

# ESP-IDF

## Praxisstart



**Das Nachschlagewerk für Arduino und  
PlatformIO Umsteiger**



**Markus Edenhauer**

# ESP-IDF

## Praxisstart



**Das Nachschlagewerk für Arduino und  
PlatformIO Umsteiger**



# **ESP-IDF Praxisstart**

Das Nachschlagewerk für Arduino und PlatformIO  
Umsteiger

Markus Edenhauser, MA MSc

2026-04-01

# ESP-IDF Praxisstart

## ESP-IDF Praxisstart

### Einleitung

Warum ESP-IDF überhaupt lernen

Warum wir bewusst mit der CLI arbeiten

Was dich in diesem Buch erwartet

Für wen dieses Buch gedacht ist

Warum Grundlagen wichtiger sind als Features

Übungsdateien und E-Book herunterladen

Weiterführende Inhalte und aktuelle Code Beispiele

KI Transparenz Hinweis

Lizenz und Haftung

Ziel dieses Buches

### ESP-IDF Installation

Windows Installation

Linux Installation (Ubuntu/Debian Beispiel)

Häufige Fehler und schnelle Checks

Schnelltest mit einem Beispielprojekt

### Erster Start mit Hello World

Warum wir hier nur mit CLI arbeiten

Neues Projekt anlegen

Upload und Monitor

Logging und Ablauf verstehen

### Projektverzeichnis verstehen

main.c und warum du sie umbenennen solltest

CMakeLists.txt verstehen

sdkconfig verstehen

idf.py menuconfig nutzen

### Workflow und Debugging in ESP-IDF

Der Standard-Workflow

Debugging im Alltag

Crash-Analyse und Speicher

Debugging-Strategie (systematisch)

## Partitionen und Flash Layout

Warum Partitionen wichtig sind

Die Standard Partitionstabelle

Wie du die Partitionstabelle in ESP-IDF nutzt

Wichtige Hinweise

## Von Arduino zu ESP-IDF

Unterschiede Arduino und ESP-IDF

Serial.print, \_printf und ESP\_LOG

millis() und esp\_timer\_get\_time().

delay() und vTaskDelay().

## GPIO Write mit unterschiedlichen Delay Varianten

gpio\_config() verstehen

LED blinken mit vTaskDelay

LED blinken mit esp\_timer

LED blinken mit Hardware Timer

Vergleich: Wann welche Methode?

## GPIO Read

GPIO lesen

Einfaches Button Beispiel

LED mit Button schalten

## FreeRTOS Tasks (Multitasking).

Konventionen

Multithreading mit FreeRTOS

Ein einfaches Multitasking-Beispiel

## GPIO Interrupts

Interrupt Beispiel

Fazit für Arduino Umsteiger

## PWM mit LEDC

analogWrite und LEDC

PWM mit dem LEDC-Peripheral

Servo Motor ansteuern

## ADC (Analog-Digital-Wandler).

analogRead und ADC Oneshot

ADC1 und ADC2 im Vergleich

Grundlegende ADC Konfiguration

Kalibrierung und Spannungsumrechnung

## I2C

Mehrere I2C Controller

I2C Scanner Beispiel

BME280 Beispiel mit Espressif Components

WiFi Grundlagen

Station vs Access Point

Verbindung ist nicht gleich Verbindung

Verbindung und Ablauf

Event System verstehen

Synchronisation ohne Polling

Stabilität und Retry

Ressourcen und Hintergrundverarbeitung

Ablauf in Kurzform

Denkweise

HTTP GET Request

Webserver

Einfacher Webserver

HTTP Webserver mit Sensordaten

NVS (Non Volatile Storage)

Beispiel

Beispiel mit String

Praxis Tipps

LittleFS

Warum LittleFS sinnvoll ist

Typisches Szenario mit Webserver

LittleFS Komponente installieren

LittleFS Praxisbeispiel

Beispiel

MQTT Publish und Subscribe

MQTT Komponente installieren

Publish

Subscribe

Subscribe mit GPIO Toggle

Abschließende Worte

Wie es weitergeht

Downloadlinks

Über den Autor  
Weiteres vom Autor

# ESP-IDF Praxisstart

Copyright ©  
April 2026 1. Auflage  
Selbstverlag

Markus Edenhauser, MA MSc  
Völser Straße 41  
6020 Innsbruck  
Österreich  
[pixeledi.eu/impressum](http://pixeledi.eu/impressum)



Du bist mein `idf.py flash`. Wenn nichts mehr geht bringst du trotzdem alles zuverlässig zum Laufen.



# Einleitung

Willkommen zu **ESP-IDF Praxisstart - Das Nachschlagewerk für Arduino und PlatformIO Umsteiger**.

Ich bin Edi und begleite dich in diesem Buch beim Wechsel von der gewohnten Arduino Welt in die strukturierte und deutlich mächtigere Welt der ESP-IDF. Wenn du bisher mit Arduino IDE oder PlatformIO gearbeitet hast und dich gefragt hast, was unter der Haube wirklich passiert, dann bist du hier genau richtig.

Viele Maker kommen irgendwann an den Punkt, an dem einfache Sketches nicht mehr ausreichen. Projekte werden größer, Anforderungen komplexer, Speicher knapper und plötzlich tauchen Begriffe wie Tasks, Partitionstabellen oder NVS auf. Genau hier setzt dieses Buch an. Es soll dir keinen theoretischen Overload liefern, sondern dir einen sauberen, verständlichen und praxisnahen Einstieg ermöglichen.

Dieses Buch ist bewusst als Grundlagenwerk konzipiert. Es soll dir ein stabiles Fundament geben, auf dem du später aufbauen kannst.

## Warum ESP-IDF überhaupt lernen

Wenn du bisher mit Arduino gearbeitet hast, hast du bereits viel richtig gemacht. Arduino ist ideal für den Einstieg, schnelle Prototypen und erste Erfolgserlebnisse. Doch der ESP32 ist mehr als nur ein Mikrocontroller mit einer bequemen Arduino Abstraktion.

Die ESP-IDF ist das offizielle Entwicklungsframework von Espressif für den ESP32. Sie gibt dir direkten Zugriff auf das System, auf FreeRTOS, auf Speicherverwaltung, auf Netzwerkfunktionen und auf viele Hardware Details, die in Arduino oft verborgen bleiben.

Mit ESP-IDF lernst du:

- wie Tasks wirklich funktionieren
- wie Speicher organisiert ist
- wie Partitionen und OTA Updates aufgebaut sind
- wie Logging professionell eingesetzt wird
- wie du strukturierte Projekte entwickelst

Du wechselst also von einer vereinfachten Umgebung in eine professionelle Firmware Architektur.

Dieses Buch zeigt dir genau diesen Übergang.

## **Dein Einstieg in die Praxis**

Dieses Buch ist praxisorientiert aufgebaut. Du wirst nicht seitenlang Theorie lesen, bevor du etwas ausprobieren darfst. Schon früh legst du ein Projekt an, kompilierst es, flasht es und arbeitest direkt mit der Konsole.

Ich lege großen Wert darauf, dass du selbst tippst und ausprobierst. Nur durch eigenes Arbeiten verstehst du:

- was idf.py wirklich macht
- wie Build Prozesse funktionieren
- wie Flashen und Monitor zusammenspielen
- wie sich Projekte strukturieren

Lesen alleine reicht nicht. Firmware Entwicklung ist Handwerk. Du musst es tun, nicht nur verstehen.

# Warum wir bewusst mit der CLI arbeiten

Vielleicht bist du es gewohnt, in der Arduino IDE auf einen Button zu klicken. In der ESP-IDF Welt ist die Befehlszeile ein zentrales Werkzeug.

Die CLI ist kein Rückschritt, sondern Kontrolle.

Wenn du mit `idf.py` arbeitest, verstehst du:

- wie der Build Prozess aufgebaut ist
- welche Schritte beim Flashen passieren
- wie Konfiguration über `menuconfig` funktioniert
- wie Projekte reproduzierbar gebaut werden

Viele Entwickler nutzen ESP-IDF heute über PlatformIO, indem sie dort als Framework `espidf` auswählen. Technisch basiert das ebenfalls auf der offiziellen ESP-IDF. In diesem Buch gehen wir jedoch bewusst den direkten Weg über das originale Tool `idf.py`.

Der Grund ist einfach: Du sollst verstehen, was im Hintergrund passiert. Der Build Prozess, die Projektstruktur, die Konfiguration und das Flashen des Boards werden hier ohne zusätzliche Abstraktionsschicht vermittelt. Du arbeitest direkt mit dem offiziellen Workflow von Espressif.

PlatformIO ist kein schlechter Ansatz. Es ist ein komfortables Werkzeug mit guter IDE Integration und Projektverwaltung. Doch Komfort darf nicht auf Kosten des Verständnisses gehen. Gerade als Umsteiger ist es entscheidend, einmal den Kern des Systems kennenzulernen.

Sobald du dieses Fundament beherrschst, kannst du jederzeit auf PlatformIO oder andere Werkzeuge wechseln. Das technische Verständnis bleibt dasselbe.

## **Was dich in diesem Buch erwartet**

Wir starten bei der Installation und prüfen, ob dein System korrekt eingerichtet ist. Danach erstellst du dein erstes Hello World Projekt und arbeitest dich Schritt für Schritt weiter.

Du lernst unter anderem:

- Projektstruktur und CMake Grundlagen
- sdkconfig und menuconfig
- Partitionstabellen und Flash Layout
- Unterschiede zwischen Arduino und ESP-IDF
- FreeRTOS Tasks
- GPIO mit verschiedenen Timing Varianten
- Interrupts
- PWM mit LEDC
- ADC Grundlagen
- I2C Kommunikation
- WiFi, HTTP und einfache Webserver
- NVS und LittleFS
- MQTT Publish und Subscribe

Alle Kapitel sind so aufgebaut, dass du sofort nachvollziehen kannst, wie ein reales Projekt entsteht.

Dieses Buch deckt bewusst nur die Grundlagen ab. Es ist kein vollständiges Referenzhandbuch und kein tiefgehendes Architektur Werk. Es ist dein strukturierter Einstieg.

# Für wen dieses Buch gedacht ist

Dieses Buch richtet sich an:

- Arduino Entwickler, die mehr Kontrolle wollen
- Maker, die ihre Projekte strukturierter aufbauen möchten
- Entwickler, die verstehen wollen, wie der ESP32 intern arbeitet
- Nutzer von PlatformIO, die wissen möchten, was hinter der Arduino Abstraktion steckt

Du solltest grundlegende C oder C++ Kenntnisse mitbringen. Absolute Anfänger ohne Programmiererfahrung werden es schwerer haben.

## Üben statt kopieren

In Zeiten von KI und Copy Paste ist es verlockend, fertigen Code einfach zu übernehmen. Doch echtes Verständnis entsteht nur, wenn du selbst schreibst, Fehler machst und sie behebst.

Tippe die Beispiele selbst ein. Verändere Parameter. Probiere aus, was passiert, wenn du Werte änderst. Spiele mit Delays, Task Prioritäten oder Buffer Größen.

Fehler sind kein Problem. Sie sind Teil des Lernprozesses.

Wenn du ein Projekt selbst zum Laufen bringst, bleibt das Wissen dauerhaft.

## Warum Grundlagen wichtiger sind als Features

Viele Tutorials springen sofort zu komplexen Themen wie BLE, Audio Streaming oder komplexen Cloud Architekturen. Das wirkt beeindruckend, führt aber oft zu oberflächlichem Verständnis.

In diesem Buch konzentrieren wir uns auf:

- saubere Struktur
- Verständnis von Tasks
- Speicherorganisation
- Logging
- Partitionen
- persistente Daten

Wenn du diese Grundlagen beherrschst, kannst du jedes weiterführende Thema selbstständig angehen.

Fundament vor Feature.

# Übungsdateien und E-Book herunterladen

Alle Unterlagen findest du im GitHub-Repository:  
<https://github.com/pixelEDI/esp-idf-course>

Im Repository findest du immer die aktuelle Version der Code-Beispiele. Du kannst das Projekt klonen oder als ZIP-Datei herunterladen - ohne Anmeldung. Ohne Git: **<> Code** → **Download ZIP** anklicken, um die Dateien direkt im Browser zu laden.

Den Download-Link für das PDF findest du am Ende des Buches im Kapitel **Downloadlinks**.

## Weiterführende Inhalte und aktuelle Code Beispiele

Die ESP-IDF entwickelt sich ständig weiter. Neue Funktionen, Änderungen an Bibliotheken oder neue Best Practices entstehen regelmäßig. Deshalb ergänze ich die Inhalte dieses Buches laufend mit zusätzlichen Videos und praktischen Beispielen.

Auf meinem YouTube Kanal veröffentliche ich regelmäßig neue Inhalte rund um den ESP32 und die ESP-IDF. Dort findest du unter anderem:

- vertiefende Erklärungen zu einzelnen Themen aus diesem Buch
- neue Projekte und Praxisbeispiele
- Updates zu Änderungen im ESP-IDF Ökosystem
- zusätzliche Tipps aus der Praxis

Gerade bei Themen wie WiFi, FreeRTOS, Speicherverwaltung oder neuen ESP32 Varianten lohnt es sich, diese Inhalte im Blick zu behalten.

Zusätzlich stelle ich dort auch **aktuelle Code Snippets und kleine Beispielprojekte** zur Verfügung, die über die Inhalte dieses Buches hinausgehen. Diese können dir helfen, eigene Projekte schneller umzusetzen oder neue Ideen auszuprobieren.

Die Links zu den Videos und Code Beispielen findest du direkt auf meinem Kanal sowie in den jeweiligen Video Beschreibungen.

<https://www.youtube.com/@pixeledi>

## **KI Transparenz Hinweis**

Dieses Buch wurde inhaltlich von mir selbst konzipiert und geschrieben. KI Werkzeuge wurden unterstützend eingesetzt, ähnlich wie Dokumentationen oder Foren. Die Konzepte, Beispiele und Struktur stammen aus meiner eigenen praktischen Erfahrung mit ESP32 Projekten.

KI war ein Werkzeug zur Unterstützung bei Formulierungen und Strukturierung, nicht der Autor.

Die Verantwortung für den Inhalt liegt vollständig bei mir.

## **Lizenz und Haftung**

Die Beispiele und Übungsprojekte dieses Buches dürfen für eigene Projekte frei verwendet werden. Du kannst sie anpassen, erweitern und in deinen eigenen Projekten einsetzen.

Ich übernehme keine Haftung für Schäden, Datenverluste oder Fehlfunktionen, die durch unsachgemäße Verwendung entstehen können. Arbeite sorgfältig und teste neue Konzepte idealerweise zunächst in kleinen Projekten.

## **Ziel dieses Buches**

Am Ende dieses Buches sollst du:

- ein ESP-IDF Projekt sicher erstellen können
- Tasks verstehen und einsetzen
- GPIO strukturiert nutzen
- einfache Netzwerk Anwendungen schreiben
- persistente Daten speichern
- wissen, wie ein professioneller ESP32 Build Prozess funktioniert

Du bekommst ein solides Fundament.

Alles Weitere baust du darauf auf.