



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

## ROČNÍKOVÁ PRÁCE S OBHAJOBOU

**Téma:** Navigační systém  
pomocí UWB

**Autor práce:** František Slepíčka

**Třída:** 3.M

**Vedoucí práce:** Jiří Švihla

**Dne:** 30.4.2025

**Hodnocení:**

## Anotace

Tato práce se zabývá návrhem a realizací hybridního lokalizačního systému, který kombinuje dvě technologie – Ultra-Wideband (UWB) pro přesné měření vzdálenosti a Wi-Fi fingerprinting pro orientační určení polohy. Při známém pohybu jednoho zařízení v prostoru je systém schopen určit polohu druhého zařízení. Ovládání je možné prostřednictvím webového rozhraní nebo sériové komunikace. Výsledný systém je navržen pro snadnou integraci do dalších projektů, a to buď jako hlavní řídicí jednotka, nebo jako rozšiřující lokalizační modul.

Můj kolega se podílel na projektu zejména v těchto oblastech: vytvořil webové stránky pro ovládání systému, navrhl design vozidla a ochranných krabiček pro elektroniku, a fyzicky osadil a pájel desky plošných spojů.

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

„Prohlašuji, že jsem nástroje UI využil v souladu s principy akademické integrity a že na využití těchto nástrojů v práci vhodným způsobem odkazuji.“

„Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.“

V Plzni dne:

Podpis:

# Obsah

<b>Úvod</b>	<b>5</b>
<b>1 Teoretický základ lokalizace</b>	<b>6</b>
1.1 Přehled metod lokalizace a jejich srovnání . . . . .	6
1.2 Existující lokalizační systémy . . . . .	7
<b>2 Návrh hybridního systému</b>	<b>9</b>
2.1 Kombinace UWB a Wi-Fi – důvody a výhody . . . . .	9
2.2 Cílové použití a požadavky na systém . . . . .	9
<b>3 Realizace systému</b>	<b>10</b>
3.1 Použitý hardware . . . . .	10
3.1.1 ESP32 . . . . .	10
3.1.2 UWB modul . . . . .	11
3.1.3 Návrh a výroba PCB . . . . .	11
3.2 Softwareová implementace . . . . .	12
3.2.1 Webové rozhraní . . . . .	13
3.2.2 Lokalizace pomocí Wi-Fi fingerprintingu . . . . .	13
3.2.3 Měření vzdálenosti pomocí UWB . . . . .	14
3.2.4 Trilaterace a výpočet polohy . . . . .	15
<b>4 Testování a vyhodnocení</b>	<b>19</b>
4.1 Testovací prostředí a postupy . . . . .	19
4.2 Přesnost jednotlivých technologií . . . . .	19
<b>Závěr</b>	<b>20</b>
Shrnutí výsledků a zhodnocení . . . . .	20
Možnosti dalšího vývoje . . . . .	20
<b>Seznam příloh</b>	<b>22</b>

# Úvod

V posledních letech roste potřeba přesné a spolehlivé lokalizace objektů nejen ve venkovním prostředí pomocí systémů jako je GPS, ale také v interiéru, kde tyto technologie ztrácejí na spolehlivosti. Téma lokalizačních systémů je tedy stále aktuální, a to zejména s ohledem na rozvoj robotiky, chytrých zařízení a automatizace. Z tohoto důvodu jsem se rozhodl vytvořit vlastní hybridní lokalizační systém, který kombinuje dvě odlišné technologie – UWB, známou pro svou vysokou přesnost měření vzdálenosti, a Wi-Fi fingerprinting, který umožňuje orientační určení polohy i bez přímé viditelnosti mezi zařízeními.

Cílem této práce je navrhnout a realizovat funkční lokalizační systém, který dokáže určit polohu zařízení v prostoru s využitím obou výše zmíněných metod. Výsledný systém má být snadno použitelný, rozšiřitelný a vhodný pro integraci do dalších projektů – například jako hlavní řídicí jednotka mobilního robota nebo jako doplňkový lokalizační modul.

Při návrhu a realizaci byly využity dostupné mikrokontroléry ESP32, přičemž komunikace a ovládání systému probíhá buď přes webové rozhraní, nebo sériově. Teoretická část práce se věnuje principům obou použitých lokalizačních metod a shrnuje jejich výhody i omezení. Praktická část se zaměřuje na implementaci systému, komunikaci mezi moduly a vizualizaci výsledků.

Přínosem této práce je vytvoření flexibilního a otevřeného řešení, které může posloužit jako základ pro další rozvoj lokalizačních systémů v hobby i profesionálních aplikacích.

# 1 Teoretický základ lokalizace

## 1.1 Přehled metod lokalizace a jejich srovnání

Lokalizace objektů v prostoru je v současnosti důležitější než kdy dříve – a to zejména díky rozvoji autonomních systémů v oblasti automatizace, dopravy a osobní elektroniky. S rostoucími nároky na přesnost, rychlost a univerzálnost lokalizačních systémů vzniklo mnoho různých přístupů k řešení tohoto problému.

Jedním z přístupů je využití **computer vision (CV)** a **lidaru** – technologií, které se nejvíce přibližují lidskému způsobu vnímání prostoru. Pomocí neuronových sítí lze z obrazových dat z kamer, případně z pointcloudu získaného lidarem, odhadnout vlastní polohu i polohu okolních objektů. Nicméně, tyto metody vyžadují značný výpočetní výkon, kvalitní senzory a rozsáhlá trénovací data, což může zvyšovat náklady a složitost implementace. Nejčastěji se uplatňují v automobilovém průmyslu nebo v pokročilé robotice.

**Global Positioning System (GPS)** je nejrozšířenější lokalizační systém. Je snadno implementovatelný a cenově dostupný, což z něj činí populární volbu v mobilních telefonech a automobilech. Avšak v interiérových prostředích je jeho využití omezené kvůli slabému nebo žádnému signálu ze satelitů, což způsobuje nízkou přesnost nebo úplnou nefunkčnost systému. Přesnost se obvykle pohybuje mezi 5 až 10 metry.

Další metodou využívanou především v interiérovém prostředí je **Wi-Fi fingerprinting**. Tato technika využívá měření síly signálu (RSSI) z několika Wi-Fi přístupových bodů a porovnává je s databází známých měření (tzv. otisků, fingerprints), která byla předem nasbírána v daném prostoru. Lokalizace pak spočívá v nalezení nejbližší shody mezi aktuálním měřením a databází. Výhodou této metody je relativně nízká cena a možnost použití běžného Wi-Fi hardwaru bez nutnosti přímé viditelnosti. Nevýhodou je nutnost předchozího podrobného mapování prostoru a náchylnost na změny prostředí (např. nové překážky nebo změny v síti), které mohou snížit přesnost. Typická přesnost se pohybuje mezi 2 až 4 metry.

Další možností je měření **síly signálu** – například pomocí indikátoru Received Signal Strength Indicator (RSSI). Tento přístup je jednoduchý, levný a nevyžaduje speciální hardware. Vzdálenost lze odhadnout z útlumu signálu – čím dál je přijímač od vysílače, tím nižší je RSSI. Přesnost této metody se pohybuje mezi 2 až 4 metry za ideálních podmínek, ale může být ovlivněna rušením, odrazy signálu a dalšími faktory prostředí, což snižuje její spolehlivost.

Zcela odlišný princip využívá **Ultra-Wideband (UWB)**, který místo síly signálu měří **čas letu (Time of Flight, ToF)**. Díky tomu je možné určit vzdálenost s přesností v řádu desítek centimetrů. Její dosah je omezen na desítky metrů a vyžaduje přímou viditelnost mezi zařízeními pro optimální výkon. I když jsou náklady na implementaci vyšší než u jiných metod, UWB nabízí

vynikající přesnost a je vhodná pro aplikace, kde je klíčová precizní lokalizace.

Tabulka 1: Porovnání lokalizačních metod

Metoda	Přesnost [m]	Náklady	Typické použití
GPS	5,0–10,0	nízké	navigace, venkovní lokalizace
Wi-Fi fingerprinting	2,0–4,0	nízké	vnitřní lokalizace, IoT
RSSI (Wi-Fi/BLE)	3,0–5,0	nízké	jednoduché sledování aktiv
UWB (ToF)	0,1–0,3	vyšší	robotika, přesné sledování
Computer Vision	<1,0	vysoké	autonomní vozidla, robotika
Lidar	<0,1	velmi vysoké	autonomní systémy, mapování

Každá z popsaných metod má své výhody i nevýhody a hodí se pro jiný typ aplikace. **GPS** je ideální pro venkovní lokalizaci díky širokému pokrytí a nízké ceně, ale v interiéru selhává. **Computer vision a lidar** nabízejí velmi přesnou prostorovou orientaci, avšak za cenu vysokých výpočetních nároků a složité implementace. **Měření síly signálu (RSSI)** je jednoduché a levné, ale značně nepřesné a nespolehlivé v proměnlivém prostředí. **Wi-Fi fingerprinting** tento přístup vylepšuje pomocí databáze měření, ale vyžaduje rozsáhlou kalibraci a je náchylný na změny v prostoru. **UWB** naopak poskytuje velmi přesné měření vzdálenosti, ale vyžaduje přímou viditelnost a má omezený dosah.

**Kombinace Wi-Fi a UWB se proto jeví jako ideální kompromis** – Wi-Fi fingerprinting pokrývá větší prostor a umožňuje rychlou hrubou lokalizaci bez nutnosti přímé viditelnosti, zatímco UWB umožní v případě potřeby zpřesnění polohy na decimetrovou úroveň v okolí aktivních kotvených bodů. Vzájemným propojením obou metod lze dosáhnout vyšší přesnosti, robustnosti a univerzálnosti celého systému, což je zvláště výhodné pro mobilní roboty nebo zařízení pohybující se v dynamickém interiérovém prostředí.

## 1.2 Existující lokalizační systémy

V předchozí části byly popsány základní principy lokalizačních metod. V této kapitole budou na tyto principy navázány konkrétní systémy, které je implementují v praxi. Jedná se o reálná řešení využívaná v různých oblastech – od osobní elektroniky přes automobilový průmysl až po průmyslové a vědecké aplikace. Srovnáním těchto systémů lze lépe porozumět výhodám a omezením jednotlivých přístupů v reálném nasazení.

**Global Positioning System (GPS)** je součástí tzv. Global Navigation Satellite Systems (GNSS) – celosvětových systémů pro určování polohy pomocí družic. Kromě amerického GPS existuje například ruský GLONASS, evropské Galileo nebo čínské BeiDou. Je to nejrozšířenější metoda pro lokalizaci ve venkovních prostorech. Systém využívá signály z družic, pomocí kterých zařízení určuje svoji polohu. Přesnost se pohybuje mezi 5–10 m, v některých případech až pod

1 m při použití korekčních služeb (např. RTK). Používá se v telefonech, navigacích, dronech i autech.

**Wi-Fi fingerprinting – Google Location Services** používají obrovskou databázi známých Wi-Fi sítí (jejich SSID, BSSID a RSSI hodnoty) ke zpřesnění lokalizace zařízení, zvláště v místech bez GNSS signálu. Tato technologie je základem lokalizace v interiéru u většiny Android zařízení. Přesnost může být mezi 5–15 m v závislosti na hustotě přístupových bodů.

**Bluetooth Low Energy (BLE) beacony – Apple iBeacon, Eddystone** jsou systémy, které vysílají krátké identifikátory, které zařízení přijímají a využívají k určení polohy. Používá se často v muzeích, obchodech nebo chytrých budovách. Přesnost je většinou v rozmezí 1–5 m. Výhodou je nízká spotřeba energie a nízké náklady, nevýhodou je nutnost velkého počtu beaconů pro vyšší přesnost.

**Ultra-Wideband (UWB) – Apple AirTag, Qorvo, Sewio** využívají moderní technologie UWB umožňující velmi přesnou lokalizaci pomocí měření času letu signálu (ToF). AirTag od Applu používá UWB pro lokalizaci v rámci systému „Find My“, kde je možné přesně určit směr a vzdálenost ztraceného zařízení. Průmyslové systémy jako Qorvo nebo Sewio nabízejí řešení pro lokalizaci s přesností pod 30 cm, často používaná v logistice, skladech nebo výrobních halách.

**Vizuální lokalizace – Google ARCore a Apple ARKit** jsou systémy využívající kamery a algoritmy pro simultánní lokalizaci a mapování (SLAM), aby určily pozici zařízení v prostoru. Používají se například v rozšířené realitě nebo autonomní robotice. Výhodou je možnost fungování bez externích senzorů, ale je potřeba výpočetně náročný hardware.

## **2 Návrh hybridního systému**

### **2.1 Kombinace UWB a Wi-Fi – důvody a výhody**

Wi-Fi a UWB v tomto systému nefungují jako propojené metody, ale jako dvě nezávislé vrstvy lokalizace, které se navzájem doplňují. Zatímco Wi-Fi fingerprinting poskytuje přibližnou orientaci v prostoru – například určení aktuální místnosti či zóny – UWB slouží k přesnému měření vzdálenosti mezi dvěma zařízeními.

Tato kombinace umožňuje přizpůsobit úroveň přesnosti konkrétní situaci. Například při ztrátě přímé viditelnosti pro UWB může hrubá Wi-Fi poloha stále poskytnout základní informaci o poloze. Naopak v případech, kde je potřeba centimetrová přesnost, se využije pohyb navigačního zařízení a sekvence UWB měření k odhadu přesné polohy cílového tagu pomocí trilaterace.

Díky této dvouvrstvé architektuře lze systém použít i v různorodých a proměnlivých prostředích, kde není k dispozici stabilní podmínky pro jednu konkrétní metodu lokalizace.

### **2.2 Cílové použití a požadavky na systém**

Navržený lokalizační systém je určen pro použití v interiérovém prostředí, kde není dostupná přesná satelitní navigace. Předpokládá se nasazení například v budovách, školách nebo skladech, kde je k dispozici stabilní Wi-Fi infrastruktura a zároveň existuje potřeba lokalizace mobilních zařízení či vozidel s vyšší přesností.

Systém by měl být jednoduchý na použití a snadno rozšiřitelný do dalších projektů. Měl by být schopen pomocí Wi-Fi fingerprintingu rozlišit jednotlivé části prostoru, vypsát vzdálenost k druhému UWB zařízení a při pohybu navigátoru použít trilateraci k určení jeho přesné polohy. Výsledky lokalizace by měly být přístupné skrze webové rozhraní a připravené k dalšímu zpracování či vizualizaci.



## 3 Realizace systému

### 3.1 Použitý hardware

Pro vytvoření funkčního lokalizačního systému bylo nejprve nutné zvolit vhodné hardwarové komponenty a navrhnout vlastní desku plošných spojů pro jejich propojení. Cílem bylo dosáhnout kompaktního, přehledného a snadno rozšiřitelného řešení, které umožní spolehlivý provoz jak Wi-Fi fingerprintingu, tak přesného UWB měření vzdáleností.

#### 3.1.1 ESP32



Obrázek 1: ESP32 DevKitC [zdroj: vlastní]

Pro tento projekt bylo zvoleno vývojové prostředí ESP32 [3] díky jeho široké dostupnosti a běžnému použití v oblasti IoT. Klíčovými vlastnostmi, které rozhodly o jeho nasazení, byly vestavěný Wi-Fi modul a dostatečně výkonná dvoujádrová architektura.

Použitým modelem je ESP32 DevKitC (viz Obrázek 1).

V rámci projektu ESP32 zajišťuje:

- provádění UWB rangingu,
- skenování dostupných Wi-Fi sítí,
- provoz webového serveru,
- výpočty pro trilateraci.

Použité periferie:

- **UART** – komunikace s připojenými zařízeními,
- **SPI** – spojení s UWB modulem,
- **GPIO** – ovládání stavové LED a možnost připojení dalších externích prvků,
- **I<sup>2</sup>C** – rezervováno pro případné rozšíření o další senzory (v aktuální verzi nejsou využity).

### 3.1.2 UWB modul



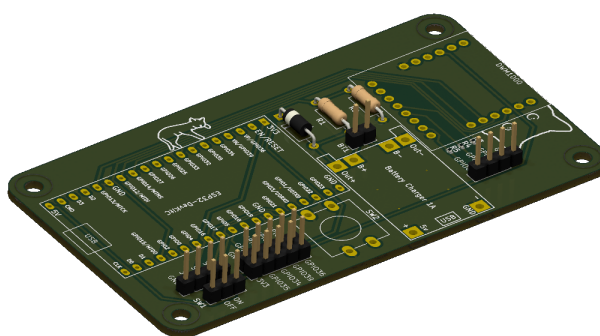
Obrázek 2: UWB modul BU01 [zdroj: vlastní]

Pro realizaci systému byl použit UWB modul BU01 (Obrázek 2), který je založen na čipu DW1000 [2] od společnosti Decawave.

Modul umožňuje přesné měření vzdáleností pomocí metody obousměrného časového měření (Two-Way Ranging, ToF) s přesností až 10 cm. Podporuje přenosové rychlosti až 6,8 Mbps a je vhodný pro real-time lokalizační aplikace v interiéru.

V projektu je modul připojen k PCB pomocí adaptérové desky, která umožňuje jednoduché propojení přes sběrnici SPI.

### 3.1.3 Návrh a výroba PCB



Obrázek 3: PCB [zdroj: vlastní]

Pro tvorbu vlastní desky plošných spojů (PCB) jsem se rozhodl kvůli přehlednosti zapojení, snížení celkových rozměrů, potřebě připojit napájecí modul a UWB modul, vlastnímu vyzkoušení návrhu PCB a také kvůli jistotě správného propojení všech součástek.

Schéma zapojení i návrh samotné desky jsem vytvořil v programu KiCAD [5]. Výroba desky proběhla u společnosti JLCPCB.

Na desce (Obrázek 3) se nachází:

- ESP32 DevKitC,
- UWB modul BU01,
- nabíjecí modul TP4056,
- rezistory s diodou pro měření stavu baterie,
- vypínač pro odpojení napájení,
- příprava pro tlačítko (na finální desce není osazeno),
- vývody pro stavové LED diody s příslušnými rezistory (indikace stavu UWB),
- konektor pro připojení baterie,
- dvakrát GND, dvakrát 3V3 a desetkrát volně přístupných GPIO pinů.

### 3.2 Softwareová implementace

Po návrhu a sestavení hardwarové části systému bylo nutné vytvořit odpovídající softwarové vybavení. Programové vybavení řídí sběr dat, komunikaci a výpočty a prezentaci výsledků celého systému.

Software je rozdělen do dvou hlavních částí:

- program běžící na mikrokontroléru ESP32, napsaný v C++ [4] s využitím platformy Arduino [1],
- podpůrné skripty běžící na počítači, napsané v Pythonu [6].

**ESP32** zajišťuje měření vzdáleností, správu webového rozhraní, sběr Wi-Fi dat a lokalizační výpočty. **Skripty v jazyce python** slouží pro sběr a přípravu fingerprintingových dat.

Softwarová architektura se dělí na následující hlavní části:

- správa webového rozhraní pro přístup k datům a ovládání systému,
- skenování Wi-Fi sítí pro potřeby fingerprintingu,
- měření vzdáleností pomocí technologie UWB,

- trilaterace a výpočet přesné polohy cílového zařízení.

V následujících podkapitolách jsou jednotlivé části popsány podrobněji.

### 3.2.1 Webové rozhraní

Webové rozhraní slouží k jednoduchému ovládání systému a zobrazení všech důležitých dat bez nutnosti fyzického připojení k zařízení. To je zvláště užitečné v případě, kdy je zařízení připevněno na pohyblivé platformě.

Webserver běží přímo na ESP32 díky vestavěnému Wi-Fi modulu. Není tedy potřeba žádné další zařízení, které zprostředkovává web, avšak je nutné znát IP adresu ESP32.

Připojení k webserveru probíhá tak, že se ESP32 připojí k existující Wi-Fi síti (nebo vytvoří vlastní přístupový bod), a po zadání jeho IP adresy do webového prohlížeče se uživatel dostane na hlavní ovládací panel.

Jednotlivé stránky webového rozhraní jsou tvořeny pomocí HTML, CSS a JavaScriptu. Stránky jsou záměrně co nejjednodušší, aby zabíraly minimum paměti a byly rychlé i na omezeném hardwaru ESP32.

Webové rozhraní je rozděleno do několika částí:

- **index.html** – hlavní stránka s odkazy na ostatní části webového rozhraní,
- **wifi\_connection.html** – správa připojení k Wi-Fi sítím,
- **location.html** – zobrazení aktuální polohy získané pomocí Wi-Fi fingerprintingu,
- **UWB.html** – zobrazování vzdálenosti a ovládání měření pomocí UWB.

### 3.2.2 Lokalizace pomocí Wi-Fi fingerprintingu

Skenování Wi-Fi sítí je klíčovým krokem pro vytvoření databáze otisků míst (fingerprintů) založených na dostupných sítích a jejich síle signálu. Z této databáze lze následně zpětně odhadovat přibližnou polohu zařízení pouze na základě aktuálně dostupných Wi-Fi sítí.

Získávání fingerprintů probíhá jednoduchým skenováním dostupných Wi-Fi sítí pomocí ESP32. Výsledkem každého skenu je seznam přístupových bodů, u kterých známe:

- **SSID** – název sítě (např. „eduroam“),
- **BSSID** – unikátní identifikátor přístupového bodu (MAC adresa, např. „00-B0-D0-63-C2-26“),

- **RSSI** – síla přijímaného signálu (v dBm, např. „-76“).

Získaná data jsou odesílána přes sériovou linku do počítače, kde je Python skript *scan.py* ukládá a přiřazuje ke každému skenu jméno místa. Výstup je uložen v jednoduchém formátu, vhodném pro další zpracování.

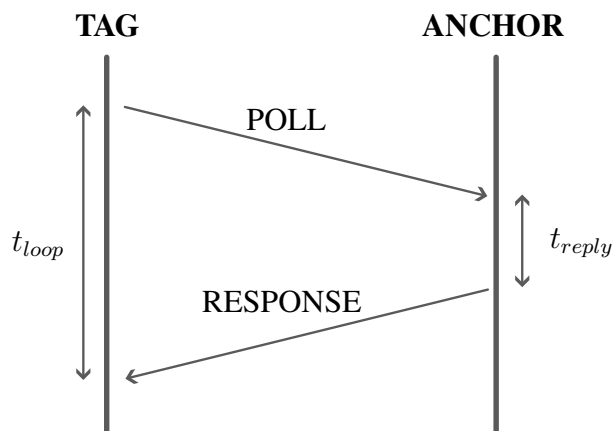
Následně druhý skript *format.py* provádí úpravu naskenovaných dat: filtruje pouze stabilní přístupové body (aby se vyloučily např. mobilní hotspoty) a průměruje více měření pro dosažení stabilnějších výsledků.

Při běhu systému lze kdykoli na vyžádání spustit Wi-Fi fingerprinting, během kterého ESP32 provede nový sken, porovná naměřené hodnoty s uloženou databází a vrátí název místa, které se nejvíce shoduje s aktuální situací.

### 3.2.3 Měření vzdálenosti pomocí UWB

UWB technologie umožňuje velmi přesné měření vzdálenosti díky měření času letu (ToF, *Time of Flight*) signálu mezi dvěma zařízeními.

Komunikace mezi ESP32 a UWB modulem probíhá pomocí sběrnice SPI. Pro ovládání modulu využívám knihovnu *thotro/arduino-dw1000* [7].



Obrázek 4: Princip měření vzdálenosti pomocí UWB [zdroj: vlastní]

Měření probíhá tak, že vysílač (tag) vyšle signál (*poll*), přijímač (anchor) na něj odpoví zprávou, ve které je obsažen údaj o čase potřebném na odpověď ( $t_{reply}$ ). Tag následně změří celkový čas od vyslání po přijetí odpovědi ( $t_{loop}$ ), viz Obrázek 4.

Z těchto dvou časů lze vypočítat skutečný čas letu podle vzorce:

$$ToF = \frac{t_{loop} - t_{reply}}{2}$$

Z naměřeného času ToF se vzdálenost určí pomocí známé rychlosti šíření signálu (rychlost světla).

Zařízení může pracovat ve dvou režimech:

- **Anchor** – čeká na zahájení komunikace,
- **Tag** – aktivně odesílá požadavky na měření.

Vzdálenost se měří opakovaně (několikrát za sekundu) a výsledky mohou být zprůměrovány pro vyšší stabilitu a eliminaci šumu.

Výstupem je vzdálenost v metrech, která je dále využívána například při výpočtu polohy pomocí trilaterace.

UWB měření je velmi přesné (běžně s odchylkou v jednotkách decimetrů), nicméně kvalita měření výrazně závisí na podmínkách prostředí, zejména na přímé viditelnosti mezi zařízeními (Line of Sight).

### 3.2.4 Trilaterace a výpočet polohy

Trilaterace [10] je analogie triangulace, která nepracuje s úhly, ale s délkami. Je založena na znalosti vzdáleností od alespoň tří (ve 2D) nebo čtyř (ve 3D) referenčních bodů. Tyto body představují statické anchory rozložené v prostoru. V našem případě se anchor pohybuje a referenční body vznikají postupně v čase, což je dostačující v případě, že je tag (měřené zařízení) statický.

Rovnice kruhů (v 2D) nebo koulí (v 3D) je dána vztahem:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = d_i^2,$$

kde  $x_i, y_i, z_i$  jsou souřadnice anchoru,  $x, y, z$  jsou souřadnice tagu a  $d_i$  je naměřená vzdálenost.

Tyto rovnice jsou kvadratické. Odečtením jedné rovnice od ostatních dojde k eliminaci kvadratických členů a vznikne soustava lineárních rovnic, kterou lze vyjádřit ve tvaru:

$$A\vec{x} = \vec{b},$$

kde  $\vec{x}$  je vektor neznámých souřadnic  $(x, y, z)$ .

V reálných scénářích, jako je trilaterace, dochází ke chybám měření a nepřesnostem polohy referenčních bodů. Proto soustava  $A\vec{x} = \vec{b}$  většinou nemá přesné řešení. Cílem je najít přibližné řešení, které minimalizuje chybu.

Metoda nejmenších čtverců hledá řešení, které minimalizuje součet druhých mocnin odchylek:

$$\|\vec{b} - A\vec{x}\|^2.$$

Tím dostaneme normální rovnici:

$$A^T A \vec{x} = A^T \vec{b},$$

jejíž řešení je dáno vztahem:

$$\vec{x} = (A^T A)^{-1} A^T \vec{b}.$$

V případě, že máme k dispozici více než minimální počet vzdáleností (například více než 3 v 2D), umožňuje tato metoda využít redundantní informace pro zpřesnění výsledné polohy.

**Stabilita rozmístění referenčních bodů** Aby bylo možné správně určit polohu, je nutné zajistit vhodné rozmístění referenčních bodů. V případě, že jsou body kolineární nebo koplanární (v 3D), trilaterace není jednoznačně řešitelná.

Kontrolu kolinearity v 2D provádím pomocí vektorového součinu. Pokud je vypočítaná plocha příliš malá, algoritmus upozorní na nevhodné rozložení bodů.

V případě 3D dat využívám Singular Value Decomposition (SVD) [9] matice bodů:

$$P = U \Sigma V^T$$

kde  $P$  je matice obsahující souřadnice referenčních bodů.

Malá hodnota třetí singulární hodnoty  $\sigma_3$  oproti  $\sigma_1$  a  $\sigma_2$  indikuje, že body jsou přibližně koplanární. Třetí sloupec  $u_3$  matice  $U$  udává normálový vektor roviny, ve které body leží.

Pokud jsou body koplanární, lze trilateraci zjednodušit následovně:

1. Nalézt rovinu pomocí normálového vektoru  $u_3$  a těžiště bodů.
2. Provést projekci všech bodů do této roviny a řešit trilateraci ve zjednodušeném 2D prostoru.
3. Výsledek poté převést zpět do 3D souřadného systému.

**Projekce bodů do roviny pomocí SVD** Pro projekci referenčních bodů i odhadované polohy do roviny využívám výstup SVD.

Nejprve spočítám těžiště všech referenčních bodů:

$$c = \frac{1}{n} \sum_{i=1}^n p_i$$

kde  $p_i$  jsou souřadnice jednotlivých bodů.

Následně odečtu těžiště od všech bodů:

$$p'_i = p_i - c$$

Tím získám souřadnice relativní k těžišti.

Rovinu, do které se mají body promítnout, určuji pomocí normálového vektoru  $\vec{n} = (a, b, c)$  a parametru  $d$  z rovnice roviny:

$$ax + by + cz + d = 0$$

Pro každý z bodů nejprve zjistím jeho vzdálenost od roviny:

$$\lambda_i = \frac{p'_i \cdot \vec{n} + d}{\|\vec{n}\|^2}$$

a následně ho posunu přímo na rovinu:

$$\bar{p}_i = p'_i - \lambda_i \vec{n}$$

Poté projektuji body do nového souřadnicového systému určeném prvními dvěma sloupci matice  $U$ :

$$q_i = \begin{bmatrix} u_1 \cdot \bar{p}_i \\ u_2 \cdot \bar{p}_i \end{bmatrix}$$

kde  $q_i$  je 2D souřadnice bodu v rovině.

Tímto způsobem jsou všechny body převedeny do 2D, kde lze provést klasickou trilateraci.

Po výpočtu polohy  $q$  v této 2D rovině se výsledek transformuje zpět do 3D prostoru:

$$p = c + \begin{bmatrix} u_1 & u_2 \end{bmatrix} q$$

Tím je zajištěno, že vypočtená poloha respektuje původní 3D souřadný systém.



**Kalmanův filtr pro zpřesnění polohy** Při určování polohy pomocí trilaterace dochází k náhodným odchylkám způsobeným šumem v měřených vzdálenostech. K potlačení těchto odchylek a ke zpřesnění výsledné polohy používám Kalmanův filtr [8].

Kalmanův filtr je optimální odhadovací algoritmus pro systémy se šumem, který kombinuje:

- předpověď polohy na základě předchozího stavu,
- nové naměřené hodnoty z trilaterace,
- model nejistoty měření a model dynamiky systému.

Filtr iterativně aktualizuje odhadované souřadnice podle rovnic:

- **Předikce stavu:**  $\vec{x}_{k|k-1} = F_k \vec{x}_{k-1|k-1} + B_k u_k$
- **Předikce kovarianční matice:**  $P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
- **Výpočet Kalmanova zisku:**  $K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$
- **Aktualizace odhadu:**  $\vec{x}_{k|k} = \vec{x}_{k|k-1} + K_k \tilde{y}_k$
- **Aktualizace kovarianční matice:**  $P_{k|k} = (I - K_k H_k) P_{k|k-1}$

## **4 Testování a vyhodnocení**

### **4.1 Testovací prostředí a postupy**

Testování probíhalo v domácím prostředí a ve škole. Pro ověření přesnosti měření jsem postupně posouval tag o známé vzdálenostní přírůstky a zaznamenával hodnoty naměřené lokalizačním systémem.

Kromě reálného testování jsem vytvořil také simulační prostředí v Pythonu, kde jsem ověřil správnost matematických výpočtů trilaterace a Kalmanova filtru. Simulace umožnila rychlejší ladění algoritmů a ověření chování systému v ideálních i zarušených podmínkách.

Porovnáním naměřených dat se skutečnými hodnotami jsem zjistil, že systém bez dodatečné kalibrace nevykazoval dostatečnou přesnost. Proto jsem implementoval kalibrační funkci, která upravuje hodnoty anténního zpoždění (antenna delay) tak, aby byla výsledná měřená vzdálenost co nejpresnější. Tento postup výrazně zlepšil shodu mezi reálnou a vypočítanou pozicí.

### **4.2 Přesnost jednotlivých technologií**

Systém využívající technologii UWB dosahoval po kalibraci přesnosti měření lepší než 10 cm. Přesnost Wi-Fi lokalizace byla výrazně nižší, avšak postačovala k určení přibližné polohy — například k rozpoznání, ve které místnosti se zařízení nachází.

# **Závěr**

## **Shrnutí výsledků a zhodnocení**

Cílem práce bylo vytvořit snadno implementovatelný navigační modul využívající technologii UWB a Wi-Fi.

Podařilo se mi realizovat měření vzdálenosti pomocí UWB, návrh a ověření trilaterálního algoritmu (simulovaného v Pythonu), lokalizaci pomocí Wi-Fi a vytvořit obecný firmware pro platformu ESP32.

Kompletní lokalizační modul zatím nebyl otestován na fyzické mobilní platformě, a tedy výsledky trilaterace jsou prozatím pouze teoretické. Ostatní funkce byly ověřeny a fungují dle očekávání.

## **Možnosti dalšího vývoje**

Co se týče zlepšení, tak protokol pro komunikaci není perfektní, celý projekt není extra uklizený, a není kompletní user manual a nějaké quality of life features.

Momentální kalibrace je jen tak slepená dohromady a není ideální pro větší změny v měření.

Vzhledem k tomu že jsme měli pouze 2 UWB tak jsme museli mít pouze jeden anchor, takže přesné nemohlo být implementováno. Architektura v kódu tam proto je, jenom to zbývá implementovat.

reálné nasazení na venkovním testování a spoluprací s tímto projektem pro doladění jakýchkoli nedostatků

Projekt by se dal nasadit v autonomní robotice.

Další rozvoj projektu by se mohl zaměřit na několik oblastí:

- Vylepšení komunikačního protokolu mezi zařízeními.
- Důkladnější úklid a optimalizaci kódu, včetně vytvoření podrobné uživatelské dokumentace a přidání funkcí pro snadnější použití.
- Rozšíření systému na více než dva UWB moduly pro umožnění plné 3D lokalizace pomocí více anchorů. Architektura v kódu je již na tuto možnost částečně připravena.
- Testování v reálných venkovních podmínkách a doladění systému na základě praktických zkušeností.

Projekt má potenciál pro nasazení v oblasti autonomní robotiky, kde by mohl sloužit jako navigační modul pro vnitřní i vnější lokalizaci.

## **Seznam příloh**

příloha I - schéma zapojení PCB

## Seznam použité literatury a zdrojů informací

- [1] Arduino. *Arduino Documentation*. 2025. URL: <https://docs.arduino.cc/> (cit. 29. 04. 2025).
- [2] DECAWAVE Ltd. *DW1000 User Manual, Version 2.05*. 2015. URL: [https://thetoolchain.com/mirror/dw1000/dw1000\\_user\\_manual\\_v2.05.pdf](https://thetoolchain.com/mirror/dw1000/dw1000_user_manual_v2.05.pdf) (cit. 29. 04. 2025).
- [3] Espressif Systems. *ESP32 Technical Reference Manual, Version 5.3*. 2025. URL: [https://www.espressif.com/sites/default/files/documentation/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf) (cit. 29. 04. 2025).
- [4] ISO/IEC. *ISO/IEC 14882:2017: Programming Languages — C++*. International Organization for Standardization. 2017.
- [5] KiCad. *KiCad Documentation*. 2025. URL: <https://docs.kicad.org/> (cit. 29. 04. 2025).
- [6] Python Software Foundation. *Python Programming Language*. 2025. URL: <https://www.python.org/> (cit. 29. 04. 2025).
- [7] Thotro. *arduino-dw1000: A library that offers functionality to use Decawave's DW1000 IC*. 2021. URL: <https://github.com/thotro/arduino-dw1000> (cit. 29. 04. 2025).
- [8] Wikipedia. *Kalman Filter*. 2025. URL: [https://en.wikipedia.org/wiki/Kalman\\_filter](https://en.wikipedia.org/wiki/Kalman_filter) (cit. 29. 04. 2025).
- [9] Wikipedia. *Singular Value Decomposition*. 2025. URL: [https://en.wikipedia.org/wiki/Singular\\_value\\_decomposition](https://en.wikipedia.org/wiki/Singular_value_decomposition) (cit. 29. 04. 2025).
- [10] Wikipedia. *Trilateration*. 2025. URL: <https://en.wikipedia.org/wiki/Trilateration> (cit. 29. 04. 2025).