

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

MICRO MOUSE

ZADANIE

**2024 Bc. Eduard Zelenay Bc. Erik Móčik Bc. Filip Lobpreis Bc.
Jakub Močarník Bc. Maroš Kocúr Bc. Matúš Machata Bc.
Timotej Polc**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

MICRO MOUSE

ZADANIE

Študijný program: Robotika a kybernetika
Predmet: I-TP – Tímový projekt
Prednášajúci: doc. Ing. Andrej Babinec, PhD.
Vedúci pámu: Bc. Maroš Kocúr

**Bratislava 2024 Bc. Eduard Zelenay Bc. Erik Móčik Bc. Filip
Lobpreis Bc. Jakub Močarník Bc. Maroš Kocúr Bc. Matúš
Machata Bc. Timotej Polc**

Podakovanie

Chceme sa podakovať doc. Ing. Andrejovi Babincovi PhD., za vybavenie financí od Fakulty elektrotechniky a informatiky STU, na motory a senzory.

Zoznam obrázkov a tabuliek

Obrázok 1.1	UTechnický nákres prevodu motora	2
Obrázok 1.2	Koleso s prevodovkou	3
Obrázok 1.3	DC Mikromotor Faulhaber 1717T003SR	4
Obrázok 1.4	Enkóder IEH2-4096	5
Obrázok 1.5	Gyroskop MPU9250	6
Obrázok 1.6	Vzdialenosný senzor VL53L1X	7
Obrázok 1.7	Návrh PCB dosky použitej vo finálnej verzii.	9
Obrázok 1.8	Konečná verzia návrhu všetkých komponentov vyrábaných pomocou 3D tlače usporiadaných na PCB.	11
Obrázok 2.1	Základný princíp ToF	13
Obrázok 2.2	Gyroskop s akcelerometrom MPU9250	13
Obrázok 2.3	UML diagram logiky algoritmu	18
Obrázok 2.4	Používateľské rozhranie HMI aplikácie	20
Obrázok 3.1	MikroMyš tímu	21

Úvod

Práca spočíva v navrhnutí, zstrojení a programovaní mobilného robota, ktorý dokáže autonómne prechádzať bludiskom. Študenti musia dodržať maximálne povolené rozmery pre súťaž a vybrať vhodné typy senzorov, ktoré umožnia vypočítať polohu na navigáciu robota v bludisku.

Úlohy:

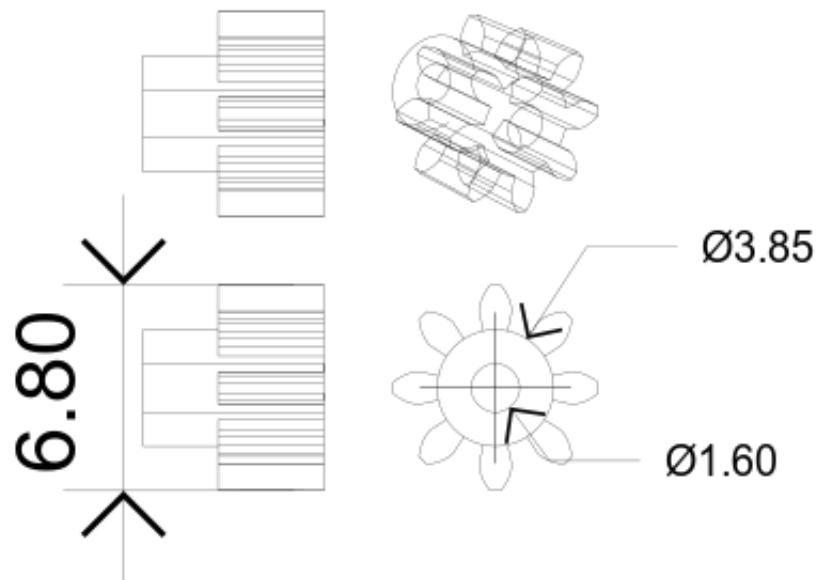
1. Navrhnutie a výroba dizajnu robota.
2. Výber vhodných motorov a snímačov (napr. ToF (Time of Flight), motorový ovládač, gyroskop).
3. Programovanie riadenia robota a implementácia spôsobu lokalizácie v bludisku.
4. Otestovanie funkčnosti zvoleného riešenia

1 Hardware

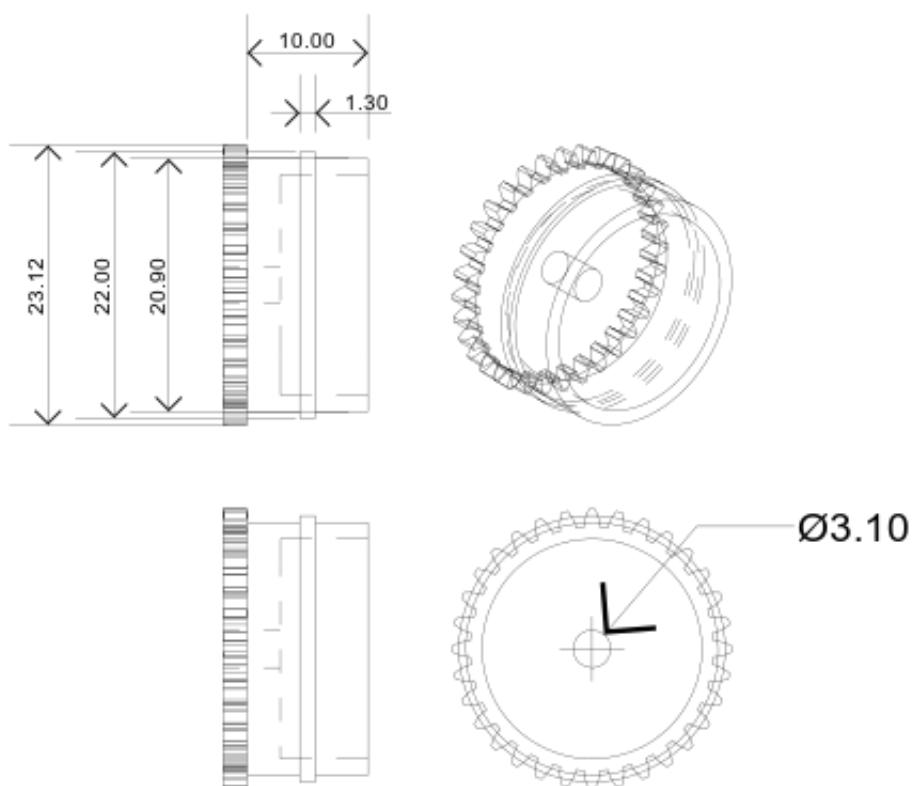
Z hárverovej stránky sme potrebovali v tomto projekte vytvoriť dosku plošných spojov (PCB - Printed Circuit Boards) a návrh držiakov motorov, gyroskopu a kolies, ktorých výroba prebiehala pomocou 3D tlače.

1.1 Model

Kolesá a ozubené koliesko pre hriadeľ motora boli vytlačené v 3D tlačiarni s resinovou náplňou FabPro 1000 Resin Cartridge Tough BLK. Na obrázkoch 1.1 a 1.2 môžeme vidieť návrhy ozubeného kolieska pre hriadeľ a kolesa.



Obr. 1.1: UTechnický nákres prevodu motora



Obr. 1.2: Koleso s prevodovkou

1.2 Motory

Použité motory boli vybrané s ohľadom na ich výkonnosť a spoľahlivosť. Sú to jednosmerné motory, vidieť ich môžeme na 1.3 a detaily o nich uvádzame nižšie:



Obr. 1.3: DC Mikromotor Faulhaber 1717T003SR

- Model: Faulhaber 1717 T
- Typ: 003 SR
- Max. točivý moment: 5.35 mNm
- Max. rýchlosť: 11000 rpm
- Odkaz na technický list: Technický list motora

1.3 Enkódery

Na monitorovanie polohy a otáčok motorov sme použili magnetické enkódery. Informáciu o rotácií získavajú ako zmenu magnetického poľa. Enkóder obsahuje dva Hallove články, ktoré merajú veľkosť magnetickej indukcie ako Hallovo napätie ako:

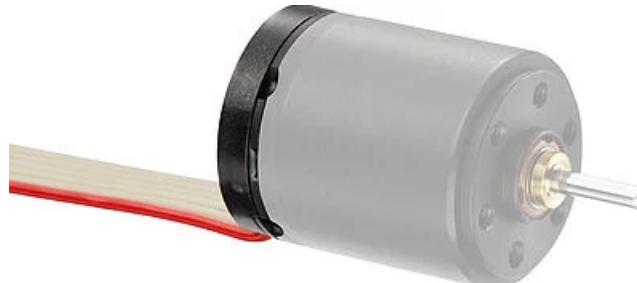
$$U_H = h_H \frac{BI_p}{h} \quad (1.1)$$

kde U_H je Hallovo napätie, h_H je Hallova konštantá, B je magnetická indukcia, I_p je pomocný prúd, h je hrúbka polovodičovej doštičky. Tieto Hallove články merajú magnetickú indukciu v dvoch osiach na seba kolmých.

Enkóder obsahuje permanentný magnet pripojený na jeho hriadeľ. Rotáciou hriadeľa sa točí magnet, a teda aj smer magnetického poľa. Zmeny v oboch osiach sú prekonvertované cez AD prevodník na digitálne signály. Skrz trigonometrické funkcie je následne

vypočítaný fázový posun medzi týmito signálmi. To je dôležité nielen kvôli výpočtu rotácie hriadeľa, ale aj pre zistenie smeru rotácie.

Nižšie sú uvedené informácie o použitých enkóderoch 1.4:



Obr. 1.4: Enkóder IEH2-4096

- Model: IEH2-4096
- Rozlíšenie: 4096
- Kanály: 2
- Odkaz na technický list: Technický list enkódera

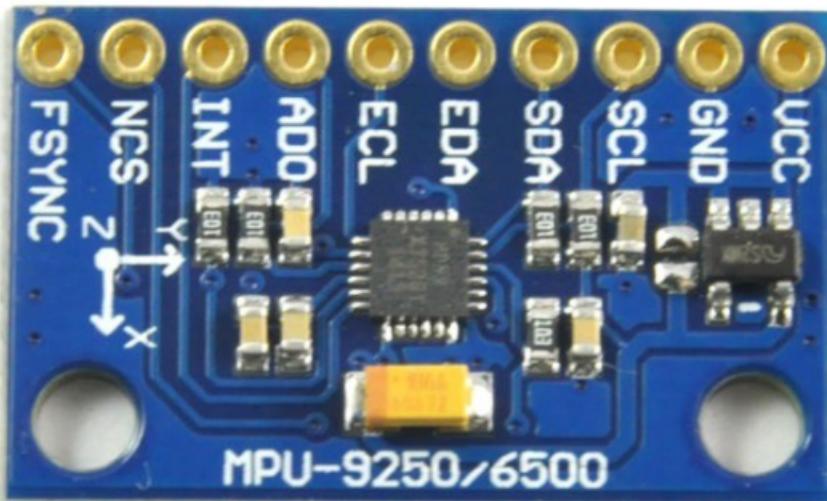
1.4 Gyroskop

Na sledovanie orientácie a pohybu robota sme si zvolili MEMS gyroskop. Tie merajú uhlovú rýchlosť prostredníctvom Coriolisovho efektu. Coriolisova sila je spôsobená inerciou objektu v priamom pohybe, kolmo na smer pohybu objektu a jej výpočet je daný:

$$\vec{F} = -2m\vec{\Omega}_z \vec{x}\vec{v} \quad (1.2)$$

MEMS Gyroskop obsahuje vybujúce elementy na pružine. Ked' sa gyroskop pohybuje kapacitný senzor deteguje vibrácie, tie sa následne premenia na uhlové zrýchlenie.

Ďalej sú uvedené podrobnosti o použitom module 1.5:



Obr. 1.5: Gyroskop MPU9250

- Model: MPU9250
- Typ: Gyroskop, akcelerometer a magnetometer
- Komunikácia: I2C, SPI
- Odkaz na technický list: MPU9250 Technical Datasheet

1.5 Laserový diaľkomer

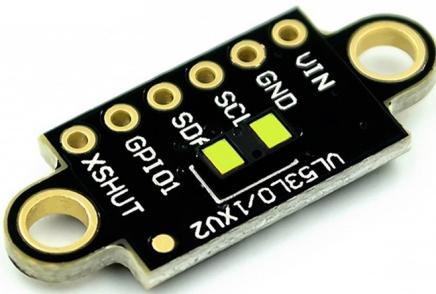
Ako vzdialenosný senzor sme zvolili Time-of-Flight (doba trvania letu) laserový merací modul poskytujúci presné meranie vzdialnosti bez ohľadu na farbu alebo odrazivost ciela. Time-of-Flight senzory fungujú na báze merania času letu laserového lúča vyslaného zo senzora, k objektu a naspäť. Vzdialenosť je z nameraného času vypočítaná podľa vzorca na 1.3:

$$d = \frac{c \cdot t}{2} \quad (1.3)$$

kde d je vypočítaná vzdialenosť, c je rýchlosť svetla a t je doba letu lúča.

Modul môžeme vidieť na obrázku 1.6. Nižšie sú uvedené parametre senzora.

- Model: VL53L1X



Obr. 1.6: Vzdialenosný senzor VL53L1X

- Maximálna vzdialenosť: 400cm
- Frekvencia merania: nastaviteľná až do 50Hz
- Zorné pole: 27°
- Komunikácia: I2C
- Odkaz na technický list: Technický list laserového diaľkomera

Na meranie vzdialenosť sme využili štyri takéto moduly pripojené na dosku vyosené relatívne od dopredného smeru myši o 45° a 10°.

1.6 PCB

1.6.1 Komponenty

- **Senzory ToF (VL53L1X)**: Pripojenie s ESP32 cez I2C rozhranie.
- **Gyroskop s akcelerometrom (MPU9250)**: Pripojenie s ESP32 cez I2C rozhranie.
- **Driver na motor (TB6612FNG)**: Umiestnený blízko motorov, posielá PWM signály motorom.
- **MCU (ESP32)**: Centrálny riadiaci prvok PCB.
- **Step-up menič na 5V**: Zabezpečenie stabilných 5V pre logické obvody a enkodéry.
- **Logic shifter**: Prekladanie signálov medzi 3,3V a 5V úrovňami.

- **Ochrana batérie:** Meranie napäťia batérie a prúdové ochrany.
- **Batéria (Li-ion 1000mAh):** Umiestnená na PCB s prístupnými napájacími konektormi a ochrannými obvodmi.

1.6.2 Proces tvorby PCB

- **Návrh schémy**

Vytvorili sme detailný návrh elektrickej schémy s prepojeniami všetkých komponentov, pričom sme zabezpečili správne napäťové úrovne a prepojenia medzi jednotlivými časťami.

- **Rozmiestnenie komponentov**

Strategicky sme umiestnili všetky komponenty na PCB tak, aby boli minimalizované rušenia a zohľadnené fyzické rozmeria a montážne obmedzenia.

- **Vedenie ciest**

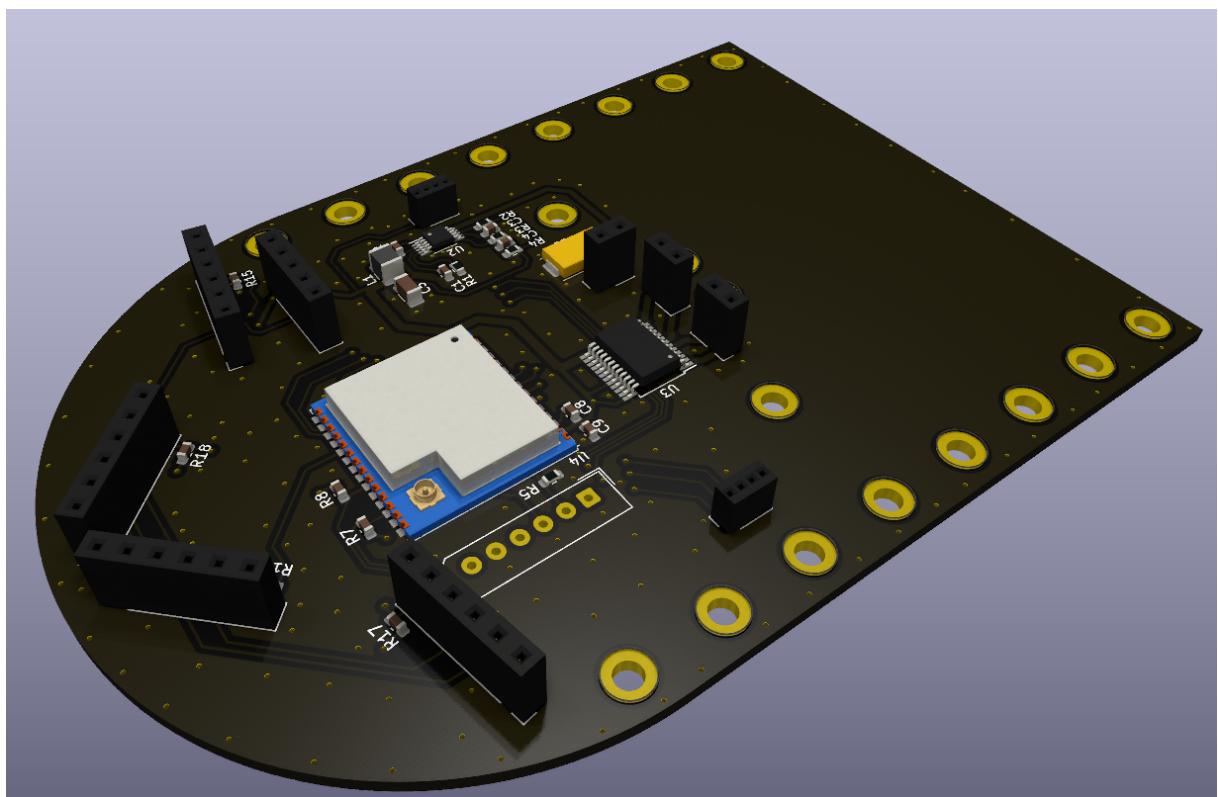
Navrhli sme cesty (trasy) pre prepojenie komponentov, s dôrazom na minimalizáciu kríženia ciest a optimalizáciu pre rušenie (napr. použitie samostatných zemných a napájacích rovín).

- **Napájanie a ochrany**

Implementovali sme obvody na stabilizáciu napäťia (3,3V a 5V), ochrany batérie, a zabezpečili, aby boli všetky komponenty správne napájané a chránené.

Taktiež sme museli myslieť na nasledovné:

- Udržanie minimálnej dĺžky kritických signálových trás (I2C, PWM).
- Oddelenie silových a signálových ciest na PCB, aby sa minimalizovalo rušenie.
- Zabezpečenie dostatočnej kapacity a filtrácie na napájacích cestách.
- Implementácia ochranných obvodov pre batériu a citlivé komponenty.



Obr. 1.7: Návrh PCB dosky použitej vo finálnej verzii.

1.7 CAD

Pri návrhu ako aj pri postupnom iterovaní sme zohľadňovali mnohé aspekty spojené s procesom pri prototypizovaní. Mnohé verzie boli navrhnuté rovnako no v konečnom dôsledku nám ich proces tlače zmenil a museli sme sa s tým vysporiadať, pretože presnosť pri našom návrhu zohrávala klúčovú úlohu. Pri tlači sme používali dva typy 3D tlače, klasickú FDA no aj živicovú SLA tlač.

- **3D návrh kolies**

Kolesá pre Micro Mouse boli navrhnuté na základe existujúcich modelov, ktoré poskytli osvedčený základ pre dizajn. Použitie osvedčených modelov umožnilo urýchliť návrhový proces a zaručilo, že výsledné kolesá budú splňať požadované špecifikácie.

- **Výber modelu**

Pre návrh kolies sme vybrali modely, ktoré sú už overené v praxi. Tento prístup minimalizoval riziko návrhových chýb a zabezpečil, že kolesá budú mať vhodné mechanické vlastnosti a správne rozmeria.

- **Návrh gumy na koliesá**

Gumy na koliesá boli použité z existujúceho modelu, čo zabezpečilo optimálnu trakciu a odolnosť. Tento krok eliminoval potrebu vlastného návrhu gumy a umožnil nám sústrediť sa na integráciu kolies do celkového návrhu Micro Mouse.

- **3D návrh úchytov**

Pre návrh úchytov sme využili iteratívny proces, ktorý umožnil presné prispôsobenie vzdialenosťí kolies a ich optimálne umiestnenie na Micro Mouse.

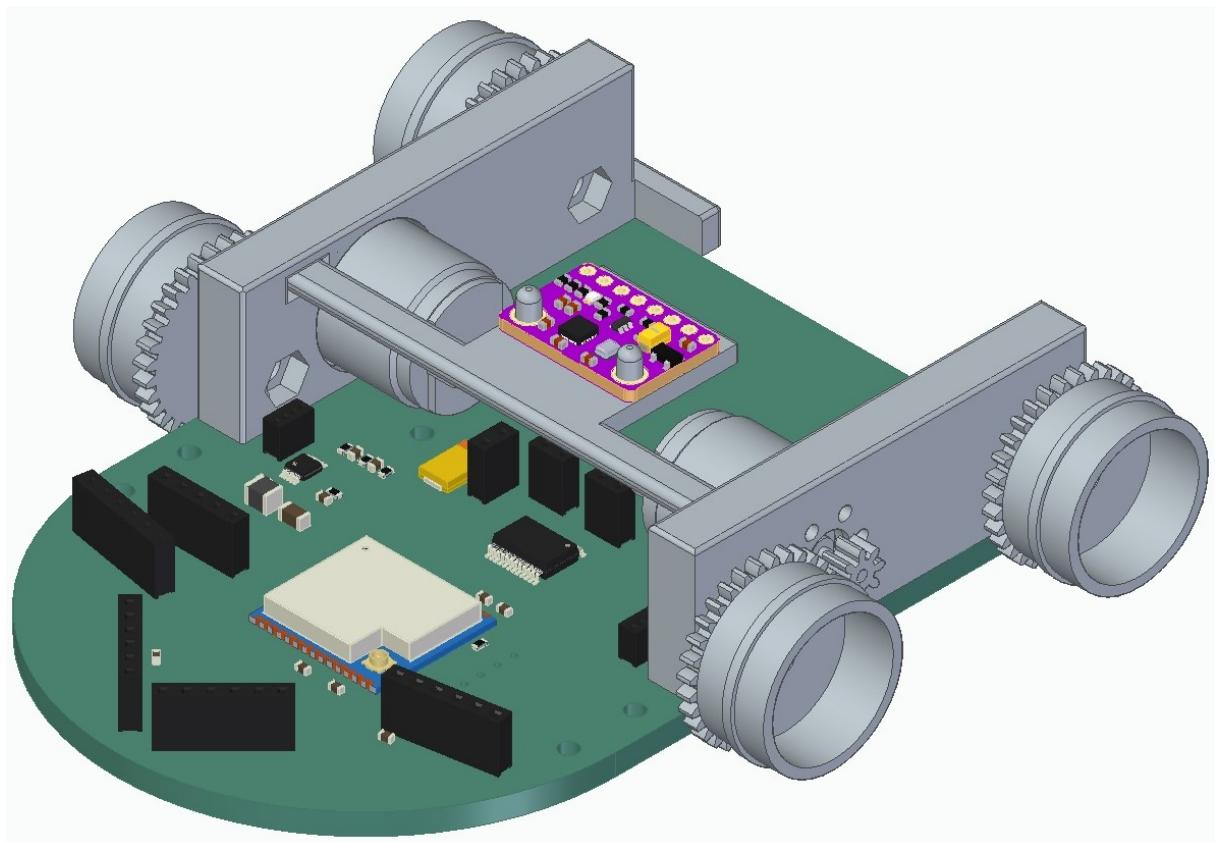
- **Iteratívny proces návrhu**

Návrh úchytov prebiehal v niekoľkých iteráciách, pričom sme upravovali vzdialenosťí kolies a ich polohu, aby sme dosiahli najlepšiu možnú stabilitu a pohyblivosť Micro Mouse. Tento proces zahŕňal testovanie rôznych konfigurácií a úpravou návrhu na základe výsledkov týchto testov.

- **Optimalizácia vzdialenosťí**

Upravovanie vzdialenosťí kolies bolo klúčové pre dosiahnutie optimálnej manévrovatelnosti. Iteratívne úpravy umožnili presné doladenie návrhu, čo viedlo k zlepšeniu celkového výkonu Micro Mouse.

Návrh a výroba 3D komponentov pre Micro Mouse zahŕňala použitie existujúcich modelov kolies a gumy, ako aj iteratívny návrh úchytov. Tento prístup umožnil efektívne vytvorenie funkčných a spoľahlivých komponentov, ktoré prispievajú k optimálnemu výkonnému Micro Mouse.



Obr. 1.8: Konečná verzia návrhu všetkých komponentov vyrábaných pomocou 3D tlače usporiadaných na PCB.

2 Software

Na písanie kódu sme použili prostredie **esp_idf** verzie 4.4.7, ktoré sme vytvorili za pomoci ich repozitára [1].

Postup na inicializáciu ESP IDF a následné spustenie projektu je nasledovný:

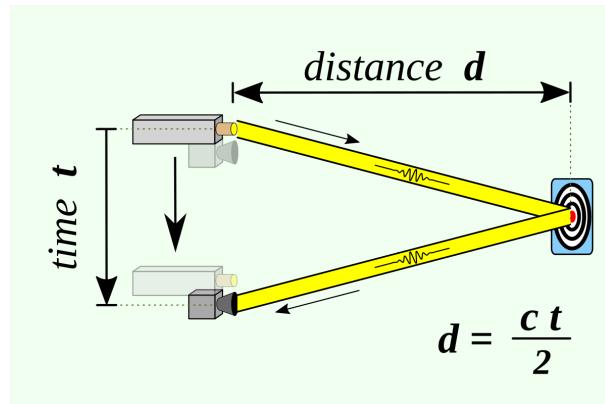
1. Stiahnutie ESP IDF z repozitára [1] (My sme použili verziu 4.4.7).
2. Spustenie príkazu `./install.sh esp32` pre Linux. Tento príkaz stiahne a nainštaluje všetky potrebne závislosti.
3. Exportovanie súboru `../export.sh` pre Linux. Tento príkaz nastaví všetky potrebne premenne prostredia.
4. Zmeniť aktuálny adresár na náš projekt.
5. Spustiť príkaz `idf.py build` pre kompliláciu projektu.
6. Spustiť príkaz `idf.py -P <PORT_NR> flash` pre nahratie binárneho súboru do ESP32.
Pre tuto operáciu je potrebne mať pripojený počítač a misku cez UART.
7. Spustiť príkaz `idf.py monitor` pre sledovanie výstupu z ESP32.

Alternatívne spustenie je možné pomocou platformy **Visual Studio Code** s nainštalovaným rozšírením **ESP-IDF** [2].

2.1 Senzory

2.1.1 VL53L1X

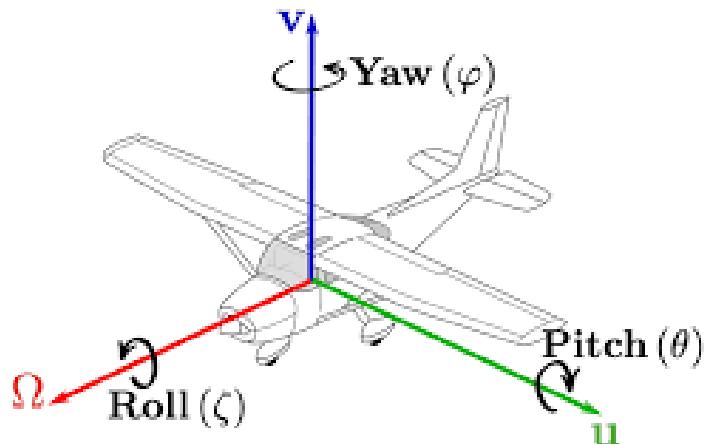
Pre snímanie vzdialenosť bola vytvorená knižnica na kalibráciu a vyčítavanie dát zo senzora VL53L1X. Komunikácia prebieha prostredníctvom komunikačného protokolu I^2C . Knižnica je nadstavbou nad voľne dostupným rozhraním od STMicroelectronics. Na obrázku 2.1 je zobrazený princíp fungovania senzora využívajúceho metódu Time-of-



Obr. 2.1: Základný princíp ToF

Flight (ToF) pre meranie vzdialenosť. Zariadenie emituje svetelný signál, ktorý sa odrazi od objektu a vráti späť k senzoru.

2.1.2 MPU9250



Obr. 2.2: Gyroskop s akcelerometrom MPU9250

V našom projekte sme použili senzor MPU9250, ktorý je trojosovým akcelerometrom a gyroskopom. Tento senzor nám umožňuje získať presné údaje o orientácii mikromyši

v trojrozmernom priestore. Klúčovou súčasťou spracovania dát z tohto senzoru je výpočet Eulerových uhlov – roll, pitch a yaw. Tieto údaje sú neoceniteľné pre stabilizáciu a presné navigačné manévre mikromyši, poskytujúc tak lepší prehľad o jej aktuálnom náklone a orientácii.

2.1.3 IEH2-4096

V rámci nášho projektu sme implementovali impulzové čítače s inkrementálnymi enkodérmi IEH2-4096, ktoré sú pripojené na motory. Tieto enkodéry generujú signál v podobe elektrických impulzov v závislosti od uhlového pohybu motorov. Pre naše účely sme nastavili systém tak, aby generoval prerušenie po každej 1/8 otáčke motora.

Každý "tick" nám indikuje 1/8 otáčky motora, a pomocou softvéru potom prepočítavame tieto hodnoty na požadované metrické jednotky (napríklad stupne alebo radiány). Tento mechanizmus nám umožňuje presne sledovať a regulovať pohyb mikromyši, čo je kritické pre presné navigačné a kontrolné úlohy v autonómnych aplikáciách.

2.2 Odometria

Pri počítaní polohy robota bola použitá odometria. Použili sme na to inkrementálny enkóder 1.3, ktorý je pripojený na motor 1.2. Z enkóderov si získame aktuálnu polohu robota cez základný vzťah:

$$\Delta x_t = x_{t-1} + \Delta s_t \cdot \cos(\phi_{t-1} + \Delta\phi_t) \quad (2.1)$$

$$\Delta y_t = y_{t-1} + \Delta s_t \cdot \sin(\phi_{t-1} + \Delta\phi_t) \quad (2.2)$$

, kde:

- Δx_t a Δy_t sú zmeny pozície robota v osiach x a y v čase t ,
- x_{t-1} a y_{t-1} sú predchádzajúce pozície robota v osiach x a y ,
- Δs_t je vzdialenosť, ktorú robot prešiel od posledného merania,
- ϕ_{t-1} je predchádzajúci uhol orientácie robota,
- $\Delta\phi_t$ je zmena uhla orientácie robota získaná z gyroskopu.

Pre lepšie pochopenie odometrie a jej presnosti je dôležité zvážiť aj potenciálne chyby z enkóderov a gyroskopu, ako aj spôsoby ich korekcie v softvérovom riešení robota.

2.3 PID regulátor motorov

Do motorov posielame ako akčný zásah signál PWM (Pulse width modulation). Tento signál môže nadobúdať hodnoty od 0 po 1023. Zároveň musíme myslieť aj na prepäťovú ochranu. Keď pošleme do motorov veľké zrýchlenie, tak sa stane, že sa program na čipe zresetuje.

Aby sme ošetrili tieto situácie, sme pridali rozbehový člen v podobe rampového signálu, ktorý zabezpečí postupné zmeny žiadanej hodnoty namiesto skokových zmien. Na riadenie motorov sme implementovali PID regulátor s Antiwindupom. Jeho obmedzenie je nastavené zdola na 0 a zhora na 1023. Toto obmedzenie zdola je nastavené na 0 kvôli implementácii nastavenia smeru otáčania motorov. V prostredí ESP musíme dopredu nastaviť smer otáčania motorov. Po jeho nastavení sa všetky hodnoty posielajú ako *unsigned*. Teda ak by sme mu poslali zápornú hodnotu, tak by sa motor rozbehol nastaveným smerom. Antiwindup zapojenie maximálnu rýchlosť obmedzuje na 1023.

Pre jednoduchosť ovládania sme nastavili regulátor tak, aby vstupná žiadana hodnota bol v milimetroch za sekundu. Regulátor si túto hodnotu prepočíta na PWM signál, ktorý pošle motorom. Spätné väzby sa získavajú z enkóderov a dopočítavajú sa na milimetre za sekundu. Tieto hodnoty sa taktiež prepočítajú na PWM signál. Prepočet z milimetrov za sekundu na PWM signál je závislý uskutočňovaný nasledujúcim výpočtom:

$$mmps = d\pi \frac{IMP_t}{IMP_{pr}i\Delta t} \quad (2.3)$$

Rovnica 2.3 obsahuje parametre:

- d - Priemer kolesa,
- IMP_t - Aktuálne impulzy z enkóderu,
- IMP_{pr} - Impulzy z enkóderu za jednu otáčku (per rotation),
- i - Pomer prevodovky motora. Na našich motoroch je to prevod 32:8,
- Δt - Časový interval od posledného merania

$$PWM = \frac{mmps}{MWSIMP_{pr}} \quad (2.4)$$

Rovnica 2.4 obsahuje parametre:

- mmps - Rýchlosť robota v milimetroch za sekundu vypočítanej v rovnici 2.3,

- MWS - Maximálna rýchlosť otacania kolies,
- IMP_{pr} - Impulzy z enkóderu za jednu otáčku (per rotation).

2.4 Riadiaci algoritmus



Obr. 2.3: UML diagram logiky algoritmu

Na obrázku 2.3 je znázornený tokový diagram, ktorý predstavuje algoritmus na rozhodovanie o pohybe v priestore. Tento diagram obsahuje niekoľko krokov a rozhodnutí:

1. **Štart** - Začiatočný bod algoritmu.
2. **Slepá ulica?** - Rozhodovací krok, kde sa algoritmus pytá, či sa pred myškou nachádza slepá ulica.
 - Ak **áno**, nasleduje krok Otoč sa, čo znamená zmenu smeru o 180 stupňov.
 - Ak **nie**, pokračuje ďalším rozhodnutím.
3. **Je ľavá stena bližšie ako pravá?** - Ďalšie rozhodovacie kritérium zvažujúce relatívne pozície stien.
 - Ak **áno**, nasleduje krok Zaboč doprava.
 - Ak **nie**, presmeruje sa na ďalší rozhodovací bod.
4. **Sú rovnako vzdialé?** - Rozhoduje, či sú obidve steny rovnako vzdialené od myšky.
 - Ak **áno**, nasleduje krok Zaboč doľava.
 - Ak **nie**, vykoná sa krok Udržuj smer, čo znamená pokračovanie v súčasnom smere bez zmeny.

2.5 Upresnenie implementacie

Myška funguje na princípe reaktívnej navigácie. V prístupe kombinujeme 2 metódy. Jedna je *Braitenbergov pohon typu fear* a *PID regulator*. Naše zariadenie používa 4 ToF snímače:

- ToF1 - Snímač na ľavej strane snímajúci smer vpred,
- ToF2 - Snímač na ľavej strane snímajúci smer vľavo pod uhlom 45 stupňov, tak aby snímali ďalšiu bunku,
- ToF3 - Snímač na pravej strane snímajúci smer vpravo pod uhlom 45 stupňov, tak aby snímali ďalšiu bunku,
- ToF4 - Snímač na pravej strane snímajúci smer vpred

Kedže bludisko je spleť uličiek, o ktorých nevieme, kadiaľ vedú, snímače nám môžu vracať rôzne hodnoty. Preto, aby sme urobili nepredvídateľné prostredie predvídateľným, obmedzili sme vzdialenosť snímačov 1 a 4 na 20 centimetrov. Toto obmedzenie sme stanovili na základe experimentálnych meraní v bludisku, kde šírka uličiek bola 22,5 centimeta.

Algoritmus funguje nasledovne:

1. Zistenie rozdielu ToF1 a ToF4.
2. Dopočítanie akčného zásahu na z PID zo zisteného rozdielu.
3. Točiť sa na základe znamienka akčného zásahu.
4. Otáčanie sa ovplyvňuje na základe vzdialenosť ToF2 a ToF3.

Ak myška dôjde do slepej uličky, taktiež sme použili všetky 4 snímače nato aby sme detegovali do ktorej strany máme viac miesta a implementovali sme otocenie na mieste o 90 stupňov pomocou údajov z gyroskopu. Pre tento typ pohybu sme implementovali P regulátor rýchlosť kolies.

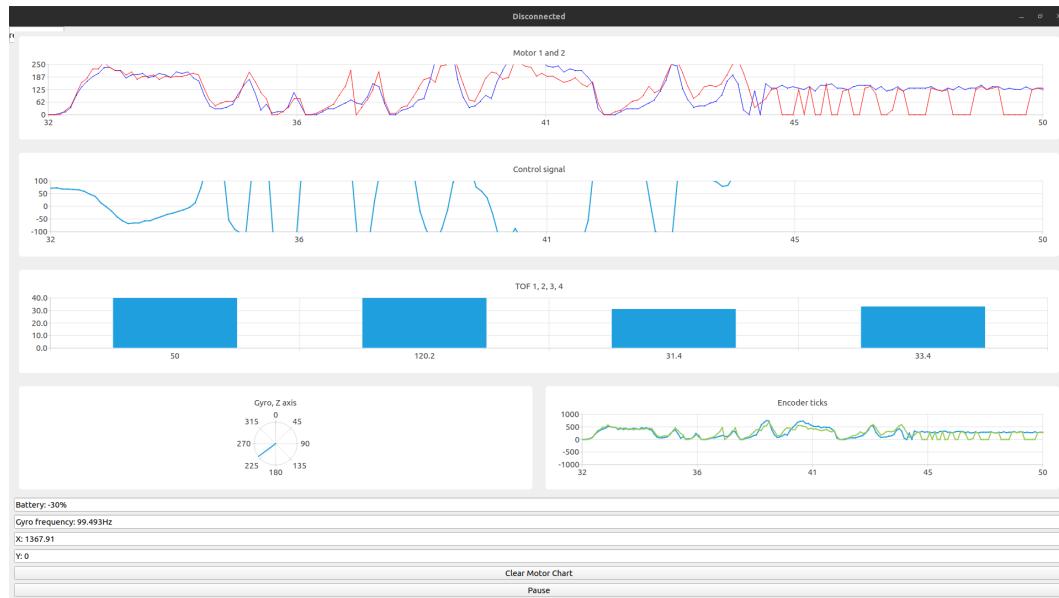
2.6 UDP Komunikácia

2.6.1 Client

Klientom v našom prípade je ESP32, ktorý sa používa v PCB mikromyši. Toto zariadenie je zodpovedné za zber údajov o stave a správaní mikromyši a ich odosielanie na server. Komunikácia s serverom prebieha prostredníctvom UDP protokolu, čo umožňuje rýchly a efektívny prenos dát.

2.6.2 Server

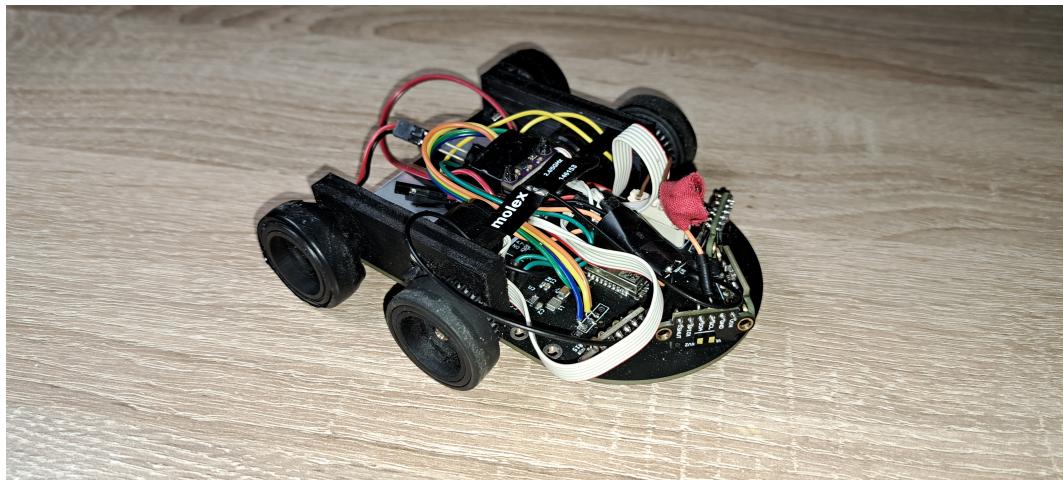
Na strane servera sme použili vývojové prostredie Qt Creator s programovacím jazykom C++ na implementáciu serverovej časti, ktorá prijíma dátu od klienta. Server je integrovaný s mini HMI (Human-Machine Interface) aplikáciou, ktorá umožňuje real-time vizualizáciu dát prostredníctvom grafov a poskytuje užívateľské rozhranie pre sledovanie a ladenie správania mikromyši.



Obr. 2.4: Používateľské rozhranie HMI aplikácie

HMI aplikácia je kľúčovým nástrojom pre efektívne diagnostikovanie a optimalizáciu výkonu autonómnej mikromyši. 2.4

3 Vizuál MikroMyšky



Obr. 3.1: MikroMyš tímu

Záver

Na tomto zadaní sme sa naučili komunikovať a riešiť problémy ako tím.

Na záver môžeme konštatovať, že úlohy projektu MicroMouse boli úspešne splnené. Plne sme využili potenciál celého tímu a maximálne sme sa snažili využiť všetky nami zadefinované senzory. Optimalizáciou softvéru sme dosiahli, že robot dokázal prechádzať úzkymi uličkami bez nárazov, čo výrazne prispelo k jeho celkovej efektivite a presnosti.

Počas projektu sme však narazili na problémy s pohonom, konkrétnie so zablokováváním prevodovky, čo spôsobovalo nepresnosť v odometrii. Tieto problémy nám znemožnili využitie odometrie na lokalizáciu robota.

V druhej verzii projektu sa zameriame na odstránenie týchto nedostatkov, aby sme mohli plne využiť odometriu na presnú lokalizáciu robota. A taktiež by sme chceli upraviť hardware, keďže máme k dispozícii presne rozmery senzorov a motorov a odladili sme chyby týkajúce sa hardware na prvej verzii, čo by mohlo pomôcť robotu pohybovať sa plynulejšie a pomôže to taktiež s váhou robota. Na nasledujúce hardwarove úpravy by sme vyladili regulátory a robot sa bude správať rovnako ako v prvej verzii.

Veríme, že tieto skúsenosti a získané poznatky nám umožnia pokračovať v ďalšom vývoji a dosiahnuť ešte lepšie výsledky v budúcich verziách MicroMouse.

Zoznam použitej literatúry

1. *ESP-IDF GitHub Repository* [online]. [cit. 2024-05-25]. Dostupné z : %7Bhttps://github.com/espressif/esp-idf/tree/v4.4.7%7D.
2. *ESP-IDF Visual Studio Code Extension* [online]. [cit. 2024-05-25]. Dostupné z : %7Bhttps://marketplace.visualstudio.com/items?itemName=espressif.esp-idf-extension%7D.