

ESPHome

[Link zu https://esphome.io/](https://esphome.io/)

- [ESPHome-Device über eine lokale Python Umgebung in Windows installieren](#)
- [ESPHome WasserMeter Sensor \(Beispieldateien\)](#)

ESPHome-Device über eine lokale Python Umgebung in Windows installieren

Vorbereitung

Alles in der Eingabeaufforderung als Administrator ausführen!

Für die Installation wird Git benötigt

```
winget install --id Git.Git -e --source winget
```

Eine neue virtuelle Umgebung für Python erstellen

```
python -m venv "C:\Program Files\Python314\esphome"
```

Die virtuelle Umgebung aktivieren

```
"C:\Program Files\Python314\esphome\Scripts\activate.bat"
```

Die virtuelle Umgebung wird nun durch ein vorangestelltes (esphome) angezeigt.
Alle weiteren Befehle erfolgen in der virtuellen Umgebung.

PIP auf die neueste Version bringen

```
python.exe -m pip install --upgrade pip
```

ESPHome in der virtuellen Umgebung installieren

```
pip install esphome
```

Die yaml-Dateien des Projekts in den Ordner kopieren

[ESPHome WasserMeter Sensor \(Beispieldateien\)](#)

```
cp c:\Users\User\Desktop\WaterMeter.yaml c:\Program Files\Python314\esphome\Scripts
cp c:\Users\User\Desktop\secret.yaml c:\Program Files\Python314\esphome\Scripts
```

Hier sind es zwei Dateien da die Informationen für das WLAN extra in der Datei secret.yaml stehen.

Die Verbindungseinstellungen anzeigen

Bei gedrückter Boot-Taste den Espressif-Microcontroller an den USB-Port anstecken. Der Befehl **mode** zeigt dann den Gerätestatus mit den Verbindungseinstellungen.

```
mode
```

Falls keine Verbindungseinstellungen angezeigt werden, ist das USB-Kabel ein reines Ladekabel.

Die Programmierung durchführen

```
(esphome) c:\Program Files\Python314\esphome\Scripts>esphome run WaterMeter.yaml --device COM3
```



```

## |
## |          EP32 |
## - | 5v          az-delivery-devkit-v4 | -- ANT
## |
## |          16 17 5  18 19          23 |
## |  └─x─x─x─x─x─x─x─x─o─x─o─o─o─o─o─o─o─o─o─o─o─o─o─┘
## |          | | | | |          | |
## |          o | | o |          |  └─o - 2 (GND)
## |          7 o | 4 o          o
## |          GD00 6 | CLK 5          3
## |          GD02 o  MISO          MOSI
## |          8
## |          CSN
##
## -----
wmbus:
  mosi_pin: GPI023  ## SI:  gelb  3: MOSI Attached to Hardware SPI controller MOSI SPI
Interface
  miso_pin: GPI019  ## SO:  violett  5: MISO Attached to Hardware SPI controller MISO SPI
Interface
  clk_pin: GPI018  ## SCLK: grau 4: SCK  Attached to Hardware SPI controller CLK
  cs_pin: GPI005   ## CSN:  weiß  8: CSN  Attached to Hardware SPI controller
  gdo0_pin: GPI016 ## GD00: orange  7: RX Clock output. High Impedance !
  gdo2_pin: GPI017 ## GD02: braun  6: TX FIFO status signals. High Impedance !

#led_pin: GPI002
frequency: 868.950
all_drivers: False
sync_mode: True
log_all: true

# Example configuration entry
#mqtt:
# broker: 192.168.1.50
# username: mqttuser
# password: !secret mqtt_password
# mqtt:
#   broker: 192.168.1.50
#   username: mqttuser

```

```
# password: !secret mqtt_password
#clients (Optional):
# name (Required): The name for this client.
# ip_address (Required): IP address.
# port (Required): Port number.
# format (Optional): Telegram format to send. HEX or RTLWMBUS. Defaults to RTLWMBUS.
# transport (Optional): TCP or UDP. Defaults to TCP.
# clients:
#   - name: "wmbusmeters"
#     ip_address: "192.168.1.100"
#     port: 9022
#     transport: TCP

#sensor:
# - platform: wmbus
#   meter_id: 18547614
#   type: hydrus
#   key: "22728910e66d83f8198560e66d83a2"
#   sensors:
#     - name: "Hauptwasserzaehler RSSi"
#       field: "rssi"
#       accuracy_decimals: 0
#       unit_of_measurement: "dBm"
#       device_class: "signal_strength"
#       state_class: "measurement"
#       entity_category: "diagnostic"

#     - name: "Hauptwasserzaehler_total"
#       field: "total_m3"
#       accuracy_decimals: 3
#       unit_of_measurement: "m³"
#       device_class: "water"
#       state_class: "total_increasing"
#       icon: "mdi:water"

#     - name: "Hauptwasserzaehler_water_temperature"
#       field: "flow_temperature_c"
#       accuracy_decimals: 1
#       unit_of_measurement: "°C"
```

```
# state_class: "measurement"
# icon: "mdi:temperature-celsius"

# - name: "Hauptwasserzaehler_remaining_battery_life_y"
# field: "remaining_battery_life_y"
# unit_of_measurement: "y"
# accuracy_decimals: 2
# icon: "mdi:battery-clock-outline"

# - name: "Hauptwasserzaehler_status"
# field: "status"
# icon: "mdi:state-machine"

# - name: "Hauptwasserzaehler_timestamp"
# field: "timestamp"
# icon: "mdi:timer-sand"

# - name: "Media"
# field: "media"

# - name: "Meter Type"
# field: "meter"

# - name: "Hauptwasserzaehler_timestamp"
# field: "timestamp"
# icon: "mdi:timer-sand"
```

sensor:

```
- platform: wmbus
  meter_id: 0x18547614
  type: hydrus
  key: "22728910e66d83f8198560e66d83a2"
  sensors:
    - name: "RSSI"
      field: "rssi"
      accuracy_decimals: 1
      unit_of_measurement: "dBm"
      device_class: "signal_strength"
```

```
state_class: "measurement"  
entity_category: "diagnostic"
```

```
- name: "Zählerstand Gesamt"  
  field: "total"  
  accuracy_decimals: 3  
  unit_of_measurement: "m³"  
  device_class: "water"  
  state_class: "total_increasing"  
  icon: "mdi:water"
```

```
- name: "Zählerstand Vormonat"  
  field: "target"  
  accuracy_decimals: 3  
  unit_of_measurement: "m³"  
  device_class: "water"  
  state_class: "total_increasing"  
  icon: "mdi:water"
```

```
- name: "Letztes Auslesedatum"  
  field: "target"  
  unit_of_measurement: "datetime"  
  icon: "mdi:water"
```

```
- name: "Wassertemperatur"  
  field: "flow_temperature"  
  accuracy_decimals: 1  
  unit_of_measurement: "°C"  
  state_class: "measurement"  
  icon: "mdi:temperature-celsius"
```

```
- name: "Restlebenszeit der Batterie"  
  field: "remaining_battery_life"  
  accuracy_decimals: 5  
  unit_of_measurement: "y"  
  entity_category: "diagnostic"  
  icon: "mdi:battery-clock-outline"
```

```
text_sensor:
```

```
- platform: wmbus
```

```
meter_id: 0x18547614
type: hydrus
key: "22728910e66d83f8198560e66d83a2"
sensors:
  - name: "Status"
    field: "status"
    icon: "mdi:state-machine"
    entity_category: "diagnostic"

# Enable logging
logger:
# level: VERBOSE
  level: INFO
## Enable Home Assistant API
api:
  encryption:
    key: "EaZvkH3GgH+LA1c7621ggc8a2wsZ9p2iynfelcI="

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password
  fast_connect: true

# Enable fallback hotspot (captive portal) in case wifi connection fails
ap:
  ssid: "Watermeter Fallback Hotspot"
  password: "PmfkIusr9lhL"

captive_portal:

web_server:
  port: 80
```

secrets.yaml

```
# Your Wi-Fi SSID and password
wifi_ssid: "home-wifi"
wifi_password: "mysecretpassword"
```