



Vyšší odborná škola
a Střední průmyslová škola
elektrotechnická
Plzeň, Koterovská 85

DLOUHODOBÁ MATURITNÍ PRÁCE

Téma: Sonar Car

Autor práce: Matěj Jun

Třída: 4.L

Vedoucí práce: Jiří Švihla

Dne: 27.3.2024

Hodnocení:



Zadání dlouhodobé maturitní práce

Žák: Matěj JUN
Třída: 4. L
Studijní obor: 78-42-M/01 Technické lyceum

Zaměření: Kybernetika

Školní rok: 2023 - 2024

Téma práce: **Sonar Car**

Pokyny k obsahu a rozsahu práce:

1. Seznámení s vývojovou platformou Raspberry Pi Pico
2. Seznámení s modelovacím softwarem
3. Seznámení s problematikou orientace v prostoru a mapováním
4. Tvorba podvozku a návrh řídicí elektroniky
5. Návrh a testování mapovací jednotky
6. Tvorba mapovacího skriptu
7. Realizace komunikace mezi jednotkami

Plán konzultací:

- | | |
|--------------|---|
| 19. 10. 2023 | Návrh modelu a elektroniky |
| 23. 11. 2023 | Kompletace hardwaru |
| 11. 1. 2024 | Práce na softwarovém vybavení |
| 8. 2. 2024 | Testování mapování a tvorba dokumentace |

Určení částí tématu zpracovávaných jednotlivými žáky:

1. Seznámení s vývojovou platformou Raspberry pi pico
2. Seznámení s problematikou orientace v prostoru a mapováním
3. Naprogramujte mapování místnosti
4. Návrh a testování mapovací jednotky
5. Tvorba mapovacího skriptu
6. Realizace komunikace mezi jednotkami

Požadavek na počet vyhotovení maturitní práce: 2 výtisky

Termín odevzdání: **27. března 2024**
Čas obhajoby: **15 minut**
Vedoucí práce: **Jiří ŠVIHLA**

Projednáno v **katedře ODP** a schváleno ředitelem školy.

Anotace

Cílem této maturitní práce je vytvoření autonomního vozítka schopného pohybu a mapování místnosti. Práce využívá platformu Raspberry Pi Pico a orientaci vozítka zabezpečuje vzdálenostní senzor umístěný na střeše vozu. Pro pohyb jsou implementována všesměrová kola Mecanum Wheels, která umožňují vozítku pohybovat se ve všech směrech.

„Prohlašuji, že jsem tuto práci vypracoval samostatně a použil(a) literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.“

V Plzni dne:

Podpis:

Obsah

1	Úvod	5
2	Cíle a požadavky	6
2.1	Program pro mapování místnosti	6
2.2	Skript pro řízení vozidla na základě mapy	6
3	Použité součástky	7
3.1	Raspberry Pi Pico W	7
3.2	Ultrazvukový senzor vzdálenosti	7
3.3	Krokový motor	8
3.4	Řadič ULN2003	9
3.5	RGB LED	9
3.6	Zapojení součástek	10
4	Ovládání sonaru	11
4.1	Měření vzdálenosti	11
4.2	Otáčení sonaru	12
4.3	Taktika mapování	12
4.4	Vytváření mapy	12
5	Model sonaru	14
5.1	Modelovací prostředí	14
5.2	Výběr rozmístění komponent	14
5.3	Návrh krabice	14
5.4	Modelování dalších komponent	15
5.5	Finální úpravy a optimalizace	16
5.6	Export do formátu pro 3D tisk	16
5.7	3D tisk	16
6	Závěr	18
	Zdroje	20
	Seznam obrázků	21
	Seznam příloh	22

1 Úvod

V dnešní době, kdy technologický pokrok zaujímá stále větší roli v našich životech, nabývá autonomní robotika a mapování prostoru stále většího významu. Autonomní robotika a mapování prostoru jsou klíčové technologie, které ovlivňují širokou škálu průmyslových sektorů, od výroby a logistiky po zdravotnictví a zemědělství. Tyto technologie také hrají roli v našem každodenním životě, například v autonomních vozidlech nebo robotických vysavačích.

Tato maturitní práce si klade za cíl vytvořit malé autonomní vozítko, které se nejenom pohybuje po místnosti, ale také aktivně mapuje své okolí. Využití platformy Raspberry Pi Pico poskytuje robustní základ pro implementaci řídicího systému vozítka. Hlavním prvkem orientace vozidla je ultrazvukový senzor, který je umístěný na střeše a umožní vozítku získávat informace o okolním prostoru. Pro zajištění plynulého pohybu byla zvolena technologie Mecanum Omni Wheels, umožňující vozítku pohybovat se ve všech směrech s výjimečnou obratností.

Tato práce zkoumá nejen technické aspekty vývoje autonomního vozidla, ale také přispívá k pochopení principů autonomní robotiky a mapování prostoru. Výsledky projektu mohou sloužit nejen jako ukázka praktického využití technologií, ale také jako inspirace pro další výzkum v oblasti autonomních systémů.

2 Cíle a požadavky

2.1 Program pro mapování místnosti

Tento program kromě mapování také přijímá aktuální polohu vozítka od skriptu pro jeho řízení. Zahrnuje:

- Vytváření mapy místnosti, kde jedna jednotka bude reprezentovat 10 cm v reálném světě.
- Přijímání aktuální polohy vozítka od skriptu pro řízení vozidla.
- Aktualizaci mapy na základě nových dat z ultrazvukového senzoru.
- Ukládání dat do pole nebo souboru, který reprezentuje mapu místnosti.
- Poskytování těchto informací skriptu pro řízení vozidla na vyžádání.

2.2 Skript pro řízení vozidla na základě mapy

Skript bude řídit pohyb vozidla na základě aktuální mapy místnosti a poskytuje aktuální informace o svém pohybu části programu pro mapování. Zahrnuje:

- Pravidelně aktualizovat program pro mapování o svůj aktuální pohyb, včetně souřadnic.
- Analýzu dat z mapy a aktuální polohy k rozhodování, kam se vozidlo má pohybovat.
- Generování příkazů pro pohyb vozidla o určitý počet souřadnic (např. 10 cm) od překážky.
- Odesílání těchto příkazů programu pro řízení vozítka.
- Tímto způsobem bude projekt umožňovat vozítku pohybovat se v místnosti, mapovat prostor a řídit svůj pohyb na základě aktuální mapy.

3 Použité součástky

Součástka	Rozměry [mm]	Pracovní napětí	Komunikační rozhraní
Raspberry Pi Pico W	51 x 21	3.3V	SPI, I2C, UART
Ultrazvukový senzor	45 x 20 x 15	5V	Digitální signál
Krokový motor 28BYJ-48	34 x 28 x 21	5V	Řadič ULN2003
Řadič ULN2003	40 x 27 x 10	5V	-
RGB LED	5 (poloměr)	3V	Analogový signál

Tabulka 1: Technické specifikace použitých součástek

3.1 Raspberry Pi Pico W

Raspberry Pi Pico W je vývojová deska s mikrokontrolérem, která využívá čip RP2040 od společnosti Raspberry Pi Foundation. Tento čip obsahuje dvoujádrový procesor ARM Cortex-M0+ s taktem až 133 MHz, 264 kB paměti RAM a 2 MB vestavěné paměti Flash. Tyto klíčové vlastnosti čipu RP2040, zejména jeho vysoký výkon a velká paměť, jsou důležité pro tento projekt, protože umožňují rychlé zpracování informací a uchování mapy a dalších dat v paměti.

Deska nabízí 28 GPIO pinů, které umožňují připojení různých periferních zařízení, jako jsou senzory, LED diody, motory a další. V tomto konkrétním projektu jsou tyto piny využívány k ovládání krokového motoru a zpracování dat z ultrazvukového senzoru.

Volba verze s Wi-Fi umožňuje pozdější bezdrátové přenosy mapy a dalších dat, což rozšiřuje možnosti projektu. Tato funkce je klíčová pro tento projekt, protože umožňuje vozítku komunikovat s vnějším světem a poskytovat užitečné informace o svém okolí.

Vývojová deska Raspberry Pi Pico W je vhodná pro širokou škálu aplikací, včetně Internetu věcí (IoT), automatizace, domácí elektroniky a dalších oblastí. Díky své cenové dostupnosti a jednoduchému použití je ideální volbou jak pro začátečníky, tak i pro pokročilé uživatele. Tato flexibilita a snadné použití, spolu s možnostmi, které nabízí autonomní robotika a mapování prostoru, jak bylo diskutováno v úvodu a závěru, činí Raspberry Pi Pico W ideální platformou pro tento projekt.

3.2 Ultrazvukový senzor vzdálenosti

Ultrazvukový měřič vzdálenosti HC-SR04 je kompaktní a cenově dostupný senzor, který měří vzdálenost od překážky pomocí ultrazvukových vln. Senzor má pracovní rozsah od 2 cm do 4 m a přesnost měření až 3 mm. Tyto klíčové vlastnosti senzoru HC-SR04, zejména jeho vysoká přesnost a široký rozsah měření, jsou důležité pro tento projekt, protože umožňují vozítku získávat přesné informace o svém okolí.

Senzor využívá princip odrazu ultrazvukových vln od objektů a měří čas, který trvá, než se vlny vrátí zpět k senzoru. Tento čas je pak převeden na vzdálenost pomocí rychlosti zvuku ve

vzduchu. Tato metoda měření umožňuje senzoru získávat přesné a spolehlivé údaje o vzdálenosti překážek.

Senzor je umístěn na střeše vozítka a je pravidelně otáčen krokovým motorem. Toto uspořádání umožňuje senzoru pokrýt celé okolí vozítka a poskytuje vozítku pohled na své okolí ve všech směrech.



Obrázek 1: Raspberry Pi Pico W [Zdroj: <https://www.amazon.ca/Waveshare-Microcontroller-Compatible-Raspberry-Pico/dp/B0BM3LCC7D>]



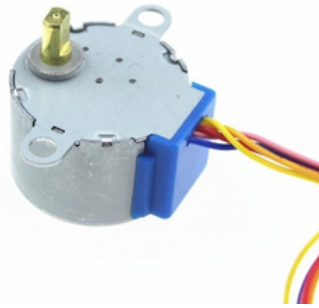
Obrázek 2: HC-SR04 [Zdroj: <https://cz.rs-online.com/web/p/doplňky-bbc-micro-bit/2153181>]

3.3 Krokový motor

Krokový motor 28BYJ-48 je malý a cenově dostupný krokový motor, který se často používá v robotických aplikacích. Motor má 4 fáze a 48 zubů, což znamená, že pro jednu otáčku motoru je potřeba 48 kroků.

Motor pracuje na principu krokování, při kterém se motor otáčí po malém úhlu v každém kroku. Tímto způsobem lze motor ovládat přesně a plynule.

Krokový motor se používá k otočení ultrazvukového senzoru ve všech směrech kolem vozítka. Jeho přesnost a možnost plynulého otočení ho činí ideální volbou pro tento účel. Motor je řízen řadičem ULN2003 a ovládán pomocí signálů z Raspberry Pi Pico.



Obrázek 3: Krokový motor 28BYJ-48 [Zdroj: <https://www.laskakit.cz/krokovy-motor-28byj-48/>]

3.4 Řadič ULN2003

Řadič ULN2003 slouží jako rozhraní mezi Raspberry Pi Pico a krokovým motorem. Zesiluje výstupní signály z mikrokontroléru a řídí napájení motoru, což umožňuje spolehlivé a bezproblémové ovládání motoru.



Obrázek 4: Řadič ULN2003 [Zdroj: <https://www.laskakit.cz/radic-uln2003-pro-krokovy-motor/>]

3.5 RGB LED

RGB LED je zkratka pro Light Emitting Diode (LED) s třemi základními barvami – červenou, zelenou a modrou. Jedná se o speciální typ LED diody, který umožňuje nezávislé ovládání těchto tří barev, což umožňuje vytváření široké škály barevných kombinací.

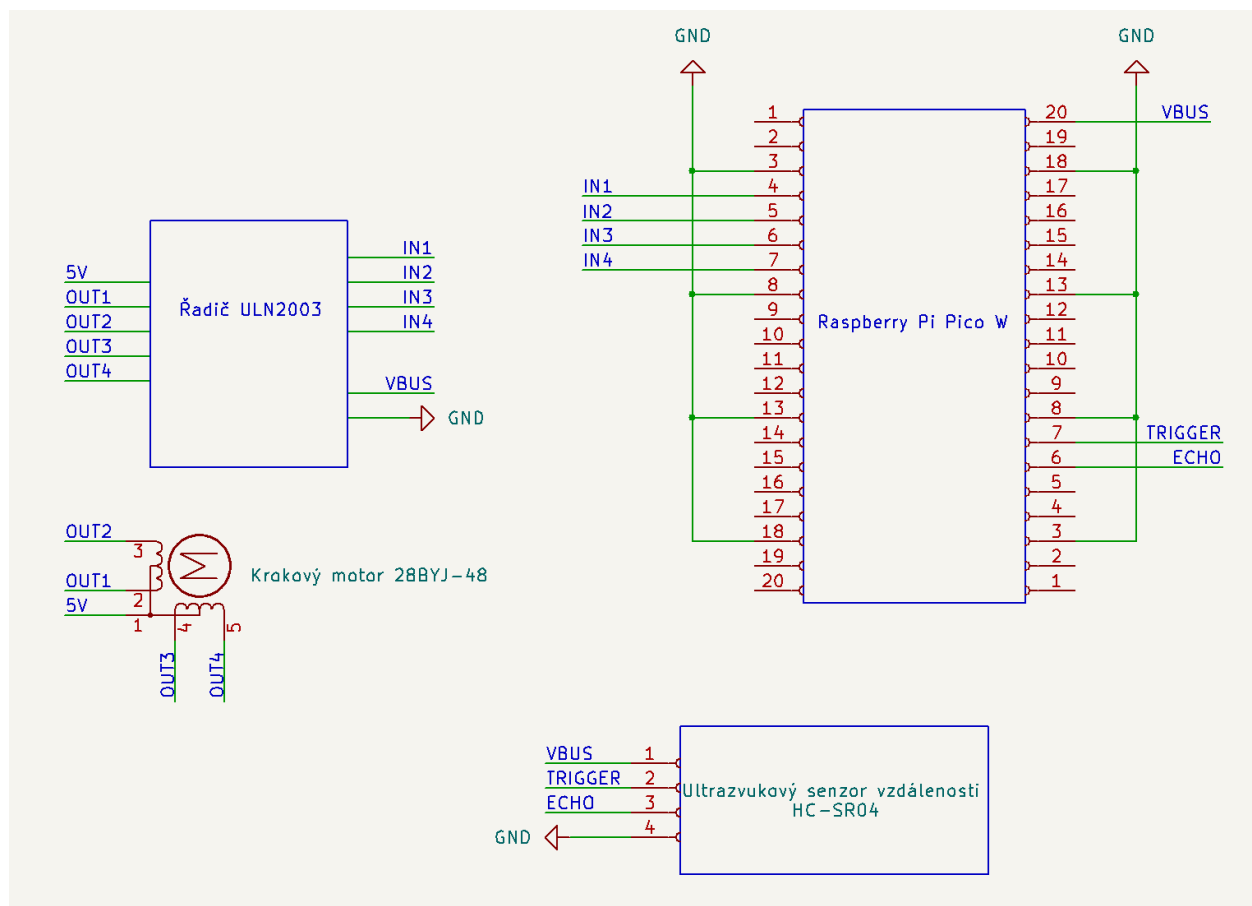
RGB LED se používá jako vizuální indikátor stavu vozítka. Může signalizovat různé události, jako je například detekce překážky, dokončení mapování nebo chybové stavy. Její různobarevné světlo poskytuje uživateli snadnou zpětnou vazbu o stavu vozítka.



Obrázek 5: RGB LED [Zdroj: <https://dratek.cz/docs/produkty/1/1298/1434544602.pdf>]

3.6 Zapojení součástek

V následujících fotografiích je zobrazeno zapojení jednotlivých součástek projektu. ZDE PÁR FOTEK ZE ZAPOJENÍ A V PRILOZE BUDE SCHEMA Z KICADU



Obrázek 6: Schéma zapojení součástek z programu KiCAD [Zdroj: vlastní]

Díky tomuto zapojení lze efektivně ovládat a monitorovat pohyb vozítka, získávat data o okolním prostředí pomocí senzorů a vizuálně signalizovat různé události pomocí RGB LED. Tímto způsobem je zajištěna plná funkčnost a správné chování vozítka v různých provozních situacích.

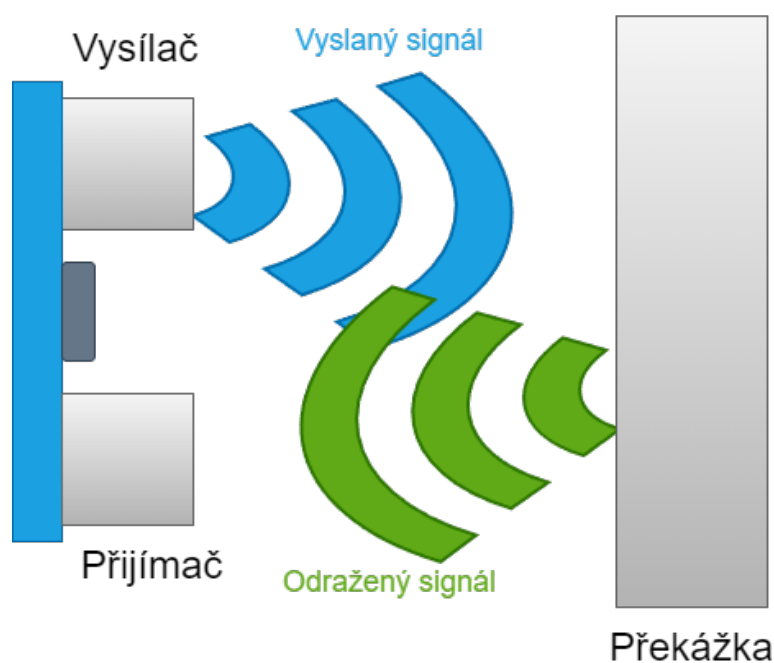
4 Ovládání sonaru

4.1 Měření vzdálenosti

„Sonar“ měří vzdálenost k objektu pomocí ultrazvukového senzoru (viz v kapitole použitých součástek). Senzor nejprve vysílá ultrazvukový impuls, který se odrazí od objektu a vrátí se zpět k senzoru. Čas, který impuls stráví cestou tam a zpět, se pak použije k výpočtu vzdálenosti k objektu pomocí vzorce:

$$\text{vzdálenost} = \frac{\text{čas} \times \text{rychlost zvuku}}{2}$$

Jelikož impuls letí k překážce, kde se odrazí a stejnou rychlostí letí zpět je nutné vzdálenost vydělit 2. Funkce, která zajišťuje správný výpočet, vrací právě tuto přepočítanou vzdálenost, která je dále používána při zpracování mapy.



Obrázek 7: Měření vzdálenosti [Zdroj: vlastní, vytvořeno pomocí draw.io]

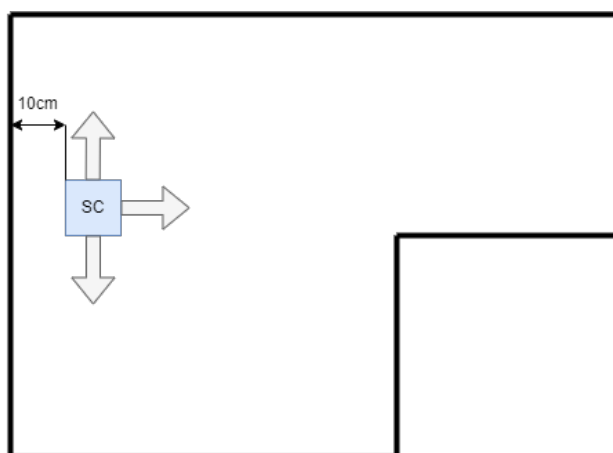
4.2 Otáčení sonaru

Pro správnou funkci mapování a detekci překážek je nezbytné, aby sonar pokrýval co největší plochu okolí vozítka. K tomu slouží otáčení sonaru, které je realizováno postupným otáčením krokového motoru o 90° . Tímto způsobem je zajištěno, že senzor bude schopen pokrýt všechny směry kolem vozítka a zachytit vzdálenost k překážkám ve všech směrech. Po každém otočení sonaru je provedeno měření vzdálenosti a aktualizace mapy místnosti podle získaných dat.

4.3 Taktika mapování

Taktika mapování místnosti je založena na pravidelném pohybu vozítka po místnosti s pevně definovanou taktikou. Tato taktika byla vybrána pro tento projekt kvůli její efektivitě a schopnosti pokrýt celý prostor místnosti. Vozítko je řízeno tak, aby se pohybovalo vždy o určitý krok, například 10 cm, od stěn místnosti, což odpovídá jedné souřadnici v mapě. Tímto způsobem je postupně zmapován celý prostor místnosti, zatímco senzor neustále snímá vzdálenost od překážek.

Na obrázku 8 je vidět Sonar Car v místnosti. Drží se 10 cm od stěny a šipkami jsou znázorněny možné pohyby vozítka.



Obrázek 8: Sonar Car v místnosti [Zdroj: vlastní, vytvořeno pomocí draw.io]

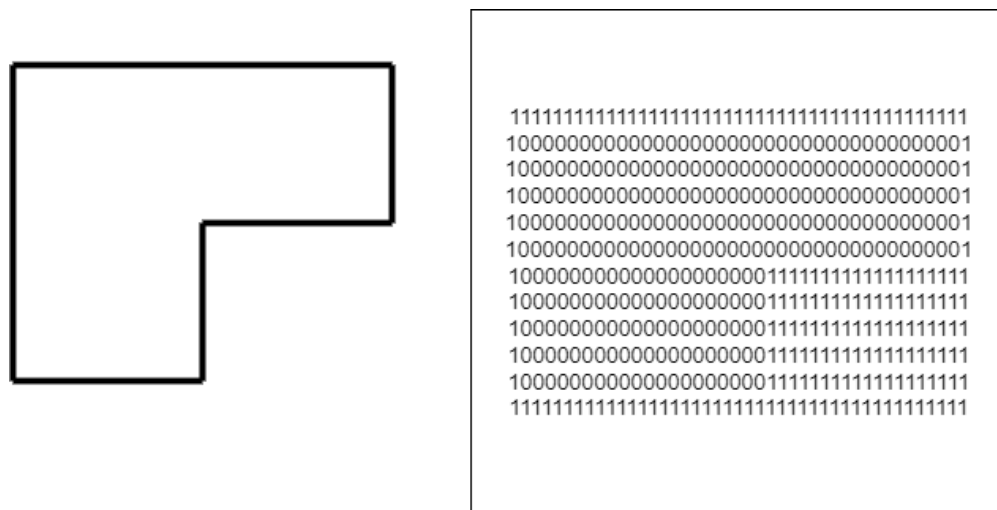
4.4 Vytváření mapy

Vytváření mapy místnosti je proces, při kterém jsou sbírána data z ultrazvukového senzoru a ukládána do dvourozměrného pole. Toto pole je klíčovou strukturou dat v tomto projektu, protože umožňuje efektivní ukládání a zpracování informací o prostředí. Každé snímání vzdálenosti od překážek je interpretováno jako bod v mapě, kde jednotlivé body jsou rozloženy v souřadnicovém systému. Na základě těchto bodů je poté vytvořena mapa místnosti, která poskytuje informace o umístění překážek a volných prostorů v místnosti.

Při vytváření mapy je nejprve vytvořeno prázdné pole o velikosti odpovídající velikosti místnosti. Poté je vozítko umístěno do středu tohoto pole a začne se otáčet a měřit vzdálenost od

Tento proces je opakován, dokud není mapa kompletní. Kompletnost mapy je určena na základě počtu prázdných bodů v poli. Pokud je počet prázdných bodů menší než určitý práh, je mapa považována za kompletní. Tento práh je stanoven tak, aby bylo zajištěno, že mapa je dostatečně detailní pro účely navigace vozítka.

Výsledná mapa je poté použita pro navigaci vozítka.



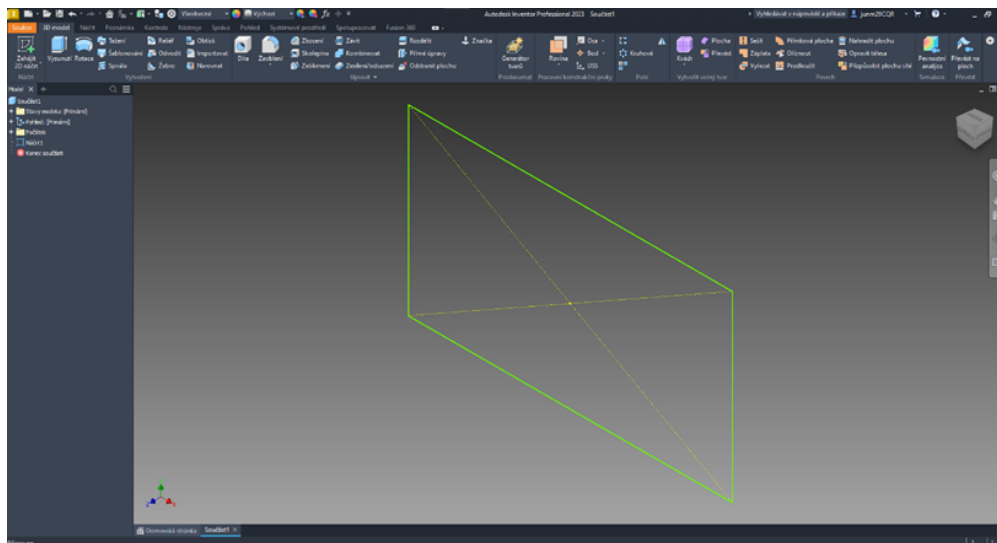
Obrázek 9: Příklad vizualizace místnosti v mapě [Zdroj: vlastní, vytvořeno pomocí draw.io]

5 Model sonaru

5.1 Modelovací prostředí

Pro vymodelování sonaru byl zvolen program Autodesk Inventor Professional 2023, protože poskytuje pokročilé nástroje pro modelování detailních a komplexních objektů. Díky jeho schopnostem bylo možné vytvořit realistický model krabičky pro sonar s důrazem na správné rozmístění komponent a estetický design.

V tomto projektu byl Autodesk Inventor Professional 2023 použit k vytvoření detailního modelu krabičky pro sonar.



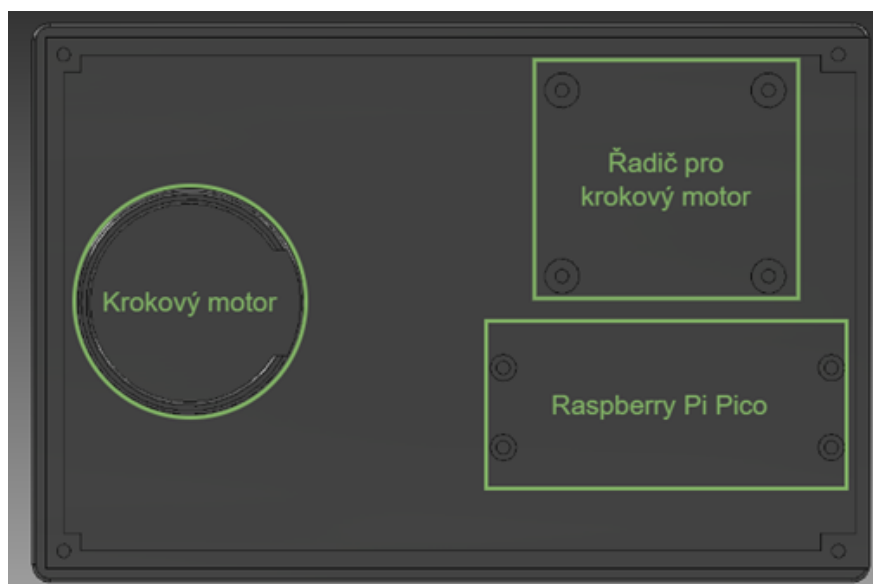
Obrázek 10: Modelovací prostředí programu Autodesk Inventor [Zdroj: vlastní]

5.2 Výběr rozmístění komponent

Prvním krokem bylo rozhodnuto o rozmístění jednotlivých komponent v krabičce. Důležité bylo zohlednit umístění ultrazvukového senzoru, Raspberry Pi Pico, krokového motoru a dalších součástí tak, aby všechny komponenty byly uspořádány efektivně a zároveň bylo zachováno dostatečné množství prostoru pro kabeláž a případné proudění vzduchu.

5.3 Návrh krabičky

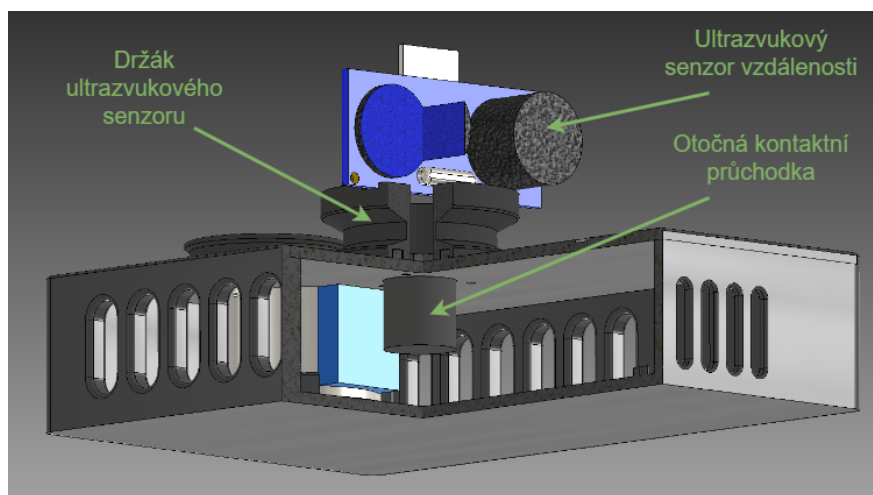
Po rozhodnutí o rozmístění komponent je vytvořen návrh krabičky v softwaru Autodesk Inventor. Tento návrh zahrnuje vytvoření 3D modelu krabičky včetně detailních rozměrů a geometrie. Během návrhu je dbáno na to, aby krabička měla dostatečně robustní konstrukci a zároveň byla dostatečně kompaktní pro snadnou montáž do vozítka.



Obrázek 11: Rozložení komponent v krabici [Zdroj: vlastní, vytvořeno pomocí draw.io]

5.4 Modelování dalších komponent

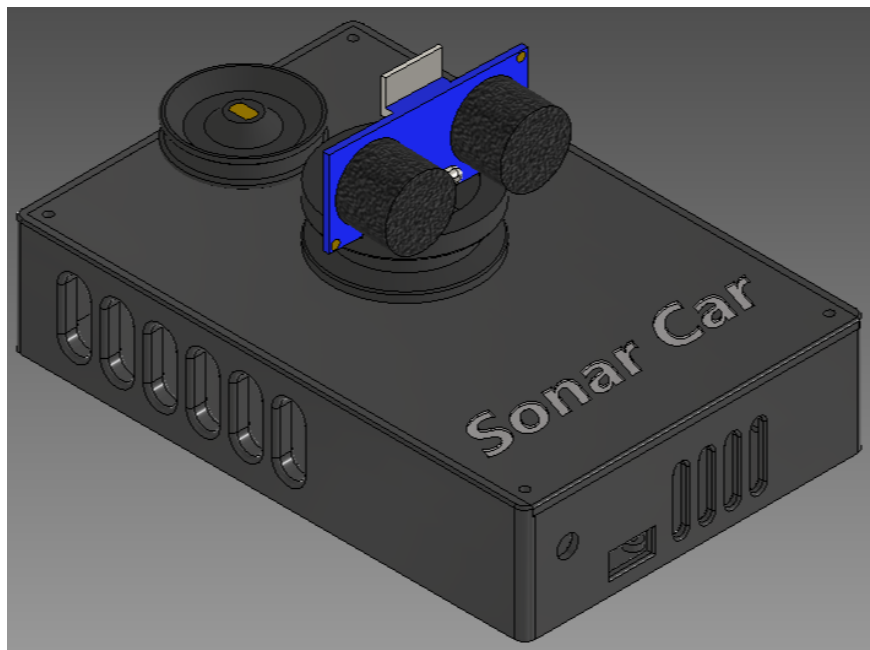
Po dokončení návrhu krabčky bylo přistoupeno k modelování dalších komponent, jako je například držák pro ultrazvukový senzor, montážní otvory otočnou kontaktní průchodku ve víku krabčky (viz obrázek 12), umístění pro Raspberry Pi Pico, krokový motor a další. Při modelování bylo dbáno na to, aby každá komponenta byla správně umístěna a připevněna v krabčce.



Obrázek 12: Pohled do vnitřku krabčky [Zdroj: vlastní, vytvořeno pomocí draw.io]

5.5 Finální úpravy a optimalizace

Po dokončení modelování byly provedeny finální úpravy a optimalizace, aby byla zajištěna správná funkce krabičky a maximální využití prostoru. To zahrnovalo například zaoblení hran a rohů pro snížení rizika zranění, přidání otvorů pro větrání a montážních bodů pro snadnou instalaci do vozítka. Dále byly provedeny úpravy pro zajištění správného umístění a uchycení jednotlivých komponent, jako je ultrazvukový senzor a krokový motor.



Obrázek 13: Model kompletního boxu s některými z komponent [Zdroj: vlastní]

5.6 Export do formátu pro 3D tisk

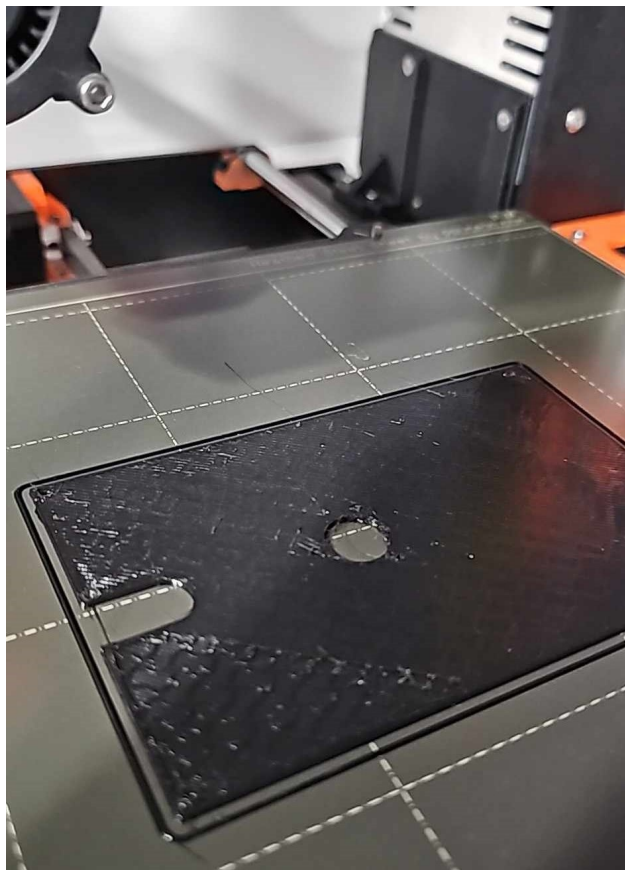
Nakonec je třeba digitální model exportovat do formátu .STL. Tento formát je standardem pro 3D tisk a je podporován většinou 3D tiskáren. Po exportu modelu do formátu .STL je potřeba model připravit pro tisk pomocí slicovacího programu. V tomto projektu byl použit program PrusaSlicer, který převede model do .gcode pro 3D tiskárnu. PrusaSlicer nabízí mnoho možností pro nastavení parametrů tisku, což umožňuje optimalizovat tisk pro konkrétní model a materiál.

5.7 3D tisk

3D tisk je moderní technologie, která umožňuje vytvářet fyzické objekty na základě digitálních modelů. Tento proces spočívá v postupném vrstvení materiálu, čímž se vytváří třírozměrný objekt. 3D tisk se používá v mnoha oblastech, jako jsou průmysl, medicína, architektura, design a mnoho dalších. Díky této technologii lze vytvářet složité geometrické tvary a prototypy, což umožňuje

rychlejší a efektivnější vývoj nových produktů. 3D tisk se stává stále populárnější a dostupnější pro širokou veřejnost, což umožňuje využití této technologie i pro domácí potřeby a hobby projekty.

Pro tisk krabičky na sonar a jeho součástí byli použity tiskárny Original Prusa i3 MK3 a Original Prusa i3 MK3S. Tyto tiskárny byly vybrány pro tento projekt kvůli jejich vysoké přesnosti a spolehlivosti. Byl zvolen filament PLA, který je ideální pro tento projekt kvůli jeho snadné tisknutelnosti, možnosti úprav i po vytištění komponenty. Všechny součástky byly tisknuty s vrstvou 0.2 mm a výplní 15 %, což bylo stanoveno na základě požadavků na pevnost a přesnost komponent.



Obrázek 14: Ukázka tisku víka pro krabičku [Zdroj: vlastní]

6 Závěr

Pokročilá verze tohoto projektu, známá jako LiDAR (Light Detection And Ranging), otevírá široké možnosti využití. LiDAR lze použít nejen k mapování místností, ale také k měření vzdálenosti, mapování terénu, monitorování atmosférických jevů a mnoha dalším aplikacím. Výsledky tohoto projektu poskytují solidní základ pro další vývoj a vylepšení, což může vést k vytvoření pokročilejších a přesnějších systémů mapování. Tento projekt ukazuje, jak jednoduché vozítko s jednoduchými součástkami a programováním může mít široké uplatnění a přinášet užitek v mnoha odvětvích.

Díky své flexibilitě a snadnému použití může tento projekt sloužit jako inspirace pro budoucí inovace v oblasti autonomních robotických systémů. Rozvoj a zdokonalení tohoto konceptu mohou přinést revoluci v průmyslu, výzkumu a každodenním životě. Jedním z možných budoucích směrů pro tento projekt je integrace pokročilejších senzorů pro zlepšení přesnosti mapování. Další možností je vývoj sofistikovanějších algoritmů pro navigaci, které by mohly zlepšit schopnost vozidla efektivně se pohybovat v různých prostředích. Kromě toho by mohlo být užitečné zkoumat možnosti zvýšení odolnosti vozidla, například prostřednictvím vývoje robustnějšího podvozku nebo použití materiálů odolných vůči různým podmínkám.

Tento projekt má potenciál přinést reálný přínos pro průmysl a společnost jako celek. Autonomní robotické systémy, jako je ten, který je předmětem této práce, mohou značně zlepšit efektivitu a produktivitu v mnoha průmyslových sektorech. Například v logistice by autonomní vozidla mohla zlepšit efektivitu skladování a distribuce. V oblasti zdravotnictví by mohla autonomní vozidla pomoci s monitorováním pacientů nebo dodávkou léků. Navíc by tato technologie mohla zlepšit kvalitu života lidí tím, že by jim poskytla nástroje pro lepší navigaci a orientaci v prostoru.

Nicméně, i přes své úspěchy a potenciál, je třeba si uvědomit některé výzvy, jako je například zvýšení přesnosti mapování, optimalizace algoritmů pro navigaci a zvýšení odolnosti vozítka v různých prostředích. Tyto výzvy představují zajímavé výzkumné a vývojové možnosti pro budoucí práce v oblasti autonomní robotiky.

V závěru lze konstatovat, že projekt Sonar Car může představovat pouze začátek dlouhé cesty

směrem k vytvoření efektivních a inteligentních autonomních systémů, které mohou přinést reálný přínos jak pro průmysl, tak pro společnost jako celek.

Zdroje

- Arduino HC-SR04 Ultrasonic Sensor Tutorial* (2024). URL: <https://lastminuteengineers.com/arduino-sr04-ultrasonic-sensor-tutorial/> (cit. 10.02.2024).
- LiDAR* (2024). URL: <https://cs.wikipedia.org/wiki/Lidar> (cit. 10.02.2024).
- OpenAI (2024). *OpenAI ChatGPT*. URL: <https://chat.openai.com/> (cit. 10.02.2024).
- Průša, Josef (2019). *Základy 3D tisku*. Praha, Czech Republic: Grada Publishing.
- Radic ULN2003 pro krokový motor* (2024). URL: <https://www.laskakit.cz/radic-uln2003-pro-krokovy-motor/> (cit. 10.02.2024).
- Raspberry Pi Documentation* (2024). URL: <https://www.raspberrypi.com/documentation/> (cit. 10.02.2024).
- Raspberry Pi Pico* (2024a). URL: <https://www.laskakit.cz/raspberry-pi-pico/> (cit. 10.02.2024).
- Raspberry Pi Pico* (2024b). URL: <https://www.raspberrypi.com/products/raspberry-pi-pico/> (cit. 10.02.2024).
- Ultrazvukový měřič vzdálenosti - návod k použití* (2024). URL: <https://dratek.cz/docs/produkty/1/1298/1434544602.pdf> (cit. 10.02.2024).
- Ultrazvukový měřič vzdálenosti HC-SR04* (2024). URL: <https://www.laskakit.cz/arduino-ultrazvukovy-meric-vzdalenosti-hc-sr04/> (cit. 10.02.2024).

Seznam obrázků

1	Raspberry Pi Pico W [Zdroj: https://www.amazon.ca/Waveshare-Microcontroller-Compatible-Raspberry-Pico/dp/B0BM3LCC7D]	8
2	HC-SR04 [Zdroj: https://cz.rs-online.com/web/p/doplňky-bbc-micro-bit/2153181]	8
3	Krokový motor 28BYJ-48 [Zdroj: https://www.laskakit.cz/krokovy-motor-28byj-48/]	9
4	Řadič ULN2003 [Zdroj: https://www.laskakit.cz/radic-uln2003-pro-krokovy-motor/]	9
5	RGB LED [Zdroj: https://dratek.cz/docs/produkty/1/1298/1434544602.pdf]	10
6	Schéma zapojení součástek z programu KiCAD [Zdroj: vlastní]	10
7	Měření vzdálenosti [Zdroj: vlastní, vytvořeno pomocí draw.io]	11
8	Sonar Car v místnosti [Zdroj: vlastní, vytvořeno pomocí draw.io]	12
9	Příklad vizualizace místnosti v mapě [Zdroj: vlastní, vytvořeno pomocí draw.io]	13
10	Modelovací prostředí programu Autodesk Inventor [Zdroj: vlastní]	14
11	Rozložení komponent v krabici [Zdroj: vlastní, vytvořeno pomocí draw.io]	15
12	Pohled do vnitřku krabice [Zdroj: vlastní, vytvořeno pomocí draw.io]	15
13	Model kompletního boxu s některými z komponent [Zdroj: vlastní]	16
14	Ukázka tisku víka pro krabici [Zdroj: vlastní]	17

Seznam příloh

- Příloha I: Výkres boxu
- Příloha II: Výkres víka boxu
- Příloha III: Výkres držáku senzoru
- Příloha IV: Výkres kroužku
- Příloha V: Schéma zapojení součástek
- Příloha VI: Veškeré kódy a software (elektronicky)