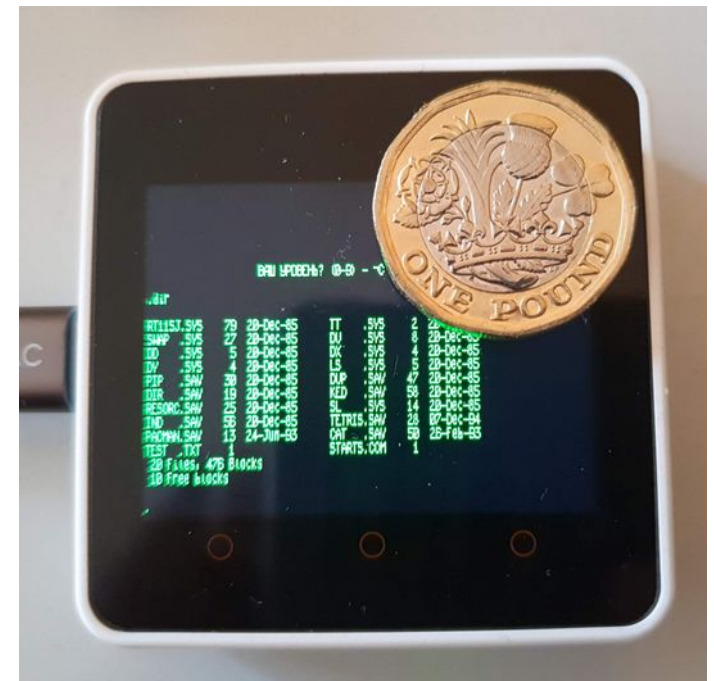


# Der Microcontroller Espressif ESP32

Matthias Brüstle  
Nürnberg 2023-11-26

# esppdp11

- 2.11BSD pl473 (aktuell pl479)
- Login: user
- Password: Kongress
- 3.5 MB RAM
- 1 GB "HDD" RA72
- 370 dhrystone/sec =  
0.21 VAX MIPS
  - PDP-11/34: 0.25 VAX MIPS
  - PDP-11/70: 0.71 VAX MIPS



# Sales Sheet

- 32-bit Dual-Core Xtensa LX6, bis 240 MHz
- 32-bit Ultra Low Power (ULP) Coprozessor, 150 kHz
- 520 kB RAM
- 448 kB ROM
- Wi-Fi 802.11 b/g/n
- Bluetooth v4.2 BR/EDR/BLE
- 34 GPIOs
- 12-bit ADCs, 8-bit DACs
- 10 Berührungssensoren
- 4 SPI, 2 I2S, 2 I2C, 3 UART
- SD/SDIO-Controller
- Ethernet Interface
- CAN bus
- IR
- PWM
- Cryptobeschleuniger (AES, SHA-2, RSA, ECC, RNG)
- Power Management
- **3.3 V**

<https://wolles-elektronikkiste.de/en/programming-the-esp32-with-arduino-code>

# Und Varianten

- ESP32-C3
  - 32-bit Single-Core RISC-V, 160 Mhz
- ESP32-S3
  - Dual-Core Xtensa LX7 mit FPU, 240 MHz
  - Low-Power RISC-V, ULP
- und noch viele weitere Geschmacksrichtungen
- <https://www.espressif.com/en/products/socs/esp32>

# ESP8266

Vorgänger des ESP32 und immer noch in Verwendung

- 32-bit Single-Core L106, bis 160 MHz
- 112 kB RAM
- üblicherweise 1 MB Flash
- Wi-Fi 802.11 b/g/n
- 17 GPIOs
- SPI, I2C, I2S, UART, 10-bit ADC

# Bastel-Varianten in der Praxis

- Als erstes zur Hand:
  - ESP32-WROOM-32D (520 kB RAM, 4 MB Flash, PCB-Antenne, USB)
  - ESP32-WROOM-32U (520 kB RAM, 4 MB Flash, Antennenstecker, USB)
  - ESP32-WROVER (520 kB RAM + typ. 8 MB PSRAM via SPI, 4 MB Flash, USB)
- Immer noch ESP8266 für kleinere Probleme
- Markt für den ESP32 ist eigentlich kommerzielle Produktentwicklung wo man das Modul direkt einbaut, im Gegensatz zu Arduino

# Entwicklungsumgebungen von Wikipedia

- ESP-IDF: Espressif IoT Development Framework
- Arduino-ESP32: Arduino-IDE mit ESP32-Board-Erweiterung
- Espruino: "JavaScript SDK das so tut als ob Node.js"
- MicroPython (and CircuitPython): "Python 3 für Microcontroller"
- Lua Network/IoT toolkit for ESP32
- Moddable SDK: "JavaScript and TypeScript with networking, graphics, and ECMA-419 APIs"
- Mongoose OS: "Betriebssystem für vernetzte Controller, JavaScript/C. Empfohlen von Espressiv, Amazon, Google."
- mruby for the ESP32
- NodeMCU: "Lua-basierte Firmware"
- PlatformIO
- Rust
- Visual Studio Code offizieller ESP-IDF-Erweiterung
- Zerynth: "Python for IoT and microcontrollers"
- Matlab
- Matlab Simulink

# Entwicklungsumgebungen (2)

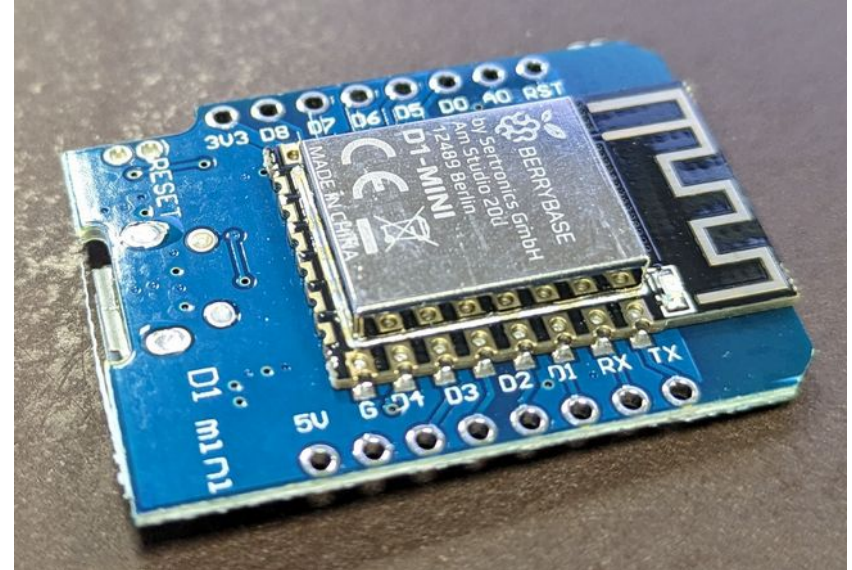
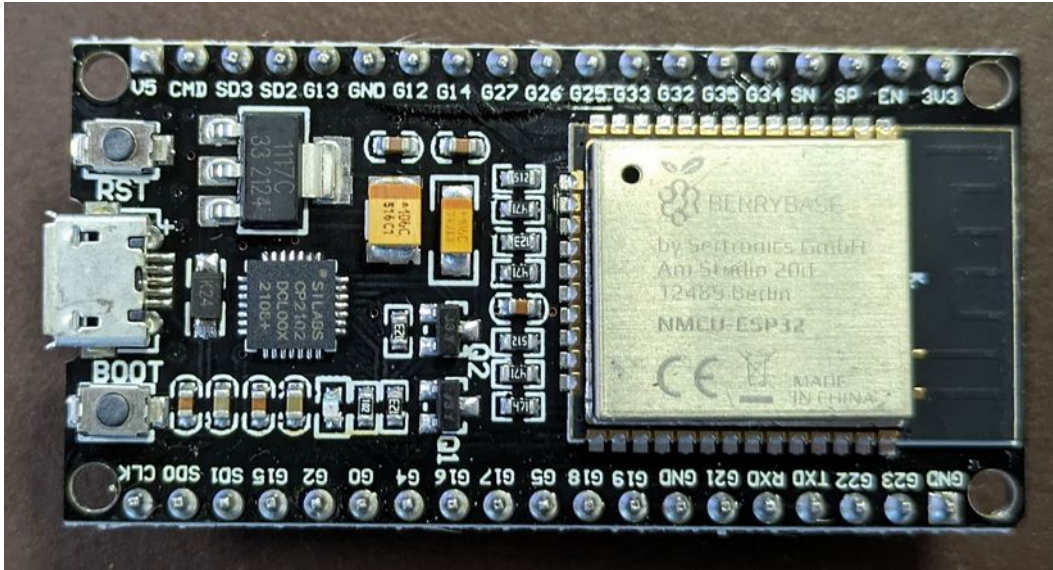
- ESP-IDF: Espressif IoT Development Framework
  - DIE offizielle Entwicklungsumgebung
  - Basiert auf gcc
  - CLI mit Python-Scripten
  - Eigenes Library-System
  - <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>
- Arduino-ESP32: Arduino IDE mit ESP32-Board-Erweiterung
  - So wie die Arduino 1.8/2.0 IDE halt ist
  - Arduino-IDE installieren, URL beim Boards Manager rein kopieren, ESP32-Boards-Erweiterung installieren, loslegen
  - Boards Manager ist von Espressif und eigentlich ein ESP-IDF



# Besonderheiten bei der Programmierung

- Auf dem ESP32 läuft mit esp-idf und Arduino FreeRTOS
- Arduino: `#include "freertos/FreeRTOS.h"`
- Tasks mit Queues, Semaphors, Mutexes
  - Bequem 1 Core für WLAN und 1 Core für Anwendung (normales Arduino)
  - Oder verteilt auf 2 Cores
- <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/index.html>

# Konkret Bastel-Hardware Dev Kit / NodeMCU



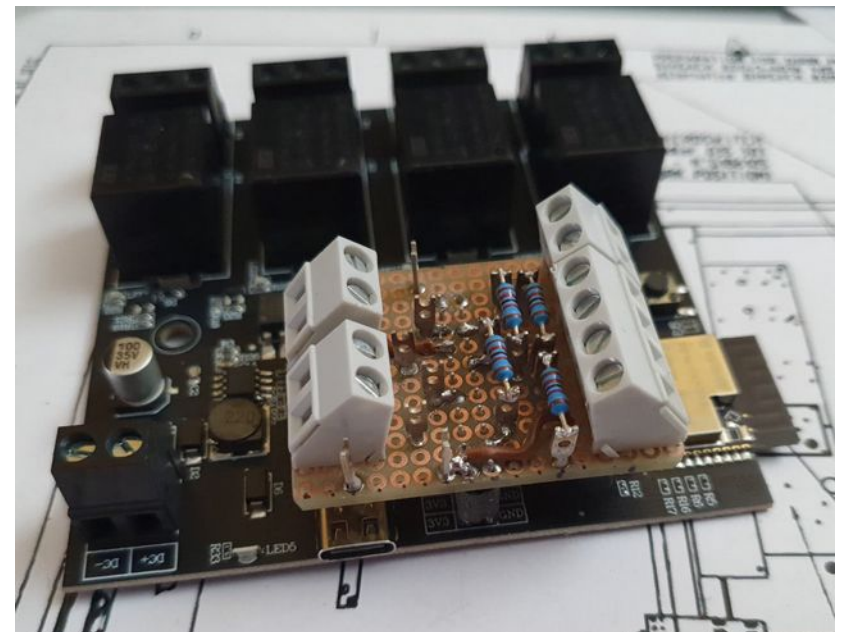
- <https://www.berrybase.de>
- <https://eckstein-shop.de>
- <https://de.aliexpress.com>



# Konkret Bastel-Hardware (3)

## LILYGO

- <https://www.lilygo.cc>
- Besondere nackte ESP-Boards mit
  - Displays
  - RF: ZigBee, LoRa
  - Mobilfunk
  - GPS
  - Relais



# Konkret Bastel-Hardware (4)

## LILYGO

- Wearables



T-Watch S3  
3 reviews  
From \$2.59 USD



T-Watch 2020 V3  
5 reviews  
From \$35.29 USD



T-Impulse LoRa Wristband  
From \$31.68 USD



T-Watch 2020 400mAh  
3 reviews  
From \$33.42 USD



T-Watch 2021  
1 review  
From \$25.62 USD



Mini E-Paper Core  
From \$12.60 USD



Sold out

T-Wristband  
1 review  
From \$28.55 USD



Watch-Keyboard-C3 V1.0  
From \$13.55 USD

# Konkret Bastel-Hardware (5)

## Home Automation

- ESP32 verbreitet, aber am absteigenden Ast
- Tasmota: <https://tasmota.github.io/docs/>
- Tasmota-Gerätedatenbank:  
<https://templates.blakadder.com/index.html>
- ESPHome: <https://esphome.io/>
- ESPHome unterstützt ESP8266, ESP32, RP2040, BK72xx, RTL87xx
- Evtl. außen gleich, innen anderer Controller
- Gelegentlich auch der Schreibschutz aktiviert

# Konkret Bastel-Hardware (6)

## Home Automation

- WLED: <https://kno.wled.ge/>
- NeoPixel-Steuerung (WS2812 / SK6812) per WiFi
- RGB und RGBW
- LED-Kette mit serieller Ansteuerung
  - 3 Drähte
  - 5V, GND, Data
- App für Android und Apple

# Meine bisherigen Projekte

Beleuchtung von Häusern auf der Modelleisenbahnanlage eines Kollegen mit WLED auf D1-Mini (ESP8266)

- Bahnhofsbeleuchtung mit Bahnhofsdisco
- Braucht nur Stromkabel



# Meine bisherigen Projekte (2)

Auswerten eines manipulationsgeschützten Magnetkontaktes (LILYGO mit Relais)

- 2 Reed-Kontakte, 4-adriger Anschluss, Abschlusswiderstand
- ADC
- Kleine Stack-Maschine zur Verknüpfung von 4 Ein- und 4 Ausgängen
- 7kB Quellcode, Arduino-IDE

# Meine bisherigen Projekte (3)

- Telegram-Dongle zum kontrollierten Senden von Meldungen in einen Telegram-Kanal
- Eingang: USB-Seriell
- Ausgang: Ethernet
- M5Stack Atom + Ethernet
- Arduino-IDE
- 12kB Quellcode mit Ethernet (*Ethernet*), TLS (SSLClient), Passwort-Authentisierung am Dongle, Konfigurierbarkeit des Kanals



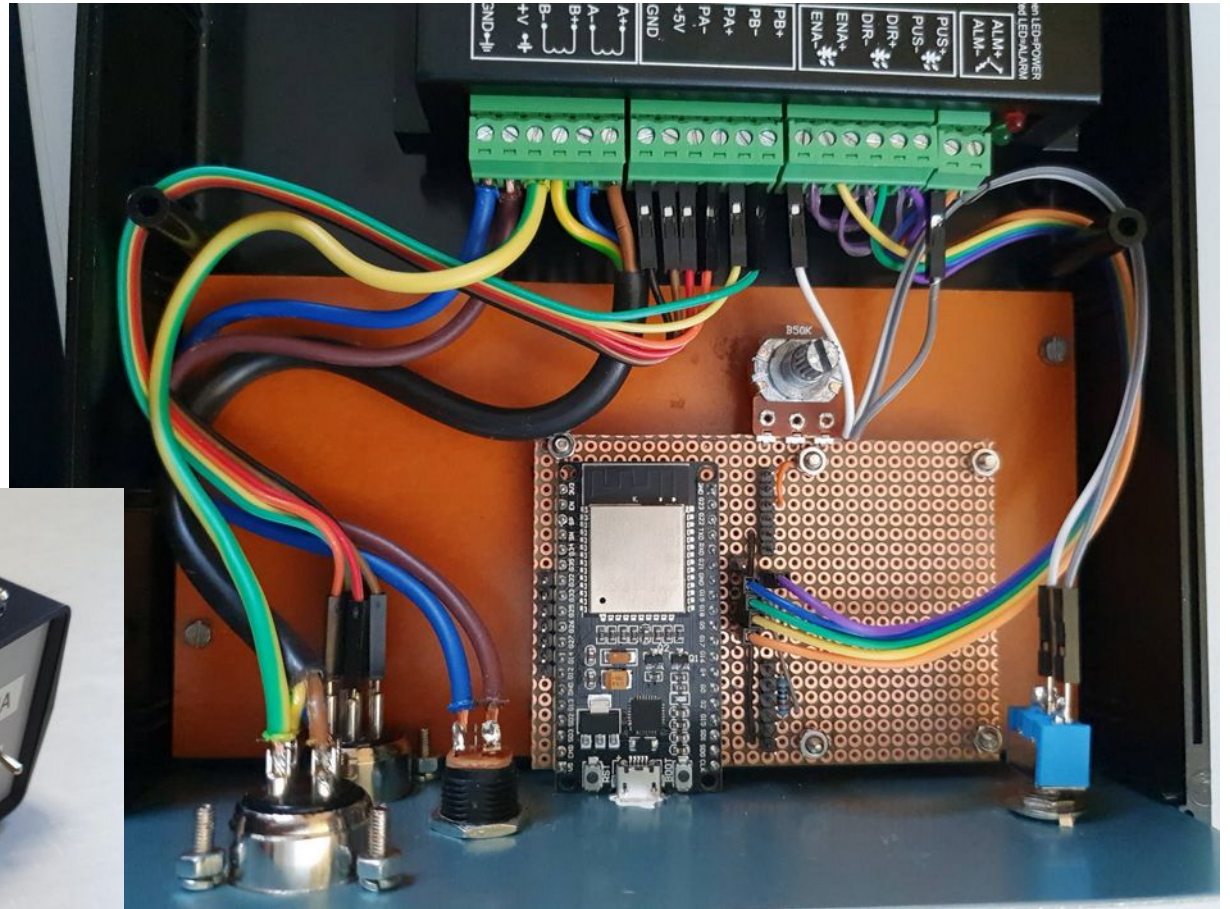
# Meine bisherigen Projekte (4)

"Autodialer", also ein Gerät, das programmgesteuert den Knopf eines mechanischen Zahlenschlosses bedient

- 90 kB Quellcode, Arduino-IDE
- Bedienung über USB-Seriell und LCD mit Ziffernblock
- Normales Dev Kit
- Closed-Loop-Motor (*FastAccel/Stepper*)
- 3 Tasks auf 1 Core (USB-Seriell, LCD/Tastatur, Motor)
- Semaphoren und Queues

# Meine bisherigen Projekte (5)

So sieht's aus



# Meine bisherigen Projekte (6)

So sieht's aus



# Meine bisherigen Projekte (6)

## esppdp11 (2)

- Basis:
  - <https://spritesmods.com/?art=minipdp11>
  - <https://www.youtube.com/watch?v=jtmNOZVrz98>
  - Entwicklungsumgebung esp-idf
- M5Stack nicht übersichtlich
- Ohne Kenntnisse von esp-idf und fremder Code
- 2 MB Quellcode
- 1.2 MB Binary
- Geht's der PDP-11 noch gut?

# Weitere Informationen

- <https://www.espressif.com/en/products/socs/esp32>
- <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/>
- <https://randomnerdtutorials.com>

# Der Microcontroller Espressif ESP32



Matthias Brüstle  
Nürnberg 2023-11-26