



Vyšší odborná škola, Střední průmyslová škola a Obchodní akademie,
Čáslav, Přemysla Otakara II. 938

DLOUHODOBÁ MATURITNÍ PRÁCE

Vytvoření základní konstrukce robotického pavouka včetně dokumentace a vizualizace.

Čestné prohlášení

Místopřísežně prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v seznamu vloženém v práci.

Prohlašuji, že tištěná verze a elektronická verze práce jsou shodné.

V , dne

.....
podpis

Poděkování

Rád bych poděkoval vedoucímu práce Ing. Karlu Mlčochovi a oponentovi Ing. Matěji Reichmannovi za jejich odbornou pomoc a vedení. Děkuji také Mgr. Karlu Kvapilovi, Markétě Klvaňové a Ing. Davidu Němcovi MBA. Zvláštní poděkování patří mým rodičům za jejich obětavou pomoc a podnětné připomínky, které mi během projektu poskytovali. Nedílnou součástí jsou studenti a doktoři z ČVUT FEL, kteří mi pomohli s návrhem. Též děkuji své kamarádce Marianě Frecerové za kontrolu písemné části.



Anotace

Cílem této práce bylo navrhnout a realizovat PCB a SW pro robotického pavouka – Rapouk. V průběhu práce proběhl kompletní návrh, testování a výroba elektroniky. Nedílnou součástí dlouhodobého maturitního projektu bylo programování softwaru mikrokontroler ESP32 v jazyce Wiring. Tato práce je navazující na mou předchozí ročníkovou práci a představuje další etapu vývoje projektu Rapouk. Věnoval jsem se letování SMD součástek, testování, reverznímu inženýrství elektronických desek, komunikace pomocí HTTP protokolu a programování pomocí umělé inteligence – včetně jejich výhod i nevýhod.

Klíčová slova

Robotika, pavouk, PCB, SW, programování, ESP32, SMD, letování, Wi-Fi, HTML, HTPP, eloxování, servomotory

Annotation

The main focus of this coursework was to create a PCB and software of robotic spider – Rapouk. The project included the complete process of electronics design, testing, and manufacturing. Essential part of this work was programming od microcontroller ESP32 in Wiring. This work builds upon a previous year's project and represents the next stage in the development of Rapouk. It includes SMD soldering, testing, reverse engineering of PCBs, HTTP protocol and programming with AI – with a particular focus on its benefits as well as its limitations.

Keywords

Robotic spider, PCB, SW, programming, ESP32, SMD, soldering, Wi-Fi, HTML, HTPP, anodising, servomotors

Obsah

1	Úvod	9
2	PCB.....	10
2.1	CAD pro tvorbu PCB – EDA	11
2.2	Vývoj PCB.....	12
2.3	Rozložení součástek a design	13
2.4	Kontrola a DRC	14
2.5	Gerber files	15
3	Požadavky a výběr součástek	15
4	ESP32	15
5	Výkonová část	16
5.1	XL4015	16
5.2	ME3116	16
5.3	AOD423.....	17
6	Pohybová část	17
7	Nabíjecí část a akumulátor	17
7.1	Akumulátor LiPo 18650 2s.....	17
7.2	IP2326	18
8	Bezdrátové nabíjení – XKT čipy	19
8.1	Reverzní inženýrství	20
9	Komunikace.....	20
9.1	CH340C	20
9.2	FSUSB42	21
10	Senzor	22
10.1	TOF050C	22
10.2	IR LM393	22

11	Měření napětí	23
12	Konektory	24
13	Letování PCB.....	25
13.1	Pájecí stanice z žehličky	27
14	Prototypy elektroniky	28
14.1	První a druhá generace.....	28
14.2	Testovací generace.....	29
14.3	Třetí generace	30
14.3.1	Senzorová deska pro třetí generaci	30
14.4	Čtvrtá generace	31
14.5	Nabíjecí dok	32
14.6	Pátá generace	33
14.7	Finální šestá generace	34
15	Kontrola a měření teplot páté generace	35
15.1	Zadání měření	35
15.2	Podmínky měření	35
15.3	Použité pomůcky.....	36
15.4	Postup měření	36
15.5	Program s Collatzovým problémem	37
15.6	Schéma.....	37
15.6.1	Rozložení cest na PCB.....	39
15.7	Naměřené hodnoty	40
15.7.1	Bod 1.....	40
15.7.2	Bod 2.....	41
15.7.3	Bod 3.....	41
15.8	Vyhodnocení měření.....	43

15.9	Závěr měření	43
16	Software	44
16.1	Prvotní myšlenka provedení	44
16.2	Uživatelské prostředí	44
16.3	Tempomat a senzor.....	45
16.4	Pohyb	46
16.5	Mikrokrokování servomotoru.....	47
16.6	Chybové hlášky.....	47
16.7	Oprava chyb v kódu.....	47
16.7.1	Méně závažné chyby.....	48
16.7.2	Velice závažné chyby	48
16.8	Další funkce softwaru	51
17	Finální podoba	51
18	Závěr	54
19	Seznam použité literatury	55
20	Seznam grafických objektů.....	57
21	Seznam tabulek	59
22	Seznam rovnic.....	60
23	Seznam zkratek	61
24	Seznam příloh	63
A	Datové listy	63
B	Schémata a fotografie	63
C	Gerber files	63
D	Ostatní	63

1 Úvod

Tato práce navazuje na předchozí ročníkovou práci, jejímž cílem byla výroba CNC stroje a návrh základní konstrukce z hliníkové slitiny. Součástí tehdejší práce byla také tvorba modelů v CAD a CAM systémech a naprogramování prvních pohybů robotického pavouka. Projekt si nadále zachovává ideu univerzálního ovladače, otevřeného softwaru a dostupnosti pro všechna „chytrá“ zařízení.

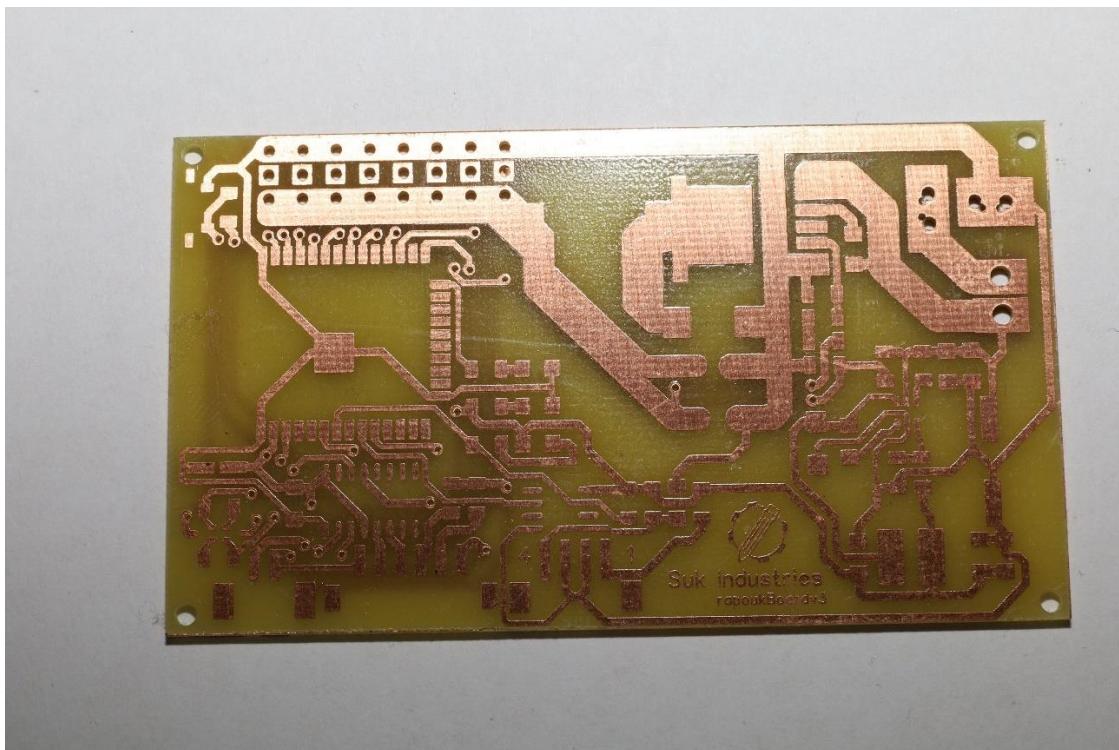
Tištěným spojům jsem se chtěl věnovat už dříve, však až teprve teď přišla správná chvíle začít s kompletně novým oborem techniky. Před tímto projektem jsem nikdy nenavrhoval PCB a neměl jsem ponětí, jak komplexní proces to bude. Ideální příležitost se naučit něco nového.

2 PCB

Printed circuit board (PCB) je deska plošných spojů (DPS), která slouží pro osazení elektronickými součástkami a je nedílnou součástí každé elektroniky.

Je složena ze dvou hlavních částí, sklolaminátového středu a měděné vrstvy. Dnes už PCB nejsou pouze zelené desky s oranžovými trasami, dá se totiž využít tzv. masky. Maska je překrývací vrstva, která chrání trasy před vnějšími vlivy a možným zkratováním. Do této masky lze zakomponovat design, popisky, loga a tak dále. Z nevhledné desky vznikne krásný DPS, se kterým se i lépe pracuje, protože jsou odhaleny pouze letovací plochy a trasy jsou „pod maskou“. Dnešní technologie umožňuje výrobu PCB s několika vrstvami mědi, což zjednoduší návrh. Svou desku jsem navrhoval s dvěma měděnými vrstvami – z horní a dolní strany. Hlavní motivací použití pouze dvouvrstvé desky byla cena a jednoduchost.

Plošné spoje lze frézovat, laserovat nebo i leptat. Leptání je nejvíce zastoupená metoda vytváření tištěných spojů. S dnešními technologiemi začíná být více populární laserová technologie, která je ale stále silně finančně náročná na provoz.



Obrázek 1 - Čistý PCB bez překrývací masky

2.1 CAD pro tvorbu PCB – EDA

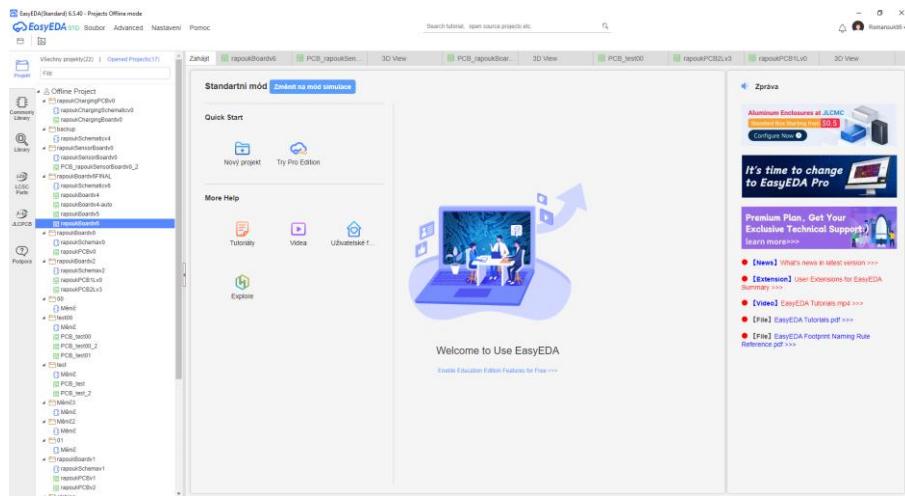
Ve strojním inženýrství se setkáme s programy, které se označují CAD – Computer-Aided Design. Pro návrh DPS se používá EDA – Electronic Design Automation. Na rozdíl od strojírenství existuje mnoho free, a i open-source, programů pro návrh PCB.

Úplně první EDA software jsem vyzkoušel EAGLE. Bohužel pro mě byl velice nepřehledný a matoucí. Knihovna se součástkami byla prakticky prázdná a návrhář si musel většinu součástek sám nakreslit. Jelikož jsem si EDA vybíral jako lajk, který tištěným spojům, a ani součástkám, moc nerozuměl, tak to bylo velice důležité kritérium. Naštěstí existuje velice jednoduchý EDA, a to EasyEDA.

V programu EasyEDA je knihovna součástek podporována komunitou vývojářů, kteří kreslí součástky, které se zobrazují v nabídce. Tato možnost sdílení součástek umožňuje okamžitý návrh PCB bez nutnosti kreslení normovaných footprintů, modelů nebo i ukázkových schémat. Z tohoto důvodu jsem si vybral čínský EasyEDA. Díky jeho jednoduchosti UI a knihovny součástek jsem pochopil nejnutnější základy tvorby schémat a návrhu tištěného spoje.

Dalším důležitým poznatkem je propojení s výrobou. Návrhář při tvorbě může využít vybrané součástky, které jsou u výrobce PCB na skladě. To umožňuje rychlé osazení desky ihned po výrobě. Tato služba je sice dražší, ale výrazně urychluje práci a snižuje chybovost.

V budoucnu bych chtěl následující projekty vyvíjet v KiCadu - je profesionálnější, avšak náročnější pro pochopení.



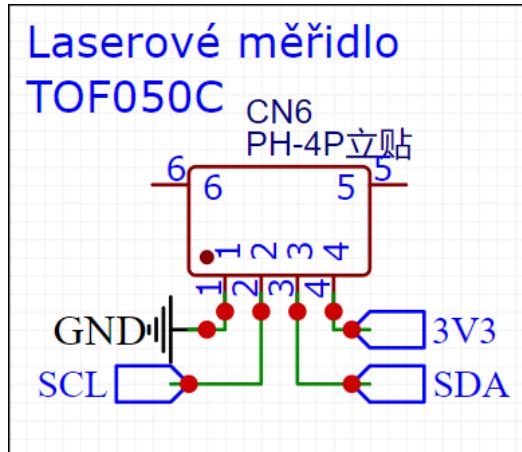
Obrázek 2 - UI EasyEDA

2.2 Vývoj PCB

Vývoj PCB vždy začíná výpočtem, popřípadě přečtením doporučených hodnot u výrobce.

Jako názorný příklad mám návrh tranzistoru.

Na YouTube kanále „GreatScott!“ jsem našel optimální hodnoty pro rezistory do zapojení tranzistoru. (1) Po návrhu součástek a hodnot je nutno vytvořit schéma, což je pouze zapojení součástek k sobě.



Obrázek 3 - Příklad schéma zapojení

Do návrhu bych rovnou zahrnul výšku vrstvy, ze kterého PCB bude vytvořen. Díky vyšší výšce následně může být tloušťka trasy menší. Tento údaj si musíme zjistit u výrobce (viz Kontrola a DRC). Výšky vrstev se udávají v imperiálních jednotkách oz. 1 oz odpovídá tloušťce 1 oz mědi rovnoměrně pokrývající plochu 1 čtvereční stopy. To je 1,4 mil, což je hmotnost na jednotku plochy vyjadřující tloušťku měděné vrstvy. Převedeno do metrického systému, tak 1 oz odpovídá 35 μm . Výhradně se používají násobky oz nebo 35, 70 nebo 105 μm . Samozřejmě je taky možno použít i poloviny oz, například 18 nebo 50 μm .

S těmito rozměry je nutno počítat z důvodu vznikajícího tepla při procházením proudu.

$$P = I^2 \times R$$

Rovnice 1 - Joulov zákon (2)

S druhou mocninou proudu se cesty více zahřívají. Plocha průřezu musí být dostatečná, aby nedošlo ke spálení cest. Existuje výpočet, kterým lze vypočítat optimální šířku, výšku, a dokonce i vzdálenost cesty. Tento výpočet je relativně složitý, takže jsem zvolil navrhovat šířku cesty pomocí tabulek.

Copper Thinkness, Line Width and Current

Thickness/1oz (35um)		Thickness/1.5oz (50um)		Thickness/2oz (70um)	
Current(A)	Line Width(mm)	Current(A)	Line Width(mm)	Current(A)	Line Width(mm)
4.5	2.5	5.1	2.5	6.0	2.5
4.0	2.0	4.3	2.5	5.1	2.0
3.2	1.5	3.5	1.5	4.2	1.5
2.7	1.2	3.0	1.2	3.6	1.2
2.3	1.0	2.6	1.0	3.2	1.0
2.0	0.8	2.4	0.8	2.8	0.8
1.6	0.6	1.9	0.6	2.3	0.6
1.4	0.5	1.7	0.5	2.0	0.5
1.1	0.4	1.4	0.4	1.7	0.4
0.8	0.3	1.1	0.3	1.3	0.3
0.6	0.2	0.7	0.2	0.9	0.2
0.2	0.2	0.5	0.2	0.7	0.2

Remark: 25°C, wire impedance: 0.0005xL/W (line length/line width)

Tabulka 1 - Hodnota proudu na šířku cesty (3)

2.3 Rozložení součástek a design

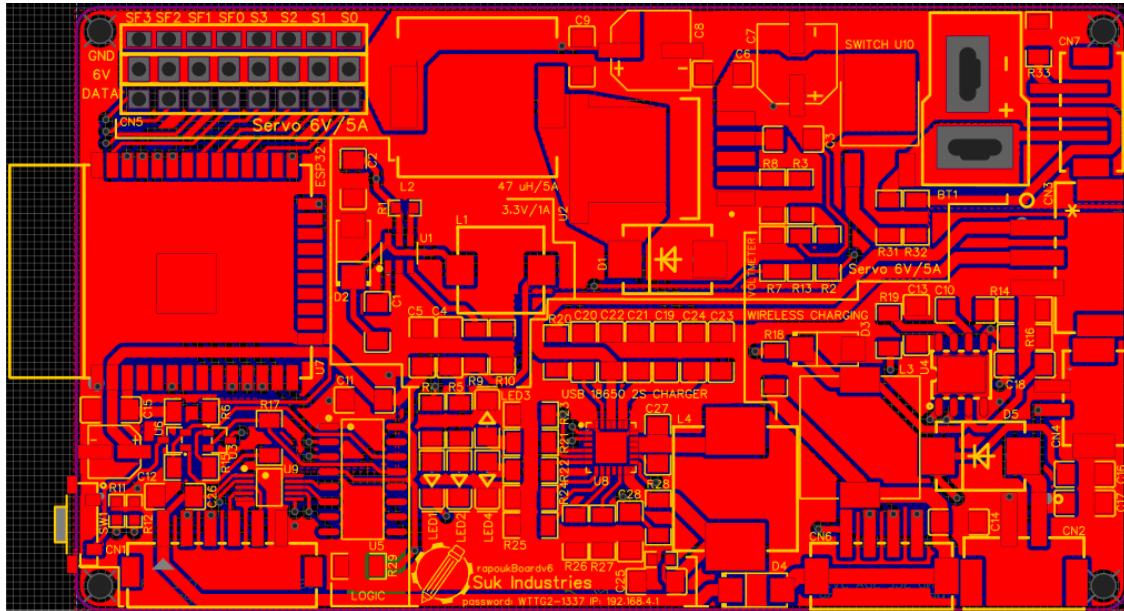
Schéma exportujeme do následující části – návrh PCB.

Při otevření nového dokumentu v návrhu PCB je nutno nastavit pravidla. Pravidly jsou miněny maximální mezery, tloušťky tras, průměr děr, prokovů a rozměry tištěné desky. Po nastavení pravidel určíme pozice jednotlivých součástek a navrhнемe trasy mezi nimi. Dle návrhu můžeme použít již zmíněné prokovy, které slouží k propojení vrstev mezi sebou.

Prokovy používáme v případě, kdy není možnost propojit součástky, aniž by došlo k propojení, tzv. křížením, tras mezi součástkami. Prokovy rozšiřují možnosti použití, letovací plochu a zmenšují velikost PC, protože se rozvrství do více vrstev.

Dále se navrhuje design PCB. Ten navrhujeme přes vnější masku, do které jsem přidal sériové číslo, verzi, název, logo a tak dále.

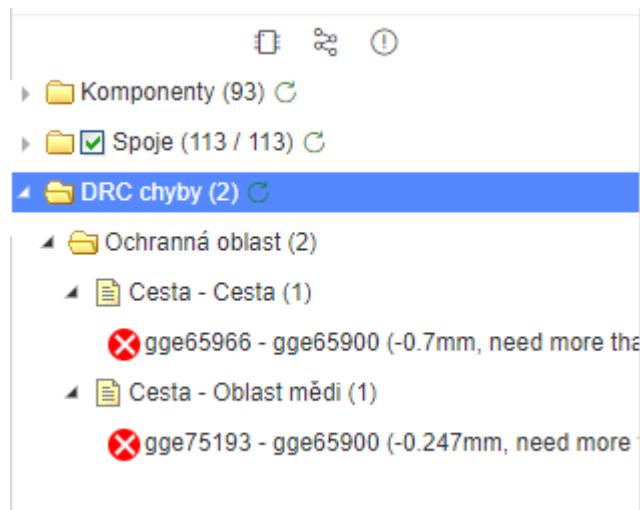
Jakmile je návrh v této fázi, pouze se doplní měděná vrstva do míst, kde chybí. Je to z důvodu nižšího šumu samotné desky a jednodušší výrobě.



Obrázek 4 - Návrh PCB

2.4 Kontrola a DRC

Po dokončení tohoto kroku je nutno desku zkontolovat, zda nedochází ke křížení tras. K tomu lze použít funkci Design Rule Checking (návrhová kontrola pravidel), která zkontovaluje, zda k tomuto křížení náhodou nedochází. Pokud dochází, DRC to zaznamená a nahlásí. Po úspěšném otestování PCB je možno desku odeslat do výroby.



Obrázek 5 – Ohlášení chyby DRC

2.5 Gerber files

Gerber files (viz Příloha O a Příloha P) jsou soubory okolo PCB, které jsou vyexportovány programem pro výrobce PCB. Obsahují všechny potřebné informace jako rozložení vrstev, trasy apod.

Většina výrobců nabízí i další možnosti výroby například výšky měděné vrstvy, lakování nebo barvy, které jsou potřeba vybrat. V mému případě většina možností byla za vysoký poplatek, a proto jsem zůstal u základní služby.

3 Požadavky a výběr součástek

Před jakýmkoliv návrhem je potřeba si urovnat výslednou podobu výrobku. Už od začátku mezi mými požadavky byla komunikace pomocí Wi-Fi, 6 voltové napájení pro osm servomotorů, LiPo akumulátor, senzor pro autonomní ovládání, možnost vlastního, nejlépe automatického, naprogramování a moderní USB-C připojení.

Obecně v tomto projektu bylo mým hlavním účelem použít velké množství moderních a normovaných součástek, které se používají i v jiných zařízeních a jsou modulární.

Tyto všechny požadavky mají relativně vysoké nároky na úplně první projekt s PCB. Z toho důvodu jsem vybral následující HW.

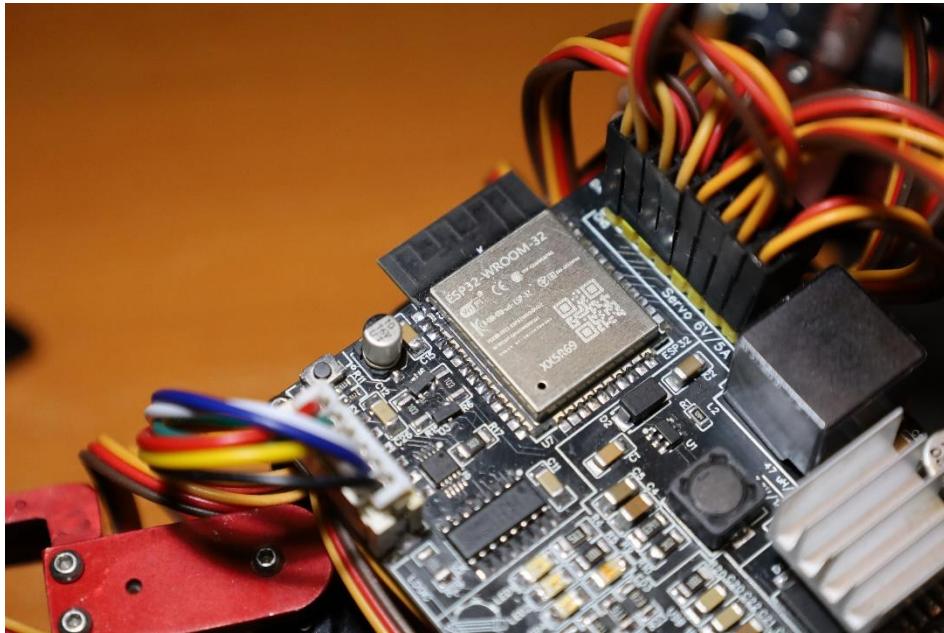
4 ESP32

ESP32 je špičkový mikrokontroler vyvinutý společností Espressif Systems, určený pro širokou škálu IoT aplikací. Díky dvoujádrovému procesoru Tensilica Xtensa LX6, který může být taktovaný až na 240 MHz s integrovanou Wi-Fi a Bluetooth modulem, jenž nabízí flexibilní platformu pro bezdrátovou komunikaci a náročnější výpočetní úlohy. Právě onen Wi-Fi modul je používán pro komunikaci mezi uživatelským zařízením a robotem. (4)

Mikrokontroler je vybaven širokou škálou periferií, včetně GPIO, ADC, PWM, I2C, což umožňuje snadné propojení s různými senzory a servomotory. Díky nízké spotřebě energie je ideální i pro aplikace napájené z baterií. (4)

V tomto projektu je ESP32 klíčovou součástí vlastního návrhu PCB, kde slouží jako centrální procesor, který řídí hlavní SW, zpracování dat a komunikaci s uživatelem.

Rozhodl jsem se využít starší varianty tohoto mikrokontroleru (viz Finální šestá generace) z důvodu levnější ceny, i přestože to výrobce již nedoporučuje (viz Příloha C).



Obrázek 6 - ESP32-WROOM-32

5 Výkonová část

5.1 XL4015

XL4025 je step-down měnič, který mění vyšší napětí na menší a funguje na bázi spínaného zdroje. Vstup je v rozmezí od 4 do 38 V s proudem 5 A. Jeho funkce je transformace napětí baterie na 6 V, což je ideální napětí pro servomotory. Jelikož LiPo akumulátor (viz Akumulátor LiPo 18650 2s) má veliký rozptyl napětí, XL4015 mi tento problém aktivně řeší stálým napětím 6 V na výstupu.

Více informaci v datovém listu (viz Příloha J).

5.2 ME3116

ME3116 je též step-down měnič a má stejnou úlohu jako XL4015. Jediný rozdíl je v napětí na výstupu, který je 3,3 V pro napájení ESP32. Jeho vstupní rozsah je od 4,75 do 40 V s proudem do 1 A. Tento měnič se mi osvědčil v modulové verzi, kterou jsem dříve používal.

Při zjišťování zapojení a aplikace reverzního inženýrství (viz Reverzní inženýrství) mi velice pomohla italská webová stránka s mnoha daty o měniči ME3116. (5)

Více informaci v datovém listu (viz Příloha I).

5.3 AOD423

Tranzistor AOD423 od firmy Alpha & Omega Industries je P-kanálový MOSFET slouží k přerušování obvodu. Hlavní výhodou tranzistorů je odstranění jiskření a nižší ztráty při spínání obvodu. Vybral jsem si P-kanál z důvodu možnosti vložení tranzistoru před celý obvod, přestože v porovnání od N-kanálového tranzistoru má vyšší odpor.

Chtěl jsem zkusit nové možnosti v každém směru. Přestože se jedná o absolutně základní znalost v elektrotechnice, tak jsem nevěděl, jak vlastně navrhnout a jaký použít tranzistor. Ten jsem nakonec použil místo standartního přímého vypínače.

Tímto musím poděkovat vyučujícím a studentům z ČVUT FEL, kteří mi pomocí platformy Discord vysvětlili všechny důležité pojmy pro výběr tranzistoru.

Více informaci v datovém listu (viz 0).

6 Pohybová část

Z předchozí ročníkové práci byly použity servomotory EMAX ES08MA II, které zde zůstávají. V minulém testování robota se servomotory osvědčily a plní mé požadavky i náročnější úlohy.

Více informaci v datovém listu (viz Příloha B).

7 Nabíjecí část a akumulátor

7.1 Akumulátor LiPo 18650 2s

Akumulátory LiPo 18650 jsou široce používané, díky napětí od 3,7 do 4,2 V. Používají se v akumulátorech v automobilech od firmy Tesla, koloběžkách Xiami anebo v téměř každé powerbance.

Jeho výhodou je vysoká kapacita a nízká hmotnost v poměru s jeho velikostí. Naopak nevýhodou je nutnost nabíjení každého článku jednotlivě, to se může řešit dvěma cestami. Buďto se akumulátorové články tzv. párují, kdy články mají stejné napětí a mohou se nabíjet všechny naráz v sériovém zapojení nebo je druhá cesta řešení a to ta, kterou využívá čip IP2326 (viz IP2326), jedná se o nabíjení a měření každého článku jednotlivě. Při nabíjení více nespárovaných článků akumulátoru by mohlo vzniknout přepětí článků a následné hoření. Další nevýhodou je veliký rozptyl napětí, takže je potřeba napětí měnit měničem na požadované napětí (viz XL4015).

Označení 2s označuje dva články, které jsou zapojeny v sérii. Osobně je v elektrotechnice využívám již několik let, takže z toho důvodu jsem si nakonec zvolil tento.

7.2 IP2326

Tento čip od firmy INJOINIC nabíjí LiPo 2 - 3s akumulátory. Jeho hlavní výhodou je nabíjení pomocí 5 V, takže možnost nabíjení pomocí USB. S podporou rychlonabíjecích protokolů je zde nutnost použít USB-C.

Přesný protokol k rychlonabíjení jsem nebyl schopen dohledat, ani v datovém listu se nezmíňují o přesném protokolu. Pouze je zde zmínka o QC (QuickCharge), ale myslím si, že je to pouze označení podpory rychlého nabíjení obecně. U USB-C se standardně používá PD protokol. QC nalezneme u USB-A, takže dle standardního protokolu označení QC, jakožto názvu protokolu, neodpovídá normě rychlonabíjecích protokolů. Zase na druhou stranu CC piny do čipu připojeny nejsou a datové piny USB konektoru jsou připojeny k čipu. Tento problém by mohl vyřešit testovací hardware, ale bohužel ten jsem v době testování neměl. Přestože by to tak být nemělo, tak vše naznačuje, že na USB-C je QC.

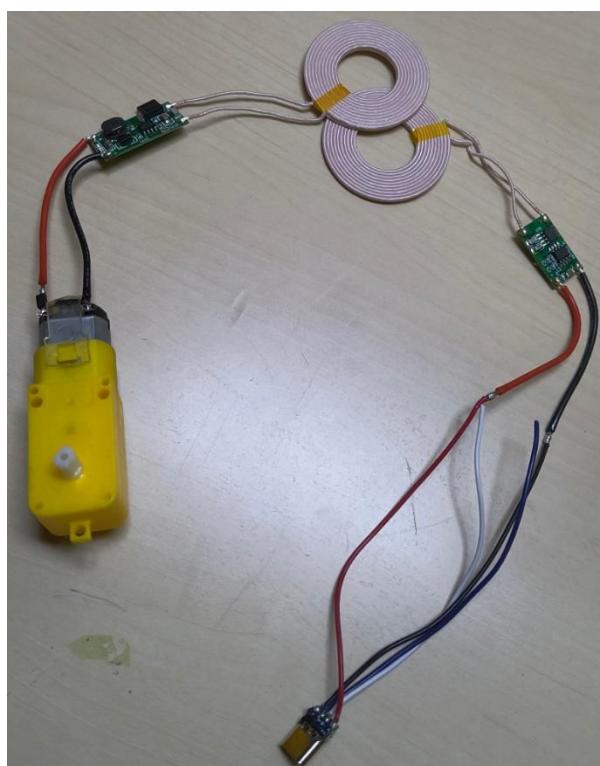
Další výhodou je kontrola teploty pomocí NTC teploměru (Negative Temperature Coefficient), který měří aktuální teplotu akumulátoru a mění charakteristiku nabíjení na bázi změrené teploty.

Nevýhoda LiPo akumulátorů je v nutnosti nabíjení jednotlivých článků (viz Akumulátor LiPo 18650 2s). Tento čip řeší tento problém s měřením napětí jednotlivých článků, což posouvá úroveň nabíjení výše. Stav nabitá baterie zde nelze sledovat, ale při nabíjení svítí LED, která přestane svítit po nabití akumulátoru.

Další nevýhodou je datový list, který je celý v čínštině, a relativně složitější zapojení. Jinak tento čip funguje více než dobře. (viz Příloha H).

8 Bezdrátové nabíjení – XKT čipy

Tyto čipy zde slouží k bezdrátovému nabíjení. XKT-3169 je u přijímače a kombinace čipů XKT-355 a XKT-412 je na vysílači. Našel jsem je v jednotlivých modulech na čínském tržišti AliExpress.com. Tyto moduly jsou jednoduché, malé elektronické destičky, které lze připojit do prototypových verzí. Jedná se o velice jednoduchou elektroniku, která není kompatibilní s ostatními bezdrátovými nabíječkami.



Obrázek 7 - Bezdrátové nabíjení XKT s USB-C a DC motorem

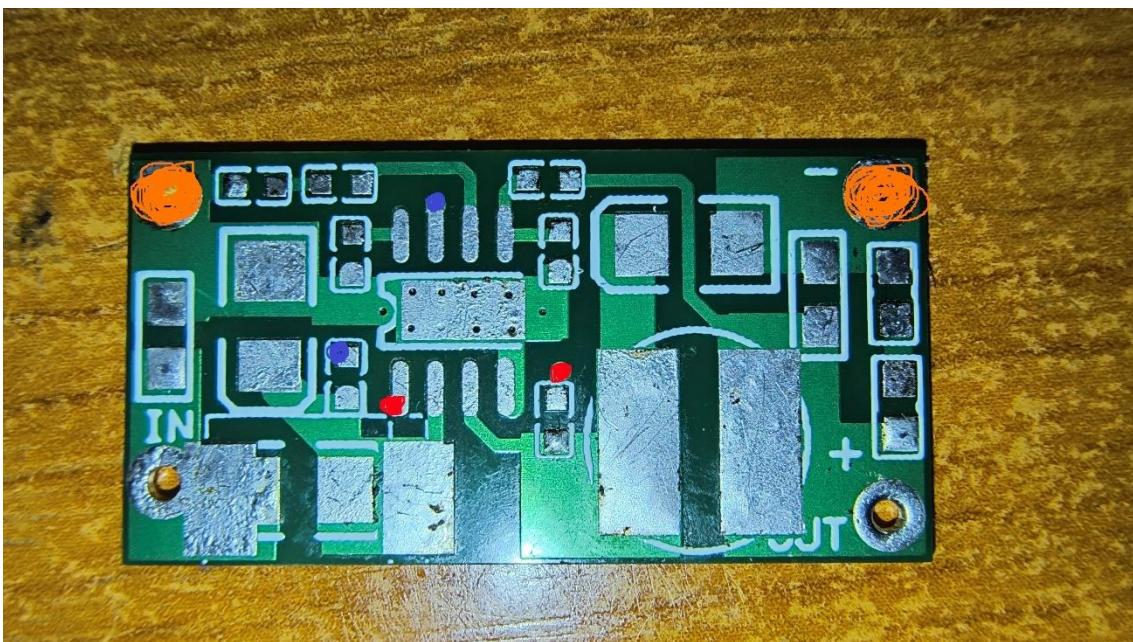
Po otestování samotného modulu jsem zjistil, že není kompatibilní s ostatními bezdrátovými nabíječkami. Takže není možné nabíjet robota pomocí jiné bezdrátové nabíječky, anebo pomocí nabíjecího doku (viz Nabíjecí dok) nabíjet jiné zařízení s jiným bezdrátovým modulem.

Schéma ani datový list jsem od této rodiny nebyl jsem schopen dohledat a je dost pravděpodobné, že ani neexistuje. Není znám ani oficiální výrobce, pouze toto označení součástek XKT, a proto mi nezbyla jiná možnost než využít cestu, dnes velice oblíbeného, reverzního inženýrství.

8.1 Reverzní inženýrství

Reverzní inženýrství je retrospektivní vývoj, to znamená, že je zde snaha zjistit, jak funguje již funkční verze. Nejvíce o tomto typu inženýrství slyšíme ve spojitosti s hackery, kteří se snaží prolomit ochranu softwaru.

Vém případě jsem postupně odebral všechny součástky, změřil a zjistil o jakou součástku se jedná. Následně jsem oskenoval zbylé PCB s cestami, které jsem následně překreslil do vlastní PCB. Získal jsem tím kompletně funkční a levné schéma bezdrátového nabíjení, které lze využít u takto malých projektů.



Obrázek 8 - PCB bezdrátového nabíjení – reverzní inženýrství

9 Komunikace

9.1 CH340C

Programátory CH340 jsou USB převaděči z USB na UART komunikaci. Standardní USB konektor obsahuje dva datové piny s označením D+ a D-, zatímco mikroprocesory používají UART (Universal Asynchronous Receiver Transmitter), který je známý jako RX/TX komunikace. Tato komunikace se využívá pro samotné naprogramování procesoru.

UART se může využívat v komunikaci mezi dalšími mikroprocesory, ale čím dál více se využívá I2C nebo SPI protokoly.

V rodině CH340 je na výběr mnoho upravených programátorů. V mé praxi jsem se nejvíce setkal se zastoupením CH340G, který potřebuje externí krystal pro časování. To CH340C nepotřeboval, což byl hlavní důvod mého zvolení.

Krom funkce převodu z USB na UART mají další funkce, kterých ESP32 využívá. Programátoři mají piny pro správné spuštění mikroprocesoru, které jsou EN a BOOT (GPIO 0). Při nahrání nového programu automaticky programátor přepne mikroprocesor do módu přeprogramování a následného restartování pomocí tranzistorů. Vše funguje automaticky a není potřeba mačkat různá tlačítka a jejich kombinace pro nahrání kódu.

Nevýhodou tohoto převodníku je nutnost instalace externích driverů do Windows. Existují i takové, které tuto výhodu nemají, ale jejich cena je na pěti, či více, násobku ceny CH340.

Více informaci v datovém listu (viz Příloha G).

9.2 FSUSB42

Z předchozích textů (viz IP2326 a CH340C) se zde vytvořil problém mezi dvěma vstupy jedním komunikačním výstupem. Přes USB-C musím komunikovat s ESP32 pro nahrání programu, ale i s IP2326 pro komunikaci o rychlonabíjení. Prvotní myšlenka byla spojit tyto kanály napřímo mezi sebou, ale z podstaty věci mi bylo jasné, že tento krok vytvoří více problémů než řešení. Přeci jenom komunikace s jedním výstupem a dvěma vstupy by byla problematická. Řešení tohoto problému jsem našel v USB multiplexeru (USB MUX) FSUSB42 od firmy Onsemi. Zjednodušeně, funguje jako relé. Standardně komunikuje s jedním vstupem, ale po přivedení napětí na snímací pin se komunikace přehodí na druhý vstup.

Hlavní komunikace je nastavena na komunikaci s IP2326. Při napájení mikrokontroleru ESP32 se napětí zvýší i na snímacím pinu, díky kterému dojde k přehození komunikace z IP2326 na ESP32.

Tato problematika se může řešit různou kombinací tranzistorů, avšak tento návrh mi přišel jako nevhodnější. Dostal jsem doporučení na tuto součástku na tzv. "Virtuální bastlírně".

Více informaci v datovém listu (viz Příloha D).

10 Senzor

10.1 TOF050C

Tento specifický senzor nahrazuje infračervený senzor. Jedná se o laserové měřidlo vzdálenosti, které můžeme naleznout v průmyslových laserových měřidlech na měření např vzdálenosti zdí na stavbě. Využívá svoji speciální knihovnu a s mikroprocesorem komunikuje přes I2C protokol. Maximální vzdálenost měření je 0,5 m.

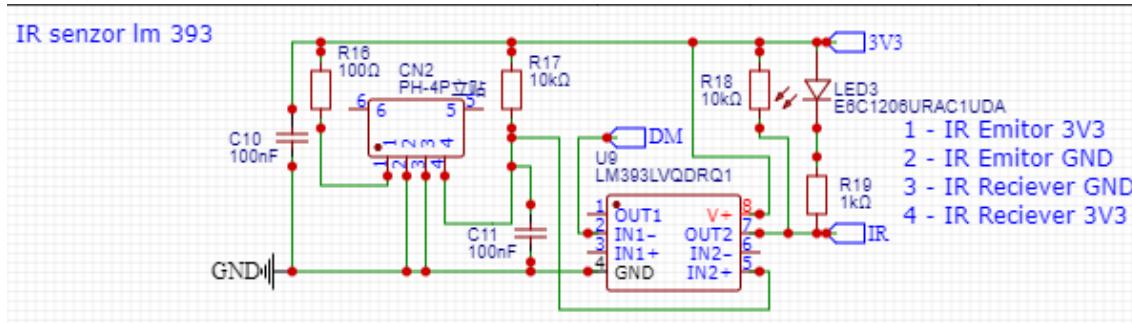


Obrázek 9 - TOF050C

10.2 IR LM393

Tento modul sloužil k detekci objektů na bázi odrazu infračerveného světla. Signál vytvořený IR světlem je zpracováván integrovaným obvodem LM393, který funguje jako komparátor a porovnává přicházející napětí od IR LED s přednastaveným napětím. Díky tomuto porovnávání vzniká kompletně digitální výstup, kterým lze indikovat, jestli je překážka před senzorem, nebo ne.

Tento princip je velice zastoupený ve světelných závorách, například na výrobních pásech. Světelné závory zabraňují lidské chybě, aby operátor nepřišel k úrazu při práci se zařízením.



Obrázek 10 - Schéma zapojení IR senzoru

11 Měření napětí

Napětí akumulátoru lze měřit mnoha způsoby. Vybral jsem si nejjednodušší možnost pomocí odporové děličky, která dělí napětí a snižuje ho na požadované napětí pro pin.

$$V_{out} = V_s \times \frac{R_2}{(R_1 + R_2)}$$

Rovnice 2 - Dělička napětí (6)

Součástí ESP32 je 12bitový převodník z analogového na digitální signál (ADC Analog-to-digital converter). Bohužel přesnost tohoto převodníku není dostatečná, takže výsledky mají vysoký rozptyl.

Možné řešení by bylo v přídavném čipu, který by samostatně měřil napětí a následně by digitálně komunikoval s ESP32. Pro tuto možnost jsem neměl prostor na PCB, a proto jsem musel vylepšit měření softwarem.

Pro výpočet napětí je nutno změřit hodnotu z pinu. Pro dosažení přesnějšího výsledku, je potřeba ji změřit vícekrát, odečíst nejvyšší a nejnižší hodnotu a vypočítat aritmetický průměr změřených hodnot.

$$Hodnota = \frac{Součet - (Nejnižší\ hodnota + Nejvyšší\ hodnota)}{Počet\ měření}$$

Rovnice 3 - Zpřesnění změřené hodnoty

Změřenou hodnotu následnou sadou rovnic přepočítáme na aktuální napětí.

$$V_{out} = Hodnota \times \frac{R_2}{(R_1 + R_2)} \times \frac{3,3}{4095}$$

Rovnice 4 - Změřené napětí (7)

Převod na procento kapacity akumulátoru funguje na porovnávání s laboratorními hodnotami, které odpovídají správnému napětí. (8) Hodnota na pinu se měří jednou za určitý časový interval.

12 Konektory

USB-C zde byla nutnost. V prosinci 2024 Evropská unie sjednotila nabíjecí konektor na USB-C (9), čehož jsem se chtěl udržet. S původní myšlenkou, že by se tento robot prodával a udržení aktuálních standardů by znamenalo velikou chybu. Dalším důvodem, proč jsem musel použít USB-C byla podpora rychlonabíjecích protokolů (viz IP2326).

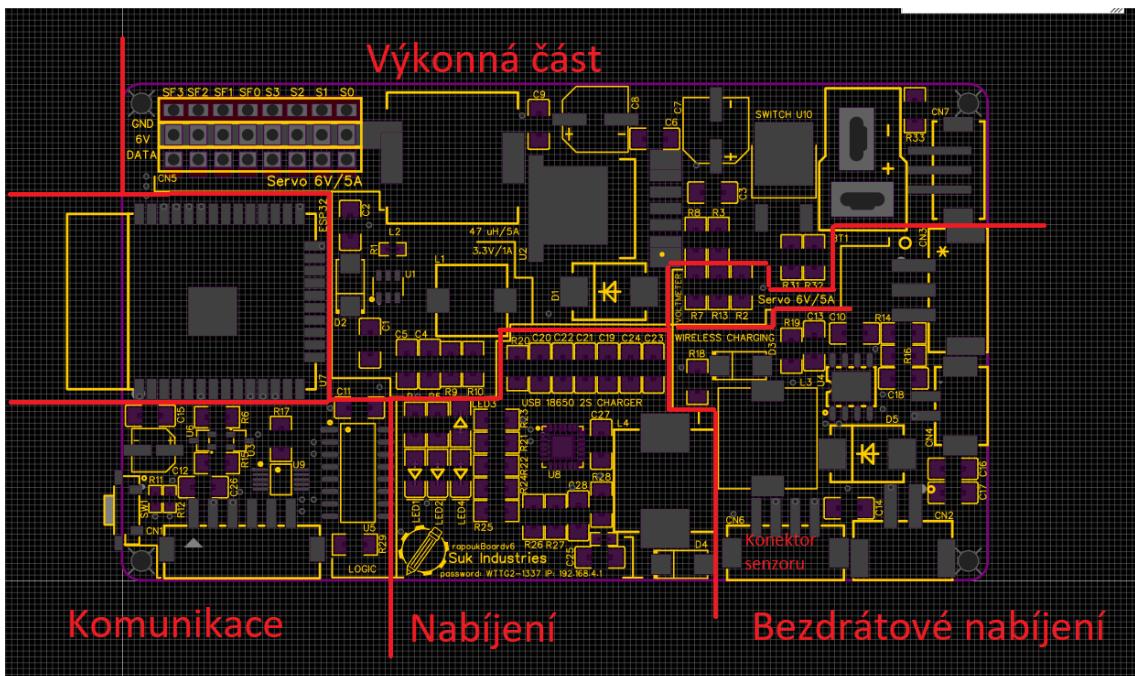
JST XH a JST PH jsou konektory pro připojení příslušenství. Jsou to velice oblíbené japonské konektory, které se využívají na těchto typech PCB. Menší konektory, PH, jsem použil pro datové toky (teplota a senzor) a pro výkonné (bezdrátové nabíjení a nabíjení akumulátoru) jsem použil větší XH konektory. Tím jsem docílil, že není možné zapojit konektory špatně. (10)

T-DEAN je proslulý konektor v oblasti RC, a hlavně Airsoftových replik zbraní, dokáže totiž přenést vysoký proud. Použil jsem tento konektor také z důvodu, že byl součástí akumulátoru od výrobce.



Obrázek 11 - JST PH a T-DEAN v porovnání s nehtem

DuPont je standartní konektor pro připojení servomotorů. Tyto konektory nalezneme především na vývojových deskách od firmy Arduino. Též se používá pro připojení servomotorů u RC modelů. Opět jsem zachoval modulárnost a standard. Pro vylepšení designu jsem v poslední verzi použil barevné rozdělení pro jednotlivé výstupy. Červená pro kladný náboj napětí, černý pro GND a žlutý pro data do servomotorů.



Obrázek 12 - Rozdělení PCB

13 Letování PCB

Před SMD (surface mounted device – povrchově namontována (naletovaná) zařízení (součástky)) se používala technologie THT (Through-Hole Technology – technologie skrz díru). Ta, jak název napovídá, se letovala přes díry v PCB, což umožňovalo velice jednoduché letování, kde stačilo použít standardní mikropájku s cínem a kalafunou, zatímco SMD se letuje pomocí speciálních past a horkého vzduchu.



Obrázek 13 - Horkovzdušná pistole

Před přiletováním se plochy PCB lehce obrousí, následně se překryjí vrstvou pájky. Na pájku se položí součástka, která je překryje kalafunou. Následně se aplikuje horký vzduch nad teplotou tavení pájky.

Pájku volíme dle několika faktorů. Použil jsem pájku Sn63Pb37 s teplotou tání při 183 °C. Z důvodu obsahu olova se jedná se o neekologickou variantu pájky. Na základně mých zkušeností jsou ale ty, s obsahem olova, nejlepší, protože nesou elektrický výkon optimálněji, než jiné.

Bohužel se občas pájení nepodaří (viz Obrázek 14).



Obrázek 14 - Spálená PCB

13.1 Pájecí stanice z žehličky

K pájení se používá horkovzdušná pistole, která fouká horký vzduch při správné teplotě. Avšak existuje ještě jedna možnost, jak letovat tištěné SMD spoje. Hojně se používá ohřívání povrchu pomocí topného tělesa – princip žehličky. Na teplém povrchu se celý PCB zahřeje na danou teplotu a pájka se změní v pevný kov. Tento postup je mnohem rychlejší, ale musí se zahřát celá deska se všemi součástkami.

Tuto techniku pájení jsem nikdy nepoužíval a nikdy jsem k ní neměl optimální hardware, a proto jsem si jej musel vyrobit.

Jako základ topného tělesa jsem použil zmíněnou starou žehličku. Do ohřívacího povrchu jsem nainstaloval teploměr a vše zapojil do laboratorního termostatu. Vše jsem následně vložil do elektroinstalační krabičky, do které jsem vyvrtal a vyrezal díry právě pro tento hardware. K topnému tělesu jsem přimontoval závitové tyče, ke kterým jsem dále přidělal zmíněnou krabičku.

Tímto jsem vyrobil ohřívací stůl pro jednodušší povrchové letování. Jelikož jsem s tímto typem letování začínal, tak ne každý spoj byl perfektní. Z toho důvodu jsem koupil speciální horkovzdušnou pistoli určenou pro toto letování. Horkým vzduchem je možno opravit chybná místa bez zahřívání celé PCB (viz Letování PCB).



Obrázek 15 - Pájecí stanice z žehličky

14 Prototypy elektroniky

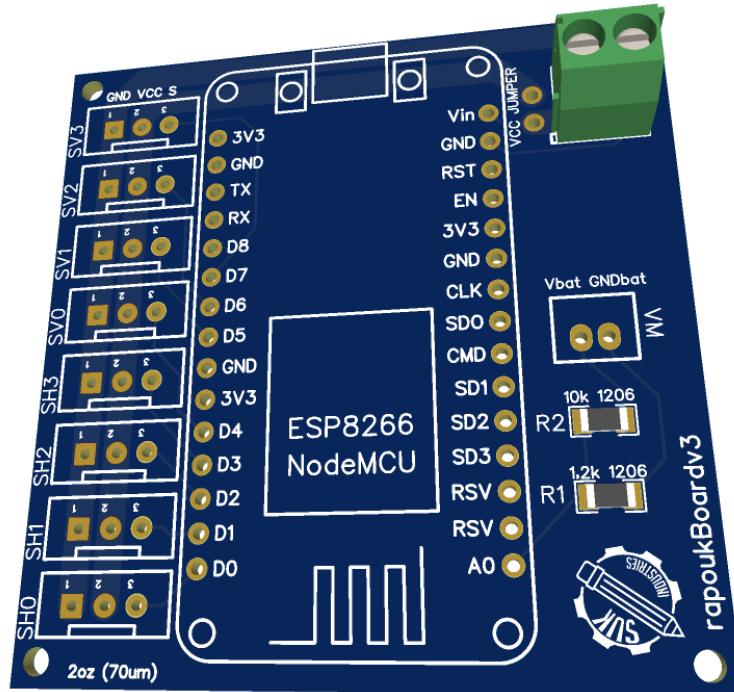
Vývoj elektroniky do projektu Rapouk si prošel několika generacemi pod technickými názvy s označením „rapoukBoardvX“, kdy „X“ je číslo příslušné generace. Sériové kódy byly vytiskeny až na finálním PCB. Existují i výjimky, které toto označení nesplňují. Před samotnou výrobou a testováním PCB jsem se musel naučit, jak se tato nová SMD technologie letuje.

14.1 První a druhá generace

Tyto generace, které nikdy nebyly vytvořeny, měly jednoduchou myšlenku – využití polotovarové vývojové desky s ještě starší generací procesorem ESP8266. Jednalo by se pouze o socket pro desku, servomotory a akumulátor s konektorem na měření napětí. U tohoto teoretického prototypu jsem počítal s 5 voltovým napájením.

Po pročtení datového listu k servomotorům (viz Příloha B) jsem zjistil, že mají vyšší výkon a kroutící moment při 6V napájení. Myslím si, že je namísto využít maximální napětí motorů pro maximální výkon.

Tento prototyp je pouze teoretický a nikdy nebyl vytvořen.



Obrázek 16 - Druhá generace

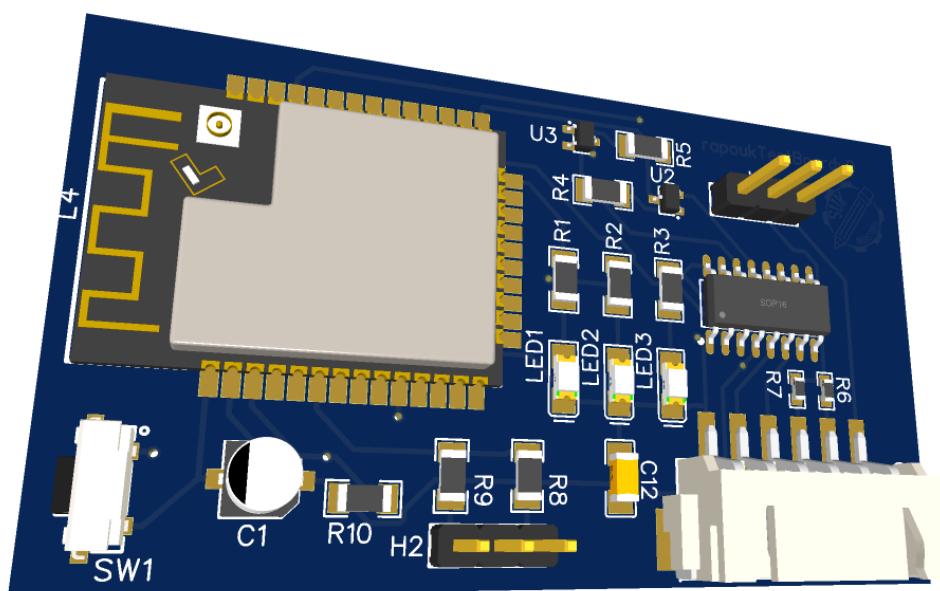
14.2 Testovací generace

Tato vývojová větev nemá moc společného s elektronikou do robota. Dokonce nesplňuje ani původní myšlenku. Též nemá správné rozměry ani díry pro uchycení. Účelem této desky bylo první seznámení se samotnou výrobou a letováním SMD PCB. Z toho důvodu jsem vytvořil jednostrannou desku s velkým prostorem mezi součástkami, abych zjistil způsob práce a postupu letování.

Tištěnou desku jsem označil „rapoukTestBoardv0“, která přesně popisuje příčinu vytvoření PCB. Vyzkoušel jsem si zapojení programátoru s automatickým naprogramování procesoru. Též jsem připojil USB-C, měření napětí a zabudoval tři, z toho dvě programovatelné, LED.

Při osazování PCB jsem narazil na různé velikosti polotovarových konektorů. Na neštěstí jsem nechal vyrobit desku pro větší konektor XH, přičemž jsem potřeboval menší PH. Z toho důvodu je USB-C konektor přiletován napřímo. Této chybě jsem se v následujících generacích vyhnul.

Oproti předchozímu prototypu tato deska fyzicky existuje.



Obrázek 17 - 3D render testovací PCB

14.3 Třetí generace

Po několika přechozích pokusech, o první tištěnou desku, přišla první funkční elektronika, která se už začala podobat mé původní vizi.

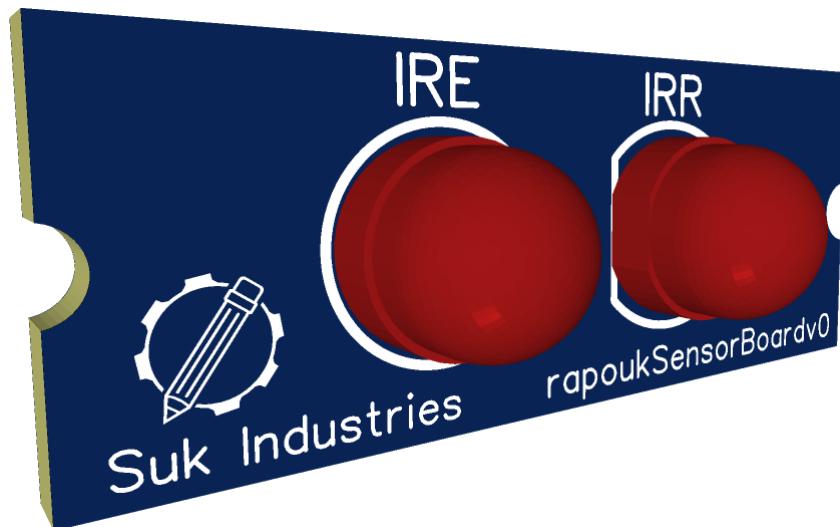
Obsahovala trasy pro 6 V a 3,3 V měnič pro napájení ESP32 a ostatních modulů. Samozřejmostí byla funkční automatické programování. Zajímavostí zde je LM393, což byla příprava pro autonomní IR senzor (viz Senzor). V této generaci jsem počítal s připojením nabíjení pro akumulátor.

Při návrhu této desky jsem narazil na polarizaci konektoru. Při kreslení PCB jsem otočil konektor, takže jsem měl zrcadlově otočeny všechny piny. To mělo za důsledek otočení polarity nabíjení, takže tyto cesty vedly do dalšího pinu, který nebyl připojen do další části desky. Naštěstí nedošlo k poškození a po otočení pinů elektronika byla plně funkční. Bohužel po výzkumu a snaze upravit funkční PCB jsem udělal chybu a zkratoval napájecí cesty. Nedošlo k poškození procesoru, ale pouze ke spálení cest mezi součátkami.

Třetí generace byla k vidění na fotografické soutěži na naší průmyslové škole, která vyhrála 2. místo (viz Příloha K).

14.3.1 Senzorová deska pro třetí generaci.

Součástí třetí generace byla senzorová deska. Sloužila jako uložení pro přijímač a vysílač IR signálu, který byl ovládán pomocí LM393 (viz IR LM393). Takto deska nebyla vyrobena, však její výroba by nebyla obtížná, protože se jednalo o velice malý jednostranný PCB. Rozměry byly 25×10 mm.



Obrázek 18 - Senzorová deska

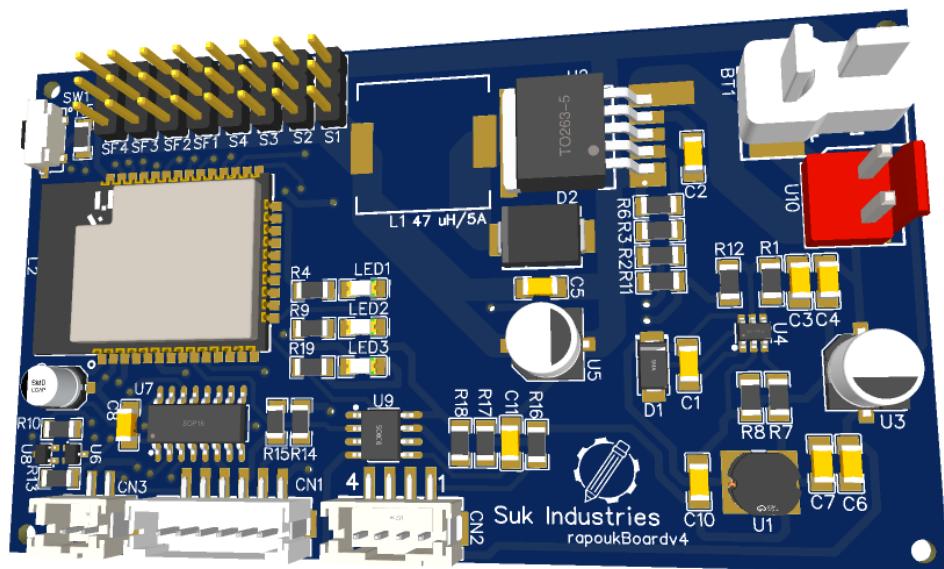
14.4 Čtvrtá generace

Čtvrtá generace už obsahovala vše, co jsem si představoval. Dokonce obsahovala i nabíjení akumulátoru. Původní myšlenka akumulátoru byla, že akumulátor bude v sobě obsahovat nabíjení. Když jsem akumulátor dostal do rukou, obdržel jsem k němu i USB nabíjení.

Při experimentování s nabíjením akumulátoru došlo k explozi tranzistoru. Neznám přímou příčinu, ale domnívám se, že hlavním důvodem byla nízká kvalita nabíjecí elektroniky. Též si myslím, že tato skutečnost je pouze pozitivní, protože jsem původní nabíjení nahradil lepším (viz IP2326).

Právě na této generaci došlo k výměně senzorů. Původní IR senzor jsem vyměnil za laserové měření vzdálenosti (viz TOF050C). Je jednodušší, na součástky levnější, a z mého pohledu i více profesionální. Též ušetřil mnoho místa na hlavní desce, které jsem mohl využít v následující generaci (viz Pátá generace).

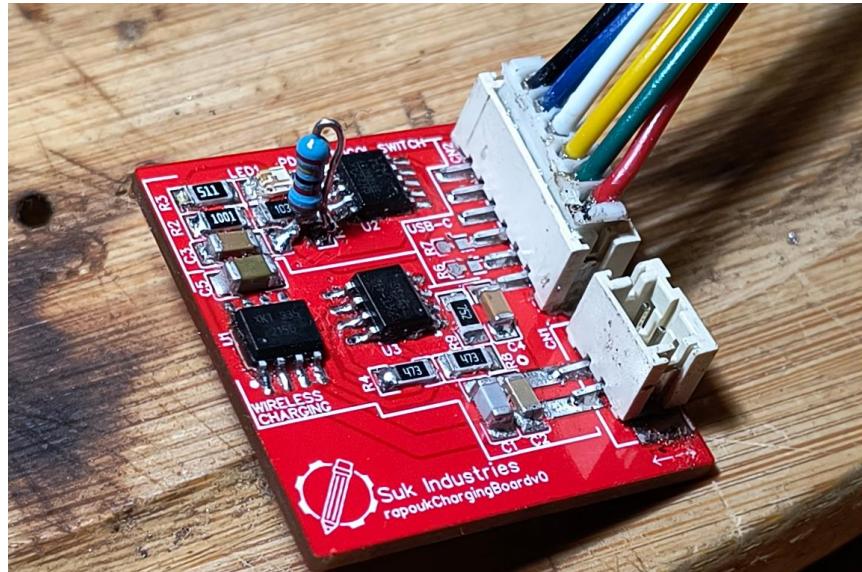
Další novinkou v této generaci byl tranzistorový spínač (viz AOD423).



Obrázek 19 - Čtvrtá generace

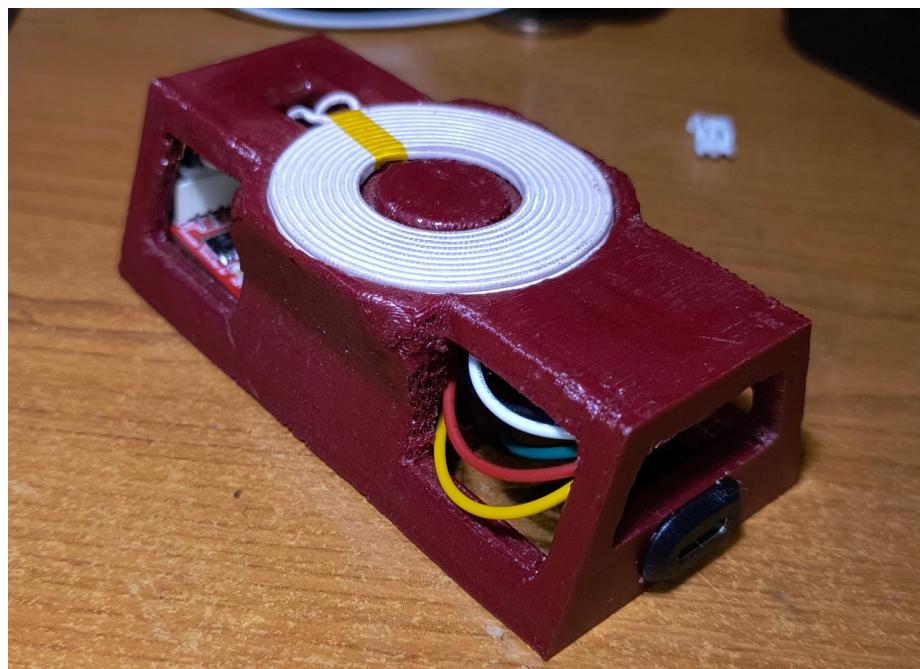
14.5 Nabíjecí dok

Společně s pátou generací jsem začal s vývojem elektroniky pro dokovací stanici „rapoukChargingBoardv0“.



Obrázek 20 - Elektronika nabíjecího doku

Jedná se o vysílač bezdrátového nabíjení a čip, který komunikuje s nabíječkou pro aktivaci rychlonabíjení až na 12 V. Výstupem je cívka, která vysílá střídavé elektromagnetické pole. To je následně zachyceno přijímací cívkou, která z něj indukuje napětí pro napájení elektroniky v pavoukově.



Obrázek 21 - Nabíjecí dok

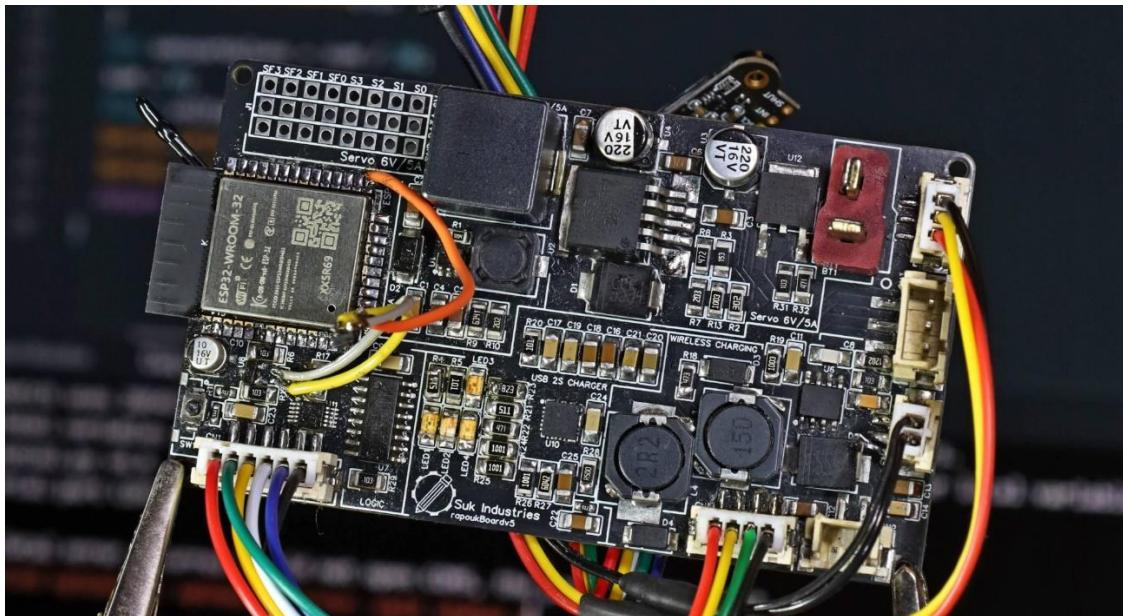
Opět jsem zde použil metodu reverzního inženýrství pro zjištění správného propojení elektronických součástek s čipy XKT (viz Reverzní inženýrství). Pro přepnutí nabíječky na vyšší napětí se používá čip CH224K, který komunikuje přes PD protokol (viz IP2326). Čip jsem aplikoval podle datového listu (viz Příloha E a Příloha F).

14.6 Pátá generace

PCB s názvem „rapoukBoardv5“ je předposlední generace elektroniky. Rozdíl oproti čtvrté generaci je v přidání alternativního, bezdrátového nabíjení a výměny otevřených tlumivek za stíněné. Taky jsem zjistil, že pro anténu procesoru není vhodné, aby byla položena na samotné desce. Je vhodné, aby anténa byla mimo desku k lepšímu šíření, popřípadě přijímaní dat. Bez této korekce bych zbytečně přišel o výkon antény, čemuž jsem se logicky chtěl vyhnout.

Přidané schéma je finální verzí, ze kterého následně vycházel design a rozložení součástek na PCB. Rozdělil jsem desku na několik částí, které jsem dále rozepsal na procesory, tranzistory a hlavní součástky obvodů. Všechny popsané části jsou součástí poslední, páté, generace elektroniky.

Tato verze byla testována v laboratoři pod termokamerou a výsledky měření se projevily na nadcházející generaci (viz Kontrola a měření teplot páté generace).



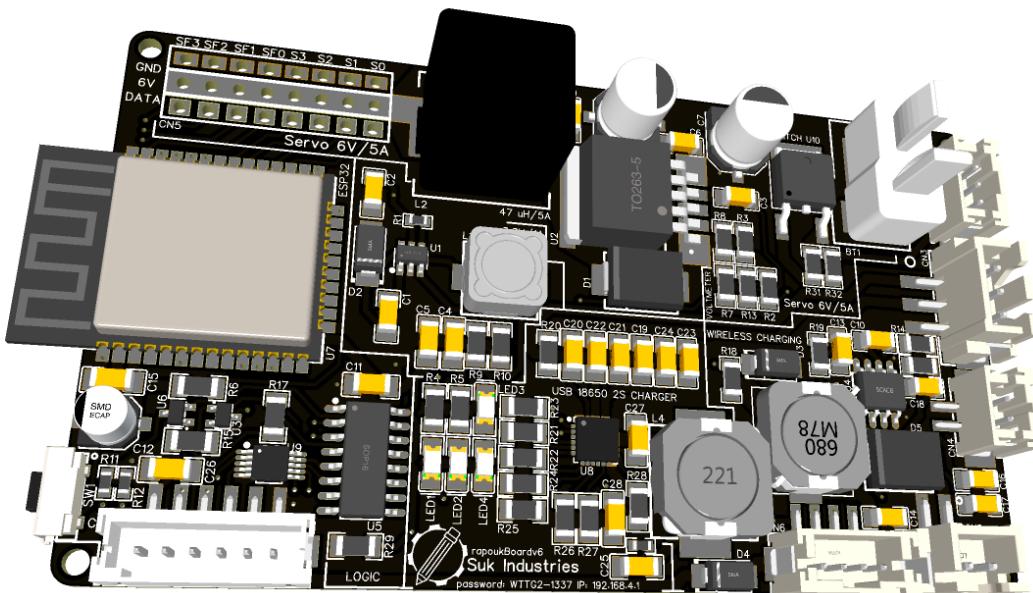
Obrázek 22 - Pátá generace

14.7 Finální šestá generace

Šestá, finální, generace se od páté liší pouze v drobnostech.

Byly přidány rezistory k tranzistorovému spínači a změna procesu z ESP-32S na ESP32-WROOM-32. Myslel jsem si, že rozdíl mezi těmito mikroprocesory je pouze v možnosti připojení antény. Bohužel těchto rozdílů bylo více. Pro mě z nejasného důvodu původní generaci s ESP-32S nebylo možné naprogramovat, ale po výměně procesorů vše fungovalo.

Dalším vylepšením bylo přidání kondenzátoru k programátoru a optimalizace tras pro I2C komunikaci se senzorem. Po testech s termokamerou jsem raději rozšířil hlavní napájecí cesty a více oddělil měniče od sebe.



Obrázek 23 – 3D render šesté generace

Při zapojování páté generace jsem zjistil špatné rozložení pinů na výstupu proti servomotorům. Servomotory mají rozložení konektoru – Data, VCC, GND, zatímco na PCB bylo GND, Data, VCC. U DuPont konektorů lze prohodit vodiče, však to by neodpovídalo všem ostatním servomotorům. To je ten důvod, proč jsem šestou verzi překreslil na správné výstupy.

V rámci této změny jsem nepoužil obyčejné černé DuPont konektory, ale barevně jsem je rozdělil. VCC na červenou, Data na žlutou a GND černou. Toto barevné rozložení upozorní na reálný výstup z PCB a též se jedná o hezký designový prvek.

Byly přidány přídavné odpory k vypínači, aby nedocházelo k jiskrám při zapojování bez vypínače. K prodloužení času v programovacím módu byl přidán kondenzátor mezi GND EN pin.

Přidány byly i popisky s IP adresou, heslem a dalšími potřebnými informacemi.

15 Kontrola a měření teplot páté generace

Před návrhem poslední generace je vždy potřeba otestovat prototyp. Zaměřil jsem se především na teploty v nad pracovních podmínkách elektroniky.

15.1 Zadání měření

- 1) Změřit pracovní teploty při použití 30 W (6 V/5 A) spotřebiče na dobu max. 5 minut:
 - a. měniče XL4015
 - b. tranzistoru AOD423
 - c. mikroprocesoru ESP32
 - d. spodní strany PCB
- 2) Otestovat
 - a. šířku cest PCB
 - b. ESP32 matematickým výpočtem
- 3) Změřit změněné napětí

15.2 Podmínky měření

	Teplota (°C)	Tlak (hPa)	Vlhkost (%)
Před	20,7	1023	38
Potom	21,0	1023	41

Tabulka 2 - Podmínky měření

15.3 Použité pomůcky

Přístroj	Výrobce	Model	Rozsah
Stejnosměrný laboratorní zdroj	Matrix	MPS-3010L-1	max. 30 V max. 10 A
Multimetr (proud)	Venlab	VM-600A	max. 20 A
Multimetr (napětí)	Voltcraft	AT-200	max. 1000 V
Kabely k měření	-	-	-
Spotřebič	Suk Industries	-	6 V 35 W
Termokamera	Testo	870-1	-20 až 280 °C
Laserový teploměr	Testo	810	-30 až 300 °C
Fotoaparát	Canon	EOS RP	-
Držáky		-	-
Stopky (telefon)	Motorola	Edge 40 neo	-
Software	Suk Industries	Collatzův problém	25 řádků 850 bajtů
Vývojové prostředí	Arduino	Arduino IDE 2.3.4	-
Počítač	-	-	-
Kabel pro připojení elektroniky	-	USB-A na USB-B	USB 2.0

Tabulka 3 - Použité pomůcky

15.4 Postup měření

- 1) Nahrát do mikroprocesoru matematický výpočet
- 2) Nastavit laboratorní zdroj na požadované napětí
- 3) Připojit potřebné zařízení k testovanému PCB dle schéma
- 4) Upevnit PCB do držáků, aby se nedotýkaly vodičů nebo pracovní desky
- 5) Připravit termokameru a fotoaparát
- 6) Zkontrolovat, zda nedochází ke zkratu na vstupu a výstupu
- 7) Zapnout laboratorní zdroj a stopky
- 8) Kontrolovat teplotu pomocí termokamery a teploměru
- 9) Zapsat hodnoty
- 10) Vypnout laboratorní zdroj
- 11) Vyčkat, dokud teplota nebude okolo 40 °C
- 12) Měření ještě jednou zopakovat
- 13) Vyhodnotit a uzavřít

15.5 Program s Collatzovým problémem

```
#define LED 2 //definování hodnoty

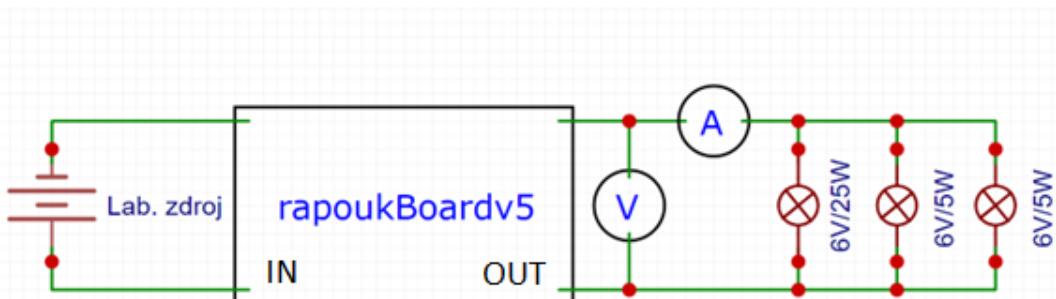
int startingNumber = 2; //počáteční číslo
int number = startingNumber;

void setup() {
    Serial.begin(9600); //zahájení sériové komunikace po COM
    pinMode(LED, OUTPUT); //nastavení pinu jako výstup
}

void loop() {
    if (number == 1) { //pokud je číslo rovno "1", zvýši číslo
        startingNumber++;
        number = startingNumber;
        Serial.println("Restartuji na nové číslo: " + String(number));
    } else if (number % 2 == 0) { //pokud je dělitelné dvěma
        number = number / 2; //vydělení dvěma
        digitalWrite(LED, HIGH); //zapnutí LED
        Serial.println(number); //Odeslání hodnoty do konzole
    } else { //v jiném případě
        number = (3 * number) + 1; //vynásob třemi a přičti jedna
        digitalWrite(LED, LOW); //vypnutí LED
        Serial.println(number);
    }
}
```

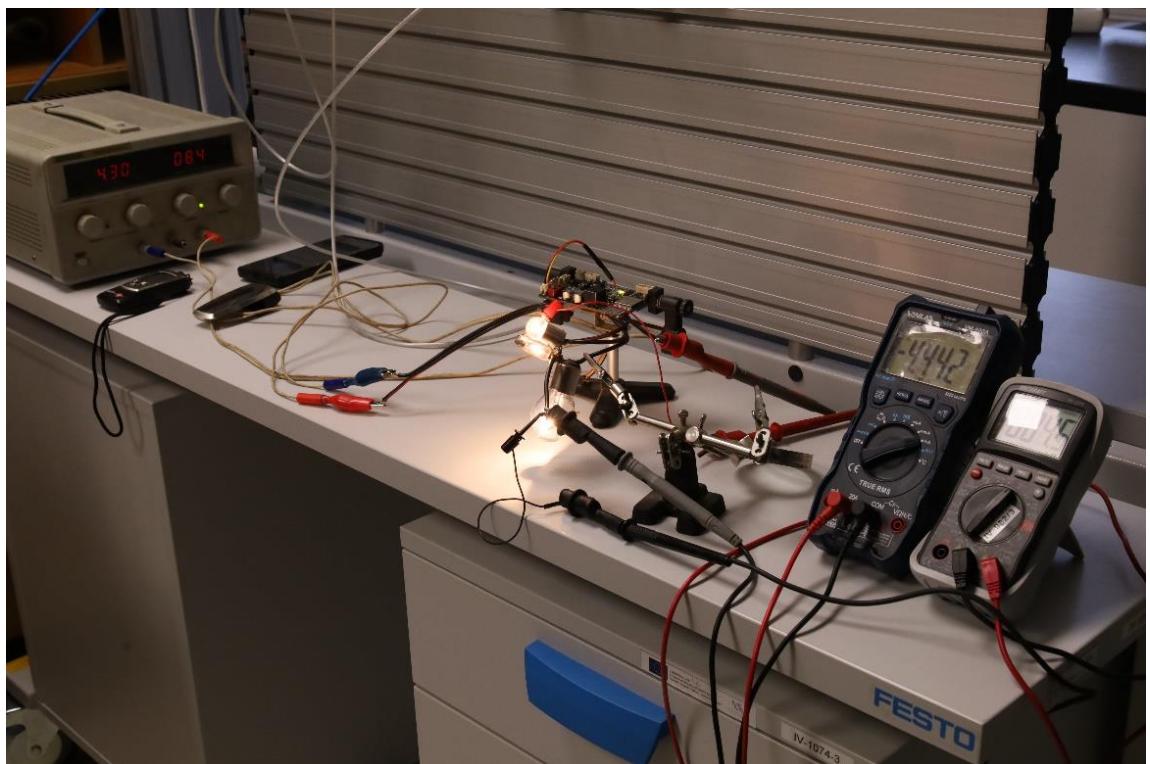
Collatzův problém je nevyřešená matematická úloha, která začíná libovolným kladným číslem. Pokud je číslo sudé, dělí se dvěma, pokud liché, vynásobí se třemi a přičte se jedna. Tento postup se opakuje a otázkou je, zda sekvence vždy skončí číslem 1. Tento problém jsem vybral, protože jeho naprogramování je jednoduché a zároveň dostatečně zatíží mikroprocesor (viz Příloha Q).

15.6 Schéma



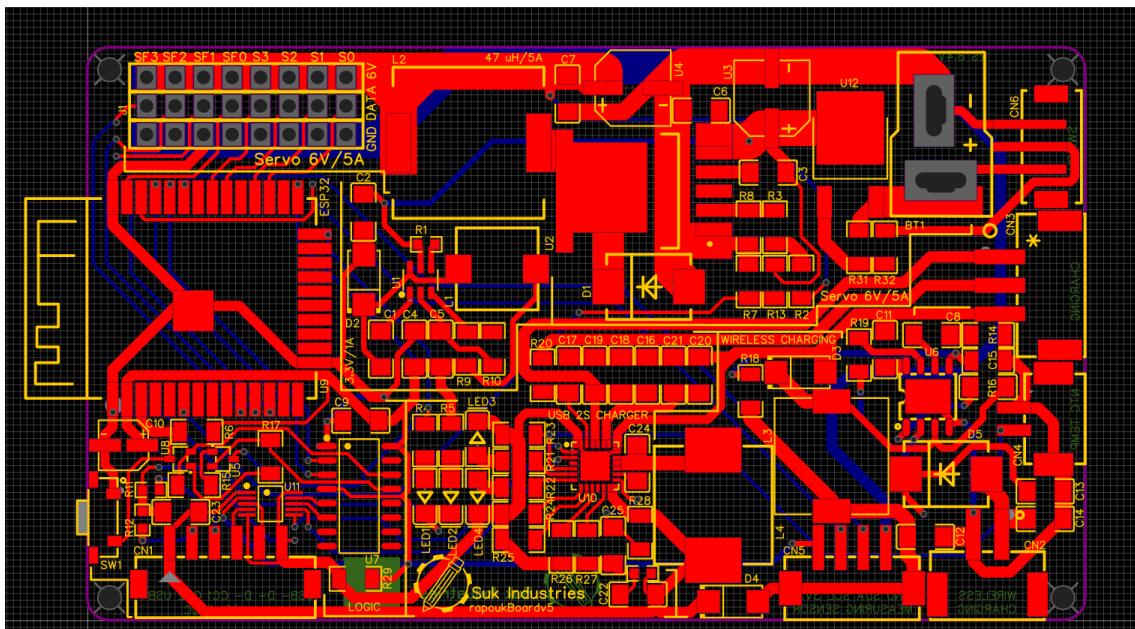
Obrázek 24 - Schéma zapojení při měření

Spotřebič jsou tři žárovky o výkonu 25 W, 5 W a 5 W zapojených paralelně pro zachování stejného napětí 6 V. Výkon spotřebiče se liší od výkonu v zadání z důvodu jiného teoretického a reálného výkonu. Žárovky byly vyjmuty z motorky Jawa 20/21, takže se nejedná o laboratorní a přesný spotřebič. Přestože teoretický proud je 6 A, reálný byl 4,45 A (což je pokles výkonu o 25,83 %). Proud nebyl omezen měřenou elektronikou, protože po připojení samotného spotřebiče na laboratorní zdroj byl proud lehce vyšší. To bylo zapříčiněno spotřebou samotného měniče na PCB.

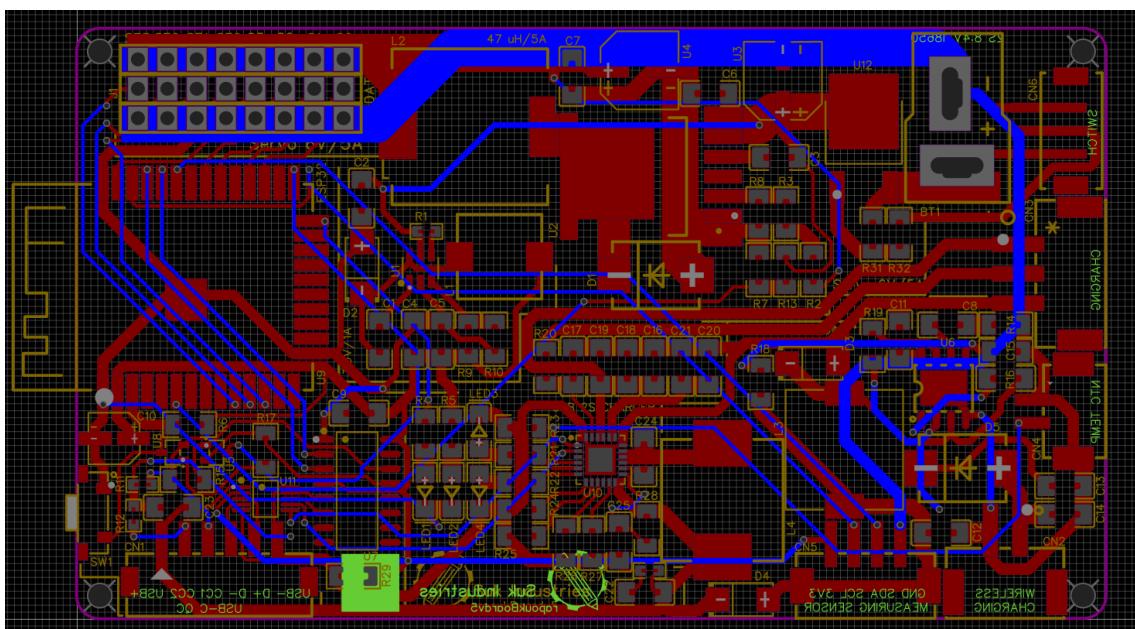


Obrázek 25 - Měření

15.6.1 Rozložení cest na PCB



Obrázek 26 - Horní strana PCB¹



Obrázek 27 - Spodní strana PCB

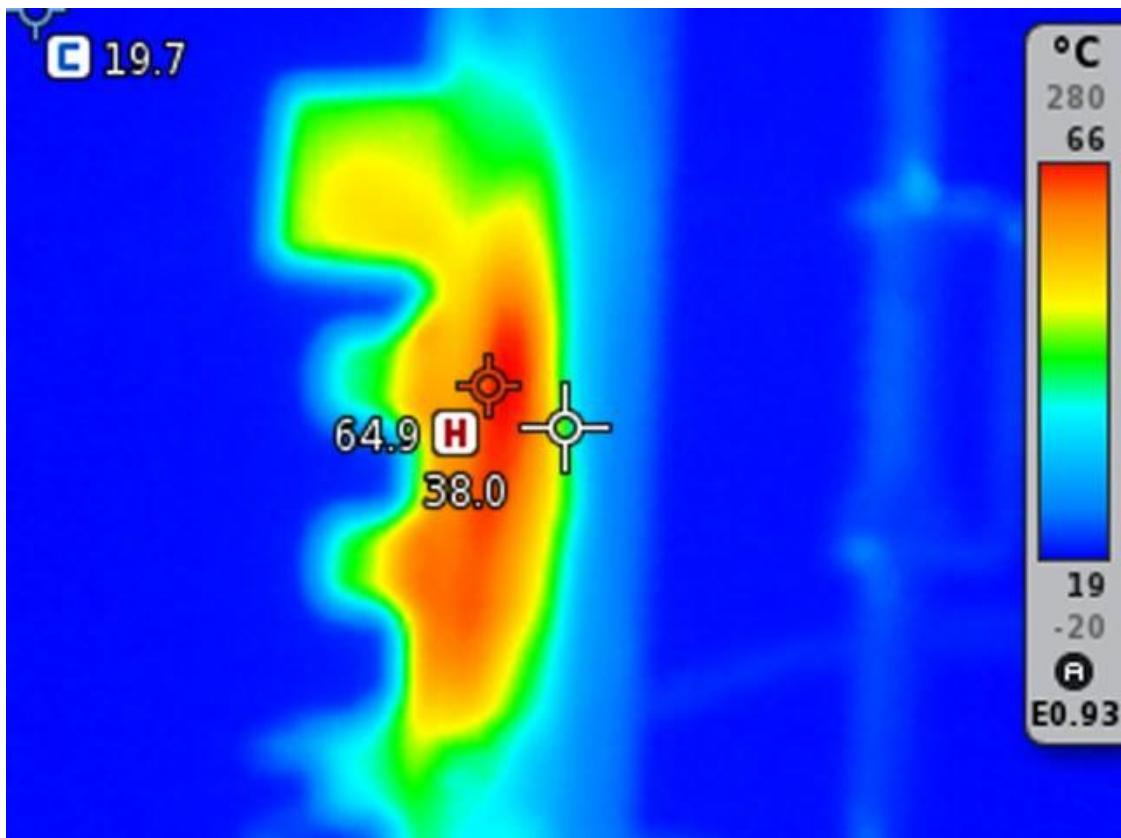
¹ Znázornění je upraveno pro lepší čitelnost. Došlo o odstranění ostrovních cest, které slouží hlavně k zmírnění rušení signálů. Měřená deska tyto ostrovní cesty obsahovala. Dalším rozdílem jsou rezistor R33 a kondenzátor C28. Tyto součástky na měřené PCB nebyly. Platí i pro Obrázek 27.

15.7 Naměřené hodnoty

15.7.1 Bod 1

Číslo měření (°C)	XL4015	AOD423	ESP32	PCB
1	115	71	30	50
2	120	82	31	83

Tabulka 4 - Měření teplot



Obrázek 28 - Teplota spodní strany PCB z termokamery

15.7.2 Bod 2

Cesty na PCB se nezahřívaly, nepoškozovaly se a byly správně navrženy. Pouze docházelo k ohřívání od ostatních součástek.

Chování a funkce PCB byly normální i ve vyšších teplotách (viz Bod 1). Nedocházelo k poškození při funkci.

Teplota mikroprocesoru ESP32 nepřevýšila 32 °C (viz Bod 1).



Obrázek 29 - Teplota spodní strany PCB pomocí teploměru

15.7.3 Bod 3

Číslo měření (V)	Změněné napětí
1	4,6
2	6

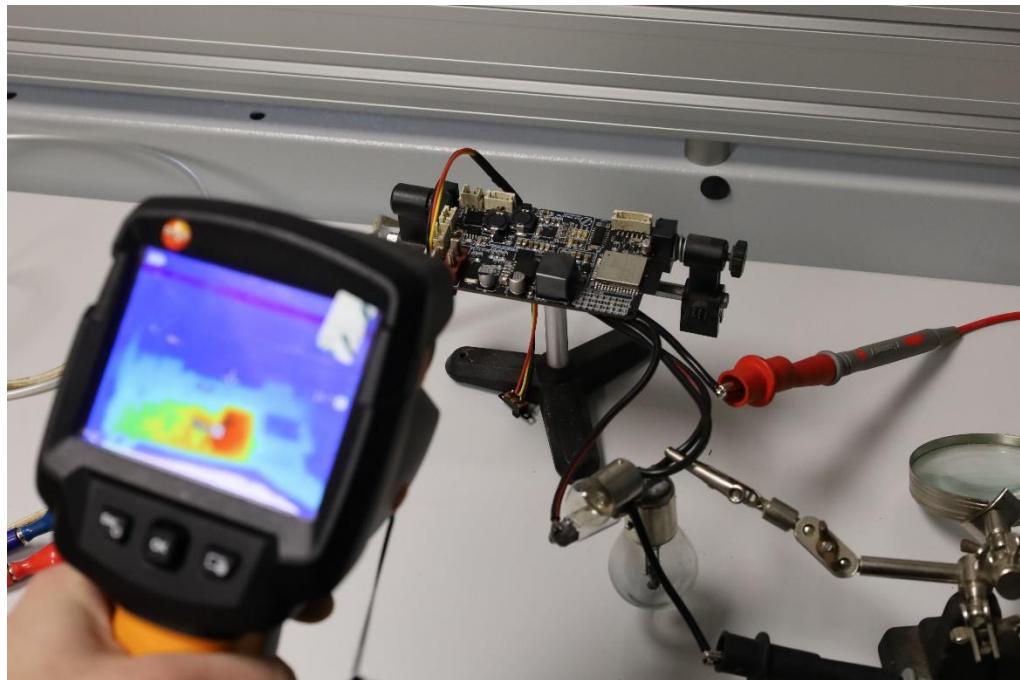
Tabulka 5 - Výsledek měření napětí

První měření nelze vzít v úvahu z důvodu špatně zvoleného měřícího bodu, ve kterém docházelo ke snížení napětí. Druhé měření je správné.



Obrázek 30 – Zapojení

15.7.3.1 Fotografie s termokamerou



Obrázek 31 - Měření termokamerou



Obrázek 32 - Fotografie z termokamery

15.8 Vyhodnocení měření

Teploty při měření byly velmi vysoké. Podle datového listu měniče XL4015 je provozní teplotní rozsah -40 °C až 125 °C.

Z toho důvodu i po dosažení hodnot 120 °C měnič fungoval stejně, ale je zde problém s okolními součástkami. Proto by bylo vhodné měnič na 3,3V posunout níž na PCB. Bohužel data k PCB nejsou k dispozici, takže maximální teploty PCB nemůžeme srovnat.

Dalším doporučením je zvětšit plochu pod XL4015 pro lepší rozprostření tepla a je nutné přidat minimálně pasivní chladič na měnič pro snížení teploty.

V reálném použití PCB proud nebude tak vysoký. Deska byla testována při 4,4A, přičemž reálný proud se bude pohybovat okolo 3,2A (8 servomotorů s proudem 0,4A dle výrobce), to znamená pokles výkonu o 27,27 %. Dodržením všech doporučení by teploty neměly dosahovat teplotních hodnot jako při měření.

Tranzistorový vypínač AOD423 dosahoval 82 °C, přičemž pracovní teplota, dle výrobce, je až do 175 °C. Při úpravě další generace bych zvětšil plochu cest mezi tranzistorem a měničem.

Teplota spodní strany PCB byla okolo 80 °C, což bych považoval za standardní a normální. Bohužel nejsou žádná referenční data k porovnání k dispozici. ESP32 si udržovalo stabilní teplotu, která nepřesáhla 35 °C a nepotřebuje žádnou úpravu.

Výchozí napětí bylo stále při 6 V.

15.9 Závěr měření

Z počátku jsem byl zaskočen teplotami, protože jsem je neočekával. V příštím měření bych se více zaměřil na práci s termokamerou, protože některé snímky byly matoucí, nezaostřené a zavádějící. Každopádně řešení ve vyhodnocení dokážou vyřešit i tyto zavádějící informace.

Měření bylo jiné, zábavné a relativně rychlé.

16 Software

Hardware projektu Rapouk by nebyl plně funkční bez SW. ESP32 lze programovat v různých prostředí, ale z důvodu psaní prvních pohybů na Arduino UNO jsem zůstal v Arduino IDE. Toto IDE používá programovací jazyk Wiring, což je zjednodušené C++.

16.1 Prvotní myšlenka provedení

Na ESP32 se uživatel bude připojovat pomocí zmíněné Wi-Fi. Následně pomocí webového prohlížeče a pevné IP adresy si zobrazí webovou stránku, která bude zastupovat kompletní uživatelské rozhraní.

Webová stránka bude rozdělena na levou, středovou a pravou část. Levá a pravá část webové stránky slouží k ovládání pohybů. Na levé jsou směrové pohyby jako pohyb dopředu, dozadu apod. Na pravé jsou speciální animace jako mávání, nebo klikování, je zde však i možnost doprogramování si vlastních animací pod tlačítka Custom0 až Custom2. Středová část slouží jako informativní s IP adresou a procentem kapacity baterie (viz Měření napětí). Též je zde posuvný prvek, který ovládá časový rozdíl mezi jednotlivými mikrokroky servomotoru.

Po každém odeslaném příkazu webová stránka počká, dokud animace neskončí. Po dokončení se odešle odpověď s dokončením příkazu. Jakmile se potvrzovací zpráva odešle, automaticky webová stránka odešle dokončovací příkaz, který vrátí pavouka do původní pozice. Přidal jsem tento příkaz z důvodu, aby se pavouk vrátil do pozice, ve které začal, čímž se zjednoduší následné ovládání.

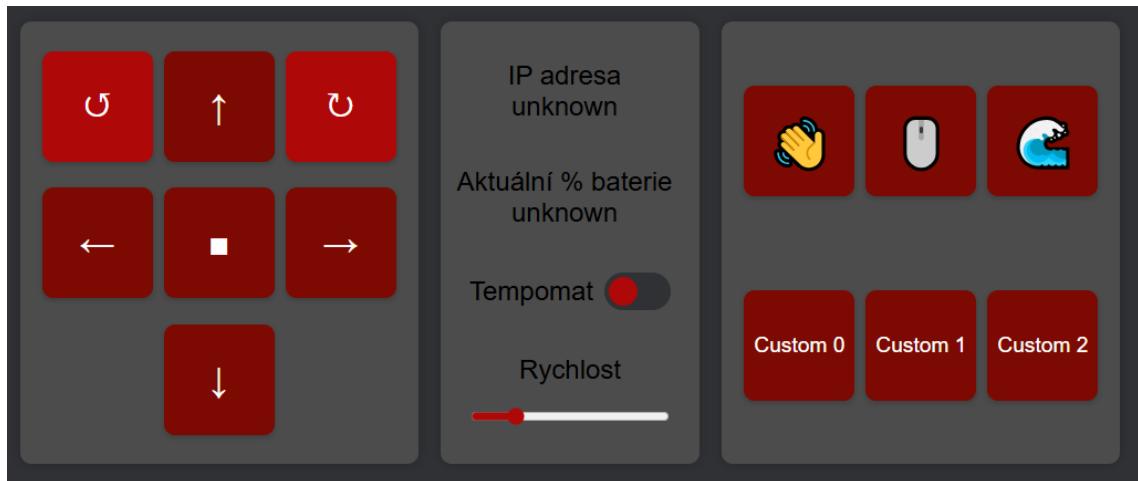
16.2 Uživatelské prostředí

Psaním HTML webových stránek jsem se zabýval na začátku mé programovací kariéry, což bývalo před pěti lety. V té době jsem pochopil strukturu stránky, však ve kaskádových stylech jsem nenašel cit a se skripty jsem nepracoval vůbec. Z důvodu nedostatku času jsem využil nejmodernější technologie – umělou inteligenci, která mi pomohla se sepsáním těchto problémových částí.

Tímto jsem si vyzkoušel kompletně jiný pohled na programování bez přesného znění JS, který komunikuje s robotem. Pokud tuto práci čte čtenář, který by chtěl začít programovat, tak nikdy nebude pozdě s programováním začít. A při generaci kódu

generovala nespočet chyb, nepřehledný a občas nelogický kód. Debugovat kód po AI je peklem každého programátora.

Pokud bych měl ukázat Achillovu patu tohoto projektu, bude to právě JS skript webové stránky. Naštěstí prošel všemi testy a funguje správně, ale i přes to se zde vyskytuje, v poměru se zbytkem práce, nespočet technických chyb.



Obrázek 33 - Finální podoba UI

Na obrázku je zobrazena chybová hodnota „unknown“, která indikuje, že nebyla načtena odpovídající data (viz Příloha V). Uživatelské rozhraní je implementováno pomocí HTML a JS, které čtou data přímo ze zařízení robota. Vzhledem k tomu, že rozhraní bylo spuštěno v prohlížeči na počítači, kde tato data nejsou dostupná, došlo k zobrazení výchozích chybových hlášení.

16.3 Tempomat a senzor

Původně tento tempomat měl být plnohodnotným autonomním ovládáním. Pavouk jde dopředu, dokud senzor nenarazí na překážku. Následně se otočí na levou stranu, změří vzdálenost, následně na pravou stranu, kde též změří vzdálenost. Následně tyto vzdálenosti porovná a otočí se na stranu, kde byla naměřena větší vzdálenost.

Po konzultaci s M. Klvaňovou jsem tuto myšlenku vyměnit za nápad s tempomatem. Pavouk půjde rovně, dokud nenarazí na překážku a následně uživatel rozhodne následující krok robota. Myslím si, že tato možnost bude lepší pro ovládání a „hraní si“ s pavoukem.

K senzoru TOF050C (viz TOF050C) existuje knihovna s ukázkovým kódem (11), který prakticky není potřeba upravovat. Senzorem lze měřit i osvětlení, které ale měřit není

potřeba. Rovnou jsem využil i možnost různých chybový výsledků, ke kterým jsem přiřadil chybovou hlášku (viz Chybové hlášky). Následně jsem pomocí jednoduché porovnávací podmínky rozdělil pohyb podle vzdálenosti, aby koncept tempomatu fungoval správně.

Přeprogramování na původní myšlenku s autonomním ovládáním by bylo pouze v části algoritmu porovnávací podmínky, která není složitá.

16.4 Pohyb

Základní pohyby jako kroky vpřed apod jsem měl naprogramovány už z minulé časti – ročníkové práce a mohl jsem začít s programováním základní komunikace přes Wi-Fi.

K pohybu servomotoru jsem použil upravenou knihovnu pro pohyb servomotory pro ESP32. Knihovna je prakticky stejná jako pro platformu Arduino, však podporuje i ESP32, což původní knihovna nepodporovala. Stejná, zde myslím, ve smyslu psaní příkazu – nemyslím tedy vnitřní strukturu knihovny.

Pouze jsem doplnil funkci, která tento pohyb rozdělí do jednotlivých mikrokroků, mezi které vloží pauzu o délce časového intervalu z posuvníku na webové stránce. Tento časový rozdíl lze nastavit od 800 do 25000 μs .² Použil jsem jednotku μs z důvodu práce s celým číslem namísto s ms s desetinnými čísly.

```
41
42 void smoothMove(Servo &servo, int start, int end) {
43     int step = (start < end) ? 1 : -1;
44     for (int pos = start; pos != end; pos += step) {
45         servo.write(pos);
46         delayMicroseconds(setSpeed);
47     }
48     servo.write(end);
49 }
```

Obrázek 34 - Ukázka kódu pro regulaci rychlosti pohybu

² Tato informace neplatí pro beta verzi SW, která je přiložena v přílozích (viz Příloha U). Původně byl časový interval od 0-50 ms, přičemž základní hodnota byla 10 ms. Samotný algoritmus mikrokrokování též není součástí beta verze v příloze.

16.5 Mikrokrování servomotoru

Mikrokrok jsem v této práci definoval jako menší krok celého otočení servomotoru – v méém případě o 1° .

Software funguje na bázi počátečního a konečného úhlu. Naprogramována funkce smoothMove má argumenty označení motoru, začátek a konec:

```
void smoothMove(Servo &servo, int start, int end) {...}  
smoothMove(S0; 0; 60);
```

Funkce spočítá orientaci a následně přičítá nebo odčítá jednotlivý mikrokroky.

Mezi mikrokroky je časová mezera o délce, která se nastavuje podle požadavku z webové stránky. Je udána v mikrosekundách.

16.6 Chybové hlášky

Další důležitou funkcí je hlášení chybových hlášek. Z počátku jsem nevěděl, jak chybu oznámit, ale v praxi jsem se setkal s několika příklady. U standardních počítačů se využívá bzučák, který pomocí pípání krátkých a dlouhých signálů označuje chybu. Při servisu automobilů se dnes připojuje diagnostika, která komunikuje s hlavní jednotkou. Pokud jednotka narazí na chybu (např. senzoru), uloží ji a po připojení ji zobrazí na diagnostice. Tím se ulehčuje odhalování chyb a poškození.

Proto jsem se rozhodl využít tyto koncepty a aplikovat je do SW robota. Jakmile vznikne nějaká chyba, rozsvítí se LED. Pro přesné zjištění chyby je nutno připojit pavouka k PC a zobrazit si chybu v konzoli.

Zobrazí pouze aktuální chybu, takže SW neukládá chyby jako ve zmíněných automobilech. Dle mého názoru by ukládání chyb pouze komplikovalo práci s robotem.

16.7 Oprava chyb v kódu

Prvotní myšlenka neměla špatný základ, však docházelo k mnoha technickým i netechnickým chybám.

V rámci komunikace s webovou stránkou docházelo k neustálému spamování komunikace, padání ESP32 nebo celkové nefunkčnosti.

16.7.1 Méně závažné chyby

Méně závažnými chybami míním chyby, u kterých nedocházelo k restartování procesoru. Jedná se hlavě o filtry proti spamování a zahlcování systému. Řešení těchto chyb je možno označit jako „optimalizování“.

16.7.1.1 Filtr proti spamu

Při prvních testech docházelo k velkému zahlcení komunikace mezi webovou stránkou a konzolí v sériovém monitoru.

Problém byl v několika nezaznamenatelných stlačeních tlačítka v době vypracovávání úlohy. To jsem vyřešil jednoduchou podmínkou. Podmínka stojí na odpovědi od ESP32. Dokud odpověď nepřijde, stránka neodešle další příkaz. Jsou zde nastaveny výjimky jako dokončovací příkaz nebo příkazy na nastavení rychlosti, měření napětí apod, protože u těchto příkazů není potřeba kontrola s odpovědí. Tato podmínka se vztahuje pouze na pohybové příkazy.

16.7.1.2 Fronta animací

Po vyřešení problému se stavovým automatem (viz „Vynalezení“ stavového automatu) vznikl problém s frontou příkazů. Příkazy během provádění úloh vytvářely frontu, která se postupně vypracovávala. To znamenalo, že pavouk vykonával onu frontu příkazů, dokud nebyla úplně prázdná.

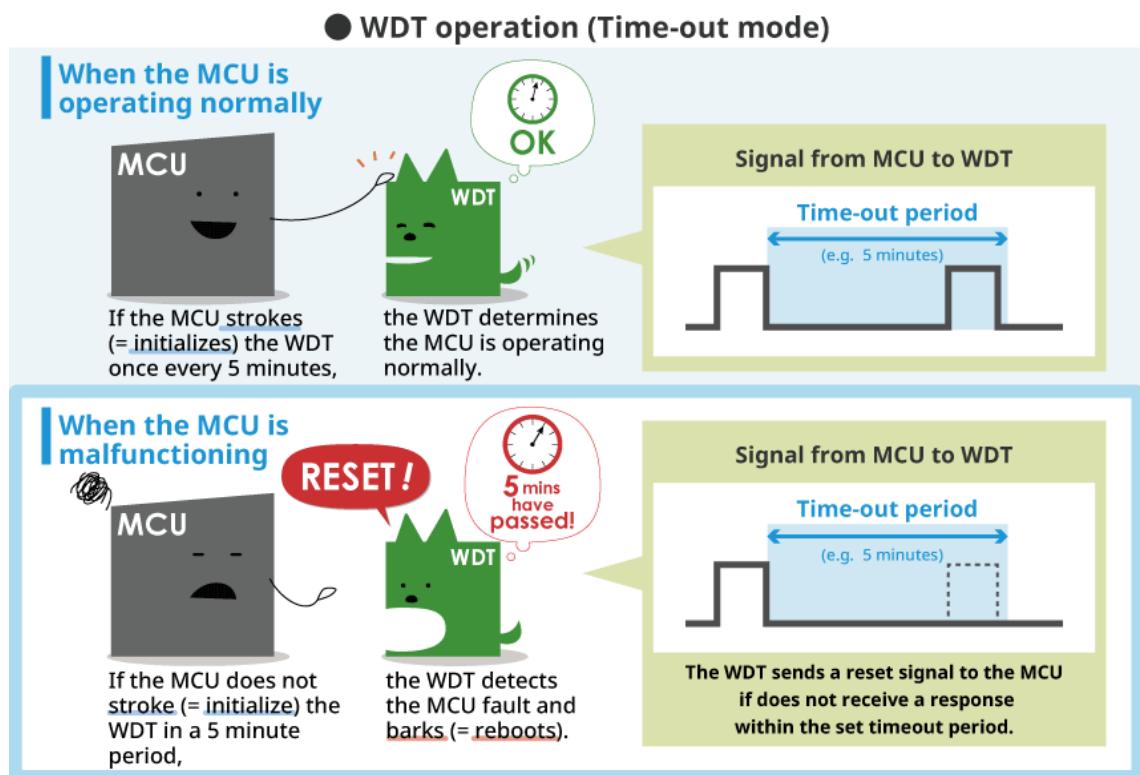
Vyřešil jsem to úpravou funkce ve stavovém automatu. Do této funkce jsem přidal jednoduchou boolean podmínu (pravda, nebo nepravda), zda je pavouk v pohybu, anebo ne. Tím jsem vyřešil, že ESP32 nebude přijímat další příkazy, pokud není tato podmínka splněna.

16.7.2 Velice závažné chyby

Při těchto chybách docházelo k restartování ESP32. Tyto chyby byly pro mě nejvíce přínosné, protože jsem ignoroval základní podstaty hardwarové stránky práce. Docházelo k opakovanému spuštění WatchDog timeru, který inicializoval procesor jako zamrzlý. Z tohoto důvodu docházelo k restartování.

WDT je klíčový bezpečnostní mechanismus, který chrání procesor před zaseknutím v nekonečné smyčce nebo jiném kritickém selhání softwaru. Je možné ho úplně vypnout,

ale to by byla zásadní technická chyba, protože by se tím zvýšilo riziko nevratného zablokování procesoru. (4) (12)



Obrázek 35 - WDT (12)

V tomto projektu WDT není natolik potřeba, protože se jedná o pouhý model a experimentální zařízení. U více sofistikovaných projektů, kde by mohlo hrozit nebezpečí lidského života (např. ovládání raketoplánu, kritické bezpečnostní systémy), WDT musí být vždy aktivní. Proto jsem se rozhodl ho ponechat i zde, abych si osvojil správné zásady programování spolehlivého softwaru a vyhnul se nekorektním a technickým chybám psaní softwaru, jako je obcházení WDT kvůli nedostatkům v kódu a zkušeností.

16.7.2.1 Využití druhého jádra

ESP32 (viz ESP32) obsahuje dvě jádra. To byla pro mě zásadní změna, protože ATMega mikrokontrolery (Arduino platforma apod), které jsem využíval přes ESP32, mají pouze jedno hlavní jádro.

Výpočetní jádro se dá považovat za jednu linku výpočtu. Pokud procesor obsahuje pouze jedno jádro, lze s ním vypočítat pouze jednu operaci za určitý čas daný frekvencí procesoru. Pokud procesor obsahuje více jader, může vypočítávat více výpočtu paralelně mezi jednotlivými jádry.

V rámci tohoto softwaru jsem rozdělil příkazy na dvě poloviny – hlavní a vedlejší. Na vedlejší se provádí postupný pohyb servomotorů a na hlavní polovině funguje Wi-Fi a měření vzdálenosti a napětí.

Toto rozdelení jsem použil, protože změření napětí je časově krátký výpočet, který neblokuje ostatní výpočty. Výpočet vzdálenosti funguje pouze v případě, kdy je zapnut tempomat. V tomto případě nedochází k široké komunikaci s příkazy a pouze se odesílájí data jako naměřené napětí apod.

Bohužel samotné rozdelení operací nestačilo, aby software byl správně optimalizován.

16.7.2.2 Dlouhá odpověď

K dalším restartům docházelo z prostého důvodu. Prvotně jsem naprogramoval software, aby se zavolaná část prováděla před odesláním odpovědi. To znamená, že po přijmutí příkazu se automaticky začne vypracovávat výpočet a následně se odešle odpověď.

Před touto myšlenkou jsem nevěděl, že WDT kontroluje čas mezi přijmutím požadavku a odpovědi, což dnes vidím jako plně logické. Jakmile se odešle požadavek a do daného časového intervalu se nevrátí odpověď, procesor je zamrzlý. Bohužel v tomto případě nebyl procesor zamrzlý, ale pouze špatně naprogramovaný.

16.7.2.3 „Vynalezení“ stavového automatu

Po hodinách programování jsem dostal nápad. Jestliže je příčinou restartování příliš dlouhý časový interval mezi odesláním požadavku a jeho odpovědí, mohu tento časový interval jednoduše zkrátit. Vytvořím jednoduchý mezikrok, který bude zapisovat aktuální příkaz do proměnné, která následně bude úlohy a animace spouštět.

Například: Když stránka odešle příkaz DOPŘEDU, proměnná POHYB se bude rovnat „1“. Jakmile se POHYB bude rovnat „1“, software spustí na druhém jádře animaci s pohybem DOPŘEDU. Rovnou jsem využil i dokončovací příkaz (viz Prvotní myšlenka provedení), který nastaví proměnnou na „0“, což znamená HOME – domácí pozice.

S pomocí umělé inteligence jsem zjistil, že touto myšlenkou jsem znova vynalezl tzv. stavový automat, který tento problém řeší už několik let. Objevení tohoto řešení vlastní cestou mi potvrdilo, že disponuji inženýrským myšlením a značným potenciálem pro inovace.

Tímto vylepšováním softwaru jsem dospěl k funkčnímu kódu bez hrubých technických chyb a se správnou funkcí WDT.

16.8 Další funkce softwaru

Tyto funkce jsem programoval postupem času společně s řešením chyb. Hlavním cílem bylo vytvořit testy, které proběhnou při zapnutí robota (viz Obrázek 36). Dalšími funkcemi jsou měření napětí (viz Měření napětí), stav serveru – zda je uživatel připojen, upozornění uživatele na otočení telefonu na šířku apod (viz Příloha U).

```
22:57:17.731 ->
22:57:17.731 -> Configuring Setup on core: 1
22:57:17.731 -> Configuring AP
22:57:17.811 -> AP IP address: 192.168.4.1
22:57:17.811 -> VL6180X Test - Passed
22:57:17.811 -> Configuring animations on core: 0
22:57:17.811 -> Core initialization - Passed
22:57:17.811 -> Starting animation: lie done
22:57:17.811 -> Server started
22:57:17.811 -> Server open
22:57:18.781 -> lie done
```

Obrázek 36 - Výpis z konzole po zapnutí ESP32

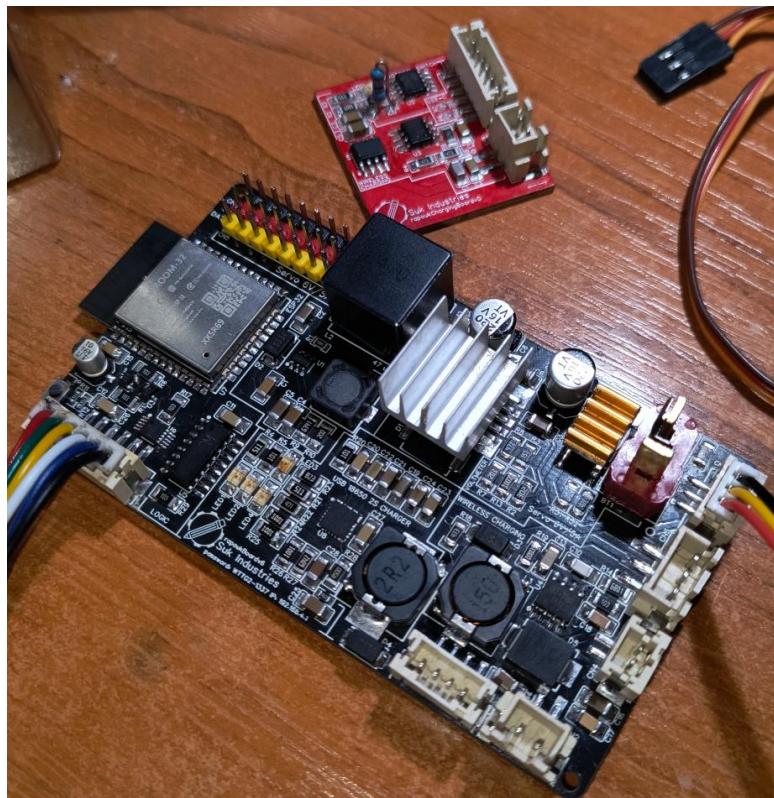
17 Finální podoba

K celému dokončení tohoto projektu jsem rám pavouka eloxoval na červenou barvu. Eloxování probíhalo podle návodu. (13) Dalším krokem bylo pogumování nožiček robota, aby byl součinitel tření větší. Využil jsem dvousložkovou hmotu Vytaflex 60, kterou jsem využil v minulosti na silentbloky. Bohužel při použití se hmota nechovala jako nová a nedošlo ke ztuhnutí materiálu. Pro potřebnou pevnost jsem použil černý silikon. Tímto vzniklo pogumování, které je z vnější strany obalu pevný, ale vnitřní, na nožičce, je měkký a tvárný.



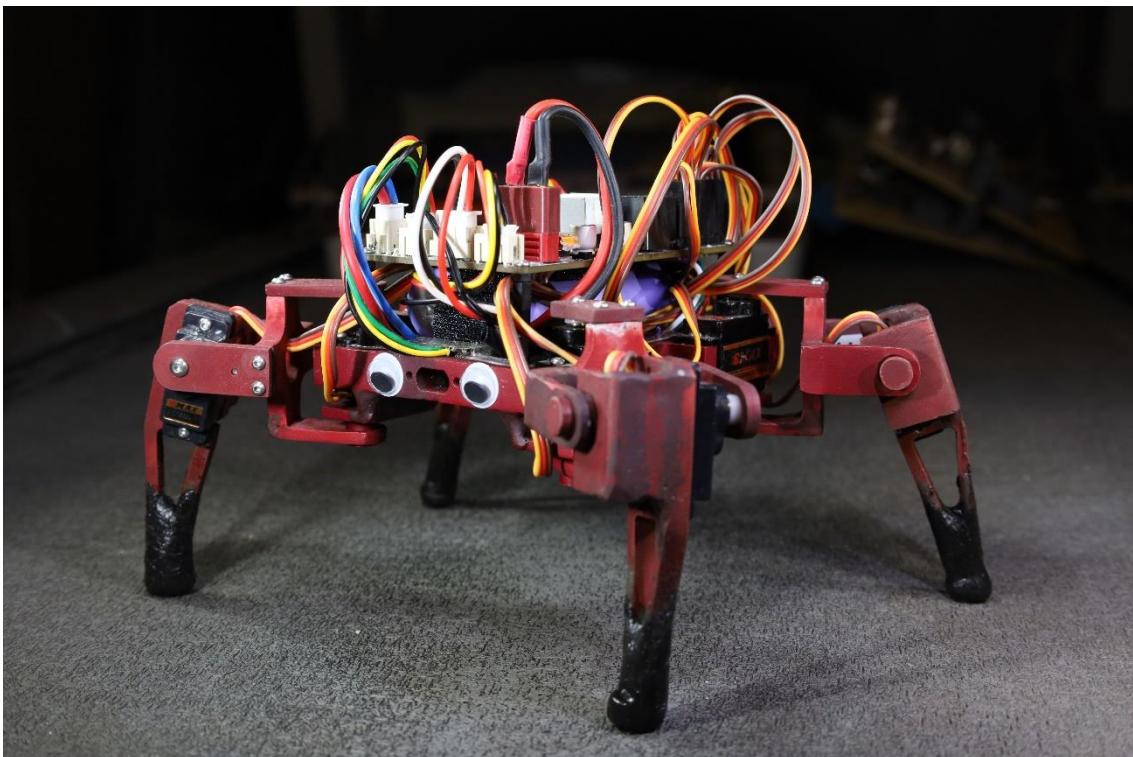
Obrázek 37 - Proces pogumování

Dalším krokem bylo propojení hlavní desky s perifériemi a uchycení kabeláže. Z protokolu o zahřívání jsem přilepil chladiče na potřebná místa. Aktualizace firmwaru z verze 1.1.0 na 3.4.0 a nahrání softwaru (14) byly poslední potřebné kroky pro složení celého pavouka a dokončení projektu Rapouk. A samozřejmě jsem nemohl zapomenout na očička (viz Obrázek 39).



Obrázek 38 - Finální verze PCBs

V době psaní a odevzdání této práce byl software ve fázi beta verze a nebyl zcela dokončen. Finální verze softwaru se proto může lišit s verzí na CD.



Obrázek 39 - Finální podoba robotického pavouka Rapouk

18 Závěr

Zadání bylo vytvořit konstrukci a elektroniku s plnohodnotným softwarem a ovládáním přes telefon pomocí Wi-Fi sítě. Z tohoto hlediska je práce, ročníková i dlouhodobá, splněna. Došlo k několika dalším vylepšením, se kterými jsem na začátku úplně nepočítal jako eloxování, pogumování nohou pro nižší koeficient tření nebo elektroniku nabíjení na základní PCB robota. Též zde nebyl předpoklad pro senzor pro částečné autonomní ovládání.

Dalším hlediskem bylo především získat zkušenosti. Nejvíce zkušeností jsem získal ve směrech jako eloxování, pochopení základních principů elektronických součástek jako tranzistor, kondenzátor nebo cívka. V SW části bylo největší zlepšení v programování stavového automatu a práce s WDT nebo HTTP protokolem. Největším potěšením bylo využití AI pro pomoc s projektem samotným, což je zkušenost, která bude měnit budoucnost a celý svět, nejen inženýrství.

Bohužel jsou zde i nějaká negativa. Hlavním minusem je částečná podpora na iOS, kde se mi nepodařilo napsat skript takový, aby celý správně fungoval na Safari (prohlížeč na iOS zařízeních). Dalším bodem jsou pohyby doleva a doprava, které by se mohly o dost vylepšit.

Včetně ročníkové části mi práce časově zabrala 631 hodin a projekt probíhal po 797 dní, což ukazuje dlouhodobé úsilí a vytrvalost pracovat na dlouhém projektu. Tato vlastnost bude v budoucím uplatnění více než potřebná (viz Příloha R).

A co toto celé úsilí pro mě znamená? Od začátku střední školy jsem se zaobíral myšlenkou mého budoucího zaměření a výběrem vysoké školy. Neuvěřitelně se mi zalíbilo ČVUT FEL, obor Kybernetika a robotika. Přestože jsem se už v té době věnoval dlouhá léta robotice, nebyl jsem si jistý, zda mám dostatečně pevný základ pro budování vědeckotechnické kariéry. Po tomto projektu se domnívám, že jakýsi potenciál, po stránce techniky, ve mně je. Stránka techniky se vždy dá vylepšit učením a vzděláváním se v oboru. Naopak, co se jednoduše naučit nelze, je chut', zápal a výdrž, kterou disponuji při plnění dlouhodobých a složitých cílů. Věřím, že tato disciplína je můj „klíč“ ke štěstí a úspěchu.

A také doufám, že Rapouk bude nejen dobrým partákem, ale i také osobní vizitkou a reprezentativní ukázkou mých schopností pro první roky vysokoškolského studia.

19 Seznam použité literatury

1. **GreatScott!** Electronic Basics #23: Transistor (MOSFET) as a Switch. *YouTube*. [Online] 24. Červenec 2016. [Citace: 10. Duben 2025.] https://www.youtube.com/watch?v=o4_NeqIJgOs.
2. **TutorChase.** What is the significance of the $P=I^2R$ equation? *TutorChase*. [Online] [Citace: 10. Duben 2025.] <https://www.tutorchase.com/answers/ib/physics/what-is-the-significance-of-the-p-i-2r-equation>.
3. **Eashub.** 1oz vs 2oz měď PCB, jaký je rozdíl? *Eashub*. [Online] [Citace: 10. Duben 2025.] <https://cs.eashub.com/1oz-vs-2oz-copper-pcb/>.
4. **OpenAI.** ChatGPT. *ChatGPT*. [Online] OpenAI. [Citace: 10. Duben 2025.] chatgpt.com.
5. **microcontroller.it.** Un convertitore buck per sostituire i 78xx. *microcontroller.it*. [Online] 10. Leden 2019. [Citace: 10. Duben 2025.] <http://www.microcontroller.it/test/DD4012SA/dd4012sa.htm>.
6. **calculator, Ohms law.** Voltage Divider Calculator. *Ohms law calculator*. [Online] [Citace: 10. Duben 2025.] <https://ohmslawcalculator.com/voltage-divider-calculator>.
7. **TV, Elektor.** How-to: Accurate Voltage Measurements with Arduino. *YouTube*. [Online] 9. Září 2022. [Citace: 4. Duben 2025.] <https://www.youtube.com/watch?v=gw72g4WBz-U>.
8. **LiPolBaterie.** Na kolik V vybijet lipol baterie. *LiPolBaterie*. [Online] [Citace: 10. Duben 2025.] <https://www.lipolbaterie.cz/navody/na-kolik-v-vybijet-lipol-baterie/>.
9. **Novotná, Barbora.** USB-C jako jednotný standard: Nová éra nabíjení v Evropské unii. *tvorimevropu.cz*. [Online] Tvoříme Evropu, 28. Leden 2025. [Citace: 10. Duben 2025.] <https://tvorimevropu.cz/2025/01/28/usb-c-jako-jednotny-standard-nova-era-nabijeni-v-evropske-unii/>.
10. **Wikipedia.** JST connector. *en.wikipedia.org*. [Online] Wikipedia, 21. Březen 2025. [Citace: 10. Duben 2025.]
11. **Adafruit.** Adafruit VL6180X Time of Flight Micro-LIDAR Distance Sensor Breakout. *Adafruit*. [Online] 8. Březen 2024. [Citace: 10. Duben 2025.]

<https://learn.adafruit.com/adafruit-vl6180x-time-of-flight-micro-lidar-distance-sensor-breakout/wiring-and-test>.

12. **ABLIC.** What is a watchdog timer (WDT)? *ablic.com*. [Online] ABLIC Inc. [Citace: 10. Duben 2025.] <https://www.ablic.com/en/semicon/products/automotive/automotive-watchdog-timer/intro/>.

13. **Hroch, Miloš.** Eloxování hliníku. *astronom.cz*. [Online] [Citace: 10. Duben 2025.] <https://www.astronom.cz/procyon/chemistry/elox.html>.

14. **Espressif Systems (Shanghai) Co., Ltd..** Released Firmware. *docs.espressif.com*. [Online] Espressif Systems (Shanghai) Co., Ltd.. [Citace: 10. Duben 2025.] https://docs.espressif.com/projects/esp-at/en/latest/esp32/AT_Binary_Lists/esp_at_binaries.html.

Neoznačené odstavce, obrázky, tabulky a další materiály jsou součástí mých původních poznatků, které vycházejí z mých vlastních výzkumů, analýz a zkušeností, a jsou tudíž mými vlastními zdroji volně k použití.

20 Seznam grafických objektů

Obrázek 1 - Čistý PCB bez překrývací masky	10
Obrázek 2 - UI EasyEDA	11
Obrázek 3 - Příklad schéma zapojení	12
Obrázek 4 - Návrh PCB	14
Obrázek 5 – Ohlášení chyby DRC.....	14
Obrázek 6 - ESP32-WROOM-32	16
Obrázek 7 - Bezdrátové nabíjení XKT s USB-C a DC motorem	19
Obrázek 8 - PCB bezdrátového nabíjení – reverzní inženýrství.....	20
Obrázek 9 - TOF050C	22
Obrázek 10 - Schéma zapojení IR senzoru	23
Obrázek 11 - JST PH a T-DEAN v porovnání s nehtem	24
Obrázek 12 - Rozdělení PCB.....	25
Obrázek 13 - Horkovzdušná pistole.....	26
Obrázek 14 - Spálená PCB	26
Obrázek 15 - Pájecí stanice z žehličky	27
Obrázek 16 - Druhá generace	28
Obrázek 17 - 3D render testovací PCB.....	29
Obrázek 18 - Senzorová deska.....	30
Obrázek 19 - Čtvrtá generace	31
Obrázek 20 - Elektronika nabíjecího doku	32
Obrázek 21 - Nabíjecí dok	32
Obrázek 22 - Pátá generace	33
Obrázek 23 – 3D render šesté generace.....	34
Obrázek 24 - Schéma zapojení při měření	37

Obrázek 25 - Měření	38
Obrázek 26 - Horní strana PCB	39
Obrázek 27 - Spodní strana PCB	39
Obrázek 28 - Teplota spodní strany PCB z termokamery	40
Obrázek 29 - Teplota spodní strany PCB pomocí teploměru	41
Obrázek 30 – Zapojení.....	41
Obrázek 31 - Měření termokamerou.....	42
Obrázek 32 - Fotografie z termokamery	42
Obrázek 33 - Finální podoba UI	45
Obrázek 34 - Ukázka kódu pro regulaci rychlosti pohybu	46
Obrázek 35 - WDT (12).....	49
Obrázek 36 - Výpis z konzole po zapnutí ESP32	51
Obrázek 37 - Proces pogumování	52
Obrázek 38 - Finální verze PCBs	52
Obrázek 39 - Finální podoba robotického pavouka Rapouk	53

Fotografie byly foceny na: Canon EOS RP s objektivem RF 24-105 mm d/4-7,1

Motorola Edge 40 Neo

21 Seznam tabulek

Tabulka 1 - Hodnota proudu na šířku cesty (3)	13
Tabulka 2 - Podmínky měření	35
Tabulka 3 - Použité pomůcky	36
Tabulka 4 - Měření teplot	40
Tabulka 5 - Výsledek měření napětí	41

22 Seznam rovnic

Rovnice 1 - Joulův zákon (2).....	12
Rovnice 2 - Dělička napětí (6).....	23
Rovnice 3 - Zpřesnění změřené hodnoty	23
Rovnice 4 - Změřené napětí (7)	24

23 Seznam zkratek

Zkratka	Původní slovo	Význam
CAD	Computer aided design	Počítačem podporované projektování
CAM	Computer aided manufacturing	Výroba podporovaná počítačem
CNC	Computer Numerical Control	Výpočtové ovládání pomocí počítače
PCB	Printed circuit board	Deska plošných spojů
DPS	Deska plošných spojů	Deska plošných spojů
RC	Remote control	Dálkové ovládání
SW	Software	Program
HW	Hardware	Fyzické komponenty výpočetní techniky
EDA	Electronic Design Automation	Software pro návrh DPS
IDE	Integrated Development Environment	Integrované vývojové prostředí
UI	Users interface	Uživatelské prostředí
DRC	Design Rule Checking	Návrhová kontrola pravidel
USB	Universal Serial Bus	Univerzální sériová sběrnice
IoT	Internet of Things	Internet věcí – síť propojených objektů a zařízení
GPIO	General-purpose input/output	Univerzální vstupní/výstupní pin
ADC	Analog-to-digital converter	Převodník z analogového signálu na digitální
SPI	Serial Peripheral Interface	Synchronní sériový datový protokol
WDT	WatchDog timer	Počítačová periferie, která resetuje systém při jeho zacyklení
PWM	Pulse Width Modulation	Pulzně šířková modulace
I2C	Two Wire Interface	Sériová sběrnice
ČVUT FEL	České vysoké učení technické v Praze – fakulta elektrotechnická	Název univerzity

QC	QuickCharge	Rychlonabíjecí protokol
IR	Infrared	Infračervené záření
LED	Light Emitting Diode	Polovodičová dioda vyzařující světlo
SMD	Surface Mounted Device	povrchově namontována (naletovaná) zařízení (součástky)
THT	Through-Hole Technology	Starší technologie DPS
UART	Universal Asynchronous Receiver Transmitter	Asynchronní přenos a příjem informací přes sériový port
VCC	Coomon Collector Voltage	Kladný pól napětí
GND	Ground	Mínus (uzemnění)
AI	Artificial intelligence	Umělá inteligence
NTC	Negative Temperature Coefficient	Negativní teplotní koeficient
JS	JavaScript	Skriptovací jazyk JavaScript
Wi-Fi	Wireless Fidelity	Skupina bezdrátových síťových protokolů

24 Seznam příloh

A Datové listy

- Příloha A AOD423.pdf
- Příloha B ES08MA.pdf
- Příloha C ESP32-WROOM-32.pdf
- Příloha D FSUSB42.pdf
- Příloha E CH224K-0.pdf
- Příloha F CH224K-1.pdf
- Příloha G CH340.pdf
- Příloha H IP2326.pdf
- Příloha I ME3116.pdf
- Příloha J XL4015.pdf

B Schémata a fotografie

- Příloha K rapoukBoardv3.jpg
- Příloha L rapoukBoardv5 schematic.pdf
- Příloha M rapoukBoardv6 schematic.pdf
- Příloha N rapoukChargingPCBv0 schematic.pdf

C Gerber files

- Příloha O Gerber_rapoukBoardv6.zip
- Příloha P Gerber_rapoukChargingBoardv0.zip

D Ostatní

- Příloha Q collatzConjectureTest.zip
- Příloha R Rapouk.xlsx
- Příloha S RapoukDMP.docx
- Příloha T RapoukDMP.pdf
- Příloha U rapoukSoftwareBeta.zip
- Příloha V rapoukUI.html