

# ToDo

ToDo	Wichtigkeit	Notes
<input type="checkbox"/> Dashboard WZ bauen	Sehr hoch	
<input type="checkbox"/> ioBroker abschalten	Sehr hoch	Defür muss Licht / Garage / Bewegung automatisiert sein !
<input type="checkbox"/> HomeMatic Gerät mit Medium umbenennen	Hoch	-
<input type="checkbox"/> HomeMatic Geräte neu importieren in HA	Hoch	Geräte werden nicht automatisch umbenannt in HA Neues Laden ist einfacher watchman für Entity / Device Prüfung durchführen!
<input type="checkbox"/> Steckdosen Keller -> neuer IP Bereich	Medium	
<input type="checkbox"/> Steckdosen Keller -> Monitoring / Notifikation	Medium	
<input type="checkbox"/> Auto : Licht Haustür	Hoch	
<input type="checkbox"/> Auto : Licht Garage	Hoch	
<input type="checkbox"/> Auto : Garage auf / Zu	Hoch	
<input type="checkbox"/> Auto : Umbau Bewegung Flur Licht	Hoch	Ist derzeit noch in HM als direkte Links hinterlegt
<input type="checkbox"/> Rauchmelder Gruppen entfernen ??	Niedrig	Kann man derzeit nicht löschen > Warum ?
<input type="checkbox"/> ESP Schaltschrank umbau auf EPHome	Hoch	 Hier muss erstmal getestet werden wie genau das Schalten erfolgt (negiert oder nicht). Sonst würden alle Relais direkt an gehen ... Vielleicht auch erst alle 230V Strippen abklemmen ....

# Briefkasten-Automatisierung in Home Assistant

## Überblick

Diese Automatisierung nutzt einen Aqara MCGQ11LM Tür-/Fenstersensor (über Zigbee2MQTT) zur Überwachung eines Briefkastens. Sie benachrichtigt bei Öffnen (z.B. Posteingang) auf zwei Handys, im Dashboard und als Popup. Zusätzlich trackt sie die Öffnungszeit und Dauer, minimiert Fehlalarme durch Self-Öffnungen und warnt bei langem Offenbleiben (z.B. blockierte Post). Throttling verhindert Spam-Benachrichtigungen.

## Voraussetzungen

- Home Assistant mit Zigbee2MQTT-Integration.

- Aqara-Sensor gepaart; Entity-ID: `binary_sensor.briefkasten_contact` (State: 'on' = geöffnet, 'off' = geschlossen).
- HA Companion App auf zwei Handys für Benachrichtigungen (Services: `notify.mobile_app_dein_handy1` und `notify.mobile_app_dein_handy2`).
- Custom Integration: Browser Mod (über HACS) für Dashboard-Popups.
- Helpers erstellen (Configuration > Helpers):
  - Input Datetime: `input_datetime.briefkasten_letztes_oeffnen` (für Timestamp).
  - Input Boolean: `input_boolean.briefkasten_snooze` (für manuellen Override).
  - Input Boolean: `input_boolean.briefkasten_offen_cooldown` (für Throttling).
- Sensor für Dauer-Tracking (in `configuration.yaml`):

```
sensor:
  - platform: history_stats
    name: Briefkasten offen Dauer
    entity_id: binary_sensor.briefkasten_contact
    state: 'on'
    type: time
    start: '{{ now().replace(hour=0).replace(minute=0).replace(second=0) }}'
    end: '{{ now() }}'
```

## Automatisierungen

Alle in `automations.yaml` oder über den visuellen Editor. Ersetze Entity-IDs bei Bedarf.

### Haupt-Automatisierung: Benachrichtigung bei Öffnen

Triggert bei kurzer Öffnung (z.B. Postbote), speichert Timestamp und Dauer.

```
automation:
  - id: briefkasten_benachrichtigung
    alias: Briefkasten geöffnet - Smarte Benachrichtigungen
    trigger:
      - platform: state
        entity_id: binary_sensor.briefkasten_contact
        from: 'off'
        to: 'on'
    condition:
      - condition: state
        entity_id: input_boolean.briefkasten_snooze
        state: 'off'
    action:
      - wait_for_trigger:
          - platform: state
            entity_id: binary_sensor.briefkasten_contact
            from: 'on'
            to: 'off'
      - timeout:
          minutes: 5
```

```

    continue_on_timeout: false
    - variables:
        open_duration: "{{ (now() -
trigger.to_state.last_changed).total_seconds() }}"
    - condition: numeric_state
        entity_id: "{{ open_duration }}"
        below: 30 # Sekunden; anpassen für kurze Öffnungen
    - service: input_datetime.set_datetime
        target:
            entity_id: input_datetime.briefkasten_letztes_oeffnen
        data:
            datetime: "{{ now() }}"
    - service: notify.mobile_app_dein_handy1
        data:
            title: "Post ist da!"
            message: "Briefkasten geöffnet für {{ open_duration }} Sekunden um
{{ now().strftime('%H:%M Uhr am %d.%m.%Y') }}."
    - service: notify.mobile_app_dein_handy2
        data:
            title: "Post ist da!"
            message: "Briefkasten geöffnet für {{ open_duration }} Sekunden um
{{ now().strftime('%H:%M Uhr am %d.%m.%Y') }}."
    - service: browser_mod.notification
        data:
            title: "Posteingang"
            message: "Kurze Öffnung ({{ open_duration }} Sek) um {{
now().strftime('%H:%M Uhr am %d.%m.%Y') }}."
    mode: single

```

## Warnung bei langem Offenbleiben

Für blockierte Post (Deckel geht nicht zu).

```

automation:
  - id: briefkasten_offen_warnung
    alias: Briefkasten steht offen - Warnung
    trigger:
      - platform: state
        entity_id: binary_sensor.briefkasten_contact
        to: 'on'
        for:
          minutes: 5
    condition:
      - condition: state
        entity_id: input_boolean.briefkasten_offen_coldown
        state: 'off'
    action:
      - service: notify.mobile_app_dein_handy1
        data:
          title: "Briefkasten steht offen!"

```

```

        message: "Der Briefkasten ist seit {{ (now() -
states.binary_sensor.briefkasten_contact.last_changed).total_seconds() | int
// 60 }} Minuten offen. Vielleicht blockiert Post? (Stand: {{
now().strftime('%H:%M Uhr') }})"
    - service: notify.mobile_app_dein_handy2
      data:
        title: "Briefkasten steht offen!"
        message: "Der Briefkasten ist seit {{ (now() -
states.binary_sensor.briefkasten_contact.last_changed).total_seconds() | int
// 60 }} Minuten offen. Vielleicht blockiert Post? (Stand: {{
now().strftime('%H:%M Uhr') }})"
    - service: browser_mod.notification
      data:
        title: "Warnung: Briefkasten offen"
        message: "Offen seit {{ (now() -
states.binary_sensor.briefkasten_contact.last_changed).total_seconds() | int
// 60 }} Min. – Überprüfen?"
        duration: 60
    - service: input_boolean.turn_on
      target:
        entity_id: input_boolean.briefkasten_offen_cooldown
      mode: restart

```

## Snooze-Reset

Auto-Reset für Snooze nach 10 Min.

```

automation:
  - id: briefkasten_snooze_reset
    alias: Briefkasten Snooze zurücksetzen
    trigger:
      - platform: state
        entity_id: input_boolean.briefkasten_snooze
        to: 'on'
        for:
          minutes: 10
    action:
      - service: input_boolean.turn_off
        target:
          entity_id: input_boolean.briefkasten_snooze

```

## Cooldown-Reset für Offen-Warnung

Throttling: Reset nach 1 Stunde.

```

automation:
  - id: briefkasten_offen_cooldown_reset
    alias: Offen-Warnung Cooldown zurücksetzen

```

```
trigger:  
  - platform: state  
    entity_id: input_boolean.briefkasten_offen_coldown  
    to: 'on'  
    for:  
      hours: 1  
action:  
  - service: input_boolean.turn_off  
    target:  
      entity_id: input_boolean.briefkasten_offen_coldown
```

## Dashboard-Integration

- Entities Card für Sensor-Status.
- Markdown Card für Timestamp: Letztes Öffnen:  
states\_input\_datetime.briefkasten\_letztes\_oeffnen
- Gauge Card für Dauer: sensor.briefkasten\_offen\_dauer
- Button für Snooze: Toggle input\_boolean.briefkasten\_snooze
- Conditional Card für Warnung: Briefkasten:  
offen\_blockiert

## Tipps und Anpassungen

- Teste mit Developer Tools > States/Services.
- Passe Dauern (z.B. 30 Sek, 5 Min) an dein Verhalten an.
- Optional: Zeitbedingungen hinzufügen (z.B. nur 8-14 Uhr).
- Erweiterbar mit Kamera oder Vibration-Sensor für bessere Genauigkeit.

Diese Setup minimiert Fehlalarme und trackt alles Elegante.

From:

<https://drklipper.de/> - Dr. Klipper Wiki

Permanent link:

<https://drklipper.de/doku.php?id=haussteuerung:todo&rev=1761633397>

Last update: **2025/10/28 07:36**

