



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INTELIGENTNÍCH SYSTÉMŮ**

DEPARTMENT OF INTELLIGENT SYSTEMS

**ŘÍZENÍ MANIPULÁTORU PRO SNÍMÁNÍ SÍTNICE OKA**

MANIPULATOR CONTROL FOR ACQUISITION OF THE EYE RETINA

**DIPLOMOVÁ PRÁCE**

MASTER'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Bc. PETR MALANÍK**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Prof. Ing., Dipl.-Ing. MARTIN DRAHANSKÝ, Ph.D.**

BRNO 2020

## Zadání diplomové práce



23217

Student: **Malaník Petr, Bc.**

Program: Informační technologie    Obor: Počítačové a vestavěné systémy

Název: **Řízení manipulátoru pro snímání sítnice oka**  
**Manipulator Control for Acquisition of the Eye Retina**

Kategorie: Vestavěné systémy

Zadání:

1. Prostudujte literaturu týkající se snímání sítnice lidského oka, zaměřte se na nemydriatické fundus kamery a rovněž nastudujte možnosti ovládání pěti-osého manipulátoru.
2. Navrhněte a zkonstruujte elektroniku, která bude schopna pěti-osým manipulátorem pohybovat tak, aby bylo možné nasnímat sítnici oka pomocí kamery s připojenou optikou.
3. Navrhněte a realizujte systém pro řízení zaostření kamery, osvětlení scény a záměrného bodu.
4. Navrhněte protokol, přes který bude možné z řídicího počítače ovládat pohyb manipulátoru a chování ostatních prvků.
5. Výše uvedenou realizaci otestujte a dosažené výsledky shrňte.

Literatura:

- HÁJEK Josef a DRAHANSKÝ Martin. Recognition-Based on Eye Biometrics: Iris and Retina. *Biometric-Based Physical and Cybersecurity Systems*. Cham: Springer International Publishing, 2018, s. 37-102. ISBN 978-3-319-98733-0.
- DRAHANSKÝ Martin. Recognition of Eye Characteristics. *Machine Learning and Biometrics*. London: InTech - Open Access Publisher, 2018, s. 7-35. ISBN 978-1-78923-590-6.
- Manuál k manipulátoru TESCAN.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 a 2.

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

Vedoucí práce: **Drahanský Martin, prof. Ing., Dipl.-Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 3. června 2020

Datum schválení: 31. října 2019

## Abstrakt

K využití snímku sítnice pro biometrické, případně lékařské účely je nutné, aby byl snímek co nejkvalitnější a ideálně zabíral co největší plochu. Je tedy nezbytné dostat snímací zařízení do co nejvhodnější polohy a při tom zajistit jeho stabilitu. Za tímto účelem je využita laboratorní manipulační platforma. Pohyb platformy je zajištěn krokovými motory. Pro jejich ovládání byla vyvinuta řídicí deska osazená mikrokontrolérem, který umožňuje provádění velmi jemných pohybů. Teoretická přesnost manipulátoru je až 20 nm. Jelikož je nutné pro snímání sítnice osvětlení, obsahuje řídicí systém i nastavitelné osvětlovací prvky v infračerveném a viditelném spektru. Výsledný systém umožňuje nasnímání sítnice oka z více úhlů, tím efektivně zvyšuje plochu, na které je možné dále hledat příznaky nemocí, případně biometrické rysy.

## Abstract

To use the image of the retina for biometric or medical purposes, it is necessary that the image is of the highest quality and ideally covers the largest possible area. It is therefore necessary to move the sensing device to the most suitable position while ensuring its stability. A laboratory handling platform is used for this purpose. The platform is moved by stepper motors. A control board equipped with a microcontroller has been developed for their control, which allows very fine movements. The theoretical accuracy of the manipulator is up to 20 nm. Since it is necessary for sensing, to have proper retina illumination, the control system also includes adjustable illumination elements in the infrared and visible spectrum. The resulting system allows scanning of the retina of the eye from multiple angles and thereby effectively increase the area on which it is possible to further look for diseases or biometric features.

## Klíčová slova

sítnice, krokové motory, vestavěné systémy, nemydriatická fundus kamera, mikrokontrolér, STM32, řízené osvětlení scény, návrh desky plošných spojů

## Keywords

retina, stepper motors, embedded systems, non-mydratic fundus camera, microcontroller, STM32, controlled scene lighting, PCB design

## Citace

MALANÍK, Petr. *Řízení manipulátoru pro snímání sítnice oka*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Ing., Dipl.-Ing. Martin Dražanský, Ph.D.

# Řízení manipulátoru pro snímání sítnice oka

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Prof. Ing., Dipl. -Ing. Martina Drahanského, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Petr Malaník  
3. června 2020



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Teorie</b>	<b>4</b>
2.1	Snímání sítnice . . . . .	4
2.1.1	Zařízení pro snímání . . . . .	5
2.1.2	Fundus kamera . . . . .	5
2.2	PID regulátor . . . . .	6
2.2.1	Softwarová varianta PID regulátoru . . . . .	8
2.3	Měření proudu na plošném spoji . . . . .	8
2.3.1	Měření pomocí Hallova jevu . . . . .	9
2.4	Řízení krokových motorů . . . . .	11
2.4.1	Porovnání variant budičů . . . . .	11
2.5	Budič krokových motorů - L6470 . . . . .	14
2.5.1	Výkonová část . . . . .	14
2.5.2	Řídicí integrovaný systém . . . . .	16
2.5.3	Komunikační rozhraní . . . . .	17
2.5.4	Funkce . . . . .	19
2.5.5	Kompenzace chyb a optimalizace . . . . .	20
2.6	Řízení LED osvětlení . . . . .	21
2.6.1	Impulsní zdroje . . . . .	22
<b>3</b>	<b>Návrh a realizace hardwaru</b>	<b>23</b>
3.1	Konstrukce manipulátoru . . . . .	23
3.1.1	Fundus kamera . . . . .	23
3.2	Požadavky na přesnost . . . . .	25
3.3	Porovnání budičů krokových motorů . . . . .	26
3.3.1	Výsledky . . . . .	27
3.4	Napájení zařízení . . . . .	28
3.5	Základní deska . . . . .	28
3.5.1	Komunikace a diagnostika . . . . .	28
3.5.2	Napájení a spotřeba . . . . .	29
3.5.3	Budiče krokových motorů . . . . .	32
3.5.4	Kontrola a řízení teploty . . . . .	34
3.5.5	Záměrný bod . . . . .	35
3.6	Osvětlení sítnice . . . . .	36
3.6.1	Proudový zdroj pro LED - NCL30160 . . . . .	36
3.7	Návrh plošného spoje . . . . .	39
3.7.1	Návrh v programu KiCAD . . . . .	40

<b>4</b>	<b>Návrh a implementace softwaru</b>	<b>41</b>
4.1	Invocation wrapper . . . . .	42
4.2	Systém přerušení . . . . .	43
4.3	Struktura zařízení . . . . .	43
4.3.1	Rozhraní příkazové řádky . . . . .	44
4.3.2	Souborový systém . . . . .	45
4.4	Součásti zařízení . . . . .	46
4.4.1	Mikrokontrolér . . . . .	47
4.4.2	Komponenty . . . . .	47
4.4.3	Nástroje . . . . .	48
4.4.4	Aplikace . . . . .	49
4.5	Komunikační protokol . . . . .	49
4.5.1	Ovládací aplikace . . . . .	50
4.5.2	Příkazy pro pohyb . . . . .	50
4.5.3	Příkazy osvětlení a záměrný bod . . . . .	52
4.5.4	Ostatní příkazy . . . . .	53
4.6	Logování . . . . .	53
4.7	Zobrazení stavu zařízení . . . . .	54
4.8	Kompilace a nahrání do mikrokontroléru . . . . .	55
<b>5</b>	<b>Závěr</b>	<b>56</b>
5.1	Výsledná přesnost zařízení . . . . .	56
5.2	Výsledná spolehlivost zařízení . . . . .	58
5.3	Výsledky osvětlovací části zařízení . . . . .	58
5.3.1	Zaměřovací bod . . . . .	59
5.4	Návrhy na vylepšení . . . . .	59
5.4.1	Doplňující Flash paměť . . . . .	59
5.4.2	Napájení přes konektor USB-C . . . . .	60
5.4.3	Rozšiřující konektor . . . . .	60
5.4.4	Přesun knihovny na RTOS . . . . .	60
	<b>Literatura</b>	<b>62</b>
<b>A</b>	<b>Schéma základní desky</b>	<b>66</b>
<b>B</b>	<b>Seznam použitých součástek</b>	<b>87</b>
<b>C</b>	<b>Osazovací plán základní desky</b>	<b>91</b>

# Kapitola 1

## Úvod

Pro snímání sítnice oka je nutné zajistit přesné umístění snímacího zařízení tak, aby bylo možné skrz zornici získat co nejkvalitnější snímek. Pro pořízení snímku musí být sítnice ozářena světlem, jinak by výsledný snímek byl velmi tmavý a nebyl by dále použitelný pro získání biometrických rysů, případně pro lékařskou analýzu. Kvůli kvalitnímu snímku musí být snímací zařízení umístěno na stabilní konstrukci. S touto konstrukcí musí být možné manipulovat tak, aby se snímací zařízení dostalo do správné pozice proti oku, jelikož hlava snímaného člověka je většinou pevně zafixovaná na jednom místě. Kvůli požadované přesnosti je manuální manipulace s takovými zařízeními velice zdoluhavá. Cílem této práce je návrh a realizace zařízení, které celý tento proces zautomatizuje.

Kromě získání snímku, který je pravoúhlý na osu oka, je vhodné získat i další snímky, které neobsahují hlavní část sítnice, ale spíše okrajové části. K tomu je nutné kameru naklonit tak, aby optická osa kamery stále procházela skrz relativně malý průchod vytvářený zornicí. Postupně lze různým nakloněním získat větší počet snímků, které pokrývají výrazně větší plochu než jen jeden snímek. Tyto snímky by poté mohly být spojeny do jednoho, čímž se zvýší plocha dostupná pro analýzu. Plocha dostupná pro analýzu by mohla být rozšířena až dvojnásobně oproti původní.

Realizace zařízení se skládá z tvorby základní desky elektroniky, softwaru pro ovládání elektroniky a tím i celého zařízení, a následně kompletace zařízení a jeho otestování. Aby bylo možné vyrobit základní desku, je nutné určit ideální dostupné součástky a vybrat ty, které nejlépe odpovídají specifikovanému účelu a na nich založit návrh plošného spoje. Jedná se tedy o součástky sloužící pro řízení krokovým motorů, kontrolu osvětlení a záměrného bodu. Problém návrhu je ve spolehlivosti řízení krokových motorů. Přestože jsou teoreticky přesné, vyžadují specifické řízení jejich chodu.

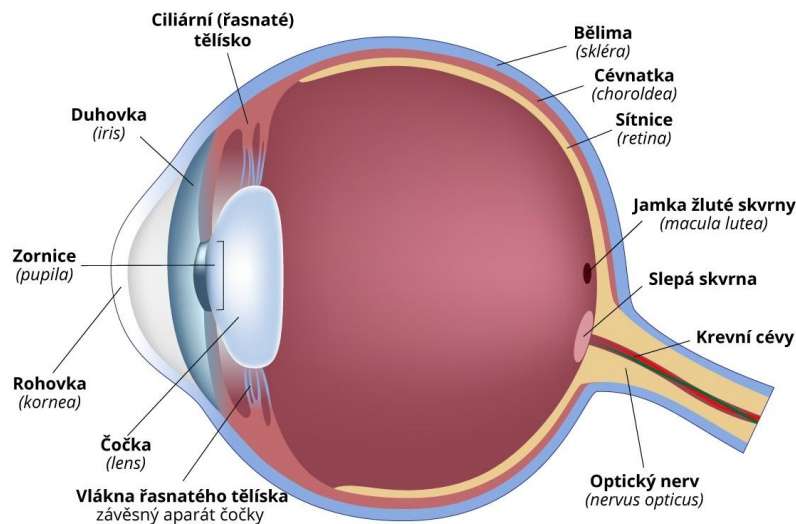
Dalším problémem je hlučnost a potenciálně i teplota, jelikož motory a ostatní elektronika mají nezanedbatelný příkon. Je tedy nutné tyto jevy minimalizovat, aby neměly vliv na funkci celého zařízení, případně na uživatelský komfort. A to i z toho důvodu, že snímaný člověk bude muset přiložit oko ke snímači, což může být pro část osob nepříjemná situace, zvláště u hlučného zařízení, které bude do oka blikat silným světlem. Stejně tak se může jednat o potenciálně nebezpečnou situaci, jelikož světlo o vysoké intenzitě může způsobit i dlouhodobé poškození sítnice. Kromě poškození světlem by se mohlo jednat i o zranění způsobené mechanikou zařízení. Všechny tyto problémy bude nutné vzít při návrhu do úvah a podle nich upravit výsledné zařízení.

# Kapitola 2

## Teorie

### 2.1 Snímání sítnice

Sítnice oka slouží pro vnímání dopadajícího světla a nachází se uvnitř oka na jeho zadní stěně. Skládá se z husté sítě světlocitlivých buněk, které absorbují dopadající fotony. Světlocitlivé buňky jsou rozděleny do dvou skupin. První jsou tyčinky, které jsou citlivé na nízkou intenzitu světla a zajišťují vnímání jasu. Druhé jsou čípky, které jsou citlivé na vlnovou délku a určují tedy barvu. Dále se sítnice skládá z optického disku, žluté skvrny a cév. Optický disk je oblast, kde do oka vstupuje zrakový nerv, tato část neobsahuje téměř žádné buňky a vidění je v tomto místě nejslabší. Další částí je žlutá skvrna, která je naopak oblast nejostřejšího vidění. Ve středu žluté skvrny se nachází fovea, kde je nejhustější koncentrace čípků [33]. Aby bylo zajištěno prokrvení sítnice, obsahuje systém cévy, které jsou nejvíce koncentrovány okolo optického disku a rozšiřují se k okrajům sítnice. Stavba lidského oka je na obrázku 2.1.



Obrázek 2.1: Stavba oka, převzato z [51]

Cílem snímání je získání obrazu, který obsahuje co nejvíce specifických objektů, jako je optický disk, žlutá skvrna a cévní řečiště. Kromě těchto rysů, které je možné použít pro biometrii, může sítnice obsahovat vady, které jsou spojeny s očními nemocemi. Obor medicíny zabývající se analýzou a léčbou očních vad a nemocí se nazývá oftalmologie.

### 2.1.1 Zařízení pro snímání

Jako základní zařízení pro snímání lze použít šterbinovou lampu. Jedná se o upravený mikroskop, který umožňuje osvětlení sledované plochy tak, že osvětlující světelné paprsky vstupují do oka rovnoběžně s pozorovaným odrazem. Toto je zajištěno většinou pomocí polopropustného zrcadla, které je umístěno v dráze pohledu mikroskopu pod úhlem  $45^\circ$ . Na toto zrcátko následně svítí zdroj světla, které se odráží od zrcátka kolmo vůči ose původního záření. Aby nebyla osvětlena celá plocha, prochází světlo šterbinou, která omezuje rozptyl světla. Hlavním účelem je snímání přední části oka tedy rohovky, duhovky a čočky. Při pozorování se využívá buď světlo přímo generované lampou a dopadající na přední část oka, nebo světlo zpětně odražené od sítnice oka a dopadající tak ze zadní strany. Zvětšení se většinou pohybuje od pětinasobného až po čtyřicetinasobné. Při využití širokoúhlé čočky lze použít šterbinovou lampu pro snímání sítnice oka, používá se v tomto případě výhradně přímé dopadající světlo. Širokoúhlé čočky ovšem nemají tak vysoké zvětšení [33]. Nevýhodou tohoto zařízení je nutnost maximálně rozšířené zornice, čehož je běžně dosahováno pomocí aplikace kapek, které roztahují zornici.

Dalším zařízením je oftalmoskop. Ten využívá bočního světla generovaného souběžně s osou pohledu, jehož ohnisko se následně nachází v ose oka. Souběžně s osou oka je možné pozorovat osvětlenou část sítnice. Opět se využívá kapek k rozšíření zornice, aby se tak zvětšila pozorovaná plocha. Oftalmoskopy lze rozdělit do dvou skupin a to přímé a nepřímé.

Přímé jsou většinou menší konstruované pro pozorování jedním okem ze vzdálenosti okolo 20 cm. Jejich zvětšení je zhruba patnáctinásobné [34]. Jsou konstruované bez stabilizační struktury pro držení v ruce. Většinou se používá pro základní vyšetření, nikoli pro pokročilou diagnózu. Jen velmi malé množství modelů umožňuje zaznamenávat snímanou sítnici. Ukázka přímého oftalmoskopu je na obrázku 2.2a.

Nepřímé oftalmoskopy většinou umožňují pozorovat větší oblast sítnice, ovšem s menším zvětšením než u přímého oftalmoskopu. Vyrábí se jako speciální brýle, kdy je možné pozorovat sítnici pomocí optického systému. Také obsahují osvětlení, které je směřováno skrz čočky do optické osy snímaného oka. Často obsahují infračervené i viditelné osvětlení [42]. Většinou umožňují zaznamenání snímku. Pozorování se provádí z co nejmenší vzdálenosti a úhel zorného pole je mezi 3 a 16 stupni [47].

### 2.1.2 Fundus kamera

Jedná se o pokročilejší zařízení využívající většinou digitálního snímacího prvku pro zachycení obrazu, který obsahuje sítnici. Mechanickou stavbou se podobá šterbinové lampě, je ovšem primárně určena pro snímání sítnice. Světlo ovšem může ozařovat celou sítnici, díky čemuž může být získán snímek větší plochy než při použití šterbinové lampy. Snímání a osvětlení probíhá souběžně. Osvětlení sítnice se provádí jen po velmi krátkou dobu, jelikož zdroj osvětlení musí být dostatečně silný, aby se od sítnice odrazil obraz zpět ke snímacímu zařízení. Odraz světla je komplikovaný kvůli velmi nízké odrazivosti sítnice [27]. Pro osvětlení sítnice se využívají dva způsoby odlišující se polohou, kde je v rámci snímací sestavy umístěno zrcadlo odrážející světlo dovnitř oka. První variantou je umístění zrcadla mimo snímací systém, jedná se tedy o externí osvětlení. Osa zdroje světla je v obou případech kolmá k ose snímacího systému. Externí osvětlení je odráženo pomocí hranolu směrem dovnitř oka [43]. Druhou variantou je umístění odrazného prstence, na který dopadá světlo směřující rovnoběžně s osou snímacího jednotky. Paprsek snímaného světla prochází středem prstence. Tento systém ovšem nemá schopnost efektivně eliminovat zpětné odrazy [53]. Externí osvětlení je jednodušší, ovšem zabírá více prostoru. Interní osvětlení je složitější na



(a) Přímý oftalmoskop [20]



(b) Fundus kamera [23]

Obrázek 2.2: Zařízení pro snímání sítnice

realizaci a obsahuje více problémů s odlesky. Blokové schéma fundus kamery s externím osvětlením, která je použita v realizaci projektu je v sekci 3.1.1, obrázek 3.2. Na obrázku 2.2b je ukázka komerčně dostupné fundus kamery.

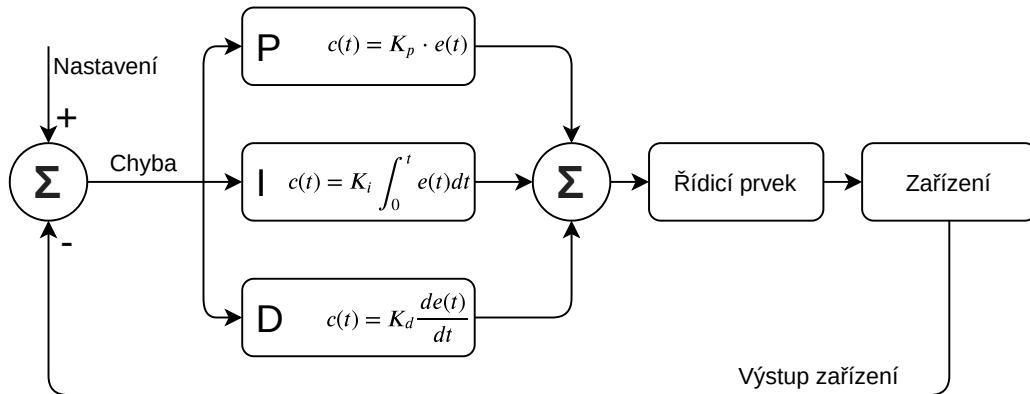
Při snímání se zornice oka umísťuje co nejbližší čočce, která směřuje světlo skrz zornici. Díky tomu je možné zajistit dostatečný zorný úhel i bez rozkapávání oka. Díky známé vzdálenosti snímacího zařízení a zornice lze kontrolovat její průměr. Pro zajištění rovnoběžnosti osy oka s osou snímacího systému se využívá záměrného bodu, jehož poloha by měla být ve středu snímacího zařízení. To je ovšem technologicky velmi složité zajistit, proto se využívá dalšího polopropustného zrcátka, na které je tento bod promítnut.

## 2.2 PID regulátor

Regulátory obecně slouží k ovládní výstupního jevu systému. Většinou funguje tento regulátor jako součást kontrolní smyčky. Sleduje tedy výstup zařízení, který slouží jako zpětná vazba a na základně něj provede akci, aby změnil výstup na požadovanou hodnotu. Například pokud je výstupní veličina blízko hodnotě požadované, bude výstup regulátoru malý nebo žádný. Samotná velikost chyby, tedy rozdílu mezi požadovanou a skutečnou hodnotou, většinou není dostačující k přesnému ovládní systému. Takto popsaný systém odpovídá P regulátoru.

Jednou z nejrozšířenějších variant regulátorů je PID regulátor [40], který kromě proporcionální složky (P) přidává také složky integrační (I) a derivační (D), od jejichž zkratk je odvozen název regulátoru. Často se také využívá jen část regulátoru a to hlavně hlavní varianty: PI a PD. Zjednodušené varianty se používají hlavně z důvodu snadnější implementace kontrolní smyčky v případě, že jedna složka regulátoru by se velmi blížila nulové hodnotě.

Výstup regulátoru se skládá ze sumy jednotlivých složek. Obecné schéma PID regulátoru je znázorněno na obrázku 2.3. Každá složka má vlastní koeficient, kterých slouží jako zesílení (případně i zeslabení jednotlivých členů). Samotné složky regulátoru jsou na sobě nezávislé, vstupem je pouze velikost chyby.



Obrázek 2.3: Blokové schéma obecného PID regulátoru

Pro proporcionální člen tedy platí vzorec [40]:

$$c(t) = K_p \cdot e(t) \quad (2.1)$$

kde:  $c(t)$  – výstupní kontrolní signál  
 $e(t)$  – velikost chyby  
 $K_p$  – koeficient proporcionální složky

Kontrolní signál je tedy v každý časový okamžik proporčně úměrný velikosti chyby. Jedná se o funkci aktuální hodnoty chyby. I samotná proporcionální složka může sloužit jako regulační člen za předpokladu, že reakce systému je velmi rychlá a bez setrvačnosti. U velké části systémů tomu tak není a výsledkem potom mohou být oscilace výstupní veličiny okolo požadované hodnoty. V případě, že výstupní hodnota je blízko požadované hodnotě bude kontrolní signál velmi malý. V závislosti na systému, může být tak malý, že systém na něj nebude ani reagovat a požadovaná výstupní hodnota nebude nikdy dosažena.

Z toho důvodu je přidávána integrální složka. Pro tu platí vzorec [40]:

$$c(t) = K_i \int_0^t e(t) dt \quad (2.2)$$

kde:  $c(t)$  – výstupní kontrolní signál  
 $e(t)$  – velikost chyby  
 $K_i$  – koeficient integrální složky

V příkladu výše, kdy není možné dosáhnout požadované hodnoty pouze proporcionální složkou, by spolu s integrální složkou docházelo k postupnému sčítání chyby tak dlouho, než by výstup integrálního členu nebyl dostatečně velký na to, aby ovlivnil celkový kontrolní výstup a tím i výstupní veličinu. Sám o sobě je integrální člen nepoužitelný, obecně funguje jako složka zpomalující reakci. V případě, že je integrační koeficient příliš velký, bude opět docházet k oscilaci, i když většinou menší než u proporcionální složky. Čím vyšší je integrační koeficient, tím pomalejší je doba odezvy a zároveň jsou více redukovány oscilace [39].

Derivační člen slouží pro ovlivnění kontrolního signálu podle velikosti aktuální změny. Platí pro něj tedy vzorec [40]:

$$c(t) = K_d \frac{de(t)}{dt} \quad (2.3)$$

kde:  $c(t)$  – výstupní kontrolní signál  
 $e(t)$  – velikost chyby  
 $K_d$  – koeficient derivační složky

Funguje tedy jako kompenzace předpokládaných budoucích hodnota výstupu. Derivační složka má největší vliv na začátku regulačního děje, tedy ve stavu, kdy je požadovaná hodnota nejdále od výstupní hodnoty a díky proporcionalní složce je tedy změna maximální. Derivační složka má také největší vliv na zmírnění oscilací výstupní veličiny. Účelem derivačního členu je zastavit změny systému, samotný člen D tedy nemůže dosáhnout požadované hodnoty.

### 2.2.1 Softwarová varianta PID regulátoru

V případě že je PID regulátor implementován programově, není možná okamžitá zpětná vazba jako u analogových systémů. Z toho důvodu dochází k výpočtu regulačního signálu ve smyčce s definovanou frekvencí. Frekvence PID regulátoru se odvíjí od rychlosti reakce systému. Mezi jednotlivými cykly PID regulátoru se přenáší chyba z předchozí smyčky pro výpočet derivačního členu a celková suma integračního členu. Všechny výpočty jsou stejné jako u analogové varianty regulátoru.

## 2.3 Měření proudu na plošném spoji

Schopnost změřit aktuální proud procházející obvodem umožňuje optimalizovat chování zařízení tak, aby byla snížena špičková spotřeba zařízení rozložením událostí v čase, čímž lze šetřit na nákladech na napájecí zdroje. Dále je možné sledováním protékajícího proudu zamezit přetížení, případně poškození obvodů. Eventuálně lze také sledovat spotřebu za účelem optimalizace návrh další verze zařízení.

Jelikož měření proudu vyžaduje rozpojení obvodu, nastává problém jak měřit proud na hotové desce plošných spojů. Ideálním způsobem je zakomponovat vybavení pro měření proudu již do návrhu obvodu. Jednou z možností může být i využití signálových hřebínků, které budou v běžném provozu přemostěny a pro případ měření proudu k nim bude připojen ampérmetr. Toto je vhodné jen pro malé proudy, protože jinak by přechodový odpor přemostění mohl být dostatečně velký k tomu, aby docházelo ke ztrátám nebo opalování kontaktů. Přechodovému odporu se lze vyhnout využitím pájecích propojek. Jedná se o plošky podobné těm, které se používají pro připájení SMD rezistorů, ale jsou umístěny blíže k sobě, aby je bylo možné spojit můstkem z pájky. Pro měření se rozpojí a je k nim připojeno měřicí zařízení.

Výše navržená měření ovšem slouží spíše k hledání chyb a k jednorázovému měření, než k trvalému sledování. Pro trvalé měření se využívá měření napěťové ztráty na snímacím rezistoru [31]. Podle Ohmova zákona lze následně z rozdílu napětí před a za rezistorem a jeho hodnoty určit protékající proud. Problém této metody je nepřesnost snímacího odporu. Tyto odpory se vyrábí většinou s přesností 1 % a jejich odpor stoupá s teplotou zvyšující se kvůli protékajícímu proudu. Maximální výkonový limit je navíc pro rezistory poměrně nízký, například pro pouzdro 2010 (5×2,5 mm) je limit 1 W. V případě rezistoru o odporu



100 m $\Omega$ , je maximální proud 10 A. Při tomto proudu dochází k ohřívání, proto je vhodné tyto rezistory používají jen do 25 % proudového limitu, kdy je efekt ohřívání minimální. Napěťová ztráta na snímacím rezistoru ovšem nemusí být ideální vzhledem k zařízení, které je napájeno. Vhodné je tedy používat rezistory s co nejmenším odporem, u kterých se ovšem snižuje výrobní přesnost nebo zvyšuje cena. Čím nižší je odpor tím nižší také bude efekt zahřívání.

Pokud je odpor příliš malý, na nejistotu měření může mít poté vliv i délka cest mezi snímacím rezistorem a například mikrokontrolérem, který měření provádí. A to zvláště pokud snímacím rezistorem protéká malý proud a ztráta napětí na rezistoru je tak velmi malá. Při malém napětí na snímacím rezistoru může také při převodu pomocí AD převodníku dojít vlivem kvantizace ke značné nejistotě. Z toho důvodu je vhodné co nejlíže ke snímacímu rezistoru připojit diferenciální (rozdílový) zesilovač [31]. Díky němuž lze zesílit napětí na rezistoru tak, aby bylo snáze měřitelné a nebylo tolik ovlivňováno odporem plošného spoje.

Jednou z dalších možností je také využít část cesty na plošném spoji jako snímací rezistor. Například cesta o délce 5 cm s šířkou 2,5 mm by měla mít odpor okolo 10 m $\Omega$  a takovou cestu lze bezpečně použít k vedení až 3 A bez výrazného nárůstu teploty. Diferenciální zesilovač by měl být umístěn tak, aby délka cest mezi vstupy zesilovače a konci snímané oblasti cesty byla co nejkratší. Problémem zde může ovšem být nepřesnost výroby u leptaných plošných spojů a případ, kdy by deska byla ohřívána okolním prostředím. V tom případě by musela být měřena ještě teplota mědi a na základě toho prováděny kompenzace.

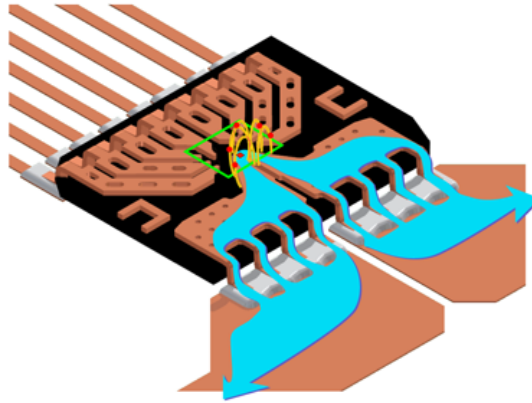
Kromě výše zmíněných možností existují i sondy pro osciloskopy a multimetry, které umožňují nedestruktivní měření proudu na hotové desce plošných spojů. Jedná se o sondy využívající výše zmíněného způsobu měření pomocí diferenciálního zesilovače. Tyto sondy ovšem vyžadují, aby na příslušném místě desky byla odstraněna pájecí maska, s tím lze případně počítat již v návrhu. Pro správné měření je ovšem nutné znát výrobní vlastnosti měřeného spoje, tedy šířka a tloušťka mědi. Dále existují i sondy založené na Hallově jevu, popsaném v sekci 2.3.1, které umožňují měření proudu i přes pájecí masku. Velice důležité je ovšem využití sondy pro správnou šířku plošného spoje a umístění sondy kolmo ke směru protékajícího proudu. Nevýhodou těchto sond je jejich velmi vysoká cena a minimální kompatibilita s ostatními zařízeními.

### 2.3.1 Měření pomocí Hallova jevu

Nejpokročilejší možností je použití proudových senzorů založených na Hallově jevu. Uvnitř pouzdra čipu je umístěna polovodičová destička, přes kterou protéká měřený proud. Za předpokladu, že okolo trasy kudy teče proud není žádné magnetické pole, protéká proud přímo. Když ale existuje magnetické pole kolmé na polovodič dochází k tomu, že trasa proudu není již rovná, ale zakřivuje se. Následně lze na stranách polovodiče měřit elektrický náboj odpovídající protékajícímu proudu. Toto napětí je velmi malé a proto senzory obsahují integrovaný zesilovač. Výhodou těchto integrovaných senzorů je izolace napájecího napětí od logického napětí použitého pro komunikaci se senzorem [36], čehož je dosaženo tím, že vodiče nejsou nikde fyzicky propojeny. Dalšími výhodami jsou nízké ztráty při průchodu proudu senzorem a vysoká teplotní stabilita. Často senzory obsahují i teplotní kompenzaci pro použití v průmyslovém prostředí při vysokých teplotách.

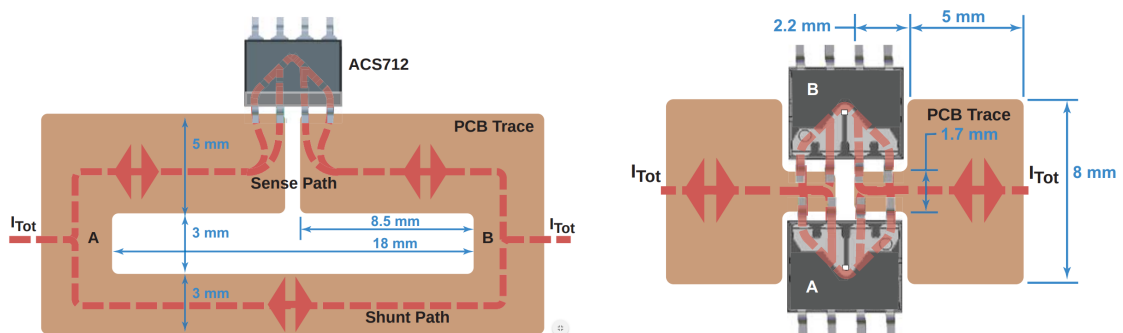
Tento typ senzoru je často umístěn uvnitř běžně používaných pouzder pro povrchovou montáž jako je například SO-8, SO-16 nebo DFN8. Napájecí cesta je přerušena a vedena přes snímací vstupy obvodu. Celkový odpor trasy uvnitř obvodu by při teplotě 20 °C měl být jen 200  $\mu\Omega$  [36]. Podle typu pouzdra jsou stanoveny maximální přípustné proudy, které

se pohybují v rozmezí od 20 do 100 A. Snímání proudu je možné v obou směrech. Výstup obvodu je analogový signál s klidovou hodnotou napětí 2,5 V při nulovém protékajícím proudu. Změna výstupního napětí je proporcionální vzhledem k protékajícímu proudu. Reakce výstupu pro senzor s maximálním limitem proudu 20 A je 100 mV/A [18]. Při dodržení návrhových pravidel u plošného spoje, která jsou popsána v dokumentaci [12] by chyba senzoru měla odpovídat maximálně 0,5 % z maximálního rozsahu.



Obrázek 2.4: Vnitřní struktura obvodu pro měření proudu, převzato z [50]

V případě, že je proud stále vyšší než maximální povolený limit obvodu, lze využít variantu, kdy dojde k rozdělení napájecí cesty tak, že přes obvod teče jen třetina celkového proudu [48]. Díky tomu je možné měřit větší proud, ovšem při nižším rozlišení. Návrh plošného spoje pro toto zapojení je ukázán na obrázku 2.5a. Tato varianta ale zabírá velkou plochu na plošném spoji a nejistota měření je závislá na výrobním procesu plošného spoje. V případě, že je potřeba měřit proud bez ztráty rozlišení, lze využít zapojení dvou proudových senzorů paralelně (zapojení na obrázku 2.5b). Je tedy možné měřit dvojnásobný proud s plným rozlišením. Důležité je dodržet symetrické zapojení, aby odpor obou cest byl naprosto stejný, jinak by mohlo dojít k poškození jednoho ze senzorů nadměrným proudem. Jelikož jsou senzory obousměrné, bude výstup jednoho odpovídat invertovanému výstupu druhého.



(a) Zapojení využívající vedlejší cestu jako bočník (b) Zapojení využívající součet ze dvou senzorů

Obrázek 2.5: Varianty zapojení senzoru pro měření velkých proudů, převzato z [48]

## 2.4 Řízení krokových motorů

Pasivní krokový motor se skládá z rotoru, který je vyroben z magneticky měkkého materiálu a statoru, tvořeného cívkami fungujícími jako elektromagnety. Průřez rotoru odpovídá tvarem ozubenému kolu, jednotlivé zuby jsou stahovány magnetickým polem cívek. Při průchodu proudem cívkou se vytváří magnetické pole, které přitáhne nejbližší zub rotoru, čímž dojde k jeho pootočení. Při aktivaci další cívky v pořadí dojde opět k posuvu rotoru tak, aby byly zuby rotoru zarovnané na aktivní cívky. Jedná se tedy o nespojitý pohyb označovaný jako krok motoru. Spojitého pohybu lze dosáhnout postupnou aktivací cívek. Velikost kroku a tedy i přesnost motoru je závislá na počtu odpovídajících dvojic cívek a zubů rotoru. Běžné krokové motory provádějí 200 až 400 kroků na jedno otočení rotoru a obsahují 2 až 4 cívky.

Pro ovládání krokového motoru je tedy potřeba řídicí člen, který umožňuje provádět buzení cívek uvnitř motoru. Tento prvek se nazývá budič krokového motoru. V nejjednodušší variantě se může jednat jen o tranzistory připojené k cívkám motoru, při jejich aktivaci prochází cívkou proud. Je ovšem nutné zařídit, že cívkou prochází maximální proud, pro který je motor dimenzován. Složitější integrované budiče obsahují většinou rozhraní pro komunikaci s kontrolním obvodem (převážně MCU), podporu mikrokrokování nebo případně i detekce ztráty kroku a podobně. Mikrokrokování je způsob řízení krokového motoru, který zvyšuje rozlišení jednotlivých kroků a to tím, že najednou aktivuje dvě sousední cívky. Za předpokladu že jimi prochází stejný proud, dojde jen k polovině kroku a tedy motor bude mít dvojnásobné rozlišení kroku. Při různých proudcích v cívkách je možné interpolovat libovolné pozice mezi celými kroky. Většina dostupných budičů podporuje interpolaci a to 8 až 256 mikrokroků, podle varianty budiče. Využitím mikrokroků se snižují vibrace generované motorem při nízkých rychlostech. Také se snižuje kroutící moment, jelikož zub rotoru se pohybuje v oblasti, kde není magnetického pole tak silné jako když je přímo zarovnan na cívku.

Rozbor budičů motorů a jejich parametrů je vztažen vůči platformě manipulátoru, která bude použita pro tento projekt. Platforma samotná je detailně popsána v kapitole 3.1. Jedním z aspektů porovnání je také skutečnost, zda dokáže řídicí část budiče dosáhnout maximální rychlosti krokového motoru a tuto rychlost udržet. Je totiž nutné udržovat vysokou frekvenci kroků, jelikož jsou také využity mikrokroky z důvodu vyšší přesnosti posunu a také nižší hlasitosti. Budič musí provádět až 1,152 milionů mikrokroků za sekundu, tedy při PWM řízení je délka jednoho pulsu  $0,868 \mu\text{s}$  (50 otáček za vteřinu, 180 kroků, 128 mikrokroků). Rovněž musí udržet přesné časování během doby, kdy dochází k akceleraci, v případě výchylky v časování může dojít ke ztrátě kroku, který může při vysoké akceleraci vést až k zastavení motoru.

### 2.4.1 Porovnání variant budičů

Budiče je možné rozdělit do tří skupin podle toho, jaké jsou možnosti jejich řízení. Většinou se od toho také odvíjí množství funkcí, které poskytují. Mezi ně může patřit detekce zaseknutí motoru, přehřátí budiče nebo úroveň mikrokroků.

Jedno z dalších rozdělení může být například integrace silové části budiče přímo do čipu. Tedy zda jsou MOSFET tranzistory řídicí proud motorem umístěny na řídicím čipu, nebo jestli je nutné je připojit externě. Při externích tranzistorech lze ještě rozlišit zda řídicí čip obsahuje budič tranzistoru nebo jestli i ten musí být zajištěn samostatně. Výhodou externích tranzistorů je hlavně možnost škálovat výkon a lepší rozvod tepla. Nevýhodou je

vyšší cena, jelikož se jedná o více součástek, které ještě navíc zabírají více plochy na desce plošných spojů.

### STEP/DIR rozhraní

První skupinou jsou budiče, které pro své nastavení využívají jen vnější prostředky. Tyto obvody mají většinou nastavitelné jen parametry jako je proud fází motoru a požadované mikrokrokování. Proud bývá nastavován jako referenční napětí pomocí potenciometru a mikrokrokování pomocí propojek, které nastavují logickou úroveň na vstupech čipu, z jejichž kombinace je určen počet mikrokroků. Samotné ovládání těchto budičů je poté vyřešeno přes takzvané STEP/DIR rozhraní. Jedná se o dva vstupy řídicího čipu, kdy náběžná hrana na vstupu STEP značí, že se má provést krok/mikrokrok a logická úroveň na vstupu DIR značí směr, kterým se tento krok provede.

Problémem těchto budičů je, že velká část výpočetního výkonu odpovědného za krokování se přenáší do mikrokontroléru, který se musí starat o časování a akceleraci. V situaci, kdy je požadováno provedení 1,152 milionu kroků za vteřinu znamená to, že na přípravu nového kroku má mikrokontrolér dobu jen asi 70 taktů. To za předpokladu, že je mikrokontrolér taktován na 80 MHz. V případě konstantní rychlosti se nejedná o problém, jelikož tyto pulzy mohou být generovány pomocí PWM kanálů časovače, ovšem během akcelerace a decelerace je nutné vypočítávat délku následných pulzů. To může způsobit nadměrné vytížení procesoru zejména ve chvíli, kdy je nutné řídit zrychlení/zpomalení více motorů najednou. Délky pulzů lze vypočítat podle následujícího vzorce 2.4 [49]:

$$t_i = t_{i-1} \cdot (\sqrt{n+1} - \sqrt{n}) \quad (2.4)$$

kde:  $t_{i-1}$  – délka předcházející intervalu  
 $n$  – index aktuálního intervalu

Případně podle optimálnější varianty, tedy vzorce 2.5 [49]:

$$t_i = t_{i-1} - (t_{i-1}^3 \cdot (1+a)) \quad (2.5)$$

kde:  $t_{i-1}$  – délka předcházejícího intervalu  
 $a$  – rychlost akcelerace motoru

Takto vysoké frekvence kroků nejsou u tohoto typu řídicích čipů příliš časté, jelikož mají většinou 32 mikrokroků. I tak je čas pro výpočet intervalu krátký (okolo 270 taktů). Počet taktů je odvozen pro nejhorší případ, tedy pro nejkratší pulzy akcelerace těsně před dosažením maximální rychlosti. Délky pulzů lze samozřejmě předpočítat, pokud mají všechny motory maximální rychlost a akceleraci stejnou. Při akceleraci 4 000 mm/s<sup>2</sup> bude potřeba minimálně 36 kB paměti, což může být u některých mikrokontrolérů vcelku podstatná část paměti, například mikrokontrolér STM32L452 osazený na vývojovém kitu Nucleo32 má 256kB paměti [14]. Tento problém s pamětí může být minimalizován, například použitím vhodné externí paměti. Vhodnou pamětí je NOR Flash, která by byla připojena přes sběrnici QuadSPI pro dostatečnou propustnost.

Mezi tento typ budičů patří například čip DRV8825, který se často využívá ve 3D tiskárnách a levnějších CNC strojích, dalším podobným čipem je A4988. Tyto čipy mají maximální povolený proud 2,5 A, což by mohlo být většinu času dostatečné. Jenže v případě,

že by dlouhodobě byly na hraně vytížení, může to u nich vést ke snížení životnosti, větší nepřesnosti kroků, ale také k vytváření více tepla, které by bylo nutné odvádět.

Poněkud pokročilejší variantou z této třídy budičů krokových motorů je TMC2100 od firmy Trinamic. Jedná se o řídicí obvod s 16-mikrokroky, umožňuje ovšem vnitřní interpolaci až na 256 mikrokroků (výsledná funkce se podobá funkci sinus). Tato interpolace je prováděna vždy mezi dvěma pulzy na vstupu DIR. Jelikož samotný řídicí obvod nezná dobu za kterou přijde následující pulz, odvozuje ji z času mezi předchozími pulzy. To má za následek, že v případě prudké změny rychlosti dochází k nepřesnostem a jsou tak zpomaleny i pulzy, které jsou přesně vygenerovány mikrokontrolérem, a to do té doby, než se budič adaptuje na novou frekvenci pulzů. Proto není příliš doporučeno využívat interpolaci pro systémy, kde dochází často ke změně směru případně k prudkým změnám rychlosti. Což by u tohoto zařízení nejspíš neměl být problém. Díky interpolaci kroků jsou motory řízené těmito budiči méně hlasité než při použití dříve uvedených budičů.

### Nastavení přes sběrnici

Tato třída budičů obsahuje stejné STEP/DIR rozhraní jako předchozí varianta, ovšem navíc umožňují nastavení více parametrů přes nějaký další komunikační kanál. V dále popisovaných případech se jedná o SPI nebo UART. Díky možnostem konfigurace tyto řídicí obvody nabízí více funkcí.

Těmito budiči jsou například obvody TMC2130 [17] (využívající SPI), případně TMC2208 (využívající UART). Tyto budiče mají maximální proud fázi 1,7 A, proto nejsou vhodné pro řízení větších motorů na plánovaném zařízení. Jejich silnější variantou je obvod TMC2660, který může být konfigurován přes SPI sběrnici. A umožňuje řídit motory proudem až 2,8 A.

Tento typ obvodů poskytuje velké množství dodatečných funkcí. Mezi ně patří například technologie CoolStep. Ta umožňuje regulovat proud fázi podle aktuální zátěže. Ve výsledku může vést až na 30% snížení hodnoty proudu, což má za následek snížení teploty integrovaného obvodu i motoru. Tato funkce je vhodná v případě, že nedochází ke změnám v zatížení motoru. Budič pracuje tak, že dochází k měření točivého momentu motoru a následný proud je tak upravován, aby měl motor rezervu stále 30–50 %. Díky tomu může být při vyšších rychlostech uspořena energie. Tato technologie má hlavní využití v zařízeních, kde je nutné šetřit energii, například pokud je zařízení napájené z baterie. Samotné měření vychází z funkce StallGuard.

StallGuard je další technologií, která je dostupná u velké části konfigurovatelných řídicích obvodů. Jedná se o detekci uvážnutí motoru na místě, tedy požadovaná sekvence kroků nebyla ještě dokončena, ovšem motor se nemůže dále otáčet. Obecně tato funkce měří rezervu točivého momentu. Navíc v případě uvážnutí dochází ke krátkému nárůstu proudu ve fázi motoru. Další variantou jak detekovat uvážnutí motoru, je využití enkodéru. Toto řešení vyžaduje využití dalších integrovaných prvků a další výpočetní výkon. Výhodou této funkce je to, že není nutné využívat koncové snímače k určení počáteční pozice. Na snímací platformě není nutné použití této funkce, jelikož už koncové snímače obsahuje, nicméně může sloužit pro případ, kdy by došlo k poruše, případně detekci zaseknutí, pokud by se do zařízení dostal nějaký cizí prvek.

### Vestavěné řízení

Tato třída budičů obsahuje kompletní systém pro řízení krokových motorů, jedná se o řešení ve stylu SoC (System-On-Chip). Samotný čip generuje krokové pulzy na základě nastavení v registrech. Velkou výhodou těchto řídicích obvodů je výrazně snížení zátěže mikrokontro-

léru. U těchto čipů je přes komunikační rozhraní přenesen příkaz o tom kolik kroků se má vykonat. Celý následný proces akcelerace, decelerace, kontrola zátěže je řízen samotným řídicím čipem motoru.

Jedním z těchto budičů je další integrovaný obvod od firmy Trinamic a to TMC5160. Ten umožňuje nastavení jak přes SPI, tak UART. Obsahuje uvedené technologie CoolStep, StallGuard a interpolaci mikrokroků. Jeho hlavní nevýhodou je, že neobsahuje integrované výkonové tranzistory. Díky tomu je možné libovolně dimenzovat výkon řídicího systému. Nicméně toto řešení má výrazný vliv na výslednou cenu zařízení, ale i na úroveň integrace celé základní desce.

Další variantou z této skupiny budičů je L6470 od firmy STMicroelectronics. Tento čip je detailněji popsán ihned v následující kapitole 2.5. Praktické experimenty s krokovými motory a jejich výběr je popsán v kapitole 3.3.

## 2.5 Budič krokových motorů - L6470

Budič krokových motorů L6470 slouží pro ovládání pohybu všech krokových motorů umístěných na platformě. Tento plně integrovaný systém se skládá ze tří základních částí. První je vestavěný kontrolér řídicí zpracování signálů a kontrolní logiku. Druhá je výkonová část, která zajišťuje vedení proudu do jednotlivých fází motoru. Třetí částí je rozhraní, přes které probíhá komunikace s entitou zajišťující ovládání.

Ovládán může být jeden dvou-fázový bipolární motor, přičemž je možné dosáhnout mikrokrokování s rozlišením až 128 mikrokroků. Pro přesné ovládání výstupního výkonu obsahuje výkonová část zpětnovazební obvody pro měření proudu, procházejícího fázemi. Dále obvod obsahuje pojistky, které mají zajistit ochranu a spolehlivé fungování. Mezi ně patří například ochrana proti zaseknutí motoru nebo přehřátí integrovaných tranzistorů.

Obvod je vyráběn ve dvou provedeních pouzdra, přičemž obě varianty jsou určeny pro povrchovou montáž (SMD) a to hlavně kvůli efektivnějšímu odvodu tepla skrz zemnicí plochy na desce plošných spojů. Jedná se o varianty pouzder HTSSOP28 a PowerSO36.

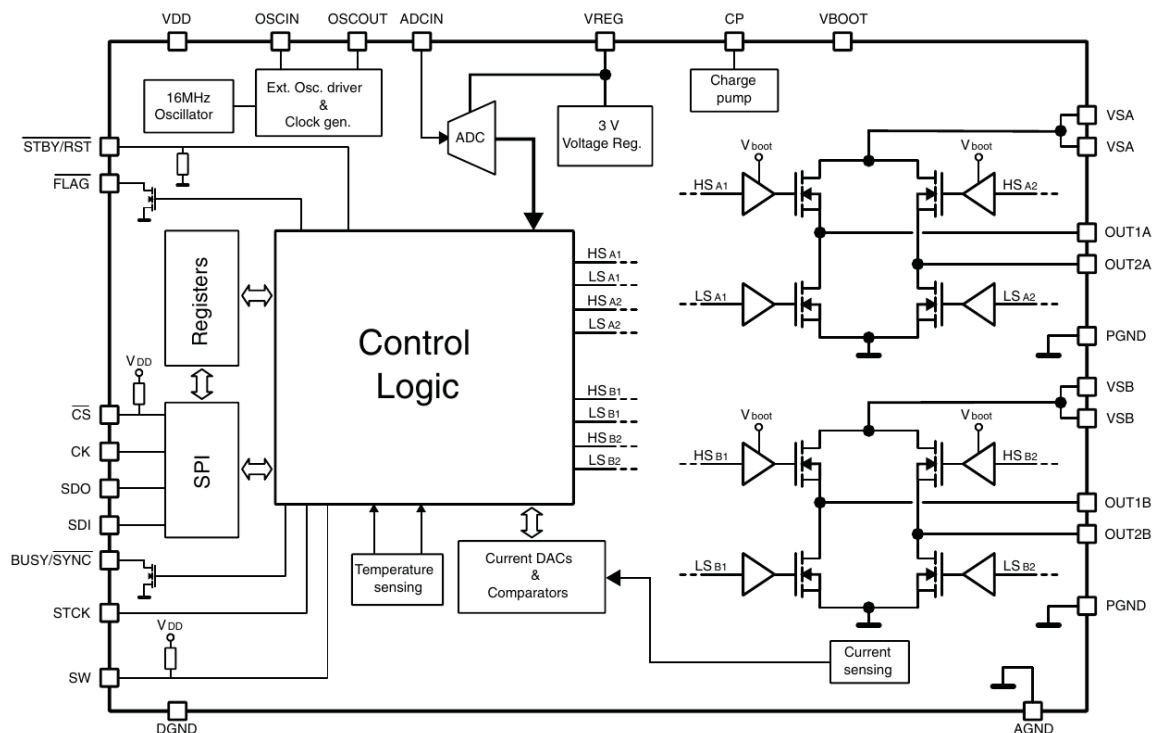
### 2.5.1 Výkonová část

Výkonové tranzistory, ze kterých se skládají můstky, jsou největším zdrojem tepla v celém obvodu. Z toho důvodu jsou použity tranzistory, jejichž odpor je v sepnutém stavu nízký. A to v rozmezí  $0,3\text{--}0,51\ \mu\Omega$ . Aby bylo možné dosáhnout tak nízkého odporu v sepnutém stavu, jsou použity MOSFET tranzistory s kanálem typu N. Pro otevření tranzistoru v horní části můstku musí být na elektroně Gate zajištěno větší napětí než je napájecí napětí obvodu. Z toho důvodu obsahuje čip i nábojovou pumpu, která spolu s několika externími součástkami zajišťuje buzení MOSFET tranzistorů.

Výkonovou část je možné odpojit přechodem do stavu vysoké impedance. Výsledný efekt je stejný jako by byl motor odpojen od řídicí čipu, nemá tedy žádný přídržný moment. Dále to má za následek výrazné snížení spotřeby elektrické energie a celkového tepla vyzařovaného obvodem.

Maximální hodnota spínaného efektivního proudu jsou 3 A. Po dobu kratší než 1 ms je ovšem možné spínat až 7 A špičkového proudu bez poškození řídicí elektroniky. To vše ovšem jen za podmínky, že je zajištěno spolehlivé chlazení obvodu. Teplota obvodu nesmí přesáhnout  $150\text{ }^{\circ}\text{C}$ .





Obrázek 2.6: Blokové schéma budiče krokových motorů L6470, převzato z [6]

## Napájení

Obvod obsahuje několik napájecích vstupů pro jednotlivé části. Jedná se o hlavní výkonové napájení označované jako  $V_{MOT}$ , které je rozdělené pro jednotlivé spínací obvody ( $V_{SA}$ ,  $V_{SB}$ ). Toto napětí slouží pro napájení fáze motoru, maximální povolené napětí je 45 V. Pro napájení logické části obvodu složí vestavěný regulátor, který stabilizuje napětí z hlavní napájecí větve pro motory ( $V_{MOT}$ ). Výstup tohoto regulátoru je vyveden na  $V_{REG}$  výstup. Výstupní napětí interního regulátoru má hodnotu 3 V. Toto napětí může být také použito k napájení rozhraní obvodu, skrze napájecí vstup  $V_{DD}$ . Podle napětí na vstupu  $V_{DD}$  je určen rozsah napětí pro komunikační kanály. V případě použití externího regulátoru k napájení rozhraní obvodu je možné pro komunikaci využít 5 V komunikační úrovně. Interní regulátor může také sloužit jako napájení dalším externím součástkám za předpokladu, že nebude překročen maximální odebíraný proud 40 mA.

## Reset

Po připojení logické části k napájení je nutné dodržet několik podmínek, aby byl reset obvodu korektně proveden a obvod tak přešel do pracovního módu, ve kterém může být využit. Těmito podmínkami je napájecí napětí motorů vyšší než 8,2 V v opačném případě přejde obvod do uzamčeného stavu, kdy není možné aktivovat žádnou výkonovou funkci. Tento stav je signalizován příznakem  $UVLO$ , příznaky jsou popsány v sekci 2.5.3. A napájecí napětí logické části (tedy vstup  $V_{REG}$ ) je minimálně 2,8 V.

## 2.5.2 Řídicí integrovaný systém

Integrované řízení umožňuje distribuovat zátěž z hlavního řídicího prvku na samotný budič. Díky větší kontrole tak může budič samotný implementovat více funkcí a poskytovat snadnější přístup k nim.

Jelikož čip obsahuje vlastní oscilátor, není závislý na časování a případném zpoždění externího obvodu, což má za následek zpřesnění mikrokrokování a rychlejší reakci na prudké změny zátěže motoru. Oscilátor může být použit jak interní tak externí umožňující využít vyšší a přesnější řídicí kmitočet. Interní oscilátor umožňuje nastavit frekvenci až 16 MHz, s externím lze dosáhnout frekvence 32 MHz.

### Kontrola pozice

Jelikož řízení kroků motoru je prováděno pomocí zabudovaného systému, nemůže externí mikrokontrolér tyto kroky počítat. I když mikrokontrolér může zadat příkaz pro provedení zadaného počtu kroků, existují příkazy, u kterých nemůže nadřízený prvek vědět, kolik bude provedeno kroků. Jedná se například o příkazy, kdy se motor pohybuje dokud nenarazí na překážku nebo koncový spínač. Aby bylo i v takovém případě možné určit absolutní polohu motoru, případně akčního členu ovládaného motorem, obsahuje obvod vestavěný čítač kroků, jehož hodnotu lze číst z registrového pole. Tento absolutní čítač se nuluje po startu obvodu a při každém najetí do domovské pozice. Ta může být definována koncovým snímačem (2.5.4) nebo předáním příkazu k vynulování. Pokud je nastaveno mikrokrokování, je při každém mikrokroku čítač inkrementován.

### Mikrokrokování

Za účelem zvýšení rozlišení kroků podporuje obvod mikrokrokování, které díky rozdílnému proudu fázemi umožňuje přesně nastavit polohu rotoru mezi dvěma celými kroky. Jeden celý krok tak lze rozdělit až na 128 mikrokroků, čímž se u běžného krokového motoru s 200 celými kroky změní rozlišení kroku z  $1,8^\circ$  na  $0,14^\circ$ . Zvyšující se počet mikrokroků má negativní vliv hodnotu točivého momentu. Z toho důvodu je úroveň mikrokrokování nastavitelná. Nastavení může probíhat jen ve stavu, kdy jsou můstky ve stavu vysoké impedance. Po změně nastavení je automaticky vynulován absolutní čítač pozice.

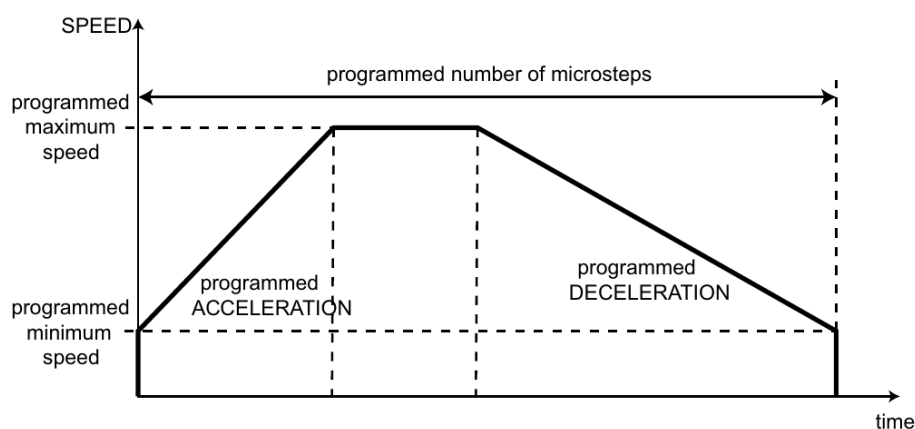
Při mikrokrokování dochází k interpolaci sinusoidy tak, aby hodnota odpovídala velikosti napětí na fázi a aktuální pozici motoru v rámci jedné periody (mezi plnými kroky). Sinusoida je generována dvěma PWM modulátory (pro každou fázi je určen jeden), které mají vzájemný fázový posun  $90^\circ$  [52]. Přesnost mikrokrokování je tedy závislá na frekvenci modulátoru. Tu lze ovlivňovat pomocí předděličky a násobičky frekvence oscilátoru.

Mikrokrokování má vliv nejen při nízké rychlosti, kdy je většinou požadována přesnost, ale také umožňuje dosáhnout vyšší akcelerace. Vyšších otáček tak lze dosáhnout rychleji, ovšem při vysokých otáčkách motoru je většinou vyžadován vyšší točivý moment, který převážně omezuje maximální rychlost otáčení. Jelikož mikrokrokování snižuje točivý moment, obsahuje řídicí systém funkci, kdy dochází k automatickému přepnutí mezi mikrokrokováním a plným krokem na základě prahové rychlosti. Pokud je rychlost nižší než stanový práh, je využito nastavené mikrokrokování a pokud je vyšší je využit plný krok.



## Řízení pohybu

Rychlostní profil pohybu se skládá ze tří částí, akcelerace, udržování konstantní rychlosti a následné decelerace (znázorněno na obrázku 2.7). Jelikož při nízkých rychlostech může vlivem velmi slabého elektromagnetického pole okolo fáze docházet k nepřesnému pohybu, neprobíhá akcelerace z nulové rychlosti. Tato minimální hodnota rychlosti (*programmed minimum speed*) je nastavitelná přes podporované příkazy. Je možné aktivovat optimalizaci pro nízké rychlosti, která zařizuje kompenzaci elektromagnetického pole tak, aby byl pohyb přesnější. Tato optimalizace je popsána v kapitole 2.5.5. Obdobně je možné nastavit hodnotu akcelerace (*programmed ACCELERATION*), maximální rychlosti otáčení (*programmed maximum speed*) a decelerace (*programmed DECELERATION*). Konstantní rychlosti nemusí být dosaženo, pokud se zadaný pohyb neskládá z dostatečného počtu kroků, v takovém případě navazuje fáze decelerace přímo na fázi akcelerace.



Obrázek 2.7: Průběh fází programovatelného rychlostního profilu, převzato z [6]

Příkaz k zastavení pohybu je generován vnitřním systémem řízení nebo externím zadáním příkazu (přes komunikační sběrnici). Vnitřně může být příkaz generován například při překročení maximální provozní teploty. Existují 4 varianty zastavení. Jedná se o postupné zastavení, kdy dochází k deceleraci na minimální rychlost, následně podle vybrané varianty zůstavení můstky aktivované (tzv. *SoftStop*) nebo přejdou do stavu vysoké impedance (tzv. *SoftHiz*). Další variantou je zastavení bez zpomalovací fáze, opět existují varianty kdy zůstávají můstky aktivní (tzv. *HardStop*) a kdy přecházejí do stavu vysoké impedance (tzv. *HardHiz*). Zastavení bez akcelerace jsou určena pro kritické situace, kdy nemůže být dovolen žádný další pohyb. Ovšem při zastavení typu *HardHiz* je možné, že dojde k následnému pohybu po zastavení krokování. Tato situace může nastat, jelikož motor nemá po tomto zastavení téměř žádný přídržný moment a ten může být překonán silou setrvačnosti akčního prvku.

### 2.5.3 Komunikační rozhraní

Komunikace probíhá přes sběrnice SPI, kde L6470 je v roli podřízeného zařízení (slave). V rámci komunikace probíhá přenos příkazů, které umožňují ovládání obvodu a také zápis do interních registrů čipu. Pomocí registrů se nastavují parametry, podle kterých bude prováděno řízení.

## Příkazy

Aplikační příkazy se skládají z jednoho úvodního bajtu, který udává, o který příkaz se jedná (provedení zadaného počtu kroků) a může v sobě nést i informaci, například jakým směrem má být proveden pohyb (definováno posledním bitem). Podle typu příkazu se odesílají další bajty, které blíže specifikují detaily příkazu. To může být počet kroků, které mají být vykonány nebo pozice na kterou se má motor přesunout.

Speciální příkaz slouží pro manipulaci s interními registry, popsány v následující sekci, v rámci úvodního bajtu je vybrán registr a specifikována akce (čtení/zápis). Dále jsou přenesena data. Speciální příkaz slouží pro získání hodnoty stavového registru, který je dále popsán v sekci 2.5.3-Příznaky.

Ostatní příkazy slouží k provádění pohybu a zastavení. Pohyb může být specifikován absolutně, nebo relativně. Absolutní pohyb je specifikován počtem kroků od hodnoty absolutního čítače. Relativní se odvíjejí od aktuální pozice a je u nich tedy specifikován počet kroků a směr. Existují také relativní příkazy, u nichž není předem definován počet kroků, jedná se o příkazy, kdy motor běží dokud není zastaven, případně slouží k uvolnění koncového spínače, po jeho uvolnění je motor ihned zastaven a je tak získána nová domovská pozice.

## Konfigurační registry

Registry slouží pro parametrizaci prováděných příkazů, nastavení kompenzací a detekcí nebo zjišťování stavu. Velikost registru není fixní, pohybuje se od 4 do 22 bitů. Při přenosu se ovšem vždy přenáší počet bitů dělitelný 8 a prvních  $n$  přebývajících je zahozeno. Kompletní tabulka registrů je v dokumentaci obvodu [6] na straně 40.

Registry obsahují aktuální hodnoty absolutního pozičního registru, rychlost a elektrickou pozici, což je poloha motoru mezi dvěma fázemi vyjádřena v počtu mikrokroků od posledního průchodu nulovou pozicí. Tyto registry jsou přístupné jen pro čtení.

Další registry slouží k nastavení rychlostní profilů (akcelerace, decelerace, minimální a maximální rychlost). K nastavení detekcí jako je například hraniční proud pro detekci zaseknutí nebo kritického proudu. Případně pro přizpůsobení kompenzací jako je násobitel pro eliminaci teplotního driftu, tyto kompenzace jsou popsány v sekci 2.5.4. Také se zde nastavuje napětí, které bude aplikováno na fáze během běhu, stání, akcelerace a decelerace, určení počtu mikrokroků a konfigurace interního oscilátoru a modulovaného PWM generátoru sinusového signálu.

## Příznaky

Jedním ze speciálních registrů je i stavový registr. Ten informuje o kritických událostech, jako je například ztráta kroku, zamknutí obvodu z důvodu nízkého napájecího napětí a detekce nadměrného proudu. Další příznaky informují o překročení doporučené teploty a o uzamčení z důvodu překročení kritické teploty. Další se vztahují k aplikačním příkazům a to například když není možné provést příkaz, například není možné zastavit motor, protože už stojí nebo takový příkaz není definován. Ostatní příznaky se týkají jen pracovního stavu motoru mezi ně patří směr pohybu, stav externího spínače, případně indikace že je nějaký příkaz aktuálně prováděn (BUSY). Příznaky informující o kritických událostech jsou při přečtení registru vynulovány.

Za účelem signalizaci stavu obsahuje obvod výstupy FLAG a BUSY. Ty jsou aktivní v nízké logické úrovni. Výstup FLAG informuje, že došlo ke kritické události, jedná se

o stejné události, které jsou obsaženy ve stavovém registru. Výstup BUSY reflektuje stav BUSY příznaku z registru příznaků.

#### 2.5.4 Funkce

Kromě základní funkčnosti, kterou je řízení otáček a provádění kroků obsahuje řídicí systém množství dalších funkcí sloužících k ochraně zařízení, zvýšení přesnosti nebo zjednodušení obsluhy.

##### Externí detekce

Funkce pro externí detekci jsou využity v případě, že motor slouží jako pohon akčního prvku, který vykonává pohyb po předem definované ose, jež je neměnná vůči počtu opakování. Aby bylo možné takový pohyb řídit, je potřeba referenční bod, od kterého se bude odvíjet adresování pozice na této ose pohybu. Příkladem může být lineární vedení a pozice vozíku na něm.

Pro určení koncové pozice se používají koncové snímače, které mohou mít podobu tlačítkového spínače, případně optobrány. Když je akční prvek přesunut do koncové pozice, aktivuje svým pohybem tento snímač, čímž definuje svoji počáteční pozici. Aby nemusel být koncový snímač kontrolován mikrokontrolérem, obsahuje L6470 vstupní pin, na který lze připojit tento snímač. Při jeho aktivaci nuluje čítač kroků. Předpokladem je, že při aktivaci snímače je na jeho výstupu sestupná hrana a neaktivní hodnota výstupu snímače je ve vysoké logické úrovni. V případě tlačítkového snímače musí být odfiltrovány jeho zákmity externí nízko-propustným filtrem.

Dále je možné detekovat uváznutí motoru. Tato situace nastává, pokud motor nemá dostatečný výkon k tomu, aby se dále otáčel. To může být například vlivem externí překážky, která mu brání v pohybu. Bez detekce uváznutí by dále docházelo k alternaci napětí na fázích, ovšem bez pohybu motoru, motor by tedy ztrácel kroky. Uváznutí je detekovatelné na základě rychlosti motoru a charakteristiky úhlu zatížení [32], [26]. Překročení hraniční podmínky je detekováno programovatelným proudovým komparátorem. Hodnotu prahu pro detekci je nutno určit experimentálně tak, aby běžný pracovní proud tuto hranici nepřesahoval a zároveň byl tento práh menší než proud při uváznutí.

Jak při aktivaci koncového snímače tak při detekci uváznutí jsou nastaveny příslušné příznaky ve stavovém registru a následně aktivován signál informující mikrokontrolér.

##### Ochrana

Aby bylo zajištěno spolehlivé fungování budícího obvodu a řízeného motoru, jsou v čipu obsaženy ochranné prvky, které mají zamezit kritickým situacím. Tyto ochrany mohou přímo zasahovat do řídicího systému a provádět v něm změny. Neslouží tedy jen k notifikaci mikrokontroléru o takovéto události, ale přímo jí zabraňují.

Jednou takovou ochranou je detekce nadměrného proudu. Jedná se hlavně o ochranu fázi motoru, ty jsou dimenzovány na určitý nominální proud a tomu je uzpůsobena jejich impedance. Pokud by byl proud výrazně vyšší nominální proud, hrozí poškození vodiče, ze kterého se fáze skládá. Ovšem i při nekritickém překročení proudu může docházet ke změně parametrů motoru. K eliminaci této situace slouží stejný proudový komparátor, který je využit pro detekci uváznutí. Do příslušného registru je zapsána hodnota, vůči které se bude porovnávat aktuální proud, při překročení dojde k nastavení příznaku a zastavení motoru. Postup pro zastavení motoru může být programově vybrán, musí být ovšem použita varianta

bez zpomalení (viz. 2.5.2). Pokud je zároveň používána detekce uvážnutí musí být hodnota detekce nadměrného proudu vyšší než hodnota pro detekci uvážnutí.

Další ochranou je tepelná pojistka. Ta má dvě úrovně, první je pouze varování, které nastavuje příznak ve stavovém registru 2.5.3 a další úroveň je nouzové vypnutí, kdy dojde k resetování celého obvodu až do doby, než teplota klesne pod kritickou úroveň. Teplota je měřena pomocí interního teplotního senzoru umístěného v pouzdře čipu.

Ochrana proti kritické úrovni napětí je aktivována, pokud napájecí napětí klesne pod hodnotu 7,2 V. Pokud je napětí nižší, není zaručena spolehlivá funkčnost budiče. Jakmile je napětí opět zvýšeno, přechází obvod do běžného režimu. Kromě zabránění nespolehlivé funkčnosti má tato ochrana chránit i napájecí zdroj v případě, že by zátěž řízená budičem měla za následek tak kritický pokles napájecího napětí. Za předpokladu, že napájecí zdroj nemá ochranu proti nadměrnému proudovému odběru, mohlo by dojít k jeho poškození.

### 2.5.5 Kompenzace chyb a optimalizace

Kompenzace mají za úkol vyvažovat nedostatky, které vznikají kvůli vedlejším jevům jako je například ohřívání motoru, případně změna magnetického pole, která ovšem nebyla zamýšlená a může tak mít vliv na přesnost.

#### Optimalizace při nízkých rychlostech

V případě, že je motor řízen při nízkých otáčkách nízkým napětím může nastávat problém se zkreslením proudu ve fázi. To nastává vlivem toho, že magnetické pole vytvořené cívkou motoru je natolik slabé, že magnetické pole sousední cívky překoná statický odpor a dojde k předčasnému zarovnání cívky na pól rotoru. Je tedy potlačeno několik mikrokroků v oblasti okolo nulového fázového posunu mezi póly. To má za následek, že nastavená (ideální) poloha se liší od skutečné pozice. V případě že se motor pohybuje rychle, není tento efekt vlivem silnějšího magnetického natolik znatelný. Aby byla zajištěna přesnost i v nízkých rychlostech, existuje možnost zapnout optimalizaci, která přepisuje funkčnost minimální rychlosti 2.5.2. Při aktivní optimalizaci nedojde k zastavení, pokud je rychlost nižší než minimální, je kompenzován proud fázemi a neodpovídá nadále křivce generované pro mikrokrokování, ale během průchodu jedné fáze nulou je navýšen proud na sousední fázi, aby bylo tak vykompenzováno slabé magnetické pole cívky. Minimální rychlost motoru není v tomto módu omezena.

#### Kompenzace poklesů napájecího napětí

Vlivem náhlého proudového odběru může docházet ke kolísání napětí zdroje, zejména krokový motor může být zdrojem takovýchto nárazových odběrů. Jelikož sinusová křivka generovaná PWM modulátor je přímo proporcionální vůči napájecímu napětí motorů ( $V_{MOT}$ ), tak při jeho poklesu dochází k tomu, že fáze motoru je řízena nesprávným napětím, což má za následek chybu v mikrokrokování a motor tak není ve stejné pozici, jako by byl při nominálním napětí. Z toho důvodu obsahuje obvod integrovaný AD převodník, ke kterému je přes odporový dělič připojeno napájecí napětí. Výstupní napětí děliče by mělo odpovídat polovině napětí pro napájení logiky ( $V_{REG}$ ). Na základě nastavení oscilátoru tak dochází ke vzorkování napájecího napětí motorů. Jelikož u napájení logiky se nepředpokládají takové výkyvy jako na napájení výkonové části, slouží tak  $V_{REG}$  jako referenční napětí AD převodníku. Podle výstupu z AD převodníku je tedy možné určit poměr napájecího napětí

motoru vůči napájecímu napětí logiky a na základě toho detekovat poklesy napětí. Nižší napětí zdroje lze následně kompenzovat zvýšením proudu ve fázi motoru.

### **Kompenzace teplotního driftu ve vinutí**

Při používání motoru dochází k jeho zahřívání. Zvedá se tedy i teplota vinutí ve fázích. Spolu se zvedající se teplotou vinutí se zvyšuje odpor a při konstantním napětí se snižuje proud. Což má za následek slabší elektromagnetické pole okolo fáze, následkem toho může být až ztráta kroků motoru. Obvod tedy obsahuje registr, který slouží pro násobení střídy signálu na výstupu PWM generátoru. To má za následek zvýšení napětí na fáze a tedy i zvýšení proudu. Obvod ovšem neposkytuje rozhraní pro měření teploty ani algoritmus pro určení hodnoty v kompenzačním registru. Obojí musí být implementováno řídicím obvodem.

## **2.6 Řízení LED osvětlení**

Jas LED diody je přímo úměrný proudu, který přes ni prochází. Využití proudového zdroje je tedy nejvhodnější způsob řízení pro LED diody, u kterých je nutné nastavovat přesnou úroveň jasu. Proudový zdroj funguje tak, že zvyšuje napětí postupně až do té doby, dokud připojeným obvodem neprochází nastavený proud nebo není dosaženo maximální povolené napětí. Další výhodou při využití proudového zdroje je zabránění kolísání jasu kvůli zvyšující se teplotě diody. V případě, že by byla LED dioda napájena běžným zdrojem s tím, že by na začátku bylo nastaveno napětí takové, aby byl dosažen požadovaný jas, v průběhu svícení by vlivem ohřívání diody docházelo ke zvyšování vnitřního odporu a tedy i změně proudu a tím i jasu. Při napájení proudovým zdrojem při snížení procházejícího proudu dojde okamžitě ke zvýšení napájecího napětí na takovou hodnotu, aby proud procházející diodou byl konstantní.

Proudový zdroj lze realizovat pomocí snímacího a regulačního prvku. Snímací prvek slouží jako zpětná vazba a bývá realizován jako odpor s nízkou hodnotou, na kterém se měří úbytek napětí. Jako regulační prvek je možné použít tranzistor, který by následně byl ovládán řídicím napětím. V případě bipolárního PNP tranzistoru je jeho kolektor připojen na napájecí napětí, k emitoru je připojena řízená dioda, za diodou následuje snímací odpor. Podle známé hodnoty odporu a úbytku napětí lze vypočítat proud, který prochází přes diodu. Podle požadovaného proudu je poté nastaveno napětí na bázi tranzistoru tak, aby byl otevřen natolik, že procházející proud odpovídal požadovanému. Tranzistor v tomto případě slouží zjednodušeně jako napětím řízený odpor.

Řízení napětí na bázi může být řešeno analogově i digitálně. Jako digitální kontrolní prvek lze například využít PID regulátor z kapitoly 2.2. Jelikož tranzistor reaguje velice rychle na změnu řídicího napětí, stačila by jen proporcionální člen. Pro analogové řízení lze využít snadnější realizaci a to použít operační zesilovač zapojený jako komparátor napětí. Určí se požadované napětí na snímacím rezistoru a napětí o takové hodnotě je přivedeno na neinvertující vstup komparátoru, skutečné napětí na snímacím rezistoru je přivedeno na invertující vstup. V případě, že požadované napětí je vyšší než skutečné, je na výstupu kladné saturační napětí komparátoru a řídicí tranzistor je tak plně otevřen a procházející proud je maximální. Jakmile stoupne úbytek napětí na rezistoru nad požadovanou hodnotu, dojde k překlopení komparátoru a tranzistor je uzavřen. Změna těchto úrovní je okamžitá a tak dochází k plynulé regulaci tranzistoru. Napětí se tedy rozdělí mezi snímací odpor, LED diodu a tranzistor.

Nevýhodou tohoto řešení je právě úbytek napětí na tranzistoru. Pokud je napájecí napětí výrazně vyšší než napětí potřebné k dosažení požadovaného proudu, bude většina tohoto napětí právě na regulačním tranzistoru. Což bude mít za následek značnou výkonostní ztrátu, která bude vyzářena ve formě tepla. Je nutné zařídit odvod tepla, tedy chladič nebo dostatečně rozměrnou zemnicí plochu na plošném spoji fungující jako chladič. Díky ztrátám v podobě tepla také výrazně klesá efektivita tohoto řešení. Dále jsou tu opět problémy se snímacím rezistorem, jako byly popsány v kapitole 2.3.

### 2.6.1 Impulsní zdroje

Efektivnější variantou je využití impulsního zdroje napětí (spínaný zdroj). Pro nastavení výstupního napětí využívají impulsní zdroje zpětnou vazbu, která funguje jako napěťová reference (většinou realizovaná jako odporový dělič), podle ní je poté řízeno spínání regulačního tranzistoru. Impulsní měnič tak lze upravit na proudový zdroj tak, že je napěťová reference je současně snímacím odporem. Tedy v případě že impulsní měnič využívá napěťovou referenci o hodnotě 0,2 V a požadovaný proud je 100 mA odpovídá snímací odpor hodnotě  $2\ \Omega$ . Efektivita takového měniče se podle řídicího čipu a regulačního tranzistoru pohybuje většinou okolo 80–95 %.

Pro srovnání, proudový zdroj s použitím komparátoru ve zpětné vazbě bude mít účinnost podobnou nebo nižší. Například mějme zdroj napětí o hodnotě 24 V, 4 kusy diod, v jejichž volt-ampérové charakteristice odpovídá napětí 3,3 V proudu 100 mA. Za předpokladu odporu od hodnotě  $2\ \Omega$  na něm bude napěťová ztráta 0,2 V. Na tranzistoru tedy bude nutné mít napěťovou ztrátu 10,6 V. Efektivita tohoto zapojení bude zhruba 55,8 %, s tím, že 1,06 W bude na tranzistoru přeměněno na teplo. V případě použití 6 LED diod se efektivita zvýší na 83,3 %, což je okolo spodní hranice efektivit spínaného zdroje.

Nevýhodou spínaného zdroje je ovšem potenciální rušení, které vychází z toho, že regulační tranzistor je spínán na vysoké frekvenci, ta je většinou v řádech MHz [41]. Také má pomalejší reakci na změnu zatěžovacího proudu, oproti tomu proudový zdroj s komparátorem reaguje okamžitě. Impulsní zdroj má také výrazně vyšší zvlnění výstupního napětí, frekvence tohoto zvlnění ovšem odpovídá spínací frekvenci regulačního obvodu, je tedy velmi vysoká a těžko měřitelná.

## Kapitola 3

# Návrh a realizace hardwaru

Tato kapitola se zabývá výběrem vhodných součástek, které budou využity v návrhu základní desky. Součástí výběru je i porovnání vlastností jednotlivých dostupných součástek. Na základě tohoto výběru jsou popsány vlastnosti a parametry součástek. Dále je popsáno obecné schéma bloků, ze kterých se bude skládat základní deska zařízení a následně rozbor jednotlivých bloků. Součástí této kapitoly je také popis vlastností manipulátoru a jeho propojení se základní deskou.

### 3.1 Konstrukce manipulátoru

Motorizovaný manipulátor od firmy TESCOAN má 5 stupňů volnosti. Může se tedy lineárně pohybovat po osách X, Y a Z a následně může provádět rotaci okolo os Y a Z. K pohonu pohybu v rámci os je využito 5 krokových motorů. Na každé ose je také umístěn snímač, který informuje o dosažení konce osy a je oproti nim určována absolutní pozice každé z os. Na manipulátoru je osazena fundus kamera, sloužící k pořízení snímku sítnice oka. Průřez zařízením můžete vidět na obrázku 3.1b, autorem návrhu je Ing. Jaroslav Pokorný z firmy creinos. Samotný manipulátor je poté na fotce 3.1a.

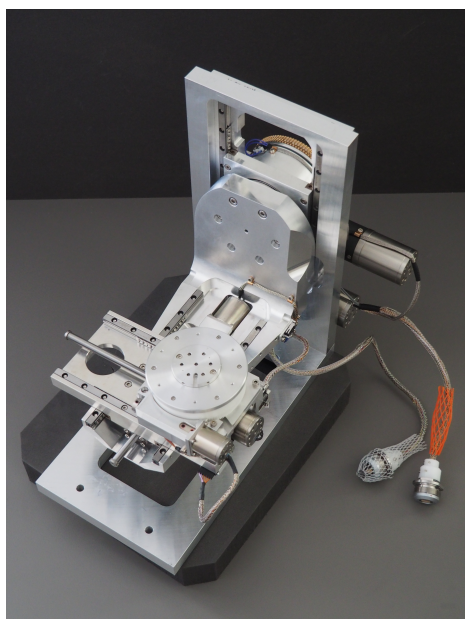
Motory manipulátoru [15] jsou zpřevodovány za účelem dosažení maximální přesnosti. Například s využitím mikrokrokového řízení při použití 128 mikrokroků a motoru, který má krok o velikosti  $1,8^\circ$  lze na ose Y dosáhnout teoretické přesnosti až 20 nm. Kvůli tak velkému zpřevodování bude také nutné zařídit dostatečnou rychlost motorů, aby přesun na vhodnou pozici pro snímání netrval příliš dlouho.

Zařízení je rozděleno na dvě sekce. Horní sekce obsahuje manipulátor a fundus kameru. Dolní sekce obsahuje základní desku, zdroje sloužící pro napájení celého zařízení a jednodeskový počítač, který bude zprostředkovávat rozhraní pro komunikaci. Obě části zařízení budou odděleny od sebe přepážkou a odvětrávání tepla z každé části bude zařízeno ventilátory umístěnými na šasi zařízení. Dále může být jednodeskový počítač připojen k displeji upevněnému na zařízení a realizovat komunikační rozhraní s uživatelem. V případě nedostatečného chlazení základní desky mohou být ještě osazeny dva ventilátory o velikosti 80 mm přímo nad ni a zajišťovat tak kvalitnější chlazení výkonové části.

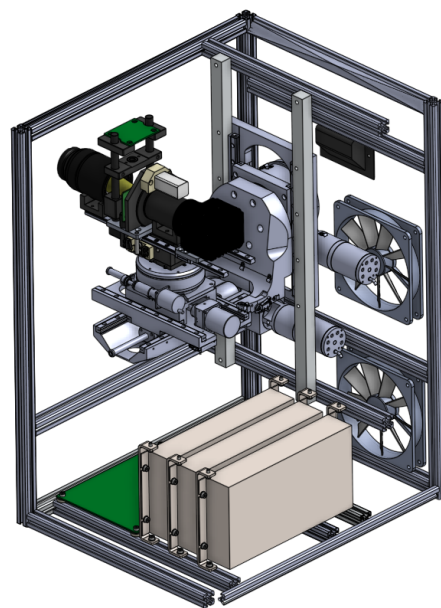
#### 3.1.1 Fundus kamera

Blokové schéma fundus kamery umístěné na manipulátoru je obrázku 3.2. Základním prvkem je barevná kamera s vysokým rozlišením, konkrétně v této realizaci zařízení je použita kamera od firmy Allied Vision a to PROSILICA GT-5120. Kamera získává snímky





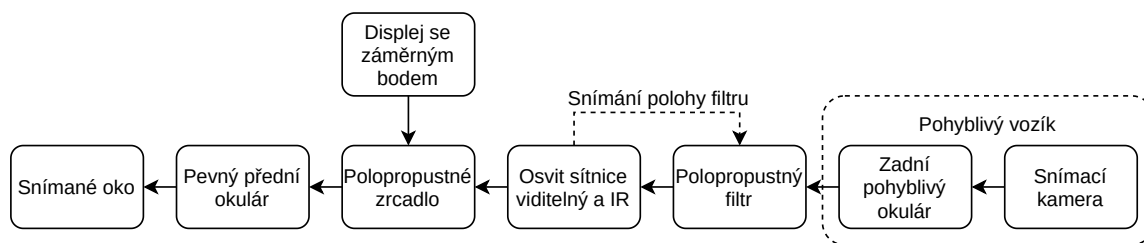
(a) Detail manipulátoru



(b) Průřez celým zařízením

Obrázek 3.1: Manipulátor použitý v zařízení pro snímání sítnice

v rozlišení  $5120 \times 5120$  pixelů, neobsahuje žádné filtry, jelikož budou pořizovány i snímky s použitím infračerveného osvětlení. Kamera je připojena k jednodeskovému počítači přes gigabitový Ethernet a napájena pomocí PoE injektoru. Kamera je umístěna na pohyblivém vozíku spolu se zadním okulárem, který slouží jako objektiv pro kameru. Pohon tohoto vozíku je zajišťován krokovým motorem. Vozík se pohybuje po dvojici lineárních vedení, ke kterým je připevněn pomocí 4 pojezdů. Lineární vedení mají zajišťovat maximální plynulost pohybu. Rozsah pohybu vozíku je omezen koncovými spínači, délka dráhy je 37 mm. Posun vozíku s kamerou a zadním okulárem vůči statickému přednímu okuláru zajišťuje ostření kamery. Model fundus kamery je na obrázku 3.3.



Obrázek 3.2: Blokové schéma fundus kamery EYRINA v4

Aby se zabránilo odleskům světla od sítnice, je mezi osvětlovací jednotku a kameru vložen polarizační filtr. Úhel polarizačního filtru je nastavitelný v plném rozsahu ovládnání je opět realizováno pomocí krokového motoru. Aby bylo možné určit úhel filtru je na desce obsahující osvětlení umístěn i reflexivní snímač. Tento snímač obsahuje diodu vysílající modulované světlo a detektor. V případě, že dojde k odražení světla vysílače do detektoru, dojde k sepnutí interního tranzistoru a změně logické úrovně na výstupu snímače, která může být dále detekována. Na polarizačním filtru je umístěno malé zrcátko, které slouží

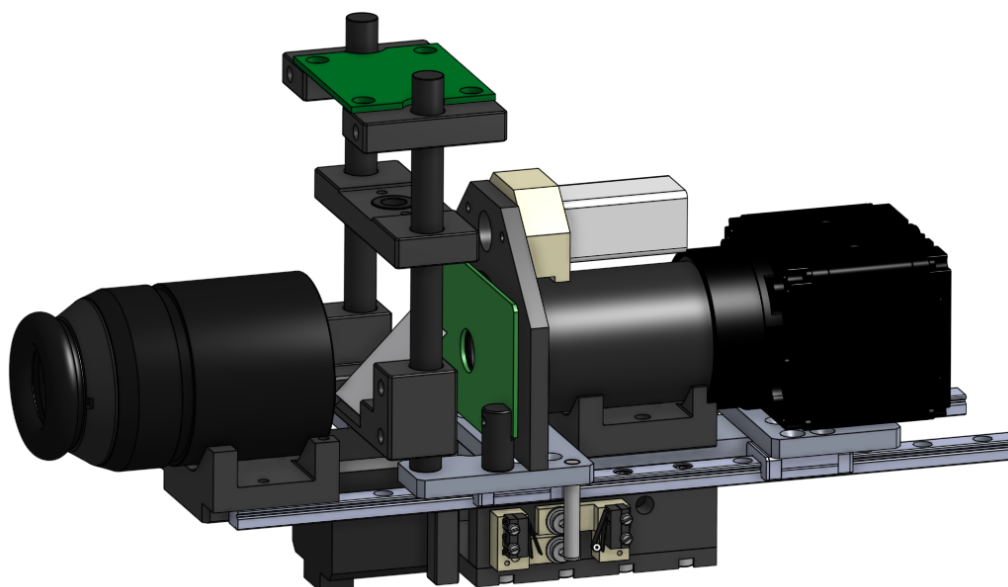


jako odražeč pro reflexivní spínač. Základní poloha polarizačního filtru odpovídá pozici kdy je zrcátko zarovnáno se spínačem.

Pro osvětlení sítnice oka slouží rozšiřující deska, která se připojuje k hlavní desce obsahující logiku a výkonovou část. Deska obsahuje dvě sady po 4 LED, které jsou umístěny v kruhu okolo otvoru pro průchod světla je kamery. Detailnější popis desky je v kapitole 3.6.

Před deskou zajišťující osvětlení sítnice je umístěno polopropustné zrcadlo, které slouží k promítání záměrného bodu. Polopropustné zrcadlo je osazeno tak, že s optickou osou zařízení svírá úhel  $45^\circ$ . Samotný záměrný bod je reprezentován jako shluk pixelů na OLED displeji, princip záměrného bodu podrobněji rozebrán v kapitole 3.5.5. Displej je připevněn ve vzdálenosti 72 mm od středu polopropustného zrcadla a osa vyzařování pixelů je kolmá na optickou osu kamery.

Posledním prvkem fundus kamery je přední pevný okulár, který směřuje světlo skrz zornici na plochu sítnice. Přední část okuláru obsahuje upevnění pro gumový kroužek ke kterému se přikládá snímané oko.



Obrázek 3.3: Model fundus kamery EYRINA v4, Autor: Ing. Jaroslav Pokorný

## 3.2 Požadavky na přesnost

Požadavky jsou založeny na konzultacích s oftalmology a na praktických pokusech manuálně ovládanou fundus kamerou. Na základě těchto informací jsou vytvořeny specifikace pro přesnost v jednotlivých osách zařízení.

Pro osy X, Z je minimální požadovaná přesnost 0,05 mm. Tato přesnost je založena na velikosti zornice oka, jejíž průměrná velikost je okolo 3 mm. Použitelný snímek vzniká, pokud kamera míří do středu zornice. Odchylka od středu sítnice může být maximálně 0,1 mm. Nicméně pro kvalitní snímek je potřeba přesnost vyšší, proto byla vybrána přesnost 0,05 mm. Maximální odchylka od středu zornice by tedy měla být 0,025 mm. Pro osu Y není přesnost klíčová, jelikož lze kompenzovat pohybem okuláru, nicméně pokud by byla tato

osa použita i pro vytvoření přehledového snímku, podle kterého by se poté provádělo přesné zaměření, tak přesnost osy by měla být alespoň 0,1 mm.

Aby byl obraz z kamery ostrý, je nutné promítnout každý snímáný bod na plochu, která je menší než velikost pixelu kamery. Z výkresu optické soustavy a parametrů kamery vychází velikost kroku na 0,025 mm. Na základě návrhu optika je ovšem ideálně velikost kroku o něco menší a to 0,02 mm.

Pro osy, okolo kterých se provádí rotace, nebylo možné získat data, jelikož takové zařízení pro snímání neexistuje. Ovšem na základě konzultací a výpočtů provedených pro osy X a Z, vychází přesnost natočení  $0,1^\circ$  jako dostatečná pro experimenty.

### 3.3 Porovnání budičů krokových motorů

Aby bylo možné porovnat budiče krokových motorů a jejich schopnosti, byl připraven test, který měl hlavně vyzkoušet, jaké maximální rychlosti a při jaké akceleraci jsou schopny dosáhnout. Pro test byly využity řídicí obvody popisované v kapitole 2.4.1. Konkrétně se jednalo o DRV8825, A4988, TMC2130, TMC2208, TMC2660 a L6470. Podmínky při testu se pokud možno co nejvíce blížily podmínkám na manipulátoru/platformě (hlavně osy X a Y). V době porovnávání budičů ovšem tato zařízení nebylo dostupné. Pro testy byla využita konstrukce 3D tiskárny a jednalo se o pohyb po předem definovaných drahách. Zátěží bylo závaží o hmotnosti 1,2 kg, které se pohybovalo po lineárním vedení, odpor při pohybu by tedy měl odpovídat tomu, jaký odpor bude vznikat při pohybu manipulátoru. Zátěž zhruba odpovídá hmotnosti snímací soustavy, která je umístěna na nástavbě manipulátoru. Při testech maximální rychlosti a zrychlení byl motor zpřevodován v poměru 10:1. To z toho důvodu, aby bylo možné tyto parametry vůbec ověřit, jelikož celková délka lineárního vedení byla jen 50 cm. Motory použité pro test byly typu NEMA17 jejichž parametry jsou: 200 kroků na jedno otáčku a maximální proud 1,4 A na cívku. Jejich princip a i parametry jsou podobné jako u motorů, které jsou osazeny na platformě manipulátoru a jedná se o identický motor, jako ten který je použit pro ostření kamery.

Při testech, bylo zkoušeno, jaké maximální rychlosti dokáže budič s motorem dosáhnout, jakou maximální akceleraci je schopen vyvinout, a také jaký hluk motor při plné rychlosti vydává. V rámci měření hluku byl problém přesně stanovit jeho hodnotu jednak kvůli okolnímu hluku, a také nepříliš kvalitnímu nahrávacímu zařízení. U toho důvodu jsou výsledky u hluku uváděny jen relativně na stupnici 1 až 5, kde 5 je největší úroveň hluku.

Pro testování akcelerace byla nastavena základní hranice zrychlení  $1\ 000\ \text{kroků/s}^2$ . Hranicí maximální rychlosti bylo 11,11 otáček za vteřinu (což je úhlová rychlost rotoru  $4\ 000\ ^\circ/\text{s}$ ). Aby motor splnil tento test musel být schopen  $5\times$  po sobě dosáhnout maximální rychlosti a tu následně udržet po 3 vteřiny. Akcelerace se postupně zvyšovala s přírůstkem  $500\ \text{kroků/s}^2$ . Případně snížením přírůstku o 200, pokud motor nebyl schopen akcelarovat rychleji. Test bylo nutné dokončit úspěšně  $5\times$ , jelikož se často stávalo, že v několika případech se podařilo akcelarovat úspěšně, ale někdy se motor ani nepohnul, jelikož budič nebo i mikrokontrolér nebyl schopen udržet přesné časování kroků.

Testování maximální rychlosti probíhalo na základě zjištěné akcelerace. Pro tento test byla použita akcelerace o hodnotě 70 % maximální akcelerace, aby byla zajištěna spolehlivost. Cílem tedy bylo vyvinout maximální rychlost a udržet ji po pokud možno co nejdelší dobu. Délka tohoto testu byla výrazně ovlivněna délkou lineárního vedení, nebylo tedy možné ověřit, zda je možné udržet maximální rychlost po delší dobu, pro splnění testu se jednalo o zhruba 3 vteřiny.

V dalším testu se ověřovala spolehlivost, hluk a teplota. Jednalo se o opakované zrychlení, udržení konstantní rychlosti a zpomalení. Pohyb byl prováděn po celé délce lineárního vedení a během testu byl opakován celkem 100× tam a zpět. Na závěr byla změřena teplota motoru a budiče infračerveným teploměrem. Infračervený teploměr není příliš vhodný na měření tak malých objektů jako jsou integrované obvody, ovšem výsledné teploty dávají alespoň přehled o namáhání a potřebě chlazení. Test probíhal při 70 % maximální akcelerace a 70 % maximální rychlosti. Test probíhal 3× po sobě a byla mezi nimi doba okolo 2 minut. Toto zatížení překračuje předpokládané zatížení, kterému bude platforma vystavena.

Při testech bylo využito nejvyšší možné mikrokrokování. Budiče od firmy Trinamic jsou testovány ve dvou nastaveních. První nastavení využívalo 32 mikrokroků bez interpolace, druhé nastavení obsahovalo interpolaci a zapnutí funkce StealtChop, tzv. tichý mód, která by měla mít za následek snížení produkovaného hluku. Proud byl nastaven tak, aby odpovídal nominálnímu proudu krokového motoru, což bylo 1,4 A.

### 3.3.1 Výsledky

Název čipu	Akcelerace [ $^{\circ}/s^2$ ]	Rychlost [ $^{\circ}/s$ ]	Teplota budiče [ $^{\circ}C$ ]	Teplota motoru [ $^{\circ}C$ ]	Relativní hluk
A4988	3 000	6 000	85	37	5
DRV8825	4 000	6 500	72	38	5
TMC2130	5 700	7 800	61	33	3
TMC2130 tichý mód	2 500	6 900	60	29	1
TMC2208	5 000	7 800	64	32	3
TMC2208 tichý mód	2 000	6 700	62	27	1
TMC2660	6 000	8 000	53	35	3
L6470	7 000	8 000	67	42	2

Tabulka 3.1: Porovnání jednotlivých budičů

Z výsledků lze vidět, že budiče od firmy Trinamic výrazně vynikají tichostí, avšak zvláště při testu spolehlivosti měly problémy s akcelerací. Akcelerace v tichém módu byla někdy až poloviční oproti normálnímu módu a také byla výrazně snížena maximální rychlost a to až o 25%.

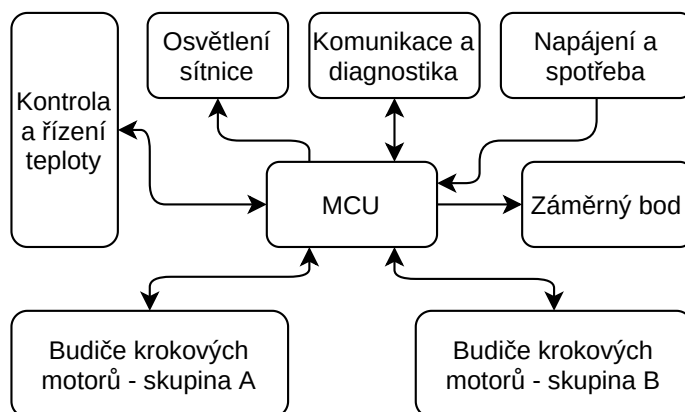
Budiče využívající jen STEP/DIR rozhraní měly problém hlavně při deceleraci, přestože impulzy od mikrokontroléru přicházely přesně načasovány, pulzy na výstupech budičů byly opožděné a nepřesně časované. To mělo za následek výrazný hluk a v několika případech i ztrátu kroků a následné velice hlučné zastavení motoru. Nicméně i během konstantní rychlosti byly tyto budiče velice hlučné.

Cena všech budičů se řádově neliší, nejlevnější DRV8825 se prodává za cenu okolo 100 Kč a nejdražší, L6470 za cenu okolo 190 Kč. Jelikož v zařízení se jich nachází 8 nejedná se o příliš velký cenový rozdíl a lze je tedy přímo porovnávat podle funkčnosti, kterou poskytují. Z toho důvodu byl i pro realizaci řídicí desky použit L6470.

### 3.4 Napájení zařízení

Pro napájení zařízení slouží 3 samostatné zdroje napájení. Všechny tyto zdroje jsou stejné a poskytují na výstupu napětí 36 V a mají maximální výkon až 480 W. Jedná se o zdroje Mean Well HLG-480H-36 [8]. Bylo by možné použít i jeden samostatný zdroj, ten by musel mít výkon minimálně 1,2 kW. Jelikož cena takového zdroje je téměř dvojnásobná jako cena tří jednotlivých, bylo jednodušší napájení rozdělit. Dva zdroje slouží pro napájení krokových motorů, které jsou rozděleny do dvou skupin po čtyřech, třetí slouží pro napájení logiky a ostatních zařízení na desce, jako je třeba osvětlení. Zdroje jsou uloženy ve spodní kryté části zařízení a jsou chlazeny pomocí dvou ventilátorů, kterou jsou řízeny z hlavní desky. Konektory použité pro napájení desky jsou rychlo-rozpojovací svorkovnice s maximální nominálním proudem 11 A na kontakt. Jelikož zdroj je schopen poskytnout až 13,3 A trvale, je použit konektor se 4 kontakty a vždy dva jsou využity jsou pro kladný pól a dva pro záporný.

### 3.5 Základní deska



Obrázek 3.4: Blokové schéma základní desky

Řídícím prvkem základní desky je mikrokontrolér od firmy STMicroelectronics. Konkrétně se jedná o MCU s označením STM32L452CEU6 [16]. Jedná se o mikrokontrolér založený na jádře ARM Cortex-M4 s FPU jednotkou. Mikrokontrolér běží na frekvenci 80 MHz, obsahuje flash paměť o velikosti 512 kB a 160 kB SRAM. Mikrokontrolér je v pouzdře UFQFPN48 o rozměrech 7×7 mm a je napájen napětím o hodnotě 3,3 V.

K hlavní řídicí desce se dále připojuje sekundární deska obsahující osvětlovací LED prvky. Tato deska je spolu i s řídicí elektronikou, která je ale umístěna na základní desce popsána v samostatné sekci 3.6.

#### 3.5.1 Komunikace a diagnostika

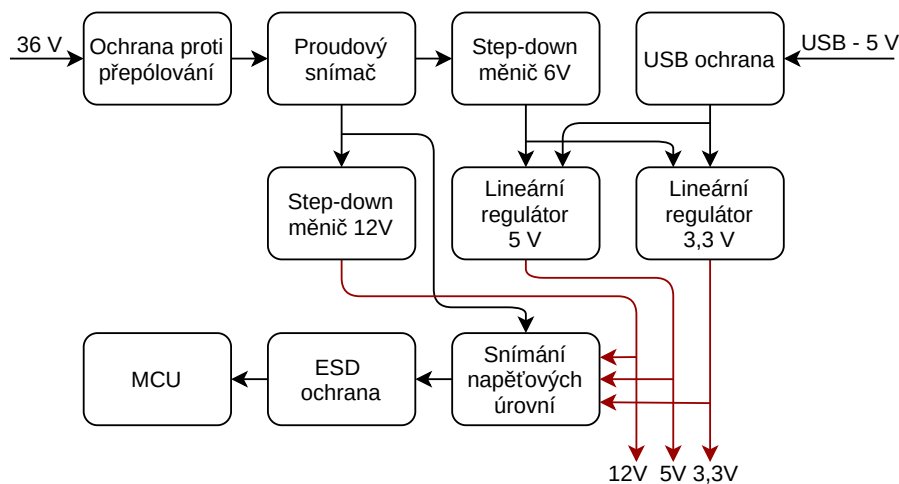
Pro komunikaci s deskou je použito rozhraní USB 2.0. USB kabel se připojuje k desce přes konektor USB-B, ten byl vybrán jelikož na desce příliš nezáleží na velikosti konektoru a jedná se o nejdolnější z relevantních variant. Mikrokontrolér obsahuje rozhraní pro připojení USB, není tedy potřeba žádný externí obvod, který by převáděl USB na sériovou linku. Pro ochranu komunikačních linek před elektrostatickými výboji je použit integrovaný ob-

vod USBLC6-2 [3], který chrání sběrnici před výboji až do napětí 8 kV. Ochranný obvod má velmi nízkou kapacitu, která je maximálně 3,5 pF. Stínění konektoru je připojeno k zemnicí ploše plošného spoje přes feritovou perlu. Ta má při frekvenci 100 MHz impedanci 22  $\Omega$ , přičemž je osazena co nejbližše samotnému konektoru, aby maximálně potlačila potenciální rušení.

Na datových linkách jsou osazeny rezistory o hodnotě 22  $\Omega$  sloužící jako terminační rezistor, aby se hodnota impedance spoje co nejvíce přiblížila hodnotě ze specifikace USB, což je 30  $\Omega$ . Samotné cesty jsou vedeny jako diferenciální spoj [25]. Tedy tak, aby byl rozdíl v délce mezi jednotlivými vodiči minimální. Stejně tak je přizpůsobena impedance mezi datovými cestami, aby odpovídala hodnotě 90  $\Omega$ .

Na desce je dále osazen konektor pro připojení IDC kabelu s 10 žilami. Na tento konektor jsou vyvedeny datové a časovací vodiče nejvíce využívaných sběrnic a rozhraní. Obsahuje také ladící sériové rozhraní, přes které je možné připojit se k terminálu běžícím v mikrokontroléru. Rozmístění pinů v konektoru je kompatibilní s rozmístěním používaném v logických analyzátoch Saleae. Tento analyzátor lze tedy k desce přímo připojit a sledovat komunikaci mezi periferiemi a řídicím mikrokontrolérem.

### 3.5.2 Napájení a spotřeba



Obrázek 3.5: Blokové schéma napájení

Napájení základní desky je rozděleno na tři části. Dvě zajišťují napájení pro krokové motory a jsou popsány v kapitole 3.5.3. Zbývající část zajišťuje napájení logické části desky, tedy mikrokontrolér a sekundární obvody, osvětlení, záměrný bod a ventilátory. Jelikož všechny tyto zmíněné obvody pracují na nižším napětí než je napětí zdroje, tedy 36 V, obsahuje základní deska obvody pro snížení tohoto napětí na hodnoty 12 V, 5 V a 3,3 V. Tato část napájení je blokově znázorněna na obrázku 3.5.

První částí této napájecí sekce je ochrana proti přepólování, která je realizovaná pomocí MOSFET tranzistoru s P-kanálem [13] (dále označovaný jako P-MOSFET). Pokud je napájecí napětí zapojeno správně, tak je tranzistor otevřen, jelikož napětí  $V_{GS}$  je větší než prahové napětí  $V_{GS(th)}$ . V případě, že je napětí připojeno opačně je tranzistor uzavřen, jelikož napětí  $V_{GS}$  není dostatečné pro otevření tranzistoru. V propustném případě je  $V_{GS}$  větší než maximální povolené, což pro použitý tranzistor je maximálním napětí  $V_{GS} \pm 20$  V.

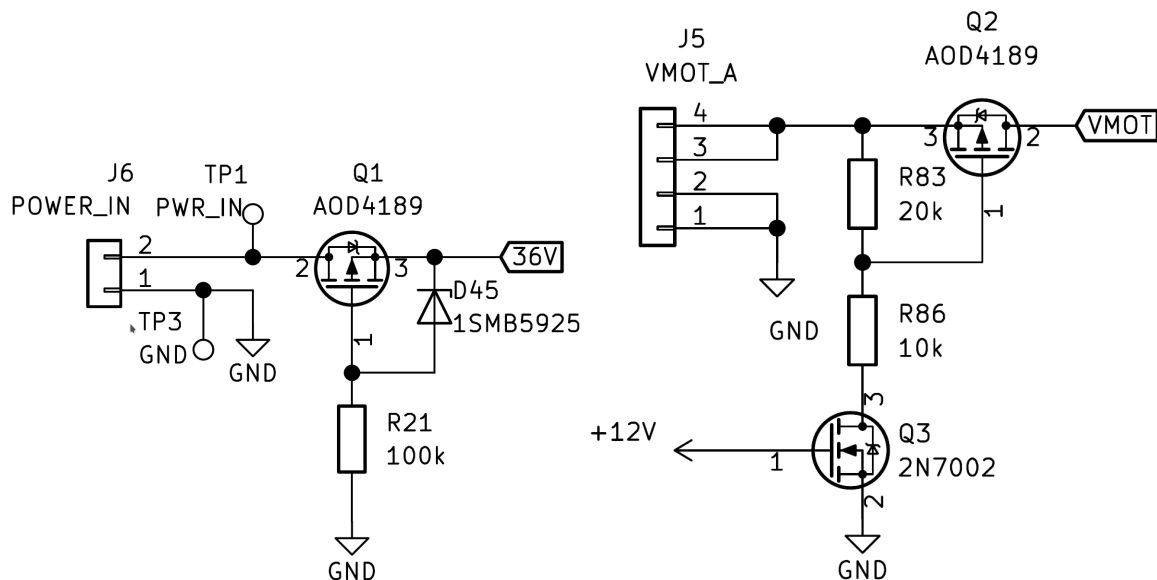
K ochraně P-MOSFET tranzistoru slouží zenerova dioda, které zajistí, že maximální napětí  $V_{GS}$  bude na úrovni zenerova napětí diody, pro použitou diodu 1SMB592 je zenerovo napětí 10 V [41]. Rezistor zapojený sériově s diodou limituje proud diodou. Jelikož plně otevřený tranzistor má odpor jen 22 m $\Omega$ , tak při procházejícím proudu 10 A je výkonová ztráta 2,2 W, což je výkon, který je pouzdro schopno vyžářit při teplotě 50°C bez přídavného chlazení. Při normálním provozu je proud většinou pod 1 A, tedy výkonová ztráta je méně než 0,022 W. Schéma zapojení je na obrázku 3.6a.

Následuje blok pro měření spotřeby, který využívá proudový snímač ACS712ELCTR [18] založený na Hallově jevu, který je popsán v kapitole 2.3.1. Výstup senzoru je připojen na ochranu diodu, která zajišťuje maximální limitaci napětí na hodnotu 3,3V. Jelikož odezva senzoru je 100 mV/A, maximální snímatelný proud odpovídá 8 A, což je zhruba čtyřikrát více než maximální předpokládané zatížení. Řešením této limitace by bylo zapojení výstupu senzoru na napěťový dělič, což by mělo ovšem za následek snížení rozlišení. Proudový senzor obsahuje vstupní pin pro připojení filtračního kondenzátoru o hodnotě 1  $\mu$ F, který spolu s integrovaným rezistorem o hodnotě 1,7 k $\Omega$  tvoří dolno-propustný filtr s mezní frekvencí 93,6 Hz. Díky tomuto filtru jsou odstraněny velmi krátké pulsní odběry proudu, čímž nedochází ke zkreslení v případě měření spotřeby zařízení. Výstup senzoru je připojen na vstup mikrokontroléru.

Na desce jsou dále dva impulsní zdroje založeny na integrovaném obvodu L5973D [1]. Jedná se snižující měnič s maximálním napájecím napětím 36 V a výstupním proudem až 2,5 A. Dále obsahuje zabudovanou ochranu proti přetížení, přehřátí a neočekávanému odpojení zátěže. Pouzdro obsahuje také integrovaný spínací tranzistor, čímž se minimalizuje množství externích součástek. Výstupní napětí se nastavuje pomocí zpětnovazebního děliče. K hornímu rezistoru děliče je vždy připojen potenciometr, který umožňuje jemné doladění výstupního napětí. První dělič má výstupní napětí 12 V, které slouží primárně pro napájení ventilátorů. Druhý měnič má výstupní napětí 6 V, které slouží jako vstup lineárních regulátorů, které snižují napětí dále na 5 V a 3,3 V. Spínaný zdroj je předřazen před lineární regulátory z důvodu zvýšení efektivity. Při použití samotného lineárního regulátoru připojeného přímo na 36 V a výstupním napětím 5 V je efektivita 13,8 %. S využitím spínaného zdroje se efektivita při odběru vyšším než 500 mA pohybuje okolo 75 %.

Napájení logiky je také možné realizovat přes USB. V tomto případě ovšem není funkční 12 V větev napájení a to z důvodu, že napětí z USB konektoru nahrazuje 6 V impulsní zdroj. Napájení z USB je připojeno přes polymerovou pojistku s nominálním spínacím proudem 200 mA a schottkyho diodou, aby se zabránilo zvýšení napětí na USB. To má ale za následek i napěťovou ztrátu na diodě a to 0,26 V. Díky tomu je napětí na 5 V větvi jen 4,7 V při napájení z USB. To je ovšem dostatečné napětí pro funkčnost všech obvodů napájených touto větví. Z USB je samozřejmě taky napájena 3,3 V větev, tedy mikrokontrolér a ostatní obvodu mohou pracovat jako při běžném napájení.

Všechna důležitá napětí jsou monitorována z mikrokontroléru. Kvůli tomu je každé napětí vyšší než 3,3 V pomocí odporového děliče sníženo pod tuto hranici. Kromě děliče jsou tato napětí vedena přes dolno-propustný filtr, který filtruje zákmity, filtry mají mezní frekvenci mezi 50 a 200 Hz. Tato napětí jsou také chráněna ochrannými ESD diodami, které zabraňují výbojům. Dále jsou přivedeny na vstupy mikrokontroléru, které je možné měřit pomocí AD převodníku. Takto sledovaná napětí jsou vstupní napájení 36 V, napájecí větve 12 V, 5 V, 3,3 V a napětí USB.



(a) Ochrana proti opačné polaritě napájení

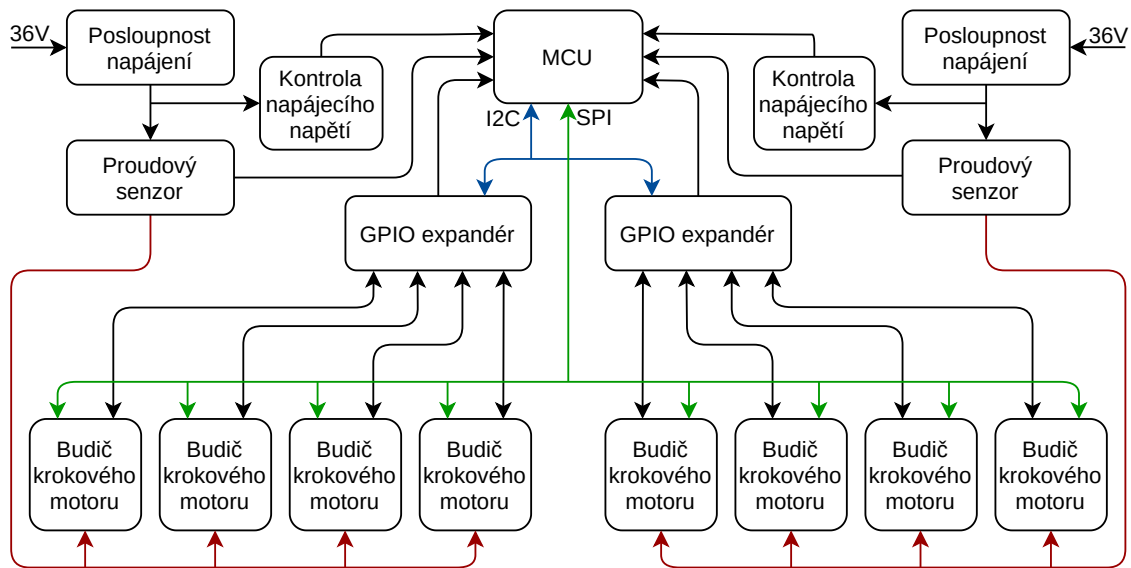
(b) Zapojení zajišťující pořadí napájení

Obrázek 3.6: Zapojení bránící špatnému připojení napájení

U budičů je nutné zajistit, že hlavní napájení bude přivedeno k čipu až po napětí pro napájení logiky čipu. V opačném případě může dojít i k poškození obvodu. Z toho důvodu obsahuje napájení budičů tranzistor s kanálem typu P, ten je spínán pomocí dalšího tranzistoru s kanálem typu N, který je spínán napětím z 12 V napájecí větve. Pokud tedy nebude aktivní napájení logické části (12 V větev) nedojde ani k aktivaci výkonového napájení pro budiče. Čímž je zajištěno pořadí napájení. Následuje proudový senzor, zapojený stejně jako pro měření proudu v logické části desky. Výstup senzoru je také připojen k mikrokontroléru. Napájecí napětí motorů je přes odporový dělič přivedeno na další vstup mikrokontroléru a tedy i toto napětí je možné sledovat. Zapojení zajišťující pořadí napájení je zobrazeno na obrázku 3.6b. Celé schéma zapojení napájecí části desky ve schématu v příloze A na straně 2 schématu. Obvod zařizující posloupnost napětí je na stranách schématu 3 a 18.



### 3.5.3 Budiče krokových motorů



Obrázek 3.7: Blokové schéma napájení

Budiče krokovým motorů jsou rozděleny na dvě části, kde každá obsahuje 4 budiče a vlastní napájení. Každá část má také vlastní GPIO expandér popsany dále.

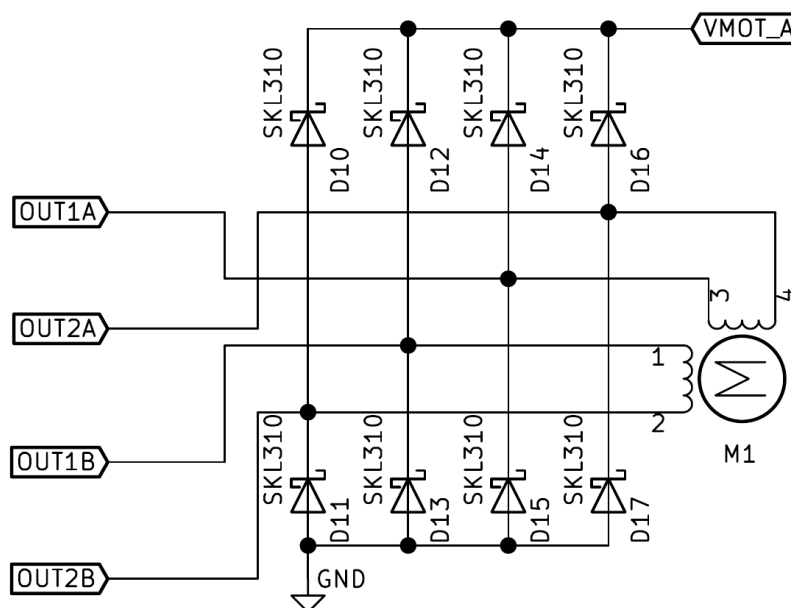
Samotné budiče jsou připojeny přes rozhraní SPI, přes které jsou předávány příkazy. Jelikož mikrokontrolér má omezený počet vstupně výstupních pinů, je využit GPIO expandér, který obsahuje 16 libovolně použitelných pinů. Tento expandér je připojen přes I2C sběrnici. Každý expandér využívá pro komunikaci s budičem 4 vodiče, které primárně slouží k výběru zařízení se kterým bude MCU komunikovat, generující ChipSelect signál. Také slouží k přenosu informace o tom, jestli je nastaven některý z příznaků FLAG nebo BUSY na budičích. Tyto příznaky jsou aktivní v nízké logické úrovni. Také expandér zprostředkovává čtení stavu koncového spínače, i když je tato informace dostupná rovněž přes stavové registry budiče. Každý expander má tedy plně využitě porty. Signály BUSY a FLAG každého budiče jsou také signalizovány LED prvky. Zapojení expandéru včetně ochrany napájení a signalizace je v příloze A na stranách 3 a 18. Zapojení samotných budičů obsahují strany 4, 5, 19 a 20.

#### Ochrana budičů

Jelikož krokový motor může sloužit také jako generátor, obsahuje deska ochranu proti vygenerovanému napětí. To může vzniknout například pokud by se někdo pokusil manuálně pootočit některou částí manipulátoru. Jelikož jsou některé motory zpřevodovány, může mít pomalý pohyb za následek velký počet otáček motoru a tím i vyšší napětí na výstupu motoru. Toto napětí by mohlo následně při překročení limitu poškodit budič krokového motoru, případně na něj navazující další elektroniku. Napětí vzniká na cívkách motoru, je tedy nutné chránit 2 vinutí. Ochrana je řešena pomocí shottkyho diod, které jsou umístěny vždy mezi zemnicí vedení a napájecí vodič cívky. Další dioda je umístěna mezi napájecí vodič cívky a hlavní napájecí větev. Tyto diody také chrání budič v případě, že by došlo



k neočekávanému odpojení motoru v případě, kdy jsou cívky pod napětím. Schéma zapojení je zobrazeno na obrázku 3.8.



Obrázek 3.8: Schéma zapojení ochranných diod

Použité diody mají označení SKL310, jejich prahové napětí  $V_F$  se pohybuje okolo 0,65 V. Maximální napětí v záměrném směru je až 100 V a maximální průřezový proud je 3 A. Tedy pokud by došlo k tomu, že bude generováno napětí větší než napájecí a prahové napětí diody, bude dioda otevřena a sloužit pro limitaci výstupního napětí. Takové diody jsou integrovány i do budiče, nejsou ovšem schopny snášet tak vysoké napětí jako přidané externí diody.

Jelikož pro napájení je použit spínaný zdroj může, vlivem kolísající zátěže, docházet k výraznějším výkyvům napětí než u klasického transformátorového zdroje. K potlačení tohoto efektu jsou co nejlépe každého budiče osazeny keramické kondenzátory s velmi nízkým sériovým odporem (dále zkráceno jako ESR). Při nabíjení těchto kondenzátorů může dojít k výrazným napěťovým špičkám a to hlavně během zapínání zařízení [46]. Tyto špičky jsou způsobeny parazitní induktancí vodičů, kolem kterých se vytvoří magnetické pole, které následně po nabití kondenzátoru na napětí zdroje zkolabuje a má za následek další zvýšení napětí. Často je toto napětí i několikrát vyšší než napájecí napětí. Toto napětí může potenciálně poškodit velké množství připojených obvodů. Tento efekt se opakuje několikrát, dokud se napětí neustálí na napětí zdroje. Snížení napěťových špiček je možné snížením indukčnosti vedení, což není kvůli návrhu plošného spoje možné v dostatečné míře, případně zpomalením nabíjení kondenzátoru a to přidáním odporu sériově s kondenzátorem, čímž se ovšem výrazně zhorší jeho vlastnosti. Nejlepším řešením tohoto problému je přidání dalšího paralelně zapojeného kondenzátoru s výrazně vyšší kapacitou, většinou se tedy jedná o elektrolytický kondenzátor, které mají řadově větší ESR. Tímto kondenzátorem nejsou ovlivněny parametry menšího keramického jen je zpomaleno jeho nabíjení.

Kondenzátory určené pro výkonovou část budičů jsou nabíjeny na napětí 36 V. Keramické kondenzátory jsou dimenzovány na maximální napětí až 50 V, což by v případě napěťových špiček nemuselo být dostatečné, proto jsou co nejlépe osazeny elektrolytické

kondenzátory o hodnotě 100 nF a s maximálním napětím až 63 V. Při této kombinaci jsou napěťové špičky maximálně eliminovány a dosahují tak maximální velikosti 1 V nad napájecí napětí, což je daleko od potenciálně nebezpečného napětí pro kondenzátory a součástky. Pro logickou část napájení, s napětím 3,3 V, jsou použity stejné keramické kondenzátory jako pro výkonovou část a to hlavně z důvodu minimalizace počtu odlišných součástek. Elektrolytické kondenzátory mají maximální napětí jen 6,3 V a to z toho důvodu, že kondenzátory dimenzované na napětí 63 V jsou výrazně větší a dražší, na desce zabírají zhruba 3×větší plochu.

### Koncový snímač

Koncový snímač je zapojen tak že jeho výstup je v normální stavu rozpojen. Vstup budiče je pomocí integrovaného pull-up rezistoru nastaven na vysokou logickou úroveň a při sepnutí spínače se počítá s tím že na vstup bude logická nula, proto jsou koncové snímače připojeny mezi snímací vstup a zem. Mezi koncovým snímačem je osazen ještě dolno-propustný filtr, který má za účel odstranit zákmity tlačítka, které by mohly způsobit neočekávané chování budiče. Na základě dokumentace k demonstrační desce obsahující stejný integrovaný budič je filtr navržen tak, že jeho mezní frekvence je 1,591 kHz.

### Expandér GPIO - MCP23017

Expandéry MCP23017 [9] slouží pro komunikaci s budiči a jelikož se jedná spíše jen o přenos binárních příznaků není nutné využívat vstupně výstupní porty mikrokontroléru. Navíc pro tuto komunikaci je potřeba velké množství cest na plošném spoji, což by vedlo na složitější návrh v okolí mikrokontroléru. Také by bylo nutné využít obvod z vyšší řady obsahující více vstupně výstupních portů a to nejspíše v pouzdře typu BGA. K mikrokontroléru jsou expandéry připojeny přes sběrnici I2C, jejíž komunikační frekvence je nastavena na hodnotu 400 kHz, přestože je tento typ expandéru schopen komunikovat až na frekvenci 1 MHz. Vyšší rychlost není využita z důvodu, že na sběrnici jsou také připojena další zařízení, která tuto vysokou frekvenci nepodporují. Datové a časovací linky jsou stejně jako SPI chráněny pomocí ESD diod.

Expandéry jsou napájeny z 3,3 V napájecí větve. K tomuto napětí jsou poté vztaženy i logické úrovně vstupů/výstupů. Z obou expandérů jsou vyvedeny obě linky přenášející informaci o přerušení. Pro výběr adresy slouží tři vstupy, kterými je možné konfigurovat poslední tři bity I2C adresy.

#### 3.5.4 Kontrola a řízení teploty

Pro měření teploty slouží NTC termistory připojené na vstupy AD převodníku uvnitř mikrokontroléru. A na základě snížení odporu těchto termistorů lze určit teplotu povrchu, ke kterému jsou připojeny. Předpokládá se připojení termistorů s nominální hodnotou 100 kΩ. Termistory mohou být připojeny ke krytu napájecího zdroje, který zároveň slouží jako chladič zdroje a na základě jeho teploty poté mohou být regulovány ventilátory. Dále může být termistor připojen na chladič pro budiče krokových motorů a regulovat podle jeho teploty ventilátory chladičí přímo desku. V případě, že zařízení nebude v neustálém běhu je velmi pravděpodobné, že bude postačovat pasivní chlazení.

Dalšími možnostmi pro měření teploty je integrovaný obvod, sloužící pro měření teploty plošného spoje, a to kvůli ohřívání způsobenému hlavně budiči krokových motorů skrz

zemnicí plochu. Dále teplotní senzor integrovaný do mikrokontroléru. Ten slouží spíše jen pro měření interní teploty obvodu a není příliš ovlivňován okolními vlivy.

### **Řízení otáček ventilátoru - EMC2305**

Pro řízení otáček ventilátorů slouží integrovaný obvod EMC2305 [2]. Ten umožňuje nezávisle řídit až pět ventilátorů. U ventilátorů se předpokládá vstup pro PWM signál, ovšem za předpokladu, že ventilátor tento vstup nemá a jeho otáčky jsou ovládány analogově. Nastavením vstupního napětí je možné přepnout obvod do režimu, kdy je PWM na vyšší frekvenci 26 kHz a PWM výstup použit pro napájení ventilátoru, což je pro takové řízení vhodnější než běžné nízké frekvence okolo 1 kHz, které se používají pro ventilátory říditelné pomocí PWM signálu.

Obvod také umožňuje operovat jako zařízení s uzavřenou regulační smyčkou, kdy zpětná vazba je generována pomocí tachometrického signálu ventilátoru. Jedná se pulsy na výstupu ventilátoru, ke kterým dochází dvakrát za otočení ventilátoru. Díky tomu je možné detekovat otáčky ventilátoru a přesně je tak regulovat. Také je možné kontrolovat jestli nedošlo k zaseknutí ventilátoru. Lze tak dříve zabránit přehřívání kritických oblastí. V případě, že obvod detekuje zaseknutí signalizuje jej přes přerušovací linku mikrokontroléru. Ten poté z registrů regulátoru zjistí o který ventilátor se jedná.

Obvod také umožňuje postupnou akceleraci otáček. Pro dodržení správných otáček a maximální akceleraci obsahuje obvod pro každý kanál vestavěný PID regulátor 2.2. Koefficienty jednotlivých členů je možné nastavit přes registry obvodu a upravit tak regulátor pro různé typy ventilátorů.

Regulátor je připojen přes sběrnici SMBus, která je odvozena od sběrnice I2C. Regulátor je připojen na I2C sběrnici společnou pro všechna zařízení, jelikož tento obvod podporuje komunikaci na frekvenci 400 kHz. Adresa zařízení je nastavována pomocí externího rezistoru, připojeného na jeden ze vstupů a napájecí napětí. Podle hodnoty rezistoru se následně určuje adresa.

### **3.5.5 Záměrný bod**

Záměrný bod musí být pohyblivý a to kvůli tomu, že jej ideálně bude snímáný pacient sledovat celou dobu během snímání. V případě, že by se zařízení natočilo, aby pořídilo snímek z jiného úhlu, tak se by stejně pohnul i záměrný bod. A pokud by jej pacient sledoval výsledný snímek, by byl stejný jako ten před natočením kamery. Proto se záměrný bod musí posunout při každé změně úhlu kamery a tím zajistí, že oko pacienta bude mít ideálně fixní polohu.

Pro realizaci byl vybrán OLED displej, jelikož je poté možné snadno měnit pozici záměrného bodu. OLED displej byl vybrán hlavně proto, že neobsahuje žádné přídavné podsvícení a do snímací soustavy tak nepřispívá žádným nežádoucím světlem.

Kvůli návrhu fundus kamery a snímací soustavy je maximální možné natočení zhruba 15–20°. Každých 5° náklonu kamery musí být kompenzováno posunem záměrného bodu o 1,26 mm v opačném směru. Samotná zobrazovací plocha displeje má rozměry 21,8×10,9 mm. Velikost tedy ideálně odpovídá, jelikož při maximálním náklonu 20° bude záměrný bod posunut o 5,04 mm od středu displeje. Kvůli rozlišení displeje není možné jemně pohybovat se záměrným bodem, protože displej má rozlišení pouze 128×64 pixelů. To má za následek, že množství snímacích pozic bude muset být uzpůsobeno přesnosti umístění bodu na displeji.

Ovládání modulu s displejem je realizováno přes sběrnici I2C. Skrz ni jsou nahrávána data do bufferu displeje, který je poté zobrazován. Displej je napájen z 5 V napájecí větve.

Během návrhu se počítalo i s jinými displeji, přičemž některé nebyly typu OLED, je možné napájení odpojit pomocí spínacího tranzistoru. A to z důvodu omezení světla generovaného podsvícením displeje například u LCD displejů, které všem mohou poskytnou vyšší hustotu pixelů a tím i vyšší přesnost umístění záměrného bodu.

## 3.6 Osvětlení sítnice

Osvětlení je realizováno jako pomocná deska umístěna uvnitř fundus kamery, která se připojuje ke konektorům pro osvětlení na hlavní řídicí desce. Tato deska obsahuje pouze dvě sady po čtyřech LED diodách a reflexivní snímač pro kontrolu pozice polarizačního filtru. Diody z každé sady jsou zapojeny sériově. Jsou rovnoměrně rozmístěny do kruhu okolo otvoru pro průchod světla směrem ke kameře.

Jako bílé LED jsou použity diody CREE-CLM3C [5], maximální doporučený proud diodou je 20 mA při napětí 3,2 V. Konkrétně se jedná o variantu se studeným světlem o teplotě 5 500 K. Infračervené diody jsou OIS-330 770[10]. Tyto diody obsahují integrovanou optiku s vyzařovacím úhlem 40°, jejich typická vlnová délka odpovídá 770 nm. Maximální proud je 50 mA při napětí 1,65 V. Obě varianty diod mají typickou lineární závislost jasu na proudu. Kvůli nízkému prahovému napětí infračervených LED je napětí nutné pro jejich řízení velmi malé, jen 6,6 V, z důvodu zvýšení efektivity napájecího zdroje pro diody je využit impulsní zdroj 2.6.1.

Vlnová délka světla se pro snímání sítnice většinou využívá v rozmezí vlnových délek 500–600 nm [35], tedy téměř infračervené světlo. Tato vlnová délka se většinou využívá kvůli tomu, že je při ní obraz nejvíce kontrastní a nejvíce vynikne okysličená krev a díky tomu i celé krevní řečiště. Výhodou vyšších vlnových délek jsou snímky s vyšším špičkovým poměrem signálu k šumu a to díky tomu, že při vyšší vlnové délce lze použít vyšší intenzitu světla [28]. Další výhodou je také, že sítnice lépe odráží delší vlnové délky. Při delších vlnových délkách také výrazněji vynikají věkem podnícené makulární degradace, které mohou vyústit až ke ztrátě zraku [21].

V dalších fázích experimentů se zařízením je možné diody vyměnit za jiné, za předpokladu, že je bude možné osadit na desku. Kvůli této variantě jsou na desce použity pájecí plošky umožňující bezproblémové osazení pouzder 1206, 0803, ale i PLCC2. Zdroj pro ovládní diod je dimenzovaný tak, aby bylo možné využít sériové zapojení diod s maximálních součtem napětí až 28 V. Maximální proud je možné nastavit rezistorem a výchozí varianta obsahuje nastavení pro maximální proud 50 mA.

### 3.6.1 Proudový zdroj pro LED - NCL30160

Jako řídicí obvod pro zdroj napájení LED prvků slouží NCL30160 [4] od firmy OnSemiconductors. Jedná se o konstantní proudový impulsní zdroj vhodný pro řízení výkonových LED s maximálním výstupním proudem až 1 A. Slouží pro snižování napětí na takovou úroveň, kdy bude dosažen požadovaný proud, obsahuje vestavěný tranzistor pro regulování výkonu. Pracovní rozsah napětí je od 6,3 V do 40 V.

Pracovní frekvence impulsního zdroje se pohybuje mezi 600 kHz až do 1,4 MHz. Pro zajištění ochrany obvodu obsahuje čidlo teploty a při překročení teploty 165 °V dojde k vypnutí obvodu. Dále také obsahuje ochranu proti zkratu připojených diod. Maximální zvlnění napětí by měl odpovídat 5 % střední hodnoty napětí. Regulace napětí má dvě fáze. První, kdy je regulační tranzistor sepnut a napětí postupně narůstá a druhá fáze, kdy dojde k odpojení napájecího napětí a výstupní tak postupně klesá. Fáze mají délku označenou jako

$t_{ON}$  pro první fázi a  $t_{OFF}$  pro druhou. Aby bylo možné dodržet tyto časy i při inicializaci obvodu, kdy není možné sledovat výstupní napětí, musí být čas  $t_{OFF}$  nastaven pomocí externí rezistoru  $R_{OT}$ . Pro výpočet hodnoty rezistoru je použit vzorec 3.1.

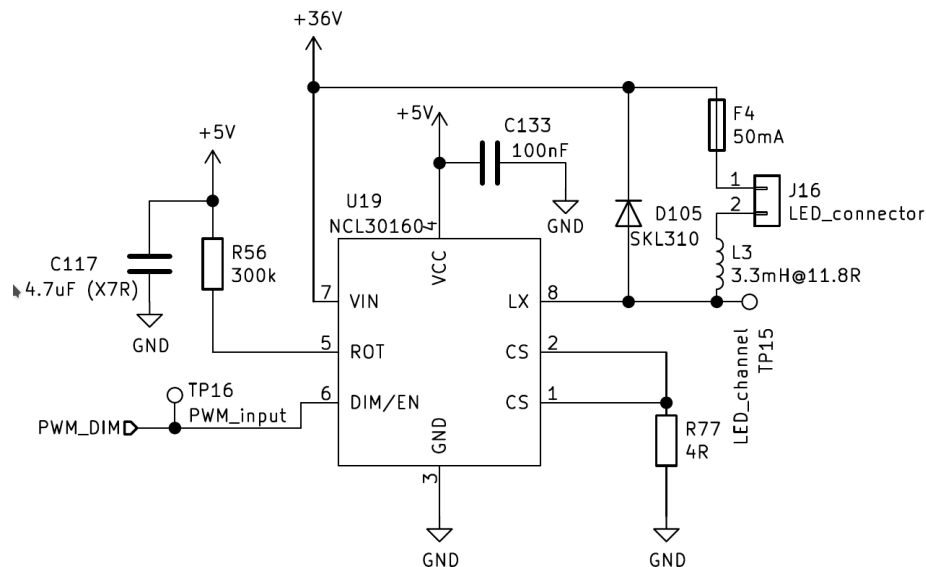
$$R_{OT} = \frac{L \cdot \Delta I}{V_{IN} - V_{LED} - I_{OUT} \cdot (FET_{RDS(on)} + DCR_L + R_{SENSE})} \cdot 10^{-11} \quad (3.1)$$

kde:	$L$	– indukčnost cívky
	$\Delta I$	– velikost maximální zvlnění výstupního proudu
	$V_{IN}$	– napájecí napětí
	$V_{LED}$	– součet napětí na diodách při maximálním nastaveném proudu
	$I_{OUT}$	– proud diodami
	$FET_{RDS(on)}$	– odpor regulačního tranzistoru v sepnutém stavu
	$DCR_L$	– sériový odpor cívky
	$R_{SENSE}$	– nominální hodnota snímacího odporu

Vybraná cívka má indukčnost 3,3 mH, odpor 16,5  $\Omega$  a maximální pracovní proud 100 mA, impulsně po dobu maximálně 1 s až 200 mA. Předpokládané přípustné zvlnění je 2,5 mA při výstupních 50 mA. Snímací odpor má hodnotu 4  $\Omega$ . Odpor tranzistoru v sepnutém stavu je 55 m $\Omega$ . Pro uvedené součástky a infračervené diody s prahovým napětí 1,6 V vychází hodnota časovacího rezistoru  $R_{OT}$  na 761,09 k $\Omega$ . Pro použití bílé LED s prahovým napětím 3,2 V musí být použit časovací rezistor o hodnotě 408,8 k $\Omega$ .

Proud obvodem je nastavován pomocí externího snímacího rezistoru. S tím že napěťová reference obvodu je 0,2 V. Na základě Ohmova zákona tedy pro proud 50 mA vychází snímací odpor o hodnotě 4  $\Omega$ . Pouzdro odporu musí být dimenzované na proud procházející LED. Na základě kapitoly 2.3 je vhodné vybrat pouzdro s dostatečnou rezervou. Špičkový výkon se pohybuje jen okolo 0,01 W. Což je hodnota, pro kterou lze použít i miniaturní pouzdro s označením 0201. Jelikož je ovšem možné, že v budoucnosti bude využit vyšší proud, je zvoleno pouzdro 1206, které je dimenzováno pro výkon 0,25 W. Za předpokladu, že bude dodržena rezerva kvůli ohřívání rezistoru je možné osadit snímací rezistor pro nastavení maximální proudu až do 300 mA. Takový proud by měl u většiny diod za následek velmi vysoký jas, schopný poškodit lidské oko.

Důležitou funkcí obvodu je také možnost snížit jas diod v závislosti na vstupním řídicím signálu. Jedná se o PWM signál generovaný řídicím mikrokontrolérem. Tento signál musí mít frekvenci větší než 100 Hz a ideálně menší než 20 kHz. Na základě střídý signálu je poté proporcionálně snížen proud protékající diodami. Proud je nastavitelný v celém rozsahu lineárně od 0 do 100 %. V základním zapojení je tento signál deaktivovaný pomocí pull-up rezistoru, takže po připojení jsou diody řízeny maximálním nastaveným proudem. Je důležité, aby vždy když je obvod aktivní a nastaven tak, aby napájel LED prvky libovolným proudem. Je-li tedy aplikován PWM signál se střídou větší než 0 %, musí být tyto diody připojeny k obvodu. V opačném případě může velmi pravděpodobně dojít ke zničení integrovaného regulačního tranzistoru, což lze vyřešit jen výměnou řídicího obvodu.



Obrázek 3.9: Schéma zapojení obvodu pro řízení jasu s řídicím obvodem NCL30160

Celé schéma zapojení obvodu pro řízení osvětlení je zobrazeno na obrázku 3.9. Na konektor označený jako LED\_connector se připojují sériově zapojené diody osvětlující sítnici. Signál označený jako PWM\_DIM slouží pro nastavení proudu v rozsahu definovaném snímacím rezistorem R77. Výsledná deska obsahuje nakonec tři instance tohoto obvodu, přičemž každou je možné nastavit a řídit samostatně.

Za účelem maximální ochrany snímané osoby je sériově s LED diodami zapojena pojistka F4 s maximálním povoleným proudem vybraným tak, aby nehrozilo poškození oka a to 50 mA. Pro testovací verzi desky jsou osazeny polymerové pojistky. Tyto pojistky jsou vícenásobně použitelné, v případě překročení maximálního povoleného proudu se ohřejí natolik, že se jejich proud mnohonásobně navýší a přestanou tak téměř vést proud, jedná se vlastně o nelineárně termistor. Po snížení proudu a zchlazení proud opět vedou. Jejich nevýhodou je dlouhá doba reakce. Při nominálním spínacím proudu může být obvod přerušen až za 0,5 až 1,5 s. Což může být nebezpečně dlouhá doba. I v případě, kdy prochází proud o hodnotě 0,5 A je reakční doba pojistky až 0,1 s [7]. Pro produkční verzi se tedy více hodí klasické tavné pojistky. Jejich výhodou je naopak velmi rychlá doba reakce. Při proudu 0,5 A reaguje typicky za 0,1 ms, tedy až 1 000×rychleji než PTC pojistka [19].

Potenciální problém může nastat v případě, že by docházelo k výraznějšímu ohřívání pojistky, jelikož se jejím ohříváním by se zvyšoval i její odpor a snižoval protékající proud. V případě pomalého zvyšování odporu pojistky by mohlo dojít k tomu, že zdroj bude toto snížení kompenzovat zvýšením napětí na diodách. Čímž by došlo k jejich zničení, při nejmenším alespoň jedné s z nich. Při proudu pod 50 mA by nemělo dojít k ohřívání tak velkému, aby k tomuto došlo.



### 3.7 Návrh plošného spoje

Samotný plošný spoj má velikost  $200 \times 100$  mm. Velikost plošného spoje je spíše dána počtem konektorů, které jsou osazeny okolo okrajů desky, než množstvím samotných součástek. Konektory jsou umístěny jen na okrajích, aby bylo poté možné jednoduše osadit ventilátory, které by poté shora ochlazovaly desku bez toho, aby hrozilo, že se do jejich lopatek později zamotají dráty.

Deska je realizována na dvouvrstevném plošném spoji vyrobeném podle třídy přesnosti 5 a s minimální šířkou cesty 0,25 mm. Deska obsahuje celkem 481 součástek, včetně konektorů. Rezistory a kondenzátory, u kterých je to výkonově možné, jsou v pouzdrech 0603. Preferované pouzdro pro integrované obvody je typu QFN (Quad Flat No-leads Package) a to z důvodu, že pájení je prováděno pomocí horkého vzduchu.

Další preferencí pro výrobu použít desku, která má nanесenu dvojitou vrstvu mědi, tedy 2oz. To z důvodu, aby bylo zajištěno dimenzování výkonových cest na maximální možné proud. Při použití měděné vrstvy o běžné tloušťce jsou šířky cest navrženy tak, aby vydržely pouze předpokládaný proud namísto maximálního možného. Například napájecí cesty s šířkou 3 mm umožňují trvalé zatížení až 7,2 A. Což je předpokládaný proud v případě, že jsou výkonnější motory rovnoměrně rozloženy mezi obě skupiny motorů. Také se předpokládá souběžný pohyb všech motorů ve skupině, což není příliš pravděpodobné. Pro dvojitou vrstvu tloušťku měděné vrstvy je proudová zatížitelnost okolo 11,9 A. Což je více než dostatečné i pro výkonné motory. Všechny vypočtené hodnoty proudového zatížení počítají s maximálním zvýšením teploty měděné cesty o maximálně 20 °C nad okolní teplotu.

Osazené součástky také generují teplo, jedná se hlavně o budiče krokových motorů a impulsní měniče. Ty obsahují na spodní části pouzdra plochu, která se připájí k zemnicí ploše desky a má za účel zvýšit odvod tepla. Aby nedocházelo k lokálnímu ohřívání zemnicí plochy, obsahuje deska termální prokovy. Jedná se o průchodky mezi zemnicími vrstvami, které jsou umístěny v mřížce, vzdálenost jednotlivých prokovů je 1,5 mm. Jsou umístěny všude tam, kde je možné umístit průchodku tak, aby byla vzdálena minimálně 1 mm od nejbližší cesty, pájecí plošky nebo průchodky, která není učena pro rozvod tepla. Tyto prokovy také zkracují délku efektivní cesty proudu v zemnicí ploše a omezují vznik úzkých průchodů [38]. Díky prokovům se také zvyšuje celková tepelná kapacita plochy, která slouží pro chlazení osazených obvodů.

Deska také obsahuje otvory, kterými je možné připevnit chladič usazený na budiče krokových motorů. Ideální chladičem je chladicí extrudovaný profil a délce 200 mm a šířce až 22 mm. Výška chladiče omezena není, ovšem předpokládaná vzdálenost ventilátorů od desky je 40 mm. Použitý chladič na výsledném zařízení má šířku 20 mm, výšku 14 mm a celkem 5 žebírek po celé délce profilu. Budiče nemusí mít stejnou výšku od plošného spoje a to z důvodu, že spodní plocha je připájena k desce a může pod nimi být různé množství pájky. Výškový rozdíl může být až v řádu desetin milimetru, většinou se ovšem jedná desítky  $\mu\text{m}$ . Tento rozdíl ve výšce mezi jednotlivými obvody může mít za následek zhoršení přenosu tepla mezi některými z obvodů a chladičem, pokud by byl chladič velmi pevně utažen tak hrozí až poškození pouzdra čipu. Z tohoto důvodu a také ke zvýšení efektivity přenosu tepla mezi budičem a chladičem je mezi nimi vložena teplovodivá podložka. Finální deska obsahuje teplovodivé podložky o výšce 1 mm, určené pro chlazení pamětí na grafických kartách.

### 3.7.1 Návrh v programu KiCAD

Návrh plošného spoje je proveden v návrhovém programu KiCAD<sup>1</sup>. Jedná se o multiplatformní open-source nástroj v němž je možné provádět kompletní návrh plošného spoje. Výstupem tohoto návrhu je schéma plošného spoje v příloze **A**, dále seznam součástek, kterou jsou nutné k osazení desky v příloze **B** a osazovací plán desky v příloze **C**.

Podklady pro výrobu data ve formátu Gerber jsou v souborech přiložených k práci. Stejně tak data pro vrtání ve formátu Excellon. Při návrhu desky bylo předpokládáno ruční osazení, primárně pájecí pomocí horkého vzduchu. Z toho důvodu je většina pouzder typu QFN.

---

<sup>1</sup><https://kicad-pcb.org/>



## Kapitola 4

# Návrh a implementace softwaru

Veškerá funkčnost zařízení je implementována jako program pro mikrokontrolér umístěný na základní desce zařízení. Součástí této implementace je i rozhraní pro komunikaci s uživatelem realizované pomocí příkazové řádky.

Program mikrokontroléru je implementovaný v jazyce C++ s využitím standardu C++17. Hlavní výhodou implementace ve standardu C++17 je funkce dedukce argumentů pro šablonové třídy. Dále je využíváno zachytávání aktuálního objektu v rámci lambda funkcí. Program je založen na objektové hierarchii zobrazující strukturu zařízení. Konkrétní systém dekompozice zařízení je popsán v kapitole 4.3.

Výhodou implementace v C++ je snadnější kontrola nad rozhraními jednotlivých součástí, což se projevuje snadnější údržbou a rozšiřování existujících částí. Díky definovatelným rozhraním je snazší dodržet komunikační protokoly a i při zapouzdření komponent udržet přehlednost kódu.

Nevýhodami je poté složitější hledání chyb, jelikož rozdělení paměti není tak transparentní. Například při použití virtuálních tříd místo struktur. Kompilátor C++ také generuje větší binární soubor, tento rozdíl je nicméně výrazný je u velmi krátkých programů. Větší programy se stejnou funkčností mají přibližně podobnou velikost.

Téměř všechny koncepty z jazyka V je možné přenést do jazyka C++ a zařídit tím větší organizovanost programu například pomocí jmenných prostorů [30]. Například přetěžování operátorů v C++ může poskytnout větší přehlednost oproti funkcím zpracovávající ukazatele na struktury.

Důvod pro integraci komunikačního rozhraní ve formě příkazové řádky je kompatibilita výsledného zařízení. Díky integrovanému rozhraní pro uživatele není nutné, aby existovala ovládací aplikace na řídicím počítači. Odpadá tedy problém s externími závislostmi jako jsou dynamické knihovny a závislost na operačním systému. Aktuálně k ovládní stačí jen libovolný program schopný komunikovat přes sériovou linku. U Windows se jedná například o program Putty u zařízení založených na Linuxovém jádře je možné použít například `screen` nebo `picocom`. Komunikace může probíhat přes vestavěnou sériovou linku, případně ta může být emulována pomocí USB rozhraní v módu VCP (Virtual COM Port).

Za účelem přenositelnosti programu mezi jednotlivými mikrokontroléry, a případně i mezi řadami mikrokontroléru od firmy STMicroelectronics, je využita jimi poskytovaná abstraktní vrstva. HAL (Hardware Abstraction Layer) je knihovna funkcí, která zapouzdřuje práci s periferiemi. Existují různé verze pro všechny řady mikrokontrolérů, ovšem všechny mají stejné rozhraní funkcí. To umožňuje pracovat s periferiemi stejným způsobem na různých mikrokontrolérech, přestože samotné periferie se od sebe mohou lišit. Rozdílnost může být na úrovních od odlišného názvu registrů až po výrazné rozšíření funkčnosti.

Pro tyto nadstandardní funkce potom slouží rozšiřující moduly knihovny. Tato knihovna je ovšem napsaná v jazyce C a neposkytuje příliš velkou míru abstrakce.

Celý systém je založen na řetězení callback funkcí jejich start je inicializovaný z přerušení. Aby bylo možné takto volat nejen funkce, ale i metody objektů, musí být zapouzdřeny ve struktuře, kterou je možné uložit a dále použít. K tomu slouží `Invocation_wrapper`, popsany v následující sekci.

Za účelem zvýšení znovupoužitelnosti programových bloků byla v rámci této práce napsána objektově orientovaná knihovna pojmenovaná ALOHAL. Ta slouží jako další vrstva nad knihovnou HAL a využívá její funkce pro práci s periferiemi. Díky čemuž je také přenositelná mezi mikrokontroléry a je poměrně jednoduché rozšířit její kompatibilitu i na nové produkty za předpokladu, že pro ně existuje knihovna HAL, která je dostupná aktuálně pro všechny mikrokontroléry STM32. Knihovna ALOHAL vznikla na základě původní verze použité v bakalářské práci. Nicméně nyní se jedná o kompletně nový návrh a nové programové řešení.

Cílem knihovny ALOHAL je maximální znovupoužitelnost a univerzalita jejich součástí. Toho je dosaženo pomocí oddělení funkčních prvků do samotných bloků, které jsou postupně sestavovány. Například komunikační rozhraní UART je sestaveno z bloků pro samotnou komunikaci, management přerušení a DMA přenosy.

## 4.1 Invocation wrapper

Jedná se o třídu obsahující objekt a jeho metodu, druhou variantou je zapouzdření funkce. Všechny informace jsou uloženy ve formě ukazatele, tedy ukazatel na objekt a ukazatel na metodu třídy, ze které je uložený objekt instancionalizován nebo ve druhém případě ukazatel na funkci.

Aby mohly být takto zapouzdřené objekty uloženy do homogenní struktury, přestože obsahují zapouzdřené objekty pocházející z různých tříd, je běžně používaná třída `Invocation_wrapper` zděděna ze základní třídy `Invocation_wrapper_base`. V rámci dědičnosti získává také rozhraní (skupinu virtuálních metod), aby mohly být prováděny operace nad homogenní strukturou bez specializace konkrétní zapouzdřené třídy.

Aby bylo možné zapouzdřit různé třídy bez jejich předchozí znalosti, používá tato třída šablony. Šablona obsahuje tři argumenty a to je: zapouzdřená třída, návratový typ metody nebo funkce a datové typy argumentů funkce. Parametr obsažené třídy obsahuje pouze derivovaná třída `Invocation_wrapper`.

Jelikož není možné jako parametr šablony předávat `void` jako argument funkce kvůli definovanému rozhraní, ve kterém musí být návratový typ existovat, existuje pro tento případ specializace šablony. Ta se od běžné verze liší jen tím, že automaticky vkládá typ `void` jako argument volání a tím i mění rozhraní pro invokaci metody.

Jakmile je takovýto objekt vytvořen, může být provedena aktivace, tedy invokace metody nad uloženým objektem. K tomu slouží metoda `Invoke`. Tuto operaci je možno provést z libovolné části programu a to i z přerušení. Je ale nutné zaručit, že objekt na uložené adrese stále existuje a nebyl mezi uložení a vyvoláním dealokován. V rámci aktivace je taky možné předávat argumenty pro vykonání a po vykonání získat návratovou hodnotu metody/funkce.

Hlavní výhodou takového zapouzdření je možnost pracovat s objektem, přestože není známá jeho funkčnost. Funkčnost je definována už při zapouzdření a to výběrem metody, případně ji lze modifikovat argumenty, pokud to metoda podporuje. Následně lze během

přerušeni od tlačítka aktivovat odeslání zprávy přes sériovou linku a zároveň spustit regulační smyčku regulátoru.

Díky možnosti oddělit od sebe dva systémy, kdy jeden je jen spouští a druhý provede akci, je možné výrazně zvýšit znovupoužitelnost programu pro jednotlivé podsystémy. Například může existovat podsystém měřící teplotu a při překročení určité teploty vyvolat akci, tedy aktivovat zapouzdřený objekt, který mu je přidělen. Takový přidělený objekt může mít různé funkce podle určení celého systému. Například variantami aktivovaného objektu může být kontrolér ventilátoru, odeslání varovné zprávy nebo pokud se jedná o měření teploty procesoru, tak třeba jeho dynamické snížení frekvence.

Výsledkem oddělení mohou být samostatné bloky, které čekají na aktivaci a případně i aktivaci sami provádějí. Ty je následně možné slučovat do bloků a z nich sestavit celý program zařízení. Za předpokladu, že objekt umožňuje aktivovat více než jeden přiřazený objekt, je možné snadno i škálovat velikost a výkonost systému. Omezením už jsou poté jen požadavky jako je výkon a případné reakce v reálném čase.

Díky možnosti vytvářet zapouzdření samotných funkcí je možné i použití lambda funkce pro případ, že je nutné akci více specifikovat, případně invokovat metodu objektu s argumenty, které nemohou být vygenerovány procesem, jenž vede k aktivaci.

## 4.2 Systém přerušeni

Výše popsaná třída `Invocation_wrapper` je nejvíce využita pro spouštění programů, které jsou založeny na externích událostech. Ve vestavěných systémech se nejčastěji jedná o přerušeni, ale může jít být i vstup od uživatele.

Všechna přerušeni jsou založena na společné třídě, která obsahuje možnost registrace zapouzdřených objektů. Takových objektů je možné zaregistrovat libovolné množství. Třída spravující toto přerušeni může být zakomponována do jiné třídy spravující podsystém. Například třída představující periférii UART může obsahovat správce přerušeni, které vždy při aktivaci přerušovací rutiny aktivuje všechny registrované objekty. Těmi mohou být objekty, jež provedou odeslání potvrzovací zprávy zpět a dále kontrola zprávy jestli jsou ve správném formátu. Díky univerzalitě tohoto řešení je možné v průběhu implementace přidávat další registrované objekty, například uloženi zprávy do logovací paměti. To má za následek i snadnější modifikace produktu v případě změny specifikace, eventuálně tvorby další verze zařízení.

Pro případ, že dané třídy obsahují přerušeni z více zdrojů, je možné při registraci zadat klíč, určující ke kterému konkrétnímu přerušeni je zapouzdřená struktura registrována. Například pro UART je možné se registrovat pro aktivaci u přijatých zpráv řetězcem "receive" a pro odeslané "transmit". V případě registrace přerušeni u portů se jedná o číslo pinu. Registrace může být nicméně provedena i u samotného pinu, ovšem ten jen přesměruje požadavek na port, s tím že doplní své číslo.

## 4.3 Struktura zařízení

Celé zařízení včetně jeho funkčnosti a součástí je popsáno jednou kořenovou třídou pojmenovanou `Device`. Tato třída je univerzální pro všechny zařízení založená na vrstvě ALOHAL. Jedná se o modulární třídu, jejíž jednotlivé části lze měnit. Samotná třída slouží spíše jako kostra a sama plní jen minimální funkčnost. Až postupným přidáváním jednotlivých prvků, popsaných v sekci 4.4, se zvyšují možnosti této třídy. Část struktury této třídy je zca-

dlena v souborovém systému, přes který je možné k těmto součástem přistupovat. Samotný souborový systém a jeho funkce jsou popsány v kapitole 4.3.2.

Jelikož nedává smysl, aby v programu byla obsažena více jak jedna instance třídy `Device`, je tato třída navržena podle návrhového vzoru singleton. Při inicializaci programu je vytvořena první instance třídy. Při pokusu vytvořit novou instanci této třídy je ukazatel vždy vrácen na první již existující instanci třídy. Díky této možnosti přístupu k adrese je tento objekt dostupný z libovolné oblasti programu. A není tedy omezen rozsahem platnosti nebo jmenným prostorem. Díky tomu nemusí být exportován ukazatel na tento objekt do jednotlivých modulů.

Účelem třídy `Device` je popsat jak softwarové, tak hardwarové součásti zařízení. Tato struktura je inspirována strukturou Linuxového jádra, které využívá pro reprezentaci zařízení stromu souborového systému s využitím `sysfs` [37], [24]. Za účelem minimalizace velikosti výsledného programu je tak souborový systém jen volitelných doplňkem. Pro přístup a komunikaci se zařízením je použito zabudované rozhraní příkazové řádky.

### 4.3.1 Rozhraní příkazové řádky

Je volitelným rozšířením, které je možno připojit ke třídě `Device`. Může sloužit jak pro debugovací účely, tak pro ovládání při nasazení výrobku. Může být použito propojeno s libovolným komunikačním sériovým rozhraním, jako je například UART. Rozhraní příkazové řádky (dále označované jako CLI) může být propojeno s virtuálním souborovým systémem.

Terminál na zařízení, které se připojuje na příkazovou řádku zobrazuje ve většině případech jen přijaté znaky. Aby tedy rozhraní fungovalo jako příkazová řádky jsou znaky přijaté na straně mikrokontroléru přeposlány zpět. A vytváří dojem, že se přímo zobrazují při zadání na komunikačním zařízení. Nedochozí ovšem k přenosu všech znaků. Například znaky pro nový řádek, backspace a speciální sekvence nejsou odesílány zpět. Tyto znaky mají ovšem speciální funkci na straně mikrokontroléru.

Obsah aktuálního řádku je udržován na straně mikrokontroléru za účelem zpracování. Kromě toho může být prováděno mazání již napsaných znaků pomocí znaku DEL (ASCII hodnota 127). Při zadání znaku konce řádku, což je pro většinou sériových rozhraní znak CR (ASCII hodnota 13), dojde ke zpracování zadaného řádku. Předpokládá se, že jako první řetězec je zadán příkaz a poté následují parametry.

CLI umí zpracovávat příkazy, tyto příkazy mají definovány pomocí speciální třídy `Command`, obsahující `Invocation_wrapper` a text, který odpovídá názvu příkazu. Příkazová řádky uchovává seznam dostupných příkazů. Další příkazy je možné registrovat, pokud splňují podmínky, že jejich návratovým typem je `int` a jako argumenty přijímají vektor stringů. Ke každému příkazu může také náležet nápověda, která částečně nahrazuje program `man`. Základní dostupné příkazy umožňují pouze zjistit například datum kompilace programu, použitý kompilátor nebo verzi knihovny ALOHAL. Veškeré další příkazy jsou doplněny moduly připojitelnými k CLI jako je například souborový systém.

Každý řádek je uvozen posloupností znaků, stejně tak jako v linuxovém terminálu. Tato posloupnost je v základním nastavení jen symbol `>` obarvený pomocí escape sekvencí. Před tento uvozovací symbol může být přidán prefix, který se vztahuje k nějakému modulu příkazové řádky. Aktuálně je využit pouze souborovým systémem a zobrazuje aktuální pozici v souborovém systému.

Funkcionalita příkazové řádky může být dále rozšířena pomocí dalších doplňků, jako je například automatické doplňování, nebo historie použitých příkazů. Tyto doplňky nejsou ve výchozím stavu připojeny a to hlavně z důvodu paměťových omezení a možných vlivů

na výkonost systémů. Nicméně jedná se o doplňky, která práci s CLI výrazně zefektivňují a minimálně v době vývoje zařízení je vhodné je využít.

## Historie

Historie je řešena pomocí kruhového bufferu, do kterého se postupně přidávají použité příkazy a nejstarší jsou postupně přepisovány. Při každém odeslání symbolu nového řádku je aktuální řádek napřed uložen do historie a poté zpracován. Nejsou ukládány prázdné řádky. Při povolení tohoto doplňku příkazové řádky je nutné zvolit vhodný limit počtu udržovaných záznamů. To z důvodu, že všechny tyto použité příkazy jsou udržovány v RAM paměti mikrokontroléru. Tedy limit používaný na běžných systémech, což je většinou 1 000 příkazů, je zde nereálný. Doporučeným limitem je okolo 10 až 20 příkazů. Pro listování v historii příkazů je možné použít šipky na klávesnici. Tyto klávesy odesílají ANSI escape sekvence, které jsou nahrány a podle jejich významu provedena akce.

## Automatické doplňování

Automatické doplňování slouží pro urychlení práce při průchodu souborovým systémem. Je aktivován stejně jako na běžných zařízeních a to odesláním symbolu HT (neboli Horizontal Tab). Následně se vytvoří seznam dostupných kandidátů. Ten se skládá z názvů položek v aktuálně aktivní složce. Zkontroluje se, zda nějaký kandidát má prefix společný s obsahem řádku CLI. Pokud je takových kandidátů více jsou vypsány všechny možnosti. Pokud existuje jen jeden, tak je doplněn na vstup příkazové řádky. Nemá-li zařízení aktivní souborový systém, není tato funkčnost dostupná.

K problémům s výkonností může docházet na předpokladu, že originální seznam kandidátů je příliš velký. V takovém případě může hledání shodných prefixů zabrat relativně dlouhou dobu a mít za následek zpomalení celého systému. I proto je tento doplněk volitelný a ve výchozím stavu není aktivní.

### 4.3.2 Souborový systém

Jedná se pouze o virtuální souborový systém, který není reprezentovaný strukturami uloženými v programové paměti. Také se nejedná o perzistentní úložiště a celá struktura je při restartu sestavována programově, nicméně může se odkazovat na perzistentní data. Je volitelným doplňkem struktury zařízení. Slouží hlavně pro zpřístupnění funkčnosti jednotlivých prvků zařízení. Navazuje na funkčnost příkazové řádky zařízení a dále rozšiřuje možnosti manipulace, které může uživatel nebo programátor využívat během práce se zařízením.

Souborový systém může být propojen s rozhraním příkazové řádky. Při tomto propojení jsou registrovány nové příkazy pro CLI. Těmito příkazy jsou `ls`, `cd`, `pwd`, `cat`. Jejich funkčnost je obdobná jako v linuxových systémech, jen neobsahují tak široké možnosti přepínačů, kvůli úspoře paměti. Krom toho je přidána možnost spouštění objekty souborového systému z příkazové řádky, tento objekt musí být spustitelný. Dále je při každé změně aktuální pozice v souborovém systému změněn prefix, který CLI zobrazuje před uvozovacím znakem.

Základní obsah souborového systému využitá ve struktuře zařízení se skládá ze složek: `apps`, `components`, `tools`, `mcu`. Do těchto složek jsou poté přidávány dále popisované součásti zařízení popsané v kapitole 4.4.

Základním blokem souborového systému je záznam. Od tohoto záznamu se poté odvozují jednotlivé třídy jako je soubor, složka a spustitelný záznam. Každý záznam musí obsahovat

minimálně svůj název a ukazatel na složku, ve které je uložen. Záznamy jsou ukládány do klasického stromu, kdy kořenová složka má název `/`.

## Složka

Funkčnost není odlišná od běžného významu složky v souborovém systému. Umožňuje ukládat jiné záznamy, rekurzivně kontrolovat existenci záznamů s požadovaným jménem případně poskytovat obsah externím modulům jako je například CLI. V názvu mohou být libovolné znaky. V každé složce existují výchozí složky s názvy `.` a `..` se stejným významem jako v linuxovém systému. Tyto složky nejsou vypisovány při zadání příkazu `cd`, to jen z důvodu snížení zátěže pro komunikační kanál a zpřehlednění výstupu.

## Soubor

Jedná se pouze o textový soubor. Má celkem tři možné varianty, podle toho kde je jeho obsah uložen. Možnostmi jsou uložení v RAM nebo Flash paměti, případně varianta virtuálního souboru. U varianty RAM je obsah uložen v proměnné, soubor by tedy neměl být příliš velký, aby neomezoval běh programu. V případě Flash paměti je uložen na konci paměti určené pro program. Pro tento způsob uložení je ovšem nutný modul pro ukládání do Flash, který není zatím dostupný pro použitý mikrokontrolér.

Poslední variantou je virtuální soubor, který opět vychází z linuxového jádra. Obsah tohoto souboru je programově vytvořen až v době, kdy dojde k jeho čtení. Tento typ souboru obsahuje `Invocation_wrapper`, který musí vracet textovou nebo číselnou hodnotu. Číselná je jen převedena na textovou. Výhodou takového typu souboru je, že obsah může být aktuální, ale obsah souboru nemusí být periodicky aktualizován.

Příkladem virtuální varianty může být soubor obsahující teplotu jádra mikrokontroléru. Ve chvíli kdy je použit příkaz `cat` na tento soubor, je spuštěna metoda, která získá teplotu a vrátí ji jako číslo, které je poté převedeno na text a ten poté vypsán. Stejně jako by se jednalo o text uložený v paměti. Problémem ovšem může být prodleva při získávání aktuální hodnoty, proto by tato možnost měla být použita jen pro hodnoty, které je možné získat rychle, aby nebylo CLI blokováno.

## Spustitelný záznam

Tento typ záznamu reprezentuje uložený `Invocation_wrapper`. Tímto způsobem je tedy z příkazové řádky možné modifikovat chování některých částí systému, případně spouštět akce podle zadání uživatele. Podmínkou je, že metoda musí vracet typ `int` a přijít jako argumenty vektor stringů, stejně tak jako příkaz v CLI.

Příkladem takového spustitelného souboru může být odkaz na metodu `Reset` patřící objektu reprezentující mikrokontrolér, které provede jeho restart.

## 4.4 Součásti zařízení

Samotné součásti zařízení jsou rozděleny do různých skupin hlavně podle úrovně abstrakce a množství zahrnutých objektů. Každá z následujících skupin je založena na základní třídě, od které jsou odvozeny další třídy v této skupině. Díky tomu je zajištěno známé rozhraní pro celou skupinu. Tyto skupiny jsou spravovány strukturou zařízení.

#### 4.4.1 Mikrokontrolér

Jedná se o speciální součást celého zařízení a předpokládá se jen jedna instance třídy `mcu`, která je spojena se samotným zařízením, nicméně to není podmínkou. Tato třída se skládá z jednotlivých periférií mikrokontroléru. Je tedy tvořena třídami jako například UART, ADC, SPI. Umožňuje přístup k těmto zanořeným objektům a zpřístupňuje jejich používání.

Kromě zpřístupnění periférií přes programovou strukturu třídy je možné k některým částem přistupovat i přes rozhraní příkazové řádky, pokud je propojeno se souborovým systémem. Není možné ovládat periférie, jelikož to by mohlo být až destruktivní. Spíše je možné sledovat jejich stav. Například lze přes CLI zjistit aktuální teplotu mikrokontroléru nebo napájecí napětí určené podle vnitřní reference. Dostupné funkce mikrokontroléru jsou uloženy ve složce `/mcu/`.

#### Konfigurace

Při kompilaci se předává definice určující o jaký mikrokontrolér se jedná. Toto nastavení je využito při kompilaci knihovny HAL ale i konfiguraci samotné třídy `mcu`. Na základě tohoto nastavení jsou například nadefinovány adresy, kde jsou uložena kalibrační data teplotního senzoru nebo přizpůsobení napěťové reference. Tento způsob nastavení vychází z konfigurace prováděné pomocí `device tree` v rámci Linuxového jádra [24].

#### 4.4.2 Komponenty

Mezi hardwarové součásti třídy `Device` mohou být například jednotlivé součástky osazené na desce zařízení, které jsou relevantní pro jeho chod a interagují s řídicím programem zařízení, tedy téměř vždy s mikrokontrolérem. Touto interakcí může být například jen reakce na změnu logické úrovně na výstupu mikrokontroléru, což může být například signalizační LED. Odlišnou variantou může být integrovaný obvod obsahující vlastní paměť a vstupy komunikující s mikrokontrolérem přes libovolný komunikační kanál, například SPI sběrnici.

Pro programový popis takovýchto součástek slouží třída `Component`. Jakmile je vytvářena třída, která by měla popisovat součástku nebo obvod interagující s programem zařízení, tak tato nová třída by měla dědit vlastnosti třídy `Component`. V rámci této dědičnosti musí být známo jméno komponenty, díky kterému jej lze následně rozeznat ve struktuře zařízení. Jelikož by ale při vytvoření více instancí jedné součástky vznikaly duplicity, je pro identifikaci objektu k názvu získanému ze třídy přidáno číslo. V rámci konstruktoru komponenty je zaregistrována pod zařízením.

Například senzor BME280 je popsán vlastní třídou, která obsahuje název součástky `BME280` a je odvozen od třídy `Component`. Při vytváření objektu podle této třídy je nalezeno první volné identifikační číslo a přidáno k názvu součástky. Za předpokladu, že by se jednalo o první takový objekt ve struktuře zařízení, tak následně by bylo možné jej dohledat ve struktuře pod označením `BME280_#1`. Přidělování čísel spravuje třída `Device`. Kromě textového označení obsahuje třída také unikátní číselný identifikátor, který je použit pro logovací výpisy.

V rámci souborového systému jsou všechny komponenty reprezentovány jako složky uvnitř `/components/`. Název složky popisující komponentu je jméno komponenty včetně číselného označení. Následně každá komponenta může vytvářet virtuální soubory nebo spustitelné záznamy. Všechny tyto záznamy souborového systému jsou uvnitř složky odpovídající komponentě. Soubory obsahující přímo text nejsou povoleny, jelikož by tak mohlo dojít



k velmi rychlému zaplnění RAM v případě, že by bylo vytvořeno více instancí komponenty k odpovídajícím obvodům.

### 4.4.3 Nástroje

Mezi softwarové součásti zařízení mohou patřit různé softwarové nástroje. Ty jsou derivovány od třídy `Tools`. Jedná se většinou o rozšíření funkčnosti komponent nebo mikrokontroléru. Opět jsou zpravovány pomocí třídy `Device` a v souborovém systému jsou uloženy ve složce `/tools/`. V rámci hierarchie se jedná o část využívající komponenty a mikrokontrolér. Často je jejich účelem jen zefektivnění vývoje, případně sledování stavu systému.

Mezi jeden z nástrojů patří například průzkumník pro sběrnici I2C, který odešle první paket obsahující adresu zařízení a čekání na odpověď ve formě acknowledgment bitu. Takto postupně zkontroluje všechny adresy, tedy 127 zařízení a zjistí, která adresy na sběrnici jsou obsazeny, na základě toho potom na výstup CLI vypíše zprávu. Následuje popis několika dalších nástrojů, které byly využity v průběhu vývoje zařízení.

#### Tasker

Mezi tyto nástroje patří například `Tasker`, který zajišťuje plánování málo frekventovaných událostí. Hlavním důvodem jeho využití je šetření časovačů MCU. Může být tedy využit pro události, ke kterým dochází jen několikrát za vteřinu nebo méně často, případně pomocí něj mohou být události provedeny jen jednou ovšem s odložením. Bez využití `Tasker` by musel být použit časovač, kterých je omezený počet, případně ani nemusí mít dostatečnou kapacitu čítače nebo by muselo být použito aktivní čekání. Samotný `Tasker` využívá jen jeden časovač, u kterého automaticky nastavuje čítač a předděličku podle toho jak vzdálená je další událost v jeho kalendáři akcí. Události obsahují čas, počet opakování a `Invocation_wrapper`, který bude v příslušnou dobu aktivován. Velmi snad tak lze vyřešit situace kdy je například rozsvícena LED v reakci na příchod zprávy a má být zhasnuta za 200 ms a to bez opakování. `Tasker` není možné ovládat přímo z rozhraní příkazové řádky, jelikož by tím mohlo dojít k narušení časování důležitých systémů.

#### Plánovače

Jedná se o nástroj sloužící pro synchronizaci většího počtu současných dějů. Posloupnost těchto dějů lze rozdělit do fází. Každá fáze se skládá ze vstupních podmínek a výstupních efektů. Při vytvoření objektu plánovače je definován počet fází aktuálního plánu. Každé fázi se následně přiřadí určitý počet vstupních podmínek, označovaných jako spouště. Ty fungují jako metoda s argumentem, kterým je číslo. Prvek, fungující jako spoušť, zavolá metodu s přiděleným číslem. Tímto prvkem může být například přerušení, kterému je zaregistrována lambda funkce volající metodu plánovače s číslem spouště. Jakmile jsou aktivovány všechny spouště dané fáze provedou se výstupní efekty. Jedná se opět vektor obsahující `Invocation_wrapper`. Tímto výstupním efektem může být například změna logické úrovně na výstupu mikrokontroléru. Spolu s provedením výstupních efektů se aktivuje další fáze, která opět čeká na aktivaci všech spouští. Toto je prováděno dokud v plánu existují nedokončené fáze.

Jednoduchý příkladem může být plán skládající se ze dvou fází. První fáze má dvě spouště a to jsou externí přerušení ze dvou tlačítek, výstupem fáze je aktivaci motoru na výstupu. Druhá fáze má je jednu spoušť a ta je aktivována jakmile čidlo otáček zjistí, že

motor dosáhl požadovaných otáček. Výstupním efektem druhé fáze je odeslání zprávy přes komunikační rozhraní, že otáček bylo dosaženo.

Díky tomuto plánovači lze dále rozšířit koncept oddělených systémů, který je zaveden přerušeními. Plánovače také umožňují synchronizovat nezávislé události a jejich efekty.

#### 4.4.4 Aplikace

Jedná se o nejvyšší formu zapouzdření využívající většinou všechny vrstvy pod ní. Jedná se o součást zařízení, která většinou řídí hlavní funkčnost zařízení. Řídí komplexnější chování zařízení a většinou uživatel komunikuje jen s ní. Vytvářené aplikace jsou opět odvozeny od třídy `Application` a jsou uloženy v souborovém systému ve složce `/apps/`. Samotné zařízení jich může obsahovat více, počet není omezen.

Rozdíl mezi nástroji a aplikacemi je v předpokládaném použití. Nástroje slouží spíše jen jako pomůcka usnadňující vývoj. Oproti aplikaci je nástroj výrazně univerzálnější. Aplikace plní samotný účel zařízení. Nástroje by také měly sloužit opakovaně a být použité ve více instancích zařízení, ovšem aplikace by měla sloužit pro jedno konkrétní zařízení. Vychází tedy z knihovny `ALOHAL`, ovšem její finální varianta by neměla být v knihovně zahrnuta.

Třída `Application` poskytuje jen rozhraní pro derivované třídy. Toto rozhraní obsahuje metody pro inicializaci a spuštění aplikace. Během inicializace se provede nastavení používaných komponent a nástrojů, v rámci čehož je možné i konfigurovat MCU a jeho přerušeni. Metoda pro spuštění aplikace má stejné rozhraní jako příkazy pro příkazovou řádku, vrací `int` a přijímá vector stringů, obdobně jako běžný program napsaný v C nebo C++. Každá aplikace také musí mít jméno, která je v unikátní v rámci všech aplikací v zařízení.

## 4.5 Komunikační protokol

Komunikace je obstarávána aplikací v zařízení. Konkrétně se jedná o aplikaci s názvem `Eyrina_control`. Ta přijímá příkazy, které slouží k ovládání pohybu, světla a ostatních prvků. Návrh komunikačního protokolu je navržen na základě numerického programovacího jazyka G-code. Jedná se o seznam příkazů jdoucích lineárně za sebou které jsou použity pro naprogramování dráhy CNC zařízení. V protokolu neexistují žádné složitější konstrukce jako jsou proměnné nebo skoky, jedná se jen o sled příkazů, které jsou interpretovány zařízením [29].

Pro ovládání jsou využity již existující příkazy používané pro lineární pohyb [44]. Rozšířené o pohyb po rotačních osách a ovládání specializovaných prvků zařízení jako je ovládání záměrného bodu a osvětlení.

Samotný příkaz se skládá z kódu příkazu, který definuje o jakou činnost se jedná a parametrů příkazu. Kód je složen z písmena a následně až tři číslic. Písmeno určuje třídu příkazu, tím může být lineární pohyb (G, M), příkaz specifický pro typ zařízení (E) nebo příkaz pro okamžité provedení (R). Následující číslice určují přesný typ příkazu, například jestli se jedná o pohyb v relativních nebo absolutní koordinátech. Za kódem příkazu následují mezerou oddělené parametry příkazu. Každý parametr se skládá z jednoho písmene, které označuje o jaký parametr se jedná. Za označením parametru se nachází číselná hodnota, kterou může být i desetinné číslo, pokud jej příkaz podporuje. Parametr ovšem může být i binární a tedy sloužit jen jako příznak, v tom případě se neočekává žádná hodnota ale ihned následuje oddělovací mezera.

### 4.5.1 Ovládací aplikace

Aplikace `Eyrina_control` přijímá jednotlivé příkazy a následně je interpretuje. Příkazy je možné předávat aplikaci jako argumenty pro její spuštění. V tomto případě je možné zadat jen jeden příkaz a poté je nutné počkat než bude vykonán. Případně lze využít možnost, kdy samotná aplikace přebere řízení CLI a je možné zasílat dávkové příkazy, které jsou odděleny odřádkováním. Varianta, kdy je CLI řízení aplikací, je určena pro praktické využití zařízení, kdy se předpokládá, že uživatel nepotřebuje přístup ke struktuře zařízení a nepotřebuje pracovat se třídou `Device`.

Jelikož vykonání příkazu trvá v některých případech až několik vteřin, zvláště u příkazů pro pohyb, tak je nutné zajistit relativně pomalý tok příkazů. Vstupní buffer pro příkazy je omezen jen na 50 příkazů, pak jsou příkazy zahazovány. Jedná se o poměrně přísné omezení nicméně množství příkazů najednou není většinou tak velké, aby byl tento limit naplněn. Omezení je zavedeno z důvodu paměťového omezení a zajištění stability systému. Tento problém by bylo možné vyřešit přidáním externí paměti, kterou by mohla být například EEPROM nebo Flash paměť, do které by se ukládaly příkazy, které není možné vložit do bufferu uvnitř paměti RAM.

Aplikace kontroluje syntaxi příkazů. V případě, že je nalezena chyba v příkazu, tak je celý zahozen. Informace o tom, že příkaz je vyřazen je vypsána na výstup. Jako reakce na tento případ existují tři možnosti. První je jen ignorování tohoto příkazu a následně je možné pokračovat dále, toto je potenciálně nebezpečné, protože tím mohou být porušeny určité předpoklady pro ostatní příkazy. Může například dojít ke kolizi částí zařízení a i případnému poškození. Druhou variantou je pozastavení vykonávání příkazů dokud není odeslán speciální příkaz. Tento příkaz odblokuje frontu příkazů a jejich vykonávání může okamžitě pokračovat. Jedná se speciální příkaz patřící do zvláštní skupiny, příkazy z této skupiny nejsou zařazovány do fronty, ale jsou vykonány okamžitě. Tato možnost předpokládá, že před zadáním odblokovávacího příkazu byl problém s chybným příkazem vyřešen. To tím že byla vyčištěna fronta příkazů a zadány nové, nebo že bylo zkontrolováno a zajištěno, aby ke kolizi nedošlo a chybějící příkaz nezpůsobil problémy. Třetí možností je naprosté zablokování ovládací aplikace, která následně zabráni zadání jakýchkoliv příkazů. Jedná se o nejbezpečnější variantu, protože vyžaduje vyřešení problému a nelze ji obejít. Jediným řešením tohoto zablokování je totiž restartování celého zařízení a odeslání nových příkazů. Výběr v jakém módu bude aplikace pracovat je definován během kompilace programu a nelze změnit jinak než přeprogramováním zařízení. To z důvodu, že třetí mód je určen pro nasazení a první dva pro vývoj a testování.

Níže jsou popsány dostupné příkazy. U každého z parametrů je popsán jeho typ. Možné typy jsou `<b>`, jedná se pouze o přepínač, který je a nebo není uveden, nemůže s ním být předána žádná jiná hodnota. Další možností je `<i>` jedná se o celočíselnou hodnotu, rozsah je omezen jen vlastnostmi samotného příkazu. Poslední variantou typu je `<f>`, což je desetinná hodnota, přesnost tohoto čísla není omezena, ovšem téměř všechny příkazy počítají s maximálně 6 desetinnými místy.

### 4.5.2 Příkazy pro pohyb

Tato sada příkazů slouží k ovládní pohybu. Pro každou osu je udržována aktuální pozice, na které se nachází. Tato pozice je vztažena k nulovému bodu osy. Nulová pozice je definována jako pozice nejbližší ke koncovému spínači, kdy ještě spínač není spuštěn. Najetí do nulové pozice je aktivováno příkazem `G28`. Pro všechny příkazy funguje ochrana koncových bodů.

Tedy pokud je nastavena pozice větší než maximální délka osy, tak pohyb je prováděn jen na konec této osy. Ochrana je aktivní pouze pokud je známa aktuální absolutní pozice.

### G0 - Absolutní posun

Příkaz provede přesun jednotlivých os na definované pozice. Pozice je definována vzdáleností od nulového bodu. Minimálně jedna z os musí být zadána. Pro pohyb je použita maximální dostupná rychlost všech os. Před použitím tohoto příkazu musí být zjištěna absolutní pozice, musí tedy být provedeno zajetí na nulové koordináty pomocí příkazu G28. Číslo musí být větší než nula, jinak je příkaz považován za chybný.

$$G0 \ X<f> \ Y<f> \ Z<f> \ F<f> \ T<f> \ R<f> \quad (4.1)$$

parametry:  $X, Y, Z$  – Posun po zadaných osách [mm]  
 $F$  – Posun posun ostříčího okuláru [mm]  
 $T, R$  – Rotace okolo zadaných os [°]

### G1 - Relativní posun

Příkaz provede posun jednotlivých os o definovanou vzdálenost od aktuální pozice. Pro tento příkaz nemusí být předtím provede příkaz G28. Nicméně nemohou být poté kontrolovány hranice pohyby. Tedy tedy důležité věnovat zvýšenou pozornost takovému pohybu nebo provedením příkazu G28 aktivovat ochranu pro přejetí přes konec osy. Velikost pohybu může být i záporné číslo, tím se provádí pohyb směrem ke koncovému spínači.

$$G1 \ X<f> \ Y<f> \ Z<f> \ F<f> \ T<f> \ R<f> \quad (4.2)$$

parametry:  $X, Y, Z$  – Posun po zadaných osách [mm]  
 $F$  – Posun posun ostříčího okuláru [mm]  
 $T, R$  – Rotace okolo zadaných os [°]

### G28 - Přesun na nulové souřadnice

Zahájí pohyb definovaných os na jejich domovské pozice. Může být vybrána může být libovolná kombinace os. Za předpokladu, že není žádná osa vybrána tak dojde automaticky k pohybu všech. Po tomto příkazu je aktivní ochrana hraničních pozic, proto je tento příkaz doporučeno provést jako první.

$$G28 \ X<b> \ Y<b> \ Z<b> \ F<b> \ T<b> \ R<b> \quad (4.3)$$

parametry:  $X, Y, Z, F, T, R$  – Označení os, na kterých bude prováděn přesun

### M0 - Zastavení

V případě, že aktuálně probíhá nějaký pohyb, tak je zastaven. Tento příkaz se vykonává okamžitě jakmile přijde na řadu, nečeká se na dokončení předchozího příkazu. Nicméně tento příkaz nepředbíhá frontu čekajících příkazů. Zpomalení není okamžité, je provedeno včetně fáze postupně decelerace rychlosti.

$$M0 \ X<b> \ Y<b> \ Z<b> \ F<b> \ T<b> \ R<b> \quad (4.4)$$

parametry:  $X, Y, Z, F, T, R$  – Označení os, na kterých bude zastaven pohyb

### M10 - Uvolnění motorů

Provede odpojení napájení od motorů, které je do této chvíli drženy v pozici magnetickým polem, které je vytvářeno cívkami. Za předpokladu, že je některá z os v pohybu, tak dojde ke stejné akci, jako by byl proveden příkaz M0. Problém s tímto příkazem je potenciální ztráta aktuální pozice, jelikož následně je možné manuálně pohnout některou z os. Nicméně samovolný pohyb kvůli tuhosti os není pravděpodobný. Nejistota pozice vzniklá ztrátou magnetického pole motoru odpovídá vzdálenosti, která je způsobena pohybem o jeden celý krok. To platí pokud nebylo manuálně pohnuto osou, pak je pozice ztracena naprosto. Tento příkaz je vhodné zadat, pokud má zařízení dlouhou dobu stát v klidu, výrazně se tím sníží odběr zařízení a teplo vyzařované budiči krokových motorů. Tento příkaz je také vykonán okamžitě jakmile přijde na řadu a nečeká na dokončení předchozího.

$$M10 \ X<b> \ Y<b> \ Z<b> \ F<b> \ T<b> \ R<b> \quad (4.5)$$

parametry:  $X, Y, Z, F, T, R$  – Označení os, na kterých budou uvolněny motory

### 4.5.3 Příkazy osvětlení a záměrný bod

Příkazy speciálně určené jen pro zařízení EYRINA v4. Jsou vykonávány v běžném pořadí jako příkazy jejichž kód začíná písmenem G. Jsou ukládány do fronty příkazů.

#### E1 - Nastavení intenzity světla

Nastaví intenzitu výstupního světelného kanálu na definovanou hodnotu. V jednom příkazu je možné nastavit jen jeden výstupní kanál. Index kanálu musí být v rozsahu 1 až 3. Intenzita kanálu může být desetinné číslo v rozsahu 0–100. Intenzita je udávána v procentech. Vypnutí je provedeno nastavením intenzity na 0 %.

$$E1 \ C<i> \ I<f> \quad (4.6)$$

parametry:  $C$  – Index výstupního světelného kanálu (1-3)

$I$  – Výstupní intenzita (0-100 %)

#### E2 - Aktivace/Deaktivace záměrného bodu

Provede rozsvícení nebo zhasnutí záměrného bodu na aktuální pozici. Výchozí pozice záměrného bodu po zapnutí zařízení je [3, 3].

$$E2 \quad (4.7)$$

### E3 - Nastavení pozice záměrného bodu

Pro záměrný bod je předem definováno 25 pozic. Tyto pozice mají tvar mřížky, kde střed odpovídá nulovému nahnutí fundus kamery. Posun bodu o jednu pozici ve vybraném směru odpovídá nahnutí platformy o 5°. Index pozice záměrného bodu určuje pozici v mřížce. Index [3,3] odpovídají nulovému náklonu, tedy středu mřížky.

$$\text{E3 } X < i > \quad Y < i > \quad (4.8)$$

parametry:  $X$  – Index sloupce mřížky  
 $Y$  – Index řádku mřížky

#### 4.5.4 Ostatní příkazy

Příkazy v této sekci slouží pro vyřízení kritických událostí, ke kterým by nemělo běžně dojít. Všechny příkazy jejichž kód začíná písmenem R předbíhají celou frontu a jsou vykonány okamžitě, jakmile jsou přijaty aplikací a to i za předpokladu, že nějaký příkaz probíhá.

#### R0 - Nouzové zastavení

Provede okamžité zastavení na všech osách. Zastavení je provedeno bez zpomalení. Pohyb je zastaven velmi prudce, proto je tento příkaz dobře používat jen v nouzových situacích. Pro případ běžného zastavení je vhodnější použít příkaz M0, případně M10. Bez zadaného příznaku M provede i vypnutí záměrného bodu a nastaví intenzitu všech světelných kanálů na nulovou hodnotu.

$$\text{R0 } M < b > \quad (4.9)$$

parametry:  $M$  – Pokud je tento příznak aktivní dojde jen k zastavení pohybu

#### R10 - Odblokování fronty

Odblokuje frontu příkazů, pokud došlo k její blokaci vlivem chybné syntaxe předchozího příkazu. Následně se začne okamžitě vykonávat první příkaz ve frontě. Je nutné zajistit bezpečné vykonání následujících příkazů. Funguje pouze v případě, že aplikace běží v daném bezpečnostním módu.

$$\text{R10} \quad (4.10)$$

## 4.6 Logování

Pro logování stavu existuje speciální třída `Logger`. Jedná se opět o návrhový vzor singleton. A všechny požadavky pro zápis logovacího záznamu jsou řešeny touto jednou třídou. Aby bylo možné z libovolného objektu ukládat záznamy, existuje další třída `Loggable`. Jedná se o třídu, která slouží jako rodičovská pro třídy, ze kterých je možné ukládat záznamy. V rámci dědičnosti získá derivovaná třídy přístup k metodám, které ukládají logovací záznamy. Návrh tohoto systému je inspirován systémem knihovny POCO [45].

Díky tomuto principu dědičnosti v rámci třídy `Loggable` je možné v derivované třídě do uloženého záznamu přidat název třídy, která záznam vytvořila spolu s adresou. Díky

oddělení třídy, která provádí samotný zápis zpráv lze tak jednoduše nastavit globální chování při ukládání záznamu.

Lze například nastavit úroveň vážnosti, která bude zapisována. Existuje celkem 5 úrovní rozdělených vážností záznamu. Tyto úrovně jsou: **Fatal**, **Error**, **Warning**, **Notice**, **Debug**. Pro **Logger** existuje nastavená úroveň, která určuje, že zprávy s úrovní nižší než je nastavená budou zahozeny. Díky tomu lze jednoduše omezit počet zpráv v záznamu při nasazení zařízení bez toho, aby musely být z kódu odmazány tyto řádky. Při vhodném využití `const expr` se lze vyhnout i výpočetními využití během zápisu takových zpráv, pokud je hranice nastavená pro logování konstantní.

## 4.7 Zobrazení stavu zařízení

Aktuální stav zařízení je symbolizován RGB diodou, která je umístěna na základní desce. Proud jednotlivými prvky diody je řízen integrovaným obvodem WS2818B [11].

Nejzajímavější částí implementace, která je spojena s tímto obvodem je komunikační protokol. To je vysokofrekvenční PWM signálem, u kterého posloupnost stříd jednotlivých period signálu určuje intenzitu jednotlivých barev. Pro každou barevnou složku slouží 8 period signálu, tedy celkem 24 period signálu, kde každá složka má 256 úrovní jasu. Tento protokol má komunikační frekvenci 1 MHz. A žádná periferie není přímo uzpůsobena pro tento protokol. Programově není možné tak rychlý protokol řídit z mikrokontroléru s frekvencí 80 MHz, za předpokladu, že mají být souběžně prováděny další časově kritické operace.

Ke komunikaci je tedy využita kombinace 16-bitového časovače v mu **OutputCompare** a DMA přenosu. Časovač v módu **OutputCompare** funguje tak, že ve chvíli kdy čítací registr dosáhne hodnoty v komparačním registru provede invertování logické úrovně na výstupním pinu, který je spojen s časovačem. Časovač dále pokračuje v čítání, dokud opět nemá shodu s komparačním registrem, to může nastat ale až po přetečení čítače.

DMA přenos je nastaven tak, aby při aktivaci přenesl 16-bitové číslo z paměti do 16-bitového komparačního registru časovače. Při každém přenosu je adresa zdroje inkrementována o 2, takže nová adresa ukazuje na další číslo. V paměti na adrese, kde přenos začíná jsou předpočítány časy tak, aby odpovídali požadovanému přenosu. Určují tedy délky jednotlivých logických úrovní, přičemž se kumulativně zvyšují, to aby nemusel být časovač po každé shodě nulován. To by vedlo na nutnost programově zasáhnout. Pro 24 period PWM signálu je tedy určeno 48 hodnot. Předdělička časovače je nastavena na hodnotu 0, aby bylo dosaženo maximální přesnosti.

Časovač při shodě s komparačním registrem provede kromě invertování výstupu také aktivaci DMA přenosu, čímž dojde k přenosu nové hodnoty do komparačního registru. Poslední hodnotou, sloužící jako zarážka na hodnota 1, aby před další změnou logické úrovně muselo napřed dojít k přerušení generovanému přetečením registru. Přerušení ovšem obsahuje registrovanou lambda funkci, která odpojí časovač od zdroje hodinového signálu a tím jej zastaví. Díky tomu nedojde k aktivaci dalšího DMA přenosu, který by již načítal data z neinicializované oblasti paměti.

Výhodou tohoto přístupu je vysoká přesnost časování, která není závislá na vytížení mikrokontroléru a využívá jen minimum zdrojů. Jakmile jsou předpočítány časy, stačí jen aktivovat DMA s časovačem a vše další je provedeno bez programového zásahu. Jedná se o efektivní způsob jak implementovat libovolný digitální vysokorychlostní protokol. Maximální spolehlivá frekvence takto generovaného signálu je okolo 8 MHz na použitém mikrokontroléru. S vyšší frekvencí není již časování natolik přesné.



## 4.8 Kompilace a nahrání do mikrokontroléru

Pro sestavení celého programu je využit Makefile. Jelikož knihovna je přímo zabudována do programu, je předkládána souběžně se zbytkem programu. Překlad je rozdělen na dvě části, jednou je knihovna HAL, napsaná v jazyce C a je kompilována pomocí `gcc`, druhou je knihovna ALOHAL, která je spolu s hlavním programem kompilována pomocí `g++`. Aby nedocházelo ke kompilaci celé knihovny, ale jen částí nutných pro běh programu je s pomocí překladače vytvořen seznam závislostí. Kromě zdrojových souborů je přidán ještě Linker script, definující paměťové oblasti a programové sekce. Dalším souborem Start-up script, jedná o kód v assembleru, který inicializuje procesor a následně spouští hlavní program.

Jakmile je dokončen překlad jednotlivých součástí, tak jsou objektové soubory slinkovány. Při linkování je využita optimalizace LTO (Link-Time Optimization). Ta umožňuje výrazně snížit velikost výsledného binárního souboru. U tohoto projektu se tato optimalizace projeví zmenšením programu až o 20 %.

Pro kompilaci slouží výchozí cíl Makefile souboru a to cíl `make all`. Tento cíl spouští kompilaci na všech dostupných jádrech. Pro naprogramování mikrokontroléru slouží cíl `make flash`. Mikrokontrolér musí být připojen přes programování rozhraní na desce k programátoru St-Link v2. Pro naprogramování mikrokontroléru existují dvě varianty jedna využívá program `STM32CubeProgrammer` poskytovaný firmou STMicroelectronic, konkrétně rozhraní jeho příkazové řádky pro programování. Druhou variantou je open source program `OpenOCD` (Open On Chip Debugger). Obě varianty jsou uvedeny v Makefile souboru, výchozí je varianta `OpenOCD`. Minimální verzi `OpenOCD`, která je schopná nahrát program do mikrokontroléru na desce je verze 0.9.0, která přidává podporu pro čipy z řady STM32L45x.

Pro kompilaci je využit GNU Arm Embedded Toolchain. Minimální verze potřebná k úspěšné kompilaci je 2019q4. Předchozí verze mají problém se sestavováním virtuálních tabulek pro třídy se specializovanou šablonou. Nicméně GDB dodávané v rámci tohoto toolchain nepodporuje skriptování v jazyce Python. Z toho důvodu je nutné pro přehledné debugování programu zkompilovat GDB samostatně a povolit v něm skriptování v jazyce Python.

# Kapitola 5

## Závěr

Výsledkem práce je zařízení splňující a v určitých oblastech i převyšující zadané parametry. Zařízení je připraveno hlavně pro provádění experimentů, podle kterých bude následně provedena specifikace následující verze, jež by měla být vyráběna v počtu více kusů a tedy ne jen prototyp. Tato následující zařízení by měla být použita v ordinacích oftalmologických lékařů. Jejich testování v praktických podmínkách by mělo následně poskytnout data pro další optimalizace a vylepšení zařízení. Ohledně výroby a financování dalšího vývoje se bude jednat nejspíše s firmou Siemens, či dalšími partnery.

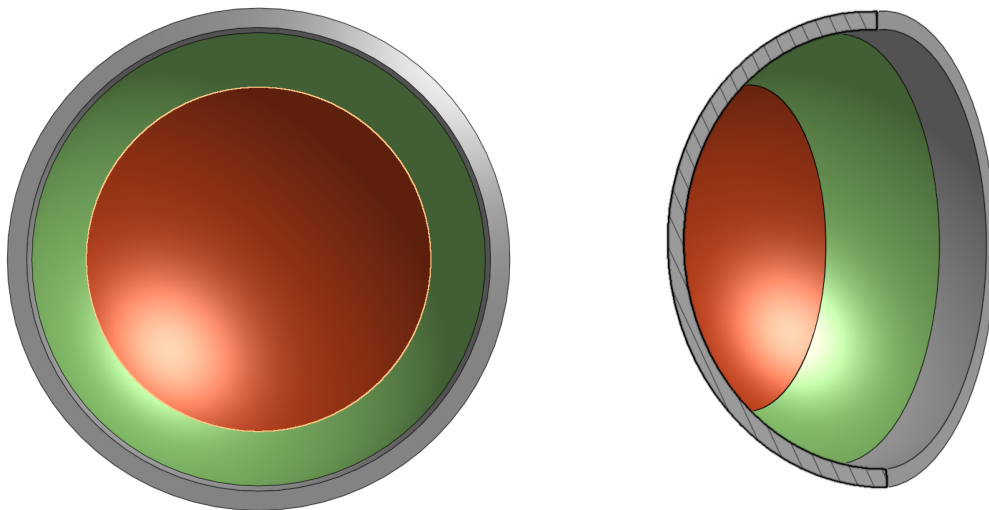
Největší část práce byla odvedena na základní desce zařízení. V rámci výroby této desky byla navržena kompletní specifikace všech jejích součástí. Následně proveden návrh schématu a design plošného spoje. Tento plošný spoj byl vyroben, osazen a oživen. Součástí zařízení je firmware mikrokontroléru umožňující ovládání všech motorů, světel a záměrného bodu. Toto ovládání je možné provádět na základě navrženého a implementovaného protokolu založeného na ovládacím jazyce G-code.

Na základě parametrů snímacího zařízení byla určena plocha, kterou lze pomocí naklonění nasnímat jako více než dvojnásobná oproti standardnímu snímku. Vizualizace snímané plochy je na obrázku 5.1. Tato vizualizace představuje průřez okem a barevně vyznačuje oblasti, které je možné nasnímat za použití různých postupů. Červená plocha znázorňuje plochu sítnice, která je získána bez nahnutí, tedy snímek, jež je poskytován většinou komerčních fundus kamer. Tento snímek pokrývá oblast o ploše 299 mm<sup>2</sup>. Zelená plocha znázorňuje oblast sítnice, kterou lze navíc získat při snímání, kdy dojde k naklonění fundus kamery. Plocha této rozšířené oblasti je 416 mm<sup>2</sup>. Celková plocha je tedy téměř o 140 % větší než plocha původní. Šedá plocha je nenasnímatelná oblast sítnice. Tato vizualizace počítá s průměrem vnitřní části oka 24 mm. Dále jsou využity parametry manipulátoru a kamery, tedy zorný úhel kamery 48° a naklonění manipulátoru o 15° vůči ose oka. Rozšířená plocha by byla získána poskládáním obrazu z více snímků získaných při natočení kamery do různých směrů.

### 5.1 Výsledná přesnost zařízení

V porovnání s požadovanými parametry uvedenými v sekci 3.2 jsou výsledné parametry výrazně lepší. Výsledné přesnosti bylo dosaženo pomocí velmi přesného manipulátoru v kombinaci s budičí, umožňující velmi přesné řízení.

Požadovaná přesnost na osách X a Z byla 50 μm. Teoretická přesnost zařízení pro osu X je 19,5 nm a pro osu Z je 9,76 nm. Pro osu Y byla minimální přesnost pohybu stanovena



(a) Snímaná plocha z čelního pohledu

(b) Průřez snímanou plochou

Obrázek 5.1: Vizualizace snímané plochy

na  $100\ \mu\text{m}$ . Výsledná přesnost by měla být stejná jako u osy X, tedy  $19,5\ \text{nm}$ . Teoretická přesnost osy určené pro ostření kamery je ve výsledku  $78,1\ \text{nm}$ , což je více než dostatečné oproti požadovanému kroku  $20\ \mu\text{m}$ .

Nejistota způsobená mechanickými odchytkami a volností os by měla dosahovat maximálně 3 % z uvedené teoretické přesnosti. Nejistota způsobená řízením motoru by měla dosahovat maximální 5 % z teoretické přesnosti, typická nejistota by měla být maximálně okolo 2 %.

Pro ověření zda zařízení má skutečně tak vysokou přesnost, by bylo nutné využít optické měření vzdálenosti například pomocí zařízení na konfokálním principu, a to z důvodu, že mechanické měřicí nástroje nemají dostatečné rozlišení pro tak jemné měření. Takovéto zařízení není ovšem pro ověření přesnosti k dispozici a takovou přesnost není ani nutné ve výsledném zařízení dodržet. Pro měření byl nakonec použit odchytkoměr, který ovšem umožňuje přesně měření jen v desítkách  $\mu\text{m}$ . Ovšem výsledky by měly být dostatečné k tomu, aby bylo potvrzeno, zda bylo dosaženo požadovaných přesností.

Měření bylo prováděno tak, že měřicí přístroj byl upevněn ke konstrukci manipulačního zařízení. Následně se osa pohybovala tak, aby vždy skončila ve stejné pozici, čím byla měřena i opakovatelnost pohybů. Z výsledných čísel byl vypočten rozsah a směrodatná odchylnka. Bylo prováděno deset měření pro každou osu.

Výsledkem měření ovšem byla jen nulová čísla, jelikož zařízení má výrazně vyšší přesnost, než jaká je měřitelná použitím měřidlem. Při měření byla vždy získána stejná čísla při všech z deseti pokusů. Toto se opakovalo u všech os. Lze tedy prohlásit, že přesnost zařízení je lepší než  $10\ \mu\text{m}$  a je téměř jistě i vyšší jak  $5\ \mu\text{m}$ . Výsledná změřená přesnost je tedy více než dostatečná pro praktické použití zařízení. Díky vysoké přesnosti se dá snadněji experimentovat s procesem získání snímků a ověřit jakou přesnost by měla mít další verze zařízení, za předpokladu že by se vyrábělo.

## 5.2 Výsledná spolehlivost zařízení

Na základě pokusů, které simulovaly předpokládané vytížení zařízení, se ukázalo, že regulace teploty a napájecí zdroje byly výrazně nadimenzovány. Při plném využití zařízení jsou napájecí zdroje určeny pro silovou část základní desky na zhruba 45 % jejich výkonnostní kapacity. Díky tomu je jejich ohřívání jen minimální a projeví se až při delším použití. Maximální měřená teplota zdrojů byla při těchto pokusech jen 30 °C nad okolní teplotou. Nebylo tedy nutné chlazení pomocí ventilátorů.

Chlazení budičů krokových motorů bylo také dostatečné i bez ventilátorů. Chladič budičů dosáhl maximální teploty 40 °C nad okolní teplotu. Nicméně většina tepla je vyzářena skrz chladičí plochu na spodní straně čipu, čili jako chladič slouží zemnicí plocha. V případě, že by chlazení nebylo dostatečné, je možné připevnit dvojici ventilátorů o průměru 80 mm přímo na ochranný kryt desky. Nikdy během testů nedošlo k aktivaci tepelné pojistky uvnitř budičů.

Samotný postup testů spočíval v přesunu manipulátoru na pozice, které byly odhadnuty jako vhodné pro snímání sítnice. Jednalo se tedy o devět pozic pro každé snímání. Postupně bylo provedeno 50 snímání. Mezi každým snímáním nastala pauza 30 vteřin. Mezi jednotlivými sekvencemi testu nebyl prováděn přesun na domovskou pozici. Zařízeno splnilo tento test úspěšně, celková délka testu byla okolo jedné hodiny.

Při těchto testech byl vypínán proud protékající motory, během stavu nečinnosti zařízení. Což má za následek, že části systému generující nejvíce tepla měli možnost zchladnout mezi sekvencemi pohybů. Byla ovšem ztracena absolutní přesnost motorů, proto by bylo nutné před každým dalším pohybem provést přesun všech os na domovskou pozici. Tento přesun může trvat i několik desítek vteřin za předpokladu, že všechny osy jsou maximálně vzdáleny, což by mohlo zpomalit celkové zacházení se zařízením. Je možné, že v praktickém využití by byla tato doba příliš dlouhá a proud protékající motory by se vypínal až po několika minutách, případně desítkách minut nečinnosti. To by mělo za následek větší zahřívání součástí. Nicméně, aby bylo možné otestovat teplotu v tomto případě, je nutné získat data z reálného použití lékaři.

## 5.3 Výsledky osvětlovací části zařízení

Osvětlení je dostatečné pro získání snímku sítnice. K ověření byl použit fantom oka. Jedná se o napodobeninu s podobnými optickými vlastnostmi jako je skutečné oko. Jelikož zařízení zatím nemá lékařskou certifikaci, nelze použít pro snímání skutečné sítnice.

Osvětlení je schopné pracovat neomezeně dlouho, jelikož jeho výkon není příliš vysoký a tedy nehrozí přehřívání žádné z komponent. K problémům by mohlo dojít při odpojení osvětlovací desky, tedy i diod. Ve chvíli, kdy je střída PWM signálu vyšší jak 0 %, pak hrozí poškození řídicího obvodu. Aby bylo možné detekovat odpojení, musel by být využit jiný typ konektoru, který podporuje detekci odpojení. Jednou z možností je použít 4-kolíkový konektor, dva z nich použít pro napájení LED, další dva propojit jako detekční. Na jeden z detekčních by se poté připojilo napětí a na druhém by se detekovala logická úroveň. Při odpojení by došlo ke změně logické úrovně na detekčním vstupu a napájení řídicího obvodu LED prvků by bylo vypnuto. Nicméně, aby byl řídicí obvod naprosto ochráněn, je nutné odpojit jeho napájení několik milisekund před odpojením diod. Napřed by tedy musel být odpojen detekční kontakt a poté napájení, i když takové konektory existují, většina z nich není vhodná pro základní desku a to z důvodu velkých rozměrů, jelikož jsou často dimenzovány na velká napětí.

### 5.3.1 Zaměřovací bod

Navržený OLED displej, sloužící jako zobrazovač záměrného bodu, je dostatečný pro pokusy, nicméně bylo by vhodné zajistit variantu s větším jemností displeje. To by mělo za následek, že je možné natočit zařízení do více úhlů. Nyní využitý displej má šířku pixelu 0,17 mm, ideální pixel by měl mít velikost 0,057 mm, to vyžaduje jemnost displeje okolo 440 PPI. Nyní posun manipulátoru o 1° vůči vodorovné ose zařízení odpovídá posunu záměrného bodu o 1,48 pixelu. Díky čemuž jsou přesné jen polohy, kde úhel manipulátor je sudé číslo.

OLED displeje s PPI okolo hodnoty 400, jejichž velikost je akceptovatelná pro koncept zařízení slouží většinou pro konstrukci nositelné elektroniky, jako jsou například chytré hodinky. Tyto displeje ovšem obsahují složitější rozhraní. To by muselo být integrováno na další verzi desky.

## 5.4 Návrhy na vylepšení

Jako další potencionální vylepšení jsou navrženy následující varianty. Ty budou pravděpodobně zakomponovány do návrhu další verze zařízení, která by se měla více přiblížit zařízení, jež bude možné vyrábět ve více kusech.

### 5.4.1 Doplnující Flash paměť

Jedním z vylepšení by měla být Flash paměť, která by byla připojena přes sběrnici SPI. Ta by měla být určena pro ukládání logovacích zpráv a perzistentně je uchovávat. Hlavním využitím by bylo sledování provozu zařízení a stav jednotlivých jeho součástí. Zjišťování těchto informací by probíhalo v režimu, kdy by bylo zařízení používáno oftalmology v jejich ordinacích. Na základě těchto dat by bylo možné provést optimalizace, na základě kterých by mohlo dojít ke snížení ceny, případně zvýšení spolehlivosti. Ukládána by byla data o četnosti použití, teplotách jednotlivých komponent, nutnosti chlazení, vytížení jednotlivých napájecích sekcí a podobně.

Vhodná paměť by měla mít velikost alespoň 64 MB. To by mělo poskytnout dostatek prostoru pro ukládání záznamů až po dobu jednoho měsíce, při vysokém vytížení zařízení. Nicméně paměti jsou dostupné i ve velikostech až do 256 MB. Následné přečtení paměti by mohlo být provedeno přes příkazové rozhraní struktury zařízení nebo pomocí externího zařízení pro čtení flash pamětí.

Samotné záznamy by mohly být ukládány jako prostá sekvence znaků spolu s časovými razítky získanými z RTC na mikrokontroléru. Tato sekvence by se nacházela přímo v externí paměti flash bez vyšší hierarchie uspořádání.

Případně pro vyšší přehlednost by bylo možné využít nějaký souborový systém určený pro vestavěná zařízení, jako je například souborový systém LittleFS. Jeho hlavními výhodami jsou odolnost proti ztrátě napájení, rozložení zápisů mezi všechny bloky paměti a velmi nízká spotřeba zdrojů [22]. Oproti souborovému systému SPIFFS<sup>1</sup>, který je také často používaný ve vestavěných zařízeních, nabízí ovšem výrazně vyšší rychlost zápisu, čtení a formátování.

S využitím souborového systému LittleFS by bylo možné rozdělit záznamy do souborů, například podle data a snadněji v nich vyhledávat. Výhodou je také, že při implementaci ovladače do mikrokontroléru by bylo možné zpřístupnit tento souborový systém v řídicím

<sup>1</sup><https://github.com/pellepl/spiffs>

počítači. Propojení je možné realizovat přes FUSE<sup>2</sup>, jehož podpora je v Linuxové jádře již od verze 2.6. Samotné rozhraní pro LittleFS na straně řídicího počítače se nazývá littlefs-fuse<sup>3</sup>.

### 5.4.2 Napájení přes konektor USB-C

S využitím standardu PowerDelivery 2.0 pro USB verze 3.0 je možné dosáhnout napájecího výkonu až 100 W, při napětí až 20 V. To je dostatečný výkon pro napájení všech součástí kromě motorů. U motorů by tento výkon nestačil ani pokud by byl jejich pohyb rozložen tak, aby se najednou pohyboval jen jeden motor. Nicméně by tak bylo možné zařídit běh vestavěného počítače, displeje a řídicí desky. Bylo by možné tak například zajistit přístup ke snímkům, které jsou uloženy ve vestavěném počítači. To není realizovatelné přes aktuální USB napájení, které umožňuje maximální výkon 0,5 W.

Pro realizaci tohoto způsobu napájení je vhodné využít integrovaný obvod, který v rámci interního protokolu USB zajistí potřebné napájení a následně bude toto napájení spravovat. Vhodným kontrolérem je například obvod STUSB4500<sup>4</sup>, díky němuž není potřeba implementovat komunikaci a vyjednání napájecího limitu v hlavním mikrokontroléru.

V případě že by další verze byly založeny na jiném manipulátoru, jehož motor nejsou tak výkonově náročné, by bylo potenciálně možné realizovat veškeré napájení přes USB-C.

### 5.4.3 Rozšiřující konektor

Pro další libovolná rozšíření bez zásahu do desky by bylo možné přidat speciální rozšiřující konektor. Ten by umožnil připojení samotných senzorů nebo i případně rozšiřující desky s dalším mikrokontrolérem. K tomu může do jisté míry sloužit i konektor, který funguje jako debugovací zařízení. Nicméně při vývoji rozšiřující desky by bylo vhodné mít je oba k dispozici najednou.

K rozšiřující desce by poté mohly být připojeny senzory, které by zamezovaly pohybu, jež by potenciálně mohl poškodit zařízení. Při manuálním ovládní je totiž možné nastavit osy tak, aby se při přejetí na domovskou pozici dostaly do kolize. Díky senzorům umístěným například na pevném okuláru fundus kamery by tomu bylo možné zabránit.

### 5.4.4 Přesun knihovny na RTOS

S využitím Real-time operačního systému (dále zkráceno jako RTOS) by bylo možné snadněji řešit neprioritní situace spouštěné přerušením. Aktuálně je každá situace aktivována přerušením považovaná jako prioritní k té co aktuálně běží. Jediné řešení jak tomuto zamezit je nyní maskování přerušení s nižší prioritou, což je poměrně těžko udržitelné řešení a návrhem nezapadá do knihovny ALOHAL. Stále také hrozí vyhladovění některých vláken. V případě využití RTOS by bylo možné z rutiny přerušení vytvořit nové vlákno o předem dané prioritě. Díky vhodnému nastavení by bylo možné zamezit vyhladovění nízko prioritních vláken. K vyhladovění může aktuálně vzácně dojít, například v případě že dochází k přesunu na nulovou pozici a rychle po sobě dorazí více os do výchozí pozice, tak dochází ke zpoždění na komunikačním rozhraní.

Nevýhodou využití RTOS je souběžný přístup ke zdrojům z více vláken programu. Například přístup k perifériím MCU musí být kontrolován pomocí semaforů, případně jiných synchronizačních mechanismů, jinak by mohlo dojít k přerušení aktivní funkce periférie.

<sup>2</sup><https://github.com/libfuse/libfuse>

<sup>3</sup><https://github.com/ARMmbed/littlefs-fuse>

<sup>4</sup><https://www.st.com/en/interfaces-and-transceivers/stusb4500.html>

Další dosud neznámé problémy mohou vzniknout v rámci knihovny HAL, která není pro vícevláknové vykonávání přizpůsobena. Potenciální použitelný operačním systémem je FreeRTOS, který podporuje téměř všechny mikrokontroléry z portfolia STMicroelectronics.



# Literatura

- [1] *L5973D - 2.5 A switch step down switching regulator*. 03IN1439. STMicroelectronics, leden 2008. Rev. 16.
- [2] *EMC2305 - Multiple RPM-Based PWM Fan Controller for Five Fans*. SMSC, květen 2011. Rev. 1.3.
- [3] *USBL6-2 - Very low capacitance ESD protection*. 11265. STMicroelectronics, říjen 2011. Rev. 5.
- [4] *NC23017 - 1.0A Constant-Current Buck Regulator for Driving High Power LEDs*. NCL30160/D. On Semiconductors, leden 2012. Rev. 1.
- [5] *Cree PLCC2 1-in-1 SMD LED CLM3C-WKW/MKW*. CLD-Ct1076.004. CREE, Inc., 2014. Rev. 1.
- [6] *L6470 datasheet- Fully integrated microsteppingmotor driver with motion engine and SPI*. 16737. STMicroelectronics, březem 2015. Rev. 7.
- [7] *PTS1206 - DC surface mount resettable PTC devices*. TD-4397. EATON, říjen 2015. Rev. 1.
- [8] *HLG-480H series Constant Voltage LED driver*. Mean well, prosinec 2016. Rev. 1.
- [9] *MCP23017/MCP23S17 datasheet*. DS20001952C. Microchip, červenec 2016. Rev. C.
- [10] *OIS-330 770*. 95/EC. Osa - optolight, září 2017. Rev. 1.
- [11] *WS2818B - Single-line 256 Gray-level 3-channel Constant Current LED Driver IC*. 0523. WorldSemi, květen 2017. Rev. 1.
- [12] *ACHS712 - Fully Integrated, Hall-Effect-Based Linear Current Sensor IC*. DS101. Broadcom, únor 2018.
- [13] *AOD4189 P-Channel Enhancement Mode Field Effect Transistor*. 8X11. ALPHA and OMEGA semiconductors, listopad 2018. Rev. 1.
- [14] *STM32L432xx datasheet*. DS11451. STMicroelectronics, květen 2018. Rev. 4.
- [15] *VSS / VSH Stepper Motor for Applications up to Ultra-High-Vacuum*. DS-074-A015 EN / 15. Phytron, listopad 2018.
- [16] *STM32L452xx datasheet*. DS11912. STMicroelectronics, leden 2019. Rev. 5.

- [17] *TMC2130 datasheet - Universal high voltage driver for two-phase bipolar stepper motor*. Trinamic, listopad 2019. Rev. 1.11.
- [18] *ACS712 - Fully Integrated, Hall-Effect-Based Linear Current Sensor IC*. MCO-0000197. Allegro MicroSystems, LLC, leden 2020. Rev. 19.
- [19] *USFF 1206 - Surface Mount Fuse, Super-Quick-Acting FF*. E41599. EATON, březen 2020. Rev. B.
- [20] ALL CB, s. *Oftalmoskop HEINE BETA 200S* [online]. 2020 [cit. 2020-05-23]. Dostupné z: <https://www.allcb.cz/image.php?nid=11760&oid=7562030&width=900>.
- [21] ANGELICA LY, LISA NIVISON, NAGI ASSAAD. Infrared reflectance imaging in age-related macular degeneration. Květen 2016.
- [22] ARMMBED. LittleFS. *Github.com* [online]. 2020 [cit. 2020-03-15]. Dostupné z: <https://github.com/ARMBED/littlefs>.
- [23] BEYE. *TRC-NW300 Non-Mydriatic Retinal Camera* [online]. 2020 [cit. 2020-05-23]. Dostupné z: <https://www.beye.com/product/trc-nw300-non-mydriatic-retinal-camera>.
- [24] CHRIS SIMMONDS. *Mastering Embedded Linux Programming*. 2th. Packt Publishing, 2017. ISBN 978-1-78728-328-2.
- [25] CLYDE F. COOMBS, HAPPY HOLDEN. *Printed Circuits Handbook*. 7th. McGraw-Hill Education, 2016. ISBN 978-0071833950.
- [26] DHRUTI RANJAN GAAN, MANOJ KUMAR. Real-Time Precise Position Tracking With Stepper Motor Using Frequency Modulation Based Microstepping. Leden 2018, s. 693–701. ISSN 1939-9367.
- [27] EDWARD DEHOOG, JAMES SCHWIEGERLING. Fundus camera systems: a comparative analysis. *Annalen der Physik*. 2009, roč. 322, č. 10, s. 891–921.
- [28] EMANUELE TRUCCO, TOM MACGILLIVRAY. *Computational Retinal Image Analysis: Tools, Applications and Perspectives*. Academic Press, 2019. ISBN 9780081028162.
- [29] EVANS, K. *Programming of CNC Machines*. 4th. Industrial Press Inc., 2016. ISBN 0831135247.
- [30] HERITY, D. Modern C++ in embedded systems. *Embedded* [online]. 2020 [cit. 2020-03-04]. Dostupné z: <https://www.embedded.com/modern-c-in-embedded-systems-part-1-myth-and-reality/>.
- [31] HUDGINS, D. *Integrated-Resistor Current Sensors Simplify PCB Design*. SBOA197. Texas Instruments Incorporated, 2017.
- [32] I. C. STIKA AND L. KREINDLER AND A. SARCA. A robust method for stepper motor stall detection. May 2013. ISSN 2068-7966.
- [33] JAN KOLÍN. *Oftalmologie praktického lékaře*. Univerzita Karlova, 1994. ISBN 80-7066-861-X.

- [34] JARMILA HEISSIGEROVÁ A KOL. *Oftalmologie*. MAXDORF, 2018. ISBN 80-7345-580-3.
- [35] JASON PORTER, HOPE QUEENER, JULIANNA LIN. *Adaptive Optics for Vision Science: Principles, Practices, Design, and Applications*. John Wiley and Sons, říjen 2005. ISBN 9780471914877.
- [36] JOHN CUMMINGS, MICHAEL C. DOOGUE, ANDREAS P. FRIEDRICH. *Recent Trends in Hall Effect Current Sensing*. AN295045. Allegro MicroSystems, LLC, 2013. Rev. 1.
- [37] JONATHAN CORBET, ALESSADRO RUBINI, GRED KROAH-HARTMAN. *Linux Device Drivers*. 3th. O'Reilly Media, 2005. ISBN 0-596-00590-3.
- [38] JONES, D. EEVBlog - PCB Design guide. 2004. Rev. A, online via Altium library. Dostupné z: <http://www.alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf>.
- [39] KARL J. ASTRÖM, TORE HÄGGLUND. *PID Controllers, Theory, Design and Tuning*. ISA, 1995. 2nd Edition. ISBN 978-1-55617-516-9.
- [40] LÁSLÓ KEVICZKY, RUTH BARS, JENŐ HETTHÉSSY. *Control Engineering*. Springer, 2018. ISBN 9789811082979.
- [41] M. FROHN, W. OBERTHUR. *Elektronika - polovodičové součástky a základní zapojení*. BEN - technická literatura, 2006. ISBN 80-7300-123-3.
- [42] NANCY B. CARLSON, DANIEL KURTZ. *Clinical Procedures for Ocular Examination, Third Edition*. McGraw-Hill Education, 2004. Medical Publishing Division. ISBN 9780071370783.
- [43] NAOHISA SHIBATA, MIWAKO TORII. *Fundus camera*. Patent No. US6654553B2, 2001.
- [44] PETER SMID. *CNC Programming Techniques*. 1th. Industrial Press Inc., 2006. ISBN 0831131853.
- [45] POCO PROJECT. Poco Logger. *PocoProject.org* [online]. 2020 [cit. 2020-02-20]. Dostupné z: <https://pocoproject.org/docs/Poco.Logger.html>.
- [46] POLOLU CORPORATION. Understanding Destructive LC Voltage Spikes. 2019, 0J16.
- [47] RAAB, W. *Indirect Ophthalmoscope*. Springer Berlin Heidelberg, 2018. 926–927 s. ISBN 978-3-540-69000-9.
- [48] RICHARD DICKINSON, ANDREAS FRIEDRICH. *Using Allegro Current Sensor ICs in Current Divider Configurations for Extended Measurement Range*. AN295036. Allegro MicroSystems, LLC, 2013. Rev. 3.
- [49] SCARPINO, M. *Motors for Makers: A Guide to Steppers, Servos, and Other Electrical Machines*. Que Publishing, 2015. ISBN 978-0852964170.
- [50] SHAUN MILANO. Allegro ACS716 Fully Integrated Current Sensor. *Allegro MicroSystems, LLC* [online]. 2020 [cit. 2020-04-06]. Dostupné z: [https://www.allegromicro.com/en/products/sense/current-sensor-ics/zero-to-fifty-amp-integrated-conductor-sensor-ics/acs716/allegro-acs716-fully-integrated-current-sensor-ics-for-use-in-3\\_3-v-applications/](https://www.allegromicro.com/en/products/sense/current-sensor-ics/zero-to-fifty-amp-integrated-conductor-sensor-ics/acs716/allegro-acs716-fully-integrated-current-sensor-ics-for-use-in-3_3-v-applications/).

- [51] SYMPTOMY.CZ. *Stavba oka* [online]. 2020 [cit. 2020-05-23]. Dostupné z:  
<https://www.symptomy.cz/anatomie/oci/stavba-oka.jpg>.
- [52] T. ZHAO, W. SHEN, N. JI, H. LIU. Study and implementation of SPWM microstepping controller for stepper motor. May 2018, s. 2298–2302. ISSN 2158-2297.
- [53] TSUGUO NANJO, MASUNORI KAWAMURA. *Fundus camera*. Patent No. US5742374A, 1996.

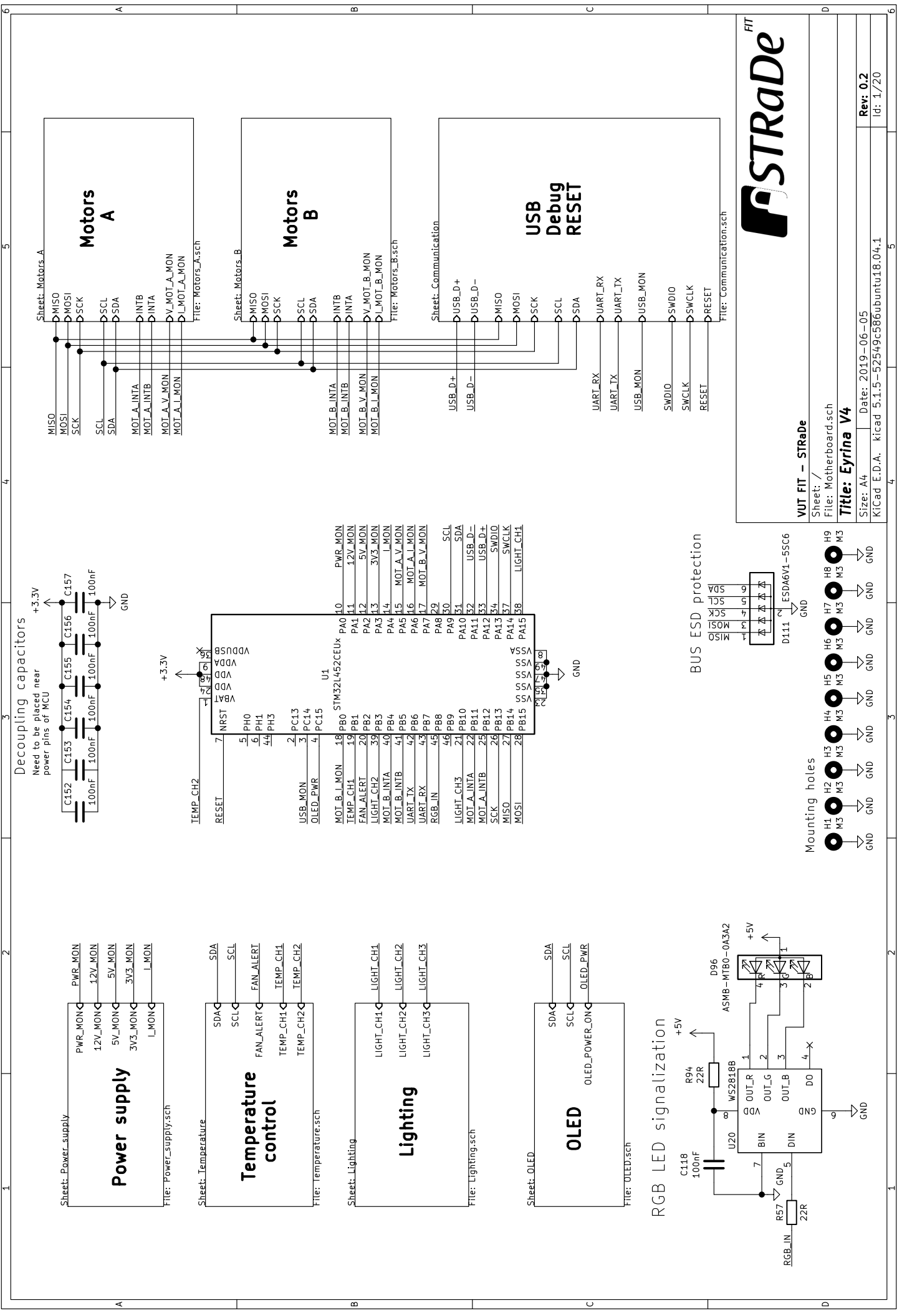
# Příloha A

## Schéma základní desky

Návrh plošného spoje je proveden v programu KiCAD<sup>1</sup>. Návrh je hierarchický, proto se opakuje několik variant stejného schématu. Opakování instancí listů schématu je nutné kvůli odlišným referenčním značkám součástek v jednotlivých instancích.

---

<sup>1</sup><https://kicad-pcb.org/>



1 2 3 4 5 6

A B C D

Sheet: Power\_supply

PWR\_MON  
12V\_MON  
5V\_MON  
3V3\_MON  
L\_MON

**Power supply**

File: Power\_supply.sch

Sheet: Temperature

SDA  
SCL  
FAN\_ALERT  
TEMP\_CH1  
TEMP\_CH2

**Temperature control**

File: temperature.sch

Sheet: Lighting

LIGHT\_CH1  
LIGHT\_CH2  
LIGHT\_CH3

**Lighting**

File: Lighting.sch

Sheet: OLED

SDA  
SCL  
OLED\_POWER\_ON

**OLED**

File: OLED.sch

Sheet: Motors\_A

MISO  
MOSI  
SCK  
SCL  
SDA  
MOT\_A\_INTA  
MOT\_A\_INTB  
MOT\_A\_V\_MON  
MOT\_A\_L\_MON

**Motors A**

File: Motors\_A.sch

Sheet: Motors\_B

MISO  
MOSI  
SCK  
SCL  
SDA  
MOT\_B\_INTA  
MOT\_B\_INTB  
MOT\_B\_V\_MON  
MOT\_B\_L\_MON

**Motors B**

File: Motors\_B.sch

Sheet: Communication

USB\_D+  
USB\_D-  
MISO  
MOSI  
SCK  
SCL  
SDA  
UART\_RX  
UART\_TX  
USB\_MON  
SWDIO  
SWCLK  
RESET

**USB Debug RESET**

File: Communication.sch

Sheet: U1

7 NRST  
5 PH0  
6 PH1  
4 PH3  
2 PC13  
3 PC14  
4 PC15  
10 PWR\_MON  
11 12V\_MON  
12 5V\_MON  
13 3V3\_MON  
14 L\_MON  
15 MOT\_A\_V\_MON  
16 MOT\_A\_L\_MON  
17 MOT\_B\_V\_MON  
18 MOT\_B\_L\_MON  
19 PB1  
20 PB2  
39 PB5  
40 PB4  
41 PB6  
42 PB6  
43 PB7  
44 PB8  
46 PB8  
49 PB9  
21 PB10  
22 PB11  
25 PB12  
26 PB12  
27 PB13  
28 PB14  
28 PB15  
PA0  
PA1  
PA2  
PA3  
PA4  
PA5  
PA6  
PA7  
PA8  
PA9  
PA10  
PA11  
PA12  
PA13  
PA14  
PA15  
SCL  
SDA  
USB\_D-  
USB\_D+  
SWDIO  
SWCLK  
LIGHT\_CH1

**Decoupling capacitors**

Need to be placed near power pins of MCU

+3.3V

C152  
C153  
C154  
C155  
C156  
C157  
100nF  
100nF  
100nF  
100nF  
100nF

TEMP\_CH2  
RESET  
VDDA  
VDD  
VDD

Sheet: U2

RGB\_IN  
GND  
DO  
OUT\_L  
OUT\_G  
OUT\_B

**RGB LED signalization**

File: RGB\_LED.sch

Sheet: D96

ASMB-MTB0-0A3A2  
+5V  
GND

**BUS ESD protection**

D111 ESDA6V1-55C6

Mounting holes  
H1 M3  
H2 M3  
H3 M3  
H4 M3  
H5 M3  
H6 M3  
H7 M3  
H8 M3  
H9 M3

GND  
GND  
GND  
GND  
GND  
GND  
GND  
GND  
GND

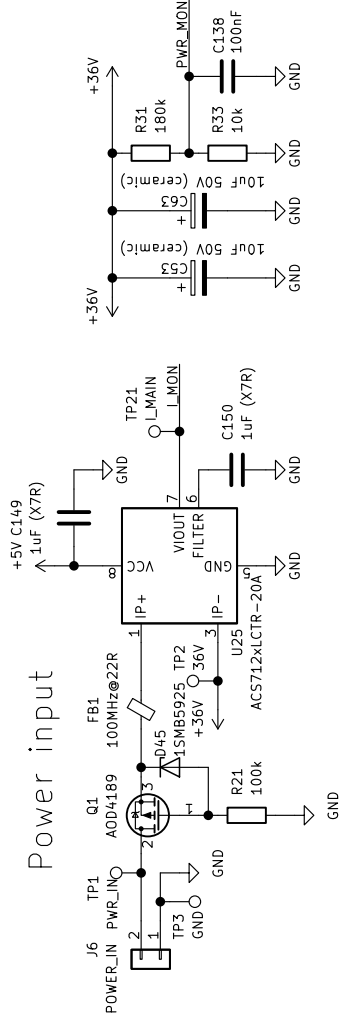


VUT FIT - STRaDe  
Sheet: /  
File: Motherboard.sch  
**Title: Eyrina V4**  
Size: A4  
Date: 2019-06-05  
KiCad E.D.A. kicad 5.1.5-52549c586buntuntu18.04.1

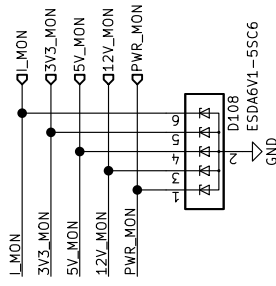
Rev: 0.2

Id: 1/20

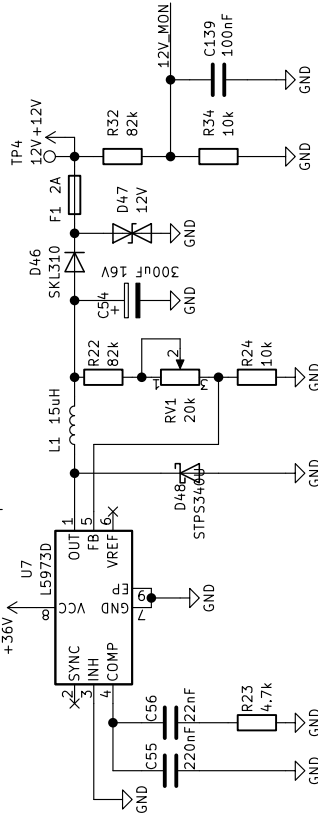
### Power input



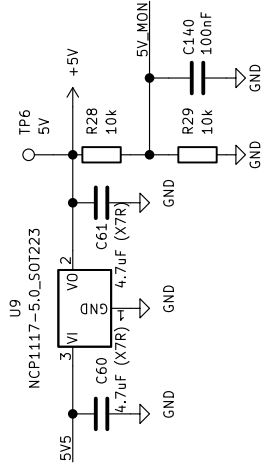
### ESD protection



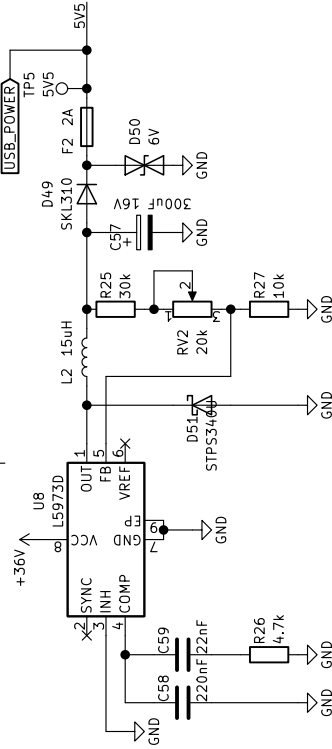
### 12V step-down



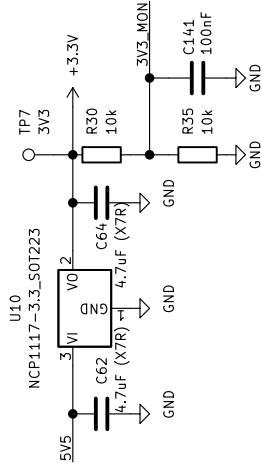
### 5V regulator



### 6V step-down



### 3V3 regulator



VUT FIT - STRaDe

Sheet: /Power\_supply/  
File: Power\_supply.sch

Title: Eyrina V4

Size: A4

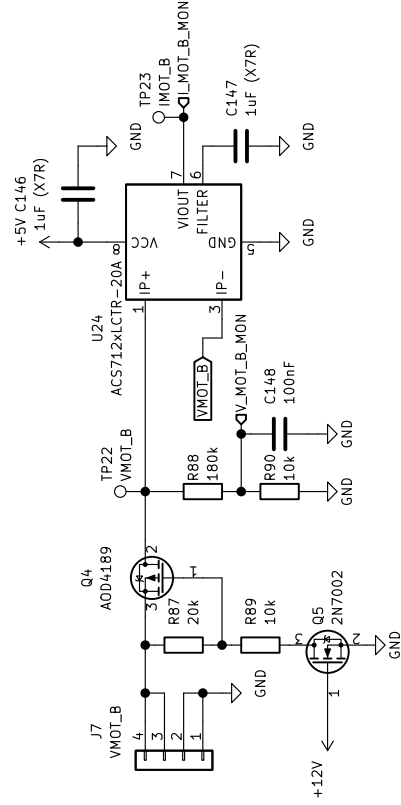
Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

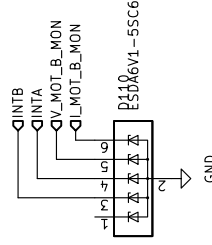
Rev: 0.2

Id: 2/20

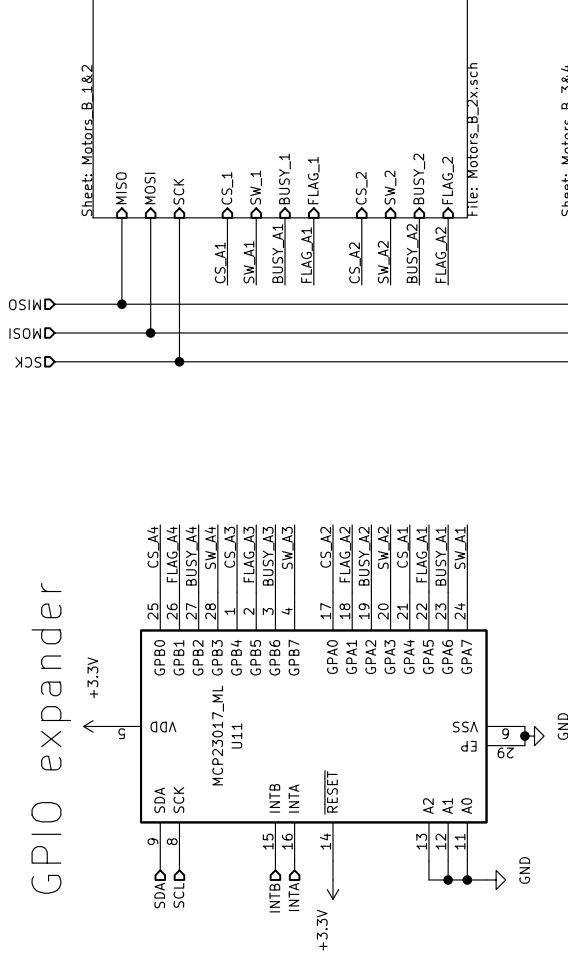
# Power monitoring



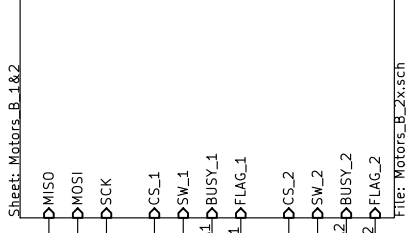
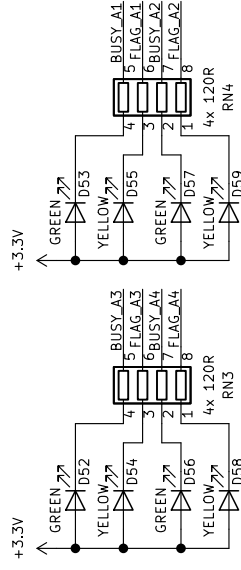
# ESD protection



# GPIO expander



# Stepper signals



VUT FIT - STRaDe

Sheet: /Motors\_B/

File: Motors\_B.sch

Title: Eyrina V4

Size: A4

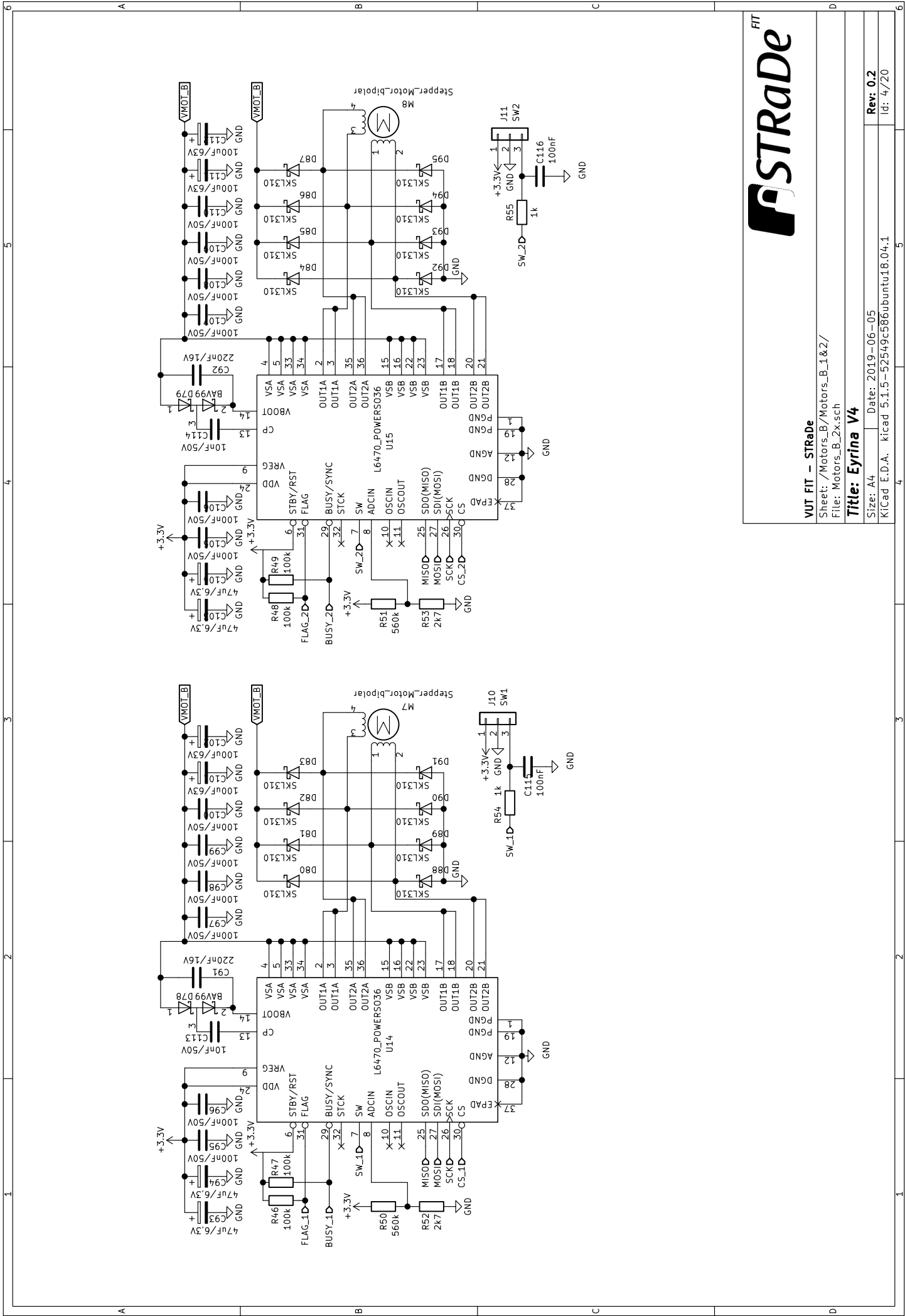
Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

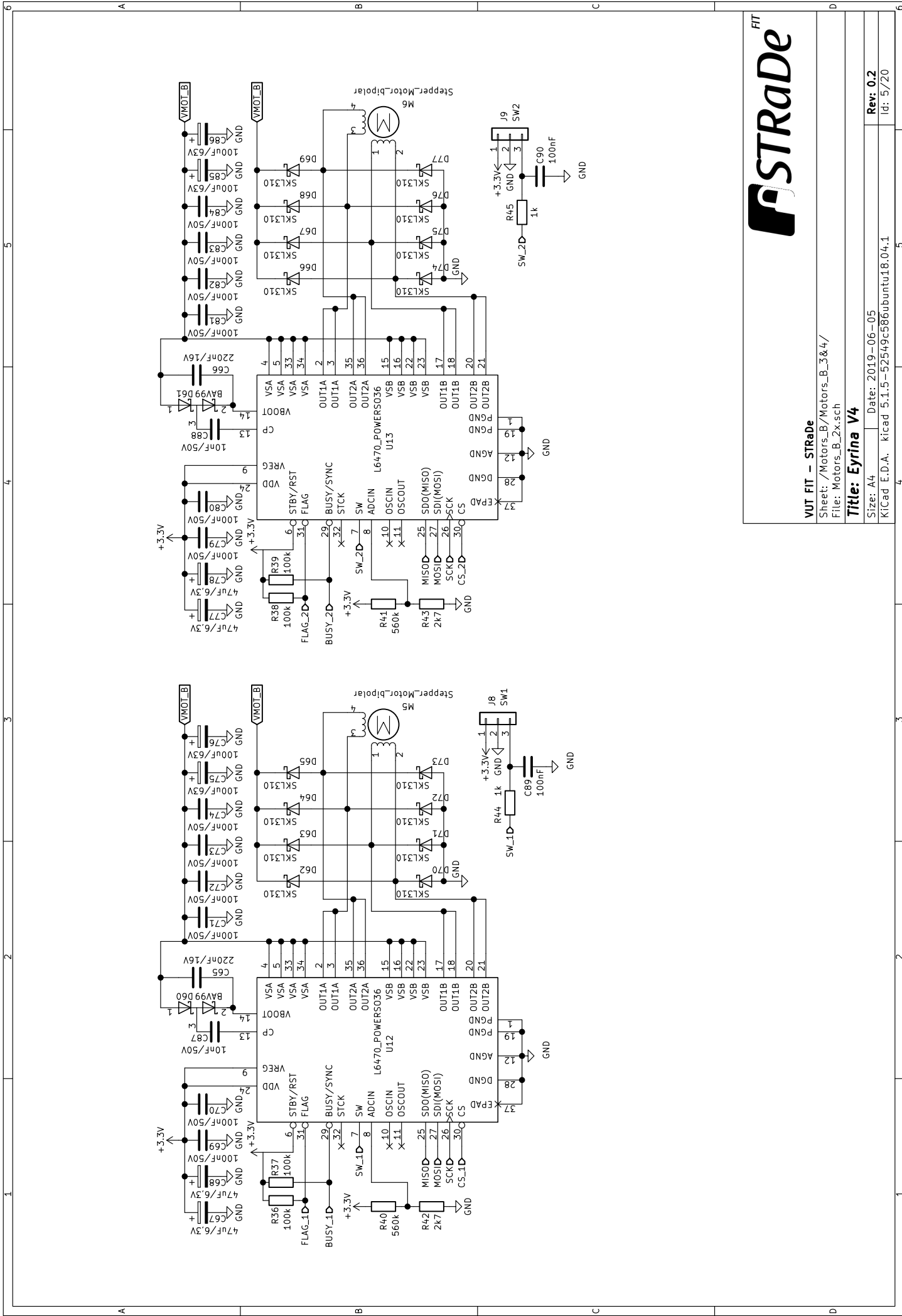
Rev: 0.2

Id: 3/20





<b>VUT FIT - STRaDe</b>	
Sheet: /Motors_B/Motors_B_1&2/	
File: Motors_B_2x.sch	
<b>Title: Eyrina V4</b>	
Size: A4	Date: 2019-06-05
KiCad E.D.A. kicad 5.1.5-52549c586bunt18.04.1	
Rev: 0.2	Id: 4/20

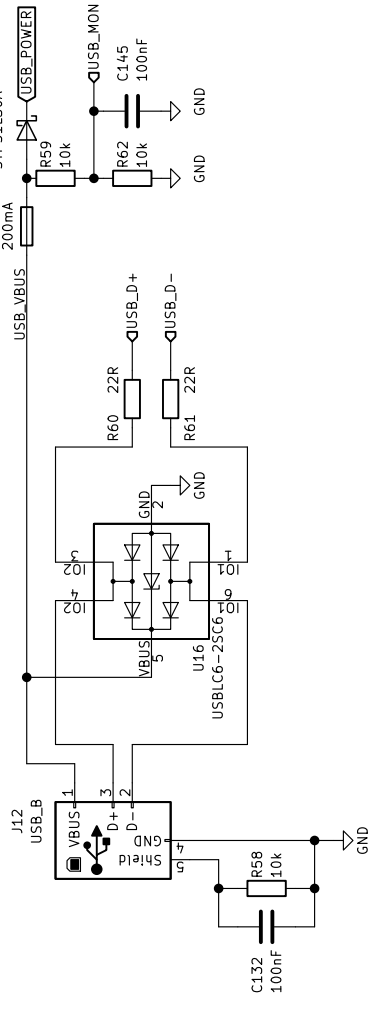


**VUT FIT - STRaDe**  
 Sheet: /Motors\_B/Motors\_B\_3&4/  
 File: Motors\_B\_2x.sch

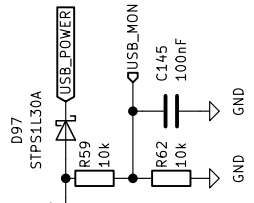
**Title: Eyrina V4**

Size: A4	Date: 2019-06-05
KiCad E.D.A. kicad 5.1.5-52549c586bunttu18.04.1	Rev: 0.2
	Id: 5/20

### USB input

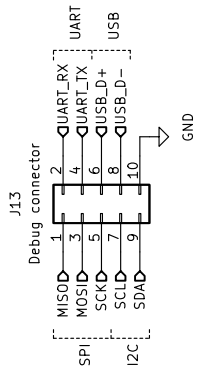


### USB power

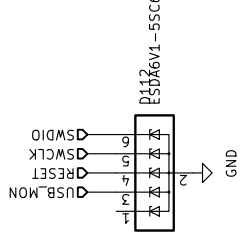
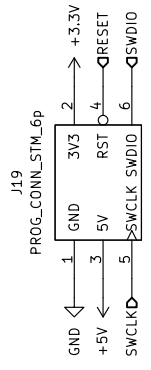


### Debug connector

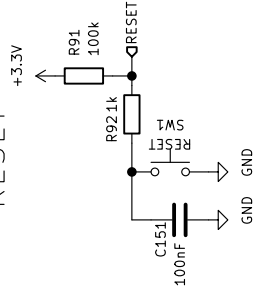
Compatible layout with Salae 8CH



### Programming connector



### RESET



VUT FIT - STRaDe

Sheet: /Communication/  
File: Communication.sch

**Title: Eyrina V4**

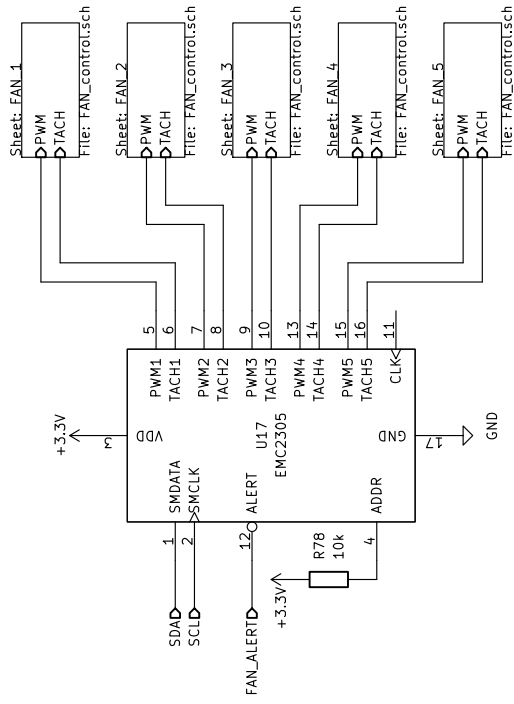
Size: A4 Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586ubuntu18.04.1

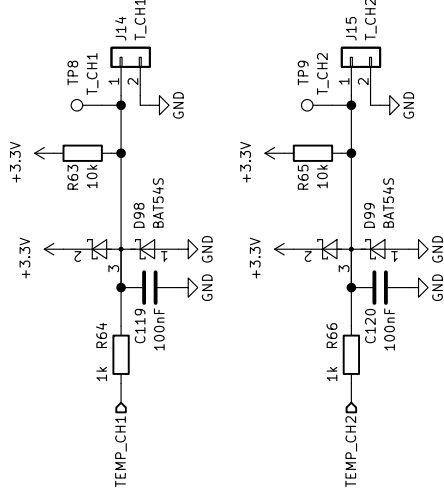
Rev: 0.2

Id: 6/20

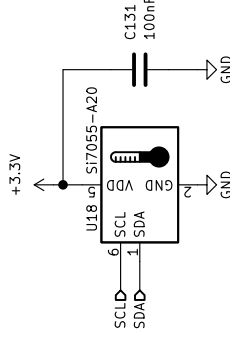
# FAN Control



# Termistors



# Ambient temperature



VUT FIT - STRaDe

Sheet: /Temperature/  
File: Temperature.sch

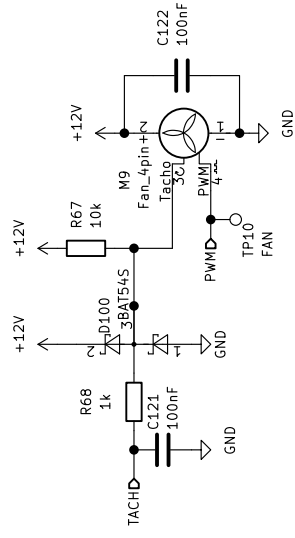
Title: Eyrina V4

Size: A4 Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

Rev: 0.2

Id: 7/20



VUT FIT – STRaDe

Sheet: /Temperature/FAN\_1/  
File: FAN\_control.sch

**Title: Eyrina V4**

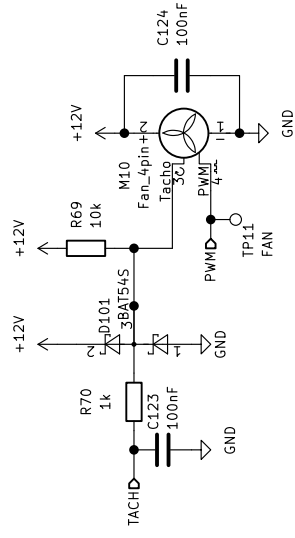
Size: A4

Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

Rev: 0.2

Id: 8/20



VUT FIT – STRaDe

Sheet: /Temperature/FAN\_2/  
File: FAN\_control.sch

**Title: Eyrina V4**

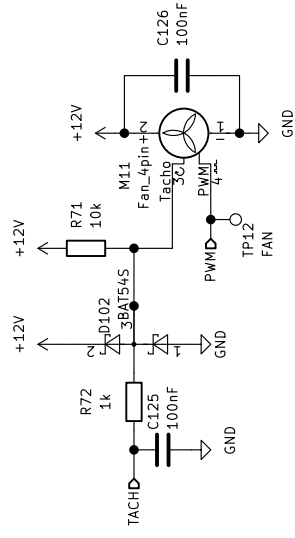
Size: A4

Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

Rev: 0.2

Id: 9/20



VUT FIT – STRaDe

Sheet: /Temperature/FAN\_3/  
File: FAN\_control.sch

**Title: Eyrina V4**

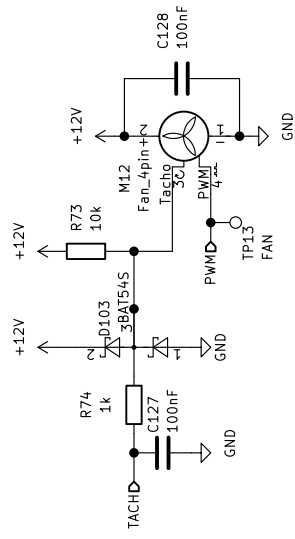
Size: A4

Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntui18.04.1

Rev: 0.2

Id: 10/20



VUT FIT - STRaDe

Sheet: /Temperature/FAN\_4/  
File: FAN\_control.sch

**Title: Eyrina V4**

Size: A4

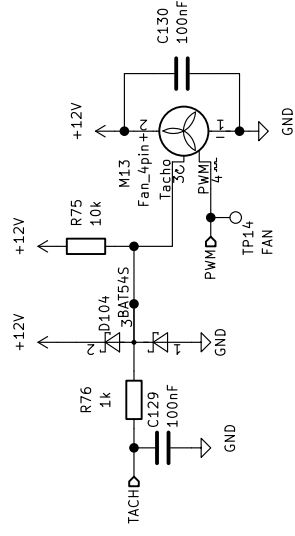
Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

Rev: 0.2

Id: 11/20





VUT FIT – STRaDe

Sheet: /Temperature/FAN\_5/  
File: FAN\_control.sch

**Title: Eyrina V4**

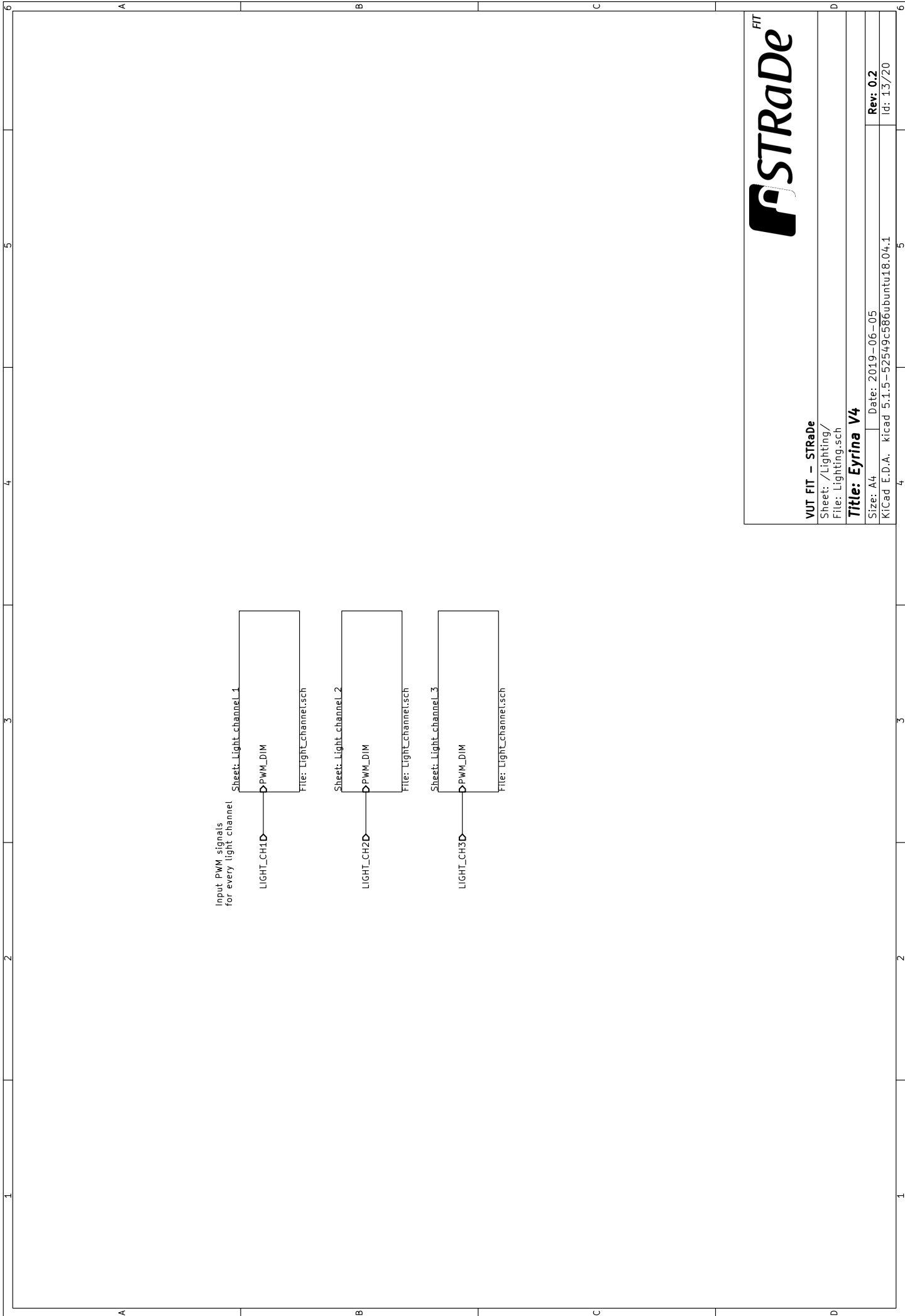
Size: A4

Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntui18.04.1

Rev: 0.2

Id: 12/20



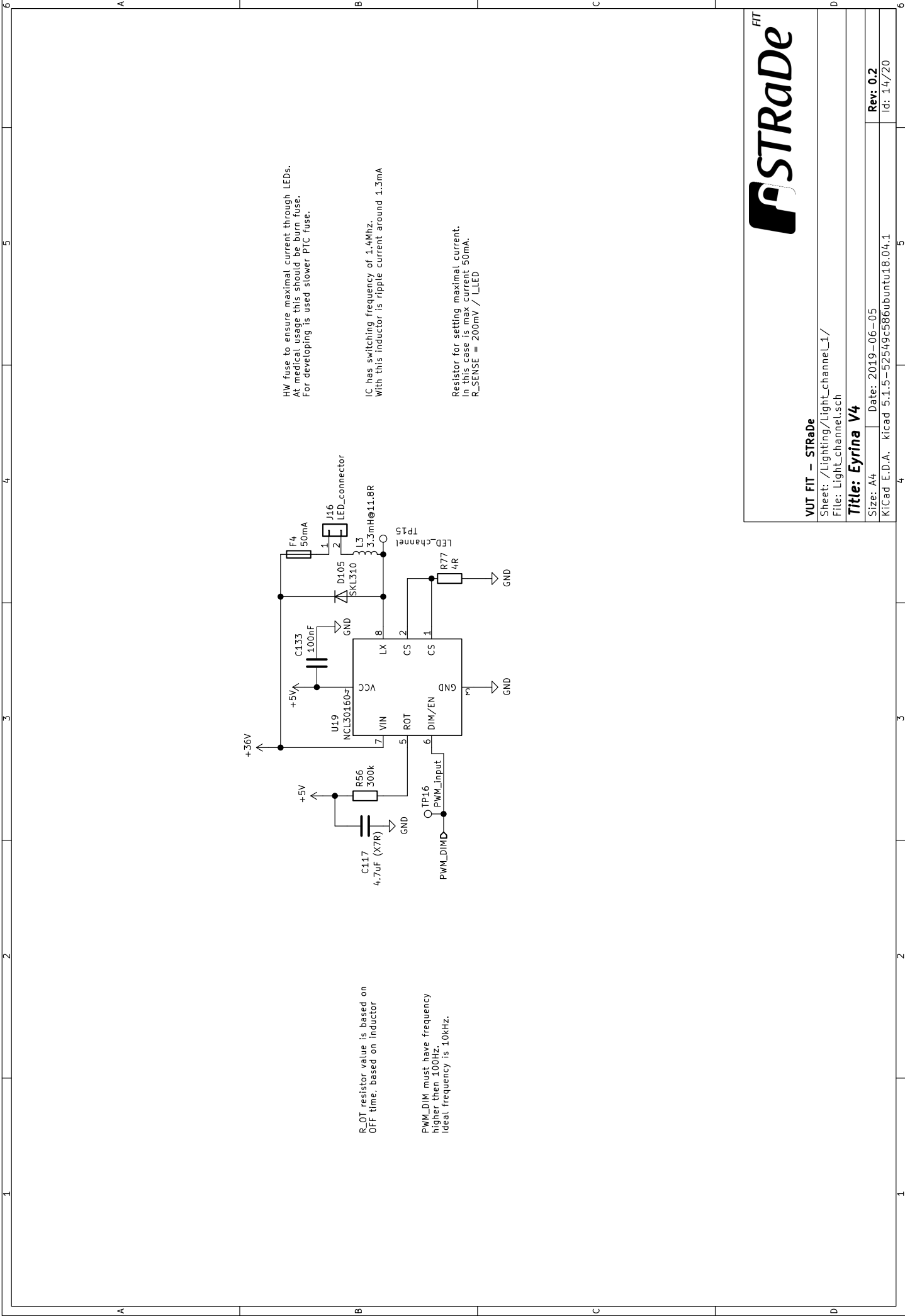
VUT FIT – STRaDe

Sheet: /Lighting/  
File: Lighting.sch

**Title: Eyrina V4**

Size: A4 Date: 2019-06-05  
KiCad E.D.A. kicad 5.1.5-52549c586bunttu18.04.1

Rev: 0.2  
Id: 13/20



R\_OT resistor value is based on OFF time, based on inductor

PWM\_DIM must have frequency higher than 100kHz. Ideal frequency is 10kHz.

HW fuse to ensure maximal current through LEDs. At medical usage this should be burn fuse. For developing is used slower PTC fuse.

IC has switching frequency of 1.4Mhz. With this inductor is ripple current around 1.3mA

Resistor for setting maximal current. In this case is max current 50mA. R\_SENSE = 200mV / I\_LED

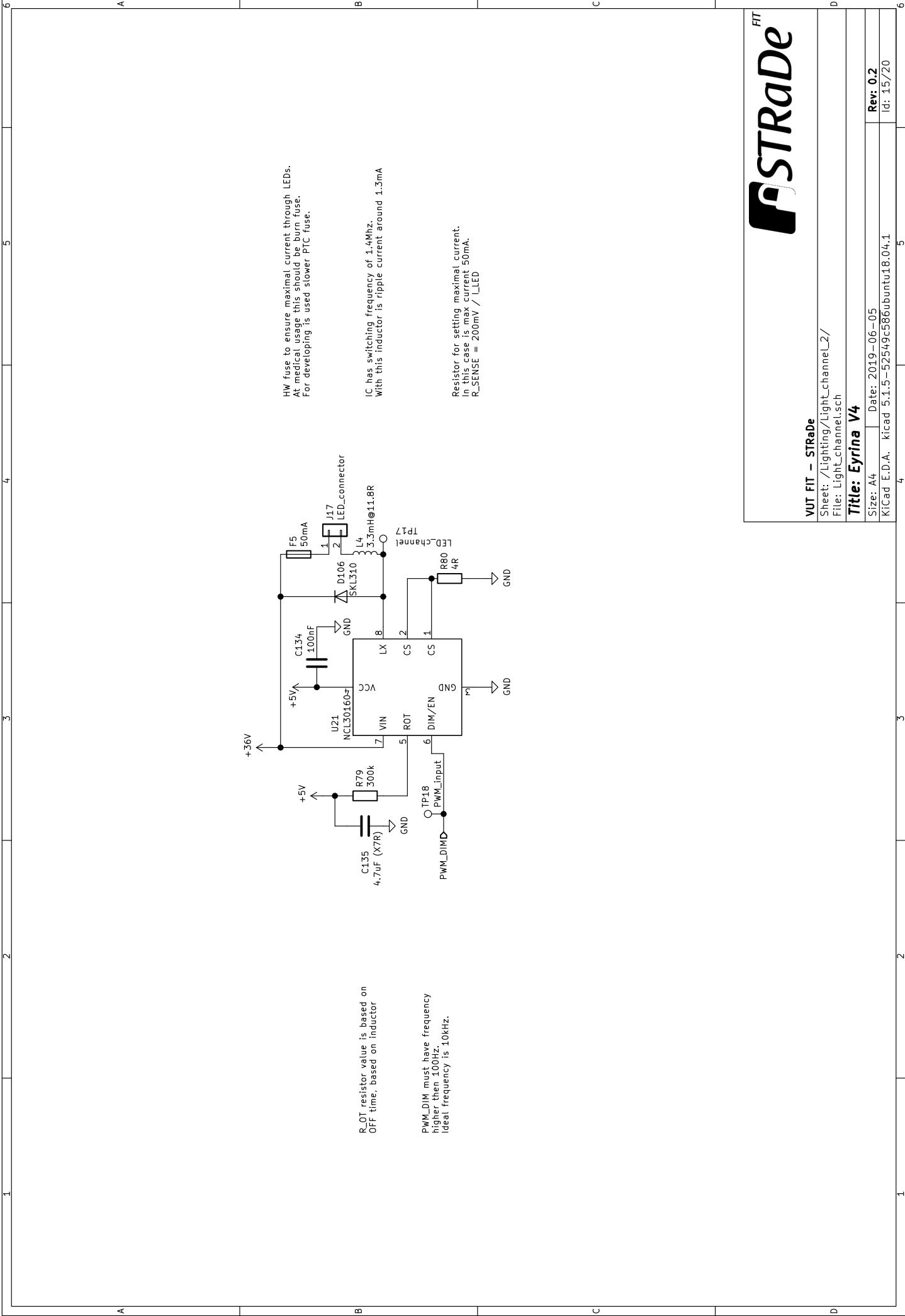


**VUT FIT – STRaDe**

Sheet: /Lighting/Light\_channel\_1/  
File: Light\_channel.sch

**Title: Eyrina V4**

Size: A4	Date: 2019-06-05	Rev: 0.2
KiCad E.D.A.	kiCad 5.1.5-52549c586buntu18.04.1	Id: 14/20



HW fuse to ensure maximal current through LEDs.  
 At medical usage this should be burn fuse.  
 For developing is used slower PTC fuse.

IC has switching frequency of 1.4Mhz.  
 With this inductor is ripple current around 1.3mA

Resistor for setting maximal current.  
 In this case is max current 50mA.  
 $R_{SENSE} = 200mV / I_{LED}$

R\_OT resistor value is based on  
 OFF time, based on inductor

PWM\_DIM must have frequency  
 higher then 100kHz.  
 Ideal frequency is 10kHz.



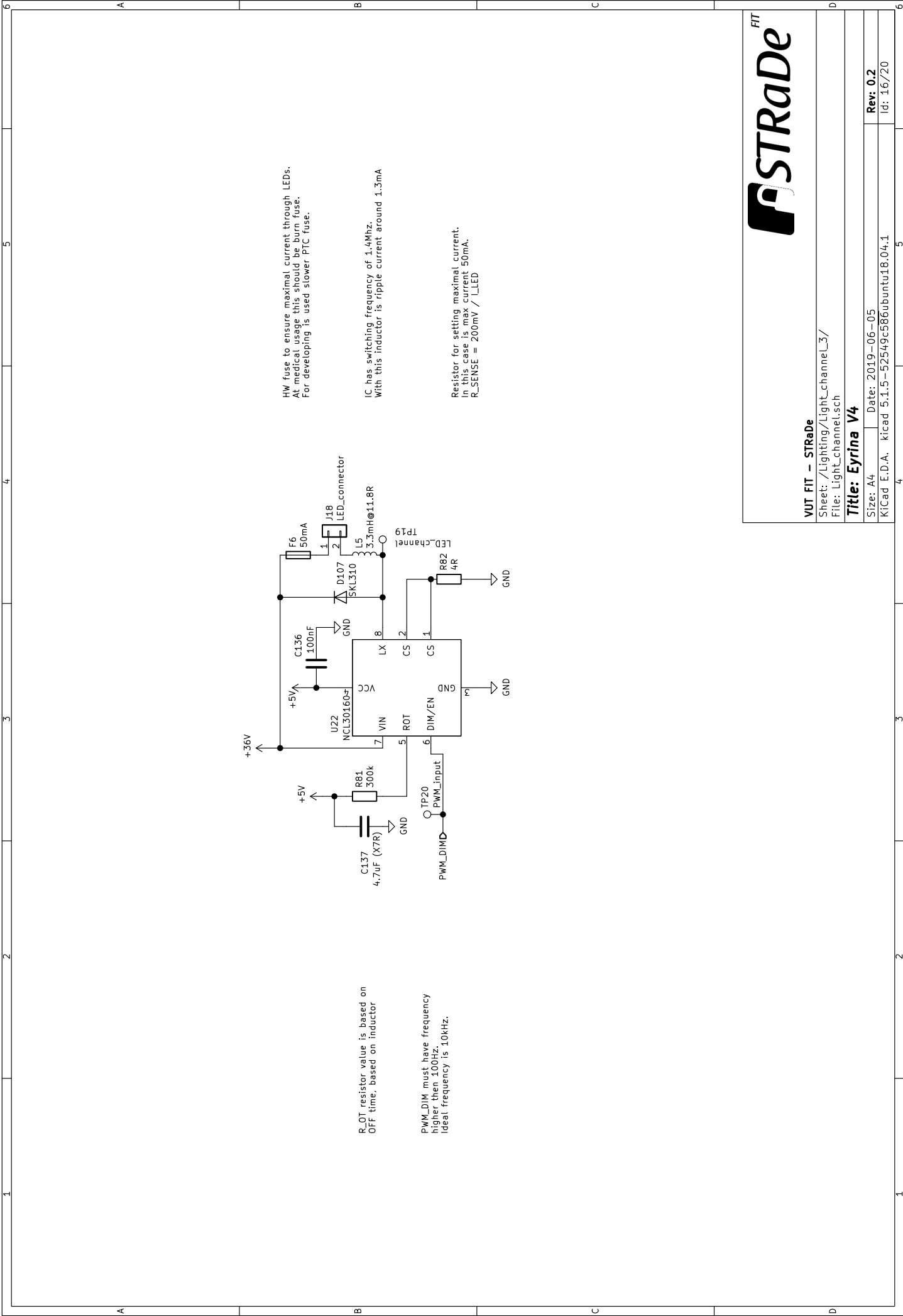
VUT FIT – STRaDe

Sheet: /Lighting/Light\_channel\_2/  
 File: Light\_channel.sch

**Title: Eyrina V4**

Size: A4	Date: 2019-06-05
KiCad E.D.A.	kiCad 5.1.5-52549c586buntu18.04.1

Rev: 0.2  
 Id: 15/20



R\_OT resistor value is based on OFF time, based on inductor

PWM\_DIM must have frequency higher than 100Hz. Ideal frequency is 10kHz.

HW fuse to ensure maximal current through LEDs. At medical usage this should be burn fuse. For developing is used slower PTC fuse.

IC has switching frequency of 1.4Mhz. With this inductor is ripple current around 1.3mA

Resistor for setting maximal current. In this case is max current 50mA. R\_SENSE = 200mV / I\_LED

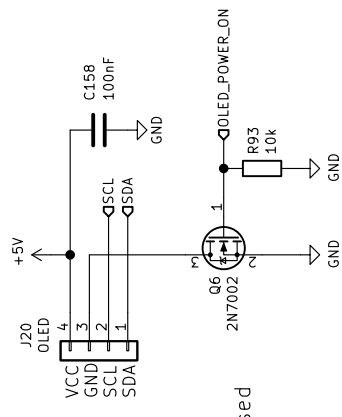


**VUT FIT – STRaDe**

Sheet: /Lighting/Light\_channel\_3/  
File: Light\_channel.sch

**Title: Eyrina V4**

Size: A4	Date: 2019-06-05	Rev: 0.2
KiCad E.D.A.	kiCad 5.1.5-52549c586buntu18.04.1	Id: 16/20



MOSFET for disabling  
LCD backlight white not used



VUT FIT - STRaDe

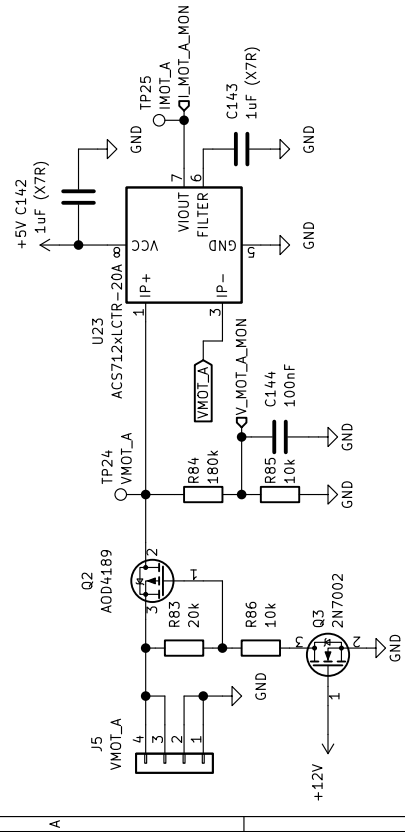
Sheet: /OLED/  
File: OLED.sch

**Title: Eyrina V4**

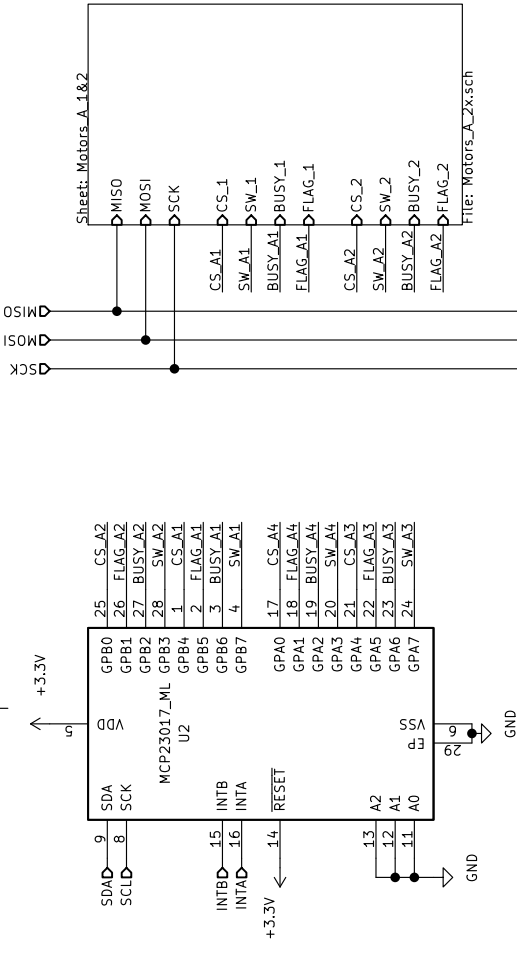
Size: A4 Date: 2019-06-05  
KiCad E.D.A. kicad 5.1.5-52549c586buntui18.04.1

Rev: 0.2  
Id: 17/20

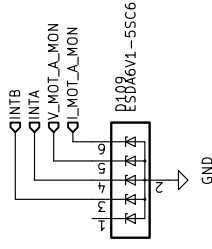
# Power monitoring



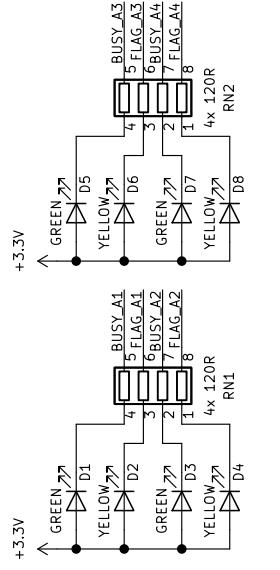
# GPIO expander



# ESD protection



# Stepper signals



VUT FIT - STRaDe

Sheet: /Motors\_A/

File: Motors\_A.sch

**Title: Eyrina V4**

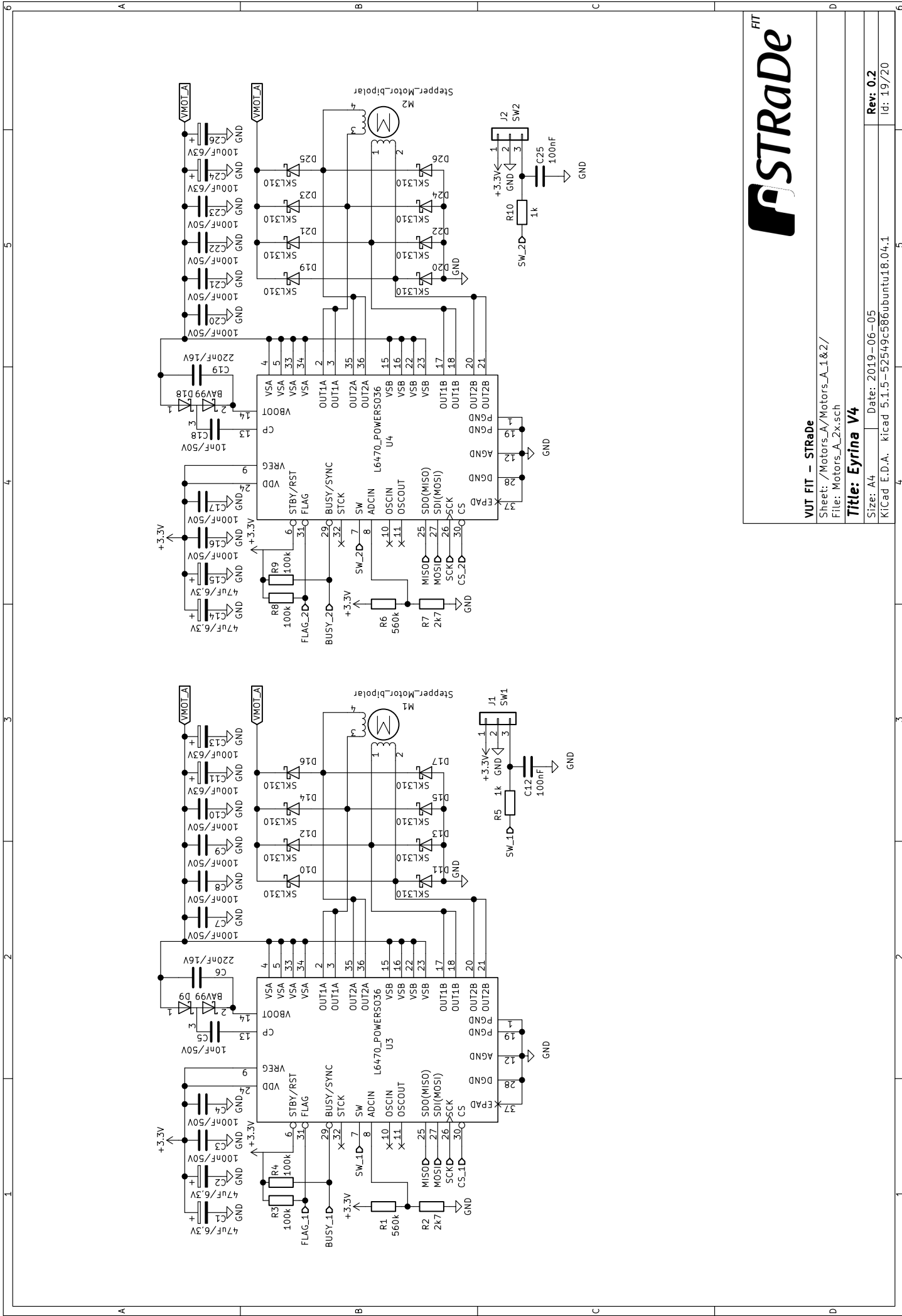
Size: A4

Date: 2019-06-05

KiCad E.D.A. kicad 5.1.5-52549c586buntu18.04.1

Rev: 0.2

Id: 18/20



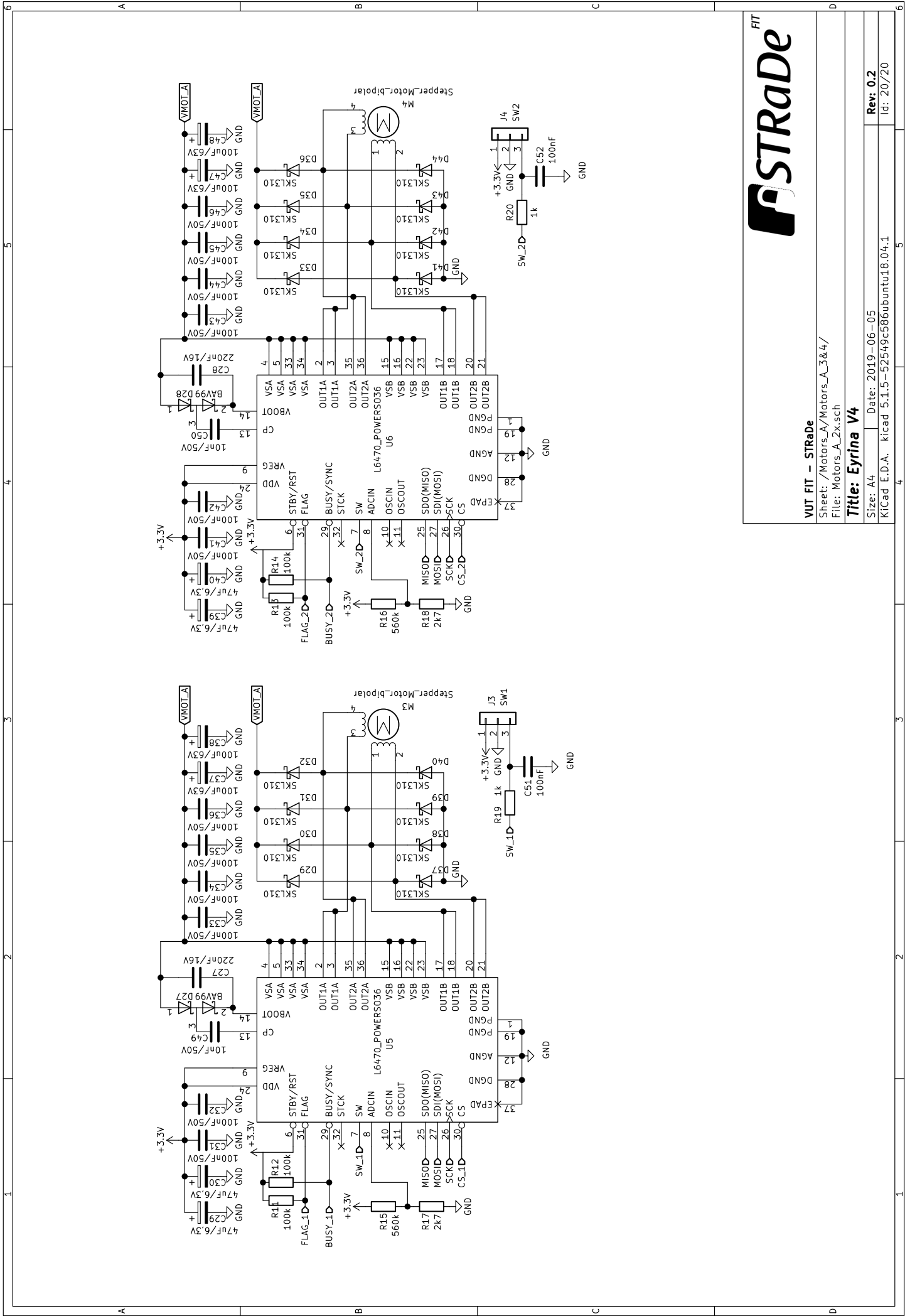
VUT FIT - STRaDe

Sheet: /Motors\_A/Motors\_A\_1&2/  
File: Motors\_A\_2x.sch

Title: Eyrina V4

Size: A4 Date: 2019-06-05 Rev: 0.2  
KiCad E.D.A. kicad 5.1.5-52549c586buntuu18.04.1 Id: 19/20





VUT FIT - STRaDe

Sheet: /Motors\_A/Motors\_A\_3&4/  
File: Motors\_A\_2x.sch

Title: Eyrina V4

Size: A4	Date: 2019-06-05	Rev: 0.2
KiCad E.D.A. kicad 5.1.5-52549c586buntbu18.04.1		Id: 20/20

## Příloha B

# Seznam použitých součástí

Seznam použitých součástí obsahuje reference vztahující se k označení součástí ve schématu v příloze **A**. Stejné referenční označení je použito v osazovacím plánu v příloze **C**.

Reference	Množství	Hodnota
C1 C2 C14 C15 C29 C30 C39 C40 C67 C68 C77 C78 C93 C94 C103 C104	16	47uF/6
C3 C4 C7 C8 C9 C10 C16 C17 C20 C21 C22 C23 C31 C32 C33 C34 C35 C36 C41 C42 C43 C44 C45 C46 C69 C70 C71 C72 C73 C74 C79 C80 C81 C82 C83 C84 C95 C96 C97 C98 C99 C100 C105 C106 C107 C108 C109 C110	48	100nF/50V
C5 C18 C49 C50 C87 C88 C113 C114	8	10nF/50V
C6 C19 C27 C28 C65 C66 C91 C92	8	220nF/16V
C11 C13 C24 C26 C37 C38 C47 C48 C75 C76 C85 C86 C101 C102 C111 C112	16	100uF/63V
C12 C25 C51 C52 C89 C90 C115 C116 C118 C119 C120 C121 C122 C123 C124 C125 C126 C127 C128 C129 C130 C131 C132 C133 C134 C136 C138 C139 C140 C141 C144 C145 C148 C151 C152 C153 C154 C155 C156 C157 C158	41	100nF
C53 C63	2	10uF 50V (ceramic)
C54 C57	2	300uF 16V
C55 C58	2	220nF
C56 C59	2	22nF
C60 C61 C62 C64 C117 C135 C137	7	4.7uF (X7R)
C142 C143 C146 C147 C149 C150	6	1uF (X7R)
D1 D3 D5 D7 D52 D53 D56 D57	8	GREEN
D2 D4 D6 D8 D54 D55 D58 D59	8	YELLOW
D9 D18 D27 D28 D60 D61 D78 D79	8	BAV99
D10 D11 D12 D13 D14 D15 D16 D17 D19 D20 D21 D22 D23 D24 D25 D26 D29 D30 D31 D32 D33 D34 D35 D36 D37 D38 D39 D40 D41 D42 D43 D44 D46 D49 D62 D63 D64 D65 D66 D67 D68 D69 D70 D71 D72 D73 D74 D75 D76 D77 D80 D81 D82 D83 D84 D85 D86 D87 D88 D89 D90 D91 D92 D93 D94 D95 D105 D106 D107	69	SKL310
D45	1	1SMB5925
D47	1	12V
D48 D51	2	STPS340U
D50	1	6V
D96	1	ASMB-MTBO-0A3A2
D97	1	STPS1L30A
D98 D99	2	BAT54S
D100 D101 D102 D103 D104	5	BAT54S
D108 D109 D110 D111 D112	5	ESDA6V1-5SC6
F1 F2	2	2A
F3	1	200mA
F4 F5 F6	3	50mA
FB1	1	100MHz@22R

H1 H2 H3 H4 H5 H6 H7 H8 H9	9	M3
J1 J3 J8 J10	4	SW1
J2 J4 J9 J11	4	SW2
J5	1	VMOT_A
J6	1	POWER_IN
J7	1	VMOT_B
J12	1	USB_B
J13	1	Debug connector
J14	1	T_CH1
J15	1	T_CH2
J16 J17 J18	3	LED_connector
J19	1	PROG_CONN_STM_6p
J20	1	OLED
L1 L2	2	15uH
L3 L4 L5	3	3.3mH@11.8R
M1 M2 M3 M4 M5 M6 M7 M8	8	Stepper_Motor_bipolar
M9 M10 M11 M12 M13	5	Fan_4pin
Q1 Q2 Q4	3	AOD4189
Q3 Q5	2	2N7002
Q6	1	2N7002
R1 R6 R15 R16 R40 R41 R50 R51	8	560k
R2 R7 R17 R18 R42 R43 R52 R53	8	2k7
R3 R4 R8 R9 R11 R12 R13 R14 R21 R36 R37 R38 R39 R46 R47 R48 R49 R91	18	100k
R5 R10 R19 R20 R44 R45 R54 R55 R64 R66 R68 R70 R72 R74 R76 R92	16	1k
R22 R32	2	82k
R23 R26	2	4.7k
R24 R27 R28 R29 R30 R33 R34 R35 R58 R59 R62 R63 R65 R67 R69 R71 R73 R75 R78 R85 R86 R89 R90 R93	24	10k
R25	1	30k
R31 R84 R88	3	180k
R56 R79 R81	3	300k
R57 R60 R61	3	22R
R77 R80 R82	3	4R
R83 R87	2	20k
R94	1	22R
RN1 RN2 RN3 RN4	4	4x 120R
RV1 RV2	2	20k
SW1	1	RESET
TP1	1	PWR_IN

TP2	1	36V
TP3	1	GND
TP4	1	12V
TP5	1	5V5
TP6	1	5V
TP7	1	3V3
TP8	1	T_CH1
TP9	1	T_CH2
TP10 TP11 TP12 TP13 TP14	5	FAN
TP15 TP17 TP19	3	LED_channel
TP16 TP18 TP20	3	PWM_input
TP21	1	I_MAIN
TP22	1	VMOT_B
TP23	1	IMOT_B
TP24	1	VMOT_A
TP25	1	IMOT_A
U1	1	STM32L452CEUx
U2 U11	2	MCP23017_ML
U3 U4 U5 U6 U12 U13 U14 U15	8	L6470_POWERSO36
U7 U8	2	L5973D
U9	1	NCP1117-5.0_SOT223
U10	1	NCP1117-3.3_SOT223
U16	1	USBLC6-2SC6
U17	1	EMC2305
U18	1	Si7055-A20
U19 U21 U22	3	NCL30160
U20	1	WS2818B
U23 U24 U25	3	ACS712xLCTR-20A

## Příloha C

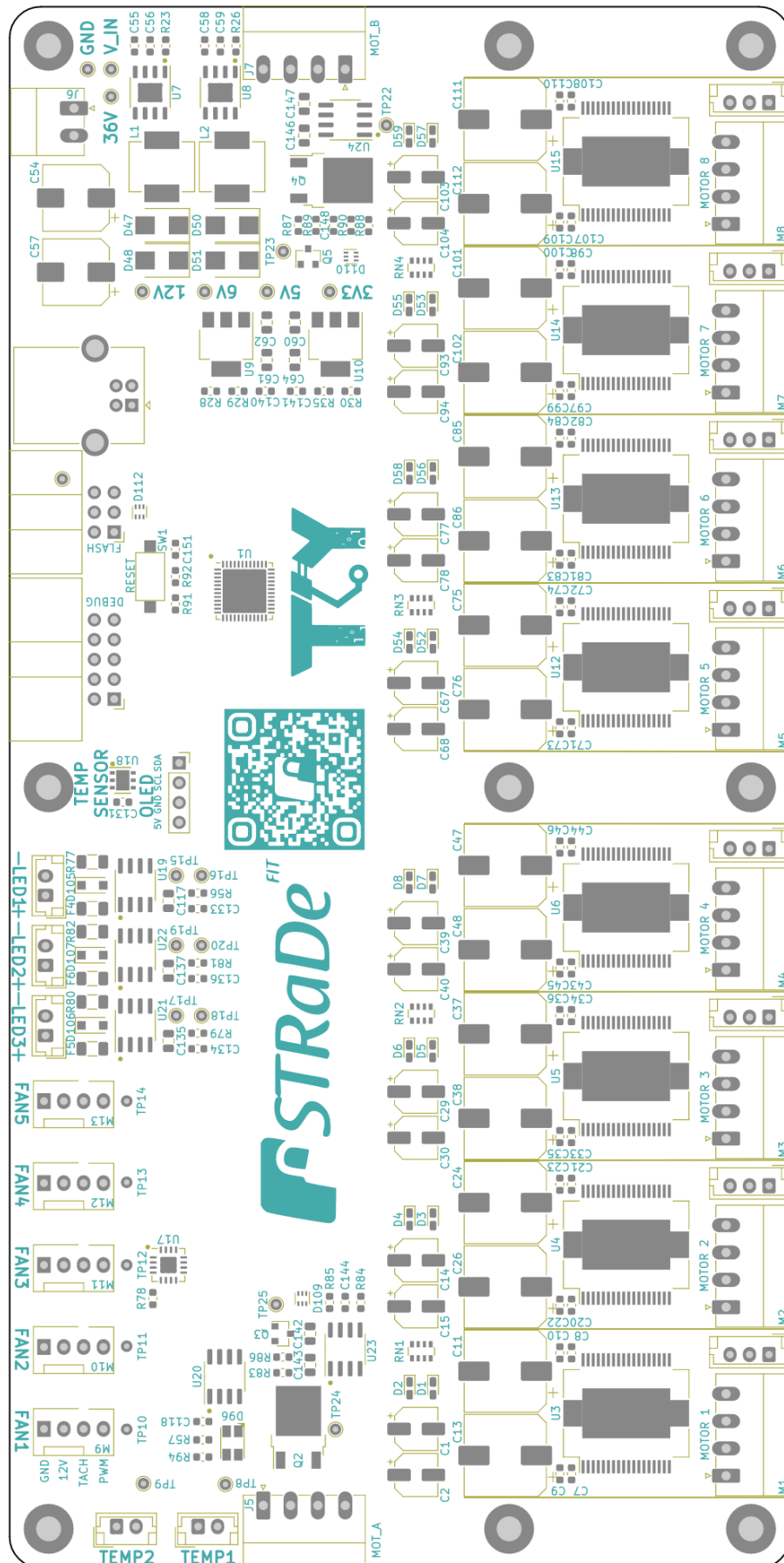
# Osazovací plán základní desky

Osazovací plán obsahuje kromě referenčních označení součástek a jejich ohraničení také vrstvu obsahující potisk. Vrstva s popisem jsou přidány kvůli snadnější orientaci. Referenční označení a vrstva potisku je vyznačena světle modrou barvou. Pájecí plochy jsou vyznačeny šedou a ohraničení plochy součástek béžovou. Hodnoty součástek vychází ze seznamu použitých součástek v příloze B.

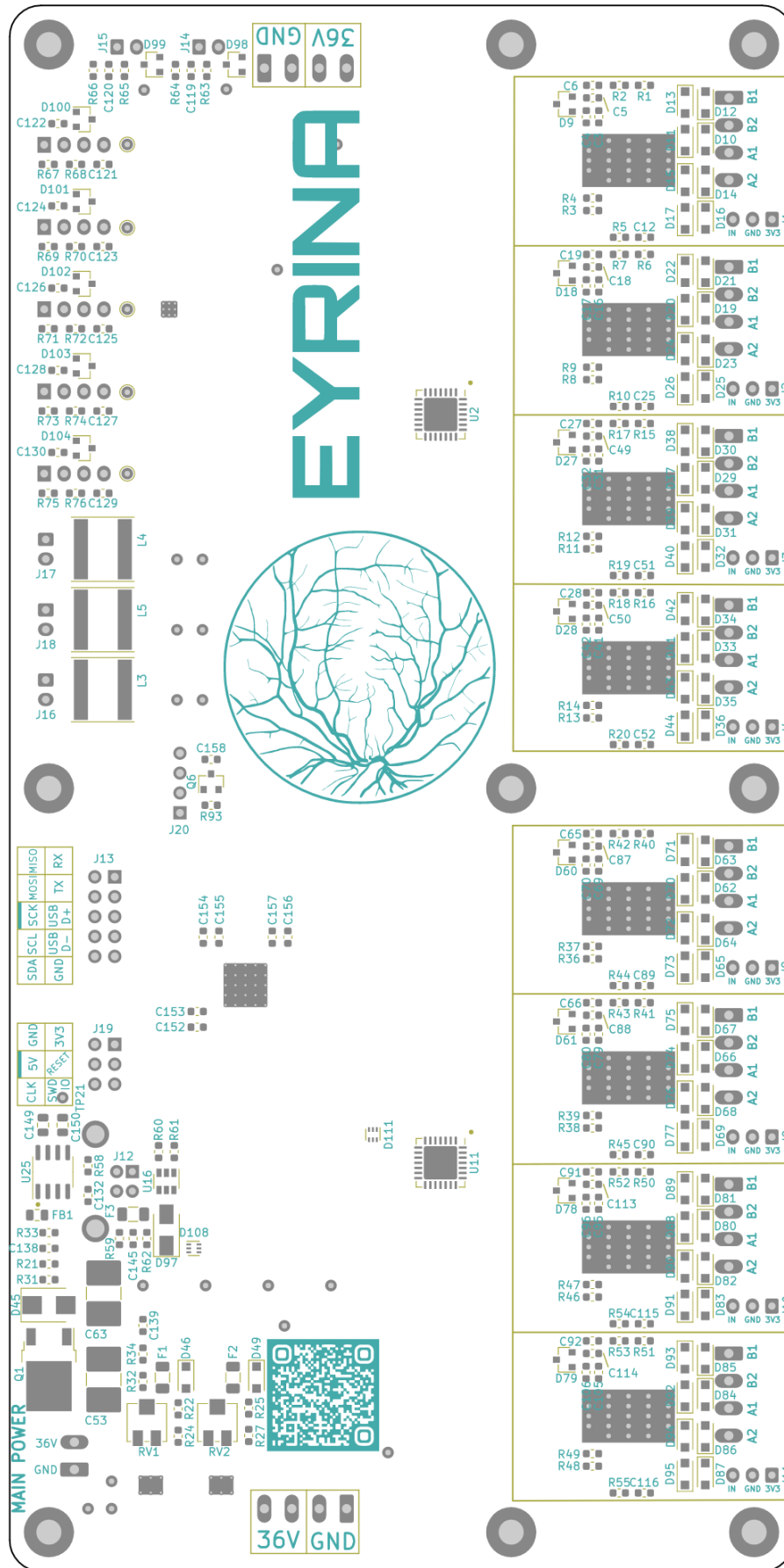
Obrázky jsou vygenerované z návrhového prostředí KiCAD. Jsou generovány pomocí rozšiřujícího pluginu Interactive Bill of Materials<sup>1</sup>. Ten kromě osazovacího plánu generuje také verzi zobrazitelnou ve webovém prohlížeči. V této verzi lze snadno vyhledávat pozice všech součástek a zrychlit tak osazení desky. Tato interaktivní verze osazovacího plánu je přiložena mezi digitálními souboru k této práci.

---

<sup>1</sup><https://github.com/openscopeproject/InteractiveHtmlBom>



Obrázek C.1: Osazovací plán základní desky, přední strana



Obrázek C.2: Osazovací plán základní desky, zadní strana