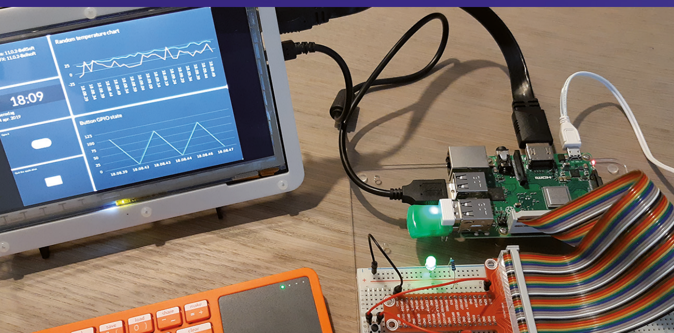
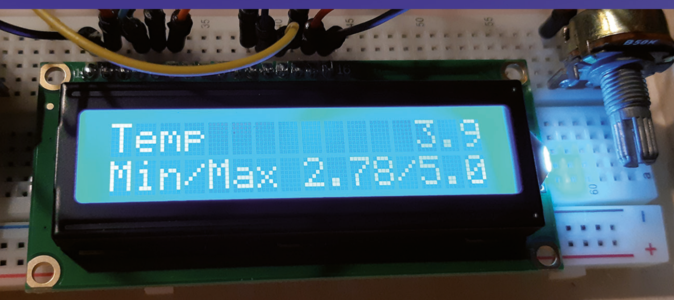
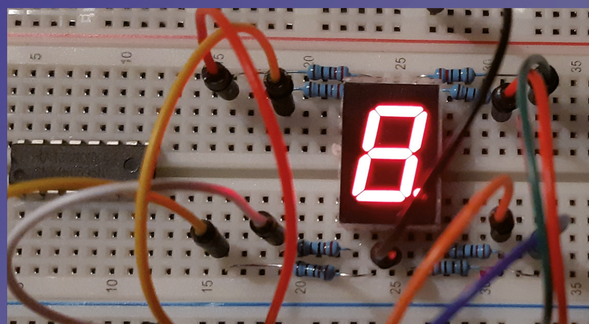
Getting Started with Java on the Raspberry Pi







OpenJDK

spring

JavaFx

Pi4J

Frank Delporte

elektor

LEARN DESIGN SHARE

Getting Started with Java
on the Raspberry Pi

Frank Delporte

elektor
LEARN DESIGN SHARE

● This is an Elektor Publication. Elektor is the media brand of
Elektor International Media B.V.
78 York Street, London W1H 1DP, UK
Phone: (+44) (0)20 7692 8344

● Declaration

The author and publisher have used their best efforts in ensuring the correctness of the information contained in this book. They do not assume, or hereby disclaim, any liability to any party for any loss or damage caused by errors or omissions in this book, whether such errors or omissions result from negligence, accident or any other cause..

● British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

● **ISBN 978-1-907920-91-2**

Prepress Production: D-Vision, Julian van den Berg
First published in the United Kingdom 2020

Chapter 1: Introduction 13

Content overview 14

About me 15

Sources and scripts used in this book 16

Where to find them 16

Get the sources 16

The styling used in the book 17

Read the README! 18

Guidelines 18

Example 18

What's next? 19

Thanks to…. 19

Chapter 2: Tools and hardware used in this book 21

Raspberry Pi 21

Prepare the Pi 22

Connections between Pi and breadboard 26

Software tools on the Pi 28

Linux commands crash course 28

Firefox 32

VNC server 32

Enable SSH on the Pi 33

Free software tools on PC 34

Integrated development environment aka IDE 34

Remote connection to a Raspberry Pi (SSH) 34

Wiring diagrams 37

Schematic drawings 37

Hardware components 38

Resistors 38

LEDs 40

RGB-LED 41

LED-strips 41

Electronic kit with Arduino board 43

Just a thought: Learn by educating 45
Interview with Karen Mouws. 46
Chapter 3 • Choosing an IDE 48
IntelliJ IDEA 48
Using IntelliJ IDEA with the example projects 50
Interview with Trisha Gee. 53
Visual Studio Code (VSC) 55
VSCodium the free non-tracking alternative to VSC 56
Java development with Visual Studio Code 57
Using Visual Studio Code on the PC with code on the Pi 58
Interview with Xiaokai He. 61
Chapter 4: About Java 62
History. 63
Java files versus byte code 64
JVM versus JRE versus JDK 64
JVM = Java Virtual Machine 64
JRE = Java Runtime Environment. 65
JDK = Java Development Kit 65
Version history 65
JDK providers 66
Oracle. 66
AdoptOpenJDK. 66
Azul Zing and Zulu 67
BellSoft Liberica 67
Interview with Alexander Belokrylov 68
Installing the Java JDK 69
Install Java JDK on a Windows PC 69
Install Java JDK on a Linux PC with SDKMAN 70
Install Java JDK on a Raspberry Pi 72
Some of the changes between Java versions 73
Changes between Java 8 and 11 73
What's next after Java 11? 75

Java crash course 77
HelloWorld! Running a single-file Java-application 77
Using the start-up arguments 79
Working with numbers 80
If, Then, Else 81
Enum and Switch 82
Using methods 83
Using objects 84
Reading a text file 87
Using streams 89
What's next? 91
Interview with Jakob Jenkov 92
Chapter 5 • Raspberry Pi pinning 94
Raspberry Pi types 95
Models 95
Major versions 96
Board versions 96
Pin types 98
Power and ground 98
Digital GPIO 98
Pin functions 98
Universal Asynchronous Receiver and Transmitter (UART - Serial) 98
General Purpose Clock (GPCLK) 99
Inter-Integrated Circuit (I²C) 99
Serial Peripheral Interface (SPI) 100
Pulse-Width Modulation (PWM) 100
Header types 101
40-pin header 101
26-pin header - Type 1 and 2 103
8-pin header 103
Different pinning numbering schemes 104
Board (pin) number 104

BCM number 104
WiringPi number 104
Chapter 6 • What is Maven? 107
Install Maven 109
On Windows PC 109
On Raspberry Pi 110
Generate a new Maven project 110
Project structure 110
A minimal pom.xml example 111
Maven pom-files used in this book 112
Add application logging with Maven and log4j 113
Just a thought: Abbreviations 117
Chapter 7 • About JavaFX 119
History 119
Interview with Johan Vos 121
Sample libraries to extend JavaFX 123
TilesFX 123
FXRibbon 123
ControlsFX 124
PickerFX 124
Interview with Gerrit Grunwald 126
Minimal JavaFX 11 sample application 127
Add new archetypes to Maven 127
Creating an empty application 127
Running the empty application from Visual Studio Code 128
Running the application on the Pi 129
Example 1: TilesFX dashboard 131
Wiring and testing in terminal 131
Blink an LED with Java 133
Building our first JavaFX application 134
Run the application on PC 147

Run the application on the Pi 148
Conclusion. 149
Start a Java application when the Pi starts up. 149
Disable screensaver. 150
Example 2: Interact with an I²C relay board. 151
Enable and test I²C 151
Coding the I²C controller application 152
Running the relay controller on the Pi 157
Example 3: Build a UI with FXML 158
Generate an empty FXML project as a starting point. 158
Scene Builder 160
Just a thought - Beware of the PAF 162
Chapter 8 • Bits and bytes 163
Convert bits to a numeric and hex value 164
Calculate a byte value 164
Value ranges in Java 165
Difference between Byte, Short, Integer, and Long. 165
Minimum and maximum values in Java 165
Signed versus unsigned 166
Conclusion. 167
What can we do with this? 168
Web colors 168
Controlling a numeric segment display 169
Chapter 9 • PI4J. 186
Installation. 187
Programming with Pi4J 187
Sources 187
Maven dependencies 187
Running the examples 189
Digital GPIO input and output examples. 191
Example 1: Digital output with RGB-LED 191

Example 2: Digital input with a button 194
Example 3: Distance sensor 198
PWM example 203
Wiring a single LED 203
Code to control an LED with PWM 204
Running the example 205
SPI example with MAX7219 and 8x8 LED-matrix 205
Wiring 206
SPI example code 207
Running the application and created matrix output 218
SPI conclusion 220
Serial communication example with an Arduino 220
Wiring 221
Arduino code 221
Detecting the serial interface on the Pi 223
Raspberry Pi code 223
Running the Java serial application 228
What's next 229
LCD-display with the weather forecast 229
Wiring to connect an LCD to the Pi 230
Get an AppID on OpenWeatherMap 231
Weather LCD application code 231
Running the LCD weather application 242
Conclusion 243
Just a thought: Switching social 244
Chapter 10 • Spring 246
What is Spring Boot? 247
What is Spring Initializr? 247
Interview with Mark Heckler 251
Example 1: Minimal web server on the Pi 252
Start from the Initializr project and modify pom.xml 252

Application properties 253
Image controller 253
Swagger config 258
Run on the Pi 259
Conclusion 259
Example 2: Database REST-service for IoT data on Pi 260
pom.xml settings 260
Creating the database entities 261
Storing data in the database 265
Adding the REST-services 266
Adding Swagger 268
Running the application and using the REST-services 268
Configuration to run on the Pi 272
Conclusion 274
Interview with Vlad Mihalcea 275
Example 3: REST-service on the Pi to toggle an LED 277
Info REST-controller 277
GPIO Manager 278
GPIO REST-controller 280
Running the application on a Pi 281
Conclusion 285
Example 4: Reactive data 285
The code 285
Running the streaming application on the Pi 289
Conclusion 290
Just a thought: Impostor Syndrome 291
Chapter 11 • Message Queues 292
Using Mosquitto on the Pi 293
Installation 293
Testing Mosquitto on the Pi 295
Example 1: Share data between Pi and PC 296

Modifying the pom and module-info 296
Connecting and publishing to Mosquitto 297
Subscribing to Mosquitto 297
The user interface 298
Example 2: Control Arduino from JavaFX via Mosquitto 299
Defining the messages 300
The Arduino part 301
The Java application 309
The finished setup 321
Tip: Checking the network packages between Arduino and Pi 321
Conclusion 323
Used materials 324
Index 325

Chapter 1: Introduction

Welcome! Whether you are a newbie in Java, and want to learn from examples, or know the language, and want to get started with electronics and a Raspberry Pi, I hope you will find valuable new information in here!

This is not a book to learn every aspect of the Java language. There are already a lot of those written by way better programmers than me, so please check your bookstore or an online course if you want to get a real deep knowledge of the Java programming language. My goal was to bundle **a lot of sample code and information**, I collected while learning and experimenting, and to get anyone interested in Java to take **a quick start by using the examples to set up their experiments**. By following these examples, you will also learn the language bit by bit.

Over the last couple of years, I focused on Java in my professional job. But on the other hand, I also got involved in CoderDojo (a coding club for children) where I first was able to make a blinking LED with some simple code on Arduino and a Pi.

If I see how kids learn to work with electronics and programming at CoderDojo, I'm jealous those things didn't exist when I was a kid. Yes, I managed to control a relay-board with my Commodore 64 but it took me years to collect all needed knowledge and components to reach that goal. Now thanks to online shopping and knowledge sharing that same result can be achieved in days or hours…

As a professional Java-developer with a love for open-source, I set myself as a personal goal for 2019 to run a recent Java version (Java 11 or newer) on the Pi, with a **JavaFX user interface, and control an LED** with it. Very basic I know, but it resulted in a series of blog posts about each step to reach this goal. Based on these blog posts, I also wrote an article for MagPi (in the Dutch and French July 2019 edition) and produced an Udemy course[1].

And as a **blinking LED is the equivalent of a "Hello World" program**, I continued with experimenting and ended up with… this getting started book! A whole list of easy "recipes" to get you started with many different Java libraries and electronics to build your projects. Take one or more of these "recipes", combine and mix them to your taste, experiment and explore!

Oh, and this is not an anti-Python or anti-C book! Java is the language I love and use the most, **but for every problem, you need to select the best solution**. And in some cases, this could be Java or something else…

1 https://www.udemy.com/course/use-java-11-and-java-fx-11-on-a-raspberry-pi

Content overview

You can read this book from start to end, but can also use it as a "cookbook" and select the topics of your interest. For instance, you don't need to know the history of Java to work with it, but it's here for your reference.

In between some of the chapters you find a short "Just a thought"-piece about the things I find important and I would like to share with you. I'm also very happy I could interview some of my heroes, included in the chapters they are related to.

Chapter 1: Introduction: On the next pages you can find more info about the sources used in this book and are available for free online, and some additional info about read-me-files and a thank- you-list.

Chapter 2: Tools and hardware used in this book: Setup a Raspberry Pi with Raspbian and an overview of the software used on both Pi and PC. Also generic info about some of the most-used hardware components in this book.

Chapter 3: Choosing an IDE: Info about IntelliJ IDEA and Visual Studio Code.

Chapter 4: About Java: History of Java, different versions and tools within the eco-system and how to install it on your Pi and PC. And of-course a crash course so you get a grasp of the language if it's completely new to you.

Chapter 5: Raspberry Pi pinning: Different types of Raspberry Pi-boards and the different headers, pins and pin types and how you can use them to connect different types of hardware.

Chapter 6: What is Maven?: More info on Maven, the tool we will use to test, build and run our Java projects on PC and Pi.

Chapter 7: About JavaFX: The visual Java framework we will use to build beautiful user interfaces to interact with our Pi and hardware.

Chapter 8: Bits and bytes: The magic of ones and zeros and how they are combined into longer values.

Chapter 9: PI4J: A framework that makes it easier to work with the GPIOs from Java with multiple hardware examples.

Chapter 10: Spring: Building a Java application which exposes our Pi as a web service to store data or control the GPIOs.

Chapter 11: Message Queues: Use your Pi as a message handler to receive and send messages to different devices like other Pi's, PCs or Arduino's.

About me

I'm Frank Delporte (@FrankDelporte[2] on Twitter, °1973) who started programming at age 11 when I got a Commodore 64. With Basic as the programming language, some magnetic switches, and a relay board, I managed to control my Lego train. After studying at a film school, I started my professional career as a video editor, but quickly got involved in the first multimedia and online video projects in Belgium. Combining my interest in multimedia and programming, this resulted in a journey through C#, Flash, HTML/JavaScript, Flex, SQL, Qt, Java…

Currently, I work as a technical lead and software developer at Televic Rail in Belgium with a team of highly skilled hard- and software-engineers. We are building the next generation of Passenger Information Systems used onboard rail vehicles to inform the passengers in the best possible way so they are fully informed about their journey. Making a bridge between on- and off-board data sources empower us to combine information into nice looking and easy to read screens.

I always try to **"get things moving and done"** while trying to stick to the **KISS principle** (see "Important abbreviations").

As the best way to learn is teaching, I was one of the first lead coaches of CoderDojo in Belgium and started two Dojo's (Roeselare and Ieper) where I first came into contact with Arduino, Pi, and all those other amazing tools, all thanks to the inspiring colleague-coaches. On my blog webtechie.be[3], some projects are shared, which resulted in an article for MagPi which was the starting point for this book.

I live together with my wife Conny and son Vik in Passendale (Belgium), a small town that was one of the main battlegrounds of WWI.

2 https://twitter.com/FrankDelporte
3 www.webtechie.be

Sources and scripts used in this book

Within this book, many open-source projects are described and used. So **all the sources and scripts which are given as examples in this book are also shared as open source**.

Where to find them

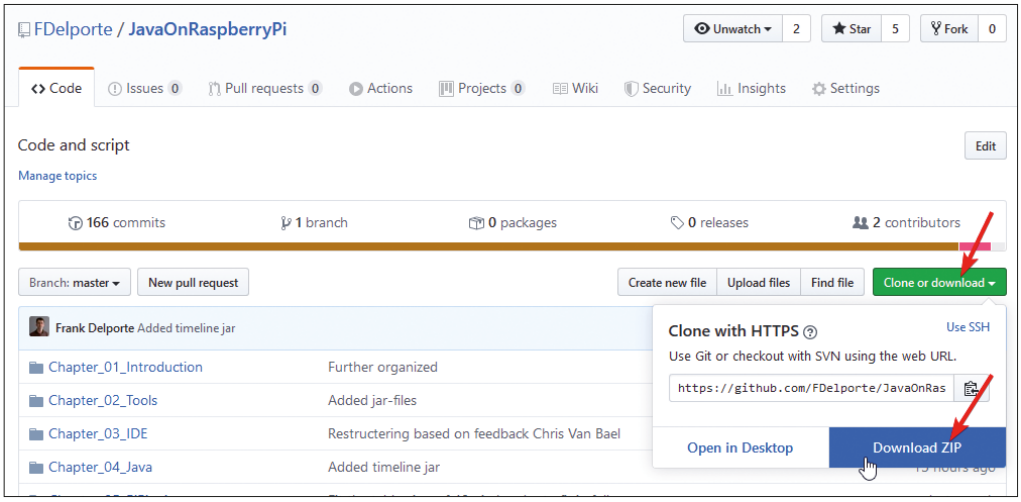
Everything is combined into one GitHub project:

JavaOnRaspberryPi[4]. Keep an eye on this project for any changes and corrections. Changes will be listed in the file CHANGES.md[5].

The sources are structured to match the chapters in this book so you can easily find each example.

Get the sources

If you are new to GitHub (source control systems), just click the download button on the site and you will get a full copy of everything in one zip file.



Get sources from GitHub as ZIP

On the other hand, if you know how to checkout code, please do! It will give you a copy of all the code and scripts but also allow you to make changes and improvements and send them as pull requests if you know how to work with git. **It would be great if the samples could be further improved by you! I'm looking forward to your pull request.**

4 https://github.com/FDelporte/JavaOnRaspberryPi
5 https://github.com/FDelporte/JavaOnRaspberryPi/blob/master/CHANGES.md

Here is how you check out the code into your home directory on the Pi:

```
$ cd /home/pi
$ mkdir Book
$ cd Book
$ git clone https://github.com/FDelporte/JavaOnRaspberryPi.git
```

Inside most of the projects, you'll find a "target" directory with the jar-file which you can run directly if you don't want to modify the code and/or build the project yourself. More info on building and running later in this book…

The styling used in the book

All the code and scripts in this book are formatted like this, where "…" means a portion of the code is skipped in this book, but fully available in the sources.

```
public class HelloWorldExample{
  public static void main(String args[]){
    System.out.println("Hello World !");
    ...
  }
}
```

Command and script examples have the same styling, and when the line starts with a "$" that means it is a command you must type in in the terminal, the following lines are the output you will get, for example, for the command "java -version":

```
$ java -version
openjdk 11-BellSoft 2018-09-25
OpenJDK Runtime Environment (build 11-BellSoft+0)
OpenJDK Server VM (build 11-BellSoft+0, mixed mode)
```

All references to the sources in the GitHub project will look like this, referring to the subdirectory in the GitHub project[6]:

Chapter_01_Introduction > README.md

Tips will be displayed like this.

Don't forget to check the README file!

6 https://github.com/FDelporte/JavaOnRaspberryPi

Read the README!

There is a README file in the root folder, but also in each chapter, sometimes also in the projects. This will help you to easily use the scripts, set up a project, etc.

The README of most chapters also has a "Read more" section where you can find links to more info about the topics of that chapter.

Adding a README file is a good practice, even when you are the only person working on a project! Minimally it needs to contain a title and description, how to work on the project and how to build, test and run it.

Guidelines

A README uses a list of conventions for the formatting, the ones used most in this book:

```
* "#", "##", "###"… for titles
* "*" for bullets
* "[title](link)" for links
* Three single ` before and after a code block
```

Example

This is a short example file:

```
# Chapter 3: Java
Chapter describing the history and current state of Java and how to install on a Pi.

## Links
* [AdoptOpenJDK](https://adoptopenjdk.net/)
* [BellSoft Liberica JDK](https://bell-sw.com/)
* [Duke the Java Mascot, explained](https://jaxenter.com/duke-the-java-mascot-explained-118397.html)
* [SDKMAN](https://sdkman.io/)
* [SDKMAN on Windows](https://ngeor.com/2019/12/07/sdkman-windows.html)

## Scripts
See the scripts directory to install specific Liberica SDK versions on the Pi.

To check the installed java version run this command in the terminal:

```
$ java -version
```
```

Screenshot of the source of a demo readme file

When you look at this file online it will look like this:

Screenshot of the demo readme file as shown on GitHub

What's next?
In the source files you can find an overview of links with interesting articles on how and why to write a good README:

Chapter_01_Introduction > README.md

Thanks to...

…**all open-source contributors and bloggers** who are constantly pushing Java, JavaFX, Pi4J and all those other marvelous tools, frameworks, libraries forward. Without them, we wouldn't be able to produce such easy to build and good-looking applications! Only a limited list is mentioned in this book, but there are a lot more and they are constantly sharing their work and knowledge on the internet. Keep an eye out for them!

…**my colleagues** who are always critical when doing pull requests and sharing their knowledge. I fully agree with the phrase "What one developer can do in one month, can be done by two developers in two months", but two experienced and critical developers will produce better code, than a solo player.

Read the book "The Mythical Man-Month" by Fred Brooks[7] if you want to know why a late software project becomes even later when adding manpower.

Or even better, buy two copies of this book for your manager, so he can read it twice as fast ;-)

7 https://en.wikipedia.org/wiki/The_Mythical_Man-Month

…**Elektor** who triggered me to start writing this book and publish it as a real paper book! A true bucket list thing achieved now.

…my wife and son, who I have neglected too much in the last months. I promised them that would change once this book was finished. It's a pity they know me too well and immediately asked what my next project would be :-)

…**the reviewers, interviewees and everyone who helped me to realize this book!**

Reviews, contributions, tips, feedback, ideas… by Stijn Deroo, Nathalie Delporte, Lieven Baes, Trisha Gee, Vlad Mihalcea, Kasper Zutterman, Ludo Joris, Jan Lamote, Kim Harding, Catherine Van Damme, Chris Van Bael, Mark Heckler, Pieterjan Deconinck, Kasia Pranke, Marian Faydula, Wouter Vanhauwaert, Michael Egger and… you? Please send me your feedback via e-mail on javaonraspberrypi@webtechie.be! You can also send me pull-requests on GitHub if you want to contribute to the examples.

Legal stuff

Index

Symbolen
8x8 LED-matrix 205
1602A 229
5101AS 170

A
ActiveMQ 292
AdoptOpenJDK 66, 69, 77
annotation 246
apt 30
apt-get 30
Arduino 43, 220, 221, 299
assertEquals 200
Azul 67

B
BCM 104
BellSoft 67, 68
bits 163
bit shift register 170
break 83
button 194
byte 163, 165
Byte.toUnsignedInt 167

C
CoderDojo 45
ControlsFX 124
CRUD 265

D
distance sensor 198
dmesg 223
Double 80
DRY 117
DTO 286
Duke 63

E
Eclipse 48
enum 82, 153

F
Firefox 32
Float 80
Fritzing 37
FXML 158
FXRibbon 123

G
GluonHQ 119
GPCLK 99
GPIO 94, 98, 191
GraalVM 119, 122, 126
ground pin 98

H
H2 260
H2-console 270
HC-SR04 198
header 101
Hibernate 275

I
I²C 99, 151
IDE 48
If, Then, Else 81
Impostor Syndrome 291
Integer 80, 165
Integer.toUnsignedString 167
IntelliJ IDEA 48
Inversion Of Control 246

J
JavaFX 74, 119, 309
JDK 65
JEP 75
JPA 260, 275
JRE 65
jshell 73
JSON 220
JsonIgnore 265
JVM 64

K
KISS 15, 117

L
LCD-display 229
LED 40, 131
LED-strip 41, 299
Liberica 67, 68, 72
light sensor 220
Linux commands 28
log4j 113
Long 165
ls 30

M
main 79
Maven 95, 107
MAX7219 205
message queue 292
methods 83
Micronaut 246
MobaXterm 34
module-info 103, 146, 296
Mosquitto 292, 293
MqttCallback 297
MqttClient 297

N
nano 31, 78
NetBeans 48
NIH 117
NullPointerExceptions 77

O
object-oriented programming 84
OpenJFX 121
OpenWeatherMap 231
Oracle 63, 66, 119

P
Package 148
password 33
Pi4J 186, 277
PickerFX 124
Pivotal 246, 251
POGE 117
POJO 246
pom.xml 111, 134
power pin 98
power supply 41
Project Loom 276
Project Panama 68, 126
Project Valhalla 276
pull-down 197
push-button 131
PWM 100, 203
Python 174

Q
Quarkus 246
qubits 163

R
RabbitMQ 292
Raspbian 22, 72
Reactive programming 285
README 18
Relay 151
resistor 38
REST 260, 266
RGB-LED 41, 191
RTFM 118

S
Scanner 88
Scene Builder 160
screensaver 150
SD card 22
SDKMAN 70
segment display 169
serial 220
Serial-In-Parallel-Out 170
Short 165
signed 166
SN74HC595 170
sources 16
SPI 100, 205
Spring 246
Spring Boot 247
Spring Initializr 247
SSH 33
start automatically 149
start-up arguments 79
static 137, 138
STEAM 45

stream 89
Strings 79
Sun Microsystems 63, 119
Swagger 258, 268
switch, case, default 83
Switch Expressions 75, 76
System on a Chip 251
System.out.println 78

T
TDD 118
Text Blocks 76
Thymeleaf 257
TilesFX 107, 123, 126, 131

U
UART 98
Undertow 309
unit test 188
unsigned 166

V
Visual Studio Code 55
VMware 251
VNC 32
VSCodium 56

W
web server 252
Wireshark 321
WiringPi 104, 186
WS2812 43
WS2813 43

Y
YAGNI 117
yEd Graph Editor 37

Getting Started with Java on the Raspberry Pi

Frank Delporte is a software developer and technical lead with more than 25 years of experience in video, multimedia, technical project management, digital signage, and (web) programming. Since starting with a Commodore 64, his interests have evolved to include multiple technologies. Delporte loves to experiment and share his knowledge through blogging (webtechie.be), educating, and writing.

Controlling physical things with some lines of code can seem like magic! Getting Started with Java on the Raspberry Pi provides a thorough introduction to Java as well as instructions for installing it on the Raspberry Pi. You also receive many tips and tricks for improving your skills as a developer. And, above all, the book provides easy-to-understand examples on a wide variety of topics:

- The magic of bits and bytes and solving the confusion of Java signed values with the help of an LED number display.
- Beautiful user interfaces made with JavaFX so you can interact with the hardware.
- Pi4J applications to be able to control different types of hardware like LEDs, buttons, displays, LED strips, relays boards and more.
- Spring applications so you can interact with your RPi via web interfaces.
- How to set up a queue so you can send and receive messages to and from Arduino boards or other RPis.
- And a lot of other inspirational ideas and starter examples that will help you to build your dream do-it-yourself project.

If you are new to Java, you will learn the language bit by bit by following the examples. If you are an experienced Java programmer, you can extend your knowledge and learn to control the world around you with simple and inexpensive components.

ISBN 978-1-907920-91-2

9 781907 920912

Elektor International Media

www.elektor.com

elektor

LEARN DESIGN SHARE