Packt Publishing 1800560206

Creative DIY Microcontroller Projects with TinyGo and WebAssembly

A Practical Guide to Building Embedded Applications

Introduction

This guide serves as an instruction manual for the book "Creative DIY Microcontroller Projects with TinyGo and WebAssembly". It provides a structured overview of the book's content, how to approach the projects, and where to find additional support. The book explores embedded programming, offering hands-on experience with real-world embedded projects related to IoT, low-powered devices, and complex systems using TinyGo and WebAssembly. While Go is known for creating large executables, TinyGo is a compiler that enables developers to compile Go programs for low-memory or low-powered devices like microcontrollers and IoT devices, without requiring significant code modification. This book is a practical guide filled with DIY projects designed to teach you how to build embedded applications.

Product Overview

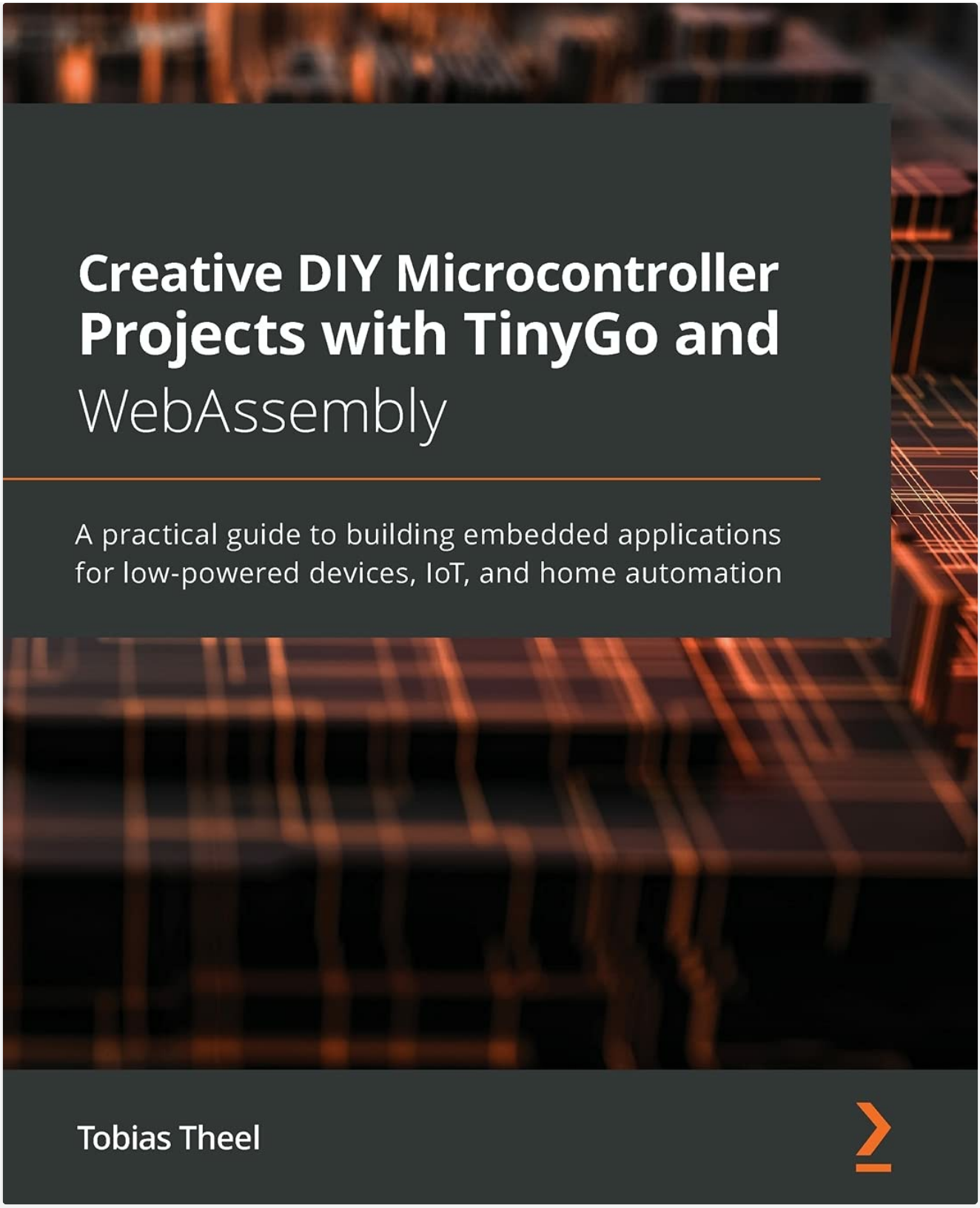

Image: Front cover of the book "Creative DIY Microcontroller Projects with TinyGo and WebAssembly" featuring the title, subtitle, and author name.

The book covers programming sensors and working with microcontrollers such as Arduino UNO and Arduino Nano IoT 33. It guides you through developing multiple real-world embedded projects using various popular devices like LEDs, 7-segment displays, and timers. You will build interactive prototypes such as a traffic lights system and a touchless hand wash timer. Advanced topics include creating an IoT prototype for a weather alert system and displaying alerts on a TinyGo WASM dashboard, culminating in a home automation project with stats displayed on the TinyGo WASM dashboard.

Getting Started (Setup)

To begin your journey with the projects in this book, ensure you have the necessary software and hardware components. The initial chapters will guide you through setting up your development environment.

Prerequisites:

- **Basic Go Programming Knowledge:** While the book is hands-on, a foundational understanding of Go will be beneficial.
- **Development Environment:** Set up Go and TinyGo compilers on your system. Instructions for this are provided early in the book.
- **Microcontrollers:** Acquire the recommended microcontrollers, such as Arduino UNO and Arduino Nano IoT 33, and other components like LEDs, sensors, and displays as specified for each project.
- **USB Cables:** For connecting microcontrollers to your computer.

The book's first chapter, "Getting Started with TinyGo," provides detailed instructions on setting up your environment and preparing for the projects.

Project Implementation (Operating)

Each project in the book is designed to be a hands-on learning experience. Follow the step-by-step instructions to build and program embedded applications.

General Workflow:

1. **Understand the Concept:** Read the introduction to each project to grasp its objective and the underlying principles.
2. **Gather Components:** Ensure you have all the required hardware components listed for the specific project.
3. **Assemble Hardware:** Follow the wiring diagrams and assembly instructions carefully. Pay attention to pin connections and power requirements.
4. **Write/Modify Code:** Implement the TinyGo code as described. The book provides code examples and explanations.
5. **Compile and Upload:** Use the TinyGo compiler to build your code and upload it to the microcontroller.
6. **Test and Debug:** Verify the project's functionality. If issues arise, refer to the troubleshooting section or re-check your wiring and code.

The book covers various projects, including traffic light control, safety locks, plant watering systems, touchless handwash timers, and more complex IoT and home automation prototypes.

Maintaining Your Skills (Maintenance)

The field of embedded systems and programming evolves rapidly. To maintain and enhance your skills beyond the book's projects:

- **Stay Updated:** Regularly check for updates to TinyGo, Go, and related libraries.
- **Explore Documentation:** Refer to the official TinyGo and Go documentation for deeper understanding.
- **Join Communities:** Participate in online forums, GitHub discussions, and local meetups related to TinyGo, Go, and embedded development.
- **Experiment:** Modify existing projects or create new ones to solidify your understanding and explore new concepts.

Troubleshooting Common Issues

When working with embedded systems, you may encounter various challenges. Here are some common issues and general troubleshooting tips:

- **Compilation Errors:** Double-check your code for syntax errors, missing imports, or incorrect variable names. Ensure your TinyGo environment is correctly set up.
- **Upload Failures:** Verify that your microcontroller is correctly connected via USB, drivers are installed, and the correct port is selected in your development environment. Ensure the microcontroller is in programming mode if required.
- **Hardware Not Responding:** Check all wiring connections for looseness or incorrect placement. Ensure components are powered correctly and that their polarity (if applicable) is observed. Test individual components if possible.
- **Unexpected Behavior:** Review your code logic. Use print statements or a debugger (if supported by your setup) to trace program execution and variable values. Simplify the circuit and code to isolate the problem.
- **Power Issues:** Ensure your power supply can provide sufficient current for all connected components. Brownouts can cause erratic behavior.

For specific project-related issues, refer back to the relevant chapter in the book. Online communities and forums can also be valuable resources for debugging.

Specifications

Attribute	Detail
Title	Creative DIY Microcontroller Projects with TinyGo and WebAssembly
Author	Tobias Theel
Publisher	Packt Publishing
Publication Date	May 14, 2021
Language	English
Print Length	322 pages
ISBN-10	1800560206
ISBN-13	978-1800560208
Item Weight	1.22 pounds
Dimensions	7.5 x 0.73 x 9.25 inches

Support and Resources

For further assistance and to engage with the community:

- **Publisher Website:** Visit the Packt Publishing website for errata, code downloads, and additional resources related to the book.
- **Author's Resources:** The author, Tobias Theel, may provide additional resources or updates through his professional channels or GitHub.

- **TinyGo Community:** Explore the official TinyGo website and its community forums for support on the TinyGo compiler and ecosystem.
- **Go Programming Language:** The official Go website provides comprehensive documentation and community links.

As this is a book, traditional product warranties do not apply. However, the publisher and author are committed to providing accurate and helpful content.