



# ibaPDA-Interface-Modbus-TCP-Client

## Datenschnittstelle Modbus-TCP-Client

Handbuch  
Ausgabe 3.0

Messsysteme für Industrie und Energie  
[www.iba-ag.com](http://www.iba-ag.com)

---

## Hersteller

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland

## Kontakte

Zentrale	+49 911 97282-0
Telefax	+49 911 97282-33
Support	+49 911 97282-14
Technik	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts sind nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz.

© iba AG 2023, alle Rechte vorbehalten.

Der Inhalt dieser Druckschrift wurde auf Übereinstimmung mit der beschriebenen Hard- und Software überprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass für die vollständige Übereinstimmung keine Garantie übernommen werden kann. Die Angaben in dieser Druckschrift werden jedoch regelmäßig aktualisiert. Notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten oder können über das Internet heruntergeladen werden.

Die aktuelle Version liegt auf unserer Website [www.iba-ag.com](http://www.iba-ag.com) zum Download bereit.

Version	Datum	Revision	Autor	Version SW
3.0	10-2023	Neue Version ibaPDA v8	RM/IP	8.4.0

Windows® ist eine Marke und eingetragenes Warenzeichen der Microsoft Corporation. Andere in diesem Handbuch erwähnte Produkt- und Firmennamen können Marken oder Handelsnamen der jeweiligen Eigentümer sein.

## Inhalt

<b>1</b>	<b>Zu dieser Dokumentation .....</b>	<b>5</b>
1.1	Zielgruppe und Vorkenntnisse .....	5
1.2	Schreibweisen .....	5
1.3	Verwendete Symbole .....	6
<b>2</b>	<b>Systemvoraussetzungen .....</b>	<b>7</b>
<b>3</b>	<b>Datenschnittstelle Modbus-TCP-Client .....</b>	<b>8</b>
3.1	Allgemeine Informationen .....	8
3.1.1	Modbus TCP/IP .....	8
3.1.2	Modbus Datenmodell .....	9
3.1.3	Modbus Adressierung .....	9
3.1.4	Client/Server-Architektur .....	10
3.1.5	Modbus Protokoll .....	11
3.1.6	Modbus TCP/IP – Telegrammaufbau .....	13
3.1.6.1	Daten lesen .....	13
3.1.6.2	Daten schreiben .....	15
3.1.7	Referenzen .....	16
3.2	Konfiguration und Projektierung ibaPDA .....	17
3.2.1	Allgemeine Einstellungen .....	17
3.2.2	Allgemeine Einstellungen der Schnittstelle .....	18
3.2.3	Modul hinzufügen .....	19
3.2.3.1	Allgemeine Moduleinstellungen .....	20
3.2.3.2	Signalkonfiguration .....	23
3.2.4	Konfiguration ibaPDA-Ausgangsmodule .....	24
3.2.4.1	Allgemeine Moduleinstellungen ibaPDA-Ausgangsmodule .....	24
3.2.4.2	Signalkonfiguration .....	25
3.2.5	Moduldiagnose .....	26
3.2.6	Erfassung starten .....	27
<b>4</b>	<b>Diagnose .....</b>	<b>28</b>
4.1	Lizenz .....	28
4.2	Sichtbarkeit der Schnittstelle .....	28
4.3	Protokolldateien .....	29

---

4.4	Verbindungsdiagnose mittels PING .....	30
4.5	Überprüfung der Verbindung .....	31
4.6	Diagnosemodule.....	33
<b>5</b>	<b>Anhang .....</b>	<b>39</b>
5.1	Fehlerbehandlung.....	39
5.1.1	Telegrammaufbau.....	39
5.1.2	Modbus Fehlercodes .....	40
<b>6</b>	<b>Support und Kontakt .....</b>	<b>41</b>

# 1 Zu dieser Dokumentation

Diese Dokumentation beschreibt die Funktion und Anwendung der Software-Schnittstelle *ibaPDA-Interface-Modbus-TCP-Client*.

Diese Dokumentation ist eine Ergänzung zum *ibaPDA*-Handbuch. Informationen über alle weiteren Eigenschaften und Funktionen von *ibaPDA* finden Sie im *ibaPDA*-Handbuch bzw. in der Online-Hilfe.

## 1.1 Zielgruppe und Vorkenntnisse

Diese Dokumentation wendet sich an ausgebildete Fachkräfte, die mit dem Umgang mit elektrischen und elektronischen Baugruppen sowie der Kommunikations- und Messtechnik vertraut sind. Als Fachkraft gilt, wer auf Grund der fachlichen Ausbildung, Kenntnisse und Erfahrungen sowie Kenntnis der einschlägigen Bestimmungen die übertragenen Arbeiten beurteilen und mögliche Gefahren erkennen kann.

Im Besonderen wendet sich diese Dokumentation an Personen, die mit Projektierung, Test, Inbetriebnahme oder Instandhaltung von Speicherprogrammierbaren Steuerungen der unterstützten Fabrikate befasst sind. Für den Umgang mit *ibaPDA-Interface-Modbus-TCP-Client* sind folgende Vorkenntnisse erforderlich bzw. hilfreich:

- Betriebssystem Windows
- Grundkenntnisse *ibaPDA*
- Kenntnis von Projektierung und Betrieb des betreffenden Messgeräts/-systems

## 1.2 Schreibweisen

In dieser Dokumentation werden folgende Schreibweisen verwendet:

Aktion	Schreibweise
Menübefehle	Menü <i>Funktionsplan</i>
Aufruf von Menübefehlen	<i>Schritt 1 – Schritt 2 – Schritt 3 – Schritt x</i> Beispiel: Wählen Sie Menü <i>Funktionsplan – Hinzufügen – Neuer Funktionsblock</i>
Tastaturtasten	<Tastename> Beispiel: <Alt>; <F1>
Tastaturtasten gleichzeitig drücken	<Tastename> + <Tastename> Beispiel: <Alt> + <Strg>
Grafische Tasten (Buttons)	<Tastename> Beispiel: <OK>; <Abbrechen>
Dateinamen, Pfade	<i>Dateiname, Pfad</i> Beispiel: <i>Test.docx</i>

## 1.3 Verwendete Symbole

Wenn in dieser Dokumentation Sicherheitshinweise oder andere Hinweise verwendet werden, dann bedeuten diese:

---

### Gefahr!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die unmittelbare Gefahr des Todes oder der schweren Körperverletzung!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Warnung!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr des Todes oder schwerer Körperverletzung!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Vorsicht!



**Wenn Sie diesen Sicherheitshinweis nicht beachten, dann droht die mögliche Gefahr der Körperverletzung oder des Sachschadens!**

- Beachten Sie die angegebenen Maßnahmen.

---

### Hinweis



Hinweis, wenn es etwas Besonderes zu beachten gibt, wie z. B. Ausnahmen von der Regel usw.

---

### Tipp



Tipp oder Beispiel als hilfreicher Hinweis oder Griff in die Trickkiste, um sich die Arbeit ein wenig zu erleichtern.

---

### Andere Dokumentation



Verweis auf ergänzende Dokumentation oder weiterführende Literatur.

## 2 Systemvoraussetzungen

Folgende Systemvoraussetzungen sind für die Verwendung der Datenschnittstelle Modbus-TCP-Client erforderlich:

- Lizenz für *ibaPDA-Interface-Modbus-TCP-Client*
- *ibaPDA* v8.0.0 oder höher
- Netzwerkanschluss 10/100 Mbit

In der *ibaPDA*-Dokumentation finden Sie weitere Anforderungen an die Computer-Hardware und die unterstützten Betriebssysteme.

---

### Hinweis



Es wird empfohlen, die TCP/IP-Kommunikation auf einem separaten Netzwerksegment durchzuführen, um eine gegenseitige Beeinflussung durch sonstige Netzwerkkomponenten auszuschließen.

---

### Systemeinschränkungen

- Die maximale Länge einer Modbus TCP/IP-Nachricht ist auf 244 Bytes beschränkt.

### Lizenzen

Bestell-Nr.	Produktbezeichnung	Beschreibung
31.001022	ibaPDA-Interface-Modbus-TCP-Client	Erweiterungslizenz für ein ibaPDA-System um eine Modbus-TCP-Client-Schnittstelle Anzahl der Verbindungen: 64
31.101022	one-step-up-Interface-Modbus over TCP/IP-Client	Lizenz für die Erweiterung einer vorhandenen Schnittstelle um 64 weitere Modbus-TCP-Client-Verbindungen, maximal 3 zulässig

## 3 Datenschnittstelle Modbus-TCP-Client

### 3.1 Allgemeine Informationen

#### 3.1.1 Modbus TCP/IP

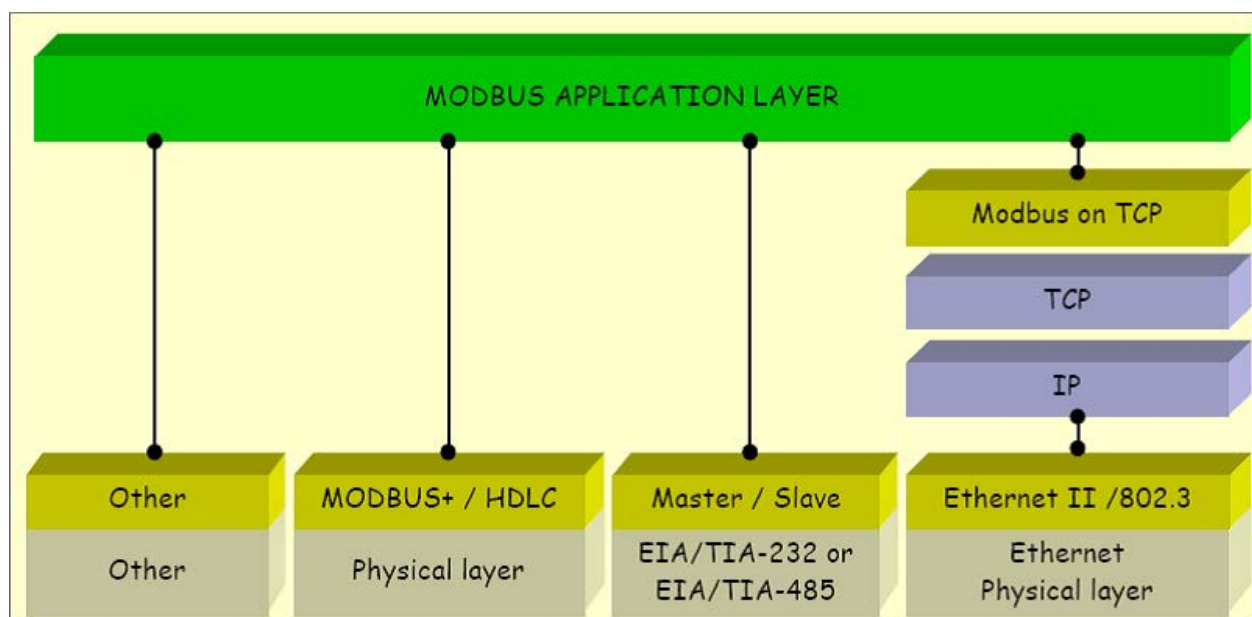
Das Transmission Control Protocol (TCP) ist eines der Kernprotokolle der Internetprotokoll-Familie.

IP arbeitet auf der unteren Vermittlungsschicht und ist für die Vermittlung von Nachrichten zwischen Rechnern im Internet zuständig. TCP ist auf einer höheren Schicht (Transportschicht) angeordnet und befasst sich mit den beiden Endsystemen. TCP sorgt für einen zuverlässigen Datenstrom von einem Programm auf einem Rechner zu einem anderen Programm auf einem anderen Rechner. TCP ist in RFC1180 und in RFC793 beschrieben (siehe [Referenzen](#), Seite 16).

Modbus ist ein Protokoll für eine Client/Server-Kommunikation zwischen Geräten an unterschiedlichen Bussen oder Netzwerken.

Modbus ist derzeit in den folgenden Bussen oder Netzwerken implementiert, wie in folgender Abbildung dargestellt:

- TCP/IP über Ethernet
- Asynchrone serielle Übertragung über unterschiedliche Medien
- Modbus PLUS (High-Speed-Kommunikation über ein Token-Passing-Netzwerk)



*ibaPDA* bietet die Möglichkeit, Signale mit dem Modbus-Protokoll über serielle Verbindungen (Modbus ASCII und Modbus RTU) und über TCP/IP zu messen. Dieses Handbuch beschreibt die Kommunikation über TCP/IP und als Variante die Übertragung des Modbus RTU Protokolls über TCP/IP, wobei *ibaPDA* als Client agiert.

Jedes System, das in der Lage ist, Nachrichten mit dem MODBUS-TCP-Protokoll als Server zu empfangen und zu beantworten, kann mit *ibaPDA* kommunizieren.



### 3.1.2 Modbus Datenmodell

Das Modbus-Datenmodell basiert auf verschiedenen Grundtypen mit unterschiedlichen Merkmalen. Die 4 Grundtypen sind:

Grundtypen	Objekttyp	Dienst	Kommentar
Discrete Inputs (Eingänge)	einzelne Bits	nur lesen	Daten, die beispielsweise von den E/A eines Geräts kommen.
Coils (Ausgänge)	einzelne Bits	lesen / schreiben	Bitinformationen, die durch das Anwenderprogramm änderbar sind.
Input Registers (Eingänge)	16-Bit Worte	nur lesen	Daten, die beispielsweise von den E/A eines Geräts kommen.
Holding Registers (Ausgänge)	16-Bit Worte	lesen / schreiben	Bitinformationen, die durch das Anwenderprogramm änderbar sind.

Für die Unterscheidung von Eingängen und Ausgängen, und Bit- und Wort-adressierten Dateneinheiten ist kein Applikationsprogramm erforderlich. Die vier Grundtypen können sich überlappen, sofern das Zielsystem diese richtig interpretiert.

Für jeden Grundtyp erlaubt das Protokoll die individuelle Auswahl von 65536 Dateneinheiten. Das Lesen oder Schreiben dieser Dateneinheiten kann mehrere aufeinanderfolgende Register umfassen bis zu einer maximalen Datengröße, die vom Funktionscode abhängt.

32-Bit Werte (DINT, FLOAT) sind in zwei aufeinanderfolgenden Registern abgelegt.

### 3.1.3 Modbus Adressierung

Die 4 Datentypen sind in unterschiedlichen, evtl. auch überlappenden Speicherbereichen abgelegt. Die Abbildung der Zugriffe auf die physikalische Speicheradresse übernimmt die Applikation auf dem Modbus-Server.

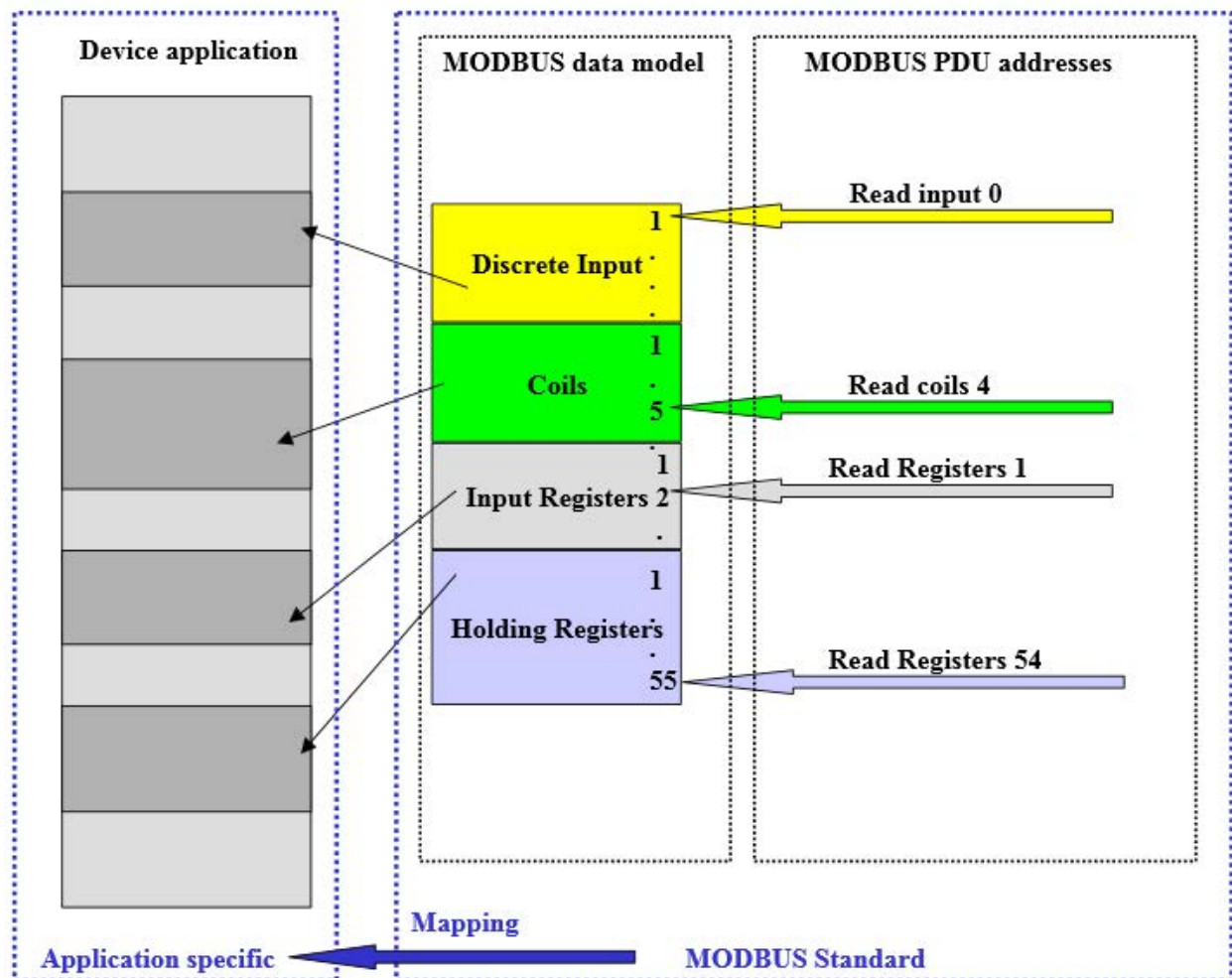
Modbus unterscheidet zwischen der internen Nummerierung (Discrete Inputs, Coils, Register), die generell mit 1 beginnt, und der Adressierung der Objekte, die mit 0 beginnt.

Beispiel: In vielen Modbus-Servern werden die Grundtypen auf folgende Adressbereiche abgebildet:

Coils:	0x00000
Inputs:	0x10000
Input Registers:	0x30000
Holding Registers:	0x40000

Das bedeutet, das Holding Register 1 ist im Speicher auf Adresse 0x40000 abgelegt. Der Zugriff auf Holding Register 1 erfolgt mit der logischen Referenznummer (Adresse) 0. Das Input Register 1 liegt im Speicher auf Adresse 0x30000 abgelegt. Der Zugriff auf Input Register 1 erfolgt ebenso mit Adresse 0.

Die logischen Modbus Referenznummern, die bei Modbus-Funktionen verwendet werden, sind vorzeichenlose, ganzzahlige Indizes, beginnend bei 0.

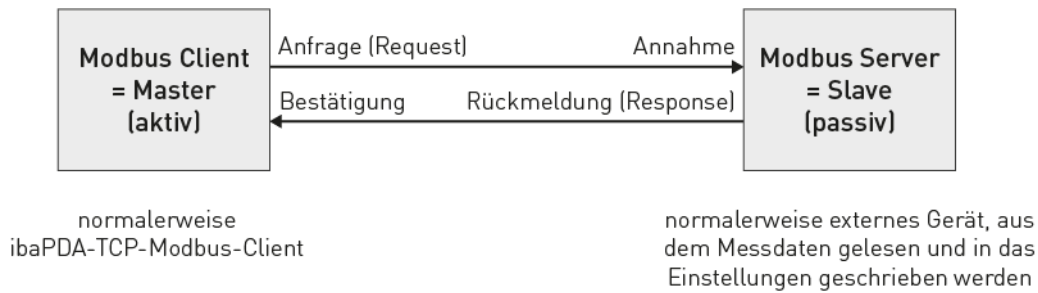


### 3.1.4 Client/Server-Architektur

Der Modbus-Dienst unterstützt eine Client/Server-Kommunikation für Geräte, die über Ethernet TCP/IP verbunden sind.

Das Client/Server-Modell basiert auf 4 Nachrichtentypen:

- Anfrage (Request)
- Annahme (Indication)
- Rückmeldung (Response)
- Bestätigung (Confirmation)



**Daten lesen:** Der Modbus-TCP-Client (*ibaPDA*) baut die Verbindung zu dem Modbus-Server auf, sendet periodisch die Anfrage und wartet auf die Rückmeldung, welche die angeforderten Daten enthält.

**Daten schreiben:** Der Modbus-TCP-Client (*ibaPDA*) baut die Verbindung zu dem Modbus-Server auf, sendet die Anfrage, welche die Ausgabedaten enthält, und wartet auf die Rückmeldung.

Standardmäßig wird für die Modbus-TCP/IP-Kommunikation der Port 502 verwendet, jedoch besteht in *ibaPDA* die Möglichkeit, andere Portnummern einzugeben.

Mit einer *ibaPDA-Interface-Modbus-TCP-Client*-Lizenz kann *ibaPDA* bis zu 64 Verbindungen aufbauen, d.h. es sind Verbindungen zu 64 Modbus-Servern möglich. Die Anzahl kann durch mehrfaches Laden der Lizenz auf max. 256 Verbindungen erweitert werden.

### 3.1.5 Modbus Protokoll

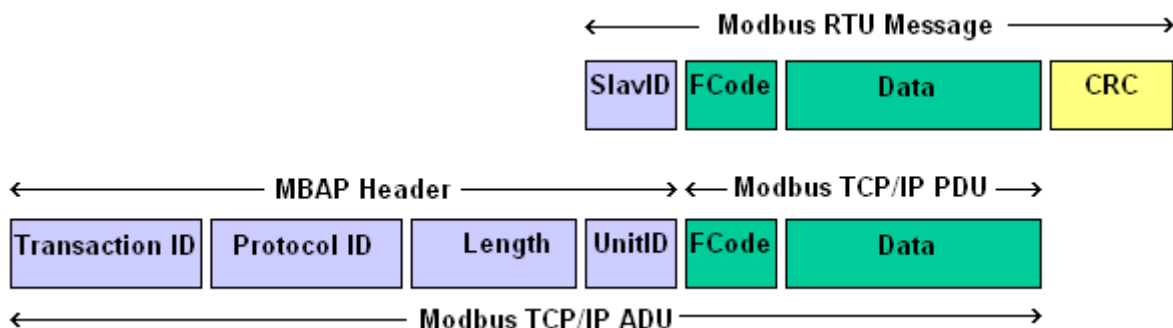
#### Byte-Reihenfolge

Modbus verwendet "BIG ENDIAN", d.h. in den Telegrammen werden die höherwertigen Bytes zuerst gesendet und folglich in den niederwertigen Adressen im Speicher abgelegt.

*ibaPDA* wandelt alle empfangenen 16- und 32-Bit-Werte in das Intel-Format "LITTLE ENDIAN" um ("Swapping"). Für Daten, die nicht von einer ursprünglichen Modbus-Steuerung kommen, bietet *ibaPDA* die Möglichkeit, die Swapping-Methode einzustellen. Siehe [Allgemeine Modulleistungen](#), Seite 20.

#### Modbus RTU / Modbus TCP

Die folgende Darstellung zeigt den grundsätzlichen Aufbau des Modbus-Protokolls und die Unterschiede zwischen Modbus RTU und Modbus TCP.



RTU	Remote Terminal Unit
MBAP	Modbus Application Protocol
ADU	Application Data Unit
PDU	Protocol Data Unit

Bei Modbus-TCP wird dem Funktionscode der MBAP Header vorangestellt, die Unit-ID entspricht dem Slave-ID des RTU-Protokolls und der CRC-Code entfällt.

Es gibt auch eine Variante "Modbus RTU over TCP/IP". Dabei wird die Modbus RTU-Message über TCP/IP übertragen. Dies wird beim Einsatz von Gateways Seriell/Ethernet verwendet.

### MBAP Header

Der MBAP Header wird bei der Kommunikation mit TCP/IP verwendet, um die Modbus Applikationsdaten zu identifizieren. Der Header enthält folgende Felder:

Felder	Bytes	Beschreibung
Transaktions-ID	2	Kennzeichnung einer Modbus Request/Response Aktion
Protokoll-ID	2	0 = Modbus-Protokoll
Länge	2	Anzahl der folgenden Bytes
Unit-ID	1	Adressierung eines Remote-Slaves der mit dem Modbus-Server verbunden ist.

- Transaktions-ID: Kennzeichnet die Zuordnung der Aktionen. Wird vom Modbus-Client in der Anfrage (request) gesendet, der Modbus-Server kopiert die Transaktions-ID in das Antwort-Telegramm (response).
- Protokoll-ID: Wird bei Multiplex-Verfahren verwendet, das Modbus-Protokoll hat den Wert 0.
- Länge: Gibt die Anzahl der folgenden Bytes wieder, inklusive Unit-ID, Funktionscode und Datenfelder.
- Unit-ID (Geräteadresse): Das Feld wird vom Modbus-Client in der Anfrage (request) gesendet und vom Server mit demselben Wert in seiner Antwort zurückgeschickt. Dieses Feld wird zur Kommunikation zu einem Modbus-Slave, der seriell mit dem Modbus-Server verbunden ist, verwendet.

### Funktionscode:

Ein Byte enthält den Funktionscode, der festlegt, welche Funktion aufgrund eines Requests durch den Server ausgeführt werden soll.

Der *ibaPDA Modbus-TCP-Client*-Treiber unterstützt die Funktionen:

- 0x01: Read Coils (Ausgänge lesen)
- 0x02: Read Discrete Inputs (Eingänge lesen)
- 0x03: Read Holding Registers (Halteregister lesen)
- 0x04: Read Input Registers (Eingaberegister lesen)
- 0x0F: Write Multiple Coils
- 0x10: Write Multiple Registers

## Datenfelder

Der Nutzdatenbereich ist in mehrere Unterbereiche unterteilt, wie Startadresse, Anzahl der Register, Anzahl der Bytes und die aktuellen Daten. Der Inhalt dieser Felder hängt vom verwendeten Funktionscode ab.

### 3.1.6 Modbus TCP/IP – Telegrammaufbau

Das Request-Telegramm hat das gleiche Layout für alle Zugriffsarten. Der anfordernde Controller (*ibaPDA* als Modbus-Client) bestimmt die Modbus Slave-Nummer (Unit-ID), die Zugriffsart (Funktionscode), die Startadresse (Adresse der ersten Daten) und die Anzahl der Daten.

#### 3.1.6.1 Daten lesen

Request *ibaPDA* -> Modbus Server:

	Offs	Bytes	Typ	Modbus Beschreibung	Inhalte (hex)	ibaPDA Beschreibung
MBAP	00	2	UINT	Transaktions-ID	xx xx	Wird automatisch mit jedem Zyklus inkrementiert
	02	2	UINT	Protokoll-ID	00 00	0
	04	2	UINT	Cmd Länge	00 06	6
	06	1	BYTE	Unit-ID	xx	Modbus Slave-Nummer
Fcode	07	1	BYTE	Funktionscode	xx	01: Read Coils 02: Read Discrete Inputs 03: Read Holding Register 04: Read Input Register
Daten	08	2	UINT	Startadresse	xx xx	1. Adresse
	10	2	UINT	Anzahl der Daten	xx xx	Anzahl der Coils, Input Bits, Holding Registers oder Input Registers

Response Modbus -> Server *ibaPDA*:

	Offs	Bytes	Typ	Modbus Beschreibung	Inhalt (hex)	ibaPDA Beschreibung
MBAP	00	2	UINT	Transaktions-ID	xx xx	Spiegel der Anfrage
	02	2	UINT	Protokoll-ID	00 00	0
	04	2	UINT	Cmd Länge	00 07	= nBytes + 3
	06	1	BYTE	Unit-ID	xx	Spiegel der Anfrage
Fcode	07	1	BYTE	Funktionscode	xx	Spiegel der Anfrage
Daten	08	1	BYTE	Anzahl der Bytes	xx	nBytes
	09	n	BYTE	Daten	xx xx	Eingabewerte

**Beispiele:**

Funktion 01: Read Coils			
Request	(hex)	(hex)	Response
Trans-ID	00 01	00 01	Trans-ID
Prot-ID	00 00	00 00	Prot-ID
Cmd LEN	00 06	00 06	Cmd LEN
Unit-ID	00	00	Unit-ID
FCode	01	01	FCode
Start Adr (Hi)	00	03	Anz. Byte
Start Adr (Lo)	13	xx	Coils 27-20 <sup>1)</sup>
Anz. Werte (Hi)	00	xx	Coils 35-28
Anz. Werte (Lo)	14	xx	Coils 39-36

Funktion 02: Read Discrete Inputs			
Request	(hex)	(hex)	Response
Trans-ID	00 02	00 02	Trans-ID
Prot-ID	00 00	00 00	Prot-ID
Cmd LEN	00 06	00 05	Cmd LEN
Unit-ID	00	00	Unit-ID
FCode	02	02	FCode
Start Adr (Hi)	00	02	Anz. Byte
Start Adr (Lo)	C4	xx	Inputs 204-197
Anz. Werte (Hi)	00	xx	Inputs 206-205
Anz. Werte (Lo)	0A		

Funktion 03: Read Holding Registers			
Request	(hex)	(hex)	Response
Trans-ID	00 03	00 03	Trans-ID
Prot-ID	00 00	00 00	Prot-ID
Cmd LEN	00 06	00 0B	Cmd LEN
Unit-ID	00	00	Unit-ID
FCode	03	03	FCode
Start Adr (Hi)	00	08	Anz. Byte
Start Adr (Lo)	6B	xx xx	Register 108 (Hi, Lo)
Anz. Werte (Hi)	00	xx xx	Register 109 (Hi, Lo)
Anz. Werte (Lo)	04	xx xx	Register 110 (Hi, Lo)
		xx xx	Register 111 (Hi, Lo)

<sup>1)</sup> Bits werden immer in der Reihenfolge MSB links, LSB rechts übertragen und gespeichert. Im letzten Byte werden die restlichen Bits (MSB, links) mit 0 gefüllt.

### 3.1.6.2 Daten schreiben

#### Request ibaPDA -> Modbus Server:

	Offs	Bytes	Typ	Modbus Beschreibung	Inhalte (hex)	ibaPDA Beschreibung
MBAP	00	2	UINT	Transaktions-ID	xx xx	Wird automatisch mit jedem Zyklus inkrementiert
	02	2	UINT	Protokoll-ID	00 00	0
	04	2	UINT	Cmd Länge	xx xx	nBytes + 7
	06	1	BYTE	Unit-ID	xx	Modbus Slave-Nummer
Fcode	07	1	BYTE	Funktionscode	xx	0F: Write Multiple Coils 10: Write Multiple Registers
Daten	08	2	UINT	Startadresse	xx xx	1. Adresse
	10	2	UINT	Anzahl Werte	xx xx	Anzahl der Coils oder Holding Registers
	12	1	BYTE	Anzahl Bytes	xx	
	13	n	BYTE	Daten	xx xx	Ausgabewerte

#### Response Modbus -> Server ibaPDA:

	Offs	Bytes	Typ	Modbus Beschreibung	Inhalt (hex)	ibaPDA Beschreibung
MBAP	00	2	UINT	Transaktions-ID	xx xx	Spiegel der Anfrage
	02	2	UINT	Protokoll-ID	00 00	0
	04	2	UINT	Cmd Länge	00 06	
	06	1	BYTE	Unit-ID	xx	Spiegel der Anfrage
Fcode	07	1	BYTE	Funktionscode	xx	Spiegel der Anfrage
Daten	08	2	UINT	Startadresse	xx xx	Spiegel der Anfrage
	09	4	UINT	Anzahl Werte	xx xx	Spiegel der Anfrage

#### Beispiele:

Funktion 0F: Write Multiple Coils			
Request	(hex)	(hex)	Response
Trans-ID	00 05	00 05	Trans-ID
Prot-ID	00 00	00 00	Prot-ID
Cmd LEN	00 09	00 06	Cmd LEN
Unit-ID	01	01	Unit-ID
FCode	0F	0F	FCode
Start Adr (Hi)	00	00	Start Adr (Hi)
Start Adr (Lo)	03	00	Start Adr (Lo)
Anz. Werte (Hi)	00	00	Anz. Werte (Hi)

Funktion 0F: Write Multiple Coils			
Anz. Werte (Lo)	0A	03	Anz. Werte (Lo)
Anz. Bytes	02		
Coils 11-8	xx		
Coils 13-12	xx		

Funktion 10: Write Multiple Registers			
Request	(hex)	(hex)	Response
Trans-ID	00 06	00 06	Trans-ID
Prot-ID	00 00	00 00	Prot-ID
Cmd LEN	00 13	00 06	Cmd LEN
Unit-ID	02	02	Unit-ID
FCode	10	10	FCode
Start Adr (Hi)	00	00	Start Adr (Hi)
Start Adr (Lo)	20	20	Start Adr (Lo)
Anz. Werte (Hi)	00	00	Anz. Werte (Hi)
Anz. Werte (Lo)	06	06	Anz. Werte (Lo)
Anz. Bytes	0C		
Register 33	xx xx		
Register 34	xx xx		
Register 35	xx xx		
Register 36	xx xx		
Register 37	xx xx		
Register 38	xx xx		

### 3.1.7 Referenzen

#### Andere Dokumentation



- ibaPDA-Handbuch
- A TCP/IP tutorial, RFC1180 (<ftp://ftp.ripe.net/rfc/rfc1180.txt>)
- Transmission Control Protocol, RFC793 (<ftp://ftp.ripe.net/rfc/rfc793.txt>)
- Modbus Messaging Implementation Guide V1 ( <http://www.modbus.org>)
- Modbus Application Protocol V1.1( <http://www.modbus.org>)
- Modbus Protocol Reference Guide Rev J, Modicon

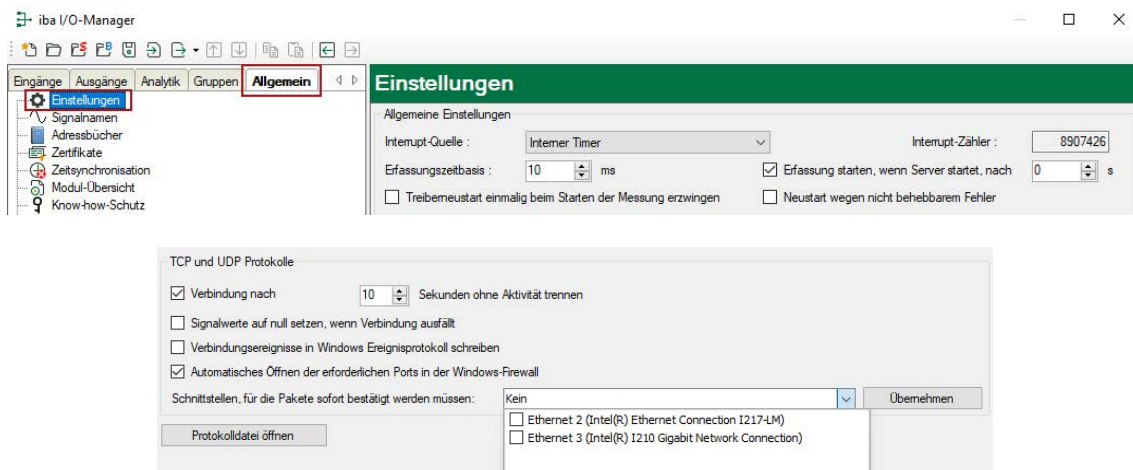


## 3.2 Konfiguration und Projektierung ibaPDA

Nachfolgend ist die Projektierung in *ibaPDA* beschrieben. Wenn alle Systemvoraussetzungen erfüllt sind, bietet *ibaPDA* im Schnittstellenbaum des I/O-Managers die Schnittstelle *Modbus TCP Client* an.

### 3.2.1 Allgemeine Einstellungen

Die "Totmann-Timeout" Konfiguration erfolgt für alle von *ibaPDA* unterstützen TCP- und UDP-Protokolle gemeinsam.



#### Verbindung nach ... Sekunden ohne Aktivität trennen

Verhalten und Timeout-Zeit ist vorgebbar.

#### Signalwerte auf null setzen, wenn Verbindung ausfällt

Wenn deaktiviert, bleibt der zuletzt gelesene Wert erhalten.

#### Verbindungsereignisse in Windows Ereignisprotokoll schreiben

Aktuelle Ereignisse werden in Windows protokolliert.

#### Automatisches Öffnen der erforderlichen Ports in der Windows-Firewall

Wenn aktiviert, werden vom *ibaPDA*-Server-Dienst alle Ports, die für die aktuell lizenzierten Schnittstellen benötigt werden, automatisch in der Firewall freigeschaltet.

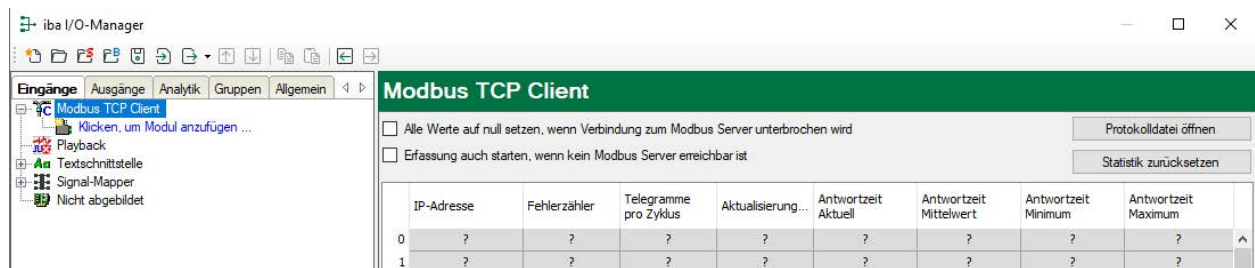
Wenn deaktiviert, können die benötigten Ports im I/O-Manager der lizenzierten Schnittstellen über <Port in Firewall zulassen> freigeschaltet werden.

#### Schnittstellen, für die Pakete sofort bestätigt werden müssen

Auswahl der erforderlichen Schnittstellen

### 3.2.2 Allgemeine Einstellungen der Schnittstelle

Die Schnittstelle hat folgende Funktionen und Konfigurationsmöglichkeiten:



#### Alle Werte auf null setzen, wenn Verbindung zum Modbus Server unterbrochen wird

Wenn nicht aktiviert, bleiben die letzten erfassten Werte stehen.

#### Erfassung auch starten, wenn kein Modbus Server erreichbar ist

Standardmäßig kann die Erfassung nicht gestartet werden, wenn ein Modbus-Server nicht erreichbar ist. Ist diese Option ausgewählt, wird die Erfassung mit Warnung gestartet. Kommt die Verbindung zum Modbus-Server danach erst zustande, werden die projektierten Werte erfasst, ohne dass *ibaPDA*-Neustart notwendig ist. Diese Option empfiehlt sich für einen automatischen Neustart.

#### <Protokolldatei öffnen>

Wenn Verbindungen zu Steuerungen hergestellt wurden, dann werden alle verbindungs-spezifischen Aktionen in einer Textdatei protokolliert. Diese Datei können Sie über diesen Button öffnen und einsehen. Im Dateisystem auf der Festplatte finden Sie die Protokolldatei von *ibaPDA*-Server (`..\ProgramData\iba\ibaPDA\Log`).

Der Dateiname der aktuellen Protokolldatei lautet `SchnittstelleLog.txt`, der Name der archivierten Protokolldateien lautet `SchnittstelleLog_yyyy_mm_dd_hh_mm_ss.txt`.

#### <Statistik zurücksetzen>

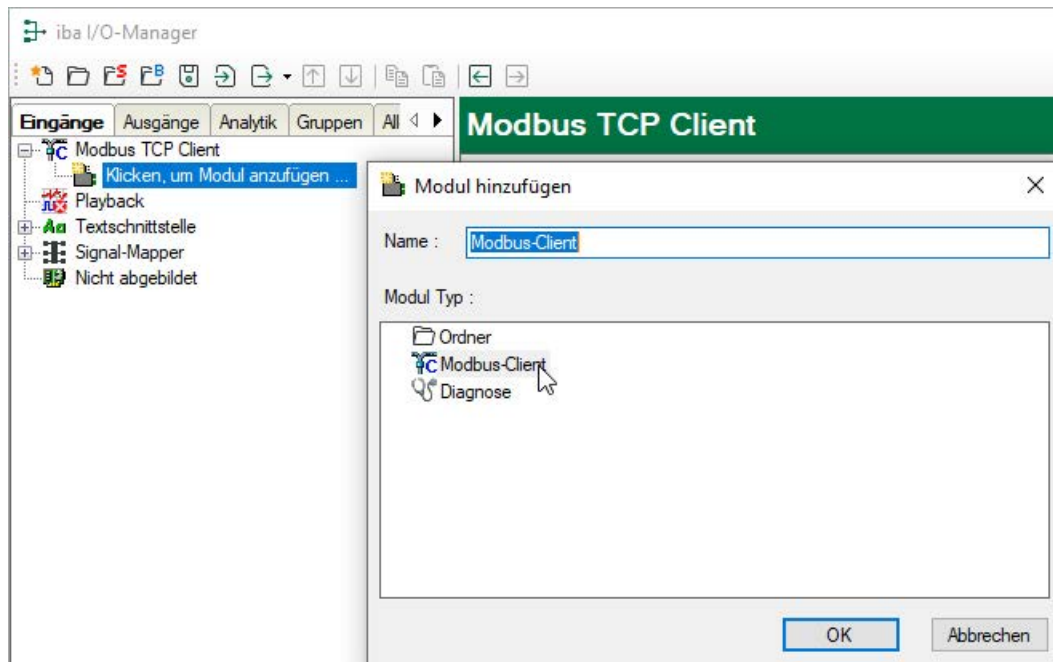
Über diesen Button können Sie die berechneten Zeitwerte und den Fehlerzähler in der Tabelle auf 0 setzen.

#### Verbindungsübersicht:

Sobald eine Verbindung aufgebaut ist, werden Live-Daten in der Übersicht angezeigt werden. Siehe auch *Überprüfung der Verbindung*, Seite 31.

### 3.2.3 Modul hinzufügen

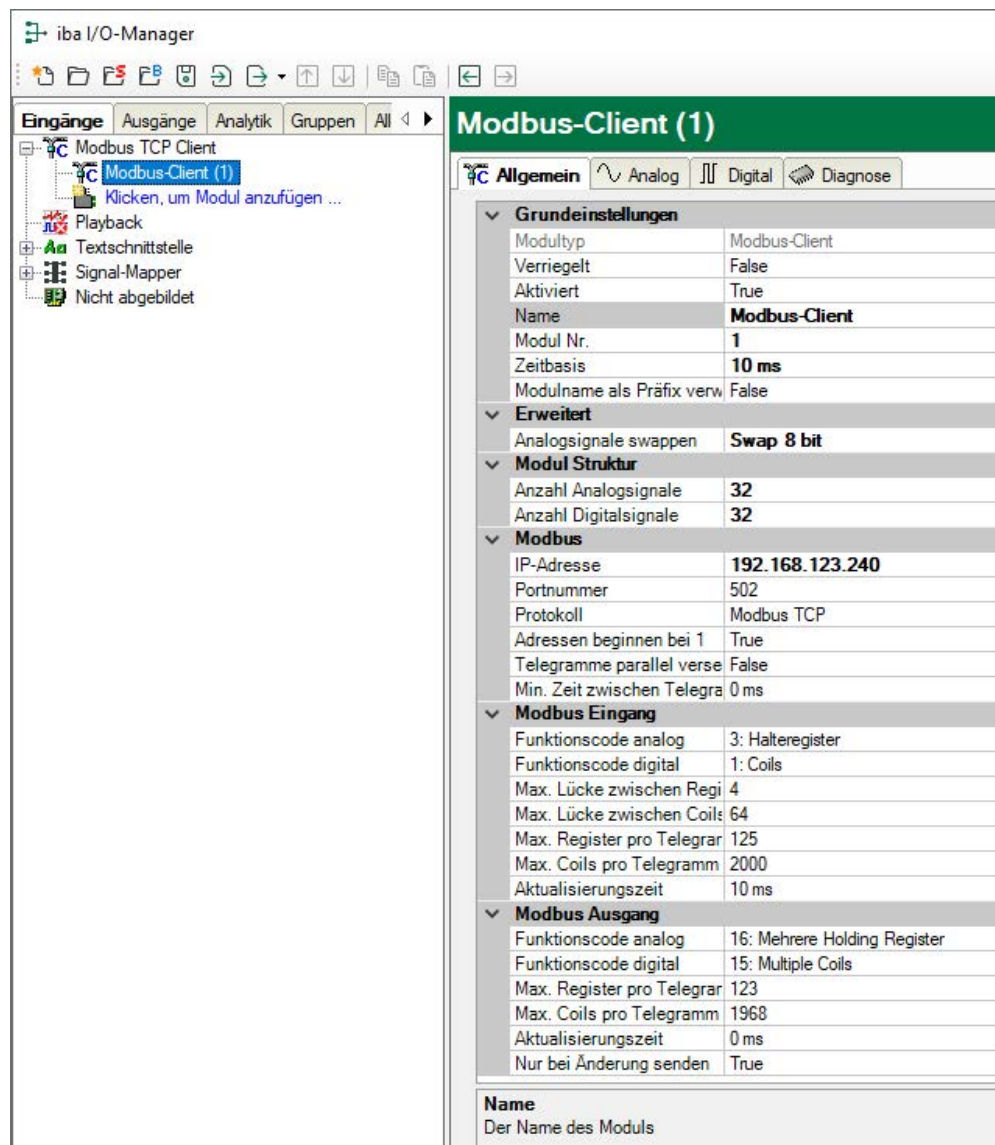
1. Klicken Sie auf den blauen Befehl *Klicken, um Modul anzufügen*, der sich unter jeder Datenschnittstelle im Register *Eingänge* oder *Ausgänge* befindet.
2. Wählen Sie im Dialogfenster den gewünschten Modultyp aus und vergeben Sie bei Bedarf einen Namen über das Eingabefeld.
3. Bestätigen Sie Ihre Auswahl mit <OK>.



### 3.2.3.1 Allgemeine Moduleinstellungen

Um ein Modul zu konfigurieren, markieren Sie es in der Baumstruktur.

Alle Module haben die folgenden Einstellmöglichkeiten.



#### Grundeinstellungen

##### Modultyp (nur Anzeige)

Zeigt den Typ des aktuellen Moduls an.

##### Verriegelt

Sie können ein Modul verriegeln, um ein versehentliches oder unautorisiertes Ändern der Einstellungen zu verhindern.

##### Aktiviert

Aktivieren Sie das Modul, um Signale aufzuzeichnen.

##### Name

Hier können Sie einen Namen für das Modul eintragen.

**Modul Nr.**

Diese interne Referenznummer des Moduls bestimmt die Reihenfolge der Module im Signalbaum von *ibaPDA-Client* und *ibaAnalyzer*.

**Zeitbasis**

Alle Signale dieses Moduls werden mit dieser Zeitbasis erfasst.

**Name als Präfix verwenden**

Diese Option stellt den Modulnamen den Signalnamen voran.

**Erweitert****Analogsignale swappen**

Möglichkeit die Auswertereihenfolge der Byte zu ändern.

---

**Tipp**

Für das System Modicon M580 wählen Sie die Einstellung "Swap 8 bit"!

---

**Modul Struktur****Anzahl der Analogsignale/Digitalsignale**

Stellen Sie die Anzahl der konfigurierbaren Analogsignale bzw. Digitalsignale in den Signaltabellen ein. Der Standardwert ist jeweils 32. Der Maximalwert beträgt 1000. Die Signaltabellen werden entsprechend angepasst.

**Modbus****IP-Adresse**

IP-Adresse des Modbus-Servers.

**Portnummer**

Standardeinstellung ist Port 502.

**Protokoll**

Standardeinstellung ist "Modbus TCP". Alternativ kann "Modbus seriell" ausgewählt werden. Dadurch wird das serielle RTU-Modbus-Protokoll, aber über TCP/IP übertragen, ausgewählt. Verwenden Sie dieses, wenn das Gerät mit einem Ethernet/Seriell-Gateway verbunden ist, wie z. B. das IF2E001 von IME. Siehe ➔ *Modbus Protokoll*, Seite 11.

**Adressen beginnen mit 1**

Bei Modbus werden standardmäßig die internen Bereiche (Register, Coils) ab 1, aber die Referenzadressen ab 0 nummeriert.

True: Die Adressen in den Registern *Analog* und *Digital* beginnen mit 1.

False: Die Adressen in den Registern *Analog* und *Digital* beginnen mit 0.

---

**Tipp**

Für das System Modicon M580 wählen Sie die Einstellung "False"!

---

**Telegramme parallel versenden**

True: Alle Telegramme, die zum Lesen aller Signale dieses Moduls erforderlich sind, werden parallel gesendet, dann wird auf alle Antworten gewartet.

False: Nach jedem Request-Telegramm wird auf das Response-Telegramm gewartet, dann wird erst das nächste gesendet.

**Min. Zeit zwischen Telegrammen**

Geben Sie an, in welchem zeitlichen Mindestabstand die Telegramme an den Modbus-Server gesendet werden. 0 bedeutet, dass alle Telegramme direkt nacheinander gesendet werden.

**Modbus Eingang****Funktionscode analog**

Wählen Sie den analogen Datentyp "Halteregeister" oder "Eingaberegister".

**Funktionscode digital**

Wählen Sie den digitalen Datentyp "Coils", "Diskrete Eingaben" oder "Halteregeister". Bei "Halteregeister" können Sie einzeln Bits aus Halteregeistern erfassen.

**Maximale Lücke zwischen Registern**

Nicht aufeinanderfolgende Register werden in einer Anforderung gesendet, wenn die Lücken zwischen den Registern nicht größer als der angegebene Wert sind. Standardeinstellung ist 4.

**Maximale Lücke zwischen Coils**

Nicht aufeinanderfolgende Coils oder Input-Bits werden in einer Anforderung gesendet, wenn die Lücken zwischen den Bitadressen nicht größer als der angegebene Wert sind. Standardeinstellung ist 64.

**Max. Register/Max. Coils pro Telegramm**

Hier kann die maximale Anzahl an Register bzw. Coils pro Telegramm bestimmt werden. Standardmäßig sind die, üblicherweise lt. Norm, eingetragenen Defaultwerte vorgegeben. Manche Geräte erlauben andere Register bzw. Coils pro Telegramm, die hier eingestellt werden können.

**Aktualisierungszeit**

Diese Zeit gibt an, wie schnell die Daten vom Modbus-Server angefordert werden. Standard ist identisch zu Parameter "Zeitbasis".

**Modbus Ausgang**

siehe Abschnitt "Modbus Eingang".

**Nur bei Änderung senden**

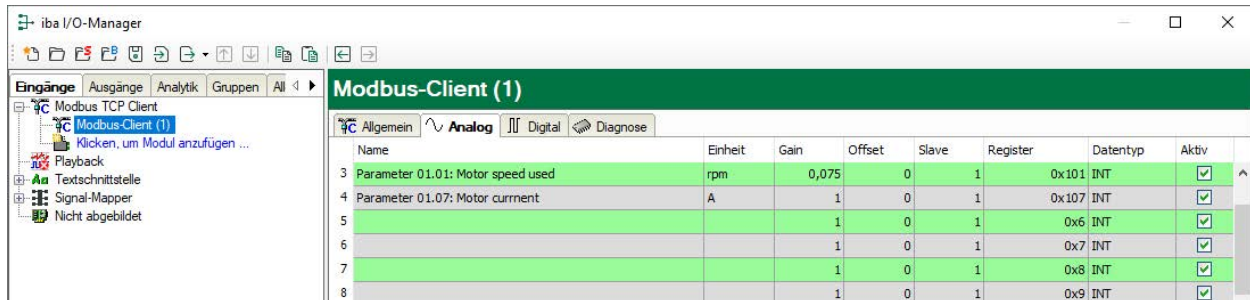
False: das Ausgangstelegramm wird im oben angegebenen Aktualisierungszyklus gesendet.

True: das Ausgangstelegramm wird nur nach der Änderung eines Wertes im Telegramm gesendet, aber spätestens nach 1 min.

### 3.2.3.2 Signalkonfiguration

In den Registern *Analog* und *Digital* konfigurieren Sie die zu messenden Signale. Die Länge der Signaltabellen, d. h. die Anzahl der Signale pro Tabelle, stellen Sie im Register *Allgemein* unter *Modul Struktur* ein.

#### Register Analog und Digital



Name	Einheit	Gain	Offset	Slave	Register	Datentyp	Aktiv
3 Parameter 01.01: Motor speed used	rpm	0,075	0	1	0x101	INT	<input checked="" type="checkbox"/>
4 Parameter 01.07: Motor current	A	1	0	1	0x107	INT	<input checked="" type="checkbox"/>
5		1	0	1	0x6	INT	<input checked="" type="checkbox"/>
6		1	0	1	0x7	INT	<input checked="" type="checkbox"/>
7		1	0	1	0x8	INT	<input checked="" type="checkbox"/>
8		1	0	1	0x9	INT	<input checked="" type="checkbox"/>

Sie können die Signale mit Namen, Einheit, Skalierungsfaktor und Datentyp bzw. Bit-Nummer versehen, sowie aktiv oder inaktiv setzen.

#### Andere Dokumentation



Eine Beschreibung der Spalten finden Sie im *ibaPDA-Handbuch*.

Spezifische Spalten Modbus TCP-Client Module:

#### Slave

Die Slave-Nummer ist nur dann relevant, wenn Sie mit einem Gateway arbeiten, an dem mehrere Modbus-Slaves angeschlossen sind. Wird im Modbus-Telegramm in das Feld Unit-ID eingetragen. Im Modbus TCP-Modus wird die Slave-Nummer von den meisten Modbus-Servern ignoriert.

#### Register

Die Registernummer geht von 0 bis 65535 oder von 1 bis 65536, abhängig von der Option „Adressen beginnen mit 1“ im Register *Allgemein*. Jedes Register ist 16 Bit lang. Die Register können im Hexadezimal- oder Dezimal-Format angegeben werden. Über ein Kontextmenü kann das Format umgestellt werden.

Für Digitalsignale erscheint diese Spalte nur bei dem digitalen Zugriffstyp "Halteregister", bei dem die Bits (0-15) aus einem Halteregister, z. B. einem Statuswort, erfasst werden.

#### Datentyp (nur Analogsignale)

Wählen Sie einen der folgenden Datentypen aus: BYTE, WORD, DWORD, INT, DINT, FLOAT, DOUBLE.

Die Werte der Datentypen FLOAT, DINT und DWORD benötigen jeweils 2 Register.

Der Datentyp DOUBLE belegt 4 Register, 2 Werte des Datentyps BYTE benötigen ein Register.

#### Bit-Nr. (nur Digitalsignale)

Die Bitnummer entspricht der Coil-Nummer, der „Discrete Input“-Bitnummer oder der Bitnummer innerhalb des Halteregisters. Der Wertebereich ist 0 bis 65535, bzw. 1 bis 65535, abhängig vom Parameter "Adressen beginnen mit 1".



**Tipp**

Sie können das automatische Ausfüllen der Spalten verwenden (siehe *ibaPDA*-Handbuch).

### 3.2.4 Konfiguration ibaPDA-Ausgangsmodule

Wenn alle Systemvoraussetzungen erfüllt sind, bietet *ibaPDA* im Schnittstellenbaum des Registers *Ausgänge* die Schnittstelle *Modbus Client* an. Ein manuelles Hinzufügen der Schnittstelle ist nicht erforderlich.

Ausgangsmodule fügen Sie auf die gleiche Weise hinzu wie Eingangsmodule.

#### 3.2.4.1 Allgemeine Moduleinstellungen ibaPDA-Ausgangsmodule

Um ein Ausgangsmodul zu konfigurieren, markieren Sie es in der Baumstruktur des Registers *Ausgänge*.

The screenshot shows the 'iba I/O-Manager' interface. On the left, a tree view under 'Ausgänge' shows a 'Modbus-Client (1)' module selected. The right pane displays the configuration for this module, with tabs for 'Allgemein', 'Analog', 'Digital', and 'Diagnose'. The 'Allgemein' tab is active, showing various settings organized into sections: 'Grundeinstellungen', 'Erweitert', 'Modul Struktur', 'Modbus', 'Modbus Eingang', and 'Modbus Ausgang'.

Grundeinstellungen	
Modultyp	Modbus-Client
Verriegelt	False
Aktiviert	True
Name	Modbus-Client
Modul Nr.	1
Berechnungszeitbasis	10 ms
Minimale Ausgangszeitbasis	50 ms
Name als Präfix verwenden	False

Erweitert	
Analogsignale swappen	Swap 8 bit

Modul Struktur	
Anzahl analoge Ausgangssignale	32
Anzahl digitale Ausgangssignale	32

Modbus	
IP-Adresse	192.168.123.240
Portnummer	502
Protokoll	Modbus TCP
Adressen beginnen bei 1	True
Telegramme parallel versenden	False
Min. Zeit zwischen Telegrammen	0 ms

Modbus Eingang	
Funktionscode analog	3: Haltereister
Funktionscode digital	1: Coils
Max. Lücke zwischen Registern	4
Max. Lücke zwischen Coils	64
Max. Register pro Telegramm	125
Max. Coils pro Telegramm	2000
Aktualisierungszeit	10 ms

Modbus Ausgang	
Funktionscode analog	16: Mehrere Holding Register
Funktionscode digital	15: Multiple Coils
Max. Register pro Telegramm	123
Max. Coils pro Telegramm	1968
Aktualisierungszeit	0 ms
Nur bei Änderung senden	True

At the bottom, the 'Name' field is labeled 'Der Name des Moduls'.



Die Parameter sind nahezu identisch zu denen der Input-Module, siehe [↗ Allgemeine Moduleinstellungen](#), Seite 20.

Beachten Sie folgende Unterschiede zu den Einstellungen der Input-Module:

### Berechnungszeitbasis

Zeitbasis (in ms), die für die Berechnung der Ausgangswerte verwendet wird.

Technisch ist die Berechnungszeitbasis identisch mit der Zeitbasis auf der Eingangsseite des Moduls. Das heißt, eine Änderung der Berechnungszeitbasis ändert auch die Modulzeitbasis der Eingangsseite und umgekehrt!

Die Berechnungszeitbasis ist nicht gleichbedeutend mit der Ausgangszeitbasis, mit der die Werte ausgegeben werden!

### Minimale Ausgangszeitbasis

Zeitbasis, mit der die Ausgänge schnellstens aktualisiert werden können.

Der Wert wird vom System anhand der aktuellen I/O-Konfiguration automatisch ermittelt und hier nur angezeigt. Die Ausgangszeitbasis ergibt sich aus dem kleinsten gemeinsamen Vielfachen aller Modulzeitbasen, bzw. beträgt mindestens 50 ms.

## 3.2.4.2 Signalkonfiguration

In den Registern *Analog* und *Digital* können Sie die Ausgangssignale wie bei den virtuellen Signalen festlegen.

### Tipp



Wenn Sie die Ausgabedaten in einem virtuellen Modul definieren und hier nur die Referenzen zu diesen Daten eintragen, besteht die Möglichkeit, diese Daten auch in die Datenaufzeichnung aufzunehmen.

### Register Analog und Digital

Modbus-Client (6)						
<div> <span>Allgemein</span> <span>Analog</span> <span>Digital</span> <span>Diagnose</span> </div>						
	Name	Ausdruck		Slave	Register	Datentyp
0	Const max	$\text{fx}$ 1000		7	0x201	DINT
1	Const min	$\text{fx}$ 1		7	0x203	FLOAT
2		$\text{fx}$		1	0x3	INT
3		$\text{fx}$		1	0x4	INT

### Ausdruck

Definieren Sie die gewünschten Ausgangssignale auf ähnliche Weise wie bei den virtuellen Signalen. Einfache Ausdrücke oder Referenzen zu bestehenden Signalen können Sie direkt in die Tabellen eingeben. Über den Button  $\text{fx}$  öffnen Sie den Ausdruckseditor. Einen fehlerhaften Ausdruck können Sie über den Button  $\text{fx}$  analysieren.

### Name, Slave, Register, Datentyp, Bit-Nr., Aktiv

Die Spalten sind für Eingangs- und Ausgangsmodule identisch, siehe [↗ Signalkonfiguration](#), Seite 23

## Andere Dokumentation









Eine Beschreibung der Spalten finden Sie im *ibaPDA*-Handbuch.



### 3.2.5 Moduldiagnose

Die Moduldiagnose gibt es sowohl für die Eingangs- als auch für die Ausgangsrichtung. Dazu gibt es zwei Subregister *Analog* und *Digital*, in denen die aktuellen Werte numerisch angezeigt werden.

Die Tabelle "Analog" (Eingänge) zeigt immer die unskalierten Rohwerte im Gleitpunktformat.

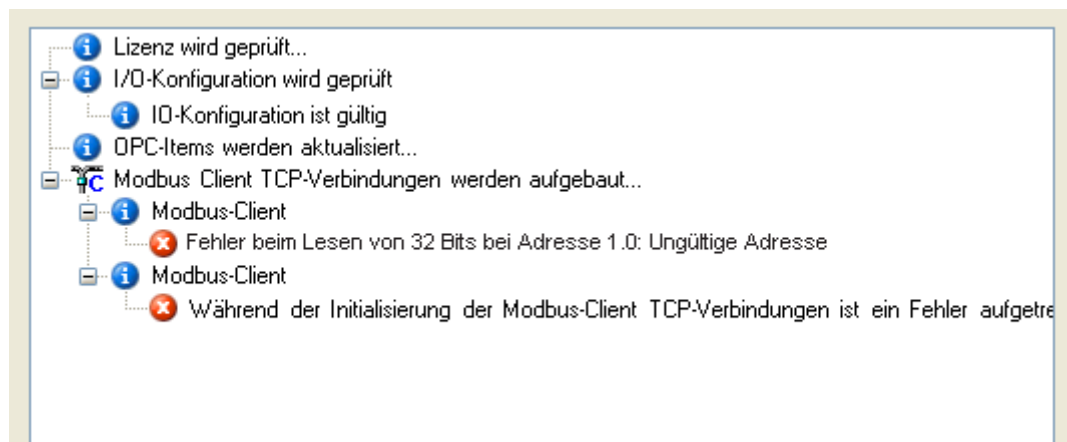
Modbus-Client (5)		
 Allgemein	 Analog	 Digital
		 Diagnose
 Analogwerte	 Digitalwerte	
Name	Adresse	Wert
0 Parameter 01.01: Motor speed used	1.101	20000
1 Parameter 01.07: Motor current	1.107	2
2 Parameter 01.10: Motor torque	1.110	0
3 Parameter 01.11: DC voltage	1.111	5684
4	?	?
5	?	?

Folgende Fehler können auf der Eingangsseite erkannt werden:

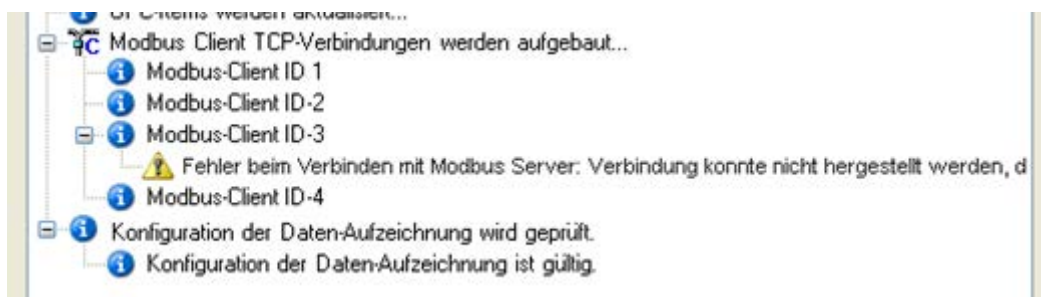
- Es werden keine Daten angezeigt:
  - Keine Verbindung zum Modbus-Server.
  - Fehlermeldung durch den Modbus-Server, siehe  *Fehlerbehandlung*, Seite 39.
- Es werden falsche Werte angezeigt:
  - Bytereihenfolge ist falsch eingestellt, siehe  *Allgemeine Moduleinstellungen*, Seite 20.

### 3.2.6 Erfassung starten

Mit dem Beginn der Erfassung versucht *ibaPDA* eine Verbindungen zum Modbus-Server aufzubauen und die Daten zu lesen. Wenn der Server einen Fehler entdeckt, erscheint eine Meldung.



Wenn die Option "Erfassung auch starten, wenn kein Modbus Server erreichbar ist", (siehe ↗ *Allgemeine Einstellungen der Schnittstelle*, Seite 18) ausgewählt ist, wird die Meldung "Fehler beim Verbinden mit Modbus Server..." als Warnung angezeigt und die Erfassung trotzdem gestartet.



## 4 Diagnose

### 4.1 Lizenz

Falls die gewünschte Schnittstelle nicht im Signalbaum angezeigt wird, können Sie entweder in *ibaPDA* im I/O-Manager unter *Allgemein – Einstellungen* oder in der *ibaPDA* Dienststatus-Applikation überprüfen, ob Ihre Lizenz für diese Schnittstelle ordnungsgemäß erkannt wird. Die Anzahl der lizenzierten Verbindungen ist in Klammern angegeben.

Die folgende Abbildung zeigt beispielhaft die Lizenz für die Schnittstelle *Codesys-Xplorer*.

The screenshot shows a software interface for license management. On the left, under 'Lizenzinformationen', there are several input fields: 'Lizenzcontainer:' with value '3-...', 'Kundenname:' with value '...', 'Nutzungsdauer:' with value 'Unbegrenzt', 'Containertyp:' with value 'WIBU CmStick v4.40', 'Container-Host:' with value '...', 'Erforderl. EUP-Datum:' with value '01.02.2023', and 'EUP-Datum:' with value '31.12.2025'. On the right, under 'Lizenzen:', there is a list box containing several license entries. One entry, 'ibaPDA-Interface-Codesys-Xplorer (16)', is highlighted with a red rectangular box.

### 4.2 Sichtbarkeit der Schnittstelle

Ist die Schnittstelle trotz gültiger Lizenz nicht zu sehen, ist sie möglicherweise verborgen.

Überprüfen Sie die Einstellung im Register *Allgemein* im Knoten *Schnittstellen*.

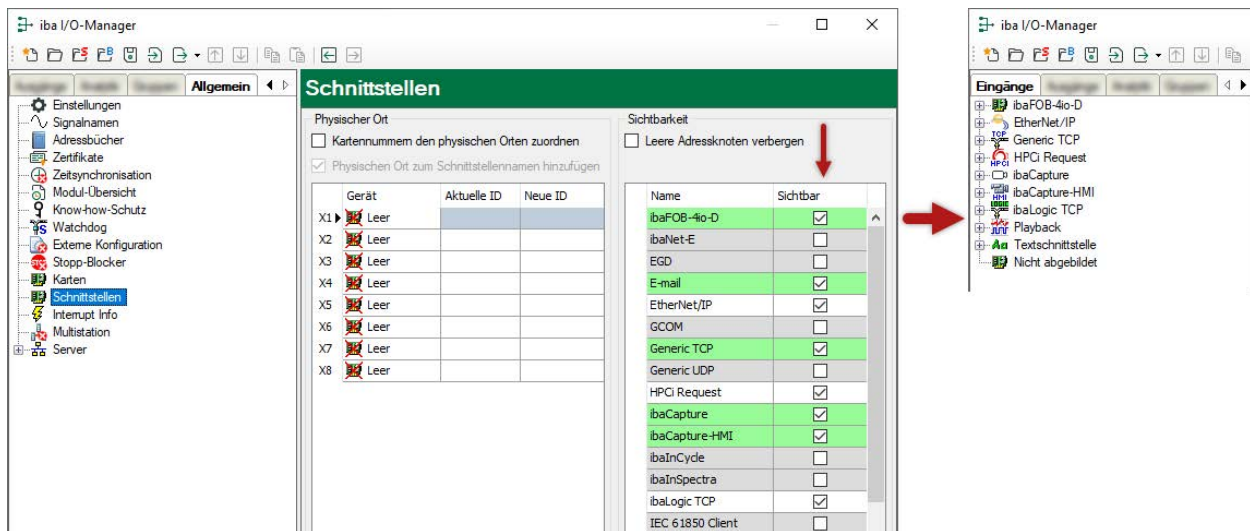
#### Sichtbarkeit

Die Tabelle *Sichtbarkeit* listet alle Schnittstellen auf, die entweder durch Lizenzen oder installierte Karten verfügbar sind. Diese Schnittstellen sind auch im Schnittstellenbaum zu sehen.

Mithilfe der Häkchen in der Spalte *Sichtbar* können Sie nicht benötigte Schnittstellen im Schnittstellenbaum verbergen oder anzeigen.

Schnittstellen mit konfigurierten Modulen sind grün hinterlegt und können nicht verborgen werden.

Ausgewählte Schnittstellen sind sichtbar, die anderen Schnittstellen sind verborgen:



### 4.3 Protokolldateien

Wenn Verbindungen zu Zielsystemen bzw. Clients hergestellt wurden, dann werden alle verbindungs-spezifischen Aktionen in einer Textdatei protokolliert. Diese (aktuelle) Datei können Sie z. B. nach Hinweisen auf mögliche Verbindungsprobleme durchsuchen.

Die Protokolldatei können Sie über den Button <Protokolldatei öffnen> öffnen. Der Button befindet sich im I/O-Manager:

- bei vielen Schnittstellen in der jeweiligen Schnittstellenübersicht
- bei integrierten Servern (z. B. OPC UA-Server) im Register Diagnose.

Im Dateisystem auf der Festplatte finden Sie die Protokolldateien von *ibaPDA*-Server (... \ProgramData\iba\ibaPDA\Log). Die Dateinamen der Protokolldateien werden aus der Bezeichnung bzw. Abkürzung der Schnittstellenart gebildet.

Dateien mit Namen *Schnittstelle.txt* sind stets die aktuellen Protokolldateien. Dateien mit Namen *Schnittstelle\_yyyy\_mm\_dd\_hh\_mm\_ss.txt* sind archivierte Protokolldateien.

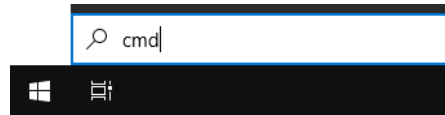
Beispiele:

- *ethernetipLog.txt* (Protokoll von EtherNet/IP-Verbindungen)
- *AbEthLog.txt* (Protokoll von Allen-Bradley-Ethernet-Verbindungen)
- *OpcUAServerLog.txt* (Protokoll von OPC UA-Server-Verbindungen)

## 4.4 Verbindungsdiagnose mittels PING

Ping ist ein System-Befehl, mit dem Sie überprüfen können, ob ein bestimmter Kommunikationspartner in einem IP-Netzwerk erreichbar ist.

1. Öffnen Sie eine Windows Eingabeaufforderung.



2. Geben Sie den Befehl "ping" gefolgt von der IP-Adresse des Kommunikationspartners ein und drücken Sie <ENTER>.

→ Bei bestehender Verbindung erhalten Sie mehrere Antworten.

```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

Ping wird ausgeführt für 192.168.1.10 mit 32 Bytes Daten:
Antwort von 192.168.1.10: Bytes=32 Zeit=1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.1.10: Bytes=32 Zeit<1ms TTL=30

Ping-Statistik für 192.168.1.10:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

→ Bei nicht bestehender Verbindung erhalten Sie Fehlermeldungen.

```
Administrator: Eingabeaufforderung
Microsoft Windows [Version 10.0]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\system32>ping 192.168.1.10

Ping wird ausgeführt für 192.168.1.10 mit 32 Bytes Daten:
Antwort von 192.168.1.10: Zielhost nicht erreichbar.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.1.10:
    Pakete: Gesendet = 4, Empfangen = 1, Verloren = 3
    (75% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 1ms, Mittelwert = 0ms

C:\Windows\system32>
```

## 4.5 Überprüfung der Verbindung

Wenn Sie im Signalbaum des I/O-Managers die Datenschnittstelle Modbus-Client markieren, erhalten Sie im rechten Fenster eine Übersicht über alle Verbindungen dieser Schnittstelle.

	IP-Adresse	Fehlerzähler	Telegramme pro Zyklus	Aktualisierungszeit	Antwortzeit Aktuell	Antwortzeit Mittelwert	Antwortzeit Minimum	Antwortzeit Maximum
0	192.168.50.53	5	10	0 ms	0 ms	86 ms	65 ms	179 ms
1	192.168.50.53	5	12	0 ms	0 ms	102 ms	92 ms	344 ms
2	192.168.162.128	0	2	64 ms	63 ms	63 ms	37 ms	127 ms
3	192.168.162.128	0	4	122 ms	122 ms	126 ms	119 ms	221 ms
4	192.168.162.128	0	2	60 ms	60 ms	63 ms	56 ms	157 ms
5	192.168.162.128	0	1	29 ms	29 ms	32 ms	11 ms	129 ms
6	?	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?	?

### Buttons:

- Mit <Statistik zurücksetzen> können der Fehlerzähler und die mittleren, minimalen und maximalen Antwortzeiten zurückgesetzt werden.
- Mit <Protokolldatei öffnen> wird die Protokolldatei in einem ASCII-Editor geöffnet werden, um den Fehlerverlauf zu verfolgen.

### In der Verbindungsliste sind folgende Werte zu sehen:

- IP-Adresse: Adresse des Modbus-Servers
- Fehlerzähler: wird hochgezählt bei sporadischen Übertragungsfehlern (Verbindungsabbruch) oder wenn der Modbus-Server einen Fehlercode sendet, siehe [Fehlerbehandlung](#), Seite 39.
- Telegramme pro Zyklus: zeigt an, wieviel Telegramme zum Empfang der Daten dieses Moduls notwendig sind
- Aktualisierungszeit: Zyklus in dem *ibaPDA* die Daten vom Modbus-Server anfordert (Maximum aus eingestellter Aktualisierungszeit und Modbus-Server-Zykluszeit)
- Antwortzeit Aktuell: Zeitintervall mit dem der Modbus-Server sendet (Zeit zwischen Request und Response)
- Antwortzeit Mittelwert, Minimum, Maximum: berechnet Werte seit Start der Erfassung oder Rücksetzen der Zähler mit dem Button <Statistik zurücksetzen>

### Farben:

- Grün: Die Verbindung ist OK.
- Rot: Die Verbindung ist abgebrochen

**Verbindungsabbruch kann folgende Ursachen haben:**

- Modbus-Server ist in Stopp
- keine Ethernet-Verbindung zwischen *ibaPDA*-PC und dem Modbus-PLC
- Fehler in der Verbindungs-Projektierung:
  - falsche Remote-IP-Adresse
  - *ibaPDA*-Portnummer stimmt nicht mit der Verbindungs-Projektierung überein.

**Weitere Fehler:**

- Zählen Werte in der Spalten "Fehlerzähler" hoch, deutet das auf einen der folgenden Fehler hin:
  - Fehler im Telegramm-Header



## 4.6 Diagnosemodule

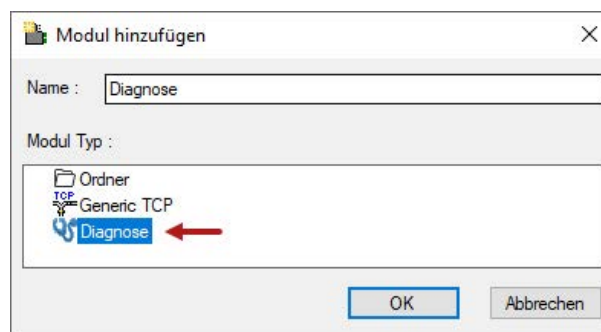
Diagnosemodule sind für die meisten Ethernet-basierten Schnittstellen und Xplorer-Schnittstellen verfügbar. Mit einem Diagnosemodul können Informationen aus den Diagnoseanzeigen (z. B. Diagnoseregister und Verbindungstabellen einer Schnittstelle) als Signale erfasst werden.

Ein Diagnosemodul ist stets einem Datenerfassungsmodul derselben Schnittstelle zugeordnet und stellt dessen Verbindungsinformationen zur Verfügung. Durch die Nutzung eines Diagnosemoduls können die Diagnoseinformationen auch im *ibaPDA*-System durchgängig aufgezeichnet und ausgewertet werden. Diagnosemodule verbrauchen keine Verbindung der Lizenz, da sie keine Verbindung aufbauen, sondern auf ein anderes Modul verweisen.

Nutzungsbeispiele für Diagnosemodule:

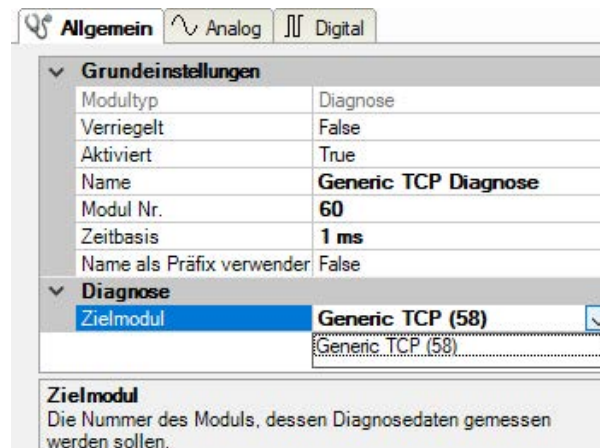
- Wenn der Fehlerzähler einer Kommunikationsverbindung einen bestimmten Wert überschreitet oder eine Verbindung abbricht, kann eine Benachrichtigung generiert werden.
- Bei einem Störfall können die aktuellen Antwortzeiten im Telegrammverkehr in einem Störsreport dokumentiert werden.
- Der Status der Verbindungen kann in *ibaQPanel* visualisiert werden.
- Diagnoseinformationen können über den in *ibaPDA* integrierten SNMP-Server oder OPC DA/UA-Server an übergeordnete Überwachungssysteme wie Netzwerkmanagement-Tools weitergegeben werden.

Wenn für eine Schnittstelle ein Diagnosemodul verfügbar ist, wird im Dialog "Modul hinzufügen" der Modultyp "Diagnose" angezeigt (Beispiel: Generic TCP).



### Moduleinstellungen Diagnosemodul

Bei einem Diagnosemodul können Sie folgende Einstellungen vornehmen (Beispiel: Generic TCP):



Die Grundeinstellungen eines Diagnosemoduls entsprechen denen der anderen Module. Es gibt nur eine für das Diagnosemodul spezifische Einstellung, die vorgenommen werden muss: das Zielmodul.

Mit der Auswahl des Zielmoduls weisen Sie das Diagnosemodul dem Modul zu, dessen Verbindungsinformationen erfasst werden sollen. In der Drop-down-Liste der Einstellung stehen die unterstützten Module derselben Schnittstelle zur Auswahl. Pro Diagnosemodul kann genau ein Datenerfassungsmodul zugeordnet werden. Wenn Sie ein Modul ausgewählt haben, werden in den Registern *Analog* und *Digital* umgehend die verfügbaren Diagnosesignale hinzugefügt. Welche Signale das sind, hängt vom Schnittstellentyp ab. Im nachfolgenden Beispiel sind die Analogwerte eines Diagnosemoduls für ein Generic TCP-Modul aufgelistet.

Allgemein Analog Digital						
	Name	Einheit	Gain	Offset	Aktiv	Istwert
0	IP-Adresse (Teil 1)		1	0	<input checked="" type="checkbox"/>	
1	IP-Adresse (Teil 2)		1	0	<input checked="" type="checkbox"/>	
2	IP-Adresse (Teil 3)		1	0	<input checked="" type="checkbox"/>	
3	IP-Adresse (Teil 4)		1	0	<input checked="" type="checkbox"/>	
4	Port		1	0	<input checked="" type="checkbox"/>	
5	Telegrammzähler		1	0	<input checked="" type="checkbox"/>	
6	Unvollständig		1	0	<input checked="" type="checkbox"/>	
7	Paketgröße (aktuell)	Bytes	1	0	<input checked="" type="checkbox"/>	
8	Paketgröße (max)	Bytes	1	0	<input checked="" type="checkbox"/>	
9	Zeit zwischen Daten (aktuell)	ms	1	0	<input checked="" type="checkbox"/>	
10	Zeit zwischen Daten (min)	ms	1	0	<input checked="" type="checkbox"/>	
11	Zeit zwischen Daten (max)	ms	1	0	<input checked="" type="checkbox"/>	

Die IP(v4-)Adresse eines Generic TCP-Moduls, z. B. (siehe Abbildung), wird entsprechend der 4 Bytes bzw. Oktetts in 4 Teile zerlegt, um sie leichter lesen und vergleichen zu können. Andere Größen, wie Portnummer, Zählerstände für Telegramme und Fehler, Datengrößen und Laufzeiten für Telegramme werden ebenfalls ermittelt. Im nachfolgenden Beispiel sind die Digitalwerte eines Diagnosemoduls für ein Generic TCP-Modul aufgelistet.

Allgemein Analog Digital		
	Name	Aktiv Istwert
0	Aktiver Verbindungsmodus	<input checked="" type="checkbox"/>
1	Ungültiges Paket	<input checked="" type="checkbox"/>
2	Verbinde	<input checked="" type="checkbox"/>
3	Verbunden	<input checked="" type="checkbox"/>

## Diagnosesignale

Abhängig vom Schnittstellentyp stehen folgende Signale zur Verfügung:

Signalname	Bedeutung
Aktiv	Nur für redundante Verbindungen relevant. Aktiv bedeutet, dass die Verbindung zur Messung der Daten verwendet wird, d. h. bei redundanten Standby-Verbindungen steht der Wert 0. Bei normalen/nicht redundanten Verbindungen steht immer der Wert 1.
Aktualisierungszeit (Istwert/konfiguriert/max/min/Mittelwert)	Gibt die Aktualisierungszeit an, in der die Daten aus der SPS, der CPU oder vom Server abgerufen werden sollen (konfiguriert). Standard ist gleich dem Parameter "Zeitbasis". Während der Messung kann die reale aktuelle Aktualisierungszeit (Istwert) höher sein als der eingestellte Wert, wenn die SPS mehr Zeit zur Übertragung der Daten benötigt. Wie schnell die Daten wirklich aktualisiert werden, können Sie in der Verbindungstabelle überprüfen. Die minimal erreichbare Aktualisierungszeit wird von der Anzahl der Signale beeinflusst. Je mehr Signale erfasst werden, desto größer wird die Aktualisierungszeit.  Max/min/Mittelwert: statische Werte der Aktualisierungszeit seit dem letzten Start der Erfassung bzw. Rücksetzen der Zähler
Anforderungen Sendewiederholung	Anzahl der nochmals angeforderten Datentelegramme (in) bei Verlust oder Verspätung
Antwortzeit (aktuell/konfiguriert/max/min/Mittelwert)	Antwortzeit ist die Zeit zwischen Messwertanforderung von <i>ibaPDA</i> und Antwort von der SPS bzw. Empfang der Daten.  Aktuell: Istwert  Max/min/Mittelwert: statische Werte der Antwortzeit seit dem letzten Start der Erfassung bzw. Rücksetzen der Zähler
Anzahl Anforderungsbefehle	Zähler für Anforderungstelegramme von <i>ibaPDA</i> an die SPS/CPU
Aufgebaute Verbindungen (in)	Anzahl der aktuell gültigen Datenverbindungen für den Empfang
Aufgebaute Verbindungen (out)	Anzahl der aktuell gültigen Datenverbindungen für das Senden
Ausgangsdatenlänge	Länge der Datentelegramme mit Ausgangssignalen in Bytes ( <i>ibaPDA</i> sendet)
Datenlänge	Länge der Datentelegramme in Bytes

Signalname	Bedeutung
Datenlänge des Inputs	Länge der Datentelegramme mit Eingangssignalen in Bytes ( <i>ibaPDA</i> empfängt)
Datenlänge O->T	Größe des Output-Telegramms in Byte
Datenlänge T->O	Größe des Input-Telegramms in Byte
Definierte Topics	Anzahl der definierten Topics
Empfangene Telegramme seit Konfiguration	Anzahl der empfangenen Datentelegramme (in) seit Beginn der Erfassung
Empfangene Telegramme seit Verbindungsstart	Anzahl der empfangenen Datentelegramme (in) seit Beginn des letzten Verbindungsaufbaus
Empfangszähler	Anzahl der empfangenen Telegramme
Exchange ID	ID des Datenaustauschs
Falscher Telegrammtyp	Anzahl der Empfangstelegramme mit falschem Telegrammtyp
Fehlerzähler	Zähler der Kommunikationsfehler
Gepufferte Anweisungen	Anzahl der noch nicht ausgeführten Anweisungen im Zwischenspeicher
Gepufferte Anweisungen sind verloren	Anzahl der gepufferten aber nicht ausgeführten und verlorenen Anweisungen
Gesendete Telegramme seit Konfiguration	Anzahl der gesendeten Datentelegramme (out) seit Beginn der Erfassung
Gesendete Telegramme seit Verbindungsstart	Anzahl der gesendeten Datentelegramme (out) seit Beginn des letzten Verbindungsaufbaus
ID der Verbindung O->T	ID der Verbindung für Output-Daten (vom Zielsystem an <i>ibaPDA</i> ) Entspricht der Assembly-Instanznummer
ID der Verbindung T->O	ID der Verbindung für Input-Daten (von <i>ibaPDA</i> an Zielsystem) Entspricht der Assembly-Instanznummer
IP-Adresse (Teil 1-4)	4 Oktets der IP-Adresse des Zielsystems
IP-Quelladresse (Teil 1-4) O->T	4 Oktets der IP-Adresse des Zielsystems Output-Daten (vom Zielsystem an <i>ibaPDA</i> )
IP-Quelladresse (Teil 1-4) T->O	4 Oktets der IP-Adresse des Zielsystems Input-Daten (von <i>ibaPDA</i> an Zielsystem)
IP-Zieladresse (Teil 1-4) O->T	4 Oktets der IP-Adresse des Zielsystems Output-Daten (vom Zielsystem an <i>ibaPDA</i> )
IP-Zieladresse (Teil 1-4) T->O	4 Oktets der IP-Adresse des Zielsystems Input-Daten (von <i>ibaPDA</i> an Zielsystem)
KeepAlive-Zähler	Anzahl der vom OPC UA-Server empfangenen KeepAlive-Telegramme
Lesezähler	Anzahl der Lesezugriffe/Datenanforderungen
Multicast Anmeldefehler	Anzahl der Fehler bei Multicast-Anmeldung
Nachrichtenzähler	Anzahl der empfangenen Telegramme
Paketgröße (aktuell)	Größe der aktuell empfangenen Telegramme

Signalname	Bedeutung
Paketgröße (max)	Größe des größten empfangenen Telegramms
Ping-Zeit (Istwert)	Antwortzeit für ein Ping-Telegramm
Port	Portnummer für die Kommunikation
Producer ID (Teil 1-4)	Producer-ID als 4 Byte unsigned Integer
Profilzähler	Anzahl der vollständig erfassten Profile
Pufferdateigröße (aktuell/mittl./max)	Größe der Pufferdatei zum Zwischenspeichern der Anweisungen
Pufferspeichergröße (aktuell/mittl./max)	Größe des belegten Arbeitsspeichers zum Zwischenspeichern der Anweisungen
Schreibverlustzähler	Anzahl missglückter Schreibzugriffe
Schreibzähler	Anzahl erfolgreicher Schreibzugriffe
Sendezähler	Anzahl der Sendetelegramme
Sequenzfehler	Anzahl Sequenzfehler
Synchronisation	Gerät wird für die isochrone Erfassung synchronisiert
Telegramme pro Zyklus	Anzahl der Telegramme im Zyklus der Aktualisierungszeit
Telegrammzähler	Anzahl der empfangenen Telegramme
Topics aktualisiert	Anzahl der aktualisierten Topics
Trennungen (in)	Anzahl der aktuell unterbrochenen Datenverbindungen für den Empfang
Trennungen (out)	Anzahl der aktuell unterbrochenen Datenverbindungen für das Senden
Unbekannter Sensor	Anzahl unbekannter Sensoren
Ungültiges Paket	Ungültiges Datenpaket erkannt
Unvollständig	Anzahl unvollständiger Telegramme
Unvollständige Fehler	Anzahl unvollständiger Telegramme
Verarbeitete Anweisungen	Anzahl der ausgeführten SQL-Anweisungen seit dem letzten Start der Erfassung
Verbinde	Verbindung wird aufgebaut
Verbindungsphase (in)	Zustand der ibaNet-E Datenverbindung für den Empfang
Verbindungsphase (out)	Zustand der ibaNet-E Datenverbindung für das Senden
Verbindungsversuche (in)	Anzahl der Versuche, die Empfangsverbindung (in) aufzubauen
Verbindungsversuche (out)	Anzahl der Versuche, die Sendeverbindung (out) aufzubauen
Verbunden	Verbindung ist aufgebaut
Verbunden (in)	Eine gültige Datenverbindung für den Empfang (in) ist vorhanden
Verbunden (out)	Eine gültige Datenverbindung für das Senden (out) ist vorhanden

Signalname	Bedeutung
Verlorene Images	Anzahl der verlorenen Images (in), die selbst nach einer Sendewiederholung nicht empfangen wurden
Verlorene Profile	Anzahl unvollständiger/fehlerhafter Profile
Zeilen (letzte)	Anzahl der Ergebniszeilen der letzten SQL-Abfrage (innerhalb der projektierten Anzahl von Ergebniszeilen)
Zeilen (Maximum)	Höchste Anzahl der Ergebniszeilen einer SQL-Abfrage seit dem letzten Start der Erfassung (maximal gleich der projektierten Anzahl von Ergebniszeilen)
Zeit zwischen Daten (aktuell/max/min)	Zeit zwischen zwei korrekt empfangenen Telegrammen Aktuell: zwischen den letzten zwei Telegrammen Max/min: statistische Werte seit Start der Erfassung oder Rücksetzen der Zähler
Zeit-Offset (Istwert)	Gemessene Zeitdifferenz der Synchronität zwischen dem ibaNet-E-Gerät und <i>ibaPDA</i>

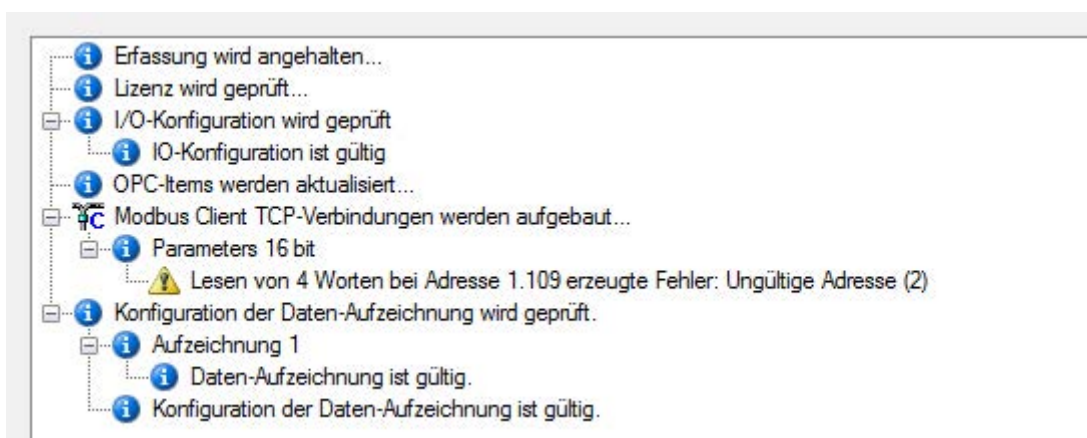
## 5 Anhang

### 5.1 Fehlerbehandlung

Bei einem Übertragungsfehler oder falls ein nicht vorhandenes (oder ausgeschaltetes) Gerät adressiert wird, sendet der Server keine Quittierung an den Client. Dies führt zu einem Timeout. Erfolgt die Kommunikation jedoch über ein Modbus RTU/TCP Gateway, erhält man von diesem eine Fehlermeldung, dass das angesprochene Gerät nicht antwortet.

Fehler werden vom Server mit einer entsprechenden Fehlermeldung an den Client zurückgeschickt.

*ibaPDA* meldet den Fehler beim Start entweder als Fehler oder als Warnung, abhängig vom Fehlercode:



#### 5.1.1 Telegrammaufbau

**Response Modbus-Server Client, mit Fehlermeldung:**

	Offs	Bytes	Typ	Modbus Beschreibung	Inhalt (hex)	ibaPDA Beschreibung
MBAP	00	2	UINT	Transaktions-ID	xx xx	Spiegel der Anfrage
	02	2	UINT	Protokoll-ID	00 00	0
	04	2	UINT	Cmd Länge	00 03	
	06	1	BYTE	Unit-ID	xx	Spiegel der Anfrage
Fcode	07	1	BYTE	Funktionscode "Exception"	0x80 + xx	Spiegel des Funktionscodes mit MSB=1
Daten	08	1	BYTE	Fehlercode	xx	Fehlercode

Der empfangene Funktions-Code wird kopiert und das höchstwertige Bit (MSB) gesetzt.

**Beispiel:**

Funktion 03: Read Holding Registers			
Request	(hex)	(hex)	Response
Trans-ID	00 03	00 03	Trans-ID
Prot-ID	00 00	00 00	Prot-ID
Cmd LEN	00 06	00 03	Cmd LEN
Unit-ID	01	01	Unit-ID
FCode	03	83	FCode mit Fehlerkennung
Start Adr (Hi)	00	02	Fehlercode
Start Adr (Lo)	6D		
Anz. Werte (Hi)	00		
Anz. Werte (Lo)	04		

**5.1.2 Modbus Fehlercodes**

Fehlercodex (hex)	Name	Bedeutung
01	Illegal Function	Verwendung eines nicht unterstützten Funktions-codes
02	Illegal Data Address	Verwendung einer ungültigen Speicher-Adresse oder Versuch auf eine schreibgeschützte Adresse zu schreiben
03	Illegal Data Value	Verwendung unerlaubter Datenwerte, z.B. eine unerlaubte Anzahl Register
04	Slave Device Failure	Nicht behebbarer Fehler
05	Acknowledge	Der Server braucht lange Zeit, um den Request zu bearbeiten. Um einen Timeout des Clients zu vermeiden, sendet er ein Acknowledge.
06	Slave Device Busy	Gerät kann zur Zeit keine Modbus-Befehle verarbeiten
0A	Gateway Path Unavailable	Gateway ist überlastet oder falsch projiziert
0B	Gateway Target Device Failed to Respond	Fehlermeldung des Gateways: Keine Antwort vom adressierten Gerät.



## 6 Support und Kontakt

### Support

Tel.: +49 911 97282-14  
Fax: +49 911 97282-33  
E-Mail: [support@iba-ag.com](mailto:support@iba-ag.com)

---

### Hinweis



Wenn Sie Support benötigen, dann geben Sie bitte bei Softwareprodukten die Nummer des Lizenzcontainers an. Bei Hardwareprodukten halten Sie bitte ggf. die Seriennummer des Geräts bereit.

---

### Kontakt

#### Hausanschrift

iba AG  
Königswarterstraße 44  
90762 Fürth  
Deutschland

Tel.: +49 911 97282-0  
Fax: +49 911 97282-33  
E-Mail: [iba@iba-ag.com](mailto:iba@iba-ag.com)

#### Postanschrift

iba AG  
Postfach 1828  
90708 Fürth

#### Warenanlieferung, Retouren

iba AG  
Gebhardtstraße 10  
90762 Fürth

#### Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite:

**[www.iba-ag.com](http://www.iba-ag.com)**