

# Inhaltsverzeichnis

|                                      |   |
|--------------------------------------|---|
| Voraussetzungen .....                | 1 |
| <b>Was tut das Plugin?</b> .....     | 1 |
| <b>Funktionen</b> .....              | 1 |
| <b>Unterstützte Geräte</b> .....     | 1 |
| <b>Zweck &amp; Anwendung</b> .....   | 2 |
| Zweck .....                          | 2 |
| Anwendung .....                      | 2 |
| <b>Installation</b> .....            | 2 |
| Anforderungen .....                  | 2 |
| Melsec-SPS-Einstellungen .....       | 2 |
| <b>Konfiguration</b> .....           | 3 |
| Übersicht .....                      | 3 |
| Kanalspezifische Einstellungen ..... | 3 |
| Variablen .....                      | 4 |
| <b>Fehlerdiagnose</b> .....          | 5 |
| Kanal .....                          | 5 |
| Variablen .....                      | 6 |
| Logdatei .....                       | 6 |
| <b>Entities</b> .....                | 6 |
| Device .....                         | 6 |
| Channel .....                        | 7 |
| Variable .....                       | 7 |
| <b>Ordner &amp; Dateien</b> .....    | 7 |
| Ordner .....                         | 7 |
| Dateien .....                        | 7 |
| <b>Versionsinformation</b> .....     | 8 |
| Dieses Dokument .....                | 8 |
| Plugin .....                         | 8 |
| Assembly .....                       | 8 |



# Melsec QJ Device Plugin

Das Melsec QJ Device Plugin stellt eine Verbindung zwischen CoDaBix und physischen SPSen der Mitsubishi Melsec-Q-Serie und Melsec-A-Serie über Ethernet (TCP/IP) zur Verfügung.

## Voraussetzungen

- Dieses Plugin derzeit ist nur bei **Codabix für Windows (x64 und x86)** verfügbar.

## Was tut das Plugin?

Das Melsec QJ Device Plugin verbindet und wartet Verbindungen zu einem oder mehreren Melsec-Q- und Melsec-A-SPS-Geräten via TCP/IP.

Jede SPS kann über einen separaten Kanal (Channel) verbunden werden.

Für jeden Channel wird die Verbindung definiert durch:

- IP-Adresse
- Port
- Device-Typ (Q-Serie, A-Serie).

Die in der Melsec-SPS verfügbaren Variablen können für jeden Kanal separat konfiguriert werden. Das Plugin synchronisiert die Variablenwerte zwischen der SPS und CoDaBix®.

## Funktionen

- Lese- und Schreibzugriff auf Melsec-Datentypen (Bit, Word, DWord, Real).
- Automatisiertes Verbindungshandling, auch Auto-Reconnect.

## Unterstützte Geräte

- Mitsubishi Melsec-Q-Serie
- Mitsubishi Melsec-A-Serie

## Zweck & Anwendung

### Zweck

Die angeschlossenen Geräte können einfach mit CoDaBix® gesteuert werden. Durch die Verknüpfung der SPS-Speicher mit den in CoDaBix® definierten Nodes kann die SPS direkt mit vielen anderen Nodes, Geräten, Diensten usw., die in CoDaBix® gewartet werden, interagieren. Auch andere CoDaBix® Teilnehmer können mit den durch das Melsec Device Plugin bereitgestellten SPS-Geräten interagieren.

### Anwendung

- Dynamische Generierung von Fertigungsdaten basierend auf verschiedenen Bedingungen und Daten, die von Maschinen, Benutzern, Aufträgen, Zuständen, Diensten usw. generiert werden
- Zusätzliche Sicherheit durch die Überwachung der gesamten Anlage, einschließlich der betriebsinternen Interaktionen
- Zentrale Steuerung und Durchflussüberwachung zur Früherkennung von möglichen Störungen
- Verbesserung des Anlagenleitstands, indem die von den Geräten erzeugten Produktionsdaten überwacht und aufgezeichnet werden

## Installation

Dieses Plugin ist Bestandteil des CoDaBix® Setups. Bitte konsultieren Sie [CoDaBix® Setup und erster Start](#) für weitere Informationen darüber, wie dieses Plugin installiert und deinstalliert werden kann.

### Anforderungen

- Standardanforderungen von CoDaBix®
- aktivierte ausgehende Verbindungen via TCP/IP

### Melsec-SPS-Einstellungen

Bitte folgen Sie dieser Anleitung, um die Melsec-SPS für den Zugriff durch das Melsec QJ Device Plugin zu konfigurieren: [Melsec-SPS-Einstellungen](#)

# Konfiguration

## Übersicht

Die gesamte Melsec QJ Device Plugin-Konfiguration befindet sich unter dem Nodepfad /System/Device/Melsec QJ Device.

The screenshot shows the Node-Builder-X interface. On the left, a tree view shows the hierarchy: System > Devices > Melsec QJ Device > Channels > Kanal A. The 'Kanal A' node is selected. On the right, a table displays the configuration for 'Kanal A'.

| Name      | Display Name | Actual Value | Value Type | Description                                      | Path |
|-----------|--------------|--------------|------------|--------------------------------------------------|------|
| Settings  | Settings     |              |            | Defines the different settings which apply to... |      |
| Control   | Control      |              |            | Provides device channel control mechanism.       |      |
| Status    | Status       |              |            | Provides information about the device chann...   |      |
| Variables | Variables    |              |            |                                                  |      |

Der Nodebaum im oberen Bild zeigt den Standardnodebaum des Melsec QJ Device Plugins. Um eine oder mehrere Melsec-QJ-Kanäle aufzusetzen, fügen Sie einen Folder Node unter dem Node Melsec QJ Device/Channels hinzu, oder machen Sie einen Rechtsklick auf den Melsec QJ Device/Channels-Node und wählen Sie Add Channel aus.

The screenshot shows the 'Add Channel' dialog box open over the configuration tree. The dialog box has the following fields:

- Name: (empty)
- Display Name: (empty)
- Address: 127.0.0.1
- Port: 5002
- Device Type: Q-PLC (dropdown menu)
- Network Number: 0
- PC Number: 255

At the bottom right of the dialog box, there are checkmark and X icons for confirmation and cancellation.

## Kanalspezifische Einstellungen

| Name           | Typ     | Beschreibung                                                                                                                                  |
|----------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Address        | String  | Die IP-Adresse der Melsec-SPS, zu der die Verbindung hergestellt werden soll.                                                                 |
| Port           | Integer | Der Port, zu dem die Verbindung hergestellt werden soll. Standardwert: 5002                                                                   |
| Device Type    | Enum    | Der Typ der Melsec-SPS, zu dem die Verbindung hergestellt werden soll.<br>Gültige Werte: Q - PLC (Standard) für Q-Serie, A - PLC für A-Serie. |
| Network Number | Integer | Die Netzwerknummer der Melsec-SPS. Normalerweise ist dies 0.                                                                                  |
| PC Number      | Integer | Die PC-Nummer für die Melsec-SPS. Normalerweise ist dies 255 (für 0xFF).                                                                      |

Nachdem Sie „Save“ geklickt haben, wird die Kanalnode erstellt. Sie können den Kanal starten, indem Sie die Kanalnode auswählen und den Startbutton klicken:

## Variablen

Unter dem Variables-Node können Sie Datenpunktnodes erstellen, die mit der SPS verbunden werden.

Die Value Type-Eigenschaft muss dabei auf den Variablentyp festgelegt werden (Boolean, Int16, UInt16, Int32, UInt32, Double, oder die entsprechenden Arraytypen).

Die Path-Eigenschaft des Nodes muss dazu die SPS-Adresse enthalten (Datenbereich und Startnummer, diese ist je nach Datenbereich dezimal oder hexadezimal).

Für den bitweisen Zugriff auf Datenbereiche, die keinen Bitzugriff unterstützen, muss zusätzlich die (dezimale) Bitnummer (0-15) nach dem Punkt angegeben werden, z.B. D 100.2. Bei Verwendung eines Arraytyps (z.B. Int16-Array) muss im Path zusätzlich nach dem Komma die Arraylänge angegeben werden.

## Beispiele:

| Value Type | Path     | Erklärung                                             |
|------------|----------|-------------------------------------------------------|
| Int16      | D 100    | liest 2-Byte-Wort an der Wortadresse 100              |
| Boolean    | D 100.15 | liest Bit 15 an der Wortadresse 100                   |
| Boolean    | M 123    | direkter Bitzugriff auf Bitnummer 123                 |
| Boolean    | W AF.10  | liest Bit 10 an der Wortadresse AF <sub>H</sub> (175) |

| Value Type    | Path        | Erklärung                                     |
|---------------|-------------|-----------------------------------------------|
| Double        | D 200       | liest 4-Byte-Real-Wert an der Wortadresse 200 |
| Int32-Array   | D 220, 10   | liest 10 4-Byte-Werte ab der Wortadresse 220  |
| Boolean-Array | D 100.15, 3 | liest die Bits 100.15, 101.0, 101.1           |

|                          | Name          | Display Name  | Actual Value | Value Type | Description | Path   | Status |
|--------------------------|---------------|---------------|--------------|------------|-------------|--------|--------|
| <input type="checkbox"/> | DataWord 10   | DataWord 10   | 1490         | Int16      |             | D 10   | Good   |
| <input type="checkbox"/> | DataDWord 10  | DataDWord 10  | 363333074    | Int32      |             | D 10   | Good   |
| <input type="checkbox"/> | DataWord 11   | DataWord 11   | 5544         | Int16      |             | D 11   | Good   |
| <input type="checkbox"/> | DataBit 10.0  | DataBit 10.0  | False        | Boolean    |             | D 10   | Good   |
| <input type="checkbox"/> | DataBit 10.1  | DataBit 10.1  | True         | Boolean    |             | D 10.1 | Good   |
| <input type="checkbox"/> | DataBit 10.2  | DataBit 10.2  | False        | Boolean    |             | D 10.2 | Good   |
| <input type="checkbox"/> | DataBit 10.8  | DataBit 10.8  | True         | Boolean    |             | D 10.8 | Good   |
| <input type="checkbox"/> | MerkerWord 14 | MerkerWord 14 | 64           | Int16      |             | M 14   | Good   |
| <input type="checkbox"/> | MerkerWord 15 | MerkerWord 15 | 0            | Int16      |             | M 15   | Good   |
| <input type="checkbox"/> | MerkerWord 16 | MerkerWord 16 | 0            | Int16      |             | M 16   | Good   |

## Fehlerdiagnose

Das Melsec QJ 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:

- System
  - Devices
    - Melsec QJ Device
      - Settings
      - Control
      - Status
      - Channels
        - Kanal A
        - Kanal B**
          - Settings
          - Control
          - Status
          - Variables

**Kanal B**

☒ ● The operation has timed-out.

|                          | Name      | Display Name | Actual Value | Value Type | Description        |
|--------------------------|-----------|--------------|--------------|------------|--------------------|
| <input type="checkbox"/> | Settings  | Settings     |              |            | Defines the dif... |
| <input type="checkbox"/> | Control   | Control      |              |            | Provides devic...  |
| <input type="checkbox"/> | Status    | Status       |              |            | Provides infor...  |
| <input type="checkbox"/> | Variables | Variables    |              |            |                    |

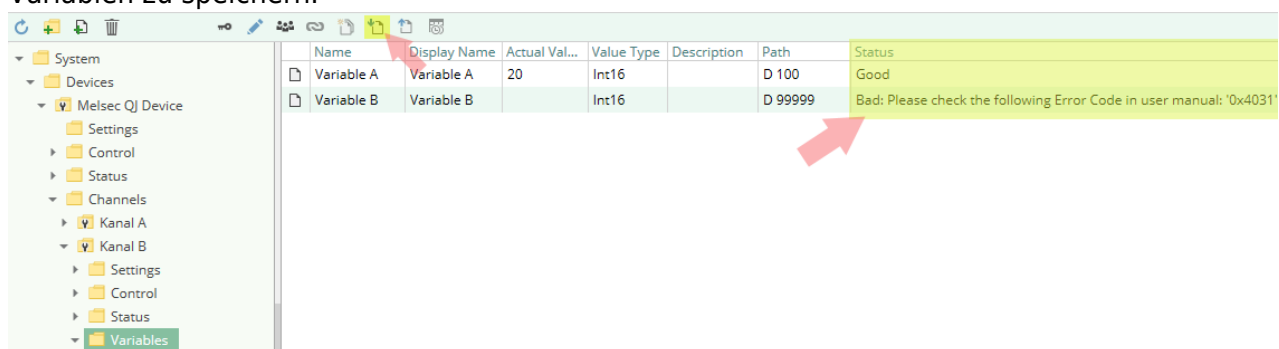
Das obige Bild zeigt das Bedienfeld des SPS-Kanals, das alle statusrelevanten Informationen anzeigt. Das Bedienfeld aktualisiert automatisch seine Statusinformation, wenn ein neuer Status verfügbar ist.

## 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 ihn durch Klick auf den ■-Button stoppen.   |
|  | 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 Sie den Button „Read actual Value“, um die Werte aus der SPS auszulesen und das Ergebnis in den Variablen zu speichern.



| Name       | Display Name | Actual Val... | Value Type | Description | Path    | Status                                                              |
|------------|--------------|---------------|------------|-------------|---------|---------------------------------------------------------------------|
| Variable A | Variable A   | 20            | Int16      |             | D 100   | Good                                                                |
| Variable B | Variable B   |               | Int16      |             | D 99999 | Bad: Please check the following Error Code in user manual: '0x4031' |

## Logdatei

Alle kanalbezogenen Statusinformationen werden auch in die kanalspezifische Logdatei im [LoggingFolder] protokolliert. Jede Logdatei wird nach dem Namensschema Melsec QJ 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 Melsec QJ Device Plugin das grundlegende CoDaBix Device-Modell.

### Device

Der Devicetyp MelsecDevice des Plugins definiert auch den MelsecDeviceChannel und

erweitert somit die grundlegenden CodabixDevice- und CodabixDeviceChannel-Entities. Während das MelsecDevice lediglich eine Konkretisierung des CodabixDevice repräsentiert, erweitert der MelsecDeviceChannel den CodabixExchangeChannel mit SQL Tabellenentities.

## 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.

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 Melsec-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 Melsec-Variable kann anhand eines SPS-Adressoperanden und der Startadresse auf den SPS-Speicher zugreifen. Die Interpretation hängt jedoch von der Value Type-Auswahl ab. Unterstützte Variablenformate sind Skalar- und Arraytypen.

# Ordner & Dateien

## Ordner

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

## Dateien

| Typ      | Pfad                                                        | Zweck / Verwendung                             |
|----------|-------------------------------------------------------------|------------------------------------------------|
| Assembly | [AssemblyFolder]/CoDaBix.MelsecDevicePlugin.dll             | Die Plugin-Assemblydatei.                      |
| DLL      | [AssemblyFolder]/native/windows-x86/codabix-melsecqjdll.dll | Die native Plugin-DLL-Datei für Windows (x86). |
| DLL      | [AssemblyFolder]/native/windows-x64/codabix-melsecqjdll.dll | Die native Plugin-DLL-Datei für Windows (x64). |
| Logging  | [LoggingFolder]/Melsec QJ Device.<ChannelName>.log          | Die Logdatei.                                  |

## Versionsinformation

### Dieses Dokument

|                |            |
|----------------|------------|
| <b>Datum</b>   | 2018-04-13 |
| <b>Version</b> | 1.0        |

### Plugin

|                |                                 |
|----------------|---------------------------------|
| <b>Name</b>    | Melsec QJ Device Plugin         |
| <b>Node</b>    | /System/Device/Melsec QJ Device |
| <b>Version</b> | 1.0.0                           |

### Assembly

|                |                                |
|----------------|--------------------------------|
| <b>Name</b>    | CoDaBix.MelsecDevicePlugin.dll |
| <b>Datum</b>   | 2018-04-13                     |
| <b>Version</b> | 1.0.0.0                        |

From:

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

Permanent link:

<https://www.codabix.de/de/plugins/device/melsecdeviceplugin>

Last update: **2021/07/30 13:39**