Practical IoT Handbook

Programming IoT by implementing hands-on projects with Arduino, Python, and Raspberry Pi

Rodrigo J Hernandez

www.bpbonline.com

First Edition 2025

ISBN: 978-93-65892-659

LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY

The information contained in this book is true and correct to the best of author's and publisher's knowledge. The author has made every effort to ensure the accuracy of these publications, but the publisher cannot be held responsible for any loss or damage arising from any information in this book.

All trademarks referred to in the book are acknowledged as properties of their respective owners but BPB Publications cannot guarantee the accuracy of this information.

www.bpbonline.com

Dedicated to

*My beloved daughter, **Ana Paula***
and
*My love and supportive partner, **Vivi***

About the Author

Rodrigo J Hernandez is an electronic engineer passionate about IoT even before it existed. He has been working on tech for more than 20 years until now. For several years he has been focusing on the IoT ecosystem. He is currently giving consultations about IoT systems to clients around the world. He also produces content online about IoT and related subjects and the content is available on his blog, YouTube channel, and social networks - mainly LinkedIn. He also writes for companies that need good-quality content about their products and services. His main objective nowaday is to help others to understand and implement IoT solutions.

About the Reviewers

- **Asim Zulfiqar** is a blogger and tech content creator who has been writing tutorials on embedded systems and IoT on his blog and YouTube channel, High Voltages. Currently, he is working as a scientific programmer for IoT research projects.

  He completed his bachelor's degree in electronic engineering at Sir Syed University of Engineering and Technology, Pakistan. After that, he completed his Erasmus Mundus joint master's degree program in Photonics Integrated Circuits, Sensors, and Networks at Scuola Superiore Sant'Anna (Italy), Aston University (U.K), and Osaka University (Japan).

- **Rodrigo A. Aliaga Velez** is a Bolivian engineer with extensive expertise in **Industrial Internet of Things** (**IIoT**), **Internet of Things** (**IoT**), automation, and industrial optimization. With a background in mechatronics engineering and a specialization in lean energy, he has developed IIoT solutions for leading global companies such as AB InBev, Nestlé Vietnam, Acwa Power UAE, and Redev AI UK. His work focuses on enhancing efficiency, safety, and real-time data-driven decision-making through smart industrial systems.

  Rodrigo is a Lean Six Sigma Black Belt certified professional and is currently pursuing a master's degree in corporate management and Industry 4.0, further strengthening his expertise in digital transformation and industrial automation. He has successfully managed IIoT and IoT projects across various industries, driving technological innovation and process optimization.

  As a technical reviewer for an IoT Handbook, Rodrigo ensured the accuracy, clarity, and real-world applicability of the book's content. His contributions focused on key areas such as microcontrollers, sensors, actuators, IoT connectivity (MQTT, CoAP, HTTP), data management, and data visualization. With his background in industrial automation and smart technologies, he provided valuable insights into practical IoT implementations, including home automation and weather station development. His dedication to advancing industrial technology makes him a key contributor to Industry 4.0.

Acknowledgement

First and foremost, I want to thank my partner, Vivi, who always encouraged me to work on the book.

My gratitude also goes to the team at BPB Publications for their support and comprehension of the long time it took me to finish the book.

Preface

This book has been written with a clear objective: to give the reader a good introduction to the IoT world.

In this book, you will learn many concepts about hardware, embedded programming, protocols, platforms, tools, and many more topics.

The book covers theoretical concepts that are later applied to practical examples.

The book will lead you from basic definitions to advanced topics that you can apply to your projects.

This book is divided into 11 chapters. Every chapter develops a specific topic, like microcontrollers, sensors and actuators, computer Raspberry Pi 4, protocols, data storage, data visualization, and home automation, among others.

Chapter 1: Meet the Boards – In this chapter, you will explore the hardware you will use through the book to implement the projects

Chapter 2: Installing the Software Environment – In this chapter, we will see how to install and configure the software needed to develop the projects.

Chapter 3: Microcontrollers, Sensors, and Actuators – This chapter covers the fundamentals of sensors and actuators. You will learn about the different types of sensors and actuators, and how you can use them.

Chapter 4: Interfacing with Raspberry Pi – In this chapter, you will learn to connect different types of sensors and actuators with the Raspberry Pi 4 computer. You will learn how to configure the computer to be able to use it. Also, you will install Node-RED on the Raspberry Pi and use it to interact with the peripherals.

Chapter 5: Connecting IoT Devices using MQTT – This chapter covers the fundamentals of the MQTT protocol. You will learn how to use it to communicate with your IoT devices and transfer data between them and the MQTT broker installed in the Raspberry Pi 4.

Chapter 6: CoAP for IoT Connectivity – In this chapter, you will learn about the CoAP protocol. You will use different methods to exchange data between devices and a CoAP server in the Raspberry Pi.

Chapter 7: Using HTTP and WebSockets in IoT – This chapter covers both HTTP and WebSockets for exchanging data between devices and the Raspberry Pi. It explains the fundamentals of both protocols and provides many examples. It also emphasizes the differences between both protocols.

Chapter 8: Storing Internet of Things Data – In this chapter, you will learn how to store IoT data using time-series databases. In particular, this chapter focuses on the InfluxDB database, which is widely used in IoT systems. This chapter covers the fundamentals and shows you how to use InfluxDB in a real project.

Chapter 9: Visualizing Internet of Things Data – In this chapter, you will learn how to install, configure, and use Grafana to visualize IoT data. You will also see how to create alarms and notifications in Grafana.

Chapter 10: Building a Weather Station – This chapter covers the implementation of a weather station using the tools that you learned to use in the previous chapters.

Chapter 11: Home Automation – In this chapter, you will learn how to implement home automation using both Home Assistant and openHAB, which are open-source systems. You will also learn to use the Tasmota and ESPHome firmware for programming the devices that will connect with the home automation platforms.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/hy233yt

The code bundle for the book is also hosted on GitHub at
https://github.com/bpbpublications/Practical-IoT-Handbook.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

CHAPTER 1

Meet the Boards

Introduction

This chapter will provide you with an overview of the boards used throughout the book. We will explore their features, capabilities, and uses. Boards are divided into three groups: Espressif devices (ESP8266 and ESP32), ARM microcontroller (Raspberry Pi Pico), and Single Board Computer (Raspberry Pi 4). As you are aware, there exists a wide array of devices. Nevertheless, incorporating a comprehensive coverage of them would introduce unwarranted intricacy into this book.

On the other hand, this book proposes popularly used boards. Exploring these boards will establish a concrete base for learning programs on other platforms.

Structure

In this chapter, we will discuss the following topics:

- Espressif devices
- Raspberry Pi Pico
- Raspberry Pi Computer

Objectives

The objective of this chapter is to obtain a first approach to the hardware that we will use during the rest of the book.

Here you will learn the features and characteristics of microcontrollers and computers that we will use to build IoT systems.

Espressif devices

Espressif is a company that develops 32-bit microcontrollers with communications capabilities. These microcontrollers include either WiFi and **Bluetooth Low Energy** (**BLE**) 102 connectivity or both. The products are divided into three categories: SoC, modules, and development boards. Let us explore each of them.

System on chip

SoC is the acronym for system on chip. This type of device is a chip that includes all the necessary components from a functional point of view, and SoC may include the following:

- A 32-bit microcontroller, RAM, and ROM
- Peripheral interfaces (GPIO) digital inputs and outputs, ADC and DAC, (serial communication) I2C, I2S, SPI, UART, (memory access) DMA - GDMA, etc.
- Cryptographic hardware
- Real-time clock
- Clocks and timers
- WiFi and BLE controllers
- Radio frequency circuit

On the other hand, an SoC lacks:

- Programming interface, like USB
- Voltage regulators
- Antennas
- Any external component that is necessary to interface or feed the SoC

You can see an example of a SoC in *Figure 1.1*:

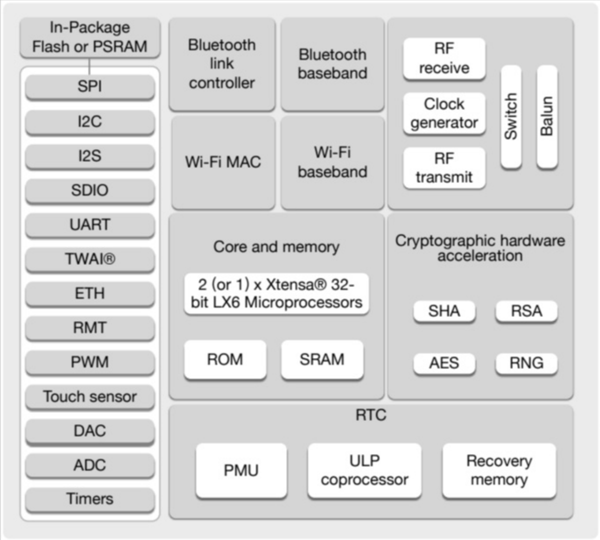

***Figure 1.1**: SoC example*[1]

Using SoCs

You may want to use a SoC for sending your product to mass production, as it is the cheapest and most efficient option to include a microcontroller in a new device. As you can see in *Figure 1.1*, a SoC has all the elements needed to obtain a fully working microcontroller with communication capabilities. However, you must design and build your **Printed Circuit Board** (**PCB**) to host the SoC accurately. This process may require several iterations. Moreover, a new PCB may require expensive certifications to be commercialized. On the other hand, you can add an SoC easily on any PCB design, given that it is a single chip.

Finally, if you are in the **proof of concept** (**PoC**) stage, you should not use an SoC as it adds unnecessary complexities.

Modules

The modules provide a higher level of integration with respect to SoCs.

The modules provide the following features:

- They include a SoC.
- They are fully certified with integrated antenna or connectors.

[1] Source: **www.espressif.com**

When to use modules

Modules are a good option if you want to produce a new device without the hassle of obtaining wireless certifications. You can add the module to your board by soldering it.

While modules come at a higher cost compared to SoCs, they offer the advantage of a reduced time-to-market.

On the other hand, modules do not provide voltage regulators or programming interfaces, so you are responsible for implementing all the necessary circuits.

You can see an example of a module in *Figure 1.2*:

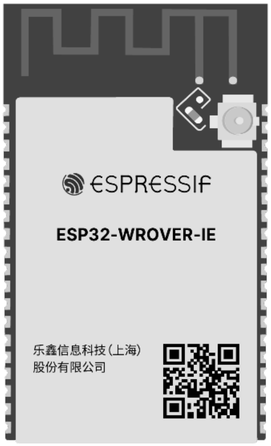

***Figure 1.2**: ESP32-based module*

Development boards

The development boards provide all the necessary elements to start testing a PoC or prototype immediately.

These boards typically offer some of these features:

- One or more microcontrollers or SOCs.
- They include all the circuits to bring you a programming interface. You can download your firmware using some kind of serial interface, typically in a USB format.
- Voltage regulators, and even battery chargers and energy harvesting interfaces (for example, to connect a solar panel).
- LEDs, buttons, displays, breadboards, connectors, SD card slots, etc.

Refer to *Figure 1.3* for an example of a development board:

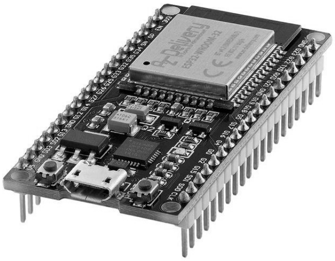

Figure 1.3*: ESP32 development board*

When to use development boards

A development board is the best option if you need to build a PoC fast and easily.

These boards are obviously more expensive than modules, but you will not use them in mass production. However, if the project involves fewer units, perhaps you can consider using them as a final solution.

We will use development boards to show the concepts described in this book with practical examples. Regarding Espressif, we will use ESP8266 and ESP32 devices.

Note: You can choose any development board to perform the examples given in this book. Just take into consideration the characteristics of your board, like GPIO pins.

Programming environments

You can program Espressif devices using several programming environments. These include ESP-IDF, Arduino, and PlatformIO.

Regarding programming languages, you can use C, C++, Arduino, MicroPython, CircuitPython, and others.

ESP-IDF

Espressif owns a programming ecosystem, called **Espressif's IoT Development Framework** (**ESP-IDF**). With it, you can program ESP32, ESP32-S, ESP32-C, and ESP32-H SoCs.

The **Software Development Kit** (**SDK**) allows you to develop fully functional IoT applications using C or C++.

The main features of ESP-IDF are the following:

- **Open source**: It is delivered under the Apache 2.0 license, and you can obtain it freely on GitHub.