



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

SYSTÉM CHYTRÝCH OKENNÍCH ZÁVĚSŮ

INTELLIGENT WINDOW SHUTTER SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MAREK FIALA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2022

Zadání bakalářské práce



Student: **Fiala Marek**
Program: Informační technologie
Název: **Systém chytrých okenních závěsů
Intelligent Window Shutter System**
Kategorie: Vestavěné systémy

Zadání:

1. Proveďte rešerši dostupných řešení z oblasti automatizovaného ovládání stínicí techniky a asistentů tzv. chytré domácnosti.
2. Analyzujte požadavky na systém chytrých okenních závěsů. Navrhněte a zdůvodněte koncepci systému, definujte potřebné komponenty.
3. Navrhněte a realizujte funkční vzorek zařízení chytrých okenních závěsů na základě požadavků definovaných v bodě 2 zadání.
4. Vytvořte programové vybavení pro samotné zařízení, možnost jeho vzdáleného monitorování (např. z osobního počítače) a připojení zařízení k systémům domácí automatizace.
5. Vhodným způsobem demonstруйте vlastnosti vytvořeného zařízení a ověřte jeho plnou funkčnost dle požadavků z bodu 2 zadání.
6. Zhodnoťte dosažené výsledky a pokuste se navrhnout případná vylepšení či další směřování vývoje.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 až 3 zadání.

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

Vedoucí práce: **Šimek Václav, Ing.**

Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 29. října 2021

Abstrakt

Cílem této práce je vymyslet a sestavit funkční prototyp systému chytrých okenních závěsů za použití mikročipu ESP32 od firmy Espressif. Čtenář se dozví, co je to domácí automatizace, jaké systémy automatizovaných závěsů již trh nabízí a najde zde popis komponent a periférií použitých v systému. Práce nabízí kompletní postup od návrhu, jak systém může fungovat, až po fyzické sestavení. Popis zahrnuje schéma výsledného zapojení, tvorbu desky plošných spojů, 3D model systému, programové řešení a ovládání závěsů pomocí systému domácí automatizace Home assistant.

Abstract

The aim of this thesis is to develop and build a functional prototype of an intelligent window shutter system using Espressif's ESP32 microchip. The reader finds out what is home automation, what kinds of automated window shutters the current market already offers, and a description of what components and peripherals are used in the system. The thesis offers a complete process from design to physical construction. The description includes schematics of the final wiring, modeling of a printed circuit board, 3D model of the system, software solution, and user control of the system using a Home assistant automation system.

Klíčová slova

okenní závěsy, inteligentní systém, chytrá domácnost, systém domácí automatizace, ESP32, senzor, deska plošných spojů, C++

Keywords

window curtains, window shutter, smart system, smart home, home automation system, ESP32, sensor, printed circuit board, C++

Citace

FIALA, Marek. *Systém chytrých okenních závěsů*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

System chytrých okenních závěsů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Václava Šimka. Další informace mi poskytli kolegové z firmy Espressif Systems (Czech) s.r.o. Ing. Radim Karniš a Mgr. Martin Vychodil. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Marek Fiala
10. května 2022

Poděkování

Především bych rád poděkoval Ing. Václavu Šimkovi za konzultace, poradenství a vedení při vypracovávání této práce. Dále bych chtěl poděkovat Ing. Radimu Karnišovi, který mi byl nápomocen především s výběrem částí systému a uvedením do elektroinženýrství. V neposlední řadě firmě Espressif Systems (Czech) s.r.o. za poskytnutí mikročipu a místa k tvoření a zkonstruování této práce.

Obsah

1	Úvod	3
2	Možnosti domácí automatizace	5
2.1	Chytrá domácnost	5
2.2	Internet věcí	5
3	Přehled dostupných řešení	7
3.1	Nuvii	7
3.2	Somfy	7
3.3	SwitchBot Curtain	8
4	Návrh koncepce systému	10
4.1	Elektronické komponenty	10
4.2	Mikročip ESP32	19
4.3	Režimy systému okenních závěsů	20
4.4	Komunikace se systémem	20
5	Návrh obvodového řešení	22
5.1	Propojení jednotlivých komponent	22
5.2	Motiv desky plošných spojů	28
5.3	Schéma zapojení	29
5.4	3D model osazené desky plošných spojů	31
6	Mechanické řešení	32
6.1	Mechanické komponenty	32
6.2	Konstruování 3D modelu	33
6.3	Demonstrační prototyp	35
6.4	Tisk na 3D tiskárně	37
7	Programové řešení	38
7.1	Zpracování sensorových dat	38
7.2	Režimy systému okenních závěsů	38
7.3	Pohyb motoru	39
7.4	Komunikace klient - server	39
7.5	Běh programu	40
8	Zkonstruování prototypu a testování	41
9	Závěr	44

Literatura	46
A TMC2209 schema	47
B Dokumentace průběhu tvorby	49
C Zprovoznění systému	51

Kapitola 1

Úvod

Inteligentní systémy jsou technologicky vyspělé stroje, které vnímají svět kolem sebe a reagují na něj. Inteligentní systémy mohou mít mnoho podob, od automatizovaných vysavačů, sekaček na trávu nebo například závěsů, přes programy pro rozpoznávání obličejů až po osobní automatické nákupní návrhy v internetových obchodech, které uživateli dokáží opravdu nabídnout, co by mohl potřebovat.

Dříve inteligentní systémy - roboti vykonávali za stejných podmínek stále jednu činnost, která se navíc pravidelně opakovala. Dnes je robot považován za autonomní systém, který dokáže vnímat prostředí ve kterém se nachází a na základě získaných podmětů činit rozhodnutí, pro dosažení požadovaných cílů.

Klíčovými faktory, které přispěly k tomuto postupnému vývoji inteligentních systémů, jsou exponenciální růst rychlosti procesoru a kapacity paměti, stejně jako algoritmický pokrok. S tímto postupným vývojem se inteligentní systémy stávají stále dostupnější a levnější.

Cílem této práce bude vytvořit inteligentní systém okenních závěsů reagující na světlo z okolí a přizpůsobit tak polohu závěsu.

V kapitole 2 najdeme shrnutí, co to systém domácí automatizace vlastně je a proč se domácnost s takovým systémem nazývá chytrá. Kapitola 3 se věnuje dostupným řešením, která již aktuální trh nabízí.

Tvorba takového systému je vcelku náročný proces, který se skládá z jednotlivých kroků s určitou posloupností. Nejde rovnou vymyslet koncový produkt. Tvůrce neví, jaké části bude přesně používat, natož jak bude výsledek vypadat. Zpočátku je potřeba vytvořit koncept, jak by produkt mohl fungovat jak z fyzické, tak programové části, případně provést analýzu trhu, jaké nabízí možnosti jednotlivých komponent. To najdeme v kapitole 4 s popisem jednotlivých režimů ve kterých bude systém fungovat a možnostmi ovládní systému uživatelem.

Po úspěšném vybrání vhodných kandidátů je třeba zjistit, jak jednotlivé periferie s řídicí jednotkou komunikují, případně co potřebují ke své funkčnosti. To vše je popsáno v kapitole 5 i s návrhem výsledného zapojení, jak by mohla elektronická část systému vypadat.

Kapitola 6 popisuje princip mechanického fungování pohybu závěsů, modelování jednotlivých částí výsledného produktu a jejich tisk na 3D tiskárně.

Popis ovládní systému a principu komunikace klienta se serverem domácí automatizace je vysvětleno v kapitole 7, kde se nachází i popis jednotlivých knihoven použitých při tvorbě programu a vysvětlení, jak celý systém funguje z programátorského hlediska.

Po výsledném zkoumání a vymyšlení návrhu je potřeba systém sestrojít do celku, který je potřeba následně otestovat. Výsledné zhodnocení úsilí vynaložené při tvorbě této práce je popsáno v kapitole 8.

Kapitola 2

Možnosti domácí automatizace

Systém domácí automatizace je systém, který je ovládán pomocí chytrého zařízení. Dokáže ovládat domácí spotřebiče jako je světlo, ventilátory, podmínky ovzduší nebo inteligentní bezpečnostní zámek. K ovládání věcí na dálku bývá nejčastěji použita technologie bluetooth nebo Wi-Fi.

2.1 Chytrá domácnost

Žijeme v moderní generaci, kde jsou chytré a inteligentní systémy pomalu nezbytné. Používáme je, abychom si zlepšili a podstatně ulehčili život. S chytrou domácností můžeme dělat mnoho věcí rychleji, lépe a přesněji.

Elektronickým prostředím je jakékoli prostředí, které se skládá ze spotřebičů, jako jsou ventilátory, televizní přijímače, klimatizace, motory, topení, osvětlovací systémy atd... Vzdáleně přístupné prostředí je prostředí, ve kterém lze ke každému spotřebiči přistupovat vzdáleně a ovládat jej pomocí softwarového rozhraní, jako je například mobilní nebo webová aplikace. Kombinace těchto dvou prostředí vytváří základní pilíře chytré domácnosti.

Chytrá domácnost může v mnoha ohledech pomoci optimalizovat procesy či ušetřit náklady na provoz domácnosti díky kompletnímu přehledu činností domácího prostředí. V případě, že uživatel odejde z domu a zapomene zhasnout světlo, může tuto akci udělat aniž by se vracel domů nebo se musel smířit s tím, že se doma bude svítit dokud se nevrátí. Chytrá domácnost poskytuje náhled na využívání energie, čímž může pomoci stát se energeticky úspornější a dbát ohled na ekologické faktory. Detektory otevřených oken mohou automaticky vypnout topení, pokud všechna okna v místnosti nejsou zavřená. Chytrá domácnost navíc může sloužit i jako bezpečnostní ochrana. Příkladem může být kamerový systém poskytující uživateli živý přenos jeho domácnosti v případě když není doma.

Ještě efektivnější se chytrá domácnost stává, pokud všechny jednotlivé prvky jsou ovládány z jednoho místa. K tomu slouží systém domácí automatizace. Navíc mnoho chytrých domácích systémů je kompatibilní s hlasovými asistenty, jako je Amazon Alexa nebo Google Home, což ovládání chytré domácnosti ještě zjednodušuje.

2.2 Internet věcí

Internet věcí (*angl. IoT - Internet of Things*) [2][3]

Představme si svět, kde miliardy objektů mohou komunikovat a sdílet informace, všechny propojené přes veřejné nebo soukromé sítě internetového protokolu. Tyto propojené objekty

mají data pravidelně shromažďována, analyzována a používána k iniciování akce, což poskytuje bohatství inteligence pro plánování, řízení a rozhodování. To je svět internetu věcí.

Internet není jen síť počítačů, ale vyvinul se v síť zařízení všech typů a velikostí - vozidla, chytré telefony, domácí spotřebiče, hračky, fotoaparáty, lékařské nástroje, průmyslové systémy, budovy - vše propojené, vše komunikující na základě stanovených protokolů s cílem dosáhnout inteligentní organizace, lokalizace, dohledávání, bezpečnosti, ovládání a dokonce sledování v reálném čase. To vše jsou části a možnosti Internetu věcí.

Internet věcí je pojem a paradigma, které uvažuje o všudypřítomné přítomnosti nejrůznějších věcí či objektů v prostředí, které jsou za použití internetového připojení a unikátních adresovacích schémat schopny vzájemně komunikovat a spolupracovat s jinými věcmi / objekty, aby vytvořily nové aplikace / služby a dosáhly společných cílů. IoT je obecná představa věcí, zejména každodenních objektů, které jsou čitelné, rozpoznatelné a adresovatelné prostřednictvím zařízení na snímání informací a kontrolovatelné prostřednictvím internetu, bez ohledu na komunikační prostředky. Těmito prostředky mohou být identifikační možnosti na rádiové frekvenci, bezdrátové síti nebo kabelové internetové síti.

Objekty se stávají rozpoznatelnými a inteligentními tím, že činí nebo umožňují kontextová rozhodnutí díky možnosti sdílet informace mezi sebou. Mohou přistupovat k informacím, které byly shromážděny jinými zařízeními v síti, nebo mohou být součástí komplexních služeb. Tato možnost rozhodování a sdílení je možná především díky cloud computingu - *přenesení výpočetní kapacity a ukládání dat na vzdálené servery dostupné po internetu, jelikož prvky IoT často disponují omezenou výpočetní i ukládací kapacitou* a přechodem internetu na IPv6 s téměř neomezenou adresní kapacitou. Cílem internetu věcí je umožnit připojení věcí kdykoliv a kdekoliv, s čímkoli a kýmkoliv, ideálně s využitím jakékoli cesty / sítě.

Internet věcí i chytrá domácnost jsou důležité pojmy při tvoření této práce. Mým úkolem totiž je vytvořit systém, který bude právě jedním prvkem chytré domácnosti a jednou věcí v internetu věcí.

Kapitola 3

Přehled dostupných řešení

V době psaní této práce se chytrá domácnost začíná stávat přirozenější součástí našich životů. S budujícími se novými domy či byty téměř každý aspekt života, kdy technologie vstoupila do domácího prostoru (žárovky, myčky nádobí a tak dále), zažil zavedení alternativy chytré domácnosti. Není tedy divu, že podobná řešení jako se v této práci budu snažit vytvořit již existují. Vybral jsem si zde na ukázkou pár komerčních řešení, které si lze jednoduše pořídit na internetu.

3.1 Nuvii

Elektronická garnýž Nuvii¹ je samostatně prodejnou částí. Systém se skládá z kolejnice, motoru a ovladače na systém. Jednou z hlavních výhod je pohon motoru na baterii, pro uživatele, kteří se systémem elektronických závěsů při stavbě či rekonstrukci nepočítali, a nemají tudíž vyvedené elektrické zásuvky poblíž okna. V opačném případě to může být považováno i za nevýhodu, jelikož hlídat kapacitu baterie je omezující oproti systému beroucí si energii z elektrické sítě. Uváděná provozní doba do potřebné výměny nebo znovu nabití baterií je 4 - 6 měsíců v závislosti na váze závěsů.

Nuvii nelze napojit na domácího asistenta, ani s ním nijak komunikovat přes Wi-Fi síť. Jediným možným způsobem jak systém ovládat je jeden z přiložených ovladačů (dálkový ovladač, ovládací jednotka pro upevnění na zeď). Další možnost, kterou systém nabízí je manuální operace se závěsy, tu však nepokládám za podstatnou v souvislosti s automatizací, což je hlavním podmětem této práce.

Cena za soustavu s elektrickou kolejnicí v délce 1 - 4,2 metry činí cca 7000,-Kč.

3.2 Somfy

Somfy² je firma zabývající se kompletně chytrou domácností. V jejich nabídce najdeme mimo chytrých závěsů také venkovní rolety, terasové markýzy či pergoly, vjezdové brány, garážová vrata a z nepohyblivých částí chytrý zámek dveří, vytápění, osvětlení nebo například alarmy či kamery.

U Somfy není možné zakoupit pouze produkt mechanických žaluzií, firma nabízí jen komplexní řešení od změření okna přes dodání produktu až po jeho instalaci u svých smluvních

¹<https://www.nuvii.cz/Nuvii-Elektricka-garnyz-kolejnicova-elektricke-zavesy-a-zaclony-1m-4-2m-s-bateriovym-pohonem-sada-d2051.htm>

²<https://www.somfy.cz>

partnerů. Jelikož jde o komplexní systém chytré domácnosti, nabízí Somfy nejširší škálu ovládání žaluzií.

- Fyzickým ovladačem dodaným se systémem
- Přes mobilní aplikaci TaHoma³
- Pomocí hlasového asistenta - podpora: Alexa⁴, Google Assistant⁵, Apple HomeKit⁶

Vzhledem ke komplexní nabídce firmy Somfy, bez poptání produktu nebylo možné zjistit cenu systému.

3.3 SwitchBot Curtain

Zajímavým přístupem, který je v dnešní době k sehnání je zařízení od firmy SwitchBot⁷. SwitchBot se nezaměřuje na vyvíjení celých systémů, ale pouze na konstruování jednotlivých IoT zařízení, které plní jednu funkci. Jejich vývoj začal s roboty, jejich úkolem bylo mačkat určité tlačítko - například vypínač. Zkrátka fyzicky pouze malý pohyb nožičkou dopředu a zpět. V následném rozšiřování firmy SwitchBot vyvinuli SwitchBot Curtain.

Jedná se o robota, který se zavěsí na kolejnici / tyč závěsu, a následně po tomto upevnění jezdí, díky čemuž posouvá i celý závěs.

Oproti předchozím dvěma řešením, je u SwitchBot Curtain inovativní vlastností zabudovaný světelný senzor. Ten po propojení se systémem domácí automatizace dokáže vyhodnocovat kontextovou situaci, a tím dokáže například šetřit při vytápění místnosti otevřením závěsů pro rychlejší výhřev slunečními paprsky.

Podle tvaru uchycení robota na závěs jsou na výběr možnosti:



Obrázek 3.1: Možnosti uchycení robota SwitchBot Curtain. (Převzato z webu [8])

³<https://www.somfy.cz/produkty/ovladace-cidla-a-chytra-domacnost/chytra-domacnost/ovladejte-svou-chytrou-domacnost-pomoci-aplikace-tahoma>

⁷<https://www.switch-bot.com>

⁸<https://www.switch-bot.com/products/switchbot-curtain>

Tak jako systém Nuvii, i SwitchBot Curtain funguje na baterky a uváděná provozní doba je až 8 měsíců.

S robotem je možné komunikovat pomocí aplikace SwitchBot App⁹, nebo hlasových asistentů Alexa⁴, Google Assistant⁵, Apple HomeKit⁶, Clova¹⁰

Řešení je to určitě méně estetické oproti předešle zmiňovaným, avšak cena jednoho robota se pohybuje okolo 2000,-Kč, což něj dělá zajímavé řešení.

⁹<https://apps.apple.com/us/app/switchbot/id1087374760>

⁴https://cs.wikipedia.org/wiki/Amazon_Alexa

⁵<https://assistant.google.com>

⁶<https://support.apple.com/cs-cz/HT204893>

¹⁰<https://clova.ai/ko>

Kapitola 4

Návrh koncepce systému

Základ celého návrhu systému bylo vytyčit si základní věci, které má systém umět. Samozřejmě první věcí byl pohyb. Závěsy se budou otevírat a zavírat, takže zapotřebí bude potřeba nějaká hnací síla - motor. Systém bude umět reagovat na podmínky z vnějšího okolí. V tomto případě sluneční záření. Další nedílnou součástí mého systému tedy musel být světelný senzor. Dále jsem věděl, že pro chytrý systém bude potřeba řídicí jednotka - mozek všech operací - vyhodnocující jestli je ten správný čas závěsy zatáhnout, jak moc je zatáhnout, a také bude potřeba se systémem nějak komunikovat.

4.1 Elektronické komponenty

Motor

Pro výběr motoru do mého systému přicházejí v úvahu k porovnání dva kandidáti - Klasický stejnosměrný motor a krokový motor. Oba motory jsou poháněny stejnosměrným elektrickým proudem a jak jejich název napovídá, používají stejnosměrný proud pro přeměnu na mechanickou energii. Stejnosměrné motory vycházejí z principu periodického přepínání polarity napájecího stejnosměrného proudu.

Stejnosměrný motor

Stejnosměrný motor [9], ve smyslu kartáčový stejnosměrný motor (DC). K přepínání polarity dochází po každém otočení motoru o 180 stupňů. Polohu motoru tedy není možné přesně určit a zastavit přesně v požadovaný moment. Naopak výhodou kartáčového DC motoru je velmi vysoké zrychlení. Úprava vstupního napětí je přímo úměrná změně otáček motoru a obrácení polarity vodičů způsobí, že stejnosměrný motor obrátí směr. Kartáčované DC motory se používají v aplikacích, kde je třeba regulovat otáčky a točivý moment, například v elektrickém posilovači řízení, navijákových motorech a ovladačích zámku dveří.

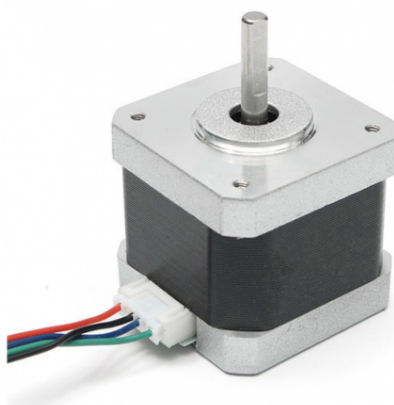
Krokový motor

Krokový motor [4] je typ DC motoru, který se pohybuje v diskrétních krocích, neboli „fázích“, což umožňuje přesnou kontrolu nad polohováním. K pohybu tohoto motoru je vždy třeba řídicí elektronika – ovladač krokového motoru, který pomáhá synchronizovat jejich rotor z jednoho pólu na druhý. Nevýhodou krokových motorů je tzv. ztráta kroku, která nastává při překročení mezního zatížení. Naopak dobrá zpráva je, že toto překročení lze de-

tekovat a po případném přetížení pohyb motoru zastavit. Rozsah otáček krokových motorů je typicky pod 2000 otáček za minutu, protože jejich točivý moment se snižuje s rostoucími otáčkami, avšak pro účely této práce nepotřebujeme dosahovat žádných závratných rychlostí. Krokové motory se používají v aplikacích, kde je vyžadováno přesné polohování, jako jsou 3D tiskárny a CNC stroje.

Z úvodního popisu se zdají být oba kandidáti vhodní pro použití na tento projekt, nicméně klíčovou vlastností je detekce přetížení u krokového motoru. Detekcí přetížení bude užitečná pro detekci překážek v cestě závěsu. Při detekování překážky bude motor zastaven a neztratí se tak pozice kroku.

Jako vhodný kandidát byl tedy zvolen krokový motor **NEMA17**.



Obrázek 4.1: NEMA17 (Převzato z webu [1])

Ovladač motoru

Krokový motor nestačí připojit ke zdroji stejnosměrného proudu, ale je k němu potřeba řídicí jednotku. Nejpoužívanějším ovladačem krokových motorů bývá ovladač **A9488**. Nevýhodou tohoto ovladače ale je vysoký hluk při pohybu motoru. Pokud se závěsy mají použít jako ranní budíček slunečními paprsky, hluk vydávaný motorem by zajisté nebyl příznivým faktorem. Naopak výhodou tohoto ovladače je jeho cena, která se pohybuje okolo 50,-Kč.

Co je ale důležitější, **A9488** neumožňuje zmiňovanou detekci zaseknutí. Po prozkoumání trhu vhodné ovladače nabízí firma **Trinamic**². Detekce zaseknutí se nazývá technologie **StallGuard**³. Touto technologií disponuje větší řada jejich mikročipů. Pro účely tohoto projektu jsem zvolil z důvodu dostupnosti mikročip **TMC2209**⁴ implementovaný na ovladači od firmy **Makerbase**⁵ (*zkr. MKS*), která se zabývá vývojem základových desek pro 3D tiskárny, kde jsou právě tyto ovladače použity. Cena jednoho tohoto ovladače se pohybuje okolo 240,-Kč.

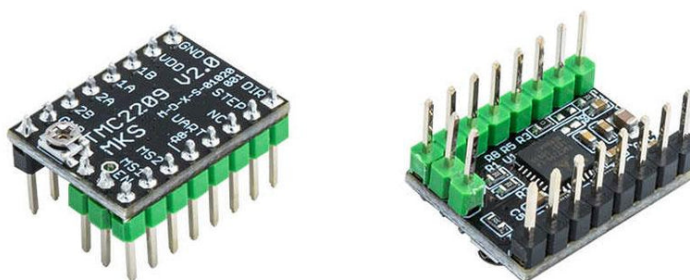
¹<https://www.majkl3d.cz/p/krokovy-motor-nema17-40mm-1-8-1-7a>

²<https://www.trinamic.com/>

³<https://www.trinamic.com/technology/motor-control-technology/stallguard-and-coolstep/>

⁴<https://www.trinamic.com/products/integrated-circuits/details/tmc2209-1a/>

⁵<https://makerbase.com.cn/en/>



Obrázek 4.2: TMC2209 (Převzato z webu [6])

Světelný senzor

Na trhu se nachází mnoho senzorů. Některé jsou specializované na určitou část jako například světlo, teplota, kvalita vzduchu či magnetické pole, některé zase kombinují měření více těchto faktorů dohromady. Pro tento účel bude stačit specializovaný pouze na světlo.

Všechny tyto světelné senzory pracují na poměrně jednoduchém principu kdy převádějí světelnou intenzitu na digitální signál. Po ozáření fotodiody integrovaný operační zesilovač převádí proud fotodiody na napětí, které je AD převodníkem převáděno na číslicovou hodnotu.

Intenzita osvětlení

Lux se značkou lx , je fotometrická jednotka intenzity osvětlení, odvozená od základních jednotek SI. Označuje osvětlení, které způsobil světelný tok jednoho lumenu (hlavní jednotka světelného toku), dopadajícího na jeden metr čtvereční.

Intenzita denního osvětlení se typicky pohybuje v rozmezí 100 - 10 000 lx , domácí osvětlení kolem 100 - 500 lx a při zatažené obloze se intenzita osvětlení pohybuje okolo 1 000 lx .

Na základě úvodního textu byli vytipováni 3 příhodní kandidáti.

- **TSL2561 [1]**

Senzor pracuje na patentované metodě 2 fotodiód - fotodióda pro **infračervené záření**⁷ a fotodióda pro celé světelné spektrum⁸, je tedy schopen snímat a vyhodnocovat 2 světelné složky spektra. **infračervené záření** a **celé světelné spektrum / světlo viditelné člověkem**.

Cena tohoto senzoru se pohybuje kolem 260,-Kč.

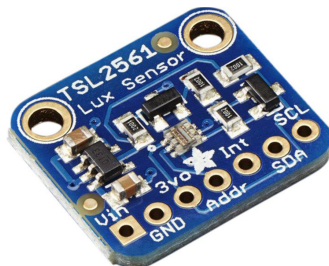
⁶<https://davincilab.co.za/products/mks-tmc2209-v2-0-stepper-driver>

⁷infračervené záření 720nm - 950nm

⁸celosvětelné spektrum 400nm - 760nm

Statistiky:

- Dynamický rozsah: 0,1 - 40 000,0 lx
- Teplotní rozsah: $-30 - 80$ °C
- Vstupní napětí: 2,7 - 5,0 V
- Komunikační rozhraní: I^2C



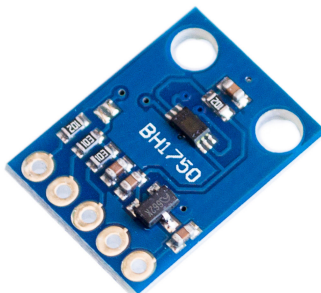
Obrázek 4.3: TSL2561 (Převzato z webu [9])

• BH1750 [7]

Tento senzor má implementovanou jednu fotodiodu snímající celé světelné spektrum. Cena se pohybuje okolo 110,-Kč.

Statistiky:

- Dynamický rozsah: 1 - 65 535 lx
- Teplotní rozsah: $-40 - 85$ °C
- Vstupní napětí: 2,4 - 3,6 V
- Komunikační rozhraní: I^2C



Obrázek 4.4: BH1750 (Převzato z webu [10])

⁹<https://www.distrelec.cz/cs/digitalni-snimac-svitivosti-tsl2561-5v-adafruit-439/p/30091216>

¹⁰<https://mbatechmeds.com/product/bh1750-light-intensity-sensor/>

- **VEML7700** [6]

Zajímavostí tohoto senzoru je, že vstupní světelné paprsky nesnímá kolmo k desce senzoru jako předešlé dva senzory, ale fotodiodu má vyvedenou pod úhlem 90°. Snímá tedy paprsky paralelně k tomuto senzoru.

Senzor má také, kromě klasických pinových dírek, již napájené JST SH¹¹ konektory, pro snazší připojení nevyžadující pájení.

Cena je kolem 200,-Kč.

Statistiky:

- Dynamický rozsah: 0 - 120 000 lx
- Teplotní rozsah: -25 - 85 °C
- Vstupní napětí: 3 - 5 V
- Komunikační rozhraní: I^2C



Obrázek 4.5: VEML7700 (Převzato z webu [12])

Všichni z výše uvedených kandidátů by vyhovovali podmínkám užití pro tento projekt. Teplotní podmínky vyhovují a rozsah světelné intenzity vzhledem ke snímání slunečního svitu také. Avšak TSL2561 dokáže měřit i infračervené záření a VEML7700 má dynamický rozsah příliš velký, což je pro tento projekt zbytečné. Vzhledem k přihlídnutí na cenu byl zvolen senzor **BH1750**.

Detektor úplného roztáhnutí/zatáhnutí

Po diskuzi s mým konzultantem z firmy Espressif - Radimem Karnišem, mi bylo navrženo a doporučeno, že by systém měl mít i jakousi zádržku. Ta je v systému proto, aby se v případě nějaké poruchy motor nesnažil roztáhnout či zatáhnout závěs za pomyslný konec garnýže. Tím je myšleno, aby se závěs nedostal k ose motoru - případně ose koncového bodu na druhé straně. Toto by pravděpodobně vedlo k přehřátí a zničení ovladače nebo motoru samotného. Návrh jsem přijal a dále jsem tento pojistný systém použil i pro inicializaci samotných závěsů - popsáno v kapitole 7.3.

¹¹<https://www.jst.com/?s=SH>

¹²<https://botland.cz/svetelne-a-barevne-senzory/21064-digitalni-snimac-okolniho-svetla-veml7700-i2c-uhlovy-stemma-qt-qwiic-adafruit-5378.html>

- Mechanický spínač - tlačítko

- Popis: Tlačítko - spínač je jedním z nejjednodušších prvků elektrického obvodu. Po zmáčknutí tlačítka, což znamená zatlačit na ovládací prvek prstem nebo jakýmkoli předmětem, se sepne elektrický obvod, a tím pádem zde může procházet elektrický proud.
- Připojení: 2 piny - vstupní I/O pin, GND pin (zem)
- Cena: okolo 15,- Kč



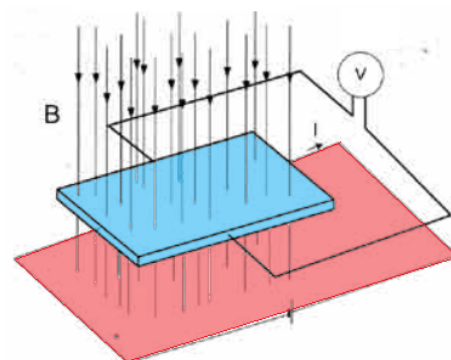
Obrázek 4.6: Součástka - tlačítko. (Převzato z webu [13])

- Hallova sonda

- Popis: Hallova sonda je součástka pomocí níž se měří magnetické pole. Je tvořena polovodičovou destičkou - *značena modře obr. B.8*, přes kterou prochází elektrický proud. Po přiblížení magnetu - *značen červeně* k destičce, skrz kterou prochází indukční tok, se náboje v destičce začnou přeskupovat. Volné nosiče náboje (elektorny) se začnou hromadit na jedné straně destičky, čímž na destičce vzniká napětí, které je možno detekovat.
- Připojení: 3 piny - vstupní I/O pin, Vcc pin (vstupní napětí), GND pin (zem)
- Cena: okolo 25,- Kč



Obrázek 4.7: Součástka - hallova sonda. (Převzato z webu [14])



Obrázek 4.8: Princip hallova jevu. polovodičová destička, magnet B - indukce magnetického pole I - elektrický proud protékající destičkou (Převzato z webu [15])

¹³<https://www.gme.cz/p-pb303b-yellow>

- Jazýčkové relé
 - Popis: Jazýčkové relé je jednoduchý magneticky ovládaný prvek se spínacími kontakty, které jsou zataveny do skleněné trubičky naplněné inertním plynem¹⁶. Kontakty, neboli jazýčky, jsou zataveny do skla tak, že se nepatrně překrývají. Po přiblížení magnetu se v magnetické poli jazýčky zmagnetují a na jejich koncích se objeví opačné póly. Tím se jazýčky přitáhnou a vytvoří spojení, přes který může proudit elektrický proud.
 - Připojení: 2 piny - vstupní I/O pin, GND pin (zem)
 - Cena: okolo 20,- Kč



Obrázek 4.9: Součástka - jazýčkové relé. (Převzato z webu [¹⁷])

Pro řešení s mechanickým spínačem, by pravděpodobně museli být spínače 2 a trajektorie zarážky s upevněním mechanických spínačů by muselo být velmi přesné. Na řemeny by z každé strany byla upevněna jedna zarážka, která by v případě chyby spínač sepnula a tím by se indikovala chyba.

Halova sonda je též zabudována ve vývojové sadě s mikročipem ESP32 - kapitola 4.2. Pro využití v tomto projektu je však nepoužitelná, jelikož by se vývojová sada musela nacházet v bezprostřední blízkosti motoru s řemenem. V případě použití halové sondy, by musela být použita externí sonda.

Po zvážení všech možností jsem se rozhodl jako detektor použít **jazýčkové relé**, jako nejjednodušší a nejlevnější řešení. Použití tlačítka by bylo zbytečně komplikované a musely by být použity dvě, a logika halovy sondy je o něco složitější než jazýčkové relé. Přitom jazýčkové relé dostatečně splní požadavky pro tuto funkci.

Tepelné čidlo

Jelikož v systému se nachází části, které mají určitou provozní teplotu, a při vykonávání své činnosti se navíc zahřívají, je potřeba kontrolovat jejich stav, aby se nezničili. V tomto projektu bude kontrolována teplota krokového motoru, aby se nepřehřál. Pro kontrolování teploty byli vytipováni 2 kandidáti.

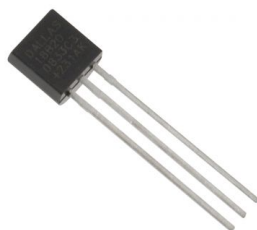
¹⁴<https://www.gme.cz/a1324lua-t>

¹⁵<https://elektrika.cz/data/clanky/clanek.2006-04-01.2480110481>

¹⁶Plyny které za daných podmínek nepodléhají chemickým reakcím.

¹⁷<http://www.implantable-device.com/category/implantable-components/magnetic-reed-switches/>

- Číslicový teplotní senzor DS18B20 [5]
 - Popis: Číslicový teplotní senzor je zařízení, jehož výstupem je kalibrované číslo udávající snímanou teplotu přímo ve stupních celsia. DS18B20 tento výstup posílá jedním ze svých pinů. Každý senzor má z výroby svou unikátní adresu, tím pádem je na jeden vodič jich možno připojit až 50.
 - Teplotní rozsah: -10 - 85 °C
 - Připojení: 3 piny - vstupní I/O pin, GND pin (zem), VCC pin (vstupní napětí)
 - Cena: okolo 150,- Kč



Obrázek 4.10: Součástka: číslicový teplotní senzor DS18B20 (Převzato z webu [18])

- Termistor [10]
 - Popis: Elektronická součástka, která mění svůj odpor v závislosti na teplotě. Rozlišujeme 2 typy termistorů
 Pozistor (PTC) - s rostoucí teplotou roste odpor
 Negistor (NTC) - s rostoucí teplotou odpor klesá
 - Teplotní rozsah: -40 - 150 °C
 - Připojení: 3 piny - vstupní I/O pin, GND pin (zem), VCC pin (vstupní napětí)
 - * Pro čtení hodnoty termistoru, je potřeba jej zapojit jako napěťový dělič
 - Cena: okolo 10 - 50,- Kč



Obrázek 4.11: Součástka: termistor (Převzato z webu [19])

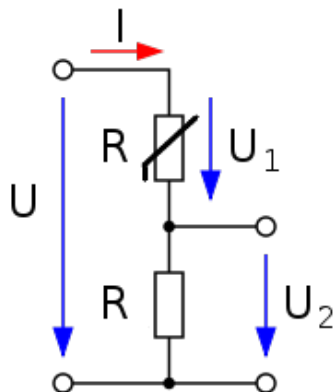
Použití číslicového teplotního senzoru by bylo nejspíše jednodušší, protože jeho výstup by se dal rovnou v programu použít. Avšak s ohledem na cenu, a také že je v systému k dispozici

¹⁸<https://www.gme.cz/ds18b20>

¹⁹<https://www.gme.cz/termistor-ntc-ntc-640-10k>

mikročip, který dokáže zpracovat měnící se odpor termistoru a převést jej na teplotu, jsem se rozhodl použít termistor.

Pro čtení dat z termistoru je zapotřebí jej zapojit jako napěťový dělič - někdy též odporový dělič.



Obrázek 4.12: Schéma napěťového děliče s termistorem (Převzato z webu [20])

Toto je jednoduché schéma kdy za použití rezistoru, je snímáno měnící se napětí na termistoru, a následně pomocí Steinhart-Hartovi rovnice, dokážeme vypočítat výslednou teplotu. Steinhart-Hartova rovnice: 4.1

$$\frac{1}{T} = A + B \ln R + C(\ln R)^3 \quad (4.1)$$

kde T je teplota v kelvinech, R je odpor v ohmech při teplotě T a A , B , C jsou Steinhart – Hartovy koeficienty, které se liší v závislosti na typu a modelu termistoru.

V mém případě znám koeficient B , a pro výpočet tedy použiji odvozenou B - parametrizovanou rovnicí 4.2

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln \frac{R}{R_0} \quad (4.2)$$

kde T_0 je teplota 25°C , B je koeficient udávaný výrobcem termistoru, R_0 je odpor termistoru při pokojové teplotě T_0 , R je naměřený odpor - v mém případě převedený na napětí pomocí odporového děliče a T je sledovaná teplota v kelvinech.

Následně už stačí požadovanou teplotu T_c převést na stupně celsia pomocí rovnice 4.3

$$T_c = T - 273.15 \quad (4.3)$$

²⁰https://cs.wikipedia.org/wiki/Dělič_napětí

4.2 Mikročip ESP32

Celým mozkiem operace musí být nějaká řídicí jednotka. Prvek který bude udávat kdy a jak která součástka bude pracovat. Dále také bude sloužit pro komunikaci s uživatelem, který bude udávat příkazy k pohybu závěsů.

Vzhledem k firmě pod kterou bylo zadání této práce vypsáno, byla rodina mikročipů pro daná. Avšak **Espressif**²¹ má ve své nabídce více sérií, ze kterých bylo možné vybírat.

Aktuální nabídka sérií ESP od firmy **Espressif**

- ESP32
- ESP32-S
 - ESP32-S2
 - ESP32-S3
- ESP32-C3
 - Novější a vylepšený nástupce předešlého modelu ESP8266, se stejným rozložením pinů.

Všechny tyto série nabízí různé možnosti jako například počet komunikačních rozhraní a jejich typů, počet pinů, podporu šifrování, typy interních senzorů atd...

Důležité je také zmínit, že kromě samotných mikročipů, **Espressif** nabízí implementované čipy na vývojových sadách (*angl. developing kit*). Tato možnost je důležitá právě pro domácí práci s mikročipy ESP, kdy tyto sady poskytují možnost připojení pomocí vyvedených pinů. Pro domácí vývoj není vždy zapotřebí ušetřit maximum místa za cenu použití mikropájení. Porovnání všech vlastností je možno vidět na oficiálních stránkách firmy **Espressif**²²

Požadované vlastnosti mikročipu pro tento projekt.

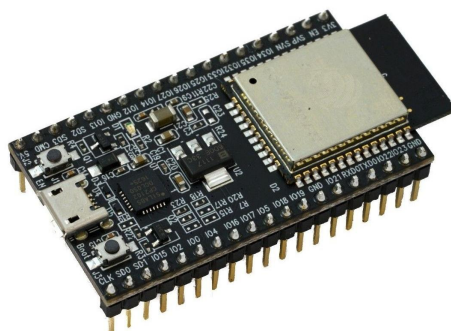
- Wifi připojení
- 5 komunikačních vstupno-výstupních pinů
 - 3 piny pro ovladač motoru
 - * krok motoru (*angl. step pin*)
 - * směr otáčení motoru (*angl. direction pin*)
 - * povolení funkčnosti motoru (*angl. enable pin*)
 - 1 pin pro čtení hodnoty jazýčkového relé
 - 1 pin pro čtení hodnoty termistoru
- napájení periférií - 2 piny
 - 3,3V napájení světelného senzoru
 - 5,0V napájení ovladače motoru
- komunikační rozhraní I2C - 2 piny
- komunikační rozhraní UART - 2 piny

²¹<https://www.espressif.com>

²²<https://www.espressif.com/en/products/modules>

Požadovaným vlastnostem pro tento projekt vyhovují všechny ze sérií ESP. Vybral jsem tedy **vývojovou sadu s mikročipem ESP32** z důvodu, že je to doposud nepoužívanější aktuálně vyráběný mikročip firmy **Espressif**.

Cena jedné takové vývojové sady se pohybuje kolem 280,-Kč.



Obrázek 4.13: Vývojová sada: ESP32 DevKitv V4 (Převzato z webu [23])

4.3 Režimy systému okenních závěsů

System nabízí 3 režimy, kterými je závěsy možno ovládat - 2 automatické a manuální.

Časový režim nabízí uživateli možnost nastavit si na každý den v týdnu v daný čas s přesností na minuty - *příklad 16:48* - na kolik procent budou závěsy roztáhnuty.

Světelný režim pomocí připojeného senzoru snímá vnější osvětlení. Uživatel si může nastavit hraniční hodnotu nad kterou když se světlo dostane, závěsy se automaticky zatáhnou. Obdobně funguje druhá hraniční hodnota pro roztáhnutí závěsů.

Poslední možností je **manuální režim**, aby si uživatel mohl kdykoli závěsy roztáhnout do polohy, která mu aktuálně vyhovuje.

4.4 Komunikace se systémem

System funguje na principu klient - server, za použití MQTT protokolu. Na mikročipu běží klient ovládající závěsy, který se připojí k serveru běžícím na nějakém zařízení. V mém případě jsem jako server zvolil platformu chytré domácnosti **Home assistant**²⁴ běžící na **Arduino UNO**²⁵, ve kterém je spuštěna aplikace **MQTT Mosquitto broker**²⁶.

²³<https://www.soselectronic.cz/products/espressif/esp32-devkitc-ver-d-esp32-devkitc-32d-305403>

²⁴<https://www.home-assistant.io>

²⁵<https://www.arduino.cc/en/main/arduinoBoardUno>

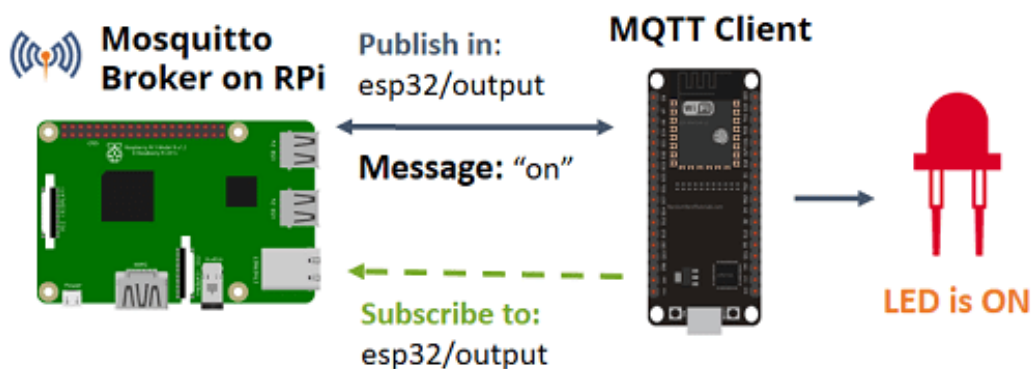
²⁶<https://www.home-assistant.io/docs/mqtt/broker/>

MQTT

MQTT [8] je zkratka pro Message Queuing Telemetry Transport. MQTT je jednoduchý protokol pro zasílání zpráv, určený pro zařízení s nízkou šířkou pásma (*angl. Low Bandwidth*). Je to tedy ideální řešení pro výměnu dat mezi více zařízeními IoT.

Komunikace MQTT funguje na základě vysílajících (*angl. publish*) a odebíraných (*angl. subscribe*) zpráv. Zařízení publikují zprávy na určité téma. Všechna zařízení, která jsou přihlášená k tomuto tématu, zprávu obdrží. Dalo by se říct že koncept je podobný multicastové komunikaci.

MQTT Broker je zodpovědný za příjem všech zpráv. Zprávy následně filtruje a rozhoduje o tom, kdo o dané zprávy má zájem a následně je odebírajícím zařízením rozesílá.



Obrázek 4.14: Znázornění komunikace zasílaných MQTT zpráv brokeru a klienta. (Převzato z webu [27])

Home assistant

Home assistant je aplikace fungující jako centrální řídicí systém pro řízení domácí automatizace. Umožňuje propojení jednotlivých chytrých zařízení nacházejících se na lokální síti. Na základě údajů shromážděných z jednotlivých senzorů umožňuje rozhodovat o činnosti jednotlivých zařízení bez zásahu člověka - automatizuje činnosti.

Poskytuje také uživatelské rozhraní na kterém si uživatel může zobrazit stavy a data připojených zařízení - obrázek 8.6. Výhodou také je, že uchovává historii snímaných dat, které si je možno nechat vykreslit v grafech - obrázek 8.5. Pomocí tohoto rozhraní je, kromě zobrazování stavů, také prvky ovládat a to vše z jednoho místa.

Pro připojení k Home assistantu není ani potřeba instalovat žádnou aplikaci. Jelikož běží jako server na lokální síti, je ovládání zprostředkováno z webového prohlížeče.

²⁷<https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>

Kapitola 5

Návrh obvodového řešení

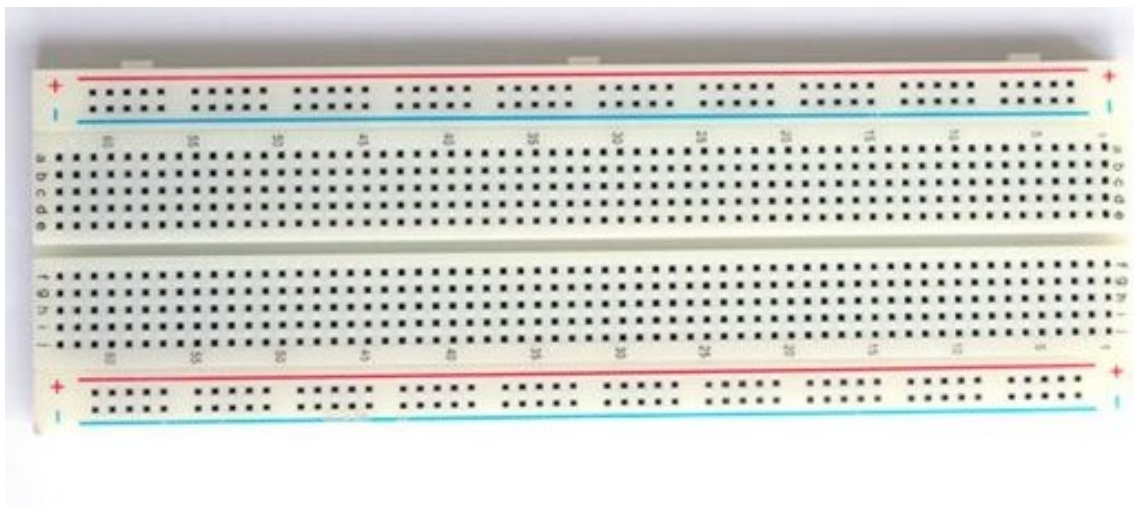
Po úspěšném vybrání všech komponent a navrhnutí jak by mohl systém fungovat, jsem začal komponenty propojovat a testovat jejich funkčnost. K prototypování a testování jednotlivých periférií s mikročipem jsem si vybral platformu **Arduino IDE**¹. Cílem navrhnutí prototypu bylo totiž ověřit základní funkčnost jednotlivých částí a zjištění jakým způsobem periférie a mikročipem komunikují. Možnost testování ve vývojovém prostředí **Arduino IDE**, je díky tomu, že firma Espressif má celý vývojový tým pro podporu mikročipů ESP v tomto vývojovém prostředí.

5.1 Propojení jednotlivých komponent

Důležitou věcí prvních elektrotechnických návrhů je nepájivé pole pinů - na obrázku 5.1. Jedná se o konstrukční základnu pro prototypování elektronických obvodů. Je to plastová destička s velkým množstvím dírek uspořádaných do řad a sloupců jako šachovnice. Jednotlivé sloupce jsou spolu propojeny, nejsou však propojeny řady.

Dále se na pravém a levém okraji (na obrázku 5.1 vrchní a spodní okraj) destičky nachází vždy dvě řady dírek, kolem kterých vede modrý nebo červený proužek. Tyto okrajové řady nejsou nijak propojeny spolu ani se zbytkem nepájivého pole. Propojení těchto okrajových dírek je opačné než u dírek na vnitřní ploše nepájivého pole, a je tedy propojena jedna celá řada (ve směru proužků), nikoli však sloupce. Tyto okrajové díčky slouží pro rozvedení vstupního napětí a logické 0 (země), aby tyto hodnoty byly přístupné na celém nepájivém poli a nemuseli se rozvádět pouze ze zdroje.

¹<https://www.arduino.cc/en/software>



Obrázek 5.1: Nepájivé pole pinů

K testování a prvnímu prototypování jsem tedy použil nepájivé pole pinů a sadu vodičů - drátů, jako nástroj propojení jednotlivých komponent. Po zjištění a nastudování jak mají jednotlivé periferie s mikročipem komunikovat jsem se pustil do testování požadované funkčnosti.

Světelný senzor BH1750

BH1750 má 5 pinů.

- VCC - napájení zařízení
- GND - přivedení logické nuly - zem
- SCL - příjem hodinového signálu komunikačního rozhraní I^2C
- SDA - příjem dat komunikačního rozhraní I^2C
- ADDR - adresa zařízení pokud by se na komunikačním rozhraní I^2C nacházelo více periferií

Zapojení bylo provedeno podle tabulky 5.1, kde levý sloupec odkazuje na piny světelného senzoru BH1750 a pravý sloupec na zozložení pinů z obrázku 5.5

BH1750	ESP32-DevkitC
VCC	3V3
GND	GND
SCL	GPIO22 / WIRE_SCL
SDA	GPIO21 / WIRE_SDA

Tabulka 5.1: Propojení pinů světelného senzoru BH1750 s vývojovou sadou ESP32-DevkitC

Pin ADDR nebyl použit z důvodu že na sběrnici se nachází pouze jedno zařízení. Nebylo tedy potřeba rozlišovat více zařízení a byla použita východzí adresa nastavená výrobcem 0x23.

Motor NEMA17 a ovladač TMC2209

Jak již bylo popsáno v kapitole 4.1 - *část ovladač motoru*, motor není možné připojit přímo k mikročipu, ale ovládání je realizováno pomocí ovladače. Z motoru vedou 2 dvojice kabelů - celkem 4 konektory, kde každá dvojice je připojena k jedné cívce v motoru.

Bohužel dokumentace² ovladače není dostatečná na přesné definování funkcí jednotlivých pinů. Nedostatečný popis pinů a jejich funkcí je pravděpodobně zapříčiněn tím, že firma **Makerbase**, se zabývá především výrobou základových desek, na nichž se nachází vyvedené zkratovací propojky (*angl. jumper*), pomocí kterých jsou na ovladači přepínány režimy - jako například přenos dat pomocí UART.

Ovladače TMC2209 jsou vyráběny i jinými společnostmi jako **silentstepstick**³ nebo **fysetc**⁴ které mají dokumentaci těchto ovladačů lepší i s popisem pinů, avšak některé piny jsou zde popsány jinými názvy a lehce se liší i popis jejich funkcí.

TMC2209 má 20 pinů.

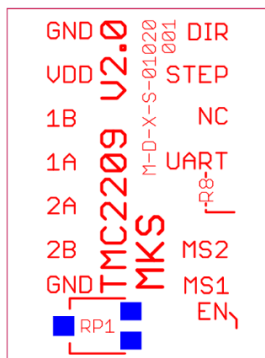
POZNÁMKA Kvůli nedostatečné dokumentaci popis pinů, které nebyly použity v tomto projektu, je pouze odhadem jejich funkčnosti na základě názvů pinů případně polohy pinu v kombinaci s dokumentací jiného výrobce.

- 2x GND - přivedení logické nuly - zem
- VDD - napájení zařízení 3,3 - 5,0V
- 1A, 2A - cívka motoru 1
- 1B, 2B - cívka motoru 2
- VM - napájení motoru 5,5 - 28V
- DIR - směr rotace motoru
- STEP - signál kroku motoru
- EN - povolení činnosti kroku motoru
- MS2 - nastavení mikro-krokování motoru
- MS1 - nastavení mikro-krokování motoru
- NC - pravděpodobně přívod externího hodinového signálu
- UART - komunikace pomocí UART
- R8 - pravděpodobně povolení komunikace UART
- DIAG - diagnostický výstup
- INDEX - indexový výstup
- VREF - analogové referenční napětí

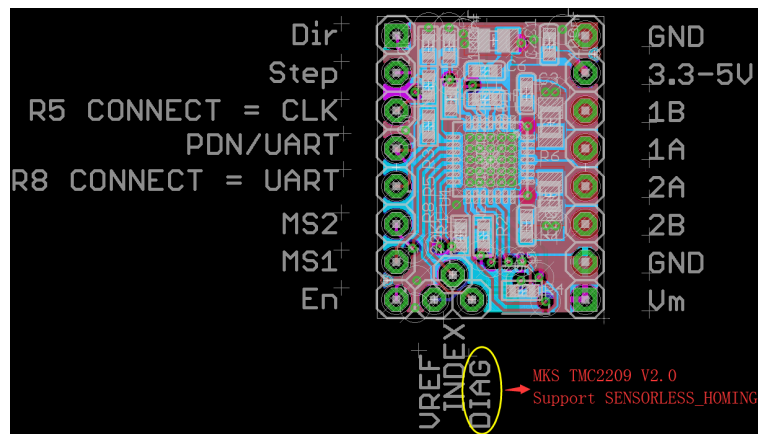
²<https://github.com/makerbase-mks/MKS-StepStick-Driver/tree/master/MKSTMC2209>

³<https://learn.watterott.com/silentstepstick/pinconfig/tmc2209/>

⁴<https://wiki.fysetc.com/Silent2209/>



Obrázek 5.2: Názvy a rozložení pinů TMC2209 (Převzato z webu [5])

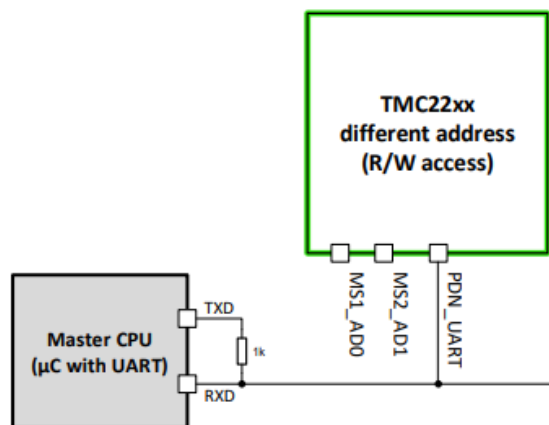


Obrázek 5.3: Popis pinů TMC2209 s popisem rezistorů potřebných k aktivaci módů (Převzato z webu [6])

V tomto kroku bylo zapotřebí zprovoznit i detekci zaseknutí motoru, kterou měla zabezpečit funkce **StallGuard** zmiňovaná v kapitole 4.1 - část ovladač motoru. Avšak pro nastavení a čtení hraniční hodnoty zaseknutí - **StallGuard Threshold**, je potřeba s ovladačem komunikovat pomocí komunikačního rozhraní UART.

To zde však není zcela jednoduché, vzhledem ke slabé dokumentaci. Z návodu k jedné ze základních desek MKS je patrné, že se UART mód musí aktivovat pomocí zkratovací propojky (*angl. jumper*). UART zde není implementován klasickým dvoudrátovým spojením, ale z ovladače vede pouze jeden drát, který se rozděluje na dva dráty až za (pravděpodobně) externím rezistorem.

Přikládám obrázek 5.4, z diskuzního fóra jak by zapojení samotného TMC2209 bez základové desky mělo nejspíše vypadat. Dále v příloze A je schéma TMC2209 s vyznačenými částmi pro UART.



Obrázek 5.4: Scéma zapojení TMC2209 pro komunikaci skrze UART (Převzato z webu [7])

⁵https://github.com/makerbase-mks/MKS-StepStick-Driver/blob/master/MKSTMC2209/MKSTMC2209V2.0_001/MKSTMC2209V2.0_001TOP.pdf

⁶https://github.com/makerbase-mks/MKS-StepStick-Driver/blob/master/MKSTMC2209/MKSTMC2209V2.0_001/MKSTMC2209V2.0_001Layout.png

Zkoušel jsem tedy zapojení vyčíst z dostupného katalogového listu⁸ (*angl. datasheet*), v kombinaci s diskuzními fóry, kde uživatelé řešili podobný problém, a někteří uživatelé uváděli přepájení určité rezistory na ovladači. Chyba byla nejspíše na mé straně, jelikož ovladač v kombinaci se základovou deskou toto spojení umožňuje, avšak bohužel nedokázal jsem z ovladače vyčíst žádnou hodnotu.

V této fázi jsem tedy prozatím upustil od detekce zaseknutí motoru, jelikož desku plošných spojů *kapitola - 5.2* bylo potřeba poslat do výroby a nějakou dobu také trvá než se deska stihne vyrobit a z Číny balík dorazit.

Zapojení bylo provedeno podle tabulky 5.2, kde první sloupec odkazuje na piny ovladače motoru TMC2209, druhý a třetí sloupec na piny mikročipu *obr. - 5.5* a motoru a poslední sloupec určuje přivedené napětí a zem do elektrického obvodu.

TMC2209	ESP32-DevkitC	NEMA 17	VCC & GND
GND			GND
VDD	5V		
1A, 2A, 1B, 2B		4 piny motoru	
GND			GND
VM			Vstupní napětí 19V
DIR	GPIO27		
STEP	GPIO25		
EN	GPIO15		

Tabulka 5.2: Propojení pinů ovladače TMC2209, vývojové sady ESP32-DevkitC, motoru NEMA 17 a vstupního napětí a země

Jazýčkové relé

Zapojení jazýčkového relé je v celku jednoduchá záležitost. Má totiž pouze 2 vývody a chová se jako rozpojený, v případě přiblížení magnetu spojený drát. Na jeden konec je přivedena logická 0 - zem. Na druhém konci, který je zapojen jako vstupní pin pro čtení hodnot GPIO19, je pomocí pull-up rezistoru přivedena uměle vytvořená logická 1. Po přiblížení magnetu je tedy obvod spojen, a na čtecím pinu se objeví logická 0.

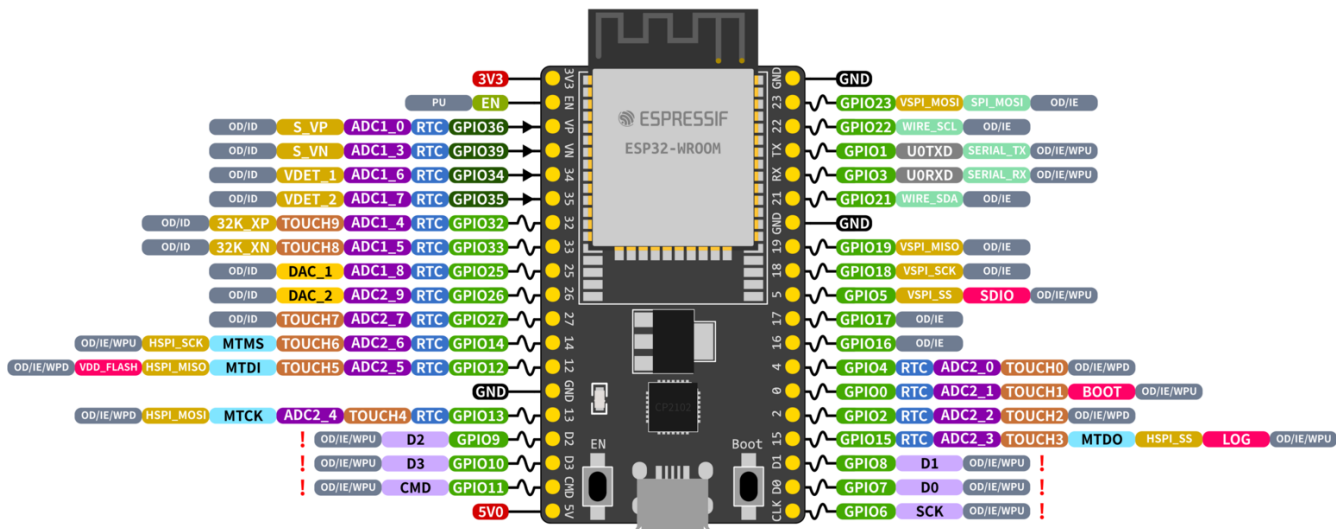
Termistor

Jak již bylo zmiňováno v předchozí kapitole, pro čtení měnicího se napětí v závislosti na odporu termistoru, je potřeba jej zapojit jako napěťový dělič. K tomu jsem použil externí rezistor s odporem 10 000Ωa použitý NPC termistor má při pokojové teplotě 25°C, odpor také 10 000Ω. Způsob zapojení je vidět na obrázku 5.6

⁷<https://stackoverflow.com/questions/67099347/arduino-communication-with-tmc2209>

⁸https://github.com/makerbase-mks/MKS-StepStick-Driver/blob/master/MKSTMC2209/MKSTMC2209V2.0_001/MKSTMC2209V2.0_001SCH.pdf

Rozložení pinů ESP32-DevkitC

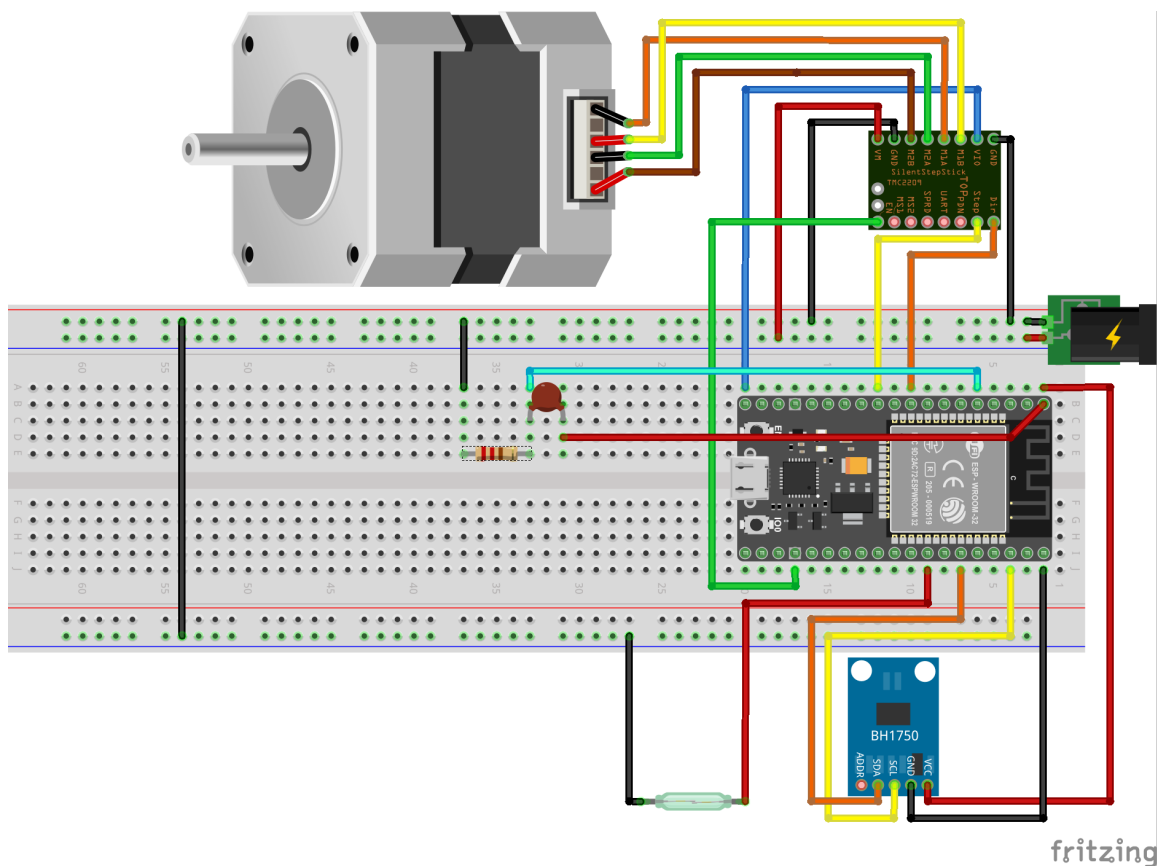


Obrázek 5.5: Rozložení pinů ESP32-DevkitC (Převzato z webu ^[9])

⁹https://docs.espressif.com/projects/esp-idf/en/latest/esp32/_images/esp32-devkitC-v4-pinout.png

Náhled testovacího prototypu

Otestování funkčnosti jednotlivých částí systému a výsledné propojení testovacího prototypu je znázorněno na obrázku 5.6. K vytvoření tohoto náhledu byl použit program fritzing¹⁰



Obrázek 5.6: Schéma zapojení jednotlivých částí systému

5.2 Motiv desky plošných spojů

Test jednotlivých součástek proběhl úspěšně a prototyp dokázal snímat světlo a roztočit motor. Problém však byl, že řešení takto pozapojované není příliš praktické. Kably by se mohly vypojet a takto velké řešení by na zdi nikdo mít nechtěl. Rozhodl jsem se tedy zkusit vytvořit desku plošných spojů. Deska plošných spojů slouží jako efektivnější náhražka nepájivého pole s dráty, kde je přesně vyznačeno které piny spolu mají být propojeny, a tyto spoje jsou následně spojeny vodičem vytvořeným v tenké ploché kovové vrstvičce na povrchu desky.

Vývojové prostředí

První rozhodnutí, které bylo potřeba učinit - výběr vývojového prostředí pro modelování desky plošných spojů. Vzhledem k tomu, že jsem nikdy nic podobného nedělal, poptal jsem se lidí kteří ano. Doporučeny mi byly 2 neplacené aplikace.

¹⁰<https://fritzing.org>

Autodesk Fusion 360¹¹, který v sobě má i modul Electronics, určený právě pro modelování desek plošných spojů a aplikace KiCad¹². Po rychlé rešerši, jakožto naprosto nezkušený tvůrce mi přišly obě prostředí dost podobné. Rozhodující nakonec bylo zamyšlení co bude se bude víc hodit v následném vývoji. Vzhledem k tomu, že mým úkolem bylo namodelovat kompletní řešení, což zahrnuje i model mechanické části viz. kapitola 6.2, dělat vše v jednom programu se zdálo jako výhoda. Vybral jsem tedy Autodesk Fusion 360.

Modularita

Testovací prototyp fungoval úspěšně, avšak výsledný produkt jsem chtěl posunout ještě o krok dál. Instalace tohoto projektu by měla být vcelku jednoduchá a intuitivní a případná deinstalace taktéž. Zkrátka odpojení a připojení jednotlivých externích čidel a motoru, ale zrovna tak i ovladače a vývojové sady by mělo být snadné ať už pro zmiňovanou deinstalaci nebo pro případ, že by se nějaký z prvků systému rozbil. Systém by měl být zkrátka modulární.

K dosažení tohoto cíle jsem do návrhu přidal konektory JST XH¹³, kdy samec tohoto konektoru bude pevně napájen na desce plošných spojů a drátek se samičí připájen k externím prvkům.

Napájení

Další věc kterou bylo potřeba vyřešit je napájení mikročipu ESP32. Doteď totiž byl mikročip napájen micro-usb kabelem, který vedl z počítače, avšak ve výsledném produktu už žádný počítač poblíž nebude.

Jako první varianta by se mohlo jevit, napojit ESP32 na zdroj, který napájí motor. To by však znamenalo jisté zničení mikročipu, jelikož mikročip lze napájet pouze napětím v rozmezí 2,3 - 3,6V. Výhoda vývojové sady s mikročipem ESP32 je v tom, že napětí zde lze přivést o něco vyšší. Pin označený 5.0 na obrázku 5.5 by měl zvládnout napětí v rozmezí 5,0 - 12,0V, avšak doporučuje se toto napětí držet mezi 5,0 - 7,0V, aby nedošlo k příliš velké ztrátě tepelné energie na regulátoru napětí.

Nicméně 12,0V je stále příliš málo pro 19,0V zdroj napájení motoru. Bylo tedy potřeba použít další součástku, a to DC-DC Stepdown nastavitelný měnič¹⁴. Ten zajistí snížení vstupního napětí na chtěných 5,0V. Tímto napětím je napájena jednak vývojová sada s ESP32, ale také ovladač motru TMC2209. Poslední součástkou přidanou do systému byl ochranný 100 μ F kondenzátor, pro případnou ochranu přepětí ze zdroje. Ten je umístěn hned za přívod 19,0V.

5.3 Schéma zapojení

Základem návrhu bylo vytvořit schéma zapojení. Na to jsem musel vytvořit nebo stáhnout knihovny jednotlivých součástek. Každá knihovna musí mít schématickou značku pro schéma modelu a také značku s přesným rozložením pinů, která se použije jako obtiskový vzor při tvorbě desky plošných spojů.

Jelikož použité elektrosoučástky mají všechny piny stejně daleko od sebe - 1 inch (2.54 cm), namodeloval jsem tyto značky také v Autodesk Fusion 360.

¹¹<https://www.fusion360.cz>

¹²<https://www.kicad.org>

¹³<https://www.jst.com/?s=XH>

¹⁴<https://dratek.cz/arduino/1303-stepdown-nastavitelny-menic-s-lm2596-dc-dc.html>

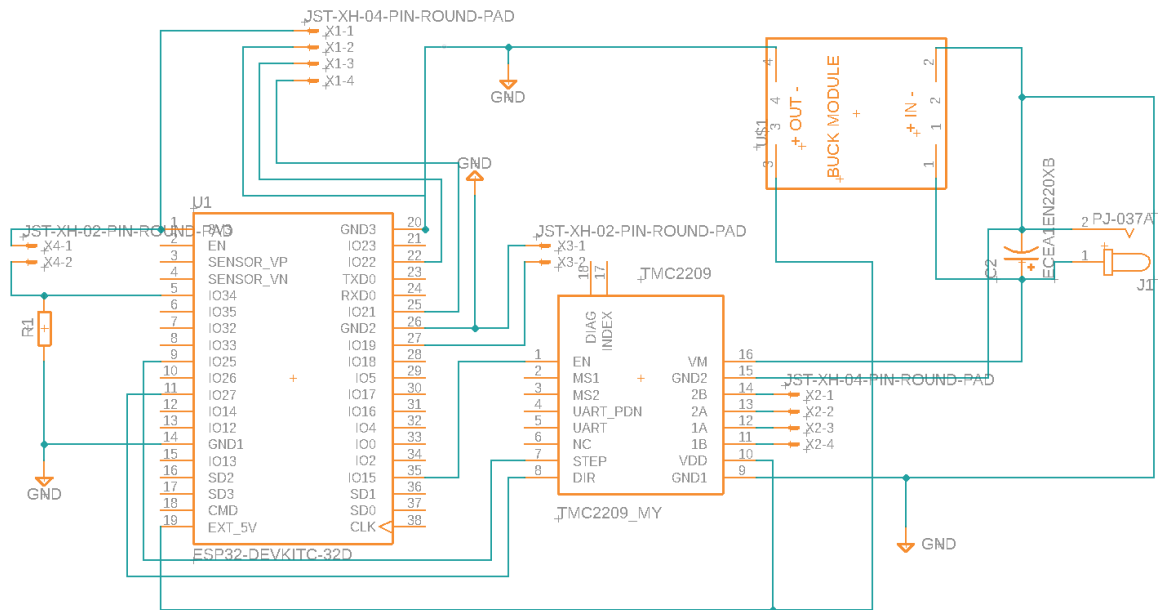
Naopak knihovny které jsem použil z internetu, byly pro připojovací konektory. Ať už pro JST XH¹⁵ nebo pro DC konektor¹⁶.

V Autodesk Fusion 360 se návrh desky plošných spojů tvoří pomocí dvou modulů. Prvním je schéma zapojení, které slouží pro propojení jednotlivých pinů všech komponent. Neudává přitom výsledné rozložení na desce plošných spojů. Všechny vložené prvky (ve formě schématických značek) do modelu schéma zapojení se automaticky přenesou do modelu druhého. Nazvěme jej dokument pro tvorbu desky plošných spojů (*angl. PCB Document*). Zde jsou prvky vložené zatím neuspořádané, s předznačenými spoji jednotlivých pinů. Poté už je na uživateli, aby si velikost desky plošných spojů s rozložením prvků na-modeloval sám.

Já jsem v rámci zmiňované modularity, umístil všechny konektory na jednu stranu desky plošných spojů, aby k těmto konektorům byl jednodušší přístup. Výsledek schématického spojení ukazuje obrázek 5.7 a PCB Dokumentu obrázek 5.8

Kvůli velkému množství navrhovaných spojení, jsem využil možnosti tvorby dvouvrstvé desky plošných spojů. Vrstva vrchní nese propojení jednotlivých pinů, zatímco spodní vrstva je připojená na logickou 0 - zem. Tím pádem všechny piny které jsou připojeny na schématickou značku GND, jsou propojeny ve spodní vrstvě a není potřeba je propojovat přímými spoji.

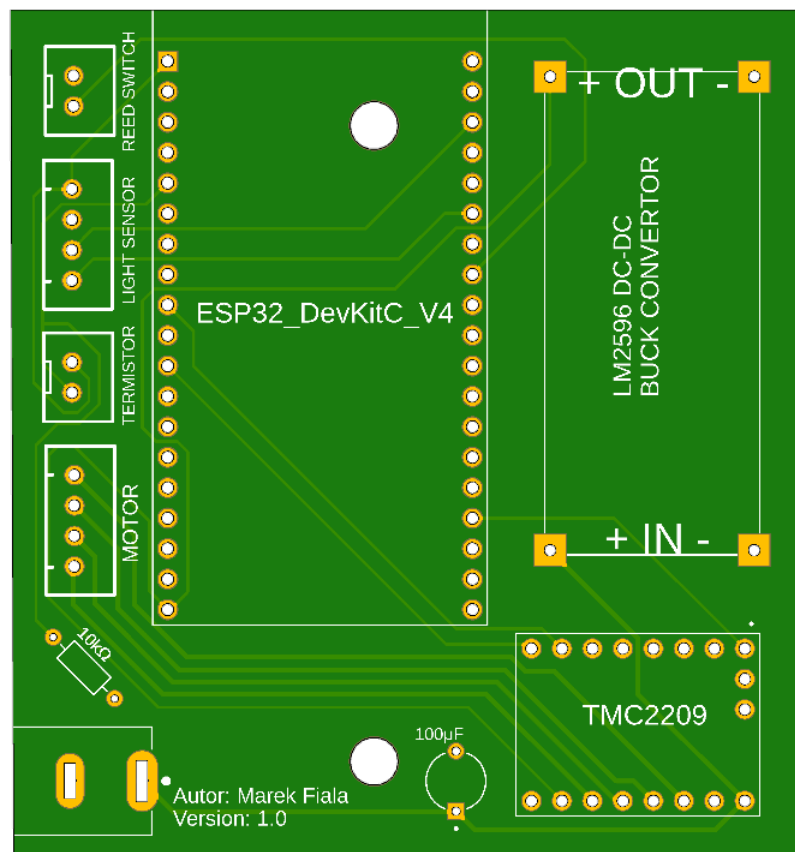
Pro snazší orientaci, kam který prvek zapojit je deska plošných spojů doplněna popisky jednotlivých konektorů a prvků.



Obrázek 5.7: Schéma zapojení

¹⁵https://grabcad.com/library/xh-connectors-1/details?folder_id=8375300

¹⁶<https://grabcad.com/library/dc-power-jacks-1>

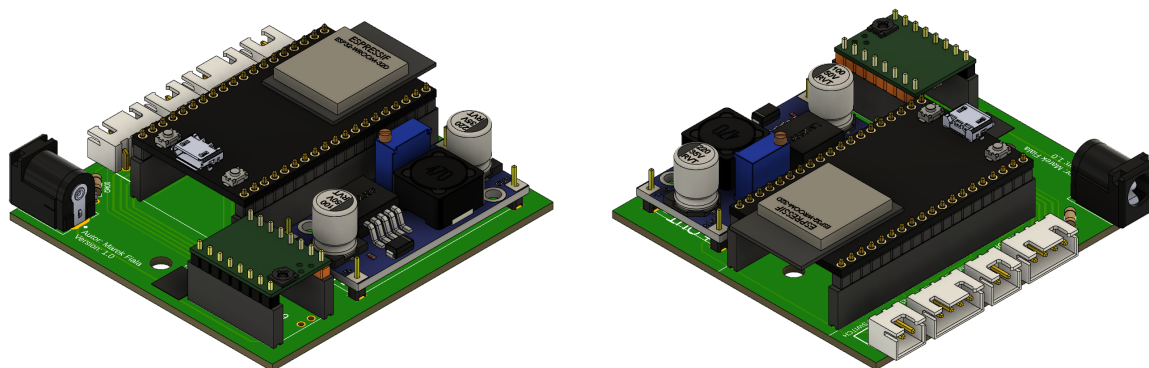


Obrázek 5.8: Výsledný otisk desky plošných spojů

5.4 3D model osazené desky plošných spojů

Pro lepší vizualizaci a následné modelování ochranného boxu pro desku plošných spojů jsem využil možnost programu Autodesk Fusion 360 vytvořit si 3D model osazené desky. Na to bylo do systému potřeba doimportovat 3D modely jednotlivých součástek.

V rámci zmiňované modularity jsem pro zapojení vývojové sady a ovladače motoru použil dutinkové lišty, které jsou napájeny na desce plošných spojů, do kterých je možno vkládat a zase odebírat zmiňované součástky.



Obrázek 5.9: 3D modely osazené desky plošných spojů

Kapitola 6

Mechanické řešení

Po zprovoznění elektrotechnické části přišla na řadu kreativnější část. Prozatím byla jediná mechanická část systému točící se osa motoru. Toto otáčení bylo potřeba přenést na pohyb závěsů, vymyslet jak motor bude k závěsu upevněn a jak bude fyzicky fungovat jazýčkové relé s magnety.

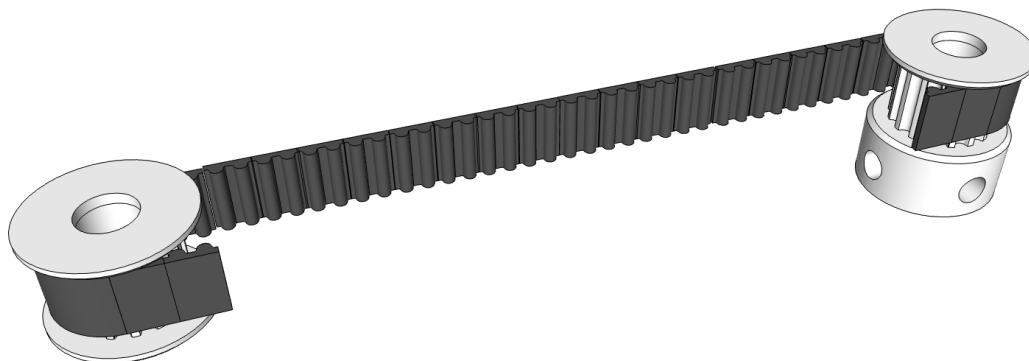
6.1 Mechanické komponenty

Prvním nápadem přenesení otáčivého pohybu osy motoru byl navinutý provázek s ložiskem na druhé straně garnýže. Jelikož ale povrchy ložiska i provázku jsou celkem hladké, mohlo by se stát, že se motor bude otáčet, ale pod tíhou závěsu by se osa pod provázkem protáčela a závěs by stál na místě. Bylo tedy zřejmé, že bude potřeba hrubší styková plocha.

Inspiroval jsem se u nyní stále se rozrůstajícího trhu 3D tiskáren. Motor který používám se v 3D tiskárnách používá také, a pohyb jednotlivých os zde zabezpečují řemenice a řemen.

Vybral jsem tedy řemen GT-2, o šířce 10mm. K němu jednu řemenici, která je nasazena na osu motoru, a plní tedy činnost přenesení otáčivého pohybu na posuvný. Z druhé strany je potřeba řemen kolem něčeho pouze vinout, zvolil jsem zde tedy klasickou kladku, který nemá hladký povrch, nicméně vroubky právě pro řemen. obr. 6.1

Při vybírání velikosti průměru řemenice (počet zubů), jsem musel zkombinovat 2 věci. Nesmí být moc velký, aby síla motoru pro pohyb s řemenem nebyla zbytečně velká, a zároveň ani moc malý, aby se mezi řemen vešlo jazýčkové relé, a vyhnuli se naproti sobě 2 míjející magnety. Jako dostačující, jsem nakonec zvolil průměr 18,5mm, což odpovídá 20-ti zubům.



Obrázek 6.1: Model řemenu s kladkou (vlevo) a řemenicí (vpravo)

6.2 Konstruování 3D modelu

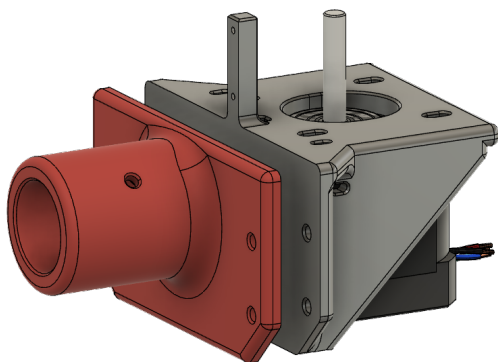
Jako úplný začátečník pro 3D modelig jsem se se musel seznámit jak vlastně 3D modelování funguje. V principu je to jednoduchý koncept. Po výběru jedné z ploch v 3D prostoy - půdorusu, nárysu nebo bokorysu - je na této ploše nakreslí náčrtek 2D útvaru, který je následně vytáhnout do 3D prostoru. Příkladem by mohl být náčrtek čtverce, který se vytáhne například do kostky nebo obdélníku. Následně se na tomto tělese vybere plocha, na které lze vytvořit další náčrtek a z něj vytáhnout nebo naopak prohloubit do tělesa díru. Opakováním tohoto postupu jsem byl schopen vymodelovat si vše potřebné. Jedním z posledních úprav na finálním modelu potom bylo zaoblení hran, aby při manipulaci s objekty nedošlo k poranění.

Upevnění systému na garnýž

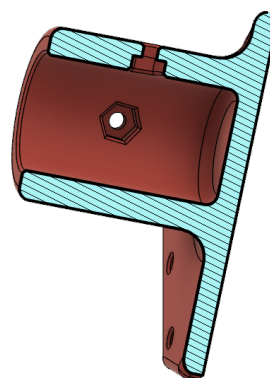
Jako model prototypového upevnění jsem si vybral kulatou garnýž, na níž jsou navlečena očka s připínacími kolíky, které drží závěs. Prvním úkolem bylo vymyslet jak motor upevnit ke garnýži. Toto řešení jsem se z důvodu jednodušší montáže, případně demontáže rozhodl zkonstruovat ze dvou částí.

První část slouží jako zasazení nosné části na garnýž - obr. 6.2 - červeně. Zde jsem vytvořil válcovitou násadu, která má průměr garnýže. V této násadě se také nacházejí 2 díry na šrouby, které slouží jako zábrana proti rotaci násady okolo osy garnýže. Tyto díry mají z vnitřní strany prohlubeň ve tvaru šestiúhelníku pro zachycení matky - obr. 6.3. Na konci je násada zakončena plochou ve tvaru obdélníku, se zkosenými dvěma rohy. Toto zkosení je zde kvůli ušetření tiskového materiálu.

Druhou část tvoří uchycení samotného motoru - obr. 6.4. Jedna ze stran kopíruje plochu násadu, a těmito stranami jsou následně části spojeny šrouby. Ve vrchní části se nachází několik děr. Tyto díry slouží k vyvedení pohyblivé osy motoru, uchycení motoru ke konstrukci a také jsou zde dírký pro vyvedení kabelů jazýčkového relé. Pro uchycení jazýčkového relé je z vrchní části namodelován tenký obdélníkový úchyt, který se ve výsledném systému bude nacházet před řemenicí mezi stranami řemenu.



Obrázek 6.2: Model celkového uchycení motoru na garnýž i s motorem. **uchycení na garnýž**, uchycení motoru

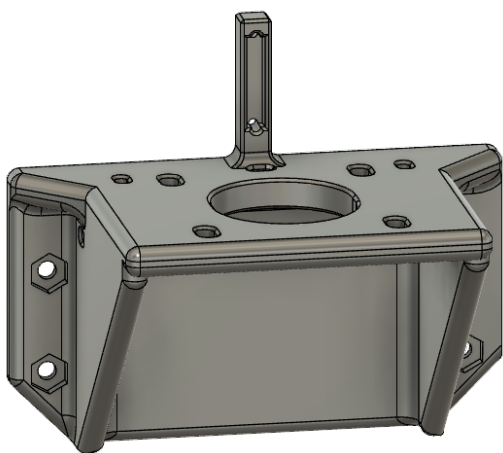


Obrázek 6.3: Průřez uchycením se znázorněnou prohlubní na matku

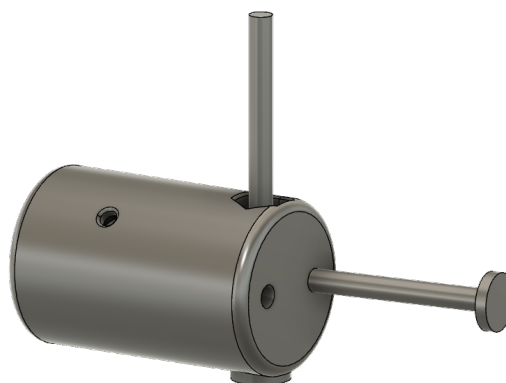
Z druhé strany garnýže jsem potřeboval umístit zmiňovanou kladku - obr. 6.5 Pro uchycení jsem použil stejný princip jako pro upevnění motoru, tedy násadu ve tvaru válce s polomě-

rem garnýže i s proti aretačními dírami. Na konci násady je ze spodní strany vymodelovaná díra pro šroub, který slouží jako osa, kolem které se bude kladka otáčet.

Důležité bylo také myslet na konstrukční řešení, a to v tomto případě, že řemen bude potřeba nějak napnout. Pro toto jsou zde ze zadní strany vymodelovány díry pro napínací šrouby. Jelikož zde bude tlak celkem velký, jsou zevnitř opět prohlubně pro zajišťující matky, aby šrouby v dírách neprokluzovali. Jelikož už prostředkem koncovky vede šroub pro osu kladky, bylo potřeba díry pro napínací šrouby vymodelovat pod úhly 6° a -6° .



Obrázek 6.4: Model uchycení motoru



Obrázek 6.5: Model koncovky pro kladku s příkladovými šrouby

Poslední částí, kterou bylo potřeba vymyslet pro kompletní funkčnost systému bylo uchycení magnetů na řemen a jak bude spojen řemen s očky jezdící po garnýži. Pro uchycení magnetů jsem si namodeloval jednoduchou skobu ve tvaru písmene U. Mezera mezi nožičkami je tak velká jako je tloušťka řemene - tedy 1,39mm. Zároveň ve stěně jedné z nožiček je prohlubeň do které se zasadí magnet.

Jelikož řemen bylo nutné slepit, rozhodl jsem se kromě lepení toto místo ještě pojistit vnější silou. Pro tento účel jsem si namodeloval destičky o šířce slepovaného spoje, které jsou k sobě přitáhnuty šrouby. Spojil jsem tuto ztužovací destičku zároveň s funkcí táhnutí oček závěsu a přimodeloval jsem na spodní stranu jedné z destiček očko, kterým povede drátek tahající závěs. Detail výsledné destičky je v příloze **B**

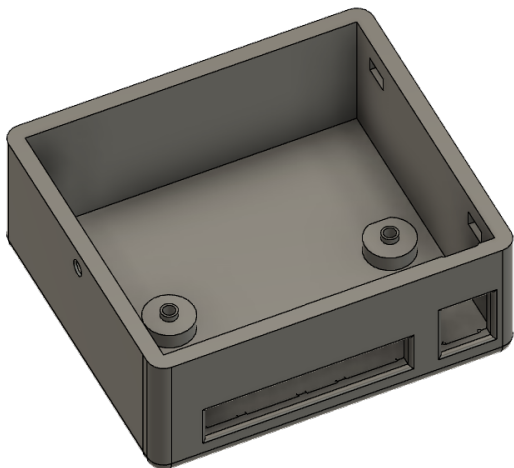
Krabička na desku plošných spojů

Aby se na osazenou desku plošných spojů neprašilo, a ve výsledku by to any nevypadalo esteticky, rozhodl jsem se pro PCB desku vytvořit krabičku. Ta se skládá ze dvou částí. Krabička ve tvaru kvádrů bez vrchní strany, aby se do ní dala deska vložit a následně víka.

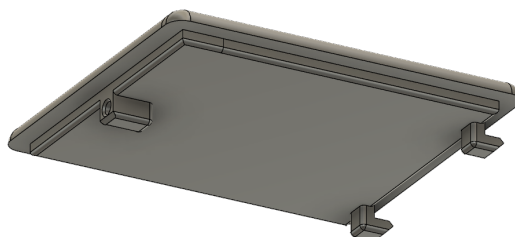
Již při modelování desky 5.8 jsem myslel na to, že bude někde upevněna. Proto se ve vrchní, a spodní části desky nachází 2 třímilimetrové díry.

Na dně krabičky - obr. 6.6 se nachází 2 vyvýšené podstavce. Toto vyvýšení je zde proto, jelikož piny pájené zespodu desky kousek přesahují, a deska by nedoléhala přímo na zem. Z těchto dvou podstavců jsou vyvýšeny 2 třímilimetrové kroužky, které přesně zapadnou do vymodelovaných děr na desce.

Z boční strany krabičky se nachází 2 díry, ty jsou zde pro zapojení konektorů všech vnějších periférií a napájecího kabelu a 3 díry pro usazení víka. 2 pro zasunutí nožiček a na opačné straně díra na šroub, která má z vnitřní strany prohlubeň na matku.

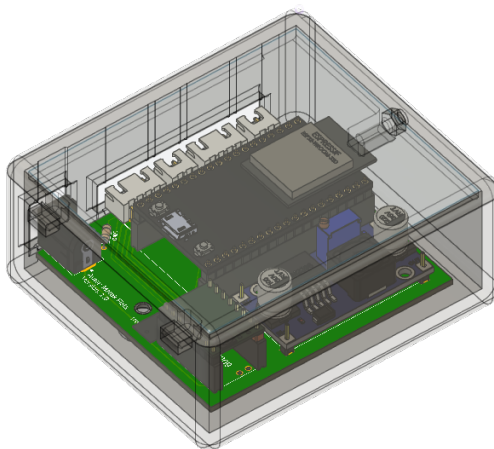


Obrázek 6.6: Krabička



Obrázek 6.7: Víko

Víko [6.7](#) kopíruje tvar krabičky s tím, že v okolí vnitřního obvodu je třímilimetrová vyvýšenina, aby se víko na krabičce neposouvalo. Dále z víka vedou 2 nožičky ve tvaru L a na opačné straně je namodelována díra na šroub, který víko na krabičce jistí.



Obrázek 6.8: Krabička s víkem a vloženým modelem desky plošných spojů (průhlednost víka 30%, průhlednost krabičky 50%)

6.3 Demonstrační prototyp

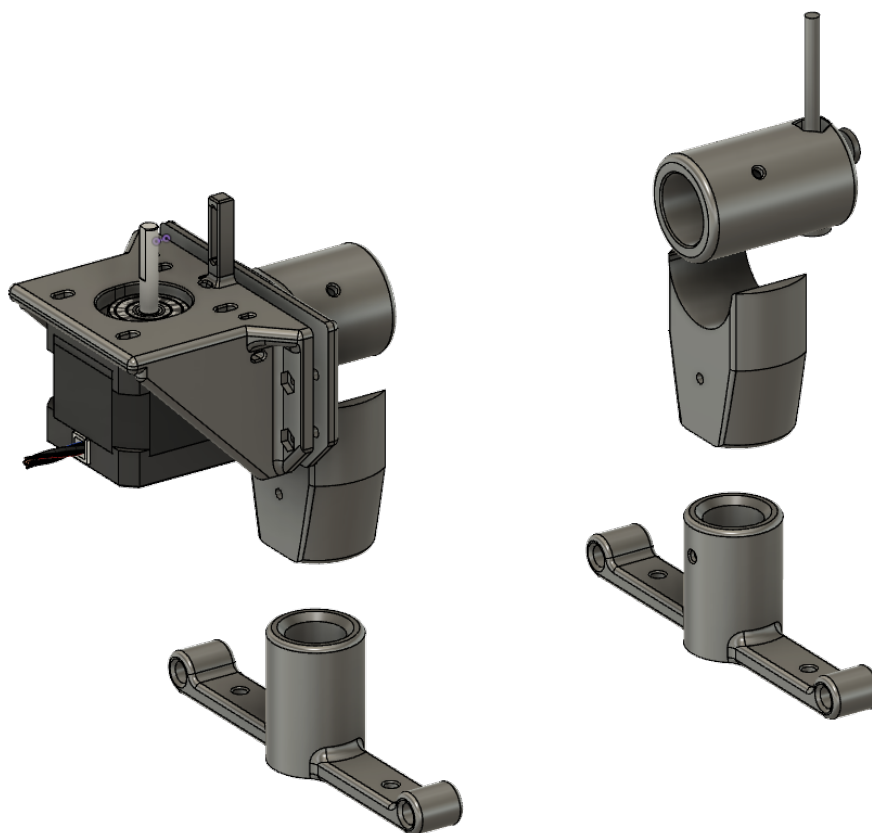
Modelovaný systém má sloužit pro reálný automatizovaný pohyb závěsů. Avšak při vyvíjení každého systému je nejprve vytvořen prototyp, na kterém se všechny potřebné části otestují a až následně je vyroben finální produkt. Já jsem se jako tento demonstrační prototyp rozhodl vytvořit zmenšenou verzi závěsu, která pro mě bude lépe dostupná. Přecjen, pro

test každé části co jsem naimplementoval se muset dostat pod strop k oknu by bylo celkem nešikovné.

Musel jsem tedy vymyslet i podpěrný systém pro tento prototyp, jelikož v reálném použití je garnýž zavěšena na háčích, které jsou přivrtány do zdi.

Koupil jsem tedy garnýž, která se skládala ze dvou částí. Širší trubky a užší trubky, která šla do té širší zasunout jako možnost nastavit si délku garnýže. Já jsem širší trubku použil jako garnýž, a užší jsem rozřízl napůl, a použil jako stojnu podpěrného systému.

Na tyto stojny jsem z vrchu namodeloval podpěry, nesoucí systém automatického závěsu, a na spodní části jsem namodeloval stojany. Pro lepší stabilitu celého prototypu jsou na koncích stojanů vymodelovaná očka, kterými jsou provlečeny závitové tyče. Celý model je vidět na obrázku 6.9.



Obrázek 6.9: Finální model prototypu i se stojanem

6.4 Tisk na 3D tiskárně

Tisk na 3D tiskárně pro mě byl také novinkou. Byl jsem obeznámen se základními pravidly 3D tisku a že i při modelování je potřeba myslet na určité aspekty. Například dávat si pozor na to, že tisknutý tvar by měl mít vždy rovnou základovou plochu, která by zároveň měla při pohledu z vrchu být tou největší plochou. Tisk totiž probíhá po vrstvách, které se na sebe kupí. Kdyby se toto nedodržovalo, bylo by to jak stavět pyramidu od špičku k základu.

Pokud už by se přečnávající část na tisknuté věci nacházela, je potřeba udržovat mírný sklon od základu tisku až k přečnávající části - jako například u obráceného kužele, nebo by bylo potřeba využít podpěr, které by následně po tisku byly odříznuty.

Vlastnost, která byla potřeba nastavit až v samotném nastavení tisku, bylo procento výplně tisknutých částí. Zatímco u částí, které nejsou namáhány žádným tahem stačila hustota vnitřní části objektu (výplně) 15%, části tahem namáhané byly tisknuty s výplní 60%.

K tisku tohoto projektu byla použita 3D tiskárna Original Prusa MINI¹ a program PrusaSlicer²

¹<https://www.prusa3d.com/cs/produkt/stavebnice-3d-tiskarny-original-prusa-mini-2/>

²https://www.prusa3d.com/cs/stranka/prusaslicer_424/

Kapitola 7

Programové řešení

Pro úspěšné dokončení této práce už zbývalo jen vymyslet programové řešení. Práci jsem psal v programovacím jazyce C++ ve vývojovém prostředí `Arduino IDE`. Zvažoval jsem i vývoj ve vývojovém prostředí `ESP-IDF` od firmy `Espressif`, avšak potýkal jsem se s určitými problémy při práci s motorem a i když v `ESP-IDF` má uživatel větší kontrolu nad všemi probíhajícími procesy, rozhodl jsem, že pro vypracování tohoto projektu bude `Arduino IDE` dostačující.

Za použití objektově orientovaného programování jsou Všechny knihovny implementovány jako třídy, jejichž atributy reprezentují hodnoty využívané danou knihovnou a metody zajišťují práci s těmito hodnotami. Každá knihovna obsahuje hlavičkový soubor s deklarací třídy a zdrojový soubor s definicí jednotlivých metod.

7.1 Zpracování senzorových dat

Získání intenzity osvětlení zajišťuje knihovna `BH1750`¹, kdy volání metody `readLightLevel()` vrací intenzitu v jednotkách [lx].

Pro získání tepelné hodnoty z termistoru slouží knihovna `MyThermistor`. Zde jsou uloženy potřebné konstanty na převod detekovaného napětí na teplotu. Převod popsán v kapitole 4 - Tepelné čidlo. Klíčovou funkcí této knihovny je metoda `getTemperature()`, která vrací hodnotu ve stupních celsia.

7.2 Režimy systému okenních závěsů

Pro funkčnost **časového režimu** je implementována knihovna `MySchedule`. Je zapotřebí si pamatovat časový údaj s pozicí na kterou má být závěs roztáhnut. K tomu slouží třída `TimeAndPosition` jejíž jedna instance je rovna jednomu záznamu s atributy hodina, minuta a pozice. Pozice je ukládána v % z celkové délky závěsu. Hodnota 0 tedy znamená plně otevřeno a 100 zavřeno. Jelikož všechny z ukládaných údajů mají číselný rozsah, v konstruktoru této třídy jsou kontrolní podmínky správnosti vkládaných údajů. Třída `Week` slouží pro ukládání časových údajů u jednotlivých dní v týdnu. Každý ze 7mi atributů je implementován jako seznam prvků typu `TimeAndPosition`.

¹<https://github.com/claws/BH1750>

Jelikož mikročip neuchovává údaje o aktuálním čase, je pro zjištění času použita knihovna `ESP32Time`², která načítá čas z internetové sítě. Zde bylo potřeba v hlavičkovém souboru upravit v deklaraci konstrukturu parametr určující časové pásmo.

Světelný režim funguje na principu vyhodnocování aktuální polohy závěsů, intenzity osvětlení a hraniční hodnoty pro zatáhnutí. Pokud je závěs roztáhnut více než z 5-ti % celkové délky a zároveň intenzita osvětlení překročila hranici pro zatáhnutí, je závěs zatáhnut. Analogicky to funguje s automatickým roztáhnutím.

7.3 Pohyb motoru

Komunikaci s motorem zajišťuje knihovna `MyStepper`. Při její tvorbě sem se inspiroval principem krokování motoru z knihovny `SpeedyStepper`³. Zvažoval jsem využít dědičnost při tvorbě vlastní třídy pro ovládání motoru, avšak metody v knihovně `SpeedyStepper` byly příliš obecné a potřeboval jsem vložit do určitých metod podmínky navíc.

Samotný pohyb motoru - vykonání jednoho kroku zajišťuje metoda `processMovement()`, kdy počet kroků a požadovaná rychlost jsou nastaveny při inicializaci atributů. Zároveň vyhodnocuje vnější podmínky kdy pohyb zastavit například při detekci zarážky.

Metoda `curtainsInit()` slouží pro prvotní inicializaci závěsů po startu systému. Motorem je otáčeno tak dlouho dokud není detekována magnetická zarážka. Následně se vykoná 400 kroků v opačném směru, což odpovídá pohybu 1 cm, kdy již není magnet detekován a zde se nastaví počáteční bod. Odsunutí je zde proto, jelikož magnetická zarážka slouží pouze pro inicializaci a jako bezpečnostní prvek, že při pohybu nedošlo k chybě - viz kapitola 4.1 - Detektor úplného roztáhnutí/zatáhnutí. Tento postup je aplikován i na otáčení v druhém směru kde je nastaven koncový bod.

Pokud by byla při pohybu motoru detekována magnetická zarážka, spustí se znovu inicializace závěsů a následně jsou závěsy roztáhnuty do polohy kam bylo původně plánováno.

V knihovně se také nacházejí metody `timeCheck(lx)` a `lightCheck(week, day, hour, min)`, které obsluhují pohyb závěsů při aktivovaném časovém nebo světelném režimu.

7.4 Komunikace klient - server

Pro komunikaci klienta se serverem je použita knihovna `PubSubClient`⁴.

Pro úsošné navázání spojení je potřeba zavolat následující metody. Po připojení zařízení k Wi-Fi za použití knihovny `WiFiClient`⁵, je nastaven server ke kterému se chce klient připojit voláním metody `setServer(client, port)`. Důležité bylo prodloužit dobu, po kterou server nechá spojení aktivní metodou `keepAlive(sec)`. Výchozí hodnota je 15 vteřin, ale jelikož při pohybu závěsů nedochází k vzájemné komunikaci, server by klienta prohlásil za neaktivního a ukončil spojení. Následně je klient k serveru připojen metodou `connect(id, username, password)`.

²<https://github.com/kerikun11/ESP32Time>

³<https://github.com/Stan-Reifel/SpeedyStepper>

⁴<https://github.com/knolleary/pubsubclient>

⁵<https://www.arduino.cc/reference/en/libraries/wifi/>

Knihovna [MyMqtt](#) zajišťuje způsob komunikace mezi systémem a `Home-assistantem` - zkr. HA, za užití protokolu MQTT. Po startu systému je potřeba zaregistrovat jednotlivé režimy a senzory v HA pomocí objevovacích zpráv - *angl. discovery message*. Tato zpráva obashuje potřebné údaje jako typ periferie, na které adrese *angl. topic* bude komunikace probíhat a jak s hodnotami v komunikaci pracovat.

Dále se knihovna stará vytvoření obsahu zprávy ve správném formátu. Hodnoty jsou přeposílány ve formátu JSON⁶ a serializaci obstarává knihovna [ArduinoJson](#)⁷

7.5 Běh programu

Hlavním souborem je `curtains.ino`, který se stará o celý běh programu. Na začátku jsou definovány konstanty jednotlivých pinů a přihlašovací údaje pro připojení k Wi-Fi síti. Následují globální proměnné ukládající hodnoty připojených senzorů, polohu závěsů v procentech a informace o stavu časového a světelného režimu.

Po startu je vykonána funkce `setup()`, zajišťující nastavení celého systému. Kontroluje připojení periférií s prvním čtením hodnot 7.1. Zajistí konfiguraci pinů a připojení klienta k Wi-Fi a MQTT serveru s registrací a inicializací všech potřebných náležitostí 7.4. Nakonec provede prvotní inicializaci závěsů a závěsy zatáhne do 50-ti % jejich délky 7.3.

Po úspěšném nastavení systému je již stále dokola vykonávána funkce `loop()`. Zde je pravidelně kontrolováno připojení a v případě zjištění, že zařízení bylo odpojeno od sítě nebo serveru, proběhne znovupřipojení.

V pravidelných 15-ti vteřinových intervalech jsou snímány hodnoty z připojených senzorů. Po každém načtení nové hodnoty je v případě aktivovaného požadovaného režimu 7.2 vyhodnoceno, zda hodnoty nevyhovují podmínkám pro pohyb závěsů. Dále jsou tyto hodnoty odesílány na server, aby zobrazované údaje byly aktuální.

Po celou dobu je nasloucháno příchozím zprávám ze strany serveru. V případě příchozí zprávy je zavolána metoda `callback()`, zajišťující její zpracování podle adresy - *angl. topic* na kterou přišla, a jsou aktualizovány hodnoty v programu.

⁶<https://www.json.org/json-en.html>

⁷<https://arduinojson.org>

Kapitola 8

Zkonstruování prototypu a testování

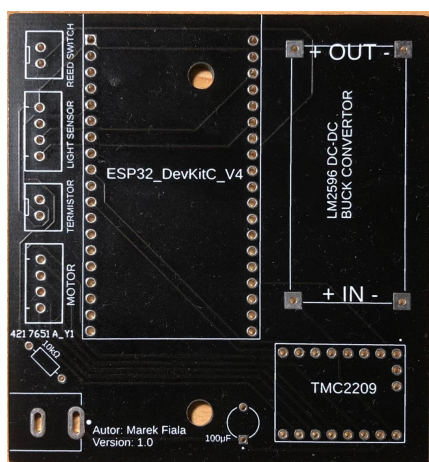
Po úspěšném vytvoření jednotlivých částí systému bylo potřeba prototyp systému sestavit a otestovat. Detailnější přehled snímků z postupu sestavování systému je v příloze B.

Také bylo nahráno demonstrační video hotového systému popisující jednotlivé funkce a režimy i s ukázkami. Odkaz zde: <https://youtu.be/f7Xdpjjuqes>

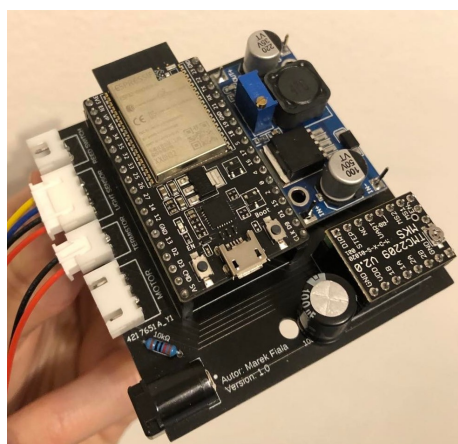
Pájení desky plošných spojů

Prvním krokem pájení jednotlivých součástek na desku plošných spojů bylo nastavení DC-DC Stepdown nastavitelného měniče na výstupní napětí 5V, aby nedošlo ke zničení mikročipu ESP32 po připojení ke vstupnímu 12V napětí. Poté byly připájeny všechny potřebné konektory, do kterých byly zasazeny jednotlivé moduly a v poslední řadě ochranný kondenzátor.

Protože jsem s podobným systémem zacházel poprvé, stalo se mi také, že jsem zničil jednu vývojovou sadu ESP32 nebo ovladač motoru. Toto bylo způsobeno především špatnou manipulací se systémem při testování, když byl zapojen do elektrické sítě a já jsem omylem spojil určité piny.



Obrázek 8.1: Vyrobena deska plošných spojů



Obrázek 8.2: Osazená deska plošných spojů

Konstruování prototypu

S postupem času jak se postupně tiskly jednotlivé části uchyvacích prvků, jsem systém začal skládat. Občas se vyskytl menší problém. Nejčastěji špatná velikost nějakého otvoru, ať už způsobená zadáním chybného údaje při modelování, nebo že při 3D tisku je potřeba počítat s určitou rezervou.

Největší překážkou při konstruování bylo spojování řemene. Řemen je totiž prodáván vcelku po 1 metru a při lepení se mi řemen 2x rozlepil. Ve výsledném prototypu je tedy spojovací místo vyztuženo ještě destičkami, které jsou k sobě stlačovány šrouby. Tyto destičky zároveň slouží jako tažné místo počátečního oka závěsů.

Po vytisknutí všech částí a smontování systému, jsem připevnil jazýčkové relé na své místo a napájel prodlužovací vodiče ke konektorům. Při upevňování jazýčkového relé bylo potřeba dávat pozor, jelikož skleněná trubička ve které se jazýčky nachází je velmi křehká. Zkonstruovaný prototyp je vidět na obrázku 8.3



Obrázek 8.3: Finální podoba zkonstruovaného modelu

Testování

Pro testování byl použit program MQTT Explorer¹, který umožňuje zobrazit veškeré MQTT spojení na dané IP adrese. V kombinaci s výpisovými zprávami, které program po připojení k počítači dokáže zobrazovat se jednalo o účinný ladící nástroj při finálním vývoji komunikace systému s Home Assistantem.

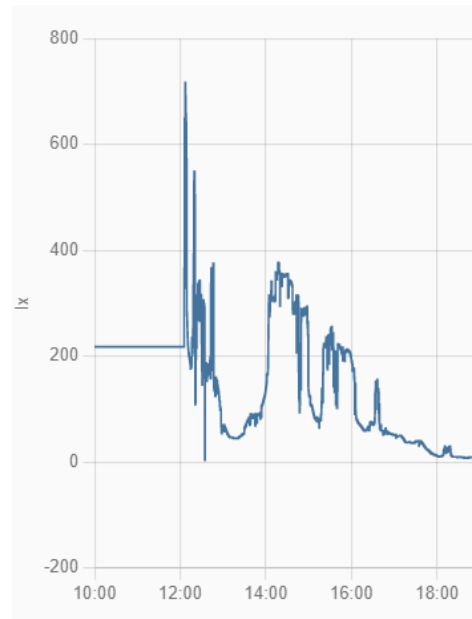
¹<http://mqtt-explorer.com>

```

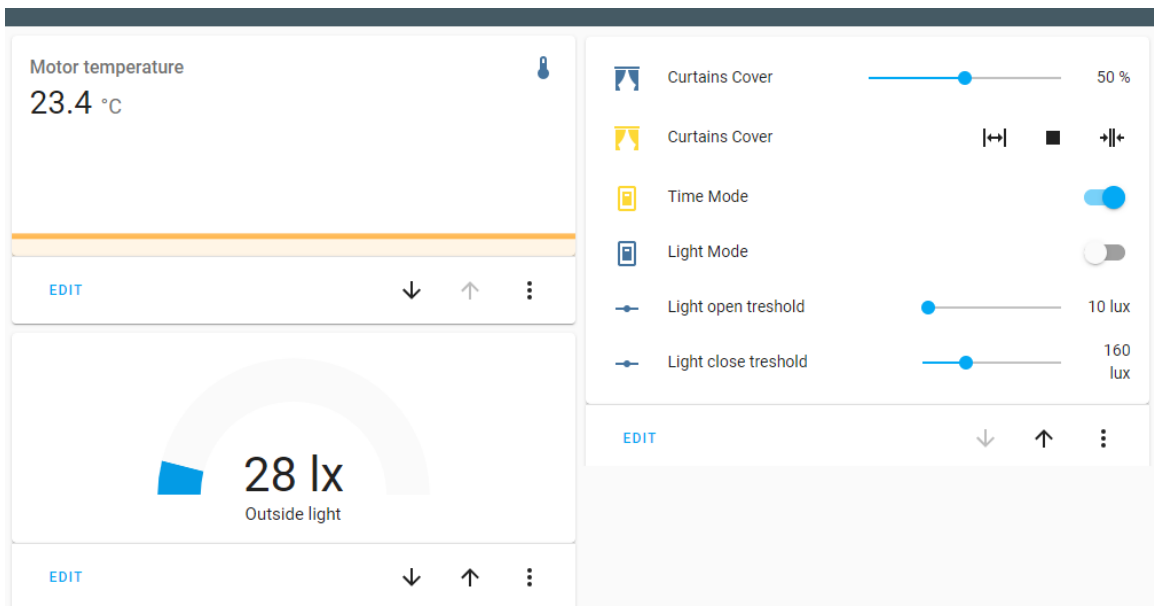
-> .....Connected to Wi-Fi
-> Connecting to MQTT
-> .Connected to MQTT
-> Curtains discovery message send
-> Light discovery message send
-> Temperature discovery message send
-> Light mode discovery message send
-> Time mode discovery message send
-> Light Open Treshold discovery message send
-> Light Close Treshold discovery message send
-> Light Mode default value published
-> Time Mode default value published
-> Position default value published
-> Light open treshold default value published
-> Light close treshold default value published
-> Sensors value succesfully updated
-> Curtains initialization
-> Succesfully opened
-> Succesfully closed
-> Curtains length [mm]12
-> home [steps]: 0
-> end [steps]: 463
-> Moving to [steps]: 232
-> Message arrived on topic: home/curtains/position. Message: 50
-> Light [lux]: 3
-> Temperature [°C]:
-> 23.32
-> Sensors value succesfully updated

```

Obrázek 8.4: Ukázka výpisových zpráv



Obrázek 8.5: Vykresení historie světelného senzoru v systému Home Assistant



Obrázek 8.6: Grafické rozhraní v systému Home Assistant

Kapitola 9

Závěr

Cílem této práce bylo sestavit a vytvořit kompletní systém chytrých okenních závěsů. Pro jeho dosažení bylo potřeba nejdříve pochopit, co takový systém obsahuje, jak může fungovat a proč se nazývá chytrý. Také je v práci popsáno, co je to systém domácí automatizace a jak přes něj jednotlivá zařízení mohou komunikovat.

Bylo také potřeba provést analýzu trhu a zjistit, jaká podobná řešení již existují, jaké mají vlastnosti a kolik pořízení takového systému stojí.

Po zjištění, co systém chytrých okenních závěsů je, bylo potřeba si jej rozebrat na jednotlivé části zajišťující jeho funkčnost. Bylo nutné provést zkoumání jednotlivých možností nástrojů, které jsou na trhu dostupné a mohou činnosti těchto částí vykonávat. Byly zkoumány typu motorů zajišťující pohybovou činnost systému a jak motory ovládat, porovnávány senzory, které by měl systém obsahovat pro splnění požadovaných vlastností automatizace a jak se senzory komunikovat. Byl proveden výběr řídicí jednotky ovládající celý systém a zhodnoceno, jak bude uživatel systém ovládat.

Následně byl vytvořen návrh obvodového řešení. Prvně byla otestována funkčnost vybraných nástrojů a provedeno jejich první propojení mezi sebou do celkového systému, za použití vodičů a nepájivého pole. Po úspěšném otestování byl namodelován motiv desky plošných spojů kopírující propojení jednotlivých pinů prototypového řešení, kde byly přidány i konektory pro jednodušší manipulaci se systémem. Následně byl vytvořen 3D model osazené desky plošných spojů pro ověření kompatibilního fyzického rozložení jednotlivých komponentů.

Po úspěšném vytvoření modelu osazené desky následoval výběr mechanických komponent pro přenesení otáčivého pohybu na pohyb posuvný. Součástí práce bylo potřeba vymyslet návrh, jak motor k závěsům upevnit. Jednotlivé části návrhu byly namodelovány a celý prototyp zde byl vizualizován v programu Autodesk Fusion 360. V rámci modelování byla vytvořena i krabička na desku plošných spojů. Všechny jednotlivé části byly následně vytisknuty na 3D tiskárně.

Implementace programového řešení byla popsána v kapitole 7. Zde je ze začátku popsán výběr vývojového prostředí, na který navazuje popis vytvořených i použitých knihoven. Ke konci kapitoly je popsán princip hlavního programu systému, jak probíhá inicializace jednotlivých částí systému, komunikace systému se serverem domácí automatizace a princip činností periodicky se opakujících v hlavní smyčce programu.

Protože součástí zadání práce bylo i zkonstruování fyzického prototypu systému, je v kapitole 8 popsána jeho realizace s následným otestováním. Kapitola obsahuje i odkaz na video s demonstrací systému.

V případě dalšího vývoje tohoto projektu by mohla být zprovozněna detekce zaseknutí závěsu, případně podpora manuálního rozpohybování závěsů kdy by potom systém pohyb sám dokončil. Z fyzické části by se mohlo jednat o namodelování uchycení na jiný tvar držáku závěsů - například kolejnice.

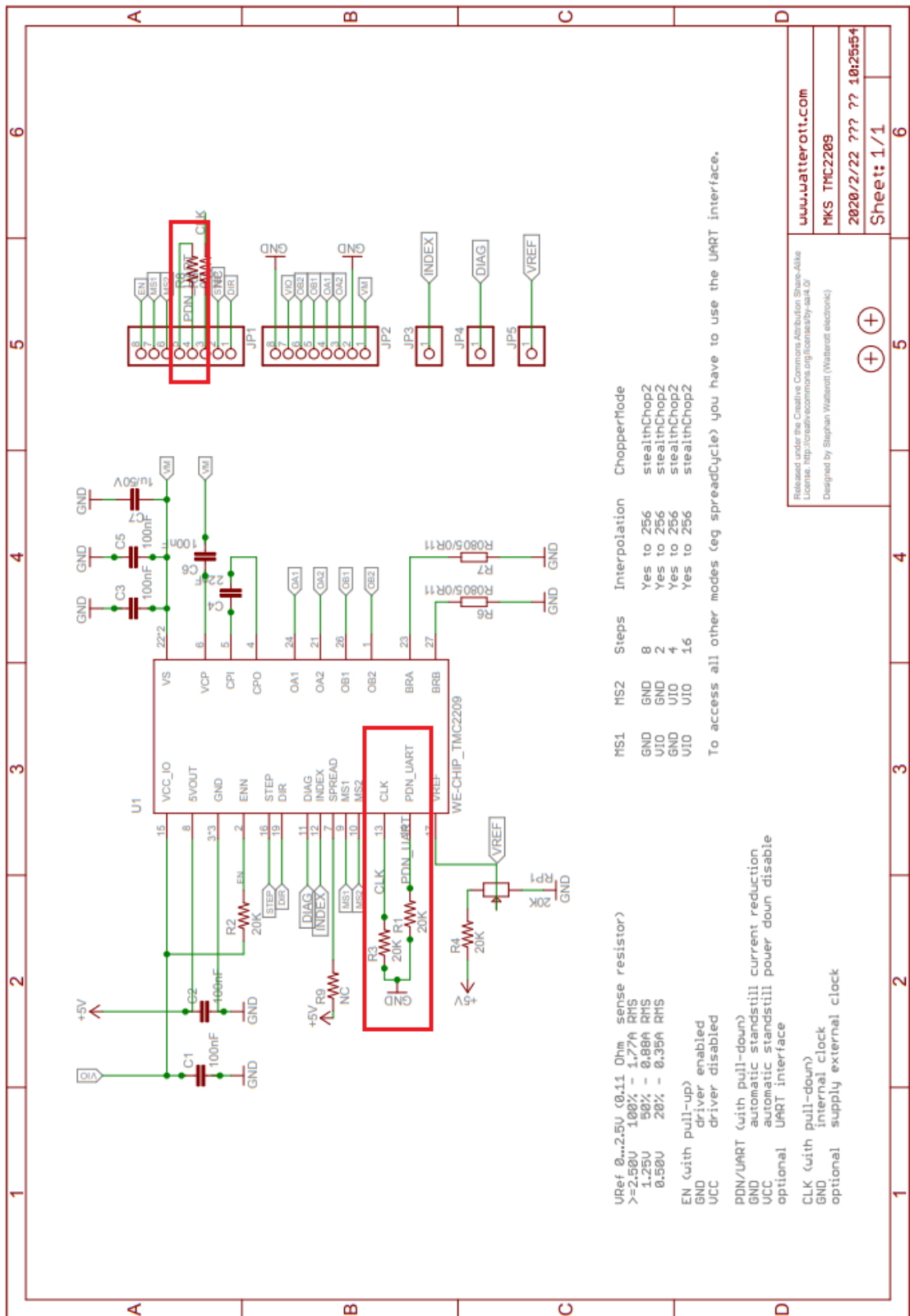
Literatura

- [1] ADA. *TSL2561 Luminosity Sensor* [online]. 1. vyd. ©Adafruit Industries, 2021, 2021-11-15 [cit. 202-04-21]. Dostupné z: <https://www.mouser.com/datasheet/2/737/tsl2561-932888.pdf>.
- [2] DR. OVIDIU VERMESAN, B. *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. 1. vyd. River Publishers, 2013. ISBN 80-7302-049-1.
- [3] DR. OVIDIU VERMESAN, B. *Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems*. 1. vyd. River Publishers, 2014. ISBN 9788793102941.
- [4] JAŠEK, D. *VZDUŠNÝ HOKEJ - REALIZACE A NÁVRH AUTOMATICKÉHO HRÁČE*. Brno, CZ, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Dostupné z: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=213962.
- [5] MAXIM INTEGRATED PRODUCTS, I. *DS18B20-PAR 1-Wire Parasite-Power Digital Thermometer* [online]. 1. vyd. Maxim Integrated Products, Inc., 2012, 2019-04 [cit. 202-04-28]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>.
- [6] REMBOR, K. *Adafruit VEML7700 Ambient Light Sensor* [online]. 1. vyd. ©Adafruit Industries, 2021, 2022-02-16 [cit. 202-04-26]. Dostupné z: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-veml7700.pdf>.
- [7] ROHM Co., L. *Digital 16bit Serial Output Type Ambient Light Sensor IC BH1750FVI* [online]. 1. vyd. ROHM Co., Ltd., 2009, 2021-04 [cit. 202-04-21]. Dostupné z: <https://e-radionica.com/productdata/BH1750FVI.pdf>.
- [8] SANTOS, R. *What is MQTT and How It Works* [online]. 2017. 2021-16-12 [cit. 2022-05-07]. Dostupné z: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>.
- [9] TKOTZ, K. Motory a generátory. In: TKOTZ, K., ed. *Příručka pro elektrotechniku*. 2. vyd. Europa - Sobotáles cz. s.r.o., únor 2006, s. 450–464. ISBN 80-86706-13-3.
- [10] VISHAY INTERTECHNOLOGY, I. *NTC Thermistors, Radial Leaded, Standard Precision* [online]. 1. vyd. VISHAY INTERTECHNOLOGY, INC., 2016, 2016-12-22 [cit. 202-04-27]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.118-044.1.pdf>.

Příloha A

TMC2209 schema

¹https://github.com/makerbase-mks/MKS-StepStick-Driver/blob/master/MKSTMC2209/MKSTMC2209V2.0_001/MKSTMC2209V2.0_001SCH.pdf

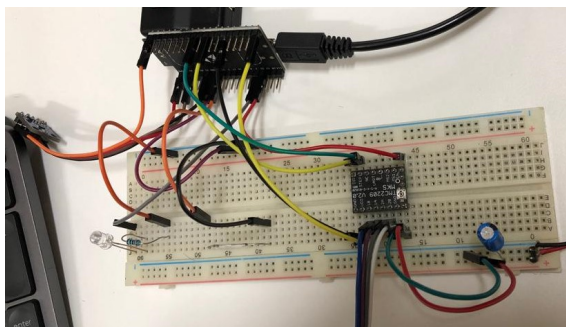


Released under the Creative Commons Attribution Share-Alike License. <http://creativecommons.org/licenses/by-sa/4.0/>
 Designed by Stephan Wellerst (Haberst electronic)
www.watterott.com
 MKS TMC2209
 2020/2/22 ??? ?? 10:25:54
 Sheet: 1/1

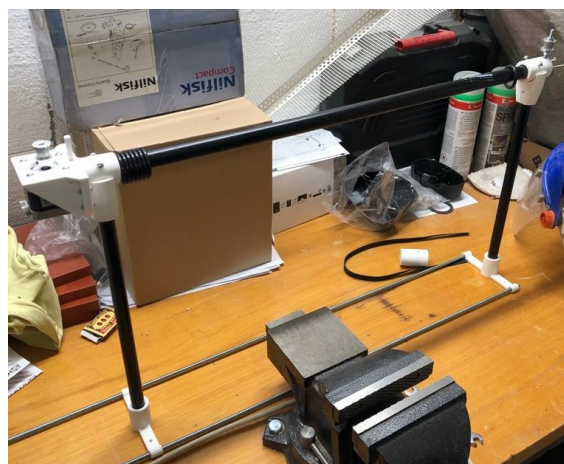
Obrázek A.1: TMC2209 Schema s vyznačnými částmi pro UART(Převzato z webu [1])

Příloha B

Dokumentace průběhu tvorby



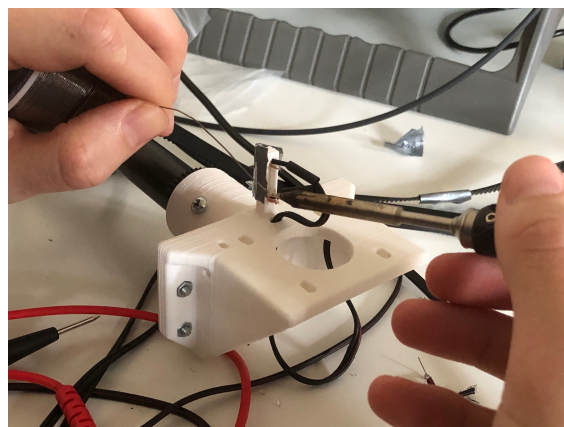
Obrázek B.1: Zapojené nepájivé pole



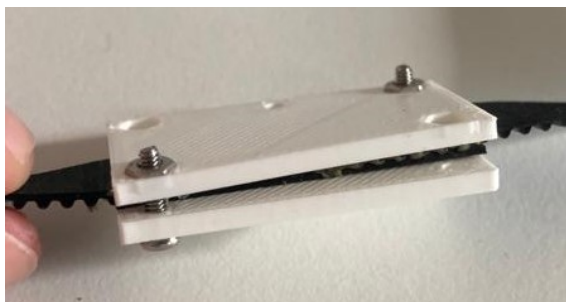
Obrázek B.2: Sestavování stojanu



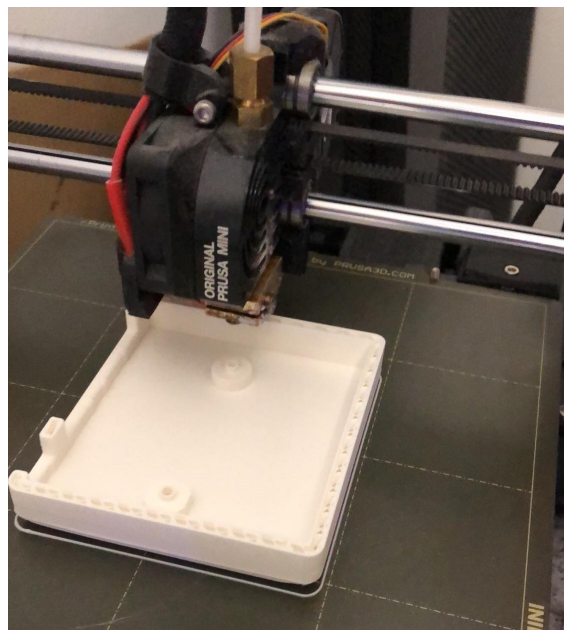
Obrázek B.3: Osazená deska v krabičce



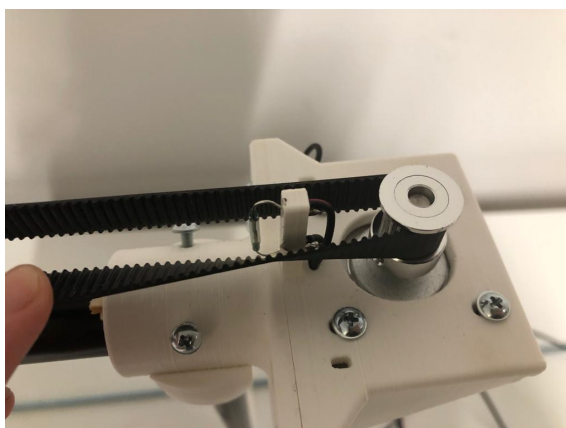
Obrázek B.4: Průběh pájení jazýčkového relé



Obrázek B.5: Detail ztužovací destičky u spoje řemenu



Obrázek B.6: 3D tisk krabičky



Obrázek B.7: Detail řemenice a jazýčkového relé



Obrázek B.8: Detail kladky s napínacími šrouby

Příloha C

Zprovoznění systému

Pro zprovoznění systému je potřeba provést posloupnost následujících kroků.

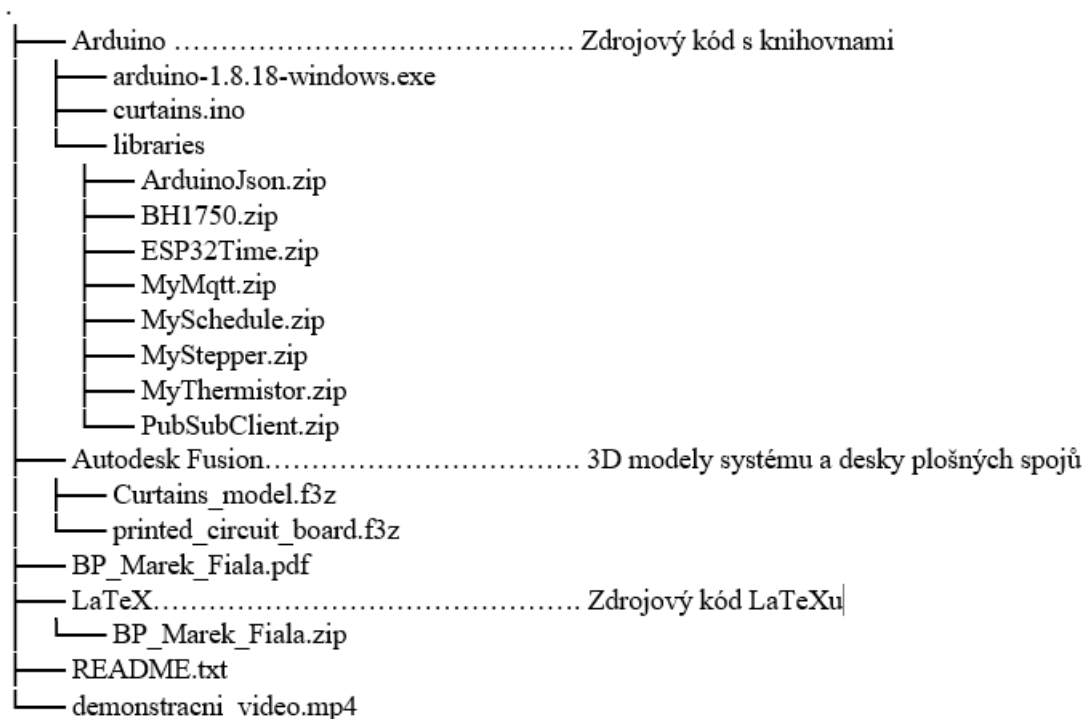
1. Instalace operačního systému `Home-assistant`.
 - Systém domácího řízení `Home-assistant` funguje jako server na připojené Wi-Fi síti. Návod na instalaci zde:^[1]
 - Při tvorbě této práce běžel `Home-assistant` na Raspberry Pi 4, ale instalační průvodce nabízí mnoho jiných možností.
 - Přidat rozšíření `Mosquito MQTT Broker` do `Home-assistantu`. Návod zde: ^[2]
2. Instalace a nastavení vývojového prostředí `Arduino IDE`.
 - Spuštění instalačního souboru `arduino-1.8.18-windows.exe` - součást CD.
 - Po nainstalování otevřít soubor `curtains.ino` v nainstalovaném `Arduino IDE`
 - Instalace balíčků podpory mikročipů `ESP32`. Návod zde: ^[3]
 - Import knihoven ze souboru `libraries`.
 - V IDE v horní liště v záložce `Projekt -> Přidat knihovnu -> Přidat .ZIP knihovnu...`
 - Vybrat všechny položky `.zip` z adresáři `libraries`
 - Nastavení přístupových údajů v kódu pro připojení k Wi-Fi a `MQTT Serveru`.
 - V souboru `curtains.ino`, na řádcích 28 - 33 je potřeba zadat údaje pro bezdrátové připojení.
 - Připojení `ESP32`, překlad a nahrání programu.
 - Po úspěšném nahrání programu na mikročip se klient připojí a provede svou inicializaci v systému `Home-assistant`. (bohužel uživatel musí pro zobrazení změn `Home-assistant` restartovat)
 - Se zaregistrovanými entitami si uživatel může grafické rozhraní nastavit podle sebe.

¹<https://www.home-assistant.io/installation/>

²<https://cyan-automation.medium.com/setting-up-mqtt-and-mosquitto-in-home-assistant-20eb810a91e6>

³<https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>

V adresáři Autodesk Fusion se nachází model celého systému pro případný tisk na 3D tiskárně, a také schéma zapojení s výsledným otiskem desky plošných spojů. Na práci s těmito soubory je potřeba mít nainstalován Autodesk Fusion 360⁴



Obrázek C.1: Hierarchické uspořádání přiloženého CD

⁴<https://www.autodesk.cz/products/fusion-360/overview>