



# Überblick zum ESP32 mit Arduino

Helmut Tschemernjak

Arduino-Hannover

[www.arduino-hannover.de](http://www.arduino-hannover.de)

# Arduino Platinen

## ■ **Arduino UNO**

Microchip 8-bit AVR MCU



## ■ **ESP8266**

Espressif WiFi Chip mit TCP/IP

Tensilica Xtensa 32-bit-Prozessor



## ■ **ESP32**

Espressif MCU welche auch WiFi/BT kann

Tensilica Xtensa Dual Core 32-bit-Prozessor



# Gemeinsamkeiten

## ■ **USB-Anschluss**

USB-zu-Seriell Adapter

Schnittstelle zum Programmieren oder Serial-Monitor Fenster

Stromversorgung per USB



## ■ **Arduino IDE**

Identische Arduino APIs für alle Arduino-Boards



## ■ **MCU**

Prozessor mit SRAM, Flash

Peripherie: GPIO-Pins, UART, SPI, I2C, PWM, Analog, Timer ...



# ESP32 ganz einfach

## ■ **ESP32 mit Batterie verbinden**

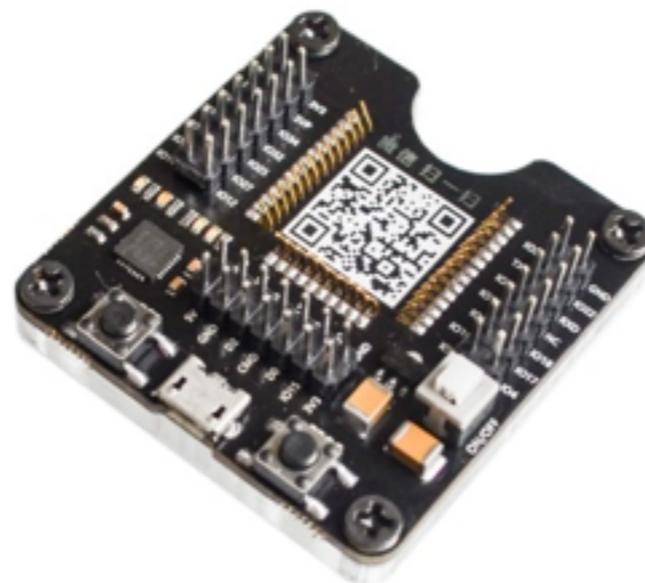
Spannungsversorgung: 2,5 bis 3,6 Volt

+ (VDD), - (GND) verbinden

## ■ **Reset-Taster**

Reset (EN), mit 10 k $\Omega$  auf VDD

Pin 2 mit 10 k $\Omega$  auf VDD



# ESP32 WROOM Module

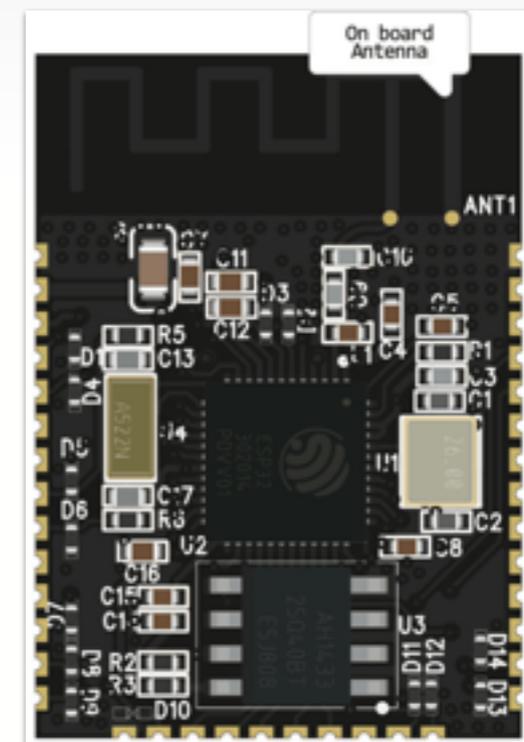
- **WiFi-Antenne**
- **ESP32 MCU**
- **4 MB Flash Chip**
- **40 MHz Quarz**
- **Stromverbrauch**

Deepsleep 7  $\mu$ A

Lightsleep 1 mA

Normal 50 mA (bei 240 MHz, mit 2 MHz nur 3,8 mA)

WiFi 80-180 mA



## Flash Aufteilung (Partition-Tabelle)

### ■ Datei: `tools/partitions/default.csv`

Es gibt: `default.csv`, `min_spiffs.csv`, `minimal.csv`, `no_ota.csv`

<http://esp-idf.readthedocs.io/en/latest/api-guides/partition-tables.html>

Name	Partition	Größe	Kommentar
<b>nvs</b>	<b>Permanentspeicher</b>	<b>20 kB</b>	<b>WiFi settings, usw.</b>
<b>otadata</b>	<b>Bootloader</b>	<b>8 kB</b>	<b>2nd Bootloader für OTA</b>
<b>app0</b>	<b>Programm1</b>	<b>1,28 MB</b>	<b>Arduino Sketch OTA0</b>
<b>app1</b>	<b>Programm2</b>	<b>1,28 MB</b>	<b>OTA1 (Over the Air)</b>
<b>eeeprom</b>	<b>EEPROM</b>	<b>4 kB</b>	<b>Arduino Emulator</b>
<b>spiffs</b>	<b>Flash Dateisystem</b>	<b>1,46 MB</b>	<b>Bilder für Webserver, usw.</b>

# ESP32 – die Zukunft für Arduino

## ■ **ESP32**

Wöchentliche Verbesserungen

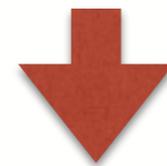
Issues: 821 closed, 379 open



## ■ **D21 Atmel**

Hier passiert über Monate nichts

Issues: 108 closed, 81 open



## ■ **AVR**

Wenige Änderungen, es gibt auch nichts Neues

**GitHub**



Stand 6/2018

# ESP32 – die Zukunft für Arduino

## ■ **Umfangreiche Unterstützung für Arduino**

Leistungsfähigste MCU für Arduino

Zusätzlich WiFi und BlueTooth

Umfangreicher Arduino Support durch Espressif

Sourcecode auf Github verfügbar (IDF und Arduino)

Technische Dokumentation ist vorbildlich

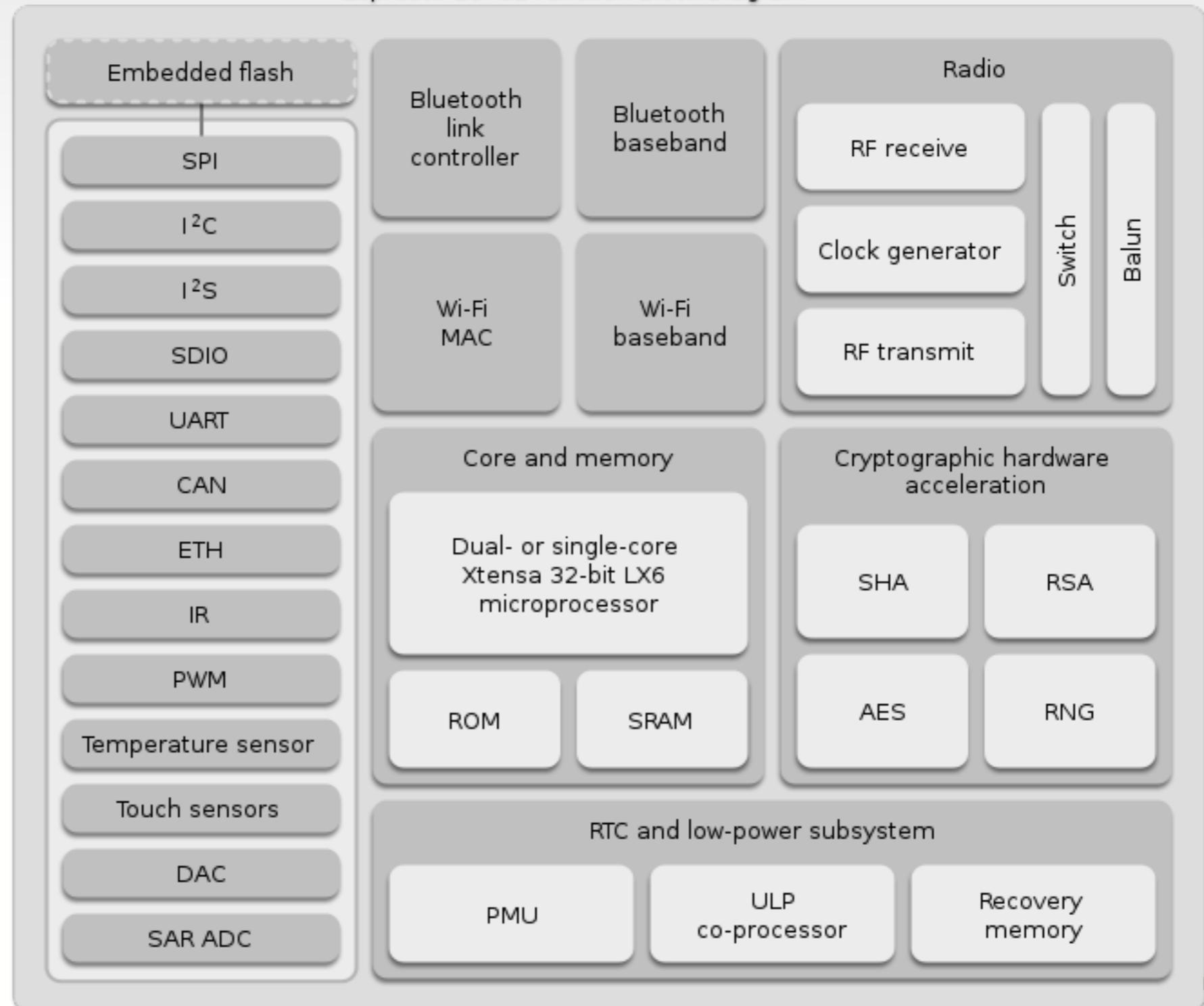
## ■ **Günstiger Einstieg**

Module für ab 3 Euro

Boards mit USB ab 5 Euro

ESP32

Espressif ESP32 Function Block Diagram



# Beispiele der Leistungsfähigkeit

## ■ **Timer 64-bit (4 Einheiten davon)**

Timer mit 16-, 24- oder 32-bit – Überlauf nach Sekunden/Minuten

Timer mit 64-bit – Überlauf nach 500 Jahren!

## ■ **Dual Core CPU**

Ein Kern kann z. B. Bildverarbeitung, WiFi machen, der Rest läuft weiter

## ■ **16 kB RTC-Memory**

RTC Speicher bleibt im Deepsleep erhalten

## ■ **Verschlüsselter Flash-Speicher**

Programme können nicht ausgelesen werden, eindeutige IDs, usw.

# Ist das wirklich alles nötig?

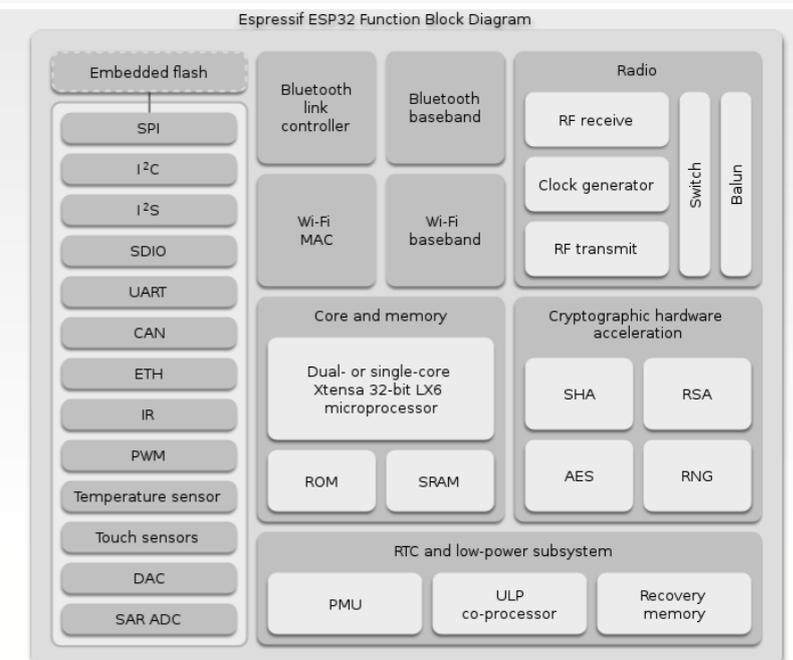
## ■ ESP32 – Dual Core 240 MHz

512 kB RAM, 4 MB Flash, WiFi, BlueTooth, ...

Als die ersten PCs 8 MHz, 32 kB RAM hatten, gab es die gleiche Diskussion

## ■ Anwendungsbeispiele:

Web-Userinterface, BlueTooth-Userinterface, Tonverarbeitung, Bildverarbeitung, Verschlüsselung, TCP/IP-Protokolle, Webclient, Lichtsteuerung, unzählige Sensoranwendungen, Datenlogger, Touchscreen, ...



# Überblick: CPU Sleep

	ESP32 Deepsleep	D21 (Ard. Zero) Suspend	STM32-L4 Stop-2
<b>Power</b>	<b>7 <math>\mu</math>A</b>	<b>150 <math>\mu</math>A</b>	<b>2 <math>\mu</math>A</b>
<b>CPU</b>	<b>aus</b>	<b>standby</b>	<b>standby</b>
<b>Memory</b>	<b>aus</b>	<b>standby</b>	<b>standby</b>
<b>Interrupts aktiv</b>	😡	😊 😡	😊
<b>RTC mit Wakeup</b>	😊	😊	😊
<b>Timer aktiv</b>	😡	😊	😊
<b>Wakeup Delay</b>	<b>250 ms</b>	<b>&lt; 100 <math>\mu</math>s</b>	<b>5 <math>\mu</math>s</b>
<b>ULP-Prozessor</b>	😊	-	-

## ESP32: Nachteile

### ■ **USB-Support fehlt**

Zum Programmieren wird ein externer USB-zu-Seriell-Wandler genutzt.  
Der ESP32 hat kein USB Host/Device Interface

### ■ **Ungenauere ADCs**

Einfache Messungen funktionieren ausreichend, für präzise Messungen entweder jeden ESP32 einzeln kalibrieren oder externen ADC nutzen

### ■ **Interrupt-Routinen müssen komplett im RAM sein**

Das externe Flash ist möglicherweise im Interrupt nicht verfügbar

### ■ **RTC-Uhr nicht brauchbar**

Die Uhr wird beim Reset auf 0 zurückgesetzt, ungenau im Sleepmodus

# ESP32: Chip-Revisions

## ■ **Revision 0**

Einfach in den Müll werfen. Es gab noch Fehler in den Timern, Sleepmodus-Probleme und Abstürze

## ■ **Revision 1**

Alles gut, keine kritischen Fehler

Neue Chips haben ADC Vref gesetzt

Die Revision wird beim Programmieren angezeigt

## Vielfalt an ESP32-Boards

- **Espressif ESP32-DevKitC**

Einfaches Referenzboard, allerdings ohne LEDs



- **Espressif ESP32-WROVER**

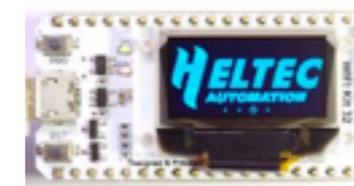
Umfangreiches Referenzboard, mit Bildschirm, LEDs, Kamera-Interface, JTAG-Debugger und SD-Steckplatz



- **Unendlich viele ESP32-Boards bei AliExpress**

Mit LiPo, OLED, Adapterplatinen, externer Antenne, ...

- **Ada Feather ESP32, Sparkfun ESP32, ...**



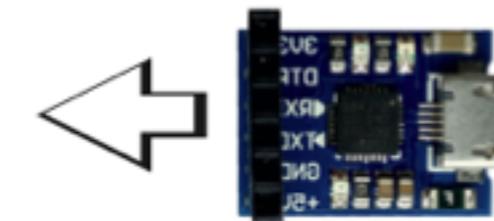
**Batteriespannungsüberwachung**

**Erweiterungsleisten**

**Präzise Uhr (RTC)**

**Externe Stromversorgung**  
Micro-USB

**USB-Programmieradapter**  
mit Stromversorg. (Micro-USB)



**ESP32 CPU**  
WiFi/Bluetooth

**User-Taste**  
**Reset-Taste**

**LEDs**  
(rot/grün)

**SMA-Antennenanschluss**  
optional, bei Bedarf anlöten

**Drahtantennenkontakt**

**U.FL-Antennenanschluss**

**LoRa-Funkmodem**

**Stromschalter (Software)**  
für externe Verbraucher ( $V_{dd}$ )

**Externer Batterieanschluss**

**5 Jahre Batteriebetrieb**  
mit einer 3-V-Lithiumbatterie (CR123)

Unterstützung für LiFePO<sub>4</sub>-Batterien  
(Rückansicht)

**Sensor Si7021**  
Temperatur/Luftfeuchtigkeit  
(Anschluss für Board  
oder externe Nutzung)



## ESP32: Zusammengefasst

- **Leistungsfähige 32-bit MCU**
- **Beste Unterstützung für Arduino**
- **Batteriebetrieb möglich**
- **Große Boardauswahl**
- **Hersteller Espressif ist sehr aktiv**

**Wir dürfen gespannt sein, was von Espressif als nächstes kommt.**



Vielen Dank!