

# Inhaltsverzeichnis

- Unterstützte Geräte** ..... 1
- Konfiguration** ..... 1
  - Channel ..... 1
  - Settings ..... 2
  - Allgemein ..... 2
  - Serial ..... 3
  - TCP ..... 4
  - Hinzufügen eines Channels ..... 4
  - Variablen ..... 5
  - Path ..... 5
- Fehlerdiagnose** ..... 6
  - Kanal ..... 6
  - Variablen ..... 7
  - Logdatei ..... 7
- Entities** ..... 8
  - Device ..... 8
  - Channel ..... 8
  - Variable ..... 9
- Ordner & Dateien** ..... 9
  - Ordner ..... 9
  - Dateien ..... 9
- Versionsinformation** ..... 9
  - Dieses Dokument ..... 9
  - Plugin ..... 9
  - Assembly ..... 9



# Omron Device Plugin

Das Omron Device Plugin ermöglicht das Lesen und Schreiben von Daten von Omron Geräten.

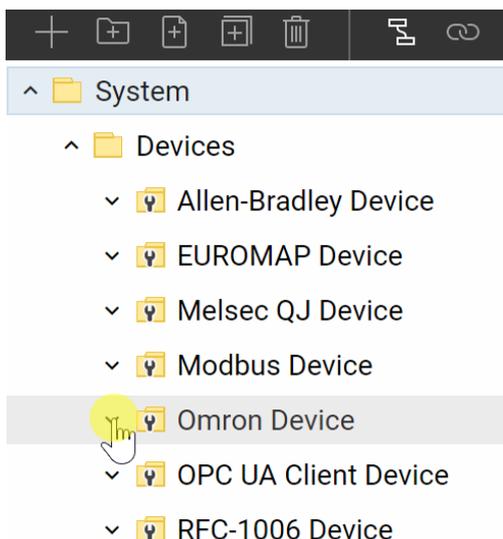
## Unterstützte Geräte

Es werden alle Omron Geräte unterstützt, die mit dem FINS-Protokoll (seriell/TCP) oder mittels Host-Link/C-Mode Commands angesprochen werden können:

- CS
- CJ
- CP
- NSJ
- CVM1
- CV

## Konfiguration

Die gesamte Omron Device Plugin-Konfiguration befindet sich unter dem Nodepfad /System/Devices/Omron Device.



## Channel

Ein Omron Device Channel repräsentiert eine Verbindung zu einem Omron Gerät.

## Settings

### Allgemein

#### Transport type

Die für diese Verbindung verwendete Transportart.

- Serial (Host-Link): Serielle Kommunikation. Unterstützt sowohl das FINS-Protokoll mit dem Host-Link-Protokoll als Transportlayer als auch Host-Link/C-Mode Commands.
- TCP: Kommunikation über TCP/IP. Unterstützt nur das FINS-Protokoll.

#### Destination Network Address

Die Destination Network Address des Ziel-Geräts.

#### Destination Node Address

Die Destination Node Address des Ziel-Geräts.

#### Destination Unit Address

Die Destination Unit Address des Ziel-Geräts.

#### Source Network Address

Die vom Treiber verwendete Source Network Address.

#### Source Node Address

Die vom Treiber verwendete Source Node Address.

#### Source Unit Address

Die vom Treiber verwendete Source Unit Address.

#### Connect Timeout

Der Timeout, der für den Verbindungsaufbau verwendet wird.

**Read Timeout**

Der Timeout, der für einen Lesevorgang verwendet wird.

**Write Timeout**

Der Timeout, der für einen Schreibvorgang verwendet wird.

**Serial**

Einstellungen, die für die Transportart Serial (Host-Link) verwendet werden.

**Protocol type**

Das Protocol, das vom Treiber verwendet wird.

- Host-Link: Host-Link/C-Mode Commands-Protokoll
- FINS: FINS-Protokoll

**Port**

Der serielle Port, der für die Kommunikation mit dem Gerät verwendet wird.

**Baudrate**

Die Baudrate, die vom Treiber verwendet wird.

**Data bits**

Die Anzahl der Datenbits, die vom Treiber verwendet wird.

**Parity bit setting**

Die Parity Bit Einstellung, die vom Treiber verwendet wird.

**Stop bits**

Die Anzahl der Stopbits, die vom Treiber verwendet wird.

### Handshake

Die Art des Handshakes, die vom Treiber verwendet wird.

### Host link unit number

Die Host-Link Unit Number des Gerätes.

## TCP

Einstellungen, die für die Transportart TCP verwendet werden.

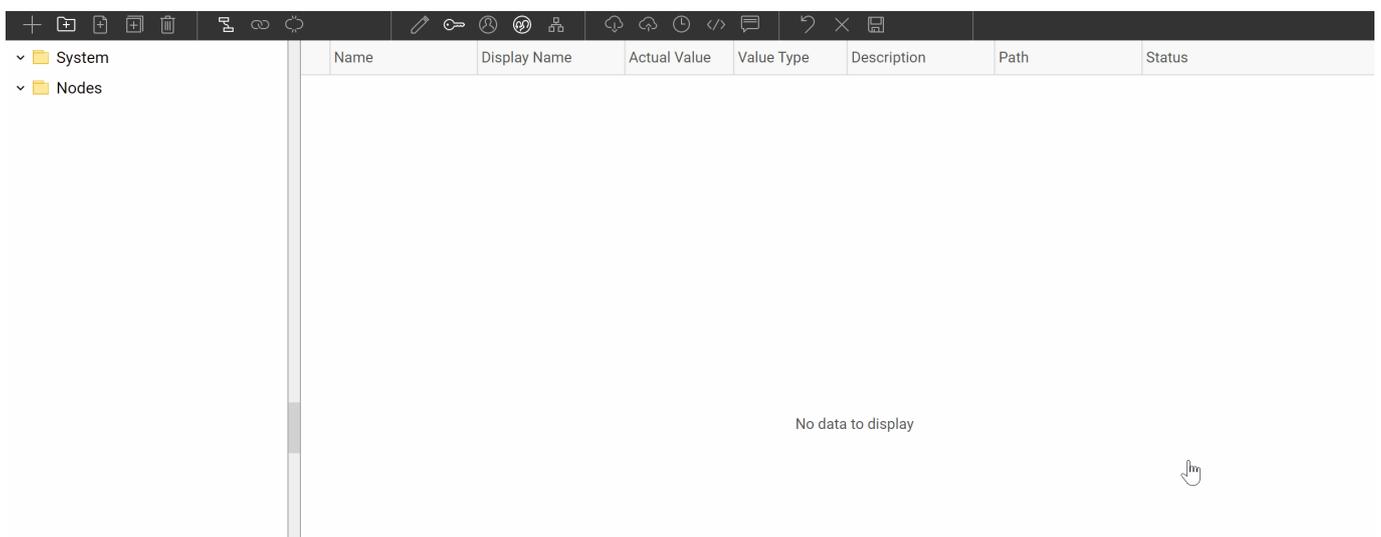
### IP Address

Die IP-Adresse des Geräts.

### Port

Der Port des Geräts. Der Standardport für das FINS-Protokoll ist 9600.

## Hinzufügen eines Channels



Um einen neuen Omron-Channel zu erstellen, gehen Sie wie folgt vor:

1. Fügen Sie einen Folder Node unter dem Node Omron Device/Channels hinzu, oder machen Sie einen Rechtsklick auf den Omron Device/Channels-Node und wählen Sie Add Channel aus.
2. Tragen Sie im Add Channel-Dialog die Settings für die Omron Verbindung ein.
3. Nachdem Sie „OK“ geklickt haben, wird die Channel-Node erstellt.
4. Sie können den Channel starten, indem Sie die Channel-Node auswählen und den Startbutton klicken.

## Variablen

Unter dem Variables-Node können Sie Datenpunktnodes erstellen, die auf dem Gerät gelesen und geschrieben werden.

Die Value Type-Eigenschaft muss dabei auf den entsprechenden Variablentyp festgelegt werden (Boolean, Byte, Int16, UInt16, Int32, UInt32 und die zugehörigen Array-Typen).

## Path

Über die Path-Eigenschaft des Nodes wird die SPS-Adresse der Variable im Geräta und die Länge der Daten (bei Arrays) definiert:

`<Area><StartAddress>.<BitNumber>,<Length>`

Platzhalter	Beschreibung	Mögliche Werte (Bedeutung)	Optional (Standard Wert)
<Area>	Speicherbereich im Gerät	CIO - Core Input Output D,DM - Data Memory E,EM - Extended Memoy W,WR - Work H,HR - Holding Relay A,AR - Auxiliary Relay T,TPV,T.PV - Time Present Value TCF,T.CF - Timer Completion Flag C,CPV,C.PV - Counter Present Value CCF,C.CF - Counter Completion Flag TK,TK.Status - Task Flag Status TK.Bit - Task Flag Bit	Ja (CIO)
<StartAddress>	Startadresse der Variablen	abhängig vom Speicherbereich und Gerätetypen - siehe nachfolgende Tabellen	Nein
<BitNumber>	Nummer des Bits bei Bit-Zugriff	0 - 15	Ja (0)
<Length>	Länge bei Array-Zugriff	abhängig von Startadresse, Speicherbereich und Gerätetypen - siehe nachfolgende Tabellen	Ja (1)

**CS / CJ / CP / NSJ-series**

Area	Data type	Path	Memory area address (dec)	READ	WRITE
<b>CIO</b>	Bit	CIOXXXX.XX	CIO000000 - CIO614315	X	X
	UInt16 / Int16	CIOXXXX	CIO0000 - CIO6143	X	X
<b>Work</b>	Bit	WXXXXX.XX	W00000 - W51115	X	X
	UInt16 / Int16	WXXX	W000 - W511	X	X
<b>Auxillary Bit</b>	Bit	AXXX.XX	A00000 - A44715	X	
			A44800 - A95915	X	X
	UInt16 / Int16	AXXX	A000 - A447	X	-
			A448 - A959	X	X
<b>Timer</b>	Completion Flag (Bit)	TCFXXXX oder T.CFXXXX	T0000 - T4095	X	-
	Present Value (Int16)	TXXXX oder T.PVXXXX	T0000 - T4095	X	X
<b>Counter</b>	Completion Flag (Bit)	CCFXXXX oder C.CFXXX	C0000 - C4095	X	-
	Present Value (Int16)	CXXXX oder C.PVXXXX	C0000 - C4095	X	X

**CVM1 / CV-series**

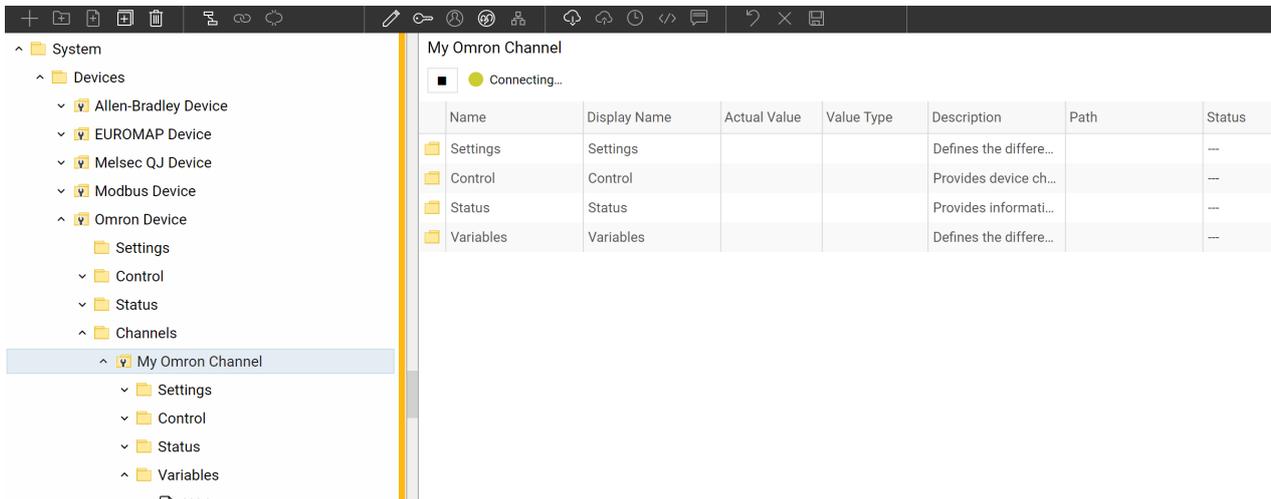
Area	Data type	Path	Memory area address (dec)	READ	WRITE
<b>CIO</b>	Bit	CIOXXXX.XX	CIO000000 - CIO255515	X	X
	UInt16 / Int16	CIOXXXX	CIO0000 - CIO2555	X	X
<b>Auxillary Bit</b>	Bit	AXXX.XX	A00000 - A44715	X	-
			A44800 - A95915	X	X
	UInt16 / Int16	AXXX	A000 - A447	X	-
			A448 - A959	X	X
<b>Timer</b>	Completion Flag (Bit)	TCFXXXX oder T.CFXXXX	T0000 - T1023/12047	X	-
	Present Value (Int 16)	TXXXX oder T.PVXXXX	T0000 - T2047 / T0000 - T1023	X	X
<b>Counter</b>	Completion Flag (Bit)	CCFXXXX	T0000 - C2047 / C0000 - C1023	X	-
	Present Value (Int 16)	CXXXX oder C.PVXXXX	C0000 - C1023/C2047	X	X

## Fehlerdiagnose

Das Omron Device Plugin liefert je nach zu untersuchender Schicht verschiedene Statusinformationen. Generell werden die kanalbasierten Diagnoseinformationen durch den Verbindungsstatus des Channels zur SPS produziert. Die variablenbasierten Diagnoseinformationen werden während des Lese-/Schreibzugriffs auf die verschiedenen Variablen produziert.

**Kanal**

Um den Status von verschiedenen SPS-Kanälen zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf das folgende Bild:



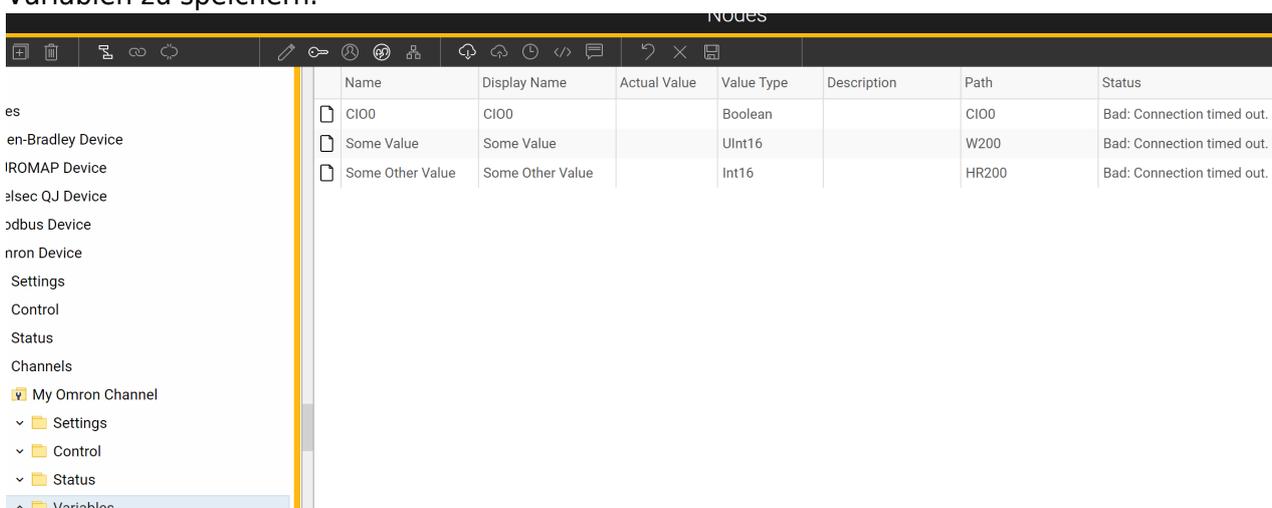
Das obige Bild zeigt das Bedienfeld des SPS-Kanals, das alle statusrelevanten Informationen anzeigt. Das Bedienfeld aktualisiert automatisch seine Statusinformation, wenn ein neuer Status verfügbar ist.

### Statuskreis

Farbe	Bedeutung
	Der Kanal ist gestoppt. Klicken Sie den ►-Button, um ihn zu starten.
	Der Kanal startet oder stoppt gerade oder wartet auf den Verbindungsaufbau.
	Der Kanal läuft und es wurde erfolgreich eine Verbindung hergestellt. Sie können durch Klick auf den ■-Button anhalten.
	Der Kanal läuft, aber die Verbindung ist momentan fehlerhaft. Bitte überprüfen Sie den Statustext für weitere Informationen.

### Variablen

Um den Status der verschiedenen Variablen zu überwachen und zu diagnostizieren, werfen Sie einen Blick auf die in Codabix® angezeigte Status-Eigenschaft der Spalte. Benutzen die den Button „Read actual Value“, um die Werte aus der SPS auszulesen und das Ergebnis in den Variablen zu speichern.



### Logdatei

Alle kanalbezogenen Statusinformationen werden auch in die kanalspezifische Logdatei im [LoggingFolder] protokolliert. Jede Logdatei wird nach dem Namensschema Omron

Device.<ChannelName>.log benannt. Der Inhalt einer solchen Logdatei kann wie folgt aussehen:

```
...  
2018-04-11 11:32:37.0 +2: [Error] Error (Severity=High): Code=[-1],  
Text=[The operation has timed-out.], Details=[]  
...
```

## Entities

Wie jedes Device Plugin erweitert das Omron Device Plugin das grundlegende Codabix® [Device-Modell](#).

### Device

Der Devicetyp `OmronDevice` des Plugins definiert auch den `OmronDeviceChannel` und erweitert somit die grundlegenden `CodabixDevice`- und `CodabixDeviceChannel`-Entities. Während das `OmronDevice` lediglich eine Konkretisierung des `CodabixDevice` repräsentiert, erweitert der `OmronDeviceChannel` den `CodabixDeviceChannel` mit Omron Variablen-Entities.

### Channel

Jeder Channel wird von einem Channel Worker behandelt, der eine physische Verbindung zur SPS herstellt. Zum Zweck der Fehlerdiagnose untersucht der Worker die SPS-Verbindung alle 10 Sekunden, um den Statuscode des Channels und die Beschreibung zu aktualisieren, damit Verbindungsausfälle aufgespürt werden.

Standardmäßig liest der Worker keine Werte. Wenn ein Client oder Plugin einen synchronen Lesevorgang des Channels anfordert, liest der Channel Worker die Variablen in Codabix® (z.B. unter Verwendung der Codabix® Webkonfigurations-Funktion „Read actual value“) aus der SPS und schreibt sie dann in die entsprechenden Codabix® Nodes.

Ähnlich schreibt der Channel Worker die Werte in die SPS, die ein Client oder Plugin in die Variablen des Channels schreibt. Beachten Sie: Nachdem die Werte in die Codabix® Nodes geschrieben wurden, werden sie solange die alten Werte beinhalten, bis der Channel Worker die Variablen zurück aus der SPS ausgelesen hat.

Um eine SPS-Variable stetig gelesen zu bekommen, müssen Sie den Node in der Webkonfiguration bearbeiten und „History Options“ auf Yes stellen (was eine interne Subscription erstellt), oder Sie können z.B. einen OPC UA Client verbunden mit dem OPC UA Server Interface Plugin benutzen und eine Subscription für die Omron-Variablenodes anlegen. In diesem Fall liest der Channel Worker die Variablen in einem gleichmäßigen Intervall aus der SPS und, falls der Wert einer der Variablen sich verändert hat, schreibt den neuen Wert in den entsprechenden CoDaBix® Node.

### Variable

Jede Omron-Variable kann anhand eines SPS-Adressoperanden und der Startadresse auf den SPS-Speicher zugreifen. Dabei werden die Variablenformate Skalar und Array unterstützt.

## Ordner & Dateien

### Ordner

Inhalt	Pfad	Zweck / Verwendung
AssemblyFolder	<CodabixInstallDir>/plugins/OmronDevicePlugin/	Beinhaltet die Plugin-Assemblydatei.
LoggingFolder	<CodabixDataDir>/log/	Beinhaltet die Plugin-Logdateien.

### Dateien

Typ	Pfad	Zweck / Verwendung
Assembly	[AssemblyFolder]/CoDaBix.OmronDevicePlugin.dll	Die Plugin-Assemblydatei.
Logging	[LoggingFolder]/Omron Device.<ChannelName>.log	Die Logdatei.

## Versionsinformation

### Dieses Dokument

<b>Datum</b>	2021-10-27
<b>Version</b>	1.0.1

### Plugin

<b>Name</b>	Omron Device Plugin
<b>Node</b>	/System/Device/Omron Device
<b>Version</b>	1.10.0

### Assembly

<b>Name</b>	CoDaBix.OmronDevicePlugin.dll
<b>Datum</b>	2021-01-18
<b>Version</b>	1.10.0.0

From:

<https://www.codabix.com/> - **CoDaBix®**

Permanent link:

<https://www.codabix.com/de/plugins/device/omrondeviceplugin>

Last update: **2021/10/27 15:17**