



# ibaPDA-Interface-S7-TCP/UDP

## Datenschnittstelle TCP/UDP zu SIMATIC S7

Handbuch  
Ausgabe 3.1

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
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 2024, 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
3.1	03-2024	Module length im Header muss gerade sein	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 TCP/UDP zu SIMATIC S7 .....</b>	<b>8</b>
3.1	Allgemeine Informationen .....	8
3.2	Konfiguration und Projektierung SIMATIC S7 .....	9
3.2.1	Datenbausteine .....	10
3.2.1.1	Header .....	11
3.2.1.2	Datenbereiche .....	12
3.2.2	S7-300/S7-400-CPU's ohne lokale Ethernet-Schnittstelle .....	13
3.2.3	S7-300/S7-400-CPU's mit lokaler Ethernet-Schnittstelle .....	16
3.2.4	S7-1200-CPU's mit lokaler Ethernet Schnittstelle .....	18
3.2.5	S7-1500-CPU's mit lokaler Ethernet Schnittstelle .....	20
3.3	Konfiguration und Projektierung ibaPDA .....	22
3.3.1	Allgemeine Einstellungen .....	22
3.3.2	Allgemeine Einstellungen der Schnittstelle .....	23
3.3.3	Modul hinzufügen .....	24
3.3.4	Allgemeine Moduleinstellungen .....	25
3.3.5	Signalkonfiguration .....	27
3.3.6	Modultyp S7 TCP/UDP Integer .....	27
3.3.7	Modultyp S7 TCP/UDP Real .....	28
3.3.8	Modultyp S7 TCP/UDP Allgemein .....	28
3.3.9	Modultypen S7 UDP Request/S7 UDP Request Decoder .....	29
3.3.10	Moduldiagnose .....	29

<b>4</b>	<b>Diagnose .....</b>	<b>30</b>
4.1	Lizenz .....	30
4.2	Sichtbarkeit der Schnittstelle .....	30
4.3	Protokolldateien .....	31
4.4	Verbindungsdiagnose mittels PING .....	32
4.5	Überprüfung der Verbindung .....	33
4.6	Diagnosemodule.....	35
<b>5</b>	<b>Anhang .....</b>	<b>41</b>
5.1	Fehlerbehebung.....	41
5.1.1	Probleme mit TCP-Performance durch Delayed Acknowledge .....	41
5.1.2	Unbrauchbare TCP-Daten als Folge des Nagle-Algorithmus .....	43
<b>6</b>	<b>Support und Kontakt.....</b>	<b>45</b>

# 1 Zu dieser Dokumentation

Diese Dokumentation beschreibt die Funktion und Anwendung der Software-Schnittstelle *ibaPDA-Interface-S7-TCP/UDP*.

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-S7-TCP/UDP* sind folgende Vorkenntnisse erforderlich bzw. hilfreich:

- Betriebssystem Windows
- Grundkenntnisse *ibaPDA*
- Kenntnis von Projektierung und Betrieb des betreffenden Steuerungssystems

## 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 S7 TCP/UDP erforderlich:

- *ibaPDA* v7.0.0 oder höher
- Lizenz für *ibaPDA-Interface-S7-TCP/UDP*
- Netzwerkanschluss 10/100 Mbit
- STEP 7 ab Version V4.0 oder TIA-Portal ab V11
- S7 CPU mit integriertem PN-Port oder Kommunikationsprozessor

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

- Unterschiedliche Behandlung des TCP/IP-Acknowledge  
siehe ➤ *Probleme mit TCP-Performance durch Delayed Acknowledge*, Seite 41 (alle *ibaPDA*-Versionen).

### Lizenzen

Bestell-Nr.	Produktbezeichnung	Beschreibung
31.001040	ibaPDA-Interface-S7-TCP/UDP	Erweiterungslizenz für ein <i>ibaPDA</i> -System um eine TCP/IP und UDP/IP Schnittstelle Anzahl der Verbindungen: 64
31.101040	one-step-up-Interface-S7-TCP/UDP	Erweiterungslizenz für eine vorhandene <i>ibaPDA-Interface-S7-TCP/UDP</i> Schnittstelle um 64 weitere S7-TCP/UDP-Verbindungen (maximal 3 Erweiterungslizenzen zulässig)

## 3 Datenschnittstelle TCP/UDP zu SIMATIC S7

### 3.1 Allgemeine Informationen

Die Schnittstelle S7-TCP/UDP ist geeignet, um Messdaten aus einer S7-Steuerung über die Standard-Netzwerkkarte des *ibaPDA*-PC mittels des Protokolls TCP/IP oder UDP zu erfassen. Hierzu ist eine Verbindungsprojektierung und Programmierung des Datenversendens in der Steuerung erforderlich.

Die Auswahl der zu messenden Signale erfolgt durch Rangieren der Werte in Datenbausteine (DB), deren Datenstrukturen durch die Modultypen von *ibaPDA* vorgegeben sind. Anschließend werden die Datenbausteine mit S7-Kommunikationsbausteinen als Telegramme an den *ibaPDA*-PC gesendet.

Im *ibaPDA-Interface-S7-TCP/UDP* sind drei Modultypen definiert:

Integer:	32 Analogwerte (Integer) und 32 Binärsignale
Real:	8, 16 oder 32 Analogwerte (Real) und 32 Binärsignale
Allgemein:	beliebige Datenstruktur mit maximaler Länge von 4096 Bytes <sup>1)</sup>

Jedes Modul ist einer Verbindung zugeordnet. Auf *ibaPDA*-Seite können max. 256 Verbindungen erstellt werden. Auf der S7-Seite hängt die max. Anzahl der Verbindungen vom CPU-Typ ab.

Der Hauptvorteil dieser Art der Messung liegt darin, dass keine spezielle Hardware benötigt wird, wenn bereits ein Ethernet-Anschluss in der Steuerung existiert.



<sup>1)</sup> Bis *ibaPDA* V6.30 begrenzt durch Eigenschaften der S7-Kommunikationsprozessoren



## TCP und UDP

Das Transmission Control Protocol, kurz TCP, ist ein verbindungsorientiertes Protokoll und soll maßgeblich Datenverluste verhindern, Dateien und Datenströme aufteilen und Datenpakete Anwendungen zuordnen können.

Das User Datagram Protocol, kurz UDP, ist ein verbindungsloses Transport-Protokoll. Es hat damit eine vergleichbare Aufgabe, wie das verbindungsorientierte TCP. Allerdings arbeitet es verbindungslos und damit unsicher. Das bedeutet, der Absender weiß nicht, ob seine verschickten Datenpakete angekommen sind. Während TCP Bestätigungen beim Datenempfang sendet, verzichtet UDP darauf. Das hat den Vorteil, dass der Paket-Header viel kleiner ist und die Übertragungsstrecke keine Bestätigungen übertragen muss. Prinzipiell ist damit eine etwas höhere Datenrate möglich.

Beide arbeiten mit dem Internetprotokoll IP, der Schicht 4 (Transportschicht) des OSI-Schichtenmodells.

---

### Hinweis



Wenn in den folgenden Beispielen auch bei UDP von Verbindungen die Rede ist, so bezieht sich diese nicht auf eine aufzubauende und abzubauenende Netzwerkverbindung, sondern bezeichnet nur den Kommunikationsweg vom Sender zum Empfänger.

---

## 3.2 Konfiguration und Projektierung SIMATIC S7

Dieser Abschnitt beschreibt die Vorgehensweise bei der Erstellung der TCP/IP- bzw. UDP-Verbindung, der notwendigen Datenbausteine und der Parametrierung der Sendebausteine. Abhängig von der CPU-Familie gibt es dafür verschiedene Varianten.

- CPUs ohne lokale Ethernet-Schnittstelle  
Die Projektierung auf der S7-Seite wird mit den STEP7-Werkzeugen "HW-Konfig" und "NetPro" durchgeführt, welche im SIMATIC Manager enthalten sind. Im Programm STEP 7 fügen Sie Sendebausteine (AG\_SEND, AG\_LSEND) ein, welche die projektieren Verbindungen verwenden.
- CPUs mit lokaler Ethernet-Schnittstelle  
Die Verbindungen müssen Sie nicht gesondert projektieren. Sowohl der Verbindungsaufbau als auch das Senden erfolgt in STEP 7 mit Standardbausteinen (TCON, TSEND, TUSEND).
- Die CPUs der Reihe S7-1200 und S7-1500  
Diese haben generell lokale Ethernet-Schnittstellen und sind nur mit dem TIA Portal projektierbar. Hier gibt es Bausteine, die sowohl den Verbindungsaufbau als auch das Datensenden übernehmen (TSEND\_C).

---

**Hinweis**

Bei allen im folgenden beschriebenen Verbindungsarten ist zu beachten:

- Bei allen Verbindungen ist die S7-CPU der aktive Partner.
  - Der Partner-Port muss mit der Einstellung in *ibaPDA* (Interface S7-TCP-UDP) übereinstimmen (Vorbereitung in *ibaPDA*: 4170).
  - Dieser Port muss in dem *ibaPDA*-PC in der Windows Firewall freigegeben werden.
  - Dieser Port darf nicht anderweitig vergeben sein.
  - Beachten Sie bei der Erstellung weiterer Verbindungen:
    - Vergeben Sie immer neue Verbindungsnamen.
    - Vergeben Sie immer neue lokale Portnummer.
    - Verwenden Sie immer dieselbe Partner-IP-Adresse.
    - Verwenden Sie immer dieselbe Partner-Portnummer.
- 

### 3.2.1 Datenbausteine

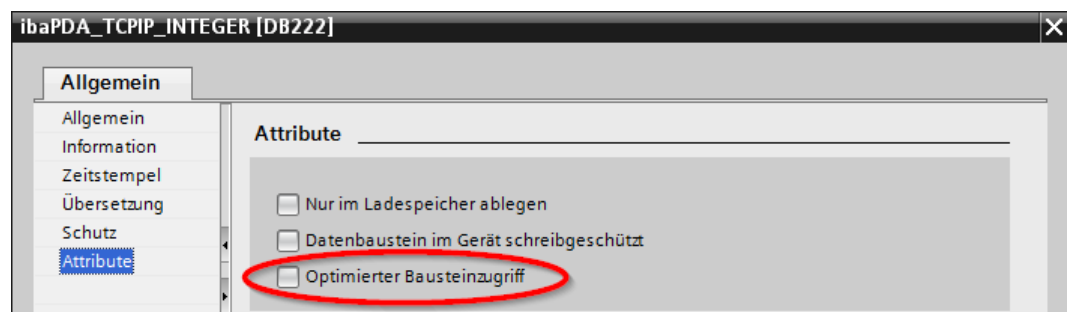
Bei allen oben genannten Methoden müssen die zu sendenden Daten in Datenbausteinen bereitgestellt werden.

Entsprechend der *ibaPDA*-Modulstruktur werden die Daten pro Modul mit einem Telegramm übertragen. Jedem Telegramm liegt ein Datenbaustein und eine Verbindung zu Grunde. Die Datenbausteine haben einen einheitlichen Header und eine dem Modultyp entsprechenden Datenstruktur.

---

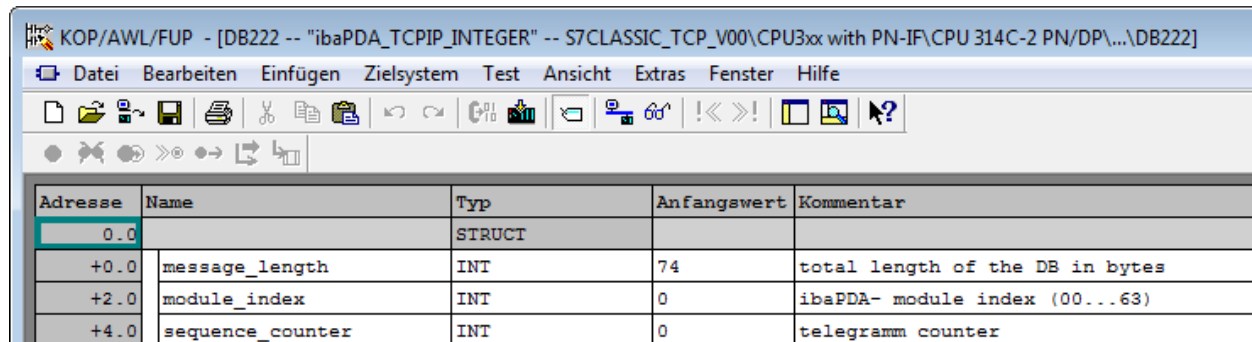
**Hinweis****Besonderheit für S7-1200-/S7-1500-Steuerungen:**

Deaktivieren Sie bei den Telegramm-Datenbausteinen das Bausteinattribut *Optimierter Bausteinzugriff*. Der Zugriff auf optimierte Datenbausteine bei S7-1200-/S7-1500-Steuerungen wird nicht unterstützt.



### 3.2.1.1 Header

Der Header besteht aus 3 Integer-Werten.



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	message_length	INT	74	total length of the DB in bytes
+2.0	module_index	INT	0	ibaPDA- module index (00...63)
+4.0	sequence_counter	INT	0	telegramm counter

#### ■ message\_length

Gesamtgröße (in Bytes) des Datenpakets. Message\_length muss eine gerade Anzahl von Bytes sein. Der Wert darf während der Datenübertragung nicht verändert werden. Dieser Wert ist auch beim Aufruf des Sendebausteins anzugeben.

Die Länge ist abhängig vom Modultyp:

bei Modultyp Integer:	74
bei Modultyp Real:	42, 74 oder 138 (bei 8, 16 oder 32 Reals)
bei Modultyp All:emein	8...4096

#### ■ module\_index

Kennung für die Zuordnung des Datensatzes zu dem Interface Modul in *ibaPDA*. In diesem Index ist auch der Modultyp verschlüsselt: Der Index wird durch eine laufende Nummer 00...63 und einem dem Modultyp und der Lizenz entsprechenden Offset gebildet.

Modultyp	1. Lizenz	2. Lizenz	3. Lizenz	4. Lizenz
Integer	0–63	1000–1063	2000–2063	3000–3063
Real	100–163	1100–1163	2100–2163	3100–3163
Allgemein	200–263	1200–1263	2200–2263	3200–3263

Der Modulindex entspricht dem Index in der *ibaPDA* Moduleinstellung. Der Wert muss eindeutig sein und darf während der Datenübertragung nicht verändert werden.

#### ■ sequence\_counter

Mit jedem erfolgreichen Sendeauftrag wird der Wert um 1 hochgezählt. Dies muss in der S7 programmiert werden. Wenn sich der Wert des Zählers nicht um +1 ändert, zeigt *ibaPDA* in der Verbindungsliste einen Sequenzfehler an.

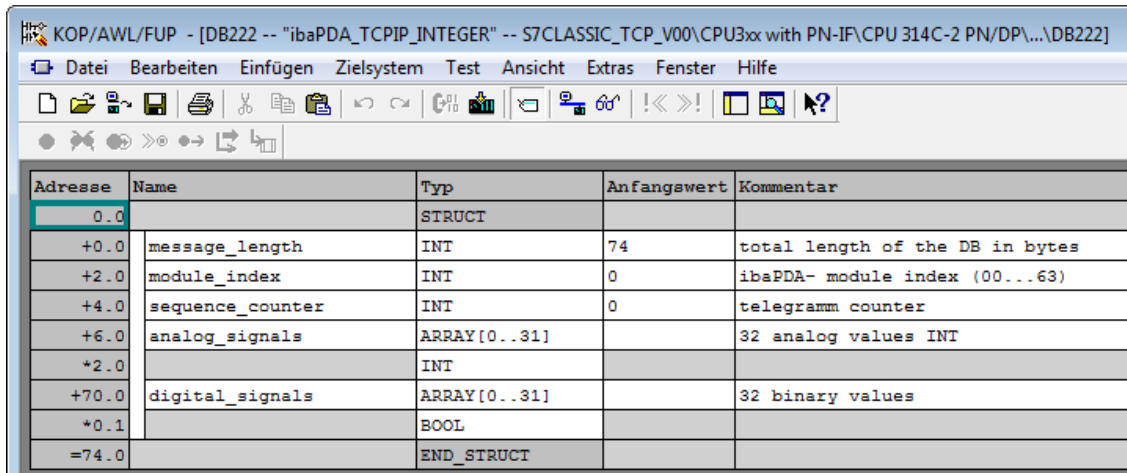
Bei Überlauf muss der Zähler von 32767 auf -32768 springen (0x7FFF → 0x8000).

### 3.2.1.2 Datenbereiche

Der Aufbau des Datenbereichs ist abhängig vom Modultyp.

#### Modultyp Integer

Nach dem Header stehen ab Offset 6 die 32 Integer-Analogwerte und anschließend ab Offset 70 die 4 Bytes Binärwerte.



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	message_length	INT	74	total length of the DB in bytes
+2.0	module_index	INT	0	ibaPDA- module index (00...63)
+4.0	sequence_counter	INT	0	telegramm counter
+6.0	analog_signals	ARRAY[0..31]		32 analog values INT
+2.0		INT		
+70.0	digital_signals	ARRAY[0..31]		32 binary values
+0.1		BOOL		
=74.0		END_STRUCT		

#### Hinweis



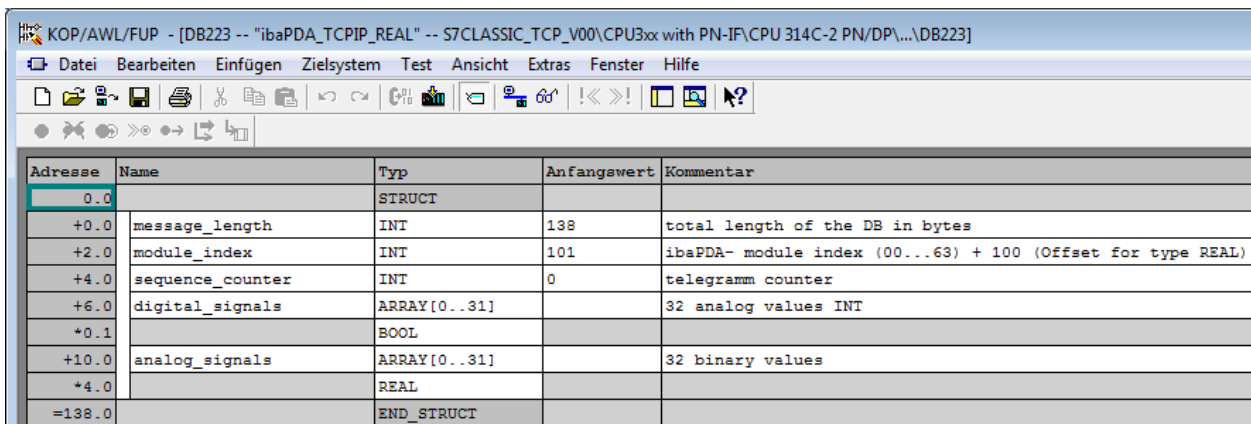
Beachten Sie bitte die unterschiedliche Byte-Ordnung zwischen S7 und *ibaPDA*.

Beispiel:

Wenn Sie das Bit DB222.DBX70.0 setzen, kommt dieses in *ibaPDA* als Bit 24 an.  
Wenn Sie aber DB222.DBD70 mit 16#00000001 beschreiben, ist in *ibaPDA* das Bit 0 gesetzt.

#### Modultyp Real

Nach dem Header stehen ab Offset 6 die 4 Bytes Binärwerte und anschließend ab Offset 10 entweder 8, 16 oder 32 Analogwerte im Real-Format.



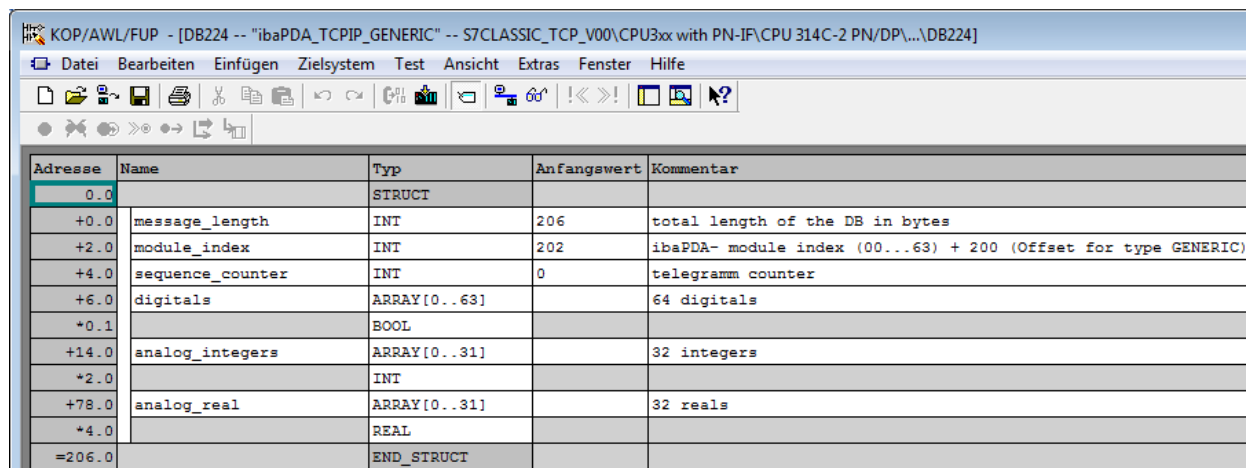
Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	message_length	INT	138	total length of the DB in bytes
+2.0	module_index	INT	101	ibaPDA- module index (00...63) + 100 (Offset for type REAL)
+4.0	sequence_counter	INT	0	telegramm counter
+6.0	digital_signals	ARRAY[0..31]		32 analog values INT
+0.1		BOOL		
+10.0	analog_signals	ARRAY[0..31]		32 binary values
+4.0		REAL		
=138.0		END_STRUCT		

## Modultyp Allgemein

Nach dem Header kann ab Offset 6 eine beliebige Folge von Daten mit unterschiedlichen Datentypen folgen. *ibaPDA* unterstützt folgende Datenformate:

BYTE, WORD, DWORD, INT, DINT und FLOAT

In *ibaPDA* muss die hier definierte Datenstruktur nachgebildet werden. Dabei können die BYTE-, WORD- und DWORD-Variablen auch als 8, 16 oder 32 einzelne Bits interpretiert werden (und umgekehrt).



Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	message_length	INT	206	total length of the DB in bytes
+2.0	module_index	INT	202	ibaPDA- module index (00...63) + 200 (Offset for type GENERIC)
+4.0	sequence_counter	INT	0	telegramm counter
+6.0	digitals	ARRAY[0..63]		64 digitals
+0.1		BOOL		
+14.0	analog_integers	ARRAY[0..31]		32 integers
+2.0		INT		
+78.0	analog_real	ARRAY[0..31]		32 reals
+4.0		REAL		
=206.0		END_STRUCT		

### 3.2.2 S7-300/S7-400-CPUs ohne lokale Ethernet-Schnittstelle

Die Projektierung der Verbindung bei CPU ohne lokalen Ethernet-Port erfolgt in folgenden Schritten:

1. SIMATIC NET Ethernet-CP in HW-Konfig konfigurieren
2. TCP- oder UDP-Verbindungen in NetPro anlegen
3. Programmcode zum Rangieren der Signaldaten in die Datenbausteine
4. Programmcode zum Inkrementieren der "sequence\_counter"
5. Kommunikationsbausteine im S7-Programm der CPU aufrufen

Für jedes in *ibaPDA* projektierte Modul müssen Sie eine eigene Verbindung projektieren.

#### Ethernet-CP in HW-Konfig konfigurieren

1. Wählen Sie die PN-IO-Schnittstelle des Ethernet-CPs (CP343-1 bzw. CP443-1).
2. Öffnen Sie die *Eigenschaften* im Register *Allgemein*.
3. Vergeben Sie die IP-Adresse und Subnetmaske der S7-Steuerung.

## Verbindung in NetPro anlegen

Wählen Sie folgende Parameter:

Verbindungspartner (Station):	unspezifiziert
Verbindung – Typ:	TCP-Verbindung oder UDP-Verbindung
Allgemein – Name:	vergeben Sie einen eindeutigen Namen
Allgemein – Aktiver Verbindungsaufbau:	
■ bei TCP-Verbindung:	aktiviert
■ bei UDP-Verbindung:	der Parameter ist nicht vorhanden (immer passiv)
Adressen – Lokal – Port:	Vergeben Sie eine eindeutige Portnummer.
Adressen – Partner – IP:	IP-Adresse des <i>ibaPDA</i> -PC
Adressen – Partner – Port:	Portnummer des <i>ibaPDA</i> -PC

### Hinweis



Für die Vergabe der Portnummern beachten Sie folgende Hinweise ➔ *Konfiguration und Projektierung SIMATIC S7*, Seite 9.

## Signaldaten rangieren

Kopieren Sie zyklisch an einer beliebigen Stelle Ihres S7-Programms die gewünschten Signaldaten in die Datenbausteine der Telegrammmodule.

### "sequence\_counter" inkrementieren

Erhöhen Sie mit jeder positiven Flanke des Ausgangs DONE den "sequence\_counter" im Telegrammdatenbaustein.

Setzen Sie den "sequence\_counter" bei CPU-Neustart zurück auf 0. Bei Überlauf muss der Zähler von 32767 auf -32768 springen (0x7FFF → 0x8000).

## Kommunikationsbausteine aufrufen

Je nach verwendeten CPU-Typ sind unterschiedliche SIMATIC-Kommunikationsbausteine notwendig.

Kommunikationsbaustein	verwendbar bei		Bemerkung
	S7-300	S7-400	
AG_SEND (FC5)	X		Mit diesen Bausteinen können bis zur Bausteinversion V3.0 maximal 240 Byte gesendet werden. Die aktuellen Bausteinversionen lassen einen Datenbereich bis zu 8192 Bytes bei TCP und 2048 Bytes bei UDP zu.
		X	Bei S7-400 ist die Datenlänge begrenzt auf 240 Bytes. Für die Übertragung größerer Datenbereiche muss der Baustein AG_LSEND verwendet werden. <sup>2)</sup>
AG_LSEND (FC50)		X	Die maximale Datenlänge beträgt 8192 Bytes bei TCP und 2048 Bytes bei UDP. Bitte informieren Sie sich in der Produktinformation des CP bzgl. des unterstützten Datenbereiches.

Die Referenz auf die in NetPro projektierte Verbindung erfolgt über den Parameter ID (1. Teil des lokalen IDs der NetPro-Verbindung) und die den Parameter LADDR (HW-Adresse aus der NetPro-Verbindung).

### Hinweis



Verwenden Sie in Ihrem STEP7-Projekt immer die aktuelle Version der SIMATIC NET-Kommunikationsbausteine. Diese finden Sie im SIMATIC Manager unter *Datei – Öffnen – Bibliotheken – SIMATIC\_NET\_CP*.

### Tipp



Bei einem zyklischen Aufruf des AG\_SEND/AG\_LSEND ist der Sendezyklus max. halb so groß wie der Aufrufzyklus, da im Aufruf im Zyklus nach dem Sendeaufruf die Ausgabeparameter des Bausteins aktualisiert werden. Sie können das vermeiden, wenn Sie in jedem Zyklus den Baustein zweimal aufrufen. Das 1. Mal um den Status abzufragen (ACT=0), beim 2. Mal um die Daten zu senden (ACT=1).

### Andere Dokumentation



Weitere Informationen zur Kommunikationsprojektierung finden Sie in der STEP 7-Online-Hilfe und in folgender FAQ von Siemens:

Projektierung einer TCP-Verbindung über Ethernet (TCP nativ) zwischen einer SIMATIC S7 und einem PC mit Socket Interface

<https://support.industry.siemens.com/cs/de/de/view/18843927>

<https://support.industry.siemens.com/cs/de/de/view/17853532>

<https://support.industry.siemens.com/cs/de/de/view/18513371>

<sup>2)</sup> Siehe auch SIMATIC STEP 7 Hilfe Thema "FCs für die SEND/RECEIVE-Schnittstelle".

### 3.2.3 S7-300/S7-400-CPU mit lokaler Ethernet-Schnittstelle

Die Projektierung der Verbindung bei CPU mit lokalen Ethernet-Port erfolgt in folgenden Schritten:

1. Ethernet-Schnittstelle der CPU in HW-Konfig konfigurieren
2. Programmcode zum Rangieren der Signaldaten in den Datenbaustein
3. Programmcode zum Inkrementieren des "sequence\_counter"
4. Verbindungsdaten anlegen und parametrieren (Datenstrukturen TCON\_PAR und ggf. TADDR\_PAR)
5. Kommunikationsbausteine im S7-Programm der CPU aufrufen

Für jedes in *ibaPDA* projektierte Modul müssen Sie eine eigene Verbindung projektieren.

#### Ethernet-Schnittstelle der CPU in HW-Konfig konfigurieren

1. Wählen Sie die PN-IO-Schnittstelle der CPU.
2. Öffnen Sie die *Eigenschaften* im Register *Allgemein*.
3. Vergeben Sie die IP-Adresse und Subnetmaske der S7-Steuerung.

#### Signaldaten rangieren

Kopieren Sie zyklisch an einer beliebigen Stelle Ihres S7-Programms die gewünschten Signaldaten in die Datenbausteine der Telegrammmodule.

#### "sequence\_counter" inkrementieren

Erhöhen Sie mit jeder positiven Flanke des Ausgangs DONE den "sequence\_counter" im Telegrammdatenbaustein.

Setzen Sie den "sequence\_counter" bei CPU-Neustart zurück auf 0. Bei Überlauf muss der Zähler von 32767 auf -32768 springen (0x7FFF → 0x8000).

#### Parametrierung der Verbindung

Legen Sie eine statische Variable oder einen DB mit der Datenstruktur TCON\_PAR (UDT65) an und tragen Sie folgende Parameter ein:

id:	Verbindungs-ID, Referenz zum dazugehörigen TCON- und TSEND- bzw. TUSEND-Baustein	
connection_type:	bei TCP: B#16#11	bei UDP: B#16#13
active_est:	bei TCP: TRUE	bei UDP: FALSE
local_device_id:	2, 3 oder 5 (abhängig von CPU-Typ) <sup>3)</sup>	
local_tsap_id:	eindeutige Portnummer für jede Verbindung	
rem_staddr:	nur bei TCP: IP-Adresse des <i>ibaPDA</i> -PC	
rem_tsap_id:	nur bei TCP: Portnummer des <i>ibaPDA</i> -PC	

<sup>3)</sup> Siehe Online-Hilfe zu Systemfunktionen Thema "Parametrierung der Kommunikationsverbindungen..."



Bei UDP wird die Remote-IP-Adresse und Remote-Portnummer nicht aus den Verbindungsdaten genommen, sondern muss in einem getrennten Datenbereich mit Struktur TADDR\_PAR (UDT66) hinterlegt werden:

rem_ip_addr	IP-Adresse des <i>ibaPDA</i> -PC
rem_port_nr	Portnummer des <i>ibaPDA</i> -PC

---

#### Hinweis



Für die Vergabe von Portnummern beachten Sie folgende Hinweise siehe  
➔ *Konfiguration und Projektierung SIMATIC S7*, Seite 9.

---

#### Kommunikationsbausteine im S7-Programm aufrufen

Folgende Kommunikationsbausteine kommen zum Einsatz:

- TCON (FB65): zum Aufbau der Verbindung
- TSEND (FB63): zum Senden der Daten per TCP
- TUSEND (FB67): zum Senden der Daten per UDP

Die Parametrierung der Verbindung erfolgt über den Datenbereich mit der vorgegebenen Struktur, der über die eindeutige Verbindungs-ID referenziert wird.

---

#### Hinweis



Verwenden Sie in Ihrem STEP 7-Projekt immer die aktuelle Version der SIMATIC NET-Kommunikationsbausteine. Diese finden Sie im SIMATIC Manager unter *Datei – Öffnen – Bibliotheken – Standard Library – Communication Blocks*.

---

#### Andere Dokumentation



Weitere Informationen zur Kommunikationsprojektierung finden Sie in der STEP 7 Online-Hilfe und in folgender FAQ von Siemens:

Wie werden die Kommunikationsbausteine FB63 "TSEND", FB64 "TRCV", FB65 "TCON" und FB66 "TDISCON" programmiert, um das TCP Protokoll für den Datenaustausch über die integrierte PROFINET-Schnittstelle einer S7-300 bzw. S7-400 CPU zu nutzen?

<https://support.industry.siemens.com/cs/de/de/view/29737950>

---

### 3.2.4 S7-1200-CPU mit lokaler Ethernet Schnittstelle

#### Hinweis



Folgende Hinweise gelten für die Firmware-Versionen bis 3.x.

Für S7-1200 ab Version 4.0 sind die Einstellungen identisch zu den S7-1500-CPU.

Siehe dazu auch ➔ *S7-1500-CPU mit lokaler Ethernet Schnittstelle*, Seite 20.

Die Projektierung der Verbindung bei CPU mit lokalem Ethernet-Port erfolgt in folgenden Schritten:

1. Ethernet-Schnittstelle der CPU in der Gerätekonfiguration konfigurieren
2. Programmcode zum Rangieren der Signaldaten in den Datenbaustein
3. Programmcode zum Inkrementieren des "sequence\_counter"
4. Verbindungsdaten anlegen und parametrieren (Datenstruktur TCON\_Param)
5. Kommunikationsbausteine im S7-Programm der CPU aufrufen

Für jedes in *ibaPDA* projektierte Modul müssen Sie eine eigene Verbindung projektieren.

#### Ethernet-Schnittstelle in der Gerätekonfiguration konfigurieren

1. Wählen Sie die Gerätekonfiguration.
2. Klicken Sie in der Grafik auf den Ethernet-Port, der mit *ibaPDA* verbunden ist.
3. Wählen Sie das Register *Allgemein – Ethernet-Adresse*.
4. Vergeben Sie die IP-Adresse und Subnetmaske der S7-Steuerung.

#### Signaldaten rangieren

Kopieren Sie zyklisch an einer beliebigen Stelle Ihres S7-Programms die gewünschten Signaldaten in die Datenbausteine der Telegrammmodule.

#### "sequence\_counter" inkrementieren

Erhöhen Sie mit jeder positiven Flanke des Ausgangs DONE den "sequence\_counter" im Telegrammdatenbaustein.

Setzen Sie den "sequence\_counter" bei CPU-Neustart zurück auf 0. Bei Überlauf muss der Zähler von 32767 auf -32768 springen (0x7FFF → 0x8000).

#### Verbindung parametrieren

Legen Sie eine statische Variable oder einen DB mit der Datenstruktur TCON\_Param an und tragen Sie folgende Parameter ein:

local_device_id:	Kennung der lokalen Schnittstelle: 1	
id:	eindeutige Verbindungs-ID	
connection_type:	bei TCP: 17	bei UDP: 19
active_est:	bei TCP: TRUE	bei UDP: FALSE
rem_staddr_len:	bei TCP: 4	bei UDP: 0
rem_tsap_id_len:	bei TCP: 2	bei UDP: 0
remote_staddr:	nur bei TCP: IP-Adresse des <i>ibaPDA</i> -PC	
remote_tsap_id:	nur bei TCP: Portnummer des <i>ibaPDA</i> -PC: 4170	
local_tsap_id_len:	2	
local_tsap_id:	eindeutige Portnummer für jede Verbindung	

Bei UDP wird die Remote-IP-Adresse und Remote-Portnummer nicht aus den Verbindungsdaten genommen, sondern muss in einem getrennten Datenbereich mit Struktur TADDR\_Param hinterlegt werden:

rem_ip_addr	IP-Adresse des <i>ibaPDA</i> -PC
rem_port_nr	Portnummer des <i>ibaPDA</i> -PC

---

#### Hinweis



Für die Vergabe der Portnummern, beachten Sie die Hinweise in [↗ Konfiguration und Projektierung SIMATIC S7](#), Seite 9.

---

#### Kommunikationsbausteine aufrufen

Folgende Kommunikationsbausteine kommen zum Einsatz:

- TSEND\_C: zum Aufbau der Verbindung und Senden der Daten bei TCP/IP
- TCON : zum Aufbau der Verbindung bei UDP
- TUSEND: Zum Senden der Daten bei UDP

Die Parametrierung der Verbindung erfolgt über einen Datenbereich mit der vorgegebenen Struktur, der über die eindeutige Verbindungs-ID referenziert wird.

---

#### Andere Dokumentation



Weitere Informationen zur Kommunikationsprojektierung finden Sie in der TIA-Online-Hilfe und in folgender FAQ von Siemens:

Wie werden die Anweisungen TSEND\_C und TRCV\_C für die offene Benutzerkommunikation über die integrierte PROFINET-Schnittstelle der S7-1200 CPU programmiert?

<https://support.industry.siemens.com/cs/de/de/view/67196808>

### 3.2.5 S7-1500-CPU mit lokaler Ethernet Schnittstelle

#### Hinweis



Folgende Anweisungen gelten auch für S7-1200 ab Version 4.0.

Die Projektierung der Verbindung bei CPU mit lokalen Ethernet-Port erfolgt in folgenden Schritten:

1. Ethernet-Schnittstelle der CPU in der Gerätekonfiguration konfigurieren
2. Programmcode zum Rangieren der Signaldaten in den Datenbaustein
3. Programmcode zum Inkrementieren des „sequence\_counter“
4. Verbindungsdaten anlegen und parametrieren (Datenstruktur TCON\_IP\_v4)
5. Kommunikationsbausteine im S7-Programm der CPU aufrufen

Für jedes in *ibaPDA* projektierte Modul müssen Sie eine eigene Verbindung projektieren.

#### Ethernet-Schnittstelle in der Gerätekonfiguration konfigurieren

1. Wählen Sie die Gerätekonfiguration.
2. Klicken Sie in der Grafik auf den Ethernet-Port, der mit *ibaPDA* verbunden ist.
3. Wählen Sie das Register *Allgemein – Ethernet-Adresse*.
4. Vergeben Sie die IP-Adresse und Subnetmaske der S7-Steuerung.

#### Signaldaten rangieren

Kopieren Sie zyklisch an einer beliebigen Stelle Ihres S7-Programms die gewünschten Signaldaten in die Datenbausteine der Telegrammmodule.

#### "sequence\_counter" inkrementieren

Erhöhen Sie mit jeder positiven Flanke des Ausgangs DONE den "sequence\_counter" im Telegrammdatenbaustein.

Setzen Sie den "sequence\_counter" bei CPU-Neustart zurück auf 0. Bei Überlauf muss der Zähler von 32767 auf -32768 springen (0x7FFF → 0x8000).

#### Verbindung parametrieren

Legen Sie eine lokale Variable oder einen DB mit der Datenstruktur TCON\_Param an und tragen Sie folgende Parameter ein:

interface_id:	Hardware-Kennung der lokalen Schnittstelle: siehe Gerätekonfiguration	
id:	Verbindungs-ID	
connection_type:	bei TCP: 11	bei UDP: 19

active_est:	bei TCP: TRUE	bei UDP: FALSE
remote_address:	nur bei TCP: IP-Adresse des <i>ibaPDA</i> -PC	
remote_port:	nur bei TCP: Portnummer des <i>ibaPDA</i> -PC: 4170	
local_port:	eindeutige Portnummer für jede Verbindung	

Bei UDP wird die Remote-IP-Adresse und Remote-Portnummer nicht aus den Verbindungsdaten genommen, sondern muss in einem getrennten Datenbereich mit Struktur UDT 66 ("TADDR\_Param") hinterlegt werden:

rem_ip_addr	IP-Adresse des <i>ibaPDA</i> -PC
rem_port_nr	Portnummer des <i>ibaPDA</i> -PC

---

#### Hinweis



Für die Vergabe der Portnummern beachten Sie die Hinweise in [↗ Konfiguration und Projektierung SIMATIC S7](#), Seite 9.

---

#### Kommunikationsbausteine aufrufen

Folgender Kommunikationsbaustein kommt zum Einsatz:

- TSEND\_C : zum Aufbau der Verbindung und Senden der Daten

Die Parametrierung der Verbindung erfolgt über einen Datenbereich mit der vorgegebenen Struktur, der über eine eindeutige Verbindungs-ID referenziert wird.

---

#### Hinweis



Der Baustein TSEND\_C liegt in zwei unterschiedlichen Varianten vor.

- Für CPU S7-1200 ≤ V3.x (siehe [↗ S7-1200-CPUs mit lokaler Ethernet Schnittstelle](#), Seite 18)
- Für CPU S7-1500 und CPU S7-1200 ≥ V4.0

Die Unterschiede beziehen sich auf unterschiedliche Systemdatentypen bei der Verbindungsparametrierung.

---

#### Andere Dokumentation



Weitere Informationen zur Kommunikationsprojektierung finden Sie in folgender FAQ von Siemens:

Wie werden die Anweisungen TSEND\_C und TRCV\_C für die offene Benutzerkommunikation über die integrierte PROFINET-Schnittstelle der S7-1200/S7-1500 CPU programmiert?

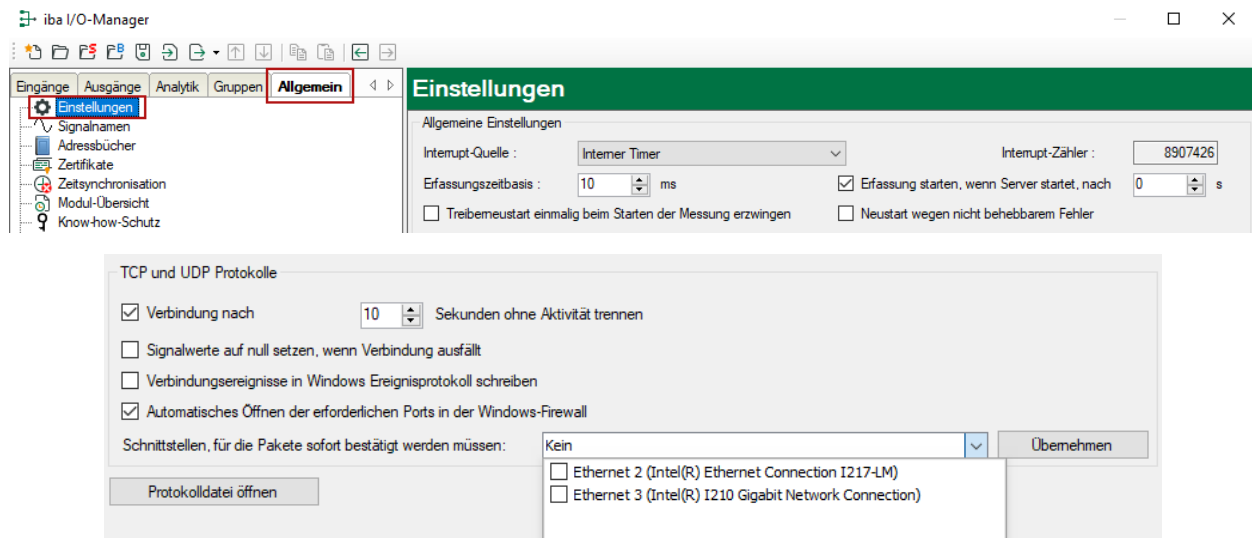
<https://support.industry.siemens.com/cs/de/de/view/67196808>

### 3.3 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 *S7 TCP/UDP* an.

#### 3.3.1 Allgemeine Einstellungen

Das "Totmann-Timeout" konfigurieren Sie für alle von *ibaPDA* unterstützten TCP- und UDP-Protokolle gemeinsam.



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

Verhalten und Timeout-Zeit ist vorgebar.

#### 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

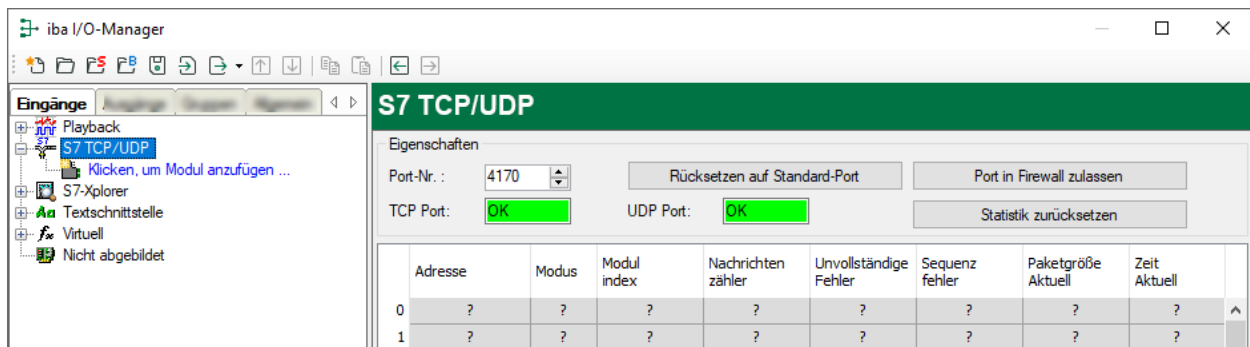
#### Hinweis



Wenn *ibaPDA* der aktive Partner (Client) ist, baut *ibaPDA* nach wenigen Sekunden die Verbindung wieder auf, um dem passiven Partner die Möglichkeit zu geben, wieder Daten zu senden.

### 3.3.2 Allgemeine Einstellungen der Schnittstelle

Die Schnittstelle hat folgende Funktionen und Konfigurationsmöglichkeiten.



#### Port-Nr.

Verwendeter Port im Rechner. Sie können die Portnummer ändern, aber in der S7-Projektierung und in *ibaPDA* müssen Sie denselben Port verwenden, um eine Verbindung herzustellen.

Die Standard-Portnummer lautet 4170.

#### <Rücksetzen auf Standard-Port>

Mit diesem Button können Sie den Port auf die Standard-Portnummer zurücksetzen.

#### <Ports in Firewall zulassen>

Bei der Installation von *ibaPDA* werden die Standard-Portnummern der verwendeten Protokolle automatisch in der Firewall eingetragen. Wenn Sie die Portnummer hier verändern oder das Interface nachträglich freischalten, müssen Sie über diesen Button diesen Port in der Firewall zulassen.

#### TCP Port/UDP Port

Anzeige zum Port-Status.

- OK: Sie können den Socket auf diesem Port öffnen.
- FEHLER: Es gibt Konflikte, z. B. der Port ist schon anderweitig belegt.

#### <Statistik zurücksetzen>

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

#### Verbindungstabelle

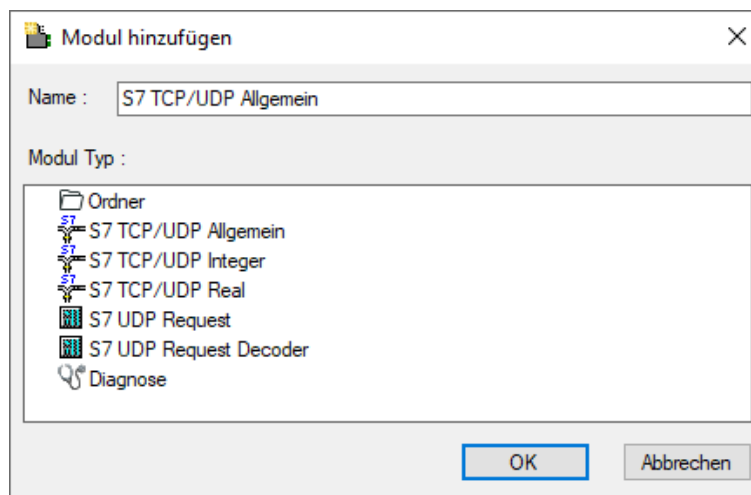
Die Tabelle zeigt für jede Verbindung den Verbindungsstatus, die aktuellen Werte für die Aktualisierungszeit (Aktuell, Istwert, Mittelwert, Min. und Max.) sowie die Datengröße. Außerdem finden Sie hier einen Fehlerzähler für die einzelnen Verbindungen während der Messung.

Siehe ➤ *Überprüfung der Verbindung*, Seite 33

### 3.3.3 Modul hinzufügen

#### Vorgehen

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>.



#### Tipp



Wenn bereits eine TCP/IP- oder UDP-Verbindung zu S7 besteht, klicken Sie mit der rechten Maustaste auf das Interface und wählen Sie *Autom. Erkennung*. Dann werden für alle vorhandenen Verbindungen automatisch die richtigen Module angelegt.

#### Modultypen

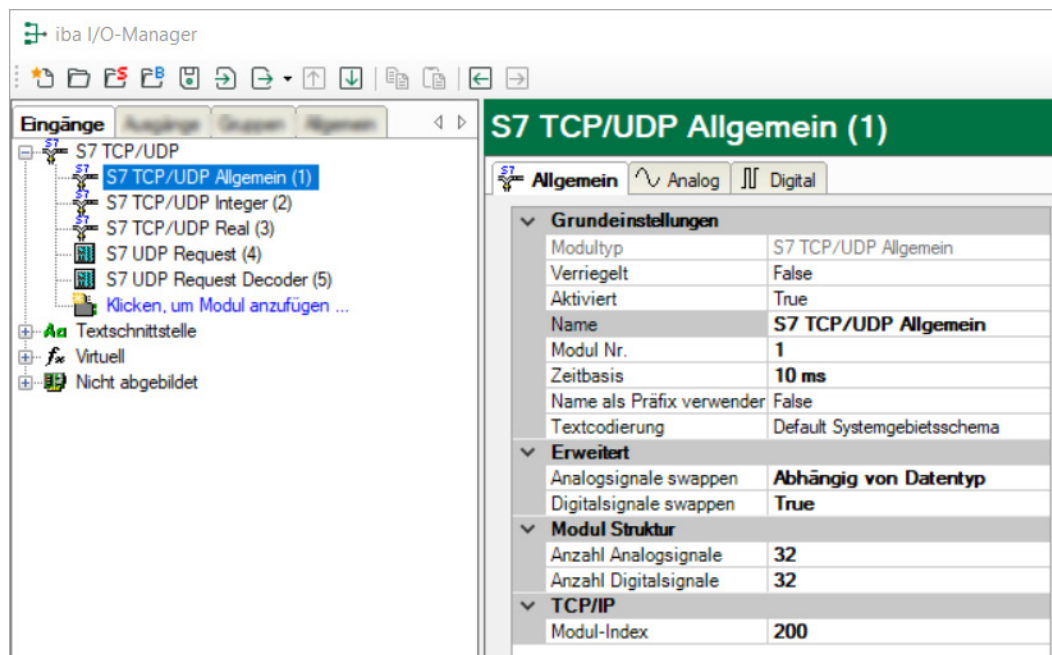
Modultyp	Beschreibung
S7 TCP/UDP Allgemein	Modul für beliebige Datenstruktur mit bis zu 4096 Bytes Länge
S7 TCP/UDP Integer	Modul für bis zu 32 analoge Signale (Integer) und 32 digitale Signale
S7 TCP/UDP Real	Modul für bis zu 32 analoge Signale (Real) und 32 digitale Signale
S7 UDP Request	Request-Modul für maximal 1024 analoge und 1024 digitale Signale Beschreibung siehe Handbuch <i>ibaPDA-Request-S7-UDP</i>
S7 UDP Request Decoder	Request-Modul für maximal 11728 digitale Signale, die in Form von max. 733 Wörtern (1466 Byte) übertragen werden. Beschreibung siehe Handbuch <i>ibaPDA-Request-S7-UDP</i>
Diagnose	Typ zur Erstellung von Diagnosemodulen



### 3.3.4 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.

##### Modulname als Präfix verwenden

Diese Option stellt den Modulnamen den Signalnamen voran.

**Textcodierung**

Für eine korrekte Interpretation und Anzeige der empfangenen Textdaten bei Eingängen bzw. der zu sendenden Textdaten bei Ausgängen können Sie hier die Form der Textcodierung, d. h. die Codepage auswählen. Zur Auswahl stehen neben dem Default-Systemgebietsschema gem. der Windows-Systemeinstellung und UTF-8 Unicode auch alle anderen üblichen Codierungen.

**Erweitert****Analogsignale swappen/Digitalsignale swappen**

Möglichkeit die Auswertereihenfolge der Byte zu ändern

**Modulstruktur****Anzahl Analogsignale/Anzahl 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.

**TCP/IP****Modul-Index**

Die Modulindizes werden durch eine laufende Nummer 00...63 und einem dem Modultyp und der Lizenz entsprechenden Offset gebildet.

Siehe auch ➤ *Header*, Seite 11.

---

**Andere Dokumentation**

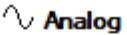
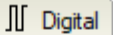
Eine detaillierte Beschreibung der Parameter finden Sie im *ibaPDA*-Handbuch.

---

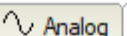
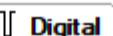
### 3.3.5 Signalkonfiguration

Die Auswahl der zu messenden Daten erfolgt auf SIMATIC S7-Seite durch Rangieren der Signale in Datenbausteine, die zyklisch an *ibaPDA* gesendet werden.

Im I/O-Manager können Sie Namen, Kommentare, ggf. Einheit und Skalierungsfaktor für die Signale eintragen sowie die Signale aktivieren.

S7 Allgemein  Analog  Digital							
	Name	Einheit	Min	Max	Aktiv	Istwert	±
0	Sinus		-32768	32767	<input checked="" type="checkbox"/>	-2	^
1	Cosinus		-32768	32767	<input checked="" type="checkbox"/>	57	
2	Triangle		-32768	32767	<input checked="" type="checkbox"/>	358	
3			-32768	32767	<input checked="" type="checkbox"/>	0	

S7 Allgemein  Analog  Digital							
	Name				Aktiv	Istwert	±
0	Clock byte bit 0				<input checked="" type="checkbox"/>	1	^
1	Clock byte bit 1				<input checked="" type="checkbox"/>	0	
2	Clock byte bit 2				<input checked="" type="checkbox"/>	1	
3	Clock byte bit 3				<input checked="" type="checkbox"/>	0	
4	Clock byte bit 4				<input checked="" type="checkbox"/>	1	
5	Clock byte bit 5				<input checked="" type="checkbox"/>	0	
6	Clock byte bit 6				<input checked="" type="checkbox"/>	1	
7	Clock byte bit 7				<input checked="" type="checkbox"/>	1	
8					<input checked="" type="checkbox"/>	0	

#### Tipp



Sie können das automatische Ausfüllen der Spalte verwenden, siehe *ibaPDA-Handbuch*.

#### Andere Dokumentation



Für eine detaillierte Beschreibung der weiteren Möglichkeiten siehe *ibaPDA-Handbuch*.

### 3.3.6 Modultyp S7 TCP/UDP Integer

Mit dem Modul *Integer* können Sie bis zu 32 Analogwerte (Integer) und 32 Binärsignale erfassen.

Das Modul hat keine modulspezifischen Einstellungen.

### 3.3.7 Modultyp S7 TCP/UDP Real

Mit dem Modul *Real* können Sie bis zu 32 Analogwerte (Real) und 32 Binärsignale erfassen.

#### Modulspezifische Einstellungen

##### Anzahl Analogsignale

Stellen Sie die Anzahl der konfigurierbaren Analogsignale in der Signaltabelle in den Stufen 8, 16, 32 ein.

### 3.3.8 Modultyp S7 TCP/UDP Allgemein

Mit dem Modul *Allgemein* können Sie eine beliebige Datenstruktur mit maximaler Länge von 4096 Bytes erfassen (siehe hierzu auch die Fußnote in [Allgemeine Informationen](#), Seite 8).

#### Modulspezifische Einstellungen

Bei der Signalkonfiguration tragen Sie für jede Variable die Adresse, d. h. den Offset im Telegrammpuffer, sowie den Datentyp ein. Achten Sie darauf, dass hier ab Nutzdatenanfang ohne Header gezählt wird. Sie müssen daher 6 Bytes von der Offsetadresse im Datenbaustein subtrahieren, um die korrekte Adresse für die Messdaten anzugeben.

#### Hinweis



Der Modultyp *S7 TCP/UDP Allgemein* unterstützt auch die Erfassung und Verarbeitung von Texten. Hierzu können im Register *Analog* die Datentypen STRING[32] oder CHAR[32] ausgewählt werden. Zur Wandlung eines Textsignals bzw. Unterteilung in mehrere Textsignale verwenden Sie ein *Texttrenner*-Modul unter der Schnittstelle *Virtuell*.

<div>  Allgemein            Analog            Digital         </div>								
	Name	Einheit	Gain	Offset	Adresse	Datentyp	Aktiv	Istwert ±
0	Digitals 0-31		1	0	0	DWORD	<input checked="" type="checkbox"/>	113
1	Digitals 32-63		1	0	4	DWORD	<input checked="" type="checkbox"/>	0
2	Sirius Integer		1	0	8	INT	<input checked="" type="checkbox"/>	-42
3	Cosinus Integer		1	0	10	INT	<input checked="" type="checkbox"/>	38
4	Triangle Integer		1	0	12	INT	<input checked="" type="checkbox"/>	312
5			1	0	14	INT	<input checked="" type="checkbox"/>	0
6			1	0	16	INT	<input checked="" type="checkbox"/>	0
7			1	0	18	INT	<input checked="" type="checkbox"/>	0
8			1	0	20	INT	<input checked="" type="checkbox"/>	0
9	Sirius Real		1	0	72	REAL	<input checked="" type="checkbox"/>	-42,6265
10	Cosinus Real		1	0	76	REAL	<input checked="" type="checkbox"/>	38,3018
11	Triangle Real		1	0	80	REAL	<input checked="" type="checkbox"/>	312
12			1	0	84	REAL	<input checked="" type="checkbox"/>	0

### 3.3.9 Modultypen S7 UDP Request/S7 UDP Request Decoder

Diese beiden Modultypen werden nur angezeigt, wenn die Lizenz *ibaPDA-Request-S7-UDP* vorhanden ist.

#### Andere Dokumentation



Die Module und ihre Funktionen sind ausführlich im Handbuch zum Produkt *ibaPDA-Request-S7-UDP* beschrieben.

### 3.3.10 Moduldiagnose

Die Tabellen *Analog* und *Digital* der Module zeigen die Inhalte der Telegramme.

S7 Allgemein Analog Digital								
	Name	Einheit	Gain	Offset	Adresse	Datentyp	Aktiv	Istwert ±
0	Digitals 0-31		1	0	0	DWORD	<input checked="" type="checkbox"/>	113
1	Digitals 32-63		1	0	4	DWORD	<input checked="" type="checkbox"/>	0
2	Sirius Integer		1	0	8	INT	<input checked="" type="checkbox"/>	-42
3	Cosinus Integer		1	0	10	INT	<input checked="" type="checkbox"/>	38
4	Triangle Integer		1	0	12	INT	<input checked="" type="checkbox"/>	312
5			1	0	14	INT	<input checked="" type="checkbox"/>	0

Folgende Fehler können auftreten:

- Es werden keine Daten angezeigt:
  - Der Telegramm-DB auf S7-Seite ist nicht richtig gefüllt.
  - Die Anschlüsse des Sendebausteins sind falsch beschaltet.
- Es werden falsche Werte angezeigt:
  - Der Telegramm-DB auf S7-Seite ist nicht richtig gefüllt (Offset-Fehler).
  - Bytereihenfolge ist falsch eingestellt, siehe [↗ Allgemeine Moduleinstellungen](#), Seite 25.
  - Es gibt mehrere Module mit dem gleichen Modulindex.
- Die Digitalsignale sind falsch sortiert:
  - Bytereihenfolge ist falsch eingestellt, siehe [↗ Allgemeine Moduleinstellungen](#), Seite 25.
- Die Telegramme kommen nicht schneller als ca. 200 ms mit Sequenzfehler:
  - Problem mit "Delayed Acknowledge", siehe [↗ Probleme mit TCP-Performance durch Delayed Acknowledge](#), Seite 41
  - Probleme durch "Nagle-Algorithmus", siehe [↗ Unbrauchbare TCP-Daten als Folge des Nagle-Algorithmus](#), Seite 43

## 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*.

### 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*.

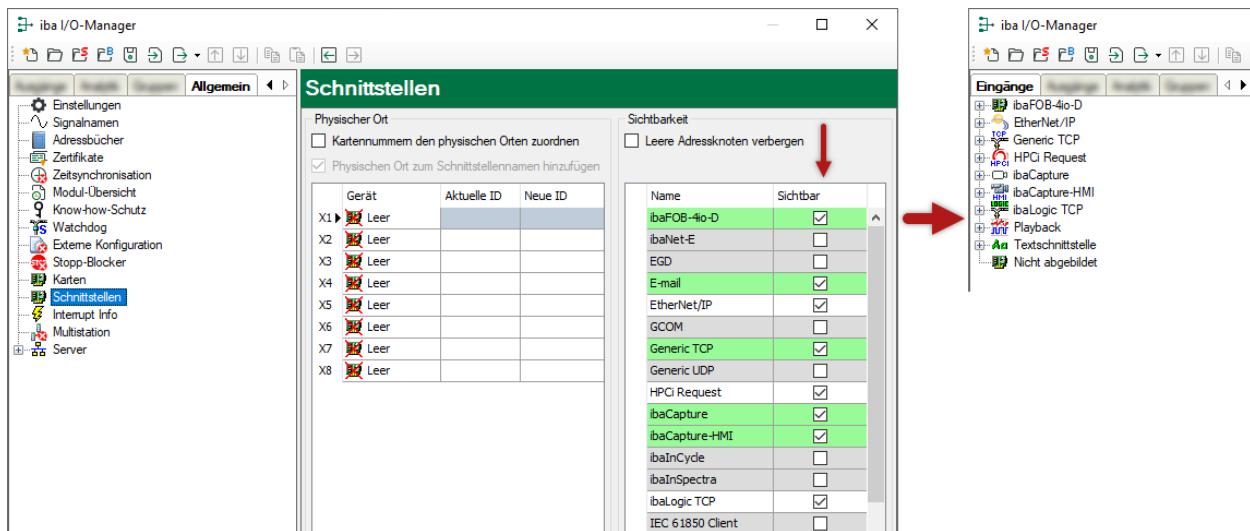
#### Sichtbarkeit

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.

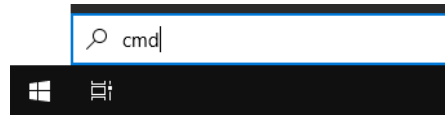
Beispiele:

- *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

Nach Übernahme der Konfiguration werden in der Verbindungsübersicht alle Verbindungen, sortiert nach Modulindex, angezeigt.

Eigenschaften

Port-Nr.:

Netzwerkschnittstellen:

TCP Port:  UDP Port:

	Adresse	Modus	Modul index	Nachrichten zähler	Unvollständige Fehler	Sequenz fehler	Paketgröße Aktuell	Zeit Aktuell
0	192.168.50.94	TCP	0	29855	0	0	74	100,6 ms
1	192.168.50.94	UDP	1	29855	0	0	74	100,9 ms
2	192.168.50.94	TCP	100	29855	0	0	138	100,5 ms
3	192.168.50.94	UDP	101	29855	0	0	138	96,9 ms
4	192.168.50.94	TCP	200	29855	0	0	206	100,7 ms
5	192.168.50.94	UDP	201	29855	0	0	206	96,1 ms
6	?	?	?	?	?	?	?	?

### Farben

Grün	Die Verbindung ist OK. Der <i>ibaPDA</i> -Modulzeitbasis ist gleich oder langsamer als der Telegrammzyklus. Der aktuelle Telegrammzyklus ist aus der Spalte "Zeit Aktuell" ersichtlich.
Orange	Die Verbindung ist OK, aber der Telegrammzyklus ist wesentlich langsamer als die <i>ibaPDA</i> -Modulzeitbasis. Es ist sinnvoll, die Modulzeitbasis an den Telegrammzyklus anzupassen.

### Verbindungsfehler

Werden die Verbindungen nicht oder nur teilweise angezeigt kann das folgende Ursachen haben:

- S7 ist in Stopp
- keine Ethernet-Verbindung zwischen *ibaPDA*-PC und dem S7 CP
- Fehler in der S7/NetPro Projektierung:
  - falsche Remote-IP-Adresse
  - Portnummer stimmt nicht mit der S7-Verbindung überein.
  - Portnummer ist durch die Firewall geblockt.
- Falscher Modulindex im Nachrichten-Header angegeben

**Weitere Fehler**

- Wenn die Nachrichtenzähler nicht kontinuierlich hoch zählen, werden auf S7-Seite die Sende-  
bausteine AG\_LSEND nicht zyklisch aufgerufen.
- Wenn Werte in der Spalten "Unvollständige Fehler" und/oder "Sequenzfehler" hoch zählen,  
deutet das auf einen der folgenden Fehler hin:
  - Die "message\_length" im DB hat nicht den erwarteten Wert.
  - Der "sequence\_counter" im DB wird nicht richtig inkrementiert.
  - Das "Delayed Acknowledge"-Problem tritt auf (siehe ➤ *Probleme mit TCP-Performance durch Delayed Acknowledge*, Seite 41).

## 4.6 Diagnosemodule

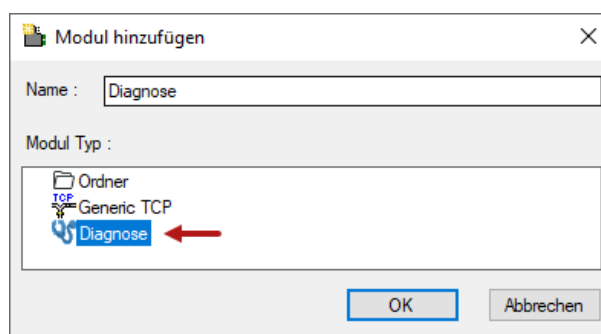
Diagnosemodule sind für die meisten Ethernet-basierten Schnittstellen und Xplorer-Schnittstellen verfügbar. Mit einem Diagnosemodul können Informationen aus den Diagnoseanzeigen (z. B. Diagnoseregister und Verbindungstabellen einer Schnittstelle) als Signale erfasst werden.

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:

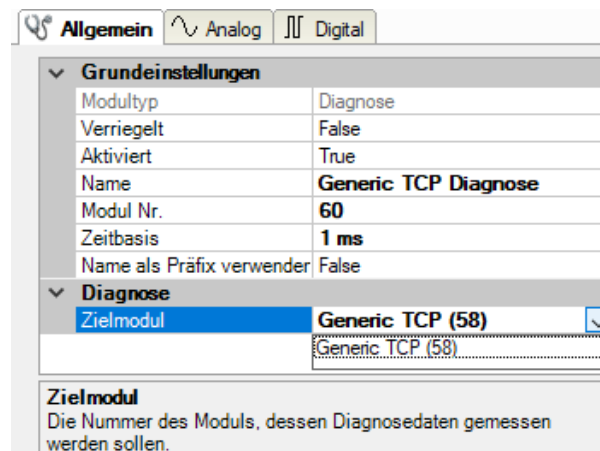
- Wenn der Fehlerzähler einer Kommunikationsverbindung einen bestimmten Wert überschreitet oder eine Verbindung abbricht, kann eine Benachrichtigung generiert werden.
- Bei einem Störfall können die aktuellen Antwortzeiten im Telegrammverkehr in einem Störungsreport dokumentiert werden.
- Der Status 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 projizierten Anzahl von Ergebniszeilen)
Zeilen (Maximum)	Höchste Anzahl der Ergebniszeilen einer SQL-Abfrage seit dem letzten Start der Erfassung (maximal gleich der projizierten 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 Fehlerbehebung

#### 5.1.1 Probleme mit TCP-Performance durch Delayed Acknowledge

##### Symptome:

*ibaPDA*-Messungen von Automatisierungsgeräten mit TCP/IP funktionieren manchmal nicht mit Zykluszeiten < 200 ms.

##### Fehlerbild in *ibaPDA*:

Unvollständige Telegramme und/oder Ausreißer in den Datenwerten (je nach Controller-Typ des Senders)

##### Ursache:

Es gibt im TCP/IP-Protokoll verschiedene Varianten, wie das "Acknowledge" behandelt wird:

Der Standard WinSocket arbeitet nach RFC1122 mit dem "Delayed Acknowledge"-Mechanismus (Delayed ACK). Dieser sagt aus, dass das "Acknowledge" verzögert wird, bis weitere Telegramme eintreffen, um diese dann gemeinsam zu quittieren. Falls keine weiteren Telegramme kommen, wird spätestens nach 200 ms (abhängig vom Socket) das ACK-Telegramm gesendet.

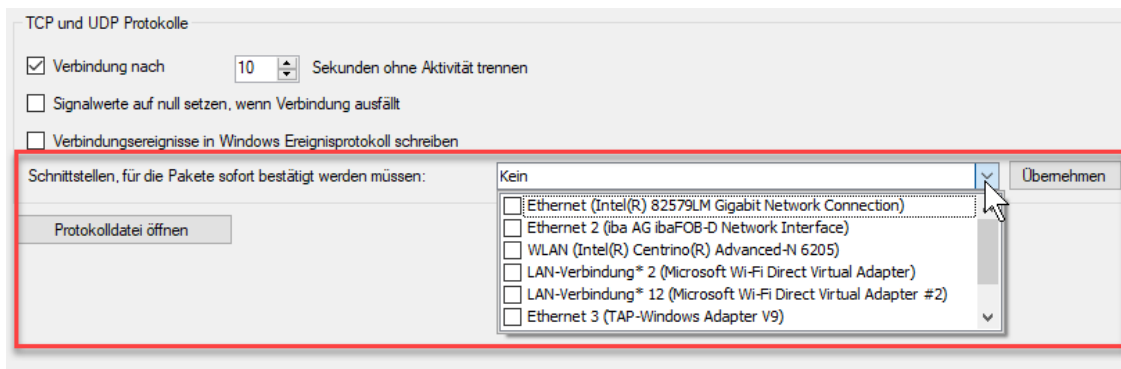
Der Datenfluss wird durch ein "Sliding Window" (Parameter Win=nnnn) gesteuert. Der Empfänger gibt an, wie viele Bytes er empfangen kann ohne eine Quittung zu senden.

Manche Controller akzeptieren dieses Verhalten nicht, sondern erwarten nach jedem Daten-telegramm eine Quittung. Falls dieses nicht innerhalb einer bestimmten Zeit (200 ms) kommt, wiederholt er das Telegramm und packt evtl. neu zu sendende Daten dazu, was beim Empfänger zu einem Fehler führt, da das vorherige Telegramm korrekt empfangen wurde.

##### Abhilfe:

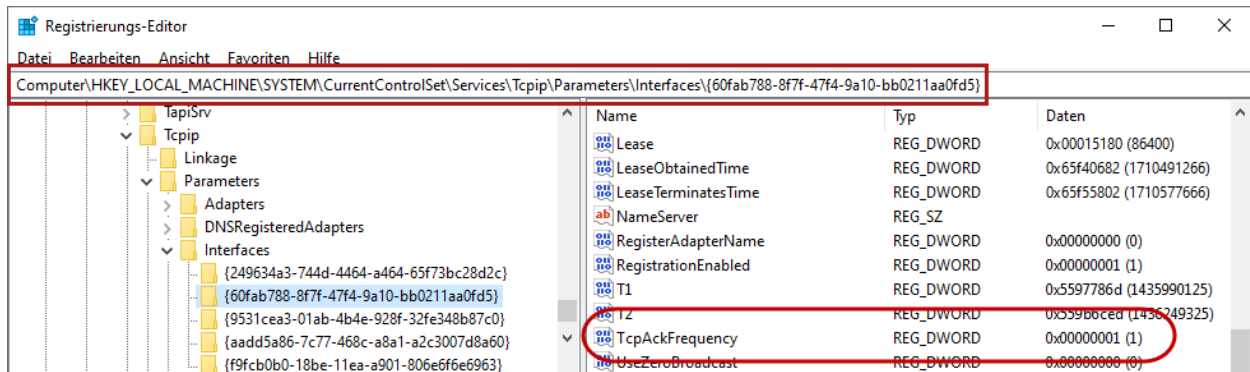
Das "Delayed Acknowledge" lässt sich einzeln pro Netzwerkadapter über einen Eintrag in der Windows Registry abschalten. Zur einfachen Änderung bietet *ibaPDA* im I/O-Manager unter *Allgemein* im Register *Einstellungen* einen entsprechenden Dialog.

Wählen Sie in der Liste der Netzwerkadapter diejenigen aus, für die das "Delayed Acknowledge" deaktiviert werden soll, und klicken Sie danach auf <Übernehmen>.



Der Parameter "TcpAckFrequency" (REG\_DWORD = 1) wird dadurch im Registry-Pfad der ausgewählten Netzwerkadapter angelegt:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{InterfaceGUID}



### Hinweis



Grundsätzlich können Sie derartige, TCP-spezifische Probleme umgehen, indem Sie *UDP* anstelle von *TCP* nutzen.

Das User Datagram Protocol (UDP) ist ein minimales, nicht verbindungsorientiertes und gegen Telegrammverluste ungesichertes Netzwerkprotokoll. Dabei wird u. a. auf Empfangsquittierung der gesendeten Daten verzichtet. In stabilen und performanten Netzwerken fällt dies jedoch nicht nennenswert ins Gewicht und kann aufgrund der bei *ibaPDA* üblichen zyklischen Datenübertragung vernachlässigt werden.

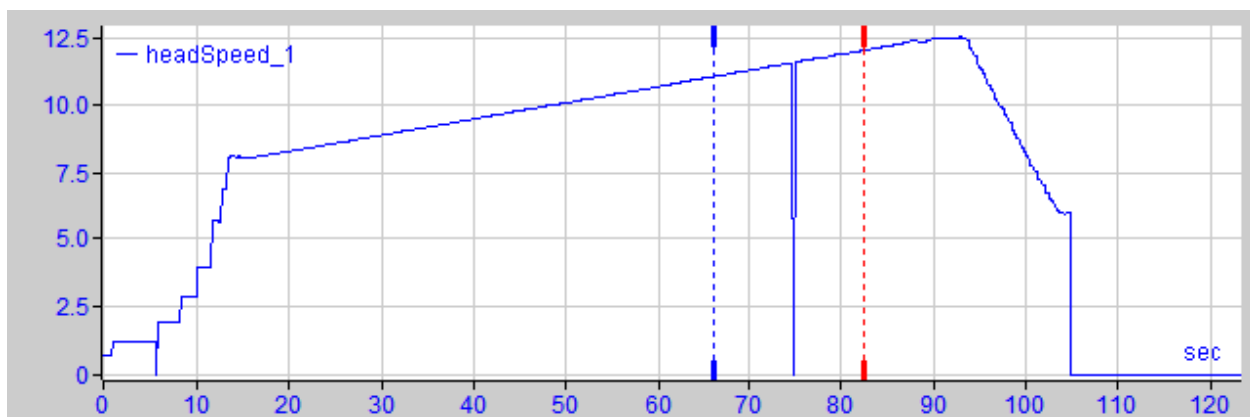
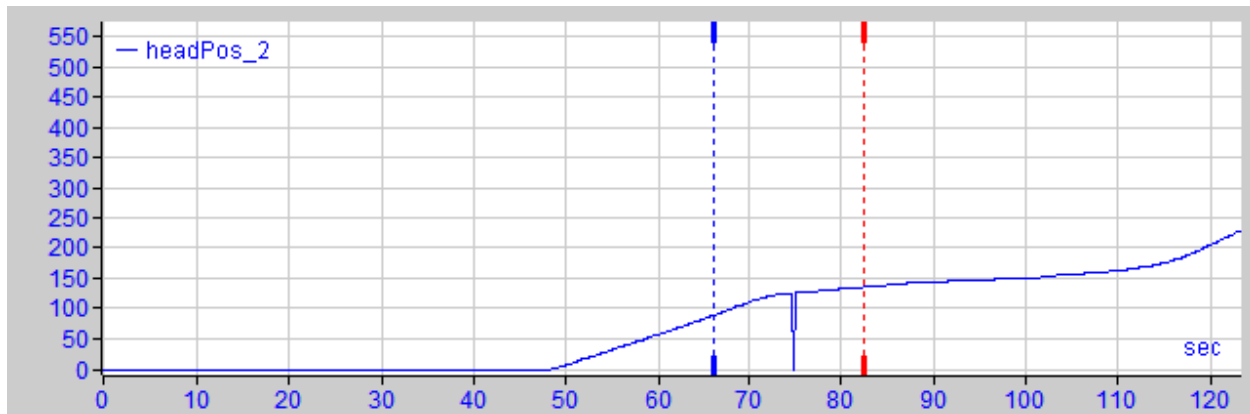
### 5.1.2 Unbrauchbare TCP-Daten als Folge des Nagle-Algorithmus

#### Symptome:

*ibaPDA*-Messungen von Automatisierungsgeräten mit TCP/IP zeigen Ausreißer in den Messwerten.

#### Fehlerbild in *ibaPDA*:

Unvollständige Telegramme und/oder Ausreißer in den Datenwerten (siehe Beispiele in den folgenden Abbildungen)



#### Ursache:

Der Nagle-Algorithmus (Nagle's algorithm) ist ein Mechanismus zur Verbesserung der TCP-Effizienz, indem er die Anzahl der über das Netz gesendeten kleinen Pakete reduziert und mehrere Datenblöcke sammelt, bevor die Daten über das Netz gesendet werden.

Da die Schnittstelle "Generic-TCP" kein Protokoll auf Anwendungsebene verwendet, kann *ibaPDA* als Empfänger diese zusammengefassten Nachrichten nicht korrekt verarbeiten. Die Schnittstelle *Generic TCP* erwartet nur ein Datagramm pro TCP-Telegramm mit stets gleichem Layout und gleicher Länge.

Aber der Nagle-Algorithmus und die Option *Delayed ACK* spielen in einem TCP/IP-Netzwerk nicht gut zusammen, siehe [Probleme mit TCP-Performance durch Delayed Acknowledge](#), Seite 41:

Der Delayed ACK-Mechanismus versucht, mehr Daten pro Segment zu senden, wenn er kann. Ein Teil des Nagle-Algorithmus hängt aber von einem ACK ab, um Daten zu senden. Delayed ACKs warten also darauf, das ACK zu senden, während der Nagle-Algorithmus darauf wartet, das ACK zu empfangen.

Dies führt zu zufälligen Verzögerungen von 200 ms bis 500 ms bei Segmenten, die sonst sofort gesendet und an den empfangsseitigen Stack von *ibaPDA* als Anwendung übergeben werden könnten.

#### Abhilfe:

Es wird empfohlen, zunächst den *Delayed ACK*-Mechanismus zu deaktivieren, wie in Kapitel 7 *Probleme mit TCP-Performance durch Delayed Acknowledge*, Seite 41 erläutert. In einer typischen Echtzeitanwendung schickt der Sender dann die neuen Daten mit einer bestimmten Zykluszeit an *ibaPDA*, da die vorherigen Daten sofort quittiert wurden. Je nach Implementierung des TCP/IP-Stacks auf der Senderseite kann der Nagle-Algorithmus dennoch aktiv werden und automatisch eine Reihe kleiner Puffernachrichten aggregieren, wodurch der Algorithmus die Übertragung absichtlich verlangsamt.

Dies kann auch sporadisch durch eine kurzzeitige Überlastung auf der Senderseite geschehen, die den Stack dazu veranlasst, einige Nachrichten zusammenzulegen.

Um den puffernden Nagle-Algorithmus zu deaktivieren, verwenden Sie die Socket-Option *TCP\_NODELAY*. Die Socket-Option *TCP\_NODELAY* ermöglicht es dem Netzwerk, die durch den Nagle-Mechanismus verursachten Delays zu umgehen, indem der Nagle-Algorithmus deaktiviert wird und die Daten gesendet werden, sobald sie verfügbar sind.

Die Aktivierung von *TCP\_NODELAY* zwingt einen Socket, die Daten in seinem Puffer zu senden, unabhängig von der Paketgröße. Das *TCP\_NODELAY*-Flag ist eine Option, die für jeden einzelnen Socket aktiviert werden kann und beim Erstellen eines TCP-Sockets angewendet wird.

(Siehe Eigenschaft *Socket.NoDelay* in .NET-Anwendungen im Namespace *System.Net.Sockets*)

---

#### Hinweis



Grundsätzlich können Sie derartige, TCP-spezifische Probleme umgehen, indem Sie *UDP* anstelle von *TCP* nutzen.

Das User Datagram Protocol (UDP) ist ein minimales, nicht verbindungsorientiertes und gegen Telegrammverluste ungesichertes Netzwerkprotokoll. Dabei wird u. a. auf Empfangsquittierung der gesendeten Daten verzichtet. In stabilen und performanten Netzwerken fällt dies jedoch nicht nennenswert ins Gewicht und kann aufgrund der bei *ibaPDA* üblichen zyklischen Datenübertragung vernachlässigt werden.

---

## 6 Support und Kontakt

### Support

Tel.: +49 911 97282-14

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

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)**