

Vorwort

An einem Spätsommertag des Jahres 2013 stand ich auf einer Bühne in einem Zelt in Bletchley Park, an dem Ort, an dem die britischen Codeknacker (unter ihnen der berühmte Alan Turing) im Zweiten Weltkrieg die Enigma-Verschlüsselung überlisteten. Es war einer der schönsten Tage in meinem Leben, da sich gerade zwei wunderbare Dinge ereignet hatten.

Erstens hatte ich es irgendwie geschafft, einen Hacking-Wettbewerb zu gewinnen (weshalb ich auf der Bühne stand). Das National Museum of Computing, das sich ebenfalls in Bletchley Park befand, benötigte technische Hilfe, um webgestützte, interaktive Ausstellungsobjekte zur Darstellung der Chronologie zu kreieren. Ich arbeitete die Nacht durch, um eine quelloffene JavaScript-Bibliothek zu erweitern und einen Prototyp zu entwickeln. Dies war mein Beitrag, der zu meiner großen Freude zum besten in dem Wettbewerb gekürt wurde. Die zweite großartige Sache bestand darin, dass ich einen Preis gewonnen hatte – und nicht einfach irgendeinen Preis!

Meine Belohnung war ein originales Arduino-Uno-Starterkit, das aus einer Arduino-Platine, einer Zusammenstellung von elektronischen Bauteilen und einem Buch als Anleitung bestand. Das hat mein Leben entscheidend verändert. Ich stellte fest, dass die Kombination meiner neuen Elektronikkenntnisse mit meiner bisherigen täglichen Beschäftigung – der Programmierung von quelloffenen, auf Standards gestützten Websites und Anwendungen –, eine der faszinierendsten Mischungen ergab, die ich je kennengelernt hatte: JavaScript für physische Objekte. Ich konnte die Sprache JavaScript, die ich bereits beherrschte, als Turbolader für elektronische Basteleien und das Internet der Dinge (Internet of Things, IoT) einsetzen.

Bis es so weit war, dauerte es jedoch eine Weile. Zunächst einmal musste ich lernen, einfache elektronische Schaltungen zu bauen, indem ich die Beispiele aus dem Begleitbuch durcharbeitete und später das Web durchforstete, um noch mehr Kenntnisse zu gewinnen. Ich erfuhr, wie ich die Schaltungen logisch steuern konnte, indem ich den Mikrocontroller des Arduino Uno programmierte und einfache Sketche (Programme) in der Arduino-Sprache schrieb, die C/C++ (sehr) ähnelt, aber für den (sehr) beschränkten Programm- und Arbeitsspeicher der Platine optimiert ist.

Ende 2013 entdeckte ich schließlich Johnny-Five. Dieses quelloffene Node.js-Framework war damals noch jung, aber bereits sehr leistungsfähig. Anstatt maschinennahen, eingeschränkten Arduino-Code zu schreiben, konnte ich meinen

Uno auch mit JavaScript-Programmen höherer Ebene steuern. »Wow«, dachte ich, »ich wünschte, das hätte ich eher gewusst!«

Die Kombination von JavaScript und Mikrocontrollern ist nicht nur ein Zaubertrick, der dazu dient, die ganze bekannte Welt mit JavaScript zu überziehen. Zu Anfang war selbst ich als Node.js-Anhängerin skeptisch: Vielleicht war es sinnlos; vielleicht würde es niemals Fuß fassen.

Keine Sorge: Es war nicht sinnlos und es fasste Fuß.

Kurioserweise machte die Hinzunahme von JavaScript die Sache für mich *einfacher*, sodass ich Prototypen meiner kleinen Projekte viel schneller erstellen konnte. Ich war in der Lage, Arbeitsabläufe für die Entwicklung zu nutzen, die mir als Webentwicklerin viel besser vertraut waren. Ich musste mich nicht mehr so sehr um maschinennahe Speicherverwaltung und Ressourcenoptimierung kümmern. Die Kapselung des Verhaltens in *Komponentenklassen* höherer Ebenen, die Johnny-Five bietet, ist intuitiv zugänglich und der resultierende Code kann viel sauberer und leichter handhabbar sein als viele Arduino-Bibliotheken. Außerdem konnte ich damit über npm das nahezu unerschöpfliche Angebot der weltweiten Node.js-Community nutzen und wie in jedem anderen Node.js Module importieren. Das war wundervoll.

Um es ganz klar zu sagen: Es ist nichts daran verkehrt, mit der Arduino-Sprache zu arbeiten und Mikrocontroller auf die herkömmlichere, auf C gestützte Weise zu programmieren. Beispielsweise kann es gute Gründe dafür geben, sich selbst um die Speicherverwaltung zu kümmern, insbesondere beim Schreiben von Firmware oder beim Bauen von Geräten für die Produktion. Das Arduino-System ist ein kleines Wunder: Sein ganzer Daseinszweck besteht darin, Einsteigern eingebettete Elektronik nahezubringen. Bei null mit dem Arduino und der Arduino-Programmiersprache anzufangen, ist eine sinnvolle und zu bewältigende Vorgehensweise.

Allerdings kann JavaScript Webentwicklern helfen, schneller den Einstieg in die Welt der Elektronik zu finden. Erstens werden bei einem Großteil des Lernmaterials für den Arduino (und andere Plattformen) keinerlei Programmierkenntnisse vorausgesetzt, weshalb Sie sich oft durch lange Erklärungen quälen müssen, was ein Array ist oder wie Schleifen funktionieren. Die Einschränkungen und Besonderheiten von Mikrocontrollern können verwirrend wirken, wenn Sie einfach nur herausfinden wollen, wie etwas funktioniert. Auch die IDEs sind manchmal unhandlich. Manchmal verwenden Sie mehr Zeit auf die Einrichtung als darauf, tatsächlich produktiv tätig zu werden. JavaScript kapselt eine ganze Menge Einzelheiten, sodass Sie sich auf die wichtigen neuen Dinge konzentrieren können, die Sie lernen möchten.

Dieser Gedanke war der Ausgangspunkt für dieses Buch: die Vorstellung, dass JavaScript als Zugang zur Elektronik dienen und es leichter machen kann, zu lernen, wie man coole Sachen mit einem Minimum an Kopfzerbrechen baut. JavaScript ist die beliebteste Programmiersprache der Welt und die De-facto-Sprache des Web, und das Internet der Dinge und die Kultur der Elektronikbastler sind

sowohl kreativ als auch kommerziell anregend. Warum die beiden also nicht vereinigen?

Vergessen Sie auch nicht, dass *die Sache Spaß macht!* Es ist ein tolles Gefühl, sich eine eigene Erfindung auszudenken und dann auch tatsächlich zu realisieren. Grundkenntnisse in Niederspannungsschaltungen und der Funktionsweise von eingebetteten Systemen in der Praxis heben das Selbstvertrauen.

Vielleicht gefällt es Ihnen ebenso wie mir. Vielleicht werden Sie sich auch an Open-Source-Projekten beteiligen. Vielleicht bauen Sie einen genialen Apparat. Vielleicht bringen Sie anderen bei, was Sie gelernt haben.

Vielleicht haben Sie auch einfach nur Spaß. Das ist schon mehr als genug.

Danksagung

Mit meinem Dank möchte ich dort beginnen, wo meine Liebe zur Hardwarebasterei begann: bei den Mitarbeitern der Tagung *Over the Air* und dem National Museum of Computing, von dem ich mein erstes Arduino-Starterkit bekommen habe. Ohne dieses Ereignis wäre alles ganz anders gekommen. Dan Appelquist, Margaret Gold und Matthew Cashmore: Vielen Dank, dass ihr eine solche hervorragende Veranstaltung ausgerichtet und mich dazu eingeladen habt, und das auch noch mehr als einmal!

Rick Waldon schafft an einem Tag mehr als ich in einem ganzen Monat. (Rick, dein Händchen für JavaScript ist eine Legende!) Seine Beteiligung an TC-39, der ECMA-Arbeitsgruppe, die für die Sprache JavaScript selbst verantwortlich ist, macht ihn unverzichtbar. Darüber hinaus hat er auch noch Johnny-Five erfunden, das führende quelloffene Node.js-Framework für Robotik und das IoT, das im Mittelpunkt dieses Buchs steht. Ich könnte noch seiten-, ja kapitelweise weiter über Rick schwadronieren.

Ein Buch zu schreiben, erfordert unglaublich viel Zeit. Daher möchte ich meinen Vorgesetzten und Kollegen bei Bocoup vielen Dank dafür sagen, dass sie mir die Zeit und Hilfe gegeben haben, die ich brauchte, und für den ständigen Zustrom an Ermutigung, den sie in meine Richtung gepumpt haben. Vielen Dank auch an meine Partner und Mitarbeiter bei Cloud Four für ihre Geduld.

In einer Welt, in der der Wert von Lektorierung und Rückmeldung oft geringgeschätzt wird, sind großartige Lektoren ein wahres Geschenk. Meine Lektorin Susanna Kline hat mir auf dem ganzen langen Weg hilfreiche Unterstützung und Einsicht geboten. Brad Luyster, deine fachliche Rückmeldung war phänomenal; es ist schwer, dir angemessen zu danken. Auch verschiedene andere Personen haben mir in den verschiedenen Stadien des Manuskripts wertvolle Rückmeldung gegeben: Alessandro Campeis, Amit Lamba, Andrew Meredith, Bruno Sonnino, Earl Bingham und Kevin Liao. Ich möchte meinem Verleger Marjan Bace von Manning und dem Rest des Redaktions- und Herstellungsteams danken, die hinter den Kulissen an diesem Buch gearbeitet haben.

Danke auch an Francis Gulotta für technische Hinweise, an Kyle Jackson von Manning, der sich um meinen Bedarf an technischer Unterstützung kümmerte, und an meinen Kumpel Chau Doan, der mir seine Erkenntnisse über Firmware und eingebettete Elektronik mitteilte.

Die Communitys rund um Johnny-Five und JavaScript-on-Things waren einfach wunderbar! Danke, Derek Runberg von SparkFun! Danke, Donova Buck, David Resseguie, Brian Genisio und alle anderen Johnny-Five-Mitarbeiter!

Shawn Hymels praktische Arduino-Bibliothek für den Sensor APDS-9960 war der Ausgangspunkt für Teile von Kapitel 9 und die Roverbeispiele in den Kapiteln 6 und 8 basieren auf Code von Rick Waldron (noch einmal danke, Rick!).

Ebenso wichtig wie diejenigen, die einen bei dem Projekt selbst unterstützen, sind auch diejenigen, die einem helfen, geistig gesund zu bleiben, während sie es mit einem aushalten: Danke euch, meine wunderbaren Verwandten und Freunde!

Den allerbesten Platz habe ich reserviert, um meinem Partner (und fantastischen Menschen) Bryan Fox zu danken: Ohne seine fröhliche und beständige Gegenwart wäre dieses Buch nicht möglich gewesen.