

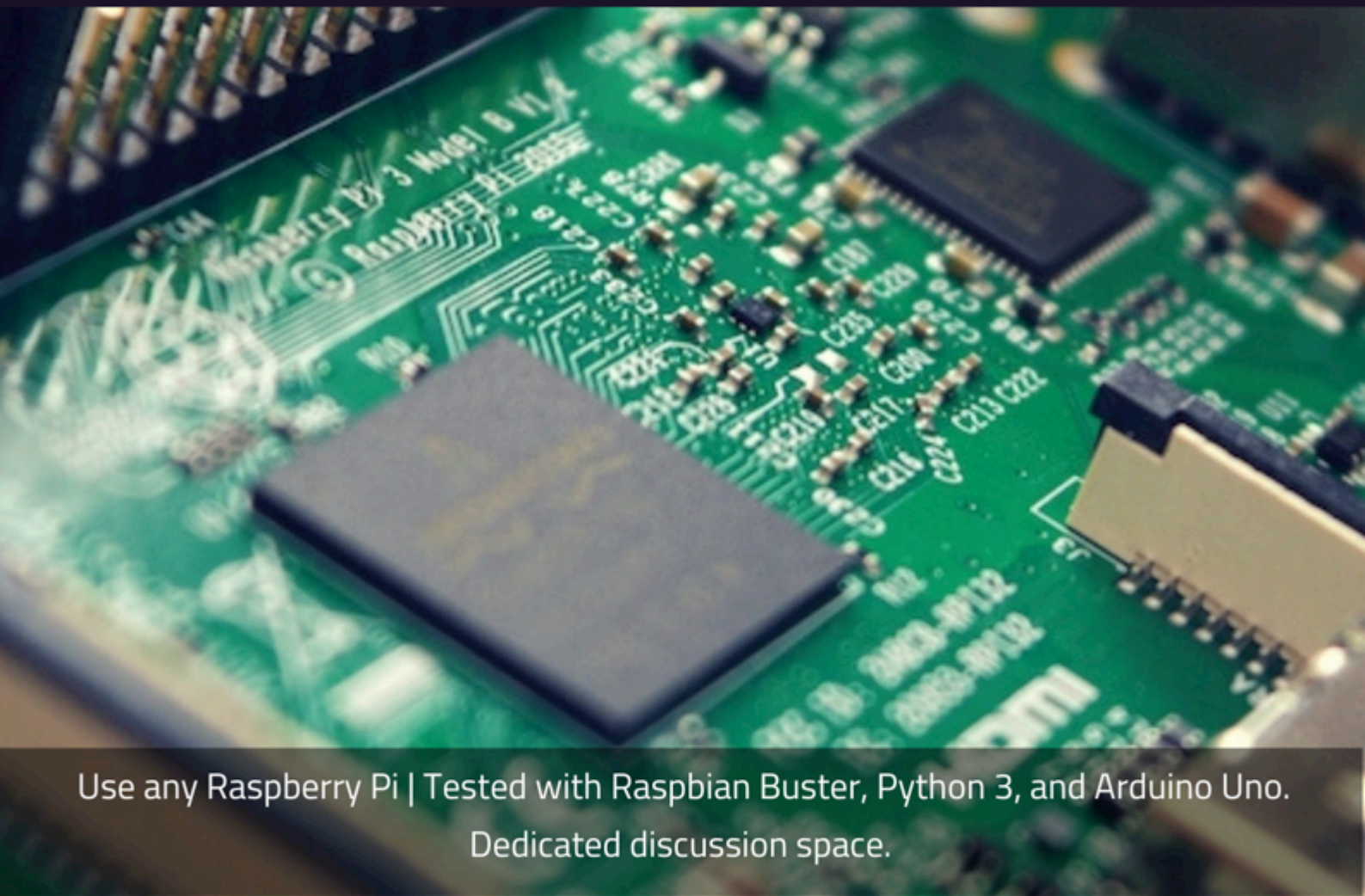
2ND EDITION

# RASPBERRY PI FULL STACK

A COMPREHENSIVE PROJECT COURSE THAT WILL TEACH YOU HOW TO BUILD A MODERN INTERNET-OF-THINGS APPLICATION WITH YOUR RASPBERRY PI AND ARDUINO. LEARN PYTHON AND JAVASCRIPT, SENSORS, BUTTONS & LEDS, IF-THIS-THEN-THAT, GOOGLE SHEETS, GOOGLE CHARTS, PLOTLY, AND COMMUNICATE WITH THE NRF24.

---

DR PETER DALMARIS



Use any Raspberry Pi | Tested with Raspbian Buster, Python 3, and Arduino Uno.  
Dedicated discussion space.

# RASPBERRY PI FULL STACK 0.42

Peter Dalmaris

## **Raspberry Pi Full Stack, 2<sup>nd</sup> Edition**

By Dr Peter Dalmaris

Copyright © 2020 by Tech Explorations<sup>TM</sup>

All rights reserved. This book or any portion thereof may not be reproduced or used in any manner whatsoever without the express written permission of the publisher except for the use of brief quotations in a book review.

Printed in Australia

First Printing: 2020

ISBN: 978-1-63649-322-0

Tech Explorations Publishing  
PO Box 22, Berowra 2081 NSW  
Australia

[www.techexplorations.com](http://www.techexplorations.com)

**Cover designer:** Michelle Dalmaris

### **Disclaimer**

The material in this publication is of the nature of general comment only, and does not represent professional advice. It is not intended to provide specific guidance for particular circumstances and it should not be relied on as the basis for any decision to take action or not take action on any matter which it covers. Readers should obtain professional advice where appropriate, before making any such decision. To the maximum extent permitted by law, the author and publisher disclaim all responsibility and liability to any person, arising directly or indirectly from any person taking or not taking action based on the information in this publication.

Version 0.42

# Table of contents

<b>Part 1: Getting Started with the Raspberry Pi Full Stack</b>	<b>10</b>
Chapter 1 - What is this book about?	11
Chapter 2 - A walk-through the Full Stack project	13
Chapter 3 - Required hardware	18
Chapter 4 - How to get help	23
Chapter 5 - The code repository	25
<b>Part 2: Raspberry Pi, Arduino, and Raspberry Pi Zero W</b>	<b>27</b>
Chapter 6 - Raspberry Pi vs Arduino high level comparison	28
Chapter 7 - Need for efficiency: The Raspberry Pi Zero W	35
Chapter 8 - Need for speed: The Raspberry Pi 4 (and 3)	38
<b>Part 3: How to setup the operating system</b>	<b>40</b>
Chapter 9 - Operating systems for the Raspberry Pi	41
Chapter 10 - What is a 'headless' operating system	45
Chapter 11 - How to download and install Raspbian	49
Chapter 12 - How to setup SSH and Wifi in headless mode	53
Chapter 13 - How to set a hostname	56
Chapter 14 - Boot into Raspbian for the first time	60
Chapter 15 - How to set a fixed IP address	65
Chapter 16 - Basic configuration	67
Chapter 17 - Working as the 'root' user	70
<b>Part 4: How to backup and restore your SD card</b>	<b>74</b>
Chapter 18 - Backup an SD card - Mac OS	75
Chapter 19 - Restore an SD card - Mac OS	78
Chapter 20 - Backup an SD card - Windows	80
Chapter 21 - Restore an SD card - Windows	83
<b>Part 5: Pins, GPIOs and how to control them with Python</b>	<b>85</b>
Chapter 22 - Raspberry Pi pins, roles, and numbers	86
Chapter 23 - A taste of Python on the Command Line Interpreter	90
Chapter 24 - Python Functions	95
Chapter 25 - A simple Python program	101
Chapter 26 - Wire a simple circuit	108
Chapter 27 - Control an LED with GPIOZERO	111

Chapter 28 - Control an LED with rpi.gpio.....	115
Chapter 29 - Read a button with GPIOZERO .....	119
Chapter 30 - Read a button with RPi.GPIO.....	122
Chapter 31 - Control an LED with a button.....	124
Chapter 32 - Setup the DHT22 sensor with Git.....	127
Chapter 33 - Use the DHT22 sensor.....	132
Chapter 34 - DHT with the CircuitPython library.....	136
<b>Part 6: Setup the Web application Stack.....</b>	<b>138</b>
Chapter 35 - The Web Application Stack.....	139
Chapter 36 - The Python Virtual Environment.....	145
Chapter 37 - Increase the disk swap file size.....	149
Chapter 38 - Set up system Python - preparation.....	152
Chapter 39 - Download, compile and install Python 3.....	154
Chapter 40 - Setup the app Python Virtual Environment.....	158
Chapter 41 - Setup Nginx.....	161
Chapter 42 - Setup Flask.....	163
Chapter 43 - A tour of a simple Flask app.....	166
Chapter 44 - UWSGI installation.....	169
Chapter 45 - Nginx configuration.....	172
Chapter 46 - UWSGI configuration.....	176
Chapter 47 - UWSGI and Nginx configuration testing.....	179
Chapter 48 - Configure systemd to auto-start uwsgi.....	182
<b>Part 7: Setup the database.....</b>	<b>185</b>
Chapter 49 - Install the SQLite3 database.....	186
Chapter 50 - Hand-on with the SQLite3 CLI.....	189
<b>Part 8: Styling with Skeleton.....</b>	<b>195</b>
Chapter 51 - Static assets and the Skeleton boilerplate CSS.....	196
Chapter 52 - Setup the static assets directory.....	200
Chapter 53 - Introducing the Skeleton boilerplate CSS.....	203
Chapter 54 - Copying files using SFTP.....	207
Chapter 55 - Flask templates.....	212
Chapter 56 - Debugging a Flask app.....	217
<b>Part 9: Capture and record sensor data.....</b>	<b>221</b>
Chapter 57 - Introduction to Part 9.....	222
Chapter 58 - Install the DHT library and the rpi-gpio module.....	223

Chapter 59 - Install the DHT library and the rpi-gpio module (legacy).....	227
Chapter 60 - Display the current sensor values in the browser.....	232
Chapter 61 - Create a database to store sensor data.....	238
Chapter 62 - Capture sensor data with a Python script.....	240
Chapter 63 - Schedule sensor readings with cron.....	244
Chapter 64 - Update the application file and template file.....	246
<b>Part 10: Implement the date range selection feature.....</b>	<b>252</b>
Chapter 65 - Introduction to Part 10.....	253
Chapter 66 - Prototype datetime range of records in SQLite CLI.....	256
Chapter 67 - Prototype datetime range in the browser.....	259
Chapter 68 - URL querystring validation.....	265
Chapter 69 - Quick tidying up.....	270
Chapter 70 - Use radio buttons for easy timedate range selection.....	275
Chapter 71 - Provision the Python script to work with the radio buttons.....	279
<b>Part 11: Google Charts and Datetime widgets.....</b>	<b>283</b>
Chapter 72 - Introduction to Part 11.....	284
Chapter 73 - Implement Google Charts.....	288
Chapter 74 - Test Google Charts.....	295
Chapter 75 - The datetime picker widget.....	297
Chapter 76 - Implement the datetime picker widget.....	300
Chapter 77 - Test the datetime picker widget.....	303
<b>Part 12: Dealing with time zones.....</b>	<b>306</b>
Chapter 78 - Adjust datetimes to local time zone on the client side.....	307
Chapter 79 - Introduction to Arrow.....	312
Chapter 80 - Implement Arrow.....	316
Chapter 81 - Upload timezone changes and test.....	319
Chapter 82 - Link the two pages of the application.....	322
<b>Part 13: Charting with Plotly.....</b>	<b>326</b>
Chapter 83 - What is Plotly and how to install it.....	327
Chapter 84 - Try out Plotly on the CLI.....	331
Chapter 85 - Implement Plotly support on the client side.....	335
Chapter 86 - Add Plotly support on the server side.....	341
Chapter 87 - How to debug Javascript.....	346
Chapter 88 - Server side debugging example.....	352
<b>Part 14: Access your application from the Internet.....</b>	<b>356</b>

Chapter 89 - How to access your application from the Internet?	357
Chapter 90 - Set a static IP address	360
Chapter 91 - Expose your app to the Internet with port forwarding	363
Chapter 92 - Create a self-signed certificate for application	369
Chapter 93 - Edit Nginx configuration to use SSL	376
Chapter 94 - Test SSL in Firefox, Safari, Chrome	381
<b>Part 15: Data-logging with Google Sheet</b>	<b>386</b>
Chapter 95 - What is datalogging, and why Google Sheet?	387
Chapter 96 - Setup Google API credentials	390
Chapter 97 - Setup the Python libraries and Google Sheet	402
Chapter 98 - Implement of Google Sheet data-logging	408
<b>Part 16: Setup a remote Arduino Sensor node with the nRF24</b>	<b>411</b>
Chapter 99 - Why setup an Arduino remote node?	412
Chapter 100 - The Arduino node wiring	414
Chapter 101 - The Arduino node sketch	418
Chapter 102 - Raspberry Pi and nRF24 wiring	423
Chapter 103 - The Raspberry Pi nRF24 receiver script	426
Chapter 104 - How to install the Python nRF24 modules on the Raspberry Pi	433
Chapter 105 - Test the nRF24 communications	439
Chapter 106 - Modify the front end of the application to show remote node data	442
<b>Part 17: If This Than That alerts</b>	<b>447</b>
Chapter 107 - An introduction to If This Then That	448
Chapter 108 - Create an IFTTT webhook and applet	450
Chapter 109 - Add IFTTT code in the application and testing	460
Chapter 110 - Install the node listener script as a systemd service	464
<b>Part 18: Wrapping up</b>	<b>467</b>
Chapter 111 - Make lab_env_db page update every 10 minutes	468
Chapter 112 - Recap and what's next	470
<b>Part 19: Project extension: Text messaging using Twilio</b>	<b>472</b>
Chapter 113 - What is this project extension all about?	473
Chapter 114 - An introduction to Twilio	476
Chapter 115 - Setup Twilio account	478
Chapter 116 - Create a useful bash shell script	484

Chapter 117 - Add Twilio support to Raspberry Pi.....	488
Chapter 118 - Install Twilio CLI.....	491
Chapter 119 - Create local and public DNS hostnames.....	496
Chapter 120 - Create trusted SSL/TLS certificate.....	502
Chapter 121 - Send text alert messages.....	513
Chapter 122 - Receive text message commands.....	517



# Chapter 34

## DHT with the CircuitPython library

In this chapter I will show you how to use the DHT sensor with the newer DHT library from Adafruit that is part of the CircuitPython project<sup>9</sup>. CircuitPython is the Python programming language specifically adapted for using with microcontrollers and, of course, the Raspberry Pi. CircuitPython is a fork of MicroPython<sup>10</sup>, and implementation of the Python language for microcontrollers such as the pyboard<sup>11</sup> and the ESP32<sup>12</sup>.

To install the CircuitPython DHT module, you can use apt-get, instead of Git. The module requires the libgpiod C library for its operation, so let's install it first.

I assume that you are logged in to your Raspberry Pi as the “pi” user.

At the prompt, type this:

```
$ sudo apt-get install libgpiod-dev
```

Accept the installation, and wait for a few seconds.

When the prompt comes back, use pip3 to install the DHT library (in the output below I have omitted most of the text and only kept the first and last few lines):

```
$ pip3 install adafruit-circuitpython-dht
```

```
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
```

---

9. Learn more about CircuitPython: <https://circuitpython.org/>

10. Learn more about MicroPython: <https://store.micropython.org/>

11. See the family of pyboard microcontrollers: <https://store.micropython.org/>

12. Quick reference for the ESP32 implementation of microPython: <http://docs.micropython.org/en/latest/esp32/quickref.html>

```
Collecting adafruit-circuitpython-dht
  Downloading https://www.piwheels.org/simple/adafruit-circuitpython-dht/
adafruit_circuitpython_dht-3.5.1-py3-none-any.whl
...
Successfully installed Adafruit-Blinka-5.4.0 Adafruit-PlatformDetect-2.17.0 Adafruit-
PureIO-1.1.5 adafruit-circuitpython-dht-3.5.1 pyftdi-0.51.2 pyusb-1.1.0 rpi-ws281x-4.2.4 sysv-
ipc-1.0.1
```

The CircuitPython DHT library is now installed, and you can use it in your Python programs. Let's try it out in the Python Command Line Interface. Type "python3" to start the CLI, and copy the program as you see it below (in bold is the text to enter):

```
$ python3
Python 3.7.3 (default, Apr 3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import board
>>> import adafruit_dht
>>> dhtDevice = adafruit_dht.DHT22(board.D17)
>>> temperature_c = dhtDevice.temperature
>>> temperature_f = temperature_c * (9 / 5) + 32
>>> humidity = dhtDevice.humidity
>>> print("Temp: {:.1f} F / {:.1f} C  Humidity: {}% ".format(temperature_f, temperature_c,
humidity))
Temp: 91.0 F / 32.8 C  Humidity: 25.8%
>>> exit()
```

The program is just as simple as the one you used in the previous chapter. You must import "adafruit\_dht", and then create the DHT object by calling "adafruit\_dht.DHT22". The only parameter to pass is the data PIN number (17). To get the temperature, call "temperature" on the DHT object. You can get the humidity by calling "humidity".

In later parts of this project, you can use the CircuitPython DHT library as a drop-in replacement of the earlier version of the library.