

# Einleitung

Der Raspberry Pi ist ein Minirechner, den Sie sehr flexibel für die unterschiedlichsten Aufgaben einsetzen können. Zum einen haben Sie mit dem gerade mal scheckkartengroßen Rechner einen vollwertigen Desktop-PC, auf dem Sie Office-Anwendungen laufen lassen und Ihre E-Mails bearbeiten können. Auch Drucken, Surfen, Netzwerkverbindung über Kabel, Wireless LAN und Bluetooth gehören dazu. Zum anderen können Sie an dem Minirechner eine Maus und Tastatur über die USB-Ports und einen Bildschirm über HDMI oder einen Fernseher über Composite anschließen. Oder Sie betreiben den Raspberry Pi über einen Remote-Zugriff. Für diese Aufgaben unterstützt Sie dabei das für den Raspberry Pi optimierte Debian-Linux. Aber keine Angst, auch der ungeübte Anwender, der mit der Konfiguration eines Linux-Rechners nicht vertraut ist, wird so weit unterstützt, dass das Debian-Linux ohne Vorkenntnisse installiert und konfiguriert werden kann.

Die zweite Welt, in die Sie mit dem Raspberry Pi eintauchen können, ist die faszinierende Verbindung von Software und Hardware. Diese reicht von einfachen digitalen Ein- und Ausgängen bis hin zur Ansteuerung von komplexen Schaltkreisen, die über den I<sup>2</sup>C-Bus, den SPI-Bus oder die RS232-Schnittstelle angesteuert werden können. Über diese Schnittstellen lassen sich dann die unterschiedlichsten Sensoren ansprechen. Den Zugang zu diesen Systemen stellt der Raspberry Pi über eine 26-polige oder je nach Modell auch über eine 40-polige Stiftleiste zur Verfügung. Und genau dahin möchte ich Sie mit diesem Buch begleiten.

Alle Projekte aus diesem Buch werden mit der Programmiersprache Java umgesetzt und es soll nicht nur den erfahrenen Java-Entwickler, sondern auch Umsteiger, die vielleicht bisher mit C oder Python Erfahrung haben, ansprechen. Aber auch alle Technikfreaks, die einfach neugierig sind und etwas Neues ausprobieren wollen. Für diesen Leserkreis gibt es im Anhang dieses Buches eine Einführung in Java, in der die grundlegenden Strukturen und Prinzipien dieser Hochsprache erläutert werden. In den Projekten finden Sie dann weiterführende Themen zu Java.

## Wie Sie mit diesem Buch arbeiten können

Das Buch ist so konzipiert, dass die Kapitel aufeinander aufbauen. So gibt es zum Beispiel Java-Klassen, die einmal ausführlich erläutert und dann in nachfolgenden Kapiteln eingesetzt werden. Der gesamte Programmcode teilt sich dabei in einen allgemeinen und in einen für das jeweilige Kapitel spezifischen Teil. Den allgemeinen Teil des Programmcodes können Sie dann später sehr einfach in eigene Projekte einbinden und er bildet in seiner Summe ein kleines Framework für die Java-Programmierung auf dem Raspberry Pi.

Den ganzen Programmcode, den Sie für die Umsetzung der Projekte benötigen, können Sie sich von dem Verlagsserver unter [www.miptp.de/055](http://www.miptp.de/055) herunterladen (unter dem Reiter Downloads) und auf dem Raspberry Pi installieren. Eine Anleitung dazu finden Sie in dem Kapitel »Projektvorbereitung«.

Alle Programme sind direkt auf dem Raspberry Pi entwickelt worden. Hierzu wurde die Entwicklungsumgebung BlueJ verwendet, die auch in diesem Buch vorgestellt wird und mit der Sie die wichtigsten Grundlagen für den Umgang an die Hand bekommen.

Da in diesem Buch die Verbindung zwischen Soft- und Hardware beschrieben wird, sind die Projekte fast ausschließlich technischer Natur. Zu jedem Programmcode gehört demnach auch immer ein Stück Elektronik. Hierbei habe ich stets versucht, den Aufwand für Sie möglichst gering zu halten. Es lassen sich daher alle Versuche sehr einfach auf einem Steckbrett nachbauen.

Für einen ersten Überblick über die Themen finden Sie im Folgenden eine kurze Beschreibung zu jedem Kapitel.

### **Kapitel 1: Der Raspberry Pi**

In dem ersten Kapitel lernen Sie die Hardware des Raspberry Pi kennen, indem die verschiedenen Komponenten und Schnittstellen auf dem Board beschrieben werden. Hierzu zählen der Prozessor, die Grafikeinheit, die USB-Anschlüsse, das Netzwerk und die Audio- und Video-Ausgabe. Hierbei wird auch auf die Unterschiede zwischen den verschiedenen Modellen eingegangen. Außerdem wird beschrieben, wo Sie das Kameramodul und ein Display anschließen können.

Für den ersten Start des Raspberry Pi benötigen Sie eine SD-Karte, eine Tastatur, eine Maus und ein HDMI- oder Composite-Kabel und ein Netzteil. Damit Sie hier die richtigen Komponenten anschließen, zeige ich Ihnen, auf welche technischen Werte Sie achten müssen.

### **Kapitel 2: Booten und konfigurieren**

Nachdem alle Komponenten für einen ersten Testlauf des Raspberry Pi verfügbar sind, wird beschrieben, wie Sie das richtige Betriebssystem für den Raspberry Pi auf eine SD-Karte kopieren können und wie Sie den Raspberry Pi konfigurieren. Anschließend werden noch verschiedene Möglichkeiten vorgestellt, um den Raspberry Pi in Ihrem Heimnetzwerk über LAN oder WLAN zu integrieren. Danach wird der Mini-Rechner so konfiguriert, dass Sie über SSH mit PuTTY und WinSCP auf den Raspberry Pi zugreifen können. Weiterhin beschreibe ich, wie Sie mit TightVNC eine Remote-Verbindung aufbauen, um so von einem anderen Rechner auf den grafischen Desktop des Raspberry Pi zuzugreifen.

### **Kapitel 3: Linux auf dem Raspberry Pi**

Auch wenn der Raspberry Pi über einen grafischen Desktop verfügt, werden Sie für bestimmte Aufgaben mit der Linux-Konsole arbeiten müssen. Um hier die wichtigsten Befehle kennenzulernen, finden Sie diese in diesem Kapitel nebst ihrer Anwendung.

## Kapitel 4: Projektvorbereitung

In der Projektvorbereitung wird die Java-Entwicklungsumgebung BlueJ installiert und ich beschreibe die wichtigsten Funktionen. Sie installieren dann die Pi4J-Bibliothek, die Sie benötigen, damit Sie die GPIO-Ports des Raspberry Pi ansprechen können. Dies ist die Schnittstelle für alle technischen Projekte.

Wenn Sie nicht direkt mit einem Steckbrett arbeiten wollen, um die ersten Versuche aufzubauen, zeige ich Ihnen, wie Sie das Erweiterungsboard Gertboard mit dem Raspberry Pi verbinden und mit einem einfachen Java-Programm testen können. Auf diesem Board finden Sie einige Schaltkreise, die auch in den verschiedenen Projekten benutzt werden.

## Kapitel 5: Datum und Zeit

Im Kapitel »Datum und Zeit« werden Ihnen der Umgang mit Datum und Zeit in Java und die dafür bereitgestellten Klassen `Date`, `Calendar` und `GregorianCalendar` vorgestellt. Hierzu gehört auch das Formatieren und Parsen von Datums- und Zeitangaben. Als praktische Übung schreiben Sie zuerst ein Programm, das den Wochentag Ihres Geburtstags für ein beliebiges Jahr berechnen kann. Und danach kommt die Berechnung der Arbeitstage für ein beliebiges Jahr, die zwischen Weihnachten und Silvester liegen.

Als Nächstes werden Sie das GPIO-Interface des Raspberry Pi über Java ansprechen und Relais ansteuern. Als praktische Übung werden hierzu mit zeitgesteuerten Relais Akkus geladen und eine Spiegelreflexkamera gesteuert.

## Kapitel 6: Digitale Eingänge und Sensoren

Das Thema »Digitale Eingänge und Sensoren« zeigt Ihnen, wie Sie mithilfe der Pi4J-Bibliothek auf externe Ereignisse reagieren können. Ein Programm muss also einen Pegelwechsel, der an einem der GPIO-Pins ausgeführt wird, erfassen und darauf reagieren. Als erstes Beispiel werden Sie hierzu ein Programm schreiben, das einen Tastendruck auswertet und daraufhin einen externen Prozess startet. Unter einem externen Prozess versteht man hier zum Beispiel das Ausführen eines Linux-Befehls oder eines beliebigen Programms. In diesem Beispiel wird durch den Tastendruck der Raspberry Pi heruntergefahren und der TightVNC-Server gestartet oder beendet.

Als zweites Anwendungsbeispiel wird ein Bewegungsmelder eingesetzt, um in einer Toilette das Licht beim Betreten einzuschalten und Musik zu spielen. Sie lernen also auch, wie man mit Java Audio-Dateien abspielt.

Den Ultraschallsensor HC-SR04, den man zur Abstandsmessung einsetzt, werden Sie mit einem weiteren Programm testen. Hierbei muss der Raspberry Pi zwei sehr schnell hintereinander auftretende Signale auswerten.

Als Letztes werden dann analoge Sensoren über einen Schmitt-Trigger an den Raspberry Pi angeschlossen, um somit als Schwellenwerterkennung eingesetzt zu werden.

## Kapitel 7: Swing lernen

Das Kapitel »Swing lernen« ist ein reiner Java-Kurs, den Sie auch auf jedem anderen Computer testen können. Hier lernen Sie, wie mit Java grafische Oberflächen erzeugt werden und auf Ereignisse reagiert werden kann. Da die Entwicklung grafischer Oberflächen ein sehr komplexes Thema ist, wird hier nur auf einen kleinen Bereich eingegangen, der aber

alles abdeckt, was Sie für die Projekte in diesem Buch wissen müssen. Es dient Ihnen aber auch als Grundlage, um tiefer in dieses Thema einzusteigen.

## **Kapitel 8: Analoge Eingänge und Sensoren**

Um mit dem Raspberry Pi analoge Signale verarbeiten zu können, benötigen Sie einen externen Analog-Digital-Wandler. Im Falle des Raspberry Pi können Sie Wandler verwenden, die entweder über einen SPI- oder einen I<sup>2</sup>C-Bus verfügen. Als Einstieg in dieses Thema wird der Wandler MCP3002 vorgestellt, der auch auf dem Gertboard zu Verfügung steht und über den SPI-Bus angesteuert wird. Hierzu beschreibe ich zuerst die grundlegende Arbeitsweise des SPI-Bus und wie Sie diesen mit Java über die Pi4J-Bibliothek ansprechen können.

Als analoge Sensoren werden hier ein NTC-Widerstand zur Temperaturmessung und ein LDR-Widerstand zur Messung der Lichtstärke eingesetzt. Außerdem zeige ich Ihnen, wie Sie die Kennlinie solcher Sensoren linearisieren und entsprechende Verstärker zur Anpassung an den Wandler berechnen können.

Da es auch vorkommen kann, dass in einem Datenblatt zu einem Sensor keine Kennlinie, sondern eine Tabelle mit Stützwerten angegeben wird, zeige ich Ihnen, wie Sie diese Stützwerte in Java verarbeiten und Zwischenwerte interpolieren können. Als Beispiel verwenden Sie hierzu ein Programm für die Distanzsensoren der Firma Sharp.

## **Kapitel 9: Messwerte speichern**

Immer wenn Sie mit Sensoren arbeiten, werden Sie die unterschiedlichsten Messwerte speichern und aus einer Datei auslesen müssen. Hierzu habe ich einen Java-Kurs geschrieben, der das Arbeiten mit Dateien und Verzeichnissen erläutert. Außerdem lernen Sie die wichtigsten Java-Klassen kennen, um Daten als Liste, Menge oder in einem assoziativen Speicher abzulegen und formatiert zu speichern. Als praktisches Beispiel werden Sie hierzu einen Datenlogger schreiben, der beliebige Messwerte verarbeiten kann.

## **Kapitel 10: Messwertgrafiken mit Swing**

In diesem Kapitel zeige ich Ihnen, wie Sie mit Java eigene grafische Komponenten entwickeln können. Diese können Sie dann genauso in eigene Programme einsetzen, wie Sie es mit den Swing-Komponenten aus der Java-Bibliothek gewohnt sind. Als Beispiel werden Sie hier ein Diagramm programmieren, mit dem Sie Messwerte im zeitlichen Verlauf grafisch darstellen.

## **Kapitel 11: Threads und reguläre Ausdrücke**

Mit Threads werden in Java nebenläufige Prozesse abgebildet, um eine quasi parallele Verarbeitung von Daten in einer Anwendung zu realisieren. Ich zeige Ihnen in diesem Einführungskurs, wie Sie Threads starten und beenden. Außerdem lernen Sie, wie man auf Daten zugreift, die von mehreren Threads gelesen und geschrieben werden.

Der zweite Block in diesem Kapitel beschäftigt sich mit regulären Ausdrücken, mit deren Hilfe Sie ein sehr mächtiges Werkzeug an die Hand bekommen, um auch komplexe Suchen von Zeichenketten zu realisieren.

Beide Themen aus diesem Kapitel bilden die Grundlage für die nächsten Projekte.

## Kapitel 12: Füllstandsanzeige für eine Zisterne

Mit einem Distanzsensor können Sie den Wasserstand in einer Zisterne ermitteln und zum Beispiel als Funktion über die Zeit grafisch darstellen. Die Bereitstellung der Messwerte und die Anzeige werden hier in zwei getrennte Java-Anwendungen aufgeteilt. Die Messwerte werden über einen HTTP-Server zur Verfügung gestellt, den Sie mit ganz einfachen Mitteln selber schreiben. Die Darstellung und das Abholen der Daten von dem Server übernimmt hier ein HTTP-Client, der sich die Daten über einen HTTP-GET-Befehl erfragt. Die grafische Anzeige wird durch die selbst geschriebene Swing-Komponente realisiert.

Mit diesem Konzept kann der HTTP-Client auf jedem Rechner in Ihrem Heimnetzwerk laufen und sich die Daten von dem HTTP-Server, der auf dem Raspberry Pi gestartet wurde, abholen.

## Kapitel 13: Eine Heizungssteuerung

Die Heizungssteuerung, die in diesem Kapitel beschrieben wird, folgt dem gleichen Softwarekonzept wie im vorherigen Kapitel, nur dass Server und Client um den HTTP-POST-Befehl erweitert werden und Daten vom Client auf den Server geschrieben werden.

Mit der vorgestellten Schaltung können Sie eine Heizung oder einen Heizkörper steuern.

## Kapitel 14: Eine Zeitschaltuhr

Und auch die Zeitschaltuhr teilt sich in Server und Client auf und kann beliebig viele Schaltvorgänge verarbeiten. Sie können damit also Relais für jeden Wochentag beliebig oft ein- und ausschalten. Außerdem zeige ich Ihnen, wie Sie die Anwendung in den Autostart aufnehmen und Desktopverknüpfungen erstellen.

## Kapitel 15: E-Mail-Alarm

Im Kapitel »E-Mail-Alarm« lernen Sie die grundlegenden Methoden, um mit Java E-Mails zu verarbeiten. Die E-Mails werden hierbei über einen Provider verschickt und empfangen. Als Anwendung werden Sie eine kleine Alarmanlage programmieren, die einen Alarm durch einen Bewegungsmelder auslöst und eine E-Mail verschickt. Außerdem lässt sich die Alarmanlage über eine E-Mail einschalten.

## Kapitel 16: Der I<sup>2</sup>C-Bus

Der I<sup>2</sup>C-Bus ist neben dem SPI-Bus der zweite Bus, den der Raspberry Pi auf seinem GPIO-Interface bereitstellt. Wie Sie diesen Bus aktivieren und über die Pi4J-Bibliothek ansprechen, lernen Sie in diesem Kapitel. Außerdem wird das Protokoll des I<sup>2</sup>C-Bus eingehend behandelt, damit Sie das grundlegende Verständnis für die Arbeit mit diesem Bus erlangen. Als Beispiel werden Sie den Temperatursensor MCP9808 an den I<sup>2</sup>C-Bus anschließen und den Sensor in verschiedenen Betriebsmodi ansprechen.

## Kapitel 17: Die serielle Schnittstelle RS232

Auf dem GPIO-Interface des Raspberry Pi liegt auch ein UART-Baustein bereit, mit dem Sie eine RS232-Schnittstelle aufbauen können und damit die Möglichkeit haben, andere Geräte über eine lange Leitung ansprechen zu können. In einer einfachen Anwendung wird hier der Ultraschallsensor SRF02 über die RS232-Schnittstelle mit dem Raspberry Pi

verbunden und über die Pi4J-Bibliothek angesprochen. Außerdem lernen Sie die Leitungstreiber kennen, die Sie für die Anpassung an lange Leitungen benötigen.

## **Kapitel 18: Textausgabe mit LCD-Displays**

Für viele kleine Anwendungen ist es oft hilfreich, wenn Sie die Möglichkeit haben, kurze Texte über ein einfaches Display anzeigen zu können. Hierdurch können Sie oft auf ein teures TFT-Display verzichten. Die Grundlage für dieses Kapitel ist der HD-44780-Displaytreiber, der in sehr vielen Displays verbaut wird. Sie werden hier den Befehlssatz des HD 44780 kennenlernen und verschiedene Testprogramme schreiben. Dabei werden Sie ein Display im 4-Bit- und 8-Bit-Modus ansteuern. Und Sie werden den IO-Expander MCP23017 einsetzen, um ein einfaches Display über den I<sup>2</sup>C-Bus anzusteuern.

## **Anhang A: Java**

Wenn Sie mit Java noch keine Erfahrung haben, sollten Sie sich im Anhang den Grundkurs Java anschauen. Dort habe ich die wichtigsten Grundlagen in einem Kurs zusammengefasst. Sie lernen als Erstes das Grundkonzept von Java und der objektorientierten Programmierung kennen. Anschließend erörtere ich, wie in Java mit Variablen, Arrays und Operatoren gearbeitet wird und wie Sie den Programmfluss kontrollieren können. Danach erfolgt eine Einführung in das objektorientierte Programmieren mit Java und die dafür notwendige Syntax. Abschließend bekommen Sie einen Überblick über Wrapper-Klassen, die Fehlerbehandlung in Java und die Organisation von Klassen in sogenannte Packages. Tiefer gehende Themen zu Java werden dann noch in verschiedenen Kapiteln behandelt, die einen konkreten Bezug zu den beschriebenen Projekten haben.

## **Über den Autor**

Nach dem Studium der Elektrotechnik arbeitet Wolfgang Höfer seit über 20 Jahren als Software-Entwickler. Im Laufe seines Berufslebens hat er Projekte in den Programmiersprachen C, Pascal, Basic, Assembler, PL/SQL und Java umgesetzt. Hierbei hat er in den letzten 15 Jahren mit Java hauptsächlich Datenbank-gestützte Client/Server- und Webanwendungen entwickelt. Aus Interesse zur Hardware hat er sich in seiner Freizeit mit der Programmierung von ATMEL-Controllern beschäftigt und damit verschiedene Projekte umgesetzt. Dabei ging es hauptsächlich um die Auswertung von Sensoren und einfachen regelungstechnischen Aufgaben aus der Hausautomation. Bei diesen Projekten fehlte aber immer wieder die Anbindung eines Controllers an einen leistungsstarken Rechner, mit dem Benutzeroberflächen, Webanwendungen, Webservices und die Speicherung von größeren Datenmengen realisiert werden konnten. Da ein normaler PC wegen des relativ hohen Stromverbrauches für den Dauereinsatz ausschied, suchte er nach preiswerten Alternativen. Und genau diese hat er dann mit dem Raspberry Pi gefunden. Die leichte Integration in ein Computernetzwerk und die Fähigkeit, verschiedene Bussysteme und Schnittstellen ansteuern zu können, machen den Raspberry Pi zu einem perfekten System, um Steuer- und Sensorelektronik mit modernen Anwendungen zu verbinden.