Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Diplomová práce

Návrh a implementace autonomního stroje ke kontrole/ochraně budov

Bc. Josef Kolář

© 2020 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Josef Kolář

Systémové inženýrství a informatika
Informatika

Název práce

Návrh a implementace autonomního stroje ke kontrole/ochraně budov

Název anglicky

Design and implementation of autonomous machine for building inspection/protection

---

Cíle práce

Cílem práce je navrhnout a zkonstruovat prototyp (demonstrátor) autonomního stroje, který bude automaticky schopen pohybu v okolí budov, snímání obrazu budov a okolí.

Metodika

Navrhněte autonomní stroj, který bude schopen následovat uživatelem zvolenou trasu (alespoň 4 otočné body).

1) Navrhněte mechanismus vyhnutí se překážce na cestě, a to v denním i nočním provozu

2) Vhodný navigační systém (GPS či jiný)

3) Automatický návrat na místo startu (pro budoucí účel nabíjení)

4) Mechanismus přenosu obrazu na zvolený server

Řešení otestujte ve zvolené lokalitě.

Doporučený rozsah práce

60-80

Klíčová slova

Autonomní stroj, Navigace, Robot, Přenos obrazu, Navigační systém, GPS, Otočný bod, Překážka

---

Doporučené zdroje informací

NOVÁK, P. *Mobilní roboty : pohony, senzory, řízení.* Praha: BEN – technická literatura, 2005. ISBN 80-7300-141-1.
robotika.cz

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Josef Pavlíček, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 31. 3. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 31. 3. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 02. 04. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Návrh a implementace autonomního stroje ke kontrole/ochraně budov" jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor(ka) uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 6. 4. 2020 _____

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce Ing. Josefu Pavlíčkovi, Ph.D. za odborné konzultace, užitečné rady a motivující přístup. Také chci poděkovat svým rodičům, kteří mě vytrvale podporují po dobu mých studií. Děkuji.

Návrh a implementace autonomního stroje ke kontrole/ochraně budov

Abstrakt

Diplomová práce se zabývá tvorbou robotické platformy, schopné autonomního pohybu v okolí budov, které lze zároveň monitorovat. Teoretická část práce je věnována současným řešením, dále se zabývá možnostmi lokalizace a použitelných prostředků k vývoji stroje po hardwarové tak softwarové stránce. Praktická část představuje komponenty a technologie použité při realizaci prototypu, které jsou dále důkladněji popsány. Další kapitoly popisují testování prototypu. Závěr je věnován zhodnocením výsledků a dosaženým výsledkům z pohledu dílčích částí, které jsou rozšířeny o potenciální vylepšení v případě dalšího vývoje.

Klíčová slova: Autonomní stroj, Navigace, Robot, Přenos obrazu, Navigační systém, GPS, Otočný bod, Překážka

Design and implementation of autonomous machine for building inspection/protection

Abstract

This thesis deals with the creation of a robotic platform capable of autonomous movement around buildings that can be monitored at the same time. The theoretical part is devoted to current solutions, then deals with the possibilities of localization and usable means for the development of the machine in hardware and software. The practical part presents the components and technologies used in the prototype implementation, which are described in more detail below. Next chapters are devoted to prototype testing. The conclusion is devoted to the evaluation of results and achieved results in terms of sub-parts, which are extended by potential improvements in case of further development.

Keywords: Autonomous vehicle, Navigation, Robot, Image transmission, Navigation system, GPS, Waypoint, Obstacle

Obsah

1 Úvod **11**

2 Cíl práce a metodika **12**

2.1 Cíl práce 12

2.2 Metodika 12

3 Teoretická východiska **13**

3.1 Příklady současných řešení 14

3.2 Metody určení polohy 17

3.2.1 Úhel příchodu signálu (AoA) 17

3.2.2 Čas příchodu signálu (ToA) 17

3.2.3 Indikátor síly přijímaného signálu (RSSI) 18

3.2.4 Časový rozdíl doletu (TDoA) 18

3.3 Rozdělení technik lokalizace 19

3.3.1 Proximity Detection (Conectivity based positioning) 20

3.3.2 Navigace výpočtem (Dead Reckoning) 20

3.3.3 Map Matching 21

3.3.4 Navigace podle mapy (Map-based navigation) 21

3.3.5 Navigace podle orientačních bodů (Landmark navigation) 21

3.3.6 Odometrie 22

3.4 Senzory pro určení polohy 22

3.4.1 Enkodéry 22

3.4.2 Akcelerometry 24

3.4.3 Gyroskop 24

3.4.4 Ultrazvukové senzory vzdálenosti 24

3.4.5 Optický senzor (Optical Flow Sensor) 24

3.4.6 IR senzory 25

3.4.7 LIDAR 25

3.4.8 Elektronický kompas 26

3.4.9 Globální družicový polohový systém (GNSS) 27

3.4.10 Radiofrekvenční systémy 28

3.4.11 Inerciální měřící jednotka (IMU) 28

3.4.12 RFID (Radio Frekquency Identification) 29

3.4.13 Kamera (Strojové vidění) 29

3.5 Řídící elektronika 29

3.5.1 Jednodeskové počítače 30

3.5.2 Mikrokontroléry ... 30
3.6 Komunikační protokoly ... 30
3.6.1 Formát dat ... 32
3.6.2 Serializace a Deserializace ... 32
3.6.2.1 BSON ... 32
3.6.2.2 MessagePack ... 33
3.6.2.3 ProtocolBuffers ... 33
3.6.2.4 Cap´n Proto ... 33
3.7 Frameworky pro vývoj robotů ... 33
3.8 Autopiloty ... 35
4 Vlastní práce ... 36
4.1 Výběr technologií ... 36
4.1.1 Volba platformy ... 36
4.1.2 Výběr navigačního systému ... 37
4.1.3 Typ podvozku ... 38
4.2 Použitý HW ... 39
4.2.1 Podvozek ... 39
4.2.2 Motory ... 39
4.2.3 Řízení motorů ... 40
4.2.4 Kola ... 40
4.2.5 Napájení ... 41
4.2.6 Počítač ... 41
4.2.7 Mikrokontroléry – Arduino ... 42
4.2.8 Kamera ... 43
4.2.9 Kamerový držák ... 43
4.2.10 Kompas ... 44
4.2.11 GNSS ... 44
4.2.12 Ultrazvukové dálkoměry ... 45
4.3 Použitý SW ... 45
4.3.1 Komunikace ... 45
4.3.2 Komunikační knihovna pro mikrokontroléry ... 46
4.3.3 Hlavní program ... 46
4.3.4 Webová aplikace ... 49
4.3.5 JavaScript ... 49
4.3.6 Mikrokontroléry ... 50
4.4 Popis jednotlivých subsystémů ... 50

4.4.1 Systém detekce překážek ... 51
4.4.2 Systém magnetometru a řízení serv ... 51
4.4.3 Systém řízení motorů ... 51
4.4.4 Systém GNSS ... 52
4.4.5 Hlavní počítač ... 53
4.4.6 Webové rozhraní ... 53
4.5 Testování ... 55
4.5.1 Porovnání GNSS modulů ... 55
4.5.2 Testování 1. trasy ... 57
4.5.3 Testování 2. trasy ... 58
4.5.4 Testování kamery ... 60
4.5.5 Testování vyhýbání se překážkám ... 60
5 Výsledky a diskuse ... 61
5.1 Konstrukce ... 62
5.2 Elektronika a senzory ... 62
5.3 Hlavní program ... 63
5.4 Webová aplikace ... 64
5.5 Webové rozhraní ... 64
6 Závěr ... 66
7 Seznam použitých zdrojů ... 68
8 Seznam obrázků ... 74
9 Přílohy ... 76

1 Úvod

Dnešní doba je plná robotických zařízení, která zastávají stále složitější funkce. Objevují se častěji a častěji v různých oblastech lidského snažení jako je automobilový průmysl, letectví, zemědělství nebo medicína. Zastávají práci na linkách jako robotické manipulátory nebo CNC stroje. Nejrůznější druhy kolaborativních robotů pracují doslova ruku v ruce s lidmi. V dnešní době se můžeme setkat i s multirobotickými systémy, které musí kromě své úlohy řešit ještě vzájemnou koordinaci a kooperaci.

Tato práce se zabývá problematikou venkovního mobilního robotu sloužícího ke kontrole budov. Tato aplikace zahrnuje výběr vhodné platformy a jejích komponent jak po hardwarové, tak softwarové stránce. Konstrukce zajistí pohyb robotu, senzory musí být schopny navigace v požadovaném prostoru. Musejí poskytnout co možná nejlepší odhad polohy a orientace stroje v prostoru, ale také polohy potenciálních překážek. Softwarová stránka by měla zajišťovat optimální přenos dat napříč celým systémem, rovněž vhodné zpracování jednotlivých dat a zajištění potřebných funkcionalit, které má robot nabídnout.

Velká část teoretické části je věnována samotným možnostem lokalizace. To zahrnuje principy měření, příklady technik lokalizace a také senzory, které lokalizaci umožňují. Závěr teoretické části je věnován řídícím systémům jak po stránce hardwarové, tak softwarové a s tím souvisejícími komunikačními protokoly. Úvod praktické části obsahuje výběr technologií, které jsou následně detailněji popsány. Další část je věnována popisu jednotlivých vytvořených subsystémů. Navazující kapitola popisuje testování výsledného řešení jak z hlediska navigačních možností, tak pořizování snímků z otočných bodů. Kapitolu uzavírá testování překážek. Závěr praktické části v souhrnu popisuje dosažených výsledků a předkládá také výsledky a budoucí návrhy na zlepšení dle dílčích částí řešení.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je navrhnout a zkonstruovat prototyp (demonstrátor) autonomního stroje, který bude automaticky schopen pohybu v okolí budov, snímání obrazu budov a okolí.

2.2 Metodika

Navrhněte autonomní stroj, který bude schopen následovat uživatelem zvolenou trasu (alespoň 4 otočné body).

1) Navrhněte mechanismus vyhnutí se překážce na cestě, a to v denním i nočním provozu
2) Vhodný navigační systém (GPS či jiný)
3) Automatický návrat na místo startu (pro budoucí účel nabíjení)
4) Mechanismus přenosu obrazu na zvolený server

Řešení otestujte ve zvolené lokalitě.

3 Teoretická východiska

Robotů schopných autonomního provozu je celá řada. Ať již jde o profesionální roboty nebo hobby projekty, jejich množství a reálná využitelnost roste. Existuje celá řada různorodých řešení a přístupů k realizaci. Ať už však jde o bezpilotní letadlo nebo pozemní robot, který má vykonávat autonomně nějakou činnost, musí mít stroj implementovanou určitou formu orientace v prostředí, ve kterém se pohybuje. To primárně můžeme dělit na prostředí vně a uvnitř budov. Důležité pak taky je, zda se bude robot pohybovat po zemi, ve vzduchu, případně ve vodě. Na tomto základě pak zpravidla můžeme rozhodnout, které metody navigace využít a které ne. Například nepoužijeme GPS navigaci uvnitř budov. Vzhledem k typu práce budeme tedy dále uvažovat techniky pro venkovní prostředí. Typický příklad pro lokalizaci ve venkovním prostředí je GNSS v kombinaci s kompasem, mezi další nejčastěji používané senzory se řadí inerciální jednotka (IMU), Lidar, ultrazvukové senzory, IR senzory či optické senzory. Tyto senzory se zpravidla používají ať jde o dron či pozemní robot. Velmi důležitým aspektem je požadovaná přesnost navigačního systému. Máme-li vyřešenou lokalizaci a plánování, můžeme se posunout ke konkrétní aplikaci. Existují řešení, která mapují prostředí, na jehož základě se pak orientují. Jde o metodu SLAM (Simultaneous localization and mapping), kterou využívají například robotické vysavače.

3.1 Příklady současných řešení

Následující řešení jsou příkladem profesionálních robotů a jejich různorodých nasazení, velký důraz je věnován hlavně použitým komponentám k určení polohy, případně orientace v prostoru.

Husky od firmy Clearpath Robotics je jedna z mnoha robotických platforem, které firma nabízí. Husky platforma je nabízena jak samostatně, tak v rámci balíčků, které obsahují příslušenství umožňující konkrétní aplikaci. Je tedy k dispozici například „explorer package“ který obsahuje robotickou platformu s IMU, GPS a Lidar. Tato konfigurace propůjčuje platformě možnost lokalizace, mapování, a vyhýbání se překážkám. Za zmínku také stojí, že všechny platformy obsahují počítače, na kterých běží ROS (Robot Operating System), který bude představen dále. (1)

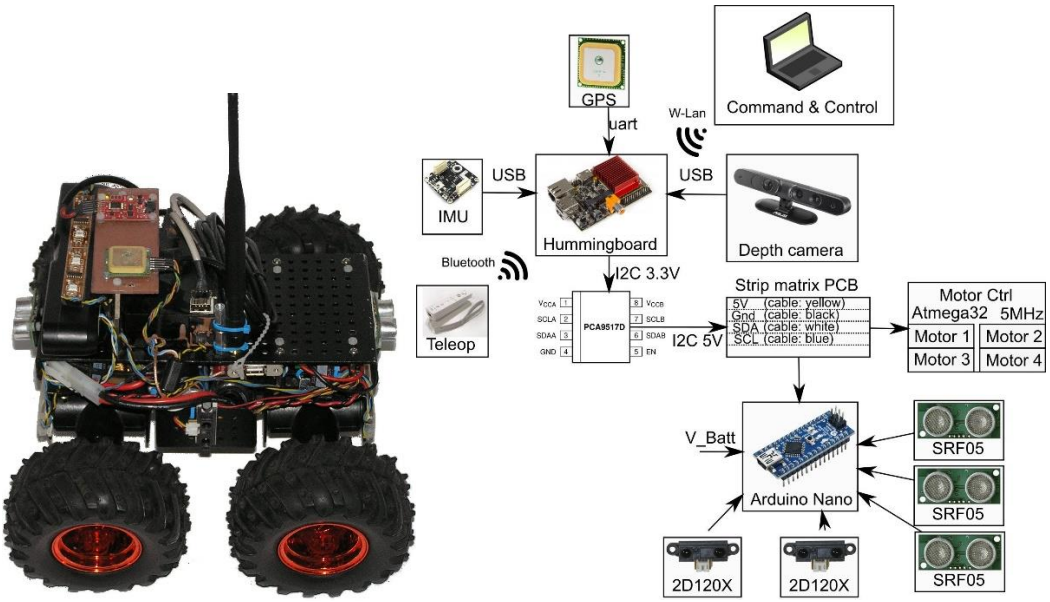
Obrázek 1- Husky (1)

Firma SMP Robotics nabízí velké množství aplikací od hlídkovacích robotů přes roboty detekující plyn, plašiče ptáků, transportní roboty až po platformu umožňující vzlet a přistání dronu. Robotická platforma je založena na čtyřkolovém podvozku, na které jsou následně umístěny komponenty pro danou aplikaci. O navigaci v prostředí se stará kombinace vizuální navigace, GPS a IMU. Toto řešení neobsahuje žádný nákladný Lidar, ten je ovšem nahrazen stereovizí a algoritmy hlubokého učení. To si žádá vysoký výkon na zpracování a know-how. (2)

Obrázek 2 - SMP robot (2)

Následující řešení jsou z dílen nadšenců, tato řešení mohou být dobrou inspirací, zpravidla nedosahují parametrů profesionálních robotů, ale demonstrují použití různorodých komponent pro splnění určitých aplikací. Velkou výhodou je, že velká většina použitých komponent je finančně dostupná.

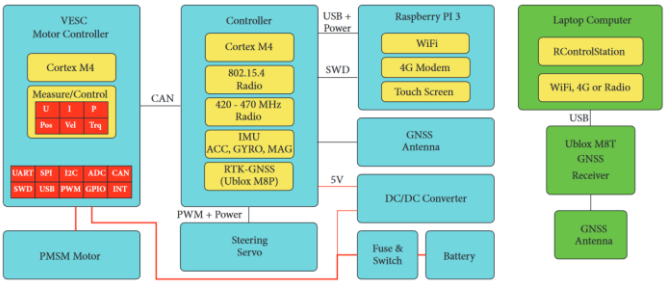
Zajímavou ukázkou je robot s názvem Wild Thumper, jehož cílem je možnost navigace uvnitř budov pomocí SLAM, venku pomocí GPS. Co se týče senzorického vybavení, robot obsahuje GPS, IMU, stereokameru, ultrazvukové senzory, optické senzory vzdálenosti a enkodéry. Robot je postaven na systému ROS. (3)



Obrázek 3 - Wild Thumper schéma (3)

Obrázek 4 - Wild Thumper (3)

Další ukázkou je platforma SDVP (Self-Driving Model Vehicle Platform) vytvořená Švédským institutem, respektive HW prototyp, který pro svoji lokalizaci využívá RTK-GNSS a fúzi senzorů z odometrie a IMU. (4)



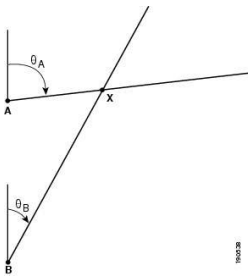
Obrázek 5 - SDVP schéma testovací platformy (4)

3.2 Metody určení polohy

Nejčastějším řešením určení polohy je aplikace Trilateračních nebo Triangulačních algoritmů. Triangulace vyžaduje minimálně dvě stanice, které jsou schopné zachytit příchozí signál od objektu a určit z jakého úhlu signál přichází. Trilaterace vyžaduje minimálně tři stanice se známou polohou, které aktivně vysílají signál. Zpracováním těchto signálů, respektive výpočtem vzdáleností, lze určit polohu objektu. (5) (6)

3.2.1 Úhel příchodu signálu (AoA)

Metoda AoA (Angle of Arrival) nebo také DoA (Direction of Arrival) spočívá ve výpočtu úhlu, v němž přichází signál ke stanici. Přijmou-li příchozí signál alespoň dvě stanice se známou pozicí, můžeme s relativně dobrou přesností určit, odkud signál přichází. V případě detekce signálu více stanicemi dojde k přesnějšímu určení pozice. Úskalím této technologie je pak relativně komplikovaný anténní systém stanic. (7)

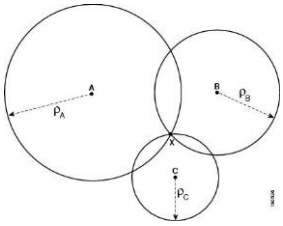


Obrázek 6 - AoA metoda (7)

3.2.2 Čas příchodu signálu (ToA)

Metoda ToA (Time of Arrival) nebo také ToF (Time of Flight) je jedna z jednodušších technik lokalizace, při které se využívá přesného času vyslání signálu a přesného času jeho příjmu. Známe-li rychlost šíření signálu a časový rozdíl mezi jeho vysláním a příjmem, lze relativně přesně určit, jak daleko je vysílač od přijímače. Přesnost metody záleží na použitém zařízení, a to nejen na jeho schopnosti určení samotného času, ale také na schopnosti zařízení změřit přesný čas vyslání a detekce příjmu signálu. Metoda lokalizace je opět založena na zpracování signálů z více zdrojů, kde více zdrojů zpravidla zpřesní určení polohy. Tato metoda není pouze doménou RF zařízení, ale používá se například v ultrazvukových nebo Lidar senzorech.

Důležitou poznámkou pro tyto systémy je existence LoS (Line of sign problem). Pro metody založené na době letu signálu mezi vysílačem a přijímačem (kdy se využívá přímých radiových vln) je ideální přímá viditelnost mezi anténou vysílače a přijímače spolu s volnou Fresnelovou zónou. Vysílané vlny se v tomto případě dostanou přímou cestou bez odrazů k přijímači a výsledný čas doby letu pak skutečně odpovídá vzdálenosti mezi vysílačem a přijímačem, resp. jejich anténami. V ostatních případech může dojít k odrazu vlny od překážky, což v důsledku prodlouží naměřený čas a dojde tedy k určité větší či menší chybě v určení vzdálenosti. Velmi častým problémem je vícecestné šíření vln. Vysílaná vlna dorazí jak v přímém, tak i v odraženém směru. Nejčastějším opatřením proti odraženým vlnám je omezení jejich příjmu a filtrace od přímé vlny. Další možností je použití nízko interferenčních signálů. (7) (8)



Obrázek 7 - ToA metoda (7)

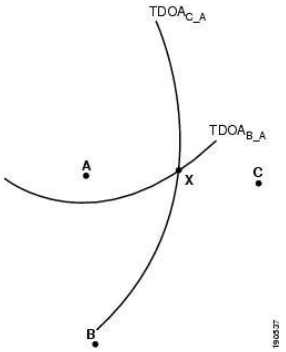
3.2.3 Indikátor síly přijímaného signálu (RSSI)

Metoda RSSI (Received Signal Strength Indicator) sloužící k přibližné lokalizaci funguje na principu měření síly signálu. Vyhodnocením síly signálu ze tří a více bodů můžeme vypočítat přibližnou polohu. Jde o velmi jednoduchou metodu, která ovšem nepřináší takovou přesnost, a navíc bývá zatížena chybou v důsledku prostředí. Může jít o šum nebo objekty mezi stanicí, které mohou utlumit signál a měření je zkreslené. (7)

3.2.4 Časový rozdíl doletu (TDoA)

Metoda TDoA (Time difference of arrival) založená na rozdílu časů příchodu signálů, která je především doménou lokalizace pomocí radiofrekvenčních vln. Za zmínku jistě stojí historické pozadí této technologie, které je úzce spjato s Československem. Prvním ze systémů je Britský systém GEE, později se objevil Americký systém LORAN, oba systémy jsou z dob 2. světové války. LORAN pracuje na principu zachycení signálů ze tří pevných vysílačů, které jsou synchronizovány. Tyto signály jsou zachyceny a zpracovány

přijímačem, který vždy ze dvojice signálů vypočítá hyperbolu. Na základě vypočtených hyperbol, hlavně pak jejich průsečík, nám jednoznačně identifikuje polohu přijímače. Koncem padesátých let byla právě v Československu vynalezena takzvaná časově-hyperbolická metoda (za kterou stojí doc. Ing. Vlastimil Pech, Csc.) postavená na třech pozemních stanicích a vyhodnocovacím stanovišti. Jde v podstatě o obrácený princip LORANu, přijímací stanoviště jsou od sebe vzdálena a dojde-li k jejich ozáření signálem, je ve vyhodnocovacím stanovišti vypočtena pozice na základě časových diferencí ozáření jednotlivých přijímacích stanovišť. Teorie byla rychle aplikována do praxe a vznikl první přesný radiotechnický pátrač PRP-1 KOPÁČ (KOrelační PÁtraČ). Následovala RAMONA, TAMARA a VĚRA, která je v různých verzích úspěšně provozována dodnes, a to jak v civilních verzích, tak pro vojenské účely. (7) (9) (10)



Obrázek 8 - TDoA metoda (7)

3.3 Rozdělení technik lokalizace

Lokalizace je jednou z hlavních problematik, kterým robotická aplikace čelí. Určení polohy a orientace robotu je zcela zásadní. Se znalostí polohy a orientace lze plánovat další pohyb. Mezi základní rozdělení technik lokalizace je relativní a absolutní lokalizace.

Relativní lokalizace spočívá v odhadu nové polohy na základě sledování relativního pohybu například z enkodéru nebo inerciální jednotky a předchozí polohy. Nevýhoda je kumulace chyb, protože každý odhad je zatížen nepřesností měření nebo šumem senzorů.

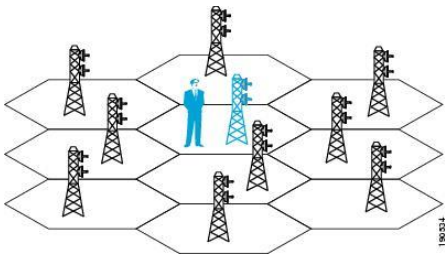
Absolutní lokalizace spočívá v určení polohy jednorázovým měřením bez znalosti předchozí polohy. Typickým příkladem je GPS.

Většina aplikací využívá oba přístupy současně, ve snaze eliminovat nedostatky obou technik. Typický příklad je fúze inerciální jednotky a GPS. Inerciální jednotka dokáže v krátkodobém časovém horizontu udržet informaci o absolutní poloze v případě

krátkodobého výpadku GPS zapříčiněný průjezdem v tunelu. V robotických aplikacích jde pak spíše o rychlost aktualizace dat, kdy inerciální jednotka poskytuje data v krátkých časových intervalech oproti systému GPS. Je tedy možné počítat odhad polohy v kratších intervalech a tím zajistit plynulý pohyb zařízení. (11) (12)

3.3.1 Proximity Detection (Conectivity based positioning)

Jde o jednu z nejjednodušších metod určování polohy, co se týče její implementace. Pozice je určena na principu Cell of Origin (CoO), což je propojený systém buněk. Každá buňka je charakterizována zpravidla poloměrem, ve kterém může komunikovat. Každá z buněk je statická a je tedy známa její poloha. V případě přiblížení sledovaného zařízení do poloměru stanice a navázání spojení, pak v rámci sítě buněk víme, kde se zařízení nachází. Předpokladem je, že zařízení bude vždy navazovat spojení s nejsilnějším zdrojem signálu, protože signály v rámci této sítě se mohou překrývat. Příkladem takového systému je síť pro mobilní telefony GSM. Tato metoda je implementovatelná v bezdrátových technologiích postavených zejména na radiových vlnách jako je Bluetooth, Wi-Fi, RFID, GSM, nebo obecně na RF modulech. Ovšem je klidně možné metodu implementovat v rámci IR. (7)



Obrázek 9 - CoO schéma (7)

3.3.2 Navigace výpočtem (Dead Reckoning)

Jde o metodu odhadu pozice od posledního známého bodu podle odhadnuté rychlosti za uplynulý čas a směru pohybu. Jde o velmi starou metodu, která je i v dnešní době hojně využívána. Její výhoda tkví v její jednoduchosti, avšak velkou nevýhodou je fakt, že s každým dalším odhadem polohy, která je založena pouze na této metodě, se kumuluje chyba měření. (13)

3.3.3 Map Matching

Map matching je metoda, která se využívá na silniční síti. Jde o určení polohy vozidla zpravidla za pomoci zeměpisné šířky a délky a dat o pohybu. Lze tedy zpřesnit polohu díky znalosti silniční sítě. (14)

3.3.4 Navigace podle mapy (Map-based navigation)

Navigace podle mapy je velmi rozšířená metoda. Spočívá v mapování prostředí, ve kterém se robot pohybuje, pomocí senzorů a data ze senzorů jsou transformována do mapové podoby. V základu se mapy dělí na metrické a topologické.

Metrické mapy jsou realizovány jako kartézská soustava souřadnic, která je pro člověka dobře čitelná. Případně do takovéto mapy lze jednoduše zavést další orientační body, například virtuální plot. Nevýhodou je vysoká citlivost na chyby plynoucí z nepřesnosti lokalizace a zaměření vzdálenosti překážek při tvorbě mapy. Mapa také přímo neposkytuje podklad pro navigaci.

Topologická mapa je vlastně grafem. Obsahuje hrany a uzly, kde uzly jsou význačné (orientační) body a hrany představují možné cesty. Tento druh mapy není pro člověka dobře čitelný, ale podporuje možnost navigace (15)

3.3.5 Navigace podle orientačních bodů (Landmark navigation)

Navigace založená na rozpoznání orientačního bodu v prostředí. Orientační bod se může skládat z jedné nebo více charakteristik, které může robot detekovat. Ideální pak je, aby byl bod v daném operačním prostoru unikátní. Může jít o jednoznačný bod v podobě QR kódu, čárového kódu, RFID čipu, tvaru nebo barvě. Bod se také může skládat z více detekovatelných charakteristik, například může jít o specifickou kombinaci vzdáleností a rohu. Tento druh navigace je zpravidla podpořen další metodou určování polohy a detekcí orientačního bodu lze pak provádět patřičné navigační korekce. Samotné body jsou pak dvojího druhu, přirozené a umělé. Přirozené body jsou součástí prostředí, ve kterém se robot pohybuje, ve vnitřním prostředí to mohou být rohy místností, zdi, dveře, okna nebo nábytek.

Ve vnějším prostředí to mohou být stromy, plot či budovy. Mezi umělé body můžeme zařadit QR kódy, čárové kódy, RFID čipy, radiofrekvenční nebo infračervené majáky. Lokalizace touto metodou předpokládá mapu nebo data, podle kterých pak může nasnímanou realitu porovnávat, což může mít za následek vyšší výpočetní náročnost této metody. (16)

3.3.6 Odometrie

Odometrie je metoda založená na odhadu relativní pozice pomocí snímání otáčení kol nebo pohonné jednotky enkodérem. Enkodér je zařízení schopné transformovat otáčivý pohyb na elektrický signál, který je dále zpracovatelný například mikrokontrolérem. Pro aplikaci odometrie je zapotřebí znát typ podvozku, respektive geometrický model. Podle geometrického modelu, umístění enkodéru a jeho typu lze následně stanovit patřičné vztahy pro výpočet pohybu. Hlavní nevýhodou použití odometrie je její kumulativní chybovost s ujetou vzdáleností. Mezi hlavní zdroje chyb patří rozdíl obvodu kol nebo drift. Nutno však dodat, že velmi záleží na typu podvozku, protože aplikací odometrie na nevhodném typu podvozku získáme výsledky zatížené značnou chybou. (17)

3.4 Senzory pro určení polohy

Tato kapitola je věnována zařízením, která se využívají k lokalizaci případně k detekci překážek.

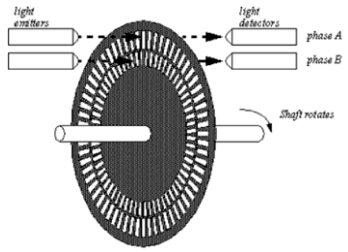
3.4.1 Enkodéry

Enkodéry jsou typické součástky pro aplikaci odometrie. Převádějí otáčení kol na zpracovatelný signál, na jehož základě jsme schopni určit pozici kola (úhel jeho natočení), případně podle patřičných matematických vztahů i ujetou vzdálenost. Mezi nejrozšířenější enkodéry patří optické a magnetické.

Optické enkodéry

- reflexivní (odrazové) - vysílač a přijímač jsou na jedné straně vyplněného disku, které na sobě má reflexní a matné plošky. Zpracováním intenzity ozáření přijímače je možné generovat obdélníkový signál, který je možné dále zpracovat.

- transmisivní - vysílač a přijímač jsou umístěny naproti sobě a mezi nimi je děravý disk.



Obrázek 10 - Optický enkodér (18)

Magnetické enkodéry

Jsou nejčastěji realizovány jako mnohopólové magnety rotující na hřídeli spojené s motorem. Pro čtení je použita sada dvou halových sond nebo pokročilý obvod, který pak přináší jemnější rozlišení.

Obrázek 11 - Magnetický enkodér (19)

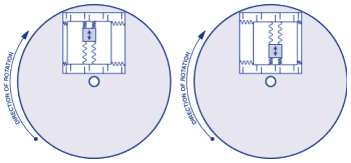
Enkodéry lze také dělit na inkrementální a absolutní. Inkrementální enkodéry jsou schopny zachytit změnu o daný krok, na druhé straně absolutní enkodéry jsou schopny podat informaci o aktuální poloze v rámci 360 stupňů. (20) (19)

3.4.2 Akcelerometry

Akcelerometr slouží k měření zrychlení a lze jej použít k měření úhlu náklonu, setrvačných sil, otřesů atd. Akcelerometr přeměňuje změnu pohybu na elektrický signál. Samotné zrychlení má pak dvě složky, statické a dynamické. Dynamické zrychlení se vytvoří změnou rychlosti zařízení. Statické zrychlení pak vzniká působením zemské gravitace. (21)

3.4.3 Gyroskop

Gyroskop slouží k měření úhlové rychlosti. Tedy je schopen měřit rychlost otáčení ve stupních za sekundu. Jde o měření relativního pohybu nezávislého na gravitaci. Existuje mnoho druhů, jako například mechanické gyroskopy, optoelektrické, laserové a velmi rozšířené MEMS gyroskopy. Gyroskopy využívají principu Coriolisovy síly. MEMS senzor je tvořen periodicky se pohybující strukturou upevněnou na pružinách. Měřený směr pohybu musí být vždy kolmý ke směru otáčející se vnitřní struktury. (22)



Obrázek 12 - princip MEMS gyroskopu (22)

3.4.4 Ultrazvukové senzory vzdálenosti

Slouží k detekci předmětu. Na rozdíl od optických senzorů nemají problém s průhlednými materiály. Jeho možnou nevýhodou je rozptyl vysílaného ultrazvukového paprsku, pohlcování zvuku, nemožnost detekce objektu pod určitým úhlem a relativně pomalé měření. Detekce vzdálenosti od překážky je založená na časové diferenci mezi vysláním ultrazvukového signálu a jeho příjmem, tedy doby letu (ToF). Tento fakt zároveň nastiňuje další z nevýhod, a to závislost na okolní teplotě vzduchu, protože rychlost šíření zvuku je nezanedbatelně ovlivněna právě teplotou. (23)

3.4.5 Optický senzor (Optical Flow Sensor)

Optický senzor je založený na optické navigaci, měřící změny v pozici snímáním povrchu a následném vyhodnocování směru a velikosti pohybu. Senzor snímá nasvícenou oblast, data

ze senzoru jsou zpracovány digitálním signálovým procesorem, který vypočítá diferenci v souřadném systému oproti minulému snímku. Tato diference v kontextu vzdálenosti od snímaného povrchu pak může sloužit k určování relativní pozice. Jde o modifikovaný princip z optických počítačových myší. (24)

Obrázek 13 - Optický senzor (24)

3.4.6 IR senzory

IR neboli infračervené senzory využívají infračervené světlo. Standardní uspořádání je ve formě vysílače a přijímače. Existuje řada použití, nejběžnějším je vysílač na jedné straně a přijímač na druhé, kde pomocí modulovaných dat můžeme odeslat signál, který je na straně přijímače demodulován a data dekódována. Toto schéma můžeme vidět například u ovladače k televizi. Dalším uspořádáním je vysílač a přijímač vedle sebe nebo v jednom pouzdře. Takové schéma pak slouží nejčastěji k detekci bezprostřední překážky nebo detekci změny barvy, nejčastěji černá a bílá, kde je dostatečný rozdíl v odrazu emitovaného paprsku. Dalším uspořádáním může být IR senzor pro měření vzdálenosti. Takový senzor je pak vybaven zdrojem IR záření a snímačem. Odražené světlo dopadá na čočku, která vychyluje paprsek na základě odražené vzdálenosti od objektu na určité místo snímače, který pak díky podpůrným obvodům převede místo dopadu na elektrické napětí, které je přímo úměrné vzdálenosti od objektu. (25)

3.4.7 LIDAR

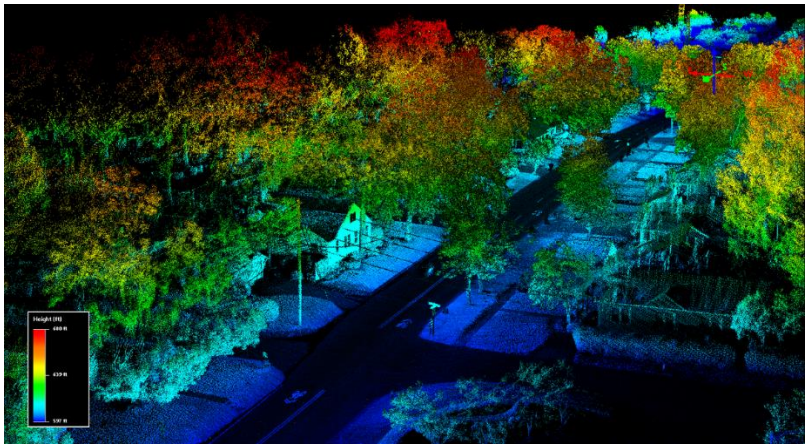
Lidar (Light Detection And Ranging) je zařízení, které slouží k měření vzdáleností pomocí laserového paprsku. Princip činnosti spočívá ve vyslání laserového paprsku, který se odrazí od snímaného objektu zpět, kde jej zachytí detektor. Výpočtem času mezi vysláním paprsku a jeho případným příjmem lze změřit vzdálenost od objektu. Mezi klíčové vlastnosti patří

rychlost a přesnost měření. Stinnou stránkou je pořizovací cena, problematický provoz v dešti nebo mlze a nemožnost detekce průhledného objektu. Velmi často je právě z těchto důvodů Lidar doprovázen ultrazvukovým senzorem.

Tato zařízení jsou již velmi rozšířená a jejich využití stále roste. Mezi příklady uplatnění této technologie je zemědělství, lesnictví, archeologie, automobilový průmysl. Lidar už je v dnešní době běžnou součástí robotických vysavačů.

Lidar systémy, lze rozdělit z pohledu dimenzí vidění na 1D, 2D a 3D. Obyčejná 1D zařízení jsou nejdostupnější variantou, ale poskytují měření pouze v jedné ose. V hobby aplikacích se můžeme často setkat s umístěním těchto senzorů na servomechanismy a tím docílit měření ve více osách. Kategorie 2D zařízení je velkou doménou průmyslových robotů například ke kontrole povrchů. Jsou také nedílnou součástí mnohých robotických vysavačů. Poslední kategorií jsou 3D zařízení, která se uplatňují zejména v mapovacích aplikacích případně v odvětví autonomních automobilů.

Důležitým kritériem pro dělení je také venkovní a vnitřní prostředí. Lidar zařízení určené pro venkovní použití je zpravidla složitější a tím pádem také dražší.
(26) (27)



Obrázek 14 Ukázka mapování pomocí LIDAR (28)

3.4.8 Elektronický kompas

Kompas patří mezi základní navigační prvky, slouží k určení světových stran na základě zemského magnetismu. Pod pojmem kompas si lze představit klasický kompas obsahující magnetickou střelku, pro účely digitálního zpracování je pak vhodný elektronický kompas. Elektronických kompasů je celá řada, od jednoduchých, které obsahují zpravidla pouze 3-

osý magnetometr, po složitější, které obsahují i akcelerometr a gyroskop, ty jsou pak schopny kompenzace náklonu. Elektronické kompasy můžeme rozdělit na indukční a magnetorezistivní. Samotné senzory spolu s další elektronikou jsou v dnešní době zpravidla umístěné v jednom integrovaném obvodu a komunikace s modulem je realizována pomocí I2C sběrnice. Existuje také celá řada modulů, které ulehčují propojení integrovaného obvodu s mikrokontroléry, ty mají vyvedené potřebné piny pro napájení a komunikaci. (29)

3.4.9 Globální družicový polohový systém (GNSS)

GNSS (Global Navigation Satellite System) jsou již nedílnou součástí pro určování polohy, případně času pro mnohá zařízení po celém světě. Ať už jde o mobilní telefony nebo autonomní drony. Jde o dálkoměrný systém zajišťovaný sítí družic kolem Země, které vysílají signál. Na základě zpracování tří a více signálů z různých družic lze vypočítat přibližnou polohu. Zástupci těchto systémů jsou GPS, GLONASS a brzkými Galileo a BeiDou-3. Současné přijímače jsou schopny zpracovávat data z více GNSS systémů současně, což vede k větší přesnosti. Pro zvýšení přesnosti, respektive eliminaci chyb existují diferenční GNSS (DGNSS) systémy. Mezi tyto systémy můžeme zařadit SBAS a GBAS. SBAS (Satellite Based Augmentation Systems) je systém referenčních stanic poskytující korekce pro eliminaci chyb z GNSS družic. Korekce jsou odeslány SBAS satelitům, které je dále distribuují. V Evropě jde o systém EGNOS. Síť GBAS (Ground Based Augmentation Systems), je velmi podobná SBAS, hlavní rozdíl je pak v distribuci korekčních dat. Data jsou distribuována nejčastěji pomocí mobilních sítí nebo radiovým vysíláním. V ČR jde o službu CZEPOS, kterou zajišťuje Zeměměřický úřad. Velmi rozšířenou metodou je také použití RTK GNSS (Real-Time Kinematic GNSS). Jde o určení polohy v reálném čase, kde je zapotřebí dvou GNSS modulů podporující RTK. Jeden z modulů je v roli pevné stanice (base), která zná svou polohu a zasílá korekční zprávy druhému modulu (rover), který je zpravidla v pohybu. Zpracováním těchto korekčních zpráv lze zpřesnit polohu. Hlavním rozdílem oproti DGPS je nutnost použití fázových přijímačů oproti jednodušším kódovým přijímačům, s čímž také souvisí dosažitelná přesnost, která může být u DGPS do 10 cm a v případě RTK jde o centimetrovou přesnost. (30) (31) (32)

3.4.10 Radiofrekvenční systémy

Velké množství lokalizačních systémů je vyvinuto na technologiích Wi-Fi, Bluetooth, UWB. Tyto technologie se nejvíce využívají v budovách, kde není GPS signál. Následující tabulka ukazuje možnosti jednotlivých systémů.

Technology		Accuracy	Range	Suitable for	Tracking	Transmitter power supply	Battery lifetime
Wi-Fi		< 15 m	< 150 m	area detection		or	medium
BLE	4.0	< 8 m	< 75 m	area detection			high
	5.1	< 1 m with line-of-sight					
UWB		< 30 cm	< 150 m	area detection		or	medium
RFID		presence detection only	< 1 m	spot detection		— (passive RFID tag)	— (passive RFID tag)

Obrázek 15 - porovnání pozičních systémů (33)

Jak je vidět podle tabulky, pro lokalizaci robotické platformy se technologie UWB jeví jako vhodné řešení. Běžné technologie jako je Wi-Fi, Bluetooth, LoRa, ZigBee jsou primárně určeny k přenosu dat. Zpravidla jedinou možností, jak tyto technologie využít k lokalizačním účelům je využití RSSI. Naopak technologie UWB nabízejí kromě přenosu dat také lokalizační a navigační použití. Existuje celá řada lokalizačních řešení na technologiích UWB. Velmi známý UWB modul nabízí firma DecaWave DW1000, který dle dokumentace dosahuje přesnosti 10 cm. (33) (34) (35)

3.4.11 Inerciální měřící jednotka (IMU)

IMU (Inertial measurement unit)) je systém pro inerciální měření, který může podpořit přesnost navigace. Jednotka IMU se nejčastěji skládá z akcelerometru, gyroskopu a magnetometru. Jednotlivé senzory jsou zpravidla v tříosých verzích, což jim dovoluje měření v prostoru. Nejčastější použití tohoto systému je ve spojení s GPS, kdy lze zpracováním IMU a GPS dat velmi zpřesnit výslednou polohu. (12)

3.4.12 RFID (Radio Frekquency Identification)

RFID je technologie sloužící k identifikaci objektu na krátkou vzdálenost pomocí radiofrekvenčních vln. Tato technologie se v hojné míře využívá ve skladových a logistických procesech nebo pro sledování zvířat. Technologie je založená na čtečce RFID čipů. Čipy jsou dvojího typu, aktivní a pasivní. Aktivní jsou méně rozšířené, jejich charakteristickou funkcí je aktivní vysílání signálu, a tedy potřeby baterie. Na druhou stranu je možné takový druh čipů číst na velké vzdálenosti. Pasivní čipy lze přečíst jen v bezprostřední blízkosti čtečky, jsou však dostupnější a nepotřebují údržbu v podobě výměny baterie. (36)

3.4.13 Kamera (Strojové vidění)

Strojové vidění je metoda zpracování obrazových informací a extrakci dat na míru požadované úloze. Příkladem je detekce okrajů, měření rozměrů objektu, rozpoznávání vzorů, čtení čárových kódů, rozpoznání textů nebo analýza barev. Strojové vidění lze využít i v navigaci a orientaci v prostředí, zejména pak s využitím stereo kamery, která nám dovolí určování vzdáleností. Důležitým aspektem, je také software, který s obrazem pracuje. Většina řešení v podobě průmyslových linek má pevně stanovené algoritmy pro danou aplikaci. Výsledkem je zpravidla signál ano/ne. V případě orientace robotického zařízení v prostoru spíše využijeme hluboké učení, jehož výsledkem bývá určitá pravděpodobnost. (37)

3.5 Řídící elektronika

Řídící elektronika sehrává důležitou roli, zajišťuje vyhodnocování dat ze senzorů, řídí jednotlivé komponenty jako jsou motory, serva a relé. Pro jednodušší aplikace lze využít jeden, případně více mikrokontrolérů. V případě složitějších úloh můžeme volit mezi jednodeskovým počítačem, který již obsahuje mikrokontrolér, nebo použít jednodeskový počítač a ten propojit s mikrokontrolérem. Můžeme však také využít principu distribuovaného systému a náročnější úlohy vykonávat na jiném zařízení, může jít například o aplikace hlubokého učení.

3.5.1 Jednodeskové počítače

V dnešní době je na trhu k dispozici velké množství jednodeskových počítačů. Jejich popularita neutichá. Roste jejich nasazení v nejrůznějších odvětvích, zejména pak v kontextu aplikací internetu věcí (IoT) nebo Průmyslu 4.0. V rámci robotických aplikací tyto počítače umožňují zpracování většího množství senzorických dat, které by mikrokontroléry nezvládly, může jít o zpracování dat z kamery nebo Lidaru. Množství aplikací také vyžaduje spouštění neuronových sítí, například za účelem rozpoznání obrazu. Mezi nejznámější počítače patří Raspberry Pi, Orange Pi, Banana Pi, Odroid, UDOO, ASUS Tinker Board a Jetson Nano. Existují také hybridní řešení, která na jedné desce poskytují jak výhody počítače, tak mikrokontroléru, například LattePanda či BeagleBone. (38)

3.5.2 Mikrokontroléry

Mikrokontrolér je nezbytnou součástí robotických aplikací. Jsou schopny interagovat s elektronickými moduly jako jsou servomechanismy, h-můstky, relé, akcelerometry, gyroskopy, enkodéry, displeje, RF moduly atd. Mezi typická rozhraní mikrokontrolérů patří I2C, SPI, UART, analogové vstupy/výstupy a vstupně/výstupní porty, které mohou sloužit pro generování PWM (Pulse Width Modulation) pulzů. Mezi známé mikrokontroléry patří 8-bitová rodina ATmega, konkrétně pak ATmega328 a ATmega2560 firmy Atmel. Těchto mikrokontrolérů využívá populární platforma Arduino. Arduino také disponuje moduly s výkonnějšími 32-bitovými ARM mikroprocesory. Za zmínku také stojí připravovaný model Portenta H7 s podporou umělé inteligence. Stále rostoucí popularitě se také těší rodina mikrokontrolérů STM32 díky své ceně a vyššímu výkonu oproti běžným ATmega. Poslední skupinu tvoří ESP32 případně ESP8266, které mají ještě větší výkon oproti STM32, navíc nabízí také integrovaný Wi-Fi a Bluetooth modul. (39) (40) (41)

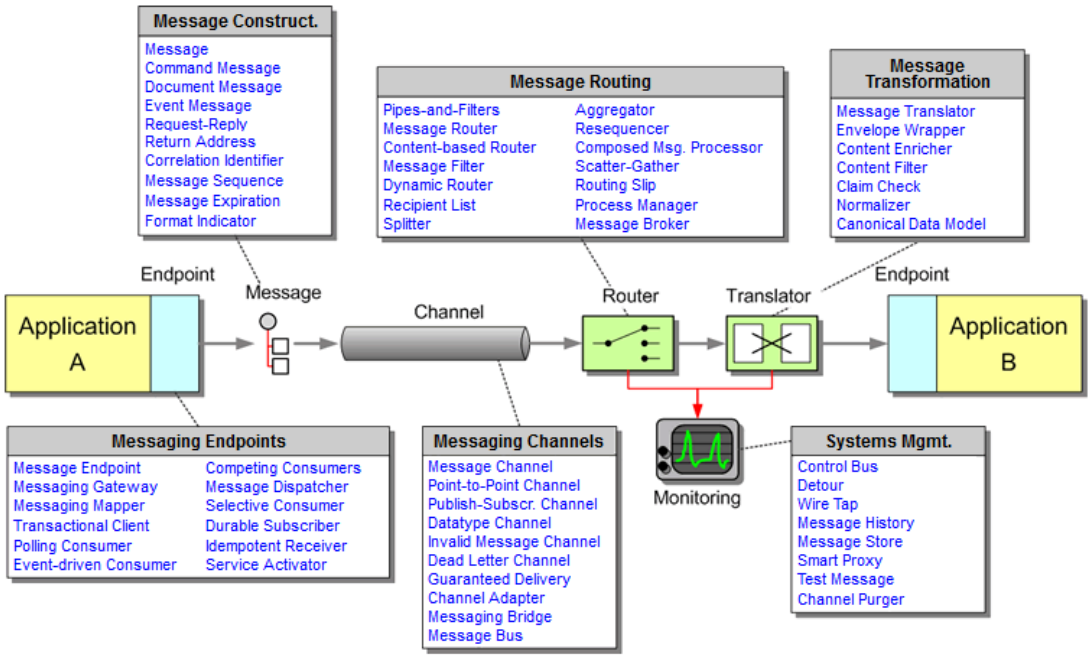
3.6 Komunikační protokoly

Existuje celá řada komunikačních protokolů a paradigmat pro předávání zpráv mezi systémy. Samotných protokolů může být v robotickém projektu celá řada, komunikace senzor – mikrokontrolér, mikrokontrolér – mikrokontrolér, mikrokontrolér – počítač, případně počítač - webový prohlížeč. Co se týče komunikace mikrokontrolérů se senzory,

jsme omezeni jejich rozhraním, což může být I2C, SPI, UART nebo CAN. Protokol pak určuje senzor samotný, například NMEA u GNSS modulu. Pro komunikaci mezi mikrokontroléry nebo počítačem je pak klíčový samotný formát přenášených dat z důvodu limitované paměti a výpočetního výkonu. Zpravidla chceme zachovat co možná nejvíce prostředků pro obsluhu připojených zařízení a případných výpočtů.

V případě počítače máme k dispozici nepřeberné množství možností realizace komunikačního rozhraní s řídicí aplikací. Samotná komunikace musí být zajištěna technologií umožňující přenášení dat z kamery, což značně redukuje použitelné technologie. Mezi vhodné technologie můžeme zařadit Wi-Fi a mobilní datové sítě, které nejenže poskytují dostatečný datový tok, ale také TCP/IP protokol, který umožní komunikaci v rámci počítačové sítě.

Komunikace mezi aplikacemi může být zrealizována přímo na TCP nebo UDP. Je však vhodné využít implementaci z aplikační vrstvy, například WebSocket, Socket.IO, gRPC, ZeroMQ atd. Případně lze využít protokolů IoT, například MQTT, CoAP nebo AMQP. Ať je již použit jakýkoli protokol, klíčové je, zda jeho implementace dokáže přenést požadovanou zprávu v požadovaném čase. Výhodou je také nativní podpora komunikačních paradigmat Publish/Subscribe a Request/Response. Velké množství informací o problematice přenosu zpráv lze získat například zde (42)



Obrázek 16 - Schéma vzory pro přenos zpráv (42)

3.6.1 Formát dat

Formáty pro výměnu dat mezi systémy jsou velmi důležitou volbou, která se neobejde bez kompromisů. Formáty můžeme rozdělit na textové a binární. Textové formáty jsou čitelné člověkem lze je jednoduše modifikovat, na druhou stranu zpravidla představují větší velikost a vyšší HW nároky na jejich zpracování. Příkladem takových formátů je JSON, XML, CSV či YAML. Binární formáty nejsou přímo čitelné člověkem, mívají menší velikost a nižší HW nároky na zpracování. Jako příklad nám mohou posloužit formáty BSON, MessagePack nebo ProtocolBuffers.

3.6.2 Serializace a Deserializace

Pro přenos dat mezi systémy je třeba data na jedné straně určitým způsobem zakódovat (serializovat) tak, aby bylo možné tato data přenést typicky jako množinu bajtů komunikačním kanálem a na druhé straně zase dekódovat (deserializovat). Ve světě počítačů se tak děje zpravidla pomocí JSON, XML či CSV. Na druhé straně ve světě mikrokontrolérů už taková komunikace nemusí být zcela přijatelná z hlediska velikosti přenášených dat nebo rychlosti jejich zpracování. Mikrokontroléry bývají omezeny jak výpočetním výkonem, tak pamětí. Chceme-li tedy přenášet data s minimálním dopadem na rychlost jejich zpracování a paměťovou náročnost, použijeme binární formát.

3.6.2.1 BSON

Jde o binární verzi populárního formátu JSON. Jedná se o binárně kódovanou datovou strukturu, která je oproti formátu JSON rychle zpracovatelná a lze její pomocí přímo přenášet binární data, například obrázky, bez nutnosti konverze, například pomocí base64. Velkou výhodou tohoto formátu je uchování typu ukládaných dat a jejich velikosti v bajtech, což ale na druhou stranu přináší nepatrný nárůst objemu dat. Dále jsou pak data ukládána ve formě množiny páru klíč - hodnota, což také skrývá jak výhodu, že víme co hodnota obsahuje, ale na druhou stranu může být nepotřebným zdrojem velkého objemu dat.

3.6.2.2 MessagePack

Jde také o obdobu JSON, která má stejně jako BSON binární kódování. Tento protokol má oproti BSON menší nároky na HW a výsledná zpráva je zpravidla menší. I tento formát obsahuje množinu klíčů a hodnot.
(43)

3.6.2.3 ProtocolBuffers

Protocol buffers je velmi zajímavý počin od firmy Google. Jde rovněž o binární přenos zpráv, velkou výhodou je možnost vytvořit platformě nezávislou zprávu, která může obsahovat primitivní datové typy nebo další zprávy. Po definování zpráv je lze přeložit do mnoha programovacích jazyků. To umožňuje jednoduchou komunikaci napříč aplikacemi v různých jazycích. Velkou výhodou je částečná absence modelu klíč/hodnota, ve zprávě je obsažena informace o pořadí proměnné, což zásadně redukuje velikost výsledné zprávy. ProtocolBuffers využívají také "variable-length integer encoding". Podstatou je, že serializujeme-li číslo, které je reprezentováno pouze pomocí 8 bitů, pak se odešle právě jako 8 bitů, ačkoli je ve zprávě definováno jako číslo 32-bitové. Tento postup má kladný dopad na velikost zprávy, který je ale vykoupen výpočetním časem jak na straně kódování, tak na straně dekódování. (44)

3.6.2.4 Cap´n Proto

Cap´n Proto vychází z ProtocolBuffers, jeho hlavní odlišností je absence optimalizací, tj. zmenšování počtu bitů u čísel a vynechávání prázdných tagů. Tím naroste výsledná velikost, ale nemusí se vykonávat optimalizační algoritmy. Největší výhoda pak tkví v tom, že známe-li schéma, můžeme přesně v posloupnosti bytů určit, kde se jaký tag (proměnná) nachází. (45)

3.7 Frameworky pro vývoj robotů

S rozvojem robotických aplikací vzniká také potřeba jisté podpory při jejich vývoji. Velké množství firem má své uzavřené frameworky pro vývoj, ale jsou zde také volně dostupné frameworky například MRPT (Mobile Robot Programming Toolkit), MRDS (Microsoft

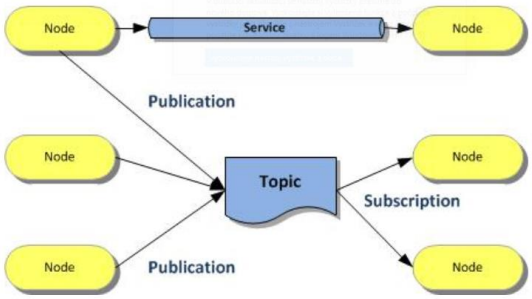
Robotics Developer Studio), CARMEN (Carnegie Mellon Robot Navigation Toolkit). Nejrozšířenějším je ale ROS (Robot Operating System). ROS jako takový je robotický middleware, stará se o vzájemnou komunikaci mezi jednotlivými moduly (nódy). Moduly mohou být primárně psané v jazycích C++, Python a Lisp. Částečně je také podporována Java a Lua. Hlavní komponenty frameworku jsou:

- Nódy (Nodes) reprezentují jednotlivé procesy, které zajišťují nějakou funkci nebo funkce robota a vzájemně spolu interagují pomocí zpráv nebo služeb. Každý nód se může přihlásit k odběru zprávy pod určitým tématem, nebo naopak zprávy publikovat.

- Zprávy (Messages) umožnují vzájemnou komunikaci mezi nódy. Každá zpráva obsahuje téma (Topic), podle kterého je zaslána příslušné množině nódů přihlášených k jejich odběru. Zprávy jsou implementací paradigmatu Publish/Subscribe

- Služby (Services) umožňují, stejně jako zprávy, vzájemnou komunikaci, ale v tomto případě je užité jiné komunikační paradigma a to Request/Response.

- Parametrický server (Parameter server) umožňuje číst a zapisovat zejména konfigurační parametry, může jít o výchozí hodnoty potřebné k inicializaci, například rozlišení kamery nebo koeficienty PID regulátoru.

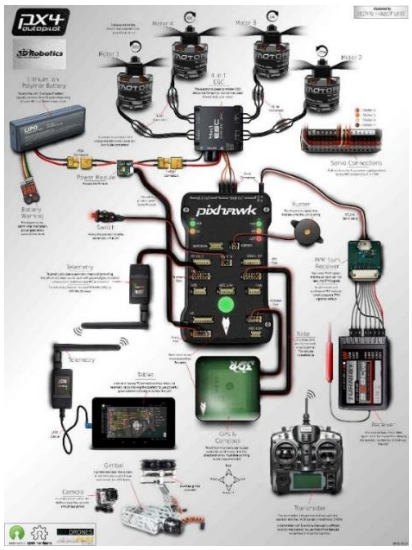


Obrázek 17 - ROS schéma (46)

Framework také obsahuje mnoho užitečných diagnostických, vizualizačních a simulačních nástrojů. Má velkou podporu a širokou komunitu, existuje celá řada balíčků, které již řeší pokročilé aplikace, jako je lokalizace na základě fúze dat různých senzorů, mapování a mnoho dalších. (47) (46)

3.8 Autopiloty

S rozvojem dostupných dronů rostou i jejich funkcionality, jako třeba autonomní let. Mnoho takových řešení je přímo integrované v zařízení, ale existuje i řada samostatných autopilotů, které lze integrovat do vlastního dronu. Jsou autopiloty, respektive softwarová řešení, která jsou vhodná jak pro drony, tak pro pozemní vozidla, lodě atd. Jednou z nejpopulárnějších platforem je Ardupilot. Ardupilot je open-source projekt pro řízení kopterů, letadel, vrtulníků, pozemních vozidel, lodí, ponorek. Jeho použití je možné na široké škále HW, například APM či Pixhawk. Ardupilot poskytuje možnost různých módů od přímého řízení přes poloautomatické, až po možnosti plně autonomního letu. Platforma je schopná v omezené míře používat některé externí senzory jako Lidar, ultrazvukový dálkoměr nebo optický senzor. Rovněž existují knihovny schopné platformu ovládat. Za zmínku také stojí Navio2. Jde o rozšiřující desku senzorů k jednodeskovému počítači Raspberry Pi. Dodávaný operační systém pro počítač pak obsahuje předinstalovaný ROS spolu s Ardupilotem. Získáme prostředí obsahující spolehlivého autopilota, na jehož základě můžeme pomocí ROSu vybudovat požadovanou aplikaci. (48)



Obrázek 18 - Pixhawk schema (48)

4 Vlastní práce

4.1 Výběr technologií

4.1.1 Volba platformy

Zásadní otázkou pro výběr komponent a technologií, které budou použity k vlastnímu sestavení robota je, zda sestavit pozemní stroj nebo létající dron. Obě varianty mají svá pro a proti.

Klady pro volbu dronu:
- vyšší mobilita, možnost pohybu ve třech osách.

Zápory pro volbu dronu:
- selhání řídícího systému nebo některé z komponent může vést v krajním případě k újmě na zdraví či majetku.
- v rámci snížení pravděpodobnosti požáru od Lithium-polymerových baterií použít Lithium-iontové, což zvýší velikost dronu.
- špatné vyhodnocení senzorických dat může vést k poškození dronu.
- počasí

Klady pro volbu pozemního robota:
- porucha zpravidla neznamená jakékoli ohrožení okolí nebo stroje samotného.
- špatné vyhodnocení senzorických dat většinou nezpůsobí poškození stroje.
- možnost zastavení a získání času na případné lepší vyhodnocení.

Zápory:
- omezený pohyb stroje v prostředí.
- omezený rozhled.

K realizaci byl zvolen pozemní stroj, zejména kvůli jednodušší implementaci a menším provozním rizikům.

4.1.2 Výběr navigačního systému

Základní ideou při realizaci byla cena v poměru k použitelnosti zařízení. Ideální provedení by mělo být dostupné, jednoduché na údržbu a provoz. Rovněž by mělo být snadno aplikovatelné do různorodého operačního prostředí. V realitě ale musíme přikročit k mnoha kompromisům.

Jedním z prvních návrhů bylo použití systému relativní lokalizace pomocí RFID, RSSI nebo ToF na běžně dostupných modulech LoRa, Bluetooth, případně Wi-Fi. Primární funkcí všech těchto zařízení je však přenos dat, jejich použití pro získání hodnot doby letu signálu je zpravidla nemožné. V případě použití RSSI přesnost neodpovídá potřebám lokalizace. Jejich použitelnost, s přihlédnutím ke stabilitě a celkové možnosti navigace pomocí RSSI a ToF na běžných modulech, byla tedy vyloučena. Jedinou možností je použití modulů, případně hotového řešení, které jsou pro tuto aplikaci vyrobené, ty jsou však velmi nákladné. I v takovém případě by pak k realizaci tohoto řešení byla potřeba celá sada těchto vysílačů po celé oblasti, kde by robot operoval a každé zařízení by potřebovalo vlastní napájení. Otázka napájení však nemusí být tak jednoduchou záležitostí, existuje mnoho různých variant. Nejjednodušší řešení je napájení z elektrického zdroje, který by byl připojen ke každému zařízení, což je ale přijatelné pouze v případě, že si můžeme takový rozvod energie dovolit. Například při nasazení systému na zahradě, by bylo třeba rozvést energii ke každému zařízení, což není zrovna ideální. Další možností je použití baterie. Běžná zařízení ale nejsou schopná fungovat na baterii déle než rok. Bylo by tedy nutné přidat způsob jejich dobíjení. Jednou z možností je použití solárního panelu, to však v mnohých případech není aplikovatelné řešení. Další alternativou by mohlo být bezdrátové nabíjení. To by spočívalo v nabíjení baterie pomocí bezdrátové nabíječky, robot by tedy musel jednou za určitý čas projet všechna vysílací stanoviště a dobít je. Tedy je zde předpoklad, že všechny vysílače musí být robotem dosažitelné. Velkou nevýhodou je čas, který by byl potřebný k nabití takové stanice.

Další volbou je použití pasivních prvků, které nepotřebují napájení, což může být právě technologie RFID. Ovšem nasazení takového řešení nemusí být vždy možné, na druhou stranu je to velmi jednoduché a levné provedení, které se při jeho testování jevilo zcela použitelné. Hlavním problémem je, že robot by měl být schopný dostat se z jednoho čipu na druhý co nejpřesněji, ale zároveň, aby vzdálenost mezi čipy nebyla příliš malá. Další

překážkou je, že vzdálenost čtení pasivního čipu je velmi limitována. Běžná zařízení pracují maximálně do 5-ti centimetrů, na trhu jsou ale i relativně levná zařízení schopná číst až do teoretických 40 cm. Testování potvrdilo možnost čtení čipu, který byl umístěn 5 cm pod zemí, přičemž čtecí anténa byla přibližně 5 cm nad povrchem. Ovšem z testování vyplynul fakt, že detekce čipu není vždy zaručena, a to ani v případě použití více RFID čteček, u kterých naopak nastává problém se vzájemným rušením.

Finální navigační technologií, která je velmi snadno použitelná, ovšem za vyšší cenu, je kit RTK GNSS modulů. Moduly jsou schopné určit přesnou polohu v řádech centimetrů, jde o systém absolutní lokalizace. Lze tedy snadno využít mapových podkladů bez nutnosti transformací souřadnic.

4.1.3 Typ podvozku

Při výběru typu podvozku byl brán zřetel na jednoduchost realizace, manévrovatelnost, prostupnost terénem a nosnost. Nejjednodušším řešením by byl výběr stávajícího podvozku použitého při Bakalářské práci. Jde o robustní kovovou konstrukci vybavenou čtveřicí motorů a velkými koly, ovšem tento model má omezení týkající se rychlosti pohybu, ale hlavně pak absenci enkodérů pro aplikaci odometrie. V ostatních ohledech je však toto uspořádání vyhovující. Tento typ podvozku zvládá prostup obtížnějším terénem, je schopný otočení na místě a disponuje relativně velkou nosností, realizace je například oproti Ackermanově a všesměrovému typu podvozku jednoduchá. Uvažujeme-li diferenciální podvozek, máme volbu kolového nebo pásového provedení. V rámci jednoduchosti vyhrává kolový podvozek, který je snazší na realizaci i pozdější údržbu. Dalším krokem je pak výběr krokových motorů nebo převodovaných motorů s enkodéry, aby byla aplikovatelná odometrie. V rámci zachování jednoduchosti realizace bylo zvoleno hotové řešení převodovaného motoru s enkodéry a kolem.

4.2 Použitý HW

4.2.1 Podvozek

Pro prototyp byl vybrán diferenciální čtyřkolový podvozek, zejména pro jeho lepší manévrovatelnost oproti Ackermanově uspořádání. Velkou výhodou tohoto typu podvozku je jednoduchost, stabilita a prostup terénem. Lepší volbou s ohledem na terén je pásový podvozek, který je ale konstrukčně složitější. Velmi dobrou alternativou je také diferenciální dvoukolový podvozek, kde lze snadněji aplikovat odometrii, ale na druhou stranu je nevhodný v terénu.

Obrázek 19 - Podvozek (49)

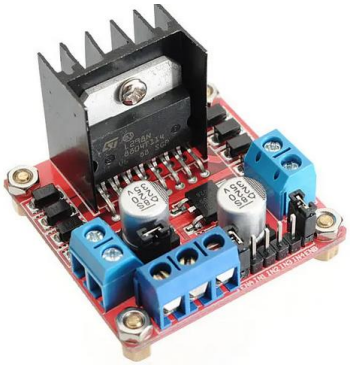
4.2.2 Motory

Pohyb robota zajišťují čtyři motory 107 ot/min s převodovkou 1:90 a dvoukanálovým enkodérem. Přítomnost převodovky zajišťuje optimální otáčky a kroutící moment. Robot není schopen vyvinout velkou rychlost, což v tomto případě není ani žádoucí. Na druhou stranu jsou motory schopny poskytnout sílu k otočení robotu kolem své osy na místě, což je obrovská výhoda pro možnosti manévrování. Integrovány jsou rovněž dvoukanálové inkrementální rotační enkodéry na magnetickém principu. (49)

Obrázek 20 - Motor (49)

4.2.3 Řízení motorů

K řízení motorů je použit jednoduchý modul založený na obvodu L293. Tento obvod dokáže řídit dva DC motory nebo jeden krokový motor. V našem případě je tedy schopen regulovat rychlost a směr otáčení motorů pomocí PWM. Modul je napájen přímo z baterie a řídící příkazy jsou generovány mikrokontrolérem. Primárně je modul určen k řízení dvou DC motorů, ale kvůli jednoduchosti jsou zapojeny všechny čtyři motory. Tedy levý přední a zadní motor jsou na prvním výstupu a pravé na druhém výstupu. Tento modul dále obsahuje chladič, svorky pro napájení, motory i piny na řídící signály. (49)



Obrázek 21 - h-můstek (49)

4.2.4 Kola

Byla použita terénní kola o průměru 13 cm a šířce 6 cm, která umožňují pohyb ve vyšší trávě nebo v menším terénu. (49)

Obrázek 22 - Kolo (49)

4.2.5 Napájení

Napájení celého systému zajišťuje olověný akumulátor 12V 5Ah. Olověný akumulátor byl zvolen zejména pro svou jednoduchost nabíjení a cenu, dalším kladem je větší odolnost vůči nižším teplotám. Velkou nevýhodou olověného akumulátoru je však nízká energetická hustota, a tedy větší rozměry i váha. Napětí dovoluje přímé napájení motorů. Pro napájení elektroniky byly použity napěťové měniče.

Obrázek 23 - Baterie (50)

4.2.6 Počítač

V roli hlavního počítače je použit Raspberry Pi 2B. Jde o jednodeskový počítač s dobrým poměrem výkonu vzhledem ke spotřebě pro tuto aplikaci. Na počítači je nainstalován operační systém Raspbian 9 stretch, jde o modifikovaný Debian pro rodinu počítačů Raspberry Pi. Nejedná se o jedinou možnost, existuje velké množství Ubuntu distribucí, Windows 10 IoT Core a mnoho dalších. Na počítači je dále nainstalován Python interpret ve verzi 3.7.0, který je nutný pro běh programů. Programy, tedy webová aplikace a řídicí program, mají každý oddělené virtuální prostředí. Virtuální prostředí slouží k instalování knihoven závislých pro běh programu, aniž by se ovlivnila systémová nastavení nebo jiná virtuální prostředí. Každé jednotlivé virtuální prostředí obsahuje potřebné knihovny a jejich verze a také cestu k Python interpretu, který se má pro běh kódu použít. (51)

Obrázek 24 - RPi 2B (52)

4.2.7 Mikrokontroléry – Arduino

Pro řízení jednotlivých komponent robotu je použita sada Arduino desek založených na mikrokontroléru ATmega2560. Celý robot je vybaven třemi deskami. Jsou použity hned dva typy těchto desek. Prvním typem je standardní klon desky Arduino Mega 2560. Druhým typem je zmenšená verze stejné desky, která přišla o některé I/O piny a napájecí konektor. Na druhou stranu je deska o poznání menší. Mikrokontroléry jsou převážně použity jako komunikační most mezi komponentami a hlavním počítačem. (39)

Obrázek 25 - Arduino mikrokontroléry (53)

4.2.8 Kamera

Pro pořizování vizuálního záznamu byl použit oficiální kamerový modul k Raspberry Pi ve verzi 1. Tento modul dokáže pořizovat video záznam v rozlišení 1080p při 30 snímcích za vteřinu. Fotky lze pořizovat v maximálním rozlišení 2592 x 1944 px. Komunikace probíhá pomocí CSI. (52)

Obrázek 26 - RPi kamera (52)

4.2.9 Kamerový držák

Kamera je umístěna na 2DOF plošině, která umožňuje její horizontální pohyb v úhlu ~ 180° a vertikální pohyb v rozsahu 100°. Pohyb zajišťují dvě digitální 9g serva SG90, jde o klasická modelářská serva umožňující pohyb v rozsahu přibližně 180° se silou 1,8 kg/cm. Pro jeho řízení je za potřebí PWM signál s pulzy od 1 do 2 ms. (53)

Obrázek 27 - Kamerový držák (53)

4.2.10 Kompas

Pro určení natočení robotu vůči světovým stranám je prototyp vybaven elektronickým kompasem HMC5883L. Jde o 3-osý magnetometr, který komunikuje přes rozhraní I2C. Jedná se o jednoduchý modul, který neobsahuje žádnou kompenzaci náklonu. Modul dokáže měřit maximální rychlostí 75 Hz, při speciálním zapojení pak až 160 Hz. Podporuje měření ve více módech, například měření na dotaz nebo kontinuální měření. Zajímavou funkcí je také průměrování až osmi naměřených hodnot. (54)

Obrázek 28 - Kompas hmc588l (53)

4.2.11 GNSS

Pro určení polohy je použit kit GNSS C94-M8P, který je vhodný pro RTK aplikace. Tato sada obsahuje dvě desky, každá z desek obsahuje GNSS a RF modul. GNSS modul je založen na ublox NEO-M8P-2, které jsou vhodné pro režimy "base station" a "rover". RF modul slouží pro komunikaci s ostatními moduly. Tato komunikace slouží typicky k distribuci korekčních dat z "Base station" do "rover" modulu, jejichž zpracováním je schopen zpřesnit svou polohu. (55)

Obrázek 29 - GNSS kit (56)

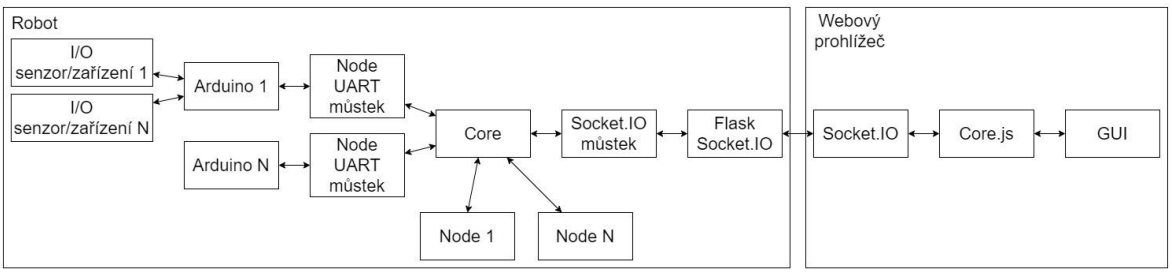
4.2.12 Ultrazvukové dálkoměry

Pro detekci překážek je robot vybaven pěti ultrazvukovými dálkoměry HC-SR04. Tyto dálkoměry měří v rozsahu od 2 do 450 cm. Komunikace s modulem probíhá pomocí vstupně/výstupních pinů. Modul podporuje pouze jednoduchou komunikaci, kdy je mu na pin „trigger“ zaslán pulz. Senzor provede měření a výsledek odešle jako pulz na výstupním portu, jehož doba odpovídá době letu zvukové vlny. (49)

Obrázek 30 - Ultrazvukový senzor HC-SR04 (49)

4.3 Použitý SW

Robot obsahuje množství programů, hlavní program v počítači je napsaný v jazyce Python. Programy v mikrokontrolérech jsou napsány v jazyce Wiring, případně C/C++. Ovládací rozhraní je pak kombinací Python, HTML, CSS a JavaScript.



Obrázek 31 - Diagram programových komponent Zdroj: vlastní tvorba

4.3.1 Komunikace

Formát pro komunikaci mezi hlavním programem a webovou aplikací je JSON, který je jednoduše aplikovatelný, snadno se s tímto formátem pracuje jak na straně hlavního programu v Pythonu, tak na straně webové aplikace v JavaScriptu. Náročnější volbou pak bylo zvolit formát pro komunikaci s mikrokontroléry. Jasnou volbou byla binární forma komunikace, protože dekódování textu i v takto malém objemu, jako jsou předávané zprávy,

je při požadované rychlosti aktualizace dat neproveditelné. Velmi lákavou možností je použití ProtocolBuffers, která má i svoji implementaci pro rodinu produktů Arduino. Vzhledem k velikosti předávaných zpráv je však optimalizace velikosti v protokolu spíše na škodu, protože za cenu zmenšení zprávy o pár bajtů, přijdeme o možnost pevných velikostí datových struktur a toto zpracování stojí procesorový čas. Ideální řešení by tedy bylo použití Cap'n Proto, který bohužel nemá pro Arduino podporu. Konečná volba tedy padla na malou knihovnu SimpleSerialization, která nese aspekty Cap'n Proto, a navíc obsahuje implementaci zpětného volání (callback) v případě příchozí zprávy.

Co se týče komunikačního paradigmatu, pro primární komunikaci je použit model Publish/Subscribe. Pro podpůrné funkce pak Request/Response.

4.3.2 Komunikační knihovna pro mikrokontroléry

Knihovna obstarávající serializaci a deserializaci dat se jmenuje SimpleSerialization. Tato knihovna je napsaná přímo pro komunikaci mezi Arduino mikrokontrolérem a Java aplikací. Knihovna je schopná převádět základní datové typy do pole bajtů, navíc lze takto převádět i celé struktury, které dědí ze základní zprávy této knihovny. Obrovskou výhodou knihovny je malá náročnost na výpočetní výkon i paměť oproti jakékoli práci s řetězci. Nevýhodou byla původní implementace v Javě pro komunikaci s počítačem. Nebyl však takový problém tento kód přepsat do jazyka Python. (57)

4.3.3 Hlavní program

Jazyk pro implementaci hlavního programu byl ovlivněn webovým frameworkem a teoretickou možností použití ROS middlewaru pro komunikaci komponent. Během vývoje bylo však od použití ROSu upuštěno. Ačkoli jde o velmi povedený framework, není úplně jednoduché se v něm zorientovat, zejména pak v pokročilých navigačních balíčcích. Velkou nevýhodou tohoto frameworku je komunikace mezi jednotlivými nódy, která probíhá pomocí TCP/IP a UDP což může do jisté míry zpožďovat přenos zpráv. Co se integrace s Arduino systémy týče, existuje balíček rosserial, který dovoluje komunikaci mikrokontroléru pomocí sériové linky. Má však omezené funkce, zejména pak při práci s předáváním parametrů. Další nepříjemnou vlastností je paměťová náročnost v případě, že

chceme použít více "publish/subscribe" zpráv. Framework ROS nebyl použit, byly však použity některé jeho principy.

Hlavní program je napsaný v jazyce Python, otestován na Pythonu ve verzi 3.7.4. Jsou použity zejména knihovny pro práci s JSON formátem, knihovna pro soketovou komunikaci Socket.IO a knihovna implementující PID regulátor. Kód je inspirován architekturou ROS. Z této architektury přejímá princip Publish/Subscribe a Request/Response. Dále přejímá získávání a nastavení parametrů podle tématu a samotné uzly (nodes), které si vzájemně zprávy vyměňují. Samotné jádro programu, které se stará o předávání zpráv mezi nódy se jmenuje Core. Core umožňuje jednotlivým nódům registraci k tématu nebo posílání zpráv. Dále je možné získávání a nastavování parametrů. Parametry jsou v tomto případě primitivní typy, které je možné pomocí jejich cesty získat nebo uložit, jde například o komunikační porty nebo různé koeficienty. Dále je zde mnoho nadefinovaných zpráv pro přenos strukturovaných dat a konečně samotné nódy.

Příklady nódů:

- Nód pro přenos zpráv ze soketového připojení zajišťuje výměnu zpráv a parametrů mezi hlavním programem a Socket.IO serverem, který dále komunikuje s webovou aplikací

- Nód pro přenos zpráv z a do mikrokontrolérů zajištuje komunikaci po sériové lince s mikrokontrolérem a stará se o překlad zpráv.

- Nód pro synchronizaci parametrů mezi mikrokontrolérem umožňuje nastavit a získat parametr v mikrokontroléru, navíc lze vytvořit posloupnost úloh, které jsou volány nadefinovaným tématem. Tato posloupnost postupně může nastavovat parametry v mikrokontroléru a zároveň je kontroluje. V případě, že mikrokontrolér nedostal požadavek na změnu parametru nebo nebyla obdržena odpověď, odešle se opakovaně do nastaveného limitu.

- Nód obsluhy kamery zodpovídá za pořizování snímků z webové kamery a jejich ukládání, rovněž obsluhuje kamerovou platformu. Nód generuje malý JSON dokument, který obsahuje informace o pořízených fotkách. Dokument pak slouží pro zobrazení informací o cestě ve webovém rozhraní.

- Nód pro spouštění úkolů funguje v případě, že chceme spouštět procesy, které potřebují k dokončení nějaký čas a nechceme brzdit vykonávání ostatního kódu. Nód je spuštěn definovanou zprávou a obsahuje seznam úkolů. Každý úkol obsahuje zprávu, která se odešle a očekávanou zprávu, která dá najevo, že je operace dokončena a lze tak přejít ke zpracování další. Po dokončení všech úkolů se odešle patřičná zpráva. Tento nód je využit například pro pořízení snímků v otočném bodě. Jakmile je dosaženo bodu, robot je zastaven a spustí se sekvence úkolů. Nastaví se poloha kamery, pořídí se snímek, a to i opakovaně. V momentě, kdy jsou úkoly hotovy, se robot přesouvá na další bod.

- Nód řízení motorů pomocí gamepadu nebo autopilota se jmenuje SimpleEnginesController a odebírá zprávy jak z gamepadu tak z autopilota. V případě autonomního režimu je směr robotu řízen PID regulátorem z poskytovaného směru pohybu a azimutu, rychlost může být jak fixní z konfigurace, tak řízená gamepadem. Ruční řízení je pouze přímé zasílání požadovaného směru a rychlosti.

- Nód „Navigate“ slouží k distribuci otočného bodu. Čeká na příchod zprávy obsahující cestu, po jejímž obdržení začne distribuovat traťové body. Každý další distribuovaný bod je podmíněn dosažením bodu předešlého dle nastavené tolerance.

- Dalším z nódů je „ToPointNavigationNode“, který zajišťuje publikování azimutu a vzdálenosti k zadanému otočnému bodu dle aktuální pozice.

- Nód objíždění překážek přijímá zprávu o azimutu, ta projde systémem detekce překážek a je publikována nová zpráva, která obsahuje modifikovaný azimut

v případě detekce překážky. Samotný princip objíždění překážek je založený na „bubble rebound algorithm“ (58). Algoritmus se v případě detekce překážky, jejíž vzdálenost překročí nastavenou mez, snaží s ohledem na vzdálenost a k nastavenému koeficientu kompenzovat azimut tak, aby se robot překážce vyhnul. Kompenzace azimutu se ukládá, při novém výpočtu se přičítá k případné nové kompenzaci, nebo se zmenšuje v případě, kdy senzory žádnou překážku nedetekují, až do doby jejího vynulování.

4.3.4 Webová aplikace

Pro vývoj webové aplikace byl zvolen webový framework Flask. Byl vybrán z důvodu jednoduchosti, relativně malému nároku na HW a dobré dokumentaci. Mezi další populární frameworky můžeme zařadit Django a Pyramid.

Webová aplikace slouží k ovládání robota pomocí webových stránek. Aplikace jako taková nemá mnoho funkcí, pouze generuje webový obsah. Dále slouží, pomocí knihovny Socket.IO, jako soketový server pro komunikaci mezi JavaScriptem ve webovém prohlížeči a hlavním programem, respektive nódem pro přenos zpráv soketového připojení. Příkazy pro samotné ovládání robota jsou pak výsledkem JavaScriptu v prohlížeči. (59) (60)

4.3.5 JavaScript

JavaScript slouží pro předávání dat mezi HTML strukturou a soketovým klientem v podobě knihovny Socket.IO. Tato knihovna zajišťuje obousměrnou komunikaci mezi webovým klientem a serverem v reálném čase, což poskytuje ideální řešení pro výměnu dat bez nutnosti přenačítání stránky, nebo neustálého AJAXového volání. Pro zajištění komunikace a výměnu zpráv slouží třída Core. Core je třída zajišťující komunikaci se serverem a překlad JSON zprávy na objekt. Tento objekt pak má všechny náležitosti jako zpráva v hlavním programu. Je zde tedy téma (topic) zprávy a data pod stejnými názvy. Třída umožňuje registraci tématu a zpětné volání (callback), ve kterém se má příchozí zpráva zpracovat. Samozřejmostí je také možnost odeslání zprávy.

Pro snadnější integraci přenosu zpráv v HTML struktuře je zde množina funkcí ze skriptu coreUtils. Zde jsou funkce pro aktualizace dat, nebo generování zprávy. Využívá se zde hierarchie HTML elementů a data atributů.

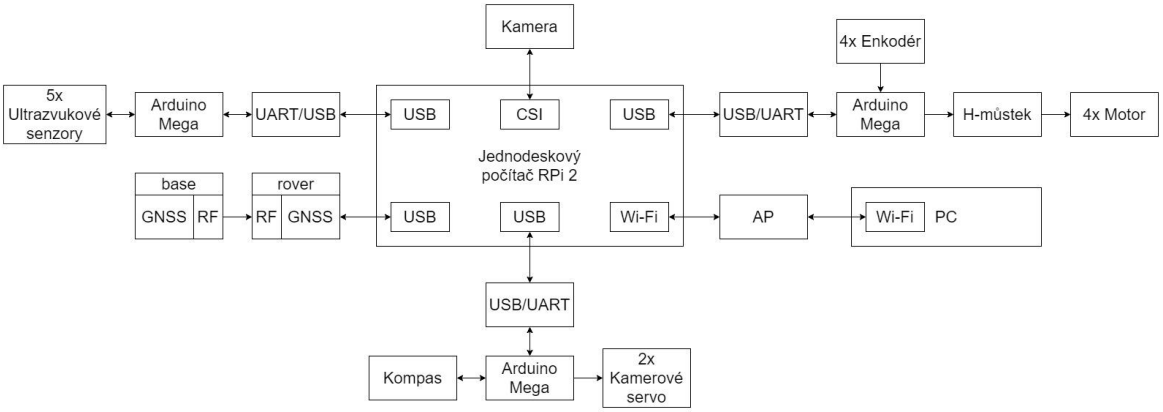
Dalším významným skriptem je knihovna pro zjednodušení obsluhy gamepadu a obslužný skript. Ten reaguje na události, které knihovna generuje a překládá je do systému, kde jsou pak v nastaveném intervalu odeslány.

Poslední kategorií skriptů jsou utility pro převod dat a konečně skripty ovládající vizuální komponenty jako je menu.

4.3.6 Mikrokontroléry

Každý mikrokontrolér obsahuje knihovnu SimpleSerialization pro komunikaci přes sériovou linku. Knihovna disponuje nástroji pro kódování a dekódování zpráv, z nichž každá je definována svojí preambulí, kontrolním součtem, daty a postambulí. Data, respektive jejich kódování a dekódování, se řídí podle příslušné obsluhy dané třídy, která dědí ze třídy Message. V případě, že chceme přijímat zprávu, je třeba ji zaregistrovat a zadat příslušnou obslužnou metodu. Odchozí zpráva se jednoduše odešle. Jednotlivé mikrokontroléry také obsahují knihovny zajišťující komunikaci s připojenými vstupně/výstupními zařízeními.

4.4 Popis jednotlivých subsystémů



Obrázek 32 - Diagram systémů Zdroj: vlastní tvorba

4.4.1 Systém detekce překážek

Systém pro měření vzdáleností před robotem se sestává z pěti ultrazvukových senzorů, mikrokontroléru a vlastního měniče napětí, který tento subsystém napájí. Senzory jsou umístěny v přední části robota, jeden míří přímo ve směru pohybu další jsou umístěné v úhlu 45° a 90° vzhledem k prvnímu senzoru. Toto uspořádání umožňuje získat základní představu o překážkách před robotem i po jeho bocích. Na druhou stranu je zde nezanedbatelná šance, že senzory překážku nezachytí. Velmi záleží na konkrétní překážce a její poloze vůči senzorickému uspořádání. Komunikace s nadřazeným systémem (hlavním počítačem) je realizována pomocí sériové linky. Systém po zapnutí začne měřit vzdálenosti, které s příslušným úhlem odesílá ke zpracování do hlavního počítače.

4.4.2 Systém magnetometru a řízení serv

Tento systém obsahuje magnetometr, dvě serva a dvojí napájení, jedno pro elektroniku a druhé pro serva. I tento systém je propojen sériovou linkou s hlavním počítačem. Komunikace tohoto systému je již sofistikovanější, typické použití je odesílání kalibrovaného úhlu natočení robotu a příjem dat pro natočení kamerových serv. Další komunikace je příjem nebo odesílání hodnot pro kompenzaci magnetometru a možnost spuštění samotné kalibrace.

4.4.3 Systém řízení motorů

Systém pro řízení motorů obsahuje motory s enkodéry, h-můstek, mikrokontrolér a vlastní měnič napětí. Komunikace opět probíhá po sériové lince s hlavním počítačem. Tento systém je nejkomplexnější z hlediska výpočtů na mikrokontroléru, který provádí řízení motorů na základě zpětné vazby z enkodérů pomocí PID regulátoru. Řídící zprávy jsou tedy pokyny, kolik otáček za časový úsek by měli motory na levé a pravé straně vykonat. Je zde také možnost nastavit, po jaké vzdálenosti se má motor zastavit. Výstupem tohoto systému jsou tiky enkodéru za určitý čas, jakož i absolutní tiky za celou dobu jízdy. Co se enkodérů týče, bylo třeba udělat kompromis v jejich připojení. Každý enkodér obsahuje dvě fáze, tedy dva výstupy x 4 enkodéry, které je nutné sledovat. Rychlost změny na výstupu z enkodéru převyšuje rychlost mikrokontroléru softwarově zachycovat jeho změny. Je zde třeba využít

možnosti procesorového přerušení a přepočtu tiků enkodéru. Problém nastává v tom, že použitý mikrokontrolér poskytuje pouze 6 pinů pro přerušení. Použitá knihovna dokáže pracovat i s piny, které přerušení nepodporují, avšak za cenu nepřesných výsledů. Nutno dodat, že při testování na rovné podlaze v malých rychlostech není chyba nijak významná. S ohledem na konstrukci, tedy že jde o čtyřkolové uspořádání, jde o chybu zanedbatelnou. Naopak při testování ve venkovních podmínkách, kde se robot pohyboval po zmrzlém travnatém povrchu, chyba prokluzu kol narostla tak, že hodnoty byly zcela nesmyslné. I tento systém komunikuje s hlavním počítačem po seriové lince.

4.4.4 Systém GNSS

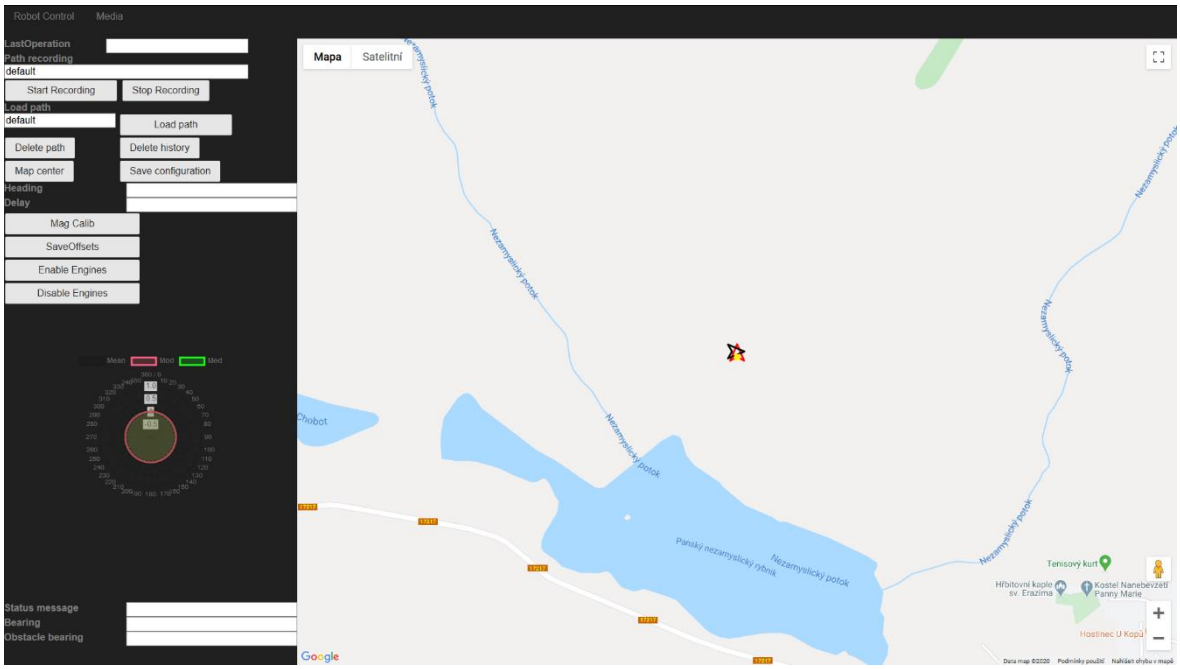
Poslední systém připojený k hlavnímu počítači je GNSS modul. Moduly jsou použity v režimu "base" a "rover" to je základní uspořádání, kdy je jeden z modulů umístěn typicky na robotu, případně libovolné pohyblivé platformě, a druhý modul je stacionární. Stacionárnímu modulu pak lze nastavit jeho pozici pomocí zeměpisné šířky, délky a výšky nebo nastavit požadovanou přesnost a dobu po kterou modul sám počítá svou polohu. S každým novým měřením se pak typicky snižuje chyba polohy, když se modul dostane na požadovanou přesnost zahájí vysílání korekčních dat. Druhý z modulů začne tato korekční data přesněji RTCM v3 v reálném čase zpracovávat čímž dosáhne větší přesnosti, která může být menší než 2,5cm. Jedná se zde však o optimální podmínky, které nejsou vždy snadno dosažitelné. Modul v režimu "rover" se v momentu příjmu korekční zprávy RTCM přepne do módu RTK FLOAT a pak do módu RTK FIXED při vyřešení ambiguit (určení počtu nosných vln mezi družicí a přijímačem). Nezanedbatelnou měrou se na výsledcích podílejí v kitu dodávané antény, které rovněž hrají klíčovou roli ve výsledné přesnosti. (61) Kit je také dodáván s plechy pod antény, které eliminují odražené vlny.

4.4.5 Hlavní počítač

S hlavním počítačem jsou přímo propojeny všechny výše zmíněné systémy. Program obsažený v počítači pak komunikuje s těmito systémy, které dále komunikují s koncovými komponentami robotu. K počítači jsou všechny tyto systémy připojeny pomocí převodníků USB/Serial, výjimkou je kamera, která je na rozhraní CSI a webové rozhraní, které komunikuje přes Wi-Fi rozhraní. I tento systém má vlastní měnič napětí.

4.4.6 Webové rozhraní

Webové rozhraní je velmi jednoduché, obsahuje čtyři hlavní komponenty a to menu, ovládací prvky, zobrazení informací o detekci překážek a mapu.



Obrázek 33 - řídící GUI Zdroj: vlastní tvorba

Komponenta s ovládacími prvky dovoluje nahrávání trasy během manuálního řízení prototypu pro případný pozdější autonomní průjezd. Dále je zde možnost zadat trasu, kterou má robot autonomně projet a další funkce jako kalibrace magnetometru, uložení konfigurace atd. Mnoho funkcí, jako je manuální ovládání robotu, lze však provádět pouze pomocí připojeného gamepadu. Levá spodní část obsahuje graf a status případných překážek. Pravá část je věnována mapě, která obsahuje navigační ukazatele. Prvním ukazatelem je šipka představující polohu robotu a směr natočení, dále je zobrazena šipka představující azimut k

nejbližšímu nebo cílovému bodu. Velký čtvereček představuje nejbližší otočný bod. Při načtení trasy jsou jednotlivé traťové body zaneseny na mapu v podobě malých křížků. Samotný pohyb stroje, respektive každá příchozí zpráva o zeměpisné poloze, je také zanesena malým čtverečkem do mapy.

Obrázek 34 - záznam z projeté trasy GUI Zdroj: vlastní tvorba

Další významnou stránkou je zobrazení projetých tras. Každá projetá trasa v autonomním režimu má svůj záznam v podobě JSON souboru s informacemi o trase a pořízených snímcích z kamery. Záznamy jsou přístupné podle roku měsíce a dne pořízení a konečně časem výjezdu robota na trasu. V levé části stránky se nachází menu, ve kterém lze najít podle výše uvedeného konkrétní záznam a ten pak zobrazit. Jde o základní informace jako je čas, název projeté trasy a snímky z traťových bodů.

4.5 Testování

Testování probíhalo na zahradě, která je pro lepší představu na obrázku níže. Vyznačené úseky jsou 16 x 16 metrů. Hlavním cílem je poukázat na ne zcela ideální podmínky pro využití GNSS zařízení v této lokalitě. Západní strana je zastíněna domem, jižní a východní pak obklopují stromy. Jediný volný výhled poskytuje pouze severní strana. Testovací podmínky tedy nejsou pro moduly ideální a prověří se tím jejich možnosti uplatnění ve ztížených podmínkách. Testovací podmínky nejsou zdaleka tak extrémní, jakou může představovat městská zástavba. Přesto však poskytnou reálná data za ztížených podmínek. Veškeré testy byly prováděny za krásných slunných podmínek bez mraků. Před jakýmkoli testováním byly GNSS moduly nejméně hodinu spuštěny.

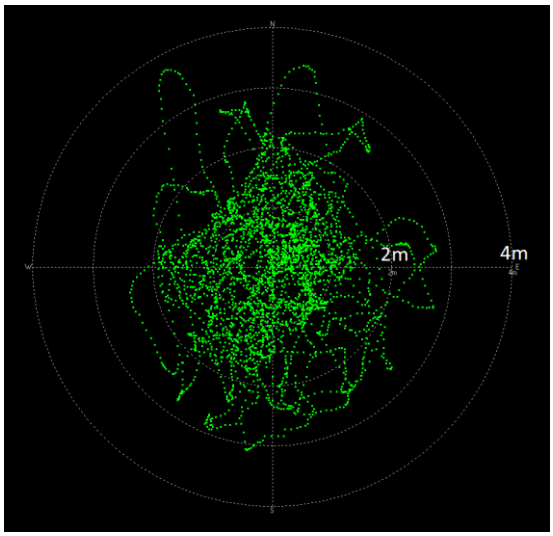
Obrázek 35 Testovací plocha (62) Zdroj: mapy.cz

4.5.1 Porovnání GNSS modulů

První provedený test si klade za cíl přiblížit rozdíl v přesnosti modulů M8P s běžným modulem M8N. Metodika spočívá v ponechání každého testovaného modulu, v případě M8P obou modulů, zapnutého přibližně jednu hodinu. Po uplynulí této doby zahájit sběr dat z modulů programem u-center. Měření probíhá rovněž přibližně hodinu.

Celý test probíhal na zahradě v místě s co možná největším výhledem na oblohu. Moduly (respektive anténa v případě M8P modulu) byly umístěny přibližně 20 cm nad zemí. Po celou dobu experimentu byla modrá obloha bez mraků.

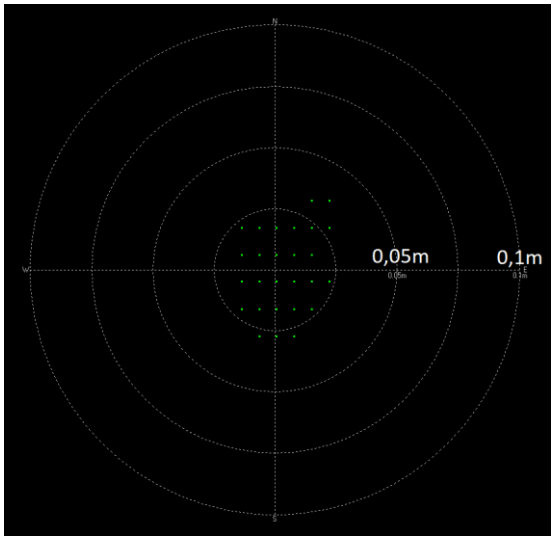
Prvním testovaným modulem je Radiolink TS100. Po celou dobu testu měl modul 3D fix. Mapa odchylek po dobu hodinového měření vypadá následovně.



Obrázek 36 - Mapa odchylek M8N Zdroj: u-center

Z mapy je patrné, že navigace v prostředí zahrady o rozloze 16 x 16 metrů by byla pouze na základě dat z tohoto modulu zcela nedostatečná.

Dalším testovaným modulem je implementovaný kit C94-M8P, který byl po celou dobu testu v módu RTK FIXED

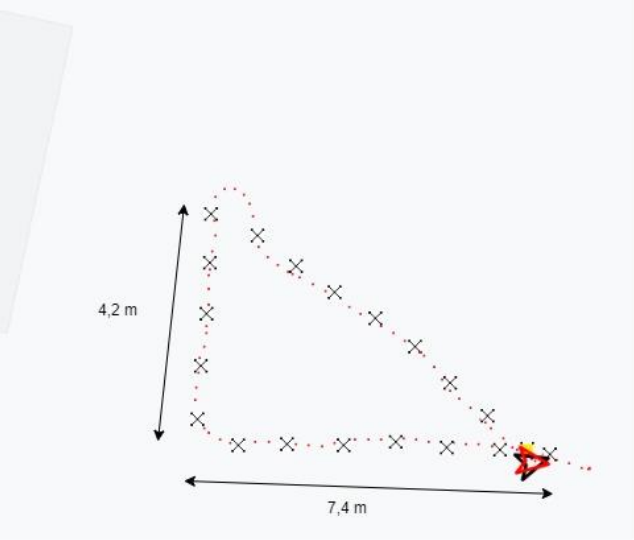


Obrázek 37 - Mapa odchylek M8P Zdroj: u-center

Z mapy odchylek je u tohoto modulu zřejmé, že má lepší predispozice k lokalizaci oproti modulu M8N.

4.5.2 Testování 1. trasy

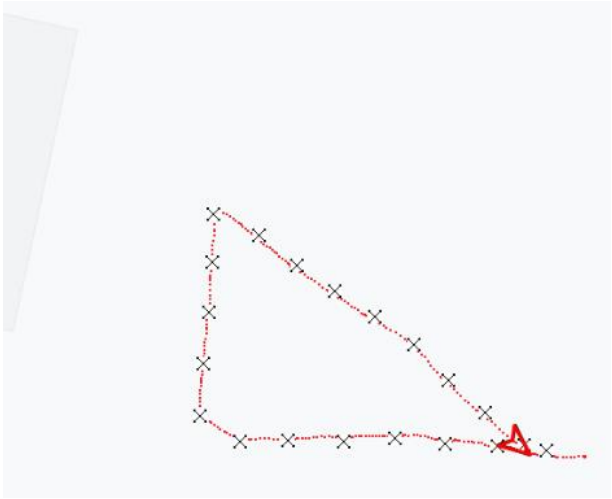
Dalším testem bylo samotné projetí trasy robotem. Tento test se skládal ze tří částí. V první části byla manuálně projeta trasa ve tvaru trojúhelníku, při které se ukládaly zeměpisné souřadnice. Pro autonomní testování je zaznamenaná trasa automaticky upravena tak, aby obsahovala pouze body v určité nastavené vzdálenosti od sebe. V tomto případě šlo o 1 metr. Druhá fáze testování spočívala v autonomním projetí zaznamenané trasy plnou rychlostí. Zatímco robot autonomně projížděl celou trasu, byli na náhodné otočné body vloženy kartičky pro pozdější srovnání reálné přesnosti projeté trasy.



Obrázek 38 - 1. trasa rychlý průjezd Zdroj: maps.google.com

Obrázek autonomně projeté trasy při nejvyšší rychlosti ukazuje, že GNSS modul poskytuje velmi dobrou přesnost. Záznam vypovídá, že navigační program, respektive PID regulátor není optimálně nastavený a u horního vrcholu cesty vytvořil menší s křivku. Další fází bylo opětovné projetí trasy a měření odchylek mezi průjezdem otočným bodem a kartičkou, která se pohybovala v rozmezí od 5 do 10 cm. V nejhorším případě pak 20 cm.

Finálním testem první trasy byl průjezd velmi malou rychlostí, který měl omezit vliv navigačního systému.

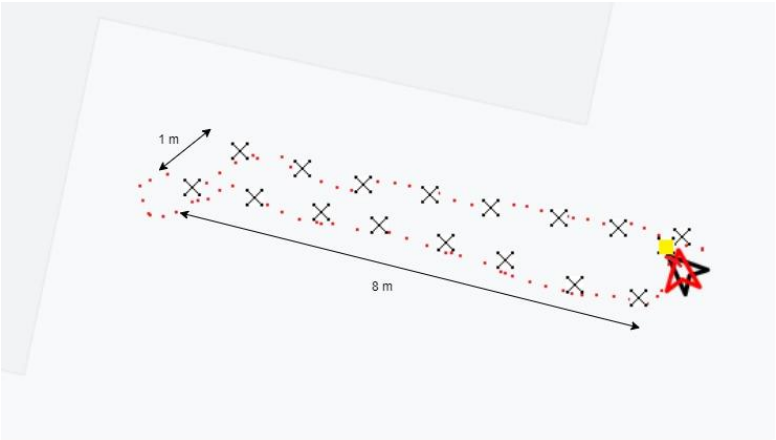


Obrázek 39 - 1. trasa pomalý průjezd Zdroj: maps.google.com

Jak je vidět z projeté mapy, je zde mnohem větší hustota naměřených dat z důvodu malé rychlosti. Je také patrná lepší návaznost projetých otočných bodů, ovšem kromě plynulejších pohybů platformy zůstává velmi podobná reálná přesnost.

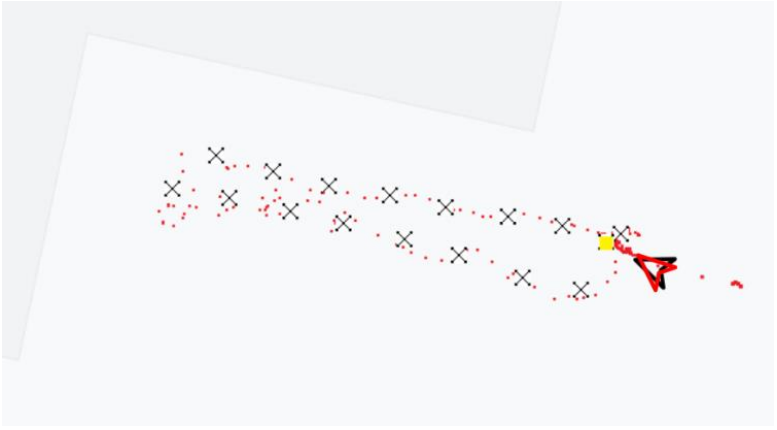
4.5.3 Testování 2. trasy

Dalším provedeným testem je projetí trasy, v plném zákrytu „výhledu“ na severní a západní světovou stranu. Testovací trasa vedla přibližně 1,6m od zdi budovy. Testování probíhalo ve stejných krocích jako test předešlý.



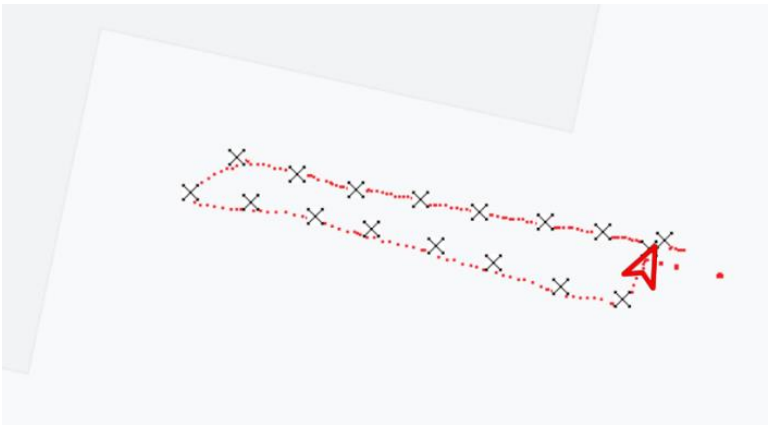
Obrázek 40 - 2. trasa rychlý průjezd Zdroj: maps.google.com

Tento test znovu ukazuje, že navigační program potřebuje nutně implementaci zpomalení v případě prudké zatáčky.



Obrázek 41 - 2. trasa rychlý průjezd číslo 2 Zdroj: maps.google.com

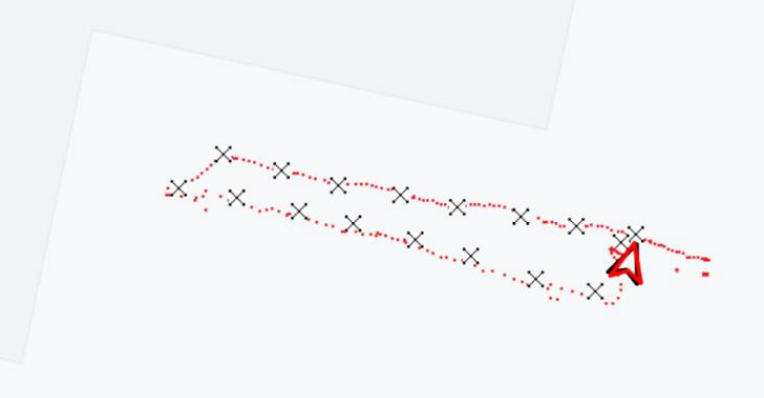
Zajímavý je ale test druhého rychlého průjezdu, kdy se razantně zvýšila nepřesnost GNSS modulu a odchylka od náhodného měřeného bodu v nejhorším případě dosáhla 70 cm.



Obrázek 42 - 2. trasa pomalý průjezd Zdroj: maps.google.com

V případě pomalého průjezdu trasy, lze vidět relativně čistý průjezd. Který odpovídal skutečnosti a robot bez sebemenších problémů projel zadanou trasu.

Poslední testovanou jízdou je druhý pomalý průjezd trasou, který byl opět provázen kolísáním hodnot z GNSS modulu, který zapříčinil přibližně 50 cm maximální odchylku od měřeného bodu.



Obrázek 43 - 2. trasa pomalý průjezd 2 Zdroj: maps.google.com

4.5.4 Testování kamery

Jedním z testů bylo autonomní projetí bodů, kde v každém bodě byli pořízeny tři snímky v požadované pozici kamery. Po projetí všech bodů byl vygenerován soubor obsahující informace o absolvované trase s příslušnými otočnými body, které obsahují odkazy na příslušné snímky.

4.5.5 Testování vyhýbání se překážkám

Posledním testem je schopnost vyhýbání se překážkám, které se mohou na cestě vyskytnout. Pro účely testování byla využita plastová krabice, která byla umístěna do plánované cesty. Následně pak z různých stran, aby se nenacházela v přímé linii plánovaného průjezdu. Robot byl schopen vyhnout se překážce, avšak tento výsledek nebyl pravidlem. V případech umístění krabice mimo přímou linii průjezdu, byl robot schopen téměř bezproblémového průjezdu. V případech, kdy byla krabice umístěna přímo v linii neměl většinou robot problém její detekce a začal se překážce vyhýbat. Hlavní problém nastal v objížděcím manévru, kdy se v mnoha případech vyskytl problém s boční detekcí překážky. Robot tedy nebyl v některých případech schopen bezpečně objet testovací překážku.

5 Výsledky a diskuse

Obrázek 44 - finální podoba stroje Zdroj: vlastní tvorba

Robot obsahuje nejnutnější komponenty pro lokalizaci a jednoduchou detekci okolních vzdáleností. Nejnutnější pro dosažení geografického bodu je informace o aktuální poloze (GNSS) a směru (magnetometr). Instalované senzory jsou toho schopny. Robot je schopen na základě zadané trasy popsané množinou otočných bodů, které obsahují geografické souřadnice a název bodu, autonomně projet tyto body. Může být definována rutina vztahující se ke konkrétnímu názvu bodu, například v jakých polohách se mají nasnímat obrázky okolí. Po ukončení průjezdu trasy je vygenerován záznam o jejím průjezdu, včetně obrazových záznamů a ten je dostupný v rámci webového rozhraní. Robot je dále schopen přímého řízení, kdy je možné uložit manuálně projetou trasu k pozdější autonomní jízdě. Robot je dále schopen autonomního objetí jednoduché překážky. Co se týká přesnosti projetí otočných bodů, jako projetý bod se uznává okruh ve vzdálenosti 30 cm od tohoto bodu. Při

praktickém testování se ukázal menší problém s přesností navigace v bezprostřední blízkosti budovy, kde byla odchylka od požadovaného bodu až 70 cm. V ostatních případech robot neměl sebemenší problém projet celou trasu s typickou maximální odchylkou 10 cm.

Následující kapitoly budou věnovány výsledkům dílčích částí robotu a aplikace, závěr každé kapitoly obsahuje potenciál pro zlepšení.

5.1 Konstrukce

Konstrukce robotu je realizována na čtyřkolovém podvozku. Prototyp rovněž obsahuje nástavbu s elektronikou v podobě plastových krabic. Motory poskytují dostatečnou sílu k pohybu robotu. Pro testovací účely se tato platforma osvědčila. Jednou ze slabin jsou použitá kola, která v případě nízkých teplot sníží svoji přilnavost. Robot se může na namrzlé trávě při pomalém pohybu snadno zastavit.

Potenciální vylepšení, co se konstrukce týče, je použití většího podvozku z hlediska rozvoru a délky, který by poskytl dostatečný prostor pro elektroniku a baterii. To by pomohlo ke snížení těžiště a větší stabilitě platformy. Zvětšení podvozku by mohlo také vést ke zkrácení délky datových a napájecích vodičů a tím potenciálního snížení rušení.

Předcházení zastavení robotu (drift) může být pouhá úprava software, případně použití jiných kol, záleží na konkrétní aplikaci a jejím požadavkům.

V případě dalšího vývoje nebo nasazení je také potřeba zabezpečit robota vůči nízkým teplotám, dešti a případnému sněhu.

5.2 Elektronika a senzory

Obsažená elektronika je rozdělena do jednotlivých nezávislých subsystémů. Řešení je tedy do jisté míry modulární. K hlavnímu počítači lze snadno připojit další zařízení, které se s patřičným programovým balíčkem může začlenit do systému. Jde o velmi praktické řešení z hlediska vývoje a testování. Každý modul obsahuje svojí výpočetní jednotku a zdroj napájení, takže je přímo schopný komunikovat s hlavním programem a lze jej snadno připojit k energetickému rozvodu. Během testování může být modul připojen k počítači, kde lze modifikovat program, případně komponenty. Následně jej lze opět jednoduše vrátit zpět.

Co se modulů týče, velmi dobře pracují moduly pro řízení motorů a řízení serv s magnetometrem. Velkým překvapením jsou také RTK GNSS moduly. Naopak zklamáním je systém detekce překážek, který neposkytuje konzistentní výsledky a detekce není zcela spolehlivá.

Potenciální vylepšení stroje z hlediska lokalizace by mohla být inerciální jednotka. Spolu s odometrií a aplikací Kalmanova filtru, by mohlo dojít k potenciálnímu vylepšení lokalizace. Co se týče systému detekce překážek, jednoduché zlepšení by bylo přidání 1D Lidaru na servomechanismus. To by pomohlo ultrazvukovým senzorům při detekci překážek. Případnou překážku by pak bylo možné sledovat a lépe na ni reagovat. Bohužel toto řešení během testování nemělo požadovanou funkčnost, kterou přikládám rozvodu energie a levných serv. Další možností je použití Lidaru schopného snímat v rozsahu 360°, který je schopen pracovat ve venkovním prostředí, což ovšem téměř zdvojnásobí cenu robotu. Dalším řešením by také mohla být implementace stereovize, rozpoznání obrazu a algoritmy hlubokého učení. To je však velmi náročné řešení s nejistými výsledky, které vyžaduje dlouhodobé úsilí.

Pro dlouhodobé operace je zcela zásadní vyřešení dokování robotu, které by pak sloužilo minimálně jako nabíjecí stanice, v ideálním případě pak jako ochrana před vnějšími vlivy. Samotná implementace by v konečném případě ani nevyžadovala aktivní prvek na straně dokovací stanice. GNSS systém je natolik přesný, aby mohl platformu navést na pozici, kde by využitím konstrukce dokovací stanice a senzorů vzdálenosti bylo možné přesného navádění na napájecí konektor. Je však pravdou, že současná konfigurace senzorů by nemusela být pro tento manévr dostačující. Napájecí konektor na straně robotu by pak mohl obsahovat měřič napětí a v případě správného napětí a polarity by pak mohl přímo přes napěťový měnič s regulací proudu nabíjet palubní baterii.

5.3 Hlavní program

Implementace hlavního programu zajišťuje chod jádra systému včetně jeho nódů, systém je relativně nezávislý na platformě, je potřeba pouze interpret jazyka Python a menší množství knihoven. Systém je nenáročný na HW a snadno rozšiřitelný. Ovšem není zdaleka tak robustní jako framework ROS. Jeho slabinou je, absence přímé podpory paralelního

zpracování nebo implementace souběžnosti. Hlavní motivací bylo vytvoření jednoduchého systému s co možná nejmenším dopadem na HW prostředky.

Potenciální vylepšení hlavního programu a zároveň jeho velká slabina spočívá v nemožnosti přihlášení k odběru témat z koncových zařízení. Jedná se o komunikační mosty mezi mikrokontroléry a webovým rozhraním. Další vylepšení by spočívalo primárně ve sjednocení a normalizaci rozhraní, které by vneslo větší přehlednost do systému. Zejména sjednocení nódů, které by nově musely obsahovat seznam přihlášených nebo publikovaných témat, případně parametrů. Nerealizovanou ideou byla také implementace automatických ladicích algoritmů, které by, například automaticky kalibrovali magnetometr či umožnili automatickou kalibraci odometrie na základě počtu tiků enkodérů a naměřené vzdálenosti z GNSS.

5.4 Webová aplikace

Webová aplikace zastupuje roli poskytování obsahu webovým prohlížečům. Aplikace samotná neobsahuje téměř žádnou funkcionalitu, kromě zpracování a poskytování multimediálního obsahu z absolvovaných cest. Veškerá další komunikace je zajištěna pomocí Socket.IO pluginu.

Potenciální zlepšení webové aplikace je implementace autorizace a případných rolí. Lze nastavit jednotlivá práva osobám, respektive rolím v interakci se zařízením.

5.5 Webové rozhraní

Implementace webového rozhraní je velmi praktická z hlediska použitelnosti zařízení, není potřeba instalace žádného dodatečného softwaru a stroj je možné ovládat a zobrazit jeho stav. Lze si prohlédnout galerii pořízených snímků z jednotlivých cest. Cílem bylo vytvořit prostředí, které nebude náročné a bude obsahovat co nejmenší množství frameworků a knihoven.

Vylepšení použitelnosti by dopomohla možnost veškerého nastavení přes webové rozhraní. Rovněž by zde mohla být implementace importu a exportu nastavení a tras. Důležitým vylepšením je také redesign stránek do uživatelsky příjemného a použitelného stavu. Současné řešení sice neobsahuje mnoho frameworků, ale také není zcela přehledné a

použitelné. Případný budoucí vývoj by spočíval v nahrazení Pure CSS a jQuery za framework BootstrapVue.

6 Závěr

Cílem práce bylo navrhnout a zkonstruovat prototyp (demonstrátor) autonomního stroje, který je automaticky schopen pohybu v okolí budov, snímání obrazu budov a okolí.

V teoretické části bylo nejprve představeno několik příkladů robotických zařízení, které souvisejí s cílem této práce. Představují nám venkovní autonomní stoje a poukazují na jejich senzorické uspořádání, které dovoluje jejich lokalizaci. Lokalizace je námětem i dalších kapitol teoretické části. V první řadě jsou popsány základní principy pro určení polohy, respektive principy určení vzdáleností. Na toto téma navazuje samotná aplikace těchto principů v podobě rozdělení lokalizačních, případně navigačních technik, které se zabývají postupy a metodami, jak lze realizovat robotickou lokalizaci, případně navigaci. Následující kapitola je věnována různorodým senzorům, které souvisejí s navigací. Jsou zde zmíněny senzory relativní i absolutní lokalizace. Každý příklad obsahuje princip jeho činnosti.

V návaznosti na senzory je dále popsána řídící elektronika, která je nezbytnou součástí robotických aplikací. Kapitola popisuje nejprve malé počítače a přechází k mikrokontrolérům.

Následuje část věnovaná komunikačním protokolům, kde jsou letmo zmíněna fyzická komunikační rozhraní, která přísluší senzorům a mikrokontrolérům. Okrajově je zde také zmíněna problematika komunikačních protokolů a příklady využitelných protokolů aplikační vrstvy. Detailněji jsou popsány formáty dat pro komunikaci zejména pak z pohledu mikrokontrolérů. Závěr teoretické části nastiňuje možnost použití frameworku pro vývoj robotické aplikace, případně popisuje autopiloty, které již řeší navigaci.

Praktická část v úvodu přináší výběr robotické platformy a výsledného navigačního systému. Detailně jsou dále popsány jednotlivé komponenty robotické platformy. Navazuje popis použitého softwaru a knihoven, je zde popsán hlavní program a některé jeho komponenty spolu s webovou aplikací.

Další kapitola se zaměřuje na popis jednotlivých subsystémů, ze kterých je robot tvořen.

Kapitola testování přináší informace o provedených testech, ukazuje výsledky projetých tras a orientační srovnání přesnosti GNSS RTK modulů M8P s běžným modulem M8N.

Poslední kapitola se zabývá popisem, jak je robot realizován, jeho případné klady a zápory spolu s potenciálními možnostmi vylepšení z různých hledisek.

Výsledkem této práce je prototyp robotického zařízení schopného autonomních operací. Zařízení zvládne uskutečnit zadanou cestu, při které může pořizovat snímky okolí. Stroj má webové rozhraní, k ovládání nebo zobrazení projetých tras stačí pouze webový prohlížeč. Reálné uplatnění prototypu by však vyžadovalo další vývoj.

7 Seznam použitých zdrojů

1. Husky UNMANNED GROUND VEHICLE. *Clearpathrobotics.* [Online] [Citace: 14. Březen 2020.] https://clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/.

2. Technology of Autonomous Mobile Robots. *Smprobotics.* [Online] [Citace: 14. Březen 2020.] https://smprobotics.com/technology_autonomous_mobile_robot/.

3. Wild Thumper based ROS robot. *Hackaday.* [Online] [Citace: 14. Březen 2020.] https://hackaday.io/project/25406-wild-thumper-based-ros-robot.

4. Vedder, Benjamin , Vinter, Jonny a Jonsson, Magnus. A Low-Cost Model Vehicle Testbed with Accurate Positioningfor Autonomous Driving. *Hindawi.* [Online] [Citace: 14. Březen 2020.] https://www.hindawi.com/journals/jr/2018/4907536/.

5. Dlouhý, Martin a Winkler, Zbyněk. Úvod do mobilní robotiky. *Robotika.* [Online] 20. Srpen 2005. [Citace: 14. Březen 2020.] https://robotika.cz/guide/umor05/cs.

6. Geodetické základyBodová pole a sítě bodů. *Mendelova univerzita v Brně.* [Online] [Citace: 18. Břenen 2020.] http://uhulag.mendelu.cz/files/pagesdata/cz/geodezie/geodezie_2018/geodeticke_zaklady.pdf.

7. Location Tracking Approaches. *Cisco.* [Online] [Citace: 14. Březen 2020.] https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Mobility/WiFiLBS-DG/wifich2.html.

8. Line of Sight Obstruction. *Campbell Scientific Companies.* [Online] [Citace: 19. Březen 2020.] https://s.campbellsci.com/documents/au/technical-papers/line-of-sight-obstruction.pdf.

9. Svoboda, Petr. Radiotechnické pasivní systémy PRP-1 KOPÁČ, RAMONA, TAMARA. *Československá lidová armáda.* [Online] [Citace: 14. Březen 2020.] http://www.csla.cz/technika/specialtech/tamara.htm.

10. VOJENSKÉ TECHNOLOGIE. *ERA.* [Online] [Citace: 14. Březen 2020.] https://www.era.aero/cs/military-security.

11. Novák, Petr. *Mobilní roboty - pohony, senzory, řízení.* místo neznámé : BEN - technická literatura, 2005. 80-7300-141-1.

12. Michalec, Libor. Když GPS nestačí, inerciální měření IMU mu pomůže. *vyvoj.hw.* [Online] 15. Leden 2019. [Citace: 14. Březen 2020.] https://vyvoj.hw.cz/kdyz-gps-nestaci-inercialni-mereni-imu-mu-pomuze.html.

13. Dead Reckoning (DR). *Furuno.* [Online] [Citace: 14. Březen 2020.] https://www.furuno.com/en/gnss/technical/tec_dead.

14. Ochieng, Washington, a další. A General Map Matching Algorithm for Transport Telematics Applications. *Researchgate.* [Online] 2003. https://www.researchgate.net/publication/48353309_A_General_Map_Matching_Algorithm_for_Transport_Telematics_Applications.

15. Burian, František. Robotika. *uamt.feec.vutbr.* [Online] 25. Březen 2013. [Citace: 14. Březen 2020.] http://www.uamt.feec.vutbr.cz/%7Erobotika/prednasky/2014_BROB_08_MapovaniANavigace.pdf.

16. Skalka, Marek. Srovnání lokalizačních technik v robotice. *marek.sk.sweb.* [Online] [Citace: 14. Březen 2020.] http://marek.sk.sweb.cz/lokalizace/kapitola4.html.

17. Winkler, Zbyněk. Odometrie. *Robotika.* [Online] 5. Prosinec 2005. [Citace: 14. Březen 2020.] https://robotika.cz/guide/odometry/cs.

18. Principle of operation of an optical encoder. *Researchgate.* [Online] [Citace: 14. Březen 2020.] https://www.researchgate.net/figure/Principle-of-operation-of-an-optical-encoder_fig1_324777358.

19. Collins, Danielle. FAQ: How do magnetic encoders work? *Design World.* [Online] 13. Listopad 2015. [Citace: 14. Březen 2020.] https://www.designworldonline.com/faq-how-do-magnetic-encoders-work/.

20. Vojáček, Antonín. Rotační enkodéry pro řízení pohonů a pohybu. *automatizace.hw.* [Online] 30. Duben 2017. [Citace: 14. Březen 2020.] https://automatizace.hw.cz/rotacni-enkodery-pro-rizeni-pohonu-a-pohybu.html.

21. Juránek, Martin. Prostředky automatického řízení. *homel.vsb.* [Online] [Citace: 14. Březen 2020.] http://homel.vsb.cz/~jur286/prostredky_aut_rizeni/preklad.htm.

22. Vojáček, Antonín. Integrované MEMS GYROSKOPY. *automatizace.hw.* [Online] 11. Říjen 2009. [Citace: 14. Březen 2020.] https://automatizace.hw.cz/integrovane-mems-gyroskopy.

23. Informace o ultrazvukových senzorech. *Pepperl-Fuchs.* [Online] 21. Leden 2014. [Citace: 14. Březen 2020.] https://www.pepperl-fuchs.com/czech_republic/cs/24907.htm.

24. Mouse-based Optical Flow Sensor (ADNS3080). *ArduPilot.* [Online] [Citace: 14. Březen 2020.] http://ardupilot.org/copter/docs/common-mouse-based-optical-flow-sensor-adns3080.html.

25. *RoboDoupě.* [Online] [Citace: 20. Březen 2020.] http://robodoupe.cz/tag/senzory/.

26. What is LiDAR technology? *Generationrobots.* [Online] [Citace: 2. Duben 2020.] https://www.generationrobots.com/blog/en/what-is-lidar-technology/.

27. Kilián, Karel. Čím se LIDAR liší od radaru a jaká je jeho role v autonomních vozidlech. *vtm.zive.* [Online] 12. Říjen 2018. [Citace: 2. Duben 2020.] https://vtm.zive.cz/clanky/cim-se-lidar-lisi-od-radaru-a-jaka-je-jeho-role-v-autonomnich-vozidlech/sc-870-a-195431/default.aspx.

28. Nichols, Greg. AM vs FM: The battle brewing in lidar technology. *ZDNet.* [Online] [Citace: 2. Duben 2020.] https://www.zdnet.com/article/am-vs-fm-the-battle-brewing-in-lidar-technology/#ftag=RSSbaffb68.

29. Elektronický kompas a gyroskop. *vyvoj.hw.* [Online] 17. Leden 2003. [Citace: 14. Březen 2020.] https://vyvoj.hw.cz/teorie-a-praxe/konstrukce/elektronicky-kompas-a-gyroskop.html.

30. Štroner, Martin. Globální navigační satelitní systémy (GNSS). *ČVUT.* [Online] [Citace: 14. Březen 2020.] http://k154.fsv.cvut.cz/vyuka/geodezie_geoinformatika/vy1/OBS/GNSS_obs.pdf.

31. Švábensý, Otakar, Weigel, Josef a Machota, Radovan. Seminář GPS. *ČVUT.* [Online] [Citace: 14. Březen 2020.] http://fast.darmy.net/opory%20-%20IV%20nMgr/HE09_M01-Semin%C3%A1%C5%99%20GPS.pdf.

32. Informace o službách a produktech. *Zeměměřický úřad.* [Online] [Citace: 14. Březen 2020.] http://czepos.cuzk.cz/_servicesProducts.aspx.

33. Wurms, Gina. Technologies for Server-Based Indoor Positioning Compared: Wi-Fi vs. BLE vs. UWB vs. RFID. *Infsoft.* [Online] 13. Červenec 2017. [Citace: 1. Duben 2020.] https://www.infsoft.com/blog/technologies-for-server-based-indoor-positioning-compared.

34. Určování polohy v reálném čase s přesností na 10 cm. *Vyvoj.hw.* [Online] 19. Červenec 2018. [Citace: 1. Duben 2020.] https://vyvoj.hw.cz/urcovani-polohy-v-realnem-case-s-presnosti-10-cm.html.

35. Connell, Ciaran. What's The Difference Between Measuring Location By UWB, Wi-Fi, and Bluetooth? *ElectronicDesign.* [Online] 6. Únor 2015. [Citace: 16. Březen 2020.] https://www.electronicdesign.com/technologies/communications/article/21800581/whats-the-difference-between-measuring-location-by-uwb-wifi-and-bluetooth.

36. A Hierarchical Algorithm for Indoor Mobile RobotLocalization Using RFID Sensor Fusion. [Online] [Citace: 14. Březen 2020.] https://lucidar.me/en/bibliography/files/a-hierarchical-algorithm-for-indoor-mobile-robot-localization-using-rfid-sensor-fusion.pdf.

37. Co je to strojové vidění a jak může pomoci? *Vše o průmyslu.* [Online] 30. Březen 2019. [Citace: 16. Březen 2020.] https://www.vseoprumyslu.cz/automatizace/kontrola-procesu/co-je-to-strojove-videni-a-jak-muze-pomoci.html.

38. The list of top 10 IoT boards for development and prototyping in 2020 is dominated by a mix of Single Board Computers (SBCs), microcontroller boards and cellular-enabled boards... . *Youngwonks.* [Online] 6. Únor 2019. [Citace: 6. Duben 2020.] https://www.youngwonks.com/blog/Top-10-IoT-boards-for-2019.

39. *Arduino.* [Online] [Citace: 4. Duben 2020.] https://www.arduino.cc/.

40. Vývojová deska ARM STM32F103C8T6 STM32. *Arduino Návody.* [Online] 20. Únor 2018. [Citace: 4. Duben 2020.] https://navody.arduino-shop.cz/navody-k-produktum/vyvojova-deska-arm-stm32f103c8t6-stm32.html.

41. *Espressif.* [Online] [Citace: 4. Duben 2020.] https://www.espressif.com/en/products/hardware/esp32/overview.

42. Messaging Patterns Overview. *Enterprise Integration Patterns.* [Online] [Citace: 15. Březen 2020.] https://www.enterpriseintegrationpatterns.com/patterns/messaging/.

43. Data Serialization Comparison: JSON, YAML, BSON, MessagePack. *SitePoint.* [Online] 8. Listopad 2016. [Citace: 13. Březen 2020.] https://www.sitepoint.com/data-serialization-comparison-json-yaml-bson-messagepack/.

44. Protocol buffers. *Google.* [Online] [Citace: 13. Březen 2020.] https://developers.google.com/protocol-buffers.

45. Instruction. *Capnproto.* [Online] [Citace: 13. Březen 2020.] https://capnproto.org/.

46. Mazzari, Vanessa. ROS – Robot Operating System. *Generationrobots.* [Online] 26. Březen 2016. [Citace: 15. Březen 2020.] https://www.generationrobots.com/blog/en/ros-robot-operating-system-2/.

47. Documentation. *ROS.* [Online] [Citace: 14. Březen 2020.] http://wiki.ros.org/.

48. ArduPilot Documentation. *ArduPilot.* [Online] [Citace: 15. Březen 2020.] https://ardupilot.org/ardupilot/.

49. *Banggood.* [Online] https://www.banggood.com.

50. *Gme.* [Online] [Citace: 4. Duben 2020.] : https://www.gme.cz/oloveny-akumulator-westinghouse-wa1250-12v-5ah-6-3mm.

51. *Raspberrypi.* [Online] [Citace: 4. Duben 2020.] https://www.raspberrypi.org/.

52. *Rpishop.* [Online] https://rpishop.cz.

53. *Aliexpress.* [Online] [Citace: 28. Březen 2020.] aliexpress.com.

54. 3-Axis Digital Compass ICHMC5883L. *Adafruit.* [Online] [Citace: 4. Duben 2020.] https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf.

55. *u-blox.* [Online] [Citace: 4. Duben 2020.] https://www.u-blox.com/en/product/c94-m8p.

56. *Digikey.* [Online] [Citace: 26. Březen 2020.] https://www.digikey.cz/product-detail/en/u-blox-america-inc/C94-M8P-2-10/672-C94-M8P-2-10-ND/6150728#images.

57. Simpleserialization. *Google Code.* [Online] [Citace: 4. Duben 2020.] https://code.google.com/archive/p/simpleserialization/.

58. Susnea, Ioan, Mînzu, Viorel a Vasiliu, G. Simple, real-time obstacle avoidance algorithm for mobile robots. *Researchgate.* [Online] Leden 2009. [Citace: 28. Březen 2020.] https://www.researchgate.net/publication/228955195_Simple_real-time_obstacle_avoidance_algorithm_for_mobile_robots.

59. Flask's documentation. *Palletsprojects.* [Online] [Citace: 4. Duben 2020.] https://flask.palletsprojects.com/en/1.1.x/.

60. *Socket.IO.* [Online] [Citace: 4. Duben 2020.] https://socket.io/.

61. Centimeter-Level Positioning with Single-band RTK. *Taoglas.* [Online] 18. Říjen 2018. [Citace: 3. Únor 2020.] https://www.taoglas.com/centimeter-level-positioning-with-single-band-rtk/.

62. *Mapy.* [Online] mapy.cz.

63. Ray, Brian. How An Indoor Positioning System Works. *Airfinder*. [Online] 16. Srpen 2018. [Citace: 14. Březen 2020.] https://www.airfinder.com/blog/indoor-positioning-system.

8 Seznam obrázků

Obrázek 1- Husky (1) 14
Obrázek 2 - SMP robot (2) 15
Obrázek 3 - Wild Thumper schéma (3) 16
Obrázek 4 - Wild Thumper (3) 16
Obrázek 5 - SDVP schéma testovací platformy (4) 16
Obrázek 6 - AoA metoda (6) 17
Obrázek 7 - ToA metoda (6) 18
Obrázek 8 - TDoA metoda (6) 19
Obrázek 9 - CoO schéma (6) 20
Obrázek 10 - Optický enkodér (14) 23
Obrázek 11 - Magnetický enkodér (15) 23
Obrázek 12 - princip MEMS gyroskopu (18) 24
Obrázek 13 - Optický senzor (20) 25
Obrázek 14 Ukázka mapování pomocí LIDAR (23) 26
Obrázek 15 - porovnání pozičních systémů (28) 28
Obrázek 16 - Schéma vzory pro přenos zpráv (38) 31
Obrázek 17 - ROS schéma (42) 34
Obrázek 18 - Pixhawk schema (44) 35
Obrázek 19 - Podvozek (45) 39
Obrázek 20 - Motor (45) 39
Obrázek 21 - h-můstek (45) 40
Obrázek 22 - Kolo (45) 40
Obrázek 23 - Baterie (46) 41
Obrázek 24 - RPi 2B (47) 41
Obrázek 25 - Arduino mikrokontroléry (48) 42
Obrázek 26 - RPi kamera (47) 43
Obrázek 27 - Kamerový držák (48) 43
Obrázek 28 - Kompas hmc588l (48) 44
Obrázek 29 - GNSS kit (49) 44
Obrázek 30 - Ultrazvukový senzor HC-SR04 (45) 45

Obrázek 31 - Diagram programových komponent Zdroj: vlastní tvorba 45
Obrázek 32 - Diagram systémů Zdroj: vlastní tvorba..... 50
Obrázek 33 - řídící GUI Zdroj: vlastní tvorba 53
Obrázek 34 - záznam z projeté trasy GUI Zdroj: vlastní tvorba..... 54
Obrázek 35 Testovací plocha (52) Zdroj: mapy.cz..... 55
Obrázek 36 - Mapa odchylek M8N Zdroj: u-center 56
Obrázek 37 - Mapa odchylek M8P Zdroj: u-center..... 56
Obrázek 38 - 1. trasa rychlý průjezd Zdroj: maps.google.com..... 57
Obrázek 39 - 1. trasa pomalý průjezd Zdroj: maps.google.com..... 58
Obrázek 40 - 2. trasa rychlý průjezd Zdroj: maps.google.com..... 58
Obrázek 41 - 2. trasa rychlý průjezd číslo 2 Zdroj: maps.google.com 59
Obrázek 42 - 2. trasa pomalý průjezd Zdroj: maps.google.com..... 59
Obrázek 43 - 2. trasa pomalý průjezd 2 Zdroj: maps.google.com..... 60
Obrázek 44 - finální podoba stroje Zdroj: vlastní tvorba..... 61

9 Přílohy

Zdrojové kódy programů na CD