



ibaPDA-Interface-EtherNet/IP

Datenschnittstelle für EtherNet/IP

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

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

© iba AG 2023, alle Rechte vorbehalten.

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

Die aktuelle Version liegt auf unserer Website www.iba-ag.com zum Download bereit.

Version	Datum	Revision	Autor	Version SW
3.0	08-2023	Neue Version ibaPDA v8	RM	8.0.0

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 diesem Handbuch.....	5
1.1	Zielgruppe und Vorkenntnisse.....	5
1.2	Schreibweisen.....	5
1.3	Verwendete Symbole.....	6
2	Systemvoraussetzungen	7
3	EtherNet/IP-Schnittstelle.....	8
3.1	Einleitung.....	8
3.2	EtherNet/IP-Kommunikation	8
3.3	ibaPDA-Modultypen	10
3.4	Konfiguration und Projektierung SPS.....	11
3.4.1	Konfiguration für I/O-Modul mit RSLogix 5000	11
3.4.2	Konfiguration für Produced Tag-Modul mit RSLogix5000.....	16
3.4.3	Konfiguration für I/O-Modul mit Schneider Electric Unity Pro XL	17
3.5	Konfiguration und Projektierung ibaPDA.....	28
3.5.1	Allgemeine Einstellungen der Schnittstelle.....	28
3.5.2	Modul hinzufügen	30
3.5.3	Allgemeine Moduleinstellungen.....	31
3.5.4	Signalkonfiguration	35
3.6	Konfiguration ibaPDA-Ausgangsmodule.....	37
3.6.1	Allgemeine Moduleinstellungen ibaPDA-Ausgangsmodule.....	37
3.6.2	Signalkonfiguration.....	38
3.7	Multiple IP-Adressen für ibaPDA anlegen.....	38
4	Diagnose und Störungsbehebung.....	40
4.1	Wichtige Eigenschaften von Ethernet Switches für Ethernet/IP.....	40
4.2	Konflikte mit anderen Ethernet/IP-bezogenen Programmen	43
4.3	Lizenz	44
4.4	Verbindungsprobleme mit Lizenzbezug.....	45
4.5	Verbindungstabelle.....	46
4.6	Verbindungsdiagnose mittels PING	48
4.7	Protokolldateien	49

4.8	Diagnosemodule.....	50
5	Support und Kontakt.....	56

1 Zu diesem Handbuch

Diese Dokumentation beschreibt die Funktion und Anwendung der Software-Schnittstelle *ibaPDA-Interface-EtherNet/IP*.

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.

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 EtherNet/IP erforderlich:

- *ibaPDA* v8.0.0 oder höher
- *ibaPDA* Basislizenz + Lizenz für *ibaPDA-Interface-EtherNet/IP*
Mit dieser Lizenz können Sie bis zu 64 Verbindungen nutzen.
- Für mehr als 64 Verbindungen benötigen Sie zusätzliche *one-step-up-Interface-Ethernet/IP-Lizenzen* für jeweils 64 Verbindungen.
- Wenn Sie noch eine ältere Lizenz *ibaPDA-Interface-EtherNet/IP* verwenden (vor Juni 2015), die keine Produced Tags oder I/O-Scanner unterstützt, Sie aber Produced Tags- oder I/O-Scanner-Module nutzen wollen, dann müssen Sie ein Upgrade auf die Zusatzlizenz *Add-on-EtherNet/IP-Produced-Tags* durchführen.

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.

Hinweis



Bei dem Betrieb auf einer virtuellen Maschine ist zu beachten, dass für jede virtuelle Maschine ein Dongle mit einer gültigen Lizenz auf dem Host gesteckt sein muss. Die verwendeten USB-Ports sind den jeweiligen virtuellen Maschinen explizit zuzuweisen.

Lizenzinformationen

Bestell-Nr.	Produktbezeichnung	Beschreibung
31.001.005	ibaPDA-Interface-EtherNet/IP	Erweiterungslizenz für ein <i>ibaPDA</i> -System um eine EtherNet/IP-Schnittstelle Anzahl der Verbindungen: 64
31.101.005	one-step-up-Interface-EtherNet/IP	Erweiterungslizenz für die Erweiterung einer vorhandenen Schnittstelle um 64 weitere EtherNet/IP-Verbindungen, maximal 3 Erweiterungslizenzen zulässig
31.111.005	Add-on-EtherNet/IP-Produced-Tags	Erweiterungslizenz zur Nutzung von Produced Tags; gilt nur für ältere Lizenzen ohne Produced Tags-Unterstützung

Tab. 1: Verfügbare EtherNet/IP-Lizenzen

3 EtherNet/IP-Schnittstelle

3.1 Einleitung

iba hat einen TCP/IP-Treiber entwickelt, der das EtherNet/IP-Protokoll via TCP/IP und UDP lesen kann. Der Treiber arbeitet eng mit der *ibaPDA*-Software zusammen. Je nach ausgewählter Modularität bzw. Kommunikationsart agiert *ibaPDA* als ein Gerät, das passiv auf Verbindungen von einem Scanner wartet (I/O-Modul) oder als ein Scanner, der aktiv die Verbindungen (Produced Tag, I/O-Scanner) aufbaut.

Des Weiteren kann *ibaPDA* Daten wie Ausgangssignale zurück an den EtherNet/IP-Controller senden (nur I/O-Modul und I/O-Scanner).

Die folgenden Controller wurden erfolgreich getestet:

Rockwell Automation Controller:

(Verwendete Modularitäten in Klammern)

- CompactLogix (I/O-Module, Produced Tag)
- FlexLogix (I/O-Module, Produced Tag)
- ControlLogix (I/O-Module, Produced Tag)
- SoftLogix 5800 (I/O-Module, Produced Tag)
- MicroLogix (I/O-Module, Produced Tag)

Andere Controller:

- Schneider Electric M580 ePAC (I/O-Module)
- Keyence CB-EP100 (I/O-Scanner)
- WAGO 750-352 (I/O-Scanner)
- Omron C2JM (Produced Tag)

3.2 EtherNet/IP-Kommunikation

Das Ethernet Industrial Protocol (EtherNet/IP) ist ein offener industrieller Netzwerk-Standard, der sowohl Real-Time E/A Nachrichten als auch den Nachrichtenaustausch unterstützt.

Das EtherNet/IP-Netzwerk bietet eine ganze Bandbreite an Kontroll-, Konfigurierungs- und Datenerfassungsdiensten an. EtherNet/IP nutzt TCP/IP für allgemeine Nachrichten-/Informationsaustausch-Dienste und UDP/IP für E/A Nachrichten-Dienste für Steuerungsanwendungen. Diese Verbindung von allgemein akzeptierten Standards sorgt für die Funktionalität, die benötigt wird, um sowohl Informationsdatenaustausch als auch Steuerungsanwendungen zu unterstützen.

EtherNet/IP definiert zwei primäre Kommunikationsarten: „explicit“ und „implicit“.

CIP Telegrammtyp	CIP Kommunikationsbeziehung	Transportprotokoll	Kommunikationsart	Typische Anwendung
Explicit	Verbunden oder nicht verbunden	TCP/IP	Anforderung/ Antwort-Transaktionen	Zeitunkritische Informationsdaten
Implicit	Verbunden	UDP/IP	I/O Datentransfers	Echtzeit-I/O-Daten

Tab. 2: EtherNet/IP-Kommunikationsarten

Hinweis

CIP (Common Industrial Product) ist ein Protokoll für industrielle Automatisierungsanwendungen. CIP umfasst eine vielfältige Sammlung von Nachrichten und Diensten zur Unterstützung von Applikationen der Fertigungsautomatisierung – Steuerung, Sicherheit, Synchronisation, Bewegung, Konfiguration und Information.

Explicit Messaging funktioniert allgemein nach einem Anforderung/Antwort-Mechanismus (oder Client/Server). Diese Kommunikationsart wird für Daten verwendet, die nicht in Echtzeit übertragen werden, meist Informationen. Für Ethernet/IP: Explicit Messaging verwendet TCP.

Implicit Messaging wird oft auch als "I/O" bezeichnet und ist zeitkritisch. Üblicherweise wird diese Kommunikationsart für Echtzeit-Datenaustausch genutzt, wenn Geschwindigkeit und niedrige Latenz wichtig sind. Implicit Messaging-Telegramme enthalten sehr wenige Informationen über ihre Bedeutung, so dass die Übertragung zwar sehr effizient aber weniger flexibel als bei Explicit Messaging ist. Die Interpretation der übertragenen Daten ist schnell. Mit Implicit Messaging bauen Sie einen Verband (eine „CIP-Verbindung“) zwischen zwei Geräten auf und erzeugen Implicit-Telegramme gemäß zuvor bestimmter Triggermechanismen, typischerweise in einer spezifizierten Paketrage. Beide Geräte kennen und akzeptieren die Datenformate, die sie nutzen werden („impliziertes“ Format). Für EtherNet/IP: Implicit Messaging verwendet UDP und unterstützt Multicast oder Unicast.

Hinweis

ibaPDA unterstützt sowohl Multicast- als auch Unicast-Telegramme in seinen Modultypen "I/O-Modul" und "I/O-Scanner". Der Modultyp "Produced Tag" unterstützt nur Unicast.

Trotzdem empfiehlt iba die Verwendung von Unicast-Telegrammen, um einen Multicast-Überlauf im Netzwerk zu vermeiden. Die Nutzung eines separaten Netzwerks wird ebenfalls empfohlen.

3.3 ibaPDA-Modultypen

ibaPDA kann unterschiedliche Methoden verwenden, um sich mit einem Controller oder einem Gerät zu verbinden. Diese Methoden werden mit drei unterschiedlichen Modultypen realisiert.

Modultyp EtherNet/IP...	Scanner	Gerät	Dateneingang für ibaPDA	Datenausgang für ibaPDA
I/O-Modul	SPS	PDA	Output Assembly-Instanz des Geräts (=PDA), identifiziert durch seine Nummer	Input Assembly-Instanz des Geräts (=PDA), identifiziert durch seine Nummer
Produced Tag	PDA	EIP-Gerät oder Controller, z. B. ControlLogix, Omron, ...	Ein Tag, identifiziert durch seinen Namen	Kein
I/O-Scanner	PDA	EIP-Gerät, z.B. Keyence CB-EP100	Input Assembly-Instanz des Geräts (=CB-EP100), identifiziert durch seine Nummer	Output Assembly-Instanz des Geräts (=CB-EP100), identifiziert durch seine Nummer

Tab. 3: Kommunikationsmethoden der 3 Modultypen

Allgemeines Prinzip

Der Scanner initialisiert die Verbindung mit Senden eines Forward-Open-Befehls. In dem Forward-Open-Befehl muss der Scanner angeben, welche Daten er austauschen will (Assembly-Instanz oder Tag) und wie er sie austauschen will (Unicast/Multicast, Zykluszeit, Timeouts, Datengröße usw.).

I/O-Modul

Bei Nutzung des Modultyps I/O agiert *ibaPDA* wie ein I/O-Adapter oder Gerät. Der SPS-Controller ist ein Scanner. Es wird nur Implicit Messaging angewendet. *ibaPDA* wartet passiv auf den Verbindungsaufbau durch die SPS und die Übertragung der Daten. Daher wird bei der Modulkonfiguration keine IP-Adresse benötigt.

Produced Tag

Bei Nutzung des Modultyps Produced Tag agiert *ibaPDA* wie ein Scanner. Der SPS-Controller ist ein I/O-Adapter oder Gerät. Explicit Messaging wird von *ibaPDA* für einige herstellereigenspezifische Objekte verwendet, um Informationen zur Verfügbarkeit und Konfiguration von Produced Tags zu erhalten (gilt nur für Rockwell-SPSen). Als Scanner baut *ibaPDA* die Verbindung aktiv mittels Explicit Messaging auf. Daher werden bei der Modulkonfiguration die IP-Adresse (des Controllers), der Produced Tag und seine Größe benötigt. Die Daten von der SPS oder dem Gerät werden schließlich per Implicit Messaging übertragen.

Zurzeit kann *ibaPDA* nur Tag-Informationen von Rockwell SPSen lesen und nicht von SPSen anderer Hersteller. Aber es ist möglich, den Tag-Namen manuell einzugeben und dann die Signale zu konfigurieren, so dass sie der Tag-Struktur entsprechen.

Der Modultyp Produced Tag unterstützt CIP-Routing, d. h. eine Verbindung auch über wechselnde Bussysteme (EtherNet/IP, ContolNet, DeviceNet) hinweg, wenn die Zielsteuerung nicht direkt von *ibaPDA* erreichbar ist.

Hinweis



Wenn die Erfassung startet, dann versucht *ibaPDA* sich mit dem Produced Tag auf der CPU zu verbinden. Schlägt dies fehl, dann gibt *ibaPDA* einen Fehler im Validierungsdialog aus, aber die Erfassung startet trotzdem. *ibaPDA* versucht dann periodisch sich mit dem Produced Tag zu verbinden, solange die Erfassung läuft. Wenn die Erfassung stoppt, löst *ibaPDA* die Verbindung zum Produced Tag.

I/O-Scanner

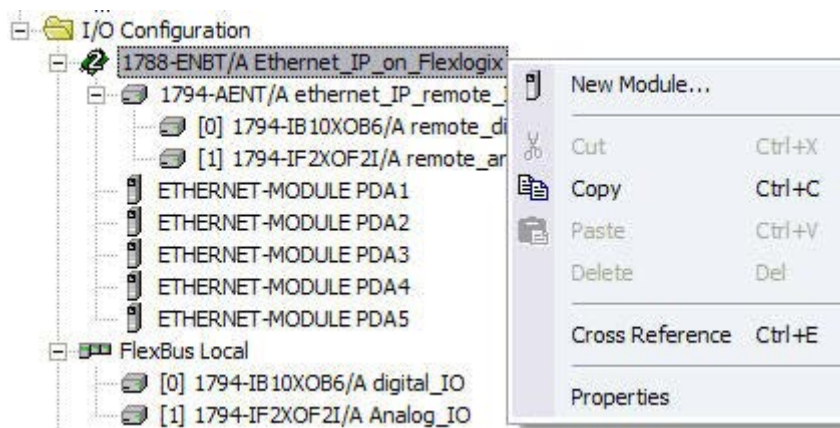
Bei Nutzung des Modultyps I/O-Scanner agiert *ibaPDA* wie ein Scanner und baut die Verbindung zu einem Gerät auf. Bei solchen Geräten handelt es sich üblicherweise um Remote I/O-Units (z. B. von WAGO) oder Messgeräte (z. B. von Keyence). Für die Modulkonfiguration werden die IP-Adresse (des Geräts), die Assembly-Instanz und deren Größe benötigt. Die Daten vom Gerät werden schließlich per Implicit Messaging übertragen.

3.4 Konfiguration und Projektierung SPS

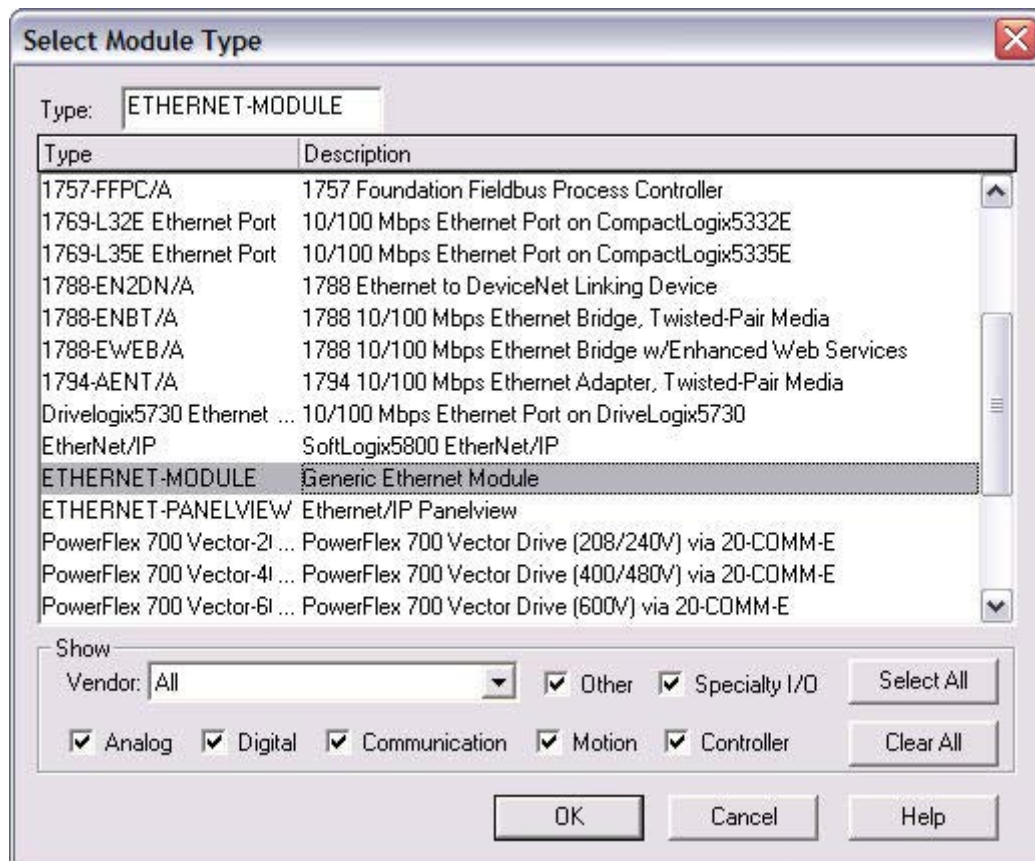
3.4.1 Konfiguration für I/O-Modul mit RSLogix 5000

(nur Geräte von Allen-Bradley/Rockwell)

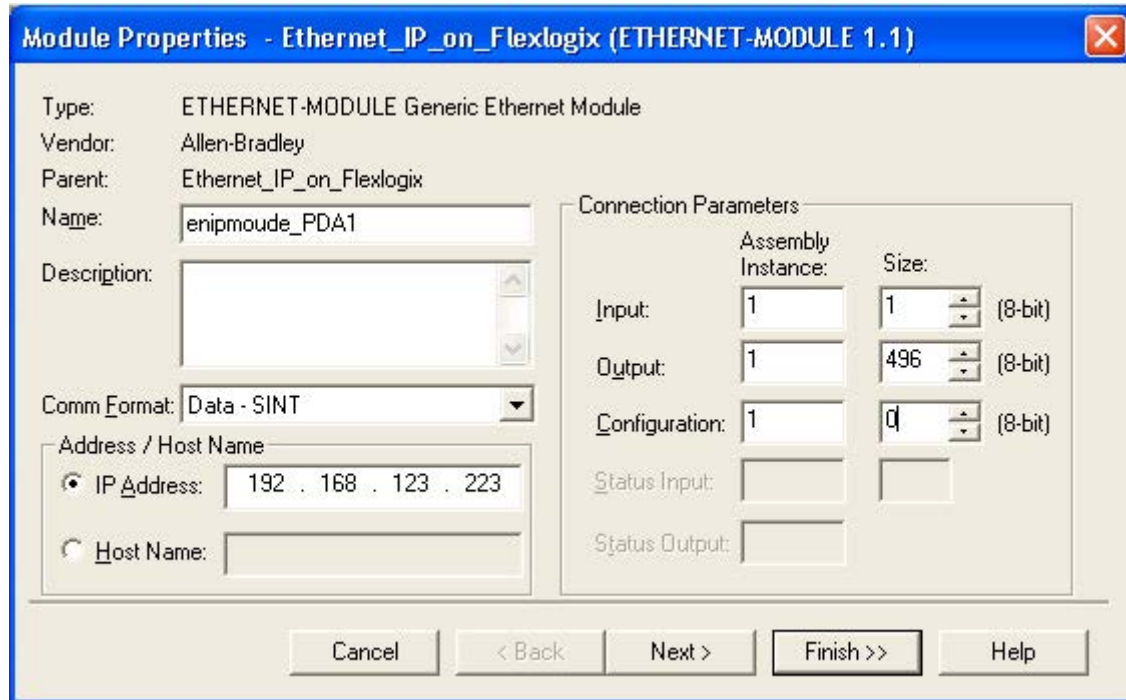
1. Fügen Sie in RSLogix 5000 ein neues "ETHERNET-MODULE" zu der EtherNet/IP- Schnittstellenkarte hinzu (z. B. 1788-ENBT).
2. Klicken Sie mit der rechten Maustaste auf die EtherNet/IP-Schnittstellenkarte. Das Kontextmenü wird angezeigt.



3. Klicken Sie auf *New Module...*
4. In der Liste der Modultypen, wählen Sie das "ETHERNET-MODULE" mit der Beschreibung "Generic Ethernet Module".



5. Nachdem Sie das neue Modul angelegt haben, sehen Sie den Dialog *Module Properties*.



In diesem Dialog konfigurieren Sie die folgenden Einstellungen:

- Name des Moduls (das Array, das später genutzt wird, um Daten nach *ibaPDA* zu kopieren, trägt den Namen des Moduls)
- Comm(unication) Format (siehe unten)
- IP-Adresse oder Host-Name des *ibaPDA*-Systems (Empfehlung: IP-Adresse angeben)
- Assembly Instanzen und Größen (die Assembly-Instanznummern müssen für alle drei Assemblies identisch sein). Wenn die Größe in der Konfigurierung 0 beträgt, können diese Daten nicht in *ibaPDA* verarbeitet werden.

Die Instanznummern können zwischen 1 und 64 liegen.

Es können bis zu 4 EtherNet/IP-Lizenzen mit einer Gesamtanzahl von bis zu 256 Verbindungen aktiviert werden. In diesem Fall kann die Instanznummer bis 256 betragen.

Hinweis



Wenn mehr als ein Generic-Ethernet-Modul mit Verbindung zu *ibaPDA* benötigt wird, geben Sie bitte verschiedene IP-Adressen für jedes Modul ein. Jedes Generic-Ethernet-Modul entspricht einer TCP/IP-Verbindung, die wiederum einem Modul in *ibaPDA* entspricht. Aus diesem Grunde müssen dem Netzwerk-Adapter von *ibaPDA* mehrere IP-Adressen zugewiesen werden.

Siehe ➤ *Konfiguration und Projektierung ibaPDA*, Seite 28 für Setup-Informationen.

Die folgenden Datentypen sind für die Nutzung in *ibaPDA* möglich:

- SINT (8 bit integer)
- INT (16 bit integer)
- DINT (32 bit integer)
- REAL (32 bit real)

Nur Datentypen ohne den Zusatz "with status" sind erlaubt.

Bedeutung von Assemblies und Größendefinitionen:

- Die Größenangaben sind definiert als Mehrfaches des Datentyps. Wenn Sie "Data - DINT" (32 Bit) und als Größe 4 verwenden, können vier 32-Bit Integer übertragen werden.
- Input: Um den Netzwerkverkehr zu reduzieren, sollte der Input in die SPS aus *ibaPDA* die Größe eins (1) haben. Wenn Sie auch Daten von *ibaPDA* zurück an die SPS senden wollen, dann bestimmt die Input-Größe die Länge der Daten, die *ibaPDA* senden kann. Der Maximalwert für RSLogix beträgt 500 Bytes. Sollten Sie einen anderen Datentyp als SINT verwenden, liegt der Maximalwert bei 500 geteilt durch die Größe des Datentyps in Byte.
- Output: An *ibaPDA* übertragene Daten (Maximalwert: 496 Byte). Sollten Sie einen anderen Datentyp als SINT verwenden, liegt die Maximalgröße bei 496 geteilt durch die Größe des Datentyps in Byte.
- Konfigurierung: Legt fest, wie viele Konfigurationsdaten-Bytes innerhalb des Forward-Open-Befehls übertragen werden können. *ibaPDA* unterstützt die Konfigurierung über diesen Befehl nicht. Tragen Sie hier also 0 als Größe ein.

Hinweis

- Die Assembly-Instanzen für Input, Output und Konfigurierung **müssen identisch sein**.

- Die Assembly-Instanznummern müssen zwischen 1 und 64 liegen.

Es können bis zu 4 EtherNet/IP-Lizenzen mit einer Gesamtanzahl von bis zu 256 Verbindungen aktiviert werden. In diesem Fall kann die Instanznummer bis 256 betragen.

- Diese Assembly-Instanznummer entspricht der EtherNet/IP-Verbindungsnummer in *ibaPDA*. Die Nummer sollte entsprechend einmalig für ein *ibaPDA* sein.

Konfigurieren Sie nicht zwei Verbindungen mit identischen Assemblies, die eine Verbindung zu demselben *ibaPDA*-Host herstellen wollen!

Hinweis

Die Größe von Inputs UND Outputs wird stets in der SPS-Konfiguration definiert (z. B. in RSLogix oder in Unity Pro von Schneider SPSen).

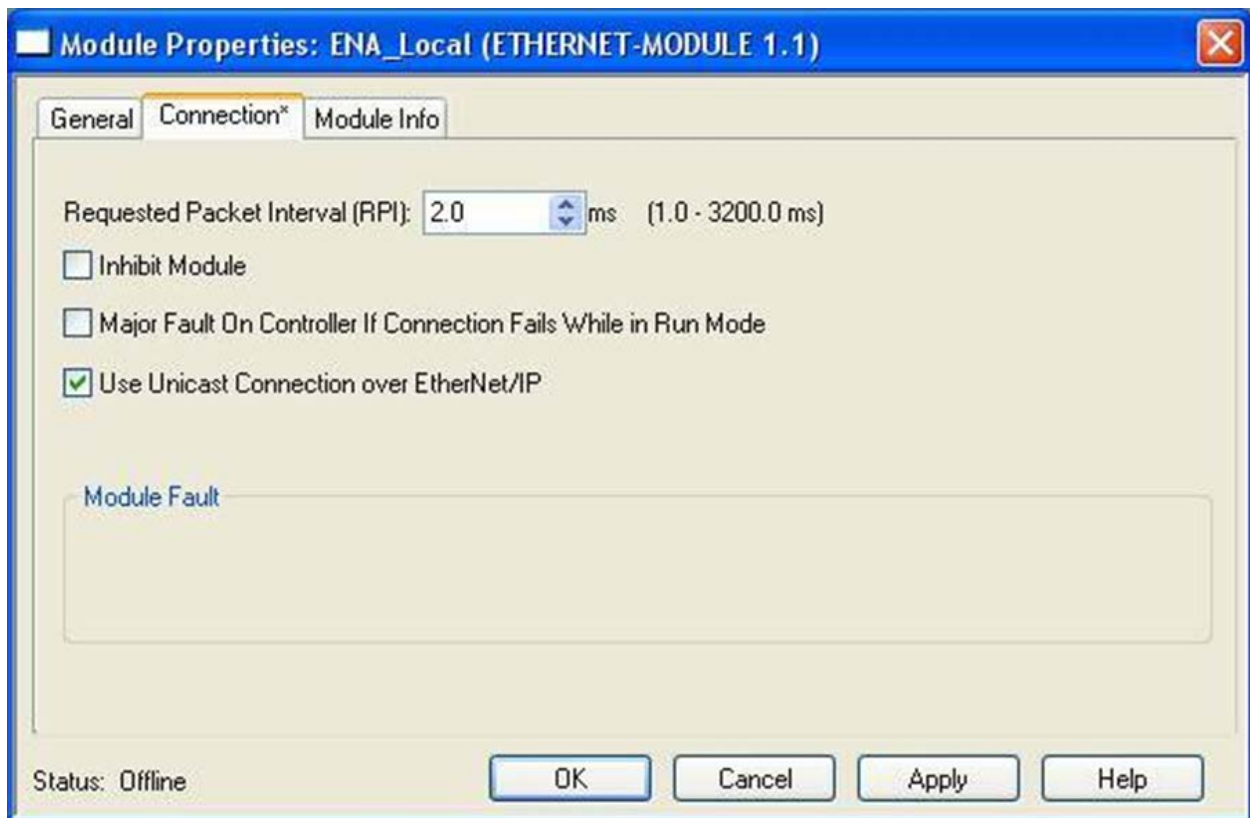
ibaPDA prüft die Größe anhand der Forward-Open-Information und warnt den Anwender, wenn Signal-Offset + Größe (Datentyp) in *ibaPDA* größer ist als die auf SPS-Seite definierte Größe.

Wenn Adress-Offset + Größe (Datentyp) größer ist als 1024 Bytes für die Inputs und 1004 Bytes für die Outputs von *ibaPDA*, dann wird die Validierung mit einer Fehlermeldung abgebrochen.

Wenn Sie auf <Next> klicken, gelangen Sie zur Definition des Zeitintervalls. *ibaPDA* unterstützt Werte zwischen 1 ms und 1000 ms. Je nachdem, welche SPS und welche Programme Sie auf dem Gerät haben, können Sie keine sehr kleinen Intervalle nutzen (z. B. arbeitet eine Allen-Bradley FlexLogic nicht stabil, wenn der Wert unter 3 ms liegt).

Es werden Unicast Reply Verbindungen über EtherNet/IP unterstützt. Sie können also die Option *Use Unicast...* aktivieren.

Die Konfigurierung der Kommunikation ist mit Unicast etwas einfacher, da keine präventiven Maßnahmen gegen unerwünschte Effekte von Multicast getroffen werden müssen. IGMP Snooping und IGMP Query sind nicht erforderlich.



Nachdem das neue Modul hinzugefügt wurde, können Daten von der SPS innerhalb von Programmen und Tasks in das neue Array kopiert werden (welches den Namen des Moduls trägt).

Dieser Array hat die folgenden drei Teilnehmer:

- Input: Enthält alle Daten, die von *ibaPDA* an die SPS übertragen werden.
- Output: Enthält alle Daten, die über EtherNet/IP an *ibaPDA* gesendet werden.
- Config: Senden von Konfigurierungswerten mit dem Befehl "ForwardOpen". Wird von *ibaPDA* nicht unterstützt.
Das Config-Array wird auch dann existieren, wenn das ETHERNET-MODULE mit einer Konfigurierungsgröße von 0 belegt wurde.

Jedes dieser Sub-Arrays hat eine Option "Daten", welche die Werte enthält. Jede Task in der SPS hat Schreib- und Lesezugriff auf diese Option "Daten". Es ist allerdings nur sinnvoll, die "Output"-Daten zu schreiben.

Wenn Sie Daten nach *ibaPDA* kopieren möchten, d. h. in das Output-Array, können Sie einen der folgenden Befehle nutzen (Ladder-, Funktionsbaustein- oder Structured Text-Befehle):

- MOV – Kopiert einen Tag (oder ein Element eines Arrays oder einer Struktur) an ein Ziel (hier: ein Element des Output-Array). Die Quelle und das Ziel müssen denselben Datentyp aufweisen. Andernfalls wird die SPS die Daten in den Datentyp des Ziels umwandeln oder ein Fehler tritt auf. Sie können diese Option nutzen, wenn Sie Daten nur eines Typs innerhalb einer EtherNet/IP-Verbindung an *ibaPDA* übertragen.
- COP – Kopiert einen Tag (oder ein Element eines Arrays oder einer Struktur) an ein Ziel (hier: ein Element des Output-Array) mit einer festgelegten Länge. Sie können die Länge festlegen.

Die Quelle und das Ziel können verschiedene Datentypen aufweisen. Nichts wird von der SPS verworfen. Wenn Sie das ETHERNET-MODULE als Datentyp "SINT" (8 bit) definieren, können Sie die Daten beliebig von der SPS kopieren (Sie können eine eigene Struktur anlegen). Mit dieser Option ist es möglich, verschiedene Datentypen innerhalb derselben Verbindung an *ibaPDA* zu übertragen.

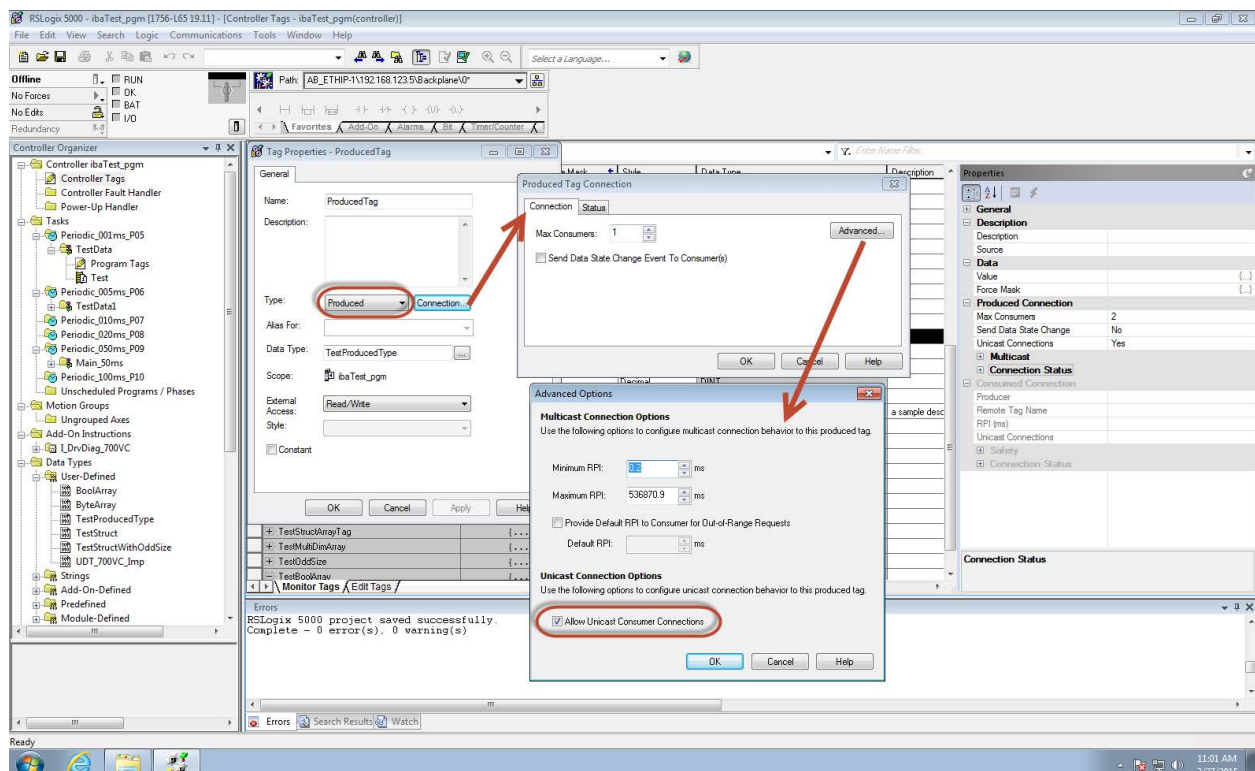
Andere Dokumentation



Schlagen Sie bitte für weiterführende Informationen in der RSLogix-Dokumentation nach.

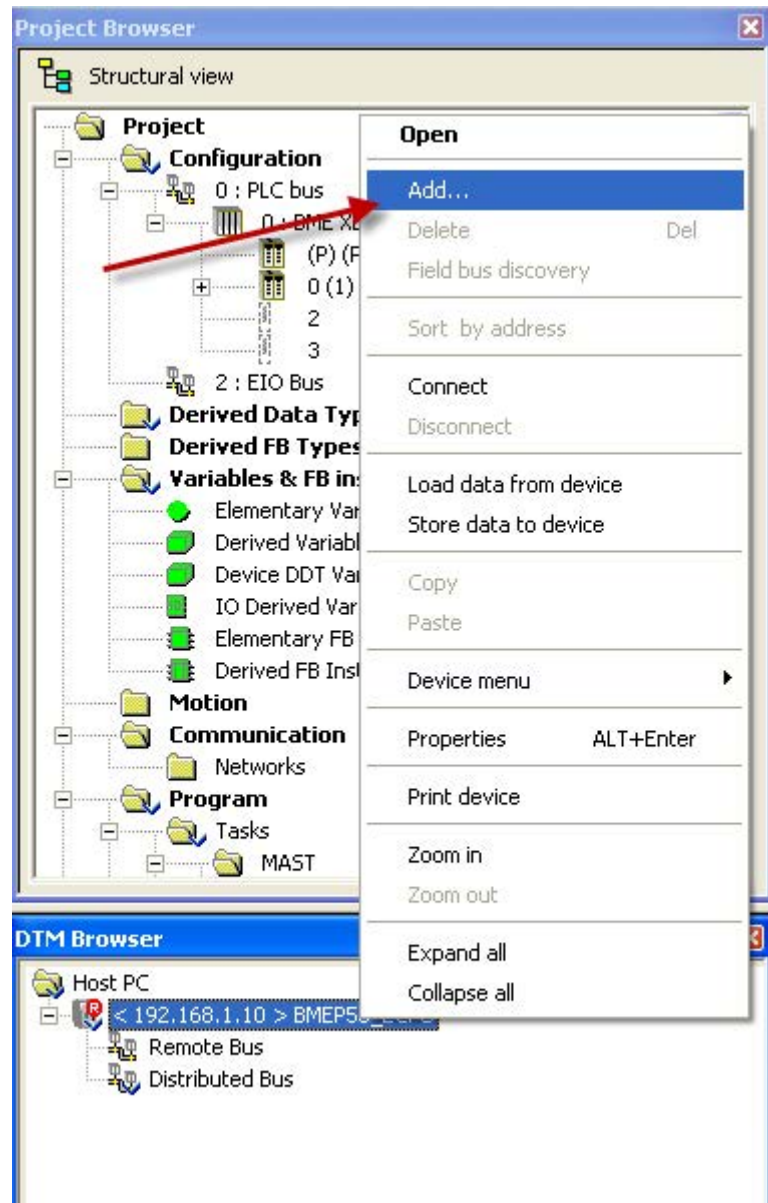
3.4.2 Konfiguration für Produced Tag-Modul mit RSLogix5000

In RSLogix5000 und Studio5000 können Sie einen Controller-Tag als Produced Tag konfigurieren. Ein Produced Tag erlaubt anderen Systemen sich mit ihm zu verbinden und periodisch Daten des Tags zu empfangen. In den Tag-Eigenschaften können Sie den Typ auf *Produced* einstellen. Standardmäßig erlaubt ein Produced Tag einen „Consumer“, aber Sie können die Anzahl erhöhen. Standardmäßig werden Unicast-Verbindungen erlaubt. *ibaPDA* unterstützt für Produced Tags nur Unicast-Verbindungen. Stellen Sie daher sicher, dass Unicast aktiviert ist. Ein Produced Tag hat normalerweise einen anwenderdefinierten Datentyp. Die Maximalgröße der Daten beträgt 500 Bytes.

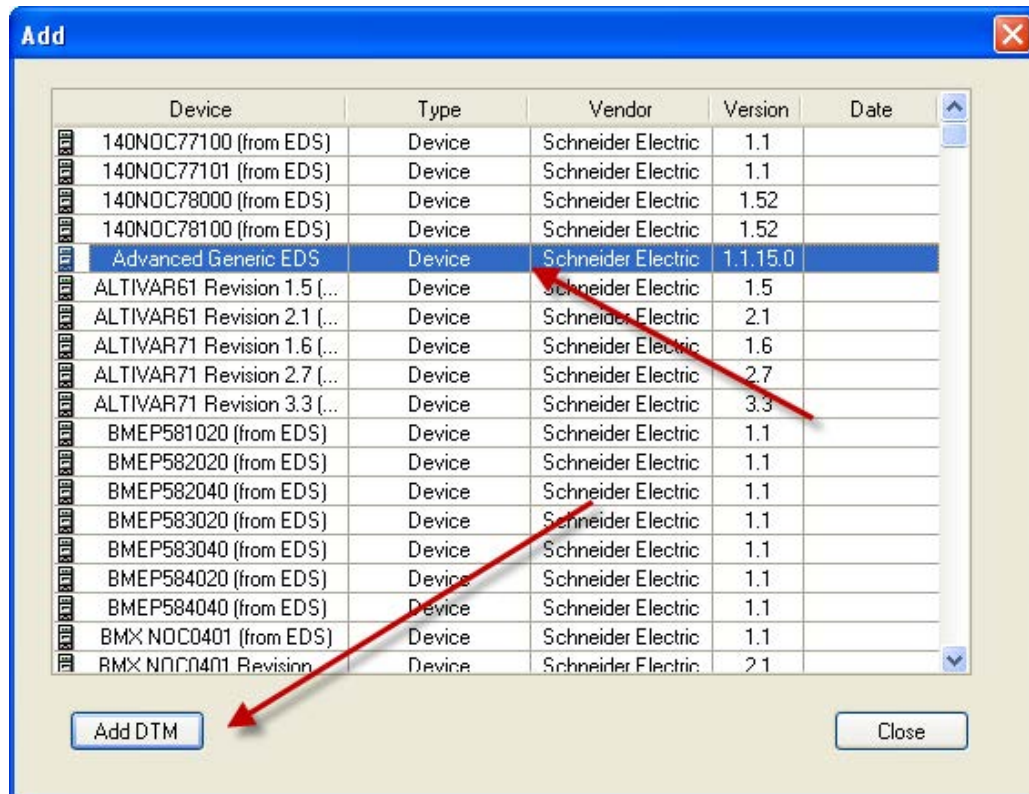


3.4.3 Konfiguration für I/O-Modul mit Schneider Electric Unity Pro XL

1. Öffnen Sie den Device Type Manager (DTM) Browser in Unity Pro, machen Sie einen rechten Mausklick auf den Controller und klicken Sie *Add*:



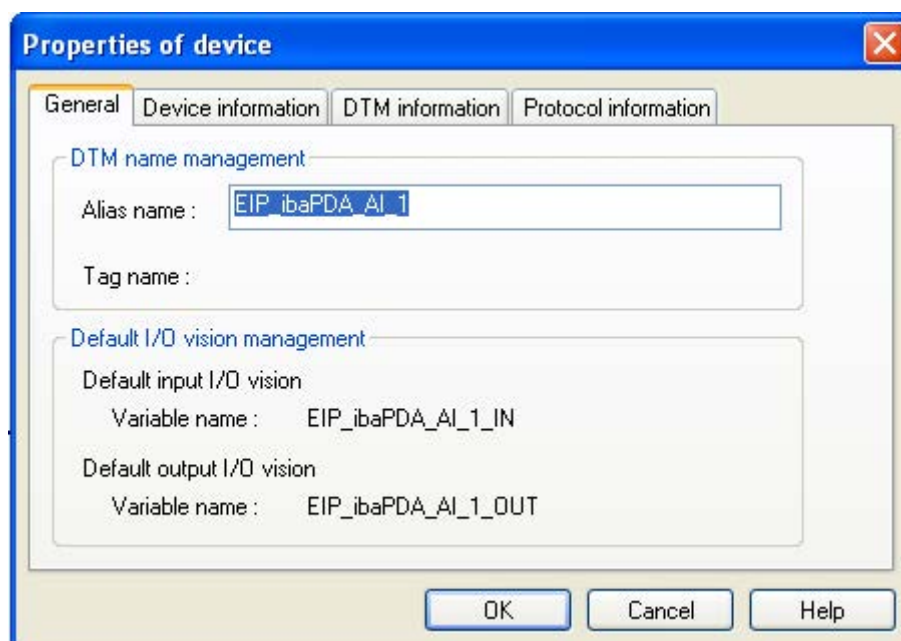
2. Wählen Sie das "Advanced Generic EDS" Device und klicken Sie <Add DTM>.



3. Es erscheint der Eigenschaften-Dialog (Properties):

Geben Sie dem Gerät einen Alias-Namen wie unten gezeigt.

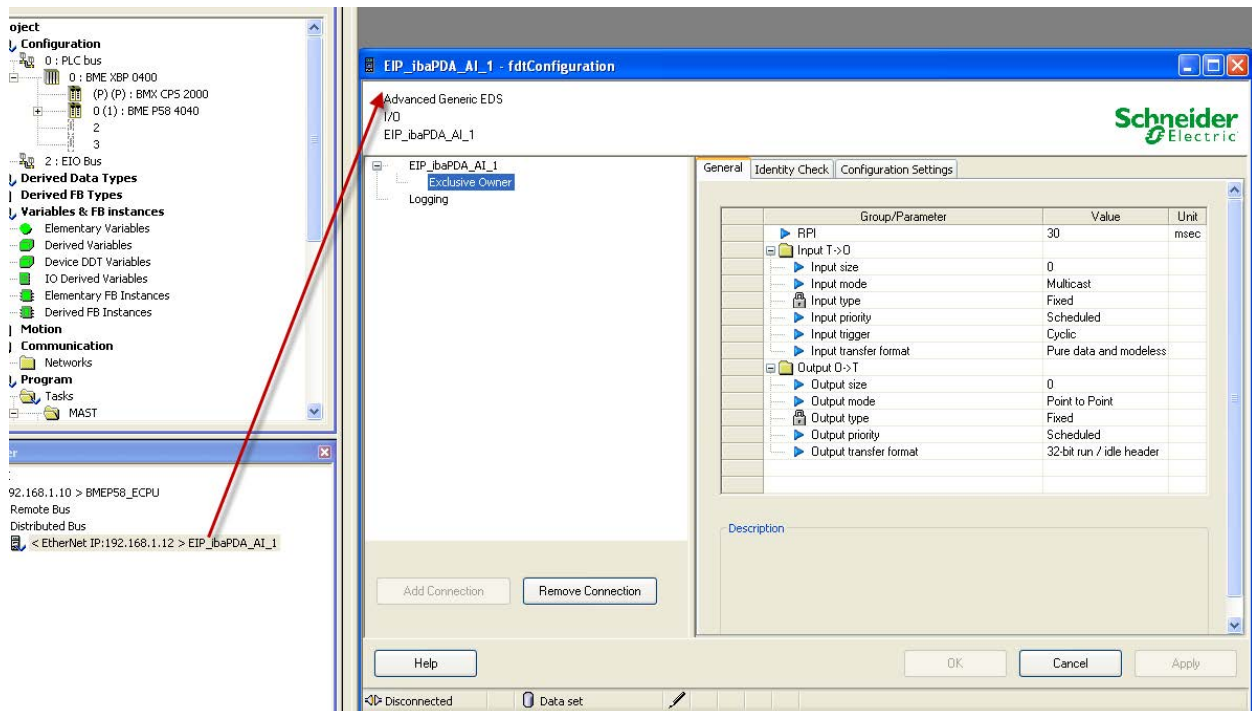
Dieser Name wird für die Erzeugung der I/O-Variablenamen verwendet. Geben Sie also einen verständlichen Namen, der das Gerät beschreibt. Das Beispiel unten zeigt ein EtherNet/IP-Modul für das Senden von Daten an *ibaPDA* mit Assembly Instanz 1.



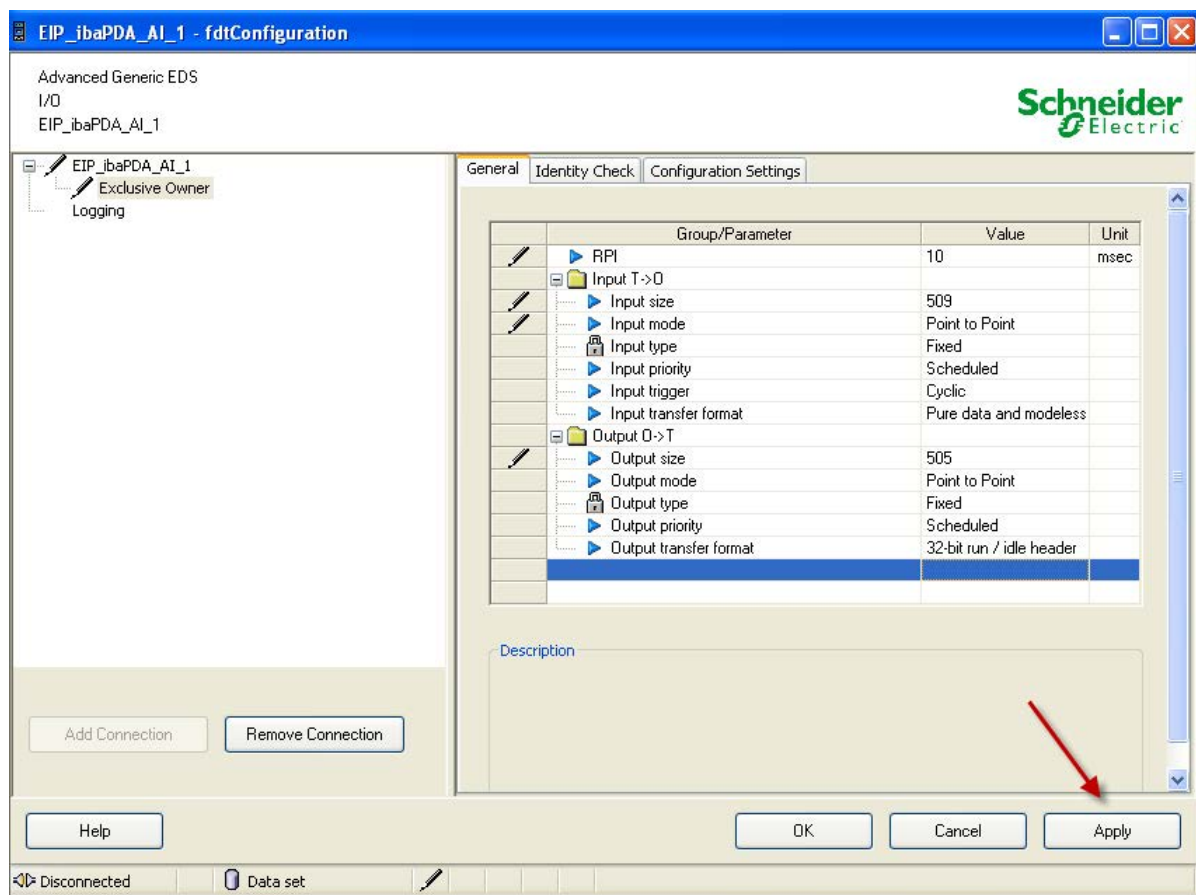
Klicken Sie auf <OK>, um das Gerät dem DTM-Browser hinzuzufügen.

4. Machen Sie einen Doppelklick auf das neu hinzugefügte Gerät im DTM-Browser:

Der Konfigurationsdialog öffnet sich. Wählen Sie das "Exclusive Owner"-Element, wie unten gezeigt.



5. Im Register *General* editieren und konfigurieren Sie die folgenden Einstellungen:



- RPI:

Requested Packet Interval in ms. *ibaPDA* unterstützt Werte zwischen 1 ms und 1000 ms. Je nach SPS-Typ und den Programmen, die auf dem Gerät laufen, können Sie keine sehr kurzen Intervalle nutzen (z. B. liegt die untere Grenze bei der SPS M580 bei 2 ms).

- Input T-> O

Input size: Die Anzahl Bytes, die für Daten von *ibaPDA* reserviert sind. Um Netzwerklast zu minimieren, sollte die Größe 4 Byte betragen, wenn kein Input von *ibaPDA* erforderlich ist. Der Maximalwert beträgt 509 Bytes.

Input mode: *ibaPDA* unterstützt Unicast Reply-Verbindungen über EtherNet/IP. Somit können Sie die Option *Point to point* aktivieren. Die Konfigurierung der Kommunikation ist mit Unicast etwas einfacher, da keine präventiven Maßnahmen gegen unerwünschte Effekte von Multicast getroffen werden müssen. IGMP Snooping und IGMP Query sind nicht erforderlich.

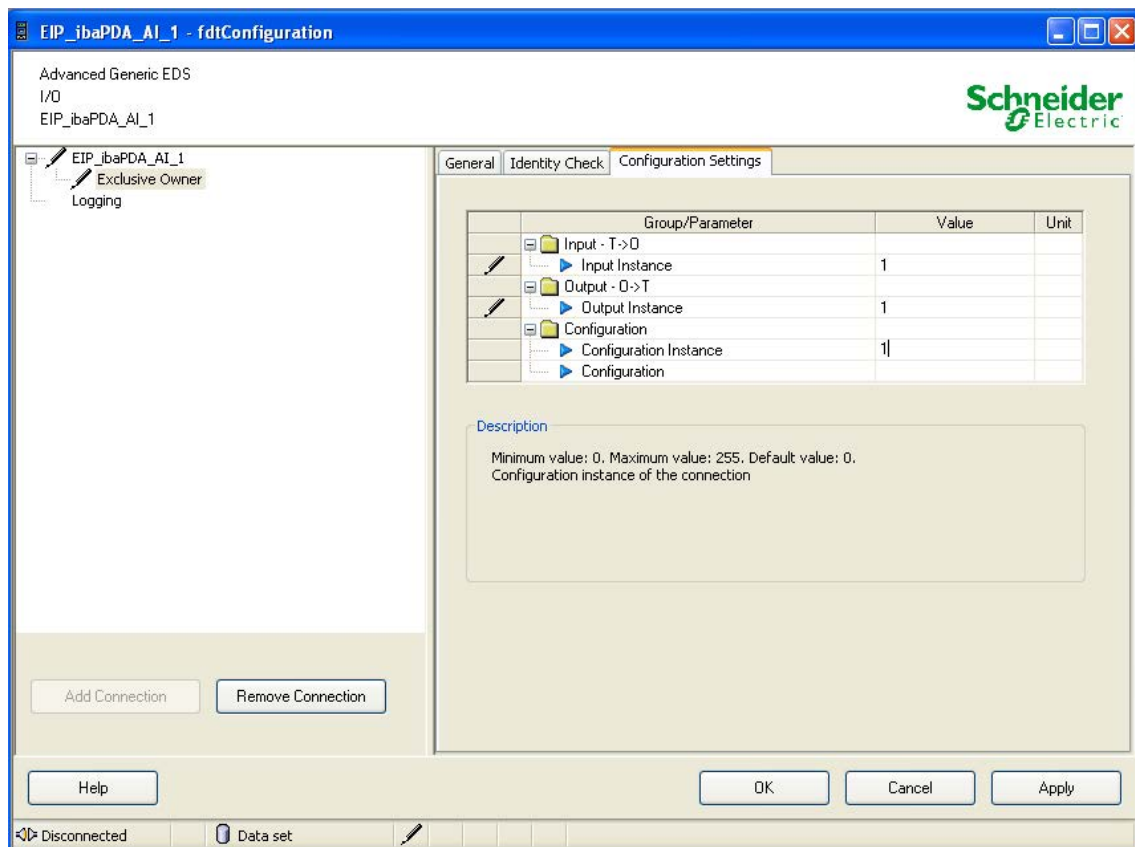
- Output O -> T

Output size: Die Anzahl Bytes, die für Daten an *ibaPDA* reserviert sind. Der Maximalwert beträgt 505 Bytes.

Output mode: *ibaPDA* unterstützt Unicast Reply-Verbindungen über EtherNet/IP. Somit können Sie die Option *Point to point* aktivieren.

6. Klicken Sie auf <Apply>, um die Einstellungen zu übernehmen.

7. Im Register *Configuration Settings* nehmen Sie die folgenden Einstellungen vor:



- Input Instance:

Die gerätespezifische Assembly-Nummer, die Eingangsübertragungen (Inputs, T -> O) zugeordnet ist.

- Output Instance:

Die gerätespezifische Assembly-Nummer, die Ausgangsübertragungen (Outputs, O-> T) zugeordnet ist.

- Configuration Instance:

Die gerätespezifische Assembly-Nummer, die Gerätekonfigurationseinstellungen zugeordnet ist.

Hinweis



- Die Assembly-Instanzen für Input, Output und Konfigurierung **müssen identisch sein**.

- Die Assembly-Instanznummern müssen zwischen 1 und 64 liegen.

Es können bis zu 4 EtherNet/IP-Lizenzen mit einer Gesamtanzahl von bis zu 256 Verbindungen aktiviert werden. In diesem Fall kann die Instanznummer bis 256 betragen.

- Die Assembly-Instanznummer entspricht der EtherNet/IP-Verbindungsnummer in *ibaPDA*. Die Nummer sollte entsprechend einmalig für ein *ibaPDA* sein.

Konfigurieren Sie nicht zwei Verbindungen mit identischen Assemblies, die eine Verbindung zu demselben *ibaPDA*-Host herstellen wollen!

Hinweis



Die Größe von Inputs UND Outputs wird stets in der SPS-Konfiguration definiert (z. B. in RSLogix oder in Unity Pro von Schneider SPSen).

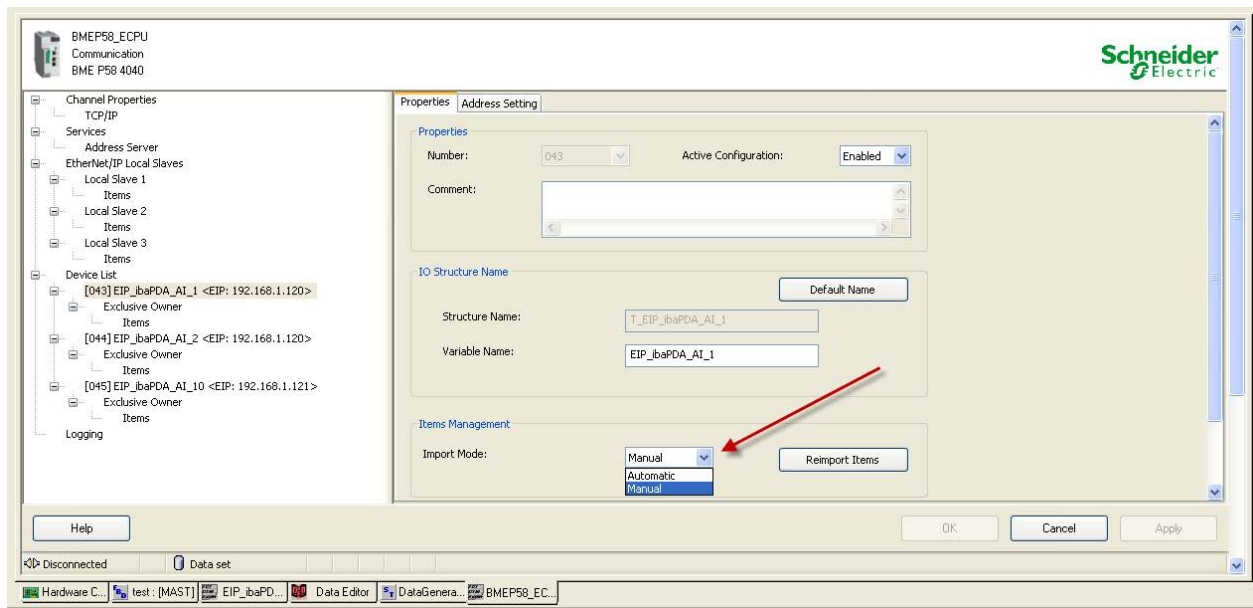
ibaPDA prüft die Größe anhand der Forward-Open-Information und warnt den Anwender, wenn Signal-Offset + Größe (Datentyp) in *ibaPDA* größer ist als die auf SPS-Seite definierte Größe.

Wenn Adress-Offset + Größe (Datentyp) größer ist als 1024 Bytes für die Inputs und 1004 Bytes für die Outputs von *ibaPDA*, dann wird die Validierung mit einer Fehlermeldung abgebrochen.

8. Klicken Sie auf <Apply>, um die Einstellungen zu übernehmen.

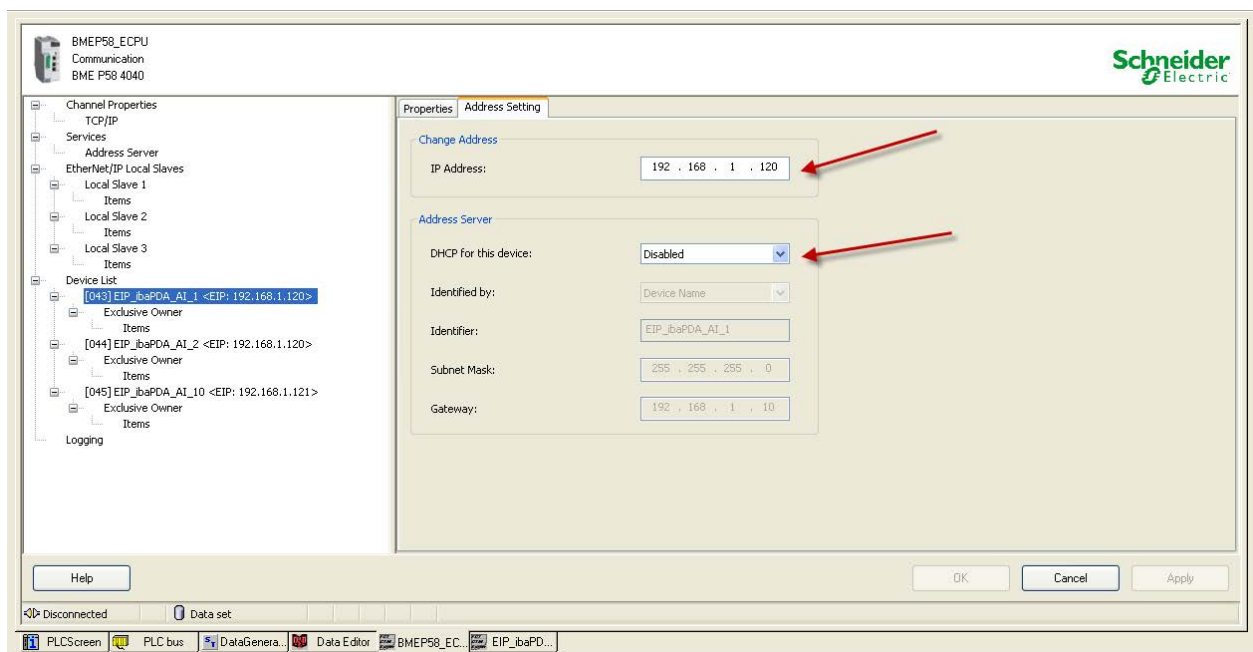
9. Machen Sie einen Doppelklick auf die CPU im DTM-Browser.

Der folgende Dialog öffnet sich:

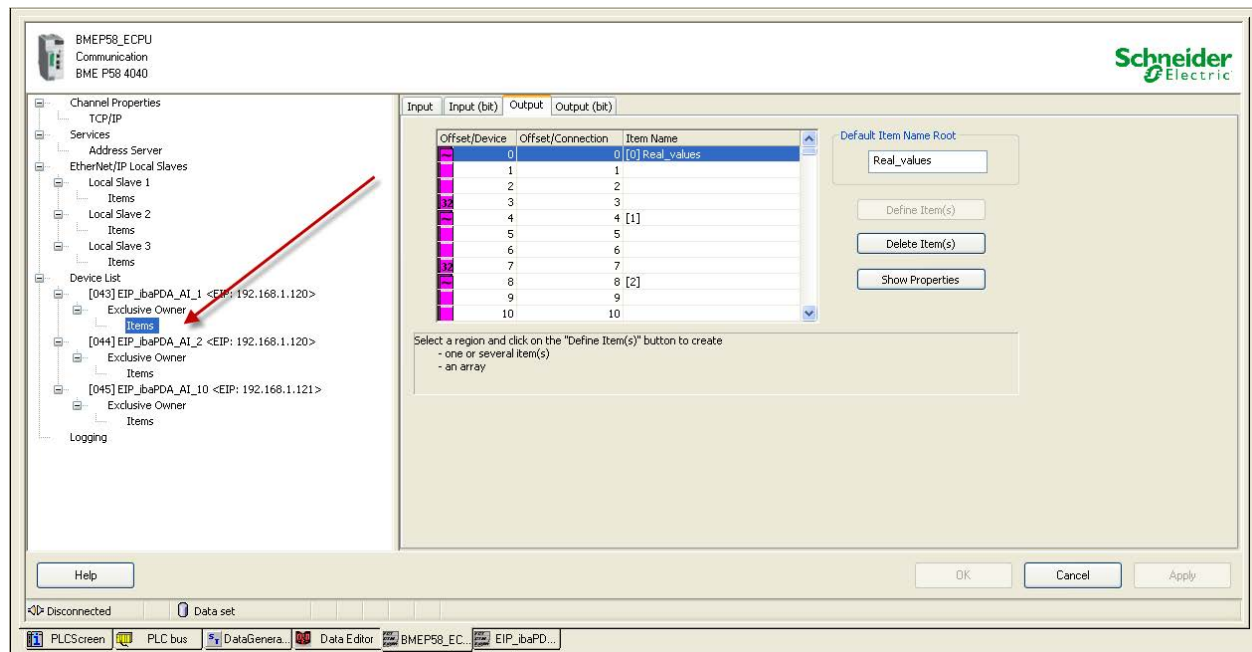


Für jede Verbindung:

- Stellen Sie den *Import Mode* auf "Manual" im Register *Properties*, so wie im Bild oben gezeigt. I/O-Elemente werden hinzugefügt, wenn das Geräte-DTM erstmalig in Unity Pro hinzugefügt wird. Danach werden alle I/O-Elemente manuell im Geräteeeditor erstellt.
- Im Register *Address Setting* tragen Sie die IP-Adresse des *ibaPDA*-Servers ein. Stellen Sie außerdem *DHCP for the device* auf "Disabled".

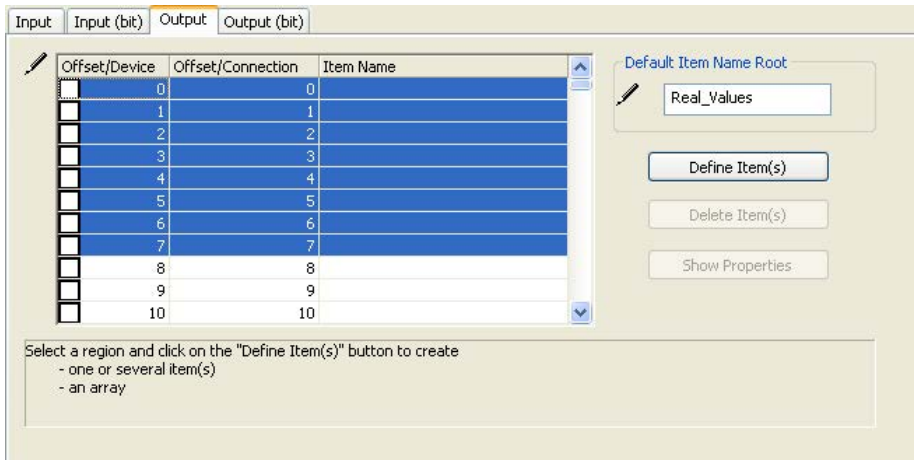
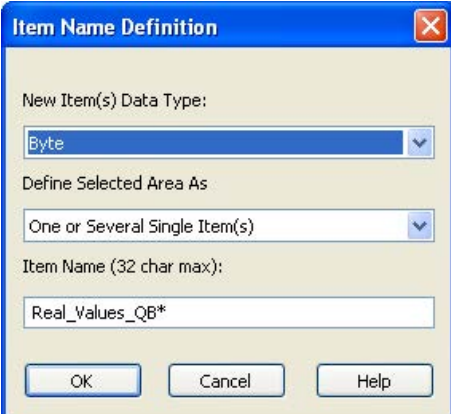


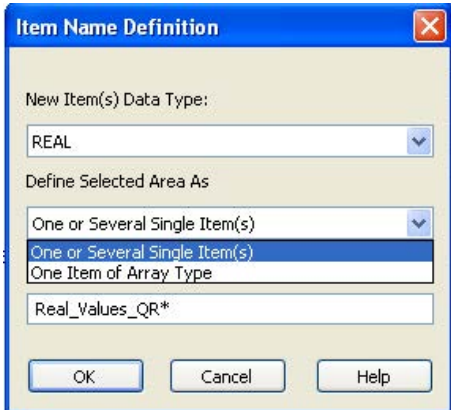
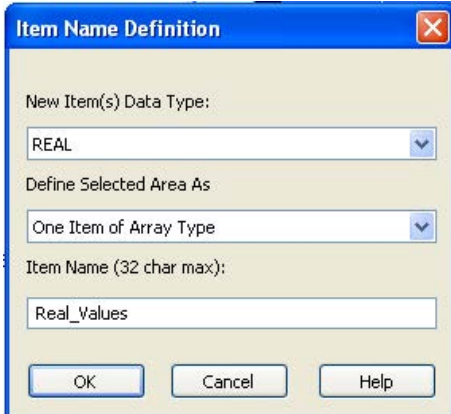
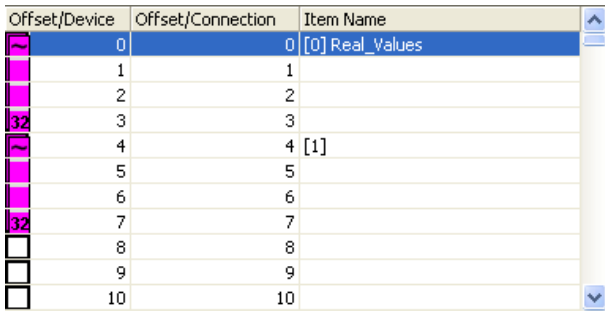
10. Wählen Sie die Elemente "Exclusive Owner" aus.

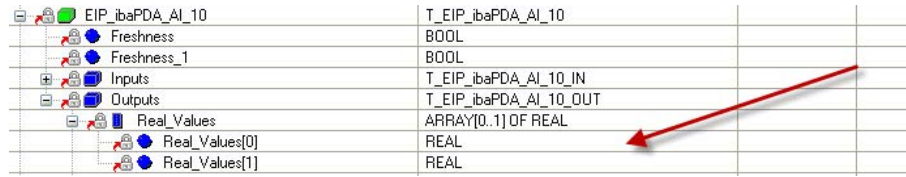


Hier können Sie die I/O-Elemente konfigurieren:

- Sie können Eingangs- und Ausgangselemente in Gruppen von 1 oder mehreren Einzelbits, 8-Bit Bytes, 16-Bit WORDs, 32-Bit DWORDs oder 32-Bit IEEE Gleitkommawerten konfigurieren. Die Anzahl der Elemente, die Sie erzeugen, hängt vom Datentyp und der Größe jedes einzelnen Elements ab.
- Wählen Sie das Register *Input* oder *Output*, je nachdem wo Sie I/O-Elemente erzeugen wollen.
- Das folgende Beispiel zeigt die nötigen Schritte, um Ausgangselemente für das Senden von Daten an *ibaPDA* zu deklarieren:

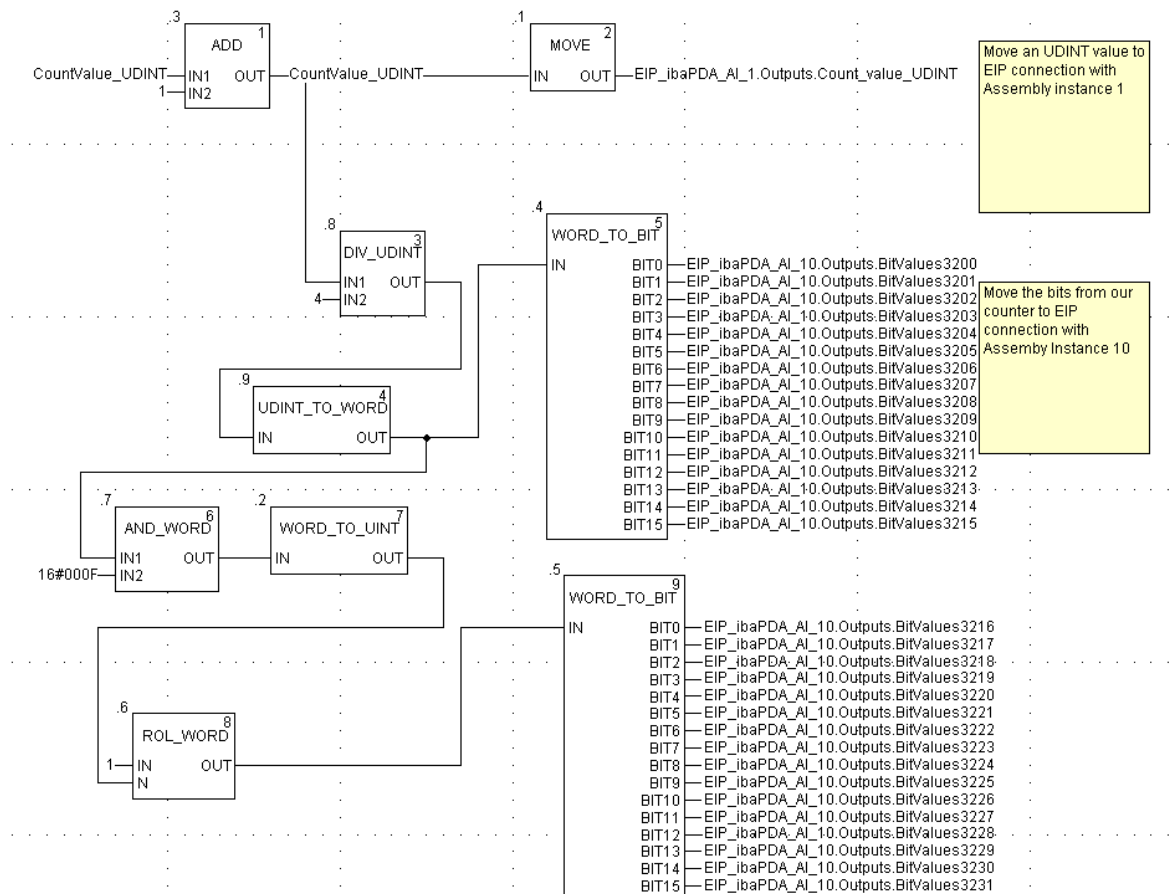
Schritt	Aktion
1	<p>Klicken Sie auf das Register <i>Output</i>, um die folgende Seite zu öffnen:</p>  <p>Hinweis: In diesem Beispiel stellt jede Zeile ein Byte dar. Je nach Datentyp des Elements, das Sie erzeugen wollen, wählen Sie mehrere Zeilen aus.</p> <p>Wählen Sie, beginnend am Anfang der Tabelle, die ersten n Zeilen aus:</p> <p>Wenn Sie z. B. 2 REAL-Werte erzeugen wollen, wählen Sie genau 8 Zeilen aus, wie oben gezeigt.</p>
2	<p>Im Eingabefeld <i>Default Item Name Root</i> tragen Sie den Root-Namen des Elements ein, z. B. "Real_Values".</p>
3	<p>Klicken Sie auf den Button <Define Item(s)>.</p> <p>Ergebnis: Der Dialog <i>Item Name Definition</i> öffnet sich:</p> 

Schritt	Aktion
4	<p>Wählen Sie "REAL" als Datentyp des/der neuen Elements/Elemente.</p> <div></div> <p>Da oben ein Vielfaches von 4 Bytes ausgewählt wurde, besteht nun die Wahl, die Elemente einzeln oder als Array zu handhaben. Hier sollen die Werte als Array verwendet werden. Der Name des Elements wird mit "Real_Values" eingetragen.</p> <div></div> <p>Klicken Sie auf <OK> und als Ergebnis sind 2 neue Elemente erzeugt worden:</p> <div></div>
5	<p>Klicken Sie auf <OK>, um das Elemente-Fenster zu schließen.</p>

Schritt	Aktion
6	<p>Wählen Sie Build->Analyze, um Ihre Bearbeitung zu sichern und die Device DDT-Variablen zu aktualisieren.</p> 
	<p>Wiederholen Sie die Schritte 1 - 6 im Beispiel, um alle I/O-Elemente für Ausgangs- und Eingabevariablen sowie für Eingangs- und Ausgangsbits zu definieren.</p>

Verwenden Sie die oben erzeugten I/O-Elemente in Ihrem Programm, um Daten an *ibaPDA* zu senden:

Das Beispiel unten zeigt Code-Muster in FBD und Structured Text.



```
FOR i:= 0 TO 99 BY 1 DO

  i_real := udint_to_real(i);

  (* generate some Ramp signals on EIP connection with assembly ID 1 *)
  (*-----*)
  testRealValuesRampSignal[i]:= testRealValuesRampSignal[i]+ (i_real*100.0);
  (* limit the ramps by 100000000*)

  if (testRealValuesRampSignal[i] > 100000000.0) then
    testRealValuesRampSignal[i] := 0.0;
  end_if;

  EIP_ibaPDA_AI_1.Outputs.Real_values[i]:= testRealValuesRampSignal[i];

  (* generate some Sine wave signals on EIP connection with assembly ID 2*)
  (*-----*)

  testRealValuesSineSignal[i] := i_real * SIN ( 2.0 * pi_constant * (i_real / 10.0) * elapsed_time_in_sec_for_calc);

  EIP_ibaPDA_AI_2.Outputs.Real_values[i]:= testRealValuesSineSignal[i];

  (* generate some Cosine wave signals on EIP connection with assembly ID 10 *)
  (*-----*)

  testRealValuesCosineSignal[i] := i_real * COS (2.0 * pi_constant * (i_real / 10.0) * elapsed_time_in_sec_for_calc);

  EIP_ibaPDA_AI_10.Outputs.Real_values[i]:= testRealValuesCosineSignal[i];

END_FOR ;
```

Andere Dokumentation



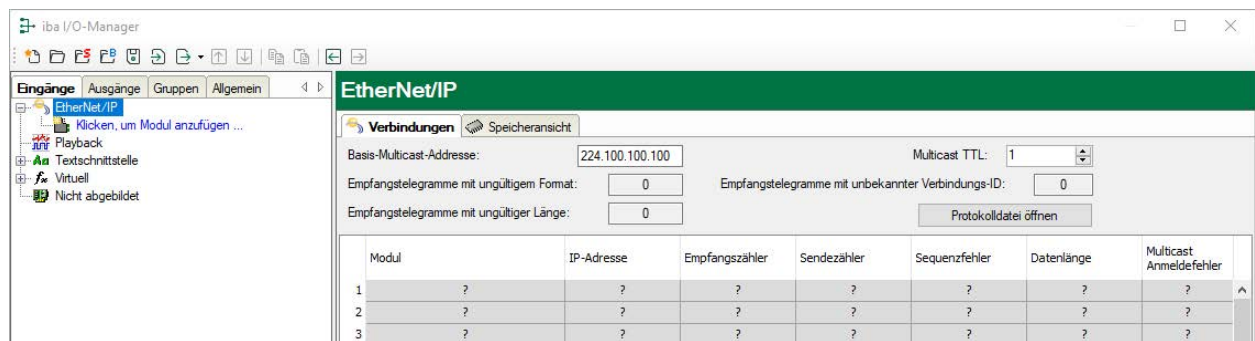
Bitte nutzen Sie für weitere Informationen die Schneider Electric Unity Pro XL Dokumentation.

3.5 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 *EtherNet/IP* an.

3.5.1 Allgemeine Einstellungen der Schnittstelle

Die Schnittstelle bietet folgende Funktionen und Konfigurationsmöglichkeiten:



Basis-Multicast-Adresse

Diese Einstellung ist nur für Multicast-Kommunikation relevant. Diese Basis-Multicast-Adresse wird als Zieladresse genutzt, an die *ibaPDA* seine Antwort schickt. Der hintere Teil der Adresse wird automatisch durch die Assembly-Instanznummer ersetzt. Im Allgemeinen können Sie die Default-Adresse behalten, die durch *ibaPDA* vergeben wird. Dies gilt nur nicht für den Fall, dass diese Adresse bereits benutzt wird oder Änderungen durch Router- oder Firewall-Konfigurierung erforderlich sind.

Multicast TTL

Der Parameter TTL (Time-to-Live) ist als Default auf 1 gesetzt. Jeder Router zwischen SPS und *ibaPDA* setzt den TTL-Wert um 1 nach unten, wenn ein Multicast-Datenpaket ankommt. Der Router lehnt ein Paket ab, wenn der TTL-Wert 0 erreicht. Sie müssen den Wert nur größer 1 setzen, wenn eine SPS an einem oder mehreren Routern hängt.

Sie können für diese Verbindungen auf einfache Weise Module anlegen. Klicken Sie dazu mit der rechten Maustaste auf das EtherNet/IP -Schnittstellen-Icon in der Baumansicht und wählen Sie im Kontextmenü *Autom. Erkennung*. Alternativ dazu können Sie diese Module auch händisch anlegen, ohne dass eine Verbindung existiert. Klicken Sie dazu auf das Item im Baum oder das blaue *Klicken, um Modul anzufügen...*

Die Nummer der Verbindung gleicht der konfigurierten Assembly-Instanz für ForwardOpen. Wenn Sie den Cursor über eine IP-Verbindung in der Baumansicht bewegen, sehen Sie einige zusätzliche Informationen in einem Tool-Tip-Fenster, wie z. B. die Verbindungslänge und IP-Adressen.

Telegramm- und Fehlerzähler

Die Zähler für empfangene und gesendete Telegramme sowie die Fehlerzähler dienen nur zur Anzeige.

<Protokolldatei öffnen>

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

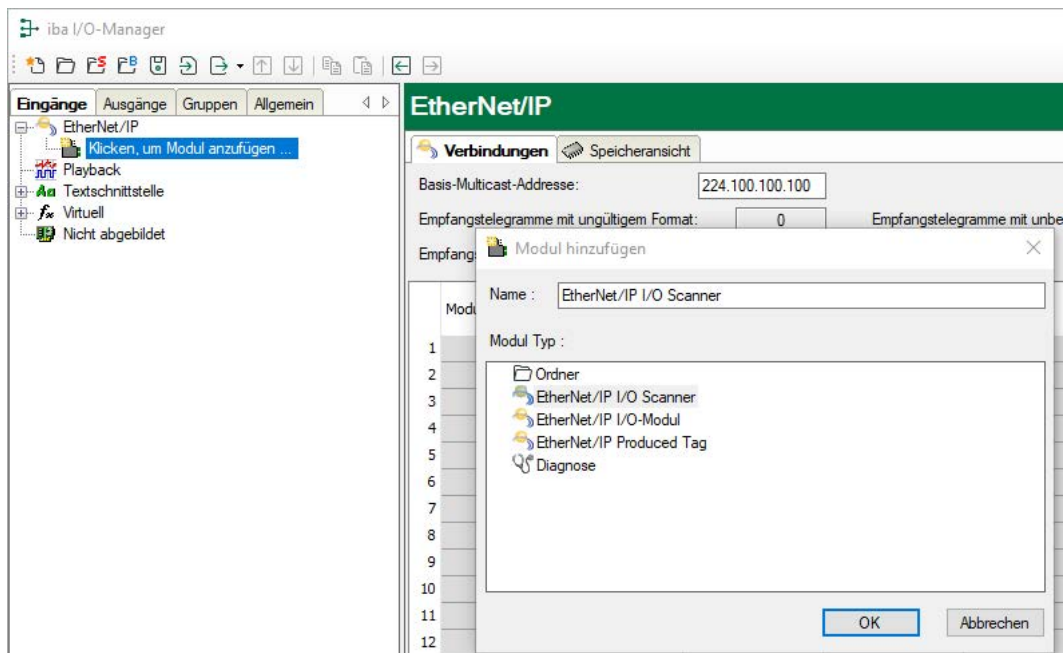
Der Dateiname der aktuellen Protokolldatei lautet [SchnittstelleLog.txt](#), der Name der archivierten Protokolldateien lautet [SchnittstelleLog_yyyy_mm_dd_hh_mm_ss.txt](#).

Weitere Informationen siehe [↗ Verbindungstabelle](#), Seite 46

3.5.2 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>.



Modultypen

Folgende Modultypen können Sie zur Schnittstelle hinzufügen:

EtherNet/IP I/O Scanner

EtherNet/IP I/O-Modul

EtherNet/IP I/O Produced Tag

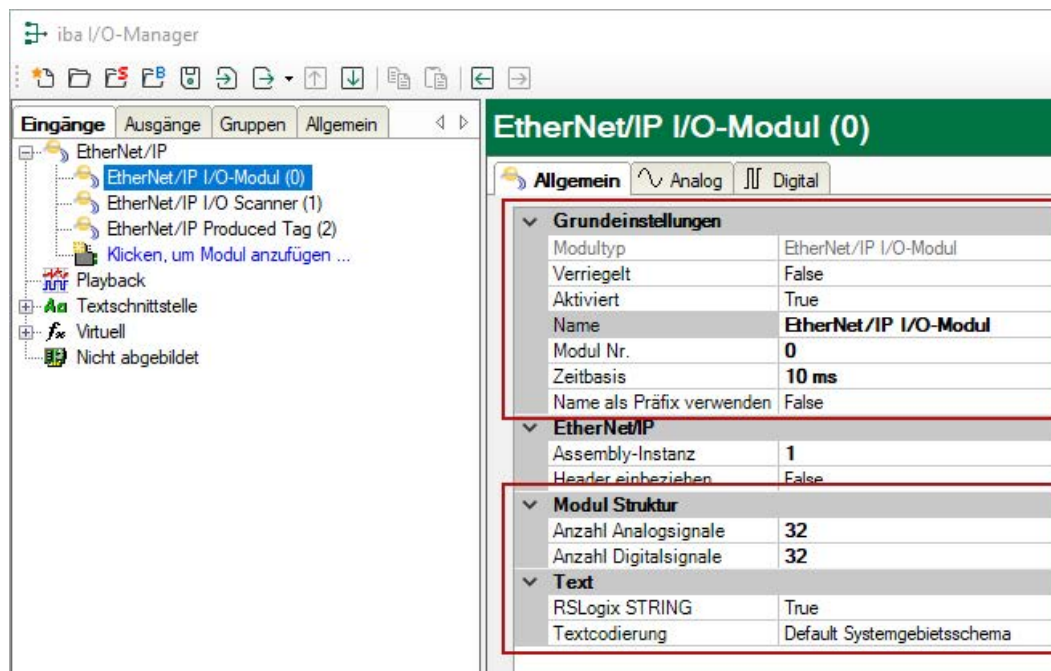
Für Informationen zu den EtherNet/IP-Modultypen siehe ➔ *ibaPDA-Modultypen*, Seite 10

Für Informationen zum Modultyp Diagnose siehe ➔ *Diagnosemodule*, Seite 50

3.5.3 Allgemeine Moduleinstellungen

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

Alle Module haben die folgenden Einstellmöglichkeiten.



Die rot umrandeten Einstellungen sind für alle 3 Modultypen gleich.

Grundeinstellungen

Modultyp (nur Anzeige)

Zeigt den Typ des aktuellen Moduls an.

Verriegelt

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

Aktiviert

Aktivieren Sie das Modul, um Signale aufzuzeichnen.

Name

Hier können Sie einen Namen für das Modul eintragen.

Modul Nr.

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

Zeitbasis

Alle Signale dieses Moduls werden mit dieser Zeitbasis erfasst.

Name als Präfix verwenden

Diese Option stellt den Modulnamen den Signalnamen voran.

Modulstruktur

Anzahl der Analogsignale/Digitalsignale

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

Text

RSLogix STRING

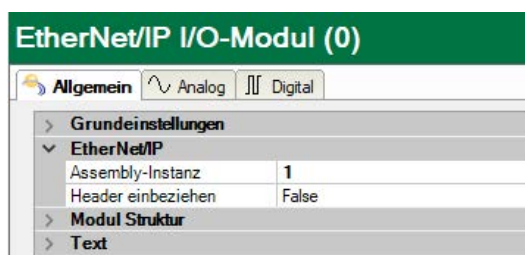
Aktivieren Sie diese Option ("True"), wenn Signale vom Datentyp STRING auf RSLogix-Strings verweisen.

Textcodierung

Für eine korrekte Interpretation und Anzeige der empfangenen Textdaten können Sie hier die Form der Textcodierung bzw. die Codepage auswählen. Zur Auswahl stehen:

- Default Systemgebietsschema: Codierung gem. der Windows-Systemeinstellungen
- Western European (1252): 8-Bit-Kodierung inkl. ASCII, westeurop. Umlaute, Sonderzeichen
- UTF-8: Unicode, für universelle Zeichen inkl. ASCII, asiatische Sprachen usw.

Modultyp EtherNet/IP I/O-Modul



Assembly-Instanz

Hier müssen Sie die Nummer der Assembly-Instanz eintragen, wie sie in der SPS für diese Verbindung projiziert wurde.

Hinweis



Für die Assembly-Instanznummern gibt es einige Einschränkungen hinsichtlich der EtherNet/IP-Schnittstellenlizenzen.

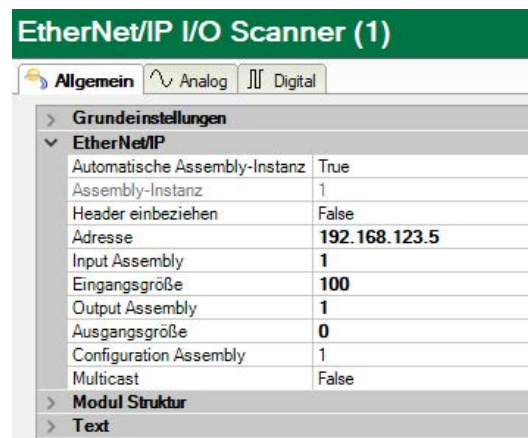
Die Assembly-Instanznummer muss in folgenden Bereichen liegen...

- 1 bis 64 für die erste Lizenz (Basislizenz *ibaPDA-Interface-EtherNet/IP*),
- 65 bis 128 für die zweite (1. *one-step-up-Interface-EtherNet/IP-Lizenz*),
- 129 bis 192 für die dritte (2. *one-step-up-Interface-EtherNet/IP-Lizenz*),
- 193 bis 255 für die vierte (3. *one-step-up-Interface-EtherNet/IP-Lizenz*).

Header einbeziehen

Wenn diese Option aktiviert ist ("True"), dann werden nicht nur die "reinen" Daten gemessen. *ibaPDA* hat auch Zugriff auf die Header-Informationen des UDP-CIP-Pakets. Dies ist nur sinnvoll für Debugging-Zwecke.

Modultyp EtherNet/IP I/O-Scanner



Automatische Assembly -Instanz

Setzen Sie diese Einstellung auf "True", um diesem Modul automatisch eine freie Assembly-Instanz durch *ibaPDA* zuweisen zu lassen. Setzen Sie diese Einstellung auf "False", wenn Sie die Assembly-Instanz manuell zuweisen wollen. In letzterem Fall stellen Sie sicher, dass die Assembly-Instanz nicht von einem anderen Modul verwendet wird.

Assembly-Instanz

Für dieses Modul nicht relevant (nur Anzeige).

Header einbeziehen

Wenn diese Option aktiviert ist („True“), dann werden nicht nur die "reinen" Daten gemessen. *ibaPDA* hat auch Zugriff auf die Header-Informationen des UDP-CIP-Pakets. Dies ist nur sinnvoll für Debugging-Zwecke.

Adresse

Geben Sie hier die IP-Adresse des Geräts ein, mit dem dieses Modul verbunden sein soll.

Input Assembly

Geben Sie hier die Input Assembly-Instanznummer des Geräts ein, von dem Sie Daten lesen wollen.

Eingangsgröße

Geben Sie hier die Größe (in Bytes) der gesamten Input Assembly-Instanz ein.

Output Assembly

Geben Sie hier die Output Assembly-Instanznummer des Geräts ein, in das Sie Daten schreiben wollen.

Ausgangsgröße

Geben Sie hier die Größe (in Bytes) der gesamten Output Assembly-Instanz ein.

Configuration Assembly

Geben Sie hier die Konfigurations-Assembly-Instanznummer des Geräts ein. Wenn Sie diese nicht kennen, dann lassen Sie den Wert bei 1.

Multicast

Wenn Sie diese Option auf "True" stellen, dann werden die Eingangsdaten via Multicast statt Unicast empfangen.

Hinweis



Falls ein I/O-Adapter oder Gerät Eingänge und Ausgänge hat, Sie aber nur die Eingänge erfassen wollen, dann sollten Sie eine spezielle Assembly-Nummer in das Feld *Output Assembly* eintragen. Diese Assembly-Nummer – meist als „Input only assembly“ bezeichnet – sollte in der Herstellerdokumentation vermerkt sein. Für WAGO, z. B., lautet sie „198“. Als *Ausgangsgröße* tragen Sie dann „0“ ein.

Modultyp EtherNet/IP Produced Tag

EtherNet/IP Produced Tag (2)	
Allgemein Analog Digital	
Grundeinstellungen	
EtherNet/IP	
Automatische Assembly-Instanz	True
Assembly-Instanz	1
Header einbeziehen	False
Adresse	192.168.123.5
Routing verwenden	False
Slot-Nr.	0
Produced Tag	
Datengröße	100
Modul Struktur	
Text	

Automatische Assembly-Instanz

Setzen Sie diese Einstellung auf "True", um diesem Modul automatisch eine freie Assembly-Instanz durch *ibaPDA* zuweisen zu lassen. Setzen Sie diese Einstellung auf "False", wenn Sie die Assembly-Instanz manuell zuweisen wollen. In letzterem Fall stellen Sie sicher, dass die Assembly-Instanz nicht von einem anderen Modul verwendet wird.

Assembly-Instanz

Für dieses Modul nicht relevant (nur Anzeige)

Header einbeziehen

Wenn diese Option aktiviert ist („True“), dann werden nicht nur die "reinen" Daten gemessen. *ibaPDA* hat auch Zugriff auf die Header-Informationen des UDP-CIP-Pakets. Dies ist nur sinnvoll für Debugging-Zwecke.

Adresse

Geben Sie hier die IP-Adresse der CPU ein, mit der dieses Modul verbunden sein soll.

Routing verwenden

Aktivieren Sie diese Option, wenn die Zielsteuerung nicht direkt von *ibaPDA* erreichbar ist. Der Zugriff kann über wechselnde Bussysteme realisiert werden. Wenn die Zielsteuerung direkt von *ibaPDA* erreicht werden kann, deaktivieren Sie diese Option (=False).

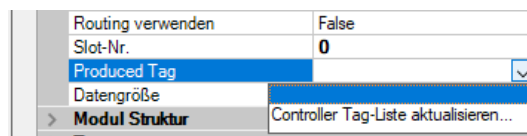
Slot-Nr./Pfad

Wenn Routing deaktiviert ist, geben Sie hier die Slot-Nummer (Steckplatznummer) der CPU ein, mit der dieses Modul verbunden sein soll. Wenn Routing aktiviert ist, dann tragen Sie hier den Verbindungspfad ein, der beschreibt, wie die CPU erreicht werden kann. Er besteht aus mehreren Segmenten, die durch Komma getrennt sind. Jedes Segment besteht aus 2 Teilen, die auch durch Komma getrennt sind: Der Port und die Zieladresse auf dem Port. Der Port kann Backplane, A, B oder eine Nummer sein. Die Zieladresse kann ein Rückwandslot, eine DH+ Adresse, eine ControlNet-Adresse oder eine IP-Adresse sein.

Beispiel: Backplane,1,A,192.168.200.154,Backplane,0

Produced Tag

Nachdem Sie Adresse und Slot-Nummer eingetragen haben, können Sie die Controller-Tags von der CPU laden, indem Sie hier *Controller Tag-Liste aktualisieren...* aus der Drop-down-Liste wählen.



Wenn Sie dies tun, dann verbindet sich *ibaPDA* mit der CPU und liest die Controller-Tags mit ihren Strukturen aus und trägt sie in die Drop-down-Liste ein. Sie können dann einen Tag auswählen. Nachdem Sie einen Tag ausgewählt haben, wird die Größe des Tags automatisch im Feld *Datengröße* eingetragen.

Des Weiteren werden Sie gefragt, ob *ibaPDA* die Analog- und Digitalsignale aktualisieren soll, so dass sie der Struktur des Tags entsprechen. Wenn Sie *Ja* wählen, wird die Anzahl der Analog- und Digitalsignale angepasst.

Datengröße

Datengröße des ausgewählten Tags. Wird automatisch nach Auswahl des Tags eingetragen.

3.5.4 Signalkonfiguration

Grundsätzlich ist das Prinzip der Signalkonfiguration für alle Module gleich.

In den Registern *Analog* und *Digital* jedes Moduls im I/O-Manager können Sie den Signalen Namen, Kommentar und – wenn gewünscht – Skalierungsfaktor zuweisen sowie die Signale aktivieren oder deaktivieren. Die Eingabe von Adresse und Datentyp ist essentiell.

Nur beim Modultyp *Produced Tag* werden die analogen und digitalen Messsignale automatisch eingetragen, wenn Sie nach dem Laden der Controller-Tag-Liste die Frage nach Aktualisierung der Analog- und Digitalsignale mit *Ja* beantworten, siehe ➔ *Allgemeine Moduleinstellungen*, Seite 31

Register Analog und Digital

Name	Einheit	Gain	Offset	Adresse	Datentyp	Aktiv	Istwert
0 TestStructArrayTag[0].First		1	0	0	FLOAT	<input checked="" type="checkbox"/>	0
1 TestStructArrayTag[0].Second		1	0	4	FLOAT	<input checked="" type="checkbox"/>	0
2 TestStructArrayTag[0].Third		1	0	8	INT	<input checked="" type="checkbox"/>	0
3 TestStructArrayTag[0].FourthNonAligned		1	0	10	DINT	<input checked="" type="checkbox"/>	0
4 TestStructArrayTag[0].dint_tussen_bits		1	0	14	DINT	<input checked="" type="checkbox"/>	0
5 TestStructArrayTag[1].First		1	0	18	FLOAT	<input checked="" type="checkbox"/>	0
6 TestStructArrayTag[1].Second		1	0	22	FLOAT	<input checked="" type="checkbox"/>	0
7 TestStructArrayTag[1].Third		1	0	26	INT	<input checked="" type="checkbox"/>	0
8 TestStructArrayTag[1].FourthNonAligned		1	0	28	DINT	<input checked="" type="checkbox"/>	0
9 TestStructArrayTag[1].dint_tussen_bits		1	0	32	DINT	<input checked="" type="checkbox"/>	0

Adresse

Der Adressraum hängt vom Datentyp ab. Nach dem Ändern von Datentypen kann so eine Anpassung 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 aus der Drop-Down-Liste aus.

Die folgenden Datentypen stehen zur Verfügung:

ibaPDA Datentyp	ControlLogix Datentyp	Unity PRO Datentyp
BYTE / SINT (8 Bit)	SINT	Byte
INT / WORD (16 Bit)	INT	INT / WORD / UINT
DINT / DWORD (32 Bit)	DINT	DINT / DWORD / UDINT
FLOAT (32 Bit)	REAL	REAL
DOUBLE ¹⁾ (64 bit)	n.a	n.a
STRING (32 Bit)		

¹⁾ Es gibt andere SPS-Systeme auf dem Markt, z. B. OMRON, die DOUBLE-kompatible Datentypen wie LREAL oder LWORD unterstützen.

Hinweis



Bei einem Produced Tag-Modul können Signalnamen, Adressen und Datentypen automatisch aus der CPU geladen werden.

Bit-Nr. (nur Digitalsignale)

Mit den Digitalsignalen haben Sie die Möglichkeit, 32 Einzelbits aus einem Double Integer zu erhalten. Nach Eingabe der Adresse wird die Bit-Nummer automatisch um 1 von 0 bis 31 erhöht, dann wird die Adresse um 4 erhöht.

3.6 Konfiguration ibaPDA-Ausgangsmodule

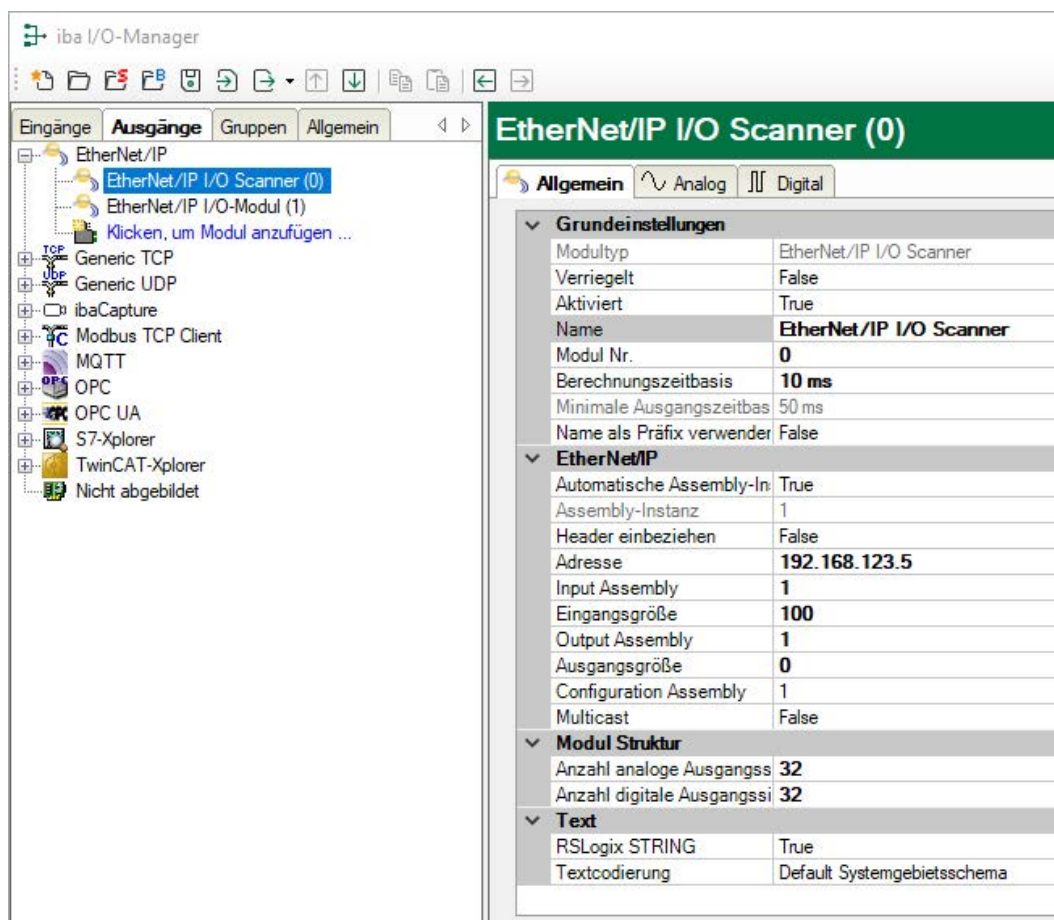
Wenn alle Systemvoraussetzungen erfüllt sind, bietet *ibaPDA* im Schnittstellenbaum des I/O-Managers im Register *Ausgänge* die Schnittstelle *EtherNet/IP* an. Ein manuelles Hinzufügen der Schnittstelle ist nicht erforderlich.

Ausgangsmodule fügen Sie auf die gleiche Weise hinzu wie Eingangsmodule.

Ausgangssignale stehen nur bei den Modultypen EtherNet/IP „I/O-Modul“ und „I/O-Scanner“ zur Verfügung.

3.6.1 Allgemeine Moduleinstellungen ibaPDA-Ausgangsmodule

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



Die Parameter sind nahezu identisch zu denen der Eingangsmodule, siehe ➤ *Allgemeine Moduleinstellungen*, Seite 31

Beachten Sie folgende Unterschiede zu den Einstellungen der Eingangsmodule:

Berechnungszeitbasis

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

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

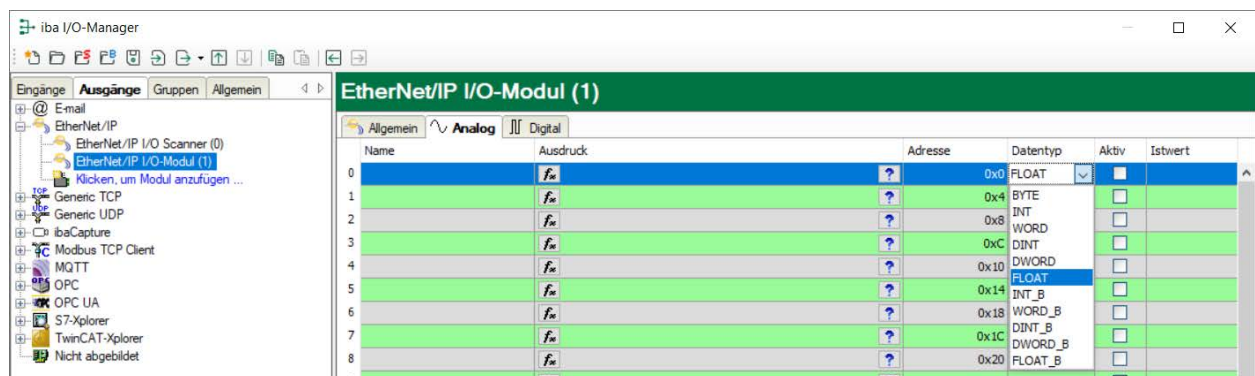
Minimale Ausgangszeitbasis

Zeitbasis, mit der die Ausgänge schnellstens aktualisiert werden können.

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

3.6.2 Signalkonfiguration

Sie können die Anzahl der analogen und digitalen Ausgangssignale bestimmen. Für jedes Ausgangssignal müssen Sie dessen Wert mithilfe eines Ausdrucks definieren sowie Adresse und Datentyp angeben.



Hinweis



ibaPDA verarbeitet Ausgangssignale im Vergleich zu den Eingangssignalen mit einer niedrigeren Priorität und einem Aktualisierungszyklus von mindestens 50 ms, abhängig von der I/O-Konfiguration.

3.7 Multiple IP-Adressen für ibaPDA anlegen

Hinweis



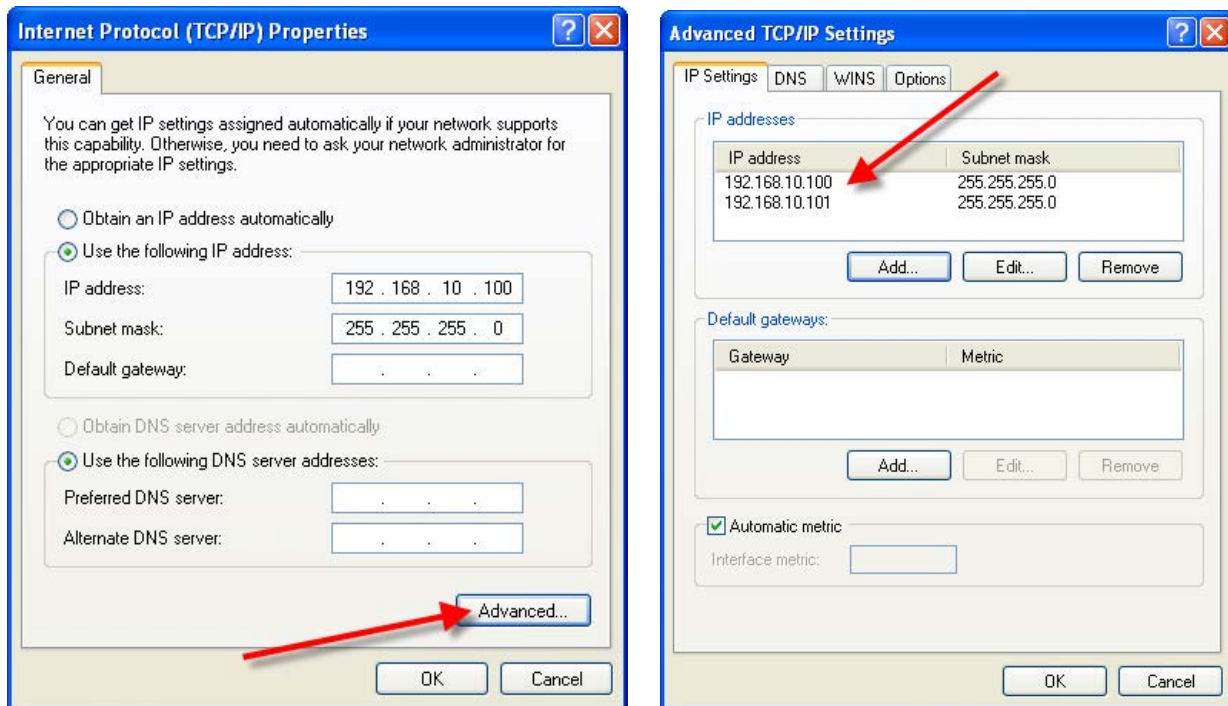
Nur relevant für Verbindungen mit Rockwell-SPSen.

ibaPDA arbeitet als Server, der nach Clients sucht, die eine Verbindung anfragen.

ibaPDA ist so konfiguriert worden, dass es an dem konfigurierten TCP/IP-Port alle EtherNet/IP-Verbindungsanfragen abhört. Die Rockwell/AB SPS ist somit der Master (Initiator) der Kommunikation. Das bedeutet, dass die SPS die Verbindung herstellt und *ibaPDA* ein passiver Teilnehmer ist. Der Status einer jeden EtherNet/IP-Verbindung wird in der Übersicht der Schnittstelle im *ibaPDA* I/O-Manager angezeigt.

Es ist wichtig, zu berücksichtigen, dass die IP-Adressen nur einmal vergeben werden sollten. Normalerweise kann es allerdings mehr als einen Link von derselben Ziel-IP-Adresse (SPS) geben. Da das Generic-Ethernet-Module nicht mehrere Verbindungen mit derselben Ziel-IP-Adresse herstellen kann, muss die Netzwerk-Schnittstellen-Karte (NIC) im *ibaPDA*-System für mehrere IP-Adressen auf derselben Schnittstellen-Karte konfiguriert werden. In Bezug auf jede einzelne SPS muss für jedes Generic-Ethernet-Module eine andere IP-Adresse definiert werden.

Mit dieser Methode werden einer Netzwerk-Schnittstellen-Karte mehrere IP-Adressen zugewiesen. Dies wird über die Windows® Systemsteuerung realisiert. Nachdem Sie den Dialog Eigenschaften an der LAN-Verbindung, die für EtherNet/IP verwendet wird, geöffnet haben, wählen (markieren) Sie Internet Protocol (TCP/IP) und klicken anschließend auf den Button <Eigenschaften>. Klicken Sie als nächstes auf den Button <Erweitert>. Sie werden eine Dialogbox sehen, wie Sie unten dargestellt ist. Ausgehend von hier, können Sie jeder Netzwerk-Schnittstellen-Karte (NIC) in Ihrem System weitere IP-Adressen zuweisen.



Jede EtherNet/IP-Verbindung sollte als *ibaPDA*-Modul angesehen werden, das eine eindeutige Modul-Nummer-ID benötigt. Diese ID wird automatisch von *ibaPDA* vergeben. Die Referenz für die Zuweisung des Moduls/der Verbindung ist die Assembly-Instanz. Die Assembly-Instanz wurde in den Konfigurierungsdialog des Generic-Ethernet-Module in RSLogix eingetragen.

Dieselbe Assembly-Instanznummer muss im I/O-Manager von *ibaPDA* in das Feld *Assembly-Instanz* des Registers *Allgemein* des betreffenden Moduls eingetragen werden.

4 Diagnose und Störungsbehebung

4.1 Wichtige Eigenschaften von Ethernet Switches für Ethernet/IP

Die sorgfältige Auswahl von Switches für Echtzeitanwendungen (I/O) in EtherNet/IP-Netzwerken ist entscheidend. Einige wichtige Eigenschaften gewährleisten eine funktionale Infrastruktur für Ihre Anwendung und sollten beachtet werden:

Erforderlich:

- Volle Duplex-Fähigkeit an allen Ports
- IGMP Snooping
- Port Mirroring

Hinweis



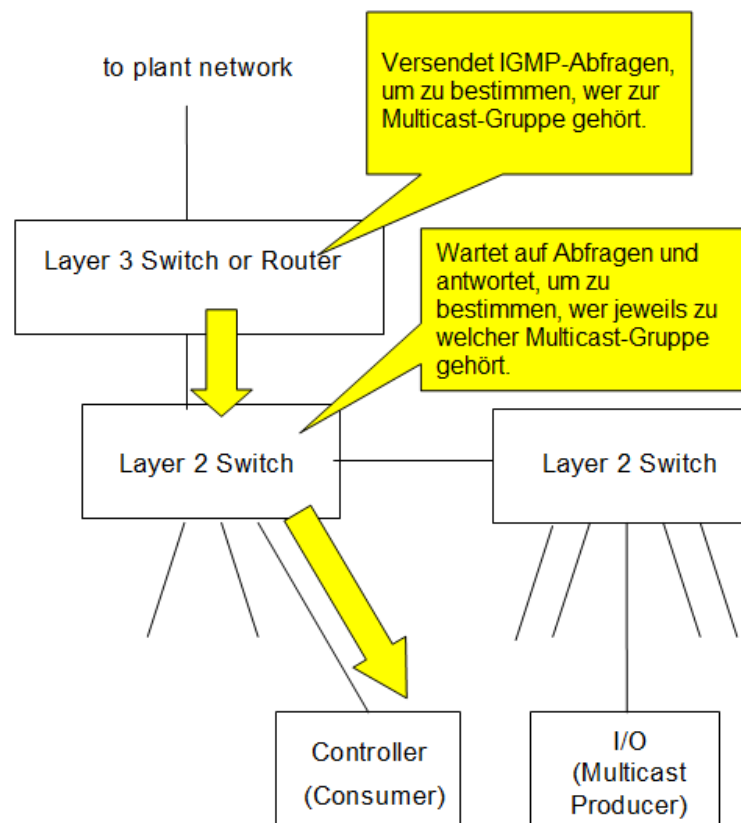
Die Auswahl des richtigen Switches ist besonders wichtig, wenn der EtherNet/IP-Controller IP-Multicast-Telegramme verwendet. Daher empfiehlt iba Uni-cast-Telegramme zu verwenden, wenn dies vom Controller unterstützt wird.

■ Volle Duplex-Fähigkeit an allen Ports

Volle Duplex-Fähigkeit verhindert Kollisionen auf der Leitung wegen der getrennten Sende- und Empfangskanäle für jedes Gerät. In Kombination mit der Geschwindigkeit heutiger Switches können Verzögerungen durch Kollisionen oder hoher Netzwerklast vernachlässigt werden. Folglich können Sie damit ein hoch deterministisches Verhalten mit einem EtherNet/IP-Netzwerk erreichen und für I/O-Steuerung verwenden.

■ IGMP Snooping

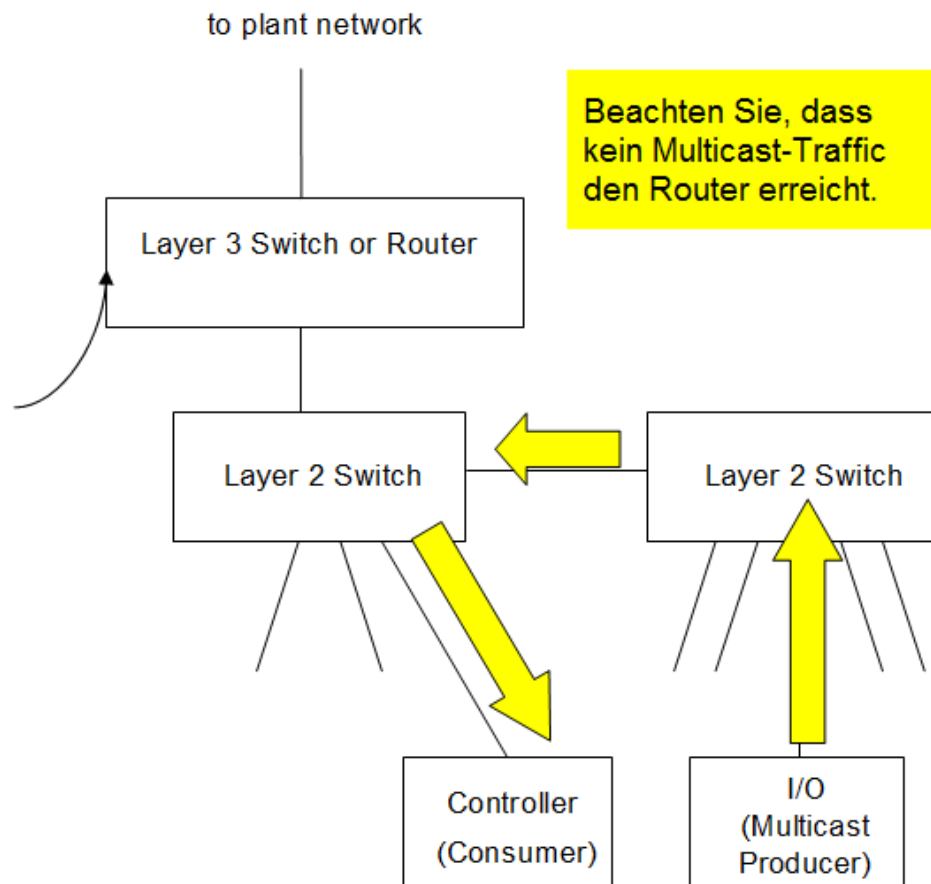
IGMP Snooping begrenzt einen Überlauf durch Multicast-Datenverkehr mit einer dynamischen Konfiguration der Switch-Ports, so dass der Multicast-Datenverkehr nur auf die Ports gelenkt wird, die zu einer IP-Multicast-Gruppe gehören.



Normalerweise benötigt ein Schicht 2-Switch, der IGM Snooping unterstützt, einen Router (der ein Schicht 3-Switch sein kann), um die IGMP-Abfragen zu versenden, mit denen er lernt, welche Geräte Mitglieder der Multicast-Gruppe sind (IGMP Querier).

Hinweis

Einige industrielle Schicht 2-Switches unterstützen IGMP Snooping ohne die Notwendigkeit eines Routers oder Schicht 3-Switches, um die IGMP-Abfragen zu versenden.

**■ Port Mirroring:**

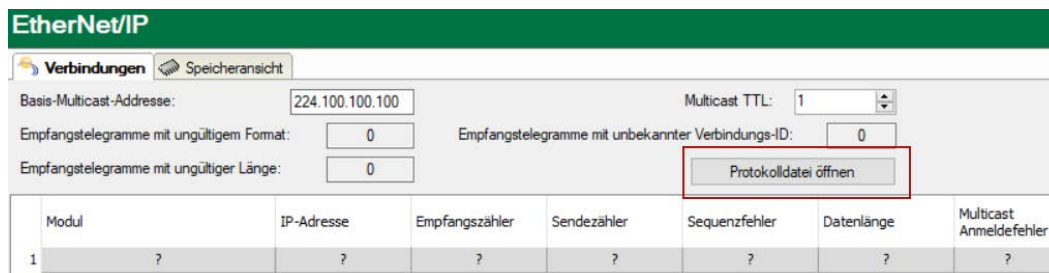
Port Mirroring bezeichnet die Fähigkeit, ein Duplikat der von einem Port übertragenen Pakete (Frames) direkt auf einen anderen Port zu leiten. Dadurch ist es einem Traffic Analyzer möglich, sich mit einem Switch zu verbinden und den Traffic an einem bestimmten Port zu überwachen. Ohne Port Mirroring kann ein Traffic Analyzer die Pakete an anderen Ports nicht erkennen.

Traffic Analyzers (wie die Freeware Wireshark) werden von vielen Personen genutzt, die mit Ethernet-Netzwerken arbeiten. Daher ist es sehr wichtig, dass ein Switch ausgewählt wird, der Port Mirroring unterstützt, damit der Traffic Analyzer korrekt funktioniert.

4.2 Konflikte mit anderen Ethernet/IP-bezogenen Programmen

Ein weit verbreitetes Problem von *ibaPDA*-Serversystemen, die die Ethernet/IP-Schnittstelle verwenden, ist ein Konflikt mit anderen Programmen, die ebenfalls den Ethernet/IP Port 44818 nutzen.

Wenn ein anderer Dienst, z. B. Rockwell RSLinx Service, startet bevor der *ibaPDA*-Server startet, dann horcht dieses weitere Gerät bereits am Port 44818 bevor der *ibaPDA*-Dienst die Möglichkeit dazu hat. Um diesen Umstand zu überprüfen, öffnen Sie die Ethernet/IP-spezifische Protokolldatei im I/O-Manager von *ibaPDA*, so wie unten gezeigt:



Die folgende Fehlermeldung würde in der Protokolldatei stehen:

```
2014-07-23 17:06:49.227 [14: EthernetIP listening TCP thread] : **** ERROR **** : Creating Ethernet/IP listening socket : Only one usage of each socket address (protocol/network address/port) is normally permitted
```

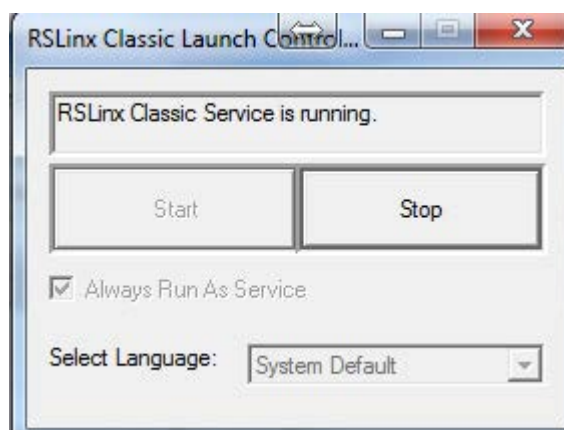
Hinweis



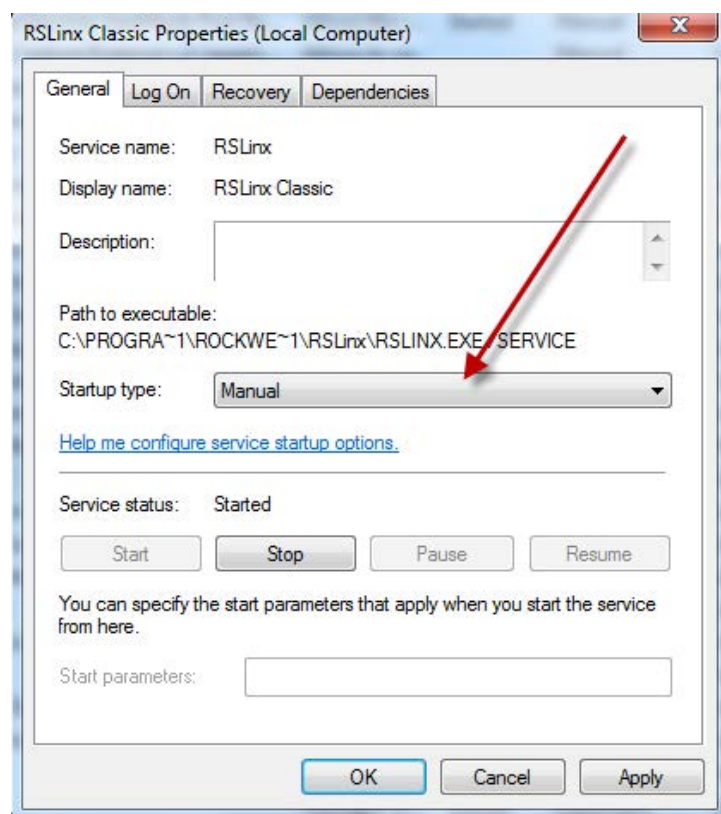
Ein Weg, den Verursacher zu finden, ist die Verwendung des Befehls **Netstat -b -a** in einem DOS-Fenster und nach dem Port 44818 zu suchen:

```
TCP      0.0.0.0:44818          note-car1o7:0          LISTENING
[RSLINX.EXE]
```

Wenn Sie diese Fehlermeldung sehen und RSLinx auf dem *ibaPDA*-Server installiert ist, dann deinstallieren Sie RSLinx oder stoppen Sie den RSLinx-Dienst über die „Launch“-Konsole:



Wenn RSLinx auf dem *ibaPDA*-Server wirklich gebraucht wird, dann wäre eine andere Lösung, den Starttyp des RSLinx-Dienstes auf „Manuell“ zu stellen.

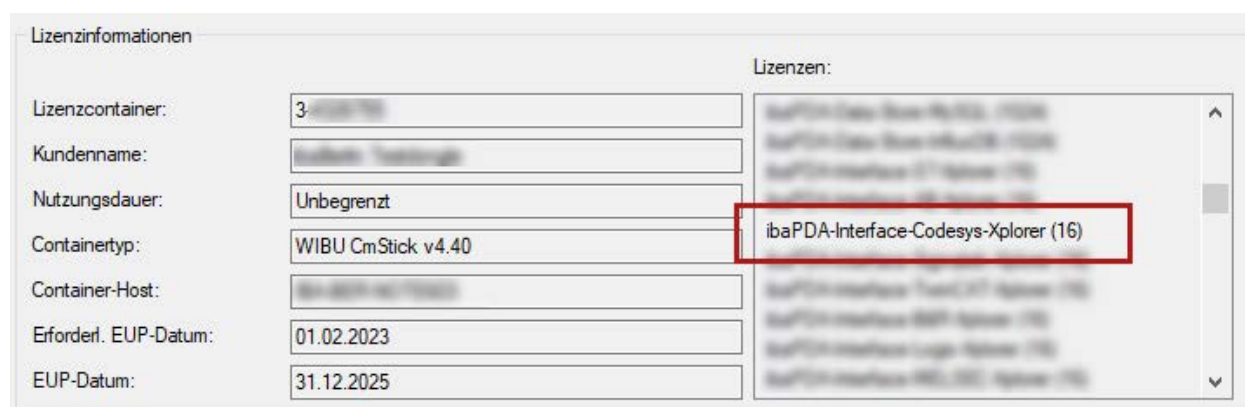


Dies verhindert beim Booten, dass RSLink startet bevor *ibaPDA*-Server startet.

4.3 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*.



4.4 Verbindungsprobleme mit Lizenzbezug

Hinweis



Für die Assembly-Instanznummern gibt es einige Einschränkungen hinsichtlich der EtherNet/IP-Schnittstellenlizenzen.

Die Assembly-Instanznummer muss in folgenden Bereichen liegen...

- 1 bis 64 für die erste Lizenz (Basislizenz *ibaPDA-Interface-EtherNet/IP*),
- 65 bis 128 für die zweite (1. *one-step-up-Interface-EtherNet/IP-Lizenz*),
- 129 bis 192 für die dritte (2. *one-step-up-Interface-EtherNet/IP-Lizenz*),
- 193 bis 255 für die vierte (3. *one-step-up-Interface-EtherNet/IP-Lizenz*).

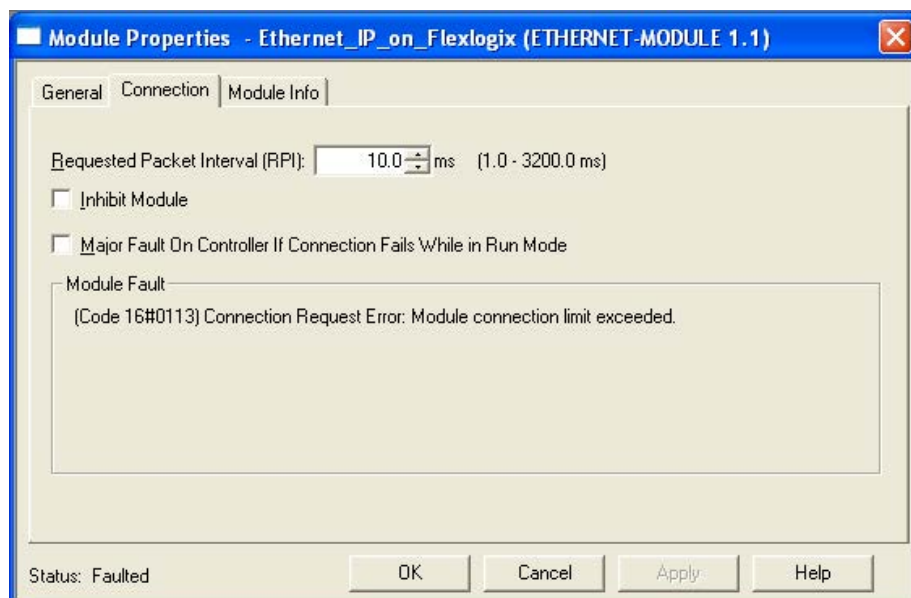
Wenn eine SPS versucht, sich mit einer Assembly-Instanz zu verbinden, deren Nummer höher als die Anzahl der lizenzierten Verbindungen ist, dann finden Sie beispielsweise folgende Meldungen in der Ethernet/IP-Protokolldatei:

```
2013-08-28 13:43:13.990 [14: EthernetIP listening TCP thread] : **** ER-
ROR **** : Forward_Open: Connection not established because driver didn't
allow connection (The parameter is incorrect.(0x00000057))
```

```
2013-08-28 13:43:13.990 [14: EthernetIP listening TCP thread] : **** ER-
ROR **** : Forward_Open: Connection not established, ConnectionInstance
could not be created or configured
```

In RSLogix 5000, z. B., sehen Sie einen Modulfehler, wenn die Assembly-Instanznummer nicht zur *ibaPDA*-Lizenz passt:


"Module connection limit exceeded."




4.5 Verbindungstabelle

Die Verbindungstabelle der EtherNet/IP-Schnittstelle bietet zusätzliche Informationen.

EtherNet/IP

 Verbindungen

 Speicheransicht

Basis-Multicast-Adresse:

224.100.100.100

Multicast TTL:

1

Empfangstelegramme mit ungültigem Format:

0

Empfangstelegramme mit unbekannter Verbindungs-ID:

0

Empfangstelegramme mit ungültiger Länge:

0

Protokolldatei öffnen

	Modul	IP-Adresse	Empfangszähler	Sendezähler	Sequenzfehler	Datenlänge	Multicast Anmeldefe..
1	Produced tag (0)	?	?	?	?	?	?
2	Other produced tag (4)	192.168.123.5	20631	20652	0	120	0
3	?	?	?	?	?	?	?
4	?	?	?	?	?	?	?
5	?	?	?	?	?	?	?
6	?	?	?	?	?	?	?
7	?	?	?	?	?	?	?
8	?	?	?	?	?	?	?
9	?	?	?	?	?	?	?
10	?	?	?	?	?	?	?
11	?	192.168.123.5	3694179	3694178	0	520	0
12	?	?	?	?	?	?	?
13	?	?	?	?	?	?	?
14	?	?	?	?	?	?	?
15	?	?	?	?	?	?	?
16	?	?	?	?	?	?	?

Jede Zeile entspricht einer Verbindung bzw. einem Modul und einer Assembly-Instanz.

Die Zeilennummer entspricht der Assembly-Instanz innerhalb von *ibaPDA*.

Für EtherNet/IP I/O-Module entspricht dies der Assembly-Instanz, wie sie im allgemeinen (generic) Ethernet-Modul der SPS I/O-Konfiguration projektiert wurde.

Für EtherNet/IP Produced Tag- und I/O-Scanner-Module wird diese automatisch erzeugt.

Die Spalte *Modul* zeigt, mit welchem Modul diese Verbindung korrespondiert.

Die Spalte *IP-Adresse* zeigt die IP-Adresse der verbundenen SPS.

Neben den Spalten für Empfangs- und Sendezähler sowie Sequenzfehler gibt es die Spalte *Datenlänge*. Die Spalte *Datenlänge* zeigt die Größe des UDP-Telegramms ohne UDP-Header. Die UDP-Daten enthalten einen EtherNet/IP-Header, der für Produced Tags 20 Bytes und für I/O-Verbindungen 24 Bytes lang ist.

Hinweis



Wenn Sie einen Doppelklick auf eine Zeile machen, gelangen Sie zu dem entsprechenden Offset im Register *Speicheransicht*.

Zusätzliche Informationen liefert die Hintergrundfarbe der Zeilen:

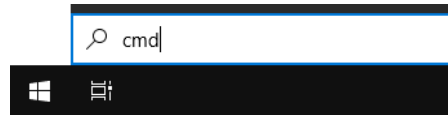
Farbe	Bedeutung
Grün	Modul für diese Assembly-Instanz ist definiert und die Verbindung ist OK.
Orange	Die Verbindung ist OK, trotzdem gibt es kein Modul, das für diese Assembly-Instanz definiert ist. Hinweis: Dies kann nur auftreten, wenn allgemeine (generic) Ethernet-Module in der I/O-Konfiguration der SPS definiert wurden. Wenn Sie auf der EtherNet/IP-Schnittstelle <i>Autom. Erkennung</i> ausführen, dann werden für die orangefarbenen Verbindungen I/O-Module hinzugefügt.
Rot	Modul für diese Assembly-Instanz ist definiert, aber die Verbindung zur SPS ist fehlgeschlagen.
Grau	Für diese Assembly-Instanz sind keine Verbindung und kein Modul definiert.

Tab. 4: Farbcode für Hintergrundfarben der Verbindungstabelle

4.6 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>
```

4.7 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)

4.8 Diagnosemodule

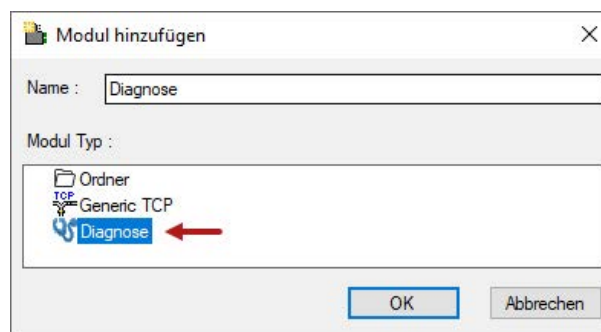
Diagnosemodule sind für die meisten Ethernet-basierten Schnittstellen und Xplorer-Schnittstellen verfügbar. Mit einem Diagnosemodul können Informationen aus den Diagnoseanzeigen (z. B. Diagnoseregister und Verbindungstabellen einer Schnittstelle) als Signale erfasst werden.

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
Paketgröße (aktuell)	Größe der aktuell empfangenen Telegramme
Paketgröße (max)	Größe des größten empfangenen Telegramms

Signalname	Bedeutung
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
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
Verlorene Images	Anzahl der verlorenen Images (in), die selbst nach einer Sendewiederholung nicht empfangen wurden
Verlorene Profile	Anzahl unvollständiger/fehlerhafter Profile

Signalname	Bedeutung
Zeilen (letzte)	Anzahl der Ergebniszeilen der letzten SQL-Abfrage (innerhalb der projektierten Anzahl von Ergebniszeilen)
Zeilen (Maximum)	Höchste Anzahl der Ergebniszeilen einer SQL-Abfrage seit dem letzten Start der Erfassung (maximal gleich der projektierten Anzahl von Ergebniszeilen)
Zeit zwischen Daten (aktuell/max/min)	Zeit zwischen zwei korrekt empfangenen Telegrammen Aktuell: zwischen den letzten zwei Telegrammen Max/min: statistische Werte seit Start der Erfassung oder Rücksetzen der Zähler
Zeit-Offset (Istwert)	Gemessene Zeitdifferenz der Synchronität zwischen dem ibaNet-E-Gerät und <i>ibaPDA</i>

5 Support und Kontakt

Support

Tel.: +49 911 97282-14
Fax: +49 911 97282-33
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
Fax: +49 911 97282-33
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.