



Ekonomická  
fakulta  
Faculty  
of Economics

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích

Ekonomická fakulta

Katedra aplikované matematiky a informatiky

Diplomová práce

# Otevřená interaktivní robotická platforma pro domácí využití

Vypracoval: Bc. Martin Stropek

Vedoucí práce: Ing. Jan Fesl Ph.D.

České Budějovice 2023

## ZADÁVACÍ PROTOKOL MAGISTERSKÉ PRÁCE

**Student: Bc. Martin Stropek**.....  
(jméno, příjmení, tituly)

**Obor – zaměření studia:** Aplikovaná informatika

**Katedra:** Ústav aplikované informatiky.....

**Školitel: Ing. Jan Fesl, Ph.D.**.....  
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

**Garant z PřF:** .....  
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

**Školitel – specialista, konzultant:** .....  
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

**Téma magisterské práce: Otevřená interaktivní robotická platforma pro domácí využití**.....

### Popis práce :

Předmětem práce je návrh a implementace mobilního robota, kterého bude možné vzdáleně ovládat prostřednictvím interaktivní aplikace. Robot bude schopen interagovat s okolím (přenášet/přehrávat video a hlas oběma směry), popř. dokáže své okolí ovlivnit. Tento typ robota bude tedy možné typicky využít k monitorování (např. dohledování domácích mazlíčků), resp. v místech, kde uživatel potřebuje interagovat s okolím, avšak fyzicky být přítomen nemůže.

Celý projekt bude realizován jako otevřený, tzn. na bázi open-source, veškeré zdrojové kódy, schémata zapojení a postup pro zprovoznění budou zveřejněny na vhodné veřejně dostupné on-line platformě (GitHub/GitLab). Pro výrobu robota budou voleny finančně dostupné součástky či moduly, umožňující snadné masivní rozšíření mezi členy uživatelské komunity. Důraz při vývoji bude kladen i na to, aby návrh zařízení byl pojat modulárně a bylo tudíž možné bez větších komplikací řešení rozšířit o další případné funkcionality.

### Cíle práce :

- 1) Provést důkladnou hardwarovou a softwarovou rešerši pro realizaci robota (řízení, pohon, interakce, komunikace).
- 2) Navrhnout, implementovat a odladit robota po hardwarové stránce na pečlivě vybrané mikropočítačové platformě (Raspberry Pi, Arduino, Nvidia Jetson, atd.), která bude stěžejní pro realizaci pokynů od uživatele – tj. pomocí impulzů řídit jeho pohyb, resp. zajišťovat veškeré další funkcionality pro interakci s okolím.
- 3) Navrhnout, implementovat a otestovat ovládací aplikaci, pomocí které bude možné přehrávání/příjem vizuálních a zvukových jevů a ovládat pohyb robota. Student uváže využití aditivních technologií jako jsou např. 3D brýle popřípadě joystick pro dosažení co nejlepší interakce s okolím.

## Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to – v nezkrácené podobě – elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 13. 4. 2023

Podpis...*Stropek*.....

## **Poděkování**

Rád bych poděkoval panu Ing. Janu Feslovi, Ph.D. za cenné rady, užitečné připomínky a trpělivost při vedení mé diplomové práce.

# Obsah

1	Úvod .....	5
2	Cíle .....	6
3	Teoretická část .....	7
3.1	Pohyb robotů .....	7
3.1.1	Kola .....	7
	.....	8
3.1.2	Pásky .....	8
3.1.3	Nohy .....	9
3.1.4	Řízení směru jízdy (zatáčení) .....	9
3.2	Pohon .....	10
3.2.1	Elektromotory .....	10
3.2.2	Převodovka .....	11
3.2.3	Ovladač motoru .....	11
3.2.4	PWM regulace .....	12
3.3	Indukční nabíjení .....	13
3.4	Vývojové platformy .....	14
3.4.1	Arduino .....	14
3.4.2	Raspberry Pi .....	15
3.4.3	Nvidia Jetson .....	16
3.4.4	Hardwarové akcelerátory .....	16
3.5	Detekce překážek .....	17
3.5.1	Akustická čidla .....	17
3.5.2	Optická čidla .....	18
3.6	Rozpoznávání objektů pomocí kamery .....	18
3.6.1	Detekce pomocí klíčových bodů .....	18
3.6.2	Klasifikátory .....	18

3.6.3	Kaskáda klasifikátorů .....	19
3.6.4	Haarovy příznaky .....	19
3.6.5	Využití umělé inteligence a neuronových sítí.....	20
3.7	Webová komunikace .....	21
3.7.1	Komunikační protokoly .....	21
3.7.2	HTTP .....	21
3.7.3	TCP 3-Way Handshake .....	22
3.7.4	WebSocket .....	22
3.8	WEB RTC .....	24
3.9	Formáty videa.....	25
3.9.1	MJPEG.....	25
3.9.2	H.264 a H.265 .....	25
4	Možnosti vzdáleného přístupu.....	26
4.1	Přímý přístup přes veřejnou IP adresu .....	26
4.1.1	Přesměrování portu .....	26
4.1.2	DDNS.....	26
4.2	VPN.....	27
4.3	Využití služeb třetích stran.....	27
5	Metodika.....	28
5.1	Hardwarová implementace.....	28
5.2	Řídící aplikace.....	28
5.3	Audiovizuální přenos .....	29
5.4	Webové rozhraní .....	29
6	Použitý hardware .....	30
6.1	Řídící jednotka .....	30
6.2	Podvozek s motory .....	31
6.3	Ovladač motorů.....	32

6.4	Kamera .....	32
6.5	Otáčení kamery .....	33
6.6	Detektor překážek .....	33
6.7	Audio.....	34
6.8	Mechanická konstrukce.....	35
6.9	Napájení .....	35
6.10	Indikátor stavu baterie .....	36
7	Hardwarová implementace .....	37
7.1	Číslování pinů Raspberry Pi.....	37
7.2	Schéma ovladače motorů .....	38
7.3	Zapojení motorů .....	39
7.4	Zapojení dálkoměru .....	40
7.5	Zapojení servomotorů .....	41
7.6	Zapojení kamery.....	41
7.7	Zapojení zesilovače s reproduktory .....	42
7.8	Konfigurace audio vstupu a výstupu v systému.....	42
7.9	Zapojení voltmetru .....	43
8	Použitý software .....	44
8.1	Operační systém .....	44
8.2	Programovací jazyk pro řízení robota .....	44
8.2.1	Knihovny .....	44
8.3	Software pro zajištění přenosu videa a audia .....	45
8.4	Uživatelské webové rozhraní .....	45
9	Softwarová implementace .....	46
9.1	Architektura řešení .....	46
9.2	Architektura řídicí aplikace .....	47
9.3	Ovládání motorů.....	47

9.4	Ovládání servomotorů .....	51
9.5	Obsluha dálkoměru .....	52
9.6	Zjišťování informací o síti .....	52
9.7	Měření stavu baterie .....	53
9.8	Webová aplikace .....	55
9.8.1	Ovládání pohybu robota.....	55
9.8.2	Odesílání informací ze serveru na klienta.....	57
9.9	Ovládání pomocí joysticku .....	59
9.10	Přenos videa a audia .....	61
9.10.1	Rozpoznávání objektů v obraze:.....	62
10	Výsledná podoba uživatelského rozhraní .....	65
11	Testování .....	66
12	Možný rozvoj práce .....	67
13	Závěr.....	68
I.	Summary and keywords .....	69
II.	Seznam použitých zdrojů .....	70
III.	Seznam obrázků.....	78
IV.	Seznam tabulek.....	79
V.	Seznam příloh.....	80
a.	Zdrojové kódy.....	80
b.	Fotografie zařízení .....	80
VI.	Přílohy .....	80



# 1 Úvod

V dnešní moderní době se neustále zvyšuje poptávka po možnostech monitorování a kontrolování míst, kde nemůžeme být fyzicky přítomni. Zároveň stále roste obliba robotů a celkově IT zařízení, které si každý může přizpůsobit podle svého, přičemž nové technologie jsou stále dostupnější.

S tímto trendem vznikl inovativní nápad na vývoj mobilního robota, kterého si potenciální uživatelé mohou sami sestavit za nízké pořizovací náklady. Původní záměr tohoto projektu spočíval v návrhu robota, který by byl využitelný pro hlídání domácích zvířat. V porovnání s běžnými IP kamerami by tento robot mohl po bytě libovolně přejíždět a se zvířetem interagovat tak, že by mohlo přes vestavěné reproduktory slyšet přímo hlas svého pánička. Na projekt však lze nahlížet mnohem komplexněji tak, že robot může být užitečný kdekoli, kde uživatel potřebuje interagovat s okolím, avšak fyzicky být přítomen nemůže. Uživatelé rovněž mohou dle vlastních požadavků, potřeb a schopností přidávat k základnímu modelu robota další funkcionality.

Tento robot může být v neposlední řadě výborným zdrojem zábavy a vzdělávání pro všechny věkové kategorie a může sloužit jako výuková platforma, která přiláká nejen mladší generace do světa robotiky.

## 2 Cíle

Cílem práce je navrhnout a implementovat funkčního mobilního robota, kterého bude možné vzdáleně ovládat prostřednictvím interaktivní webové aplikace. Robot bude umět přenášet video a obousměrné audio.

Celý projekt bude realizován jako otevřený, tzn. na bázi open-source. Veškeré zdrojové kódy, schémata zapojení a postup pro zprovoznění budou zveřejněny. Pro výrobu robota budou voleny finančně dostupné součástky či moduly, umožňující snadné masivní rozšíření mezi členy uživatelské komunity. Důraz při vývoji bude kladen i na to, aby návrh zařízení byl pojat modulárně a bylo tudíž možné bez větších komplikací řešení rozšířit o další případné funkcionality.

### **Kroky k naplnění hlavního cíle:**

- 1) Provést důkladnou hardwarovou a softwarovou rešerši pro realizaci robota (řízení, pohon, interakce, komunikace).
- 2) Navrhnout, implementovat a odladit robota po hardwarové stránce na pečlivě vybrané mikropočítačové platformě (Raspberry Pi, Arduino, Nvidia Jetson, atd.), která bude stěžejní pro realizaci pokynů od uživatele – tj. pomocí impulzů řídit jeho pohyb, resp. zajišťovat veškeré další funkcionality pro interakci s okolím.
- 3) Navrhnout, implementovat a otestovat ovládací aplikaci, pomocí které bude možné přehrávání/příjem vizuálních a zvukových jevů a ovládání pohybu robota.

## 3 Teoretická část

Tato kapitola teoreticky popisuje různé technologie, jejich vlastnosti, výhody a nevýhody a principy se kterými je nutné se seznámit před realizací praktické části práce.

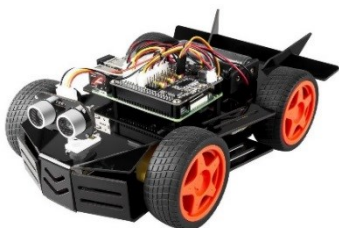
### 3.1 Pohyb robotů

Existuje celá řada různých platforem, které mohou být využity jako základ mobilního robota. Jelikož cílem práce je sestavení pojízdného robota pro domácí využití, omezíme se v této kapitole na srovnání různých způsobů pohybu pozemních robotů. To nám později pomůže zvolit vhodný podvozek.

#### 3.1.1 Kola

Nejběžnějším způsobem pro uvedení robota do pohybu je roztočení kol. Zde se nabízí mnoho různých možností provedení podle počtu kol, způsobu zatáčení a rozdělení hnací síly na jednotlivá kola. Podvozek může fungovat na podobném principu jako u automobilu, kde směr jízdy určuje natočení kol, nebo se ke změně směru jízdy využívá difference otáček hnaných kol (viz kapitola: řízení směru jízdy 3.1.4). Velmi oblíbený je například tříkolový podvozek, kde jsou dvě hlavní hnaná kola a třetí kolo na konci slouží pouze jako opěrný bod. Existuje mnoho různých variant kolových podvozků, některé z nich jsou znázorněny na následujících obrázcích. [1] [2]

Každá z variant podvozku se svými vlastnostmi výrazně liší. Pokud bychom ale chtěli vypsát výhody kol obecně, je to jednoduchost, nízká cena a to, že díky malému odporu potřebují k rozhybání nižší točivý moment. Naopak jejich slabou stránkou je, že mohou snadno prokluzovat a také hůře překonávají nerovnosti nebo překážky. [3]



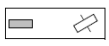

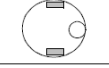




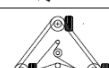
Obrázek 2: 4-kolový podvozek [4]






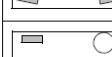
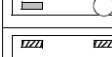
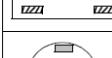

Obrázek 3: 6-kolový podvozek [5]



Obrázek 1: 2-kolový podvozek [6]

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cyber personal robot
3		Two-wheel centered differential drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1
		Three motorized Swedish or spherical wheels arranged in a triangle; omnidirectional movement is possible	Stanford wheel Tribolo EPFL, Palm Pilot Robot Kit (CMU)
		Three synchronously motorized and steered wheels; the orientation is not controllable	"Synchro drive" Denning MRV-2, Georgia Institute of Technology, I-Robot B24, Nomad 200

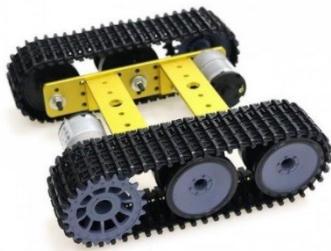
Obrázek 4: Kolové podvozky [2]

# of wheels	Arrangement	Description	Typical examples
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

Obrázek 5: Kolové podvozky (2. část)[2]

### 3.1.2 Pásky

Díky velké styčné ploše s terénem mají pásky výrazně větší tření než kola, což logicky vede k mnohem menší tendenci prokluzovat. Na druhou stranu to však znamená, že je zapotřebí vyšší točivý moment k jejich rozhybání. I z toho důvodu bývají pásová vozidla zpravidla pomalejší než kolová. Kromě již zmíněné vyšší trakce je hlavní výhodou pásů lepší průjezdnost terénem (snáze přejedou různé nerovnosti). Pásky jsou mechanicky složitější konstrukcí z čehož teoreticky vyplývá vyšší cena a poruchovost. Pásová vozidla mohou být také hlučnější [1] [3]



Obrázek 7: Pásový podvozek [7]



Obrázek 6: Pásový podvozek [8]

Změnou velikosti hnacího kola pásu lze velmi účinně měnit převod. Čím je hnací kolo menší, tím je robot zpřevodován více do pomala. Proto je podvozek na pravém obrázku výrazně pomalejší, ale o to disponuje větší silou.

### 3.1.3 Nohy

Roboti, kteří používají ke svému pohybu nohy, se více podobají lidem, případně jiným živočichům. Jejich výhodou je, že mohou potenciálně překonat větší překážky a procházet tak i nerovný terén. Hlavní nevýhoda spočívá v tom, že tento koncept je výrazně složitější, a to jak mechanická konstrukce, tak software pro její řízení. U amatérských projektů se nejčastěji používá šest nebo více nohou, protože tak je zajištěna statická stabilita robota. [1]



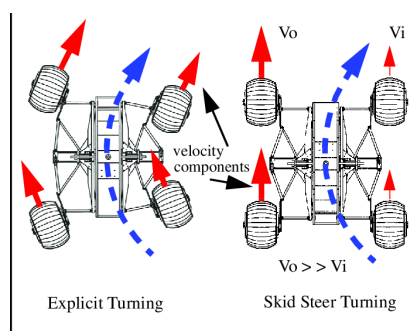
Obrázek 8: Chodící roboti [1]

### 3.1.4 Řízení směru jízdy (zatačení)

Jednou z možností řízení směru jízdy je natáčení kol ve směru jízdy (tak jako třeba u automobilu). Další možností je využít diferenci otáček, což je typické pro pásová vozidla, ale využívá se i u velkého množství kolových robotů.

#### Diference otáček:

Rozdílná rychlost otáčení pásů/kol na pravé a levé straně podvozku způsobí zatočení stroje. Tento způsob zatačení je v robotice velmi oblíbený, neboť konstrukce podvozku je jednoduchá a ani kód pro ovládání robota nemusí být příliš složitý. Další výhodou je výborná manévrovatelnost, vozidlo je možné otočit na místě (opačný směr otáčení pásů/kol). Nevýhodou je, že pásy/kola jsou při zatačení vystaveny velkému působení bočních sil, což může vést k poškození podkladu nebo opotřebení pásů/kol. Tímto způsobem řízení je velmi obtížné dosáhnout plynulého zatačení, zejména ve vysokých rychlostech, je tak vhodný pro pomalou jízdu [9] [10]



Obrázek 9: způsoby řízení [10]

## 3.2 Pohon

### 3.2.1 Elektromotory

Elektromotory jsou standartní pohonnou jednotkou většiny mobilních robotů. Zde je vhodné zmínit dva základní parametry, které udávají jejich vlastnosti.

**Točivý moment** vyjadřuje působení síly na bod vzdálený od osy otáčení.

**Otáčky za minutu (rpm)** udávají rychlost otáčení osy.

#### 3.2.1.1 DC motory

Stejnosemřné motory jsou poháněné stejnosměrným proudem, což je velmi výhodné, jelikož jako zdroj elektrické energie mobilního robota obvykle slouží stejnosměrný akumulátor. Mikropočítače a ostatní elektronické komponenty rovněž pracují se stejnosměrným proudem. [11]

Tento druh motorů je velmi oblíbený i díky jednoduchosti zapojení a řízení. Otáčky motoru se regulují jednoduše změnou velikosti napájecího napětí. Směr otáčení motoru se mění změnou polaroty napájení. Z toho vyplývá, že k řízení motoru nejsou potřeba žádné složité regulátory.

DC motory pracují při vysokých otáčkách (až 10 000 rpm) a disponují relativně nízkým točivým momentem. [11]

#### 3.2.1.2 AC motory

Střídavé motory jsou poháněny střídavým proudem, a proto je pro jejich řízení nutná další chytrá elektronika, pokud je chceme napájet z akumulátoru, který dodává stejnosměrný proud. Výhodou je vysoký výkon, točivý moment a efektivita.

#### 3.2.1.3 Servomotory

Tento typ motoru se vyznačuje tím, že u něj lze nastavit přesnou polohu otáčecí osy (u malých modelářských serv bývá rozsah často omezen na 180°). [11] Využívá se k polohování (například zatáčení kol nebo pohyb kloubů různých ramen) a není vhodný k pohonu.



Obrázek 10: Modelářský servomotor [11]

### 3.2.1.4 Krokové motory

Krokový motor se umí otočit o požadovaný počet kroků. Jeho vlastnosti jsou tedy velmi podobné servomotoru, jen jeho pohyb není spojitý. Principiálně je tento druh motoru jednodušší, tudíž by měl být levnější a spolehlivější. Při současné cenové dostupnosti servomotorů však už nemá takový význam.

### 3.2.2 Převodovka

Zejména u stejnosměrných pohonných jednotek často potřebujeme zvětšit točivý moment a snížit počet otáček. To řeší převodovka. Nejjednodušším řešením bývá zakoupit motor se zabudovanou převodovkou jako jeden celek. [11]

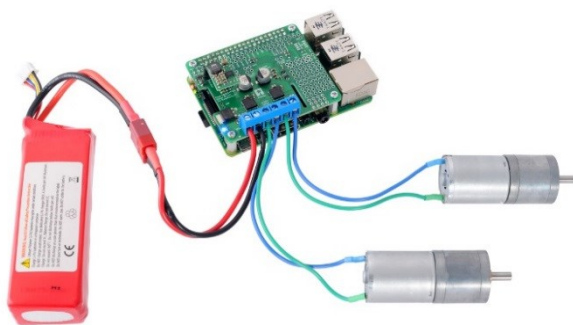
### 3.2.3 Ovladač motorů

Výstupy řídicí jednotky pracují s velmi nízkým napětím a nelze je zatěžovat velkým proudem, proto jimi nelze motory ovládat přímo. Je nutné využít elektrický obvod, který přivede motorům potřebný proud a napětí a umožní nám pomocí výstupů mikropočítače motory programově ovládat.

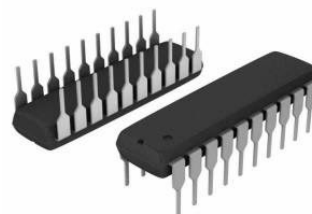
Typický obvod k tomuto určený je **H-můstek**, který lze zakoupit jako samostatný integrovaný obvod a vyřešit zapojení. Další možností je využít hotové rozšiřující desky (také zvané HAT-Hardware Attached on Top), jejíž součástí již bývá dvojice H-můstků a může řešit různé další věci, jako například napájení samotného mikropočítače z akumulátoru.[12]

#### **H-můstek:**

Jedná se o elektrický obvod složený z tranzistorů, který na základě povelů z mikropočítače napájí motory. Oproti obyčejnému MOSFET tranzistoru dokáže otočit polaritu napětí (změnit směr jízdy) a obvykle podporuje i PWM regulaci. [13]



Obrázek 12: Řízení motorů přes rozšiřující desku [14]



Obrázek 11: Integrovaný obvod L293 H-můstek [15]

### 3.2.4 PWM regulace

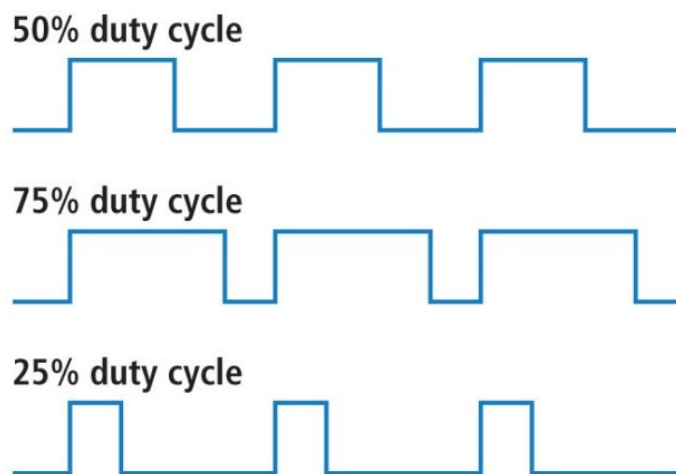
PWM (Pulse Width Modulation), neboli pulsně šířková modulace je způsob regulace v elektrickém obvodu. Používá se tam, kde potřebujeme pomocí digitálního výstupu, který má jen dva stavy (zapnuto/vypnuto), plynule regulovat výstupní hodnotu.

Hodnota přenášeného signálu je zakódována jako poměr mezi stavy napětí zapnuto/vypnuto. Tomuto poměru se říká **střída** (v angličtině *duty cycle*). Přenáší se relativní hodnota v rozsahu 0 až 100 %. Čím delší střída, tím vyšší hodnota. Jedná se tedy v podstatě o velice rychlé „blikání“ s proměnnou délkou bliknutí a konstantní frekvencí.

[16]

PWM se využívá pro regulaci otáček stejnosměrných motorů. Motor chvíli má a chvíli nemá napětí (běží/neběží), čímž díky setrvačnosti můžeme regulovat jeho celkový výkon. Dále se PWM využívá například pro regulaci jasu u levnějších zdrojů podsvícení.

[17]



Obrázek 13: PWM regulace [18]

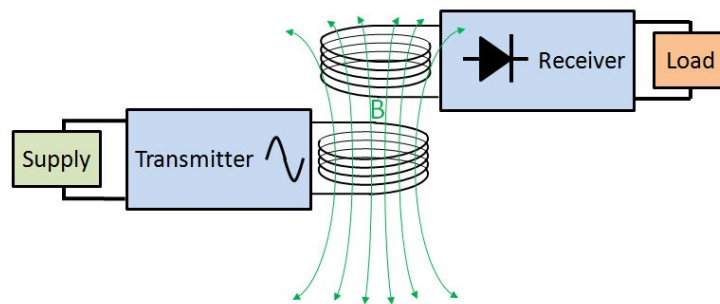


### 3.3 Indukční nabíjení

S bezdrátovým nabíjením experimentoval americký vynálezce Nikola Tesla již na konci 19. století. Avšak k jeho masivnímu rozšíření u komerčních produktů došlo až relativně nedávno. Vedle elektromagnetické indukce (která je nejběžnější u mobilních telefonů a menších chytrých zařízení) existuje ještě bezdrátový přenos založený na principu rádiových vln a rezonanční nabíjení. [19]

Přenos elektrického náboje je zajištěn dvěma cívkami. V nabíječe se nachází vysílací cívka a v nabíjeném zařízení přijímací cívka. Vysílací cívka vytváří pulzující elektromagnetické pole. Druhá cívka jej přijímá a mění pomocí elektromagnetické indukce na proud. [20]

Nabíjecí část je napájena stejnosměrným proudem ze zdroje. Ten je následně pomocí jednoduchého elektronického obvodu konvertován na střídavý proud, který je přiveden do vysílací cívky. Kolem ní vzniká kmitající elektromagnetické pole. Když se v dosahu elektromagnetického pole nachází přijímací cívka, indukuje se na ní střídavý proud, který je opět pomocí elektronických obvodů usměrněn tak, aby z něj mohlo být napájeno zařízení samotné (respektive jeho akumulátor). [20]



Obrázek 14: Princip indukčního nabíjení [21]

Dochází tedy nejméně ke třem konverzím z čehož logicky plyne, že dochází ke zdatelným ztrátám.

S rostoucí vzdáleností mezi cívkami efektivita nabíjení výrazně klesá. [20]

Kromě výše zmíněné neefektivitě patří mezi nevýhody bezkontaktního nabíjení větší komplikovanost a z toho plynoucí vyšší cena a menší spolehlivost.

Hlavní výhoda je zjevná, není nutný přímý kontakt mezi nabíjecí a nabíjenou částí. Dá se tedy nabíjet „na dálku“ a bez potřeby přesného zaměření na nabíjecí konektor. Z této vlastnosti logicky plyne i to, že nedochází k žádnému opotřebení konektorů.

### 3.4 Vývojové platformy

Mozkem každého robota je řídicí jednotka, která dokáže na základě vstupů vytvářet určité výstupy. Díky ní je možné robota kompletně ovládat a naprogramovat, jak se má ve které situaci zachovat. Výběr vhodné vývojové platformy, která bude roli řídicí jednotky zastávat, je tedy naprosto klíčový. Mezi vývojové platformy patří různé mikrokontrolery a mikropočítače.

**Mikrokontroler** přímo vykonává program, který na něj byl nahrán pomocí programátoru.

**Jednodeskový počítač** je plnohodnotný malý počítač s jednou deskou plošných spojů, na kterém běží nějaký operační systém.

#### 3.4.1 Arduino

Arduino je otevřená platforma skládající se z desky založené na mikrokontroleru od firmy Atmel a vývojového prostředí, které slouží k tvorbě softwaru ve vlastním programovacím jazyce vycházejícím z jazyka Wiring (framework jazyka C++). [22]

Platforma vznikla pro rychlé a snadné vytváření prototypů a je určena zejména pro studenty začínající s programováním a elektrotechnikou. Brzy se začala těšit velké oblibě. [22]

Mezi výhody Arduina patří jednoduchost, nízká cena a dobré možnosti komunikace s hardwarem (analogové vstupy, hardwarové PWM výstupy) a spolehlivost.

Největší slabinou je nižší výpočetní výkon společně s menší operační pamětí. Možnosti využití jsou zde více omezené než u plnohodnotných jednodeskových počítačů. Nemáme například takové možnosti, co se týče webových aplikací. Také jsme omezeni pouze na jeden programovací jazyk.

Je tak vhodné spíše pro jednoúčelová zařízení.

### 3.4.2 Raspberry Pi

Raspberry Pi je typickým představitelem malých jednodeskových počítačů. Nejvýkonnější modely odpovídají velikosti kreditní karty. Jeho základem je mikroprocesor architektury ARM. Po připojení patřičných periférií jej můžeme provozovat jako klasický stolní počítač. [23]

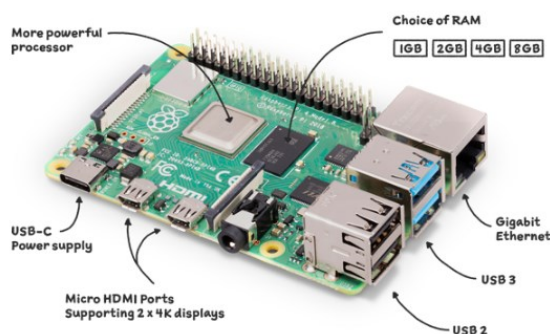
Největší uplatnění však nalezne ve světě IoT (Internet of Things). K jeho vstupně/výstupním GPIO pinům je možné připojit různé senzory, spínače nebo celé přídatné moduly, pomocí kterých můžeme interagovat s okolním světem. Je však třeba mít na paměti, že vnitřní logika Raspberry Pi je 3,3V (některé periferie, zejména ty určené pro Arduino, používají 5V logiku). [24]

Jako operační systém nejčastěji slouží Raspbian (odvozený z Debianu), který byl pro Raspberry Pi speciálně vyvinut. Možné je však nainstalovat i jiné upravené verze Linuxu, případně i speciální verzi Windows. Operační systém i všechna ostatní data jsou uloženy na Micro SD kartě.

Tento mikropočítač byl vyvinut v roce 2012 britskou organizací Raspberry Pi Foundation jako nástroj pro výuku programování. Využití těchto zařízení je téměř neomezené, a jsou tak součástí nespočtu různých školních nebo domácích projektů. [23]

Oproti jednoduchým mikrokontrolerům je výrazně komplexnější a výkonnější. Nevýhodou je vyšší cena, spotřeba elektrické energie a horší stabilita.

Hlavním důvodem, proč z velkého množství nabízených mikropočítačů zvolit konkrétně Raspberry Pi, je především jeho rozšířenost. Díky obrovské komunitě uživatelů nalezneme na síti nepřeberné množství návodů, rad a tipů pro práci s ním, tvorbu prakticky čehokoli nebo řešení problémů. Rovněž nabídka kompatibilního příslušenství je velmi bohatá.



Obrázek 15: Raspberry Pi 4 [25]

### 3.4.3 Nvidia Jetson

V roce 2019 americká společnost NVIDIA, uvedla do prodeje jednodeskový počítač Jetson Nano. Stejně jako Raspberry Pi nabízí i tento mikropočítač 40 GPIO pinů a běží na něm operační systém na bázi Linuxu. V čem se oproti Raspberry Pi liší, je výrazně výkonnější GPU a další integrovaný hardware pro akceleraci strojového učení. Je tak vhodný zejména pro projekty využívající umělou inteligenci (například roboti s vyšší mírou autonomie). Nevýhodou je vyšší cena a menší komunita uživatelů. [26]



Obrázek 16: NVIDIA Jetson Nano [27]

### 3.4.4 Hardwarové akcelerátory

V případě potřeby lze mikropočítače rozšířit o externí hardwarové akcelerátory. Jedná se zpravidla o USB zařízení obsahující vlastní specializovanou výpočetní jednotku, která dokáže provádět operace spojené se strojovým učení a výrazně tak urychlit výpočet a odlehčit hlavnímu CPU mikropočítače. Tyto zařízení bývají velmi efektivní, dosahují vysokého výpočetního výkonu při malé spotřebě energie. Příkladem takových akcelerátorů jsou například Google Coral USB Accelerator nebo Intel Neural Compute Stick. [28]



Obrázek 17: USB akcelerátor Intel Neural Compute Stick [29]

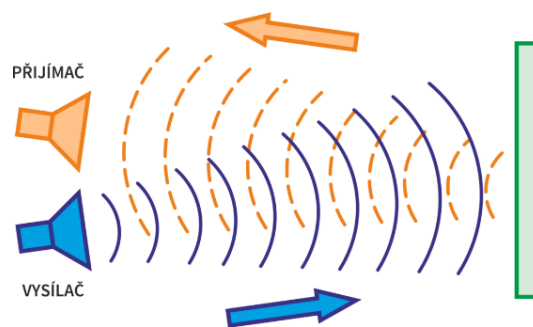
## 3.5 Detekce překážek

Senzory jsou pro robota velmi důležité, neboť díky nim může získávat informace o prostředí ve kterém se nachází. Dá se nadneseně říci, že senzory tvoří jeho smysly. Senzorický systém je tak klíčovou součástí každého autonomního robota. [2] Avšak i u robota řízeného uživatelem mohou být tyto informace velmi nápomocné.

Pro detekci překážek se využívají různé technologie. Ty nejběžnější využívají metody pracující se zvukem nebo optikou.

### 3.5.1 Akustická čidla

Sonar generuje akustický impulz, který se šíří prostorem rychlostí zvuku, a následně čeká na ozvěnu. Pokud zvukový impulz narazí na nějaký předmět, část zvukové vlny se odrazí zpět. Podle času, za který se zvuková vlna vrátí, lze poměrně přesně dopočítat vzdálenost dané překážky. Impulz bývá ultrazvukový (lidským uchem neslyšitelný). V robotice je tento druh senzorů velmi oblíbený, neboť je levný, poměrně jednoduše připojitelný a pro dané účely dostatečně přesný. [30]



Obrázek 18: Princip ultrazvukového senzoru [30]

Pro získání vzdálenosti od překážky lze použít následující vzorec:

$$\text{vzdálenost} = \text{čas} * \text{rychlost zvuku} / 2$$

Pokud čas budeme měřit v sekundách a rychlost zvuku dosadíme v metrech za sekundu (343 m/s), dostaneme výslednou hodnotu v metrech. Výsledek dělíme dvěma, jelikož naměřený čas představuje hodnotu, za kterou zvuk urazil vzdálenost ke překážce a zase zpátky.

### **3.5.2 Optická čidla**

Další možností detekce překážek je využít infračervené světlo (**IR=InfraRed**). Aktivní IR senzor funguje na principu přijímací a vysílací diody. Vysílačem je zde světelná dioda (IR LED) a jako detektor slouží infračervená fotodioda, která je citlivá na infračervené světlo stejné vlnové délky, které vyzařuje vysílač. Vysílané světlo se odráží od světlých ploch (objektů) a přijímač ho tak může zachytit. Tyto senzory lze využít nejen k detekci překážek, ale například i pro sledování čáry. [31]

## **3.6 Rozpoznávání objektů pomocí kamery**

Typickým úkolem počítačového vidění je identifikace konkrétních objektů v obraze. K tomuto účelu slouží různé algoritmy. Dnes již existuje celá řada metod, které dokáží dosáhnout poměrně vysoké úspěšnosti a přesnost rozpoznávání se neustále zlepšuje. Stále však záleží na konkrétních podmínkách a okolnosti jako horší osvětlení, přítomnost rušení v obraze nebo rychlý pohyb objektu mohou být problémem.

### **3.6.1 Detekce pomocí klíčových bodů**

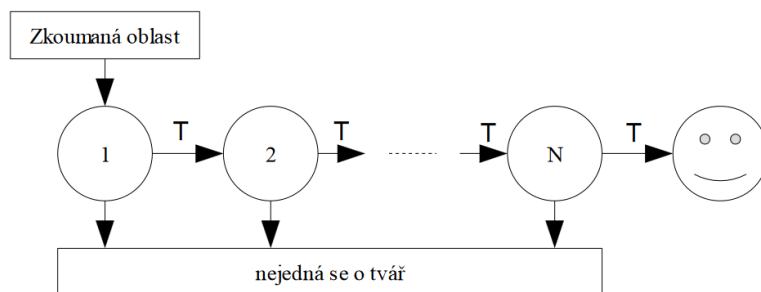
Detekce příznaků se snaží najít konkrétní body v obraze, které jsou dobře rozeznatelné a mají vlastnosti, které se vyskytují pouze u daného objektu. Například v případě detekce obličeje by těmito body byly oči, pusa, nos a tak dále. To, jak jsou tyto body nalezeny, se liší podle konkrétního algoritmu, nejlépe se dají detekovat pomocí hran. [32]

### **3.6.2 Klasifikátory**

Klasifikátor určuje, s jakou pravděpodobností se na daném místě nachází hledaný objekt. Klasifikátory obvykle posuzují skupinu příznaků, které slouží jako základní popis dané části obrazových dat. Při správném vyhodnocení těchto příznaků dostaneme odpověď na otázku, zda daný objekt patří do hledané skupiny. [33]

### 3.6.3 Kaskáda klasifikátorů

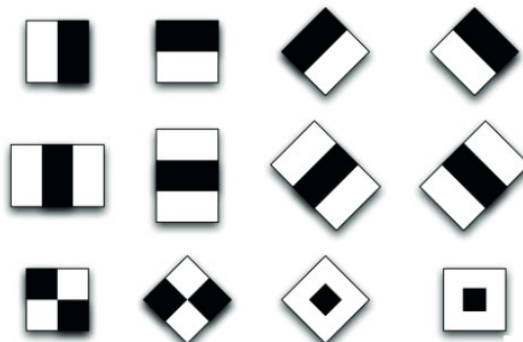
Kaskádové zapojení klasifikátorů se používá k zvýšení účinnosti a snížení chybovosti. Zkoumaný výřez obrazu prochází postupně přes všechny klasifikátory. Pokud je některým z nich označen jako nevyhovující, již neprochází na další stupeň kaskády. Pokud výřez úspěšně projde přes všechny klasifikátory, je vyhodnocen jako hledaný objekt. Jednotlivé klasifikátory nemusí dosahovat vysoké bezchybnosti, ale jako celek mají velmi dobrou úspěšnost. Výhodou tohoto přístupu je vyřazení oblastí, které neobsahují objekt již na začátku. Na začátku kaskády jsou z pravidla klasifikátory jednodušší a rychlejší než klasifikátory na konci, které mají již podstatně větší složitost. Takto je dosaženo vysoké efektivity. [33] [32]



Obrázek 19: Kaskáda klasifikátorů [33]

### 3.6.4 Haarovy příznaky

Příznaky obsahují informaci o charakteru vybrané oblasti obrazu. Pokud se tato informace dá vyjádřit jednoduchým způsobem, je vyhodnocení velmi rychlé. Haarovy příznaky jsou jednoduché konvoluční filtry.



Obrázek 20: Haarovy příznaky [32]

Jedná se o obdélníkové oblasti, které se postupně aplikují na obraz (velikost příznaků se postupně zvětšuje až na velikost celého obrazu). Pro výpočet Haarových příznaků je třeba obraz nejprve převést do černobílé podoby, neboť při výpočtu se pracuje pouze s intenzitami pixelů a nelze tedy využít barevnou informaci. Hodnota odezvy příznaku odpovídá intenzitě obrazu pod danou plochou. Výpočet se provádí tak, že od sumy bílých oblastí odečteme sumu černých oblastí. Haarovy příznaky jsou založeny na sumách obdélníkových oblastí. Tento výpočet můžeme vykonávat prostým sčítáním, které je citlivé na velikost sčítané oblasti, nebo lze použít tzv. integrální obraz. (který výpočet výrazně urychlí). [32] [33]

### **3.6.5 Využití umělé inteligence a neuronových sítí**

Další možností je pro rozpoznávání objektů v obraze využít neuronových sítí a pokročilé umělé inteligence. V poslední době se tento přístup stává stále více populárním, jelikož přináší do rozpoznávání objektů v obraze určité výhody oproti tradičním algoritmům.

Tradiční algoritmy zmíněné v předchozích kapitolách pracují s ručně navrženými funkcemi, které se snaží co nejlépe popsat daný problém. Tyto algoritmy tedy mohou být poměrně úspěšné, pokud jsou použity na správné problémy a mají dostatek trénovacích dat. Avšak ručně navrhované funkce mohou být omezené svou schopností popsat velkou škálu různorodých datových sad, což může vést k horší úspěšnosti při rozpoznávání nových objektů. [34] [35]

Na druhé straně umělá inteligence a neuronové sítě se snaží naučit se popsat problém automaticky pomocí neuronových sítí. Tento proces trénování využívá velké množství dat a učí se vytvářet vlastní funkce, které co nejlépe popisují daný problém. Tento přístup může být velmi efektivní při rozpoznávání nových objektů, protože neuronové sítě jsou schopny vytvářet velmi složité a robustní funkce, které dokážou popsat velkou škálu různých datových sad. [35] [36]

Výhodou neuronových sítí oproti tradičním algoritmům je tedy větší schopnost adaptace na nové problémy, schopnost vytvářet vlastní funkce pro popis problému a schopnost pracovat s velkými a různorodými datovými sadami. Avšak výpočetně jsou neuronové sítě náročnější a vyžadují větší množství dat pro trénování. [35]



## 3.7 Webová komunikace

### 3.7.1 Komunikační protokoly

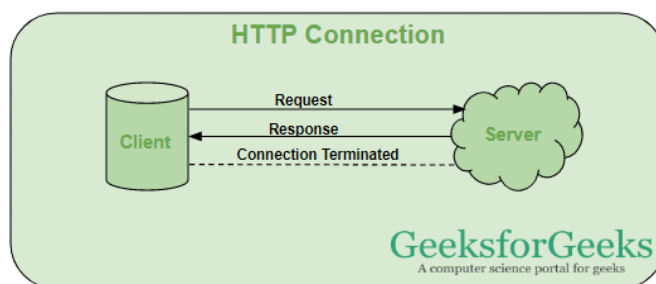
Komunikační protokol je soubor pravidel, který definuje, jakým způsobem bude docházet k výměně informací mezi oběma komunikujícími stranami. Popisuje syntaxi, způsob přenosu dat a synchronizaci vzájemné komunikace.

Typickým příkladem webové komunikace je model **klient-server**, kde klient odesílá požadavek(**request**) a server odpovídá zprávou nebo službou (**response**).

HTTP i WebSocket se řadí mezi protokoly aplikační vrstvy, které zprostředkovávají komunikaci mezi klientem a serverem.

### 3.7.2 HTTP

HTTP (Hypertext Transfer Protocol) je internetový protokol, který se již od devadesátých let využívá ke komunikaci s WWW servery. Jedná se o **bezstavový protokol**. Komunikace funguje tak, že klient otevře spojení a odešle požadavek na server, který odpoví a spojení se opět uzavře. Klient a server o sobě vědí jen během konkrétního požadavku. [37] [38]



Obrázek 21: HTTP komunikace [39]

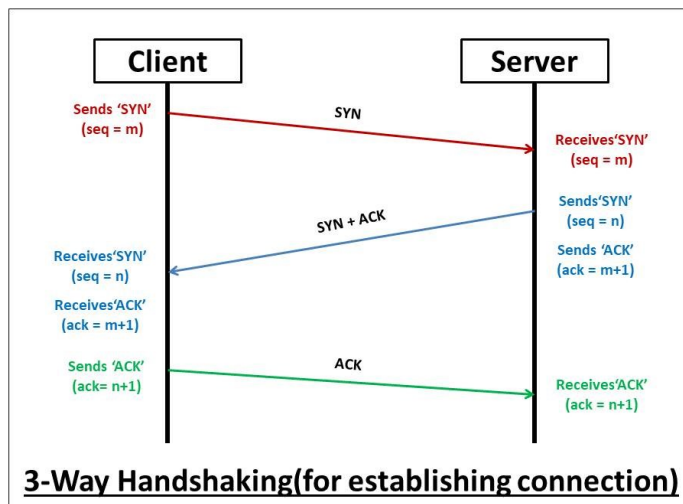
S každým dalším požadavkem klienta, musí být spojení navázáno znovu a celý tento proces se opakuje. Další možnou nevýhodou této architektury je fakt, že aktivita musí vždy vycházet ze strany klienta, server sám od sebe nemůže nic odesílat. Komunikace probíhá v **half-duplex** módu, což znamená, že v jeden moment může vysílat vždy pouze jedna ze stran (není možná obousměrná komunikace současně). [37] [40]

Výhodou tohoto protokolu je jednoduchost, nízká zátěž serveru a vysoká spolehlivost. Jedná se tak o ideální volbu pro webové aplikace pracující s daty, které se nemění tak často a samozřejmě statické webové stránky. [40]

### 3.7.3 TCP 3-Way Handshake

Pro navázání spojení se využívá třícestný handshake (potřesení rukou) ve kterém si klient se serverem vymění tři zprávy. Tento proces je zajištěn pomocí TCP protokolu. [37] [41]

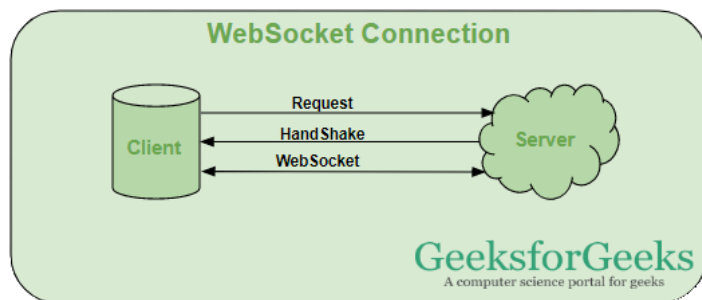
- Klient pošle **SYN** (synchronize) paket s číslem sekvence
- Server přijme číslo sekvence a prostřednictvím **SYN-ACK** (synchronize-acknowledgment) paketu potvrdí přijetí
- Klient odpoví **ACK** (acknowledgment) paketem a komunikace je navázána



Obrázek 22: TCP Handshake [42]

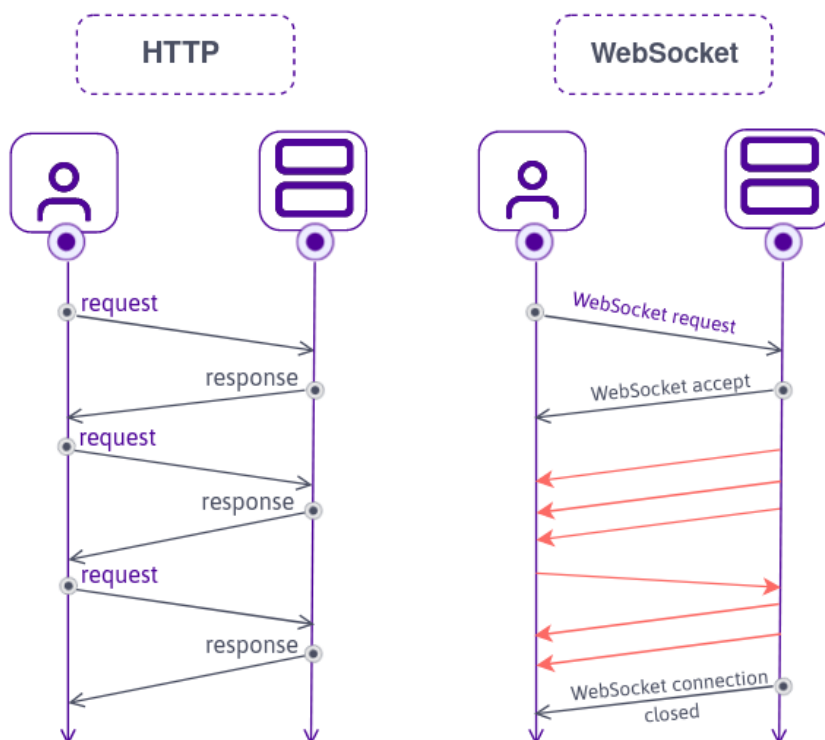
### 3.7.4 WebSocket

WebSocket byl vyvinut v roce 2008 W3C (World Wide Web Consortium) tak, aby umožnil obousměrnou a plynulou komunikaci mezi klientem a serverem. Jedná se o stavový protokol, takže spojení mezi klientem a serverem trvá, dokud není jednou ze stran uzavřeno. Úvodní handshake proběhne pouze na začátek. Poté, co je komunikace navázána, mohou jak klient, tak server libovolně odesílat zprávy. WebSocket tedy můžeme považovat za jakýsi obousměrný komunikační kanál. Přenos dat je zajištěn pomocí protokolu TCP. Jakmile se jedna ze stran rozhodne komunikaci ukončit, spojení je uzavřeno pro obě strany. [37] [43]



Obrázek 23: WebSocket komunikace [39]

Tento protokol nalezne uplatnění zejména u real-time aplikací, her, chatovacích aplikací a všude tam, kde je zapotřebí pružná odezva. Jelikož se ale jedná o složitější a náročnější proces, není vhodné jej využívat tam, kde to není nutné. [39]



Obrázek 24: Srovnání HTTP a WebSocketu [43]

### 3.8 WEB RTC

WebRTC (Web Real-Time Communication) je open source technologie, která umožňuje komunikaci v reálném čase prostřednictvím webových aplikací. Tato technologie podporuje obousměrný přenos audiovizuálních dat a také umožňuje sdílení souborů mezi uživateli. Lze ji tedy využít pro videohovory, konference, vytváření aplikací pro vzdálenou správu a mnoho dalšího. [44][45]

WebRTC se skládá z několika hlavních komponent, které umožňují komunikaci v reálném čase. Tyto komponenty zahrnují SDP (Session Description Protocol), ICE (Interactive Connectivity Establishment), STUN (Session Traversal Utilities for NAT) a TURN (Traversal Using Relay NAT). [46]

WebRTC je založena na klasických webových standardech, což umožňuje její širokou kompatibilitu s moderními webovými prohlížeči bez potřeby instalace zásuvných modulů. [45]

**WebRTC API** jsou součástí standardu webových API a jsou dostupné v každém moderním webovém prohlížeči. Tyto API umožňují využívat funkcionality WebRTC a integrovat je do vlastních webových aplikací pomocí JavaScriptu. Samotná API jsou napsána v čistém JavaScriptu a mohou být použita v libovolné webové aplikaci bez potřeby instalace zvláštních knihoven nebo rozšíření. [46]

Mezi hlavní API patří **MediaStream API**, **RTCPeerConnection API** a **RTCDataChannel API**.

**MediaStream API**, umožňuje získání přístupu k audiovizuálním datům z kamery a mikrofonu uživatele a následné zobrazení těchto dat na stránce webové aplikace.

**RTCPeerConnection API** je zodpovědné za vytvoření peer-to-peer spojení mezi dvěma prohlížeči, což umožňuje obousměrnou komunikaci mezi nimi.

**RTCDataChannel API** poskytuje přístup k datovému kanálu, který umožňuje posílat a přijímat data mezi prohlížeči bez nutnosti využití serveru jako prostředníka.

[47]

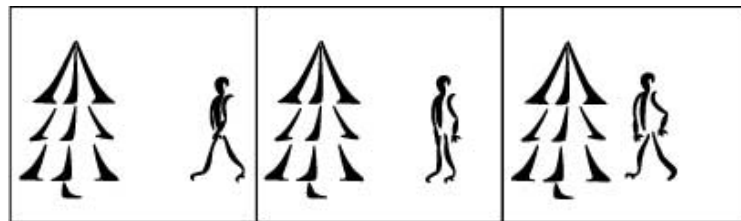


Obrázek 25: WebRTC [44]

## 3.9 Formáty videa

### 3.9.1 MJPEG

MJPEG (Motion JPEG) je formát, který komprimuje každý snímek zvlášť podle standardu JPEG. Nejedná se tak o klasický video formát, ale o sekvenci plnohodnotných jednotlivých JPEG snímků. Mezi jeho výhody patří dobrá obrazová kvalita a robustnost (výpadek snímku neovlivní zbytek videa). Jedná se o velmi jednoduchý formát, není výpočetně náročný (není třeba složitě dopočítávat rozdílová data) a podporují ho všechny internetové prohlížeče. Jeho zásadní nevýhodou je velmi vysoká datová náročnost. Ze samotné podstaty fungování součástí tohoto formátu nemůže být zvuk. [48] [49]

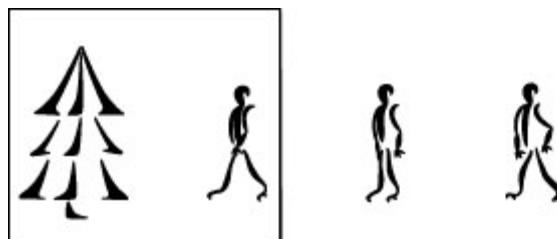


Obrázek 26: MJPEG sekvence [49]

### 3.9.2 H.264 a H.265

H.264 (nebo také MPEG-4 AVC) je klasický formát videa, který funguje na principu, že kompletní snímky se odesílají s určitým časovým rozestupem a mezi nimi jsou přenášena jen rozdílová data. Při stejné kvalitě má tak video znatelně menší datový tok (respektive i při nízkém datovém toku můžeme dosáhnout relativně kvalitního a plynulého obrazu). [48] [49]

H.265 je nástupcem standardu H.264, který dosahuje výrazně lepší efektivity komprese (výsledný datový tok může být menší až o polovinu). [50]



Obrázek 27: MPEG přenos [49]

## 4 Možnosti vzdáleného přístupu

Tato kapitola pojednává o různých možnostech, jak se ke konkrétnímu zařízení v lokální síti (jako je náš robot) připojit odkudkoliv z internetu.

### 4.1 Přímý přístup přes veřejnou IP adresu

V případě, že nám poskytovatel internetového připojení poskytl veřejnou IP adresu, je možné se k serveru připojit přes ni. Jedná se o nejpřímější cestu, která s sebou ovšem přináší značná bezpečnostní rizika. Na tuto adresu se může připojit kdokoli z internetu, tudíž je náš server velmi snadno napadnutelný a je nutné, aby byl velmi dobře zabezpečen.

#### 4.1.1 Přesměrování portu

Aby mohl vnější uživatel přistupovat k serveru uvnitř privátní sítě, je třeba na routeru správně nastavit přesměrování portu (**port forwarding**). Tento přístup spočívá v přesměrování požadavků z vnější sítě na specifický port, který odpovídá dané službě uvnitř privátní sítě. Tato funkce je realizována díky technologii překladu síťových adres (NAT), která zajišťuje překlad IP adres a portů mezi privátní a veřejnou sítí. Router tak může podle aplikačního portu směřovat příchozí síťové požadavky na konkrétní IP adresu zařízení ve vnitřní síti. [51]

#### 4.1.2 DDNS

V případě, že naše veřejná IP adresa není statická, ale je přidělována poskytovatelem internetu dynamicky (například pomocí DHCP protokolu) je možné použít některou ze služeb DDNS (Dynamic Domain Name System), která umožní přístup k serveru prostřednictvím názvu domény namísto IP adresy. Mezi nejznámější služby, které poskytují bezplatné subdomény (a zároveň zajistí automatickou aktualizaci IP adresy) patří například DuckDNS nebo No-IP. [52] [53]

## **4.2 VPN**

Pomocí VPN (Virtual Private Network) je možné připojit se do své privátní sítě. Poté již lze přistupovat k serveru stejně, jako bychom se nacházeli fyzicky v naší domácí síti. Využití VPN je z hlediska bezpečnosti jedna z nejlepších variant. VPN vytváří šifrovaný tunel mezi naší sítí a připojeným bodem. Veškerá komunikace je přenášena prostřednictvím tohoto tunelu, díky tomu je chráněna před neoprávněným přístupem a je tak zaručeno soukromí. [53]

## **4.3 Využití služeb třetích stran**

V případě, že veřejnou IP adresou nedisponujeme, je možné využít některou ze služeb třetích stran. Tyto služby nám umožní připojit se k serveru umístěnému v jiné síti, který zprostředkuje spojení s naší sítí.

## 5 Metodika

Práce se skládá z několika základních částí. První částí je hardwarová implementace robota. Následuje vývoj řídicí aplikace, zajištění audiovizuálního přenosu a tvorba uživatelského webového rozhraní.

### 5.1 Hardwarová implementace

Základnu robota tvoří podvozek se dvěma pásy, které jsou poháněny stejnosměrnými 9V motory s vestavěnou převodovkou. K řízení směru jízdy u pásového podvozku slouží diference otáček. Je tedy možné robota otočit na místě protichůdným pohybem pásů, případně jen mírněji zatočit krátkým zastavením jednoho z pásů během jízdy. Mozkem robota je mikropočítač Raspberry Pi verze 4. Aby s ním bylo možné řídit motory, je na něj nasazena řídicí deska se dvěma H-můstky, která se rovněž stará o napájení samotného Raspberry Pi z akumulátoru. Rychlost otáčení pásů je řízena pomocí PWM. Ke snímání obrazu slouží kamera připojená přes CSI port a je možné s ní otáčet v obou osách díky dvěma modelářským servomotorům. Audio výstup představují malé reproduktory připojené přes Jack konektor k interní zvukové kartě a jako vstup slouží USB mikrofon. Součástí sensorického systému robota je rovněž ultrazvukový dálkoměr. Celá konstrukce drží pohromadě díky použití šroubů a několika dílů ze stavebnice Merkur. O přísun elektrické energie se stará akumulátor o kapacitě 2000 mAh, který je nabíjený bezdrátově pomocí elektromagnetické indukce. Stav akumulátoru je hlídán pomocí voltmetru, který s Raspberry Pi komunikuje pomocí sběrnice I2C.

### 5.2 Řídicí aplikace

Aplikace pro ovládání robota byla napsána v jazyce Python a je rozdělena do několika souborů (jednotlivé třídy). Povelů pro řízení pohybu robota a rovněž stavové informace jsou přenášeny prostřednictvím obousměrného WebSocket kanálu, který zajišťuje knihovna Tornado.



### **5.3 Audiovizuální přenos**

O streamování videa k uživateli a obousměrný přenos audia pomocí protokolu WebRTC se stará UV4L server, který rovněž funguje jako HTTP server a odesílá uživateli samotnou ovládací HTML stránku.

### **5.4 Webové rozhraní**

K ovládání robota slouží jednoduchá webová aplikace, kde lze pomocí tlačítek na ovládacím panelu spustit nebo zastavit streamování videa a zapnout/vypnout zvuk pro oba jednotlivé směry. Dále zde uživatel může sledovat různé stavové informace, jako je síla signálu wifi, vzdálenost od překážky před robotem nebo stav nabití baterie. Pohyb robota je možné ovládat pomocí kláves nebo tlačítek na obrazovce. Uživatel také může nastavit rychlost otáčení pásů (v procentech) pomocí jednoduchého posuvníku.

## 6 Použitý hardware

### 6.1 Řídicí jednotka

Jako řídicí jednotka robota bylo zvoleno **Raspberry Pi 4**. Použita byla levnější varianta s operační pamětí o kapacitě 2GB, která by měla být pro dané využití dostačující.

Vzhledem k potřebě provozu webových serverů a přenosu videa bylo vhodnější zvolit výkonnější mikropočítač. Důvodem proč konkrétně Raspberry Pi je především zmiňovaná obrovská komunita uživatelů, kteří mohou být nápomocni při stavbě robota i přidávání dalších funkcionalit.

Nejnovější verze 4 nabízí dobrý výpočetní výkon i konektivitu. Její nevýhodou je vyšší spotřeba elektrické energie a vyšší cena.

#### Specifikace:

CPU	Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
RAM	1GB / 2GB / 4GB / 8GB
Úložiště	Micro-SD
Síť	2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE Gigabit Ethernet
GPIO	40 pinů (zpětně kompatibilní)
Konektivita	2 USB 3.0; 2 USB 2.0 2 × micro-HDMI 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port
Napájení	5V DC, 3A přes USB-C nebo GPIO

Tabulka 1: Specifikace RPi 4 [25]

Jako paměťové médium slouží MicroSDHC karta o kapacitě 32GB. Konkrétně byla vybrána karta od společnosti SanDisk ve verzi Extreme, která patří mezi nejlépe hodnocené karty pro použití s Raspberry Pi. [54]



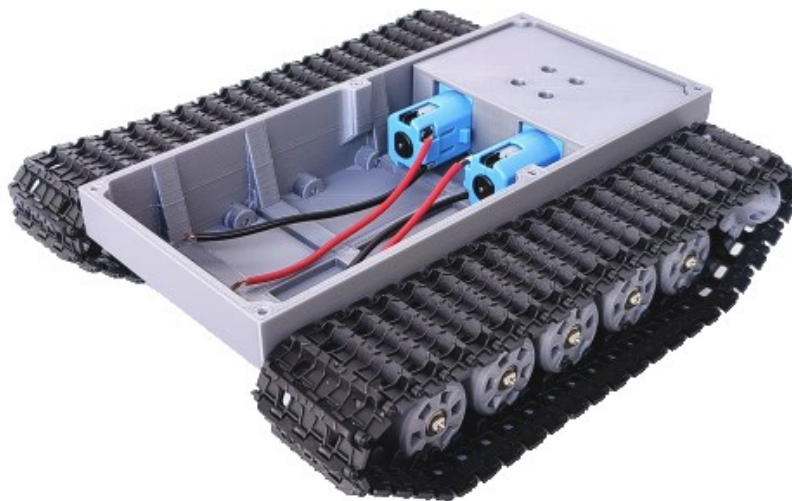
Obrázek 28: SanDisk Extreme SD [54]

Dále byl na procesor nalepen pasivní hliníkový chladič pro lepší odvod tepla.

## 6.2 Podvozek s motory

Pro daný účel byl jako nejvhodnější varianta zvolen pásový podvozek. Pásky jsou nejsnáze ovladatelné, robota lze otočit i na místě a po zastavení se už nikam neposune kvůli setrvačnosti (jako tomu může být v případě kol).

Rozměry podvozku jsou 20\*17\*5cm (délka, šířka, výška) a hmotnost činí 460g. Většina jeho součástí byla vytištěna na 3D tiskárně. [55]

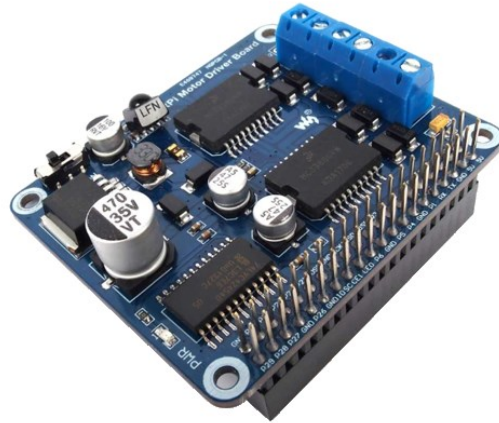


Obrázek 29: Použitý podvozek s motory [55]

O pohon se starají dva stejnosměrné 9V TT motory s vestavěnou převodovkou. Záměrně byl vybrán model s co nejpomalejším převodem, tak aby robota bylo možné plynule ovládat.

## 6.3 Ovladač motorů

Pro ovládání motorů slouží hotová rozšiřující deska od společnosti Waveshare, která obsahuje dva výkonné H-můstky **Freescale MC33886**. Tyto H-můstky dokáží pracovat se vstupním napětím až 40V a provozním proudem až 5A. [56] [57]



Obrázek 30: Waveshare RPi Motor Driver Board [57]

Kromě řízení motorů se deska postará rovněž o napájení samotného Raspberry Pi z akumulátoru. K tomu slouží **LM2596** regulátor, který sníží napětí z akumulátoru na 5V.

## 6.4 Kamera

O snímání obrazu se stará 5Mpx kamera s čipem **OV5647**, kterou lze přes CSI port zapojit přímo do Raspberry Pi.

Výhodou této kamery je, že díky IR filtru a dvojici pomocných IR reflektorů dokáže zaznamenávat obraz i v noci. Zatímco během dne kamera zaznamenává obraz normálně, při nízké úrovni okolního osvětlení díky fotorezistorům sepnou IR reflektory a kamera začne obraz snímat jen černobíle. Kamera byla testována se dvěma objektivy 2.8mm a



Obrázek 31: Kamera s nočním viděním [57]

3.8mm, přičemž kratší ohnisko se ukázalo jako vhodnější (díky širšímu záběru se lépe orientuje v prostoru). Kamera nemá automatické ostření, ale objektiv je možné manuálně přestříit na jinou vzdálenost. [57] [58]

## 6.5 Otáčení kamery

Pro otáčení kamery byla zakoupena jednoduchá plastová základna, do které lze zasadit dvě malá serva pro otáčení v horizontální i vertikální ose. [59]



Obrázek 32: Otočná základna pro kameru [59]

Použity byly levné a velmi oblíbené servomotory **micro servo SG90 9g** ve verzi, která umožňuje otáčení v rozsahu 180°. Tato serva pracují při provozním napětí v rozsahu 3.0-7.2V, lze je tedy napájet přímo z Raspberry Pi. [60]



Obrázek 33: Servomotory [60]

## 6.6 Detektor překážek

Pro detekování překážek slouží ultrazvukový dálkoměr **HC-SR04-P**. Důležité je, že novější verze s „P“ na konci pracuje s napětím **3,3V**.



Obrázek 34: Ultrazvukový dálkoměr HC-SR04 [61]

Lze jej tedy zapojit přímo do Raspberry Pi. Klasická verze (určená zejména pro Arduino) používá 5V. Aby ji bylo možné použít, bylo by nutné použít napěťový dělič. Tento dálkoměr je velmi oblíbený a cenově dostupný. Nevyniká sice dokonalou přesností, jelikož nám jde ale pouze o detekci překážek (nikoli přesné měření vzdálenosti) poslouží naprosto dostatečně. [57] [61]

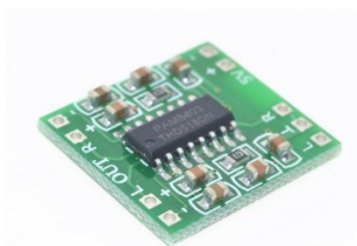
## 6.7 Audio

Jako zvukový výstup slouží malé reproduktory o výkonu 2W s impedancí 8Ω a s rozměry 18x53x15mm. Hlasitost ani kvalita akustické reprodukce nejsou příliš vysoké, nicméně jsou dostatečné pro účely mluveného slova. Hlavními přednostmi těchto reproduktorů jsou nízká cena, kompaktní rozměry a možnost snadného uchycení. [62]



Obrázek 35: Reproduktory [62]

K zesílení signálu pro tyto reproduktory slouží **2x3W digitální audio zesilovač PAM8403**. [63]



Obrázek 36: Audio zesilovač PAM8403 [63]

Pro zvukový vstup byl vybrán **USB mikrofon**, který funguje na principu plug and play a není tedy nutné u většiny linuxových distribucí (včetně Raspberry Pi OS) doinstalovávat žádné další ovladače. [64]



Obrázek 37: USB Mikrofon [64]

## 6.8 Mechanická konstrukce

Jako základní konstrukce pro uchycení většiny komponent k podvozku slouží stavebnice Merkur, u které se dá očekávat, že mnoho potencionálních uživatelů už ji bude vlastnit, případně se dá relativně levně pořídit. Tato stavebnice splňuje požadavek na modulárnost (každý ji může různě rozšířit podle svých konkrétních potřeb).

V našem konkrétním případě byly použity díly ze stavebnice MERKUR 3 BIG SET. [65]



Obrázek 38: stavebnice Merkur 3 BIG SET [65]

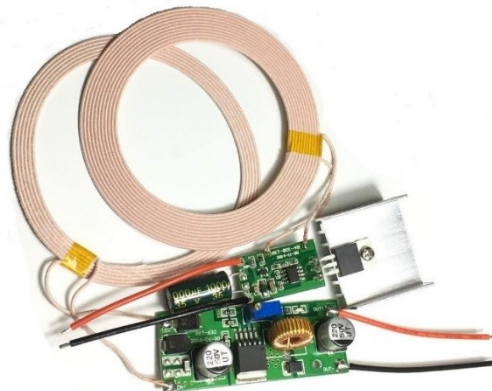
## 6.9 Napájení

O přísun elektrické energie se stará akumulátor **18650 battery pack** s napětím 11.1V a o kapacitě 2000 mAh. [66]



Obrázek 39: Akumulátor [66]

Nabíjení zajišťuje indukční nabíječka výrobce Taidacent, která jako jedna z mála dokáže nabíjet napětím 12V (většina prodávaných nabíječek je určena pro mobilní telefony, které jsou nabíjeny pomocí 5V). Do vzdálenosti 9mm dosahuje výstupní proud 2A (se vzrůstající vzdáleností tato hodnota výrazně klesá při 10mm činí 1.9A a při 18mm již pouhých 800mA). [67]

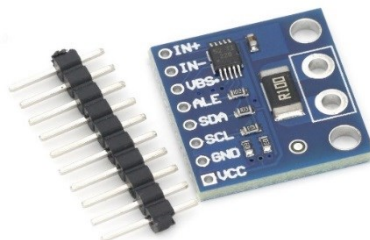


Obrázek 40: Indukční nabíječka Taidacent [67]

Nabíječka byla umístěna do plastové krabičky o výšce 2cm a šířce 8,7cm, do které byly vyvrtány otvory pro lepší odvod tepla. Pro správné navedení robota k nabíjecí cívce byly vytvořeny okolo krabičky nájezdy z kartonu, fotografie nabíjecí stanice je k nalezení v přílohách.

## 6.10 Indikátor stavu baterie

Jako indikátor stavu baterie slouží malý a cenově dostupný modul s voltmetrem **INA226**. Tento modul pracuje s provozním napětím v rozsahu 2.7-5.5V a dokáže měřit napětí v rozsahu 0-32V. S mikropočítačem komunikuje pomocí sběrnice I2C. [68]



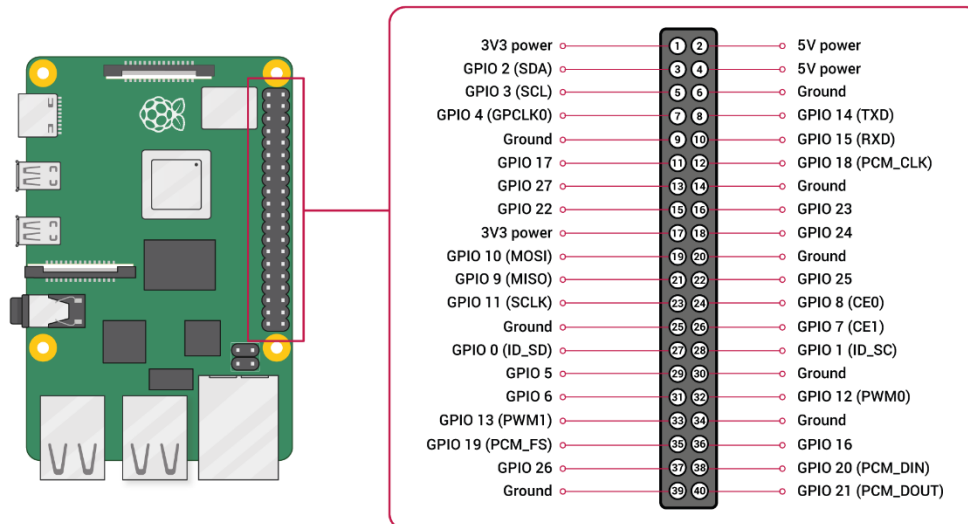
Obrázek 41: Modul pro měření napětí INA226 [68]



## 7 Hardwarová implementace

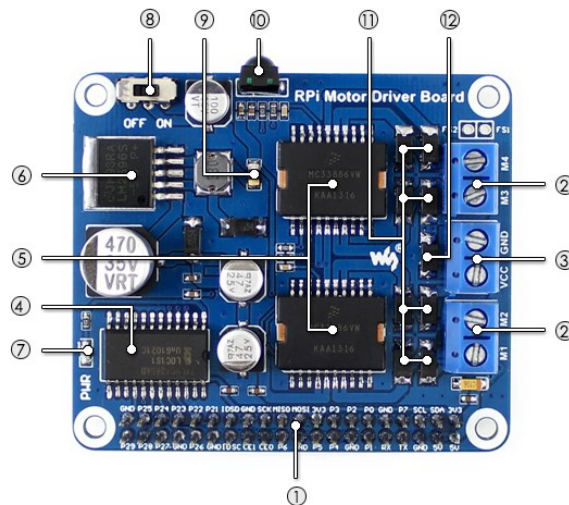
### 7.1 Číslování pinů Raspberry Pi

Pro GPIO piny Raspberry Pi se používají dvě možnosti číslování. První možnost se nazývá **BOARD** a představuje fyzické pořadí pinů na desce (čísla psaná uvnitř). Druhá možnost **BCM** (čísla z vnějšku) je běžně používána při psaní aplikací. **Kdekoli v této práci bude zmíněno číslo GPIO pinu, bude myšleno dle číslování BCM.**



Obrázek 42: Číslování GPIO pinů Raspberry Pi [64]

## 7.2 Schéma ovladače motorů



Obrázek 43: Waveshare RPi Motor Driver Board [56]

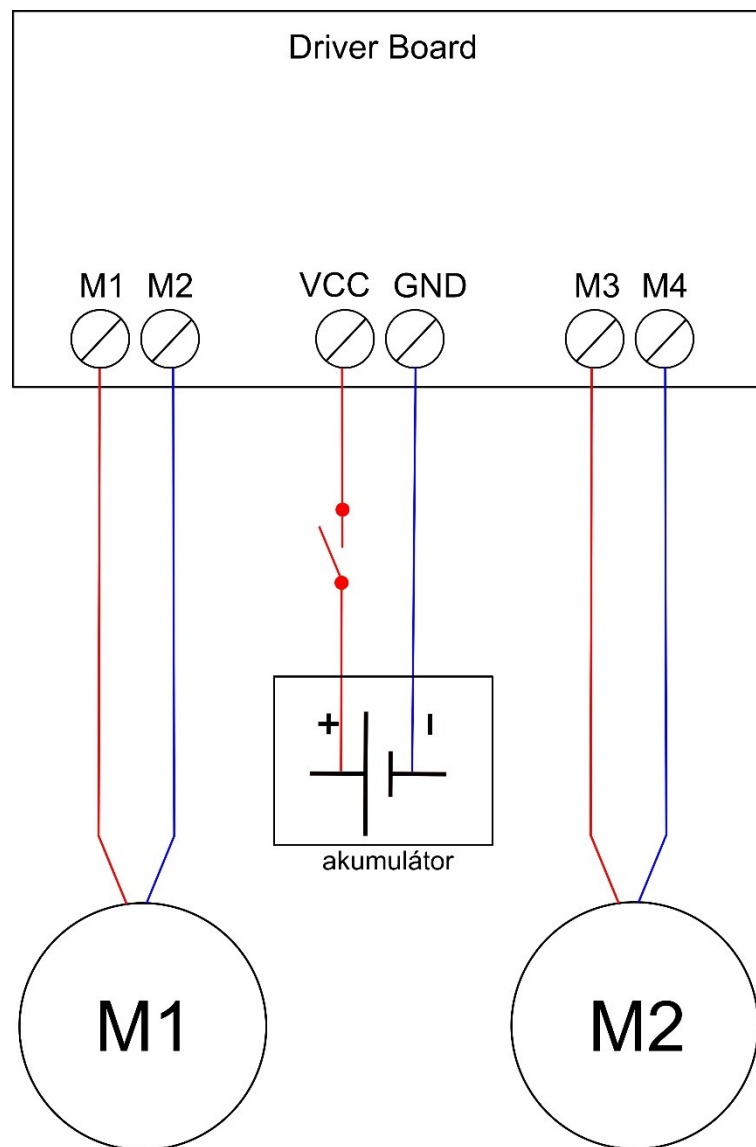
1. Raspberry Pi GPIO interface: pro připojení k Raspberry Pi
2. Šroubová svorkovnice: pro připojení motorů
3. Svorkovnice pro připojení externího zdroje energie (akumulátoru)
4. 74LVC4245AD: čip pro převod úrovně napětí (buffer)
5. MC33886: ovladače motoru (H-můstky)
6. LM2596-5.0: napěťový měnič na 5V (pro napájení Raspberry Pi z akumulátoru)
7. Indikátor zapnutí (power led)
8. Přepínač napájení Raspberry Pi
  - OFF: Raspberry Pi napájí Driver Board
  - ON: Driver Board napájí Raspberry Pi
9. 2A pojistka
10. IR přijímač
11. Schottkyho diody: ochrana motorů
12. Antireverzní dioda napájení

Ovladače motoru obsadí piny **6, 12, 13, 20, 21, 26**. Do pinů odpovídajícím těmto číslům není tedy již možné dále cokoli zapojovat. Ostatní piny jsou prodlouženy skrz desku a lze je využít úplně stejně jako bychom periferie připojovali napřímo do Raspberry Pi.

Piny 12 a 26 slouží k aktivaci motorů (případně nastavení rychlosti pomocí PWM). Piny 20 a 21 nastavují směr otáčení u prvního motoru a piny 6 a 13 u motoru druhého.

Pro provoz robota z akumulátoru je nutné nastavit přepínač do polohy ON, tak aby Raspberry Pi bylo napájeno skrze GPIO porty z Driver Boardu. V poloze OFF napájí naopak desku Raspberry Pi z vlastního zdroje (USB), což se může hodit při vývoji, kde nemusíme zbytečně vybíjet baterii.

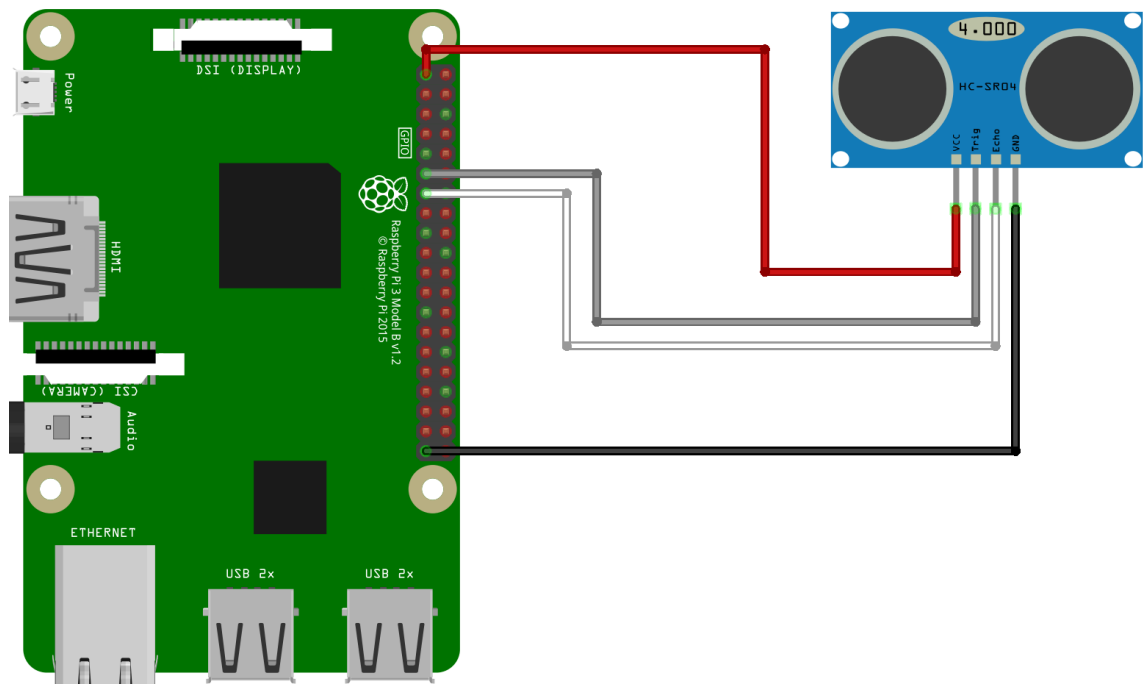
### 7.3 Zapojení motorů



Obrázek 44: Schéma zapojení motorů

Motory a akumulátor jsou zapojeny do svorkovnice rozšiřující desky dle schématu výše. Aby bylo možné celé zařízení vypnout, byl mezi akumulátor a napájecí svorku vřazen mechanický vypínač.

## 7.4 Zapojení dálkoměru



fritzing

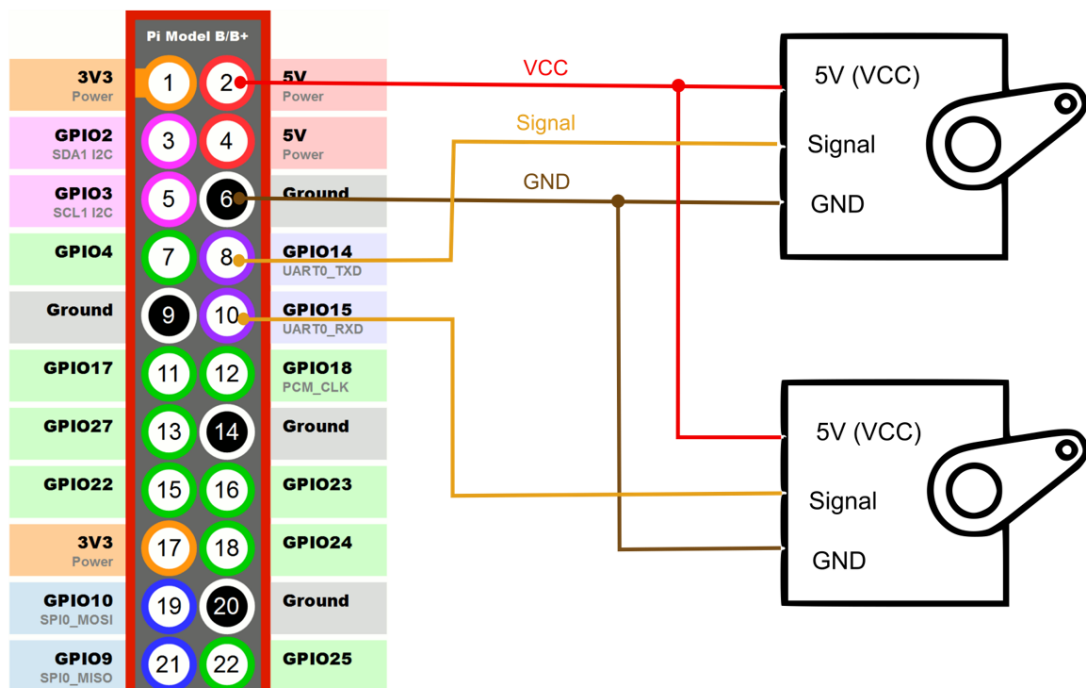
Obrázek 45: Zapojení dálkoměru [61]

HCSR04P	RASPBERRY PI
VCC	3V3
GND	GND
Trigger	GPIO 17
Echo	GPIO 27

Tabulka 2: Zapojení ultrazvuku

Jelikož se jedná o novější verzi ultrazvuku s označením HCSR04-P, je možné zapojit napájení přímo na 3V3 pin, který se na Raspberry Pi nachází v levém horním rohu (pin 1 podle číslování Board). GND (ground) lze zapojit na libovolný GND pin. Piny Trigger a Echo lze zapojit na kterékoli neobsazené GPIO piny.

## 7.5 Zapojení servomotorů



Obrázek 46: Zapojení servomotorů

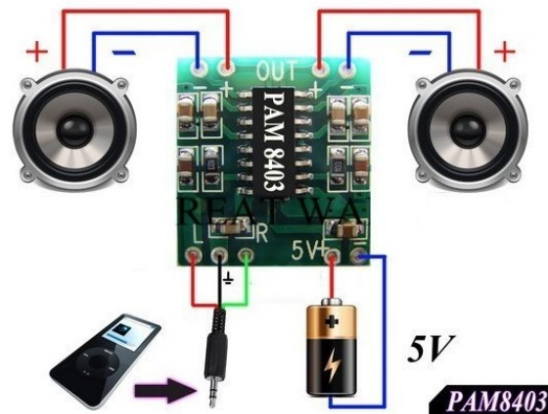
Červený drát značí napájení 5V (protože Raspberry Pi disponuje pouze dvěma 5V napájecími výstupy a další bylo nutné použít pro napájení audio zesilovače, byl napájecí drát rozdělen). Hnědý drát značí GND (Ground). První servo je skrze PWM ovládáno pinem 14 a druhé servo pinem 15.

## 7.6 Zapojení kamery

Kameru je možné připojit plochým kabelem do příslušného CSI portu. (Pro snazší manipulaci bylo nutné dokoupit delší kabel).

Po připojení je nutné v nastavení aktivovat příslušný interface pro kameru. U novějších verzí Raspberry Pi OS (od verze Bullseye) je již tato možnost označována jako *legacy camera support*.

## 7.7 Zapojení zesilovače s reproduktory



Obrázek 47: Zapojení zesilovače a reproduktorů [63]

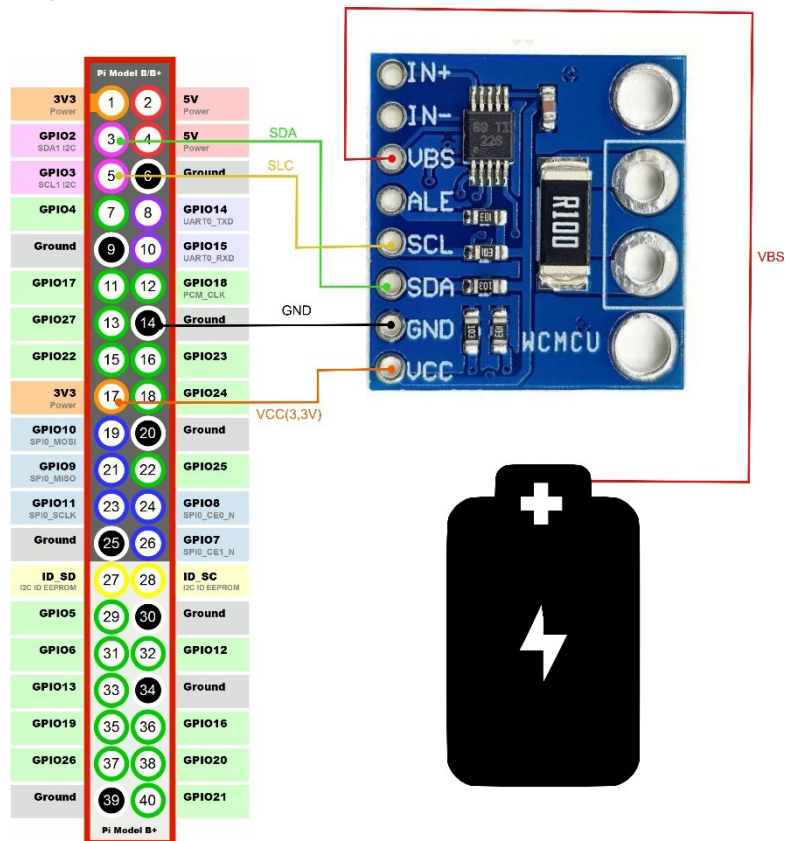
Ilustrační schéma přímo ze stránek prodejce zesilovače značí, jak bylo nutné přiletovat reproduktory, vstup pro napájení a audio výstup (3,5mm jack konektor). Místo baterie v našem případě jako napájení slouží 5V a GND piny Raspberry Pi.

## 7.8 Konfigurace audio vstupu a výstupu v systému

Jelikož je audio výstup připojen přes jack konektor k interní zvukové kartě a jako vstup slouží USB mikrofon, je nutné upravit konfigurační soubor `/etc/asound.conf`. Výchozí zvuková zařízení definujeme tak, aby zvukový výstup (playback) používal interní zvukovou kartu (hw:0), zatímco zvukový vstup (capture) USB mikrofon připojený jako dsnoop:1. plug.

```
pcm.!default {
    type asym
    playback.pcm "plug:hw:0"
    capture.pcm "plug:dsnoop:1"
}
```

## 7.9 Zapojení voltmetru



Obrázek 48: Zapojení modulu pro měření napětí

INA226	RASPBERRY PI
VCC	3V3
GND	GND
SDA	GPIO 2
SLC	GPIO 3

Tabulka 3: Zapojení voltmetru

Modul je napájen 3,3V z Raspberry Pi. GND lze zapojit na libovolný GND pin. SDA a SLC musí být zapojeny na příslušné piny GPIO 2 a GPIO 3 podle tabulky zapojení. Na VBS pin modulu je přivedeno měřené napětí akumulátoru. Jelikož se jedná o akumulátor, ze kterého je napájeno i samotné Raspberry Pi (přes driver board), není nutné zapojovat záporný pól akumulátoru k modulu, neboť ten je již připojen z GND Raspberry Pi.

V nastavení stačí aktivovat I2C interface.

## 8 Použitý software

### 8.1 Operační systém

Jako operační systém byl podle očekávání zvolen **Raspberry Pi OS**. Jedná se o oficiální operační systémem Raspberry Pi Foundation, který byl dříve nazýván **Raspbian**. [70]

Lze tedy očekávat, že bude nejlépe odladěn a také bude nabízet nejlepší podporu. Konkrétně byla nainstalována verze 11 (Bullseye) s jádrem Linux 5.15.61 -v7l+.

### 8.2 Programovací jazyk pro řízení robota

Jakožto nejpoužívanější programovací jazyk pro vývoj na Raspberry Pi byl vybrán **Python** (verze 3.9.2). Jedná se o vysokoúrovňový skriptovací dynamicky interpretovaný programovací jazyk, jehož interpret je defaultně předinstalovaný v Raspberry Pi OS. U většiny volně dostupných návodů je pro programování Raspberry Pi využíván právě tento programovací jazyk. Existuje zde celá řada užitečných knihoven.

#### 8.2.1 Knihovny

Velmi důležitá je pro nás knihovna **RPi.GPIO**, která umožňuje programově ovládat vstupy a výstupy přes piny rozhraní GPIO.

**Tornado** je webový framework pro jazyk Python, který podporuje asynchronní síťové programování. V našem případě je zásadní, že umožňuje navázat spojení prostřednictvím protokolu WebSocket.

Pro asynchronní spouštění funkcí, tak aby neblokovaly hlavní smyčku programu, lze v kombinaci s knihovnou Tornado využít knihovnu **asnycio**.

Přes knihovnu **os** lze v Pythonu provádět příkazy operačního systému.

Knihovna **re** slouží pro práci s regulárními výrazy.

Informace ze sběrnice I2C lze číst pomocí knihoven **board**, **busio** a **adafruit**.



### **8.3 Software pro zajištění přenosu videa a audia**

UV4L (User space Video For Linux) je softwarový balík určený primárně pro webové streamování videa a audia. Nabízí vysokou míru konfigurovatelnosti a podporuje širokou škálu protokolů a rozhraní. Tento software obsahuje jak serverovou, tak i klientskou komponentu. Pro nás je nejpodstatnější, že umožňuje obousměrný přenos videa a zvuku v reálném čase prostřednictvím WebRTC protokolu (viz kapitola WebRTC 3.8). [71]

UV4L je široce využíván ve světě IoT pro video monitoring, bezpečnostní systémy a mnoho dalšího, zejména u amatérských projektů. Jeho otevřená architektura umožňuje snadnou integraci s různými softwarovými a hardwarovými platformami. Dále je k dispozici mnoho pluginů, které umožňují rozšíření funkcionalit, jako jsou například detekce pohybu, zpracování obrazu a mnoho dalšího.

V nové verzi je k dispozici ovladač raspicam-ai, který umožňuje zpracování videa pomocí umělé inteligence. Lze zde spouštět modely neuronových sítí Tensor Flow Lite pro detekci objektů v reálném čase. Pro zpracování obrazových dat je možné využít CPU mikropočítače, což však není příliš efektivní, nebo připojit akcelerátor Google Edge TPU USB. [72]

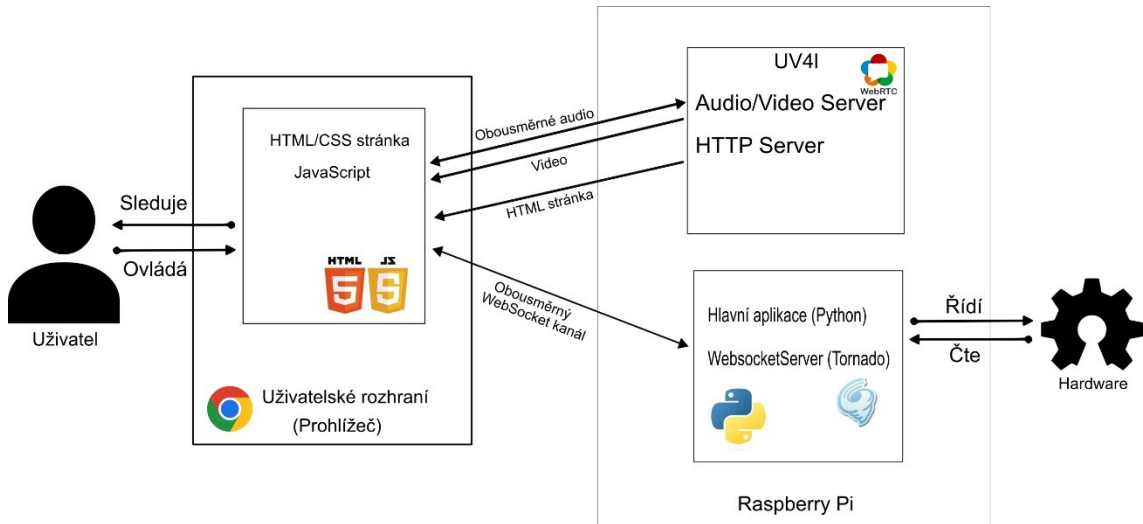
UV4L je k dispozici zdarma jako open source software.

### **8.4 Uživatelské webové rozhraní**

Ovládací webová stránka byla napsána za pomoci HTML/CSS a JavaScriptu bez použití jakýchkoliv dalších rozšiřujících frameworků nebo knihoven.

## 9 Softwarová implementace

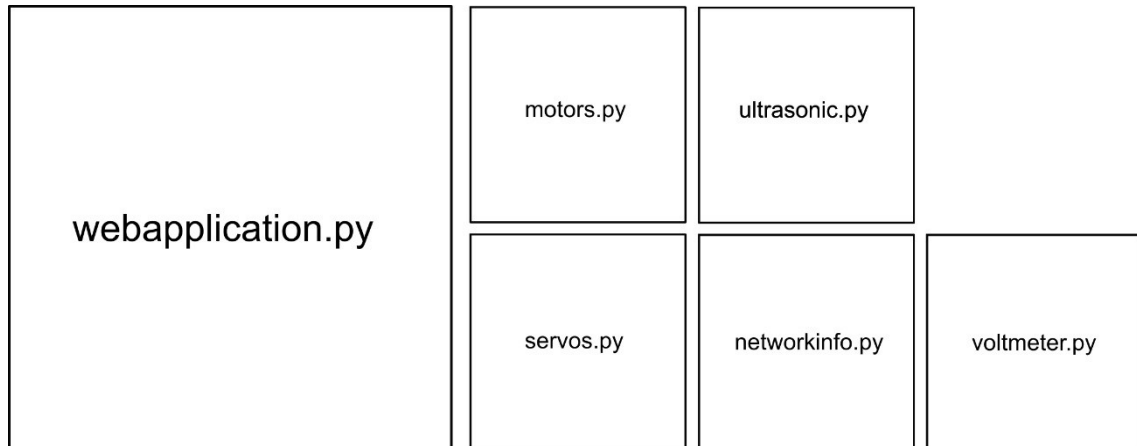
### 9.1 Architektura řešení



Obrázek 49: Architektura řešení

Na Raspberry Pi se nachází hlavní pythonovská aplikace pro řízení robota a dále zde běží tři servery. Prvním z nich je WebSocket server, který je implementován přímo v kódu hlavní Python aplikace za pomoci knihovny Tornado. Díky tomuto serveru je možné zpracovávat povely uživatele pro ovládání robota a zároveň odesílat z robota naměřené informace na uživatelskou webovou stránku. Dále zde běží UV4L server, který zajišťuje streamování videa a obousměrný přenos audio prostřednictvím protokolu WebRTC. Součástí UV4L softwaru je také HTTP server, který odesílá ovládací HTML stránku prohlížeči uživatele. Tento způsob je výhodnější a jednodušší na implementaci (pro zprovoznění obousměrného audio a dalších funkcionalit, které uv4l nabízí), než kdyby stránku vykresloval server Tornado (který tuto možnost také nabízí) a v něm jsme se odkázali na video-stream z UV4L.

## 9.2 Architektura řídicí aplikace



Obrázek 50: Python aplikace

Hlavní část aplikace představuje soubor `webapplication.py`, kde se nachází třída `WebSocketConnection` pro zajištění webové komunikace a rovněž hlavní kód aplikace.

Ostatní soubory představují jednotlivé moduly pro práci s konkrétními hardwarovými komponentami a obsahují stejnojmenné třídy.

V následujících kapitolách jsou detailněji popsány jednotlivé části aplikace.

## 9.3 Ovládání motorů

Jak je zřejmé ze specifikací driver-boardu (viz kapitola 7.2) motory jsou ovládány pomocí pinů 6, 12, 13, 20, 21 a 26. Přičemž piny 6 a 12 slouží jako aktivační a ostatní mění směr otáčení motorů.

Princip ovládání je nejlépe zřejmý z následující tabulky pro jeden motor (v našem případě pravý):

Pin 26 (aktivační)	Pin 20	Pin 21	Výstup
0			Motor je vypnutý
1			Motor aktivován
1	1	0	Dozadu
1	0	1	Dopředu
1/0	0	0	Neotáčí se
1	1	1	Tento stav by neměl nastat

Tabulka 4: Ovládání motoru

H-můstek podle logických hodnot výše nastaví polaritu na svorkách a motor se roztočí zvoleným směrem. [57]

Pro nastavení rychlosti otáčení motoru je možné využít pulzně šířkovou modulaci (PWM). Stačí na aktivačním pinu místo logické jedničky nastavit PWM signál (v programu se zadává v procentech).

Pro ovládání motorů byla vytvořena třída **Motors**.

Na začátku je nutné naimportovat knihovnu **RPi.GPIO**, která se stará o práci s GPIO piny.

```
import RPi.GPIO as GPIO
```

V konstruktoru jsou provedeny všechny inicializační úkony.

```
def __init__(self) -> None:
    # nastavení frekvence pwm
    self.PWM_FREQUENCY = 50
    self.SPEED = 90
    # označení pinů
    self.MR_ACTIVATE = 26
    self.ML_ACTIVATE = 12
    self.MR_F = 21
    self.MR_B = 20
    self.ML_F = 13
    self.ML_B = 6
```

```

# Nastavení GPIO pinů jako výstup
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(self.MR_ACTIVATE, GPIO.OUT)
GPIO.setup(self.ML_ACTIVATE, GPIO.OUT)
GPIO.setup(self.MR_F, GPIO.OUT)
GPIO.setup(self.MR_B, GPIO.OUT)
GPIO.setup(self.ML_F, GPIO.OUT)
GPIO.setup(self.ML_B, GPIO.OUT)

# nastavení pwm frekvence motoru
self.power_MR = GPIO.PWM(self.MR_ACTIVATE, self.PWM_FREQUENCY)
self.power_ML = GPIO.PWM(self.ML_ACTIVATE, self.PWM_FREQUENCY)

```

Na začátku nastavíme pracovní frekvenci PWM a výchozí rychlost motorů. Dále jsou pro přehlednost pojmenovány piny pro ovládání motorů. Všechny piny je třeba nastavit jako výstupní.

K aktivaci motorů slouží metoda **start()**:

```

def start(self):
    self.power_MR.start(self.SPEED)
    self.power_ML.start(self.SPEED)

```

Motorům je při startu nastavena výchozí rychlost otáčení.

K deaktivaci motorů slouží metoda **stop()**:

```

def stop(self):
    self.power_MR.stop()
    self.power_ML.stop()

```

Metoda **forward()** pro jízdu vpřed:

```

def forward(self):
    GPIO.output(self.MR_F, 1)
    GPIO.output(self.MR_B, 0)
    GPIO.output(self.ML_F, 1)
    GPIO.output(self.ML_B, 0)

```

Jak je zřejmé, u obou motorů nastavíme logickou jedničku u piny určující směr dopředu.

Metoda **left()** slouží pro otočení doleva na místě (protichůdný chod pásů) :

```
def left(self):
    GPIO.output(self.MR_F, 1)
    GPIO.output(self.MR_B, 0)
    GPIO.output(self.ML_F, 0)
    GPIO.output(self.ML_B, 1)
```

Metoda **turn\_left()** slouží pro zatočení za jízdy (pravý pás se otáčí vpřed zatímco levý stojí):

```
def turn_left(self):
    GPIO.output(self.MR_F, 1)
    GPIO.output(self.MR_B, 0)
    GPIO.output(self.ML_F, 0)
    GPIO.output(self.ML_B, 0)
```

Pro zastavení je zde metoda **brake()**:

```
def brake(self):
    GPIO.output(self.MR_F, 0)
    GPIO.output(self.MR_B, 0)
    GPIO.output(self.ML_F, 0)
    GPIO.output(self.ML_B, 0)
```

Metoda **change\_speed()** přenastaví rychlost otáčení pásů (změna hodnoty PWM):

```
def change_speed(self, new_speed):
    self.power_MR.ChangeDutyCycle(new_speed)
    self.power_ML.ChangeDutyCycle(new_speed)
```

Pro opačné směry je princip obdobný. Kompletní zdrojový kód je k dispozici na GitHubu.

## 9.4 Ovládání servomotorů

Jako první musíme nainportovat knihovnu `gpiozero`, ze které potřebujeme modul `Servo` pro samotné ovládání servomotorů a `PiGPIOFactory`, pomocí kterého je možné pro řízení servomotorů generovat přesnější PWM signál hardwarově. Eliminuje se tak chvění, ke kterému dochází při řízení pomocí standardně softwarově generovaného PWM signálu.

```
from gpiozero import Servo
from time import sleep
from gpiozero.pins.pigpio import PiGPIOFactory
```

Pro správnou funkci modulů je nejprve nutné spustit na pozadí server pomocí příkazu:

```
$ sudo pigpiod
```

Je důležité vědět, že tento server obsadí **port 8888**, takže jej již není možné použít pro komunikaci s naší webovou aplikací.

```
def __init__(self) -> None:
    FACTORY = PiGPIOFactory()

    # piny serv
    S_HORIZONTAL_PIN = 14
    S_VERTICAL_PIN = 15

    self.servo_horizontal = Servo(S_HORIZONTAL_PIN,
pin_factory=FACTORY)
    self.servo_vertical = Servo(S_VERTICAL_PIN, pin_factory=FACTORY)

    self.servo_horizontal.value = 0
    self.servo_vertical.value = 0
```

V konstruktoru třídy `Servos` nastavíme příslušné piny a pomocí výše zmíněných modulů definujeme serva pro otáčení v obou osách.

Takto by vypadala metoda pro otočení směrem vlevo o předem daný úhel `ANGLE`.

```
def left(self):
    if (self.servo_horizontal.value < (0.9-ANGLE)):
        self.servo_horizontal.value += ANGLE
```

Metody pro ostatní směry jsou obdobné.

## 9.5 Obsluha dálkoměru

```
from gpiozero import DistanceSensor
from gpiozero.pins.pigpio import PiGPIOFactory

class Ultrasonic:

    def __init__(self) -> None:
        FACTORY = PiGPIOFactory()
        TRIGGER_PIN = 17
        ECHO_PIN = 27

        self.sensor = DistanceSensor(trigger=TRIGGER_PIN,
                                     echo=ECHO_PIN, pin_factory=FACTORY)

    def get_distance(self):
        distance = self.sensor.value
        distance_cm = distance*100
        return distance_cm
```

Pro používání ultrazvukového dálkoměru byla vytvořena třída **Ultrasonic**. Stejně jako u servomotorů i zde byla využita knihovna **gpiozero**, která obsahuje modul **DistanceSensor**. Tento modul dokáže změřit vzdálenost v metrech, která je následně dostupná přes atribut *value*. Protože pro naše účely je nejvhodnější měřit vzdálenost od překážky v centimetrech, násobíme získanou vzdálenost stem.

## 9.6 Zjišťování informací o síti

```
import os
import re

class NetworkInfo:

    DEFAULT_INTERFACE = "wlan0"

    def __init__(self) -> None:
        pass

    # síla wifi signálu
    def get_wifi_signal(self, interface=DEFAULT_INTERFACE):
        signal = os.popen('iwconfig ' + interface +
                          ' | grep "Signal level"').read().strip()
        result = re.findall(r"(-\d+)\s*dBm", signal)
        return int(result[0])
```



```
# IP adresa
def get_ip(self, interface=DEFAULT_INTERFACE):
    ip = os.popen("ip a show " + interface +
                 "| grep inet").read().split("\n")[0].strip()
    ips = re.findall(r"(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})", ip)
    return ips[0]

zdroj: [57]
```

Třída **NetworkInfo** slouží ke zjišťování síly wifi signálu a aktuální IP adresy zařízení. K tomuto účelu jsou využity funkce operačního systému a regulární výrazy se kterými je možné pracovat pomocí knihoven `os` a `re`. [57]

Jako výchozí `interface` je přednastavena `wlan0`, ale při volání metod je možné zadat i jiný parametr.

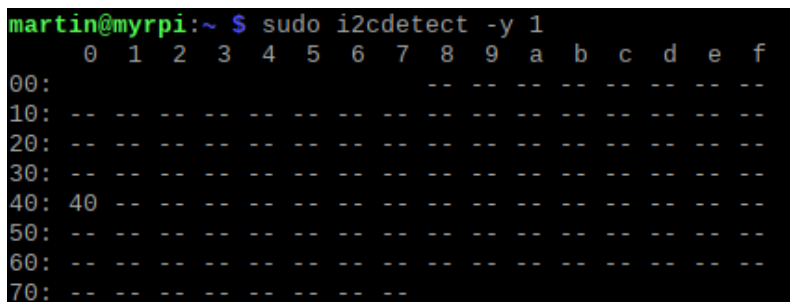
## 9.7 Měření stavu baterie

V době tvorby práce bohužel nebyla dostupná žádná pythonovská knihovna určená konkrétně pro modul `INA226`. Bylo tedy nutné využít obecnou knihovnu `adafruit_bus_device.i2c_device`, pomocí které lze číst data I2C sběrnice.

Ověřit na které adrese sběrnice se voltmetr nachází, lze pomocí příkazu:

```
$ sudo i2cdetect -y 1
```

Typicky je identifikovatelný na adrese 40.



```
martin@myrpi:~ $ sudo i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  40  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Obrázek 51: Adresy I2C sběrnice

Pro práci s voltmetrem byla vytvořena třída **Voltmeter**. K přečtení naměřeného napětí slouží následující kód.

```
import board
import busio
from adafruit_bus_device.i2c_device import I2CDevice
import struct

I2C_ADDRESS = 0x40
REGISTER_BUS_VOLTAGE = 0x02

i2c = busio.I2C(board.SCL, board.SDA)
device = I2CDevice(i2c, I2C_ADDRESS)

class Voltmeter:

    def read_voltage(self):
        with device:
            device.write(bytes([REGISTER_BUS_VOLTAGE]))
            voltage_data = bytearray(2)
            device.readinto(voltage_data)
            voltage_raw = struct.unpack(">H", voltage_data)[0]
            voltage = voltage_raw * 1.25 / 1000.0
            return voltage
```

Abychom v aplikaci mohli přehledně zobrazit přibližný stav baterie, je nutné naměřené napětí přepočítat na procenta.

```
def get_battery_percent(self, min=9, max=12):
    voltage = self.read_voltage()

    result = 0

    if voltage < min:
        result = 0
    elif voltage > max:
        result = 100
    else:
        result = ((voltage - min) / (max - min)) * (100)
    return result
```

Horní mez udává napětí plně nabitě baterie, dolní mez napětí, při kterém baterie již není schopná dodat dostatek energie pro činnost robota. Uvažujeme, že stav napětí při vybíjení klesá lineárně. Můžeme tedy aktuální napětí přepočítat na procenta podle vzorce výše.

## 9.8 Webová aplikace

### 9.8.1 Ovládání pohybu robota

#### 9.8.1.1 Serverová část

O příjem instrukcí od uživatele se stará **WebSocket server** implementovaný pomocí knihovny Tornado. Pro tento účel byla vytvořena třída **WebSocketConnection**, která je odvozena z třídy `WebSocketHandler`. V této třídě je možné pomocí metod `open()`, `on_close()` a `on_message()` definovat, co se má stát po připojení klienta, pokud se klient odpojí a jak reagovat na přijaté zprávy od klienta.

Klient je po připojení uložen do seznamu klientů, pokud se jedná o prvního klienta, nastartují se motory a spustí se plánovače, které odesílají informace ze serveru.

```
def open(self):
    websocket_clients.append(self)
    ...
    if len(websocket_clients) == 1:
        move.start()
        scheduler_ultrasonic.start()
        scheduler_battery.start()
        scheduler_signal.start()
```

Po odpojení je klient ze seznamu odstraněn. Pokud už není připojen žádný klient, motory jsou vypnuty a plánovače zastaveny.

```
def on_close(self):
    websocket_clients.remove(self)
    if len(websocket_clients) == 0:
        move.stop()
        scheduler_ultrasonic.stop()
    ...
```

Nejzajímavější je pro nás metoda `on_message()`, kde podle obdrženého znaku od klienta zavoláme příslušnou metodu pro ovládání motorů.

```
def on_message(self, message):
    if message == "F":
        move.forward()
    ...
    elif message == "B":
        move.backward()
    ...
```

Stejně vypadá i řízení ostatních směrů a obdobně je řešeno i ovládání servomotorů pro otáčení kamery:

```
elif zprava == "CU":
    camera.up()
```

Protože jako webserver slouží UV4L nesmíme zapomenout na začátek třídy `WebSocketConnection` vložit následující kód, který povolí navázat spojení z vnější stránky, což je z bezpečnostních důvodů defaultně zakázáno.

```
def check_origin(self, origin):
    return True
```

### 9.8.1.2 Klientská část

O odesílání pokynů od klienta se stará jednoduchý JavaScript, který při stisku klávesy odešle příslušný řídicí kód.

```
window.addEventListener("keydown", function (event) {

    if (event.code === "ArrowUp") {
        UP = true
        ...

    if (UP && LEFT) {
        ws.send("LF")
    }
    else if (LEFT) {
        ws.send("L")
    }
    ...
})
```

Je důležité rozlišit, zda je šipka vlevo stisknuta samostatně nebo společně se šipkou dopředu. Podle toho je na serveru zavolána funkce `left()`, která slouží k otočení na místě nebo funkce `turn_left()` která slouží k zatočení za jízdy (viz kapitola ovládání motorů 9.3).

Jakmile uživatel klávesu pustí, je odeslán kód „X“ který signalizuje zastavení. V případě šipek pro zatáčení je kód trochu komplikovanější, jelikož chceme, aby robot stále pokračoval v jízdě dopředu, pokud byly šipky stisknuty současně a šipka dopředu ještě nebyla puštěna.

```

window.addEventListener('keyup', function (event) {
    if (event.code === "ArrowUp") {
        ws.send("X");
        UP = false
    }
    ...
    else if (event.code === "ArrowLeft") {
        LEFT = false
        if (UP) {
            ws.send("F");
        }

        else if (DOWN) {
            ws.send("B");
        }

        else {
            ws.send("X")
        }
    }
}
...

```

Robota lze rovněž ovládat stisknutím tlačítka na obrazovce pomocí myši.

```

$("#forward").mousedown(function () {
    ws.send("F");
});

```

## 9.8.2 Odesílání informací ze serveru na klienta

### 9.8.2.1 Serverová část

WebSocket umožňuje obousměrnou komunikaci, tudíž je možné vytvořený kanál využít rovněž pro odesílání informací ze serveru na klienta.

Pro tento účel byla vytvořena funkce `websocket_send_message()`, kterou můžeme následně využít pro odesílání naměřených hodnot z dálkoměru a informací o síle wifi signálu a stavu baterie. Na začátek zprávy je přidán znak, podle kterého lze určit, o jakou hodnotu se jedná.

```

def websocket_send_message(message):
    for websocket_client in websocket_clients:
        websocket_client.write_message(message)

```

```

async def loop_ultrasonic():
    distance = ultrasonic_sensor.get_distance()
    websocket_send("D" + str(distance))
...

```

Funkce pro odesílání naměřených hodnot jsou pomocí klíčového slova **async** definovány jako asynchronní. V plánovačích jsou spouštěny pomocí asynchronní smyčky.

```

scheduler_ultrasonic = ioloop.PeriodicCallback(
    lambda: asyncio.ensure_future(loop_ultrasonic()), 600)

```

Tak je díky knihovně **asyncio** zajištěno, že jsou spouštěny asynchronně a neblokují hlavní smyčku programu ani metodu `on_message()`. Díky tomu by program měl běžet plynule, aniž by došlo ke konfliktům nebo zaseknutí.

### 9.8.2.2 Klientská část

```

ws.onmessage = function (message) {
    if (message.data.substring(0, 1) == "D") {
        var distance = parseInt(message.data.substring(1));
        $("#distance").html(distance + " cm");
        if (distance < dangerous_distance) {
            $("#distance").css("color", "red");
        }
        else {
            $("#distance").css("color", "white");
        }
    }
    else if (message.data.substring(0, 1) == "W") {
        var signal = parseInt(event.data.substring(1))
        $("#signal").html(signal + " dBm");
    }
}
...

```

Zprávy ze serveru jsou u klienta přijaty pomocí WebSocket spojení, rozříděny podle přiloženého znaku, dále zpracovány a vykresleny na ovládací panel uživatelského rozhraní.

## 9.9 Ovládání pomocí joysticku

Pro ovládání robota přes Bluetooth pomocí ovladače Dualshock pro PlayStation 4 lze využít jednoduchou pythonovskou knihovnu **pyPS4Controller**. Tuto knihovnu je nejprve nutné nainstalovat pomocí příkazu:

```
$ sudo pip install pyPS4Controller
```

V ovládacím skriptu vytvoříme třídu, která je odvozena od třídy `Controller`, kde přepsáním příslušných metod definujeme, jak náš program bude reagovat na stisk konkrétních tlačítek a pohyb analogových páček. Třída obsahuje metody `„on_tlacitko_press()“`, které definují, co se stane při stisku tlačítka a metody `„on_tlacitko_release()“`, které definují, co se stane po puštění tlačítka.

Pokud bychom například chtěli použít tlačítko „X“ pro jízdu vpřed, náš kód by vypadal takto:

```
def on_x_press(self):
    move.change_speed(DEFAULT_SPEED)
    move.forward()
```

Po puštění tlačítka by robot opět zastavil.

```
def on_x_release(self):
    move.brake()
```

Pro plynulé ovládání rychlosti lze využít analogových joysticků.

```
def on_R3_up(self, value):
    speed = value
    speed_normalized = speed * -100 / 32767
    move.change_speed_right(speed_normalized)
    move.forward()

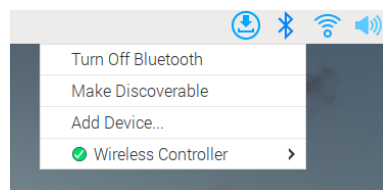
def on_R3_y_at_rest(self):
    move.change_speed_right(0)
```

Hodnoty **value**, které ovladač vysílá při pohybu joysticků, se pohybují v rozmezí od -32767 do 32767 pro každou osu (x a y). Přičemž minusové hodnoty značí pohyb nahoru a doleva a plusové dolů a doprava. Abychom mohli hodnotu použít pro řízení rychlosti, je nutné ji normalizovat. V příkladu výše se hodnoty nachází v rozmezí 0 (páčka je

uprostřed) až -32767 (páčka je nahoře). Rychlost pomocí PWM se nastavuje v hodnotách 0-100 (procenta) je tedy nutné hodnotu vynásobit číslem  $(-100) / 32767$ .

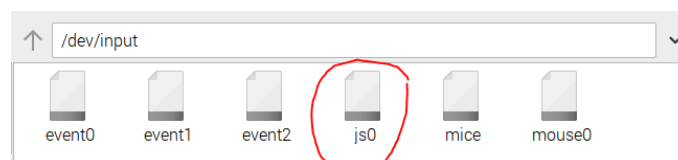
V našem případě bylo ovládání vyřešeno tak, že levá páčka ovládá rychlost a směr (dopředu/dozadu) levého pásu a pravá páčka rychlost a směr pravého. Obdobným způsobem bylo řízení řešeno například u starých bagrů nebo jednoduchých lodí na dálkové ovládání.

Ovladač je nejprve nutné s Raspberry Pi správně spárovat. Nejsnáze toho docílíme přes grafické uživatelské rozhraní, pokud máme možnost připojení k Raspberry Pi přes vzdálenou plochu. Nejprve je nutné přepnout gamepad do režimu párování (současně stiskneme tlačítka *share* a *playstation*, dokud indikační led dioda nezačne blikat). Poté na Raspberry Pi v nastavení Bluetooth přidáme nové zařízení a vybereme náš gamepad. Pokud vše dopadlo správně, ukazuje se náš ovladač jako připojený a indikační led dioda souvisle svítí. Po příštím spuštění by se již ovladač měl připojit automaticky.



Obrázek 52: Připojený gamepad

Po správném spárování se ovladač objeví ve složce `/dev/input`. Pokud se jedná o první připojený ovladač, název by měl být `js0`.



Obrázek 53: umístění gamepadu

Toto jméno se následně použije jako parametr konstruktoru při vytváření instance třídy pro ovladač.

```
controller = MyController(interface="/dev/input/js0",  
                           connecting_using_ds4drv=False)
```

Tento způsob řízení lze samozřejmě využít jen v případě, pokud se nacházíme ve stejné místnosti (respektive v dosahu Bluetooth přijímače).

Kód pro řízení pomocí gamepadu se nachází v souboru `gamepad.py` a nijak nesouvisí s hlavní aplikací.



## 9.10 Přenos videa a audia

Na Raspberry Pi byl nejprve nainstalován softwarový balík UV4L. Postup instalace je detailně popsán na oficiálních stránkách organizace:

<https://www.linux-projects.org/uv4l/installation/>

Správnou funkcionalitu a různé možnosti nastavení serveru před implementací vlastní webové stránky lze vyzkoušet přes předpřipravená demo, která běží defaultně na portu 8080. Pokud byly všechny součásti správně nainstalovány, zobrazí se v prohlížeči po zadání IP adresy zařízení a portu následující obrazovka:



Obrázek 54: UV4L Demos [71]

Zajímavá je pro nás především první karta **Two-way Audio/Video** (WebRTC). Kde lze vyzkoušet přenos videa s různými parametry a přenos audia ze serveru i na server.

Veškeré možnosti konfigurace lze nastavit v konfiguračním souboru `raspicam.conf`, ke kterému se dostaneme příkazem:

```
$ sudo nano /etc/uv4l/uv4l-raspicam.conf
```

Pro zprovoznění WebRTC serveru pro přenos videa a audia je nutné v konfiguračním souboru aktivovat webrtc přenos videa a audia.

```
### WebRTC options:
server-option = --enable-webrtc=yes
server-option = --enable-webrtc-datachannels=yes
server-option = --webrtc-datachannel-label=uv4l
server-option = --webrtc-datachannel-socket=/tmp/uv4l.socket
server-option = --enable-webrtc-video=yes
server-option = --enable-webrtc-audio=yes
server-option = --webrtc-receive-audio=yes
```

Video je přenášeno ve formátu H.264. Rozlišení činí 1280 x 720 pixelů a frekvence 30 snímků za vteřinu. Parametry je možné měnit podle potřeby.

Pro spuštění HTTP serveru je nutné nastavit cestu k HTML souboru, který bude přenášán a zvolit volný port na kterém server poběží.

```
### These options are specific to the HTTP/HTTPS Server
### serving custom Web pages only:
server-option = --enable-www-server=yes
server-option = --www-root-path=/home/martin/robot
server-option = --www-index-file=index.html
server-option = --www-port=8877
```

Kompletní konfigurační soubor je rovněž přiložen na GitHubu.

### 9.10.1 Rozpoznávání objektů v obraze:

Pokud chceme pro detekci objektů využít umělé inteligence je nutné použít ovladač **raspicam-ai**.

Na začátku konfiguračního souboru lze vybrat, který ovladač bude použit.

```
#####
# uv4l core options
#####

driver = raspicam-ai
```

V sekci *TensorFlow model options* lze nastavit cestu k souboru s modelem, parametry pro rozpoznávání a zapnout vykreslování hranic objektů do přenášeného obrazu. Volitelně lze také nastavit cestu k souboru s popisky.

```
### TensorFlow model options (the EdgeTPU accelerator is
optional)
  tflite-model-file = /home/martin/aimodels/face.tflite
  tflite-model-output-topk = 5
  tflite-model-output-threshold = 0.25
  tflite-overlay-model-output = yes # to draw boundary boxes,
# tflite-labels-file = /home/martin/aimodels/labeld.txt
```

Podrobný popis, jak do svého projektu zapojit umělou inteligenci, je také k dispozici na stránkách tvůrců UV4L:

<https://www.linux-projects.org/uv4l/tutorials/video-tracking-with-tensor-flow/>

Natrénované testovací modely TensorFlow, je možné stahovat z oficiálních stránek:

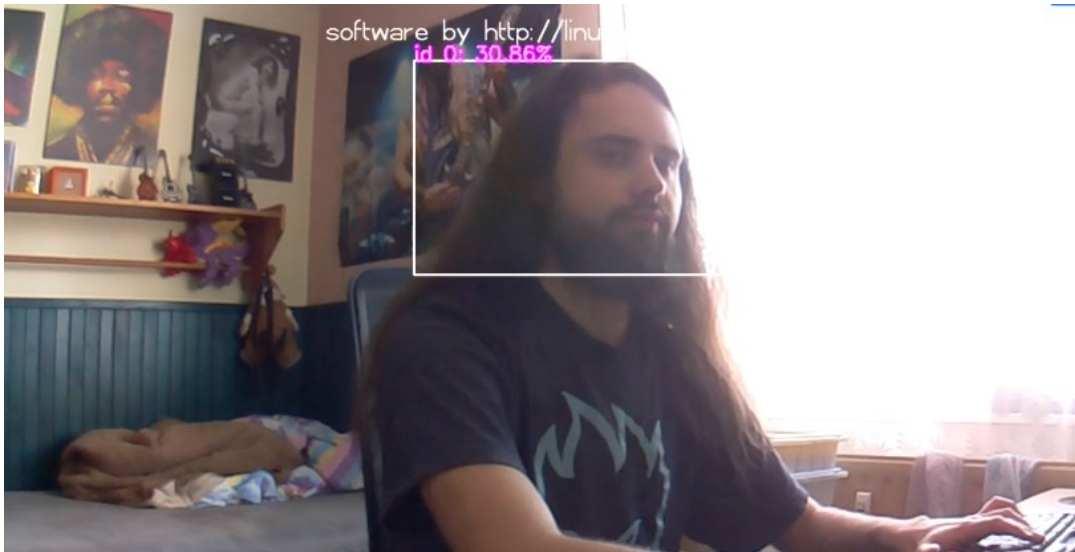
<https://coral.ai/models/>

U většiny modelů je možné vybrat verzi, kterou lze spustit přímo na CPU mikropočítače nebo Edge TPU verzi, která je zkompileována pro použití s USB akcelerátorem.

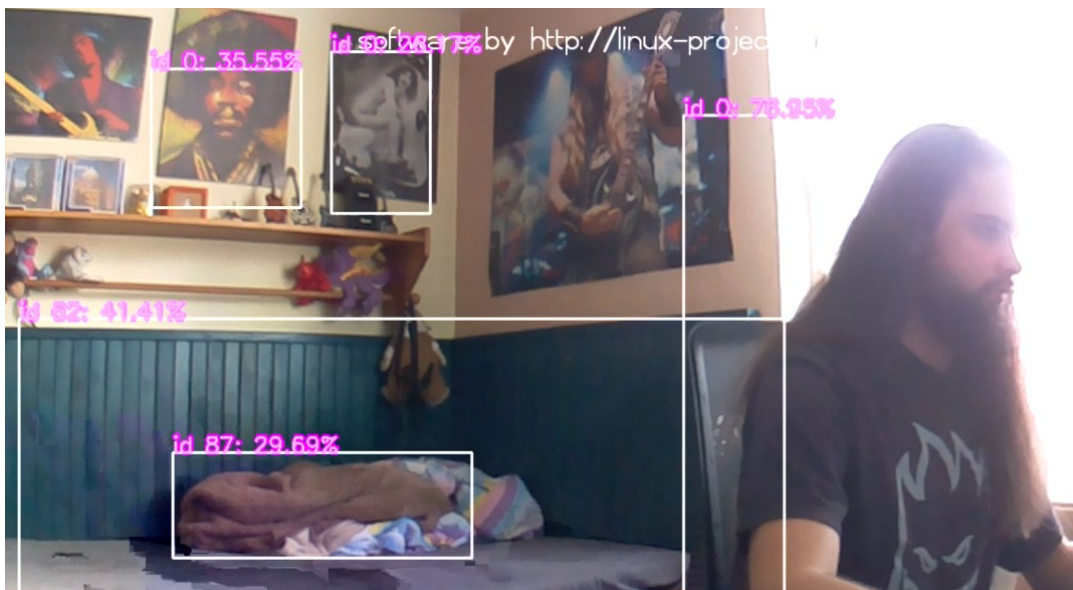
V rámci této práce byly modely testovány pouze s využitím CPU, což se ukázalo jako pomalé. Jako jediný použitelný model se jevil SSD MobileNet V2 Faces, pro rozpoznávání obličejů, i zde však bylo znát výrazné zpomalení. Latence videa se zvýšila tak, že bylo na první pohled snadno pozorovatelné zpoždění a zpomalení pohybu. Při detekci obličeje pomocí OpenCV, které nevyužívá neuronových sítí, ale předem definovaných algoritmů (některé z nich jsou zmíněny v kapitole 3.6) byl přenos výrazně svižnější, i když i zde bylo viditelné lehké zpoždění oproti videu bez zpracování.

Vzorový příklad využívající OpenCV je popsán na následujícím odkaze a měl by být automaticky stažen při instalaci UV4L.

<https://www.linux-projects.org/uv4l/tutorials/custom-webapp-with-face-detection/>

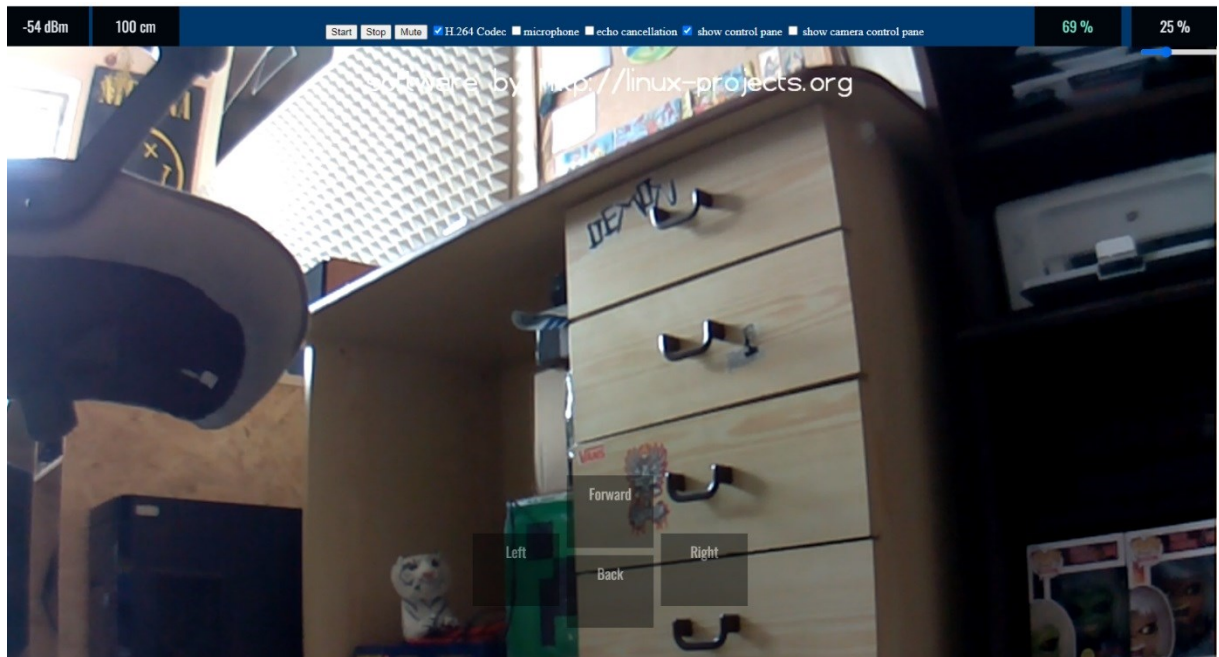


Obrázek 56: Detekce obličejů



Obrázek 55: Detekce objektů

## 10 Výsledná podoba uživatelského rozhraní



Obrázek 57: Uživatelské rozhraní

Horní lišta obsahuje tlačítka **start** a **stop** pro zapnutí a vypnutí video-streamu, tlačítko **mute/(unmute)** pomocí kterého lze ztlumit zvuk zapnutého streamu a zaškrťovací políčko pro mikrofon, kterým lze aktivovat přenos hlasu klienta směrem na server. Dále se zde nacházejí tři informační panely pro zobrazení síly signálu wifi, vzdálenost od překážky před robotem a stav nabití baterie. Pokud některá z hodnot klesne pod kritickou mez, údaj se zbarví červeně. U posledního panelu lze pomocí posuvníku nastavit rychlost otáčení motorů (v procentech). V dolní části stránky se nacházejí tlačítka, kterými lze ovládat pohyb robota a otáčení kamery (pokud nemůžeme využít klávesnice). Tyto tlačítka je možné zobrazit nebo skrýt pomocí jednotlivých checkboxů na ovládacím panelu.

## 11 Testování

Všechny součásti projektu byly testovány průběžně během vývoje. Po dokončení byl robot testován uživatelsky (pouze autorem práce).

Zařízení bylo testováno pouze v rámci lokální sítě, avšak při použití VPN nebo jiné z možností vzdáleného přístupu by nemělo nastat k žádné změně funkčnosti. Z webových prohlížečů se nejlépe osvědčil Google Chrome, jako funkční se jevil i Microsoft Edge, u prohlížeče Mozilla Firefox se vyskytl problém s kodekem přenášeného videa.

V průběhu testování bylo nalezeno několik nedostatků zejména v hardwarové části, které v rámci aktuálního prototypu nebylo možné snadno odstranit, nepředstavují však žádný problém pro běžné využití.

Například při otáčení horizontálního servomotoru se v určitých úhlech místy vyskytují vibrace. Tento neduh je způsoben pravděpodobně tím, že držák kamery je koncipován pro klasickou RPi kameru a při využití větší kamery s IR reflektory je servo (které drží váhu celé konstrukce) příliš zatíženo. Nejjednodušším řešením, jak toto chvění eliminovat, by tedy bylo použít klasickou kameru bez nočního vidění. V případě potřeby nočního vidění se lze obejít bez jejího otáčení do stran (tato funkce je víceméně nadbytečná, jelikož se lze snadno otočit celým robotem). V případě potřeby obou těchto funkcionalit by bylo nutné přejít na jiný výrazně komplikovanější systém uchycení kamery.

Použitý mikrofon vykazoval slyšitelný šum, nicméně přenášený zvuk byl srozumitelný.

Řízení přes klávesnici bylo díky motorům s pomalým převodem velmi plynulé a celkový dojem z ovládání byl dobrý.

## 12 Možný rozvoj práce

Je nutné si uvědomit, že vytvořený projekt představuje spíše jakýsi koncept než finální produkt. Při vytváření platformy byly využity pouze veřejně dostupné open-source technologie, standardizované protokoly a běžně dostupné komponenty. Nebyly použity žádné proprietární technologie. Mělo by tedy být možné bez větších obtíží robota libovolně rozšiřovat, vylepšovat a upravovat dle konkrétních požadavků pro specifické využití jak v hardwarové, tak v softwarové oblasti.

Například lze použít větší a odolnější podvozek, přidat uzavřenou (avšak dobře větranou) krabici pro schování komponent. Využít akumulátor o větší kapacitě, případně přidat solární panel pro jeho postupné dobíjení a modul pro připojení k mobilnímu internetu. Tak by bylo možné využít robota i ve venkovních prostorech. Tyto modifikace nijak neovlivní základní principy fungování platformy a měly by být snadno proveditelné bez výraznějších úprav v softwaru.

Dalším možným rozvojem projektu by mohlo být zapojení umělé inteligence pro částečnou automatizaci a autonomní činnost robota. Zde se nabízí například využití neuronových sítí a naučení robota, jak vykonávat různé úkoly, jako například rozpoznávání objektů a překážek, plánování trasy a autonomní orientaci v prostředí. Pro tento účel by byl lepší volbou mikropočítač Nvidia Jetson s vestavěnou grafickou kartou pro výpočetně náročné úlohy. Avšak i v případě Raspberry Pi existují různé možnosti jako použití externí USB jednotky pro akceleraci výpočtů.

## 13 Závěr

V práci byly teoreticky popsány různé možnosti pohonu a řízení robotů, protokoly webové komunikace, způsoby detekce překážek a počítačové vidění, důkladně porovnány některé vývojové platformy a vysvětleny principy související s těmito technologiemi.

Následně byl navržen a implementován funkční prototyp robota, který je schopný živě přenášet video a obousměrné audio a je jej možné ovládat prostřednictvím webové aplikace. Součástí práce je jak softwarová, tak i hardwarová část projektu.

Celý projekt byl realizován na bázi open-source a pro jeho implementaci byly použity standardizované protokoly a běžně dostupné komponenty. Všechny zdrojové kódy jsou veřejně dostupné na on-line platformě GitHub a všechny hardwarové části včetně schémat jejich zapojení jsou součástí textu práce.

Nakonec byly navrženy možnosti dalších vylepšení a rozšíření robota a bylo nastíněno, jakým směrem by se mohl celý projekt dále ubírat.



## **I. Summary and keywords**

The thesis focuses on the design and implementation of a functional mobile robot, which can be controlled through an interactive web application. The robot is capable of transmitting video and two-way audio, enabling its use in areas where remote monitoring is necessary (e.g. monitoring pets at home). The project is designed as an open-source platform with an emphasis on low cost and easy expansion of additional functionalities.

### **Keywords:**

Robotics, Raspberry Pi, DIY, monitoring, open-source, remote control, video and audio transmission, robot propulsion and control

## II. Seznam použitých zdrojů

- [1] How to Make a Robot: Lesson 2: Choosing a Robotic Platform. Robotshop.com: community [online]. 17. 9. 2018 [cit. 2022-11-22]. Dostupné z: <https://community.robotshop.com/tutorials/show/how-to-make-a-robot-lesson-2-choosing-a-robotic-platform>
- [2] SIEGWART, Roland a Illah NOURBAKHS. Introduction to autonomous mobile robots. Cambridge, Massachusetts: A Bradford Book, 2004. ISBN 0-262-19502-X.
- [3] Wheels vs Continuous Tracks: Advantages and Disadvantages. Intorobotics.com [online]. 11. 11. 2013 [cit. 2022-11-22]. Dostupné z: <https://www.intorobotics.com/wheels-vs-continuous-tracks-advantages-disadvantages/>
- [4] Robotshop.com [online]. [cit. 2022-11-22]. Dostupné z: <https://www.robotshop.com/en/raspberry-pi-car-robot-kit-compatible-w-pi-4b3b3b.html>
- [5] Robosavvy.co.uk [online]. [cit. 2022-11-22]. Dostupné z: <https://robosavvy.co.uk/dagu-wild-thumper-6wd-black-w-wheel-encoders-34-1-gearboxes.html>
- [6] Leeselectronic.com [online]. [cit. 2022-11-22]. Dostupné z: <https://leeselectronic.com/en/product/100259-robot-3-wheel-chassis.html>
- [7] Laskakit.cz: Pásový robotický podvozek TP100 [online]. [cit. 2022-11-23]. Dostupné z: [https://www.laskakit.cz/pasovy-roboticky-podvozek-tp100/?gclid=CjwKCAiApvebBhAvEiwAe7mHSE-myEIN6jzEjtaAEI0UAcJaiVk64Ig\\_6xu17BusqnnCJGuF2123NRoC5ooQAvD\\_BwE](https://www.laskakit.cz/pasovy-roboticky-podvozek-tp100/?gclid=CjwKCAiApvebBhAvEiwAe7mHSE-myEIN6jzEjtaAEI0UAcJaiVk64Ig_6xu17BusqnnCJGuF2123NRoC5ooQAvD_BwE)
- [8] HORÁK, Jan. Záznam o závěrečné práci na Katedře robotiky: Konstrukční návrh pásového podvozku pro větší nosnost. VŠB TECHNICKÁ UNIVERZITA OSTRAVA: fakulta strojní, katedra robotiky [online]. 22. 05. 2009 [cit. 2022-12-05]. Dostupné z: <http://robot2.vsb.cz/zaverecne-prace/487/>

- [9] MAREŠ, Michal. Konstrukce mobilního pásového podvozku transportního stroje [online]. Brno, 2015 [cit. 2022-12-05]. Dostupné z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=103078](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=103078)  
Bakalářská. VUT. Vedoucí práce Ing. Václav Pištěk, DrSc.
- [10] SHAMAN, Ben, Dimi APOSTOLOPOULOS, Michael WAGNER a William WHITTAKER. Effect of Tire Design and Steering Mode on Robotic Mobility in Barren Terrain [online]. In: . Field Robotics Center, The Robotics InstituteCarnegie Mellon UniversityPittsburgh PA 15213, 30. 4. 2014 [cit. 2022-12-12]. Dostupné z: [https://www.researchgate.net/publication/237768170\\_Effect\\_of\\_Tire\\_Design\\_and\\_Steering\\_Mode\\_on\\_Robotic\\_Mobility\\_in\\_Barren\\_Terrain](https://www.researchgate.net/publication/237768170_Effect_of_Tire_Design_and_Steering_Mode_on_Robotic_Mobility_in_Barren_Terrain)
- [11] How to Make a Robot: Lesson 3: Making Sense of Actuators. Robotshop.com: community [online]. 17. 9. 2018 [cit. 2022-12-05]. Dostupné z: <https://community.robotshop.com/tutorials/show/how-to-make-a-robot-lesson-3-making-sense-of-actuators>
- [12] PAŠTYLKA, Jan. DC motory řízené pomocí H můstku [online]. 2017 [cit. 2022-12-13]. Dostupné z: [https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H\\_Bridge.pdf](https://embedded.fel.cvut.cz/sites/default/files/kurzy/lpe/h-bridge/H_Bridge.pdf) Dokumentace. České vysoké učení technické v Praze.
- [13] ČÍŽEK, Jakub. Postavíme čtyřmotorové autíčko. Zive.cz: Pojd'me programovat elektroniku [online]. 6. 11. 2016 [cit. 2022-12-13]. Dostupné z: <https://www.zive.cz/clanky/pojdme-programovat-elektroniku-postavime-ctyrmotorovy-tank/sc-3-a-184808/default.aspx>
- [14] Dual TB9051FTG Motor Driver for Raspberry Pi. Pololu.com: robotics & electronics [online]. [cit. 2022-12-13]. Dostupné z: <https://www.pololu.com/product/2762>
- [15] Integrovaný obvod L293 H-můstek. Postavrobota.cz [online]. [cit. 2022-12-13]. Dostupné z: <https://www.postavrobota.cz/Integrovaný-obvod-L293-H-mustek-d1832.htm>
- [16] VIDIM, Jan. DOMAT CONTROL SYSTEM S.R.O. PWM signál a práce s ním. Tzb-info.cz: vytápění [online]. 17. 11. 2014 [cit. 2022-12-05]. Dostupné z: <https://vytapani.tzb-info.cz/mereni-a-regulace/11990-pwm-signal-a-prace-s-nim>

- [17] PAVLIS, Jakub. PWM regulace podsvícení displejů i klávesnic aneb proč nás bolí oči. Notebook.cz: Technologie [online]. 19. 7. 2017 [cit. 2022-12-05]. Dostupné z: <https://notebook.cz/clanky/technologie/2017/pwm-regulace>
- [18] Pulse Width Modulation. Sparkfun.com: Learn [online]. [cit. 2022-12-05]. Dostupné z: <https://learn.sparkfun.com/tutorials/pulse-width-modulation/all>
- [19] Bezdrátové nabíjení. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 21. 5. 2022 [cit. 2023-02-26]. Dostupné z: [https://cs.wikipedia.org/wiki/Bezdr%C3%A1tov%C3%A9\\_nab%C3%ADjen%C3%AD](https://cs.wikipedia.org/wiki/Bezdr%C3%A1tov%C3%A9_nab%C3%ADjen%C3%AD)
- [20] Jak funguje bezdrátové nabíjení: Články. Alza.cz [online]. 28. 3. 2022 [cit. 2023-02-26]. Dostupné z: <https://www.alza.cz/jak-na-bezdratove-nabijeni>
- [21] Wireless Power Transfer. Watelectronics.com: Embedded Systems [online]. 8. 1. 2022 [cit. 2023-02-26]. Dostupné z: <https://www.watelectronics.com/wireless-power-transfer/>
- [22] What is Arduino?. Arduino.cc [online]. 5. 2. 2018 [cit. 2022-12-05]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>
- [23] UPTON, Eben a Gareth HALFACREE. Raspberry Pi: uživatelská příručka. Brno: Computer Press, 2013. ISBN 978-802-5141-168.
- [24] BECHYNSKÝ, Štěpán. Raspberry Pi a GPIO: co funguje. Zdrojak.cz [online]. 27. 2. 2015 [cit. 2022-12-05]. Dostupné z: <https://zdrojak.cz/clanky/raspberry-pi-gpio-co-funguje/>
- [25] Raspberry Pi 4. Raspberrypi.com [online]. [cit. 2022-12-12]. Dostupné z: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- [26] TAN, Cherie. Jetson Nano vs Raspberry Pi 4: The Differences. All3dp.com [online]. 20. 10. 2022 [cit. 2022-12-05]. Dostupné z: <https://all3dp.com/2/raspberry-pi-vs-jetson-nano-differences/>
- [27] Jetson Nano Developer Kit. Nvidia.com: developer [online]. [cit. 2022-12-13]. Dostupné z: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>

- [28] PRASANNA, Shashank. A complete guide to AI accelerators for deep learning inference: GPUs, AWS Inferentia and Amazon Elastic Inference. Towardsdatascience.com [online]. 21. 10. 2020 [cit. 2023-03-31]. Dostupné z: <https://towardsdatascience.com/a-complete-guide-to-ai-accelerators-for-deep-learning-inference-gpus-aws-inferentia-and-amazon-7a5d6804ef1c>
- [29] Intel® Neural Compute Stick 2. Intel.com [online]. [cit. 2023-03-31]. Dostupné z: <https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html>
- [30] Micro:bit a sonar. Robodoupe.cz: web nejen o robotice [online]. 6. 6. 2017 [cit. 2022-12-13]. Dostupné z: <http://robodoupe.cz/2018/microbit-a-sonar/>
- [31] Co je IR senzor: Schéma zapojení a jeho fungování. Cs.jf-parede.pt: elektrický [online]. [cit. 2022-12-13]. Dostupné z: <https://cs.jf-parede.pt/what-is-an-ir-sensor>
- [32] KLUSOŇ, Martin. Rozpoznávání karet v systému Android [online]. Brno, 2015 [cit. 2023-03-31]. Dostupné z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=115297](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=115297).  
Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Alena Pavelková.
- [33] PEŠA, Jan. Obličejový anonymizér [online]. Brno, 2009 [cit. 2023-03-31]. Dostupné z: <https://core.ac.uk/download/pdf/44390528.pdf>. Bakalářská práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Aleš Láník.
- [34] VIOLA, Paul a Michael JONES. Rapid Object Detection using a Boosted Cascade of Simple Features [online]. Cambridge, 2001 [cit. 2023-04-01]. Dostupné z: <https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf>
- [35] O' MAHONY, Niall. Deep Learning vs. Traditional Computer Vision [online]. IMaR Technology Gateway, Institute of Technology Tralee, Tralee, Ireland [cit. 2023-04-01]. Dostupné z: <https://arxiv.org/ftp/arxiv/papers/1910/1910.13796.pdf>

- [36] LEE, Andy. Comparing Deep Neural Networks and Traditional Vision Algorithms in Mobile Robotics [online]. Swarthmore College [cit. 2023-04-01]. Dostupné z: <https://www.cs.swarthmore.edu/~meeden/cs81/f15/papers/Andy.pdf>
- [37] WebSocket Vs HTTP: 6 Unique Differences & Use Cases. Proxyscrape.com [online]. proxyscrape, 2. 12. 2022 [cit. 2023-02-04]. Dostupné z: <https://proxyscrape.com/blog/websocket-vs-http>
- [38] HTTP - Overview. Tutorialspoint.com: HTTP Tutorial [online]. [cit. 2023-02-04]. Dostupné z: [https://www.tutorialspoint.com/http/http\\_overview.htm](https://www.tutorialspoint.com/http/http_overview.htm)
- [39] What is web socket and how it is different from the HTTP?. Geeksforgeeks.org [online]. 21.2.2022 [cit. 2023-02-04]. Dostupné z: <https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/>
- [40] VALÁŠEK, Michal. Stavové HTTP: jak fungují Cookies, Session a ViewState a proč je nepoužívat. Altair.blog [online]. 20. 3. 2008 [cit. 2023-02-04]. Dostupné z: <https://www.altair.blog/2008/03/stavove-http-jak-funguji-cookies-session-a-viewstate-a-proc-je-nepouzivat>
- [41] BOUŠKA, Petr. TCP/IP - navázání a ukončení spojení. Samuraj-cz.com [online]. 13. 9. 2007 [cit. 2023-02-04]. Dostupné z: <https://www.samuraj-cz.com/clanek/tcpip-navazani-a-ukonceni-spojeni/>
- [42] What is a TCP 3-way handshake process?. Afteracademy.com [online]. 10. 2. 2020 [cit. 2023-02-04]. Dostupné z: <https://afteracademy.com/blog/what-is-a-tcp-3-way-handshake-process/>
- [43] JONES, Rowena a Grégorie DE TURCKHEIM. IoT Hub: What Use Case for WebSockets?. Scaleway.com: blog [online]. 22. 2. 2021 [cit. 2023-02-06]. Dostupné z: <https://blog.scaleway.com/iot-hub-what-use-case-for-websockets/>
- [44] Real-time communication for the web. Webrtc.org: home [online]. [cit. 2023-04-01]. Dostupné z: <https://webrtc.org>

- [45] Základy: co potřebujete vědět o WebRTC. Freeconference.com [online]. [cit. 2023-04-01]. Dostupné z: <https://www.freeconference.com/cs/blog/the-basics-what-you-need-to-know-about-webrtc/>
- [46] WebRTC API. Developer.mozilla.org [online]. [cit. 2023-04-01]. Dostupné z: [https://developer.mozilla.org/en-US/docs/Web/API/WebRTC\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API)
- [47] WebRTC: Real-Time Communication in Browsers. W3.org [online]. 6. 3. 2023 [cit. 2023-04-01]. Dostupné z: <https://www.w3.org/TR/webrtc/>
- [48] WHAT IS THE DIFFERENCE BETWEEN MJPEG AND H.264?. Angelcam.com: TIP & TRICKS [online]. 29. 8. 2015 [cit. 2022-12-05]. Dostupné z: <https://blog.angelcam.com/what-is-the-difference-between-mjpeg-and-h-264/>
- [49] Standardy komprese videa. Netcam.cz: Encyklopedie síťového videa [online]. [cit. 2022-12-05]. Dostupné z: <https://netcam.cz/encyklopedie-ip-zabezpeceni/standardy-kompresie-videa.php>
- [50] Jaký je rozdíl mezi kompresí videa H.265 a H.264?. Slkamery.cz [online]. 1. 1. 2018 [cit. 2022-12-12]. Dostupné z: <https://www.slkamery.cz/Jaky-je-rozdil-mezi-kompresi-videa-H-265-a-H-264-b3359.htm>
- [51] Přesměrování portu. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 8. 8. 2021 [cit. 2023-04-01]. Dostupné z: [https://cs.wikipedia.org/wiki/P%C5%99esm%C4%9Brov%C3%A1n%C3%AD\\_portu](https://cs.wikipedia.org/wiki/P%C5%99esm%C4%9Brov%C3%A1n%C3%AD_portu)
- [52] Dynamic DNS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 10. 11. 2022 [cit. 2023-04-01]. Dostupné z: [https://en.wikipedia.org/wiki/Dynamic\\_DNS](https://en.wikipedia.org/wiki/Dynamic_DNS)
- [53] FITZPATRICK, Jason. How to Easily Access Your Home Network From Anywhere. Howtogeek.com [online]. 3. 2. 2023 [cit. 2023-04-01]. Dostupné z: <https://www.howtogeek.com/66438/how-to-easily-access-your-home-network-from-anywhere/>
- [54] DELGADO, Camilo. Best Raspberry Pi SD Card in 2022: Get the most out of your Raspberry Pi with some of the best SD Cards. Pcguide.com: Raspberry Pi [online]. 13. 12. 2022 [cit. 2022-12-13]. Dostupné z: <https://www.pcguides.com/raspberry-pi/guide/best-sd-card/>

- [55] Track chassis tank intelligent diy climbing car robot toy 3D printing upgrade accessories TT motor version. Aliexpress.com: AliExpress SINONING Official Store [online]. [cit. 2023-01-14]. Dostupné z: [https://www.aliexpress.com/item/1005003906186142.html?spm=a2g0o.order\\_list.order\\_list\\_main.25.624c18026tD6M5](https://www.aliexpress.com/item/1005003906186142.html?spm=a2g0o.order_list.order_list_main.25.624c18026tD6M5)
- [56] RPi Motor Driver Board. Waveshare.com [online]. [cit. 2023-01-17]. Dostupné z: <https://www.waveshare.com/rpi-motor-driver-board.htm>
- [57] ČÍŽEK, Jakub. Špionážní tank s nočním viděním. Zive.cz: Pojdme programovat elektroniku [online]. 1. 4. 2018 [cit. 2023-01-14]. Dostupné z: <https://www.zive.cz/clanky/pojdme-programovat-elektroniku-spionazni-tank-s-nocnim-videnim/sc-3-a-192485/default.aspx>
- [58] Raspberry Pi 4 Night Vision Fisheye Camera 5MP OV5647 70/95/160 Degree Focal Adjustable Camera for Raspberry Pi 3 Model B Plus. Aliexpress.com [online]. [cit. 2023-04-02]. Dostupné z: [https://www.aliexpress.com/item/1005003187542470.html?spm=a2g0o.order\\_list.order\\_list\\_main.46.678a1802bXBxUm](https://www.aliexpress.com/item/1005003187542470.html?spm=a2g0o.order_list.order_list_main.46.678a1802bXBxUm)
- [59] Lehká otočná základna pro kameru, bez serva. Laskakit.cz [online]. [cit. 2023-04-02]. Dostupné z: <https://www.laskakit.cz/lehka-otocna-zakladna-pro-kameru-bez-serva/>
- [60] Plastové micro servo SG90 9g, 180°. Laskakit.cz [online]. [cit. 2023-04-02]. Dostupné z: <https://www.laskakit.cz/plastove-micro-servo-sg90-9g--180/>
- [61] POUNDER, Les. Tooling Tuesday: HCSR04P Ultrasonic Sensor. Bigl.es [online]. 25. 8. 2018 [cit. 2023-02-08]. Dostupné z: <https://bigl.es/tooling-tuesday-hcsr04p-ultrasonic-sensor/>
- [62] Reproduktor 2W 8Ω 18x53x15mm Stereo. Laskakit.cz [online]. [cit. 2023-04-02]. Dostupné z: <https://www.laskakit.cz/reproduktor-2w-8-18x53x15mm-stereo/>
- [63] 2x3W digitální audio zesilovač PAM8403. Laskakit.cz [online]. [cit. 2023-04-02]. Dostupné z: <https://www.laskakit.cz/2x3w-digitalni-audio-zesilovac-pam8403/>
- [64] Mini USB mikrofon. Rpishop.cz [online]. [cit. 2023-04-02]. Dostupné z: <https://rpishop.cz/mikrofony/2678-mini-usb-mikrofon.html>



- [65] MERKUR 3 stavebnice. Svet-stavebnice.cz [online]. [cit. 2023-04-02]. Dostupné z: [https://www.svet-stavebnice.cz/cs/velke-stavebnice-merkur/96-merkur-m3-stavebnice.html?gclid=Cj0KCQiAutyfBhCMARIsAMgcRJR PonoxeVIVCLCWBF3AupyK0yKlb50GYzYDWra6O4ciroTYtv2yw2waAv-yEALw\\_wcB](https://www.svet-stavebnice.cz/cs/velke-stavebnice-merkur/96-merkur-m3-stavebnice.html?gclid=Cj0KCQiAutyfBhCMARIsAMgcRJR PonoxeVIVCLCWBF3AupyK0yKlb50GYzYDWra6O4ciroTYtv2yw2waAv-yEALw_wcB)
- [66] 18650 battery pack 11.1V. Techfun.sk [online]. [cit. 2023-04-02]. Dostupné z: [https://techfun.sk/cs/produkt/18650-battery-pack-11-1v/?lang=cs&currency=CZK&gclid=Cj0KCQiAutyfBhCMARIsAMgcRJSJSIh2\\_x2-zhZz4zWj8olRC-MpugkneRgOFCjV2FEuowiYkqL9jp4aAmL0EALw\\_wcB](https://techfun.sk/cs/produkt/18650-battery-pack-11-1v/?lang=cs&currency=CZK&gclid=Cj0KCQiAutyfBhCMARIsAMgcRJSJSIh2_x2-zhZz4zWj8olRC-MpugkneRgOFCjV2FEuowiYkqL9jp4aAmL0EALw_wcB)
- [67] Taidacent 12V 2A High Power 8mm to 18mm Wireless Charging Mat Module DIY Inductive Modules Wireless Charger Module Power Supply Kit Receiver and Transmitter. Amazon.com: electronics [online]. [cit. 2023-04-02]. Dostupné z: [https://www.amazon.com/gp/product/B076GZ59HR/ref=ppx\\_od\\_dt\\_b\\_asin\\_title\\_s0?ie=UTF8&psc=1](https://www.amazon.com/gp/product/B076GZ59HR/ref=ppx_od_dt_b_asin_title_s0?ie=UTF8&psc=1)
- [68] Modul s INA226 pro měření napětí, proudu a výkonu s nastavitelným alarmem. Pajenicko.cz: senzory [online]. [cit. 2023-04-02]. Dostupné z: [https://pajenicko.cz/komponenty/senzory/ostatni-senzory/modul-s-ina226-pro-mereni-napeti-proudu-a-vykonu-s-nastavitelnym-alarmem?limit=25&gclid=Cj0KCQjwz6ShBhCMARIsAH9A0qX8WKvIcI3eISVB\\_V\\_YGynG6GBBADUmh2UoQUI78I9NEjTrkcB9RIGYaAiSYEALw\\_wcB](https://pajenicko.cz/komponenty/senzory/ostatni-senzory/modul-s-ina226-pro-mereni-napeti-proudu-a-vykonu-s-nastavitelnym-alarmem?limit=25&gclid=Cj0KCQjwz6ShBhCMARIsAH9A0qX8WKvIcI3eISVB_V_YGynG6GBBADUmh2UoQUI78I9NEjTrkcB9RIGYaAiSYEALw_wcB)
- [69] Raspberry Pi hardware: GPIO and the 40-pin Header. Raspberrypi.com: Raspberry Pi Documentation [online]. [cit. 2023-01-17]. Dostupné z: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>
- [70] Raspberry Pi OS. Raspberrypi.com: software [online]. [cit. 2023-01-17]. Dostupné z: <https://www.raspberrypi.com/software/>
- [71] UV4L. Linux-projects.org: (advanced) Projects [online]. [cit. 2023-04-02]. Dostupné z: <https://www.linux-projects.org/uv4l/>
- [72] A Robot for real-time object detection with accelerated TensorFlow models, tracking and WebRTC streaming. Linux-projects.org: (advanced) Projects [online]. [cit. 2023-04-02]. Dostupné z: <https://www.linux-projects.org/uv4l/tutorials/video-tracking-with-tensor-flow/>

### III. Seznam obrázků

Obrázek 1: 4-kolový podvozek.....	7
Obrázek 3: 2-kolový podvozek.....	7
Obrázek 2: 6-kolový podvozek.....	7
Obrázek 4: Kolové podvozky [2].....	8
Obrázek 5: Kolové podvozky (2. část)[2].....	8
Obrázek 6: Pásový podvozek [8].....	8
Obrázek 7: Pásový podvozek [7].....	8
Obrázek 8: Chodící roboti [1].....	9
Obrázek 9: způsoby řízení [10].....	9
Obrázek 10: Modelářský servomotor [11].....	10
Obrázek 11: Integrovaný obvod L293 H-můstek [15].....	11
Obrázek 12: Řízení motorů přes rozšiřující desku [14].....	11
Obrázek 13: PWM regulace [18].....	12
Obrázek 14: Princip indukčního nabíjení [21].....	13
Obrázek 15: Raspberry Pi 4 [25].....	15
Obrázek 16: NVIDIA Jetson Nano [27].....	16
Obrázek 17: USB akcelerátor Intel Neural Compute Stick [29].....	16
Obrázek 18: Princip ultrazvukového senzoru [30].....	17
Obrázek 19: Kaskáda klasifikátorů [33].....	19
Obrázek 20: Haarovy příznaky [32].....	19
Obrázek 21: HTTP komunikace [39].....	21
Obrázek 22: TCP Handshake [42].....	22
Obrázek 23: WebSocket komunikace [39].....	23
Obrázek 24: Srovnání HTTP a WebSocketu [43].....	23
Obrázek 25: WebRTC [44].....	24
Obrázek 26: MJPEG sekvence [49].....	25
Obrázek 27: MPEG přenos [49].....	25
Obrázek 28: SanDisk Extreme SD [54].....	31
Obrázek 29: Použitý podvozek s motory [55].....	31
Obrázek 30: Waveshare RPi Motor Driver Board [57].....	32
Obrázek 31: Kamera s nočním viděním [57].....	32
Obrázek 32: Otočná základna pro kameru [59].....	33

Obrázek 33: Servomotory [60] .....	33
Obrázek 34: Ultrazvukový dálkoměr HC-SR04 [61] .....	33
Obrázek 35: Reproduktory [62] .....	34
Obrázek 36: Audio zesilovač PAM8403 [63].....	34
Obrázek 37: USB Mikrofon [64] .....	34
Obrázek 38: stavebnice Merkur 3 BIG SET [65] .....	35
Obrázek 39: Akumulátor [66].....	35
Obrázek 40: Indukční nabíječka Taidacent [67].....	36
Obrázek 41: Modul pro měření napětí INA226 [68] .....	36
Obrázek 42: Číslování GPIO pinů Raspberry Pi [64].....	37
Obrázek 43: Waveshare RPi Motor Driver Board [56] .....	38
Obrázek 44: Schéma zapojení motorů .....	39
Obrázek 45: Zapojení dálkoměru [61].....	40
Obrázek 46: Zapojení servomotorů .....	41
Obrázek 47: Zapojení zesilovače a reproduktorů [63].....	42
Obrázek 48: Zapojení modulu pro měření napětí .....	43
Obrázek 49: Architektura řešení .....	46
Obrázek 50: Python aplikace .....	47
Obrázek 51: Adresy I2C sběrnice .....	53
Obrázek 52: Připojený gamepad .....	60
Obrázek 53: umístění gamepadu.....	60
Obrázek 54: UV4L Demos [71].....	61
Obrázek 55: Detekce objektů.....	64
Obrázek 56: Detekce obličeje .....	64
Obrázek 57: Uživatelské rozhraní.....	65

## IV. Seznam tabulek

Tabulka 1: Specifikace RPi 4 [25].....	30
Tabulka 2: Zapojení ultrazvuku .....	40
Tabulka 3: Zapojení voltmetru .....	43
Tabulka 4: Ovládání motoru .....	48

## **V. Seznam příloh**

- a. Zdrojové kódy**
- b. Fotografie zařízení**

## **VI. Přílohy**

### **Příloha A**

#### **Zdrojové kódy**

Kompletní zdrojové kódy jsou umístěny ve veřejném repozitáři na GitHubu.

Aplikace:

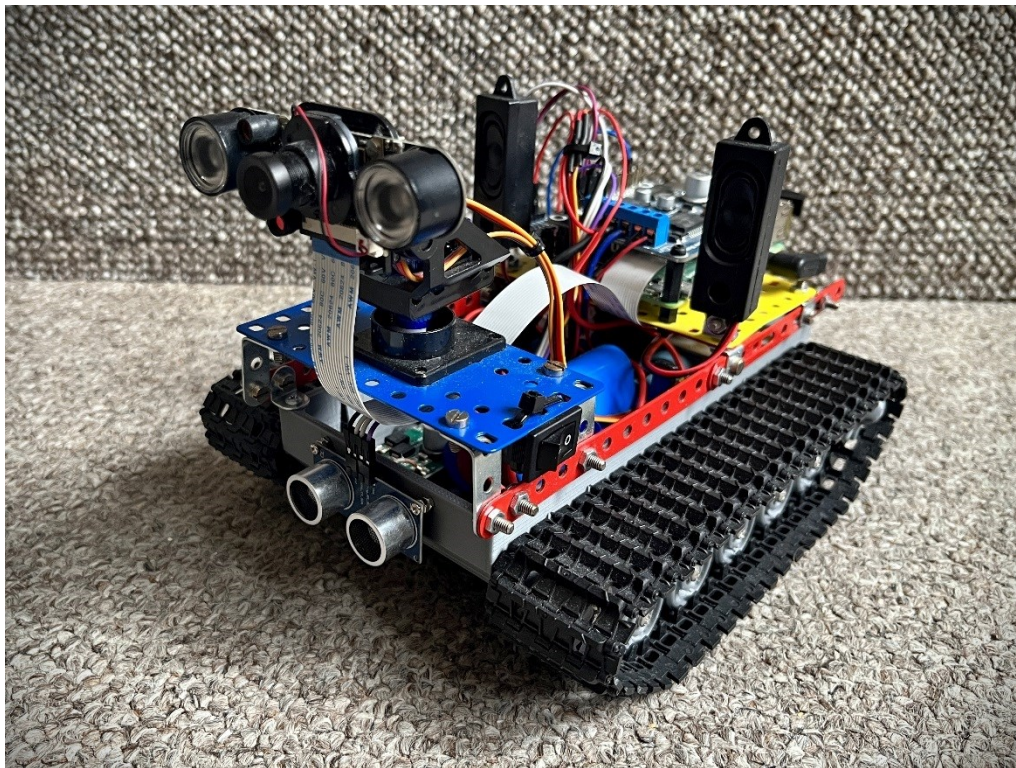
<https://github.com/stropak/robot>

Uživatelské rozhraní:

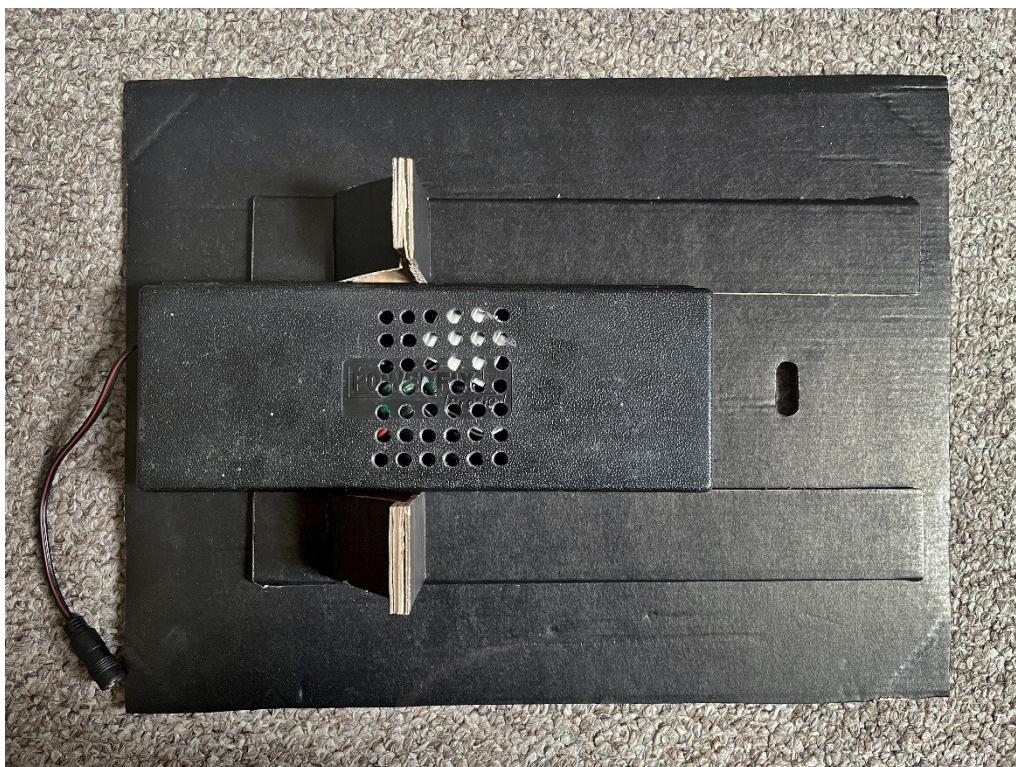
<https://github.com/stropak/robot/tree/main/User%20interface>

## Příloha B

### Fotografie zařízení



*Robot*



*Nabíjecí stanice*