

# SHRDZM Controller

## (SHRDZM-GD28)

04.03.26

### Inhaltsverzeichnis

1. Systemübersicht.....	2
2. Nutzungsszenarien.....	2
2.1 Betrieb als SHRDZM Smartmeter Display.....	2
2.2 Individuelle Nutzung (REST-Modus).....	2
2.3 Gateway-Funktionalität (MQTT & TLS).....	2
3. Ersteinrichtung & Updates.....	3
3.1 Erstinbetriebnahme.....	3
3.2 Over-the-Air (OTA) Updates.....	3
4. REST-API Referenz.....	4
4.1 Anzeige-Steuerung (/api/view).....	4
4.2 Hardware-Befehle (HTTP GET).....	4
5. MQTT Listener.....	5
5.1 Anzeige-Steuerung.....	5
5.2 OTA Update.....	5
6. Home Assistant Integration.....	5
7. Technische Spezifikationen.....	6

---

# 1. Systemübersicht

Der SHRDZM Controller (SHRDZM-GD28) ist eine Anzeigeeinheit auf Basis des **ESP32**-Mikrocontrollers. Es dient der Visualisierung von Energiedaten in Echtzeit und bietet sowohl eine native Integration für SHRDZM Smartmeter-Module als auch eine offene REST-API für Drittsysteme. Durch seine modulare Softwarearchitektur bildet es die **Basis für zukünftige Funktionserweiterungen**.

---

## 2. Nutzungsszenarien

### 2.1 Betrieb als SHRDZM Smartmeter Display

Das Display koppelt sich direkt mit einem SHRDZM Smartmeter-Modul im lokalen Netzwerk.

- Voraussetzung:** Das Smartmeter-Modul benötigt **Firmware**  $\geq$  **v1.3.1**. Ältere Versionen müssen vorab über das Web-Interface des Moduls per **OTA-Update** aktualisiert werden. Am Smartmeter-Modul sind keine spezifischen Einstellungen durchzuführen.
- Datentransfer:** Die Kommunikation erfolgt verschlüsselt via **UDP-Multicast**
- Netzwerkanforderung:** Das WLAN muss UDP-Multicast (IGMP) unterstützen. In Mesh-Systemen ist sicherzustellen, dass Multicast-Pakete zwischen den Nodes nicht blockiert werden. (UDP Multicast : IP 239.0.1.12, Port 21220 , 21221 , 21222)

### 2.2 Individuelle Nutzung (REST-Modus)

Das Display kann als universelles Dashboard für Smart-Home-Systeme (z. B. Home Assistant, ioBroker) genutzt werden. Über die API lassen sich Texte, Werte und Farben unabhängig von SHRDZM-Smartmetermodul steuern.

### 2.3 Gateway-Funktionalität (MQTT & TLS)

Das Gerät fungiert als Bridge: Empfangene lokale Smartmeter-Daten können verschlüsselt per **MQTT(S)** an externe Broker (z. B. Energiegemeinschaften) weitergeleitet werden.

---

## 3. Ersteinrichtung & Updates

### 3.1 Erstinbetriebnahme

1.**Access-Point Modus:** Zum Konfigurieren den AP Modus aktivieren. Es wird ein eigener Hotspot (SHRDZM-GD28-XXXX) mit dem Passwort shr dzm123 gestartet. Mit diesem verbinden.

2.**Konfiguration WLAN:** Über die IP 192.168.4.1 erreicht man die Konfiguration. Als erstes WLAN Einstellungen eingeben und Modul neu starten.

3.**Zugriff:** Browser mit der am Display angezeigten IP Adresse verbinden und restliche Einstellungen vornehmen.

4.**Kopplung mit Smartmetermodul:** Im Menü "SHRDZM Modul" wird ein Netzwerk-Scan gestartet. Gefundene Module können per Klick gekoppelt werden.

### 3.2 Over-the-Air (OTA) Updates

Das System verfügt über eine integrierte **OTA-Update-Funktion** .

- Firmware-Updates können direkt über das Web-Interface geladen werden. Nach der ersten Inbetriebnahme wird empfohlen auf die aktuellste Version upzudaten.

- Dies ermöglicht die einfache Installation neuer Features und Sicherheits-Fixes, ohne das Gerät physisch an einen Computer anschließen zu müssen.

---

## 4. REST-API Referenz

### 4.1 Anzeige-Steuerung (/api/view)

Dieser Endpunkt erwartet einen **HTTP POST** mit einem JSON-Body.

**JSON-Struktur Beispiel:**

```
{
  "view_id": 1,
  "params": {
    "value": 450.5,
    "header": "PV-Anlage",
    "unit": "Watt",
    "color": "#4BED00",
  },
  "max_val": 2000,
  "custom_icon":
  {
    "icon_name": "solar",
    "show": true
  }
}
```

**view\_id:** 1 (Text), 2 (Balken), 3 (Gauge/Zeiger), 10 (Split).

**value:** Anzuzeigender Wert (nur Zahlen)

**header:** Anzuzeigender Header. (optional)

**unit:** Anzuzeigende Einheiten. (optional)

**color:** HTML-Farbcodes oder Name der Farbe. (optional)

**max\_val:** Maximalwert für Balken und Analoganzeige (optional)

**custom\_icon/icon\_name:** Name des anzuzeigenden Symbols ("solar" | "heater") (optional)

**custom\_icon/show:** Anzeigen oder löschen (optional)

### 4.2 Hardware-Befehle (HTTP GET)

Endpunkt	Parameter	Beschreibung
----------	-----------	--------------

/api/bright	v=0..255	Setzt die Backlight-Helligkeit.
-------------	----------	---------------------------------

/api/invert	v=0/1	Invertiert die Displayfarben.
-------------	-------	-------------------------------

/api/flip	v=0/1	Dreht die Anzeige um 180° (inkl. Redraw).
-----------	-------	---

/api/status	-	Liefert aktuellen Wert, Header und RSSI als JSON.
-------------	---	---

/api/reboot	-	Löst einen System-Neustart aus.
-------------	---	---------------------------------

## 5. MQTT Listener

### 5.1 Anzeige-Steuerung

Topic : <gesetztes MQTT Topic>/api/view)

Gleiche Struktur wie bei 4.1 Anzeige-Steuerung (/api/view)

### 5.2 OTA Update

Topic : <gesetztes MQTT Topic>/startota)

**JSON-Struktur Beispiel:**

```
{
  "ota": true,
  "url": "https://upgrade.shrdzm.com"
}
```

## 6. Home Assistant Integration

Beispiel Einbindung über `rest_command` in der `configuration.yaml`:

yaml

```
rest_command:
  update_shrdzm_display:
    url: "http://[IP-DES-DISPLAYS]/api/view"
    method: post
    content_type: "application/json"
    payload: '{"view_id":1,"params":
{"value":"2322","header":"Verbrauch","unit":"Watt","color":"#4
BED00"}}'
```

## 7. Technische Spezifikationen

- MCU:** ESP32 Dual-Core (240 MHz).
- Display:** 2.8" TFT, 320x240 Pixel.
- Treiber:** ST7789 (Grafik) & XPT2046 (Touch).
- Backlight:** Hardware-PWM Steuerung.
- Persistenz:** Zentrales Speichermanagement im NVS-Flash.
- Verschlüsselung:** Crypto Algorithmus für UDP-Daten Empfang vom Smartmetermodul.