

# Inhaltsverzeichnis

- Voraussetzungen** ..... 1
  - Systemanforderungen ..... 1
  - Erforderliche Plugins ..... 1
- 1. Schritt: Einrichten von CoDaBix** ..... 2
  - 1.1 Installation ..... 2
  - 1.2 Konfiguration ..... 2
- 2. Schritt: Einrichten der Prozessdatenerfassung** ..... 3
  - 2.1 Anlegen eines neuen Transfers ..... 4
  - 2.2 Auswahl der zu erfassenden Daten ..... 6
  - 2.3 Auswahl der Datenziele ..... 9
  - 2.4 Festlegen des Triggers ..... 10
- 3. Schritt: Aktivieren und Funktionalität verifizieren** ..... 11
  - 3.1 Ansicht im SQLite-Browser ..... 12
  - 3.2 Ansicht im CoDaBix Dashboard ..... 15



# Erfassen von Prozessdaten

Diese Anleitung beschreibt das „Schritt für Schritt“-Vorgehen, um Prozessdaten basierend auf definierbaren Triggern in eine Datenbank zu schreiben.

Als Datenquelle wird hierbei beispielhaft eine **SIMATIC S7 SPS** und als Datenziel eine lokale **SQLite Datenbank** verwendet.

Durch das einheitliche Interface, das CoDaBix für den Zugriff auf angebundene Geräte und Datenbanken bietet, ist dieses Vorgehen jedoch auf **jede Art von Daten**, die in CoDaBix definiert sind, anwendbar.

## Voraussetzungen

### Systemanforderungen

Für die Ausführung der folgenden Schritte benötigen Sie einen Rechner (mit Windows oder Linux Betriebssystem), auf dem Sie die erforderlichen Rechte besitzen, um Anwendungen installieren zu können.

Desweiteren ist eine Internetverbindung notwendig, um CoDaBix und die bereitgestellte Standardkonfiguration laden und auf Beispieldaten von unserer öffentlichen SIMATIC S7 zugreifen zu können.

Die genauen Systemanforderungen an Hardware und Betriebssystemversion finden Sie hier:

- [Systemanforderungen für Windows](#)
- [Systemanforderungen für Linux](#)

### Erforderliche Plugins

Folgende CoDaBix Plugins sind erforderlich

- [S7 Device Plugin](#): Anbindung der SIMATIC S7 SPS
- [SQL Exchange Plugin](#): Verbindung zur SQLite Datenbank
- [Script Interface Plugin](#): Verarbeiten der Trigger und Transfer der Daten

# 1. Schritt: Einrichten von CoDaBix

In diesem Schritt installieren Sie CoDaBix auf Ihrem System.  
Für diese Anleitung stellen wir Ihnen eine Grundkonfiguration für die Prozessdatenerfassung bereit.

## 1.1 Installation

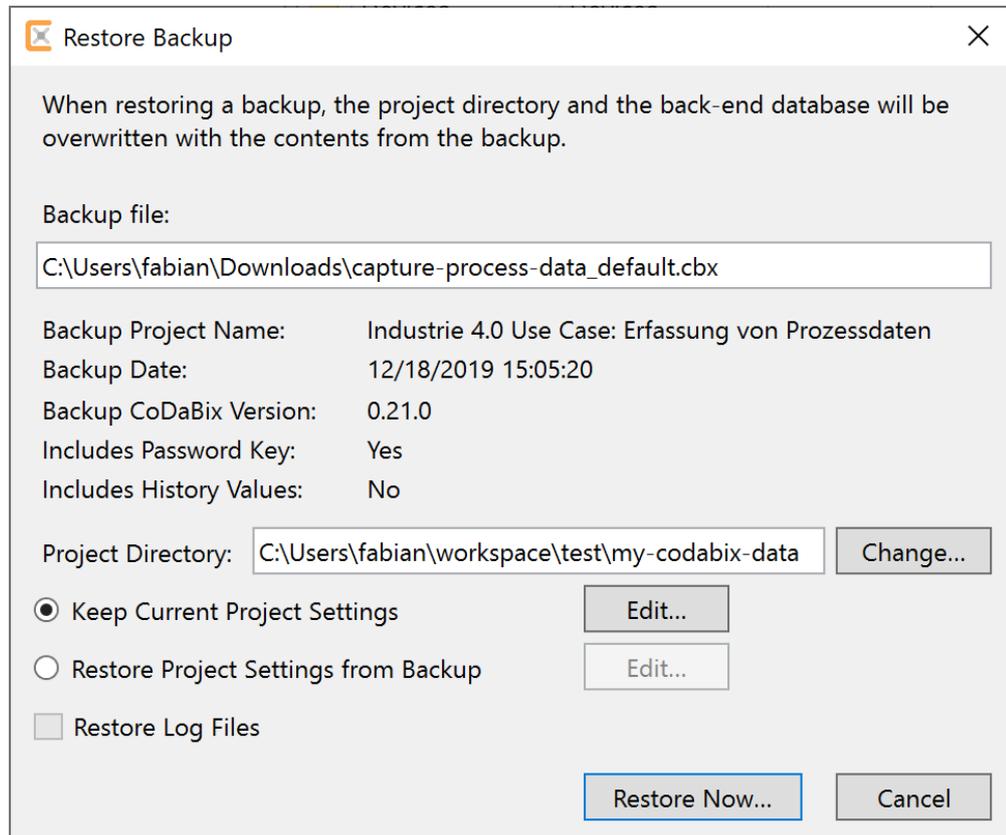
- Laden Sie im **Downloadbereich** die aktuelle Version von CoDaBix (diese Anleitung setzt **mindestens v0.21.0** voraus) für das System herunter,

auf dem Sie CoDaBix verwenden wollen.

- Nach erfolgreichem Download installieren Sie CoDaBix auf Ihrem System und folgen den Schritten für den ersten Start:
  - **Installation und erster Start unter Windows**
  - **Installation und erster Start unter Linux und Raspberry Pi**
  - **Siemens IOT2050 Image**

## 1.2 Konfiguration

- Laden Sie folgende Standardkonfiguration herunter: [capture-process-data\\_default\\_2020-12-04.cbx](#)
- Importieren Sie die Konfiguration in CoDaBix
  - Windows:
    - Klicken Sie auf  rechts in der Taskleiste
    - Wählen Sie im Backup-Dialog die zuvor geladene Datei aus und starten den Import durch Klicken auf Restore Now...



- Linux/Raspberry Pi:
  - Wählen Sie in der Konsolenanwendung den Menüpunkt 4) Restore Backup
  - Geben Sie den Pfad zu der geladenen Datei an und starten den Import durch drücken der Enter-Taste
- Docker:
  - Das Vorgehen zum Wiederherstellen eines Backups innerhalb eines Docker Containers finden Sie in dem zugehörigen Github-Repository:  
<https://github.com/Traeger-GmbH/codabix-docker#restoring-from-a-backup-file>

## 2. Schritt: Einrichten der Prozessdatenerfassung

In diesem Beispiel sollen die Werte von zwei Variablen (Duration und Quantity) aus der SPS gelesen und daraus ein neuer Eintrag in einer Datenbank-Tabelle erzeugt werden, der zusätzlich die ID, über welche die SPS identifiziert werden kann, und den aktuellen Zeitstempel enthält. Dieser Vorgang soll ausgeführt werden, wenn die steigende Flanke des Bits JobCompleted in der SPS detektiert wird.

Die **SPS Variablen** sind wie folgt definiert:

### Prozessdaten

- Duration
  - Datentyp: Real
  - Adresse: DB511.DBD 0
- Quantity
  - Datentyp: DInt
  - Adresse: DB511.DBD 4

## Trigger

- JobCompleted
  - Datentyp: Bit
  - Adresse: DB511.DBX 10.0

Die **Datenbank-Tabelle** namens oemmachinerecord besitzt folgende Spalten:

- ID
  - Datentyp: INT
- MachineID
  - Datentyp: INT
- Duration
  - Datentyp: Double
- Quantity
  - Datentyp: INT
- Timestamp
  - Datentyp: DATETIME

Um die Datenerfassung (nachfolgend Transfer genannt) in CoDaBix abzubilden, wird sie in drei Komponenten eingeteilt:

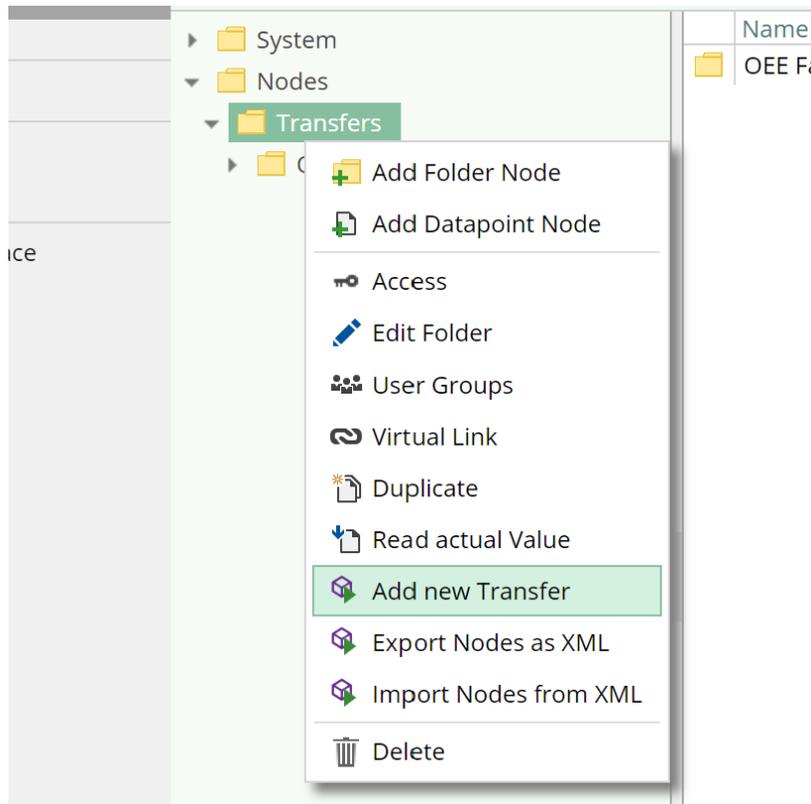
- **Inputs:** Variablen, deren Werte erfasst werden sollen
- **Outputs:** Variablen, in welchen die zu erfassenden Daten gespeichert werden sollen
- **Triggers:** Variablen, die einen Datentransfer von den **Inputs** zu den **Outputs** auslösen

Diese Komponenten werden über eine Ordner-Struktur innerhalb von CoDaBix abgebildet. Dabei repräsentiert jeder Ordner, der unterhalb von /Nodes/Transfers liegt, einen unabhängigen Transfer.

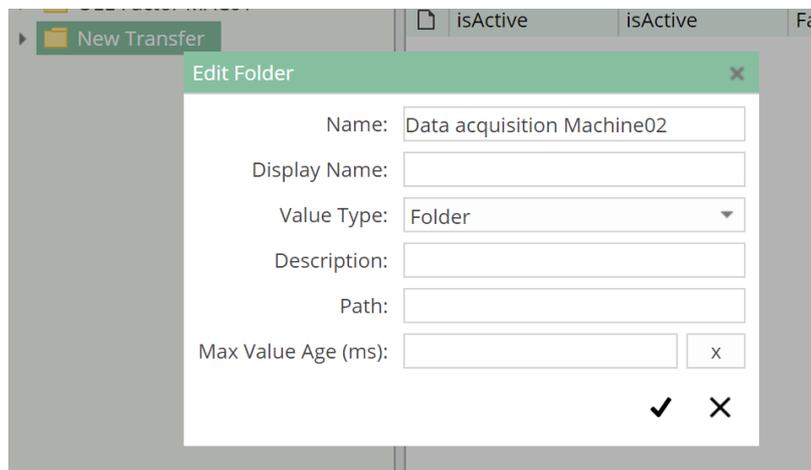
## 2.1 Anlegen eines neuen Transfers

Um einen neuen Transfer anzulegen, navigieren Sie im Node-Baum zum Node „/Nodes/Transfers“.

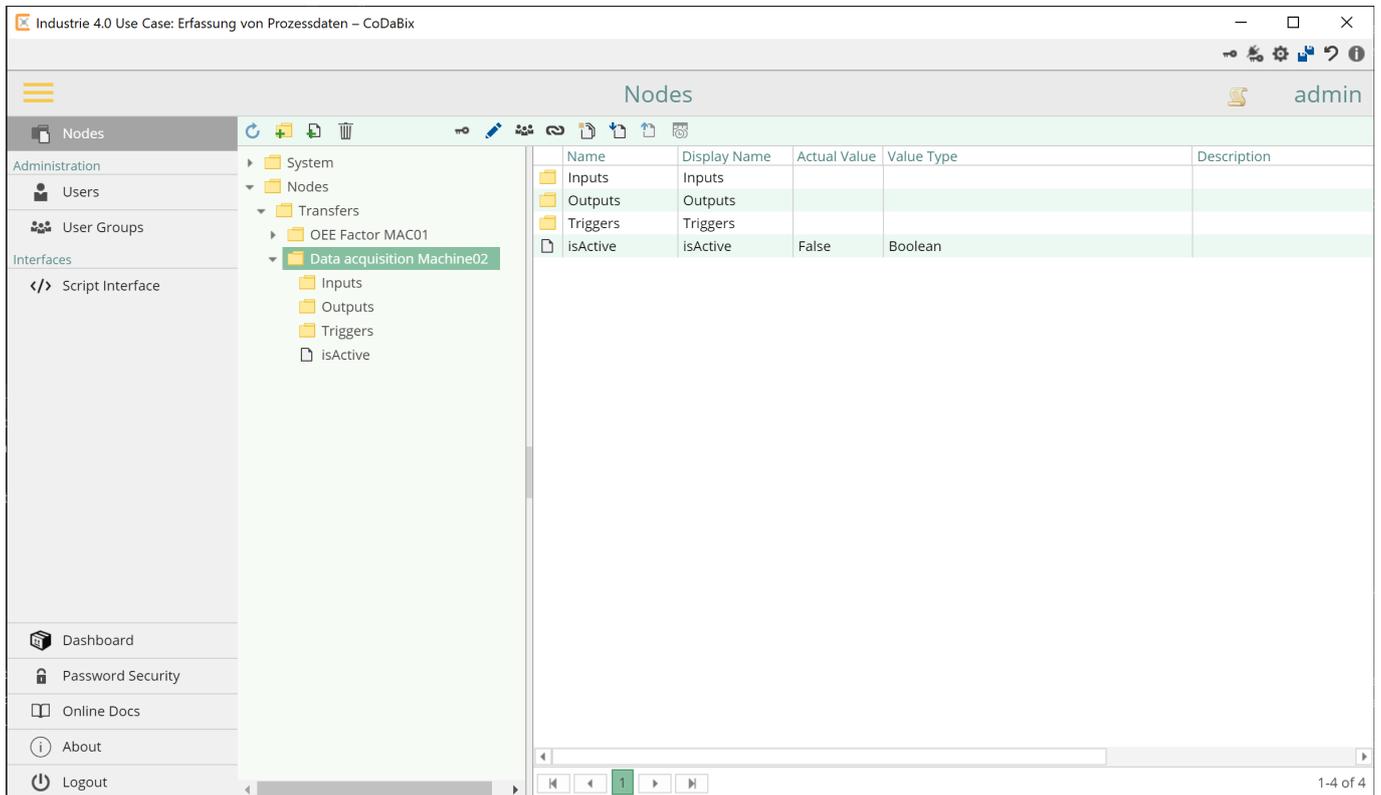
- Erstellen Sie einen neuen Transfer:
  - Klicken Sie mit der rechten Maustaste auf den Node /Nodes/Transfers
  - Klicken Sie in dem Kontextmenü auf den Eintrag Add new Transfer



- Benennen Sie den neuen Transfer:
  - Klicken Sie mit der rechten Maustaste auf den neuen Node New Transfer
  - Klicken Sie auf den Kontextmenü-Eintrag Edit Folder
  - Ändern Sie in dem Edit Folder-Dialog den Namen in Data acquisition Machine02
  - Bestätigen Sie durch Klicken auf ✓



Daraufhin sollte der Node-Baum wie folgt aussehen:



## 2.2 Auswahl der zu erfassenden Daten

Um Variablen festzulegen, deren Werte erfasst werden sollen, müssen sich diese Variablen unter dem Folder-Node Inputs befinden. Neben den Variablen aus der SPS soll die ID der Maschine (für dieses Beispiel ID = 2) erfasst werden.

Deshalb wird zuerst ein Node von dem Typen UInt16 angelegt, in dessen Wert konstant diese ID gespeichert wird:

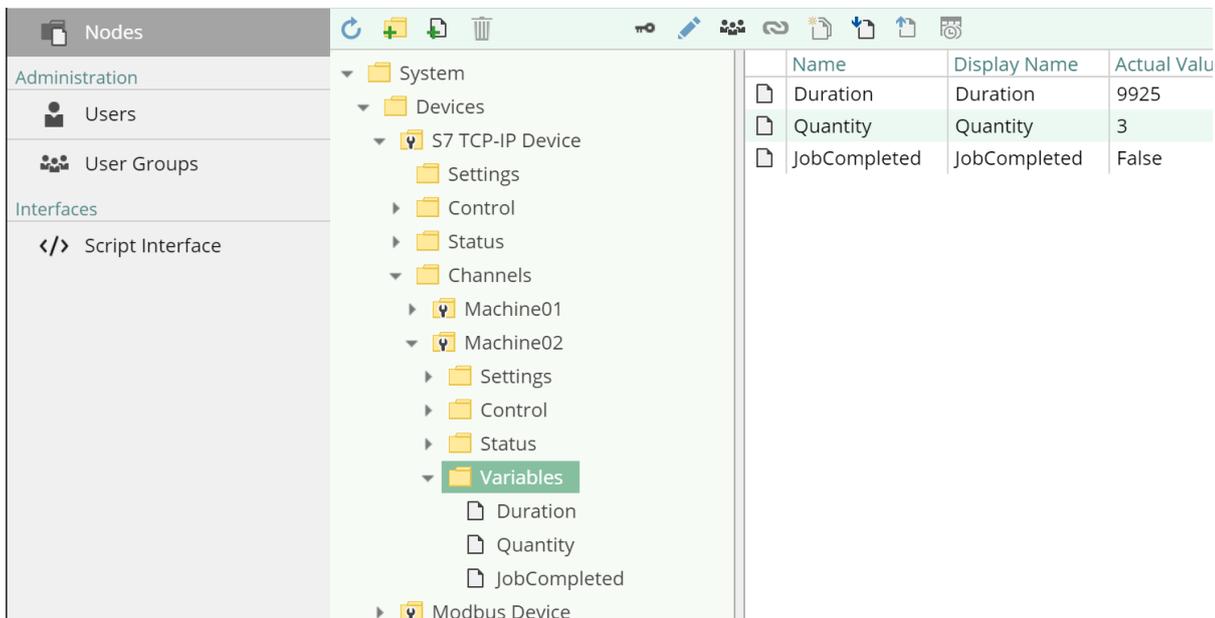
- Klicken Sie mit der rechten Maustaste auf den Node Inputs
- Wählen Sie Add Datapoint Node aus dem Kontextmenü
- Im Add Datapoint Node-Dialog:
  - Geben Sie ID of MAC02 als Name ein
  - Wählen Sie UInt16 als **Value Type**
  - Bestätigen Sie durch Klicken auf ✓

- Klicken Sie rechts auf den neuen Node
- Wählen Sie Write a new Value
- Im Write a new Value-Dialog:
  - Geben Sie den Wert 2 als neuen Wert ein
  - Bestätigen Sie mit durch Klicken auf ✓

Um die Werte der SPS Variablen als Eingänge für den Transfer zu definieren, wird im Inputs-Ordner jeweils ein verlinkter Node erstellt, dessen Linkziel den Value-Node beinhaltet, der die entsprechende SPS Variable repräsentiert.

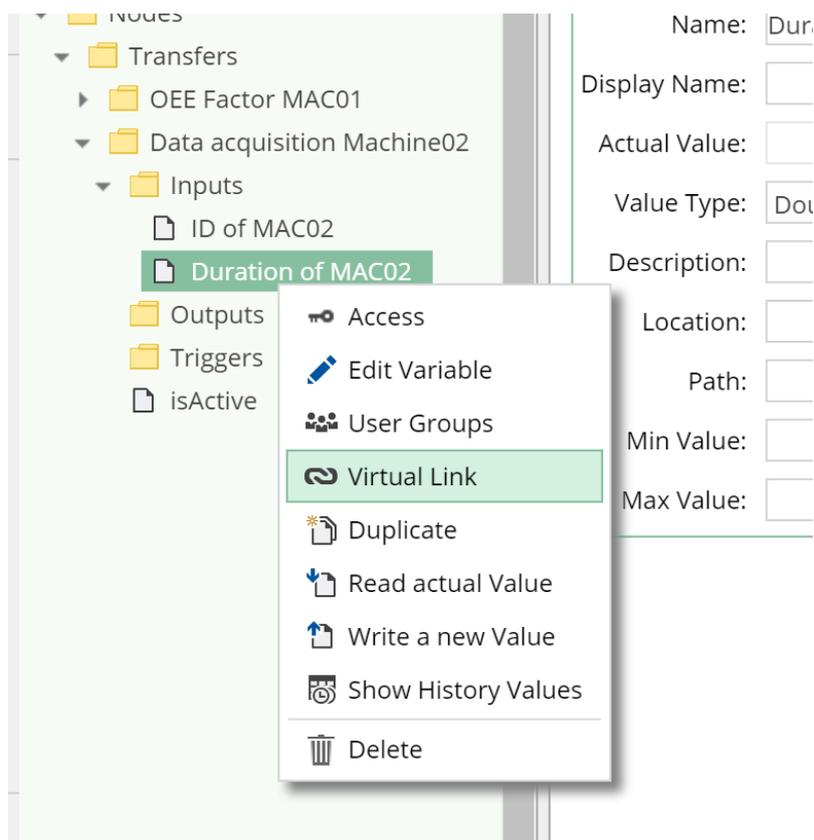
**Verlinkte Nodes** verhalten sich ähnlich zu symbolischen Links, wie sie im Unix-Dateisystem zur Verfügung stehen. Sie bieten den Vorteil, dass die Struktur von Daten unabhängig gestaltet werden kann, ohne die ursprüngliche Struktur zu beeinflussen.

Die Nodes, die diese Variablen abbilden, finden Sie unter dem Pfad /System/Devices/S7 TCP-IP Device/Channels/Machine 02/Variables:

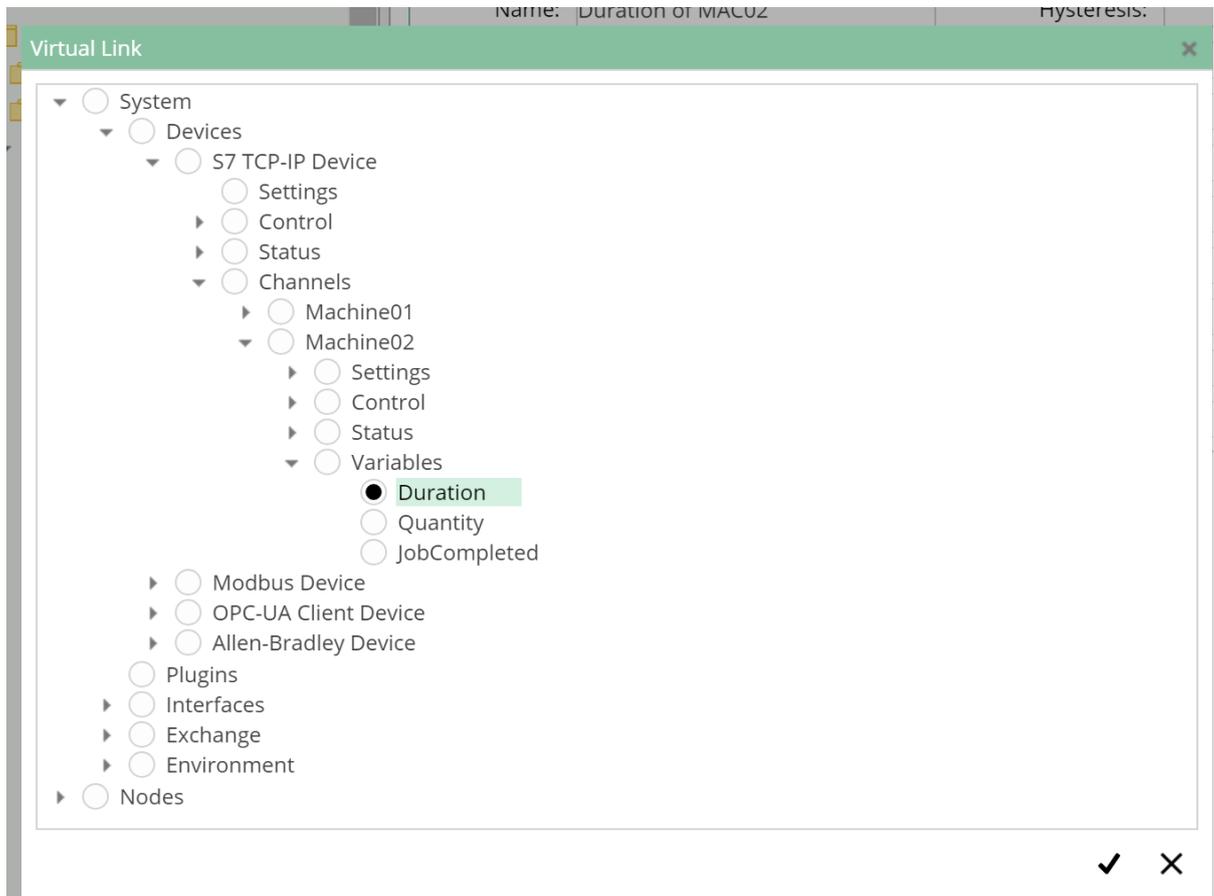


Führen Sie für diese beiden Nodes die folgenden Schritte aus, um Sie im Inputs Folder zu verlinken:

- Klicken Sie mit der rechten Maustaste auf Inputs
- Klicken Sie im Kontextmenü auf Add Datapoint Node
- Im Add Datapoint Node-Dialog:
  - Geben Sie als Namen den Namen des ursprünglichen Nodes ein (der Name des Link-Nodes ist frei wählbar, in diesem Fall soll jedoch klar ersichtlich sein, welcher Node das Linkziel ist)
  - Bestätigen Sie durch Klicken auf
- Klicken Sie rechts auf den neu erstellten Node
- Wählen Sie im Kontextmenü die Option Virtual Link



- Im Virtual Link-Dialog:
  - Navigieren Sie in dem geöffneten Dialog zu dem Node, auf den der virtuelle Link zeigen soll
  - Bestätigen Sie durch Klicken ✓



Da zusätzlich zu den SPS-Daten der aktuelle Zeitstempel gespeichert werden soll, muss noch eine Node hinzugefügt werden, die diesen Wert bereitstellt. Zu diesem Zweck gibt es in CoDaBix einen vordefinierten Value-Node, der unter dem Pfad /System/Environment/DateTime/UtcNow zu finden ist. Beim Lesen dieser Node wird immer der aktuelle Zeitstempel zurückgegeben.

Erstellen Sie nun analog zu den vorhergehenden Schritten einen weiteren Link namens Current Timestamp, der auf diesen Node (/System/Environment/DateTime/UtcNow) verweist.

## 2.3 Auswahl der Datenziele

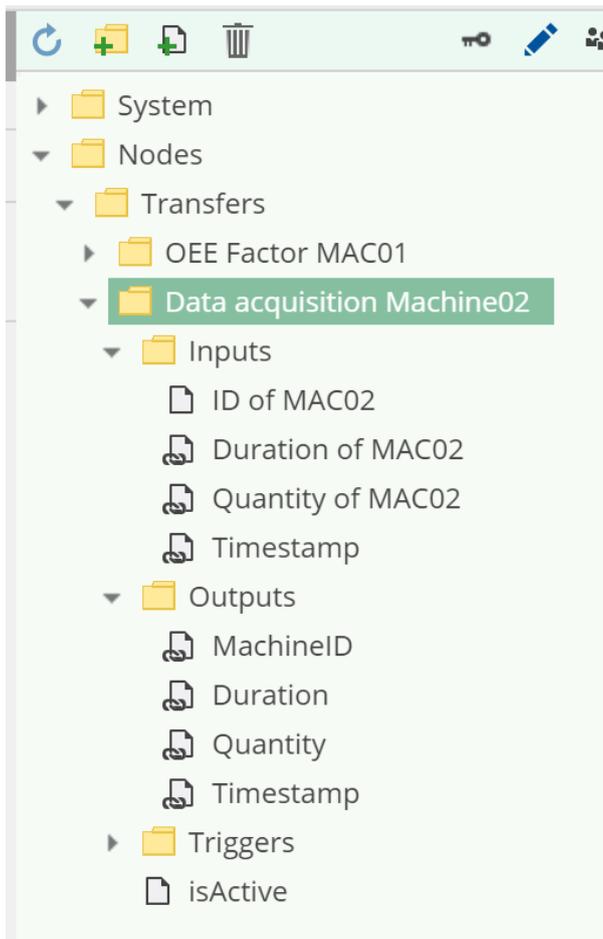
Analog zu den Quellen der Daten werden die Ziele unterhalb des Folder-Nodes Outputs definiert.

Dazu soll die Tabelle oemmachinerecord der Datenbank verwendet werden. Die Nodes, die die Spalten dieser Tabelle repräsentieren, befinden sich unter dem Pfad /System/Exchange/SQL Exchange/Databases/OEE/Tables/oemmachinerecord/Columns. Legen Sie nun, analog zum vorhergehenden Schritt, unterhalb von Outputs Linknodes an, die mit folgenden Nodes verlinkt sind:

- **MachineID:** /System/Exchange/SQL Exchange/Databases/OEE/Tables/oemmachinerecord/Columns/MachineID
- **Duration:** /System/Exchange/SQL

- Exchange/Databases/OEE/Tables/oemachinerecord/Columns/Duration
- **Quantity:** /System/Exchange/SQL  
Exchange/Databases/OEE/Tables/oemachinerecord/Columns/Quantity
- **Timestamp:** /System/Exchange/SQL  
Exchange/Databases/OEE/Tables/oemachinerecord/Columns/Timestamp

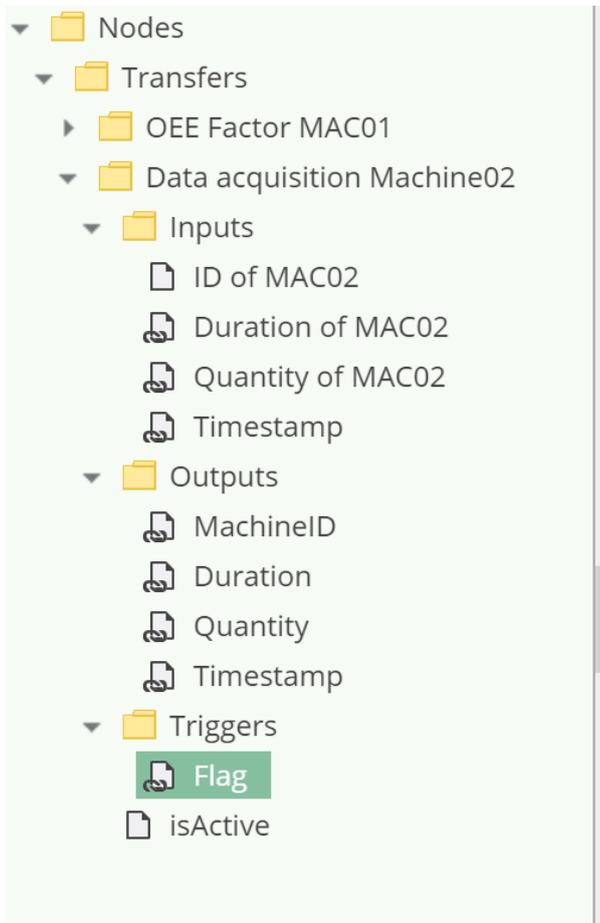
Der Transfer-Mechanismus verwendet die Reihenfolge der Input-Nodes und Output-Nodes, um zu differenzieren, welches Datum in welchen Node geschrieben werden soll. Der Wert des ersten Input-Nodes wird dabei in den ersten Output-Node geschrieben, der des zweiten in den zweiten Output-Node und so weiter. Ordnen Sie nun die Nodes unter Inputs und Outputs per Drag-and-Drop so an, dass Sie die Reihenfolge der in diesem Screenshot entsprechen:



## 2.4 Festlegen des Triggers

Um den Trigger für unser Beispiel einzurichten, legen Sie nun unter dem Triggers-Ordner einen virtuellen Link an, dessen Ziel der Node /System/Devices/S7 TCP-IP Device/Channels/Machine 01/Variables/JobCompleted ist.

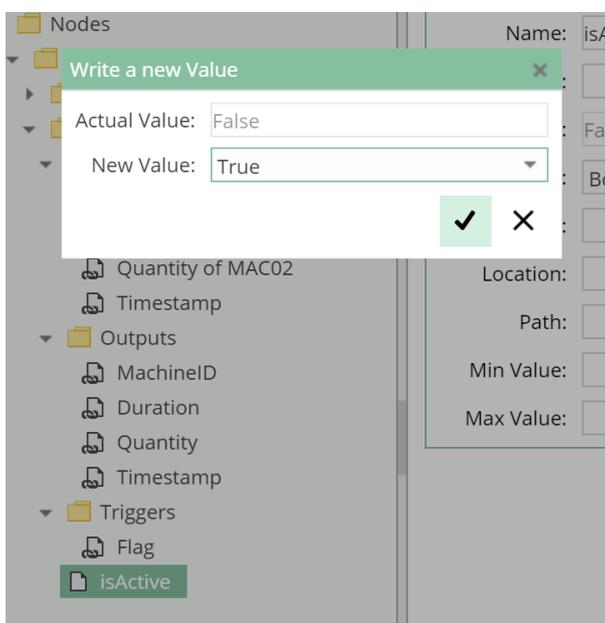
Somit sollte der Node-Baum wie folgt aussehen:



### 3. Schritt: Aktivieren und Funktionalität verifizieren

Nach Durchführung der vorangegangenen Schritte ist die Einrichtung der Datenerfassung abgeschlossen.

Um den Transfer zu aktivieren, muss der Wert des Nodes `isActive` in dem Ordner des Transfers auf `true` gesetzt werden. Schreiben Sie dazu `true` als neuen Node-Wert:



Der Trigger wird nun alle 500ms von CoDaBix gelesen und auf eine steigende Flanke geprüft. Ist dies der Fall, werden die Inputs gelesen, der Trigger zurückgesetzt (der Wert `false` geschrieben) und die Daten in die Outputs geschrieben. Dadurch wird in der Datenbank-Tabelle ein neuer Eintrag erzeugt, der diese Daten enthält.

Das SPS Programm wartet, bis der Trigger wieder zurückgesetzt wurde. Dann werden innerhalb von 10 Sekunden neue, zufällige Werte in die Variablen geschrieben und der Trigger wird erneut auf `true` gesetzt.

### 3.1 Ansicht im SQLite-Browser

Sie können die Funktionalität nun durch einen SQLite-Browser verifizieren:

- Wenn Sie nicht bereits einen SQLite-Browser auf Ihrem System installiert haben, laden Sie unter <https://sqlitebrowser.org/dl/> das für Ihr System passende .zip-Archiv
- Entpacken und starten Sie den SQLite-Browser
- Öffnen Sie die Datenbank-Datei unter folgendem Pfad  
`<CodabixProjectDirectory>/userdata/mydata.sqlite` an.  
`<CodabixProjectDirectory>` steht hierbei für den Pfad zum Projektverzeichnis,

das Sie beim ersten Start von CoDaBix ausgewählt haben.

Dieses kann über die Settings ausgelesen werden:

**Edit CoDaBix Settings**

Project Directory:

**Basic Settings**

Project Name	Industrie 4.0 Use Case: Erfassung von Prozessdaten
File Access Security	Normal
Log Level	Info
Log File Retention Days	30
Update DB Mode	Normal
Back-end Database Mode	Embedded (SQLite)

**Web Server Settings**

Web Server Mode	HTTP.sys (Windows only)
Local HTTP Port	8181
HTTP(S) Bindings	(None)
Serve Static Web Files	True
Custom HTTP Redirect URL	

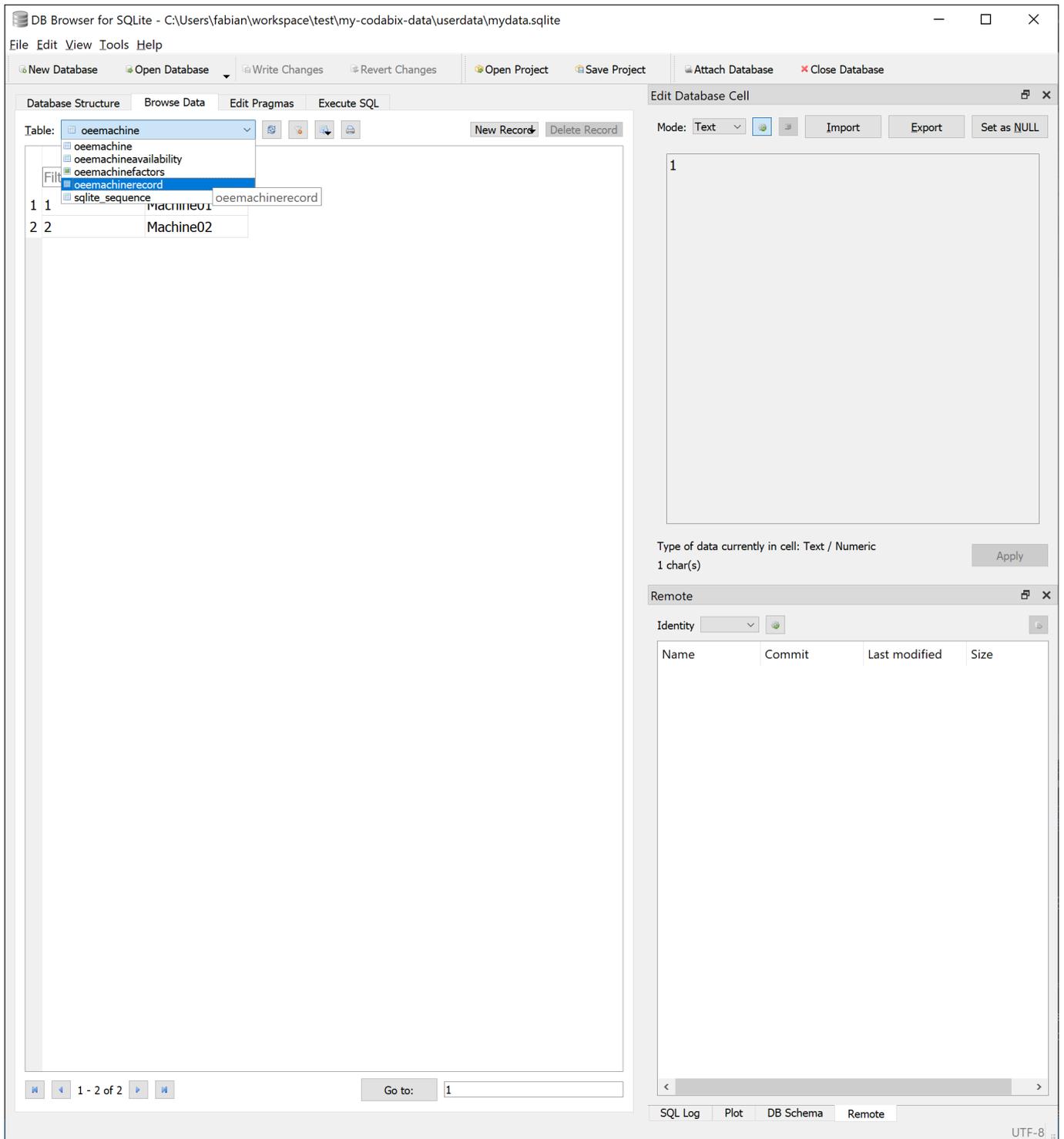
**MySQL/MariaDB/MSSQL Settings**

**Project Name**  
Allows to specify a name for this project, which will then be displayed in the UI.

**Service Management**

Service Status: Service is not installed.

- Öffnen Sie im SQLite-Browser die Tabelle oemachinerecord



- Dort sollten Sie nun bereits vorhandene Einträge vorfinden, während im Takt von etwa 10 Sekunden neue hinzukommen

The screenshot shows the DB Browser for SQLite interface. The main window displays a table named 'oemmachinerecord' with the following data:

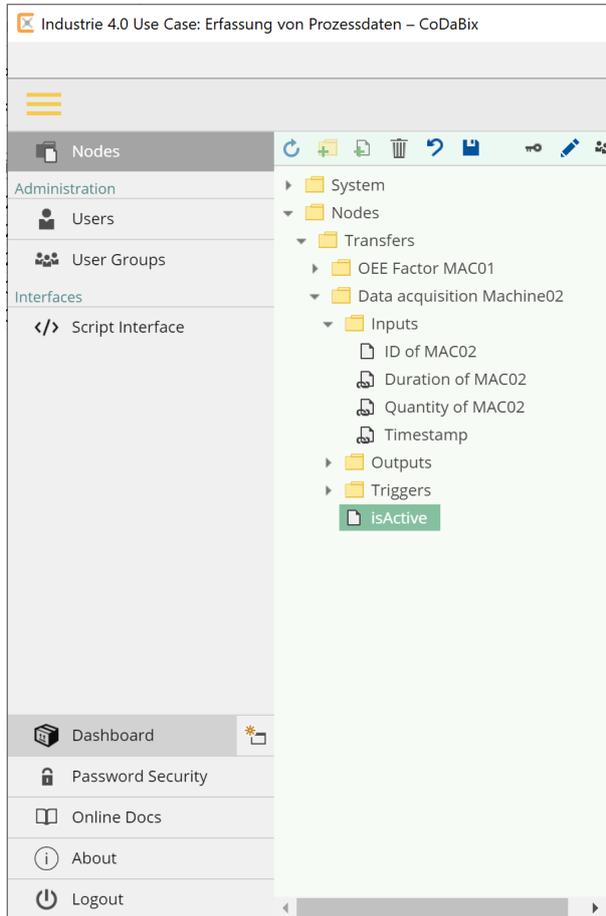
ID	MachineID	Duration	Quantity	Timestamp
1 527	2	7437.0	10	2019-12-18 15:10:41....
2 528	2	6563.0	1	2019-12-18 15:10:44....
3 529	2	7644.0	1	2019-12-18 15:10:50....
4 530	2	5375.0	7	2019-12-18 15:10:50....
5 531	2	6014.0	8	2019-12-18 15:11:09....

The 'Edit Database Cell' panel on the right shows the value '527' in a text mode. Below it, a status bar indicates 'Type of data currently in cell: Text / Numeric' and '3 char(s)'. At the bottom of the interface, there are navigation buttons and a 'Go to:' field with the value '1'. The bottom right corner shows 'UTF-8' encoding.

### 3.2 Ansicht im CoDaBix Dashboard

Eine andere Möglichkeit ist die Anzeige in der in CoDaBix integrierten Dashboard-Applikation.

- Klicken Sie hierzu in der linken Seitenleiste auf den Menüpunkt Dashboard



- Navigieren Sie hier in den Ordner Data acquisition Machine02
- Klappen Sie durch Klicken auf das  -Zeichen die drei Ordner Inputs, Outputs und Triggers auf
- Sie sehen die Änderung der Node-Werte nun in nahezu Echtzeit und können so das Auslösen und Zurücksetzen des Triggers

und den Transfer der Daten zwischen den Inputs und Outputs live beobachten:

Dashboard admin

[Nodes](#) > [Transfers](#) > Data acquisition Machine02

### Inputs

ID of MAC02	<input type="text" value="2"/>	
Duration of MAC02	<input type="text" value="5305"/>	
Quantity of MAC02	<input type="text" value="10"/>	
Timestamp	<input type="text" value="12/18/2019, 4:14 PM"/>	

### Outputs

MachineID	<input type="text" value="2"/>	
Duration	<input type="text" value="5305"/>	
Quantity	<input type="text" value="10"/>	
Timestamp	<input type="text" value="12/18/2019, 4:14 PM"/>	

### Triggers

Flag	<input type="checkbox"/>	
------	--------------------------	--

isActive

Version: 0.0.28

From:

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

Permanent link:

<https://www.codabix.com/de/tutorials/erfassen-von-prozessdaten>

Last update: **2021/04/25 17:37**