

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra řídicí techniky

Robotické skládání Rubikovy kostky

Hana Nováková

Vedoucí: Ing. Jiří Zemánek
Obor: Systémy a řízení
Studijní program: Kybernetika a robotika
Leden 2018

Poděkování

Děkuji ČVUT, že mi je tak dobrou *alma mater*. Mému vedoucímu za poskytnutí nezměrného množství rad a přístupů do laboratoří. A mým blízkým za psychickou podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně, a že jsem uvedla veškerou použitou literaturu.

V Praze, 9. ledna 2018

Abstrakt

Robotické skládání Rubikovy kostky je zajímavé téma pro výuku i prezentaci kombinací oborů robotiky, informatiky, elektrotechniky a strojírenství.

Tato práce se zabývá návrhem a stavbou takového robota, zaměřeného na jednoduchost konstrukce, cenovou dostupnost a dostupnost hardwarových i softwarových prvků. Je sestaven přehled existujících konstrukcí s jejich přednostmi a slabinami.

Na základě těchto znalostí je poté sestaven návrh mechanické a elektronické konstrukce. Konstrukce některých částí je vyřešena pomocí 3D tisku.

Práce se dále zabývá návrhem softwaru, především řešením úkolu počítačového vidění v algoritmu pro zjištění počátečního barevného uspořádání kostky. V této části je řešena detekce kostky v obraze a eliminace vlivu okolního světla na klasifikaci barev. Je využit klasifikátor nejbližšího souseda.

Standardní notace postupu řešení hlavolamu je převedena na signály ovládající mechanickou část robota. K samotnému výpočtu tahů vedoucích ke složení kostky je využito Kociembova dvoufázového algoritmu, který byl již několikrát součástí řešení, jež překonaly světový rekord pro nejrychlejší robotické skládání Rubikovy kostky.

Práce obsahuje návod na sestavení konstrukce i zdrojové kódy pro tisk, je k dispozici také obslužný software. S těmito prostředky lze robota sestavit.

Klíčová slova: robot, Rubikova kostka, Raspberry Pi, STM32F4, rozpoznávání obrazu

Vedoucí: Ing. Jiří Zemánek
ČVUT FEL, Katedra řídicí techniky,
Karlovo náměstí 13/E,
Praha 2

Abstract

Robotic Rubik's Cube solving is an interesting topic for educational and presentational purposes, as it combines robotics, informatics, electrotechnics and engineering.

This thesis is dealing with design and construction of such a robot, focusing on construction simplicity, affordability and availability of its hardware and software components. An overview of existing builds is presented along with their advantages and disadvantages.

Based on that knowledge, mechanical and electronic designs are created. Several parts are produced using a 3D printer.

The paper also focuses on robot's software design, mainly the computer vision problem of detecting colors from the cube at the beginning of the task. Localization of the cube and adaptation of the color classification algorithm to the unknown lighting conditions is discussed.

Another algorithm translates the standardized cube solution notation string into the signals controlling robot's mechanical parts. The solution string itself is obtained using Kociemba's two phase algorithm. This algorithm was used in several robots breaking the world time record in robotic Rubik's Cube solving.

A construction manual and source codes of the 3D printed parts are included as well as the control software. With these materials provided, one is able to construct the designed robot.

Keywords: robot, Rubik's cube, Raspberry Pi, STM32F4, computer vision

Title translation: Robotic Rubik's Cube Solver

Obsah

1 Úvod	1
2 Přehled existujících systémů skládajících Rubikovu kostku	3
2.1 Lego Mindstorms	3
2.1.1 MindCuber (NXT) a MindCub3r (EV3)	3
2.1.2 CubeStormer	5
2.1.3 Další roboti z Lega	6
2.2 Ruku	6
2.3 RuBot2	8
2.4 Robot vytisknutý převážně na 3D tiskárně	8
2.5 High Frequency Twister	9
2.6 Sub1 - aktuální světový rekord ..	9
3 Hardware a montáž	11
3.1 Výběr součástek	12
3.2 Modelování a tisk součástek	13
3.3 Elektronická část návrhu	16
3.4 Sestrojení robota	20
4 Software	25
4.1 Raspberry Pi	25
4.1.1 Rozpoznávání obrazu	25
4.1.2 Klasifikace naskenovaných barev	27
4.1.3 Výpočet tahů	27
4.1.4 Převedení oficiální notace na příkazy pro motory	29
4.2 Ovládání motorů mikroprocesorem	30
4.2.1 Použité periferie	30
4.2.2 Komunikace	32
4.2.3 Vypnutí a zapnutí motorů ...	34
4.2.4 Před prvním spuštěním	34
4.3 Uživatelský návod	34
5 Závěr	35
5.1 Současný stav	35
5.2 Možnosti rozšíření	36
A Řídící deska a konstrukce	37
B Obsah přiloženého CD	41
C Literatura	43
D Zadání práce	47

Obrázky

2.1 MindCub3r, zdroj [2]	5
2.2 CubeStormer 3, zdroj [3]	6
2.3 První Lego Mindstorms NXT robot skládající Rubikovu kostku, zdroj [8]	7
2.4 Ruku, zdroj [10]	7
2.5 RuBot 2, zdroj [11]	8
2.6 Robot vytisknutý na 3D tiskárně, zdroj [18]	9
2.7 HighFrequencyTwister, zdroj [13]	10
2.8 Sub1, zdroj [7]	10
3.1 Tvar rámu robota	11
3.2 Umístění kostky, motorů a kamery	12
3.3 Srovnání vnitřní stavby různých konstrukcí Rubikovy kostky	13
3.4 Model kolečka	14
3.5 Ojnice	14
3.6 Rameno posunu	15
3.7 Model držáku na spodní motor .	15
3.8 Propojka dolního motoru s ojnicí	15
3.9 Držák pro horní motory	16
3.10 Úchop na kostku	16
3.11 Blokové schéma zapojení elektrické části	17
3.12 Zapojení konektoru a ochrany proti přepólování zdroje	17
3.13 Zapojení spínaného regulátoru napětí	18
3.14 Zapojení DRV8825 řídicích modulů	19
3.15 Osazený plošný spoj	20
3.16 Uchycení dolního motoru	21
3.17 Konstrukce pro hodní motory .	22
3.18 Spojování součástek	22
3.19 Sestavená konstrukce robota . .	23
4.1 Fáze detekce barev	26
4.2 Pořadí bodů detekovaných na fotografii	26
4.3 Čítač s porovnávacím registrem .	32
4.4 Výstup z logického analyzátoru .	34
A.1 Schéma zapojení řídicích modulů a STM kitu	38
A.2 Plošný spoj	39

Tabulky

3.1 Vybrané součástky na stavbu robota.	12
3.2 Seznam vytisknutých součástek .	14
3.3 Seznam dalších použitých elektronických součástek	20
3.4 Délky profilu jednotlivých částí konstrukce	21
4.1 Přehled oficiální notace	29
4.2 Převod oficiální notace na příkazy pro STM32F4	29
4.3 Přeznačení tahů otočené kostky .	30
4.4 Význam přijatých znaků	33
A.1 Seznam potřebných elektronických součástek do DPS	37
A.2 Seznam propojovacích součástek	37



Kapitola 1

Úvod

Tato práce se zabývá hardwarovým i softwarovým návrhem robota, který je schopen řešit hlavolam - složit Rubikovu kostku. Při tom využívá kameru k určení počátečního stavu a klasifikaci barev kostky, mikropočítač Raspberry Pi pro výpočet tahů vedoucích k vyřešení hlavolamu a mikroprocesor k ovládání krokových motorů.

Aby bylo možné dosáhnout robotického skládání Rubikovy kostky, je potřeba se zaměřit na několik oblastí. První z nich je návrh konstrukce. A to jak bude vypadat, jaké komponenty je potřeba opatřit a jak je spojit dohromady. Další částí je výběr elektronických součástek a návrh jejich propojení. Je potřeba zjistit, v jakém stavu se kostka nachází a nalézt vhodné řešení této situace. Dále je potřeba vyřešit, jak pomocí robota fyzicky vykonat tahy, které byly vypočteny.

V kapitole 2 je uveden základní přehled existujících robotů, které řeší stejný problém v různých úrovních profesionality, složitosti robota, jeho rychlosti a cenové náročnosti.

Kapitola 3 obsahuje popis výroby hardwaru robota, který byl sestrojen v rámci této bakalářské práce. Obsahuje konstrukční návrh a postup sestavení. Dále pak elektronický návrh a jeho realizaci.

Ve 4. kapitole je pak popsán vytvořený a použitý software. Jde o skenování obrazu kamerou, rozpoznání stavu kostky, výpočty postupu složení a převod oficiální notace na příkazy pro motory. Tyto příkazy dále zpracovává mikroprocesor a informaci předává přes řídicí modul motorům.

Kapitola 2

Přehled existujících systémů skládajících Rubikovu kostku

Roboti jsou využíváni v podstatě v každém aspektu lidského života. Je možné je najít ve výrobě, v kuchyních, automatech na nápoje a jídlo nebo také mezi druhy zábavy. Robot skládající Rubikovu kostku je jeden z těch, který byl vytvořen pro zábavu. Takový systém je hojně využíván také ve výuce. Děti se na různých robotických systémech mohou naučit technickému myšlení a právě roboti skládající Rubikovu kostku jsou jedna z vhodných variant. Na takovém systému je možno ukázat základy programování, sestrojení mechanické konstrukce a na některých typech dokonce elementární principy elektrotechniky. Některé vývojářské týmy se dokonce snaží dosáhnout světových rekordů.

Robotů zabývajících se touto problematikou již bylo vytvořeno nepřehledné množství. Někteří skládají klasickou $3 \times 3 \times 3$ verzi tohoto hlavolamu, jiné zvládají i $7 \times 7 \times 7$, případně jiný tvar než krychli. V této kapitole jsou popsány nejvýrazněji rozšířené nebo nejrychlejší konstrukce zabývajících se skládáním Rubikovy kostky $3 \times 3 \times 3$, i další zajímavá řešení. Jsou zmíněny i někteří netypičtí roboti. Základní a však výrazně neúplný seznam lze najít na [20].

2.1 Lego Mindstorms

Jedním z výukových systémů jak pro děti tak i dospělé je robotická stavebnice Lego Mindstorms. Tato stavebnice nabízí mnoho využití a dává prostor dětské kreativitě.

2.1.1 MindCuber (NXT) a MindCub3r (EV3)

Na stránkách výrobce [1] je k dispozici návod na sestavení tohoto robota a také program pro jeho řídicí jednotku (tzv. inteligentní kostku robota). Tento kód si mohou zájemci stáhnout, pročíst a případně upravit.

Konstrukce je založena pouze na třech motorech (stavebnice jich více v základní výbavě nemá). Jeden motor je využit při skenování barev a tedy na skládání nemá vliv. Pro vykonání postupu, který vypočte inteligentní kostka tedy zbývají dva motory. Jeden natáčí kostku tak, aby strana, kterou

je potřeba pohnout, byla v "košíčku" ovládaném druhým motorem. Toto otočení může být kombinováno otáčením celé kostky druhým motorem a může být potřeba provést víckrát. Při samotném otočení strany je také potřeba, aby toto rameno kostku shora přidrželo.

Skenování stavu kostky je zajištěno pomocí světelného senzoru. Tento senzor vrací pouze intenzitu odraženého světla (ve stupních šedi) a je potřeba jej pro každou barvu zkalibrovat. Bohužel senzor zvládne v jedné pozici naskenovat pouze jedno políčko kostky a proto skenování celé kostky trvá dlouho.

U stavebnice NXT je dalším výrazným zpomalením výpočetní výkon inteligentní kostky a její malá paměť. Výpočet tahů trvá relativně dlouho a tahy nejsou optimalizované. Postup se v průměru skládá z většího počtu tahů než u novější verze stavebnice, která zvládá alespoň základní optimalizaci.

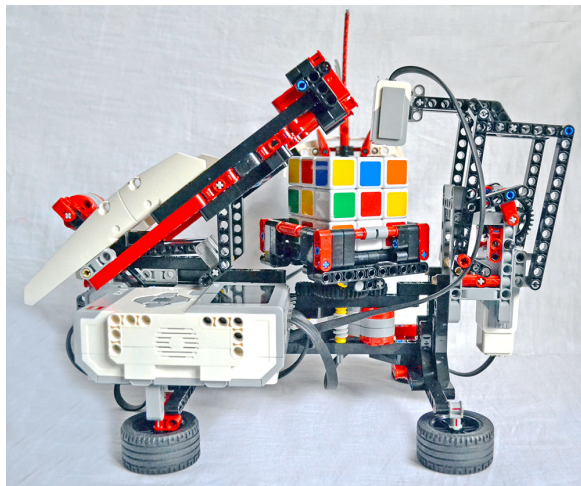
■ Výhody

- Relativně jednoduchá konstrukce
- Stavebnice obsahuje všechny potřebné součástky
- Existující návod
- Dostupnost funkčního kódu

■ Nevýhody

- Finanční náklady (pořízení stavebnice je přijatelné pro školu, pro domácnost jde však o větší výdaj)
- Problém s různým osvětlením, při skenování barev
- Pomalé skenování barev
- Relativně pomalá aplikace postupu (způsobeno konstrukcí)
- Náchylnost na mechanickou chybovost (ramenu pro překlopení kostky se někdy tato operace nepovede)
- Neexistence zpětné kontroly správného dokončení postupu (předpokládá, že aplikace postupu proběhla vždy správně)

K nové verzi stavebnice (EV3) je možno zakoupit také snímač barev, dosahuje tedy menší chybovosti klasifikace rozmístění kostky než u verze předchozí. Stále ale platí, že skenování trvá relativně dlouhou dobu z důvodu skenování každého políčka zvlášť. Na obrázku 2.1 je zobrazen tento typ robota ze stavebnice EV3. Konstrukci vytvořil David Gilday.



Obrázek 2.1: MindCub3r, zdroj [2]

■ 2.1.2 CubeStormer

Ze stejné stavebnice byl sestaven také CubeStormer. V roce 2011 tento robot získal světový rekord ve své první podobě. Poslední verzí je CubeStormer 3, který pokořil světový rekord časem 3,253 s a to v květnu 2014 [4].

CubeStormer byl sestaven z více stavebnic a má tedy k dispozici osm inteligentních kostek a větší počet motorů než MindCuber. Poslední model tohoto robota byl postaven ze stavebnice EV3. Na skenování stavu kostky však nepoužívá senzory ze stavebnice Lega, ale chytrý telefon Samsung Galaxy S4 s osmijádrovým procesorem. Využívá nejen jeho kamery pro skenování, ale také jeho výpočetního výkonu pro sestavení postupu. Motory jsou ovládány za pomoci inteligentních kostek.

Tvůrci David Gilday a Mike Dobson na poslední verzi robota pracovali ve svém volném čase 18 měsíců.

- Výhody
 - Rychlost
- Nevýhody
 - Není volně dostupná konstrukce
 - Cena za osm stavebnic Lego Mindstorms je pro většinu lidí příliš vysoká
 - Potřeba dostatečně výkonného chytrého telefonu

V tomto případě však nejde o možnost si robota sestavit doma, ale o vývoj robota, který pokoří světový rekord, a nevýhody jsou tedy irelevantní. Tohoto robota je možné vidět na obrázku 2.2.



Obrázek 2.2: CubeStormer 3, zdroj [3]

2.1.3 Další roboti z Lega

Ze stavebnice Lego byly vytvořeny různé podomácku vyrobené konstrukce. Jde jak o úpravy verze MindCuber, tak o úplně nové konstrukce.

Stejný tvůrce, který pracoval na obou výše zmíněných robotech vytvořil také MicroCuber, který složí kostku $3 \times 3 \times 3$ o velikosti hrany 12 mm. Vytvořil také robota, který vyřeší kostku $4 \times 4 \times 4$ a MultiCuber, který skládá kostku $7 \times 7 \times 7$. Složení tohoto rozšířeného hlavolamu mu trvá 40 minut.

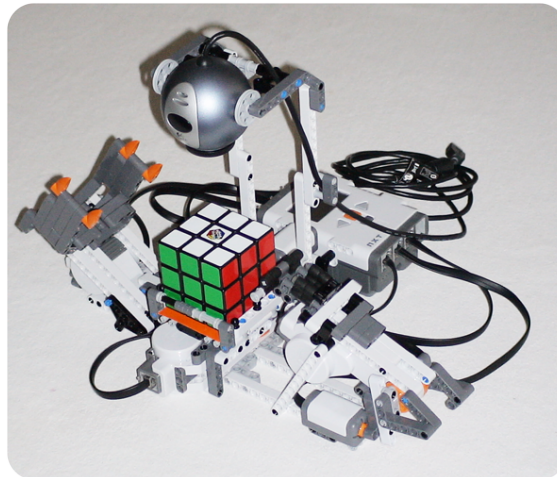
David Gilday se zabýval také stavbou robota, který vyřeší obdobu Rubikovy kostky v netypickém tvaru. Vytvořil tak konstrukci, která byla schopná složit hlavolam Megaminx s 12-ti barvami.

Vůbec prvního robota řešící klasickou Rubikovu kostku vytvořil Daniele Benedettelli. Byl nazván Lego Rubik Utopia a vytvořen již v roce 2007. Výpočet postupu prováděl notebook připojený pomocí USB. Skenování barev bylo zajištěno kamerou. Tohoto robota je možné vidět na obrázku 2.3.

2.2 Ruku

Další robot pro výuku byl zamýšlen pro STEM vzdělávání (věda, technika, strojírenství a matematika). Tento projekt zamýšleli tvůrci dostat do škol, a tak byl na přelomu roku 2014/2015 prezentován na portálu kickstarter [9], kde tvůrci prodávali prvních pár kusů, a zvýšením produkce chtěli dosáhnout snížení nákladů na výrobu a tím zlepšit dostupnost pro školy. Cílem takového robota ve školách by měla být výuka programování, robotiky, matematiky a informatiky zábavným způsobem. Měl probudit přirozenou dětskou zvědavost.

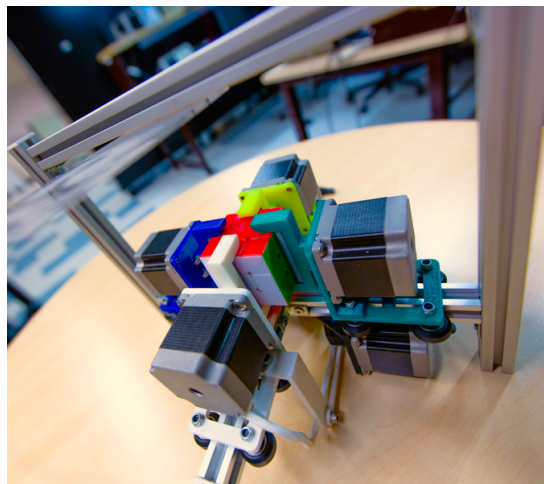
Ruku byl navržen jako open source a open hardware. Robot sestavený v rámci této bakalářské práce se jím výrazně inspiruje. Konstrukce je složena z hliníkového profilu a součástek vytisknutých na 3D tiskárně. Využívá chytrého telefonu a Raspberry Pi. Telefon plní funkci kamery pro určení stavu



Obrázek 2.3: První Lego Mindstorms NXT robot skládající Rubikovu kostku, zdroj [8]

kostky a výpočetní jednotky. S Raspberry Pi komunikuje pomocí WiFi. Deska plošného spoje obsahuje univerzální řídicí modul a šest programovatelných driverů pro krokové motory.

Samotný výpočet provádí aplikace pro Android, kterou lze využívat i bez robota. Po naskenování stavu kostky aplikace ukazuje jednotlivé kroky. Lze dokonce použít i v případě, že není k dispozici Rubikova kostka. Aplikace je totiž ve 3D a postupy se dají učit i pouze na telefonu. Fotka je na obrázku 2.4.



Obrázek 2.4: Ruku, zdroj [10]

2.3 RuBot2

Tento robot v roce 2010 nastavil světový rekord 37,5s. Stvořil jej Pete Redmond. Zajímavá je hlavně konstrukce, která je vysoce inspirovaná člověkem. Přečíst si o něm je možné v článku [12] a je také zmíněn v [19].

Robot má dvě web-kamery s vysokým rozlišením, které jsou umístěny jako oči. Navíc při hledání řešení nepoužívá jen pevně daný algoritmus, ale umělou inteligenci s neuronovým klasifikačním softwarem. Dále má dvě ruce, které se podobají lidským a jsou ovládány pneumaticky. Konstrukce je zachycena na obrázku 2.5.

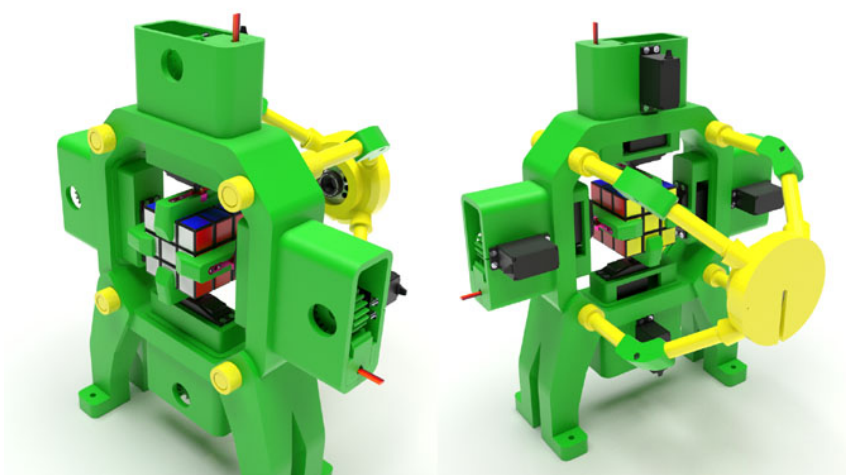


Obrázek 2.5: RuBot 2, zdroj [11]

2.4 Robot vytisknutý převážně na 3D tiskárně

Tento robot je tvořen převážně z tištěných součástí. Tisk dohromady trvá přes 67 hodin a je potřeba téměř 0,9 kg materiálu. Dále je potřeba k tomuto stroji připojit Raspberry Pi. Pro naskenování stavu kostky je využita web-kamera, na kterou je vytvořen speciální držák. Jako napájení slouží zdroj na 6 V / 3 A. Využívá čtyři servomotory pro ovládání úchopů kostky a další čtyři k ovládání uchopení kostky. Dále vyžaduje kontrolér k těmto motorům. Robot je popsán v článku [16] a [17].

Veškeré modely pro tisk a zdrojové kódy jsou volně dostupné pro nekomerční použití [18]. Na stránkách výrobce je podrobný návod, jak robota z vytisknutých součástí sestavit. Jde o zajímavý způsob skládání kostky pro domácí podmínky. Jeho tisk však trvá hodně dlouho a je potřeba mít k dispozici dostatek materiálu. Finální vzhled je na obrázku 2.6.



Obrázek 2.6: Robot vytisknutý na 3D tiskárně, zdroj [18]

2.5 High Frequency Twister

Tvůrci tohoto robota jsou Jau Flatland a Paul Rose. Robot na malou chvíli držel světový rekord a to s časem 0,9 sekundy. Jde o převážně tisknutou konstrukci, která obsahuje šest krokových motorů. Rozdíl oproti modelu popsanému výše je v uchopení kostky. Kostka má vyvrtány díry ve středech stran a motory jsou upevněny přímo do tohoto středu, což znamená, že není potřeba kostku otáčet či předávat. Tím dojde k výraznému zrychlení oproti předchozím modelům.

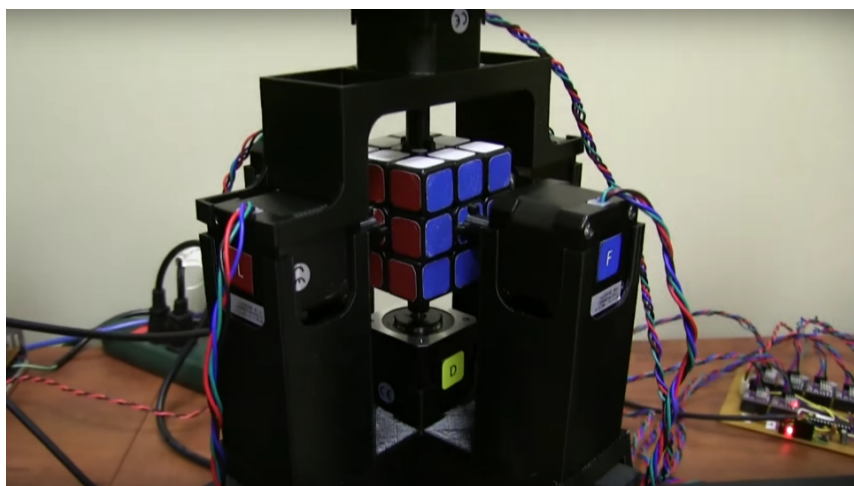
Pro správu řídicích modulů k motorům je využito Arduino. Kód, který se stará o výpočet tahů, je naprogramován v C++ a nahrán do vývojové desky ArduinoUno.

Začátkem roku 2017 byl tento projekt uvolněn pod licencí GPL. Je možné si stáhnout repozitář z GitHubu [14], kde je zveřejněn celý projekt. Nachází se tam také 3D modely součástek vytvořené v programu CAD, které jsou pro sestavení tohoto robota potřeba. Celá jeho konstrukce je na obrázku 2.7. Článek je k dispozici na [15].

2.6 Sub1 - aktuální světový rekord

Tento robot, vyvinutý německou firmou Infineon, překonal světový rekord poprvé začátkem roku 2016 v čase 0,887 sekundy. Jeho upravená verze pak ještě v téže roce rekord posunula na 0,637 sekundy. Úpravou se zabýval Albert Beer a byla nazývána "Sub1 Reloaded". O robotu bylo napsáno spoustu článků do internetových novin a to například [5], [6], je také zmíněn na Rubik's Cube Wiki [19].

Stejně jako v předchozím případě, i tato konstrukce potřebuje speciálně upravenou kostku, která má vyvrtány otvory do jejích středů pro upevnění hřídelí motorů. Potvrzení světového rekordu přišlo až několik měsíců poté,

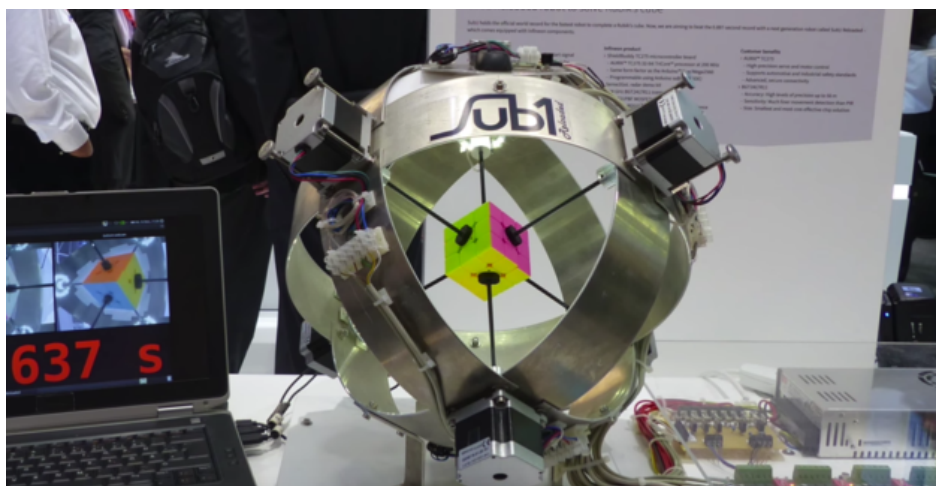


Obrázek 2.7: HighFrequencyTwister, zdroj [13]

co jej bylo dosaženo. Guinnessova kniha rekordů totiž potřebovala ověřit, že úpravy kostky jsou v souladu s celosvětovými standardy.

Robot komunikuje s notebookem. Počítač obdrží pouze dvě fotografie kostky, ze kterých je schopen určit její rozložení. Poté využije dvoufázového algoritmu pro hledání řešení, které je pomocí "Arduino-compatible Infineon AURIX microcontroller board" aplikováno na šesti krokových motorech.

Konstrukce je kulovitěho tvaru a uvnitř osvětlena LED diodami, pro jednodušší rozpoznávání obrazu. Konstrukci je vidět na obrázku 2.8.

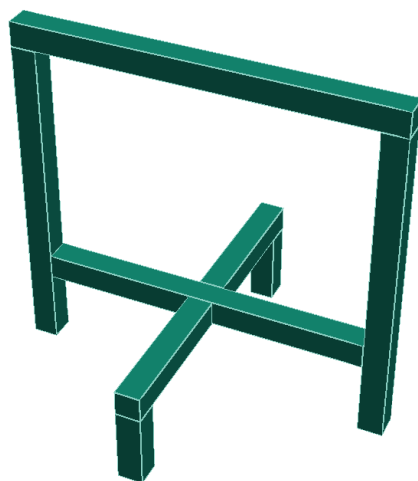


Obrázek 2.8: Sub1, zdroj [7]

Kapitola 3

Hardware a montáž

Robot sestavený v rámci této práce vychází z konstrukce Ruku robota [9] popsaného v kapitole 2.2. Konstrukce je upevněna k rámu (nákres viz. obrázek 3.1), který je složen z 20 mm hliníkových profilů. Ve středu tohoto rámu je vložena Rubikova kostka, na jejíž stranách jsou čtyři motory s jednoduchými rameny, které jsou schopny uchopit jednu stranu kostky a otočit s ní. Kostka je v každém okamžiku uchopena minimálně ze dvou protilehlých stran. Na obrázku 3.2a je znázorněna kostka světle zelenou barvou a motory tmavě zelenou. V dolní části konstrukce se pak nachází další dva motory, které přikládají nebo oddalují horní motory. Na obrázku 3.2b jsou znázorněny tmavě zelenou barvou. Každý spodní motor posouvá vždy dva protilehlé horní motory. Tímto způsobem je možné "předávat" kostku z jedné dvojice motorů do druhé, a také při otáčení strany přidržet prostřední vrstvu kostky na místě.

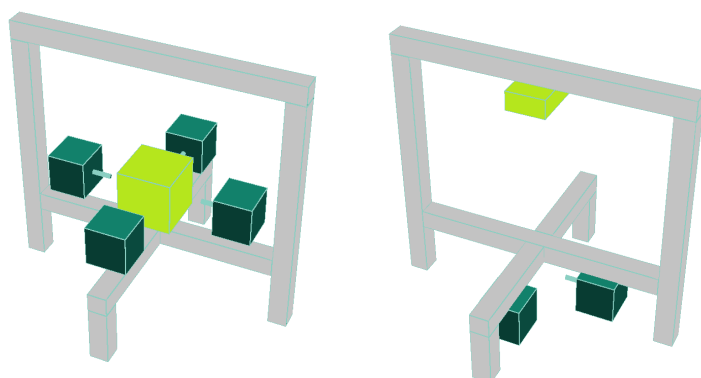


Obrázek 3.1: Tvar rámu robota

K manipulaci s kostkou dochází pomocí ramen na horních motorech, a to pouze ze čtyř stran. Pro manipulaci se zbylými stranami je potřeba vždy celou kostku otočit o 90 stupňů.

Rám robota je složen také z profilu, který vede přímo nad kostkou, což je vhodné pro umístění kamery, která snímá počáteční stav hlavolamu. Na ob-

rázku 3.2b je znázorněna světle zelenou barvou.



(a) : Umístění horních motorů a kostky

(b) : Umístění dolních motorů a kamery

Obrázek 3.2: Umístění kostky, motorů a kamery

3.1 Výběr součástek

Nejdříve bylo potřeba vybrat vhodné součástky, které se nedají vytisknout na 3D tiskárně. Jednalo se o motory a jejich řídicí moduly, profil na konstrukci rámu a Rubikovu kostku. Vzhledem k omezenému času bylo zapotřebí tyto části kupovat v České Republice a nikoli v zahraničí. Nakonec byly vybrány součástky sepsané v tabulce 3.1.

Součástka	Vybraný typ	Počet [ks]
Motor	SX17-1005LQCEF (Nema 17)	6
Řídicí modul krokového motoru	DRV8825	6
Hliníkový profil	X PROFIL 20x20 2 m	1
Záslepky pro hliníkový profil	X PROFIL záslepka 20x20	8
Rubikova kostka	Dayan V Zhanchi (speed cube 3×3×3)	1
Ložiska	ložisko MR 93 ZZ (3 × 9 × 4 mm)	32

Tabulka 3.1: Vybrané součástky na stavbu robota.

Při výběru motorů bylo potřeba se zaměřit na jejich krouticí moment a jmenovitý proud. Pro otočení kostky postačuje relativně nepatrná síla. Pro dolní motory však bylo potřeba zvolit dostatečně velký krouticí moment. Pro zjednodušení nakonec byly vybrány totožné motory, v případě potřeby

je ale možné horní motory nahradit slabšími. Vybraný typ motoru dosahuje krouticího momentu 0,5 Nm. V tomto konkrétním případě je maximální proud procházející motorem stanoven výrobcem na 1 A.

K omezování vstupních proudů do motoru a ke spínání cívek se správnou polaritou jsou využity DRV8825 moduly. V tomto případě byl vybrán modul používaný v 3D tiskárnách vzhledem k tomu, že motory byly vybrány stejného typu jako v konstrukci těchto tiskáren.

Pro robotické skládání je důležitá i volba kostky. Levné kusy jsou jednodušší konstrukce, která je velmi náchylná na přesnost otočení stran, a zasekává se. Aplikací síly v takto zaseklém stavu dochází k rozložení kostky na díly.

K těmto problémům dochází i při běžném řešení hlavolamu lidmi. Pro závodní účely vznikly varianty kostek, kterým se říká Speed cube. Kostky mají kulatou vnitřní konstrukci, a není jednoduché je neúmyslně vychýlit tak, aby se rozpadly. Konstrukce má také zaoblené rohy jednotlivých součástek, které kostku při nedokonalém otočení samy dorovnají při její další manipulaci.

Síla potřebná pro otočení strany je také mnohem nižší než u původní konstrukce. Na obrázku 3.3 jsou vidět rozložené obě konstrukce pro znázornění hlavních rozdílů mezi nimi.



(a) : Klasická konstrukce [21] (b) : Speed cube konstrukce [22]

Obrázek 3.3: Srovnání vnitřní stavby různých konstrukcí Rubikovy kostky

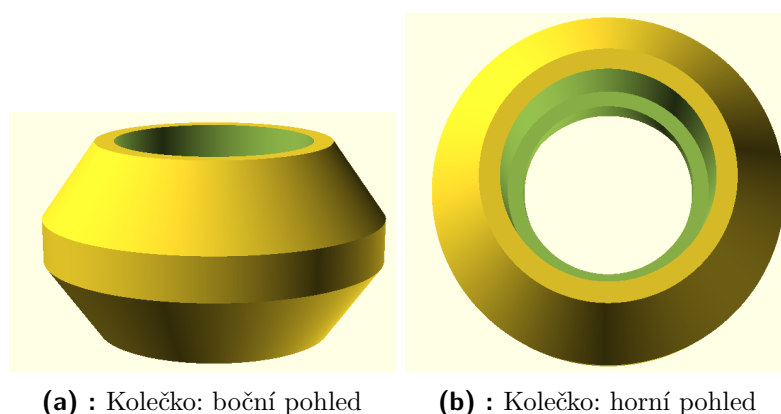
3.2 Modelování a tisk součástek

Ostatní součástky pro konstrukci robota byly vytištěny na 3D tiskárně. K tomu byl využit program OpenSCAD a následně vygenerován G-code v programu Cura a ten nahrán do systému Octoprint. Seznam a počet výtisků těchto součástek je uveden v tabulce 3.2. Zdrojové kódy k modelům i softwaru je možné nalézt na <https://github.com/novakha5/Robotic-Rubik-s-Cube-Solver>.

Na obrázku 3.4 je zachycen model kolečka. Kolečka slouží k upevnění motorů k hliníkovému profilu tak, aby se mohl volně pohybovat v jedné ose a tím se oddalovat či přibližovat ke kostce. Jeden motor k tomuto upevnění potřebuje čtyři kolečka. Po vytisknutí tohoto modelu bylo z každé strany vloženo ložisko. Středem byl následně protažen šroub M3 délky 20 mm a na jeho konci připevněn držák druhého motoru. Pod tímto držákem byla umístěna matka s pružnou podložkou a podložkou pod šroub.

Součástka (Název)	Počet [ks]
Držák horního motoru	4
Držák spodního motoru	2
Úchop kostky	4
Propojka ojnice s dolním motorem	2
Rameno posunu	4
Ojnice	4
Kolečko	16

Tabulka 3.2: Seznam vytisknutých součástek



(a) : Kolečko: boční pohled (b) : Kolečko: horní pohled

Obrázek 3.4: Model kolečka

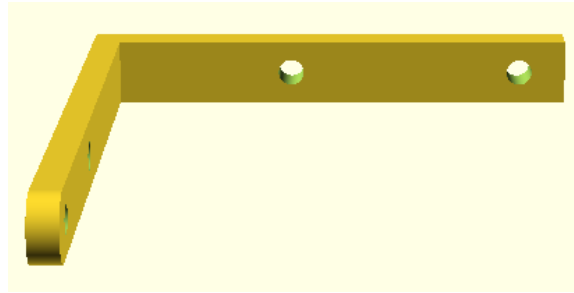
Obrázek 3.5 zachycuje model ojnice, která slouží k propojení horních motorů s motorem dolním, který s nimi pohybuje. Tato část zajišťuje pohyb osy pro uchopení nebo uvolnění kostky. Součástky byly propojeny pomocí 3 mm nýtu a závlačky příslušné velikosti.



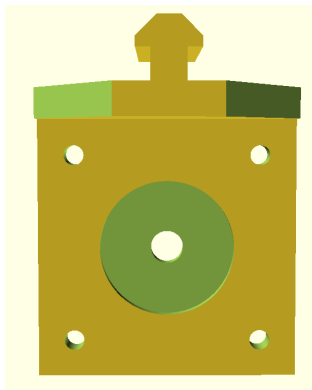
Obrázek 3.5: Ojnice

Dále je možné na obrázku 3.6 vidět zobrazení ramene posunu, které bylo následně přišroubováno k držáku na horní motor. Součástka je připojena z druhé strany ojnice stejným způsobem, jako bylo popsáno výše.

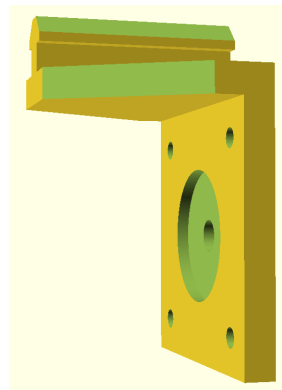
Model na obrázku 3.7 zobrazuje držák pro spodní motory. Hřídel byla protažena otvorem uprostřed a motor přišroubován šrouby velikosti M3 k tomuto držáku. Držák byl poté nasunut do profilu ze spodní strany a před ním i za ním byla umístěna gumová zarážka k zamezení lineárního pohybu.



Obrázek 3.6: Rameno posunu



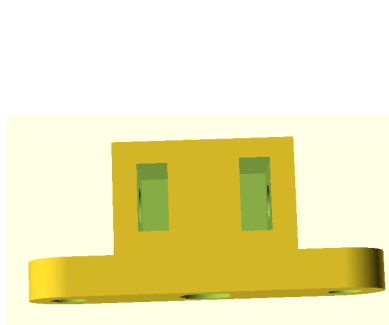
(a) : Držák spodního motoru: pohled ze předu



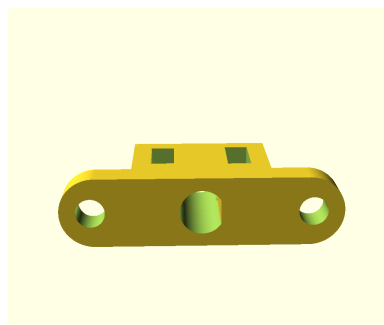
(b) : Držák spodního motoru: boční pohled

Obrázek 3.7: Model držáku na spodní motor

Výše zmíněnou propojku motoru s ojnicí je možno vidět na obrázku 3.8. Dva otvory po stranách modelu jsou určeny k připevnění ojnice. Otvor procházející středem je seříznutý tak, aby kopíroval zářez na hřídeli motoru. Upevnění je také pojištěno šroubky velikosti M3, které na místě drží matky vložené do podlouhlých otvorů



(a) : Propojka dolního motoru s ojnicí: pohled shora

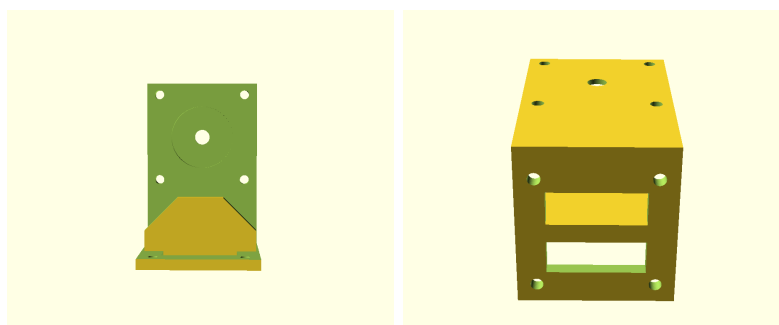


(b) : Propojka dolního motoru s ojnicí: pohled zepředu

Obrázek 3.8: Propojka dolního motoru s ojnicí

Pro horní motory bylo potřeba vytvořit držáky, které dostatečně vyzdvihují

motor, aby v prostoru mezi hřídelí motoru a hliníkovým profilem bylo možné otočit kostku bez kontaktu s konstrukcí. Také bylo potřeba, aby se pod motor daly umístit šrouby na připevnění koleček. Model tohoto držáku je znázorněn na obrázku 3.9. Uchyacení motoru a prostor pro hřídel jsou řešeny stejným způsobem jako u spodních motorů.

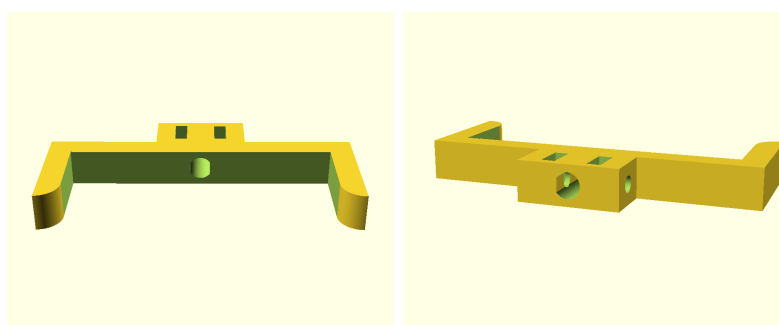


(a) : Držák pro horní motory:
pohled ze předu

(b) : Držák pro horní motory:
pohled zdola

Obrázek 3.9: Držák pro horní motory

Poslední tisknutou součástí byl úchop pro manipulaci s kostkou. Tato součástka byla k horním motorům připevněna stejným způsobem jako spojka dolního motoru s ojnící. Rozpětí úchopu má velikost kostky a šíře úchopu je 5 mm. Vnitřní povrch byl vystlán měkkým gumovým materiálem, pro jemnější zacházení s kostkou a zlepšení přitlaku motorů při manipulaci s ní. Tento úchop je možné vidět na obrázku 3.10.



(a) : Úchop kostky: pohled
z přední strany

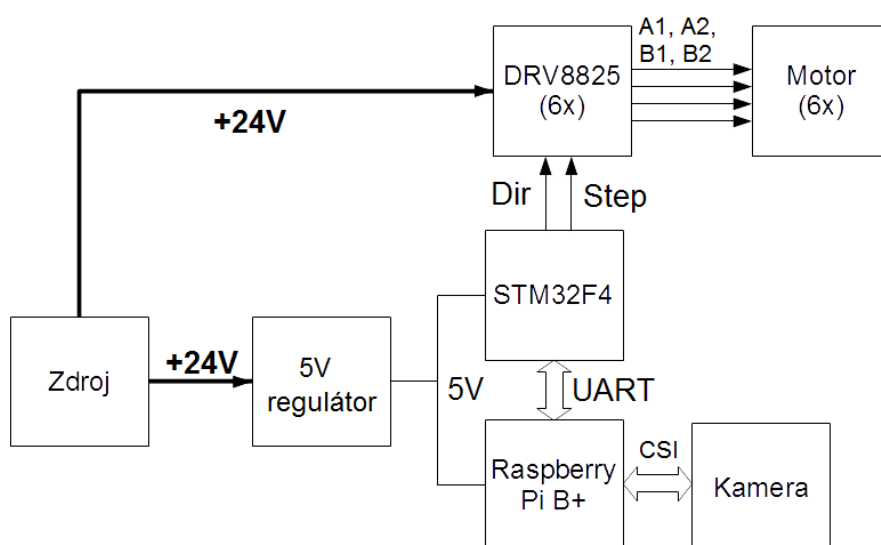
(b) : Úchop kostky: pohled
ze zadní strany

Obrázek 3.10: Úchop na kostku

3.3 Elektronická část návrhu

Další důležitou částí hardwarového návrhu je elektronika řízení krokových motorů. Blokové schéma zapojení je vidět na obrázku 3.11 a je řešeno plošným spojem. Seznam použitých součástek je uveden v příloze, v tabulce A.1.

Elektronické schéma (viz obrázek A.1) bylo nakresleno v programu Eagle

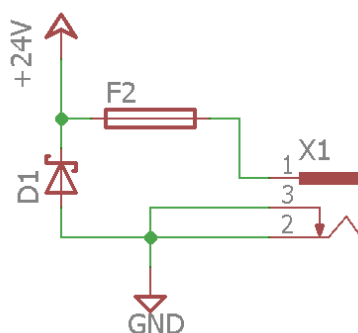


Obrázek 3.11: Blokové schéma zapojení elektrické části

Layout Editor 7.7.0. V témže programu byl sestaven i návrh plošného spoje. Jednotlivé části zapojení jsou popsány níže.

Napájecí část je odvozena od potřeb motorů a jejich budičů. Zvolené moduly DRV8825 pracují při vstupních napětích 8,2-45 V. Na základě diskuze s vedoucím práce bylo zvoleno preferované vstupní napětí 24V, a to kvůli svižnější reakci motorů v porovnání s nižšími napětími. Zapojení je možné napájet i nižším napětím, pokud je dodán dostatečný proud.

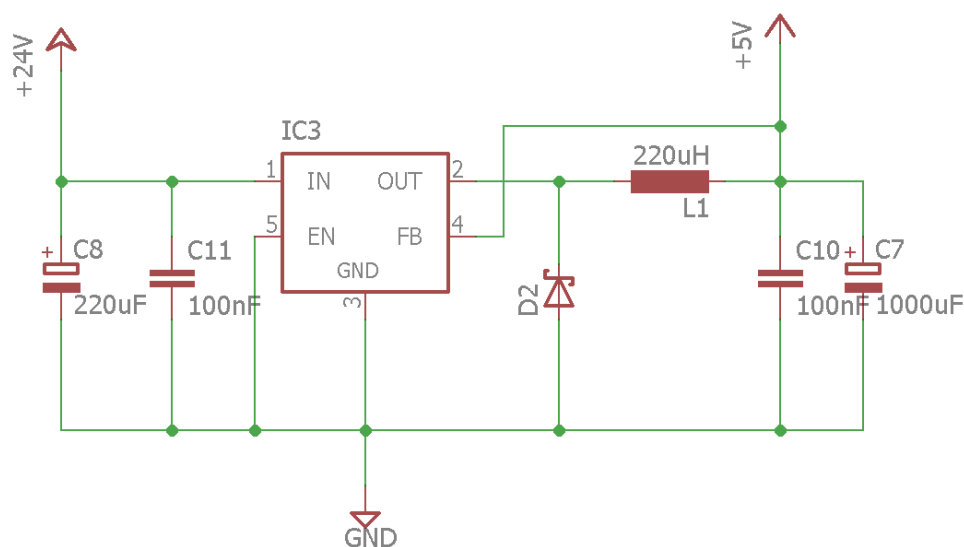
Na obrázku 3.12 je vidět ošetření vstupu napájení. Deska používá běžný souosý napájecí konektor, vnitřní kontakt je +24V, vnější kontakt GND. Pojistka F2 chrání zdroj před přetížením a zároveň před zapojením zdroje s chybnou polaritou. Dioda D1 v takovém případě krátkodobě zkratuje zdroj, což způsobí přepálení pojistky.



Obrázek 3.12: Zapojení konektoru a ochrany proti přepólování zdroje

Pro napájení mikroprocesoru a Raspberry Pi je potřeba 5 V. V této konkrétní konstrukci byl použit Raspberry Pi model B+, který má doporučený

proud dodaný zdrojem 1,8 A, avšak typická spotřeba samotného kitu bez přídatných modulů je 330 mA. Bylo tedy potřeba získat 5 V napětí pro napájení výše zmíněných částí obvodu. To bylo dosaženo pomocí spínaného regulátoru napětí 5 V / 3 A. Lineární regulátor napětí by byl v tomto případě nedostačující, neboť ztrátový výkon by převyšoval $(24 - 5) \text{ V} \cdot 0,330 \text{ A} = 6,27 \text{ W}$, což by bylo tepelně neúnosné. V zapojení součástky bylo potřeba umístit vhodné kondenzátory, cívku a diodu. Toto zapojení je vidět na obrázku 3.13. Výběr velikosti kondenzátorů a cívky vychází z dokumentace k spínanému regulátoru napětí [24]. Při zapojení těchto součástek je důležité umístit kondenzátory co nejbližší, aby vyrovnávaly napěťové výkyvy a spínání zdroje způsobovalo pouze minimální rušení. Schottkyho dioda D2 pracuje v okamžiku, kdy je odpojeno napětí 24V od cívky L1, což způsobí prudký pokles napětí na jednom konci cívky.



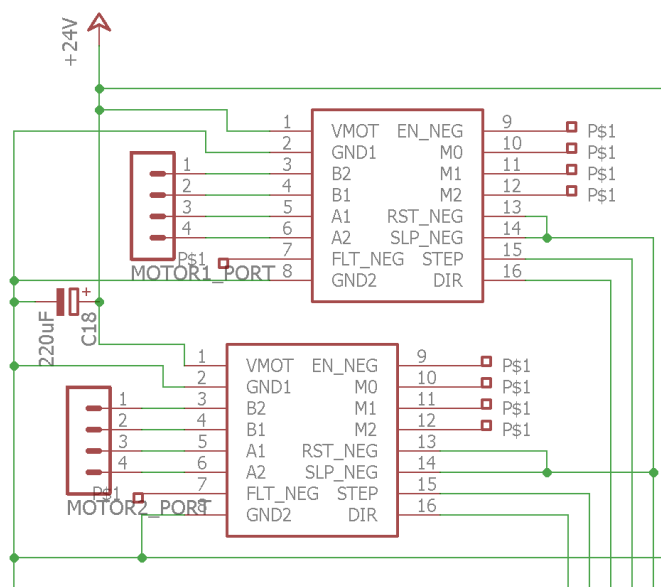
Obrázek 3.13: Zapojení spínaného regulátoru napětí

Pro připojení řídicích modulů k motorům byly vyvedeny piny A1, A2, B1, B2. Na tyto piny jsou pak připojeny kabely od motorů, a to podle dokumentace [23]. Napájení motorů V_{mot} je dodáváno přímo ze zdroje 24 V. Mezi tímto napájením a zemí jsou umístěny elektrolytické kondenzátory velikosti $220 \mu\text{F}$. Výrobce modulů [25] doporučuje kvůli ochraně před rušením připojit k budičům kondenzátor o velikosti alespoň $100 \mu\text{F}$. Kvůli úspoře místa na desce byly k 6 budičům připojeny 4 kondenzátory $220 \mu\text{F}$.

Signály Reset a Sleep jsou připojeny k pinu PE12 na STM kitu. Aby nedošlo k nechtěné aktivaci motorů před a během startu mikrokontroléru, jsou tyto signály trvale přizemněny rezistorem o velikosti $47 \text{ k}\Omega$. Mikrokontrolér je schopen tyto signály stále ovládat. Signály Reset a Sleep jsou ze všech řídicích modulů připojeny na stejný pin a tím je zajištěno spuštění i vypnutí všech motorů naráz. Řídicí modul DRV8825 podporuje mikro-krokování motorů. V případě tohoto robota není tak jemné rozlišení potřeba, a piny M0 až M2 nejsou použity. Pinu Enable byla ponechána implicitní hodnota

zajištěná pulldown rezistorem na DRV modulu, protože tuto funkci zajišťují výše zmíněné piny Sleep a Reset.

Piny Dir (směr) a Step (krok) jsou připojeny vždy na příslušný pin STM kitu, který řídicímu modulu dodává potřebné informace k pohybu. Část zapojení řídicích modulů je vidět na obrázku 3.14.



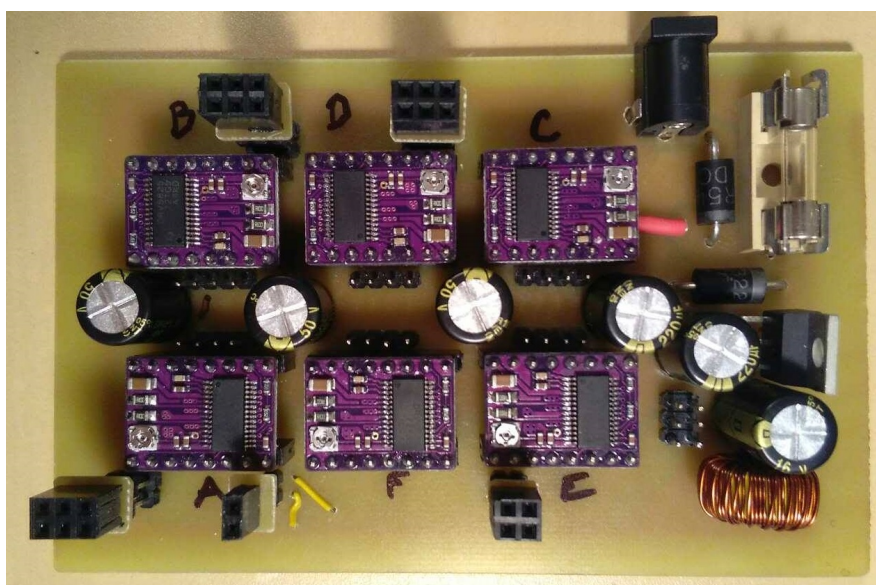
Obrázek 3.14: Zapojení DRV8825 řídicích modulů

Takto navržený obvod byl dále zpracován na plošný spoj. Vytvořené nebo již předem existující footprinty byly podle schématu vygenerovány a poté manuálně umístěny a propojeny.

Deska byla z časových důvodů vyrobena v domácích podmínkách. Na fotosenzitivní jednostranný plošný spoj byla přiložena vytištěná předloha plošného spoje a přes něj osvětlena UV světlem. Spoj byl ponechán nějaký čas ve vývojce (NaOH) a následně vyleptán chloridem železitým (FeCl_3). Dále byla potřeba vyvrtat průchody v potřebných místech, aby bylo možné tento spoj osadit.

Protože plošný spoj byl vytvořen pouze jednostranný, bylo potřeba vyrobit několik drátových propojek. Jedna z propojek vede silové napětí 24V. Další dvě propojky vedou sériovou komunikaci USART mezi STM kitem a konektorem pro Raspberry Pi. Tato komunikace je na STM ovládána piny RX (PD6) a TX (PD5). Moduly DRV jsou usazeny v dutinkových lištách stejných rozměrů. Pro zvednutí STM kitu do výšky nad kondenzátory byla využita kombinace žebříčkovité pinové lišty a dutinkové lišty. Tyto dutinkové lišty kombinované s kolíky jsou v praxi běžně používány v tzv. shieldech - rozšiřujících deskách pro vývojové kity. Při konstrukci řídicí desky nebyly tyto komponenty k dostání, a tak byly provizorně nahrazeny kombinací dutinkové lišty a pinů. Osazený plošný spoj včetně řídicích modulů před přidáním STM kitu je možné vidět na obrázku 3.15.

K propojení mezi motory, plošným spojem a Raspberry Pi byl potřeba plochý kabel, dutinky a konektory. Seznam součástek včetně počtu kusů



Obrázek 3.15: Osazený plošný spoj

je v příloze v tabulce A.2.

Důležitou součástí výběru komponent robota jsou programovatelné kity. Byl vybrán mikroprocesor STM32F401, Raspberry Pi B+ a s ním kompatibilní kamera. Seznam těchto součástek je v tabulce 3.3.

Součástka (Typ)	Počet [ks]
STM32F401TC discovery kit	1
Raspberry Pi B+	1
Raspberry Pi Camera V2	1

Tabulka 3.3: Seznam dalších použitých elektronických součástek

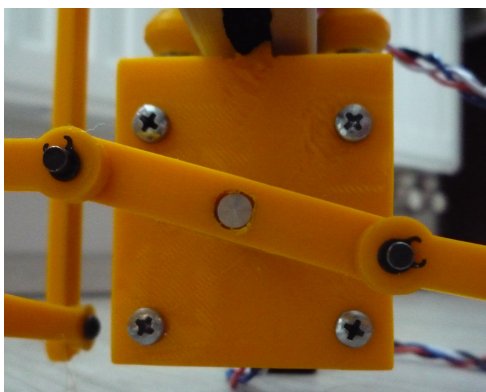
3.4 Sestrojení robota

Hliníkový profil bylo potřeba nařezat na části, vyvrtat otvory pro šroubky a vytvořit závity M5 pro jejich spojení. Seznam délek a počtu nařezaných částí je v tabulce 3.4. Jak již bylo zmíněno v úvodu kapitoly, cílem je vytvořit rám znázorněný na obrázku 3.1. Nejdelší kus profilu (300 mm) představuje horní část konstrukce. Bylo potřeba vyvrtat díry ve vzdálenosti 10 mm na obou stranách. Dvě části o délce 260 mm byly použity jako boční strana konstrukce. Na horní straně bylo potřeba vytvořit závity v již existující vnitřní drážce a dále ve výšce 7 mm od spodní strany provrtat díry pro šrouby. Třetí část stejné délky pak byla potřeba pro vedení motorů. V této části byly vytvořeny na obou koncích závity M5 a uprostřed (130 mm) vyvrtána díra pro závitovou tyč. V profilech o délce 120 mm byly vytvořeny na jedné straně závity a na druhé straně ve vzdálenosti 10 mm provrtány díry. Nejkratší části potřebovaly pouze závit na jedné straně.

Délka [mm]	300	260	120	60
Počet [ks]	1	3	3	2

Tabulka 3.4: Délky profilu jednotlivých částí konstrukce

Po nařezání profilu a přípravě součástek popsaných v kapitole 3.1, 3.2 a 3.3 byla sestavena konstrukce robota. Nachystané hliníkové profily byly spojeny šrouby M5, středem protažena závitová tyč. Dva motory byly uchyceny do držáků spodních motorů a umístěny do příslušných drážek. Uchycení dolních motorů je možné vidět na obrázku 3.16.



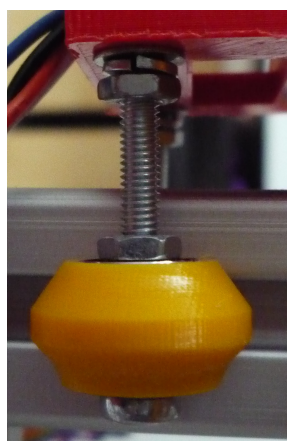
(a) : Uchycení dolního motoru: pohled z přední strany



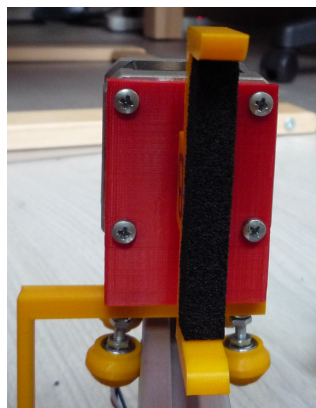
(b) : Uchycení dolního motoru: pohled z boční strany

Obrázek 3.16: Uchycení dolního motoru

K držákům horních motorů byla přišroubována kolečka a ramena posunu. Čtyři zbylé motory upevněné v takto připravených držácích pak byly umístěny na osy ve směru pohybu. Obrázek 3.17 ukazuje přišroubování kolečka a umístění motoru na profil. Propojení ramen a dolních motorů pomocí ojnice bylo zajištěno nýty o průměru 3 mm a délky 10 mm. Zajištění proti vypadnutí a posunu bylo vyřešeno pojistnými kroužky o průměru 3 mm. Toto propojení je vidět na obrázku 3.18a. Na motorové hřídele byly umístěny příslušné propojky s ojnicí nebo úchopy na kostku. Způsob uchycení je vidět na obrázku 3.18b.

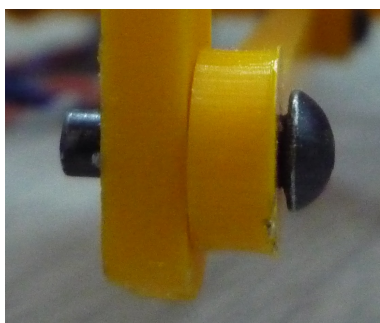


(a) : Přišroubované kolečko



(b) : Uchycení horního motoru:
pohled z přední strany

Obrázek 3.17: Konstrukce pro hodní motory



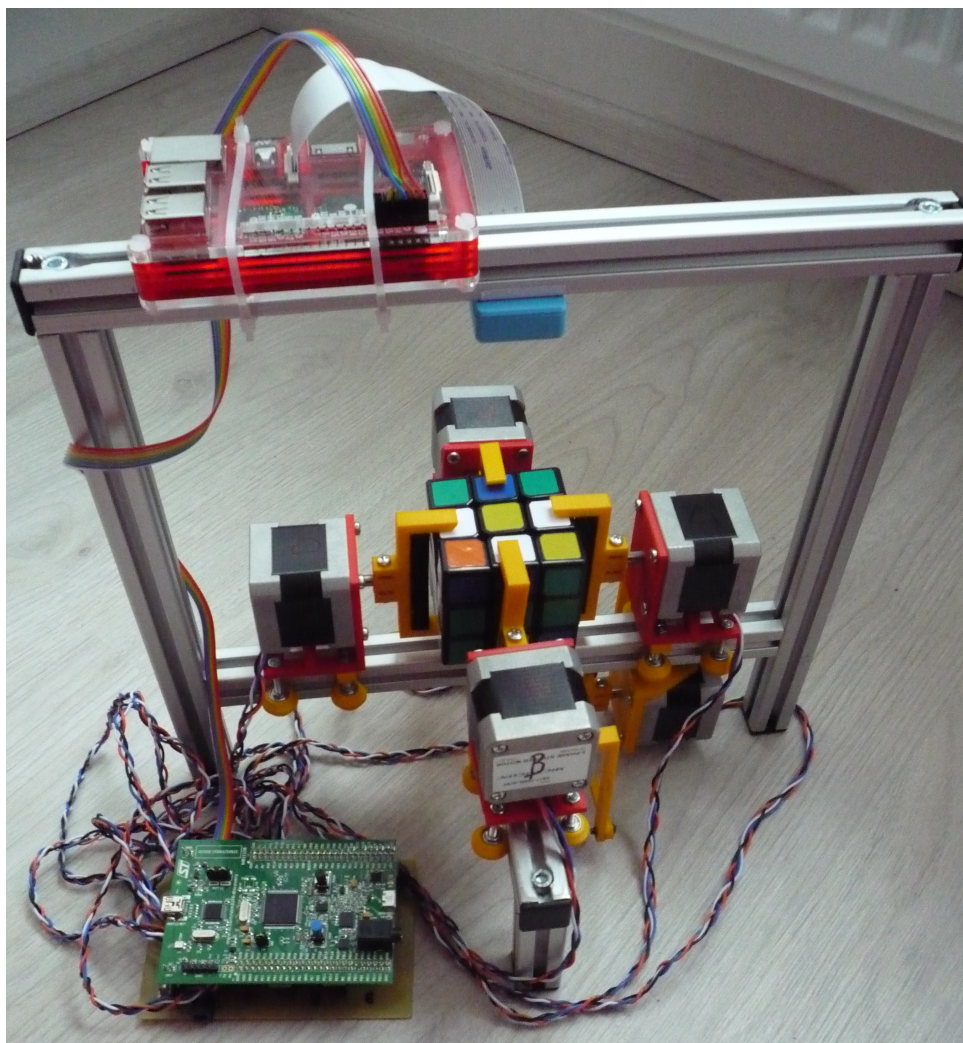
(a) : Spojení ramena posunu
s ojnicí



(b) : Uchycení úchopů nebo propojek
na hřídel

Obrázek 3.18: Spojování součástí

Do osazeného plošného spoje byly připojeny motory, Raspberry Pi a STM kit. K Raspberry Pi bylo připojeno napájení, komunikační vodiče a kamera. Mikropočítač byl upevněn pomocí stahovacích pásek připevněno na konstrukci robota shora, kamera se nachází uprostřed horního profilu. Nakonec byl připojen zdroj. Celá konstrukce je vidět na obrázku 3.19.



Obrázek 3.19: Sestavená konstrukce robota

Kapitola 4

Software

4.1 Raspberry Pi

V práci je využito Raspberry Pi B+, na které byl nainstalován systém Raspbian s verzí jádra 4.9. K tomuto počítači je připojena kamera Raspberry Pi Camera V2.

Implementace řešení byla naprogramována v jazyce Python s využitím knihoven OpenCV a numpy. Z důvodu využití balíku cv2 (OpenCV verze 2) je většina programu napsána v Pythonu verze 2.7.

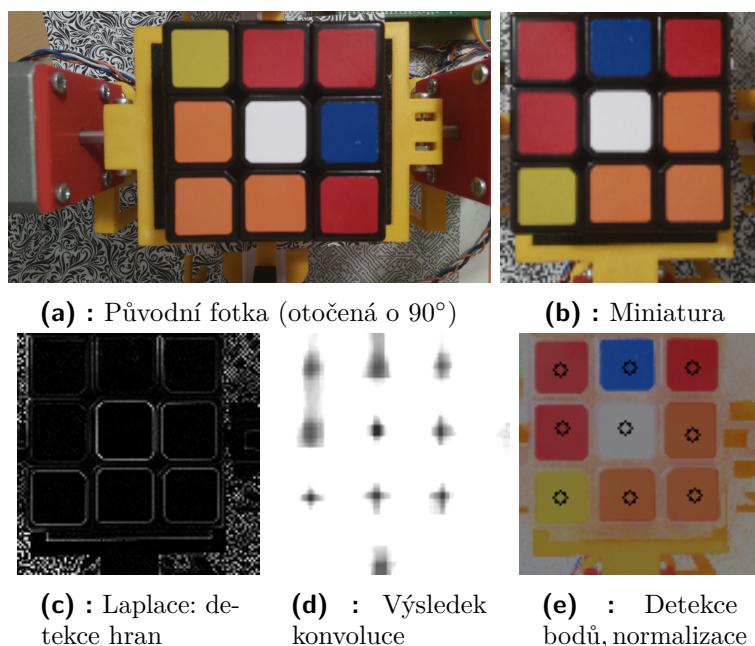
Kód se skládá z několika částí, které jsou postupně volány z hlavního programu "main.py". Ten nejdříve komunikuje s STM32F4 a naskenuje stav kostky. Jednotlivé fotografie načtené do proměnné typu numpy pak předá funkci pro rozpoznávání obrazu. Tato část vrátí pole trojic čísel (barevných hodnot BGR). Pole je následně předáno klasifikaci barev, která ho zpracuje a vrátí řetězec reprezentující stav kostky. Takto zjištěné zadání hlavolamu je předáno algoritmu na vyhledávání řešení, který vygeneruje postup řešení jako řetězec znaků. Řešení ve standardním formátu je předáno interpretu, který je přeloží na řetězec zpracovatelný STM kitem. Nakonec postup odešle do mikrokontroléru.

4.1.1 Rozpoznávání obrazu

Postup zpracování obrazu z kamery je ilustrován na obrázku 4.1. Nejdříve je snímek oříznut z obou stran tak, aby zůstal jen střed s poměrem stran 1:1. Obraz z kamery má zbytečně velké rozlišení, a tak je zmenšen (viz obrázek 4.1b), při čemž dochází k průměrováním sousedních pixelů a tím k základnímu potlačení šumu. Zpracování v menším rozlišení je také mnohem rychlejší.

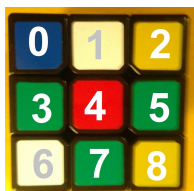
Dále jsou nalezeny pozice barevných ploch kostky. Při tom je využito faktu, že plošky jsou souvislé barvy, zatímco jejich okolí obsahuje hrany. Zdroj [34] dobře popisuje princip laplaceovského detektoru hran, který je použit. Výsledkem aplikace filtru knihovnou OpenCV je obrázek 4.1c. Tmavé oblasti ukazují hladké plochy, bílé skvrny pozice hran.

Známa je i přibližná velikost barevných plošek. Aplikací konvoluce s jádrem o průměru srovnatelném s hranou plošky je získán obrázek 4.1d. Místa s nejhladším okolím utvoří ve výsledku černé skvrny. V obrázku je nalezen



Obrázek 4.1: Fáze detekce barev

nejtmavší pixel, tj. nejpravděpodobnější pozice jedné barevných plošek. Aby nedošlo k vícenásobnému "nalezení" téže plošky, je nalezený bod i jeho okolí vymazáno z obrázku. Zbýlých 8 barevných plošek je nalezeno stejným způsobem. Výsledkem této fáze je 9 souřadnic středů plošek, které jsou seřazeny podle obrázku 4.2 a uloženy.



Obrázek 4.2: Pořadí bodů detekovaných na fotografii

Se znalostí souřadnic se program vrátí k barevné verzi obrázku, který převede z barevného prostoru BGR do HSV. Tento barevný prostor obsahuje souřadnice pro odstín (Hue), sytost (Saturation) a jas (Value) barvy. Aby byl co nejvíce eliminován vliv světelných zdrojů, je provedena normalizace jasové složky na hodnotu 130, viz obrázek 4.1e. Tím je minimalizován problém stínů, které vrhá konstrukce robota a umožňuje mu pracovat i v situacích, kdy je světlo nerovnoměrné a přichází např. pouze z jedné strany.

Okolní světlo může způsobit posun barev v barevném spektru, tedy odstín (souřadnici H), avšak pokud jsou jím ovlivněny všechny plošky stejně, je problém eliminován později, při klasifikaci barev.

Takto upravený obrázek je převeden zpátky do prostoru BGR, který se chová lépe s použitou metrikou klasifikátoru. V určených souřadnicích je přečtena

barevná hodnota a uložena ve stejném pořadí, jako souřadnice.

Stejný proces je proveden pro všechny strany kostky, vrácené hodnoty jsou v předem určeném pořadí uloženy za sebe do jednoho pole. Toto pole má tedy nakonec velikost 54 hodnot. Pole je dále předáno klasifikátoru.

Při zpracování obrazu touto metodou nastává problém při špatném natočení kamery. Ta je však umístěna napevno uprostřed horního profilu a tedy by k takovým problémům nemělo docházet. Dalším omezením je postavení robota na příliš jednobarevnou plochu. I když program okolí kostky výrazně ořezává, vždy nějaká část zůstane a mohla by se vyhodnotit jako ploška kostky. V implementaci je zahrnuta i kontrola, zda se nalezené body nachází přibližně ve čtvercové matici. Pro detekci je však potřeba mít určitou toleranci polohy, aby algoritmus nezahazoval validní body. Tudíž stále může teoreticky dojít k nesprávné detekci mimo kostku. Jednoduchým řešením je pod robota umístit plochu, která nemá jednolitou barvu. Čím členitější podklad bude, tím menší šance na špatné detekování.

4.1.2 Klasifikace naskenovaných barev

Klasifikátor zná polohy barevných ploch na kostce, z obdrženého pole. Proto je možné odečíst z každé strany hodnotu středové plošky. Ty jsou přiřazeny podle naskenovaného pořadí do kategorií U, D, L, R, F, B (horní, dolní, levá, pravá, přední, zadní). Na ostatní hodnoty je aplikován klasifikátor nejbližšího souseda (1NN). Tento klasifikátor používá eukleidovskou metriku z jednotlivých složek barvy. Metrika je vypočtena dle rovnice 4.1,

$$V = \sqrt{|R - r|^2 + |G - g|^2 + |B - b|^2} \quad (4.1)$$

kde V je vypočítaná vzdálenost, R , G a B jsou barevné hodnoty středu (červená, zelená a modrá složka), se kterým je porovnávána hodnota aktuálně zařazovaného bodu, r , g a b jsou hodnoty tohoto bodu.

Bod je zařazen do kategorie k jejíž středu má nejmenší vzdálenost. Název této kategorie je pak uložen do pole, které má stejné pořadí jako pole obdržené. Po zařazení všech bodů je vytvořen řetězec, kde jsou kategorie zapsány v pořadí, které vyžaduje část programu pro výpočet tahů.

4.1.3 Výpočet tahů

Algoritmů na skládání Rubikovy kostky existuje velké množství. Jsou takové, které aplikují na každý stav kostky předem nalezené postupy, ale i takové, které pro nalezení řešení prohledávají stavový prostor. Výpočet postupu pomocí prohledávání je složitý matematický problém. Bylo matematicky dokázáno, že při optimálním řešení je jakýkoli stav Rubikovy kostky možný vyřešit do 20ti tahů [33]. Toto je nazýváno Božským číslem (God's Number).

Existující algoritmy - předem nalezené postupy

Tyto algoritmy mají pevně daný postup. Ať už začnou nejdříve umisťovat rohy, skládat vrstvy, nebo postupovat od jednoho rohu, vždy je pro každý

překlopit a uchopit ji ze správné strany, aby mohl takový tah provést. Tím je možné dosáhnout toho, aby algoritmus optimalizoval tahy vhodné pro robota, ne člověka. Taková úprava z časových důvodů není součástí práce.

Použitá implementace dvoufázového algoritmu hledá řešení délky dvaceti tahů nebo kratší. Na čas výpočtu řešení je aplikován horní limit, při jehož dosažení je optimalizace tahů předčasně ukončena, a poskytnuté řešení je delší než 20 tahů. Prakticky k této situaci nedochází. Časový limit je možno při volání kódu nastavit. Existuje také grafický program využívající těchto algoritmů, zvaný "Cube Explorer 5.13", který je součástí zdrojových kódů ke stažení.

4.1.4 Převod oficiální notace na příkazy pro motory

Notace, kterou se značí tahy po celém světě je popsána v tabulce 4.1. Všechny variace jsou uvedeny pouze pro otočení pravé strany (R), ale stejným způsobem se značí i strany ostatní. Otočení značené velkým písmenem bez dalšího znaku je otočením po směru hodinových ručiček z čelního pohledu na danou stranu. Obrázky znázorňující pohyb jsou k nalezení na[29].

Znak	Význam
R	Otoč pravou stěnu kostky od sebe
R2	Otoč pravou stěnu kostky o 180°
r, R3, R', Ri	Otoč pravou stěnu kostky k sobě
L	Otoč levou stěnu kostky k sobě
U	Otoč horní stěnu kostky doprava
D	Otoč dolní stěnu kostky doleva
F	Otoč přední stěnu kostky po směru hodinových ručiček
B	Otoč zadní stěnu kostky proti směru hodinových ručiček

Tabulka 4.1: Přehled oficiální notace

Mikrokontrolér je však schopen zpracovat jen příkazy otočení jednotlivých motorů. Je tedy potřeba tuto notaci převést na značení motorů. Přímá interpretace je vidět v tabulce 4.2.

Znak	Překlad	Poznámka
R	C	Pravá strana
L	D	Levá strana
F	A	Přední strana
B	B	Zadní strana
U	E cD e A	Je potřeba otočit kostku
D	E cD e B	Je potřeba otočit kostku
2	2	
3	znak.lower()	Změní velké písmeno na malé (opačný směr)
" "	" "	

Tabulka 4.2: Převod oficiální notace na příkazy pro STM32F4

V uživatelském manuálu [27] je na obrázku č. 5 "STM32F401VCT6 block diagram" (strana 14) znázorněn blokový diagram procesoru. Kromě samotných časovačů je používána jednotka pro sériovou komunikaci USART2, a vstupně-výstupní porty (brány GPIO). Na diagramu je přehledně vidět, na kterou sběrnici jsou používané periferie připojeny. To je důležité vědět, neboť každý modul je implicitně vypnut deaktivací hodinového signálu. Nejprve je nutné aktivovat hodinové signály všech modulů.

■ Časovače

Pro řízení motorů robota je potřeba periodicky vysílat signály STEP (krok), které jsou vždy pevné délky, avšak je nutné přesně dávkovat jejich počet. Navíc je vhodné postupně měnit intervaly mezi nimi, aby se motor pohyboval plynule.

Výstupní logiku každého kanálu časovače je možné nastavit do tzv. PWM režimu. V tomto módu časovač neustále porovnává s hodnotou v CCRx registru. Pokud je hodnota časovače menší než CCRx, je na výstupu tohoto kanálu log. 1. Ve chvíli, kdy časovač dosáhne této hodnoty, je výstup přepnut do log. 0 až do konce periody. Tím je možné produkovat pulzy libovolné délky.

V této aplikaci jsou zajímavé jen dvě hodnoty CCRx registru, neboť signál STEP má být buďto v klidu ($CCRx = 0$), nebo produkovat pulzy pevné velikosti ($CCRx = PULSE_WIDTH$). To, co umožňuje měnit interval mezi kroky motoru je zmíněná maximální hodnota (perioda) časovače, při které je časovač resetován zpět do hodnoty 0.

■ Počítání kroků

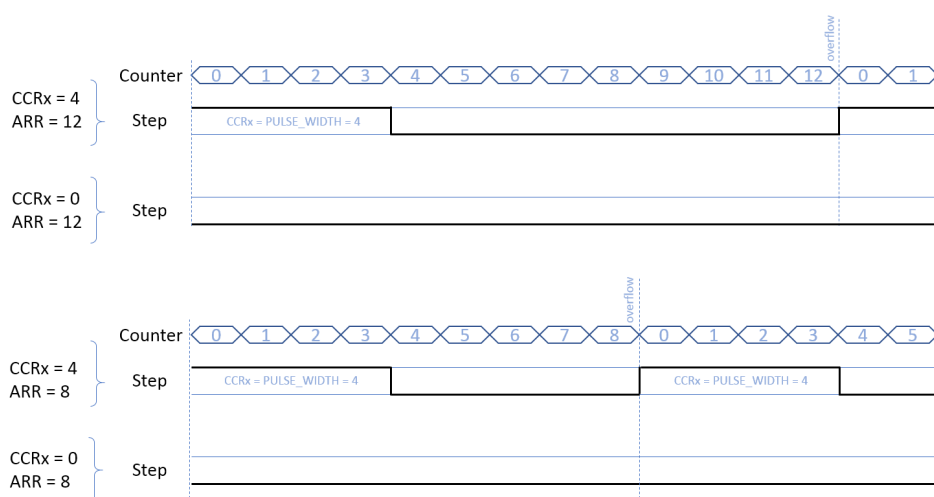
Požadavek přesného dávkování počtu kroků je řešen přerušením. Vzhledem k značnému rozdílu frekvencí kroků motoru (STEP) ~ 840 Hz a taktu procesoru ~ 84 MHz (řádově 1:100000) je možné při každém kroku generovat přerušeni, jehož obslužný kód započítá krok a upraví rychlost motoru pro krok následující. Frekvence kroků motoru je v tomto odhadu nadhodnocena. Fyzicky by taková frekvence znamenala 4,2 otáček/s, což vzhledem k nutnosti předávat kostku mezi dvojicemi motorů, není mechanicky možné. Díky výkonu mikroprocesoru lze tedy použít přerušeni k monitorování kroků.

Na obrázku 4.3 jsou vidět výstupní signály pro signál STEP. Při náběžné hraně tohoto signálu je nastaven příznak přerušeni IF (Interrupt Flag). Příznak je vynulován v okamžiku, kdy procesor přerušeni obslouží.

■ Ovládání časovačů za běhu

Obrázek 4.3 vysvětluje fungování PWM režimu při dvou různých periodách časovače. Ta je nastavena pomocí registru ARR (Auto Reload Register). Změny periody jsou použity pro změnu rychlostí motorů, která je aktualizována během jednoho tahu funkcí SpeedCurve.

Jak bylo zmíněno výše, porovnávací registr (CCRx) má v programu pouze dvě zajímavé hodnoty. Pro hodnotu 0 je signál pro obě periody stejný (signál



Obrázek 4.3: Čítač s porovnávacím registrem

je trvale v hodnotě logické 0). Pro hodnotu `PULSE_WIDTH` (na obrázku jde o hodnotu 4, ale v programu je výrazně vyšší) se signál liší v závislosti na délce periody.

První případ na obrázku 4.3 ukazuje situaci, kde perioda trvá 13 taktů časovače (0 až 12). Je tedy vidět, že výstupní signál pro Step je v logické 1, dokud nedosáhne čísla 4 a v tu chvíli se změní na logickou 0. Když čítač dosáhne hodnoty ARR, počítá znovu od 0 a signál přejde zpátky na logickou 1 a se situace opakuje.

V druhém případě je $ARR = 8$. Protože CCRx nastavuje stejnou délku pulzu, je začátek výstupního signálu stejný jako v předchozím případě. Rozdíl nastává, když čítač dosáhne hodnoty ARR. V tomto případě dosáhne časovač své periody již při hodnotě 8. Poslání následujícího pulzu tedy proběhne dříve.

Nejrychlejší kroky jsou generovány, pokud je střída signálu 50%. Jde o nejkratší možnou periodu, kterou je budič schopen spolehlivě zaznamenat.

Před začátkem otočení kostky, jsou všechny časovače nastaveny na výchozí hodnotu, která je o něco větší než `PULSE_WIDTH`. Toto opatření ponechává signál Step nezměněn po zbytek periody.

4.2.2 Komunikace

Pro jednodušší komunikaci STM a Raspberry Pi byla naprogramována vyrovnávací paměť v podobě kruhové fronty do níž Raspberry zapíše, s kterými motory, jakým směrem a v jakém pořadí se má pohybovat. Mikroprocesor si data postupně čte a vykonává je. Je tím zajištěno, že Raspberry může poslat celý postup prakticky najednou a STM si jej zpracuje takovou rychlostí, jakou bude potřebovat. Během čtení z této fronty bylo potřeba zajistit, aby ve stejnou chvíli nebylo zapisováno, proto během čtení je zakázáno (maskováno) přerušení. Pokud v tuto dobu přijdou data, příznak se objeví, ale

procesor ho po tuto dobu ignoruje. Data zůstanou v přijímacím datovém registru a po zpětném povolení přerušeni jsou zpracována.

■ Protokol mezi mikrokontrolérem a Raspberry Pi

Aplikace je spuštěna až po zmáčknutí tlačítka Start (User button) na STM kitu. Po úvodní inicializaci a startu aplikace se program dostává do nekonečné smyčky, kde sleduje, zda vyrovnávací paměť není prázdná. Pokud jsou přítomna data, přiřadí přečtenému znaku příslušný úkon. Motory jsou na konstrukci i na plošném spoji označeny A až F. Stejným způsobem se na ně odkazují i příkazy poslané z Raspberry Pi do STM kitu. Ten jednotlivé znaky zpracovává podle tabulky 4.4.

Příkaz '2' je využíván v případě, že je potřeba kostku otočit o 180°. Příkaz 's' slouží k jemnému posunutí motoru o minimální počet kroků. Je využíván při kalibraci, kdy je např. potřeba přitisknout úchyty ke kostce, která je již téměř uchopena. Každý příkaz z Raspberry Pi je zakončen mezerou nebo odřádkováním.

Znak (oddělen " ")	Význam
R	Raspberry Pi je připraveno, zapni modrou LED
A B C D E F	Pro příslušný motor nachystej otočení o 90° dopředu
a b c d e f	Pro příslušný motor nachystej otočení o 90° dozadu
2	Vynásob zadaný počet kroků dvěma
s	Pomocný příkaz pro kalibraci
" " "\r"	Vykonej, co máš připraveno
.	Vykonal jsi poslední krok

Tabulka 4.4: Význam přijatých znaků

Na začátku komunikace pošle kit řetězec "Ready" a počká, až dostane odpověď od Raspberry Pi. To po startu aplikace posílá znak 'R'. Případně když byl restartován program pomocí tlačítka Reset na kitu, pak Raspberry reaguje na přijetí řetězce "Ready". STM kit dá najevo navázání komunikace s Raspberry Pi pomocí modré LED. Poté již bude aktivní uživatelské tlačítko Start. Kit pošle "Start" Raspberry Pi, že má začít posílat sekvence pohybů pro naskenování kostky. Zde probíhá komunikace, kdy Raspberry vždy pošle příkaz na otočení a STM kit mu odpoví "Done" až tento příkaz dokončí a Raspberry si může zachytit fotografii nastavené strany. Hned poté pošle příkaz na otočení pro další stranu a mezitím zpracovává aktuální fotografii.

Po dokončení skenování je provedena klasifikace barev a řetězec určující stav kostky odeslán do dvoufázového algoritmu. Výstup z něj je pak převeden na řetězec znaků, které umí zpracovat STM a tento formát odeslán sériovým portem.

Automatická kalibrace motorů nebyla z časových důvodů vyřešena. Je potřeba nastavit výchozí pozici jak motorů, tak úchopů na kostku. Toto by bylo možné řešit koncovými snímači, což je možnost rozšíření do budoucna.

4.2.3 Vypnutí a zapnutí motorů

Pokud již program dokončil všechny příkazy ve vyrovnávací paměti, vypne motory do doby, než je znovu zmáčknuto tlačítko Start. Vypnutí je zajištěno pomocí SLEEP/RESET pinů modulu DRV8825. Díky tomuto se motory zbytečně nezahřívají. Dalším plusem je jednodušší manipulace při vyjmutí a vložení kostky.

Tlačítko Reset na kitu také způsobí vypnutí motorů. Pokud se robot zasekne a mohl by ublížit sobě nebo kostce, stačí toto tlačítko zmáčknout a nastavit robota do správného stavu (tzv. "kill switch"). Dále slouží jako bezpečnostní opatření také v případě, když je robot spuštěn a uživatel do něj vloží ruku.

4.2.4 Před prvním spuštěním

Než bylo možné tento program spustit na reálném systému, bylo potřeba se ujistit o jeho správnosti. K tomu byl využit 8-bitový logický analyzátor od firmy Saleae v kombinaci s programem Logic 1. 2. 10. a USB-TTL UART modul s programem PuTTY. Řídicí signály jednoho motoru (A), zachyceny během vývoje, jsou na obrázku 4.4. Je vidět nedokonalá křivka změny rychlosti otáčení. Je také patrná změna směru i dvojnásobný počet kroků. Příkaz poslaný po sériovém portu byl "a A2 .".



Obrázek 4.4: Výstup z logického analyzátoru

4.3 Uživatelský návod

Po připojení robota k napájení začne bootovat systém Raspbian. Je nutné počkat než začne blikat modrá LED. Mezitím může uživatel umístit zamíchanou kostku robotovi do úchopů. Je potřeba motory co nejvíc přitlačit ke kostce a úchopy otočit kolmo k podložce, tak jak je znázorněno na obrázku konstrukce 3.19. Pokud je systém nachystán a kostka umístěna, stačí zmáčknout tlačítko Start a robot začne hlavolam řešit.

V situaci jako je technický problém nebo spuštění robota v době, kdy ještě není umístěna kostka, poslouží tlačítko Reset na kitu. To zajistí vypnutí motorů a postup pro zapnutí je možné opakovat.

Po dokončení budou motory vypnuty, ale program stále poběží. Kostka v tuto chvíli může být vyjmuta a znovu zamíchána. Pro vložení kostky a spuštění aplikace lze použít stejný postup jako v prvním případě.

Kapitola 5

Závěr

V rámci této bakalářské práce byl sestaven robot skládající Rubikovu kostku. Byl navržen hardware tohoto systému, vybrány potřebné komponenty, vytvořeny 3D modely pro součástky, které nejsou volně prodejné, a nakonec vytisknuty na 3D tiskárně. Navržená deska plošného spoje byla vyrobena, osazena součástkami a úspěšně oživena.

Dále byl naprogramován software pro nafocení kostky a rozpoznání jejího stavu z těchto fotografií. Hledání postupu složení bylo převzato z [32]. Byl implementován překladač vypočtených tahů na pohyby robota. K tomu bylo využito Raspberry Pi s kamerou a STM32F4 kit.

Implementaci softwaru i zdrojové kódy pro 3D tisk je možné nalézt na <https://github.com/novakha5/Robotic-Rubik-s-Cube-Solver>.

5.1 Současný stav

Systém je schopen uchopit kostku a otáčet její strany či rotovat celou kostkou. Je schopen nafotit a rozpoznat počáteční rozložení hlavolamu a zpracovat postup předaný z dvoufázového algoritmu na signály pro mikroprocesor.

Protože došlo k elektronické závadě na řídicím kitu, není část komunikace mezi Raspberry Pi a STM kitem odzkoušena. Její řešení je plánováno na následující dny. Pokud by došlo ke změně kódu, projeví se v repositáři na GitHubu.

Slabou stránkou rozpoznávání obrazu je náchylnost na chyby při jednobarevném podložení robota. To je možné eliminovat podložním členitým obrázkem nebo položením robota na vícebarevný povrch.

Systém má v konstrukci mírné zalomení v ose motorů A a B. To způsobuje, že úchopy v případě vodorovného nastavení nejsou úplně kolmé ke kostce. Systému také chybí připevnění řídicího kitu, které by se dalo řešit například pevnou podstavou. Tím by bylo vyřešeno i podložení robota pro rozpoznávání obrazu.

5.2 Možnosti rozšíření

Do budoucna je možné systém optimalizovat pro rychlost skládání kostky. Momentální hledání postupu cílí na minimální počet tahů pro člověka. Pro robota však některé tahy znamenají více kroků, kvůli přístupu k horní a dolní vrstvě. Bylo by tedy vhodné upravit heuristickou funkci algoritmu pro výpočet tahů tak, aby s touto skutečností program počítal.

Další možností je optimalizovat překlad tahů na signály tak, aby otočení kostky pro příkazy U a D bylo vyžadováno co nejméně.

System lze rozšířit o kosmetické prvky jako jsou světla či zkrácení vodičů motorů. Bylo by také možné upravit desku plošného spoje a místo STM kitu použít mikrokontrolér osazený přímo na desce. Ta se tím zmenší, tato úprava navíc sníží náklady. Dalšími možnostmi rozšíření je například přidání displaye, který by zobrazoval vypočítaný postup, počet celkových a zbývajících kroků a měřil čas. Přidáním ovládacího panelu by se také mohlo zjednodušit ovládání pro uživatele.

U motorů by bylo možné přidat "koncové" snímače, které by zajistily kalibraci a nebylo by potřeba stavět je ručně do určité polohy. Kalibrace by byla dosažitelná také opticky pomocí kamery, ale vyžadovala by propracovanější rozpoznávání obrazu.

Příloha A

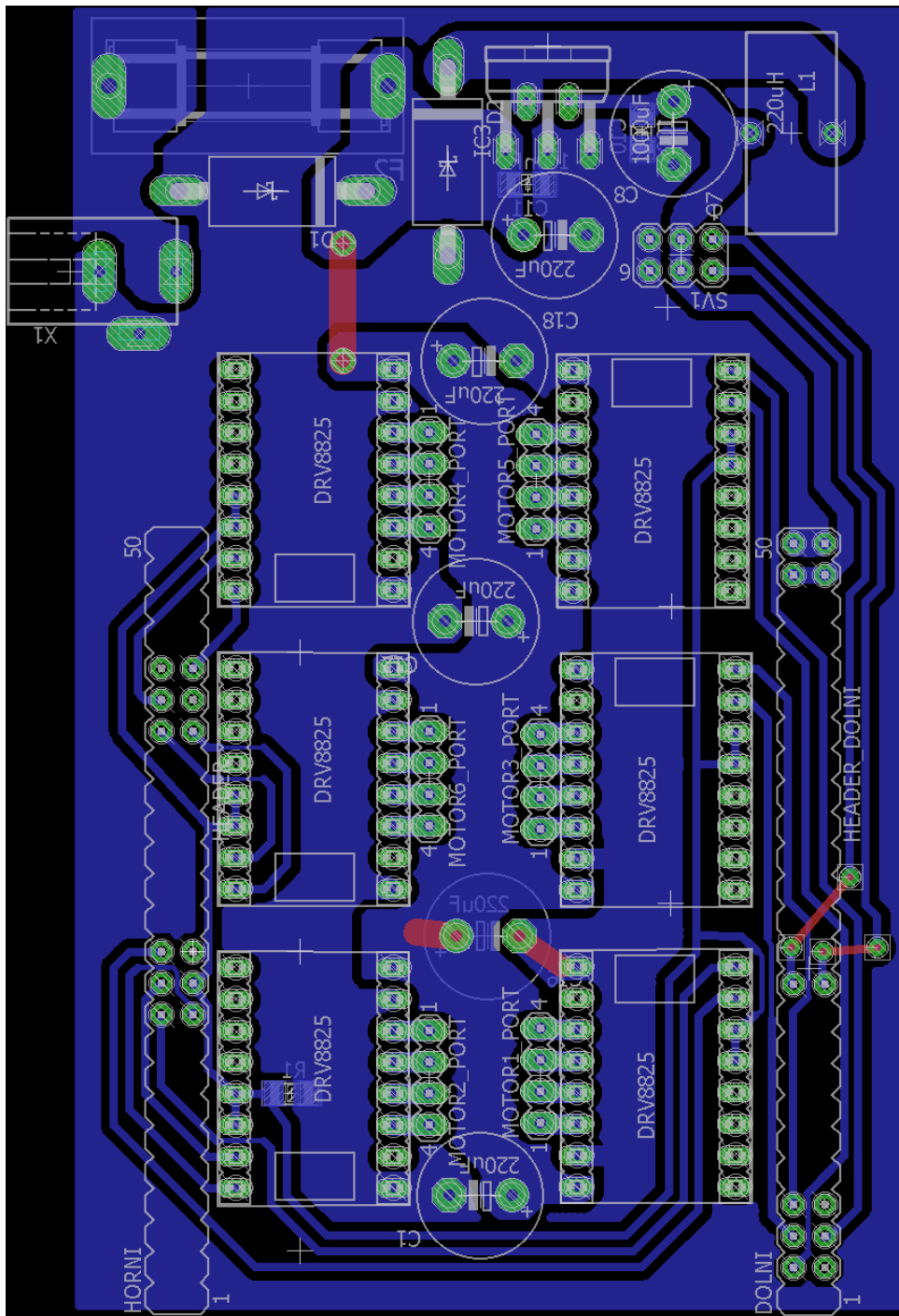
Řídící deska a konstrukce

Součástka (Název)	Počet [ks]
Řídící modul DRV8825	6
Napájecí souosý konektor PC-GK2.1	1
Pojistkové pouzdro do DPS	1
Pojistka trubičková F 2 A	3
Spínaný regulátor napětí LM2576T-5	1
Tlumivka toroidní vertikální DPO220A1	1
Schottkyho dioda 1N5822	1
Schottky dioda SR540-TAP	1
Elektrolytický kondenzátor CE 220u/50VIT	6
Elektrolytický kondenzátor CE 1000u/16VIT	1
Keramický kondenzátor CKS1206 100n/50V	2
Rezistor SMD 1206 47K Ohm	1
Zásuvka 1řadá, přímá, 8 kontaktů	12
Lišta kontaktoá, 1 řada, 4 kontakty	6
Lišta 2řadá, přímá, 2 x 3 kontakty	1
Oboustranný kolík přímý, výška 20mm, 2 x 3 kontakty	4
Zásuvka 2řadá, přímá, 2 x 3 kontakty	4

Tabulka A.1: Seznam potřebných elektronických součástek do DPS

Součástka (Název)	Počet [ks]
Kabel plochý, 10 žil, barevný	1 m (využito výrazně méně)
Dutinka	15
Dvouřadý konektor 6 pinů (2x3)	2
Konektor 4 piny	6

Tabulka A.2: Seznam propojovacích součástek



Obrázek A.2: Plošný spoj

Příloha B

Obsah přiloženého CD

- Zadání bakalářské práce
- Text bakalářské práce
- Složku **tex** se zdrojovými kódy k textu bakalářské práce *.tex
- Složku **OpenSCAD** se zdrojovými kódy 3D modelů *.scad
- Složku **raspi** se zdrojovými kódy pro Raspberry Pi *.py
- Složku **stm** se zdrojovými kódy pro STM kit *.c, *.h
- Složku **foto** s fotografiemi vytvořeného systému a videi ovládání motorů
- Složku **eagle** s návrhem zapojení a plošného spoje

Příloha C

Literatura

- [1] David Gilday, *Build your own LEGO MINDSTORMS robot to solve the Rubik's Cube* <<http://mindcuber.com/>, (2013 - 2016)
- [2] Kisliakov Artem, *MindCub3r in Russian — we do the robot which can collect Rubik's cube* <https://hsto.org/getpro/habr/post_images/39c/261/0e5/39c2610e53bb15db324e789aca22f971.jpg, (15. 10. 2014)
- [3] Olga Kolková, Redaktorka serveru CDR.cz, *Nový světový rekord: Robot z LEGA složil Rubikovu kostku za necelé 4 sekundy* <https://cdr.cz/sites/default/files/cubestormer_3.jpg, (18. 3. 2014)
- [4] Business Wire NewsHQsm, *ARM-Powered® Robot Breaks World Speed Record For Solving A Rubik's Cube At UK's Big Bang Fair 2014* <<https://www.businesswire.com/news/home/20140315005008/en/ARM-Powered%C2%AE-Robot-Breaks-World-Speed-Record-Solving#.UyXSRvldUu7>, (15. 3. 2014)
- [5] Rachel Swatman, *Robot breaks world record solving Rubik's cube in 0.637 seconds* <<http://www.guinnessworldrecords.com/news/2017/3/video-robot-breaks-world-record-solving-rubiks-cube-in-0-637-seconds-464392>, (3. 3. 2017)
- [6] Oscar Williams, *This Robot Just Broke The Guinness World Record For Solving A Rubik's Cube* <http://www.huffingtonpost.co.uk/entry/robot-guinness-world-record-rubiks-cube_uk_58c00fe2e4b0d1078ca2a5cf, (8. 3. 2017)
- [7] Rachel Swatman, *Robot breaks world record solving Rubik's cube in 0.637 seconds* <http://www.guinnessworldrecords.com/Images/Sub1-Reloaded-record-attempt_tcm25-464396.jpg, (3. 3. 2017)
- [8] Daniele Benedettelli, *The first NXT LEGO Rubik's cube solver* <<http://robotics.benedettelli.com/wp-content/uploads/2013/09/lego-rubik-solver-new.png>, (2005 - 2015)
- [9] Deryl Stimm a William Mutterspaugh, *Ruku: a Rubik's cube solving robot for STEM education.* <<https://www.kickstarter.com/>

projects/1772835819/ruku-a-rubiks-cube-solving-robot-for-stem-educatio, (18. 12. 2014)

- [10] Deryl Stimm a William Mutterspaugh, *Ruku: a Rubik's cube solving robot for STEM education*. <https://ksr-ugc.imgix.net/assets/003/036/334/14b842abd6801ecfc5ff0289dbd40122_original.png?w=680&fit=max&v=1418314362&auto=format&lossless=true&s=70ccf1dbcf00fa5a763919bfc5965da3>, (18. 12. 2014)
- [11] Chief Robot, *Rubot II New World Record* <<http://www.robotliving.com/wp-content/uploads/Rubot-II.jpg>>, (27. 9. 2010)
- [12] James Rivington, *Robot can beat any Rubik's Cube in 35 seconds* <<http://www.techradar.com/news/world-of-tech/future-tech/robot-can-beat-any-rubik-s-cube-in-35-seconds-211397>>, (28. 1. 2008)
- [13] MATTHEW HART, foto: Jay Flatland, *WORLD'S FASTEST RUBIK'S CUBE-SOLVING MACHINE SOLVES IN JUST OVER A SECOND* <<https://nerdist.com/wp-content/uploads/2016/01/Worlds-Fastest-Rubiks-Cube-Solving-Robot-Feature-Image-01252016.jpg>>, (26. 1. 2016)
- [14] Jay Flatland, *HighFrequencyTwister* <<https://github.com/jayflatland/HighFrequencyTwister>>, (15. 1. 2017)
- [15] Nancy Owano, *Record-seeking pair show off robot solving Rubik's Cube* <<https://techxplore.com/news/2016-01-record-seeking-pair-robot-rubik-cube.html>>, (26. 1. 2016)
- [16] O.T. Vinta, *Fully 3D-Printed Rubik's Cube Solving Robot* <<https://www.thingiverse.com/thing:2471044>>, (17. 8. 2017)
- [17] Alex Bate, *3D PRINT YOUR OWN RUBIK'S CUBE SOLVER* <<https://www.raspberrypi.org/blog/rubiks-cube-solver/>>, (29. 8. 2017)
- [18] O.T. Vinta, *3D Model 12: 3D-Printable Rubik's Cube Solving Robot* <http://www.otvinta.com/download12.html#rcr_3>, (11. 12. 2017)
- [19] Ruwix, *Rubik's Cube Robots* <<https://ruwix.com/the-rubiks-cube/lego-rubiks-cube-robots-rubot2/>>, (2017)
- [20] Speedsolving Wiki, *List of cube solving robots* <https://www.speedsolving.com/wiki/index.php/List_of_cube_solving_robots>, (12. 11. 2016)
- [21] Toby Heavenrich a Pierre Dyer, *Rubik's Cube* <http://wikis.swarthmore.edu/ENGR005_2008/images/b/b0/Internet_Rubik%27s_Cube.jpg>, (9. 12. 2008)

- [22] Jr Toy Company, *Rubik's Speed Cube* <https://www.kidsscience toys.ca/images/thumbnails/500/500/detailed/24/Rubik%27s_Speed_Cube_2.jpg, (2016 - 2018)
- [23] Microcon, *Technické parametry krokového motoru SX17-1005LQCEF* <<http://www.krokovemotory.cz/SX17-1005LQCEFds.pdf>, (31. 12. 2017)
- [24] National Semiconductor, *LM2576/LM2576HV Series (datasheet)* <<https://www.gme.cz/data/attachments/dsh.330-113.1.pdf>, (1999)
- [25] Texas Instruments Incorporated, *DRV8825 Stepper Motor Controller IC (datasheet)* <<http://www.ti.com/lit/ds/symlink/drv8825.pdf>, (2016)
- [26] STMicroelectronics, *STM32F401xB STM32F401xC (datasheet)* <<http://www.st.com/content/ccc/resource/technical/document/datasheet/9e/50/b1/5a/5f/ae/4d/c1/DM00086815.pdf/files/DM00086815.pdf/jcr:content/translations/en.DM00086815.pdf>, (2017)
- [27] STMicroelectronics, *UM1669 User manual Discovery kit for STM32F401 line* <https://keilpack.azureedge.net/content/Keil.STM32F4xx_DFP.2.11.0/MDK/Boards/ST/STM32F401C-Discovery/Documentation/DM00093902.pdf, (2017)
- [28] STMicroelectronics, *RM0368 Reference manual STM32F401xB/C and STM32F401xD/E* <http://www.st.com/content/ccc/resource/technical/document/reference_manual/5d/b1/ef/b2/a1/66/40/80/DM00096844.pdf/files/DM00096844.pdf/jcr:content/translations/en.DM00096844.pdf, (2015)
- [29] Preetam Priyabrat, *How to solve a rub-ix cube* <<http://cdn.instructables.com/FLJ/JV7Y/HGHC8ETP/FLJJV7YHGHC8ETP.LARGE.jpg>, (29. 4. 2015)
- [30] WonderHowTo, *List of Algorithms for the Rubik's Cube* <https://tag.wonderhowto.com/list-of-algorithms-for-the-rubik_27_s-cube/2/, (2017)
- [31] Herbert Kociemba, *The Two-Phase-Algorithm* <<http://kociemba.org/cube.htm>, (2017)
- [32] Herbert Kociemba, *RubiksCube-TwophaseSolver* <<https://github.com/hkociemba/RubiksCube-TwophaseSolver>, (26. 5. 2017)
- [33] Wikipedia, *Optimal solutions for Rubik's Cube* <https://en.wikipedia.org/wiki/Optimal_solutions_for_Rubik%27s_Cube#Kociemba.27s_Algorithm, (27. 11. 2017)
- [34] Gary Bradski a Adrian Kaehler, *Learning OpenCV*, 1. vydání, O'Reilly, 2008, 555 stran, ISBN: 978-0-596-51613-0

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nováková** Jméno: **Hana** Osobní číslo: **434933**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra řídicí techniky**
Studijní program: **Kybernetika a robotika**
Studijní obor: **Systémy a řízení**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Robotické skládání Rubikovy kostky

Název bakalářské práce anglicky:

Robotic Rubik's Cube Solver

Pokyny pro vypracování:

1. Sestavte stručný přehled existujících systémů, které automaticky skládají Rubikovu kostku.
2. Navrhnete a realizujete hardware i software robota, který bude schopný samostatně poskládat Rubikovu kostku. Konstrukci můžete založit na již existujícím projektu.
3. Celou konstrukci kvalitně zdokumentujte tak, aby si ji mohli snadno postavit další zájemci.
4. Systém koncipujte tak, aby byl vhodný pro propagační akce - připravený pro transport, jednoduchý na obsluhu (automatický start aplikace, automatická kalibrace obrazu i pohonů), vizuálně zajímavý pro diváky a robustní (fungující za různého osvětlení, mechanicky odolný).
5. Při řešení použijte ideálně softwarové i hardwarové prvky, které jsou snadno dostupné.

Seznam doporučené literatury:

- [1] <http://mindcuber.com/>
- [2] <https://www.raspberrypi.org/blog/cube-solver/>
- [3] <https://www.raspberrypi.org/blog/rubiks-cube-solver/>
- [4] <http://rukurobot.weebly.com/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Jiří Zemánek, katedra řídicí techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **29.11.2016** Termín odevzdání bakalářské práce: **09.01.2018**

Platnost zadání bakalářské práce: **30.06.2018**

Podpis vedoucí(ho) práce

Podpis vedoucí(ho) ústavu/katedry

Podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studentky