

Making Things Talk

MAKE: PROJECTS

Vorwort

Vor einigen Jahren schrieb Neil Gershenfeld ein schlaues Buch namens *When Things Start to Think* (Wenn Dinge zu denken beginnen). Darin ging es um eine Welt, in der alltägliche Dinge und Geräte über Rechenleistung wie die eines Computers verfügen, anders gesagt: die Welt von heute. Er befasste sich mit den Auswirkungen von Geräten, die Informationen über unsere Identitäten, Fähigkeiten und unser Handeln miteinander austauschen. Es ist ein gutes Buch, aber ich finde, der Titel passt nicht. Ich hätte es *When Things Start to Gossip* (Wenn Dinge zu tratschen beginnen) genannt, denn – seien wir ehrlich – sogar die aufregendsten Gedanken sind nur dann etwas wert, wenn man mit jemand anderem darüber spricht. *Making Things Talk* wird dir zeigen, wie du selbst Dinge baust, die dank ihrer Rechenleistung miteinander sprechen können, und wie man andere Menschen in die Lage versetzt, mithilfe dieser Geräte zu kommunizieren.

Seit einigen Jahrzehnten benutzen Informatiker den Begriff **objektorientierte Programmierung** für eine Art der Softwareentwicklung, in der Programme und Unterprogramme als Objekte angesehen werden. Genau wie physische Objekte haben sie Eigenschaften und Verhaltensweisen. Sie erben diese Eigenschaften von den **Prototypen**, von denen sie abstammen. Die Normalform jedes Objekts in Software ist der Code, der seine Art beschreibt. Software-Objekte machen es einfach, Objekte auf neuartige Weise zu rekombinieren. Man kann ein Software-Objekt wiederverwenden, wenn man seine **Schnittstelle** kennt, also die Sammlung von Eigenschaften und Methoden, auf die der Autor des Objekts den Zugriff erlaubt. (Man benötigt natürlich auch eine Anleitung, damit man weiß, wie es geht.) Es ist dabei egal, wie ein Software-Objekt macht, was es tut, solange es das konsistent tut. Software-Objekte sind am effektivsten, wenn sie einfach zu verstehen sind und wenn sie gut mit anderen Objekten zusammenarbeiten.

In der physischen Welt sind wir von allen möglichen elektronischen Objekten umgeben: Radiowecker, Toaster, Handys, MP3-Player, Kinderspielzeug usw. Es kann sehr viel Arbeit sein und eine erhebliche Menge an Wissen erfordern, ein nützliches elektronisches Gadget zu bauen. Es kann fast genauso viel Wissen erfordern, diese Gadgets miteinander auf nützliche Weise reden zu lassen. Aber das muss nicht der Fall sein. Elektronische Geräte können aus einfachen Modulen aufgebaut sein, und oft sind sie das auch. Solange man die Schnittstellen versteht, kann man daraus alles bauen. Stell dir das wie **objektorientierte Hardware** vor. Das Verständnis der Art und Weise, wie Dinge miteinander reden, ist für den Arbeitserfolg von zentraler Bedeutung, egal ob das Objekt ein Toaster, ein E-Mail-Programm auf deinem Laptop oder eine vernetzte Datenbank ist. Alle diese Objekte können verbunden werden, wenn man herausfinden kann, wie sie kommunizieren. **Dieses Buch** ist eine Anleitung für einige Werkzeuge, mit denen sich diese Verbindungen herstellen lassen.

x

“ Für wen dieses Buch gedacht ist

Dieses Buch wurde für Menschen geschrieben, die Dinge mit anderen Dingen sprechen lassen wollen. Vielleicht bist du eine Physiklehrerin, die ihren Schülern zeigen will, wie man das Wetter an verschiedenen Orten in der Stadt gleichzeitig messen kann, oder ein Bildhauer, der einen Raum voller choreografiertes mechanischer Skulpturen bauen will. Du könntest auch ein Industriedesigner sein, der schnell Prototypen neuer Produkte aufbauen muss, die sowohl Form als auch Funktion nachbilden. Vielleicht hast du auch eine Katze und möchtest auch mit ihr spielen können, wenn du gar nicht zu Hause bist. Dieses Buch ist die Einstiegslektüre für alle mit wenig technischer Ausbildung und großem Interesse. Dieses Buch ist für Menschen gedacht, die Projekte auch fertigstellen wollen.

Die wichtigsten Instrumente in diesem Buch sind PCs, Webserver und Mikrocontroller, die Minicomputer in Alltagsgeräten. In den letzten zehn Jahren haben sich Mikrocontroller und ihre Programmierertools von geheimnisvollen Objekten zu gewöhnlichen, einfach zu bedienenden Elementen entwickelt. Grundschüler benutzen heute Tools, vor denen Studenten vor einem Jahrzehnt ratlos herumsaßen. Während dieser Zeit haben meine Kollegen und ich Menschen aus den unterschiedlichsten Bereichen (darunter nur wenige gelernte Programmierer)

beigebracht, wie man diese Tools verwenden kann, um den Bereich der physischen Aktivität zu vergrößern, in dem Computer reagieren, wahrnehmen und interpretieren können.

In den letzten Jahren haben die Nutzer von Mikrocontrollern ein wachsendes Interesse daran, ihre Geräte nicht nur die Umwelt wahrnehmen und steuern zu lassen, sondern auch mit anderen Dingen darüber zu kommunizieren, was sie wahrnehmen und steuern. Wenn du schon einmal

etwas mit einer Basic Stamp oder einem Lego-Mindstorms-Set gebaut hast und dieses Etwas mit anderen Dingen, die du oder andere gebaut haben, kommunizieren lassen willst ist dieses Buch das Richtige für dich. Es ist auch für Softwareprogrammierer nützlich, die sich mit Netzwerken und Webservices auskennen und eine Einführung in Embedded-Netzwerkprogrammierung suchen.

Wenn du der Typ Mensch bist, der gerne bis in den inneren Kern einer Technologie vordringt, findest du vielleicht in

diesem Buch nicht, was du suchst. Es gibt keine detaillierten Codeschnipsel für Bluetooth- oder TCP/IP-Stacks, und auch keine Schaltpläne für Ethernet-Controllerchips. Mit den verwendeten Bauteilen wird versucht, ein Gleichgewicht zwischen Einfachheit, Flexibilität und Kosten herzustellen. Sie bestehen aus objektorientierter Hardware, die mit relativ wenig Verdrahtung und Code auskommt. Sie sind dafür entworfen, dich so schnell wie möglich ans Ziel zu bringen und Dinge miteinander reden zu lassen.

X

“ Was du wissen musst

Um den größten Nutzen aus diesem Buch zu ziehen, solltest du über Grundkenntnisse der Elektronik und Programmierung von Mikrocontrollern verfügen, dich etwas mit dem Internet auskennen und zu beidem Zugang haben.

Viele Menschen, deren Programmiererfahrung mit Mikrocontrollern beginnt, können mit einigen Sensoren und ein paar Servomotoren wunderbare Dinge anstellen. Sie haben aber oft kaum etwas getan, um die Kommunikation zwischen dem Mikrocontroller und anderen Programmen auf einem PC zu ermöglichen. Ebenso haben viele erfahrene Netzwerk- und Multimedia-Programmierer nie mit Hardware irgendwelcher Art, einschließlich Mikrocontrollern, experimentiert. Wenn du eine dieser Personen bist, ist dieses Buch etwas für dich. Weil die Zielgruppe dieses Buchs so vielfältig ist, findest du Teile der Einleitungen vielleicht etwas zu einfach, je nach deinem fachlichem Hintergrund. Wenn ja, zögere nicht, das, was du schon weißt, zu überblättern und da weiterzulesen, wo es interessant wird.

Wenn du noch nie einen Mikrocontroller benutzt hast, brauchst du ein wenig Hintergrundwissen, bevor du mit diesem Buch anfängst. Ich empfehle mein voriges Buch **Physical Computing: Sensing and Controlling the Physical World with Computers** (Thomson), von mir und Dan O'Sullivan geschrieben, das die Grundlagen

von Elektronik, Mikrocontrollern und „Physical Interaction Design“ beschreibt.

Du solltest ebenfalls ein grundlegendes Verständnis von Computerprogrammierung haben, bevor du weiterliest. Wenn du noch nie programmiert hast, sieh dir die Programmierumgebung Processing auf www.processing.org an. Processing ist eine einfache Sprache, die entwickelt wurde, um Nichtprogrammierern das Programmieren beizubringen. Sie ist aber mächtig genug, um viele fortgeschrittene Aufgaben zu lösen. Processing wird in diesem Buch durchgängig verwendet, um grafische Interfaces zu programmieren.

Dieses Buch enthält Codebeispiele in ein paar verschiedenen Programmiersprachen. Es sind alles ziemlich einfache Beispiele. Wenn du also nicht mit der dargestellten Sprache arbeiten willst, kannst du die Beispiele anhand der Kommentare in ihnen in deiner bevorzugten Sprache nachprogrammieren.

X

“ Inhalt dieses Buchs

Dieses Buch erklärt die Konzepte, die vernetzten Objekten zugrunde liegen, und stellt dann Rezepte vor, die jeden Satz von Konzepten veranschaulichen. Jedes Kapitel enthält Bauanleitungen für funktionierende Projekte, die auf den neuen Ideen aus diesem Kapitel basieren.

In Kapitel 1 werden dir die wichtigsten Programmier-Tools des Buchs begegnen und mit jedem wird es ein „Hallo Welt!“ geben.

Kapitel 2 stellt die grundlegenden Konzepte vor, die benötigt werden, um Dinge miteinander sprechen zu lassen. Es deckt die Eigenschaften ab, die im Vorfeld vereinbart werden müssen, und zeigt, wie es bei der Fehlersuche hilft, diese Dinge gedanklich voneinander zu trennen. Du wirst als Beispiel für eine Modemverbindung ein einfaches Projekt bauen, das eine serielle Eins-zu-eins-Kommunikation zwischen einem Mikrocontroller und einem PC über Bluetooth-Funk enthält. Du wirst etwas über Datenprotokolle, Modems und Adress-Schemata lernen.

Kapitel 3 stellt ein komplexeres Netzwerk vor: Das Internet. Das Kapitel behandelt die grundlegenden Geräte, die es zusammenhalten, und die grundlegenden Beziehungen zwischen den Geräten. Du wirst die Meldungen sehen, die den häufigsten Vorgängen zugrunde liegen, die du jeden Tag im Internet machst, und du wirst lernen, wie man diese Meldungen senden kann. Du wirst deinen ersten Satz von Programmen schreiben, um Daten über das Internet zu schicken, die auf einem physischen Vorgang bei dir zu Hause basieren.

In Kapitel 4 wirst du dein erstes Embedded-Gerät bauen. Du wirst sicherer im Umgang mit Verbindungen ins Internet auf der Kommandozeile und du wirst einen Mikrocontroller mit einem Webserver verbinden, ohne dass einen PC oder Laptop dazwischengeschaltet ist.

Kapitel 5 bringt die Internetverbindung noch einen Schritt weiter und erklärt Socket-Verbindungen, die längere Interaktionen ermöglichen. Du wirst lernen, dein eigenes Serverprogramm zu schreiben, mit dem du alles verbinden kannst, das mit dem Internet verbunden ist. Du wirst die Verbindung zu diesem Serverprogramm von der Kommandozeile und über einen Mikrocontroller aufbauen, um zu

verstehen, wie man verschiedene Arten von Geräten über denselben Server verbinden kann.

Kapitel 6 führt in die drahtlose Kommunikation ein. Du wirst einige der Eigenschaften von Funkkommunikation kennenlernen, sowie deren Möglichkeiten und Grenzen. Mehrere kurze Beispiele in diesem Kapitel ermöglichen es dir, auf verschiedene Weise „Hallo Welt!“ per Funk zu übertragen.

Kapitel 7 bietet eine Alternative zu den Socket-Verbindungen aus Kapitel 5 und führt nachrichtenbasierte Protokolle wie UDP im Internet sowie ZigBee und 802.15.4 für Drahtlosnetzwerke ein. Anstelle das Client-Server-Modell aus früheren Kapiteln zu verwenden, erfährst du hier, wie du Nachrichtenübertragungen entwirfst, bei denen jedes Objekt im Netz gleichberechtigt ist und zur Kommunikation eine Nachricht nach der anderen austauscht.

Kapitel 8 dreht sich um Positionsbestimmung. Darin werden einige Tools vorgestellt, die dabei helfen, Dinge räumlich zu verorten. Es bietet auch einige Gedanken über die Beziehungen zwischen räumlichem Standort und Netzwerkbeziehungen.

Kapitel 9 handelt von der Identifikation im physischen Raum und im Netzwerkraum. Du wirst einige Methoden kennenlernen, um einzigartige Netzwerkidentitäten zu erzeugen, die auf physischen Eigenschaften basieren. Außerdem erfährst du etwas darüber, wie du Eigenschaften eines vernetzten Geräts bestimmst.

Kapitel 10 führt Mobilfunknetze ein und beschreibt viele der Dinge, die man inzwischen mit Handys und Handynetzen machen kann.

Kapitel 11 gibt einen Rückblick über die verschiedenen Arten von Protokollen, die das Buch behandelt, und bringt diese in einen Zusammenhang, so dass du diese in Zukunft dort nachschlagen kannst.

x

“ Bauteile kaufen

Für alle Projekte in diesem Buch brauchst du eine Menge an Teilen. Dabei wirst du von vielen Anbietern erfahren. Weil es in meiner Stadt keinen großen Handel für Elektronikteile gibt, bestelle ich immer im Internet. Wenn du das Glück hast, in einer Stadt zu leben, in der es ein Ladengeschäft dafür gibt: Glück gehabt! Falls nicht, mach dich mit einigen der Onlinehändler vertraut.

Eine vollständige Originalliste der Bezugsquellen ist im Anhang enthalten. Nach eigenem Ermessen kannst du natürlich Bauteile, mit denen du dich auskennst, durch andere ersetzen.

Weil es sehr einfach ist, Waren online zu bestellen, bist du vielleicht versucht, mit den Händlern nur über ihre Website zu kommunizieren. Du solltest aber nicht zögern, auch das Telefon zu benutzen. Insbesondere wenn du das erste Mal ein bestimmtes Projekt angehst, hilft es, mit jemandem über die bestellten Teile zu reden und Fragen zu stellen. Es ist bei den hier aufgelisteten Händlern wahrscheinlich,

hilfsbereite Menschen am anderen Ende der Leitung vorzufinden. Ich habe die Telefonnummern angegeben, wo immer möglich. Mach davon Gebrauch.

Wir empfehlen dir, die genauen Produktbezeichnungen bei den amerikanischen Händlern nachzuschlagen und dann zu recherchieren, welcher deutsche Händler diese Produkte vorrätig hat. Eine Liste der deutschsprachigen Elektronik-Versender und eine Liste der amerikanischen Händler, sortiert nach den Buchprojekten, findest du im Anhang.

X

“ Nutzung von Codebeispielen

Dieses Buch soll dir helfen, deine Idee zu verwirklichen. Im Allgemeinen darfst du den Code in diesem Buch in deinen Programmen und Dokumentationen benutzen. Du musst nicht unsere Erlaubnis einholen, sofern du nicht einen größeren Anteil des Codes reproduzierst.

Beispielsweise bedarf es keiner Genehmigung, ein Programm zu schreiben, das mehrere Codeabschnitte aus diesem Buch verwendet. Der Verkauf oder Vertrieb einer CD-ROM mit Beispielen aus O'Reilly-Büchern erfordert hingegen eine Genehmigung. Eine Frage zu beantworten, indem man dieses Buch zitiert und Beispielcode daraus verwendet, erfordert keine Genehmigung. Eine größere Menge von Beispielcode aus diesem Buch in der Dokumentation deines Produkts zu verwenden, erfordert allerdings eine Genehmigung.

Wir freuen uns über jeden Quellennachweis. Ein Nachweis enthält normalerweise Titel, Autor, Verlag und ISBN, zum Beispiel so: „*Making Things Talk: Practical Methods for Connecting Physical Objects*, von Tom Igoe. Copyright 2011 O'Reilly Media, 978-1-4493-9243-7.“ Wenn du glaubst, dass die von dir beabsichtigte Nutzung der Codebeispiele nicht unter die oben angegebenen Erlaubnisse fällt, dann schreib uns gerne an permissions@oreilly.com.

X

““ Nutzung der Schaltungsbeispiele

Beim Bau der Projekte aus diesem Buch wirst du Dinge kaputtmachen und Garantiebestimmungen verletzen. Falls du dem ablehnend gegenüberstehst, solltest du dieses Buch jetzt schließen und nicht wieder öffnen. Dies ist kein Buch für diejenigen, die zögern, Dinge auseinanderzunehmen, wenn unsicher ist, ob man sie wieder zusammensetzen kann.

Auch wenn wir dich dazu anstiften wollen, abenteuerlustig zu sein, möchten wir auch, dass deine Sicherheit gewährleistet ist. Bitte geh keine unnötigen Risiken beim Bau der Projekte aus diesem Buch ein. Jede Anleitung wurde mit deiner Sicherheit im Hinterkopf geschrieben. Ignorieren der Sicherheitshinweise geschieht auf eigene Gefahr. Du solltest dir sicher sein, dass du das ausreichende Maß an Wissen und Erfahrung hast, um die Arbeit in einer sicheren Weise durchzuführen.

Bitte beachte, dass die Projekte und Schaltungen in diesem Buch nur als Handlungsanleitungen gedacht sind. Details wie Stromversorgung, automatische Resets, HF-Abschirmung und andere Dinge, die für die Zertifizierung eines für den Handel bestimmten elektronischen Geräts nötig sind, sind hier nicht enthalten. Wenn du echte Produkte entwirfst, die von anderen Menschen als dir selbst benutzt werden sollen, verlass dich bitte nicht nur auf die Informationen aus diesem Buch.

x

““ Danksagungen für die Erstauflage

Dieses Buch ist das Ergebnis vieler Gespräche und Kooperationen. Es wäre ohne die Unterstützung und Ermutigung meines eigenen sozialen Netzes nicht möglich gewesen.

Das Interactive Telecommunications Program in der Tisch School of the Arts der New York University ist seit mehr als einem Jahrzehnt meine Heimat. Dieser lebendige und angenehme Arbeitsplatz ist voller tatentierter Menschen. Dieses Buch entstand dem Seminar „Vernetzte Objekte“, das ich dort seit mehreren Jahren gebe. Ich hoffe, dass die Ideen in diesem Buch den Geist dieses Ortes wiedergeben und dir ein Gefühl meiner Freude vermitteln, dort zu arbeiten.

Red Burns, die Gründerin des Fachbereichs, hat mich unterstützt, seitdem ich auf diesem Gebiet angefangen habe. Sie hat meine viele Höhenflüge mit Nachsicht bedacht und hat mich immer wieder auf die Erde zurückgeholt, wenn es nötig war. Bei jedem Projekt fordert sie mich heraus, Technik nicht um ihrer selbst willen einzusetzen, sondern immer so, dass sie den Menschen befähigt.

Dan O'Sullivan, mein Kollege und mittlerweile Leiter des Studiengangs, brachte mich mit Physical Computing in Kontakt und gestattete mir dann großzügigerweise, in der Lehre und in der Ausformung des Schwerpunkts des ITP mitzuarbeiten. Er ist ein großartiger Berater und Mitspieler und hat mir bei der Arbeit durchgängig sein Feedback angeboten. Die meisten Kapitel entstanden aus Plaudereien mit Dan. Seine Fingerabdrücke finden sich überall in diesem Buch, und es ist dadurch ein besseres Buch geworden.

Clay Shirky, Daniel Rozin und Dan Shiffman waren ebenfalls enge Berater bei diesem Projekt. Clay schaute geduldig darüber hinweg, wie der Berg an Bauteilen in unserem Büro wuchs. Er war so nett, sein eigenes Schreiben zu unterbrechen, um seine Meinung zu meinen Ideen zu äußern. Daniel Rozin bot ebenfalls wertvolle kritische Anmerkungen. Seine Ideen haben das Buch stark beeinflusst. Dan Shiffman hat viele Konzepte gelesen und hilfreiche Rückmeldungen gegeben. Er hat außerdem viele großartige Codebeispiele und -bibliotheken beigetragen.

Die Kolleginnen und Kollegen am Fachbereich, Marianne Petit, Nancy Hechinger und Jean-Marc Gauthier, haben mich in der Zeit des Schreibens unterstützt, ermutigt und inspiriert, haben mir Verwaltungsarbeiten abgenommen und mich durch ihre eigene Arbeit inspiriert.

Die übrigen Dozenten und Mitarbeiter des ITP haben ebenfalls beigetragen. George Agudow, Edward Gordon, Midori Yasuda, Megan Demarest, Nancy Lewis, Robert Ryan, John Duane, Marlon Evans, Tony Tseng und Gloria Sed haben jede Art von Wahnsinn im Namen des Physical Computing und vernetzter Objekte toleriert und mir und den anderen Dozenten und auch den Studenten die Beschäftigung damit ermöglicht. Die wissenschaftlichen Mitarbeiter Carlyn Maw, Todd Holoubek, John Schimmel, Doria Fan, David Nolen, Peter Kerlin und Michael Olson haben den Dozenten und Studenten der vergangenen Jahre dabei geholfen, Projekte durchzuführen, die die Projekte beeinflusst haben, die in diesem Buch behandelt werden. Die Dozenten Patrick Dwyer, Michael Schneider, Greg Shakar, Scott Fitzgerald, Jamie Allen, Shawn Van Every, James Tu und Raffi Krikorian haben die Tools aus diesem Buch in ihren Seminaren eingesetzt oder mit ihren eigenen Methoden die hier beschriebenen Projekte geprägt.

Die Studierenden am ITP haben die Grenzen des Möglichen in diesem Bereich immer wieder erweitert. Ihr Beitrag findet sich in vielen Projekten wieder. Ich werde Personen an den entsprechenden Stellen erwähnen, aber allgemein möchte ich allen Studierenden danken, die mein Seminar „Vernetzte Objekte“ besucht haben. Sie haben mir geholfen zu verstehen, worum es bei all dem geht. Die Jahrgänge 2006 und 2007 waren besonders einflussreich, weil sie alles aus den ersten Entwürfen dieses Buchs lernen mussten. Sie haben außerdem diverse wichtige Fehler im Manuskript gefunden.

Mehrere Einzelpersonen haben mit großen Mengen an Code, Ideen oder Arbeit geholfen. Geoff Smith war der Urheber der Seminarbezeichnung »Vernetzte Objekte« und hat mich an die Idee von objektorientierter Hardware herangeführt. John Schimmel hat mir gezeigt, wie man HTTP-Aufrufe mit einem Mikrocontroller macht. Dan O'Sullivan's Server-Code war der Ausgangspunkt meines gesamten Server-Codes. Mein gesamter Processing-Code ist dank Dan Shiffmans Ratschlägen zum Programmierstil besser lesbar. Robert Faludi hat viele Zeilen Code beige-steuert, hat die Lesbarkeit der XBee-Beispiele in diesem Buch verbessert und darin viele Fehler behoben. Max Whitney hat mir dabei geholfen, die Bluetooth-Vermittlung

zum Laufen zu bringen und das Katzenbett fertigzustellen (trotz ihrer Allergien!). Dennis Crowley hat mir die Möglichkeiten und Grenzen von 2-D-Barcodes verdeutlicht. Chris Heathcote hat meine Ideen zur Ortung stark beeinflusst. Durrell Bishop hat mir dabei geholfen, über Identität nachzudenken. Mike Kuniavsky und die Teilnehmer der „Sketching in Hardware“-Workshops der Jahre 2006 und 2007 haben mir geholfen, diese Arbeit als Teil einer größeren Community zu sehen, und mir sehr viele neue Werkzeuge vorgestellt. Die Katze Noodles musste ziemlich viel Unsinn ertragen, damit das Katzenbett fertiggestellt und fotografiert werden konnte. Für die Herstellung dieses Buches sind keine Tiere zu Schaden gekommen, eines wurde jedoch mit Katzenminze bestochen.

Casey Reas und Ben Fry haben die Softwareseite dieses Buchs ermöglicht, indem sie Processing erschaffen haben. Ohne Processing wäre der Softwarebestandteil von vernetzten Objekten sehr viel schmerzhafter. Ohne Processing gäbe es keine einfache, elegante Programmierschnittstelle für Arduino und Wiring. Die Erfinder von Arduino und Wiring haben die Hardwareseite möglich gemacht: Massimo Banzi, Gianluca Martino, David Cuartielles und David Mellis für Arduino, Hernando Barragán für Wiring und Nicholas Zambetti für die Zusammenführung beider. Ich hatte das Glück, mit ihnen zusammenarbeiten zu dürfen.

Obwohl ich versucht habe, viele Hardwarehersteller in diesem Buch zu nutzen und auf sie zu verweisen, muss ich Nathan Seidle von SparkFun gesondert erwähnen. Dieses Buch wäre ohne ihn nicht das, was es ist. Während ich jahrelang von objektorientierter Hardware erzählte, haben Nathan und die Leute von SparkFun sie stillschweigend zur Realität werden lassen.

Danke auch an das Support-Team bei Lantronix. Ihre Produkte sind gut und ihre Unterstützung ist hervorragend. Garry Morris, Gary Marrs und Jenny Eisenhauer haben meine zahllosen E-Mails und Telefonanrufe hilfreich und gut gelaunt beantwortet.

Die Projekte in diesem Buch greifen die Ideen von vielen Kollegen aus aller Welt auf, die ich in Gesprächen bei Workshops und Besuchen erfahren habe. Vielen Dank an die Dozenten und Studenten, mit denen ich am „Interaction Design“-Programm des Royal College of Art, am „Digital Media | Arts“-Programm der UCLA, am „Interaction Design“-Programm an der Oslo School of Architecture and Design, am Interaction Design Institute Ivrea und dem Copenhagen Institute of Interaction Design zusammengearbeitet habe.

Viele Projekte mit vernetzten Objekten haben mein Schreiben inspiriert. Ich danke allen, deren Arbeit die Kapitel bebildert: Tuan Anh T. Nguyen, Joo Youn Paek, Doria Fan, Mauricio Melo und Jason Kaufman; Tarikh Korula und Josh Rooke-Ley von Uncommon Projects; Jin-Yo Mok, Alex Beim, Andrew Schneider, Gilad Lotan und Angela Pablo; Mouna Andraos und Sonali Sridhar; Frank Lantz und Kevin Slavin von Area/Code, sowie Sarah Johansson.

Für MAKE zu schreiben, war eine großartige Erfahrung. Dale Dougherty hat mich bei allen meiner Ideen ermutigt, geduldig meine Verzögerungen ertragen und es mir gegönnt, neue Dinge auszuprobieren. Er hat nie Nein gesagt, ohne eine akzeptable Alternative anzubieten (die oft auch besser war). Brian Jepson hat weit über seine Anforderungen als Lektor hinaus alle Projekte nachgebaut. Änderungen vorgeschlagen, Programme

debuggt, bei den Fotos und Abbildungen geholfen und mich pausenlos ermutigt. Es ist keine Untertreibung, dass ich es ohne ihn nicht hätte schaffen können. Ich hätte mir keinen besseren Lektor wünschen können. Danke an Nancy Kotary, die das Manuskript exzellent redigiert hat. Katie Wilson hat dafür gesorgt, dass dieses Buch besser aussieht und einfacher lesbar ist, als ich jemals gehofft hätte. Ein Dank auch an Tim Lillis für die Illustrationen. Danke an alle im MAKE-Team.

Danke an meine Agenten: Laura Lewin, die den Ball ins Rollen brachte, Neil Salkind, der danach übernommen hat, und dem ganzen Team bei Studio B. Zu guter Letzt vielen Dank meiner Familie und meinen Freunden, die mir zuhören mussten, wie ich in der Entstehungszeit des Buches sowohl begeistert vor mich hingeredet als auch mich bitter beschwert habe. Meine Liebe gilt euch allen.

X

“ Hinweis zur zweiten Auflage

Zwei allgemeine Veränderungen haben die Neufassung dieses Buchs nötig gemacht: Die Entstehung einer Open Source-Hardwarebewegung und das Wachstum einer Mitmachkultur, besonders beim Bau von interaktiven Objekten. Die Arduino-Community und die Open Source-Hardwarebewegung im Allgemeinen sind schnell gewachsen. Die Auswirkungen davon kann man gerade erst feststellen, aber eines ist klar: Objektorientierte Hardware und Physical Computing werden Teil des Alltags. Sehr viel mehr Menschen als ich es mir 2005 vorgestellt habe, bauen heutzutage elektronische Dinge.

Bevor eine Technologie in den allgemeinen Gebrauch übernommen wird, muss es einen Platz für sie in der Vorstellungskraft der Menschen geben. Auch Menschen, die kein Wissen über die Technologie haben, müssen eine gewisse Ahnung haben, was sie ist und wofür man sie benutzen kann. Bis zum Jahr 2005 habe ich sehr viel Zeit darauf verwendet, Menschen zu erklären, was Physical Computing ist und was ich meine, wenn ich von „vernetzten Objekten“ spreche. Heutzutage kann jeder den Wii-Controller oder die Kinect als Beispiel für ein Gerät nennen, das den Bereich des körperlichen Ausdrucks des Menschen, den Computer nutzen können, erweitert hat. Inzwischen es es schwierig geworden, elektronische Geräte zu finden, die nicht vernetzt sind.

Zwar ist es toll zu erleben, dass diese Ideen allgemein besser verstanden werden. Es ist aber noch aufregender, dass sie auch mehr genutzt werden. Die Menschen nutzen ihre Kinects nicht nur für Spiele, sondern bauen damit unterstützende Interfaces für ihre körperbehinderten Kunden. Sie spielen nicht nur mit der Wii, sie benutzen sie auch als Steuerung für Musikinstrumente. Die Leute haben sich daran gewöhnt, dass sie die Nutzung ihrer elektronischen Geräte verändern können, und sie tun es auch.

Als ich mich dem Projekt anschloss, hoffte ich, dass der Arduino vielleicht das Bedürfnis nach einer Lösung stillen könnte, die einfacher anpassbar war als damalige Unterhaltungselektronik und gleichzeitig einfacher zu erleben als Mikrocontroller-Systeme. Ich hielt den Open Source-Ansatz für einen guten Weg, weil er bedeutete, dass die Ideale der Plattform im besten Falle nicht nur auf unsere Modelle beschränkt bleiben würden. Diese Hoffnung hat sich in den zahlreichen davon abgeleiteten Modulen, Shields und Nebenprodukten und dem Zubehör bewahrt, die in den letzten paar Jahren aufgetaucht sind. Es ist wunderbar, dass so viele Menschen nicht nur Elektronik

bauen, auf deren Basis andere weiterarbeiten können, sondern es auch so tun, dass man ohne Fachkompetenz einsteigen kann.

Die Zunahme bei Arduino-Shields und -Programmibliotheken ist so groß, dass ich diese Neuauflage fast so hätte schreiben können, dass du überhaupt nicht selbst programmieren oder eine Schaltung aufbauen müsstest. Für fast jedes Projekt in diesem Buch gibt es ein Shield oder eine Bibliothek. Allerdings kannst du nicht besonders viel lernen, wenn du nur fertige Teile zusammensteckst, daher habe ich versucht, einige der Grundsätze der elektronischen Kommunikation und physischer Schnittstellen zu zeigen. Wenn es eine einfache Hardwarelösung gibt, habe ich das vermerkt, aber auch die zugrunde liegende Schaltung angegeben. Die besten Codebibliotheken und Schaltungen machen etwas, das ich als „Glasgehäuse“ verstehe: Sie umschließen die hässlichen Details und bieten eine komfortable Schnittstelle an, aber falls du interessiert bist, kannst du hineinschauen und beobachten, was vor sich geht. Darüber hinaus sind sie so gut konstruiert, dass die hässlichen Details gar nicht so hässlich aussehen, wenn man sie aus der Nähe betrachtet. Ich hoffe, dass dieses Buch auch so funktioniert.

Software-Referenz

Seitdem ich mit dieser Auflage angefangen habe, hat es viele größere Veränderungen bei der Arduino-Plattform gegeben. Als dieses Buch geschrieben wurde, war die Arduino-IDE noch in der Betaphase. Wenn du das hier liest, ist Version 1.0 bereits erschienen. Wenn du dich bereits mit dem Arduino auskennst, solltest du sicherstellen, dass du die IDE in Version 1.0 oder neuer heruntergeladen hast. Dieses Buch wurde mit Arduino 1.0 beta1 geschrieben, die du online unter <http://code.google.com/p/arduino/wiki/Arduino1> findest. Die fertige Version 1.0 ist auf der Downloadseite unter www.arduino.cc zu finden. Auf der Arduino-

Website findest du immer die letzten Updates. Den Programmcode aus diesem Buch findest du online in meinem GitHub-Repository unter <https://github.com/tigoe/MakingThingsTalk2>. Alle Änderungen finden sich auch (auf Englisch) in meinem Blog www.makingthingstalk.com.

Hardware-Referenz

Um den Fokus auf die Kommunikation zwischen physischen Geräten zu legen, habe ich mich für diese Ausgabe für den Arduino Uno als Referenz-Hardware entschieden. Alles in diesem Buch funktioniert mit einem Arduino Uno mit dem entsprechenden Zubehör oder Shields. Einige Projekte nutzen die spezielleren Arduino-Modelle wie den Arduino Ethernet oder Arduino LilyPad, weil die jeweilige Bauform am sinnvollsten war, aber auch diese Projekte wurden mit dem Uno getestet. Jedes Teil, das mit dem Uno kompatibel ist, sollte in der Lage sein, diesen Code auszuführen und mit diesen Schaltungen zu funktionieren.

Danksagungen für die zweite Auflage

Das Netz der Menschen, die dieses Buch möglich machen, wächst immer weiter.

Die Änderungen in dieser Ausgabe sind im Wesentlichen der Arbeit meiner Partner im Arduino-Team geschuldet. Die Zusammenarbeit mit Massimo Banzi, David Cuartielles, Gianluca Martino und David Mellis ist weiterhin erfreulich, herausfordernd und voller Überraschungen. Ich kann mich glücklich schätzen, sie als Mitstreiter zu haben.

Das Interactive Telecommunications Program an der NYU unterstützt mich weiterhin in meiner beruflichen Arbeit. Nichts hiervon wäre ohne das Engagement meiner Kollegen dort möglich. Dan O'Sullivan war wie immer ein geschätzter Berater für viele der behandelten Projekte. Daniel Shiffman und Shawn Van Every halfen bei den Desktop- und Android-Versionen von Processing. Marianne Petit, Nancy Hechinger, Clay Shirky und Marina Zurkow lieferten kritische und moralische Unterstützung. Nach wie vor inspiriert mich auch Red Burns immer wieder, Menschen neue Möglichkeiten an die Hand zu geben, indem man ihnen beibringt, die Technologien, die ihr Leben gestalten, zu verstehen.

Die Gruppe von wissenschaftlichen Mitarbeitern und Privatdozenten am ITP ist ständig im Wechsel, aber immer hilfreich. Bei der Arbeit an dieser Ausgabe halfen die Research Residents Mustafa Bağdatlı, Caroline Brown,

Jeremiah Johnson, Meredith Hasson, Liesje Hodgson, Craig Kapp, Adi Marom, Ariel Nevarez, Paul Rothman, Ithai Benjamin, Christian Cerrito, John Dimatos, Xiaoyang Feng, Kacie Kinzer, Zannah Marsh, Corey Menscher, Matt Parker und Tym Twillman mit Beispielen, testeten Projekte und unterstützten die tägliche Arbeit am ITP, wenn ich nicht verfügbar war. Die Lehrbeauftragten Thomas Gerhardt, Scott Fitzgerald, Rory Nugent und Dustyn Roberts erwiesen sich als wertvolle Mitarbeiter, indem sie diese Stoffe im Kurs „Einführung in Physical Computing“ behandelten.

Rob Faludi ist weiterhin meine Quelle für alles, was mit XBee und Digi zu tun hat.

Dank an Antoinette LaSorsa und Lille Troelstrup von der Adaptive Design Association für die Erlaubnis, ihr Neigungs Brett-Konzept in Kapitel 5 zu verwenden.

Viele Menschen trugen auf unserer Entwickler-Mailingliste zur Entwicklung von Arduino bei. Insbesondere haben Mikal Hart, Michael Margolis, Adrian McEwen und Limor Fried dieses Buch durch ihre Arbeit an den wichtigsten Kommunikationsbibliotheken wie SoftwareSerial, Ethernet und TextFinder beeinflusst, und ebenso durch ihre persönliche Beratung und Gutmütigkeit bei der Beantwortung meiner vielen Fragen außerhalb der Mailingliste. Michael Margolis' *Arduino Kochbuch* (O'Reilly, ISBN 978-3-86899-353-0) hat ebenfalls als Codereferenz für Programme in diesem Buch gedient. Dank auch an Ryan Mulligan und Alexander Brevig für ihre Programmbibliotheken, die ich für dieses Buch benutzt und angepasst habe.

Limor Fried und Phillip Torrone, die Inhaber von Adafruit, waren ständige Berater, Kritiker und Cheerleader für dieses Buch. Ebenso ist Nathan Seidle bei SparkFun weiterhin einer meiner wichtigsten Kritiker und Berater. Adafruit und SparkFun sind meine wichtigsten Quellen für Bauelemente, weil sie Sachen herstellen, die gut funktionieren.

Diese Ausgabe hat dank Fritzing bessere Grafiken. Fritzing ist ein Open Source-Programm für Schaltpläne, zu finden unter <http://fritzing.org>. Reto Wettach, André Knörig und Jonathan Cohen haben ein tolles Hilfsmittel erschaffen, um Schaltungen und Schaltpläne einfacher zugänglich zu machen. Danke an Ryan Owens von SparkFun dafür, dass er mir Vorabzugriff auf mehrere Bauteilzeichnungen gewährt hat. Danke auch an Giorgio Olivero und Jody Culklin für zusätzliche Zeichnungen in dieser Ausgabe.

Danke an David Boyhan, Jody Culkin, Zach Eveland und Gabriela Gutiérrez, die Teile des Manuskripts gelesen und kritisiert haben.

Danke an Keith Casey von Twilio, Bonifaz Kaufmann, den Autor von Amarino, Andreas Göransson für seine Hilfe zu Android und Casey Reas und Ben Fry für den Android-Modus von Processing sowie für Feedback zum Abschnitt über Android.

Die neuen Teile in diesem Buch wurden von neuen Projekten inspiriert. Danke an Benedetta Piantella und Justin Downs von Groundlab und an Meredith Hasson, Ariel Nevarez und Nahana Schelling, die Schöpfer von SIMbalink. Danke an Timo Arnall, Einar Sneve Martinussen und Jørn Knutsen von www.nearfield.org für ihre Inspiration und Mitarbeit beim Thema RFID. Danke an Daniel Hirschmann, weil er mich daran erinnert hat, wie spannend Beleuchtung ist und wie einfach DMX-512 sein kann. Danke an Mustafa Bağdatlı für seinen Rat zu Poker Face und an Frances Gilbert und Jake für ihre Rolle im CatCam-2-Projekt. Entschuldigung an Anton Tschechow. Danke an Tali Padan für die komödiantische Inspiration.

Danke an Giana Gonzalez, Younghui Kim, Jennifer Magnolfi, Jin-Yo Mok, Matt Parker, Andrew Schneider, Gilad Lotan, Angela Pablo, James Barnett, Morgan Noel, Noodles und Monski für den Bau von Projekten in den Kapiteln.

Danke wie immer an das MAKE-Team, insbesondere an meinen Lektor und Mitstreiter Brian Jepson. Seine Geduld und Beharrlichkeit haben eine neue Ausgabe ermöglicht. Danke an den technischen Redakteur Scott Fitzgerald, der ebenfalls geholfen hat, alle Teile zusammenzusuchen. Wenn du ein Bauteil aus diesem Buch im Internet findest, solltest du Scott danken. Vielen Dank auch an meinem Agenten Neil Salkind und allen bei Studio B.

In den letzten Wochen der Arbeit an dieser Ausgabe hat mir eine Gruppe enger Freunde geholfen und mir ermöglicht, was ich nicht alleine geschafft hätte. Zach Eveland, Denise Hand, Jennifer Magnolfi, Clive Thompson und Max Whitney haben ihre Tage und Abende geopfert, um beim Sägen, Löten, Verkabeln und Zusammenbauen von vielen der letzten Projekte zu helfen, und sie haben mir beim Schreiben Gesellschaft geleistet. Joe Hobaica, der mehrere Tage investierte, kümmerte sich um die Produktionsleitung, um das Buch fertigzustellen. Er inszenierte die fotografische Dokumentation der meisten neuen Projekte, organisierte meinen Workflow, führte Aufgabenlisten, kaufte diverse Bauteile, überprüfte die Kontinuität und erinnerte mich daran, zu essen und zu schlafen. Sie alle haben mich daran erinnert, dass man Dinge am besten zusammen mit Freunden zum Reden bringt.

X