Inhalt

[03.10.2024 – Protokoll 1](#_Toc187333645)

[Motorsteuerung mit Webserver 1](#_Toc187333646)

[09.01.2025 - Protokoll 9](#_Toc187333647)

[Motoransteuerung über API-Requests 9](#_Toc187333648)

[Login 9](#_Toc187333649)

[Versuch mit Postman 9](#_Toc187333650)

[Python Code 10](#_Toc187333651)

[Login 10](#_Toc187333652)

[Browse 11](#_Toc187333653)

[Motor starten 11](#_Toc187333654)

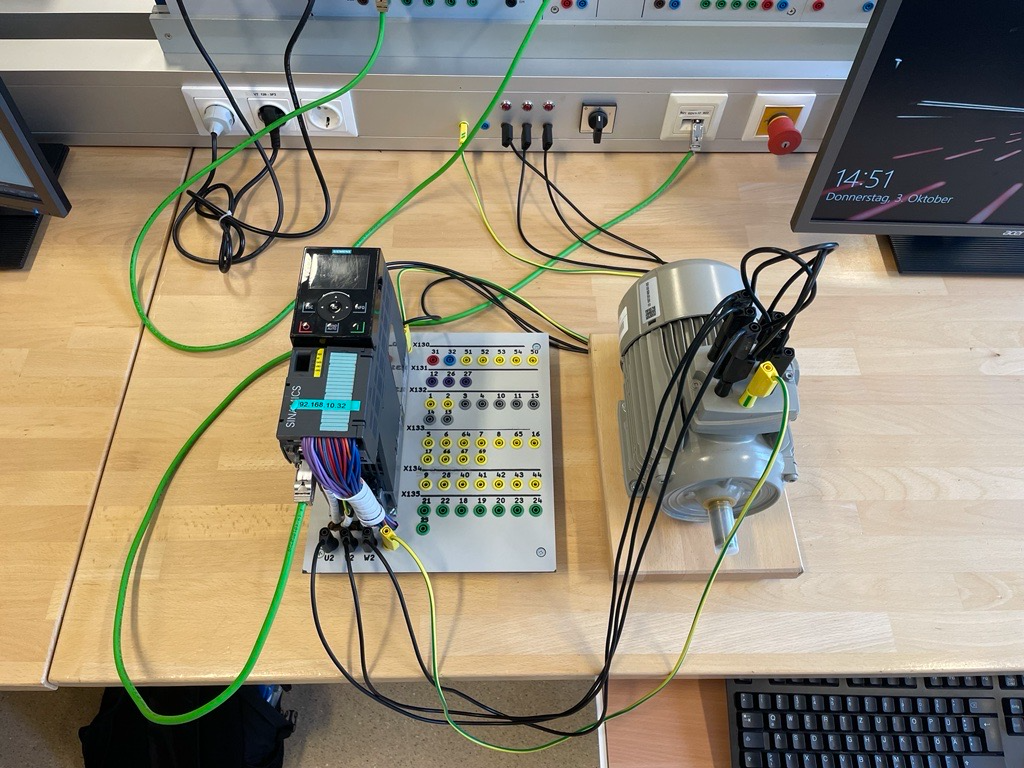
[Motor stoppen 12](#_Toc187333655)

[Erste Applikation 13](#_Toc187333656)

# 03.10.2024 – Protokoll

## Motorsteuerung mit Webserver

Aufbau:



Zuerst Namen der CPU und des FU’s (Frequenzumrichter) ändern 🡪 aufgrund der IPs die auf den Geräten stehen.

IP von FU eingeben und Subnetz auf 255.255.254.0 stellen

Ein Bild, das Text, Software, Screenshot, Computersymbol enthält.

Automatisch generierte Beschreibung

Danach auf „Online Verbinden“ klicken und hier die richtigen Schnittstellen einstellen

Ein Bild, das Text, Screenshot, Software, Display enthält.

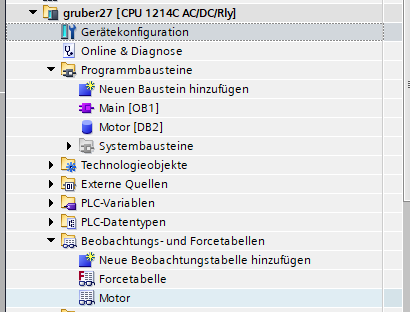
Automatisch generierte Beschreibung

* Dann auf „Suche starten“ klicken, danach auf „Verbinden“
* Soll die IP-Adresse hinzugefügt werden?: JA
* Danach Reiter „Inbetriebnahme auswählen und durchklicken bis zu den Motordaten
* Motordaten anhand von der Plakette auf dem Motor eingeben und weiter durchklicken
* Motor mit Steuertafel in Betrieb nehmen

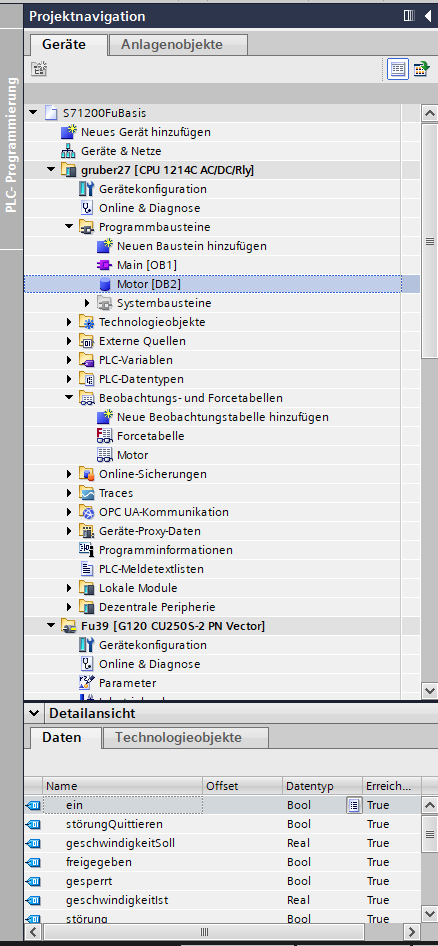
Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

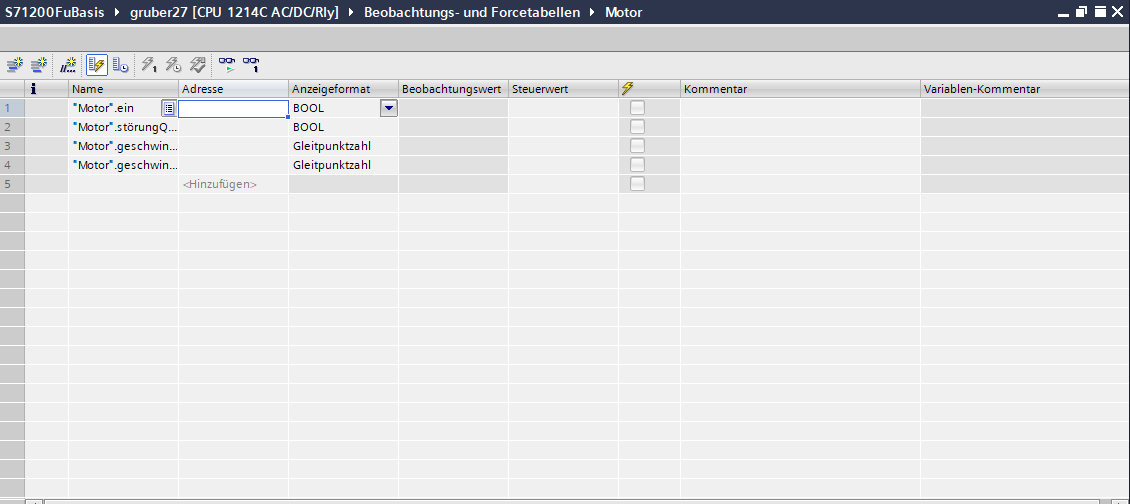
* FU mit CPUverbinden
* Rechtsklick auf FU und Gerätenamen zuweisen
* Liste aktualisieren, dann Gerät auswählen und Namen zuweisen
* Gerätekonfiguration der CPU öffnen



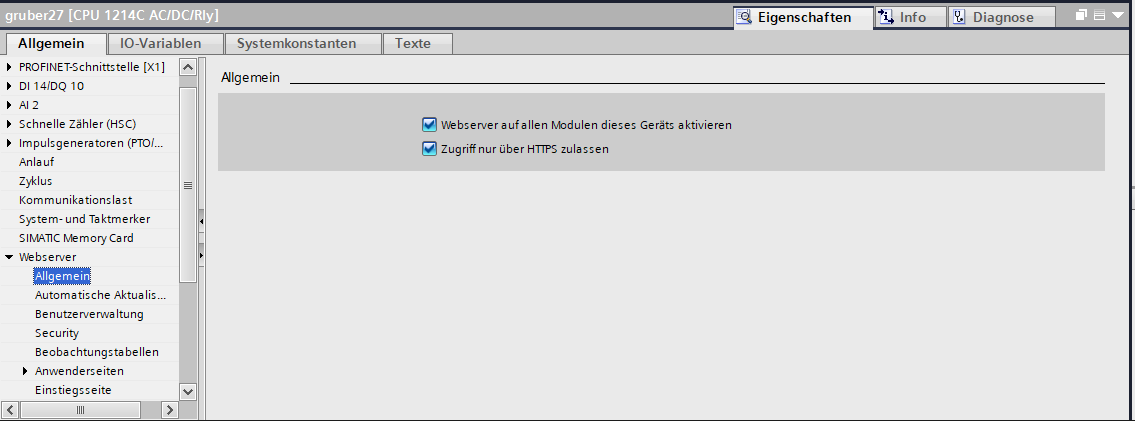
* Neue Beobachtungstabelle „Motor“ erstellen
* Programmbausteine Motor[DB2] auswählen



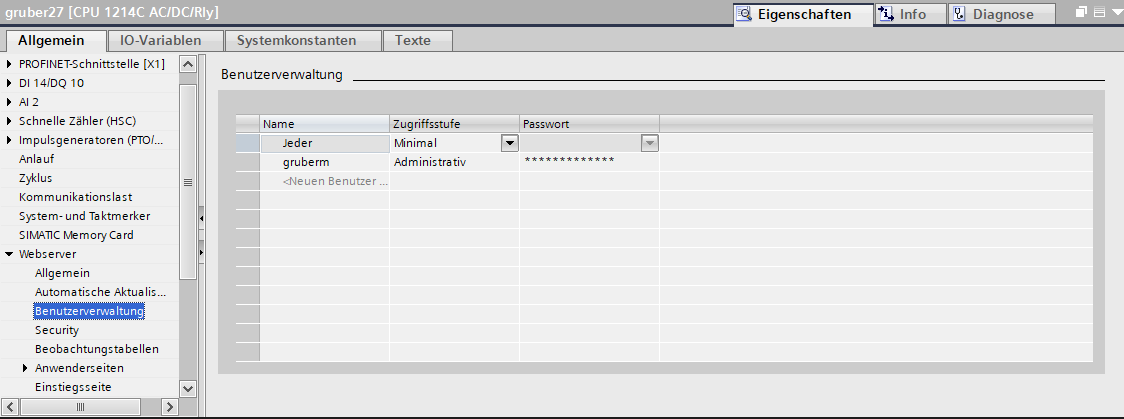
* Felder „**ein**“, „**störungQuittieren**“, „**geschwindigkeitSoll**“ und „**geschwindigkeitIst**“ in die Beobachtungstabelle reinziehen



* Über „Gerätekonfiguration“ 🡪 „Eigenschaften“ 🡪 „Webserver“ 🡪 „Allgemein“ den Webserver auf allen Modulen dieses Geräts aktivieren

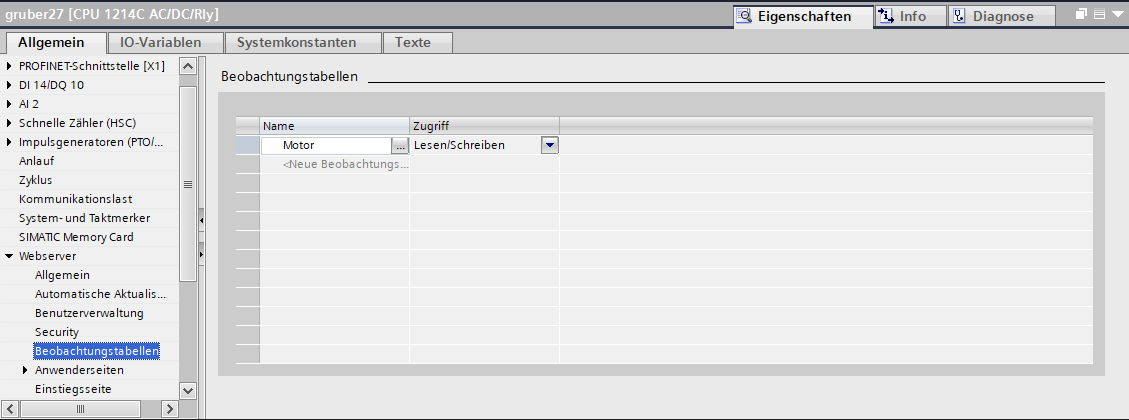


* In der Benutzerverwaltung einen neuen Benutzer mit administrativen Rechten erstellen

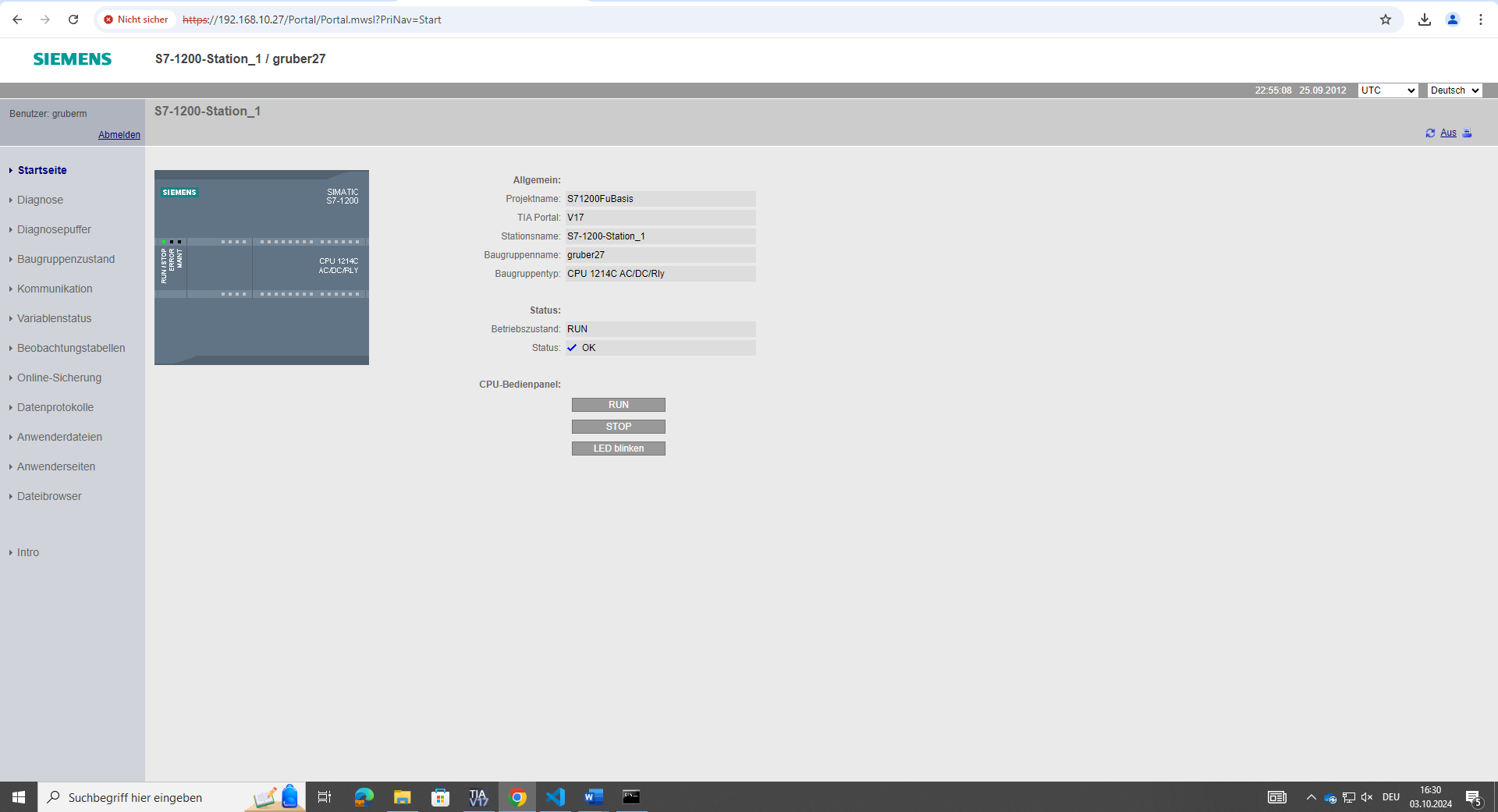


* Ein Bild, das Text, Software, Computersymbol, Webseite enthält.

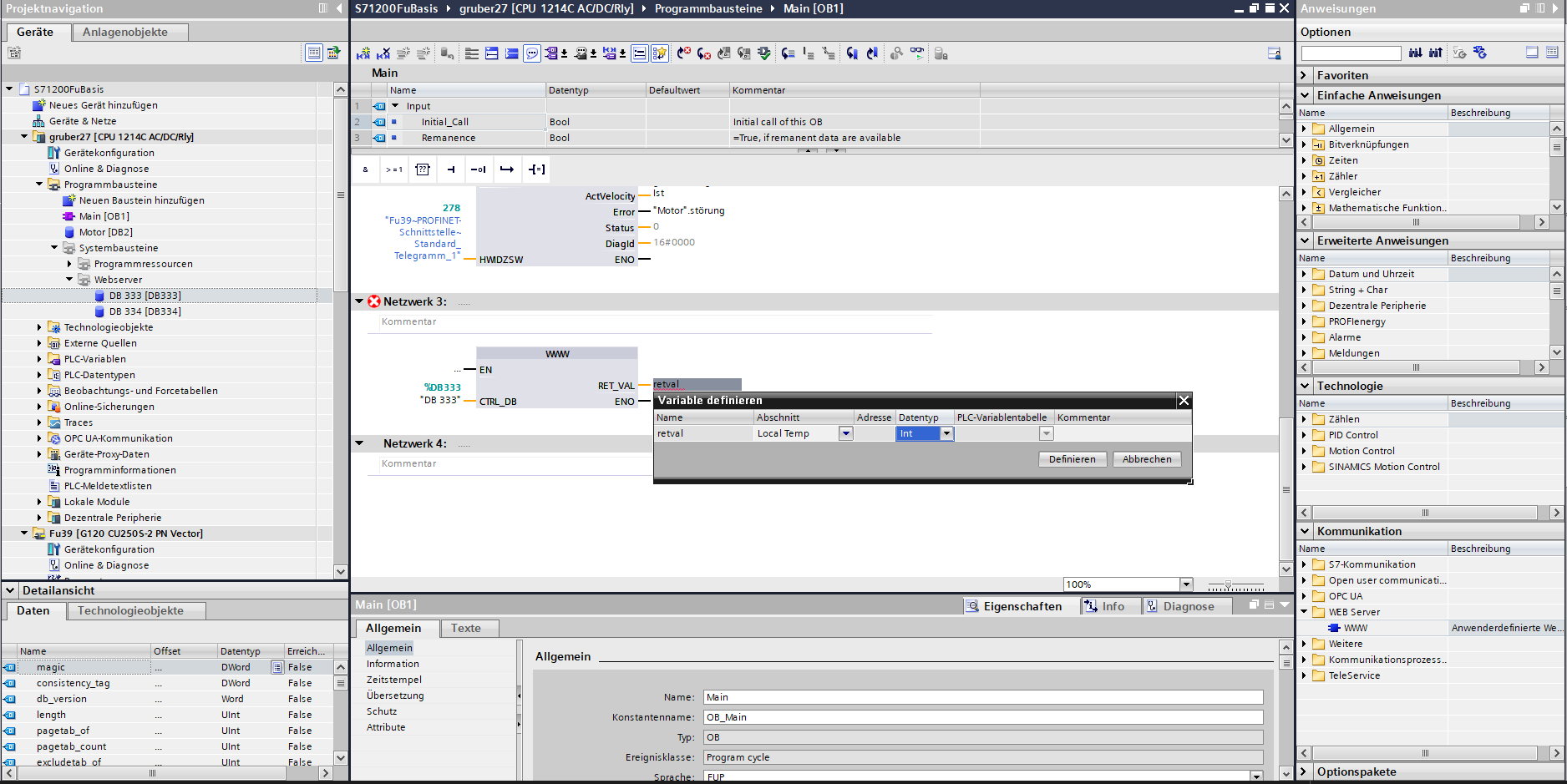
  Automatisch generierte BeschreibungBei „Security“ den Zertifikatstyp auf „Hardwaregeneriert“ umstellen
* Beobachtungstabelle „Motor“ unter „Beobachtungstabellen“ mit Lese/Schreib-Rechten hinzufügen



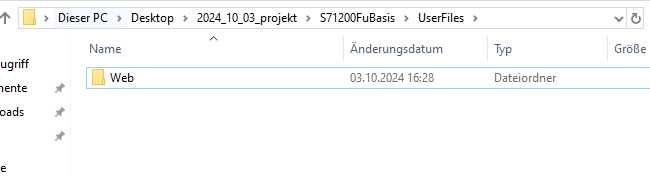
* Unter der IP-Adresse den Webserver aufrufen



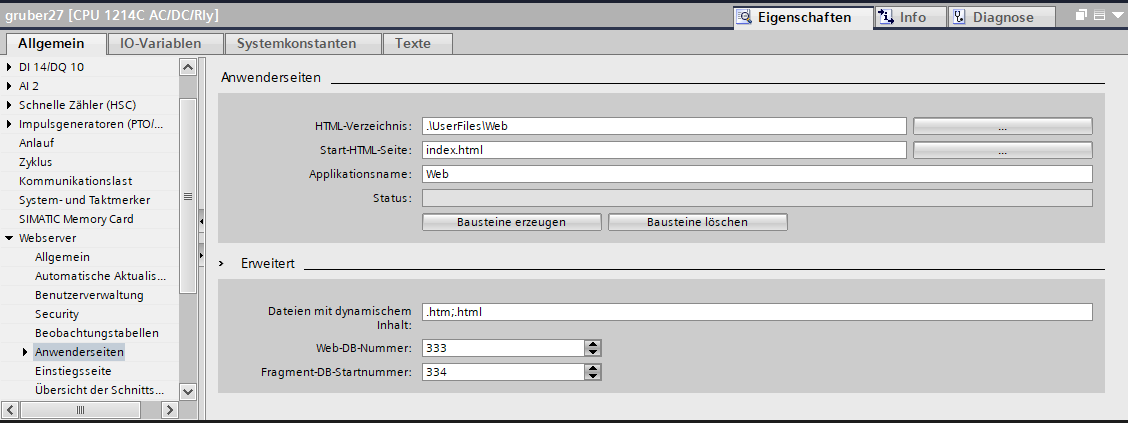
* Unter „Systembausteine/Webserver“ „DB333“ auswählen
* Rechts unter „Kommunikation/Webserver“ das Modul „WWW“ reinziehen
* RET\_VAL als retval setzen und die Variable mit Rechtsklick definieren



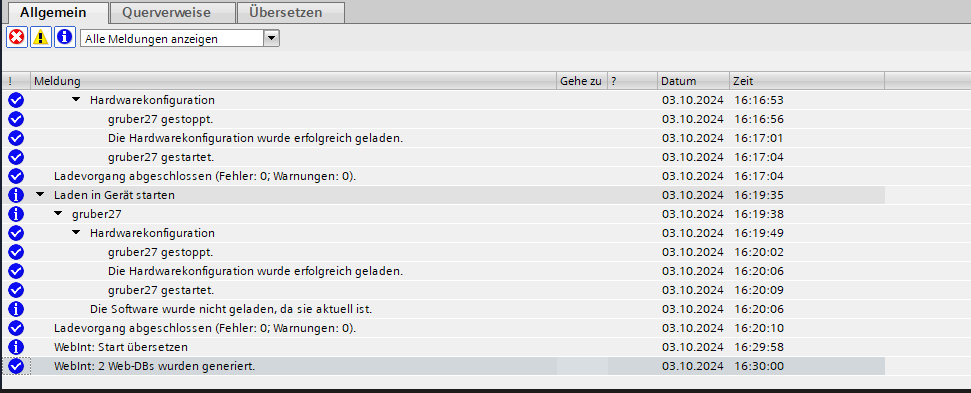
* Im Project ornder unter UserFiles den Ordner „Web“ erstellen und die Startseite die wir vom Fachlehrer bekommen haben reinkopieren



* Unter „Webserver/Anwenderseiten“ das HTML Verzeichnis und den Applikationsnamen definieren



* Folgende Ausgabe erscheint unter Allgemein wenn alles funktioniert hat



* Auf der vorher generierten Seite unter Anwenderseiten kann man nun auf die erstellte Website zugreifen und den Motor über ebendiese steuern.



# 

# 09.01.2025 - Protokoll

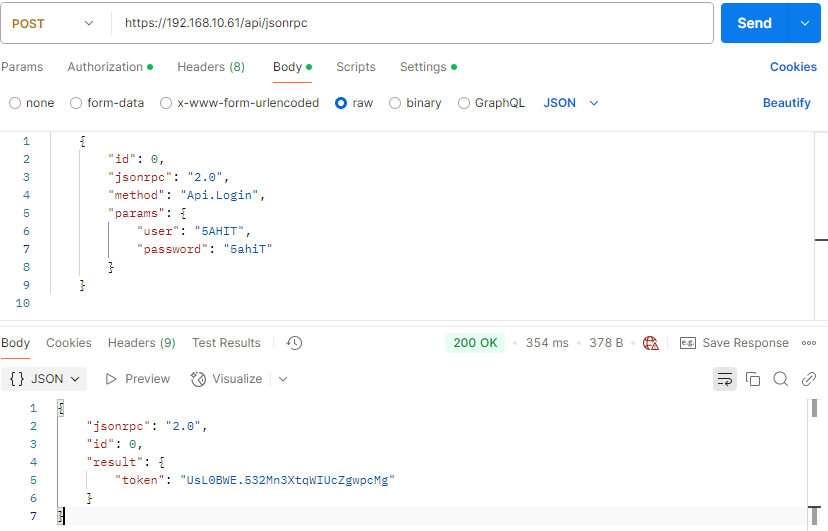
## Motoransteuerung über API-Requests

### Ziel

Das Ziel dieser Übung ist es, einen Motor über API-Requests zu steuern. Dies wird zunächst mit Postman getestet, um die grundlegenden API-Funktionen zu verstehen, und anschließend in einem Python-Skript umgesetzt.

### Login mit Postman

* In Postman wird ein POST-Request an die /jsonrpc-Route gesendet.
* Das JSON-Body enthält folgende Struktur:



* Als Antwort wird ein Token zurückgegeben, das für alle weiteren API-Requests erforderlich ist.

### API-Calls mit Python

Um die API einfacher zu verwenden, wird der Login in einer Funktion gekapselt.

### Login-Funktion in Python

Die folgende Funktion sendet den Login-Request und gibt den Token zurück:

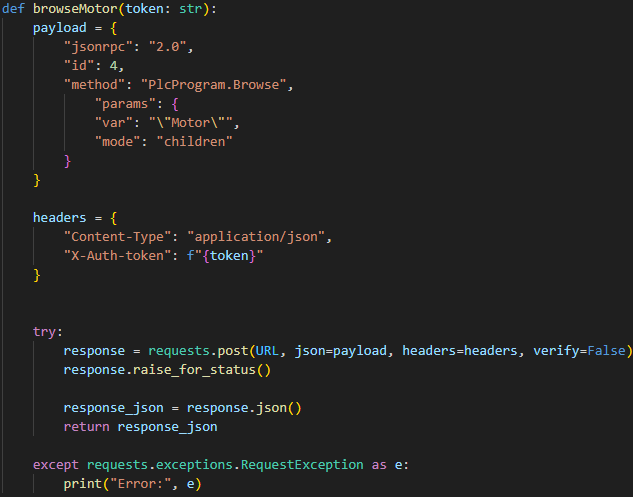


### Funktionen durch Browse entdecken

Um verfügbare Funktionen für den Motor herauszufinden, kann die Browse-Route verwendet werden.

Diese Route wird durch folgenden Code aufgerufen. In diesem Fall wird geschaut, welche Funktionen man für die Komponente „Motor“ ausführen kann.

Der vorher erhaltene Token muss angegeben werden.



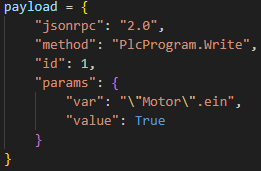
Folgende Ausgabe wird erzeugt, wenn man den Return-Wert dieser Funktion ausgibt.



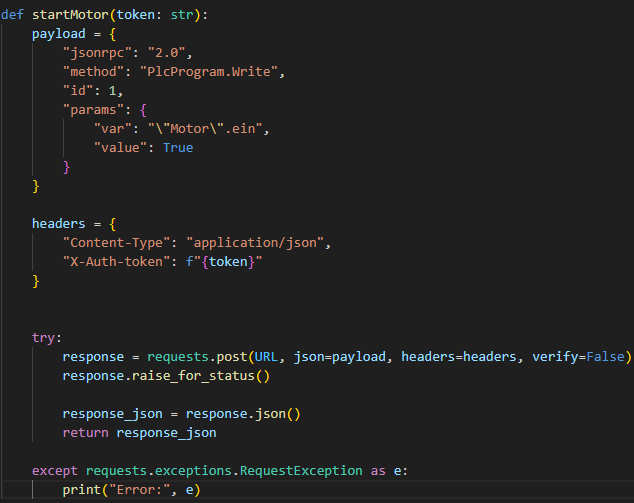
### Motor starten

Durch das vorherige Browsen wurde festgestellt, dass der Motor ein „.ein“ Attribut hat, welches den Motor startet.

Setzt man dieses Attribut hier auf True wird der Motor beim Aufrufen dieser Route gestartet.



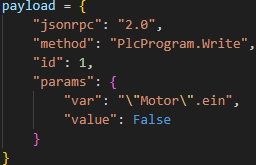
Folgendes Bild zeigt die ganze Funktion die den Motor startet.



### Motor stoppen

Auf ähnlichem Weg kann der Motor auch wieder gestoppt werden.

Das Motor.ein Attribut muss auf False gesetzt werden.



Der restliche Code ist gleich wie beim Starten des Motors.

### Erste Applikation

Durch folgenden Code wird der Motor für 5 Sekunden eingeschaltet und dann wieder aus.

