ToddySoft Connect Modbus Driver - Ignition 8.3

Ignition-8.3-Modul zur Anbindung von **Modbus-TCP**-Geräten — SPSen, Zählern, Antrieben, I/O-Blöcken und Gateways. Coils, Discrete Inputs, Input Register und Holding Register werden direkt adressiert, mit einem Typsuffix, das dem Treiber sagt, wie die Bytes zu dekodieren sind.

Die Gerätekonfiguration verwendet ein eigenes Konfigurationsformular: Einstellungen, die zur getroffenen Auswahl nicht passen, bleiben Ihnen aus dem Weg, und Felder mit sicherheitsrelevanter Wirkung sagen das dort, wo Sie sie lesen.

> **Modbus kennt keine Tag-Liste — das ist eine Eigenschaft des Protokolls und keine Lücke im Treiber.** Auf der Leitung zählt nichts Adressen auf; Adressen werden daher von Hand aus der Modbus-Tabelle des Geräts übernommen. Alle anderen ToddySoft-Connect-Module durchsuchen eine Steuerung nach ihren Symbolen; dieses kann es nicht, und keine Einstellung ändert daran etwas. Geräte kündigen sich auch nicht an, lassen sich aber finden: siehe Geräte im Netzwerk finden.

Installation

> **Dieses Modul ist ausschließlich für Ignition 8.3.** Es wird gegen das Ignition-8.3-SDK gebaut und läuft auf keiner anderen Hauptversion. Für Ignition 8.1 installieren Sie stattdessen den separaten `…-8.1.modl`-Build — die beiden sind nicht austauschbar, und jedes Gateway benötigt den zu ihm passenden Build.
>
> **Ignition 8.0 und älter werden nicht unterstützt und können es auch nicht.** Das Modul baut auf der Gerätetreiber-API auf, die Ignition mit 8.1 eingeführt hat; unter 8.0 scheitert das Laden mit `NoClassDefFoundError: … opcua.server.api.AbstractDeviceModuleHook`.
>
> Beide Builds tragen in der Modulliste des Gateways denselben Namen. Unterscheiden Sie sie anhand der installierten Datei oder anhand der Modulbeschreibung, die die Ignition-Version nennt.
>
> **Das OPC-UA-Modul von Ignition muss installiert sein und laufen.** Dies ist ein OPC-UA-Gerätetreiber: Er registriert sich am OPC-UA-Server von Ignition, und jeder Tag, den er bereitstellt, wird über diesen ausgeliefert. Fehlt das Modul, kann das Gateway das vorliegende nicht laden; im Log häufen sich dann `ClassNotFoundException` bzw. `NoClassDefFoundError` zu Klassen aus `com.inductiveautomation.opcua…`. Das OPC-UA-Modul gehört zum Lieferumfang von Ignition und ist standardmäßig installiert; praktisch relevant wird das daher nur auf Gateways, auf denen es entfernt oder deaktiviert wurde. Dieses Modul deklariert die Abhängigkeit als **required** — ein Mechanismus, den 8.3 besitzt und 8.1 nicht. Ignition versucht daher gar nicht erst, es zu laden, und nennt stattdessen den Grund, statt mit einem Stacktrace zu scheitern. Unter **Diagnostics > Logs** erscheint:
>
> `Module "com.toddysoft.connect.…" depends on "com.inductiveautomation.opcua", which is either not installed or inactive on this gateway.`
>
> Die Treiber von Inductive Automation werden in derselben Situation genauso behandelt.

1. Öffnen Sie die Weboberfläche des Ignition Gateways und navigieren Sie zu **Platform > System > Modules**.
2. Klicken Sie auf **Install or Upgrade Module**.
3. Wählen Sie die Datei `ToddySoft-Connect-Modbus-Driver-8.3.modl` aus und laden Sie sie hoch.
4. Bestätigen Sie die Lizenz- bzw. Zertifikatsabfrage, um die Installation fortzusetzen.

5. **Starten Sie das Gateway neu.** Ignition 8.3 lädt ein neu installiertes oder aktualisiertes Modul nicht im laufenden Betrieb: Die Modulliste zeigt es als *pending restart*, und ein Banner **Gateway Restart Required** unten auf der Modulseite bietet **Restart Gateway** an. Mehrere Module lassen sich zuerst installieren oder aktualisieren und dann mit einem Neustart übernehmen. Planen Sie ihn ein: Der Neustart unterbricht kurz alle Geräte und Clients des Gateways.

6. Prüfen Sie nach dem Neustart, dass das Modul in der Modulliste als **ACTIVE** erscheint.

Vorbereitung: Was Sie über das Gerät wissen müssen

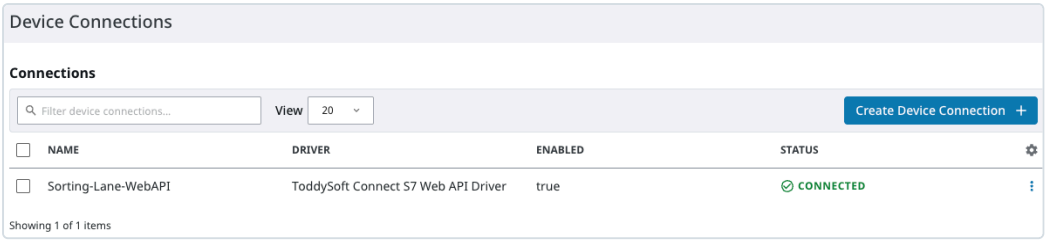
- **Seine Modbus-Tabelle.** Welche Register was enthalten und in welcher Darstellung. Das Gerät lässt sich nicht danach fragen; diese Angaben stammen aus seiner Dokumentation.
- **Seinen Unit-Identifier** (Slave-ID), falls er nicht `1` ist.
- **Seine Byte-Reihenfolge** für alles, was breiter als ein Register ist. Modbus legt die Reihenfolge der Bytes innerhalb eines Registers fest (höherwertiges Byte zuerst), nicht aber die der Register eines breiteren Werts — Hersteller unterscheiden sich hier. Siehe *Default Byte Order*.
- **TCP-Port 502** muss vom Gateway-Host aus erreichbar sein, sofern das Gerät nicht woanders lauscht.

Es gibt keine Zugangsdaten: Modbus ist nicht authentifiziert. Alles im Netz, was den Port erreicht, kann lesen und schreiben — das ist bei der Frage, wo das Gateway steht, zu bedenken.

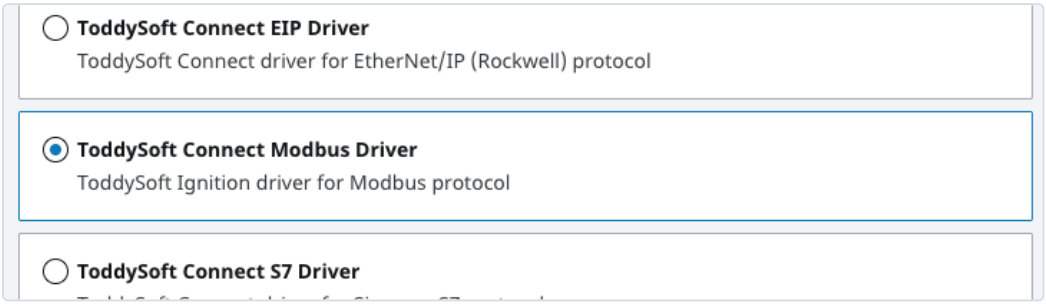
> **Dieses Modul spricht ausschließlich Modbus TCP.** Unter Ignition 8.3 heißen sowohl das Modul als auch der Gerätetyp *ToddySoft Connect Modbus Driver*; unter Ignition 8.1 schreibt der Gerätetyp es als *Modbus TCP Driver* aus. Die Verbindung ist in beiden Fällen dieselbe. Der zugrunde liegende Treiber implementiert zwar auch Modbus RTU und ASCII über serielle Schnittstellen, UDP und TLS, doch das Ignition-Modul erzeugt eine `modbus-tcp://`-Verbindung und bietet keine Transportauswahl; diese Varianten sind von hier aus also nicht erreichbar. Ein serielles Gerät benötigt ein vorgeschaltetes Modbus-Gateway.

Erstellen einer Geräteverbindung (Device Connection)

1. Gehen Sie zu **Connections > Devices > Connections**.

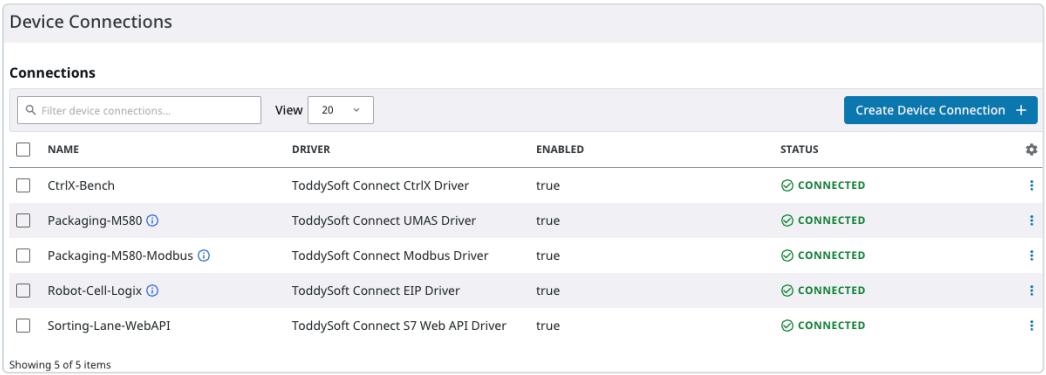


2. Klicken Sie auf **Create Device Connection** und wählen Sie **ToddySoft Connect Modbus Driver** als Gerätetyp.



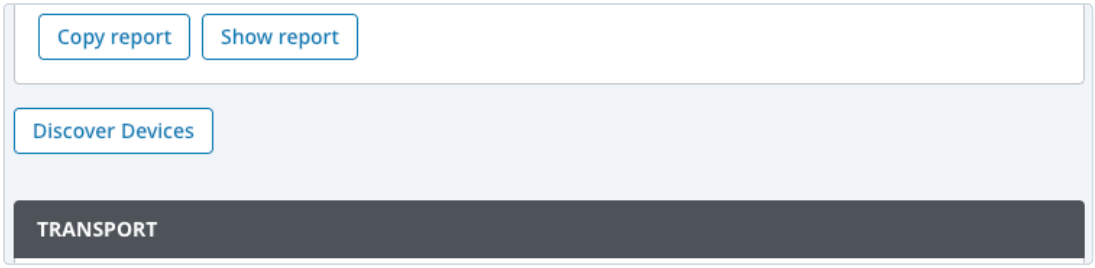
3. Konfigurieren Sie die Verbindungseinstellungen (siehe die Abschnitte unten) und klicken Sie auf **Create Device Connection**.

4. Die neue Verbindung erscheint anschließend in der Liste der Device Connections.

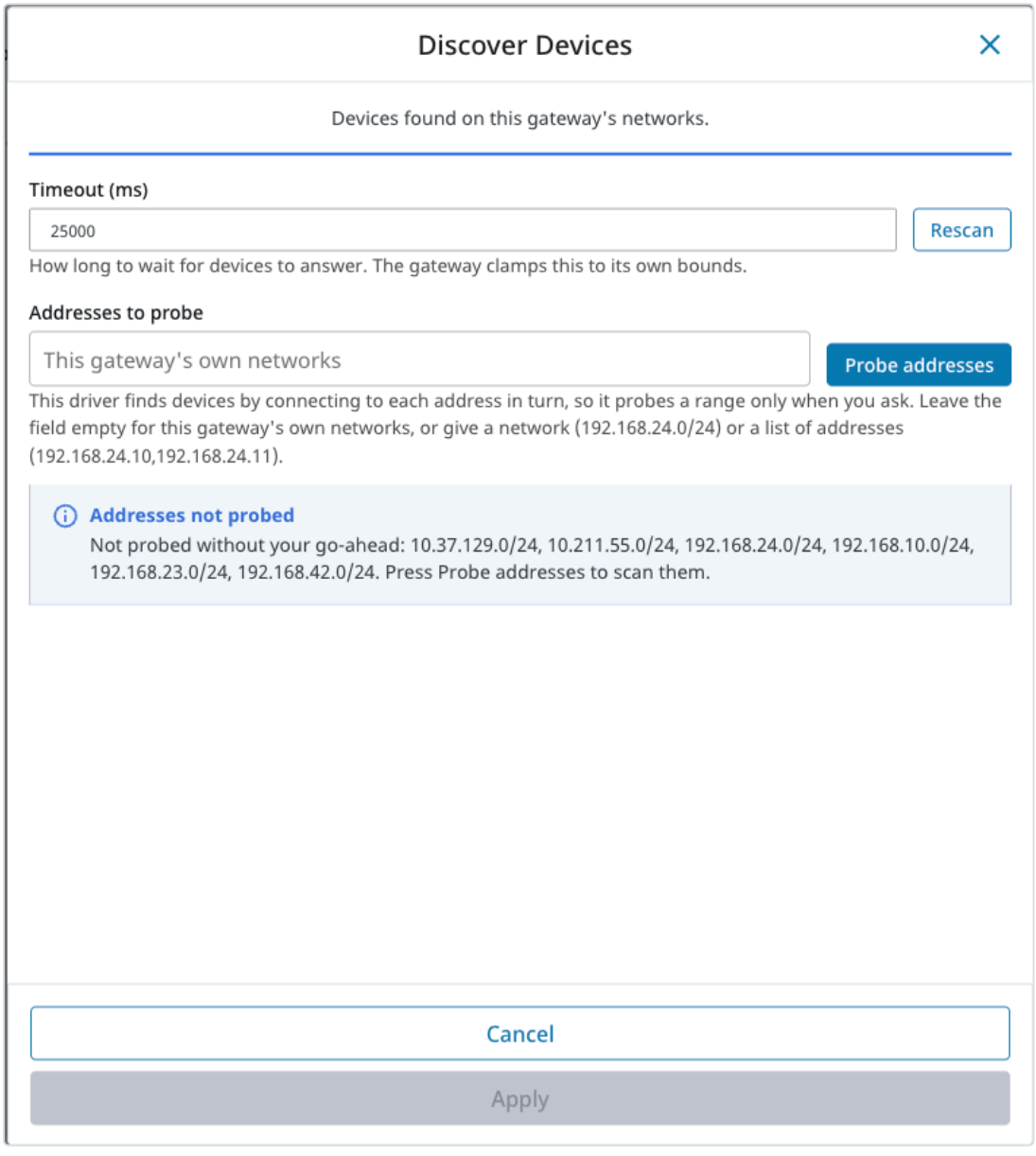


Geräte im Netzwerk finden

Das Konfigurationsformular enthält eine Schaltfläche **Discover Devices**. Modbus-Geräte kündigen sich nicht an, daher funktioniert die Suche anders als bei den Modulen mit Broadcast: Das Gateway **verbindet sich nacheinander mit jeder Adresse** auf Port 502 und bittet, was antwortet, sich auszuweisen. Aufgeführt wird nur ein Gerät, das wie ein Modbus-Server antwortet; eine Adresse mit offenem Port 502, die kein Modbus spricht, erscheint nicht.



Weil das bedeutet, jede Adresse eines Netzes anzusprechen, **wird nichts abgefragt, bevor Sie es erlauben.** Die erste Suche nach dem Öffnen des Dialogs fragt keinen Bereich ab und nennt unter **Addresses not probed** die Netze, die sie ausgelassen hat:

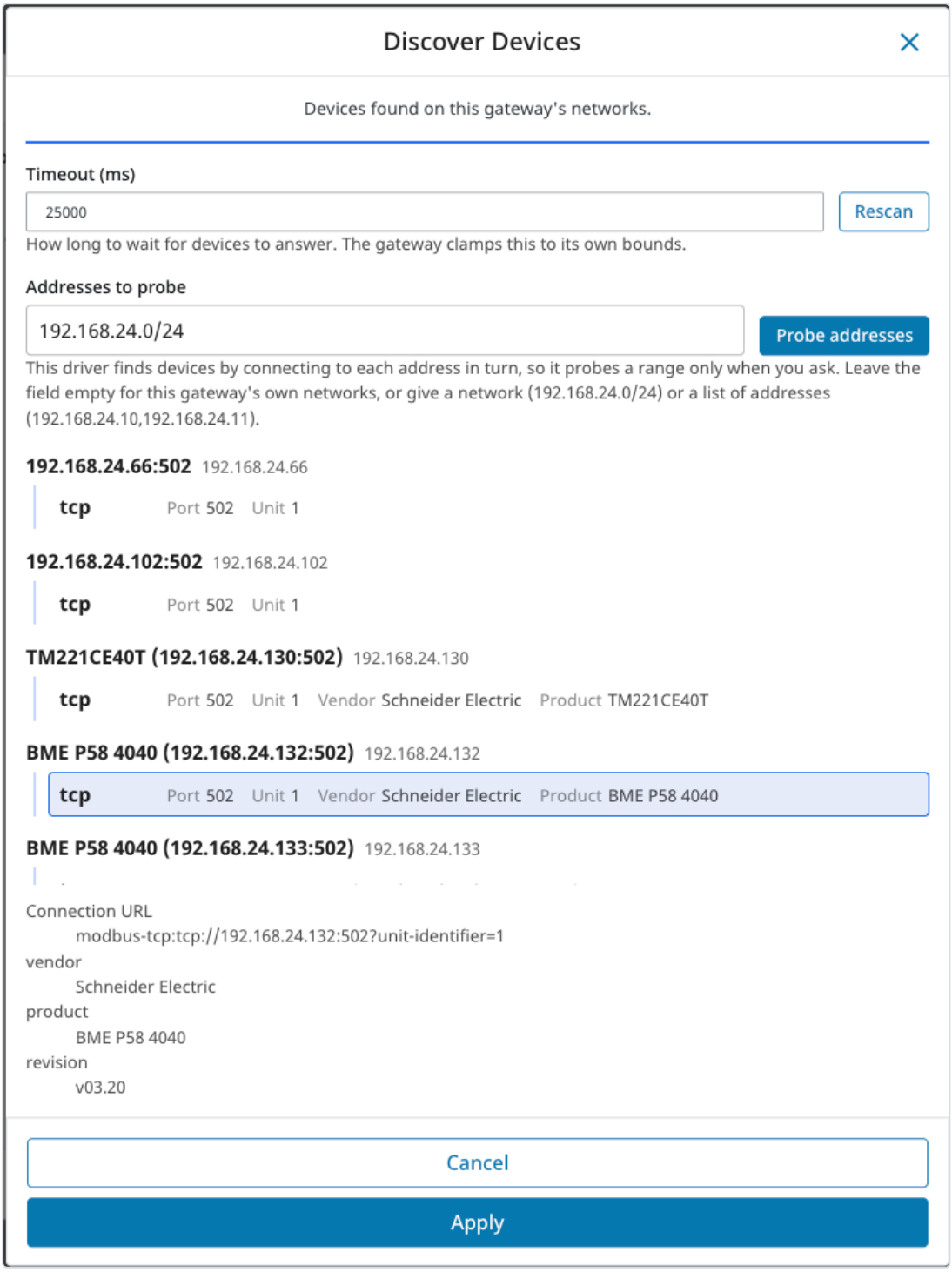


Danach:

1. Tragen Sie bei Bedarf unter **Addresses to probe** ein, was abgefragt werden soll: ein Netz (`192.168.24.0/24`) oder eine kommagetrennte Liste von Adressen

(`192.168.24.10,192.168.24.11`). Leer gelassen gelten die eigenen Netze des Gateways.
2. Klicken Sie auf **Probe addresses**. Ein späteres **Rescan** fragt dieselben Adressen erneut ab; mit dem Schließen des Dialogs verfällt die Erlaubnis.

Die Auswahl eines Ergebnisses füllt **Host**, **Port** und **Unit Identifier** — die Unit, auf der das Gerät geantwortet hat, sodass die Verbindung genau das öffnet, was abgefragt wurde. Beantwortet das Gerät die Identifikationsanfrage, zeigt das Ergebnis zusätzlich Hersteller und Produkt.



- **Ein Bereich mit mehr als 1024 Adressen wird abgelehnt**, bevor etwas gesendet wird, und unter **Range too large** genannt; geben Sie einen kleineren an.
- **Das Timeout gilt für die gesamte Abfrage.** Läuft es ab, zeigt die Liste, was rechtzeitig geantwortet hat, und sagt es unter **Ran out of time**; erhöhen Sie das Timeout oder verkleinern Sie den Bereich.
- **Die Abfrage überquert geroutete Netze**, anders als ein Broadcast: Ein Gerät in einem anderen Subnetz oder hinter einem VPN wird gefunden, solange das Gateway seinen Port 502 erreicht. Ein Gerät, das nur eine andere Unit als 1 bedient, beantwortet die Identifikationsanfrage auf Unit 1 nicht; tragen Sie es von Hand ein.
- **Die Abfrage ist im Netz sichtbar.** Ein Intrusion-Detection-System kann sie als Portscan melden; stimmen Sie sie mit dem Netzbetrieb ab, bevor Sie ein Produktionsnetz abfragen.

> Eine Aktion *Test Connection* gibt es für diesen Treiber nicht: Legen Sie das Gerät an und lesen Sie seinen Status.

Betrieb in Docker: der Discovery-Agent

Ein Gateway in einem Docker-Container sieht nur die Netze des Containers, nicht die der Anlage: ein leeres **Addresses to probe** bedeutet das eigene Netz des Containers, in dem es nichts zu finden gibt. Ein eingetragener Bereich funktioniert auch ohne den Agenten, weil das Abfragen geroutete Netze überquert; mit dem Agenten fragt ein leeres Feld stattdessen die eigenen Netze der Anlagenmaschine ab. Der **ToddySoft Discovery-Agent** schließt diese Lücke. Er ist ein kleiner, eigener Container, der auf derselben Maschine im Host-Netzwerk läuft und die Anlagennetze im Auftrag des Gateways durchsucht. Der Agent sucht nur: Verbinden, Durchsuchen und Lesen bleiben beim Gateway und laufen über das normale Routing der Maschine. Ein Gateway, das direkt auf der Anlagenmaschine installiert ist oder als Container im Host-Netzwerk läuft, braucht ihn nicht.

Installation. Der Agent wird als Image-Archiv zusammen mit diesem Modul geliefert, in derselben Freigabe, aus der Sie das Modul herunterladen: `discovery-agent-<Version>-<Architektur>.tar.gz`, für `amd64` oder `arm64`. Verwenden Sie den Agenten in der Version des Moduls. Auf der Maschine, auf der der Ignition-Container läuft (Linux):

```
docker load < discovery-agent-1.3.0-amd64.tar.gz
docker run -d --name toddysoft-discovery-agent \
  --restart unless-stopped --network host --cap-add NET_RAW \
  toddysoft/discovery-agent:1.3.0
```

`--network host` lässt ihn in den Anlagennetzen suchen, `NET_RAW` erlaubt ihm, die Suchrahmen zu senden und mitzuschneiden. Docker Desktop (macOS, Windows) betreibt Container in einer virtuellen Maschine; der Agent würde deren Netze durchsuchen statt die der Anlage.

In Ignition ist nichts einzustellen. Bei jedem **Discover Devices** sucht das Modul den Agenten am Default-Gateway seines Containers, unter Docker also auf der Maschine selbst. Ein später gestarteter Agent wird bei der nächsten Suche gefunden, ohne Neustart. Eine Zeile über den Ergebnissen sagt, was geschehen ist:

- *via discovery agent*: Der Agent hat gesucht; seine Geräte stehen zusammen mit denen, die das Gateway selbst gefunden hat, jedes einmal in der Liste.
- *no discovery agent found · container networks only*: Es läuft kein Agent; die Suche ist wie bisher die des Gateways.
- *discovery agent refused: …* oder *discovery agent not usable: …*: Der Grund folgt, etwa die Lizenz oder das gemeinsame Geheimnis (siehe unten).

Ohne Ihre Freigabe wird nichts abgefragt, auch nicht über den Agenten: Er fragt erst nach **Probe addresses** ab, und nur die dort eingetragenen Adressen oder, bei leerem Feld, die eigenen Netze seiner Maschine.

Lizenz. Der Agent sucht nur für ein Gateway, das für diesen Treiber lizenziert ist. Mit jeder Suche sendet das Modul die Lizenz, mit der es läuft, die eingebaute Testlizenz eingeschlossen, über eine verschlüsselte Verbindung; der Agent prüft sie, verwendet sie für diese eine Suche und speichert nichts.

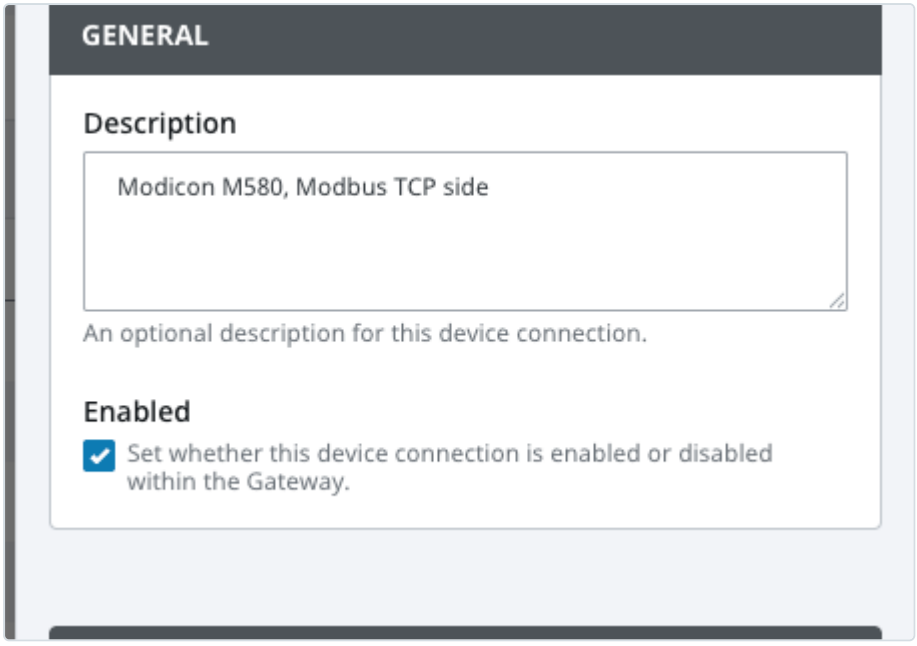
Sicherheit. Der Agent lauscht nur auf den eigenen Adressen der Docker-Netze auf dieser Maschine (Port 47900), nie in einem Anlagennetz, und nimmt Verbindungen nur aus den Container-Netzen an. Wer ihn weiter einschränken will, gibt ihm ein gemeinsames Geheimnis: `-e TODDYSOFT_AGENT_SECRET=<Geheimnis>` am Agenten und `-e TODDYSOFT_DISCOVERY_AGENT_SECRET=<Geheimnis>` am Ignition-Container. Ist ein Geheimnis gesetzt, wird ein Gateway ohne dieses oder mit einem anderen abgewiesen. Soll das Modul gar nicht nach einem Agenten suchen, setzen Sie `TODDYSOFT_DISCOVERY_AGENT=off` am Ignition-Container.

Einstellungsblöcke

Die Konfiguration ist in Kategorien gegliedert: **General**, **Transport**, **Modbus**, **TCP** und **Audit Log**.

General

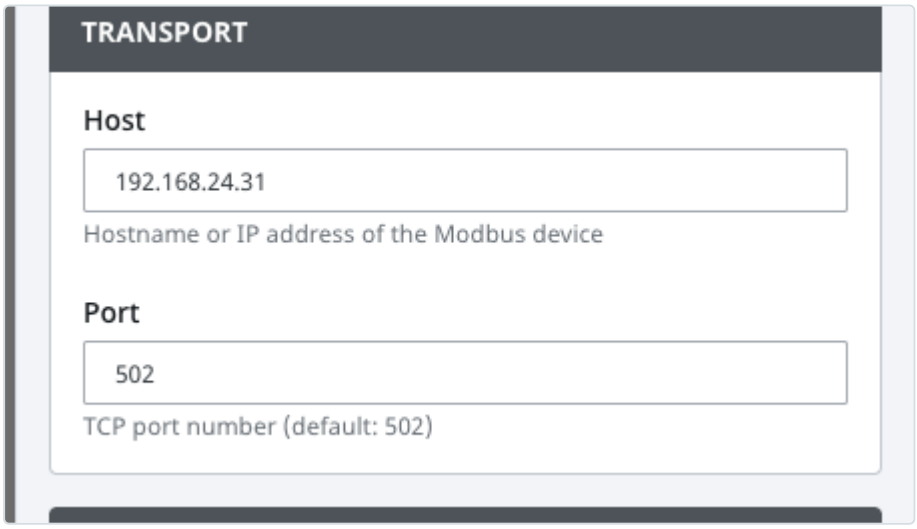
Vergibt einen Namen (und optional eine Beschreibung) für die Verbindung und steuert, ob sie aktiv ist.



Einstellung	Beschreibung	Standard
Name	Eindeutiger Name dieser Geräteverbindung	
Description	Optionale Beschreibung dieser Verbindung	
Enabled	Ob die Verbindung aktiv ist	`true`

Unter diesem Namen wird die Verbindung anschließend in anderen Teilen von Ignition referenziert.

Transport



Einstellung	Beschreibung	Standard
Host	Hostname oder IP-Adresse des Modbus-Geräts	
Port	TCP-Portnummer	`502`

Modbus

MODBUS

Unit Identifier

1

Unit-identifier or slave-id that identifies the target PLC (default: 1)

Request Timeout

5000

Default timeout for all types of requests in milliseconds

Ping Address

4x00001:BOOL

Address to check connection status (default: holding register 1)

Default Byte Order

BIG_ENDIAN

Default encoding for register values: BIG_ENDIAN, LITTLE_ENDIAN, BIG_ENDIAN_BYTE_SWAP, LITTLE_ENDIAN_BYTE_SWAP

Max Coils Per Request

2000

Maximum number of coils addressable in one request

Max Registers Per Request

125

Maximum number of registers addressable in one request

Max Address Gap To Merge

10

Maximum address gap between tags to merge into single request

Max Concurrent Requests

1

Maximum number of concurrent in-flight requests (increase with caution)

Browse Queries

Query string for symbol browsing. Syntax is protocol-specific; leave empty to disable browsing.

Einstellung	Beschreibung	Standard
Unit Identifier	Unit-Identifier bzw. Slave-ID des Zielgeräts	`1`
Request Timeout	Millisekunden, die eine einzelne Anfrage dauern darf	`5000`
Ping Address	Adresse, die zur Prüfung der Verbindung gelesen wird	`4x00001:BOOL`
Default Byte Order	Standardkodierung mehrregistriger Werte	`BIG_ENDIAN`
Max Coils Per Request	Höchstzahl der in einer Anfrage adressierbaren Coils	`2000`
Max Registers Per Request	Höchstzahl der in einer Anfrage adressierbaren Register	`125`
Max Address Gap To Merge	Größter Adressabstand, über den Tags noch zu einer Anfrage zusammengefasst werden	`10`
Max Concurrent Requests	Gleichzeitig offene Anfragen	`1`
Browse Queries	Bei diesem Treiber wirkungslos — Modbus kann keine Tags aufzählen	

Default Byte Order

Modbus legt die Reihenfolge der Bytes innerhalb eines Registers fest (höherwertiges Byte zuerst), nicht aber die der *Register* eines mehrregistrigen Werts, und manche Geräte vertauschen zusätzlich die Bytes. Hersteller unterscheiden sich daher tatsächlich, und es gibt keinen richtigen Standardwert. Eine falsche Einstellung schlägt nicht fehl — sie liefert plausiblen Unsinn, und genau darauf ist zu achten: ein `REAL`, das als absurd große oder winzige Zahl erscheint, oder eine Ganzzahl mit vertauschten Bytes.

Zulässig sind vier Werte: `BIG_ENDIAN`, `LITTLE_ENDIAN`, `BIG_ENDIAN_BYTE_SWAP` und `LITTLE_ENDIAN_BYTE_SWAP`. In den Begriffen der Gerätehandbücher: ABCD ist `BIG_ENDIAN`, BADC (Byte-Tausch) `BIG_ENDIAN_BYTE_SWAP`, CDAB („Word Swap", „Low Word First") `LITTLE_ENDIAN_BYTE_SWAP` und DCBA `LITTLE_ENDIAN`. Dies ist der verbindungsweite Standard; einzelne Tags können ihn übersteuern (siehe *Optionen je Tag*) — genau das brauchen Sie für ein Gerät, das in sich selbst uneinheitlich ist.

Anfragegröße und Zusammenfassung

Max Coils Per Request (`2000`) und **Max Registers Per Request** (`125`) sind die Obergrenzen des Protokolls. Viele Geräte akzeptieren weniger; antwortet ein Gerät auf kleine Lesezugriffe und liefert bei großen Fehler, senken Sie diese Werte, bevor Sie etwas anderes vermuten.

Max Address Gap To Merge erlaubt dem Treiber, nahe beieinanderliegende Tags zu einer Anfrage zusammenzufassen und dabei einige nicht benötigte Register mitzulesen, um einen Roundtrip zu sparen. Ein höherer Wert tauscht Bytes auf der Leitung gegen weniger Anfragen. Vorsicht beim Zusammenfassen *über* eine Lücke hinweg, die das Gerät als unzulässig betrachtet: Manche Geräte quittieren einen Lesezugriff über eine nicht gemappte Adresse mit einem Fehler — dann ist dieser Wert zu senken.

Max Concurrent Requests steht standardmäßig auf `1` , und das „increase with caution“ der Beschreibung ist verdient: Viele Modbus-Geräte sind Single-Threaded und beantworten gleichzeitige Anfragen mit Fehlern, verworfenen Frames oder vertauschten Antworten. Erhöhen Sie den Wert nur für ein Gerät, für das dies dokumentiert ist.

Ping Address

Die Adresse, die regelmäßig gelesen wird, um die Verbindung zu prüfen. Der Standardwert `4x00001:BOOL` ist Holding Register 1 — das es auf den meisten, aber nicht auf allen Geräten gibt. Läuft ein Gerät beim Ping auf Fehler, während gewöhnliche Tags korrekt lesen, richten Sie diesen Wert auf ein Register, von dem Sie wissen, dass es gemappt ist.

TCP

Feineinstellungen auf Socket-Ebene. Die Standardwerte passen für nahezu jedes Gerät; ändern Sie sie für ein konkret diagnostiziertes Problem und nicht als routinemäßige Optimierung.

TCP

Connect Timeout

5000

Connection timeout in milliseconds (0 = default)

Read Timeout

0

Socket read timeout in milliseconds (0 = no timeout)

Write Timeout

0

Socket write timeout in milliseconds (0 = no timeout)

TCP No Delay

☑

Enable TCP_NODELAY (disable Nagle's algorithm)

TCP Keep Alive

☐

Enable TCP_KEEPALIVE

Send Buffer Size

81920

Send buffer size in bytes

Receive Buffer Size

81920

Receive buffer size in bytes

Local Address

Local address to bind to (leave empty to bind to all interfaces)

Local Port

0

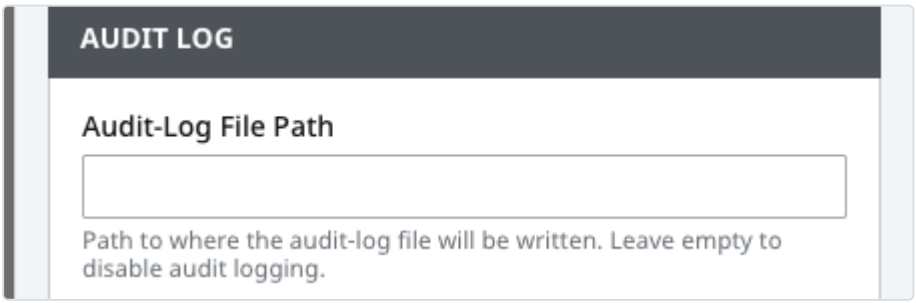
Local port to bind to (0 = use ephemeral port)

Einstellung	Beschreibung	Standard
Connect Timeout	Verbindungs-Timeout in Millisekunden (`0` = Standard)	`5000`
Read Timeout	Socket-Lese-Timeout in Millisekunden (`0` = kein Timeout)	`0`
Write Timeout	Millisekunden, die ein Schreibvorgang darauf warten darf, dass das Gerät Daten annimmt; danach wird die Verbindung geschlossen und neu aufgebaut (`0` = keine Grenze)	`30000`
TCP No Delay	`TCP_NODELAY` aktivieren (Nagle-Algorithmus abschalten)	`true`
TCP Keep Alive	`TCP_KEEPALIVE` aktivieren	`false`
Send Buffer Size	Größe des Sendepuffers in Bytes	`81920`
Receive Buffer Size	Größe des Empfangspuffers in Bytes	`81920`
Local Address	Lokale Bind-Adresse; leer bindet an alle Schnittstellen	
Local Port	Lokaler Bind-Port (`0` = ephemer)	`0`

TCP No Delay ist standardmäßig aktiv, weil Modbus-Austausche kleine Request/Response-Paare sind — genau das Muster, das der Nagle-Algorithmus verzögert.

Local Address wird gelegentlich auf einem mehrfach angebundenen Gateway benötigt, damit die Verbindung über eine bestimmte Schnittstelle hinausgeht.

Audit Log



Einstellung	Beschreibung	Standard
Audit-Log File Path	Pfad zu einer Datei, in die Protokoll-Audit-Ereignisse geschrieben werden; leer deaktiviert	

Das Audit-Log zeichnet Ereignisse auf Protokollebene auf und wächst schnell. Aktivieren Sie es daher nur zur Untersuchung eines Problems.

Tag-Adressierung

```
<Bereichspräfix>:<Startregister>[:<Datentyp>][[<Auswahl>]]
[{<Option>: <Wert>[, ...]}]
```

Bereichspräfixe sind `holding-register`, `input-register`, `coil` und `discrete-input`; auch Schneiders Form `%MW0` / `%M0` wird akzeptiert.

Adresse	Bedeutung
`holding-register:1:INT`	Eine vorzeichenbehaftete 16-Bit-Ganzzahl, das erste Holding-Register
`holding-register:1:DINT`	32-Bit-Ganzzahl über die Register 1–2
`holding-register:1:REAL`	IEEE-754-Gleitkommazahl über die Register 1–2
`holding-register:1:STRING(20)`	Eine 20-Zeichen-Zeichenkette ab Register 1
`holding-register:1[0..9]:DINT`	Zehn 32-Bit-Ganzzahlen, Register 1–20
`holding-register:1[9]:DINT`	**Ein** DINT, nämlich das zehnte
`holding-register:1[0..19]:CHAR`	Zwanzig einzelne Zeichen als Liste
`coil:1`	Ein einzelnes Coil
`coil:1[0..7]:BOOL`	Acht Coils als BOOL-Array

> **Registernummern zählen ab 1 innerhalb ihres Bereichs.** `holding-register:1` ist das erste Holding-Register (Offset 0 auf der Leitung). Die `40001` eines Gerätehandbuchs ist dasselbe Register in der klassischen Schreibweise, die der Treiber auch als `4x00001` annimmt; als `holding-register:40001` geschrieben ist es dagegen Register 40001, meist jenseits der Belegung des Geräts.

> **Ein bloßes `[N]` ist ein Element und keine Anzahl von N**, und die Auswahl steht **vor** dem `:TYP`-Suffix. Beides wurde in ToddySoft Connect 1.2 gemeinsam geändert, damit eine für eine frühere Version geschriebene Adresse **nicht mehr geparst wird**, statt stillschweigend ein Register zu liefern, wo zuvor zehn geliefert wurden. Jede Parse-Fehlermeldung nennt die Ersetzung.

`STRING` und `WSTRING` müssen eine Länge tragen, da auf der Leitung nichts eine solche ankündigt: `STRING(20)[3]` sind drei 20-Zeichen-Zeichenketten, `CHAR[20]` dagegen zwanzig einzelne Zeichen. Ein `STRING` ohne Länge wird beim Parsen der Adresse abgelehnt.

Optionen je Tag

Ein angehängtes `{…}` übersteuert Verbindungseinstellungen für ein einzelnes Tag.

Option	Werte	Bedeutung
`unit-id`	0–255	Adressiert dieses Tag an eine bestimmte Unit bzw. Slave
`byte-order`	`BIG_ENDIAN`, `LITTLE_ENDIAN`, `BIG_ENDIAN_BYTE_SWAP`, `LITTLE_ENDIAN_BYTE_SWAP`	Byte-Reihenfolge für dieses Tag

```
holding-register:100:INT{unit-id: 5}
holding-register:100[0..9]:DINT{unit-id: 5, byte-order:
LITTLE_ENDIAN}
```

Über `unit-id` erreichen Sie mehrere Geräte hinter einer Verbindung — typischerweise eine serielle Multidrop-Leitung hinter einem Modbus-Gateway. Tags, die an unterschiedliche Units adressiert sind, werden nie zu einer Anfrage zusammengefasst; ein Gerät mit vielen Units kostet also mindestens eine Anfrage je Unit.

Tag-Gruppen (Tag Groups)

Standardmäßig werden alle Tags in der Standard-Tag-Gruppe erfasst, das heißt, alle ausgewählten Tags werden im Standardintervall von einmal pro Sekunde abgefragt.

Werden andere Intervalle gewünscht, legen Sie zusätzliche Tag-Gruppen an, konfigurieren dort die gewünschten Intervalle und weisen die Tags den entsprechenden Gruppen zu.

Modbus kennt keine Änderungsbenachrichtigung; Subscriptions werden daher gepollt, und das Intervall der Tag-Gruppe *ist* die Abfragerate. Tags, die im Adressraum nahe beieinanderliegen und im selben Intervall gelesen werden, werden zu einer Anfrage zusammengefasst — nach *beidem* zu gruppieren, nach Adressnähe und nach Rate, hält also die Anzahl der Anfragen niedrig.

Ausgelöste Gruppen (Triggered Groups)

Eine ausgelöste Gruppe ist eine Menge von Tags, die gemeinsam in einer Anfrage gelesen werden, wenn sich ein Trigger-Tag so ändert, wie eine Bedingung es beschreibt, statt in einem festen Abfrageintervall: zum Beispiel die Qualitätsdaten eines fertigen Teils. Ohne Gruppen wird jeder Tag im Intervall seiner Tag-Gruppe gelesen.

Die Karte **Triggered Groups** listet die Gruppen des Geräts. **Add group** öffnet einen Dialog für eine Gruppe (Name, Trigger-Tag, Bedingung, *Fire on the first value after start* und die Mitglieder, eine Adresse je Zeile); gespeichert werden sie als die unten gezeigte JSON-Liste, die sich auch einfügen lässt.

```
[
  {
    "name": "part-done",
    "trigger": "<Adresse des Trigger-Tags>",
    "condition": "oldValue == false && value == true",
    "members": ["<Adresse>", "<Adresse>"]
  }
]
```

Eigenschaft	Bedeutung	Standard
`name`	1-64 Buchstaben, Ziffern, `_` oder `-`; eindeutig im Gerät (ohne Unterschied der Schreibweise)	erforderlich
`trigger`	SPS-Adresse des Tags, dessen Änderung die Gruppe auslöst, geschrieben wie die Adresse eines Tags	erforderlich
`condition`	Wann eine Änderung auslöst, über `oldValue` und `value`	`value != oldValue`
`fireOnFirstValue`	Auch einmal beim ersten Wert nach dem Start des Geräts auslösen	`false`
`members`	SPS-Adressen, die beim Auslösen gelesen werden, höchstens 1.000	keine

Die Steuerung kann Änderungen nicht selbst melden; der Treiber liest den Trigger-Tag deshalb im Abstand von **Subscription Poll Interval (ms)** (Kategorie *Timing*, Standard 1000 ms). Eine Änderung, die schneller wieder zurückgenommen wird, bleibt unbemerkt; für kurze Impulse das Intervall verkleinern.

Jede Gruppe erscheint im Tag-Baum des Geräts als Ordner `$triggered/<Name>` mit ihren Mitgliedern, `$trigger` (dem letzten Wert des Triggers), `$fireCount`, `$lastFire` (Zeitpunkt des letzten Auslösens), `$coalesced` (Auslösungen, die kamen, während noch gelesen wurde, und in den nächsten Lesezugriff übernommen wurden) und `$status` (`ESTABLISHED`, solange sie läuft, sonst der Zustand mit seinem Grund). Eine Gruppe mit ungültiger Definition lässt sich nicht speichern: Das Geräteformular nennt jedes Problem. Eine Gruppe behält den letzten Wert ihres Triggers über einen Verbindungsabbruch hinweg; eine Änderung während des Ausfalls löst deshalb trotzdem einmal aus.

Lizenz aktivieren

Ohne Lizenz läuft das Modul in Ignitions Testphase: jeweils zwei Stunden, danach stehen seine Geräte, bis jemand im Banner des Gateways **Reset Trial** klickt – beliebig oft. Eine Lizenz für dieses Modul kommt von ToddySoft per E-Mail als **Ignition-Lizenzschlüssel** mit einem **Aktivierungs-Token**. Die ToddySoft-Lizenz für die Treiber steckt in diesem Schlüssel: Ihn auf dem Gateway zu aktivieren ist alles, was zu tun ist. Es gibt keine Lizenzdatei zu kopieren und keine Lizenzeinstellung im Geräteformular. Halten Sie Schlüssel und Token vertraulich: Zusammen aktivieren sie die Lizenz.

Online (das Gateway erreicht das Internet)

1. Zuerst das Modul installieren (siehe Installation).
2. In der Weboberfläche des Gateways **Platform > System > Licensing** öffnen, **Activate a License** klicken, den Schlüssel unter **License Key** eingeben und **Next** klicken.
3. **Online Activation** wählen und **Activate License** klicken. Fragt das Gateway nach einem Aktivierungs-Token, das Token aus der E-Mail einfügen: eine Zeile, ohne die Zeilen `BEGIN` und `END`.
4. Die Lizenzseite zeigt den Schlüssel als aktiviert, und unter **Platform > System > Modules** steht dieses Modul als **Activated**.

Das Gateway fragt dafür den Aktivierungsdienst von Inductive Automation (`api.inductiveautomation.com`, HTTPS); eine Firewall muss diese eine ausgehende Verbindung zulassen.

Offline (das Gateway erreicht das Internet nicht)

Die Offline-Aktivierung ist ein Austausch zweier Dateien zwischen dem Gateway und einem beliebigen Rechner mit Internetzugang. Sie funktioniert mit sechsstelligen Lizenzschlüsseln.

1. Auf dem Gateway **Platform > System > Licensing** öffnen, **Activate a License** klicken, den Schlüssel unter **License Key** eingeben und **Next** klicken.
2. **Offline Activation** wählen, dann **Download Activation Request** klicken. Fragt das Gateway nach einem Aktivierungs-Token, das Token aus der E-Mail einfügen. Das Gateway speichert die Datei `activation_message.txt`.
3. Auf einem Rechner mit Internetzugang https://links.inductiveautomation.com/activation öffnen, `activation_message.txt` auswählen und die zurückgegebene Datei `license.ipl` speichern.
4. Zurück auf der Seite **Offline Activation** des Gateways `license.ipl` hochladen und **Activate** klicken.

Nach der Aktivierung

Ein Neustart ist nicht nötig. Das Modul übernimmt die Lizenz, sobald Ignition die Aktivierung meldet, und Geräte, die mangels Lizenz abgewiesen wurden, verbinden sich beim nächsten Versuch wieder. Der Lizenzstatus zeigt **VALID** mit Stufe und Wartungsende (siehe Lizenzdiagnose); alles andere wird dort erklärt.

Lizenz verlängern oder erweitern

- **Hat ToddySoft die Lizenz auf Ihrem bestehenden Schlüssel aktualisiert** (weitere Treiber, längere Wartung), muss das Gateway sie neu abholen. Online: in **Platform > System > Licensing** das Drei-Punkte-Menü des Schlüssels öffnen und **Reactivate** wählen. Offline: den Schlüssel zuerst deaktivieren (unten), dann erneut offline aktivieren. Inductive Automation warnt, dass eine Offline-Aktivierung ohne vorheriges Deaktivieren das Gateway in den Notfall-Aktivierungsmodus (Emergency Activation Mode) versetzen kann.
- **Haben Sie einen neuen Schlüssel erhalten**, aktivieren Sie ihn und deaktivieren den alten. Halten Sie je Gateway genau **einen** aktivierten Schlüssel, der dieses Modul abdeckt: Bei zweien gibt Ignition dem Modul nur einen davon (siehe Fehlerbehebung).

Lizenz auf ein anderes Gateway übertragen

Den Schlüssel auf dem alten Gateway deaktivieren, dann auf dem neuen aktivieren. Die E-Mail nennt, auf wie vielen Gateways ein Schlüssel gleichzeitig aktiviert sein darf.

- Online: in **Platform > System > Licensing** das Drei-Punkte-Menü des Schlüssels öffnen, **Unactivate** wählen und mit **Unactivate** bestätigen.

- Offline: in **Platform > System > Licensing** das Drei-Punkte-Menü des Schlüssels öffnen, **Unactivate** wählen und mit **Unactivate** bestätigen. Das Gateway speichert `unactivation_message.txt`; diese Datei von einem Rechner mit Internetzugang auf https://links.inductiveautomation.com/activation hochladen.

Ignition Edge: Eine Lizenz für ein Drittanbieter-Modul unter Ignition Edge wird über Inductive Automation eingerichtet. Wenden Sie sich vor der Bestellung an ToddySoft.

Lizenzdiagnose

Meldet ein Gerät ein Lizenzproblem, erklärt das Modul die Ursache direkt am Gateway. Öffnen Sie die Einstellungen des Geräts (**Connections > Devices > Connections**, Gerät bearbeiten oder neu anlegen): Die Karte **Licence** oben im Formular zeigt die Diagnose. Sie erscheint auch beim Anlegen eines neuen Geräts, sodass sich die Lizenz prüfen lässt, bevor das erste Gerät existiert.

Die Diagnose zeigt:

- **welches Gateway** es ist, und seine Rolle, wenn es Teil eines redundanten Paares ist;
- **die Ignition-Lizenz** dieses Moduls auf diesem Gateway: aktiviert, Testphase läuft (mit ihrem Ende) oder Testphase abgelaufen;
- **welche ToddySoft-Lizenzdaten angekommen sind**: wie viele Teile, ob sie vollständig sind, und eine Prüfsumme, die die Lizenz kennzeichnet, ohne sie preiszugeben;
- das **Urteil des Treibers** und die **Ursache** in Worten.

Ursache	Bedeutung	Was tun
`IGNITION_UNLICENSED`	Ignition wendet auf diesem Gateway keine aktivierte Lizenz auf dieses Modul an (Testphase läuft oder ist abgelaufen). ToddySoft-Lizenzdaten kommen nur mit einer aktivierten Lizenz an.	Einen Lizenzschlüssel aktivieren, der dieses Modul auf diesem Gateway abdeckt. In einem redundanten Paar braucht jeder Knoten seine eigene Aktivierung. Ist ein Schlüssel aktiviert und diese Ursache bleibt, siehe Fehlerbehebung: Ignition beurteilt das Modul unter Umständen nach einem älteren, abgelaufenen Schlüssel, der es ebenfalls abdeckt.

Ursache	Bedeutung	Was tun
`NO_DATA_FOR_DRIVER`	Der aktivierte Schlüssel enthält keine ToddySoft-Lizenz für diesen Treiber.	Den Bericht an ToddySoft senden.
`DATA_INCOMPLETE`	Die Lizenzdaten sind unvollständig oder beschädigt angekommen.	Den Bericht an ToddySoft senden; die Lizenz muss neu geliefert werden.
`DATA_REJECTED`	Der Treiber hat die gelieferte Lizenz abgelehnt; der Grund wird angezeigt.	Den Bericht an ToddySoft senden.
`RUNNING_ON_BUNDLED_TRIAL`	Eine Lizenz wurde geliefert, aber nicht angenommen; der Treiber läuft mit seiner eingebauten Testlizenz, bis diese endet.	Den Bericht an ToddySoft senden.
`NOT_CHECKED`	Lizenzdaten sind angekommen, aber der Treiber hat kein Urteil abgegeben – etwa weil er nicht geladen werden konnte.	Den Bericht an ToddySoft senden.
`VALID`	Lizenziert; Stufe und Ablaufdatum werden angezeigt.	Nichts.
`NOT_YET_KNOWN`	Die ersten Sekunden nach dem Start, bevor Ignition einen Lizenzstatus meldet.	Kurz warten.

Einen Bericht an den Support senden. Die Diagnose gibt es als Klartext-Bericht – das, was bei jeder Lizenzfrage mitzuschicken ist. **Copy report** legt ihn in die Zwischenablage; **Show report** zeigt ihn an. Derselbe Text steht angemeldeten Benutzern unter `http://<gateway>:`

`<port>/data/com.toddysoft.connect.modbus/licence-report` bereit. Er enthält keine Lizenzdaten, keine Schlüssel, keine Zugangsdaten und keine Kundendaten. In einem redundanten Paar schicken Sie den Bericht beider Knoten.

Das Gateway-Log enthält denselben Bericht einmal bei jeder Änderung der Lizenzsituation.

Fehlerbehebung

Lizenzprobleme nach dem Aktivieren eines neuen oder verlängerten Lizenzschlüssels — `DATA_INCOMPLETE` mit „duplicate fragment index" oder `IGNITION_UNLICENSED` mit „trial expired" in der Lizenzdiagnose:

- **Am Gateway sind gleichzeitig zwei Lizenzschlüssel aktiviert, die dieses Modul abdecken**, meist ein verlängerter oder erweiterter Schlüssel neben dem alten. Ignition übergibt dem Modul die Lizenzdaten beider Schlüssel in nummerierten Teilen mit denselben Nummern; das Modul kann nicht erkennen, welche Teile zusammengehören, behält von jedem Paar einen, und die zusammengesetzte Lizenz ist unlesbar. Je nachdem, welche Teile es behält, werden Geräte abgewiesen oder laufen mit der eingebauten Testlizenz des Moduls, bis diese abläuft.
- Ist einer der beiden Schlüssel **abgelaufen**, beurteilt Ignition das Modul unter Umständen allein nach diesem: Die Diagnose zeigt `IGNITION_UNLICENSED` („trial expired - no activated licence applies to this module") und „Data: none received", obwohl der neue Schlüssel erfolgreich aktiviert wurde.
- Pro Gateway genau **einen** aktivierten Lizenzschlüssel behalten, der dieses Modul abdeckt: den alten unter **Platform > System > Licensing** deaktivieren und anschließend das Modul (oder das Gateway) neu starten. In einem redundanten Paar jeden Knoten prüfen.

Das Modul lädt nicht, oder im Log erscheinen `ClassNotFoundException` bzw. `NoClassDefFoundError`:

- Das **OPC-UA**-Modul von Ignition fehlt oder ist deaktiviert. Dieser Treiber ist ein OPC-UA-Gerätetreiber und kann ohne dieses Modul nicht geladen werden. Installieren bzw. aktivieren Sie es unter **Platform > System > Modules** und starten Sie das Gateway neu — siehe Installation

Das Gerät zeigt „Faulted“:

- Prüfen Sie, ob das Gerät erreichbar und TCP-Port 502 (bzw. Ihr abweichender Port) offen ist
- Prüfen Sie den **Unit Identifier** — ein Gerät hinter einem Gateway antwortet selten auf der `1`
- Prüfen Sie, ob die **Ping Address** auf diesem Gerät existiert; Standard ist Holding Register 1

- Prüfen Sie die Gateway-Logs: **Diagnostics > Logs**

Werte wirken plausibel, sind aber falsch — absurde Gleitkommazahlen, vertauschte Ganzzahlen:

- Fast immer **Default Byte Order**. Probieren Sie die anderen drei Werte; für ein in sich uneinheitliches Gerät übersteuern Sie je Tag mit `{byte-order: …}`

Kleine Lesezugriffe funktionieren, größere schlagen fehl:

- Senken Sie **Max Registers Per Request** bzw. **Max Coils Per Request**. Die Obergrenzen des Protokolls sind nicht das, was jedes Gerät akzeptiert
- Oder senken Sie **Max Address Gap To Merge**: Ein zusammengefasster Lesezugriff kann über eine nicht gemappte Adresse laufen, die das Gerät ablehnt

Sporadische Fehler, verworfene oder vertauschte Antworten:

- Setzen Sie **Max Concurrent Requests** zurück auf `1`. Viele Geräte sind Single-Threaded

Im Designer lässt sich nichts browsen:

- Erwartet. Modbus kennt keine Tag-Aufzählung; schreiben Sie Adressen von Hand aus der Modbus-Tabelle des Geräts

Eine bisher funktionierende Adresse wird nicht mehr geparst:

- Die Array-Grammatik hat sich geändert: Die Auswahl steht vor dem Typ, und `[N]` ist ein Element statt einer Anzahl. Die Fehlermeldung nennt die Ersetzung

Docker-bezogene Probleme

Das Modbus-Protokoll überträgt keine Identität dieser Verbindungsseite. Die interne Adresse eines Containers wird dem Gerät daher nie mitgeteilt, und es muss nichts abgeglichen werden. Ein Gateway in einem Bridged-Container verbindet sich normal, sofern der Port des Geräts routbar ist.

Die Gerätesuche braucht den Discovery-Agenten. *Discover Devices* sucht in den Netzen, an denen das Gateway selbst hängt — bei einem Container also in denen von Docker und nicht in denen der Anlage. Betreiben Sie den ToddySoft Discovery-Agenten neben dem Container — siehe Betrieb in Docker: der Discovery-Agent — oder tragen Sie den Host von Hand ein.

VPN-bezogene Probleme

Ein über VPN erreichtes entferntes Gerät benötigt von diesem Treiber nichts Besonderes: Die Verbindung ist eine gewöhnliche ausgehende TCP-Sitzung. Beachten Sie jedoch, dass Latenz hier stärker ins Gewicht fällt als bei den meisten Treibern, weil

Modbus streng nach Request/Response arbeitet — bei **Max Concurrent Requests** auf `1` kostet jeder Tag-Lesezugriff einen vollen Roundtrip.

Versionshistorie

1.3.0 — 2026-11-13

- **Neu:** Ausgelöste Gruppen - Tags, die gemeinsam gelesen werden, wenn sich ein Trigger-Tag ändert, statt in einem festen Intervall. Siehe Ausgelöste Gruppen.
- **Behoben:** Schreibzugriffe auf erweiterte Register (`6x`) funktionieren; bisher scheiterte jeder, bevor er gesendet wurde. Große zusammengefasste Schreibzugriffe halten die Grenze von Modbus pro Anfrage ein.
- **Behoben: Write Timeout** ist zurück und wirkt, mit 30 s als Vorgabe. Nimmt das Gerät keine Daten mehr an, schlägt ein Schreibvorgang nach dieser Zeit fehl und die Verbindung wird neu aufgebaut, statt für immer zu hängen. Frühere Versionen boten die Einstellung an, wendeten sie aber nie an; ein an einem bestehenden Gerät gespeicherter Wert wird deshalb nicht übernommen. Setzen Sie ihn neu, wenn Sie einen anderen Wert als 30 s brauchen.
- **Korrigiert:** die Installationsschritte für Ignition 8.3: Nach dem Installieren oder Aktualisieren des Moduls muss das Gateway neu gestartet werden, das Menü ist **Platform > System > Modules** und der Status lautet **ACTIVE**. Alle Menüpfade folgen jetzt Ignition 8.3 (Geräte unter **Connections > Devices > Connections**, Logs unter **Diagnostics > Logs**). Siehe Installation.
- **Neu:** wie der Lizenzschlüssel online und offline aktiviert, erneuert und auf ein anderes Gateway übertragen wird. Siehe Lizenz aktivieren.
- Erste Ausgabe dieses Handbuchs.
- **Neu: Discover Devices** findet Modbus/TCP-Geräte durch Abfrage von Adressen, mit Ihrer Erlaubnis. Siehe Geräte im Netzwerk finden.
- **Neu:** Lizenzdiagnose – warum ein Gerät lizenziert ist oder nicht, direkt am Gateway, mit einem Bericht für den Support. Siehe Lizenzdiagnose.
- **Neu:** Hinweis zur Fehlerbehebung, wenn an einem Gateway zwei Lizenzschlüssel gleichzeitig aktiviert sind, die dieses Modul abdecken. Siehe Fehlerbehebung.
- **Neu:** Gerätesuche, wenn Ignition in Docker läuft, über den ToddySoft Discovery-Agenten. Siehe Betrieb in Docker: der Discovery-Agent.
- **Behoben:** Ein Schreibzugriff, den die SPS ablehnt, schlägt jetzt auch in Ignition fehl, mit dem Grund als Status (zum Beispiel `Bad_TypeMismatch` für einen Wert des falschen Typs). Bisher wurde jeder Schreibzugriff als erfolgreich gemeldet, und die Ablehnung stand nur im Gateway-Log.
- **Behoben:** Ein Tag, der noch nicht gelesen wurde oder dessen letzter Lesezugriff fehlschlug, liest nicht mehr 0 mit guter Qualität (Good). Vor dem ersten Lesen hat er keinen Wert und die Qualität Uncertain; nach einem fehlgeschlagenen Lesen behält er seinen letzten Wert mit schlechter Qualität (Bad) bis zum nächsten erfolgreichen Lesen.

- **Behoben:** Wird der letzte Tag einer Tag-Gruppe (oder Abfragerate) entfernt, zeigt das Gerät nicht mehr *Connection Lost*. Ein Lesezugriff, der in diesem Moment noch lief, ließ das Gerät mit `Batch '...' not found` im Gateway-Log fehlschlagen, und es blieb bis zu einem Neustart in diesem Zustand.