ADDI-DATA®
PARTNER FÜR PRÄZISION

Praktische Anleitung

Intelligente Ethernet-E/A-Systeme (Teil 1)

MSX-Exxxx und SPS: Grundlagen

DIN EN ISO 9001:2015 zertifiziert

Ausgabe: 02.05-09/2018

Produktinformation
Dieses Handbuch enthält die technischen Anlagen, wichtige Anleitungen zur korrekten Inbetriebnahme und Nutzung sowie Produktinformation entsprechend dem aktuellen Stand vor der Drucklegung.
Der Inhalt dieses Handbuchs und die technischen Daten des Produkts können ohne vorherige Ankündigung geändert werden. Die ADDI-DATA GmbH behält sich das Recht vor, Änderungen bzgl. der technischen Daten und der hierin enthaltenen Materialien vorzunehmen.

Gewährleistung und Haftung
Der Nutzer ist nicht berechtigt, über die vorgesehene Nutzung des Produkts hinaus Änderungen des Werks vorzunehmen sowie in sonstiger Form in das Werk einzugreifen.
ADDI-DATA übernimmt keine Haftung bei offensichtlichen Druck- und Satzfehlern. Darüber hinaus übernimmt ADDI-DATA, soweit gesetzlich zulässig, weiterhin keine Haftung für Personen- und Sachschäden, die darauf zurückzuführen sind, dass der Nutzer das Produkt unsachgemäß installiert und/oder in Betrieb genommen oder bestimmungswidrig verwendet hat, etwa indem das Produkt trotz nicht funktionsfähiger Sicherheits- und Schutzvorrichtungen betrieben wird oder Hinweise in der Betriebsanleitung bzgl. Transport, Lagerung, Einbau, Inbetriebnahme, Betrieb, Grenzwerte usw. nicht beachtet werden. Die Haftung ist ferner ausgeschlossen, wenn der Betreiber das Produkt oder die Quellcode-Dateien unbefugt verändert und/oder die ständige Funktionsbereitschaft von Verschleißteilen vorwerfbar nicht überwacht wurde und dies zu einem Schaden geführt hat.

Urheberrecht
Dieses Handbuch, das nur für den Betreiber und dessen Personal bestimmt ist, ist urheberrechtlich geschützt. Die in der Betriebsanleitung und der sonstigen Produktinformation enthaltenen Hinweise dürfen vom Nutzer des Handbuchs weder vervielfältigt noch verbreitet und/oder Dritten zur Nutzung überlassen werden, soweit nicht die Rechtsübertragung im Rahmen der eingeräumten Produktlizenz gestattet ist. Zuwiderhandlungen können zivil- und strafrechtliche Folgen nach sich ziehen.

ADDI-DATA-Software-Produktlizenz
Bitte lesen Sie diese Lizenz sorgfältig durch, bevor Sie die Standardsoftware verwenden.
Das Recht zur Benutzung dieser Software wird dem Kunden nur dann gewährt, wenn er den Bedingungen dieser Lizenz zustimmt.
Die Software darf nur zur Einstellung der ADDI-DATA-Produkte verwendet werden.
Das Kopieren der Software ist verboten (außer zur Archivierung/Datensicherung und zum Austausch defekter Datenträger). Disassemblierung, Dekompilierung, Entschlüsselung und Reverse Engineering der Software ist verboten. Diese Lizenz und die Software können an eine dritte Partei übertragen werden, sofern diese Partei ein Produkt käuflich erworben hat, sich mit allen Bestimmungen in diesem Lizenzvertrag einverstanden erklärt und der ursprüngliche Besitzer keine Kopien der Software zurückhält.

Warenzeichen
- ADDI-DATA, APCI-1500, MSX-Box und MSX-E sind eingetragene Warenzeichen der ADDI-DATA GmbH.
- Turbo Pascal, Delphi, Borland C, Borland C++ sind eingetragene Warenzeichen der Borland Software Corporation.
- Microsoft .NET, Microsoft C, Visual C++, MS-DOS, Windows XP, Windows 7, Windows 8, Windows 10, Windows Server 2000, Windows Server 2003, Windows Embedded und Internet Explorer sind eingetragene Warenzeichen der Microsoft Corporation.
- LabVIEW, LabWindows/CVI, DASYLab, DIAdem sind eingetragene Warenzeichen der National Instruments Corporation.
- CompactPCI ist ein eingetragenes Warenzeichen der PCI Industrial Computer Manufacturers Group.
- VxWorks ist ein eingetragenes Warenzeichen von Wind River Systems, Inc.
- RTX ist ein eingetragenes Warenzeichen von IntervalZero.
- Mozilla Firefox ist ein eingetragenes Warenzeichen der Mozilla Foundation.
- SIMATIC S7, S7-300, STEP7, TIA Portal, CPU313C-2DP und CP343-1 Lean sind eingetragene Warenzeichen der Siemens AG.

Warnung!

Bei unsachgemäßem Einsatz und bestimmungswidrigem Gebrauch des Ethernet-Systems

können Personen verletzt werden

können Ethernet-System, PC und Peripherie beschädigt werden

kann die Umwelt verunreinigt werden.

- Schützen Sie sich, andere und die Umwelt!
- Lesen Sie unbedingt die Sicherheitshinweise (gelbe Broschüre)!

  Liegen Ihnen keine Sicherheitshinweise vor, so fordern Sie diese bitte an.
- Beachten Sie die Anweisungen dieses Handbuchs!

  Vergewissern Sie sich, dass Sie keinen Schritt vergessen oder übersprungen haben!
  Wir übernehmen keine Verantwortung für Schäden, die aus dem falschen Einsatz des Ethernet-Systems hervorgehen könnten.
- Beachten Sie folgende Symbole:

HINWEIS!
Kennzeichnet Anwendungstipps und andere nützliche Informationen.

ACHTUNG!
Bezeichnet eine möglicherweise gefährliche Situation.
Bei Nichtbeachten des Hinweises können Ethernet-System, PC und/oder Peripherie **zerstört** werden.

WARNUNG!
Bezeichnet eine möglicherweise gefährliche Situation.
Bei Nichtbeachten des Hinweises können Ethernet-System, PC und/oder Peripherie **zerstört** und Personen **gefährdet** werden.

Inhaltsverzeichnis

Warnung! 3
1 Dokumentation 5
1.1 Inhalt 5
1.2 Aufbau 5
2 Das Prinzip der Kombination 6
2.1 Leistungsfähigere SPS durch intelligente Ethernet-E/A-Systeme 6
2.2 Voraussetzungen hinsichtlich der Messelektronik des MSX-E-Systems 6
2.3 Die wesentlichen Vorteile der MSX-E-Systeme für die SPS 7
2.4 Anbindung eines MSX-E-Systems an eine SPS 8
2.5 Die Verbindung von MSX-E-System und SPS 9
2.5.1 Erfassungsstart ohne Programmierung (Autostart) 9
2.5.2 Erfassungsstart mit TCP/IP-Modbus 10
3 Anhang 11
3.1 Glossar 11
3.2 Index 13
4 Kontakt und Support 14

Abbildungen

Abb. 2-1: MSX-E-System: Aufbau 8
Abb. 2-2: MSX-E-Systeme: Anbindung an eine SPS 8
Abb. 2-3: SPS: Abfrage externer Messwerte 9

1 Dokumentation

1.1 Inhalt

Diese Dokumentation beschreibt, wie mit Hilfe intelligenter Ethernet-E/A-Systeme, Ethernet und eines Standardprotokolls wie z.B. TCP/IP komplexe und schnelle Messaufgaben oder Signalbearbeitungen (wie z.B. Mittelwertermittlung) einfach, zuverlässig und in wirtschaftlicher Weise via SPS lösbar sind.

Ausführliche Informationen zu den jeweiligen MSX-E-Systemen[1] finden Sie im ADDI-DATA-Produktkatalog und in den entsprechenden Handbüchern.

1.2 Aufbau

Die Dokumentation gliedert sich in drei Teile:

- **Teil 1: MSX-Exxxx und SPS: Grundlagen**
  Im ersten Teil der Dokumentation werden die Grundlagen der Kombination aus SPS und MSX-E-System beschrieben.
- **Teil 2: Anbindung an eine SPS SIMATIC S7**
  In Teil 2 werden folgende Fragen beantwortet:
  Welche Voraussetzungen sind nötig, um ein MSX-E-System an eine SPS anzuschließen (Hardware etc.)?
  Wie werden SPS und MSX-E-System konfiguriert?
  Welche Einstellungen sind generell vorzunehmen – und aus welchem Grund?
  Was ist bei der Anbindung eines (oder mehrerer) MSX-E-Systeme an eine SPS zu beachten?
- **Teil 3: ADDI-DATA Modbus TCP Client-Bibliothek für eine SPS SIMATIC S7**
  In Teil 4 wird beschrieben, wie die MSX-E-Systeme mit Hilfe einer Modbus TCP Client-Bibliothek direkt von einer SPS aus verwaltet und gesteuert werden können.

[1] **MSX-Exxxx** = gemeinsame Bezeichnung für die Ethernet-E/A-Systeme

2 Das Prinzip der Kombination

2.1 Leistungsfähigere SPS durch intelligente Ethernet-E/A-Systeme

Hohe Flexibilität und preiswerte Baugruppen haben dem SPS-Konzept zum Siegeszug verholfen. Neben der Steuerung und Regelung übernehmen SPSen zunehmend weitere Aufgaben wie Alarmfunktion, Visualisierung und Data-Logging. Sensoren und Aktoren werden längst nicht mehr diskret, sondern direkt bei verringertem Verdrahtungsaufwand mit der SPS verbunden. Ethernet-Netze haben sich in der Büro-IT längst bewährt. Die Verknüpfung zwischen Büro-IT und Fertigungs-IT kommt zugleich den Informations-, Kontroll- und Sicherheitsanforderungen modernen Managements entgegen.

Durch die serielle Arbeitsweise sind speicherprogrammierbare Steuerungen zwar langsamer als verbindungsprogrammierte Steuerungen (VPS). Doch anspruchsvolle Aufgaben wie das zeitgleiche Abtasten mehrerer Kanäle oder die gleichzeitige Erfassung mehrerer Signaltypen, die von induktiven/ digitalen Tastern, Drehgebern oder analogen Eingängen kommen, werden mit Hilfe leistungsfähiger Messelektronik gelöst. Beispiele dafür sind das Erfassen vieler Messwerte in einem Puffer, um die Daten später zur Verfügung zu stellen, oder das sehr schnelle Erfassen und Berechnen von Werten unabhängig vom SPS Zyklus.

2.2 Voraussetzungen hinsichtlich der Messelektronik des MSX-E-Systems

- Gleichzeitiges Erfassen verschiedener Messwerte (z.B. Oberflächengüte und Position)
  - ➢ Erlaubt positionsgenaue Ausbesserungen an den erkannten Fehlerstellen
- Berechnung von Minimal- und Maximalwerten bzw. Mittelwerten
  - ➢ um z.B. Unebenheiten innerhalb oder außerhalb einer vordefinierten Toleranz zu kategorisieren bzw. die SPS zu entlasten
- Direkter Einsatz in der Fertigungshalle nah am Prüfling
- Zuverlässiges Arbeiten bei hoher Temperaturbelastung und Strahlwasser
- Einfache Anbindung an die SPS
- Kommunikation via Standard-Ethernet
  - ➢ Weiterleitung der Werte auch an die IT-Ebene für eine zeitnahe Auswertung der Produktions-qualität
- Konfiguration ohne Programmierung, z.B. über eine integrierte Weboberfläche
  - ➢ Integration so einfach wie möglich gestalten
- Diagnostik und Monitoring ohne Sonderprogramme oder SPS
- Onboard-Zwischenspeicher, um keine Werte zu verlieren

2.3 Die wesentlichen Vorteile der MSX-E-Systeme für die SPS

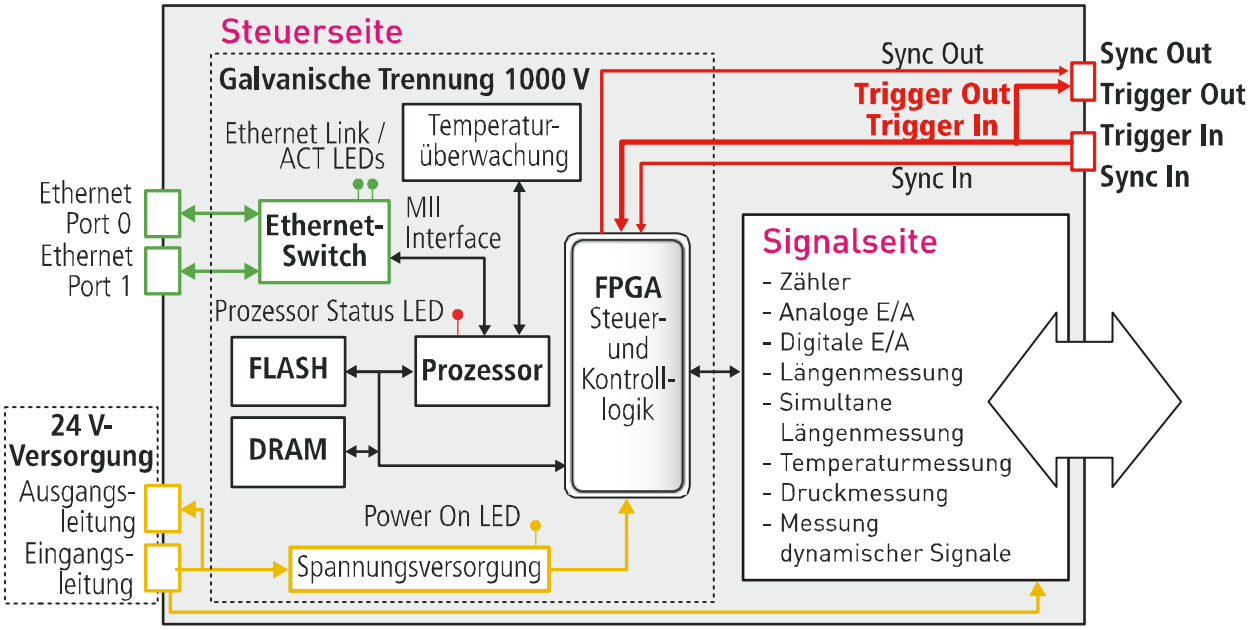
- Kaskadierbar und im µs-Bereich synchronisierbar
  - ➢ Gleichzeitige Erfassung verschiedener Messwerte, z.B. induktive Längenmesstaster und Position des Prüflings
- Erweiterter Betriebstemperaturbereich von -40 °C bis + 85 °C
- Schutzart IP 65: Schutz gegen Strahlwasser und Staub
  - ➢ Kein „Schutz“ durch Schaltschränke notwendig; Geld- und Platzersparnis
- Zahlreiche Schutzbeschaltungen: galvanische Trennung bis 1000 V, Kurzschluss- und Verpolungsschutz
- Weboberfläche zur Konfiguration, Steuerung und Überwachung der Erfassung
  - ➢ Messwerte direkt auf den Bildschirm des Techniker-PCs
  - ➢ Zeitersparnis, insbesondere bei der Messstelleneinrichtung, und Flexibilität
- Einfache Anbindung an die SPS über TCP/IP
  - ➢ Problemlose Weiterleitung der Messdaten an die IT-Ebene
  - ➢ Fernwartung mit Passwortschutz und Verschlüsselung
- On-Board-RAM für die Datenspeicherung
- Installation einfacher Software-Updates auf die MSX-E-Systeme jederzeit möglich
- Durchführung komplexer Rechenaufgaben (z.B. Verrechnung mehrerer Sensoren zu einem Ergebnis) bei Bedarf realisierbar

Die MSX-E-Systeme sind für unterschiedliche Signaltypen erhältlich:

- Digitale E/A, 24 V
- Analoge Eingänge, Spannung oder Strom, 16 Bit
- Analoge Ausgänge, Spannung oder Strom, 16 Bit
- Induktive Längenmesstaster (HB, LVDT, Mahr-kompatibel)
- Inkrementaldrehgeber und -taster
- Sin/Cos-Signale (1 V_{ss} / 11 μA_{ss}) für Drehgeber oder Digitaltaster
- etc.

Die Funktionsweise ist bei allen MSX-E-Systemen identisch, d.h., jedes System verfügt über eine Steuerseite und eine Signalseite. Die Anbindung an die Steuerseite ist für alle Systeme gleich.

Abb. 2-1: MSX-E-System: Aufbau

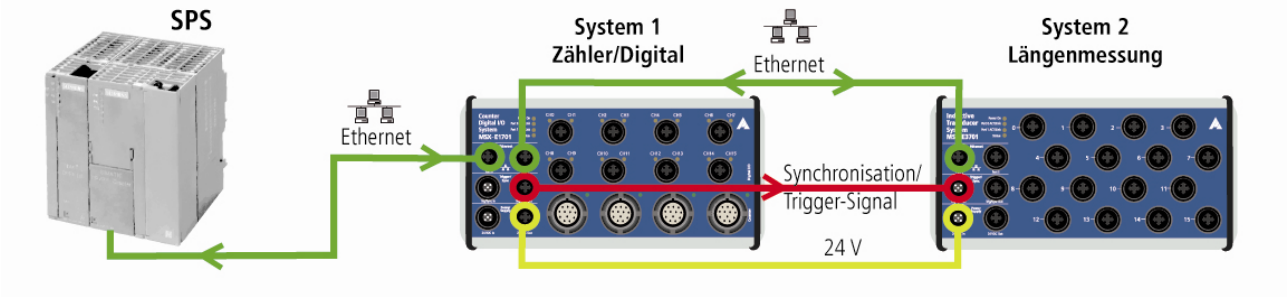


2.4 Anbindung eines MSX-E-Systems an eine SPS

Die MSX-E-Systeme werden per Ethernet an die SPS angebunden, wodurch keine speziellen Leitungen oder Kommunikationsbusse nötig sind. Weitere Systeme können dem Netzwerk auf einfache Weise hinzugefügt werden.

Mit Hilfe eines Ethernet-Controllers, der das TCP/IP-Protokoll verwaltet, kann die SPS auf die zusätzlichen Systeme zugreifen. Hierfür gibt es sowohl Hardwarelösungen (Kommunikationsprozessor, CP) als auch Softwarelösungen, wobei sich je nach Applikation eine Hardwarelösung besser eignet. Durch einen separaten Hardware-Controller wird die Kommunikation schneller und belastet die CPU der SPS weniger als die softwaremäßige Ethernet-Verwaltung über die CPU der SPS.

Abb. 2-2: MSX-E-Systeme: Anbindung an eine SPS



2.5 Die Verbindung von MSX-E-System und SPS

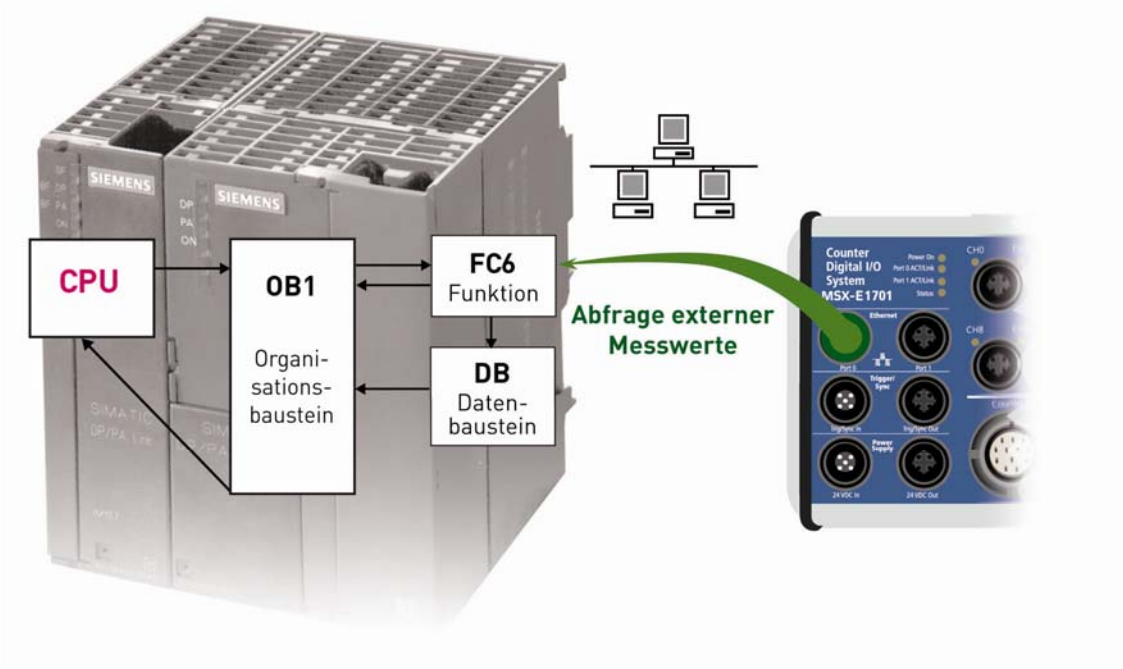
2.5.1 Erfassungsstart ohne Programmierung (Autostart)

Die eigentliche Erfassung wird auf den MSX-E-Systemen über die Weboberfläche konfiguriert und gespeichert. Über die Autostart-Funktion ist es möglich, die Einstellungen der Erfassung beim Booten des MSX-E-Systems zu laden und automatisch auszuführen. Insofern entfällt der zusätzliche Programmieraufwand für die SPS.

Beispiel: Es sollen 16 analoge Eingänge (8 x ± 10 V, 8 x ± 5 V) mit einer Geschwindigkeit von je 25 kHz für insgesamt 0,5 Sekunden erfasst werden. Die Erfassung soll erst dann gestartet werden, wenn der Prüfling in der richtigen Position ist, d.h., die SPS signalisiert dem MSX-E-System den Start der Erfassung über einen digitalen 24 V-Ausgang (Hardware-Trigger).
Wurde die Erfassung konfiguriert und auf dem MSX-E-System gespeichert, dann bootet das System nach dem Anschalten und wartet auf den Hardware-Trigger. Wenn der Hardware-Trigger erkannt wurde, startet die Erfassung und das System erfasst 0,5 Sekunden lang die Messwerte – anschließend wartet es auf den nächsten Trigger.

Über die Empfangsfunktion (FC6) kann die SPS auf die Werte zugreifen und diese in einem Datenbaustein (DB) abspeichern (1 DWORD pro Kanal). Anschließend können die Werte beliebig weiterverarbeitet werden. Die Systeme sind aufgrund ihrer eigenen Intelligenz (ARM9) in der Lage, Berechnungen wie z.B. die des Minimal-, Maximal- und Mittelwerts selbständig durchzuführen.

Abb. 2-3: SPS: Abfrage externer Messwerte



Eine genaue Beschreibung der Konfiguration von SPS und MSX-E-Systemen befindet sich in **Teil 2** der **Praktischen Anleitung** der Ethernet-E/A-Systeme.

2.5.2 Erfassungsstart mit TCP/IP-Modbus

Die Erfassung der Messwerte kann auch über TCP/IP-Modbus gestartet werden. Hierbei wird die Datenerfassung nicht auf den MSX-E-Systemen konfiguriert und beim Hochfahren geladen, sondern über Modbus-Funktionen in der SPS aufgerufen.

Nähere Informationen hierzu finden Sie in **Teil 3** der **Praktischen Anleitung** der Ethernet-E/A-Systeme sowie im Dokument **MODBUS Interface Description** auf der MSX-E-CD (Datei „msxexxxx_modbus.html“ im Verzeichnis „\[MSX-E-System]\Modbus\Documentation\html“).

3 Anhang

3.1 Glossar

Ethernet
Hierbei handelt es sich um ein Basisband-Bussystem, das ursprünglich für die Verknüpfung von Minicomputern entwickelt wurde. Es basiert auf dem CSMA/CD-Zugriffsverfahren. Als Übertragungsmedium dienen Koaxialkabel bzw. Twisted-Pair-Leitungen.
Die Übertragungsgeschwindigkeiten betragen 10 Mbit/s (Ethernet), 100 Mbit/s (Fast Ethernet) sowie 1 Gbit/s bzw. 10 Gbit/s (Gigabit-Ethernet). Diese weit verbreitete Technik zum Vernetzen von Rechnern in einem LAN ist seit 1985 genormt (IEEE 802.3 und ISO 8802-3).
Die Ethernet-Technologie hat sich im Bürobereich allgemein durchgesetzt. Nach Ermöglichung auch sehr harter Echtzeitanforderungen und Anpassung der Gerätetechnik (Buskabel, Patchfelder, Anschlussdosen) an die rauen Einsatzbedingungen des industriellen Umfelds dringt sie zunehmend in die Feldbereiche der Automatisierungstechnik vor.

IP-Adresse
Die IP-Adresse ist eine numerische Adresse, die einem Rechner im Internet zugeordnet ist und diesen eindeutig identifizierbar macht.
Sie besteht aus einer Folge von vier Zahlengruppen mit je maximal drei Ziffern, die jeweils durch Punkte voneinander getrennt sind. Fest mit dem Internet verbundene Rechner haben eine feste IP-Adresse. Benutzer, die sich von Zeit zu Zeit ins Netz einwählen, erhalten vom Provider eine dynamische IP-Adresse, die nach Beendigung der Sitzung an den nächsten Client weitervergeben wird.

LSB
= Least Significant Bit

Die Bits sind unterschiedlich gewichtet, wobei das am weitesten rechts stehende die geringste Wertigkeit hat. Es wird daher als „Least Significant Bit“ (LSB) bezeichnet. Das erste Bit wird als „Most Significant Bit“ (MSB) bezeichnet, da es den höchsten Wert hat.

Modbus TCP
Das Modbus-TCP-Protokoll ist ein offenes Anfrage/Antwort-Protokoll für die Kommunikation zwischen Master und Slave bzw. Client und Server. Als Master, der die Kommunikationsvorgänge initialisiert, kann z.B. eine SPS fungieren. Die Daten werden in Form von TCP/IP-Paketen übertragen.

MSB
= Most Significant Bit

Die Bits sind unterschiedlich gewichtet. Das erste Bit hat den höchsten Wert und wird als „Most Significant Bit“ (MSB) bezeichnet.

PLC
= Programmable Logic Controller

PLC ist die englische Bezeichnung für SPS.

Socket
Die bidirektionale Softwareschnittstelle zur Interprozess- (IPC) oder Netzwerkkommunikation wird Socket genannt. Sockets bilden eine standardisierte Schnittstelle (API) zwischen dem Betriebssystem und der eigentlichen Anwendungssoftware.

SPS
= Speicherprogrammierbare Steuerung

Die SPS ist ein rechnerbasiertes Steuergerät, dessen Funktionalität durch ein Anwenderprogramm festgelegt wird. Mittels genormter Fachsprachen ist dieses Anwenderprogramm relativ einfach zu erstellen. Aufgrund der seriellen Arbeitsweise sind die Reaktionszeiten der SPS langsamer als bei VPS. Als Gerätefamilien mit abgestuften und leistungsmäßig aufeinander abgestimmten Komponenten beherrschen die SPS heute alle Ebenen einer Automatisierungshierarchie.

Trigger
Der Trigger ist ein Impuls oder Signal zum Starten oder Stoppen einer besonderen Aufgabe. Trigger werden häufig zur Steuerung des Datenerfassungsbetriebs eingesetzt.

Zähler
Der Zähler ist ein Schaltkreis, der Impulse zählt oder die Dauer von Impulsen messen kann.

3.2 Index

Anbindung an SPS 8
Erfassungsstart
  Autostart 9
  TCP/IP-Modbus 10
Glossar 11
Signaltypen 7
Voraussetzungen 6
Vorteile 7

4 Kontakt und Support

Haben Sie Fragen? Schreiben Sie uns oder rufen Sie uns an:

Postanschrift: ADDI-DATA GmbH
Airpark Business Center
Airport Boulevard B210
77836 Rheinmünster
Deutschland

Telefon: +49 7229 1847-0
Fax: +49 7229 1847-222

E-Mail: info@addi-data.com

Handbuch- und Software-Download im Internet:

www.addi-data.de

ADDI-DATA®
PARTNER FÜR PRÄZISION

Praktische Anleitung

Intelligente Ethernet-E/A-Systeme (Teil 2)

Anbindung an eine SPS SIMATIC® S7®

DIN EN ISO 9001:2015 zertifiziert

Ausgabe: 02.05-09/2018

Produktinformation
Dieses Handbuch enthält die technischen Anlagen, wichtige Anleitungen zur korrekten Inbetriebnahme und Nutzung sowie Produktinformation entsprechend dem aktuellen Stand vor der Drucklegung.
Der Inhalt dieses Handbuchs und die technischen Daten des Produkts können ohne vorherige Ankündigung geändert werden. Die ADDI-DATA GmbH behält sich das Recht vor, Änderungen bzgl. der technischen Daten und der hierin enthaltenen Materialien vorzunehmen.

Gewährleistung und Haftung
Der Nutzer ist nicht berechtigt, über die vorgesehene Nutzung des Produkts hinaus Änderungen des Werks vorzunehmen sowie in sonstiger Form in das Werk einzugreifen.
ADDI-DATA übernimmt keine Haftung bei offensichtlichen Druck- und Satzfehlern. Darüber hinaus übernimmt ADDI-DATA, soweit gesetzlich zulässig, weiterhin keine Haftung für Personen- und Sachschäden, die darauf zurückzuführen sind, dass der Nutzer das Produkt unsachgemäß installiert und/oder in Betrieb genommen oder bestimmungswidrig verwendet hat, etwa indem das Produkt trotz nicht funktionsfähiger Sicherheits- und Schutzvorrichtungen betrieben wird oder Hinweise in der Betriebsanleitung bzgl. Transport, Lagerung, Einbau, Inbetriebnahme, Betrieb, Grenzwerte usw. nicht beachtet werden. Die Haftung ist ferner ausgeschlossen, wenn der Betreiber das Produkt oder die Quellcode-Dateien unbefugt verändert und/oder die ständige Funktionsbereitschaft von Verschleißteilen vorwerfbar nicht überwacht wurde und dies zu einem Schaden geführt hat.

Urheberrecht
Dieses Handbuch, das nur für den Betreiber und dessen Personal bestimmt ist, ist urheberrechtlich geschützt. Die in der Betriebsanleitung und der sonstigen Produktinformation enthaltenen Hinweise dürfen vom Nutzer des Handbuchs weder vervielfältigt noch verbreitet und/oder Dritten zur Nutzung überlassen werden, soweit nicht die Rechtsübertragung im Rahmen der eingeräumten Produktlizenz gestattet ist. Zuwiderhandlungen können zivil- und strafrechtliche Folgen nach sich ziehen.

ADDI-DATA-Software-Produktlizenz
Bitte lesen Sie diese Lizenz sorgfältig durch, bevor Sie die Standardsoftware verwenden.
Das Recht zur Benutzung dieser Software wird dem Kunden nur dann gewährt, wenn er den Bedingungen dieser Lizenz zustimmt.
Die Software darf nur zur Einstellung der ADDI-DATA-Produkte verwendet werden.
Das Kopieren der Software ist verboten (außer zur Archivierung/Datensicherung und zum Austausch defekter Datenträger). Disassemblierung, Dekompilierung, Entschlüsselung und Reverse Engineering der Software ist verboten. Diese Lizenz und die Software können an eine dritte Partei übertragen werden, sofern diese Partei ein Produkt käuflich erworben hat, sich mit allen Bestimmungen in diesem Lizenzvertrag einverstanden erklärt und der ursprüngliche Besitzer keine Kopien der Software zurückhält.

Warenzeichen
- ADDI-DATA, APCI-1500, MSX-Box und MSX-E sind eingetragene Warenzeichen der ADDI-DATA GmbH.
- Turbo Pascal, Delphi, Borland C, Borland C++ sind eingetragene Warenzeichen der Borland Software Corporation.
- Microsoft .NET, Microsoft C, Visual C++, MS-DOS, Windows XP, Windows 7, Windows 8, Windows 10, Windows Server 2000, Windows Server 2003, Windows Embedded und Internet Explorer sind eingetragene Warenzeichen der Microsoft Corporation.
- LabVIEW, LabWindows/CVI, DASYLab, DIAdem sind eingetragene Warenzeichen der National Instruments Corporation.
- CompactPCI ist ein eingetragenes Warenzeichen der PCI Industrial Computer Manufacturers Group.
- VxWorks ist ein eingetragenes Warenzeichen von Wind River Systems, Inc.
- RTX ist ein eingetragenes Warenzeichen von IntervalZero.
- Mozilla Firefox ist ein eingetragenes Warenzeichen der Mozilla Foundation.
- SIMATIC S7, S7-300, STEP7, TIA Portal, CPU313C-2DP und CP343-1 Lean sind eingetragene Warenzeichen der Siemens AG.

Warnung!

Bei unsachgemäßem Einsatz und bestimmungswidrigem Gebrauch des Ethernet-Systems

können Personen verletzt werden

können Ethernet-System, PC und Peripherie beschädigt werden

kann die Umwelt verunreinigt werden.

- Schützen Sie sich, andere und die Umwelt!
- Lesen Sie unbedingt die Sicherheitshinweise (gelbe Broschüre)!

  Liegen Ihnen keine Sicherheitshinweise vor, so fordern Sie diese bitte an.
- Beachten Sie die Anweisungen dieses Handbuchs!

  Vergewissern Sie sich, dass Sie keinen Schritt vergessen oder übersprungen haben!
  Wir übernehmen keine Verantwortung für Schäden, die aus dem falschen Einsatz des Ethernet-Systems hervorgehen könnten.
- Beachten Sie folgende Symbole:

HINWEIS!
Kennzeichnet Anwendungstipps und andere nützliche Informationen.

ACHTUNG!
Bezeichnet eine möglicherweise gefährliche Situation.
Bei Nichtbeachten des Hinweises können Ethernet-System, PC und/oder Peripherie **zerstört** werden.

WARNUNG!
Bezeichnet eine möglicherweise gefährliche Situation.
Bei Nichtbeachten des Hinweises können Ethernet-System, PC und/oder Peripherie **zerstört** und Personen **gefährdet** werden.

Inhaltsverzeichnis

Warnung! 3
Kapitelübersicht 6
1 Konfiguration von MSX-E-System und SPS 7
1.1 Voraussetzung 7
1.2 Prinzipieller Aufbau 7
1.3 Programmübersicht 8
1.4 Datenkonvertierung 8
1.4.1 Little-Endian-Format (Intel-Format) 9
1.4.2 Big-Endian-Format (Motorola-Format) 9
2 Parametrierung der SPS 10
2.1 Software 10
2.2 S7-Ethernet-Konfiguration 10
3 Lesen der Messdaten 15
3.1 Voraussetzungen 15
3.2 S7: Beschreibung der Funktionsbausteine 15
3.2.1 FC1: Konvertierung von Daten in das Motorola-Format 16
3.2.2 FC3: Konvertierung eines DWORD-Datentyps in einen REAL-Datentyp 17
3.2.3 FC4: Konvertierung in Millimeter (für Messtaster) 17
3.2.4 FC7: Konvertierung in Volt bzw. mA 19
3.2.5 FC20: Dynamisches Lesen eines DWORD-Werts aus einer Tabelle 20
3.2.6 FC21: Dynamisches Schreiben eines REAL-Werts in eine Tabelle 21
3.2.7 FC6: Lesen von Daten aus einer Netzwerkverbindung 21
4 Einfaches S7-Beispiel 24
4.1 Konfiguration des MSX-E-Systems über die Weboberfläche 24
4.1.1 Auto-Refresh-Modus 24
4.1.2 Sequenz-Modus 26
4.2 S7: Beschreibung des Beispiels 29
4.2.1 OB1: Hauptbaustein 29
4.2.2 Netzwerk 1: Definition der Anzahl der zu lesenden Werte 29
4.2.3 Netzwerk 2: Auswahl der passenden Konvertierungseinheit 30
4.2.4 Netzwerk 3: Lesen der Messwerte aus der TCP-Verbindung 30
4.2.5 Netzwerk 4: Initialisierung des Index 31
4.2.6 Netzwerk 5: Darstellung im Motorola-Format 32
4.2.7 Netzwerk 6: Verwaltung des Sequenz-Zähler-Werts 32
4.2.8 Netzwerk 7: Typecast statt Konvertierung 33
4.2.9 Netzwerk 8: Konvertierung in Millimeter 33
4.2.10 Netzwerk 9: Konvertierung in Volt 34
4.2.11 Netzwerk 10: Bearbeitung des Sequenz-Zähler-Werts 34
4.2.12 Netzwerk 11: Speichern und Prüfen 35
4.3 TIA Software-Umgebung mit VIPA CPU 015 36
5 Empfehlungen 38
5.1 Auto-Refresh-Modus und Sequenz-Modus 38
5.1.1 Erfassung im Auto-Refresh-Modus 38
5.1.2 Erfassung im Sequenz-Modus 39
5.2 Trigger 40
5.2.1 Systemkonfiguration 40
5.2.2 BasicSample: Beispiel für die Trigger-Erweiterung 40
5.3 Abfrage der aktuellen Messwerte im Auto-Refresh-Modus 41
5.3.1 Systemkonfiguration 41
5.3.2 BasicSample: Anpassungsbeispiel zur Abfrage der Messwerte 41
5.4 Anschließen mehrerer Systeme an eine SPS 42
6 Anhang 43

6.1 Glossar 43
6.2 Index 45
7 Kontakt und Support 46

Abbildungen

Abb. 1-1: MSX-E-System und SPS: Prinzipieller Aufbau 7
Abb. 1-2: Abfrage externer Messwerte 8
Abb. 2-1: SIMATIC Manager: S7-Ethernet-Konfiguration 10
Abb. 2-2: Ethernet-Konfiguration: TCP-Verbindung 11
Abb. 2-3: Eigenschaften – TCP-Verbindung: Allgemein 12
Abb. 2-4: Eigenschaften – TCP-Verbindung: Adressen 13
Abb. 2-5: Login-Fenster 13
Abb. 2-6: Data server: Configuration 14
Abb. 2-7: Data server: Set and save 14
Abb. 2-8: Data server: Blocking transfer 14
Abb. 3-1: SIMATIC-Manager: Auflistung der Bausteine 15
Abb. 3-2: Eigenschaften – TCP-Verbindung: Eingabe-Parameter 22
Abb. 3-3: Acquisition: Data server frame format 23
Abb. 3-4: Datenbaustein mit der Tabelle RAW 23
Abb. 3-5: Tabelle RAW im Datenbaustein 23
Abb. 5-1: Auto-Refresh-Modus: Erfassungszyklus 38
Abb. 5-2: Sequenz-Modus: Erfassungszyklus 39
Abb. 5-3: SIMATIC Manager: S7-Ethernet-Konfiguration 42

Tabellen

Tabelle 3-1: FC1: Reihenfolge der Funktionsaufrufe 16
Tabelle 3-2: FC3: Reihenfolge der Funktionsaufrufe 17
Tabelle 3-3: FC4: Reihenfolge der Funktionsaufrufe 18
Tabelle 3-4: FC7: Reihenfolge der Funktionsaufrufe 20

Kapitelübersicht

In dieser Anleitung finden Sie folgende Informationen:

Kapitel	Inhalt
1	Grundlagen der Konfiguration von MSX-E-System und SPS
2	Beschreibung der SPS-Parametrierung bzw. Ethernet-Konfiguration
3	Erläuterung der S7-Funktionsbausteine
4	Beschreibung der Netzwerke des S7-Beispiels
5	Empfehlungen bzgl. Erfassungsmodus, Trigger etc.
6	Anhang mit Glossar und Index
7	Kontakt- und Support-Adresse

1 Konfiguration von MSX-E-System und SPS

1.1 Voraussetzung

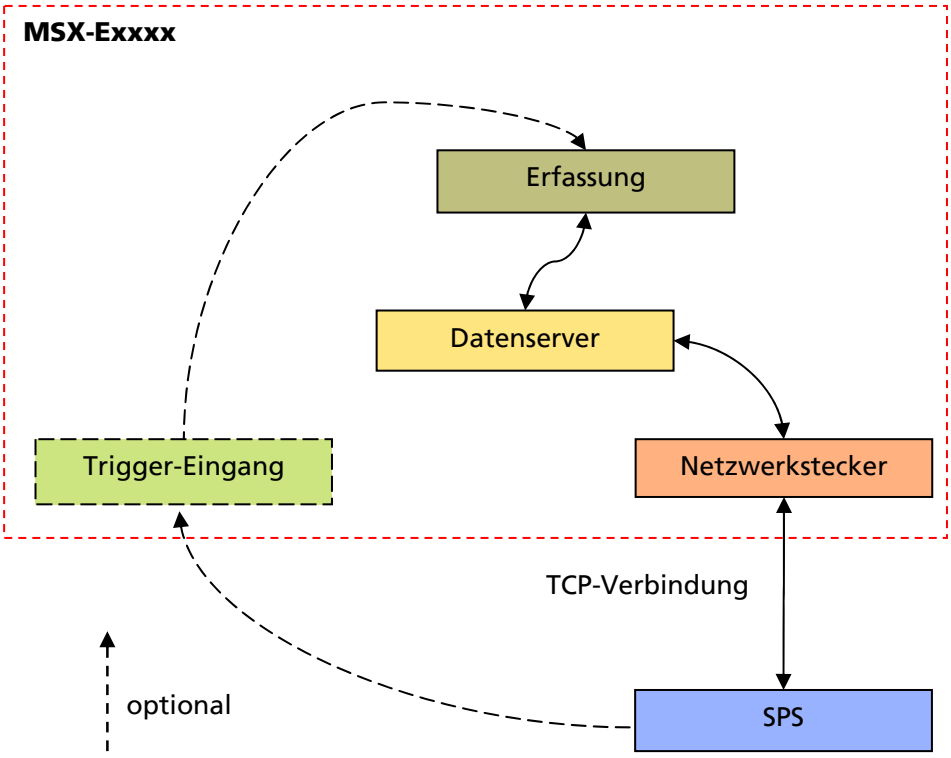
Voraussetzung für die Verwendung von MSX-E-Systemen mit einer SPS ist, dass die SPS über eine Ethernet-Schnittstelle verfügt und eine aktive TCP-Verbindung (TCP-Sockets) öffnen kann.

1.2 Prinzipieller Aufbau

Jedes MSX-E-System verfügt über einen Datenserver, der die erfassten Messwerte zur Verfügung stellt. Über eine TCP-Verbindung kann die SPS auf diese Messwerte zugreifen und somit die Daten lesen.

Die Erfassung selbst kann je nach Applikation entweder durch einen externen Trigger (digitaler 24 V-Eingang) oder automatisch gestartet werden.

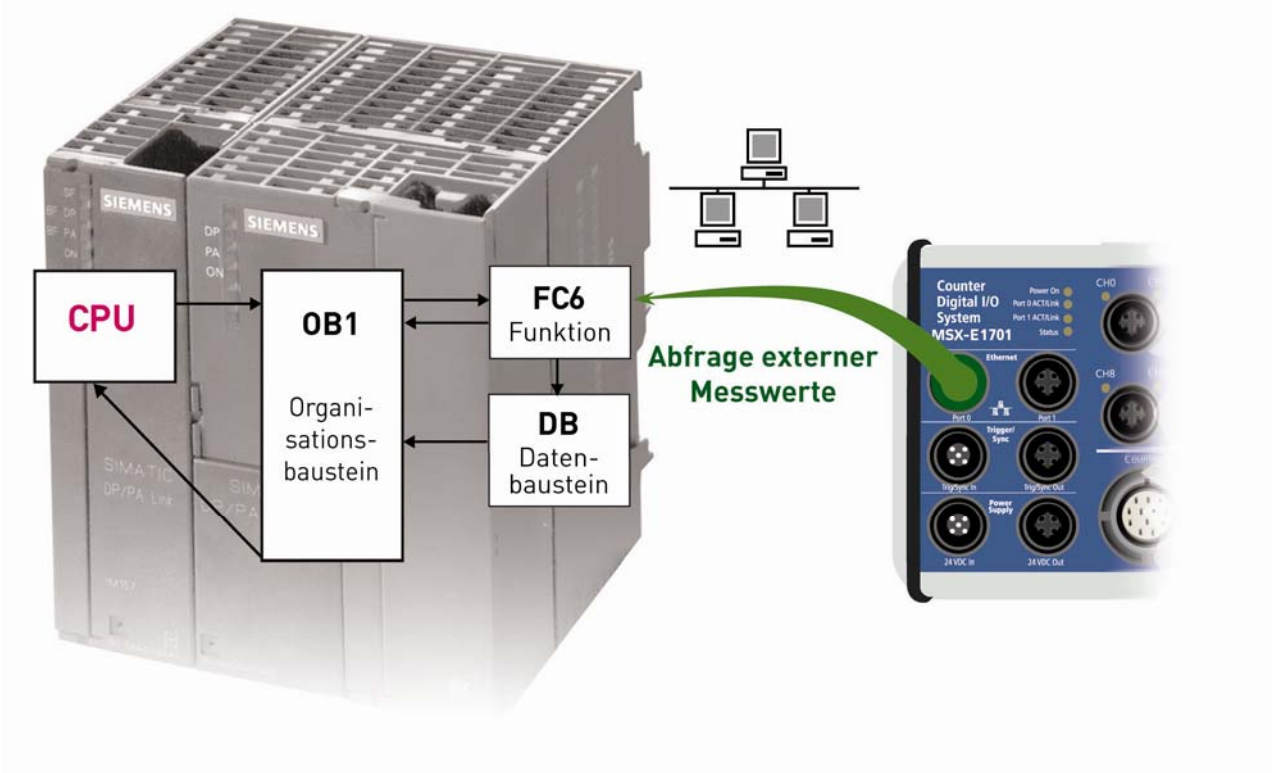
Abb. 1-1: MSX-E-System und SPS: Prinzipieller Aufbau



In der SPS werden die Daten mit der Funktion FC6 (Empfangsfunktion) gelesen und danach in einem Datenbaustein (DB) abgelegt. Von dort aus können sie beliebig weiterverarbeitet werden.

1.3 Programmübersicht

Abb. 1-2: Abfrage externer Messwerte



OB1: Dieser Organisationsbaustein ruft die Empfangsfunktion (FC6) auf.

FC6: Mit dieser Funktion werden Daten aus einer TCP-Verbindung (Socket) gelesen und anschließend in einem Datenbaustein gespeichert.

1.4 Datenkonvertierung

Es ist zu beachten, dass die MSX-E-Systeme Daten aus dem Datenserver im Little-Endian-Format (Intel-Format) senden, wohingegen eine **S7-SPS** mit dem Big-Endian-Format (Motorola-Format) arbeitet.
Somit müssen die Werte der **MSX-E-Systeme** auf der SPS in das Big-Endian-Format (Motorola-Format) konvertiert werden. Diese Datencodierung spielt eine wichtige Rolle, da sonst die Messergebnisse falsch interpretiert werden. Eine entsprechende Konvertierungsfunktion wird von ADDI-DATA zur Verfügung gestellt:

- **Funktion:** FC1
- **Name:** INTEL_TO_MOTOROLA

1.4.1 Little-Endian-Format (Intel-Format)

Das niedrigstwertige Bit (LSB) wird zuerst übertragen:

16-Bit	0x1234	>>	0x34	0x12		
32-Bit	0x12345678	>>	0x78	0x56	0x34	0x12

1.4.2 Big-Endian-Format (Motorola-Format)

Das höchstwertige Bit (MSB) wird zuerst übertragen:

16-Bit	0x1234	>>	0x12	0x34		
32-Bit	0x12345678	>>	0x12	0x34	0x56	0x78

2 Parametrierung der SPS

2.1 Software

ADDI-DATA liefert Ihnen ein Softwarepaket, das die Verbindung zwischen einem MSX-E-System und einer S7-SPS zeigt. Das Programmierbeispiel basiert auf einer SIMATIC **S7** SPS von Siemens.

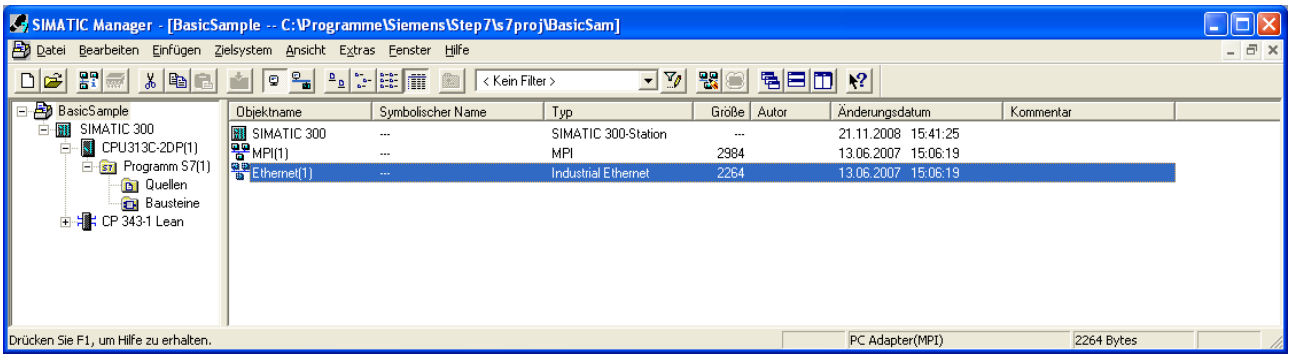
Wenn Sie eine **S7**-kompatible SPS eines anderen Herstellers (z.B. IBHsoftec, VIPA etc.) einsetzen möchten, müssen Sie die Original-FCs des jeweiligen Herstellers verwenden. Eventuell sind auch die Datenbausteine etc. anzupassen (Siemens: 9 DWORD; VIPA: 36 Byte).

Das erwähnte Programmierbeispiel für die **S7** von Siemens heißt **BasicSample** und wurde für die Ethernet-Kommunikation der **S7-300** mit **CP343-1** entwickelt. Es dient dem Lesen von Werten aus der TCP-Verbindung (TCP-Socket).

2.2 S7-Ethernet-Konfiguration

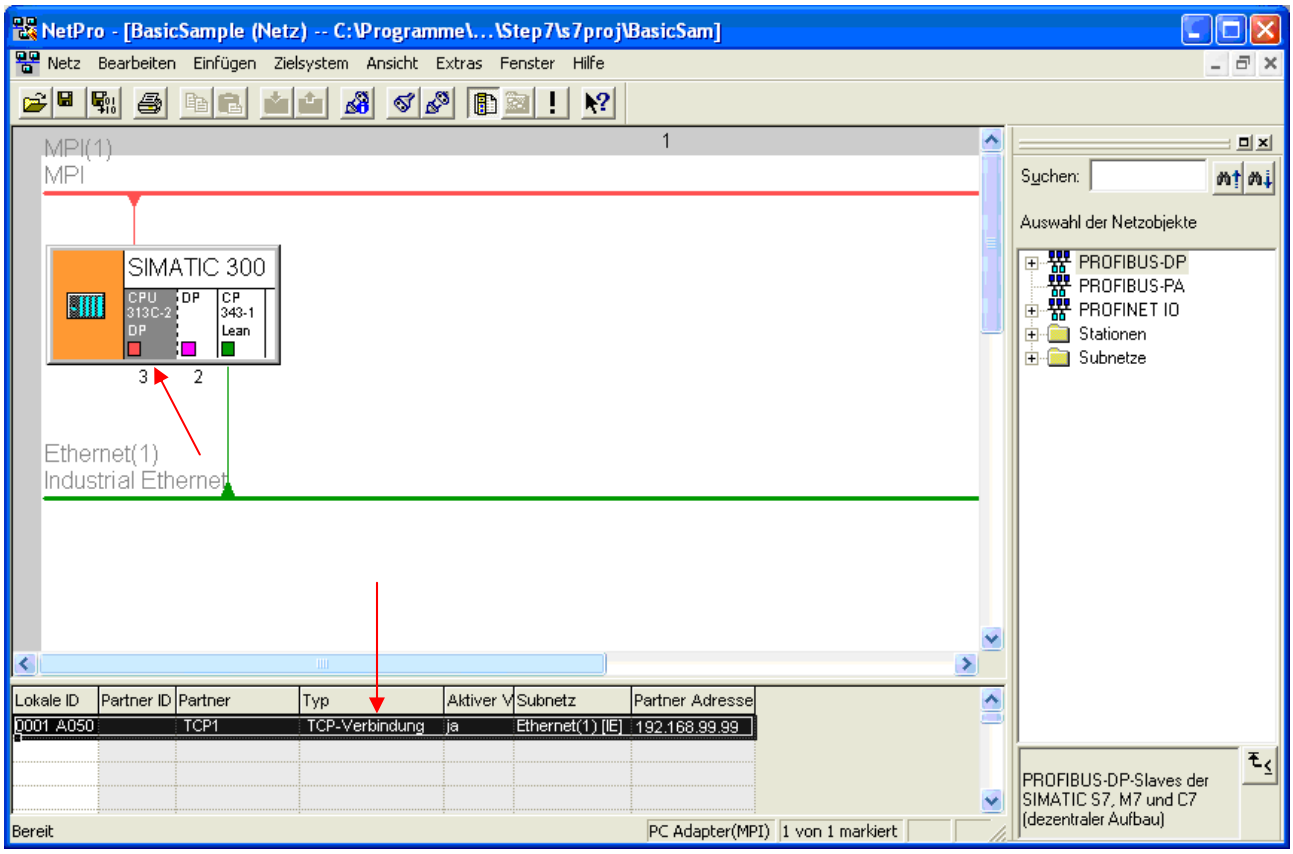
Die nachfolgende Konfiguration basiert auf einer **S7-300** mit **CP343-1**. Eventuell werden bei Ihrer SPS andere Menüs angezeigt, aber die Einstellungen sind dieselben.

Abb. 2-1: SIMATIC Manager: S7-Ethernet-Konfiguration



- Öffnen Sie den SIMATIC-Manager und anschließend das Projekt **BasicSample**.
- Zur Konfiguration der TCP-Verbindung doppelklicken Sie auf den Objektnamen **Ethernet**.

Abb. 2-2: Ethernet-Konfiguration: TCP-Verbindung

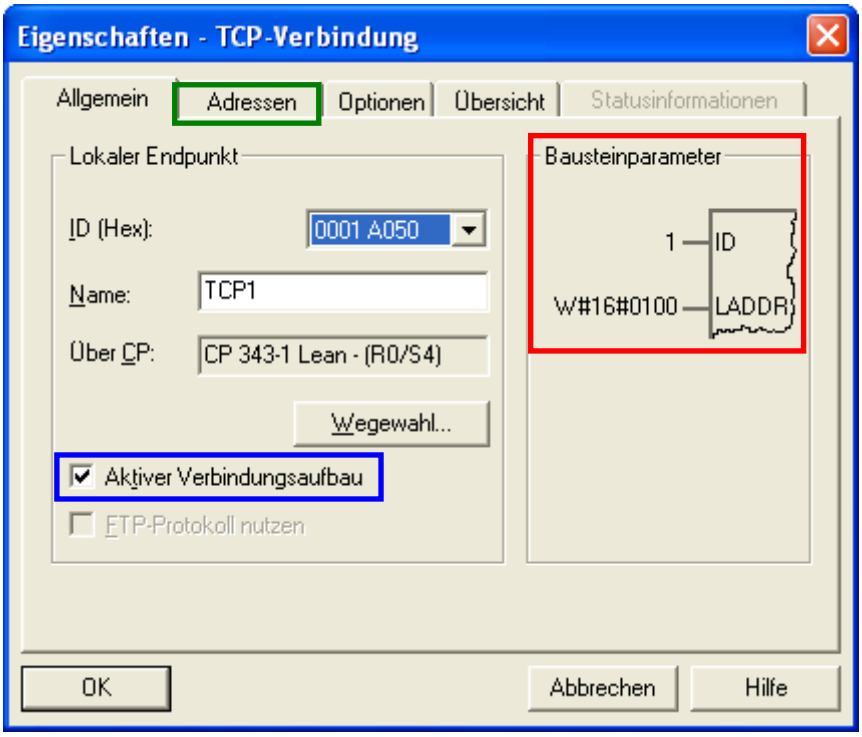


- Klicken Sie auf **CPU313C-2DP** (siehe Pfeil oben).
- Doppelklicken Sie in der Tabelle unten auf die markierte Zeile (siehe Pfeil unten).

In der Tabellenspalte **Typ** steht, dass die Ethernet-Verbindung eine **TCP-Verbindung** ist.

Danach wird folgendes Fenster angezeigt:

Abb. 2-3: Eigenschaften – TCP-Verbindung: Allgemein

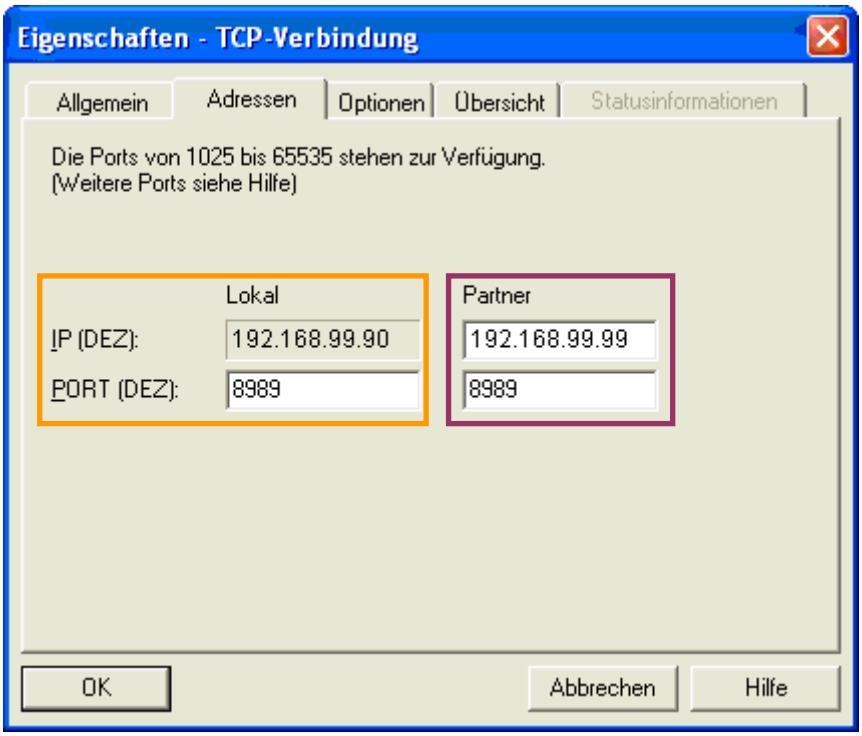


Damit die **S7** die Verbindung zum MSX-E-System herstellt, muss **Aktiver Verbindungsaufbau** aktiviert sein.

Rechts im Fenster sehen Sie die **Bausteinparameter**, die zusammen mit der Funktion „FC6 (AG_RECV)“ verwendet werden müssen. Die Verbindungs-ID (Socket-ID) und die Adresse sind angegeben.

- Klicken Sie auf den Registerreiter **Adressen**.

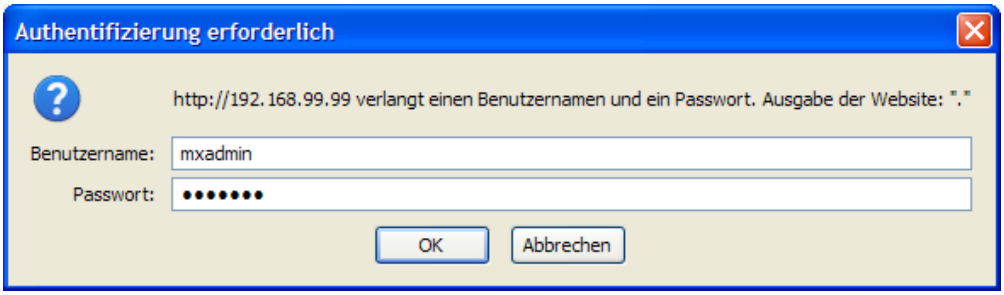
Abb. 2-4: Eigenschaften – TCP-Verbindung: Adressen



- Geben Sie im Feld **PORT** im Bereich **Lokal** die Portnummer der **S7** ein. Diese muss der Portnummer des MSX-E-Systems entsprechen.
- Geben Sie im Feld **IP** im Bereich **Partner** die IP-Adresse des **MSX-E-Systems** ein sowie im Feld **Port** die Portnummer des MSX-E-Systems. Diese muss mit der Portnummer übereinstimmen, die auf der Weboberfläche des MSX-E-Systems festgelegt wurde.
- Um die Portnummer des MSX-E-Systems zu überprüfen, öffnen Sie einen Webbrowser (z.B. Mozilla Firefox, Internet Explorer etc.) und geben Sie folgende Adresse ein: „http://[IP-Adresse des MSX-E-Systems]“.

Ein Login-Fenster wird angezeigt:

Abb. 2-5: Login-Fenster



- Geben Sie als Benutzername und Passwort jeweils **mxadmin** ein.

- Klicken Sie auf der Weboberfläche des MSX-E-Systems auf den Menüpunkt **Data server**.

Abb. 2-6: Data server: Configuration

Protocol	TCP
TCP port number	8989
SO_SNDBUF (in bytes)	104448
TCP/IP network clients filter	
UDP/IP network targets	

- Passen Sie im Abschnitt **Configuration** im Feld **TCP port number** die Portnummer an, falls Sie die Standard-Portnummer (8989) nicht verwenden können.

Abb. 2-7: Data server: Set and save

Set and save Reload

- Wenn Sie die Portnummer geändert haben, klicken Sie auf **Set and Save** und danach auf **Reload**.
- Klicken Sie auf den Registerreiter **Blocking transfer**.

Abb. 2-8: Data server: Blocking transfer

Activate blocking TCP/IP transfer	No
TCP/IP transfer timeout	1,0

- Bitte achten Sie darauf, dass im Abschnitt **Configuration** bei **Activate blocking TCP/IP transfer** die Option **No** ausgewählt ist.
- Wenn Sie diese Änderung vorgenommen haben, klicken Sie auf **Set and Save** und danach auf **Reload**.

3 Lesen der Messdaten

Wie bereits in Kap. 2 erwähnt, stellt Ihnen ADDI-DATA das Programmierbeispiel einer S7-SPS von Siemens zur Verfügung. Ziel dieses Beispiel ist es zu demonstrieren, wie die SPS Messwerte eines MSX-E Systems aus einer TCP-Verbindung liest.

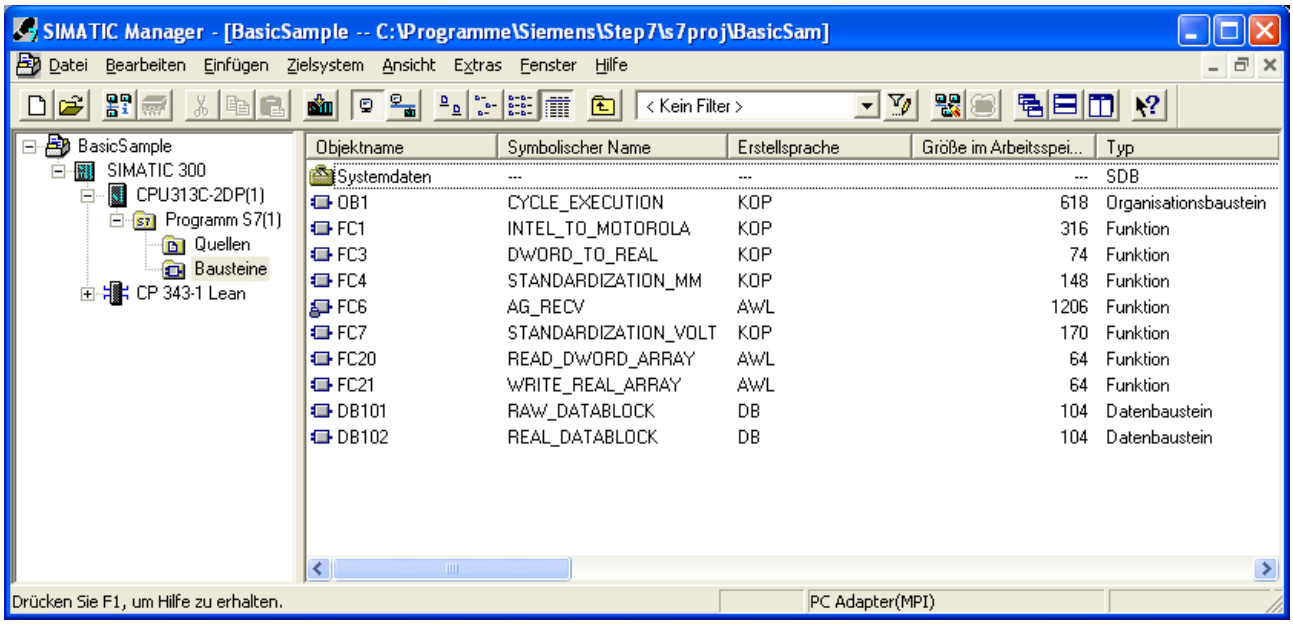
3.1 Voraussetzungen

Bitte stellen Sie sicher, dass folgende Voraussetzungen erfüllt sind:

- Siemens CPU313C-2DP (SPS)
- Siemens CP343-1 Lean (Ethernet-System für die SPS)
- FC6 (AG_RECV) für S7-300 (Funktion zur asynchronen Kommunikation)
- S7-Beispiel für STEP7.

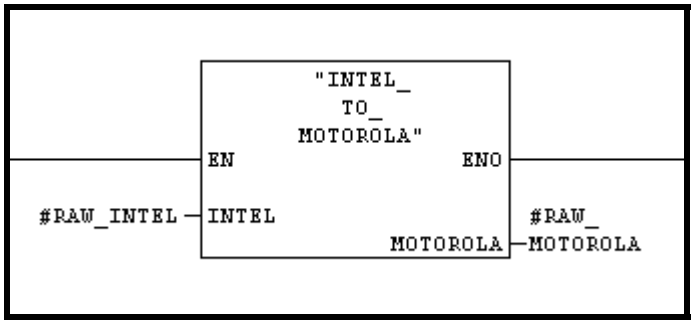
3.2 S7: Beschreibung der Funktionsbausteine

Abb. 3-1: SIMATIC-Manager: Auflistung der Bausteine



3.2.1 FC1: Konvertierung von Daten in das Motorola-Format

Beschreibung:
Bei den MSX-E-Systemen werden die Daten im Intel-Format dargestellt, bei der S7-SPS dagegen im Motorola-Format.
Wenn Ihre SPS bereits das Intel-Format verwendet, benötigen Sie diesen Funktionsbaustein nicht.
Arbeitet Ihre SPS aber im Motorola-Format, so können Sie mit diesem Baustein die Werte konvertieren.



Eingabe-Parameter:

INTEL: Rohwert im **Intel Format** als DWORD-Datentyp, der aus dem MSX-E-System gelesen wird

Ausgabe-Parameter:

MOTOROLA: Wert im Motorola-Format als DWORD-Datentyp

Beispiel für die Reihenfolge der Funktionsaufrufe:

Tabelle 3-1: FC1: Reihenfolge der Funktionsaufrufe

FC-Nummer	Name	Beschreibung
FC6	**AG_RECV**	Wert aus dem Netzwerk lesen
FC1	**INTEL_TO_MOTOROLA**	Wert in das Motorola-Format konvertieren

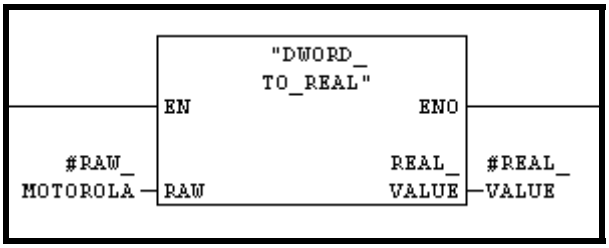
3.2.2 FC3: Konvertierung eines DWORD-Datentyps in einen REAL-Datentyp

Beschreibung:
Auf den MSX-E-Systemen werden die Messwerte auf 4 Byte kodiert und über die TCP-Verbindung als Byte-Pakete gesendet. Standardmäßig werden die Messwerte als Rohwerte gesendet.

Beispiel: System MSX-E3011 mit Gain 1 Unipolar
Wenn 10 Volt gemessen werden, sendet das System nicht den Wert 10, sondern 65535. Dieser Wert wird als Rohwert bezeichnet.

65535 ist ein Rohwert, der in die passende Einheit konvertiert wird:
((Spannungsbereich / Auflösung) x Rohwert) + Minimalspannung
((20 / 65535) x 65535) + (-10) = 10 Volt

Falls die Systeme die Rohwerte sofort in die passende Einheit konvertieren und senden, so ist keine weitere Konvertierung auf der SPS nötig. Die aus der TCP-Verbindung gelesenen Werte sollen zuerst richtig dargestellt und danach mit einem Typecast in den REAL-Datentyp umgewandelt werden.
Der Typecast wird benötigt, weil die empfangenen Werte nicht direkt als REAL-Typ gesendet und empfangen werden.



Eingabe-Parameter:

RAW: Rohwert, der vom System in die passende Einheit konvertiert wurde, als DWORD-Datentyp im **Motorola-Format**

Ausgabe-Parameter:

REAL_VALUE: Typecast-Wert als REAL-Datentyp

Beispiel für die Reihenfolge der Funktionsaufrufe:

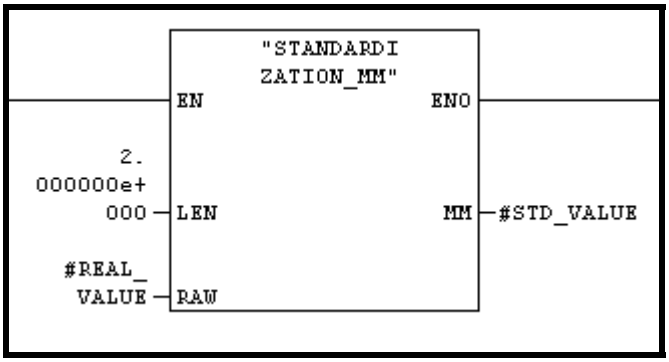
Tabelle 3-2: FC3: Reihenfolge der Funktionsaufrufe

FC-Nummer	Name	Beschreibung
FC6	**AG_RECV**	Wert aus dem Netzwerk lesen
FC1	**INTEL_TO_MOTOROLA**	Wert in das Motorola-Format konvertieren
FC3	**DWORD_TO_REAL**	Wert als REAL-Typ darstellen (Typecast)

3.2.3 FC4: Konvertierung in Millimeter (für Messtaster)

Beschreibung:

Diese Funktion kann bei den Systemen MSX-E370x und MSX-E3711 verwendet werden, um Rohwerte in die Einheit Millimeter zu konvertieren.



Eingabe-Parameter:

LEN: Wert als REAL-Datentyp, der den Messtasterweg in Millimeter definiert

RAW: Wert als REAL-Datentyp, der bereits im **Motorola-Format** dargestellt wird (siehe **Reihenfolge der Funktionsaufrufe**)

Ausgabe-Parameter:

MM: Messwert als REAL-Datentyp in Millimeter

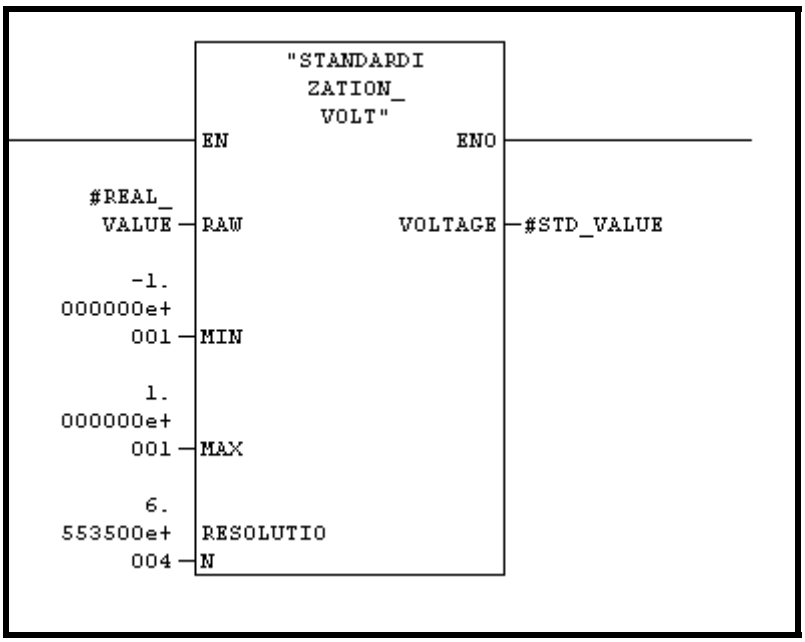
Beispiel für die Reihenfolge der Funktionsaufrufe:

Tabelle 3-3: FC4: Reihenfolge der Funktionsaufrufe

FC-Nummer	Name	Beschreibung
FC6	**AG_RECV**	Wert aus dem Netzwerk lesen
FC1	**INTEL_TO_MOTOROLA**	Wert in das Motorola-Format konvertieren
FC3	**DWORD_TO_REAL**	Wert als REAL-Typ darstellen (Typecast)
FC4	**STANDARDIZATION_MM**	Wert in Millimeter konvertieren

3.2.4 FC7: Konvertierung in Volt bzw. mA

Beschreibung:
Diese Funktion kann beim System MSX-E3011 verwendet werden, um Rohwerte in die Einheit Volt zu konvertieren.



Eingabe-Parameter:

RAW: Wert als REAL-Datentyp, der bereits im **Motorola-Format** dargestellt wird (siehe **Reihenfolge der Funktionsaufrufe**)

MIN: Wert als REAL-Datentyp, der den Minimalspannungswert des Spannungsbereichs definiert.

Beispiel: System MSX-E3011
Mit Gain 1 Bipolar ist dieser Wert gleich -10 Volt. Mit Gain 2 Bipolar beträgt dieser Wert -5 Volt.

MAX: Wert als REAL-Datentyp, der den Maximalspannungswert des Spannungsbereichs definiert.

Beispiel: System MSX-E3011
Mit Gain 1 Bipolar ist dieser Wert gleich +10 Volt. Mit Gain 2 Bipolar beträgt dieser Wert +5 Volt.

RESOLUTION: Wert als REAL-Datentyp, der die Auflösung des Messbereichs definiert.

Beispiel: Ein System MSX-E3011 hat eine Auflösung von 16 Bit (= 65535 Schritte).

Ausgabe-Parameter:

VOLTAGE: Messwert als REAL-Datentyp in Volt

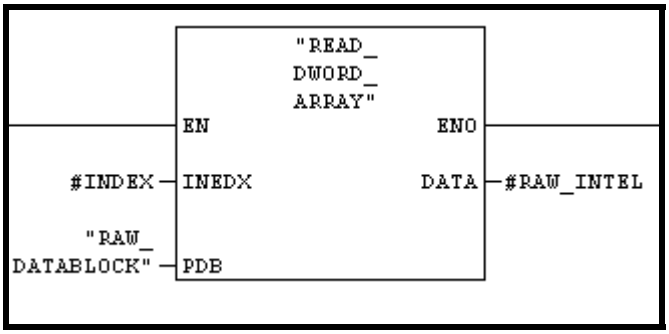
Beispiel für die Reihenfolge der Funktionsaufrufe:

Tabelle 3-4: FC7: Reihenfolge der Funktionsaufrufe

FC-Nummer	Name	Beschreibung
FC6	**AG_RECV**	Wert aus dem Netzwerk lesen
FC1	**INTEL_TO_MOTOROLA**	Wert in das Motorola-Format konvertieren
FC3	**DWORD_TO_REAL**	Wert als REAL-Typ darstellen (Typecast)
FC7	**STANDARDIZATION_VOLT**	Wert in Volt konvertieren

3.2.5 FC20: Dynamisches Lesen eines DWORD-Werts aus einer Tabelle

Beschreibung:
Dieser Funktionsbaustein wird nicht unbedingt benötigt. Er ermöglicht es, einen DWORD-Wert aus einer DWORD-Tabelle (die sich selbst in einem Datenbaustein befindet) zu lesen. Statt den Index der Tabelle fest zu definieren, kann dieser durch den Baustein dynamisch verwaltet werden. FC20 enthält allerdings keine Fehlerverwaltung (z.B.: Fehler = Index > Tabellenbreite).



Eingabe-Parameter:

INDEX: Index als INT-Datentyp, der aus der Tabellenzelle zu lesen ist

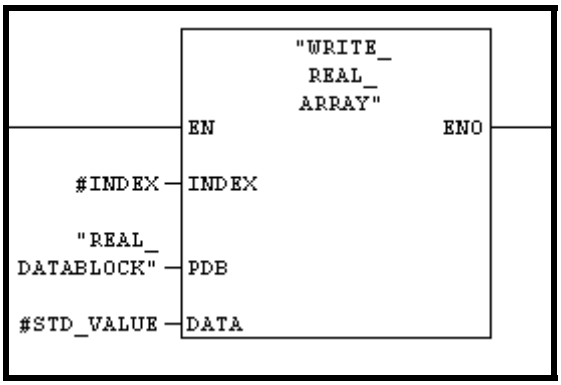
PDB: Datenbaustein, in dem sich die Tabelle als DWORD-Datentyp befindet

Ausgabe-Parameter:

DATA: Gelesener Wert als DWORD-Datentyp

3.2.6 FC21: Dynamisches Schreiben eines REAL-Werts in eine Tabelle

Beschreibung:
Dieser Funktionsbaustein wird nicht unbedingt benötigt. Er ermöglicht es, einen REAL-Wert in eine REAL-Tabelle (die sich selbst in einem Datenbaustein befindet) zu schreiben. Statt den Index der Tabelle fest zu definieren, kann dieser durch den Baustein dynamisch verwaltet werden. FC21 enthält allerdings keine Fehlerverwaltung (z.B.: Fehler = Index > Tabellenbreite).



Eingabe-Parameter:

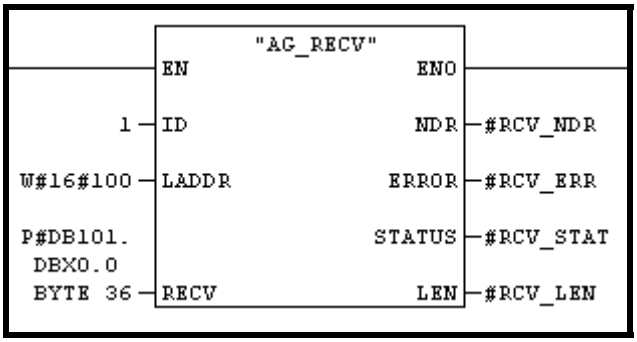
INDEX: Index als INT-Datentyp, der in die Tabellenzelle zu schreiben ist

PDB: Datenbaustein, in dem sich die Tabelle als REAL-Datentyp befindet

DATA: Wert, der als REAL-Datentyp zu schreiben ist

3.2.7 FC6: Lesen von Daten aus einer Netzwerkverbindung

Beschreibung:
Dieser Funktionsbaustein wurde von Siemens erstellt und ermöglicht es, Daten aus einer Netzwerkverbindung zu lesen.

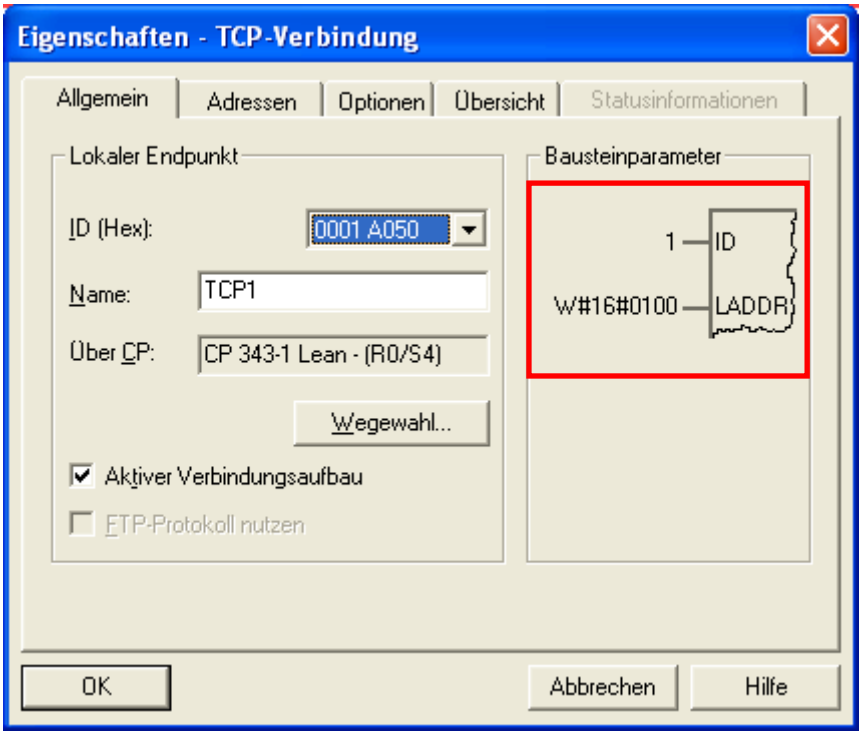


Eingabe-Parameter:

ID: 1 ist die Nummer des Kommunikationsidentifikators des Datenservers.

LADDR: Der Wert „W#16#0100“ ist die Adresse des Funktionsbausteins.

Abb. 3-2: Eigenschaften – TCP-Verbindung: Eingabe-Parameter



RECV: Zeiger, in dem die Daten gespeichert werden müssen; in diesem Fall in **DB101**.

P#DB101.DBX0.0 BYTE 36 bedeutet, dass ein Zeiger auf dem Offset 0 dem Datenbaustein DB101 (RAW_DATABLOCK) übergeben wird und 36 Byte (9 DWORD) erwartet werden. Die Anzahl der erwarteten Bytes kann sich je nach Bedarf ändern. Achten Sie darauf, dass die Tabelle, in der die Werte gespeichert werden, breit genug ist. Die Anzahl der Bytes entspricht immer einem Datenpaket.

Die Größe eines Datenpakets wird auf der Weboberfläche des MSX-E-Systems angegeben:

- Klicken Sie auf der Weboberfläche auf den Menüpunkt **Acquisition**.
- Wählen Sie den **Auto-Refresh-** oder **Sequenz-Modus** aus.

Im Abschnitt **Data server frame format** wird die Größe des Datenpakets angezeigt.

Abb. 3-3: Acquisition: Data server frame format

Size	Field	Description
4 Bytes	Inductive transducer 1	Digital value, encoded on 24 bits.
4 Bytes	Inductive transducer 6	Digital value, encoded on 24 bits.
4 Bytes	Inductive transducer 3	Digital value, encoded on 24 bits.

Wenn der Funktionsbaustein die Werte aus dem MSX-E-System empfangen hat, finden Sie diese in der Tabelle des Zieldatenbausteins in folgender Reihenfolge vor:

RAW[0]: enthält den Wert „counter“
RAW[1]: enthält den Wert „channel 1“
RAW[2]: enthält den Wert „channel 2“
…
RAW[9]: enthält den Wert „channel 8“

Das nachfolgende Beispiel basiert auf einem S7-Screenshot **(Data server frame format)** und einem Datenbaustein, in dem sich die Tabelle **RAW** als DWORD-Datentyp mit einer Breite von 36 Byte befindet.

Abb. 3-4: Datenbaustein mit der Tabelle RAW

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	RAW	ARRAY[0..8]		
*4.0		DWORD		
=36.0		END_STRUCT		

Abb. 3-5: Tabelle RAW im Datenbaustein

Address	Name	Type	Initial value	Actual value	Comment
0.0	RAW[0]	DWORD	DW#16#0	DW#16#6F8A0700	
4.0	RAW[1]	DWORD	DW#16#0	DW#16#BB06BF00	
8.0	RAW[2]	DWORD	DW#16#0	DW#16#8B447F00	
12.0	RAW[3]	DWORD	DW#16#0	DW#16#A1FF7F00	
16.0	RAW[4]	DWORD	DW#16#0	DW#16#EEFB7F00	
20.0	RAW[5]	DWORD	DW#16#0	DW#16#7EFE7F00	
24.0	RAW[6]	DWORD	DW#16#0	DW#16#FCFE7F00	
28.0	RAW[7]	DWORD	DW#16#0	DW#16#7DFF7F00	
32.0	RAW[8]	DWORD	DW#16#0	DW#16#B3028000	

Ausgabe-Parameter:

NDR: Wenn die Anzahl der Bytes gelesen und NDR gleich 1 ist, dann ist kein Fehler aufgetreten. Beträgt NDR 0, so sind ERROR und STATUS zu überprüfen, um den angezeigten Fehler zu finden.

LEN: Anzahl von Lese-Bytes. Die Anzahl muss 36 betragen (= 9 x 4 Byte).

4 Einfaches S7-Beispiel

Bitte konfigurieren Sie die Netzwerke 1 bis 3 gemäß der Beschreibung in den nachfolgenden Kapiteln.

4.1 Konfiguration des MSX-E-Systems über die Weboberfläche

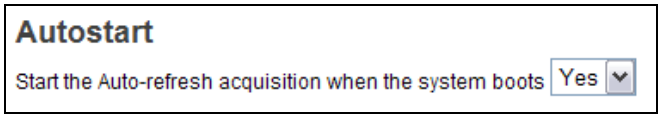
In diesem Kapitel werden nur die Schritte aufgeführt, die notwendig sind, um das Programmierbeispiel **BasicSample** bei einem MSX-E-System anzuwenden. Eine detaillierte Beschreibung der MSX-E-Weboberfläche finden Sie im allgemeinen Handbuch der MSX-E-Systeme (siehe PDF-Link) sowie im jeweiligen systemspezifischen MSX-E-Handbuch.

Mit der Standardeinstellung des **BasicSample** können Sie eine Erfassung im Auto-Refresh- bzw. Sequenz-Modus konfigurieren.
Es ist zu beachten, dass im Sequenz-Modus der Wert des Sequenz-Zählers (Sequence counter) nicht standardmäßig übertragen wird, während im Auto-Refresh-Modus dieser Wert immer zu Verfügung steht. Aus diesem Grund muss die entsprechende Einstellung auf der Weboberfläche aktiviert werden (siehe Kap. 4.1.2).

4.1.1 Auto-Refresh-Modus

- Wählen Sie links auf der Weboberfläche den Menüpunkt **Acquisition** und rechts die Registerkarte **Auto-refresh** aus.

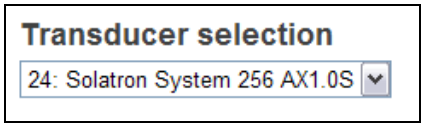
- Optional: Wählen Sie im Abschnitt **Autostart** die Option **Yes** aus, um den Autostart zu aktivieren.



- Wählen Sie im Abschnitt **Channel configuration** die zu erfassenden Kanäle aus.

Channel	Transducer	Selection
Channel 0	Inductive transducer 0	☑
Channel 1	Inductive transducer 1	☑
Channel 2	Inductive transducer 2	☐
Channel 3	Inductive transducer 3	☐
Channel 4	Inductive transducer 4	☐
Channel 5	Inductive transducer 5	☐

- Wählen Sie den angeschlossenen Messtastertyp aus. Dieser ist für alle Kanäle gleich.

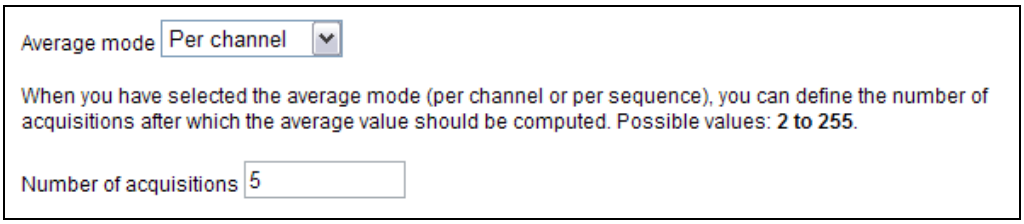


Optional: Abschnitt **Average**

Das MSX-E-System kann für jeden Kanal einen Mittelwert berechnen.
Im Feld **Number of acquisitions** ist die Anzahl der Erfassungen (2 bis 255) einzugeben, nach denen diese Berechnung erfolgen soll.

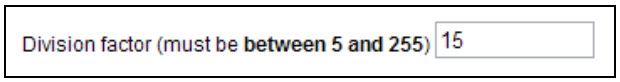
Die Erfassung ist auf zwei Arten **(Average mode)** möglich:

- **Per sequence:** Bei der Erfassung pro Sequenz werden alle ausgewählten Kanäle gleichzeitig erfasst.
- **Per channel:** Bei der Erfassung pro Kanal werden die ausgewählten Kanäle jeweils einzeln erfasst.



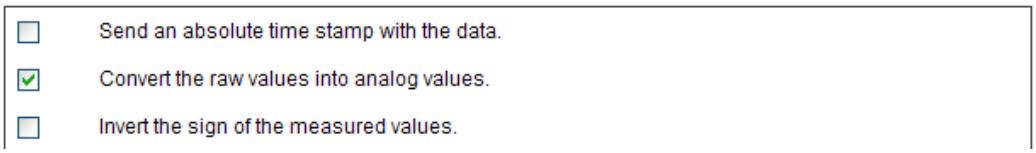
Im Abschnitt **Division factor** wird die Einschwingzeit gesetzt, d.h. die Zeit, die benötigt wird, um von einem Kanal auf den anderen umzuschalten. Bei nur einem ausgewählten Kanal ist dieser Parameter unerheblich.

- Geben Sie den **Division factor** ein (Wert zwischen 5 und 255).



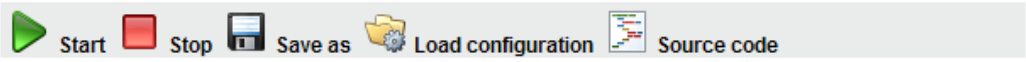
Im S7-Beispiel ist die Einstellung Typecast (siehe Kap. 4.2.3) standardmäßig ausgewählt. Somit muss im Abschnitt **Data server frame configuration** die Option **Convert the raw values into analog values** aktiviert werden, damit das System die Rohwerte sofort in die richtige Einheit umrechnet. Diese ist abhängig vom Systemtyp.

- Klicken Sie auf **Convert the raw values into analog values**.



- Optional: Klicken Sie auf **Set and Save**, um die aktuelle Konfiguration zu speichern.

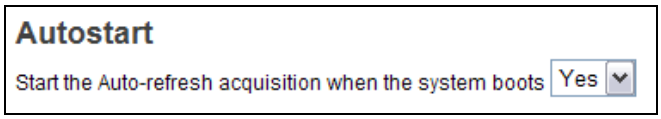
- Um die Erfassung zu starten, klicken Sie auf **Start**.



4.1.2 Sequenz-Modus

- Wählen Sie links auf der Weboberfläche den Menüpunkt **Acquisition** und rechts die Registerkarte **Sequence** aus.

- Optional: Wählen Sie **Yes** aus, um den Autostart zu aktivieren.

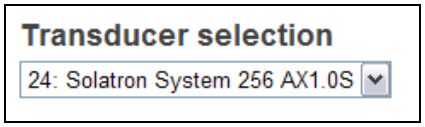


- Wählen Sie im Abschnitt **Channel configuration** die zu erfassenden Kanäle aus.

Sie können die Reihenfolge, in der die Kanäle erfasst werden sollen, selbst definieren. Diese wird in der Spalte „Acquisition order“ angezeigt, sobald Sie einen Kanal ausgewählt haben. Pro Sequenz kann jeder Kanal nur einmal erfasst werden.

Channel	Transducer	Selection	Acquisition order
Channel 0	Inductive transducer 0	☑	2
Channel 1	Inductive transducer 1	☑	4
Channel 2	Inductive transducer 2	☑	1
Channel 3	Inductive transducer 3	☑	0
Channel 4	Inductive transducer 4	☑	5
Channel 5	Inductive transducer 5	☑	3

- Wählen Sie den angeschlossenen Messtastertyp aus. Dieser ist für alle Kanäle gleich.



Im Abschnitt **Division factor** wird die Einschwingzeit gesetzt, d.h. die Zeit, die benötigt wird, um von einem Kanal auf den anderen umzuschalten. Bei nur einem ausgewählten Kanal ist dieser Parameter unerheblich.

- Geben Sie den **Division factor** ein (Wert zwischen 5 und 255).

Division factor (must be **between 5 and 255**) 15

Optional: Im Abschnitt **Delay** haben Sie die Möglichkeit, die Wartezeit zwischen den einzelnen Sequenzen zu definieren. Es gibt zwei Modi:

- **Modus 1:** Als Wartezeit ist die Zeit zwischen dem jeweiligen Beginn zweier aufeinanderfolgender Sequenzen definiert.
- **Modus 2:** Als Wartezeit ist die Zeit zwischen dem Ende einer Sequenz und dem Beginn der darauf folgenden Sequenz definiert.

Bei **Delay** sind der Wert und die Einheit der Wartezeit (ms oder s) festzulegen.

Delay mode Mode 1

Possible values:

- If **mode 1** is selected, the delay value must be a value **between the switching time between channels and 65535** (milliseconds or seconds).
- If **mode 2** is selected, the delay value must be a value **between 0 and 65535** (milliseconds or seconds).

Delay 100 Millisecond

Im Abschnitt **Sequence measurement** wird im Feld **Number of sequences** die Anzahl der zu erfassenden Sequenzen eingegeben (1 bis 4294967295). Lautet dieser Wert 0, so findet eine Dauererfassung statt.

Im Feld **Number of data frames** wird die Anzahl der Sequenzen (1 bis 4096) festgelegt, die zu erfassen sind, bevor die Messwerte an das Zielsystem gesendet werden.

HINWEIS!

Der eingegebene Wert darf nicht höher als der Wert im Feld „Number of sequences“ sein. Letzterer muss durch diesen Wert teilbar sein.

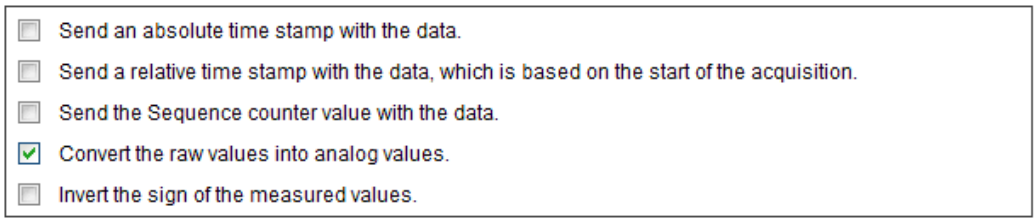
Falls der Speicherplatz auf dem MSX-E-System nicht ausreicht, um die gewünschte Anzahl von Sequenzen zu speichern, so werden die Messwerte früher gesendet, d.h. bevor die maximale Anzahl der zu erfassenden Sequenzen erreicht ist. Dies dient dazu, die Belastung des Netzverkehrs und der CPU-Ressourcen der MSX-E-Systeme zu reduzieren.

Number of sequences	0
Number of data frames	1

Im S7-Beispiel ist die Einstellung Typecast (siehe Kap. 4.2.3) standardmäßig ausgewählt. Somit muss im Abschnitt **Data server frame configuration** die Option **Convert the raw values into analog values** aktiviert werden, damit das System die Rohwerte sofort in die richtige Einheit umrechnet. Diese ist abhängig vom Systemtyp.

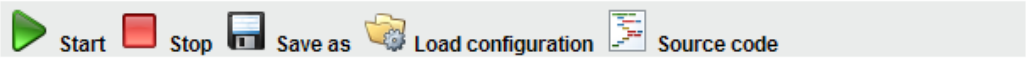
- Klicken Sie auf **Convert the raw values into analog values**.

Im Sequenz-Modus zählt der Sequenz-Zähler (Sequence counter) die Anzahl der bereits erfassten Sequenzen. Falls die Option **Send the Sequence counter value** aktiviert ist, so wird neben den Daten auch der Wert des Sequenz-Zählers gesendet.



- Optional: Klicken Sie auf **Set and Save**, um die aktuelle Konfiguration zu speichern.

- Um die Erfassung zu starten, klicken Sie auf **Start**.



4.2 S7: Beschreibung des Beispiels

4.2.1 OB1: Hauptbaustein

Beschreibung:

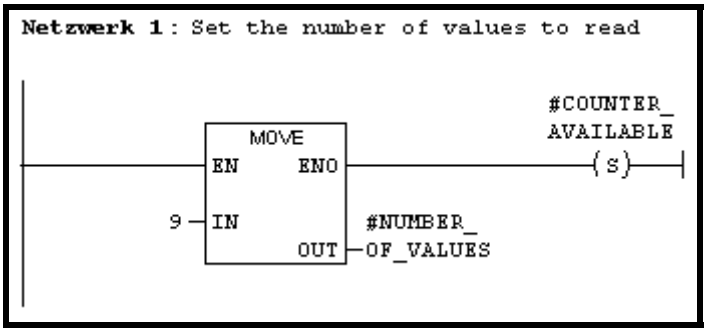
Der OB1 (Organisationsbaustein 1) wird periodisch ausgeführt. Sobald er beendet ist, startet er erneut. Um einen einfachen Überblick über den gesamten Quellcode zu erhalten, werden im **BasicSample** alle Funktionsbausteine aus OB1 aufgerufen.

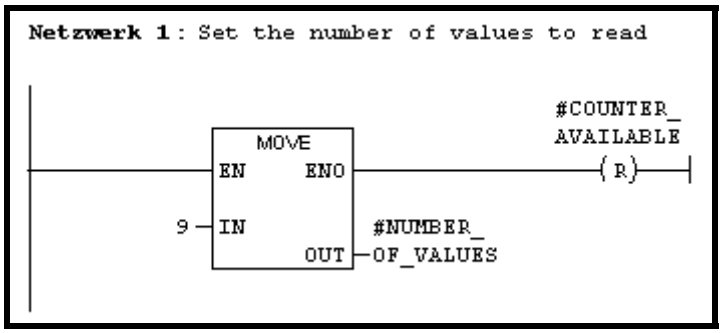
4.2.2 Netzwerk 1: Definition der Anzahl der zu lesenden Werte

Beschreibung:

Geben Sie die Anzahl der zu lesenden Werte an.

In diesem Beispiel werden 9 Werte gelesen. Der erste Wert kann ein Wert des Sequenz-Zählers oder ein Wert der 8 Kanäle sein. Wenn ein Sequenz-Zähler-Wert erwartet wird, dann ist die Variable COUNTER_AVAILABLE zu setzen. Falls nicht, so ist diese Variable zurückzusetzen.





4.2.3 Netzwerk 2: Auswahl der passenden Konvertierungseinheit

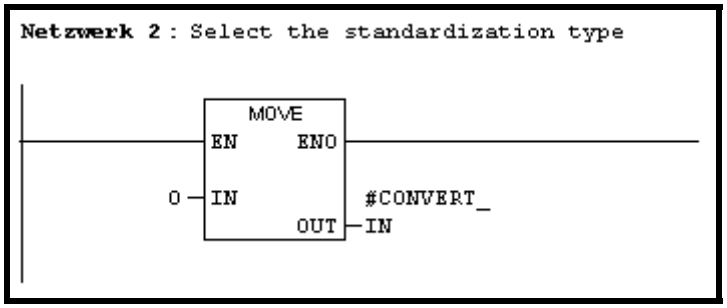
Beschreibung:

Bevor das **BasicSample** auf die S7 übertragen wird, ist in Netzwerk 2 die Einheit der Konvertierung über die Variable CONVERT_IN festzulegen.

CONVERT_IN = 0: Das System liefert keine Rohwerte, sondern gleich die konvertierten Werte (Typecast wird ausgeführt).

CONVERT_IN = 1: Das System liefert Rohwerte. Beim System MSX-E37xx werden diese von der SPS in die Einheit Millimeter konvertiert.

CONVERT_IN = 2: Das System liefert Rohwerte. Beim System MSX-E3011 werden diese von der SPS in die Einheit Volt konvertiert.



Wie bereits erwähnt, können Sie auf der Weboberfläche des MSX-E-Systems festlegen, dass die Messwerte in die passende Einheit konvertiert werden (siehe Kap. 4.1.2, Abschnitt **Data server frame configuration**).

4.2.4 Netzwerk 3: Lesen der Messwerte aus der TCP-Verbindung

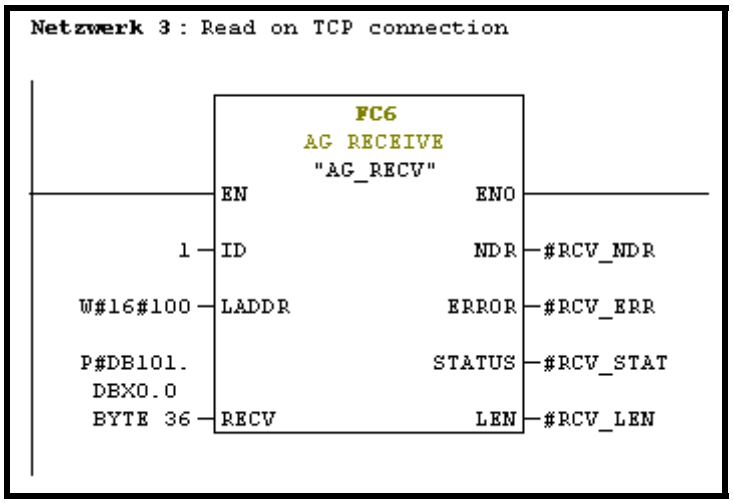
Beschreibung:

FC6 versucht, 36 Byte vom MSX-E-System aus der TCP-Verbindung ID 1 und ADDR W#16#100 zu lesen. Wenn keine Verbindungsfehler aufgetreten sind und die 36 Byte zu Verfügung stehen, wird die Variable RCV_NDR auf „wahr“ gesetzt.

Der Eingangparameter RECV (P#DB101.DBX0.0) ist ein Zeiger auf die Daten.

DB101 ist ein Datenbaustein, in den die 36 Byte kopiert werden (36 Byte = 9 x DWORD). RECV kann an den jeweiligen Bedarf angepasst werden. Falls die Anzahl der zu lesenden Bytes geändert wurde, ist auch zu prüfen, ob die Anzahl der Elemente in den Datenbausteinen DB101 und DB102 ausreichend ist.

Weitere Informationen dazu finden Sie in Kapitel 3.2.7.

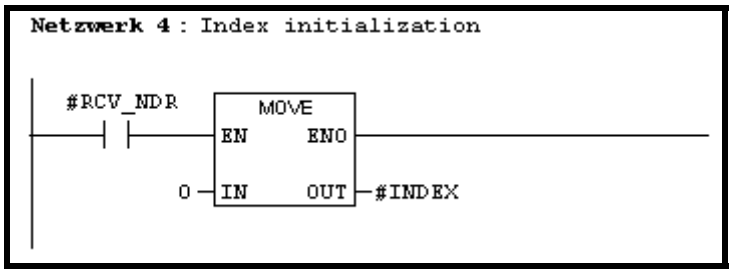


4.2.5 Netzwerk 4: Initialisierung des Index

Beschreibung:

Das Programm verwendet zwei Datenstrukturen.

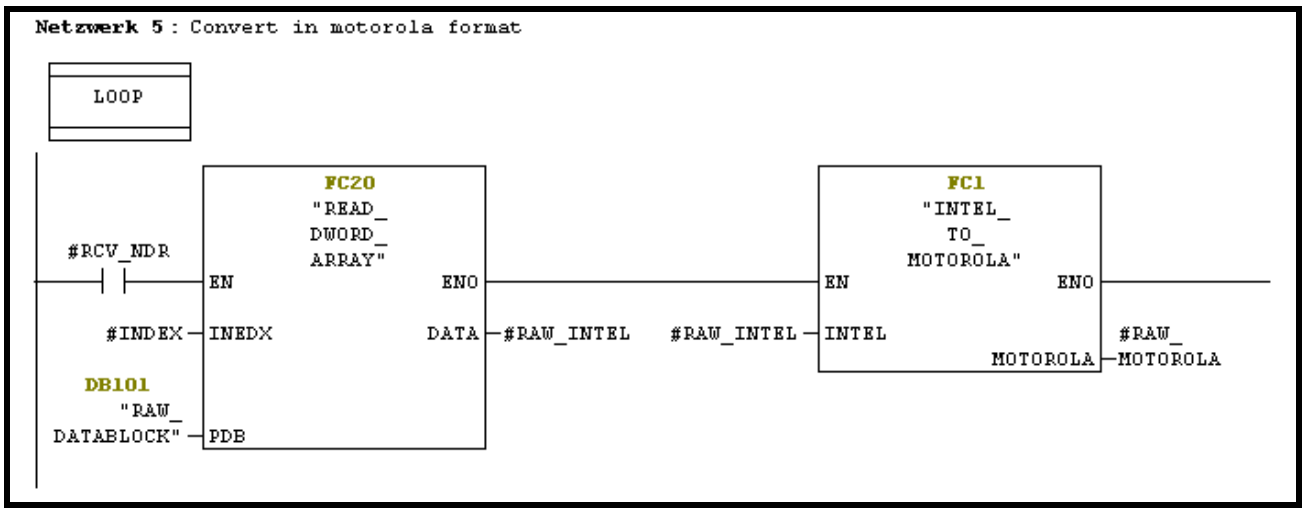
Die Struktur RAW_DATABLOCK (DB101) enthält eine DWORD-Tabelle, in der die gelesenen Rohwerte zur Verfügung stehen.
Die Datenstruktur REAL_DATABLOCK (DB102) enthält eine REAL-Tabelle, in der die konvertierten Rohwerte als REAL-Datentyp gespeichert werden. Die Variable INDEX wird auf 0 gesetzt. INDEX wird verwendet, um den Tabellenindex dynamisch zu verwalten.



4.2.6 Netzwerk 5: Darstellung im Motorola-Format

Beschreibung:

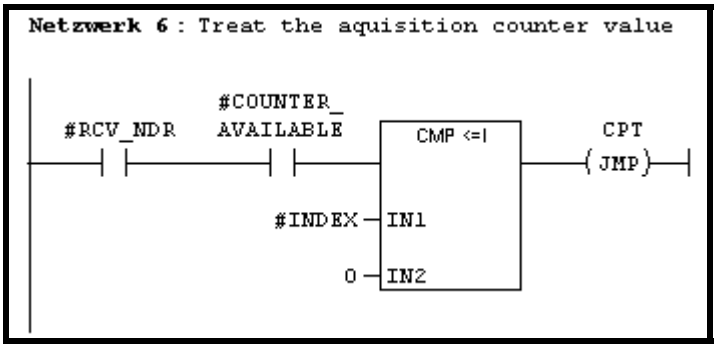
Aus der Tabelle RAW_DATABLOCK wird ein Rohwert aus der jeweiligen Zelle des INDEX gelesen. Dieser Wert wird vom Intel-Format (MSX-E-System) in das Motorola-Format (SPS) konvertiert.



4.2.7 Netzwerk 6: Verwaltung des Sequenz-Zähler-Werts

Beschreibung:

Wenn der Sequenz-Zähler-Wert erwartet wird und der INDEX gleich 0 ist, dann ist der Wert ein Sequenz-Zähler-Wert. Anschließend springt das Programm in das Netzwerk, das die Markierung CPT enthält. Der Sequenz-Zähler-Wert ist immer der erste Wert in der Tabelle.

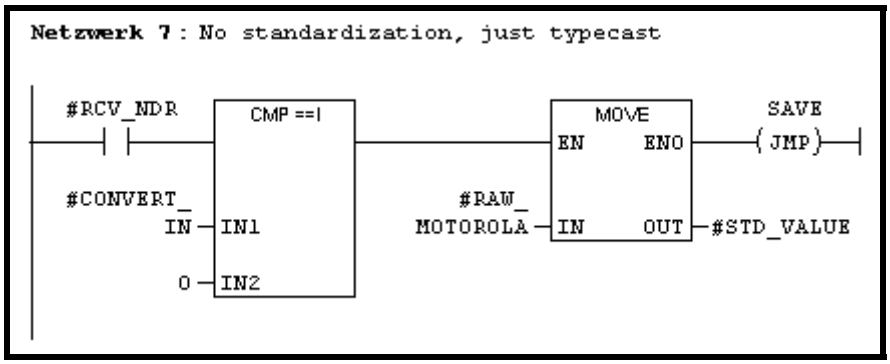


4.2.8 Netzwerk 7: Typecast statt Konvertierung

Beschreibung:

Da das System bereits konvertierte Werte liefert, ist keine weitere Konvertierung mehr nötig.

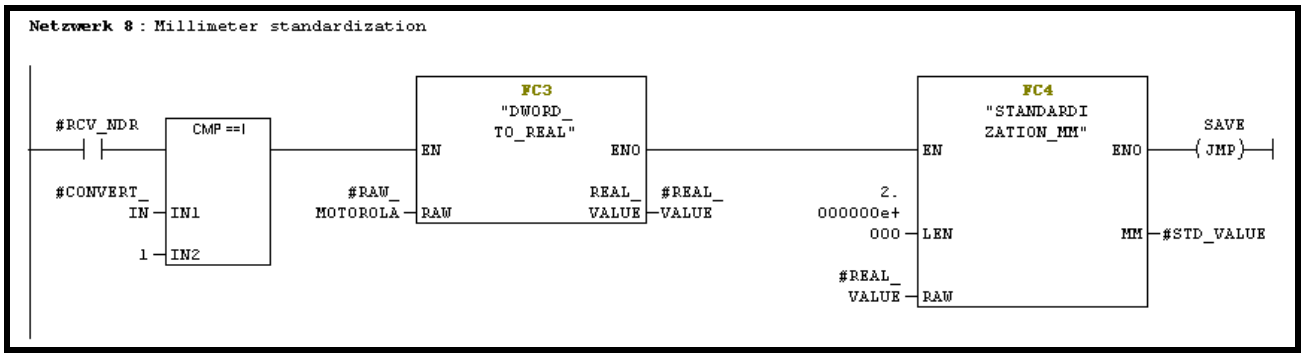
MOVE kopiert die DWORD-Rohwerte in eine Variable des Datentyps REAL (Typecast). Danach springt das Programm in das Netzwerk mit der Markierung SAVE.



4.2.9 Netzwerk 8: Konvertierung in Millimeter

Beschreibung:

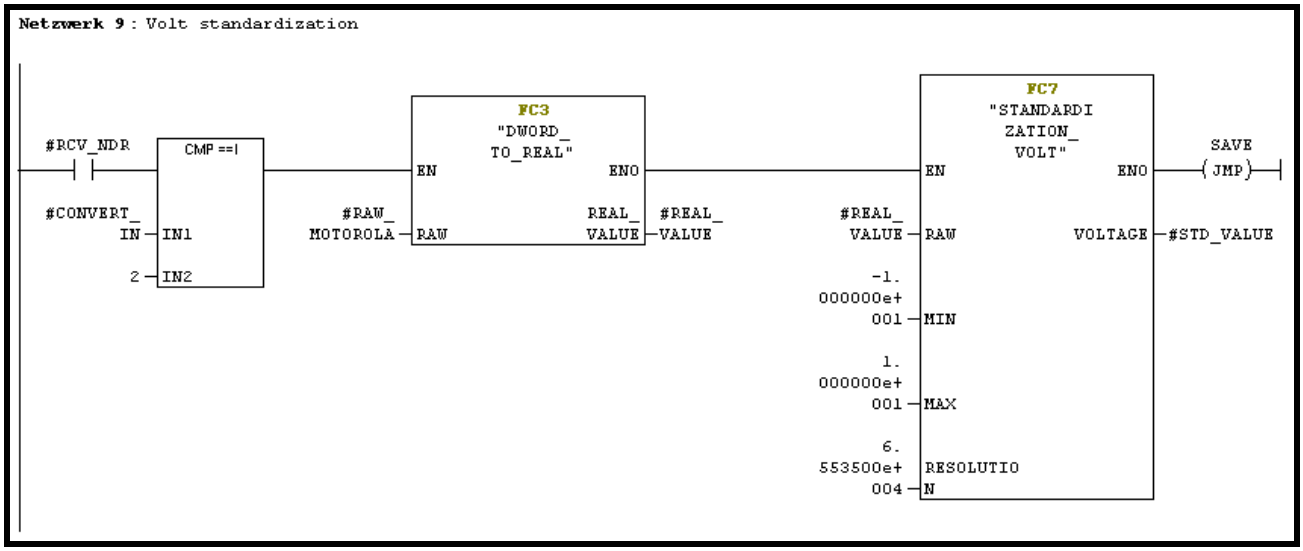
Das System liefert Rohwerte. Mit FC3 werden die DWORD-Rohwerte in den REAL-Datentyp konvertiert und mit FC4 in die Einheit Millimeter. Anschließend springt das Programm in das Netzwerk mit der Markierung SAVE.



4.2.10 Netzwerk 9: Konvertierung in Volt

Beschreibung:

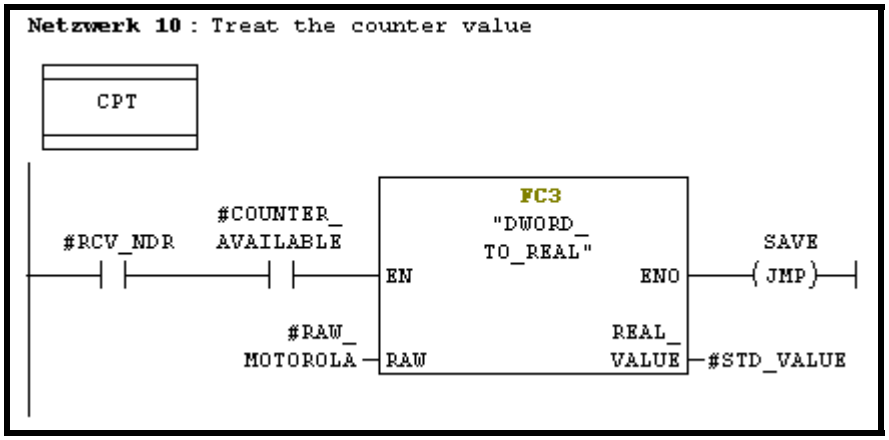
Das System liefert Rohwerte. Mit FC3 werden die DWORD-Rohwerte in den REAL-Datentyp konvertiert und mit FC7 in die Einheit Volt. Danach springt das Programm in das Netzwerk, das die Markierung SAVE enthält.



4.2.11 Netzwerk 10: Bearbeitung des Sequenz-Zähler-Werts

Beschreibung:

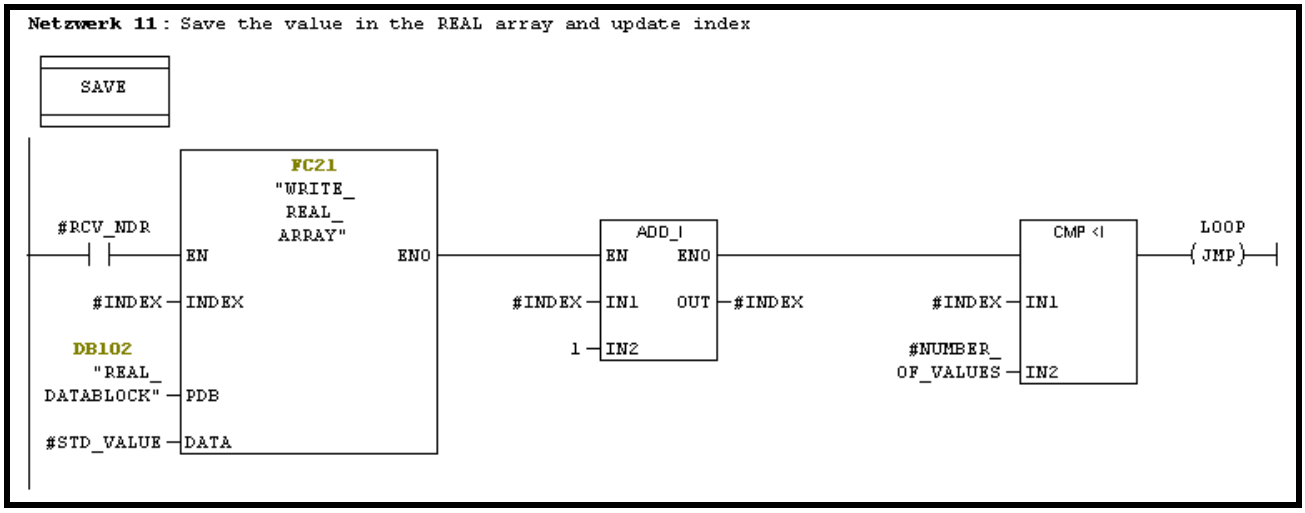
Der Sequenz-Zähler-Wert wird als DWORD-Datentyp in REAL konvertiert. Das Programm springt anschließend in das Netzwerk mit der Markierung SAVE.



4.2.12 Netzwerk 11: Speichern und Prüfen

Beschreibung:

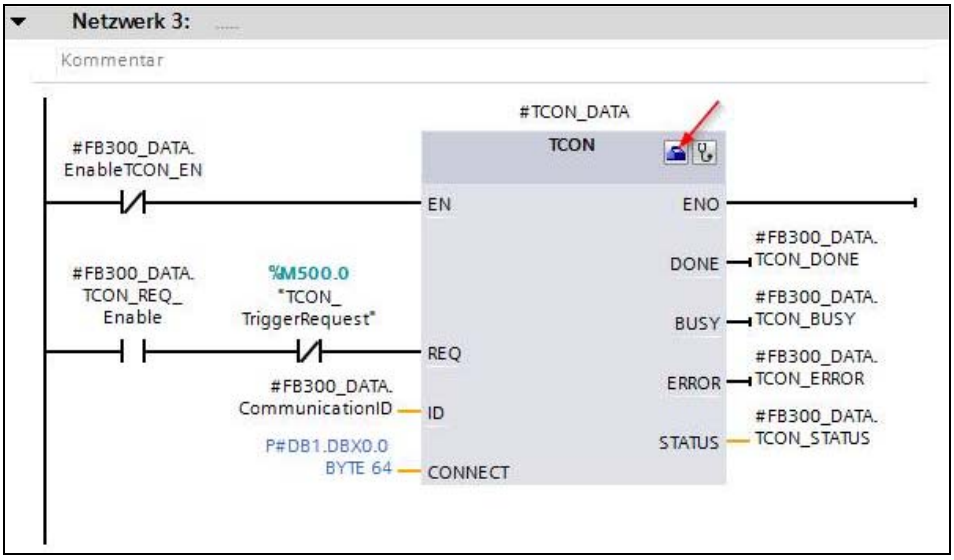
FC21 speichert den Wert in der REAL-Tabelle. ADD_I erhöht die Variable INDEX, damit der nächste Rohwert gelesen werden kann. CMP < I prüft, ob alle Rohwerte gelesen wurden. Falls nicht, dann springt das Programm in das Netzwerk mit der Markierung LOOP.



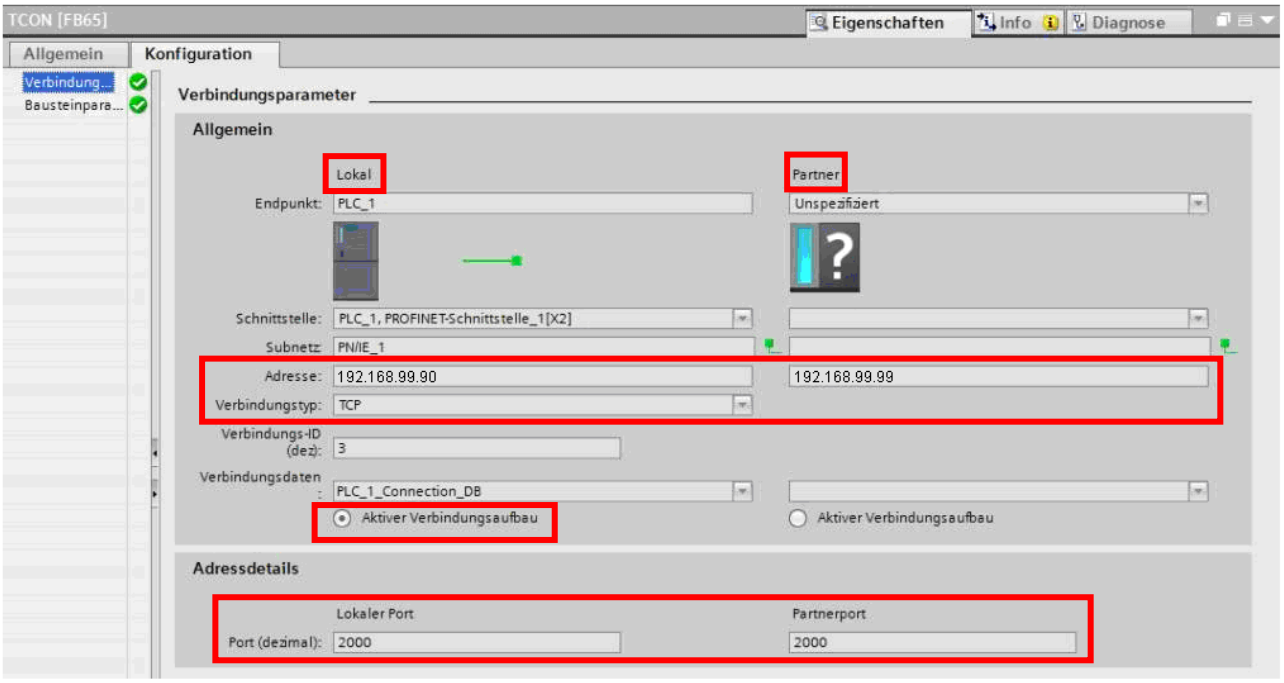
4.3 TIA Software-Umgebung mit VIPA CPU 015

Nachfolgend finden Sie eine kurze Beschreibung des Beispielprogramms **CPU015_TRECV** für das Auslesen von MSX-E-Daten mit einer SPS mit T-Bausteinen.
Das Beispielprogramm wurde mit **TIA Portal V13** erstellt und kann in den Programmiersprachen AWL, KOP und FUP angezeigt werden.

- Verwenden Sie die Netzwerk-Schnittstelle **X4 PN** der SPS.
- Klicken Sie auf das Werkzeugkasten-Symbol (siehe roter Pfeil), um die Kommunikationsparameter unter **FB300 Netzwerk 3 (TCON)** einzustellen.



Folgendes Fenster wird angezeigt:



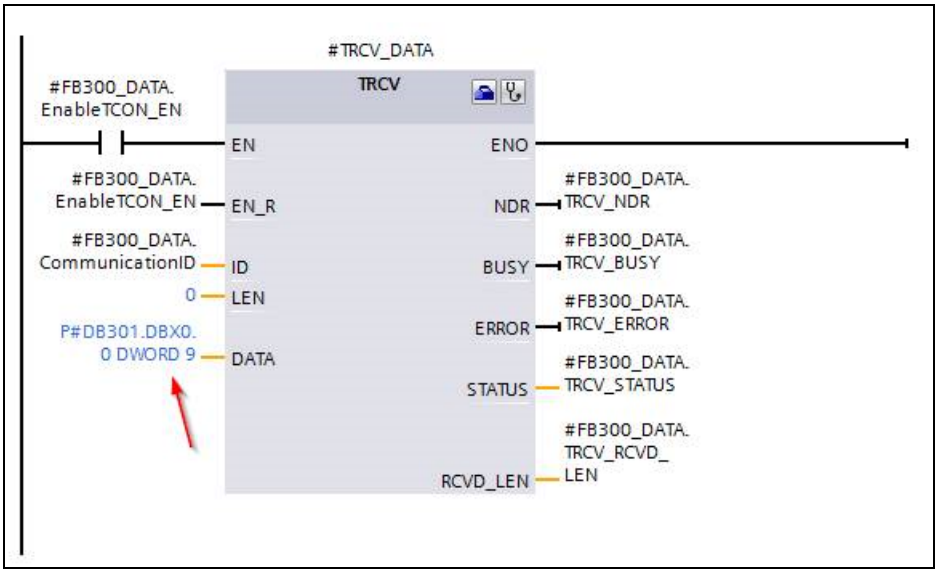
- Geben Sie im Abschnitt **Allgemein** im Feld **Adresse** die IP-Adresse der SPS (Bereich **Lokal**) und die des MSX-E-Systems (Bereich **Partner**) ein. Die beiden IP-Adressen müssen sich im gleichen Netzwerkbereich befinden.

- Wählen Sie den Verbindungstyp **TCP** aus.

- Aktivieren Sie den SPS-Modus **Aktiver Verbindungsaufbau** (Bereich **Lokal**).

- Geben Sie im Abschnitt **Adressdetails** im Bereich **Lokaler Port** (SPS) und **Partnerport** (MSX-E-System) jeweils die Portnummer **2000** ein. Die beiden Portnummern müssen gleich sein und mit der Portnummer übereinstimmen, die auf der Weboberfläche des MSX-E-Systems festgelegt wurde (siehe Kap. 2.2).

Die Anzahl der zu empfangenden Daten muss in **Netzwerk 10** eingestellt werden. **LEN** soll gleich 0 sein. Im **DATA**-Zeiger wird die Anzahl von DWORD angegeben.
Beispiel: Für eine MSX-E-Erfassung im Auto-Refresh-Modus mit 8 Känalen müssen 9 DWORD festgelegt werden (1 Zählerwert + 8 Kanalwerte entsprechen jeweils 1 DWORD).



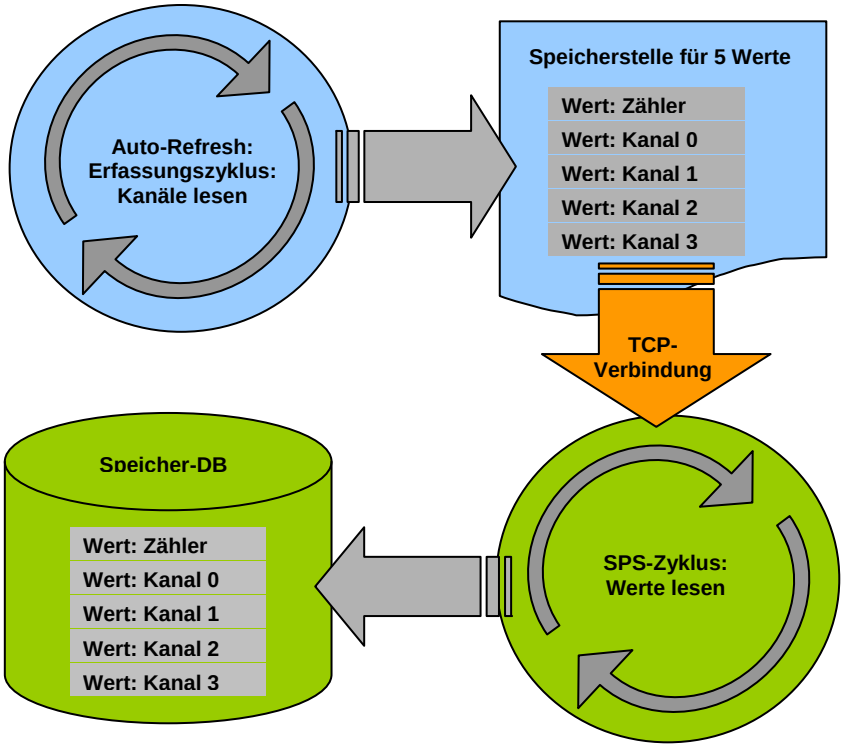
5 Empfehlungen

5.1 Auto-Refresh-Modus und Sequenz-Modus

HINWEIS!
Wenn im Auto–Refresh-Modus keine Mittelwertberechnung bzw. im Sequenz-Modus keine Wartezeit konfiguriert ist, benötigt die SPS eine gewisse Zeit, um die Werte aus der TCP-Verbindung zu lesen (SPS-Zyklus > Erfassungszyklus). Ebenso ist dies der Fall, wenn in beiden Modi ein kleiner **Division factor** konfiguriert ist. Somit kann es zu einer Verzögerung zwischen dem aktuellen Status der Kanäle und der gelesenen Messwerte kommen.
Falls die Pakete aus der TCP-Verbindung nicht schnell genug gelesen werden, kann es einen FIFO-Überlauf geben, wodurch das MSX-E-System die Verbindung zu der SPS abbricht.

5.1.1 Erfassung im Auto-Refresh-Modus

Abb. 5-1: Auto-Refresh-Modus: Erfassungszyklus



Der Erfassungszyklus wird mit vier Kanälen eines MSX-E-Systems im Auto-Refresh-Modus dargestellt (hellblau). Da für die fünf Werte nur eine Speicherstelle vorhanden ist, werden nach jedem Zyklus die Werte des vorherigen Zyklus mit denen des aktuellen Zyklus überschrieben.

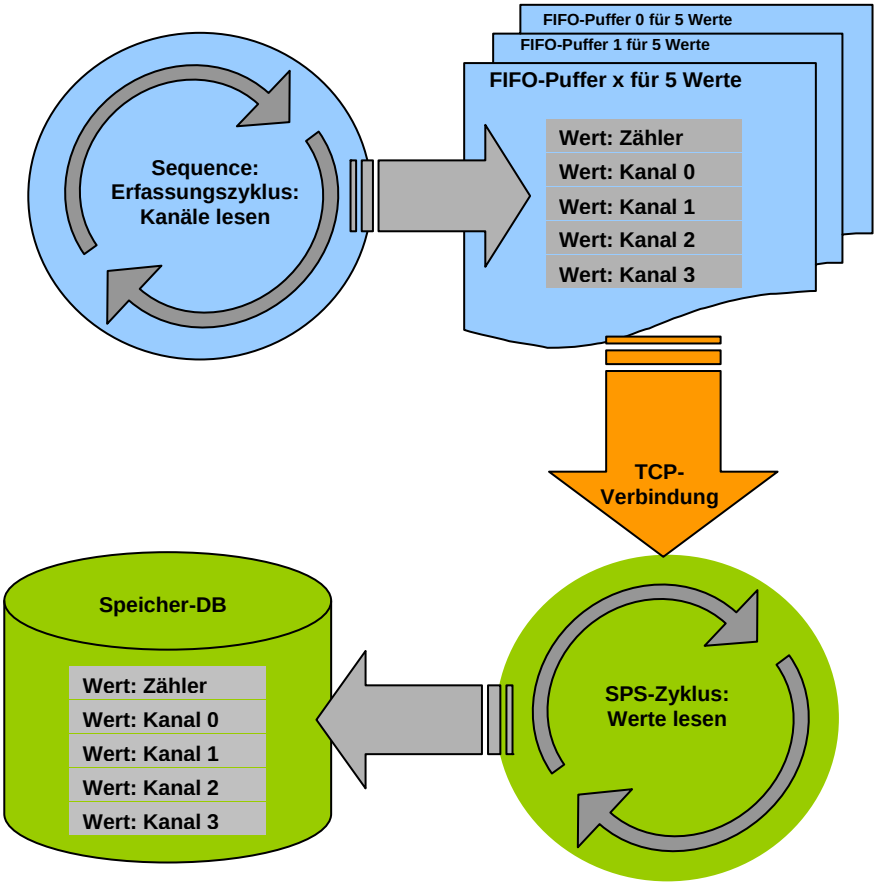
Das MSX-E-System sendet die Werte an die SPS, sobald diese angeschlossen ist. Im SPS-Zyklus werden die Werte aus der TCP-Verbindung gelesen (grün).

MSX-E-Erfassungszyklus (Auto-Refresh-Modus) < SPS-Zyklus	->	Werte können verloren gehen
MSX-E-Erfassungszyklus (Auto-Refresh-Modus) > SPS-Zyklus	->	alle Werte werden empfangen
MSX-E-Erfassungszyklus (Auto-Refresh-Modus) = SPS-Zyklus	->	alle Werte werden empfangen

HINWEIS!
Bei einem Socket-FIFO-Überlauf können Werte verloren gehen.

5.1.2 Erfassung im Sequenz-Modus

Abb. 5-2: Sequenz-Modus: Erfassungszyklus



Der Erfassungszyklus wird mit vier Kanälen eines MSX-E-Systems im Sequenz-Modus dargestellt (hellblau). Die Werte des aktuellen Zyklus werden jeweils in einen neuen FIFO-Puffer geschrieben, d.h., es werden keine Werte überschrieben.
Das MSX-E-System sendet die Werte an die SPS, sobald diese angeschlossen ist. Im SPS-Zyklus werden die Werte aus der TCP-Verbindung gelesen (grün).

MSX-E-Erfassungszyklus (Sequenz-Modus) < SPS-Zyklus	->	alle Werte werden empfangen
MSX-E-Erfassungszyklus (Sequenz-Modus) > SPS-Zyklus	->	alle Werte werden empfangen
MSX-E-Erfassungszyklus (Sequenz-Modus) = SPS-Zyklus	->	alle Werte werden empfangen

HINWEIS!
Bei einem Socket-FIFO-Überlauf können Werte verloren gehen.

5.2 Trigger

Bei den MSX-E-Systemen kann die Messung durch einen Trigger gestartet werden.
Weitere Informationen dazu finden Sie im allgemeinen Handbuch der MSX-E-Systeme (siehe PDF-Link) sowie im jeweiligen systemspezifischen MSX-E-Handbuch.

Nachfolgend wird beschrieben, wie mit einem Hardware-Trigger der Start der Erfassung gesteuert werden kann. Jedes Mal, wenn eine steigende Flanke am Trigger-Eingang des MSX-E-Systems erkannt wird, sendet das MSX-E-System ein Paket mit einem Sequenz-Zähler-Wert und acht Messwerten.

5.2.1 Systemkonfiguration

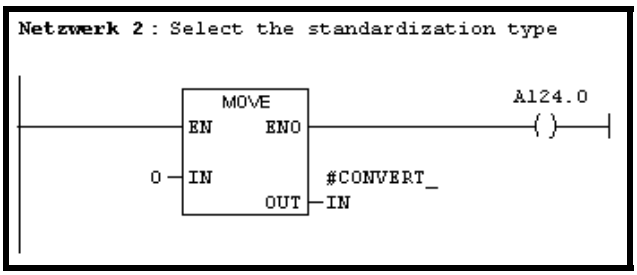
Die in Kap. 4.1.1 (Auto-Refresh-Modus) bzw. Kap. 4.1.2 (Sequenz-Modus) beschriebene Konfiguration kann als Basis verwendet werden, um folgende Trigger-Konfiguration zu definieren:

	Trigger source	Trigger mode	Number of sequences per trigger
Description	Trigger mask (API)		Number of sequences to be acquired at each trigger event
Value	Hardware	Sequence	1 (1 - 65535)

	Hardware trigger active edge	Hardware trigger count
Description		Number of trigger events before the acquisition starts
Value	Rising	1 (1 - 65535)

5.2.2 BasicSample: Beispiel für die Trigger-Erweiterung

Um die Erfassung zu starten, können Sie in **Netzwerk 2** einen digitalen Ausgang (A124.0) der Systeme setzen. Dieser muss mit dem digitalen Trigger-Eingang des Systems **(Trig/Sync In)** verbunden sein.



5.3 Abfrage der aktuellen Messwerte im Auto-Refresh-Modus

Die SPS bietet die Möglichkeit, einen oder mehrere aktuelle Messwerte asynchron abzufragen (ohne Hardware-Trigger). Diese Methode kann verwendet werden, wenn nur einer bzw. mehrere aktuelle Messwerte des Systems zu einem bestimmten Zeitpunkt benötig werden.

Im folgenden Kapitel wird beschrieben, wie ein aktuelles Paket mit einem Sequenz-Zähler-Wert und 8 Messwerten gelesen wird. Zu diesem Zweck wird das **BasicSample** entsprechend angepasst (siehe Kap. 5.3.2).

Es ist zu beachten, dass sich die Übertragungszeit verlängert, falls mehrere SPSen gleichzeitig auf ein MSX-E-System zugreifen.

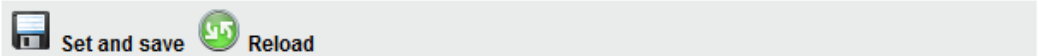
5.3.1 Systemkonfiguration

Die in Kap. 4.1.1 (Auto-Refresh-Modus) bzw. Kap. 4.1.2 (Sequenz-Modus) beschriebene Konfiguration kann als Basis verwendet werden.

- Klicken Sie auf der Weboberfläche auf den Menüpunkt **Data server**.
- Wählen Sie auf der Registerkarte **Blocking transfer** bei **Activate blocking TCP/IP transfer** die Option **Yes** aus.

Bei **TCP/IP transfer timeout** können Sie ein Timeout von 2 Sekunden einstellen, damit der Datenserver nicht dauerhaft blockiert.

- Klicken Sie auf **Set and Save** und danach auf **Reload**.

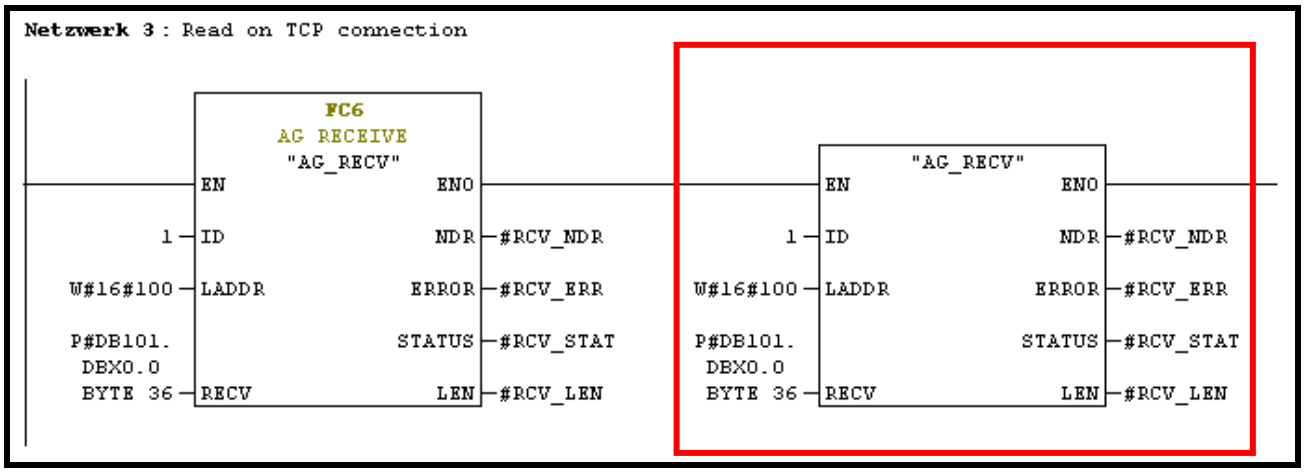


5.3.2 BasicSample: Anpassungsbeispiel zur Abfrage der Messwerte

Um die aktuellen Werte lesen zu können, müssen zwei Datenpakete erfasst werden. Dabei wird nur das zweite Paket berücksichtigt, da das erste Paket frühere Messwerte enthält.

Das **BasicSample** wird wie folgt angepasst:

- Kopieren Sie in **Netzwerk 3** den Funktionsbaustein FC6 und fügen Sie diesen rechts neben dem Originalbaustein ein.

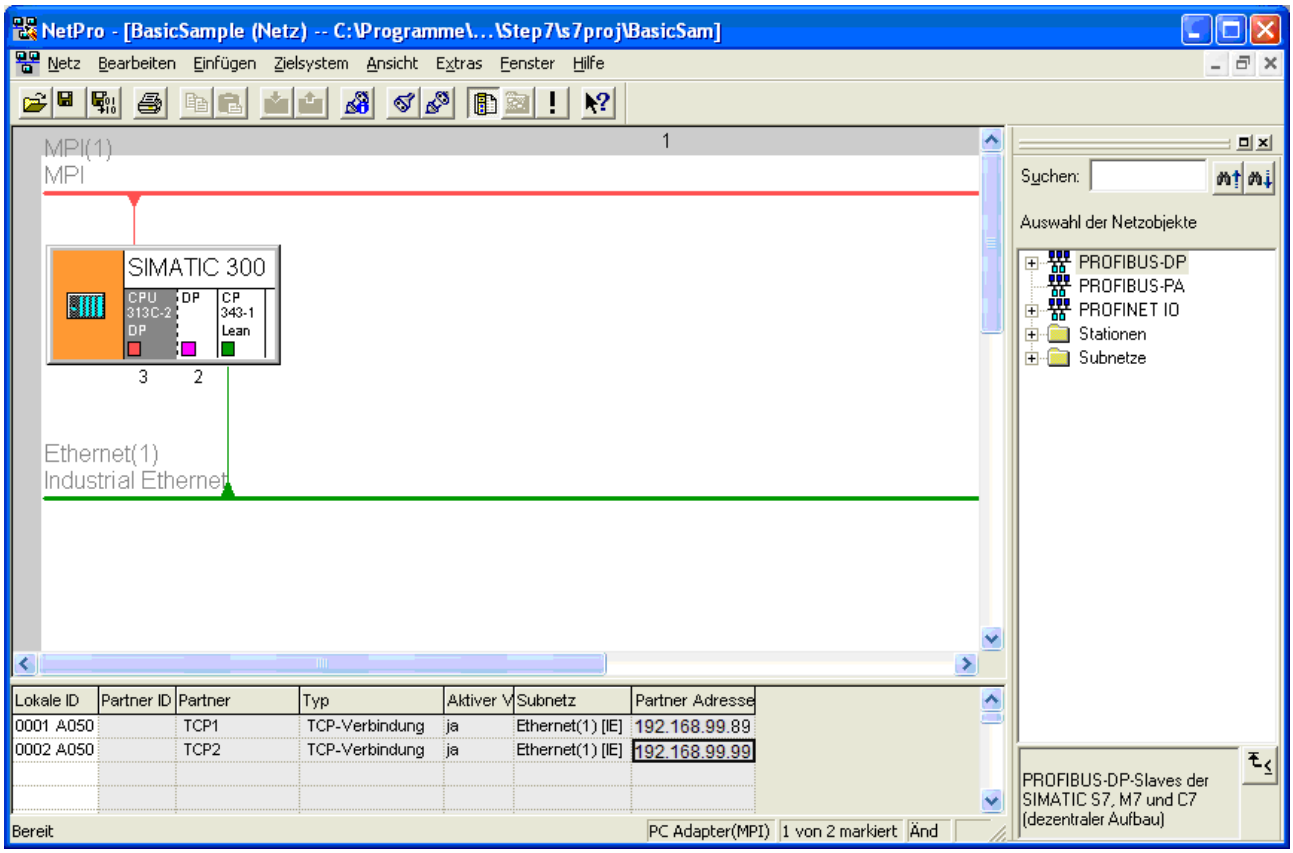


5.4 Anschließen mehrerer Systeme an eine SPS

Sie können mehrere MSX-E-Systeme an eine SPS anschließen. In diesem Fall müssen Sie für jedes System eine neue TCP-Verbindung (Socket) herstellen.

- Befolgen Sie hierzu bitte die Anweisungen in Kap. 2.2 und geben Sie im Fenster **Eigenschaften – TCP-Verbindung** (siehe Abb. 2-4) jedem System eine andere Portnummer.

Abb. 5-3: SIMATIC Manager: S7-Ethernet-Konfiguration



6 Anhang

6.1 Glossar

AWL
= Anweisungsliste

AWL ist eine Programmiersprache nach IEC 61131-3 zur SPS-Programmierung.

Datenbaustein (DB)
Bei Datenbausteinen handelt es sich um Datenbereiche im Siemens STEP 7-System, in denen die Daten des Anwenderprogramms gespeichert werden. Es gibt zwei Datenbaustein-Typen: Instanz-Datenbausteine und Global-Datenbausteine. Erstere enthalten statische Lokaldaten, letztere Informationen, auf die ein Zugriff von allen Code-Bausteinen aus erfolgen kann. Instanz-Datenbausteine werden bei der Erstellung von Funktionsbausteinen (FBs) erzeugt, während Global-Datenbausteine vom Anwender programmiert werden müssen.

Ethernet
Hierbei handelt es sich um ein Basisband-Bussystem, das ursprünglich für die Verknüpfung von Minicomputern entwickelt wurde. Es basiert auf dem CSMA/CD-Zugriffs-verfahren. Als Übertragungsmedium dienen Koaxialkabel bzw. Twisted-Pair-Leitungen.
Die Übertragungsgeschwindigkeiten betragen 10 Mbit/s (Ethernet), 100 Mbit/s (Fast Ethernet) sowie 1 Gbit/s bzw. 10 Gbit/s (Gigabit-Ethernet). Diese weit verbreitete Technik zum Vernetzen von Rechnern in einem LAN ist seit 1985 genormt (IEEE 802.3 und ISO 8802-3).
Die Ethernet-Technologie hat sich im Büro-bereich allgemein durchgesetzt. Nach Ermöglichung auch sehr harter Echtzeit-anforderungen und Anpassung der Geräte-technik (Buskabel, Patchfelder, Anschlussdosen) an die rauen Einsatzbedingungen des industriellen Umfelds dringt sie zunehmend in die Feldbereiche der Automatisierungstechnik vor.

FIFO
= First in, First out

Organisationsprinzip für die Bedienung von Warteschlangen, bei dem die Abarbeitung von Aufträgen in der gleichen Reihenfolge erfolgt wie die Annahme. So werden z.B. beim Leeren eines Speichers zuerst eingespeicherte Daten als erste wieder ausgegeben.

Funktion (FC)
Im Siemens STEP 7-System gibt es den Code-Bausteintyp „Funktionen“, der Funktionen strukturierter Anwenderprogramme enthält. Dieser Code-Bausteintyp ist parametrierbar und kann Lokaldaten temporär speichern.

Funktionsbaustein (FB)
Funktionsbausteine strukturierter Anwender-programme zählen im Siemens STEP 7-System zu den Code-Bausteinen und enthalten aufgabenspezifische Unterprogramme. Zu einem Funktionsbaustein gehört immer ein Instanz-Datenbaustein, d.h. ein eigener Speicherbereich für spezielle Daten. FBs sind parametrierbar und können Lokaldaten sowohl statisch als auch temporär speichern.

IP-Adresse
Die IP-Adresse ist eine numerische Adresse, die einem Rechner im Internet zugeordnet ist und diesen eindeutig identifizierbar macht.
Sie besteht aus einer Folge von vier Zahlen-gruppen mit je maximal drei Ziffern, die jeweils durch Punkte voneinander getrennt sind. Fest mit dem Internet verbundene Rechner haben eine feste IP-Adresse. Benutzer, die sich von Zeit zu Zeit ins Netz einwählen, erhalten vom Provider eine dynamische IP-Adresse, die nach Beendigung der Sitzung an den nächsten Client weitervergeben wird.

LSB
= Least Significant Bit

Die Bits sind unterschiedlich gewichtet, wobei das am weitesten rechts stehende die geringste Wertigkeit hat. Es wird daher als „Least Significant Bit“ (LSB) bezeichnet. Das erste Bit wird als „Most Significant Bit“ (MSB) bezeichnet, da es den höchsten Wert hat.

MSB
= Most Significant Bit

Die Bits sind unterschiedlich gewichtet. Das erste Bit hat den höchsten Wert und wird als „Most Significant Bit“ (MSB) bezeichnet.

Organisationsbaustein (OB)
Organisationsbausteine gehören im Siemens STEP 7-System zu den Code-Bausteinen und bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. OBs können durch das Betriebssystem in bestimmten Situationen aufgerufen werden, wenn der sonst endlos ausgeführte OB1 unterbrochen wird.

Socket
Die bidirektionale Softwareschnittstelle zur Interprozess- (IPC) oder Netzwerk-kommunikation wird Socket genannt. Sockets bilden eine standardisierte Schnittstelle (API) zwischen dem Betriebssystem und der eigentlichen Anwendungssoftware.

SPS
= Speicherprogrammierbare Steuerung

Die SPS ist ein rechnerbasiertes Steuergerät, dessen Funktionalität durch ein Anwender-programm festgelegt wird. Mittels genormter Fachsprachen ist dieses Anwenderprogramm relativ einfach zu erstellen. Aufgrund der seriellen Arbeitsweise sind die Reaktionszeiten der SPS langsamer als bei VPS. Als Geräte-familien mit abgestuften und leistungsmäßig aufeinander abgestimmten Komponenten beherrschen die SPS heute alle Ebenen einer Automatisierungshierarchie.

Trigger
Der Trigger ist ein Impuls oder Signal zum Starten oder Stoppen einer besonderen Aufgabe. Trigger werden häufig zur Steuerung des Datenerfassungsbetriebs eingesetzt.

Zähler
Der Zähler ist ein Schaltkreis, der Impulse zählt oder die Dauer von Impulsen messen kann.

6.2 Index

Anschließen mehrerer Systeme 42
Datenbaustein 7
Datenformat
  Intel-Format 9
  Motorola-Format 9
Datenkonvertierung 8, 16
  Big-Endian-Format 9
  Little-Endian-Format 9
Dynamisches Lesen 20
Dynamisches Schreiben 21
Empfehlungen 38
Funktionsbausteine
  FC1 16
  FC20 20
  FC21 21
  FC3 17
  FC4 17
  FC6 21
  FC7 19
Glossar 43
Hauptbaustein 29
Konfiguration
  Ethernet – S7 10
  MSX-E-System 24
Konvertierung
  Datentyp 17
  Einheit 17
  Millimeter 17, 33
  Volt 19, 34
Motorola-Format 16, 32
Netzwerk 1 29
Netzwerk 10 34
Netzwerk 11 35
Netzwerk 2 30
Netzwerk 3 30
Netzwerk 4 31
Netzwerk 5 32
Netzwerk 6 32
Netzwerk 7 33
Netzwerk 8 33
Netzwerk 9 34
OB1 29
Organisationsbaustein 29
Programmübersicht 8
TIA Portal V13 36
Trigger 40
Trigger-Erweiterung 40
Typecast 17
Typecast 33
VIPA CPU 015 36
Voraussetzungen S7 15
Weboberfläche
  Auto-Refresh-Modus 24
  Sequenz-Modus 26

7 Kontakt und Support

Haben Sie Fragen? Schreiben Sie uns oder rufen Sie uns an:

Postanschrift: ADDI-DATA GmbH
Airpark Business Center
Airport Boulevard B210
77836 Rheinmünster
Deutschland

Telefon: +49 7229 1847-0
Fax: +49 7229 1847-222

E-Mail: info@addi-data.com

Handbuch- und Software-Download im Internet:

www.addi-data.de

ADDI-DATA®
PARTNER FÜR PRÄZISION

Praktische Anleitung

Intelligente Ethernet-E/A-Systeme (Teil 3)

ADDI-DATA Modbus-TCP-Client-Bibliothek
für eine SPS SIMATIC® S7®

DIN EN ISO 9001:2015 zertifiziert

Ausgabe: 02.02-09/2018

Produktinformation
Dieses Handbuch enthält die technischen Anlagen, wichtige Anleitungen zur korrekten Inbetriebnahme und Nutzung sowie Produktinformation entsprechend dem aktuellen Stand vor der Drucklegung.
Der Inhalt dieses Handbuchs und die technischen Daten des Produkts können ohne vorherige Ankündigung geändert werden. Die ADDI-DATA GmbH behält sich das Recht vor, Änderungen bzgl. der technischen Daten und der hierin enthaltenen Materialien vorzunehmen.

Gewährleistung und Haftung
Der Nutzer ist nicht berechtigt, über die vorgesehene Nutzung der Karte hinaus Änderungen des Werks vorzunehmen sowie in sonstiger Form in das Werk einzugreifen.
ADDI-DATA übernimmt keine Haftung bei offensichtlichen Druck- und Satzfehlern. Darüber hinaus übernimmt ADDI-DATA, soweit gesetzlich zulässig, weiterhin keine Haftung für Personen- und Sachschäden, die darauf zurückzuführen sind, dass der Nutzer die Karte unsachgemäß installiert und/oder in Betrieb genommen oder bestimmungswidrig verwendet hat, etwa indem die Karte trotz nicht funktionsfähiger Sicherheits- und Schutzvorrichtungen betrieben wird oder Hinweise in der Betriebsanleitung bzgl. Transport, Lagerung, Einbau, Inbetriebnahme, Betrieb, Grenzwerte usw. nicht beachtet werden. Die Haftung ist ferner ausgeschlossen, wenn der Betreiber die Karte oder die Quellcode-Dateien unbefugt verändert und/oder die ständige Funktionsbereitschaft von Verschleißteilen vorwerfbar nicht überwacht wurde und dies zu einem Schaden geführt hat.

Urheberrecht
Dieses Handbuch, das nur für den Betreiber und dessen Personal bestimmt ist, ist urheberrechtlich geschützt. Die in der Betriebsanleitung und der sonstigen Produktinformation enthaltenen Hinweise dürfen vom Nutzer des Handbuchs weder vervielfältigt noch verbreitet und/oder Dritten zur Nutzung überlassen werden, soweit nicht die Rechtsübertragung im Rahmen der eingeräumten Produktlizenz gestattet ist. Zuwiderhandlungen können zivil- und strafrechtliche Folgen nach sich ziehen.

ADDI-DATA-Software-Produktlizenz
Bitte lesen Sie diese Lizenz sorgfältig durch, bevor Sie die Standardsoftware verwenden.
Das Recht zur Benutzung dieser Software wird dem Kunden nur dann gewährt, wenn er den Bedingungen dieser Lizenz zustimmt.
Die Software darf nur zur Einstellung der ADDI-DATA Karten verwendet werden.
Das Kopieren der Software ist verboten (außer zur Archivierung/Datensicherung und zum Austausch defekter Datenträger). Disassemblierung, Dekompilierung, Entschlüsselung und Reverse Engineering der Software ist verboten. Diese Lizenz und die Software können an eine dritte Partei übertragen werden, sofern diese Partei eine Karte käuflich erworben hat, sich mit allen Bestimmungen in diesem Lizenzvertrag einverstanden erklärt und der ursprüngliche Besitzer keine Kopien der Software zurückhält.

Warenzeichen
- ADDI-DATA, APCI-1500, MSX-Box und MSX-E sind eingetragene Warenzeichen der ADDI-DATA GmbH.
- Turbo Pascal, Delphi, Borland C, Borland C++ sind eingetragene Warenzeichen der Borland Software Corporation.
- Microsoft .NET, Microsoft C, Visual C++, MS-DOS, Windows XP, Windows 7, Windows 8, Windows 10, Windows Server 2000, Windows Server 2003, Windows Embedded und Internet Explorer sind eingetragene Warenzeichen der Microsoft Corporation.
- LabVIEW, LabWindows/CVI, DASYLab, DIAdem sind eingetragene Warenzeichen der National Instruments Corporation.
- CompactPCI ist ein eingetragenes Warenzeichen der PCI Industrial Computer Manufacturers Group.
- VxWorks ist ein eingetragenes Warenzeichen von Wind River Systems, Inc.
- RTX ist ein eingetragenes Warenzeichen von IntervalZero.
- Mozilla Firefox ist ein eingetragenes Warenzeichen der Mozilla Foundation.
- SIMATIC S7, S7-300, STEP 7, CPU313C-2DP und CP343-1 Lean sind eingetragene Warenzeichen der Siemens AG.

Warnung!

Bei unsachgemäßem Einsatz und bestimmungswidrigem Gebrauch des Ethernet-Systems

können Personen verletzt werden

können Ethernet-System, PC und Peripherie beschädigt werden

kann die Umwelt verunreinigt werden.

- Schützen Sie sich, andere und die Umwelt!
- Lesen Sie unbedingt die Sicherheitshinweise (gelbe Broschüre)!

  Liegen Ihnen keine Sicherheitshinweise vor, so fordern Sie diese bitte an.
- Beachten Sie die Anweisungen dieses Handbuchs!

  Vergewissern Sie sich, dass Sie keinen Schritt vergessen oder übersprungen haben!
  Wir übernehmen keine Verantwortung für Schäden, die aus dem falschen Einsatz des Ethernet-Systems hervorgehen könnten.
- Beachten Sie folgende Symbole:

HINWEIS!
Kennzeichnet Anwendungstipps und andere nützliche Informationen.

ACHTUNG!
Bezeichnet eine möglicherweise gefährliche Situation.
Bei Nichtbeachten des Hinweises können Ethernet-System, PC und/oder Peripherie **zerstört** werden.

WARNUNG!
Bezeichnet eine möglicherweise gefährliche Situation.
Bei Nichtbeachten des Hinweises können Ethernet-System, PC und/oder Peripherie **zerstört** und Personen **gefährdet** werden.

Inhaltsverzeichnis

Warnung! ... 3
Kapitelübersicht ... 6
1 ADDI-DATA Modbus-TCP-Client-Bibliothek ... 7
1.1 Kurzbeschreibung ... 7
1.2 Modbus-TCP-Protokoll ... 7
1.3 Softwarepaket ... 7
1.4 Struktur der ADModbus-Bibliothek ... 8
1.5 Zustandsmaschine ... 9
2 MSX-E-Modbus-TCP-Funktionen ... 10
2.1 Funktionsbausteine ... 10
2.2 Funktionsnamen ... 11
2.3 Beschreibung der Funktionsparameter ... 12
2.3.1 SPS- und Modbus-TCP-spezifische Parameter ... 13
2.3.2 Allgemeine Parameter der MSX-E-Modbus-TCP-Funktionen ... 15
2.3.3 MSX-E-System-Fehler ... 17
3 ADModbus-Bausteine ... 19
3.1.1 Verwendung ... 19
3.1.2 OB1 und OB100 ... 19
4 Einbindung der ADModbus-Bibliothek ... 20
4.1 Funktionen und Bausteine ... 20
4.2 Voraussetzungen ... 21
4.3 Einbindung in den SIMATIC Manager ... 21
5 SPS-Konfiguration ... 22
5.1 Taktmerker ... 22
5.2 TCP-Verbindung ... 23
5.2.1 Modbus-Server ... 23
5.2.2 Datenserver ... 28
6 Funktionsbausteine ... 32
6.1 Umsetzung der AWL-Quellcode-Dateien in Funktionsbausteine ... 32
6.2 Mnemonische Symbole ... 35
7 Lesen von Daten aus dem MSX-E-Datenserver ... 37
7.1 Datenkonvertierung ... 37
7.1.1 Little-Endian-Format (Intel-Format) ... 37
7.1.2 Big-Endian-Format (Motorola-Format) ... 37
7.2 Funktion „FC6 (AG_RECV)“ ... 38
8 S7-Programmierbeispiel ... 41
8.1 Einführung ... 41
8.2 Voraussetzungen ... 41
9 Auto-Refresh-Modus und Sequenz-Modus ... 43
9.1 Erfassung im Auto-Refresh-Modus ... 43
9.2 Erfassung im Sequenz-Modus ... 44
10 Anhang ... 46
10.1 Glossar ... 46
10.2 Index ... 48
11 Kontakt und Support ... 49

Abbildungen

Abb. 1-1: ADModbus-Bibliothek: Struktur-Überblick 8
Abb. 1-2: Zustandsmaschine: Beispiel 9
Abb. 2-1: MSX-E370x-Funktionen: AWL-Quellcode-Dateien 10
Abb. 2-2: MSX-E370x-Funktionen: Funktionsbausteine 10
Abb. 2-3: MSX-E-Modbus-TCP-Funktionen und entsprechende SPS-Funktionsbausteine 11
Abb. 2-4: Teil des Funktionsbausteins „370x_InitStartSeq“ 12
Abb. 2-5: Spezifische Parameter des Funktionsbausteins „370x_InitStartSeq“ 13
Abb. 2-6: Eigenschaften – TCP-Verbindung: Eingabe-Parameter 14
Abb. 2-7: Allgemeine Parameter des Funktionsbausteins „370x_InitStartSeq“ 15
Abb. 2-8: 370x_InitStartSeq: Beschreibung der Eingabe-Parameter 16
Abb. 2-9: 370x_InitStartSeq: Beschreibung der Ausgabe-Parameter 16
Abb. 2-10: GetLastCommandStatus: Funktionsbaustein 17
Abb. 2-11: InitStartSeq: Rückgabewerte 18
Abb. 3-1: ADModbus-Bausteine 19
Abb. 4-1: ADModbus-Bibliothek: Funktionen und Bausteine 20
Abb. 5-1: SIMATIC Manager: S7-Taktmerker-Konfiguration 22
Abb. 5-2: Ethernet-Konfiguration: TCP-Verbindung 23
Abb. 5-3: Ethernet-Konfiguration: Modbus-Server 24
Abb. 5-4: Eigenschaften – TCP-Verbindung: Allgemein 25
Abb. 5-5: Eigenschaften – TCP-Verbindung: Adressen 26
Abb. 5-6: Login-Fenster 27
Abb. 5-7: Modbus server: Configuration 27
Abb. 5-8: Modbus server: Set and save 27
Abb. 5-9: Ethernet-Konfiguration: Datenserver 28
Abb. 5-10: Eigenschaften – TCP-Verbindung: Allgemein 29
Abb. 5-11: Eigenschaften – TCP-Verbindung: Adressen 30
Abb. 5-12: Login-Fenster 30
Abb. 5-13: Data server: Configuration 31
Abb. 5-14: Data server: Set and save 31
Abb. 5-15: Data server: Blocking transfer 31
Abb. 6-1: SIMATIC Manager: Quellenverzeichnis 32
Abb. 6-2: SIMATIC Manager: Auswahl der AWL-Quellcode-Dateien 33
Abb. 6-3: SIMATIC Manager: AWL-Quellcode-Dateien 33
Abb. 6-4: SIMATIC Manager: Bausteine 35
Abb. 6-5: SIMATIC Manager: Mnemonische Symbole 36
Abb. 7-1: Eigenschaften – TCP-Verbindung: Eingabe-Parameter 38
Abb. 7-2: Acquisition: Data server frame format 39
Abb. 7-3: Datenbaustein mit der Tabelle „DATA_SERVER“ 39
Abb. 7-4: Tabelle „DATA_SERVER“ im Datenbaustein 40
Abb. 8-1: S7: LED-Visualisierung der digitalen Ausgänge 41
Abb. 8-2: SIMATIC Manager: LED-Konfiguration der digitalen Ausgänge 42
Abb. 9-1: Auto-Refresh-Modus: Erfassungszyklus 43
Abb. 9-2: Sequenz-Modus: Erfassungszyklus 44

Kapitelübersicht

In dieser Anleitung finden Sie folgende Informationen:

Kapitel	Inhalt
1	Allgemeine Informationen zur ADModbus-Bibliothek, dem Modbus-TCP-Protokoll etc.
2	Beschreibung der MSX-E-Modbus-TCP-Funktionen
3	Hinweise zu den ADModbus-Bausteinen
4	Informationen zur Einbindung der ADModbus-Bibliothek
5	Vorgehensweise bei der Konfiguration der SPS
6	Importieren und Kompilieren der MSX-E-Modbus-TCP-Funktionen
7	Erläuterung des Lesens von Daten aus dem MSX-E-Datenserver
8	Hinweise zum S7-Programmierbeispiel
9	Informationen zur Messwerterfassung im Auto-Refresh- bzw. Sequenz-Modus
10	Anhang mit Glossar und Index
11	Kontakt- und Support-Adresse

1 ADDI-DATA Modbus-TCP-Client-Bibliothek

1.1 Kurzbeschreibung

Modbus TCP ist ein herstellerunabhängiges Kommunikationsprotokoll, das in der Industriewelt von unterschiedlichen Gerätetypen, wie z.B. einer SPS, unterstützt wird.
Die MSX-E-Systeme von ADDI-DATA verfügen über einen Modbus-TCP-Server und können mit Hilfe einer Modbus-TCP-Client-Bibliothek[1] direkt von einer SPS aus verwaltet und gesteuert werden. Über Telegramme kann die SPS beispielsweise Systeminformationen lesen und Messungen starten bzw. stoppen.
Für jede Funktion der MSX-E-Systeme gibt es einen Funktionsbaustein (FB) nach der Norm IEC 61131-3, der in den Standard-SIMATIC Manager importiert werden kann. Da die Funktionsbausteine an den Modbus-TCP-Server der MSX-E-Systeme angepasst sind, ist eine schnelle Konfiguration der MSX-E-Systeme im SIMATIC Manager möglich.

1.2 Modbus-TCP-Protokoll

Das Modbus-TCP-Protokoll ist ein Anfrage/Antwort-Protokoll (Query/Response), das mit Hilfe definierter Klassen (FC = Function Code) verschiedene Aufgaben ausführen kann.
Der Modbus-TCP-Server der MSX-E-Systeme unterstützt die Basis-Klasse „Class 0“. Diese besteht aus den Modbus-TCP-Funktionen, die ein Client und ein Server mindestens verwenden sollten:

- **FC3** („Read multiple registers“): Inhalt mehrerer Register lesen
- **FC16** („Write multiple registers“): in mehrere Register schreiben.

1.3 Softwarepaket

Das auf einer CD mitgelieferte Softwarepaket ist wie folgt aufgebaut:

- **MXXXXX{0000}_XXXX_modbus_client**
  - **Documentation**
    - **html** (Beschreibung der MSX-E-Modbus-TCP-Funktionen)
  - **S7**
    - **src** (AWL-Quellcode-Dateien der MSX-E-Modbus-TCP-Funktionen)
    - **samples.zip** (S7-Programmierbeispiel zur Nutzung der ADModbus-Bibliothek)

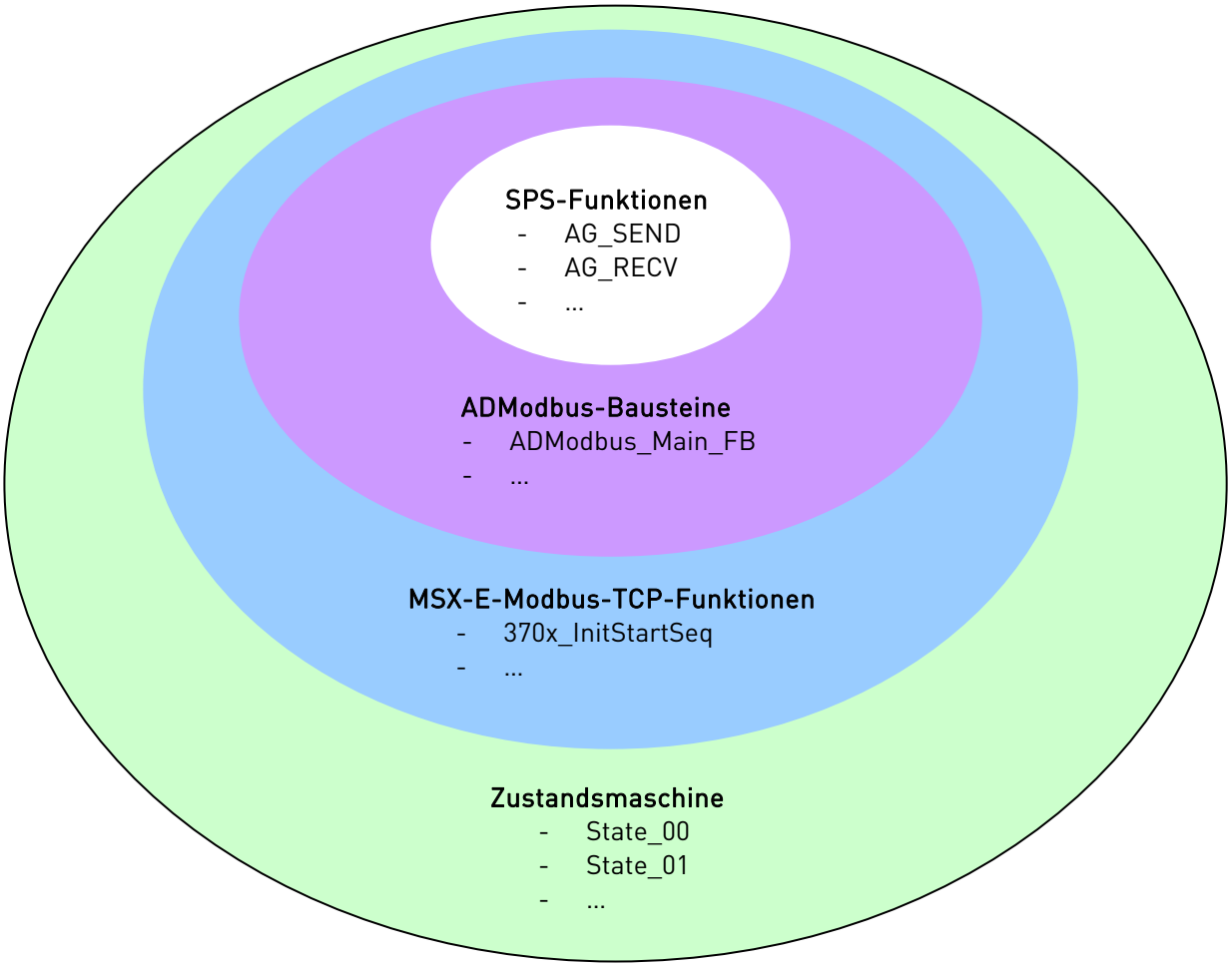
[1] Kurzform: ADModbus-Bibliothek

1.4 Struktur der ADModbus-Bibliothek

Die ADModbus-Bibliothek besteht aus drei Programmierebenen: SPS-Funktionen, ADModbus-Bausteine und MSX-E-Modbus-TCP-Funktionen. ADDI-DATA bietet letztere zur Programmierung der Kommunikation zwischen einer SPS und einem MSX-E-System über Modbus TCP.

Das Modbus-TCP-Protokoll wird von der ADModbus-Bibliothek automatisch verwaltet, so dass der Anwender keine Modbus-TCP-Kenntnisse benötigt, um die MSX-E-Modbus-TCP-Funktionen nutzen zu können.

Abb. 1-1: ADModbus-Bibliothek: Struktur-Überblick

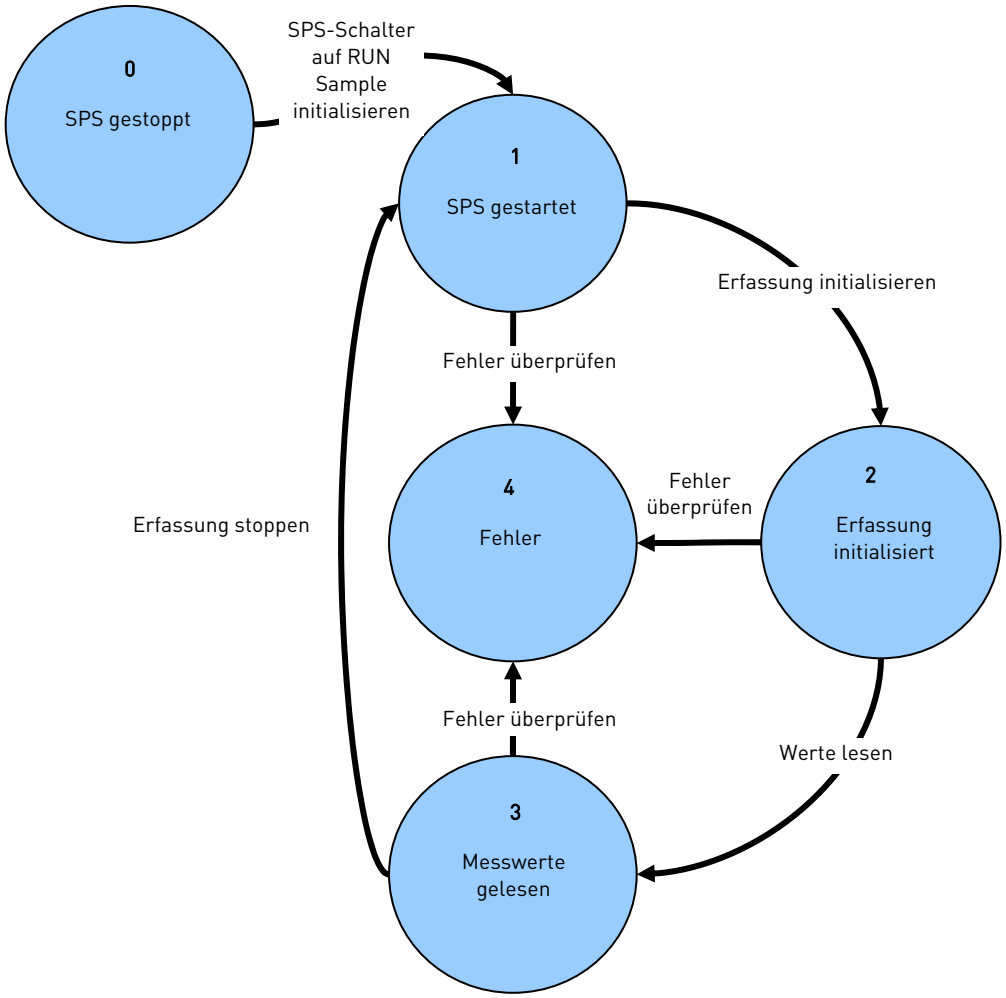


1.5 Zustandsmaschine

Die Zustandsmaschine gehört nur indirekt zur ADModbus-Bibliothek. Dennoch spielt sie eine wichtige Rolle, weil sie die Funktionsaufrufe in einer definierten Reihenfolge verwaltet.

Beispielsweise können die erfassten Daten eines Systems **MSX-E-370x** mit folgender Zustandsmaschine gelesen werden:

Abb. 1-2: Zustandsmaschine: Beispiel



HINWEIS!
Pro Zustand darf maximal eine MSX-E-Modbus-TCP-Funktion aufgerufen werden.

2 MSX-E-Modbus-TCP-Funktionen

2.1 Funktionsbausteine

Die MSX-E-Modbus-TCP-Funktionen sind für jedes MSX-E-System, das einen Modbus-Server enthält, in Form von AWL-Quellcode-Dateien verfügbar.

Abb. 2-1: MSX-E370x-Funktionen: AWL-Quellcode-Dateien

TestCustomerID.awl	10 KB	AWL-Datei
StopRelSyncTimer.awl	4 KB	AWL-Datei
SetHwTrigFiltTime.awl	5 KB	AWL-Datei
SetCustomerKey.awl	11 KB	AWL-Datei
Reboot.awl	4 KB	AWL-Datei
mnemonics.sdf	5 KB	SQL Server Compact Edition Database File
InitStartSyncTimer.awl	8 KB	AWL-Datei
GetTime.awl	4 KB	AWL-Datei
GetModuleType.awl	31 KB	AWL-Datei
GetLastCommandStatus.awl	20 KB	AWL-Datei
370x_StopRelSeq.awl	4 KB	AWL-Datei
370x_StopRelAutoRef.awl	4 KB	AWL-Datei
370x_SetDataBaseCursor.awl	4 KB	AWL-Datei
370x_InitStartSeq.awl	22 KB	AWL-Datei
370x_InitStartAutoRef.awl	11 KB	AWL-Datei
370x_GetTypeInfo.awl	22 KB	AWL-Datei
370x_GetNbrOfType.awl	4 KB	AWL-Datei
370x_GetNbrOfChannels.awl	4 KB	AWL-Datei
370x_GetDataBaseCursor.awl	4 KB	AWL-Datei
370x_GetAutoRefVal.awl	12 KB	AWL-Datei

Die AWL-Quellcode-Dateien können in den SIMATIC Manager importiert und zu Funktions- und Datenbausteinen kompiliert werden. Jeder Funktionsbaustein funktioniert mit einem Instanz-Datenbaustein (Name = I wie **I**nstanz + Name der MSX-E-Modbus-TCP-Funktion) und einem Global-Datenbaustein (Name = SR wie **S**end **R**eceived + Name der MSX-E-Modbus-TCP-Funktion). Im Global-Datenbaustein befinden sich die Anfrage- und Antwort-Telegramme.
Ein Teil der im Programmierbeispiel enthaltenen MSX-E-Modbus-TCP-Funktionen ist bereits kompiliert. Weitere Informationen zum Importieren der MSX-E-Modbus-TCP-Funktionen finden Sie in Kap. 6.

Abb. 2-2: MSX-E370x-Funktionen: Funktionsbausteine

DB102	SRGetLastCommandStatus	DB
DB218	SR370x_StopRelSeq	DB
DB216	SR370x_InitStartSeq	DB
DB103	IGetLastCommandStatus	DB
DB219	I370x_StopRelSeq	DB
DB217	I370x_InitStartSeq	DB
FB102	GetLastCommandStatus	AWL
FB218	370x_StopRelSeq	AWL
FB216	370x_InitStartSeq	AWL

2.2 Funktionsnamen

Die Namen der MSX-E-Modbus-TCP-Funktionen wurden aufgrund der Zeichenbegrenzung im SIMATIC Manager automatisch abgekürzt.
Um die Beschreibung einer bestimmten Funktion in der Dokumentation „msxeXXXX_modbus.html“ (siehe Kap. 1.3) finden zu können, müssen Sie unter Umständen erst in der Tabelle im Kapitel „Siemens STEP 7 compatibility information (AWL/SDF code)“ (siehe folgende Abbildung) prüfen, welcher SPS-Funktionsbaustein zu welcher MSX-E-Modbus-TCP-Funktion gehört.

Beispiel:

„370x_InitStartSeq“ ist der Name eines SPS-Funktionsbausteins, der in der rechten Spalte der Tabelle zu finden ist. In der linken Spalte steht der Name der entsprechenden MSX-E-Modbus-TCP-Funktion.

Abb. 2-3: MSX-E-Modbus-TCP-Funktionen und entsprechende SPS-Funktionsbausteine

6 SIEMENS STEP 7 COMPATIBILITY INFORMATION (AWL/SDF CODE)

Top

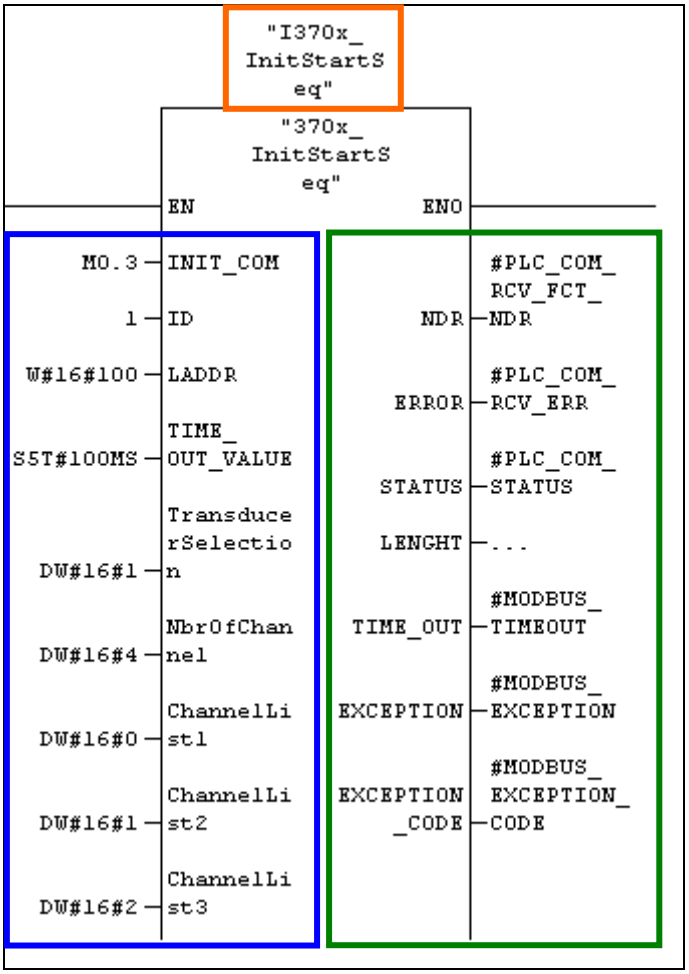
Due to limitations of the S7 platform, some names of function and parameter have been shortened in the AWL and S7 code. This table summarizes the changes against the standard version as described above.

Function/Parameter	Renamed as
MXCommon__GetModuleType	GetModuleType
MXCommon__GetTime	GetTime
MXCommon__TestCustomerID	TestCustomerID
MX370x__getNumberOfChannels	370x_GetNbrOfChannels
MX370x__TransducerGetAutoRefreshValues	370x_GetAutoRefVal
MX370x__TransducerGetNbrOfType	370x_GetNbrOfType
MX370x__GetTransducerDatabaseCursor	370x_GetDataBaseCursor
MX370x__TransducerGetTypeInformation	370x_GetTypeInfo
MXCommon__SetHardwareTriggerFilterTime	SetHwTrigFiltTime
MXCommon__InitAndStartSynchroTimer	InitStartSyncTimer
MXCommon__StopAndReleaseSynchroTimer	StopRelSyncTimer
MXCommon__Reboot	Reboot
MXCommon__SetCustomerKey	SetCustomerKey
MX370x__TransducerInitAndStartAutoRefresh	370x_InitStartAutoRef
HardwareTriggerCount	HwTrigCount
HardwareTriggerFilterTime	HwTrigFilterTime
ByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MX370x__TransducerInitAndStartSequence	370x_InitStartSeq
HardwareTriggerCount	HwTrigCount
HardwareTriggerFilterTime	HwTrigFilterTime
NbrMaxSequenceToTransfer	NbrMaxSeqToTransfer
ByTriggerNbrOfSeqToAcquire	ByTrigNbrOfSeqToAcq
MX370x__TransducerStopAndReleaseSequence	370x_StopRelSeq
MX370x__SetTransducerDatabaseCursor	370x_SetDataBaseCursor

2.3 Beschreibung der Funktionsparameter

Als Beispiel dient der Funktionsbaustein der MSX-E-Modbus-TCP-Funktion „370x_InitStartSeq“.

Abb. 2-4: Teil des Funktionsbausteins „370x_InitStartSeq“

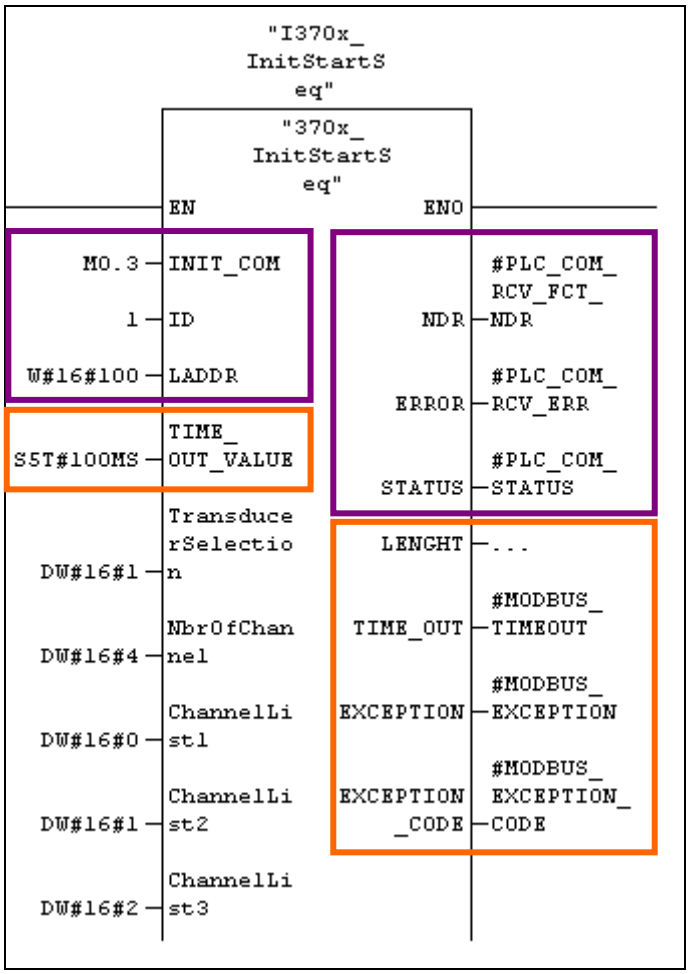


Die **Eingabe-Parameter** stehen immer links; die **Ausgabe-Parameter** immer rechts. Der **Instanz-Datenbaustein** befindet sich oben in der Mitte.

2.3.1 SPS- und Modbus-TCP-spezifische Parameter

Die MSX-E-Modbus-TCP-Funktionen enthalten **S7-spezifische Parameter**. Diese Parameter gehören zu der SPS-Funktion „AG_SEND“ bzw. „AG_RECV“.
Bei den orange markierten Parametern handelt es sich um die **Modbus-TCP-spezifischen**, die bei allen MSX-E-Modbus-TCP-Funktionen gleich sind.
Die spezifischen Parameter werden immer in Großbuchstaben geschrieben.

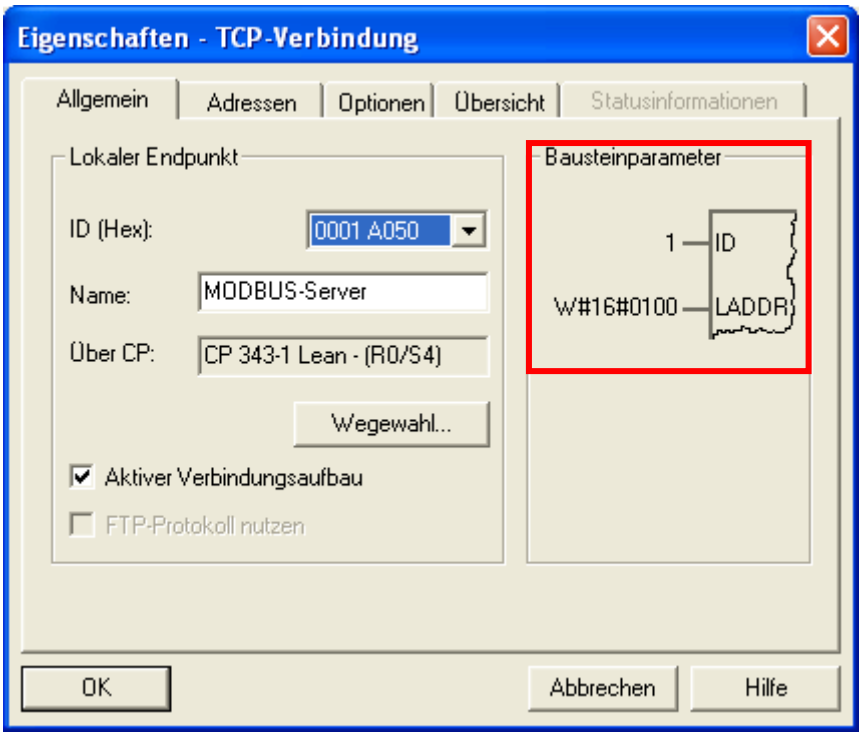
Abb. 2-5: Spezifische Parameter des Funktionsbausteins „370x_InitStartSeq“



Eingabe-Parameter:

- **INIT_COM:** Wird dieses Bit auf 1 gesetzt, so werden die Eingabe-Parameter der Funktion „AG_SEND“ übernommen. Wenn das Bit auf 0 gesetzt wird, werden die Ausgabe-Parameter aktualisiert (siehe SIMATIC Manager-Hilfe: „AG_SEND ACT“-Parameter).
- **ID:** Kommunikations-ID in Zusammenhang mit der Modbus-Server-TCP-Verbindung (siehe NetPro-Eigenschaften)
- **LADDR:** Adresse der Bausteine in Zusammenhang mit der Modbus-Server-TCP-Verbindung (siehe NetPro-Eigenschaften)

Abb. 2-6: Eigenschaften – TCP-Verbindung: Eingabe-Parameter



- **TIME_OUT_VALUE:** Sie können einen Time-out einstellen, dessen Wert größer 0 ist (Beispiel: 50 ms = S5#T#50MS). Damit lässt sich feststellen, ob die Zeit zwischen dem Senden eines Modbus-Requests und dem Empfangen der Antwort oder Exception durch ein Partnergerät innerhalb des definierten Time-out liegt. Außerdem gibt der Time-out Auskunft darüber, ob der Netzwerkverkehr zu groß ist oder ob es einen Verbindungsabbruch gibt.

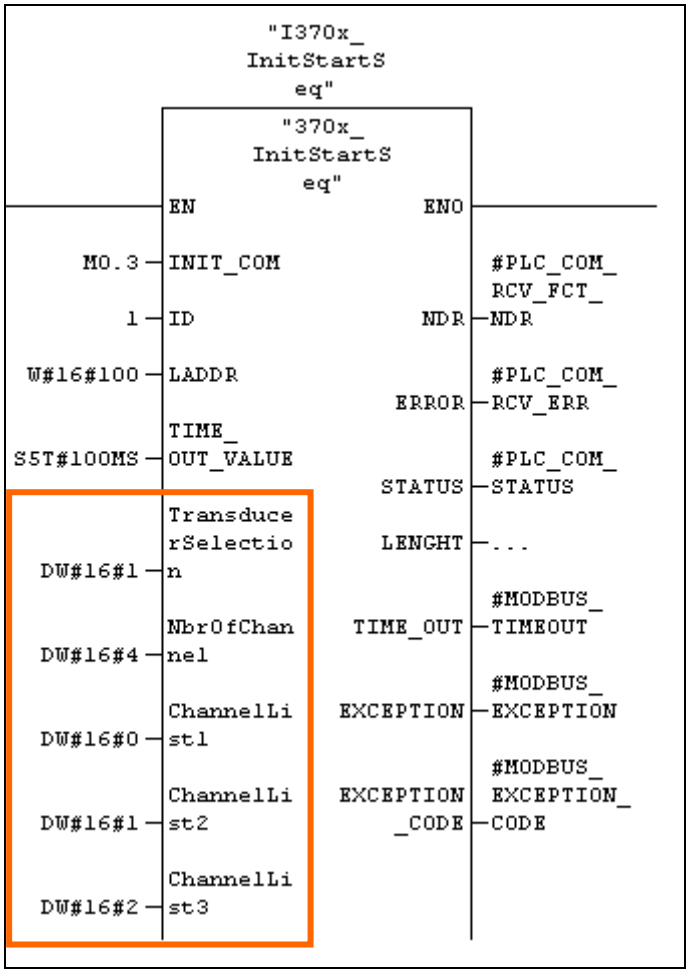
Ausgabe-Parameter:

- **NDR:** Wenn der Funktionsaufruf (Senden eines Modbus-Requests und Empfangen einer Antwort oder Exception) abgeschlossen ist, steht **N**ew **D**ata **R**eceive auf TRUE; ansonsten auf FALSE.
- **ERROR:** Rückmeldung, wenn bei der Funktion „AG_RECV“ ein Fehler auftritt
- **STATUS:** Dieser Parameter gibt den Fehlerstatus der Funktion „AG_RECV“ zurück (siehe SIMATIC Manager-Hilfe: „AG_RECV“).
- **LENGTH:** Dieser Parameter gibt die Länge in Byte der empfangenen Modbus-Telegramme an (nur „Nutzdaten“ aus dem Antwort-Telegramm ohne Header).
- **TIME_OUT:** Nach einem Time-out wird TRUE angezeigt (siehe **TIME_OUT_VALUE**).
- **EXCEPTION:** Bei einem Modbus-Fehler wird TRUE angezeigt.
- **EXCEPTION_CODE:** Dieser Code kann gelesen werden, um die Fehlernummer zu erhalten.
  Die Fehlernummern werden in der Dokumentation „msxeXXXX_modbus.html“ (siehe Kap. 1.3) im Kapitel „Exception code description“ erläutert.
  Wenn der **EXCEPTION_CODE** gleich 9 ist, dann ist kein Modbus-Fehler aufgetreten, sondern ein Fehler auf dem MSX-E-System. Details zu diesem Fehler können Sie sich mit Hilfe der MSX-E-Common-Funktion „GetLastCommandStatus“ anzeigen lassen.

2.3.2 Allgemeine Parameter der MSX-E-Modbus-TCP-Funktionen

Alle Parameter-Namen, die Kleinbuchstaben enthalten, sind allgemeine Parameter der MSX-E-Modbus-TCP-Funktionen.

Abb. 2-7: Allgemeine Parameter des Funktionsbausteins „370x_InitStartSeq“



Die Beschreibung der allgemeinen Parameter finden Sie in der Dokumentation „msxeXXXX_modbus.html“ (siehe Kap. 1.3) im Kapitel „Siemens STEP 7 compatibility information (AWL/SDF code)“.

Hinweise zu den Funktionsnamen erhalten Sie in Kap. 2.2 dieses Handbuchs.

Beispiel:

Der SPS-Funktionsbaustein „370x_InitStartSeq“ entspricht der MSX-E-Modbus-TCP-Funktion „MX370x__TransducerInitAndStartSequenceEx“.

Die Eingabe-Parameter sind durch das Präfix „[Query frame layout]“ gekennzeichnet.

Abb. 2-8: 370x_InitStartSeq: Beschreibung der Eingabe-Parameter

3.16 Function MX370x__TransducerInitAndStartSequenceEx

Description

Initialise and start the transducer sequence acquisition mode

Parameters:

[Query frame layout] **TransducerSelection :** Transducer type selection

[Query frame layout] **NbrOfChannel :** Number of channel in the sequence

[Query frame layout] **ChannelList :** List of the channel index (0 to MaxChannel-1) who compose the sequence

[Query frame layout] **DivisionFactor :** Division factor (min: 5, max: 255)

[Query frame layout] **NbrOfSequence :** Number of sequence to acquire :

- 0 : continuous mode
- > 0 : number of sequence

[Query frame layout] **NbrMaxSequenceToTransfer :** This parameter defined the minimal number of sequences to acquire between each send of data by the modul.

Warning : They are two possibilities that the number of sequences sent doesn't reach the minimal number:
- By the end of the acquisition.
- If the memory capicity is not big enough.

[Query frame layout] **DelayMode :** Delay Mode :

- ADDIDATA_DELAY_NOT_USED 0 : Delay is not used.
- ADDIDATA_DELAY_MODE1_USED 1 : The delay time defines the time between 2 sequence beginnings.
- ADDIDATA_DELAY_MODE2_USED 2 : The delay time defines the time between the end of a sequence until the beginning of the next sequence.

[Query frame layout] **DelayTimeUnit :** Selection of the delay time unit

- 0: ms
- 1: s

Die Ausgabe-Parameter enthalten das Präfix „[Response frame layout]“.

Abb. 2-9: 370x_InitStartSeq: Beschreibung der Ausgabe-Parameter

2.10 Function MX370x__getNumberOfChannelsEx

Description

Return the number of transducer channels on the module (4,8 or 16)

Parameters:

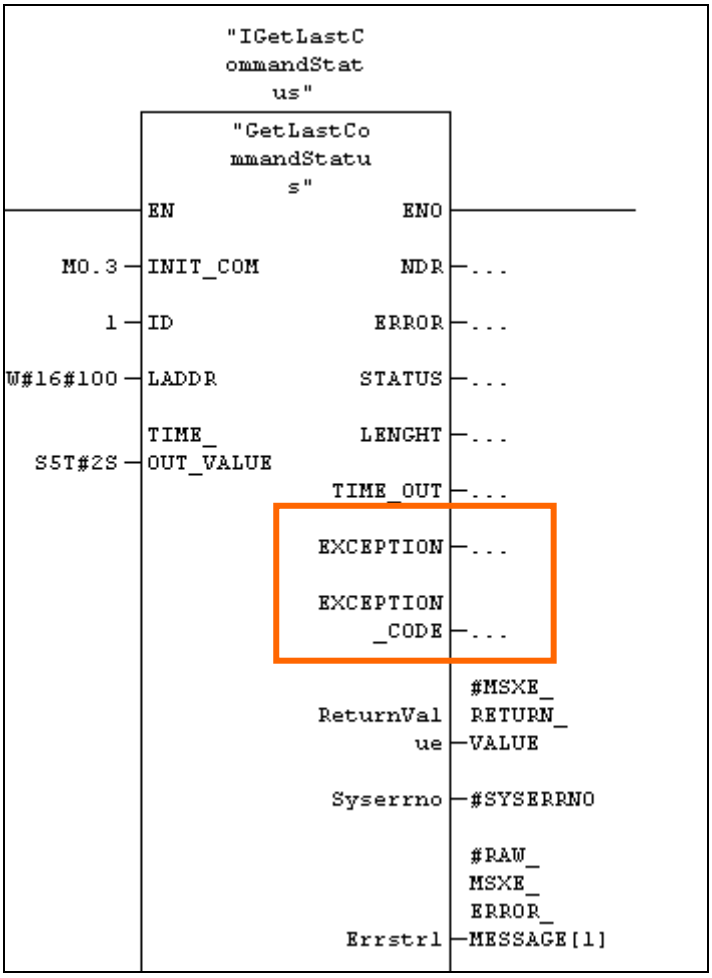
[Response frame layout]***ChannelNumber:*** Number of channels

2.3.3 MSX-E-System-Fehler

Um festzustellen, ob eine Funktion nach ihrem Aufruf einen Fehler erzeugt hat, können Sie die Ausgabe-Parameter **EXCEPTION** und **EXCEPTION_CODE** überprüfen:

- **EXCEPTION:** Bei einem Modbus-Fehler wird TRUE angezeigt.
- **EXCEPTION_CODE:** Dieser Code kann gelesen werden, um die Fehlernummer zu erhalten.
  Die Fehlernummern werden in der Dokumentation „msxeXXXX_modbus.html“ (siehe Kap. 1.3) im Kapitel „Exception code description“ erläutert.
  Wenn der **EXCEPTION_CODE** gleich 9 ist, dann ist kein Modbus-Fehler aufgetreten, sondern ein Fehler auf dem MSX-E-System. Details zu diesem Fehler können Sie sich mit Hilfe der MSX-E-Common-Funktion „GetLastCommandStatus“ anzeigen lassen.

Abb. 2-10: GetLastCommandStatus: Funktionsbaustein



- **ReturnValue:** Dieser Parameter liefert einen Rohwert, der immer gleich oder größer 0 ist.
  Wenn der Funktionsparameter **TransducerSelection** im Funktionsbaustein **InitStartSeq** falsch angegeben wird, ist **ReturnValue** gleich 0xfffffffe. Der Rückgabewert wird wie folgt berechnet:

ReturnValue - 0x100000000 = Rückgabewert

Beispiel:
0xfffffffe - 0x100000000 = -2

Die Rückgabewerte können in der Dokumentation „msxeXXXX_modbus.html“ (siehe Kap. 1.3) in der Beschreibung der Funktion „MX370x__TransducerInitAndStartSequenceEx“ überprüft werden.

Abb. 2-11: InitStartSeq: Rückgabewerte

Returns:

Possible return value on the remote system (read them with GetLastCommandStatus)

- 0 : success
- -1: means an system error occured
- -2: Tranducer selection error
- -3: Number of channel error
- -4: Channel array selection error
- -5: Division factor error
- -6: Incorrect value for Hardware Trigger Mode
- -7: Incorrect value for Hardware Trigger Front
- -8: Incorrect value for Synchro Trigger Mode
- -9: Incorrect value for Hardware Trigger Count
- -10: Incorrect value for Hardware Trigger filter time
- -11: Incorrect value for "trigger number of sequences to acquire"
- -12: Delay Mode selection error
- -13: Delay time unit selection error
- -14: Delay value
- -15: Wrong data format parameter (ulOption1)
- -16: A value for Hardware Trigger front was defined but Hardware Trigger Mode is not set
- -17: Cannot use both triggers at the same time
- -18: Incorrect value for the hardware trigger stop front
- -19: Hardware trigger stop can not be used by this configuration of hardware trigger start
- -100: TransducerInit kernel function error
- -101: InitConvertTimeDivisionFactor kernel function error
- -102: InitEnableDisableSequenceDelay kernel function error
- -103: InitDigitalInputFilter kernel function error
- -104: InitEnableDisableHardwareTrigger kernel function error
- -105: InitEnableSynchroTrigger kernel function error
- -106: DisableSynchroTrigger kernel function error
- -107: SetTriggerSequenceCount kernel function error
- -108: InitSequence kernel function error
- -109: StartStopSequence kernel function error

- **Syserrno:** Dieser Parameter funktioniert wie **ReturnValue**. Wenn ein Syserrno-Fehler auftritt, können Sie die Fehlermeldung mit Hilfe des Ausgabe-Parameters **ErrstrX** lesen.
- **ErrstrX:** Jeder ErrstrX-Parameter entspricht einem ASCII-Wert bzw. einem Buchstaben der Fehlermeldung. Wenn der ASCII-Code der Fehlermeldung übersetzt wird, erhalten Sie die Fehlermeldung in Buchstaben.

3 ADModbus-Bausteine

3.1.1 Verwendung

Die ADModbus-Bausteine stehen im Programmierbeispiel als SPS-Bausteine zur Verfügung. Mit diesen wird das Modbus-TCP-Protokoll zwischen der SPS und den MSX-E-Systemen automatisch verwaltet.

Abb. 3-1: ADModbus-Bausteine

FC14	ADModbus_WTo2B_FC	KOP
OB100	ADModbus_StartUp_OB	KOP
FB100	ADModbus_Main_FB	KOP
DB100	ADModbus_Main_DB	DB
FC16	ADModbus_IntelToMoto_FC	KOP
FC17	ADModbus_DWToR_FC	KOP
FC12	ADModbus_DWToB_FC	KOP
OB1	ADModbus_CycleExec_OB	KOP
DB101	ADModbus_AGRECV_IDB	DB
FB101	ADModbus_AGRECV_FB	KOP
FC15	ADModbus_4BToR_FC	KOP
FC11	ADModbus_4BToDW_FC	KOP
FC13	ADModbus_2BToW_FC	KOP

Wenn Sie die MSX-E-Modbus-TCP-Funktionen verwenden, werden im Hintergrund die ADModbus-Bausteine verarbeitet.

3.1.2 OB1 und OB100

Je nach Projekt müssen Sie die Organisationsbausteine OB1 und OB100 anpassen.

HINWEIS!
Die ADModbus-Bibliothek verwendet die Merker M0.1, M0.2, M0.3. Falls diese schon von einer anderen Anwendung genutzt werden, müssen Sie darauf achten, dass die Ersatz-Merker wie im OB1 bzw. OB100 des Programmierbeispiels definiert sind.

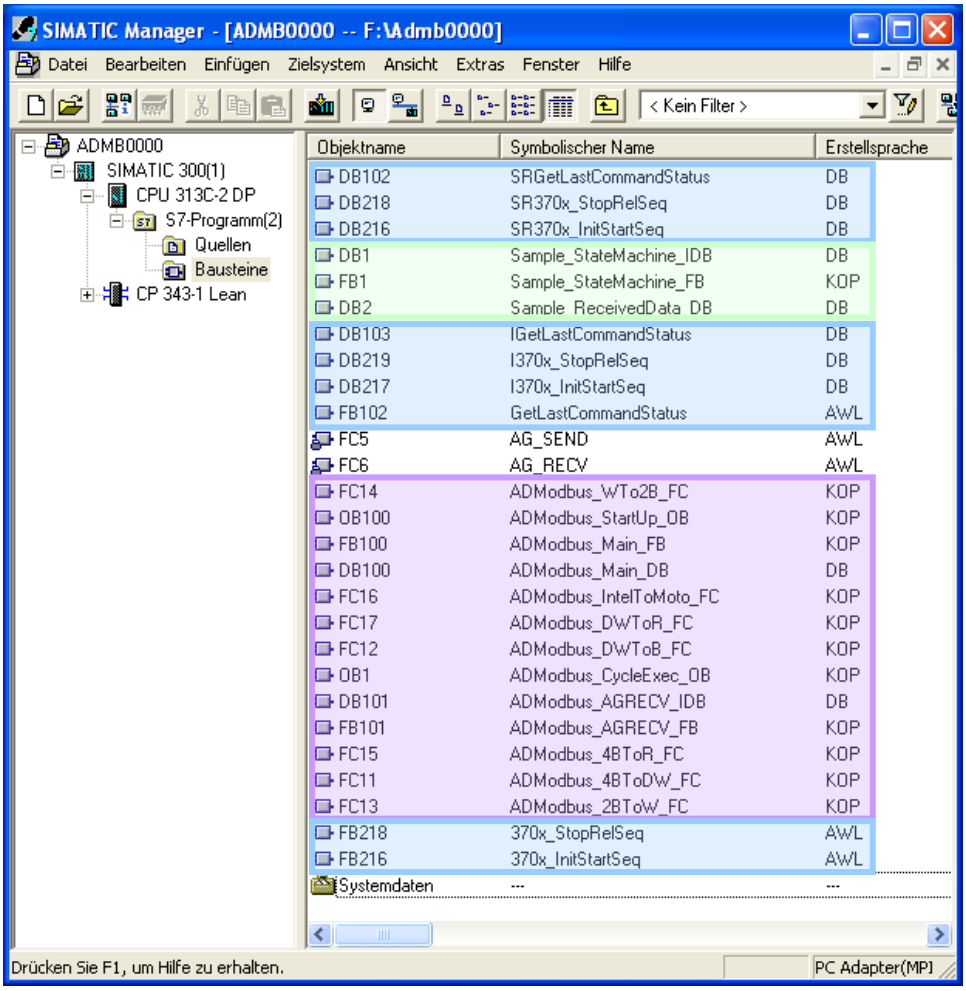
4 Einbindung der ADModbus-Bibliothek

4.1 Funktionen und Bausteine

Ein Teil der ADModbus-Bibliothek wird im Programmierbeispiel mitgeliefert. Dieses können Sie als Projektbasis nutzen oder Sie können die benötigten MSX-E-Modbus-TCP-Funktionen in Ihr eigenes Projekt importieren bzw. kopieren.

Die folgende Abbildung bezieht sich auf eine Sequenzerfassung des Ethernet-Systems **MSX-E370x**.

Abb. 4-1: ADModbus-Bibliothek: Funktionen und Bausteine



- Alle Bausteine mit dem Präfix „ADModbus“ (siehe violette Markierung) gehören zu den ADModbus-Bausteinen und werden als Siemens-Bausteine im Programmierbeispiel mitgeliefert.
- Die MSX-E-Modbus-TCP-Funktionen werden als AWL-Quellcode-Dateien mitgeliefert. Die blau markierten Bausteine werden im Programmierbeispiel schon als Bausteine mitgeliefert. Die restlichen MSX-E-Modbus-TCP-Funktionen können Sie in den SIMATIC Manager importieren und als Bausteine kompilieren lassen (siehe Kap. 6).
- Die grün markierten Siemens-Bausteine gehören zum Programmierbeispiel des MSX-E-Systems.

4.2 Voraussetzungen

Bitte stellen Sie sicher, dass folgende Voraussetzungen erfüllt sind:

- Siemens **CPU313C-2DP** (oder kompatible CPU)
- Siemens **CP343-1 Lean** (oder kompatibles Ethernet-System für die SPS)
- FC5 (AG_SEND) für **S7-300** (Funktion zur asynchronen Kommunikation)
- FC6 (AG_RECV) für **S7-300** (Funktion zur asynchronen Kommunikation)
- S7-Beispiel für **STEP 7**.

4.3 Einbindung in den SIMATIC Manager

Wie bereits in Kap 4.1 erwähnt, können Sie das Programmierbeispiel als Projektbasis nutzen oder die benötigte ADModbus-Bausteine aus dem Programmierbeispiel in Ihr Projekt kopieren.
Um die ADModbus-Bibliothek in den SIMATIC Manager einzubinden, sind folgende Funktionen und Bausteine erforderlich:

- S7-Funktionen: „AG_SEND“ und „AG_RECV“ (siehe SIMATIC Manager-Hilfe)
- alle ADModbus-Bausteine
- MSX-E-Modbus-TCP-Funktionen (je nach Bedarf)
- Zustandsmaschine, um die Reihenfolge der Funktionsaufrufe zu definieren.

HINWEIS!
Die ADModbus-Bibliothek verwendet die Merker M0.1, M0.2, M0.3. Falls diese schon von einer anderen Anwendung genutzt werden, müssen Sie darauf achten, dass die Ersatz-Merker wie im OB1 bzw. OB100 des Programmierbeispiels definiert sind.

5 SPS-Konfiguration

Die nachfolgende Konfiguration basiert auf einer SPS **S7-300** mit **CP343-1**. Eventuell werden bei Ihrer SPS andere Menüs angezeigt, jedoch sind die Einstellungen dieselben.

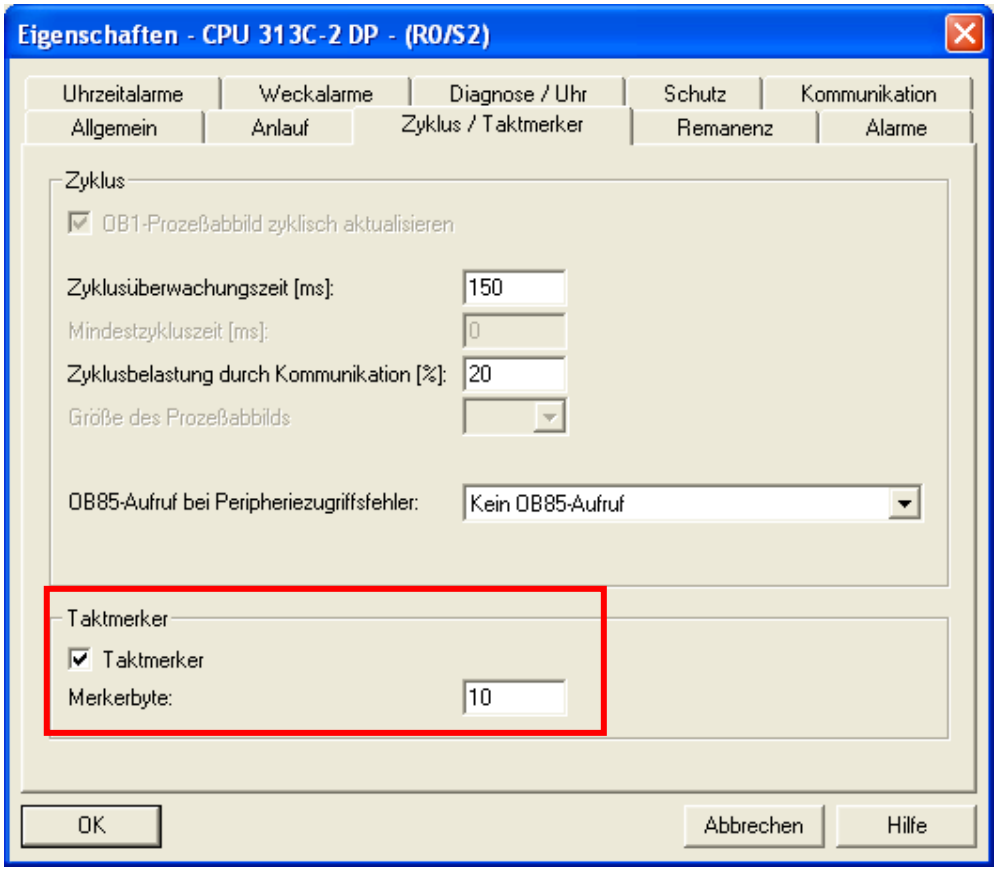
5.1 Taktmerker

Um die ADModbus-Bibliothek nutzen zu können, muss zuerst der Taktmerker konfiguriert werden.

- Klicken Sie auf „Station SIMATIC“.
- Doppelklicken Sie auf „Hardware“.
- Klicken Sie in der Spalte „Baugruppe“ mit der rechten Maustaste auf „CPU“ und wählen Sie den Menüpunkt „Objekteigenschaften“ aus.

Folgendes Fenster wird angezeigt:

Abb. 5-1: SIMATIC Manager: S7-Taktmerker-Konfiguration



- Geben Sie bei „Merkerbyte“ den Wert „10“ ein.

Wenn Sie bereits einen Taktmerker verwenden, können Sie folgende ADModbus-Bausteine anpassen:

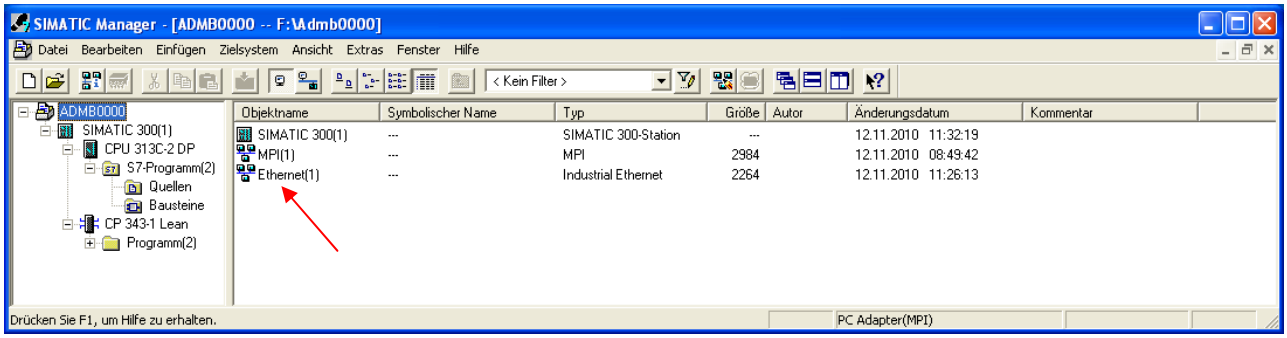
- ADModbus_Main_FB
- ADModbus_StartUp_OB.

5.2 TCP-Verbindung

5.2.1 Modbus-Server

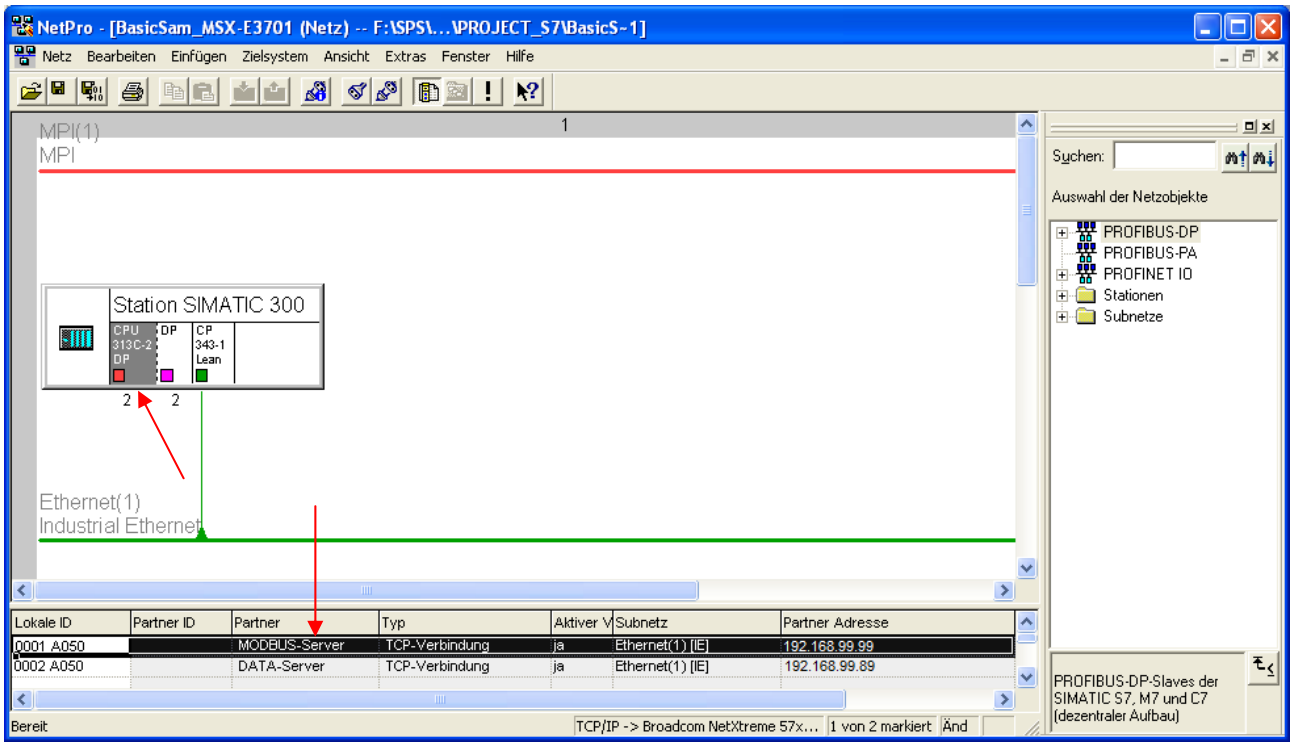
Standardmäßig benötigt die ADModbus-Bibliothek eine TCP-Verbindung, um den Modbus-Server des MSX-E-Systems ansprechen zu können.

Abb. 5-2: Ethernet-Konfiguration: TCP-Verbindung



- Öffnen Sie den SIMATIC Manager und anschließend das Projekt „ADMB0000“.
- Zur Konfiguration der TCP-Verbindung doppelklicken Sie auf den Objektnamen „Ethernet“ (siehe Pfeil).

Abb. 5-3: Ethernet-Konfiguration: Modbus-Server

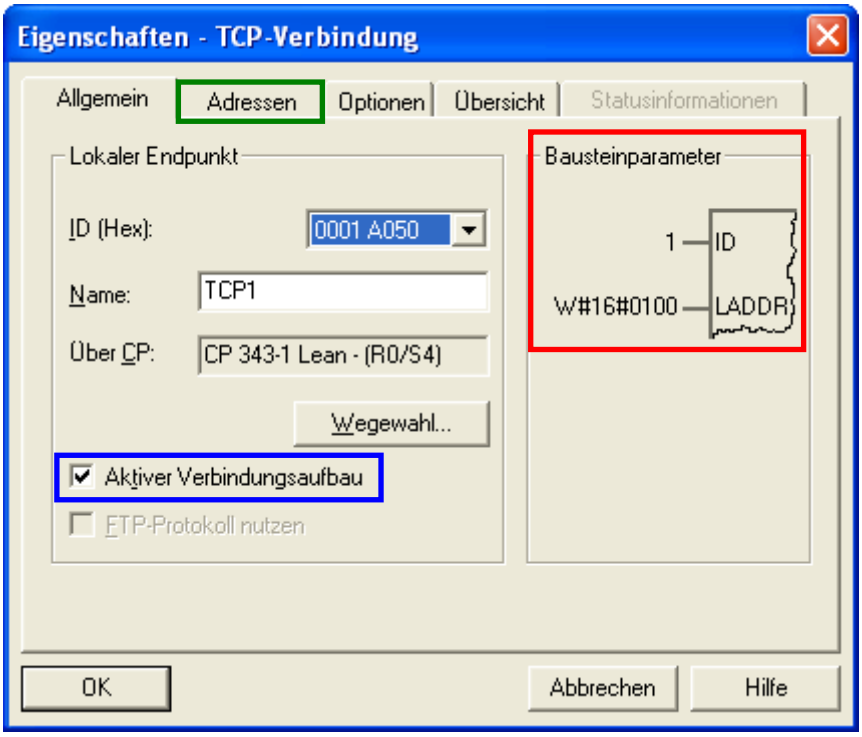


- Klicken Sie auf „CPU313C-2DP“ (siehe Pfeil oben).
- Doppelklicken Sie in der markierten Zeile der Tabelle unten auf „MODBUS-Server“ (siehe Pfeil unten).

In der Tabellenspalte „Typ“ steht, dass die Ethernet-Verbindung eine TCP-Verbindung ist.

Danach wird folgendes Fenster angezeigt:

Abb. 5-4: Eigenschaften – TCP-Verbindung: Allgemein

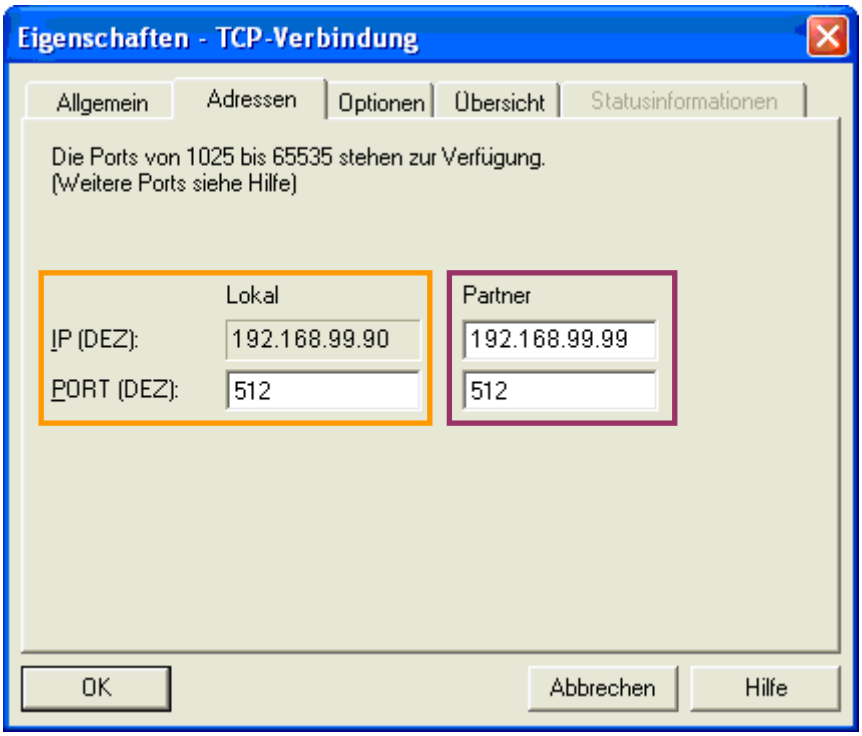


Damit die S7 die Verbindung zum MSX-E-System herstellt, muss **Aktiver Verbindungsaufbau** aktiviert sein.

Rechts im Fenster sehen Sie die **Bausteinparameter**, die zusammen mit dem Funktionsbaustein „ADModbus_Main_FB“ verwendet werden müssen. Die Verbindungs-ID (Socket-ID) und die Adresse sind angegeben.

- Klicken Sie auf den Registerreiter **Adressen**.

Abb. 5-5: Eigenschaften – TCP-Verbindung: Adressen



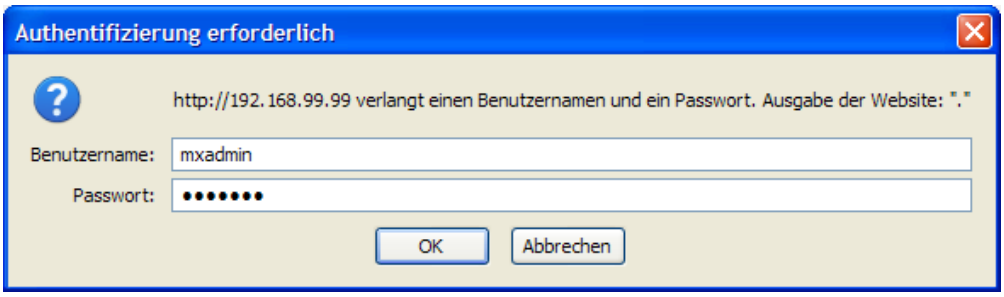
- Geben Sie im Feld „Port“ im Bereich **Lokal** die Portnummer der S7 ein. Diese muss der Portnummer des MSX-E-Systems entsprechen und kann nur einmal verwendet werden.
  Um mehrere MSX-E-Systeme zu nutzen, muss jeweils eine weitere lokale Portnummer angegeben werden.

- Geben Sie im Feld „IP“ im Bereich **Partner** die IP-Adresse des MSX-E-Systems ein sowie im Feld „Port“ die Portnummer des MSX-E-Systems.
  Die Partner-Portnummer ist standardmäßig immer auf 512 eingestellt, weil dieser Port für die Modbus-Kommunikation reserviert ist. Diese Portnummer muss mit der Portnummer übereinstimmen, die auf der Weboberfläche des MSX-E-Systems festgelegt wurde.

- Um die Portnummer des MSX-E-Systems zu überprüfen, öffnen Sie einen Webbrowser (z.B. Mozilla Firefox, Internet Explorer etc.) und geben Sie folgende Adresse ein: „http://[IP-Adresse des MSX-E-Systems]“.

Ein Login-Fenster wird angezeigt:

Abb. 5-6: Login-Fenster



- Geben Sie als Benutzername und Passwort jeweils „mxadmin“ ein.
- Klicken Sie auf der Weboberfläche des MSX-E-Systems auf den Menüpunkt „Modbus server“.

Abb. 5-7: Modbus server: Configuration

- Wählen Sie im Abschnitt „Configuration“ bei „Endianness“ die Option „Big“ aus sowie bei „Protocol“ die Option „TCP/IP“.
- Passen Sie die Portnummer im Feld „Port number“ an, falls Sie die Standard-Portnummer (512) nicht verwenden können.

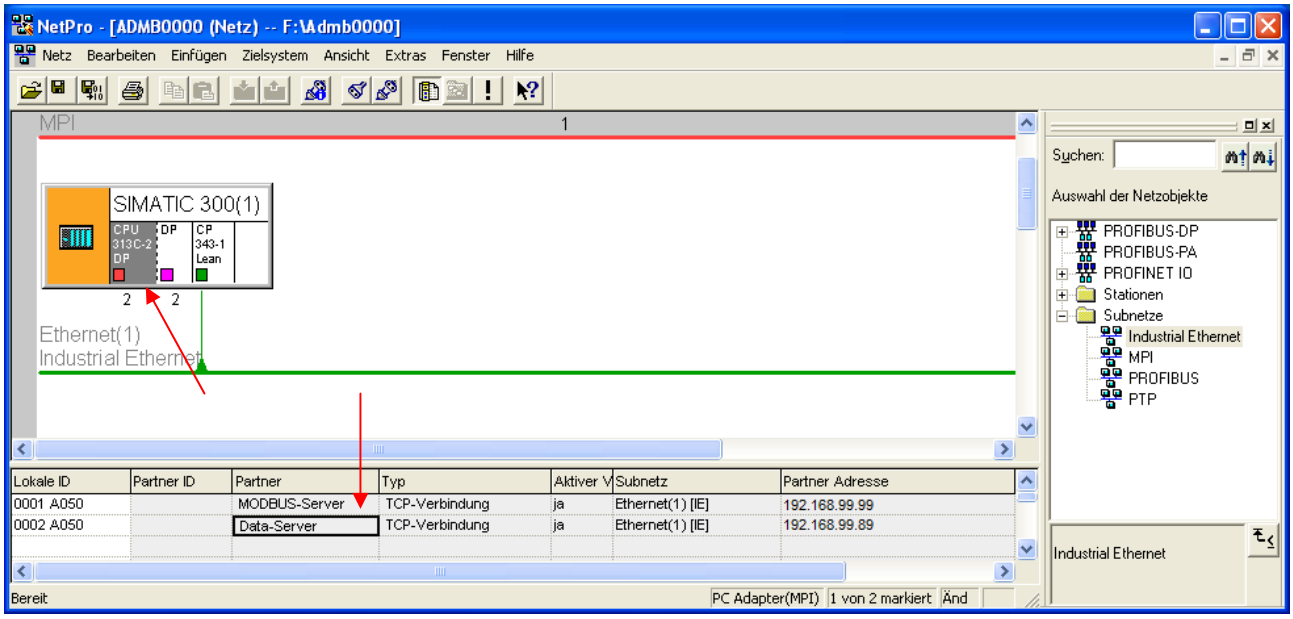
Abb. 5-8: Modbus server: Set and save

- Wenn Sie die Portnummer etc. geändert haben, klicken Sie auf „Set and Save“ und danach auf „Reload“.

5.2.2 Datenserver

Wenn Sie die Messwerte aus dem Datenserver des MSX-E-Systems lesen möchten, müssen Sie eine weitere TCP-Verbindung einrichten, da der Modbus-Server nur für Kommandos verwendet wird.

Abb. 5-9: Ethernet-Konfiguration: Datenserver

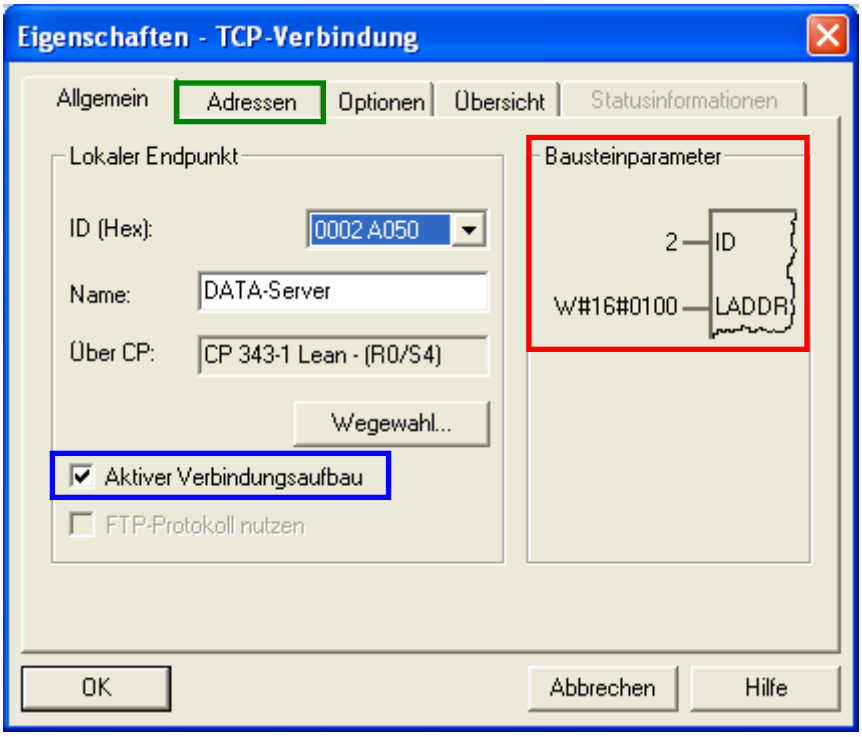


- Klicken Sie auf „CPU313C-2DP“ (siehe Pfeil oben).
- Doppelklicken Sie in der markierten Zeile der Tabelle unten auf „Data-Server“ (siehe Pfeil unten).

In der Tabellenspalte „Typ“ steht, dass die Ethernet-Verbindung eine TCP-Verbindung ist.

Danach wird folgendes Fenster angezeigt:

Abb. 5-10: Eigenschaften – TCP-Verbindung: Allgemein

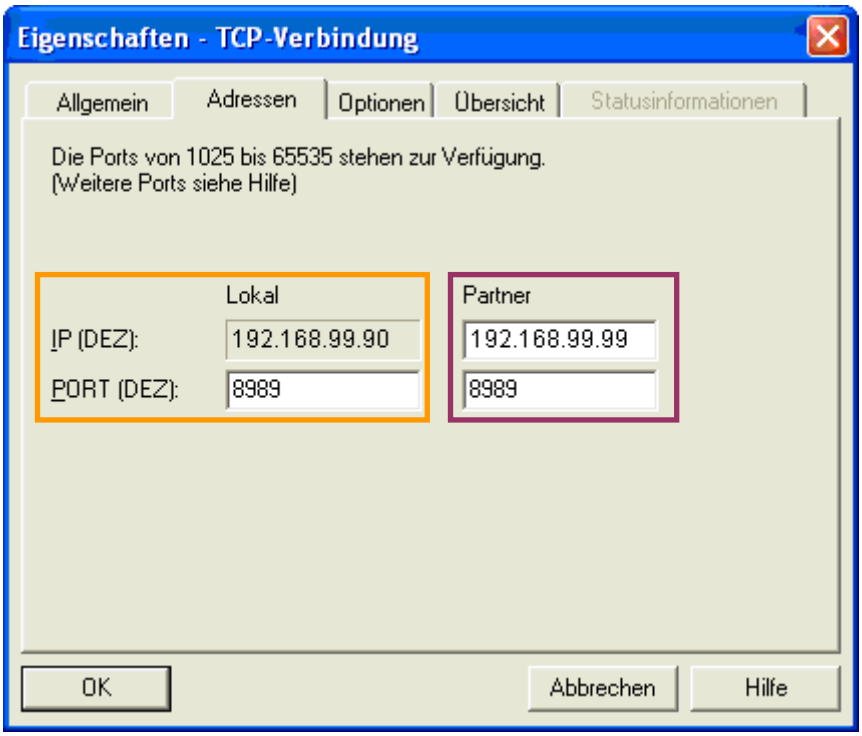


Damit die S7 die Verbindung zum MSX-E-System herstellt, muss **Aktiver Verbindungsaufbau** aktiviert sein.

Rechts im Fenster sehen Sie die **Bausteinparameter**, die zusammen mit der Funktion „FC6 (AG_RECV)“ verwendet werden müssen. Die Verbindungs-ID (Socket-ID) und die Adresse sind angegeben.

- Klicken Sie auf den Registerreiter **Adressen**.

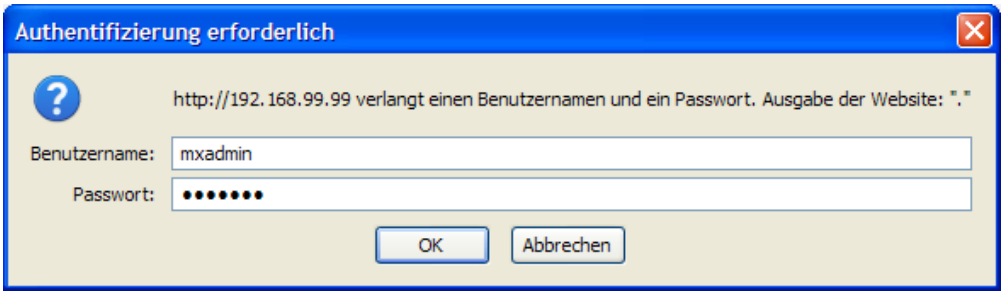
Abb. 5-11: Eigenschaften – TCP-Verbindung: Adressen



- Geben Sie im Feld „Port“ im Bereich **Lokal** die Portnummer der S7 ein. Diese muss der Portnummer des MSX-E-Systems entsprechen.
- Geben Sie im Feld „IP“ im Bereich **Partner** die IP-Adresse des MSX-E-Systems ein sowie im Feld „Port“ die Portnummer des MSX-E-Systems. Diese muss mit der Portnummer übereinstimmen, die auf der Weboberfläche des MSX-E-Systems festgelegt wurde.
- Um die Portnummer des MSX-E-Systems zu überprüfen, öffnen Sie einen Webbrowser (z.B. Mozilla Firefox, Internet Explorer etc.) und geben Sie folgende Adresse ein: „http://[IP-Adresse des MSX-E-Systems]“.

Ein Login-Fenster wird angezeigt:

Abb. 5-12: Login-Fenster



- Geben Sie als Benutzername und Passwort jeweils „mxadmin“ ein.

- Klicken Sie auf der Weboberfläche des MSX-E-Systems auf den Menüpunkt „Data server“.

Abb. 5-13: Data server: Configuration

Protocol	TCP
TCP port number	8989
SO_SNDBUF (in bytes)	104448
TCP/IP network clients filter	
UDP/IP network targets	

- Passen Sie im Abschnitt „Configuration“ im Feld „TCP port number“ die Portnummer an, falls Sie die Standard-Portnummer (8989) nicht verwenden können.

Abb. 5-14: Data server: Set and save

- Wenn Sie die Portnummer geändert haben, klicken Sie auf „Set and Save“ und danach auf „Reload“.
- Klicken Sie auf den Registerreiter „Blocking transfer“.

Abb. 5-15: Data server: Blocking transfer

Activate blocking TCP/IP transfer	No
TCP/IP transfer timeout	1,0

- Bitte achten Sie darauf, dass im Abschnitt „Configuration“ bei „Activate blocking TCP/IP transfer“ die Option „No“ ausgewählt ist.
- Wenn Sie diese Änderung vorgenommen haben, klicken Sie auf „Set and Save“ und danach auf „Reload“.

6 Funktionsbausteine

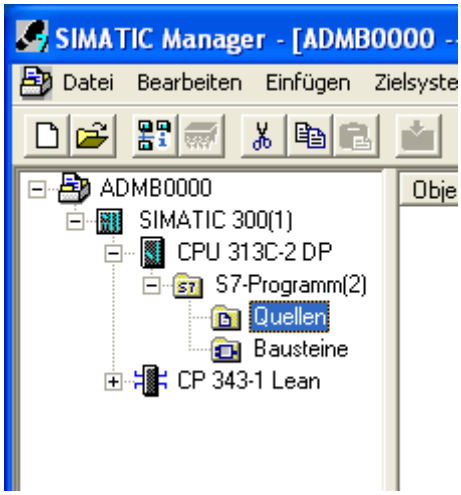
6.1 Umsetzung der AWL-Quellcode-Dateien in Funktionsbausteine

Die MSX-E-Modbus-TCP-Funktionen, die im Programmierbeispiel nicht zur Verfügung stehen, können über die AWL-Quellcode-Dateien in den SIMATIC Manager importiert werden.
Jede MSX-E-Modbus-TCP-Funktion, die auf dem MSX-E-System implementiert wird, hat danach einen Funktionsbaustein, der zusammen mit einem Instanz-Datenbaustein und einem Global-Datenbaustein auf die Funktion zugreift.
Da es für jede MSX-E-Modbus-TCP-Funktion jeweils einen AWL-Quellcode gibt, sind in diesem somit drei Bausteine implementiert. Die AWL-Quellcodes können im SIMATIC Manager geöffnet und kompiliert werden.

- Klicken Sie im SIMATIC-Manger mit der rechten Maustaste auf „Quellen“.

- Wählen Sie den Menüpunkt „Neues Objekt einfügen“ aus und klicken Sie danach auf „Externe Quelle“.

Abb. 6-1: SIMATIC Manager: Quellenverzeichnis

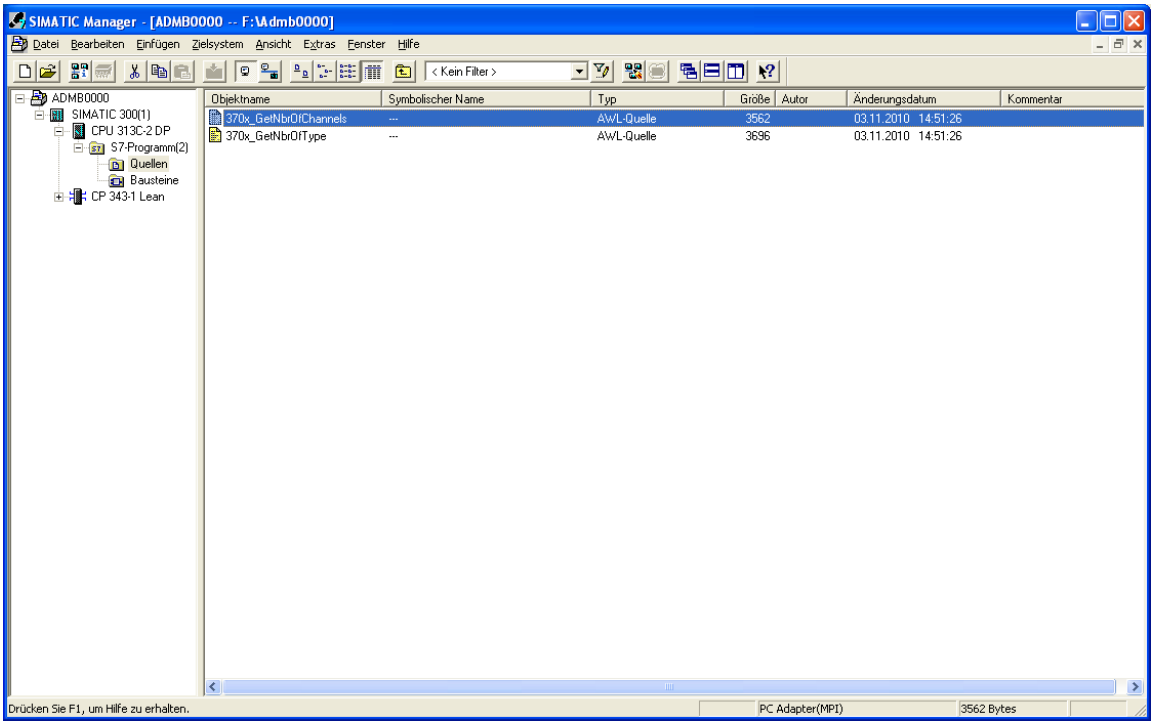


- Wählen Sie den Ordner aus, in dem sich die AWL-Quellcode-Dateien mit den Quellcodes der MSX-E-Modbus-TCP-Funktionen befinden. Markieren Sie die zu importierenden Dateien und klicken Sie auf die Schaltfläche „Öffnen“.

Abb. 6-2: SIMATIC Manager: Auswahl der AWL-Quellcode-Dateien

Die ausgewählten AWL-Quellcode-Dateien befinden sich nun im Ordner „Quellen“.

Abb. 6-3: SIMATIC Manager: AWL-Quellcode-Dateien



- Um die einzelnen MSX-E-Modbus-TCP-Funktionen in den SIMATIC Manager zu importieren, doppelklicken Sie auf die jeweilige AWL-Quellcode-Datei. Dadurch haben Sie Zugriff auf den AWL-Quellcode.

HINWEIS!

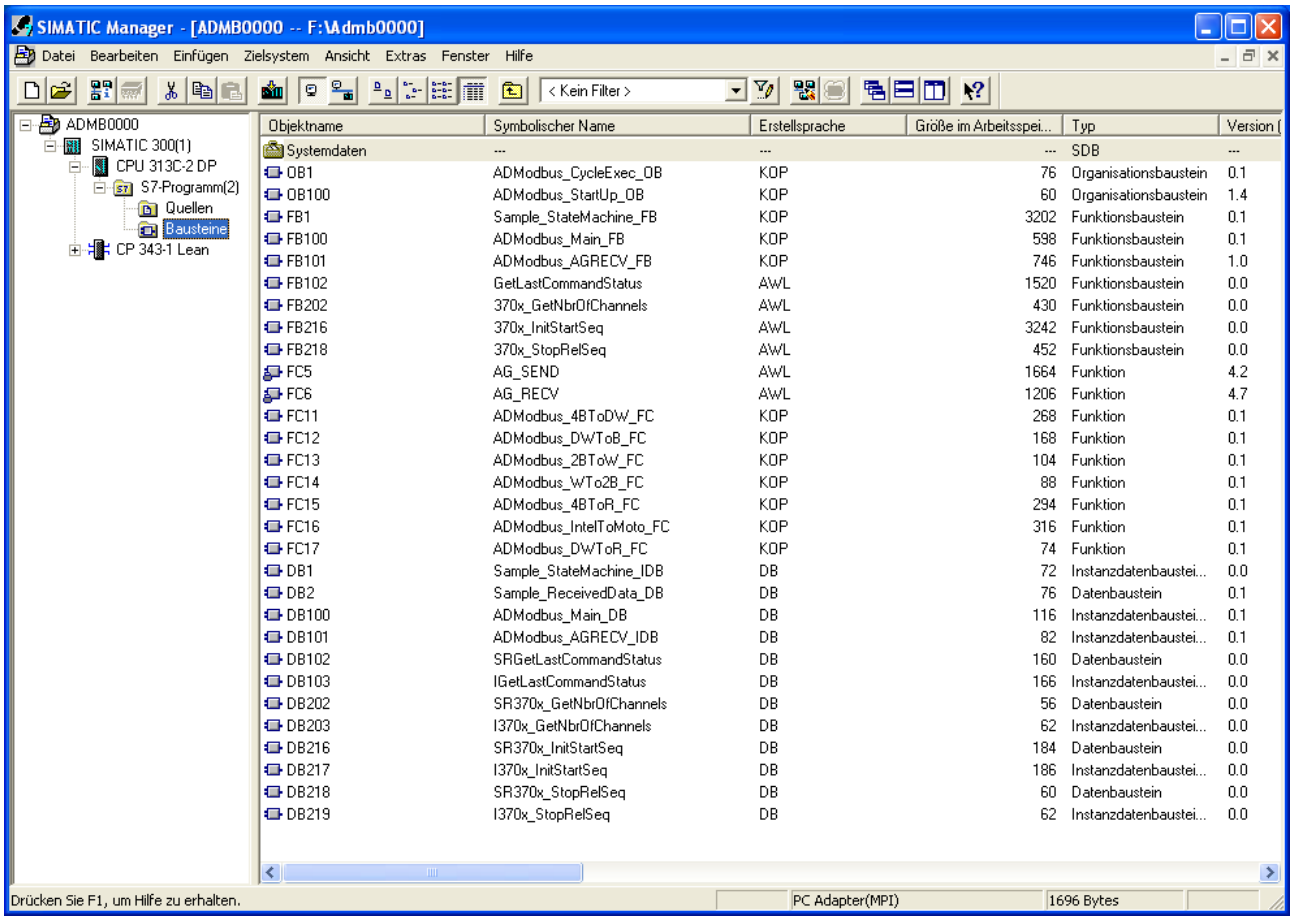
Die Nummerierung der MSX-E-Modbus-TCP-Funktionen ist bei allen MSX-E-Systemen gleich, auch wenn die Funktionen unterschiedlich sind.
Wenn die AWL-Quellcode-Datei geöffnet ist, können Sie bei allen drei Bausteinen jeweils die Nummer ändern. Bitte beachten Sie, dass eine Baustein-Nummer nur einmal vergeben werden darf, da ansonsten der Baustein mit derselben Nummer bei der AWL-Kompilierung überschrieben wird!
Jede MSX-E-Modbus-TCP-Funktion ist wie folgt indexiert (Beispiel):

- **Funktionsbaustein: FB202 370x_GetNbrOfChannels**
- **Global-Datenbaustein: DB202 SR370x_GetNbrOfChannels (Anfrage/Antwort-Modbus-TCP-Telegramme)**
- **Instanz-Datenbaustein: DB203 I370x_GetNbrOfChannels.**

- Wenn Sie die Quellcode-Änderungen in einer Datei abgeschlossen haben, klicken Sie in der Menüleiste auf „Datei“ und wählen den Menüpunkt „Übersetzen“ aus.

Die kompilierten Dateien befinden sich nun als Bausteine im gleichnamigen Ordner („Bausteine“).

Abb. 6-4: SIMATIC Manager: Bausteine



6.2 Mnemonische Symbole

Pro MSX-E-System gibt es eine Symbol-Datei (.sdf), in der alle verfügbaren Symbole bzw. MSX-E-Modbus-TCP-Funktionen mit jeweils drei Bausteinen aufgelistet sind.
Die Symbol-Datei kann u.a. genutzt werden, um Bausteine zu benennen. Das heißt, dass anstelle der Baustein-Nummer aus dem Quellcode der symbolische Name verwendet wird. Dazu müssen Sie die mitgelieferte Datei „mnemonics.sdf“ über den Symbol-Editor in das Projekt importieren.

- Öffnen Sie den Symbol-Editor und klicken Sie in der Menüleiste auf „Tabelle“.
- Wählen Sie den Menüpunkt „Importieren“ aus und anschließend die Datei „mnemonics.sdf“. Klicken Sie danach auf „Öffnen“.
- Wählen Sie in der angezeigten Symbol-Datei die benötigten Funktionen aus und löschen Sie jene, die nicht verwendet werden sollen.

Abb. 6-5: SIMATIC Manager: Mnemonische Symbole

Symbol Editor - [S7-Programm(2) (Symbole) -- ADMB0000\SIMATIC 300(1)\CPU 313C-2 DP]

Tabelle Bearbeiten Einfügen Ansicht Extras Fenster Hilfe

Alle Symbole

	Status	Symbol	Adresse	Datentyp	Kommentar
1		370x_GetNbrOfChannels	FB 202	FB 202	
2		370x_InitStartSeq	FB 216	FB 216	MX370x__TransducerInitAndStartSequence
3		370x_StopRelSeq	FB 218	FB 218	MX370x__TransducerStopAndReleaseSequence
4		ADModbus_2BToW_FC	FC 13	FC 13	
5		ADModbus_4BToDW_FC	FC 11	FC 11	
6		ADModbus_4BToR_FC	FC 15	FC 15	
7		ADModbus_AGRECV_FB	FB 101	FB 101	
8		ADModbus_AGRECV_IDB	DB 101	FB 101	
9		ADModbus_CycleExec_OB	OB 1	OB 1	
10		ADModbus_DWToB_FC	FC 12	FC 12	
11		ADModbus_DWToR_FC	FC 17	FC 17	
12		ADModbus_IntelToMoto_FC	FC 16	FC 16	
13		ADModbus_Main_DB	DB 100	FB 100	
14		ADModbus_Main_FB	FB 100	FB 100	
15		ADModbus_StartUp_OB	OB 100	OB 100	
16		ADModbus_WTo2B_FC	FC 14	FC 14	
17		AG_RECV	FC 6	FC 6	AG RECEIVE
18		AG_SEND	FC 5	FC 5	AG SEND
19		GetLastCommandStatus	FB 102	FB 102	GetLastCommandStatus
20		I370x_GetNbrOfChannels	DB 203	FB 202	
21		I370x_InitStartSeq	DB 217	FB 216	MX370x__TransducerInitAndStartSequence.Instance
22		I370x_StopRelSeq	DB 219	FB 218	MX370x__TransducerStopAndReleaseSequence.Instance
23		IGetLastCommandStatus	DB 103	FB 102	GetLastCommandStatusInstance
24		Instanz_DB_FB3	DB 3	DB 3	
25		Sample_ReceivedData_DB	DB 2	DB 2	
26		Sample_StateMachine_FB	FB 1	FB 1	
27		Sample_StateMachine_IDB	DB 1	FB 1	
28		SR370x_GetNbrOfChannels	DB 202	DB 202	
29		SR370x_InitStartSeq	DB 216	DB 216	MX370x__TransducerInitAndStartSequence.Send/Recevice
30		SR370x_StopRelSeq	DB 218	DB 218	MX370x__TransducerStopAndReleaseSequence.Send/Recevice
31		SRGetLastCommandStatus	DB 102	DB 102	GetLastCommandStatus.Send/Receive
32		STATUS	FB 5	FB 5	FMS STATUS
33					

Drücken Sie F1, um Hilfe zu erhalten. NUM

HINWEIS!

Wenn Sie die Änderungen durchgeführt haben, klicken Sie nur auf „Speichern“.

7 Lesen von Daten aus dem MSX-E-Datenserver

7.1 Datenkonvertierung

Es ist zu beachten, dass die MSX-E-Systeme Daten aus dem Datenserver im Little-Endian-Format (Intel-Format) senden, wohingegen eine S7-SPS mit dem Big-Endian-Format (Motorola-Format) arbeitet.
Somit müssen die Werte der MSX-E-Systeme in der SPS in das Big-Endian-Format (Motorola-Format) konvertiert werden. Diese Datencodierung spielt eine wichtige Rolle, da sonst die Messergebnisse falsch interpretiert werden. Eine entsprechende Konvertierungsfunktion wird von ADDI-DATA zur Verfügung gestellt:

- **Funktion:** FC1
- **Name:** ADModbus_IntelToMoto_FC

7.1.1 Little-Endian-Format (Intel-Format)

Das niedrigstwertige Bit (LSB) wird zuerst übertragen:

16-Bit	0x1234	>>	0x34	0x12		
32-Bit	0x12345678	>>	0x78	0x56	0x34	0x12

7.1.2 Big-Endian-Format (Motorola-Format)

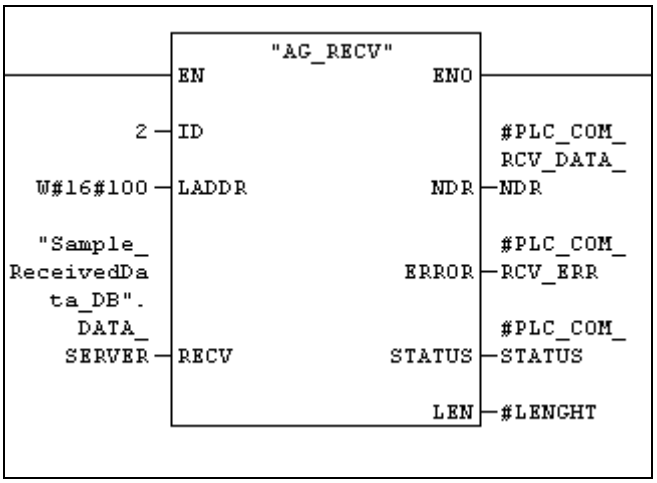
Das höchstwertige Bit (MSB) wird zuerst übertragen:

16-Bit	0x1234	>>	0x12	0x34		
32-Bit	0x12345678	>>	0x12	0x34	0x56	0x78

7.2 Funktion „FC6 (AG_RECV)“

Beschreibung:

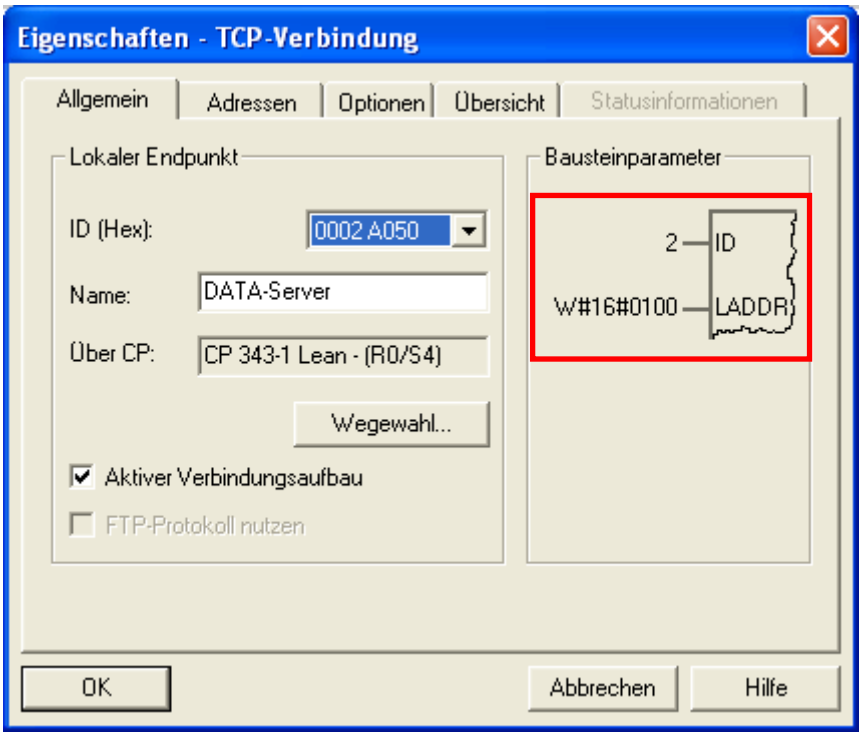
Diese Funktion wurde von Siemens erstellt und ermöglicht es, Daten aus einer Netzwerkverbindung zu lesen.



Eingabe-Parameter:

- **ID:** 2 ist die Nummer des Kommunikationsidentifikators des Datenservers.
- **LADDR:** Der Wert „W#16#0100“ ist die Adresse des Funktionsbausteins.

Abb. 7-1: Eigenschaften – TCP-Verbindung: Eingabe-Parameter



- **RECV:** Zeiger, in dem die Daten gespeichert werden müssen; in diesem Fall in **Sample_ReceivedData_DB.DATA_SERVER**. Dies bedeutet, dass ein Zeiger auf dem Offset „DATA_SERVER“ dem Datenbaustein DB2 („Sample_ReceivedData_DB“) übergeben wird und 20 Byte (5 DWORD) erwartet werden. Die Anzahl der erwarteten Bytes kann sich je nach Bedarf ändern.
- **DATA_SERVER:** Tabelle, die 5 DWORD groß ist. Achten Sie darauf, dass die Tabelle, in der die Werte gespeichert werden, breit genug ist. Die Anzahl der Bytes entspricht immer einem Datenpaket.

Die Größe eines Datenpakets wird auf der Weboberfläche des MSX-E-Systems angegeben:

- Klicken Sie auf der Weboberfläche auf den Menüpunkt „Acquisition“.
- Wählen Sie den Auto-Refresh- oder Sequenz-Modus aus.

Im Abschnitt „Data server frame format“ wird die Größe des Datenpakets angezeigt.

Abb. 7-2: Acquisition: Data server frame format

Size	Field	Description
4 Bytes	Inductive transducer 1	Digital value, encoded on 24 bits.
4 Bytes	Inductive transducer 6	Digital value, encoded on 24 bits.
4 Bytes	Inductive transducer 3	Digital value, encoded on 24 bits.

Wenn der Funktionsbaustein die Werte aus dem MSX-E-System empfangen hat, finden Sie diese in der Tabelle des Zieldatenbausteins in folgender Reihenfolge vor:

DATA_SERVER [0]: enthält den Wert „counter“
DATA_SERVER [1]: enthält den Wert „channel 1“
DATA_SERVER [2]: enthält den Wert „channel 2“
…
DATA_SERVER [4]: enthält den Wert „channel 8“

Das nachfolgende Beispiel basiert auf einem S7-Screenshot („Data server frame format“) und einem Datenbaustein, in dem sich die Tabelle „DATA_SERVER“ als DWORD-Datentyp mit einer Breite von 20 Byte befindet.

Abb. 7-3: Datenbaustein mit der Tabelle „DATA_SERVER“

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	DATA_SERVER	ARRAY[0..4]	DW#16#0	RAW Data
*4.0		DWORD		
+20.0	COUNT_MOTOROLA	DWORD	DW#16#0	Converted sequence counter
+24.0	CHANNEL	ARRAY[0..3]	0.000000e+000	Converted channels values
*4.0		REAL		
=40.0		END_STRUCT		

Abb. 7-4: Tabelle „DATA_SERVER“ im Datenbaustein

Adresse	Name	Typ	Anfangswert	Aktualwert	Kommentar
0.0	DATA_SERVER[0]	DWORD	DW#16#0	DW#16#16000000	RAW Data
4.0	DATA_SERVER[1]	DWORD	DW#16#0	DW#16#00000040	
8.0	DATA_SERVER[2]	DWORD	DW#16#0	DW#16#01C835BA	
12.0	DATA_SERVER[3]	DWORD	DW#16#0	DW#16#015828BA	
16.0	DATA_SERVER[4]	DWORD	DW#16#0	DW#16#01582FBA	
20.0	COUNT_MOTOROLA	DWORD	DW#16#0	DW#16#00000016	Converted sequence counter
24.0	CHANNEL[0]	REAL	0.000000e+000	1.073742e+009	Converted channels values
28.0	CHANNEL[1]	REAL	0.000000e+000	-1.170881e+009	
32.0	CHANNEL[2]	REAL	0.000000e+000	-1.171761e+009	
36.0	CHANNEL[3]	REAL	0.000000e+000	-1.171302e+009	

Ausgabe-Parameter:

- **NDR:** Wenn die Anzahl der Bytes gelesen und NDR gleich 1 ist, dann ist kein Fehler aufgetreten. Beträgt NDR 0, so sind ERROR und STATUS zu überprüfen, um den angezeigten Fehler zu finden.
- **LEN:** Anzahl von Lese-Bytes. Die Anzahl muss 20 betragen (= 5 x 4 Byte).

8 S7-Programmierbeispiel

8.1 Einführung

ADDI-DATA liefert Ihnen ein Softwarepaket, um die Verbindung zwischen einem MSX-E-System und einer S7-SPS über Modbus TCP zu ermöglichen. Das Programmierbeispiel basiert auf einer **SIMATIC S7** von Siemens.

Wenn Sie eine S7-kompatible SPS eines anderen Herstellers (z.B. IBHsoftec, VIPA etc.) verwenden möchten, müssen Sie auf die Original-FCs des jeweiligen Herstellers zurückgreifen. Eventuell sind auch die Datenbausteine etc. anzupassen (Siemens: 5 DWORD; VIPA: 20 Byte).

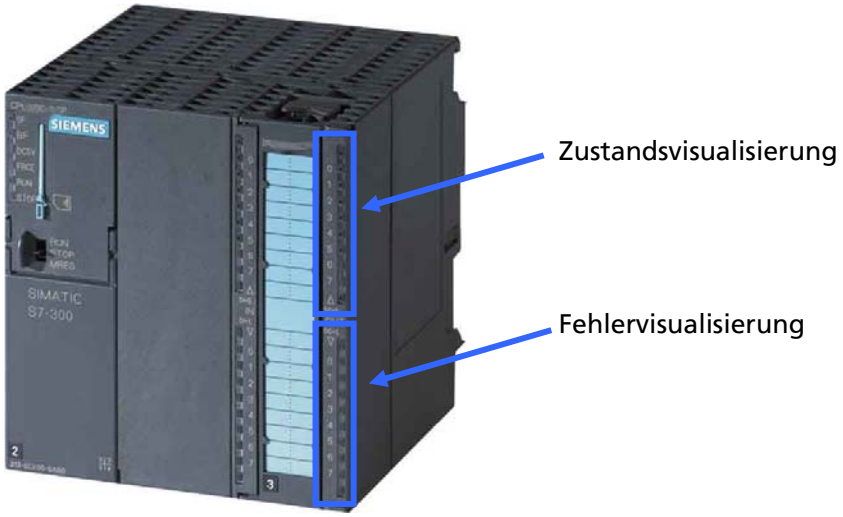
Das Programmierbeispiel „Sample MSXE-3701 (ADMB0000)“ wurde für die Ethernet-Kommunikation der **S7-300** mit **CP343-1** über das Modbus-TCP-Protokoll entwickelt. Es dient dem Lesen von Werten aus der TCP-Verbindung (TCP-Socket).

8.2 Voraussetzungen

Das Programmierbeispiel verwendet eine Zustandsmaschine zur Verwaltung der Funktionsaufrufe. Es besteht die Möglichkeit, den aktuellen Zustand der digitalen SPS-Ausgänge mit Hilfe von LEDs zu visualisieren. Diese LEDs können auch verwendet werden, um Fehler und Warnungen während der Entwicklungsphase anzuzeigen.

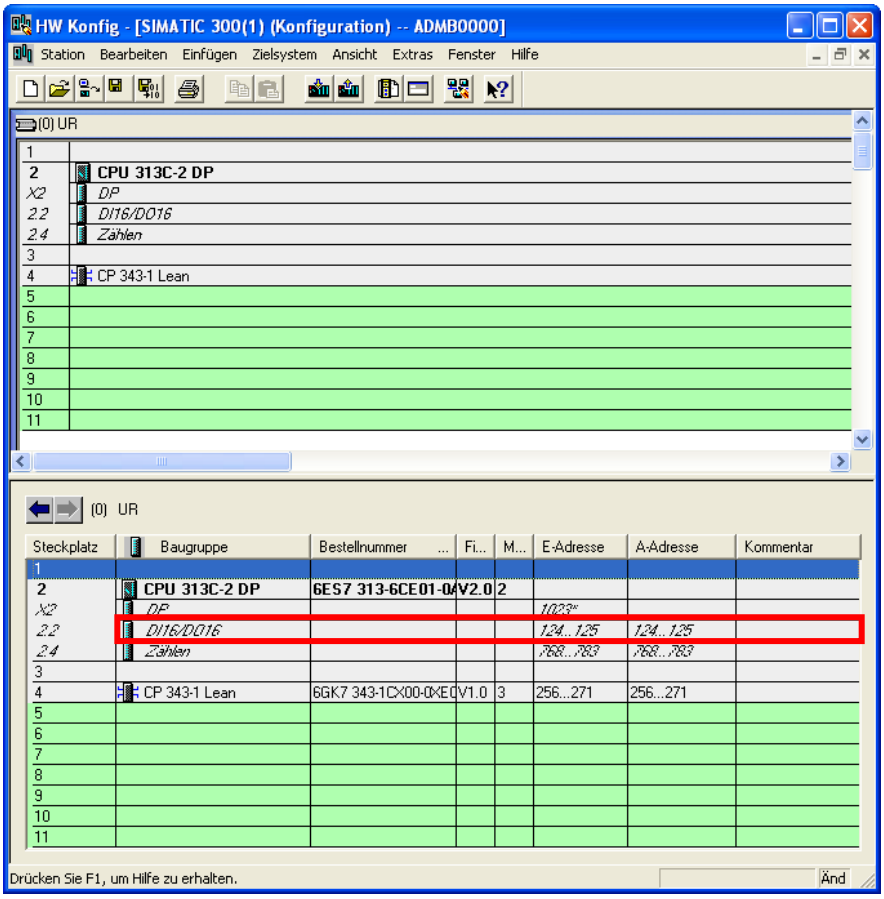
Wenn Sie die LEDs der digitalen Ausgänge nutzen möchten, müssen Sie im Programmierbeispiel die Registeradresse der digitalen Ausgänge an Ihre Hardware-Konfiguration anpassen.

Abb. 8-1: S7: LED-Visualisierung der digitalen Ausgänge



Die Bedeutung der einzelnen LEDs wird im Kommentar des Programmierbeispiels erläutert.

Abb. 8-2: SIMATIC Manager: LED-Konfiguration der digitalen Ausgänge



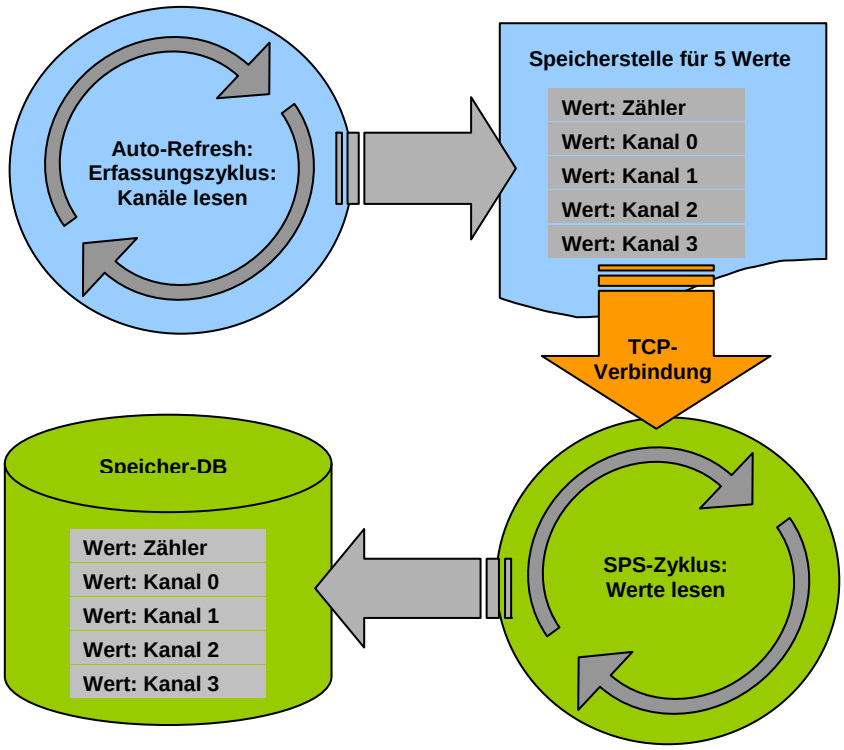
OB1 ruft die Zustandsmaschine FB1 auf, in der alle Schritte des Programmierbeispiels beschrieben werden.

9 Auto-Refresh-Modus und Sequenz-Modus

HINWEIS!
Wenn im Auto–Refresh-Modus keine Mittelwertberechnung bzw. im Sequenz-Modus keine Wartezeit konfiguriert ist, benötigt die SPS eine gewisse Zeit, um die Werte aus der TCP-Verbindung zu lesen (SPS-Zyklus > Erfassungszyklus). Ebenso ist dies der Fall, wenn in beiden Modi ein kleiner „division factor“ konfiguriert ist. Somit kann es zu einer Verzögerung zwischen dem aktuellen Status der Kanäle und der gelesenen Messwerte kommen.
Falls die Pakete aus der TCP-Verbindung nicht schnell genug gelesen werden, kann es einen FIFO-Überlauf geben, wodurch das MSX-E-System die Verbindung zu der SPS abbricht.

9.1 Erfassung im Auto-Refresh-Modus

Abb. 9-1: Auto-Refresh-Modus: Erfassungszyklus



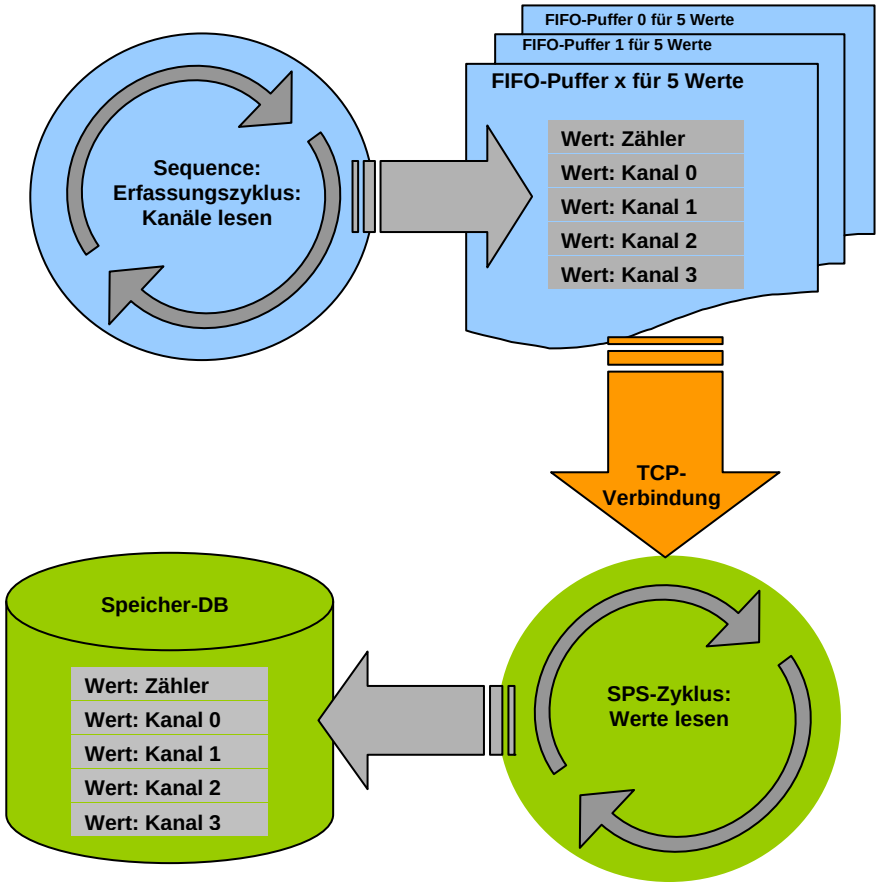
Der Erfassungszyklus wird mit vier Kanälen eines MSX-E-Systems im Auto-Refresh-Modus dargestellt (hellblau). Da für die fünf Werte nur eine Speicherstelle vorhanden ist, werden nach jedem Zyklus die Werte des vorherigen Zyklus mit denen des aktuellen Zyklus überschrieben.
Das MSX-E-System sendet die Werte an die SPS, sobald diese angeschlossen ist. Im SPS-Zyklus werden die Werte aus der TCP-Verbindung gelesen (grün).

MSX-E-Erfassungszyklus (Auto-Refresh-Modus) < SPS-Zyklus	->	Werte können verloren gehen
MSX-E-Erfassungszyklus (Auto-Refresh-Modus) > SPS-Zyklus	->	alle Werte werden empfangen
MSX-E-Erfassungszyklus (Auto-Refresh-Modus) = SPS-Zyklus	->	alle Werte werden empfangen

HINWEIS!
Bei einem Socket-FIFO-Überlauf können Werte verloren gehen.

9.2 Erfassung im Sequenz-Modus

Abb. 9-2: Sequenz-Modus: Erfassungszyklus



Der Erfassungszyklus wird mit vier Kanälen eines MSX-E-Systems im Sequenz-Modus dargestellt (hellblau). Die Werte des aktuellen Zyklus werden jeweils in einen neuen FIFO-Puffer geschrieben, d.h., es werden keine Werte überschrieben.
Das MSX-E-System sendet die Werte an die SPS, sobald diese angeschlossen ist. Im SPS-Zyklus werden die Werte aus der TCP-Verbindung gelesen (grün).

MSX-E-Erfassungszyklus (Sequenz-Modus) < SPS-Zyklus	->	alle Werte werden empfangen
MSX-E-Erfassungszyklus (Sequenz-Modus) > SPS-Zyklus	->	alle Werte werden empfangen
MSX-E-Erfassungszyklus (Sequenz-Modus) = SPS-Zyklus	->	alle Werte werden empfangen

HINWEIS!
Bei einem Socket-FIFO-Überlauf können Werte verloren gehen.

10 Anhang

10.1 Glossar

AWL
= Anweisungsliste

AWL ist eine Programmiersprache nach IEC 61131-3 zur SPS-Programmierung.

Code-Baustein
Im Siemens STEP 7-System entsprechen Code-Bausteine Organisationseinheiten des Anwenderprogramms. Man unterscheidet zwischen folgenden Code-Bausteintypen: Organisationsbaustein (OB), Funktionsbaustein (FB) und Funktion (FC).

Datenbaustein (DB)
Bei Datenbausteinen handelt es sich um Datenbereiche im Siemens STEP 7-System, in denen die Daten des Anwenderprogramms gespeichert werden. Es gibt zwei Datenbaustein-Typen: Instanz-Datenbausteine und Global-Datenbausteine. Erstere enthalten statische Lokaldaten, letztere Informationen, auf die ein Zugriff von allen Code-Bausteinen aus erfolgen kann. Instanz-Datenbausteine werden bei der Erstellung von Funktionsbausteinen (FBs) erzeugt, während Global-Datenbausteine vom Anwender programmiert werden müssen.

Ethernet
Hierbei handelt es sich um ein Basisband-Bussystem, das ursprünglich für die Verknüpfung von Minicomputern entwickelt wurde. Es basiert auf dem CSMA/CD-Zugriffs-verfahren. Als Übertragungsmedium dienen Koaxialkabel bzw. Twisted-Pair-Leitungen. Die Übertragungsgeschwindigkeiten betragen 10 Mbit/s (Ethernet), 100 Mbit/s (Fast Ethernet) sowie 1 Gbit/s bzw. 10 Gbit/s (Gigabit-Ethernet). Diese weit verbreitete Technik zum Vernetzen von Rechnern in einem LAN ist seit 1985 genormt (IEEE 802.3 und ISO 8802-3). Die Ethernet-Technologie hat sich im Büro-bereich allgemein durchgesetzt. Nach Ermöglichung auch sehr harter Echtzeit-anforderungen und Anpassung der Geräte-technik (Buskabel, Patchfelder, Anschlussdosen) an die rauen Einsatzbedingungen des industriellen Umfelds dringt sie zunehmend in die Feldbereiche der Automatisierungstechnik vor.

FIFO
= First in, First out

Organisationsprinzip für die Bedienung von Warteschlangen, bei dem die Abarbeitung von Aufträgen in der gleichen Reihenfolge erfolgt wie die Annahme. So werden z.B. beim Leeren eines Speichers zuerst eingespeicherte Daten als erste wieder ausgegeben.

Funktion (FC)
Im Siemens STEP 7-System gibt es den Code-Bausteintyp „Funktionen“, der Funktionen strukturierter Anwenderprogramme enthält. Dieser Code-Bausteintyp ist parametrierbar und kann Lokaldaten temporär speichern.

Funktionsbaustein (FB)
Funktionsbausteine strukturierter Anwender-programme zählen im Siemens STEP 7-System zu den Code-Bausteinen und enthalten aufgabenspezifische Unterprogramme. Zu einem Funktionsbaustein gehört immer ein Instanz-Datenbaustein, d.h. ein eigener Speicherbereich für spezielle Daten. FBs sind parametrierbar und können Lokaldaten sowohl statisch als auch temporär speichern.

IP-Adresse
Die IP-Adresse ist eine numerische Adresse, die einem Rechner im Internet zugeordnet ist und diesen eindeutig identifizierbar macht.
Sie besteht aus einer Folge von vier Zahlen-gruppen mit je maximal drei Ziffern, die jeweils durch Punkte voneinander getrennt sind. Fest mit dem Internet verbundene Rechner haben eine feste IP-Adresse.

Benutzer, die sich von Zeit zu Zeit ins Netz einwählen, erhalten vom Provider eine dynamische IP-Adresse, die nach Beendigung der Sitzung an den nächsten Client weitervergeben wird.

LSB
= Least Significant Bit

Die Bits sind unterschiedlich gewichtet, wobei das am weitesten rechts stehende die geringste Wertigkeit hat. Es wird daher als „Least Significant Bit“ (LSB) bezeichnet. Das erste Bit wird als „Most Significant Bit“ (MSB) bezeichnet, da es den höchsten Wert hat.

Modbus TCP
Das Modbus-TCP-Protokoll ist ein offenes Anfrage/Antwort-Protokoll für die Kommunikation zwischen Master und Slave bzw. Client und Server. Als Master, der die Kommunikationsvorgänge initialisiert, kann z.B. eine SPS fungieren. Die Daten werden in Form von TCP/IP-Paketen übertragen.

MSB
= Most Significant Bit

Die Bits sind unterschiedlich gewichtet. Das erste Bit hat den höchsten Wert und wird als „Most Significant Bit“(MSB) bezeichnet.

Organisationsbaustein (OB)
Organisationsbausteine gehören im Siemens STEP 7-System zu den Code-Bausteinen und bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. OBs können durch das Betriebssystem in bestimmten Situationen aufgerufen werden, wenn der sonst endlos ausgeführte OB1 unterbrochen wird.

Socket
Die bidirektionale Softwareschnittstelle zur Interprozess- (IPC) oder Netzwerk-kommunikation wird Socket genannt. Sockets bilden eine standardisierte Schnittstelle (API) zwischen dem Betriebssystem und der eigentlichen Anwendungssoftware.

SPS
= Speicherprogrammierbare Steuerung

Die SPS ist ein rechnerbasiertes Steuergerät, dessen Funktionalität durch ein Anwender-programm festgelegt wird. Mittels genormter Fachsprachen ist dieses Anwenderprogramm relativ einfach zu erstellen. Aufgrund der seriellen Arbeitsweise sind die Reaktionszeiten der SPS langsamer als bei VPS. Als Geräte-familien mit abgestuften und leistungsmäßig aufeinander abgestimmten Komponenten beherrschen die SPS heute alle Ebenen einer Automatisierungshierarchie.

Zähler
Der Zähler ist ein Schaltkreis, der Impulse zählt oder die Dauer von Impulsen messen kann.

10.2 Index

ADModbus-Bausteine 19
ADModbus-Bibliothek
  Einbindung 20
  Struktur 8
  Voraussetzungen 21
Auto-Refresh-Modus 43
Datenformat
  Intel-Format 37
  Motorola-Format 37
Datenkonvertierung 37
  Big-Endian-Format 37
  Little-Endian-Format 37
FC6 38
Funktionsbausteine 32
Glossar 46
Mnemonische Symbole 35
Modbus-TCP-Protokoll 7
MSX-E-Modbus-TCP-Funktionen
  Funktionsbausteine 10
  Funktionsnamen 11
  Funktionsparameter 12
  MSX-E-System-Fehler 17
Organisationsbausteine 19
Programmierbeispiel 41
Rückgabewert 17
Sequenz-Modus 43
Softwarepaket 7
SPS-Konfiguration 22
Taktmerker 22
TCP-Verbindung
  Datenserver 28
  Modbus-Server 23
Zustandsmaschine 9

11 Kontakt und Support

Haben Sie Fragen? Schreiben Sie uns oder rufen Sie uns an:

Postanschrift: ADDI-DATA GmbH
Airpark Business Center
Airport Boulevard B210
77836 Rheinmünster
Deutschland

Telefon: +49 7229 1847-0
Fax: +49 7229 1847-222

E-Mail: info@addi-data.com

Handbuch- und Software-Download im Internet:

www.addi-data.de