

Matter na ESP32:

Jak jsem to málem vzdal

Václav Steiner • LinuxDays 2026 • 4. října 2026

Co nás dnes čeká

- Trochu teorie
- Série pokusů a omylů

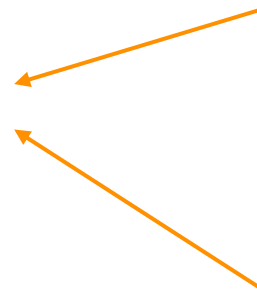
Co je Matter?

- Univerzální komunikační standard pro IoT, bez cloudu
- Open-source, licence Apache 2.0
- Konsorcium Connectivity Standards Alliance (dříve Zigbee Alliance)
- Původní pojmenování: Connected Home over IP (CHIP)
- E2EE komunikace, šifrování AES-128-CCM
- Používá se výhradně IPv6
 - přímé propojení mezi zařízeními → žádný NAT
 - rozšířená podpora multicastu → objevování zařízení
 - SLAAC, IPSec zabezpečení komunikace

Co je Matter? Certifikace

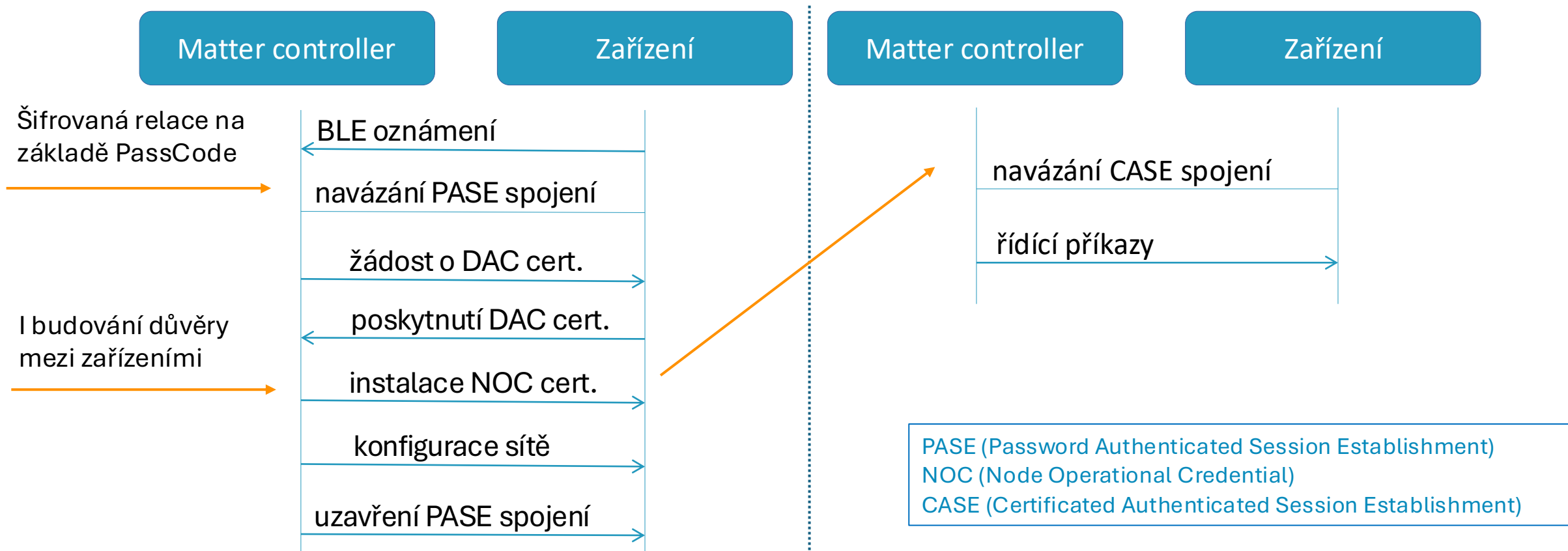
- spravuje CSA, splnění standardů → nebude vendor-lock
 - Device Attestation (na principu PKI)
 - zařízení je pravé, pochází od daného výrobce, prošlo testy
- Certifikáty
 - DAC (Device Attestation Certificate)
 - PAI (Product Attestation Intermediate)
 - PAA (Product Attestation Authority)
- (DCL) Distributed Compliance Ledger
 - Uložené všechny PAA, drží CSA

DAC+PIA
nahráno z výroby



PAI drží výrobce,
podepisuje PAA

Párování zařízení a následná komunikace



Co potřebujeme pro Matter?

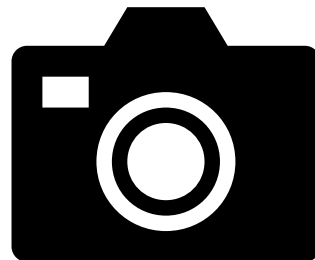
- Matter controller/hub
 - Fyzické zařízení:
 - Apple TV, Apple Home Pod, Google Home, Amazon Echo Dot, ...
 - Softwarová implementace:
 - <https://github.com/matter-js/python-matter-server> (archived 06/2026)
 - <https://github.com/matter-js/matterjs-server>
 - Párování, nastavení, ovládání, správa zařízení
- Nezávislost na výrobci zařízení



Co ještě potřebujeme?

- Mobil se zapnutým Bluetooth
 - QR kód ze zařízení nebo číselný kód
 - Aplikaci Matter controlleru/hubu
-

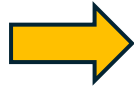
Stačí vyfotit...



QR kód podrobněji

- Vlastní base38 kódování s omezenou znakovou sadou

MT:GE.016P614GEOP4C910



```
# snap install chip-tool  
# chip-tool payload parse-setup-  
payload XXXX
```

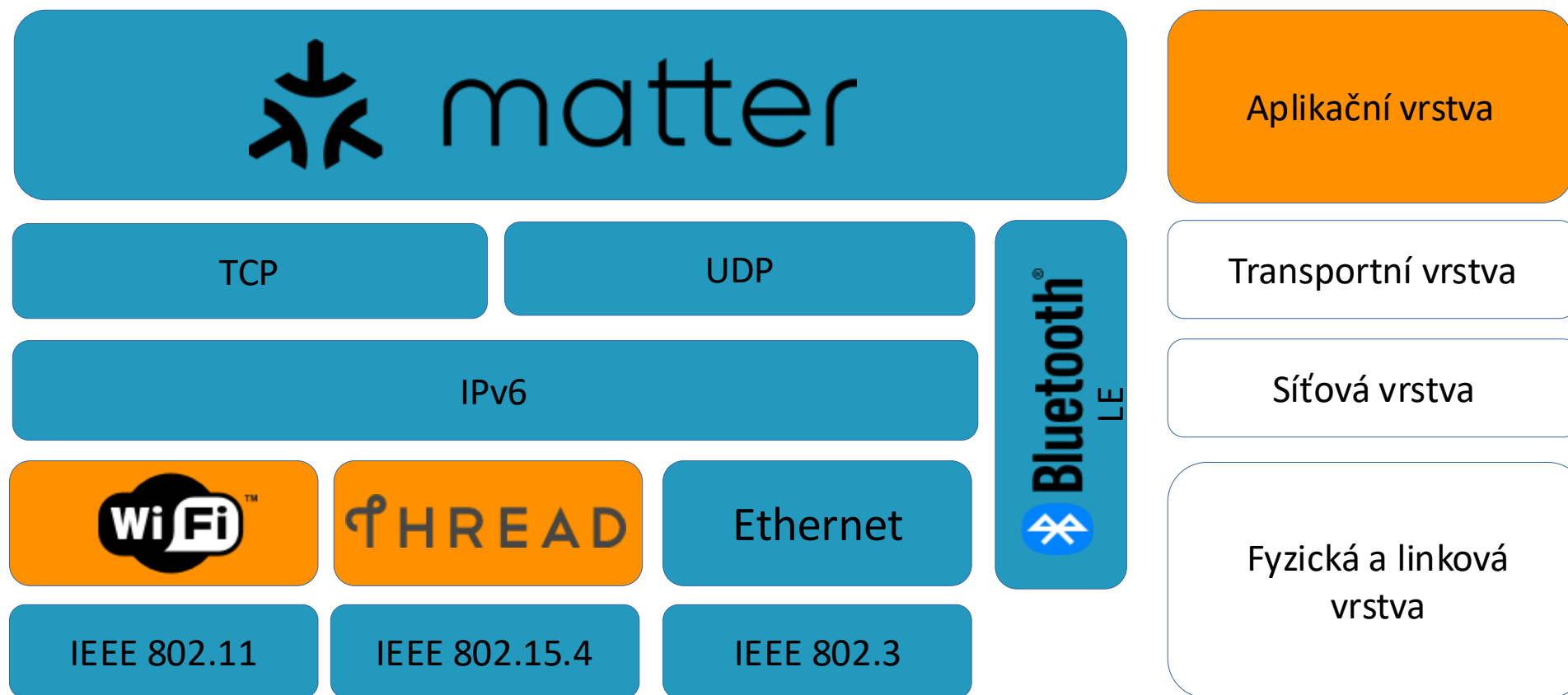
Pole	Bity	Pozice	Náš QR kód
Version	3	0-2	0
Vendor ID	16	3-18	5245
Product ID	16	19-34	1412
Custom Flow	2	35-36	0
Discovery Caps	8	37-44	0x02
Discriminator	12	45-56	740
Passcode	27	57-83	58935902

<https://tommie.github.io/tools/matter-qr-decoder>



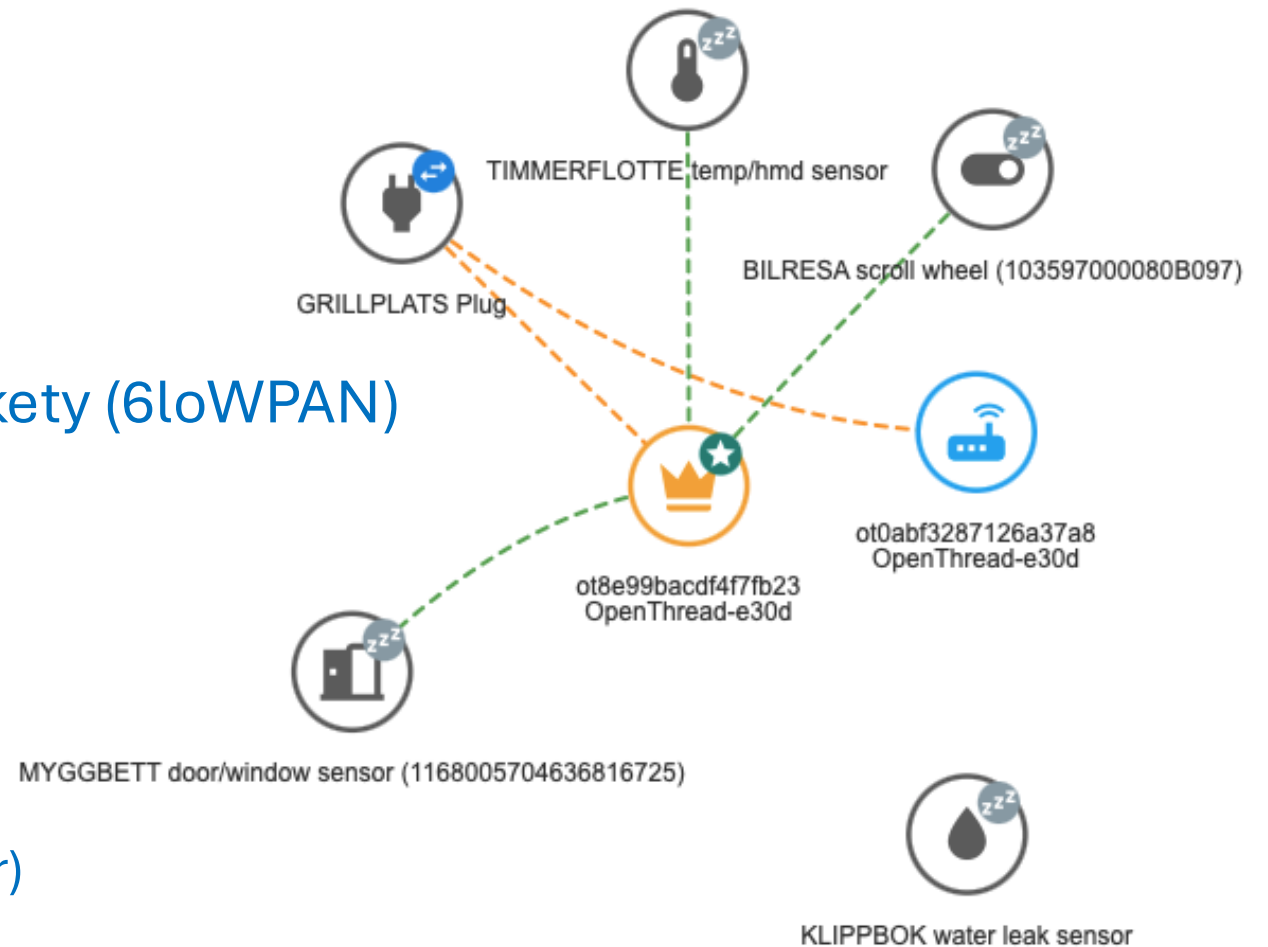
Najdete rozdíly?

Matter totiž neřeší komunikační protokol



Matter over ...?

- Wi-Fi
- Thread
 - Navrženo pro malé, úsporné pakety (6LoWPAN)
 - 2,4 GHz pásmo, pozor...
 - Vytváří mesh síť
 - Role:
 - Leader (1)
 - Router (až 32)
 - Koncové zařízení (až 511 per router)
 - Border router (=/ Leader)



Co potřebujeme pro Thread?

- OpenThread Border Router
 - Fyzické zařízení:
 - Apple TV, Apple Home Pod, Google Home, Amazon Echo Dot, ...
 - Softwarová implementace:
 - <https://github.com/openthread/ot-br-posix>
 - → <https://hub.docker.com/r/openthread/border-router>
 - USB dongle:
 - Home Assistant Connect ZBT-2
 - Sonoff Dongle Plus MG24 || starší ZBDongle-E



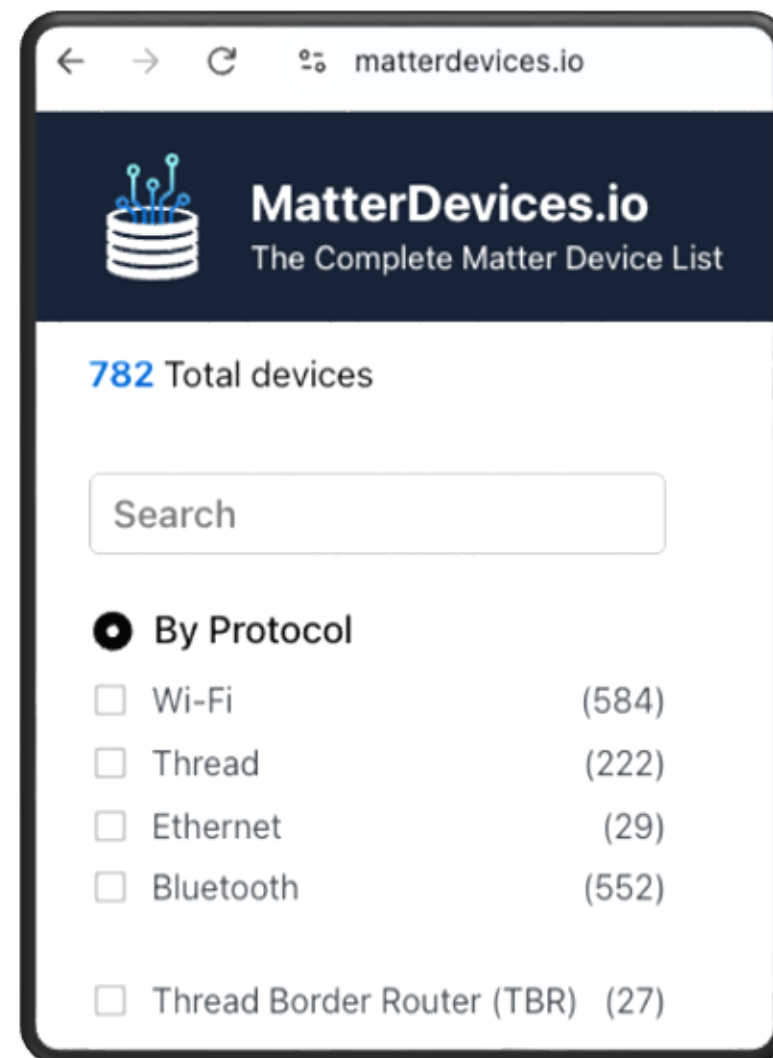
Matter server + OTBR

docker compose

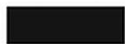
```
matter-server:
  container_name: matter-server
  image: ghcr.io/matter-js/matterjs-server:latest
  network_mode: host
  volumes:
    - /mnt/docker/matter:/data
  environment:
    - PRIMARY_INTERFACE=enol
  restart: always
```

```
otbr:
  container_name: otbr
  image: openthread/border-router:latest
  network_mode: host
  privileged: true
  environment:
    OT_RCP_DEVICE: "spinel+hdlc+uart:///dev/ttyUSB0?uart-
baudrate=460800"
    OT_INFRA_IF: "enol"
    OT_THREAD_IF: "wpan0"
  devices:
    - /dev/serial/by-id/usb-XXX:/dev/ttyUSB0
    - /dev/net/tun:/dev/net/tun
  volumes:
    - /mnt/docker/otbr:/data
  restart: always
```

Kde najít podporovaná zařízení?



Motivace se do toho pustit?



25. 6. 2026 0:56

Chápu to tedy dobře, že sice protokol jako takový je otevřený a svobodný, ale pokud si chci ubastlit vlastní senzor, tak mám smůlu, protože PKI and stuff?



zdroj: root.cz

Jasně, ESP32!

To je přeci jednoduché...

+ esp32 esphome openthread howto

Režim AI →

Hledat Googlem

Zkusím štěstí



Matter Alpha

<https://www.matteralpha.com> › e...



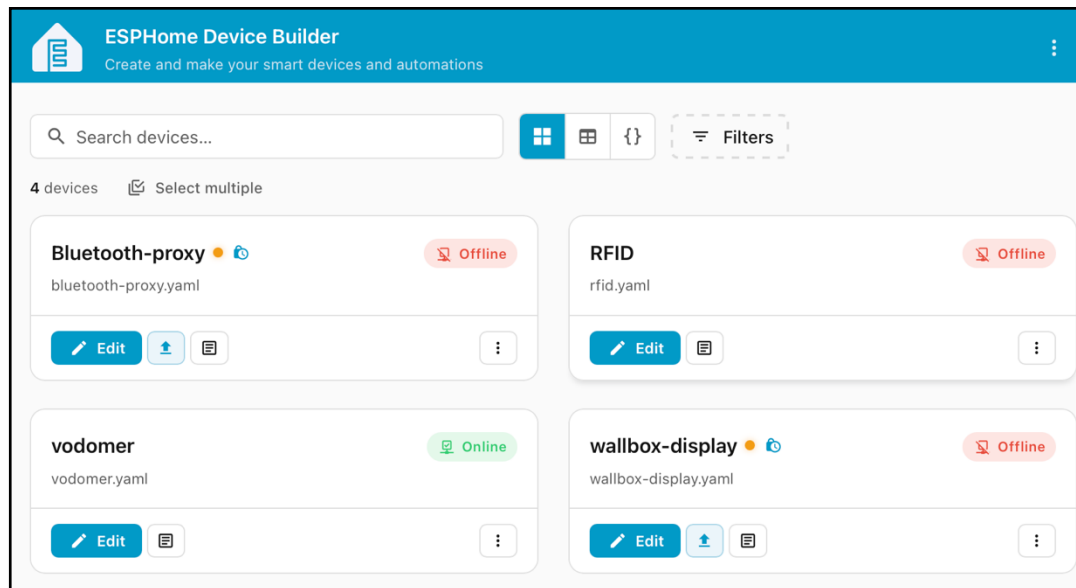
ESPHome 2025.6 adds OpenThread support

23. 6. 2025 — **Create a new profile in the web UI**, and flash the demo code (please replace the credentials with your own.) esphome: name: thread-c6

Co potřebujeme?

docker compose

- ESPHome
 - aktuální verze 2026.9.1 (30.9. 2026)



```
services:
  esphome:
    image: ghcr.io/esphome/esphome:stable
    container_name: esphome
    privileged: true
    network_mode: host
    environment:
      - ESPHOME_DASHBOARD_USE_PING=true
    volumes:
      - /mnt/docker/esphome/config:/config
      - /etc/localtime:/etc/localtime:ro
    restart: always
```

- ESP vývojovou desku

ESP vývojové desky pro Matter

	CPU	Wi-Fi 2.4GHz	BLE	Thread	USB	RAM
ESP32-C3	RISC-V 160Mhz	4	5	—	USB Serial/JTAG	~400 kB
ESP32-C5	RISC-V 240MHz	6 (+ 5GHz)	5	—	—	~384 kB
ESP32-C6	RISC-V 160Mhz	6	5	✓	USB Serial/JTAG	~512 kB
ESP32-H2	RISC-V 96Mhz	—	5	✓	USB Serial/JTAG	~320 kB
ESP32-S2	Xtensa 240MHz	4	—	—	USB OTG	~320 kB
ESP32-S3	Xtensa 240MHz (2c)	4	5	—	USB OTG	~512 kB

↙ Joint Test Action Group

- **USB Serial/JTAG** – podpora konzole/logy + debugging mikrokontroleru
- **USB OTG** – navíc USB host (klávesnice, myš, kamera, flash disk)
- **C5** potřebuje externí USB-UART a externí JTAG debugger

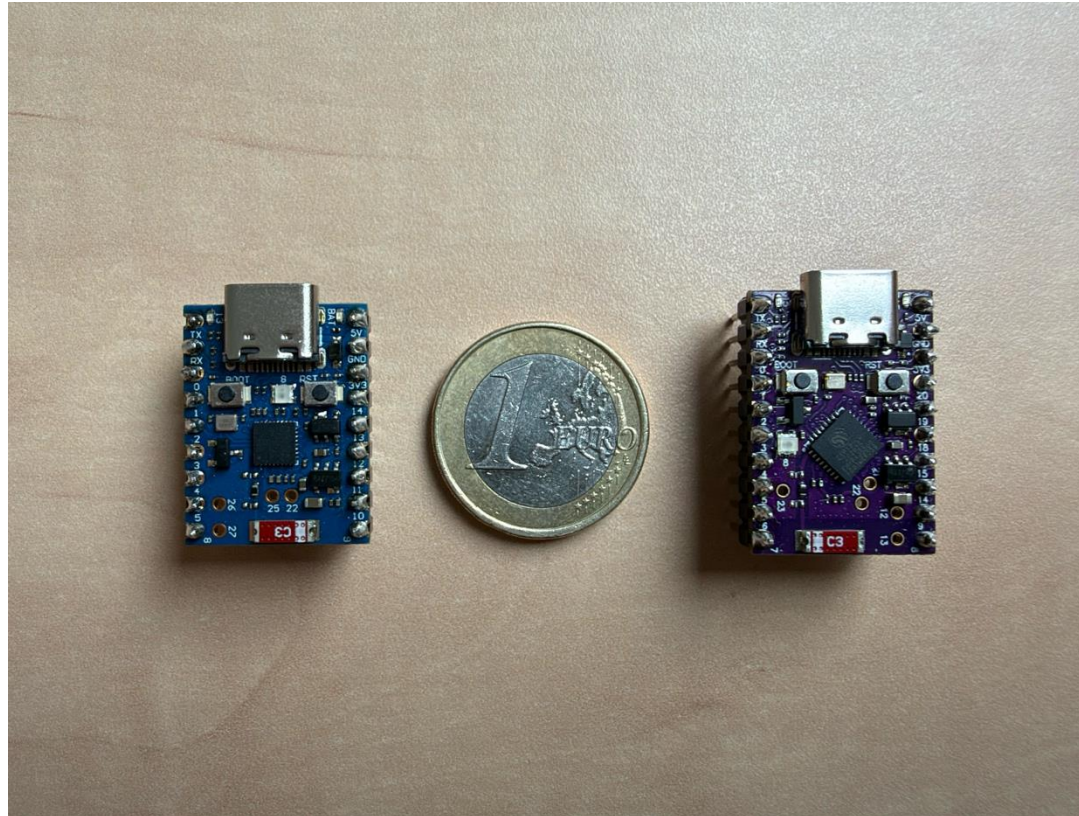
Vývojové desky pro testování

ESP32-H2 Super Mini

- nízká spotřeba
- pouze Thread

ESP32-C6 Super Mini

- univerzální
- Wi-Fi 6, Thread
- vyšší výkon



cca 22x18mm, ±\$5.00

ESPHome - sensor s BME280

- Teplota, vlhkost, tlak
- Google Chrome! (Web Serial API)
- Create device → Create new project → Search boards... ESP32-C6 → Generic ESP32-C6 Board → Select
- Název: **esp32c6-test**
- Kontrola:

```
esp32:  
  variant: esp32c6  
  framework:  
    type: esp-idf
```



ESPHome - sensor s BME280

- Přidat pro OpenThread

```
network:  
    enable_ipv6: true  
  
openthread:  
    device_type: FTD  
    tlv: !secret thread_dataset_tlv
```



- Odstranit Wi-fi

```
wifi:  
    captive_portal:
```

Active dataset z Open Thread Borderu

Home Assistant

OpenThread-e30d

1 hraniční router

OT

OpenThread BorderRouter #FB23
ot8e99bacdf4f7fb23.local.

i



×

OpenThread-e30d

Network name: OpenThread-e30d

Channel: 18

Dataset ID: 01K [REDACTED]

PAN ID: e30d

Extended PAN ID: b48d043c81e1b488

OTBR URL: http://192.168.100.250:8081

Active dataset TLVs:

0e0 [REDACTED]

ESPHome - sensor s BME280

- Standardní konfigurace sensorů

i2c:

sda: GPIO20

scl: GPIO19

scan: true

sensor:

- platform: bme280_i2c

address: 0x76

temperature:

name: "Thread Teplota"

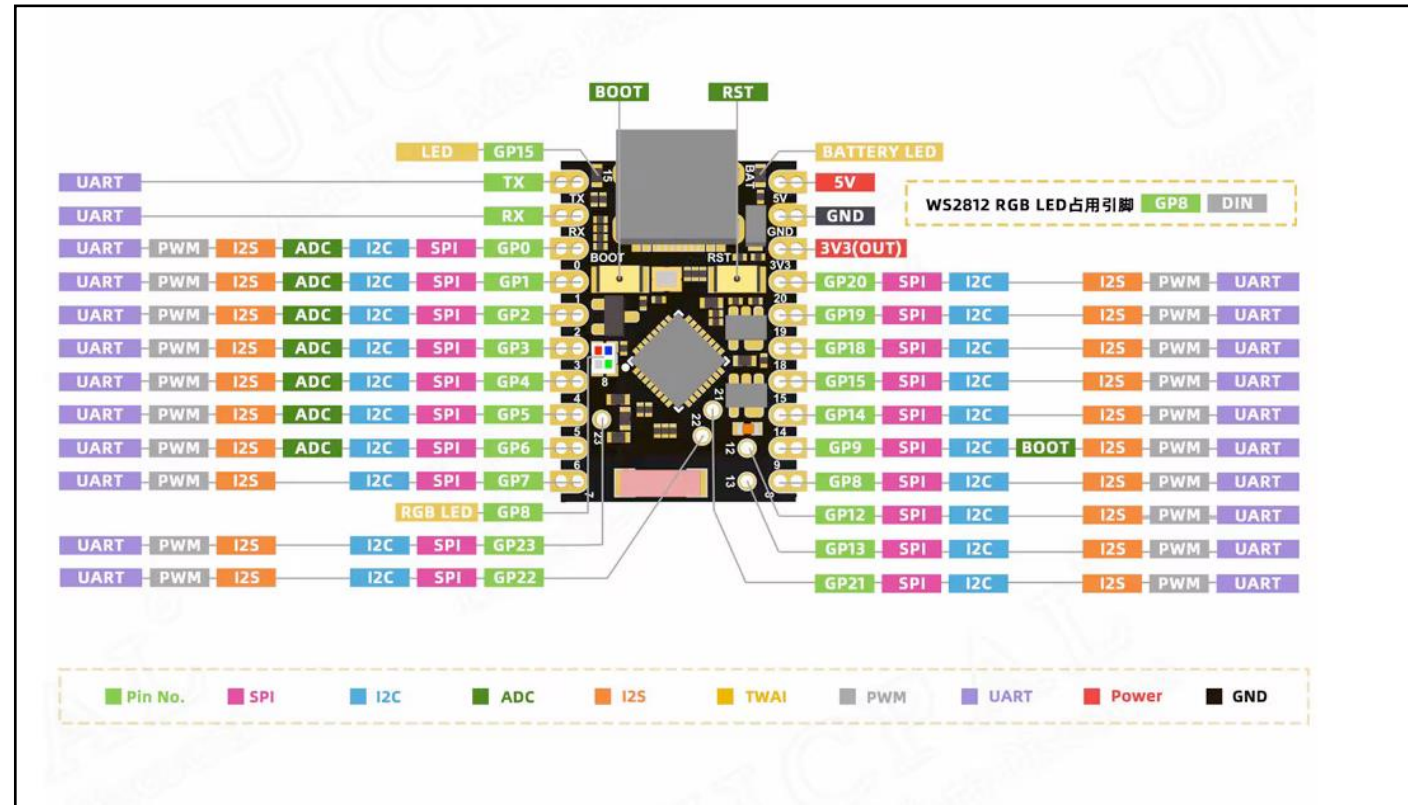
humidity:

name: "Thread Vlhkost"

pressure:

name: "Thread Tlak"

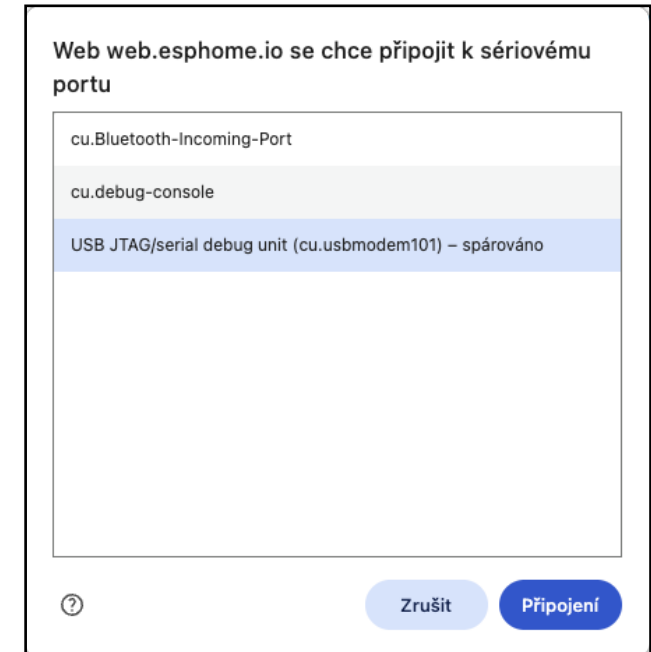
update_interval: 30s



zdroj: <https://www.espboards.dev/esp32/esp32-c6-super-mini/>

ESPHome - sensor s BME280

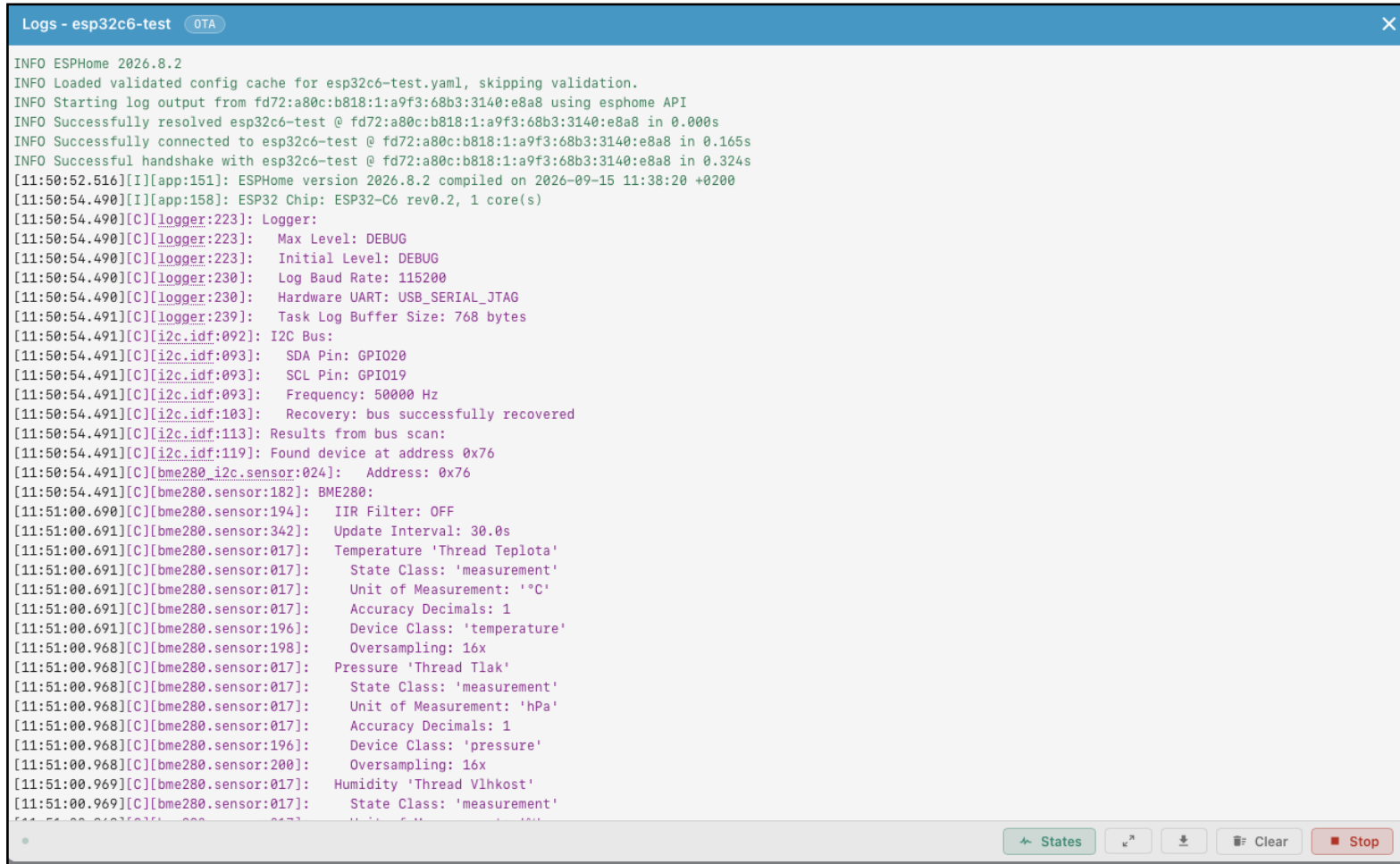
- Tlačítko *Install* → *Plug into this computer*
 - Zkompiluje se firmware
 - ESP-IDF (zastaralá verze 5.5.5), cmake, ninja, riscv32-esp-elf, esp-rom-elfs
- Tlačítko *Open USB flasher* → web web.esphome.io → tlačítko *Connect & install*, vybrat USB JTAG → Tlačítko *Nainstalovat*
- Po nahrání FW se automaticky otevře LOG konzole



Memory Type/Section	Used [bytes]	Used [%]	Remain [bytes]	Total [bytes]
Flash Code	750322			
.text	613846			
.rodata	135976			
.appdesc	256			
.init_array	244			
DIRAM	109932	24.32	342180	452112
.text	54724	12.1		
.bss	44208	9.78		
.data	10904	2.41		
.noinit	96	0.02		
LP SRAM	24	0.15	16360	16384
.rtc_reserved	24	0.15		

Total image size: 815706 bytes (.bin may be padded larger)
RAM: [==] 24.3% (used 109932 bytes from 452112 bytes)
Flash: [====] 44.5% (used 815706 bytes from 1835008 bytes)
INFO Creating factory.bin...
INFO Created: /config/.esphome/build/esp32c6-test/build/firmware.factory.bin
INFO Created: /config/.esphome/build/esp32c6-test/build/firmware.ota.bin
INFO Created: /config/.esphome/build/esp32c6-test/build/firmware.elf

ESPHome - sensor s BME280

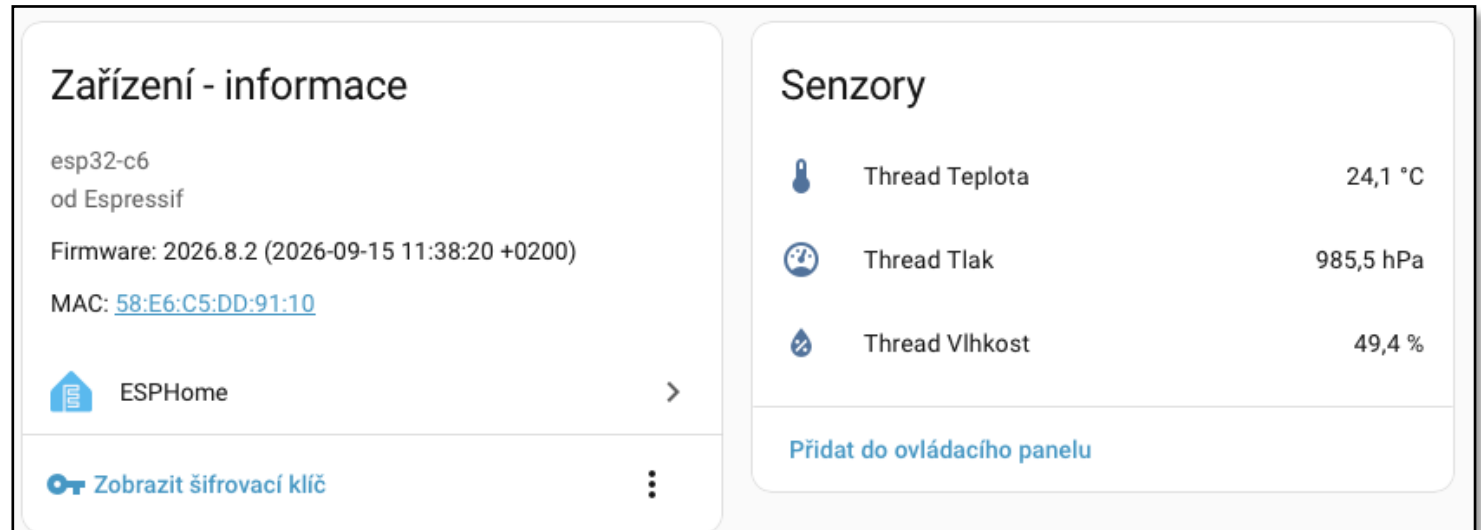
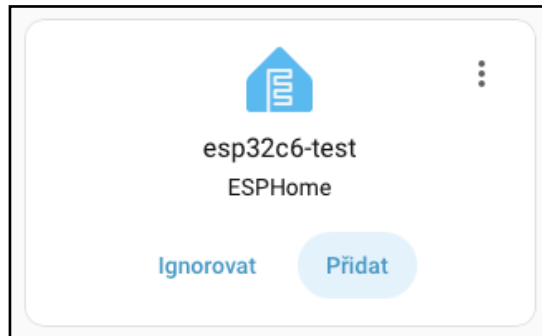


```
Logs - esp32c6-test OTA
INFO ESPHome 2026.8.2
INFO Loaded validated config cache for esp32c6-test.yaml, skipping validation.
INFO Starting log output from fd72:a80c:b818:1:a9f3:68b3:3140:e8a8 using esphome API
INFO Successfully resolved esp32c6-test @ fd72:a80c:b818:1:a9f3:68b3:3140:e8a8 in 0.000s
INFO Successfully connected to esp32c6-test @ fd72:a80c:b818:1:a9f3:68b3:3140:e8a8 in 0.165s
INFO Successful handshake with esp32c6-test @ fd72:a80c:b818:1:a9f3:68b3:3140:e8a8 in 0.324s
[11:50:52.516][I][app:151]: ESPHome version 2026.8.2 compiled on 2026-09-15 11:38:20 +0200
[11:50:54.490][I][app:158]: ESP32 Chip: ESP32-C6 rev0.2, 1 core(s)
[11:50:54.490][C][logger:223]: Logger:
[11:50:54.490][C][logger:223]:   Max Level: DEBUG
[11:50:54.490][C][logger:223]:   Initial Level: DEBUG
[11:50:54.490][C][logger:230]:   Log Baud Rate: 115200
[11:50:54.490][C][logger:230]:   Hardware UART: USB_SERIAL_JTAG
[11:50:54.491][C][logger:239]:   Task Log Buffer Size: 768 bytes
[11:50:54.491][C][i2c.idf:092]: I2C Bus:
[11:50:54.491][C][i2c.idf:093]:   SDA Pin: GPIO20
[11:50:54.491][C][i2c.idf:093]:   SCL Pin: GPIO19
[11:50:54.491][C][i2c.idf:093]:   Frequency: 50000 Hz
[11:50:54.491][C][i2c.idf:103]:   Recovery: bus successfully recovered
[11:50:54.491][C][i2c.idf:113]: Results from bus scan:
[11:50:54.491][C][i2c.idf:119]: Found device at address 0x76
[11:50:54.491][C][bme280_i2c.sensor:024]:   Address: 0x76
[11:50:54.491][C][bme280.sensor:182]: BME280:
[11:51:00.690][C][bme280.sensor:194]:   IIR Filter: OFF
[11:51:00.691][C][bme280.sensor:342]:   Update Interval: 30.0s
[11:51:00.691][C][bme280.sensor:017]:   Temperature 'Thread Teplota'
[11:51:00.691][C][bme280.sensor:017]:     State Class: 'measurement'
[11:51:00.691][C][bme280.sensor:017]:     Unit of Measurement: '°C'
[11:51:00.691][C][bme280.sensor:017]:     Accuracy Decimals: 1
[11:51:00.691][C][bme280.sensor:196]:   Device Class: 'temperature'
[11:51:00.968][C][bme280.sensor:198]:   Oversampling: 16x
[11:51:00.968][C][bme280.sensor:017]:   Pressure 'Thread Tlak'
[11:51:00.968][C][bme280.sensor:017]:     State Class: 'measurement'
[11:51:00.968][C][bme280.sensor:017]:     Unit of Measurement: 'hPa'
[11:51:00.968][C][bme280.sensor:017]:     Accuracy Decimals: 1
[11:51:00.968][C][bme280.sensor:196]:   Device Class: 'pressure'
[11:51:00.968][C][bme280.sensor:200]:   Oversampling: 16x
[11:51:00.969][C][bme280.sensor:017]:   Humidity 'Thread Vlhkost'
[11:51:00.969][C][bme280.sensor:017]:     State Class: 'measurement'
```

Ukázka ještě z verze 2026.8.2

Integrace s Home Assistant

- Pro přidání zařízení šifrovací klíč
 - V ESPHome, sekce api: → encryption: key:



Ale tohle není Matter!

Ale tohle není Matter!

Tohle je přece ESP32 na Thread síti!

Takže znovu...

Příprava vývojového prostředí

- Git, Python > 3.10 a další nástroje

```
$ brew install python cmake ninja gn
$ brew link --overwrite python@3.14
$ echo 'export
    PATH="/opt/homebrew/opt/python@3.14/libexec/bin:$PATH"'
    >> ~/.zprofile
$ source ~/.zprofile
```



- Espressif IoT Development Framework (ESP-IDF)

- Oficiální SDK přímo od výrobce čipů
- Kompilátor/debugger, build systém (CMake), FreeRTOS (4-9kB), ovladače periférií, síťové stacky, system utilities (OTA, power management, ...)

1. část: ESP-IDF instalace

- Espressif IoT Development Framework (ESP-IDF)
 - <https://github.com/espressif/esp-idf>
 - aktuální verze **6.1** (27.8. 2026)

```
$ mkdir ~/esp; cd ~/esp
```

```
$ git clone -b v6.1 -recursive https://github.com/espressif/esp-idf.git esp-idf-v6.1
```

```
$ ~/esp/esp-idf-v6.1/install.sh
```

Potřebujeme i submoduly



ESP-IDF příprava

Před každým builděním!



```
$ source ~/esp/esp-idf-v6.1/export.sh
Checking "python3" ...
Checking "python" ...
Python 3.14.7
"python" has been detected
Activating ESP-IDF 6.1
Setting IDF_PATH to '/Users/vaclavsteiner/esp/esp-idf-v6.1'.
* Checking python version ... 3.14.7
* Checking python dependencies ... OK
* Deactivating the current ESP-IDF environment (if any) ... OK
* Establishing a new ESP-IDF environment ... OK
* Identifying shell ... zsh
* Detecting outdated tools in system ... OK - no outdated tools found
* Shell completion ... Autocompletion code generated

Done! You can now compile ESP-IDF projects.
```

2. část: ESP-Matter

- Espressif's SDK for Matter
 - podpora CHIP (Connected Home over IP) → multiplatformní referenční implementace Matter protokolu. Velice nízkoúrovňový a nepohodlný.
 - zjednodušené API (C++) pro Matter endpointy a clustery
 - ukázkové příklady
 - build skripty
 - přímá integraci s ESP-IDF
 - požadovaná provázanost verzí
- **Výhoda:** velké zjednodušení, žádné složité portování CHIP na ESP, vlastní podpory vývojových desek apod.

ESP-Matter instalace

Ne rekurzivně!



```
$ cd ~/esp
$ git clone --depth 1 https://github.com/espressif/esp-matter.git
$ cd es-matter
$ git submodule update --init --depth 1
• Jen ESP32 platformu a nástroje pro daný OS
$ cd connectedhomeip/connectedhomeip
$ ./scripts/checkout_submodules.py --platform esp32 darwin --shallow
[{ameba, android, asr, bl602, bouffalolab, cc13xx_26xx, cc32xx, darwin, esp32, genio, infineon, linux, nrfconnect, nuttx, nxp, qpg, realtek, rw61x, silabs, silabs_docker, stm32, telink, tizen, unit_tests, webos} ...]
```

ESP-Matter instalace

```
$ ~/esp/esp-matter/install.sh
```

- případně

```
$ ~/esp/esp-matter/install.sh --no-host-tool
```

- neinstaluje nástroje chip-tool apod. vhodné pro debug

- Nějaký prostor to zabírá...

```
$ du -hs ~/esp/*
```

```
2,7G  esp-idf-v6.1
```

```
6,6G  esp-matter
```

Před každým builděním!



```
$ source ~/esp/esp-matter/export.sh
```

```
Setting ESP_MATTER_PATH to '/Users/vaclavsteiner/esp/esp-matter'
```

Je libo GUI nástroje?

- Arduino IDE
 - Matter over Wifi
 - Matter over Thread (via ESP-IDF)
- VSCode s ESP-IDF
 - Rozšíření ESP-IDF od Espressif
 - ESP-IDF Installation Manager (<https://dl.espressif.com/dl/eim/>)
 - ~/Library/Application\ Support/Code/User/settings.json

```
"idf.espIdfPath": "~/.esp/esp-idf",  
"idf.toolsPath": "~/.espressif",  
"idf.pythonBinPath":  
"/.opt/homebrew/opt/python@3.14/libexec/bin/python"
```
 - Cmd(Ctrl)+Shift+P
 - ESP-IDF: Set Espressif Device Target | ESP-IDF: Build your Project
 - ESP-IDF: Flash your Project | ESP-IDF: Monitor Device



První Matter projekt

Light

Struktura Matter projektu

```
muj_projekt/  
├── main/  
│   ├── CMakeLists.txt      ← podružný build skript  
│   ├── app_main.cpp        ← hlavní aplikace  
│   ├── app_driver.cpp      ← ovládání HW  
│   ├── app_priv.h          ← privátní deklarace  
│   └── idf_component.yml    ← případné další závislosti  
├── CMakeLists.txt          ← hlavní build skript  
├── partitions.csv          ← rozvržení flash paměti  
├── sdkconfig.defaults      ← společná, obecná nastavení  
├── sdkconfig.defaults.<target> ← nastavení per čip (např. esp32-c6, esp32-h2)  
└── README.md
```

První build Matter projektu

- Ukázkový příklad **light**

```
$ cd ~/esp/esp-  
matter/examples/light  
  
$ idf.py set-target esp32c6  
  
$ idf.py build  
  
$ #idf.py erase-flash  
  
$ idf.py -p /dev/cu.usbmodem101  
flash monitor
```

```
Connected to ESP32-C6 on /dev/cu.usbmodem101:  
Chip type:          ESP32-C6FH4 (QFN32) (revision v0.2)  
Features:           Wi-Fi 6, BT 5 (LE), IEEE802.15.4,  
Single Core + LP Core, 160MHz, Embedded Flash 4MB  
Crystal frequency:  40MHz  
USB mode:           USB-Serial/JTAG  
MAC:                58:e6:c5:ff:fe:dd:91:10  
BASE MAC:           58:e6:c5:dd:91:10  
MAC_EXT:            ff:fe
```

```
Stub flasher running.  
Changing baud rate to 460800...  
Changed.
```

```
Configuring flash size...
```

```
Writing 'bootloader/bootloader.bin' at 0x00000000...
```

První build Matter projektu

- Ukázkový příklad **light**

```
$ cd ~/esp/esp-  
matter/examples/light  
  
$ idf.py set-target esp32c6  
  
$ idf.py build  
  
$ #idf.py erase-flash  
  
$ idf.py -p /dev/cu.usbmodem101  
flash monitor
```



Matter over Wi-Fi

```
W (1313) wifi:Haven't to connect to a suitable AP now!
```

```
Connected to ESP32-C6 on /dev/cu.usbmodem101:  
Chip type:          ESP32-C6FH4 (QFN32) (revision v0.2)  
Features:           Wi-Fi 6, BT 5 (LE), IEEE802.15.4,  
Single Core + LP Core, 160MHz, Embedded Flash 4MB  
Crystal frequency:  40MHz  
USB mode:           USB-Serial/JTAG  
MAC:                58:e6:c5:ff:fe:dd:91:10  
BASE MAC:           58:e6:c5:dd:91:10  
MAC_EXT:            ff:fe  
  
Stub flasher running.  
Changing baud rate to 460800...  
Changed.  
  
Configuring flash size...  
  
Writing 'bootloader/bootloader.bin' at 0x00000000...
```

První problém

- Debugging konzole nereaguje na klávesnici...

- Přidat do sdkconfig.defaults.esp32c6

```
CONFIG_ESP_CONSOLE_USB_SERIAL_JTAG=y
```

```
$ rm sdkconfig
```

```
$ idf.py set-target esp32c6
```

```
$ idf.py build flash monitor
```

Nebo také pomocí

```
$ idf.py menuconfig
```

Generuje nový sdkconfig

Ve skutečnosti se volá

```
$ idf.py -D
```

```
SDKCONFIG_DEFAULTS=„sdkconfig.defaults; sdkconfig.defaults.esp32c6“
```

Můžeme pokračovat

- Konzole už reaguje na klávesnici 😊

```
> matter esp wifi connect SSID PASS
```

```
...
```

```
I (40512) chip[DL]: IPv4 address changed on WiFi station interface:  
192.168.100.65/255.255.255.0 gateway 192.168.100.1
```

```
I (40512) chip[DL]: IPv4 Internet connectivity ESTABLISHED
```

Můžeme pokračovat

- Konzole už reaguje na klávesnici 😊

```
> matter esp wifi connect SSID PASS
```

```
...
```

```
I (40512) chip[DL]: IPv4 address changed on WiFi station interface:  
192.168.100.65/255.255.255.0 gateway 192.168.100.1
```

```
I (40512) chip[DL]: IPv4 Internet connectivity ESTABLISHED
```

```
> matter onboardingcodes ble qrcode
```

```
MT:Y.K9042C00KA0648G00
```



<https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT:Y.K9042C00KA0648G00>

```
Done
```

```
> matter onboardingcodes ble manualpairingcode  
34970112332
```

```
Done
```

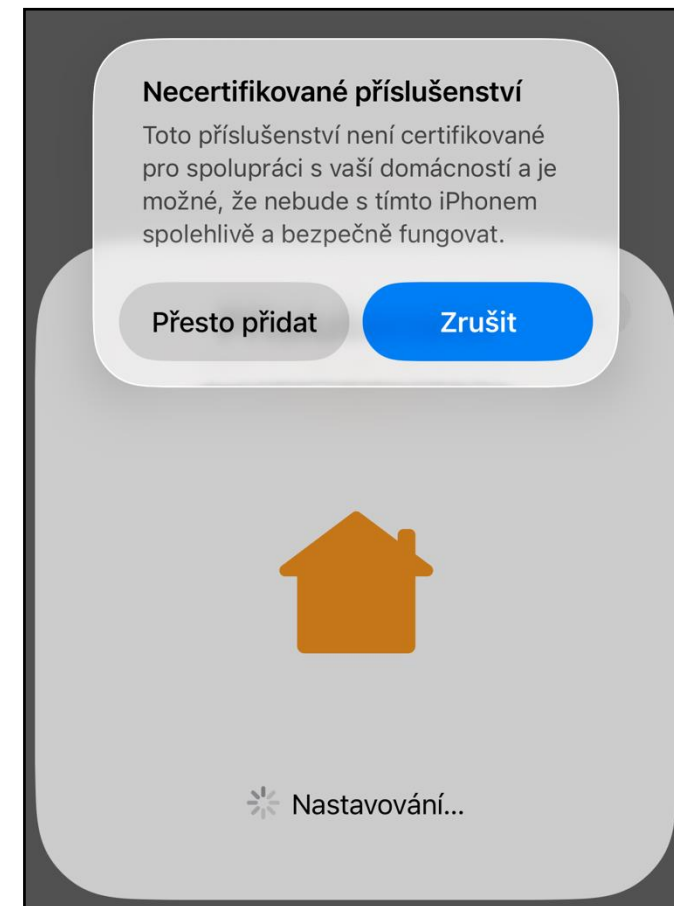
Můžeme pokračovat

Od verze 2026.9

- Ted' už stačí načíst QR kód mobilem
 - nativní aplikace (na iOS app Domácnost)
 - Home Assistant → Nastavení → Připojení → Matter → Přidat zařízení
- Nebo vložit PIN zařízení ručně

Chybové hlášení je očekávané...

- Do iOS app Domácnost úspěšně přidáno
 - via Apple TV



Druhý problém

- Do Home Assistant nepřidáno 😞
 - Logy z ESP-IDF konzole? ❌
 - Logy z Matter serveru



```
2026-09-17 10:07:00.997 INFO    Controller~ndHandler Attestation finding (error):
TrustedAsTestCertificate PAA 6afd22771f511fecbf1641976710dc31a1717e is a test/development
certificate; device presented a valid test attestation but production trust policy is in effect
2026-09-17 10:07:00.997 INFO    Controller~ndHandler Attestation finding (info):
PaaTrustStoreTimeMismatch PAA was fetched at 2026-07-10T03:09:02.100Z which is after DAC
notBefore 2022-02-05T00:00:00.000Z; cannot confirm PAA was in DCL when DAC was issued
2026-09-17 10:07:00.998 INFO    Controller~ndHandler Attestation finding (info):
CertificationTypeTest Certification Declaration has certification_type=0 (test)
2026-09-17 10:07:00.998 NOTICE Controller~ndHandler Attestation rejected: This device uses a
test/development certificate. To commission it, enable the "Test DCL" option in the settings –
only do this if you trust the vendor.
```

Úprava Matter serveru

- v docker-compose povolíme testovací zařízení

`environment:`

- `- ENABLE_TEST_NET_DCL=true`

- v Home Assistant konečně dostupné



Informace o Matter		^
ID uzlu:	61	
Typ sítě:	Wi-Fi	
Typ zařízení:	Koncové zařízení	
Název sítě:		
MAC adresa:	58:e6:c5:dd:91:10	
IP adresy:	fe80::5ae6:c5ff:fedd:9110	
	fdb4:8d04:3c81:b488:5ae6:c5ff:fedd:9110	
	192.168.100.65	



A co Matter over Thread?

- Stačí upravit sdkconfig.defaults.esp32c6

```
# Enable OpenThread
CONFIG_OPENTHREAD_CONSOLE_TYPE_USB_SERIAL_JTAG=y
CONFIG_OPENTHREAD_ENABLED=y
CONFIG_OPENTHREAD_SRP_CLIENT=y
CONFIG_OPENTHREAD_LOG_LEVEL_DYNAMIC=n
CONFIG_OPENTHREAD_LOG_LEVEL_NOTE=y
CONFIG_OPENTHREAD_CLI=n
```

```
# LwIP config for OpenThread
CONFIG_LWIP_IPV6_NUM_ADDRESSES=8
CONFIG_LWIP_MULTICAST_PING=y
```

```
# MDNS platform
CONFIG_USE_MINIMAL_MDNS=n

# Disable STA for ESP32C6
CONFIG_ENABLE_WIFI_STATION=n
CONFIG_ENABLE_WIFI_AP=n
```

```
$ rm sdkconfig
$ idf.py set-target esp32c6
$ idf.py erase-flash
$ idf.py build flash monitor
```

Opravdu Matter over Thread!

Zařízení - informace

TEST_PRODUCT (32768)
od TEST_VENDOR

Firmware: 1.0

Hardware: TEST_VERSION

Informace o Matter

ID uzlu: 63

Typ sítě: Thread

Typ zařízení: Koncové zařízení

Název sítě: OpenThread-e30d

MAC adresa: 22:c8:25:46:d6:8b:74:a6

IP adresy: fd72:a80c:b818:1:2743:c78:a949:3073

 Matter

 Sdílet zařízení

Ovládací prvky

 TEST_PRODUCT ☒

[Přidat do ovládacího panelu](#)

Nastavení

 Firmware Aktuální

 Chování při zapnutí
Předchozí stav

 Úroveň při zapnutí 64

 Úroveň při zapnutí 64

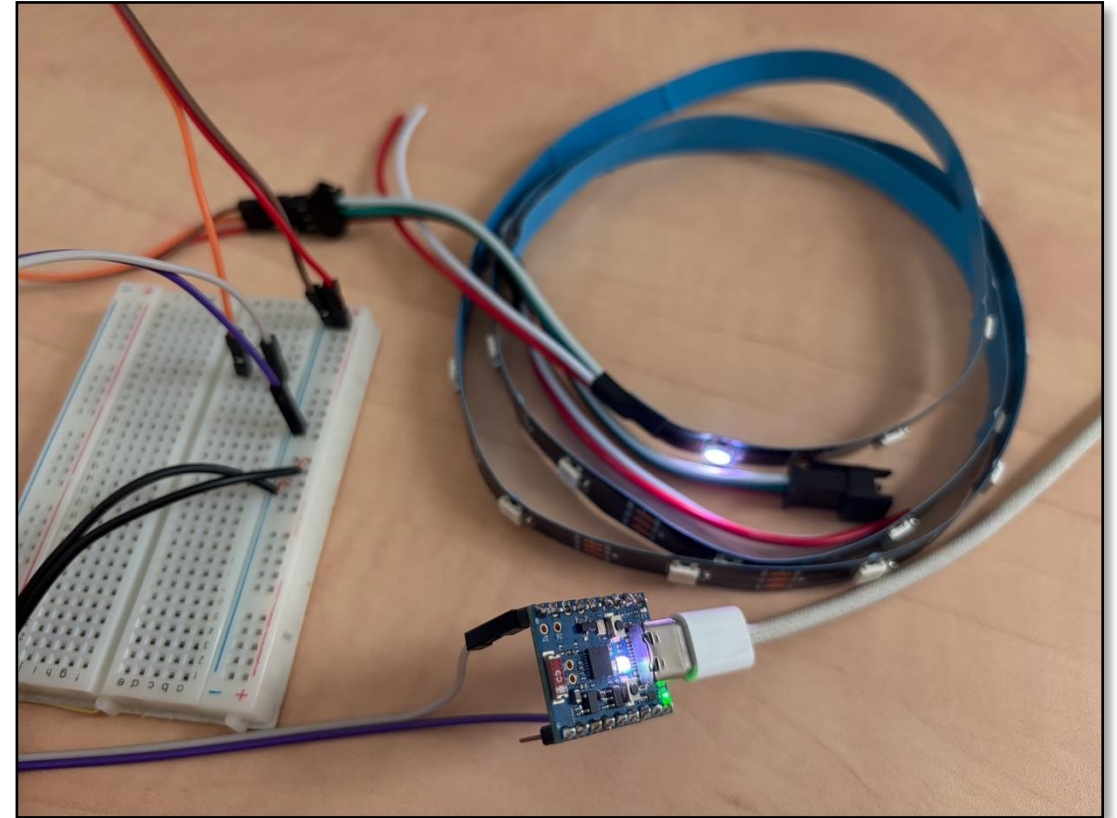
Pojďme zkusit rozšíření

LED pásek

ESP32-H2 + LED pásek WS2812

- RGB pásek, externí 5V napájení
- Řídící vodič pásky připojit k GPIO 8
- Stejný Matter příklad **light**

```
$ rm sdkconfig  
$ idf.py set-target esp32h2  
$ idf.py erase-flash  
$ idf.py build flash monitor
```

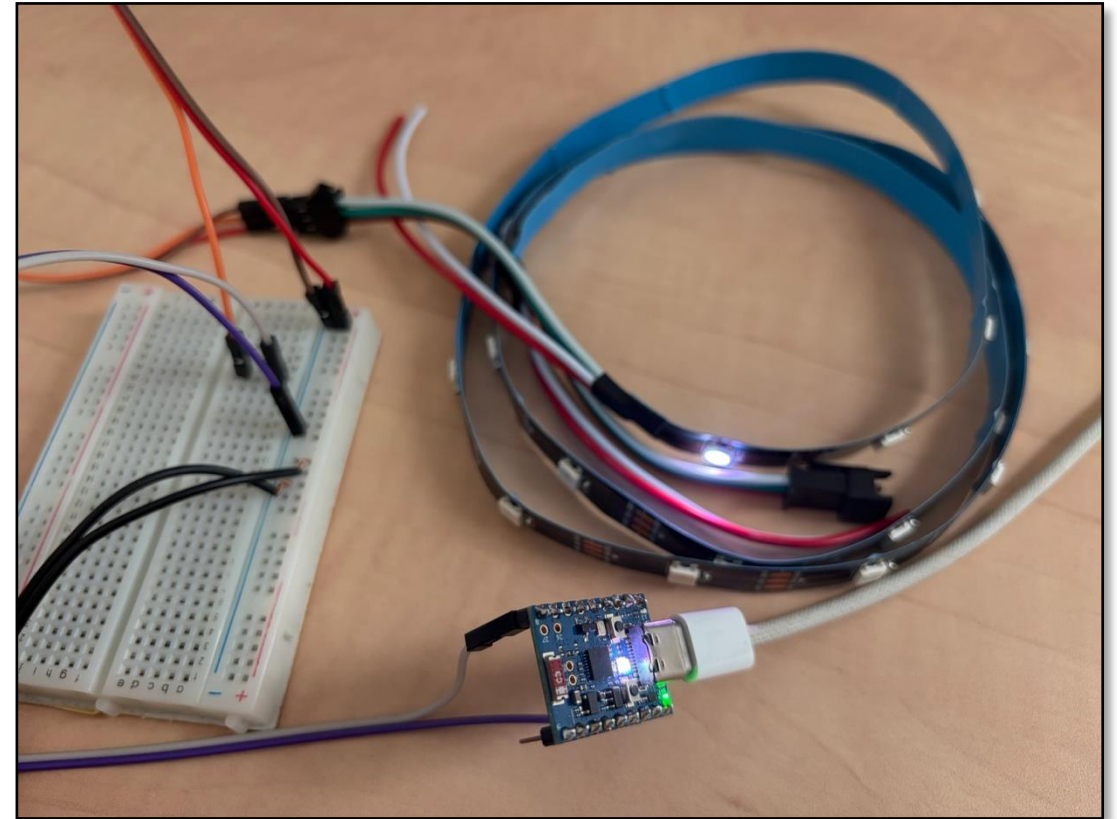


ESP32-H2 + LED pásek WS2812

- RGB pásek, externí 5V napájení
- Řídící vodič pásky připojit k GPIO 8
- Stejný Matter příklad **light**

```
$ rm sdkconfig  
$ idf.py set-target esp32h2  
$ idf.py erase-flash  
$ idf.py build flash monitor
```

- **Problém:**
 - svítí jen jedna LED ze třiceti ☹️



LED pásek - úpravy

```
~/esp/esp-matter/device_hal/led_driver/backends/ws2812.c
```

```
+ #define STRIP_LED_COUNT 30
```

```
...
```

```
esp_err_t led_driver_backend_create(const led_driver_config_t *config,  
led_driver_backend_t *backend)
```

```
{
```

```
    const led_strip_config_t strip_config = {
```

```
        .strip_gpio_num = config->gpio,
```

```
        - .max_leds = 1,
```

```
        + .max_leds = STRIP_LED_COUNT,
```

```
        .color_component_format = LED_STRIP_COLOR_COMPONENT_FMT_GRB,
```

```
        .led_model = LED_MODEL_WS2812,
```

```
        .flags.invert_out = config->output_invert,
```

```
    };
```

LED pásek - úpravy

```
esp_err_t led_driver_backend_render(led_driver_backend_t *backend, uint16_t hue, uint8_t
saturation, uint8_t brightness)
{
    led_strip_handle_t strip = (led_strip_handle_t)backend->handle;
    - esp_err_t err = led_strip_set_pixel_hsv(strip, 0, hue, saturation, brightness);
    + esp_err_t err;
    + for (int i = 0; i < STRIP_LED_COUNT; i++) {
        + err = led_strip_set_pixel_hsv(strip, i, hue, saturation, brightness);
        if (err != ESP_OK) {
            return err;
        }
    +}
    return led_strip_refresh(strip);
}
```

Pozor na velmi aktivní vývoj

- Co fungovalo v srpnu, nefunguje v září...
- Dříve `device_hal/led_driver/ws2812/led_driver.c`
- Komponenta `led_strip` špatně pracovala s RGB. WS2812 je GRB.
 - `color_component_format = LED_STRIP_COLOR_COMPONENT_FMT_GRB`
- Kompletní refaktoring ovladače
- Od 14.8. – 15.9. ~70 commitů

```
commit 4b6af289859e15f63091dfff425a042e32f05015f (grafted)
Author: Shu Chen <chenshu@espressif.com>
Date:   Tue Sep 8 08:24:32 2026 +0000

    Merge branch 'led-driver-refactor' into 'main'

    device_hal/led_driver: refactor driver codes

    See merge request app-frameworks/esp-matter!1659
```

A ještě jedou BME 280

Matter over Thread

Matter BME280

Stejné zapojení ESP32-C6 + BME280
SDA → GPIO 20, SCL → GPIO 19, I2C_ADDR → 0x76

- Ukázkový příklad sensors (shtc3 a pir pomocí i2c)

```
$ cd ~/esp/esp-matter/examples
```

```
$ cp -avR sensors bme280
```

- Velké množství (22) závislostí (button, led, strip, ...), odstranit z main/idf_component.yml

```
esp_bsp_generic:
```

```
  version: ^3
```

- Odstranit z main/app_main.cpp volání factory_reset_button_register() +

```
#include <app_reset.h>
```

- Odstranit z /CMakeLists.txt řádek "\${ESP_MATTER_PATH}/examples/common"

- Přidat knihovnu pro bme280

```
$ idf.py add-dependency "espressif/bme280=*"
```

```
$ idf.py reconfigure
```

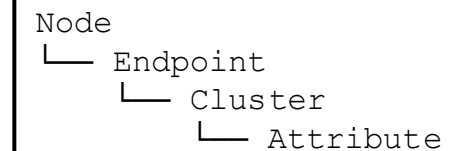
 Přidá se do main/idf_component.yml

Matter BME280 - další úpravy

```
|— main/  
|— managed_components/  
|   |— espressif__bme280/  
|   |   |— include/bme280.h  
|   |— espressif__i2c_bus/  
|   |   |— include/i2c_bus.h
```

- Ukázkový ovladač *shtc3* nelze použít
 - převzít strukturu callbacků a timerů
 - smazat `main/drivers/pir.{cpp,h}` a `main/drivers/shtc3.{cpp,h}`
 - podle *bme280.h* a *i2c_bus.h* vlastní kód `drivers/bme280_matter.{cpp,h}`
- Z `main/app_main.cpp` inspirace funkcí `temp_sensor_notification()`
- **Problém:** existující funkce pro zápis atributů (teplota, ...) vracely chyby
 - `err: 262 (ESP_ERR_NOT_SUPPORTED)`
 - Matter stack to odmítá přijmout. Proč?
→ Původní logika převzatá ještě z Zigbee ZCL

Pomohl(a) Claude AI 😊



Atributy - původní logika

```
static void temp_sensor_notification(uint16_t endpoint_id, float temp, void *user_data)
{
    chip::DeviceLayer::SystemLayer().ScheduleLambda([endpoint_id, temp]() {
        attribute_t * attribute = attribute::get(endpoint_id,
            TemperatureMeasurement::Id,
            TemperatureMeasurement::Attributes::MeasuredValue::Id);

        esp_matter_attr_val_t val;
        attribute::get_val(attribute, &val);
        val.val.i16 = static_cast<int16_t>(temp * 100);

        attribute::update(endpoint_id, TemperatureMeasurement::Id,
            TemperatureMeasurement::Attributes::MeasuredValue::Id, &val);
    }); }
```

změna

update

3 „souřadnice“ k
atributu v centrální
tabulce

- Nová architektura Data Model Interface (DMI)
 - Robustnější, více možností, komplexnější logika. Více kódu a větší HW nároky.
 - Clustery (skupina atributů, příkazů a událostí) už nejsou v jedné centrální tabulce atributů
 - Samostatné C++ objekty, každý si řeší stav, reporting, ...

Atributy - nová logika

```
static void temp_sensor_notification(uint16_t endpoint_id, float temp, void *user_data)
{
    chip::DeviceLayer::SystemLayer().ScheduleLambda([endpoint_id, temp]() {

        auto *iface = esp_matter::data_model::provider::get_instance()
            .registry()
            .Get(chip::app::ConcreteClusterPath(endpoint_id, TemperatureMeasurement::Id));

        if (iface == nullptr) { return; }

        auto *cluster = static_cast<chip::app::Clusters::TemperatureMeasurementCluster *>(iface);
        int16_t value = static_cast<int16_t>(temp * 100);

        CHIP_ERROR err = cluster->SetMeasuredValue(chip::app::DataModel::MakeNullable(value));

        (void)err;

    });
}
```

Konečně to funguje

- Konečně to měří a posílá data!

I (30749) bme280 matter: Temperature:
20.44 degC, Humidity: 44.95 %, Pressure:
980.02 hPa

I (60749) bme280 matter: Temperature:
20.39 degC, Humidity: 44.98 %, Pressure:
980.09 hPa

I (90749) bme280 matter: Temperature:
20.35 degC, Humidity: 45.19 %, Pressure:
980.11 hPa

I (120749) bme280 matter: Temperature:
20.40 degC, Humidity: 45.09 %, Pressure:
980.03 hPa

- Ještě jedna věc...

Zařízení - informace

TEST_PRODUCT (32768)
od TEST_VENDOR

Firmware: 1.0

Hardware: TEST_VERSION

Informace o Matter

ID uzlu: 66

Typ sítě: Thread

Typ zařízení: Koncové zařízení

Název sítě: OpenThread-e30d

MAC adresa: 52:65:08:43:9f:38:82:80

IP adresy: fd72:a80c:b818:1:dae:d3ea:30dd:b632

 Matter

Senzory

	Teplota	20,4 °C
	Tlak	980,00 hPa
	Vlhkost vzduchu	45,36 %

[Přidat do ovládacího panelu](#)

Nastavení

	Firmware	Neznámý
--	----------	---------

[Přidat do ovládacího panelu](#)

Konečně to funguje

- Konečně to měří a posílá data!

I (30749) bme280 matter: Temperature: 20.44 degC, Humidity: 44.95 %, Pressure: 980.02 hPa

I (60749) bme280 matter: Temperature: 20.39 degC, Humidity: 44.98 %, Pressure: 980.09 hPa

I (90749) bme280 matter: Temperature: 20.35 degC, Humidity: 45.19 %, Pressure: 980.11 hPa

I (120749) bme280 matter: Temperature: 20.40 degC, Humidity: 45.09 %, Pressure: 980.03 hPa

- Všechny ukázky

- Stejný párovací/QR kód
- TEST_PRODUCT/TEST_VENDOR
- Firmware a hardware verze

Zařízení - informace

TEST_PRODUCT (32768)
od TEST_VENDOR

Firmware: 1.0

Hardware: TEST_VERSION

Informace o Matter

ID uzlu: 66

Typ sítě: Thread

Typ zařízení: Koncové zařízení

Název sítě: OpenThread-e30d

MAC adresa: 52:65:08:43:9f:38:82:80

IP adresy: fd72:a80c:b818:1:dae:d3ea:30dd:b632

 Matter

Senzory

	Teplota	20,4 °C
	Tlak	980,00 hPa
	Vlhkost vzduchu	45,36 %

[Přidat do ovládacího panelu](#)

Nastavení

	Firmware	Neznámý
--	----------	---------

[Přidat do ovládacího panelu](#)

Vlastní párovací kód

1. PIN + discriminator + kontrolní číslice → 11 znaků
2. PIN + SALT + iterace → zašifrovaný otisk (verifier)

PBKDF2 + elipt. křivky



pip install ecdsa



```
$ SALT=$(openssl rand -base64 24)
$ echo "SALT: $SALT"
$ /opt/homebrew/bin/python3 ~/esp/esp-
matter/connectedhomeip/connectedhomeip/scripts/tools/spake2p/
spake2p.py gen-verifier -p PIN -s "$SALT" -i ITERATION_COUNT
```

- Proč takto? Bezpečnost.
 - Musí se použít oba způsoby
 - PIN z telefonu se neposílá “vzduchem”.
 - Verifier se zapisuje přímo do firmware.
 - Sůl a iterace jako ochrana

Vlastní párovací kód

- sdkconfig.defaults.esp32c6

```
CONFIG_CHIP_PROJECT_CONFIG="main/include/CHIPProjectConfig.h"
```

- main/include/CHIPProjectConfig.h

```
#define CHIP_DEVICE_CONFIG_USE_TEST_SETUP_PIN_CODE 11995500  
#define CHIP_DEVICE_CONFIG_USE_TEST_SETUP_DISCRIMINATOR 0xF00
```

```
#define CHIP_DEVICE_CONFIG_USE_TEST_SPAKE2P_ITERATION_COUNT  
1000  
#define CHIP_DEVICE_CONFIG_USE_TEST_SPAKE2P_SALT "xxx"  
#define CHIP_DEVICE_CONFIG_USE_TEST_SPAKE2P_VERIFIER "xxx"
```

Jméno, název, verze

- main/include/CHIPProjectConfig.h

```
#define CHIP_DEVICE_CONFIG_DEVICE_VENDOR_NAME "Vaclav Steiner"  
#define CHIP_DEVICE_CONFIG_DEVICE_PRODUCT_NAME "Matter BME280  
Sensor"  
#define CHIP_DEVICE_CONFIG_DEFAULT_DEVICE_HARDWARE_VERSION_STRING  
"1.0.1"
```

- CMakeLists.txt

```
set(PROJECT_VER "1.0.4")  
set(PROJECT_VER_NUMBER 10004)
```

```
$ rm sdkconfig  
$ rm -rf build  
$ idf.py set-target esp32c6  
$ idf.py erase-flash  
$ idf.py build flash monitor
```

Takhle to má být

Zařízení - informace

Matter BME280 Sensor (32768)
od Vaclav Steiner

Firmware: 1.0.4

Hardware: 1.0.1

Informace o Matter



Matter



Sdílet zařízení



Senzory



Teplota

24,5 °C



Tlak

98,60 kPa



Vlhkost vzduchu

37,79 %

[Přidat do ovládacího panelu](#)

Nastavení



Firmware

Aktuální

Thread síť

WebUI matter-server :5580

```
I(23583) OPENTHREAD:[N] RLOC16 4c0c -> 8400
I(23583) OPENTHREAD:[N] Role child -> router
I(23583) OPENTHREAD:[N] Partition ID 0x7ce699e
```

Thread Network Mesh

Search label, node id, or address

Find

Diagram of a Thread Network Mesh. The central node is a router (orange circle with a crown icon) labeled "ot8e99bacdf4f7fb23 OpenThread-e30d · 4 children". It is connected to several child nodes: "BILRESA scroll wheel (103597000080B097)", "MYGGBETT door/window sensor (7843294728486466894)", "MYGGBETT door/window sensor (1168005704636816725)", "GRILLPLATS Plug", "Matter BME280 Sensor", "KLIPPBOK water leak sensor", and "TIMMERFLOTTE temp/hmd sensor". There is also a node labeled "ot0abf3287126a37a8 OpenThread-e30d" connected to the router.

Node 67 @1:43

Name:

Matter BME280 Sensor

Vendor:

Vaclav Steiner

Product:

Matter BME280 Sensor

Network:

Thread

Status:

Online

THREAD NETWORK

Role:

Router

Router: forwards traffic for other nodes in the Thread mesh.

Thread version:

Thread 1.4

Channel:

18

Extended address:

0E62650B91F380B1

Direct neighbors:

2

Routable destinations:

2

CONNECTIONS (2)

Node 20 @1:14: GRILLPLATS Plug

RSSI: -28 dBm, LQI: 3, Bidir: 3, Cost: 1

ot8e99bacdf4f7fb23

RSSI: -77 dBm, LQI: 3, Bidir: 3, Cost: 1

Poslední fail

- ! sdkconfig.defaults.esp32c6 !

```
#CONFIG_ESP_CONSOLE_USB_SERIAL_JTAG=y
```

```
#CONFIG_OPENTHREAD_CONSOLE_TYPE_USB_SERIAL_JTAG=y
```

...jinak to nebude fungovat bez PC 😊

Vaše otázky?



vaclav@steinerovi.cz

<https://www.vaclavsteiner.cz>

**Děkuji za
pozornost**