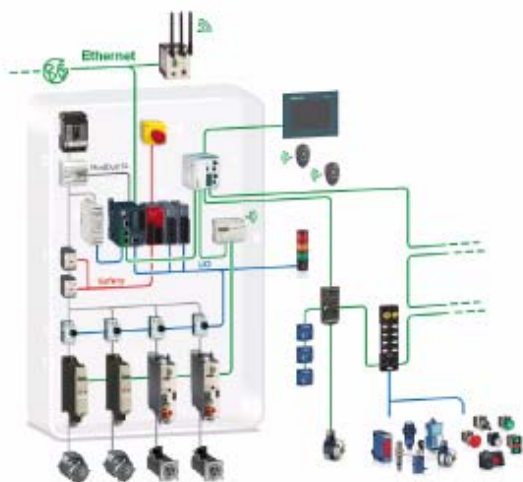


# SoMachine Modbus TCP IOScanner Benutzerhandbuch

09/2014



---

Die Informationen in der vorliegenden Dokumentation enthalten allgemeine Beschreibungen und/oder technische Leistungsmerkmale der hier erwähnten Produkte. Diese Dokumentation dient keinesfalls als Ersatz für die Ermittlung der Eignung oder Verlässlichkeit dieser Produkte für bestimmte Verwendungsbereiche des Benutzers und darf nicht zu diesem Zweck verwendet werden. Jeder Benutzer oder Integrator ist verpflichtet, angemessene und vollständige Risikoanalysen, Bewertungen und Tests der Produkte im Hinblick auf deren jeweils spezifischen Verwendungszweck vorzunehmen. Weder Schneider Electric noch deren Tochtergesellschaften oder verbundene Unternehmen sind für einen Missbrauch der Informationen in der vorliegenden Dokumentation verantwortlich oder können diesbezüglich haftbar gemacht werden. Verbesserungs- und Änderungsvorschläge sowie Hinweise auf angetroffene Fehler werden jederzeit gern entgegengenommen.

Dieses Dokument darf ohne entsprechende vorhergehende, ausdrückliche und schriftliche Genehmigung durch Schneider Electric weder in Teilen noch als Ganzes in keiner Form und auf keine Weise, weder anhand elektronischer noch mechanischer Hilfsmittel, reproduziert oder fotokopiert werden.

Bei der Montage und Verwendung dieses Produkts sind alle zutreffenden staatlichen, landesspezifischen, regionalen und lokalen Sicherheitsbestimmungen zu beachten. Aus Sicherheitsgründen und um die Übereinstimmung mit dokumentierten Systemdaten besser zu gewährleisten, sollten Reparaturen an Komponenten nur vom Hersteller vorgenommen werden.

Beim Einsatz von Geräten für Anwendungen mit technischen Sicherheitsanforderungen sind die relevanten Anweisungen zu beachten.

Die Verwendung anderer Software als der Schneider Electric-eigenen bzw. einer von Schneider Electric genehmigten Software in Verbindung mit den Hardwareprodukten von Schneider Electric kann Körperverletzung, Schäden oder einen fehlerhaften Betrieb zur Folge haben.

Die Nichtbeachtung dieser Informationen kann Verletzungen oder Materialschäden zur Folge haben!

© 2014 Schneider Electric. Alle Rechte vorbehalten.



	<b>Sicherheitshinweise</b> .....	<b>5</b>
	<b>Über dieses Buch</b> .....	<b>7</b>
<b>Kapitel 1</b>	<b>Modbus TCP IOScanner – Beschreibung</b> .....	<b>9</b>
	Modbus TCP IOScanner – Überblick .....	<b>10</b>
	Architektur .....	<b>12</b>
	Prinzipien .....	<b>14</b>
<b>Kapitel 2</b>	<b>Netzwerkinstallation</b> .....	<b>17</b>
	Netzwerkplanung .....	<b>18</b>
	IP-Adressen-Konfiguration .....	<b>20</b>
	Netzwerktests .....	<b>23</b>
<b>Kapitel 3</b>	<b>Modbus TCP IOScanner – Konfiguration</b> .....	<b>25</b>
	Hinzufügen eines Slaves zum Modbus TCP-E/A-Scanner .....	<b>26</b>
	Konfiguration eines Modbus TCP IOScanner .....	<b>27</b>
	Konfigurieren eines verteilten Advantys OTB-E/A-Moduls für den Modbus TCP-E/A-Scanner .....	<b>29</b>
	Konfigurieren eines vordefinierten Slave-Geräts für den Modbus TCP- E/A-Scanner .....	<b>32</b>
	Konfiguration eines generischen Geräts für den Modbus TCP-E/A- Scanner .....	<b>34</b>
<b>Kapitel 4</b>	<b>Modbus TCP IOScanner – Betrieb</b> .....	<b>37</b>
	Modbus TCP IOScanner-Ressourcenprüfung .....	<b>38</b>
	Modbus TCP IOScanner-Betriebsmodi .....	<b>40</b>
	Anwendungsschnittstelle .....	<b>44</b>
<b>Kapitel 5</b>	<b>Modbus TCP IOScanner – Wartung</b> .....	<b>47</b>
	Diagnose: SoMachine-Onlinemodus .....	<b>48</b>
	Diagnose: Webserver .....	<b>52</b>
	Fehlerbehebung .....	<b>54</b>
<b>Anhang</b>	.....	<b>55</b>
<b>Anhang A</b>	<b>Modbus TCP IOScanner-Funktionen</b> .....	<b>57</b>
	IOS_GETSTATE: Status des Modbus TCP IOScanner lesen .....	<b>58</b>
	IOS_START: Modbus TCP IOScanner starten .....	<b>59</b>
	IOS_GETHEALTH: Wert des Funktionsfähigkeitsbits lesen .....	<b>60</b>
	IOS_STOP: Modbus TCP IOScanner anhalten .....	<b>61</b>
	CONFIGURE_OTB: Softwarekonfiguration des Advantys OTB senden .....	<b>62</b>

---

<b>Anhang B</b>	<b>Modbus TCP IOScanner – Datentypen</b> .....	<b>65</b>
	losStateCodes: Modbus TCP IOScanner-Statuswerte .....	<b>66</b>
	CommunicationErrorCodes: Fehlercodes .....	<b>67</b>
	configurationOTBErrorCodes: Codes für Fehler in der OTB-Konfiguration .....	<b>68</b>
<b>Anhang C</b>	<b>Darstellung von Funktionen und Funktionsbausteinen</b> .....	<b>69</b>
	Unterschiede zwischen einer Funktion und einem Funktionsbaustein	<b>70</b>
	Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache AWL .....	<b>71</b>
	Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache ST .....	<b>75</b>
<b>Glossar</b>	.....	<b>79</b>
<b>Index</b>	.....	<b>83</b>



## Wichtige Informationen

### HINWEISE

Lesen Sie diese Anweisungen sorgfältig durch und machen Sie sich vor Installation, Betrieb und Wartung mit dem Gerät vertraut. Die nachstehend aufgeführten Warnhinweise sind in der gesamten Dokumentation sowie auf dem Gerät selbst zu finden und weisen auf potenzielle Risiken und Gefahren oder bestimmte Informationen hin, die eine Vorgehensweise verdeutlichen oder vereinfachen.



Wird dieses Symbol zusätzlich zu einem Sicherheitshinweis des Typs „Gefahr“ oder „Warnung“ angezeigt, bedeutet das, dass die Gefahr eines elektrischen Schlags besteht und die Nichtbeachtung der Anweisungen unweigerlich Verletzung zur Folge hat.



Dies ist ein allgemeines Warnsymbol. Es macht Sie auf mögliche Verletzungsgefahren aufmerksam. Beachten Sie alle unter diesem Symbol aufgeführten Hinweise, um Verletzungen oder Unfälle mit Todesfälle zu vermeiden.

### **GEFAHR**

**GEFAHR** macht auf eine gefährliche Situation aufmerksam, die, wenn sie nicht vermieden wird, einen schweren oder tödlichen Unfall **zur Folge hat**.

### **WARNUNG**

**WARNUNG** verweist auf eine Gefahr, die, wenn sie nicht vermieden wird, Tod oder Verletzungen **zur Folge haben kann**.

### **VORSICHT**

**VORSICHT** verweist auf eine Gefahr, die, wenn sie nicht vermieden wird, leichte Verletzungen **zur Folge haben kann**.

### **HINWEIS**

**HINWEIS** gibt Auskunft über Vorgehensweisen, bei denen keine Verletzungen drohen.

---

## **BITTE BEACHTEN**

Elektrische Geräte dürfen nur von Fachpersonal installiert, betrieben, bedient und gewartet werden. Schneider Electric haftet nicht für Schäden, die durch die Verwendung dieses Materials entstehen.

Als qualifiziertes Personal gelten Mitarbeiter, die über Fähigkeiten und Kenntnisse hinsichtlich der Konstruktion und des Betriebs dieser elektrischen Geräte und der Installationen verfügen und eine Schulung zur Erkennung und Vermeidung möglicher Gefahren absolviert haben.

# Über dieses Buch



## Auf einen Blick

### Ziel dieses Dokuments

Dieses Dokument unterstützt Sie bei folgenden Aufgaben:

- Planen des Modbus TCP IOScanner-Netzwerks.
- Installation und Konfiguration des Modbus TCP IOScanner-Netzwerks.
- Betrieb und Wartung des Modbus TCP IOScanner-Netzwerks.

**HINWEIS:** Machen Sie sich mit diesem Dokument und allen verwandten Dokumenten vertraut, bevor Sie Ihre Steuerung installieren, betreiben oder warten.

### Gültigkeitsbereich

Diese Dokumentation wurde für die Herausgabe von SoMachine V4.1 Modbus-TCP-E/A-Scanner Add-On aktualisiert.

Die technischen Merkmale der hier beschriebenen Geräte sind auch online abrufbar.

Die in diesem Handbuch vorgestellten Merkmale sollten denen entsprechen, die online angezeigt werden. Im Rahmen unserer Bemühungen um eine ständige Verbesserung werden Inhalte im Laufe der Zeit möglicherweise überarbeitet, um deren Verständlichkeit und Genauigkeit zu verbessern. Sollten Sie einen Unterschied zwischen den Informationen im Handbuch und denen online feststellen, nutzen Sie die Online-Informationen als Referenz.

### Weiterführende Dokumentation

Titel der Dokumentation	Referenz-Nummer
Modicon M251 Logic Controller – Programmierhandbuch	EIO0000001462 (ENG), EIO0000001463 (FRE), EIO0000001464 (GER), EIO0000001465 (SPA), EIO0000001466 (ITA), EIO0000001467 (CHS)
SoMachine – Programmierhandbuch	EIO0000000067 (ENG); EIO0000000069 (FRE); EIO0000000068 (GER); EIO0000000071 (SPA); EIO0000000070 (ITA); EIO0000000072 (CHS)
Grundlagenhandbuch: Netzwerke, Konnektivität und Webserver	DIA6ED2130205EN (ENG)

Diese technischen Veröffentlichungen sowie andere technische Informationen stehen auf unserer Website [www.schneider-electric.com](http://www.schneider-electric.com) zum Download bereit.

## Produktbezogene Informationen

### **WARNUNG**

#### **STEUERUNGS AUSFALL**

- Bei der Konzeption von Steuerungsstrategien müssen mögliche Störungen auf den Steuerungspfaden berücksichtigt werden, und bei bestimmten kritischen Steuerungsfunktionen ist dafür zu sorgen, dass während und nach einem Pfadfehler ein sicherer Zustand erreicht wird. Beispiele kritischer Steuerungsfunktionen sind die Notabschaltung (Not-Aus) und der Nachlauf-Stopp, Stromausfall und Neustart.
- Für kritische Steuerungsfunktionen müssen separate oder redundante Steuerpfade bereitgestellt werden.
- Systemsteuerpfade können Kommunikationsverbindungen umfassen. Dabei müssen die Auswirkungen unerwarteter Sendeverzögerungen und Verbindungsstörungen berücksichtigt werden.
- Sämtliche Unfallverhütungsvorschriften und lokale Sicherheitsrichtlinien sind zu beachten.<sup>1</sup>
- Jede Implementierung des Geräts muss individuell und sorgfältig auf einen einwandfreien Betrieb geprüft werden, bevor das Gerät an Ort und Stelle in Betrieb gesetzt wird.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

<sup>1</sup> Weitere Informationen finden Sie in den aktuellen Versionen von NEMA ICS 1.1 „Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control“ sowie von NEMA ICS 7.1, „Safety Standards for Construction and Guide for Selection, Installation, and Operation of Adjustable-Speed Drive Systems“ oder den entsprechenden, vor Ort geltenden Vorschriften.

### **WARNUNG**

#### **UNBEABSICHTIGTER BETRIEBSZUSTAND DES GERÄTS**

- Verwenden Sie mit diesem Gerät nur von Schneider Electric genehmigte Software.
- Aktualisieren Sie Ihr Anwendungsprogramm jedes Mal, wenn Sie die physikalische Hardwarekonfiguration ändern.

**Die Nichtbeachtung dieser Anweisungen kann Tod, schwere Verletzungen oder Sachschäden zur Folge haben.**

---

# Kapitel 1

## Modbus TCP IOScanner – Beschreibung

---

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Modbus TCP IOScanner – Überblick	10
Architektur	12
Prinzipien	14

## Modbus TCP IOScanner – Überblick

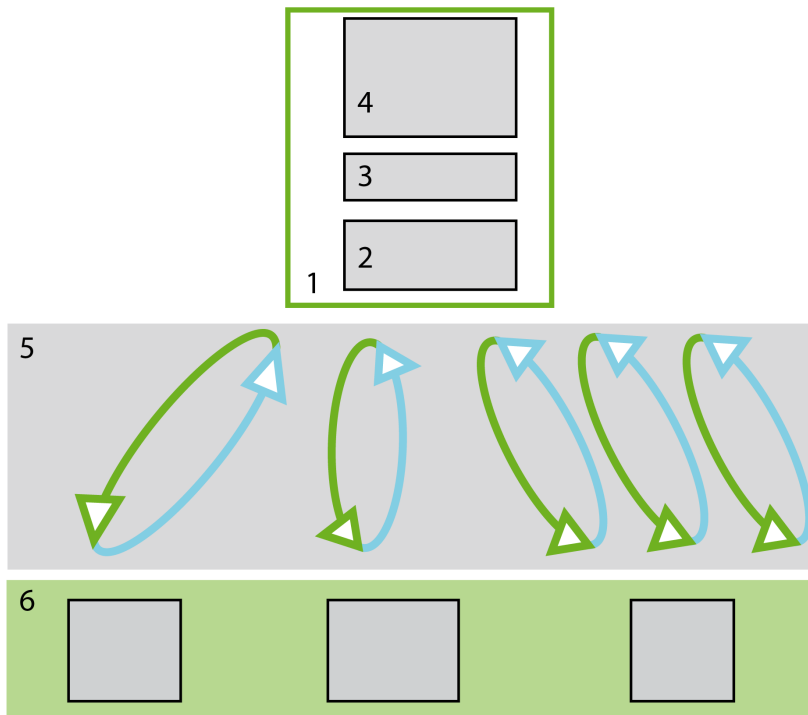
### Beschreibung

Der Modbus TCP IOScanner ist ein auf Ethernet basierender Dienst, der kontinuierlich Slave-Geräte anfragt, um Daten sowie Status- und Diagnoseinformationen auszutauschen. Bei diesem Vorgang werden bei den Slave-Geräten die Eingänge überwacht und die Ausgänge gesteuert.

Der Modbus TCP IOScanner basiert auf dem Modbus-TCP-Standard. Der Kern dieses Standards ist ein Master/Slave-Netzwerkmodell. Der eindeutige Master ist die Steuerung.

Die Kommunikation mit den Slaves erfolgt über Modbus-TCP-Kanäle (*siehe Seite 14*).

### Prinzip



- 1 Steuerung
- 2 E/A-Images
- 3 Anwendungsschnittstelle (*siehe Seite 44*)
- 4 Anwendung
- 5 Modbus-Kanäle (*siehe Seite 14*)
- 6 Slave-Geräte (*siehe Seite 14*)

### **Architektur des Systems**

Der Modbus TCP IOScanner basiert auf Folgendem:

- Ethernet-Netzwerk mit der Steuerung, den Slaves und der Infrastrukturtechnik (*siehe Seite 12*)
- Softwarekonfiguration (*siehe Seite 13*)

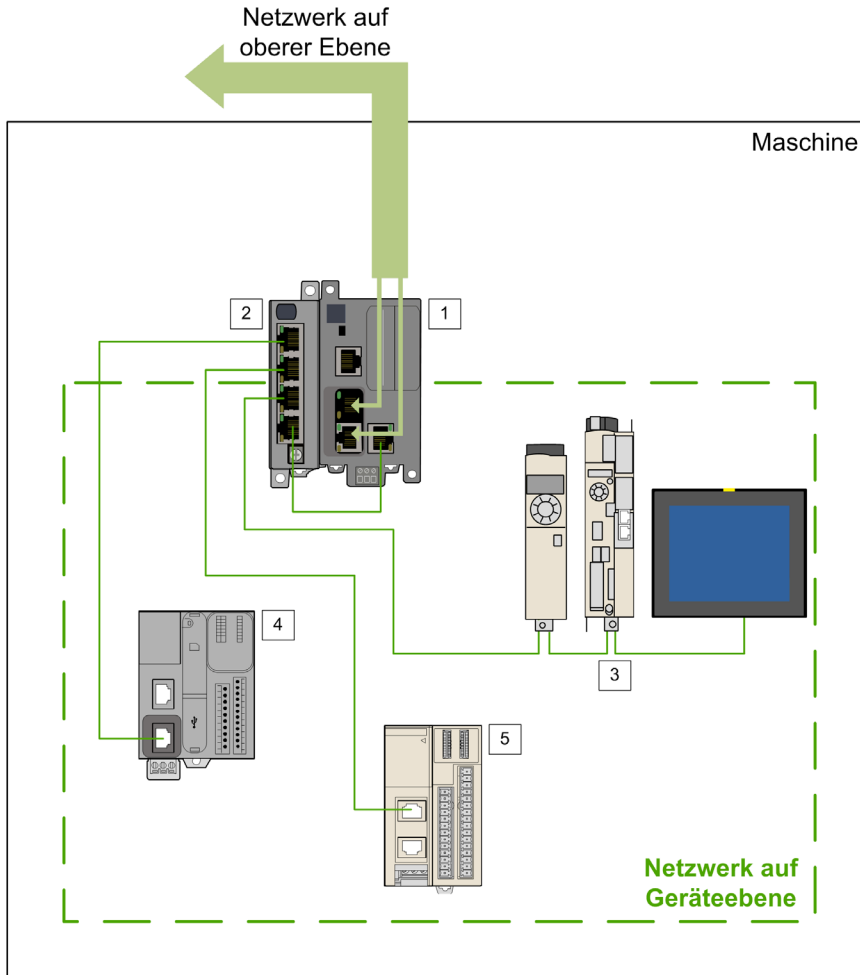
### **Kompatibilität der Steuerung**

Der Modbus TCP IOScanner-Dienst steht auf der TM251MESE-Steuerung zur Verfügung.

## Architektur

### Ethernet-Netzwerk

In dieser Abbildung ist eine typische Modbus TCP IOScanner-Architektur dargestellt.



- 1 Steuerung/Modbus-Master
- 2 TM4ES4 als eigenständiger Ethernet-Switch verwendet
- 3 Slaves verkettet (Daisy Chain)
- 4 Modbus-Slave
- 5 E/A-Insel

Die Steuerung ist sowohl mit dem Netzwerk der oberen Ebene als auch mit dem Netzwerk auf Geräteebene verbunden.

Das Netzwerk auf Geräteebene wird vom Modbus TCP IOScanner gesteuert.

Die Steuerung kann als Gateway (*siehe Seite 20*) zwischen den zwei Netzwerken verwendet werden.

### Softwarekonfiguration

Der Netzwerkbereich der Slaves und der einzelnen E/A wird softwareseitig konfiguriert:

- Die Kommunikationskonfiguration definiert die Adressierungs- und Kommunikationszeiträume.
- Die Gerätekonfiguration definiert das Geräteverhalten.

Die Slave-Konfiguration ermöglicht eine Variablenzuordnung zum Optimieren der Überwachung.

Über die E/A-Konfiguration wird die Überwachungsqualität an die Bandbreite des Netzwerks angepasst.

Sie können mehrere Parameter (*siehe Seite 15*) festlegen, um die Leistung zu optimieren.

### Optionale Dienste

Der Modbus TCP IOScanner kann mehreren optionalen Diensten zugeordnet werden:

- DHCP: Der DHCP-Server (Dynamic Host Configuration Protocol) weist einem Slave-Gerät eine IP-Adresse zu, wenn es eine anfordert.
- FDR: Der FDR-Server (Fast Device Replacement) konfiguriert ein ausgetauschtes Remote-Gerät, das vom Modbus TCP IOScanner gesteuert wird, ohne die Anwendung anzuhalten.
- Webserver (*siehe Seite 52*).

## Prinzipien

### Überblick

Der Modbus TCP IOScanner liest Eingänge und schreibt Ausgänge der Slave-Geräte.

Die Kommunikation zwischen Modbus TCP IOScanner und den Slave-Geräten erfolgt über Modbus-Kanäle.

Die Kommunikation im Modbus TCP IOScanner wird über die SoMachine-Software konfiguriert.

### Slave-Typen

Es gibt drei verschiedene Typen von Modbus TCP IOScanner-Slaves in der SoMachine-Software:

- **Advantys-OTB-Slave-Geräte** werden für digitale und analoge Remote-E/As verwendet. Verwenden Sie die SoMachine-Software für die spezifische Konfiguration des Geräts und der zugeordneten E/A-Module.
- **Vordefinierte Slave-Geräte** sind normale Modbus-Geräte mit verschiedenen vordefinierten Kommunikationsparametern. Verwenden Sie eine spezielle Software und/oder eine lokale HMI, um die Geräte zu konfigurieren. Mit der FDT/DTM-Technologie können in SoMachine vordefinierte Slave-Geräte mit erweiterten Einstellungen konfiguriert werden. Weitere Informationen hierzu finden Sie im Device Type Manager (DTM) - Benutzerhandbuch.
- **Generische Slave-Geräte** werden für alle anderen Modbus-Slave-Geräte verwendet. Die gesamte Gerätekonfiguration erfolgt mit einer Fremdanbietersoftware und/oder einer lokalen HMI. Mit der FDT/DTM-Technologie können in SoMachine vordefinierte einige Geräte konfiguriert werden. Weitere Informationen hierzu finden Sie im Device Type Manager (DTM) - Benutzerhandbuch.

### Modbus-Kanal

Ein Modbus-Kanal überträgt einen Modbus-Request zwischen dem Master und einem Slave.

Advantys OTB und vordefinierte Slave-Geräte verwenden einen Kanal pro Gerät. Dieser Kanal wird mit der SoMachine-Software konfiguriert.

Für ein generisches Slave-Gerät können Sie mehrere Kanäle verwenden. Um an ein Gerät mehrere Requests zu senden, erstellen Sie mehrere Kanäle.

## Konfigurationsparameter für die Kommunikation

Konfigurieren Sie jeden Slave im Modbus TCP IOScanner-Netzwerk mithilfe der SoMachine-Software.

In dieser Tabelle werden die Konfigurationsparameter für die Kommunikation aufgeführt:

Parameter	Beschreibung
IP-Adresse	Die IP-Adresse des Slaves im Ethernet-Netzwerk.
Timeout für Funktionsfähigkeit	Zeitwert in ms. Wenn der Modbus TCP IOScanner nach dieser Frist noch keine Antwort vom Slave erkannt hat, tritt ein Fehlerzustand ein.
Wiederholungsrate	Zeitwert in ms. Dies ist der zeitliche Abstand zwischen zwei Sendevorgängen eines Requests. Dieser Wert muss niedriger sein als das Timeout für Funktionsfähigkeit.
Kanal-ID	Eindeutige Kennung eines Kanals. Dieser Wert wird von der SoMachine-Software automatisch erstellt, wenn ein neuer Kanal hinzugefügt wird. Sie können diesen Wert auf der Registerkarte <b>Modbus TCP Kanalkonfiguration</b> bzw. beim Advantys-OTB-Slave auf der Registerkarte <b>Modbus TCP Slave-Konfiguration</b> einsehen.



---

# Kapitel 2

## Netzwerkinstallation

---

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Netzwerkplanung	18
IP-Adressen-Konfiguration	20
Netzwerktests	23

## Netzwerkplanung

### Zweck

Ein geplantes Netzwerk steigert die Installationseffizienz und senkt die Installationsdauer und -kosten. Die Schnittstellen zwischen den Materialien (Switches, Kabel, Ports) müssen konzipiert werden, bevor das Netzwerk geplant werden kann.

### Netzwerkdesign

Informationen zum Gestalten und Planen des Modbus TCP IOScanner-Netzwerks finden Sie in der entsprechenden Dokumentation, z. B. im Media Planning and Installation Manual von der ODVA. Sie können dieses Handbuch von der [ODVA-Website](#) herunterladen.

### Switcharten

Verwenden Sie die Switchart, die den spezifischen Anforderungen Ihres Netzwerks entspricht:

Anforderung	Erforderliche Switches
Netzwerkdiagnose- und Betriebsinformationen	Verwaltbare Switches
Kommunikationsverfügbarkeit bei physischem Verbindungsverlust	Redundante Switches
Netzwerk mit geringer Reichweite (Glasfaser)	Switch mit Duplex-SC-Anschluss

**HINWEIS:** Verwenden Sie keinen Hub, um ein Modbus TCP IOScanner-Netzwerk einzurichten. Weitere Informationen zu Switches finden Sie im **Grundlagenhandbuch: Netzwerke, Konnektivität und Webserver**.

## Kabelarten

In dieser Tabelle sind Kabelmodelle aufgeführt, die im Netzwerk verwendet werden können.

Weitere Informationen zu Kabeln finden Sie im **Grundlagenhandbuch: Netzwerke, Konnektivität und Webserver**.

In einer Standardinstallation können Sie die folgenden Kabel verwenden:

Referenz	Beschreibung	Details	Länge
490NTW000**	Geschirmtes Ethernet-Kabel für DTE-Verbindungen	Standardkabel, an beiden Enden mit RJ45-Steckanschlüssen für DTE ausgestattet CE-konform	2, 5, 12, 40 oder 80 m (6.56, 16.4, 39.37, 131.23 oder 262.47 ft)
490NTW000**U		Standardkabel, an beiden Enden mit RJ45-Steckanschlüssen für DTE ausgestattet UL-konform	2, 5, 12, 40 oder 80 m (6.56, 16.4, 39.37, 131.23 oder 262.47 ft)
TCSECE3M3M**S4		Kabel für raue Umgebungen, an beiden Enden mit RJ45-Steckanschlüssen ausgestattet CE-konform	1, 2, 3, 5 oder 10 m (3.28, 6.56, 9.84, 16.4, 32.81 ft)
TCSECU3M3M**S4		Kabel für raue Umgebungen, an beiden Enden mit RJ45-Steckanschlüssen ausgestattet UL-konform	1, 2, 3, 5 oder 10 m (3.28, 6.56, 9.84, 16.4, 32.81 ft)

In Glasfasernetzwerken können Sie die folgenden Kabel verwenden:

Referenz	Beschreibung	Details	Länge
490NOC00005	Glasfaserkabel für DTE-Verbindungen	1 SC-Stecker 1 MT-RJ-Stecker	5 m (16,4 ft)
490NOT00005		1 ST-Stecker (BFOC) 1 MT-RJ-Stecker	5 m (16,4 ft)
490NOR00003		2 MT-RJ-Stecker	3 m (9,8 ft)
490NOR00005		2 MT-RJ-Stecker	5 m (16,4 ft)

## IP-Adressen-Konfiguration

### Voraussetzungen

Jedes Gerät im Modbus TCP IOScanner-Netzwerk erhält eine eigene IP-Adresse. Alle IP-Adressen müssen eindeutig sein.

**HINWEIS:** Weisen Sie für ein Modbus TCP IOScanner-Netzwerk IP-Adressen der Klasse C zu. Die IP-Adressen werden in zwei Schritten konfiguriert:

Schritt	Beschreibung
1	<p>Konfigurieren Sie mithilfe der SoMachine-Software den Steuerungsport für den Modbus TCP IOScanner:</p> <ul style="list-style-type: none"> <li>● IP-Adresse,</li> <li>● Subnetzmaske,</li> <li>● Standard-Gateway.</li> </ul> <p><b>HINWEIS:</b> Verwenden Sie nur feste Adressen.</p> <p><b>HINWEIS:</b> Verwenden Sie nicht die Post-Konfigurationsdatei, um die Steuerungsports zu adressieren.</p>
2	<p>Konfigurieren Sie alle Slave-Geräte im Modbus TCP IOScanner-Netzwerk:</p> <ul style="list-style-type: none"> <li>● IP-Adresse,</li> <li>● Subnetzmaske,</li> <li>● Standard-Gateway.</li> </ul>

### Slave-Adressenzuweisung

Weisen Sie im Modbus TCP IOScanner-Netzwerk die IP-Adressen der Slaves mit der folgenden Methode zu (je nach Gerätetyp):

- Mit dem DHCP-Server zum Verwalten aller IP-Adressen des Modbus TCP IOScanner-Netzwerks über die SoMachine-Software, oder wenn Sie den FDR-Dienst benötigen
- Mit einer Fremdanbietersoftware oder einer lokalen HMI
- Über die Konfiguration erweiterter Einstellungen auf der seriellen Modbus-Leitung mit der FDT/DTM-Technologie; siehe Device Type Manager (DTM) - Benutzerhandbuch

### FDR-Dienst

Einige Slaves unterstützen den FDR-Dienst (Fast Device Replacement).

Der FDR-Dienst speichert Netzwerk- und Betriebsparameter von Geräten im Netzwerk. Wenn ein Gerät ausgetauscht wird, konfiguriert der Dienst automatisch das Ersatzgerät mit den gleichen Parametern wie das entfernte Gerät.

Informationen zum Konfigurieren dieses Dienstes im Slave finden Sie in der Dokumentation zum Slave-Gerät.

## Parameter für Master-IP-Adresse von Slaves

Einige Slaves verfügen über den Parameter **Master-IP-Adresse**, damit nur eine deklarierte Mastersteuerung auf das Slave-Gerät zugreifen kann.

Gerätekonfiguration	Folge
Ist zur Verwendung des Modbus TCP IOScanner konfiguriert	Konfigurieren Sie den Parameter <b>Master-IP-Adresse</b> innerhalb des Geräts (siehe unten).
Ist nicht zur Verwendung des Modbus TCP IOScanner konfiguriert	Verwenden Sie 0.0.0.0 für den Parameter <b>Master-IP-Adresse</b> im Gerät.

Der Parameter **Master-IP-Adresse** des Slaves muss auf die IP-Adresse der Steuerung gesetzt sein, die den Modbus TCP IOScanner unterstützt (**Ethernet 2-Port**).

Informationen zum Konfigurieren dieses Parameters im Slave finden Sie in der Dokumentation zum Slave.

## Gateway-Parameter der Slaves

Der Gateway-Parameter der Slaves muss auf die IP-Adresse des Ethernet-Ports der Steuerung gesetzt sein, die den Modbus TCP IOScanner unterstützt (**Ethernet 2**).

Ein Konfigurationstool muss die Slaves erreichen können, um deren Parameter festzulegen.

Eigenschaft des Konfigurationstools	Folge
Ist mit dem Netzwerk der oberen Ebene verbunden	Aktualisieren Sie den Gateway-Parameter der Slaves (siehe unten).
Ist mit dem Netzwerk auf Geräteebene verbunden	Der Gateway-Parameter wird nicht verwendet.
Verwendet ein anderes Protokoll als TCP/IP	Der Gateway-Parameter wird nicht verwendet.

Informationen zum Konfigurieren dieses Parameters im Slave finden Sie in der Dokumentation zum Slave.

**HINWEIS:** Wenn der DHCP-Dienst verwendet wird, um die Slaves zu adressieren, wird der Gateway-Parameter in der DHCP-Tabelle der Steuerung festgelegt.

## PC-Routing

Der PC, der das Konfigurationstool unterstützt, muss so konfiguriert werden, dass er mit den Slaves kommunizieren kann.

Konfiguration des Slaves	Folge
Über erweiterte Einstellungen (FDT/DTM) als vordefinierter Slave konfiguriert	Es ist keine spezielle Parametrisierung des Computers erforderlich. <b>HINWEIS:</b> Die Computerkonfiguration wird nicht verändert.
Über ein anderes Tool konfiguriert	Aktualisieren Sie die Routingtabelle des PCs (siehe unten).

Um die Routingtabelle des PCs zu aktualisieren, stoppen Sie alle Verbindungen vom PC zur Steuerung und/oder anderen Geräten. Führen Sie anschließend in einer Windows-Eingabeaufforderung den Befehl `route ADD Destination MASK Subnet_Mask Gateway` aus.

In dieser Tabelle sind die in dieser Befehlssyntax zu aktualisierenden Parameter aufgeführt:

Parameter	Wert
Destination (Ziel)	IP-Adresse des Modbus TCP IOScanner-Netzwerks
Subnet_Mask (Subnetzmaske)	Subnetzmaske des Modbus TCP IOScanner-Netzwerks
Gateway	Adresse des Steuerungsports, der mit dem Netzwerk der oberen Ebene verbunden ist

Um diese Parameter in einer Windows-Eingabeaufforderung zu prüfen, führen Sie den Befehl `route PRINT` aus.

Um diese Route vom PC zu entfernen, führen Sie in einer Windows-Eingabeaufforderung den Befehl `route DELETE Destination` aus, wobei `Destination` für die zuvor eingegebene IP-Adresse des Modbus TCP IOScanner-Netzwerks steht.

## Netzwerktests

### Zweck

Testen Sie das Netzwerk, bevor Sie den Modbus TCP IOScanner in Betrieb nehmen.

Prüfen Sie Folgendes:

- Die Adressen aller Geräte wurden plangemäß konfiguriert.
- Jedes Gerät wurde ordnungsgemäß verdrahtet.

Im Folgenden finden Sie einige übliche Testmethoden:

### Status-LED

Vergewissern Sie sich, dass die Status-LEDs eine korrekte Verdrahtung anzeigen (abhängig von Ihren Geräten).

### Prüfung per Computer

Prüfen Sie mithilfe eines Computers, ob alle Slave-Geräte angeschlossen und adressiert sind:

Schritt	Aktion
1	Verbinden Sie den Computer mit dem Modbus TCP IOScanner-Netzwerk.
2	Rufen Sie die Eingabeaufforderung auf.
3	Verwenden Sie den Befehl <code>ping xxx.xxx.xxx.xxx</code> , um jeden Slave zu kontaktieren. xxx.xxx.xxx.xxx = IP-Adresse des zu testenden Slaves. <b>HINWEIS:</b> Der Befehl <code>ping -h</code> zeigt die Hilfe für den Befehl <code>ping</code> an.

### Prüfung per Webserver

Prüfen Sie mithilfe des Webserver der Steuerung, ob die Steuerung mit jedem Slave-Gerät kommunizieren kann:

Schritt	Aktion
1	Rufen Sie den Webserver der Steuerung auf.
2	Öffnen Sie die Seite <b>Ethernet-Diagnose</b> .
3	Verwenden Sie die Funktion ( <i>siehe Seite 52</i> ) <b>Remote-Ping</b> für jedes Slave-Gerät.



---

# Kapitel 3

## Modbus TCP IOScanner – Konfiguration

---

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Hinzufügen eines Slaves zum Modbus TCP-E/A-Scanner	26
Konfiguration eines Modbus TCP IOScanner	27
Konfigurieren eines verteilten Advantys OTB-E/A-Moduls für den Modbus TCP-E/A-Scanner	29
Konfigurieren eines vordefinierten Slave-Geräts für den Modbus TCP-E/A-Scanner	32
Konfiguration eines generischen Geräts für den Modbus TCP-E/A-Scanner	34

## Hinzufügen eines Slaves zum Modbus TCP-E/A-Scanner

### Überblick

In diesem Abschnitt wird beschrieben, wie dem **Modbus TCP-E/A-Scanner** ein Slave-Gerät hinzugefügt wird.

Die Slave-Geräte werden in 3 Kategorien untergliedert:

- Vordefinierte Geräte für Modbus-Geräte von Schneider Electric (ATV, LXM und ZBRN)
- Advantys OTB für Modbus TCP-Module mit konfigurierbaren E/A
- Generische Geräte für alle anderen Modbus TCP-Slaves

Für jedes generische Gerät müssen Sie durch Hinzufügen von Kanälen die an das Gerät zu sendenden Modbus-Requests definieren. Jeder Kanal entspricht einem Modbus-Request und kann über eine eigene Wiederholungsrate verfügen.

### Hinzufügen eines Slave-Geräts zum Modbus TCP-E/A-Scanner

Um einen Slave zum Modbus TCP-E/A-Scanner (**Modbus TCP IOScanner**) hinzuzufügen, wählen Sie das gewünschte Gerät im **Hardwarekatalog** aus, ziehen Sie es in die **Gerätebaumstruktur** und legen Sie es auf dem Knoten **Ethernet 2** in der **Gerätebaumstruktur** ab.

Weitere Informationen zum Hinzufügen von Slave-Geräten zu einem Projekt finden Sie unter:

- Verwenden der Methode Drag&Dop (*siehe SoMachine, Programmierhandbuch*) (Ziehen und Ablegen)
- Verwenden der Kontextmenüs oder Plus-Schaltflächen (*siehe SoMachine, Programmierhandbuch*)

## Konfiguration eines Modbus TCP IOScanner

### Voraussetzungen

Bevor Sie den Modbus TCP IOScanner konfigurieren:

- Setzen Sie die IP-Adresse von Ethernet 2 auf den **festen Modus**. Die Adresse darf nicht 0.0.0.0 lauten.
- Die verbundenen Geräte müssen sich im gleichen Subnetzwerk befinden wie der Ethernet 2-Port.

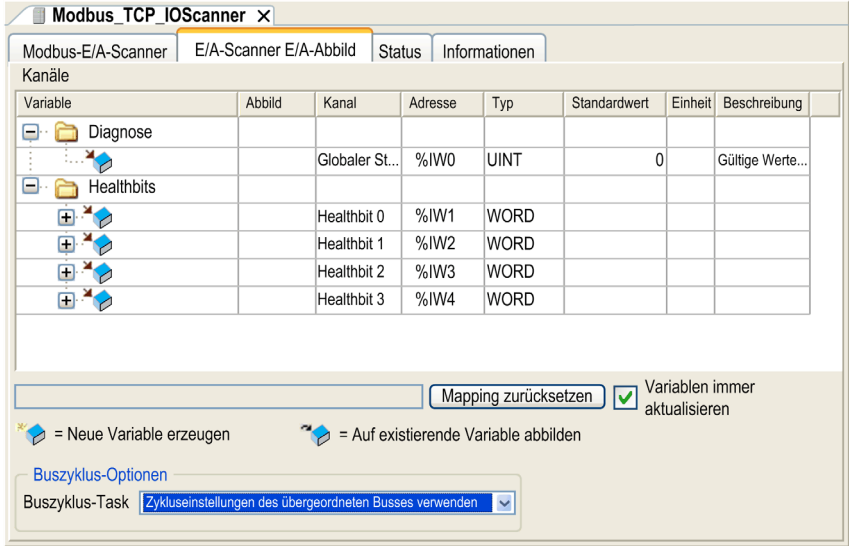
Weitere Informationen zur IP-Adresse finden Sie unter Ethernet-Konfiguration (*siehe Modicon M251 Logic Controller, Programmierhandbuch*).

### Modbus TCP IOScanner hinzufügen

Der Knoten Modbus TCP IOScanner wird automatisch hinzugefügt, sobald ein Slave im **Ethernet 2**-Knoten (*siehe Seite 26*) hinzugefügt wird.

### Modbus TCP IOScanner konfigurieren

Gehen Sie zur Konfiguration eines Elements Modbus TCP IOScanner vor wie folgt:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf <b>Modbus_TCP_IOScanner</b> . <b>Ergebnis:</b> Das Konfigurationsfenster wird angezeigt.
2	Wählen Sie die Registerkarte <b>E/A-Scanner E/A-Abbild</b> aus. 

Schritt	Aktion
3	<p>Wählen Sie in der Liste den Eintrag <b>Buszyklus-Task</b> aus:</p> <ul style="list-style-type: none"><li>● <b>Zykluseinstellungen des übergeordneten Busses verwenden</b> (Standardeinstellung)</li><li>● <b>MAST</b> oder:</li><li>● Ein vorhandener Task des Projekts.</li></ul> <p><b>HINWEIS:</b> Der Parameter <b>Buszyklus-Task</b> im E/A-Abbild-Editor des Geräts, das den Modbus TCP IOScanner enthält, definiert den für die Aktualisierung der E/A-Images (%QW, %IW) verantwortlichen Task. Diese E/A-Images entsprechen dem Modbus-Request, der an die Modbus-Slaves und die Funktionsfähigkeitsbits gesendet wird.</p>

**HINWEIS:** Wenn der Modbus TCP IOScanner konfiguriert ist, wird die Post-Konfigurationsdatei für das Ethernet-2-Netzwerk ignoriert.

## Konfigurieren eines verteilten Advantys OTB-E/A-Moduls für den Modbus TCP-E/A-Scanner

### OTB-Slave konfigurieren

Gehen Sie zur Konfiguration des dem **Modbus TCP-E/A-Scanner** hinzugefügten OTB-Slave-Geräts vor wie folgt:

Schritt	Aktion
1	<p>Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf den Knoten des Advantys OTB-Geräts. Ergebnis: Das Konfigurationsfenster wird angezeigt.</p> 
2	Geben Sie in das Feld <b>Slave-IP-Adresse</b> die IP-Adresse des Advantys OTB ein.
3	Geben Sie einen Wert für <b>Timeout für Funktionsfähigkeit (ms)</b> ein (standardmäßig 1000). Dabei handelt es sich um die Verzögerung (in ms) zwischen einem Request vom Modbus TCP IOScanner und der Antwort vom Slave. Nach Ablauf des Funktionsfähigkeits-Timeouts wird das zugehörige Funktionsfähigkeitsbit auf 0 gesetzt. Die Werte der Funktionsfähigkeitsbits können auf der Registerkarte E/A-Scanner E/A-Abbild ( <i>siehe Seite 27</i> ) oder über den Webserver angezeigt werden. Der Funktionsfähigkeits-Timeout gilt für alle Kanäle des Slave-Geräts.
4	Geben Sie einen Wert für <b>Wiederholungsrate (ms)</b> ein (standardmäßig 20). Der <b>Timeout für Funktionsfähigkeit (ms)</b> muss höher sein als die <b>Wiederholungsrate (ms)</b> .
5	Konfigurieren Sie die E/A des Advantys OTB-Geräts auf der Registerkarte <b>OTB E/A-Konfiguration</b> .
6	Fügen Sie die mit dem OTB verbundenen TM2-Erweiterungsmodule hinzu und konfigurieren Sie sie.
7	Rufen Sie einen <code>CONFIGURE_OTB</code> Funktionsbaustein ( <i>siehe Seite 62</i> ) zum Aktualisieren der Advantys OTB-Konfiguration mit den in den vorherigen Schritten erstellten Daten auf.

**HINWEIS:** Die Advantys OTB-Expertenfunktionen, wie z. B. Zähler, schnelle Zähler und Impulsgeneratoren, können nicht direkt im Modbus TCP IOScanner eingesetzt werden.

### Kompatibilität der TM3-Module

TM3-Module sind nicht mit dem Advantys OTB kompatibel.

### Kompatibilität der TM2-Module

In der nachstehenden Tabelle werden die mit Advantys OTB kompatiblen TM2-Module aufgeführt:

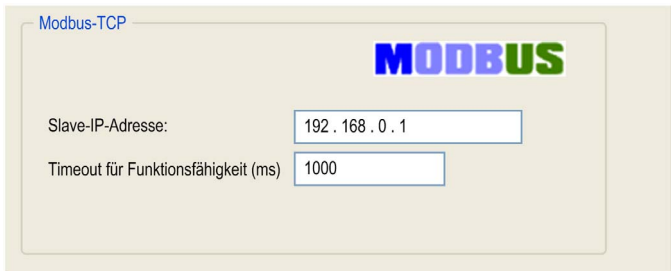
Referenz	Typ
TM2AMI2HT	2 Analogeingänge
TM2AMI2LT	2 Analogeingänge
TM2AMI4LT	4 Analogeingänge
TM2AMI8HT	8 Analogeingänge
TM2ARI8HT	8 Analogeingänge
TM2ARI8LRJ	8 Analogeingänge
TM2ARI8LT	8 Analogeingänge
TM2AMO1HT	1 Analogausgang
TM2AVO2HT	2 Analogausgänge
TM2AMM3HT	2 Analogeingänge 1 Analogausgang
TM2AMM6HT	4 Analogeingänge 2 Analogausgänge
TM2ALM3LT	2 Analogeingänge 1 Analogausgang
TM2DAI8DT	8 Digitaleingänge Signaltyp: AC
TM2DDI8DT	8 Digitaleingänge Signaltyp: Senke/Quelle
TM2DDI16DT	16 Digitaleingänge Signaltyp: Senke/Quelle
TM2DDI16DK	16 Digitaleingänge Signaltyp: Senke/Quelle
TM2DDI32DK	32 Digitaleingänge Signaltyp: Senke/Quelle
TM2DRA8RT	8 Kontakte in 1 gemeinsamen Leitung Ausgangstyp: Relais (NO-Kontakte)
TM2DRA16RT	16 Kontakte in 2 gemeinsamen Leitungen Ausgangstyp: Relais (NO-Kontakte)
TM2DDO8UT	8 Transistorausgänge in 1 gemeinsamen Leitung Signaltyp: Senke
TM2DDO8TT	8 Transistorausgänge in 1 gemeinsamen Leitung Signaltyp: Quelle

Referenz	Typ
TM2DDO16UK	16 Transistorausgänge in 1 gemeinsamen Leitung Signaltyp: Senke
TM2DDO16TK	16 Transistorausgänge in 1 gemeinsamen Leitung Signaltyp: Quelle
TM2DDO32UK	32 Transistorausgänge in 2 gemeinsamen Leitungen Signaltyp: Senke
TM2DDO32TK	32 Transistorausgänge in 2 gemeinsamen Leitungen Signaltyp: Quelle
TM2DMM8DRT	4 Digitaleingänge Signaltyp: Senke/Quelle 1 gemeinsame Leitung mit 4 Kontakten Ausgangstyp: Relais (NO-Kontakte)
TM2DMM24DRF	16 Digitaleingänge Signaltyp: Senke/Quelle 2 gemeinsame Leitungen mit je 8 Kontakten Ausgangstyp: Relais (NO-Kontakte)

## Konfigurieren eines vordefinierten Slave-Geräts für den Modbus TCP-E/A-Scanner

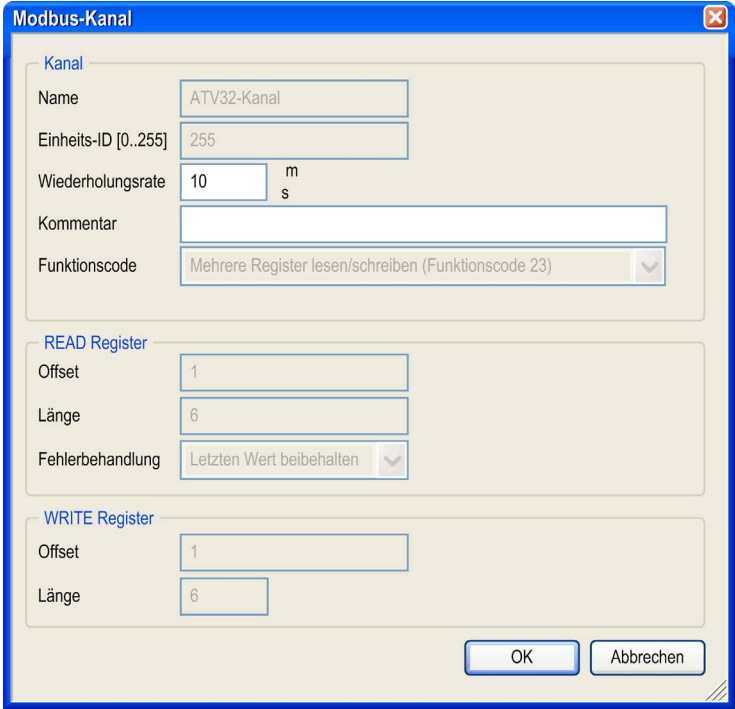
### Vordefinierten Slave konfigurieren, der Modbus TCP IOScanner hinzugefügt wurde

Gehen Sie zur Konfiguration des dem Modbus TCP IOScanner hinzugefügten vordefinierten Slave-Geräts vor wie folgt:

Schritt	Aktion
1	<p>Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf den Knoten des hinzugefügten Slaves. Ergebnis: Das Konfigurationsfenster wird angezeigt.</p> 
2	Geben Sie in das Feld <b>Slave-IP-Adresse</b> die IP-Adresse des Modbus-Slaves ein.
3	Geben Sie einen Wert für <b>Timeout für Funktionsfähigkeit (ms)</b> ein (standardmäßig 1000). Dabei handelt es sich um die Verzögerung (in ms) zwischen einem Request vom Modbus TCP IOScanner und der Antwort vom Slave. Nach Ablauf des Funktionsfähigkeits-Timeouts wird das zugehörige Funktionsfähigkeitsbit auf 0 gesetzt. Die Werte der Funktionsfähigkeitsbits können auf der Registerkarte E/A-Scanner E/A-Abbild ( <i>siehe Seite 27</i> ) oder über den Webserver angezeigt werden. Der Funktionsfähigkeits-Timeout gilt für alle Kanäle des Slave-Geräts.
4	Für Geräte mit erweiterten Einstellungen können einige zusätzliche Einstellungen erforderlich sein. Weitere Informationen hierzu finden Sie im Device Type Manager (DTM) - Benutzerhandbuch.

## Modbus-TCP-Kanal bearbeiten

Gehen Sie zum Bearbeiten von Modbus-Kanal-Parametern für einen vordefinierten Slave vor wie folgt:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf den Knoten des hinzugefügten Slaves.
2	<p>Wählen Sie die Registerkarte <b>Modbus TCP Kanalkonfiguration</b>, und klicken Sie auf die Schaltfläche <b>Bearbeiten....</b>            Ergebnis: Das Fenster <b>Modbus-Kanal</b> wird angezeigt.</p> 
3	Geben Sie die <b>Wiederholungsrate</b> für den Kanal ein. Die Wiederholungsrate ist das Abfrageintervall der Modbus-Requests.
4	Sie können auch einen <b>Kommentar</b> zum Kanaleingeben.
5	Klicken Sie auf <b>OK</b> .

## Konfiguration eines generischen Geräts für den Modbus TCP-E/A-Scanner

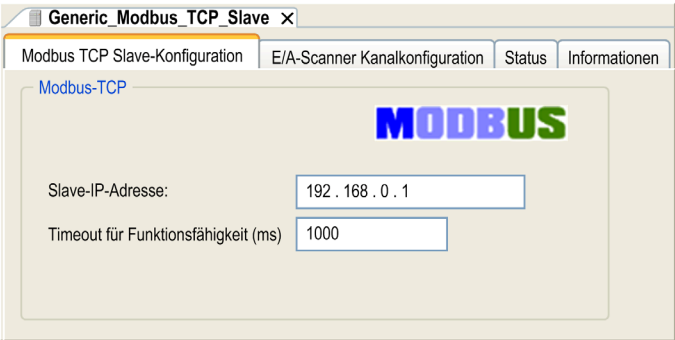
### Überblick

Zum Konfigurieren eines dem Modbus TCP IOScanner hinzugefügten generischen Geräts füllen Sie die Parameter in den folgenden zwei Registerkarten aus:

- **Modbus TCP Slave-Konfiguration**
- **E/A-Scanner Kanalkonfiguration**


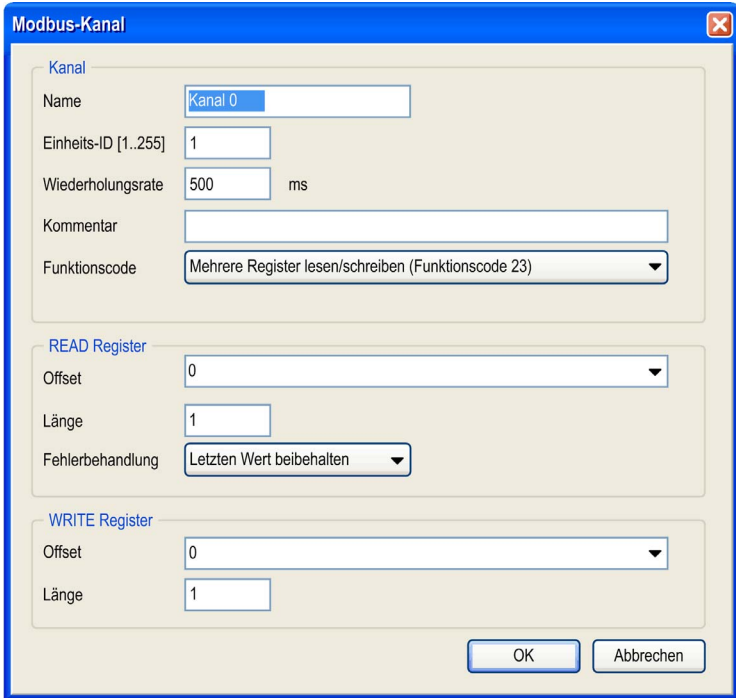
### Registerkarte Modbus TCP Slave-Konfiguration

Gehen Sie zur Konfiguration der Parameter auf der Registerkarte **Modbus TCP Slave-Konfiguration** vor wie folgt:

Schritt	Aktion
1	<p>Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf <b>Generic_Modbus_TCP_Slave</b>. Ergebnis: Das Konfigurationsfenster wird angezeigt.</p> 
2	Geben Sie einen Wert für die <b>Slave-IP-Adresse</b> ein (Standard: 192.168.0.1).
3	Geben Sie einen Wert für <b>Timeout für Funktionsfähigkeit (ms)</b> ein (standardmäßig 1000). Dabei handelt es sich um die Verzögerung (in ms) zwischen einem Request vom Modbus TCP IOScanner und der Antwort vom Slave. Nach Ablauf des Funktionsfähigkeits-Timeouts wird das zugehörige Funktionsfähigkeitsbit auf 0 gesetzt. Die Werte der Funktionsfähigkeitsbits können auf der Registerkarte E/A-Abbild angezeigt werden ( <i>siehe Seite 27</i> ).

## Registerkarte E/A-Scanner Kanalkonfiguration

Gehen Sie zur Konfiguration der Parameter auf der Registerkarte **E/A-Scanner Kanalkonfiguration** vor wie folgt:

Schritt	Aktion
1	<p>Klicken Sie auf die Registerkarte <b>E/A-Scanner Kanalkonfiguration</b>.</p> 
2	Um einen Kanal zu entfernen, wählen Sie ihn aus, und klicken Sie auf <b>Löschen</b> .
3	Um die Parameter eines Kanals zu ändern, wählen Sie den Kanal aus, und klicken Sie auf <b>Bearbeiten</b> .
4	<p>Um einen Kanal hinzuzufügen, klicken Sie auf <b>Kanal hinzufügen</b>. Daraufhin erscheint ein Dialogfeld:</p> 

Schritt	Aktion
5	<p>Im Bereich <b>Kanal</b> können Sie Folgendes definieren:</p> <ul style="list-style-type: none"> <li>● <b>Name:</b> Optionale Zeichenkette zur Benennung des Kanals</li> <li>● <b>Einheits-ID [1..255]:</b> Einheits-ID des Modbus TCP-Slave-Geräts (Standard: 255). Siehe Hinweis.</li> <li>● <b>Wiederholungsrate:</b> Abfrageintervall des Modbus-Requests (Standard: 20 ms)</li> <li>● <b>Kommentar:</b> Optionales Feld für eine Beschreibung des Kanals</li> <li>● <b>Funktionscode:</b> Typ des Modbus-Requests: <ul style="list-style-type: none"> <li>● <b>Mehrere Register lesen/schreiben (Funktionscode 23)</b> (Standard)</li> <li>● <b>Haltereister lesen (Funktionscode 03)</b></li> <li>● <b>Haltereister schreiben (Funktionscode 16)</b></li> </ul> </li> </ul> <p>Im Bereich <b>LESEN-Register</b> können Sie Folgendes definieren:</p> <ul style="list-style-type: none"> <li>● <b>Offset:</b> Anfangsregisternummer für den Lesevorgang (Wert zwischen 0 und 65535)</li> <li>● <b>Länge:</b> Anzahl der zu lesenden Register (je nach Funktionscode)</li> <li>● <b>Fehlerbehandlung:</b> Definition des Fehlerwerts bei Unterbrechung der Kommunikation: <ul style="list-style-type: none"> <li>● <b>Letzten Wert beibehalten</b> (Standard): Der letzte gültige Wert wird beibehalten.</li> <li>● <b>Auf 0 setzen:</b> Alle Werte werden auf 0 zurückgesetzt.</li> </ul> </li> </ul> <p>Im Bereich <b>SCHREIBEN-Register</b> können Sie Folgendes definieren:</p> <ul style="list-style-type: none"> <li>● <b>Offset:</b> Anfangsregisternummer für den Schreibvorgang (Wert zwischen 0 und 65535)</li> <li>● <b>Länge:</b> Anzahl der zu schreibenden Register (je nach Funktionscode)</li> </ul>
6	Klicken Sie auf <b>OK</b> , um die Konfiguration des Kanals zu bestätigen.
7	Wiederholen Sie die Schritte 4 bis 6, um andere Kanäle für die Modbus-Kommunikation mit dem Gerät zu definieren. Für jeden Modbus-Request muss ein Kanal erstellt werden.

**HINWEIS:** Die Einheitenkennung wird für Modbus-TCP-Geräte verwendet, die aus mehreren Modbus-Geräten bestehen, z. B. auf RTU-Gateways von Modbus TCP zu Modbus. In einem solchen Fall ermöglicht die Einheitenkennung, die Slave-Adresse des Geräts hinter dem Gateway zu erreichen. Modbus/TCP-fähige Geräte ignorieren standardmäßig den Parameter Einheitenkennung.

---

# Kapitel 4

## Modbus TCP IOScanner – Betrieb

---

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Modbus TCP IOScanner-Ressourcenprüfung	38
Modbus TCP IOScanner-Betriebsmodi	40
Anwendungsschnittstelle	44

## Modbus TCP IOScanner-Ressourcenprüfung

### Zweck

Auf der Registerkarte **Modbus TCP-IO Scanner-Ressourcen** kann die Last für die Modbus TCP IOScanner-Funktionalität abgeschätzt werden. Prüfen Sie diese Last, bevor Sie die Maschine in Betrieb nehmen.

Zur Verwaltung der Last können Sie einen oder mehrere der folgenden Lastfaktoren anpassen:

- Anzahl Slaves
- Anzahl Kanäle
- Wiederholungsrate

### Lastabschätzung

Mithilfe dieser Gleichung kann die Last für die Modbus TCP IOScanner-Komponente abgeschätzt werden:

$$\text{IOScanner-Last (\%)} = \sum_{\text{Kanal} = 1}^{\text{Anz. Kanäle}} \frac{50}{\text{Wiederholungsrate}_{\text{Kanal}}}$$

In der SoMachine-Software steht eine automatische Lastberechnung zur Verfügung:

Schritt	Aktion
1	Doppelklicken Sie in der <b>Gerätebaumstruktur</b> auf den Knoten <b>Modbus_TCP_IOScanner</b> .
2	Wählen Sie die Registerkarte <b>Modbus TCP-IO Scanner-Ressourcen</b> .
3	Klicken Sie auf <b>Berechnen</b> .

## Beschreibung

In dieser Abbildung ist die Registerkarte **Modbus TCP-IO Scanner-Ressourcen** dargestellt:

The screenshot shows a software window titled "Modbus\_TCP\_IOScanner" with four tabs: "E/A-Scanner E/A-Abbild", "Modbus TCP-IO Scanner-Ressourcen", "Status", and "Informationen". The "Modbus TCP-IO Scanner-Ressourcen" tab is active. It contains a configuration table with four rows, each with a text label, a numeric input field, and a progress bar. The values in the input fields are 6, 22, 22, and 18% respectively. Below the table is a "Berechnen" button.

Parameter	Value	Progress Bar
Anzahl konfigurierter Kanäle	6	2/6 bars
Eingangswörter	22	1/22 bars
Ausgangswörter	22	1/22 bars
IOScanner-Last (%)	18%	4/22 bars

Berechnen

**HINWEIS:** Die Last muss unter 100 % liegen.

## Modbus TCP IOScanner-Betriebsmodi

### Modbus TCP IOScanner-Zustände

Der Modbus TCP IOScanner-Zustand definiert das Verhalten der verschiedenen Geräte im Modbus TCP IOScanner-Netzwerk. Zu jedem Zustand gehören spezifische Überwachungsinformationen (Funktionsfähigkeitsbit, Kommunikationszustände usw.).

Der Modbus TCP IOScanner-Zustand richtet sich nach dem Zustand der Steuerung:

Steuerungszustand	Modbus TCP IOScanner-Zustand
EMPTY	IDLE
STOPPED	STOPPED
HALT	STOPPED
RUNNING	OPERATIONAL
RUNNING with breakpoint	OPERATIONAL mit bestimmtem Verhalten

### Steuerungszustand EMPTY

TCP/IP-Verbindungen sind geschlossen.

Die Zustände der Slave-Geräte werden entsprechend ihrem individuellen Betriebsmodus verwaltet. Im Zustand EMPTY wird der Modbus TCP IOScanner nicht erzeugt. Daher stehen weder Funktionsfähigkeitsbits noch E/A-Images zur Verfügung.

Die folgende Abbildung stellt die Webserverseite in diesem Zustand dar:

The screenshot shows the web interface for the TM251MESE device. The top navigation bar includes 'Home', 'Monitoring', 'Diagnostics', and 'Maintenance'. The 'Diagnostics' menu is expanded, showing options for 'Diagnose', 'Controller', 'TM3 Expansion', 'Ethernet', 'Serial', and 'Scanner-Status'. The 'Scanner-Status' page is active, displaying the following information:

- Scanner-Status:** Ruhe (indicated by a minus sign icon).
- Verbindungsstatistik:**
  - Insg. gesendete Übertragungen: 0
  - Anzahl konfigurierter Verbindungen: 0
  - Ausführungszeit IOScanner (µs): 0
- Status gescannter Geräte:** Keine gescannten Geräte (with up and down arrow icons).
- Legend:**
  - Nicht konfiguriert
  - Gescannt
  - Fehler

## Steuerungszustand STOPPED

TCP/IP-Verbindungen sind geschlossen. Wenn der Modbus TCP IOScanner vom Zustand OPERATIONAL in den Zustand STOPPED übergeht, werden alle Verbindungen zu den Slaves im einseitigen Modus geschlossen.

Die Slave-Geräte werden entsprechend ihrem individuellen Betriebsmodus verwaltet.

In dieser Tabelle werden die SoMachine-Variablen aufgeführt:

Variable	Wert	Kommentare
Wert des Funktionsfähigkeitsbits	0	-
Eingangs-Image	0 oder der zuletzt gelesene Wert	Die Eingangswerte richten sich nach dem Parameter <b>Fehlerbehandlung</b> . Die Eingangswerte entsprechen denen vom Zeitpunkt des Eintritts der Steuerung in den Zustand STOPPED und geben daher möglicherweise nicht den anschließenden tatsächlichen Zustand des Eingangs an.

Die folgende Abbildung stellt die Webserverseite in diesem Zustand dar:

The screenshot shows the web interface for TM251MESE. The top navigation bar includes Home, Monitoring, Diagnostics (selected), and Maintenance. A left sidebar menu lists: Diagnose, Controller, TM3 Expansion, Ethernet, Serial, and Scanner-Status (selected). The main content area is titled 'E/A-Scanner' and contains the following information:

- Scanner-Status:** Gestoppt (indicated by a green checkmark icon).
- Verbindungsstatistik:**
  - Insg. gesendete Übertragungen: 0
  - Anzahl konfigurierter Verbindungen: 3
  - Ausführungszeit IOScanner (µs): 24
- Status gescannter Geräte:** A grid of 64 device status indicators. The first three are marked with a green checkmark icon (Gestoppt), and the remaining 61 are marked with a grey square icon (Nicht konfiguriert). The grid is organized in four rows of 16 devices each, with row numbers 0, 16, 32, and 48 on the left and column numbers 15, 31, 47, and 63 on the right.

At the bottom of the grid, there is a legend:

- Nicht konfiguriert
- Gescannt
- Fehler

### Steuerungszustand HALT

Gleiches Verhalten wie beim Steuerungszustand STOPPED.

### Steuerungszustand RUNNING

TCP/IP-Verbindungen sind offen.

Die Slave-Geräte werden von der Steuerung verwaltet.

In dieser Tabelle werden die SoMachine-Variablen aufgeführt:

Variable	Wert	Kommentare
Wert des Funktionsfähigkeitsbits	0...1	0: Keine Antwort vom Slave bis zum Ablauf des Timeouts. 1: Requests gesendet und beantwortet vor Ablauf des Timeouts.
Eingangs-Image	Letzter gelesener Wert	Die Werte werden synchron mit dem Task <i>(siehe Seite 44)</i> aktualisiert, der den Modbus TCP IOScanner ansteuert.

Die folgende Abbildung stellt die Webserverseite in diesem Zustand dar:

The screenshot shows the web interface for TM251MESE. The navigation bar includes Home, Monitoring, Diagnostics, and Maintenance. The left sidebar shows a 'Diagnose' menu with options: Controller, TM3 Expansion, Ethernet, Serial, and Scanner-Status. The main content area is titled 'E/A-Scanner' and contains:

- Scanner-Status:** In Betrieb (indicated by a green checkmark icon).
- Verbindungsstatistik:**
  - Insg. gesendete Übertragungen: 18639
  - Anzahl konfigurierter Verbindungen: 3
  - Ausführungszeit IOScanner (µs): 123
- Status gescannter Geräte:** A grid of 64 checkboxes (8 rows by 8 columns) representing scanned devices. The first row shows the first three devices as 'Fehler' (red X), the next two as 'Gescannt' (green checkmark), and the remaining three as 'Nicht konfiguriert' (empty). The grid is numbered 0 to 63.

Legend for device status:

- Nicht konfiguriert
- Gescannt
- Fehler

### Steuerung im Zustand RUNNING mit Haltepunkt

TCP/IP-Verbindungen sind offen.

Die Slave-Geräte werden von der Steuerung verwaltet.

In dieser Tabelle werden die SoMachine-Variablen aufgeführt:

Variable	Wert	Kommentare
Wert des Funktionsfähigkeitsbits	1	-
Eingang-Image	Letzter gelesener Wert	Die Eingangswerte entsprechen denen vom Zeitpunkt des Eintritts der Steuerung in den Zustand RUNNING mit Haltepunkt und geben daher möglicherweise nicht den anschließenden tatsächlichen Zustand des Eingangs an.

## Anwendungsschnittstelle

### Überblick

Die Anwendungsschnittstelle ist eine Gruppe von Funktionen und Variablen, die die Kommunikation zwischen der Anwendung und dem Modbus TCP IOScanner ermöglichen:

- Buszyklus-Task
- Statusvariablen
- E/A-Image-Variablen
- Funktionsbausteine

### Buszyklus-Task

Der Modbus TCP IOScanner und die Anwendung tauschen in jedem Zyklus eines Anwendungstasks Daten aus.

Mit dem Parameter **Buszyklus-Task** können Sie den Anwendungstask auswählen, der den Modbus TCP IOScanner verwaltet.

- **Zykluseinstellungen des übergeordneten Busses verwenden:** Ordnet dem Modbus TCP IOScanner den Anwendungstask zu, der die Steuerung verwaltet.
- **MAST:** Ordnet dem Modbus TCP IOScanner den MAST-Task zu.
- **Anderer vorhandener Task:** Sie können einen Task erstellen und dem Modbus TCP IOScanner zuordnen.

Weitere Informationen über die Anwendungstasks finden Sie im SoMachine-Programmierhandbuch.

### Statusvariablen

Es gibt zwei Arten von Statusvariablen:

- **Funktionsfähigkeitsbits:** Variablen zum Angeben des Kommunikationsstatus der Kanäle. Pro Kanal gibt es ein Funktionsfähigkeitsbit.
- **Globaler Scannerstatus:** Variable zum Angeben des Modbus TCP IOScanner-Status.

In dieser Tabelle werden die Werte der Funktionsfähigkeitsbits aufgeführt:

Wert des Funktionsfähigkeitsbits	Kommunikationszustand des Kanals
0	Timeout für Funktionsfähigkeit abgelaufen, ohne dass eine Antwort empfangen wurde.
1	Keine Fehler erkannt. Request und Antwort empfangen.

### E/A-Image-Variablen

Der Modbus TCP IOScanner erfasst Daten von den Slave-Geräten und schreibt Daten auf die Slave-Geräte. Diese Variablen werden E/A-Images genannt.

## Variablenadressen

Jede Variable erhält eine eigene Adresse:

Variable	Typ	Menge
E/A-Image	%IW für Eingänge %QW für Ausgänge	Pro Kanal wird ein Wort-Array erstellt.
Funktionsfähigkeitsbit	%IW	Vier aufeinanderfolgende Wörter
Globaler Scannerstatus	%IW	Ein Wort

Das E/A-Abbild (*siehe Seite 49*) organisiert die Variablen:

- Prüfung der Adressen
- Zuweisung von Variablennamen (Abbild)

## Funktionsbausteine zur Steuerung des Modbus TCP IOScanner

Die Anwendung verwendet mehrere Funktionsbausteine (*siehe Seite 57*), um mit der Steuerung und den Slaves zu kommunizieren:

- CONFIGURE\_OTB
- IOS\_GETSTATE
- IOS\_START
- IOS\_GETHEALTH
- IOS\_STOP

## Funktionsbausteine zur Steuerung von ATV- und Lexium-Geräten

Verwenden Sie die PLC-Open-Funktionsbausteine, um ATV- und Lexium-Geräte zu steuern. Auf diese Funktionsbausteine kann in der Modbus-TCP-Altivar-Bibliothek zugegriffen werden. Weitere Informationen finden Sie in der ATV Modbus TCP Funktionsbausteine Bibliotheks-Anleitung und in der LXM Modbus TCP Funktionsbausteine Bibliotheks-Anleitung.



---

# Kapitel 5

## Modbus TCP IOScanner – Wartung

---

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Diagnose: SoMachine-Onlinemodus	48
Diagnose: Webserver	52
Fehlerbehebung	54

## Diagnose: SoMachine-Onlinemodus



### Überblick

Im Onlinemodus können Sie den Modbus TCP IOScanner in SoMachine mit dem folgenden Methoden überwachen:

- Symbole in der **Gerätebaumstruktur**
- Status-Registerkarte des Modbus TCP IOScanner und der verschiedenen Slave-Geräte
- Registerkarte "E/A-Abbild" des Modbus TCP IOScanner
- Registerkarten "E/A-Abbild" der Slave-Geräte
- Modbus TCP IOScanner-Ressourcenregisterkarte

### Gerätebaumstruktur

Der Kommunikationsstatus des Modbus TCP IOScanner und der Slaves wird in der **Gerätebaumstruktur** mit Symbolen dargestellt:

Symbol	Bedeutung
	Kommunikation mit dem Gerät funktioniert. <b>HINWEIS:</b> Modbus TCP IOScanner wird immer mit diesem Symbol angezeigt.
	Die Steuerung kann nicht mit dem Gerät kommunizieren. <b>HINWEIS:</b> Wenn sich der Modbus TCP IOScanner im Zustand STOPPED befindet, werden alle Geräte mit diesem Symbol angezeigt.

## Modbus TCP IOScanner-Abbild

E/A-Scanner E/A-Abbild		Modbus TCP-IO Scanner-Ressourcen	Status	Informationen			
Kanäle							
Variable	Abbild	Kanal	Adresse	Typ	Stand...	Aktuel...	Vorb...
Diagnose							
Global St...		Global St...	%I...	UINT	0	2	
Healthbits							
Healthbit...		Healthbit...	%I...	WORD		63	
Healthbits_OTB1EODM9LP		Bit 0	%IX...	BOOL	FALSE	TRUE	
Healthbits_Altivar32		Bit 1	%IX...	BOOL	FALSE	TRUE	
Healthbits_Lexium32M		Bit 2	%IX...	BOOL	FALSE	TRUE	
Healthbits_Generic_Slave_channel3		Bit 3	%IX...	BOOL	FALSE	TRUE	
Healthbits_Generic_Slave_channel4		Bit4	%IX...	BOOL	FALSE	TRUE	
Healthbits_Generic_Slave_channel5		Bit5	%IX...	BOOL	FALSE	TRUE	
		Bit6	%IX...	BOOL	FALSE	FALSE	
		Bit 7	%IX...	BOOL	FALSE	FALSE	
		Bit8	%IX...	BOOL	FALSE	FALSE	
		Bit 9	%IX...	BOOL	FALSE	FALSE	
		Bit 10	%IX...	BOOL	FALSE	FALSE	
		Bit 11	%IX...	BOOL	FALSE	FALSE	
		Bit 12	%IX...	BOOL	FALSE	FALSE	
		Bit 13	%IX...	BOOL	FALSE	FALSE	
		Bit 14	%IX...	BOOL	FALSE	FALSE	
		Bit 15	%IX...	BOOL	FALSE	FALSE	
		Healthbit...	%I...	WORD		0	
		Healthbit...	%I...	WORD		0	
		Healthbit...	%I...	WORD		0	

Spalte		Verwendung	Kommentar
Variable	Diagnose	Weisen Sie der globalen Scannerstatusvariablen einen Namen zu.	–
	Healthbits	Weisen Sie jedem Funktionsfähigkeitsbit einen Namen zu. Benennen Sie ein Funktionsfähigkeitsbit beispielsweise mit dem Namen des zugehörigen Geräts.	Funktionsfähigkeitsbits werden in 4 Unterordnern zu je 16 Bits gruppiert.

Spalte	Verwendung	Kommentar
<b>Adresse</b>	Rufen Sie die Adressen der einzelnen Variablen ab.	Adressen können geändert werden, wenn die Konfiguration verändert wird.
<b>Aktueller Wert</b>	Überwachen Sie das Modbus TCP IOScanner-Netzwerk	Für Boolesche Werte (Funktionsfähigkeitsbit): <ul style="list-style-type: none"> <li>● <b>TRUE</b> = 1</li> <li>● <b>FALSE</b> = 0</li> </ul>

### Slave-Abbildung

In dieser Abbildung ist ein Beispiel für die Registerkarte "E/A-Abbild" für ein Advantys OTB-Gerät dargestellt:

Modbus TCP Slave-Konfiguration		OTB-E/A-Konfiguration		Modbus-E/A-Scanner E/A-Abbild		Status	Informationen
Kanäle							
Variable	Abbild	Kanal	Adresse	Typ	Standardwert	Aktueller Wert	
Eingänge							
iwOTB1EODM9LP_Read_Inputs		Read Inputs	%IW18	WORD		2049	
		Bit 0	%IX3...	BOOL	FALSE	TRUE	
		Bit 1	%IX3...	BOOL	FALSE	FALSE	
		Bit 2	%IX3...	BOOL	FALSE	FALSE	
		Bit 3	%IX3...	BOOL	FALSE	FALSE	
		Bit 4	%IX3...	BOOL	FALSE	FALSE	
		Bit 5	%IX3...	BOOL	FALSE	FALSE	
		Bit 6	%IX3...	BOOL	FALSE	FALSE	
		Bit 7	%IX3...	BOOL	FALSE	FALSE	
		Bit 8	%IX3...	BOOL	FALSE	FALSE	
		Bit 9	%IX3...	BOOL	FALSE	FALSE	
		Bit 10	%IX3...	BOOL	FALSE	FALSE	
		Bit 11	%IX3...	BOOL	FALSE	TRUE	
Ausgänge							
qwOTB1EODM9LP_Output_commands		Output co...	%QW1...	WORD		255	
		Bit 0	%QX2.0	BOOL	FALSE	TRUE	
		Bit 1	%QX2.1	BOOL	FALSE	TRUE	
		Bit 2	%QX2.2	BOOL	FALSE	TRUE	
		Bit 3	%QX2.3	BOOL	FALSE	TRUE	
		Bit 4	%QX2.4	BOOL	FALSE	TRUE	
		Bit 5	%QX2.5	BOOL	FALSE	TRUE	
		Bit 6	%QX2.6	BOOL	FALSE	TRUE	
		Bit 7	%QX2.7	BOOL	FALSE	TRUE	

Spalte		Verwendung	Kommentar
Variable	Eingänge	Weisen Sie jedem Eingang des Geräts einen Namen zu.	Außerdem kann jedes Bit zugeordnet werden.
	Ausgänge	Weisen Sie jedem Ausgang des Geräts einen Namen zu.	
Adresse		Rufen Sie die Adressen der einzelnen Variablen ab.	Adressen können geändert werden, wenn die Konfiguration verändert wird.
Aktueller Wert		Verfolgen Sie die Werte der Geräteeingänge in Echtzeit.	Für Boolesche Werte (jedes Bit): <ul style="list-style-type: none"><li>● TRUE = 1</li><li>● FALSE = 0</li></ul>

## Diagnose: Webserver

### Ethernet-Seite

**TM251MESE**

Home | Monitoring | **Diagnostics** | Maintenance

Diagnose  
Controller  
TM3 Expansion  
**Ethernet**  
Serial  
Scanner-Status

**Ethernet**

**Remote Ping Service**

Enter IP address to ping from Controller:

**Statistics**

Ethernet 1	Ethernet 2
MAC-Adresse 0.80.F4.A.1.17	MAC-Adresse 0.80.F4.A.1.16
IP-Adresse 85.15.3.51	IP-Adresse 95.15.1.51
Subnetzmaske 255.0.0.0	Subnetzmaske 255.0.0.0
Gateway-Adresse 0.0.0.0	Gateway-Adresse 0.0.0.0
Status Link up (1)	Status Link up (1)

Ethernet statistics	Modbus statistics
Opened Top connections 2	Messages transmitted OK 181
Fehlerfrei gesendete Frames 355796031	Messages received OK 181
Fehlerfrei empfang. Frames 362481331	Fehlermeldungen 0
Buffers transmitted NOK 0	IpMaster connection status Not connected (1)
Buffers received NOK 0	IpMaster timeout event counter 0

Ethernet IP statistics
IO Messages transmitted 0

In dieser Tabelle sind die Ping-Testergebnisse auf der Seite **Ethernet** aufgeführt:

Symbol	Bedeutung
	Der Kommunikationstest war erfolgreich.
	Die Steuerung kann nicht mit der definierten IP-Adresse kommunizieren.



## Fehlerbehebung

### Häufigste Fehler

Symptom	Mögliche Ursachen	Behebung
Modbus TCP IOScanner wird in der <b>Gerätebaumstruktur</b> mit einem roten Dreieck angezeigt.	die Konfiguration entspricht nicht der Steuerungsversion.	<ul style="list-style-type: none"> <li>● Bereinigen.</li> <li>● Alles neu aufsetzen.</li> <li>● Sicherstellen, dass die Steuerung über die neueste Firmware-Version verfügt.</li> </ul>
Ein Slave wird in der <b>Gerätebaumstruktur</b> mit einem roten Dreieck angezeigt.	Die Steuerung kann nicht mit dem Slave kommunizieren.	<ul style="list-style-type: none"> <li>● Verdrahtung und Spannungsversorgung des Slaves prüfen.</li> <li>● IP-Adresse des Slaves prüfen (über Remote-Ping-Dienst (<i>siehe Seite 52</i>) mit der IP-Adresse des Slave-Geräts).</li> <li>● Prüfen, ob der Slave den Lese-/Schreib-Request unterstützt.</li> <li>● Prüfen, ob die Register, auf die zugegriffen wird, für diesen Slave relevant sind.</li> <li>● Sicherstellen, dass die Register, auf die zugegriffen wird, nicht schreibgeschützt sind.</li> <li>● Sicherstellen, dass der FDR-Dienst (Fast Device Replacement) im Slave ordnungsgemäß konfiguriert ist.</li> <li>● Sicherstellen, dass der Parameter <b>Master-IP-Adresse</b> im Slave ordnungsgemäß konfiguriert ist.</li> </ul>
Ein Slave/Kanal wird <b>vorübergehend</b> rot angezeigt.	Die Verdrahtung ist instabil.	Prüfen Sie die Verdrahtung.
	Konfiguration erfordert Anpassung.	<ul style="list-style-type: none"> <li>● Wert Timeout für Funktionsfähigkeit erhöhen.</li> <li>● Wert Wiederholungsrate erhöhen.</li> </ul>
	Die Last ist zu hoch für den Modbus TCP IOScanner.	Prüfen Sie die Registerkarte <b>Modbus TCP-IO Scanner-Ressourcen</b> .
Einige Zustände des Slaves werden in der Anwendung nicht angezeigt.	Die Wiederholungsrate ist zu langsam (Wert zu hoch).	Verringern Sie den Wert für die Wiederholungsrate bei den diesem Slave zugeordneten Kanälen.
	Die Buszyklus-Task ist nicht schnell genug.	<ul style="list-style-type: none"> <li>● Ordnen Sie dem Modbus TCP IOScanner einen anderen Task zu.</li> <li>● Verringern Sie den Zykluswert des zugeordneten Task.</li> </ul>



## Inhalt dieses Anhangs

Dieser Anhang enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
A	Modbus TCP IOScanner-Funktionen	57
B	Modbus TCP IOScanner – Datentypen	65
C	Darstellung von Funktionen und Funktionsbausteinen	69



---

# Anhang A

## Modbus TCP IOScanner-Funktionen

---

### Überblick

Dieses Kapitel behandelt die in der `ModbusTCPIOScanner`-Bibliothek enthaltenen Funktionen.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
IOS_GETSTATE: Status des Modbus TCP IOScanner lesen	58
IOS_START: Modbus TCP IOScanner starten	59
IOS_GETHEALTH: Wert des Funktionsfähigkeitsbits lesen	60
IOS_STOP: Modbus TCP IOScanner anhalten	61
CONFIGURE_OTB: Softwarekonfiguration des Advantys OTB senden	62

## IOS\_GETSTATE: Status des Modbus TCP IOScanner lesen

### Beschreibung der Funktion

Diese Funktion gibt den Wert zurück, der dem Status des Modbus TCP IOScanner entspricht.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel Darstellung von Funktionen und Funktionsbausteinen ([siehe Seite 69](#))

### Beschreibung der E/A-Variablen

Die folgende Tabelle beschreibt die Ausgangsvariable:

Ausgang	Typ	Kommentar
IOS_GETSTATE	iosStateCodes ( <a href="#">siehe Seite 66</a> )	Rückgabewerte: iosStateCodes enum

### Beispiel

Dies ist ein Beispiel für einen Aufruf dieser Funktion:

```
mystate := IOS_GETSTATE() ; (* 0=NOT CONFIGURED 2=OPERATIONAL or
3=STOPPED. *)
```

## IOS\_START: Modbus TCP IOScanner starten

### Beschreibung der Funktion

Diese Funktion startet den Modbus TCP IOScanner.

Sie ermöglicht die Laufzeitsteuerung der Ausführung des Modbus TCP IOScanner. Standardmäßig startet der Modbus TCP IOScanner als Anwendung.

Dieser Funktionsaufruf wartet, bis der Modbus TCP IOScanner physisch gestartet wurde, kann also bis zu 5 ms dauern.

Das Starten eines bereits gestarteten Modbus TCP IOScanner hat keine Wirkung.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel Darstellung von Funktionen und Funktionsbausteinen (*siehe Seite 69*).

### Beschreibung der E/A-Variablen

Die folgende Tabelle beschreibt die Ausgangsvariable:

Ausgang	Typ	Kommentar
IOS_START	UDINT	0 = erfolgreicher Start Anderer Wert = Start nicht erfolgreich

### Beispiel

Dies ist ein Beispiel für einen Aufruf dieser Funktion:

```
rc := IOS_START() ;
IF rc <> 0 THEN (* Abnormal situation to be processed at application level
*)
```

## IOS\_GETHEALTH: Wert des Funktionsfähigkeitsbits lesen

### Beschreibung der Funktion

Diese Funktion gibt den Wert des Funktionsfähigkeitsbits eines bestimmten Kanals zurück.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel Darstellung von Funktionen und Funktionsbausteinen (*siehe Seite 69*).

### Beschreibung der E/A-Variablen

In der folgenden Tabelle wird die Eingangsvariable beschrieben:

Eingang	Typ	Kommentar
channelID	UINT	Kanal-ID ( <i>siehe Seite 15</i> ) des zu überwachenden Kanals-

In der folgenden Tabelle wird die Ausgangsvariable beschrieben:

Ausgang	Typ	Kommentar
IOS_GETHEALTH	UINT	0: E/A-Werte des Kanals werden nicht aktualisiert 1: E/A-Werte des Kanals werden aktualisiert

### Beispiel

Dies ist ein Beispiel für einen Aufruf dieser Funktion:

```
chID:=1 ;
channelHealth := IOS_GETHEALTH(chID) (* Get the health value (1=OK, 0=Not
OK) of the channel number chID. The channel ID is displayed in the
configuration editor of the device *)
```

## IOS\_STOP: Modbus TCP IOScanner anhalten

### Beschreibung der Funktion

Diese Funktion hält den Modbus TCP IOScanner an.

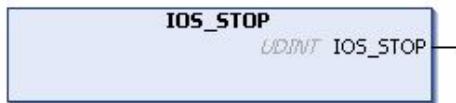
Sie ermöglicht die Laufzeitsteuerung der Ausführung des Modbus TCP IOScanner. Standardmäßig wird der Modbus TCP IOScanner angehalten, wenn die Steuerung in den Zustand STOPPED übergeht.

Der Modbus TCP IOScanner muss beim ersten Anwendungszyklus angehalten werden, da es sich bei dieser Funktion um einen synchronen Aufruf handelt, der einige Zeit dauern kann.

Der Funktionsaufruf kann bis zu 5 ms dauern, da er warten muss, bis Modbus TCP IOScanner physisch angehalten wurde.

Das Anhalten eines bereits angehaltenen Modbus TCP IOScanner hat keine Wirkung.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel Darstellung von Funktionen und Funktionsbausteinen (*siehe Seite 69*).

### Beschreibung der E/A-Variablen

Die folgende Tabelle beschreibt die Ausgangsvariable:

Ausgang	Typ	Kommentar
IOS_STOP	UDINT	0 = erfolgreich angehalten Anderer Wert = Anhalten nicht erfolgreich

### Beispiel

Dies ist ein Beispiel für einen Aufruf dieser Funktion:

```
rc := IOS_STOP() ;
IF rc <> 0 THEN (* Abnormal situation to be processed at application level
*)
```

## CONFIGURE\_OTB: Softwarekonfiguration des Advantys OTB senden

### Beschreibung der Funktion

Diese Funktion sendet die SoMachine-Konfigurationsdaten eines Advantys OTB über Modbus TCP an das physische Gerät.

Sie ermöglicht die Aktualisierung des Konfigurationsparameters einer E/A-Insel ohne Fremdanzbietersoftware.

Der Modbus TCP IOScanner muss angehalten werden, bevor diese Funktion aufgerufen wird.

Die Ausführung des Bausteins erfolgt asynchron. Um den Fortschritt der Konfiguration zu prüfen, müssen die Ausgangs-Flags `Done`, `Busy` und `Error` bei jedem Anwendungszyklus getestet werden.

### Grafische Darstellung



### Darstellung in AWL (IL) und ST

Sie finden eine allgemeine Darstellung in AWL (IL) oder ST im Kapitel Darstellung von Funktionen und Funktionsbausteinen ([siehe Seite 69](#)).

### Beschreibung der E/A-Variablen

In der folgenden Tabelle werden die Eingangsvariablen beschrieben:

Eingang	Typ	Kommentar
<code>Execute</code>	BOOL	Aktivierungseintrag. Startet die Konfiguration an der steigenden Flanke.
<code>sAddr</code>	STRING	OTB-IP-Adresse. Das Format der Zeichenfolge muss <code>3{xx.xx.xx.xx}</code> sein.

In der folgenden Tabelle werden die Ausgangsvariablen beschrieben:

Ausgang	Typ	Kommentar
Done	BOOL	Auf TRUE gesetzt, wenn die Konfiguration erfolgreich abgeschlossen wurde.
Busy	BOOL	Auf TRUE gesetzt, wenn die Konfiguration noch läuft.
Error	BOOL	Auf TRUE gesetzt, wenn die Konfiguration mit einem Fehler beendet wurde.
ConfError	configurationOTBError-Codes ( <i>siehe Seite 68</i> )	Rückgabewerte: configurationOTBErrorCodes
CommError	CommunicationError-Codes ( <i>siehe Seite 67</i> )	Rückgabewerte: CommunicationErrorCodes

### Beispiel

Dies ist ein Beispiel für einen Aufruf dieser Funktion:

```
VAR
```

(\*Funktionsbaustein zur Konfiguration des OTB; IOScanner muss angehalten werden, bevor FB ausgeführt wird\*)

```
configure_OTB1: CONFIGURE_OTB;
```

(\*Initialwert von 16#00000000 verschieden; IO\_start\_done=0 bei erfolgreichem Start\*)

```
IO_start_done: UDINT := 1000;
```

(\*Initialwert von 16#FFFFFFFF verschieden; IO\_start\_done=16#FFFFFFFF bei erfolgreichem Anhalten\*)

```
IO_stop_done: UDINT := 1000;
```

(\*Configure\_OTB\_done= true bei erfolgreicher Konfiguration des OTB; anschließend kann der IOScanner gestartet werden\*)

```
Configure_OTB_done: BOOL;
```

```
myBusy: BOOL;
```

```
myError: BOOL;
```

```
myConfError: configurationOTBErrorCodes;
```

```
myCommError: UINT;
```

```
myExecute: BOOL;
```

```
END_VAR
```

(\*IOScanner erst anhalten, bevor OTB konfiguriert wird\*)

```
IF NOT myExecute THEN  
IO_stop_done:=IOS_STOP();  
END_IF
```

(\*Konfigurationsdaten an OTB senden, an IP-Adresse 95.15.3.1, wenn myExecute = TRUE\*)

```
configure_OTB1(  
Execute:= myExecute,  
sAddr:='3{95.15.3.1}' ,  
Done=> Configure_OTB_done,  
Busy=> myBusy,  
Error=&gt; myError,  
ConfError=&gt; myConfError,  
CommError=&gt; myCommError);
```

(\*Nachdem OTB erfolgreich konfiguriert wurde, IOScanner starten\*)

```
IF Configure_OTB_done THEN  
IO_start_done:=IOS_START();  
END_IF
```

---

# Anhang B

## Modbus TCP IOScanner – Datentypen

---

### Überblick

In diesem Kapitel werden die Datentypen der `ModbusTCPIOScanner`-Bibliothek beschrieben.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
<code>IoStateCodes</code> : Modbus TCP IOScanner-Statuswerte	66
<code>CommunicationErrorCodes</code> : Fehlercodes	67
<code>configurationOTBErrorCodes</code> : Codes für Fehler in der OTB-Konfiguration	68

## IosStateCodes: Modbus TCP IOScanner-Statuswerte

### Beschreibung des Aufzählungstyps

Der Aufzählungsdatentyp `IosStateCodes` enthält die folgenden Werte:

Enumerator	Wert	Kommentar
<code>IosErr</code>	0	Modbus TCP IOScanner befindet sich in einem Fehlerzustand.
<code>IosIdle</code>	1	Modbus TCP IOScanner befindet sich im Zustand IDLE. Die Konfiguration ist leer oder nicht konform.
<code>IosOperationnal</code>	2	Modbus TCP IOScanner befindet sich im Zustand OPERATIONAL.
<code>IosStopped</code>	3	Modbus TCP IOScanner befindet sich im Zustand STOPPED.

## CommunicationErrorCodes: Fehlercodes

### Beschreibung des Aufzählungstyps

Der Aufzählungsdattentyp `CommunicationErrorCodes` enthält die folgenden Werte:

Enumerator	Wert	Kommentar
<code>CommunicationOK</code>	hex 00	Austausch ist korrekt.
<code>TimedOut</code>	hex 01	Austausch wegen Timeouts angehalten.
<code>Canceled</code>	hex 02	Austauschs auf Verlangen des Benutzers angehalten.
<code>BadAddress</code>	hex 03	Adressenformat ist ungültig.
<code>BadRemoteAddr</code>	hex 04	Remote-Adresse ist ungültig.
<code>BadMgtTable</code>	hex 05	Format der Verwaltungstabelle ist ungültig.
<code>BadParameters</code>	hex 06	Spezifische Parameter sind ungültig.
<code>ProblemSendingRq</code>	hex 07	Fehler beim Senden des Requests an das Ziel festgestellt.
<code>RecvBufferTooSmall</code>	hex 09	Größe des Empfangspuffers zu klein.
<code>SendBufferTooSmall</code>	hex 0A	Größe des Sendepuffers zu klein.
<code>SystemResourceMissing</code>	hex 0B	Systemressource fehlt.
<code>BadTransactionNb</code>	hex 0C	Transaktionsnummer ist ungültig.
<code>BadLength</code>	hex 0E	Länge ist ungültig.
<code>ProtocolSpecificError</code>	hex FE	Festgestellter Betriebsfehler enthält protokollspezifischen Code.
<code>Refused</code>	hex FF	Transaktion abgelehnt.

## configurationOTBErrorCodes: Codes für Fehler in der OTB-Konfiguration

### Beschreibung des Aufzählungstyps

Der Aufzählungsdatentyp `configurationOTBErrorCodes` enthält die folgenden Werte:

Enumerator	Wert	Kommentar
<code>ConfigurationOK</code>	hex 00	OTB-Konfiguration erfolgreich durchgeführt.
<code>IPAddrErr</code>	hex 01	Eingangsparameter <code>sAddr</code> ist ungültig.
<code>ChannelNbErr</code>	hex 02	Für diese IP-Adresse ist kein OTB-Kanal-Initialisierungswert vorhanden.
<code>ChannelInitValueErr</code>	hex 03	OTB-Kanal-Initialisierungswert kann nicht abgerufen werden.
<code>CommunicationErr</code>	hex 04	OTB-Konfiguration aufgrund eines Fehlers angehalten.
<code>IosStateErr</code>	hex 05	Modbus TCP IOScanner wird ausgeführt. Modbus TCP IOScanner muss angehalten werden, bevor der Funktionsbaustein <code>CONFIGURE_OTB</code> ausgeführt werden kann.

---

# Anhang C

## Darstellung von Funktionen und Funktionsbausteinen

---

### Übersicht

Jede Funktion kann in den folgenden Sprachen dargestellt werden.

- AWL: Anweisungsliste
- ST: Strukturierter Text
- KOP: Kontaktplan
- FBD: Funktionsbausteindiagramm
- CFC: Continuous Function Chart

Dieses Kapitel enthält Darstellungen von Funktionen und Funktionsbausteinen und erläutert deren Verwendung in den Sprachen AWL und ST.

### Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Unterschiede zwischen einer Funktion und einem Funktionsbaustein	70
Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache AWL	71
Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache ST	75

## Unterschiede zwischen einer Funktion und einem Funktionsbaustein

### Funktion

Eine Funktion hat die folgenden Eigenschaften:

- Ist eine POU (Program Organization Unit), die ein einzelnes direktes Ergebnis zurückgibt
- Wird direkt über ihren Namen aufgerufen (nicht über eine Instanz)
- Ist nicht instanziiert
- Kann als Operand in anderen Ausdrücken verwendet werden

**Beispiele:** Boolesche Operatoren (AND), Berechnungen, Konvertierung (BYTE\_TO\_INT)

### Funktionsbaustein

Ein Funktionsbaustein hat die folgenden Eigenschaften:

- Ist eine POU (Program Organization Unit), die ein oder mehrere direkte Ausgänge zurückgibt
- Muss von einer Instanz aufgerufen werden (Funktionsbausteinkopie mit dediziertem Namen und Variablen)
- Hat für jede Instanz einen persistenten Status (Ausgänge und interne Variablen) von einem Aufruf zum anderen aus einem Funktionsbaustein oder Programm

**Beispiele:** Zeitgeber, Zähler

In dem nachstehenden Beispiel ist `Timer_ON` eine Instanz des Funktionsbausteins `TON`:

```
1  PROGRAM MyProgram_ST
2  VAR
3      Timer_ON: TON; // Function Block Instance
4      Timer_RunCd: BOOL;
5      Timer_PresetValue: TIME := T#5S;
6      Timer_Output: BOOL;
7      Timer_ElapsedTime: TIME;
8  END_VAR
```

```
1  Timer_ON(
2      IN:=Timer_RunCd,
3      PT:=Timer_PresetValue,
4      Q=>Timer_Output,
5      ET=>Timer_ElapsedTime);
```

## Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache AWL

### Allgemeine Informationen

In diesem Abschnitt wird das Implementieren einer Funktion und eines Funktionsbausteins in der Sprache AWL beschrieben.

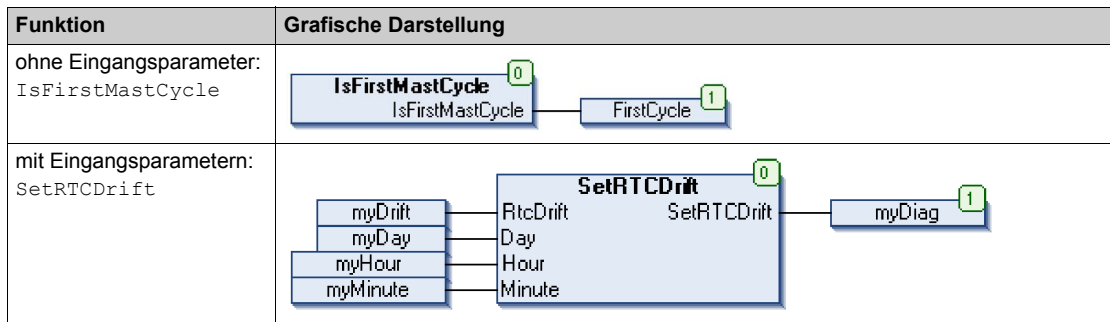
Die Funktionen `IsFirstMastCycle` und `SetRTCDrift` und der Funktionsbaustein `TON` werden als Implementierungsbeispiele verwendet.

### Verwenden einer Funktion in der AWL-Sprache

Im Folgenden wird das Einfügen einer Funktion in der AWL-Sprache beschrieben:

Schritt	Aktion
1	Öffnen oder erstellen Sie eine neue POU in der AWL-Sprache. <b>HINWEIS:</b> Die Vorgehensweise zum Erstellen einer POU wird hier nicht erläutert. Weitere Informationen finden Sie unter Hinzufügen und Aufrufen von POU's ( <i>siehe SoMachine, Programmierhandbuch</i> ).
2	Erstellen Sie die Variablen, die für die Funktion erforderlich sind.
3	Wenn die Funktion über mindestens einen Eingang verfügt, beginnen Sie mit dem Laden des ersten Eingangs mithilfe der LD-Anweisung.
4	Fügen Sie unten eine neue Zeile ein, und gehen Sie wie folgt vor: <ul style="list-style-type: none"> <li>● Geben Sie den Namen der Funktion in der Operator-Spalte (linkes Feld) ein.</li> <li>● Oder verwenden Sie die <b>Eingabehilfe</b>, um die Funktion auszuwählen. (Wählen Sie im Kontextmenü <b>Baustein aufruf einfügen</b>.)</li> </ul>
5	Wenn die Funktion über mehr als einen Eingang verfügt und die Eingabehilfe verwendet wird, wird die erforderliche Anzahl von Zeilen automatisch mit ??? in den Feldern rechts erstellt. Ersetzen Sie ??? durch den geeigneten Wert oder die Variable, die der Reihenfolge der Eingänge entspricht.
6	Fügen Sie eine neue Zeile ein, um das Ergebnis der Funktion in der entsprechenden Variable zu speichern: Geben Sie die ST-Anweisung in die Operator-Spalte (linkes Feld) und einen Variablennamen in das rechte Feld ein.

Die Funktionen `IsFirstMastCycle` (ohne Eingangsparameter) und `SetRTCDrift` (mit Eingangsparametern) werden im Folgenden grafisch dargestellt:



In der AWL-Sprache wird der Funktionsname direkt in der Operator-Spalte verwendet:

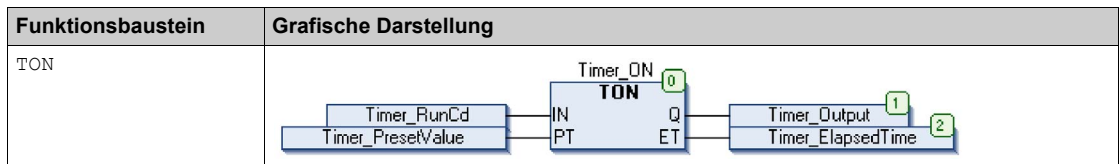
Funktion	Darstellung im SoMachine POU-Editor in AWL
Beispiel einer Funktion ohne Eingangsparameter in der AWL-Sprache: <code>IsFirstMastCycle</code>	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      FirstCycle: BOOL; 4  END_VAR 5 </pre> <hr/> <pre> 1  IsFirstMastCycle    ST          FirstCycle </pre>
Beispiel einer Funktion mit Eingangsparametern in der AWL-Sprache: <code>SetRTCDrift</code>	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      myDrift: SINT (-29..29) := 5; 4      myDay: DAY_OF_WEEK := SUNDAY; 5      myHour: HOUR := 12; 6      myMinute: MINUTE; 7      myDiag: RTCSETDRIFT_ERROR; 8  END_VAR 9 </pre> <hr/> <pre> 1  LD          myDrift    SetRTCDrift myDay            myHour            myMinute    ST          myDiag </pre>

## Verwenden eines Funktionsbausteins in der AWL-Sprache

Im Folgenden wird das Einfügen eines Funktionsbausteins in der AWL-Sprache beschrieben:

Schritt	Aktion
1	Erstellen Sie eine neue POU in der AWL-Sprache. <b>HINWEIS:</b> Die Vorgehensweise zum Erstellen einer POU wird hier nicht erläutert. Weitere Informationen finden Sie unter Hinzufügen und Aufrufen von POU's ( <i>siehe SoMachine, Programmierhandbuch</i> ).
2	Erstellen Sie die Variablen, die für den Funktionsbaustein erforderlich sind, einschließlich des Instanznamens.
3	Funktionsbausteine werden mithilfe einer CAL-Anweisung aufgerufen: <ul style="list-style-type: none"> <li>• Verwenden Sie die <b>Eingabehilfe</b>, um den FB auszuwählen. (Klicken Sie mit der rechten Maustaste, und wählen Sie im Kontextmenü <b>Baustein aufruf einfügen</b> aus.)</li> <li>• Die CAL-Anweisung und der entsprechende E/A werden erstellt.</li> </ul> Jeder Parameter (E/A) ist eine Anweisung: <ul style="list-style-type: none"> <li>• Werte für Eingänge werden mit ":" festgelegt.</li> <li>• Werte für Ausgänge werden mit "=" festgelegt.</li> </ul>
4	Ersetzen Sie im rechten CAL-Feld die ??? durch den Instanznamen.
5	Ersetzen Sie weitere ??? durch eine geeignete Variable oder einen direkten Wert.

Der grafisch dargestellte Funktionsbaustein TON dient in diesem Beispiel zur Veranschaulichung:



In der AWL-Sprache wird der Name des Funktionsbausteins direkt in der Operator-Spalte verwendet:

Funktionsbaustein	Darstellung im SoMachine POU-Editor in AWL
TON	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3  Timer_ON: TON; // Function Block instance declaration 4  Timer_RunCd: BOOL; 5  Timer_PresetValue: TIME := T#5S; 6  Timer_Output: BOOL; 7  Timer_ElapsedTime: TIME; 8  END_VAR 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 </pre>

## Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache ST

### Allgemeine Informationen

In diesem Teil wird die Implementierung einer Funktion oder eines Funktionsbausteins in der ST-Sprache erläutert.

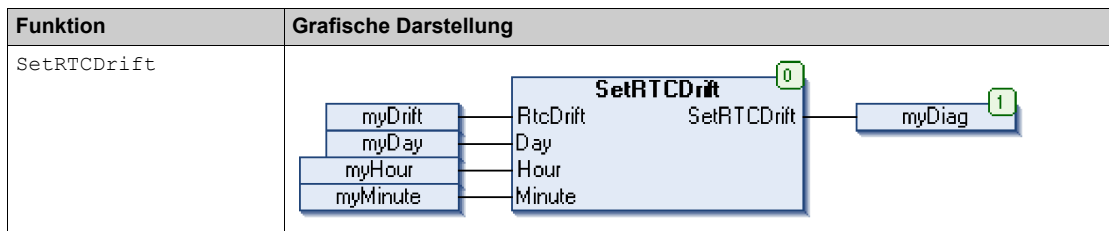
Dabei werden die Funktion `SetRTCDrift` und der Funktionsbaustein `TON` als Beispiele verwendet.

### Verwenden einer Funktion in der ST-Sprache

Im Folgenden wird das Einfügen einer Funktion in der ST-Sprache beschrieben:

Schritt	Aktion
1	Erstellen Sie eine neue POU in der ST-Sprache. <b>HINWEIS:</b> Die Vorgehensweise zum Erstellen einer POU wird hier nicht erläutert. Weitere Informationen finden Sie unter Hinzufügen und Aufrufen von POU's ( <i>siehe SoMachine, Programmierhandbuch</i> ).
2	Erstellen Sie die Variablen, die für die Funktion erforderlich sind.
3	Verwenden Sie im <b>POU-ST-Editor</b> die allgemeine Syntax zur Darstellung einer Funktion in der ST-Sprache. Die allgemeine Syntax lautet: Funktionsergebnis:= Funktionsname(VarEingang1, VarEingang2,.. VarEingangx);

Zur Veranschaulichung dieses Verfahrens betrachten wir die grafisch dargestellte Funktion `SetRTCDrift`:



In der ST-Sprache wird diese Funktion folgendermaßen dargestellt:

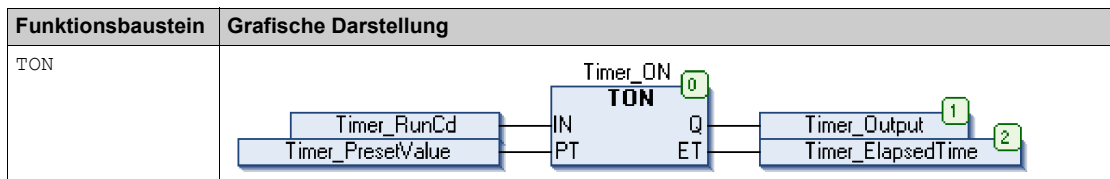
Funktion	Darstellung im SoMachine POU-Editor in der ST-Sprache
SetRTCDrift	<pre> PROGRAM MyProgram_ST VAR myDrift: SINT(-29..29) := 5; myDay: DAY_OF_WEEK := SUNDAY; myHour: HOUR := 12; myMinute: MINUTE; myRTCAjust: RTCDRIFT_ERROR; END_VAR myRTCAjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);                     </pre>

### Verwenden eines Funktionsbausteins in der ST-Sprache

Im Folgenden wird das Einfügen eines Funktionsbausteins in der ST-Sprache beschrieben:

Schritt	Aktion
1	<p>Erstellen Sie eine neue POU in der ST-Sprache.</p> <p><b>HINWEIS:</b> Die Vorgehensweise zum Erstellen einer POU wird hier nicht erläutert. Weitere Informationen zum Hinzufügen, Deklarieren und Aufrufen von POU's finden Sie in der entsprechenden Dokumentation (<i>siehe SoMachine, Programmierhandbuch</i>).</p>
2	<p>Erstellen Sie die Eingangs- und Ausgangsvariablen und die Instanzen, die für den Funktionsbaustein erforderlich sind:</p> <ul style="list-style-type: none"> <li>Die Eingangsvariablen sind die für den Funktionsbaustein erforderlichen Eingangsparameter.</li> <li>Die Ausgangsvariablen erhalten den vom Funktionsbaustein zurückgegebenen Wert.</li> </ul>
3	<p>Verwenden Sie im <b>POU-ST-Editor</b> die allgemeine Syntax zur Darstellung eines Funktionsbausteins in der ST-Sprache. Die allgemeine Syntax lautet:</p> <p>Funktionsbaustein_Instanzname (Eingang1:=VarEingang1, Eingang2:=VarEingang2, ... Ausgang1=&gt;VarAusgang1, Ausgang2=&gt;VarAusgang2, ...);</p>

Der grafisch dargestellte Funktionsbaustein TON dient in diesem Beispiel zur Veranschaulichung:



Die folgende Tabelle zeigt Beispiele für den Aufruf eines Funktionsbausteins in der ST-Sprache:

Funktionsbaustein	Darstellung im SoMachine POU-Editor in der ST-Sprache
TON	<pre> 1  PROGRAM MyProgram_ST 2  VAR 3      Timer_ON: TON; // Function Block Instance 4      Timer_RunCd: BOOL; 5      Timer_PresetValue: TIME := T#5S; 6      Timer_Output: BOOL; 7      Timer_ElapsedTime: TIME; 8  END_VAR  1  Timer_ON( 2      IN:=Timer_RunCd, 3      PT:=Timer_PresetValue, 4      Q=&gt;Timer_Output, 5      ET=&gt;Timer_ElapsedTime); </pre>





## B

### Byte

In einem 8-Bit-Format codierter Typ. Gültiger Bereich in hexadezimaler Darstellung: 16#00 bis 16#FF.

## C

### CFC

(*Continuous Function Chart*) Grafische Programmiersprache (Erweiterung des Standards IEC 61131-3) auf der Grundlage der FBD-Sprache (Funktionsbausteindiagramm), die wie ein Flussdiagramm aufgebaut ist. Grafische Elemente werden allerdings, sofern möglich, ohne die Verwendung von Netzwerken frei positioniert, sodass Rückkopplungsschleifen möglich sind. Bei jedem Baustein befinden sich die Eingänge links und die Ausgänge rechts. Sie können die Bausteinausgänge mit den Eingängen anderer Bausteine verbinden, um komplexe Ausdrücke zu erstellen.

## D

### DHCP

(*Dynamic Host Configuration Protocol*) Hochentwickelte Erweiterung von BOOTP. Das DHCP-Protokoll ist ausgereifter, doch sowohl DHCP als auch BOOTP sind gängig. (DHCP kann BOOTP-Client-Requests verarbeiten.)

## F

### FB

(*Function Block: Funktionsbaustein*) Nützlicher Programmiermechanismus, der eine Gruppe von Programmieranweisungen zur Durchführung eines spezifischen und normierten Vorgangs konsolidiert, z. B. Drehzahlregelung, Intervallkontrolle oder Zählen. Ein Funktionsbaustein kann Konfigurationsdaten, eine Gruppe interner oder externer Betriebsparameter und in der Regel 1 oder mehrere Dateneingänge und -ausgänge umfassen.

### FDR

(*Fast Device Replacement: Schneller Geräte austausch*)

### **Funktionsbausteindiagramm (Programmiersprache)**

Eine von 5 Sprachen für die Logik oder Steuerung, die von dem Standard IEC 61131-3 für Steuerungssysteme unterstützt wird. Es handelt sich hierbei um eine grafisch orientierte Programmiersprache. Sie arbeitet mit einer Liste von Netzwerken, wobei jedes Netzwerk eine grafische Struktur von Feldern und Verbindungslinien enthält, die entweder einen logischen oder einen arithmetischen Ausdruck, den Aufruf eines Funktionsbausteins, einen Sprung oder einen Rückkehrbefehl darstellen.

### **Funktionsfähigkeitsbit**

Variable, die den Kommunikationszustand der Kanäle angibt.

## I

### **IL**

(*Instruction List: Anweisungsliste (AWL)*) Ein in Anweisungsliste geschriebenes Programm besteht aus einer Abfolge textbasierter Anweisungen, die von der Steuerung der Reihe nach ausgeführt werden. Jede Anweisung besteht aus einer Zeilennummer, einem Anweisungscode und einem Operanden (siehe IEC 61131-3).

### **INT**

(*Integer: Ganzzahl*) Über 16 Bits codierte Ganzzahl.

## L

### **LD**

(*Ladder Diagramm: Kontaktplan (KOP)*) Grafische Darstellung der Anweisungen eines Steuerungsprogramms mit Symbolen für Kontakte, Spulen und Bausteine in einer Abfolge von Programmbausteinen, die von der Steuerung der Reihe nach ausgeführt werden (siehe IEC 61131-3).

## M

### **Modbus Kanal**

Kommunikationsshuttle, der einen Modbus-Request zwischen Master und Slave überträgt.

## P

### **POU**

(*Program Organization Unit: Programmierorganisationseinheit*) Variablendeklaration im Quellcode und der entsprechende Anweisungssatz. POU's ermöglichen die modulare Wiederverwendung von Softwareprogrammen, Funktionen und Funktionsbausteinen. Sobald POU's deklariert sind, stehen sie sich gegenseitig zur Verfügung.

**S****ST**

(*Structured Text: Strukturierter Text*) Programmiersprache, die komplexe und verschachtelte Anweisungen umfasst (z. B. Iterationsschleifen, bedingte Ausführungen oder Funktionen). ST ist IEC 61131-3-kompatibel.

**T****Timeout für Funktionsfähigkeit**

Bei diesem Wert handelt es sich um die maximale Zeit (in ms) zwischen einem Request des Modbus TCP-E/A-Scanners und der Antwort des Slaves.

**V****Variable**

Speichereinheit, die von einem Programm adressiert und geändert werden kann.

**W****Wiederholungsrate**

Abfrageintervall des gesendeten Modbus-Requests.





## A

- Anwendungsschnittstelle
  - Modbus-TCP-E/A-Scanner, 44
- Architektur
  - Modbus-TCP-E/A-Scanner, 12

## B

- Betriebsmodi
  - Modbus-TCP-E/A-Scanner, 40
- Buszyklus-Task
  - Modbus-TCP-E/A-Scanner, 44

## C

- CommunicationErrorCodes
  - Datentypen, 67
- configurationOTBErrorCodes
  - Datentypen, 68
- CONFIGURE\_OTB
  - Funktionen, 62

## D

- Datentypen
  - CommunicationErrorCodes, 67
  - configurationOTBErrorCodes, 68
  - iosStateCodes, 66

## F

- Fehlerbehebung
  - Modbus-TCP-E/A-Scanner, 54
- Funktionen
  - CONFIGURE\_OTB, 62
  - IOS\_GETHEALTH, 60
  - IOS\_GETSTATE, 58
  - IOS\_START, 59
  - IOS\_STOP, 61
  - Unterschiede zwischen einer Funktion

- und einem Funktionsbaustein, 70
- Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache AWL, 71
- Verwenden einer Funktion oder eines Funktionsbausteins in der Sprache ST, 75
- Funktionsbausteine
  - Modbus-TCP-E/A-Scanner, 45

## I

- IOS\_GETHEALTH
  - Funktionen, 60
- IOS\_GETSTATE
  - Funktionen, 58
- IOS\_START
  - Funktionen, 59
- IOS\_STOP
  - Funktionen, 61
- iosStateCodes
  - Datentypen, 66
- IP-Adressen
  - Modbus-TCP-E/A-Scanner, 20

## L

- Last berechnen
  - Modbus-TCP-E/A-Scanner, 38

## M

- M251-Webserver
  - Modbus-TCP-E/A-Scanner, 52
- Modbus TCP-E/A-Scanner
  - Hinzufügen eines Geräts, 26
- Modbus TCP-IO Scanner-Ressourcen, Registerkarte
  - Modbus-TCP-E/A-Scanner, 38

Modbus-TCP-E/A-Scanner

- Anwendungsschnittstelle, 44
- Architektur, 12
- Betriebsmodi, 40
- Buszyklus-Task, 44
- Fehlerbehebung, 54
- Funktionsbausteine, 45
- IP-Adressen, 20
- Konfigurieren eines generischen Geräts, 34
- Konfigurieren eines OTB-Geräts, 29
- Konfigurieren eines vordefinierten Geräts, 32
- Last berechnen, 38
- M251-Webserver, 52
- Modbus TCP-IO Scanner-Ressourcen, Registerkarte, 38
- Netzwerkplanung, 18
- Netzwerktests, 23
- Prinzipien, 14
- Überblick, 10
- Überwachung mit SoMachine , 48
- Zustände, 40

Modbus-TCP-E/A-Scanners

- Hinzufügen und Konfigurieren, 27

## N

Netzwerkplanung

- Modbus-TCP-E/A-Scanner, 18

Netzwerktests

- Modbus-TCP-E/A-Scanner, 23

## P

Prinzipien

- Modbus-TCP-E/A-Scanner, 14

## U

Überblick

- Modbus-TCP-E/A-Scanner, 10

Überwachung mit SoMachine

- Modbus-TCP-E/A-Scanner , 48

## Z

Zustände

- Modbus-TCP-E/A-Scanner, 40