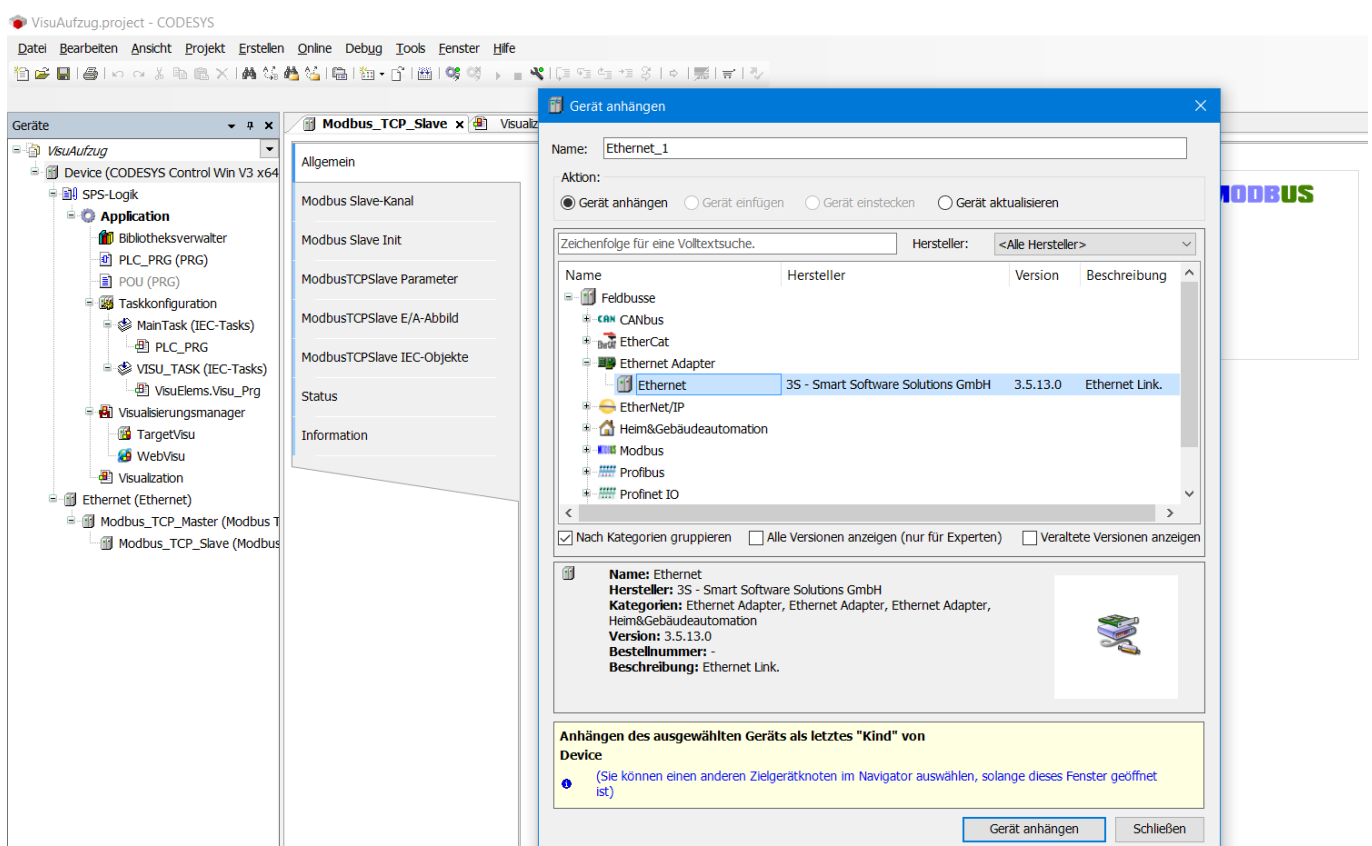
Wenn die Firewall blockiert, dann muss man den Port öffnen. In dem Fall muss man unter Windows in **Firewall & Netzwerkschutz/Erweiterte Einstellungen/Eingehende Regeln -> Neue Regel/Port**

Die Modbus Kommunikation verwendet standardmäßig den Port 502. Daher wird der auch bei der Eingangsregel in der Firewall erlaubt.

## Master - Konfiguration

### Geräte hinzufügen

Da man in der Aufgabe mit Ethernet arbeitet, muss als erstes ein Ethernet Knoten erstellt werden. Darunter werden dann der Master und der Slave erstellt und konfiguriert. Damit ein Ethernet-Knoten erstellt wird, muss man folgendermaßen navigieren: **Rechtsklick auf Device -> Gerät anhängen** Daraufhin muss unter **Ethernet Adapter Ethernet** ausgewählt werden, damit der Knoten erstellt wird.



Anschließend muss der Modbus TCP Master an das zuvor erstellte Ethernet-Element angehängt werden. Dafür **Rechtsklick auf Ethernet -> Gerät anhängen... -> Modbus -> Modbus TCP Master -> Modbus TCP Master -> Gerät anhängen**

### Ethernet konfigurieren

Nun muss der Ethernet-Knoten konfiguriert werden. **Doppelklick auf Ethernet -> Auf den Reiter Allgemein switchen -> Auf den Button "... drücken** und das gewünschte Netzwerkinterface auswählen. Die IP-Adresse, Subnetzmaske und das Gateway wird automatisch eingegeben. Somit ist die Konfiguration für den Master abgeschlossen

### Slave mit Master verknüpfen

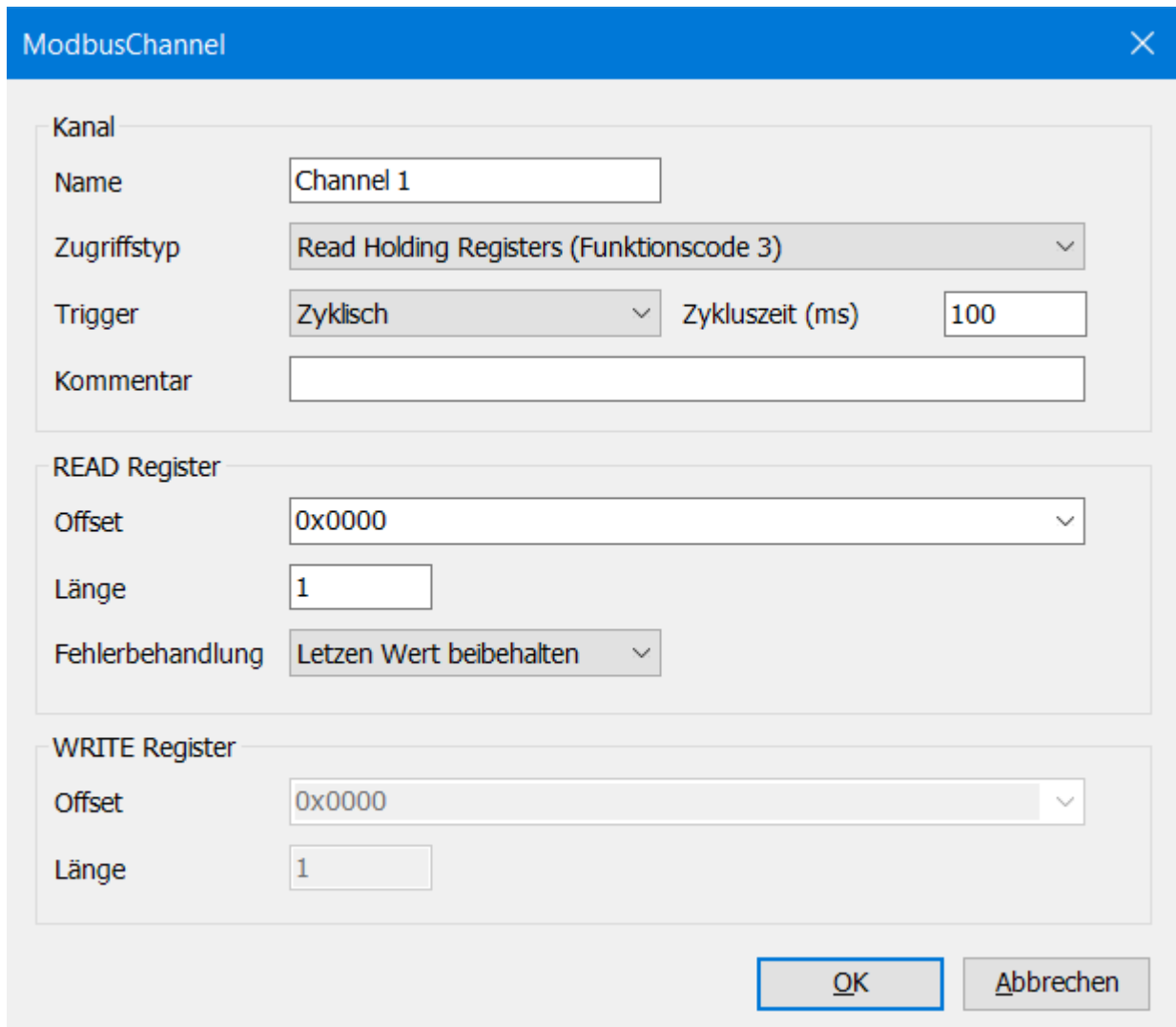
Nun muss der Slave mit dem Master verknüpft werden. Dazu muss unter dem Masterknoten der Modbus TCP Slave eingebunden werden. **Dazu Rechtsklick auf Modbus\_TCP\_Master -> Gerät anhängen... -> Modbus TCP Slave**

### Slave konfigurieren

**Linksklick auf Modbus\_TCP\_Slave -> Allgemein -> IP-Adresse und Port eingeben vom Slave auf der 2. Steuerung**

### Kanal hinzufügen

Damit die beide Steuerungen kommunizieren können muss ein Kanal hinzugefügt werden. Dafür am Slave auf **Modbus- Slave-Kanal -> Kanal hinzufügen...** und folgende Parameter eingeben, sodass man 2 beschreibbare Ein- und Ausgänge hat:



The screenshot shows a 'ModbusChannel' configuration window. It has a blue title bar with a close button. The window is divided into three main sections: 'Kanal', 'READ Register', and 'WRITE Register'. In the 'Kanal' section, 'Name' is 'Channel 1', 'Zugriffstyp' is 'Read Holding Registers (Funktionscode 3)', 'Trigger' is 'Zyklisch' with a 'Zykluszeit (ms)' of '100', and there is an empty 'Kommentar' field. The 'READ Register' section has 'Offset' as '0x0000', 'Länge' as '1', and 'Fehlerbehandlung' as 'Letzen Wert beibehalten'. The 'WRITE Register' section has 'Offset' as '0x0000' and 'Länge' as '1'. At the bottom right are 'OK' and 'Abbrechen' buttons.

Kanal	
Name	Channel 1
Zugriffstyp	Read Holding Registers (Funktionscode 3)
Trigger	Zyklisch
Zykluszeit (ms)	100
Kommentar	

READ Register	
Offset	0x0000
Länge	1
Fehlerbehandlung	Letzen Wert beibehalten

WRITE Register	
Offset	0x0000
Länge	1

OK Abbrechen

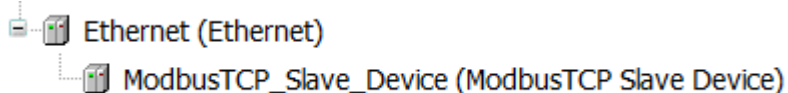
## Slave - Konfiguration

### Slave anhängen

Als erstes einen „Ethernet Knoten“ im Gerätebaum einfügen ⇒ **Device (rechtsklick)** ⇒ **Gerät anhängen...**

An diesen Ethernet Knoten wird dann ein Modbus-Slave angehängt ⇒ **Ethernet Kn. (rechtsklick)** ⇒ **Gerät anhängen...** ⇒ **Modbus** ⇒ **Modbus TCP Slave Device** ⇒ **Gerät anhängen** ⇒ **Schließen**.

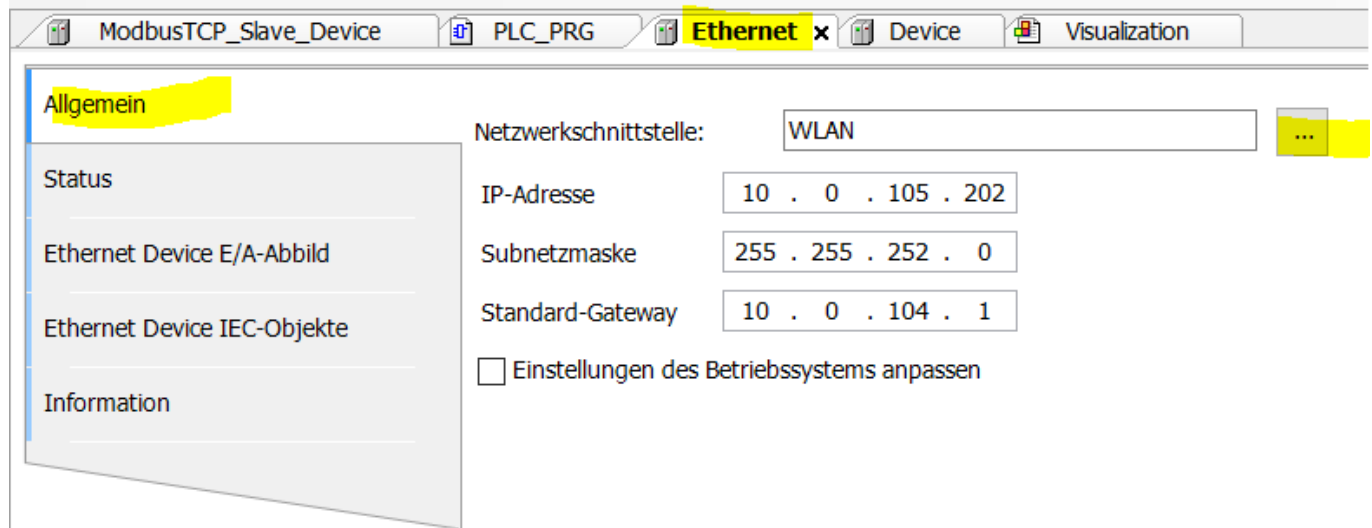
Danach sollte es so aussehen:



### Ethernet - Verbindung

Jetzt muss auch hier die **TCP Verbindung konfiguriert** werden. Dazu muss in den Einstellungen vom Ethernetknoten, die richtige Netzwerkschnittstelle ausgewählt werden. In diesem Fall haben wir uns entschieden über die WLAN-Schnittstelle zu arbeiten.

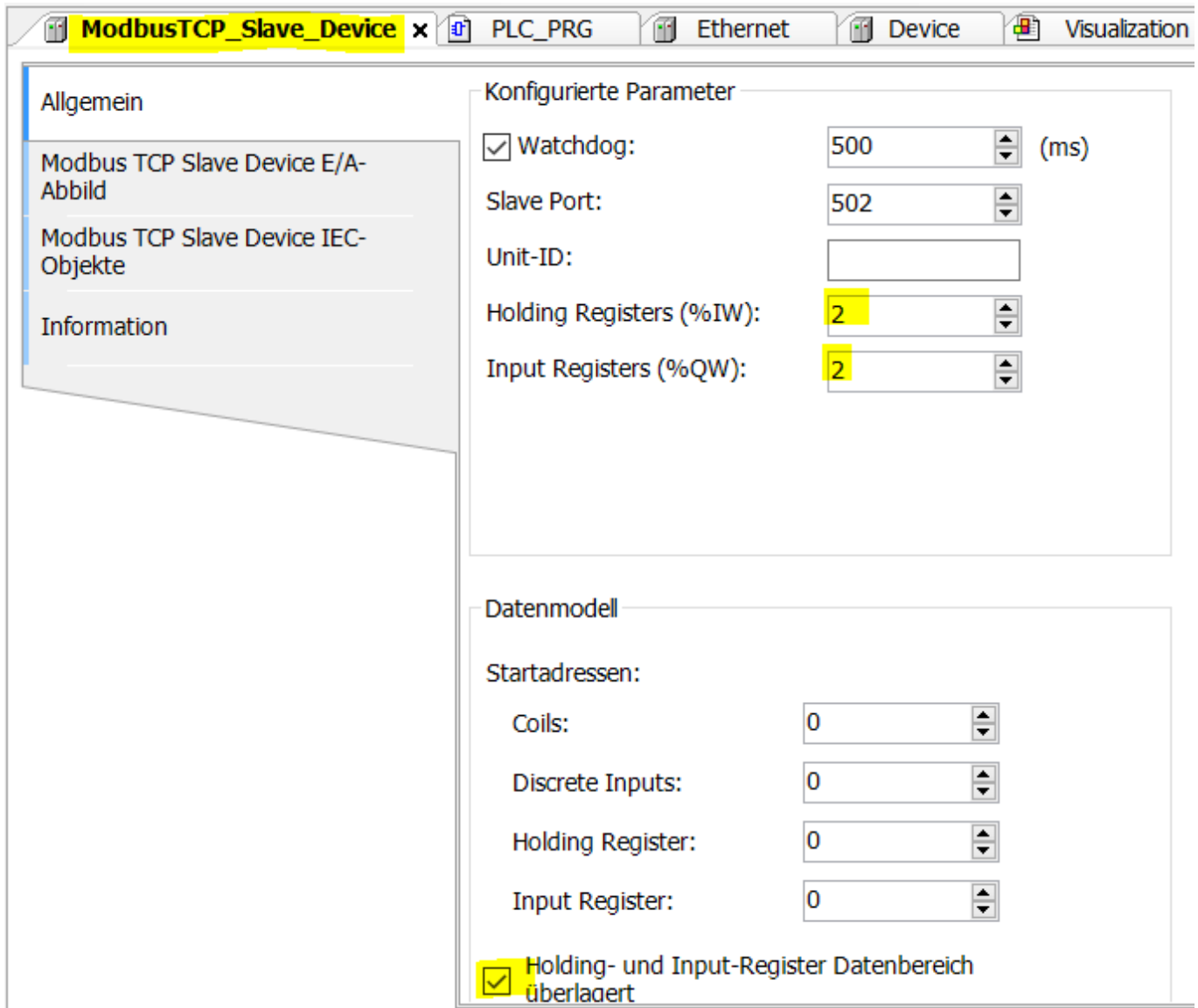
Um die auszuwählen klickt man einfach auf die drei Punkte.



**!!! ACHTUNG !!!** Damit man hier was finden kann muss **Codesys Controll WIN V3 - x64 Systray** laufen.  
(Wenn nicht läuft dann siehe *Starten*)

## Slave - Einstellungen

Beim Slave muss man eigentlich nicht viel einstellen. Das einzige was man machen muss ist, dass man die **Anzahl an Input und Holding Register** angibt. Diese Anzahl sollte **mit der am Master übereinstimmen**. Weiters wählen wir auch den Haken an, dass diese Register überlagert werden sollen. (*Damit schaffen wir einen gemeinsamen Speicherbereich*)



## E/A-Abbild Einstellungen

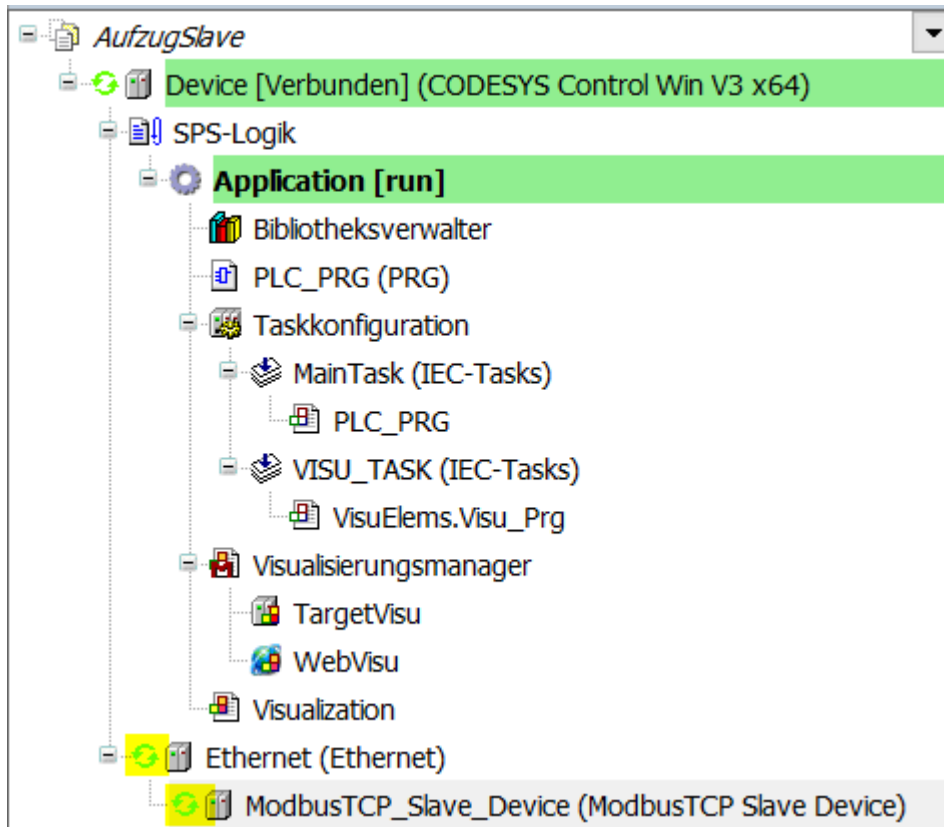
Hier muss noch die Buszykluszeit und bei den Variablen etwas eingestellt werden. Siehe Bild  
![slaveBuszyklus.PNG]

## Starten

Bevor man das Programm starten kann muss man sicherstellen, dass **Codesys Controll WIN V3 - x64 Systray** läuft (Rechts unten bei diesen kleinen Symbolen, Rechtsklick auf das Programm ⇒ Start PLC).

Dann in Codesys ⇒ **Einloggen** ⇒ **Starten**

Jetzt sollte beim Slave und beim Master folgendes zu sehen sein:



## Eingänge -> Ausgänge

Bei der Kommunikation zwischen Master und Slave wird ein Register als Speicherplatz für gemeinsam genutzte Variablen verwendet. Die Eingänge am Slave sind die Ausgänge am Master (und umgekehrt).

Die Eingänge sind im **Holding-Register** und die Ausgänge im **Input-Register**

Device

ModbusTCP\_Slave\_Device x

PLC\_PRG

Ethernet

Visualization

Allgemein

Modbus TCP Slave Device E/A-Abbild

Modbus TCP Slave Device IEC-Objekte









Information

Suchen

Filter

Alle anzeigen

FB für E/A-Kanal hinzufügen...

Variable	Mapping	Kanal	Adresse	Typ	Einheit	Beschreibung
		Eingänge	%IW0	ARRAY [0..1] OF WORD		Modbus Holding Registers
		Eingänge[0]	%IW0	WORD		
		Eingänge[1]	%IW1	WORD		
		Ausgänge	%QW0	ARRAY [0..1] OF WORD		Modbus Input Registers
		Ausgänge[0]	%QW0	WORD		
		Ausgänge[1]	%QW1	WORD		
	Application.PLC_...	Bit0	%QX2.0	BOOL		
		Bit1	%QX2.1	BOOL		

Diesen Eingängen/Ausgängen kann man auch **Variablen aus den Programmen zuweisen**. So kann man einen Schalter als Ausgangsvariable deklarieren.

Der andere definiert eine Variable als Eingang. Diese Variable weist er einer Lampe zu -> Wenn jetzt der Schalter auf der einen Maschine aktiviert wird, leuchtet die Lampe auf der anderen Maschine.

## Überprüfung

Zur Überprüfung wurden auf dem Master und dem Slave jeweils eine Visualisierung, bestehend aus einer Lampe und einem Schalter, erstellt. Diesen Komponenten wurden die Variablen aus dem Register zugewiesen. Wenn nun bei dem einem Gerät der Schalter aktiviert wird, so leuchtet die Lampe am anderen Gerät - und umgekehrt.

## Quelle

<https://www.codesys-blog.com/kommunikation/modbus-kommunikation-in-codesys/>