

# **E/A Treiber HandBuch**

---

Version 11.3 - Ed. Feb. 2012

Cod. DOCS 11 DRV-D Build 1101

# Inhaltsverzeichnis

<b>1. EINFÜHRUNG</b> .....	<b>3</b>
1.1. DER TREIBER IM PROJEKT .....	3
1.2. SPEICHERBEREICHE.....	5
1.3. ADRESSIERUNG ÜBER DYNAMISCHE VARIABLEN .....	6
1.4. STATISCHE TASKS .....	7
1.5. INSTALLATION VON TREIBERN .....	8
1.6. TREIBER-VERDOPPLUNG .....	9
1.7. TREIBERNAUSFÜHRUNG.....	11
1.8. HARDWARE-FEHLER RS232 .....	11
<b>2. ALLGEMEINE EINSTELLUNGEN</b> .....	<b>13</b>
2.1. ALLGEMEINE (TREIBER).....	13
2.2. DEBUG .....	15
<b>3. EINSTELLUNGEN DER STATIONS</b> .....	<b>17</b>
3.1. STATIONEN .....	17
3.2. ALLGEMEINEN (STATIONEN).....	18
3.3. EINSTELLUNGEN DES SERIELLEN PORTS.....	19
3.4. PUFFER-GRÖÖE .....	20
3.5. TIMEOUTS .....	20
3.6. EINSTELLUNGEN DES BRIDGING-DIENSTES.....	21
3.7. TAPI - EINSTELLUNGEN .....	22
3.8. TCP/IP - EINSTELLUNGEN .....	24
3.9. RAS - EINSTELLUNGEN .....	25
3.10. SONDERKONFIGURATIONEN (TAPI UND RAS) .....	27
<b>4. EINSTELLUNGEN VON TASKS</b> .....	<b>29</b>
4.1. TASKS.....	29
4.2. ALLGEMEINE EIGENSCHAFTEN DER STATISCHEN TASKS .....	29
4.3. DYNAMISCHE TASKS.....	32
<b>5. IMPORTIERUNG DER DATENBANK DES GERÄTS</b> .....	<b>37</b>
5.1. IMPORT DER DATENBANK DES GERÄTS .....	37
<b>6. FEHLER</b> .....	<b>39</b>
6.1. BESCHREIBUNG DER FEHLER.....	39
6.2. ÜBER.....	45

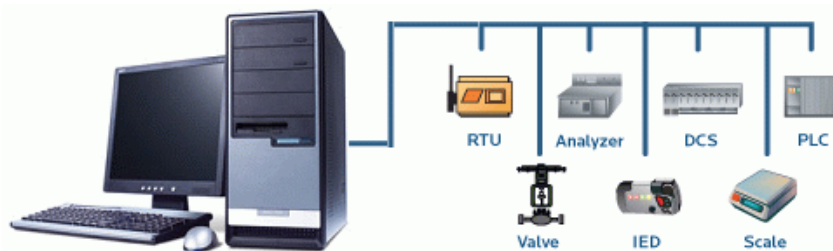
# 1. Einführung

---

Die in diesem Abschnitt behandelten Themen betreffen die Einstellungen, die für die Kommunikationstreiber von Supervisor durchgeführt werden müssen, welche in der Ressource Echtzeitdatenbank bei jedem Projekt von Supervisor vorhanden sind.

Die Treiber setzen sich aus dynamischen Bibliotheken (Dateien vom Typ .DLL) zusammen, welche auf einer Logik der Ausnahmehandlung basierend und die Aufgabe haben, die aus dem Speicherbereich des verbundenen Geräts erhaltenen Informationen nach vordefinierten Einstellungen zum Speicherbereich von Supervisor zu kommunizieren, und umgekehrt.

Mithilfe der Einstellungen des Treibers wird die Verknüpfung zwischen den Variablen des Feldes und denen von Supervisor festgelegt. Das System sorgt dafür, dass die Variablen "aus" der Anlage, (bzw. "an" die Anlage) mithilfe der seriellen Schnittstelle, Feldbus-Systems oder des verwendeten Netzes nach den konfigurierten Modalitäten gelesen und geschrieben werden.



Die Technologie, auf der die Treiber von Supervisor basieren, besteht aus raffinierten Thread-Pooling-Mechanismen, durch welche die Informationen auf die möglichst effiziente Weise mit dem Feld ausgetauscht werden, wobei eine automatische Optimierung der Kommunikation nach des effektiven Einsatz der Variablen des ausgeführten Projekts erreicht wird.

Nur die **tatsächlich von dem System eingesetzten** Variablen werden mit dem Feld ausgetauscht, wobei dem Treiber überlassen wird, die Kommunikation zu optimieren und effizient zu machen.

Die Treiber von Supervisor besitzen zudem besondere Eigenschaften, mit denen die Kommunikation des Supervisors äußerst leistungsfähig und flexibel gemacht wird.

Dank der einzigartigen Besonderheiten, die allen Treibern gemeinsam sind, wird folgendes erreicht:

1. Eine effiziente und optimierte Kommunikation
2. Links lassen sich zu SPS-Adressen direkt auf Tags oder indirekt durch "Aufgaben" verwalten.
3. Laufzeit-Konfigurierbarkeit durch VBA Scripts-Schnittstelle
4. Automatische Importierung der Gerät-Datenbank
5. Überbrückungs-Funktion (Bridging) für einen transparenten externen Zugang zum Gerät über Modem (z.B. Teleservice)
6. TAPI-Funktionen, um automatische Anrufe an remote, seriell verbundene Geräte über Modem zu ermöglichen
7. RAS-Funktionen, um automatische Anrufe an remote Ethernet-Geräte über Modem zu ermöglichen
8. Erweiterte Debugging-Funktionen
9. Der Verkabelung- und Kommunikationstest erfolgt direkt und sofort.

## 1.1. Der Treiber im Projekt

---

Ein Supervisor-Projekt setzt sich aus der Gruppierung der Ressourcen und Objekten zusammen. Wenn diese einmal compiliert und in Laufzeit geladen werden, ist das Projekt in der Lage, Informationen, Logik und die Benutzerschnittstelle genau wie programmiert zu verwalten.

Die Kommunikation mit der Anlage erfolgt durch Einsatz der Kommunikationstreiber (**neben der OPC-Technologie**). Die Treiber, in Form von dynamischen Bibliotheken, haben die Aufgabe, die

Speicherbereiche des Geräts zu lesen/schreiben, das mit den von dem Projekt verwalteten Speicherbereichen verbunden ist.



Zum Beispiel bei SPS-Geräten benutzt der Treiber das von dem Gerät vorgesehene Kommunikationsprotokoll, um Speicherbereiche von SPS zum Supervisor zu lesen oder zu schreiben, und umgekehrt. Dies erfolgt je nach den Konfigurationen und den Datenverknüpfungen, welche in den Treibereigenschaften und/oder in den Variableneigenschaften des Projekts eingestellt wurden.

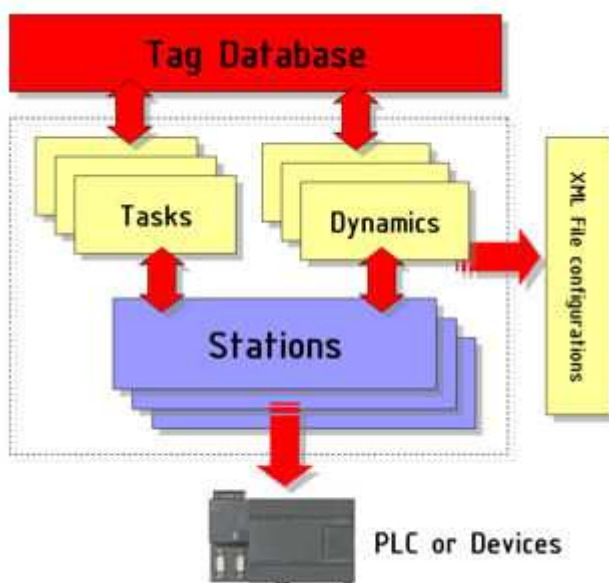
Die Speicherbereiche, in denen der Treiber lesen oder schreiben kann, neben den Übertragungsmodalitäten, werden vom Hersteller des Geräts bestimmt, und aus diesen Gründen müssen Sie zuerst den Anweisungen folgen, die spezifisch für das zu verbindende Gerät vorgesehen sind.



**Lesen Sie sorgfältig die Anweisungen des Herstellers des Geräts für die Verbindung und die Daten-Verknüpfung zwischen Gerät und Supervisor.**

Ganz unabhängig vom Protokoll und Hardwarehersteller lassen sich die Supervisor-Treiber von dem Programmierer visuell konfigurieren, so dass die Kommunikation nach den möglichen vorgesehenen Einstellungen erfolgen kann.

Die Struktur eines Kommunikationstreiber von Supervisor wird im folgenden Schema beschrieben:



Nach dem in der Abbildung dargestellten Schema wird das Kommunikationsprotokoll von dem Treiber zu einer niedrigen Ebene verwaltet.

1. Mithilfe des Konzepts "**Station**" werden die Hauptparameter der Kommunikation definiert (je nachdem, ob es sich um Seriell- oder Netz-Treiber handelt, müssen die entsprechenden Parameter definiert werden).
2. Mithilfe des Konzepts "**Tasks**" können Sie die "indirekte" Verknüpfung zwischen Adressen des Geräts und der Variablen des Projekts von Supervisor definieren. Die Tasks bieten deshalb die Möglichkeit, wie die Kommunikation in Datenblöcke definiert wird, wobei eine Variable, oder eine Gruppe von Variablen definiert werden, welche mit einer (Start-) Adresse des Geräts verknüpft sind. Auf diese Weise kann der Benutzer indirekt, also unabhängig vom Projekt, die Links zu diesen Speicherbereichen des Geräts konfigurieren.
3. Mithilfe des Konzepts "**Dynamischer Adressen**" erlaubt der Treiber eine direkte Verknüpfung der Speicheradresse in den Variableneigenschaften des Projekts (Tag) vorzunehmen. Auf diese Weise zeigt die Variable direkt auf die Adresse des Geräts, indem dem Treiber die Aufgabe zur "dynamischen" Erzeugung von Kommunikation-Tasks überlassen wird, was für eine optimierte Kommunikation sorgt.
4. Der Treiber ist deshalb eng mit der "**Echtzeitdatenbank**" des Projekts am Supervisor verbunden. Folglich lassen sich die Variablen direkt (Eigenschaften der Tags) oder indirekt

(Tasks) verknüpfen. In jedem Fall bleibt die Kommunikation nach dem "Variablen in Einsatz"-Konzept optimiert.

5. Die Konfiguration des Treibers wird in den entsprechenden **XML-Dateien** in Verzeichnis "Ressourcen" des Projekts gespeichert. Der höchsten Transparenz halber basieren die Dateien, ebenso wie das ganze Projekt, auf der XML-Metasprache. Die Dateien des Treibers sind folgende:

<name\_driver>.drvsettings = Datei, welche die allgemeinen Einstellungen des Treibers enthalten

<name\_driver>.dynsettings = Datei, welche beim Starten mit den Parametern der berechneten dynamischen Tasks erzeugt wird.

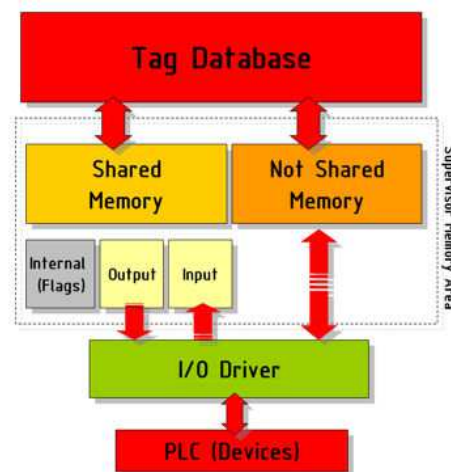


Die Treiber von Supervisor sind Bibliotheken, die sich auch unabhängig von den Supervisor-Projekten verwenden lassen, und auch in andere mit der ActiveX-Technologie kompatiblen Programmierumgebungen eingeführt werden können.

## 1.2. Speicherbereiche

Die Tags-Variablen der EchtzeitDB von Supervisor werden im Speicher des Computers, ganz unabhängig von der eingesetzten Art von Kommunikation, dynamisch abgebildet. Die Abbildung bei den Speicherbereiche der Tags des Supervisors kann in den "Allgemeine Eigenschaften" jeder Tag einzeln definiert werden.

Die Variablen können in zwei verschiedenen Datenbereiche abgebildet werden: "**Nicht Geteilten**"-Datenbereich und Datenbereich "**Geteilten**"-Datenbereich.



### Supervisor-Bereich "Nicht-Geteilt"

Der "Not-Shared"-Datenbereich wird defaultsmässig für die Erzeugung der Tag-Variablen in einem Projekt von Supervisor voreingestellt. Unter Anwendung des Nicht-Gemeinsamer Bereiches wird dem Supervisor überlassen, wie und wohin die Tags-Variablen allokiert werden. Der Benutzer muss sich deshalb keine Sorgen machen, weitere andere Adressen zuzuweisen als nur die eventuelle Link zum Gerät.

Deshalb ist es zweckmässig, diese Option zu nutzen, um potenzielle Risiken von falschen Datenallokierungen innerhalb des Supervisors zu vermeiden. Supervisor kann die eigenen Informationen automatisch intern verwalten und kommuniziert sich nur dann mit dem Treiber, wenn die Variable mit dem Gerät verbunden (eingeschaltet) ist.

### Supervisor-Bereich "Geteilt"

Der "Geteilte" Datenbereich, wenn ausgewählt, erlaubt Ihnen, die Allokierung für die Daten in dem internen Speicherbereich des Supervisors zu definieren, und zwar unabhängig von der Treiberkommunikation. Für die Auswahl dieses Bereiches ist **erforderlich, den Typ vom Bereich und die absolute Adresse der Tag im Supervisor-Speicher zuzuweisen**.

Der Programmierer muss immer die richtige Zuweisung der Adresse überprüfen, um eine unerwünschte Überlappung von Variablen im Speicher zu vermeiden.

Der Gemeinsame Bereich sieht vor, zwischen folgenden **Bereich-Typen** auszuwählen:

1. **Input-Bereich**
2. **Output-Bereich**
3. **Merker-Bereich (Intern)**

Unabhängig von dem betreffenden ausgewählten Speicherbereich, können Sie die Variablen des Geteilten Bereiches mit der Anlage, wobei Aufgaben vom Typ Eingang, Ausgang oder Eingang/Ausgang eingesetzt werden.

#### **Link zum Gerät**

Bei den Tasks müssen Sie die Verbindung zwischen den Variablen und des Supervisors, sowie die Speicherbereiche des Geräts definieren. Der Supervisor bietet Ihnen die Möglichkeit, Kommunikations-Tasks in zweierlei Moden auszuführen: Statische oder Dynamische Tasks.

**Statische Tasks:** Dies sind Tasks, die von dem Programmierer definiert werden. Dabei muss der Programmierer die auszutauschenden Variablen-Pakete sowie die Kommunikations-Parameter definieren.

**Dynamische Tasks:** Diese werden automatisch von dem Kommunikationstreiber bei dem Start des Projekts gemäß der Verbindung erstellt, die in der Eigenschaft "Dynamisch" (Physische E/A-Adresse) der Variablen des Projekts konfiguriert wurde. So wird der Treiber die Eingruppierung der Variablen verwalten, um die Kommunikation zu optimieren.

## **1.3. Adressierung über Dynamische Variablen**

---

### **Dynamische Tags**

Der Supervisor erlaubt Ihnen, die Adresse des Geräts direkt in den allgemeinen Eigenschaften der Tag-Variablen, in den Ressourcen Echtzeitdatenbank des Projekts einzugeben. Auf diesen Weg werden die Kommunikationstasks auf eine dynamische Weise von dem Treiber generiert, wobei vordefinierte Parameter für die Optimierung und Eingruppierung der Daten eingehalten werden. Beim Starten des Projekts erzeugt der Treiber tatsächlich eine angemessene Anzahl von Tasks, indem die Daten zu Blocks gruppiert und nur dann kommuniziert werden, wenn die Variablen im Projekt in Einsatz kommen.

Um eine Adresse auf dynamischen Modus zuzuweisen müssen Sie folgendes tun:

1. In den Allgemeinen Eigenschaften der Variablen wählen Sie "Dynamische Adresse" aus
2. Auf dem Fenster "Tags Explorer" wählen Sie Tabs-Treiber aus
3. Doppelklicken Sie auf den gewünschten Treiber von der Liste. Dann erscheint das Fenster zur Zuweisung der Adresse
4. Geben Sie die spezifische Adresse auf dem absoluten Speicherbereich des verbundenen Geräts ein
5. Definieren Sie in dem Feld "Type" den Typ für das Management der Daten (Lesen, Schreiben, oder beide)

Die Auswahl der Adresse des entsprechenden Geräts kann neben der Methode für Konfiguration auch direkt durch Eingabe der Adressierungssyntax erfolgen, welche normalerweise folgendermaßen aufgebaut wird:

[DRV]<Treibername>.Sta=< Name der Station>|Addr=< Adresse des Geräts>



**Wir verweisen jedoch auf die Spezifikationen, welche den entsprechenden Treiber betreffen.**

### **Über dynamische Adressen**

Wie schon erklärt lässt Supervisor beim Starten des Projekts die für die Kommunikation nötige Anzahl von Tasks erstellen.

Der determinierende Parameter zur automatischen Erstellung der Tasks bei den Eigenschaften des Treibers ist die "Minimum Threshold" (Minimale Schwelle). Mithilfe dieses Parameters wird dem Treiber die minimale Anzahl als Verteilungsschwelle für die Erstellung der Task zugewiesen.

**Zum Beispiel:**

VAR00001 ist verbunden mit der Adresse Wort 0 des Geräts  
 VAR00002 ist verbunden mit der Adresse Wort 3 des Geräts  
 VAR00003 ist verbunden mit der Adresse Wort 12 des Geräts  
 VAR00004 ist verbunden mit der Adresse Wort 18 des Geräts

Standardmässig ist der Parameter "Minimum Threshold" des Treibers zu 5 eingestellt. In unserem Beispiel beim Starten des Projekt wird der Treiber die XML-Datei ".dynsettings" in Verzeichnis "Projekt-Ressourcen" dynamisch erstellen, wobei die Erzeugung zweier dynamischer Tasks an den Treiber berechnet wird.

In der ersten Task werden die Words von 0 bis 3 gelesen, in der zweiten Task (sie wird benötigt, weil das darauf folgende zu lesende Word liegt auf einer Adresse, die höher als 5 Byte ist, wie vom Parameter "Minimum Threshold" angegeben) werden die Words von 12 bis 18 gelesen.

Würde nun der Parameter "Minimum Threshold" zu 20 eingestellt, liesse der Treiber, bei dem nächsten Start des Projekts, eine einzige Aufgabe dynamisch erstellen, indem er von Wort 0 bis Wort 18 lesen würde.



**Die Anzahl von automatisch erzeugten Tasks hängt deshalb von dem Parameter ab, der als "Minimum Threshold" eingestellt ist.**

Die Variablen, welche auf "**dynamische**" Weise mit dem Treiber verknüpft sind, werden standardmässig in Lese-Schreibe-Modalität automatisch von dem Treiber verwaltet. Er wird dann entscheiden, je nach Möglichkeiten des Bereiches des verbundenen Geräts, ob es in "nur-lese"- oder lese/schreibe-Modalität ausgeführt wird. Jedenfalls können Sie diese Einstellung durch Bearbeitung der XML-Datei, die beim Starten des Projekts in Run automatisch erstellt wird, modifizieren. Der Wert in der XML-Datei, der den Ausführungstyp betrifft, ist folgender:

<NameType> wo der Wert "1", der standardmässig eingestellt ist, mit folgenden Werten modifiziert werden kann:

- 0 = Input
- 1 = Input/Output
- 2 = Output zur Laufzeit
- 3 = Output ununterbrochen

## 1.4. Statische Tasks

Supervisor lässt die Verknüpfung zwischen Gerätsadressen und Echtzeitdatenbank-Variablen des Projekts mithilfe der Tasks auf eine indirekte Weise zuzuweisen. Durch Anwendung dieser Methode wird der Treiber erfordern, in den Eigenschaften der Konfiguration von jeder einzelnen Task, die Verknüpfung zwischen den vom Projekt definierten Variablen und die Speicherbereiche des Geräts einzugeben. Außerdem ist der Typ von Kommunikation zu definieren (Lesen, Schreiben oder beide). Die erstellte Taskliste werden in der XML-Datei ".drvsettings" gespeichert und somit unabhängig von dem Projekt bleiben.

Um Tasks zu konfigurieren und einzufügen müssen Sie folgendes tun:

1. Auf den Treibereinstellungen wählen Sie "Task" aus
2. Fügen Sie eine neue Task mithilfe der Schaltfläche "Add" hinzu
3. Konfigurieren Sie die Taskseigenschaften gemäß den Anforderungen an Daten-Austausch
4. Bestätigen Sie mit OK und setzen Sie diese Methode mit den übrigen Tasks fort.



**Wichtig: Um eine Liste von verbundenen Variablen in einer Task zu definieren, ausgehend von einer Adresse des Geräts, müssen Sie in den Variableneigenschaften die Namen der Variablen, getrennt vom Zeichen ";" manuell eingeben.**

Eine Task, die mit dieser Prozedur erstellt wird, kann eine oder mehrere Variablen einschließen, die der gleichen Art sein müssen (z.B. alle vom Typ Word, oder alle vom Typ Float, usw.) sowie einen darauffolgenden Speicherbereich der SPS lesen/schreiben. Die Startadresse, die in der Task angegeben werden soll, ist die Adresse der ersten Variable, die in der SPS gelesen wird. So wird diese Variable mit der ersten Variable der Task-Liste verknüpft.

Nur wenige Treiber erlauben die Erstellung von Tasks, die Variablen verschiedener Typen enthalten (etwa Byte und Word, usw.). Für weitere Informationen verweisen wir auf die Dokumentation der betreffenden Treiber.

Beim Ausführen von Tasks müssen Sie folgende Grundlagen beachten:

- Lesetasks** Sie lesen aus dem verbundenen Gerät und schreiben in die Supervisorvariablen. Sie können auch bei Ereignis ausgeführt werden. In diesem Fall wird die Task nur dann ausgeführt, wenn die interne Logik von Supervisor die zur Ausführung der Task entsprechenden Variablen aktiviert werden, wobei das Lesen der Daten an das Gerät ausgeführt wird. Andernfalls wird es ignoriert. Wenn die Lesetask nicht einer Ausführung bei Ereignis unterworfen ist, wird die Task unter Anwendung der Technologie "Polling" zusammen mit den weiteren Lesetasks ausgeführt.
- Schreibetasks** Sie schreiben Daten aus den Variablen von Supervisor in das verbundene Gerät. Die Schreibetasks werden mit Technologie "Ereignisgesteuert" von Supervisor ausgeführt, d.h. sie werden nur dann ausgeführt, wenn Daten der Task im Supervisor ändern und deshalb an die SPS oder an das verbundene Gerät benachrichtigt werden müssen. Der Treiber gestattet das Schreiben auch bei der kontinuierlichen Aktualisierung der Daten.
- Lese/Schreibetasks** Die Lese-Schreibe-Tasks werden in "Polling" verwaltet, um die vom Gerät gelesenen Daten aktualisiert zu halten, während das Schreiben der Werte, (wenn am Supervisor geändert), nur bei Ereignis ausgeführt werden, wobei das Wiederschreiben der geänderten Daten zum verbundenen Gerät erzeugt wird. Die Lese-Schreibe-Tasks führen außerdem immer das Lesen der Daten von dem Gerät aus und dann, wenn nötig, das Schreiben.
- Task COM (OLE2)** Supervisor kann auch andere Tasks ausführen, die nicht im Treiber direkt konfiguriert, sondern von der Basic-Script-Logik zur Laufzeit erzeugt werden. In diesem Fall werden die Tasks nach Programm ebenfalls von dem Treiber ausgeführt, wobei Lese- oder Schreibe-Tasks synchron oder asynchron ausgeführt werden.

Die Verwendung von Tasks muss von dem Programmierer besonders beachtet werden, als Ausgleich dazu bietet sie aber eine bessere Flexibilität in der Anwendung, wobei es dem Programmieren (und nicht dem Supervisor) überlassen bleibt, zu entscheiden, wie die Daten ausgetauscht werden sollen. Die Kommunikationstasks werden nur nach Variablennamen und nicht nach absoluten Adressen ausgeführt.



Es ist daran zu erinnern, dass die Tasks auch die COM-Schnittstelle (Component Object Model) für die eventuelle Verwaltung oder Erzeugung der Kommunikationstasks zum Treiber mithilfe **VBA-Scripts** gestatten.

---

## 1.5. Installation von Treibern

---

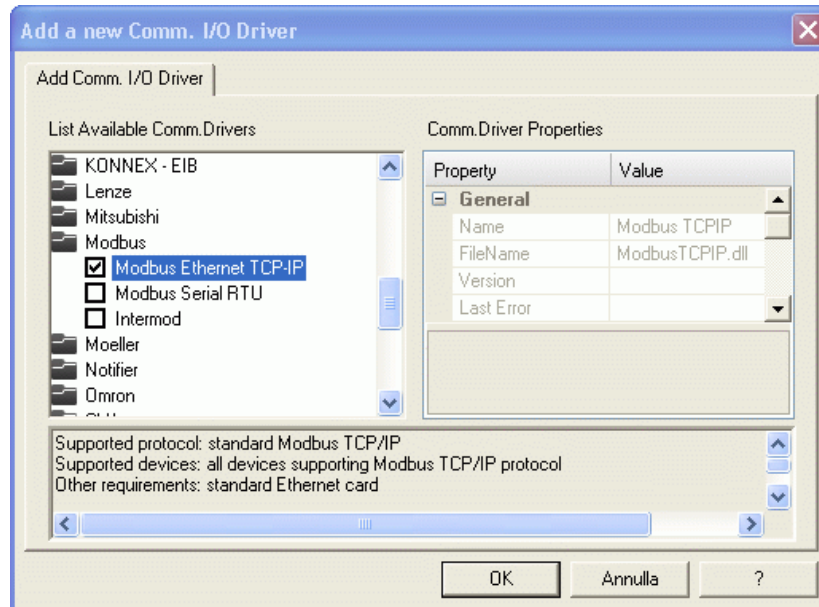
Die Installation von Supervisor lässt die Bibliothek der verfügbaren Treiber zusammen mit der Plattform automatisch installieren.

Die Treiber von Supervisor, da sie aus einfachen .DLL-Dateien bestehen, können auf eine einfache Weise und unabhängig von der Entwicklungsplattform erweitert oder aktualisiert werden. Um einen neuen Treiber zu installieren bzw. zu aktualisieren, müssen Sie einfach die entsprechende .DLL-Datei in Treiber-Verzeichnis kopieren, das sich innerhalb des Installationsverzeichnis von Supervisor befindet (normalerweise C:\Program Files\Progea\Movicon11.2\Drivers).



Der Programmierer wird je nach den verschiedenen Anwendungen in Funktion der Bedürfnisse auswählen können, welcher Kommunikationstreiber eingefügt oder konfiguriert werden soll, der einfach aus der Liste ausgewählt werden kann.

Die Einfügung und Einstellung eines Kommunikationstreibers erfolgt bei Supervisor in Programmierungs-Modus mithilfe der Ressource "Comm.Drivers-Liste" aus der Gruppe "Echtzeitdatenbank", die sich am Fenster "Projekt-Explorer" befindet. Wird der Befehl "Neuen Treiber hinzufügen" betätigt, wird das entsprechende Dialogfenster mit der Liste der vorhandenen Treiber visualisiert.



Das Fenster zum Auswählen eines Kommunikationstreibers visualisiert unten einige wichtige Informationen:

- Kurze Beschreibung des Protokolls
- Liste der unterstützten Geräte
- Benötigte Netzwerkkarten oder Bibliotheken
- Eventuelle Beschränkungen des Treibers: Nicht unterstützte Bereiche, nicht unterstützte Verbindungen mit mehreren SPSen, usw.

Nachdem der Treiber schon eingefügt wurde, können Sie ihn mithilfe des Fensters "Eigenschaften" von Supervisor konfigurieren.

Natürlich können Sie mehrere Kommunikationstreiber in ein einziges Projekt einfügen, sofern dies mit den in dem Hardware-Schlüssel definierten Optionen kompatibel ist.

Die Liste der "Funktionalitäten" der Kommunikationstreiber enthält nun das neue Item "Renaming Manager", das besagt, ob der Treiber die Umbenennungsfunktion unterstützt. Die Treiber, welche die Umbenennungsfunktion unterstützen (mit Optionsbit auf 'true'), lassen die Variablen mit dem geänderten Namen in ihren Konfigurationsfenstern anzeigen und unterstützen den Befehl der Anwendung der Umbenennung.

## 1.6. Treiber-Verdopplung

Supervisor gestattet zwei oder mehr Treiber im System zu installieren, die sich von der Liste der verfügbaren Treiber auswählen lassen, wie wir schon in den vorhergehenden Paragraphen gesehen haben. Wollen Sie jedoch zwei oder mehrere identische Treiber installieren (für SPS oder Geräte der gleichen Art), werden Sie dazu den folgenden, einfachen Anweisungen folgen müssen.

Gegebenenfalls müssten Sie z.B. zwei oder mehrere Treiber der gleichen Art installieren, um in derselben Systemumgebung gleichzeitig zu kommunizieren, was aber nicht genug ist, um einfach zwei verschiedene Stations mit demselben Protokoll für das Gerät vom gleichen Typ zu konfigurieren.

Beispiel: Sie wollen den Rechner, an dem Supervisor installiert ist, mit zwei identischen Geräten kommunizieren, die an zwei verschiedenen Kanäle von PC-Kommunikation sind. Um den Kommunikationstreiber zu verdoppeln, müssen Sie folgendes tun:

1. Verdoppeln Sie die betreffende DLL, die sich im Ordner "Drivers" auf dem Installationsverzeichnis befindet. Die Datei muss einen anderen Namen besitzen (von dem ursprünglichen verschieden), bspsw. denselben Namen plus einen fortschreitenden Index.
2. Aktualisieren Sie die "Drivers.xml"-Datei, indem die soeben verdoppelte .DLL-Datei angegeben wird. Geben Sie ihnen Name-Beschreibungen, so dass sie wie zwei identische Treiber erscheinen, aber unter verschiedenen Namen und Beschreibungen in der Liste der Treiber von Supervisor.



**Ist ein Treiber verdoppelt, wird er alle Einwirkungen eines neuen Treiber ergeben. Deshalb werden Sie einen zusätzlichen Treiber bei der Lizenz-Optionen beschaffen. Wenn Sie zwei Treiber benutzen, selbst wenn sie wegen deren Verdopplung der gleichen Art sind, muss die Lizenz für zwei Kommunikationstreiber eingeschaltet werden.**

**Ein Beispiel in der Praxis:**

Sie wollen ein Projekt verwalten, in dem zwei "Modbus Serial"-Treiber verwendet werden müssen. Dazu muss man folgendes tun:

**Dll - Verdopplung**

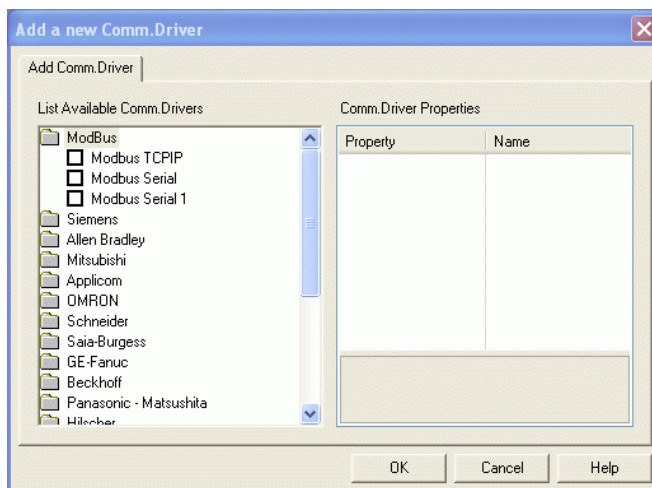
In dem Unterordner "Drivers" der Supervisor-Installation führen Sie eine Kopie des Originals "**Modbus.dll**" aus und benennen Sie zu bspsw. "**Modbus1.dll**" um.

**Bearbeitung der Datei Drivers.xml**

Öffnen Sie die Datei "Drivers.xml" aus dem Unterordner "Drivers" mit einem Text-Editor. In der Tag <DriverFactory Factory="ModBus"> fügen Sie den neuen Treiber ein, indem irgendeine Beschreibung und der Name des neuen dll eingegeben wird, z.B.: <Driver Name="Modbus Serial 1">ModBus1.dll</Driver>. Mit der bearbeiteten Datei bekommt man:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DriverList>
  <DriverFactory Factory="ModBus">
    <Driver Name="Modbus TCPIP">ModBusTCPIP.dll</Driver>
    <Driver Name="Modbus Serial">ModBus.dll</Driver>
    <Driver Name="Modbus Serial 1">ModBus1.dll</Driver>
  </DriverFactory>
  ...
  ...
```

Auf diese Weise, wenn Sie das Fenster der vorhandenen Treiber öffnen, werden Sie auch den verdoppelten Treiber finden.



## 1.7. Treiberausführung

Der (Die) installierte(n) Kommunikationstreiber werden mit dem Start des Projekts ausgeführt und bleiben während der gesamten Verarbeitungsdauer des Projekts aktiv, je nach den ausgewählten Ausführungsmodalitäten oder der VBA-Script-Logik, welche die COM-Schnittstelle der Treiber benutzen.

Jedesmal wenn die Kommunikation mit dem Feld aktiviert wird, registriert das System in das historische Log eine Meldung zur Benachrichtigung über den Status der Kommunikation.



Die Darstellung einer grünen Leuchtdiode auf der Statusleiste am Fuss des Supervisor-Fensters (wenn visualisiert) signalisiert, dass die Kommunikation mit dem Feld des installierten Treibers richtig läuft. Die rote Farbe in der Leuchtdiode signalisiert indes ein Fehler in der Kommunikation.

Eventuelle Kommunikationsprobleme (Kabel, Verbindung, Einstellungen, usw.) werden Kommunikationsfehler verursachen, die von dem Treiber angegeben, in der Statusleiste hervorgehoben und in das historische Log registriert werden.



Bemerkung: Die Treiber sind unabhängig vom Projekt, und deren Konfiguration wird in entsprechende Dateien getrennt, individuell, unter der Erweiterung ".drvsettings" und "dynsettings" gespeichert. Diese Philosophie erlaubt Ihnen, das Projekt bei einer eventuellen Änderung der SPS oder Kommunikationsgerät ganz unberührt zu halten.

Die Ausführung eines Treibers im Projekt während der Laufzeit von Supervisor kann den von Programmierer gestellten Bedingungen unterworfen sein.

## 1.8. Hardware-Fehler RS232

Die Autodiagnose der Kommunikationstreiber von Supervisor sieht die Erzeugung von Hardware-Codes der Hardware-Fehler bei der Kommunikation vor, gemäß dem Statusregister des UART-Chips der im Rechner installierten Seriellen Karte.

Wir empfehlen Seriellen mit UART-Chip vom Typ 16550A zu benutzen, welche einen FIFO-Code von 16 Byte benutzen. Die im Rechner installierte Serielle lässt sich leicht durch Ausführung der Datei Microsoft Diagnostic MSD.EXE erkennen.

Die Hardware-Fehler der Kommunikation werden normalerweise von folgenden Ursachen hervorgerufen:

- Störungen über serielle Linie
- Potenzialdifferenz (Spannung) zwischen den Erdungen der Geräten
- Seriellkarte, die für die benötigten Anforderungen ungeeignet ist
- Kommunikationskabel defekt oder ungeeignet
- zu hohe Baud-Frequenz für die verwendete Hardware
- Defekte bei Kommunikationsgeräten

Die vom Treiber selbst angezeigten Hardware-Fehler berücksichtigen den im Fehlerregister herausgegebenen Code vom Chip UART der Seriellen Karte.

**Bei Hardware-Fehler zeigt Supervisor einen allgemeinen Fehler, der eine Nummer aufweist. Mithilfe der angezeigten Zahl (zu Binärzahlen übersetzt) können die Fehler je nach der in der Tabelle angegebenen Bedeutung erkannt werden:**

Wert	CODE	BEDEUTUNG
1	RX OVER	Die Serielle hat mehr Zeichen als zulässig in ihrem eigenen Puffer erhalten
2	OVERRUN	Bevor das vorletzte vom System verarbeitet werden konnte, hat die Serielle ein weiteres Zeichen erhalten
4	RX PARITY	Fehler bei der Datenparität, Unübereinstimmung zwischen erhaltene und

		eingestellte Parität
8	FRAME	Fehler bei dem Datensatz. Die erhaltenen Daten berücksichtigen die eingestellten Kennwerten nicht (Länge, Stopbit, usw.)
16	BREAK	Breakstatus (Wartezeit), die von einem Teilnehmer verlangt wurde



Alle anderen Kommunikationsfehler hängen vom spezifischen Protokoll ab: Wir verweisen auf die spezifischen Anweisungen für den betreffenden Treiber. Die Fehlermeldungen der Treiber werden in der Statusleiste angezeigt und werden im historischen Log visualisiert.

# 2. Allgemeine Einstellungen

---

## 2.1. Allgemeine (Treiber)

---

In dieser Einstellungskarte können Sie einige Eigenschaften konfigurieren, die alle Kommunikationstreiber gemeinsam haben. Normalerweise ist es nicht nötig, die Standardeinstellungen zu modifizieren.

### Wait Time

Pausezeit. Sie wird in Millisekunden angegeben und zwischen die Ausführung zweier darauffolgender Kommunikationstasks (Datenblocks) gezählt. Standardmässig wird der Wert 0 (keine Pause) eingestellt. Die Änderung des Standardwerts (0) könnte für gewisse Geräte nötig sein, welche eine Pause zwischen darauffolgenden Abfragen brauchen (z.B. Geräte mit beschränkten Leistungen).

### Timeout

Timeout zur Ausführung von synchronen Tasks. Der Wert wird in Millisekunden angegeben.

### Minimum Threshold

Dieser Parameter bestimmt die minimale Schwelle für die Verteilung der mit dem Gerät ausgetauschten Datensätze. Wie in der Einführung erklärt, berechnet Supervisor bei der dynamischen Verwaltung der Kommunikation automatisch (beim Starten des Projekts) die Größe und die Anzahl von dynamischen Tasks, die für die Kommunikation bezüglich der dynamischen Variablen (dynamische Adressierung der Tags) erstellt werden sollen.

Supervisor wird tatsächlich versuchen, die Kommunikation durch Hinzufügen der möglich höchsten Anzahl von Daten in eine einzige Task zu optimieren. Wenn Daten zwischen entfernten Adressen miteinander verbunden sind, bestimmt dieser Wert den Abstand in Byte, der darüber entscheiden lässt, ob Supervisor eine neue Task für den nächsten Datenblock erstellen soll.

### Zum Beispiel:

VAR00001 ist verbunden mit der Adresse Wort 0 des Geräts  
VAR00002 ist verbunden mit der Adresse Wort 3 des Geräts  
VAR00003 ist verbunden mit der Adresse Wort 12 des Geräts  
VAR00004 ist verbunden mit der Adresse Wort 17 des Geräts

Standardmässig ist der Parameter "Minimum Threshold" des Treibers zu 5 eingestellt. In unserem Beispiel beim Starten des Projekts wird der Treiber die XML-Datei ".dynsettings" in Verzeichnis "Projekt-Ressourcen" dynamisch erstellen, wobei die Erzeugung zweier dynamischer Tasks an den Treiber berechnet wird.

In der ersten Task werden die Words von 0 bis 3 gelesen, in der zweiten Task (sie wird benötigt, weil das darauf folgende zu lesende Word liegt auf einer Adresse, die höher als 5 Byte ist, wie vom Parameter "Minimum Threshold" angegeben) werden die Words von 12 bis 17 gelesen.

Würde nun der Parameter "Minimum Threshold" zu 20 eingestellt, liesse der Treiber, bei dem nächsten Start des Projekts, eine einzige Aufgabe dynamisch erstellen, indem er von Wort 0 bis Wort 18 lesen würde.



**Die Anzahl von automatisch erzeugten Tasks hängt deshalb von dem Parameter ab, der als "Minimum Threshold" eingestellt ist.**

### Hinzufügungslimit

Dieser Parameter erlaubt Ihnen, die maximale Anzahl von Bytes einzugeben, die für jede dynamische Task hinzugefügt wird. Wird dieser Parameter zu Null gelassen (Standard-Wert), wird der Treiber den maximalen Wert annehmen, der in dem ausgewählten Protokoll als maximales Limit eingestellt wurde. Diese Modifikation ist dann nötig, wenn man Geräte benutzt, welche ein niedrigeres Maximum-Limit von Byte-Anzahl besitzen, verglichen mit dem Protokoll, das für die Austausch von Task verwendet wird.

### Synch.Startup

Diese Option bestimmt beim Starten des Projekts die Synchronisierung zwischen Logik und Treiberkommunikation.

Wird sie zu "true" eingestellt, sorgt Supervisor dafür, dass die statischen Input-Tasks vollständig ausgeführt werden, bevor die Logik und Scripts des Projekts verarbeitet werden.

Standardmässiger Wert = false.

Diese Option bewirkt eine längere Zeit beim Starten des Projekts, aber die Logik wird mit der Genauigkeit von "aktualisierten" Input-Werten ausgeführt.

### COM Interface

Das zeigt, ob der Treiber die COM-Schnittstelle (Component Object Model) unterstützt. Die COM-Schnittstelle, auch OLE2 genannt, gestattet die Verwendung der Logik vom VBA-Scripts für das Management der Treiber, deren Methoden, Eigenschaften und Ereignisse in den entsprechenden Informationen beschrieben sind.

Standardmässiger Wert = true.

### Polling Time

Dieser Parameter (in Millisekunden angegeben) stellt die minimale Polling-Zeit für die Ausführung der Tasks zur Aktualisierung der Daten **wenn die Variablen nicht im Einsatz sind**.

Dieser Wert wird von allen dynamischen Tasks angenommen, welche deshalb die gleiche Polling Time haben werden (nur für einige Treiber besteht die Möglichkeit, eine verschiedene Polling Time je dynamische Task zu definieren). Bei **statischen** Tasks wird dieser Wert indes als Standardwert (default) in deren gleichnamigen Eigenschaft ("Polling Time") eingestellt, sobald sie erstellt werden. Mithilfe von der Eigenschaft der Statischen Task kann dieser Wert nachträglich modifiziert werden.

Wird dieser Wert zu Null gesetzt, werden die Daten mit der schnellstmöglichen Geschwindigkeit aktualisiert.

### Unused Polling Time

Dieser in Millisekunden angegebene Parameter erlaubt Ihnen, eine Datenaktualisierung zu forcieren, wenn die Variablen **nicht in Einsatz sind, wobei eine Polling-Zeit festgelegt wird**.

Dieser Wert wird von allen dynamischen Tasks angenommen, welche deshalb die gleiche Polling Time haben werden (nur für einige Treiber besteht die Möglichkeit, eine verschiedene Polling Time je dynamische Task zu definieren). Bei **statischen** Tasks wird dieser Wert indes als Standardwert (default) in deren gleichnamigen Eigenschaft ("Unused Polling Time") eingestellt, sobald sie erstellt werden. Mithilfe von der Eigenschaft der Statischen Task kann dieser Wert nachträglich modifiziert werden.



Wenn Sie diesen Parameter zu 0 einstellen, werden keine Tasks ausgeführt, wenn die Variablen **nicht im Einsatz** sind.

### Error Polling Time

Dieser Parameter (in Millisekunden angegeben) ist die Zeit, während der keine Task für diejenige Station ausgeführt werden muss, in ein Fehler aufgetreten ist. Wenn ein Fehler in einer Station auftritt, wartet der Treiber diese Zeit, bevor ein neuer Versuch durchgeführt wird.

### Def. Refresh Time

Noch nicht benutzt. Das wird für zukünftige Anwendungen benutzt.

### Protocol Priority

Dieses Kästchen erlaubt Ihnen, die Priorität der Kommunikationsthreads einzustellen, d.h. diejenige Priorität, welche die Ausführung des Treibers bezüglich anderer Supervisor-Vorgänge haben soll.

Die mögliche Werte, in wachsendes Prioritätsniveau, sind folgende:

- **Normal**
- **High**
- **Very High**
- **Real Time**



Es ist ratsam, bei der Modifizierung des Standard-Werts (Normal) vorsichtig umzugehen, um Risiken von übertrieben viele Arbeitsaufgaben der CPU (CPU zu 100%) zu vermeiden.

Dabei kann es nützlich sein, die Prozess-Priorität des Treibers zu steigern, wenn zu viele Kommunikationsaufgaben vorliegen. In diesem Fall empfehlen wir, den Parameter Wait Time zu einem von Null verschiedenen Wert einzustellen, um zu vermeiden, dass der Treiber zu viele Systemressourcen in Anspruch nimmt.

### Suspend tasks in case of error

Diese Eigenschaft ist nur für jene Treiber aktiv, welche die symbolische Adressierung der Tasks, "Tag Name", zulassen. Sie bleibt mit dem Wert "True" für jene Treiber deaktiviert, die nur die numerische Adressierung, "Address Type", zulassen.

Wird diese Eigenschaft auf "True" gesetzt, werden im Falle eines Kommunikationsfehlers alle Tasks deaktiviert, mit Ausnahme der Auftragsvariable, die den Fehler erzeugt hat. Dieses Verhalten kann unangemessen sein, sollten Treiber wie Beckhoff TwinCAT oder Allen-Bradley Ethernet/IP die Verwendung der symbolischen Adressierung ermöglichen (Namen der Gerätevariablen). Also kann die Option auf "False" gesetzt werden, wodurch alle Tasks auch im Fehlerfall aktiv bleiben. Wird die Option "Suspend tasks in case of error" auf "False" gesetzt:  
 setzt der Treiber die Abfragen auch dann fort, wenn eine Abfrage nicht erfolgreich abgeschlossen wurde;  
 wird eine nicht erfolgreich abgeschlossene Abfrage nach Verstreichen der im Parameter "Error polling Time" eingestellten Zeit neu gestartet;  
 wirkt sich jeder Kommunikationsfehler auf den Kommunikations-Bit-Status (\_SysVar\_:CommDriverStatus) aus.

**Management of "In Use" state for structures**

Wird dieser Parameter auf "True" gesetzt, lässt er alle Elemente einer Strukturvariable als verwendet aktivieren (im Einsatz), auch wenn nur ein Teil dieser effektiv im Projekt verwendet wird. Das verleiht die Gewissheit, dass der Treiber die gesamte Strukturvariable austauscht, auch wenn nur eines ihrer Elemente im Projekt im Einsatz ist. Auf "False" gesetzt wird der Status "Im Einsatz" einzeln für jedes Element der Struktur verwaltet; somit tauscht der Treiber nur die effektiv im Projekt verwendeten Elemente aus.



Achtung! Mit dieser Eigenschaft auf "True" könnten die Kommunikationsleistungen deutlich schwächer werden, vor allem, wenn das Projekt zahlreiche Strukturvariablen mit vielen Elementen enthält.

**Direct Output for Input/Output**

In dieser Eigenschaft kann das Verhalten des Treibers bei der Ausführung dynamischer Tasks vom Typ "Eingabe/Ausgabe" geändert werden. Ist sein Wert auf "False" gesetzt und wurde der Variablenwert über den Supervisor geändert, erfolgt vor dem Ausgabeprozess ein Eingabeprozess und wird anschließend, falls der Wert im Gerät nicht geändert wurde, der Ausgabevorgang abgeschlossen. Setzt man den Parameter dagegen auf "True", kann eine Eingabe/Ausgabe-Task direkt in Ausgabe, d. h. ohne ein erneutes Ablesen ausgeführt werden.

## 2.2. Debug

In diesem Einstellungskarte können Sie einige Eigenschaften für den Debug konfigurieren, die alle Kommunikationstreiber gemeinsam haben.  
 Normalerweise ist es nicht nötig, die standardmässigen Einstellungen zu modifizieren.

**Debug Window**

Wird "true" ausgewählt, zeigt der Treiber in dem entsprechenden Debugfenster (das sich aus dem Outputleiste des Arbeitsbereiches von Supervisor visualisieren lässt) alle Diagnose- und Debug-Nachrichten, die eventuell vom Treiber erzeugt werden.  
 Wird "false" ausgewählt, wird das Debugfenster nicht visualisiert.

**Max Entries**

Dieses Kästchen dient zur Definition des Werts, der die maximale Anzahl von Zeilen der Diagnosemeldungen bestimmt, die im Debugfenster angezeigt werden sollen, bevor alles aktualisiert wird. Wenn Sie den standardmässigen Wert belassen, behält das Fenster die letzten 10 000 Zeilen der Nachrichten visualisiert.

**Enable Log to file**

Wird "true" ausgewählt, registriert der Treiber über die entsprechende Log-Datei alle Diagnose-Nachrichten von Debug, die eventuell vom Treiber erzeugt werden.  
 Die Anzahl von registrierten Daten ist die in der vorigen Einstellung eingestellten, und der Name und Pfad der Datei, werden in der nächsten Einstellung definiert.

**Log FileName**

In diesem Kästchen müssen Sie Namen und Pfad der Datei für das Diagnose-Log des Treibers angeben.  
 Die erstellte Datei ist eine normale Text-Datei.





# 3. Einstellungen der Stations

---

## 3.1. Stationen

---

In dieser Einstellungskarte müssen Sie die "**Kommunikationsstations**" einfügen und konfigurieren. Alle Treiber müssen mindestens eine eingefügte und konfigurierte Kommunikationsstation haben.



Es ist daran zu erinnern, dass Supervisor die Möglichkeit bietet, die Treiberkommunikation sowohl über "dynamische Aufgaben" als auch über "**Tasks**" einzustellen (sie sind in entsprechenden Abschnitten beschrieben). Die dynamischen Aufgaben werden automatisch von dem Treiber beim Starten des Projekts erstellt, in Funktion der Links von Adressen des Geräts, die in den Eigenschaften "Dynamische Tasks" jeder einzelnen Tag definiert werden.

Die Kommunikationsstationen lassen definieren, wie der Treiber die Kommunikation verwalten soll, wobei alle "Stationen" einen Kommunikationskanal zu einem der vorgesehenen Geräten darstellt.

### Add

Die Schaltfläche "Add" erlaubt Ihnen, eine neue "Station" für den Kommunikationstreiber einzufügen. Wird eine neue Station eingefügt, erscheint automatisch das entsprechende Einstellungsfenster der dafür nötigen Kommunikationsparameter. Die eingefügte(n) Station(en) können nun mithilfe der unten beschriebenen Schaltflächen modifiziert oder entfernt werden.

### Edit

Die Schaltfläche "Edit" erlaubt Ihnen, die Parameter einer vorher eingefügten "Station" zu modifizieren. Sie müssen also die gewünschte Station auswählen und die Schaltfläche "Edit", oder Doppelklick, betätigen.

### Remove

Die Schaltfläche "Remove" erlaubt Ihnen, eine vorher eingefügte Station zu entfernen. Sie müssen also die gewünschte Station auswählen und die Schaltfläche "Remove" betätigen.

### Test Cable/Comm.

Diese Schaltfläche erlaubt Ihnen, einen schnellen Kommunikationstest mit dem Gerät auszuführen. Mithilfe dieser nützlichen Funktion hält der Treiber eine Kommunikation, wobei die richtige Verbindung in der Verkabelung überprüft wird.



Der Kommunikationstest wird durch das Lesen von Daten nach spezifischen Testkriterien, die von jedem einzigen Protokoll abhängig sind, ausgeführt. Es ist deshalb ratsam, die Spezifikationen jedes Treibers genau zu lesen, um den Test-Typ zu entscheiden.

Der Erfolg des Tests kann tatsächlich von dem spezifischen Gerät abhängen. Z.B. bei allgemeinen Protokollen wie "Modbus" sieht der Test die Sendung einer Abfrage von Daten vor, die nach einem gewissen Function-Code gelesen werden sollen, der nicht in dem verbundenen Gerät implementiert sein könnte.

Außerdem ist es daran zu erinnern, dass der Test nur die richtige Konfiguration des Kabels und allgemeinen Parameter überprüft. Es bleibt immer Aufgabe des Programmierers, die richtige Verknüpfung von Daten zwischen Supervisor und Gerät zu überprüfen.

## 3.2. Allgemeinen (Stationen)

In dieser Einstellungskarte müssen Sie für die ausgewählte Station die Einstellungen bezüglich der Gruppe von "allgemeinen " Eigenschaften definieren.

### Station Name

Name zur Kennzeichnung der Station, welche dem Gerät entspricht, mit dem man kommunizieren will. Er ist ein interner Name im Treiber.

Werden mehrere Stationen definiert, müssen die Namen jeweils eindeutig sein.

Standard Wert = "Default Station".

### Error Threshold

Wenn Kommunikationsfehler auftreten, definiert dieser Parameter die erreichte Fehleranzahl, die der Treiber tatsächlich als Kommunikationsfehler benachrichtigen wird. Ein interner Zähler ermöglicht, bei gestörten Kommunikationen, Fehler nicht sofort anzuzeigen, sondern die Kommunikation wiederherzustellen versuchen. Wird die Anzahl von Versuchen erreicht, gibt der Treiber erst dann die Fehler-Benachrichtigung heraus.

Standardmässiger Wert = 1.

**Status/Befehl Variable** (Unterstützt nur von denjenigen Treibern, die einer Treiber-Bibliothek ab dem Build 250 gehören)

In dieser Eigenschaft können Sie den Namen einer numerischen Variable eingeben, die bereits in Movicon definiert ist (vom Typ "Byte" empfohlen), um folgende Funktionen zu verrichten: a) den Status der Kommunikation mit der gewählten Station zu kontrollieren; b) die Kommunikation mit der Station zu aktivieren / deaktivieren; c) TAPI (nur für serielle Treiber) bzw. RAS (nur für Ethernet-Treiber) zu starten/stoppen; d) zwischen mehreren Servers TCP / IP (nur für Ethernet-Treiber) umschalten. Unten ist die Bedeutung von jedem Bit der Variable dargestellt.

Es ist anzumerken, dass einige Bits können nur zur Verifizierung des Status der Station, während andere zum Erteilen von Befehlen an Stationen eingesetzt werden.

- Das Bit Nr 1 der Variable kann zum Kontrollieren des Status der Station eingesetzt werden (Active / Inactive, Aktiviert/Deaktiviert).
- Die Bits 4, 5, 6 können zum Management einer TAPI-Verbindung (mit seriellem Treiber) oder RAS-Verbindung (mit einem Ethernet-Treiber).

Es ist ebenso möglich, einfach eine Variable vom Typ Bit als Status-Variable einzusetzen, sofern die einzige Information, die signalisiert werden muss, alleine der Status der Kommunikation mit der Station ist.

<b>Bit 0 (Status)</b>	Kommunikations-Status : 0 = OK 1 = Fehler
<b>Bit 1 (Status/Befehl)</b>	Station-Status: 0 = aktiv/aktiviert 1 = inaktiv/deaktiviert
<b>Bit 2 (Befehl)</b>	Schaltet den aktiven TCP Server um
<b>Bit 3 (Status)</b>	TCP server aktiv: 0 = Primärserver 1 = Backup-Server
<b>Bit 4 (Status)</b>	Status der Kommunikation über Modem: 0 = nicht verbunden 1 = verbunden
<b>Bit 5 (Befehl)</b>	Öffnet die Modem-Verbindung
<b>Bit 6 (Befehl)</b>	Trennt die Modem-Verbindung

<b>Bit 7 (Nicht benutzt)</b>	Immer auf 0
----------------------------------	-------------

**Keep Opened**

Dies erlaubt Ihnen darüber zu definieren, ob der Treiber (**nur für serielle Treiber**) den Kommunikation-Port immer geöffnet (also immer beschäftigt) halten muss. Wird der Wert "true" eingestellt, wird der Treiber beim Starten der Ausführung des Projekts geladen, wobei der verknüpfte Kommunikation-Port immer geöffnet (beschäftigt) bleibt.

Wird indes "false" eingestellt, wird der serielle Kommunikation-Port nach jeder Lese- oder Schreibe-Aktion von dem Treiber geschlossen, wobei der Port also freigegeben bleibt.

## 3.3. Einstellungen des Seriellen Ports

In dieser Einstellungskarte müssen Sie die Einstellungen zur Konfiguration des seriellen Ports für die ausgewählte Station definieren.

In Verwendung der TAPI-Einstellungen der Station zur Verwaltung der Modem-Kommunikation wird der serielle Port vom Modemtreiber angesteuert. Die Einstellungen, mit denen der serielle Port geöffnet wird, sind in den erweiterten Eigenschaften des Modemtreibers festgelegt, die andere sein können als die in der Station des Movicon-Treibers eingestellten ("Einstellungen des Seriellen Ports").

**Port**

Das definiert die Nummer des seriellen COM-Ports, der für die Kommunikation benutzt wird.



Bemerkung: vergewissern Sie sich, dass keine Konflikte beim Verwenden der Ports in Windows entstehen. Z.B. wenn Sie die Ports Com3 und Com4 installieren, müssen Sie sicher sein, dass die zugewiesene Adresse und die IRQ mit der Konfiguration des Rechners kompatibel sind. Wir empfehlen dazu adressierbare serielle Karten.

**Baudrate**

Das definiert die Geschwindigkeit der seriellen Kommunikation (Baudrate). Sie müssen den Geschwindigkeitswert einstellen, die dem Gerät entspricht, mit dem man kommunizieren will.

**Byte Size**

Das definiert die Anzahl von Byte, die von dem betreffenden Kommunikationsprotokoll verlangt wird.

**Parity**

Das definiert die Parität, die von dem betreffenden Kommunikationsprotokoll verlangt wird.

**Stop Bits**

Das definiert die Anzahl von Bit, die von dem betreffenden Kommunikationsprotokoll verlangt wird.

**Flow Control**

Das definiert die Datenfluss-Kontrolle für den verwendeten Typ von Kommunikation. Insbesondere erlaubt dies Ihnen, die Kommunikation anzupassen, je nach den Bedürfnissen, welche von dem seriellen Port des verbundenen Geräts in niedrigeren Ebenen erfordert werden. Standardmässig stellt der Treiber "None" ein, d.h. keine Flusskontrolle, aber je nach Gerät könnte es nötig sein, einen Typ von Flusskontrolle auszuwählen (z.B. falls Fehler mit dem Code "1" benachrichtigt werden). Die möglichen Optionen sind folgende:

**None:** Keine Flusskontrolle. Die Kontrolle wird nicht von dem Protokoll verlangt.

**Hardware:** Die Flusskontrolle wird von elektrischen Signalen aus der seriellen Linie (z.B. RTS, CTS, usw.) gesteuert.

**Xon/Xoff:** Die Flusskontrolle ist vom Typ Xon/Xoff.

**NONE (Signal disabled):** Stellt die Flusskontrolle auf NONE ein und konfiguriert den seriellen Port so, dass die Verwaltung der DTR- und RTS-Signale deaktiviert ist.

**RTS Toggle:** Stellt den seriellen Port so ein, dass das RTS-Kontrollsignal im Umschaltmodus verwaltet wird, das heißt der Port hält das Signal aktiv, solange zu sendende Zeichen vorhanden sind.

## 3.4. Puffer-Größe

---

In dieser Einstellungskarte müssen Sie die Einstellungen zur Konfiguration der Puffergröße (Queue Size) des seriellen Ports für die ausgewählte Station definieren.



**Normalerweise sind diese Einstellungen für fortgeschrittene Benutzer reserviert. Es ist deshalb ratsam, die standardmässige Einstellungen unverändert zu lassen.**

### Rx Queue

Das definiert die Anzahl von Byte, die von dem seriellen Port als Empfänger-Puffer verwaltet soll. Wenn nicht anders angegeben wird der standardmässige Wert 4096 angenommen.

Für Expertenbenutzer kann dieser Wert zur Anpassung nach den Bedürfnissen von dem verwendeten System modifiziert werden.

### Tx Queue

Das definiert die Anzahl von Byte, die von dem seriellen Port als Sender-Puffer verwaltet soll. Wenn nicht anders angegeben wird der standardmässige Wert 4096 angenommen.

Für Expertenbenutzer kann dieser Wert zur Anpassung nach den Bedürfnissen von dem verwendeten System modifiziert werden.

## 3.5. Timeouts

---

In dieser Einstellungskarte müssen Sie die Einstellungen bezüglich der Timeouts der seriellen Kommunikation für die ausgewählte Station definieren.



**Normalerweise sind diese Einstellungen den fortgeschrittenen Benutzern vorbehalten. Es ist deshalb ratsam, die Standard-Einstellungen unverändert zu belassen.**

### Rx Timeout

Das definiert die Zeit in Millisekunden, nach welcher der Treiber die Timeout Kommunikation "im Empfang" benachrichtigen soll.

Die Timeout bezieht sich auf den Empfang der gefragten Daten.

### Tx Timeout

Das definiert die Zeit in Millisekunden, nach welcher der Treiber die Timeout Kommunikation "in Sendung" benachrichtigen soll.

Die Timeout bezieht sich auf die Sendung der Daten.

### Folgende Parameter sind nur auf die Serielle Treiber bezogen:

#### CD Timeout

Zur Zeit nicht benutzt. Das ist für zukünftige Anwendungen reserviert.

#### CTS Timeout

Das definiert die Zeit in Millisekunden, nach der ein Treiber die Timeout der Kommunikation für den seriellen CTS-Parameter benachrichtigen soll. Sie definiert die Zeit, innerhalb der die einzelne Lese-Aktion zu niedriger Ebenen (API von Windows) über seriellen Port vollendet werden muss.

#### DSR Timeout

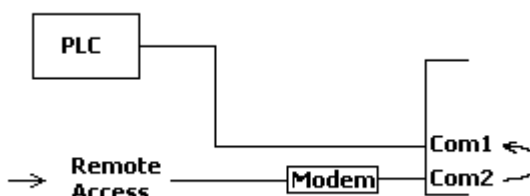
Das definiert die Zeit in Millisekunden, nach der ein Treiber die Timeout der Kommunikation für den seriellen DSR-Parameter benachrichtigen soll. Er definiert das maximale Zeitintervall, das zwischen dem Empfang eines Zeichens und dem nächsten vergeht. Außerdem soll er die maximale Zeitspanne definieren, innerhalb der die einzelne Lese-Aktion in niedriger Ebenen (API von Windows) über den seriellen Port vollendet werden muss.

## 3.6. Einstellungen des Bridging-Dienstes

In dieser Einstellungsgruppe müssen Sie die Konfigurationen der Bridging-Funktion für die ausgewählte Station definieren.

Diese Überbrückungsfunktionen erlauben Ihnen, den Supervisor als "Brücke" für den Dienst des Teleservice zu verwenden, wobei sich einen eventuellen Modem-Port im Rechner zu einer remoten SPS transparent machen kann, d.h. ein entfernter Rechner kann auf eine SPS über den Kommunikationsport des Treibers zugreifen.

Die folgende Abbildung zeigt die Art dieser Verbindung:



**Natürlich wird der Treiber während des Zugriffs in "externen Bridging" die eigene Kommunikation unterbrechen.**

### Enable

Das ermöglicht Ihnen, die Bridging-Funktion einzuschalten. Wird sie zu "true" gesetzt, schaltet der Treiber den spezifizierten seriellen Port. Falls ein remoter Anruf auftritt (z.B. für Teleservice), unterbricht der Treiber die Kommunikation mit der SPS und lässt den Modem-Port mit dem an der SPS verbundenen Port auf eine transparente Weise einschalten, bis das Schlußcommando (Ausschaltung) betätigt wird, was seinerseits die Kommunikation mit dem Supervisor wiederherstellen lässt.

**Standardmässig ist dieser Dienst ausgeschaltet.**

### Port

Das definiert die Nummer des zu verwendenden seriellen COM-Ports, der für den Bridging-Dienst benutzt wird und mit dem der Modem verbunden ist.



Bemerkung: vergewissern Sie sich, dass keine Konflikte beim Verwenden von Ports in Windows entstehen. Z.B. wenn Sie die Ports Com3 und Com4 installieren, müssen Sie sicher sein, dass die zugewiesene Adresse und die IRQ mit der Konfiguration des Rechners kompatibel sind. Wir empfehlen dazu adressierbare serielle Karten.

### Baudrate

Das definiert die Geschwindigkeit der seriellen Kommunikation (Baud Rate). Sie müssen diejenige Kommunikationsgeschwindigkeit einstellen, die dem Gerät entspricht, mit dem man kommunizieren will.

### Byte Size

Das definiert die Anzahl von Byte, die von dem betreffenden Kommunikationsprotokoll verlangt wird.

### Parity

Das definiert die Parität, die von dem betreffenden Kommunikationsprotokoll verlangt wird.

### Stop Bits

Das definiert die Anzahl von Bit, die von dem betreffenden Kommunikationsprotokoll verlangt wird.

### Flow Control

Das definiert die Datenfluss-Kontrolle für den verwendeten Typ von Kommunikation. Insbesondere erlaubt dies Ihnen, die Kommunikation anzupassen, je nach den Bedürfnissen, welche für den seriellen Port des verbundenen Geräts in niedriger Ebenen erforderlich ist. Standardmässig stellt der Treiber "None" ein, d.h. keine Flusskontrolle, aber je nach Gerät könnte es nötig sein, einen Typ von Flusskontrolle auszuwählen (z.B. falls Fehler mit dem Code "1" benachrichtigt werden).

Die möglichen Optionen sind folgende:

**None:** Keine Flusskontrolle. Die Kontrolle ist nicht vom Protokoll verlangt.

**Hardware:** Die Flusskontrolle wird von elektrischen Signalen aus der seriellen Linie (z.B. RTS, CTS, usw.) gesteuert.

**Xon/Xoff:** Die Flusskontrolle ist vom Typ Xon/Xoff.

### **Rx Queue**

Das definiert die Byte-Anzahl, die von dem seriellen Port als Empfänger-Puffer verwaltet soll. Wenn nicht anders spezifiziert wird der standardmässige Wert 4096 angenommen.

Für Expertbenutzer kann dieser Wert zur Anpassung nach den Bedürfnissen von dem verwendeten System modifiziert werden.

### **Tx Queue**

Das definiert die Byte Anzahl, die von dem seriellen Port als Sender-Puffer verwaltet soll. Wenn nicht anders spezifiziert wird der standardmässige Wert 4096 angenommen.

Für Expertbenutzer kann dieser Wert zur Anpassung nach den Bedürfnissen von dem verwendeten System modifiziert werden.

### **CD Timeout**

Zur Zeit nicht benutzt. Das ist für zukünftige Anwendungen reserviert.

### **CTS Timeout**

Das definiert die Zeit in Millisekunden, nach der der Treiber die Auszeit der Kommunikation für den seriellen CTS-Parameter benachrichtigen soll. Sie definiert die Zeit, innerhalb der die einzelne Schreibe-Aktion zu niedriger Ebenen (API von Windows) über seriellen Port vollendet werden muss.

### **DSR Timeout**

Das definiert die Zeit in Millisekunden, nach der der Treiber die Auszeit der Kommunikation für den seriellen DSR-Parameter benachrichtigen soll. Er definiert das maximale Zeitintervall, das zwischen dem Empfang eines Zeichens und dem nächsten vergeht. Außerdem muss man die maximale Zeitspanne definieren, innerhalb der die einzelne Lese-Aktion zu niedriger Ebenen (API von Windows) über den seriellen Port vollendet werden muss.

### **Display Dialog**

Wenn eingeschaltet (= true) erlaubt dem Supervisor, ein Dialogfenster zum Zeitpunkt der Verbindung in Bridging zu visualisieren, was nun dem lokalen Benutzer ermöglicht, die Verbindung aus der Ferne zu löschen.

### **Disconn.Delay**

Das definiert die Verzögerungszeit in Millisekunden, ab den Empfang des Ausschaltungssignal, um die Schließung der Bridging-Verbindung wirklich zu aktivieren.

### **Connection String**

Das definiert den empfangenen String, der von dem Modem empfangen werden soll, was das Ersuchen einer Fernverbindung bestimmt. Wird dieser String empfangen, aktiviert der Treiber das Ersuchen von Bridging Service.

### **Disconnection String**

Das definiert den String für den Empfang im Modem, die das Gesuch um Beenden der Verbindung bestimmt. Wird diese Zeichenfolge empfangen, deaktiviert der Treiber den Bridging-Dienst und lässt die normale Kommunikation des Treibers wiederherstellen.

### **Init String**

Das definiert den Anfang-String des Modems.

### **OK String**

Das definiert den OK-String des Modems.

## 3.7. TAPI - Einstellungen

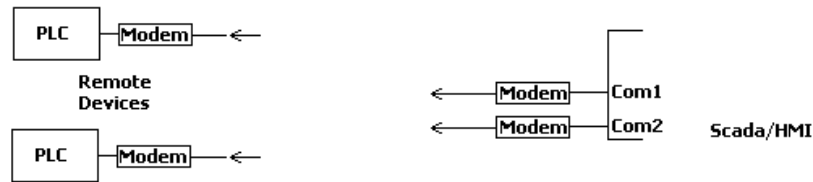
---

In dieser Einstellungsgruppe müssen Sie die Konfigurationen der TAPI-Funktionen des Treiber für die ausgewählte Station definieren.

Die TAPI-Funktionen erlauben dem Treiber, sich mit der entfernten Station über Modem auf eine transparente Weise und automatisch zu verbinden.

Wenn Sie über SPS-Stationen oder entfernte Geräte verfügen, die mit dem Supervisor mit dem Modem verbunden sind, erlauben die TAPI-Funktionen das Kommunikationsprotokoll durchzuführen, nachdem der Modem in Verbindung gesetzt wurde.

Folgende Abbildung zeigt die Art von Verbindung:



Der Modem der SPS muss zum Empfangen konfiguriert werden und bereit sein, die Anrufe des Supervisor empfangen zu können. Der Supervisor wird den Anruf an einen bestimmten Modem für die Stations des betreffenden Treibers automatisch machen, und genau zum Zeitpunkt, in dem die verknüpften Variablen "in Einsatz" werden.

In Verwendung der TAPI-Einstellungen der Station zur Verwaltung der Modem-Kommunikation wird der serielle Port vom Modemtreiber angesteuert. Die Einstellungen, mit denen der serielle Port geöffnet wird, sind in den erweiterten Eigenschaften des Modemtreibers festgelegt, die andere sein können als die in der Station des Movicon-Treibers eingestellten ("Einstellungen des Seriellen Ports").



**Es ist daran zu erinnern, die Ressourcen, welche kontinuierlich ausgeführt werden (wie DataLoggers, Alarmer, Zeitsteuerungen, Allgemeine Logik), lassen die Variablen immer "in Einsatz" halten.**



**Die RAS-Funktionen sind nur dann verfügbar, wenn die Eigenschaft "Aktivieren" belegt wurde.**

Während der Nutzung der TAPI-Verbindung können bei einem fehlgeschlagenen Anruf verschiedene Verhaltensweisen auftreten, die von den in der Station vorgenommenen Einstellungen abhängig sind. Die Kombinationen sind folgende:

1. Die Status-Variablen der Station ("State/Command Variable") wird benutzt und Eigenschaft "Dial Only on Command" aktiviert. Bei einem fehlgeschlagenen Anruf besteht folgendes Verhalten:
  - Das "Bit 1" der Status-Variablen wird auf "True" gesetzt, wenn die programmierte Anzahl der Anruf-Versuche bei erfolgloser Verbindung wurde.
  - Ein Fehler tritt im Ereignisprotokoll auf, zusammen mit der Beschreibung der Ursache, weswegen der Anruf an die Station nicht zustande gekommen ist.
  - Das "Bit 1" der Status-Variablen wird automatisch auf "False" von dem Treiber gesetzt, sobald eine andere Verbindung ausgeführt wird ("Bit 5").
2. Die Status-Variablen der Station ("State/Command Variable") wird benutzt und Eigenschaft "Dial Only on Command" deaktiviert. Bei einem fehlgeschlagenen Anruf besteht folgendes Verhalten:
  - Das Modem startet den Anruf, sobald die Variablen der Station eingesetzt werden (z.B. beim Laden einer Seite).
  - Das "Bit 1" der Status-Variablen wird auf "True" gesetzt, wenn die programmierte Anzahl der Anruf-Versuche bei erfolgloser Verbindung wurde.
  - Ein Fehler tritt im Ereignisprotokoll auf, zusammen mit der Beschreibung der Ursache, weswegen der Anruf an die Station nicht zustande gekommen ist.
  - Wird das "Bit 1" auf "False" gesetzt (die Station wird wieder aktiviert), dann versucht der Treiber die Station wieder anzurufen, sofern noch ausstehende aktive Task nach dieser Station vorhanden sind.
3. Die Status-Variablen ("State/Command Variable") wird nicht in der Station benutzt. Bei einem fehlgeschlagenen Anruf besteht folgendes Verhalten:

- Der Treiber versucht, die Station zu verbinden, solange aktive Tasks für diese Station noch bleiben. Die unmittelbar folgenden Versuche sind in dem Parameter "Versuche" definiert, dann folgt noch, zwischen einer Versuch-Gruppe und der folgenden, eine Wartezeit, die im Parameter "Wartezeit zwischen Versuchen (sec)" konfiguriert ist.

**Phone Number**

Das definiert die Telefonnummer, die gewählt werden soll, mit der das remote Gerät verbunden ist. Bei denjenigen Treibern, die eine Treiberbibliothek ab Build 250 benutzen, können Sie hier eine String-Variable des Supervisor-Systems eingeben. So können Sie in Laufzeit durch Zuweisung einer bestimmten Nummer an diese Variable die genaue Telefonnummer definieren, die gewählt werden soll.

**Versuche**

Dies definiert die Anzahl von Versuche für aufeinander folgende Anrufe (ohne Wartezeit), wenn die Verbindung mit der remoten Station fehlschlägt. Wenn Sie diesen Parameter auf "0" setzen, führt das System nur einen einzigen Verbindungsversuch durch und wartet es dann die Zeit, die im Parameter "Wartezeit zwischen Versuchen (sec)" definiert ist. Anschließend wird nochmals ein einziges Verbindungsversuch durchgeführt.

**Verbindung unterbrechen nach**

Das definiert die Inaktivitätszeit in Millisekunden, bevor die Schließung der Verbindung ausgeführt wird. Es wird verbunden, sobald die betreffenden Variablen im Projekt "in Einsatz" werden. Wenn die betreffenden Variablen nicht mehr in Einsatz sind und die eingestellte Zeit einmal vergangen ist, schließt der Supervisor die Kommunikation.

**Wartezeit zwischen Versuchen (sec)**

Dies definiert die Wartezeit zwischen eine Gruppe von Verbindungsversuchen und der nächsten. Die Anzahl von Anrufe für jede Gruppe ist der Wert, der im Parameter "Versuche" (+ 1) konfiguriert ist. Die maximale Anzahl von Versuche zum Verbinden, die man ausführen kann, ist unbegrenzt. Der Treiber wird somit weiter versuchen, solange die Station aktiv bleibt und die mit ihr verknüpften Variablen weiter im Einsatz bleiben.

**Enable**

Wenn auf "true" gesetzt, lässt das die TAPI-Fernsteuerung Funktion und die Anrufe über Modem mit dem entfernten Gerät einschalten.

**Prompt before connect**

Wenn aktiviert (true) visualisiert das System ein Dialogfenster zur Bestätigung des Anrufs und zur Aktivierung der entfernten Verbindung, jedes Mal wenn das System eine Verbindung über Modem ausführen muss.

**Show Dlg**

Wenn aktiviert (true) visualisiert das System ein Info-Dialogfenster, um den Benutzer über die laufende Verbindung und deren Status zu informieren.

**Dial only on command** (unterstützt nur bei Treibern, die eine Treiberbibliothek ab Version 250 benutzen)

Wird sie auf True gesetzt, wird die Verbindung, durch das Bit 5 der Variable Status/Befehl, nur auf Befehl aktiviert, und durch das entsprechende Bit derselben Variable (Bit 6) getrennt. Für weitere Einzelheiten lesen Sie die Eigenschaft "Variable Status/Befehl". Diese Option ist vom Nutzen in einer Kommunikation, bei der mehrere remoten Stationen über ein einziges Modem kommunizieren, über das die Anrufe durchgeführt werden. Der Projektierer kann einen Treiber für jede Remote Station konfigurieren, und dabei jeden Treiber mit einer verschiedenen Telefonnummer und mit einer verschiedenen Variable Status/Befehl. Somit wird der Bediener in der Lage sein, sich mit verschiedenen (wenngleich immer einzeln) remoten Stationen zu verbinden, indem jeweils die betreffende Variable Status/Befehl eingesetzt wird. Wenn der Datenaustausch mit der aktuellen Station einmal beendet ist, kann dann der Bediener die Kommunikation abrechnen und mit einer anderen Station verbinden.

---

## 3.8. TCP/IP - Einstellungen

---

In dieser Einstellungskarte müssen Sie die Einstellungen der Zugriffsparameter von Ethernet TCP-IP für die ausgewählte Station definieren.



**Server Address**

Das erlaubt Ihnen, die IP-Adresse des Servers oder Netzgeräts einzugeben, mit dem man sich verbinden will.

Beispiel: 192.168.0.1; localhost; Server1

**Server Port**

Das ermöglicht Ihnen, den TCP-Port des Servers oder Geräts einzugeben, mit dem man verbinden will. Der Wert ergänzt die IP-Adresse des Geräts. Z.B. für Modbus TCP-IP muss der Port immer die Nummer 502 sein (gemäß Protokoll vorgesehen), während für andere Geräte müssen Sie die Spezifikationen des Geräts nachschlagen.

**Backup Server Address**

Hier müssen Sie die Serveradresse eingeben. Wird die Adresse als Leerstellen (Standardwert = leere Zeile) gesetzt, versucht der Treiber, sich mit dem Backup-Server zu verbinden, sofern es unmöglich ist, sich mit dem Hauptserver zu kommunizieren. Das geschieht bei Redundanzfall in Treiber-Ebene. Bei Unterbrechung der Kommunikation mit dem Backup-Server wird der Treiber versuchen, sich mit dem Hauptserver wieder zu verbinden usw.

**Switch Server Timeout**

Hier müssen Sie die Zeit in Millisekunden eingeben, die vergehen soll, während eine Kommunikationsfehler an einem Server verifiziert und mit einem anderen Server versucht wird, sich zu verbinden. Standardmässig wird 10000 ms (10 Sekunden) angenommen.

**Local Bound Address**

Das ermöglicht Ihnen, die lokale IP-Adresse einzugeben, d.h. die der Ethernet-Karte des Rechners, die man für die Kommunikation verwenden will. Normalerweise ist nichts spezifizieren, könnte dies aber nötig sein, falls im Rechner mehrere installierten Ethernet-Karten vorliegen. Wird das Feld leer gelassen (wie standardmäßig), nimmt das System die standardmäßig von dem Betriebssystem eingestellte Adresse an.

**Local Bound Port**

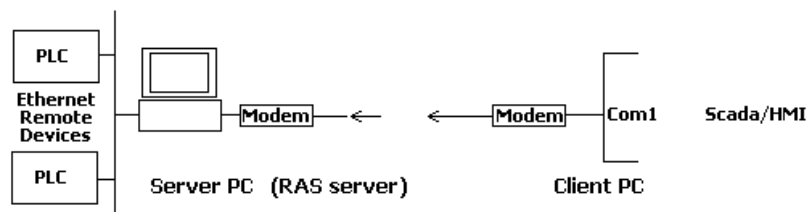
Das ermöglicht Ihnen, die IP-Adresse des lokalen TCP-Ports einzugeben, bezogen auf die Ethernet-Karte des Rechners, die man für die Kommunikation verwenden will. Normalerweise ist nichts spezifizieren, könnte dies aber ein Parameter sein, die von dem Protokoll oder verwendeten Gerät verlangt werden könnte. Wird das Feld leer gelassen (wie standardmäßig), wird die Auswahl des Ports dem Betriebssystem überlassen.

### 3.9. RAS - Einstellungen

In dieser Einstellungsgruppe müssen Sie die Konfigurationen der Parameter für den Zugriff auf remote Geräte über RAS-Funktionen (Remote Access Service) für die ausgewählte Station definieren. Die RAS-Funktionen des Treibers erlauben Ihnen, sich mit einem entfernten Netz über Modem auf eine transparente Weise automatisch zu verbinden.

Wenn man über Netzgeräte verfügt, die mit einem remoten Rechner verbunden sind, ermöglichen die RAS-Funktionen das Protokoll der Netzkommunikation auszuführen, nachdem die Verbindung über Modem durch Zugriff auf einen RAS-Server angefangen hat, d.h. ein Rechner zum Zugriff aufs Netz, das sich über Modem erreichen lässt. Nach die RAS-Verbindung kann der Treiber auf das Netz zugreifen.

Die folgende Abbildung zeigt die Art von Verbindung:



Der Modem des RAS-Servers muss zum Empfangen konfiguriert werden und bereit sein, die Anrufe des Supervisor zu empfangen. Der Supervisor wird den Anruf an einen bestimmten Modem für die Stationen des betreffenden Treibers automatisch durchführen, und genau zum Zeitpunkt, in dem die verknüpften Variablen "in Einsatz" werden.



**Es ist daran zu erinnern, die Ressourcen, welche "kontinuierlich ausgeführt" werden (wie DataLoggers, Alarmer, Befehlszeitsteuerung, Allgemeine Logik), lassen die Variablen immer "in Einsatz" halten.**



**Die RAS-Funktionen sind nur dann verfügbar, wenn die Eigenschaft "Aktivieren" belegt wurde.**

### Dial-up

Das erlaubt Ihnen, eine der vom Betriebssystem definierten RAS-Verbindungen zu spezifizieren. Wenn leer gelassen bittet der Treiber darum, Telefonnummer, Benutzername und Kennwort der Station manuell einzugeben, mit der man verbinden will. Sobald einer der verknüpften Variablen bei dem laufenden Projekt "in Einsatz" gesetzt wird, wird die Verbindung verlangt.

### Phone Number

Das definiert die Telefonnummer der zu verbindenden remoten Station. Bei denjenigen Treibern, die eine Treiberbibliothek ab Build 250 benutzen, können Sie hier eine String-Variable des Supervisors-Systems eingeben. So können Sie in Laufzeit durch Zuweisung einer bestimmten Nummer an diese Variable die genaue Telefonnummer definieren, die gewählt werden soll.

### User Name

Das gibt den Benutzernamen (Username) für den Zugang auf die entfernte Station ein.

### Password

Das gibt das Kennwort für den Zugang auf die entfernte Station ein.

### Retries

Das definiert die maximale Anzahl von Verbindungsversuchen, falls die Verbindung fehlerhaft ist. Ist diese Versuchenanzahl erreicht und die Kommunikation noch nicht gesetzt, wird der Fehler angezeigt.

### Disconnect After

Das definiert die Inaktivitätszeit in Millisekunden, bevor die Schließung der Verbindung ausgeführt wird. Es wird verbunden, sobald die betreffenden Variablen im Projekt "in Einsatz" werden. Wenn die betreffenden Variablen nicht mehr in Einsatz sind und die eingestellte Zeit einmal vergangen ist, schließt der Supervisor die Kommunikation.

### Retry Hold Time (sec)

Das definiert die Zeitspanne in Millisekunden, die vergehen soll, bevor die Verbindung wieder versucht wird.

### Enable

Wenn zu "true" gesetzt lässt das die TAPI-Fernsteuerung Funktion und die Anrufe über Modem mit dem entfernten Gerät einschalten.

### Prompt before connect

Wenn aktiviert (true) visualisiert das System ein Dialogfenster zur Bestätigung für die Durchführung des Anrufs und zur Aktivierung der entfernten Verbindung, jedes Mal wenn das System Verbindungen über Modem ausführen muss.

### Show Dlg

Wenn aktiviert (true) visualisiert das System ein Info-Dialogfenster, um den Benutzer über die laufende Verbindung und deren Status zu informieren.

**Dial only on command** (Dies wird nur von denjenigen Treibern unterstützt, welche die Treiber-Bibliothek ab Build 250 benutzen)

Wenn Sie diese Eigenschaft aktivieren, wird die Modem-Verbindung nur auf Befehl durch Einsatz des entsprechenden Bit der Variable Status/Befehl (Bit 5) aktiviert. Beendet wird die Kommunikation durch das entsprechende Bit derselben Variable (Bit 6). Für Einzelheiten lesen Sie bitte Variable Status/Befehl. Diese Option ist vom Nutzen in einer Kommunikation, bei der mehrere remoten Stationen über ein einziges Modem kommunizieren, mit dem die Anrufe durchgeführt werden. Der Projektierer kann einen Treiber für jede Remote Station konfigurieren, und dabei jeden Treiber mit

einer verschiedenen Telefonnummer und mit einer verschiedenen Variable Status/Befehl konfiguriert. Somit wird der Bediener in der Lage sein, sich mit verschiedenen (wenngleich immer einzeln) remoten Stationen zu verbinden, indem jeweils die betreffende Variable Status/Befehl eingesetzt wird. Wenn der Datenaustausch mit der aktuellen Station einmal beendet ist, kann dann der Bediener die Kommunikation abbrechen und nun mit einer anderen Station verbinden.

## **3.10. Sonderkonfigurationen (TAPI und RAS)**

---

Die RAS- TAPI-Funktionen werden von dem Treiber benutzt, um die Supervision von Projekten auszuführen, welche eine Fernsteuerung erfordern. Dabei sollten Sie besonderes Augenmerk auf einige Aspekte schenken.

Die gewöhnlichen Fernsteuerungs-Architekturen sind folgende:

1. Eine Remote Station, und ein Supervisions-Projekt, das mit der Station über Modem kommuniziert.
2. Mehrere Remote Stationen, die in verschiedene Zonen stehen, und ein Supervisions-Projekt, das mit den Stationen über ein einziges Modem kommuniziert.
3. Mehrere Remote Stationen, die sich in dem gleichen Netzwerk stehen, und ein Supervisions-Projekt, das mit den Stationen über ein einziges Modem kommuniziert.

Die möglichen Lösungen für eine optimale Kommunikation sind folgende:

1. Sie können eine Station in dem Treiber definieren. In diesem Fall kann das Anrufmanagement automatisch (der Treiber beginnt einen Anruf, sobald die Variable eingesetzt wird) oder manuell ausgeführt werden, unter Anwendung der Variable Status/Befehl.
2. Sie können eine oder mehrere Stationen definieren, und jede von diesen mit ihrer entsprechenden Variable Status/Befehl. Bei einem solchen Fall müssen die Anrufe durch die Variable Status/Befehl verwaltet werden, um gleichzeitige Anrufe zu vermeiden.
3. Sie können eine oder mehrere Stationen definieren, jede von diesen mit der entsprechenden Variable Status/Befehl, aber nur eine Station kann für den RAS- oder TAP-Anruf eingesetzt werden. Wenn die Verbindung mit der Hauptstation einmal hergestellt ist, kann die Kommunikation mit den übrigen Stationen durch Einsatz des Bit 1 der verknüpften Variablen Status/Befehl aktiviert werden.

Um weitere Einzelheiten über die Einstellungen der Variablen Status/Befehl lesen Sie den Abschnitt unter Allgemeines (Station)



# 4. Einstellungen von Tasks

---

## 4.1. Tasks

---

In diesem Abschnitt müssen Sie die "**Kommunikationsaufgaben**" (**Tasks**) einfügen und einstellen, wenn Sie diese Art von Kommunikation benutzen wollen.



Es ist daran zu erinnern, dass der Supervisor die Möglichkeit bietet, die Treiberkommunikation sowohl über "dynamische Aufgaben" als auch über "**Tasks**" einzustellen (sie sind in entsprechenden Abschnitten beschrieben). Die dynamischen Aufgaben werden automatisch von dem Treiber beim Starten des Projekts erstellt, in Funktion der Links von Adressen des Geräts, die in den Eigenschaften "Dynamische Tasks" jeder einzelnen Tag definiert werden.

Die Kommunikationsaufgaben erlauben Ihnen, "**statische**" Aufgaben zum Treiber zuzuweisen, die nach den vorgesehenen Geräten in Polling ausgeführt werden.



Wenn Sie die Kommunikation in Tasks wählen, müssen Sie die Daten-Verknüpfungen zwischen den Adressen des Geräts und den Supervisor-Variablen statisch zu definieren.

### Add

Die Schaltfläche "Add" erlaubt Ihnen, eine neue "Aufgabe" (Task) für den Kommunikationstreiber hinzuzufügen. Wird sie hinzugefügt, wird auf dem entsprechenden Einstellungsfenster der verlangten Kommunikationsparameter visualisiert.

Die eingefügten Parameter können nun mithilfe der unten beschriebenen Schaltflächen bearbeitet oder entfernt werden.

### Edit

Die Schaltfläche "Edit" erlaubt Ihnen, Parameter einer vorher eingefügten "Aufgabe" zu bearbeiten. Dafür müssen Sie die gewünschte Aufgabe auswählen und dann die Schaltfläche "Edit" oder Doppelklick benutzen.

### Remove

Die Schaltfläche "Remove" erlaubt Ihnen, eine vorher eingefügten "Aufgabe" zu entfernen. Dafür müssen Sie die gewünschte Aufgabe auswählen und dann die Schaltfläche "Remove" benutzen.

## 4.2. Allgemeine Eigenschaften der Statischen Tasks

---

In diese Einstellungskarte müssen Sie die Eigenschaften der Gruppe "Allgemeinen" für die ausgewählte Task definieren.

### Station

Das ist der Name zur Identifikation der Station, welche dem Gerät entspricht, mit dem man kommunizieren will. Die Stationen (mindestens eine) müssen in der entsprechenden Einstellungskarte definiert werden.

### Task Name

Das ist der Name, welcher die Task bei der Einstellungsphase identifiziert. Sie können hier eine beliebige Zeichenfolge für die Task definieren.

Jede Task aus der Liste muss einen eindeutigen Namen besitzen.

## Variables

Dies erlaubt Ihnen, die Supervisors-Variable(n) zum Management der Task zu verknüpfen. Mithilfe der rechts stehenden Auswahl-Schaltfläche können Sie irgendeine Variable wählen, die zuvor in der Echtzeit-DB eingefügt wurde, und ggf. eine neue Variable erstellen. Die eingefügte Variable wird automatisch in die Echtzeit-Datenbank hinzugefügt. Die Variablen **müssen in dem Feld mit dem ";"-Zeichen getrennt werden, und sollten am Gerät in fortlaufender Folge ab dem Wert liegen, welcher in der Eigenschaft "Address" definiert wurde.**



**WICHTIG: Es wird dringend empfohlen, eine Liste von Variablen des gleichen Typs zu benutzen, um unvorhersehbare Fehler zu vermeiden. Für komplexe Datenstrukturen am Gerät (deren Elemente auch von verschiedenen Typen sein können), können Sie sich von Datenstrukturen bedienen, die in der Echtzeit-Datenbank definiert sind, und direkt die Strukturvariable mit dem Gerät über eine statische Task verbinden.**

Der Treiber wird die zu lesende/schreibende Anzahl von Bytes automatisch berechnen, basierend auf der Adresse, die für das Gerät definiert ist, wobei die Größe jeder in der Liste vorhandenen Variable nach der Definition der Variablen in der Echtzeit-DB des Projekts hinzugefügt wird.

Der Treiber wird weitere Kontrollen in Funktion der eingegebenen Adresse durchführen.

### Adressierung auf Bit

Wenn man auf Bit-Adressen am Gerät zugreifen muss, alle in der Eigenschaft "Variables" definierten Variablen müssen in dem Projekt als "Bit" deklariert werden. Ansonsten wird eine Fehlermeldung visualisiert. Die Anzahl von Bits, die mit dem Gerät ausgetauscht werden muss, ist gleich der Anzahl von Variablen, die in der Liste "Variables" deklariert wurde.

Die Startadresse ist jene Adresse, die in der Eigenschaft "Start Address" deklariert ist.

### Adressierung auf Byte, Word o Double Word

Die Variablen, die in der Eigenschaft "Variables" eingefügt sind, müssen nicht vom Typ Bit in der Echtzeit-DB des Projekts deklariert werden. Ansonsten wird eine Fehlermeldung visualisiert.

## Conditional Variable

Dies erlaubt Ihnen, eine Variable des Projekts zu verknüpfen, deren Status die bedingte Ausführen der Kommunikationstask bestimmen wird.

Mit der rechts stehenden Auswahl-Taste können Sie eine beliebige Variable auswählen, die zuvor in die Ressource Echtzeit-Datenbank des Projekts hinzugefügt wurde.

Die Variable (von einem beliebigen Typ) wird somit die Ausführung der Task bedingen: Wird sie auf einem "von Null verschiedenen" Wert gesetzt ( $>0$ ), wird die Kommunikationstask von dem Treiber ausgeführt.



**Nach der wirklichen Ausführung der Task wird der Treiber die verknüpfte Variable automatisch auf Null setzen. Deshalb ist dies beim Verwalten der Variablen mit eventuellen Logischen Anweisungen besonders zu berücksichtigen.**

## Polling Time

Dieser Parameter (angegeben in Millisekunden) definiert die minimale Polling-Zeit der Ausführung einer einzelnen Task, um die Daten zu aktualisieren **wenn die Variablen im Einsatz sind.**

Die defaultmäßige Polling Time, die bei der Erstellung der Task gesetzt wird, wird von dem in der (Allgemeinen) Eigenschaft ("Polling Time") festgelegte Wert definiert, lässt sich aber später nach Belieben für jede Task einzeln modifizieren.

Man kann einen noch höheren Wert konfigurieren, wenn es bspw. keiner zu großen Frequenz zur Aktualisierung von Daten bedarf.

## Unused Polling Time

Dieser Parameter (angegeben in Millisekunden) gestattet es, eine Aktualisierung der Daten einer Einzeltask auch dann zu forzieren, **wenn die Variablen nicht im Einsatz sind**, wobei **ein Polling-Zeit festgelegt wird.**

Die defaultmäßige Polling Time, die bei der Erstellung der Task gesetzt wird, wird von dem in der (Allgemeinen) Eigenschaft ("Unused Polling Time"), festgelegte Wert definiert, lässt sich aber später nach Belieben für jede Task einzeln modifizieren.

Wird dieser Wert zu 10 000 (= 10 Sekunden) eingestellt, wird die Task mit einer Minimalzeit von 10 Sekunden ausgeführt, und zwar auch wenn deren Variablen nicht im Einsatz sind.



**Wenn Sie diesen Parameter auf Null konfigurieren, wenn die Variablen nicht im Einsatz sind, werden die Tasks nie ausgeführt. Ist dieser Wert von Null verschieden, werden alle Variablen, die nicht im Einsatz sind und deren Task vom Typ Input/Output ist, zum Lesen mit dieser Frequenz aktualisiert. Dieser Mechanismus könnte bei einer hohen Anzahl von Variablen die Kommunikation verlangsamen.**

**Address offset variable** (Dies wird nur von denjenigen Treibern unterstützt, welche die Treiber-Bibliothek ab dem Build 250 benutzen)

In diesem Feld können Sie den Namen einer in der Echtzeit-Datenbank definierten Variable vom Typ Integer eingeben. Der Wert dieser Variable kann in Laufzeit modifiziert werden. Er wird als Positiver bzw. Negativer Offset benutzt, je nach dem erreichten Wert, bezüglich der Anfangsadresse, die in der Task konfiguriert wurde. Auf diese Weise können Sie in verschiedenen Stellen des Speichers des Geräts lesen, indem Sie einfach diesen Wert in Laufzeit ändern.

Die Offset-Einheit hängt von dem Treiber sowie von dem Adressierungstyp:

- Bei bitweisen Adressierungs-Typen, ein Offsets-Wert gleich 1 entspricht einer Verschiebung auf Bit.
- Bei byteweisen Adressierungs-Typen, ein Offsets-Wert gleich 1 entspricht einem Byte
- Bei wortweisen Adressierungs-Typen, ein Offsets-Wert gleich 1 entspricht einem Word

Der in der Eigenschaft Address Offset eingegebene Wert kann je nach dem eingesetzten Treiber verschiedene Bedeutungen haben. Bei einem Siemens-Treiber, wenn eine Word-Variable gelesen werden soll, müssen Sie einen Offset-Wert von +2 oder von einem Vielfachen von 2 wählen, weil die Adressierung bei Siemens byteweise ausgeführt wird. Bei einem Modbus-Treiber müssen Sie für das Lesen einer Variable vom Typ Word indes einen Offset-Wert von +1 oder einem Vielfachen von 1 wählen. Modbus lässt die Adressierung tatsächlich wortweise ausführen.

**Beachte: Auch durch Änderung des Offsets-Werts werden die Tasks nicht sofort ausgeführt, sondern nur in geeigneten Zeitpunkten. Um die Gewissheit darüber zu haben, dass eine Task wirklich ausgeführt wurde, können Sie eine bedingte Variable benutzen. So wird bspw. eine Exception-Output-Task mit dem neuen Offset nur dann ausgeführt, wenn der Wert der verknüpften Variable verändert wird.**

Beispiel: Um in verschiedenen Bereichen des Speichers des Geräts zu schreiben, kann man eine Task eines Unconditional Output mit einer Bedingungs-Variable und einer Offset-Variable benutzen. Wenn der Offset-Wert und der zu schreibende Wert einmal modifiziert sind, es genügt einfach die Bedingungs-Variable auf 1 zu setzen. Geht die Bedingungs-Variable wieder zu Null, hat man die Gewissheit, dass die Aufgabe wirklich ausgeführt wurde. Nachher kann man auf die nächste Adressierung gehen.

#### Swap Byte

Diese Option gestattet es, bei Daten vom Typ "Word", die Bytes zueinander zu invertieren. So ergibt sich, dass die Daten zwischen dem Supervisor und dem Gerät das "hohes" Byte zu dem "niedrigen" Byte (und umgekehrt) invertiert werden, für alle Daten vom Typ Word, die der Task gehören.

#### Swap Word

Diese Option gestattet es, bei Daten vom Typ "Double Word", die Wörter zueinander zu invertieren. So ergibt sich, dass die Daten zwischen dem Supervisor und dem Gerät das "hohes" Wort zu dem "niedrigen" Wort (und umgekehrt) invertiert werden. Dies wird für alle Daten vom Typ "Double" im Task ausgeführt.

#### Type

In dieser Option müssen Sie den Typ von Ausführung konfigurieren, welche der Task zugewiesen werden soll.

Die vorhandenen Möglichkeiten sind folgende:

- |                     |   |
|---------------------|---|
| <b>Input</b>        | Diese Option definiert den Typ von Task als "Nur Lese". Auf diese Weise, wenn die Variablen im Einsatz sind, wird der Treiber das Polling beim Lesen an den für das betreffende Gerät verknüpften Adressen ausführen, und alle die betreffenden Variablen des Projekt übertragen. |
| <b>Input/Output</b> | Diese Option definiert den Typ von Task als "Lese-Schreibe". Auf diese Weise, wenn die Variablen im Einsatz sind, wird der Treiber das Polling beim Lesen an den für das betreffende Gerät verknüpften Adressen ausführen, und alle die betreffenden                              |

Variablen des Projekt übertragen. Wird der Wert einer Variable von dem Supervisor geändert, wird der Treiber das Schreiben der Daten an dem Gerät ausführen und nachher in Lesen zurückkehren.

**Exception Output** Diese Option definiert den Typ von Task als "Nur Schreibe", wobei das Schreiben nur bei Ausnahme ausgeführt wird, d.h. geschrieben wird nur dann, wenn irgendeine Änderung in den Daten hervorgerufen wird.

**Unconditional Output** Diese Option definiert den Typ von Task als "Nur Schreibe". Das Schreiben der Daten wird stets ausgeführt, und ungeachtet der Tatsache, dass die Daten ändern.

**Write Outputs at Startup**

Diese Eigenschaft wird nur bei einer Eingang/Ausgang-Task (bzw. einer "Exception Output"-Task) verwendet. Wenn auf "True" gesetzt, wird die Task in Ausgang bei dem Start des Projekts ausgeführt.

**Device Data**

Die Parameter dieser Gruppe bestimmen die Adressierung der Speicherbereiche des betreffenden Geräts.

**Lesen Sie bitte die spezifische Dokumentation des Protokolls.**

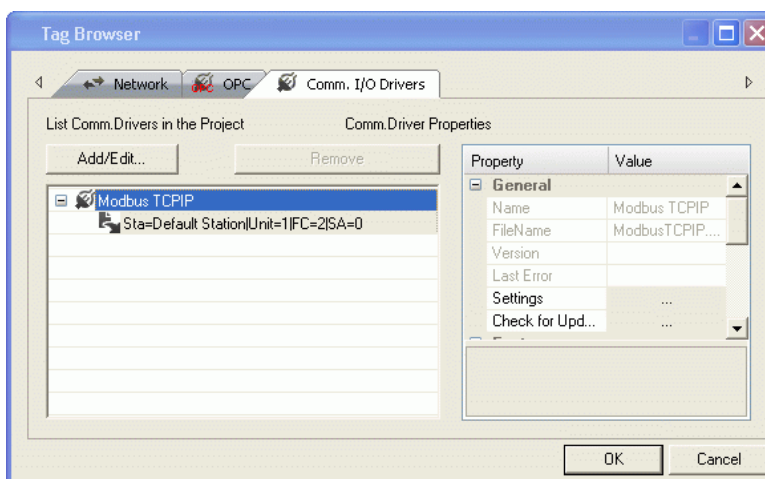
## 4.3. Dynamische Tasks

Die Dynamische Tasks werden bei dem Start des Projekts, basierend auf der in der Eigenschaft "Dynamisch" (Physische E/A-Adresse) eingegebene dynamische Adresse für jede Variable, automatisch von dem Treiber erstellt.

Die Einstellungen für die dynamischen Tasks werden nicht durch den Befehl "Treibereinstellungen" konfiguriert, sondern spezifisch für jede Einzeltask am Fenster "Variablen Explorer" definiert.

### Einfügung der "Physischen E/A-Adresse" einer Variable

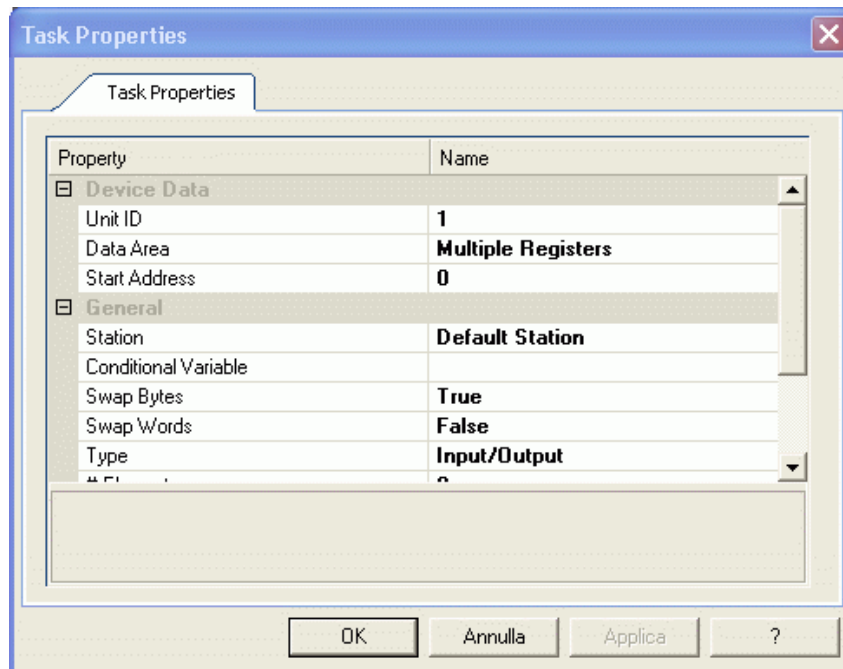
Um die dynamischen Eigenschaften der Variablen zu konfigurieren, müssen Sie die Liste der Variablen aus der Echtzeit-Datenbank am Fenster Projekt Explorer auswählen, dann die betreffende Variable wählen und anschließend das Fenster Eigenschaften visualisieren. Mithilfe von der Auswahl-Schaltfläche, die sich rechts der Eigenschaft "Dynamisch" (Physische E/A-Adresse" befindet, öffnen Sie das Fenster "Variablen Explorer".



Fenster Variablen Explorer



Am Fenster Variablen Explorer wählen Sie die Registerkarte "Kommunikationstreiber". Auf diesem Fenster wird die Liste der im Projekt eingefügten Treiber visualisiert, und für jeden Treiber die bereits definierten dynamischen Links aufgelistet. Zudem gibt es eine Schaltfläche "Hinzufügen/Bearbeiten...", mit der Sie das Fenster "Task Properties" öffnen können, um die dynamischen Links zu bearbeiten, und eine Schaltfläche "Entfernen", mit der die markierten dynamischen Links entfernt werden können.



Fenster Task Properties.

Einmal geöffnet, können Sie am Fenster "Variablen Explorer" die Einfügung/Bearbeitung eines dynamischen Links folgendermaßen ausführen:

#### Einfügung eines neuen dynamischen Links

1. In der Eigenschaft "Dynamisch" (Physische E/A-Adresse) einer Variable öffnen Sie das Fenster "Variablen Explorer".
2. Wählen Sie den Namen des Treibers, den Sie benutzen wollen.
3. Klicken Sie auf die Schaltfläche "Hinzufügen/Bearbeiten...", um das Fenster "Task Properties" zu öffnen.
4. Wenn die Variable mit keinem dynamischen Link verknüpft war, wird das Fenster "Task Properties" mit den Standardwerten geöffnet. Hatte die Variable bereits ihren verknüpften dynamischen Link, öffnet das Fenster "Task Properties" mit den Werten dieses Links. Nun konfigurieren Sie die betreffenden Parameter.
5. Schließen Sie das Fenster "Task Properties" mit "OK". Auf diese Weise wird die dynamische Adresse mit den konfigurierten Werten aufgebaut, und der neue Link in die Liste der dynamischen Adressen hinzugefügt, wobei die Links fokussiert werden.
6. Schließen Sie das Fenster "Variablen Explorer" mit OK, damit der dynamische Link in die Variablen-Eigenschaft "Dynamisch" eingefügt wird.

#### Bearbeitung eines vorhandenen dynamischen Links

1. In der Eigenschaft "Dynamisch" (Physische E/A-Adresse) einer Variable öffnen Sie das Fenster "Variablen Explorer".
2. Wählen Sie einen dynamischen Link aus der Liste der Treiber aus, und klicken Sie auf die Schaltfläche "Hinzufügen/Bearbeiten...", um das Fenster "Task Properties" zu öffnen.
3. Wenn die Variable mit keinem dynamischen Link verknüpft war, wird das Fenster "Task Properties" mit den Standardwerten geöffnet. Hatte die Variable bereits ihren verknüpften dynamischen Link, öffnet das Fenster "Task Properties" mit den Werten dieses Links. Nun konfigurieren Sie die betreffenden Parameter.
4. Schließen Sie das Fenster "Task Properties" mit "OK". Auf diese Weise wird die dynamische Adresse mit den konfigurierten Werten aufgebaut, und der neue Link in die Liste der dynamischen Adressen hinzugefügt, wobei die Links fokussiert werden.

5. Schließen Sie das Fenster "Variablen Explorer" mit OK, damit der dynamische Link in die Variablen-Eigenschaft "Dynamisch" eingefügt wird.

**Auswahl eines vorhandenen dynamischen Link**

1. In der Eigenschaft "Dynamisch" (Physische E/A-Adresse) einer Variable öffnen Sie das Fenster "Variablen Explorer".
2. Wählen Sie einen dynamischen Link aus der Liste der Treiber.
3. Schließen Sie das Fenster "Variablen Explorer" mit OK, damit der dynamische Link in die Variablen-Eigenschaft "Dynamisch" eingefügt wird.

Es ist anzumerken, dass die Liste der dynamischen Links eines Treibers, die am Fenster "Variablen Explorer" visualisiert sind, lediglich eine Hilfe-Funktion für den Programmierer darstellen. Wurde der Link mit der Variable tatsächlich verknüpft, kann der Link aus der Liste entfernt werden, ohne dass Fehler im Projekt auftreten. Die Liste der dynamischen Links werden in der Datei ".dyndrv" des Treibers und wird im Ordner "RESOURCES" des Projekts abgelegt. Wird diese Datei gelöscht, wird die Liste der dynamischen Adressen nicht mehr zur Verfügung sein, welche bis diesen Zeitpunkt eingefügt waren. In der Eigenschaft "Dynamisch" der Variablen wird jedoch diejenige dynamische Adresse gespeichert bleiben, die zuvor verknüpft wurde.

**Eigenschaften dynamischer Tasks**

---

Anhand des Fensters "Task Properties" kann man die Einstellungen des dynamischen Links der Variable definieren. Mit diesem Fenster können Sie die allgemeinen Eigenschaften der Task (Gruppe "Allgemeine") konfigurieren sowie die spezifische Eigenschaften für die Adressierung des Geräts, welche von Gerät zu Gerät abweichen. Jetzt wird eine Beschreibung der Allgemeine Eigenschaften der Task angegeben. Für eine ausführliche Beschreibung über die Adressierung des Geräts, lesen Sie bitte das Handbuch des betreffenden Treibers.

**Station**

Standardwert = leer. Das ist der Name zur Identifikation des Geräts, mit dem man kommuniziert. Die Stationen (mindestens eine) müssen in der betreffende Einstellungskarte definiert werden. Werden mehrere Stationen definiert, müssen Sie in diesem Feld die betreffende Station, für welche die Task ausgeführt werden soll.

**Conditional Variable**

Standardwert = leer. Dies erlaubt Ihnen, eine Variable des Projekts zu verknüpfen, deren Status die bedingte Ausführung der Kommunikations-Task bestimmen wird. Mithilfe von der rechts stehenden Schaltfläche können Sie irgendeine Variable auswählen, die bereits in der Ressource "Echzeit-Datenbank" hinzugefügt wurde, sowie neue Variablen erstellen. Die Variable (von jedem Typ) wird somit die Ausführung der Task bestimmen: Wird sie auf einen "von Null verschiedenen" Wert gesetzt (><0), wird die Kommunikationstask von dem Treiber ausgeführt. Einmal auf den Wert Null wieder gesetzt, signalisiert dies, dass die Task ausgeführt wurde.

**Swap Byte**

Das ist ein Standardwert, der bei jedem Treiber spezifisch ist. Diese Option gestattet es, bei Daten vom Typ "Word", die Bytes zueinander zu invertieren. So ergibt sich, dass die Daten zwischen dem Supervisor und dem Gerät das "hohes" Wort zu dem "niedrigen" Wort (und umgekehrt) invertiert werden. Defaultsmäßig ist dies auf "False" konfiguriert, und Sie müssen diese Konfiguration nur dann ändern, wenn das Gerät die normale Reihenfolge der Bytes in den Daten invertiert. Diese Eigenschaft ist nur bei einigen Treiber verfügbar.

**Swap Word**

Das ist ein Standardwert, der bei jedem Treiber spezifisch ist. Diese Option gestattet es, bei Daten vom Typ "Double Word", die Wörter zueinander zu invertieren. So ergibt sich, dass die Daten zwischen dem Supervisor und dem Gerät das "hohes" Wort zu dem "niedrigen" Wort (und umgekehrt) invertiert werden. Dies wird für alle Daten vom Typ "Double" im Task ausgeführt. Defaultsmäßig ist dies auf "False" konfiguriert, und Sie müssen diese Konfiguration nur dann ändern, wenn das Gerät die normale Reihenfolge der Wörter in den Daten invertiert. Diese Eigenschaft ist nur bei einigen Treiber verfügbar.

**Type**

In dieser Option müssen Sie den Typ von Ausführung konfigurieren, welche der Task zugewiesen werden soll. Die vorhandenen Möglichkeiten sind folgende:

<b>Input</b>	Diese Option definiert den Typ von Task als "Nur Lese". Auf diese Weise, wenn die Variablen im Einsatz sind, wird der Treiber das Polling beim Lesen an den für das betreffende Gerät verknüpften Adressen ausführen, und alle die betreffenden Variablen des Projekt übertragen.
<b>Input/Output</b>	Diese Option definiert den Typ von Task als "Lese-Schreibe". Auf diese Weise, wenn die Variablen im Einsatz sind, wird der Treiber das Polling beim Lesen an den für das betreffende Gerät verknüpften Adressen ausführen, und alle die betreffenden Variablen des Projekt übertragen. Wird der Wert einer Variable von dem Supervisor geändert, wird der Treiber das Schreiben der Daten an dem Gerät ausführen und nachher in Lesen zurückkehren.
<b>Exception Output</b>	Diese Option definiert den Typ von Task als "Nur Schreibe", wobei das Schreiben nur bei Ausnahme ausgeführt wird, d.h. geschrieben wird nur dann, wenn irgendeine Änderung in den Daten hervorgerufen wird.
<b>Unconditional Output</b>	Diese Option definiert den Typ von Task als "Nur Schreibe". Das Schreiben der Daten wird stets ausgeführt, und ungeachtet der Tatsache, dass die Daten ändern.

**#Elements**

Standardwert = leer. Dies definiert die Anzahl der zu lesenden/schreibenden Elemente. Die mögliche Werte sind 0 oder 1.

Wird er auf Null belassen, berechnet der Treiber automatisch die Anzahl von Bytes in Funktion des Typ von Variable, die in der Echtzeit-Datenbank definiert ist.

Ist die verknüpfte Variable eine vom Typ Word, werden, für die Task, 2 Bytes ab der definierten Adresse angenommen.

Und umgekehrt: Ist die #Elements gleich 1, wird der Treiber, ungeachtet des für die verknüpften Variable definierten Typs, automatisch die Anzahl von Bytes in Funktion des Adressierungs-Typs berechnen. Wenn bspw. der im Gerät adressierte Bereich auf Word ist, dann werden 2 Bytes gelesen/geschrieben, ungeachtet der Tatsache, dass die Stützvariable vom Typ Word ist.

**Write Outputs at Startup**

Standardwert = False. Diese Eigenschaft wird nur bei einer Eingang/Ausgang-Task (bzw. einer "Exception Output"-Task) verwendet. Wenn auf "True" gesetzt, wird die Task in Ausgang bei dem Start des Projekts ausgeführt.



# 5. Importierung der Datenbank des Geräts

## 5.1. Import der Datenbank des Geräts

Dank dieser wichtigen Funktionalität ist es möglich, direkt auf die SPS- oder Gerät-Datenbank (oder auf eine entsprechende Datenquelle) zuzugreifen, um die gewünschten Variablen (Tags) in das Projekt importieren zu können.

**Wenn die Operation vollendet ist, wird die Echtzeitdatenbank des Projekts geladen, wobei alle die importierten Variablen eingefügt werden. Diese Variablen sind als Dynamische Tags in "Nicht Gemeinsamen" Bereich, zusammen mit dem entsprechenden Variablen-Typ und sowie mit der bereits von dem Gerät zugewiesenen Adresse definiert.**

Bei Betätigung des Befehls wird das Fenster zur Auswahl der "Datenquelle" visualisiert, d.h. die SPS-Datenbank, die Symbole-Datei oder die .CSV-Datei, die man aus der Exportierung der Daten aus der SPS oder Gerät empfangen hat.

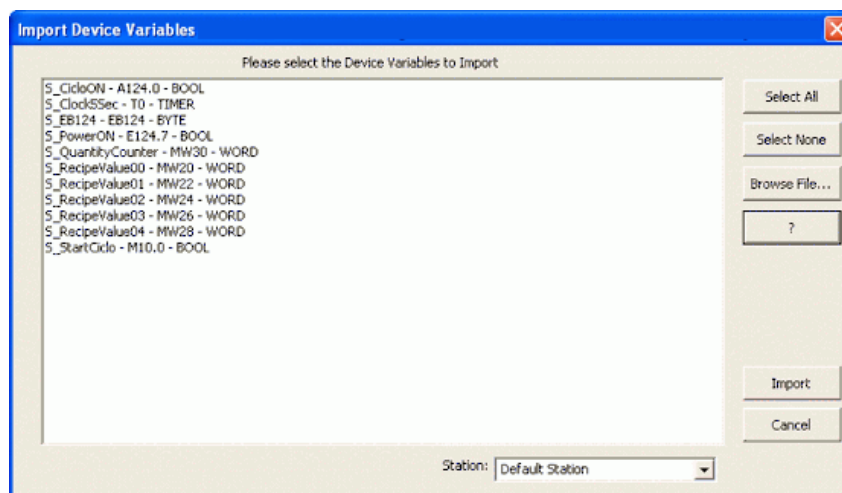
Sie müssen deshalb diese Datei auswählen. Dann erscheint das unten abgebildete Fenster, das die eingeschlossenen Variablen darstellt.



**Achtung: Die Importierung der Daten aus SPS wird von allen Treibern und von den wichtigsten Geräten unterstützt. Sie müssen die Zugriffsmodalitäten und Anforderungen der "Datenquelle" überprüfen, die von Gerät zu Gerät verschieden sein können.**



Werden beim Datenimport Variablen erfasst, die bereits in der Echtzeitdatenbank des Projektes vorhanden sind, werden diese automatisch ohne Bestätigungsaufforderung überschrieben. Dabei werden die Eigenschaften "Typ" und "Dynamik" überschrieben, während die Eigenschaft "Variablenbeschreibung" unverändert belassen wird. Die Eigenschaft "Variablenbeschreibung" wird nur importiert, wenn die Variable nicht in der Echtzeitdatenbank des Projektes vorhanden ist.



**Bei der Auswahl einer hohen Anzahl von zu importierenden Variablen kann der Import einige Zeit brauchen. Das Verfahren kann über die Taste "ESC" abgebrochen werden.**

Die Struktur- und Array-Variablen können über das Import-Tool der Datenbank des Gerätes auf zwei Weisen importiert werden:

Wird im Tool-Dialogfenster der Stamm der Strukturvariable gewählt und die Importfunktion über die Schaltfläche "Import" ausgeführt, werden im Projekt ein Prototyp, bestehend aus den Strukturelementen, und eine Variable derselben Art des Prototyps erstellt. So kann eine Mehrfachauswahl von Strukturvariablen importiert werden. Dieselbe Funktion kann auch über die Auswahlsequenz der Schaltfläche "Select All" und anschließend über die Schaltfläche "Import" ausgeführt werden.

Werden ein oder mehrere Elemente einer oder mehrerer Strukturen gewählt und anschließend importiert, so werden die Variablen vom Typ der gewählten Elemente erstellt, nicht jedoch die Strukturprototypen. Dasselbe Ergebnis lässt sich durch Anklicken der Schaltfläche "Expand All" erzielen, um die Elemente der Strukturen zu durchsuchen; anschließend wird auf die Schaltfläche "Select All" für die Auswahl aller Elemente und schließlich auf die Schaltfläche "Import" für das Importieren der einzelnen Elemente als verschiedene Variablen geklickt.

Sind in der Liste der Gerätevariablen sowohl einzelne Variablen als auch Strukturvariablen vorhanden, hat die Schaltfläche "Select All" die Wirkung, sie alle auszuwählen.

Wird - in Bezug auf den Import von Array-Variablen - der Array-Stamm gewählt und importiert, wird im Projekt eine Array-Variable mit der Eigenschaft "Typ der Elemente" erstellt, die im Array-Typ eingestellt ist. Dieselbe Funktion kann auch über die Auswahlsequenz der Schaltfläche "Select All" und anschließend über die Schaltfläche "Import" ausgeführt werden.

Werden ein oder mehrere Elemente eines oder mehrerer Arrays gewählt und anschließend importiert, werden die Variablen vom Typ der gewählten Elemente erstellt, nicht jedoch die Array-Variablen. Dasselbe Ergebnis lässt sich durch Anklicken der Schaltfläche "Expand All" erzielen, um die Elemente der Arrays zu durchsuchen; anschließend wird auf die Schaltfläche "Select All" für die Auswahl aller Elemente und schließlich auf die Schaltfläche "Import" für das Importieren der einzelnen Elemente als verschiedene Variablen geklickt.

Sind in der Liste der Gerätevariablen sowohl einzelne Variablen als auch Array-Variablen vorhanden, hat die Schaltfläche "Select All" die Wirkung, sie alle auszuwählen.

## Select All

Das erlaubt Ihnen, alle Variablen der Datei zum Import auszuwählen.

Falls nur partiell ausgewählt wird, müssen Sie die Tastenkombination CTRL+Klick oder SHIFT+Klick benutzen.

## Select None

Das erlaubt Ihnen, alle Variablen der Datei zum Import abzuwählen.

Falls etwas nur partiell abgewählt wird, müssen Sie die Tastenkombination CTRL+Klick oder SHIFT+Klick benutzen.

## Browse File...

Das erlaubt Ihnen, die Quelle-Datei zu ändern, wobei das Standardfenster zur Auswahl der Dateien aktiviert wird.

## ? (help)

Das aktiviert die Visualisierung des Führers, der alle entsprechenden Informationen bezüglich der Anforderungen für das Format der Daten enthält.

## Import

Das lässt die Importierung der Variablen aus der Quelle-Datei (Datenquelle des Geräts) in das Supervisorprojekt ausführen. Ist die Importierung vollendet, wird die Ressource Echtzeitdatenbank mit allen importierten Variablen bereichert.



**Da die "Datenquellen" vom Gerät abhängen und diese ändern können, wird dabei empfohlen, die Eigenschaften der importierten Variablen immer zu überprüfen. Sie müssen die richtige Ausführung des automatischen Parsings, die Zuweisung des Typs und Adresse des Geräts nach dem Importierungsprozess überprüfen.**

## Cancel

Diese Schaltfläche löscht die Importierungsaktion.

## Station:

Dieses Kästchen erlaubt Ihnen, die Station des Treibers auszuwählen, welche den importierten Variablen zugewiesen wird, falls dem Treiber mehrere Stationen definiert wurden.

# 6. Fehler

## 6.1. Beschreibung der Fehler

Die Fehler sind unten nach Bereich eingruppiert nach alphabetische Reihenfolge aufgelistet.

<b>Entwicklung</b>				
<b>Fehlermeldung</b>	<b>Phase</b>	<b>Schweregrad</b>	<b>Mögliche Ursache</b>	<b>Mögliche Lösung</b>
Internal Error in file '<file_name>' at the line '<line_number>'. Please contact the Technical Support	Entwicklung	ernst	Fehler im Code.	Rufen Sie den Support Center an.
Invalid Address Offset Variable. Enter the name of an existing integer variable	Entwicklung	nur informativ	Die ausgewählte Variable entspricht nicht dem richtigen Typ.	Geben sie den Namen einer Variable vom Typ Integer aus der EchtzeitDB
Invalid conditional variable '<variable_name>' for the task '<task_name>'	Entwicklung	nur informativ	Die ausgewählte Variable ist ungültig: Existiert nicht, wurde gelöscht, oder entspricht nicht dem richtigen Typ.	Geben sie den Namen einer Variable vom Typ Integer aus der EchtzeitDB
Invalid conditional variable. Enter only one existing variable of numeric type.	Laufzeit	nur informativ	Die ausgewählte Variable ist ungültig: Existiert nicht, wurde gelöscht, oder entspricht nicht dem richtigen Typ.	Geben sie den Namen einer Variable vom Typ Integer aus der EchtzeitDB
ListView out of memory	Entwicklung	ernst	Unzureichende Speicher zum Management der Task-Liste oder der Stationen.	Sie müssen Speicher freigeben. Schließen Sie eventuell Prozesse oder Applikationen, die nicht benötigt werden, und das Supervisorsystem (und wenn nötig auch den PC) neu starten.
Name '<name>' is already in use. Please, choose a different name	Entwicklung	nur informativ	Der Stations- oder Task-Name wurde bereits vergeben. Doppelnamen sind unzulässig.	Ändern Sie den Namen der Station oder der Task.
Task size exceeds the maximum allowed size for this protocol	Laufzeit	nur informativ	Die Größe der Task übersteigt die von dem Protokoll maximal zugelassene Größe für das Paket.	Reduzieren Sie die mit der Task verknüpften Variablen.
The selected Dynamic Settings String is either invalid or not allowed for this Variable!	Entwicklung	nur informativ	Die dynamische Adresse für die Variable ist ungültig oder	Überprüfen Sie das Format der dynamischen Adresse, oder den Variablentyp.

The selected COM port is either not supported or is being used by another application. Please select another port.	Entwicklung	nur informativ	unzulässig. Die in der Eigenschaft Station/SerialPort konfigurierte Port-Nummer ist ungültig	Für die Eigenschaft Station/SerialPort settings/Port sind nur ganzzahlige positive Nummern zugelassen (Null auch nicht)
Value cannot be empty! Please enter a value	Entwicklung	nur informativ	Nur für Ethernet-Treiber. Die in der Eigenschaft TCPIP Station/Server Port eingegebenen Werte sind ungültig.	Geben Sie eine gültige Adresse ein: IP-Adresse im Format xxx.xxx.xxx.xxx , oder irgendeinen Namen eines Servers, z.B. 'server1' Geben Sie eine Portnummer: ganzzahlig, positiv (Der Wert Null ist unzulässig)
Value cannot be zero! Please enter a number greater then 0	Entwicklung	nur informativ	Der Wert für die Eigenschaft 'Rx queue size' oder 'Tx queue size in Station/Queue size' ist ungültig	Geben Sie einen ganzzahligen, positiven Wert ein (Der Wert Null ist unzulässig)
Variable '<variable_name>' is not valid for the task '<task_name>'	Laufzeit	nur informativ	Die ausgewählte Variable ist ungültig: Existiert nicht, wurde gelöscht, oder entspricht nicht dem richtigen Typ.	Geben sie den Namen einer Variable vom Typ Integer aus der EchtzeitDB ein. Überprüfen Sie die Taskeinstellungen und die Adressierungsregeln.

**Validierung von Adressen**

Fehlermeldung	Phase	Schweregrad	Mögliche Ursache	Mögliche Lösung
Job <job_name> (station <station_name>) is invalid and cannot be promoted to the protocol mng	Laufzeit	ernst	Die Task ist nicht gültig und wird nicht ausgeführt.	Überprüfen Sie die Konfiguration der Task
Setting of addresses for fields of the structure <structure_name> interrupted on unsupported field <field_name>	Laufzeit	nur informativ	Fehler beim Management der dynamischen Adressierung der Elemente einer Struktur (z.B. Strings). Ein Element wird nicht unterstützt: Ausgetauscht werden nur die Elemente, welche sich (innerhalb der betreffenden Struktur) vor dem fehlerhaften Element befinden.	Ändern Sie die Struktur oder benutzen Sie einfache Variablen für Typen wie z.B. Strings.
The Task <task_name> for the Station <station_name> is invalid	Laufzeit	ernst	Die Task ist nicht gültig und wird nicht ausgeführt.	Überprüfen Sie die Parameter der Kommunikation der Task.

**Hardware**

Fehlermeldung	Phase	Schweregrad	Mögliche Ursache	Mögliche Lösung
Error loading <drivername>.dll	Laufzeit	unbehebbarer Fehler	DLL-Datei des Treibers nicht gefunden	Überprüfen Sie, dass die Datei <nomedriver>.dll bereits im Unterordner 'Drivers' vorhanden ist. Dieser Unterordner befindet sich innerhalb des Installationsverzeichnis des Supervisors.



<b>Kommunikation</b>				
<b>Fehlermeldung</b>	<b>Phase</b>	<b>Schweregrad</b>	<b>Mögliche Ursache</b>	<b>Mögliche Lösung</b>
Communication error : station <station_name>, error <error_description>	Laufzeit	unbestimmt	Das ist eine allgemeine Meldung für die Kommunikation: Die Beschreibung des Fehlers gibt eine ausführlichere Erklärung des Fehlers.	Überprüfen Sie die möglichen Fehler anhand der in der Meldung beinhalteten Informationen.
Error ! Driver has already been initialized !	Laufzeit	unbehebbarer Fehler	Unerwarteter Vorgang. Man hat eine Initialisierung des Treibers unternommen, wenn dieser bereits in Ausführung war.	Kontaktieren Sie den Support.
Failed to open the communication device. The configured COM port is either not supported or is being used by another application	Kommunikations-Test	ernst	Der Versuch zum Öffnen des seriellen Ports (nur für serielle Treiber) oder eines Sockets ist fehlgeschlagen.	Überprüfen Sie die Nummer des Ports und den korrekten Kontakt der Kabel. Vergewissern Sie sich, dass der Port nicht bereits von einer anderen Applikation verwendet wird. Überprüfen Sie den Zugang des Geräts.

<b>Kommunikationsmeldungen</b>				
<b>Fehlermeldung</b>	<b>Phase</b>	<b>Schweregrad</b>	<b>Mögliche Ursache</b>	<b>Mögliche Lösung</b>
Communication established : station <station_name>	Laufzeit	Information	Die Kommunikation wurde korrekt hergestellt.	
Communication restored	Laufzeit	Information	Die Kommunikation wurde nach einer Kommunikationspanne wieder hergestellt.	
Station quality changed : station <station_name>, quality bitfield <quality_value>	Laufzeit (Debug-Fenster)	Information	Die Qualität der in der Station vorhandenen Daten ist geändert. Die mögliche Werte sind folgende: bad, good, uncertain, not connected.	
Task <task_name> (Station <station_name>) is now in use...	Laufzeit (Debug-Fenster)	Information	Wenn das Management der Variablen Im Einsatz ist aktiv, werden die dynamischen Tasks nur dann aktiv sein, wenn die Variablen im Einsatz sind.	
Task <task_name> (Station <station_name>) is now NOT in use...	Laufzeit (Debug-Fenster)	Information	Wenn das Management der Variablen Im Einsatz ist aktiv, werden die dynamischen Tasks NICHT aktiv sein, wenn die Variablen NICHT im Einsatz sind.	
The station <station_name> has switched from server <primary_server_name> to <secondary_server_name>	Laufzeit	Information	Spezifisch für Ethernet-Treiber. Man hat einen Backup-Server in der Eigenschaft Network services/Redundancy, und der Primärserver ist nicht verfügbar. Die Station wird von der Backup-Server und umgekehrt.	

<b>Serielle Kommunikation (Nur für Serielle Treiber)</b>				
<b>Fehlermeldung</b>	<b>Phase</b>	<b>Schweregrad</b>	<b>Mögliche Ursache</b>	<b>Mögliche Lösung</b>
Break Detect !	Laufzeit	ernst	Die Linie ist unterbrochen, und es gibt deshalb keine Kommunikation.	Überprüfen die Verkabelung und die Einstellungen der Geräte, an der Applikation und an dem Gerät selbst.
Carrier Detect Timeout !	Laufzeit	ernst	Der Treiber hat keine Antwort während der vorgesehenen Zeit erhalten (timeout )	Überprüfen die Verkabelung und die Einstellungen der Geräte, an der Applikation und an den Geräten selbst.
Clear To Send Timeout !	Laufzeit	ernst	Der Treiber hat keine Antwort während der vorgesehenen Zeit erhalten (timeout )	Überprüfen die Verkabelung und die Einstellungen der Geräte, an der Applikation und an den Geräten selbst.
Data Set Ready Timeout !	Laufzeit	ernst	Der Treiber hat keine Antwort während der vorgesehenen Zeit erhalten (timeout )	Überprüfen die Verkabelung und die Einstellungen der Geräte, an der Applikation und an den Geräten selbst.
FATAL ERROR! The Windows message queue is full! Serial Comm messages lost! Increase the message queue depth.	Laufzeit	unbehebbarer Fehler	Die Queue der Meldungen ist voll. Sie müssen die Größe der Queue anpassen (einige Meldungen sind verloren gegangen)	Vergrößern Sie die Queue der Meldungen.
Frame Error !	Laufzeit	ernst	Rausch in der Kommunikationslinie.	Entfernen Sie die mögliche Ursache der Geräusche und überprüfen Sie die das Parität-Bit bzw. das Stop-Bit.
Overrun Error !	Laufzeit	ernst	Daten angekommen, bevor das vorherige Paket empfangen wurde. Das Betriebssystem ist überlastet.	Reduzieren Sie den Wert des baud rate, überprüfen Sie die Hardware, schließen Sie eventuelle Applikationen, welche die Ressource des Systems stark in Anspruch nehmen.
Parity Error !	Laufzeit	ernst	Rausch in der Kommunikationslinie.	Entfernen Sie die mögliche Ursache der Geräusche und überprüfen Sie die das Parität-Bit bzw. das Stop-Bit.
Rx Timeout !	Laufzeit	ernst	Der Treiber hat keine Antwort während der vorgesehenen Zeit erhalten (timeout )	Überprüfen die Verkabelung und die Einstellungen der Geräte, an der Applikation und an dem Gerät selbst. Überprüfen Sie, ob die Einstellungen des Converters RS232/RS485 existieren.
Rx Buffer Overflow !	Laufzeit	ernst	Der Empfangspuffer ist voll. Mögliche Datenverluste.	Stellen Sie die Eigenschaft Station/Queue size/Rx nach oben ein, aktivieren Sie die Eigenschaft Station/serial port settings/Flow control oder hardware flow control, wenn vom Gerät unterstützt.
The selected COM port is either not supported or is being used by another	Laufzeit	ernst	Der COM-Port existiert nicht oder wird nun von einer anderen Applikation verwendet.	Überprüfen Sie die Nummer des Ports in Station/SerialPort settings/Port; Sie müssen

application. Please select another port.				den Port freigeben. Das gelingt durch Schließen der Applikation, welche den Port verwendet und (falls nötig) den PC neu starten.
Tx Buffer Full !	Laufzeit	ernst	Der Ausgangspuffer ist voll, die Meldungen im Ausgang werden nicht gesendet.	Stellen Sie die Eigenschaft Station/Queue size/Tx nach oben ein
Unexpected WM_QUIT received !	Laufzeit	Information	Der Treiber hat eine Schluß-Meldung von dem Betriebssystem empfangen, und wird geschlossen.	

**TAPI (Nur für Serielle Treiber)**

<b>Fehlermeldung</b>	<b>Phase</b>	<b>Schweregrad</b>	<b>Mögliche Ursache</b>	<b>Mögliche Lösung</b>
Call to <phone_number> has been hung up	Laufzeit	Information	Die Kommunikation wurde von dem Anrufer unterbrochen.	
Disconnected: Bad Address	Laufzeit	ernst	Die Adresse ist ungültig	Überprüfen Sie die zu wählende Telefonnummer.
Disconnected: Busy	Laufzeit	ernst	Die remote Station ist besetzt	Warten, bis die Remote Station frei ist.
Disconnected: Congestion	Laufzeit	ernst	Das Netz ist besetzt. Deshalb kann der Anruf nicht zustande kommen.	Warten, bis die Linie frei ist.
Disconnected: Incompatible	Laufzeit	ernst	Man hat eine Inkompatibilität bezüglich der aufgeforderten Anrufs festgestellt.	Überprüfen Sie die Hardware der remoten Station
Disconnected: Local phone picked up	Laufzeit	ernst	Die Kommunikation wurde von irgendeinem unterbrochen.	
Disconnected: No Answer	Laufzeit	ernst	Der Anruf wurde durchgeführt, man hat aber keine Antwort bekommen.	
Disconnected: No Dial Tone	Laufzeit	ernst	Kein Ton in der Linie.	Überprüfen Sie die Integrität der Verbindungen und der Linie
Disconnected: Unknown reason	Laufzeit	ernst	Der Anruf wurde aus unbekanntem Ursachen unterbrochen.	
Disconnected: Unreachable	Laufzeit	ernst	Die Telefonnummer ist nicht erreichbar.	
Line busy	Laufzeit	nur informativ	Man versucht anzurufen, obwohl die Leitung besetzt ist.	Warten, bis die Leitung frei ist.
Placing call to <phone_number> ...	Laufzeit	Information	Gibt die Nummer an, die man versucht, anzurufen.	
Remote Party Disconnected	Laufzeit	ernst	Die Kommunikation wurde vom dem Angerufenen beendet.	
Remote Party rejected call	Laufzeit	ernst	Der Anruf wurde von dem Angerufenen abgelehnt.	
TAPI in special information state, probably couldn't dial number	Laufzeit	ernst	Die Verbindung ist fehlgeschlagen.	
Unable to find modem	Laufzeit	ernst	Das Modem konnte nicht in dem definierten Port	In der Systemsteuerung überprüfen, ob dem Modem der richtige Port

associated with <port>		gefunden werden.	zugewiesen wurde; ebenfalls die Einstellungen und Verkabelung des Modems kontrollieren.
---------------------------	--	------------------	---

**Bridging (Nur für Serielle Treiber)**

Fehlermeldung	Phase	Schweregrad	Mögliche Ursache	Mögliche Lösung
A modem has been connected to the Bridging Port, station <station_name>	Laufzeit	Information		
A remote device or computer has taken control through the Bridging Port! - station <station_name>	Laufzeit	Information	Die Bridging-Operation wurde gestartet.	
Connection window creation failed : station <station_name>	Laufzeit oder Kommunikations-Test	ernst	Möglicherweise Speicher für Ressourcen unzureichend.	Überprüfen Sie den für die Ressource verfügbaren Speicher.
Could not initialise the Modem on the Bridging Port, station <station_name>	Laufzeit	ernst	Der Befehl zur Initialisierung wurde gesendet, man hat aber keine korrekte Antwort bekommen.	Überprüfen den Zustand des Modems, Einstellungen und Verkabelung.
Could not perform read operation on the Bridging Port, station <station_name>	Laufzeit	ernst	Unmöglich, Daten zu lesen.	Überprüfen den Zustand des Modems, Einstellungen und Verkabelung.
Could not perform write operation on the Bridging Port, station <station_name>	Laufzeit	ernst	Unmöglich, Daten zu schreiben.	Überprüfen den Zustand des Modems, Einstellungen und Verkabelung.
Device control has been restored - Bridging ended! - station	Laufzeit	Information	Bridging-Operation beendet.	
Disconnection sequence received on the Bridging Port, station	Laufzeit	Information	Es gibt keine Bridging-Operation, deshalb wird die Kommunikation unterbrochen.	
The modem has been disconnected from the Bridging Port, station <station_name>	Laufzeit	Information		
Too many characters received on the Bridging Port, station	Laufzeit	ernst	Es wurden mehr Zeichen als maximal zugelassen (interne Größe des Puffers).	Bridging-Operation zurücksetzen.

**RAS (Nur für Ethernet-Treiber)**

Fehlermeldung	Phase	Schweregrad	Mögliche Ursache	Mögliche Lösung
Connection window creation failed : station <station_name>	Laufzeit oder Kommunikations-Test	ernst	Interner Fehler: Möglicherweise Speicher für Ressourcen unzureichend.	Überprüfen Sie den verfügbaren Speicher für die Ressourcen.
RAS Dial Error : <error-	Laufzeit oder	ernst	Überprüfen Sie den	Überprüfen Sie

description>, Station <station_name>	Kommunikations- Test		spezifischen Fehler.	den spezifischen Fehler.
Station <station_name> is about to dial-up. Continue ?	Laufzeit	Information	Verbindungsaufforderung. Diese Meldung wird visualisiert, wenn die Eigenschaft 'Prompt before connect' Station\RAS ist aktiviert.	
Undefined RAS Dial Error <error_number>, Station<station_name>	Laufzeit	ernst	Unbekannter Fehler in der RAS-Verbindung	Überprüfen Sie die Parameter RAS in den Eigenschaften der Station, die Verkabelung und die RAS- Einstellungen des angerufenen PC.

## 6.2. Über

Mithilfe dieses Fenster können Sie die **Treiberversion** verifizieren, indem man die eventuellen Kommentare und entsprechenden Beschreibungen auswählt.



Es ist ratsam zu überprüfen, dass die vorhandene Version der Datei des Treibers von Supervisor immer die letzte Version ist. Sollte diese Datei nicht die zuletzt zur Verfügung gestellte sein, wird empfohlen, die .DLL-Datei zu aktualisieren, die sich aus der Website nach vorgesehenen Modalitäten herunterladen lässt.



Movicon ist eine HMI/SCADA-Software für das Betriebssystem Microsoft Win32.  
Movicon ist von Progea entwickelt und hergestellt. © 1992 – 2008 Alle Rechte vorbehalten  
Kein Teil dieser Dokumentation oder des Programms darf ohne die schriftliche Erlaubnis von Progea in irgendeiner Form reproduziert oder übertragen werden.  
Informationen in diesem Dokument können ohne Ankündigung geändert werden und sind nicht für den Hersteller bindend.

INDUSTRIAL AUTOMATION SOFTWARE  
**progea**  
Via S.Anna, 88/E  
41100 Modena - Italy  
Tel. +39 059 451060  
Fax +39 059 451061  
Email: info@progea.com  
Http://www.progea.com

INDUSTRIAL AUTOMATION SOFTWARE  
**progea**  
Via XX Settembre, 30  
Tecnocity Alto Milanese  
20025 Legnano (MI) Italy  
Tel. +39 0331 486653  
Fax +39 0331 455179  
Email: willems@progea.com

INDUSTRIAL AUTOMATION SOFTWARE  
**progea**  
Progea Deutschland GmbH  
Marie-Curie-Str. 12  
D-78048 VS-Villingen  
Tel: +49 (0) 7721 / 99 25 992  
Fax: +49 (0) 7721 / 99 25 993  
info@progea.de

INDUSTRIAL AUTOMATION SOFTWARE  
**progea**  
Progea International Ltd  
via Penate 16  
6850 Mendrisio - Switzerland  
tel +41 (91) 9676610  
fax +41 (91) 9676611  
international@progea.com

INDUSTRIAL AUTOMATION SOFTWARE  
**progea**  
Progea USA LLC  
2800 East Enterprise Avenue  
Appleton, WI 54914  
Tel. +1 (888) 305 2999  
Fax. +1 (920) 257 4213  
info@progea.us