



ibaPDA-Interface-VIP-TCP/UDP

Datenschnittstelle TCP/UDP für VIP-Protokoll

Handbuch
Ausgabe 3.0

Messsysteme für Industrie und Energie
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. Zu widerhandlungen 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 zum Download bereit.

Version	Datum	Revision	Autor	Version SW
3.0	01-2025	Neue Version ibaPDA v8	nm	8.6.0

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

1	Zu dieser Dokumentation	5
1.1	Zielgruppe und Vorkenntnisse	5
1.2	Schreibweisen.....	6
1.3	Verwendete Symbole.....	7
2	Systemvoraussetzungen	8
3	Datenschnittstelle TCP/UDP für VIP	9
3.1	Allgemeine Informationen.....	9
3.1.1	Vendor Internet Protokoll (VIP)	9
3.1.2	TCP/IP und UDP	9
3.1.3	Arbeitsweise von VIP	10
3.2	Kommunikation zwischen ibaPDA und ABB (Controller)	12
3.3	Datenstrukturen	15
3.4	Projektierungshinweise	17
3.5	Konfiguration & Projektierung ibaPDA	18
3.5.1	Allgemeine Einstellungen	18
3.5.2	Allgemeine Einstellungen der Schnittstelle.....	19
3.5.3	Modul hinzufügen	20
3.5.4	Allgemeine Moduleinstellungen.....	20
3.5.4.1	Modultyp Integer.....	22
3.5.4.2	Modultyp Real	22
3.5.4.3	Modultyp Allgemein	22
3.5.5	Generelle Signalkonfiguration	23
3.5.6	Moduldiagnose.....	24
4	Diagnose	25
4.1	Lizenz	25
4.2	Sichtbarkeit der Schnittstelle.....	25
4.3	Protokolldateien	26
4.4	Verbindungsdiagnose mittels PING	27
4.5	Überprüfung der Verbindung	28
4.6	Diagnosemodule.....	30

5	Anhang	36
5.1	Fehlerbehebung.....	36
5.1.1	Probleme mit TCP-Performance durch Delayed Acknowledge	36
5.1.2	Unbrauchbare TCP-Daten als Folge des Nagle-Algorithmus	38
6	Support und Kontakt.....	40

1 Zu dieser Dokumentation

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

Andere Dokumentation



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 elektronischen 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-VIP-TCP/UDP* 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	<Tastenname> Beispiel: <Alt>; <F1>
Tastaturtasten gleichzeitig drücken	<Tastenname> + <Tastenname> Beispiel: <Alt> + <Strg>
Grafische Tasten (Buttons)	<Tastenname> Beispiel: <OK>; <Abbrechen>
Dateinamen, Pfade	Dateiname, Pfad Beispiel: Test.docx

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 TCP/UDP für VIP-Protokoll erforderlich:

- *ibaPDA* v8.0.0 oder höher
- Lizenz für *ibaPDA-Interface-VIP-TCP/UDP*
- Netzwerkanschluss 10/100 Mbit
- ABB Controller mit TCP/IP Kommunikationsinterface, z. B. CI861

Sonstige Voraussetzungen an die eingesetzte PC-Hardware und die unterstützten Betriebssysteme entnehmen Sie der *ibaPDA*-Dokumentation.

Hinweis



iba empfiehlt, 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 36 (alle *ibaPDA*-Versionen).

Lizenzen

Bestell-Nr.	Produktbezeichnung	Beschreibung
31.001065	ibaPDA-Interface-VIP-TCP/UDP	Erweiterungslizenz für ein <i>ibaPDA</i> -System um eine TCP/IP und UDP/IP Schnittstelle Anzahl der Verbindungen: 64
31.101065	one-step-up-Interface-VIP-TCP/UDP	Erweiterungslizenz für eine vorhandene <i>ibaPDA-Interface-VIP-TCP/UDP</i> -Schnittstelle um 64 weitere VIP-TCP/UDP-Verbindungen, maximal 3 Erweiterungslizenzen zulässig

3 Datenschnittstelle TCP/UDP für VIP

3.1 Allgemeine Informationen

3.1.1 Vendor Internet Protokoll (VIP)

Das Vendor Internet Protocol (VIP) dient vorrangig der Kommunikation zwischen ABB AC450RMC-Controllern sowie anderen Computern und Systemen, die nicht von ABB stammen aber dieses Protokoll verarbeiten können.

Die VIP-Funktionalität setzt auf dem Transmission Control Protocol (TCP), dem User Datagram Protocol (UDP) und dem Internet Protocol (IP) für Ethernet auf.

Im Kontext dieser Dokumentation wird das VIP-Protokoll dazu verwendet, verschiedenste Daten von einem ABB-Controller mit dem Datenerfassungssystem *ibaPDA* zu messen.

Die Auswahl der zu messenden Signale erfolgt durch Rangieren der Werte in Telegrammpuffer, deren Datenstrukturen durch die Modultypen von *ibaPDA* vorgegeben sind. Die Telegramme werden mit Standard-Sendebausteinen an den *ibaPDA*-PC gesendet.

Im *ibaPDA-Interface-VIP-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 40.961 Bytes

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

Die folgenden ABB-Controller können über das VIP-Protokoll mit *ibaPDA* kommunizieren:

- AC450 RMC
- AC800M
- AC80
- AC800 PEC

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

3.1.2 TCP/IP und UDP

TCP/IP ist ein Datentransportprotokoll, das Daten von beliebigen Applikationen oder Teilnehmern beinhalten kann. TCP/IP ist somit der Weg, mit dem Daten mittels eines Standardprotokolls auf standardisierten Schnittstellen (z. B. Ethernet-Netzwerkkarten, die von einer Vielzahl von Anbietern verfügbar sind) übertragen werden können.

Auch wenn ein TCP/IP-Treiber Daten senden und empfangen kann, muss der Nutzdateninhalt vom Anwender interpretiert werden. Nur die Nutzdaten haben für den Anwender wirklich eine Bedeutung.

TCP/IP-Paket	
Kopf (header)	Daten

Der TCP/IP-Nachrichtenkopf (header) beinhaltet im Normalfall neben Steuerinformationen auch die Quell- und Zieladresse der Nachricht. Der Datenteil (data) beinhaltet Daten mit einer speziellen Strukturierung, so dass eine *ibaPDA*-Applikation diese interpretieren kann.

Für die Verbindung zwischen ABB und *ibaPDA* ist sowohl das Verständnis der Datenstruktur wichtig als auch die Übertragungsreihenfolge und der Nachrichtenkopf. Damit ist es möglich, einen speziellen TCP/IP-Treiber zu schreiben, welcher die Datenpakete ausliest und dem *ibaPDA*-Interface zur Aufzeichnung und Analyse zur Verfügung stellt. Mit dem TCP/IP-Treiber arbeitet *ibaPDA* wie ein Verbindungsserver, die Automatisierungsgeräte arbeiten als Clients. Dies bedeutet, dass *ibaPDA* gesendete Informationen und Verbindungsanforderungen abhört.

ibaPDA kann die ABB TCP/IP-Verbindung mit einer maximalen Abtastrate von 5 ms abarbeiten. Allerdings kann es zusätzliche Restriktionen auf der Sendeseite geben. Diese werden später weiter präzisiert.

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 Transportprotokoll und arbeitet auf der Schicht 4, der Transportschicht, des OSI-Schichtenmodells. 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.

3.1.3 Arbeitsweise von VIP

Die Topologie der Kommunikation wird in nachstehendem Bild dargestellt:

Network

Node1

Link1

Kanal1

Kanal2

:

Kanal n

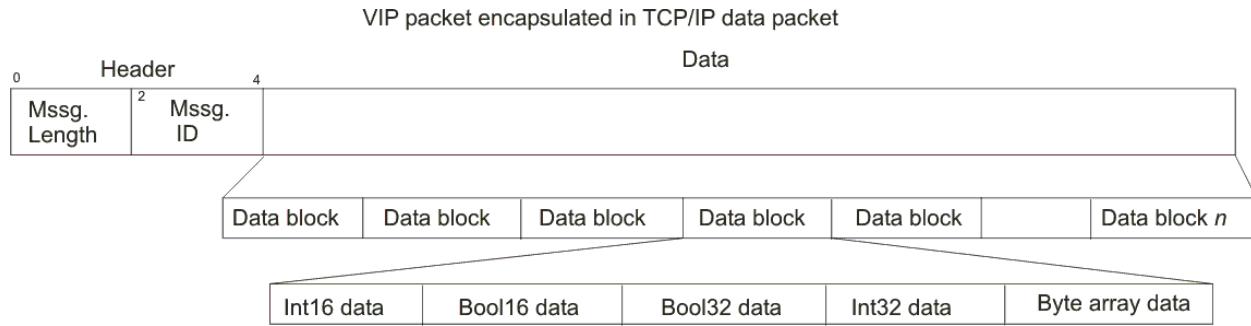
Link2

Node2

Link1

Jede Station muss am Ethernet mit einer eindeutigen Knotennummer (Node 1..99) und einer eindeutigen IP-Adresse versehen werden. Jeder Node kann zu maximal fünf verschiedenen Datenquellen Verbindungen herstellen. Jede Station (Controller) kann als Client oder Server konfiguriert werden (Aufrufe VIP-NETW, VIP-NODE, VIP-LINK, siehe auch VIP-Benutzerhandbuch, Seiten 43, 45 und 39).

Die innerhalb eines TCP/IP- bzw. UDP-Paketes eingeschlossene VIP-Paketstruktur wird in der nachstehenden Abbildung TCP/IP-VIP-Telegrammaufbau erläutert:



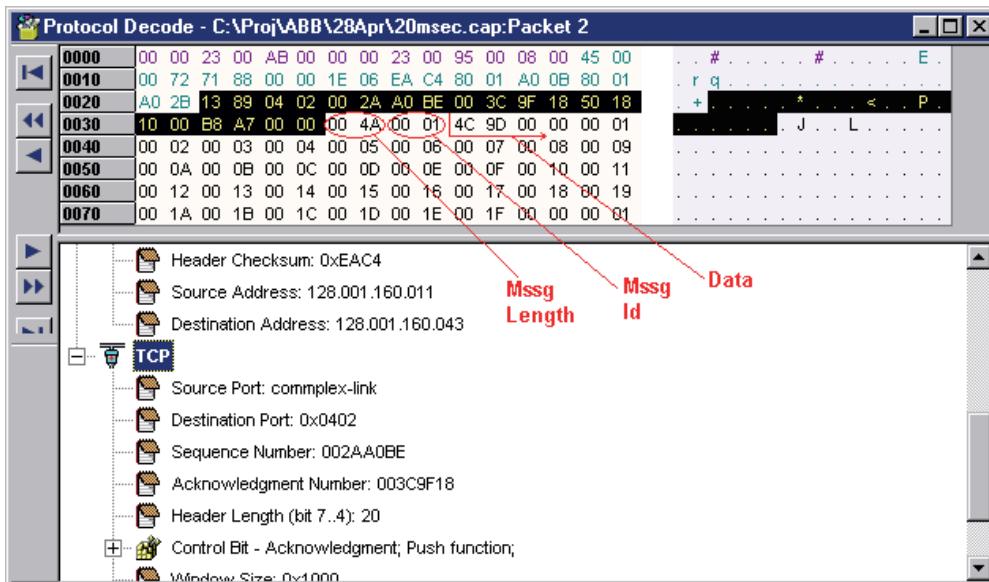
Beispiel

Angenommen, der Benutzer sendet nur zwei INT16, ein Bool32 und einen INT32 Wert mit der Message ID 2, dann sieht die Datenstruktur wie folgt aus:

| 16 | 2 | Int16 Val1 | Int16 Val2 | Bool32 Val | Int32 Val |

Die Länge in Bytes 16 beträgt demnach: 2 für Message Length, 2 für Message ID, 2x2 für die beiden INT16 Werte, und je 4 für Bool32 und INT32.

Überprüfung des Datenpakets: Es ist zu beachten, dass die Notation in Hex-Format vorliegt und die einzelnen Bytes durch zwei Zeichen repräsentiert werden, wobei die Bytes jeweils durch einen schmalen Leerbereich getrennt sind.



Das komplette TCP/IP-Paket ist in unterschiedlichen Farben, je nach Zugehörigkeit dargestellt. Am Ende des in Abbildung TCP/IP-VIP Inhalt des Datenpakets schwarz markierten Bereiches beginnt der Datenbereich des VIP-Paketes. Die ersten beiden Bytes 00 4A stellen die Nachrichtenlänge (Message Length) in Bytes dar, sie beträgt demzufolge 74 Byte. Die beiden nächsten Bytes 00 und 01 repräsentieren die Message-ID, also hier 1. Danach folgt der 2-Byte INT16-Wert 4C 9D (19613 dezimal). Anschließend folgen 32 INT16-Werte mit den Inhalten 00 00 bis 00 1F (0 bis 31 dezimal). Am Ende stehen 4 Bytes mit dem INT32-Wert 1. Zusammen demnach 74 Byte Gesamtlänge.

3.2 Kommunikation zwischen ibaPDA und ABB (Controller)

iba AG hat einen TCP/IP-Treiber entwickelt, um Daten aus dem ABB VIP-Paket zu entnehmen und diese mit *ibaPDA* anzuzeigen.

Unterstützte Verbindungen in *ibaPDA*:

- Mit *ibaPDA* ab Version 6.14.0 sind für die Schnittstelle TCP/IP-VIP insgesamt bis zu 64 Module der Typen Integer, Real und/oder "Generisch" konfigurierbar. Jedes Integer- oder Real-Modul kann bis zu 32 Analog- und Digitalsignale enthalten. Generische Module können bis zu 1000 Analog- und Digitalsignale enthalten.

- Ab *ibaPDA*-Version 6.31.0 werden wie bisher bis zu 64 Module (Verbindungen) pro Interface unterstützt. Jedoch können insgesamt 4 Lizenzen in *ibaPDA* verwendet werden, damit sind bis zu 256 Verbindungen möglich.
Die maximale Länge der Nachricht ist auf 4096 Bytes begrenzt. Generische Module können wie bisher bis zu 1000 Analog- und 1000 Digitalsignale enthalten.
- Ab *ibaPDA*-Version 6.33.2 wird auch das UDP-Protokoll unterstützt. Der Sender kann die Daten über TCP/IP oder UDP senden. Die Kommunikationsparameter und die Datenstrukturen sind identisch.

ibaPDA arbeitet als Server, der auf Clients hört, welche einen gültigen Connect-Request und anschließend Daten schicken (siehe auch Abbildung in  Sequenzähler, Seite 14). Jede TCP/IP- bzw. UDP-Verbindung kann im Diagnosefenster von *ibaPDA* bzw. im I/O-Manager eingesehen werden. *ibaPDA* ist so voreingestellt, dass der Port 5001 auf ABB VIP-Informationen abgehört wird. Deshalb muss dieser immer als Zielport eingetragen werden. Beachten Sie, dass die IP-Adresse nur einmalig vorkommen darf, aber mehr als eine Verbindung pro IP-Adresse möglich ist. Jede Verbindung entspricht einem Modul in *ibaPDA*, weshalb eine eindeutige Message ID (entspricht dem *ibaPDA* Modulindex) benötigt wird.

Die Message ID muss im gesamten System einmalig sein, auch wenn mehrere ABB-Controller genutzt werden. Das bedeutet, dass der ABB VIP-Controller bis zu 5 solcher Links aufbauen und damit jeder Controller bis zu 5 *ibaPDA*-Module belegen kann. Jedes Modul benötigt hierzu eine einmalige Quellportnummer, welche automatisch vom System vergeben wird, wenn der Benutzer für jedes Modul einen neuen Link konfiguriert.

Modulindex

Der Modulindex ist die 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.

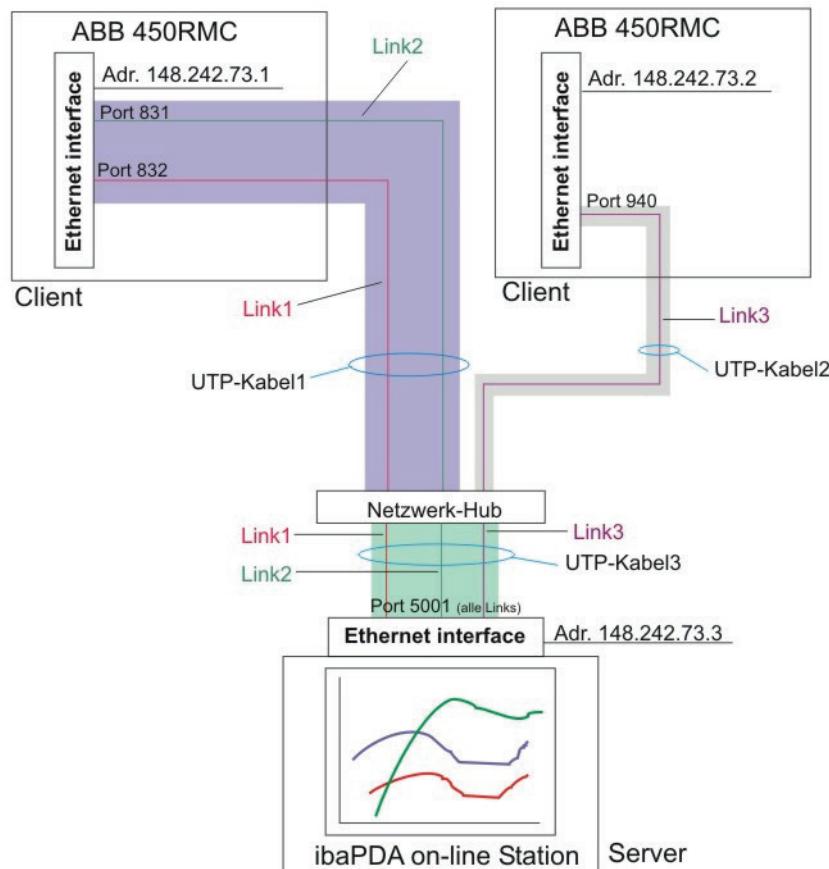
Modultyp	1. Lizenz	2. Lizenz	3. Lizenz	4. Lizenz
Integer	0-63	1000-1063	2000-2063	3000-3063
Real	100-163	1100-1163	2100-2163	3100-3163
Allgemein	200-263	1200-1263	2200-2263	3200-3263

Der Modulindex entspricht dem Index in der *ibaPDA*-Moduleinstellung. Der Wert darf während der Datenübertragung nicht verändert werden.

Um *ibaPDA* darüber zu informieren, welches dieser Module adressiert werden soll, muss in der Message ID der Modulindex hinterlegt werden. Diese ID muss auf der ABB VIP-Seite eingetragen werden.

Sequenzzähler

ibaPDA ist darüber hinaus in der Lage, Kommunikationsfehler zu erkennen. Für diese Funktion hat *iba* festgelegt, dass die ersten beiden Datenbytes einen Aufwärtszähler bilden sollen, welcher bei jedem Sendezyklus um 1 inkrementiert werden muss. Dieser Zähler muss auf der ABB Sendeseite der erste INT16 Wert sein. Bei Überlauf muss der Zähler von 32767 auf -32768 springen (0x7FFF à 0x8000) bzw. von 65535 auf 0 (0xFFFF à 0x0000).



Beachten Sie, dass in der Abbildung zwei ABB AC450-Controller als Clients agieren, wobei jeder über sein eigenes Ethernet-Modul verfügt, welches jeweils über ein Kabel mit dem Netzwerk-Hub verbunden ist. Es existiert ein weiteres Kabel, welches *ibaPDA* mit dem Hub verbindet (die Kabel sind hier mit unterschiedlicher Dicke und Färbung dargestellt). Damit besteht das Netzwerk aus drei Kommunikationspartnern mit eindeutigen IP-Adressen.

Anmerkung: Ob die Verbindungen als UTP, ThinNetCoax oder optisch auszuführen sind, ist für die jeweilige Applikation festzulegen. Da drei Teilnehmer vorhanden sind, bedeutet dies mindestens drei verwendete Module innerhalb von *ibaPDA* (hier angedeutet mit drei Kurven in verschiedenen Farben). Beachten Sie, dass jeder Teilnehmer über eine zufällige Quellportnummer verfügt (940, 831 und 832), während sie alle dieselbe Zielportnummer (Target Port No.) 5001 haben.

Hinweis

Wichtig für die Verwendung eines Cross-over-Kabels zwischen dem ABB-Controller und *ibaPDA*:

Da die meisten ABB-Controller mit 10MBit/s im Halbduplex-Modus kommunizieren, wird empfohlen, die Ethernet-Schnittstelle des *ibaPDA*-PCs auch auf 10 MBit/s, halbduplex, einzustellen, wenn Sie ein Cross-over-Kabel verwenden.

Wenn Sie die Übertragungsgeschwindigkeit nicht auf halbduplex setzen können, dann verwenden Sie einen Netzwerk-Hub oder -Switch zwischen dem Controller und *ibaPDA*.

3.3 Datenstrukturen

Da *ibaPDA* die exakte Lage und den Typ eines jeden Wertes kennen muss, hat iba folgende Telegrammstrukturen für die verschiedenen Datenformate festgelegt.

Pro Nachricht kann nur eine Telegrammstruktur verwendet werden.

VIP_32_integer: 32 Integer Werte + 32 Binärwerte

rel. #	Bytes	C - Type	Beschreibung	Bemerkung
00	2	short int	Message Length	Festgelegt auf 74 Byte
02	2	short int	Message ID	Modulindex in <i>ibaPDA</i> , siehe Kommunikation zwischen <i>ibaPDA</i> und ABB (Controller) , Seite 12
04	2	unsigned short	1. INT16-Signal	Sequenzähler: Bei jedem Sendzyklus inkrementieren! Bei Überlauf 65535 -> 0
06	64	short int	32 Analogwerte	
70	4	long int	32 Digitalwerte	

VIP_8_real: 8 Real Werte + 32 Binärwerte

rel. #	Bytes	C - Type	Beschreibung	Bemerkung
00	2	short int	Message Length	Festgelegt auf 42 Byte
02	2	short int	Message ID	Modulindex in <i>ibaPDA</i> , siehe Kommunikation zwischen <i>ibaPDA</i> und ABB (Controller) , Seite 12
04	2	unsigned short	1. INT16-Signal	Sequenzähler: Bei jedem Sendzyklus inkrementieren! Bei Überlauf 65535 -> 0
06	4	long int	32 Digitalwerte	
10	32	float	8 Analogwerte	IEEE Format

VIP_16_real: 16 Real Werte + 32 Binärwerte

rel. #	Bytes	C - Type	Beschreibung	Bemerkung
00	2	short int	Message Length	Festgelegt auf 74 Byte
02	2	short int	Message ID	Modulindex in <i>ibaPDA</i> , siehe Kommunikation zwischen ibaPDA und ABB (Controller) , Seite 12
04	2	unsigned short	1. INT16-Signal	Sequenzzähler: Bei jedem Sendezyklus inkrementieren! Bei Überlauf 65535 -> 0
06	4	long int	32 Digitalwerte	
10	64	float	16 Analogwerte	IEEE Format

VIP_32_real: 32 Real Werte + 32 Binärwerte

rel. #	Bytes	C - Type	Beschreibung	Bemerkung
00	2	short int	Message Length	Festgelegt auf 138 Byte
02	2	short int	Message ID	Modulindex in <i>ibaPDA</i> , siehe Kommunikation zwischen ibaPDA und ABB (Controller) , Seite 12
04	2	unsigned short	1. INT16-Signal	Sequenzzähler: Bei jedem Sendezyklus inkrementieren! Bei Überlauf 65535 -> 0
06	4	long int	32 Digitalwerte	
10	128	float	32 Analogwerte	IEEE Format

VIP_Allgemein: max. 4096 Bytes

rel. #	Bytes	C - Type	Beschreibung	Bemerkung
00	2	short int	Message Length	Maximum 4102 Bytes
02	2	short int	Message ID	Modulindex in <i>ibaPDA</i> , siehe Kommunikation zwischen ibaPDA und ABB (Controller) , Seite 12
04	2	unsigned short	1. INT16-Signal	Sequenzzähler: Bei jedem Sendezyklus inkrementieren! Bei Überlauf 65535 -> 0
06	max. 4096	byte, int, word, double int, dou- ble word, float, string auch gemischt	Analog- / Digi- tal-Werte	Analogwerte unterstützen Textsig- nale

3.4 Projektierungshinweise

Mit ABB-Controllern, die das TCP/IP- oder das UDP-VIP-Protokoll unterstützen, können bis zu 64 Verbindungen genutzt werden, um Daten an *ibaPDA* zu senden, wobei in *ibaPDA* bis zu 64 Module konfiguriert werden können.

Ab *ibaPDA*-Version 6.31.0 werden bis zu 4 TCP/IP-VIP-Interfaces unterstützt, damit sind bis zu 256 Verbindungen möglich.

Es muss sichergestellt sein, dass jede Message ID (Modulindex) einmalig ist und im TCP/IP- bzw. UDP-Telegramm zu *ibaPDA* enthalten ist und damit jeweils genau ein Modul in *ibaPDA* adressiert.

ibaPDA reagiert in Bezug auf ABB VIP auf den Port 5001.

Hinweis



Ab *ibaPDA*-Version 6.33.2 kann in *ibaPDA* auch ein anderer Port eingestellt werden.

ABB Controller dürfen also nur diesen Port benutzen. Bei der Nutzung von Technostrings muss daher auf eine andere verfügbare Portnummer umgeschaltet werden.

Bei jedem Link muss das erste INT16-Signal als Aufwärtszähler projektiert sein, welcher mit jedem Sendezyklus inkrementiert werden muss.

Die Länge der Nachricht (Telegrammlänge / Message Length) hängt vom gewählten Telegramm ab (32 INT, 8-, 16- oder 32 REAL oder Allgemein). Die ABB Controller müssen sendeseitig die Layoutanforderungen erfüllen, da sonst u. U. die Meldung "Unvollständige Datenpakete" im *ibaPDA*-Diagnosefenster (I/O-Manager) erscheint.

Es sollten nur *ibaPDA* und der/die ABB-Controller am Bus angeschlossen sein.

Um die Zeitbasis (Grundtakt) für *ibaPDA* zu generieren, ist in jedem Fall die Installation einer *ibaFOB*-Karte als Interrupt-Quelle zu empfehlen.

Hinweis



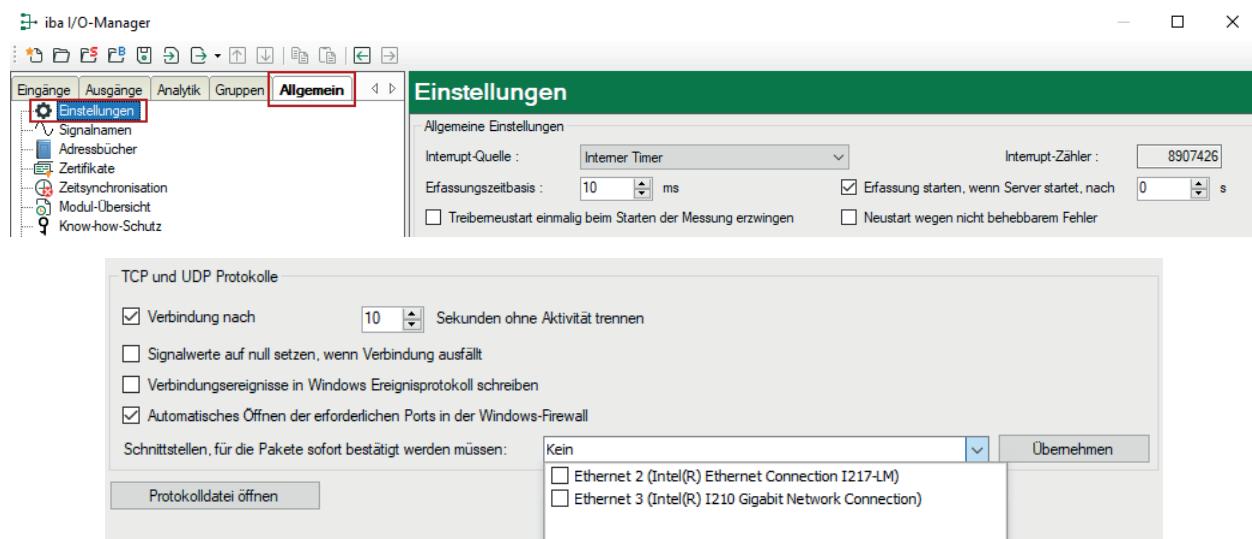
Innerhalb des VIP-Netzwerkes sollten neben *ibaPDA* und den beteiligten Controllern keine weiteren Teilnehmer angeschlossen sein, um eine hohe Dynamik des Systems sicherzustellen.

3.5 Konfiguration & Projektierung ibaPDA

Nachfolgend ist die Projektierung in *ibaPDA* beschrieben. Wenn alle Systemvoraussetzungen erfüllt sind, wird im Signalbaum die Schnittstelle *VIP TCP/UDP* angezeigt.

3.5.1 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

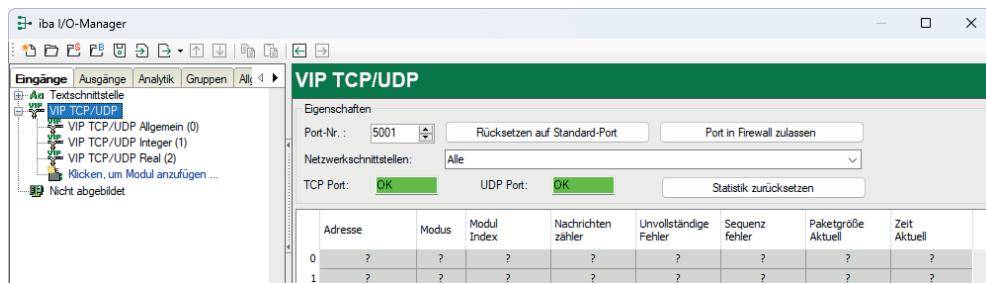
Hinweis



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.

3.5.2 Allgemeine Einstellungen der Schnittstelle

Die Schnittstelle selbst hat folgende Funktionen und Konfigurationsmöglichkeiten:



Port-Nr.

Verwendeter Port PC-seitig. Die Portnummer muss in der VIP-Verbindungsprojektierung identisch verwendet werden. Ab *ibaPDA*-Version 6.33.2 kann die Portnummer hier frei gewählt werden.

<Rücksetzen auf Standard-Port>

Die Portnummer 5001 wird eingestellt.

<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.

Netzwerkschnittstellen

Mit dieser Drop-down-Liste können Sie bestimmen, welcher Netzwerkadapter des betreffenden Rechners für diese Schnittstelle verwendet wird. Nur auf den ausgewählten Netzwerkadapters werden die Ports für die Kommunikation geöffnet. Falls auf einem Netzwerkadapter mehrere IP-Adressen konfiguriert sind, wird für all diese IP-Adressen ein Socket geöffnet. Damit die Schnittstellenkonfiguration validiert werden kann, muss mindestens ein Netzadapter ausgewählt sein. Wenn Sie die Auswahl *kein* treffen, wird bei der Validierung der I/O-Konfiguration eine Fehlermeldung angezeigt. Standardmäßig sind alle Netzwerkadapter ausgewählt.

TCP Port / UDP Port

Hier erfolgt die Anzeige OK wenn der Socket auf diesem Port geöffnet werden kann. Die Anzeige FEHLER kommt, wenn Konflikte auftreten, z. B. wenn der Port schon anderweitig belegt ist.

Verbindungstabelle

siehe **➤ Überprüfung der Verbindung**, Seite 28

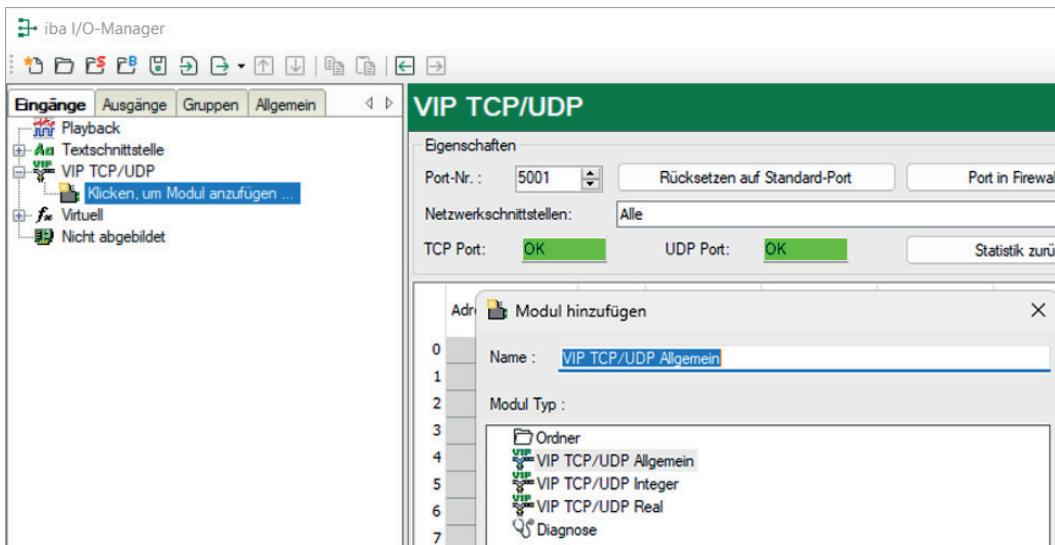
<Statistik zurücksetzen>

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

3.5.3 Modul hinzufügen

Vorgehen

1. Klicken Sie auf den blauen Link *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>.



Hinweis



Wenn bereits eine TCP/IP- oder UDP-Verbindung 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

Folgende Modultypen können Sie zur Schnittstelle hinzufügen:

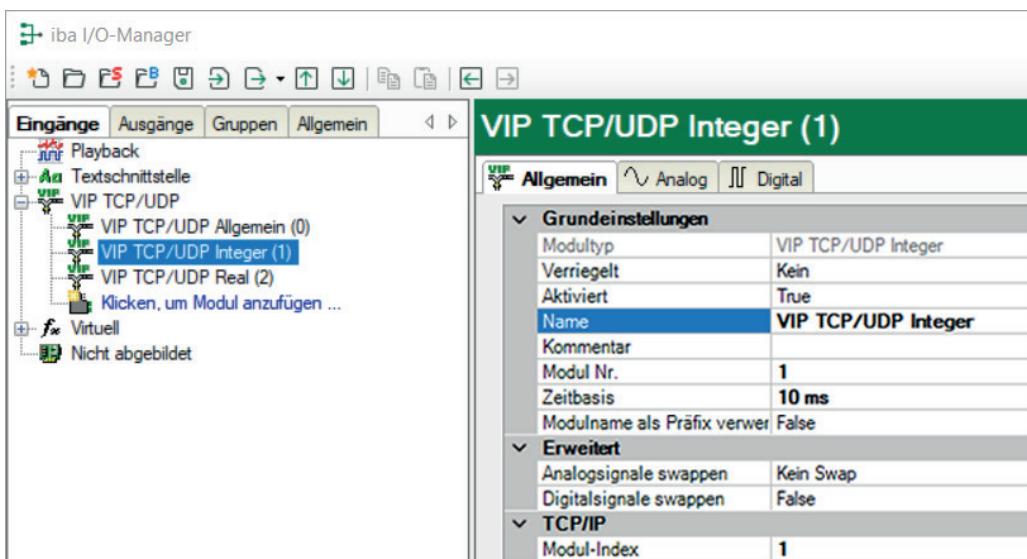
- VIP TCP/UDP Allgemein
- VIP TCP/UDP Integer
- VIP TCP/UDP Real

siehe *Allgemeine Moduleinstellungen*, Seite 20

3.5.4 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.

Kommentar

Hier können Sie einen Kommentar oder eine Beschreibung zum Modul eintragen. Dies wird dann als Tooltip im Signalbaum angezeigt.

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 Signallnamen voran.

Erweitert

Analogsignale swappen/Digitalsignale swappen

Möglichkeit die Auswertereihenfolge der Byte zu ändern. Der zu wählende Swap-Modus hängt vom Swap-Modus der Signalquelle ab.

TCP/IP

Modulindex

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

Siehe dazu [Kommunikation zwischen ibaPDA und ABB \(Controller\)](#), Seite 12.

3.5.4.1 Modultyp Integer

Mittels des Moduls Integer können bis zu 32 Analogwerte (Integer) und 32 Binärsignale erfasst werden.

Das Modul hat keine modulspezifischen Einstellungen.

3.5.4.2 Modultyp Real

Mittels des Moduls Real können bis zu 32 Analogwerte (Real) und 32 Binärsignale erfasst werden.

Folgende Moduleinstellungen sind modulspezifisch:

Anzahl Analogsignale

Die zu erfassende Anzahl der Analogsignale ist in den Stufen 8, 16, 32 konfigurierbar (Die Anzahl der Digitalsignale liegt fest bei 32).

3.5.4.3 Modultyp Allgemein

Mit dem Modul *Allgemein* können Sie eine beliebige Datenstruktur mit maximaler Länge von 4096 Bytes erfassen.

Folgende Moduleinstellungen sind modulspezifisch:

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.

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.

Hinweis

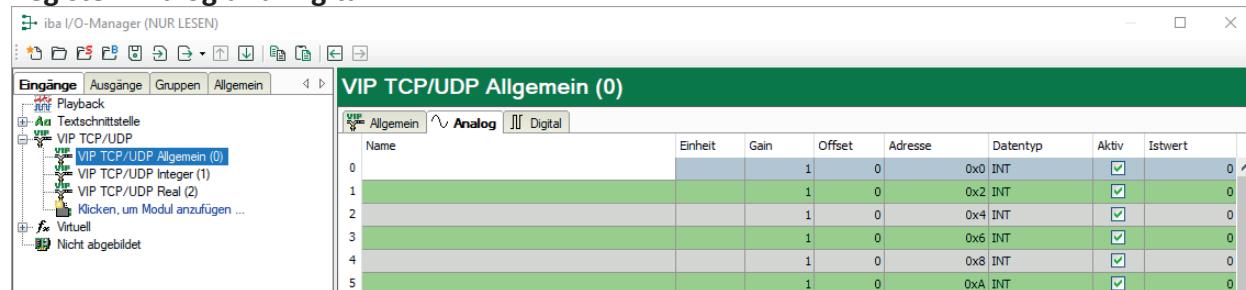


Der Modultyp *VIP TCP/UDP Allgemein* unterstützt auch die Erfassung und Verarbeitung von Texten. Hierzu kann im Register *Analog* der Datentyp STRING[32] ausgewählt werden. Zur Wandlung eines Textsignals bzw. Unterteilung in mehrere Textsignale verwenden Sie ein *Texttrenner*-Modul unter der Schnittstelle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

3.5.5 Generelle Signalkonfiguration

Die Auswahl der zu messenden Daten erfolgt auf ABB-Seite durch Rangieren der Signale in die Datenbereiche die zyklisch an *ibaPDA* gesendet werden.

Register Analog und Digital



Name

Eingabe eines aussagekräftigen Klartextnamens als Signalbezeichnung

Einheit (nur Analogsignale)

Eingabe einer physikalischen Einheit für das Signal

Sie können maximal 11 Zeichen eingeben, das Feld wird nur als Kommentarfeld betrachtet. Die Einheit erscheint immer in Verbindung mit einer numerischen Anzeige der Werte.

Gain, Offset (nur Analogsignale)

Angabe von Gain (Verstärkung) und Offset (Signalwert im Nullpunkt) zur Skalierung der einge-henden Werte

Diese Werte beschreiben eine lineare Kennlinie für die Skalierung zur Umrechnung in physika-lische Einheiten. Wenn eingehende Werte in physikalischen Einheiten angegeben werden, kön-nen Sie diese Funktion ignorieren, also Gain = 1 und Offset = 0 setzen.

Adresse

Der Adressraum hängt vom Datentyp ab. Nach dem Ändern von Datentypen kann so eine An-passung der Adresseinträge erforderlich sein.

Die Adressierung der Digitalsignale erfolgt über die Spalten *Adresse* und *Bit-Nr. 0 – 31*

Datentyp (nur Analogsignale)

In den Feldern dieser Spalte können Sie den Datentyp für jedes Signal auswählen. Klicken Sie einfach in das entsprechende Feld und wählen den Datentyp in der Liste aus.

Die folgenden Datentypen stehen zur Verfügung: INT, SINT, BYTE, WORD, DINT, DWORD, FLOAT, DOUBLE, STRING[32]

Aktiv

Aktivierung oder Deaktivierung des jeweiligen Signals

Istwert

Anzeige des aktuellen Istwerts des Signals

Andere Dokumentation



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

Tipp



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

3.5.6 Moduldiagnose

In den Tabellen *Analog* und *Digital* der VIP-TCP/UP-Module werden die Inhalte der Telegramme angezeigt.

VIP	Allgemein	Analog	Digital	Name	Einheit	Gain	Offset	Adresse	Datentyp	Aktiv	Istwert
0	TCP Generic Digitals 0-31					1	0	0x0	DWORD	<input checked="" type="checkbox"/>	1,12886e...
1	TCP Generic Digitals 32-63					1	0	0x4	DWORD	<input checked="" type="checkbox"/>	1,16029e...
2	TCP Generic Int 0					1	0	0x8	INT	<input checked="" type="checkbox"/>	-20421
3	TCP Generic Int 1					1	0	0xA	INT	<input checked="" type="checkbox"/>	-20421
4	TCP Generic Int 2					1	0	0xC	INT	<input checked="" type="checkbox"/>	-20421
5	TCP Generic Int 3					1	0	0xE	INT	<input checked="" type="checkbox"/>	-20421
6	TCP Generic Int 4					1	0	0x10	INT	<input checked="" type="checkbox"/>	-20421
7	TCP Generic Int 5					1	0	0x12	INT	<input type="checkbox"/>	-20421
8	TCP Generic Int 6					1	0	0x14	INT	<input type="checkbox"/>	-20421

Folgende Fehler können auftreten:

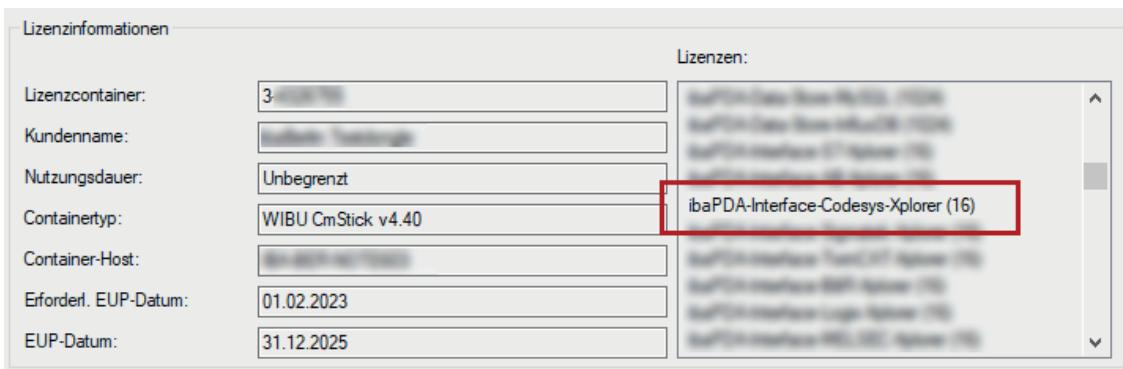
- Es werden keine Daten angezeigt:
 - Der Telegrammpuffer auf der Senderseite ist nicht richtig gefüllt
 - Die Anschlüsse des Sendbausteins sind falsch beschaltet
- Es werden falsche Werte angezeigt:
 - Der Telegrammpuffer auf der Senderseite ist nicht richtig gefüllt (Offset-Fehler)
 - Bytereihenfolge ist falsch eingestellt (siehe **Allgemeine Moduleinstellungen**, Seite 20)
 - Es gibt mehrere Module mit dem gleichen Modulindex
- Die Digitalsignale sind falsch sortiert
 - Bytereihenfolge ist falsch eingestellt (siehe **Allgemeine Moduleinstellungen**, Seite 20).
- Die Telegramme kommen nicht schneller als ca. 200 ms mit Sequenzfehler
 - Problem mit "Delayed Acknowledge", siehe **Probleme mit TCP-Performance durch Delayed Acknowledge**, Seite 36
 - Probleme durch "Nagle-Algorithmus", siehe **Unbrauchbare TCP-Daten als Folge des Nagle-Algorithmus**, Seite 38

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 die Schnittstelle *ibaPDA-Interface-VIP-TCP/UDP* 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*.

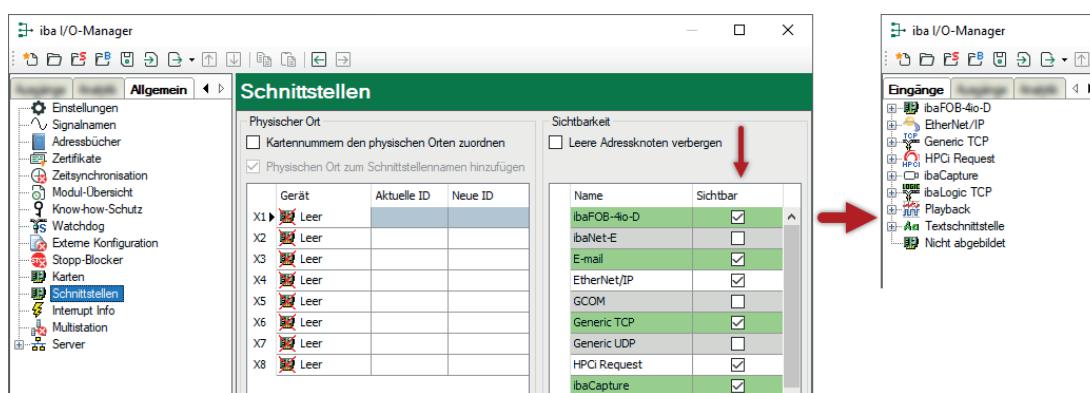
Sichtbarkeit

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 verbindungsspezifischen 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_ss_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.

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

C:\Windows\system32>ping 192.168.81.10

Ping wird ausgeführt für 192.168.81.10 mit 32 Bytes Daten:
Antwort von 192.168.81.10: Bytes=32 Zeit=1ms TTL=30
Antwort von 192.168.81.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.81.10: Bytes=32 Zeit<1ms TTL=30
Antwort von 192.168.81.10: Bytes=32 Zeit<1ms TTL=30

Ping-Statistik für 192.168.81.10:
    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.

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

C:\Windows\system32>ping 192.168.81.10

Ping wird ausgeführt für 192.168.81.10 mit 32 Bytes Daten:
Antwort von 192.168.81.10: Zielhost nicht erreichbar.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.81.10:
    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.

VIP TCP/UDP								
Eigenschaften								
Port-Nr. :	5001	Rücksetzen auf Standard-Port	Port in Firewall zulassen					
Netzwerkschnittstellen:	Alle							
TCP Port:	OK	UDP Port:	OK	Statistik zurücksetzen				
Adresse	Modus	Modul index	Nachrichtenzähler	Unvollständige Fehler	Sequenzfehler	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 ?	?	?	?	?	?	?	?	
7 ?	?	?	?	?	?	?	?	

Die Hintergrundfarbe der Zeilen hat folgende Bedeutung:

Farbe	Bedeutung
Grün	Die Verbindung ist OK. Der <i>ibaPDA</i> -Modulzeitbasis ist gleich oder langsamer als der Telegrammzyklus. Der aktuelle Telegrammzyklus ist aus der Spalte <i>Zeit Aktuell</i> 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.

Wenn die Verbindungen nicht oder nur teilweise angezeigt werden, kann das folgende Ursachen haben:

- Sender ist in Stopp
- keine Ethernet-Verbindung zwischen *ibaPDA*-PC und der ABB-Steuerung
- Fehler in der Verbindungsprojektierung:
 - falsche Remote-IP-Adresse
 - *ibaPDA*-Portnummer stimmt nicht mit der Verbindungsprojektierung überein
 - Portnummer ist durch die Firewall geblockt
- Falscher Modulindex im Telegramm-Header angegeben

Weitere Fehler

- Wenn die Nachrichtenzähler nicht kontinuierlich hochzählen, werden auf Senderseite die Telegramme nicht zyklisch gesendet.
- Wenn Werte in der Spalte "Unvollständig Fehler" und/oder "Sequenzfehler" hochzählen, deutet das auf einen der folgenden Fehler hin:
 - Die "message_length" im Telegrammheader hat nicht den erwarteten Wert.
 - Der "sequence_counter" im Telegrammheader wird nicht richtig inkrementiert.
 - Das "Delayed Acknowledge"-Problem tritt auf, siehe [**→ Probleme mit TCP-Performance durch Delayed Acknowledge**](#), Seite 36

4.6 Diagnosemodule

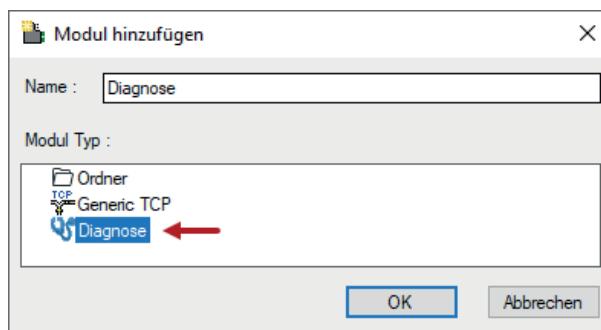
Diagnosemodule sind für die meisten Ethernet-basierten Schnittstellen und Xplorer-Schnittstellen verfügbar. Mit einem Diagnosemodul können Informationen aus den Diagnoseanzeigen (z. B. Diagnoseregister und Verbindungstabellen einer Schnittstelle) als Signale erfasst werden.

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:

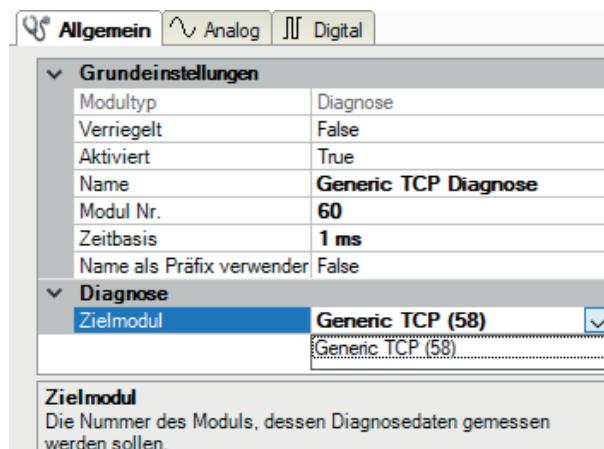
- Wenn der Fehlerzähler einer Kommunikationsverbindung einen bestimmten Wert überschreitet oder eine Verbindung abbricht, kann eine Benachrichtigung generiert werden.
- Bei einem Störungsfall können die aktuellen Antwortzeiten im Telegrammverkehr in einem Störungsreport dokumentiert werden.
- Der Status 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 Auswahlliste der Einstellung stehen die unterstützten Module derselben Schnittstelle zur Auswahl. Pro Diagnosemodul kann genau ein Datenfassungsmodul 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.

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. Oktette 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.

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 Ergebnissezeilen der letzten SQL-Abfrage (innerhalb der projektierten Anzahl von Ergebnissezeilen)
Zeilen (Maximum)	Höchste Anzahl der Ergebnissezeilen einer SQL-Abfrage seit dem letzten Start der Erfassung (maximal gleich der projektierten Anzahl von Ergebnissezeilen)
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

Im Folgenden finden Sie Hilfestellung zu möglichen Fehlern bei der Anwendung mit *ibaPDA-Interface-VIP-TCP/UDP*. Wenden Sie sich bei weitergehenden Fragen oder im Zweifelsfall an den iba-Support.

5.1.1 Probleme mit TCP-Performance durch Delayed Acknowledge

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.

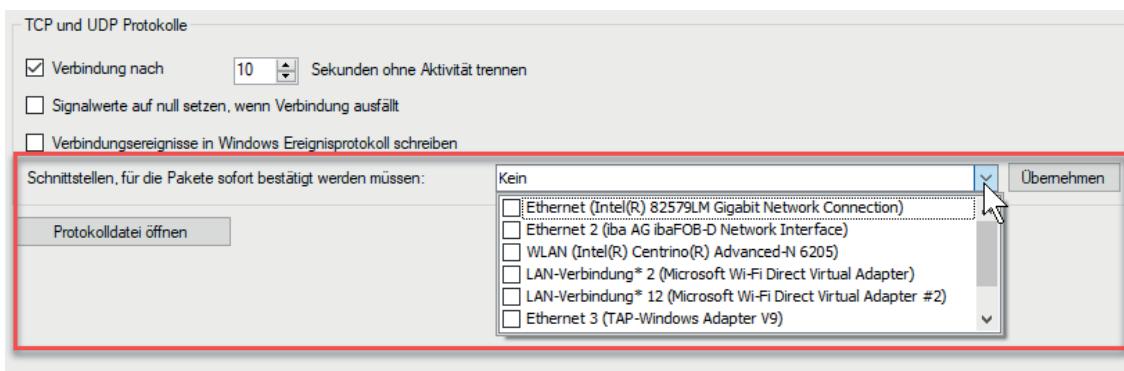
Der Datenfluss wird durch ein "Sliding Window" (Parameter Win=nnnn) gesteuert. Der Empfänger gibt an, wie viele Bytes er empfangen kann, ohne eine Quittung zu senden.

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

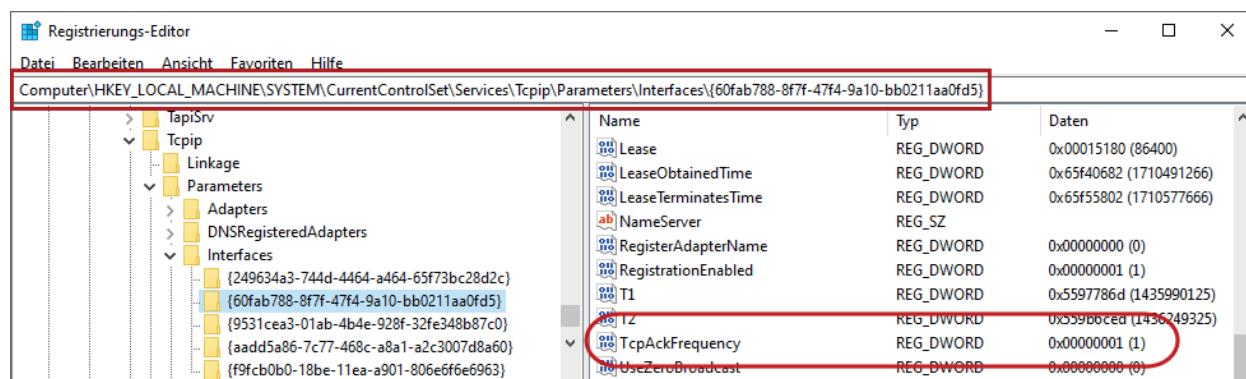
Das "Delayed Acknowledge" lässt sich einzeln pro Netzwerkadapter über einen Eintrag in der Windows Registry abschalten. Zur einfachen Änderung bietet *ibaPDA* im I/O-Manager unter *Allgemein* im Register *Einstellungen* einen entsprechenden Dialog.

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.

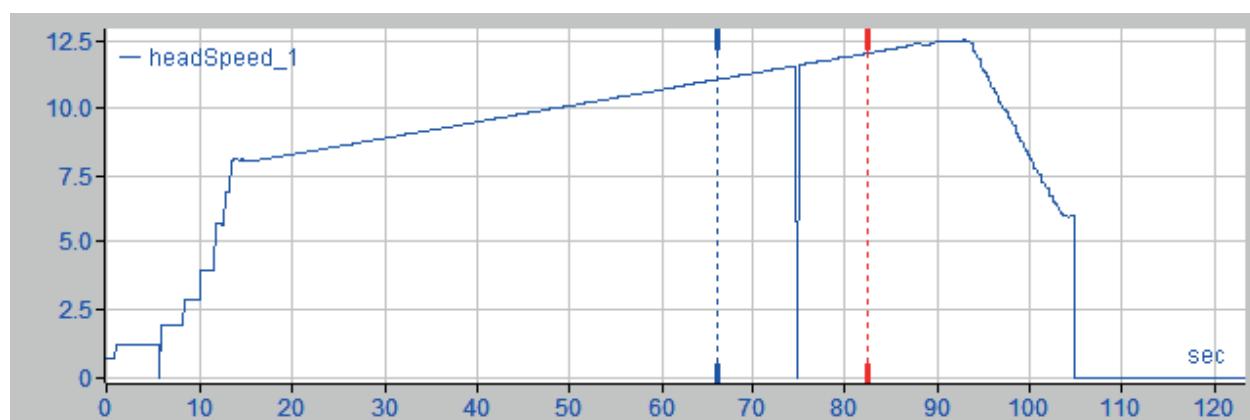
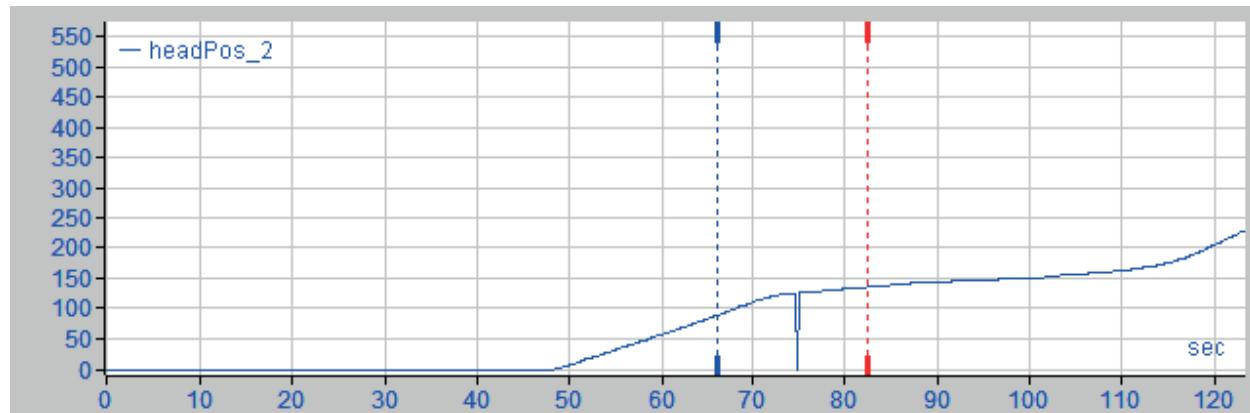
5.1.2 Unbrauchbare TCP-Daten als Folge des Nagle-Algorithmus

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 36:

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, siehe Kapitel **→ Probleme mit TCP-Performance durch Delayed Acknowledge**, Seite 36 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

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

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