
Vorwort

Software für die Automatisierung bildet durch die Vielfalt der Schnittstellen ein komplexes Systemverhalten zwischen allen Komponenten und Geräten. Dies gilt ganz besonders für die industrielle Verwendung. Der Programmierer ist in dieser Kette der schöpferische Teil, – wie es in so vielen Softwareprodukten bereits praktiziert wurde.

In aller Munde sind die interessanten Themen *IoT* und *Industrie 4.0*, deren Anwendungen jedoch sehr vielschichtig sein können und über die Medien leider oft verschönt dargestellt werden. Hier ist zusätzliches Fachwissen erforderlich, welches für den Automatisierer nicht immer im Lernplan eingebunden ist. Zudem besteht je nach Anwendungsfall meist eine sehr starke Abhängigkeit von Drittanbietern, z. B. einem externen Server, auch als *Cloud* bezeichnet, wenn man nicht in der Lage ist, einen eigenen *Private-Server* zu besitzen.

Als Quereinsteiger zur neuen *Web-Technologie* – ich denke das sind die meisten Automatisierer – muss ein verständliches Vorgehen erarbeitet werden. Für dieses Buch und die Idee, mit dem Internet zu automatisieren, gehören die Werkzeuge *node.js* und die Programmiersprache *JavaScript*. Die Anwendungen erfolgen auf einem *Browser* (*Google Chrome*) und bilden zusammen eine gemeinsame Ebene für den *Server* und *Client*, denn beide werden mit der gleichen Programmiersprache *JavaScript* programmiert. Dieser Vorteil erlaubt es, uns auf das Wesentliche zu konzentrieren. Das sind *SVG*-Objekte und deren Anwendung, mit der Betrachtung damit Aufgaben aus der Automatisierung umzusetzen. Die so übliche und bekannte Webseite ist damit nicht gemeint, sondern eine browserorientierte *Web-Applikation*. *IoT* assoziiert auch u. A. Visualisierung (*HMI*), damit wir etwas beobachten können und *Industrie 4.0* eine Datenschnittstelle, damit wir etwas analysieren können. Die Datenschnittstelle des *Servers* verbindet unseren *Browser* mit den Geräten, welche Aktionen auslösen und Daten zur Visualisierung liefern. Das in diesem Buch angewendete Basis-Beispiel mit der *nodeMCU* im letzten Kapitel, soll das verdeutlichen und verständlich machen.

Deswegen habe ich Wert darauf gelegt, *JavaScript* und *node.js* einfach und schlicht jeweils gestaffelt in der wachsenden Anwendung vorzustellen. Es gibt also keine Hierarchie nach syntaktischen Sprachelementen, wie diese meistens in der Fachliteratur veröffentlicht werden. Hier soll der Leser in der Anwendung lernen, wie *JS* für die Automatisierung funktionieren kann. Das bringt mit sich, dass der Leser sich kapitelweise fortbildet, um weitere Dinge zum Sprachumfang *JS* zu finden, verdeutlicht aber „learning bei doing“. ☺

Spanien, Herbst 2019

Johannes Hofer