BECKHOFF New Automation Technology

Handbuch | DE

TS6250

TwinCAT 2 Modbus TCP Server

Supplement | Communication

10.08.2023 | Version: 1.2

Inhaltsverzeichnis

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

1.2 Zu Ihrer Sicherheit

Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit.
Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

Warnungen vor Personenschäden

⚠ GEFAHR

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

⚠ WARNUNG

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

⚠ VORSICHT

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

Warnung vor Umwelt- oder Sachschäden

HINWEIS

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

Information zum Umgang mit dem Produkt

Diese Information beinhaltet z. B.:
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.

1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem https://www.beckhoff.de/secguide.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

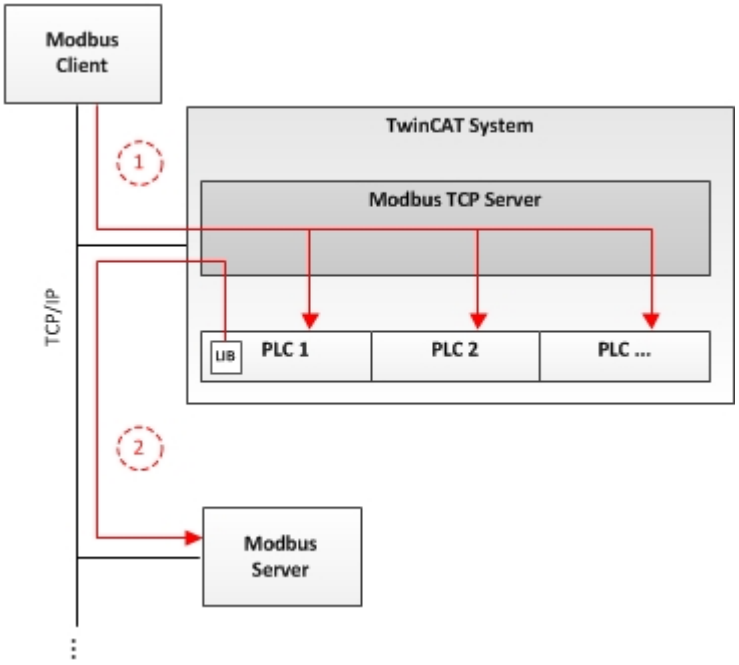
Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter https://www.beckhoff.de/secinfo.

2 Übersicht

Das Supplement ermöglicht über eine Netzwerkverbindung (TCP/IP) über das offene Protokoll Modbus zu kommunizieren.

Modbus ist ein offener Industriestandard der von der unabhängigen Modbus Organization gepflegt und verwaltet wird.

Das Protokoll basiert auf einer Client-/Server-Architektur. Das Supplement bietet die Möglichkeiten als Modbus Client oder Server zu dienen:



Server-Funktionalität: [▶ 12]

(1) Der TwinCAT Modbus TCP Server ermöglicht über das Netzwerk auf den vollständigen Speicherbereich der TwinCAT SPS zu zugreifen.

Client-Funktionalität: [▶ 18]

(2) Die mitgelieferte SPS-Bibliothek ermöglicht es, mit weiteren Modbus-Teilnehmern zu kommunizieren (2), um Informationen auszutauschen (z.B. Messwerte, Zustände) und zu steuern.

3 Systemvoraussetzungen

Technische Daten	TwinCAT Modbus TCP Server
Zielsystem	Windows NT/2000/XP/Vista/7 PC (x86-kompatibel)
Min. TwinCAT-Version	2.8.0
Min. TwinCAT-Level	TwinCAT PLC

4 Installation

Dieser Teil der Dokumentation führt den Benutzer Schritt-für-Schritt durch den Installationsvorgang des TwinCAT Modbus TCP Server Supplements für Windows basierte Betriebssysteme. Es wird hierbei auf die folgenden Themen eingegangen:

- Herunterladen der Setup-Datei
- Starten der Installation
- Nach der Installation

Herunterladen der Setup-Datei

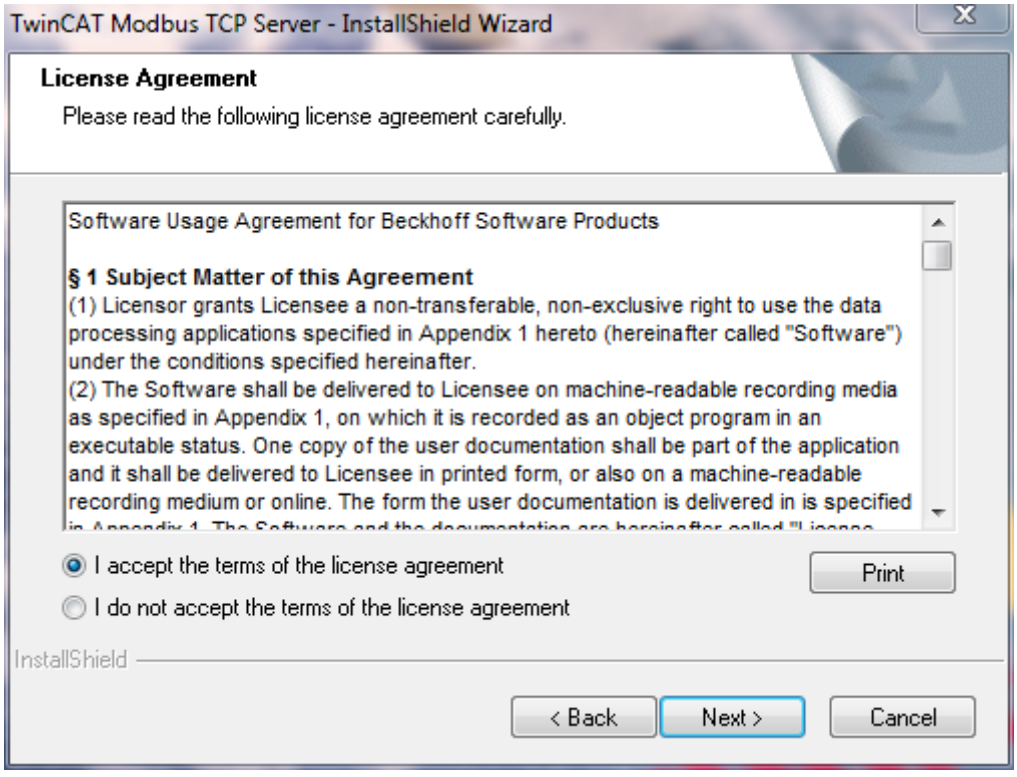
Wie viele andere TwinCAT Supplement-Produkte auch, steht TwinCAT Modbus TCP Server als Download auf dem Beckhoff FTP-Server zur Verfügung. Es handelt sich hierbei um die jeweils aktuelle Version des Produkts, welche sowohl in einer 30-Tage Demoversion, als auch als Vollversion lizensierbar ist. Bitte führen Sie die folgenden Schritte durch, um die Setup-Datei zu downloaden:

- Öffnen Sie eine Verbindung zu: TwinCAT Supplement Communication.
- Wählen Sie TS6250 TwinCAT Modus Server und starten Sie den Download über den Warenkorb.
- (Optional) Übertragen Sie die Datei auf das TwinCAT-Laufzeitsystem, auf welchem Sie das Supplement installieren möchten.

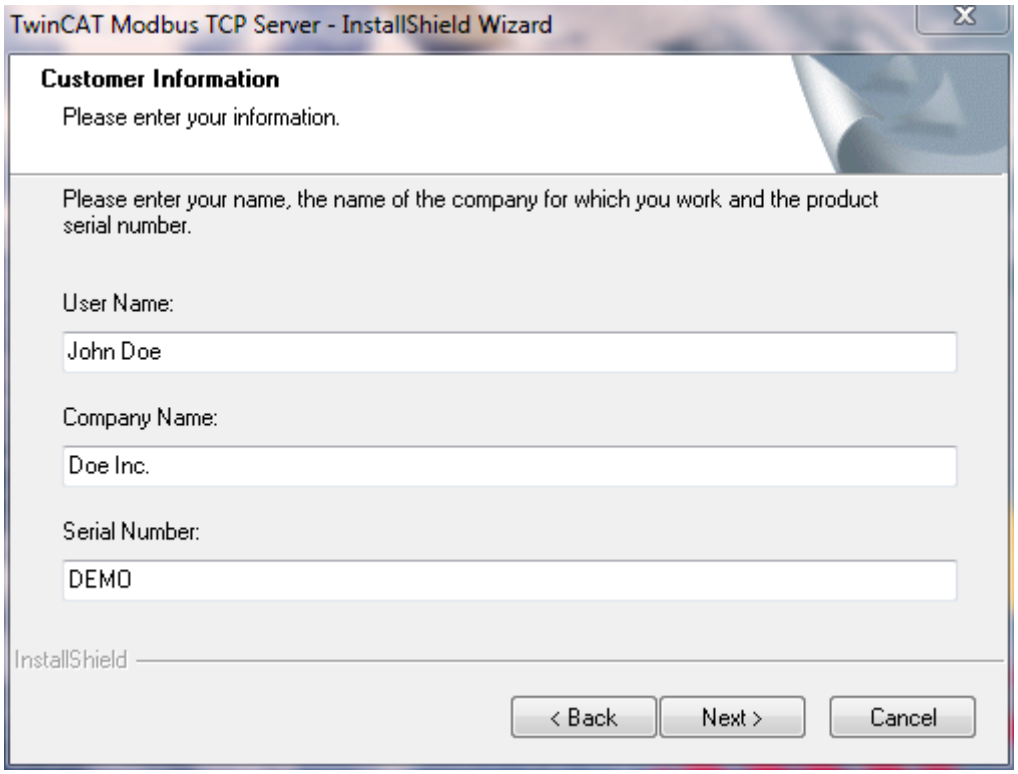
Starten der Installation

Um das Supplement zu installieren, führen Sie bitte die folgenden Schritte durch:

- Führen Sie einen Doppelklick auf die heruntergeladene Datei aus.
  Starten Sie die Installation unter Windows "Als Administrator ausführen", indem Sie die Setup-Datei mit der rechten Maustaste anklicken und die entsprechende Option im Kontextmenü auswählen.
- Wählen Sie eine **Sprache** aus, in der Sie die Software installieren möchten
- Klicken Sie auf "Next" und akzeptieren Sie dann die **Endbenutzervereinbarung**



- Geben Sie Ihre **Benutzerdaten** ein. Alle sichtbaren Felder sind hierbei Pflichtfelder. Möchten Sie eine 30-Tage Demoversion installieren, so geben Sie als Lizenzschlüssel bitte "DEMO" ein.



- Klicken Sie auf **"Install"** um die Installation zu starten
- Zum Abschluss der Installation starten Sie Ihren Computer neu

Nach der Installation

Das Supplement-Produkt "TwinCAT Modbus TCP" wird automatisch durch das Setup konfiguriert.

Weitere mögliche Schritte sind:

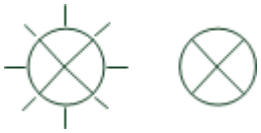
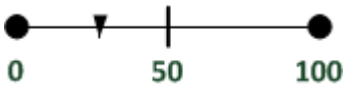
- Der Modbus TCP Server wird von TwinCAT mitgestartet und stellt das Mapping [▶ 12] für Modbus und des physikalischen Prozessabbilds per default bereit
- Mittels des mitgelieferten Konfigurators [▶ 12] können Sie das Mapping des Modbus TCP Servers anpassen

5 Konfiguration

Der TwinCAT Modbus TCP Server kann Modbus-Funktionen über TCP/IP empfangen.

Modbus-Bereiche

Nach der Modbus Spezifikation sind die folgenden vier Modbus-Bereiche definiert:

Modbus-Bereiche	Datentyp	Zugriff	Anwendungsbeispiel
digitale Eingänge (Discrete Inputs)	1 Bit	nur Lesen	
digitale Ausgänge (Coils)	1 Bit	Lesen und Schreiben	
Eingangs-Register	16 Bit	nur Lesen	
Ausgangs-Register	16 Bit	Lesen und Schreiben	

Nach der Installation sind die Modbus-Bereiche auf die Speicherbereiche der SPS gemappt.

Die Standardeinstellung des Modbus TCP Servers können Sie im Artikel über das Default-Mapping [▶ 14] finden. Das Anpassen der Einstellung ermöglicht der Konfigurator [▶ 12].

ADS-Zugriff

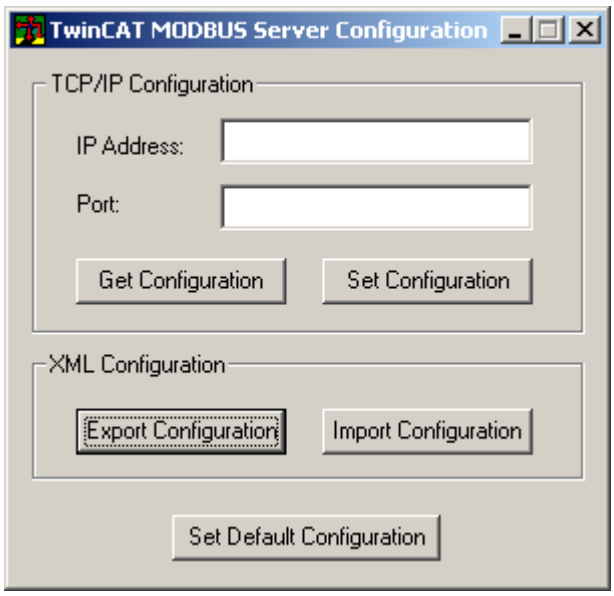
Damit auf die spezifizierten Modbus-Bereiche per ADS zugegriffen werden kann, fügen Sie bitte folgende Globalen Variablen zu Ihrem SPS-Projekt hinzu.

```
VAR_GLOBAL
    mb_Input_Coils       : ARRAY [0..255] OF BOOL;
    mb_Output_Coils      : ARRAY [0..255] OF BOOL;
    mb_Input_Registers   : ARRAY [0..255] OF WORD;
    mb_Output_Registers  : ARRAY [0..255] OF WORD;
END_VAR
```

5.1 TwinCAT Modbus TCP Konfigurator

Das Setup installiert den Konfigurator (TcModbusSrvCfg.exe) standardmäßig unter dem Verzeichnis **\TwinCAT\TcModbusSrv**.

Er ermöglicht die aktive Einstellung des Servers auszulesen und zu ändern.



IP Address: Adresse worunter der Server erreichbar ist. Wenn keine Adresse gesetzt ist, wird die lokale IP-Adresse verwendet (Default-Einstellung).

Port: Konfigurierter Port worunter der TwinCAT Modbus TCP Server angesprochen werden kann (Default-Port = 502).

Get Configuration: Auslesen der IP-Adresse und des Ports.

Set Configuration: Setzen der IP-Adresse und des Ports.

Export Configuration: Auslesen der aktuellen Konfiguration.

Import Configuration: Einlesen einer neuen Konfiguration.

Set Default Configuration: Stellt die Standarteinstellung wieder her (verwenden der lokalen IP-Adresse, Port = 502, Default Modbus-Mapping [▶ 14]).

Zum Auslesen und Setzen der Konfiguration muss TwinCAT gestoppt werden, was vom Konfigurator übernommen wird.

Auslesen der Konfiguration

Die Konfiguration ist XML-basierend und kann mit einem Texteditor gelesen und modifiziert werden

Mit Ausführen von "Export Configuration" kann die aktuelle Konfiguration als XML-Datei lokal gespeichert werden.

Am einfachsten ist es, eine zuvor exportierte Konfiguration zu editieren und dann wieder zu importieren.

Einlesen der Mapping-Information

Mit "Import Configuration" kann eine (geänderte) Konfiguration eingelesen und aktiviert werden.

Beispiel eines einfachen Mappings:

```
<Configuration>
    <Port>502</Port>
    <IpAddr\>
        <Mapping>
            <InputRegisters>
                <MappingInfo>
                    <!--Port 801 = PLC1 TC2 -->
                    <AdsPort>801</AdsPort>
                    <StartAddress>0</StartAddress>
```

```
                    <EndAddress>32767</EndAddress>
                    <!-- IndexGroup 61472 = 0xF020 ->physical plc input register %I -->
                    <IndexGroup>61472</IndexGroup>
                    <IndexOffset>0</IndexOffset>
                </MappingInfo>
                <MappingInfo>
                    <AdsPort>801</AdsPort>
                    <!-- Modbus input registers -->
                    <StartAddress>32768</StartAddress>
                    <EndAddress>33023</EndAddress>
                    <VarName>.mb_Input_Registers</VarName>
                </MappingInfo>
            </InputRegisters>;
            <OutputRegisters/>
            <InputCoils/>
            <OutputCoils/>
        </Mapping>
</Configuration>
```

Diese Beispiel mappt die Eingangsregister (Indexgroup 0xF020) des ersten TwinCAT2 Laufzeitsystems (Port = 801) und stellt die Modbus-Eingangsregister zur Verfügung.

Das Mapping kann per Variablennamen oder IndexGroup/Offset (performanter) erfolgen.

Die ausführliche XML der Standardkonfiguration finden Sie unter dem Default Modbus-Mapping [▶ 14].

5.2 Mapping zwischen Modbus und ADS

Das Standard-Mapping des Servers wird in der folgenden Tabelle f r das erste Laufzeitsystem dargestellt:

Modbus-Bereiche	Modbus-Adresse	ADS-Bereich	
digitale Eing nge (Inputs)	0x0000 - 0x7FFF	**Indexgruppe**	**Indexoffset**
		0xF021 - Prozessabbild der physikalischen Eing nge (Bit-Zugriff)	0x0
	0x8000 - 0x80FF	**Name der Variablen im SPS-Programm**	**Datentyp**
		.mb_Input_Coils	ARRAY [0..255] OF BOOL
digitale Ausg nge (Coils)	0x0000 - 0x7FFF	**Indexgruppe**	**Indexoffset**
		0xF031 - Prozessabbild der physikalischen Ausg nge (Bit-Zugriff)	0x0
	0x8000 - 0x80FF	**Name der Variablen im SPS-Programm**	**Datentyp**
		.mb_Output_Coils	ARRAY [0..255] OF BOOL
Eingangs-Register (Input Registers)	0x0000 - 0x7FFF	**Indexgruppe**	**Indexoffset**
		0xF020 - Prozessabbild der physikalischen Eing nge	0x0
	0x8000 - 0x80FF	**Name der Variablen im SPS-Programm**	**Datentyp**
		.mb_Input_Registers	ARRAY [0..255] OF WORD
Ausgangs-Register (Registers)	0x0000 - 0x2FFF	**Indexgruppe**	**Indexoffset**
		0xF030 - Prozessabbild der physikalischen Ausg nge	0x0
	0x3000 - 0x5FFF	0x4020 - SPS-Memory-Bereich	0x0
	0x6000 - 0x7FFF	0x4040 - SPS-Daten-Bereich	0x0

Modbus-Bereiche	Modbus-Adresse	ADS-Bereich	
	0x8000 - 0x80FF	**Name der Variablen im SPS-Programm**	**Datentyp**
		.mb_Output_Registers	ARRAY [0..255] OF WORD

Der Server mappt diese auf die einzelnen Ads-Bereiche und ermöglicht den Zugriff auf das physikalische Prozessabbild und die SPS-Merker Bereiche.

Das Anpassen der Einstellung ermöglicht der Konfigurator [▶ 12].

Default XML

Die Standartkonfiguration sieht wie folgt aus:

```
<Configuration>
    <!-- Modbus TCP port, default = 502-->
    <Port>502</Port>
    <!-- optional IP configuration for Modbus TCP server-->
    <IpAddr/>
    <Mapping>
    <InputCoils>
        <MappingInfo>
        <!-- AdsPort: TwinCAT2 PLC1 = 801, PLC2 = 811...-->
        <AdsPort>801</AdsPort>
        <StartAddress>0</StartAddress>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 61473 = 0xF021 -> physical plc inputs %IX -->
        <IndexGroup>61473</IndexGroup>
        <!-- Bit offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus input coils -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Input_Coils</VarName>
        </MappingInfo>
    </InputCoils>
    <OutputCoils>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 61489 = 0xF031 -> physical plc outputs %QX -->
        <IndexGroup>61489</IndexGroup>
        <!-- Bit offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus output coils-->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Output_Coils</VarName>
        </MappingInfo>
    </OutputCoils>
    <InputRegisters>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>0</StartAddress>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 61472 = 0xF020 -> physical plc input register %I -->
        <IndexGroup>61472</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus input registers -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Input_Registers</VarName>
        </MappingInfo>
```

```
    </InputRegisters>
    <OutputRegisters>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>0</StartAddress>
        <EndAddress>12287</EndAddress>
        <!-- IndexGroup 61488 = 0xF030 -> physical plc output register %Q -->
        <IndexGroup>61488</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>12288</StartAddress>
        <EndAddress>24575</EndAddress>
        <!-- IndexGroup 16416 = 0x4020 -> plc memory area %M -->
        <IndexGroup>16416</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <StartAddress>24576</StartAddress>
        <EndAddress>32767</EndAddress>
        <!-- IndexGroup 16448 = 0x4040 -> plc data area -->
        <IndexGroup>16448</IndexGroup>
        <!-- Byte offset-->
        <IndexOffset>0</IndexOffset>
        </MappingInfo>
        <MappingInfo>
        <AdsPort>801</AdsPort>
        <!-- Modbus output registers -->
        <StartAddress>32768</StartAddress>
        <EndAddress>33023</EndAddress>
        <VarName>.mb_Output_Registers</VarName>
        </MappingInfo>
    </OutputRegisters>
    </Mapping>
</Configuration>
```

6 Modbus ADS Diagnose Interface

Per ADS können folgende Information abgefragt werden:

Index Group	Index Offset	Zugriff	Datentyp	Beschreibung	Minimale Modbus Server Version
0x2000	0	ADS Read	UINT32	**GetConnectedClientCount** Rückgabe der Anzahl von verbundenen Modbus Clients	1.0.50
0x2000	1	ADS Read	UINT32	**GetModbusRequestCount** Rückgabe der empfangenen Modbus Anfragen	1.0.50
0x2000	2	ADS Read	UINT32	**GetModbusResponseCount** Rückgabe der gesendeten Modbus Antworten	1.0.50

7 SPS-Bibliotheken

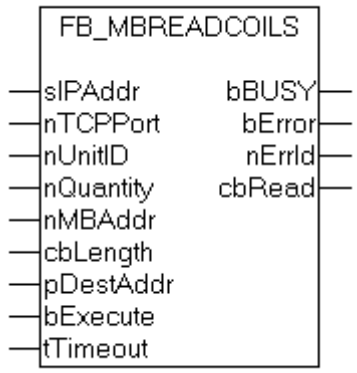
7.1 TcModbusSrv

Die im Modbus-Protokoll definierten Funktionen sind als SPS-Bausteine in der Bibliothek TcModbusSrv.lib realisiert.

Modbus TCP Funktion	Funktions-Code	SPS-Baustein
Read Coils	1	FB_MBReadCoils [▶ 18]
Read Inputs	2	FB_MBReadInputs [▶ 20]
Read Registers	3	FB_MBReadRegs [▶ 22]
Read Input Registers	4	FB_MBReadInputRegs [▶ 24]
Write Single Coil	5	FB_MBWriteSingleCoil [▶ 26]
Write Single Register	6	FB_MBWriteSingleReg [▶ 28]
Write Multiple Coils	15	FB_MBWriteCoils [▶ 30]
Write Multiple Registers	16	FB_MBWriteRegs [▶ 32]
Read/Write Multiple Registers	23	FB_MBReadWriteRegs [▶ 34]
Diagnose	8	FB_MBDiagnose [▶ 36]

Modbus UDP Funktion	Funktions-Code	SPS-Baustein
Read Coils	1	FB_MBUdpReadCoils [▶ 38]
Read Inputs	2	FB_MBUdpReadInputs [▶ 39]
Read Registers	3	FB_MBUdpReadRegs [▶ 41]
Read Input Registers	4	FB_MBUdpReadInputRegs [▶ 42]
Write Single Coil	5	FB_MBUdpWriteSingleCoil [▶ 43]
Write Single Register	6	FB_MBUdpWriteSingleReg [▶ 44]
Write Multiple Coils	15	FB_MBUdpWriteCoils [▶ 46]
Write Multiple Registers	16	FB_MBUdpWriteRegs [▶ 47]
Read/Write Multiple Registers	23	FB_MBUdpReadWriteRegs [▶ 48]
Diagnose	8	FB_MBUdpDiagnose [▶ 50]

7.1.1 FB_MBReadCoils (Modbus-Funktion 1)



Diese Funktion wird zum Lesen von 1 bis 2048 digitaler Ausgänge (Coils) benutzt. Ein digitaler Ausgang entspricht einem Bit der gelesenen Datenbytes.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden digitalen Eingänge (Datenbits). Der Wert Null ist unzulässig.

nMBAddr: Startadresse der zu lesenden digitalen Eingänge (Bitoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer muss mindestens die Bytegröße: *(nQuantity + 7) / 8* besitzen.

pDestAdd: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

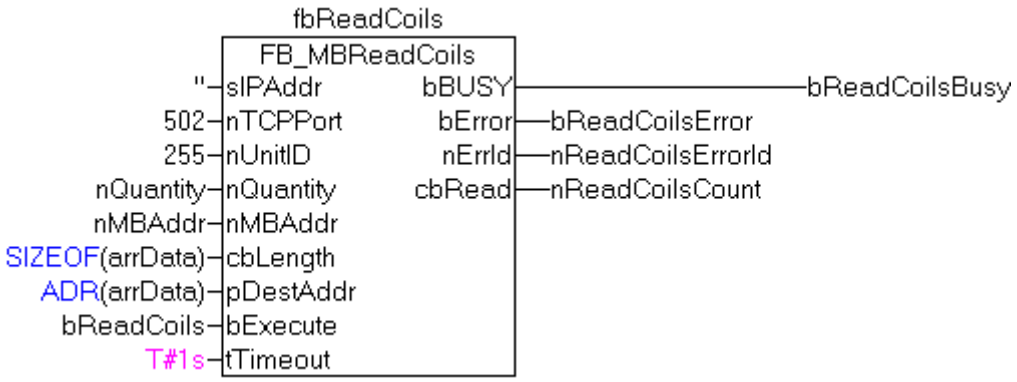
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbReadCoils         : FB_MBReadCoils;
    bReadCoils          : BOOL;
    bReadCoilsBusy      : BOOL;
```

```
    bReadCoilsError     : BOOL;
    nReadCoilsErrorId   : UDINT;
    nReadCoilsCount     : UDINT;
    nQuantity           : WORD := 10;
    nMBAddr             : WORD := 5;
    arrData             : ARRAY [1..2] OF BYTE;
END_VAR
```



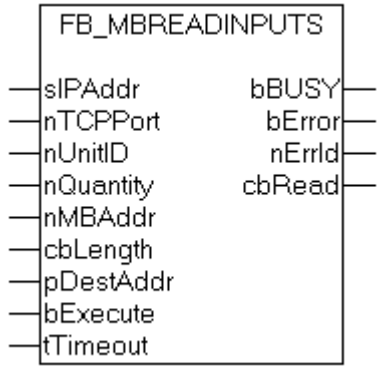
Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadCoils-Befehls, wird der Inhalt der digitalen Ausgänge 6 - 15 in das Array arrData geschrieben:

Digitale Ausgänge	Array-Offset	Status
6-13	1	0x54 Status des Ausgangs 13 ist das MSB dieses Bytes (ganz links) Status des Ausgangs 6 ist das LSB dieses Bytes (ganz rechts)
14-15	2	0x02 da nur 10 Ausgänge gelesen werden sollen, werden die restlichen Bits (3-8) auf 0 gesetzt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.2 FB_MBReadInputs (Modbus-Funktion 2)



Diese Funktion wird zum Lesen von 1 bis 2048 digitalen Eingängen benutzt. Ein digitaler Eingang entspricht einem Bit der gelesenen Datenbytes.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden digitalen Eingänge (Datenbits). Der Wert Null ist nicht zulässig.

nMBAddr: Startadresse der zu lesenden digitalen Eingänge (Bitoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers für die zu lesenden Datenbytes. Der Puffer muss mindestens die Bytegröße: *(nQuantity + 7) / 8* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

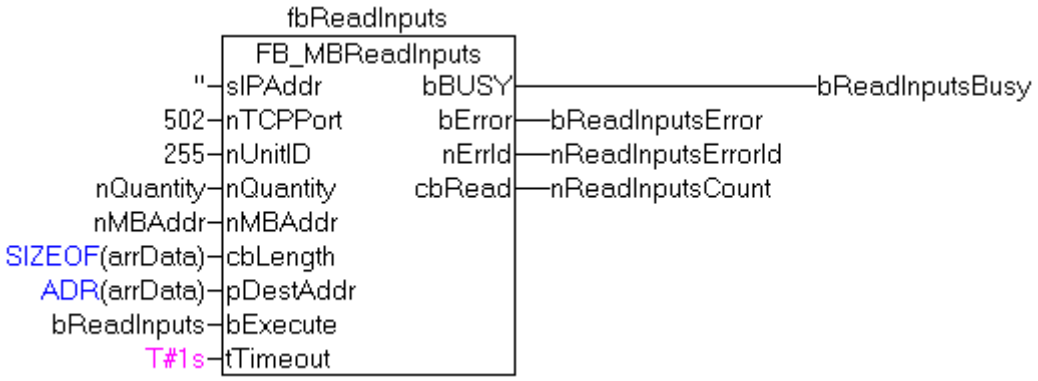
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbReadInputs            : FB_MBReadInputs;
    bReadInputs             : BOOL;
    bReadInputsBusy         : BOOL;
```

```
    bReadInputsError        : BOOL;
    nReadInputsErrorId      : UDINT;
    nReadInputsCount        : UDINT;
    nQuantity               : WORD := 20;
    nMBAddr                 : WORD := 29;
    arrData                 : ARRAY [1..3] OF BYTE;
END_VAR
```



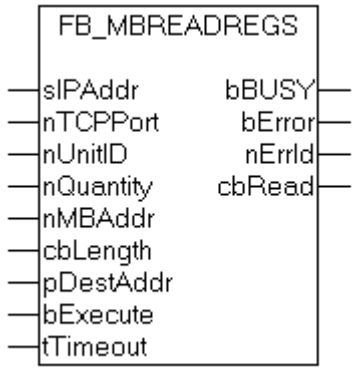
Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadInputs-Befehls, wird der Inhalt der digitalen Eingänge 30 - 49 in das Array arrData geschrieben:

Digitale Ausgänge	Array-Offset	Status
29-36	1	0x34 Status des Eingangs 36 ist das MSB dieses Bytes (ganz links) Status des Eingangs 29 ist das LSB dieses Bytes (ganz rechts)
37-44	2	0x56 Status des Eingangs 44 ist das MSB dieses Bytes (ganz links) Status des Eingangs 37 ist das LSB dieses Bytes (ganz rechts)
45-49	3	0x07 da nur 20 Eingänge gelesen werden sollen, werden die restlichen Bits (5-8) auf 0 gesetzt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.3 FB_MBReadRegs (Modbus-Funktion 3)



Diese Funktion wird zum Lesen von 1 bis 128 Ausgangs-Register (16 Bit) benutzt. Das erste Byte enthält die unteren acht Bits und das zweite Byte die oberen acht Bits.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden Ausgangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBAddr: Startadresse der zu lesenden Ausgangs-Register (Wortoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers für die zu lesenden Registerwerte. Der Puffer muss mindestens die Bytegröße: *nQuantity * 2* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

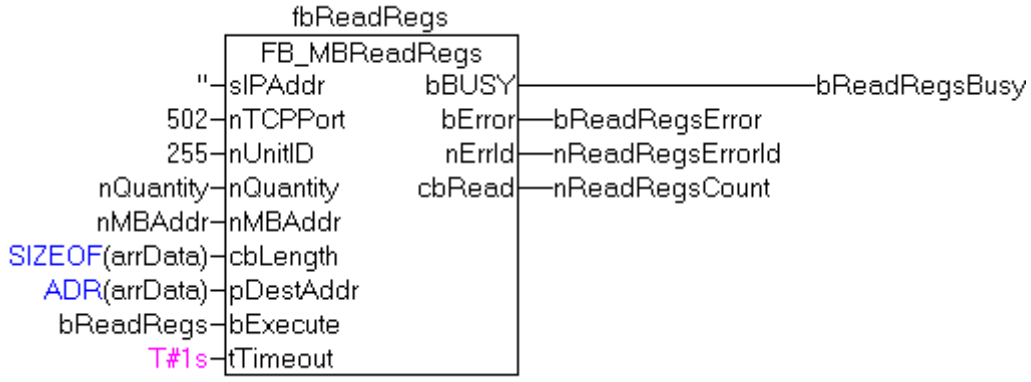
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbReadRegs          : FB_MBReadRegs;
    bReadRegs           : BOOL;
    bReadRegsBusy       : BOOL;
    bReadRegsError      : BOOL;
    nReadRegsErrorId    : UDINT;
    nReadRegsCount      : UDINT;
    nQuantity           : WORD:=2;
    nMBAddr             : WORD:=24;
    arrData             : ARRAY [1..2] OF WORD;
END_VAR
```



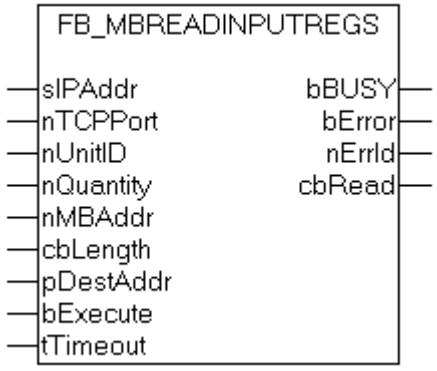
Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadRegs-Befehls, befinden sich der Inhalt der Register 25 und 26 in dem Array arrData:

Register	Array-Offset	Status
25	1	0x1234 (als Byte 0x34 0x12)
26	2	0x5563 (als Byte 0x63 0x55)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.4 FB_MBReadInputRegs (Modbus-Funktion 4)



Diese Funktion wird zum Lesen von 1 bis 128 Eingangs-Register (16Bit) benutzt. Endian

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden Eingangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBAddr: Startadresse der zu lesenden Eingangs-Register (Wortoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers. Der Puffer muss mindestens die Bytegröße: *nQuantity * 2* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

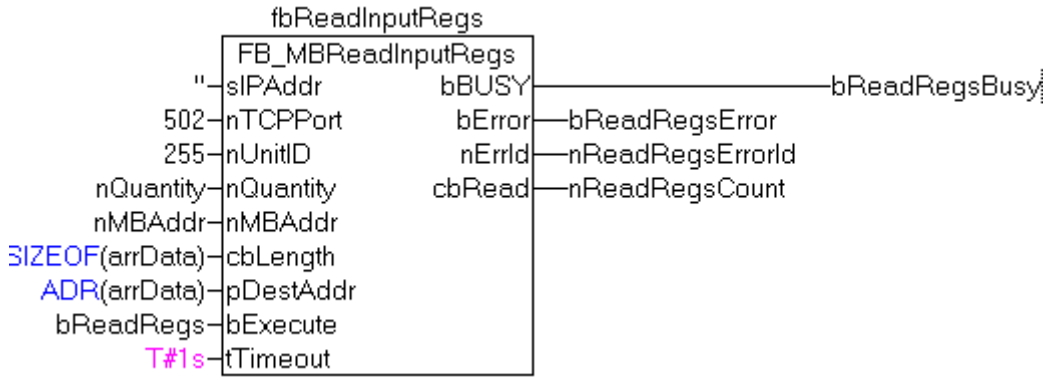
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbReadRegs          : FB_MBReadRegs;
    bReadRegs           : BOOL;
    bReadRegsBusy       : BOOL;
```

```
    bReadRegsError      : BOOL;
    nReadRegsErrorId    : UDINT;
    nReadRegsCount      : UDINT;
    nQuantity           : WORD := 3;
    nMBAddr             : WORD:= 2;
    arrData             : ARRAY [1..3] OF WORD;
END_VAR
```



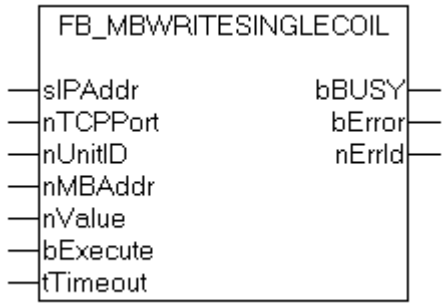
Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadRegs-Befehls, befinden sich der Inhalt der Register 3-5 in dem Array arrData:

Register	Array-Offset	Status
3	1	0x4543 (als Byte 0x43 0x45)
4	2	0x5234 (als Byte 0x34 0x52)
5	2	0x1235 (als Byte 0x35 0x12)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.5 FB_MBWriteSingleCoil (Modbus-Funktion 5)



Diese Funktion wird zum Beschreiben eines digitalen Ausgangs benutzt (Coil). Dabei handelt es sich um einen Bit-Zugriff.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nMBAddr         : WORD;
    nValue          : WORD;
```

```
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nMBAddr: Adresse des digitalen Ausgangs (Bitoffset).

nValue: Wert, der in den digitalen Ausgang geschrieben werden soll. Der Wert 16#FF00 schaltet den Ausgang ein und der Wert 16#0000 schaltet den Ausgang ab.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

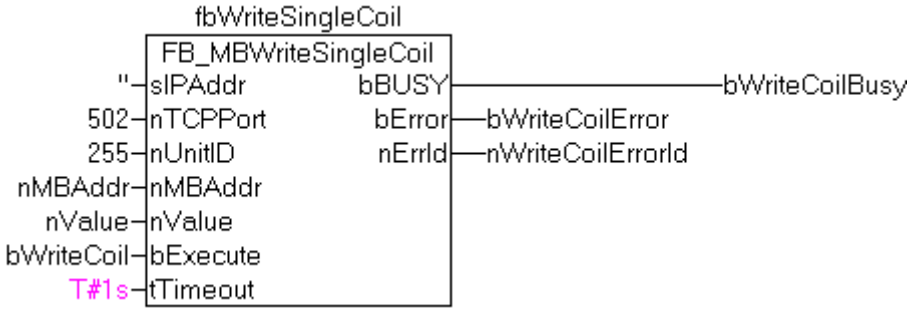
bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbWriteSingleCoil       : FB_MBWriteSingleCoil;
    bWriteCoil              : BOOL;
    bWriteCoilBusy          : BOOL;
    bWriteCoilError         : BOOL;
    nWriteCoilErrorId       : UDINT;
    nMBAddr                 : WORD := 3;
    nValue                  : WORD := 16#FF00;
END_VAR
```

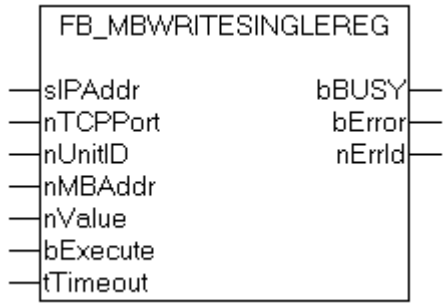


Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des WriteSingleCoil-Befehls, wird der digitale Ausgang 4 angeschaltet.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.6 FB_MBWriteSingleReg (Modbus-Funktion 6)



Diese Funktion wird zum Beschreiben eines einzelnen Ausgangsregisters benutzt. Dabei handelt es sich um einen 16 Bit-Zugriff.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nMBAddr         : WORD;
    nValue          : WORD;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nMBAddr: Adresse des Ausgangs-Registers (Wortoffset).

nValue: Wert, der in das Register geschrieben werden soll (Datenwort).

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

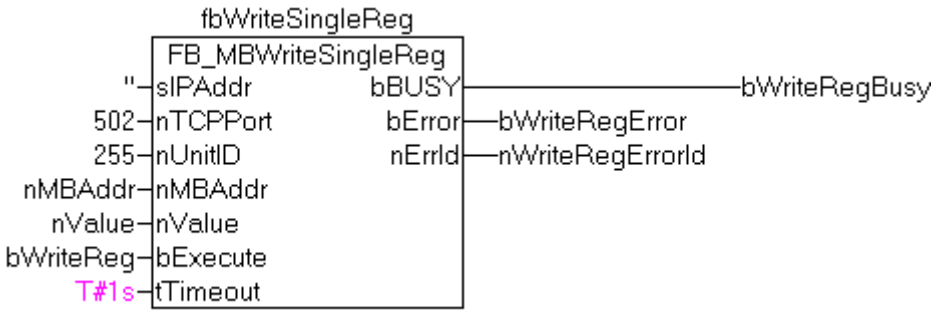
bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbWriteSingleReg    : FB_MBWriteSingleReg;
    bWriteReg           : BOOL;
    bWriteRegBusy       : BOOL;
    bWriteRegError      : BOOL;
    nWriteRegErrorId    : UDINT;
    nMBAddr             : WORD := 4;
    nValue              : WORD := 16#1234;
END_VAR
```

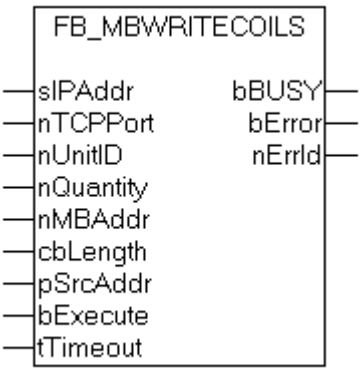


Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des WriteSingleReg-Befehls, wird in das Register 5 der Wert 16#1234 geschrieben.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.7 FB_MBWriteCoils (Modbus-Funktion 15)



Diese Funktion wird zum Beschreiben von 1 bis 2048 digitaler Ausgänge (Coils) benutzt. Ein digitaler Ausgang entspricht einem Bit der geschriebenen Datenbytes.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pSrcAddr        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der digitalen Ausgänge, die beschrieben werden sollen (Datenbits). Der Wert Null ist unzulässig.

nMBAddr: Startadresse der digitalen Ausgänge, die beschrieben werden sollen (Bitoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer muss mindestens die Bytegröße: *(nQuantity + 7) / 8* besitzen.

pSrcAddr: Enthält die Adresse des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

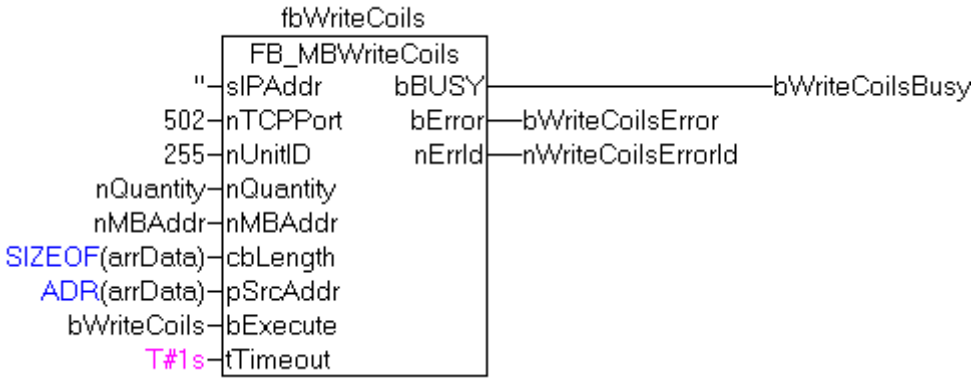
bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbWriteCoils        : FB_MBWriteCoils;
    bWriteCoils         : BOOL;
    bWriteCoilsBusy     : BOOL;
    bWriteCoilsError    : BOOL;
    nWriteCoilsErrorId  : UDINT;
    nWriteCoilsCount    : UDINT;
    nQuantity           : WORD := 10;
    nMBAddr             : WORD := 14;
    arrData             : ARRAY [1..2] OF BYTE := 16#75,16#03;
END_VAR
```



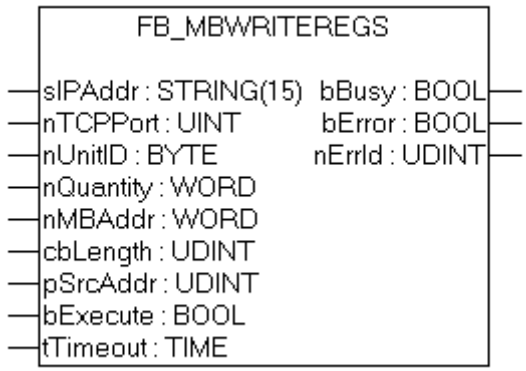
Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadCoils-Befehls, wird der Inhalt des Arrays arrData in die Ausgänge 15-24 geschrieben:

Bit	0	1	1	1	0	1	0	1	0	0	0	0	0	0	1	1
Output	22	21	20	19	18	17	16	15	X	X	X	X	X	X	24	23

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.8 FB_MBWriteRegs (Modbus-Funktion 16)



Diese Funktion wird zum Beschreiben von 1 bis 128 Ausgangs-Register (16 Bit) benutzt.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pSrcAddr        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu schreibenden Ausgangs-Register (Datenworte).

nMBAddr: Startadresse der zu schreibenden Ausgangs-Register (Wortoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Quellpuffers der die zu schreibende Registerwerte enthält. Der Puffer muss mindestens die Bytegröße: *nQuantity * 2* besitzen.

pSrcAddr: Enthält die Adresse des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

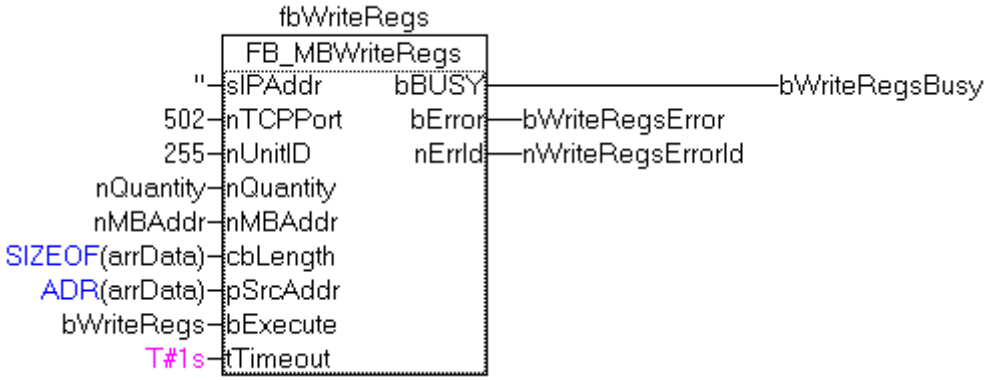
bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbWriteRegs         : FB_MBWriteRegs;
    bWriteRegs          : BOOL;
    bWriteRegsBusy      : BOOL;
    bWriteRegsError     : BOOL;
    nWriteRegsErrorId   : UDINT;
    nWriteRegsCount     : UDINT;
    nQuantity           : WORD := 3;
    nMBAddr             : WORD := 4;
    arrData             : ARRAY [1..3] OF WORD;
END_VAR
```

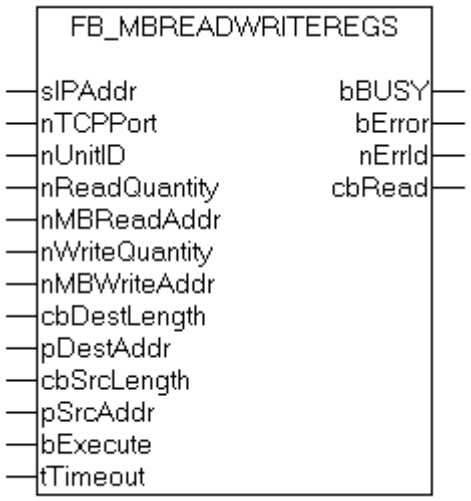


Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadRegs-Befehls, wird der Inhalt des Arrays arrData in die Register 5-7 geschrieben.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.9 FB_MBReadWriteRegs (Modbus-Funktion 23)



Diese Funktion liest zuerst 1 bis 128 Ausgangs-Register (16 Bit) und beschreibt danach 1 bis 128 Ausgangs-Register (16 Bit).

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nReadQuantity   : WORD;
    nMBReadAddr     : WORD;
    nWriteQuantity  : WORD;
    nMBWriteAddr    : WORD;
    cbDestLength    : UDINT;
    pDestAddr       : UDINT;
    cbSrcLength     : UDINT;
    pSrcAddr        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nReadQuantity: Anzahl der zu lesenden Ausgangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBReadAddr: Startadresse der zu lesenden Ausgangs-Register (Wortoffset).

nWriteQuantity: Anzahl der zu schreibenden Ausgangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBWriteAddr: Startadresse der zu schreibenden Ausgangs-Register (Wortoffset).

cbDestLength: Enthält die max. verfügbare Bytegröße des Zielpuffers für die zu lesenden Registerwerte. Der Puffer muss mindestens die Bytegröße: *nReadQuantity * 2* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

cbSrcLength: Enthält die max. verfügbare Bytegröße des Quellpuffers der die zu schreibende Registerwerte enthält. Der Puffer muss mindestens die Bytegröße: *nWriteQuantity* 2* besitzen.

pSrcAddr: Enthält die Adresse des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

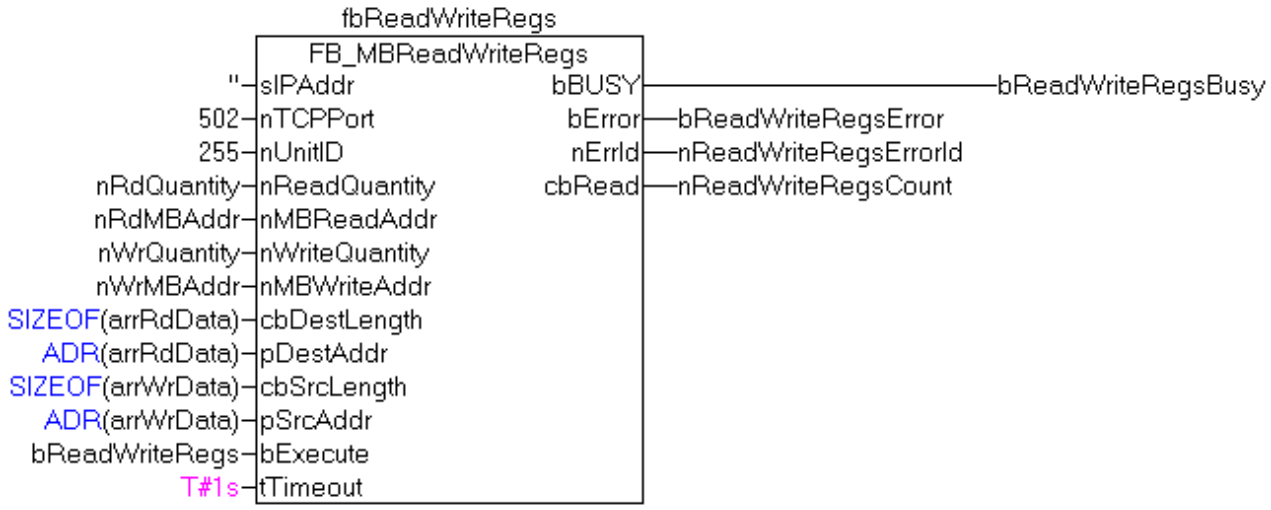
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbReadWriteRegs         : FB_MBReadWriteRegs;
    bReadWriteRegs          : BOOL;
    bReadWriteRegsBusy      : BOOL;
    bReadWriteRegsError     : BOOL;
    nReadWriteRegsErrorId   : UDINT;
    nReadWriteRegsCount     : UDINT;
    nRdQuantity             : WORD;
    nRdMBAddr               : WORD;
    nWrQuantity             : WORD;
    nWrMBAddr               : WORD;
    arrRdData               : ARRAY [1..9] OF WORD;
    arrWrData               : ARRAY [1..9] OF WORD;
END_VAR
```

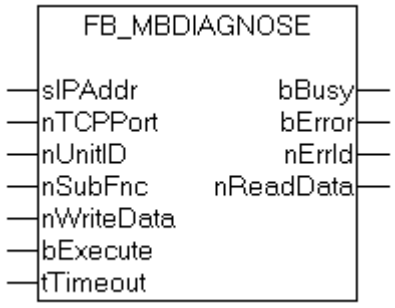


Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des ReadWriteRegs-Befehls, befinden sich in arrRdData die gelesenen Daten der Register und die Daten aus arrWrData werden in die Register geschrieben.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.10 FB_MBDiagnose (Modbus-Funktion 8)



Die Diagnose-Funktion stellt eine Reihe von Tests für die Überprüfung des Übertragungssystems zwischen dem Master und dem Slave, oder für die Überprüfung der verschiedenen internen Fehlerzustände innerhalb des Slaves, zur Verfügung.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nSubFnc         : WORD;
    nWriteData      : WORD;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nSubFnc: Die Subfunktion, die ausgeführt werden soll.

nWriteData: Das Datenwort, das geschrieben werden soll.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    nReadData   : WORD;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

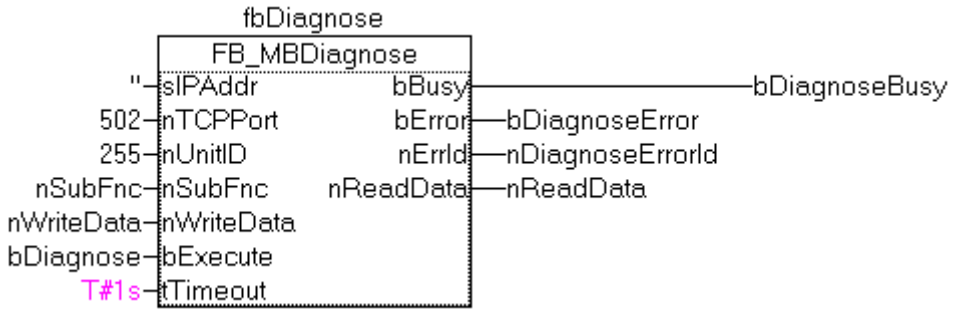
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

nReadData: Liefert das gelesene Datenwort.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Beispiel für den Aufruf des Bausteins in FBD:

```
PROGRAM Test
VAR
    fbDiagnose          : FB_MBDiagnose;
    bDiagnose           : BOOL;
    bDiagnoseBusy       : BOOL;
    bDiagnoseError      : BOOL;
    nDiagnoseErrorId    : UDINT;
    nSubFnc             : WORD;
    nReadData           : WORD;
    nWriteData          : WORD;
END_VAR
```



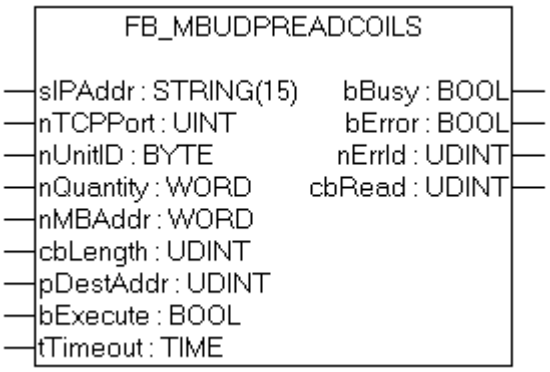
Nach steigender Flanke von "bExecute" und erfolgreicher Ausführung des Diangose-Befehls, befindet sich in nReadData das gelesene Datenwort.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11 UDP

7.1.11.1 FB_MBUdpReadCoils (Modbus-Funktion 1)



Diese Funktion wird zum Lesen von 1 bis 2048 digitaler Ausgänge (Coils) benutzt. Ein digitaler Ausgang entspricht einem Bit der gelesenen Datenbytes.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden digitalen Eingänge (Datenbits). Der Wert Null ist unzulässig.

nMBAddr: Startadresse der zu lesenden digitalen Eingänge (Bitoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer muss mindestens die Bytegröße: *(nQuantity + 7) / 8* besitzen.

pDestAdd: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

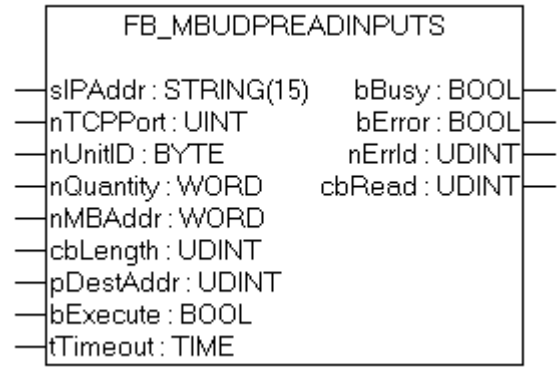
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.2 FB_MBUdpReadInputs(Modbus-Funktion 2)



Diese Funktion wird zum Lesen von 1 bis 2048 digitalen Eingängen benutzt. Ein digitaler Eingang entspricht einem Bit der gelesenen Datenbytes.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden digitalen Eingänge (Datenbits). Der Wert Null ist nicht zulässig.

nMBAddr: Startadresse der zu lesenden digitalen Eingänge (Bitoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers für die zu lesenden Datenbytes. Der Puffer muss mindestens die Bytegröße: *(nQuantity + 7) / 8* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

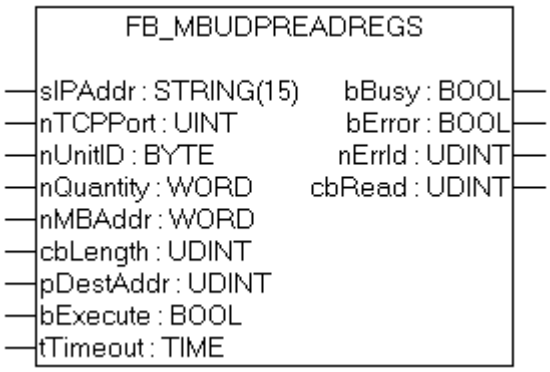
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.3 FB_MBUdpReadRegs (Modbus-Funktion 3)



Diese Funktion wird zum Lesen von 1 bis 128 Ausgangs-Register (16 Bit) benutzt. Das erste Byte enthält die unteren acht Bits und das zweite Byte die oberen acht Bits.

VAR_INPUT

```
VAR_INPUT
    sIPAddr          : STRING(15);
    nTCPPort         : UINT:= MODBUS_TCP_PORT;
    nUnitID          : BYTE:=16#FF;
    nQuantity        : WORD;
    nMBAddr          : WORD;
    cbLength         : UDINT;
    pDestAddr        : UDINT;
    bExecute         : BOOL;
    tTimeout         : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden Ausgangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBAddr: Startadresse der zu lesenden Ausgangs-Register (Wortoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers für die zu lesenden Registerwerte. Der Puffer muss mindestens die Bytegröße: *nQuantity * 2* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY        : BOOL;
    bError       : BOOL;
    nErrId       : UDINT;
    cbRead       : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

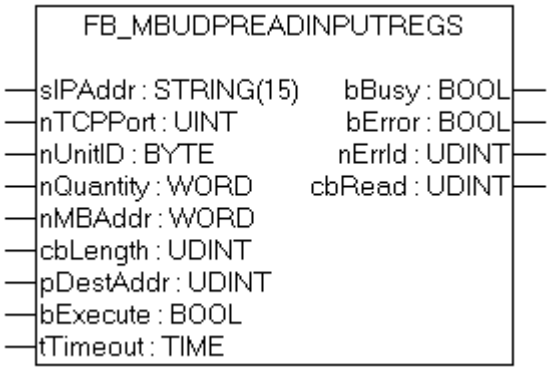
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.4 FB_MBUdpReadInputRegs(Modbus-Funktion 4)



Diese Funktion wird zum Lesen von 1 bis 128 Eingangs-Register (16Bit) benutzt. Endian

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pDestAddr       : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu lesenden Eingangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBAddr: Startadresse der zu lesenden Eingangs-Register (Wortoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Zielpuffers. Der Puffer muss mindestens die Bytegröße: *nQuantity * 2* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

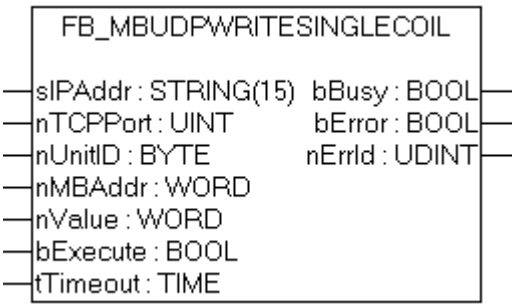
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.5 FB_MBUdpWriteSingleCoil (Modbus-Funktion 5)



Diese Funktion wird zum Beschreiben eines digitalen Ausgangs benutzt (Coil). Dabei handelt es sich um einen Bit-Zugriff.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
```

```
    nMBAddr         : WORD;
    nValue          : WORD;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nMBAddr: Adresse des digitalen Ausgangs (Bitoffset).

nValue: Wert, der in den digitalen Ausgang geschrieben werden soll. Der Wert 16#FF00 schaltet den Ausgang ein und der Wert 16#0000 schaltet den Ausgang ab.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

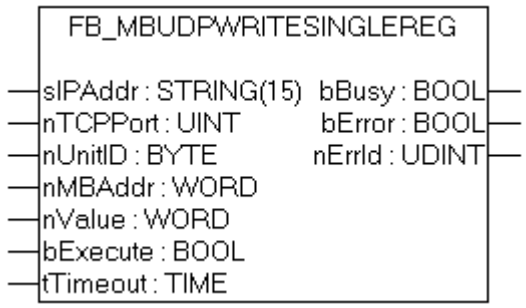
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.6 FB_MBUdpWriteSingleReg(Modbus-Funktion 6)



Diese Funktion wird zum Beschreiben eines einzelnen Ausgangsregisters benutzt. Dabei handelt es sich um einen 16 Bit-Zugriff.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nMBAddr         : WORD;
    nValue          : WORD;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nMBAddr: Adresse des Ausgangs-Registers (Wortoffset).

nValue: Wert, der in das Register geschrieben werden soll (Datenwort).

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

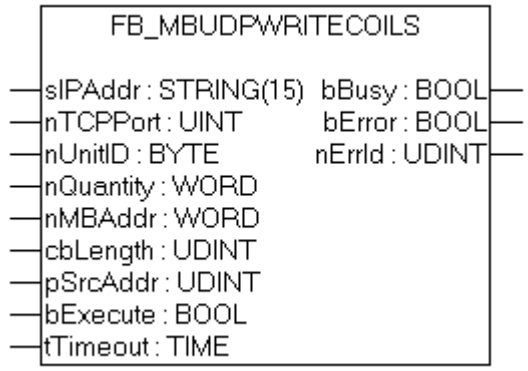
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.7 FB_MBUdpWriteCoils (Modbus-Funktion15)



Diese Funktion wird zum Beschreiben von 1 bis 2048 digitaler Ausgänge (Coils) benutzt. Ein digitaler Ausgang entspricht einem Bit der geschriebenen Datenbytes.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pSrcAddr        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der digitalen Ausgänge, die beschrieben werden sollen (Datenbits). Der Wert Null ist unzulässig.

nMBAddr: Startadresse der digitalen Ausgänge, die beschrieben werden sollen (Bitoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer muss mindestens die Bytegröße: *(nQuantity + 7) / 8* besitzen.

pSrcAddr: Enthält die Adresse des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

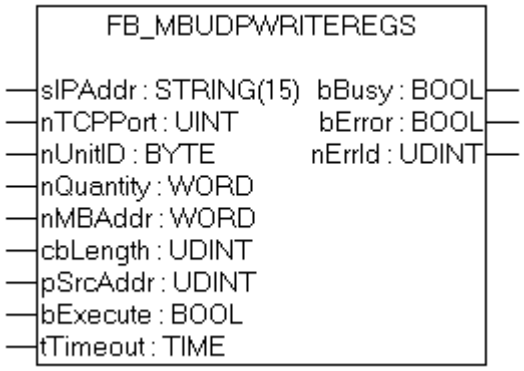
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.8 FB_MBUdpWriteRegs (Modbus-Funktion 16)



Diese Funktion wird zum Beschreiben von 1 bis 128 Ausgangs-Register (16 Bit) benutzt.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nQuantity       : WORD;
    nMBAddr         : WORD;
    cbLength        : UDINT;
    pSrcAddr        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nQuantity: Anzahl der zu schreibenden Ausgangs-Register (Datenworte).

nMBAddr: Startadresse der zu schreibenden Ausgangs-Register (Wortoffset).

cbLength: Enthält die max. verfügbare Bytegröße des Quellpuffers der die zu schreibende Registerwerte enthält. Der Puffer muss mindestens die Bytegröße: *nQuantity * 2* besitzen.

pSrcAddr: Enthält die Adresse des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

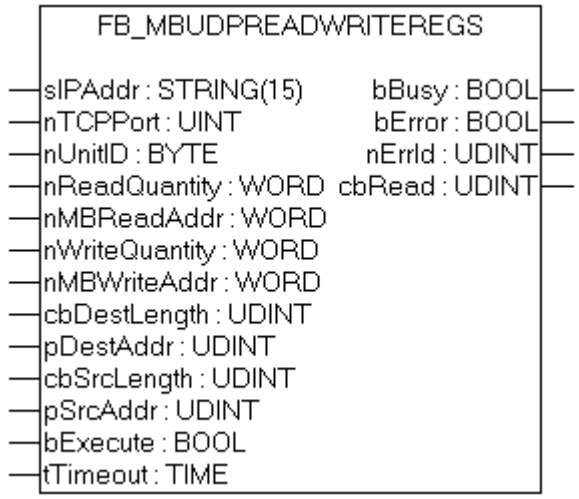
nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.9 FB_MBUdpReadWriteRegs (Modbus-Funktion 23)



Diese Funktion liest zuerst 1 bis 128 Ausgangs-Register (16 bit) und beschreibt danach 1 bis 128 Ausgangs-Register (16 Bit).

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nReadQuantity   : WORD;
    nMBReadAddr     : WORD;
    nWriteQuantity  : WORD;
    nMBWriteAddr    : WORD;
    cbDestLength    : UDINT;
    pDestAddr       : UDINT;
    cbSrcLength     : UDINT;
    pSrcAddr        : UDINT;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nReadQuantity: Anzahl der zu lesenden Ausgangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBReadAddr: Startadresse der zu lesenden Ausgangs-Register (Wortoffset).

nWriteQuantity: Anzahl der zu schreibenden Ausgangs-Register (Datenworte). Der Wert Null ist nicht zulässig.

nMBWriteAddr: Startadresse der zu schreibenden Ausgangs-Register (Wortoffset).

cbDestLength: Enthält die max. verfügbare Bytegröße des Zielpuffers für die zu lesenden Registerwerte. Der Puffer muss mindestens die Bytegröße: *nReadQuantity * 2* besitzen.

pDestAddr: Enthält die Adresse des Zielpuffers, in den die Daten gelesen werden sollen. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

cbSrcLength: Enthält die max. verfügbare Bytegröße des Quellpuffers der die zu schreibende Registerwerte enthält. Der Puffer muss mindestens die Bytegröße: *nWriteQuantity* 2* besitzen.

pSrcAddr: Enthält die Adresse des Quellpuffers, der die zu schreibenden Daten enthält. Der Puffer kann eine Einzelvariable, ein Array oder eine Struktur sein, dessen Adresse mit dem ADR - Operator ermittelt werden kann.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBUSY       : BOOL;
    bError      : BOOL;
    nErrId      : UDINT;
    cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

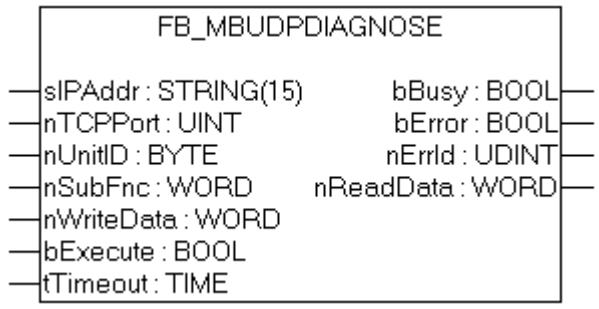
cbRead: Enthält die Anzahl der aktuell gelesenen Bytes.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

7.1.11.10 FB_MBUdpDiagnose (Modbus-Funktion 8)



Die Diagnose-Funktion stellt eine Reihe von Tests für die Überprüfung des Übertragungssystems zwischen dem Master und dem Slave, oder für die Überprüfung der verschiedenen internen Fehlerzustände innerhalb des Slaves, zur Verfügung.

VAR_INPUT

```
VAR_INPUT
    sIPAddr         : STRING(15);
    nTCPPort        : UINT:= MODBUS_TCP_PORT;
    nUnitID         : BYTE:=16#FF;
    nSubFnc         : WORD;
    nWriteData      : WORD;
    bExecute        : BOOL;
    tTimeout        : TIME;
END_VAR
```

sIPAddr: Ist ein String, der die IP-Adresse des Zielgerätes enthält.

nTCPPort: Portnummer des Zielgerätes.

nUnitID: Identifizierungsnummer eines Gerätes eines seriellen Sub-Netzwerkes. Wenn ein Gerät direkt über TCP/IP angesprochen wird, muss dieser Wert 16#FF entsprechen.

nSubFnc: Die Subfunktion, die ausgeführt werden soll.

nWriteData: Das Datenwort, das geschrieben werden soll.

bExecute: Durch eine steigende Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
```

```
    nErrId      : UDINT;
    nReadData   : WORD;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

nReadData: Liefert das gelesene Datenwort.

Function specific ADS error code	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS Bibliotheken
TwinCAT v2.8.0	PC (i386)	TcModbusSrv.Lib

8 Beispiele

8.1 Beispiel: Digitaler IO Zugriff (Lauflicht)

Dieses Beispiel verdeutlicht, wie Sie per Modbus auf die Echtzeitumgebung eines TwinCAT Systems zugreifen können.

Das Standard-Mapping [▶ 14] des ModbusServers mappt die digitalen Ausgänge (Coils) auf das Prozessabbild der physikalischen Ausg nge der SPS.

```
PROGRAM MAIN
VAR
    Q00 AT%QX0.0            : BOOL;
    Q01 AT%QX0.1            : BOOL;
    Q02 AT%QX0.2            : BOOL;
    Q03 AT%QX0.3            : BOOL;
    Q04 AT%QX0.4            : BOOL;
    Q05 AT%QX0.5            : BOOL;
    Q06 AT%QX0.6            : BOOL;
    Q07 AT%QX0.7            : BOOL;

    fbWriteCoils            : FB_MBWriteCoils;
    bWrite                  : BOOL;
    nValue                  : INT;
END_VAR
```

```
IF NOT bWrite THEN
    nValue := nValue + 1;
    bWrite := TRUE;
    fbWriteCoils.nQuantity := 8;
    fbWriteCoils.cbLength := SIZEOF(nValue);
    fbWriteCoils.pSrcAddr := ADR(nValue);
    fbWriteCoils.tTimeout := T#5s;
    fbWriteCoils(bExecute:=TRUE);
ELSE
    IF NOT fbWriteCoils.bBUSY THEN
        bWrite :=FALSE;
    END_IF
    fbWriteCoils(bExecute:=FALSE);
END_IF
```

Nach steigender Flanke an bWrite, wird das Lauflicht in den Bereich der physikalischen Ausgänge der SPS geschrieben (Q00-Q07).

Die Bitorder wird wie folgt gesetzt:

Bit	8 MSB	7	6	5	4	3	2	1 LSB
Output	7	6	5	4	3	2	1	0

MSB = Most significant bit (Höchstwertiges Bit)

LSB = Least significant bit (Niederwertiges Bit)

https://infosys.beckhoff.com/content/1031/tcmodbussrv/Resources/11379454731.zip

8.2 Beispiel: Schreiben mehrerer Register

Dieses Beispiel verdeutlicht, wie Sie per Modbus auf die Echtzeitumgebung eines TwinCAT Systems zugreifen können.

Die Modbusadresse **0x3000** zeigt bei einem Standard-Mapping [▶ 14] des ModbusServers auf den Merkerbereich der SPS (ADS-Indexgruppe 0x4020).

Nachdem Sie bWriteRegs aufgerufen haben, wird das Array **arrValue** in den Merkerbereich und somit in die Variable M0 geschrieben.

```
PROGRAM MAIN
VAR
   ipAddr          : STRING(15) := '';
```

```
   M0 AT%MB0      : ARRAY [0..3] OF WORD;
   nValue         : ARRAY [0..3] OF WORD := 0,10,100,1000;
   fbWriteRegs    : FB_MBWriteRegs;
   bWriteRegs     : BOOL;
END_VAR
```

```
IF NOT bWriteRegs THEN
    nValue[0]:= nValue[0]+1;
    nValue[1]:= nValue[1]+1;
    nValue[2]:= nValue[2]+1;
    nValue[3]:= nValue[3]+1;
    bWriteRegs :=TRUE;
    fbWriteRegs.sIPAddr :=ipAddr;
    fbWriteRegs.nQuantity := 4;
    fbWriteRegs.nMBAddr := 16#3000;
    fbWriteRegs.cbLength := SIZEOF(nValue);
    fbWriteRegs.pSrcAddr := ADR(nValue);
    fbWriteRegs.tTimeout := T#5s;
    fbWriteRegs(bExecute:=TRUE);
ELSE
    IF NOT fbWriteRegs.bBUSY THEN
        bWriteRegs :=FALSE;
    END_IF
    fbWriteRegs(bExecute:=FALSE);
END_IF
```

https://infosys.beckhoff.com/content/1031/tcmodbussrv/Resources/11379456139.zip

9 Return Codes

Hex	Dezimal	Quelle
0x00000000-0x00007800	0-30720	TwinCAT System Fehlercodes
0x00008000-0x0080FF	32768-33023	Interne TwinCAT Modbus TCP
0x80070000-0x8007FFFF	2147942400-2148007935	Fehlerquelle = Fehlercode - 0x80070000 = Win32 System Fehlercode

TwinCAT Modbus TCP return code

Hex	Possible reason
0x8001	Nicht implementierte Modbus-Funktion
0x8002	Ungültige Adresse oder Länge
0x8003	Ungültige Parameter: - falsche Registeranzahl
0x8004	Fehler im Modbus-Server

Mehr Informationen:
www.beckhoff.de/ts6250

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.com
www.beckhoff.com