

Tasmota: Okos eszközök egyszerűen



Felhasznált és ajánlott irodalom

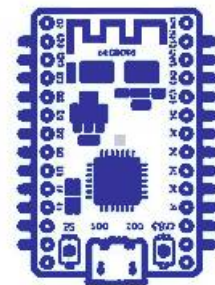
- ❖ Tasmota Dokumentáció – <https://tasmota.github.io/docs>
- ❖ Tasmota firmware kiadások – <https://github.com/arendst/Tasmota/releases>
- ❖ Tasmota Web Installer – <https://tasmota.github.io/install>
- ❖ Tasmotizer – <https://github.com/tasmota/tasmotizer>
- ❖ ESPTool – <https://github.com/espressif/esptool>
- ❖ Matter fejlesztők: [Matter Handbook](#)
- ❖ Connectivity Standards Alliance: [Matter: The Foundation for Connected Things](#)
- ❖ MQTT alapok – <https://www.hivemq.com/mqtt-essentials>
- ❖ MQTT Explorer – <https://mqtt-explorer.com>
- ❖ MQTT Dash – play.google.com/store/apps/details?id=net.routix.mqttdash
- ❖ IKEA: [DIRIGERA Hub okos termékekhez](#)

Mi a TASMOTA?

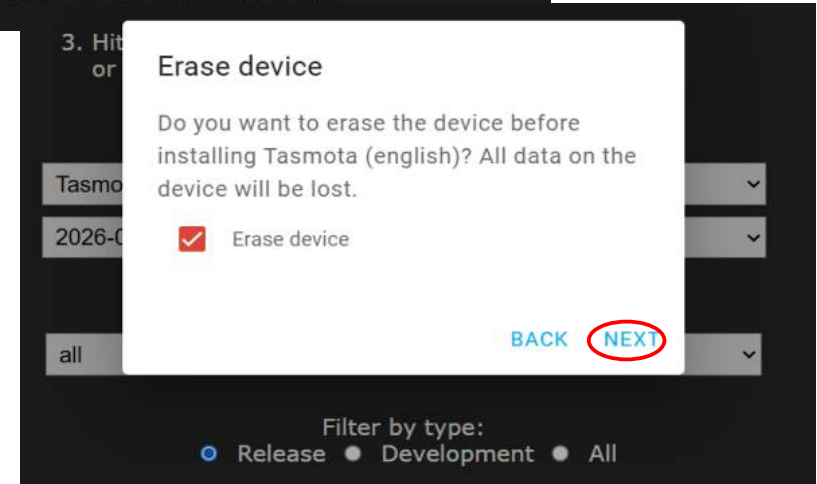
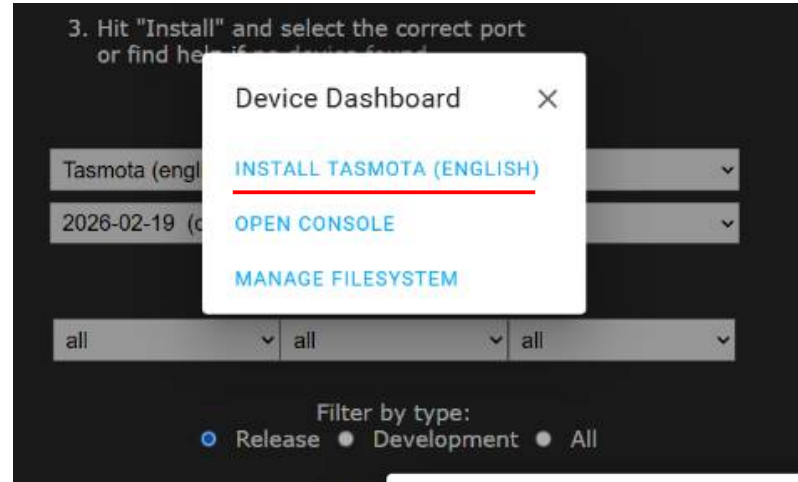
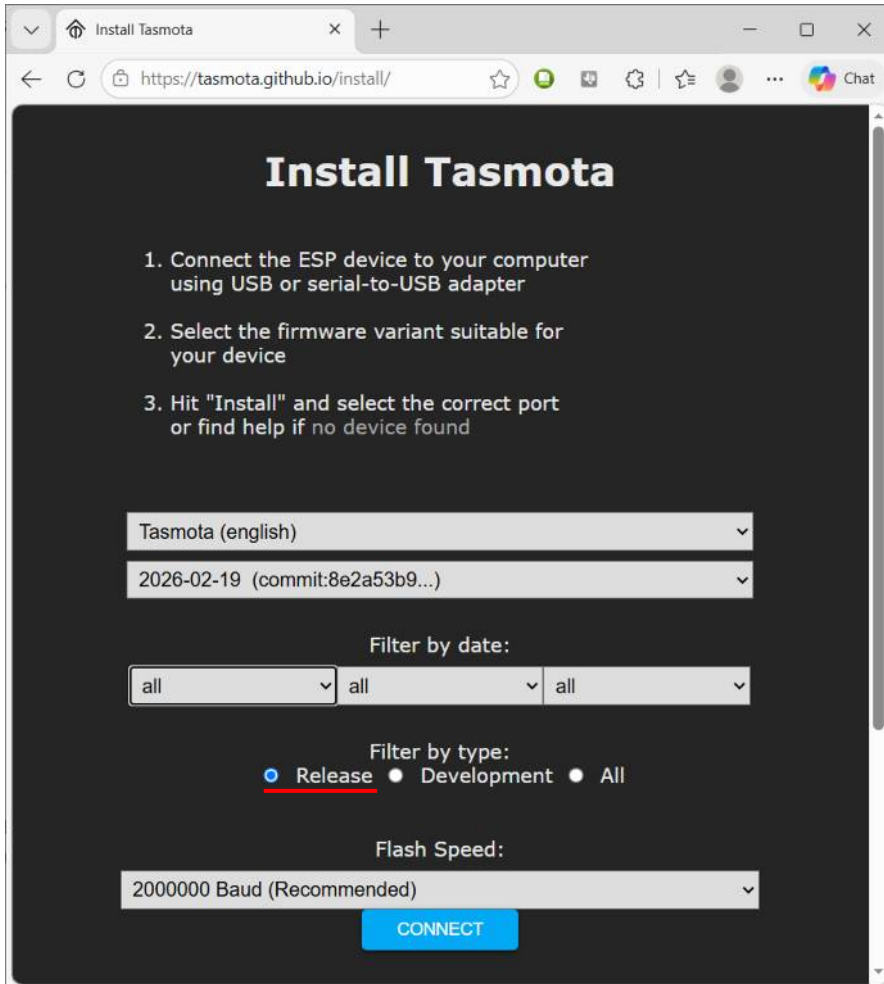
- ❖ A **Tasmota** egy nyílt forráskódú, alternatív firmware **ESP8266** és **ESP32** chipes okoseszközökhöz, amelyet **Theo Arends** indított útjára 2017 elején.
- ❖ A név eredete a fejlesztő nevének és a Sonoff, MQTT, OTA szavaknak az összevonása (Theo-Arends-Sonoff-MQTT-OTA). Mára egy globális közösség fejleszti, célja pedig egy lokális irányítású okosotthon rendszer megvalósítása, a felhőalapú szolgáltatások kikényszerített igénybevétele nélkül
- ❖ **Főbb jellemzők**
 - **Kódolás nélkül:** egy barátságos webes felületen konfigurálhatjuk eszközeinket
 - **Adatbiztonság:** Cloud-free, az adatok a helyi hálózaton maradnak
 - **Rugalmasság:** Webböngészőn, MQTT-n vagy HTTP-n keresztül vezérelhető
 - **Testreszabhatóság:** Több száz szenzort és kiegészítőt támogat
 - **OTA frissítés:** Vezeték nélküli (Over-The-Air) firmware frissítés lehetősége

A Tasmota firmware telepítése

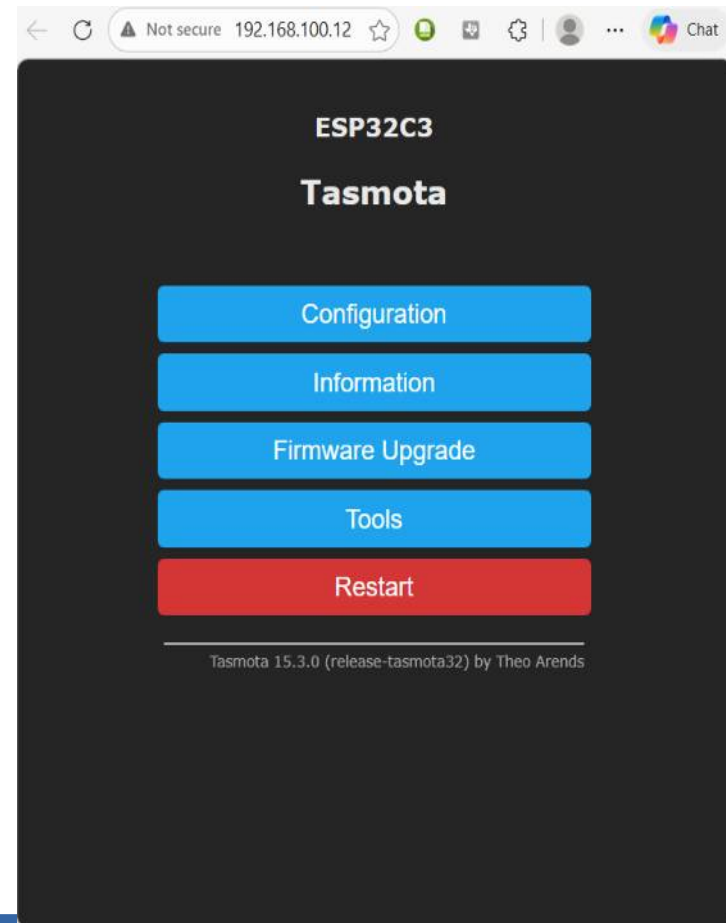
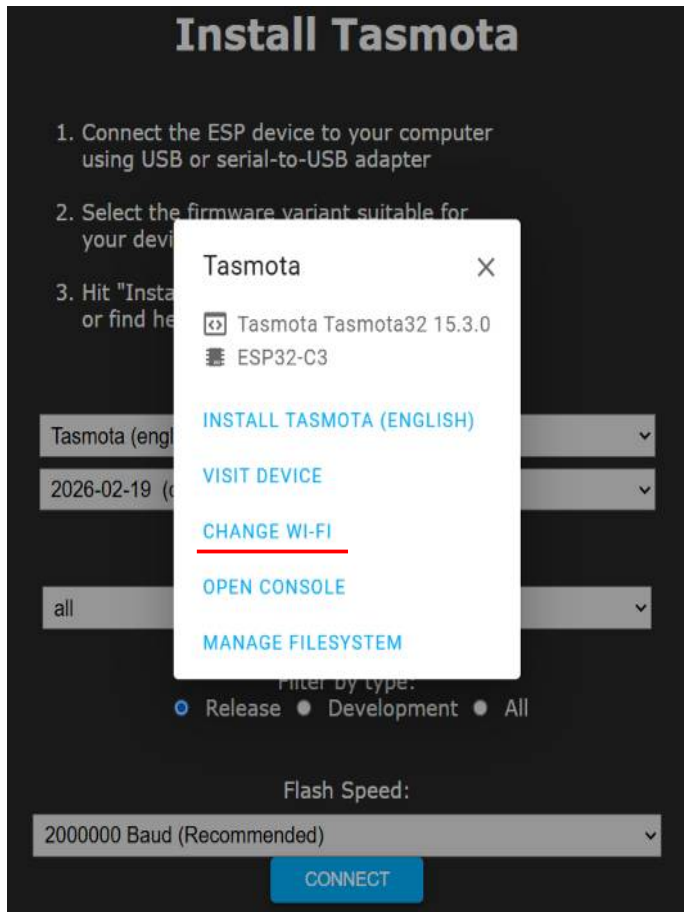
- ❖ A Tasmota firmware telepítése két úton történhet, a használt hardvertől és a kényelmi igényektől függően. A feltöltéshez fizikai (soros) kapcsolatra van szükség az eszköz és a számítógép között
- ❖ A legegyszerűbb módszer a Webes telepítés ([Tasmota WebInstaller](#)), amely főként az újabb chipkészleteknél (pl. ESP32-C3) tündököl. A böngészőben (Chrome vagy Edge szükséges hozzá) megnyitott telepítőfelület automatikusan felismeri az USB-n csatlakoztatott MCU típusát, majd a firmware feltöltése után egy felugró ablakban azonnal konfigurálhatjuk a WiFi elérést
- ❖ A hagyományos vagy „kézi” telepítés esetében egy külön szoftverrel ([Tasmotizer](#) vagy **esptool**) töltjük fel a letöltött *.bin* fájlt. A telepítés után az eszköz saját WiFi hozzáférési pontot hoz létre (Soft AP), amelyre csatlakozva a **192.168.4.1** címen érhetjük el a konfigurációs felületet. Ez a „B-terv” minden ESP alapú eszköznél üzembiztosan működik, akkor is, ha a webes út valamiért nem járható



Tasmota firmware telepítés ESP32-C3-ra



Csatlakozás a helyi WiFi hálózatra



Konfigurálás

1. GPIO4
legyen a PWM1
kimenet!

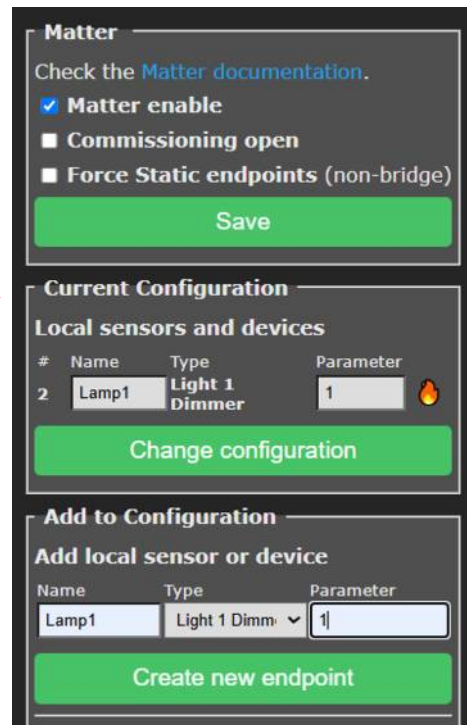
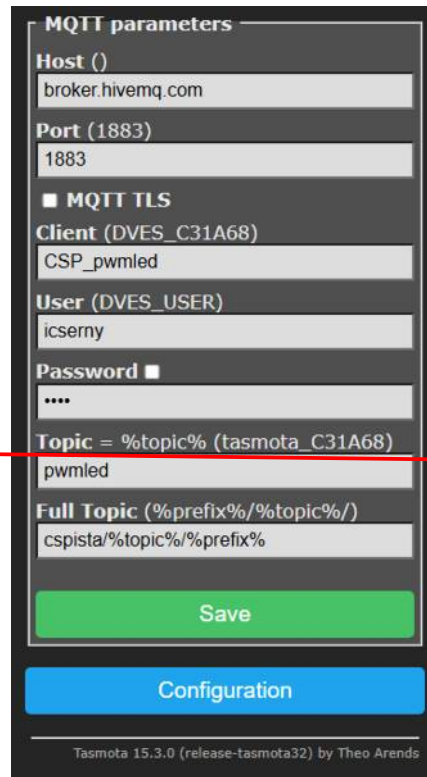
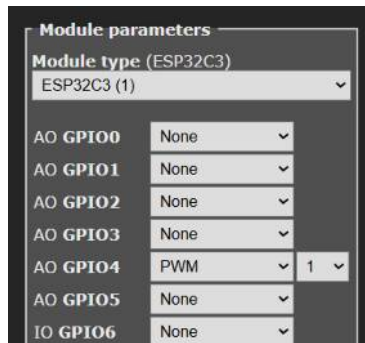
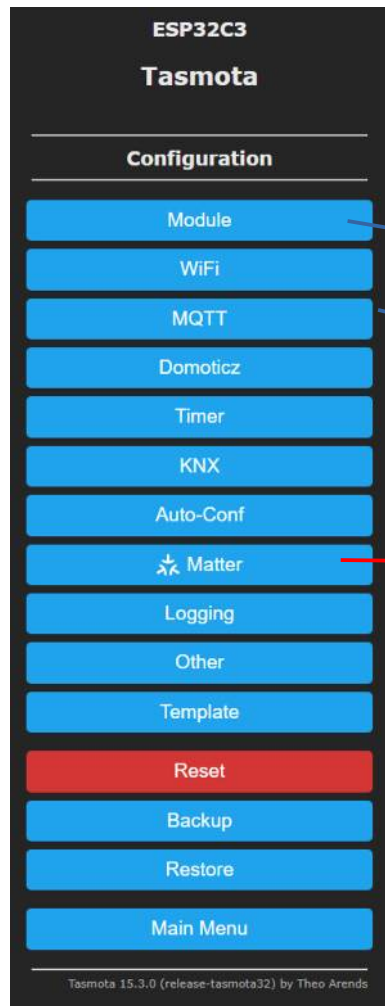
Az „előlapon” feltűnik egy csúszka

MQTT konfigurálás

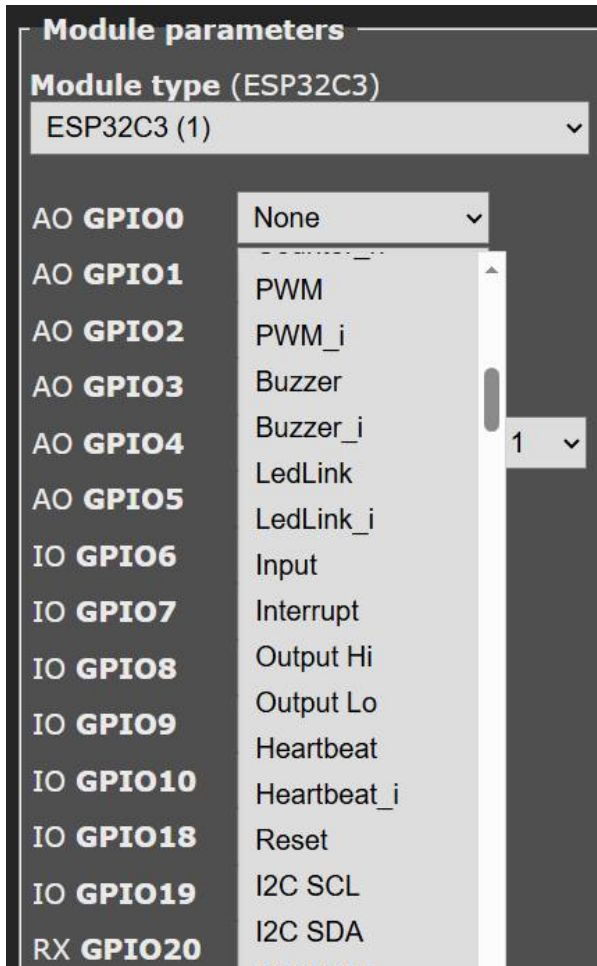
Matter konfigurálás

2. Nem tipikus eset, de most küldjük ki
az állapotot a HiveMQ szerverre!

3. Matter konfigurálás: PWM1-et itt a
Light1 fényforrás dimmelésére rendeljük



Szenzorok és aktuátorok konfigurálása

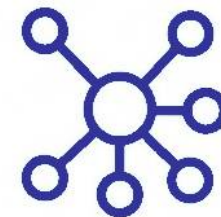


- ❖ Az alkatrészek kommunikációja eltérő: az egyszerű elektromos jelektől a digitális protokollokon át a gyári késztermékekig minden eszközt máshogy kell „megszólítani”
- ❖ **Grafikus modulbeállítás (Web UI):** egyszerű I/O eszközök-nél mi rendeljük hozzá a funkciót a lábhoz, mert a chip nem tudja, hogy mi van rákötve (GPIO, PWM, ADC)
- ❖ **Automatikus felismerés (I2C/Digitális):** az I2C eszközöknél, mint pl. a BMP180, elég a megfelelő lábakat (SDA/SCL) konfigurálni. A **Tasmota** felismeri az eszköz egyedi címét, és automatikusan elkezdi mérni a nyomást
- ❖ **Sablonok (Templates):** Késztermékek „gyári” beállítására. A sablon (egy JSON fájl) betölti a fix lábkiosztást, és az eszköz azonnal működik
- ❖ A **Tasmota** firmware által támogatott gyári eszközök sablonjai a <https://templates.blakadder.com/> címen érhetők el

Konfigurálás a Console ablakban

- ❖ A konzol a grafikus felületről nem elérhető, mélysztintű paraméterek és egyedi működési logikák beállítására szolgál. Minden itt kiadott beállítás a flash memóriába kerül, így az eszköz újraindítás után is megőrzi a beállított értékeket
 - **Hostname:** Az eszköz hálózati azonosítója (pl. **Hostname pwmled**). Ezután a **pwmled.local** néven is elérhetjük az eszközünket (mDNS)
 - **TelePeriod:** A szenzoradatok (pl. BMP180 értékek) küldési gyakorisága másodpercben (alapértelmezett: 300s). Például: **TelePeriod 60** egy percet állít be
 - **PWMFrequency:** A fényerőszabályzás frekvenciája (pl. **PWMFrequency 1000**). Ezzel szüntethető meg a LED vibrálása vagy a tápegység zúgása.
 - **Altitude:** A tengerszinti légnyomás pontos átszámításához szükséges tengerszint feletti magasság megadása méterben (pl. **Altitude 139**)
 - **Rules (Eseménykezelés):** Itt definiálhatók az automatizmusok, amelyek eseményekre (pl. hőmérsékletváltozás, gombnyomás) reagálnak. Segítségükkel az eszköz képes adatokat küldeni (pl. HTTP kéréssel a **ThingSpeak** felhőbe) vagy helyi akciókat végrehajtani

MQTT: Távvezérlés és kommunikáció



- ❖ Az **MQTT** protokoll teszi lehetővé, hogy az eszköz adatokat küldjön (*Publish*) és parancsokat fogadjon (*Subscribe*) egy központi kiszolgálón (*Broker*) keresztül.
- ❖ **Broker (Host):** A központi „postahivatal” címe (pl. **broker.hivemq.com**), amely továbbítja az üzeneteket az eszközök között ez lehet helyi vagy távoli szerver
- ❖ **Topic (Téma):** Az üzenetek címe/útvonala. A képen a **pwmled** az egyedi azonosító, amivel megszólíthatjuk az eszközt.
- ❖ **Full Topic:** Meghatározza az üzenetek hierarchiáját (pl. **cspista/%topic%/%prefix%/**). Ez segít rendszerezni több száz eszközt egy hálózaton belül.
- ❖ **Client ID:** Az eszköz egyedi neve a bróker felé (pl. **CSP_pwmled**), ami segít az azonosításban és a kapcsolat fenntartásában

A screenshot of the Tasmota MQTT configuration interface. The interface is dark-themed with white text. At the top, it says "MQTT parameters". Below this, there are several input fields: "Host ()" with the value "broker.hivemq.com", "Port (1883)" with the value "1883", a checkbox for "MQTT TLS" which is unchecked, "Client (DVES_C31A68)" with the value "CSP_pwmled", "User (DVES_USER)" with the value "icserny", and a "Password" field with masked characters. Below these is a "Topic" field with the value "%topic% (tasmota_C31A68) pwmled" and a "Full Topic" field with the value "%prefix%/%topic%/ cspista/%topic%/%prefix%". At the bottom of the form is a green "Save" button. Below the form is a blue "Configuration" button. At the very bottom, in small text, it says "Tasmota 15.3.0 (release-tasmota32) by Theo Arends".

Matter: A gyártófüggetlen jövő



- ❖ A **Tasmota** a 13.0-s verziótól támogatja a **Matter** protokollal való közvetlen kommunikációt. Ez lehetővé teszi, hogy az eszközünk bridge nélkül kapcsolódjon az Apple Home, Google Home vagy IKEA Home rendszerekhez
- ❖ **Egyszerű konfiguráció:** Kiválasztjuk az eszköz kategóriáját (pl. Light 1 Dimmer) és hozzárendeljük a megfelelő belső adatforrást (végpontot)
- ❖ **Végpontok (Endpoints):** A Matter rendszerben minden funkció egy végpont. A **Tasmota** ezeken keresztül „hirdeti meg” magát a hálózaton: „*Én egy lámpa vagyok, és tudok fényerőt szabályozni.*”
- ❖ **Adat-tolerancia:** A szabvány rendkívül rugalmas. Ha az eszközünk több adatot szolgáltat (pl. a hőmérsélet mellett légnyomást is mérünk), a fogadó rendszer (pl. IKEA Home) egyszerűen figyelmen kívül hagyja, amit nem tud értelmezni

The screenshot shows the Tasmota Matter configuration interface. It has three main sections: 'Matter', 'Current Configuration', and 'Add to Configuration'. The 'Matter' section has a link to 'Matter documentation', a checked 'Matter enable' checkbox, and two unchecked checkboxes: 'Commissioning open' and 'Force Static endpoints (non-bridge)'. Below these is a green 'Save' button. The 'Current Configuration' section has a heading 'Local sensors and devices' and a table with columns '#', 'Name', 'Type', and 'Parameter'. The table contains one entry: '# 2', 'Name Lamp1', 'Type Light 1 Dimmer', and 'Parameter 1'. To the right of the parameter is a small flame icon. Below the table is a green 'Change configuration' button. The 'Add to Configuration' section has a heading 'Add local sensor or device' and a form with three fields: 'Name' (containing 'Lamp1'), 'Type' (a dropdown menu showing 'Light 1 Dimmer'), and 'Parameter' (containing '1'). Below these fields is a green 'Create new endpoint' button.

#	Name	Type	Parameter
2	Lamp1	Light 1 Dimmer	1

Közvetlen vezérlés: WebSend (HTTP)

- ❖ A **WebSend** parancs lehetővé teszi, hogy egy **Tasmota** eszköz közvetlenül, a helyi hálózaton keresztül adjon utasítást egy másiknak
- ❖ **Függetlenség:** Nem igényel **MQTT** brókert vagy központi szerveret; „pont-pont” kapcsolat van az eszközök között
- ❖ **Célzott elérés:** Az eszközt a neve alapján szólítjuk meg (pl. **nappali.local**), így az IP-cím változása sem okoz gondot (**mDNS** támogatás)
- ❖ **Szintaxis:** `WebSend [pwmled.local] Power1 TOGGLE` – ezzel parancssorból vagy egy szabályból (Rule) átkapcsolhatjuk a másik szobában lévő lámpát
- ❖ **Permanens:** Ha egy szabályba (Rule) foglaljuk, ez a közvetlen kapcsolat az eszközök között áramszünet után is azonnal működőképes marad.



Mintapéldák

PWM LED - Matter lámpa dimmer / MQTT

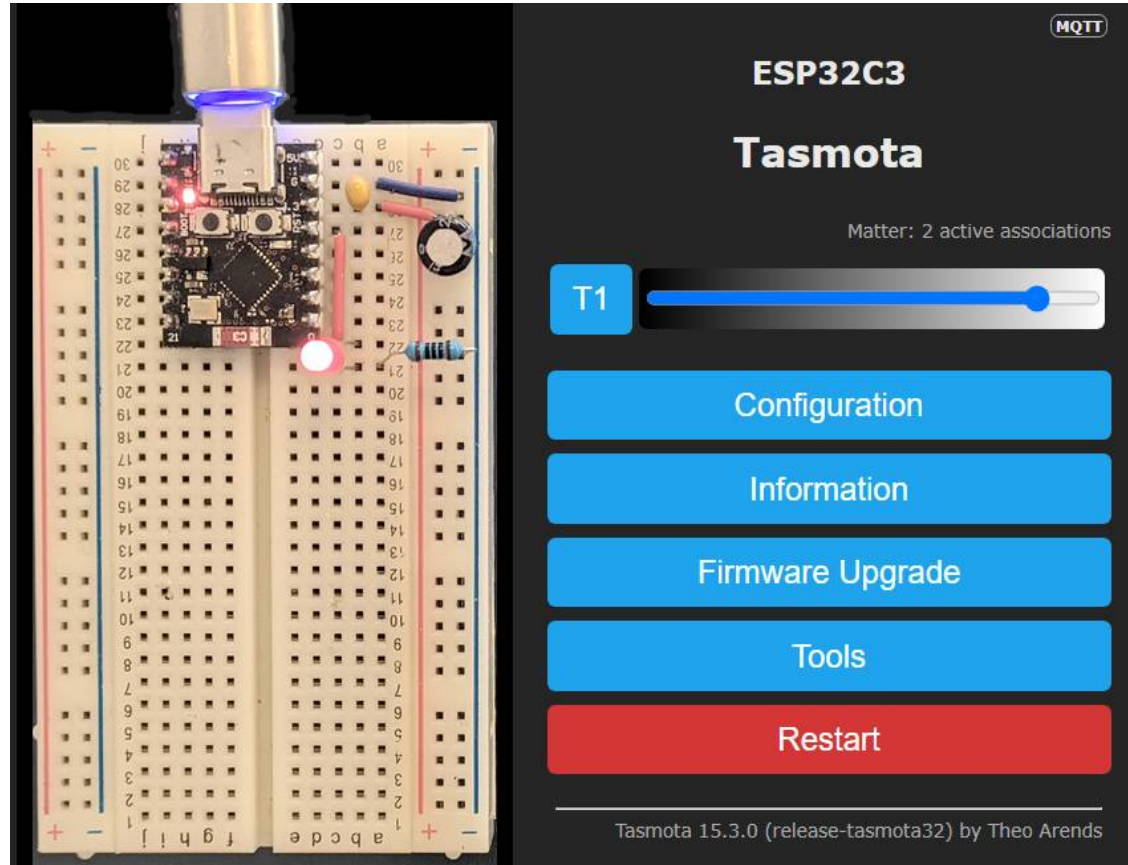
BMP180 - Matter hőmérő / MQTT

BME280 - Matter hőmérő és higrométer /
ThingSpeak HTTP kliens



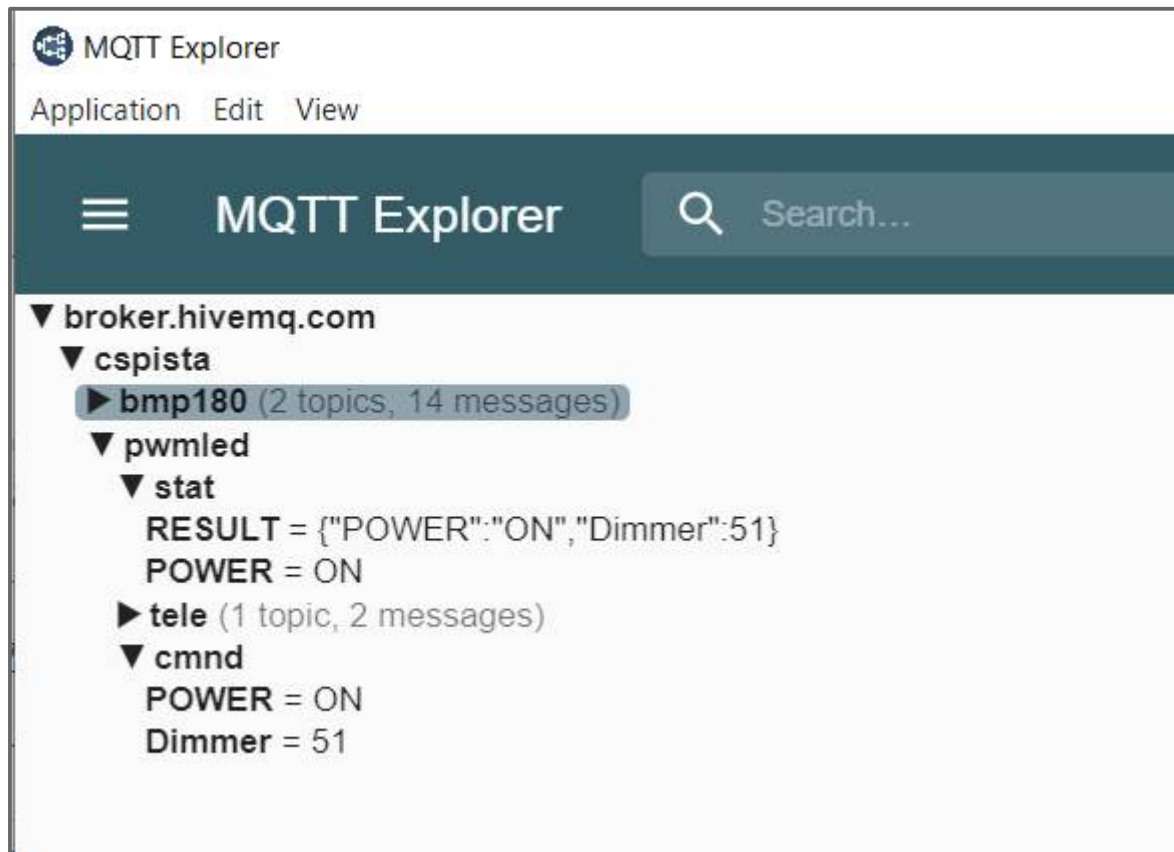
PWMLED – Matter/MQTT eszköz

- ❖ Hardver: **ESP32-C3**, GPIO4 PWM kimenet, ami egy LED-et vezérel
- ❖ Szoftver: **tasmota32c3.bin** v15.3.0, melynek konfigurálása az előző oldalakon látható (Matter eszköz és MQTT Publisher a HiveMQ nyilvános brókerhez)
- ❖ **Matter** eszközként **Lamp1** nevet kapott
- ❖ **MQTT** eszközként a **cspista/pwmled/stat** és a **cspista/pwmled/cmnd** topik a legfontosabb (POWER és Dimmer)



Ellenőrzés MQTT Explorerrel

- ❖ A szenzor(ok) által küldött MQTT üzeneteket legkönnyebben az MQTT Explorerrel ellenőrizhetjük
- ❖ Az állapotjelzések a **cspista/pwmled/stat/RESULT** topikba érkeznek. Sajnos, a **Dimmer** értéket csak akkor kapjuk meg, ha változás van
- ❖ Ha a **pwmled**-et vezérelni akarjuk, akkor az új beállításokat a **cspista/pwmled/cmnd** megfelelő altopikjába kell küldeni:
POWER = ON/OFF (vagy 1/0)
Dimmer = 0..100



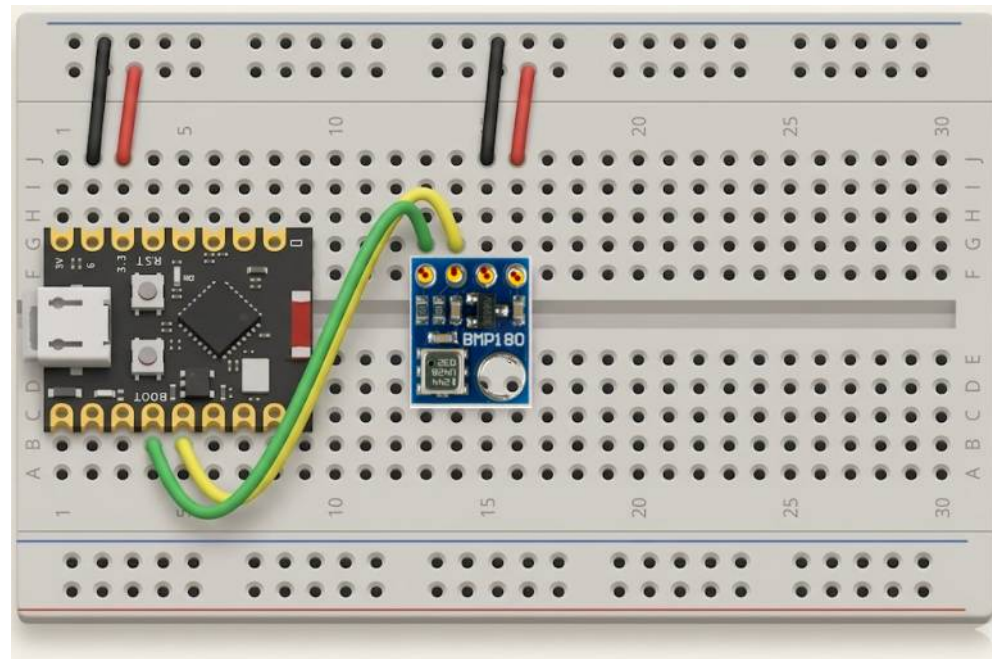
Csatlakozás a Matter hálózathoz

- ❖ A már konfigurált **Matter** eszközt a **Commissioning open** opció bejelölésére és a **Save** gomb kattintása után vehetjük fel a Matter hálózatba (esetemben az **IKEA Home** rendszerébe) a megjelenő QR kód, vagy a **Manual pairing code** segítségével



BMP180 – Matter hőmérő /MQTT szenzor

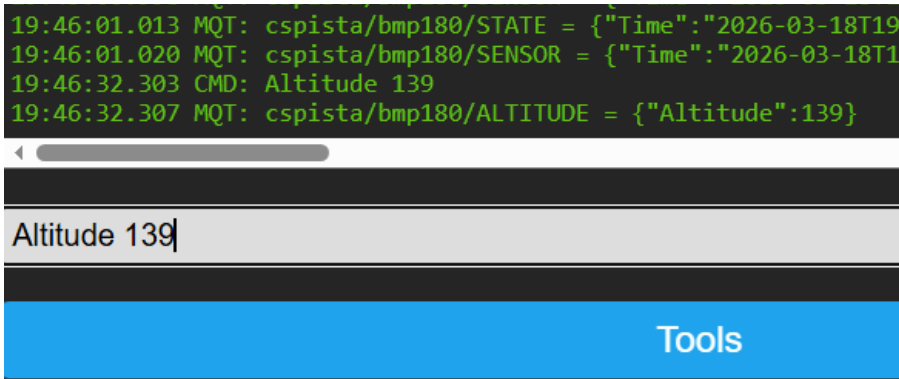
- ❖ **Modul konfigurálás:** elegendő a **GPIO8 – I2C_SDA** és **GPIO9 – I2C_SCL** hozzárendelést megadni, a szenzort a TASMOTA automatikusan felismeri
- ❖ **Matter konfigurálás:** a beépített hőmérő és a nyomásmérő külön-külön végpontot kíván ezért az **Add local sensor or device** szekcióban először felvesszünk egy **Temperature** típusú végpontot, melynek paramétere **BMP180#Temperature** majd a **Create new endpoint** gombra kattintunk, majd létrehozunk egy **Pressure** típusú végpontot is, melynek paramétere **BMP180#Pressure** legyen
- ❖ Az **IKEA Home** nem használja a Pressure kategóriájú adatokat



SeaPressure – egy hasznos tipp

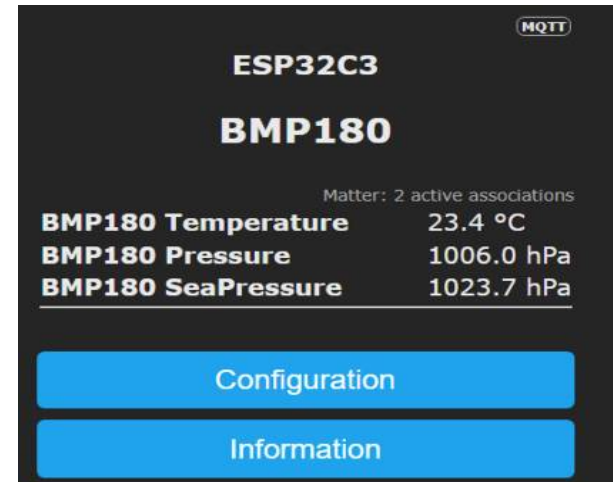
- ❖ A nyomásmérők által mért légnyomást a **Tasmota firmware** automatikusan átszámítja tengerszintre, ha megadjuk a mérés helyének tengerszint feletti magasságát. Erre nincs menüpont ezt csak konzol paranccsal tudjuk megadni
- ❖ Nyissuk meg a **Main menu** → **Tools** → **Console** menüpontot
- ❖ Írjuk be, hogy **Altitude 139** (Debrecen, 5. emelet) majd nyomjunk **Enter**-t
- ❖ A Main menüben megjelenik a **SeaPressure** paraméter, ami a **Matter** vagy **MQTT** adatküldéseknél is felhasználható

```
19:46:01.013 MQTT: cspista/bmp180/STATE = {"Time":"2026-03-18T19:46:01.020 MQTT: cspista/bmp180/SENSOR = {"Time":"2026-03-18T19:46:32.303 CMD: Altitude 13919:46:32.307 MQTT: cspista/bmp180/ALTITUDE = {"Altitude":139}
```



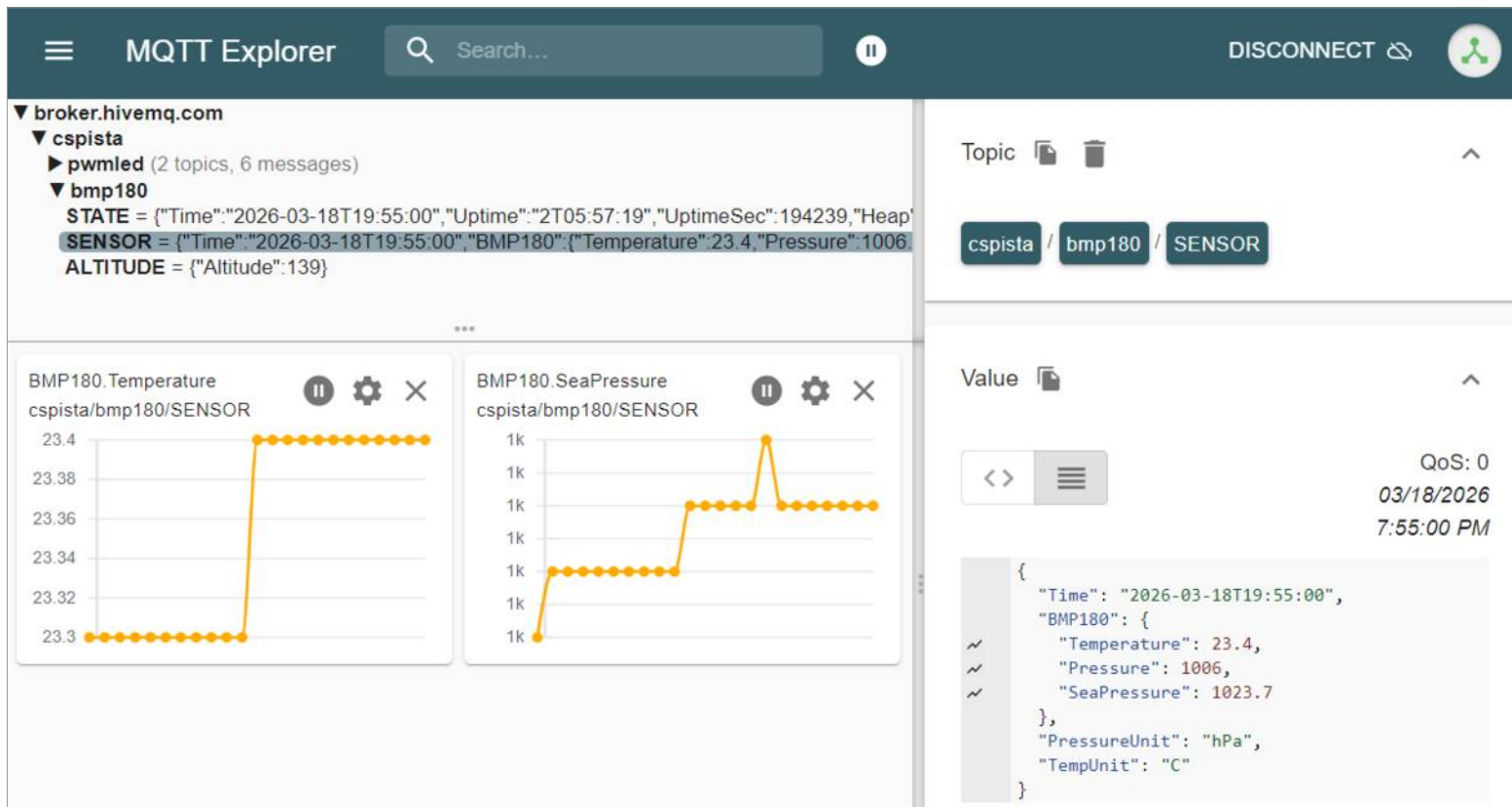
Altitude 139

Tools



BMP180 – MQTT konfigurálás

- ❖ Helyi szerver hiányában a **HiveMQ** publikus szerverét, a **broker.hivemq.com** szervert használjuk



The screenshot shows the configuration form for the ESP32C3 BMP180. The form is titled **ESP32C3 BMP180**. It contains the following fields:

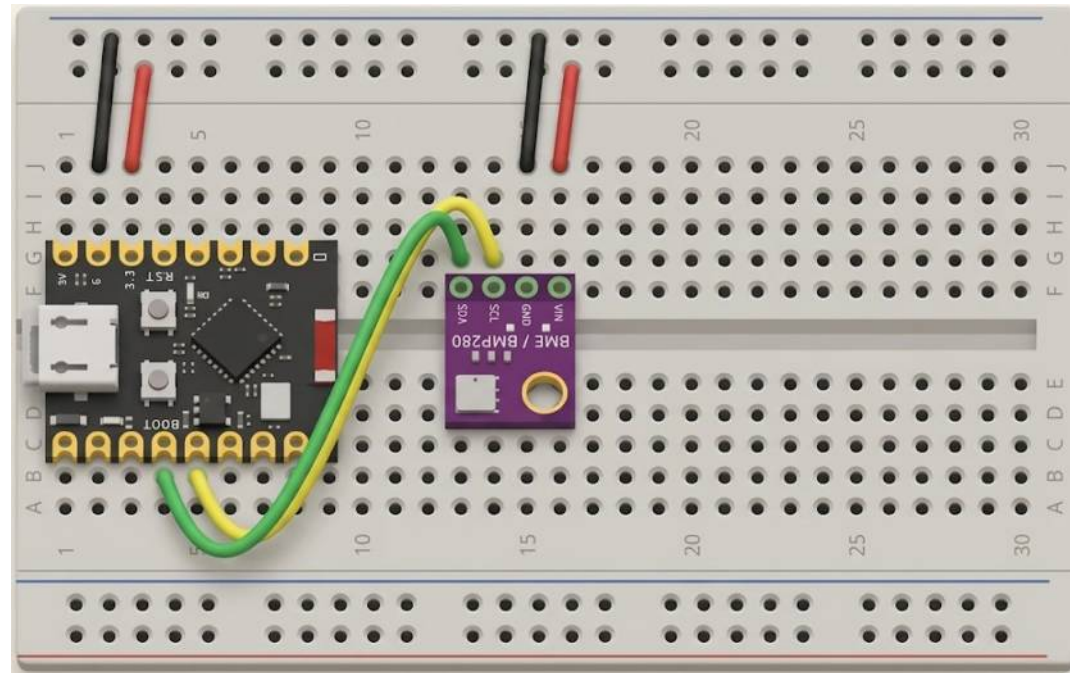
- MQTT parameters**
 - Host** (): broker.hivemq.com
 - Port** (1883): 1883
 - MQTT TLS**
 - Client** (DVES_2A9188): ESP32_bmp180
 - User** (DVES_USER): icserny
 - Password** ():
 - Topic** = %topic% (tasmota_2A9188): bmp180
 - Full Topic** (%prefix%/%topic%/): cspista/%topic%/
- Save** (green button)
- Configuration** (blue button)

BME280 – Matter szenzor /Thingspeak kliens

- ❖ **Modul konfigurálás:** elegendő a **GPIO8 – I2C_SDA** és **GPIO9 – I2C_SCL** hozzárendelést megadni, a szenzort a TASMOTA automatikusan felismeri
- ❖ A **Console** ablakban adjuk meg a magasságot (esetemben **Altitude 139**)
- ❖ **Matter konfigurálás:** a beépített hőmérő és a nyomásmérő külön-külön végpontot kíván ezért az **Add local sensor or device** szekcióban az alábbi végpontokat vesszük fel:

Temperature	BME280#Temperature
Humidity	BME280#Humidity
Pressure	BME280#SeaPressure

- ❖ Az **IKEA Home** nem használja a **Pressure** kategóriájú adatokat



Adatküldés a ThingSpeak szerverre

- ❖ A menüből konfigurálható **MQTT** adatküldés bőbeszédősége miatt nem alkalmas a kötött formátumú adatküldésre
- ❖ **A megoldás:** a **Console** ablakban eseménykezelő szabály (**Rule**) és a **WebSend** parancs segítségével állítjuk össze az üzenetet (a küldéshez a ThinSpeak Write API Key kell)



Rule1

```
ON Tele-BME280#Temperature DO var1 %value% ENDON
ON Tele-BME280#Humidity DO var2 %value% ENDON
ON Tele-BME280#SeaPressure DO backlog var3 %value%;
WebSend [api.thingspeak.com]/update?api_key=<WRITE_KEY>&field1=%var1%&field2=%var3%&field3=%var2% ENDON
```

- ❖ Az üzenetküldések gyakorisága a **Teleperiod n** paranccsal állítható be, ahol az **n** a másodpercekben megadott periódusidő, értéke 10 – 3600 közötti szám lehet



MQTT Dash

Az MQTT Dash Android alkalmazással telefonon ellenőrizhetjük/irányíthatjuk eszközeinket

Szerver konfigurálás

A kezelőfelület

BMP180 hőmérő

BMP180 nyomásmérő

PWMLED Dimmer

MQTT Dash

Name
HiveMQ

Address
35.158.68.5 (broker.hivemq.com)

Port
1883

User name
icserny

User password
.....

Client ID (must be unique)
mqttdash-csp-rn7

Tile size
☐ Small
☒ Medium
☐ Large

Metrics columns count for vertical orientation (0 - auto)
2

Metrics columns count for horizontal orientation (0 - auto)
4

HiveMQ

Temperature
24.2°C

Pressure
1023.7hPa

pwmLED
0%

PWM
0%

Subscribe:
cspista/pwmled/stat/RESULT
\$.POWER

Publish:
cspista/pwmled/cmnd/POWER

MQTT Dash

This metric is intended for displaying payload text (e.g. temperature displaying). Payload is expected to be string.

Name
Temperature

Topic (sub)
cspista/bmp180/SENSOR

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>

\$.BMP180.Temperature

☐ Enable publishing

Prefix
Postfix
°C

Main text size
☐ Small
☐ Medium
☒ Large

Main text color
[Red]

Other settings
☒ QoS(0)
☐ QoS(1)

MQTT Dash

This metric is intended for displaying payload text (e.g. temperature displaying). Payload is expected to be string.

Name
Pressure

Topic (sub)
cspista/bmp180/SENSOR

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>

\$.BMP180.SeaPressure

☐ Enable publishing

Prefix
Postfix
hPa

Main text size
☐ Small
☒ Medium
☐ Large

Main text color
[Blue]

Other settings
☒ QoS(0)
☐ QoS(1)

MQTT Dash

Name
PWM

Topic (sub)
cspista/pwmled/stat/RESULT

Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:
<https://github.com/jayway/JsonPath/blob/master/README.md>

\$.Dimmer

☒ Enable publishing

Topic (pub) - keep empty if the same as sub
cspista/pwmled/cmnd/Dimmer

☒ Update metric on publish immediately (do not wait for incoming message to update visual state)

Min
0.0
Max
100.0

Prefix
Postfix
%

Precision
0

☒ Display payload value instead of percentage

Progress color
[Blue]

Other settings
☒ QoS(0)
☐ QoS(1)