elektor books

Control Your Home with Raspberry Pi

Secure, Modular, Open-Source and Self-Sufficient

Koen Vervloesem

elektor
design > share > sell

Control Your Home with Raspberry Pi

Koen Vervloesem

an Elektor Publication

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

First published in the United Kingdom 2020

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

● **ISBN 978-1-907920-94-3**

● **EISBN 978-3-89576-383-0**

● **EPUB 978-3-89576-382-3**

Prepress production: DMC ¦ daverid.com
Printed in the Netherlands by Wilco

Images and logos used in this book are courtesy of Material Design Icons, Cisco, and the Raspberry Pi Foundation

Table of Contents

• Preface

Ever since the Raspberry Pi was introduced, the popular single-board computer has been used by enthusiasts to automate their home. That's not a coincidence: the Raspberry Pi is a powerful computer in a small package, with lots of interfacing options to control various devices.

In this book, I'll show you how you can automate your home with a Raspberry Pi. You can do this in many ways and with various software and hardware choices. I'll show you one way, which is a bit different from what you'll read in many other books, but my approach has its merits, and I'll explain why.

You'll learn how to use various wireless protocols for home automation, such as Bluetooth, 433.92 MHz radio waves, Z-Wave, and Zigbee. Soon you'll automate your home with Python, Node-RED, and Home Assistant, and you'll even be able to speak to your home automation system. All of this in a secure way, with a modular system, completely open-source and without relying on third-party services.

At the end of the book, you can install and configure your Raspberry Pi as a highly flexible home automation gateway for your protocols of choice and link various services with MQTT to make it a system of your own. This DIY (do it yourself) approach is a bit more laborious than just installing an off-the-shelf home automation system, but in the process, you learn a lot, and in the end, know exactly what's running your house and how to tweak it. And that's why you were interested in the Raspberry Pi in the first place, right?

Koen Vervloesem, May 2020

Chapter 1 • Introduction

In this introductory chapter, I give a short overview of what home automation is and why you would use a Raspberry Pi as a home automation gateway. Then I describe what I consider properties of a 'good' home automation system:

- secure
- modular
- open-source
- self-sufficient

In the rest of this book, I'll explain step by step how to create such a good home automation system with a Raspberry Pi.

1.1 • What is home automation?

Home automation is the process or result of automating systems that are running at home: lighting, HVAC (heating, ventilation, and air conditioning), appliances such as washing machines, blinds, and roller shutters, and so on. A home automation system is also able to use information from environmental sensors (temperature, humidity, pressure, CO2, …), smart meters, movement sensors, presence sensors, cameras, and so on.

A home automation system typically consists of:

- a central gateway (also called controller or hub), which controls devices and reads sensor measurements
- controllable devices
- sensors

The controllable devices and sensors are regularly called "smart devices", although almost none of them are really smart. Another name you'll see for them is IoT (Internet of Things) devices because they can be (directly or indirectly) linked to and controlled over the internet.

Figure 1.1 A home automation system consists of a central gateway and various controllable devices and sensors.

A home automation gateway always has a user interface. Of course, the purpose of home automation is to automate as much as possible, so the idea is that the user shouldn't have to use this user interface that much. But a user interface is still essential to:

- configure the home automation gateway: for instance if the sun goes down, close the blinds;
- manually control your devices: this should still be possible because you can't automate everything;
- show you a nice dashboard of sensor measurements: for instance to see the inside and outside temperature.

This user interface can come in many forms:

- Most home automation gateways have a web server running, which supplies a web interface as the user interface. You can access this web interface on any computer or mobile device.
- Some systems have a mobile app for Android or iOS, which is generally better adapted to the specific requirements of mobile devices, such as a smaller screen.
- It's also possible to use a dedicated touch screen, for instance hanging on the wall, to control your home automation system, and to show you some nice graphs.[1]
- Last but not least, in recent years home automation systems have added a new kind of user interface: speech. With a so-called voice assistant or smart assistant (again, they are not that smart), you can give speech commands to your home automation system and it replies with spoken messages.

This book is not focused on any of these user interfaces; it's more about the backend services and how to link and automate them. However, I cover two home automation projects with a web interface in Chapter 10 (Home Assistant and Node-RED), and create a voice assistant for your home automation system with Rhasspy (Chapter 12). You should consult the documentation of these projects if you're more interested in the user interface side of home automation.

1.2 ● Why use a Raspberry Pi as a home automation gateway?

If you buy an off-the-shelf home automation system, the gateway is a small box that looks somewhat like a router or a Wi-Fi access point. In this book, we'll show you how you can create your own home automation gateway with a Raspberry Pi.

But why would you do that? Because you can, of course! More seriously, the Raspberry Pi is what makes the approach in this book possible. We'll go into the advantages of this approach in the next section, but the number one reason to use a Raspberry Pi as your home automation gateway is: you are in control.

An off-the-shelf home automation gateway isn't flexible: you can only do with it what the

1 In many cases this touch screen is just running a fullscreen web browser visiting the home automation gateway's embedded web server.

manufacturer allows, you rely on the manufacturer's goodwill to receive updates and new functionality, and most of the time you can't "hack" on it yourself.[2]

Contrast this with the Raspberry Pi. You can choose your operating system (which we'll do in the next chapter), you can choose which communication protocols you'll want to support (which we'll cover in various chapters in this book), you can choose your user interface, and so on. You can even choose the case to protect your Raspberry Pi and which expansion boards you connect, as there's a whole ecosystem of hardware for the Raspberry Pi.

Of course, you can also run home automation software on a more powerful system, such as a NAS (network-attached storage) or a home server. But the Raspberry Pi has several advantages to these systems:

- It's very low-power, so it doesn't cost you much to keep it running 24/7.
- For most home automation tasks you don't need the processing power that these other systems offer.
- The ecosystem of software and hardware for the Raspberry Pi is immense, as well as the number of resources where you find more information about it.[3] This is also a reason to choose the Raspberry Pi over similar single-board computers from other manufacturers.

1.3 ● The properties of a good home automation system

A good home automation system should:

- be secure, so you don't risk someone else controlling your house or spying on you at home;
- be modular, to make it easy to plug in other protocols or applications;
- only use open-source software;
- be self-sufficient, not relying on cloud systems from Google, Amazon, or other parties.

This is my highly opinionated vision, and it's this vision that I build upon in this book.
If you consider these properties for a moment, you'll see that this vision is almost diametrically opposed to most off-the-shelf systems you'll find. I'll give some examples in the next subsections.

It's possible that you don't agree with some of these properties, or that you don't have such strong feelings about them as I do. That's OK: while I explain an approach in this book

2 With hacking I don't mean gaining unauthorized access to a computer (which is the connotation the word unfortunately has received). The Jargon File describes a hacker as "a person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary." (The Jargon File, http://www.catb.org/jargon/html/H/hacker.html)

3 For instance, the Raspberry Pi Foundation publishes its official magazine about the Raspberry Pi, MagPi (https://magpi.raspberrypi.org), and Elektor (the publisher of this book) publishes Dutch (https://www.magpi.nl) and French (https://www.magpi.fr) editions of the magazine.

that implements this vision, thanks to its modularity you can certainly plug in proprietary software or cloud systems if you prefer these. Heck, you can even add monolithic and insecure software, but I don't tell you how.

In the next subsections, I'll go over these four properties in more detail, and I hope that at the end of this chapter you'll agree with me that this approach to home automation is the right one.

1.3.1 ● Secure

Of course, a home automation system should be secure. No one can be against it, can't they? A home automation system controls your home, so whoever can break into it can make your life very miserable.

Unfortunately, even if a manufacturer tells you that his system is secure, chances are that it isn't. Security is very difficult to attain, and most manufacturers don't want to spend the resources needed to secure their system.[4]

Home automation and IoT devices are notoriously insecure. At the Usenix Security Conference 2019, the Czech security software company Avast and Stanford University presented their research of household IoT devices. Avast scanned 83 million IoT devices in 16 million homes around the world of people who agreed to share these data. The results of the study published in "All Things Considered: An Analysis of IoT Devices on Home Networks" (https://press.avast.com/hubfs/stanford_avast_state_of_iot.pdf) were staggering:

- 7% percent of all IoT devices support an obsolete, insecure, and completely unencrypted protocol such as Telnet or FTP.
- Of these, 17% exhibit weak FTP passwords, and 2% have weak Telnet passwords.
- Surveillance cameras have the weakest Telnet profile, with more than 10% of them that support Telnet with weak credentials.
- 3% percent of the homes are externally visible on the internet and more than half of those have a known vulnerability or a weak password.

This is not an isolated study. Not a week goes by without some news items about insecure devices, most of the time because basic security measures such as strong passwords are not enforced by the manufacturer or basic programming errors have been made. To give you an idea about what can happen: in 2018 nude videos of the Dutch women's handball team appeared on a popular porn website because the surveillance cameras of the dressing room of a sauna were broken into. Imagine if someone can access your baby monitor with a camera or your security camera in your living room or bedroom…

So what can you do to secure your home automation system? If you choose an off-the-shelf system: not much. You fully rely on the manufacturer's ability to create a secure

4 Most consumers probably wouldn't want to pay more for a secure home automation system anyway.

system and the goodwill to keep supplying patches that solve security issues that have been discovered. And the home automation and IoT industries have clearly shown they are not up to the task. This is one of the reasons why I prefer open-source software. Not because it is always secure, but because the transparency of the open-source development process forces developers to create more secure software.

Security is such an important property of a home automation system that I dedicate an entire chapter in this book about it. It's such a vast topic that entire books are written about it, and I encourage you to read much more about computer security than I can tell you here. In Chapter 3 I'll cover the most important tools you need to secure your home automation system, so you don't need to be paranoid and continuously think about the possibility that someone is currently spying on you.

1.3.2 ● Modular

There are many competing standards and communication protocols for home automation, such as Z-Wave, Zigbee, and KNX. Other protocols aren't specific to home automation but are very usable in this domain too, such as Wi-Fi, Bluetooth, or Near Field Communication (NFC).

Unfortunately, many off-the-shelf home automation gateways support only a small subset of these protocols or even use a proprietary protocol that locks you into using devices of the same manufacturer. That severely limits your choice of products.

You can't know which protocols will become popular in a few years, and maybe you like one product that uses Z-Wave and another product that uses Zigbee. It should be easy to interconnect these devices, even when they use different protocols.

This is why a good home automation system should be modular, which makes it possible to plug in new components when you want to support a new protocol, add a new user interface or extend its functionality in another way.

Many of the wireless communication protocols for home automation need a dedicated transceiver because they work on a specific radio frequency. That's where the Raspberry Pi shines: you can easily connect Z-Wave, Zigbee, or 433.92 MHz transceivers using the USB ports or the GPIO header. So you can start with a basic Raspberry Pi setup supporting only IoT devices that are communicating over Wi-Fi and Bluetooth, add an RTL-SDR USB dongle to read the measurements of your 433.92 MHz weather sensors, later add a Z-Wave HAT on the board when you start adding Z-Wave sensors to your house and then add a Zigbee USB transceiver when you want to control some Zigbee lights.

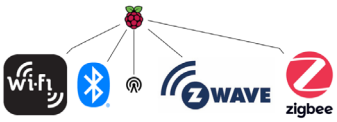

Figure 1.2 A good home automation system is modular enough to support many home automation protocols.

Modularity is also important for software. There's a lot of user-friendly software to make your Raspberry Pi a home automation gateway.[5] So you just install this software on your Raspberry Pi and that's it: you have a gateway that supports several devices. Some of these systems are very modular and extensible, others aren't. Many of them support MQTT (Message Queuing Telemetry Transport), a common language to exchange messages.

MQTT has become the standard for interoperability between various home automation devices. For instance, if your home automation gateway of choice doesn't support Zigbee but it does support MQTT, then you only have to run the Zigbee2mqtt software (see Chapter 9), which translates the Zigbee protocol to MQTT messages. Your gateway can then talk to your Zigbee devices using MQTT.

Modularity also means that you don't have to have one gateway. You can perfectly have your main gateway in your basement, but install a second gateway with your 433.92 MHz receiver for your environmental sensors in your living room because that gives you better coverage to receive data from these wireless sensors. If you're using MQTT, that's very simple to implement: you just relay the sensor readings that your gateway in the living room receives to your MQTT broker, after which your main gateway receives the readings in the MQTT format.

In short: a good modular home automation system means that you can mix and match the devices that you like, irrespective of their protocol, and you can use the software and hardware components of your choice, in various locations in your house.

1.3.3 ● Open-Source

Source code is code written in a human-readable programming language, that specifies the actions a computer has to perform. The source code of a program is then compiled to machine code that the computer can execute, or it's interpreted on the fly to machine code and thus immediately executed by the computer.

Most software is being distributed as machine code, so it's not readable for us. If you buy an off-the-shelf home automation gateway, you generally don't get access to its source code, so you cannot peek into it to see what it does or to assess its quality. You just have to believe the manufacturer on his word. Is that enough for you if it's about software that will get to know you intimately because it processes sensor readings and even camera images about you in your home? Not for me.

But there's a type of software where you do get access to the source code: free and open-source software (sometimes abbreviated as FOSS or even FLOSS).[6] If you really want to be precise, there's free software and open-source software, but for the end-user, the differences are minimal and mostly philosophical. When I talk about "open-source

5 An example is Home Assistant, which I will introduce in Chapter 10.

6 The L in FLOSS stands for "libre", which is kind of a synonym to "free" but making it clearer that it's about maintaining the user's civil liberties: "free" as in "free speech", not as in "free beer", as they say.

software" in this book, I mean free and open-source software.

But what when you're not a programmer and don't even understand the source code of your home automation system? Even then the use of open-source software has a lot of advantages. It's not because you don't have the programming experience that others can't help. Most open-source projects have a decentralized software-development model that encourages open collaboration.

So if you find a bug in the software, or see something wrong in its source code but don't have the programming experience needed to fix it, just report the bug on the issue tracker of the project, and hopefully, someone else in the project's community will step in and fix it. It all depends on the health of the project's community. But a good open-source project has a vibrant community of developers and users who collaborate to continuously improve the software.

In the process of writing this book, I participated in various communities of the programs I covered. I opened issues to report bugs, fixed some bugs, added support for new devices, contributed documentation, and helped people build their software for Raspberry Pi. This was all only possible because they are open-source.

And open-source doesn't just mean getting access to the source code, it's much more than that. If you want to get an idea about what open-source is, I recommend you to read the Open Source Definition on https://opensource.org by the Open Source Initiative. For instance, with open-source software you are not only able to read its source code, you are also allowed to modify it and distribute your modifications.

Even more, when you know a specific software project is open-source, you know that it doesn't arbitrarily restrict what you can do with it. The Open Source Definition even explicitly lists that the license of open-source software must not discriminate against any person or group of persons, nor restrict anyone from making use of the program in a specific field of endeavor.

Open-source is a complex and nuanced topic, and I recommend you to read about it more if you're not accustomed to it. The decentralization and transparency of the open-source development model gives power back to the users, where it belongs. For home automation that's even more important. I don't want a company having control over my house. That's why you'll see that all software in this book is open-source.

1.3.4 ● Self-sufficient

During the last ten years, there has been a worrying development in the computer industry: we all depend more and more on (centralized) cloud systems. Unfortunately, the home automation industry didn't escape this fate. Many popular home automation and IoT systems depend on a cloud server. Some examples:

- the Ring video doorbell with Wi-Fi camera;

- the Nest Learning Thermostat;
- so-called 'smart speakers' running voice assistants, like Amazon Echo and Google Home;
- the IFTTT service that links various other services.

This isn't without its problems. In the last couple of years a couple of cloud services for home automation stopped working for their users:

Revolv Hub
In 2014 Nest bought the company that was selling the Revolv Hub home automation system, not long after Nest itself was acquired by Google. In 2016 Nest shut down the servers Revolv Hub depended on, after announcing it with a quiet note on the website of Revolv a few months earlier. That meant that the $300 Revolv Hub ceased functioning entirely.

Best Buy Insignia devices
At the end of 2019, Best Buy announced that several of their Insignia-branded smart devices would stop working because they decided to shut down the corresponding backend systems.

Wink devices
In May 2020 Wink (with the catchphrase "A simpler way to a smarter home") announced with just a week's notice that it would start charging a monthly fee for the use of their services. Users that didn't want to pay were no longer be able to access their Wink devices and their automations were disabled. The Wink Hub, that had been in stores with the clear description "no required monthly fees, ever", was rendered useless. Ironically, the announcement ended with the message "Our user community is integral to Wink, and we want to continue to be your trusted smart home provider."

Now imagine when the highly popular IFTTT would stop their service: suddenly the automations of millions of people would stop working.

It also just makes no sense to use cloud services to automate your home. Home automation comes down to: you want one device to be able to respond to another device in your home. For instance: the motion sensor in your bathroom detects motion at night, and this turns the bathroom light on for five minutes. If you use IFTTT to link both devices, the motion sensor has to send a message to the internet, IFTTT relays the message to your bathroom light (for instance a Philips Hue light), and the bathroom light turns on.

But there's no need for an internet service like IFTTT between both devices because they both are in your home, so it makes much more sense to link them locally, using a server at home. This could be a Raspberry Pi running home automation software that doesn't need a cloud server to function, but does all its processing on-device (or on the edge, as it's called now). You can perfectly do this with Node-RED or Home Assistant (see Chapter 10). Then there's no way a company can render your home automation system useless by shutting down their service, or your bathroom light doesn't turn on at night because your internet

connection or IFTTT's servers are down. You're fully in control of your home automation system.

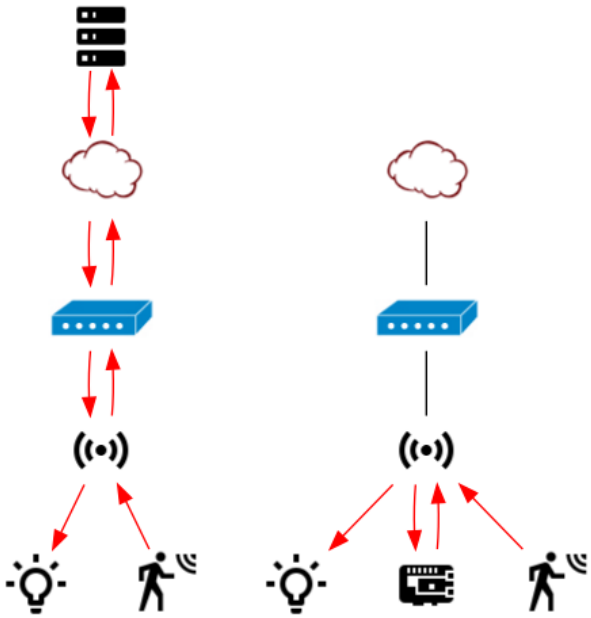

Figure 1.3 The home automation at the left is cloud-based: a simple motion detection message first goes to a server over the internet before returning to your light. The self-hosted system on the right makes much more sense: a Raspberry Pi on your network relays the message without using the internet detour.

Using cloud services for your home automation system has another risk: it invades your privacy. Look at some of the privacy issues with the services I talked about above:

- If you use the Ring video doorbell, it sends a video of everyone that steps on your porch to the manufacturer. The Ring company (which has been bought by Amazon in 2018) has a questionable approach to privacy: in January 2019 it was uncovered that employees have access to the video recordings of all Ring devices and even that the data are stored unencrypted.
- If you use the Nest Learning Thermostat, Google knows precisely when you are home and when you aren't.
- If you run a smart speaker like the Amazon Echo, what you tell your house members will be sent countless times inadvertently to Amazon because the Echo thinks it has heard its wake word.[7] Moreover, Amazon's employees listen to a part of all 'conversations' with the Echo to improve its algorithms.
- If you use the IFTTT service to link your various other services, you give one company

7 In 2020, a team of researchers at Northeastern University and Imperial College London simulated real-world conditions by playing popular TV shows. They found that a variety of smart speakers would activate by mistake up to 19 times each day on average. The study can be found here: https://moniotrlab.ccis.neu.edu/smart-speakers-study/.

access to all your home automation services, which is too much power concentrated in one company's hands: they can see exactly what you're doing.

1.4 ● How to use this book

This book describes a lot of components to support various home automation and IoT protocols. I don't expect that everyone will want to support all these protocols by installing these components manually. Home automation platforms such as Home Assistant (described in Chapter 10) have a fairly complete support for these and many more protocols. You can use such a platform to turn your Raspberry Pi into a home automation gateway.

However, I'm also a big believer in choice. You will have your preferences. Perhaps:

- you don't like these user-friendly home automation platforms.
- you do like one of these platforms, but it doesn't support one of the protocols.
- one of these platforms does support the protocol, but only with a local transceiver and you need a transceiver in another location for better coverage.

In all of these cases, you can use one of the open-source projects in this book to link this protocol to your gateway.

You can look at this book as a DIY manual to create your home automation gateway from scratch. But it's also describing a software architecture for a secure, modular, open-source, and self-sufficient home automation system. It's up to you to choose the components that implement this architecture in your own house, and if a home automation platform such as Home Assistant or Mozilla IoT WebThings Gateway does the job for you, that's fine. They don't lock you into their way of doing things, so you could perfectly use them and still link them to other systems thanks to MQTT and other protocols.

This book expects some familiarity with Raspberry Pi OS (formerly called Raspbian) or Linux in general. I explain most commands the first time I use them, but if you have never worked with a Linux system, I recommend reading an introductory text about Linux, Debian, Raspberry Pi OS, or bash (the Linux command line used in this book).

In various chapters, I show short Python programs that interact with your home automation system. The Python code in this book is not that advanced. If you don't know Python, you should be able to understand what the code does, and maybe you will even be able to adapt it because Python is known for its clarity. However, if you want to make the most out of this book, I do recommend you to learn some Python. The official Python documentation (https://docs.python.org/3/), especially the Python tutorial (https://docs.python.org/3/tutorial/index.html), is a good way to start. A home automation system is typically something very personal, and being able to program it is the best way to customize[8] it to your taste.

8 All Python code in this book has been developed for and tested on Python 3. Its predecessor, Python 2, has been retired on January 1 2020 (https://pythonclock.org), and shouldn’t be used anymore.

Note:
This book is not about how you connect sensors, relays, and so on directly to your Raspberry Pi using the GPIO header, but it strictly covers how you use your Raspberry Pi as a home automation gateway, collecting data and controlling devices remotely using radio and network protocols. Of course, it's perfectly possible to let your Raspberry Pi combine both tasks, but I wouldn't recommend it because it could lead to stability problems, and your gateway should be as reliable as possible.

Here's a short overview of what I'll cover in this book:

Chapter 1: Introduction
The theoretical foundation for this book, with my vision of what good home automation should look like and why you should use a Raspberry Pi for it.

Chapter 2: The Raspberry Pi as a home automation gateway
The practical foundation for this book, where you prepare your Raspberry Pi for its task as a home automation gateway.

Chapter 3: Secure your home automation system
Some general computer security principles and specific instructions to keep your home automation gateway secure, including encryption of all network traffic.

Chapter 4: MQTT (Message Queuing Telemetry Transport)
The lightweight network protocol that is at the center of this book, including the installation of an MQTT broker with encryption and authentication of all messages and the exploration of some MQTT clients.

Chapter 5: TCP/IP
Some TCP/IP-based protocols that everyone can use for home automation without the need for specialized transceivers, such as Wake-on-LAN, SSH, SNMP, HTTP/REST, and video surveillance systems.

Chapter 6: Bluetooth
An introduction to Bluetooth Low Energy, including a way to investigate Bluetooth Low Energy devices and using them to read sensor data and for presence detection.

Chapter 7: 433.92 MHz
The use of the RTL-SDR dongle to receive sensor measurements from devices transmitting on the popular 433.92 MHz frequency.

Chapter 8: Z-Wave
An introduction to the Z-Wave mesh protocol for wireless home automation.

Chapter 9: Zigbee
An introduction to the Zigbee mesh protocol for wireless home automation made popular

by Philips Hue and IKEA TRÅDFRI products.

Chapter 10: Automating your home
Complete home automation platforms such as Node-RED, Home Assistant, and AppDaemon, including dashboards for your home automation gateway.

Chapter 11: Notifications
Email and push notifications to warn you about events in your home, including an easy way to create notifications on receiving specific MQTT messages.

Chapter 12: Voice control
Voice control for your home automation gateway, without depending on online servers.

Chapter 13: Remote access
An overview of ways to remotely access your home automation gateway, with specific instructions about how to create a VPN.

Chapter 14: Conclusion
A wrap-up of this book, with a dashboard for all the discussed services.

Appendix
Some specialized tips that could come in handy in various situations.

Note:
Code examples from this book are published on https://github.com/koenvervloesem/raspberry-pi-home-automation. They can be copied from the GitHub repository one-by-one when trying the various applications in this book. You can also download them all at once. Refer to the instructions in the GitHub repository for more information.

1.5 ● Summary and further exploration

In this introductory chapter, I gave a theoretical foundation for this book, with my vision of what good home automation should look like and why you should use a Raspberry Pi for it. I argued why your home automation system should be:

- secure
- modular
- open-source
- self-sufficient

I hope the examples I gave you have convinced you that these are important properties of your home automation system. In the rest of this book, I show you how you create such a system with a Raspberry Pi.

If you want to read more about some issues I raised in this chapter, I can recommend

the special edition of Mozilla's Internet Health Report of November 2019, called "*Privacy Included: Rethinking the Smart Home" (https://foundation.mozilla.org/en/privacy-included/). It talks about the so-called "smart home" from a more general point of view and is much in line with the approach I advocate in this book. The report stresses the importance of privacy, security, interoperability, and sustainability for smart home devices.

Note:
The use of the ⏎ symbol denotes that code should be typed contiguously on the same line.

• Index

elektorbooks

Control Your Home with Raspberry Pi

Secure, Modular, Open-Source and Self-Sufficient

Ever since the Raspberry Pi was introduced, it has been used by enthusiasts to automate their homes. The Raspberry Pi is a powerful computer in a small package, with lots of interfacing options to control various devices. This book shows you how you can automate your home with a Raspberry Pi. You'll learn how to use various wireless protocols for home automation, such as Bluetooth, 433.92 MHz radio waves, Z-Wave, and Zigbee. Soon you'll automate your home with Python, Node-RED, and Home Assistant, and you'll even be able to speak to your home automation system. All this is done securely, with a modular system, completely open-source, without relying on third-party services. You're in control of your home, and no one else.

At the end of this book, you can install and configure your Raspberry Pi as a highly flexible home automation gateway for protocols of your choice, and link various services with MQTT to make it your own system. This DIY (do it yourself) approach is a bit more laborious than just installing an off-the-shelf home automation system, but in the process, you can learn a lot, and in the end, you know exactly what's running your house and how to tweak it. This is why you were interested in the Raspberry Pi in the first place, right?

- Turn your Raspberry Pi into a reliable gateway for various home automation protocols.
- Make your home automation setup reproducible with Docker Compose.
- Secure all your network communication with TLS.
- Create a video surveillance system for your home.
- Automate your home with Python, Node-RED, Home Assistant and AppDaemon.
- Securely access your home automation dashboard from remote locations.
- Use fully offline voice commands in your own language.

Koen Vervloesem has been writing for over 20 years on Linux, open-source software, security, home automation, AI, and programming. He holds a Master's degree in Computer Science Engineering, a Master's degree in Philosophy and an LPIC-3 303 Security certificate. He is editor-in-chief of the Dutch MagPi magazine and is a board member of the Belgian privacy activist organization, the Ministry of Privacy.

Elektor International Media BV
www.elektor.com

ISBN 978-1-907920-94-3
9 781907 920943

elektor
design > share > sell