Gaszähler

- über Hall Sensor
- https://www.laszloturi.eu/2023/12/how-to-integrate-mechanical-gas-meter.html
- SS49E.PDF
- https://arduinomodules.info/ky-024-linear-magnetic-hall-module/
- https://github.com/red5alex/hall-effect-gas-meter-sensor

Aufbau

- Sensor : SS49E

Kalibrieren

ALT

download

```
substitutions:
  devicename: "iot-kl-wl-sc-gaszaehler"

esphome:
  name: iot-kl-wl-sc-gaszaehler
  friendly_name: IOT_KL_WL_SC_GasZaehler

rp2040:
  board: rpipicow

preferences:
  flash_write_interval: 5min  # set to 5min to prevent wearing out the
onboard flash module too quickly

# Enable logging
logger:
  baud_rate: 115200
  level: DEBUG

status_led:
  pin: 32

globals:
  - id: total_pulses          # Gesamte Impulse (Counter)
    type: int
    restore_value: true       # Speichert im Flash (nach Neustart
```

```
erhalten)
    initial_value: '0'
  - id: impuls_factor         # Kalibrierung: m³ pro Impuls (z. B.
0.01)
    type: float
    restore_value: true
    initial_value: '0.01'     # Passe an deinen Zähler an!
  - id: initial_consumption   # Offset: Aktueller Zählerstand zum Start
    type: float
    restore_value: true
    initial_value: '1234.56'  # Dein aktueller Zählerstand (hier
kalibrieren!)

# Enable Home Assistant API
api:
  encryption:
    key: !secret api_encryption_key
  services:
      - service: set_initial_gas_reading
        variables:
          reading: float
        then:
          - lambda: |-
              id(initial_consumption) = reading;
              ESP_LOGI("calib", "Initialwert gesetzt: %.3f m³",
reading);
          - homeassistant.service:
              service: persistent_notification.create
              data:
                title: "Gaszähler kalibriert"
                message: "Neuer Startwert: {{ reading }} m³"

ota:
  - platform: esphome
    password: !secret ota_password

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password
  ap:
    ssid: "Gaszaehler Fallback Hotspot"
    password: "schmidt01"
  #manual_ip:
  #  static_ip: 192.168.30.133
  #  gateway: 192.168.30.1
  #  subnet: 255.255.255.0
binary_sensor:
  - platform: gpio
    pin:
```

```
      number: GPIO22  # Beliebiger freier GPIO (z. B. GP22 = Pin 29)
      mode: INPUT_PULLUP
      inverted: true
    name: "Gaszähler Impuls"
    id: gas_pulse
    filters:
      - delayed_on: 100ms  # Entprellung
    on_press:
      then:
        - lambda: |-
            id(total_pulses)++;
            ESP_LOGD("gas", "Impuls! Total: %d", id(total_pulses));
        - homeassistant.service:
            service: notify.mobile_app
            data:
              message: "Gaszähler: +{{ id(impuls_factor) }} m³"

sensor:
  - platform: uptime
    name: "RP2040 Uptime"
    unit_of_measurement: "s"
    icon: mdi:timer-outline
  - platform: internal_temperature
    name: "RP2040 CPU Temperatur"
    unit_of_measurement: "°C"
    device_class: temperature
    state_class: measurement
    icon: mdi:chip
# === SENSOR: Verbrauch in m³ ===
  - platform: template
    name: "Gasverbrauch m³"
    id: gas_m3
    device_class: gas
    state_class: total_increasing
    unit_of_measurement: "m³"
    accuracy_decimals: 3
    update_interval: 10s
    lambda: |-
      return id(initial_consumption) + (id(total_pulses) *
id(impuls_factor));
# === Optional: Energie in kWh (mit Brennwert) ===
  - platform: template
    name: "Gas Energie kWh"
    device_class: energy
    state_class: total_increasing
    unit_of_measurement: "kWh"
    accuracy_decimals: 2
    update_interval: 10s
    lambda: |-
      float brennwert = 11.2;     // Aus Gasrechnung (kWh/m³)
      float zustandszahl = 0.95;  // Aus Rechnung
```

```
      return (id(initial_consumption) + (id(total_pulses) *
id(impuls_factor))) * brennwert * zustandszahl;
```