



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**MODUL CHYTRÉHO OVLÁDÁNÍ ŽALUZIÍ V NÁSTROJI
HOME ASSISTANT**

SMART COVER CONTROL MODULE FOR HOME ASSISTANT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MATÚŠ JUŠTIK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN PLUSKAL, Ph.D.

BRNO 2024

Zadání bakalářské práce



153141

Ústav: Ústav informačních systémů (UIFS)
Student: **Juštik Matúš**
Program: Informační technologie
Název: **Modul chytrého ovládání žaluzií v nástroji Home Assistant**
Kategorie: Vestavěné systémy
Akademický rok: 2023/24

Zadání:

1. Nastudujte možnosti ovládání žaluzií v platformě pro domácí automatizace Home Assistant.
2. Navrhněte modul, který bude schopen ovládat žaluzie pouze pomocí příkazů chod nahoru, chod dolů a zastavení. Modul bude schopen provést minimálně automatizace: stínění interiéru podle polohy slunce s maximalizací prostupu denního světla, uzavření žaluzií na základě události jako je poloha slunce, denní doba, zapnutí vnitřních světel v kombinaci s polohou slunce, ochrana žaluzií při dešti či větru, případně další automatizace dle domluvy s vedoucím.
3. Implementujte navržený modul z bodu 2. Konfigurace modulu musí být přívětivá pro uživatele pomocí funkcionality přidání integrace.
4. Otestujte Vámi navržené řešení na laboratorním či domácím nasazení nástroje Home Assistant. Diskutujte dosažené výsledky, stabilitu a spolehlivost Vašeho řešení.
5. Zveřejněte Vaši implementaci vhodnou formou, aby byla dostupná pro uživatele. Promuňte Vaši práci vhodnými kanály a vyhodnoťte zpětnou vazbu od uživatelů.

Literatura:

- ESPHome. *ESPHome* [online] [viz. 2023-09-27]. Dostupné z: <https://esphome.io/index.html>
- Tasmota. *Tasmota Documentation - Tasmota* [online] [viz. 2023-09-27]. Dostupné z: <https://tasmota.github.io/docs/>
- Home Assistant. *Home Assistant* [online] [viz. 2023-09-27]. Dostupné z: <https://www.home-assistant.io/>

Při obhajobě semestrální části projektu je požadováno:
Body 1 a 2

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Pluskal Jan, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 5.5.2024

Abstrakt

Cielom tejto práce je štúdium a implementácia ovládania vonkajších žalúzií pre platformu domácej automatizácie Home Assistant. Hlavným cieľom je navrhnuť a implementovať modul schopný ovládať žalúzie, ktoré nedisponujú spätnou väzbou, iba pomocou príkazov nahor, nadol a stop. Navrhnutý modul musí minimálne spĺňať automatizácie, ako je napríklad zatienenie a odtienenie interiéru podľa polohy slnka s maximálnym využitím denného svetla, adekvátne reagovanie žalúzií na ochranu pred nepriaznivým počasím a ďalšie možné automatizácie. Modul bude následne publikovaný na vhodné kanály, aby sa uplatnil v komunite používateľov Home Assistant. Táto práca prinesie používateľom lepšie a efektívnejšie riešenie pre ovládanie žalúzií v rámci Home Assistant.

Abstract

The aim of this work is the study and implementation of external blinds control for the Home Assistant of home automation platform. The main goal is to design and implement a module capable of controlling blinds that do not have feedback, using only up, down and stop commands. The designed module must at least comply with automation, such as shading and illumination of the interior according to the position of the sun with maximum use of the day light, adequate response of blinds to protect against possible weather conditions and other automations. The module will then be published to the appropriate channels to reach the Home Assistant users' community. This work brings a better and more efficient solution for controlling blinds within the Home Assistant.

Kľúčové slová

Domáce automatizácie, IoT, Ovládanie roliet, ESP Home, Home Assistant, Integrácia, Príspevok komunity

Keywords

Home automation, IoT, Blind control, ESP Home, Home Assistant, Integration, Contribution to the community

Citácia

JUŠTIK, Matúš. *Modul chytrého ovládání žaluzií v nástroji Home Assistant*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal, Ph.D.

Modul chytrého ovládání žaluzií v nástroji Home Assistant

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Jana Pluskala, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Matúš Juštík
7. mája 2024

Podakovanie

Chcel by som vyjadriť svoju vďaku Ing. Janovi Pluskalovi, Ph.D., za jeho vedenie a odbornú podporu počas vypracovania tejto práce.

Obsah

1	Úvod	4
2	Prieskum riešení a technológií používaných pre Smart Home	6
2.1	Home Assistant	6
2.1.1	Platforma Home Assistant	6
2.1.2	Architektúra a komponenty Home Assistantu	8
2.1.3	Komunitný obchod Home Assistant	9
2.2	Technológie používané v Home Assistant	9
2.2.1	ESPHome	10
2.2.2	Yet Another Markup Language (YAML)	10
2.2.3	Native API TCP	11
2.2.4	Python	11
2.2.5	Wi-Fi	11
2.3	Dostupné možnosti ovládania žalúzií	12
2.3.1	Bluetooth ovládače	12
2.3.2	Ovládače s infračervenou LED	12
2.3.3	Rádiové frekvenčné ovládače	12
2.3.4	Fyzické nástenné vypínače	13
2.3.5	Inteligentné vypínače	13
2.4	Použitý Hardware	13
2.4.1	Sonoff T2EU3C nástenný vypínač	14
2.4.2	Mikrokontrolér ESP8266	14
3	Návrh	15
3.1	Konkurenčné riešenia	15
3.2	Funkčné a nefunkčné požiadavky	15
3.3	Príkazy ovládania	16
3.3.1	Základná logika	16
3.4	Automatizácie	17
3.4.1	Automatizácia na základe času	18
3.4.2	Prepojenie iných vypínačov na svetlo	18
3.4.3	Ochrana žalúzií pri daždi a vetre	18
3.5	Inštalčný návrh	18
3.5.1	Prvotná konfigurácia a úprava konfigurácie	19
3.6	Používateľské rozhranie a používanie	19
3.7	Hlavné problémy	21
3.7.1	Výpočet času	21
3.7.2	Synchronizácia s ovládaním mimo našej integrácie	22

3.7.3	Odozva	22
3.7.4	Konfigurácia a jej úprava	22
3.8	Distribúcia	22
4	Implementácia	23
4.1	Spracovanie údajov pri konfigurácii	23
4.1.1	Vizuál formuláru	24
4.1.2	Prepojenie údajov a ich zobrazenie	25
4.2	Pridanie ovládacích prvkov	25
4.2.1	Vizuál ovládania	27
4.3	Logika programu	27
4.3.1	Výpočet pozície	28
4.3.2	Riešenie problému synchronizácie ovládania	28
4.4	Automatizácie	28
4.5	Rozšírenia	31
5	Testovanie	33
5.1	Testovacie prostredie	33
5.1.1	Výsledky testovania	34
5.2	Šírenie v komunite používateľov	35
5.2.1	Pridanie do komunitného obchodu	35
5.2.2	Spätná väzba	35
6	Záver	37
	Literatúra	38
A	Plagát	40
B	Príspevky na platformách	42
C	Používanie riešenia	44
D	Obsah priloženého pamäťového média	48

Zoznam obrázkov

2.1	Na obrázku sú zobrazené možnosti inštalácií, a to, čo zahŕňa ktorá možnosť. Prevzaté z [7].	7
2.2	Architektúra Home Assistant jadra. Prevzaté z [6].	8
2.3	Architektúra Home Assistant Supervisora. Prevzaté z [6].	9
2.4	Schéma možnosti zapojenia Sonoff T2EU3C.	13
2.5	ESP8266 v konfigurácii ESP12 (ukážka pre predstavu). Prevzaté z [13]. . .	14
3.1	Stavový diagram návrhu žalúzií.	17
3.2	Prvotný návrh používateľského rozhrania.	19
3.3	Návrh s možnosťou tienenia po naštudovaní možností Home Assistant. . . .	20
3.4	Návrh bez možnosti tienenia po naštudovaní možností Home Assistant. . . .	20
4.1	Formulár pri prvotnej konfigurácii.	24
4.2	Zobrazované údaje po kliknutí možnosti pod ovládacími prvkami.	25
4.3	Vizuál ovládania s ovládaním tienenia.	26
4.4	Vizuál ovládania bez ovládania tienenia.	26
4.5	Ovládanie z hlavného panelu.	27
4.6	Vývojový diagram automatizácie na zatvorenie žalúzie pri určenom čase. . . .	29
4.7	Vývojový diagram automatizácie na otvorenie žalúzie pri východe slnka. . . .	30
4.8	Vývojový diagram automatizácie na zatvorenie žalúzie pri zapnutí svetla v noci. .	31
5.1	Aktivita príspevku na komunitnom fóre Home Assitantu k dňu 7. mája 2024. . . .	35
5.2	Požiadavka na pull, kritéria na schválenie.	36
5.3	Kladná spätná väzba od používateľa.	36
A.1	Plagát z konferencie Excel@FIT 2024.	41
B.1	Príspevok na platforme komunitného fóra.	42
B.2	Príspevok na platforme Reddit.	43
C.1	Bez príkazu, čaká sa na príkaz.	44
C.2	Príkaz smeruj dole, ovláda sonoff T3 L2.	44
C.3	Posuvné tlačidlá, čaká na príkaz.	45
C.4	Posuvné tlačidlá, zatvarajú na určitú pozíciu alebo do prerušenia, ovláda sonoff T3 L2.	45
C.5	Posuvné tlačidlá, konečná pozícia, čaká na príkaz.	46
C.6	Posuvné tlačidlá, otvárajú na určitú pozíciu alebo do prerušenia, ovláda sonoff T3 L1.	46
C.7	Tlačidlá ovládania, otvárajú do prerušenia, ovláda sonoff T3 L1.	47

Kapitola 1

Úvod

Inteligentné zariadenia (IoT zariadenia) v našich domácnostiach majú stále väčší význam. Môže ísť o rôzne senzory, doplnky, ale aj vylepšenie iného spotrebiča. Tieto zariadenia často majú komerčné riešenia od výrobcov, s príchodom ďalších a ďalších riešení používateľom začalo byť nepraktické, mať na každé zariadenie iné riešenie alebo nutnosť používať iný software. Preto vznikol Home Assistant 2.1, ktorého komunita sa snaží vytvárať riešenia pre tieto zariadenia a držať ich na jednom mieste. Cieľ tejto práce predstavuje vytvorenie open-source riešenia pre ovládanie externých žalúzií. Toto riešenie by malo byť pre používateľa ľahko implementovateľné a následne aj ľahko ovládateľné.

IoT zariadenia zažívajú zlatú éru, hovorí aj skutočnosť, že v roku 2022 bolo celosvetovo pripojených 14,3 bilióna zariadení. A očakáva sa, že v roku 2023 bol až 16 % nárast, čo by v celosvetovej mierke mohlo znamenať až 16,7 bilióna zariadení. Tento nárast sa v blízkej dobe spomaľovať ani nebude [8]. Existuje množstvo firiem, ktoré si už nechali viaceré automatizácie vyhotoviť na zákazku alebo o tom uvažujú. A rozmanitosť využití týchto zariadení bude len podporovať vytváranie rôznych používateľských riešení pre domáce použitie, ale aj firemné použitie.

Hlavnou motiváciou tejto práce je súčasná absencia ideálneho riešenia v rámci Home Assistantu. Existujúce riešenia 3.1 obsahujú menej schopné výpočty polohy alebo neponúkajú možnosti ovládania tienenia. Taktiež používajú zastaralý spôsob konfigurácie prostredníctvom konfiguračných súborov a nezahŕňajú zabudované automatizácie, čo môže byť obzvlášť nepríjemné pre používateľov Home Assistantu, ktorí majú stovky automatizácií, o ktoré sa musia starať. Naším cieľom je odstrániť tieto nedostatky a postaviť riešenie odznovu tak, aby bolo lepšie a použiteľnejšie ako súčasné možnosti, a zároveň získať pre naše riešenie pozornosť používateľov.

V kapitole 2 tejto práce sa čitateľ zoznámí so systémom Home Assistant. Okrem základného fungovania Home Assistant sa dozvie aj o jeho jednotlivých častiach a ich účele. Spomenú sa aj dôležité doplnky ako HACS spomenuté v sekcii 2.1.3 a ESPHome spomenuté v sekcii 2.2.1, ktoré rozširujú funkcionality Home Assistantu. Ďalej, v tejto kapitole sa budeme venovať aj technológiám využívaným pri riešení problémov domácej automatizácie.

Následne sa po teoretickom úvode a uvedení do problematiky presunieme do kapitoly 3, kde bude prezentovaný návrh nášho vlastného riešenia a jeho implementácia. Táto kapitola sa bude tiež zaoberať predpokladanými problémami a spôsobmi, ako ich plánujeme riešiť.

V kapitole 4 sa detailne pozrieme na konkrétne riešenie, ktoré sme podľa návrhu popísanom v predchádzajúcej kapitole implementovali. Budeme skúmať fungovanie našej integrácie, použité knižnice a technológie a ako sme sa vyrovnali s problémami, ktoré sme počas

implementácie zistili. Okrem toho, sa v tejto kapitole zhodnotia aj pridané rozšírenia do nášho riešenia.

Časť 5 sa bude venovať procesu testovania implementovaného riešenia. Okrem samotného testovania sa pozrieme aj na spätnú väzbu od užívateľov a spôsoby, ako sme zdieľali naše riešenie s komunitou. V tejto kapitole sa tiež zhodnotia dosiahnuté výsledky testovania.

V kapitole 6 následne nájdeme zhodnotenie dosiahnutých výsledkov a popis ďalších možných postupov v rámci bakalárskej práce. Tento záver bude obsahovať aj návrhy na ďalšie kroky, ktoré by mohli nasledovať po ukončení tejto práce.

Kapitola 2

Prieskum riešení a technológií používaných pre Smart Home

Táto kapitola sa zameriava na systém Home Assistant a vysvetľuje, prečo bol zvolený ako hlavný nástroj v tejto práci. Okrem toho bude hovoriť o jeho zložení a architektúre, pričom sa venuje jednotlivým častiam tohto systému. Ďalej sa pozrieme na komunitný obchod pre tento systém, jeho účel a proces pridávania nových integrácií. Bude tiež vysvetlené fungovanie integrácií a ich význam v rámci systému Home Assistant. Hoci sa zatiaľ nebudeme venovať procesu tvorby integrácií, táto téma sa preberie v nasledujúcich kapitolách. Ďalším bodom bude ESPhome, ktorý umožňuje pripájanie Home Assistant k rôznym zariadeniam s ESP mikročipmi. Nakoniec sa pozrieme na rôzne technológie využívané v domácej automatizácii a spôsoby, ako je možné v súčasnosti ovládať vonkajšie žalúzie.

2.1 Home Assistant

Popisovaná a skúmaná automatizačná platforma Home Assistant bola zvolená na použitie v tejto práci z dôvodu svojej popularity, jednoduchosti nasadenia, možnosti písania integrácií v programovacom jazyku Python a možnosti nasadenia na vývojovej doske Raspberry Pi alebo virtuálnom počítači, vytvárajúc tak lokálny server. Aj keď existujú iné automatizačné systémy ako napríklad Domoticz¹ alebo openHAB², avšak Home Assistant sa po preštudovaní ostatných platforiem ukázal ako výhodná voľba, najmä vďaka vyššie spomínaným výhodám.

2.1.1 Platforma Home Assistant

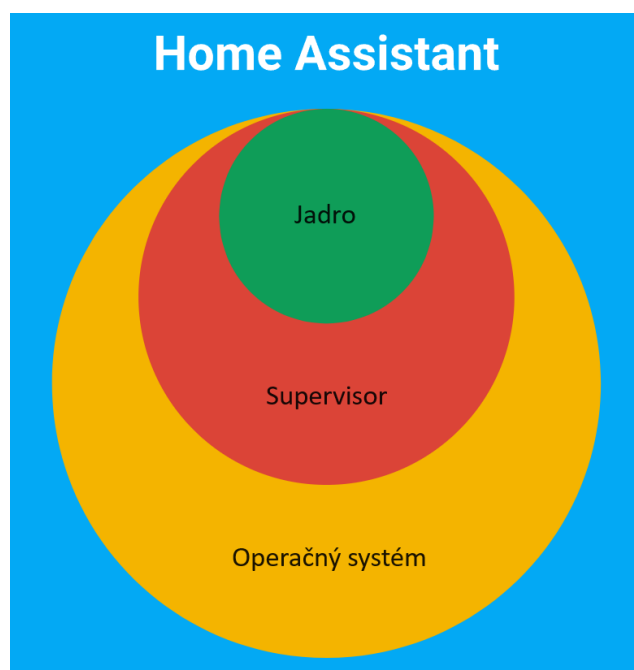
Home Assistant je software vyvinutý v roku 2013 a používaný ako domáci automatizačný server [2]. Home Assistant je vstavaný systém napísaný v jazyku Python 3 a je navrhnutý tak, aby sa jednoducho rozširoval pomocou rôznych integrácií [6]. Tieto integrácie sú vytvárané celosvetovou komunitou, ktorá sa usiluje o podporu čo najväčšieho množstva zariadení. Integrácie dokážu sledovať stav zariadení a podľa toho spustiť určité akcie, udržiavať stav zariadenia na základe nejakého vstupu alebo akcie alebo vytvárať rôzne grafy, ktoré používateľom môžu dávať rôzne užitočné informácie ako napríklad dĺžku používania alebo prepočítavať spotrebu elektrickej energie. Hlavnými prioritami Home Assistantu sú lokálne

¹Oficiálna stránka Domoticz: <https://www.domoticz.com/>

²Oficiálna stránka openHAB: <https://www.openhab.org/>

ovládanie a zabezpečenie súkromia [6]. Home Assistant je open source systém pre domáce automatizácie. Jeho inštalácia je jednoduchá, kde stačí stiahnuť obraz pre konkrétne zariadenie alebo systém, ktorý chceme používať a následne postupovať podľa jednoduchých krokov uvedených na oficiálnych stránkach Home Assistantu³.

Home Assistant používateľom ponúka viacero metód inštalácie vid' obrázok 2.1, pričom najväčšiu podporu a najviac funkcií obsahuje Home Assistant Operačný Systém (HAOS). Funkcie, ktoré nemusia byť pri ostatných metódach inštalácie dostupné je zálohovanie, obnovenie záloh, doplnky ako aj integrácie [7]. Medzi iné metódy inštalácie patrí Home Assistant Container alebo pre pokročilých používateľov tu je Home Assistant Supervised alebo Home Assistant Core [2]. Operačný systém je možné nainštalovať ako virtuálny počítač na systémoch Windows, Linux alebo aj MacOS. V prípade virtuálneho prostredia na počítači, umožňuje Home Assistant plne využiť potenciál starších, ale aj novších počítačov, pričom poskytuje flexibilitu pri voľbe hardvéru a operačného systému. Na tieto systémy taktiež je možné zvoliť rôzne metódy inštalácie. Pre Windows to je Home Assistant Operating System ako virtuálny stroj alebo Home Assistant Core, ktorý pobeží na WSL. Pre MacOS to je Home Assistant Operating System ako virtuálny stroj alebo inštalácia Home Assistant Core a pre Linux to sú všetky zvyšné možnosti: Home Assistant OS, Container, Supervised, alebo Core [7].



Obr. 2.1: Na obrázku sú zobrazené možnosti inštalácií, a to, čo zahŕňa ktorá možnosť. Prevzaté z [7].

Jedna z jednoduchších platforiem na inštaláciu — Raspberry Pi, ponúka stabilnú a optimalizovanú inštaláciu pre tieto konkrétne zariadenia [7]. Raspberry Pi je taktiež jedno z viac populárnych zariadení, ktoré sa používajú [2]. Inštaláciu je možné vykonať na zariadeniach Raspberry Pi 5, Raspberry Pi 4 a aj na staršom modeli Raspberry Pi 3 Model B, no nie Raspberry Pi 3 Model A, ktorý nemá dostatok pamäte RAM potrebnej na chod Home Assistantu [7]. Tieto rôzne možnosti inštalácie umožňujú používateľom prispôbiť si svoje

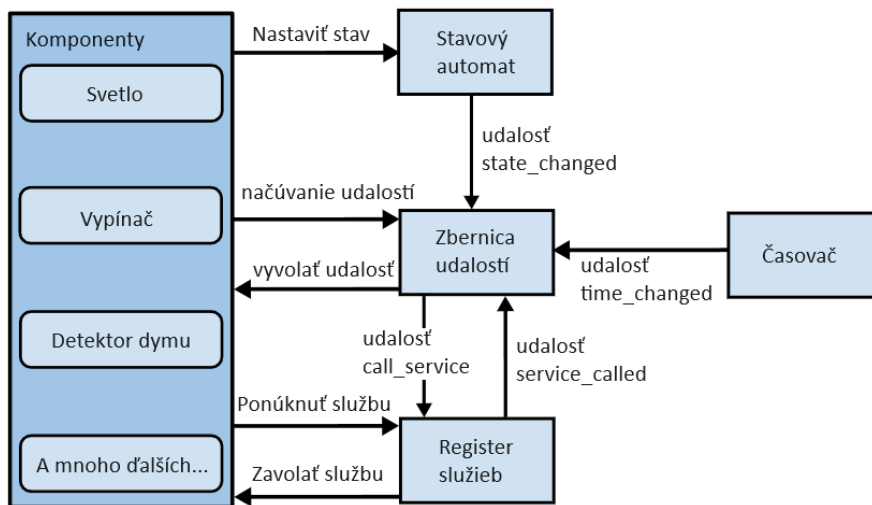
³Postup inštalácie dostupný na: <https://www.home-assistant.io/installation/>

prostredie podľa individuálnych potrieb a zabezpečujú, že Home Assistant je dostupný pre rôzne typy používateľov a zariadení.

2.1.2 Architektúra a komponenty Home Assistantu

Home Assistant sa skladá z troch hlavných častí: prvou dôležitou časťou je operačný systém, ktorý poskytuje základné prostredie pre beh Home Assistantu. Druhou časťou je tzv. supervisor, ktorý má na starosti nielen správu operačného systému, ale aj koordináciu a riadenie ďalších procesov a služieb a tretia časť predstavuje jadro celého Home Assistantu, ktoré zabezpečuje správu a integráciu jednotlivých komponentov a zariadení. Tieto tri časti úzko spolupracujú a poskytujú užívateľovi komplexné prostredie na automatizáciu a riadenie jeho domácnosti [7].

Operačný systém Home Assistant bol špeciálne navrhnutý pre použitie na jednočipových počítačoch a systémoch x86-64 [6]. Jeho hlavné vlastnosti zahŕňajú robustnosť a bezúdržbovosť počas používania [7]. Home Assistant operačný systém využíva nástroj Buildroot na vytvorenie svojej vlastnej distribúcie Linuxu. Buildroot nie je priamo samotná distribúcia, ale poskytuje základ pre jej vytvorenie a umožňuje aj krížovú kompiláciu pre rôzne architektúry, čo je užitočné najmä pre systémy na báze Arm [6]. Na vytvorenie operačného systému Home Assistant boli využité nasledujúce technológie: štandardné balíky Linuxu a GNU softvéru, Linux, GNU C knižnice, systemd init daemon, Docker container engine (ktorý sa používa pre Supervisor). Na aktualizácie sa využíva RAUC, AppArmor (zabezpečuje bezpečnosť), ako bootovací systém sa používa Barebox alebo U-Boot a pre systémové súbory sa využívajú SquashFS a ZRAM využívajúce kompresiu LZ4 [6].



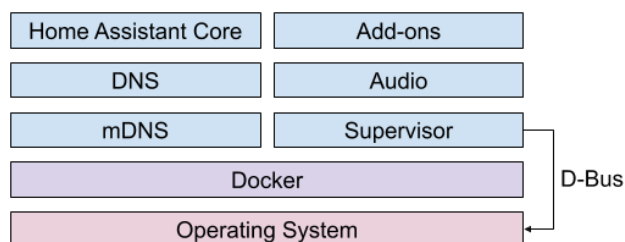
Obr. 2.2: Architektúra Home Assistant jadra. Prevzaté z [6].

Jadro Home Assistantu, rovnako ako ostatné komponenty tejto platformy, je napísané v jazyku Python a poskytuje schopnosť sledovať, ovládať a automatizovať zariadenia [4]. Jadro Home assistantu sa skladá zo štyroch hlavných častí (viď obrázok 2.2):

- **Zbernica udalostí:** Nazývané srdcom Home Assistantu. Načítava a spúšťa udalosti vznikajúce alebo nastávajúce na základe iných udalostí [6].

- **Stavový automat:** Udržiava stavy a generuje udalosť `state_changed` v prípade zmeny akéhokoľvek stavu [6].
- **Register služieb:** Počúva zbernicu udalostí a umožňuje kódu vykonať danú požiadavku, ak je vyzvaný zbernicou [6].
- **Časovač:** Posiela udalosť `time_changed` každú sekundu na zbernicu udalostí [6].

Home Assistant Supervisor má na starosti niekoľko dôležitých úloh, aby zabezpečil správne fungovanie Home Assistantu a umožnil používateľom spravovať svoju inštaláciu priamo cez Home Assistant. Jeho architektúra je zobrazená na obrázku 2.3. Medzi jeho hlavné úlohy patrí správa jadra, aktualizácie jadra ako taktiež operačného systému, správa zjednoteného audia a vytváranie záloh a ich následné obnovenie [6].



Obr. 2.3: Architektúra Home Assistant Supervisoru. Prevzaté z [6].

Home Assistant je zlúčenie jadra Home Assistantu a rôznych nástrojov a integrácií [7]. Integrácie sú často vytvárané samotnými používateľmi, aby vyhoveli ich osobným potrebám. Okrem toho Home Assistant ponúka aj zabudované integrácie, ktoré sú k dispozícii okamžite po inštalácii [6]. Každá integrácia zabezpečuje určitú časť funkcionality v rámci Home Assistantu, či už ide o spúšťanie služieb, poskytovanie konkrétnych funkcií alebo iných úloh. Integrácie sa skladajú zo základnej logiky a platforiem, ktoré sú časťami schopnými interakcie s ostatnými integráciami. Integrácie sú taktiež implementované v jazyku Python, čo umožňuje tvorbu integrácií s variabilitou a jednoduchosťou [6].

2.1.3 Komunitný obchod Home Assistant

Home Assistant Community Store ďalej len ako HACS, je v podstate doplnok pre Home Assistant, ktorý rozširuje jeho schopnosti. Po jeho nainštalovaní do Home Assistantu umožňuje sťahovanie a aktualizovanie používateľských aj iných doplnkov, skriptov, ako aj tém pre používateľské rozhranie, ktoré nie sú súčasťou základnej verzie Home Assistantu [5]. HACS nám umožní prídanie našej integrácie z platformy GitHub⁴ do Home Assistantu. Komunita HACS vytvára množstvo video návodov⁵, ako pracovať s HACS a ako využiť plný potenciál tohto nástroja v rámci Home Assistant.

2.2 Technológie používané v Home Assistant

Táto časť poskytuje prehľad o niekoľkých kľúčových prvkoch v oblasti domácej automatizácie a vývoja nášho riešenia. Začína sa popisom nástroja ESPHome, ktorý je open source

⁴Oficiálna stránka GitHub: <https://github.com/>

⁵Návody dostupné na: https://hacs.xyz/docs/community_guides

a určený na vytváranie vlastného firmvéru pre mikrokontroléry ESP8266 a ESP32. Ďalej sa zaoberá YAML formátom, ktorý je využívaný na definovanie konfigurácií a štruktúrovaných informácií v prostredí ESPHome. Native API TCP protokol je ďalším dôležitým prvkom, ktorý umožňuje spoľahlivú komunikáciu medzi ESPHome a Home Assistantom. Jazyk Python, známy svojou jednoduchosťou a univerzálnosťou, sa využíva na implementáciu rôznych funkcií a automatizácií v projektoch domácej automatizácie.

2.2.1 ESPHome

ESPHome je open source nástroj navrhnutý na vytváranie vlastného firmwaru na mikrokontroléri ESP8266 a ESP32 [3]. Poskytuje jednoduchý spôsob definovania konfigurácií pomocou YAML súborov, čo umožňuje používateľom rýchlo vytvárať firmvér bez nutnosti programovania základných funkcionalít [3]. Konfigurácia zariadenia v ESPHome sa uskutočňuje pred nahraťím firmwaru na zariadenie. To znamená, že pri akýchkoľvek úpravách je potrebné prekompilovať a znova nahráť celý firmware na zariadenie. Konfigurácia je jednoduchá aj vďaka webovému dashboardu, ktorý ESPHome ponúka. Dashboard sa používa najmä na generovanie, kompilovanie a nahrávanie firmwaru na ESP kontrolér. Ponúka taktiež textový výstup pre výpis logov z ESP kontroléru, čo vie byť užitočné pri hľadaní chýb alebo testovaní [3].

Okrem toho, ESPHome podporuje komunikáciu pomocou MQTT protokolu alebo Native API TCP protokolu. Native API TCP kóduje informácie do optimalizovaného formátu, čo znižuje režiu oproti MQTT. Pri použití Native API TCP protokolu sa systém pripája priamo na zariadenia, takže nie je nutné mať server ako v prípade MQTT. Avšak, tento protokol podporuje iba samotné ESPHome v prepojení s Home Assistant systémom [3].

Podobný nástroj ako ESPHome je Tasmota⁶. Tasmota je populárnym nástrojom pre ESP8266 a ESP32, ale jeho konfigurácia môže byť komplexnejšia a nie je tak flexibilný pri používaní v porovnaní s ESPHome [3]. Jednou z výhod ESPHome oproti Tasmote je používaný protokol. Tasmota používa pomalší MQTT, zatiaľ čo ESPHome používa Native API TCP [3]. To zohráva veľkú úlohu v tejto práci, keďže sa snažíme zachovať čo najnižšiu odozvu pri ovládaní žalúzií.

2.2.2 Yet Another Markup Language (YAML)

YAML je jazyk na serializáciu dát, vytvorený s cieľom byť použiteľný pre programovacie jazyky [2]. Jeho syntax je navrhnutý tak, aby bol intuitívny pre ľudí a umožňoval jednoduché čítanie a zápis dátových štruktúr. YAML je nezávislý na programovacím jazyku a podporuje rôzne dátové typy, ako sú zoznamy, slovníky, reťazce a ďalšie, ukázané na obrázku 2.1 [17]. Tento formát tiež umožňuje vytváranie komentárov a zahŕňa mechanizmy na zjednodušenie opakovania dát a referencovania. YAML je často využívaný na konfiguráciu softvérových aplikácií, dátových modelov a komunikačného protokolu [17].

```
set_known_position:
  description: Set the known position of the blinds
  fields:
    entity_id:
      name: Entity ID
      description: The ID of the entity
      example: 'cover.living_room_blinds'
```

⁶Odkaz na dokumentáciu nástroja Tasmota: <https://tasmota.github.io/docs/>

```
position:
  name: Entity ID
  description: The position to set
  example: 100
```

Kód 2.1: V ukážke kódu je demonštrovaný jazyk YAML, ktorý slúži na ukladanie údajov medzi reštartom Home Assistantu, aby sa zabránilo stratám dát.

2.2.3 Native API TCP

Native API TCP je optimalizovaný protokol určený na komunikáciu medzi ESPHome a Home Assistantom. Tento protokol umožňuje obom platformám efektívnu a spoľahlivú výmenu informácií [3]. Funguje nad upraveným protokolom TCP, ktorý zabezpečuje spoľahlivú a bezpečnú komunikáciu cez internet alebo lokálnu sieť [3]. TCP zaisťuje, že dáta sú prenášané spoľahlivo a v správnom poradí, čo je dôležité pre aplikácie, ako sú domáce automatizácie, kde presnosť a spoľahlivosť sú často dôležité [3]. Oproti MQTT je Native API TCP stabilnejšie, efektívnejšie, jednoduchšie na konfiguráciu, a čo je dôležité, má menšiu odozvu [3].

2.2.4 Python

Python je populárny programovací jazyk, ktorý je ľahko zrozumiteľný a jednoducho sa naučí. Poskytuje prostredie pre jednoduché, vysokoúrovňové a efektívne objektovo orientované programovanie [11]. Je silným nástrojom v oblasti automatizácií, analýzy dát, vedeckého výpočtového prostredia a vývoja webových aplikácií. Rozsiahla a aktívna komunita prispieva vytváraním modulov a knižníc, ktoré rozširujú jeho funkcionality. Interpret a štandardná knižnica jazyka sú voľne dostupné na oficiálnej stránke Pythonu, čo umožňuje každému získať binárnu alebo zdrojovú verziu Pythonu každej verzie [11]. V štandardnej knižnici Pythonu nájdeme množstvo užitočných dátových typov, ako sú celé, desatinné či komplexné čísla. No Python je dynamicky typovaný jazyk, čo znamená, že operácie medzi nekompatibilnými typmi môžu spôsobiť výnimky. Python tieto výnimky umožňuje vyvolávať a odchytať za účelom riešenia chýb [11]. Interpreter je možné rozšíriť tak, aby bolo možné používať funkcie a dátové typy, ktoré sú využívané napríklad v jazykoch C/C++ [11]. Python podporuje objektovo orientované programovanie vrátane viacnásobnej dedičnosti. Kód umožňuje importovať špecifické moduly, triedy, funkcie a iné objekty [11]. Python ako vysokoúrovňový jazyk automaticky spravuje pamäť, čo eliminuje potrebu manuálnej alokácie a uvoľňovania pamäte v kóde [11].

2.2.5 Wi-Fi

Wi-Fi je bezdrôtová sieťová technológia definovaná štandardom IEEE 802.11. Tento štandard opisuje protokoly, ktoré umožňujú komunikáciu prostredníctvom Wi-Fi, aj fungovanie bezdrôtových smerovačov a prístupových bodov. Wi-Fi umožňuje zariadeniam pripojiť sa k internetu prostredníctvom bezdrôtového smerovača [15]. Názov Wi-Fi vznikol v roku 1999 ako ochranná známka vytvorená obchodným združením Wi-Fi Alliance. Predchádzajúci názov, WaveLAN, bol vytvorený v roku 1991 spoločnosťami NCR Corporation a AT&T a slúžil ako predchodca názvu Wi-Fi [16]. Sieť Wi-Fi predstavuje lokálnu bezdrôtovú sieť, ktorú označujeme ako WLAN (Wireless Local Area Network). Táto sieť funguje tak, že dáta sú prenášané pomocou rádiových vln vo forme paketov. Každý paket obsahuje vrstvy informácií a inštrukcií spolu s jedinečnými adresami — MAC adresy [15].

2.3 Dostupné možnosti ovládania žalúzií

Táto časť poskytuje prehľad o rôznych technológiách používaných na ovládanie domácich zariadení. Medzi ne patria Bluetooth ovládače, ktoré umožňujú bezdrôtové ovládanie, ovládače s infračervenou LED, využívajúce infračervenú komunikáciu na diaľkové ovládanie, rádiové frekvenčné ovládače, ktoré prenášajú signály pomocou rádiových vln, fyzické nástenné vypínače, zastupujúce tradičný pohľad na ovládanie, a nakoniec inteligentné vypínače, pripojené do siete.

2.3.1 Bluetooth ovládače

Technológia Bluetooth bola vyvinutá švédskou spoločnosťou Ericsson v roku 1994. V roku 1998 sa k nim pripojili ďalšie spoločnosti, ktoré pomohli rozvinúť Bluetooth pre svoje produkty, ako sú bezdrôtové myši a klávesnice [1]. Tieto spoločnosti vytvorili Bluetooth Special Interest Group (SIG), čo je organizácia zameraná na udržiavanie a šírenie tejto technológie. Vytvorenie tejto organizácie znamenalo, že žiadna z týchto spoločností nevlastní ani vlastníť technológiu nebude [1]. Dnes poznáme technológiu Bluetooth pod štandardom IEEE 802.15.1. V súčasnosti, vďaka rozšíreniu a zlepšeniu technológie Bluetooth, sú k dispozícii Bluetooth ovládače, ktoré umožňujú napríklad aj ovládanie externých žalúzií [1].

2.3.2 Ovládače s infračervenou LED

Infračervená komunikácia (IR) patrí medzi jednu z najčastejšie používaných technológií na ovládanie zariadení a je jednou z jednoduchších a cenovo dostupnejších možností implementácie. Ovládače IR využívajú NEC IR prenosový protokol, kde logické stavy sú zakódované pomocou impulzov rôznych dĺžok, pričom HIGH reprezentuje logickú 1 a LOW logickú 0 [9]. Fungovanie spočíva v infračervenej LED dióde, ktorá vytvára svetlo s vlnovou dĺžkou okolo 950 nanometrov a frekvenciou 38 000 Hz, neviditeľné pre ľudské oko [9]. Nevýhodou IR komunikácie sú rôzne rušenia, ako rôzne druhy svetiel, slnečné žiarenie a iné IR žiarenie, ktoré môžu ovplyvniť signál [9]. Problémom je tiež absencia zabezpečenia, pretože medzi odosielateľom a príjemcom typicky neexistuje žiadny handshake, autentifikácia ani autorizácia. To môže viesť k ovládaniu zariadenia niekým iným bez potreby autorizácie [9].

2.3.3 Rádiové frekvenčné ovládače

Rádiové frekvenčné (RF) ovládanie je bežný spôsob riadenia rôznych zariadení, podobne ako infračervené ovládanie. RF ovládanie sa často používa v bránových diaľkových ovládačoch, kľúčoch k automobilom alebo hračkách. Na rozdiel od IR ovládania, RF nevyužíva svetelné signály, ale rádiové vlny s frekvenciou od 3 kHz do 300 GHz [12]. Signály sú kódované a odpovedajú binárnym príkazom pre konkrétne tlačidlo. Na druhej strane je prijímač, ktorý má za úlohu prijímať a dekódovať tieto signály [9]. RF ovládanie však môže trpieť rôznymi rušeniami, ako sú iné RF ovládače, mobilné telefóny či vysielачky. Niektoré pokročilejšie systémy riešia tento problém vysielaním na špecifických rádiových frekvenciách a použitím digitálnych adres, čo umožňuje prijímačovi rozlišovať medzi signálmi, ktoré má prijať a ktoré nie [9]. Jednou z výhod RF ovládania oproti IR je jeho väčší dosah. Zatiaľ čo IR dosahuje približne 10 metrov, RF môže dosiahnuť aj viac ako 100 metrov. Navyše, rádiové vlny dokážu preniknúť cez steny, na rozdiel od infračervenej technológie, ktorá vyžaduje priamu viditeľnosť medzi ovládačom a prijímačom [12].

2.3.4 Fyzické nástenné vypínače

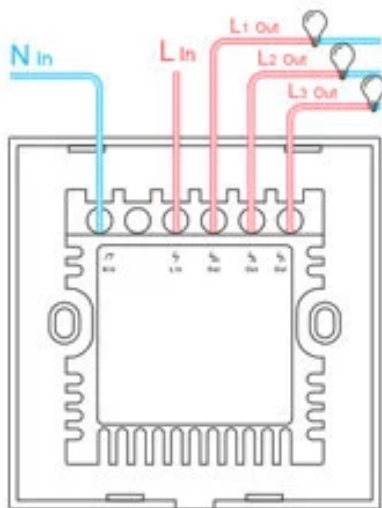
Vypínače sú zariadenia, ktoré slúžia na uzatváranie, otváranie alebo prepínanie elektrických obvodov. Sú bežnou súčasťou domácností a umožňujú ovládať široké spektrum elektrických zariadení. Existujú dva hlavné typy vypínačov: dočasné a udržiavané [10]. Dočasný vypínač sa po stlačení udržia v novej polohe len krátku chvíľu alebo sa vráti do pôvodného stavu po uvoľnení stlačenia. Udržiavané vypínače zase zotrávajú v novej polohe až do ďalšej zmeny stavu [10].

2.3.5 Inteligentné vypínače

Inteligentné vypínače sú zariadenia, ktoré majú možnosť pripojenia do nejakej siete. Najčastejšie sa pripájajú do siete pomocou WIFI. Oproti klasickým fyzickým nástenným vypínačom sú tie inteligentné schopné diaľkového ovládania aj mimo domova za pomoci telefónu alebo iného inteligentného zariadenia, nastavovať rutiny počas dňa, merať spotrebu elektrickej energie a mnoho ďalšieho. Nevýhodou je proces inštalácie, ktorá môže byť pre niekoho zložitejšia. Ďalšou nevýhodou je množstvo aplikácií, ktoré dané zariadenia vyžadujú.

2.4 Použitý Hardware

V tejto časti sa zaoberáme nástenným vypínačom Sonoff T2EU3C, ktorý umožňuje ovládanie rôznych elektrických zariadení v domácnostiach prostredníctvom dotykových tlačidiel, ktoré ovládajú relé napojené z elektrickej siete vid' obrázok 2.4. Obsahuje mikrokontrolér ESP8266 a poskytuje spoľahlivé a efektívne riadenie. Významnou výhodou je možnosť preprogramovania čipu vypínača pomocou platformy ESPHome. V ďalšej časti bude spomenutý mikrokontrolér ESP8266.



Obr. 2.4: Schéma možnosti zapojenia Sonoff T2EU3C. Dostupné z ⁷

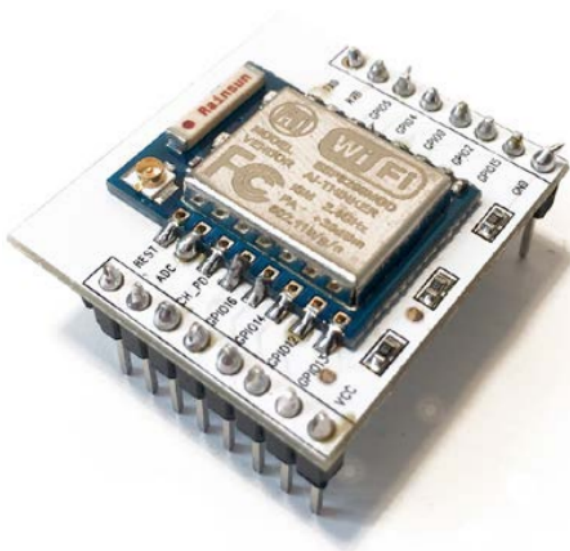
⁷Odkaz schémy vypínača: <https://itead.cc/product/sonoff-t2eu-rf-86-type-433-wall-sticky-rf-remote/>

2.4.1 Sonoff T2EU3C nástenný vypínač

Ide o 3-kanálový vypínač od spoločnosti Sonoff, ktorý ponúka možnosť ovládania rôznych elektrických zariadení v domácnostiach. Disponuje tromi dotykovými tlačidlami, ktoré sú vybavené LED podsvietením, čo umožňuje ľahké a intuitívne ovládanie. Vybavený je Wi-Fi, RF prijímačom a mikrokontrolérom ESP8266, ktorý zabezpečuje efektívne riadenie tohto zariadenia [14]. Tento vypínač umožňuje ovládanie buď pomocou aplikácie na smartfóne alebo aj fyzicky, pomocou tlačidiel vypínača. RF komunikácia tohto zariadenia pracuje na frekvencii 433.92 MHz, čo umožňuje bezdrôtovú komunikáciu so širokým spektrom kompatibilných zariadení [14]. Pre naše potreby je tento vypínač, konkrétnejšie jeho čip, možné preprogramovať pomocou ESPHome, čo nám umožní väčšiu kontrolu nad zariadením. Týmto spôsobom nebudeme závislí na aplikácii od spoločnosti Sonoff a ich serveroch. Tento prístup nám umožní rýchlejšiu odozvu vďaka využívaniu lokálnej siete.

2.4.2 Mikrokontrolér ESP8266

Mikrokontrolér ESP8266, viď obrázok 2.5, podporuje 3,3 V a 5 V, a ak sa napríklad skúša napájať čip ESP8266 z 3,3 V prichádzajúcich z dosky FTDI alebo z dosky Arduino, nebude fungovať správne. Preto pre väčšinu modulov ESP8266 potrebujeme samostatný zdroj napájania, ktorý môže bezpečne dodávať aspoň 300 mA [13]. Niektoré dosky majú integrovaný micro-USB port a regulátor napätia, ktorý môže poskytnúť požadovaný prúd ESP8266. Primárny spôsob použitia modulu ESP8266 je pomocou sériových príkazov, pretože čip je v podstate Wi-Fi sériový vysielač a prijímač [13].



Obr. 2.5: ESP8266 v konfigurácii ESP12 (ukážka pre predstavu). Prevzaté z [13].

Kapitola 3

Návrh

V tejto kapitole sa zameriame na návrh riešenia pre našu integráciu na ovládanie žalúzií. Naším cieľom je vytvoriť integráciu, ktorá ponúkne užívateľom efektívne a intuitívne riešenie pre ovládanie ich žalúzií a zabezpečí synchronizáciu medzi stavom v Home Assistante a reálnym stavom žalúzií. Táto kapitola sa zameriava na analýzu existujúcich konkurenčných riešení, návrh integrácie, vrátane logiky ovládania a automatizácií, inštalačný návrh a používateľské rozhranie.

3.1 Konkurenčné riešenia

Na platforme Home Assistant sa momentálne nachádzajú dve riešenia podobného problému. Avšak každé z týchto riešení má nedostatky, ktoré sa snažíme v tejto práci odstrániť. Jedno¹ riešenie neumožňuje nastavenie náklonu žalúzií, zatiaľ čo druhé² nedokáže detekovať stlačenie určitého vypínača alebo zmenu stavu spôsobenú prostredníctvom inej integrácie. Okrem toho nezvláda správne počítať polohu náklonu pri pohybe žalúzií — ak sa žalúzia pohne nahor, náklon, hoci reálny, sa na začiatku pohybu nastaví na úplne otvorený, a v prípade, že sa žalúzia pohne smerom nadol, náklon sa nastaví na úplne zatvorený. Toto spôsobuje problémy pri krátkych pohyboch so žalúziami a vedie k nesprávnej polohe náklonu zobrazovanej pre používateľa.

Tieto integrácie taktiež nemajú užívateľsky priateľské rozhranie na prvotnú inštaláciu alebo prípadnú úpravu zadaných hodnôt neskôr a vyžadujú úpravu súborov v konfiguračných súboroch Home Assistant — `configuration.yaml`. Toto môže byť pre niektorých používateľov menej intuitívne a takáto možnosť konfigurácie by sa už nemala používať. Spomenuté problémy by mala táto práca odstrániť, a tým spríjemniť používanie.

3.2 Funkčné a nefunkčné požiadavky

Pred začatím práce si musíme určiť funkčné a nefunkčné požiadavky pre naše riešenie. Bude sa jednať o kritéria, ktoré budeme chcieť pri vypracovaní riešenia dodržať. Funkčné požiadavky budú špecifikovať, aké konkrétne funkcie alebo operácie musí naše riešenie poskytovať. Nefunkčné požiadavky budú definovať kvalitatívne aspekty nášho riešenia.

¹Odkaz na prvé riešenie: <https://github.com/kotborealis/home-assistant-custom-components-cover-time-based-synced>

²Odkaz na druhé riešenie: <https://github.com/Sese-Schneider/ha-cover-time-based>

Medzi funkčné požiadavky bude spadať:

- **Ovládanie žalúzií:** Možnosť otvárať, zatvárať a zastavovať žalúzie cez Home Assistant.
- **Plánovanie a automatizácie:** Možnosť nastavenia plánov a vytvorenia automatizácií pre automatické riadenie žalúzií.
- **Manuálne ovládanie.**
- **Synchronizácia so zariadeniami mimo integráciu:** Schopnosť synchronizovať stav žalúzií s inými zariadeniami mimo našu integráciu.
- **Presné ovládanie pozície:** Možnosť presného nastavenia polohy žalúzií.
- **Dostupnosť a spoľahlivosť.**
- **Jednoduché spustenie a konfigurácia:** Intuitívny proces inštalácie a nastavenia.

Medzi nefunkčné požiadavky bude spadať:

- **Jednoduchosť používania:** Požívateľské rozhranie a intuitívna navigácia.
- **Vývoj a údržba:** Jednoduchý vývoj a údržba.
- **Dizajn a estetika:** Esteticky príjemné užívateľské rozhranie a jednoduchá integrácia.
- **Open source:** Zdrojový kód integrácie je voľne dostupný, umožňujúc užívateľom skúmať, upravovať a prispievať k jeho vývoju.

3.3 Príkazy ovládania

Ovládacie príkazy v tejto práci budú pochádzať konkrétne zo zariadenia Sonoff. Avšak práca by mala byť schopná rozpoznávať príkazy aj z iných podobných zariadení, ktoré budú mať dva stavy: zapnutý — on a vypnutý — off. Integrácia, keďže sa budeme snažiť o synchronizovanú prevádzku, znamená, že zariadenie budeme schopní ovládať z našej integrácie, teda z ovládacieho panela. Ale aj pripojené ovládacie zariadenie (vypínač, ktorý bude v home assistant ako entita) bude schopné ovládať našu integráciu. To bude veľkým plusom, pretože integrácia bude počítat aj so stlačením vypínača mimo integrácie a bude správne počítat čas, ktorý zostáva pri pohybe žalúziami nahor a nadol, a pre náklon. Synchronizovanú prevádzku dosiahneme tým, že budeme načúvať zmene stavu, a ak sa stav zmení a bude to náš ovládacie prvok, budeme vedieť, že máme zapnúť časovač.

3.3.1 Základná logika

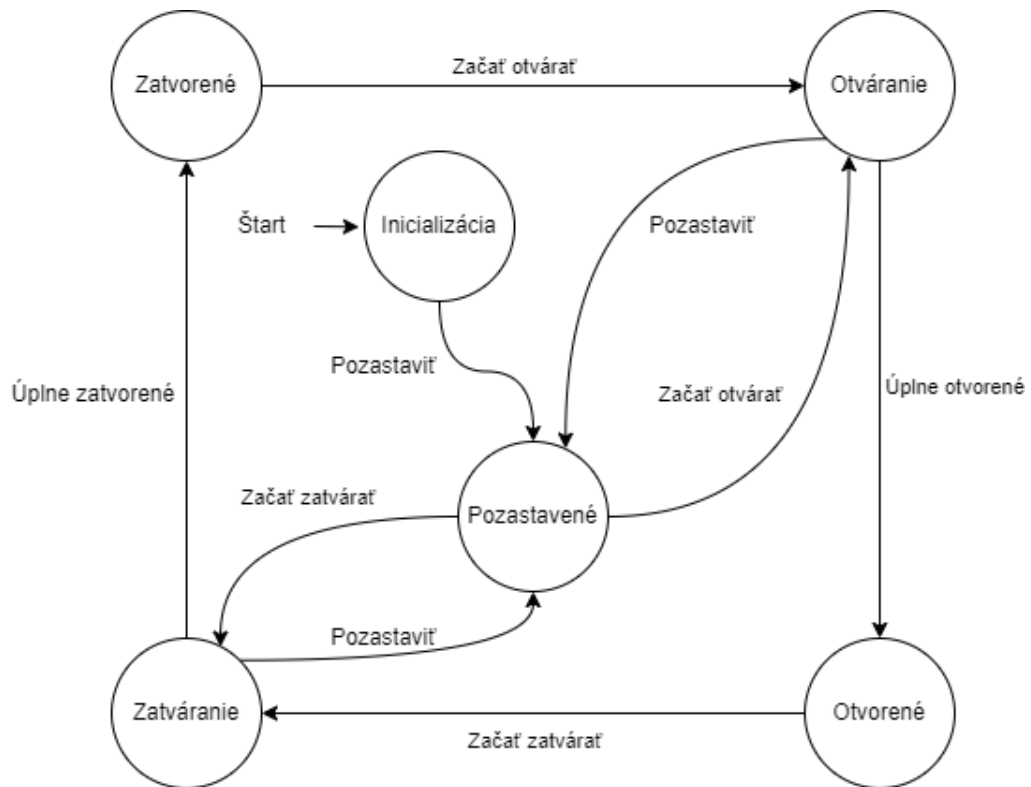
Základná logika systému bude postavená na správnom interpretovaní príkazov ovládania vid' obrázok 3.1, a udržiavaní synchronizácie medzi stavom v Home Assistane a reálnym stavom žalúzie. Pri príjme príkazu ho systém okamžite zistí ako, aj jeho obsah a určí, či ide o žiadosť o zapnutie alebo vypnutie daného tlačidla na ovládanie. Následne vykoná príslušnú akciu a aktualizuje svoj vnútorný stav.

V prípade, že príkaz pochádza z ovládacieho panela alebo pripojeného ovládacieho zariadenia, systém sa uistí, či sa daný príkaz voči stavu vie vykonať a ak nie, tak sa príkaz

nevykoná alebo vyvolá chybu. V prípade, že príkaz sa vykonať vie, integrácia vykoná príkaz a aktualizuje svoj vnútorný stav tak, aby bol v súlade s reálnym stavom zariadenia.

Ďalším dôležitým aspektom základnej logiky je schopnosť reagovať na zmeny stavu ovládacích prvkov, ako je stlačenie vypínača mimo integrovaného systému. Integrácia bude aktívne sledovať tieto udalosti a pri zmene stavu vykoná príslušné kroky na aktualizáciu svojho stavu a udržanie synchronizácie so zmenami.

Celkovo bude základná logika zabezpečovať správne spracovanie príkazov ovládania, udržiavanie synchronizácie stavmi a efektívne reagovanie na zmeny ovládania.



Obr. 3.1: Stavový diagram návrhu žalúzií.

3.4 Automatizácie

Zadanie ďalej špecifikuje pridanie automatizácií priamo do integrácie. Automatizácie sa budú pridávať aj preto, lebo spomenuté riešenia, ktoré na platforme existujú, nemajú takúto priamo zabudovanú možnosť. V prípade, že používateľ chce pridávať automatizácie, musí to vykonať cez Home Assistant automatizácie, čo pri veľkom množstve automatizácií môže byť zdĺhavé a nepohodlné pre používateľa. Naše automatizácie by sa pridávali ihneď pri konfigurácii, čo uľahčí proces automatizácie. Automatizácie sa budú spúšťať na základe udalostí, ktoré nastanú, ako je čas, východ a západ slnka, rýchlosť vetra alebo nepriaznivé počasie. Používateľ by mal byť schopný nastaviť automatizácie pre jednotlivé žalúzie zvlášť, aby vyhovovali jeho jedinečným potrebám.

3.4.1 Automatizácia na základe času

Automatizácie, ktoré budeme spúšťať na základe času, budú zahŕňať otváranie a zatváranie žalúzií podľa denného času, pričom používateľ bude mať možnosť zvoliť si čas na otvorenie alebo zatvorenie. V praxi to znamená, že ak si používateľ nastaví, že žalúzie sa majú zavrieť o 15:00, tak sa v danej dobe každý deň automaticky zatvoria.

Ďalšia automatizácia, ktorá bude mať ako spúšťačiu udalosť čas, bude otváranie a zatváranie žalúzií podľa času na základe informácií o východe a západe slnka, ktoré automaticky zistí integrácia z Home Assistantu. Týmto spôsobom si bude používateľ môcť nastaviť, že večer, keď slnko zapadne, sa žalúzie na noc zatvoria, a ráno, keď slnko vychádza, sa opäť otvoria. Pri tejto automatizácii bude možné nastaviť aj offset, čo znamená, že sa bude dať určiť, že sa žalúzie budú rolovať buď skôr alebo neskôr oproti času východu alebo západu slnka.

Jedna z posledných automatizácií, ktoré súvisia s časom, je stmievanie interiéru podľa polohy slnka. To znamená, že pomocou času, ktoré udáva východ a západ slnka, sa určí poloha slnka na oblohe a pri pohybe žalúzií sa tienenie bude otvárať, aby sa do priestoru dostalo čo najviac svetla.

3.4.2 Prepojenie iných vypínačov na svetlo

Pri tejto automatizácii sa bude sledovať, či nastane súčasne dvojica udalostí, a to, či už slnko zapadlo a je noc a zároveň, či sme používateľom zvoleným vypínačom rozsvietili svetlo. V prípade, že takáto situácia nastane, žalúzie sa automaticky zatvoria.

3.4.3 Ochrana žalúzií pri daždi a vetre

Táto automatizácia bude automaticky zatvárať žalúzie v prípade nepriaznivého počasia alebo silného vetra. Na dosiahnutie tohto cieľa plánujeme integrovať vhodné API pre počasia, ktoré nám poskytnú informácie o aktuálnej rýchlosti vetra a výstrahách pred nepriaznivým počasím pre daný deň. Integrácia bude schopná automaticky zisťovať polohu, aby používateľ nemusel zadávať údaje manuálne.

Ak by táto automatizácia bola používaná v rámci Českej alebo Slovenskej republiky, môžeme využiť API od miestnych hydrometeorologických ústavov, čo by bol Český hydrometeorologický ústav alebo Slovenský hydrometeorologický ústav. Avšak, keďže testovať a používať integráciu môžu a budú aj používatelia z iných krajín, vhodným riešením bude bezplatné a open-source API od spoločnosti Open-Meteo. Toto API nám poskytne informácie o rýchlosti vetra a kódy na predpoveď počasia od Svetovej meteorologickej organizácie na daný deň.

3.5 Inštalačný návrh

Inštalácia v Home Assistane by mala byť pre používateľa jednoduchá. Preto sa budeme snažiť využiť možnosti Home Assistantu tak, aby sa používateľ vyhol úprave alebo prepisovaniu `.yaml` súborov, a to pri inštalácii, pridávaní roliet a ich prípadnej úprave, ak by to bolo potrebné.

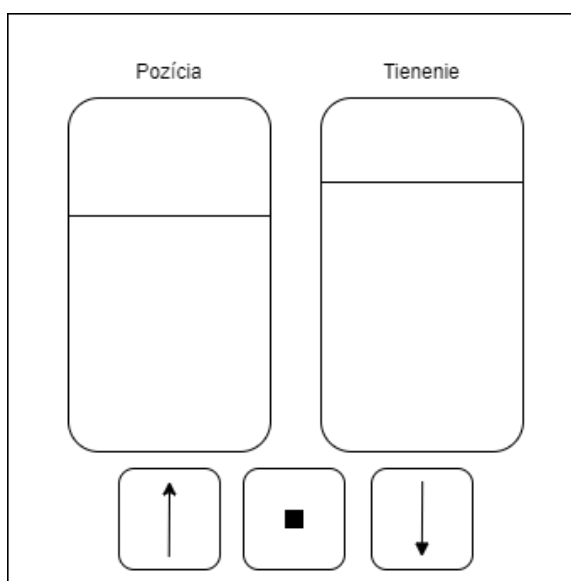
3.5.1 Prvotná konfigurácia a úprava konfigurácie

Po nainštalovaní by malo byť možné pridať ovládanie pre danú žalúziu. Používateľ by mal mať k dispozícii dopredu namerané hodnoty, konkrétne časy trvania otvárania a zatvárania žalúzie, a prípadne aj dobu, počas ktorej sa žalúzia stemňuje zo stavu neprepustenia svetla až po úplne otvorenú pozíciu. Po zadaní názvu a vyplnení hodnôt bude môcť používateľ tiež vybrať automatizácie, ktoré chce pre danú žalúziu aktivovať.

Po uložení konfigurácie by mu malo byť umožnené pridať ovládanie na hlavný ovládací panel, odkiaľ bude môcť žalúzie manuálne ovládať. Na hlavnom paneli by malo byť možné kontrolovať aj časy a v prípade, že by niečo nefungovalo správne, ich upraviť v nastaveniach podľa potreby. Úpravy nastavení z prvotnej konfigurácie budú vykonávané na rovnakom mieste ako prvotná inštalácia v nastaveniach.

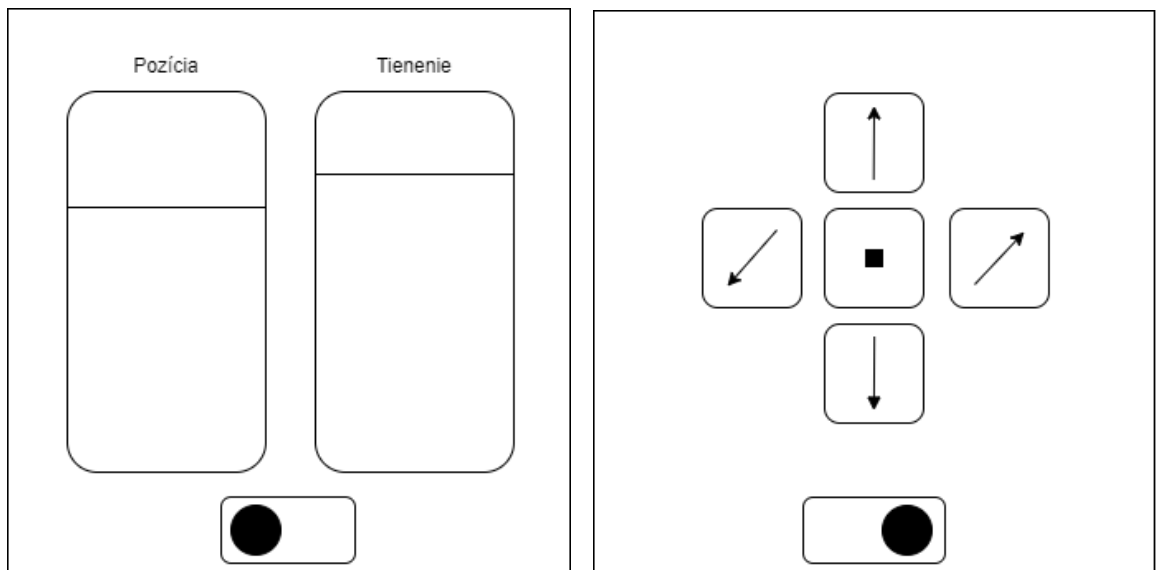
3.6 Používateľské rozhranie a používanie

Pri používaní integrácie by používateľ mal rozumieť tomu čo ovláda, ktorým tlačidlom. To by v tejto práci nemal byť problém, keďže ide o ovládanie iba nahor, nadol a stop, kde sa dá rozšíriť ovládanie o náklon, inak povedané tienenie interiéru. Ale taktiež je tu možnosť ovládania pomocou posuvného tlačidla, kde by používateľ vedel zadať presnú hodnotu, ktorú by chcel a žalúzia by sa automaticky posunula na danú hodnotu. To isté bude možné spraviť pre tienenie.

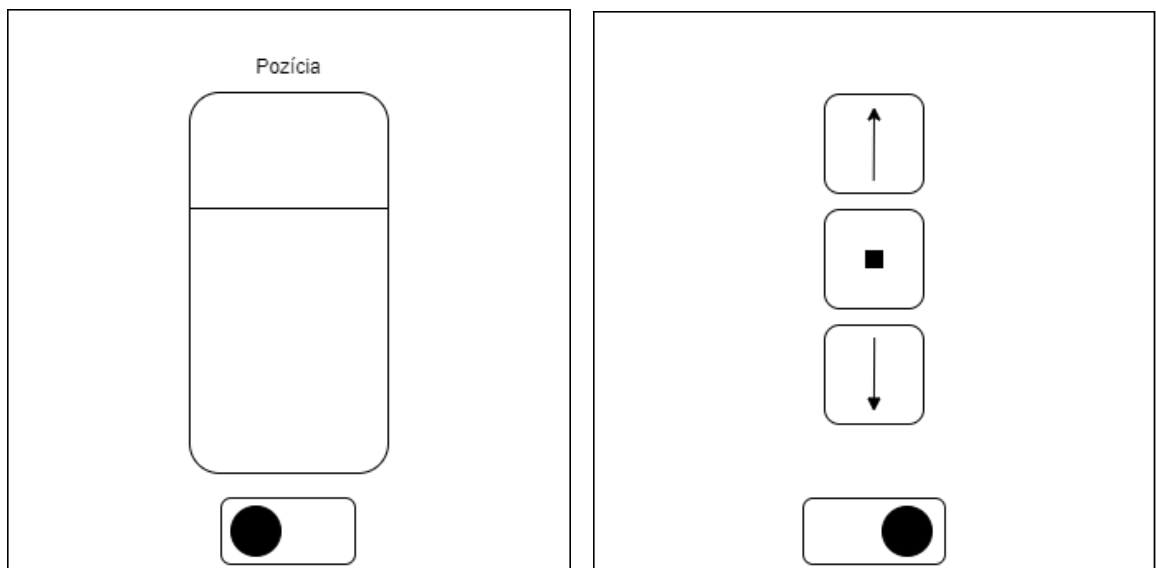


Obr. 3.2: Prvotný návrh používateľského rozhrania.

Návrh vyššie na obrázku 3.2, vznikol ako prvý, keď sa rodili nápady na používateľské rozhranie. Malo ísť o jednoduchý panel, v ktorom by bolo možné ovládať presnú pozíciu a tienenie s tým, že by bola možnosť ovládania aj pomocou tlačidiel. Následný návrh, viď obrázky 3.3 a 3.4, vznikol po naštudovaní možností v Home Assistane, ktoré sú spomenuté v sekcii nižšie.



Obr. 3.3: Návrh s možnosťou tinenia po naštudovaní možností Home Assistant.



Obr. 3.4: Návrh bez možnosti tinenia po naštudovaní možností Home Assistant.

Využitie Home Assistant

Importovanie `homeassistant.components.cover` v Home Assistane umožňuje pridanie základných ovládacích prvkov do Home Assistantu. Ovládacie prvky môžu zahŕňať posuvné tlačidlo na nastavenie polohy a tiež posuvné tlačidlo na nastavenie tienenie, ako aj jednoduché tlačidlá, ktoré určujú smer pohybu žalúzie a prípadne náklonu.

Po importovaní `homeassistant.components.cover` by sme mali získať základný vizuál ovládania na hlavnom paneli. Návrh ovládania by mal byť intuitívny, na prvej stránke by mali byť umiestnené tlačidlá na zmenu polohy, tienenia a zastavenia. A na ďalšej stránke by mali byť umiestnené dve posuvné tlačidlá na presné nastavenie polohy a tienenia, pri ktorých ak sa nastaví daná hodnota, tak po dosiahnutí sa pohyb žalúzie zastaví. Pri pohybe by sa žalúzia mala zastaviť po dosiahnutí otvorenia na 100 percent alebo 0 percent, to isté bude platiť pre tienenie.

3.7 Hlavné problémy

V tejto časti sa zamyslíme nad hlavnými problémami, ktoré sa vyskytujú v kontexte nášho riešenia. Identifikujeme kľúčové výzvy, ktoré je potrebné prekonať pre úspešnú implementáciu integrácie. Hľadáme spôsoby, ako týmto problémom čeliť a riešiť ich. Jedným z hlavných problémov, ktoré musíme vyriešiť, je správne synchronizovať pohyb s kalkulačkou spojenou s tienením žalúzií. Ďalším dôležitým aspektom je zabezpečiť spoľahlivú a rýchlu odozvu systému, či už pri ovládaní pomocou inteligentného vypínača alebo prostredníctvom iných ovládacích prvkov. Okrem toho je dôležité, aby sme zabezpečili jednoduchú a intuitívnu konfiguráciu a možnosť následnej úpravy nastavení pre používateľov.

3.7.1 Výpočet času

Pri výpočte času sa bude používať kalkulačka, ktorá je už používaná pri obdobných riešeniach a funguje správne. Použila by sa kalkulačka `XKNX — travelcalculator.py`. Ide o asynchrónnu knižnicu napísanú v jazyku Python. Táto kalkulačka poskytuje veľa užitočných funkcií ktoré sa budú pri implementovaní integrácie používať.

Táto kalkulačka berie do úvahy rôzne faktory, ako je čas a smer pohybu, aby určila, ako sa zariadenie pohybuje. Napríklad, ak chceme otvoriť žalúziu na 50 percent z inej polohy, kalkulačka vypočíta, ako dlho sa má daná žalúzia pohybovať, aby dosiahla požadovanú polohu. Keď žalúzia dosiahne zadanú polohu, kalkulačka bude vo funkcii odosielať signál oznamujúci, že žalúzia dosiahla cieľovú polohu.

Kalkulačka tiež zohľadňuje obmedzenia zariadenia, ako sú maximálna a minimálna možná poloha a rýchlosť pohybu. To znamená, že ak nastavíte žalúziu na otvorenie na 100 percent, kalkulačka bude vedieť, že sa žalúzia má pohybovať nahor a zastaviť sa, keď dosiahne maximálnu otvorenú polohu.

Keďže budeme v integrácii pracovať s dvoma rôznymi údajmi — pohyb nahor, nadol a tienenie, ktoré budeme musieť počítať, tak budeme danú kalkulačku využívať na obe tieto aktivity. Tu nastane problém, ktorý sa bude musieť správne naimplementovať, a to, ak sa žalúzia pohybuje, tak sa zároveň počíta aj s tienením ktoré sa pohybuje taktiež. Môžeme uviesť príklad: žalúzia je v polohe úplne otvorená a chceme, aby začala cestovať nadol. V momente, keď začne žalúzia cestovať, tienenie by sa malo postupne nastaviť na 0. Nie ako v prípade konkurenčného riešenia, kde sa tienenie okamžite posunie na 0 a vzniká tak problém, že žalúzie v integrácii udávajú nesprávny údaj o aktuálnej polohe tienenia.

Podobná situácia ešte vzniká pri pohybe nahor, keď by sa tienenie malo postupne nastaviť do otvorenej pozície.

3.7.2 Synchronizácia s ovládaním mimo našej integrácie

Tento problém bol taktiež naznačený v časti 3.1. Pod synchronizáciou sa rozumie, že kalkulačka zohľadňuje aj ovládanie mimo nášho riešenia. Tento cieľ je možné dosiahnuť tým, že budeme určitým spôsobom sledovať zmeny stavov a v prípade, že ide o naše zvolené ovládanie žalúzie, naša kalkulačka bude brať do úvahy pohyb smerom nahor, nadol a aj stop. A do tejto logiky bude zapracované taktiež tienenie.

3.7.3 Odozva

Pri našom riešení, pri použití inteligentného vypínača Sonoff, by vznikol problém, že odozva pri komerčnom pripojení zariadenia cez server danej spoločnosti by mohla spôsobovať nepríjemnosti v tom, že integrácia by mohla pôsobiť oneskorene pri ovládaní pomocou vypínača na stene. Tento problém by malo vyriešiť použitie ESPHome. ESPHome umožňuje lokálne integrovať a riadiť zariadenie, čo zvyšuje rýchlosť a spoľahlivosť odozvy. Týmto sa minimalizujú oneskorenia a zvyšuje sa celková používateľská spokojnosť.

3.7.4 Konfigurácia a jej úprava

Pri už spomenutých riešeniach je spomenuté, že prvotná konfigurácia a následná úprava konfigurácie nie je z používateľského hľadiska veľmi priateľská. V týchto riešeniach sa pracuje s konfiguračnými súbormi, ktoré, ak nie sú správne nakonfigurované, môže sa stať, že Home Assistant konfiguráciu z tejto integrácie nemusí vôbec spustiť, a priamo pri písaní v konfiguračných súboroch neexistuje žiadna kontrola zadaných údajov. Naše riešenie by malo túto situáciu riešiť pomocou takzvaného `config flow` pre prvotnú konfiguráciu žalúzií a `option flow`, ktorý sa bude starať o úpravu zadaných údajov z prvotnej konfigurácie. Pri tejto metóde bude možné priamo kontrolovať zadané údaje tak, aby nedošlo k neúmyselnému zadaniu nesprávnych údajov. Kontrola údajov bude potrebná z dôvodu, že ak by boli zadané nesprávne údaje, mohlo by dôjsť k nefunkčnosti programu alebo dokonca k poškodeniu žalúzií.

3.8 Distribúcia

V tejto časti sa spomína, akým spôsobom sa bude dané riešenie šíriť a ako získať spätnú väzbu od používateľov. Na to budú použité rôzne fóra a komunikačné kanály, kde sa riešenie predstaví, popíše a bude zdieľané. Toto umožní získať spätnú väzbu od používateľov a zistiť potenciál daného riešenia v porovnaní s ostatnými.

Pridanie na platformu HACS

Jeden z nápadov pre ľahšie pridanie tejto integrácie do Home Assistantu je pridanie na HACS. Konkrétne, aby sa táto integrácia nachádzala na danom komunitnom obchode a ponúkla ešte jednoduchšie pridanie do Home Assistantu. V prvotnej fáze sa tieto integrácie pridávajú ako integrácie na mieru a nenachádzajú sa v HACS priamo, ale musia byť pridané s pár krokmi navyše. Ak by bola integrácia priamo v HACS, stačilo by si ju nájsť, stiahnuť a následne po reštarte Home Assistantu začať používať.

Kapitola 4

Implementácia

Implementácia tohto riešenia nasledovala návrh z kapitoly 3. Hlavná časť implementácie je vykonaná v jazyku Python, podobne ako je tvorená väčšina Home Assistantu. V prvom kroku sme sa venovali spracovaniu údajov z konfigurácie, ich kontrole a prepojeniu tak, aby sme ich mohli efektívne využiť. Následne sme riešili samotnú logiku programu a integráciu externej knižnice umožňujúcej výpočet pozície. Počas implementácie sme museli riešiť aj problémy s konkurenčným načítaním pri synchronizácii s ovládaním zariadenia mimo integráciu, čo spôsobovalo problémy s posunom žalúzií. Po vyriešení tohto problému a overení správnej funkčnosti ovládania sme mohli pridať aj automatizácie, ktoré boli spomenuté v zadaní. Počas implementácie sa nám podarilo riešenie rozšíriť o niektoré prvky navyše v porovnaní s návrhom, čím sme zlepšili použiteľnosť integrácie. Výsledkom našej práce je funkčné riešenie pripravené na testovanie a distribúciu medzi používateľov.

4.1 Spracovanie údajov pri konfigurácii

Na získanie potrebných údajov na ovládanie od používateľa bol zvolený spôsob, pri ktorom používateľ nemusí ručne upravovať konfiguračný súbor `configuration.yaml`. Použitím `config flow` na inicializáciu a `option flow` na následné úpravy konfigurácie sa zabezpečuje, že proces je pre používateľa prívetivejší. Tento postup vytvára formulár, ktorý používateľ vyplní, a ak zadá neplatnú hodnotu, upozorní ho a neuloží sa, pokiaľ nie je správne vyplnený. Na kontrolu sa využíva knižnica `voluptuous`. Táto knižnica slúži na overovanie a validáciu dát.

```
vol.Required("ent_name"): str,
vol.Required("entity_up", default=None): vol.In(self._get_entity_ids()),
vol.Required("time_up"): vol.All(vol.Coerce(float), vol.Range(min=0)),
```

Kód 4.1: Ukážka použitia knižnice `voluptuous`.

Na ukážke vyššie, ktorá je prebratá z nášho riešenia, je vidieť, ako používame knižnicu `voluptuous`. V tomto príklade kontrolujeme, či premenná `ent_name` obsahuje string. Pri `entity_up` povolíme výber iba entít, ktoré sa nachádzajú v Home Assistantovi, aby sme predišli chybám zo strany používateľa pri zadávaní názvu a pri `time_up` kontrolujeme, či ide o číslo vo formáte `float` a že číslo nie je záporné. V prípade nesprávne zadaných hodnôt sa vyvolá chybové hlásenie a upozorní používateľa.

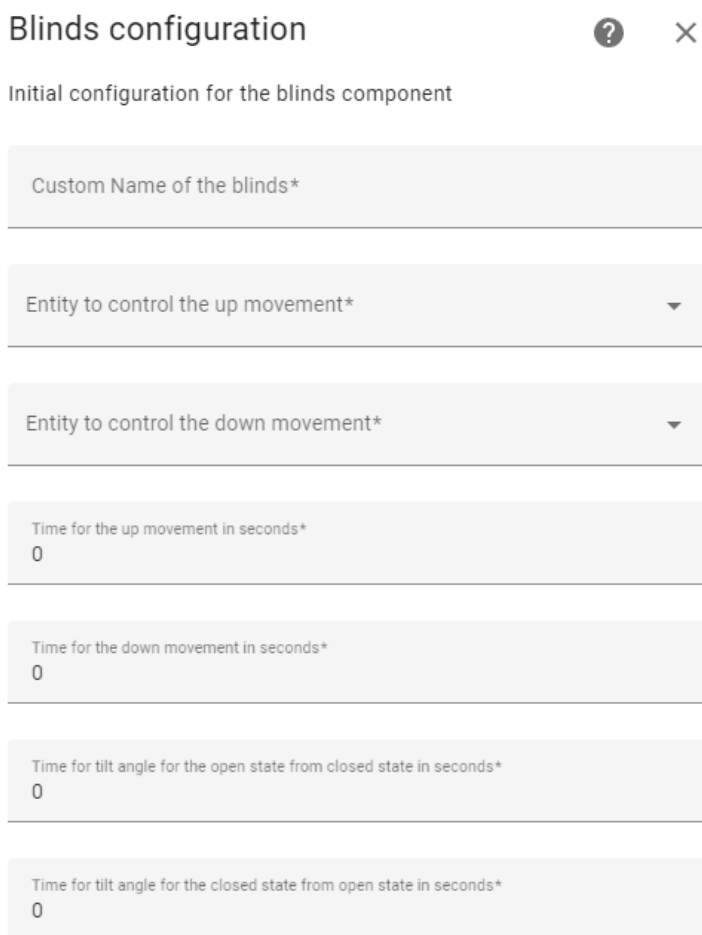
Vo formulári sú od používateľa požadované údaje, ako je názov danej žalúzie, čas, ako dlho žalúzia cestuje nahor a dole, názov prvku, ktorý ovláda smer nahor a smer nadol, čas,

ako dlho trvá žalúzii tieniť do úplnej pozície, úplné zatienenie aj opakované, a ďalšie údaje týkajúce sa automatizácie.

Niektoré z týchto údajov sú povinné na vyplnenie, zatiaľ čo iné sú voliteľné. Medzi povinné údaje patria hlavne názov danej žalúzie, čas, kedy sa pohybuje smerom nahor a nadol, a samotné ovládacie prvky, ako aj zvolená automatizácia, ktorú používateľ chce aktivovať. Zvyšné údaje sa týkajú iba informácií o automatizáciách a doplnkových funkciách integrácie.

4.1.1 Vizuál formuláru

Formulár viď obrázok 4.1, ktorý bol používaný nielen na úvodnú konfiguráciu, ale aj na jej následnú úpravu, mal pevne danú štruktúru. Možnosti úprav zahŕňali pridanie názvu, odkazu na read.me na GitHub stránkach, popisu formulára a poradia zobrazovaných údajov. Tieto údaje boli usporiadané pod sebou a bolo možné pridávať popisky a prednastavené hodnoty.



Blinds configuration

Initial configuration for the blinds component

Custom Name of the blinds*

Entity to control the up movement*

Entity to control the down movement*

Time for the up movement in seconds*
0

Time for the down movement in seconds*
0

Time for tilt angle for the open state from closed state in seconds*
0

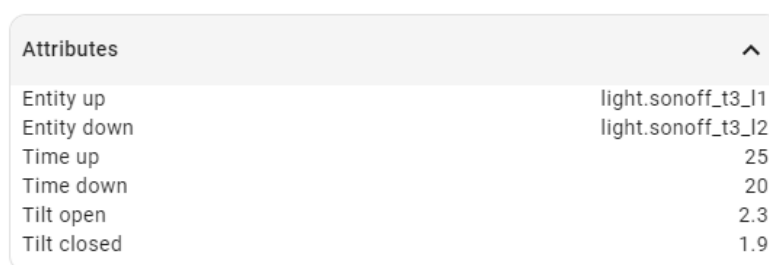
Time for tilt angle for the closed state from open state in seconds*
0

Obr. 4.1: Formulár pri prvotnej konfigurácii.

4.1.2 Prepojenie údajov a ich zobrazenie

Na prepojenie údajov sme použili funkciu `async_setup_entry` v súbore `__init__.py`, ktorá prepojí dáta z `config_flow` do doménového priestoru (domain) v Home Assistant. Táto funkcia sa používa na uloženie konfiguračných údajov pre daný záznam (entry) v doménovom priestore. Ďalej je použitá metóda `async_forward_entry_setup` na prepojenie konfiguračného záznamu s integráciou.

V súbore `cover.py` sa následne priradia hodnoty z konfiguračného záznamu do lokálnej premennej. Na zobrazenie týchto hodnôt sa na testovacie účely použila funkcia `extra_state_attributes` označená ako `@property`, ktorá zobrazila zadané údaje na hlavnom paneli vid' obrázok 4.2. Po overení, že zadané údaje sú správne prepojené a správne sa menia pri úprave konfigurácie, sa mohlo pokračovať ďalej. Údaje zobrazované pomocou tej istej funkcie `extra_state_attributes` zostali aj v finálnej verzii programu keďže sa ukázali ako užitočné pri používaní a prípadnej potrebnej úprave údajov a identifikácii chýb.



Attributes		^
Entity up	light.sonoff_t3_l1	
Entity down	light.sonoff_t3_l2	
Time up	25	
Time down	20	
Tilt open	2.3	
Tilt closed	1.9	

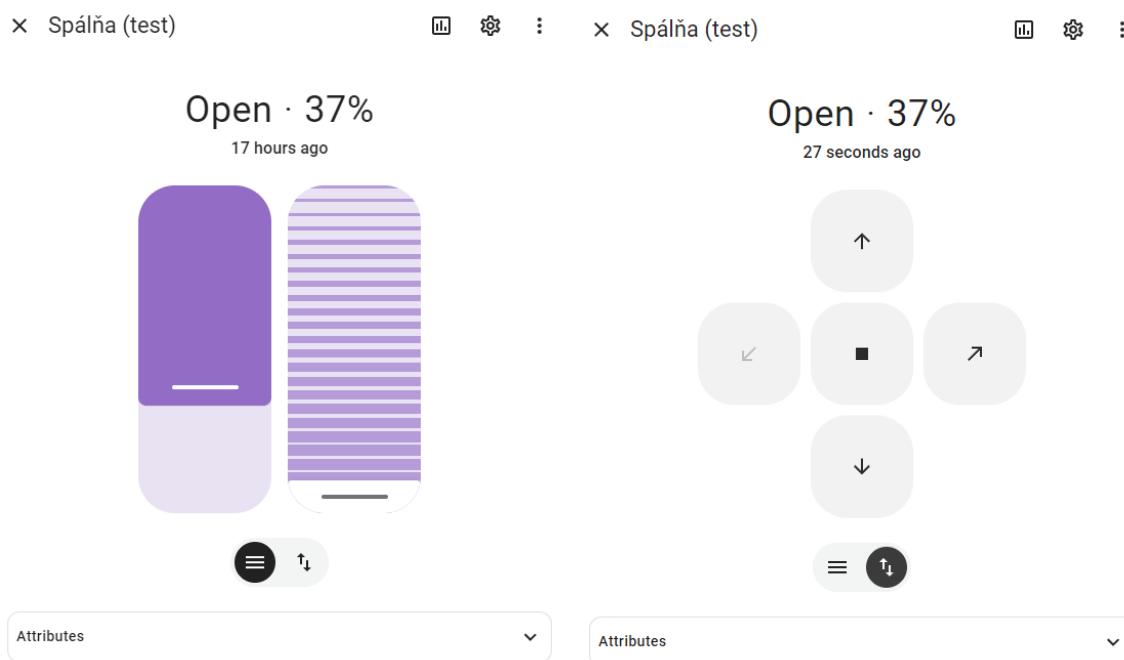
Obr. 4.2: Zobrazované údaje po kliknutí možnosti pod ovládacími prvkami.

4.2 Pridanie ovládacích prvkov

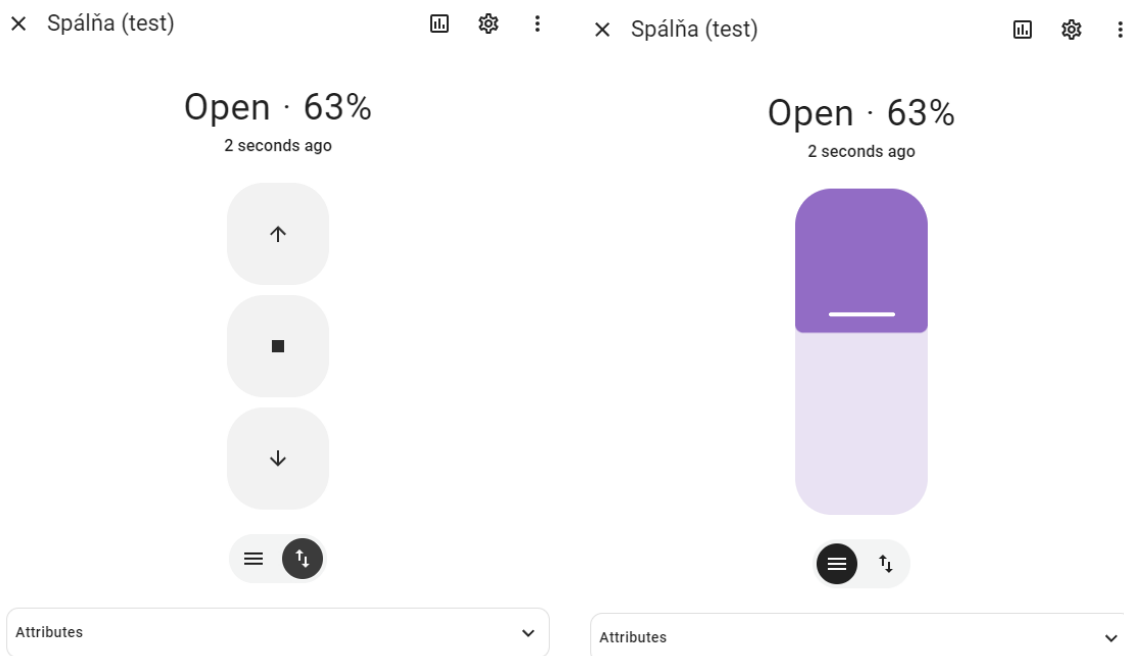
Ovládacie prvky, teda klasické tlačidlá a posuvné tlačidlá, sú pridané pomocou knižnice `homeassistant.components.cover`. Táto knižnica sa tiež stará o posielanie dát z používateľského rozhrania a spúšťanie funkcií pri použití ovládacích prvkov.

Pomocou tejto knižnice môžeme pridať 5 klasických ovládacích tlačidiel, ako sú 3 tlačidlá pre klasické ovládanie žalúzií: otvorenie—`CoverEntityFeature.OPEN`, smer zatvorenie `CoverEntityFeature.CLOSE`, zastavenie pohybu `CoverEntityFeature.STOP`, alebo 2 tlačidlá pre nastavovanie tienenia žalúzií: otvorenie `CoverEntityFeature.OPEN_TILT` a zatvorenie `CoverEntityFeature.CLOSE_TILT`. Taktiež dokážeme pridať 2 posuvné tlačidlá a to na nastavenie presnej polohy žalúzie `CoverEntityFeature.SET_POSITION` a tlačidlo pre nastavenie tienenia `CoverEntityFeature.SET_TILT_POSITION`. Posuvné tlačidlá ponúkajú svoju hodnotu v percentách. Výsledný vizuál je vidieť na obrázkoch 4.3, 4.4 a 4.5

Každé z týchto tlačidiel pri svojom stlačení aktivuje dopredu definovanú funkciu, ktorá je upravená tak, aby sa správala podľa potreby. Napríklad, ak stlačíme tlačidlo pre smer hore, použijeme údaje z formulára na identifikáciu entity, ktorá ovláda žalúziu smerom hore, spustíme ju a kalkulačka bude vnútorne aktualizovať aktuálnu polohu žalúzie. Ak použijeme posuvné tlačidlo, zistíme aktuálnu polohu a podľa nej určíme smer pohybu. Po stlačení tlačidla stop zastavíme akýkoľvek pohyb žalúzie. Konkrétnejšie sú tieto pohyby popísané v logike programu.



Obr. 4.3: Vizuál ovládania s ovládaním tienenia.



Obr. 4.4: Vizuál ovládania bez ovládania tienenia.

4.2.1 Vizuál ovládania

Pri riešení vizuálu ovládania, ako pri riešení formuláru, sa zistilo, že vizuál sa nedá veľmi meniť. Prvky sú pevne dané v rámci knižnice `homeassistant.components.cover`, ktorá tieto prvky spravuje. Jednou z vecí, ktorá sa ale mení je to, ak používateľ zadá, či chce alebo nechce ovládať tienenie. Ak áno, zobrazujú sa mu aj ovládacie prvky ovládajúce tienenie, no ak tienenie ovládať nechce, tieto prvky sa odstránia.

Na obrázku je vidieť ovládanie z hlavného panelu, kde je minimalistický dizajn a poskytuje základné ovládanie. Po kliknutí sa toto ovládanie rozšíri o už spomínané prvky.



Obr. 4.5: Ovládanie z hlavného panelu.

4.3 Logika programu

Logika programu sa točí hlavne okolo sledovania stavov vypínačov spomenuté v sekcii [4.3.2](#) a správneho počítania pozície pomocou kalkulačky spomenuté v sekcii [4.3.1](#). Home Assistant sa už postará o veľa vecí, ktorými uľahčuje prácu na tejto integrácii, najmä pokiaľ ide o spúšťanie príkazov z frontendu Lovelace UI do backendu.

Na začiatku programu, po prvotnom nastavení alebo prípadnej úprave konfigurácie, sa hodnoty z formulára priradia do lokálnych premenných a nastaví sa kalkulačky podľa vstupných údajov zadaných vo formulári. Následne sa čaká na príkazy buď z frontendu alebo z iného zariadenia, ktoré nie je súčasťou integrácie, ale ovláda danú žalúziu.

Príkazy majú vopred daný názov, pričom si následne len nastavíme, čo sa má vykonať pri ich spustení. Pri spustení príkazov dostaneme prípadné doplňujúce údaje ohľadom ich pozície. To znamená, že napríklad, ak sa stlačí tlačidlo na zatvorenie, spustí sa príkaz `async def async_close_cover`. Tento príkaz najprv skontroluje, či sa žalúzia nenachádza už v úplne zatvorenej pozícii. Ak nie, začne pohyb smerom nadol pomocou `_async_handle_command` a spustí sa počítanie kalkulačky `start_travel_down()`. Pri používaní tienenia sa tiež aktualizuje pomocou druhej kalkulačky s rovnakým príkazom `start_travel_down()`. Príkaz `_async_handle_command` následne ďalej zapne fyzické relé, čo dá do pohybu žalúziu. Žalúzia potom cestuje smerom nadol až do momentu, kým ju používateľ nezastaví pomocou tlačidla na zastavenie, alebo ak dosiahne pozíciu 0, čo značí, že je na konci cesty. Rovnakým spôsobom funguje aj cestovanie smerom nahor, ibaže v opačnom smere.

Pri použití posuvných tlačidiel je postup podobný, avšak pri zadaní pozície funkcia dostane údaj `kwargs`, v ktorom sa nachádza hodnota, ktorá hovorí o pozícii, ktorú používateľ chce pre žalúziu nastaviť. Po obdržaní tejto hodnoty ju porovnáme s aktuálnou pozíciou `current_position()` a rozhodneme, či bude žalúzia cestovať smerom nahor alebo nadol. V každom prípade sa skontroluje, či sa žalúzia nenachádza na konci svojej dráhy, teda na 100 alebo 0, a následne začne cestovať na zvolenú pozíciu, pričom aktualizuje kalkulačku hlavnej pozície aj tienenia, keďže to sa pohybuje tiež. Pohyb žalúzie sa zastaví, ak dosiahne danú pozíciu, alebo ak ho používateľ zastaví príkazom STOP. Rovnakým princípom funguje aj nakláňanie pomocou posuvného tlačidla, s tým rozdielom, že sa pracuje s druhou kalkulačkou.

4.3.1 Výpočet pozície

Na výpočet pozície sa využíva kalkulačka `xknx`¹. Táto kalkulačka implementuje algoritmus na výpočet aktuálnej polohy na základe poskytnutých vstupných údajov. Údaje sa v našom prípade delia do dvoch kalkulačiek: prvá slúži na výpočet hlavnej pozície, kde časy zodpovedajú cestovaniu žalúzie z úplne vrchnej pozície na úplne dolnú pozíciu a z úplne dolnej pozície na úplne vrchnú. Druhá kalkulačka slúži na výpočet pozície, ak sa používateľ rozhodne využiť možnosť tienenia. V tomto prípade daná kalkulačka využíva čas z polohy úplne otvoreného tienenia na úplne zatvorené tienenie a tiež z úplne zatvorenej na úplne otvorenú pozíciu tienenia.

Môžeme uviesť príklad. Uvažujme, že by naša žalúzia cestovala z vrchu nadol, pre jednoduchosť 100 sekúnd. Chceme sa dostať z pozície 90 % na pozíciu 60 %.

- V čase 0 sa nachádzame na pozícii 90 %. (vráti sa: koncová pozícia nie je dosiahnutá).
- V čase 10 sa nachádzame na pozícii 80 %. (vráti sa: koncová pozícia nie je dosiahnutá).
- V čase 20 sa nachádzame na pozícii 70 %. (vráti sa: koncová pozícia nie je dosiahnutá).
- V čase 30 sa nachádzame na pozícii 60 %. (vráti sa: koncová pozícia je dosiahnutá).

Kalkulačka sa používa pomocou jednoduchých príkazov, ako napríklad pri cestovaní smerom nahor sa spustí `start_travel_up` alebo smerom nadol sa spustí `start_travel_down`. Užitočné sú aj volacie príkazy, ako je zistenie aktuálnej pozície `current_position` alebo overenie, či sa žalúzie momentálne pohybujú a kalkulačka počíta `is_traveling`. Po dosiahnutí zadaného bodu kalkulačka vráti hodnotu `position_reached`. Okrem toho kalkulačka disponuje aj ďalšími funkciami, ale vyššie spomenuté boli v našej práci najčastejšie využívané.

4.3.2 Riešenie problému synchronizácie ovládania

Problém synchronizácie ovládania vznikol v momente implementácie, keď sme sa snažili ovládať zariadenie mimo Home Assistantu. Ovládanie mimo Home Assistantu, napríklad pomocou nástenného vypínača, spôsobovalo nesprávne fungovanie kalkulačky a rýchle "klikanie" relé na vypínači. Dôvodom problému bolo to, že integrácia nedokázala rozoznať, aký signál prichádza a ako s ním má následne vysporiadať. Teda sa stávalo, že ovládanie z hlavného panela Home Assistantu hovorilo `stop`, zatiaľ čo vypínač hovoril `hore` alebo `dole`, prípadne naopak.

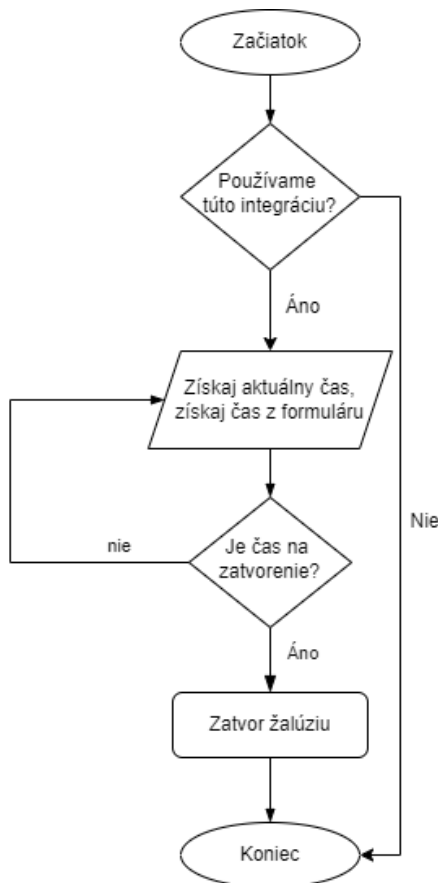
Na riešenie tohto problému sme zvolili asynchrónne sledovanie stavu, ktoré monitoruje zmeny stavov vstupných zariadení a vykonáva príslušné akcie na základe týchto zmien. Týmto spôsobom sa nemôže stať, že bude naraz zapnuté ovládanie smeru `hore` alebo `dole`. Okrem toho sa nestane, že je zadaný príkaz `hore` a zároveň zostane zapnutý príkaz `stop`. Taktiež, ak by nastala situácia, že sa stlačí tlačidlo proti aktuálnemu pohybu, tak sa tlačidlá prepnú. Keďže pri podobnej situácii nastáva nepríjemné klikanie relé.

4.4 Automatizácie

Po vyriešení základnej logiky a skontrolovania základnej funkčnosti ovládania spojenej s kalkulačkou a vyriešením problému synchronizácie ovládania sme boli schopní pridať automatizácie zo zadania a rozšíriť ich o ďalšie užitočné automatizácie a funkcie.

¹Odkaz na repozitár s kalkulačkou: <https://github.com/XKNX/xknx>

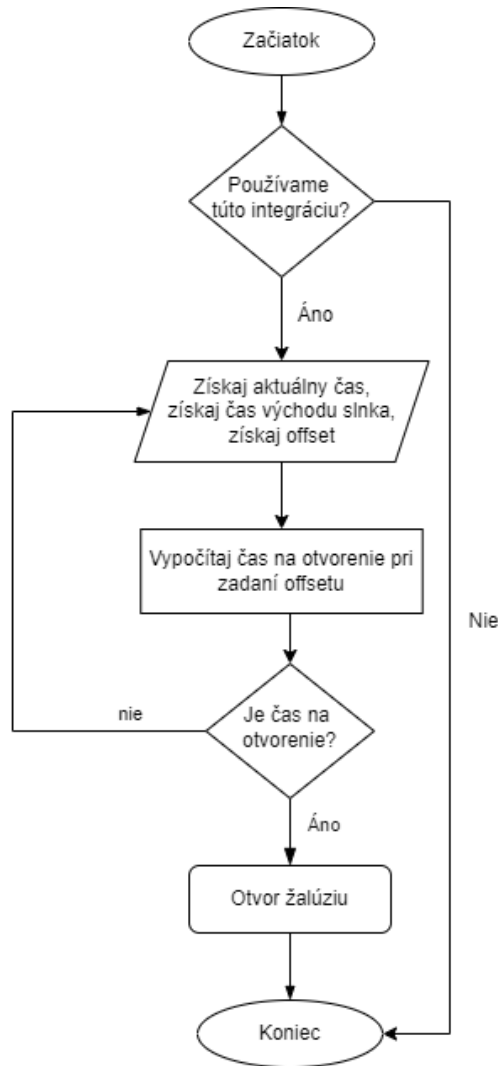
Prvá z riešených automatizácií predstavovala zatváranie a otváranie žalúzií na základe času. To je riešené tak, že získame aktuálny čas z Home Assistantu, ktorý sa bude porovnávať s časom, ktorý používateľ zadal vo formulári buď na otváranie, alebo zatváranie. Ak sa bude čas zhodovať napríklad s časom, na otvorenie žalúzie, tak sa spustí funkcia na otvorenie pomocou príkazu `await self.async_open_cover()`. Presne takto to funguje aj v opačnom prípade pre zatvorenie, no spustí sa funkcia na zatvorenie pomocou príkazu `await self.async_close_cover()`, taktiež ukázané na obrázku 4.6.



Obr. 4.6: Vývojový diagram automatizácie na zatvorenie žalúzie pri určenom čase.

Na podobnom princípe funguje aj otváranie a zatváranie žalúzií v prípade východu a západu slnka. Z Home Assistantu získame čas a taktiež údaje o východe a západe slnka, ktoré Home Assistant automaticky získa z Met.no. Keďže používateľ môže nastaviť offset na čas buď východu alebo západu slnka, vypočítame čas, kedy sa má žalúzia reálne otvoriť alebo zatvoriť, a ďalej to funguje obdobne ako pri ovládaní pomocou času. Ak sa čas bude zhodovať napríklad s časom na otvorenie žalúzie, spustí sa funkcia na otvorenie pomocou príkazu `await self.async_open_cover()`. Presne takto to funguje aj v opačnom prípade pre zatvorenie, no spustí sa funkcia na zatvorenie pomocou príkazu `await self.async_close_cover()`, ukázané na obrázku 4.7.

Ďalšia automatizácia spočíva v tom, že v prípade, že je noc a zapneme vnútorné osvetlenie, automaticky sa zatvoria žalúzie. Tento proces je dosiahnutý pomocou získania údajov o východe a západe slnka z Home Assistantu a určením, že slnko už nie je na oblohe, čo znamená, že nastala noc. Ak sa potom zistí, že je noc a následne sa zaznamená zmena

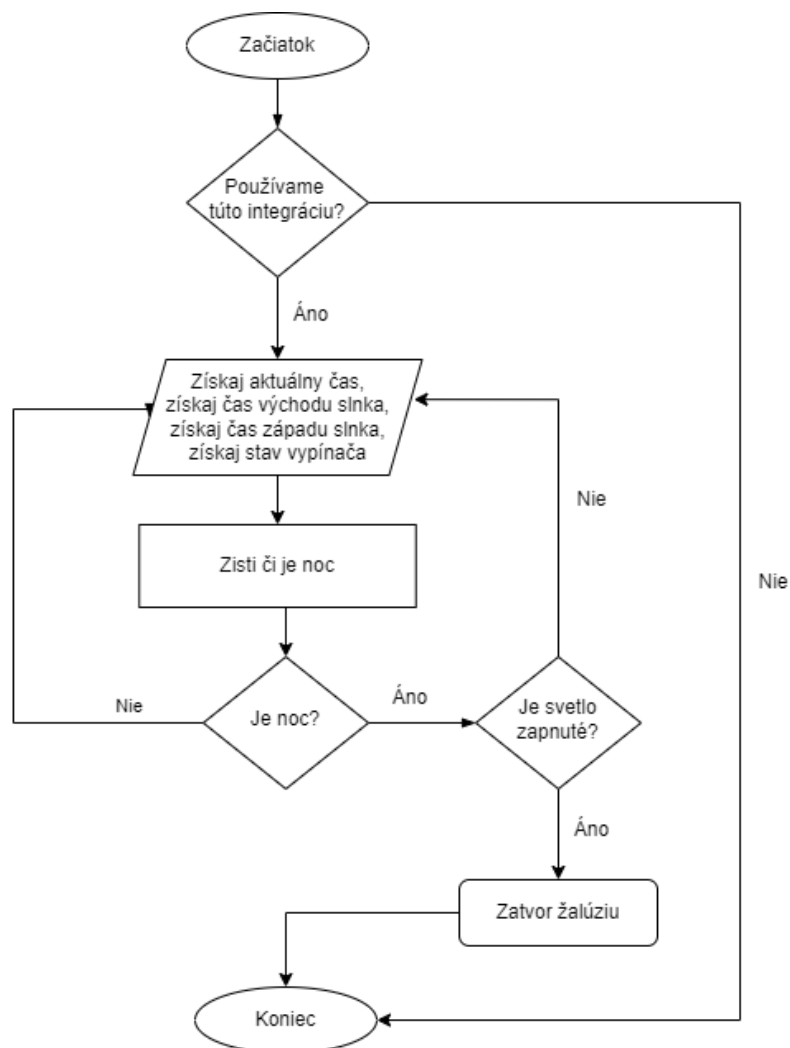


Obr. 4.7: Vývojový diagram automatizácie na otvorenie žalúzie pri východe slnka.

stavu zvoleného vnútorného osvetlenia alebo je už dané svetlo zapnuté, automaticky sa zatvoria žalúzie pomocou príkazu `await self.async_close_cover()`, taktiež ukázané na obrázku 4.8. V prípade, že sa žalúzia pohybuje pri vykonávaní tohto príkazu, tak sa spustenie pozastaví a príkaz sa spustí až po zastavení žalúzie.

Automatizácia, ktorá sa stará o maximálny prechod slnečných lúčov do miestnosti, funguje na podobnom princípe ako automatizácia, ktorá zatvára žalúzie v noci. Akurát nezávisť pomocou údajov z Home Assistantu, kedy je noc, ale zistí, či je deň. Ak sa používateľ rozhodne túto automatizáciu používať, každých 10 minút sa skontroluje, či je otvorené tieň, a ak nie, a je deň, tak ho otvorí pomocou príkazu `await self.async_open_cover_tilt()`.

Poslednou automatizáciou je možnosť nastavenia ochrany žalúzií v prípade silného vetra alebo nepriaznivého počasia. Na získanie údajov o rýchlosti vetra a upozornení na nepriaznivé počasia sa stará API od OpenMeteo. Toto API funguje jednoducho a stačí mu len zemepisná šírka a výška. Následne, ak je požiadavka správna, dostaneme v odpovedi aktuálnu rýchlosť vetra a upozornenie pre celý deň vo formáte WMO kódu. Ak je možnosť ochrany pred nepriaznivým počasím zaškrtnutá, tak sa každých 15 minút kontroluje rých-



Obr. 4.8: Vývojový diagram automatizácie na zatvorenie žalúzie pri zapnutí svetla v noci.

losť vetra a WMO kód pre celý deň. Ak jedna z týchto hodnôt nevyhovuje, tak sa žalúzie zatvoria pomocou príkazu `await self.async_close_cover()`.

4.5 Rozšírenia

Jedno rozšírenie sa týka ochrany žalúzií. Používateľ, ktorý používa službu Netatmo a má ju správne nakonfigurovanú v Home Assistannte, tak si bude môcť nastaviť dve dôležité veci, ktoré umožnia ochranu žalúzií ešte lepšie. Ide o rýchlosť vetra a rýchlosť nárazového vetra, ktorý je taktiež dôležitý, keďže ten zväčša zafúkne a poškodí žalúziu, lebo fúka väčšou silou. Ďalšia z výhod Netatmo je to, že ako si ju používateľ nastavuje, ide o viac lokálnu službu než API od OpenMeteo. A to kvôli tomu, že môže vlastniť vlastnú meteostanicu alebo sa dokáže prihlasovať na stanice od ostatných používateľov, ktorí svoju stanicu verejne zdieľajú. Jednou z ďalších výhod oproti API od OpenMeteo je, že OpenMeteo má obmedzený počet volaní API za deň.

Rozšírením zadania je možnosť vypnutia vypínania spínača pri dokončení pohybu. Táto funkcia umožňuje užívateľom, ktorí nepoužívajú vzájomne prepojené relé, zamedziť nežiadúcemu spusteniu skriptu Stop po dosiahnutí koncových pozícií. To môže byť prospešné pre rôzne používateľské prípady, kde nie je potrebné spúšťať skript Stop po dosiahnutí koncových pozícií. Čiže predvolená hodnota je nastavená na True. Ak je nastavená na True, skript sa spustí po tom, ako žalúzia dosiahne hodnoty 0 alebo 100 (čo znamená úplné zavretie alebo úplné otvorenie). To je vhodné pre ľudí, ktorí používajú vzájomne prepojené relé, ktoré musia byť uvoľnené, keď sa žalúzie dostanú na koncové pozície.

Kapitola 5

Testovanie

V tejto sekcii sa zameriame na testovanie implementácie našej integrácie do systému Home Assistant. Prvá časť popisuje testovacie prostredie, ktoré sme použili na overenie funkčnosti nášho riešenia. Následne sa pozrieme na výsledky testovania, kde analyzujeme identifikované chyby a opatrenia na ich odstránenie. Ďalej sa venujeme šíreniu nášho riešenia v komunite používateľov a spätnú väzbu, ktorú sme získali od užívateľov po zdieľaní nášho projektu. Tento prehľad umožní lepšie porozumieť procesu testovania, a taktiež popíše reakcie komunity na našu integráciu.

5.1 Testovacie prostredie

Testovacie prostredie bolo zostavené pomocou vypínača Sonoff, na ktorý sme nahrávali vlastný firmvér pomocou ESPHome. Tento vypínač poskytol platformu na testovanie našej integrácie. Spolu s tým sme využili Raspberry Pi ako hostiteľský systém, na ktorom sme nasadili operačný systém Home Assistant, poskytujúci prostredie pre naše testy v domácich podmienkach. Tento prístup nám umožnil simulovať skutočné prostredie, v ktorom by sa naša integrácia mala v budúcnosti nasadiť. S vypínačom Sonoff sme mohli simulovať ovládanie žalúzií a sledovať ich správne fungovanie v rôznych situáciách.

Ďalej sme identifikovali a pripravili rôzne testovacie scenáre, ktoré pokrývali rôzne aspekty integrácie, aby sme zabezpečili systematické preverenie všetkých kľúčových funkcií a vlastností, ktoré od nášho riešenia požadujeme. Po dokončení testov sme zhodnotili výsledky, identifikovali prípadné nedostatky a navrhli možné vylepšenia. Následne sme riešenie zdieľali a hodnotili spätnú väzbu od používateľov.

Testovacie scenáre:

1. Overenie zobrazovania a ukladania údajov:
 - Kontrola, či sa formulár správne zobrazuje, a či sa údaje z neho správne ukladajú do systému.
2. Testovanie úpravy údajov z formulára:
 - Overenie, či je možné správne upravovať uložené údaje, a či sa tieto úpravy správne aktualizujú v systéme.
3. Testovanie funkcionality integrácie s vypínačmi a žalúziami:

- Testovanie, či integrácia úspešne prepojí údaje z formulára s hlavným skriptom, a či je možné správne ovládať vypínače, ktoré riadia žalúzie.
4. Testovanie kombinácií príkazov:
 - Skúsiť rôzne rýchle kombinácie zadania príkazov alebo zadanie viacerých príkazov naraz, aby sa zistilo, či systém správne reaguje a vyhodnocuje tieto príkazy. Taktiež testovanie rôznych okrajových podmienok ako napríklad používanie integrácie pri zadaní veľkých hodnôt alebo naopak malých hodnôt pre čas.
 5. Testovanie časovania:
 - Overenie, či časovanie pri zadávaní rôznych príkazov funguje správne a či sa žalúzie správne zastavia po dosiahnutí maximálnej alebo minimálnej polohy.
 6. Testovanie reakcie na externé príkazy:
 - Testovanie, či skript správne reaguje na príkazy, ktoré prichádzajú mimo integrácie, a či tieto príkazy správne vyhodnocuje.
 7. Testovanie automatizácií:
 - Overenie, či automatizácie, ktoré sú aktívne fungujú správne.
 8. Testovanie dlhodobej prevádzky:
 - Overenie, či integrácia vykazuje chyby počas dlhodobej prevádzky počas niekoľkých po sebe nasledujúcich dní.
 9. Testovanie výpadku systému:
 - Overenie, či hodnoty, s ktorými integrácia pracuje, sa uložia a v prípade reštartu alebo výpadku systému Home Assistant tieto hodnoty integrácia znovu načíta.

5.1.1 Výsledky testovania

Počas nášho testovania sme určili niekoľko chýb, ktoré sme museli preskúmať a opraviť, aby sme zabezpečili bezchybný chod nášho riešenia. Jednou z týchto chýb bolo nesprávne spracovanie údajov zo vstupného formulára, čo viedlo k nesprávnemu fungovaniu hlavného skriptu. Tento problém sme riešili analýzou zdrojového kódu a následnou úpravou logiky spracovania údajov. Implementovali sme kontrolu vstupných dát a zabezpečili sme, že všetky údaje zo vstupného formulára sú korektne interpretované a spracované hlavným skriptom.

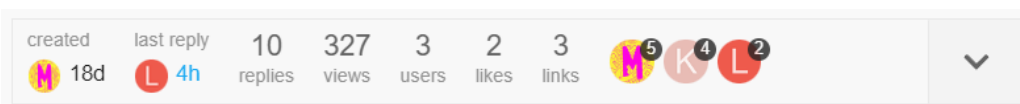
Ďalšou identifikovanou chybou bola nesprávna reakcia na príkazy, najmä tie, ktoré prichádzali mimo našej integrácie. Tento problém sme úspešne vyriešili implementáciou sledovania zmeny stavov, čo nám umožnilo reagovať na externé aj interné príkazy a správne ich vyhodnocovať v našom riešení integrácie.

Posledná identifikovaná chyba sa vyskytla po dlhodobom používaní a prejavila sa v nedokončení pohybu hlavnej pozície pri používaní pohyblivého tlačidla. Chyba spočívala v tom, že kalkulácia hlavnej pozície nedosiahla svoj cieľ. Tento problém sme vyriešili úpravou podmienky, ktorá určuje konečnú pozíciu hlavnej pozície, čím sme zabezpečili jej správne fungovanie aj pri dlhšom časovom používaní.

Po úspešnom identifikovaní a opravení všetkých chýb sme opätovne testovali naše riešenie a zistili sme, že funguje spoľahlivo a bezchybne. V tomto momente sme boli spokojní s dosiahnutými výsledkami a boli sme pripravení zdieľať náš projekt s komunitou.

5.2 Šírenie v komunite používateľov

Keďže sme si uvedomili, že najlepšiu spätnú väzbu môžeme získať od používateľov, ktorí možno ešte len hľadajú vhodné riešenia pre ovládanie svojich žalúzií a zároveň možno ešte nepoužívajú platformu Home Assistant, rozhodli sme sa aktívne zdieľať naše riešenie¹ prostredníctvom vhodných komunikačných kanálov, ukážka príspevku v prílohe B. Ako vhodné kanály sa ukázali oficiálne fórum podpory komunity², komunitný obchod, Discord kanály a kanál Home Assistant (aktivita viď obrázok 5.1) na Reddit-e³. Všetky naše príspevky a zdieľania sme realizovali v anglickom jazyku, aby sme mohli osloviť čo najširšiu komunitu používateľov Home Assistantu. Snažili sme sa stručne a jasne popísať podstatu nášho riešenia a poskytnúť používateľom jednoduchý návod na jeho používanie s ich inštanciou Home Assistantu. Týmto spôsobom sme sa snažili maximalizovať dostupnosť a použiteľnosť nášho riešenia pre čo najväčší počet používateľov. Zdieľanie týmto spôsobom nám pomohlo nazbierať aj potrebnú spätnú väzbu, ktorú sme od zdieľania očakávali.



Obr. 5.1: Aktivita príspevku na komunitnom fóre Home Assitantu k dňu 7. mája 2024.

5.2.1 Pridanie do komunitného obchodu

Jednou z vecí, ktorá by sa dala považovať aj za rozšírenie, je pridanie do komunitného obchodu. Tento krok sa nachádza v kapitole o testovaní, pretože v čase písania práce bolo možné integráciu pridať len ako vlastnú integráciu, no ešte sa čakalo na schválenie od komunitných administrátorov, ktorí by našu žiadosť o pridanie integrácie do komunitného obchodu schválili viď obrázok 5.2. Postup bol zdĺhavejší, než sa očakávalo, ale neočakávajú sa žiadne problémy počas schvaľovania. Pridanie našej integrácie do komunitného obchodu by malo priniesť viaceré výhody. Jednou z nich je zjednodušenie procesu stiahnutia, aktualizácie a správy integrácie v Home Assistente. Ďalšou výhodou je zvýšenie dostupnosti integrácie pre širšiu komunitu používateľov, čo by mohlo viesť k ďalšiemu používaniu nášho riešenia. Týmto spôsobom by sme mohli poskytnúť užívateľom jednoduchší a pohodlnejší spôsob, ako integrovať naše riešenie.

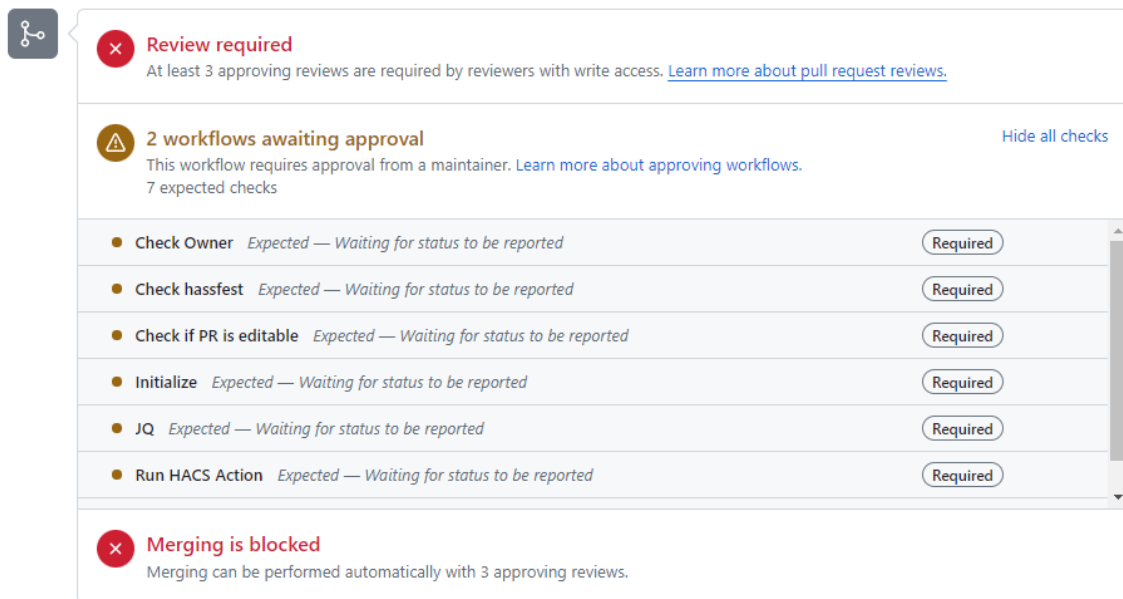
5.2.2 Spätná väzba

Spätná väzba bola pozitívna. Používateľom sa páčilo vylepšenie oproti ostatným dostupným riešeniam, hlavne synchronizácia ovládania, vylepšené počítanie pozície nakláňania a zjednodušený postup konfigurácie žalúzií. Za výhodu boli označené aj automatizácie, ktoré chránia žalúzie pred nepriaznivým počasím a zabezpečujú ich ochranu v rôznych podmienkach. Užívatelia si cenili aj možnosť ďalšieho prispôsobenia a rozšírenia systému pomocou

¹Odkaz na naše riešenie na platforme GitHub: https://github.com/MatthewOnTour/BUT_blinds_time_control

²Odkaz na naše riešenie na fóre podpory: <https://community.home-assistant.io/t/home-assistant-blinds-control-custom-integration/718612>

³Odkaz na naše riešenie na Reddit-e: https://www.reddit.com/r/homeassistant/comments/1cb20oz/home_assistant_blinds_control_custom_integration/

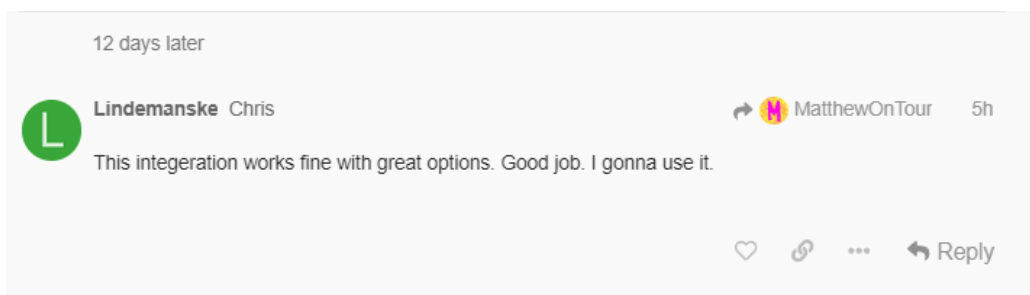


Obr. 5.2: Požiadavka na pull, kritéria na schválenie.

rôznych doplnkov a modulov, ktoré umožňujú flexibilné nastavenie a rozšírenie jeho funkcií, ale aj jednoduchú úpravu prvotnej konfigurácie.

Našli sa však aj námietky na zlepšenie. Najviac požadovanou funkciou bola možnosť pridania tretieho tlačidla, ktoré vydáva príkaz stoj. Okrem toho, užívatelia vyjadrili záujem o ďalšie možnosti personalizácie užívateľského rozhrania a zlepšenie podpory pre viacero jazykových verzí, ako aj možnosť zatvorenia žalúzií pri daždi s využitím možnosti Netatmo. Tieto námietky sú však vnímané ako príležitosť na ďalšie zlepšenie a rozšírenie funkčnosti integrácie v budúcnosti.

Jednému konkrétnemu používateľovi⁴ naše riešenie funguje a používa ho v svojej inteligentnej domácnosti. Konkrétne používa relé od spoločnosti Shelly. Aj keď mal na začiatku niekoľko problémov s nasadením do svojej inteligentnej domácnosti, tieto problémy sa ukázali byť nezávislé od nášho riešenia a podarilo sa mu ich úspešne vyriešiť. Teraz je plne spokojný s našou integráciou a využíva ju. Druhý používateľ tiež naše riešenie používa a je s ním spokojný vid' obrázok 5.3.



Obr. 5.3: Kladná spätná väzba od používateľa.

⁴Odkaz na príspevok používateľa ktorý naše riešenie používa: <https://community.home-assistant.io/t/is-it-possible-to-manually-calibrate-a-cover/720254>

Kapitola 6

Záver

Témou bakalárskej práce bolo naštudovanie a následná implementácia riešenia pre časovo ovládané žalúzie a automatizáciu s nimi spojenú. Na začiatku sme potrebovali zistiť, aké riešenia už existujú, aké sú ich nevýhody, ktoré technológie sa budú dať použiť, porovnať ich a vybrať tie, ktoré budú najvhodnejšie.

Po analýze sme identifikovali viaceré problémy, ktoré sa vyskytovali v súčasných riešeniach časového ovládania žalúzií. Medzi ne patrili: absencia možnosti nakláňania, nesprávna kalkulácia hlavnej pozície ako aj pozície tienenia, ak to riešenie obsahovalo, komplikovaná konfigurácia a následná úprava konfigurácie. Tieto nedostatky sme si stanovili za prioritné oblasti na zlepšenie. Je dobré spomenúť, že žiadne z týchto riešení neobsahovalo žiadne automatizácie, čo je ďalšia výhoda nášho riešenia.

Návrh a implementácia nového riešenia pre časovo ovládané žalúzie a ich automatizáciu boli vykonané po preskúmaní dostupných možností, ktoré nám ponúkal Home Assistant alebo používateľa Home Assistantu. Ďalej sme prešli do fázy testovania, ktorá zahŕňala overenie správnosti kalkulácie hlavnej pozície, ako aj pozície nakláňania. Testovali sme aj synchronizáciu ovládania a rôzne vstupy, ako sú neočakávané vstupy na konci pozícií alebo stlačenie viacerých príkazov naraz a podobne, ktoré by mohli používatelia potenciálne zadávať omylom alebo naschvál.

Po úspešnom testovaní sme náš projekt zdieľali na rôznych komunikačných kanáloch, ako sú oficiálne fóra, rôzne Reddit komunity alebo Discord, kde sa stretávajú používatelia Home Assistantu. Vďaka spätnej väzbe od komunity sme identifikovali ďalšie oblasti na zlepšenie a uskutočnili sme potrebné úpravy a opravy.

Do budúcnosti plánujeme sa ďalej venovať tomuto projektu a aktualizovať ho, aby sme mohli uspokojiť potreby rastúcej užívateľskej základne. S nádejou, ak bude schválený prístup na komunitný obchod, očakávame zvýšený záujem a prírastok nových používateľov. Týmto sa proces inštalácie a aktualizácie stane jednoduchší a plynulejší. Sme celkovo spokojní s vývojom a fungovaním nášho riešenia. Avšak, uvedomujeme si, že stále existuje priestor na ďalšie zlepšenia a inovácie, ktoré by mohli posilniť jeho efektivitu a použiteľnosť.

Literatúra

- [1] BLUETOOTH. *The only full-stack standard for wireless lighting control* online. 2023. Dostupné z: <https://www.bluetooth.com/>. [cit. 2023-12-20].
- [2] CARVALHO, M. *Building Smart Home Automation Solutions with Home Assistant*. 1. vyd. PACKT, 2023. ISBN 978-1-80181-529-1.
- [3] ESPHOME. *ESPHome* online. 2023. Dostupné z: <https://esphome.io/index.html#>. [cit. 2023-11-18].
- [4] HOME ASSISTANT. *Home Assistant FAQ* online. 2023. Dostupné z: <https://www.home-assistant.io/faq/>. [cit. 2023-02-24].
- [5] HACS. *Home Assistant Community Store* online. 2023. Dostupné z: <https://hacs.xyz/>. [cit. 2023-12-18].
- [6] HOME ASSISTANT. *Home Assistant Developer Documentation* online. 2023. Dostupné z: <https://developers.home-assistant.io/>. [cit. 2023-11-17].
- [7] HOME ASSISTANT. *Home Assistant Documentation* online. 2023. Dostupné z: <https://www.home-assistant.io/docs/>. [cit. 2023-11-17].
- [8] SINHA, S. *State of IoT 2023* online. 2023. Dostupné z: <https://iot-analytics.com/number-connected-iot-devices/>. [cit. 2023-02-24].
- [9] MAKER. *Understanding the Basics of Infrared Communications* online. 2023. Dostupné z: <https://www.digikey.com/en/maker/tutorials/2021/understanding-the-basics-of-infrared-communications>. [cit. 2023-12-20].
- [10] *Push Button Switches* online. 2023. Dostupné z: <https://www.iqsdirectory.com/articles/electric-switch/push-button-switches.html>. [cit. 2023-12-22].
- [11] PYTHON. *The Python Tutorial* online. 2023. Dostupné z: <https://docs.python.org/3/tutorial/index.html>. [cit. 2023-12-23].
- [12] RF WIRELESS WORLD. *IR vs RF* online. 2023. Dostupné z: <https://www.rfwireless-world.com/Terminology/Difference-between-IR-remote-and-RF-remote.html>. [cit. 2023-12-21].
- [13] SCHWARTZ, M. *Internet of Things with ESP8266*. 1. vyd. PACKT, 2016. ISBN 978-1-78646-802-4.

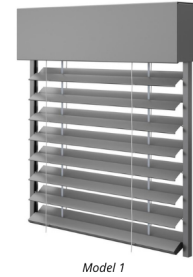
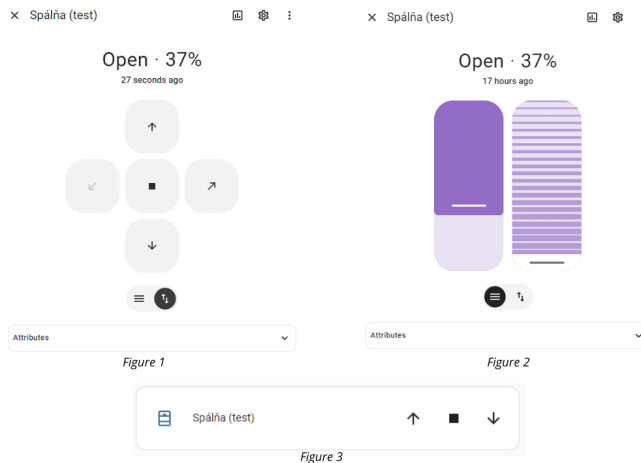
- [14] SONOFF. *TX Series Wi-Fi Smart Wall Touch Switches* online. 2023. Dostupné z: <https://sonoff.tech/product/smart-wall-switches/tx-series/>. [cit. 2023-12-23].
- [15] CISCO. *What Is Wi-Fi?* online. 2023. Dostupné z: <https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html>. [cit. 2023-12-18].
- [16] SPARKES, M. *What does Wi-Fi stand for?* online. 2023. Dostupné z: <https://www.newscientist.com/question/what-does-wi-fi-stand-for/>. [cit. 2023-12-19].
- [17] OREN BEN-KIKI, CLARK EVANS, INGY DÖT NET. *YAML Ain't Markup Language* online. 2021. Dostupné z: <https://yaml.org/spec/1.2.2/>. [cit. 2023-11-18].

Príloha A

Plagát

SMART COVER CONTROL MODULE FOR HOME ASSISTANT

Controlling



- For blinds that don't provide position feedback
- Controlling only via commands UP, DOWN, and STOP
- Easy controlling within the Home Assistant
- Supports tilting
- Open Source

Simple installation and configuration

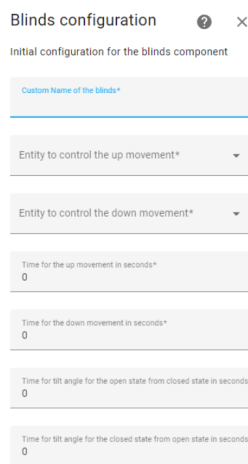


Figure 5

Adding through

- Convenient installation via the community store
- Easy configuration during initial setup
- Pre-loaded with useful automation

Synchronized control

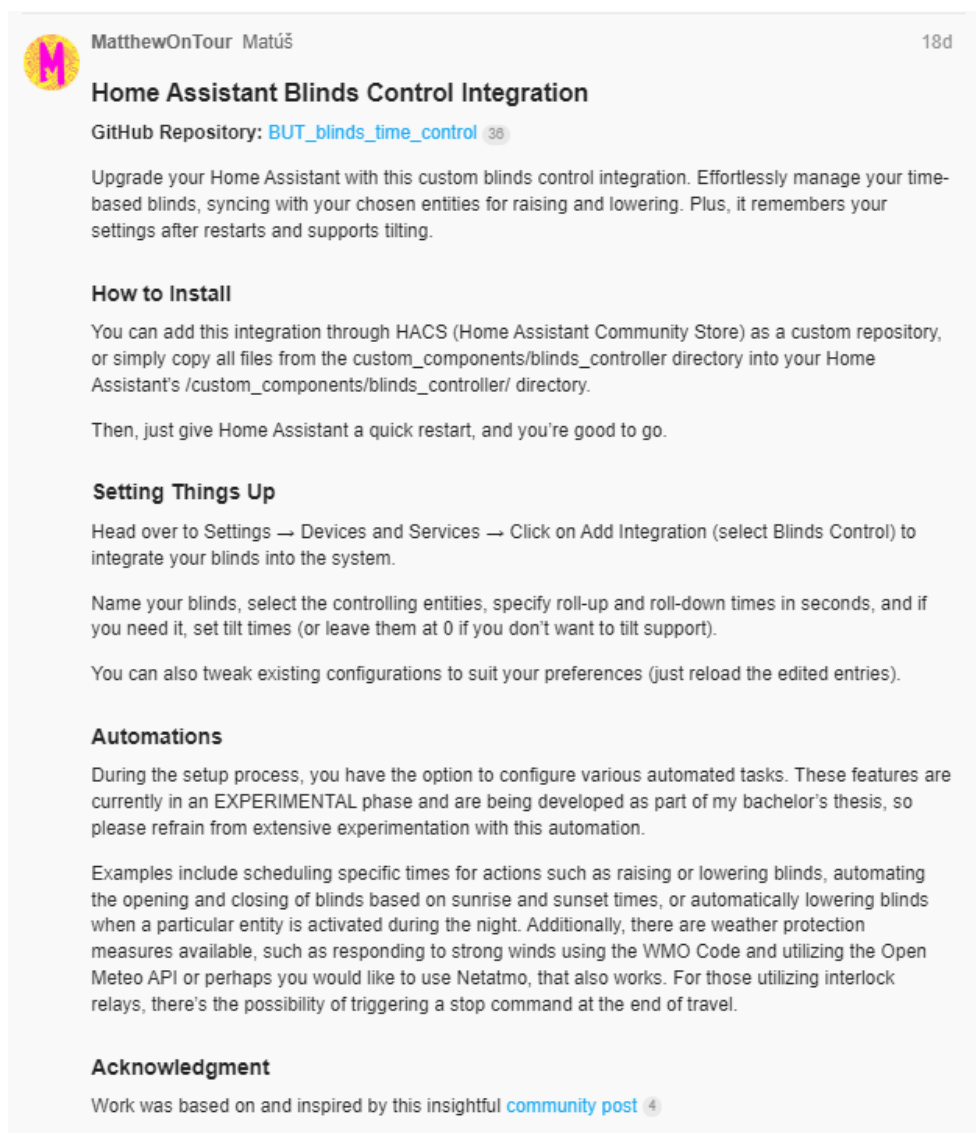


Figure 6

- Control via Home Assistant or physical switches
- Maintains states through restarts

Príloha B

Príspevky na platformách



MatthewOnTour Matúš 18d

Home Assistant Blinds Control Integration

GitHub Repository: [BUT_blinds_time_control](#) 36

Upgrade your Home Assistant with this custom blinds control integration. Effortlessly manage your time-based blinds, syncing with your chosen entities for raising and lowering. Plus, it remembers your settings after restarts and supports tilting.

How to Install

You can add this integration through HACS (Home Assistant Community Store) as a custom repository, or simply copy all files from the `custom_components/blinds_controller` directory into your Home Assistant's `/custom_components/blinds_controller/` directory.

Then, just give Home Assistant a quick restart, and you're good to go.

Setting Things Up

Head over to Settings → Devices and Services → Click on Add Integration (select Blinds Control) to integrate your blinds into the system.

Name your blinds, select the controlling entities, specify roll-up and roll-down times in seconds, and if you need it, set tilt times (or leave them at 0 if you don't want to tilt support).

You can also tweak existing configurations to suit your preferences (just reload the edited entries).

Automations

During the setup process, you have the option to configure various automated tasks. These features are currently in an EXPERIMENTAL phase and are being developed as part of my bachelor's thesis, so please refrain from extensive experimentation with this automation.

Examples include scheduling specific times for actions such as raising or lowering blinds, automating the opening and closing of blinds based on sunrise and sunset times, or automatically lowering blinds when a particular entity is activated during the night. Additionally, there are weather protection measures available, such as responding to strong winds using the WMO Code and utilizing the Open Meteo API or perhaps you would like to use Netatmo, that also works. For those utilizing interlock relays, there's the possibility of triggering a stop command at the end of travel.

Acknowledgment

Work was based on and inspired by this insightful [community post](#) 4

Obr. B.1: Príspevok na platforme komunitného fóra.

Home Assistant Blinds Control Custom Integration

News

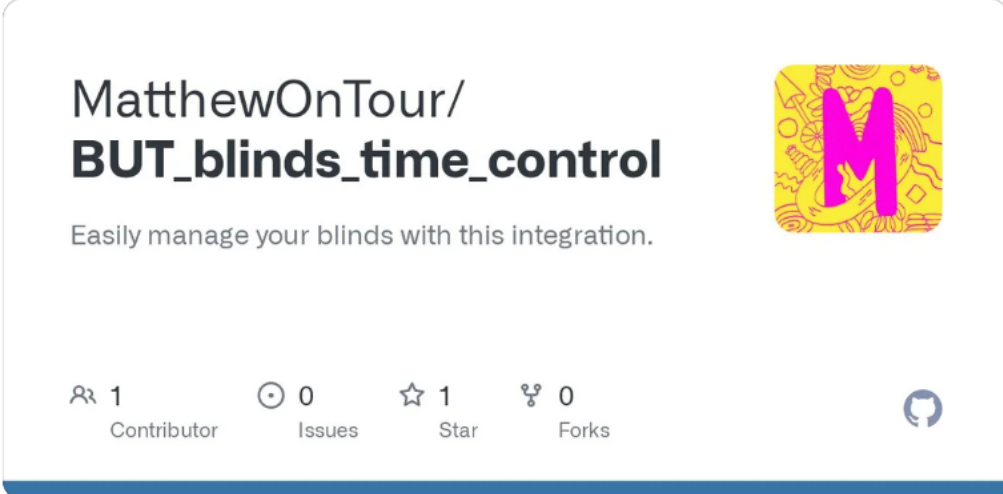
Hey there! 🙌 I've just created something that could make your Home Assistant experience even better, especially if you're into smart blinds or covers!

Introducing the Home Assistant Blinds Control Integration! 🏠 🛠️

This integration lets you effortlessly manage your time-based blinds, syncing them with your favorite entities for smooth operation. And also, it remembers your settings even after restarts! Plus, tilt support is included, covering all your needs.

Ready to try it out? Head over to my GitHub repository: [BUT_blinds_time_control](#), or check it out on the Home Assistant Community Forum [here](#).

Give it a whirl and let me know what you think! 🏠 🙌



The screenshot shows a GitHub repository page for 'MatthewOnTour/BUT_blinds_time_control'. The repository name is displayed in large, bold black text. To the right is a yellow square profile picture with a pink 'M' and circuit-like patterns. Below the name is the description 'Easily manage your blinds with this integration.' At the bottom, there are statistics: 1 Contributor, 0 Issues, 1 Star, and 0 Forks. A GitHub logo is in the bottom right corner.

MatthewOnTour/
BUT_blinds_time_control

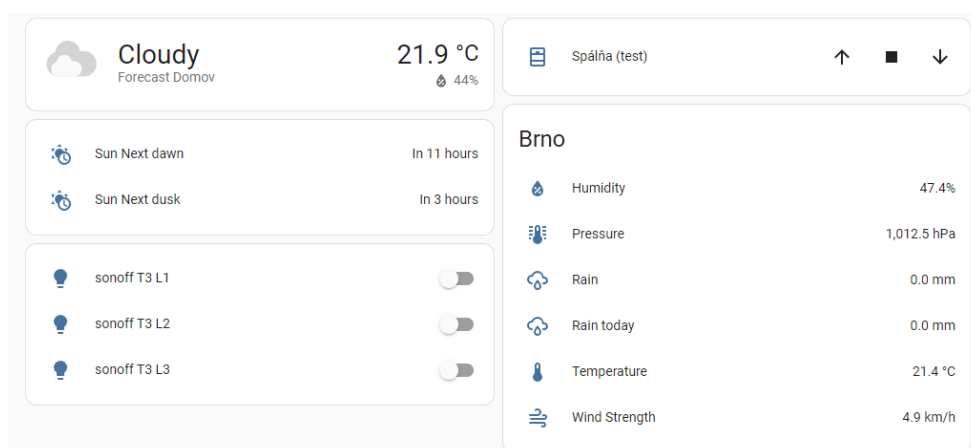
Easily manage your blinds with this integration.

1 Contributor 0 Issues 1 Star 0 Forks

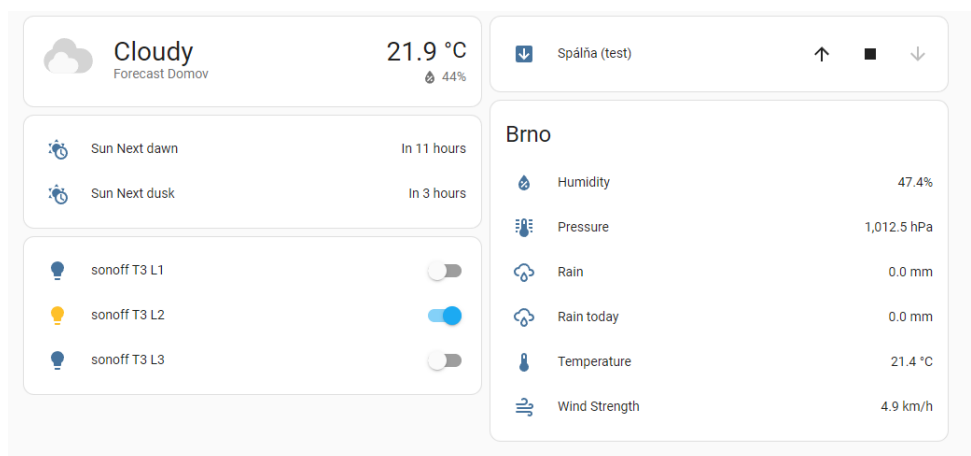
Obr. B.2: Príspevok na platforme Reddit.

Príloha C

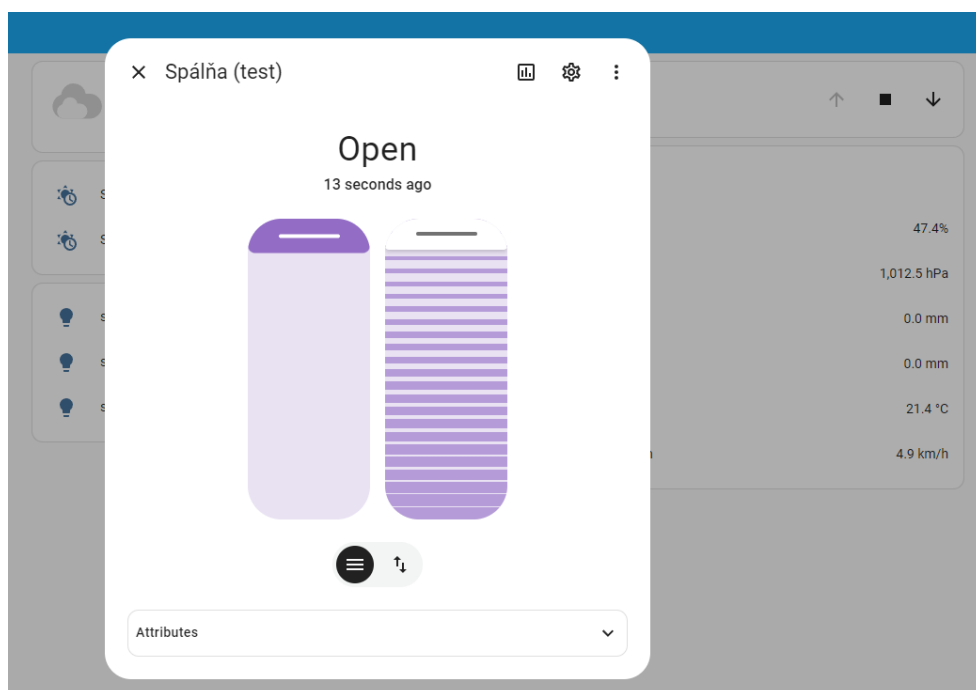
Používanie riešenia



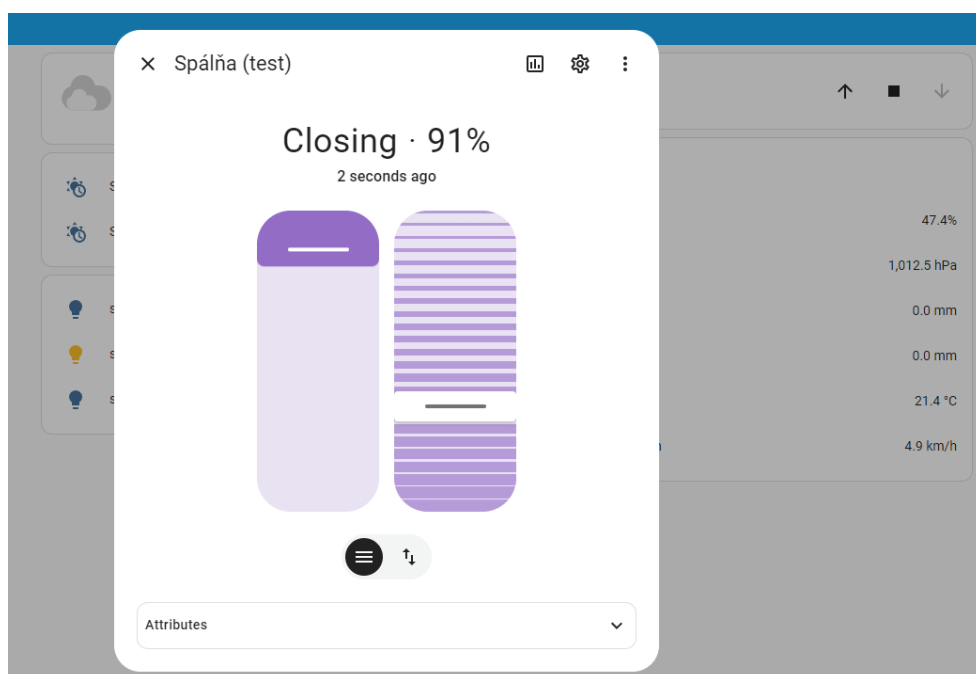
Obr. C.1: Bez príkazu, čaká sa na príkaz.



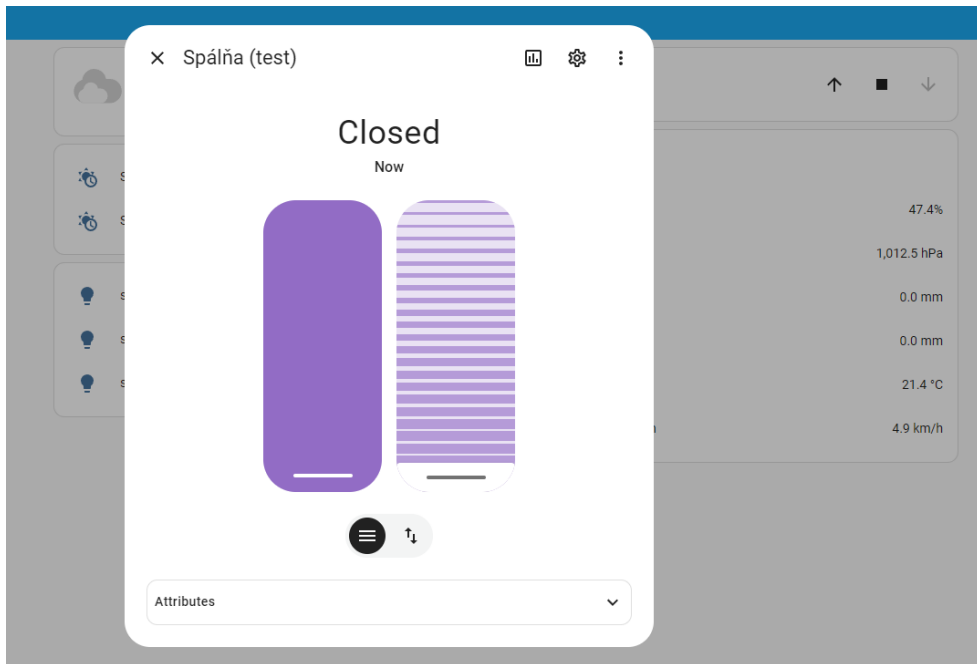
Obr. C.2: Príkaz smeruj dole, ovláda sonoff T3 L2.



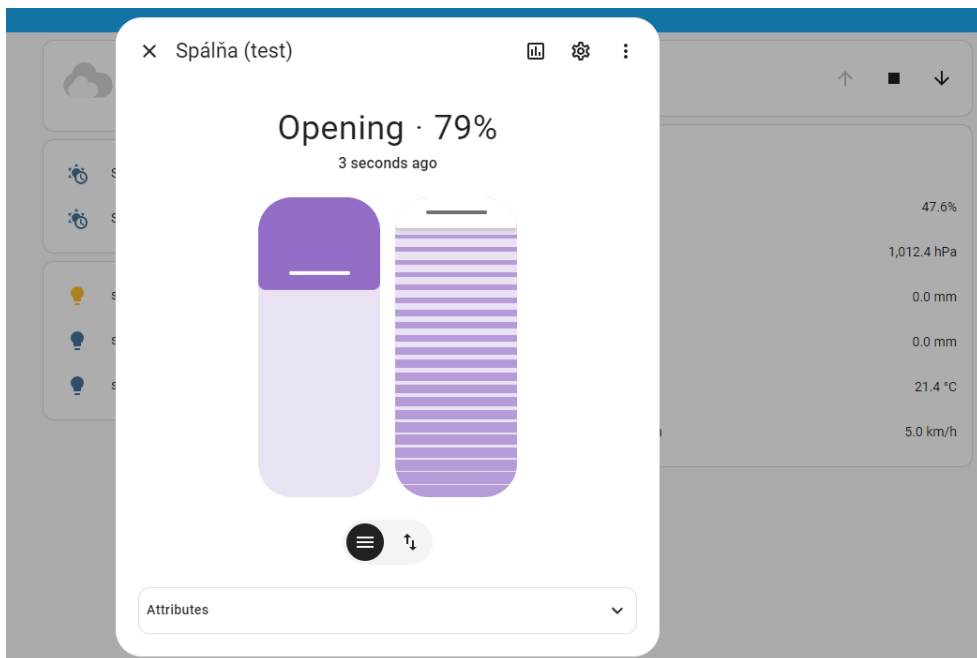
Obr. C.3: Posuvné tlačidlá, čaká na príkaz.



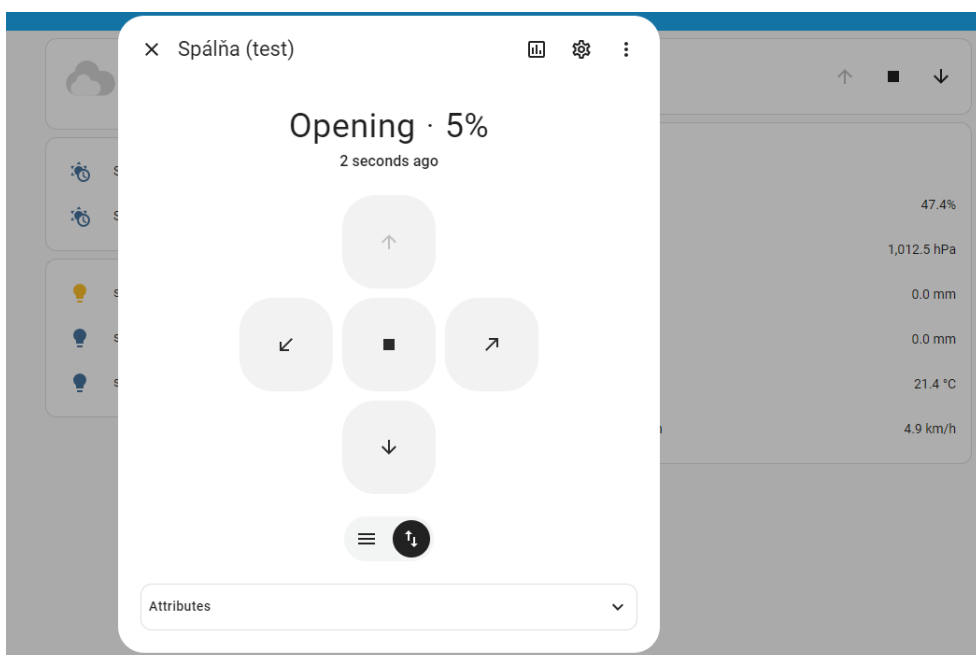
Obr. C.4: Posuvné tlačidlá, zatvarajú na určitú pozíciu alebo do prerušenia, ovláda sonoff T3 L2.



Obr. C.5: Posuvné tlačidlá, konečná pozícia, čaká na príkaz.



Obr. C.6: Posuvné tlačidlá, otvárajú na určitú pozíciu alebo do prerušenia, ovláda sonoff T3 L1.



Obr. C.7: Tlačidlá ovládania, otvárajú do prerušenia, ovláda sonoff T3 L1.

Príloha D

Obsah priloženého pamäťového média

/.....	Médium
├── text/.....	Bakalárska práca
│ └── latex/.....	LaTeX— zdrojové kódy tohto dokumentu
├── src/.....	Zdrojové kódy
├── plagat/.....	Plagát z konferencie Excel
├── video/.....	Video ukážky
│ ├── video1.mp4/.....	zobrazuje pridanie integrácie pomocou HACS
│ ├── video2.mp4/ ...	zobrazuje prvotnú konfiguráciu, zmenu a zmazanie konfigurácie
│ ├── video3.mp4/.....	zobrazuje pridanie prvku na hlavný panel
│ ├── video4.mp4/.....	zobrazuje ovládanie s tienením
│ ├── video5.mp4/.....	zobrazuje ovládanie bez tienenia
│ └── video6.mp4/....	zobrazuje ovládanie pomocou zariadenia mimo našu integráciu