

Smart Home  
oo

Mein Smart-Home-Background  
oooooooo

Mein Weg zu Home Assistant  
oooo

Home Assistant im Detail  
oooooooooooooooooooo

Ende  
ooo

# Mit Home Assistant zum Smart Home

Daniel Klaffenbach

30. Oktober 2018

# Inhalt

1 Smart Home

2 Mein Smart-Home-Background

3 Mein Weg zu Home Assistant

4 Home Assistant im Detail

# Was ist „Smart Home“?

## Definition<sup>1</sup>

„Oberbegriff für technische Verfahren und Systeme in Wohnräumen und -häusern, in deren Mittelpunkt eine Erhöhung von Wohn- und Lebensqualität, Sicherheit und effizienter Energienutzung auf Basis vernetzter und fernsteuerbarer Geräte [...] sowie automatisierbarer Abläufe steht.“

---

<sup>1</sup>Quelle: [https://de.wikipedia.org/wiki/Smart\\_Home](https://de.wikipedia.org/wiki/Smart_Home)

# Was ist „Smart Home“?

- Begriff in letzter Zeit fast inflationär benutzt
- Häufige Meinung: App-Steuerung == Smart Home
- Interessanter: Sprachsteuerung
- Interessantestes Teilfeld → Heimautomation
  - Verknüpfung verschiedenster Sensoren und Aktoren miteinander
  - Häufig zentrale Steuerung
  - Ziel: Alltägliche Prozesse ohne manuellen Eingriff ablaufen lassen. z.B: Rolladen bei Dunkelheit herunterlassen, Licht an am Abend, Aufwachlicht bei Dunkelheit am Morgen, Heizungssteuerung, ...

# Mein Smart-Home-Background

- Mai 2014: 1. Philips LivingColors mit ZigBee Fernbedienung → Ambient Light fürs Wohnzimmer
- Juni 2014: Philips Hue Bridge und Hue-LED Streifen
  - Ambient-Light wurde per Handy steuerbar und über REST-API scriptbar
  - Integration Ambient-Light ins Heimkino über Kodi-Addon
- November 2014: dimmbare ZigBee Steckdosen für Weihnachtsbeleuchtung → an/aus per CRON-Job, Schwibbogen im Wohnzimmer beim Film schauen dimmen
- Bis Ende 2016: Sammlung vieler ZigBee Leuchtmittel (E27-LEDs), LED-Streifen, Ambient-Lights und Fernbedienungen

# Expansion - mehr als nur Licht

- 2016: Aus- und Umbau einer Bestandsimmobilie
  - Nachrüstung Rolladensteuerung (Homematic)
  - ZigBee-Lichtschalter, ZigBee-UP-Dimmer und ZigBee-Bewegungsmelder
- 2017: Erweiterung des Homematic-Systems:
  - Homematic Zentrale
  - Heizungssteuerung
  - Fensterkontakte
  - Alarmsirene
- 2018:
  - Wassermelder
  - Garagentorsteuerung
  - Stromzählermonitoring
  - Seit Mai: Home Assistant

# Im Detail: Philips Hue

- Smartes Beleuchtungssystem auf Basis von ZigBee
  - 2,4 GHz, Mesh, offenes Protokoll
- Erstes Mainstream-ZigBee-Gateway: Philips Hue Bridge

# Im Detail: Philips Hue

## ■ Key Features:

- Hue Bridge: von Anfang an sehr gut dokumentierte REST-API
- ... dadurch viele Apps und bei Entwicklern beliebt
- kompatibel mit anderen ZLL/ZigBee Licht-Komponenten (OSRAM, IKEA, Busch-Jaeger, Gira, ...)
- Sehr zuverlässig und gute Reichweite durch Mesh
- Cloud nur optional

## ■ Nachteile Hue:

- Bisher nur Leuchtmittel<sup>2</sup> und Licht-Zubehör
- Programmierung der Rule-Engine in der Hue Bridge umständlich
- Event-Trigger nur innerhalb der Rule-Engine, nicht für externe Listener
- Kein Backup bzw. Restore für Programmierung

---

<sup>2</sup>Ausnahme: schaltbare Steckdosen

# Im Detail: Homematic

- Smart Home Bussystem mit mittlerweile 4 Protokollen:
  - BidCOS-RF („Homematic classic“): 868.3 MHz, bidirektional, AES-Signatur
  - BidCOS-wired („Homematic wired“): RS485-Bussystem
  - Homematic-IP: Weiterentwicklung des klassischen Homematic, 868 MHz, Adressierung über IPv6
  - Homematic-IP wired: neu seit 2018
- Sehr großes Sortiment an Aktoren und Sensoren
  - Teilweise auch als Bausatz erhältlich → relativ kostengünstig
- Direkte Verknüpfung zwischen Komponenten möglich (Kommunikation ohne Zentrale)
- Genutzt u.a. auch von Qivicon/Magenta Smart Home → große Verbreitung

# Im Detail: Homematic-Zentrale

- Zentralensoftware: mittlerweile Open Source
- Zentralen:
  - „Cloudlos“ (Ausnahme: Homematic-IP Access Point)
  - CCU2/CCU3 (vom Hersteller unterstützte Hardware) → unterstützen alle HM-Protokolle
  - Oder z.B. Raspberry-Pi mit offiziellem Funkmodul und Open Source Software des Herstellers
- Funksystem durch bidirektionale Kommunikation ebenfalls sehr zuverlässig
  - Allerdings: Positionierung der Zentrale entscheidend
- Programmierung per Web-UI (Wenn-Dann-Sonst-Regeln)
- Programmierung komplexer Abläufe per HM-Script
- Schnittstellen: XML-RPC, JSON-RPC, XML-Web-API (Addon), diverse Addons

# Im Detail: Homematic-Zentrale

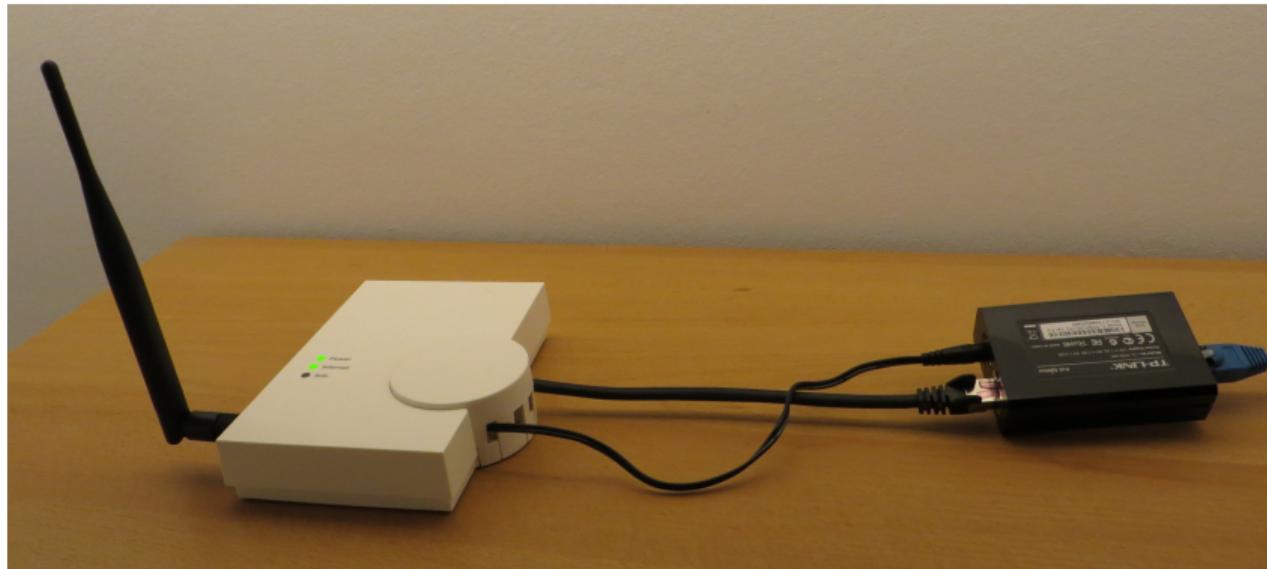


Abbildung: CCU2 mit externer Antenne und PoE-Splitter

# Motivation

- Stand Mai 2018:
  - 29 ZigBee Aktoren und 12 ZigBee Sensoren in Hue Bridge
  - 49 Homematic Geräte mit insg. 259 Kanälen und 41 Programmen
- „Probleme“:
  - Programmierung bisher nur im jeweiligen Gateway
  - Auch Homematic-Programmierung wurde komplex
  - Backupmöglichkeiten nicht optimal (in keinem der beiden Systeme)
  - Keine direkte Steuerung beider Welten über ein System<sup>3</sup>

⇒ Ich brauchte eine Smart-Home-Plattform :)

---

<sup>3</sup>Hue-Integration der Homematic-Zentrale löst nicht alle Probleme

# Wünsche und Anforderungen

- Verknüpfung der derzeit eingesetzten Systeme über eine Plattform
- Technologieneutrale Konfiguration und Verknüpfung von Komponenten
- Konfiguration über Textdateien zum einfachen Backup und grep :)
- Moderne Benutzeroberfläche und Möglichkeit zur Handysteuerung
- Evaluation verschiedener „Smart Home Plattformen“:  
OpenHAB, FHEM und Home Assistant

# Erster Start und erster Eindruck

- Installation HASS: mit Board-Mitteln der Distribution (Python/venv)
- Hue-Bridge automatisch gefunden
- IP-Adresse und Passwort der Homematic-Zentrale eingegeben
- Homematic-Geräte wurden ebenfalls alle gefunden
- Geräte waren sofort über Oberfläche bedienbar
- Direkter Start mit Gruppierung meiner Geräte in Räume
- Sofort begeistert über „Quick-Start“-Möglichkeiten, ohne Frickelei  
→ Entscheidung für HASS war nach nur wenigen Minuten gefallen

Smart Home  
oo

Mein Smart-Home-Background  
oooooooo

Mein Weg zu Home Assistant  
oooo●

Home Assistant im Detail  
oooooooooooooooooooo

Ende  
ooo

# Demo

## QuickStart-Demo

# Was ist Home Assistant?

- Plattform für Smart Home und Heimautomation
  - Ursprünglich als „Script“ für Philips Hue entwickelt (mit Anwesenheitserkennung)
  - Mittlerweile große Community (Entwickler und Nutzer)
- Funktionen:
  - Steuerung beliebiger Geräte über Web-UI (responsive Web-App)
  - Verknüpfung beliebiger Komponenten über *Automations*
  - Aufzeichnung von Änderungen (Logbuch)
  - Automatische Anwesenheitserkennung
  - Sprachsteuerung/TTS (wahlweise auch cloudlos)
  - Vielfältige Benachrichtigungsfunktionen
  - u.v.m.
- Konfiguration über YAML-Dateien oder Web-UI
- Templating über Jinja2-Templates

# Was ist Home Assistant?

- Open Source und in Python geschrieben
  - Mindestens Python 3.5 wird benötigt (asyncio)
  - Installierbar in jeder aktuellen Distribution (per venv)
  - Alternative: `hass.io` (fertiges Image für Raspberry Pi)
- Unterstützt nahezu alle Geräte, die eine API haben
  - inklusive fast aller im Smart Home relevanten Protokolle (MQTT, ZigBee, Z-Wave, ...)
  - aktuell: über 1100 Komponenten
- Optionale Cloud-Anbindung (Amazon Echo, Google Assistant, IFTTT, ...) und HomeKit-Support
- Sehr modular aufgebaut
  - Implementierungsdetails von Protokollen/Plattformen in separaten Python-Libraries (per pip installierbar bzw. automatisch installiert)

# Konzepte: Entities und State Machine

## ■ Kernkonzept: Entities

- Fast alles ist eine Entität
- Geräte können aus mehreren Entitäten bestehen  
*Beispiel:* HM-Thermostat besteht aus Regler-Entity und Feuchtigkeits-Sensor-Entity
- Auch *Gruppen, Automations* und *Scripte* sind Entities
- Entitäten haben einen Zustand („State“)  
*Beispiel:* Fenstersensor meldet entweder auf oder zu
- Entitäten haben ggf. mehrere Attribute  
*Beispiel:* Battiestatus, Anzeigename, Signalstärke, letztes Update, ...

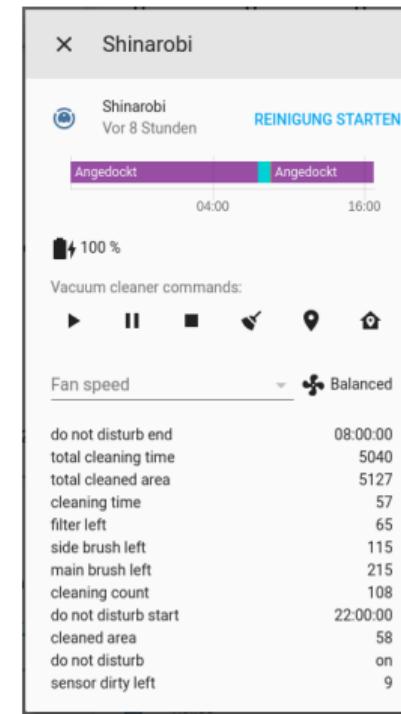
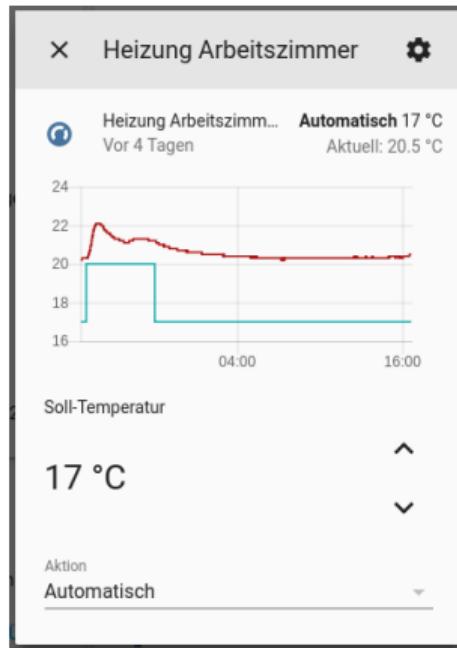
# Konzepte: Entity-Typen („Domains“)

- alarm\_control\_panel
- automation
- input\_boolean, input\_text
- remote
- binary\_sensor, sensor
- climate
- cover
- device\_tracker
- fan
- group
- light
- lock
- media\_player
- script
- switch
- vacuum
- weather
- ...

# Konzepte: Entity-Typen („Domains“)

- Jedes Gerät („Ding“) wird einem Typ zugeordnet → „Domain“
- Jedes Gerät/Ding bekommt ID: <domain>. <entity\_name>  
z.B. light.kueche\_arbeitsplatte
- Domain bestimmt mögliche Funktionen und Zustände, z.B.:
  - binary\_sensor → kann nur on/off
  - climate → Temperaturanzeige u. -einstellung, Heizprogramme
  - cover → Hoch/Runter/Stop, Anzeige der Position, ...
- Domain bestimmt Art der Anzeige in UI (Widget)
- d.h. Entity-Domain gibt „Referenz-Implementierung“ vor

# Entity-Typen in der UI



# Sensors

- Entities, die beliebige Werte für Home Assistant liefern
  - Spezieller Typ: Binärer Sensor (z.B. Fensterkontakt)
- Nicht nur „klassische“ Geräte, auch z.B.:
  - Commandline Sensor (ruft regelmäßig Script auf)
  - REST-Sensor
  - **Template-Sensor** → Stellt z.B. ein Attribut einer Entity als separaten Sensor bereit
  - iperf3-Sensor (monitort regelmäßig Netzgeschwindigkeit)
- Automatische Aufzeichnung von Zuständen (History/Logbook)
- Verwendung auch als Trigger in Automations

# Services

- Weiteres wichtiges Konzept: Dienste (**Services**)
- Verantwortlich fürs Ändern von Zuständen und Auslösen von Aktionen für Entities
- Domains stellen Dienste zur Verwendung mit ihren Entities bereit, z.B.:
  - light.turn\_on
  - switch.toggle
  - climate.set\_temperature
  - media\_player.pause
- Mindestens 1 erforderliches Argument: entity\_id
- Dazu auch Dienste von konkreten Entities (seltener), z.B.:
  - notify.mail\_an\_daniel
  - script.garagentor\_auf

# Scripte

- Einfache Script in Home Assistant → YAML
  - Streng genommen nur Macros, d.h. Abfolge von Befehlen
  - Keine Verzweigungen oder Schleifen, nur vorzeitiger Abbruch
- Python-Scripte
  - Alles, was in Python ohne separate Module funktioniert
  - d.h. `import nicht` möglich (nur Arbeiten mit Standard-Sprach-Features)
  - Zugriff auf State Machine über hass-Objekt in Scripten
- Shell-Kommando
  - Quasi keine Grenzen
  - Aber: Übergabe aller benötigter Informationen als Argument

# Automations

- Automations machen das Smart Home *wirklich* Smart
- Erlauben Reaktion auf beliebige Ereignisse → Verknüpfung unterschiedlichster Komponenten
- Bestandteile von Automations in Home Assistant:
  - 1 Trigger (mindestens einer)
  - 2 Conditions (beliebig viele, verschachtelte UND/ODER-Bedingungen möglich)
  - 3 Actions
- Jede Automation ist ebenfalls *Entity*
- Konfiguration über YAML-Dateien oder mit Automation-Editor (UI)

# Automations: Beispiele für Trigger

- Fensterkontakt meldet "offen"(State-Trigger)
- Rauchmelder meldet Rauch (State-Trigger)
- Knopf auf Fernbedienung gedrückt (Event-Trigger)
- Sonne geht in 30 Minuten unter (Sun-Trigger)
- Im Wohnzimmer sind über 25 °C (Numeric-State-Trigger)
- Es ist 12:00 (Time-Trigger)
- Es regnet (State-Trigger)
- Es wird Musik auf der Anlage abgespielt (State-Trigger)
- *Daniel* kommt heim (Presence-Trigger)

# Automations: Rollo-Automation

```
- alias: rollos_zu
  # Immer an nach Home Assistant restart
  initial_state: True
  trigger:
    - platform: sun
      event: sunset
      offset: '00:30:00'
  actions:
    - service: cover.close_cover
      # Rollogruppe "alle_rollos" ansprechen
      entity_id: cover.alle_rollos
```

# Automations: Alarm-Automation (Auszug)

**trigger:**

- **platform:** state
- entity\_id:**
  - binary\_sensor.fenster\_bad
  - binary\_sensor.fenster\_wohnzimmer
- from:** 'off'
- to:** 'on'

**condition:**

- **condition:** state
- entity\_id:** alarm\_control\_panel.alarm\_demo
- state:** armed\_away

**action:**

- **service:** alarm\_control\_panel.alarm\_trigger
- entity\_id:** alarm\_control\_panel.alarm\_demo

# Hands On

- 1** Erster Start
- 2** Konfiguration der Homematic-Komponente
- 3** Einbinden externer Sensoren (hier: command\_line-Sensor)
- 4** E-Mail-Benachrichtigung einrichten
- 5** Automation für hohe Luftfeuchtigkeit einrichten

# Ideen für Automationen

## Bitte Lüften

Sende Benachrichtigung, wenn die Luftfeuchtigkeit im Bad höher als 70 Prozent ist.

## Fenster steht lange auf

Sende eine Mail, wenn das Badfenster länger als 30 Minuten offen steht.

# Ideen für Automationen

## Anwesenheitssimulation

Schalte abends nach Sonnenuntergang zufällig das Licht an und aus, wenn niemand zu Hause ist.

## Rollladensteuerung

Rollos bei Sonnenuntergang/Sonnenaufgang schließen/öffnen.

Zusätzlich: Rollos an Wetterseite bei Regen schließen.

Rollos auf 80% fahren, wenn es tagsüber über 28 Grad sind → Hitzeschutz.

# Ideen für Automationen

## Multimedia

Dimme beim Film schauen das Licht im Wohnzimmer.

Add-On: Schalte das Telefon der FRITZ!Box stumm, wenn ein Film läuft.

## Coffee-Time

Schalte die Kaffeemaschine an, sobald der Bewegungsmelder am Morgen die erste Bewegung registriert.

# Ideen für Automationen

## Aufwachen

Dimme Werktags am Morgen das Licht im Schlafzimmer ganz langsam hoch, wenn es draußen noch dunkel ist.

Bonus: An Feiertage und Urlaub denken!

## Telefon-Terror

Wenn das Festnetz-Telefon klingelt, lasse das Licht im ganzen Haus blinken.

# Best Practices

- Konfiguration aufteilen in mehrere Dateien
- Konfiguration versionieren (git, tägliches Backup, ...)
- Konfiguration kommentieren
- Credentials in secrets.yaml
- Neue UI nutzen (Lovelace-UI)
- Monitoring der HASS-Instanz
- Regelmäßig Aktualisieren und Release-Notes verfolgen (insb. „*Breaking Changes*“)

# Best Practices

- Automatisierungen möglichst klein und atomar halten
  - Lieber mehrere Automations für das selbe Event → übersichtlicher, einzeln abschaltbar
- Automatisierungsfunktionen der jeweiligen Plattformen weiterhin sinnvoll nutzen
  - Hue: Bewegungsmelder und Schalter über Rule-Engine der Bridge programmieren, solange API keine Events unterstützt
  - Homematic: Direktverknüpfungen nutzen
  - Homematic: Zeitpläne für Heizung in der Zentrale (und damit im Thermostat) definieren
- (Off-Topic:) Beim Gerätekauf auf Unabhängigkeit von Hersteller-Clouds achten
  - Ebenfalls guter Indikator: Verfügbarkeit von Firmware-Updates

# Wo bekommt man Hilfe?

- Doku/Webseite: <https://www.home-assistant.io>
- Community: <https://community.home-assistant.io/>
- Discord: <https://discordapp.com/invite/c5DvZ4e>
- Bugs: <https://github.com/home-assistant/home-assistant/issues>

# Persönlicher Ausblick

- Teaser: Messdatenauswertung mit Grafana
- Persönliche TODOs:
  - Anwesenheitserkennung über WLAN-Controller
  - Migration von Hue auf DeCONZ (wegen Events und extrem günstigen ZigBee-Sensoren)
  - Gartenbewässerung
  - Eventuell Google Assistant

# Vielen Dank für die Aufmerksamkeit.