



Bakalářská práce

Dálkové ovládání robota typu hexapod

Studijní program:

B0714A270001 Mechatronika

Autor práce:

Jakub Adamovský

Vedoucí práce:

Ing. Lubomír Slavík, Ph.D.

Ústav mechatroniky a technické informatiky

Liberec 2023



Zadání bakalářské práce

Dálkové ovládání robota typu hexapod

<i>Jméno a příjmení:</i>	Jakub Adamovský
<i>Osobní číslo:</i>	M20000209
<i>Studijní program:</i>	B0714A270001 Mechatronika
<i>Zadávající katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2022/2023

Zásady pro vypracování:

1. Prostudujte problematiku ovládání robotů.
2. Navrhněte a realizujte interface pro ovládání robota typu hexapod s využitím 32bitového mikroprocesoru.
3. Realizujte dálkové ovládání robota pomocí dálkového ovladače (např. mobilního telefonu).

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: 30-40 stran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

- [1] ESP32-S3-WROOM-2 – datasheet verze 1.0, 2022. Dostupné z:
https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-2_datasheet_en.pdf.
- [2] ALLEN, Grant. Android 4: průvodce programováním mobilních aplikací. Brno: Computer Press, 2013. ISBN 978-80-251-3782-6.
- [3] ZÁHLAVA, Vít. Návrh a konstrukce desek plošných spojů: principy a pravidla praktického návrhu. Praha: BEN – technická literatura, 2010. ISBN 978-80-7300-266-4.

Vedoucí práce: Ing. Lubomír Slavík, Ph.D.
Ústav mechatroniky a technické informatiky

Datum zadání práce: 12. října 2022
Předpokládaný termín odevzdání: 15. května 2023

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

doc. Ing. Josef Černoorský, Ph.D.
vedoucí ústavu

V Liberci dne 12. října 2022

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Dálkové ovládání robota typu hexapod

Abstrakt

Tato práce popisuje vytvoření elektroniky ovládání robota typu hexapod a vytvoření webového rozhraní pro dálkové ovládání robota. Je zde popsán výběr mikrokontroléru, návrh schématu a desek plošných spojů. V práci je také popsán způsob komunikace mikrokontroléru a dálkového ovladače.

Klíčová slova: hexapod, mikrokontrolér, ESP32, Wi-Fi, DPS, robot, Dynamixel AX-12+, C++

Remote Control of Hexapod Type Robot

Abstract

This paper describes the creation of hexapod-type robot control electronics and the creation of a web interface for remote control of the robot. Microcontroller selection, schematic design, and printed circuit boards design are described here. The method of communication between the microcontroller and the remote controller is also described in the paper.

Keywords: hexapod, microcontroller, ESP32, Wi-Fi, PCB, robot, Dynamixel AX-12+, C++

Poděkování

Nejprve bych rád poděkoval doktoru Slavíkovi za dobré vedení mé bakalářské práce a užitečné rady při navrhování a ožívování výsledných desek plošných spojů. Rád bych také poděkoval inženýru Kajzrovi za rady ohledně robotiky a řízení servomotorů a vedení školy za umožnění a financování tohoto projektu.

Mimo univerzitu bych rád poděkoval inženýru Pavlatovi za rady ohledně návrhu a pájení desek plošných spojů a upozornění na časté chyby, kterým jsem se tímto vyvaroval a ušetřil si tak mnoho trápení. Déle bych rád poděkoval svoji rodině a přítelkyni za podporu.

Obsah

Seznam obrázků	10
Seznam tabulek	10
Seznam zdrojových kódů	10
Seznam zkratek	11
1 Úvod	12
2 Hexapod	13
2.1 Krácející podvozek	13
2.2 Dopředná kinematika	14
2.3 Inverzí kinematika	14
3 Rešerše jednotlivých částí problematiky	17
3.1 Řídicí elektronika	17
3.2 Způsob dálkového ovládnání	18
3.3 Antény	19
3.3.1 Parametry antén	19
3.3.2 Impedanční přizpůsobení	19
3.3.3 Druhy antén	19
3.4 Zdroj energie	21
3.5 Návrh desek plošných spojů	21
3.5.1 Základní materiály DPS	21
3.5.2 Stackup	22
3.6 Programovací jazyky	22
3.6.1 Programování mikrokontroléru	22
3.6.2 Programování mobilních zařízení	23
4 Praktická část	24
4.1 Použité servomotory	24
4.1.1 Dynamixel AX-12+	24
4.2 Výběr mikrokontroléru	25
4.3 Napájecí zdroj	26
4.3.1 Měření doby relaxace při vybíjení	27
4.4 Antény	29
4.5 Řešení elektroniky	30
4.5.1 Rozvržení problému	30

4.5.2	Stackup desek	31
4.5.3	Ochranná deska	32
4.5.4	Napájecí deska	36
4.5.5	Řídicí deska	39
4.6	Řídicí program	45
4.6.1	Programování ESP32	45
4.6.2	Hlavní řídicí program	46
4.7	Zkompletovaný hexapod	51
Závěr		53
Použitá literatura		54
A Přílohy		57
A.1	Kusovníky	57
A.2	Schémata	57

Seznam obrázků

1.1	Kostra hexapoda se servomotory	12
2.1	Inverzní kinematika pohled na rovinu X, Y, inspirováno [5]	15
2.2	Inverzní kinematika pohled na rovinu Z, Y, inspirováno [5]	15
2.3	Algoritmus chůze tripod, inspirováno[6], [3]	16
3.1	Raspberry Pi 3 model B	17
3.2	Anténa Yagi	20
4.1	Přední a zadní strana servomotorů AX-12+	24
4.2	Ping paket protokolu DYNAMIXEL Protocol 1.0, inspirováno [17]	25
4.3	Blokový diagram ESP32, inspirováno [18]	26
4.4	Akumulátor Power-X 3S	27
4.5	Průběh vybíjení akumulátoru Power-X6 a fáze relaxace	28
4.6	Detail relaxační fáze akumulátoru Power-X6	28
4.7	DPS meandered inverted-F anténa	29
4.8	3D vyzářovací diagram DPS meandered inverted-F antény	29
4.9	Fotka vybrané prutové antény a RP-SMA konektoru	30
4.10	Blokové schéma rozvržení elektroniky na jednotlivé DPS	31
4.11	Zvolený stackup dvouvrstvých a čtyřvrstvých desek	32
4.12	Schéma ochranné DPS pro odpojení akumulátoru, inspirováno [1],[2]	33
4.13	Průběh napětí regulátoru obvodu BQ296230 závislém na napětí článku akumulátoru, inspirováno [2]	33
4.14	Přední a zadní strana ochranné DPS	34
4.15	Přední a zadní strana osazené ochranné DPS	35
4.16	Schéma testování akumulátorové ochrany	35
4.17	Schéma vstupní části napájecí desky	36
4.18	Schéma části s konektory napájecí desky	37
4.19	Schéma části s měničem napětí napájecí desky	37
4.20	Přední a zadní strana osazené napájecí desky	38
4.21	Schéma části s MCU kontrolní desky	39
4.22	Schéma části s blokovacími kondenzátory analogového převodníku MCU	40
4.23	Schéma části s 3,3 V lineárním zdrojem	41
4.24	Schéma části s konektorem pro připojení kamery k MCU	41
4.25	Schéma části s komunikací mezi servomotory a MCU	42
4.26	Schéma s rádiovou částí	42

4.27	Přední a zadní strana kontrolní desky	43
4.28	Kalkulačka šířky impedančně řízeného spoje [27]	44
4.29	Nanesení pájecí pasty na řídicí desku	44
4.30	Programátor ESP-PROG	45
4.31	Diagram hlavního programu	46
4.32	Výsledný hexapod	51
4.33	Postup spuštění webové aplikace pro ovládání hexapoda	52
A.1	Schéma ochranné DPS	60
A.2	Schéma napájecí DPS	61
A.3	Schéma řídicí DPS	62

Seznam tabulek

4.1	Strapping piny mikrokontroléru ESP32, inspirováno [18]	40
4.2	Použité knihovny	47
4.3	Zahrnuté hlavičkové soubory s konstantami	47
A.1	Kusovník ochranné DPS	58
A.2	Kusovník napájecí DPS	58
A.3	Kusovník řídicí DPS	59

Seznam zdrojových kódů

4.1	Seznam všech vytvořených funkcí	48
-----	---	----

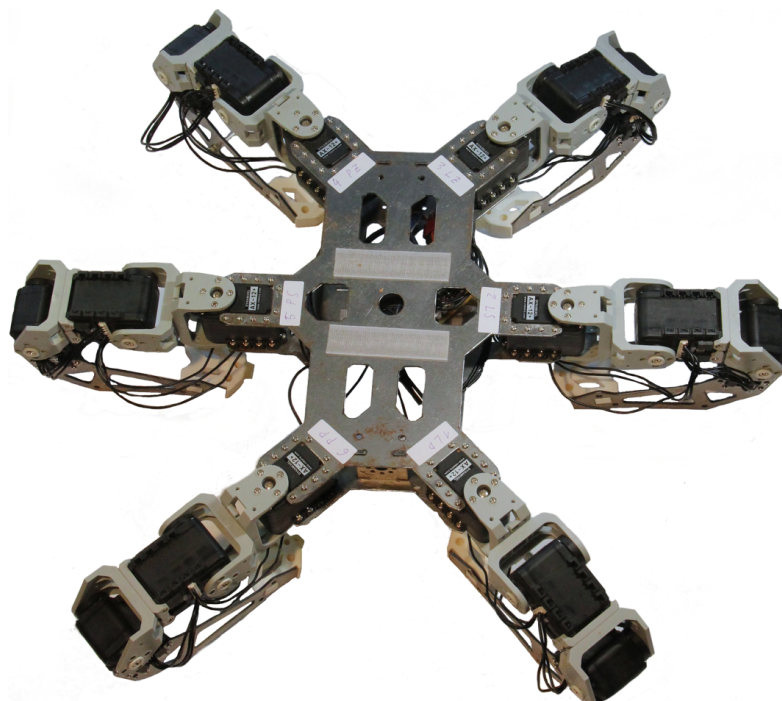
Seznam zkratek

AES	Advanced Encryption Standard
AKU	Akumulátor
ARM	Acorn RISC Machine
CPLD	Complex Programmable Logic Device
FE	Falling Edge
FM	Fakulta mechatroniky, informatiky a mezioborových studií
FPGA	Field Programmable Gate Arraye
FPU	Floating-Point Unit
HASL	Hot Air Solder Levling
HTML	HyperText Markup Language
ID	IDentification
IDC	Insulation-Displacement Connector
JTAG	Joint Test Action Group
LED	Light Emitting Diode
MCU	Microcontroller Unit
PCB	Printed circuit board
PSRAM	Pseudo-Static Random Access Memory
RAM	Random Access Memory
RE	Rising Edge
RP-SMA	Reverse-Polarity SubMiniature version A
SD	Secure Digital
SHA	Secure Hash Algorithms
SoC	System on a Chip
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver-Transmitter
UV	UnderVoltage
UVP	UnderVoltage Protection
VS Code	Visul Studio code

1 Úvod

Zadání na sestavení dálkového ovládání 6nohého kráčejičího robota typu hexapod vzniklo s požadavkem na vytvoření předváděcího produktu pro reprezentaci FM (fakulta mechatroniky) na různých veletrzích. Robot představuje spojení mechaniky, elektroniky, programování a řízení, což jsou obory, kterými se FM zabývá. Hexapod je svými rozměry a váhu vhodný pro přepravu a případní zájemci ho mohou dálkovým ovládáním interaktivně řídit.

V této práci je kladen důraz na vytvoření dálkového ovládání a řídicí elektroniky pro robota typu hexapod. Práce vychází z diplomové práce Ing. Jakuba Hirňala [3], ve které sestrojil hexapoda jenž FM zůstal. Z tohoto robota je použita samotná kostra se servomotory (k vidění na obrázku 1.1), ale řídicí elektronika postavena na minipočítači Raspberry Pi je nahrazena vlastní elektronikou s 32bitovým mikrokontrolérem ESP32. Kvůli předchozímu použití Raspberry Pi s odlišným způsobem ovládání musel být také vytvořen nový program pro řízení robota a pro dálkové ovládání.



Obrázek 1.1: Kostra hexapoda se servomotory

2 Hexapod

Hexapod je 6nohý kráčeující robot. Jedná se o koncepci zaručující průchodnost členitým terénem [4]. Jako hexapod je někdy také nazývána Stewartova plošina, což je paralelní manipulátor se šesti pohony, který je schopný šesti stupňů volnosti. V této práci je však hexapod použit výhradně pro 6nohého kráčeujícího robota.

2.1 Kráčeující podvozek

Princip pohybu kráčeujících robotů vychází z biologického modelu nohy živočichů a jejich chůze. Nejčastější inspirací pohybu je člověk nebo hmyz. Pohyb je závislý na algoritmu chůze, související s počtem nohou robota. Výhodou kráčeujících robotů je možnost překračovat velké překážky (relativně k jejich velikosti), zvládnou tak například pohyb po schodech, který jiným podvozkům může dělat problémy. Mohou překračovat díry, příkopy a výmoly, mohou se pohybovat velmi členitým terénem. Nevýhodou je složitější řízení, velký počet akčních členů zajišťující pohyb, malá energetická účinnost a výrobní a konstrukční složitost [4].

Šestinohé kráčeující roboty

Jedná se o koncept používaný u hexapodů. Noha je často konstruována pomocí tří servomotorů a třech článků označených jako Coxa, Femur a Tibia, názvů používaných pro popis končetiny hmyzu. Tři servomotory jsou minimální počet potřebný pro přímý pohyb nohy při kroku. Existují i varianty pouze s dvěma servopohony na nohu, ale zde dochází k pohybu, při němž je opisována kružnice, nebo elipsa. To způsobuje prokluzování noh na podložce a většímu namáhání servomotorů ve fázi pohybu nohy na zemi. Naopak pro lepší dosednutí nohy na podložku a úplnějšímu pracovnímu prostoru nohy existují i řešení se čtyřmi či pěti servomotory. Jejich nevýhodou je však složitější řízení, přidaná váha servomotorů a i zvýšená spotřeba. Nohy hexapoda jsou uspořádány buď jako tři páry nohou, tedy přední, střední a zadní, nebo jsou osazeny rovnoměrně po obvodu robota ve tvaru šestiúhelníku.

Kráčeující roboty s jiným počtem noh

U jednonohých a dvounohých robotů je snaha o konstrukci, s co nejmenším počtem nohou. U dvounohého robota se často jedná o inspiraci u člověka, tedy o androidy

či humanoidy. Systém je staticky nestabilní, robota je potřeba aktivně vyvažovat, při chůzi se však stává dynamicky stabilním. [4]. U čtyřnohých robotů je častá konstrukce tvořena dvěma páry nohou, tedy přední a zadní pár, jako je tomu běžně u čtyřnohých savců. Jejich často užívaný název je quadruped a pod tímto názvem lze také najít vzhledem podobné roboty typu hexapod, avšak pouze se čtyřmi nohami. Poslední zmíněnou skupinou jsou 8nohé roboty, podobné typu hexapod nazývané octopod. Zvýšením počtu nohou lze dosáhnout větší stability a nosnosti, avšak často na úkor přidané váhy a zvýšené spotřeby z důvodu navýšení počtu servopohonů a dále je potřeba složitější algoritmus řízení.

2.2 Dopředná kinematika

Pokud jsou známy fyzické rozměry jednotlivých ramen robota a jejich konstrukční pootočení, lze z aktuálních úhlů natočení servomotorů (a z aktuálních translačních posunů, pokud je daný robot má) vypočítat koncový bod nohy hexapoda. Pokud má robot v každé vazbě pohon, mající pouze jeden stupeň volnosti (translační nebo rotační), a každé vazbě je přiřazen lokální souřadnicový systém, lze popsat transformaci koncového bodu robota pomocí Denavit-Hartenbergových parametrů. Pro účely ladění algoritmu řízení byl vytvořen soubor v programu Calc (svobodného kancelářského balíku LibreOffice) s hodnotami modelu nohy hexapoda, kam je možné zadat natočení jednotlivých servomotorů a soubor sám dopočítá výslednou pozici koncového bodu. Kvůli jednoduchosti je výpočet dopředné kinematiky v souboru řešen geometrickým přístupem přenesením 3D problému do dvou 2D úloh. Soubor [Vypocet_natoceni.ods](#) je nahrán na příloženém CD.

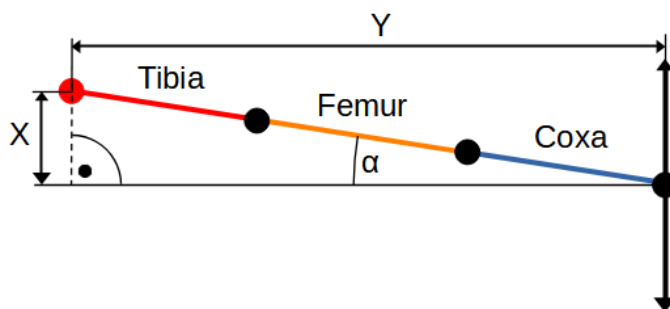
2.3 Inverzí kinematika

Aby bylo možné převést souřadnice bodu konce nohy robota získané z plánování trajektorie na natočení úhlu servomotorů, je potřeba využít inverzní kinematiku. Oproti dopředné kinematice, je tato úloha složitější a může mít i více, nekonečně mnoho nebo žádné řešení. Existují dva možné způsoby nalezení výsledku, buď pomocí numerické, anebo pomocí analytické metody. Numerické metody se často používají u paralelních soustav, kde analytická metoda nemá řešení. Oproti tomu analytická metoda se používá pro svoji menší výpočetní náročnost, a tedy i rychlejší řízení robota.

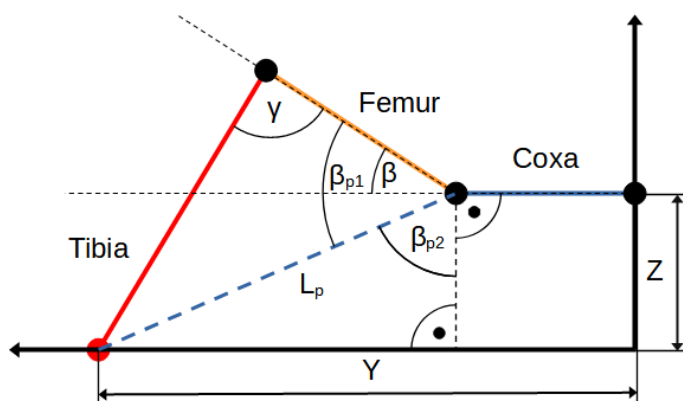
Při analytické metodě lze vycházet z transformačních matic, vytvořených Denavit-Hartenbergových konvencí. Matice se postupně rozepisují na soustavu rovnic, ve kterých se hledá nezávislá proměnná. Pokud se taková proměnná najde, vyjádří se z konstantních rozměrů robota a koncového bodu její hodnota a postupuje se tímto způsobem dále, dokud nejsou všechny proměnné vyjádřeny.

Inverzní kinematika nohy hexapoda

V této práci byl použit geometrický přístup, kde byl problém zjednodušen z 3D prostoru na dva 2D problémy, jeden v rovině X, Y a druhý v rovině Z, Y znázorněném na obrázku 2.1 a 2.2. Obrázky jsou zjednodušeny jako drátový model jedné nohy hexapoda. V úhlech α , β a γ jsou již započítány i posuny úhlů servomotorů způsobené konstrukčním řešením robota. Řešení je inspirováno zdrojem [5].



Obrázek 2.1: Inverzní kinematika pohled na rovinu X, Y, inspirováno [5]



Obrázek 2.2: Inverzní kinematika pohled na rovinu Z, Y, inspirováno [5]

V rovině X, Y na obrázku 2.1 působí pouze první servomotor části coxa. Vzdálenost Y společně se vzdáleností X tvoří odvěsny pravoúhlého trojúhelníku, a lze tak dopočítat úhel α vztahem 2.1:

$$\alpha = \arctan\left(\frac{x}{y}\right) \quad (2.1)$$

Pro řešení zbylých úhlů v obrázku 2.2 byla nejprve určena pomocná proměnná L_p , tvořící přeponu pravoúhlého trojúhelníku, kde výška Z a délka (Y-coxa) jsou jeho odvěsny. L_p lze určit pomocí rovnice 2.2:

$$L_p = \sqrt{z^2 + (y - coxa)^2} \quad (2.2)$$

Pomocí kosinové věty lze dopočítat úhel γ a pomocný úhel β_{p1} , kde obecný trojúhelník tvoří proměnná L_p a délky části nohy femur a tibia vztahem 2.3.

$$\gamma = \arccos\left(\frac{Femur^2 + Tibia^2 - L_p^2}{2 \cdot Femur \cdot Tibia}\right) \beta_{p1} = \arccos\left(\frac{Femur^2 + L_p^2 - Tibia^2}{2 \cdot Femur \cdot L_p}\right) \quad (2.3)$$

Pro řešení úhlu β byl zaveden ještě druhý pomocný úhel β_{p2} , který byl dopočítán pomocí definice funkce sinus v pravoúhlém trojúhelníku tvořeného proměnnými Z , L_p a (Y -coxa) 2.4.

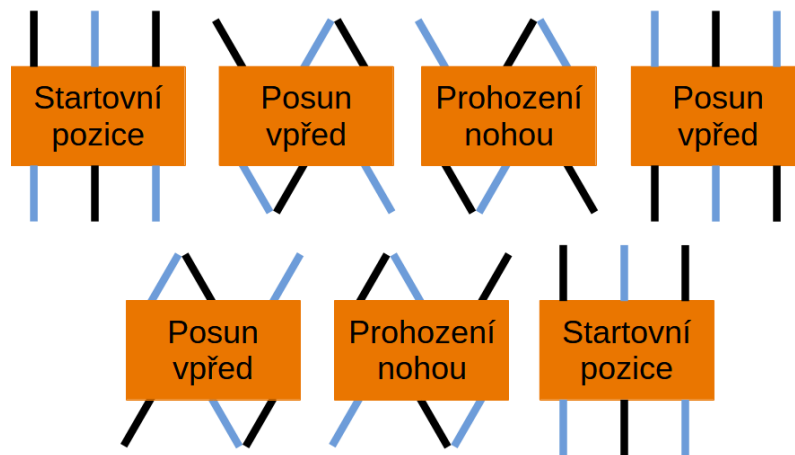
$$\beta_2 = \arcsin\left(\frac{y - coxa}{L_p}\right) \quad (2.4)$$

Výsledný úhel β se určí vztahem 2.5:

$$\beta = \beta_{p1} + \beta_{p2} - 90^\circ \quad (2.5)$$

Algoritmus chůze

Existuje několik algoritmů chůze pro robota typu hexapod [6]. Kromě pevně daných algoritmů může být vytvořen řídicí systém tak, že sám podle terénu přizpůsobí styl chůze robota pro zajištění stability a rychlého překonání terénu. Základní tři algoritmy chůze jsou tripod, vlna a vlnění [6]. V této práci byl použit algoritmus typu tripod schematicky znázorněný na obrázku 2.3.



Obrázek 2.3: Algoritmus chůze tripod, inspirováno[6], [3]

Nohy zakreslené černou barvou znázorňují položenou končetinu na zem, nohy modré barvy znázorňují končetinu ve vzduchu. Z obrázku je zřejmé, že jsou pořád tři nohy položené na zemi a robot je tak staticky stabilní v každém okamžiku, neboli v případě zastavení pohybu v jakékoliv fázi chůze, nedojde k pádu robota. Dále je vidět, že jsou zde dvě skupiny nohou, které postupně střídají fázi pohybu na zemi a ve vzduchu.

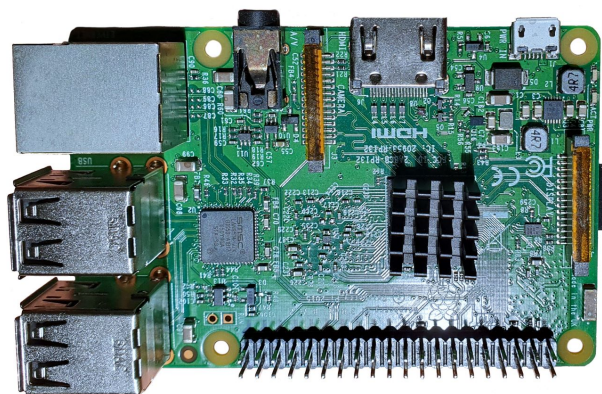
3 Rešerše jednotlivých částí problematiky

Na začátku je potřeba správně rozvrhnout přibližné řešení celého problému a zvolit správné technologie pro jednotlivé úlohy. Následná změna řešení jedné úlohy může mít totiž vliv na správnou funkci té další, například změnou druhu servomotorů může být potřeba změnit napětí napájení pohonů atd.

3.1 Řídicí elektronika

řídící elektronika hexapoda má za úkol plánování pohybu a řízení servomotorů, řešení dálkového ovládání a řešení napájení.

Jednou možností je zakoupit hotové vývojové desky s mikrokontrolérem, jako například desky Arduino. Příklady použití těchto desek pro ovládání hexapoda v různých bakalářských a diplomových pracích lze dohledat mnoho. Výhodou těchto desek je absence nutnosti návrhu řídicí desky a existence mnoha knihoven či přímo hotových řešení ze kterých lze vycházet. Nevýhodou je nutnost použití hotového návrhu, který nemusí splňovat požadavky (rozměry, váha, spotřeba, dostatek pinů pro aktuátory a senzory, absence nebo nevhodný typ délkového ovládání).



Obrázek 3.1: Raspberry Pi 3 model B

Další možností je ovládání založené na počítači (pro větší roboty) či minipočítači jako Raspberry Pi na obrázku 3.1 použitým například v práci Ing. Jakuba Hirňala [3] a v řadě dalších bakalářských a diplomových prací, které lze najít. Výhodou použití Raspberry Pi je možnost nainstalovat operační systém linux s webovým serverem (pro dálkové ovládání prostřednictvím Wi-Fi) a další potřebné programy

jako například databázový systém. V prostředí linux lze navíc pro ovládací program použít velkou škálu programovacích jazyků. Nevýhodou použití Raspberry Pi je vyšší pořizovací cena a horší odezva operačního systému linux, než operační systémy reálného času.

Poslední zmíněnou možností je návrh vlastní řídicí desky, založené na mikrokontroléru. Výhodou je možnost optimalizace návrhu na daný problém jako například výpočetní výkon, paměť, možnost připojení periférií a řešení dálkového ovládání, rozměry, váha a mnoho dalšího. Nevýhodou je složitější návrh.

3.2 Způsob dálkového ovládání

Jednoduchý způsob dálkového ovládání je pomocí kabelu. Tento způsob se používá u vývojových experimentů. Výhodou je možnost napájet robota přímo tímto kabelem a celková jednoduchost provedení. Kabelové spojení je navíc méně náchylné na rušení z jiných vysílačů. Nevýhodou je dosah omezený délkou kabelu a přidané váha kabelu. Kabel se navíc může během cesty zaseknout nebo přetrhnout.

Pro bezdrátové ovládání je možno použít infračervený paprsek, jako tomu je často například u dálkového ovládání televize či rádia. Výhodou je robustnost a jednoduchost řešení. Jedná se navíc o cenově levnou komunikaci. Další výhodou je velké množství hotových ovladačů. Nevýhodou je nutnost přímé viditelnosti vysílače a přijímače. Tento systém se spíše hodí pro občasné poslání příkazu, než pro řízení robota v reálném čase.

Další možností je rádiové spojení. Výhodou pro určité druhy je velký dosah, krátká odezva, velký datový tok. Rádiové vlny mohou prostupovat některými materiály, proto lze například ovládat tímto způsobem robota za překážkou, bez přímé viditelnosti. Nevýhodou je složitější řešení a rušení od ostatních zařízení u běžně používaných volně dostupných pásem.

Možností je buď rádiové spojení, postavené na vlastní komunikaci na jedné z frekvencí volně dostupných pro civilní použití, nebo využití existující komunikační technologie, kde je výhodou hotové řešení, například zabezpečení komunikace.

První takovou možností je technologie Bluetooth. Jedná se o rozšířenou komunikaci, využívající frekvenci 2,4 GHz. Existuje několik verzí, lišících se přenosovou rychlostí, spotřebou a dosahem [7]. Používá se často ve výpočetní technice, například pro bezdrátová sluchátka, počítačové myši a klávesnice či pro mobilní telefony. Jedná se o otevřený standart, který byl také hojně používán v internetu věcí. Běžná přenosová rychlost je 1 Mbit/s, ale jsou i možnosti pro rychlost 2 Mbit/s a dokonce až 24 Mbit/s. Běžný dosah například mobilních telefonů je okolo 10 m, ale existují implementace, díky nimž se dá dosáhnout až 100 m [7].

Další možností je ZigBee. Kromě pásma 2,4 GHz operuje i na dalších bezlicenčních frekvencích, 868 MHz v Evropě a 915 MHz v USA [7]. Technologie je koncipována pro průmysl a nízkoodběrová zařízení. Oproti Bluetooth dosahuje menších rychlostí, v pásmu používaných v Evropě dosahuje rychlost buď 20, 100 nebo 250 kbit/s [7].

Další je LoRa, určená pro dálkový odečet a internet věcí. Vyniká velkým dosahem a malým odběrem energie, ale má menší přenosovou rychlost. Dosah v zastavěné

oblasti je 2 až 5 km, ve volném prostranství až 15 km. V Evropě operuje na frekvenci 868 MHz a přenosová rychlost je adaptivní, od 0,3 kbit/s až po 5 kbit/s [7].

Poslední zmíněný způsob je Wi-Fi (rodina síťových protokolů založené na standardu IEEE 802.11). Výhodou je velká rozšířenost této technologie, dostatečný dosah a velká přenosová rychlost. Používají se dvě hlavní pásma, 2,4 GHz a 5 GHz [7]. Mnoho hotových vysílačů pro použití v návrhu však využívá spíše pásmo 2,4 GHz a pásmo 5 GHz je více využíváno pro výpočetní techniku. Existuje několik variant, lišících se například rychlostí přenosu. Ze všech zmíněných standardů dosahuje Wi-Fi nejvyšší přenosové rychlosti (jednotky až desítky Mbit/s, některé varianty až stovky Mbit/s) [7].

3.3 Antény

Anténa je zařízení, které slouží pro efektivní vyzáření, nebo zachycení radiového elektromagnetického vlnění ze svého okolí.

3.3.1 Parametry antén

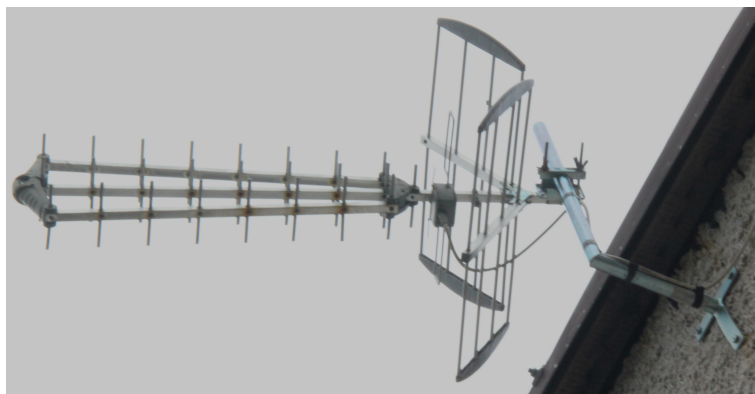
Pro správnou funkci radia je potřeba zvolit správnou anténu s požadovanými parametry. Šířka pásma a frekvence antény udává, pro jaký rozsah frekvencí je anténa schopna vysílat a přijímat elektromagnetické vlnění. Další parametr je maximální vysílací výkon. Důležitým parametrem je také impedance antény, kde typické hodnoty impedancí jsou 50, 75 a 300 Ω . Kromě dalších parametrů je také důležitý tvar vyzařovacího diagramu. S tím souvisí i parametr zisk, udávaný v logaritmické jednotce dBi, který udává, jak moc anténa ve směru svého maximálního vyzařování signál zesiluje. Další parametr je například činitel stojatého vlnění, který popisuje množství výkonu odrazené od antény zpět k napáječi [8].

3.3.2 Impedanční přizpůsobení

Pokud se impedance výstupního zesilovače, vedení a antény neshodují, dojde k odrazu putující vlny od skokové změny impedance. Kvůli tomu nebude do antény dodán maximální možný výkon a nebude tak dosaženo maximálního vysílacího výkonu [8]. Odražená vlna navíc putuje zpět směrem k zesilovači, kde cestou interferuje s vlnou postupující k anténě. Tím je signál dále rušen. Odrazy na vedení mohou navíc poškodit samotný vysílač/přijímač. Aby se těmto jevům zabránilo, volí se všechny prvky přenosové cesty se stejnou impedancí, anebo se musí na místa s rozdílnou impedancí vložit impedanční přizpůsobení [8].

3.3.3 Druhy antén

Existuje velká škála antén. U velkých vysílacích výkonů a velkých vlnových délek je běžné, že je odlišná vysílací a přijímací anténa. Například televizní a rádiové vysílání, kde velká vysílací věž je uspořádaná pro velké vysílací výkony, zatímco přijímací



Obrázek 3.2: Anténa Yagi

antény jsou podstatně menší. Příkladem je dipól, monopól, skládaný dipól, Yagi (běžná pro příjem televizního vysílání) zobrazena na obrázku 3.2 či třeba parabolická anténa (běžná pro příjem satelitního internetu nebo televize). Protože však v této práci je zapotřebí malá anténa pro frekvence okolo 2,4 GHz, připojená k DPS (deska plošných spojů), budou dále rozvedeny jen praktické možnosti, které mají smysl.

Prutová anténa

Jedná se o anténu válcovitého tvaru, většinou s délkou čtvrtiny vlnové délky vysílaného signálu. Vyzařovací diagram mají všesměrový. K desce plošných spojů se často připojují přímo pomocí konektoru, nebo je k nim signál přiveden například koaxiálním vedením.

Chip anténa

Jedná se o SMD (surface mount device) součástku malých rozměrů, která se připájí přímo na DPS. Výhodou jsou malé rozměry, váha, poměrně nízká cena a absence konektoru.

Patch anténa

Jedná se o anténu vyrobenou přímo na DPS stejným procesem, jakým je vyrobena běžná měděná cesta, tedy chemickým odleptáváním nezakrytých částí měděné fólie. Její výhodou je taková, že výsledná anténa je přímo integrovaná v DPS, není tedy potřeba žádná další součástka. Nevýhodou může být místo na DPS, které anténa zabere. Používá se spíše pro vysoké frekvence, kde jsou její rozměry malé. Používá se například pro Wi-Fi či Bluetooth.

3.4 Zdroj energie

Pro mobilní roboty se jako zdroj energie používá převážně elektrochemického akumulátoru. Prvním typem jsou olovené akumulátory. Jejich výhodou je nízká cena a velký dodávaný proud. Používají se proto často v automobilech pro startér spalovacího motoru, nebo například jako záložní zdroj energie v případě výpadku například pro bezpečnostní systémy. Nevýhodou je nízká cyklická životnost a váha.

Další možností jsou niklové akumulátory. Nikl-kadmiový akumulátor má dlouhou životnost a dokáže dodávat poměrně velké proudy, avšak má nízkou energetickou hustotu. Nikl-metal hydridový má větší energetickou hustotu, ale menší životnost.

Poslední zmíněný je lithium-iontový akumulátor. V současné době jsou dva hlavní druhy, jeden založený na gelovém polymerovém elektrolytu, označovaném často jako lithium-polymerový akumulátor a druhý založený na tekutém elektrolytu (často označovaný jako lithium-iontový akumulátor, i když lithium-polymerový akumulátor je také lithium-iontový) [9]. Lithium-polymerový akumulátor oproti lithium-iontovému má větší energetickou hustotu, životnost, bezpečnost a spolehlivost [9]. Nevýhodou jsou vyšší výrobní náklady.

Pro tyto akumulátory existuje několik možných variant lišících se chemickým složením anody a katody. Pro anodu se využívá různých prvků (resp. jejich oxidů), jako nikl, kobalt, mangan či například železo. Různou kombinací se docílí rozdílné vybíjecí charakteristiky, životnosti, energetické hustoty a dalších parametrů [9].

3.5 Návrh desek plošných spojů

Deska plošných spojů, zkráceně DPS, anglicky PCB (Printed Circuit Board), slouží pro montáž elektronických součástek a vytvoření vodivého spojení mezi nimi. Za určitých okolností může být DPS využita zároveň jako konstrukční prvek.

3.5.1 Základní materiály DPS

Prvním je standardní materiál označený jako FR-4. Jedná se o běžně používaný, levný, samozhášivý materiál (FR z anglického Flame Retardant). Lze ho nalézt v běžné spotřební elektronice. Jedná se o skelný laminát plněný epoxidovou pryskyřicí [10]. Samotné FR-4 se od sebe mohou lišit na základě použitého epoxidu. Liší se v teplotě skelného přechodu označené T_g (teplota přechodu z tuhého stavu do ohebného), dialektické konstantě, mechanické pevnosti atd. Běžné hodnoty T_g jsou 130–170 °C a hodnoty dialektické konstanty $D_k = 3,3$ –4,8 [11].

Dalším materiálem je Rogers, což je keramický materiál. Používá se pro vysokofrekvenční a kvalitní návrhy, výrobky pro kosmonautiku, armádu, telekomunikaci či ve zdravotnictví. Materiál je více homogenní a má o řád menší dialektické ztráty, než FR-4. Jeho nevýhoda je vysoká cena.

O něco levnější variantou je PTFE, což je syntetický materiál známý spíše pod označením teflon. Svými vlastnostmi se podobná materiálu Rogers. Existuje několik variant tohoto druhu, lišících se dialektickou konstantou a ztrátovým činitelem.

Další možností jsou desky s hliníkovou podložkou. Jsou navrženy pro lepší odvod tepla při použití výkonových součástek, často výkonných LED. Hliník je dobrý tepelný vodič. Deska je koncipována tak, že na hliníkové podložce je dialektický materiál, často tenký tepelně vodivý, a na něm měděná fólie pro tvorbu obvodů. Tím, jak je deska koncipována, je možné vyrábět pouze jednostranné desky osazené SMD povrchovými součástkami, protože při použití součástek s vývody dojde k vodivému propojení všech vývodů s kovovou podložkou.

3.5.2 Stackup

Stackup lze chápat jako volba skladby vrstev výsledné DPS. Volí se jak tloušťka jednotlivých vrstev mědi, tak materiál a tloušťka dialektických vrstev. Tloušťka základního měděného plátování bývá 18, 35 či 70 um, ale existují i jiné tloušťky [10]. Dále se volí využití jednotlivých vrstev. U vícevrstvých desek se volí, jak se jednotlivé vrstvy použijí. Pokud je potřeba impedančně přizpůsobit cestu, je pod touto vrstvou potřeba mít rozlitou měď s nulovým potenciálem. Skladba u dvouvrstvé DPS s impedančně přizpůsobenou cestou bude mít tedy rozložení: první vrstva signál/-napájení a druhá vrstva rozlitá měď s nulovým potenciálem. Pro čtyřvrstvé desky se často používá skladba signál, nulový potenciál, napájení, signál, nebo signál/napájení, nulový potenciál, nulový potenciál, signál/napájení, pro lepší řízení impedance z obou stran desky.

3.6 Programovací jazyky

3.6.1 Programování mikrokontroléru

Prvním je jazyk symbolických adres, slangově často označován jako assembler. Jedná se o nízkou úroňový programovací jazyk, kde napsané instrukce přímo odpovídají strojovým instrukcím mikrokontroléru. Jeho výhodou je možnost přímo ovládat veškeré registry a další části mikrokontroléru. Nevýhodou je podstatně složitější tvorba výsledného programu.

Další je jazyk C. Jedná se o jazyk, kde je kód nejprve přeložen pro cílový mikrokontrolér (takzvaná kompilace), jehož hlavní výhodou je velká rychlost. Pro přímé ovládání mikrokontroléru je možné přidat části jazyka symbolických adres, aby bylo docíleno například optimalizaci kódu. Syntaxí velmi podobný je jazyk C++, který z jazyka C vychází. Jedná se opět o jazyk, kde je zdrojový kód kompilován a až poté nahrán do mikrokontroléru. Jazyk C++ oproti C podporuje objektově orientované programování.

MicroPython je jazyk, který vychází z jazyku Python. Jedná se o interpretovaný jazyk, který je optimalizovaný pro funkci na mikrokontroléru. Kód je nejprve nahrán do paměti mikrokontroléru, kde je spuštěn program zvaný interpreter, který překládá kód na strojové instrukce. Výhodou je jednoduchost programování, kde se vývojář více soustředí na funkci programu jako celku. Nevýhodou je větší výpočetní a programová náročnost.

Poslední zmínění je kompilovaný programovací jazyk Rust, syntaxí podobný C++. Podporuje objektově orientované programování. Jeho výhodou oproti C++ je však jeho navržení, pro paměťovou bezpečnost.

3.6.2 Programování mobilních zařízení

Pro programování nativních aplikací pro operační systém Android se používá jazyk Java. Jedná se o výchozí programovací jazyk systému Android [12]. Aplikace je nejprve zařízením před spuštěním konvertována na kód, který běží na virtuálním stroji Dalvik [12].

Dále pro programování nativních aplikací pro Android lze také použít jazyk Kotlin. Na konferenci Google I/O v roce 2019 se stal Kotlin hlavním programovacím jazykem systému Android [13], kde do té doby tuto pozici zaujímal programovací jazyk Java. Podle informací na oficiálních webových stránkách jazyka Kotlin, přes 50 % profesionálních vývojářů aplikací pro Android používá jako hlavní jazyk Kotlin, zatímco Javu pouze 30 %. Kromě aplikací pro Android lze vytvářet aplikace i pro iOS [13].

Víceplatformní nativní mobilní aplikace lze psát v platformě Xamarin. Lze v něm psát aplikace pro Android, iOS a Windows. Umožňuje použít až 90 % aplikace napříč platformami. Aplikace jsou psány v objektově orientovaném jazyce C# [14].

Pro programování nativních aplikací pro systém iOS, iPadOS, macOS, tvOS a watchOS společnosti Apple, slouží programovací jazyk Swift. Jazyk je navržen pro bezpečnost, všechny proměnné jsou před použitím inicializovány, pole a jiné datové typy kontrolovány před přetečením a paměť je automaticky spravována [15].

Poslední zmíněný způsob programování je webová aplikace napsaná v HTML 5 (HyperText Markup Language) a jazyce JavaScript. Výhoda je multiplatformost a možnost používat aplikaci bez nutnosti instalace. Aplikace se pouze spustí ve webovém prohlížeči. Dříve tyto aplikace vyžadovaly pro svoji funkci stále připojení k internetu, avšak standart HTML 5 tento problém řeší umožněním webovým stránkám definovat pravidla pro ukládání do mezipaměti [12].

4 Praktická část

4.1 Použité servomotory

Hexapod má 6 nohou a na každé noze jsou 3 servomotory. Celkem je tedy použito 18 servomotorů. Použité jsou servomotory AX-12+ korejského výrobce Dynamixel. Jedná se o starší model, který se dnes již nevyrábí [16]. Byl nahrazen servem AX-12A, se kterým je zpětně kompatibilní (napájecí napětí, výkon, komunikace, rozměry...). Samotné servomotory jsou na obrázku 4.1.



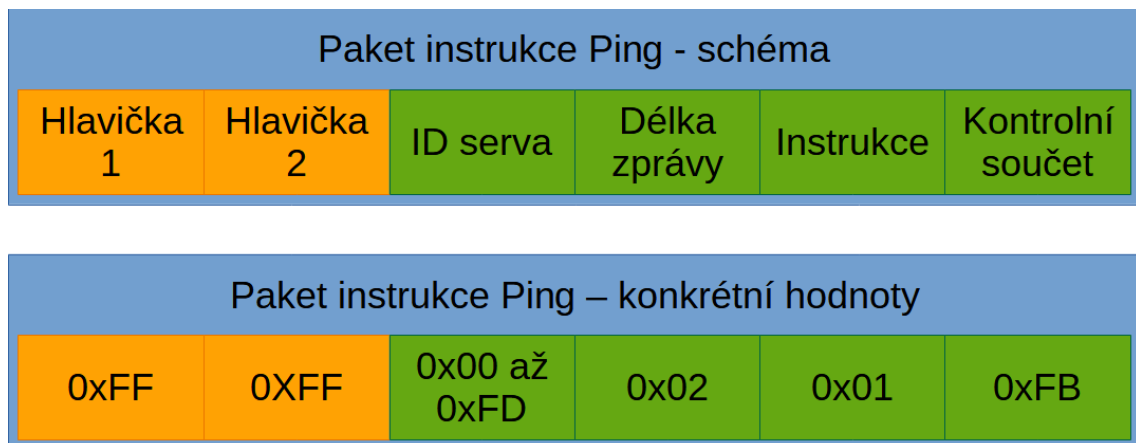
(a) Přední strana AX-12+ (b) Zadní strana AX-12+

Obrázek 4.1: Přední a zadní strana servomotorů AX-12+

4.1.1 Dynamixel AX-12+

Jedná se o inteligentní modelářské servomotory v tom smyslu, že komunikace s mikrokontrolérem probíhá přes sběrnici, na kterou jsou všechny servomotory připojeny. Komunikace probíhá ve formě posílání paketů (uceleného balíčku dat) přes komunikaci založenou na UART, avšak pouze v polovičním duplexu [16]. Převod plného duplexu mikrokontroléru na poloviční používaný komunikací servomotory je více popsán v 4.5.5. Protokol komunikace je nazván DYNAMIXEL Protocol 1.0 a jeho kompletní popis lze prostudovat v online manuálu výrobce [17].

Pakety se svou strukturou mírně liší podle svého druhu, ale základ zůstává stejný. Obrázek 4.2 znázorňuje ping (pro kontrolu, zda je dané servo dosažitelné) paket poslaný mikrokontrolérem.



Obrázek 4.2: Ping paket protokolu DYNAMIXEL Protocol 1.0, inspirováno [17]

Obě hlavičky jsou v paketu obsažené vždycky a to s hodnotou 255, následuje vždy unikátní ID (zkratka pro Identification) servomotoru, pro který je paket určen. Dále je poslána informace, kolik bude posláno dalších bajtů, související s tímto paketem. Následující bajt závisí na odesílateli, pokud paket vysílá servomotor, je další bajt kódové označení chyby (hodnota 0 značí vykonání příkazu bez chyby), pokud ho vysílá mikrokontrolér, je vysláno číslo instrukce a následně případné parametry. Každý paket je zakončen kontrolním součtem, který slouží pro kontrolu správnosti přijatých dat.

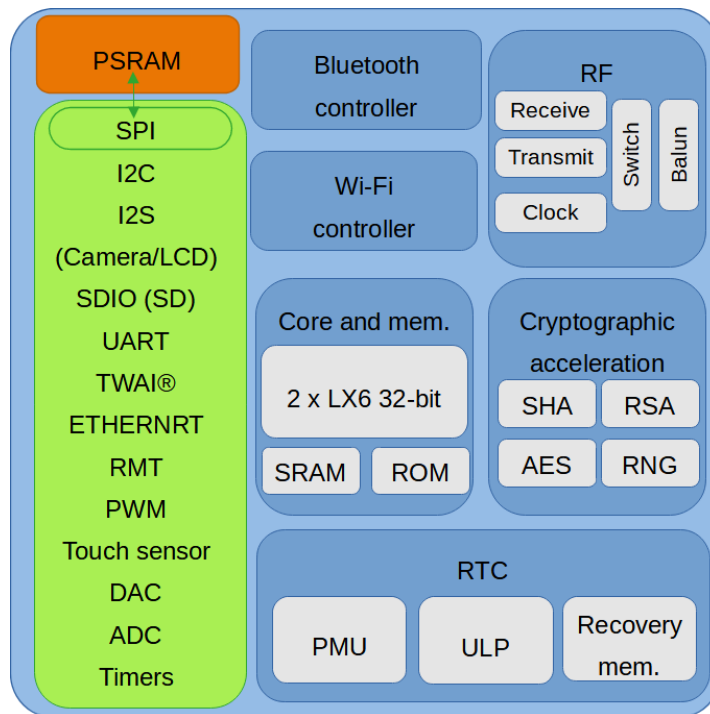
U servomotorů AX-12+ lze ovládat jak základní parametry, jako cílový úhel natočení, tak i další jako rychlost pohybu, maximální přípustný moment (při jeho překročení dojde k odeslání chybového paketu mikrokontroléru), nebo například zapnutí a vypnutí motorů samotných servopohonů. Lze z nich také vyčítat různá data, jako aktuální úhel natočení, aktuální napájecí napětí, teplotu atd.

4.2 Výběr mikrokontroléru

Mikrokontrolér v této práci řeší veškerou obsluhu robota, od ovládání servomotorů a řešení samotného algoritmu chůze, až po zajištění komunikace s dálkovým ovladačem. Vybraný mikrokontrolér pro řízení hexapoda je z rodiny ESP32 firmy Espressif System. Výběr tohoto mikrokontroléru, zkráceně MCU (microcontroller unit), nebo spíše systému na čipu, zkráceně SoC (system on a chip) byl ovlivněn nejenom svou cenou a dostupností, ale také integrovaným Wi-Fi modulem pro zajištění dálkového ovládání s minimálními požadavky na externí součástky pro správnou funkci radia. Jedná se o 32bitový MCU, což byl jeden z požadavků zadání této práce.

MCU obsahuje dvě výpočetní jádra Xtensa® LX6 obsahující také jednotku pro výpočty s čísly s plovoucí desetinnou čárkou, zkráceně FPU (floating-point unit) a dále například akcelerátory pro kryptografické funkce jako AES (Advanced Encryption Standard), SHA (Secure Hash Algorithms) tedy hašovací funkce atd. potřebné například pro zabezpečenou Wi-Fi komunikaci [18]. MCU je tedy dostatečně výpočetně vybaven pro tuto aplikaci. Dalším pozitivem rodiny ESP32 je velké množství kniho-

ven, jak ze strany samotného výrobce, tak ze strany Arduino komunity. Některé tyto knihovny řeší i komplexní složité úkoly jako obsluhu Wi-Fi a webového serveru, který je v této práci použit. V neposlední řadě je možnost výběru programování a ladění programu v prostředí Visual Studio Code v platformě PlatformIO, se kterou je autor seznámen. Funkční blok rozvržení MCU ESP32 je na obrázku 4.3.



Obrázek 4.3: Blokový diagram ESP32, inspirováno [18]

Konkrétní MCU má označení ESP32-D0WDR2-V3. Jedná se o třetí revizi tohoto čipu. Tento konkrétní druh má v sobě zabudovanou PSRAM (pseudo-static RAM) o velikosti 2 MB, avšak chybí paměť flash pro uložení programu. MCU má však dedikované rozhraní SPI (Serial Peripheral Interface) pro připojení flash paměti, konkrétní řešení zapojení je popsáno v sekci 4.5.5. MCU má také rozhraní pro připojení SD karty, nebo kamery, které bylo použito (Secure Digital) pro možné budoucí připojení kamery založené na čipu OV2640, více informací o zapojení je v sekci 4.5.5. Zvažován byl i návrh s hotovým modulem jako ESP32-S3-WROOM-2, kde se již nachází oscilátor, anténa, 32 MB flash a 8 MB PSRAM [19], avšak nevýhodou modulu je menší možnost výběru specifikací a pevně dané rozměry modulu, který pro správnou funkci musí být na výsledné desce plošných spojů umístěn.

4.3 Napájecí zdroj

Jako zdroj napájení byl zvolen lithium-iontový akumulátor z důvodu dostupnosti, ceny, velikosti energie, rozměrů, váhy, požadovaného napětí a velikosti stálého proudu. Přehled dalších možných zdrojů energie je popsán v 3.4. Jedná se o akumulátor s názvem Power-X6 se třemi sériově zapojenými články (často označováno

jako 3S) s celkovou nominální hodnotou napětí 11,1 V. Podle prodejce se jedná o lithium-polymerový akumulátor s kapacitou 4200 mAh a trvalým výstupním proudem 35C, tedy 147 A a maximálním špičkovým proudem 70C, tedy 294 A [20]. Vybraný akumulátor je na fotce 4.3. Obsahuje servisní konektor pro balancování (na fotce jako bílý konektor) a silový konektor typu T-Dean (na fotce jako červený konektor). Udávaná váha akumulátoru je 269 g a cena při nákupu byla 985 Kč.

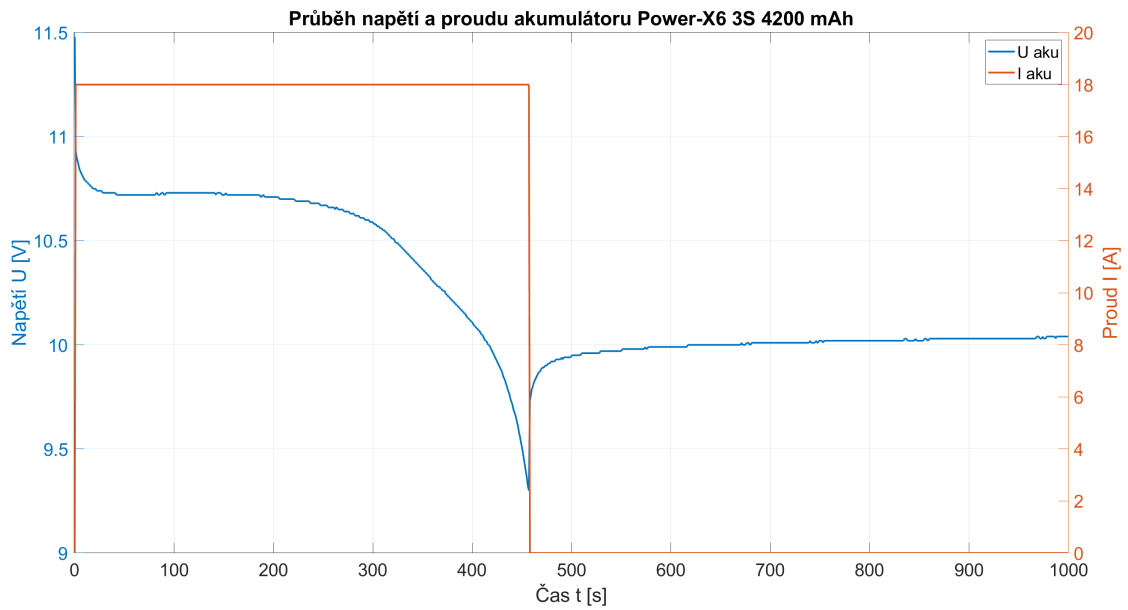


Obrázek 4.4: Akumulátor Power-X 3S

4.3.1 Měření doby relaxace při vybíjení

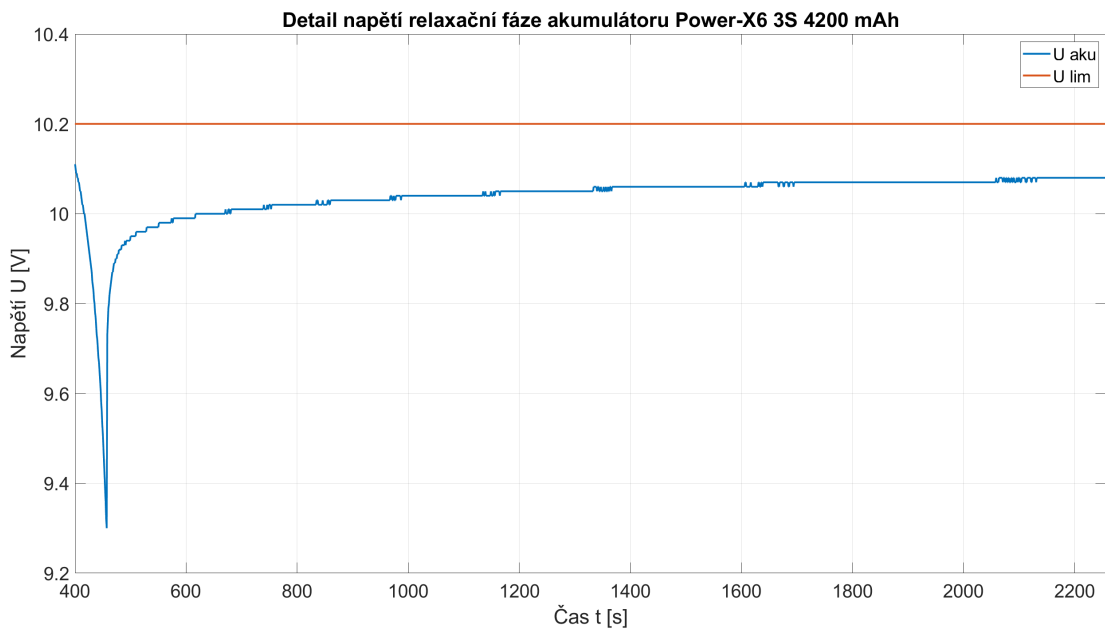
Pro zamezení vybití akumulátoru pod určitou hodnotu, která by mohla nenávratně akumulátor zničit, byla navržena DPS 4.5.3 pro ochranu baterie. Po odpojení vybitého akumulátoru nastává fáze relaxace, při které roste napětí na terminálech akumulátoru. Pokud toto napětí přesáhne určitou mez, připojí integrovaný obvod akumulátor opět ke zbytku obvodu. Pokud by toto nastalo a relaxační doba by byla velmi krátká, integrovaný obvod by neustále připojoval a odpojoval akumulátor. Tím by se poškozoval jak spínací tranzistor a zbytek obvodu, tak i samotný akumulátor. Pro kontrolu tohoto jevu bylo provedeno měření relaxační fáze akumulátoru. Měření bylo prováděno pomocí čtyřvodičové metody pro snížení vlivu odporu přívodních vodičů ve školní laboratoři na přístroji Chroma 17020.

V grafu 4.5 je vidět, že měření nebylo provedeno s plně nabitým akumulátorem, který by měl mít napětí přibližně 12,6 V, ale s částečně vybitým s hodnotou napětí 11,48 V, aby se zkrátil čas experimentu. Dále je vidět, že nastavený odebíraný proud byl 18 A. Na grafu je vidět, že v prvních sekundách došlo k významnému poklesu napětí z 11,48 V na přibližně 10,8 V. Pokles je způsoben vnitřním odporem akumulátoru, na kterém vzniká úbytek napětí (a výkonová ztráta). Od přibližně 50 s do přibližně 200 s je napětí téměř konstantní. Poté následuje fáze prudkého klesání napětí. Měřicí zařízení bylo nastaveno tak, aby odpojilo zátěž při poklesu napětí akumulátoru pod 9,3 V, v tomto případě odpojení nastalo přibližně 460 s po zahájení testu. Je patrné, že ihned po odpojení došlo k skokovému navýšení napětí na terminálu, protože akumulátorem neteče proud a nevzniká tak úbytek napětí na



Obrázek 4.5: Průběh vybíjení akumulátoru Power-X6 a fáze relaxace

vnitřním odporu. Následuje relaxační fáze, při které napětí na terminálech postupně narůstá, avšak z grafu 4.6 je patrné, že hysterezi 300 mV na článek (hodnota pevně nastavena v použitém integrovaném obvodu ochranné desky 4.5.3), v grafu vyznačenou jako U_{lim} , akumulátor nepřesáhne.



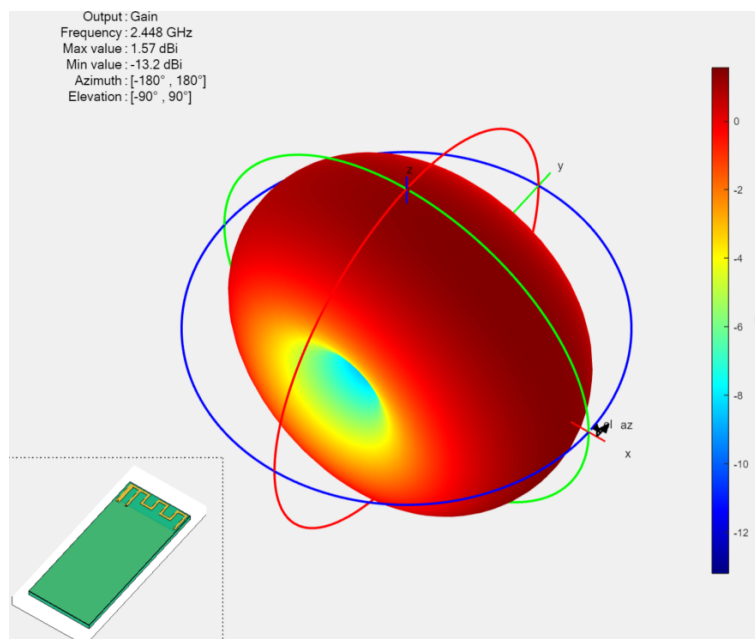
Obrázek 4.6: Detail relaxační fáze akumulátoru Power-X6

4.4 Antény

Rozbor možných antén je v části 3.3.3. Pro tento projekt byly vybrány dvě antény. První je DPS anténa, konkrétně meandered inverted-F anténa zobrazená na fotce 4.7. Jedná se tedy o anténu vytvořenou přímo na plošném spoji. Použitý footprint je z knihovny programu KiCad. Důvodem výběru je nízká cena, jednoduchost provedení, nízká váha a absence potřeby další součástky. Její nevýhodou je nevhodný vyzařovací diagram, který vzhledem k umístění řídicí DPS vyzařuje do boků robota a nad něj. Příjem signálu před a za robotem je tak horší. 3D vyzařovací diagram této antény je znázorněn na obrázku 4.8, diagram byl simulován v programu Matlab (verze R2022b) pomocí modulu PCB Antenna Designer, podle návodu na videu [21].



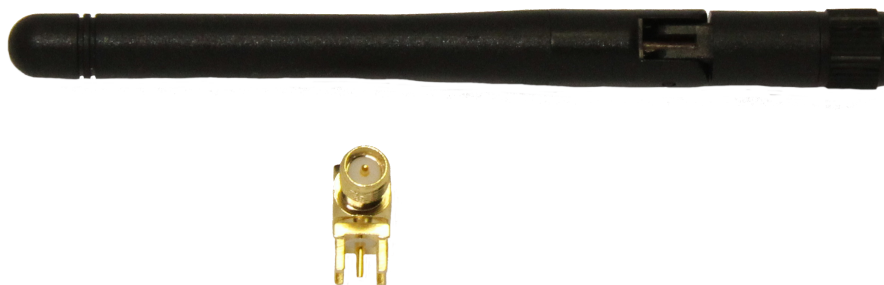
Obrázek 4.7: DPS meandered inverted-F anténa



Obrázek 4.8: 3D vyzařovací diagram DPS meandered inverted-F antény

Druhou použitou anténou je prutová všesměrová anténa firmy DATAWAVE zobrazená na obrázku 4.9. Anténa je dlouhá 95 mm a váží přibližně 6 g. Je opatřena kloubem pro napolohování antény a je zakončena konektorem RP-SMA (Reverse-Polarity SubMiniature version A). Zisk antény je 3 dBi, maximální výkon je 20 W,

rozsah frekvencí je 2400 až 2500 MHz, impedance antény je $50\ \Omega$ a činitel stojatých vln je 1:1,5. Pro připojení antény k DPS byl použit RP-SMA konektor s impedancí $50\ \Omega$ zobrazený na fotce 4.9. Důvodem pro použití této antény byly poměrně malé rozměry, vysoký zisk s vhodným vyzařovacím diagramem pro danou úlohu a dobrý poměr stojatých vln. Nevýhodou oproti PCB anténě je vyšší cena a prostorová náročnost.



Obrázek 4.9: Fotka vybrané prutové antény a RP-SMA konektoru

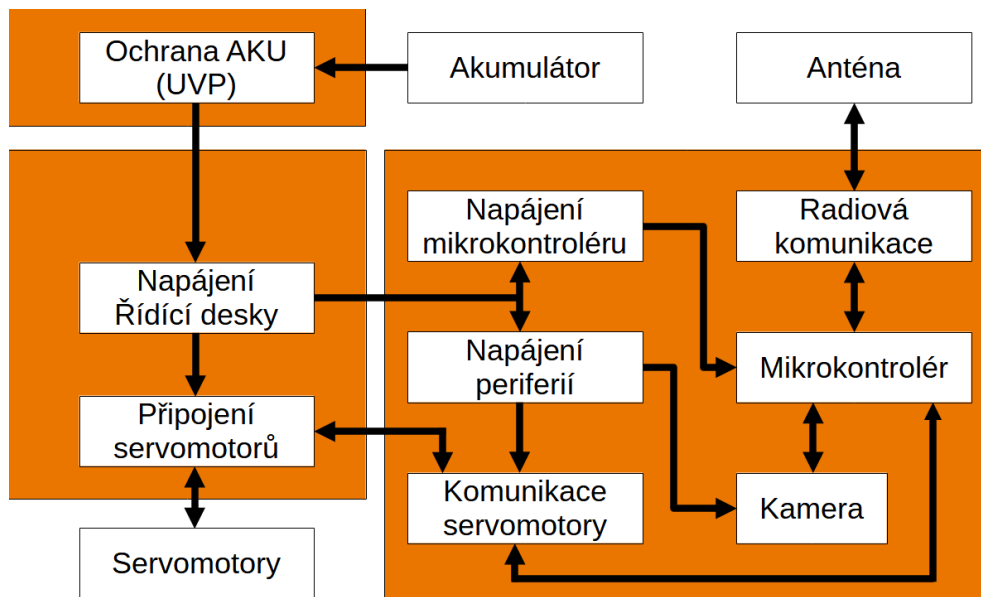
4.5 Řešení elektroniky

Všechny desky byly vyrobeny čínským výrobcem JLCpcb, jenž se zabývá výrobou prototypových DPS už od pěti kusů za nižší ceny. JLCpcb bylo zvoleno s ohledem na dostatečné požadavky technologie výroby, nízkou cenu včetně dopravy, pozitivním hodnocením a rychlou zákaznickou podporou v angličtině. Schémata a desky byly navrženy ve svobodném programu KiCad verze 6.0.

4.5.1 Rozvržení problému

Elektronika v tomto projektu řeší následující problémy: hlídání napětí akumulátoru, napájení pro servomotory, napájení mikrokontroléru a jeho periferií, řídicí část s mikrokontrolérem, komunikace se servopohony, kameru a radiovou komunikaci. Pro jednodušší návrh a oddělení výkonové části od řídicí byla rozdělena celková elektronika do tří desek viz obrázek 4.10 (oranžové obdélníky znázorňují jednotlivé DPS). Na obrázku vlevo nahoře je ochranná deska, pod ní je napájecí deska a vpravo je nejsložitější řídicí deska.

Rozdělením úlohy na více desek bylo také docíleno lepšího využití prostoru. Do středu robota byla umístěna výkonová deska s ochranou akumulátoru a akumulátorem. Tím se docílilo umístění těžších částí do těžiště hexapoda. Naopak řídicí deska byla umístěna naverch robota blíže k přední části, aby bylo při použití kamery dobře využito její zorné pole. Navíc by umístění do ocelové konstrukce rušilo radiovou komunikaci. Oddělením desek se snížil vliv rušení způsobený velkým špičkovým odběrem proudu při pohybu servomotorů. Také byly jednotlivé desky lépe přizpůsobeny proudovému a tepelnému zatížení.



Obrázek 4.10: Blokové schéma rozvržení elektroniky na jednotlivé DPS

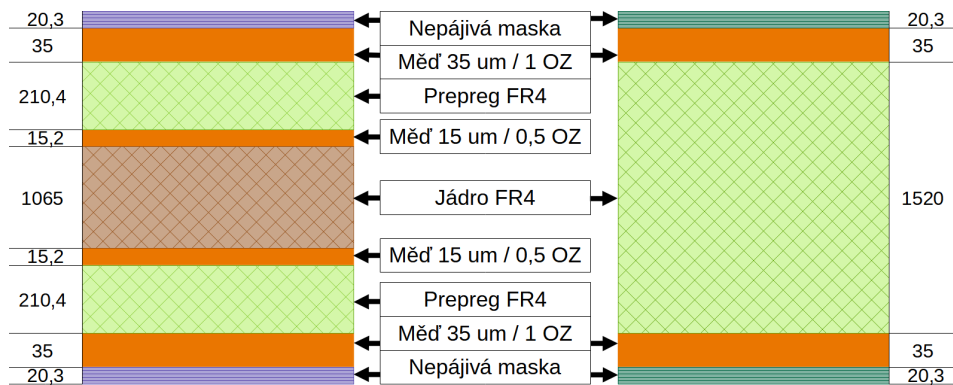
4.5.2 Stackup desek

Pro řídicí desku byla zvolena čtyřvrstvá deska se skladbou vrstev viz obrázek 4.11. Celková tloušťka desky je 1,6 mm, což je standardní tloušťka. Jako základní materiál byl zvolen FR-4. Návrh sice obsahuje 2,4 Ghz radiovou komunikaci, ale požadavky na kvalitu signálu nejsou kritické, a tak je dostatečný podstatně levnější materiál.

První měděná vrstva TOP je použita na osazení SMD součástek a pro většinu spojů. Je zde zakreslena i radiová část. Druhou vrstvu mědi tvoří rozlitá měď s nulovým potenciálem, značená jako GND. Radiová a vysokofrekvenční digitální část se kreslí v první vrstvě, aby přímo pod ní byla vrstva GND pro řízení impedance cesty. Třetí vrstva označená PWR obsahuje rozvod napájení. Poslední vrstva BOT slouží pro signály, které nepotřebují řízení impedance.

Zbylé dvě desky jsou pouze dvouvrstvé se základním materiálem FR-4. Z důvodu osazení výkonového tranzistoru byla pro lepší odvod tepla v úvahu také pro desku bateriové ochrany deska s hliníkovou podložkou (je lehčí a podstatně levnější, než deska s měděnou). Bylo by však nutné mechanické části jako svorkovnice nahradit buď SMD svorkovnicemi s menší mechanickou odolností, nebo svorkovnici přímo nahradit ploškami pro připájení vodičů. Výsledný stackup je vidět na obrázku 4.11. Výsledné tloušťky desek jsou 1,6 mm.

Rozměry uvedené na obrázku 4.11 jsou v mikrometrech. Všechny desky jsou dokončené pomocí technologie HASL (Hot Air Solder Levling), kdy se hotová deska ponoří do kádi roztavené pájky a všechna odhalená měď, která není pokrytá nepájitou maskou, je tak ošetřena pájkou proti korodování.



Obrázek 4.11: Zvolený stackup dvouvrstvých a čtyřvrstvých desek

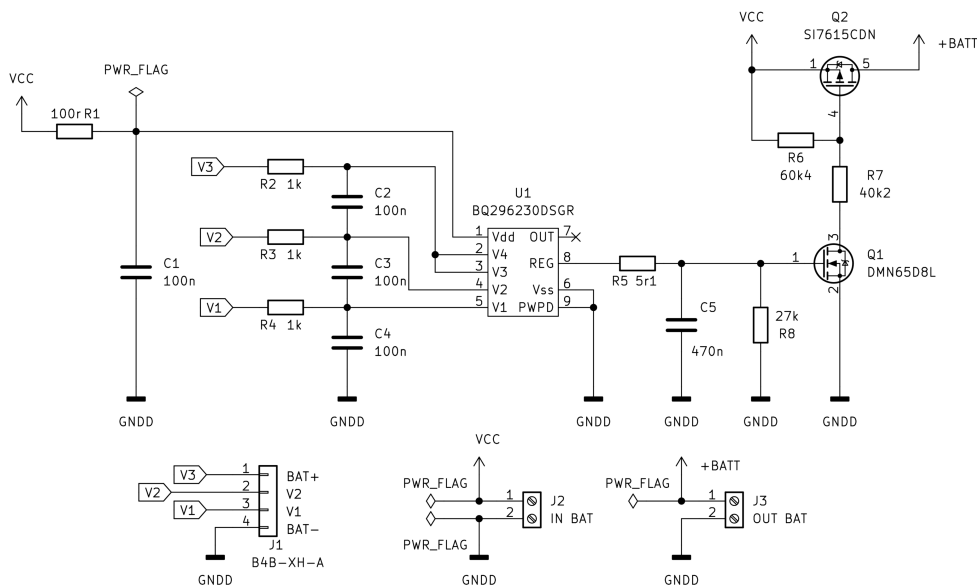
4.5.3 Ochranná deska

Slouží pro odpojení akumulátoru od zbytku obvodů hexapoda, v případě poklesu napětí na jakémkoliv ze článků akumulátoru pod hodnotu 3 V. Vybití článku pod určitou mez často vede k poškození nebo zničení akumulátoru [9]. Pro ochranu a kontrolu aktuálního stavu akumulátoru existuje několik integrovaných obvodů. Ty jsou složitější a kromě teploty, napětí a proudu také počítají uloženou a vydanou energii. V tomto projektu je hlídáno pouze vybití článku pod určité napětí. Tento typ ochrany se často značí jako UV nebo UVP z anglického undervoltage resp. undervoltage protection. Použitý integrovaný obvod je BQ296230 a slouží k nouzovému odpojení akumulátoru v případě přepětí při nabíjení pomocí tavné pojistky. Dále obsahuje regulátor napětí, jehož výstup je vypnut v případě poklesu napětí na libovolném článku pod 3 V. Katalogové schéma bylo upraveno viz schéma 4.12 pro použití výkonového tranzistoru pro odpojení akumulátoru, místo pojistky [1], [2]. V upraveném schématu se využívá výše zmíněná funkce vypnutí výstupu regulátoru při poklesu napětí.

Návrh schématu

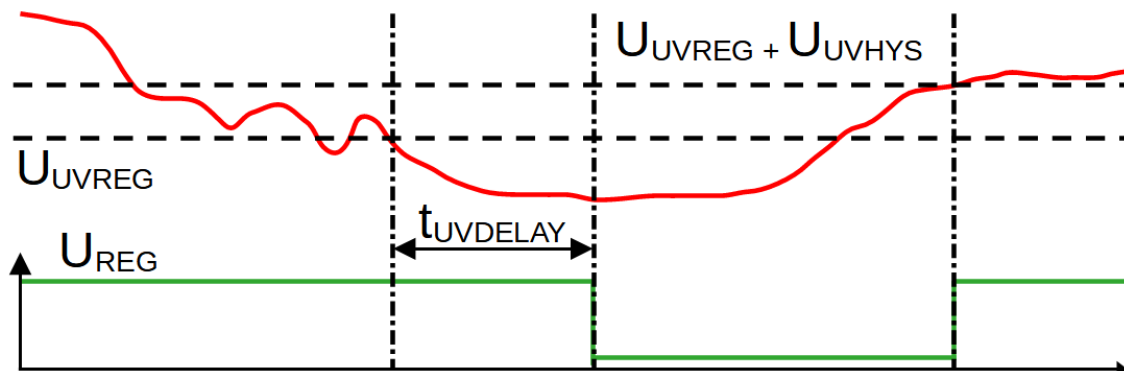
Na obrázku 4.12 je vidět schéma zapojení. Jako Vcc je zde označen kladný terminál z akumulátoru. Ovládaný výstup je označen jako +BATT. Přes tento výstup jsou kabely připojeny na napájecí desku. Konektor J1 je servisní konektor akumulátoru, na němž jsou vyvedeny napětí z jednotlivých článků, které jsou přivedeny na integrovaný obvod podle katalogového zapojení.

Změna oproti katalogovému schéma je v samotném odpojení akumulátoru. Za normálního stavu je na výstupu označeném REG napětí 3 V, které sepne tranzistor s kanálem typu N. Ten sepne výkonový tranzistor s kanálem P a proud z akumulátoru může proudit do zbytku obvodu. V případě poklesu napětí na jakémkoliv článku pod 3 V (hodnota označena jako U_{UVREG}) dojde ke spuštění časovače na dobu $U_{UVDELAY}$ šest sekund. Pokud se napětí na článku do té doby nevrátí na hodnotu 3 V nebo více, dojde k vypnutí REG na 0 V, což způsobí uzavření obou tranzistorů a odpojení akumulátoru od zbytku obvodu. Pokud se hodnota napětí na článku



Obrázek 4.12: Schéma ochranné DPS pro odpojení akumulátoru, inspirováno [1],[2]

později zvýší na U_{UVREG} zvýšenou o hysterezi U_{UVHYS} , tedy $3 + 0,3 = 3,3 V$, dojde k opětovnému sepnutí tranzistorů a připojení akumulátoru [2]. Obrázek 4.13 znázorňuje vypnutí a následné zapnutí regulátoru při výše popsaných podmínkách.

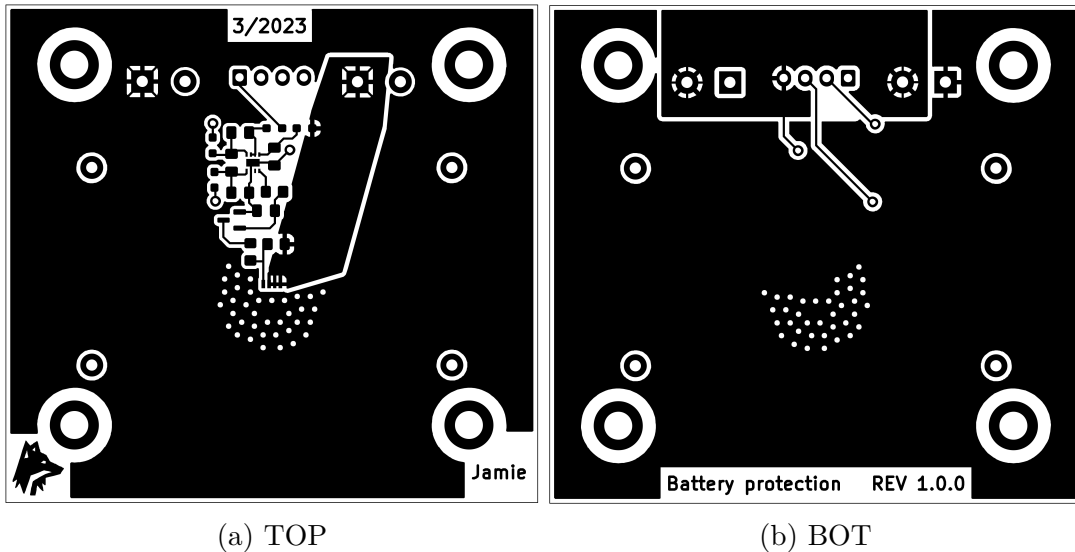


Obrázek 4.13: Průběh napětí regulátoru obvodu BQ296230 závislém na napětí článku akumulátoru, inspirováno [2]

Návrh DPS

Na desce jsou v rozích díry pro uchycení pomocí šroubů M3. Horní část obsahuje servisní konektor a svorkovnice pro připojení akumulátoru a napájecí desky. Pod nimi je integrovaný obvod s dalšími součástkami. Ve středu desky je výkonový SMD tranzistor. Obě vrstvy tvoří převážně rozlitá měď, která slouží pro odvod tepla. Pro lepší odvod tepla jsou v blízkosti tranzistoru prokovy, které část tepelné energie převádějí na druhou vrstvu desky. Výkonové cesty jsou svou šířkou přizpůsobeny

velkým proudům, potřebným pro napájení ostatních desek. DPS dále obsahuje čtyři malé otvory pro uchycení chladiče. Deska má rozměry 62×58 mm. Obě strany DPS jsou na obrázku 4.14.



Obrázek 4.14: Přední a zadní strana ochranné DPS

Výběr výkonového tranzistoru

Protože výkonovým tranzistorem musí protékat relativně velký proud (očekává se až 12 A), musí se vzít v úvahu jeho tepelná ztráta. Podle dokumentace je maximální odpor v sepnutém stavu (při dostatečném napětí na gate) $R_{DSon} = 9 \text{ m}$ [22]. Očekávaný maximální stálý proud je $I = 12 \text{ A}$. Podle výpočtu 4.1 je tedy maximální tepelná ztráta tranzistoru $P_{max} = 1,296 \text{ W}$.

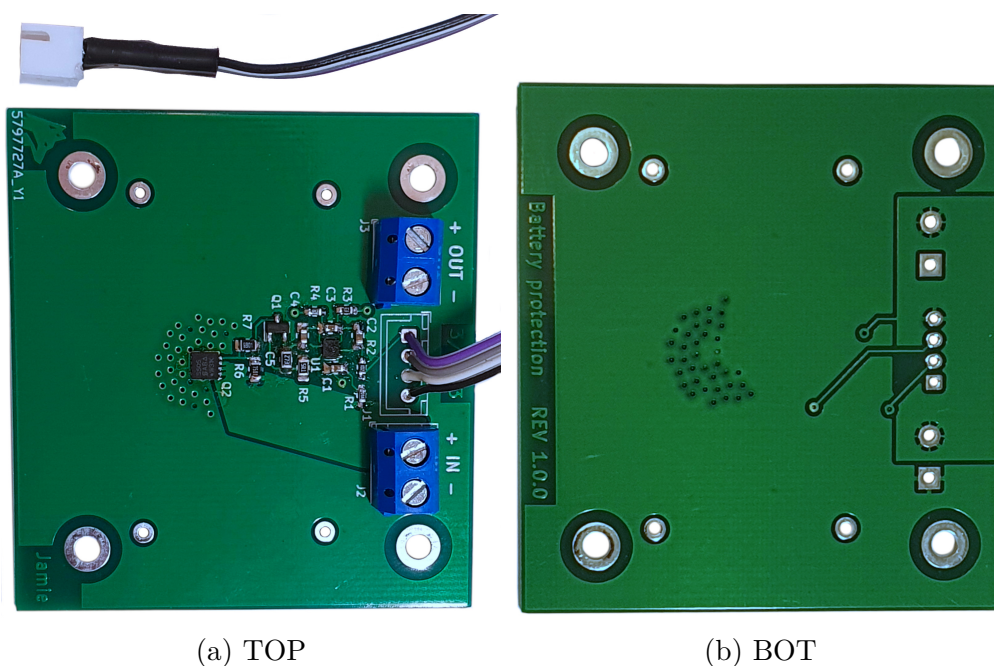
$$P_{max} = R_{DSon} \cdot I^2 = 0,009 \cdot 12^2 = 1,296 \text{ W} \quad (4.1)$$

Maximální stálý tepelný odpor mezi přechodem a prostředím při použití DPS chladiče velikosti přibližně $6,5 \text{ cm}^2$ je podle dokumentace $R_{JAmax} = 81^\circ\text{C}/\text{W}$. V návrhu je dostatek rozlité mědi, aby bylo možné počítat s touto hodnotou. Maximální povolená teplota přechodu je podle dokumentace $T_J = 150^\circ\text{C}$. Jako okolní teplota je zvolena $T_A = 25^\circ\text{C}$, protože robot slouží pro předváděcí účely v běžných podmínkách. Po dosazení hodnot vyjde maximální teplota přechodu $T_J = 130^\circ\text{C}$ viz 4.2.

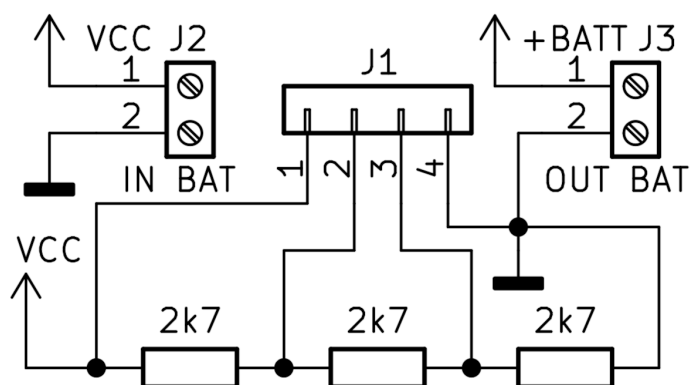
$$T_J = P_{max} \cdot R_{JAmax} + T_A = 1,296 \cdot 81 + 25 = 130^\circ\text{C} \quad (4.2)$$

Osazování a testování

Deska byla ručně osazena a součástky byly napájeny pomocí ruční páječky. Výjimkou byly výkonový tranzistor Q2 a integrovaný obvod BQ296230, které byly zapájeny pomocí horkovzdušné páječky. Nejprve se pájecí plošky těchto dvou součástek a jejich footprinty na DPS opatřily malým množstvím pájky. Následně se aplikovalo na tato místa malé množství tavidla. Deska se poté dala na spodní ohřev, dokud se plně neprohřála. Následně byly tranzistor a integrovaný obvod osazeny a pomocí horkovzdušné páječky prohřáty, dokud se pájka pod nimi neroztavila. Následně bylo provedeno měření multimetrem, zda jsou všechny součástky připájeny a zda nedošlo během pájení k nějakému zkratu. Osazená deska je na obrázku 4.15



Obrázek 4.15: Přední a zadní strana osazené ochranné DPS



Obrázek 4.16: Schéma testování akumulátorové ochrany

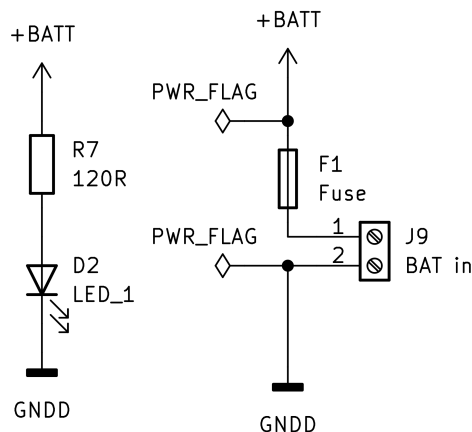
Následně byla testována správná funkce desky. Ze tří rezistorů o stejné hodnotě byl vytvořen odporový dělič napětí, který byl připojen na servisní konektor ochranné desky. Na odporový dělič a vstup desky bylo připojeno napětí 12 V podle obrázku 4.16. Multimetrem bylo naměřeno napětí na výstupu 12 V. Poté bylo napájecí napětí sníženo mírně pod 9 V, čímž byl simulován pokles napětí na článku pod 3 V a napětí na výstupu desky bylo automaticky odpojeno.

4.5.4 Napájecí deska

Tato DPS slouží pro připojení servomotorů a pro napájení kontrolní desky. Je zde měnič z akumulátorového napětí na napětí 5 V, které slouží pro napájení komunikace servomotorů a pro napájení řídicí desky. Řídicí deska je s napájecí propojená pomocí šestimístního IDC konektoru (Insulation-Displacement Connector), protože kromě napájení se přenáší data pro komunikaci se servomotory a napětí z akumulátoru pro měření přibližného stavu nabití.

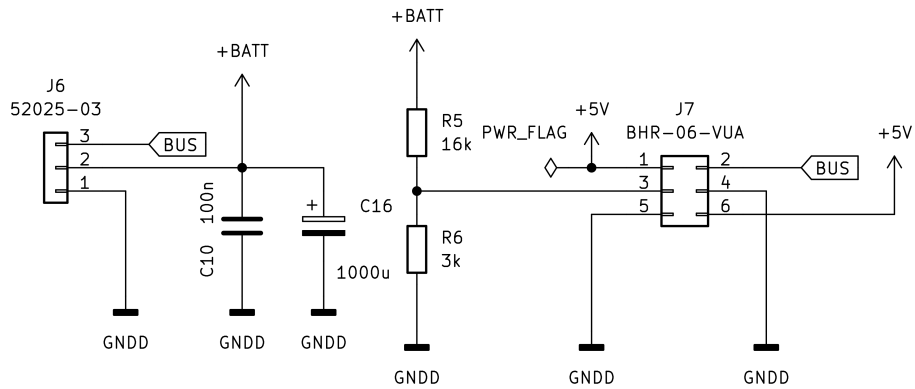
Návrh schématu

Celé schéma je obsaženo v příloze, zde jsou ukázány pouze významné části. Na obrázku 4.17 je vidět vstupní část, indikační LED a rychlá pojistka 30 A.



Obrázek 4.17: Schéma vstupní části napájecí desky

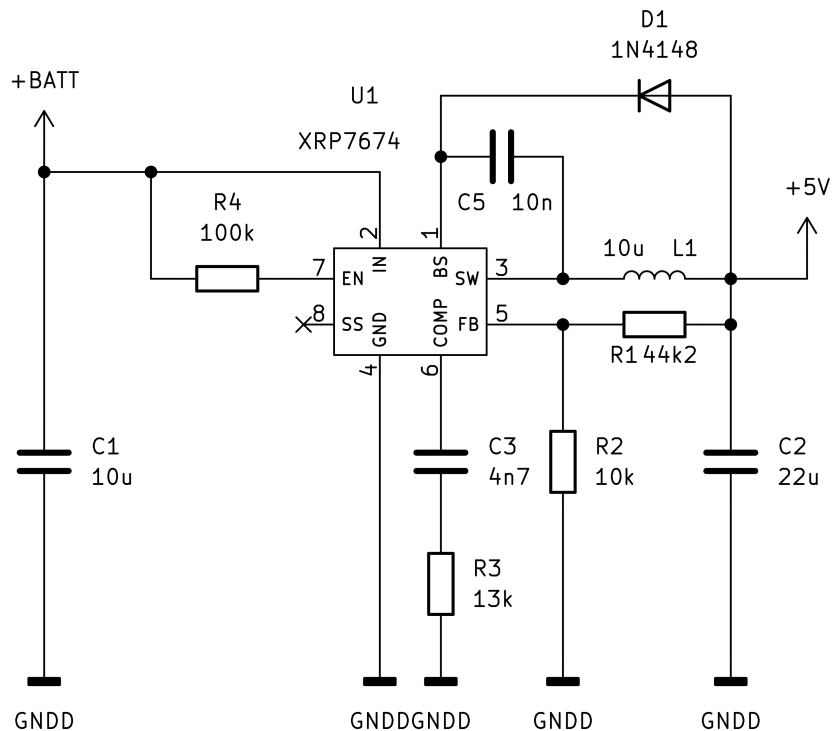
Pro připojení servomotorů se na desce nachází šest konektorů (na obrázku 4.18 je znázorněn pouze jeden). U každého konektoru jsou dva kondenzátory, jeden keramický s kapacitou $C = 100 \text{ nF}$ a druhý elektrolytický kapacitou $C = 10000 \text{ uF}$ pro dodání dostatečného proudu při špičkovém odběru servomotorů. Tím se zajistí lepší napájení pro servomotory a sníží vliv rušení zbytku obvodu. Konektor IDC slouží pro propojení s řídicí deskou. Kromě napájení je zde komunikace přes sběrnici mezi servomotory a mikrokontrolérem. Dále je zde připojeno napětí z akumulátoru přes odporový dělič, který sníží použitelný rozsah napětí z 9–12 V na 1,42–1,94, což je bezpečný rozsah pro použití mikrokontroléru.



Obrázek 4.18: Schéma části s konektory napájecí desky

Poslední část 4.19 tvoří napěťový měnič pro napájení zbytku elektroniky napětím 5 V. Jeho schéma je podle doporučeného zapojení výrobce [23]. Jedinou změnou byla volba rezistoru R1, která se musela dopočítat podle vztahu 4.3, aby výstupní napětí bylo 5 V [23]. Měnič je navržen pro maximální proud 2 A, což je dostatečná rezerva.

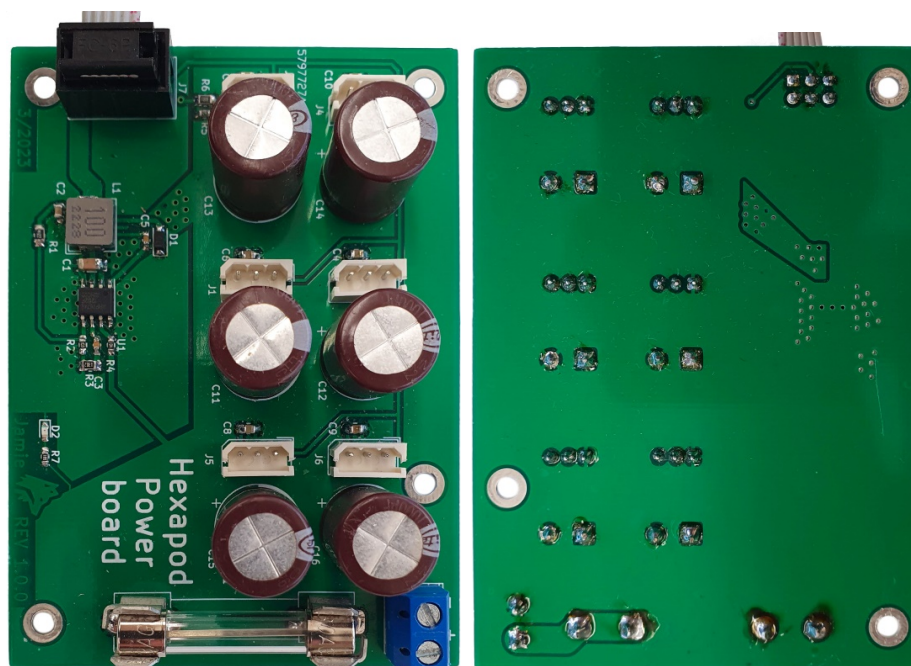
$$R_1 = R_2 \cdot \left(\frac{V_{out}}{0,925} - 1 \right) \quad (4.3)$$



Obrázek 4.19: Schéma části s měničem napětí napájecí desky

Návrh DPS

Deska na obrázku 4.20 obsahuje čtyři otvory pro montáž pomocí šroubů M3. DPS je rozdělena pomyslně na dvě poloviny. Horní část je výkonová. Je zde přívod energie z akumulátoru, respektive ochranné desky, dále pojistka a konektory pro připojení servomotorů a jejich kondenzátory. V dolní části je samotný spínaný zdroj, který je po tomto oddělení více odolný vůči rušení od motorů. Pro lepší chlazení součástky tvoří velkou část rozlévaná měď s termálními prokovy. Cesty jsou dimenzovány pro velký proudový odběr. Rozměry desky jsou 90×65 mm.



(a) Přední osazená strana

(b) Zadní osazená strana

Obrázek 4.20: Přední a zadní strana osazené napájecí desky

Oživení a měření

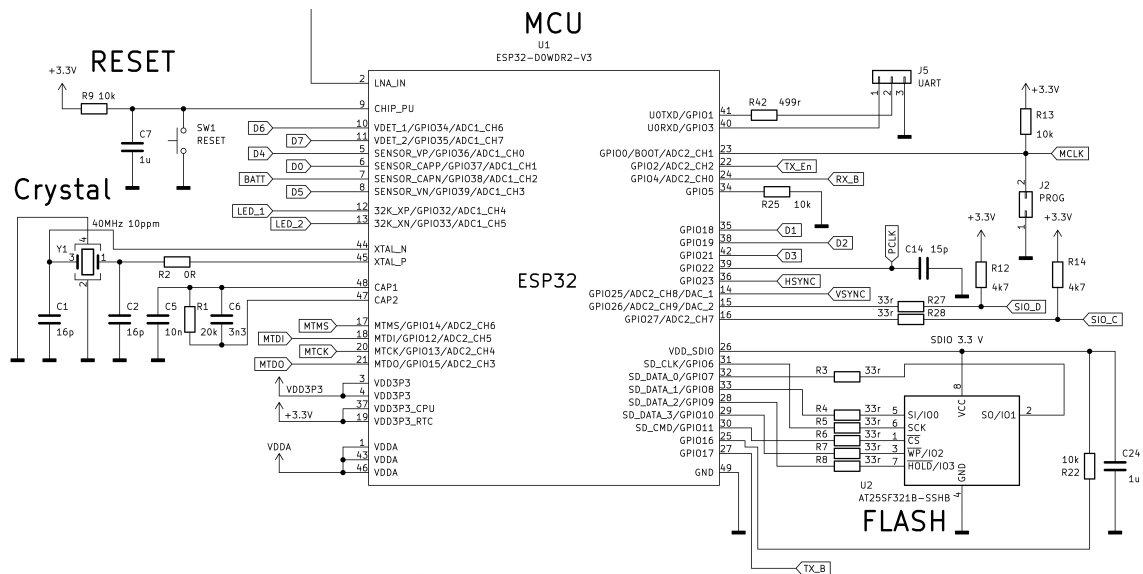
Po osazení a zapájení součástek měniče bylo připojeno napětí 12 V z laboratorního zdroje na vstup a zkontrolován výstup měniče pomocí multimetru. Oproti očekávaným 5 V bylo naměřeno napětí 1,33 V. Po kontrole případných zkratů a studených spojů bylo zjištěno, že místo hodnoty 10 k Ω rezistoru R2, byl chybně napájen rezistor s hodnotou 100 k Ω . Po výměně rezistoru za správný bylo na výstupu měniče naměřeno napětí 5 V. Poté byl napájen zbytek součástek. K desce byl následně připojen jeden servomotor a byla otestována jeho správná funkce pomocí převodníku USB2DYNAMIXEL firmy ROBOTIS. DPS je na obrázku 4.20.

4.5.5 Řídicí deska

Slouží pro řízení servomotorů, radiovou komunikaci s dálkovým ovládáním a snímání okolí pomocí senzorů, například kamery.

Návrh schématu

Zvoleným mikrokontrolérem je ESP32-D0WDR2-V3 viz podkapitola 4.2. Samotný čip vyžaduje ke své funkci další části, jako oscilátor, programovou paměť flash atd. Výřez schématu 4.21 zobrazuje tyto dodatečné obvody a součástky. Při návrhu desky a schématu se postupovalo podle doporučení výrobce [24].



Obrázek 4.21: Schéma části s MCU kontrolní desky

Na obrázku 4.21 vlevo nahoře je tlačítko pro resetování a programování MCU. Pod ním je krystal pro zdroj stabilního kmitočtu. Vpravo nahoře je vyveden konektor J5 pro připojení UART0 pro komunikaci nebo programování viz 4.2. Pod ním je konektor J2 pro uvedení MCU do režimu programování. Vpravo dole je flash paměť pro uložení programu.

Při návrhu bylo zapotřebí dbát na správné použití takzvaných strapping pinů MCU, sloužících při zapnutí MCU pro prvotní inicializaci. Tabulka 4.1 popisuje funkci jednotlivých pinů [18]. Pokud by při startu MCU byl nějaký z pinů v jiné logické hodnotě než je hodnota potřebná, došlo by k chybné inicializaci MCU.

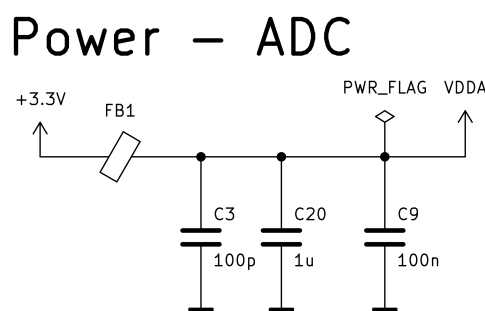
Protože logické hodnoty na pinu jsou snímány jen při startu MCU, lze po inicializaci piny využít i na jiné účely a poté se na nich může objevit i jiná logická hodnota [18]. Pin GPIO18 musí být v log. 0, protože použitá flash paměť potřebuje napájení 3,3 V. Pro normální spuštění programu z flash musí být pin GPIO0 v log. 1. Pro programování přes UART musí být piny GPIO0 a GPIO2 v log. 0. Pin GPIO0 je ve schématu připojen přes pull-up rezistor k log. 1. V případě potřeby programování se propojením konektoru J2 uzemní GPIO0. U zvolené flash

Tabulka 4.1: Strapping piny mikrokontroléru ESP32, inspirováno [18]

Boot mód					
Pin	Výchozí	SPI flash boot		UART programovní	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Nezáleží		0	
Povolení ladících hlášení během programování přes UART					
Pin	Výchozí	Hlášení aktivní		Hlášení vypnuto	
GPIO15	Pull-up	1		0	
Časování flash paměti					
Pin	Výchozí	FE samplování FE výstp	FE samplování RE výstp	RE samplování FE výstp	RE samplování RE výstp
GPIO15	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

paměti je pro správné časování potřebné nastavit vzestupnou hranu pro vzorkování a sestupnou pro výstupní data, proto je pin GPIO5 přes rezistor R25 uzemněn a GPIO15 nechán ve výchozí log. 1.

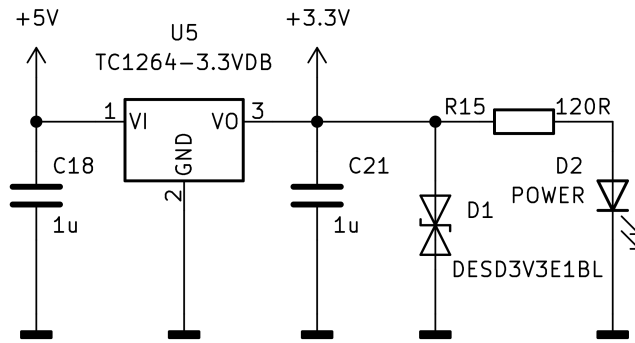
MCU podporuje větší rozsah frekvencí krystalů, ale pro správnou funkci Wi-Fi je potřeba krystal s frekvencí 40 MHz a chybou kmitočtu maximálně 10 ppm [25]. Vybranou SPI NOR flash pamětí je AT25SF321B s velikostí paměti 32 Mbit tedy 4 MiB a maximální frekvencí 108 MHz. Podporuje quad SPI mód, při kterém se využívají čtyři datové vodiče (místo běžných dvou) pro rychlejší přenos dat, čehož je ve schématu 4.21 využito [25].



Obrázek 4.22: Schéma části s blokovacími kondenzátory analogového převodníku MCU

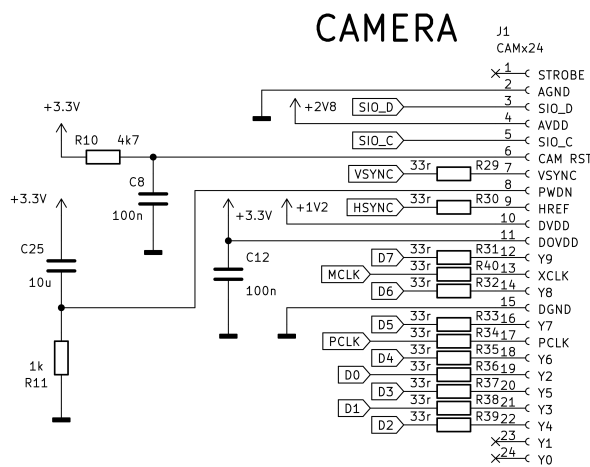
Na obrázku 4.22 je zapojení blokovacích kondenzátorů a feritové perličky pro napájení analogových obvodů MCU. Blokovací kondenzátory různých hodnot kapacit jsou přidány také na další napájecí piny MCU.

Na obrázku 4.23 je lineární stabilizátor napětí na 3,3 V. Pro signalizaci připojení napájení je k němu připojena svítivá dioda. Pro ochranu je k němu přidána ochranná dioda D1 konstruovaná na 3,3 V, která v případě vyššího napětí svede přepětí



Obrázek 4.23: Schéma části s 3,3 V lineárním zdrojem

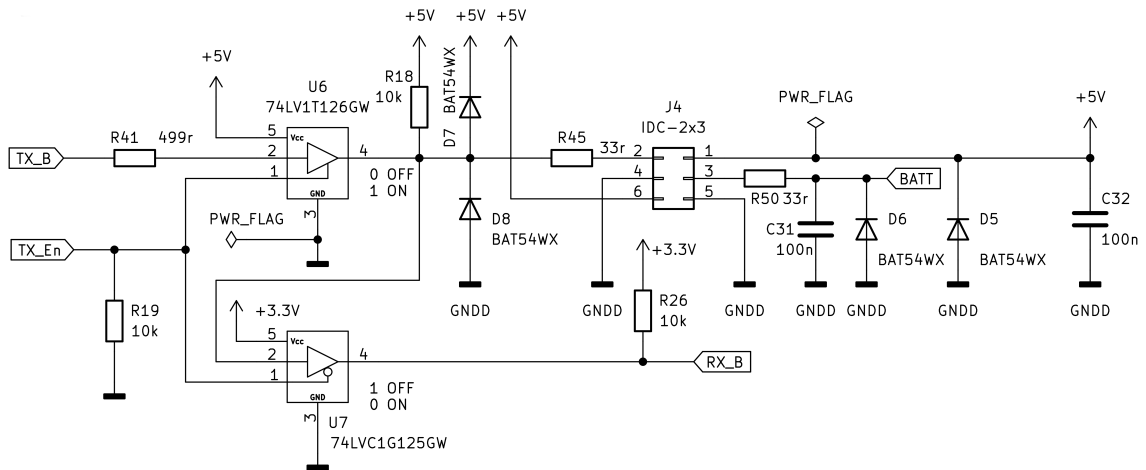
na nulový potenciál. Ve schématu jsou ještě další dva lineární stabilizované zdroje 1,2 V a 2,8 V pro napájení kamery.



Obrázek 4.24: Schéma části s konektorem pro připojení kamery k MCU

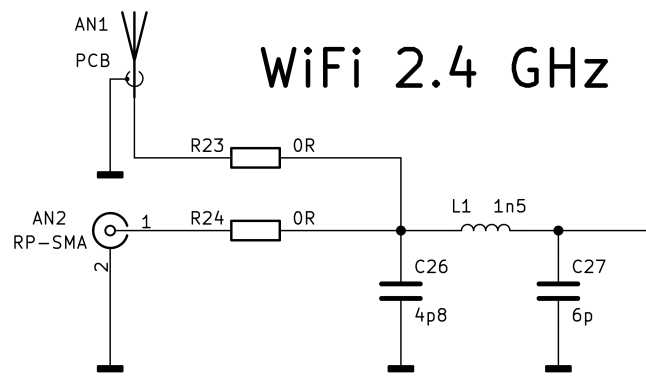
ESP32 obsahuje rozhraní I2S pro připojení některých druhů kamer [18], jako například těch založených na čipu OV2640, pro které je schéma na obrázku 4.24 uzpůsobeno. Zapojení konektoru bylo inspirováno vývojovou deskou ESP32-S3-EYE [26]. Mezi všechny datové a hodinové signály byl vložen sériový rezistor velikosti 33 Ω pro snížení vlivu případných odrazů signálu.

Na obrázku 4.25 je zapojení komunikace se servomotory. Protože servomotory používají sběrnici v polovičním duplexu na napětí 5 V [16] a mikrokontrolér používá sběrnici UART, je potřeba vložit mezi obě sběrnice mezičlánek, který zároveň převede napěťové úrovně mezi jednotlivými sběrnici a také vyřeší převedení polovičního duplexu na celý. Integrované obvody U6 a U7 jsou třístavové buffery, které slouží pro převod logických hodnot z 3,3 V logiky do 5 V logiky a obráceně. Pro svoji funkci jim stačí pouze napájecí napětí cílové logiky. Sběrnice je za normálního stavu nastavena na příjem zpráv pomocí log. úrovně 0 na pinu TX_En. Pokud má MCU vysílat zprávu na sběrnici, přepne TX_En na log. 1 a pin TX_B je tak přes obvod U6 připojen na sběrnici. Pin RX_B je od sběrnice kvůli invertované logice ovládání třetího stavu bufferu odpojen a jeho hodnota je pull-up rezistorem R26 nastavena



Obrázek 4.25: Schéma části s komunikací mezi servomotory a MCU

na log. 1. Po odeslání zprávy MCU přepne pin TX_En zpět do log. 0, vysílací pin TX_B je odpojen a přijímací pin RX_B je přes obvod U7 opět připojen. Schéma je inspirováno doporučením výrobce servomotorů [16].

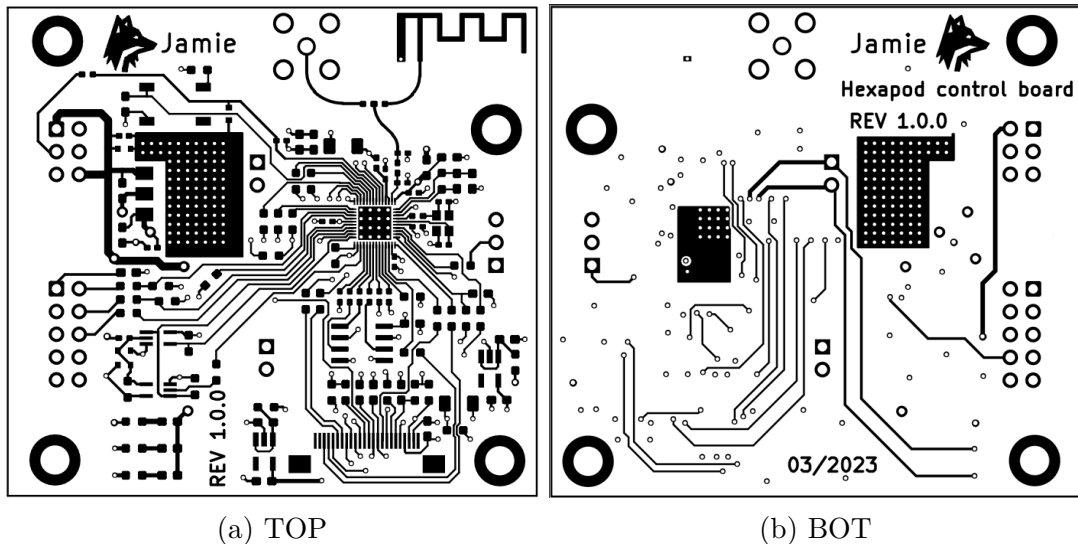


Obrázek 4.26: Schéma s rádiovou částí

Na schématu 4.26 je vidět radiová část používaná pro Wi-Fi. Podle doporučení výrobce ESP32 [18] je zde použit π -článek pro impedanční přizpůsobení MCU a přenosové cesty. Vybraný MCU má impedanci $(35 + 10j) \Omega$. Protože vybrané antény mají impedanci 50Ω 4.4, je i trasa mezi MCU a anténou impedančně přizpůsobena na 50Ω . Pro určení hodnot kondenzátorů a cívek přizpůsobovací sítě (C26, L1, C27) byl využit program Matlab (verze R2022b) s nástrojem Matching Network Designer.

Návrh DPS

Deska obsahuje čtyři otvory pro uchycení pomocí šroubů M3. Horní a dolní strana desky jsou na obrázku 4.27. Vnitřní vrstvy tvoří převážně rozlitá měď s nulovým potenciálem a napájením 3,3 V. Rozměry desky jsou 59×54 mm.



Obrázek 4.27: Přední a zadní strana kontrolní desky

V dolní části se nachází konektor J1 pro připojení kamery. Kousek od něho je většina sériových odporů a blokovací kondenzátory a lineární zdroje pro napájení kamery.

Vlevo jsou konektory IDC pro připojení JTAG rozhraní a pro propojení desky s napájecí deskou. Napájecí trasy s 5 V jsou nejširší (mají šířku 0,9 mm oproti použitým signálovým trasám šířky 0,2 mm), protože vedou nejvíce proudu a vedou rovnou do stabilizátoru, odkud je napětí 3,3 V rozváděno přes druhou vnitřní měděnou vrstvu. Pod 3,3 V stabilizátorem napětí je umístěn velký DPS chladič, aby rozptýlil tepelnou ztrátu vznikající na stabilizátoru viz výpočet 4.4. Nulový potenciál je z přívodního konektoru rovnou veden do první vnitřní měděné vrstvy.

$$P_{max} = I_{max} \cdot (U_{1N} - U_{out}) = 0,8 \cdot (5 - 3,3) = 1,36W \quad (4.4)$$

Blokovací kondenzátory jsou umístěny co nejbližší ESP32, aby co nejlépe plnily svou úlohu. Vpravo od MCU je co nejbližší umístěn krystal. Pod MCU je flash paměť propojená přes sériové rezistory. Zde bylo také dbáno na co nejkratší cestu mezi MCU a pamětí.

Pomalejší periferie jako UART a komunikace se servomotory se nachází dále od MCU. Vlevo dole se nachází indikační svítivé diody, jedna je trvale zapojena a indikuje tak připojené napájení, další dvě je možno ovládat pomocí MCU. Přibližně uprostřed se nachází konektor J2, který při propojení uvede ESP32 do režimu programování pomocí UART.

V horní části je přizpůsobovací π -článek s impedančně přizpůsobenou cestou, která vede do rozbočovače, kde se překlenutím jedné z cest vybere požadovaná anténa. Výpočet požadované tloušťky spoje pro daný stackup byl proveden na impedanční kalkulačce výrobce DPS (obrázek 4.28), kde vyšla tloušťka spoje 0,35 mm [27]. Okolo cest k anténám byl vynechán co největší možný prostor pro zmenšení vlivu rušení komunikace.

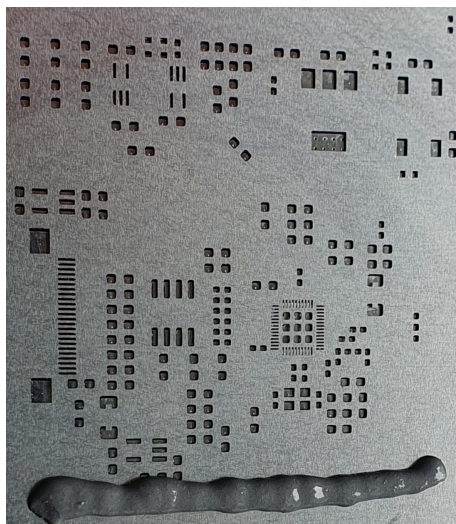
Impedance (Ω)	Type	Signal Layer	Top Ref	Bottom Ref	Trace Spacing
50	Single Ended (Non coplanar)	L1	/	L2	/

Impedance (Ω)	Type	Signal Layer	Top Ref	Bottom Ref	Trace Width
50	Single Ended (Non coplanar)	L1	/	L2	0.3493

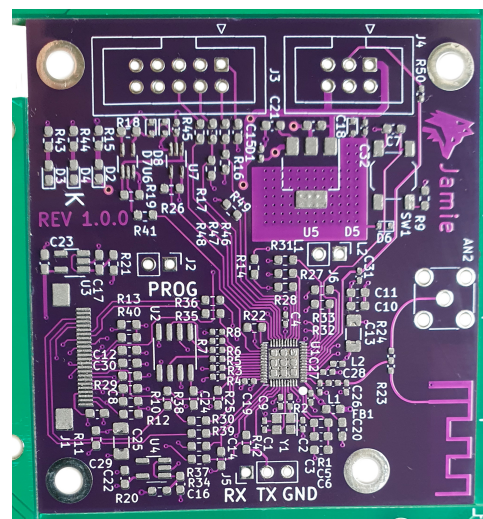
Obrázek 4.28: Kalkulačka šířky impedančně řízeného spoje [27]

Osazení a oživení

Deska byla zapájena pomocí pájecí pasty a horkovzdušné páječky. Nejprve byla deska upevněna na pracovní stůl. Následně na ní byla upevněna šablona z nerezové oceli vyrobená výrobcem desek JLCpcb. Poté byla nanášena a rozetřena vrstva pájecí pasty. Po odebrání šablony byly plošky osazeny součástkami. Poté byla deska přesunuta na spodní předehřev. Po ohřátí desky byly pomocí horkovzdušné páječky ohřívány postupně všechny součástky, dokud pájka neroztála a nevytvořila pevný spoj. Popsaný postup znázorňují obrázky 4.29. Následně bylo pomocí multimetru zkontrolováno, zda se na desce nenachází zkrat a zda jsou všechny spoje zapájené. Poté byl přes rozhraní JTAG nahrán do ESP32 testovací program pro rozblíkáání svítivých diod. Po ověření funkce programu byl nahrán do MCU výsledný řídicí program.



(a) Pájecí pasta na šabloně



(b) Pájecí pasta nanášená na plošky DPS

Obrázek 4.29: Nanesení pájecí pasty na řídicí desku

4.6 Řídicí program

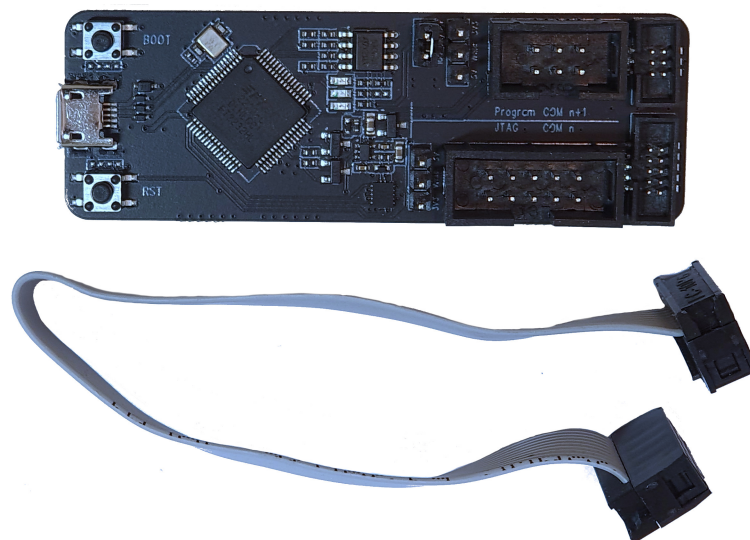
4.6.1 Programování ESP32

Hlavní řídicí algoritmus robota je napsán v jazyce C++. Jedná se o jazyk podporující objektově orientované programování, jehož kód je nejprve přeložen do strojového kódu pomocí kompilátoru na konkrétní cílový procesor, v tomto případě na ESP32, a teprve poté ho lze na procesoru spustit.

Výsledný strojový kód je uložen v externí paměti flash o velikosti 4 MB [18]. Kromě samotného vlastního kódu je také zkompilován a nahrán bootloader, což je program sloužící pro základní inicializaci mikrokontroléru, spuštění hlavní aplikace či dalších operací jako přeprogramování flash přes rozhraní UART (Universal asynchronous receiver-transmitter).

Kód byl napsán ve vývojovém prostředí Visual Studio code, zkráceně VS Code v platformě PlatformIO, zaměřující se nejenom na čipy firmy Espressif Systems, ale i na širokou škálu mikrokontrolerů [28]. Pro programování bylo využito rozhraní JTAG (Joint Test Action Group) z důvodu možnosti vyčítání dat paměti a krokování kódu pro jednodušší ladění kódu.

JTAG je čtyřvodičové sériové rozhraní, které slouží pro přístup do zabudované logiky v mnoha dnešních čipech. Pomocí JTAG rozhraní lze otestovat připojení čipu mezi ostatními čipy bez potřeby dalších externích sond. Lze pomocí něho programovat flash paměti, CPLD (Complex programmable logic device) či FPGA (Field Programmable Gate Array) přímo na desce plošných spojů a ladit program pro mikrokontroléry a mikroprocesory [29].



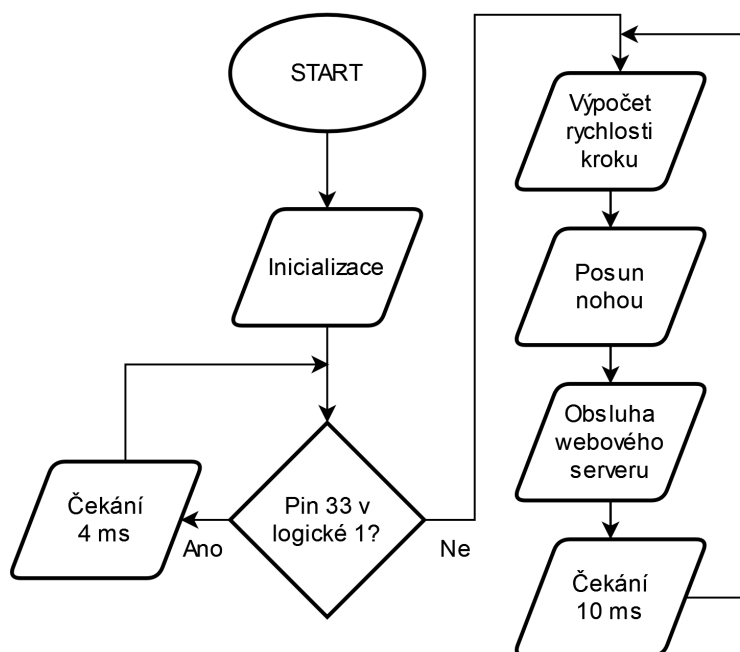
Obrázek 4.30: Programátor ESP-PROG

Jako převodník mezi USB rozhraním počítače a JTAG rozhraním MCU byl použit programátor firmy Espressif System ESP-Prog, nacházející se na obrázku 4.30, který kromě rozhraní JTAG umožňuje programování přes UART [30]. Vývojové prostředí PlatformIO navíc tento převodník podporuje jak pro programování a ladění programu, tak pro sériovou UART komunikaci. Pro propojení řídicí desky s MCU a programátoru byl vytvořen kabel s IDC konektory, který je vidět také na obrázku 4.30.

4.6.2 Hlavní řídicí program

Popis smyčky hlavního programu

Na začátku se vytvoří globální proměnné. Následuje funkce `setup()`, kde se provede inicializace MCU. Jsou zde nastavené vstupní a výstupní piny, sériové komunikace, nastavení Wi-Fi, webového serveru a smyčka, která zamezuje spuštění hlavního programu ve funkci `loop()` při propojení hřebínku J6 na kontrolní desce. Tohoto se využívalo při programování robota, aby se zajistila jeho imobilizace a nestalo se v průběhu připojení k počítači, že se robot začne pohybovat. Po inicializaci (skončení funkce `setup()`) se spustí hlavní program ve funkci `loop()`, která běží v nekonečné smyčce. Nejprve se vypočítá relativní hodnota, o kterou má být posunut krok robota. Následně je řešen posun nohou robota. Poté se zavolá obsluha webového serveru, která aktualizuje získaná data o natočení dálkového ovládání, ze kterých se počítá délka kroku robota. Jako poslední se počká 10 ms a kód ve funkci `loop()` se zase opakuje, jak je znázorněno na zjednodušeném diagramu 4.31.



Obrázek 4.31: Diagram hlavního programu

Používané knihovny a hlavičkové soubory

Program využívá několik knihoven postavených na vývojové platformě Arduino, jež jsou volně ke stažení v prostředí PlatformIO. Tabulka 4.2 popisuje knihovny použité v tomto projektu.

Tabulka 4.2: Použité knihovny

Arduino	Základní knihovna plná funkcí pro práci s periferiemi, piny atd.
cmath	Standardní knihovna pro matematické operace potřebné pro výpočet inverzní kinematiky
WebSocketsServer	Knihovna pro vytvoření socketového serveru, potřebného pro odesílání dat o natočení ovládacího zařízení
HTTPSServer	Knihovna pro vytvoření zabezpečeného serveru

Pro správnou funkci zabezpečeného serveru na ESP32 bylo potřeba upravit knihovnu HTTPSServer, kde v hlavičkovém souboru HTTPConnection.hpp byla nahrazena část `#include <hwcrypto/sha.h>` za `#include <esp32/sha.h>`[31].

Konstanty jako privátní klíč, certifikát a HTML text webové stránky, které by v hlavním souboru main.cpp zabíraly mnoho místa, byly napsány do samostatných hlavičkových souborů a následně vloženy do main.cpp pomocí příkazu `#include`. Tabulka 4.3 popisuje vložené hlavičkové soubory.

Tabulka 4.3: Zahrnuté hlavičkové soubory s konstantami

cert	Obsahuje vygenerovaný certifikát pro zabezpečení spojení
private_key	Obsahuje privátní klíč pro potřeby šifrování zabezpečeného spojení
WebPageSite	Obsahuje konstantní HTML řetězce hlavní webové stránky a chybové webové stránky

Vytvořené funkce

Přehled všech vytvořených funkcí je na ukázce kódu 4.1. Je zde vidět rozdělení funkcí na funkce, které řeší samotné ovládání robota (sekce Robot control) a na funkce, které řeší webový server (Webserver). V ukázce jsou zahrnuty i argumenty funkcí, dále jsou však v této práci funkce psány bez argumentů pro ušetření místa.

```

1 // Functions declaration
2 // Robot control
3 void FillSendData(byte sendData[BUFF_SIZE], byte instruction,
4                 int value);
5 void SendPacket(byte servoID, byte dataLength, byte instruction,
6                 byte data[BUFF_SIZE]);
7 int ReadPacket(int maxLength, int maxWait, byte recData[BUFF_SIZE]);
8 void LegInvKinematic(double x, double y, double z, Leg *leg,
9                     unsigned long timeSpan);
10 int StepPercent(int speed, unsigned long timeSpan, int percent);
11 void StepLandFunc(int length, int percent, legpos legpos);
12 void StepAirFunc(int length, int percent, legpos legpos);
13 int WalkFunction(int percent, int stepLengthLeft,
14                 int stepLengthRight);
15 void ServoSpeedSet(Leg *leg, double anglCoxa, double anglFemur,
16                   double anglTibia, unsigned long timeSpan);
17 // Webserver
18 void onWebSocketEvent(uint8_t client_num, WStype_t type,
19                       uint8_t *payload, size_t length);
20 void handleIndex(HTTPRequest *req, HTTPResponse *res);
21 void handle404(HTTPRequest *req, HTTPResponse *res);

```

Kód 4.1: Seznam všech vytvořených funkcí

Komunikace se servomotory

Pro komunikaci se servomotory MCU využívá druhý UART (ESP32 má celkem tři periferie UART [18]), který byl připojen na piny MCU 4 a 17. Pro přepínání stavu vysílání/příjem slouží pin MCU 2. Ve funkci `setup()` je inicializována UART komunikace a jsou nastaveny potřebné piny. Pro samotné poslání paketu byla vytvořena funkce `SendPacket()`. Funkce používá protokol popsáný v 4.1.1. Argumenty funkce jsou ID servomotoru, instrukce, délka dat a pole bajtů pro odeslání. Funkce nejprve přepne stav na vysílání, spočítá kontrolní součet, pošle hlavičku, veškerá data a kontrolní součet a vrátí nastavení komunikace do režimu příjmu.

Protože data jsou posílána po bajtech, avšak program pracuje s 32bitovými hodnotami, byla vytvořena funkce `FillSendData()`, která naplní pole bajtů pro odeslání ze zadané hodnoty a ze zadané adresy RAM servomotoru.

Pro čtení dat poslaných servomotory byla napsána funkce `ReadPacket()`. Po jejím zavolání MCU cyklicky kontroluje UART, zda nejsou nějaká přijatá data. Pokud ano, zapíše je do pole přijatých bajtů. Pokud je obdrženo počet bajtů zasláný v informaci paketu, je následně proveden kontrolní součet, zda byla data přijata bez chyby. Zároveň funkce hlídá nastavený časový limit (předaný jako argument), pokud by příjem dat trval déle, než jaký je limit, je operace přerušena (považuje se za chybu komunikace). Zároveň si funkce hlídá počet přijatých bajtů, pokud by překročil nastavený limit v argumentu, je tento stav opět považován za chybu komunikace a funkce se ukončí.

Řešení algoritmu chůze

Krok nohy je rozdělen na dvě fáze, pohyb na zemi a pohyb ve vzduchu. Každá tato fáze je rozdělena na 1000 pomyslných dílků nazvaných `percent`, nezávisle na velikosti a rychlosti kroku. Tím, že je aktuální postup fáze vyjádřen hodnotou v dílkách, a pro všechny nohy je tato hodnota stejná, je docíleno toho, že jsou kroky všech noh sladěny. Pokud je aktuální hodnota 0, jsou všechny nohy na startovní pozici své fáze, pokud je hodnota 500, jsou všechny nohy v 50 % fáze. Díky tomu se může změnit rychlost a délka kroku během probíhané fáze (změna způsobena změnou dat z dálkového ovladače) bez rozladění sledu nohou mezi sebou. Nejprve funkce `StepPercent()` vypočítá, o kolik dílků se má posunout fáze kroku. Velikost navýšení hodnoty `percent` je vypočítána z času, který uběhl od minulého zavolání funkce a nastavené rychlosti pohybu. Pokud funkce vyhodnotí, že je ovládací zařízení ve vodorovné poloze (respektive poslaná data o náklonu ze zařízení budou mít hodnotu blízkou 0°), funkce nechá stejnou hodnotu `percent` a nijak ji nenavýší, protože v takové situaci má hexapod stát na místě.

Samotné plánování pohybu fáze nohy ve vzduchu a po zemi řeší funkce `StepAirFunc()` respektive funkce `StepLandFunc()`. Funkce `StepAirFunc()` nejprve spočítá žádanou X pozici konce nohy závislou na aktuální hodnotě `percent` a na délce kroku, jež je předávána jako argument. Zároveň je k hodnotě X připočítán offset závislý na tom, jestli se jedná o přední, střední, nebo zadní nohu. Potom z hodnoty `percent` vypočítá žádanou pozici Z konce nohy. Hodnotu Y má pevně nastavenou po celou dobu této fáze. Funkce `StepLandFunc()` se chová obdobně s tím rozdílem, že je zde jiná hodnota Y a hodnota Z je zde pevně nastavena.

Pro převedení souřadnic X, Y, Z, konce nohy na úhel natočení jednotlivých servomotorů slouží funkce `LegInvKinematic()`. Popis geometrického řešení inverzní úlohy je 2.3. Funkce pro danou nohu vypočítá inverzní úlohu, pošle příkaz natočení jednotlivým servomotorům nohy a zavolá funkci `ServoSpeedSet()`, která pošle jednotlivým servomotorům příkaz, jakou rychlostí se mají na cílové natočení pohybovat, aby byl pohyb plynulý (malý pohyb se provede menší rychlostí, velký pohyb se provede větší rychlostí).

Funkce, která řeší samotný algoritmus chůze a otáčení robota se jmenuje `WalkFunction()`. Funkce přijímá jako argumenty velikost kroku pro levou a pravou stranu (rozdílná velikost kroku způsobuje otáčení robota) a hodnotu `percent` z funkce `StepPercent()`. Funkce nejprve rozhodne, zda je potřeba u všech nohou prohodit fáze pohybu a vynulovat hodnotu `percent`. Následuje cyklus, který podle předem vytvořeného vzoru v poli `legPatern[]` postupně každé noze přiřadí funkci pro potřebnou fázi kroku (fáze ve vzduchu nebo po zemi), vypočítá ze získaných souřadnic konce nohy inverzní kinematiku a pošle příkazy o natočení a rychlosti servomotorů nohy pomocí výše uvedených funkcí. Popis algoritmu chůze robota je v kapitole 2.3.

Řešení ovládání přes webové rozhraní

Pro dálkové ovládání byl zvolen přístup pomocí Wi-Fi a webové aplikace. Popsání výhod a nevýhod tohoto řešení je v sekci 3.2 a 3.6.2.

Webový server

Kód pro obsluhu webového serveru byl inspirován jedním ze vzorových příkladů použití knihovny `HTTPSServer` od autora [32]. Při testování aplikace s vytvořením nejprve nezabezpečeného HTTP serveru nedošlo k odesílání dat ze senzorů zařízení. Poté byla stejná webová aplikace vyzkoušena na zabezpečeném HTTPS webovém serveru. V tomto případě již byla získávána data ze senzorů bez problému. Lze usuzovat, že data ze senzorů jsou považována za citlivá a webový prohlížeč v zařízení je proto nechce v nezabezpečené komunikaci poskytnout (aplikace byla testována na zařízeních s operačním systémem Android verze 12, iOS verze 15.7 a Windows 11 verze 10.0.2262).

Pro vytvoření zabezpečené komunikace pomocí SSL je zapotřebí certifikát a privátní klíč[33]. Pro vytvoření self-signed certifikátu a privátního klíče byl použit shell skript napsaný autorem knihovny `HTTPSServer.hpp`, využívající svobodný nástroj OpenSSL pro tvorbu SSL certifikátů. Skript je dostupný na [34]. Pro spuštění skriptu na operačním systému Windows 11 byl nainstalován Windows Subsystem for Linux, podporující běh operačního systému Linux, pro který byl skript napsán.

Jako první je ve funkci `setup()` zavolána funkce `WiFi.softAP()` která nastaví ESP32 do režimu Wi-Fi přístupového bodu. Potom je vytvořen zabezpečený server, kde je nastaven maximální počet klientů, port na kterém HTTPS server přijímá spojení a SSL certifikát s privátním klíčem. Serveru jsou přiřazeny takzvané node, které řeší reakci serveru na žádost o webovou stránku a posílání dat přes socket. Následným zavoláním funkce `secureServer.start()` je zabezpečený server spuštěn. V hlavní smyčce `loop()` je při každém průchodu zavolána funkce `secureServer.loop()`, která řeší obsluhu jednotlivých node a nových příchozích spojení.

Při vytváření node pro webový socket, byla vytvořena funkce `onMessage()`, která reaguje na příchod zprávy ze zařízení. Pro ladící účely nejprve pošle zařízení potvrzovací zprávu, poté přijatou zprávu převede na `string`, který převede na čísla, která odpovídají orientaci zařízení a poté z nich dopočítá délky a rychlost kroků.

Webová aplikace

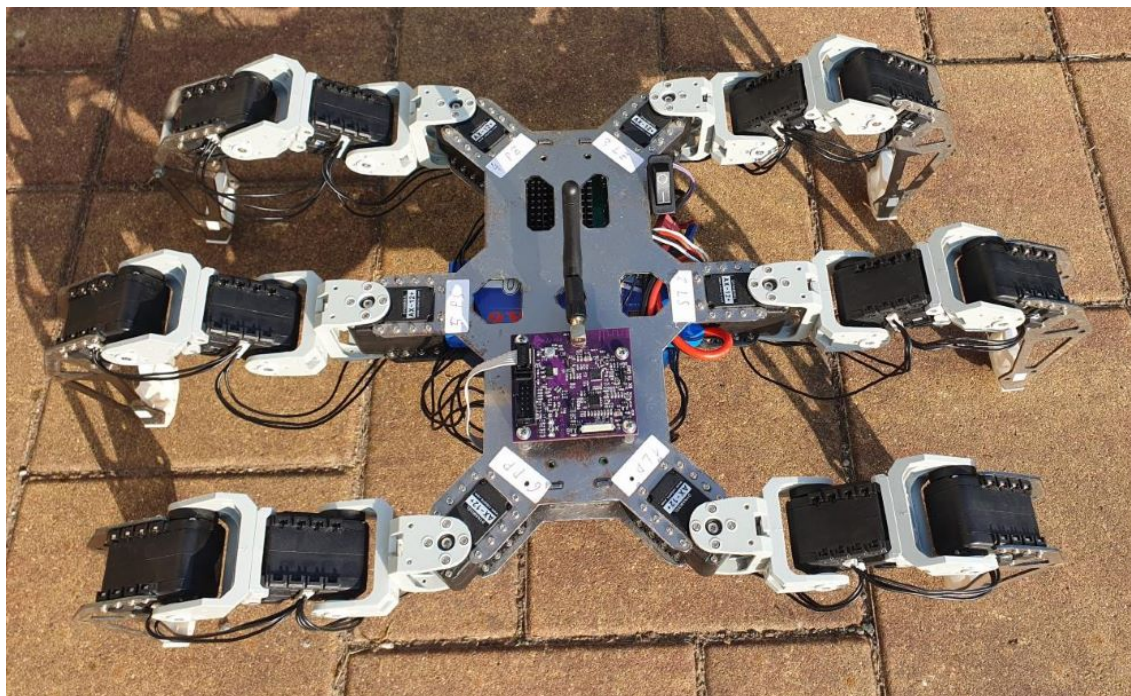
Samotná webová aplikace napsána v HTML a JavaScript je inspirována [35]. Samotný HTML kód slouží pro vykreslení tlačítka a zobrazení informací o aktuálním naklonění zařízení. JavaScript kód nejprve naváže zabezpečené socket spojení se serverem. Dále vytvoří událost, čekající na zmáčknutí tlačítka. Při stisku tlačítka se

vytvoří nová událost, jež volá funkci `handleOrientation()`, která snímá orientaci zařízení. Zároveň se při stisku tlačítka spustí funkce `startOrientationUpdates()` a nastaví časovač na 200 ms, po němž se spustí funkce `sendOrientationData()`. Tato funkce pošle naměřená data přes socket webovému serveru a zavolá funkci `startOrientationUpdates()`, která opět nastaví časovač na 200 ms a celý proces se opakuje. Tím je zajištěno, že se data posílají v daném intervalu.

4.7 Zkompletovaný hexapod

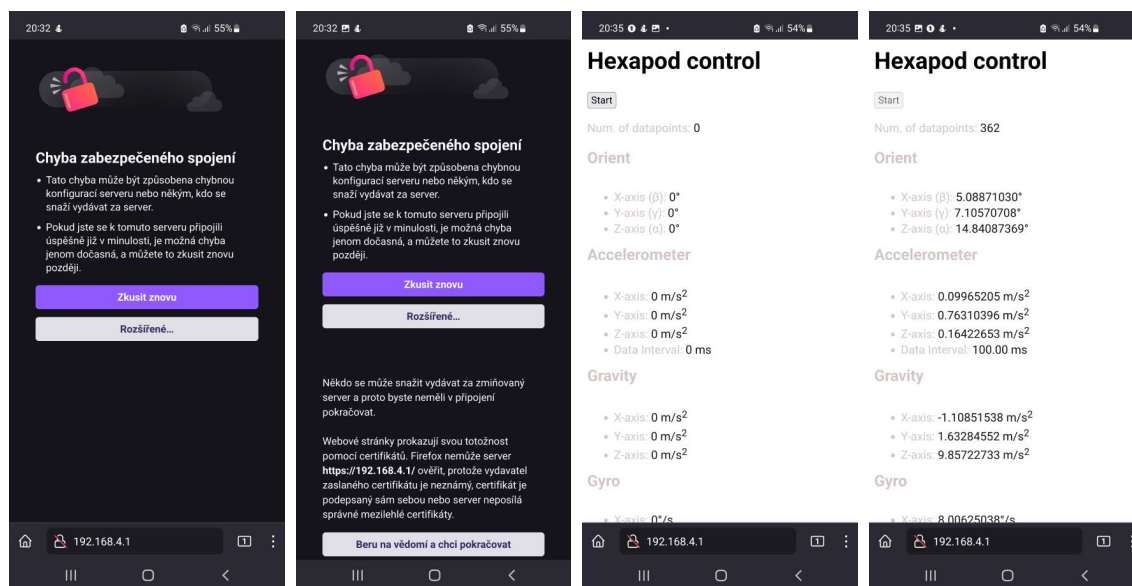
Samotná kostra se servomotory váží 2204 g. Výsledný hexapod s elektronikou včetně akumulátoru váží 2,45 kg a v základním postavení má výšku 146 mm, délku 296 mm a šířku 554 mm. Po vypnutí se dá složit do výšky 127 mm, délky 379 mm a šířky 281 mm.

V základním postavení odebírá robot z baterie proud přibližně 1,29 A. Samotná elektronika (bez kamery) odebírá proud přibližně 91,5 mA (odebíraný proud závisí na napětí akumulátoru). Při chůzi je odběr proudu okolo 3,98 A. Aby se zajistil dlouhý provoz na předváděcích akcích a veletrzích, byly zakoupeny dva stejné akumulátory, kde jeden je umístěn v hexapodu a druhý je mezitím nabíjen a poté se prohodí. Výměna akumulátoru začíná odšroubováním čtyř šroubků ve spodní části robota. Poté se vysune akumulátorová ochrana a odpojí se akumulátor (pouze jeho konektor silového přívodu a servisní konektor). Následně se vymění vybitý akumulátor za nový a opětovně se připojí. Pak se připojí bateriová ochrana a zašroubují se šroubky. Jedná se tedy o jednoduchou a rychlou činnost.



Obrázek 4.32: Výsledný hexapod

Zapnutí robota spočívá v přepnutí vypínače na boku robota do polohy zapnuto. Poté se připojí mobilní zařízení k Wi-Fi pojmenované **HEXAPOD-FM-TUL** pomocí hesla. Po připojení k hexapodu je zapotřebí otevřít webový prohlížeč a připojit se k adrese <https://192.168.4.1>. Protože certifikát je podepsaný neověřenou autoritou (self-signed certifikát), zobrazí se pravděpodobně upozornění na nebezpečnou stránku. Pro pokračování na stránku je často potřeba potvrdit tuto skutečnost kliknutím na tlačítko „Přijmout rizika“, nebo na možnost „Rozšíření“ a poté kliknout na „Vstoupit na stránku“ (různé prohlížeče se mohou v tomto lišit). Následně se zobrazí jednoduchá stránka, kde se kliknutím na tlačítko „Start“ spustí ovládání hexapoda pomocí natačení mobilního zařízení (pro správné fungování musí mít zařízení senzor natočení). Popsaný postup znázorňují obrázky 4.33.



(a) Zadání adresy do webového prohlížeče (b) Kliknutí na "rozšíření" a přijetí rizik (c) Otevření webové stránky (d) Spuštění webové aplikace tlačítkem "Start"

Obrázek 4.33: Postup spuštění webové aplikace pro ovládání hexapoda

Hexapod byl testován na různém povrchu, jako vinylová podlaha, silnice, šterk a na travnatém povrchu. Díky kráčejícímu podvozku však neměl na žádném uvedeném povrchu větší problém a menší nerovnosti zdolal bez obtíží. Maximální naměřená rychlost chůze byla 0,15 m/s, tedy 0,54 km/h. Tuto hodnotu lze však navýšit v programu zvětšením délky kroku a zvětšením výpočtu dílků kroků. Výsledný hexapod je na obrázku 4.32.

Závěr

Cílem bakalářské práce bylo vytvořit elektroniku pro dálkové ovládání robota typu hexapod s použitím 32bitového mikrokontroléru. Samotnou kostru robota se servomotory jsem převzal z diplomové práce Ing. Jakuba Hirňšala [3] a dosavadní elektroniku založenou na minipočítači Raspberry Pi společně s deskou Arduino jsem nahradil vlastními deskami. V rešeršní části jsem se zabýval výběrem vhodného mikrokontroléru a antény, možnostmi ovládání servomotorů, výběrem vhodných DPS a jejich výrobcem, výběrem vhodného akumulátoru, průzkumem hotových programů, použitých knihoven, vývojových prostředí a základu psaní webové aplikace pomocí HTML a jazyku JavaScript.

S novou elektronikou se docílilo až 4krát menší spotřeby samotné elektroniky, než při použití Raspberry Pi. Navíc byla přidána akumulátorová ochrana v podobě pojistek a vlastní navržené ochranné desky, která odpojí akumulátor při podbití, aby se zamezilo poškození akumulátoru. Původní kupovaný spínaný zdroj jsem nahradil vlastní navrženou napájecí deskou se spínaným zdrojem a konektory s mechanickými zámkami kompatibilními s konektory servomotorů. Oproti původnímu návrhu jsem tak zamezil možnost zapojit omylem konektor servomotorů obráceně. Pro dálkové ovládání jsem zvolil komunikaci pomocí Wi-Fi.

Kromě samotné elektroniky jsem vytvořil nový řídicí program v jazyce C++ s použitím prostředí Arduino, který řídí servomotory, plánuje trajektorii a vytváří přístupový bod pro Wi-Fi a webový server pro dálkové ovládání. Dále jsem napsal jednoduchou webovou aplikaci pro vytvoření zabezpečeného spojení mezi zařízením a hexapodem. Webovou aplikaci jsem napsal tak, aby ji bylo možno spustit na mobilních zařízeních nezávisle na operačním systému a webovém prohlížeči. Aplikace byla otestována na operačních systémech Windows, iOS a Android a ve webových prohlížečích Firefox, Safari, Google Chrome a Samsung Internet. Pro funkci je nutné, aby mobilní zařízení bylo schopné posílat data o svém naklonění respektive o orientaci.

Práce byla splněna ve všech bodech. Nad rámec zadání jsem navrhl akumulátorovou ochranu a přidal do návrhu možnost připojit kameru. Do budoucna lze na volné vyvedené piny mikrokontroléru připojit různé periferie, a tím rozšířit hexapoda o různé snímače, například o ultrazvukové senzory pro detekování překážek. Dále lze připojit k připravenému konektoru kameru, která společně s využitím neuronových sítí může vylepšit robota o určitou míru autonomie. Lze i přenést určité výpočty a práci s obrazem na mobilní zařízení využívané jako dálkové ovládání, a tím některé operace urychlit.

Použitá literatura

- [1] MATT_MIELKE. *Simple Undervoltage and Overcurrent Protection for Lithium-Ion and Lithium Polymer Batteries* [online]. ©1995-2023. [cit. 2023-05-20]. Dostupné z: <https://forum.digikey.com/t/simple-undervoltage-and-overcurrent-protection-for-lithium-ion-and-lithium-polymer-batteries/13123>.
- [2] TEXAS INSTRUMENTS INCORPORATED. *BQ296xxx Overvoltage Protection* [online]. 2013-2022. [cit. 2023-05-20]. Dostupné z: https://www.ti.com/lit/ds/symlink/bq2961.pdf?HQS=dis-dk-null-digikeymode-dsf-pf-null-ww&ts=1677608980891&ref_url=https%253A%252F%252Fwww.ti.com%252Fgeneral%252Fdocs%252Fsupproductinfo.tsp%253FdistId%253D10%2526gotoUrl%253Dhttps%253A%252F%252Fwww.ti.com%2525.
- [3] HIRNŠAL, Jakub. *Návrh řídicího systému pro Hexapod s Raspberry Pi* [online]. Studentská 1402/2, 461 17 Liberec 1, 2018-12-02T10:55:57Z [cit. 2023-04-18]. Dostupné z: <https://dspace.tul.cz/handle/15240/49734>. diplomové práce. Technická univerzita v Liberci, Fakulta mechatroniky, informatiky a mezioborových studií.
- [4] KOLÍBAL, Zdeněk. *Roboty a robotizované výrobní technologie*. První vydání. Brno: Vysoké učení technické v Brně - nakladatelství VUTIUM, 2016. ISBN 978-80-214-4828-5.
- [5] LIANG, Oscar. *Inverse Kinematics Basics Tutorial* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: <https://oscarliang.com/inverse-kinematics-and-trigonometry-basics/>.
- [6] ZAK, Marek, Jaroslav ROZMAN a Frantisek ZBORIL. Design and Control of 7-DOF Omni-directional Hexapod Robot. *Open Computer Science*. 2020, roč. 11, s. 80–89. Dostupné z DOI: [10.1515/comp-2020-0189](https://doi.org/10.1515/comp-2020-0189).
- [7] LEA, Perry. *IoT and edge computing for architects: implementing edge and IoT systems from sensors to clouds with communication systems, analytics, and security*. Second edition. Birmingham: Packt, 2020. ISBN 978-1-83921-480-6.
- [8] BEVELACQUA, Peter Joseph. *Antenna-Theory.com* [online]. ©2009-2022. [cit. 2023-05-31]. Dostupné z: <https://www.antenna-theory.com/>.
- [9] WU, Yuping. *Lithium-ion batteries: fundamentals and applications*. Boca Raton: CRC Press, [2015]. ISBN 978-1-4665-5733-8.

- [10] ZÁHLAVA, Vít. *Návrh a konstrukce desek plošných spojů: principy a pravidla praktického návrhu*. 1. vyd. Praha: BEN - technická literatura, 2010. ISBN 978-80-7300-266-4.
- [11] PETERSON, Zachariah. *FR4 Dielectric Constant and Material Properties* [online]. ©2023. [cit. 2023-05-29]. Dostupné z: <https://resources.altium.com/p/fr4>.
- [12] ALLEN, Grant. *Android 4: průvodce programováním mobilních aplikací*. 1. vyd. Brno: Computer Press, 2013. ISBN 9788025137826.
- [13] *Kotlin for Android* [online]. [cit. 2023-05-30]. Dostupné z: <https://kotlinlang.org/docs/android-overview.html>.
- [14] MICROSOFT. *Co je Xamarin?* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: <https://learn.microsoft.com/cs-cz/xamarin/get-started/what-is-xamarin>.
- [15] *Swift: The powerful programming language that is also easy to learn*. [online]. [cit. 2023-05-30]. Dostupné z: <https://developer.apple.com/swift/>.
- [16] *AX-12A* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/>.
- [17] ROBOTIS. *DYNAMIXEL Protocol 1.0* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: <https://emanual.robotis.com/docs/en/dxl/protocol1/>.
- [18] ESPRESSIF SYSTEMS. *ESP32 Series: Datasheet* [online]. Version 4.2. Shanghai, ©2023 [cit. 2023-05-09]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.
- [19] *ESP32-S3-WROOM-2: Datasheet* [online]. Version 1.1. Shanghai, ©2023 [cit. 2023-05-19]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-2_datasheet_en.pdf.
- [20] BIG HOBBY.CZ. *Power X6 4200 mAh 3S 35C (70C)* [online]. ©2016-2023. [cit. 2023-05-19]. Dostupné z: <https://www.bighobby.cz/power-x6-4200-mah-3s-35c--70c/>.
- [21] FERANEC, Robert a Giorgia ZUCHELLI. *How to Design and Simulate PCB Antenna* [online]. [cit. 2023-05-19]. Dostupné z: <https://www.youtube.com/watch?v=Rke7d9MKOLs>.
- [22] ALPHA & OMEGA SEMICONDUCTOR. *AON7423: 20V P-Channel MOS-FET* [online]. Rev 0. 2011. [cit. 2023-05-20]. Dostupné z: <https://www.tme.eu/Document/f617947f937affdcef3d2ac1a885827c/AON7423-DTE.pdf>.
- [23] EXAR CORPORATION. *XRP7674* [online]. Rev. 1.0.0. Fremont, 2013 [cit. 2023-05-20]. Dostupné z: https://assets.maxlinear.com/web/documents/xrp7674_ds_r100_022813.pdf.
- [24] ESPRESSIF SYSTEMS. *ESP32: Hardware Design Guidelines* [online]. Version 3.3. Shanghai, ©2022 [cit. 2023-05-09]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf.

- [25] RENESAS ELECTRONICS. *AT25SF321B: Datasheet* [online]. Rev. F. ©2022. [cit. 2023-05-20]. Dostupné z: <https://www.renesas.com/us/en/document/dst/at25sf321b-datasheet?r=1608806>.
- [26] ESPRESSIF SYSTEMS. *ESP32-S3-EYE v2.2* [online]. ©2023. [cit. 2023-05-09]. Dostupné z: https://github.com/espressif/esp-who/blob/master/docs/en/get-started/ESP32-S3-EYE_Getting_Started_Guide.md.
- [27] JLCPCB.COM. *JLCPCB Impedance Calculator* [online]. ©2023. [cit. 2023-05-24]. Dostupné z: <https://jlcpcb.com/pcb-impedance-calculator/>.
- [28] *Development Platforms* [online]. [cit. 2023-05-16]. Dostupné z: <https://docs.platformio.org/en/latest/platforms/index.html>.
- [29] JTAG TECHNOLOGIES. *What is JTAG boundary-scan?* [online]. ©2022. [cit. 2023-05-17]. Dostupné z: <https://www.jtag.com/what-is-jtag-boundary-scan/>.
- [30] ESPRESSIF SYSTEMS. *ESP-Prog* [online]. ©2023. [cit. 2023-05-09]. Dostupné z: https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/other/esp-prog/user_guide.html.
- [31] YET ANOTHER FUZZI, Bojan JURCA a Peter MAJCHRAK. *Hwcrypto/sha.h: No such file or directory* [online]. ©2023. [cit. 2023-05-17]. Dostupné z: https://github.com/fhessel/esp32_https_server/issues/143.
- [32] HESSEL, Frank. *Websocket-Chat.ino* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: https://github.com/fhessel/esp32_https_server/blob/master/examples/Websocket-Chat/Websocket-Chat.ino.
- [33] *SSL protokol* [online]. ©2008-2023. [cit. 2023-05-19]. Dostupné z: <https://www.ssl-certifikaty.cz/o-certifikatech/ssl-protokol/>.
- [34] HESSEL, Frank. *Create_cert.sh* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: https://github.com/fhessel/esp32_https_server/blob/master/extras/create_cert.sh.
- [35] ACAR, Gunes et al. *Demo.html* [online]. ©2023. [cit. 2023-05-19]. Dostupné z: <https://github.com/sensor-js/sensor-js.github.io/blob/master/demo.html>.

A Přílohy

A.1 Kusovníky

Kusovník ochranné desky [A.1](#)

Kusovník napájecí desky [A.2](#)

Kusovník kontrolní desky [A.3](#)

A.2 Schémata

Schéma ochranné desky [A.1](#)

Schéma napájecí desky [A.2](#)

Schéma kontrolní desky [A.3](#)

Tabulka A.1: Kusovník ochranné DPS

Označení	Počet	Hodnota	Značka	Footprint
C1, C2, C3, C4,	4	100n	C	0805
C5,	1	470n	C	0805
J1,	1	B4B-XH-A	Conn_01x04	JST_XH_1x04_P2.50mm
J2,	1	IN BAT	Terminal_01x02	Terminal_1x02_P5.00mm
J3,	1	OUT BAT	Terminal_01x02	Terminal_1x02_P5.00mm
Q1,	1	DMN65D8L	DMN67D7L	SOT-23
Q2,	1	SI7615CDN	Si7141DP	PowerPAK_1212-8_Single
R1,	1	100r	R	0603
R2, R3, R4,	3	1k	R	0603
R5,	1	5r1	R	0805
R6,	1	60k4	R	0805
R7,	1	40k2	R	0805
R8,	1	27k	R	0805
U1,	1	BQ296230DSGR	BQ2962xx	WSO8-8-1EP

Počet komponent 19

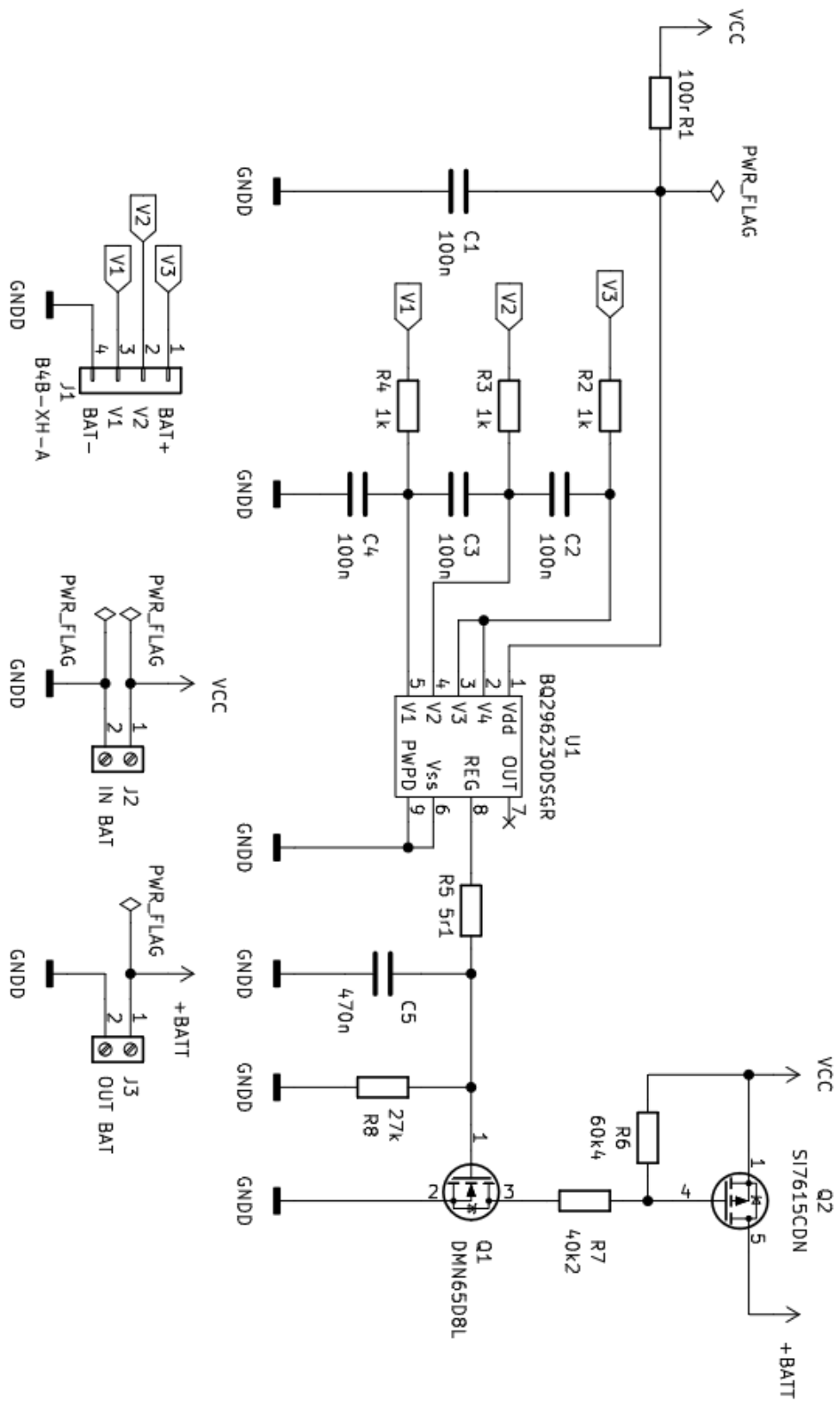
Tabulka A.2: Kusovník napájecí DPS

Označení	Počet	Hodnota	Značka	Footprint
C1,	1	10u	C	1206
C2,	1	22u	C	0603
C3,	1	4n7	C	0603
C4, C6, C7, C8, C9, C10,	6	100n	C	0805
C5,	1	10n	C	0603
C11, C12, C13, C14, C15, C16,	6	1F	C_Pol	D12.5mm_P5.00mm
D1,	1	1N4148	D	SOD-123
D2,	1	LED_1	LED	0603
F1,	1	Fuse	Fuse	6.3x32mm
J1, J2, J3, J4, J5, J6,	6	52025-03	Conn_01x03	52025-03-0100-01-TH
J7,	1	BHR-06-VUA	Conn_02x03	IDC-2x03_P2.54mm
J9,	1	BAT in	Terminalx02	02xP5.00mm
L1,	1	10u	L	LMLP07A7
R1,	1	44k2	R	0805
R2,	1	10k	R	0603
R3,	1	13k	R	0805
R4,	1	100k	R	0603
R5,	1	16k	R	0603
R6,	1	3k	R	0805
R7,	1	120R	R	0603
U1,	1	XRP7674	XRP7674	SOIC-8

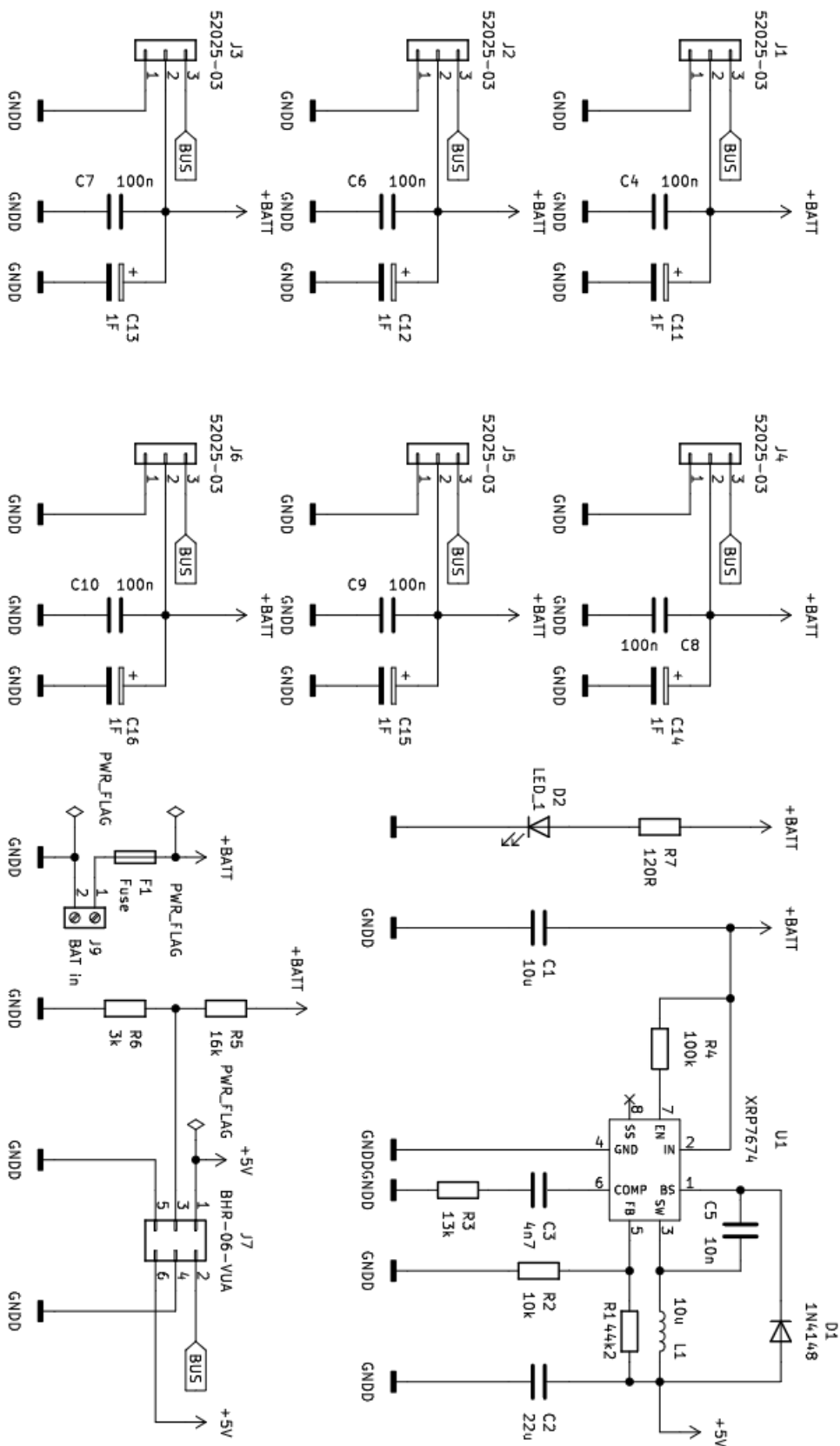
Celkový počet komponent 36

Tabulka A.3: Kusovník řídicí DPS

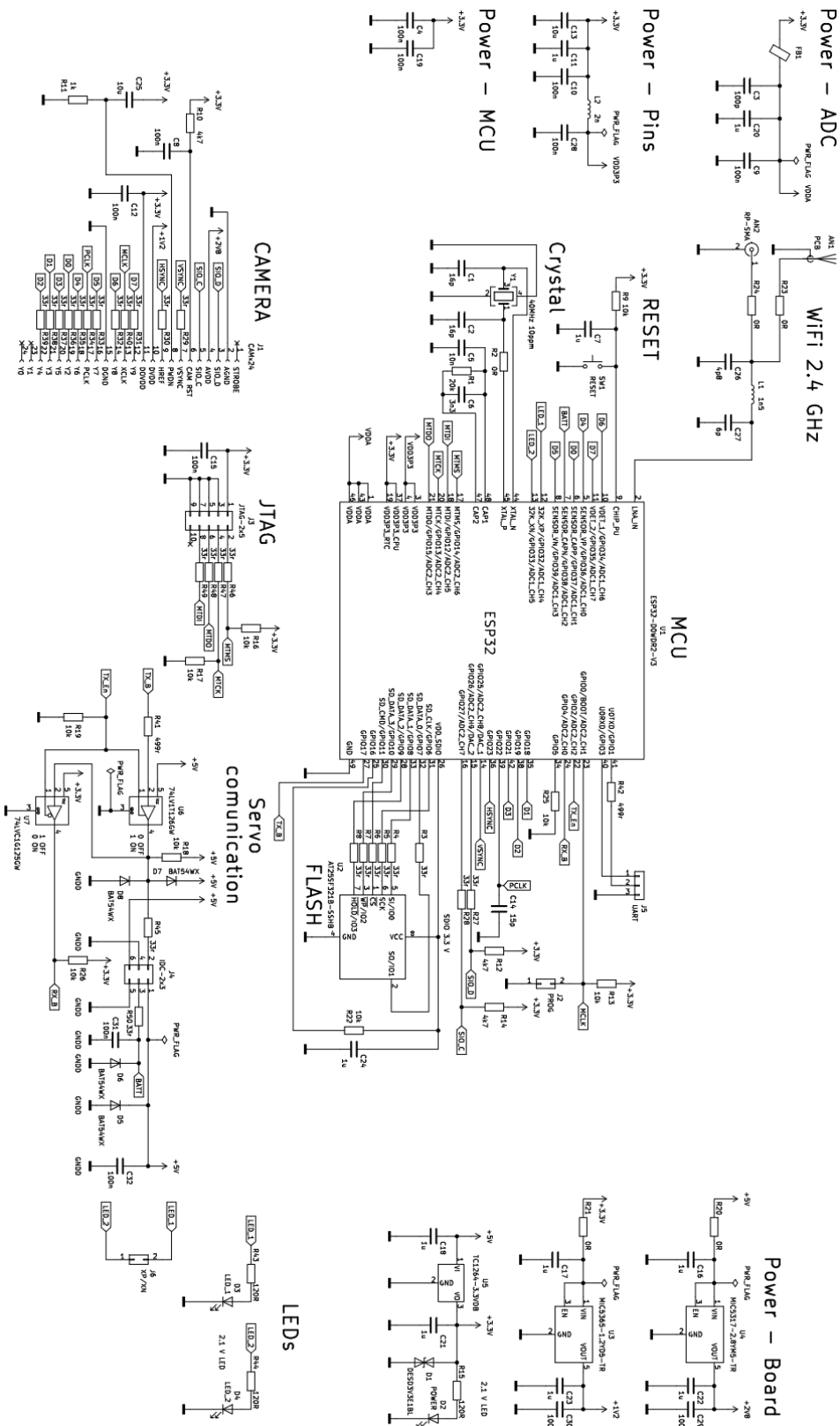
Označení	Počet	Hodnota	Značka	Footprint
AN1,	1	PCB	Antenna_Shield	Texas_SWRA117D_2.4GHz
AN2,	1	RP-SMA	Conn_Coaxial	SMA_Amphanol_Vertical
C1, C2,	2	16p	C	402
C3,	1	100p	C	402
C4, C9, C19, C28, C31, C32,	6	100n	C	402
C5,	1	10n	C	603
C6,	1	3n3	C	603
C7, C11, C16, C17, C18, C20, C21, C22, C23, C24,	10	1u	C	603
C8, C10, C12, C15, C29, C30,	6	100n	C	603
C13, C25,	2	10u	C	1206
C14,	1	15p	C	603
C26,	1	4p8	C	402
C27,	1	6p	C	402
D1,	1	DESD3V3E1BL	ESD9B3.3ST5G	402
D2,	1	POWER	LED	603
D3,	1	LED_1	LED	603
D4,	1	LED_2	LED	603
D5, D6, D7, D8,	4	BAT54WX	1N4148	D_SOD-523
FB1,	1	Ferite beam	FerriteBead	603
J1,	1	CAMx24	Conn_01x24	CviLux-CF2024
J2,	1	PROG	Conn_02x01	PinHeader_1x02_P2.54mm
J3,	1	JTAG-2x5	Conn_02x05	IDC_2x05_P2.54mm
J4,	1	IDC-2x3	Conn_02x03	IDC_2x03_P2.54mm
J5,	1	UART	Conn_01x03	PinHeader_1x03_P2.54mm
J6,	1	XP/XN	Conn_02x01	PinHeader_1x02_P2.54mm
L1,	1	1n5	L	402
L2,	1	2n	L	402
R1,	1	20k	R	603
R2, R23, R24,	3	0R	R	402
R3, R4, R5, R6, R7, R8, R45, R50,	8	33r	R	402
R9, R13, R16, R17, R18, R19, R22, R25, R26,	9	10k	R	603
R10, R12, R14,	3	4k7	R	603
R11,	1	1k	R	603
R15, R43, R44,	3	120R	R	603
R20, R21,	2	0R	R	603
R27, R28, R29, R30, R31, R32, R33, R34, R35, R36,	18	33r	R	603
R37, R38, R39, R40, R46, R47, R48, R49,				
R41, R42,	2	499r	R	603
SW1,	1	RESET	SW_Push	SW_SPST_TL3342
U1,	1	ESP32-D0WDR2-V3	ESP32	QFN-48-1EP_5x5mm
U2,	1	AT25SF321B-SSHB	AT25SF081-SSHD-X	SOIC-8
U3,	1	MIC5365-1.2YD5-TR	MIC5365-3.3YD5	SOT-23-5
U4,	1	MIC5317-2.8YM5-TR	MIC5365-3.3YD5	SOT-23-5
U5,	1	TC1264-3.3VDB	TC1262-33	SOT-223-3
U6,	1	74LV1T126GW	74LV1T126GW	SOT-353_SC-70-5
U7,	1	74LVC1G125GW	74LVC1G125GW	SOT-353_SC-70-5
Y1,	1	40MHz 10ppm	Crystal_GND24	2520-4Pin_2.5x2.0mm
Celkový počet komponent	110			



Obrázek A.1: Schéma ochrany DPS



Obrázek A.2: Schéma napájecí DPS



Obrázek A.3: Schéma řídicí DPS