

Home Assistant – IPA 2025

David Kuster MM21b

Inhalt

Inhalt	2
1 Einleitung	3
2 Home Assistant.....	4
2.1 Grundlagen	4
2.2 Installation und Ersteinrichtung.....	5
2.3 Dashboard	6
2.4 Fernzugriff.....	7
2.5 HACS.....	8
3 Geräte und ihre Konnektivität zum Home Assistant.....	10
3.1 Sonoff Display NSPanel / Pro	10
3.2 Google Nest Hub	14
3.3 LaMetric Time	17
3.4 Phillips Hue.....	18
4 Aufbau	19
5 Automatisierungen und Anwendungen	20
5.1 Katzenmelder.....	21
5.2 Wettercheck	24
5.3 Batteriewechsel.....	27
6 Die Grenzen vom Home Assistant	30
7 OpenSource – Fluch oder Segen?.....	33
7.1 Vorteile von Open Source	33
7.2 Herausforderungen und Nachteile	33
7.3 Fazit	34
8 Reflexion.....	35
9 Anhang	37

1 Einleitung

In der vorliegenden Arbeit wird die Integration verschiedener Geräte in Home Assistant untersucht und dokumentiert. Dabei liegt der Fokus auf der Einbindung von LaMetric Time, einer Philips Hue Bridge mit zwei Sensoren und einer Lampe sowie einem Sonoff Display. Ziel ist es, die Einrichtung, Funktionsweise und mögliche Herausforderungen bei der Implementierung detailliert zu analysieren und anschaulich darzustellen. Neben der Betrachtung von technischen Abläufen werden ebenfalls passende Lösungsansätze für potentielle Probleme aufgezeigt.

Durch diese Arbeit sollen nicht nur die technischen Aspekte beleuchtet, sondern auch die Vorteile einer zentralisierten Smart-Home-Steuerung verdeutlicht werden. Ein weiterer Schwerpunkt ist die kritische Betrachtung des Home Assistant als System. Dabei stellen sich zentrale Fragen wie: Welchen Mehrwert kann Home Assistant einem durchschnittlichen Schweizer Haushalt tatsächlich bieten? Wo finde ich persönlich den grössten Nutzen in meinem Alltag? Ist das System Marktauglich?

Durch die Verknüpfung von theoretischen Grundlagen sowie Erfahrungen aus der Praxis werden die Herausforderungen und Möglichkeiten des Systems erkundet, analysiert und aufgezeigt. Schrittweise wird die Integration verschiedener Geräte untersucht und die praktische Umsetzung im Detail betrachtet. Dabei werden sowohl die Konfiguration der einzelnen Komponenten angeschaut als auch deren Interaktion innerhalb des Systems. Folglich wird analysiert, wie sich die Geräte und ihre dazugehörigen Entitäten miteinander verhalten.

Abschliessend werde ich meine Leitfragen beantworten. Dafür werde ich nicht nur meinen persönlichen Bezug zu Smart Home reflektieren und meine Ansichten zu Automatisierungen aufzeigen, sondern auch analysieren, wo die Grenzen von Home Assistant liegen. Dadurch soll ein umfassendes Bild darüber entstehen, inwiefern Home Assistant eine sinnvolle Ergänzung für das smarte Zuhause darstellen kann und welche Aspekte noch Entwicklungspotenzial bieten.

2 Home Assistant

2.1 Grundlagen

Home Assistant ist eine Open-Source-Plattform zur Hausautomatisierung, die es ermöglicht, verschiedene smarte Geräte zentral zu steuern und zu automatisieren. Das System ist lokal auf einem Server oder einem Mini-Computer wie beispielsweise einem Raspberry Pi oder dedizierter Hardware installierbar und bietet eine flexible Lösung zur Integration zahlreicher Geräte und Systeme. Durch die Unterstützung von über tausend verschiedenen Marken und Protokollen, erlaubt Home Assistant eine nahtlose Vernetzung unterschiedlichster Smart-Home-Komponenten.

Ein wesentlicher Aspekt von Home Assistant ist die Automatisierung. Benutzer können individuelle Szenarien erstellen, in denen Geräte auf bestimmte Ereignisse oder Sensorwerte reagieren sollen. So kann exemplarisch eine smarte Lampe automatisch eingeschaltet werden, wenn ein Bewegungsmelder eine Person erkennt, oder die Heizung kann basierend auf der Aussentemperatur und dem Zeitplan geregelt werden. Home Assistant ermöglicht zudem eine detaillierte Analyse und Visualisierung von Sensordaten sowie die Steuerung per Sprachbefehl über Assistenten wie Google Assistant oder Amazon Alexa.

Die Entwickler von Home Assistant verfolgen das Ziel, eine datenschutzfreundliche und lokal steuerbare Smart-Home-Plattform bereitzustellen. Im Gegensatz zu vielen kommerziellen Alternativen, die auf Cloud-Dienste angewiesen sind, funktioniert Home Assistant größtenteils lokal, was die Abhängigkeit von externen Servern reduziert, und die Privatsphäre der Nutzer schützt. Zudem soll die Plattform möglichst offen und flexibel gestaltet sein, sodass Nutzer mit Programmierkenntnissen eigene Erweiterungen und Integrationen entwickeln können.

Ein weiteres Ziel ist die langfristige Unabhängigkeit von einzelnen Herstellern und deren markenrechtlich geschützten Systemen. Durch die breite Unterstützung verschiedener Protokolle und Standards können Nutzer ihre Smart-Home-Geräte auch dann weiterhin verwenden, wenn ein Hersteller seinen Dienst einstellt oder sich aus dem Markt zurückzieht.

Zusammenfassend ist Home Assistant eine leistungsstarke, flexible und datenschutzfreundliche Lösung für Smart-Home-Enthusiasten. Es ermöglicht eine umfangreiche Steuerung und Automatisierung verschiedener Geräte und verfolgt das Ziel, den Nutzern eine unabhängige und lokale Smart-Home-Lösung bereitzustellen.

2.2 Installation und Ersteinrichtung

Die Installation von Home Assistant auf dem Home Assistant Green erfolgt in nur wenigen Schritten und sehr intuitiv. Nachdem das Gerät mit LAN angeschlossen und Strom versorgt wird, startet es automatisch. Der erste Zugriff erfolgt über die im lokalen Netzwerk zu findende IP-Adresse des Gerätes. Da die Home Assistant Software auf dem Home Assistant Green per Standard bereits installiert ist, muss keine Installationsarbeit gemacht werden. Dies ermöglicht ebenfalls, dass man nach Eingeben der IP-Adresse bereits mit der Ersteinrichtung starten kann.

Zu Beginn der Installation wird der Nutzer durch eine einfache Setup-Seite geführt, auf welcher grundlegende Einstellungen wie Sprache, Zeitzone und Standort konfiguriert werden. Nachdem diese initialen Konfigurationen abgeschlossen sind, kann Home Assistant entweder über Ethernet oder WLAN mit dem Internet verbunden werden, um eine stabile Verbindung sicherzustellen.

Ein weiterer Schritt in der Ersteinrichtung ist die Erstellung eines Benutzerkontos, um Zugriff auf das System zu erhalten. Sobald das Konto erstellt ist, erkennt Home Assistant automatisch viele gängige Geräte und bietet eine einfache Möglichkeit, diese hinzuzufügen. Im Falle von wenig unterstützten Geräten oder speziellen Integrationen können diese auch manuell über die Konfigurationsdateien von Home Assistant eingebunden werden, was fortgeschrittenen Nutzern zusätzliche Flexibilität bietet.

Besonders hervorzuheben ist die benutzerfreundliche Integration von Geräten. Home Assistant Green bietet eine Vielzahl von vorgefertigten Integrationen, die es dem Nutzer ermöglichen, eine breite Palette von Smart-Home-Produkten direkt in das System einzubinden. Diese Integrationen können schnell und unkompliziert hinzugefügt werden, wodurch der Nutzer sofort von der zentralen Steuerung und Automatisierung profitiert.

Nach der Ersteinrichtung bietet Home Assistant eine übersichtliche und anpassbare Benutzeroberfläche, die dem Nutzer hilft, das System nach seinen individuellen Bedürfnissen zu gestalten. Die Plattform ist so konzipiert, dass sie von Anfang an eine stabile und funktionale Smart-Home-Steuerung ermöglicht.

2.3 Dashboard

Die Konfiguration der verschiedenen Dashboards war für mich eine Aufgabe mit sehr hoher Priorität. Der Grund dafür liegt darin, dass die Dashboards in meinem Projekt die einzige Verbindung zwischen Home Assistant und dem Nutzer darstellen. Es ist besonders wichtig, dass diese Verbindung so reibungslos und benutzerfreundlich wie möglich gestaltet wird. Denn das Dashboard dient nicht nur als Steuerzentrale für alle Smart-Home-Geräte, sondern auch als visuelle Schnittstelle, die dem Nutzer hilft, das System intuitiv und effizient zu bedienen.

Ein gut gestaltetes Dashboard sorgt dafür, dass der Nutzer schnell und einfach auf alle wichtigen Funktionen zugreifen kann, ohne sich durch komplizierte Menüs oder unübersichtliche Einstellungen navigieren zu müssen. Besonders bei der Nutzung von Home Assistant in einem komplexen Smart-Home-System ist es entscheidend, dass die Benutzeroberfläche klar und übersichtlich bleibt. Dabei spielt es eine zentrale Rolle, dass wichtige Informationen und Steuerfunktionen direkt sichtbar sind und sich problemlos bedienen lassen. Durch eine solche, benutzerfreundliche Gestaltung wird das gesamte Nutzungserlebnis erheblich verbessert, und lässt den Nutzer gut informiert über sein Zuhause fühlen.

Die Erstellung von Dashboards, die sowohl funktional als auch ästhetisch ansprechend sind, kann die Nutzung von Home Assistant erheblich vereinfachen und dabei helfen, das volle Potenzial des Systems auszuschöpfen.

2.4 Fernzugriff

Der Fernzugriff auf Home Assistant ermöglicht es, das eigene Smart-Home-System von überall aus zu steuern, unabhängig davon, ob man sich zu Hause aufhält oder man unterwegs ist. Eine der einfachsten und sichersten Methoden dafür ist die Home Assistant Cloud über Nabu Casa. Diese Lösung stellt eine verschlüsselte Verbindung her, ohne dass komplizierte Netzwerkkonfigurationen oder das Freigeben von Ports im Router notwendig sind. Wer eine kostenlose Alternative bevorzugt, kann den Fernzugriff auch über eine dynamische DNS-Adresse und eine Portweiterleitung einrichten. Allerdings ist hierbei besondere Vorsicht geboten, da eine ungesicherte Verbindung ein erhebliches Sicherheitsrisiko darstellen kann. Daher sollte in diesem Fall stets ein SSL-Zertifikat, beispielsweise über Let's Encrypt, verwendet werden. Eine weitere Möglichkeit bietet ein VPN, damit eine sichere Verbindung direkt ins Heimnetzwerk hergestellt wird, ohne dass Home Assistant selbst aus dem Internet erreichbar sein muss. Diese Methode ist besonders sicher und verhindert potenzielle Angriffe von aussen.

Sobald der Fernzugriff aktiviert ist, lassen sich alle verbundenen Geräte von überall steuern, Automatisierungen flexibel anpassen und wichtige Benachrichtigungen in Echtzeit empfangen. So bleibt das gesamte Smart Home auch unterwegs unter Kontrolle, egal ob es um die Temperaturregelung, Lichtsteuerung oder Sicherheitskameras geht. Wer Home Assistant mit Sprachassistenten wie Alexa oder Google Assistant nutzen möchte, profitiert ebenfalls vom Fernzugriff, da viele Funktionen erst dadurch vollständig verfügbar werden. Damit die Verbindung jederzeit sicher bleibt, empfiehlt es sich, die Zwei-Faktor-Authentifizierung zu aktivieren und ausschliesslich verschlüsselte Verbindungen zu verwenden. So lassen sich Komfort und Sicherheit optimal miteinander verbinden.

2.5 HACS

HACS (Home Assistant Community Store) ist eine Erweiterung für Home Assistant, die es Nutzern ermöglicht, zusätzliche Integrationen, Automatisierungen und Designs aus der Community einfach zu installieren und zu verwalten.

Anstatt Erweiterungen manuell herunterzuladen und in das System zu integrieren, bietet HACS eine benutzerfreundliche Oberfläche, über die neue Funktionen mit wenigen Klicks hinzugefügt werden können. Zudem werden installierte Erweiterungen automatisch aktualisiert, sodass Nutzer stets von den neuesten Verbesserungen profitieren.

Durch HACS wird Home Assistant flexibler und individueller anpassbar, da eine grosse Auswahl an Community-Erweiterungen zur Verfügung steht. Es erleichtert die Nutzung von Drittanbieter-Erweiterungen und trägt zur stetigen Weiterentwicklung des Smart-Home-Systems bei.

Github

GitHub ist eine Plattform, die Entwicklern ermöglicht, ihre Projekte online zu speichern, zu verwalten und mit anderen zu teilen. Besonders im Bereich von Open-Source-Projekten, wie Home Assistant, spielt GitHub eine zentrale Rolle. Viele Erweiterungen und Integrationen, die von der Community entwickelt werden, sind auf GitHub verfügbar. Diese Erweiterungen bieten neue Funktionen, Anpassungen oder auch zusätzliche Integrationen, die nicht im Standardumfang von Home Assistant enthalten sind.

Wenn Nutzer von Home Assistant nach neuen Funktionen suchen, die ihre Smart-Home-Umgebung erweitern, sind sie oft auf GitHub unterwegs. Dort können sie den Quellcode einsehen, die Erweiterungen herunterladen und auf ihrem Home-Assistant-System installieren. Das Herunterladen und Einfügen der Erweiterungen in das eigene System erfordert manchmal technisches Verständnis, da die Integrationen häufig manuell in bestimmte Verzeichnisse kopiert und in Konfigurationsdateien eingetragen werden müssen.

Ein weiterer Vorteil von GitHub ist, dass die Nutzer die Entwicklung der Erweiterungen verfolgen können. GitHub bietet zudem eine Plattform für die Zusammenarbeit, bei

der andere Entwickler oder Nutzer Rückmeldungen geben und gemeinsam an der Weiterentwicklung von Erweiterungen arbeiten können.

Zusammengefasst ist GitHub für Home Assistant ein wichtiger Hub, um Erweiterungen und Integrationen zu finden und zu teilen. Es ermöglicht Entwicklern und Nutzern, das System zu erweitern und anzupassen, um so das volle Potenzial von Home Assistant auszuschöpfen.

3 Geräte und ihre Konnektivität zum Home Assistant

3.1 Sonoff Display NSPanel / Pro

Gleich zu Beginn meiner Geräte Vorstellung möchte ich mit der grössten Knacknuss beginnen. Gestartet habe ich mit dem normalen Sonoff NSPanel, mit dem Hintergedanken, dass auf den beiden Tasten jeweils Lichter oder andere Funktionen gesteuert werden können, was für den durchschnittlichen Haushalt am nutzerfreundlichsten ist.

Die Ersteinrichtung von NSPanel ist grundsätzlich nutzerfreundlich und einfach einzurichten. Sobald man die eWeLink App heruntergeladen und einen entsprechenden Account erstellt hat, wird man sehr angenehm durch die Einrichtung geleitet. Parallel muss man auf dem Home Assistant das eWeLink Smart Home Add-On hinzufügen und sich mit denselben Account daten anmelden. Sobald das heruntergeladen und auf dem neusten Stand ist, kann man dann beginnen, die zu bedienenden SmartHome Geräte in das Add-On einzubinden, was zunächst auch bei mir gut funktioniert hat. Als ich dann aber anfangen wollte Verknüpfungen zu machen haben sich grosse Probleme erwiesen. Auf dem NSPanel wurden die Leuchten immer wieder als nicht verbindbar angezeigt. Das hat erschwert, verschiedene Befehle oder Automationen über das NSPanel überhaupt zu testen.

Nach längerer Recherchearbeit bin ich dann auf ein Youtube Video von «Mark Watt Tech» gestossen. Er hat eine Schritt für Schritt Anleitung verfilmt hat, wie man diese Verbindungsprobleme umgehen könne. Konkret zeigt er auf wie man mithilfe des ESP Home Add-On das NSPanel flashen kann. Daraufhin habe ich direkt ein entsprechendes USB Jumper-Kabel Set bestellt, um dieser Anleitung nachzugehen. Als das Set dann angekommen

ist, habe ich die Platine des Panels vom Gehäuse getrennt und an den lokalen PC angeschlossen, was schlussendlich wie folgt ausgesehen hat:

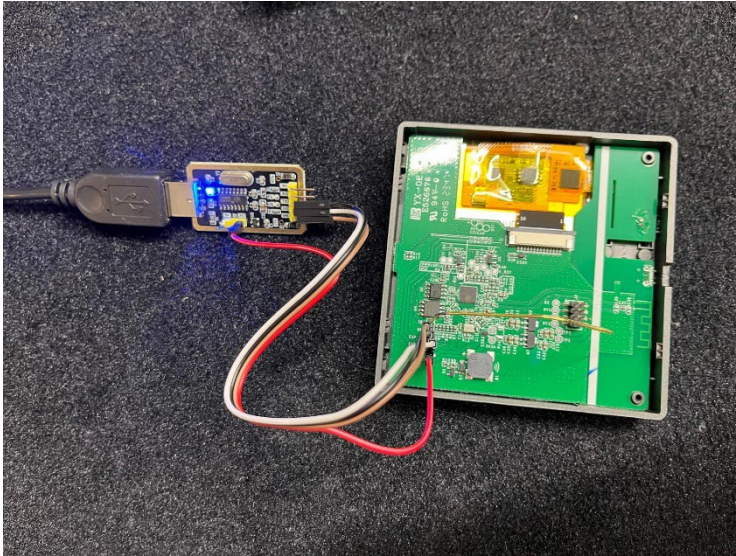


Bild 1: Eigene Aufnahme, 21.03.2025

Für die Konfiguration habe ich dann im ESP Home ein neues Projekt erstellt, in welchem es das NSPanel gleich erkannt hat. Auf der Github-Webseite habe ich daraufhin die richtige TFT Datei gefunden und bereit gemacht, um zu flashen. Dann ist das Unerwartete passiert. Sobald ich die Datei für den Flash freigeben habe, hat der PC eine beunruhigende Fehlermeldung gezeigt und der USB Jumper hat ein wenig Rauch von sich gegeben, bevor ich ihn dann rasch ausgesteckt habe. Das Teilprojekt vom Flashen des Panels wurde somit vorerst auf Eis gelegt. Da ich zu diesem Zeitpunkt aber schon sehr viel Zeit in das Projekt gesteckt hatte, hinterfragte ich nochmals meinen Prozess und habe nochmals alle Arbeitsschritte überprüft – leider erfolglos. Auch wenn es schwierig war mich von den physischen Tasten des NSPanel zu trennen war es an der Zeit, mit dem NSPanel Pro fortzufahren, um dennoch eine benutzerfreundliche Bedienung für den Home Assistant zu bieten.

Die Einrichtung vom NSPanel Pro verläuft dabei ähnlich wie bei dem Standard NSPanel und zwar über die eWeLink App. Jedoch muss anders vorgegangen werden, um das Dashboard auf das Panel zu bringen. In der eigenen Home Assistant Cloud findet man einen Fernzugriff Nabu Casa URL, welchen man sich in der Zwischenablage abspeichern muss.

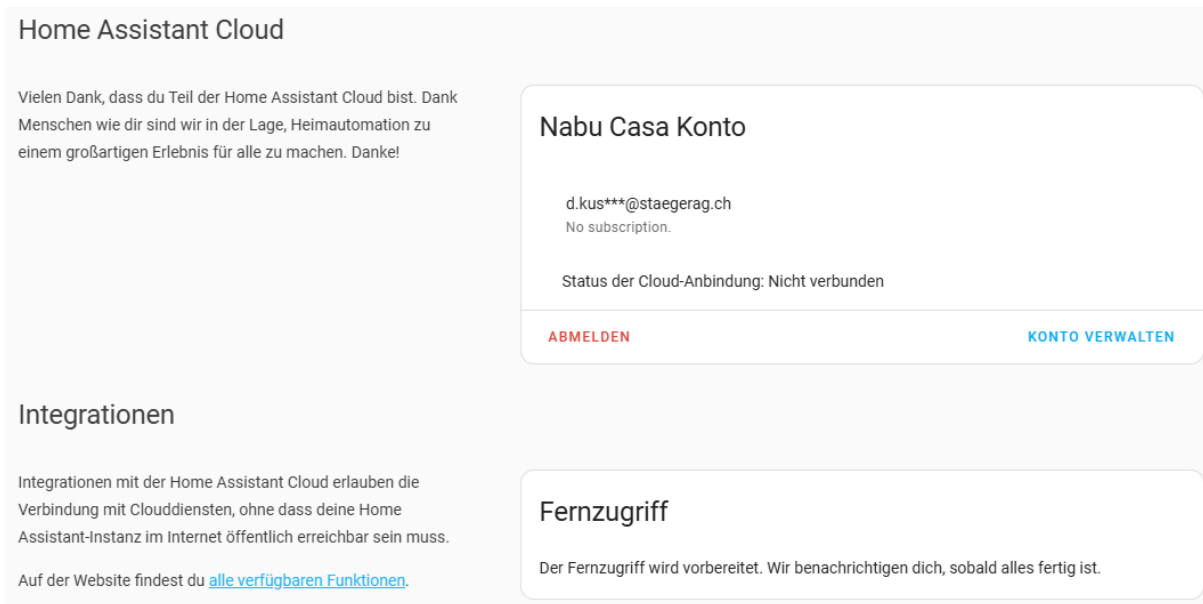


Bild 2: Screenshot, 11.04.2025

Daraufhin wechselt man wieder in die eWeLink App und wählt das zu konfigurierende NSPanel an. In den tieferen Einstellungen findet man das Webpages Icon und wählt dieses an. Dort kann man dann ein neues Icon mit dem vorhin kopierten URL erstellen.

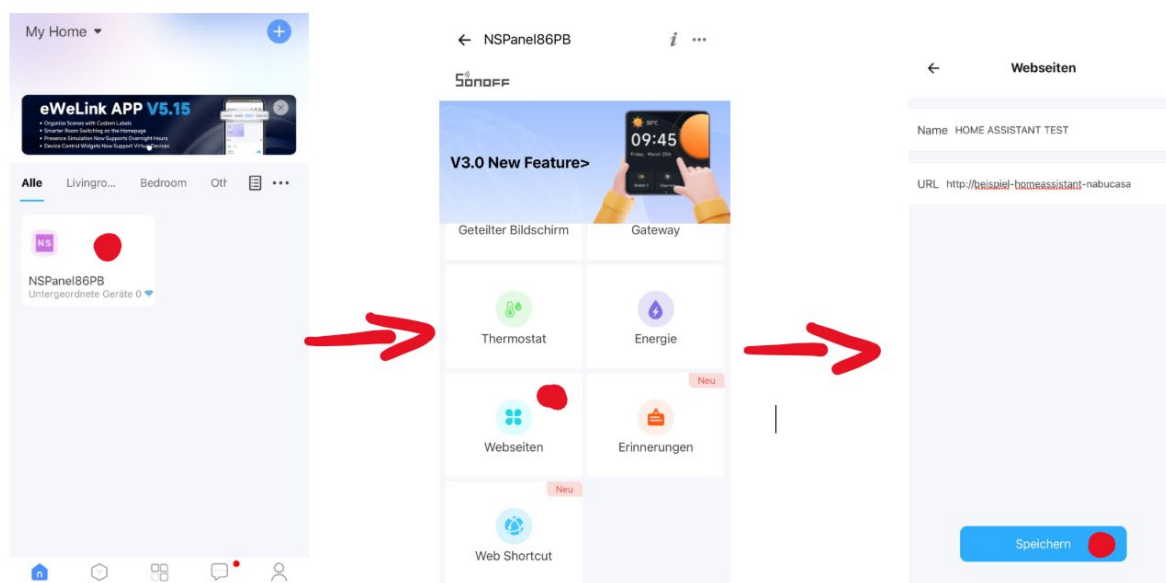


Bild 3: Screenshot, 11.04.2025

Sobald das abgespeichert ist, findet man auf dem Panel das Webpage Icon wieder. Wenn man dieses dann anwählt, kommt man auf die Übersicht vom Home Assistant. Dort habe ich noch das vorgefertigte NSPanel Dashboard als Standard Dashboard festgelegt, sodass es dann so ausgesehen hat:

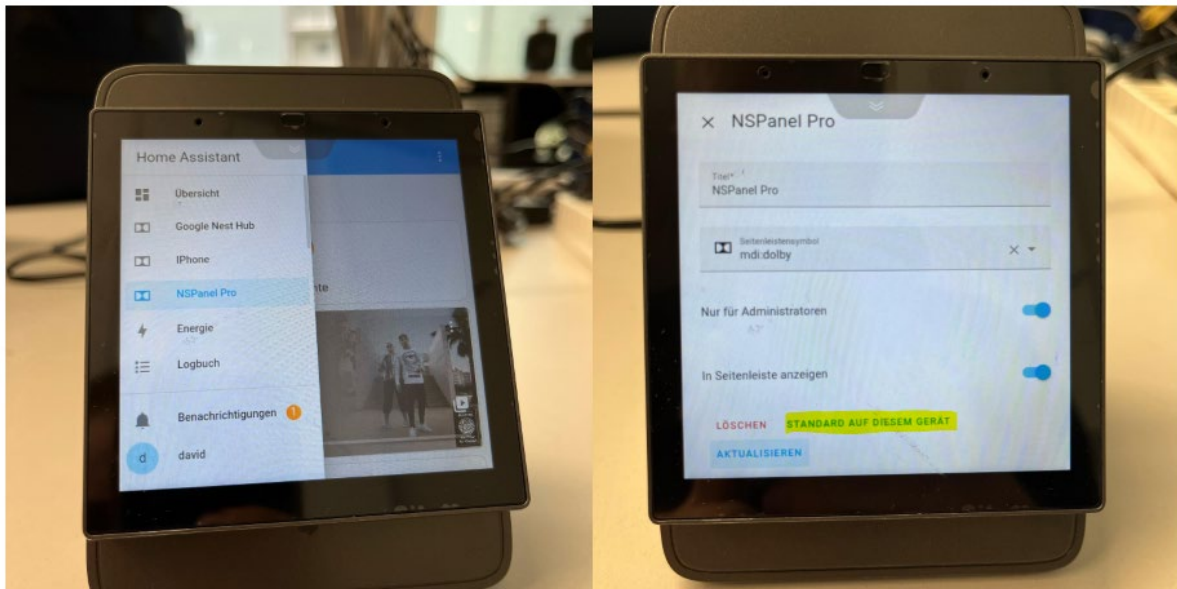


Bild 4 Eigene Aufnahme, 15.04.2025

Somit ist die Konfiguration erfolgreich abgeschlossen. Die Bedienung über das NSPanel Pro hat mich bei den ersten Versuchen schon sehr überzeugt. Die Reibungslosigkeit und Funktionalität waren mir enorm wichtig. Nach den fehlerhaften Tests mit den normalen NSPanel hat mich dies besonders erfreut und überzeugt.

3.2 Google Nest Hub

Die Idee des Google Nest Hub war, zunächst nur einen Audio Streamer zu simulieren. In welcher Form, Design und von welchem Hersteller das Gerät sein sollte, war für mich zu Beginn der Arbeit noch nicht relevant. Auf meiner Suche nach einem geeigneten Audio Streamer bin ich dann auf den Google Nest Hub gestossen. Dieser hat neben einer eingebauten Audioeinheit auch noch ein sieben Zoll Touchdisplay was sich im Verlauf meiner Arbeit ebenfalls als sehr interessant herausstellte.

Die Inbetriebnahme des Google Nest Hub erfolgt grundsätzlich und ohne Ausnahme über die Google Home App. Sobald das Gerät zum allerersten Mal eingeschaltet und vollständig hochgefahren wird, erscheint automatisch ein QR-Code auf dem Display. Dieser QR-Code ist speziell dafür vorgesehen, die initiale Konfiguration direkt über die Google Home App auf einem Smartphone oder Tablet zu starten. Nachdem der QR-Code mit der App erfolgreich gescannt wird, beginnt der Einrichtungsprozess, der den Benutzer schrittweise durch sämtliche notwendige Einstellungen führt. Dabei werden unter anderem wichtige Informationen wie die Zugangsdaten für das WLAN-Netzwerk abgefragt und eingegeben, damit das Gerät korrekt ins Heimnetzwerk integriert werden kann und alle Funktionen reibungslos zur Verfügung stehen.

Um die Verbindung mit dem Home Assistant herzustellen, fügt man zunächst die Integration «Google Cast» zum Home Assistant hinzu. Direkt nachdem die Integration erledigt ist, erscheint ein Fenster, das nach der IP-Adresse des Zielgerätes fragt, welche über die Google Home App oder über einen Lokalen IP-Scann aufzufinden ist. Folglich bekommt man die Möglichkeit dieses Gerät direkt einem Raum zuzuweisen, was in meinem Fall aber nicht nötig war. Mit diesem Schritt ist dann die Verknüpfung zum Home Assistant bereits gemacht und man kann beginnen, die verschiedenen Entitäten und Dashboard Darstellungsmöglichkeiten zu entdecken. Mit dieser Grundeinrichtung kann man bereits Mediaplayer Karten darstellen, mit denen eingestellt werden kann, was und wie laut etwas auf dem Google Nest Hub abgespielt.

Zu diesem Zeitpunkt war ich jedoch noch nicht zufrieden mit den Nutzungsmöglichkeiten des Nest Hub, da ich meine ebenfalls integrierten Smart Home Produkte nicht steuern konnte. Ich wollte eine Möglichkeit finden, mein Home Assistant Dashboard auf den Nest Hub zu casten und von dort aus auch meine

anderen eingebundenen Geräte zu steuern. Auf der Suche nach attraktiven Lösungen bin ich auf ein hilfreiches Video vom YouTube Kanal «EverythingSmartHome» gestossen. In diesem Video geht es um die Einrichtung und Integration von Google Assistant SDK. SDK steht für Software Development Kit und bietet eine breite Fassung an Varianten, Sprachbefehle in den Home Assistant einzubinden.

Die Einrichtung der Google Assistant SDK Integration verläuft anfangs über die Google Cloud vom selben Google Konto, mit dem man die eigentliche Installation des Nest Hub gemacht hat. Dort kann man ein neues Projekt eröffnen und den Google Assistant API aktivieren, um Home Assistant mit dem Google Assistant zu steuern. Unter dem Menüpunkt «Auth-Zustimmungsbildschirm» trifft man diverse Einstellungen, die am Schluss zur neuen Google Assistant SDK Integration im Home Assistant führen.

Diese Integration ermöglicht es, Google Assistant direkt mit Home Assistant zu verbinden, sodass Smart-Home-Geräte per Sprachbefehl gesteuert werden können. Mit der Google SDK-Integration kann Home Assistant so angepasst und eingerichtet werden, dass Geräte und Automationen über Google Assistant gesteuert werden können. Dadurch können Lichter, Thermostate und komplexe Automationen einfach per Sprachbefehl aktiviert und dadurch im eigenen Smart Home bequem und intuitiv bedient werden. Ein Praxisbeispiel folgt im Abschnitt 4 «Automationen und Anwendungen».

Die letzte Lösung, die ich dann noch finden wollte, beschäftigte sich damit, wie ich das Dashboard bequem auf den Google Nest Hub casten kann. Gelöst habe ich das Problem wie folgt:

In den Entwicklertools vom Home Assistant unter der Kategorie Aktionen, habe ich diverse Tests mit den neuen Befehlen vom Google Assistant SDK durchgeführt, wobei ich auf diverse andere Google Befehle gestossen bin. So zum Beispiel auf den Befehl «Google Cast: Dashboard-Ansicht anzeigen»

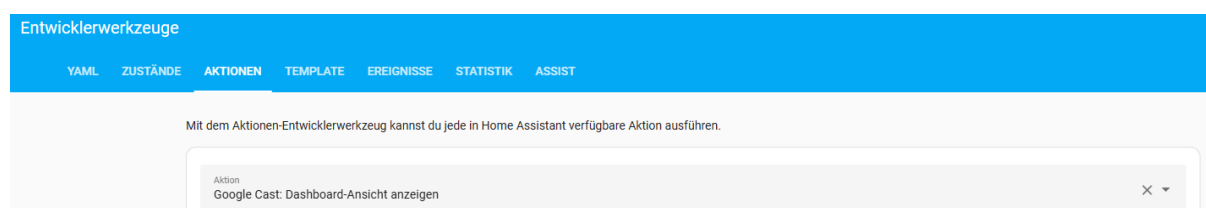


Bild 5: Screenshot, 11.04.2025

Dieser Befehl gestattet es mit richtigen Angaben von Entität, Dashboard- und URL-Pfad auf Knopfdruck auf das angegebene Dashboard den Google Nest Hub zu casten und von dort aus die weiteren Smart Home Geräte zu steuern. In meinem Fall ist mit dem im Schlafzimmer platzierten Google Nest Hub die korrekte Entität gefunden. Der Dashboard-Pfad findet man, wenn man das Dashboard auswählt und dann die Webadresse anschaut. In meinem Beispiel wäre das: <http://homeassistant.local:8123/dashboard-dashboard/0>

Der Part von «dashboard-dashboard» ist der Dashboard Pfad. Um den Ansichts-Pfad zu finden, muss man nochmals sein Zieldashboard öffnen. Unter jedem Dashboard kann man mehrere Ansichten machen, wenn man mehrere Gerätelayouts berücksichtigen möchte. Meine Zielansicht ist mit GOOGLE NEST benannt worden, was auch gleich den richtigen Ansichts-Pfad darstellt. So sieht also zum Schluss unsere «cast bereite» Aktion aus:

Aktion
Google Cast: Dashboard-Ansicht anzeigen

Zeigt eine Dashboard-Ansicht auf einem Chromecast-Gerät.

Entität
Mediaplayer-Entität, auf der die Dashboard-Ansicht angezeigt werden soll. Entität*
Schlafzimmer

Dashboard-Pfad
Der URL-Pfad des anzuzeigenden Dashboards, standardmäßig „lovelace“, wenn nicht angegeben. dashboard-dashboard

Ansichts-Pfad
Der URL-Pfad der anzuzeigenden Dashboard-Ansicht. GOOGLE NEST

[ZUM YAML-MODUS](#) AKTION AUSFÜHREN

Bild 6: Screenshot, 11.04.2025

Mit dem Drücken des «Aktion Ausführen» Buttons castet man also von hier aus das Google Nest spezifische Dashboard auf den Google Nest Hub und kann dort auch alles steuern, was im Dashboard integriert ist.

Damit war für mich die Konfiguration des Googles Nest Hub komplett und erfolgreich abgeschlossen. In der Praxis hat mir das Gerät sehr viel Freude bereitet. Da es für mich neu war, Sprachbefehle zu senden, war es umso spannender zu sehen, was man in Verknüpfung zu Home Assistant daraus machen kann.

3.3 LaMetric Time

Auf den LaMetric Time habe ich mich bereits vor Anfang der Arbeit gefreut, denn ich habe ihn vor mehreren Jahren bereits bei einem Kunden entdeckt und wollte seit jeher mehr darüber erfahren. Damals zeigte er sich noch als einfaches Uhren- und Anzeigedisplay, heute aber als intelligentes, homeassistantfähiges Gerät.

Die Ersteinrichtung überzeugt mit einer klaren Struktur und Übersichtlichkeit. Zunächst muss der LaMetric Time mit dem Stromnetz verbunden und eingeschaltet werden. Sobald das Gerät startet, zeigt es eine Begrüßungsnachricht und ein Setup-Icon an. Der nächste Schritt besteht darin, die LaMetric Time App herunterzuladen, die sowohl für iOS als auch für Android verfügbar ist. Nach dem Öffnen der App wird der LaMetric Time durch Scannen des angezeigten QR-Codes in der App mit dem WLAN verbunden.

Sobald die Verbindung zum Netzwerk hergestellt ist, erfolgt die automatische Synchronisierung der Uhrzeit und des Datums. Danach kann der Nutzer das Gerät an seine Bedürfnisse anpassen – etwa durch Auswahl von verschiedenen Widgets und Display-Layouts. Hierbei bietet der LaMetric Time eine Vielzahl von Anzeigemöglichkeiten, die von Uhrzeit und Wetterdaten bis hin zu benutzerdefinierten Benachrichtigungen reichen.

Für die Integration in Home Assistant ist der LaMetric Time über die Home Assistant-Integrationen schnell einsatzbereit. Nach der Installation der LaMetric Time-Integration im Home Assistant müssen nur noch die Verbindungsdaten eingegeben werden, und das Gerät wird sofort als Entität erkannt und kann mit Automatisierungen und Szenarien genutzt werden.



Bild 7: solarmarkt.ch, 31.03.2025

3.4 Phillips Hue

Die Integration von Phillips Hue erwies sich ebenfalls als durchaus nutzerfreundlich und simpel. Ähnlich wie bei der LaMetric Time gilt es zuerst die einzelnen Geräte in der Phillips Hue eigenen App auf dem Smartphone einzurichten. Dabei ist es wichtig, dass man den einzelnen Geräten sinnvolle und nachvollziehbare Namen verteilt. Denn bei grösseren Haushalten gibt es in der Regel sehr viele Lampen, Bewegungsmelder und weitere Leuchtmittel, wobei es essenziell ist einen guten Überblick behalten zu können. Meine Geräte habe ich wie folgt bezeichnet:

Bewegungsmelder 1 - Indoor

Bewegungsmelder 2 - Outdoor

Lampe – Ambiente

Wenn die Grundeinrichtung erledigt ist, kann man wieder zum Home Assistant zu den Integrationen wechseln. Dort gibt es die Phillips Hue Integration. Sobald diese installiert ist bekommt man sofort eine Liste mit allen Geräten, Diensten und Entitäten.

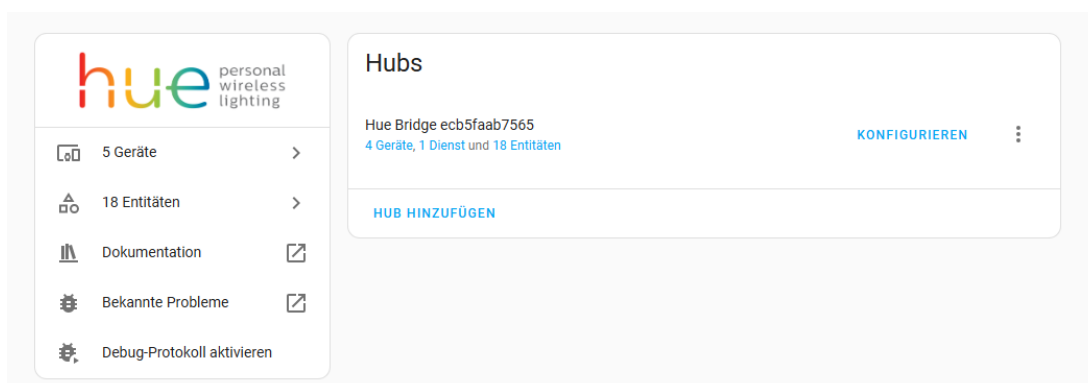


Bild 8: Screenshot, 11.04.2025

Das war tatsächlich auch schon die komplette Konfiguration von Phillips Hue im Home Assistant. Interessant war es, Versuche mit den verschiedenen Entitäten der Geräte von Phillips Hue zu machen. Denn die Bewegungsmeldern melden beispielsweise nicht nur Bewegung, sondern haben auch eingebaute Thermostate, die die aktuelle Temperatur angeben, was beim Outdoor-Sensor sehr interessant sein kann.

Zusammengefasst ist die Konfiguration von Phillips Hue sehr einfach und überzeugend. Die Möglichkeiten nach der Einrichtung sind grenzenlos, was vorteilhaft sein kann, jedoch auch ein ewiges Verweilen und Ausprobieren bedeuten kann.

4 Aufbau

Um einen umfassenden Überblick über die Struktur und Funktionsweise meiner Installation zu verschaffen, habe ich ein detailliertes Schema angefertigt. Dieses Schaubild soll nicht nur der eigenen Orientierung dienen, sondern auch jemandem der der Home Assistant Thematik fremd ist, einen einfacheren Einstieg gewährleisten.

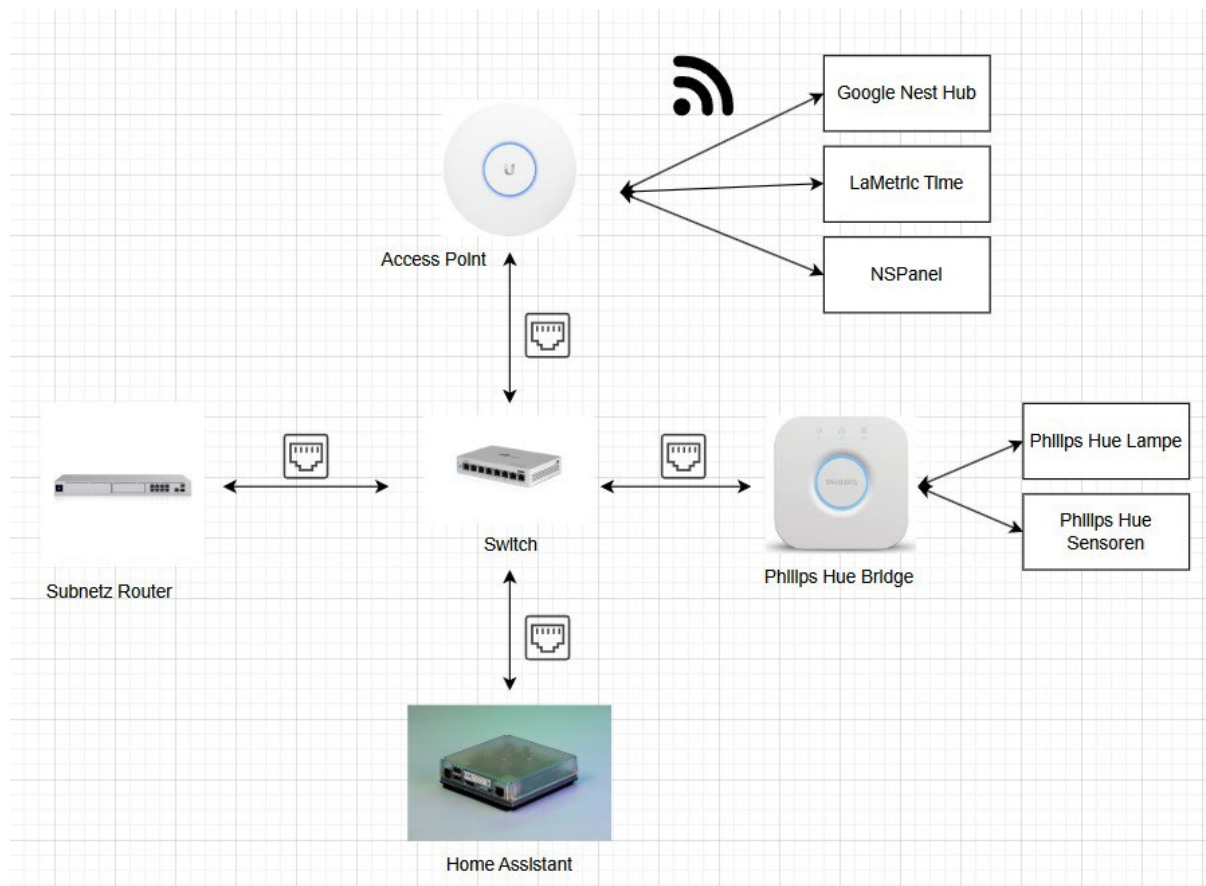


Bild 9: Eigene Zeichnung, 17.04.2025

Es zeigt die Ethernet Verbindungen jeweils mit einem entsprechenden Port an. Der Google Nest Hub, die LaMetric Time und das NSPanel sind per WLAN auf den Access Point verbunden. Dieser ist wie die Phillips Hue Bridge und der Home Assistant per LAN am Unifi Switch angeschlossen. Die Hue Bridge kommuniziert mit den dazugehörigen Sensoren und der Lampe über ZigBee.

5 Automatisierungen und Anwendungen

In dem folgenden Abschnitt habe ich meinen persönlichen Alltag genauer unter die Lupe genommen und kritisch hinterfragt. Mein Ziel war es, herauszufinden, ob es wiederkehrende oder alltägliche Aufgaben gibt, die mir der Home Assistant erleichtern oder vielleicht sogar vollständig automatisieren kann. Ich wollte herausarbeiten, in welchen Bereichen ich durch den Einsatz dieser Smart-Home-Lösung Zeit sparen und meinen Alltag komfortabler gestalten könnte. Nach einer ausführlichen Analyse meiner täglichen Abläufe habe ich mich schliesslich auf drei spezifische Automationen konzentriert, die für mich einen besonderen Mehrwert bieten: den Katzenmelder, den Wettercheck und die Batteriewechsel-Erinnerung. Diese drei Automationen schienen mir am sinnvollsten, da sie sich gut in meinen Alltag integrieren lassen und mir tatsächlich einen praktischen Nutzen bieten.

Bei der Erstellung dieser Automationen habe ich alle im vorherigen Abschnitt beschriebenen Geräte und Technologien eingesetzt. Dies bedeutete, dass ich die verschiedenen Sensoren, Aktoren und Schnittstellen, die ich bereits zuvor eingerichtet hatte, aktiv in meine Automatisierungen einbinden konnte. Dadurch konnte ich sicherstellen, dass jede dieser Automationen nicht nur zuverlässig funktioniert, sondern auch optimal an meine individuellen Bedürfnisse angepasst ist. Durch diese gezielte Auseinandersetzung mit möglichen Automationen wurde mir bewusst, wie viele Prozesse sich tatsächlich automatisieren lassen und wie Home Assistant dazu beitragen kann, mein Zuhause noch smarter und effizienter zu gestalten. Automationen folgen in der Regel einem klar definierten Schema und sind immer ähnlich aufgebaut. Zunächst benötigt jede Automation einen bestimmten Auslöser, der sie in Gang setzt und dafür sorgt, dass der nachfolgende Prozess überhaupt gestartet wird. Dieser Auslöser kann ein bestimmtes Ereignis oder eine Bedingung sein – in unserer Darstellung bezeichnen wir ihn als «X». Sobald «X» eintritt, wird die Automation aktiv und führt eine vordefinierte Aktion aus. Damit eine Automation nicht nur ausgelöst, sondern auch tatsächlich wirksam wird, benötigt sie einen weiteren Bestandteil: die auszuführende Aktion selbst. Diese bezeichnen wir als «Y». Das Grundprinzip lässt sich somit einfach zusammenfassen: Immer, wenn das Ereignis «X» eintritt, wird als Reaktion darauf die Aktion «Y» ausgelöst.

5.1 Katzenmelder

Die Idee der ersten Automation des «Katzenmelder» ist eine, die ich schon von Anfang an hatte. Obwohl wir in unserer Familie keine Haustiere haben, sind unsere Nachbarn sehr stolze Katzenbesitzer. Ebenfalls sind sie bereits mit einem SmartHome ausgestattet, mit welchem sie Lichter, Haushaltsgeräte und Rollläden steuern. Als sie dann bei einem gemeinsamen Abendessen von meiner IPA erfahren haben, wollten sie daraufhin sofort wissen, ob ich eine Möglichkeit finde ihnen aufzeigen zu können, wann ihre Katzen das Zuhause verlassen oder betreten.

Nachdem dann alle Geräte eingebunden waren, habe ich mich sofort an die Umsetzung dieser Automation gemacht. Bevor ich nach dem richtigen Auslöser gesucht habe, wollte ich wissen, wie man am besten benachrichtigt wird, wenn die Katze das Haus verlässt oder betritt. Nach kurzer Recherche bin ich auf die Notifikation Applikation von der LaMetric Time gestossen. Mit dieser kann man nämlich nach einem beliebigen Auslöser eine Textnachricht aufpoppen lassen mit individuellem Sound und Icon. Mit dieser Information galt es zunächst alles auszutesten und die geeignetsten Sounds und Icons, natürlich in Kombination einer kreativen Textnachricht zu finden. Auf der Developer Seite von LaMetric habe ich mich dann für das vorprogrammierte Icon #16009 «Mimi the Cat» entschieden. Über die Home Assistant Seite habe ich noch den richtigen Sound «Cat» gefunden. Ich konnte mich also auf die Suche nach einem passenden Auslöser für diese Automation machen.

Nach einigen kurzen Überlegungen bin ich zu dem Entschluss gekommen, einen einfachen Bewegungsmelder von Philips Hue als Auslöser für meine erste Automation zu verwenden. Dieses Gerät bietet eine unkomplizierte und zugleich effektive Möglichkeit, um die gewünschten Funktionen zu testen und darzustellen. Der Bewegungsmelder eignet sich besonders gut dafür, weil er den Idealfall eines smarten, also mit dem Smart Home-System kompatiblen Katzentürchens, auf möglichst realistische Weise simulieren kann. Immer dann, wenn der Sensor eine Bewegung erkennt, soll automatisch eine passende Benachrichtigung auf dem LaMetric Time Display erscheinen.

Die Entitäten für die Umsetzung waren eindeutig und dem entsprechend auch schnell gefunden. Für den Auslöser des Phillips Hue Bewegungsmelder namens Outdoor passt die Entität «Sobald Outdoor Bewegung meldet» perfekt. Beim LaMetric wurde es dann schon ein wenig herausfordernder, denn zunächst habe ich nur die Entität der Benachrichtigung mit dem richtigen Sound und Icon programmiert, was zunächst auch super funktioniert hat. Während den ersten Versuchen habe ich dann etwas Unvorteilhaftes bei dieser Automation entdeckt. Es hat sich herausgestellt, dass die LaMetric Time nach dem Auslösen den Sound immer auf der vorherigen Lautstärke abspielte. Das heisst, wenn man laut Musik hört und die Katze das Haus verlässt, erklingt ein ebenso lautes Miau einmal über die Lautsprecher, was kaum wünschenswert ist. Infolgedessen habe ich vor dem Benachrichtigung Baustein ein Lautstärkeregelung einfügen müssen. Dieser konnte mit der Entität «Stelle den Wert für TIME Lautstärke ein» bestmöglich erstellt werden. Denn dieser Baustein setzt die Lautstärke auf vier von zehn Lautstärkepunkten und somit auf eine klanglich ansprechende und passende Lautstärke. Auf der Automation Ansicht im Home Assistant hat das Ganze dann so ausgesehen:

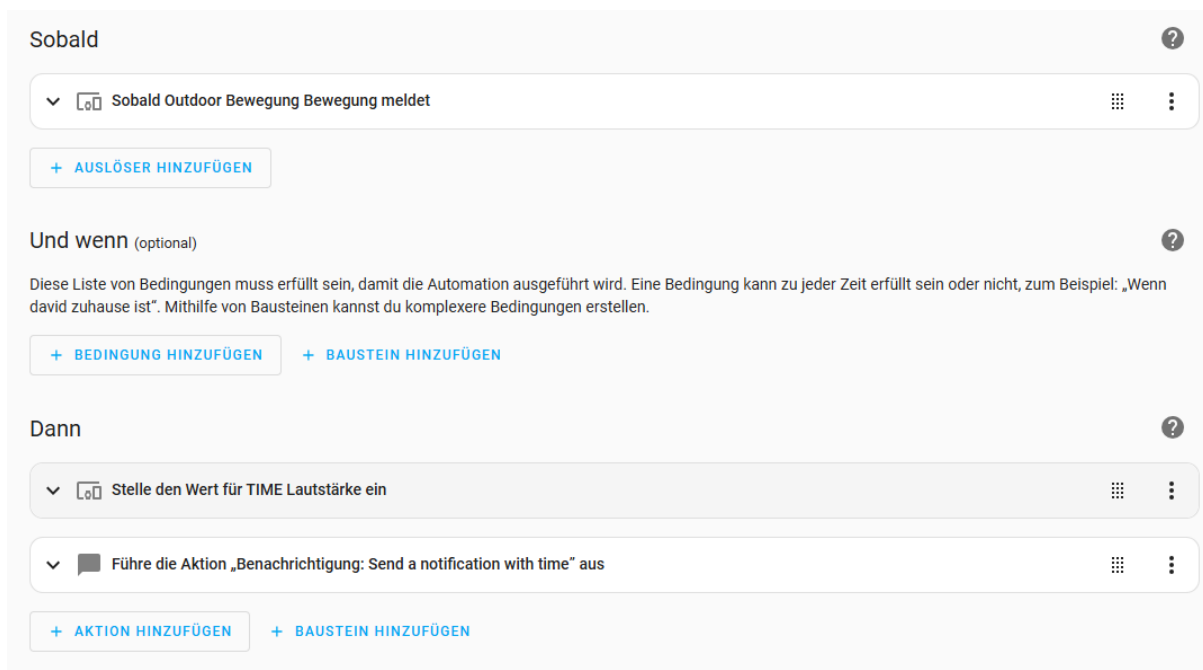


Bild 9: Screenshot, 11.04.2025

Wenn man dann noch etwas tiefer in die Programmierung von der Benachrichtigung geht, sieht der Auslöser zusammengefasst so aus:

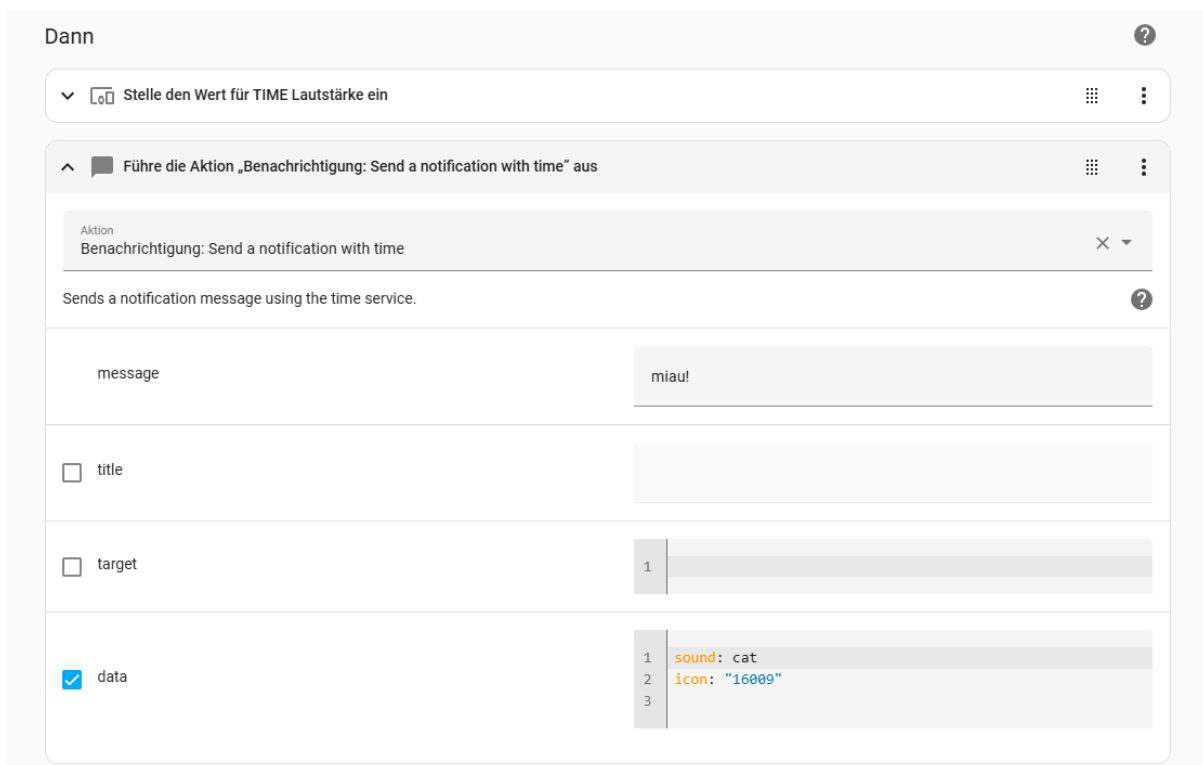


Bild 10: Screenshot, 11.04.2025

In der Praxis hat diese Automation sehr gut funktioniert. Zudem war es im Geschäft schnell ein Eyecatcher, denn alle haben sich gefragt, warum denn mein IPA-Rack «miaut», wenn man vorbeiläuft. Auch meine Nachbarn habe die Idee respektive die Umsetzung sehr positiv aufgenommen und überlegen sich gar, ihren Katzenmelder mit meiner IPA als Leitfaden zu realisieren.

5.2 Wettercheck

Jeden Morgen, wenn ich aufstehe, gibt es für mich nur eine Frage zu beantworten: «Wie ist das Wetter heute und muss ich regenresistente Kleidung anziehen?» Ich habe mich also gefragt, wie kann mir der Home Assistant diese Fragen also auf die für mich angenehmste Art beantworten. Mein Ziel war es einen Weg zu finden, der mir ohne ständiges Öffnen des Smartphones und trotzdem grosser Bequemlichkeit eben diese Fragen beantworten kann. So ist also der «Wettercheck» entstanden. Die dafür verwendeten Geräte sind der Phillips Hue Indoor Sensor, der Google Nest Hub und selbstverständlich der Home Assistant.

Angefangen habe ich damit, dass ich den Google Sprachassistenten genutzt habe, um mit Sprachbefehlen mir die aktuelle Wetterlage zu beschreiben. «Hey Google, wie wird das Wetter heute?» Die Antwort ist jeweils folgende: «In «Standort des Geräts» ist es aktuell «Wolken und Sonnenlage» bei Temperaturen von «X bis Y» Grad». Da ich aber nicht jeden Morgen mit einem virtuellen Assistenten sprechen möchte, brauchte ich einen passenden Auslöser. Der Indoor-Bewegungsmelder von Phillips hat sich für mich als sinnvolle Lösung erwiesen, um diese Automation in Bewegung zu setzen. Ich stelle mir vor, dass der Melder unauffällig hinter der Kleiderschranktür platziert ist und auslöst, sobald ich die Tür öffne und mich anziehen möchte – so kann er die Automation aktivieren und mir damit die Kleiderwahl am Morgen erleichtern.

Nach dem Auslöser braucht es bekannterweise auch eine Reaktion. Der Google Nest Hub bietet in der Richtung Sprachbefehl sehr viele Möglichkeiten und ist daher das beste Gerät für diese Automation. Zum Beispiel erlaubt mir die Google SDK Integration in Home Assistant, dass ich mit der Entität «Google Assistant SDK: Textbefehl senden» einen vorbereiteten Satz als Textbefehl an den Sprachassistenten schicken kann. Diese Entität stellt auch Fundament meiner «Wettercheck» Automation dar.

Nach zahlreichen Versuchen und Austesten von den diversen Möglichkeiten, die der Google Nest Hub bietet, bin ich bei diesem Aufbau gelandet:

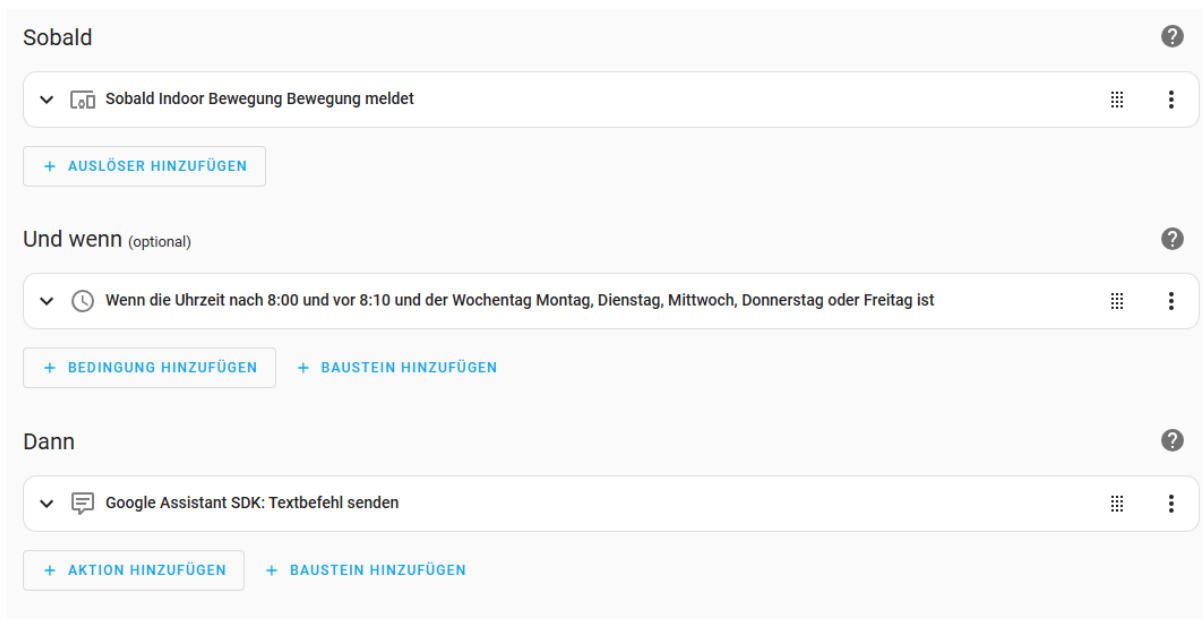


Bild 11: Screenshot, 11.04.2025

Man sieht, dass ich noch ein «und wenn» Baustein eingebaut habe. In meiner Konfiguration ist jener da, um sicherzustellen, dass der Google Nest Hub nur an den Arbeitstagen und ausschliesslich dann auslöst, wenn ich ihn brauche, also von 6:00 bis 6:10, weil das der Zeitraum in meiner Morgenroutine ist, indem ich mich anziehe und den Kleiderschrank öffne. Zudem möchte ich noch genauer auf den «dann» Baustein also auf die Reaktion zum Auslöser eingehen.

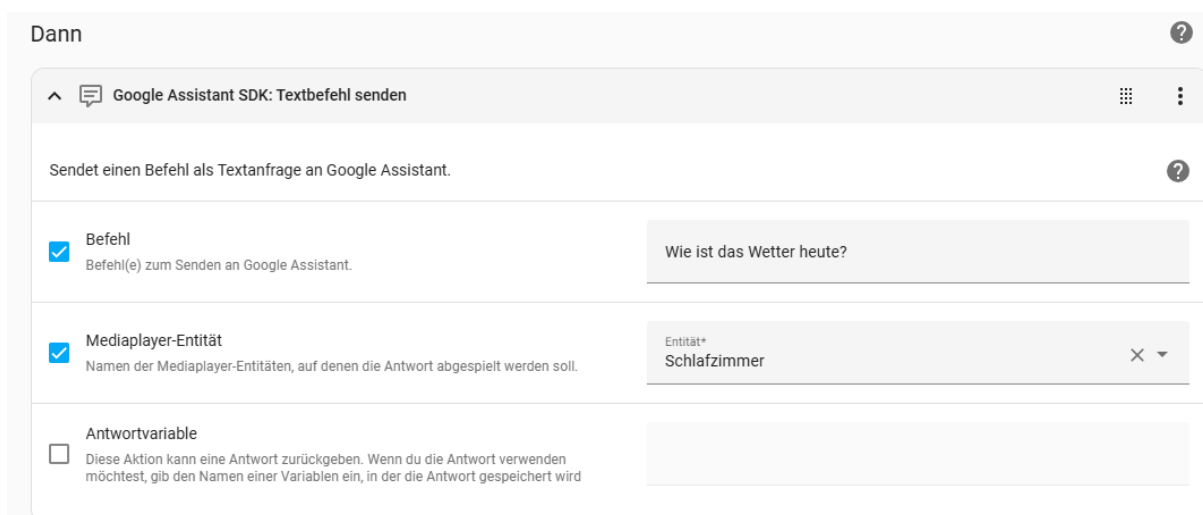


Bild 12: Screenshot, 11.04.2025

In diesem Baustein findet man hier den vorgeschriebenen Satz wieder. Unter Befehl könnte man dementsprechend alles eingeben, was man sonst den Google

Sprachassistenten mündlich fragen würde. Bei Mediaplayer-Entität muss man nur noch sein Zielgerät anwählen, was in meinem Fall klar der Google Nest Hub unter dem Namen «Schlafzimmer» ist.

Ähnlich wie der Katzenmelder hat sich auch der Wettercheck innerhalb kürzester Zeit als sehr nützlich und zuverlässig erwiesen. Dennoch hat mir auch diese Automation anfangs einige Schwierigkeiten bereitet, mit denen ich mich auseinandersetzen musste. Ein konkretes Beispiel dafür ist die Positionierung des Bewegungsmelders. Mir wurde klar, dass diese mit Bedacht gewählt werden muss, sodass der Sensor nicht auf jede kleinste Bewegung reagiert, sondern wirklich nur dann auslöst, wenn es auch beabsichtigt ist oder gebraucht wird.

Gerade in einem sensiblen Anwendungsfall wie diesem ist es wichtig, dass die Technik nicht zu aufdringlich reagiert, sondern gezielt zum Einsatz kommt. Rückblickend würde ich jedoch behaupten, dass mir auch diese Automation im Grossen und Ganzen sehr gut gelungen ist. Sie hat sich inzwischen als fester Bestandteil meines Alltags etabliert und trägt definitiv zu einem angenehmeren Start in den Tag bei.

5.3 Batteriewechsel

Erst während der Erarbeitung der anderen beiden Automationen ist mir aufgefallen, was für ein Nutzen eine Automation wie der Batteriewechsel haben kann. Meine zwei Bewegungsmelder Indoor und Outdoor waren oft und über lange Zeit in Betrieb und haben daher schneller als normale Bewegungsmelder an Batteriekapazität verloren. Wenn man das also auf einen fiktiven Haushalt hochrechnet, würde das bedeuten, dass der Katzenmelder oder Wettercheck plötzlich nicht mehr funktionieren würde. So ist die Idee des Batteriewechsels entstanden.

Mein Ziel ist es, eine Benachrichtigung einzurichten, die den Nutzer auffordert, die Batterien am jeweiligen Gerät zu wechseln. Für die Umsetzung wurden wiederum der Home Assistant, die LaMetric Time und die beiden Phillips Bewegungsmelder gebraucht.

Mit der Entität «Sobald sich der Batteriestand von Gerät ändert» war der passende Auslöser prompt gefunden. Wenn man tiefen in diese Entität hineinschaut kann man dort sagen, ob der Auslöser erst über oder unter einem gewissem Batteriestand auslösen soll. Im Beispiel mit dem Outdoor-Bewegungsmelder schaut das so aus:

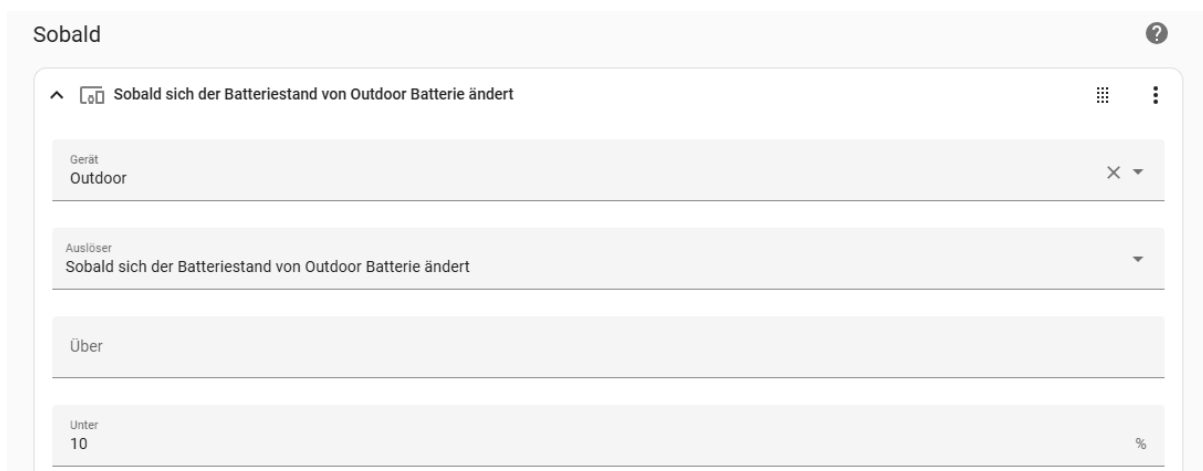


Bild 13: Screenshot, 11.04.2025

Der Auslöser war eine Art Repetition. Denn die Entität «Benachrichtigung: Send a notification with time» ist uns vom Katzenmelder schon bekannt. Gleichwohl habe ich ein entscheidendes Element hinzugefügt.

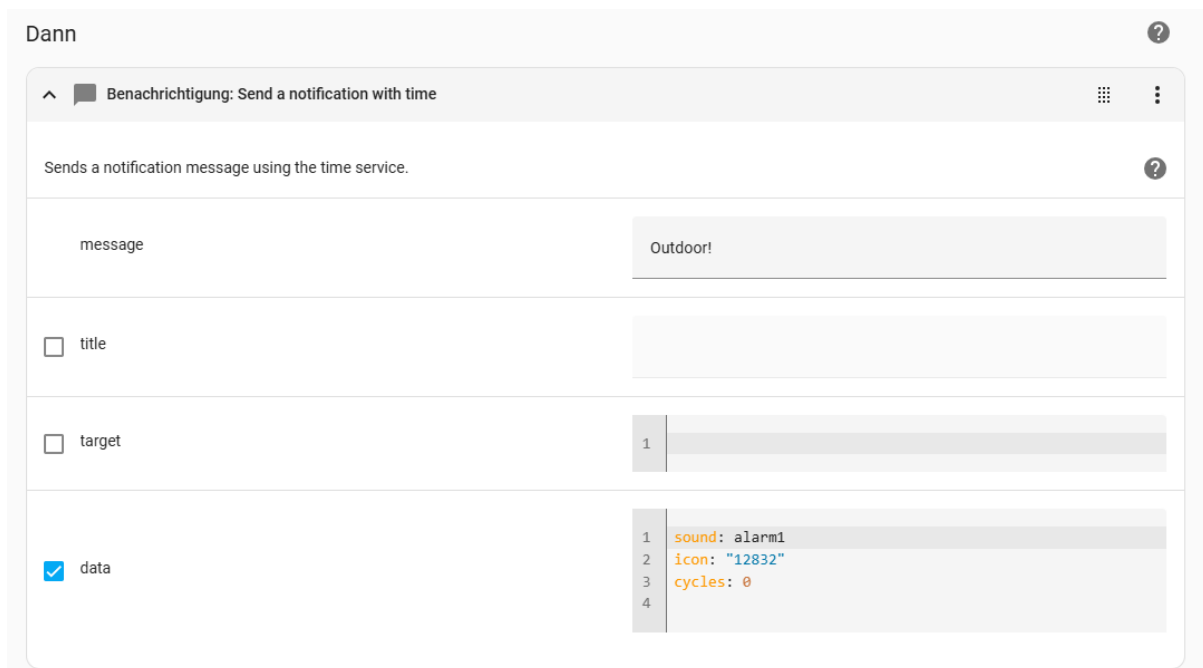


Bild 14: Screenshot, 11.04.2025

Der zusätzliche Code «Cycles» zeigt an, wie oft sich der Befehl auf der LaMetric Time wiederholen soll. Da es bei meinem Setting eher störend wäre, wenn sich der Alarm mehrmals wiederholt belasse ich es bei einer Laufzeit. Insgesamt erscheint der Batteriewechsel so:

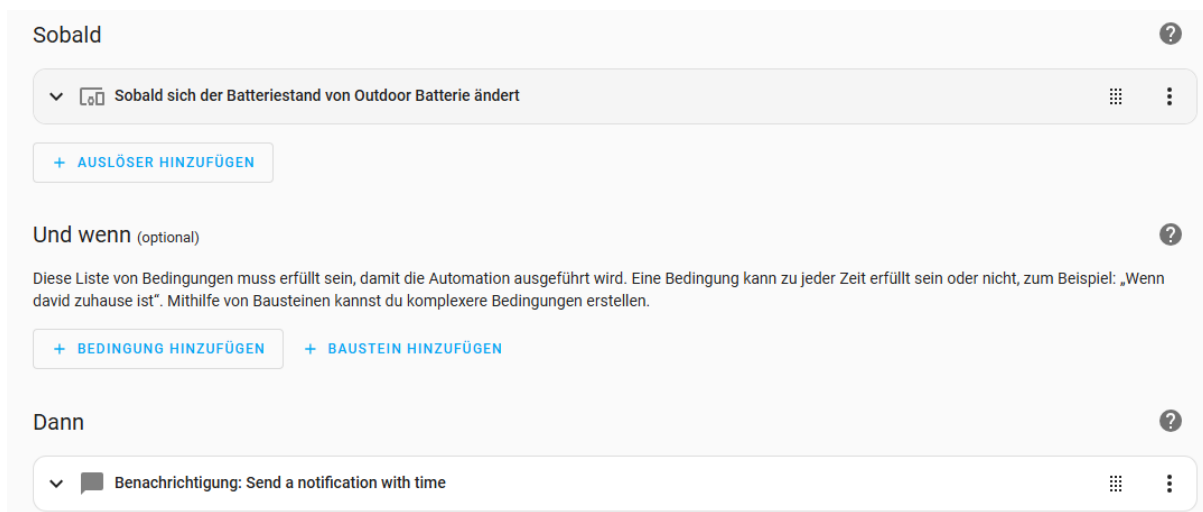


Bild 15: Screenshot, 11.04.2025

Meines Erachtens ist es so schlicht gehalten, aber dennoch essenziell für ein erfolgreiches Home Assistant Smart Home.

Als Zusatzleistung zum eigentlichen Batteriewechsel habe ich für mich selbst noch zwei ergänzende Automationen gemacht. Betitelt mit «ChargeUp» und «StopCharge». Diese beiden Zusatzaufgaben sind gedacht für die Langlebigkeit eines jeden Smartphone. Um den Akku meines iPhones zu schonen, möchte ich also das die LaMetric Time eine Benachrichtigung gibt, wenn der Akkustand über 90 oder unter 10 sind. Mit dem Ziel, dass das Smartphone dadurch einiges mehr an Laufzyklen durchmachen kann, bevor der Akku unbrauchbar wird.

Bei einem Akkustand von unter 10 greift nun also die Automation «ChargeUp» ein und die LaMetric Time zeigt «Laden» mit einem passenden Icon an. Gegenätzlich bei einem Akkustand von über 90 zeigt es «Genug» ebenfalls mit einem geeigneten Icon an, weil der «StopCharge» Befehl eingetroffen ist. Als Beispiel ist untenstehend die Automation «ChargeUp» in der Übersicht:

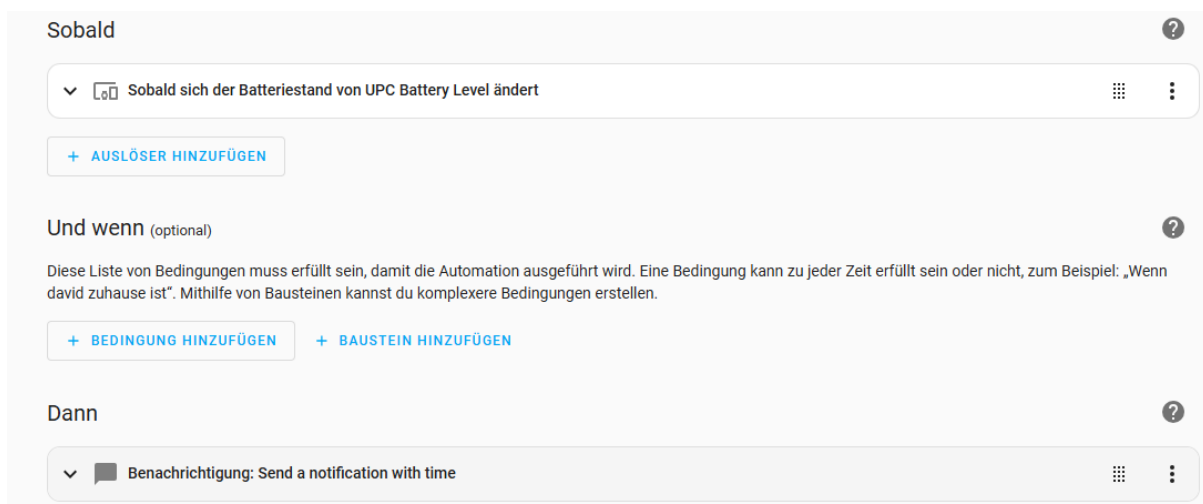


Bild 16: Screenshot, 11.04.2025

Es kann vielleicht verwirrend sein, dass dort UPC als Gerät steht, aber das ist der Gerätename meines iPhone.

Zusammengefasst erwies sich auch der Batteriewechsel inklusive ChargeUp und StopCharge als voller Erfolg. Obwohl mit den vorherigen Automationen die Grundlagen für den Batteriewechsel schon gelegt waren, war es umso spannender zu schauen, wo die Grenzen der Automationen liegen respektive ab wann sie nicht mehr ganz so verlässlich sind.

6 Die Grenzen vom Home Assistant

Home Assistant ist eine der leistungsfähigsten Plattformen zur Steuerung eines Smart Homes. Mit seiner Open-Source-Philosophie, der enormen Anpassbarkeit und der breiten Geräteunterstützung bietet es viele Vorteile gegenüber kommerziellen Lösungen. Doch trotz dieser Stärken gibt es auch einige Herausforderungen und Einschränkungen, die Nutzer beachten sollten. Wer sich für Home Assistant entscheidet, muss sich bewusst sein, dass nicht alles von Beginn weg reibungslos funktioniert und der Betrieb sowie teilweise auch die Installation mit einem gewissen Aufwand verbunden ist.

Eine der grössten Herausforderungen bei Home Assistant ist die Kompatibilität. Zwar unterstützt die Plattform eine Vielzahl von Smart-Home-Geräten und Protokollen wie ZigBee, Z-Wave, MQTT und viele weitere, doch nicht jedes Gerät wird vollständig oder problemlos erkannt. Besonders proprietäre Systeme, die sich stark an ihre eigene Cloud-Infrastruktur binden, lassen sich oft nur eingeschränkt oder mit Umwegen integrieren. Manche Funktionen können nur mit zusätzlichen Plugins oder durch Umprogrammierung genutzt werden. Dies führt dazu, dass Nutzer häufig manuell nach Lösungen suchen oder sogar eigene Skripte schreiben müssen, um alle gewünschten Geräte zu verbinden.

Ein weiteres Hindernis stellt der Einrichtungsaufwand dar. Im Gegensatz zu kommerziellen Smart-Home-Systemen, die meist über einfache Apps gesteuert werden, erfordert Home Assistant ein gewisses Mass an technischem Wissen. Während die Grundinstallation grossenteils relativ einfach ist, steigt der Schwierigkeitsgrad schnell an, wenn erweiterte Funktionen gewünscht sind. Das Anlegen komplexer Automationen, das Verknüpfen verschiedener Dienste oder das Anpassen der Benutzeroberfläche setzt oft YAML-Kenntnisse oder sogar Programmiererfahrung voraus. Für technikaffine Nutzer mag dies eine spannende Herausforderung sein, doch Einsteiger können sich schnell überfordert fühlen.

Auch die Wartung und regelmässige Updates bringen gewisse Schwierigkeiten mit sich. Da Home Assistant ständig weiterentwickelt wird, erscheinen häufig neue Versionen mit Verbesserungen und neuen Funktionen. Allerdings können diese Updates unerwartete Probleme verursachen, wenn sich beispielsweise Schnittstellen

ändern oder bestimmte Integrationen nicht mehr funktionieren. Nutzer müssen regelmässig überprüfen, ob alle Komponenten nach einem Update noch kompatibel sind und gegebenenfalls Anpassungen vornehmen. Zudem sind Backups essenziell, um im Falle eines fehlerhaften Updates das System schnell wiederherstellen zu können.

Neben diesen technischen Aspekten ist auch die Benutzerfreundlichkeit ein Punkt, der oft kritisiert wird. Während Home Assistant nahezu unbegrenzte Möglichkeiten bietet, ist die Lernkurve für Einsteiger steil. Die Benutzeroberfläche ist zwar anpassbar, wirkt aber im Vergleich zu kommerziellen Lösungen oft weniger intuitiv. Wer einfache Plug-and-Play-Lösungen gewohnt ist, könnte sich schwer damit tun, sich in die Struktur und die Funktionsweise von Home Assistant einzuarbeiten. Zwar gibt es eine grosse Community und zahlreiche Anleitungen, doch diese setzen oft ein gewisses Grundverständnis für Technik voraus.

Ein nicht zu unterschätzender Faktor sind die Leistungsanforderungen. Je nach Umfang des Smart-Home-Systems kann Home Assistant eine leistungsfähige Hardware erfordern. Zwar lässt sich die Software auf einem Raspberry Pi installieren, doch mit steigender Anzahl von Automationen, Sensoren und Geräten kann die Performance schnell an ihre Grenzen stossen. Viele Nutzer setzen daher auf leistungsfähigere Systeme wie einen dedizierten Server oder einen Mini-PC, was wiederum zusätzliche Kosten und Konfigurationsaufwand mit sich bringt.

Schliesslich spielen auch Datenschutz und Sicherheit eine grosse Rolle. Da Home Assistant lokal betrieben werden kann, sind Nutzer nicht auf externe Cloud-Dienste angewiesen – ein grosser Vorteil in puncto Privatsphäre. Allerdings bedeutet dies auch, dass die Verantwortung für Sicherheitseinstellungen und Backups vollständig beim Nutzer liegt. Es gibt keine zentrale Instanz, die sich um Updates oder Sicherheitslücken kümmert, sodass jeder selbst Massnahmen wie Firewalls, regelmässige Backups und sichere Zugangsdaten einrichten muss. Wer Home Assistant mit externen Diensten verbindet, sollte sich zudem bewusst sein, welche Daten möglicherweise nach aussen übertragen werden.

Trotz dieser Herausforderungen bleibt Home Assistant eine der flexibelsten Smart-Home-Lösungen auf dem Markt. Wer bereit ist, sich mit der Technik auseinanderzusetzen und die notwendigen Vorkehrungen zu treffen, kann ein

leistungsfähiges und individuell angepasstes System aufbauen. Allerdings sollten sich Nutzer der möglichen Einschränkungen bewusst sein, um Frustration und unerwartete Probleme zu vermeiden. Wer eine einfache, wartungsarme Lösung sucht, könnte mit einem kommerziellen Smart-Home-System möglicherweise besser bedient sein. Doch für alle, die maximale Kontrolle über ihr Smart Home haben möchten, bleibt Home Assistant eine der besten Optionen – auch wenn der Weg dorthin nicht immer ganz einfach ist.

7 OpenSource – Fluch oder Segen?

Die Open-Source-Bewegung hat die Technologiewelt revolutioniert. Statt proprietärer Software, die von Unternehmen entwickelt und kontrolliert wird, stehen Open-Source-Projekte für Offenheit, Transparenz und Zusammenarbeit. Doch ist Open Source ein uneingeschränkter Segen oder gibt es auch Schattenseiten? Dieser Frage wird im folgenden Kapitel nachgegangen.

7.1 Vorteile von Open Source

Ein zentraler Vorteil von Open-Source-Software ist die Transparenz. Der Quellcode ist frei einsehbar und kann von jedem überprüft werden. Dadurch lassen sich Sicherheitslücken schneller entdecken und beheben, da eine grosse Community von Entwicklern daran arbeitet. Dies ist besonders im Bereich Smart Home, beispielsweise bei Home Assistant, ein bedeutender Faktor. Nutzer können den Code anpassen, erweitern und sich darauf verlassen, dass keine versteckten Hintertüren existieren.

Ein weiterer Vorteil ist die Kostenersparnis. Da Open-Source-Software kostenlos verfügbar ist, profitieren sowohl private Nutzer als auch Unternehmen von kostengünstigen Lösungen. Darüber hinaus fördert Open Source Innovation, da Entwickler weltweit ihre Ideen einbringen können.

7.2 Herausforderungen und Nachteile

Trotz dieser Vorteile gibt es auch Herausforderungen. Open-Source-Projekte sind oft auf freiwillige Entwickler angewiesen, was zu unregelmässiger Wartung oder fehlender langfristiger Unterstützung führen kann. Kommerzielle Software bietet in der Regel garantierten Support, während Open-Source-Projekte stark von der aktiven Community abhängen.

Zudem besteht die Gefahr, dass Open-Source-Software für Cyberangriffe anfälliger ist, wenn sie nicht regelmässig aktualisiert wird. Angreifer können sich den offenen Quellcode zunutze machen, um Schwachstellen gezielt auszunutzen.

7.3 Fazit

Vorteile	Nachteile
<p>Transparenz – Der Quellcode ist frei einsehbar, wodurch Sicherheitslücken schneller entdeckt und behoben werden können.</p>	<p>Unregelmäßige Wartung – Open-Source-Projekte sind oft auf freiwillige Entwickler angewiesen, was zu mangelnder langfristiger Unterstützung führen kann.</p>
<p>Kostenersparnis – Open-Source-Software ist meist kostenlos, was sowohl für private Nutzer als auch für Unternehmen finanzielle Vorteile bringt.</p>	<p>Abhängigkeit von der Community – Es gibt keinen garantierten Support, sodass Nutzer auf freiwillige Hilfe der Community angewiesen sind.</p>
<p>Flexibilität & Anpassbarkeit – Nutzer können den Code modifizieren und an ihre individuellen Bedürfnisse anpassen.</p>	<p>Sicherheitsrisiken – Wenn Open-Source-Software nicht regelmäßig aktualisiert wird, können Angreifer bekannte Schwachstellen gezielt ausnutzen.</p>
<p>Innovationsförderung – Entwickler aus aller Welt können zur Weiterentwicklung beitragen und neue Ideen einbringen.</p>	<p>Einarbeitungsaufwand – Die Nutzung und Anpassung von Open-Source-Software erfordert oft technisches Wissen und kann für Einsteiger schwierig sein.</p>

8 Reflexion

Die folgende Reflexion beschäftigt sich mit meiner eigenen Durchführung einer individuellen praktischen Arbeit rund um das Überthema Home Assistant. Die Lernerfahrungen und der Lernprozess stehen dabei im Vordergrund. Des Weiteren werden auch Themen wie Erfüllen der Zielsetzung, Einhalten des Zeitplans und Verbesserungspotential kritisch hinterfragt und reflektiert.

Zielsetzung und Fragestellung

Wenn ich zurückschaue auf die Ziele, die ich mir zu Beginn dieser Arbeit gestellt habe, bin ich der Meinung, dass ich diese zu grossen Teilen erfolgreich erreicht habe.

Im Verlauf der letzten zwei Lehrjahre hat mir der Projektunterricht in der Berufsschule aufgezeigt, wie wichtig die Planung eines solchen grossen Projektes sein kann. Die exakte Planung meiner IPA war mir dem entsprechend auch von hoher Bedeutung. Diese dann noch im Gleichtakt mit dem Zeitplan zu haben ist mir ebenfalls sehr gut gelungen. Es gelang mir, Produkte und Arbeiten zeitgemäss abzugeben.

Ein weiteres Teilziel war es, für mich persönlich die Funktionalität eines Home Assistant in einem durchschnittlichen Haushalt zu erforschen. Dabei habe ich während der Arbeit immer wieder daran gedacht, wie etwas bei mir zuhause funktionieren würde. Diese Erforschung hat viel zeitlicher Aufwand beansprucht. Deshalb möchte ich gerne einen eigenen Abschnitt widmen.

Erfahrungen mit Home Assistant

Vom SmartHome Kritiker zum SmartHome Fanatiker. Das wäre vielleicht der passendste Titel gewesen für meine IPA. Denn vor meiner Arbeit hatte ich weder beruflich, geschweige denn privat viel Berührungspunkte mit dem Thema Home Assistant. Ich stand dem Thema teilweise gar kritisch gegenüber. Als dann Ende 2024 die Themenfindung zur IPA angefangen hat, wurde bei uns im Betrieb eine kleine Home Assistant-Welle ausgelöst. Angefangen hat mein Interesse mit Herrn Dürrenberger, welcher schon früh ein Spezialist auf dem Thema war und mir immer wieder aufzeigte, was mit heutiger Technologie und eben dem Home Assistant eigentlich alles möglich ist. Das hat mich bis zu meiner IPA begleitet. Ich wollte unbedingt mehr dazu herausfinden.

Meine anfängliche Skepsis ist schnell verschwunden. Rückblickend würde ich die Anfänge meiner Arbeit wie eine Art «Augenöffnen» beschreiben, ein Eintauchen in eine Welt, die ich bis dahin noch nicht kannte. Als ich dann einmal in der Thematik drin war, fühlte ich mich sofort gefesselt. Für mich war es auch schön und eindrücklich zu sehen, wie aufnahme- und begeisterungsfähig ich sein kann, wenn mich etwas brennend interessiert. Infolgedessen waren auch meine Lernerfolge schnell bemerkbar. Bis heute bin ich noch immer von Thema gepackt und überlege immer wieder neue Möglichkeiten, mein SmartHome zu erweitern.

Was hätte besser laufen können?

Auch wenn mir meine Arbeit ausnahmslos grosse Freude bereitet hat, möchte ich trotzdem zwei Punkte erwähnen, bei denen ich Verbesserungspotential in meinem Arbeits- und Lernprozess sehe.

- Konstanz

Während der Bearbeitung meiner Arbeit wurde meine verfügbare Arbeitszeit in der Regel auf ein bis zwei Tage pro Woche aufgeteilt. Diese Struktur hatte zunächst den Vorteil, dass ich regelmässig an dem Thema arbeiten konnte. Rückblickend betrachtet würde ich es jedoch bevorzugen, die zur Verfügung stehende Zeit gebündelt an aufeinanderfolgenden Tagen oder sogar am Stück zu nutzen. Der Grund dafür ist, dass es mitunter sehr mühsam war, sich nach mehreren Tagen Pause immer wieder neu in die Thematik hineinzufinden und den gedanklichen Anschluss nicht zu verlieren. Eine zusammenhängende Arbeitsweise hätte meiner Meinung nach einen flüssigeren Arbeitsprozess ermöglicht.

- Ablenkung

Ablenkung klingt zunächst nach mangelnder Selbstbeherrschung, was es in diesem Fall aber nicht ist. Vielmehr geht es darum, dass ich rückblickend nicht immer glücklich darüber war, wie manche Mitarbeitende meine geplante Arbeitszeit nicht vollständig respektiert haben. Wichtig ist mir hierbei zu betonen, dass dies kein Vorwurf an die Kolleginnen und Kollegen sein soll – vielmehr sehe ich das als einen Punkt bei mir selbst. Ich hätte deutlicher und klarer kommunizieren sollen, wie wichtig mir diese Zeiten für konzentriertes Arbeiten an meiner Aufgabe waren und dass ich sie wirklich für mich benötige.

9 Anhang

Quellenverzeichnis

Abschnitt	Link / Buch	Datum
2.5 HACS	https://www.hacs.xyz/	
3.1 Sonoff Display NSPanel / Pro	https://www.youtube.com/watch?v=p-AK4o5jOSI&t=528s	21.03.2025
4.1 Katzenmelder	https://developer.lametric.com/icons	11.04.2025
4.1 Katzenmelder	https://www.home-assistant.io/integrations/lametric/	11.04.2025
5 Die Grenzen von Home Assistant	https://de.wikipedia.org/wiki/Home_Assistant	14.03.2025
5 Die Grenzen von Home Assistant	https://www.reddit.com/r/homeassistant/comments	14.03.2025
6 OpenSource – Fluch oder Segen?	https://osb-alliance.de/featured/kommerzielle-open-source-software-segen-oder-fluch	14.03.2025

Bildverzeichnis

Abschnitt	Link / Buch	Datum
Titelbild	https://www.home-assistant.io/green/	1.3.2025
Bild 1	Eigene Aufnahme, USB Jumper	21.03.2025
Bild 2	Screenshot auf Home Assistant Interface	11.04.2025
Bild 3	Screenshot auf eWeLink App	11.04.2025
Bild 4	Eigene Aufnahme, NSPanel	15.04.2025
Bild 5 & 6	Screenshot auf Home Assistant Interface	11.04.2025
Bild 7	https://www.solarmarkt.ch/de/lametric-smart-display-lametric-time	31.03.2025
Bild 8	Screenshot auf Home Assistant Interface	11.04.2025
Bild 9	Eigene Zeichnung	17.04.2025
Bild 10-16	Screenshot auf Home Assistant Interface	11.04.2025

Arbeitsprotokoll

Datum	Zeit	ausgeführte Arbeit	Bemerkungen	erreichte Teilziele
03.03.	2h	Rackaufbau und Netzwerk konfiguration		
03.03.	2h	Ersteinrichtung von allen Gerätschaften		
03.03.	3h	Verschiedene Verküpfungen im HomeAsisstant und Test mit ersten Automationen		
03.03.	30min	Diverse Versuche mit den Bewegungsmelder		
03.03.	30min	Erstellen von Dokumenten und weiteren Formalitäten	hinterlegt bei DK	
				Rackbau abgeschlossen, stabiles Subnetz aufgebaut
07.03.	30min	Entwurf der Einleitung für die Dokumentation		
07.03.	30min	Soft- sowie Firmwareupdates auf diversen Geräten		
07.03.	3h	Verknüpfung Sonoff-HA und Google Nest-HA genauer angeschaut und verschiedene Tests durchgeführt		
07.03.	10min	Erstellen und abspeichern von Backups		
07.03.	1h	Layouten vom Dashboard und Anpassung an Google Nest Hub		
07.03.	3h	Aufbau Dokumentation und erstellen von Quellen und Bildverzeichnis		
12.03.	10min	Bestellung von USB-ttl Adapter für flashen von Sonoff Display	USB Adapter kommt anfangs KW 12 an	
14.03.	1h	Verbesserung Google Nest Integration druch Google SGK		
14.03.	1h	Diverse Applikationen auf LaMetric getestet		
14.03.	3h	Automation "Wettercheck", "Batterie wechseln" und "Katzenmelder" eingerichtet und getestet		
14.03.	3h	Dokumentation diverser Einrichtungen und Automationen		
14.03.	10min	Nachtragen Arbeitsjournal		

Alle wichtigen Automationen und Verknüpfungen sind gemacht

			funktioniert nicht! Fehlende Komponenten dokumentiert
21.03.	3h	Versuche vom flashen und konfigurieren des NSPanels	
21.03.	1h	De und Reinstallation von Rack um mobilität zu beschreiben	
21.03.	4h	Dokumentation von Flashversuchen und weiteren Paneleinstellungen	
			Erste Grenzen von Geräten und HA wurden erkannt
26.03.	4h	neues NSPanel Pro Ersteinrichtung und gleich auch aufspitzen von Software um Verbindung zu HA zu gewährleisten	
26.03.	3h	Spotify integration und anpassung von allen Dashboards	
26.03.	2h	Dokumentation	
31.03.	3h	Dokumentation	
31.03.	30 min	Racktransport zurück nach Thalwil	
31.03.	2h	Leztzte Veränderungen an den Dashboard	
31.03.	2h	Backups herunterladen und sichern	
04.04.	3h	Dokumentation	
04.04.	1h	Überprüfen der Funktionalität von allen Automationen	
11.04.	5h	Dokumentation	
14.04.	6h	Dokumentation	
14.04.	1h	Quellen und Bildverzeichnis machen	
14.04.	1h	wiederholtes kontrolliern der Funktionen	
15.04.	1h	Zwischengespräch	
15.04.	2h	Reflexion schreiben	
15.04.	6h	Dokumentation / Überarbeitung	