

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta

Ústav aplikované informatiky



System pro řízení chytré domácnosti

Bakalářská práce

Luděk Scholz

Školitel: Mgr. Jakub Geyer

České Budějovice 2021

Bibliografické údaje

Scholz, Luděk, 2021: Systém pro řízení chytré domácnosti. [Smarthome control system, Bc. Thesis, in Czech] – 77 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato bakalářská práce řeší problematiku chytré domácnosti v nasazení jako opensource systém s kladeným důrazem na nízké pořizovací náklady. V práci jsou sestrojeny tři vlastní zařízení, která jsou nasazena v reálném provozu, a to chytrá hlavice, měřič spotřeby a detektor zatopení sklepa. Práce řeší problematiku vývoje vlastních zařízení s otestováním a porovnáním oproti komerčním zařízením. Mimo jiné řeší práce i komunikaci lokální instalace chytré domácnosti a serveru. V závěru je nastíněno, jakým směrem se dále celý projekt bude vyvíjet.

Klíčová slova

Chytrá domácnost, wifi, ESP, Opensource, nízkonákladové řešení, vzdálený přístup, Arduino, chytrá hlavice, měření spotřeby, 3D tisk.

Annotation

This bachelor's thesis addresses the issue of smart home in deployment as an open source system with an emphasis on low budget. In this thesis, three devices are created from scratch, namely a smart thermostatic valve, a power consumption meter and a flood detector. All of them are already deployed. This thesis describes problems in development of own devices with testing and comparison with commercial equipment. Among other things, it solves the communication of the local installation of the smart home and the server. In conclusion, it outlined what direction to further the project will develop.

Keywords

Smarthome, wifi, ESP, Opensource, low-budget, remote control, arduino, smart thermostatic valve , power consumption meter, 3D printing.

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

Místo, datum.

V Českých Budějovicích, dne 29.4.2021

Podpis studenta

Luděk Scholz

Handwritten signature of Luděk Scholz in black ink, consisting of stylized initials and a surname.

Poděkování

Tímto bych rád poděkoval svému školiteli Mgr. Jakubu Geyerovi za odborné vedení a směřování práce, za cenné rady k tématu, možnost tisku 3D modelů a také za vždy velmi rychlé reakce na mé dotazy. Dále bych rád poděkoval své rodině za trpělivost a podporu.

Obsah

1	Úvod	1
2	Cíle práce	2
2.1	Základní funkční a nefunkční požadavky	3
3	Definice některých pojmů	4
4	Chytrá domácnost	5
4.1	Lokální systémy	5
4.1.1	Home Assistant	5
4.1.2	OpenHAB	6
4.1.3	Domoticz	6
4.1.4	Vlastní řešení výrobců	7
4.2	Cloudové systémy	7
4.2.1	ThingSpeak	7
4.2.2	Nabucasa (Home Assistant Cloud)	8
4.2.3	Vlastní řešení výrobců	8
5	Home Assistant	9
5.1	Analýza	9
5.1.1	Virtuální prostředí na domácím počítači	9
5.1.2	Využití staršího počítače nebo notebooku	9
5.1.3	Využití domácího serveru	10
5.1.4	Mikropočítač	10
5.1.5	Home Assistant Blue	11
5.2	Instalace na Raspberry Pi	11
5.3	Konfigurace Home Assistantu	12
5.4	Důležité pojmy Home Assistantu	12
5.4.1	Pojmy spojené s fungováním systému	12
5.4.2	Pojmy spojené s grafickým rozhraním	13

6	Nástavba pro komunikaci mimo vnitřní síť	15
6.1	Analýza a návrh.....	16
6.1.1	Ovládání Home Assistantu	16
6.1.2	Komunikace mezi serverem	18
6.1.3	Serverová část.....	18
6.2	Implementace.....	19
6.2.1	Lokální middleware	20
6.2.2	MQTT Broker.....	20
6.2.3	Aplikace třetí strany	21
7	Integrace zakoupených plug-n-play zařízení	22
7.1	Instalace prerekvizit.....	22
7.2	Instalace chytrých zařízení.....	24
7.3	Použití chytrých zařízení	24
8	Vlastní nízkonákladové zařízení	26
8.1	Všeobecná analýza HW komunikačních technologií.....	26
8.1.1	Wi-Fi	26
8.1.2	Bluetooth.....	27
8.1.3	Z-Wave	27
8.1.4	LoRa	28
8.1.5	Ethernet.....	28
8.1.6	Zhodnocení	29
8.2	Hardware prostředky.....	30
8.2.1	Kontroléry.....	30
8.2.2	Senzory a čidla	33
8.2.3	USB programátory	33
8.3	Potřebné programy	33
8.3.1	Arduino IDE.....	33
8.3.2	Python IDE	34
8.3.3	Autodesk Fusion 360.....	34

8.3.4	Prusa Slicer.....	34
8.3.5	Autodesk Eagle	35
9	Chytrá hlavice	36
9.1	Analýza.....	36
9.1.1	Mechanická část.....	36
9.1.2	Technologie komunikace a mikrokontroler	39
9.1.3	Napájení hlavice.....	40
9.1.4	Software.....	41
9.2	Teoretické řešení – Návrh systému	41
9.2.1	Mechanická část – uchycení a převodovka s výtlačným členem.....	41
9.2.2	Elektronická část – řízení motoru a detekce pozice.....	43
9.2.3	Elektronická část – Mikrokontroler a senzory	44
9.3	Implementace – Finální řešení	45
10	Měřič spotřeby tepelného čerpadla	48
10.1	Analýza.....	48
10.1.1	Sonda proudu	48
10.1.2	Mikrokontroler	49
10.1.3	Software.....	49
10.2	Teoretické řešení – Návrh systému	50
10.2.1	Elektronika měření	50
10.2.2	Elektronika Arduina a ESP.....	51
10.2.3	Softwarová část	51
10.3	Implementace – Finální řešení	52
11	Detektor zatopení sklepa	55
11.1	Analýza a návrh.....	55
11.1.1	Detekce zatopení.....	55
11.1.2	Mikrokontroler	56
11.1.3	Napájení.....	56

11.1.4	Software.....	57
11.2	Implementace.....	57
12	Testování a ladění	59
12.1	Chytrá hlavice.....	59
12.1.1	Testování a porovnání funkčnosti.....	59
12.1.2	Testování mechanické odolnosti.....	60
12.1.3	Porovnání spotřeby.....	61
12.1.4	Cenové porovnání.....	62
12.2	Měřič spotřeby.....	62
12.2.1	Testování a porovnání funkčnosti.....	63
12.2.2	Cenové porovnání.....	64
12.3	Detektor zatopení sklepa.....	64
12.3.1	Testování funkčnosti a odolnosti.....	65
12.3.2	Test spotřeby.....	65
12.3.3	Cena.....	66
12.4	Všeobecné porovnání.....	67
12.4.1	Porovnání univerzálnosti.....	67
12.4.2	Licence/možnost přizpůsobení (open-source).....	67
12.4.3	Náročnost instalace / tvorby.....	68
13	Závěr	69
	Vysvětlení zkratk	71
	Seznam obrázků	72
	Seznam tabulek	73
	Seznam zdrojů	74
	Seznam příloh	77

1 Úvod

V dnešní době lze pozorovat stále více se rozvíjející trend využívání chytrých zařízení v oblastech, kde před několika lety integrace s chytrým zařízením nebyla vůbec myslitelná. Tento trend se přesunul i do oblasti řízení domácnosti. Barevné LED Žárovky, pračky, topení, nebo alarm, to vše lze v dnešní době integrovat do chytré domácnosti. Využívání chytrých zařízení v domácnosti přináší mnoho výhod, od komfortu až po úsporu nákladů díky chytré optimalizaci. Využití těchto chytrých zařízení ovšem není dostupné pro každého. Pořizovací náklady běžného jednoduchého zařízení, jako je ventil na topení, se ve variantě chytrého ventilu dostávají na desetinásobky původní ceny. V konečném důsledku, při nasazování chytré domácnosti, se můžeme dostat na částky v řádu desetitisíců, v případě nasazení kompletního systému i na statisíce.¹

Chytrou domácnost si ovšem mohou vytvořit i lidé bez masivních investic. Existuje řada open-source softwarových řešení pro ovládání chytrých zařízení, jako například OpenHAB, Domoticz, nebo Home Assistant, na který navazuje tato práce. Práce vytváří novou možnost vzdáleného ovládání pomocí vlastního middleware, ale také přináší novou alternativu k chytrým zařízením pro domácnost. Základem této alternativy jsou vlastní open-source chytrá zařízení, s kladeným důrazem na nízké pořizovací náklady a užitnou hodnotu. Díky tomu lze přinést komfort chytré domácnosti i k dalším lidem. Na nová nízkonákladová zařízení jde do budoucna navázat a vytvořit celý ekosystém založený na použité technologii.

¹ Pro příklad si můžeme uvést modelovou realizaci na webu <https://www.inteligentni-bydleni.cz/cena-systemu> .

2 Cíle práce

Cílem práce je vytvořit a následně v reálné domácnosti nasadit, cenově dostupný open source ekosystém chytré domácnosti. Vzhledem k vysokým cenovým nárokům běžných chytrých zařízení, je cílem sestavit vlastní chytrá zařízení s kladeným důrazem na nízké náklady. Při tvorbě zařízení se počítá s využitím běžně dostupných elektronických a mechanických součástí, jako například mikrokontrolery, motory, nebo spojovací materiál. Dále se počítá s využitím 3D tisku pro tvorbu mechanických částí, jako jsou převody nebo krytky jednotlivých zařízení. Tvorba zařízení zahrnuje také tvorbu potřebného softwaru jak na straně zařízení, tak i potřebnou konfiguraci na straně centrálního softwaru řídicího chytrou domácnost. Součástí tohoto cíle práce je také otestování a porovnání vlastních zařízení oproti již existujícím komerčním řešením na základě technologie Z-Wave.

Dalším cílem práce je navázat na předešlou práci v projektu Smart Grid, konkrétně v řešení vzdáleného přístupu k softwaru vzdálené domácnosti, za účelem sběru dat a možnosti ovládat chytrou domácnost mimo lokální domácí síť. Současné řešení působí funkčně pouze jako proof-of-concept, tedy v reálném nasazení skrývá několik nevýhod, které lze dohledat přímo v dokumentaci projektu. Hlavním problémem je nemožnost přidávat nová zařízení do vzdáleného přístupu bez přístupu k lokální síti. Cílem práce je vytvořit vlastní modul ve formě služby, který bude nevýhody předešlé metody překonávat, tedy umožňovat ovládat a získávat data z více domácností, automaticky aktualizovat architekturu chytré domácnosti, a také umožní multiplatformní připojení dalších serverů, jako například do budoucna databázový server pro ukládání dat, nebo server vyhodnocující nasbíraná data.

2.1 Základní funkční a nefunkční požadavky²

Nejprve je nutné definovat základní funkční a nefunkční požadavky systému jako celku, tedy jaké vlastnosti musí systém mít, a jaké funkce systém musí umět vykonávat.

Funkční požadavky systému	Nefunkční požadavky systému
Ovládání dostupných zařízení	Nízká energetická náročnost
Logování a zobrazování dat	Nízké pořizovací náklady a dostupnost dílů
Vzdálený přístup	Záloha systému v případě výpadku proudu
	Dostupnost off-line

Tabulka 1 - Funkční a nefunkční požadavky.

² Detailnější funkční a nefunkční požadavky jsou uvedeny zvlášť v dalších kapitolách, upřesňující konkrétní prvky chytré domácnosti.

3 Definice některých pojmů

Pro správné pochopení práce je potřeba si nejprve definovat pojmy související s celým tématem.

Prvním pojmem je *domácnost*, respektive z toho vycházející pojem chytrá domácnost. V této práci je domácnost chápána jako fyzické místo, jako je například byt, dům, nebo chalupa. Pojem *chytrá domácnost*, jak již z názvu vyplývá, je například byt, nebo dům, vybavený zařízeními, které umožňují chytré ovládání, automatizace a analýzy, ulehčující správu domácnosti, případně umožňující energetické, či časové úspory. [1]

Pojem *systém* lze chápat pouze jako určitou část, například software, který se stará o řízení externích zařízení, nebo i jako komplexní systém skládající se z vlastních hardwarových prvků, jejich firmwaru a v neposlední řadě softwaru shromažďující data a umožňující grafický přístup uživatelům, v tomto případě se hodí spíše název *ekosystém*.

Pojem *Internet of Things*, známý pod zkratkou *IoT*, je chápán jako síť jednotlivých zařízení založená na standardizovaných protokolech, umožňující komunikaci mezi zařízeními a výměnu informací. V praxi, v chytré domácnosti, se jedná například o chytrá čidla, chytré žárovky, nebo i komplexnější zařízení jako jsou pračka, myčka, nebo lednička. Mimo chytrou domácnost se může jednat o čidla sloužící pro výzkum přírody, čidla v dopravě, nebo v průmyslovém užití. [1]

V práci se počítá s využitím *3D tisku* pro mechanické části nízkonákladových zařízení. Pojem 3D tisk znamená proces, kdy se z nefyzické předlohy, 3D modelu vytvořeném na počítači, vytvoří pomocí tisku reálný fyzický předmět (díl). [2]

4 Chytrá domácnost

Jak již bylo v předchozí kapitole řečeno, chytrá domácnost je domácnost využívající chytrá zařízení. Pro správné a efektivní využití je potřeba mít centrální systém který celou domácnost bude řídit. V následujících kapitolách se seznámíme s vybranými možnostmi, jak domácnost lze řídit.

4.1 Lokální systémy

Jako první varianta se nabízí lokální řešení. Tedy systém, který funguje v lokální síti. Obvykle se jedná o řídicí jednotku od výrobce, připojenou na síť a ovládající ostatní zařízení za pomoci datové komunikace (bezdrátové technologie). Lokální systémy však mohou být i opensource systémy běžící na počítači nebo mikropočítači jako je Raspberry Pi. Výhodou lokálních systémů je jejich funkčnost i při výpadku internetu.

4.1.1 Home Assistant

Jako první systém si představíme Home Assistant. Home Assistant se nabízí v několika variantách, ať se jedná o kontejnerovou aplikaci až po samostatný linuxový operační systém s Home Assistantem. Samotný Home Assistant je software napsaný v Pythonu. [3]

Vzhledem k obrovské a stále rostoucí komunitě je tento systém stále aktualizovaný. Obrovskou výhodou je způsob integrace jednotlivých zařízení, kdy Home Assistant obsahuje stovky integrací pro ovládání téměř jakéhokoliv chytrého zařízení. [3]

Velkou výhodou Home Assistantu je možnost instalace pluginů. Tedy možnost instalace dodatečných balíčků, nebo aplikací. To umožňuje jednoduše pro uživatele nainstalovat prakticky jakoukoliv dodatečnou funkčnost. Ať už se jedná o novou databázi, ssh terminal, nebo vizualizaci historie.

Další výhodou je automatická detekce zařízení, kdy Home Assistant je schopen v síti detekovat všechna chytrá zařízení, a díky tomu automaticky nastavit všechny integrace. Díky tomu je instalace plug-n-play zařízení do Home Assistantu otázkou několika kliknutí. [4]

Největší výhodou pro uživatele je rozhraní lovelace. Lovelace je vizuální rozhraní pro monitorování systému. Ovšem lovelace je také rozhraní pro samostatnou tvorbu uživatelského rozhraní. Snadno lze přidávat grafy, tlačítka, předpověď počasí a mnoho dalšího, bez nutnosti znalosti jakéhokoliv programovacího jazyka.

V případě, kdy uživatel potřebuje přidat nové zařízení, například vlastnoručně vytvořený senzor, lze vytvořit vlastní integraci, nebo využít volnou integraci MQTT. V MQTT integracích nalezneme prakticky všechna zařízení, která můžeme potřebovat. Ať už se jedná o termostat, senzor, nebo ovládání stmívání žárovky. [7]

Další výhodou tohoto systému je možnost pro uživatele jednoduše nastavitelných automatizací. Home Assistant obsahuje jednoduchý vizuální editor na základě kroků spouštěč – podmínka – akce.

Všechny ovládací prvky lovelace, všechny integrace, i všechny automatizace lze psát ručně v konfiguračních souborech. [7]

4.1.2 OpenHAB

Druhou možností je systém OpenHAB. OpenHAB je velmi podobný systém jako Home Assistant. Je to software napsaný v jazyku JAVA, díky tomu ho lze nainstalovat na veliké množství zařízení. Kompatibilní je s Windows, MacOS, Linux, Raspberry Pi, nebo i Synology. Jedná se o starší systém, než je Home Assistant. Aktuální verze vychází z verze z roku 2010. Stejně jako Home Assistant i OpenHAB podporuje rozšíření pomocí modulů. Ačkoliv starší verze systému OpenHAB neumožňovaly pokročilé nastavení pomocí grafického rozhraní, novější verze je již srovnatelná s uživatelským prostředím Home Assistantu. [4]

4.1.3 Domoticz

Dalším systémem je Domoticz. Stejně jako předchozí systémy, i tento je opensource. Je napsaný v C++ s kompatibilitou na většině zařízení. Díky vlastní komunitě na GitHubu je Domoticz aktualizovaný v častých intervalech. [5]

Stejně jako Home Assistant i Domoticz obsahuje vlastní grafické rozhraní, umožňující správu systému. Díky tomu uživatel nemusí pro přidání nového zařízení psát příkazy, nebo konfigurace ručně. Na rozdíl od Home Assistantu však Domoticz neobsahuje automatickou

detekci pro přidání nových zařízení. Proto je třeba každé zařízení přidat manuálně, například pomocí grafického rozhraní.

Samotný Domoticz, vzhledem ke složitosti kódu v C++ není od začátku zamýšlený jako modulární systém jako je Home Assistant. Postupem času však vznikl od komunity python framework pro integraci a instalaci vlastních modulů. [5]

4.1.4 Vlastní řešení výrobců

Kromě opensource systémů, existují také uzavřené systémy. Tyto systémy bývají vlastní řešení výrobců chytrých zařízení. Buďto se jedná o samostatnou aplikaci na chytrém telefonu, díky které se lze připojit na konkrétní zařízení a tím ho ovládat, nebo je potřeba mít HUB od výrobce, který komunikuje s dalšími chytrými zařízeními. Tento Hub poté lze ovládat v lokální síti, ale obsahuje také napojení na servery výrobce umožňující i ovládání mimo síť. Jako příklad si můžeme uvést Phillips Hue Bridge, Woon Home nebo i chytrá zařízení od obchodního řetězce Lidl.

4.2 Cloudové systémy

Jako druhé řešení je ovládání chytrých zařízení pomocí cloudu. To má několik výhod, ale i nevýhod. Výhodou je, že u některých systémů nemusíme mít žádnou centrální jednotku, protože chytrá zařízení komunikují přes síť přímo se serverem. To však přináší i nevýhodu ve formě nemožnosti ovládat chytrou domácnost v momentě výpadku internetu.

4.2.1 ThingSpeak

Prvním cloudovým řešením je ThingSpeak. Jedná se o velmi rozšířenou službu, umožňující sběr dat ze senzorů za pomoci protokolu HTTP nebo MQTT. Velikou výhodou tohoto systému je možnost data přímo vizualizovat a analyzovat za pomoci MATLABu, se kterým je ThingSpeak propojen. Samozřejmostí je možnost reagovat na podněty ze senzorů.

ThingSpeak dokáže integrovat mnoho různých zařízení. Ať už se bavíme o vlastních senzorech připojených například k Raspberry Pi, nebo vlastních zařízeních vytvořených za pomoci Mikrokontrolerů, wifi zařízení, ale také i LoRa zařízení. [8]

4.2.2 Nabucasa (Home Assistant Cloud)

Nabucasa, nebo také Home Assistant Cloud, je cloudové řešení vyvinuté vývojáři Home Assistantu. Jedná se o cloudovou nástavbu pro ovládání lokální instance Home Assistantu. Nejedná se tedy pouze o cloudové řešení, ale spíše o cloudový doplněk k lokálnímu řešení.

Samotná služba je placená, a to formou měsíčního předplatného v ceně 5 dolarů. Služba nabízí 31denní verzi zdarma pro vyzkoušení.

Instalace je velmi jednoduchá. Po zaregistrování na webu Nabucasa, se stačí ve vlastní instanci Home Assistantu v nastavení, v záložce Home Assistant Cloud přihlásit pomocí emailu a hesla. Poté Nabucasa automaticky vygeneruje vlastní subdoménu, pomocí které lze přistupovat Home Assistantu z jakékoliv sítě. [9]

4.2.3 Vlastní řešení výrobců

Jak již bylo zmíněno v předešlé kapitole 4.1.4, většina výrobců mají svá uzavřená řešení. A to i za pomoci cloudu. Díky tomu lze ovládat chytrá zařízení i za pomoci asistentu Google, nebo Alexa.

5 Home Assistant

V této kapitole se zaměříme na výběr vhodného zařízení pro provoz Home Assistantu, výběr vhodného instalačního balíku a na samotnou instalaci a konfiguraci systému. Poté si představíme základní moduly a fungování samotného systému.

5.1 Analýza

Home Assistant je systém, který vyžaduje počítač, na kterém může být spuštěn. Způsob provozu Home Assistantu lze rozdělit do čtyř skupin které byly vyzkoušené. Samostatnou skupinou je zařízení Home Assistant Blue. [7]

5.1.1 Virtuální prostředí na domácím počítači

První, nejméně nákladné na zprovoznění, je využití virtuálního prostředí na domácím počítači. To lze vytvořit pomocí virtualizačních nástrojů jako je VirtualBox, VMware, nebo na Linuxu libvirt. Toto řešení má výhody v jednoduchosti a rychlosti instalace, avšak velkou nevýhodou je nutnost stálého spuštění hostitelského počítače. Pro správné fungování Home Assistantu by to tedy znamenalo mít zapnutý domácí počítač 24 hodin, 365 dní v roce, což je energeticky nákladné, a tedy z hlediska provozu nevýhodné. Toto řešení se však hodí na testování konfigurací, vyzkoušení a seznámení se s prostředím Home Assistantu.

5.1.2 Využití staršího počítače nebo notebooku

Druhá možnost je instalace na vlastní zařízení bez virtuálního prostředí. To je možné až už na starší počítač nebo na starší notebook. Avšak stejně jako u předešlé virtuální instalace na domácím PC, je i tato metoda energeticky náročná, a tedy zbytečně neefektivní.

5.1.3 Využití domácího serveru

Třetí možností je využití jiného zařízení, které již v domácnosti běží 24 hodin denně. To může být například webový server který v objektu je. Avšak toto řešení není pro běžnou domácnost relevantní, z důvodu, že takové zařízení nebývá součástí většiny domácností.

5.1.4 Mikropočítač

Čtvrtá možnost, je využití minipočítače pro běh Home Assistantu. Jako nejpoužívanější řešení se zde jeví Raspberry Pi. Tento mikropočítač má narozdíl od běžného počítače zanedbatelnou spotřebu, a tedy z hlediska dlouhodobého provozu se jedná o nejlepší řešení. Spotřeby jsou porovnané v tabulce níže. [6]

<i>Zařízení</i>	<i>Spotřeba za hodinu</i>	<i>Spotřeba za den</i>	<i>Cena provozu za den³</i>
<i>Výkonný počítač</i>	> 300Wh	> 7200Wh	> 42 Kč
<i>Domácí počítač</i>	200Wh	4800Wh	27 Kč
<i>Starší notebook</i>	60Wh	1440Wh	9 Kč
<i>Raspberry Pi</i>	6,5Wh	156Wh	1 Kč

Tabulka 2 - Analýza spotřeby.

Na výběr máme hned z několika verzí Raspberry Pi. Od nejstarší verze 1, přes miniaturní verzi Raspberry Pi Zero, až k nejnovější verzi 4. Tyto verze se liší velikostí RAM, rychlostí procesoru, počtem jader, ale také i dostupnými periferiemi. Raspberry Pi verze 1 a miniaturní verze Zero, vychází ze stejného chipsetu s jedno-jádrovým procesorem s taktem do 1Ghz a 512 MB dostupné RAM. Toto není pro Home Assistant pro běžné pohodlné používání dostatečné. Avšak pro testování je stále k dispozici instalační balík pro tento chipset. Lepším, ale stále pomalejším řešením je Raspberry Pi 2, případně nejlepším řešením je Raspberry Pi 3 nebo 4. Zde je pořizovací cena vyšší, ale z hlediska komfortu využívání je rozdíl zanedbatelný. Pro potřebu práce byla vyzkoušena instalace na všechna dostupná zařízení kromě Raspberry Pi Zero. Pro finální instalaci bylo zvoleno Raspberry Pi 4. [10]

³ Výpočet při ceně 5.78 Kč za 1 kWh

5.1.5 Home Assistant Blue

Home Assistant Blue je vlastní řešení na základě mikropočítače ODROID-N2+. Jedná se o plug-n-play řešení, kdy odpadá nutnost vlastní instalace systému, řešení napájení, zakrytí HW a podobně. Toto zařízení je v době psaní této práce limitovaná edice, proto je možné, že v budoucnu již nebude dostupné. [11]

5.2 Instalace na Raspberry Pi

Pro instalaci a samotné fungování je třeba mít dostatečně velkou a rychlou SD kartu. Pro systém se doporučuje jako minimum 32 GB SD karta. SD karta by měla být s co nejlepší deklarovanou životností, kvůli opakovaným zápisům od Home Assistantu. Jako ideální řešení jsou uváděny karty s aplikační třídou A2. Tyto karty deklarují vyšší životnost při větším počtu IO operací.

Po vybrání vhodné SD karty je potřeba vybrat vhodný instalační obraz pro konkrétní zařízení. Odkazy na jednotlivé obrazy pro verze Raspberry Pi jsou součástí přílohy. Vždy aktuální verzi Home Assistantu lze stáhnout na stránce “<https://www.home-assistant.io/installation/raspberrypi>” Tato stránka již neuvádí verze pro starší Raspberry Pi, než je verze 3. Adresy pro stažení starších instalačních balíčků jsou součástí přílohy.

Pro nahrání systému na SD kartu byl využit program s názvem “*balenaEtcher*” s použitím možnosti zdroje z url, poté byl vybrán jako cíl SD karta a spuštěn proces. Po nahrání instalátoru proběhla jeho kontrola a následně bylo možné vyjmout SD kartu z PC a vložit kartu do Raspberry Pi. Pro instalaci je potřeba internet. Samotná instalace probíhá bez nutnosti zásahu uživatele. Rychlost instalace je závislá na užitém HW. U prvních modelů může trvat instalace až hodinu, naopak u nejnovějšího Raspberry Pi 4 trvá instalace v řádu minut. Po úspěšné instalaci lze najít rozhraní na adrese <http://IP:8123>, kde IP je IP adresa Raspberry Pi. [7]

5.3 Konfigurace Home Assistantu

Po instalaci je nutné vytvořit nového uživatele, tento uživatel bude správce celého systému. Je potřeba vyplnit jméno, uživatelské jméno a bezpečné heslo. Následující krok je pojmenování instance Home Assistantu, to může být například “Domov”, a výběr geolokace domácnosti, výběr časového pásma a také zvolení systému jednotek. Po tomto kroku Home Assistant zkouší vyhledat chytrá zařízení v síti a nabídne jejich automatickou integraci. Po tomto kroku je již Home Assistant připravený k použití. [7]

5.4 Důležité pojmy Home Assistantu

Pro pochopení práce s Home Assistantem je potřeba vysvětlit několik základních pojmů. Vzhledem k struktuře celého projektu a rapidnímu vývoji, se může prostředí zdát pro začátečníka velmi zmatečné. Vysvětlením základních pojmů získá uživatel alespoň základní přehled. Vysvětlení pojmů dělím na dvě části, pojmy týkající se funkčnosti celého systému a poté pojmy týkající se grafického rozhraní

5.4.1 Pojmy spojené s fungováním systému

Prvním pojmem je **Integrace**, ten lze chápat jako určitý ovladač pro zařízení. Každý druh zařízení, ať už se jedná o chytrou žárovku, pračku, nebo router, musí mít v Home Assistantu integraci. Každá integrace má na webu Home Assistantu návod, jak zařízení instalovat, a to buďto automaticky, nebo i manuálně pomocí konfigurace.

Druhým pojmem je **Entita**, což by se dalo chápat jako jedno konkrétní zařízení. Entita je konkrétní chytré zařízení využívající nějakou integraci v Home Assistantu. Entita může být konkrétní chytrá žárovka v kuchyni a druhá samostatná entita může být žárovka v kanceláři.

Na entitu navazuje pojem **Zařízení**, což zastupuje připojené chytré zařízení. Jako zařízení se v Home Assistantu objevují obvykle větší samostatná zařízení, jako například Smart TV, mobilní telefon s aplikací, či multimediální centrum. Jedno zařízení může obsahovat několik entit.

Dalším pojmem je *Automatizace*. Jedná se o funkci umožňující vykonávat jakékoliv akce s chytrou domácností automaticky. Díky tomu lze nastavit pravidla, aby se vykonala určitá akce v určitém případě. Například lze nastavit automatizaci pro vypnutí topení, pokud nikdo není doma, nebo zapnutí světla při západu slunce.

Dalším pojmem je *Scéna*. Ve významu Home Assistantu se jedná o určitý předdefinovaný stav entit. Lze si tedy předdefinovat různé scény, například pro náladu v pracovně, či ložnici úpravou barevnosti světla.

Pro vylepšení funkčnosti lze použít *Doplňky*, nebo také *add-ons*. Jedná se o softwarové doplňky, které lze do systému doinstalovat. Tyto doplňky rozšiřují funkčnost, mohou přidávat nové moduly i nové nástroje. Jako příklad může být rozšíření s modulem MQTT Brokerem, nebo například doplněk pro úpravu souborů, Visual Studio Code, díky kterému lze přímo z prohlížeče upravovat konfigurace systému.

Jak již bylo zmíněno, Home Assistant lze konfigurovat manuálně. Na to slouží soubory s koncovkou *.yaml* ve složce config. V následující tabulce lze vyčíst konkrétní funkčnosti konfiguračních souborů. [3, 7]

Soubor	Funkce
Configuration.yaml	Konfigurace entit
Automation.yaml	Konfigurace automatizací
Scripts.yaml	Konfigurace skriptů
Secrets.yaml	Skrytý soubor pro ukládání hesel a tokenů
Scenes.yaml	Manuální konfigurace scén

Tabulka 3 - Konfigurační soubory.

5.4.2 Pojmy spojené s grafickým rozhraním

Prvním pojmem je *Lovelace*, jedná se o název pro celé grafické rozhraní Home Assistantu. Jde o dynamické rozhraní pro konfiguraci, i pro uživatelskou kontrolu chytré domácnosti.

Pro vytvoření stránky obsahující data ze senzorů, je třeba vytvořit takzvaný *Ovládací panel* nebo také *Dashboard*. Ovládací panel je uskupení všech grafických ovládacích prvků chytré domácnosti.

Rozšířením pro dashboard je takzvaný **Pohled**, nebo také **View**. Jedná se o složky umožňující rozdělit dashboard na více částí, jako například na ovládání topení, grafy teplot, nebo informace o spotřebě elektrické energie.

Pro vlastní zobrazování informací, nebo ovládání entit, slouží takzvané **Karty**. Karta je modul grafického rozhraní umožňující přidat ovládání entit, umožňuje zobrazovat stav entit, jako třeba teplotu, ale také umožňuje vizualizaci pomocí grafů. Home Assistant obsahuje veliké množství vlastních karet, ale je zde také možnost doinstalovat nové karty s ještě pokročilejšími funkcemi. [3, 7]



Obrázek 1 - Příklad vizualizace dat.

6 Nástavba pro komunikaci mimo vnitřní síť

Jak již bylo v kapitole 4.1.1 řečeno, Home Assistant obsahuje vlastní grafické rozhraní pro ovládání. Toto rozhraní umožňuje ovládání i správu celého systému na lokální síti. Bohužel toto rozhraní není dostupné v případě nutnosti ovládat systém mimo lokální síť. Zde se nabízí jako řešení hned několik variant.

Veřejná IP adresa

Jako první varianta přichází v úvahu využití veřejné IP adresy, a tedy přesměrování veřejné adresy na interní IP adresu přidělenou pro Home Assistant. Zde je možné využít pouze IP adresu, nebo zakoupit a využívat vlastní doménu. Toto řešení bylo otestováno v této konfiguraci -> Vlastní doména směřuje na veřejnou IP adresu, která směřuje na server na lokální adrese, kdy server přesměrovává požadavky na základě domény na interní adresy. Přesměrování na základě domény je implementováno za pomoci webového serveru apache2 a konfigurace virtuálních hostů. [12]

Dynamické DNS

Jako druhá varianta je využití technologie dynamické DNS. Tato technologie funguje na principu aktualizace záznamů v reálném čase a díky tomu lze provozovat službu na dynamicky se měnících IP adresách.

Jako příklad služby fungující na tomto principu je DuckDNS. Na webu duckdns.org lze po registraci vytvořit až 5 vlastních subdomén směřující na lokální systémy. Po instalaci doplňku do Home Assistantu s názvem DuckDNS, lze v konfiguraci spojit účet a následně již přistupovat k webovému rozhraní Home Assistantu z jiné sítě. [7]

Home Assistant Cloud

Třetí varianta je využití cloudové služby, přímo podporované Home Assistantem. Jedná se o službu Nabucasa. Tato služba má hned několik výhod. Mezi nejhlavnější je jednoduchost instalace, kdy se stačí v samotné lokální instalaci Home Assistantu přihlásit k účtu na cloudu. Po přihlášení se otevírá řada pokročilých možností pro ovládání domácí instalace odkudkoliv.

Nabucasa podporuje integraci chytrých asistentů jako je Google Assistant nebo Alexa. Díky tomu lze lokální systém ovládat například i hlasem. [9]

Vlastní nastavba pro komunikaci mimo síť

Další možností je vlastní nastavba pro komunikaci mimo lokální síť. Zadáním bylo stanoveno vytvořit metodu umožňující univerzální ovládání z externího serveru, ale také umožňující serveru získávat data na vyžádání pro možnosti ukládání do vlastní databáze. Toto řešení navazuje na architekturu systému z projektu Smart Grid a zlepšuje možnosti automatického ovládání a přidávání nových zařízení bez nutnosti interakce se systémem na lokální síti.

6.1 Analýza a návrh

Pro tvorbu vlastní metody pro komunikaci se serverem je zapotřebí si nejprve definovat tři základní části systému, které je nutné vyřešit. Jako první je zapotřebí vyřešit samotné ovládání Home Assistantu, a to buď s pomocí automatizace, tedy využít systém z projektu Smart Grid, nebo vytvořit nový systém na jiném principu. Jako druhá část je potřeba vyřešit komunikaci mezi serverem na veřejné adrese a lokální instalací Home Assistantu. Poslední část je serverová část, starající se o samotné ovládání a získávání dat.

6.1.1 Ovládání Home Assistantu

Jak již bylo zmíněno, řešení je rozvinuto ze základu již hotového řešení z projektu Smart Grid. Avšak řešení v projektu Smart Grid je spíše vytvořeno jako Proof-of-concept, tedy pro reálné použití skrývá několik zásadních nedostatků. Na straně lokálního Home Assistantu toto řešení funguje na principu ruční tvorby automatizace. To znamená, že pro každou novou entitu, tedy pro každé nové přidané zařízení, je zapotřebí ručně vytvořit pravidlo pro automatizaci, díky kterému lze poté v určitých intervalech odesílat data na server, nebo naopak čekat na podnět ze serveru pro změnu stavu entity na lokální instalaci Home Assistantu. V reálném nasazení to tedy znamená, že v případě jakékoliv změny na infrastruktuře chytré domácnosti, ať už se jedná o přidání čidla na zavírání dveří, nebo složitějšího zařízení, jako je chytrá hlavice na topení, ani jedno ze zařízení nebude možné automaticky přidat do vzdáleného ovládání. Pro to, aby zařízení mohlo být vzdáleně ovládáno je, v řešení z projektu Smart Grid, vždy potřebné

vytvořit novou automatizaci, na lokální instanci Home Assistantu, popisující nové zařízení a čekající na podnět ze serveru. Až poté lze nové zařízení ovládat, nebo získávat v intervalech data. Například pro chytrou zásuvku je zapotřebí ručně vytvořit automatizaci pro odesílání dat na server, jako například odesílání aktuální spotřeby, nebo aktuálního stavu. Zároveň je třeba vytvořit další automatizaci čekající na podnět ze serveru, na jehož základě se změní stav entity, v tomto případě vypne nebo zapne zásuvku. Toto řešení je tedy zapotřebí změnit a zlepšit, aby bylo možné vzdáleně zjistit změny v architektuře chytré domácnosti, automaticky přidávat nová zařízení a také získávat data na základě podnětu ze serveru. [13]

Řešení s lokální API a MiddleWarem

Home Assistant obsahuje na lokální síti API pro webové rozhraní. Pomocí tohoto rozhraní se lze dotazovat na stavy všech entit, získávat data o všech entitách, ale také lze měnit stavy všech entit, nebo i konfigurovat Home Assistant.

V případě využití tohoto API pro vlastní řešení vzdáleného ovládání odpadají nevýhody předchozího řešení, tedy nutnost psát automatizace pro každé zařízení zvlášť a zároveň lze automaticky přidávat nová zařízení bez nutnosti interakce s lokální instancí.

Vlastní řešení tedy spočívá ve vytvoření vlastního softwaru zastávající roli middle-ware, tedy určitého překladače komunikujícího se serverem a zároveň s lokální instancí Home Assistantu. Software má tedy několik zásadních požadavků viz. tabulka.

Funkční požadavky systému	Nefunkční požadavky systému
Komunikace mezi serverem a lokálním HA ⁴	Možnost univerzálního napojení ⁵
Získávání dat	Jednoduchost systému
Ovládání Entit	System jako služba
Detekce nových entit	Dokumentace příkazů

Tabulka 4 - Funkční a nefunkční požadavky systému.

⁴ HA = zkratka pro Home Assistant

⁵ Možnost napojení systémů na základě jiných platforem

6.1.2 Komunikace mezi serverem

V původním řešení z projektu Smart Grid je komunikace mezi lokální instancí Home Assistantu řešená za pomoci technologie MQTT. Jedná se o protokol s architekturou Client – Server, typu Publish-Subscribe. Tedy jako server funguje takzvaný MQTT Broker, na který jsou připojeni klienti. Klienti mohou posílat nové zprávy v určitých tématech (Publish) a také přijímat zprávy z určitých témat ke kterým se přihlásí (Subscribe). V řešení z projektu Smart Grid se využívá jako server open source řešení *Mosquitto MQTT Broker*. Toto řešení se jeví jako ideální volba, vzhledem k možnostem konfigurace systému, kdy tento MQTT Broker podporuje zabezpečenou šifrovanou komunikaci a ACL⁶ pravidla pro klienty, díky kterým lze systém zabezpečit. [13]

Pro nový systém tedy lze zvolit technologii MQTT, lze použít i stejný MQTT Broker, ale je zapotřebí pozměnit strukturu komunikace pro užití s vlastním middlewarem. Toto řešení skrývá hned několik výhod, díky kterým splňuje požadavky.

Požadavky systému
Nutnost pouze jedné veřejné IP adresy pro MQTT Broker.
Možnost využití jednoho MQTT Brokeru pro více domácností.
Zabezpečená komunikace heslem a certifikátem pro šifrovanou komunikaci.
ACL pravidla pro oddělení jednotlivých domácností od témat jiných domácností.
MQTT jako multiplatformní řešení.

Tabulka 5 - Požadavky na systém.

6.1.3 Serverová část

V serverové části je zapotřebí vyřešit instalaci a konfiguraci MQTT Brokeru na základě požadavků z tabulky č5. Po správné konfiguraci je součástí serverové části této práce také

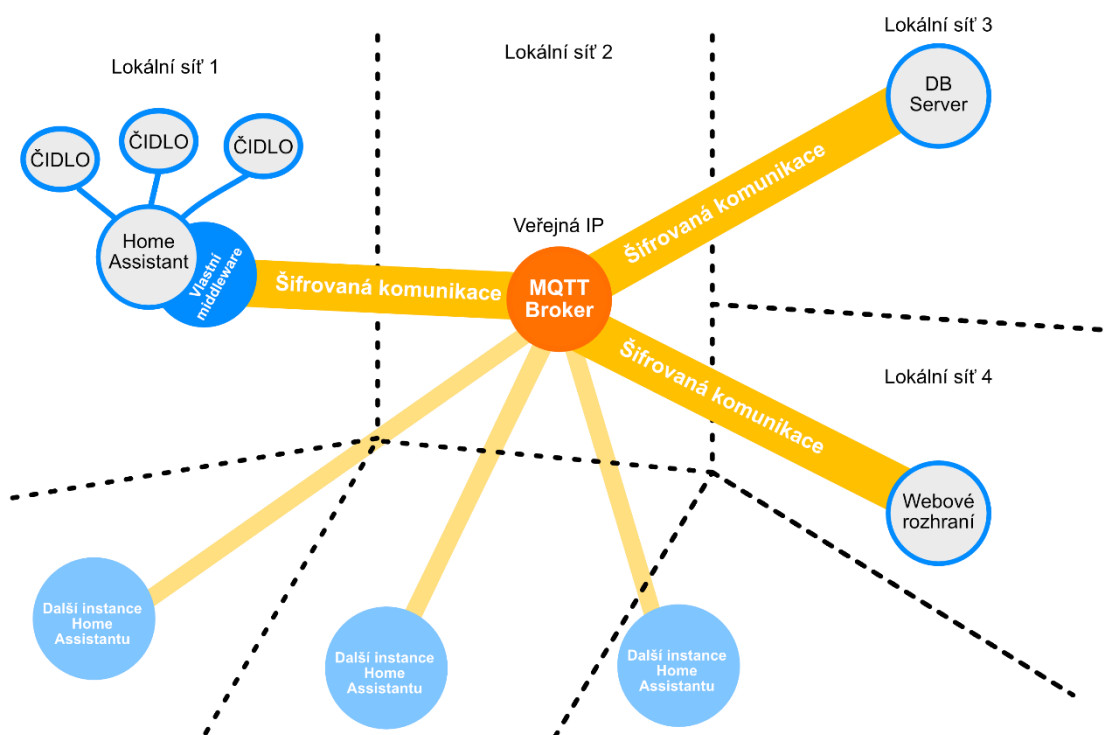
⁶ ACL = Acces control list neboli list pro řízení přístupu, je pokročilý způsob řízení uživatelů a jejich přístupu.

vytvoření jednoduché webové aplikace jako proof-of-concept demonstrující využití některých příkazů z vytvořené dokumentace.

Samostatné rozhraní pro ovládání, nebo napojení na databázi může být napsané v jakémkoliv jazyku na jakékoliv platformě podporující MQTT. Ať už se jedná o PHP, Python, C# nebo Javu. Díky tomu bude možné tento systém napojit na jakýkoliv další budoucí systém a například využívat sběr dat pro další analýzu, nebo vyhodnocování dat.

6.2 Implementace

Pro implementaci byla zvolena kombinace lokální služby běžící na Home Assistantu, napsané v jazyce python. Jako komunikační protokol byl zvolen MQTT pro jeho jednoduchost následné integrace a univerzálnost platformem. Na obrázku níže lze vidět architekturu tohoto řešení.



Obrázek 2 - Architektura nového řešení.

6.2.1 Lokální middleware

Služba, běžící na lokální instanci Home Assistantu, komunikující na lokálním webovém rozhraní. Program je napsaný v pythonu a nativně běží na systému. Program se skládá ze dvou hlavních částí.

První část je samotná komunikace s lokální instancí Home Assistantu. Ta probíhá na základě využití webového API Home Assistantu. API běží na portu 8123 na localhostu systému. Pro připojení je třeba takzvaný *long life token*, ten je nutné vygenerovat po přihlášení v Home Assistantovi. Token slouží pro autentizaci služby a má platnost po dobu 10 let. Po správném připojení program může komunikovat s lokálním rozhraním. Zde se jedná o využití příkazů z dokumentace Home Assistantu. Jedná se o HTML metody POST a GET. Součástí této části programu je také kontrola výstupu z Home Assistantu. Data z Home Assistantu přichází ve formátu JSON, je tedy potřeba validovat. Mezi zásadní příkaz patří příkaz pro výpis všech zařízení integrovaných v Home Assistantovi. Zde se JSON parseruje na jednotlivé objekty a dále posílá vyfiltrovaný výpis, včetně všech možných vygenerovaných témat pro ovládání a získávání dat.

Druhou částí je samotná komunikace s MQTT Brokerem, v programu nazvaném jako cloud. V prvních řádcích programu jsou konfigurační proměnné, kde lze nadefinovat IP cloudu, alias pro tuto konkrétní instanci Home Assistantu, heslo k brokeru a cestu k certifikátu potřebnému ke komunikaci. Následuje samotná metoda pro připojení k MQTT Brokeru a také metoda pro automatické přihlášení se k tématům pro ovládání.

Celý kód služby s popisem je dostupný v příloze této práce v části se softwarem ve složce „/Home Assistant/Vzdálené ovládání“. Pro zprovoznění služby je potřeba následovat kroky pro vytvoření služby z python souboru přímo v konfiguraci Home Assistantu. Popis je součástí přílohy č.2 Uživatelská dokumentace .

6.2.2 MQTT Broker

Pro správné fungování systému pro vzdálené ovládání je zapotřebí centrální bod, tedy MQTT Broker. Zde byl vybrán open-source Eclipse Mosquitto Broker. Broker musí být nainstalovaný na serveru s přístupem k veřejné IP adrese. Po instalaci je zapotřebí pro každou novou instanci Home Assistantu, nebo pro každou novou instanci aplikace, vytvořit nového uživatele

z důvodu zabezpečení. Tomuto uživateli je poté zapotřebí pomocí ACL pravidel nastavit přístupová práva jen k tématům která jsou pro danou instanci. Nutné je také vygenerovat certifikát pro broker. Díky tomu poté lze využívat šifrovanou komunikaci na základě ověření pomocí certifikátu. [14]

6.2.3 Aplikace třetí strany

Samotné ovládání, nebo získávání dat, probíhá na stejném principu jako odesílání dat, tedy za pomoci MQTT a zpráv odesílaných v tématech (topics). Součástí přílohy této práce je dokumentace možných příkazů, podle které se lze napojit na lokální instanci Home Assistantu a začít získávat data. V principu se jedná o zasílání příkazů do předem určeného tématu podle předlohy v dokumentaci a následné získání dat v jiném tématu určeném jako výstup. Výstup Home Assistant posílá ve formátu JSON. Jedná se tedy o strukturovaný formát, který se poté může snadno formátovat a získávat konkrétní data například pro databázový server.

Aplikace třetí strany může být vytvořena prakticky v jakémkoliv jazyku, s jedinou podmínkou, a to nutnost možnosti síťové komunikace za pomoci šifrovaného MQTT protokolu. Jako příklad možných aplikací, které jsou do budoucna zamýšlené, si můžeme uvést webové rozhraní napsané v PHP nebo jednoduchý skript na získávání historických dat a vkládání do databázového serveru.

Součástí přílohy č.1 je dokument „Příkazy podporované vzdáleným přístupem.docx“ obsahující, jak již název napovídá, příkazy a také příklady užití. Díky tomu lze v budoucnu sestavit výše popsané aplikace.

7 Integrace zakoupených plug-n-play zařízení

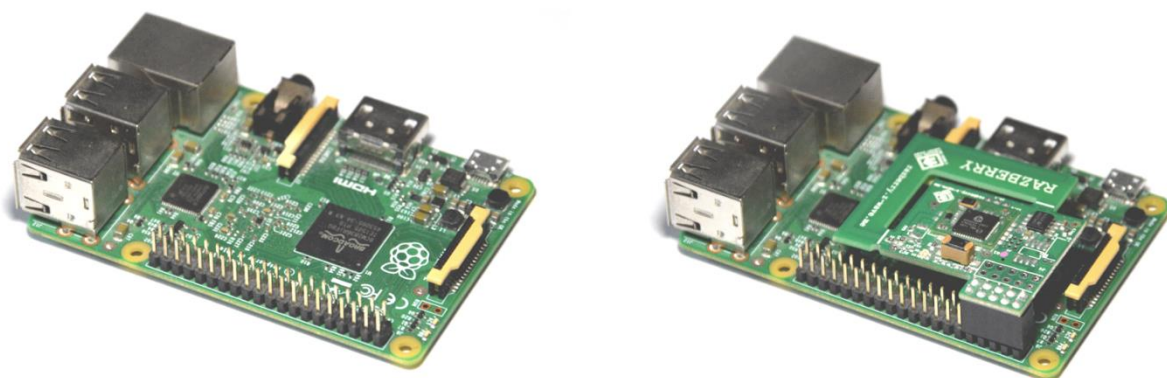
Pro otestování univerzálnosti samotného Home Assistantu je součástí práce také instalace, integrace a vyzkoušení automatizací za pomoci dostupných zařízení. Jedná se o zařízení na základě technologie Z-Wave.

7.1 Instalace prerekvizit

Jelikož komunikační technologie Z-Wave není standardní technologie jako wifi, nebo Bluetooth, běžné počítače neobsahují tento modul. Z tohoto důvodu je zapotřebí k počítači připojit buďto USB modul Z-Wave, nebo v tomto případě připojit modul přímo zapojitelný na GPIO header samotného Raspberry Pi.

RaZberry2

RaZberry2 je přídavný modul určený na GPIO headery minipočítače Raspberry Pi. Jeho kompatibilita je od nejstarší verze až po nejnovější Raspberry Pi 4, včetně miniaturizované verze Raspberry Pi Zero. Modul je napájený přímo z desky Raspberry Pi pomocí interního 3.3V stabilizátoru. Komunikace obou desek je za pomoci rozhraní UART TTL. Pro správné fungování je tedy zapotřebí modul zapojit na první GPIO headery desky. (viz obrázek)



Obrázek 3 - Raspberry Pi bez modulu Z-Wave (vlevo) a s modulem (vpravo).

Po fyzické instalaci přichází na řadu konfigurace modulu v Raspberry Pi, konkrétně v systému Home Assistantu. Konfigurace se mírně liší na základě verze použitého Raspberry Pi. U novějších verzí, které obsahují Bluetooth, je zapotřebí jej deaktivovat. A to z důvodu, že Bluetooth modul komunikuje přes rozhraní UART, které má Raspberry Pi pouze jedno. To lze udělat po otevření systémového souboru “/boot/config.txt” a přidáním posledního řádku “dtoverlay=disable-bt”. Toto platí u Raspberry Pi verze 3. V případě, kdy je nainstalovaná na Raspberry Pi starší verze systému Home Assistant, je poslední řádek konfigurace “dtoverlay=disable-bt” nahrazen řádkem “dtoverlay=pi3-disable-bt”. U nejnovější verze počítače, tedy u verze 4 je zapotřebí přidat dva řádky na konec souboru. A to jsou “dtoverlay=disable-bt” a “enable_uart=1”. Po změně konfigurace je zapotřebí Raspberry Pi nejprve restartovat bez připojeného modulu RaZberry2, poté vypnout, zapojit modul, a znovu zapnout. U starší desky verze 1 a 2 není editace systémového souboru zapotřebí.

Nyní přichází na řadu konfigurace integrace Z-Wave. Ve správě Home Assistantu, v záložce nastavení -> integrace -> přidat integraci, přidáme novou integraci s názvem Z-Wave. Pro úspěšné přidání je zapotřebí určit cestu k zařízení. V tomto případě, kdy využíváme modul RaZberry2 je cesta “/dev/ttyAMA0”. Po správné instalaci integrace můžeme přejít ke samotné konfiguraci Z-Wave integrace, nastavení -> integrace -> Z-Wave (konfigurovat). Zde je k dispozici ovládání Z-Wave brány, tedy například možnost spuštění, restartu, nebo přidání dalších chytrých zařízení do Z-Wave sítě. [15]

7.2 Instalace chytrých zařízení

Pro účel této práce je zvoleno několik chytrých zařízení využívající technologii Z-Wave. Mezi tato zařízení patří chytrá zásuvka, chytrý ventil na topení, senzor otevření dveří nebo okna, senzor teploty, světla a pohybu. Vzhledem k tomu, že tato zařízení jsou standardní zboží k dostání v obchodě, všechna obsahují návod a podrobný popis instalace. Avšak obecný postup instalace je vždy stejný.

Prvním krokem je zapojení napájení, v případě chytrých senzorů, klíčenek, nebo například hlavice na topení, se jedná o napájení pomocí baterií. V případě chytré zásuvky, nebo Z-Wave repeateru se jedná o zapojení do elektrické sítě. Po zapojení je důležité na Z-Wave bráně aktivovat režim přidání nového uzlu. Poté má každé zařízení určeno, jak se připojit. U většiny použitých zařízení je příkaz pro připojení daný několikanásobným stisknutím tlačítka na zařízení. Toto je však pro každé zařízení podrobně popsáno v návodu.

Po přidání chytrého zařízení do Z-Wave sítě je již zařízení připraveno k použití, bez jakékoli dodatečné konfigurace.

7.3 Použití chytrých zařízení

Když máme zařízení přidaná v Home Assistantu jako entity, můžeme přejít k samotnému chytrému používání a tím si sestavit chytrou domácnost. Nyní máme možnost nastavovat automatizace, například podle času zapínat zásuvku. To má praktické využití v domácnosti pro příklad jako spínání světla u akvária, nebo spínání malého elektrického topení v určitý čas. Jako další příklad automatizace může být upozornění v případě otevření okna, pomocí okenního senzoru, nebo upozornění na pohyb v místnosti za pomoci senzoru pohybu.

To ovšem není u automatizací všechno, protože hlavní výhoda jednotného systému jako je Home Assistant je možnost vytvářet komplexnější automatizace ovládající různá zařízení a reagující na různé podněty. Jako příklad je automatizace vytápění v pokoji. Home Assistant sleduje polohu uživatele a pokud uživatel ve všední dny opouští domov, vypne topení v místnosti, ale naopak když opouští místo práce, automaticky se začne místnost vytápět, aby při návratu byl dosažen teplotní komfort. Naopak o víkendech může být vytvořeno pravidlo, aby se vytápění nevypínalo, nebo když osoba odejde jen do jiné obce nakupovat.

Další možností je automatizace s podmínkami. To může být s využitím senzorů. Pro příklad vylepšíme automatizaci akvária. Zapínání světla akvária v určitém čase můžeme také obohatit o senzor světla a díky tomu svítit a zhasínat i mimo časově definovanou automatizaci.

8 Vlastní nízkonákladové zařízení

V této a následujících kapitolách se zaměřím na samotný vývoj několika nízkonákladových chytrých zařízení kompatibilních pro použití s Home Assistantem. Při vývoji těchto zařízení je kladen důraz na jednoduchost, dostupnost součástek a energetickou náročnost v případě napájení přes baterie.

8.1 Všeobecná analýza HW komunikačních technologií

Nyní se podíváme na možné technologie komunikace. Porovnáme náklady na pořízení, dosah signálu, ale také i běžnou dostupnost součástek.

8.1.1 Wi-Fi

Známou a velmi rozšířenou technologií v domácnostech je Wi-Fi. Jako technologie je wifi založená na využívání bezlicenčního pásma. Díky tomu je budování Wi-Fi sítí levné a dostupné pro každého.

Výhody technologie Wi-Fi jsou bezesporu její rychlost, pořizovací náklady, prostupnost signálu v interiéru, ale také samotný fakt, že Wi-Fi obsahuje v dnešní době prakticky každá domácnost, firma, i čím dál více i veřejné prostory. Další výhodou je provoz bez nutnosti pořizování specializovaného HUBu.

Nevýhoda této technologie je její větší spotřeba, proto není vhodná jako užití pro bateriově napájená zařízení, která potřebují mít neustálé bezdrátové připojení. Další nevýhodou může být samotný fakt rozšířenosti, kdy může, v hustě obydlených oblastech, docházet k zarušení.

V případě použití jako komunikační technologie pro nízkonákladové chytré zařízení, se tato technologie jeví jako dobrý kompromis mezi spotřebou a cenou. Největší výhodou je možnost zakoupit mikrokontroler s komunikačním čipem za cenu od 100 korun, tedy odpadají další náklady za mikrokontroler či komunikační čip. [16]

8.1.2 Bluetooth

Druhou silně využívanou technologií je Bluetooth. Tato technologie se využívá primárně pro přenos dat na kratší vzdálenosti. Například mezi mobilním telefonem a chytrým zařízením, ať už se jedná o chytré hodinky, bluetooth reproduktor, nebo infotainment v autě.

Mezi výhody této technologie patří její pořizovací cena, kdy desky podporující Bluetooth se dají pořídit okolo 50 Kč. Další výhodou je možnost připojení na mobilní telefon, tedy možnost přímého ovládání z chytrého telefonu. Největší výhodou je však velmi nízká energetická spotřeba. Naopak největší nevýhodou Bluetooth je jeho dosah. Ten je pouze v řádu několika metrů. A v případě překážek jako je kamenná zeď je prostupnost této technologie prakticky nulová. Proto je Bluetooth, na užití jako komunikační technologie pro chytrá zařízení do domu, nepoužitelné. [16]

8.1.3 Z-Wave

Dle současné nabídky chytrých zařízení pro domácnost, je nejvyužívanější technologie Z-Wave. Vzhledem k jejím technickým vlastnostem se tato technologie jeví jako ideální řešení pro vývoj vlastních zařízení.

Samotná technologie byla od začátku plánována pro využití v chytrých zařízeních, díky tomu se zaměřuje na nízkou energetickou náročnost. Nízké energetické náročnosti dosahuje způsobem komunikace, kdy odesílá jen potřebné pakety v krátkých intervalech. Další obrovskou výhodou je samotná specifikace, kdy všechna zařízení Z-Wave musejí komunikovat předem daným standardem. Díky tomu je instalace jakéhokoliv zařízení, od jakéhokoliv výrobce velmi jednoduchá. Další výhodou je funkce tvorby vlastní sítě tzv „mesh network“ kdy Z-Wave zařízení mezi sebou mohou komunikovat a předávat si zprávy dál, i mimo dosah původního hubu. Bohužel tato funkce je možná pouze u zařízení napájených ze sítě, jako jsou chytré zásuvky, nebo přímo Z-Wave opakovače. [16, 17]

Ovšem výhodou standardu je vykoupená nevýhodou. A to je nutnost licence. Všechna zařízení využívající Z-Wave potřebují licenci, kvůli tomu i mikrokontrolery postavené na Arduinu, s funkcí Z-Wave, stojí o řád více než wifi mikrokontrolery. Jako příklad si uvedeme „Z-Wave.me Z-Uno“ s cenou přesahující 1500 korun. Další nevýhodou této technologie je

nutnost HUBu, který sám o sobě stojí dalších 1500 korun ,v případě modulu pro Raspberry Pi, nebo v případě levnějších USB modulů okolo 800 korun.

Při testování technologie v domě, kde se vlastní chytrá zařízení budou nasazovat, byl zjištěn další nedostatek. Tím je nutnost využití nejméně dvou, v lepším případě třech opakovacích signálů. To dále zvyšuje pořizovací cenu. Oproti Wi-Fi signálu, signál Z-Wave nedokázal stabilně proniknout do pokojů v druhém patře domu.

8.1.4 LoRa

Další technologií, využívanou pro internet věcí, je LoRa. LoRa je zkratka pro „Long Range“. Z toho vyplývá, že tato technologie je primárně určená pro zařízení komunikující na velké vzdálenosti. Jako příklad si můžeme uvést senzory vlhkosti na různých místech na poli, senzory na dobytku, nebo meteostanice.

Obrovskou výhodou je prostupnost signálu, kdy v exteriéru se může jednat o více než 9 kilometrů. Další výhodou je nízká energetická náročnost samotných vysílačů.

K technologii LoRa patří také LoRaWAN. LoRaWAN je síť postavená na technologii LoRa. Tato síť je dostupná celosvětově, se stabilním pokrytím ve většině Evropy, Asie, Ameriky i Austrálie. V Africe je pokrytí jen částečné. Pro domácí užití je možné sestavit vlastní síť za pomoci brány a modulů připojených k mikrokontrolerům, nebo připojit se do veřejné LoRaWAN sítě. [16]

K využití v tomto případě, tedy výroba nízkonákladové hlavice a dalších chytrých zařízení, je tato technologie levnější než Z-Wave, ale náklady jsou stále násobně převyšující technologii Wi-Fi. Při sestavení zařízení pomocí LoRa je zapotřebí zakoupit jak komunikační čip, v ceně okolo 300 korun, tak i samotný mikrokontroler v ceně od 150 korun v případě neoriginálního Arduina.

8.1.5 Ethernet

Alternativou k bezdrátovým technologiím je využití síťového kabelu. Natažení vlastní kabelové sítě pro chytrá zařízení je instalačně náročná. V případě instalace chytrých hlavic, teplotních čidel a podobně, je tato instalace oproti využití bezdrátové technologie velmi časově, ale i finančně náročná. Avšak výhodou je stabilita spojení, kdy k výpadku může dojít pouze

přerušením kabelu. Další výhodou je možnost napájet zařízení přímo po síti a díky tomu odpadá nutnost měnit baterie.

Tato technologie byla pro tvorbu chytrých zařízení také zamýšlena. Dokonce v tomto případě, kdy je v domě natažená dostatečná síťová infrastruktura, je i cena a náročnost instalace velmi zjednodušená. Pro použití je zapotřebí pořídit mikrokontroler, tedy cena okolo 150 korun a k němu LAN modul, například „LAN8720“ s cenou okolo 100 korun.

8.1.6 Zhodnocení

Závěrem se tedy jeví jako nejlepší řešení využít Wi-Fi integrované mikrokontrolery. Kdy pořizovací cena převažuje nevýhodu ve větší spotřebě. Tu lze navíc pomocí omezení komunikace dostatečně snížit, pro umožnění i několikaročního provozu senzorů. Druhou možnou variantou je využití síťového kabelu.

Tabulka řešení Wi-Fi	
Pořizovací cena brány (Wi-Fi router)	Od 250 korun
Pořizovací cena komunikačního čipu	Od 100 korun
Pořizovací cena mikrokontroleru	Mikrokontroler součástí komunikačního čipu
Dosah signálu v interiéru	Od 10 po 25 metrů
Dosah signálu v exteriéru	600 metrů
Odběr při komunikaci	80 mA
Odběr při komunikaci ve špičce	Až 300 mA
Odběr při spánku	0,2 mA

Tabulka 6 - Zhodnocení řešení pomocí Wi-Fi.

8.2 Hardware prostředky

V této kapitole se zaměřím na popis fyzických součástek možných pro využití v praktické části práce. Vzhledem k cenově nejlepší variantě při užití Wi-Fi komunikace, beru v potaz zařízení komunikující na základě této technologie.

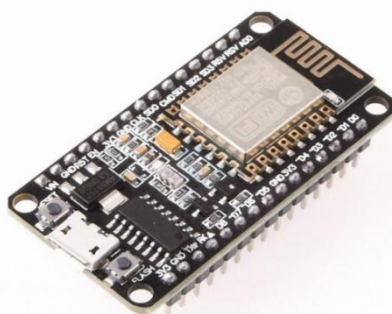
8.2.1 Kontroléry

Jako první součástka je mikrokontroler. Mikrokontroler je mozem chytrého zařízení. Jedná se o součástku umožňující vykonávání vlastního programu napsaného vývojářem. Při tvorbě nízkonákladových zařízení se počítá s využitím dvou základních typů mikrokontrolerů. Prvním jsou mikrokontrolery z rodiny ESP, které již integrují komunikační rozhraní, druhým je mikrokontroler Arduino, který pro komunikaci vyžaduje externí rozhraní.

ESP8266 Lua NodeMcu

Pro vytvoření prakticky jakéhokoliv chytrého zařízení komunikující přes Wi-Fi lze zvolit mikrokontroler ESP8266 Lua NodeMcu. Tento mikrokontroler obsahuje vlastní USB TTL převodník, díky tomu není pro programování třeba využívat externí zařízení a stačí pouze microUSB kabel. Další velikou výhodou je počet vyvedených pinů na plošném spoji. Tento model obsahuje 1 analogový vstup a 11 digitálních. S tímto modelem je také možné využívat režimu deepsleep a probouzet s využitím interního nízkoenergetického časovače.

Cenově se tento model mikrokontroleru pohybuje okolo částky 150 korun. V porovnání s ostatními mikrokontrolery využívaných v tomto projektu, se jedná o rozměrově největší mikrokontroler s rozměrem cca 35 mm x 50 mm.

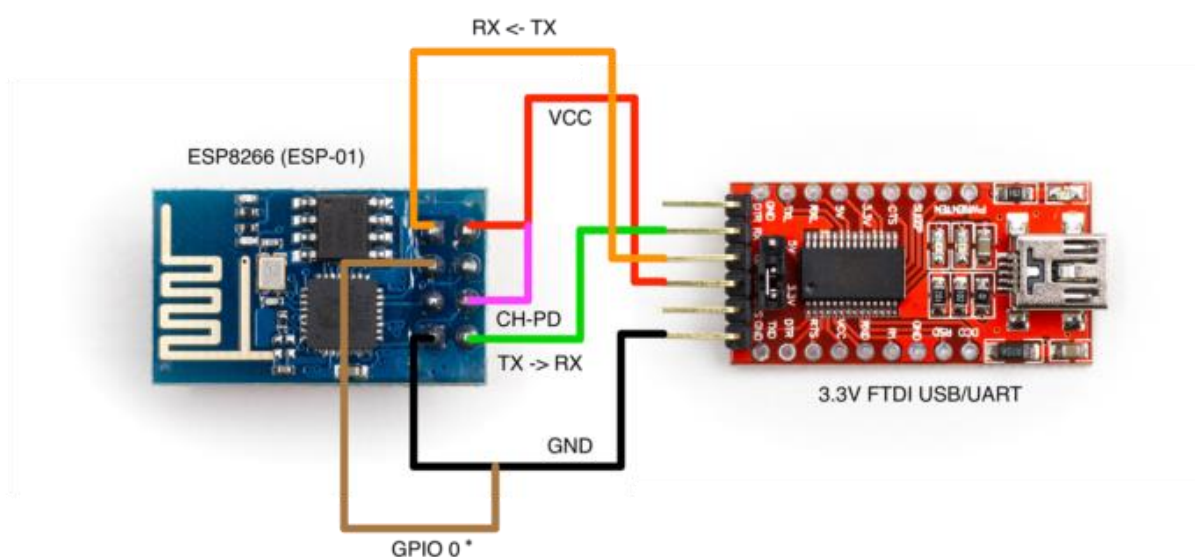


Obrázek 4 - ESP 8266 LUA NodeMcu. (zdroj: laskarduino.cz)

ESP8266-01

Model ESP8266-01 vychází z originálního ESP8266. Jedná se o zmenšenou, nízkonákladovou verzi mikrokontroleru. Tato verze obsahuje stejný čip, ale hlavní rozdíl je v plošném spoji, kdy pro použití jsou vyvedeny pouze nezbytné piny. Mezi tyto piny patří RX a TX piny pro sériovou komunikaci, napájecí piny, zem a dva digitální GPIO piny. Díky tomu je tato verze bez modifikace použitelná například jako komunikační rozhraní pro Arduino. Kdyby tento model byl zamýšlený k čidlu, kde by byla potřeba využít režimu deepsleep pro úsporu baterie, je zapotřebí připájet nový pin k samotnému čipu. Stejně pravidlo platí pro analogový vstup. To je, za určité dávky trpělivosti, proveditelné.

Vzhledem k absenci TTL USB převodníku na mikrokontroleru, je potřeba pro programování a komunikaci s PC připojit mikrokontroler k USB programátoru. Vzhledem k logickému napětí tohoto mikrokontroleru, tedy 3.3v je nutné využít programátor umožňující komunikaci na této logické úrovni napětí. Komunikace probíhá přes sériový port, tedy za využití RX a TX pinů. Správné zapojení lze vidět na obrázku. [18]



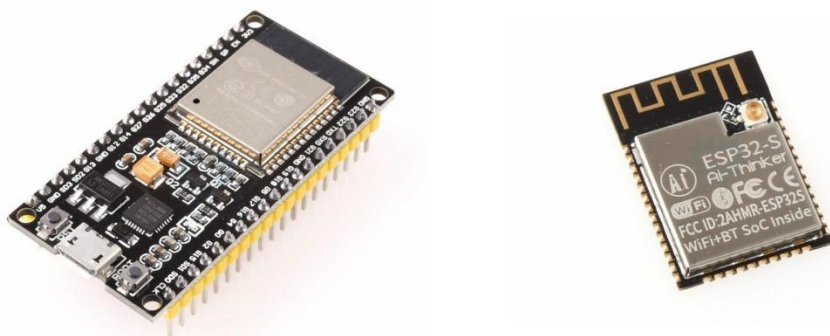
Obrázek 5 - Zapojení ESP8266-01 s programátorem. (zdroj: hwlab.cz)

Cenově se tento model mikrokontroleru pohybuje okolo částky 100 korun. Co se týká rozměrových vlastností, ESP8266-01 je bezkonkurenčně nejmenší mikrokontroler s rozměrem 15 mm x 24 mm.

ESP32

Novější verzí je ESP32, kdy se jedná o kompletně novou desku, včetně užití nového čipu. Díky tomu tento model převyšuje funkce svých předchůdců ve všech ohledech. Ať už se jedná o komunikaci, kdy dokáže komunikovat na dvojnásobné rychlosti, a to 300Mbit, nebo o integraci Bluetooth, které zcela chybí u předešlých verzí. ESP32 obsahuje dvoujádrový procesor, větší paměť, více GPIO pinů, včetně 6 vstupních analogových pinů s lepším 12bitovým rozlišením (oproti 10bitovému u ESP8266). ESP 32 se prodává v různých variantách, ať se jedná o vývojovou desku s programátorem, až po SMD komponent bez programátoru.

Cenově se tento model pohybuje okolo částky 150 korun za SMD model, tedy model bez programátoru a bez vyvedení pinů, nebo okolo částky 250 korun za model s programátorem a vyvedenými piny.



Obrázek 6 - Varianty modulu ESP 32. (zdroj: laskarduino.cz)

Arduino nano

Arduino nano je jediný využitý mikrokontroler neobsahující rozhraní pro bezdrátovou komunikaci. Ačkoliv neobsahuje rozhraní pro komunikaci, má řadu výhod. První je integrovaný programátor a integrovaný stabilizátor napětí. Druhou velikou výhodou, a také hlavním důvodem využití pro měřič spotřeby, je vyšší rozlišení analogového vstupu, tedy z toho vycházející vyšší přesnost měření. Pro budoucí vylepšení na měření celého domu, a tedy měření třech fází je výhoda i více analogových vstupů.

Cenově se tento mikrokontroler pohybuje v částkách okolo 100 korun za klon, nebo okolo 600 korun za originál. Rozdíl ve funkčnosti originálu a klonu není žádný.

8.2.2 Senzory a čidla

Pro sestavení chytrých zařízení je potřeba i další součástky kromě mikrokontrolerů. Tím jsou různá čidla a senzory. Záleží pouze na požadavku uživatele, jaká chytrá zařízení požaduje sestrojít a tím jaké senzory a čidla budou potřeba. Pro potřeby této práce, tedy pro sestrojení chytré hlavičky, měřiče spotřeby a detektoru zatopení sklepa, jsou potřeba následující zařízení. Jako první je teplotní čidlo *Dallas ds18b20* pracující na základě technologie 1-Wire. Druhým možným senzorem je *ASAIR DHT11* taktéž využívající technologii 1-Wire. Jedná se o kombinovaný senzor teploty a vlhkosti vzduchu. Tento senzor má bohužel velmi nepřesné čidlo teploty. Dalším je senzor vlhkosti půdy, použitý pro detektor zatopení. Ten je sestavený z obvodu *LM393* + *elektrody*. Jedná se o nejjednodušší senzor vlhkosti pracující na základě detekce vodivosti mezi elektrodami.

8.2.3 USB programátory

Vývojové desky bez vlastního programátoru, jako jsou například desky ESP8266-01, ESP8266-PRO nebo samostatná deska ESP32, potřebují ke komunikaci s počítačem USB programátor. Jedná se o zařízení komunikující s počítačem po sběrnici USB a s mikrokontrolerem pomocí sériové komunikace. Tyto programátory obvykle obsahují i napájecí stabilizátory. V práci byl použit programátor *CP2104 STC*.

8.3 Potřebné programy

8.3.1 Arduino IDE

Pro potřeby programování mikrokontrolerů je při realizaci praktické části využíváno vývojové prostředí Arduino IDE. Toto vývojové prostředí podporuje platformy jako je Windows, Linux nebo MacOS. Arduino IDE podporuje všechny mikrokontrolery postavené na platformě jako Arduino.

8.3.2 Python IDE

Jelikož součástí práce je také naprogramování vlastního řešení komunikace mezi serverem a Home Assistantem, je zvolen pro programování v jazyce python jeho vlastní vývojové prostředí python IDE.

8.3.3 Autodesk Fusion 360

Vzhledem k nutnosti vytvářet vlastní mechanické části pomocí technologie 3D tisku, je nutné využívat software pro tvorbu technických 3D modelů. Z důvodu osobní preference je zvolen software od společnosti Autodesk, a to Fusion 360. Jedná se o cloudový 3D editor. Data jsou vždy uložena na serverech společnosti, a tedy nehrozí jejich nenávratná ztráta.

Parametrické modelování

Nesporná výhoda editorů tohoto typu je při postupu modelování, kdy se modeluje na základě parametrů v krocích. Tato metoda může připomínat samotný reálný postup tvorby součástky, kdy máme kroky jako vyříznutí panelu, vyvrtání díry a potom následné prořezání závitu. Výhodou je, že kdykoliv v procesu modelování se můžeme vrátit na jakýkoliv krok a pozměnit parametr a tím změnit výsledný model. Díky tomu, v případě zjištění problému na závěru modelace, není potřeba celý model znovu modelovat, ale stačí pozměnit parametry.

Alternativy

Fusion 360 není jediný 3D modelovací software použitelný pro parametrické modelování. Jako alternativy můžeme použít SolidWorks, Inventor, AutoCAD, nebo Onshape.

8.3.4 Prusa Slicer

Po vytvoření 3D modelu je potřeba model převést do formátu čitelného pro 3D tiskárnu. Na to slouží tzv. slicery. Slicery jsou programy které model rozdělí na jednotlivé tiskové vrstvy, a převedou postup do kódů čitelných pro 3D tiskárnu. Celý kód je potom postup, jak má 3D tiskárna pohybovat a extrudovat materiál v jejím tiskovém prostoru.

8.3.5 Autodesk Eagle

Vzhledem k plánu sestavit více kusů jednotlivých zařízení je nejlepší řešení navrhnout vlastní plošný spoj. K tomu slouží program na plánování obvodů a plošných spojů, Autodesk Eagle. Základem je sestavit schéma obvodu, poté lze vygenerovat plošný spoj, nebo manuálně spoje nakreslit. Díky integrované databázi mnoha elektronických součástek a zároveň možnosti součástky importovat, je možné plošné spoje naplánovat přímo na použití s konkrétními součástkami.

9 Chytrá hlavice

Prvním zařízením je chytrá hlavice. Toto zařízení je z hlediska vývoje nejsložitější. Je zde zapotřebí vyřešit čtyři hlavní problémy. A to mechanickou část hlavice, ať již z pohledu uchycení hlavice na ventilu, až po pohon pro uzavírání ventilu. Druhou otázkou je zvolená technologie komunikace a s tím související zvolený mikrokontroler. Posledním problémem je způsob napájení.

9.1 Analýza

Funkční požadavky zařízení	Nefunkční požadavky zařízení
Komunikace s centrálním Hubem, v tomto případě s Home Assistantem.	Nízká pořizovací cena součástek a jako celku.
Mechanický pohon – otevírání a zavírání ventilu	Rozšiřitelnost pro další aplikace
Napájení – baterie a kabel	
Dálkové ovládání přes Home Assistant	

Tabulka 7 - Funkční a nefunkční požadavky chytré hlavice.

9.1.1 Mechanická část

Vzhledem k zadání, kdy hlavice je nízkonákladová, s možností sestavení z běžně dostupných součástek s využitím 3D tisku, je zapotřebí vyřešit několik otázek. První otázkou, je způsob pohybu s trnem ventilu. Standardní ventily u topení fungují na principu trnu, který je proti pružině vytlačen směrem od ventilu. V klidovém stavu je tento ventil otevřen, a tedy pro zavření je nutné vyvinout dostatečnou sílu proti pružině na zatlačení trnu do ventilu. To lze vyřešit dvěma způsoby.



Obrázek 7 - Foto ventilu s trnem.

Termoelektrický pohon

První je využití termoelektrického pohonu. Tento pohon funguje na principu tepelné roztažnosti materiálu. Existují dvě verze, podle toho, jaký je normální stav hlavice. “NC – normally closed” a “NO – normally open”, což znamená, že verze NC je bez napájení zavřená, oproti verzi NO, která je bez napájení otevřená. Mechanicky tento pohon funguje tak, že při chladu, je píst stlačen, a tedy nepůsobí žádnou silou, ale v momentě zahřátí se píst vytlačí. Většinou bývá napájen 230 V, ale existují modely s nižším napájením, například pouze 24 V.

Výhodou tohoto pohonu je jeho tichost, kdy v provozu nevydává žádné zvuky. Nespornou výhodou je také mechanická jednoduchost, kdy kromě pístu vytlačovaného tepelnou roztažností neobsahuje žádnou jinou důležitou mechanickou část.

Tento pohon ovšem má i své nevýhody. Nejzásadnější je energetická náročnost a neefektivita. Tento typ pohonu má pouze dva stavy. Zavřeno a otevřeno, kdy vždy jeden z nich je stav, který se musí udržovat konstantním příívodem proudu. v případě přerušení proudu se hlavice pomalu vrátí do svého výchozího stavu.

Cena těchto pohonů se pohybuje v rozmezí od 399 Kč až po ceny převyšující ceny chytrých hlavic.

Elektromotorový pohon

Druhou možností je využití motoru pro z převodovaný posun pístu, který tlačí na trn ventilu. Tento pohon je součástí většiny chytrých bezdrátových hlavici. Je tomu tak hlavně z důvodu nízké energetické náročnosti oproti termoelektrickému pohonu. Motor lze napájet stejným napětím jako mikrokontroler, a navíc po změně stavu již není nutné udržovat konstantní proud do motoru. Díky tomu motor odebírá proud pouze když mění stav, tedy když otevírá, nebo zavírá hlavici.



Obrázek 8 - Foto mechaniky rozebrané komerční hlavici.

Pro aplikaci v bezdrátové hlavici, je nejlepší řešení pohon pomocí elektromotoru, proto také bude zvolen. V průběhu psaní této práce probíhalo hledání jednotlivých již hotových variant, pro ulehčení technické náročnosti projektu. Bohužel samotný elektromotorový pohon nelze zakoupit. Zde byla uvažována i varianta modifikace levné hlavice poháněné tímto pohonem, avšak pořizovací cena by převyšovala náklady na výrobu vlastního pohonu, a tedy odporovala by zadání nízkonákladové hlavice. Tento pohon bude vytvořen pomocí technologie 3D tisku v kombinaci s pořízeným elektromotorem. Při analýze bylo vyzkoušeno několik elektromotorů, ale pouze jeden vyhovuje požadavku na velikost a výkon. Tím je *“DC motorek 6–12 V, 8–16 RPM s převodovkou a dlouhou hřídelí”* v ceně 58 Kč za kus.



Obrázek 9 - DC motor se šnekovým převodem. (zdroj: laskarduino.cz)

Druhou otázkou v řešení mechanické části hlavice je její uchycení na ventil. zkoumáním běžně dostupných hlavic na ventily, bylo zjištěno, že všechny mají stejný závit, a to metrický závit M30x1.5, na který lze přikoupit redukci v případě jiného typu ventilu. Tento závit, vzhledem k jeho rozměrům, nebude problém vytisknout

9.1.2 Technologie komunikace a mikrokontroler

Pro komunikaci mezi hlavicí a Home Assistantem lze zvolit několik možností komunikace. V úvahu připadají technologie Z-Wave, LoRa nebo Wi-Fi. Jak již bylo v analýze komunikace popsáno, technologie Z-Wave i LoRa by se v konečném důsledku stala velice nákladná, a tedy jako nejlevnější a nejlepší řešení se jeví Wi-Fi. Ačkoliv je Wi-Fi energeticky náročnější, lze dosáhnout nízké spotřeby chytrým naprogramováním softwaru. Nejzásadnějším ušetření energie je integrování režimu deepsleep, kdy hlavice s nikým nekomunikuje a má spotřebu v řádu μA . V důsledku to bude znamenat, že se hlavice bude připojovat k wifi jen v délce několika sekund a v intervalech v řádu až desítek minut. Díky tomu se větší spotřeba wifi eliminuje. Na stejném principu funguje i otestovaná hlavice na technologii Z-Wave, kdy se hlavice neaktualizuje v reálném čase, ale pouze v časových intervalech.

Využití Wi-Fi přináší kromě pořizovacích nákladů také výhodu možnosti využít mikrokontroler s integrovanou wifi, tedy eliminaci nutnosti využít dvě zařízení, což by v případě LoRa znamenalo kombinaci kontroleru Arduino Nano a desky s komunikačním rozhraním

LoRa. V tom důsledku by cena elektroniky snadno přesáhla hranici 500 Kč. V případě Wi-Fi stačí využít mikrokontroler od výrobce Espressif, jako je například ESP8266 nebo ESP32. Tyto mikrokontrolery obsahují integrované wifi rozhraní a díky tomu není třeba žádných dalších dceřiných desek. Ceny těchto desek se v české republice, v době psaní této práce, pohybují v rozmezí od 100 korun až po 160 korun.

9.1.3 Napájení hlavice

Pro napájení můžeme zvolit dva základní způsoby. Napájení pomocí baterie, nebo napájení ze sítě. V případě napájení ze sítě se jeví jako nejlepší řešení integrace microUSB portu na hlavici se stabilizátorem napětí na 3.3V, které potřebuje mikrokontroler. Toto řešení má nesporné výhody v možnosti neustálého napájení, díky tomu odpadá nutnost výměny baterií, navíc lze napájet motor 5V z USB a díky tomu dosáhnout rychlejší otevírání a zavírání ventilu. Další výhodou je možnost zkrácení intervalu komunikací mezi hlavicí a Home Assistantem.

Cena tohoto řešení se skládá z ceny USB portu 8 Kč a stabilizátoru 12 Kč, ale také ceny adaptéru pro napájení, která se může pohybovat v rozmezí od 100 Kč po několik stovek.

Druhé řešení je napájení pomocí baterie. Zde se nabízí dvě řešení. Napájení pomocí akumulátoru, například standardní článek 18650, nebo napájení pomocí vyměnitelných baterií. Jako ideální se jeví řešení s dobíjecím článkem 18650, kdy v případě vybití stačí přijít k hlavici s powerbankou a dobít jí. Ovšem zde narážíme na problém úrovně napětí, kdy mikrokontroler vyžaduje cca 3V a baterie dodává 4.2V - 3,4V, nominálních 3.6V. Díky tomu by bylo potřeba integrovat stabilizátor, který by měl konstantní spotřebu a díky tomu by se výdrž na baterii razantně zkrátila. Pokud budeme brát v úvahu spotřebu stabilizátoru 4 mA, samotný stabilizátor by dokázal vybit baterii s kapacitou 2000mAh za zhruba 20 dní, a to bez započítání spotřeby kontroleru a motoru.

$$2000\text{mAh} / 4\text{mA} = 500 \text{ hodin} \quad 500 \text{ hodin} / 24 = 20,8 \text{ dnů}$$

Správným řešením napájení za pomoci baterií je s využitím článků s nominálním napětím 3V. Díky tomu odpadá nutnost stabilizátoru, tedy zbytečné konstantní spotřeby. Všechna zkoumaná chytrá zařízení, která jsou napájena baterií, jsou napájena právě články s nominálním

napětím 3V. To lze docílit sériovým zapojením dvou článků s nominálním napětím 1.5V, nebo využitím jednoho článku 3V.

Cena tohoto řešení obsahuje cenu boxu pro baterii, cca 20 Kč v závislosti na typu baterie, ale také musí zahrnovat cenu samotných baterií která se může pohybovat v rozmezí od 120 Kč v závislosti na zvolené kapacitě, kde platí že čím větší kapacita tím delší výdrž, ale také tím vyšší cena.

9.1.4 Software

Co se týká softwaru pro hlavici, je zapotřebí vyřešit několik otázek. První je řízením motoru a detekce pozice, to se jeví jako nejlepší řešení pomocí detekce proudu za pomoci čtení analogového vstupu. Dále je potřeba vyřešit komunikaci s Home Assistantem s přihlédnutím k nároku, aby hlavice byla nízkoenergetická. To znamená že je zapotřebí připojovat se pouze v krátkých impulzech. Další částí je samotné vyhodnocení, zda je nutné topit, a také režim spánku.

9.2 Teoretické řešení – Návrh systému

Tato část práce je rozdělena do třech bloků, v souvislosti s tématem. Tedy vývoje mechanické části, v tomto případě se jedná o převodovku a systém zatlačování trnu u ventilu hlavice. Druhým blokem je vývoj elektroniky, kde je potřeba popsat funkčnost jednotlivých elektronických součástí a jejich význam. A v poslední části se zaměřím na vývoj softwaru a jeho funkčnost.

9.2.1 Mechanická část – uchycení a převodovka s výtlačným členem

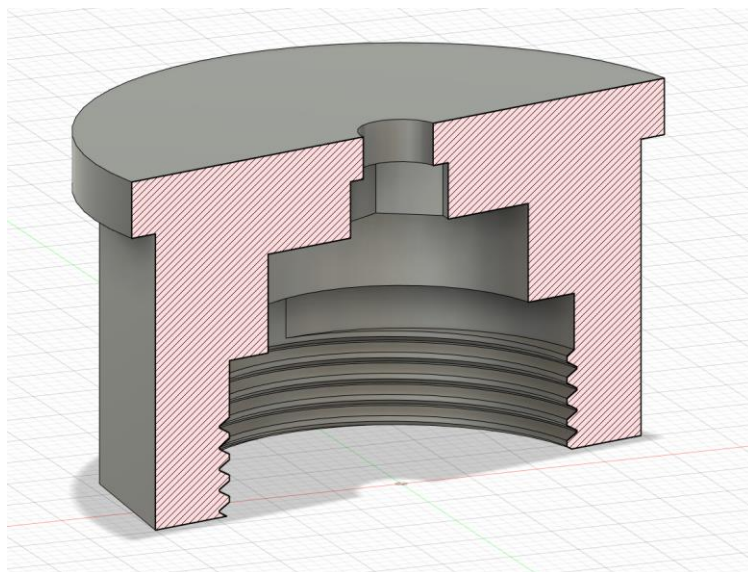
V analýze jsme zjistili, že univerzální uchycení pro ventil hlavice je za pomoci matice metrického rozměru M30x1.5. Díky velikosti závitu lze tento typ uchycení bez problému vytisknout na 3D tiskárně. Na přiloženém obrázku č.2 Lze vidět řez modelem uchycení hlavice (růžová část) na ventilu (žlutá část).

Hlavní otázkou je, jak převést rotační pohyb motoru na lineární pohyb pro tlačení trnu. Díky analýze několika komerčních řešení bylo zjištěno, že všechna využívají určitý typ závitové

tyče, pohybující se v závitě, v kombinaci se dvěma ozubenými koly, v následující konfiguraci. Větší, ale nižší ozubené kolo je osazeno na závitové tyčce, naopak menší ale vyšší ozubené kolo je usazené na samotném výstupu z převodovku/motoru. Díky tomu je závitová tyč s ozubeným kolem schopna pohybovat se nahoru, respektive dolů.

Bohužel kvůli velkým silám vyvíjeným pružinou trnu ve ventilu, není ideální možností vytisknout závit jako výtlačný člen na 3D tiskárně. Zde hrozí stržení závitu, ale také neplynulost a zbytečné zadržávání při pohybu po vrstvách které jsou způsobené technikou 3D tisku.

Jako ideální řešení je využití nerezového šroubu jako výtlačného členu a nerezové matice jako závitu, uložené v těle hlavice. Pro matici a šroub byl zvolen závit M5. Tento rozměr dosedá s dostatečnou rezervou na trn ventilu, ale zároveň nezabírá zbytečně mnoho místa v hlavici, a tím neoslabuje její pevnost. Na obrázku č.10 lze vidět průřez uchycení hlavice s místem pro vsazení matice a zároveň s otvorem umožňující šroubu volný pohyb ve směru otevírání či zavírání trnu ventilu. Matice musí být umístěna uvnitř závitu držící hlavici na ventilu, v opačném případě by vytlačování šroubu nezpůsobilo zatlačení trnu, ale pouze vytlačení matice z těla hlavice.



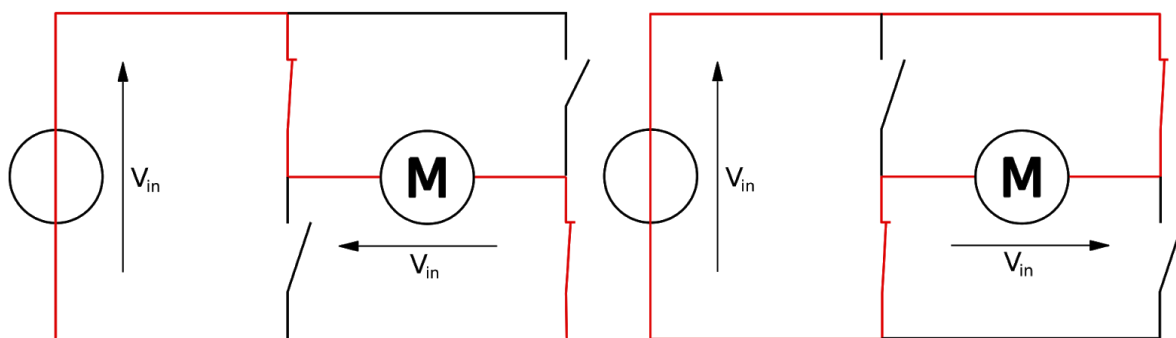
Obrázek 10 - Průřez modelem matice na topení.

Využitý motor, již obsahuje vlastní převodovku se šnekovým převodem, díky tomu odpadá nutnost projektovat vlastní převodovku a lze využít již hotové řešení. Při prototypování bylo zjištěno že motor s převodovkou dokáže zavřít většinu ventilů, ovšem u některých

docházelo k problémům. To je však vyřešeno dostatečným zubovým převodem s poměrem velikosti ozubených kol na výtlačném členu hlavice 1:2.

9.2.2 Elektronická část – řízení motoru a detekce pozice

Vzhledem k modelu motoru a jeho náročnosti na proud, není možnost napájet motor napřímo z mikrokontroleru. Mikrokontrolery, v tomto případě například ESP8266, umějí na výstupní piny poskytnout maximálně 12 mA na pin. Proto je nutné motor napájet napřímo baterií nebo zdrojem a směr otáčení určovat externí elektronickou součástkou. Na to slouží tzv H-můstek. H-můstek je název pro elektrický obvod obsahující tranzistory v sestavení umožňující změnu polaritu výstupu. Velikou výhodou integrovaných H-můstků je jejich velikost a možnost spínání většího výkonu, než je signál z mikrokontroleru. Dva základní stavy H – můstku lze vidět na obrázku č.11.



Obrázek 11 - Schéma sepnutí H-Můstku. (zdroj: Cyril BUTTAY)

Díky využití H-můstku tedy lze otevírat a zavírat hlavici změnou výstupních signálů na mikrokontroleru. Samotné řízení motoru pro správné fungování hlavice nestačí. Je zapotřebí zjišťovat kdy již do motoru nemusí jít proud, a to v případě kdy je ventil topení zavřený, nebo naopak kdy je otevřený. Otevření ventilu znamená, že je výtlačný člen hlavice naplno zasunut to těla hlavice. To lze detekovat například mikrospínačem v těle hlavice. Problém však nastává v opačném případě, kdy musíme zajistit dovření ventilu. Každý ventil může mít jinou délku chodu trnu. Nutná délka chodu výtlačného členu se mění také s využitím adaptérů. Proto nelze jednoznačně určit přesné místo v pohybu, kdy je již ventil zavřený. To lze ale zajistit detekcí proudu tekoucího do motoru. Měřením bylo zjištěno těchto stavů. Při zavírání nebo otevírání ventilu motor spotřebovává v rozmezí 50mA - 60mA při orientačních 3V. V případě větší

zátěže, tedy dovírání ventilu a následné zastavení motoru převyšuje spotřebu 100mA. Z toho vyplývá, že na základě tekoucího proudu lze určit moment, kdy je již ventil zavřený. Měření proudu lze docílit za pomoci analogového vstupu na mikrokontroleru, kdy neměříme proud, ale pokles napětí mezi rezistorem, který je v sérii s motorem. Pomocí Ohmova zákona lze vypočítat, že při užití 5R odporu docílíme poklesu napětí 0.25V ve stavu zavírání, tedy při odběru 50mA a naopak ve stavu zavření, kdy dojde k zastavení motoru, tedy odběru přes 100mA, dojde k poklesu napětí o více jak 0,5V.

9.2.3 Elektronická část – Mikrokontroler a senzory

Druhou částí elektroniky je samotný mikrokontroler. V hlavicích využívám mikrokontrolery ESP, které mají integrované wifi rozhraní. Zde se nabízí několik možností, prototypy hlavic byly sestavené s použitím dvou variant.

První variantou je “ESP8266 Lua NodeMcu”. Tento mikrokontroler je výbornou variantou pro prototypování. obsahuje vlastní programátor a má vyvedeny všechny potřebné piny. Pro potřeby hlavice je nutné mít k dispozici piny umožňující probouzení z režimu deepsleep, analogový pin a 3 digitální IO piny. To vše tento modul obsahuje.

Druhou variantou je menší “ESP8266-01“. Toto zařízení je jedno z nejmenších a nejvyužívanějších z řad ESP. Avšak jeho příznivá cena a velikost je vyvážená nedostatkem pinů. Pro použití v hlavici se jedná o fatální nedostatky. A to je absence pinu pro probuzení z režimu deepsleep a absence analogového vstupu. Mikrokontroler jako takový však tyto piny obsahuje, nejsou pouze vyvedené na plošný spoj. Lze tedy opatrně připájet tyto dva piny přímo na čip mikrokontroleru. viz obrázek. Druhou nutnou úpravou pro hlavici je odpájení led diod, pro úsporu baterie.

Takto upravené ESP8266-01 je ideální řešení pro hlavice. ať už se jedná o velikost, cenu, nebo spotřebu. Nevýhodou je nutnost vlastnictví pájecí stanice s malými hroty a nutnost zručnosti a opatrnosti při pájení.

K měření teploty je využíváný senzor teploty ds18b20, pracující na technologii 1-wire. Díky tomu je třeba k senzoru přivést pouze napájení 3V a jeden signální vodič, který je spojený s jedním IO pinem na kontroleru. U tohoto signálního vodiče je nutnost spojení s větví 3V pomocí 470k rezistoru.

9.3 Implementace – Finální řešení

V předešlých kapitolách jsme vyřešili otázky vývoje chytré hlavice. Nyní přichází na řadu samotná implementace, tedy fyzické vytvoření zařízení. Implementace se dělí na tři hlavní části, mechanická část, tedy část s využitím 3D tisku, elektronická část a softwarová část.

Princip fungování hlavice již byl popsán v analýze. Pro finální řešení byla zvolena kombinace koupěných dílů, konkrétně nerezového šroubu M5x20 s odpovídající maticí bez brzdy a kombinaci vytištěných dílů. Díly byly spojené pomocí kvalitního lepidla. Na obrázku č.12 lze vidět jednotlivé díly hlavice. První součást je matice se závitem M30x1.5 pro našroubování na ventil, druhá část je distanční sloupek mezi motor a maticí. Třetí částí jsou převody pro převedení rotačního pohybu na lineární. Součástky jsou řešeny zvlášť z důvodu možné jemné manipulace po vytištění pro správné nastavení vůlí. Materiál pro tisk byl zvolen PET-G.

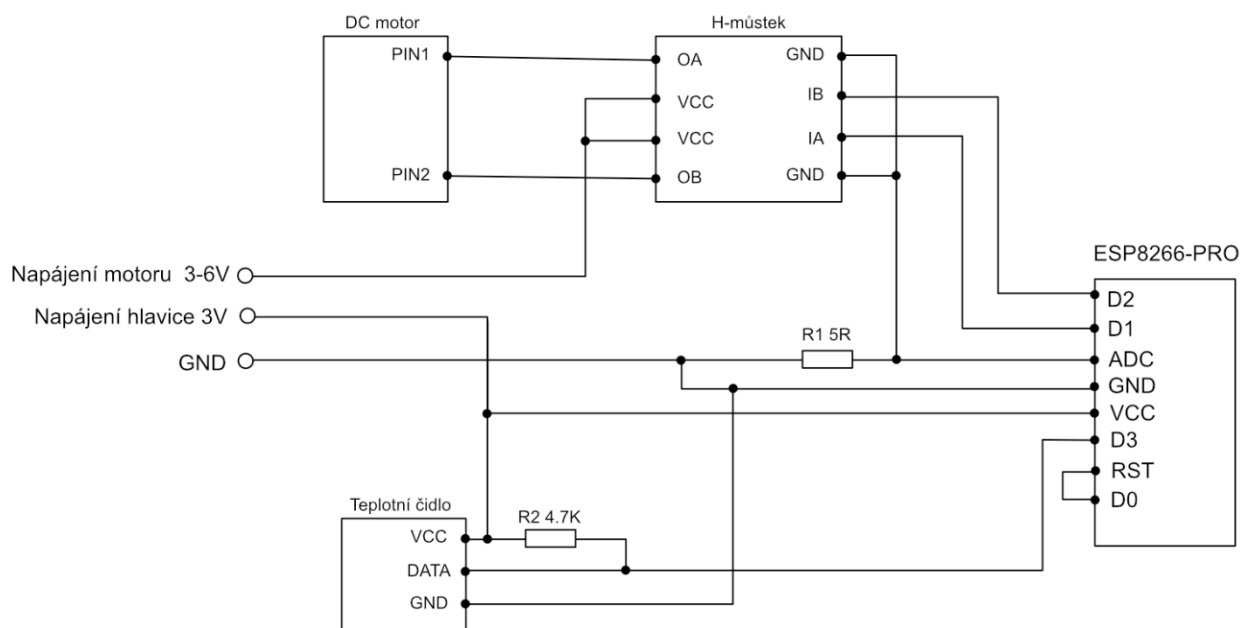


Obrázek 12 - Mechanické součástky pro hlavici.

Druhou částí je elektronika. Tedy mikrokontroler ESP8266, motor, H-můstek, senzor teploty, a rezistory. Obvod je sestaven dle schématu (obrázek č.13), tedy, přes zátěžový rezistor 5R je napájen H-můstek, starající se o ovládání směru rotace motoru na základě dvou vstupních pinů. Zátěžový rezistor na GND vodiči H-můstku slouží pro měření proudu motoru, díky poklesu napětí na rezistoru. Za rezistorem vede vodič do analogového vstupu ESP8266.

Dále je součástí teplotní čidlo, zapojené v ESP8266 zároveň s rezistorem mezi datovým pinem a VCC. Nutnost rezistoru je daná specifikací 1-Wire na základě které teplotní čidlo pracuje.

Třetí částí je program. Zde se nachází několik sekcí kódu. První sekce je konfigurace, kde se dají upravovat proměnné. Další sekcí je krátké připojení na server, toto připojení trvá



Obrázek 13 - Schéma zapojení hlavice.

3 vteřiny. Za tuto dobu je hlavice schopna z Home Assistantu získat veškerá nutná data a také nová data odeslat. Díky tomu je minimalizovaná spotřeba hlavice v režimu vysílání. Následující sekcí v programu je vyhodnocení teploty. V tomto případě se jedná o načtení teploty z čidla, tedy zjištění aktuální teploty v místnosti a vyhodnocení v porovnání s požadovanou teplotou získanou z Home Assistantu. Poté se program větví na dvě části starající se buďto o otevření hlavice, nebo naopak o zavření. Kód pro zavírání zapne motor ve směru rotace pro vysouvání trnu oproti ventilu. V průběhu je třeba kontrolovat analogový vstup, tedy kontrolovat, zda nedochází k zastavení motoru. K tomu dojde v momentě zavření ventilu. Poté tato část kódu zavolá režim spánku na čas v závislosti na nastavení. Naopak kód starající se o otevření hlavice zapne motor v opačném směru, a kontroluje, zda dojde k zastavení o tělo hlavice.

V momentě, kdy dojde k zastavení, zavolá se režim spánku na jednu minutu. Celý kód s dokumentací je součástí přílohy se softwarem ve složce „/WiFi_hlavice“.

Poslední částí je nastavení na straně Home Assistantu. K tomu je zapotřebí nejprve nastavit entitu *MQTT senzoru* a následně pomocí entity senzoru vytvořit entitu *Climate*. Tato entita podporuje nastavování teplot, zobrazování aktuální teploty, nebo i stav hlavice, zda aktuálně topí nebo ne. Konfigurace na straně Home Assistantu lze najít v příloze se softwarem ve složce „/Home Assistant/Konfigurace/“.



Obrázek 14 - Hotová hlavice ve variantě s MicroUSB napájením

10 Měřič spotřeby tepelného čerpadla

Druhým zařízením je měřič spotřeby. Účelem tohoto zařízení je měření proudu tekoucího do tepelného čerpadla, z toho následný výpočet spotřeby. Pro tento účel je tedy požadavkem, aby zařízení bylo schopno detekovat spotřebu a odesílat data na Home Assistant pro další vyhodnocení. Z hlediska spotřeby není kladen žádný požadavek na energetickou náročnost z důvodu přímého napojení na napájení. Z technického hlediska je zde třeba vyřešit několik otázek. Nejprve je třeba definovat si jakým způsobem proud měřit, poté jak proud vyhodnocovat a naposledy jak data odesílat na Home Assistant. Z hlediska cenového je zapotřebí sestavit řešení kvalitní, ale nízkonákladové.

10.1 Analýza

Funkční požadavky zařízení	Nefunkční požadavky zařízení
Komunikace s centrálním Hubem, v tomto případě s Home Assistantem.	Nízká pořizovací cena součástí a jako celku.
Napájení – kabel	
Neinvazivní měření jedné fáze napájející tepelné čerpadlo	

Tabulka 8 - Funkční a nefunkční požadavky měřiče spotřeby.

10.1.1 Sonda proudu

Na trhu jsou k dispozici různé druhy sond pro měření proudu. Lze je rozdělit do dvou skupin, invazivní a neinvazivní. Invazivní sondy proudu jsou takové, které je třeba napojit do obvodu přerušením vodiče. Druhá varianta, tedy neinvazivní, jsou sondy, které lze rozevřít a vodič do sondy vložit. Takovým sondám se říká proudový transformátor. Pro aplikaci v této práci, tedy pro měření spotřeby tepelného čerpadla, je nejlepší varianta proudový transformátor “SCT-013-000” s maximálním proudem 100 A, generující výstup v poměru 100 A : 50 mA. Ceny těchto proudových transformátorů se pohybují v české republice okolo 180 Kč, při dodávce z Číny okolo 70 Kč.

10.1.2 Mikrokontroler

Samotná proudová sonda nedokáže nijak hodnoty vyhodnocovat, ani odesílat. Proto proudový transformátor je napojený na analogový vstup mikrokontroleru. Jako nejlepší řešení se jeví využití mikrokontroleru Arduino Nano, vzhledem k počtu analogových vstupů, tedy možnosti budoucího vylepšení na měření všech třech fází.

Aby Arduino mohlo komunikovat s ostatními zařízeními, tedy odesílat data na Home Assistant, je potřeba přidat síťové rozhraní. To lze buďto bezdrátově, pomocí stejné technologie jako u hlavic, tedy přidáním například miniaturního ESP8266-01, nebo přidáním ethernetového modulu pro připojení síťovým kabelem. Cena kontroleru Arduino Nano se v české republice pohybuje od 100 Kč za klon, až po několik stovek za originální součástku.

10.1.3 Software

Pro software přichází v úvahu několik možných řešení. Prvním je napsání vlastního nového softwaru, jehož cílem je vyřešit otázku, jak získávat data, tedy čtení analogového vstupu, a následná interpretace získaných dat. Druhou možností je využít opensource knihovnu Emon-Lib, která je postavená s cílem řešit interpretaci dat získaných z analogového vstupu při užití proudového transformátoru.

Druhou částí softwaru je odesílání dat pro Home Assistant. Zde je třeba vyřešit komunikaci Arduina s ESP8266 a zároveň samotný software pro ESP8266, tedy připojení do sítě, spojení s MQTT Brokerem a odesílání dat.

10.2 Teoretické řešení – Návrh systému

Pro funkční měřič spotřeby je zapotřebí vyřešit dvě hlavní otázky, jak sestavit elektroniku a také jak měřič naprogramovat.

10.2.1 Elektronika měření

Jak již bylo v analýze řečeno, nejlepší řešení je využití neinvazivní sondy pro měření proudu, tedy sonda “SCT-013-000”. V jádru věci se jedná o proudový transformátor, tedy zařízení, které transformuje proud na primárním vinutí v poměru 100 A:50 mA na sekundární. V případě tohoto transformátoru je bezpodmínečně nutné jako první součástku zapojit na vývod zátěžový odpor. Díky tomu dostaneme na výstup rozsah napětí, které se v závislosti na proudu mění. Hodnotu tohoto odporu vypočteme pomocí následujícího vzorce, kde R je hodnota zátěžového odporu, U je maximální napětí analogového převodníku Arduino, I_{max} je maximální proud na primární straně transformátoru a nZ je počet závitů na proudovém transformátoru. [19]

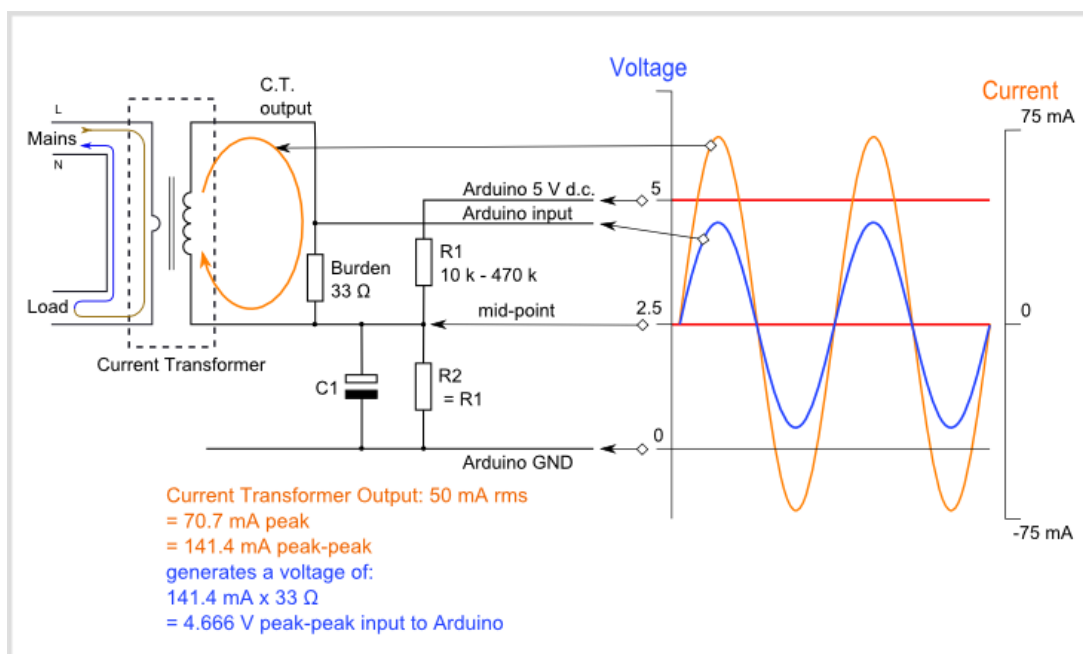
$$R = \frac{\frac{U}{2.0}}{\frac{I_{max} * 1.414}{nZ}}$$

V tomto případě dosadíme do vzorce hodnoty, kdy počet závitů transformátoru najdeme ve specifikaci produktu. V tomto případě se jedná o poměr závitů 2000:1. Po dosazení nám vyjde hodnota 35 Ω .

$$R = \frac{\frac{5}{2}}{\frac{100 * 1.414}{2000}}$$

Hodnota 35 Ω není standardní hodnota, tedy můžeme využít nejbližší sousední hodnotu, ale nesmíme překročit hodnotu 35 Ω , v opačném případě by překročení hodnoty mohlo způsobit vyšší napětí na výstupu, které by vedlo k destrukci Arduino. Zvolit lze například 33 Ω .

Výstup z transformátoru je střídavý proud, proto bylo pomocí vzorce vypočteno polo-
 viční napětí a nyní je potřeba pomocí obvodu děliče napětí hodnoty posunout do pozitivního
 spektra. Pro snížení šumu je využit kondenzátor s hodnotou $10\mu\text{F}$. [19] viz obrázek.



Obrázek 15 - Schéma obvodu s proudovým transformátorem. (zdroj: openenergymonitor.org)

10.2.2 Elektronika Arduina a ESP

Po zapojení sondy je zapotřebí vyřešit otázku propojení Arduina, které řeší čtení hodnot, s
 ESP, které data odesílá přes wifi na Home Assistant. To lze vyřešit pomocí sériové komuni-
 kace za užití pinů RX a TX na obou zařízeních. Menší komplikace je fakt rozdílných základních
 napětí, tedy Arduino 5V oproti ESP 3.3V. To lze vyřešit dvěma způsoby. Prvním je využití
 jednoduchého odporového děliče, nebo druhým řešením, je využití převodníku logických
 úrovní.

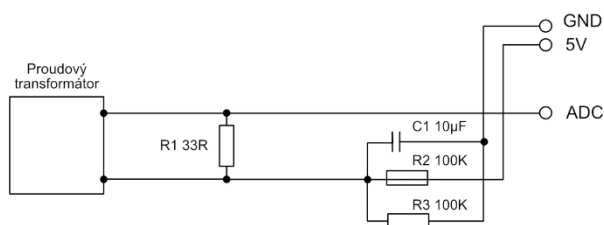
10.2.3 Softwarová část

Pro softwarovou část na Arduino byla využita open source knihovna EmonLib, doplněná o
 několik vlastních funkcí. Tato knihovna je určená pro měření proudu pomocí Arduino a prou-
 dových transformátorů. Jako vlastní doplnění je potřeba vypočítat spotřebu, tedy hodnotu v

jednotkách Wh, potažmo KWh. Druhé doplnění je již zmíněná komunikace s ESP8266 po sériovém portu. Druhá část je software ESP8266. Ten je založen na principu čtení vstupů ze sériového portu, následného převodu na proměnné a následného odeslání pomocí MQTT na Home Assistant.

10.3 Implementace – Finální řešení

První částí finálního řešení je tvorba hardwaru. To obsahuje tvorbu obvodu, případně tvorbu plošného spoje. Proudový transformátor nelze připojit napřímo k analogovému vstupu Arduino, mohlo by dojít k nenávratnému poškození samotného Arduino z důvodu vyššího napětí na vstupu. Proto je zapotřebí jako první součástku přidat zátěžový odpor na výstup proudového transformátoru, v tomto případě odpor s hodnotou $33\ \Omega$. Díky tomu na je výstupu rozsah - 2.5V až 2.5V. Další částí obvodu je dělič sloužící pro posunutí reference země, tedy posunutí sinusoidy ze záporného rozsahu do kladného, tedy v rozsahu 0V až 5V. Díky tomu se využívá celý rozsah Arduino a všechna data získaná z proudového transformátoru. Pro vyhlazení lze



Obrázek 16 - Schéma zapojení proudového transformátoru s užitím zátěžového rezistoru R1 a s děličem.

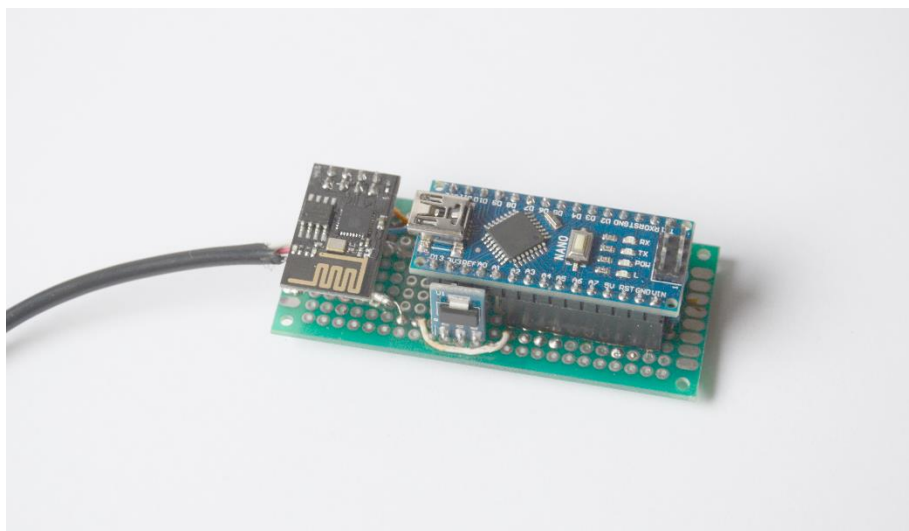
přidat kondenzátor. Obvod je sestaven dle následujícího schématu.

Další součástí hardwaru je napájení Arduino a ESP8266-01. Celá sonda bude napájena ze sítě za pomoci 5V adaptéru. Arduino obsahuje svůj vlastní stabilizátor napětí, lze ho tedy rovnou napájet 5V, to ovšem u ESP8266-01 nelze. Pro napájení je nutné použít buďto přímo 3.3V, nebo využít stabilizátor. V tomto případě byl využit stabilizátor AMS1117 3.3V, napájený ze stejné větve jako Arduino, tedy z 5V.

Poslední částí obvodu je datové spojení Arduino a ESP8266-01. To je provedeno spojením RX pinu Arduino ,a TX pin ESP a TX pinu Arduino na RX pin ESP. V komunikaci ve

směru Arduino → ESP8266-01 je zapotřebí využít buďto převodník logických úrovní, nebo využít dělič sestavený z odporů v poměru 1:2.

Po sestavení celého obvodu lze sestavit celé zařízení buďto na univerzální plošný spoj, což bylo využito v tomto případě, nebo je možné si nechat vyrobit plošný spoj přímo určený pro tento senzor. Schéma pro tvorbu plošného spoje je vytvořeno jako totožný obvod a je součástí přílohy. Obě varianty počítají s využitím konektorů, pro možnost kdykoliv odpojit Arduino nebo ESP8266-01, pro usnadnění flashování nového softwaru, nebo v případě poruchy pro snadnou výměnu.



Obrázek 17 - Hotový měřič spotřeby na vlastním plošném spoji.

Druhou polovinou finálního řešení je tvorba softwaru, ta se dělí na dvě části. Software pro Arduino a software pro ESP8266-01. Pro Arduino byla využita knihovna EmonLib umožňující měření a vypočítání aktuálního proudu pomocí výstupu proudového transformátoru na analogový vstup Arduino. Bylo zde zapotřebí definovat napětí a také zkalibrovat. Program využívající knihovnu EmonLib je postaven na základě dokumentace. Následně je do programu zahrnutá komunikace s ESP, tedy příprava a následné odesílání dat. Do ESP se po sériové lince odesílají data v intervalech 2 vteřin ve tvaru <napětí, proud, spotřeba>. Na straně ESP je několik funkcí. První je připojení na Wifi, s tím související kontrola, zda nedošlo k odpojení a následný reconnect. Druhou funkcí je připojení na MQTT Broker a odesílání dat. Třetí funkcí je čtení sériového portu a čekání na známou sekvenci, tedy na sekvenci

ve tvaru <napětí, proud, spotřeba>. Data jsou poté zpracována a napětí a proud jsou odeslána každý interval na Home Assistant. Spotřeba je odesílána ve větších časových intervalech, tedy konkrétně v intervalech 30 vteřin. To lze však změnit konfigurací. Oba programy a jejich dokumentace je součástí přílohy se softwarem ve složce „/Meric_spotreby/Arduino“ respektive „/Meric_spotreby/ESP“.

Poslední částí finální implementace je nastavení v samotném Home Assistantu. Je zapotřebí nastavit novou MQTT entitu, *MQTT sensor*, zastávající funkci zaznamenávání aktuální spotřeby. Následně lze vytvořit za pomoci integrace *utility meter* entitu, počítající spotřebu na dny, týdny nebo měsíce. Tato integrace v Home Assistantu podporuje také další pokročilejší funkce, jako například více tarifů. Konfigurace na straně Home Assistantu lze najít v příloze se softwarem ve složce „/Home Assistant/Konfigurace/“.

11 Detektor zatopení sklepa

Třetím zařízením je detektor zatopení sklepa. Toto zařízení má sloužit pro detekci většího zatopení sklepa a následnou notifikaci uživatele. Stejně jako předešlé zařízení, i u tohoto detektoru je kladen důraz na nízké náklady při jeho tvorbě. Je zde potřeba vyřešit otázky jako, jak spolehlivě detekovat zatopení, jak zařízení napájet a také jak data odesílat.

11.1 Analýza a návrh

Funkční požadavky zařízení	Nefunkční požadavky zařízení
Komunikace s centrálním Hubem, v tomto případě s Home Assistantem.	Nízká pořizovací cena součástek a jako celku.
Napájení – kabel	Jednoduchost sestavení.
Detekce vody ve výšce jednoho metru	

Tabulka 9 - Funkční a nefunkční požadavky detektoru zatopení sklepa.

11.1.1 Detekce zatopení

Vzhledem k tomu, že je sklep neustále zatopen z určité části, není žádoucí detekovat jakoukoliv hladinu vody. V tomto případě by čidlo hlásilo neustálé zatopení. Z tohoto důvodu bude měření zatopení prováděno ve výšce zhruba jednoho metru.

Pro samotnou detekci je uvažováno využít sondy vlhkosti. Na trhu jsou k dostání dvě hlavní varianty sondy vlhkosti. Sonda pro měření vlhkosti vzduchu, a naopak sonda pro měření půdní vlhkosti. Levnější varianta pro měření půdní vlhkosti je sestavena ze dvou elektrod od sebe fixně vzdálených a obvodu měřicího iontovou vodivost mezi nimi.



Obrázek 18 - Sonda půdní vlhkosti. (zdroj: laskarduino.cz)

Toto řešení se zdá jako nejjednodušší i pro měření zatopení. Jednoduše se umístí elektrody do výšky jednoho metru a v případě zatopení dojde k uzavření obvodu mezi elektrodami, tedy ke stavu určující zatopení.

Cenově se tato sonda pohybuje v České republice okolo ceny 40 korun. V případě dodání z Číny se cena pohybuje v třetinové hodnotě.

11.1.2 Mikrokontroler

Stejně jako u proudového transformátoru, ani sonda vlhkosti nedokáže vyhodnocovat a odesílat data na Home Assistant, k tomu slouží mikrokontroler. Vzhledem k jednoduchosti systému, je možno využít samostatného kontroleru ESP8266, nebo i ESP8266-01.

V případě využití ESP8266 lze využít režimu deepsleep, a tedy i při napájení z baterie dosáhnout i několikoroční výdrže. Vzhledem k tomu že není potřeba stav zatopení aktualizovat v reálném čase, stačí když se zařízení probudí jednou za půl hodiny a zkontroluje stav, a pouze v případě zatopení se připojí krátce na wifi a odešle varování. Režimu deepsleep lze docílit i při využití menšího zařízení ESP8266-01, avšak zde je potřeba připájet navíc jeden pin k samotnému čipu.

11.1.3 Napájení

Stejně jako u hlavice, i zde je možnost napájet zařízení buďto přes adaptér, nebo za pomoci baterie. I zde platí stejné pravidlo. Napájení přes síťový adaptér je výhodnější z hlediska

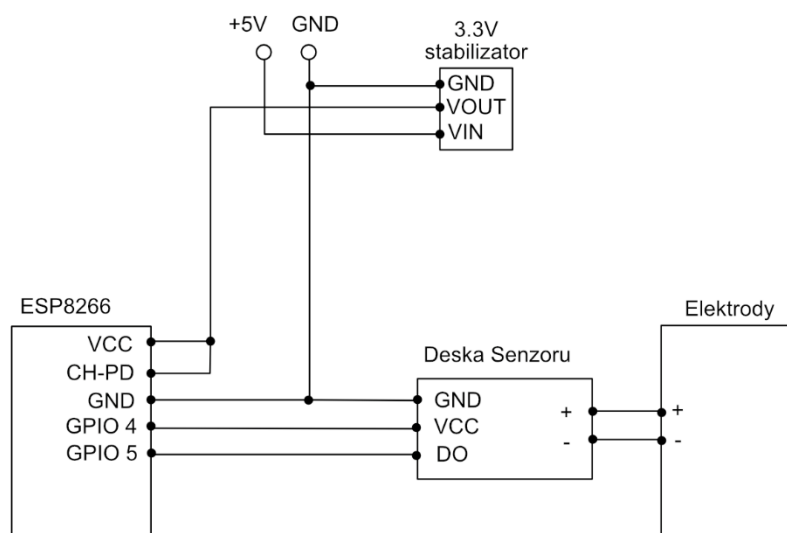
údržby, kdy není třeba kontrolovat baterie. Naopak nevýhoda spočívá v nutnosti natahovat napájecí kabel k zařízení. U bateriového napájení odpadá nutnost přívodního kabelu.

11.1.4 Software

Softwarově je potřeba vyřešit několik otázek na straně ESP8266. První je detekce signálu ze senzoru, který podle stavu mění výstup na datovém pinu na hodnotu LOW, nebo HIGH⁷. Další částí je nutnost vyřešit spotřebu samotného senzoru, je tedy zapotřebí využít funkci deepsleep ve větších intervalech. Poslední částí je nutnost vyřešit připojení do sítě, spojení s MQTT Brokerem a odesílání dat.

11.2 Implementace

Samotná implementace detektoru zatopení se skládá ze dvou částí stejně jako měřič spotřeby, hardware a software. V hardwaru je třeba vyřešit napájení ESP, to je provedeno za pomoci stabilizátoru napětí a síťového zdroje, následně je potřeba vyřešit propojení senzoru vlhkosti s ESP v souladu s příloženým schématem. Elektrody senzoru jsou poté umístěny do výšky jednoho metru pro detekci vodní hladiny v této výšce.



Obrázek 19 - Schéma zapojení detektoru zatopení sklepa

⁷ HIGH a LOW jsou napěťové úrovně 3V a 0V, fungující jako digitální vstup.

Softwarová část obsahuje dvě hlavní metody, metodu pro připojení do sítě a zároveň připojení na MQTT Broker. Druhá metoda je jednoduchý cyklus kontroly, zda nedošlo k zatopení. Tato kontrola probíhá v určeném intervalu, nastavena je jedna hodina jako výchozí interval pro kontrolu. Zdokumentovaný program je součástí přílohy se softwarem ve složce „/Cidlo_zatopeni/“.

Kromě samotné tvorby detektoru je zapotřebí zařízení také nastavit v Home Assistantu. Zde se jedná o jednoduchou implementaci entity *senzor*. Poté si uživatel již může nastavit automatizaci například pro notifikaci emailem, či za pomoci SMS v případě zatopení. Konfigurace na straně Home Assistantu lze najít v příloze se softwarem ve složce „/Home Assistant/Konfigurace/“.

12 Testování a ladění

Nyní je třeba všechny nové prvky chytré domácnosti otestovat v porovnání s komerčním řešením. Díky tomu lze zjistit silné ale i slabé stránky jednotlivých produktů.

12.1 Chytrá hlavice

Prvním zařízením na otestování je vlastní chytrá hlavice. Zařízením k porovnání je komerčně dostupná *termostatická hlavice Danfoss Z-wave*. Testování a porovnání probíhá v těchto kategoriích.

- Testování a porovnání funkčnosti
- Testování mechanické odolnosti
- Porovnání spotřeby
- Cenové porovnání

12.1.1 Testování a porovnání funkčnosti

Chytrá hlavice, vytvořená jako součást této práce, je založena na analýze běžně dostupných zařízení, ale zároveň také na požadavcích, tedy zejména na nutnosti nízkých nákladů a nízké energetické náročnosti. Kvůli tomu jsou základní funkce hlavice totožné s hlavicemi běžně dostupnými v prodeji, ale pokročilé funkce, jako je například displej nebo možnost nastavit teplotu přímo na hlavici, v současné verzi hlavice chybí. To je vyvoláno nutností vytvořit hlavici v nízkém rozpočtu a také pro snížení spotřeby, kdy by displej byl jen dalším prvkem snižující výdrž na baterii.

V průběhu testování funkčnosti, bylo odhaleno několik menších problémů v kódu, které byly opraveny. Dále bylo zjištěno několik možných vylepšení pro budoucí iterace hlavice, jako například využití novějších ESP modulů pro snížení spotřeby v režimu spánku, nebo pro zmenšení elektroniky.

Sestrojená hlavice přináší i několik svých vlastních funkcí. Nejzásadnější je možnost nastavit různé frekvence obnovy dat a kontroly teplot se serverem. Díky tomu lze, za pomoci automatizace v Home Assistantu, nastavit časy, kdy není potřeba aby hlavice komunikovala

často se serverem, jako například, když nikdo není doma, nebo v noci. Nyní následuje tabulka porovnávací funkce hlavice.

Funkčnost	Hlavice Danfoss	Vlastní řešení
Otevírání a zavírání ventilu	ANO	ANO
Samostatná regulace teplot	ANO	ANO
Odesílání dat na server/centrální jednotku	ANO	ANO
Displej s teplotou	ANO	NE
Možnost nastavení teploty na hlavici	ANO	NE
Frekvence kontroly aktuální teploty	1 minuta	Dynamická - 1-30 minut
Frekvence komunikace se serverem	30 minut	Dynamická - 5-30 minut

Tabulka 10 - Porovnání funkčnosti vlastní hlavice a hlavice Danfoss.

12.1.2 Testování mechanické odolnosti

Vzhledem k faktu, kdy je využito 3D tisku pro mechanické části, speciálně pro části silně namáhané, je třeba otestovat mechanickou odolnost těchto součástí. Hlavice obsahuje dvě mechanicky náročné součástky, které jsou kritické pro funkčnost. První část je matice pro našroubování hlavice na ventil topení, druhou částí je samotný převod z rotačního pohybu na lineární. Zde je třeba vyzkoumat mechanickou odolnost vytištěných převodů.

Samotná matice pro přišroubování hlavice k topení musí projít třemi testy. První test je na udržení hlavice na závitu. Nesmí docházet k přeskočení závitu matice na závitu u topení. Vzhledem ke kombinaci plast (Matice hlavice) a kov (závit na ventilu) je potřeba věnovat pozornost správnému našroubování, aby nedošlo ke stržení závitu na matici. V průběhu testování, byla hlavice nasazena na ventil celkem stokrát, a to jak ve stavu kdy byla hlavice zavřená, tedy šroubovalo se proti síle pružiny, tak i ve stavu otevřené hlavice, kdy se šroubuje bez

protisíly pružiny. Opotřebení lze vidět hlavně na prvním závitu na matici, ovšem toto opotřebení nemá na funkci matice zásadní vliv vzhledem k počtu dalších nepoškozených závitů.

Druhou částí testování je test odolnosti převodu. Pro účel tohoto testování byl využit nový ventil na topení a vlastní testovací program, provádějící otevírání a zavírání ventilu v cyklu, s přestávkami pro vychladnutí motoru po každém dvoustranném cyklu otevření a zavření. Při dosažení hodnoty 1000 cyklů byl test ukončen. Při pohledu běžným okem nelze poznat jakékoliv opotřebení převodů.

12.1.3 Porovnání spotřeby

Vzhledem k plánované možnosti využívat hlavici i jako napájené pomocí baterie, je zapotřebí porovnat spotřeby komerčního řešení oproti vlastnímu. Pro to slouží tabulka níže.

Stav	Hlavice Danfoss	Vlastní řešení
Spotřeba ve spánku	1-2 μ A	17 μ A
Spotřeba při komunikaci	80mA	56mA -170mA
Spotřeba v zatížení	0.4A	0.3A

Tabulka 11 - porovnání spotřeby vlastní hlavice a hlavice Danfoss.

V tabulce výše lze vidět vyšší spotřebu vlastního řešení v režimu spánku, i při vysílání. To je způsobeno použitou technologií, která byla vybrána z důvodu zásadní finanční úspory. Nikdy nelze říct přesnou výdrž jakéhokoliv zařízení, ať už se jedná o hlavici Danfoss nebo vlastní řešení, vždy záleží na četnosti otevírání a zavírání ventilu, kdy právě tato akce je největším konzumentem energie.

12.1.4 Cenové porovnání

Zásadním požadavkem na tvorbu zařízení byla jeho cena. Ideálně by se cena měla pohybovat na hranici poloviny ceny komerčního zařízení pro ospravedlnění některých ústupků ve funkčnosti při tvorbě zařízení. Tabulka níže shrnuje veškeré náklady na tvorbu hlavice.

Součástka	Cena
DC motorek	58 Kč
ESP8266-PRO	98 Kč
L9110S (H-můstek)	15 Kč
DS18B20 (teplotní čidlo)	38 Kč
Tlačítko	2 Kč
Držák baterie	22 Kč
Odporů 3x	Do 10 Kč
Nerezový šroub a matice	Do 10 Kč
Celkem	253 Kč

Tabulka 12 - Cenová kalkulace vlastního řešení hlavice.

Celková cena za součástky je 253 korun. Tato cena ovšem neobsahuje cenu za vytištěné součástky, která může cenu navýšit o zhruba 100 korun. Celková cena zařízení se může pohybovat do 350 korun. V porovnání s cenou termostatické hlavice Danfoss, která činí 1399 korun, je vlastní termostatická hlavice na čtvrtině její ceny. Díky tomu se hlavice stává velmi dostupným řešením.

12.2 Měřič spotřeby

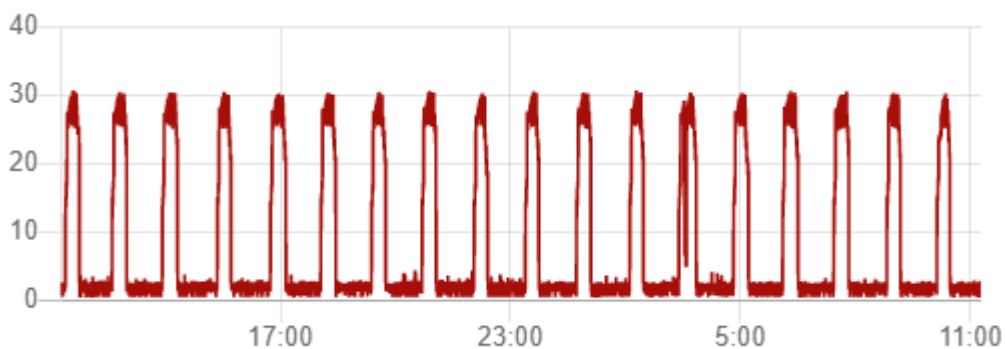
Dalším zařízením, které je třeba otestovat, je měřič spotřeby elektrické energie pro tepelné čerpadlo. Vzhledem k příliš vysokým cenám chytrých měřičů spotřeby, bylo zařízení porovnááno s měřením z klešťového měřiče proudu. Testování a porovnání probíhá v těchto kategoriích.

- Testování a porovnání funkčnosti
- Cenové porovnání

12.2.1 Testování a porovnání funkčnosti

Hlavní funkcí měřiče spotřeby je samotné měření, to probíhá na stejném principu jako u komerčního řešení. Tedy na základě využití proudového transformátoru. Hlavní funkcí je tedy získávání dat o příkonu a následném převedení do spotřeby a odesílání dat na Home Assistant.

Samotné zařízení je testováno reálným provozem již půl roku napojeném na Tepelném čerpadle. V průběhu testování byla odhalena pouze jedna zásadní chyba, a to v zaseknutí zařízení v případě výpadku signálu, kdy se zařízení již znovu nepřipojilo. Tento problém byl vyřešen implementací metody na opětovné připojení v případě ztráty připojení. Samotná přesnost měření byla kalibrována ručně za užití externího proudového měřiče.



Obrázek 20 - Příklad grafu vygenerovaném z měření proudu tekoucího do tepelného čerpadla za 24 hodin.

12.2.2 Cenové porovnání

Stejně jako u hlavice, i zde byl kladen důraz na nízkou cenu. V tabulce níže je soupis všech součástí potřebných pro tvorbu současné verze měřiče spotřeby.

Součástka	Cena
Proudový transformátor SCT-013-000	180 Kč
ESP8266-01	69 Kč
Arduino Nano	118 Kč
Stabilizátor napětí AMS1117 3.3V	12 Kč
Univerzální plošný spoj	15 Kč
Odporů 3x	Do 10 Kč
Kondenzátor	Do 10 Kč
Celkem	414 Kč

Tabulka 13 - Cenová kalkulace vlastního měřiče spotřeby.

Celková cena za součástky je 414 korun. Tato cena je ovšem v případě nákupu všech součástek v českých obchodech, v případě nákupu z ciziny je možné cenu srazit až na polovinu. V porovnání s cenou chytrých měřičů spotřeby, jako například *Legrand with Netatmo Modul Měření Energie S Cívkou*, která činí 2600 korun, je vlastní řešení na méně než pětině ceny. Díky tomu se měření proudu stává velmi dostupným řešením.

12.3 Detektor zatopení sklepa

Posledním z testovaných zařízení je detektor zatopení sklepa. Testování probíhá v těchto kategoriích.

- Testování funkčnosti a odolnosti
- Test spotřeby
- Cena

12.3.1 Testování funkčnosti a odolnosti

Hlavní funkčnost zařízení je detekce vody mezi elektrodami a následné odeslání informace na lokální instanci Home Assistantu. Vzhledem k umístění čidla ve sklepě, ve fixní výšce bylo testování provedeno vyvoláním umělého zatopení pouze formou ponoření čidla do nádoby s vodou. Čidlo vždy přesně detekovalo zatopení, i následné odčerpání vody, tedy vyschnutí.

Druhou částí je samotná odolnost elektrod. V případě, kdy by byly elektrody konstantně napájené, a tedy by konstantně detekovali, došlo by k velmi rychlé degradaci materiálu za pomoci elektrolýzy. Ovšem díky detekci pouze v intervalech 30 minut je životnost elektrod výrazně navýšena. Do konce testu nebylo zpozorované žádné výrazné narušení struktury materiálu.

12.3.2 Test spotřeby

Ačkoliv je zařízení v tomto konkrétním případě napájeno externě, je zamýšleno i jako čidlo napájené z baterie. Proto již od začátku bylo vyvíjeno s kladeným důrazem na nízkou spotřebu. Vzhledem k funkčnosti čidla, kdy se kontroluje zatopení pouze jednou za půl hodiny, je dosaženo velmi dobrého poměru mezi časem spánku a časem vysílání čidla. V tabulce níže je zapsaná spotřeba následovaná teoretickým výpočtem výdrže na 2500 mAh baterii.

Spotřeba ve spánku	17 μ A
Spotřeba při komunikaci	56mA -170mA

Tabulka 14 - Porovnání spotřeby.

Interval pro komunikaci a měření je pět vteřin, tedy za tuto dobu spotřebovaná energie 236 μ Ah, spotřeba ve spánku je 8 μ Ah za půl hodiny. Sečtením a vynásobením lze dostat konečnou spotřebu za hodinu, tedy 488 μ Ah což je 0.488mAh. vydělením celkově kapacity baterie spotřebou lze získat orientační výdrž, tedy 5122 hodin, v přepočtu přes 213 dnů . V případě zvětšení intervalu měření na jednu hodinu je výdrž až 411 dnů. Výpočet počítá s nejhorsím možným scénářem, tedy s největší možnou spotřebou.

12.3.3 Cena

Stejně jako všechna předchozí zařízení, i detektor zatopení má za cíl být co nejméně nákladný. Vzhledem k jednoduchosti celého detektoru bylo cílem sestavit zařízení s maximální cenou do 250 korun českých. Níže je tabulka užitých součástí i s celkovou cenou zařízení.

Součástka	Cena
ESP8266	148Kč
Sonda vlhkosti	38 Kč
Stabilizátor napětí AMS1117 3.3V (v případě externího napájení)	12 Kč
Univerzální plošný spoj	15 Kč
Celkem	213 Kč

Tabulka 15 - Cenová kalkulace vlastního řešení detektoru zatopení.

12.4 Všeobecné porovnání

Tato podkapitola je zaměřená na porovnání vlastností, které mají všechna zařízení společné, jako je univerzálnost, možnost přizpůsobení a náročnost instalace a tvorby.

12.4.1 Porovnání univerzálnosti

Důležitým prvkem každého chytrého zařízení v domácnosti je možnost napojení na různé systémy. Porovnávaná komerční zařízení na technologii Z-Wave jsou velmi univerzální. Komerční zařízení lze integrovat v různých systémech chytré domácnosti, ať už je to Home Assistant, Domoticz nebo OpenHAB. Výhodou komerčních zařízení je i možnost integrace s vlastními řešeními výrobců a jejich aplikací.

Vlastní nízkonákladové řešení je založeno na užití se systémem Home Assistant, ale je možné řešení upravit pro užití i v jiných opensource systémech. Ovšem pro užití s řešením od výrobců současné verze nízkonákladových zařízení nelze využít. Nevýhodou současného řešení je nutnost přepisovat přímo kód chytrého zařízení pro funkčnost s jiným systémem.

12.4.2 Licence/možnost přizpůsobení (open-source)

Velikou nevýhodou jakéhokoliv komerčního zařízení je jeho uzavřenost. V případě užití komunikační technologie Z-Wave je i nutnost zařízení licencovat. To ovšem u vlastního řešení není nutnost. Celé vlastní řešení součástí této práce je open-source, tedy kód je volně dostupný a kdokoli si může zařízení upravit podle sebe, nebo inspirovat se a vytvořit úplně nové zařízení. To je i jedním z cílů této práce. Vytvořit nízkonákladové zařízení a ukázat, že je zde možnost pro chytrou domácnost bez užití nákladných komerčních zařízení.

Výhoda open-source je možnost upravit zařízení na jakýkoliv systém. V případě potřeby je tedy možné zařízení upravit pro integraci v jiném systému, než je Home Assistant, přeprogramovat a využívat jinou komunikační technologii, nebo postavit i na kompletně jiné platformě.

12.4.3 Náročnost instalace / tvorby

Výhodou komerčních zařízení je jejich jednoduchost. Zařízení se nemusí složitě sestavovat, není potřeba znalosti programovacích jazyků ani elektrotechniky, zařízení stačí jednoduše koupit a podle návodu zapojit do sítě. V případě užití Home Assistantu je i samotná integrace poté velmi jednoduchá vzhledem k automatické detekci nových zařízení.

Naopak u vlastního řešení je zapotřebí zařízení nejprve sestavit. Vzhledem k náročnosti to může být pro řadu uživatelů velmi odrazující požadavek. Další požadavkem je seznámit se s technologií a programovacím jazykem. Každé zařízení se musí ručně integrovat do Home Assistantu. Ovšem tento projekt open-source domácnosti není primárně cílen na neznanou společnost, ale cílí spíše na elektro kutily orientující se v dané problematice.

13 Závěr

Práce si kladla za cíl vytvořit nízkonákladový ekosystém chytrých zařízení pro domácnost. Celý ekosystém se skládal ze třech dílčích úkolů, kdy prvním bylo samotné nasazení open source softwaru, konkrétně Home Assistantu, umožňující řízení dalších zařízení. Systém je reálně nasazen v domácnosti a byl otestován každodenním reálným použitím. Pro otestování bylo integrováno 5 zakoupených chytrých zařízení pracujících na technologii Z-Wave, mezi které patří chytrá hlavice, senzor teploty a světla, senzor otevření dveří, nebo také chytrá zásuvka či klíčenka s tlačítkem. Po otestování funkčnosti byl zahájen vývoj vlastních nových nízkonákladových open source zařízení.

Druhým úkolem bylo umožnit systému komunikovat se vzdáleným serverem a také umožnit ovládat Home Assistant mimo vnitřní domácí síť. Vlastní řešení vzdáleného ovládání bylo ideově založeno na řešení z projektu Smart Grid, kdy bylo užito stejného protokolu, ale k samotné komunikaci s Home Assistantem byl naprogramován software sloužící jako lokální služba. Díky tomu jsem se vyhnul problémům z řešení v projektu Smart Grid, kdy problémy jako nemožnost automaticky přidávat nová zařízení, nebo nutnost konfigurace každého nového chytrého zařízení v lokální síti pro umožnění vzdáleného ovládání, odpadly. Díky tomu se systém pro vzdálené ovládání přiblížil k možnosti reálného využití a k možnosti dalšího rozvoje, například směrem analýzy dat na serverové straně.

Posledním, ale neméně důležitým úkolem, bylo vytvořit tři nová nízkonákladová chytrá zařízení. Všechny tři zařízení byly vybrány z praktického hlediska, pro užití ve vlastní domácnosti. Jedná se o chytrou hlavici na ventil u topení, měřič spotřeby tepelného čerpadla a detektor zatopení sklepa. U chytré hlavice bylo vyřešeno hned několik kritických problémů, jako způsob zavírání ventilu s trnem, navržení součástky a následné vytisknutí, nebo například řešení poměru mezi cenou a výkonem. U měřiče spotřeby bylo zapotřebí najít způsob, jak kvalitně měřit spotřebu bez zásahu do vodičů, to hlavně z důvodu užití na kritické infrastruktuře (Otopná soustava domu) kdyby došlo k poruše měřiče, aby nedošlo k rozepnutí obvodu. U posledního zařízení se kladl důraz na jednoduchost, jedná se o velmi jednoduché ale spolehlivé čidlo detekující vodu ve sklepě.

Všechna zařízení byla nasazena v domácnosti a otestována reálným provozem. Zařízení byla sestrojena s důrazem na nízké pořizovací náklady, to se také povedlo, cena hlavice vyšla na čtvrtinu ceny kupované hlavice a cena měřiče spotřeby vyšla dokonce pouze na pětinu ceny kupovaného. Nízká cena je bohužel vykoupená nutností si zařízení sestrojít, tedy tvorba vyžaduje určitou zručnost. Celé nasazení si lze prohlédnout v příloze Č.5 – *Reálné nasazení*, obsahující fotodokumentaci a schéma chytré domácnosti.

Závěrem je nutné říct, že úkol sestrojít tři chytrá zařízení, není konečným. Jedná se o celý ekosystém možných nízkonákladových open-source zařízení postavených na stejné technologii. Celá práce je úvodem do nového ekosystému, který lze postavit na stejné technologii a zdokumentovat jako celek. Za tím účelem jsem založil webovou stránku na doméně „*opensourcehome.cz*“, kde postupem času dokumentuji další vývoj, vytvářím nová zařízení a vylepšuji stávající řešení.⁸ Do budoucna je tu možnost systém dál rozvíjet o nespočet možných dalších chytrých zařízení, jako například chytrá zásuvka, teploměry, meteostanice, nebo bezpečnostní kamera.

⁸ Modely pro 3D tisk jsou také k dispozici jako kolekce na webu <https://www.thingiverse.com/opensourcehome/collections/project-opensourcemarthome>

Vysvětlení zkratk

IP	- Internet Protocol
DNS	- Domain Name System
IoT	- Internet of Things neboli internet věcí
MQTT	- Message Queuing Telemetry Transport - protokol pro komunikaci
HUB	- centrální bod sdružující například komunikaci
PC	- počítač
HW	- hardware
SW	- software
API	- rozhraní pro určitý program
ACL	- Access control list, list upřesňující pokročilá pravidla pro zabezpečení
SMD	- Surface mount device neboli součástky pájené přímo na povrch
A	- Ampér - jednotka pro elektrický proud – značení v textu I
V	- Volt - jednotka pro elektrické napětí – značení U
F	- Farad - jednotka elektrické kapacity – značení C
Ω	- Ohm – jednotka odporu – značeno R
W	- Watt – jednotka výkonu – značeno P

Seznam obrázků

Obrázek 1 - Příklad vizualizace dat.....	14
Obrázek 2 - Architektura nového řešení.....	19
Obrázek 3 - Raspberry Pi bez modulu Z-Wave (vlevo) a s modulem (vpravo).....	23
Obrázek 4 - ESP 8266 LUA NodeMcu. (zdroj: laskarduino.cz).....	30
Obrázek 5 - Zapojení ESP8266-01 s programátorem. (zdroj: hwlab.cz).....	31
Obrázek 6 - Varianty modulu ESP 32. (zdroj: laskarduino.cz).....	32
Obrázek 7 - Foto ventilu s trnem.....	37
Obrázek 8 - Foto mechaniky rozebrané komerční hlavice.....	38
Obrázek 9 - DC motor se šnekovým převodem. (zdroj: laskarduino.cz).....	39
Obrázek 10 - Průřez modelem matice na topení.....	42
Obrázek 11 - Schéma sepnutí H-Můstku. (zdroj: Cyril BUTTAY).....	43
Obrázek 12 - Mechanické součástky pro hlavici.....	45
Obrázek 13 - Schéma zapojení hlavice.....	46
Obrázek 14 - Hotová hlavice ve variantě s MicroUSB napájením.....	47
Obrázek 15 - Schéma obvodu s proudovým transformátorem. (zdroj: openenergymonitor.org).....	51
Obrázek 16 - Schéma zapojení proudového transformátoru s užitím zátěžového rezistoru R1 a s děličem.....	52
Obrázek 17 - Hotový měřič spotřeby na vlastním plošném spoji.....	53
Obrázek 18 - Sonda půdní vlhkosti. (zdroj: laskarduino.cz).....	56
Obrázek 19 - Schéma zapojení detektoru zatopení sklepa.....	57
Obrázek 20 - Příklad grafu vygenerovaném z měření proudu tekoucího do tepelného čerpadla za 24 hodin.....	63

Seznam tabulek

Tabulka 1 - Funkční a nefunkční požadavky.	3
Tabulka 2 - Analýza spotřeby.....	10
Tabulka 3 - Konfigurační soubory.....	13
Tabulka 4 - Funkční a nefunkční požadavky systému.	17
Tabulka 5 - Požadavky na systém.	18
Tabulka 6 - Zhodnocení řešení pomocí Wi-Fi.....	29
Tabulka 7 - Funkční a nefunkční požadavky chytré hlavice.	36
Tabulka 8 - Funkční a nefunkční požadavky měřiče spotřeby.....	48
Tabulka 9 - Funkční a nefunkční požadavky detektoru zatopení sklepa.....	55
Tabulka 10 - Porovnání funkčnosti vlastní hlavice a hlavice Danfoss.....	60
Tabulka 11 - porovnání spotřeby vlastní hlavice a hlavice Danfoss.	61
Tabulka 12 - Cenová kalkulace vlastního řešení hlavice.	62
Tabulka 13 - Cenová kalkulace vlastního měřiče spotřeby.....	64
Tabulka 14 - Porovnání spotřeby.....	65
Tabulka 15 - Cenová kalkulace vlastního řešení detektoru zatopení.	66

Seznam zdrojů

- [1] *Slovník pojmů vodafone* [online]. [cit. 2021-02-11]. Dostupné z: <https://www.vodafone.cz/uzitecne-odkazy/slovník-pojmu/smart-home-chytra-domacnost/>
- [2] *STOLNÍ 3D TISKÁRNA: JAK VLASTNĚ FUNGUJE A KTEROU SI VYBRAT* [online]. Praha, 2017, 18. 1. 2017 [cit. 2021-2-12]. Dostupné z: <https://www.3dees.cz/blog/3d-tiskarna-jak-vlastne-funguje-a-kerou-si-vybrat>
- [3] *Co je Home Assistant* [online]. 26.5.2020 [cit. 2021-02-11]. Dostupné z: <https://tatageek.blog/2020/05/26/co-je-home-assistant/>
- [4] ARROWS, Kevin. *Home Assistant Vs OpenHAB* [online]. 2020 [cit. 2021-2-26]. Dostupné z: <https://appuals.com/home-assistant-vs-openhab/>
- [5] *Comparison between Domoticz and Home Assistant in 2019* [online]. 28.1.2019 [cit. 2021-03-11]. Dostupné z: <https://gadget-freakz.com/domoticz-vs-home-assistant/>
- [6] GEERLING, Jeff. *Power Consumption Benchmarks* [online]. [cit. 2021-03-11]. Dostupné z: <https://www.pidramble.com/wiki/benchmarks/power-consumption>
- [7] Dokumentace Home Assistant. *HomeAssistant.io* [online]. [cit. 2021-1-26]. Dostupné z: <https://www.home-assistant.io/docs/>
- [8] IoT Analytics - ThingSpeak Internet of Things [online]. The MathWorks, ©2021 [cit. 2021-03-10]. Dostupné z: <https://thingspeak.com/>
- [9] Home Assistant Cloud. *Nabucasa.com* [online]. [cit. 2021-3-10]. Dostupné z: <https://www.nabucasa.com/>

- [10] Raspberry Pi models comparison. *Socialcompare.com* [online]. [cit. 2021-3-14]. Dostupné z: <https://socialcompare.com/en/comparison/raspberrypi-models-comparison>
- [11] Home Assistant Blue! *Home Assistant.io* [online]. [cit. 2021-3-16]. Dostupné z: <https://www.home-assistant.io/blue/>
- [12] How To Use Apache HTTP Server As Reverse-Proxy Using mod_proxy Extension. *Digitalocean.com* [online]. [cit. 2021-3-20]. Dostupné z: https://www.digitalocean.com/community/tutorials/how-to-use-apache-http-server-as-reverse-proxy-using-mod_proxy-extension
- [13] Dokumentace projektu SMARTGRID. *Dev.azure.com* [online]. [cit. 2021-3-26]. Dostupné z: https://dev.azure.com/uai-jcu/SmartGrid%20-%20Heating%20system%20control/_wiki/wikis/SmartGrid---Heating-system-control.wiki/4/Home
- [14] Dokumentace mosquitto. *Mosquitto.org* [online]. [cit. 2021-3-26]. Dostupné z: <https://mosquitto.org/documentation/>
- [15] Dokumentace Home Assistant - integrace z-wave. *Home-assistant.io* [online]. [cit. 2021-3-27]. Dostupné z: <https://www.home-assistant.io/docs/z-wave/device-specific/#rasberry-board>
- [16] TEEL, John. Comparison of Wireless Technologies. *Predictabledesigns.com* [online]. [cit. 2021-3-27]. Dostupné z: https://predictabledesigns.com/wireless_technologies_bluetooth_wifi_zigbee_gsm_lte_lora_nb-iot_lte-m/
- [17] PRETTY, Brad. HOME AUTOMATION: Wi-Fi, Zigbee, and Z-Wave: What's the difference? *Getmysa.com* [online]. [cit. 2021-2-20]. Dostupné z: <https://getmysa.com/blogs/home-automation/wifi-zigbee-zwave>

[18] Hrátky s WiFi čipem ESP8266. *Hwlab.cz* [online]. [cit. 2021-1-26]. Dostupné z: <https://hwlab.cz/2016/12/01/hratky-s-wifi-cipem-esp8266-1-dil-seznameni/>

[19] CT Sensors: Interfacing with an Arduino. *Openenergymonitor.org* [online]. [cit. 2021-3-26]. Dostupné z: <https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/interface-with-arduino>

Seznam příloh

Příloha č.1 – Software

Příloha č.2 – Uživatelská Dokumentace

Příloha č.3 – Zdrojová data 3D modelů (CAD)

Příloha č.4 – Schémata elektroniky

Příloha č.5 – Reálné nasazení