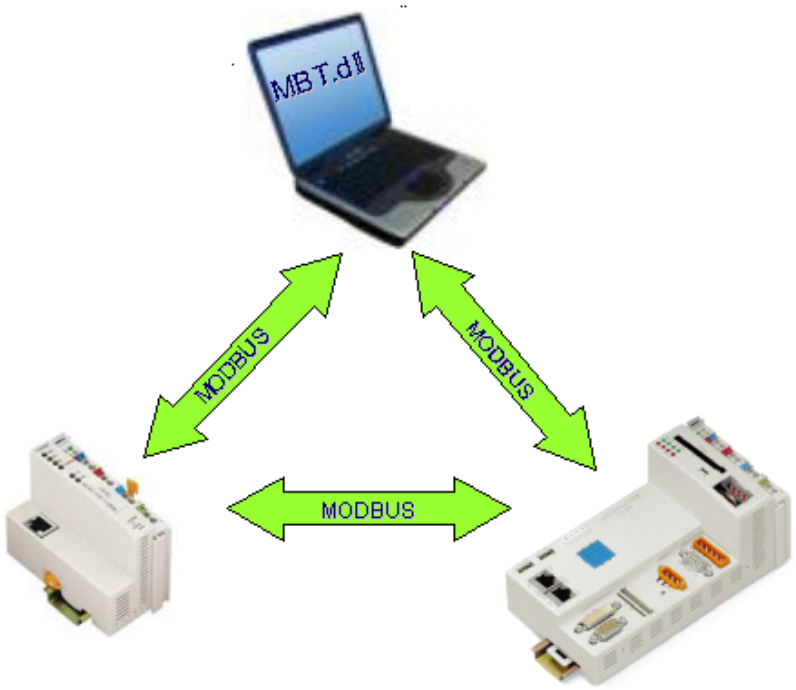
WAGO I/O SYSTEM 750

Modbuskommunikation zwischen WAGO Ethernet Kopplern und Controllern



Anwendungshinweis

A300003, Deutsch
Version 1.2.1

WAGO®
INNOVATIVE CONNECTIONS

WAGO Kontakttechnik GmbH & Co. KG
Hansastraße 27
D-32423 Minden

Tel.: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: http://www.wago.com

Technischer Support
Tel.: +49 (0) 571/8 87 – 5 55
Fax: +49 (0) 571/8 87 – 85 55

E-Mail: support@wago.com

Es wurden alle erdenklichen Maßnahmen getroffen, um die Richtigkeit und Vollständigkeit der vorliegenden Dokumentation zu gewährleisten. Da sich Fehler, trotz aller Sorgfalt, nie vollständig vermeiden lassen, sind wir für Hinweise und Anregungen jederzeit dankbar.

Wir weisen darauf hin, dass die im Dokument verwendeten Soft- und Hardwarebezeichnungen und Markennamen der jeweiligen Firmen im Allgemeinen einem Warenzeichenschutz, Markenschutz oder patentrechtlichem Schutz unterliegen.

INHALTSVERZEICHNIS

1 Wichtige Erläuterungen

Um dem Anwender eine schnelle Installation und Inbetriebnahme der beschriebenen Geräte zu gewährleisten, ist es notwendig, die nachfolgenden Hinweise und Erläuterungen sorgfältig zu lesen und zu beachten.

1.1 Rechtliche Grundlagen

1.1.1 Urheberschutz

Dieses Dokument, einschließlich aller darin befindlichen Abbildungen, ist urheberrechtlich geschützt. Jede Weiterverwendung dieses Dokumentes, die von den urheberrechtlichen Bestimmungen abweicht, ist nicht gestattet.
Die Reproduktion, Übersetzung in andere Sprachen, sowie die elektronische und fototechnische Archivierung und Veränderung bedarf der schriftlichen Genehmigung der WAGO Kontakttechnik GmbH & Co. KG, Minden. Zuwiderhandlungen ziehen einen Schadenersatzanspruch nach sich.

Die WAGO Kontakttechnik GmbH & Co. KG behält sich Änderungen, die dem technischen Fortschritt dienen, vor.
Alle Rechte für den Fall der Patenterteilung oder des Gebrauchmusterschutzes sind der WAGO Kontakttechnik GmbH & Co. KG vorbehalten. Fremdprodukte werden stets ohne Vermerk auf Patentrechte genannt. Die Existenz solcher Rechte ist daher nicht auszuschließen.

1.1.2 Personalqualifikation

Der in diesem Dokument beschriebene Produktgebrauch richtet sich ausschließlich an Fachkräfte mit einer Ausbildung in der SPS-Programmierung, Elektrofachkräfte oder von Elektrofachkräften unterwiesene Personen, die außerdem mit den geltenden Normen vertraut sind. Für Fehlhandlungen und Schäden, die an WAGO-Produkten und Fremdprodukten durch Missachtung der Informationen dieses Dokumentes entstehen, übernimmt die WAGO Kontakttechnik GmbH & Co. KG keine Haftung.

1.1.3 Bestimmungsgemäßer Gebrauch

Die Komponenten werden ab Werk für den jeweiligen Anwendungsfall mit einer festen Hard- und Softwarekonfiguration ausgeliefert. Änderungen sind nur im Rahmen der in dem Dokument aufgezeigten Möglichkeiten zulässig. Alle anderen Veränderungen an der Hard- oder Software, sowie der nicht bestimmungsgemäße Gebrauch der Komponenten, bewirken den Haftungsausschluss der WAGO Kontakttechnik GmbH.

Wünsche an eine abgewandelte bzw. neue Hard- oder Softwarekonfiguration richten Sie bitte an WAGO Kontakttechnik GmbH & Co. KG.

1.2 Gültigkeitsbereich

Dieser Anwendungshinweis basiert auf die genannte Hard- und Software der jeweiligen Hersteller sowie auf die zugehörige Dokumentation. Daher gilt dieser Anwendungshinweis nur für die beschriebene Installation.
Neue Hard- und Softwareversionen erfordern eventuell eine geänderte Handhabung.

Beachten Sie die ausführliche Beschreibung in den jeweiligen Handbüchern.

1.3 Symbole

Gefahr
Informationen unbedingt beachten, um Personen vor Schaden zu bewahren.

Achtung
Informationen unbedingt beachten, um am Gerät Schäden zu verhindern.

Beachten
Randbedingungen, die für einen fehlerfreien Betrieb unbedingt zu beachten sind.

ESD (Electrostatic Discharge)
Warnung vor Gefährdung der Komponenten durch elektrostatische Entladung. Vorsichtsmaßnahme bei Handhabung elektrostatisch entladungsgefährdeter Bauelemente beachten.

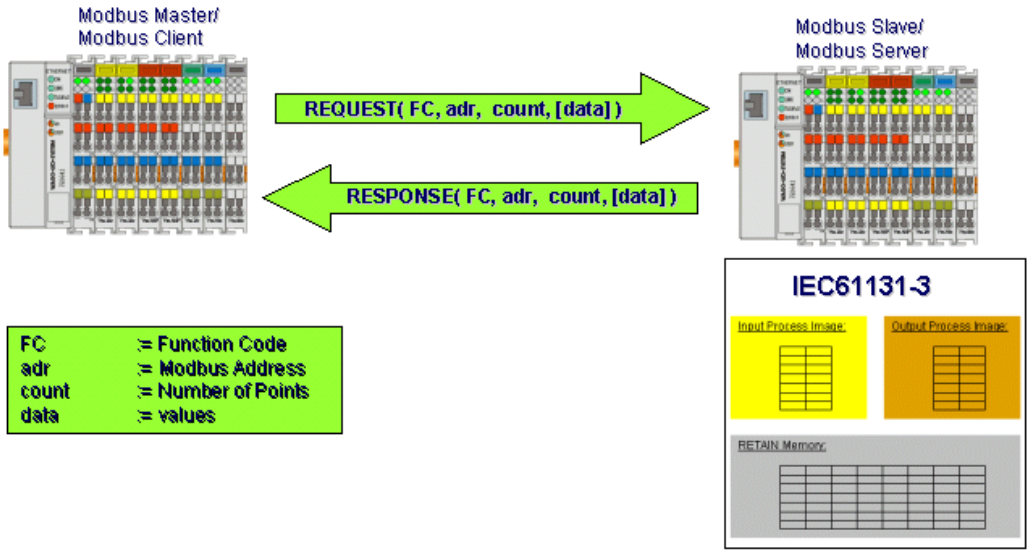
Hinweis
Routinen oder Ratschläge für den effizienten Geräteeinsatz und die Softwareoptimierung.

Weitere Informationen
Verweise auf zusätzliche Literatur, Handbücher, Datenblätter und INTERNET Seiten.

2 Beschreibung

Das modulare Konzept der WAGO I/O-Serie 750 ermöglicht es bis zu 255 I/O-Module an die Kopfstation anzuhängen. Dieser variable Knotenaufbau, sowie die große Anzahl verschiedener I/O-Module verhindert jedoch eine statische Zuordnung von I/O-Daten auf feste Modbus-Adressen.
Einzige Ausnahme sind die „digitalen“ Modbus-Dienste, bei ihnen ist die Modbus-Adresse identisch mit der Kanalnummer, das heißt den 47ten digitalen Eingang findet man immer an Modbus-Adresse „46“.

Durch das hinzufügen oder entfernen von I/O-Modulen verändert sich der Aufbau der Prozessabbilder und damit auch die Modbus-Adressen der einzelnen Kanäle der I/O-Module.
Dieser Anwendungshinweis beschreibt die Zusammenhänge zwischen Knotenaufbau, Prozessabbildern, IEC61131-Adressen und dem Modbus-Protokoll.



Mit den Modbus-Protokoll lassen sich beliebige Speicherplätze im Prozessabbild auslesen oder verändern, welches I/O-Module sich jedoch an einem bestimmten Speicherplatz befindet wird durch den Knotenaufbau bestimmt.

Nach jedem Einschalten der Versorgungsspannung ermittelt ein WAGO Ethernet Koppler oder Controller den aktuellen Knotenaufbau und erstellt daraus Prozessabbilder für Ein- und Ausgänge.

Das Regelwerk unterscheidet zwischen komplexen und digitalen I/O-Modulen.
Komplexe I/O-Module besitzen eine Datenbreite von mehr als einem Byte, zu ihnen gehören analoge Ein- und Ausgangsmodule, Zähler, Stepper, serielle Schnittstellen usw. - eben alles nicht digitale.
Bei der Erstellung der Prozessabbilder werden in einem im ersten Durchlauf die komplexen Klemmen entsprechend ihrer physikalischen Reihenfolge hinter der Kopfstation im Prozessabbild angeordnet.

In einem zweitem Durchlauf werden dann die Daten der digitalen I/O-Module entsprechend ihrer Position hinter der Kopfstation zu vollen Bytes gepackt und im Prozessabbild direkt hinter denen der „Komplexen“ angeordnet.

Alle WAGO Ethernet Koppler und Controller besitzen genau ein Prozessabbild für physikalische Eingänge und eines für physikalische Ausgänge.
In dem jeweiligen Prozessabbild werden die Daten der komplexen I/O-Module direkt gefolgt von den Daten der digitalen I/O-Modulen abgelegt.

Für WAGO-Ethernet-Koppler ist die Regelung der „Schreibberechtigung“ auf Prozessabbilder denkbar einfach:
- Physikalische Eingänge können nur gelesen werden.
- Physikalische Ausgänge können geschrieben und gelesen werden.

Die WAGO-Ethernet-Controller verfügen zusätzlich über einen remanenten Merkerbereich von typisch 8kB sowie die PFC-IN und PFC-OUT Bereiche mit einer Größe von jeweils 256 Worten (512 Byte).
Der Merkerbereich kann sowohl über das Modbus-Protokoll als auch durch das SPS-Programm gelesen wie beschrieben werden.
Hauptanwendungsgebiet des PFC-IN und PFC-OUT Bereiches ist die Realisierung von Schnittstellen zu anderen Steuerungen über das Modbus-Protokoll.
Der PFC-IN Bereich läst sich nur von „außen“, d.h. über das Modbus-Protokoll beschreiben. Aus Sicht der SPS handelt es sich bei dem PFC-IN Bereich um lokale Eingänge die nur gelesen werden können.
Der PFC-OUT Bereich gleicht physikalischen Ausgängen, er läst sich nur aus dem SPS-Programm heraus beschreiben.

Die Beschreibung des Aufbaus der Prozessabbilder erfolgt mit den Sprachelementen der IEC61131-3. Dies ist notwendig, da das Modbus-Protokoll lediglich Dienste auf Grunddatentypen definiert und nicht deren Bedeutung bzw. konkrete Adressen regelt.

Die Modbus-Kommunikation erfolgt mit Hilfe von Dienstaufrufen, dazu sendet der Modbus-Master(Client) ein Request-Telegramm an Port 502 des Modbus-Slave(Server). Der Modbus-Slave liefert das Ergebnis des Dienstaufrufes in einem Response-Telegramm an den Modbus-Master zurück.

Die wesentlichsten Elemente eines Modbus-Telegrammes sind:

Item	Description
FunctionCode (FC)	Dienstkennung: Lese- oder Schreib-Operation auf Bit’s oder WORD’s
Address	Startadresse der Operation
Count	Dienstabhängig die Anzahl Bit’s oder WORD’s(Bytes)
[Data]	Prozessdaten

Die Dienstkennung bzw. der „FunctionCode(FC)“ bestimmt zunächst ob es sich um eine Lese- oder Schreib-Operation handelt, zusätzlich bestimmt sie den Grunddatentyp auf den die Operation angewendet werden soll. Damit ist

auch die Bedeutung der Parameter „Address“ und „Count“ abhängig vom Funktionscode. So kann „address :=3“ für das vierte Bit oder Word im Ein- oder Ausgangsprozessabbild stehen.

Das Modbus-Protokoll basiert im wesentlichen auf den folgenden Grunddatentypen:

Datatype	Length	Description
Discrete Inputs	1 Bit	Digitale Eingänge
Coils	1 Bit	Digitale Ausgänge
Input Register	16 Bit	Analoge-Eingangsdaten
Holding Register	16 Bit	Analoge-Ausgangsdaten

Für jeden Grunddatentyp sind ein oder mehr „FunctionCodes“ definiert.

FC	Name	Description
FC1	Read coils	Rücklesen mehrerer digitaler Ausgänge
FC2	Read inputs discrete	Lesen mehrerer digitaler Eingänge
FC3	Read holding registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC4	Read input registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC11	Get comm event counter	Kommunikationsereigniszähler
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analoger Ausgänge
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

Obwohl digitale und analoge Prozessdaten der WAGO Ethernet Koppler und Controllern in einem Prozessabbild zusammengefasst sind, erreichen Sie mit den „digitalen“ Modbus-Diensten an Adresse 0 immer den ersten digitalen Ausgang bzw. Eingang . Das heißt die „digitalen“ Modbus-Diensten ignorieren die komplexen Klemmen.
Auf der anderen Seite, läst sich jedoch der Zustand der digitalen Ein- und Ausgänge auch über die sogenannten „Register“-Dienste ermitteln bzw. verändern.

Alle WAGO Ethernet Koppler und Controller unterscheiden nicht zwischen den Functioncodes FC1 und FC2. Beide Modbus-Dienste verwenden die gleiche Implementierung und können genutzt werden um auf digitale Ein-, Aus- und Merker-Daten zuzugreifen. Verwenden Sie einen Offset von 512 bzw. 0x200 um den Zustand des ersten digitalen Ausganges unter der selben Stelle zu lesen wie zu schreiben.

Alle WAGO Ethernet Koppler und Controller unterscheiden nicht zwischen den Functioncodes FC3 und FC4. Beide Modbus-Dienste verwenden die gleiche Implementierung und können genutzt werden um Eingangsdaten, Ausgangsdaten oder Merkerzustände zu lesen. Verwenden Sie einen Offset von 512 bzw. 0x200 um den Zustand des ersten analogen Ausgangs an der selben Adresse zu lesen wie zu schreiben.

Verschiedene SCADA-Programme sind darauf angewiesen den Zustand von physikalischen Ausgängen unter der selben Modbus-Adresse zu lesen wie zu

schreiben. Um auch diese Gruppe von Programmen zu unterstützen werden die Zustände der physikalischen Ausgänge unter zwei verschiedenen Modbus-Adressen zur Verfügung gestellt.

Stellvertretend folgt hier die Adresszuordnung für die Modbus-Dienste FC3 und FC4 auf dem 750-841. Im Anhang finden Sie weitere gerätebezogene Gegenüberstellungen von Modbus-Adressen zu IEC-Adressen.

750-841: Modbus- vs IEC61131-Addresses for FC3 and FC4			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%IW0 ... %IW255	Physical Input Area
256 ... 511	0x0100 ... 0x01FF	%QW256 ... %QW511	PFC-OUT-Variables
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-IN-Variables
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW255	Retain Memory Area
24576 ... 25340	0x6000 ... 0x62FC	%IW512 ... %IW1275	Physical Input Area
25341 ... 28671	0x62FD ... 0x6FFF	-	Modbus Exception: “Illegal data address”
28672 ... 29346	0x7000 ... 0x72FC	%QW512 ... %QW1275	Physical Output Area

Die Tabelle zeigt unter welchen Modbus-Adressen die verschiedenen Speicherbereiche des 750-841 mit den Modbus-Diensten FC3 oder FC4 ausgelesen werden können.

Soll zum Beispiel der aktuelle Prozesswert einer analogen Ausgangsklemme zurück gelesen werden, so ist zunächst der Modbus-Dienst auszuwählen, hier FC3 oder FC4.
Im nächsten Schritt ist die IEC-Adresse des betreffenden Kanals im Prozessabbild der Ausgänge zu bestimmen. In diesem Beispiel soll das die IEC61131-Adresse „%QW47“ sein.
Aus der dritten Tabellenzeile erhalten wir einen Offset von 512 bzw. 0x200 und so ergibt sich 559 bzw. 0x022F als Modbus-Adresse um „%QW47“ zu lesen an dem sich der Prozesswert einer analogen Ausgangsklemme befindet.

Zusätzlich zeigt die Tabelle, das es Lücken im Modbus-Adressraum gibt. Die Verwendung dieser Adressen führt zu einer „Illegal data address exception“.

Die Modbus-Konfigurationsregister haben keine Entsprechung im IEC-Adressraum, ist es notwendig die Konfiguration aus dem SPS-Programm heraus zu lesen oder ändern, so bleibt nur der Weg Modbus-Telegramme an die eigene IP-Adresse zu senden.

Letztlich gilt es noch einige Besonderheiten zu beachten.

So wurde in CoDeSys 2.3 das „Monitoring“ von Variablen optimiert.
Dies hat zur Folge, das deklarierte aber im Programmcode unbenutzte Variablen nicht „gemonitored“ werden, und somit im Debugger immer mit den Wert „0“ angezeigt werden.

Die WAGO Kontakttechnik GmbH & Co. KG erweitert die Modbus Familie um Modbus-UDP. Diese Variante verwendet eine verbindungslose asynchrone Client Server Kommunikation über Ethernet.
Modbus-UDP löst ein Problem, das entsteht, wenn ein Modbus-Slave(Server) nicht verfügbar ist(zB. Versorgungsspannung getrennt). Bei Modbus-TCP greifen in diesem Fall die Retransmissionsmechanismen des TCP-Stacks, die dazu führen, das der Modbus-Master(Client) erst sehr spät feststellt, das die Gegenstelle nicht verfügbar ist.
Bei Modbus-UDP erfolgt die TimeOut-Überwachung auf Applikationsebene(OSI-Schicht 7) und kann dadurch umgehend auf ein fehlendes Response-Telegramm reagieren.
Aus diesem Grund empfehlen wir, wenn möglich, den Einsatz der Modbus-UDP-Variante.

Mit den Funktionsbausteinen „SET_DIGITAL_INPUT_OFFSET“ und „SET_DIGITAL_OUTPUT_OFFSET“ aus der Bibliothek „mod_com.lib“ können die Startadressen der ersten digitalen I/O-Module an WAGO Ethernet Controllern fest vorgegeben werden. Dies erlaubt Platz für spätere Erweiterungen einzuplanen. Eine Beschreibung des speziellen Verhaltens auf den einzelnen Geräten finden Sie im Anhang.

Eine Besonderheit des 750-841 ist es, das jedem I/O-Modul eine Schreibberechtigung zugeordnet werden kann bzw. muss.
Die Schreibberechtigung kann entweder dem SPS-Programm, dem Modbus-Protokoll oder dem Ethernet_IP-Protokoll zugeordnet werden.
Gespeichert wird die Zuordnung der Schreibberechtigung in der Datei „/etc/EA-config.xml“. Fehlt die „EA-config“ Datei oder weicht die Anzahl der konfigurierten I/O-Module von der tatsächlich angeschlossen Anzahl ab, werden alle I/O-Module dem Modbus-Protokoll zugeordnet.

Ein hilfreiches Werkzeug bei der Inbetriebnahme oder Fehlersuche im Ethernet ist das Open-Source-Project „Ethereal“ bzw. „WireShark“.
Es handelt sich dabei um einen frei verfügbaren Netzwerksniffer. Das Werkzeug stellt nach der Aufzeichnung des Datenverkehrs einer Netzwerk-Schnittstelle die Daten übersichtlich in Form einzelner Pakete dar.

2.1 Die IEC61131-3 Adressen

Die nachfolgende Tabelle zeigt den Aufbau einer Hardwareadresse in der IEC61131.

Hardware address				Description
%				Preliminary character
	I Q M			Input Output Merker
		X B W D		Bit BYTE WORD (16Bit) DWORD(32Bit)
			x.y	x-Word address; y-Bit address
Examples				
%IX1.7				Eighth Bit in second word
%IW0				Input word 0
%QB47				Output byte 47
%QD2				Output double word 2
%MX3.14				Bit 14 in merker word 3
%MW3				Merker word 3

Für den Zugriff auf Peripherie sowie spezielle Datenbereiche des SPS-Systems definiert die IEC zwei Arten von Variablen „Direkt dargestellte Variable“ und „Symbolische Variable“.
Bei „Direkt dargestellte Variablen“ tauchen Hardwareadressen im Anweisungsteil des SPS-Programmes auf. Da dies die Wartbarkeit erschwert sollte dieser Typ nur für kleine bis mittlere Projekte angewendet werden.
Empfohlen wird die Verwendung von „Symbolischen Variablen“, sie tauchen im Deklarationsteil des SPS-Programmes auf und die Zuweisung der Hardware-Adresse wird mit dem Schlüsselwort „AT“ eingeleitet.

```
VAR
   xMyOutput  AT %QX0.2 : BOOL;    (* A digital output *)
   wMyInput   AT %IW1   : WORD;    (* A analog input *)
   oInterface AT %MW0   : TMyType;(* A userdefined type  *)
VAR_END
```

Auf diese Weise lassen sich beliebige typisierte Variablen im Speicher positionieren.

```
TYPE TMyType :
 STRUCT
   wState      : WORD;     (* actual state *)
   dwJobAct    : DWORD;    (* Actual job *)
   dwJobLast   : DWORD;    (* Last job *)
   dwJobNext   : DWORD;    (* Next job *)
   xFlagDoIt   : BOOL;     (* something should happen *) xFlag-
Done     : BOOL;     (* something have been done *)
 END_STRUCT
END_TYPE
```

Dieses Vorgehen kann bei der Realisierung von Software-Schnittstellen zwischen Leitsystem und Modbus-Feldgeräten vorteilhaft sein.

Innerhalb des Prozessabbildes kann jedes Bit mit den Sprachmitteln der IEC61131 adressiert werden.

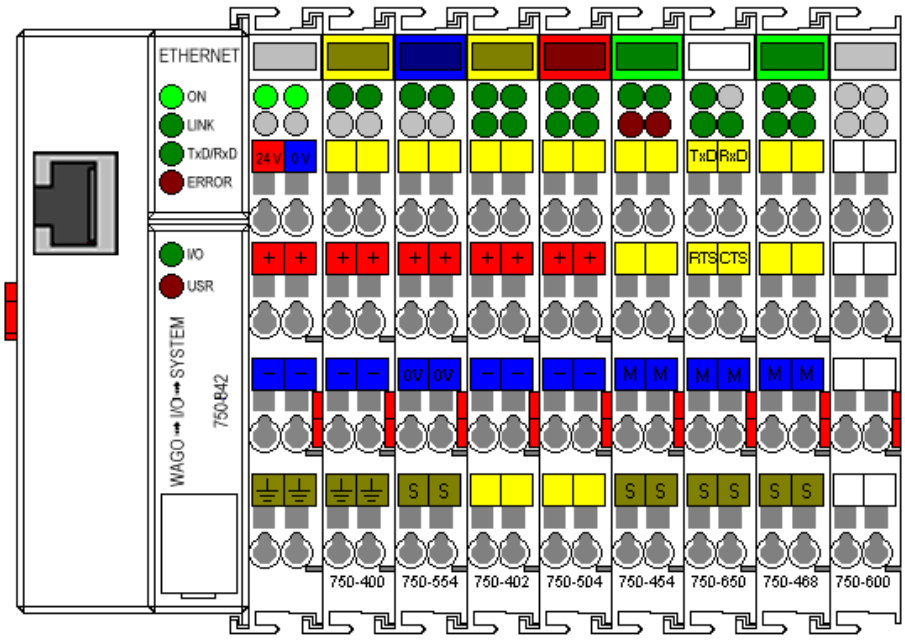
2.2 Aufbau der Prozessabbilder

Ein Prozessabbild ist ein Stück Speicher fester Größe in das die Prozesswerte der I/O-Module eingetragen werden.

Es wird genau ein Prozessabbild für Eingangsdaten und eines für Ausgangsdaten erzeugt.
Die Prozesswerte der einzelnen I/O-Module werden abhängig von Typ und Position hinter der Kopfstation in den entsprechenden Prozessabbild abgelegt.

Als Typen wird zwischen „digitalen“ und „komplexen“ I/O-Modulen unterschieden.

Komplexe I/O-Module (häufig auch als „analoge“ bezeichnet) stehen stellvertretend für alle Busklemmen, mit einer Datenbreite von mehr als einem Byte. Beispiel sind: analoge Ein- und Ausgänge, Zählerklemmen, I/O-Module für Winkel- und Wegmessung, Kommunikationsmodule wie RS232C usw. oder mit anderen Worten „Alle nicht digitalen I/O-Module“.



Für das Ein- und Ausgangsprozessabbild werden die Daten der I/O-Module in der Reihenfolge ihrer Position nach der Kopfstation in dem jeweiligen Prozessabbild abgelegt.

Dabei werden zuerst die „Komplexen“ und im Anschluss daran die „Digitalen“ im Prozessabbild abgelegt.

Die Bits der digitalen Klemmen werden zu Bytes zusammengefügt. Ist die Anzahl der digitalen E/As größer als 8 Bit, beginnt der Controller automatisch ein weiteres Byte.

Die Datenbreite eines I/O-Modules kann zwischen 0 und 46 Byte betragen. Details finden Sie im Handbuch zur Kopfstation sowie im Handbuch des betreffenden I/O-Modules.

Die nachfolgende Tabelle soll diesen Zusammenhang an einem konkretem Beispiel zeigen.

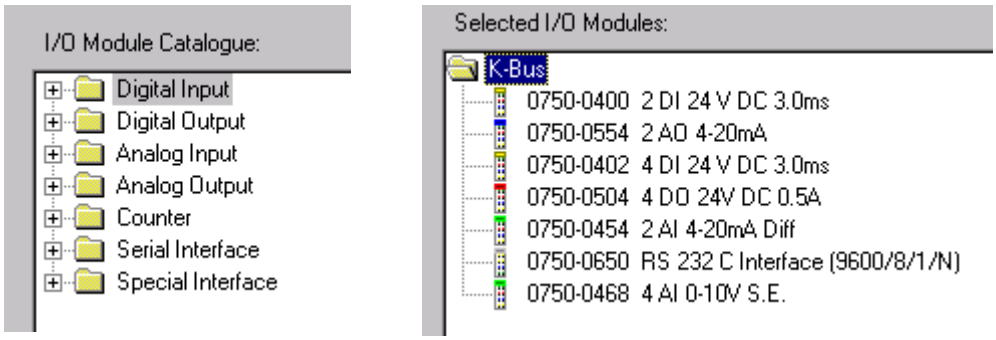
I/O-Module		Input image		Output image		Description
Type	C	run1	run2	run3	run4	
750-400	1		%IX8.0			**2 DI DC24V 3ms**: Erstes digitales Eingangs-Module mit einer Datenbreite von zwei Bit. Da die komplexen Eingangs-Module aus „run1“ bereits die ersten 8 Worte belegen, landen die digitalen Eingänge auf niederwertigsten Bits im Wort 8.
	2		%IX8.1			
750-554	1			%QW0		**2 AO 4-20mA:** Erstes analoges Ausgangs-Modul mit einer Datenbreite von zwei Worten. Diese Klemme belegt die ersten zwei Wörter im Ausgangsprozessabbild.
	2			%QW1		
750-402	1		%IX8.2			**4 DI DC24V**: Die vier digitalen Eingänge dieses I/O-Module werden hinter die zwei der 750-400 gepackt und landen im achten Wort des Eingangsprozessabbildes..
	2		%IX8.3			
	3		%IX8.4			
	4		%IX8.5			
750-504	1				%QX4.0	**4 DO DC24V**: 1. Klemme der digitalen Ausgänge. Die Klemmen der analogen Ausgänge belegen bereits die ersten 4 Wörter im Ausgangsprozessabbildes.
	2				%QX4.1	
	3				%QX4.2	
	4				%QX4.3	
750-454	1	%IW0				**2 AI 4-20mA:** 1. Klemme der analogen Eingänge. Diese Klemme belegt also die ersten zwei Wörter der Eingangstabelle.
	2	%IW1				
750-650	1	%IW2				**RS232 C 9600/8/N/1:** Das serielle Schnittstellenmodul 750-650 ist eine komplexes I/O-Modul das sowohl im Eingangsprozessabbild als auch im Ausgangsprozessabbild mit jeweils 4Byte vertreten ist.
		%IW3				
				%QW2		
				%QW3		
750-468	1	%IW4				**4 AI 0-10V S.E:** Die Klemme 750-468 folgt den 2 Eingangswörtern der 750-454 und den 2 von der 750-650 belegten Eingangswörtern. Die Klemme 750-467 belegt 4 Eingangswörter (4 Kanäle 0-10V).
	2	%IW5				
	3	%IW6				
	4	%IW7				
750-600						**End module** Die Klemme 750-600 ist eine passive Klemme.

C* : Channelcount

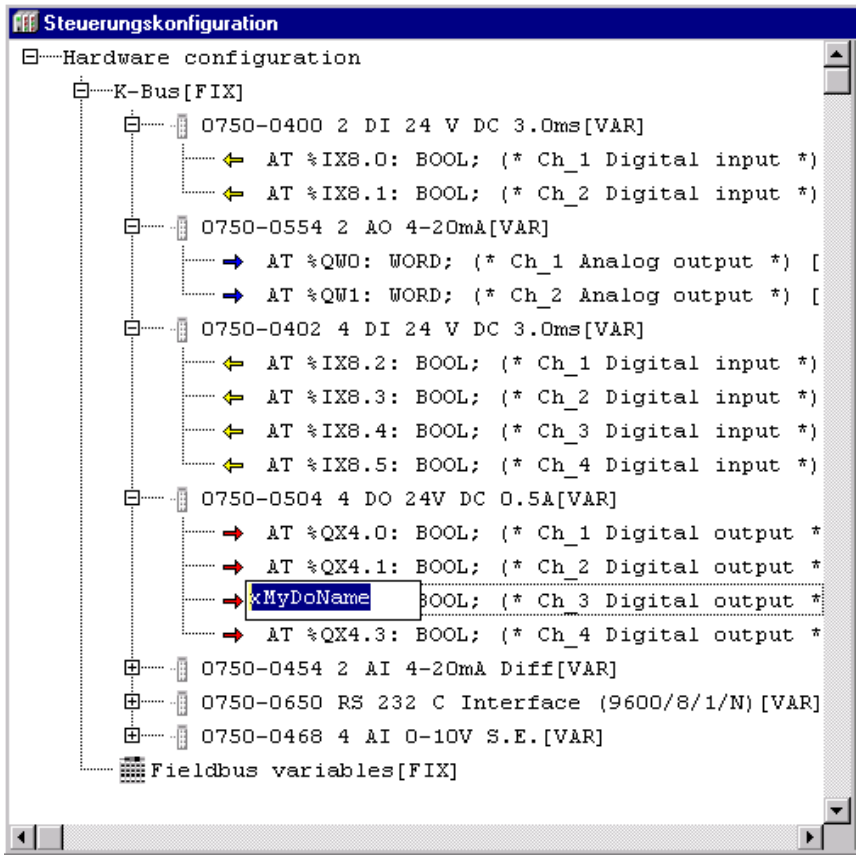
In der Tabelle bezeichnet „run1“ bis „run4“ die zeitliche Reihenfolge bei der Zusammenstellung der Prozessabbilder für Eingänge und Ausgänge in der Hochlaufphase eines WAGO Kopplers oder Controllers.

2.2.1 Der Hardwarekonfigurator

Für die programmierbaren Controller steht mit der Erstellung der „CoDeSys-Steuerungskonfiguration“ ein weiterer Weg zur Ermittlung der IEC-Adressen für Prozessdaten zur Verfügung.
Angelegt wird die Steuerungskonfiguration auf der Registerkarte „Ressourcen“. Bei der Erstellung wählt man die gesteckten I/O-Module aus einem Katalog aus.



Anschließend kann aus dem Hintergrundmenü der Hardwarekonfigurator angewiesen werden die IEC-Adressen erneut zu berechnen.



Zusätzlich erzeugt der Hardwarekonfigurator den größten Teil der Variablendeklaration, lediglich der Variablenname ist noch hinzuzufügen.

2.3 Das Modbus-Protokoll

Das bereits seit 1979 bekannte Modbus Protokoll ist mit Modbus-TCP heute ein offener Internet Draft Standard der IETF (Internet Engineering Task Force).
Die seit der Ursprungsvariante bewährten Modbus-Dienste und das Objektmodell wurden unverändert beibehalten und auf TCP/IP als Übertragungsmedium abgebildet. Kommuniziert wird über den „well known“ Port 502, der für Modbus reserviert ist.
Damit besteht die Modbus-Familie aus den klassischen Modbus-RTU und Modbus-ASCII (Asynchrone Übertragung über RS-232 oder RS-485) und Modbus-TCP (Verbindungsorientierte Client-Server Kommunikation über Ethernet).

Die WAGO Kontakttechnik GmbH & Co. KG erweitert die Modbus Familie um Modbus-UDP. Diese Variante verwendet eine verbindungslose asynchrone Client Server Kommunikation über Ethernet.
Bei Modbus-UDP erfolgt die TimeOut-Überwachung auf Applikationsebene (OSI-Schicht 7). Dadurch löst Modbus-UDP ein Problem, das entsteht, wenn der Modbus-Server(Slave) nicht verfügbar ist (zB. Versorgungsspannung getrennt).
Bei Modbus-TCP greifen in diesem Fall Retransmissionsmechanismen des TCP-Stacks, die dazu führen, das der Modbus-Client erst sehr spät feststellt, das die Gegenstelle nicht antwortet.

Allen Varianten gemeinsam ist ein einheitliches Anwendungsprotokoll, das ein universelles Objektmodell für Daten und Kommunikationsdienste für den Zugriff festlegt.

Das Modbusprotokoll arbeitet mit den folgenden Grunddatentypen:
- Discrete Inputs (digitale Eingänge),
- Coils (digitale Ausgänge),
- Input Register (16Bit Eingangsdaten) und
- Holding Register (16Bit Ausgangsdaten).

Modbus unterscheidet zwischen digitalen Diensten und Register-Diensten. Die digitalen Modbus-Diente werden häufig auch als auch Coil-Dienste bezeichnet.

Mit den digitalen Modbus-Diensten lassen ausschließlich die Zustände von digitalen I/O-Modulen ermitteln oder verändern. Komplexe I/O-Module werden ignoriert bzw. sind unerreichbar.

Für den Datenaustausch mit komplexen I/O-Modulen werden Register-Dienste eingesetzt.

Für die Bytereihenfolge in Modbus-Telegrammen legt die Spezifikation das Motorola-Format „Big-Endian“ fest, dabei wird das höchstwertige Byte (MSB) an die niedrigste Speicheradresse geschrieben.
Erfolgt die Auswertung der Telegramme auf einem PC, ist die Byteorder in das Intel-Format „Little Endian“ zu überführen.

Die nachfolgenden Tabellen zeigen beispielhaft den Aufbau des Request-Telgrammes sowie der zugehörige Response für FC3 „ReadHolding-Register“ in dem 5 Word’s ab Modbus-Adresse 0 (physikalische Eingänge) gelesen wird.

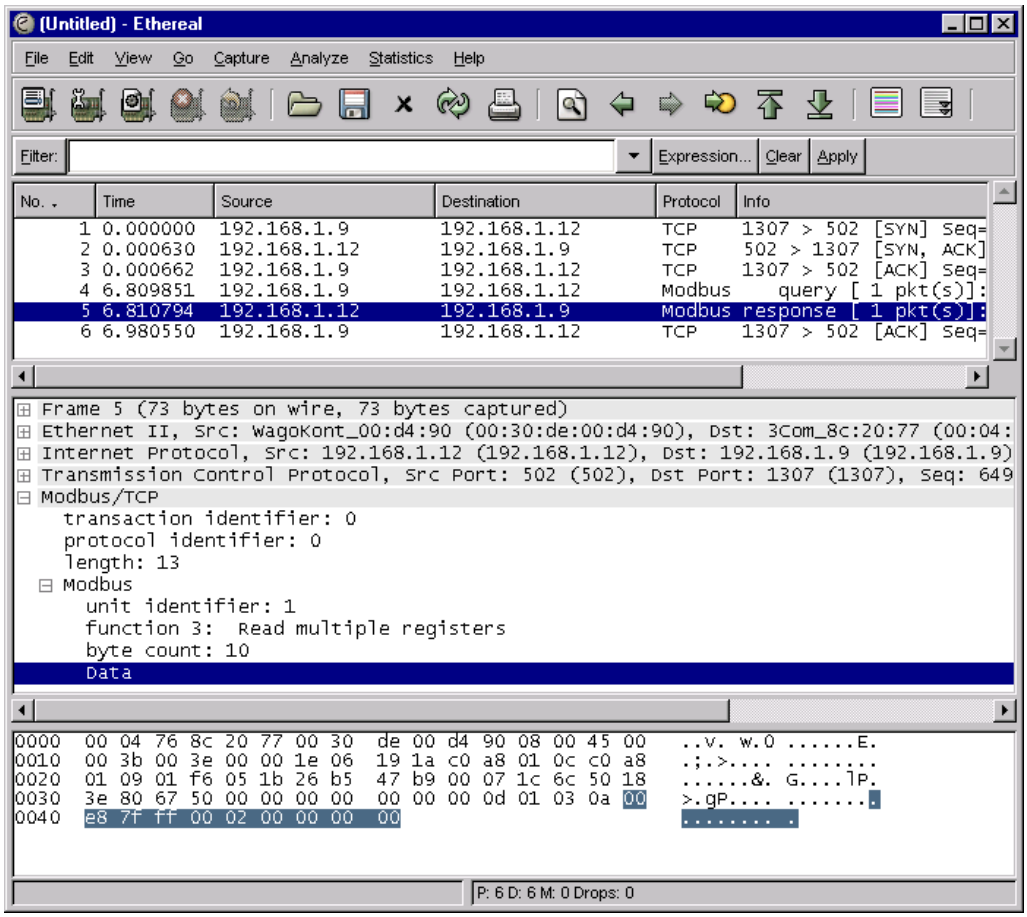
FC3 – Request:			**00 00 00 00 00 06 01 03 00 00 00 05**	
Byte	**Fieldname**	**Type**	**Example**	**Description**
0, 1	Transaction identifier	WORD	0x0000	Transaktionskennung zur Unterscheidung von Anfragen
2, 3	Protocol identifier	WORD	0x0000	Protokollkennung typisch 0
4, 5	Number of bytes following	WORD	0x0006	Anzahl folgender Bytes
6	Unit identifier	BYTE	0x01	Ehemals Slave-Id, frei für benutzerdefinerte Erweiterungen.
7	MODBUS function code	BYTE	0x03	Dienstkennung: Lesen/Schreiben von Coils/Registern
8, 9	Start address	WORD	0x0000	Modbus-Startadresse der Operation
10, 11	Word count	WORD	0x0005	Dienstabhängig die Anzahl Bit’s oder WORD’s

Die Antwortzeit eines WAGO Ethernet Koppler oder Controller auf ein Modbus-Request liegt bei ca. 2-3 Millisekunden.

FC3 – Response:			**00 00 00 00 00 0D 01 03 00 0A 00 E8 7F FF 00 02 00 00 00 00**	
Byte	**Fieldname**	**Type**	**Example**	**Description**
0, 1	Transaction identifier	WORD	0x0000	Transaktionskennung um mehrere Anfragen unterscheidbar zu machen
2, 3	Protocol identifier	WORD	0x0000	Protokollkennung typisch 0
4, 5	Number of bytes following	WORD	0x000D (13)	Anzahl folgender Bytes
6	Unit identifier	BYTE	0x01	Ehemals Slave-Id, frei für benutzerdefinerte Informationen.
7	MODBUS function code	BYTE	0x03	Dienstkennung: Lesen/Schreiben von Coils/Registern
8, 9	Byte count	WORD	0x000A (10)	Anzahl Daten Bytes
10 ... 19	Data	ARRAY OF WORD	0x00E8 0x7FFF 0x0002 0x0000 0x0000	Nutzdaten (im Motorola-Format „Big-Endian“)

Die maximale Telegrammlänge wird durch den Datentyp des Feldes „Byte count“ bestimmt. Es wurde der Datentype BYTE spezifiziert der lediglich Werte zwischen 0 und 255 annehmen kann. Damit lassen sich, abhängig vom Modbus-Dienst, ca. 120 WORD’s Nutzdaten transportieren.

Nachfolgend sehen Sie den Screenshot des Programms „Ethereal“.



Der Mitschnitt zeigt das Antworttelegramm aus dem eben beschriebenen Beispiel.

Im Unterschied zu einer großen Gruppe von Anwendungen wird das Mitschneiden einer Kommunikation nicht durch den Menüpunkt „File->New“ gestartet, sondern ist unter dem Menüpunkt „Capture“ versteckt.
Weiterführende Informationen finden Sie unter www.ethereal.com.

2.3.1.1 Funktionscode FC1 (Read Coils)

Diese Funktion liest den Inhalt mehrerer Eingangs- und Ausgangsbits.

Aufbau des Requests

Die Anfrage bestimmt die Startadresse und die Anzahl der Bits, die gelesen werden sollen.
Beispiel: Eine Anfrage, durch die die Bits 0 bis 7 gelesen werden sollen.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0006
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x01
Byte 8, 9	reference number	0x0000
Byte 10, 11	Bit count	0x0008

Aufbau der Response

Die aktuellen Werte der abgefragten Bits werden in das Datenfeld gepackt. Eine 1 entspricht dabei dem Zustand ON und eine 0 dem Zustand OFF. Das niederwertigste Bit des ersten Datenbytes enthält das erste Bit der Anfrage. Die anderen folgen aufsteigend. Falls die Anzahl der Eingänge kein Vielfaches von 8 ist, werden die restlichen Bits des letzten Datenbytes mit Nullen aufgefüllt.

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x01
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

Der Status der Eingänge 7 bis 0 wird als Byte-Wert 0x12 oder binär 0001 0010 angezeigt. Eingang 7 ist das höchstwertige Bit dieses Bytes und Eingang 0 das niederwertigste. Die Zuordnung erfolgt damit von 7 bis 0 mit OFF-OFF-OFF-ON-OFF-OFF-ON-OFF.

Bit:	0	0	0	1	0	0	1	0
Coil:	7	6	5	4	3	2	1	0

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x81
Byte 8	Exception code	0x01 oder 0x02

2.3.1.2 Funktionscode FC2 (Read Input Discretes)

Diese Funktion liest den Inhalt mehrerer Eingangsbits (Digitaler Eingänge).

Aufbau des Requests

Die Anfrage bestimmt die Startadresse und die Anzahl der Bits, die gelesen werden sollen.
Beispiel: Eine Anfrage, durch die die Bits 0 bis 7 gelesen werden sollen.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	Length field	0x0006
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x02
Byte 8, 9	reference number	0x0000
Byte 10, 11	Bit count	0x0008

Aufbau der Response

Die aktuellen Werte der abgefragten Bits werden in das Datenfeld gepackt. Eine 1 entspricht dabei dem Zustand ON und eine 0 dem Zustand OFF. Das niederwertigste Bit des ersten Datenbytes enthält das erste Bit der Anfrage. Die anderen folgen aufsteigend. Falls die Anzahl der Eingänge kein Vielfaches von 8 ist, werden die restlichen Bits des letzten Datenbytes mit Nullen aufgefüllt.

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x02
Byte 8	Byte count	0x01
Byte 9	Bit values	0x12

Der Status der Eingänge 7 bis 0 wird als Byte-Wert 0x12 oder binär 0001 0010 angezeigt. Eingang 7 ist das höchstwertige Bit dieses Bytes und Eingang 0 das niederwertigste. Die Zuordnung erfolgt damit von 7 bis 0 mit OFF-OFF-OFF-ON-OFF-OFF-ON-OFF.

Bit:	0	0	0	1	0	0	1	0
Coil:	7	6	5	4	3	2	1	0

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x82
Byte 8	Exception code	0x01 oder 0x02

2.3.1.3 Funktionscode FC3 (Read multiple registers)

Diese Funktion dient dazu, eine Anzahl von Eingangsworten (auch ”Eingangsregister”) zu lesen.

Aufbau des Requests

Die Anfrage bestimmt die Adresse des Startwortes (Startregister) und die Anzahl der Register, die gelesen werden sollen. Die Adressierung beginnt mit 0.

Beispiel: Abfrage der Register 0 und 1.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0006
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x03
Byte 8, 9	reference number	0x0000
Byte 10, 11	Word count	0x0002

Aufbau der Response

Die Registerdaten der Antwort werden als 2 Bytes pro Register gepackt. Das erste Byte enthält dabei die höherwertigen Bits, das zweite die niederwertigen.

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x03
Byte 8	Byte count	0x04
Byte 9, 10	Value Register 0	0x1234
Byte 11, 12	Value Register 1	0x2345

Aus der Antwort ergibt sich, dass Register 0 den Wert 0x1234 und Register 1 den Wert 0x2345 enthält.

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x83
Byte 8	Exception code	0x01 oder 0x02

2.3.1.4 Funktionscode FC4 (Read input registers)

Diese Funktion dient dazu, eine Anzahl von Eingangsworten (auch ”Eingangsregister”) zu lesen.

Aufbau des Requests

Die Anfrage bestimmt die Adresse des Startwortes (Startregister) und die Anzahl der Register, die gelesen werden sollen. Die Adressierung beginnt mit 0.

Beispiel: Abfrage der Register 0 und 1.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0006
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x04
Byte 8, 9	reference number	0x0000
Byte 10, 11	Word count	0x0002

Aufbau der Response

Die Registerdaten der Antwort werden als 2 Bytes pro Register gepackt. Das erste Byte enthält dabei die höherwertigen Bits, das zweite die niederwertigen.

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x04
Byte 8	Byte count	0x04
Byte 9, 10	Value Register 0	0x1234
Byte 11, 12	Value Register 1	0x2345

Aus der Antwort ergibt sich, dass Register 0 den Wert 0x1234 und Register 1 den Wert 0x2345 enthält.

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x84
Byte 8	Exception code	0x01 oder 0x02

2.3.1.5 Funktionscode FC5 (Write Coil)

Diese Funktion dient dazu, ein digitales Ausgangsbit zu schreiben.

Aufbau des Requests

Die Anfrage bestimmt die Adresse des Ausgangsbits. Die Adressierung beginnt mit 0.

Beispiel: Setzen des 2. Ausgangsbits (Adresse 1).

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0006
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x05
Byte 8, 9	reference number	0x0001
Byte 10	ON/OFF	0xFF
Byte 11		0x00

Aufbau der Response

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x05
Byte 8, 9	Reference number	0x0001
Byte 10	Value	0xFF
Byte 11		0x00

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01, 0x02 oder 0x03

2.3.1.6 Funktionscode FC6 (Write single register)

Diese Funktion schreibt einen Wert in ein einzelnes Ausgangswort (auch ”Ausgangsregister”).

Aufbau des Requests

Die Adressierung beginnt mit 0. Die Anfrage bestimmt die Adresse des ersten Ausgangswortes, das gesetzt werden soll. Der zu setzende Wert wird im Anfragedatenfeld bestimmt.

Beispiel: Setzen des zweiten Ausgangskanal (Adresse 0) auf den Wert 0x1234.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0006
Byte 6	Unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x06
Byte 8, 9	reference number	0x0001
Byte 10, 11	Register Value	0x1234

Aufbau der Response

Die Antwort ist ein Echo der Anfrage.

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x06
Byte 8, 9	Reference number	0x0001
Byte 10, 11	Register Value	0x1234

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 oder 0x02

2.3.1.7 Funktionscode FC11 (Get comm event counter)

Diese Funktion gibt ein Statuswort und einen Ereigniszähler aus dem Kommunikations-Ereigniszähler des Controllers zurück. Die übergeordnete Steuerung kann mit diesem Zähler feststellen, ob der Controller die Nachrichten fehlerlos behandelt hat.

Nach jeder erfolgreichen Nachrichtenverarbeitung wird der Zähler hochgezählt. Dieses Zählen erfolgt nicht bei Ausnahmeantworten oder Zählerabfragen.

Aufbau des Requests

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0002
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x0B

Aufbau der Response

Die Antwort enthält ein 2 Byte Statuswort und einen 2 Byte Ereigniszähler. Das Statuswort besteht aus Nullen.

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Status	0x0000
Byte 10, 11	Event Count	0x0003

Der Ereigniszähler zeigt, dass 3 (0x0003) Ereignisse gezählt wurden.

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 oder 0x02

2.3.1.8 Funktionscode FC15 (Force Multiple Coils)

Durch diese Funktion wird eine Anzahl Ausgangsbits auf 1 oder 0 gesetzt. Die maximale Anzahl ist 256 Bits.

Aufbau des Requests

Der erste Bit wird mit 0 adressiert. Die Anfragenachricht spezifiziert die Bits, die gesetzt werden sollen. Die geforderten 1-oder 0-Zustände werden durch die Inhalte des Anfragedatenfelds bestimmt.

In diesem Beispiel werden 16 Bits beginnend mit Adresse 0 gesetzt. Die Anfrage enthält 2 Bytes mit dem Wert 0xA5F0 also 1010 0101 1111 0000 binär.

Das erste Byte überträgt die 0xA5 an die Adresse 7 bis 0, wobei 0 das niederwertigste Bit ist. Das nächste Byte überträgt 0xF0 an die Adresse 15 bis 8, wobei das niederwertigste Bit 8 ist.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	Length field	0x0009
Byte 6	unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x0F
Byte 8, 9	reference number	0x0000
Byte 10, 11	Bit Count	0x0010
Byte 12	Byte Count	0x02
Byte 13	Data Byte1	0xA5
Byte 14	Data Byte2	0xF0

Aufbau der Response

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x0F
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Bit Count	0x0010

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x8F
Byte 8	Exception code	0x01 oder 0x02

2.3.1.9 Funktionscode FC16 (Write multiple registers)

Diese Funktion schreibt Werte in eine Anzahl von Ausgangsworten (auch ”Ausgangsregister”).

Aufbau des Requests

Das erste Register wird mit 0 adressiert.

Die Anfragenachricht bestimmt die Register, die gesetzt werden sollen. Die Daten werden als 2 Bytes pro Register gesendet.

Beispiel: Die Daten in den beiden Registern 0 und 1 werden gesetzt.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x000B
Byte 6	Unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x10
Byte 8, 9	reference number	0x0000
Byte 10, 11	Word count	0x0002
Byte 12	Byte Count	0x04
Byte 13, 14	Register Value 1	0x1234
Byte 15, 16	Register Value 2	0x2345

Aufbau der Response

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x10
Byte 8, 9	Reference number	0x0000
Byte 10, 11	Word Count	0x0002

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 oder 0x02

2.3.1.10 Funktionscode FC22 (Mask Write Register)

Diese Funktion dient dazu einzelne Bits innerhalb eines Registers zu manipulieren.

Aufbau des Requests

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x0002
Byte 6	Unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x16
Byte 8-9	Reference Number	0x0000
Byte 10-11	AND-Mask	0x0000
Byte 12-13	OR-Mask	0xAAAA

Aufbau der Response

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x10
Byte 8-9	Reference Number	0x0000
Byte 10-11	AND-Mask	0x0000
Byte 12-13	OR-Mask	0xAAAA

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x85
Byte 8	Exception code	0x01 oder 0x02

2.3.1.11 Funktionscode FC23 (Read/Write multiple registers)

Diese Funktion liest Registerwerte aus und schreibt Werte in eine Anzahl von Ausgangsworten (auch ”Ausgangsregister”).

Aufbau des Requests

Beispiel: Die Daten in dem Register 3 werden auf den Wert 0x0123 gesetzt und aus den beiden Registern 0 und 1 werden die Werte 0x0004 und 0x5678 gelesen.

Byte	Feldname	Beispiel
Byte 0, 1	Transaction identifier	0x0000
Byte 2, 3	protocol identifier	0x0000
Byte 4, 5	length field	0x000F
Byte 6	Unit identifier	0x01 nicht verwendet
Byte 7	MODBUS function code	0x17
Byte 8-9	reference number for read	0x0000
Byte 10-11	Word count for read (1-125)	0x0002
Byte 12-13	reference number for write	0x0003
Byte 14-15	Word count for write (1-100)	0x0001
Byte 16	Byte Count (B = 2 x word count for write)	0x02
Byte 17-(B+16)	Register Values	0x0123

Aufbau der Response

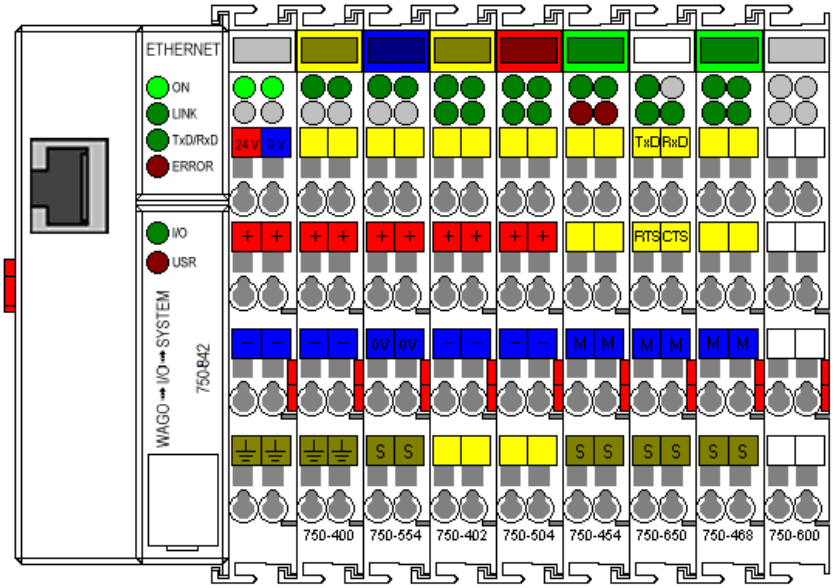
Byte	Feldname	Beispiel
....		
Byte 7	MODBUS function code	0x17
Byte 8	Byte Count (B = 2 x word count for read)	0x04
Byte 9- (B+1)	Register Values	0x0004 0x5678

Aufbau der Exception

Byte	Feldname	Beispiel
.....		
Byte 7	MODBUS function code	0x97
Byte 8	Exception code	0x01 oder 0x02

2.3.2 Beispiele

Für den aus den vorangegangenen Kapitel bekannten Knotenaufbau soll der dritte Kanal, der digitale Ausgangsklemme 750-504, über das Modbus-Protokoll verändert werden.



Grundsätzlich könnte jeder der folgenden Modbus-Dienste dazu verwendet werden den Zustand des dritten Kanals der digitale Ausgangsklemme 750-504 zu verändern, jedoch unterstützt nicht jeder Modbus-Master alle Modbus-Dienste.

FC	Name	Description
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analoger Ausgänge
FC22	Mask write	Maskiertes schreiben eines Registers
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

In diesem Kapitel sollen verschiedene Lösungen vorgestellt werden. In Variante 1 kommt der Registerdienst FC16 zum Einsatz und in Variante 2 wird der digitale Modbus-Dienste FC15 verwendet. Abschluss bildet in Variante 3 der etwas seltsam anmutende Registerdienst FC22.

Bei Registerdiensten gilt es letztlich immer die zur IEC-Adresse korrespondierende Modbus-Adresse, für den gewählten Modbus-Dienstes, zu ermitteln.

Mit diesen Informationen läst sich dann der Modbus-Master parametrieren.

2.3.2.1 Beispiel: FC16 (Write multiple register)

Bei dem Modbus-Dienst FC16 „Write multiple register“ handelt es sich um einen Register-Dienst der es ermöglicht bis zu 120 Register mit einem Telegramm zu verändern.
Neben der Modbus-Adresse des Registers wird mit „NumberOfPoints“ die Anzahl der zu verändernder Register sowie die Daten selbst an den Modbus-Slave gesendet.

Die Bestimmung der Modbus-Adresse beginnt mit der Auswertung des Knotenaufbaus bzw. mit dem Aufbau des Prozessabbildes der Ausgänge. Dies kann wie in den vorangegangenen Kapitel gezeigt „zu Fuß“ oder bei WAGO Ethernet Controllern mit Unterstützung der CoDeSys-Steuerungskonfiguration erfolgen.

Beide Verfahren sollten „%QW4.2“ als IEC-Adresse des 3ten digitalen Ausgangs der ersten 750-504 liefern. Die Modbus-Adresse kann nun der Adresszuordnungstabelle des 750-841 entnommen werden.

750-841: Modbus- vs IEC61131-Addresses for FC6, FC16, FC22 and FC23			
Modbus-Address		**IEC61131**	**Description**
[dec]	**[hex]**	**Address**	
0 ... 255	0x0000 ... 0x00FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data

Die Lesart der Adresszuordnungstabelle ist wie folgt: Das Ausgangswort %QW0 erreichen wir über die Modbus-Adresse 0.
Damit ergibt sich „4“ auch als Modbus-Adresse für %QW4.

In dieser Variante kann der 3 digitale Ausgang nicht unabhängig von den anderen gesetzt werden, was bei Verwendung einer „Statemachine“ im SPS-Programm kein Nachteil ist. So können Ausgangsmuster definiert werden und die Anlage/Maschine ist immer in einem bekannten „State“.

Wird einzig der dritte Kanal der 750-504 verwendet so läst sich der digitale Ausgang mit dem Datum 0x0004 einschalten und mit dem Datum 0x0000 zurücksetzen.

2.3.2.2 Beispiel: FC15 (Force multiple coils)

Der Modbus-Dienst FC15 „Force multiple coils“ erlaubt bis zu 512 digitale Ausgänge mit einem Telegram zu verändern. Da es sich um einen „digitalen“ Modbus-Dienst handelt, werden die komplexen I/O-Module ignoriert und die Modbus-Adresse ist gleichbedeutend mit der Frage um den wievielten digitalen Ausgang es sich handelt.
In dieser Variante ist das die Modbus-Addresse 2, da jeder Modbus-Index bei Null beginnt.

2.3.2.3 Beispiel: FC22 (Mask write)

Bei dem Modbus-Dienst FC22 „Mask write“ handelt es sich um einen Register-Dienst der es ermöglicht einzelne Bit’s in einem Register gezielt zu verändern. Neben der Modbus-Adresse des Registers wird eine UND-Maske und eine ODER-Maske übergeben an den Modbus-Slave gesendet.

Die Bestimmung der Modbus-Adresse erfolgt analog zur vorangegangenen Variante, somit befindet sich der 3te digitale Ausgang der ersten 750-504 im Modbus-Register mit der Adresse „4“.

Die beiden Maskenregister arbeiten nach folgender Regel:

```
Result:= (Content AND AndMask) OR (OrMask AND (NOT AndMask))
```

Um den 3 Ausgang zu setzen und alle anderen Ausgänge unverändert zu lassen, verwenden Sie „0xFFFB als UND-Maske und 0x0004 als ODER-Maske.

Um den 3 Ausgang zurück zu setzen und alle anderen Ausgänge unverändert zu lassen, verwenden Sie „0xFFFB als UND-Maske und 0x0000 als ODER-Maske.

3 WAGO Controller als Modbus-Master

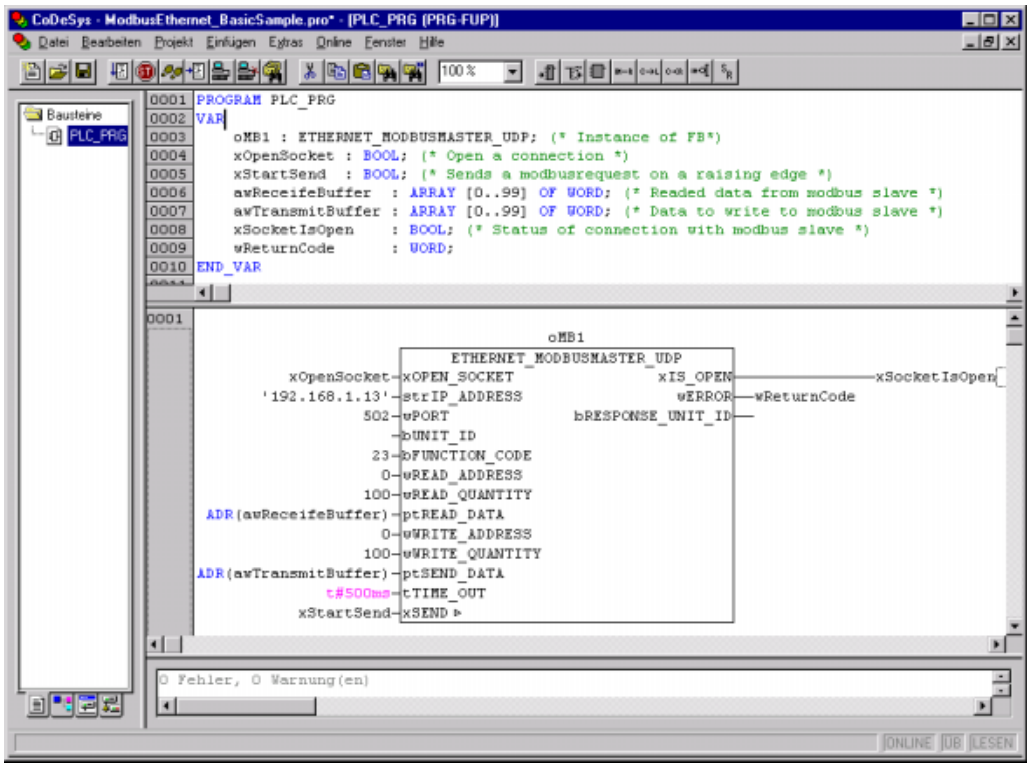
Der Unterschied zwischen WAGO Kopplern und Controllern, liegt in der Programmierbarkeit der Controller. Durch Programmierung kann jeder WAGO Ethernet Controller auch als Modbus-Master(Client) arbeiten.

Die Programmierung erfolgt mit CoDeSys CAA in den standardisierten Sprachen der IEC61131-3 (FUB, KOP, ST usw.).
WAGO unterstützt Sie bei der Programmierung mit Bibliotheken in denen die Modbus-Grundfunktionen bereitgestellt werden.

Plattform bedingt ist nicht jede Bibliothek auf jedem Controller anwendbar. Nachfolgende Tabelle zeigt den Stand im Januar 2007. Den aktuellen Stand können Sie unter support@wago.com erfragen.

Library	750-842	750-841	758-870
„ModbusEthernet_04.lib“	YES	aviable	NO
“WagoLibModbus_IP_01.lib”	NO	advised	YES

Das folgende Minimalprojekt schreibt mit dem Functioncode „23“ 100 Worte in den Modbus-Slave mit der IP-Adresse „192.168.1.13“, mit im selben Telegramm werden zusätzlich 100 Worte aus dem Modbus-Slave gelesen.



Eine Anleitung zur Bedienung des Beispiels finden Sie im Beispielprogramm.

4 PC als Modbus-Master

Mit der „MBT.dll“ 759-312 stellt WAGO eine prozedurale DLL zur Verfügung die das Modbus/TCP Protokoll implementiert.

Die Modbus/TCP DLL unterstützt die Betriebssysteme Windows 95, Windows 98, Windows NT 4.0 (abSP5), Windows 2000 und Windows XP. Für Windows 95 ist eine Aktualisierung auf „Windows Socket 2.0“ erforderlich.

Als Transportprotokoll kann wahlweise TCP oder UDP gewählt werden. WAGO empfiehlt die Verwendung UDP als Transportprotokoll, da dieses ein verbessertes „Timeout-Handling“ ermöglicht.

Die „MBT.dll“ kann aus einer Vielzahl von Programmiersprachen verwendet werden. Auf der Auslieferungs-CD finden sich Beispiele für VBA(Excel), VB6, LabView, C, VC++ 6, Delphi, vb.net und C#.

Vom Open Modbus/TCP Protokoll V1.0 werden die Kommandos: FC1, FC2, FC3, FC4, FC7, FC15 und FC16 unterstützt.

Eine Installation oder Registrierung der „MBT.dll“ ist nicht erforderlich.
Es genügt die DLL in das Windows Standardverzeichnis „\system32 zu kopieren. Wird ein anderes Verzeichnis gewählt, so muss in der Windows-Systemsteuerung bei den Umgebungsvariablen der Pfad zur „MBT.dll“ hinzu gefügt werden.

Die MBT.dll stellt die folgende Funktionen bereit:

MBTInit(); MBTExit()
MBTConnect(); MBTDisconnect()
MBTReadRegisters();MBTWriteRegisters()
MBTReadCoils();MBTWriteCoils()
MBTSwapWord(); MBTSwapDWord()

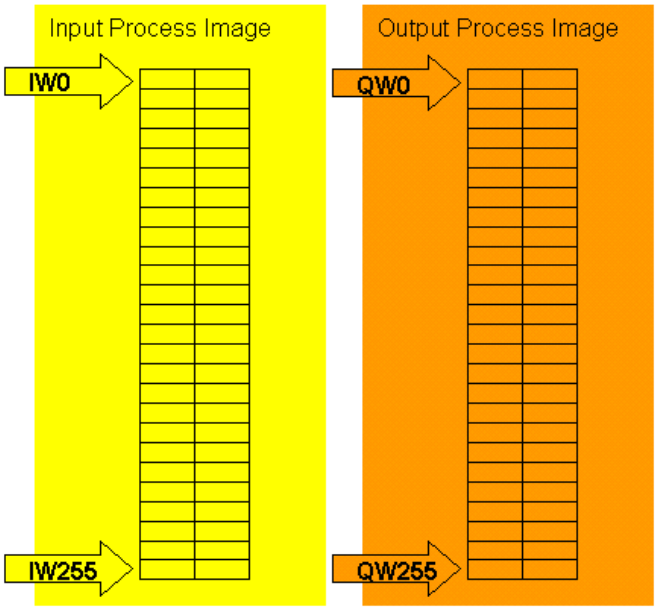
Alle Funktionen der MBT Library haben Rückgabewerte, die dem HRESULT Format entsprechen. Die Funktionen des Socket APIs geben keine Rückgabewerte dieses Formates zurück. Die MBT Library konvertiert diese Rückgabewerte mittels des Makros HRESULT_FROM_WIN32. Bei der nachfolgenden Beschreibung wird dies mittels "HR von" gekennzeichnet.

In einem Programm sollte bei Programmstart „MBTInit()“ einmalig aufgerufen werden, die Funktion beschafft benötigte Ressourcen und initialisiert die DLL. Mit „MBTConnect()“ wird eine Verbindung zu einem entfernten Modbus-Slave(Server) aufgebaut. Der Datenaustausch erfolgt mit den Funktionen „MBTWriteRegisters()“, „MBTReadRegisters()“, „MBTWriteCoils()“ und „MBTReadCoils()“. Sind alle Daten ausgetauscht wird mit der Funktion „MBTDisconnect()“ die Verbindung abgebaut. Nachfolgen kann ein erneuter Verbindungsaufbau oder das Programmende. Zur sicheren Freigabe der Ressourcen sollte bei Programmende aber auch beim Abbruch des Programmes die Funktion „MBTExit()“ einmalig ausgeführt werden.

5 Anhang

5.1 Prozessabbild des 750-342

Der 750-342 kann maximal 3 eingehende Modbus-TCP-Verbindungen bearbeiten. Der Modbus-Verbindungs-Watchdog ist im Auslieferungszustand deaktiviert.



Neben der WAGO Grundausstattung an Modbus-Diensten unterstützt der 750-342 zusätzlich den Funktionscode FC7 „Read exception status“

FC	Name	Description
FC1	Read coils	Rücklesen mehrerer digitaler Ausgänge
FC2	Read inputs discrete	Lesen mehrerer digitaler Eingänge
FC3	Read holding registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC4	Read input registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC7	Read exception status	Rücklesen der ersten 8 digitaler Ausgänge
FC11	Get comm event counter	Kommunikationsereigniszähler
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analoger Ausgänge
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

5.1.1 Register Dienste des 750-342

Über die Register-Dienste lassen sich die Zustände von komplexen und digitalen I/O-Modulen ermitteln oder verändern.

5.1.1.1 Register lesen mit FC3 und FC4:

750-342: Modbus- vs IEC61131-Addresses for FC3 and FC4			
Modbus-Address [dec]	Modbus-Address [hex]	IEC61131 Address	Description
0 ... 255	0x0000 ... 0x00FF	%IW0 ... %IW255	Physical Input Area
256 ... 511	0x0100 ... 0x01FF	-	Modbus Exception: “Illegal data address”
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area
768 ... 4095	0x0300 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 65535	0x3000 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.1.1.2 Register schreiben mit FC6 und FC16:

750-342: Modbus- vs IEC61131-Addresses for FC6 and FC16			
Modbus-Address [dec]	Modbus-Address [hex]	IEC61131 Address	Description
0 ... 255	0x0000 ... 0x00FF	%QW0 ... %QW255	Physical Output Area
256 ... 511	0x0100 ... 0x01FF	-	Modbus Exception: “Illegal data address”
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area
768 ... 4095	0x0300 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 65535	0x3000 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.1.2 Digitale Modbus-Dienste des 750-342

Mit den digitalen Modbus-Diensten lassen ausschließlich die Zustände von digitalen I/O-Modulen ermitteln oder verändern. Komplexe I/O-Module werden ignoriert bzw. sind unerreichbar.

5.1.2.1 Coils lesen mit FC1 und FC2:

750-342: Modbus-Addresses for FC1 and FC2			
Modbus-Address		Memory	Description
[dec]	[hex]	Area	
0 ... 511	0x0000 ... 0x01FF	Physical Input Area	First 512 digital inputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area	First 512 digital outputs
1024 ... 65535	0x0400 ... 0xFFFF		Modbus Exception: “Illegal data address”

5.1.2.2 Coils schreiben mit FC5 und FC15:

750-342: Modbus-Addresses for FC5 and FC15			
Modbus-Address		Memory	Description
[dec]	[hex]	Area	
0 ... 511	0x0000 ... 0x01FF	Physical Output Area	max 512 digital outputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area	max. 512 digital outputs
1024 ... 65535	0x0400 ... 0xFFFF		Modbus Exception: “Illegal data address”

5.1.3 Modbus Konfigurationsregister des 750-342

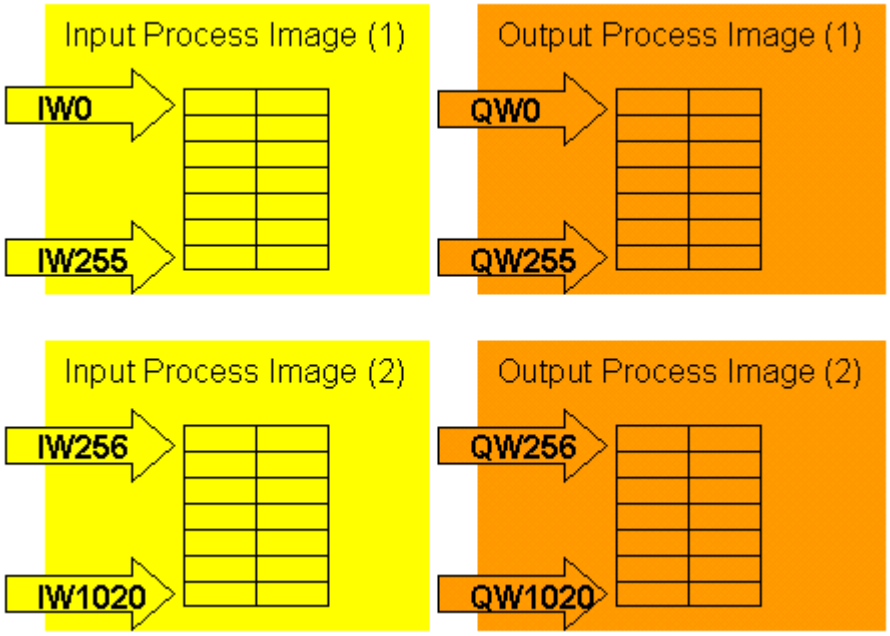
Über die Konfigurationsregister lassen sich die Eigenschaften des 750-342 ermitteln und teilweise verändern.

750-342: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
4096	0x1000	1	R/W	ModbusWatchdogTime (Vielfaches von 100ms)
4097	0x1001	1	R/W	ModbusWatchdogCodiermaske 1-16
4098	0x1002	1	R/W	ModbusWatchdogCodiermaske 17-32
4099	0x1003	1	R/W	ModbusWatchdogTrigger
4100	0x1004	1	R	Minimale Triggerzeit
4101	0x1005	1	R/W	ModbusWatchdog stoppen (0xAAAA und 0x5555)
4102	0x1006	1	R	ModbusWatchdog Status
4103	0x1007	1	R/W	ModbusWatchdog restart (0x0001)
4104	0x1008	1	R/W	ModbusWatchdog stoppen (0x55AA oder 0xAA55)
4105	0x1009	1	R/W	Modbus- und HTTP-Port schließen nach Timeout
4106	0x100A	1	R/W	ModbusWatchdog im „Modicon Mode“ starten
4107	0x100B	1	W	ModbusWatchdogParameter speichern
4128	0x1020	1	R	LED Error-Code
4129	0x1021	1	R	LED Error-Argument
4130	0x1022	1	R	Anzahl analoger Ausgänge im PA [Bit]
4131	0x1023	1	R	Anzahl analoger Eingänge im PA [Bit]
4132	0x1024	1	R	Anzahl digitaler Ausgänge im PA [Bit]
4133	0x1025	1	R	Anzahl digitaler Eingänge im PA [Bit]
4135	0x1027	1	R	Klemmenbuszyklus ausführen
4136	0x1028	1	R/W	IP-Configuration: BootP(1) or FIX(0)
4137	0x1029	18	R	Modbus-TCP-Statistik
4144	0x1030	1	R/W	Modbus Verbindungsüberwachung aktivieren
4145	0x1031	3	R	MAC-ID der Ethernetschnittstelle
8192	0x2000	1	R	0x0000 (Constant)
8193	0x2001	1	R	0xFFFF (Constant)
8194	0x2002	1	R	0x1234 (Constant)
8195	0x2003	1	R	0xAAAA (Constant)
8196	0x2004	1	R	0x5555 (Constant)
8197	0x2005	1	R	0x7FFF (Constant)
8198	0x2006	1	R	0x8000 (Constant)
8199	0x2007	1	R	0x3FFF (Constant)
8200	0x2008	1	R	0x4000 (Constant)
8208	0x2010	1	R	Firmware release
8209	0x2011	1	R	Seriencode (750)
8210	0x2012	1	R	Gerätecode (342)
8211	0x2013	1	R	Spezielle Firmwareversion (0xFFFF)
8212	0x2014	1	R	Spezielle Firmwareversion (0xFFFF)
8224	0x2020	1 .. 125	R	Gerätekurzbeschreibung
8225	0x2021	8	R	Compile-Zeit der Firmwareversion
8226	0x2022	8	R	Compile-Datum der Firmwareversion

750-342: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
8227	0x2023	32	R	Version des Firmwareloaders (FWL)
8240	0x2030	65	R	Beschreibung angeschlossener IO-Module: 0-64
8245	0x2035	1	R/W	Einstellung Prozessabbild (Table 0 register 3)
8246	0x2036	1 .. 17	R	Diagnoseinformation Gerät
8256	0x2040	1	W	Software Reset (write 0x55AA or 0xAA55)
8260	0x2044	1	W	Delete Modbus Configurationfile (write 0x55AA)

5.2 Prozessabbild des 750-341

Der 750-341 kann maximal 15 eingehende Modbus-TCP-Verbindungen bearbeiten. Der Modbus-Verbindungs-Watchdog ist im Auslieferungszustand aktiviert.



Neben der WAGO Grundausstattung an Modbus-Diensten unterstützt der 750-341 zusätzlich den Funktionscode FC22 „Mask write“.

FC	Name	Description
FC1	Read coils	Rücklesen mehrerer digitaler Ausgänge
FC2	Read inputs discrete	Lesen mehrerer digitaler Eingänge
FC3	Read holding registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC4	Read input registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC11	Get comm event counter	Kommunikationsereigniszähler
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analoger Ausgänge
FC22	Mask write	Manipulation einzelner Bits eines Registers
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

5.2.1 Registerdienste des 750-341

5.2.1.1 Register lesen mit FC3 und FC4:

750-341: Modbus- vs IEC61131-Addresses for FC3 and FC4			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%IW0 ... %IW255	Physical Input Area (1) First 256 Words of physical input data
256 ... 511	0x0100 ... 0x01FF	-	Modbus Exception: “Illegal data address”
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data
768 ... 4095	0x0300 ... 0x0FFF	-	Modbus Exception: “ Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	-	Modbus Exception: “ Illegal data address”
24576 ... 25340	0x6000 ... 0x62FC	%IW256 ... %IW1020	Physical Input Area (2) Additional 764 Words physical input data
25341 ... 28671	0x62FD ... 0x6FFF	-	Modbus Exception: “ Illegal data address”
28672 ... 29436	0x7000 ... 0x72FC	%QW256 ... %QW1020	Physical Output Area (2) Additional 764 Words physical output data
29437 ... 65535	0x72FD ... 0xFFFF	-	Modbus Exception: “ Illegal data address”

Register schreiben mit FC6 und FC16:

750-341: Modbus- vs IEC61131-Addresses for FC6 and FC16			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data
256 ... 511	0x0100 ... 0x01FF	-	Modbus Exception: “Illegal data address”
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data
768 ... 4095	0x0300 ... 0x0FFF	-	Modbus Exception: “ Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	-	Modbus Exception: “ Illegal data address”
24576 ... 25340	0x6000 ... 0x62FC	%QW256 ... %QW1020	Physical Output Area (2) Additional 764 Words physical output data
25341 ... 28671	0x62FD ... 0x6FFF	-	Modbus Exception: “ Illegal data address”
28672 ... 29436	0x7000 ... 0x72FC	%QW256 ... %QW1020	Physical Output Area (2) Additional 764 Words physical output data
29437 ... 65535	0x72FD ... 0xFFFF	-	Modbus Exception: “ Illegal data address”

5.2.2 Digitale Modbus-Dienste des 750-341

Mit den digitalen Modbus-Diensten lassen ausschließlich die Zustände von digitalen I/O-Modulen ermitteln oder verändern. Komplexe I/O-Module werden ignoriert bzw. sind unerreichbar.

5.2.2.1 Coils lesen mit FC1 und FC2:

750-341: Modbus-Addresses for FC1 and FC2			
Modbus-Address [dec]	Modbus-Address [hex]	Memory Area	Description
0 ... 511	0x0000 ... 0x01FF	Physical Input Area (1)	First 512 digital inputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area (1)	First 512 digital outputs
1024 ... 32767	0x0400 ... 0x7FFF	-	Modbus Exception: “Illegal data address”
32768 ... 34295	0x8000 ... 0x85F7	Physical Input Area (2)	Starts with the 513th and ends with the 2039th digital input
34296 ... 36863	0x85F8 ... 0x8FFF		Modbus Exception: “Illegal data address”
36864 ... 38391	0x9000 ... 0x95F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
38392 ... 65535	0x95F8 ... 0xFFFF		Modbus Exception: “Illegal data address”

5.2.2.2 Coils schreiben mit FC5 und FC15:

750-341: Modbus-Addresses for FC1 and FC2			
Modbus-Address [dec]	Modbus-Address [hex]	Memory Area	Description
0 ... 511	0x0000 ... 0x01FF	Physical Output Area (1)	First 512 digital outputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area (1)	First 512 digital outputs
1024 ... 32767	0x0400 ... 0x7FFF	-	Modbus Exception: “Illegal data address”
32768 ... 34295	0x8000 ... 0x85F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
34296 ... 36863	0x85F8 ... 0x8FFF		Modbus Exception: “Illegal data address”
36864 ... 38391	0x9000 ... 0x95F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
38392 ... 65535	0x95F8 ... 0xFFFF		Modbus Exception: “Illegal data address”

5.2.3 Modbus Konfigurationsregister des 750-341

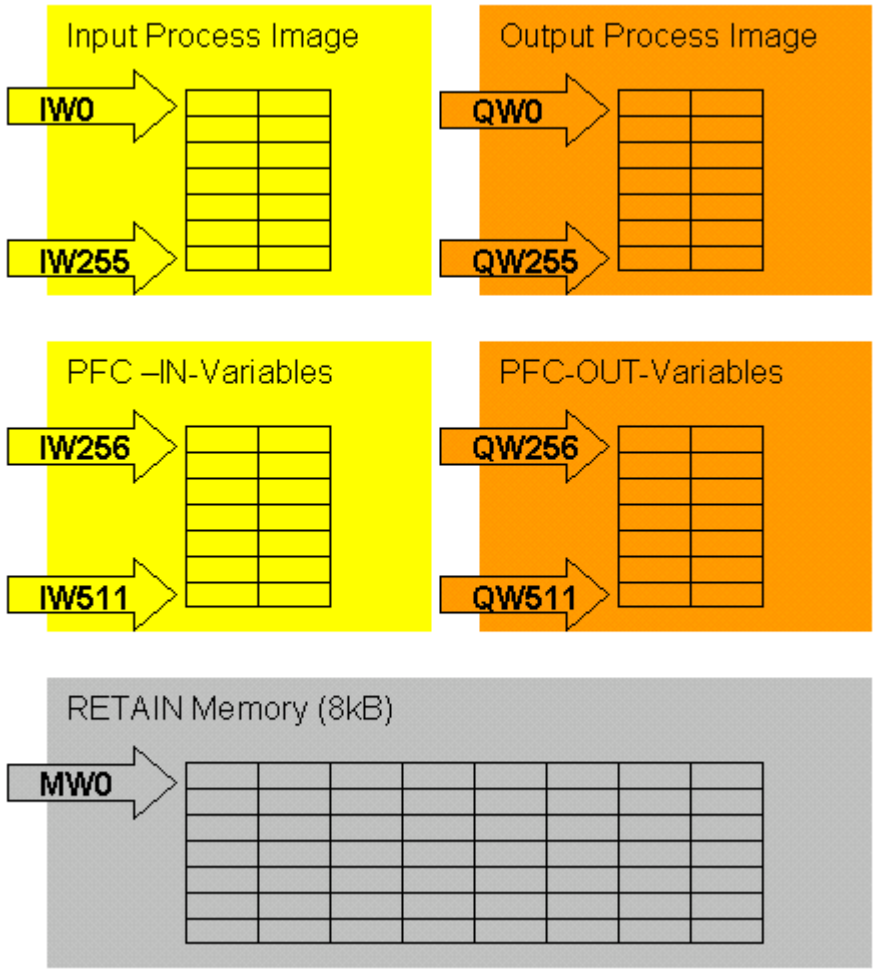
Über die Konfigurationsregister lassen sich die Eigenschaften des 750-341 ermitteln und teilweise verändern.

750-341: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address [dec]	[hex]	Length [Word]	Access	Description
4096	0x1000	1	R/W	ModbusWatchdogTime (Vielfaches von 100ms)
4097	0x1001	1	R/W	ModbusWatchdogCodiermaske 1-16
4098	0x1002	1	R/W	ModbusWatchdogCodiermaske 17-32
4099	0x1003	1	R/W	ModbusWatchdogTrigger
4100	0x1004	1	R	Minimale Triggerzeit
4101	0x1005	1	R/W	ModbusWatchdog stoppen (0xAAAA und 0x5555)
4102	0x1006	1	R	ModbusWatchdog Status
4103	0x1007	1	R/W	ModbusWatchdog restart (0x0001)
4104	0x1008	1	R/W	ModbusWatchdog stoppen (0x55AA oder 0xAA55)
4105	0x1009	1	R/W	Modbus- und HTTP-Port schließen nach Timeout
4106	0x100A	1	R/W	ModbusWatchdog im „Modicon Mode“ starten
4107	0x100B	1	W	ModbusWatchdogParameter speichern
4128	0x1020	1	R	LED Error-Code
4129	0x1021	1	R	LED Error-Argument
4130	0x1022	1	R	Anzahl analoger Ausgänge im PA [Bit]
4131	0x1023	1	R	Anzahl analoger Eingänge im PA [Bit]
4132	0x1024	1	R	Anzahl digitaler Ausgänge im PA [Bit]
4133	0x1025	1	R	Anzahl digitaler Eingänge im PA [Bit]
4136	0x1028	1	R/W	IP-Configuration: BootP(1), DHCP(2) or FIX(4)
4137	0x1029	18	R	Modbus-TCP-Statistik
4138	0x102A	1	R	Anzahl aufgebauter Modbus-TCP Verbindungen
4144	0x1030	1	R/W	Modbus Verbindungsüberwachung aktivieren
4145	0x1031	3	R	MAC-ID der Ethernetschnittstelle
4176	0x1050	3	R	Diagnoseinformationen angeschlossener IO-Module
8192	0x2000	1	R	0x0000 (Constant)
8193	0x2001	1	R	0xFFFF (Constant)
8194	0x2002	1	R	0x1234 (Constant)
8195	0x2003	1	R	0xAAAA (Constant)
8196	0x2004	1	R	0x5555 (Constant)
8197	0x2005	1	R	0x7FFF (Constant)
8198	0x2006	1	R	0x8000 (Constant)
8199	0x2007	1	R	0x3FFF (Constant)
8200	0x2008	1	R	0x4000 (Constant)
8208	0x2010	1	R	Firmware release
8209	0x2011	1	R	Seriencode (750)
8210	0x2012	1	R	Gerätecode (341)
8211	0x2013	1	R	Major Firmwareversion
8212	0x2014	1	R	Minor Firmwareversion

750-341: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
8224	0x2020	1 .. 125	R	Gerätekurzbeschreibung
8225	0x2021	8	R	Compile-Zeit der Firmwareversion
8226	0x2022	8	R	Compile-datum der Firmwareversion
8227	0x2023	32	R	Version des Firmwareloaders (FWL)
8240	0x2030	65	R	Beschreibung angeschlossener IO-Module: 0-64
8241	0x2031	64	R	Beschreibung angeschlossener IO-Module: 65-129
8242	0x2032	64	R	Beschreibung angeschlossener IO-Module: 130-194
8243	0x2033	63	R	Beschreibung angeschlossener IO-Module: 195-255
8245	0x2035	1	R/W	Einstellung Prozessabbild (Table 0 register 3)
8246	0x2036	1 .. 17	R	Diagnoseinformation Gerät
8256	0x2040	1	W	Software Reset (write 0x55AA or 0xAA55)
8257	0x2041	1	W	Format Flash-
8258	0x2042	1	W	Extract filesystem
8260	0x2044	1	W	Delete Modbus Configurationfile (write 0x55AA)

5.3 Prozessabbild des 750-842

Der 750-842 kann maximal 3 eingehende Modbus-TCP-Verbindungen gleichzeitig bearbeiten. Der Modbus-Verbindungs-Watchdog ist im Auslieferungszustand deaktiviert. Aus dem SPS-Programm lassen sich maximal zwei TCP-Verbindungen zu entfernten Servern herstellen.



Neben der WAGO Grundausstattung an Modbus-Diensten unterstützt der 750-842 zusätzlich den Funktionscode FC7 „Read exception status“

FC	Name	Description
FC1	Read coils	Rücklesen mehrerer digitaler Ausgänge
FC2	Read inputs discrete	Lesen mehrerer digitaler Eingänge
FC3	Read holding registers	Lesen mehrerer analogen Eingänge(und Ausgänge)
FC4	Read input registers	Lesen mehrerer analogen Eingänge(und Ausgänge)
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC7	Read exception status	Rücklesen der ersten 8 digitalen Ausgänge
FC11	Get comm event counter	Kommunikationsereigniszähler
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analogen Ausgänge
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

Mit den Funktionbausteinen „SET_DIGITAL_INPUT_OFFSET“ und „SET_DIGITAL_OUTPUT_OFFSET“ aus der Bibliothek „mod_com.lib“ können die Startadressen der ersten digitalen I/O-Module an WAGO Ethernet Controllern fest vorgegeben werden. Dies erlaubt Platz für spätere Erweiterungen einzuplanen. Die Angabe des OFFSET’s erfolgt in Bytes.

Memory area	Modbus-access	PLC access	Description
Physical. Input	read	read	Physikalische Eingänge (%IW0 ... %IW255)
Physical Output	read/write	read/write	Physikalische Ausgänge (%QW0 ... %QW255)
PFC-IN-	read/write	read	Flüchtige SPS-Eingangsvariablen (%IW256 ... %IW511)
PFC-OUT	read	read/write	Flüchtige SPS-Ausgangsvariablen (%QW256 ... %QW511)
Configuration register	read/(write)	---	Konfigurationsregister
RETAIN-(NOVRAM)	read/write	read/write	8kB remanenter Speicher (%MW0 ... %MW4095)

Beachten Sie, das die physikalischen Ausgänge sowohl über Modbus-Dienste wie aus dem SPS-Programm verändert werden können.

Ein Merksatz für dieses Verhalten könnte lauten:„Der Letzte gewinnt“.

Im 750-842 teilen sich „Merker-Variablen“ und „Retain-Variablen“ den gleichen Speicherplatz im NOVRAM.
Dies kann bei Überschneidungen zu unvorhersagbaren Verhalten führen.
Verwenden Sie nur einen der beiden Typen in Ihrem CoDeSys-Projekt.

5.3.1 Registerdienste des 750-842

5.3.1.1 Register lesen mit FC3 und FC4:

750-842: Modbus- vs IEC61131-Addresses for FC3 and FC4			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%IW0 ... %IW255	Physical Input Area
256 ... 511	0x0100 ... 0x01FF	%QW256 ... %QW511	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 16383	0x3000 ... 0x3FFF	%MW0 ... %MW4095	NOVRAM 8kB retain memory
16384 ... 65535	0x4000 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.3.1.2 Register schreiben mit FC6 und FC16:

750-842: Modbus- vs IEC61131-Addresses for FC6 and FC16			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%QW0 ... %QW255	Physical Output Area
256 ... 511	0x0100 ... 0x01FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	-	Configuration Register (see manual for details)
8192 ... 12287	0x2000 ... 0x2FFF	-	Modbus Exception: “Illegal data address”
12288 ... 16383	0x3000 ... 0x3FFF	%MW0 ... %MW4095	NOVRAM 8kB retain memory
16384 ... 65535	0x4000 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.3.2 Digitale Modbus-Dienste des 750-842

Mit den digitalen Modbus-Diensten lassen ausschließlich die Zustände von digitalen I/O-Modulen ermitteln oder verändern. Komplexe I/O-Module werden ignoriert bzw. sind unerreichbar.

Im PFC-IN und PFC-OUT Bereich sowie im Merker-Bereich(NOVRAM) wirken Coil-Dienste und Registerdienste auf die gleichen Speicherstellen.

Aufgrund des durch den Datentyp „WORD“ eingeschränkten Adressraumes lassen sich nicht alle Bits im 8kB Merker-Bereich durch Coils-Dienste adressieren.

5.3.2.1 Coils lesen mit FC1 und FC2:

750-842: Modbus-Addresses for FC1 and FC2			
Modbus-Address		**Memory**	**Description**
[dec]	**[hex]**	**Area**	
0 ... 511	0x0000 ... 0x01FF	Physical Input Area	First 512 digital inputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area	First 512 digital outputs
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%QX256.0 ...%QX511.15	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
12288 ... 65535	0x3000 ... 0xFFFF	%MX0.0 ... %MX3327.15	NOVRAM Retain memory

5.3.2.2 Coils schreiben mit FC5 und FC15:

750-842: Modbus-Addresses for FC5 and FC15			
Modbus-Address		**Memory**	**Description**
[dec]	**[hex]**	**Area**	
0 ... 511	0x0000 ... 0x01FF	Physical Output Area	max 512 digital outputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area	max 512 digital outputs
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
12288 ... 65535	0x3000 ... 0xFFFF	%MX0.0 ... %MX3327.15	NOVRAM Retain memory

5.3.3 Modbus Konfigurationsregister des 750-842

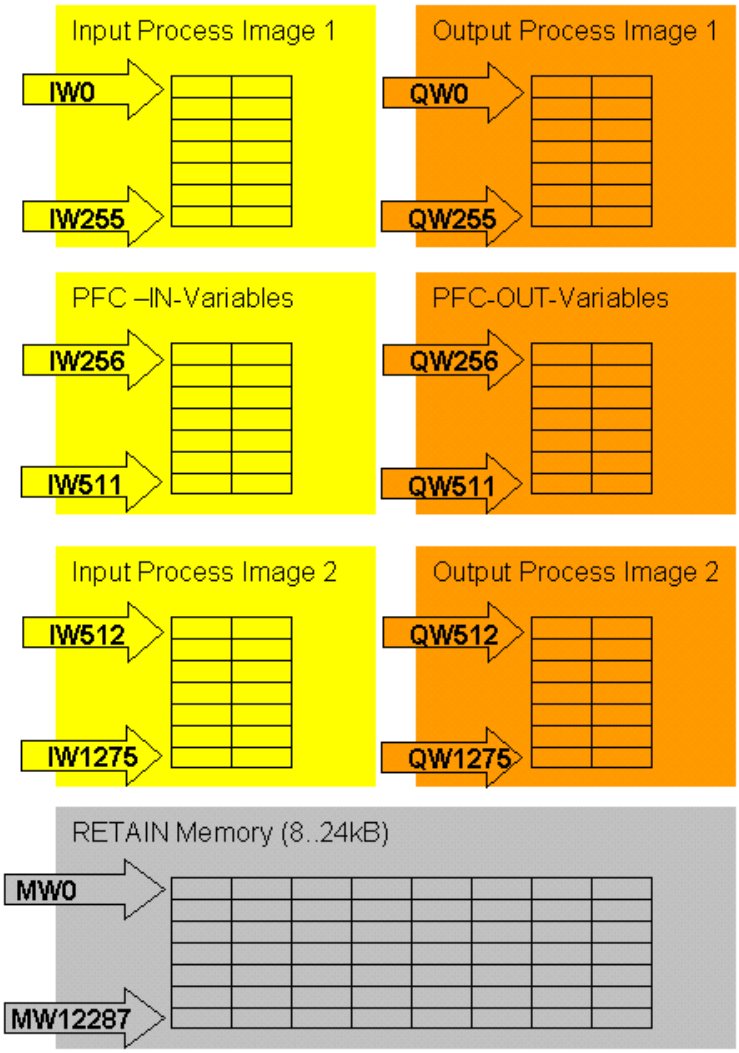
Über die Konfigurationsregister lassen sich die Eigenschaften des 750-842 ermitteln und teilweise verändern.

750-842: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
4096	0x1000	1	R/W	ModbusWatchdogTime (Vielfaches von 100ms)
4097	0x1001	1	R/W	ModbusWatchdogCodiermaske 1-16
4098	0x1002	1	R/W	ModbusWatchdogCodiermaske 17-32
4099	0x1003	1	R/W	ModbusWatchdogTrigger
4100	0x1004	1	R	Minimale Triggerzeit
4101	0x1005	1	R/W	ModbusWatchdog stoppen (0xAAAA und 0x5555)
4102	0x1006	1	R	ModbusWatchdog Status
4103	0x1007	1	R/W	ModbusWatchdog restart (0x0001)
4104	0x1008	1	R/W	ModbusWatchdog stoppen (0x55AA oder 0xAA55)
4105	0x1009	1	R/W	Modbus- und HTTP-Port schließen nach Timeout
4106	0x100A	1	R/W	ModbusWatchdog im „Modicon Mode“ starten
4107	0x100B	1	W	ModbusWatchdogParameter speichern
4128	0x1020	1	R	LED Error-Code
4129	0x1021	1	R	LED Error-Argument
4130	0x1022	1	R	Anzahl analoger Ausgänge im PA [Bit]
4131	0x1023	1	R	Anzahl analoger Eingänge im PA [Bit]
4132	0x1024	1	R	Anzahl digitaler Ausgänge im PA [Bit]
4133	0x1025	1	R	Anzahl digitaler Eingänge im PA [Bit]
4135	0x1027	1	R	Klemmenbuszyklus ausführen
4136	0x1028	1	R/W	IP-Configuration: BootP(1) or FIX(0)
4137	0x1029	18	R	Modbus-TCP-Statistik
4144	0x1030	1	R/W	Modbus Verbindungsüberwachung aktivieren
4145	0x1031	3	R	MAC-ID der Ethernetschnittstelle
4160	0x1040	1	R/W	Prozessdaten-Interface
8192	0x2000	1	R	0x0000 (Constant)
8193	0x2001	1	R	0xFFFF (Constant)
8194	0x2002	1	R	0x1234 (Constant)
8195	0x2003	1	R	0xAAAA (Constant)
8196	0x2004	1	R	0x5555 (Constant)
8197	0x2005	1	R	0x7FFF (Constant)
8198	0x2006	1	R	0x8000 (Constant)
8199	0x2007	1	R	0x3FFF (Constant)
8200	0x2008	1	R	0x4000 (Constant)
8208	0x2010	1	R	Firmware release
8209	0x2011	1	R	Seriencode (750)
8210	0x2012	1	R	Gerätecode (842)
8211	0x2013	1	R	Spezielle Firmwareversion (0xFFFF)
8212	0x2014	1	R	Spezielle Firmwareversion (0xFFFF)

750-842: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
8224	0x2020	1 .. 125	R	Gerätekurzbeschreibung
8225	0x2021	8	R	Compile-Zeit der Firmwareversion
8226	0x2022	8	R	Compile-datum der Firmwareversion
8227	0x2023	32	R	Version des Firmwareloaders (FWL)
8240	0x2030	65	R	Beschreibung angeschlossener IO-Module: 0-64
8245	0x2035	1	R/W	Einstellung Prozessabbild (Table 0 register 3)
8246	0x2036	1 .. 17	R	Diagnoseinformation Gerät
8256	0x2040	1	W	Software Reset (write 0x55AA or 0xAA55)
8260	0x2044	1	W	Delete Modbus Configurationfile (write 0x55AA)

5.4 Prozessabbild des 750-841

Der 750-841 kann maximal 15 eingehende Modbus-TCP-Verbindungen gleichzeitig bearbeiten. Der Modbus-Verbindungs-Watchdog ist im Auslieferungszustand aktiviert.



Neben der WAGO Grundausstattung an Modbus-Diensten unterstützt der 750-841 zusätzlich den Funktionscode FC22 „Mask write“.

FC	Name	Description
FC1	Read coils	Rücklesen mehrerer digitaler Ausgänge
FC2	Read inputs discrete	Lesen mehrerer digitaler Eingänge
FC3	Read holding registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC4	Read input registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC11	Get comm event counter	Kommunikationsereigniszähler
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analoger Ausgänge
FC22	Mask write	Manipulation einzelner Bits eines Registers
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

Mit den Funktionbausteinen „SET_DIGITAL_INPUT_OFFSET“ und „SET_DIGITAL_OUTPUT_OFFSET“ aus der Bibliothek „mod_com.lib“ können die Startadressen der ersten digitalen I/O-Module an WAGO Ethernet Controllern fest vorgegeben werden. Dies erlaubt Platz für spätere Erweiterungen einzuplanen. Die Angabe des OFFSET’s erfolgt in Bytes.
Jedoch ist die Wirksamkeit auf die SPS begrenzt. Bis Firmwarerelease(11) ignoriert der Modbus-Slave den digitalen Offset.

Memory area	Modbus-access	PLC access	Description
Physical. Input(1)	read	read	Physikalische Eingänge (%IW0 ... %IW255)
Physical Output(1)	read/[write][*1)]	read/[write][*1)]	Physikalische Ausgänge (%QW0 ... %QW255)
PFC-IN-	read/write	read	Flüchtige SPS-Eingangsvariablen (%IW256 ... %IW511)
PFC-OUT	read	read/write	Flüchtige SPS-Ausgangsvariablen (%QW256 ... %QW511)
Configuration register	read/(write)	---	Konfigurationsregister
NOVRAM Retain memory	read/write	read/write	8kB remanenter Speicher (max 24kB) (%MW0 ... %MW4095)
Physical. Input(2)	read	read	Physikalische Eingänge (%IW512 ... %IW1275)
Physical Output(2)	read/[write][*1)]	read/[write][*1)]	Physikalische Ausgänge (%QW512 ... %QW1275)

[][*1)] Schreibrecht wird festgelegt durch die Datei “/etc/EA-conf.xml”

Eine weitere Besonderheit des 750-841 ist es, das jedem I/O-Modul eine Schreibberechtigung zugeordnet werden kann bzw. muss.
Die physikalischen Ausgänge lassen sich entweder über Modbus-Dienste oder durch das SPS-Programm verändern. Die Schreibberechtigung wird durch die Existenz bzw. dem Inhalt der Datei „/etc/EA-conf.xml“ festgelegt.
Fehlt die Datei oder stimmt die Anzahl der konfigurierten I/O-Module nicht mit der tatsächlichen Zahl angeschlossener I/O-Module überein, dann wird die Schreibberechtigung dem Modbus-Diensten zugeordnet.
Die Datei „/etc/EA-conf.xml“ wird automatisch beim anlegen einer CoDeSys-Steuerungskonfiguration erzeugt und regelt das Schreibrecht auf I/O-Modulebene.
Ein Merksatz für dieses Verhalten könnte lauten:„Es kann nur einen geben“.

Im 750-841 sind „Merker-Variablen“ und „Retain-Variablen“ im 24kB großen NOVRAM untergebracht. Für die Merker sind im Auslieferungszustand 8kB, für die Retain-Variablen 16kB konfiguriert. Dadurch sind Überschneidungen wie im 750-842 nicht möglich.
Die Aufteilung des 24kB großen NOVRAM’s kann in den CoDeSys-Zielsystemeinstellungen verändert werden.

Bei Verwendung der SysLibSocket ist die maximale Anzahl von TCP-Socket-Verbindungen die aus einem SPS-Programm heraus erzeugt werden können nahezu unbegrenzt. In der „Ethernet.lib“ ist die maximale Anzahl von TCP-Socket-Verbindungen auf 5 begrenzt.

5.4.1 Register Dienste des 750-841

5.4.1.1 Register lesen mit FC3, FC4 und FC23:

750-841: Modbus- vs IEC61131-Addresses for FC3, FC4 and FC23			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%IW0 ... %IW255	Physical Input Area (1) First 256 Words of physical input data
256 ... 511	0x0100 ... 0x01FF	%QW256 ... %QW511	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW12287	NOVRAM 8kB retain memory (max. 24kB)
24576 ... 25340	0x6000 ... 0x62FC	%IW512 ... %IW1275	Physical Input Area (2) Additional 764 Words physical input data
25341 ... 28671	0x62FD ... 0x6FFF	-	Modbus Exception: “ Illegal data address”
28672 ... 29436	0x7000 ... 0x72FC	%QW512 ... %QW1275	Physical Output Area (2) Additional 764 Words physical output data
29437 ... 65535	0x72FD ... 0xFFFF	-	Modbus Exception: “ Illegal data address”

5.4.1.2 Register schreiben mit FC6, FC16, FC22 und FC23:

750-841: Modbus- vs IEC61131-Addresses for FC6, FC16, FC22 and FC23			
Modbus-Address		IEC61131	Description
[dec]	[hex]	Address	
0 ... 255	0x0000 ... 0x00FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data
256 ... 511	0x0100 ... 0x01FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
512 ... 767	0x0200 ... 0x02FF	%QW0 ... %QW255	Physical Output Area (1) First 256 Words of physical output data
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW12287	NOVRAM 8kB retain memory (max. 24kB)
24576 ... 25340	0x6000 ... 0x62FC	%QW512 ... %QW1275	Physical Output Area (2) Additional 764 Words physical output data
25341 ... 28671	0x62FD ... 0x6FFF	-	Modbus Exception: “ Illegal data address”
28672 ... 29436	0x7000 ... 0x72FC	%QW512 ... %QW1275	Physical Output Area (2) Additional 764 Words physical output data
29437 ... 65535	0x72FD ... 0xFFFF	-	Modbus Exception: “ Illegal data address”

5.4.2 Digitale Modbus-Dienste des 750-841

Mit den digitalen Modbus-Diensten lassen ausschließlich die Zustände von digitalen I/O-Modulen ermitteln oder verändern. Komplexe I/O-Module werden ignoriert bzw. sind unerreichbar.

5.4.2.1 Coils lesen mit FC1 und FC2:

750-841: Modbus-Addresses for FC1 and FC2			
Modbus-Address [dec]	Modbus-Address [hex]	Memory Area	Description
0 ... 511	0x0000 ... 0x01FF	Physical Input Area (1)	First 512 digital inputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area (1)	First 512 digital outputs
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%QX256.0 ...%QX511.15	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
12288 ... 32767	0x3000 ... 0x7FFF	%MX0.0 ... %MX1279.15	NOVRAM Retain-Area (8kB default)
32768 ... 34295	0x8000 ... 0x85F7	Physical Input Area (2)	Starts with the 513th and ends with the 2039th digital input
34296 ... 36863	0x85F8 ... 0x8FFF	-	Modbus Exception: “Illegal data address”
36864 ... 38391	0x9000 ... 0x95F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
38392 ... 65535	0x95F8 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.4.2.2 Coils schreiben mit FC5 und FC15:

750-841: Modbus-Addresses for FC5 and FC15			
Modbus-Address		Memory	Description
[dec]	[hex]	Area	
0 ... 511	0x0000 ... 0x01FF	Physical Output Area (1)	First 512 digital outputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area (1)	First 512 digital outputs
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
12288 ... 32767	0x3000 ... 0x7FFF	%MX0.0 ... %MX1279.15	NOVRAM Retain-Area
32768 ... 34295	0x8000 ... 0x85F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
34296 ... 36863	0x85F8 ... 0x8FFF	-	Modbus Exception: “Illegal data address”
36864 ... 38391	0x9000 ... 0x95F7	Physical Output Area (2)	Starts with the 513th and ends with the 2039th digital output
38392 ... 65535	0x95F8 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.4.3 Modbus Konfigurationsregister des 750-841

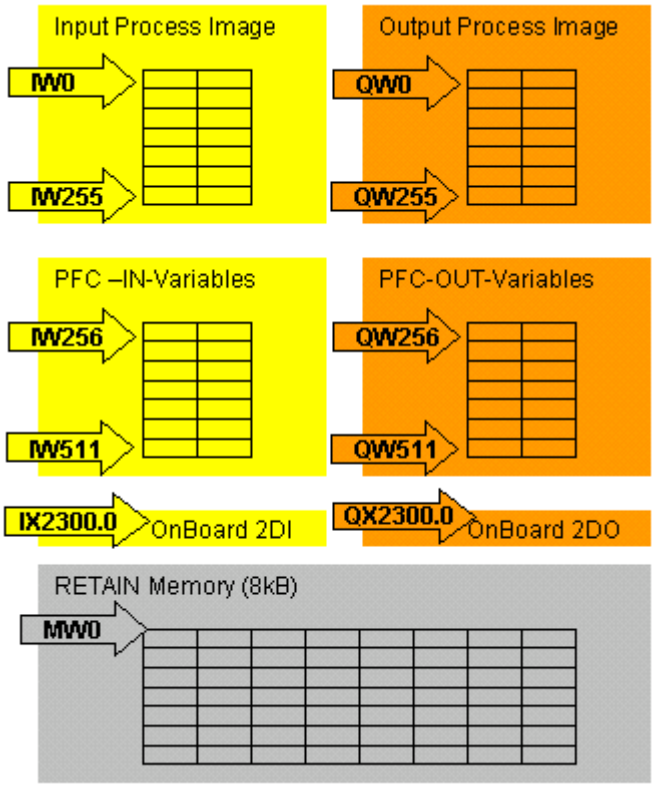
Über die Konfigurationsregister lassen sich die Eigenschaften des 750-841 ermitteln und teilweise verändern.

750-841: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
4096	0x1000	1	R/W	ModbusWatchdogTime (Vielfaches von 100ms)
4097	0x1001	1	R/W	ModbusWatchdogCodiermaske 1-16
4098	0x1002	1	R/W	ModbusWatchdogCodiermaske 17-32
4099	0x1003	1	R/W	ModbusWatchdogTrigger
4100	0x1004	1	R	Minimale Triggerzeit
4101	0x1005	1	R/W	ModbusWatchdog stoppen (0xAAAA und 0x5555)
4102	0x1006	1	R	ModbusWatchdog Status
4103	0x1007	1	R/W	ModbusWatchdog restart (0x0001)
4104	0x1008	1	R/W	ModbusWatchdog stoppen (0x55AA oder 0xAA55)
4105	0x1009	1	R/W	Modbus- und HTTP-Port schließen nach Timeout
4106	0x100A	1	R/W	ModbusWatchdog im „Modicon Mode“ starten
4107	0x100B	1	W	ModbusWatchdogParameter speichern
4128	0x1020	1	R	LED Error-Code
4129	0x1021	1	R	LED Error-Argument
4130	0x1022	1	R	Anzahl analoger Ausgänge im PA [Bit]
4131	0x1023	1	R	Anzahl analoger Eingänge im PA [Bit]
4132	0x1024	1	R	Anzahl digitaler Ausgänge im PA [Bit]
4133	0x1025	1	R	Anzahl digitaler Eingänge im PA [Bit]
4136	0x1028	1	R/W	IP-Configuration: BootP(1), DHCP(2) or FIX(4)
4137	0x1029	18	R	Modbus-TCP-Statistik
4138	0x102A	1	R	Anzahl aufgebauter Modbus-TCP Verbindungen
4144	0x1030	1	R/W	Modbus Verbindungsüberwachung aktivieren
4145	0x1031	3	R	MAC-ID der Ethernetschnittstelle
4176	0x1050	3	R	Diagnoseinformationen angeschlossener IO-Module
8192	0x2000	1	R	0x0000 (Constant)
8193	0x2001	1	R	0xFFFF (Constant)
8194	0x2002	1	R	0x1234 (Constant)
8195	0x2003	1	R	0xAAAA (Constant)
8196	0x2004	1	R	0x5555 (Constant)
8197	0x2005	1	R	0x7FFF (Constant)
8198	0x2006	1	R	0x8000 (Constant)
8199	0x2007	1	R	0x3FFF (Constant)
8200	0x2008	1	R	0x4000 (Constant)
8208	0x2010	1	R	Firmware release
8209	0x2011	1	R	Seriencode (750)
8210	0x2012	1	R	Gerätecode (841)
8211	0x2013	1	R	Major Firmwareversion
8212	0x2014	1	R	Minor Firmwareversion

750-841: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
8224	0x2020	1 .. 125	R	Gerätekurzbeschreibung
8225	0x2021	8	R	Compile-Zeit der Firmwareversion
8226	0x2022	8	R	Compile-datum der Firmwareversion
8227	0x2023	32	R	Version des Firmwareloaders (FWL)
8240	0x2030	65	R	Beschreibung angeschlossener IO-Module: 0-64
8241	0x2031	64	R	Beschreibung angeschlossener IO-Module: 65-129
8242	0x2032	64	R	Beschreibung angeschlossener IO-Module: 130-194
8243	0x2033	63	R	Beschreibung angeschlossener IO-Module: 195-255
8245	0x2035	1	R/W	Einstellung Prozessabbild (Table 0 register 3)
8246	0x2036	1 .. 17	R	Diagnoseinformation Gerät
8256	0x2040	1	W	Software Reset (write 0x55AA or 0xAA55)
8257	0x2041	1	W	Format Flash-
8258	0x2042	1	W	Extract filesystem
8260	0x2044	1	W	Delete Modbus Configurationfile (write 0x55AA)

5.5 Prozessabbild des 758-870

Der Modbus-Slave(Server) im 758-870 kann maximal 10 gleichzeitige Modbus-TCP-Verbindungen bearbeiten.



Ab dem Firmware-Image „2.4.31/0105“ werden die folgenden Modbus-Dienste unterstützt.

FC	Name	Description
FC1	Read coils	Rücklesen mehrerer digitaler Ausgänge
FC2	Read inputs discrete	Lesen mehrerer digitaler Eingänge
FC3	Read holding registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC4	Read input registers	Lesen mehrerer analoger Eingänge(und Ausgänge)
FC5	Write coil	Schreiben eines einzelnen digitalen Ausgangs
FC6	Write single register	Schreiben eines einzelnen analogen Ausgangs
FC11	Get comm event counter	Kommunikationsereigniszähler
FC15	Force multiple coils	Schreiben mehrerer digitaler Ausgänge
FC16	Write multiple registers	Schreiben mehrerer analoger Ausgänge
FC23	Read/write multiple registers	Schreib-Lese-Operation auf analoge Ein/Ausgänge

Memory area	Modbus-access	PLC access	Description
Physical. Input(1)	read	read	Physikalische Eingänge (%IW0 ... %IW255)
Physical Output(1)	[read]*1)/ write	read/ write	Physikalische Ausgänge (%QW0 ... %QW255)
PFC-IN-	read/write	read	Flüchtige SPS-Eingangsvariablen (%IW256 ... %IW511)
PFC-OUT	read	read/write	Flüchtige SPS-Ausgangsvariablen (%QW256 ... %QW511)
Merker Flag variables	read/write	read/write	8kB remanenter Speicher im SRAM (%MW0 ... %MW4095) Deklariert mit “AT %M...” Statement
On-Bord digital Input (OB2DI)	read	read	Die zwei digitalen „OnBoard“ Eingänge (%IX2300.0 und %IX2300.1)
On-Bord digital Output (OB2DO)	[read]*1)/ write	read/ write	Die zwei digitalen „OnBoard“ Ausgänge (%QX2300.0 und %IQ2300.1)

*1) Im Image „2.4.31/0105“ nicht unterstützt

Für die ordnungsgemäße Funktion der Modbus-Dienste ist architekturbedingt ein SPS-Programm im 758-870 zwingend erforderlich.
Fehlt das SPS-Programm, werden alle Modbus-Dienste auf die Speicherbereiche PFC-IN, PFC-OUT und Merker mit dem Modbus-Eceptioncode „ILLEGAL_RESPONSE_LENGTH“ beantwortet.

Nur Variablen die mit dem Statement „AT %...“- deklarierten wurden sind über Modbus-Dienste erreichbar. Variablen die in einem VAR RETAIN Block deklariert wurden sind über Modbus-Dienste nicht erreichbar.

5.5.1 Register Dienste des 758-870

5.5.1.1 Register lesen mit FC3, FC4 und FC23:

758-870: Modbus- vs IEC61131-Addresses for FC3, FC4 and FC23			
Modbus-Address [dec]	[hex]	IEC61131 Address	Description
0 ... 255	0x0000 ... 0x00FF	%IW0 ... %IW255	Physical Input Area (1) 256 Words of physical input data
256 ... 511	0x0100 ... 0x01FF	%QW256 ... %QW511	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
512 ... 767	0x0200 ... 0x02FF	-	Modbus Exception: “Illegal data address”
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW12287	Retain-Area (8kB) Nichtflüchtige SPS-Variablen
24576 ... 65535	0x6000 ... 0xFFFF	-	Modbus Exception: “ Illegal data address”

5.5.1.2 Register schreiben mit FC6, FC16 und FC23:

758-870: Modbus- vs IEC61131-Addresses for FC6, FC16, and FC23			
Modbus-Address [dec]	[hex]	IEC61131 Address	Description
0 ... 255	0x0000 ... 0x00FF	%QW0 ... %QW255	Physical Output Area 256 Words of physical Output data
256 ... 511	0x0100 ... 0x01FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
512 ... 767	0x0200 ... 0x02FF	-	Modbus Exception: “Illegal data address”
768 ... 1023	0x0300 ... 0x03FF	%IW256 ... %IW511	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
1024 ... 4095	0x0400 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 12287	0x1000 ... 0x2FFF	-	Configuration Register (see manual for details)
12288 ... 24575	0x3000 ... 0x5FFF	%MW0 ... %MW12287	Retain-Area (8kB) Nichtflüchtige SPS-Variablen
24576 ... 65535	0x6000 ... 0xFFFF	-	Modbus Exception: “ Illegal data address”

5.5.2 Digitale Modbus-Dienste des 758-870

Mit den digitalen Modbus-Diensten lassen sich ausschließlich die Zustände von digitalen I/O-Modulen ermitteln bzw. verändern. Komplexe I/O-Module werden ignoriert bzw. sind unerreichbar.

5.5.2.1 Coils lesen mit FC1 und FC2:

758-870: Modbus-Addresses for FC1 and FC2			
Modbus-Address [dec]	Modbus-Address [hex]	Memory Area	Description
0 ... 511	0x0000 ... 0x01FF	Physical Input Area	physical digital inputs
512 ... 1023	0x0200 ... 0x03FF	-	Modbus Exception: “Illegal data address”
1024 ... 1025	0x0400 ... 0x0401	%IX2300.0 ... %IX2003.1	On-Board digital Input
1026 ... 4095	0x0402 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%QX256.0 ...%QX511.15	PFC-OUT-Area Flüchtige SPS-Ausgangsvariablen
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
12288 ... 32767	0x3000 ... 0x7FFF	%MX0.0 ...%MX1279.15	Retain-Area (8kB) Nichtflüchtige SPS-Variablen
32768 ... 65535	0x8000 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.5.2.2 Coils schreiben mit FC5 und FC15:

758-870: Modbus-Addresses for FC5 and FC15			
Modbus-Address [dec]	Modbus-Address [hex]	Memory Area	Description
0 ... 511	0x0000 ... 0x01FF	Physical Output Area	physical 512 digital Outputs
512 ... 1023	0x0200 ... 0x03FF	Physical Output Area	physical 512 digital Outputs
1024 ... 1025	0x0400 ... 0x0401	%QX2300.0 ... %QX2003.1	On-Board digital Output
1026 ... 4095	0x0402 ... 0x0FFF	-	Modbus Exception: “Illegal data address”
4096 ... 8191	0x1000 ... 0x1FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
8192 ... 12287	0x2000 ... 0x2FFF	%IX256.0 ...%IX511.15	PFC-IN-Area Flüchtige SPS-Eingangsvariablen
12288 ... 32767	0x3000 ... 0x7FFF	%MX0.0 ...%MX1279.15	Retain-Area (8kB) Nichtflüchtige SPS-Variablen
32768 ... 65535	0x8000 ... 0xFFFF	-	Modbus Exception: “Illegal data address”

5.5.3 Modbus Konfigurationsregister des 758-870

Über die Konfigurationsregister lassen sich die Eigenschaften des 758-870 ermitteln und teilweise verändern.

758-870: Modbus Configuration Register for FC3, FC4, FC6 and FC16				
Modbus-Address		Length	Access	Description
[dec]	[hex]	[Word]		
4128	0x1020	1	R	LED Error-Code
4129	0x1021	1	R	LED Error-Argument
4130	0x1022	1	R	Anzahl analoger Ausgänge im PA [Bit]
4131	0x1023	1	R	Anzahl analoger Eingänge im PA [Bit]
4132	0x1024	1	R	Anzahl digitaler Ausgänge im PA [Bit]
4133	0x1025	1	R	Anzahl digitaler Eingänge im PA [Bit]
4136	0x1028	1	R/W	IP-Configuration: BootP(1), DHCP(2) or FIX(4)
4144	0x1030	1	R/W	Enable Modbus-Connection-Watchdog
4145	0x1031	3	R	ETH Interface X9: MAC-ID
4146	0x1032	3	R	ETH Interface X8: MAC-ID
8192	0x2000	1	R	0x0000 (Constant)
8193	0x2001	1	R	0xFFFF (Constant)
8194	0x2002	1	R	0x1234 (Constant)
8195	0x2003	1	R	0xAAAA (Constant)
8196	0x2004	1	R	0x5555 (Constant)
8197	0x2005	1	R	0x7FFF (Constant)
8198	0x2006	1	R	0x8000 (Constant)
8199	0x2007	1	R	0x3FFF (Constant)
8200	0x2008	1	R	0x4000 (Constant)
8213	0x2015	1	R	Modbus-Server Version
8256	0x2040	1	W	Software Reset (write 0x55AA or 0xAA55)
8260	0x2044	1	W	Delete Modbus Configurationfile (write 0x55AA)

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: http://www.wago.com