

Monitoring vozidel v průmyslovém areálu: Implementace GNSS systémů

Bc. Zdenek Tihelka

Diplomová práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav automatizace a řídicí techniky

Akademický rok: 2024/2025

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Bc. Zdenek Tihelka
Osobní číslo:	A22324
Studijní program:	N0714A150006 Automatické řízení a informatika v průmyslu 4.0
Forma studia:	Kombinovaná
Téma práce:	Monitoring vozidel v průmyslovém areálu: Implementace GNSS systémů
Téma práce anglicky:	Vehicle Tracking in Industrial Complex: GNSS System Implementation

Zásady pro vypracování

1. Rešerše dostupných řešení pro monitoring vozidel.
2. Analýza požadavků na monitorovací systém.
3. Návrh a implementace systému pro agregaci dat z GNSS modulů.
4. Začlenění systému do infrastruktury společnosti.
5. Testování a ověření funkčnosti systému.

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. MISRA, Pratap and PER ENGE. Global Positioning System: Signals, Measurements, and Performance. 3rd ed. Lincoln, MA: Ganga-Jamuna Press, 2010. ISBN 978-0-9709544-5-6.
2. TEUNISSEN, Peter J. G. and Oliver MONTENBRUCK, eds. Springer Handbook of Global Navigation Satellite Systems. Springer International Publishing, 2017. ISBN 978-3-319-42928-1.
3. BHATTA, Basudeb. Global Navigation Satellite Systems: New Technologies and Applications. 2nd ed. Boca Raton: CRC Press, 2020. ISBN 978-0-367-47882-2.
4. KUMAR, Sandeep, Arpan KUMAR KAR and Geetanjali PANDEY, eds. Internet of Things and Its Applications: Select Proceedings of ICIA 2020. Springer, 2022. ISBN 978-981-16-0215-4.
5. GILL, Saikat, ed. Internet-of-Things (IoT) Systems. Academic Press, 2020. ISBN 978-0-12-819399-8.

Vedoucí diplomové práce: **Ing. Jakub Novák, Ph.D.**
Ústav automatizace a řídicí techniky

Konzultant diplomové práce: **Ing. Tomáš Dulík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **6. prosince 2024**
Termín odevzdání diplomové práce: **27. května 2025**



doc. Ing. Jiří Vojtěšek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Ve Zlíně dne 6. prosince 2024

PROHLÁŠENÍ AUTORA DIPLOMOVÉ PRÁCE

Beru na vědomí, že

- odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení diplomové práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – diplomovou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na diplomové práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Cílem této diplomové práce je navrhnout, implementovat a otestovat komplexní systém pro monitorování vozidel v průmyslové oblasti s využitím technologií GNSS (Global Navigation Satellite System). Navržený systém umožňuje uživatelům sledovat aktuální polohu vozidel v reálném čase a zároveň poskytuje rozhraní pro správu, analýzu a vizualizaci získaných dat. Řešení zahrnuje využití moderních technologií, jako jsou GNSS moduly, mikrořadiče, Wi-Fi komunikace v prostředí multi AP a lokální ukládání dat do flash paměti. Data jsou přenášena do serverové aplikace a ukládána do databáze, což umožňuje flexibilní reporting a pokročilou vizualizaci ve formě tabulek, grafů a mapových podkladů. Nedílnou součástí práce je integrace navrženého řešení do stávající firemní infrastruktury s důrazem na zabezpečení dat v rámci vnitřní podnikové sítě. V závěrečné fázi prací je ověřována funkčnost celého systému v reálných provozních podmínkách areálu, kde jsou posuzovány funkční i nefunkční požadavky.

Klíčová slova: Monitorování vozidel, Mikrořadič, GNSS, Wi-Fi komunikace, Databáze, Vizualizace dat

ABSTRACT

The aim of this thesis is to design, implement and test a comprehensive system for vehicle monitoring in the industrial area using GNSS (Global Navigation Satellite System) technologies. The designed system allows users to track the current location of vehicles in real time and at the same time provides an interface for management, analysis and visualization of the obtained data. The solution includes the use of modern technologies, such as GNSS modules, microcontrollers, Wi-Fi communication in a multi AP environment and local data storage in flash memory. Data is transferred to a server application and stored in a database, which allows flexible reporting and advanced visualization in the form of tables, graphs and map data. An integral part of the work is the integration of the designed solution into the existing corporate infrastructure with an emphasis on data security within the internal corporate network. In the final phase of the work, the functionality of the entire system is verified in real operating conditions of the premises, where functional and non-functional requirements are assessed.

Keywords: Vehicle monitoring, Microcontroller, GNSS, Wi-Fi communication, Database, Data visualization

Děkuji tímto vedoucímu práce, panu Ing. Jakubu Novákovi, Ph.D., za cenné rady, podnětné připomínky a odborné konzultace během vypracování této diplomové práce. Současně děkuji své rodině a kolegům za pochopení, trpělivost a podporu, kterou mi poskytovali po celou dobu jejího zpracování.

OBSAH

ÚVOD.....	13
I TEORETICKÁ ČÁST	15
1 ÚVOD DO TEORETICKÉ ČÁSTI.....	16
2 REŠERŠE DOSTUPNÝCH ŘEŠENÍ.....	17
2.1 PROBLEMATIKA MONITOROVÁNÍ VOZIDEL	17
2.2 PARAMETRY REŠERŠE.....	17
2.3 PŘEHLED TECHNOLOGIÍ A KOMPONENT (GNSS, MIKROŘADIČ, WI-FI, ÚLOŽIŠTĚ DAT)	18
2.4 PROBLEMATICKÉ OBLASTI PŘI IMPLEMENTACI MONITOROVACÍCH SYSTÉMŮ.....	19
2.5 AKTUÁLNÍ TRENDY A VÝVOJ V MONITOROVÁNÍ VOZIDEL.....	19
2.6 KOMERČNÍ SYSTÉMY PRO MONITOROVÁNÍ VOZIDEL.....	20
2.7 SROVNÁNÍ SYSTÉMŮ PRO MONITOROVÁNÍ VOZIDEL	23
2.8 ZÁVĚR REŠERŠE DOSTUPNÝCH ŘEŠENÍ.....	23
3 ANALÝZA POŽADAVKŮ NA MONITOROVACÍ SYSTÉM.....	24
3.1 FUNKČNÍ POŽADAVKY	24
3.1.1 Sledování polohy vozidel v reálném čase	24
3.1.2 Záloha dat při ztrátě spojení.....	25
3.1.3 Vizualizace dat a reporting.....	26
3.2 NEFUNKČNÍ POŽADAVKY	27
3.2.1 Bezpečnost a integrita dat (necloudové řešení).....	27
3.2.2 Spolehlivost systému.....	28
3.2.3 Integrace do infrastruktury společnosti	29
3.2.4 Jednoduchost údržby a rozšiřitelnost	30
3.2.5 Závěr analýzy požadavků na monitorovací systém	31
4 TEORETICKÝ ZÁKLAD POUŽITÝCH TECHNOLOGIÍ	32
4.1 GNSS TECHNOLOGIE.....	32
4.1.1 Systém GPS, Galileo, GLONASS, BeiDou	32
4.1.2 Principy přesnosti měření.....	33
4.2 KOMUNIKAČNÍ TECHNOLOGIE	34
4.2.1 Wi-Fi (standardy, UniFi AP).....	35
4.2.2 UART (komunikace GNSS modulu, programování, konfigurace modulu)	36
4.2.3 SPI (komunikace s flash pamětí).....	37
4.3 TECHNOLOGIE SBĚRU A UKLÁDÁNÍ DAT	38
4.3.1 NOR flash paměť (výhody a omezení)	38
4.3.2 Databáze MS SQL (vhodnost pro řešení, CTE dotazy, JSON).....	39
4.4 SOFTWAREVÉ TECHNOLOGIE.....	40

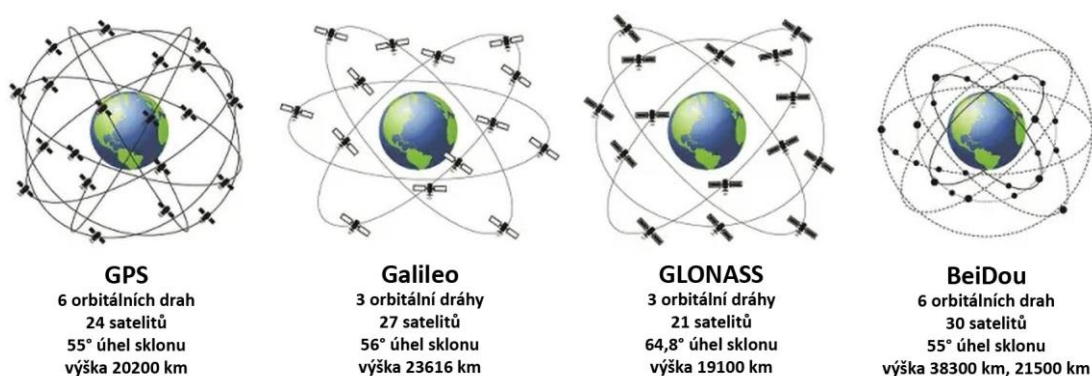
4.4.1	Arduino IDE a programování mikrořadiče (FreeRTOS)	40
4.4.2	Python jako serverová technologie.....	41
4.4.3	Vizualizační technologie (HTML, JavaScript, JSON).....	43
4.4.4	Mapové API – alternativa či rozšíření.....	44
II	PRAKTICKÁ ČÁST	45
5	NÁVRH SYSTÉMU	46
5.1	NÁVRH HARDWAROVÉ ČÁSTI.....	46
5.1.1	Schéma zapojení ESP32-C6 a U-blox NEO-M9N, zdroje, podpůrný HW	46
5.1.2	Napájení (převod 12 V → 3,3 V)	47
5.1.3	GPIO řízení zapínání a uspaní zařízení	48
5.2	NÁVRH SOFTWAREOVÉ ČÁSTI.....	49
5.3	NÁVRH SERVEROVÉ ČÁSTI.....	50
6	IMPLEMENTACE SYSTÉMU	52
6.1	IMPLEMENTACE HARDWARU.....	52
6.1.1	Konkrétní zapojení a komponenty	52
6.2	IMPLEMENTACE SOFTWAREU ZAŘÍZENÍ	57
6.2.1	Úlohy FreeRTOS pro čtení polohových dat, Wi-Fi komunikaci a další	57
6.2.2	Ukládání dat do NOR flash (management dat při nedostupnosti Wi-Fi)	59
6.3	IMPLEMENTACE SERVEROVÉ ČÁSTI.....	62
6.3.1	Struktura Python aplikace	62
6.3.2	Formát dat, komunikace JSON	63
6.3.3	Implementace databáze (struktura, dotazy CTE, JSON výstupy)	64
6.4	IMPLEMENTACE UŽIVATELSKÉHO ROZHRAŇÍ.....	64
6.4.1	Dynamická vizualizace polohy vozidel na mapě	64
6.4.2	Vizualizace reportů (HTML, JavaScript, tabulky)	65
7	INTEGRACE DO INFRASTRUKTURY SPOLEČNOSTI.....	67
7.1	POPIS STÁVAJÍCÍ INFRASTRUKTURY	67
7.2	INTEGRACE SYSTÉMU DO FIREMNÍ SÍTĚ (AP UNIFI, KANÁLY 2.4 GHz)	67
7.3	KONFIGURACE SERVERU A DATABÁZE.....	68
7.4	ZAJIŠTĚNÍ BEZPEČNOSTI (IZOLACE, BEZPEČNÝ PŘÍSTUP K DATŮM).....	69
8	TESTOVÁNÍ A OVĚŘENÍ FUNKČNOSTI SYSTÉMU	70
8.1	METODIKA TESTOVÁNÍ (V REÁLNÝCH PODMÍNKÁCH).....	70
8.1.1	Testovací scénáře (vnitřní/venkovní prostředí, výpadky Wi-Fi).....	70
8.2	FUNKČNÍ TESTY SYSTÉMU	71
8.2.1	Testování sběru dat (GNSS, NOR flash, Wi-Fi)	71
8.2.2	Testování serverové a databázové vrstvy	71
8.2.3	Testování uživatelského rozhraní	72
8.3	VYHODNOCENÍ VÝSLEDKŮ TESTOVÁNÍ.....	72

8.3.1	Splnění požadavků	73
8.3.2	Identifikace možných problémů a návrhy na zlepšení	73
ZÁVĚR		75
SEZNAM POUŽITÉ LITERATURY.....		77
SEZNAM OBRÁZKŮ		80
SEZNAM TABULEK.....		81
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		82
SEZNAM PŘÍLOH.....		86

ÚVOD

Moderní průmyslové podniky a logistické společnosti stále častěji využívají systémy pro monitorování svých vozidel. Hlavními důvody pro zavádění těchto systémů je zvyšování efektivity a bezpečnosti provozu, lepší správa vozového parku a plnění legislativních požadavků. Monitorování polohy a provozních parametrů vozidel v reálném čase umožní optimalizovat trasy, snižovat náklady na palivo i údržbu a předcházet prostojům, čímž se zvyšuje celková efektivita provozu. Nepřetržitý dohled nad vozidly zároveň poskytuje data pro prevenci rizikového chování. Systém také manažerům usnadňuje správu vozového parku, neboť mají k dispozici detailní přehled o využití a pohybu každého vozidla, což usnadňuje například plánování údržby či efektivnější využití kapacit. V neposlední řadě může být sledování vozidel vyžadováno i různými právními předpisy nebo interními směrnici, takže implementace monitorovacího systému v podniku pomáhá splnit i tyto povinnosti a standardy.

K realizaci takto pojatého monitorování vozidel je nezbytná spolehlivá technologie pro přesné určování polohy. Tou jsou globální navigační satelitní systémy (GNSS), mezi něž patří například americký GPS, evropský Galileo, ruský GLONASS či čínský BeiDou (obrázek č. 1). Tyto systémy umožňují prostřednictvím signálů z družic určit zeměpisnou polohu objektu vybaveného přijímačem s přesností na jednotky metrů. GNSS tedy tvoří základ lokalizačních služeb, neboť díky nim mohou monitorovaná vozidla poskytovat své aktuální souřadnice v reálném čase. Tato technologie je proto klíčová pro sledování pohybu vozidel, jelikož zajišťuje nepřetržitý přísun dat o jejich poloze.



Obrázek 1 GNSS – 4 systémy

(zdroj: <https://mettatec.com/how-many-channels-does-a-gnss-receiver-need/>)

GNSS technologie tedy tvoří nezbytný stavební kámen moderních systémů pro sledování vozidel. Proto se tato diplomová práce zaměřuje na návrh konkrétního monitorovacího systému využívajícího GNSS pro sledování vozidel v podmínkách jednoho průmyslového areálu. Cílem je vyvinout řešení, které umožní nepřetržité sledování pohybu vozidel v rámci areálu a využití získaných dat k zefektivnění provozních procesů a lepší správě vozového parku. Při návrhu systému jsou brány v úvahu specifické podmínky daného areálu i požadavky organizace, aby výsledné řešení spolehlivě fungovalo v reálném provozu. Očekávaným přínosem je optimalizace logistických operací v rámci areálu, zvýšení bezpečnosti a vytvoření nástroje pro efektivní dohled nad firemními vozidly.

Aby bylo možné takový systém úspěšně realizovat, je nutné navrhnout komplexní řešení, které zahrnuje více vzájemně propojených složek. Nejde pouze o vhodný hardware ve vozidlech a komunikační technologie pro přenos dat, ale také o infrastrukturní zajištění ukládání a zpracování nasbíraných údajů, jejich vizualizaci pro uživatele a komplexní zabezpečení systému. Hardwarová část sestává z GNSS palubních jednotek instalovaných ve vozidlech a dalších modulů například komunikačních, které zajišťují odesílání informací o poloze do centrálního úložiště. Na serverové straně je třeba vybudovat nebo použít stávající databázové úložiště pro přijatá data a softwarovou aplikaci, jež umožní jejich prezentaci a analýzu. Důraz je rovněž kladen na integraci navrženého řešení do stávající podnikové infrastruktury, tak, aby systém monitorování vozidel mohl komunikovat s dalšími používanými nástroji a zapadal do firemních procesů.

Následující kapitoly práce se proto postupně věnují jednotlivým krokům vedoucím k realizaci tohoto řešení. Nejprve je provedena podrobná rešerše současných technologií a existujících systémů pro monitorování vozidel, která poskytne teoretický základ pro návrh konečného řešení. Poté se práce zaměří na samotný návrh monitorovacího systému pro daný areál, včetně volby vhodných technických prostředků a softwarové architektury. Následuje popis implementace navrženého řešení a provedení testování v reálných podmínkách. Získané poznatky jsou nakonec shrnuty a vyhodnoceny, přičemž jsou posouzeny přínosy navrženého systému a navržena případná doporučení pro budoucí rozvoj.

I TEORETICKÁ ČÁST

1 ÚVOD DO TEORETICKÉ ČÁSTI

Teoretická část této diplomové práce je zaměřena na vytvoření uceleného odborného základu pro následný návrh a realizaci systému sledování vozidel s využitím globálních družicových navigačních systémů (GNSS). V této části jsou postupně rozpracována teoretická východiska nezbytná pro pochopení problematiky, definování požadavků na systém a seznámení se s klíčovými technologiemi, které budou využity v praktické části diplomové práce.

Druhá kapitola je věnována rešerši dostupných řešení monitorování vozidel. Jejím cílem je zanalyzovat současný stav technologií a existujících systémů pro sledování vozového parku, popsat jejich funkce, možnosti a omezení, a tím vymezit prostor pro inovaci nebo zvolení vlastního přístupu. Třetí kapitola se zaměřuje na analýzu funkčních a nefunkčních požadavků připravovaného systému. Tato kapitola formuluje, jaké funkce musí systém plnit a jaké kvalitativní vlastnosti (například spolehlivost, škálovatelnost či bezpečnost) musí splňovat, aby vyhověl očekáváním uživatelů a provozním podmínkám. Čtvrtá kapitola poté poskytuje přehled použitých technologií, které tvoří technický základ navrhovaného řešení. V rámci ní jsou představeny hlavní nástroje, platformy a principy (včetně GNSS a souvisejících komunikačních technologií), jež budou využity při realizaci systému.

Uvedené kapitoly společně vytvářejí teoretický základ nutný k tomu, aby bylo možné v navazující praktické části diplomové práce navrhnout, implementovat a otestovat komplexní systém pro sledování vozidel pomocí GNSS. Teoretická část tak plní roli průvodce danou problematikou a zajistí, že další části práce budou postaveny na pevných znalostních základech.

2 REŠERŠE DOSTUPNÝCH ŘEŠENÍ

Tato kapitola popisuje vymezení klíčových parametrů vlastní rešerše pro získání přehledu relevantních a efektivních řešení, které se nejvíce podobají či shodují s cílovými požadavky na chování celého monitorovacího systému.

2.1 Problematika monitorování vozidel

Monitorování vozidel představuje soubor technických a softwarových řešení, jejichž primárním účelem je efektivně zaznamenávat, sledovat, analyzovat a optimalizovat provoz vozového parku. Tato oblast má klíčový význam zejména v podnikových a průmyslových areálech, kde se kladou vysoké požadavky na spolehlivost, bezpečnost, ekonomiku provozu a logistickou efektivitu.

2.2 Parametry rešerše

Monitorovací systémy musí splňovat specifické požadavky, které zahrnují zejména:

- **Řešení založená na GNSS technologii**

Kvalita lokalizace závisí na použité technologii (GNSS, GSM triangulace, RFID, BLE). Pro většinu průmyslových aplikací je zásadní GNSS technologie, umožňující přesné určení polohy vozidel s přesností v jednotkách metrů, případně až centimetrů při použití RTK (Real-Time Kinematic).

- **Necloudová řešení (data neopouštějí infrastrukturu firmy)**

Vzhledem k citlivosti údajů o pohybu vozidel musí být systémy navrženy tak, aby data neopouštěla infrastrukturu firmy bez kontroly, a byla chráněna před neoprávněným přístupem či zneužitím.

- **Komunikace a ukládání dat (Wi-Fi, lokální úložiště)**

Průmyslové prostředí klade vysoké požadavky na spolehlivost zařízení, schopnost odolávat výpadkům spojení (ukládání dat do lokálního úložiště), a dále na odolnost vůči průmyslovým podmínkám (teplota, vlhkost, vibrace).

- **Integrace do existující infrastruktury**

Nové řešení musí být snadno integrováno do existujícího podnikového ekosystému (databáze, webové rozhraní, plánovací systémy), což vyžaduje otevřenost a dostupnost vhodných API (Application Programming Interface) či integračních nástrojů.

2.3 Přehled technologií a komponent (GNSS, mikrořadič, Wi-Fi, úložiště dat)

V oblasti monitorování vozidel jsou nejčastěji využívány následující technologie:

- **GNSS (GPS, Galileo, GLONASS, BeiDou)**

Primární způsob určení polohy vozidla, využívající satelitní signál s vysokou přesností lokalizace.

- **Mobilní síť (GSM)**

Nejčastěji využívaná technologie pro přenos dat mezi vozidlem a centrálním serverem. Má celoplošné pokrytí a umožňuje kontinuální přenos dat.

- **Wi-Fi síť (WLAN)**

Výhodné zejména v průmyslových areálech, kde je dostupná infrastruktura lokálních sítí, což zajišťuje bezpečný přenos dat a snižuje provozní náklady oproti GSM.

- **Mikrořadiče**

Klíčové prvky monitorovacích zařízení, které zajišťují samotný sběr dat z GNSS modulů, jejich předzpracování, řízení komunikace a ukládání dat.

- **Lokální ukládání dat (flash paměti, SD/eMMC úložiště)**

V případě ztráty připojení umožňuje lokální paměť zajistit nepřetržitý záznam polohy a provozních parametrů s následnou synchronizací po obnovení spojení.

2.4 Problematické oblasti při implementaci monitorovacích systémů

Zavádění systémů pro monitoring vozidel naráží na několik praktických problémů:

- **Problém s pokrytím signálu (Indoor Environment)**

GNSS signál může být oslaben nebo úplně ztracen uvnitř hal, skladů či podzemních prostor, kde se pak musí využívat alternativní technologie – například kombinace GNSS a IMU (Inertial Measurement Unit).

- **Přesnost vs. cena zařízení**

Moduly umožňující vysokou přesnost (RTK, PPP) jsou cenově výrazně nákladnější než standardní GNSS moduly.

- **Bezpečnost a ochrana dat**

Zajištění robustního zabezpečení proti kybernetickým útokům a zneužití dat je nezbytnou podmínkou při nasazení v podnikových systémech.

2.5 Aktuální trendy a vývoj v monitorování vozidel

Současné trendy v oblasti monitorování vozidel směřují k následujícím oblastem:

- **Vyšší přesnost díky RTK technologii**

Primární způsob určení polohy vozidla, využívající satelitní signál s vysokou přesností lokalizace.

- **Lokální zpracování (Edge Computing)**

Zpracování dat přímo ve vozidlech snižuje zatížení komunikační sítě a zvyšuje odezvu systému v reálném čase.

- **AI a pokročilá analytika**

Aplikace AI pro různé predikce (údržba, optimalizace tras), analýzu chování řidičů a okolního provozu či automatickou detekci mimořádných událostí.

2.6 Komerční systémy pro monitorování vozidel

Na trhu je dostupná řada komerčních řešení pro monitorování vozidel (Fleet Management), která využívá zejména technologie GNSS v kombinaci s mobilními sítěmi. Do výběru byli zahrnuti hlavní poskytovatelé mobilních služeb – O2, T-Mobile, Vodafone. Do výběru byli přidáni další systémy (např. Linde) pro porovnání klíčových parametrů: způsob lokalizace, přenos dat, zálohování při výpadku spojení, úložiště dat a možnosti integrace do jiných systémů. Kompletní srovnání hodnocených parametrů rešerše je uvedeno v tabulce č. 1.

Všechna uvedená řešení nabízejí on-line sledování polohy vozidel a elektronickou knihu jízd, liší se však implementací technických detailů a doplňkovými funkcemi:

- **O2 Car Control**

Služba od operátora O2 Czech Republic a.s., která umožňuje správu a sledování firemních vozidel pomocí instalované jednotky ve vozidle. Systém je využíván pro online monitoring vozového parku, knihu jízd, kontrolu spotřeby paliva a další fleet management funkce. Data jsou přístupná přes webovou aplikaci nebo v mobilní aplikaci O2 Car Control [1].

- **T-Mobile Autopark**

Komplexní GPS monitorovací systém provozovaný operátorem T-Mobile Czech Republic, a.s. Skládá se z palubní GPS jednotky s datovou SIM, webové aplikace pro dispečink a řady nadstavbových modulů (např. identifikace řidičů, měření paliva). Služba poskytuje on-line sledování vozidel, knihu jízd a další funkce pro efektivní správu flotily vozidel. Systém lze využívat i v režimu off-line záznamu dat pro pozdější synchronizaci [2].

- **Vodafone Vozový park (Webdispečink)**

Telematická služba od Vodafone Czech Republic a.s., určená pro online dohled nad vozovým parkem osobních i nákladních vozidel. Realizována je ve spolupráci se společností Princip a.s. (platforma Webdispečink). Systém umožňuje sledovat polohu vozidel v reálném čase, generovat elektronickou knihu jízd, vyhodnocovat styl jízdy řidičů a zobrazovat provozní údaje z vozidel. Součástí je webový portál a mobilní aplikace Webdispečink s mapovými podklady [3].

- **SHERLOG Trace**

Monitorovací systém od společnosti SHERLOG Technology, a.s., která se specializuje na elektronické zabezpečení vozidel a satelitní vyhledávání. Systém SHERLOG Trace kromě bezpečnostních funkcí nabízí plnohodnotný GPS monitoring vozidel v reálném čase a elektronickou knihu jízd. Vhodný je pro osobní i nákladní vozidla a pracovní stroje – cílem je mít neustálý přehled o pohybu vozidel, ujetých kilometrech, tankování paliva apod. Data jsou uživatelům dostupná přes webovou a mobilní aplikaci [4].

- **Linde connect (Fleet Management)**

Linde Material Handling Česká republika s.r.o. nabízí vlastní systém connect pro správu flotil manipulačních vozíků a dalších firemních vozidel v areálu. Původně byl zaměřen zejména na vysokozdvizné vozíky (sledování přístupu řidičů, provozních hodin, baterií apod.), ale novější modul connect:mapování pohybu vozíků přidal možnost přesné lokalizace strojů pomocí satelitů. Linde connect tak umožňuje dohled nad pohybem vozíků v reálném čase s přesností až 5 metrů a zobrazení polohy strojů na mapě závodu. To pomáhá jak při logistice v rozlehlých areálech, tak při zabezpečení (například rychlé dohledání vozíku v případě krádeže i mimo areál) [5].

Kromě výše uvedených hlavních hráčů existují v ČR i další komerční systémy pro sledování vozidel, často nabízené specializovanými telematickými firmami. Patří mezi ně např. GPS Dozor, LEVEL Systems, AutoGPS, KomTeS a další. Tyto systémy funkčně pokrývají podobné potřeby – online GPS lokalizaci vozidel, knihu jízd, měření provozních parametrů – a liší se spíše cílovou skupinou či formou nabídky [6; 7]:

- **JSON (JavaScript Object Notation) API**

Systém GPS Dozor je cloudový a poskytuje otevřené JSON API pro přístup k aktuálním i historickým datům o poloze. Umožňuje tak integraci dat o vozidlech do firemních aplikací a podporuje i pokročilé funkce jako propojení s elektronickými knihami jízd, tachografy nebo plánovači tras.

- **Další technologie**

Některá řešení využívají kromě GNSS i doplňkové lokační technologie. Například kombinace GPS/GSM lokalizace (při ztrátě GPS signálu se poloha odhadne z GSM sítě), identifikace pomocí RFID/Bluetooth (pro rozpoznání řidiče nebo přívěsu) či telematické jednotky napojené na CAN/FMS sběrnici vozidla (získávání údajů z řídicí elektroniky vozidla). Tyto prvky se často objevují jako součást komplexních fleet management systémů – např. jednotky Princip Vetronics umí číst data z CAN (Controller Area Network) sběrnice, mají akcelerometr pro detekci stylu jízdy a umí komunikovat i s externími čidly přes Bluetooth.

Tabulka 1 Hodnoty parametrů řešerše

Poskytovatel	Typ lokalizace	Přenos dat
O2 Car Control	GNSS (GPS, Glonass, Galileo)	Mobilní síť O2 (GSM)
T-Mobile Autopark	GNSS (GPS)	Mobilní síť T-Mobile (GSM, SMS)
Vodafone Vozový park	GNSS (GPS, Galileo, BeiDou)	Mobilní síť Vodafone (GSM)
SHERLOG Trace	GNSS (GPS)	Mobilní síť (GSM)
Linde connect	GNSS (GPS, Glonass, Galileo)	Mobilní síť (globální eSIM)

Poskytovatel	Zálohování při výpadku	Úložiště dat
O2 Car Control	Interní paměť jednotky, automatické odeslání po obnově	Cloudové (server O2)
T-Mobile Autopark	Interní paměť jednotky, kapacita cca 20000 km	Cloudové (centrální server T-Mobile)
Vodafone Vozový park	Flash paměť jednotky (uchování 2–6 měsíců dat)	Primárně cloudové, volitelně lokální
SHERLOG Trace	Interní paměť jednotky, automatický přenos po obnově	Cloudové (datové centrum Sherlog)
Linde connect	Interní paměť jednotky, upozornění (Geofencing)	Lokální nebo cloudové (volitelné)

Poskytovatel	Možnosti integrace
O2 Car Control	API, propojení s ERP/CRM, webové a mobilní rozhraní
T-Mobile Autopark	Webový portál, API (na vyžádání), integrace s externími systémy (např. CCS)
Vodafone Vozový park	API, propojení s ERP/logistikou, přístup přes web i mobilní aplikaci
SHERLOG Trace	Webové rozhraní, API (na vyžádání), podpora exportu CSV, PDF, napojení na SAP aj.
Linde connect	API/webové služby, přímá integrace s IT systémy, portál connect:desk/cloud

(zdroj: [1; 2; 3; 4; 5])

2.7 Srovnání systémů pro monitorování vozidel

Všechna výše zmíněná řešení jsou prioritně GNSS – určují polohu vozidel pomocí satelitů a data přenášejí mobilní sítí do centrální aplikace (typicky webové/cloudové). Liší se v detailech:

- Telekomunikační operátoři (O2, T-Mobile, Vodafone) nabízejí řešení s integrací do jejich IoT (Internet of Things) infrastruktury a zákaznickou podporou. Specializované firmy (Princip/Webdispečink, Sherlog, GPS Dozor) zase často pružnější integraci a širší možnosti přizpůsobení.
- Funkce zálohování dat při výpadku spojení bývá standardem u všech seriózních systémů – moderní jednotky mají interní paměť, aby zajistily nepřetržitý záznam pohybu.
- Úložiště dat je převážně cloudové, ale některé systémy (např. Linde connect, případně instalace Webdispečink) umožňují i lokální ukládání.
- Integrace do dalších systémů je v menší či větší míře podporována – většina poskytovatelů proto nabízí API nebo exporty, aby zákazník mohl propojit data o vozidlech se svým informačním systémem (například pro kombinaci s daty z logistických procesů nebo optimalizaci tras).

2.8 Závěr rešerše dostupných řešení

Z provedeného srovnání komerčně dostupných systémů pro monitorování vozidel vyplývá, že žádné z analyzovaných řešení nenabízí plnou míru otevřenosti, nezávislosti a technické přizpůsobitelnosti bez kompromisů. I když některé platformy umožňují částečné lokální ukládání dat nebo integraci do podnikových systémů, jejich provoz je zpravidla pevně svázán s infrastrukturou poskytovatele nebo s komerčním modelem dané služby. Je tedy zřejmé, že zákazník nemá přímou kontrolu nad architekturou systému ani nad nakládáním s citlivými daty. To může být v průmyslovém prostředí, kde jsou kladeny vysoké nároky na bezpečnost, spolehlivost a kontinuitu provozu, nežádoucí. Uvedené nedostatky dostupných řešení vedly k potřebě hledat alternativní přístup bez těchto omezení.

3 ANALÝZA POŽADAVKŮ NA MONITOROVACÍ SYSTÉM

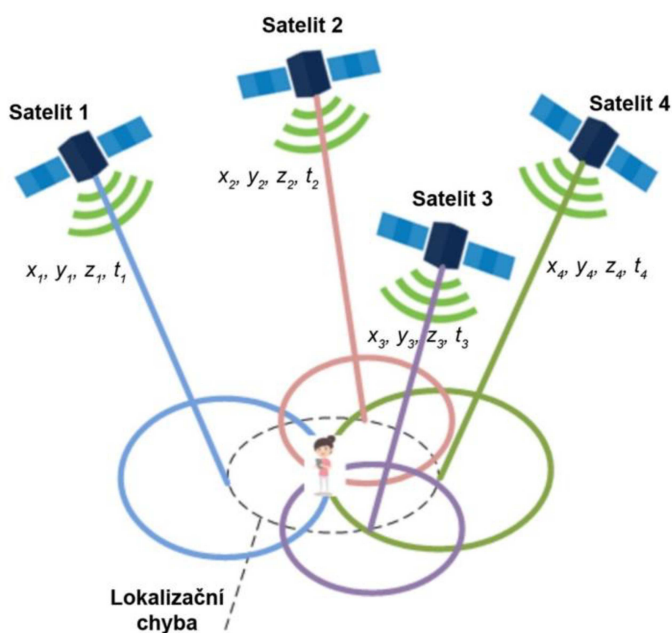
Jde o detailní rozbor požadavků kladených na systém pro monitorování vozidel v průmyslovém areálu. Požadavky jsou rozděleny na funkční a nefunkční – vycházejí z reálných potřeb podniku, technických možností a požadovaných parametrů navrhovaného řešení.

3.1 Funkční požadavky

Vymezují, co má monitorovací systém dělat – jaké funkce a služby má poskytovat. Pro systém sledování vozidel v průmyslovém areálu jsou definovány následující klíčové funkční požadavky:

3.1.1 Sledování polohy vozidel v reálném čase

Zásadním požadavkem je možnost průběžně sledovat polohu všech vozidel v areálu v reálném čase. Systém tedy musí v krátkých intervalech získávat aktuální souřadnice každého vozidla a okamžitě je předávat do centrální aplikace. Tohoto lze dosáhnout využitím technologie GNSS na vozidle a průběžným přenosem dat přes Wi-Fi síť do serveru. GNSS systém poskytuje globální určení polohy a času za všech povětrnostních podmínek, pokud má přijímač přímou viditelnost alespoň čtyř družic (obrázek č. 2) [8, s. 136–144].



Obrázek 2 Určování GNSS polohy

(zdroj: <https://compass.rauias.com/current-affairs/global-positioning-system-gps-work/>)

V kontextu průmyslového areálu to znamená, že každý vůz vybavený GNSS modulem bude schopen zjistit svou zeměpisnou polohu (zeměpisná délka, šířka) s přesností v řádu jednotek metrů. GNSS modul musí být vícesystémový přijímač (GPS, Galileo, GLONASS, BeiDou) a umožňuje tak maximalizovat dostupnost signálu a přesnost určování polohy díky paralelnímu příjmu až čtyř satelitních systémů. To je výhodné zejména v prostředí průmyslového areálu, kde mohou být části oblohy zakryté budovami či překážkami – více souběžných GNSS zvyšuje šanci, že přijímač bude mít vždy dostatek satelitů k výpočtu polohy [9, s. 385–389].

Reálný čas v tomto kontextu znamená, že informace o poloze vozidla se v systému objeví prakticky ihned (s co nejmenším zpožděním) po jejím změření. To obvykle představuje periodu aktualizace v řádu jednotek sekund. GNSS přijímač musí umožňovat výstup dat minimálně $1 \times$ za sekundu (1 Hz), což pokrývá potřeby plynulého sledování i rychleji se pohybujících vozidel. Data z GNSS jsou odesílána řídicím mikrořadičem přes Wi-Fi síť okamžitě po každém získání, aby centrální systém měl neustále aktuální přehled. V centrální aplikaci (serveru) budou polohy vozidel zobrazeny na mapovém podkladu nebo v jiné vizualizaci, což umožní dispečerům či operátorům areálu sledovat aktuální rozmístění vozidel. Takový způsob monitorování odpovídá běžné praxi fleet management systémů – vozidla vybavená GNSS přijímači průběžně zasílají svou polohu do centrálního dispečinku, kde jsou jejich pozice vykreslovány v mapě [10, s. 52–59; 12].

Reálné časové sledování polohy zvyšuje efektivitu provozu (např. lze rychleji reagovat na potřeby přesunu techniky) a zajišťuje i bezpečnost – je možné v každém momentě určit, kde se které vozidlo nachází.

3.1.2 Zálaha dat při ztrátě spojení

Monitorovací systém musí být schopen neztrácet nasbíraná data ani při výpadku spojení. V podmínkách průmyslového areálu může dojít k dočasné nedostupnosti Wi-Fi sítě (výjezd vozidla z pokrytého prostoru, rušení signálu, odstávka přístupového bodu apod.). Proto je požadováno, aby jednotka na vozidle byla schopna v případě ztráty konektivity přejít do režimu lokálního ukládání dat. Prakticky to znamená, že údaje o poloze (a případně další telemetrická data) budou ukládány do vestavěné nevolatilní paměti (např. NOR flash). Tento způsob dočasného ukládání zajistí, že se žádné naměřené body trasy neztratí a budou později odeslány po obnovení spojení. Technicky lze tento požadavek realizovat také tzv. pasivním

sledováním (Passive Tracking), kdy se data ukládají lokálně a ke stažení na server dojde dodatečně [10, s. 65–72; 13].

Za běžného provozu bude systém fungovat v režimu aktivního sledování – polohy se odesílají průběžně (Real-Time). Pokud však zařízení detekuje výpadek Wi-Fi, přepne se do pasivního režimu a začne data ukládat do paměti. Jakmile je spojení opět dostupné, záznamy se dodatečně přenesou na server. Tím se zajistí, že v centrální databázi budou uloženy všechny telemetrické hodnoty polohy vozidla, i za období, kdy bylo vozidlo off-line. Velikost paměti musí být dimenzována tak, aby pojala dostatek údajů i pro delší výpadky – například pro několik hodin jízdy (v závislosti na frekvenci vzorkování polohy).

Záloha dat výrazně přispívá ke spolehlivosti systému z hlediska datové dostupnosti – systém odolává dočasným problémům s konektivitou a splňuje tak požadavek, aby žádná naměřená data nebyla ztracena. Po obnovení spojení dojde k synchronizaci dat se serverem (Store-and-Forward mechanismus v IoT architektuře) [11, s. 139–144].

3.1.3 Vizualizace dat a reporting

Dalším funkčním požadavkem je přehledná vizualizace nasbíraných dat a generování reportů. Surová GNSS data (sekvence geografických souřadnic v čase) je potřeba transformovat do podoby, která uživatelům systému umožní snadno pochopit informace o pohybu vozidel. Systém proto musí obsahovat aplikační vrstvu, jež poskytne uživatelské rozhraní pro zobrazení polohy vozidel na mapě v reálném čase. Operátor průmyslového areálu tak uvidí aktuální rozmístění všech sledovaných vozidel přímo na obrazovce na fotomapě nebo plánu areálu.

Kromě on-line mapové vizualizace je požadováno, aby systém podporoval reporting – tvorbu sestav a přehledů z historických dat. Uživatelé (manažeři logistiky, dispečeri) budou chtít generovat například reporty o provozu vozidel: ujetá vzdálenost každého vozidla, doba pohybu vs. stání, počet jízd apod. Systém by měl umožnit vybrat období a vozidlo či skupinu vozidel a automaticky z databáze vytáhnout související data, provést potřebné výpočty a prezentovat výsledky ve srozumitelné formě (tabulka, graf).

Vizualizační a reportovací funkce zvyšují užitnou hodnotu monitorovacího systému. Umožňují nejen sledovat aktuální stav, ale i analyzovat historické trendy a optimalizovat provoz. Tyto informace mohou vést k organizačním opatřením pro zefektivnění přeprav v rámci areálu. Systém tedy musí být navržen tak, aby data nasbíraná v terénu byla snadno dostupná pro vizualizaci i další budoucí zpracování.

Pro vizualizační funkce je tedy nutné uchovávat data do databázové struktury. K dispozici je databázové řešení MS SQL (Microsoft Structured Query Language), které je kompatibilní s množstvím analytických nástrojů a reportovacích služeb. Již při návrhu databázového schématu je třeba myslet na možné dotazy a analýzy, které vzniknou v budoucnosti a uchovávat případně další atributy, aby šly snadno agregovat a filtrovat.

Výše uvedené funkční požadavky tvoří základní rámec pro realizaci monitorovacího systému a definují, jaké konkrétní funkce musí systém splňovat, aby bylo dosaženo očekávané efektivity a provozní spolehlivosti. Jejich dodržení je klíčové pro úspěšné zavedení systému do provozu a naplnění potřeb uživatelů v průmyslovém prostředí.

3.2 Nefunkční požadavky

Určují kritéria, jako je bezpečnost, spolehlivost, interoperabilita či udržitelnost systému. Pro navrhovaný monitorovací systém byly stanoveny tyto hlavní nefunkční požadavky:

3.2.1 Bezpečnost a integrita dat (necloudové řešení)

Systém musí zajišťovat bezpečné nakládání s daty a jejich integritu. Poloha firemních vozidel je citlivou informací, a proto je nutné zabránit neoprávněnému přístupu k těmto datům i jejich případné manipulaci. Architektura řešení musí být navržena jako necloudová – veškeré komponenty běží v rámci lokální infrastruktury podniku, data jsou ukládána do interní MS SQL databáze a nejsou sdílena s externími cloudovými službami. Toto samo o sobě zvyšuje bezpečnost, neboť eliminuje některá rizika spojená s přenosem dat přes internet a ukládáním u třetích stran (útoky na cloudové servery, únik dat z cizí platformy apod.). Systém bude fungovat autonomně a nebude závislý na internetovém připojení – to zaručí kontinuitu provozu i při výpadku vnější konektivity a lepší kontrolu nad tím, kdo může k systému přistupovat.

Přestože je systém interní, je třeba implementovat opatření pro ochranu dat v rámci lokální Wi-Fi sítě. Samotná Wi-Fi síť (stávající systém UniFi) musí být zabezpečena odpovídajícím šifrováním (minimálně WPA2 nebo podobně).

Integrita dat znamená, že data nelze po cestě pozměnit a že v databázi budou uložena přesně tak, jak byla naměřena. Databázová vrstva by měla používat transakce a referenční integritu, aby nedocházelo k nekonzistencím při zápisu. Důležité je také zálohování databáze, i když to již spadá do provozních opatření.

Navržený systém musí mít robustní hardware a u GNSS modulu je požadována detekce spoofingu a rušení [9, s. 385–389].

Celá komunikace a ukládání probíhá uvnitř infrastruktury podniku, což zjednodušuje kontrolu nad bezpečností – administrátor nastavuje přístupová pravidla v rámci LAN (Local Area Network), firewallu apod., aniž by musel řešit otevření systému vůči internetu.

3.2.2 Spolehlivost systému

Systém pro monitoring vozidel musí fungovat spolehlivě i v nepřetržitém provozu. Spolehlivost zahrnuje několik aspektů:

- **Dostupnost**

Systém by měl mít vysokou dobu provozuschopnosti. To znamená minimalizovat neplánované výpadky serverové aplikace a zajistit, že i jednotky na vozidlech poběží trvale. K tomu je potřeba kvalitní hardware navržený pro nepřetržitý provoz s nízkou spotřebou. Tento bude napájen z palubní sítě vozidla. Server bude běžet na operačním systému Windows Server podle firemních preferencí a databázový server MS SQL bude správně nakonfigurován, aby zvládl dlouhodobý běh bez ztráty výkonu.

- **Odolnost vůči chybám**

Jak již bylo řečeno v kontextu zálohy dat, systém musí odolat dílčím poruchám (např. výpadek Wi-Fi, restart zařízení) bez ztráty dat či dlouhého přerušení služby. Spolehlivost zahrnuje i přesnost a správnost dat: GNSS by měl poskytovat údaje v požadované kvalitě. Modul musí mít pokročilé funkce detekce rušení a spoofingu, což přispěje ke spolehlivosti – systém dokáže rozpoznat, pokud by se někdo pokusil falšovat GNSS signál, nebo pokud je signál nekvalitní vinou rušení, a může na to reagovat [9, s. 385–389].

- **Životnost a robustnost**

V průmyslovém prostředí je zařízení vystaveno prachu, výkyvům teplot, vibracím na vozidle apod. Proto by hardware měl být vhodně umístěn a chráněn (např. zapouzdřená krabička s krytím IP65 a upevněná tak, aby nevadily vibrace). To je spíše otázka fyzického provedení, ale patří to k požadavkům na systém jako celek – musí vydržet provozní podmínky prostředí areálu bez častých poruch.

- **Obnova po selhání**

V případě, že přece jen dojde k výpadku (např. server musí být restartován kvůli aktualizaci nebo dojde k poruše napájení), systém by měl být schopen rychle obnovit činnost a doplnit chybějící data. To souvisí s předchozími body – po restartu serveru se zařízení znovu připojí k síti a uložená data se odešlou na server, takže databáze bude kompletní. Celý proces by měl být co nejvíce automatizovaný, bez nutnosti ručního zásahu (Self-Healing).

3.2.3 Integrace do infrastruktury společnosti

Monitorovací systém musí být navržen tak, aby zapadl do stávající infrastruktury podniku a nenarušil její běžné fungování. To znamená využít existující zdroje a technologie, které firma již používá, namísto zavádění zcela nových, neslučitelných platforem. Především se to týká těchto rovin:

- **Síťová integrace**

Systém využívá stávající podnikovou Wi-Fi síť (UniFi) pro komunikaci zařízení. To klade požadavek, aby použité zařízení bylo schopno se k této Wi-Fi bezpečně připojit (musí podporovat odpovídající standardy a zabezpečení). UniFi infrastruktura typicky umožňuje centrální správu přístupových bodů. Systém sledování by měl být navržen tak, aby bylo možné jej do této správy zahrnout – např. zařízení mohou být v samostatné Wi-Fi či VLAN (Virtual Local Area Network) pro IoT zařízení, s omezeným přístupem jen k serveru aplikace. Integrací do sítě je také myšleno, že datový tok generovaný systémem bude odpovídat možnostem sítě.

- **Serverová a databázová integrace**

Back-end systému poběží na Python serveru napojeném na MS SQL databázi. Požadavkem je, aby nové řešení bylo kompatibilní s verzí MS SQL nasazenou ve firmě. Databáze s telemetrickými údaji (poloha vozidel atp.) může běžet na stejném databázovém serveru, který společnost využívá pro jiné aplikace, případně na další instanci serveru (pokud to bude žádoucí). Tím se zjednoduší údržba – zálohování databází, nastavení uživatelských oprávnění a další rutiny budou obdobné jako u ostatních firemních dat.

- **Softwarová integrace a interoperabilita**

Systém by měl nabízet rozhraní, která v případě potřeby umožní propojení s jinými systémy společnosti, pokud bude potřeba, a to pomocí přímo API rozhraním, přes které si jiné aplikace mohou potřebná data dotazovat. Tento požadavek zajišťuje, že řešení nebude izolované, ale může se stát součástí širšího ekosystému IT nástrojů ve firmě. Pro uživatele by mělo být možné přistupovat k monitorovacímu systému snadno z firemních počítačů – ideálně přes webové rozhraní v intranetu, bez nutnosti instalovat speciální software.

- **Organizační začlenění**

Po technické stránce by nasazení systému nemělo vyžadovat zásadní reorganizaci. Například obsluha systému (sledování mapy, generování reportů) již spadá pod existující role – např. dispečeri dopravy v areálu. Systém musí podporovat přihlašování uživatelů (autentizaci) již na serverové aplikaci napojenou na firemní adresář AD (Active Directory), aby bylo možné využít stávající účty a přístupová práva uživatelů. Tím je zajištěno jednotné přihlašování a snadná správa uživatelů v souladu s firemními standardy.

3.2.4 Jednoduchost údržby a rozšiřitelnost

Neméně důležitým nefunkčním požadavkem je, aby systém byl snadno udržovatelný a rozšiřitelný do budoucna. Vzhledem k tomu, že technologické projekty mívají dlouhou životnost, je třeba myslet na to, jak se bude systém spravovat a případně vylepšovat.

Jednoduchost údržby pro zajištění rutinní operace jako aktualizace softwaru, výměna nebo doplnění zařízení, diagnostika problémů apod., by měly být co nejvíce zjednodušené. Na úrovni serverové aplikace by aktualizace měly být plánovatelné a proveditelné bez dlouhých odstávek – ideálně s možností nasadit změny postupně.

Systém musí být v budoucnu rozšiřitelný o nové funkce s vynaložením přiměřeného úsilí, namísto nutnosti celý systém nahrazovat. Navržená architektura by měla počítat s tím, že se požadavky mohou vyvíjet. Příkladem rozšíření může být:

- **Škálovatelnost (rozšíření na více vozidel)**

Pokud se počet sledovaných vozidel zvýší, systém to musí zvládnout bez zásadní změny. To znamená, že použitá síťová infrastruktura a serverový výkon má určitou rezervu a že adresní prostor pro identifikaci zařízení je dostatečný. Rozšíření počtu

zařízení by mělo spočívat jen v přidání dalších jednotek do systému (nahrání firmwaru, přiřazení ID a konfigurace) a jejich registraci na serveru.

- **Úprava aplikačních funkcí**

Vznikne-li v budoucnu požadavek na sofistikovanější analýzy (např. prediktivní údržba na základě ujetých kilometrů, integrace s plánováním údržby strojů) měl by jít software na serveru upravit. Kód bude psán v jazyce Pythonu a data uložena v MS SQL databázi, tímto je zajištěna vysoká flexibilita – tyto technologie jsou dobře zdokumentované a rozšířené, proto nebude problém v budoucnu navázat na původní kód.

Z pohledu nefunkčních požadavků tak navržený systém bude stát na principech modulárnosti a standardizace, což zaručuje, že bude udržitelný po celou dobu životního cyklu. Pokud dojde ke změně požadavků, systém bude připraven se přizpůsobit bez potřeby začínat od nuly. Tento požadovaný přístup odpovídá obecně doporučovaným praktikám pro návrh IoT systémů – architektura by měla být dostatečně pružná, aby vyvážila současné funkce a budoucí rozvoj. Díky tomu investice do vybudování monitorovacího systému přinese dlouhodobý užitek a nebude za pár let zastaralý [11, s. 137–144].

3.2.5 Závěr analýzy požadavků na monitorovací systém

Na základě specifikovaných funkčních a nefunkčních požadavků bylo rozhodnuto vyvinout vlastní systém monitorování vozidel, který by plně odpovídal stanoveným potřebám bez nutnosti technologických kompromisů. Vlastní návrh umožní zcela kontrolovat tok dat, způsob komunikace i datové úložiště, a zároveň zajistit plnou integraci do stávající podnikové infrastruktury. Tento přístup zároveň umožňuje vysokou míru rozšiřitelnosti a nezávislosti na externích službách, což je klíčové z hlediska dlouhodobé udržitelnosti a bezpečnosti systému. Výsledkem analýzy je tedy směřování k vlastnímu řešení, které respektuje konkrétní podmínky prostředí i provozní požadavky bez nutnosti přizpůsobovat se existujícím produktům.

4 TEORETICKÝ ZÁKLAD POUŽITÝCH TECHNOLOGIÍ

Tato kapitola poskytuje teoretický základ k technologiím, které budou v rámci diplomové práce využity pro realizaci systému sledování vozidel. Věnuje se principům GNSS systémů, komunikačním rozhraním, technologiím pro ukládání dat a vybraným softwarovým řešením, která jsou důležitá pro pochopení fungování navrženého systému.

4.1 GNSS technologie

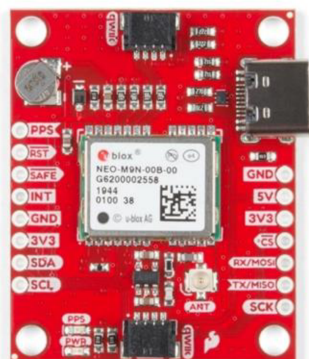
Podkapitola shrnuje klíčové principy a vlastnosti GNSS systémů. Zabývá se stručným přehledem jednotlivých družicových systémů a zdůrazňuje jejich výhody a principy dosažení přesnosti měření, které jsou zásadní pro určování polohy sledovaných vozidel.

4.1.1 Systém GPS, Galileo, GLONASS, BeiDou

Globální družicové navigační systémy zahrnují několik nezávislých soustav satelitů, z nichž nejznámější je GPS. Dále existuje GLONASS, Galileo a BeiDou – všechny tyto systémy poskytují globální pokrytí navigačním signálem. Každý ze systémů má vlastní konstelaci satelitů: GPS provozuje kolem 32 satelitů na 6 orbitálních rovinách, GLONASS pracuje s přibližně 24 satelity na 3 rovinách, Galileo plánuje plnou konstelaci 30 satelitů (24 aktivních na 3 rovinách) a BeiDou třetí generace zahrnuje přes 30 satelitů v kombinaci geostacionárních, inklinovaných a středních drah. Všechny systémy jsou vzájemně interoperabilní z pohledu uživatele – přijímač schopný přijímat více konstelací může využít podstatně více satelitů současně. Kombinované využití více GNSS soustav významně zvyšuje počet viditelných satelitů, zlepšuje jejich geometrické rozložení na obloze a tím zlepšuje přesnost, spolehlivost a dostupnost určení polohy i za ztížených podmínek (např. v zastavěných oblastech). Každý ze zmíněných GNSS systémů je navíc provozován jiným subjektem, což zvyšuje celkovou robustnost – Galileo je pod civilní kontrolou (EU) a poskytuje služby nezávislé na GPS (USA), GLONASS (Rusko) či BeiDou (Čína). GNSS tak jako celek nabízí uživatelům globální pokrytí, nepřetržitý provoz a bezplatný zdroj přesného času a polohy, což představuje zásadní výhodu oproti dřívějším regionálním nebo pozemským navigačním systémům [8, s. 136–144; 9, s. 385–389; 10, s. 174–182, 325–332].

Moderní přijímače obvykle integrují podporu všech hlavních GNSS – jde o mnoha kanálový přijímač například na obrázku č. 3, který dokáže přijímat signály ze systémů GPS, GLONASS, Galileo a BeiDou současně. Díky tomu dosahuje určování polohy přesnosti okolo 1,5 metru a zajišťuje maximální dostupnost družic pro výpočet polohy. Využití více

soustav zároveň tak uživateli poskytuje vyšší přesnost a kontinuitu navigačních služeb i v případě výpadku nebo rušení části systémů.



Obrázek 3 Vývojový GNSS modul

(zdroj: <https://www.sparkfun.com/sparkfun-gps-breakout-neo-m9n-u-fl-qwiic.html>)

4.1.2 Principy přesnosti měření

Přesnost určování polohy pomocí GNSS je ovlivněna celou řadou faktorů. Základem je princip měření pseudovzdálenosti k několika družicím – přijímač měří čas šíření signálu od družice a ze znalosti polohy družic vypočítá svou polohu. Ideálně by čtyři družice stačily k určení 3D polohy a času. V praxi se však do měření promítají systematické chyby a náhodné odchylky. Hlavní zdroje nepřesností zahrnují nepřesnosti palubních atomových hodin družic řádově ± 2 m, odchylky v odhadované oběžné dráze družice $\pm 2,5$ m, zpoždění signálu průchodem ionosférou typicky ± 5 m a troposférou $\pm 0,5$ m, dále šum a omezené rozlišení přijímače $\pm 0,3$ m a vícenásobné odrazy signálu (MultiPath) ± 1 m. Tyto chyby se vektorově sčítají a projevují se v konečném výpočtu polohy. Kromě nich má zásadní vliv také geometrie rozmístění družic vůči uživateli – je vyjádřena faktorem zředění přesnosti DOP (Dilution of Precision). Nepříznivá geometrie (vysoké hodnoty DOP) může výrazně zesílit dopad ostatních chyb na výslednou polohu.

Pro zvýšení přesnosti se používá několik metod. Jednou z nich je příjem korekčních dat ze systému SBAS (Satellite Based Augmentation System), jako je evropský EGNOS (European Geostationary Navigation Overlay Service) nebo americký WAAS (Wide Area Augmentation System), které vysílají opravné údaje pro ionosféru, dráhy a hodiny družic. Další možností je DGNSS (Differential Global Navigation Satellite System), kdy se porovnávají měření mobilního přijímače s referenční stanicí na známé pozici – odchylky změřené referenční stanicí se použijí k opravě měření uživatele. Diferenciální korekce dokáží zlepšit přesnost určování polohy na úroveň pod 1 metr.

Ještě dále jde metoda RTK, která využívá fázové měření nosné vlny a on-line přenos korekcí z referenční stanice. RTK dosahuje přesnosti 2–3 cm, avšak na omezenou vzdálenost od stanice a za cenu nutnosti stabilního datového spojení. Podobnou přesnost jako RTK nabízí i metoda PPP, která místo místních referenčních stanic využívá globální korekce drah a hodin družic vypočtené z husté sítě stanic – PPP nevyžaduje lokální referenční stanici, ale má delší konvergenční čas (desítky minut). V neposlední řadě k vyšší přesnosti přispívá i vícepásmový příjem (Multi-Frequency), kdy přijímač sleduje signály na více frekvencích (např. L1 a L5) – z rozdílu zpoždění na dvou frekvencích lze odhadnout a eliminovat ionosférické zpoždění. Kombinací více technik (více konstelací, vícepásmového příjmu a diferenciálních korekcí) je tak možné dosáhnout velmi vysoké přesnosti měření polohy (řádově decimetry až centimetry). Pro běžné civilní použití (např. turistické GNSS moduly) se většinou uplatňuje jen základní jednofrekvenční přijímač s přesností okolo několika metrů, nicméně i ten těží z výhod GNSS příjmu, má-li k dispozici např. signály GPS i Galileo současně [10, s. 381–402].

4.2 Komunikační technologie

V této části jsou popsány komunikační technologie, které umožňují přenos dat mezi jednotlivými komponentami systému. Specificky se zaměřuje na bezdrátovou technologii Wi-Fi a sériové rozhraní UART (Universal Asynchronous Receiver-Transmitter) a SPI (Serial Peripheral Interface), jež jsou klíčové pro zajištění spolehlivé výměny dat mezi mikrořadičem, GNSS modulem a datovým úložištěm. Příklad mikrořadiče (obrázek č. 4), který disponuje popisovanými komunikačními technologiemi.



Obrázek 4 Vývojový modul mikrořadiče

(zdroj: <https://www.espressif.com/en/products/devkits>)

4.2.1 Wi-Fi (standardy, UniFi AP)

Wi-Fi je označení pro bezdrátové sítě WLAN založené na standardech rodiny IEEE 802.11. Tyto standardy definují bezdrátovou komunikaci v pásmech 2,4 GHz a 5 GHz (nově i 6 GHz u Wi-Fi 6 a Wi-Fi 7) a umožňují realizaci lokálních počítačových sítí bez potřeby kabeláže. Od vydání první verze 802.11 v roce 1997 (rychlost 2 Mb/s) prošla Wi-Fi technologie dramatickým vývojem. Standard 802.11b (1999) umožnil 11 Mb/s v pásmu 2,4 GHz, následovaný 802.11g (2003) s 54 Mb/s. Významným milníkem byl 802.11n (Wi-Fi 4, 2009), který zavedl MIMO techniku a kanály 40 MHz, čímž dosáhl reálně stovek Mb/s a možnost simultánního provozu v pásmu 2,4 i 5 GHz. Další generace 802.11ac (Wi-Fi 5, 2013) působící čistě v pásmu 5 GHz přinesla širší kanály (80 a 160 MHz) a vícenásobný MIMO přenos k více uživatelům (MU-MIMO). Díky tomu 802.11ac dokáže nabídnout teoretickou propustnost až kolem 6,9 Gb/s. Novější standard 802.11ax (Wi-Fi 6, 2019) zvyšuje efektivitu pomocí technologie OFDMA a vyšší modulace (1024-QAM). U nejnovějšího standardu 802.11be (Wi-Fi 7, 2022) je signál modulován efektivněji pomocí 4096-QAM oproti 1024-QAM u Wi-Fi 6 a podporuje současnou komunikaci s velkým počtem zařízení a zlepšuje propustnost v hustých sítích. Klíčovou výhodou Wi-Fi je snadná dostupnost a interoperabilita zařízení – produkty různých výrobců spolupracují díky certifikaci Wi-Fi Alliance, což umožňuje masové rozšíření technologie a cenově efektivní budování sítí [14, s. 275–283].

Pro potřeby tohoto projektu je podstatné, že mikrořadič má zabudované rozhraní Wi-Fi, které umožní bezdrátovou komunikaci se sítí LAN. Typické nasazení je takové, že mikrořadič se připojí do existující Wi-Fi sítě a komunikuje s back-endovým serverem.

V enterprise prostředí bývá existující Wi-Fi síť realizována řadou AP s centralizovanou správou a jednou přístupovou SSID (Service Set Identifier) sítí. Jako příklad moderního přístupového bodu lze uvést – sledované zařízení (mikrořadič) je v pohybu a síť pokrývá více AP, lze tedy zajistit nepřerušované spojení při přechodu mezi jednotlivými AP. Wi-Fi technologie se tak vyznačuje širokou kompatibilitou, vysokou datovou propustností a dostatečným dosahem v interiéru (desítky metrů), což z ní činí vhodnou volbu pro bezdrátovou komunikaci mezi senzory, mikrořadiče a serverem v rámci jedné budovy či areálu [15, s. 584–591].

4.2.2 UART (komunikace GNSS modulu, programování, konfigurace modulu)

UART je univerzální asynchronní sériové rozhraní sloužící k obousměrné komunikaci mezi dvěma zařízeními. UART je hardwarový modul, který zajišťuje převod paralelních dat na sériový bitový tok a naopak, včetně časování a rámcování dat. Jde o velmi rozšířený a levný způsob full-duplex sériové komunikace, běžně využívající linkové úrovně RS-232. Komunikace přes UART nevyžaduje hodinový signál (data jsou synchronizována dohodnutou přenosovou rychlostí – tzv. baud rate) a přenáší se rámce obsahující start bit, datové bity, volitelný paritní bit a stop bity. Kvůli absenci sdílené hodinové linky musí obě strany používat předem stejnou baud rate (např. 9 600 Bd, 115 200 Bd). UART zpravidla zajišťuje pouze point-to-point komunikaci mezi dvěma uzly [16, s. 78–90].

V kontextu navrhovaného systému má UART hned několik důležitých rolí. Za prvé slouží pro komunikaci mikrořadiče s GNSS modulem. GNSS přijímače poskytují naměřená data ve formě textových zpráv protokolu NMEA 0183 nebo binárních zpráv proprietárního protokolu právě přes sériové rozhraní UART. Mikrořadič tedy přijímá každou sekundu datové věty obsahující aktuální polohu, čas, počet viditelných satelitů apod. – tyto informace dále zpracovává. UART zde funguje v asynchronním režimu s určitou rychlostí (9 600 Bd či vyšší – podle konfigurace GNSS modulu). Za druhé se UART využívá při programování mikrořadičů. Mikrořadiče mají integrovaný bootloader, který umožňuje nahrávat nový firmware do flash paměti mikrořadiče právě přes sériovou linku UART. Třetí využití sériové linky je případná konfigurace GNSS modulu či ladění. Tento modul lze konfigurovat zasíláním příkazů přes sériový port. UART je tedy jednoduchá, ale všestranná komunikační technologie v navrhovaném systému – zajišťuje nižší úroveň komunikace mezi hardware komponentami a také vstup/výstup systému vůči vývojáři.

Z technického hlediska je výhodou UARTu jeho široká podpora v mikrořadičích (většinou obsahují více nezávislých UART jednotek) a jednoduchost použití. Nemusí být složitý protokol – stačí nastavit shodnou rychlost a formát rámce na obou stranách. Omezením je, že UART je vždy pouze mezi dvěma zařízeními a nelze jednoduše rozšířit na vícebodovou sběrnici (na rozdíl od protokolu jako RS-485, který je založen na odlišné fyzické vrstvě). V rámci diplomové práce jsou však UART spojení vždy jen dvojbodová (mikrořadič ↔ GNSS modul, mikrořadič ↔ PC). Maximální dosažitelná rychlost UART komunikace závisí na konkrétním hardware, běžně v řádu desítek kbit/s. UART tedy plně dostačuje pro přenos GNSS dat (kterých je běžně pod 1 kB/s) i pro nahrávání firmware do mikrořadiče.

4.2.3 SPI (komunikace s flash pamětí)

SPI je sériová synchronní sběrnice určená pro rychlou komunikaci mezi mikrořadičem (master) a periferními zařízeními (slave). Jde o jedno z nejrozšířenějších rozhraní pro připojení čipů, jako jsou senzory, AD/DA převodníky, paměti či displeje. Na rozdíl od UARTu je SPI synchronní – používá oddělenou taktovací linku (SCLK), na jejíž hrany jsou synchronizována data. Standardní konfigurace SPI využívá čtyři vodiče:

- SCLK – sériový hodinový signál generovaný master zařízením.
- MOSI – data vysílaná z master do slave.
- MISO – data vysílaná ze slave zpět do master.
- CS – výběrový signál konkrétního slave (často aktivní v logické 0).

Master zařízení (typicky mikrořadič) řídí komunikaci, generuje hodinové impulsy a aktivuje SS příslušného podřízeného zařízení. SPI je koncipováno pro jednoho mastera a libovolný počet slave zařízení – master má pro každého slave samostatný výběrový pin SS. Data mohou proudit obousměrně současně (Full-Duplex) – zatímco master posílá na MOSI bit, slave může současně poslat bit na MISO. Díky hardwarovému posuvnému registru a synchronizaci hodin dosahuje SPI velmi vysokých rychlostí (běžně desítky MHz, některé čipy až stovky MHz). Oproti další oblíbené sériové sběrnici I²C (Inter-Integrated Circuit) má SPI výhodu vyšší rychlosti a full-duplex přenosu, za cenu většího počtu vodičů a absence standardizovaného adresování (každý slave má samostatný CS vodič) [17, s. 41–48].

V systému diplomové práce je SPI použito pro komunikaci mikrořadiče s externí flash pamětí (viz kapitola 4.3.1). V rámci aplikace SPI slouží i k logování dat do paměti – GNSS data lze ukládat do externí flash paměti pro pozdější vyzvednutí. Výhodou SPI je, že komunikace je řízená procesorem, takže ukládání do paměti může probíhat deterministicky a rychle, aniž by zatěžovalo CPU (Central Processing Unit) rozsáhlými přerušeními (jako by tomu bylo u asynchronního příjmu přes UART).

SPI tedy představuje stěžejní technologii pro rychlý sběr a ukládání dat v embedded zařízení – v tomto případě umožňuje mikrořadiči ukládat naměřené údaje do flash paměti a případně je odtud opět rychle číst při odesílání na server.

4.3 Technologie sběru a ukládání dat

Podkapitola se věnuje technologiím, které slouží k dlouhodobému ukládání a následnému zpracování dat získaných ze sledování vozidel. Popisuje vlastnosti paměti typu NOR flash pro lokální ukládání dat a MS SQL databáze používané pro dlouhodobé a strukturované uchování informací na serverové straně.

4.3.1 NOR flash paměť (výhody a omezení)

Pro ukládání dat přímo v zařízení (například záznam tras GNSS, než budou odeslány na server) je vhodná flash paměť, která uchovává data i po vypnutí napájení. Existují dva hlavní typy flash paměti – NOR a NAND flash, lišící se vnitřní architekturou a vlastnostmi. V embedded aplikacích se často používá NOR flash pro ukládání programu a konfigurace, zatímco NAND flash pro objemná data.

NOR flash vyniká schopností přímého adresovatelného přístupu po bajtech – procesor může mapovat NOR flash do paměťového prostoru a číst z ní data přímo, podobně jako z paměti ROM (Read-Only Memory). Tato vlastnost a rychlé náhodné čtení činí NOR flash ideální pro ukládání spouštěcího kódu a firmware zařízení. Další výhodou NOR je vysoká spolehlivost a dlouhodobá uchovatelnost dat – typicky uchová obsah i desítky let a na rozdíl od NAND trpí méně na vznik vadných bloků či samovolné bitové chyby. NOR flash také umožňuje zapisovat jednotlivé bajty – i když mazání probíhá po větších blocích, zápis může být flexibilnější než u NAND, která zapisuje po celých stránkách. Z hlediska integrity dat se u NOR méně často vyskytují chyby, což zjednodušuje její použití v kritických aplikacích.

Nevýhodou NOR flash jsou některá technická omezení. Zaprvé, výroba NOR buňky je plošně náročnější, což vede k vyšší ceně za bit a menší kapacitě čipů – NOR čipy mívají velikosti v jednotkách MB až desítkách MB, zatímco NAND flash je dostupná v řádech GB. Dále je zápis a mazání v NOR pomalejší; typicky mazání sektoru NOR (např. 4 kB blok) trvá milisekundy, a zápis bajtu desítky mikrosekund, což je pomalejší než u NAND, která píše stránku rychleji (ale zase musí zapisovat větší blok najednou). NOR flash také při zápisu/mazání spotřebovává více energie než NAND. Pro dané použití (datové logy GNSS) však typická velikost a rychlost NOR flash dostačuje – například 8 MB paměti pojme statisíce záznamů polohy a zápis několika kB za sekundu je možný [18, s. 99–132; 19, s. 639–660].

Shrneme-li výhody a omezení: NOR flash nabízí rychlé čtení a spolehlivé uchování dat (vhodné pro kód i citlivé informace), zatímco nevýhodou je vyšší cena a pomalejší zápis/mazání. V kontextu mikrořadiče je externí NOR flash nezbytností, neboť samotný mikrořadič nemá dostatečně velkou interní nevolatilní paměť. Navržený systém proto využívá externí NOR flash tak, že tato paměť je využita pro dočasné uložení nasbíraných dat (pokud zařízení ztratí spojení se serverem, může data logovat a později odeslat). NOR flash paměť tak představuje klíčovou technologii pro lokální sběr dat v zařízení – poskytuje nevolatilní úložiště s dostatečnou kapacitou a životností pro účely projektu.

4.3.2 Databáze MS SQL (vhodnost pro řešení, CTE dotazy, JSON)

Pro trvalé uložení většího objemu dat a jejich následnou analýzu či vizualizaci je vhodné použít robustní databázový systém na serverové straně. V rámci tohoto projektu bylo požadováno řešení postavené na MS SQL serveru – relační databázové platformě od firmy Microsoft. MS SQL je osvědčený enterprise databázový systém, který poskytuje vysokou spolehlivost, bezpečnost a výkonnost pro zpracování dat. Pro navrhovanou aplikaci (uložení historických tras, telemetrických dat atd.) je relační model vhodný, neboť data lze uložit do tabulek (tabulka záznamů polohy s atributy jako čas, souřadnice, rychlost) a pomocí jazyka SQL nad nimi provádět složité dotazy – filtrace, agregace, spojování s jinými údaji apod. MS SQL navíc dobře podporuje integraci s dalšími Microsoft technologiemi a v případě potřeby umožňuje škálovatelnost na velké objemy dat.

Pokud předpokládáme, že dat bude mnoho (řádově jednotky milionů záznamů ročně), MS SQL je schopna je efektivně spravovat. Poskytuje transakční záruky (ACID), takže lze spolehlivě zaznamenat všechny přijaté zprávy od zařízení. Díky podpoře pokročilých dotazů lze snadno vybrat například všechny body dráhy za určité období, vypočítat statistiky, nebo provádět složitější operace (např. detekce zastávek, výpočet vzdálenosti mezi body).

Výhodou jsou CTE (Common Table Expressions) a podpora JSON v MS SQL, které mohou v našem řešení sehrát roli. CTE je konstrukce T-SQL (Transact-SQL), jež umožňuje definovat dočasný pojmenovaný výsledek dotazu, který lze dále použít v následném SQL příkazu. CTE umožňuje rozdělit složitý dotaz na logické části, zvýšit jeho čitelnost nebo například provést rekurzivní dotaz. Příkladem může být výpočet délky trasy: pomocí CTE lze nejprve seřadit body trasy a spočítat dílčí vzdálenosti mezi po sobě jdoucími body, a poté vnější dotaz může tyto vzdálenosti sečíst. CTE výraz existuje pouze po dobu

vykonávání dotazu a nepředstavuje trvalou strukturu jako pohled. V našem případě jsou CTE užitečné při formulaci složitějších datových sestav nad trasami, aniž by bylo nutné vytvářet dočasné tabulky [20, s. 22–27, 62–67, 176–184].

Další moderní vlastností MS SQL je nativní podpora JSON dat (od verze MS SQL 2016). SQL Server umí pracovat s JSON textem – formátovat výsledek dotazu jako JSON. V praxi to znamená, že pokud například serverová část očekává data ve formátu JSON, lze je přímo získat z SQL dotazu. V našem projektu je JSON formát využit na rozhraní mezi serverem a webovou vizualizací, takže schopnost SQL serveru přímo generovat JSON záznamy je výhodná. Můžeme například jedním dotazem získat celou trasu ve formátu JSON pole bodů a přímo ji předat front-endu [21, s. 161–169].

Použití MS SQL tedy přináší robustní zázemí pro ukládání dat z mikrořadiče. Kombinuje tradiční sílu relačních databází (dotazovací jazyk SQL, integrita, transakce) s moderními rozšířeními jako CTE a JSON, které usnadní implementaci specifických požadavků této aplikace (složitější datové výběry, integrace s webovým rozhraním).

4.4 Softwarové technologie

Tato část představuje přehled softwarových nástrojů a technologií použitých při vývoji systému. Zabývá se vývojovým prostředím Arduino IDE (Integrated Development Environment) s využitím real-time operačního systému FreeRTOS (Real-Time Operating System) pro mikrořadiče, serverovým programovacím jazykem Python a moderními webovými technologiemi pro vizualizaci dat uživateli.

4.4.1 Arduino IDE a programování mikrořadiče (FreeRTOS)

Platforma Arduino představuje otevřený ekosystém pro jednoduché programování mikrořadičů, původně vyvinutý pro desky s procesory AVR (Atmel and Vegard's RISC). Arduino IDE je integrované vývojové prostředí, které umožňuje psát kód v zjednodušeném C/C++ (tzv. sketch), jedním kliknutím ho zkompileovat a nahrát do cílového zařízení. Arduino IDE klade důraz na uživatelskou přívětivost – poskytuje předpřipravené knihovny pro řadu periférií a hardware. Výhodou tohoto přístupu je rychlý prototypovací cyklus, obrovské množství existujících knihoven (komunikace Wi-Fi, zpracování protokolu z GNSS modulu, práce s JSON atd.) a velká komunita uživatelů, kteří řeší podobné problémy.

Současné mikrořadiče jsou tedy velmi schopné pro IoT aplikace – zvládnou sběr dat ze senzorů, jejich předzpracování a bezdrátový přenos. V našem projektu mikrořadič

zpracovává GNSS data a přes Wi-Fi je posílá serveru, případně reaguje na další povely. Díky vysoké integraci mikrořadiče nepotřebujeme mnoho externích obvodů, modul sám zajistí komunikaci i výpočty [22, s. 308–316].

Z pohledu software běží na mikrořadiči interně operační systém FreeRTOS. Jedná se o otevřený real-time operační systém pro mikrořadiče, který poskytuje jednoduchý plánovač úloh (Task Scheduler) a další OS (Operating System) služby – semaforey, fronty, časovače aj. Je navržen tak, aby usnadnil programování malých vestavných zařízení, zajišťuje spolehlivý běh více paralelních úloh a jejich plánování v reálném čase. FreeRTOS na mikrořadiči umožňuje efektivně využít možnosti čipu – například jednu úlohu nechat obsluhovat Wi-Fi komunikaci (knihovna Wi-Fi běží na pozadí v separátní úloze), druhou úlohu zpracovávat GNSS data, třetí ukládat data do flash atd. Plánovač se stará o přepínání kontextu. FreeRTOS je známý svou spolehlivostí s velmi malou režii (jádro má velikost v jednotkách kB a přepínání trvá desítky mikrosekund), je tedy ideální pro embedded zařízení s omezenými prostředky. Pro náš systém to znamená, že i na relativně malém mikrořadiči můžeme provádět více činností [23, s. 70–105].

Arduino IDE tedy zajišťuje pohodlný vývoj a nahrávání programu do mikrořadiče, zatímco FreeRTOS se stará o multitasking a real-time chod programu. Kombinace těchto technologií umožňuje rychle vyvíjet i poměrně komplexní embedded aplikace. Z hlediska teoretického základu je dobré zmínit, že FreeRTOS je de-facto standardem mezi RTOS pro mikrořadiče – pro jeho kvalitu a malou velikost. V našem projektu se tak opíráme o prověřenou softwarovou platformu: Arduino a FreeRTOS na straně zařízení, což zrychlí vývoj a zvýší spolehlivost výsledného systému.

4.4.2 Python jako serverová technologie

Na serverové straně systému bude běžet aplikace implementovaná v jazyce Python. Python je vysokoúrovňový skriptovací jazyk známý svou jednoduchou a čitelnou syntaxí a bohatým ekosystémem knihoven. Pro vývoj serverové logiky je Python velmi populární volbou – umožňuje rychle psát kód, který je snadno udržitelný a srozumitelný. Jeho syntaxe klade důraz na čitelnost a stručnost, takže danou funkcionalitu často vystihne v méně řádcích než ekvivalent v jiných jazycích.

Klíčovým důvodem pro použití Pythonu na serveru je rychlost vývoje a bohatá kolekce frameworků. Existuje množství ověřených webových frameworků, jako je Django, které značně usnadňují tvorbu webových aplikací a API. Django je plnohodnotný framework

umožňující rychlý vývoj udržitelných a zabezpečených webů, přičemž mnoho věcí řeší automaticky (ORM pro databázi, šablony pro struktury stránek, autentizaci uživatelů). Python má rozsáhlou komunitu a množství knihoven pro všechny myslitelné účely – od zpracování JSON, přes vědecké výpočty, komunikaci po síti až po specializované věci jako geodetické výpočty (např. výpočet vzdálenosti dle souřadnic). Pro náš projekt to například znamená, že implementace serveru zvládne s využitím existujících knihoven poměrně snadno: navázat spojení s HTTP (HyperText Transfer Protocol) server, přijmout data od mikrořadiče, uložit je do MS SQL a zároveň poskytovat webové rozhraní (např. vracet stránkám JSON s daty).

Python je interpretovaný jazyk, nicméně pro typické zatížení (řádově desítky zařízení odesílající data, jednotky uživatelů zobrazujících data) je jeho výkon dostatečný. Vývoj v Pythonu navíc urychluje ladění – program se nemusí kompilovat, změny lze aplikovat velmi rychle. Díky přenositelnosti Pythonu lze serverovou část provozovat na libovolné platformě (Linux/Windows), což je výhodné [24, s. 313–328].

Zmínku si zaslouží i to, že Python se často používá jako „lepidlo“ mezi systémy – je schopen snadno propojit různé komponenty. V našem případě může Python běžet na serveru, komunikovat s databází MS SQL, a současně obsluhovat HTTP rozhraní pro vizualizaci. Je tedy velmi univerzální. Jazyk jako takový je vhodný pro back-end vývoj i díky dalším vlastnostem: je vysoce čitelný (což zjednodušuje týmovou spolupráci) a má množství kvalitní dokumentace a výukových materiálů. Podle mnoha průzkumů patří Python mezi nejoblíbenější jazyky pro server-side programování vedle JavaScriptu (JS) a Javy, hlavně kvůli frameworkům jako Django a snadné integraci s front-endem (např. přes JSON API) [25, s. 205–213].

Python na serveru umožní rychlou implementaci požadovaných funkcionalit s minimálním množstvím kódu, přičemž zachová spolehlivost a bezpečnost (při správném použití frameworků). Pro diplomovou práci to znamená, že většinu logiky – přijímání dat, jejich ukládání, poskytování dalším částem systému – lze napsat efektivně v Pythonu. Díky rozsáhlým standardním knihovnám a komunitním balíčků má Python řešení pro všechny potřeby projektu, což minimalizuje nutnost „vynalézat kolo“ a umožní soustředit se na aplikační logiku.

4.4.3 Vizualizační technologie (HTML, JavaScript, JSON)

Pro prezentaci nasbíraných dat uživateli (typicky prostřednictvím webového prohlížeče) slouží kombinace technologií HTML (HyperText Markup Language), CSS (Cascading Style Sheets) a JavaScript na straně klienta. HTML je značkovací jazyk definující strukturu webové stránky – umožňuje popsat, jaké informace a v jakém členění stránka obsahuje (nadpisy, odstavce, tabulky, formuláře apod.). CSS doplňuje HTML o možnosti stylování – nastavuje barvy, rozvržení, grafický design stránek. JavaScript je skriptovací jazyk spuštěný v prohlížeči, který umožňuje stránku oživit dynamickým chováním – reagovat na akce uživatele, měnit obsah stránky za běhu, načítat data ze serveru na pozadí (AJAX) atd. Tyto tři složky tvoří základ každé interaktivní webové aplikace. V našem systému budou použity k vytvoření webových stránek, které zobrazí on-line polohy vozidel na mapě nebo data v reportech [26, s. 15–27, 62–80].

Webová aplikace v moderním pojetí typicky funguje tak, že server posílá klientovi HTML/CSS/JS kód (případně generovaný šablonami), který se vykoná v prohlížeči. Struktura stránky (HTML + CSS) může být statická, a data se do ní doplňují dynamicky pomocí JavaScriptu. Například stránka může obsahovat prázdnou tabulku nebo mapový komponent, a JavaScript po načtení stránky zavolá na server API (např. HTTP GET request na určitou URL) a obdrží data (typicky ve formátu JSON). JSON je lehký datový formát pro výměnu dat, velmi rozšířený právě ve webových aplikacích pro svou jednoduchost a čitelnost. Jedná se v podstatě o textový zápis objektů s klíč-hodnota a poli, který je přímo v syntaxi JavaScriptu (prohlížeč jej snadno zpracuje). JSON je dnes jedním z nejčastěji používaných formátů pro přenos dat mezi webovými službami a aplikacemi, vytlačil starší XML (eXtensible Markup Language) v mnoha oblastech. Pro naše účely – předávání dat o polohách, které jsou strukturované (např. pole objektů s atributy čas, poloha) – je JSON ideální. Python server může snadno serializovat data do JSON a poslat, JavaScript na stránce je přijme a zobrazí.

Konkrétní vizualizace může mít podobu například tabulky hodnot, grafu či vykreslení trasy v mapě. HTML s CSS zajistí, že stránka bude přehledná a použitelná, JavaScript se pak stará o interakci a aktualizaci dat. Příkladem použití JavaScriptu je tzv. AJAX volání – skript na stránce se v pravidelných intervalech dotazuje serveru na nové údaje a ty pak doplní do stránky bez nutnosti jejího znovunačtení. To umožní téměř real-time sledování pohybu zařízení na mapě.

Výhodou využití webových technologií pro vizualizaci je univerzální dostupnost – uživatel potřebuje jen webový prohlížeč. Technologie HTML, JS či CSS jsou standardizované konsorciem W3C a podporované ve všech moderních prohlížečích. JavaScript navíc umožňuje využít řadu front-endových knihoven a API – tvorbu interaktivních map (Google Maps, Leaflet) atd. Pro tento projekt ale postačí čistý JavaScript s DOM (Document Object Model) manipulací. Důležité je, že datové rozhraní mezi serverem a klientem bude založeno na JSON – server poskytne data v JSON a klient je pomocí JavaScriptu zpracuje a vloží do HTML. Tím se oddělí vzhled od dat.

Shrneme-li, vizualizační vrstva systému stojí na standardním webovém základu: HTML pro obsah, CSS pro vzhled, JavaScript pro interaktivitu a načítání dat. JSON pak funguje jako výměnný formát, který hladce propojí server s klientem. Tato kombinace je dnes de facto standardem pro webové aplikace a umožní snadno vytvořit uživatelsky přívětivé rozhraní pro prezentaci výsledků práce systému (např. polohy zobrazit na mapě, statistiky v reportech). Výhodou je také přizpůsobení (Responsive Design) – pomocí HTML/CSS lze zajistit, že stránka bude použitelná i na mobilních zařízeních, takže uživatel může data sledovat z jakéhokoli zařízení [27, s. 161–169].

4.4.4 Mapové API – alternativa či rozšíření

K vizualizaci geodat (poloh, tras) lze také využít mapových API. Jedná se o služby, které vývojářům poskytují hotové mapové podklady a nástroje pro zobrazení mapy v aplikaci, vyhledávání bodů, výpočet tras apod.

Příkladem je velmi rozšířené Google Maps, dále alternativy jako Mapbox nebo open-source řešení nad OpenStreetMap daty (např. knihovna Leaflet). Zapojení mapového API do projektu by umožnilo vykreslit trasu pohybu zařízení na interaktivní mapě – uživatel by tak viděl polohu přímo v kontextu silnic, terénu [28, s. 1–4].

Pro náš účel by mapové API bylo spíše nadstavbovým rozšířením. Z pohledu teoretického je vhodné uvést, že existují volně dostupné mapové služby – OpenStreetMap poskytuje otevřená geografická data, která lze využít bez licenčních poplatků, byť bez záruk a s nutností vlastní implementace některých funkcí.

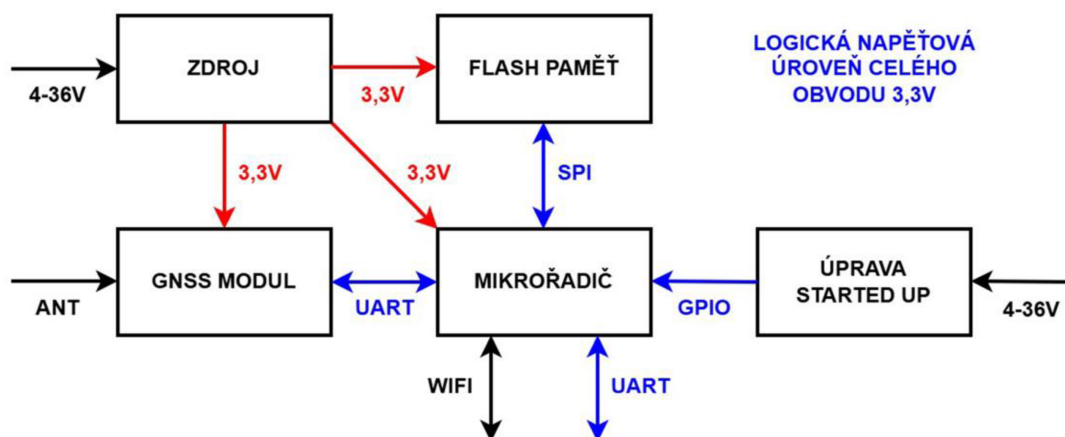
II PRAKTICKÁ ČÁST

5 NÁVRH SYSTÉMU

Tato kapitola navazuje na teoretický základ technologií (kapitola 4) a představuje návrh systému monitorování vozidel v průmyslovém areálu pomocí GNSS. Cílem je navrhnout celkovou architekturu řešení – definovat hlavní komponenty systému a jejich vzájemná propojení. Kapitola slouží jako mezičlánek mezi teorií a konkrétní implementací (kapitola 6) a obsahuje obecný návrh hardwarové, softwarové a serverové části.

5.1 Návrh hardwarové části

Hardwarová část představuje palubní jednotku, která bude instalována ve vozidle. Tato podkapitola popisuje zvolené hlavní komponenty a jejich zapojení. Jedná se o mikrořadič, GNSS modul, napájecí zdroj a další podpůrný hardware (obrázek č. 5). Cílem hardwarového návrhu je vytvořit spolehlivý a funkční základ zařízení s ohledem na provozní požadavky (např. napájení z palubní sítě vozidla v rozsahu 4–36 V a úsporný provoz se schopností uspání).



Obrázek 5 Blokové schéma hardware palubní jednotky

(zdroj: vlastní)

5.1.1 Schéma zapojení ESP32-C6 a U-blox NEO-M9N, zdroje, podpůrný HW

Hlavními komponenty palubní jednotky jsou:

- **Mikrořadič ESP32-C6**

Slouží jako řídicí jednotka celého zařízení. Jde o výkonný 32 bitový mikrořadič s integrovaným Wi-Fi modulem (2,4 GHz) a dalšími bezdrátovými rozhraními, což umožňuje přímé připojení k síti bez nutnosti externího modemu.

ESP32-C6 zpracovává data z GNSS modulu, řídí komunikaci po síti a obstarává logiku pro úsporu energie či ukládání dat.

- **GNSS modul U-blox NEO-M9N**

Zajišťuje příjem signálů satelitních navigačních systémů a výpočet polohy vozidla. Tento modul podporuje vícenásobné GNSS systémy (GPS, GLONASS, Galileo aj.) pro zvýšení přesnosti a spolehlivosti určení polohy. K ESP32-C6 je připojen pomocí rozhraní UART pro přenos dat o poloze v reálném čase. GNSS modul vyžaduje připojení externí antény, umístěné na vhodném místě vozidla (s dobrým výhledem na oblohu) pro kvalitní příjem signálu.

- **Externí paměť typu NOR Flash**

Pro případ výpadku Wi-Fi konektivity je součástí jednotky také nevolatilní paměť, kam může mikrořadič dočasně ukládat naměřené polohové údaje. Tato paměť NOR flash je připojena k ESP32-C6 prostřednictvím rozhraní SPI a umožňuje bezpečné uložení tisíců záznamů o poloze, které budou po obnovení spojení odeslány na server. Záloha dat do flash paměti zajišťuje, že nedojde ke ztrátě údajů o pohybu vozidla ani při dočasné nedostupnosti sítě, serveru či databáze.

- **Podpůrný hardware**

Návrh obsahuje i další obvody nezbytné pro stabilní provoz. Patří sem napájecí zdroj (viz kapitola 5.1.2) převádějící palubní napětí vozidla na 3,3 V pro napájení všech komponent a základní obvody pro reset a programování mikrořadiče (rozhraní UART vyvedené na konektor pro nahrání firmware). Sériové rozhraní slouží nejen k programování ESP32-C6, ale i k případné konfiguraci GNSS modulu nebo diagnostice zařízení. Dále jsou zahrnuty ochranné a filtrační prvky na vstupu napájení pro zabezpečení před přepětím a rušením z palubní sítě. Celkový koncept zapojení tedy propojuje všechny uvedené bloky: GNSS modul předává údaje mikrořadiči, který je zpracuje a přes Wi-Fi odešle na server; napájecí a podpůrné obvody zajišťují spolehlivý a bezpečný chod zařízení.

5.1.2 Napájení (převod 12 V → 3,3 V)

Zařízení musí být schopno pracovat při napájení z palubní sítě vozidla v širokém rozpětí napětí (cca 4–36 V). Proto je navrženo napájení založené na spínaném měniči (DC/DC), který ze vstupního napětí (např. nominálních 12 V či 24 V) vytvoří stabilních 3,3 V

potřebných pro všechny komponenty. Použití spínaného měniče zajišťuje vysokou účinnost (minimální tepelné ztráty) oproti lineárnímu stabilizátoru, což je důležité vzhledem k trvalému provozu zařízení a omezenému prostoru pro odvod tepla.

Napájecí vstup je chráněn proti vlivům automobilového prostředí s ochrannou pojistkou, shottkyho a transil diodou pro eliminaci přepětových špiček či zkratu v palubní síti. Na vstupu i výstupu měniče jsou zařazeny filtrační kondenzátory k vyhlazení napětových pulsů a omezení rušení. Samotný DC/DC měnič je dimenzován tak, aby spolehlivě pracoval v celém požadovaném rozsahu vstupního napětí a dodal dostatečný proud i pro špičky při vysílání Wi-Fi. Stabilita a nízký šum výstupních 3,3 V je klíčová pro správnou funkci citlivého GNSS přijímače, proto je kladen důraz na kvalitní filtraci. Navržené řešení napájení umožňuje připojit jednotku přímo k palubní síti vozidla a zajišťuje bezpečné a efektivní zásobování energií za různých podmínek.

5.1.3 GPIO řízení zapínání a uspání zařízení

V návrhu je implementována možnost přechodu zařízení do nízko příkonového režimu (uspání) a následného probuzení, aby se šetřila energie baterie, pokud nebude v provozu alternátor vozidla. Mikrořadič ESP32-C6 disponuje režimem hlubokého spánku (Deep Sleep), během něhož jeho spotřeba klesá na jednotky mikroampér. Návrh počítá s tím, že mikrořadič ESP32-C6 bude moci uspávat nejen sám sebe, ale také GNSS modul.

K identifikaci, zda má jednotka přejít do režimu spánku slouží GPIO (General Purpose Input/Output) pin mikrořadiče, který sleduje stav nastartování vozidla. Na základě stavu tohoto vstupního pinu dochází k zapnutí či vypnutí modulů. Pokud není vozidlo nastartované tak nemusí jednotka aktivně sledovat polohu, mikrořadič vyšle signál k uspání GNSS modulu a tento přejde do hlubokého spánku. Následně sám mikrořadič taktéž přejde do hlubokého spánku.

Tímto způsobem GPIO logika zajišťuje inteligentní řízení napájení. mikrořadič rozhoduje, kdy zařízení uvést do provozu a kdy jej bezpečně uspat, čímž optimalizuje spotřebu a minimalizuje zátěž baterie.

5.2 Návrh softwarové části

Softwarová část systému pro mikrořadič je navržena tak, aby zajišťovala spolehlivé získávání, ukládání a přenos dat o poloze. Softwarovou architekturu lze rozdělit na několik hlavních logických částí:

- **Komunikace s GNSS**

Zodpovídá za inicializaci a průběžné čtení dat z GNSS modulu. Při startu zařízení nastaví GNSS přijímač patřičné konfigurační parametry a následně přijímá binární zprávy UBX z modulu U-blox NEO-M9N přes UART. Tento modul parsuje získaná data (ze zpráv extrahuje např. zeměpisnou šířku, délku, rychlost, čas) a předává je dále do systému ve vhodné datové struktuře.

- **Správy dat a úložiště**

Přijátá data o poloze jsou předzpracovávána a odesílána na server k dalšímu zpracování. Pokud z jakéhokoliv důvodu nedojde k odeslání dat jsou tato ukládána do vlastní NOR flash paměti. Modul udržuje frontu záznamů čekajících na odeslání a po obnovení spojení se serverem zajistí postupné odeslání všech neodeslaných údajů, aby nedošlo ke ztrátě žádných informací.

- **Komunikace po síti**

Zajišťuje připojení zařízení k místní Wi-Fi síti a řídí odesílání dat na server. Po spuštění se přihlásí k definované Wi-Fi a naváže spojení se serverem (HTTP spojení k API). Tento modul odesílá aktuální polohové údaje v pravidelných intervalech, tzn. provádí HTTP POST požadavky s daty ve formátu JSON. Je implementována kontrola potvrzení přijetí dat; pokud server nepotvrdí příjem, záznam zůstane uložen a zařízení se pokusí o opětovné odeslání později. Modul také průběžně sleduje stav spojení a v případě výpadku přepne systém do off-line režimu (dá pokyn modulu úložiště k ukládání do flash paměti). Po obnovení spojení naopak přejde do on-line režimu a iniciuje odeslání všech dříve neodeslaných záznamů.

- **Řízení provozu a napájení**

Tento blok dohlíží na celkový chod zařízení a rozhoduje o přechodech mezi aktivním režimem a spánkem. Na základě informací od externích podnětů (stav zapalování přes GPIO) vyhodnocuje, zda má zařízení přejít do úsporného režimu. Před uspaním zajistí uložení všech dosud neodeslaných dat do paměti flash, které ještě nebyly

odeslány na server (např. není dostupné spojení) a uspání GNSS modulu. Následně nastaví vlastní probuzení a také přejde do režimu spánku. Po probuzení obnoví činnost ostatních modulů (opět zapne GNSS, naváže Wi-Fi spojení atd.). Tento modul tedy funguje jako centrální řídicí logika, která integruje informace ze všech částí systému a udržuje zařízení v optimálním stavu z hlediska dostupnosti dat a spotřeby energie.

5.3 Návrh serverové části

Serverová část představuje systém, které přijímá a zpracovává data zaslaná z jednotek ve vozidlech a poskytuje je koncovým uživatelům ve srozumitelné podobě. Popisuje obecnou architekturu serverové části a definuje hlavní komponenty, jako je přijímací server/API, databázové úložiště a webová aplikace pro vizualizaci dat. Cílem je navrhnout, jak bude server zajišťovat spolehlivé uchování polohových údajů a jejich dostupnost pro dohled a analýzu pohybu vozidel.

Hlavní komponenty serverové části jsou:

- **Serverové rozhraní pro příjem dat (API)**

Server poskytuje rozhraní, které přijímá příchozí zprávy od všech monitorovacích jednotek. Typicky jde o HTTP server přijímající POST požadavky, kam každá jednotka posílá svůj identifikátor a naměřené hodnoty (čas, souřadnice, rychlost atd.). Serverové API provede validaci dat (formátu, rozsahu hodnot, oprávnění zařízení) a poté údaje uloží do databáze. Rozhraní je navrženo pro paralelní obsluhu více zařízení a dostatečně robustní, aby zvládlo nárazově větší počet zpráv (např. po výpadku sítě při opětovném připojení jednotek).

- **Databázové úložiště**

Slouží k trvalému uchování všech přijatých údajů o poloze. Návrh počítá s použitím vhodné relační databáze, která umožní efektivně ukládat časové řady dat – typicky tabulku obsahující ID vozidla, časový údaj, zeměpisnou šířku, délku a případně další parametry. Databáze umožňuje efektivní dotazy nad historickými daty i zjištění poslední známé polohy každého vozidla.

- **Webová aplikace pro vizualizaci**

Na serveru běží také aplikační logika pro prezentaci dat uživatelům. Jde o webovou aplikaci, která z databáze získává aktuální i historické údaje a zobrazuje je na mapovém podkladu. Uživatel (např. dispečer) může v prohlížeči vidět seznam sledovaných vozidel a jejich poslední známé pozice vyznačené na mapě a další údaje jako zdaje vozidlo nastartované, jak dlouho je bez pohybu. Vizualizace polohy probíhá v reálném čase – server průběžně aktualizuje zobrazené údaje tak, aby poloha vozidel na mapě odpovídala nejnovějším dostupným datům. Aplikace je přístupná v rámci interní sítě areálu (případně i vzdáleně při zabezpečeném přístupu) a může vyžadovat přihlášení uživatele pro ochranu citlivých informací. Návrh serverové části tak zajišťuje, že data z vozidel jsou bezpečně uložena v databázi a kdykoli dostupná k monitorování či zpětné analýze.

6 IMPLEMENTACE SYSTÉMU

Kapitola implementace systému se zaměřuje na detailní popis praktické realizace navrženého řešení monitorovacího systému vozidel. Popsány jsou konkrétní komponenty, jejich propojení, a rovněž kompletní software zařízení, serverová část včetně databáze, a uživatelské rozhraní. Výsledkem této implementace je funkční prototyp systému, který splňuje veškeré požadavky definované v analytické a návrhové části práce.

6.1 Implementace hardwaru

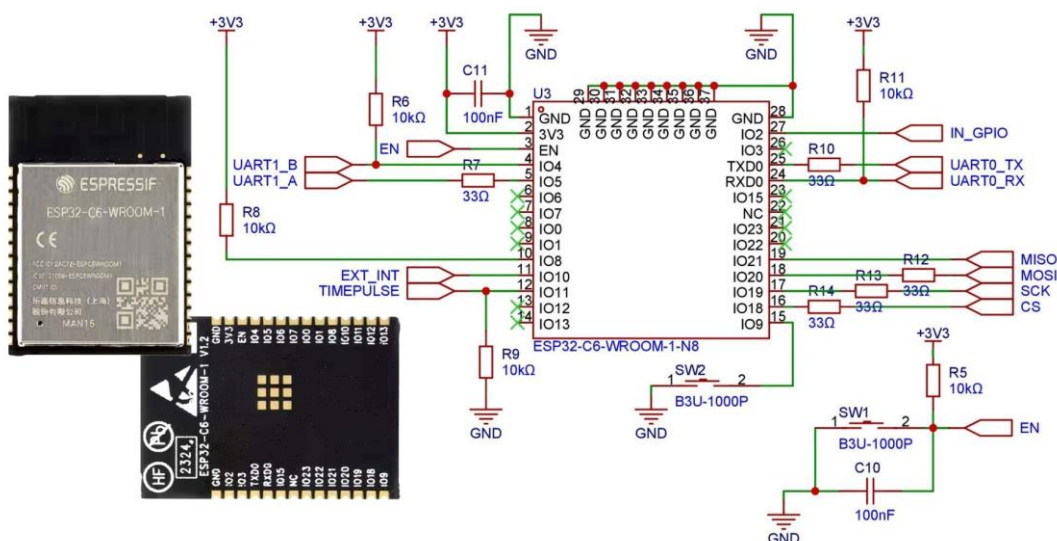
Implementace hardwaru zahrnuje sestavení vestavěného zařízení pro sledování vozidel a jeho napájecí a komunikační infrastruktury. Systém je postaven na moderním mikrořadiči s integrovaným Wi-Fi a využívá specializovaný GNSS modul pro určování polohy. Navržený hardware zajišťuje spolehlivé získávání polohy vozidla, ukládání dat při výpadku spojení a bezdrátový přenos dat na server.

6.1.1 Konkrétní zapojení a komponenty

Hlavními komponenty vestavěného zařízení jsou následující části:

- **Mikrořadič ESP32-C6-WROOM-N8**

Jedná se o modul mikrořadiče s čipem ESP32-C6, který integruje výkonný mikroprocesor (RISC-V) s 2,4GHz bezdrátovým rozhraním (Wi-Fi 6 a BLE) a podporou protokolu 802.15.4 (Zigbee/Thread). Modul disponuje 8 MB interní flash paměti pro program a několika hardwarovými UART rozhraními. ESP32-C6 zajišťuje řídicí logiku celého systému – zpracovává data z GNSS modulu, řídí komunikaci po Wi-Fi a spravuje ukládání dat do externí paměti (obrázek č. 6). Jedno z UART rozhraní je využito pro příjem dat z GNSS modulu a druhé UART slouží pro vlastní programování či ladicí výstup. ESP32 rovněž poskytuje rozhraní SPI pro komunikaci s externí NOR flash pamětí.

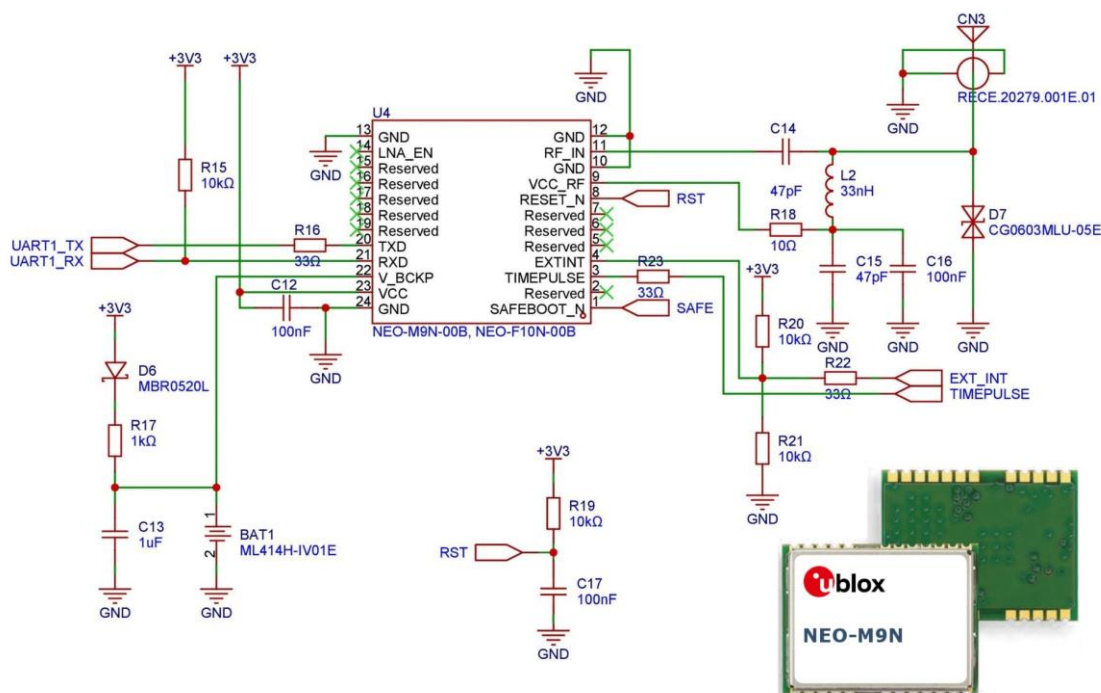


Obrázek 6 Zapojení mikrořadiče ESP32-C6

(zdroj: vlastní)

- **GNSS modul U-blox NEO-M9N-00B**

Pro určování polohy byl zvolen GNSS modul NEO-M9N od firmy U-blox. Tento modul umožňuje současný příjem signálů až ze čtyř družicových navigačních systémů (GPS, GLONASS, Galileo, BeiDou), což zajišťuje rychlou a přesnou lokalizaci i v členitém prostředí průmyslového areálu. Modul podporuje výstup dat prostřednictvím binárního protokolu UBX, který poskytuje podrobné informace o poloze, rychlosti a čase. GNSS modul je k ESP32 připojen přes sériové rozhraní UART (konkrétně UART1 na ESP32, zapojení na obrázku č. 7). Pro komunikaci je nastaven odpovídající baud rate 38 400 Bd pro rychlejší přenos dat. Datové výstupy modulu (TX) jsou připojeny na přijímací pin UART ESP32, a naopak vysílací pin ESP32 je připojen na vstup (RX) GNSS modulu pro posílání konfiguračních příkazů. GNSS modul je napájen napětím 3,3 V a vyžaduje kvalitní napájení kvůli citlivosti RF (Radio Frequency) části – napájení je proto filtrováno kondenzátory poblíž modulu. K modulu je připojena externí aktivní GNSS anténa umístěná tak, aby měla co nejlepší výhled na oblohu.

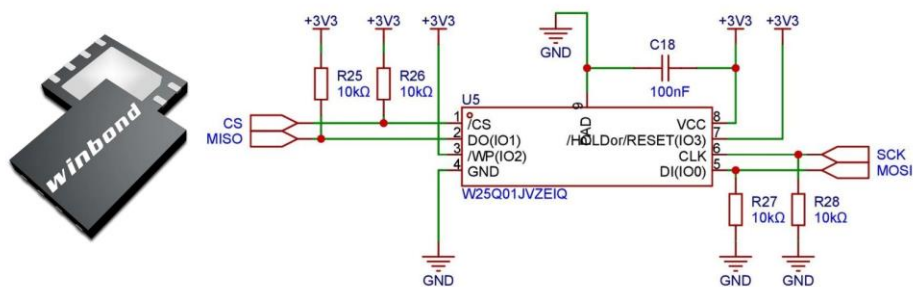


Obrázek 7 Zapojení GNSS modulu NEO-M9N

(zdroj: vlastní)

- Externí paměť NOR flash W25Q256JVEIQ**

Pro ukládání naměřených dat v případě nedostupnosti serveru nebo Wi-Fi připojení je integrována externí flash paměť typu NOR o kapacitě 256 Mbit (32 MB), konkrétně čip W25Q256JVEIQ. Tato paměť je připojena k ESP32 pomocí rozhraní SPI (obrázek č. 8). Slouží jako úložiště záznamů polohy ve formě kruhového bufferu.



Obrázek 8 Paměť NOR flash

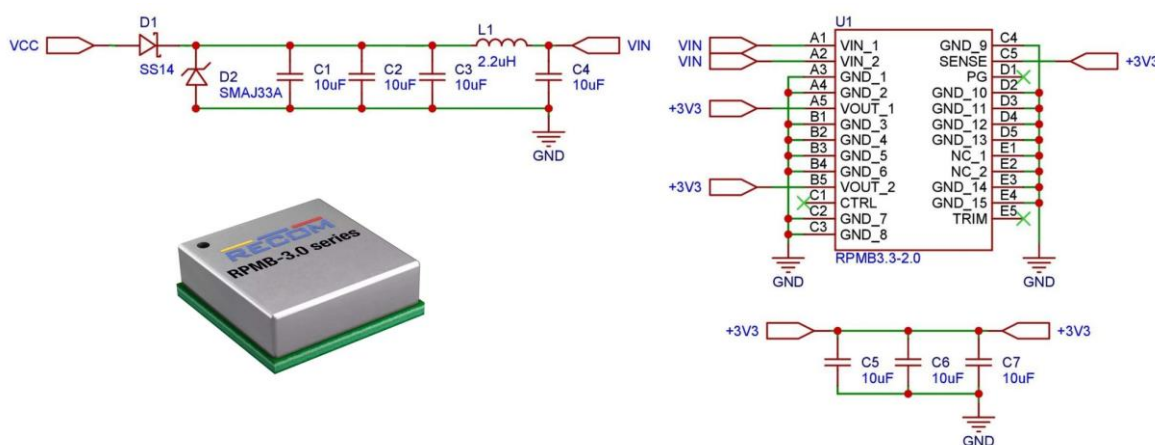
(zdroj: vlastní)

Mikrořadič ESP32 do této paměti zapisuje nové záznamy (64 bajtů na jeden záznam) a tím je zajištěno, že zařízení může dočasně uchovávat polohová data i během delší ztráty spojení, aniž by došlo k přerušení sběru dat. NOR flash paměť má organizaci

do sektorů (typicky 4 kB na sektor) a vyžaduje mazání po sektorech, což je zohledněno v návrhu kruhového bufferu (viz část 6.2.2).

- **Napájecí modul RECOM RPMB3.3-2.0**

Celé zařízení je navrženo pro napájení z elektrické soustavy vozidla. Pro převod napětí vozidla (typicky v průmyslových vozidlech 12 V nebo 24 V) na stabilních 3,3 V potřebných pro elektroniku je použit DC/DC měnič RPMB3.3-2.0 (schéma zapojení na obrázku č. 9). Tento modul zvládá vstupní napětí v rozsahu cca 4–36 V DC a poskytuje výstup 3,3 V s proudovou kapacitou až 2 A, což je dostatečné pro napájení ESP32, GNSS modulu i dalších komponent. Výhodou tohoto modulu je vysoká účinnost a integrované ochrany (proti zkratu, přepětí apod.), které zvyšují odolnost zařízení v automobilovém prostředí. Napájecí modul je zapojen tak, že jeho výstup 3,3 V napájí přímo ESP32 modul, GNSS modul i externí flash paměť (všechny hlavní komponenty využívají napětí 3,3 V).



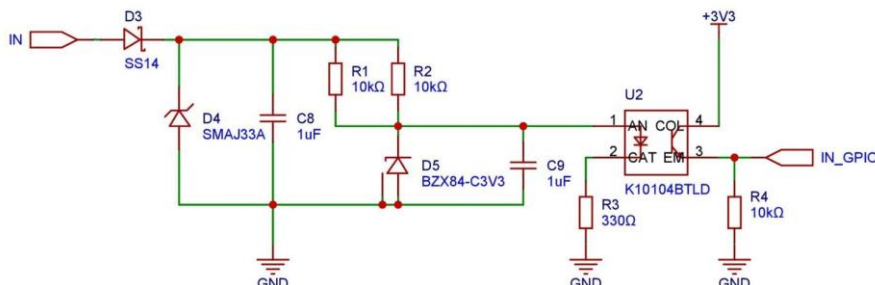
Obrázek 9 Zdroj napájení

(zdroj: vlastní)

- **Řízení zapínání přes GPIO**

Zařízení disponuje funkcí řízení běžného a úsporného režimu pomocí vstupního GPIO pinu mikrořadiče. K tomu je použita forma úpravy vstupního napětí přicházejícího ze spínací skříňky vozidla formou děliče napětí upravované dále zenerovou diodou. Takové napětí je přivedeno na optočlen, který již zajišťuje patričnou napětovou úroveň přiváděnou na vstupní pin mikrořadiče (obrázek č. 10). Princip spočívá v tom, že po přivedení napájení po nastartování vozidla (zapnutí klíčku zapalování) se zařízení probudí a mikrořadič ESP32 probudí a nastaví také

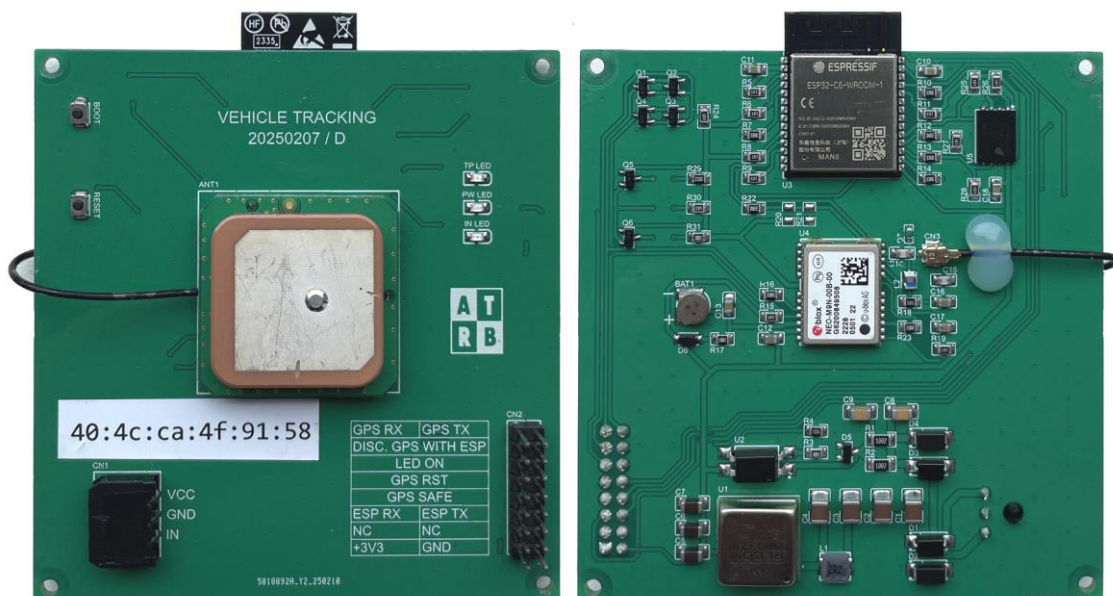
ostatní komponenty. Tato funkce je užitečná pro úsporu energie – zařízení může samo sebe uspat, pokud se vozidlo nepoužívá, a probudit se na základě vnějšího podnětu (zapnutí zapalování).



Obrázek 10 Zapínání systému přes GPIO

(zdroj: vlastní)

Kromě výše zmíněných hlavních komponent obsahuje hardware také drobné podpurné prvky: například několik LED diod indikujících stav systému (např. přítomnost napájení, stav GPS fixu, aktivitu zapalování), ochranné prvky proti přepětí a rušení na vstupu napájení a potřebné konektory pro programování a monitorování při ladění (např. konektor pro USB-UART převodník propojený s druhým UARTem ESP32 pro výstup odesílaných zpráv). Kompletní schéma GNSS modulu palubní jednotky uvedeno v příloze č. 1 a 2.



Obrázek 11 Výsledný GNSS modul palubní jednotky

(zdroj: vlastní)

Celkově hardware (obrázek č. 11) zajišťuje robustní provoz v průmyslovém prostředí, je schopen pracovat v rozšířeném teplotním rozsahu, odolává rušení od motorů a strojů (díky filtraci a kvalitnímu napájení) a poskytuje spolehlivou platformu pro běh sledovacího systému.

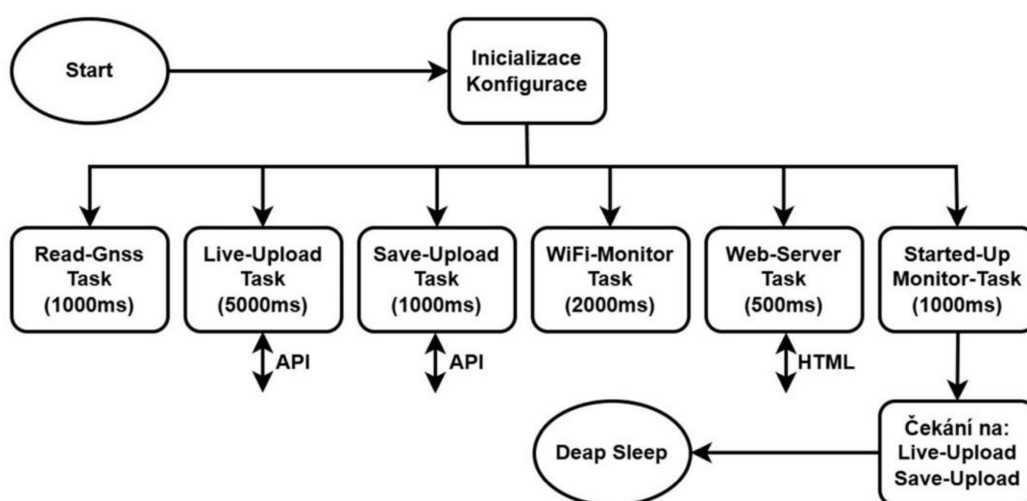
6.2 Implementace softwaru zařízení

Vestavěný software běží na mikrořadiči ESP32 v reálném čase pod operačním systémem FreeRTOS. Pro správné fungování je potřeba k základnímu Arduino IDE rozhraní instalovat i rozšíření pro ESP32 a další externí knihovny. Ucelený přehled všech potřebných softwarových částí v příloze č. 3 a dále adresářová struktura v příloze č. 4.

Hlavním úkolem software je průběžně získávat polohu z GNSS modulu, zajišťovat bezdrátovou komunikaci se serverem a ukládat data do externí paměti při výpadku spojení. Vzhledem k povaze aplikace byl zvolen vícevláknový přístup – jednotlivé funkce (čtení polohových údajů, odesílání dat, správa paměti) jsou odděleny do samostatných úloh FreeRTOS, které běží paralelně a vzájemně spolupracují. Tato architektura zvyšuje spolehlivost a plynulost provozu všech operací.

6.2.1 Úlohy FreeRTOS pro čtení polohových dat, Wi-Fi komunikaci a další

V rámci software běžícího na mikrořadiči ESP32-C6 je použito několik paralelně běžících úloh založených na operačním systému FreeRTOS (obrázek č. 12).



Obrázek 12 Blokové schéma software palubní jednotky

(zdroj: vlastní)

Každá z těchto úloh má specifickou funkci, předem danou periodu spuštění a vzájemnou koordinaci prostřednictvím sdílených struktur:

- **Inicializační úloha (Setup Task)**

Tato úloha se spouští po startu zařízení a provádí veškerou potřebnou inicializaci – nastavení periferií (UART, SPI, Wi-Fi), konfiguraci GNSS modulu pomocí binárního protokolu UBX, přípravu paměti a spuštění všech následujících úloh. Po provedení inicializace se sama ukončí.

- **Read GNSS Task**

Tato úloha je periodicky spouštěna každých 1 000 ms a zajišťuje čtení dat z GNSS modulu U-blox NEO-M9N pomocí UART a výhradně binárního protokolu UBX. Získaná data (např. souřadnice, rychlost, čas) jsou ukládána do sdílené struktury dostupné pro další úlohy.

- **Live Upload Task**

Každých 5 000 ms kontroluje dostupnost nových GNSS dat. Pokud je zařízení připojeno k Wi-Fi, odešle data na server ve formátu JSON. V případě, že není k dispozici připojení, uloží data do NOR flash paměti prostřednictvím SPI.

- **Save Upload Task**

V intervalu 1 000 ms porovnává ukazatele flash paměti na přítomnost neodeslaných dat. Pokud je dostupné Wi-Fi připojení, odesílá data zpětně na server a po úspěšném přenosu daný záznam z flash paměti odstraní.

- **Wi-Fi Monitor Task**

Každých 1 000 ms sleduje sílu a dostupnost Wi-Fi připojení. Pokud dojde ke ztrátě připojení nebo slabému signálu, automaticky inicializuje přepojení na jiný dostupný přístupový bod (stejně SSID, různé kanály – roaming).

- **Web Server Task**

V intervalu 500 ms obsluhuje požadavky na jednoduché lokální webové rozhraní, které poskytuje základní informace o systému – stav GNSS, připojení k Wi-Fi, počet záznamů flash paměti a diagnostické logy.

- **StartedUp Monitor Task**

Sleduje stav „zapalování“ prostřednictvím GPIO pinu. Každých 1 000 ms kontroluje změnu tohoto stavu. Pokud dojde k poklesu napětí (vypnutí motoru), aktivuje se proces ukončení všech úloh a přechod systému do hlubokého spánku. V tabulce č. 2 je porovnání spotřeby pro různé pracovní režimy.

Tabulka 2 Spotřeba celého obvodu – naměřené hodnoty

Celý obvod	Běžný provoz	Maximální spotřeba	Deep Sleep
Napájecí napětí 12 V	29–34 mA	41–45 mA	< 2 mA

(zdroj: vlastní)

Aby bylo zajištěno bezpečné sdílení dat mezi paralelně běžícími úlohami, využívá systém synchronizační mechanismy operačního systému FreeRTOS. Přístup ke sdíleným strukturám, jako jsou GNSS data nebo ukazatele pozice v NOR flash paměti, je řízen prostřednictvím binárních a mutexových semaforů. Tyto synchronizační prvky brání tomu, aby více úloh současně přistupovalo ke stejnému prostředku, čímž se předchází vzniku stavu souběhu (Race Condition) a poškození dat. Semaforey slouží také jako prostředek signalizace dokončení operací mezi úlohami a zajišťují konzistentní tok informací napříč celým systémem.

6.2.2 Ukládání dat do NOR flash (management dat při nedostupnosti Wi-Fi)

V situacích, kdy zařízení nemá k dispozici Wi-Fi připojení k serveru, je nezbytné uchovávat naměřená data lokálně, aby nedošlo k jejich ztrátě. K tomu slouží externí NOR flash paměť, využívaná jako kruhový buffer pro sekvenční ukládání polohových záznamů. Tato sekce detailně popisuje, jak je ukládání implementováno.

Každý záznam polohy je před uložením do paměti serializován do binární podoby s pevnou délkou. Použití pevné délky zjednodušuje implementaci kruhového bufferu, neboť není třeba řešit proměnlivou délku zpráv. V našem případě obsahuje jeden záznam *RecordSet* hodnoty uvedené v tabulce č. 3.

Celková délka jednoho záznamu v tomto binárním formátu je 45 bajtů (data) respektive po zaokrouhlení 64 bajtů (tj. 4 záznamy do 1 stránky a 16 stránek do 1 sektoru ~ 4 kB). Tím pádem 32 MB flash pojme 524 288 záznamů (což při frekvenci 0,2 Hz představuje cca 72 plných dnů kontinuálního ukládání). I při vyšší frekvenci nebo rozšíření dat je rezerva obrovská. Jelikož v původním testovacím konceptu se data ukládala jako textová s délkou

- Zapisuje se vždy po celých stránkách kvůli efektivitě.
- Když se zaplní stránka záznamy, přejde se na další stránku. Když se dojde na konec posledního sektoru, je třeba začít od začátku – před tím se však musí smazat staré sektory, aby data bylo kam zapisovat.
- Mazání probíhá po celých 4 kB sektorech. Pokud tedy jsou data z celého sektoru již odeslána (označena, že byla odeslána) je celý sektor vymazán. Jelikož NOR flash nemůže měnit bity z 0 na 1 nelze tedy data jen tak přepsat, ale právě použít funkci mazání. Tato operace je relativně pomalá (desítky milisekund) oproti samotnému zápisu dat jedné stránky.

Jak je výše uvedeno, tak každé takové vymazání sektoru snižuje jeho životnost, a proto je nutné zatěžovat celou NOR flash rovnoměrně. Toto zajistí právě kruhový buffer. Pro zajištění správné činnosti takového bufferu jsou udržovány celkem tři ukazatele (indexy):

- **Write pointer**

Jde o ukazatel ve flash paměti, kam se запиše příští nový záznam. Při každém zápisu záznamu se posune ukazatel o 1 vpřed. Pokud dosáhne konce paměti, cyklicky se vrátí na začátek paměti (0).

- **Read pointer**

Ukazatel nejstaršího neodeslaného záznamu, který čeká na zpracování (odeslání na server). Tento ukazatel se posouvá vpřed teprve ve chvíli, kdy byl daný záznam úspěšně přenesen na server a již není potřeba uchovávat ho lokálně. Zpočátku (po spuštění nebo vyprázdnění bufferu) se Write pointer = Read pointeru.

- **Null pointer**

Tento ukazatel indikuje vždy začátek 4 kB sektoru který ještě nebyl vymazán, a tudíž do něj nelze vstoupit pro zápis záznamů. Z důvodu inicializace (nalezení ukazatelů při spuštění) je vždy udržován jeden celý takový 4 kB sektor nevymazaný, aby bylo možné najít všechny tyto tři ukazatele pro dodržení koncepce kruhového bufferu. Tyto ukazatele nelze na žádné stejné místo ukládat, neboť oblasti, které by k tomuto účelu sloužily byly nadměrně zatěžovány neustálým přepisováním. Došlo by tedy k opotřebení těchto oblastí dříve než ostatních částí, což by časem

vedlo k selhání dané paměťové části (podobně jako u klasické NAND flash – FAT tabulka).

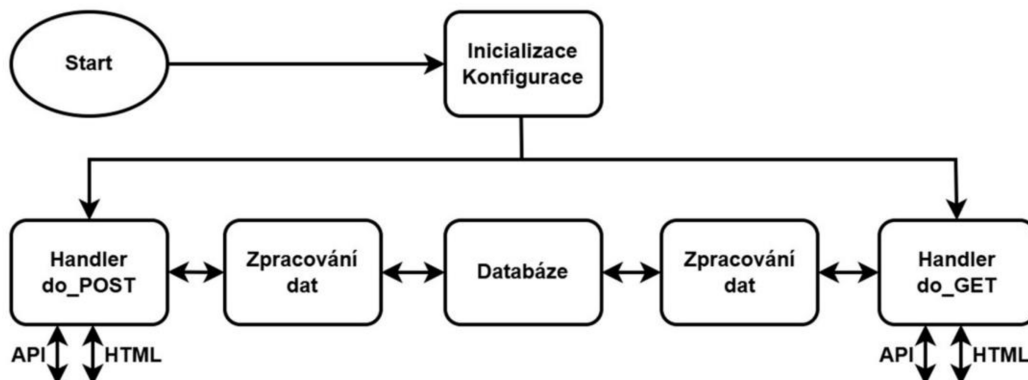
Díky tomuto systému je zajištěno, že při nedostupnosti Wi-Fi nedojde ke ztrátě dat. Data jsou uložena efektivně v binární formě, která minimalizuje objem. Jakmile se zařízení opět připojí k síti, data se odešlou a paměť se recykluje pro další použití. Toto řešení kombinuje výhody trvalého úložiště a fronty, i v případě úplného vypnutí zařízení (např. když vozidlo zcela ztratí napájení) zůstanou již zapsané údaje ve flash paměti a po opětovném zapnutí mohou být odeslány na server.

6.3 Implementace serverové části

Serverová část systému zajišťuje příjem dat ze zařízení, jejich bezpečné uložení do databáze a poskytnutí rozhraní pro vizualizaci a analýzu dat uživateli. Server byl implementován jako více vláknová aplikace v jazyce Python, běžící na vyčleněném serveru v rámci podniku. Výběr programovacího jazyka Python umožnil rychlý vývoj s využitím bohatých knihoven pro webové služby i databázové spojení. Přehled všech softwarových částí a knihoven je uveden v příloze č. 5, adresářová struktura potom v příloze č. 6.

6.3.1 Struktura Python aplikace

Serverová aplikace je strukturována tak, aby bylo dosaženo modularity, přehlednosti a efektivity při zpracování dat z GNSS modulů na vozidlech. Hlavním řídicím prvkem aplikace je skript *main.py*, který slouží jako vstupní bod a koordinátor všech částí systému. Po spuštění inicializuje prostředí, načte konfigurace a spouští hlavní vlákna pro obsluhu klientů a komunikaci s databází (obrázek č. 13).



Obrázek 13 Blokové schéma software serverové části

(zdroj: vlastní)

V aplikaci jsou využita paralelně běžící vlákna (Threading), která umožňují simultánní zpracování více požadavků. Tato vlákna obsluhují především HTTP GET a POST požadavky přicházející od GNSS zařízení, klientských aplikací či dalších částí systému. Požadavky jsou směřovány na specializované funkce umístěných v modulu *http_handlers.py*, které zpracují příchozí data a spustí potřebné akce, jako validaci dat, ukládání do databáze nebo generování HTML odpovědí.

Moduly jako *database.py* jsou odpovědné za přístup k databázi, vytváření spojení, provádění dotazů a manipulaci se záznamy. Modul *common.py* definuje globálně sdílené proměnné a konfigurace, které jsou dostupné napříč aplikací, čímž zajišťuje konzistentní prostředí. Šablony pro generování HTML výstupů a SQL dotazů s podporou společných tabulkových výrazů CTE jsou organizovány v modulech *html_template.py* a *cte_template.py*. Tyto šablony umožňují flexibilní generování strukturovaných dotazů a dynamických webových stránek dle požadavků klientů.

Všechny tyto komponenty společně tvoří kompaktní celek, který je schopen efektivně reagovat na příchozí požadavky, zajišťovat integritu dat a optimalizovat komunikaci s databází.

6.3.2 Formát dat, komunikace JSON

Komunikace mezi palubními jednotkami a serverem probíhá formou HTTP POST požadavků, přičemž data jsou serializována ve formátu JSON. Každý JSON paket obsahuje detailní informace o stavu a poloze vozidla, včetně unikátní identifikace zařízení, časové značky měření, geografických souřadnic, stavu zařízení (např. zda je vozidlo v pohybu či stojí), hodnot přesnosti měření a dalších relevantních informací potřebných pro přesnou analýzu.

Po přijetí dat server nejprve provádí jejich validaci a kontroluje, zda jsou všechna povinná pole vyplněna, zda odpovídají očekávanému typu a zda hodnoty splňují definovaná kritéria. Pokud jsou data validní, jsou následně uložena do databáze a případně okamžitě předávána klientským aplikacím ve formě JSON odpovědi. Tento formát umožňuje snadnou integraci s webovým rozhraním, dalšími aplikacemi či analytickými nástroji.

Odezvy ze serveru, rovněž informují palubní jednotku o úspěchu či případných chybách při zpracování požadavků, což napomáhá při diagnostice a řešení problémů v reálném čase.

6.3.3 Implementace databáze (struktura, dotazy CTE, JSON výstupy)

Základem aplikace je relační databáze MS SQL, která je optimalizována pro rychlé ukládání a dotazování velkého množství GNSS záznamů. Každý přijatý záznam je uložen do strukturovaných tabulek, které obsahují klíčové informace jako ID zařízení, čas měření, polohu (zeměpisnou šířku a délku), přesnost měření a stav zařízení.

Pro pokročilé analýzy a reporty aplikace využívá společné tabulkové výrazy CTE, které usnadňují tvorbu komplexních dotazů, například výpočtů celkově ujeté vzdálenosti, doby trvání jízd nebo doby stání vozidel. Výsledky těchto dotazů jsou následně serializovány do formátu JSON a poskytovány klientům jako podklady pro vizualizaci či další analytické zpracování.

Díky této struktuře je možné efektivně vyhodnocovat provozní údaje flotily vozidel, provádět optimalizaci logistických procesů a získávat detailní přehled o aktivitě vozidel v areálu.

6.4 Implementace uživatelského rozhraní

Uživatelské rozhraní systému je realizováno jako webová aplikace, která umožňuje interaktivní zobrazení polohy vozidel na mapě a zobrazení sestav (reportů) o jejich pohybu. Rozhraní je přístupné přes webový prohlížeč z intranetové sítě podniku, takže oprávnění uživatelé (např. dispečeri, správci areálu) mohou sledovat aktuální situaci a prohlížet historická data bez nutnosti instalovat speciální software. Implementace UI (User Interface) kombinuje technologie HTML, CSS a JavaScript na straně klienta a využívá dat poskytovaných serverem přes HTTP (viz předchozí část).

6.4.1 Dynamická vizualizace polohy vozidel na mapě

Klientské webové rozhraní aplikace poskytuje uživatelům živý pohled na aktuální polohu všech sledovaných vozidel. Vizualizace probíhá na mapových podkladech pomocí HTML a JavaScriptu. JavaScriptová komponenta se každých 5 sekund dotazuje serveru formou HTTP GET požadavku, načež server vrací aktuální polohy ve formátu JSON (výřez na obrázku č. 14).



Obrázek 14 Živé zobrazení polohy vozidel – výřez

(zdroj: vlastní, celý pohled v příloze č. 7)

Tato data jsou dynamicky vykreslována na interaktivní mapě, přičemž uživatel okamžitě vidí změny polohy vozidel, jejich aktuální stav a čas poslední aktualizace. Díky pravidelné aktualizaci dat a okamžitému zpracování je zajištěno efektivní monitorování pohybu vozidel v reálném čase (respektive s minimálním zpožděním).

6.4.2 Vizualizace reportů (HTML, JavaScript, tabulky)

Reportovací modul klientského rozhraní umožňuje uživatelům generovat přehledy o činnosti vozidel na základě jejich individuálních potřeb. Prostřednictvím webového formuláře uživatel definuje parametry výběru dat (vozidla, časové období apod.). Příklad sumarizačního reportu za vybraná vozidla na obrázku č. 15.

Vehicles Trips Summary

Month: 2025-05, Time Range: 00:00:00 - 23:59:59

Vehicles: ■ Expedice (20), ■ Expedice (35), ■ Expedice (36), ■ Expedice (26), ■ Údržba (18), ■ Výroba (12)

Vehicle Name	Count	Engine Time	Idle Time	Off Time	Distance [m (km)]
Σ	2150	244:05:02	77:57:34	1511:03:12	1260962 (1261.0)
■ Expedice (20)	588	26:49:23	9:02:35	213:10:27	143055 (143.1)
■ Expedice (35)	217	54:54:08	16:56:04	185:05:42	293962 (294.0)
■ Expedice (36)	196	42:01:55	12:55:53	173:57:56	221417 (221.4)
■ Expedice (26)	100	46:34:31	19:42:53	193:25:19	186305 (186.3)
■ Údržba (18)	306	28:31:49	9:36:15	355:27:55	131360 (131.4)
■ Výroba (12)	743	45:13:16	9:43:54	389:55:53	284863 (284.9)

Obrázek 15 Sumarizační report za jednotlivá vozidla

(zdroj: vlastní, další reporty v příloze č. 8 a 9)

Na základě těchto parametrů server generuje specializované SQL dotazy s využitím CTE, výsledky v JSON formátu pak převádí a vrací zpět klientovi. Klientské rozhraní tyto výsledky zpracovává a prezentuje buď ve formě přehledných HTML tabulek, nebo jiných grafických výstupů, které vizuálně znázorňují různé metriky, například ujetou vzdálenost, dobu provozu, dobu nečinnosti či četnost jízd.

7 INTEGRACE DO INFRASTRUKTURY SPOLEČNOSTI

Tato kapitola se věnuje způsobu, jakým byl navržený systém monitorování vozidel pomocí GNSS technologií začleněn do stávající podnikové infrastruktury. Cílem je zajistit, aby řešení bylo nejen technicky funkční, ale i bezpečně integrovatelné, snadno spravovatelné a odpovídalo standardům používaným v daném prostředí. Popsána je síťová infrastruktura, aplikační a databázová vrstva, jakož i opatření zajišťující bezpečnost a spolehlivost celého systému.

7.1 Popis stávající infrastruktury

Podniková infrastruktura, do které byl nový systém GNSS monitorování integrován, je komplexní a dobře udržovaná. V rámci firemní sítě je nasazena řada přístupových bodů systému UniFi od společnosti Ubiquiti, které zajišťují pokrytí celého areálu bezdrátovým Wi-Fi signálem v pásmu 2,4 GHz. Tato jednotlivá AP zařízení jsou centrálně spravována a umožňují flexibilní konfiguraci, včetně definice více SSID a řízení přístupových práv podle bezpečnostních požadavků.

Serverová část infrastruktury zahrnuje aplikační server s běžící instancí frameworku Django, který je provozován pod webovým serverem IIS. Tento server je součástí doménově spravovaného prostředí a integruje se do firemní autentizační vrstvy, která využívá doménové přihlášení uživatelů. Databázová vrstva je řešena pomocí MS SQL serveru, který běží jako samostatná služba ve stejné síti a slouží k ukládání aplikačních i analytických dat.

Uživatelé přistupují k jednotlivým modulům systému prostřednictvím intranetu. Ti, kteří jsou doménově ověřeni, mají přístup k modulům na základě přidělených oprávnění. Nedoménoví uživatelé musí být ověřeni pomocí AD. Přístup z veřejného internetu je obecně zakázán, avšak vybraným uživatelům může být umožněn prostřednictvím bezpečné VPN, kterou přiděluje IT oddělení. Tím je zajištěna jak dostupnost, tak i ochrana systému před neoprávněným přístupem.

7.2 Integrace systému do firemní sítě (AP UniFi, kanály 2.4 GHz)

Za účelem oddělení komunikační vrstvy GNSS zařízení od ostatního provozu v síti byla na vybraných přístupových bodech vytvořena nová bezdrátová síť s unikátním SSID. Toto SSID je dostupné pouze v rámci určené množiny AP, které pokrývají oblasti pohybu sledovaných vozidel, a je dedikováno výhradně pro palubní GNSS jednotky.

Zařízení se k síti připojují automaticky a pokud to je možné tak využívají roaming mezi přístupovými body bez přerušení datové komunikace. Tím je zajištěn plynulý přenos dat i při pohybu zařízení mezi jednotlivými AP, které pokrývají všechny části areálu. Komunikace mezi zařízeními a serverem probíhá výhradně prostřednictvím Wi-Fi v rámci této izolované sítě.

Přístupové body i SSID jsou centrálně spravovány prostřednictvím existujícího UniFi Controlleru, což zajišťuje jednotné řízení konfigurací, zabezpečení a přístupových oprávnění. Díky této architektuře je možné síť GNSS systémů snadno spravovat, rozšiřovat nebo modifikovat bez zásahu do zbytku infrastruktury.

7.3 Konfigurace serveru a databáze

Serverová část GNSS systému byla nasazena na stejném fyzickém serveru, na kterém již běží aplikační vrstva frameworku Django. Server je spuštěn jako samostatná služba, která obsluhuje požadavky přicházející od palubních jednotek (HTTP POST a GET). Zpracování dat probíhá přímo v této serverové aplikaci, která po validaci údajů zapisuje data do databáze MS SQL.

Tabulka 4 Struktura tabulek databáze MS SQL

table common_vehicles		table gnss_locations_records	
vehicle_id	tinyint IDENTITY(1,1)	location_record_id	int IDENTITY(1,1)
vehicle_name	nvarchar(30)	vehicle_id	tinyint
mac_address	nvarchar(17)	location_at_utc	datetime2(0)
ip_address	nvarchar(15)	latitude	decimal(9, 6)
hostname	nvarchar(15)	longitude	decimal(9, 6)
reg_number	nvarchar(15)	course	decimal(5, 2)
color	nvarchar(15)	speed	decimal(5, 2)
last_start_at	datetime	stationary_time	smallint
last_sync_at	datetime	is_started_up	bit
is_reboot	bit	is_live	bit
is_active	bit	location_valid	bit
current_values	nvarchar(max)	location_at_utc_valid	bit
fuel_type	tinyint	course_valid	bit
		counter	smallint

(zdroj: vlastní)

Databáze obsahuje dvě hlavní tabulky navržené pro tento účel. Tabulka *common_vehicles* slouží k evidenci vozidel a uchovává informace jako název, identifikátor, MAC a IP adresa zařízení, hostname a další stavové příznaky. Tabulka *gnss_locations_records* slouží k uchování jednotlivých polohových záznamů – včetně časové značky, polohových souřadnic, rychlosti, směru jízdy a dalších parametrů

souvisejících se stavem monitorovaného zařízení. Datová struktura obou databázových tabulek je popsána v tabulce č. 4.

Aplikační logika serveru GNSS je přímo napojena na tyto databázové struktury a zajišťuje jak ukládání dat, tak jejich následné zpracování. Díky tomu není nutné použití žádného externího API a veškerá komunikace probíhá výhradně v rámci vnitřní sítě. Pro generování výstupů a reportů je využito interního dotazování přes CTE a export dat do formátu JSON, který je následně interpretován klientským rozhraním.

7.4 Zajištění bezpečnosti (izolace, bezpečný přístup k datům)

Bezpečnost řešení je zajištěna především tím, že veškerý provoz a data zůstávají uvnitř podnikové infrastruktury. GNSS zařízení komunikují výhradně po oddělené Wi-Fi síti unikátního SSID, která je izolována od zbytku provozní sítě. Veškeré datové toky směřují pouze na definovaný server, který je pod plnou správou podnikového IT oddělení.

Autentizace a oprávnění k přístupu ke klientským modulům jsou spravovány prostřednictvím již existující Django aplikace. Uživatelé ověření doménovým účtem získávají oprávnění automaticky dle přiřazených rolí. Nedoménoví uživatelé, pokud jim má být přístup umožněn, jsou autorizováni pomocí AD.

Přístup k aplikaci z veřejné sítě je možný pouze přes zabezpečený VPN (Virtual Private Network) tunel, který je individuálně přidělován IT oddělením. Tím je zajištěno, že pouze autorizovaní uživatelé s přístupem k firemní VPN mohou komunikovat se serverem. Veškeré přenosy mezi zařízeními a serverem jsou monitorovány a systém je chráněn před neoprávněným přístupem jak na síťové vrstvě, tak na úrovni aplikace.

Součástí zabezpečení je také pravidelná údržba systému, aktualizace aplikačních komponent a zálohování databáze. Tato opatření jsou v souladu s bezpečnostní politikou společnosti a zajišťují stabilitu a dlouhodobou udržitelnost řešení.

8 TESTOVÁNÍ A OVĚŘENÍ FUNKČNOSTI SYSTÉMU

Kapitola se zaměřuje na testování a ověření funkčnosti vyvinutého systému monitorování vozidel v průmyslovém areálu. Nejprve je popsána metodika testování a definovány různé testovací scénáře simulující možné výpadky a nestandardní situace. Následně jsou představeny výsledky funkčních testů systému v oblastech sběru dat, serverové a databázové vrstvy a uživatelského rozhraní. Závěrem kapitoly jsou vyhodnoceny získané poznatky z testování, zhodnoceno splnění stanovených požadavků a navržena případná vylepšení.

8.1 Metodika testování (v reálných podmínkách)

V této části je popsán postup a kritéria použitá při testování systému. Testování probíhalo podle definovaných scénářů, které simulují klíčové provozní situace a možné poruchy. Cílem metodiky bylo ověřit odolnost systému vůči výpadkům a jeho chování v různých podmínkách reálného provozu.

8.1.1 Testovací scénáře (vnitřní/venkovní prostředí, výpadky Wi-Fi)

Bylo definováno několik testovacích scénářů pokrývajících kritické situace, se kterými se systém může setkat:

- **Výpadek napájení zařízení**

Test simuloval náhlé odpojení napájení palubní jednotky a její následné restartování. Cílem bylo ověřit, že po obnovení napájení nedojde ke ztrátě uložených dat a systém pokračuje v normálním provozu.

- **Výpadek připojení Wi-Fi**

Test simuloval dočasnou ztrátu Wi-Fi konektivity, během níž nemohla palubní jednotka odesílat data na server. Zkoumalo se, zda jednotka v době výpadku správně ukládá polohová data do interní paměti NOR flash a po obnovení spojení je následně všechny odešle na server.

- **Výpadek signálu GNSS**

Test ověřoval chování systému při ztrátě signálu z družic (např. vjezdem vozidla do haly). Pozorovalo se, jak zařízení reaguje při dočasné nedostupnosti polohových dat a zda správně obnoví sledování polohy po opětovném zachycení signálu.

- **Roaming mezi AP**

Test zjišťoval, jak se zařízení chová při přechodu mezi různými AP sítě Wi-Fi. Cílem bylo ověřit, že krátká ztráta spojení při změně sítě neovlivní kontinuální sběr dat a jejich následný přenos.

- **Nedostupnost serveru či databáze**

Simulovala se situace, kdy je serverová aplikace nebo databáze dočasně nedostupná. Sledovalo se, zda zařízení dokáže dočasně ukládat data lokálně a po opětovném spuštění serveru je automaticky odeslat, aniž by došlo ke ztrátě či poškození dat při synchronizaci.

8.2 Funkční testy systému

V rámci funkčních testů byl systém prověřen v reálném provozu podle výše uvedených scénářů. Tato část shrnuje výsledky testování jednotlivých komponent systému, a to od sběru a ukládání dat na palubní jednotce, přes zpracování a ukládání dat na serveru, až po odezvu a použitelnost uživatelského rozhraní.

8.2.1 Testování sběru dat (GNSS, NOR flash, Wi-Fi)

Testy sběru dat ověřovaly schopnost palubní jednotky spolehlivě získávat a ukládat polohové informace z GNSS modulu. Simulovaly se výpadky spojení, aby se ověřilo ukládání dat do NOR flash paměti během přerušené komunikace a následné obnovení komunikace s postupným přenosem uložených dat. Výsledky potvrdily, že jednotka dokáže kontinuálně zaznamenávat GNSS data v nastaveném intervalu a při přerušení spojení je bezpečně ukládá do nevolatilní paměti. Po obnovení spojení zařízení odeslalo postupně všechny nahromaděné záznamy na server bez ztráty dat a bez narušení posloupnosti časových značek. Dále byl ověřen správný formát a obsah přenášených dat (souřadnice, časové údaje). Po uložení v databázi se zaznamenané hodnoty shodovaly s očekávanými.

8.2.2 Testování serverové a databázové vrstvy

Testy serverové a databázové vrstvy se zaměřily na výkonnost a stabilitu systému při zpracování většího objemu dat. Bylo provedeno zátěžové testování simulující příjem velkého množství polohových zpráv v krátkém čase, zejména při přenosu dat po obnovení spojení od všech palubních jednotek. Sledovala se rychlost zpracování příchozích dat, vytížení serveru a odezva databáze při ukládání dat a provádění dotazů.

Výsledky ukázaly, že serverová aplikace dokáže i při nadměrném zatížení přijmout a uložit data beze ztrát. Kontrola integrity dat potvrdila, že nedošlo k žádnému poškození ani duplicitě záznamů v databázi (toto je kontrolováno klíčem tabulky), a to ani při nečekaném přerušení spojení během zápisu. Rovněž byly testovány typické databázové dotazy (např. vyhledání historie polohy vozidla za určité období) a ověřeno, že systém na ně reaguje v přiměřeném čase a poskytuje správné výsledky.

8.2.3 Testování uživatelského rozhraní

Testování uživatelského rozhraní probíhalo na různých zařízeních a v různých síťových prostředích, aby se ověřila univerzálnost a dostupnost systému pro koncové uživatele. Byla vyzkoušena funkčnost webové aplikace na stolním počítači, notebooku i mobilním zařízení a ověřeno, že rozhraní se správně zobrazuje a ovládá na různých velikostech obrazovek. Dále byl testován přístup k systému z vnitřní sítě podniku i z vnějších sítí prostřednictvím internetu.

Vnější přístup byl realizován přes zabezpečené VPN spojení, čímž se ověřila funkčnost systému i pro vzdálené uživatele při zachování bezpečnosti. Testy uživatelských účtů a přístupových práv potvrdily, že systém správně uplatňuje omezení dle rolí uživatelů (běžný uživatel vs. administrátor) a nepovoluje neautorizované operace.

V rámci testů výkonnosti bylo zjištěno, že aplikace se načítá rychle a polohy vozidel se na mapě aktualizují téměř v reálném čase. Odezva uživatelského rozhraní tak byla vyhodnocena jako dostatečně plynulá pro pohodlné používání.

8.3 Vyhodnocení výsledků testování

V závěrečné části testování jsou shrnuty výsledky vzhledem ke stanoveným cílům a požadavkům. Přehled klíčových testovaných oblastí a výsledků je uveden v tabulce č. 5. V dalších bodech je posouzeno, do jaké míry systém splňuje všechny funkční i nefunkční požadavky definované v úvodních kapitolách. Současně jsou identifikovány oblasti, v nichž se projeví nedostatky, a navržena opatření pro jejich odstranění nebo zlepšení.

Tabulka 5 Přehled klíčových testovaných oblastí a výsledků

Testovaná oblast	Stav	Poznámka
Sběr dat a ukládání v zařízení	Splněno	Zařízení spolehlivě zaznamenává GNSS údaje; při výpadku spojení ukládá data do paměti a po obnovení je automaticky odešle.
Wi-Fi konektivita (výpadky, roaming)	Převážně splněno	Krátkodobé výpadky jsou pokryty ukládáním dat, přechod mezi AP funguje automaticky. V okrajových zónách areálu byl však zaznamenán slabý signál.
Serverový příjem a zpracování dat	Splněno	Server aplikační vrstvy zvládá průběžný příjem i hromadné dávky dat po výpadku, veškeré výpočty nad daty byly ověřeny pomocí kontrolních dotazů a jsou správné.
Databázová vrstva (výkon, integrita)	Splněno	Data se ukládají konzistentně a bez ztráty; při testovaném zatížení nedošlo k výrazné degradaci výkonu. Dlouhodobě je vhodné sledovat růst objemu dat.
Uživatelské rozhraní	Splněno	Rozhraní je funkční na různých zařízeních a prohlížečích; uživatelská práva jsou správně nastavena. Odezva systému byla rychlá, pouze přes VPN mírně pomalejší (stále přijatelná).
Celkové splnění funkčních požadavků	Splněno	Systém naplnil všechny hlavní definované požadavky. Menší nedostatky se projevíly jen ve vedlejších oblastech (Wi-Fi pokrytí, škálovatelnost při extrémní zátěži), které nebrání nasazení.

(zdroj: vlastní)

8.3.1 Splnění požadavků

Na základě provedených testů lze konstatovat, že systém většinu stanovených požadavků splňuje. Všechny klíčové funkce definované v zadání práce byly ověřeny. Systém dokáže v reálném čase sledovat polohu vozidel (jen s dopravním zpožděním), bezpečně ukládat a přenášet data a poskytovat je uživatelům prostřednictvím přehledného rozhraní. Dosažený výkon a spolehlivost odpovídají očekáváním. Nedošlo k žádné ztrátě dat při výpadcích spojení a odezva aplikace zůstala v akceptovatelných mezích i při zátěži. Menší odchylky od ideálního stavu byly zaznamenány pouze v dílčích oblastech, zejména při přechodu mezi buňkami sítě, které však nebrání celkové funkčnosti systému, neboť výpadky spojení bezproblémově řeší ukládání dat do flash paměti.

8.3.2 Identifikace možných problémů a návrhy na zlepšení

Testování odhalilo několik oblastí, kde je prostor pro zlepšení. Například byla zaznamenána drobná prodleva při prvním navázání spojení po výpadku a v některých případech snížená přesnost určení polohy v zastavěných částech areálu. Tyto nedostatky sice nemají zásadní vliv na základní funkčnost systému, ale jejich odstranění by dále zvýšilo spolehlivost a uživatelský komfort.

Mezi navrhovaná vylepšení patří optimalizace algoritmu obnovy spojení pro rychlejší synchronizaci dat po výpadku což je spojeno s optimálním rozmístěním přístupových bodů sítě a jejich konfigurací. V neposlední řadě lze zvážit úpravy uživatelského rozhraní na základě zpětné vazby od dispečerů. Tyto návrhy vycházejí z poznatků získaných během testování a jejich realizace by mohla dále zvýšit celkovou přidanou hodnotu vyvinutého řešení.

ZÁVĚR

Diplomová práce se zaměřila na návrh, realizaci a nasazení systému pro monitorování vozidel v průmyslovém areálu s využitím GNSS. Všechny definované cíle a požadavky uvedené v zadání byly splněny a výsledné řešení je plně funkční a již nasazeno do ostrého provozu. Systém v praxi spolehlivě monitoruje polohu vozidel v reálném čase a poskytuje efektivní nástroj pro dohled nad vozidly v rámci areálu.

Navržené řešení se vyznačuje architektonickou nezávislostí a necloudovým charakterem – nevyužívá žádnou cloudovou infrastrukturu a může být plně provozováno v rámci lokální podnikové infrastruktury. Tím je zajištěna plná kontrola nad daty i integrací systému a zároveň nezávislost na externích službách či dodavatelích. Dalším významným rysem je modulární struktura systému a možnost jeho rozšíření. Jednotlivé komponenty, jako je modul pro sběr dat z vozidel, databázové úložiště či uživatelské rozhraní, jsou navrženy odděleně a lze je v budoucnu upravovat nebo doplňovat bez nutnosti zásahů do ostatních částí. Díky této modularitě je systém připraven na snadné přizpůsobení novým požadavkům nebo rozšíření své funkčnosti podle potřeb provozu.

Maximální důraz byl při vývoji kladen na robustnost, stabilitu a přizpůsobení systému provozním podmínkám. Řešení bylo proto důkladně testováno v reálném prostředí průmyslového areálu a prokázalo stabilní chod. Systém dokáže spolehlivě pracovat i v náročných situacích, například při krátkodobých výpadech nebo rušení signálu GNSS, a celkově odpovídá požadavkům nepřetržitého provozu. Neméně zásadní je skutečnost, že návrh, vývoj a implementace vycházeli z vlastních požadavků, nikoliv z přizpůsobování se stávajícím produktům. Tento přístup umožnil vytvořit řešení přesně na míru specifickým potřebám uživatele a vyhnout se kompromisům vyplývajícím z omezení hotových systémů. Celkově byly naplněny všechny cíle práce. Výsledkem je ucelený, plně funkční systém, který splňuje veškeré stanovené požadavky a zároveň poskytuje stabilní základ pro další rozvoj. Již v současnosti probíhá implementace dalších funkcí, například monitoringu spotřeby paliva a souvisejících emisí CO₂, společně s rozšířenými grafickými výstupy a reporty, které dále zvýší užitnou hodnotu celého systému.

SEZNAM POUŽITÉ LITERATURY

- [1] O2 Czech Republic a.s. O2 Car Control – GPS monitoring vozidel [online]. Praha: O2 Czech Republic, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.o2.cz/podnikatel-a-firma/car-control/>
- [2] T-MOBILE CZECH REPUBLIC a.s. T-Mobile Autopark – systém pro správu vozového parku [online]. Praha: T-Mobile Czech Republic, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.t-mobile.cz/autopark/>
- [3] VODAFONE CZECH REPUBLIC a.s. Vozový park – Webdispečink [online]. Praha: Vodafone Czech Republic, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.vodafone.cz/business/sluzby/webdispecink/>
- [4] SHERLOG TECHNOLOGY a.s. Sherlog Trace – GPS monitoring vozidel [online]. Praha: Sherlog Technology, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.sherlog.cz/produkty-a-sluzby/monitorovani-vozidel/>
- [5] LINDE MATERIAL HANDLING ČESKÁ REPUBLIKA s.r.o. Fleet Management Linde connect [online]. Praha: Linde Material Handling Česká republika, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.linde-mh.cz/cs/Reseni-a-sluzby/Flotila/>
- [6] TLV s.r.o. GPS Dozor – systém GPS monitoringu vozidel [online]. Praha: TLV, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.gpsdozor.cz/>
- [7] LEVEL s.r.o. Elektronický monitoring [online]. Náchod: LEVEL, ©2025 [cit. 2025-04-13]. Dostupné z: <https://www.level.systems/cs/>
- [8] MISRA, Pratap a ENGE, Per. *Global Positioning System: Signals, Measurements, and Performance*. 3rd ed. Lincoln: Ganga-Jamuna Press, 2011. 136-144, 385-389 s. ISBN 978-0-9709544-5-6.
- [9] TEUNISSEN, Peter J. G. a MONTENBRUCK, Oliver, eds. *Springer Handbook of Global Navigation Satellite Systems*. Cham: Springer, 2017. 385-389, 1327 s. ISBN 978-3-319-42928-1.
- [10] BHATTA, Basudeb. *Global Navigation Satellite Systems: New Technologies and Applications*. 2nd ed. Boca Raton: CRC Press, 2021. 52-59, 65-72, 174-182, 325-332, 381-402 s. ISBN 978-0-367-47882-2.

- [11] GILL, Saikat, ed. *Internet-of-Things (IoT) Systems*. Cham: Springer, 2020. 137-144 s. ISBN 978-0-12-819399-8.
- [12] KANDAR, D. a MISHRA, P. K. GPS-GIS integrated systems for transportation engineering [online]. *Geospatial World*, 2001 [cit. 2025-04-18]. Dostupné z: <https://www.geospatialworld.net/article/gps-gis-integrated-systems-for-transportation-engineering/>
- [13] European Space Agency. Fleet Management and Vehicle Tracking [online]. *Navipedia*, 2011 [cit. 2025-04-18]. Dostupné z: https://gssc.esa.int/navipedia/index.php/Fleet_Management_and_Vehicle_Tracking/
- [14] OHLHORST, Frank. *Wi-Fi Handbook: Building 802.11b Wireless Networks*. New York: McGraw-Hill Education, 2003. 275-283 s. ISBN 978-0071412513.
- [15] GAST, Matthew S. *802.11 Wireless Networks: The Definitive Guide*. 2nd ed. Sebastopol: O'Reilly Media, 2005. 584-591 s. ISBN 978-0596100520.
- [16] MAZIDI, Muhammad Ali, NAIMI, Sepehr a NAIMI, Sarmad. *The AVR Microcontroller and Embedded Systems Using Assembly and C*. Upper Saddle River: Prentice Hall, 2010. 78-90 s. ISBN 978-0138003319.
- [17] YU, Chien-Hung. *The STM32F103 Arm Microcontroller and Embedded Systems: Using Assembly and C*. Lake Dallas: E-Man Press, 2019. 41-48 s. ISBN 978-0982692639.
- [18] MICRON TECHNOLOGY, INC. *NOR Flash Memory: Technical Note TN-13-30*. Boise: Micron Technology, 2017. 99-132 s. Dostupné z: https://media-www.micron.com/-/media/client/global/documents/products/technical-note/nor-flash/tn1330_nor_flash_guide.pdf [cit. 2025-04-29].
- [19] CHEN, Jian-Ping. *Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices*. Hoboken: Wiley-IEEE Press, 2021. 639-660 s. ISBN 978-1119695095.
- [20] DELANEY, Kalen a kol. *Microsoft SQL Server 2019 Internals*. Redmond: Microsoft Press, 2020. 22-27, 62-67, 176-184 s. ISBN 978-0136719946.
- [21] JARKA, Dejan Sarka. *JSON in SQL Server 2016 and Azure SQL Database*. 2nd ed. Seattle: SolidQ, 2016. 161-169 s. ISBN 978-1530065415.

- [22] BLUM, Jeremy. *Exploring Arduino: Tools and Techniques for Engineering Wizardry*. 2nd ed. Indianapolis: Wiley, 2019. 308-316 s. ISBN 978-1119405373.
- [23] BARRY, Richard. *Mastering the FreeRTOS Real-Time Kernel*. Raleigh: Real Time Engineers Ltd., 2016. 70-105 s. ISBN 978-1907920444.
- [24] BEAZLEY, David a JONES, Brian K. *Python Cookbook: Recipes for Mastering Python* 3. 3rd ed. Sebastopol: O'Reilly Media, 2013. 313-328 s. ISBN 978-1449340377.
- [25] FORCIER, Jeff, BOWER, Paul a VELTMAN, Wesley Chun. *Python Web Development with Django*. Boston: Addison-Wesley Professional, 2009. 205-213 s. ISBN 978-0132356138.
- [26] DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. Indianapolis: Wiley, 2011. 15-27, 62-80 s. ISBN 978-1118008188.
- [27] FLANAGAN, David. *JavaScript: The Definitive Guide*. 7th ed. Sebastopol: O'Reilly Media, 2020. 161-169 s. ISBN 978-1491952023.
- [28] PETERSON, Michael P. *Mapping in the Cloud*. New York: Guilford Press, 2014. 1-4 s. ISBN 978-1462510385.

SEZNAM OBRÁZKŮ

Obrázek 1 GNSS – 4 systémy.....	13
Obrázek 2 Určování GNSS polohy.....	24
Obrázek 3 Vývojový GNSS modul.....	33
Obrázek 4 Vývojový modul mikrořadiče.....	34
Obrázek 5 Blokové schéma hardware palubní jednotky.....	46
Obrázek 6 Zapojení mikrořadiče ESP32-C6.....	53
Obrázek 7 Zapojení GNSS modulu NEO-M9N	54
Obrázek 8 Paměť NOR flash	54
Obrázek 9 Zdroj napájení.....	55
Obrázek 10 Zapínání systému přes GPIO.....	56
Obrázek 11 Výsledný GNSS modul palubní jednotky	56
Obrázek 12 Blokové schéma software palubní jednotky.....	57
Obrázek 13 Blokové schéma software serverové části.....	62
Obrázek 14 Živé zobrazení polohy vozidel – výřez	65
Obrázek 15 Sumarizační report za jednotlivá vozidla	65

SEZNAM TABULEK

Tabulka 1 Hodnoty parametrů řešerše	22
Tabulka 2 Spotřeba celého obvodu – naměřené hodnoty	59
Tabulka 3 Struktura dat jednoho záznamu pro uložení do flash paměti.....	60
Tabulka 4 Struktura tabulek databáze MS SQL	68
Tabulka 5 Přehled klíčových testovaných oblastí a výsledků	73

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ACID	Atomicity, Consistency, Isolation, Durability
AD	Active Directory
AD/DA	Analog to Digital / Digital to Analog
AI	Artificial Intelligence (Umělá inteligence)
AJAX	Asynchronous JavaScript and XML/JSON
AP	Access Point (Přístupový bod)
API	Application Programming Interface
AVR	Alf and Vegard's RISC
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CCS	Czech Card Service (CCS tankovací karta)
CPU	Central Processing Unit
CRM	Customer Relationship Management (Řízení vztahů se zákazníky)
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CTE	Common Table Expressions
DC/DC	Direct Current / Direct Current converter (DC/DC měnič)
DGNSS	Differential Global Navigation Satellite System
DOM	Document Object Model
DOP	Dilution of Precision (Rozptyl přesnosti)
EGNOS	European Geostationary Navigation Overlay Service
eMMC	Embedded MultiMediaCard
ERP	Enterprise Resource Planning (Plánování podnikových zdrojů)
eSIM	Embedded SIM
FAT	File Allocation Table

GET	HTTP GET Method
GNSS	Global Navigation Satellite System (Globální navigační satelitní systém)
GPIO	General Purpose Input/Output (Vstupně-výstupní univerzální port)
GPS	Global Positioning System (Globální poziční systém)
GSM	Global System for Mobile Communications
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protokol
HW	Hardware
I ² C	Inter-Integrated Circuit
ID	Identification (Identifikace)
IDE	Integrated Development Environment (Integrované vývojové prostředí)
IMU	Inertial Measurement Unit (Inerciální měřící jednotka)
IoT	Internet of Things (Internet věcí)
IP	Internet Protocol
IT	Information Technology (Informační technologie)
JS	JavaScript
JSON	JavaScript Object Notation
LAN	Local Area Network (Lokální počítačová síť)
LED	Light Emitting Diode (Světelná dioda)
MAC	Media Access Control
MIMO	Multiple Input Multiple Output
MISO	Master In Slave Out
MOSI	Master Out Slave In
MS	Microsoft
MU	Multi-User
NAND	Not AND Flash

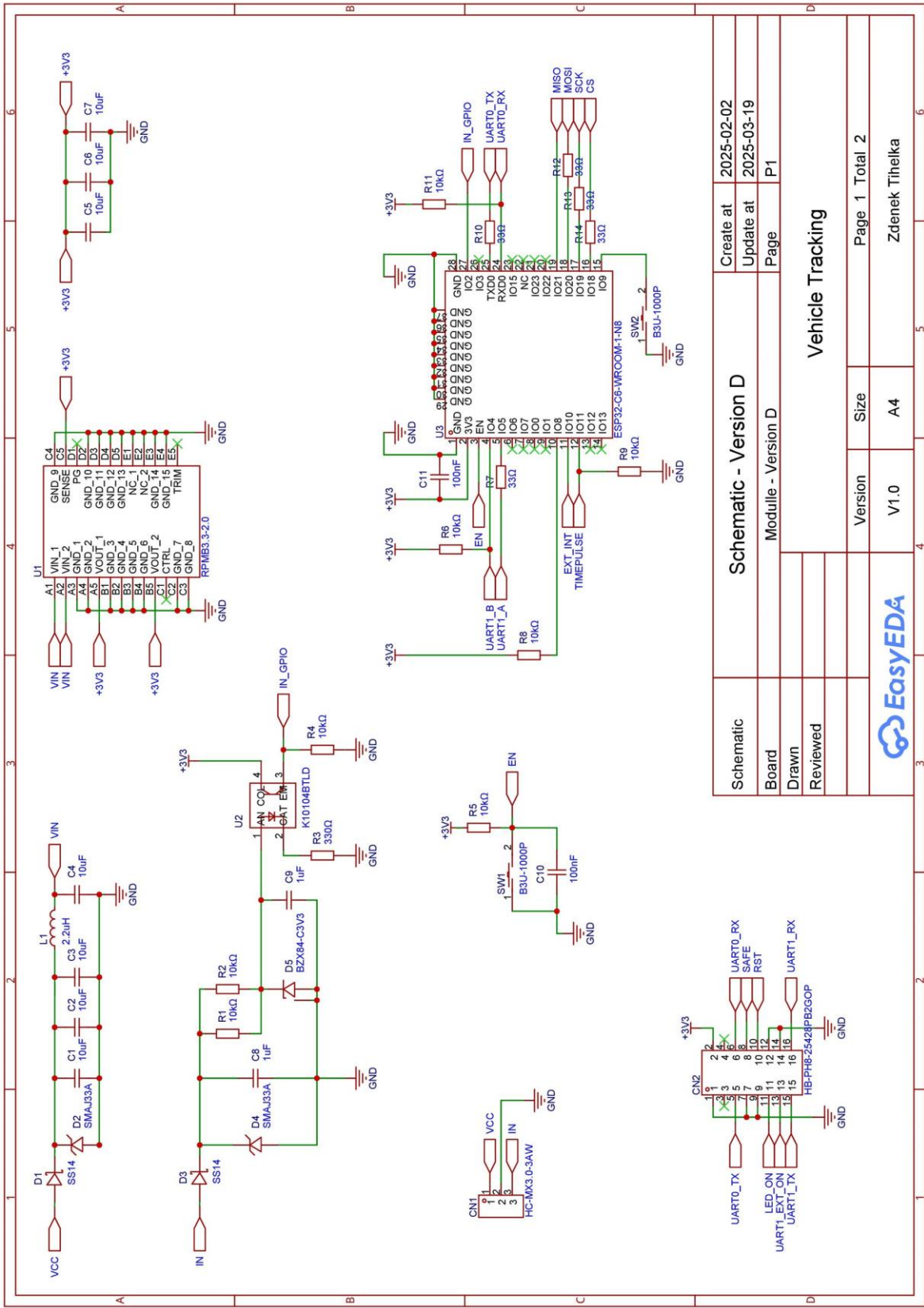
NMEA	National Marine Electronics Association
NOR	NOR (Negated OR) Flash
OFDMA	Orthogonal Frequency Division Multiple Access
ORM	Object Relational Mapping
OS	Operating System (Operační systém)
PDF	Portable Document Format
POST	HTTP POST Method
PPP	Precise Point Positioning
QAM	Quadrature Amplitude Modulation
RF	Radio Frequency (Rádiová frekvence)
RFID	Radio Frequency Identification
RISC-V	Reduced Instruction Set Computer - Five
ROM	Read-Only Memory (Paměť pouze pro čtení)
RTK	Real-Time Kinematic
RTOS	Real-Time Operating System (Operační systém reálného času)
RX	Receive (Příjem)
SAP	Systems, Applications, and Products
SBAS	Satellite Based Augmentation System
SCLK	Serial Clock
SD	Secure Digital
SIM	Subscriber Identity Module
SMS	Short Message Service (Krátká textová zpráva)
SPI	Serial Peripheral Interface
SQL	Structured Query Language
SS	Slave Select
SSID	Service Set Identifier (Název bezdrátové sítě)

T-SQL	Transact-SQL
TX	Transmit (Vysílání)
UART	Universal Asynchronous Receiver-Transmitter
UBX	U-Blox Binary Protocol (binární protokol společnosti U-Blox)
UI	User Interface (Uživatelské rozhraní)
UniFi	UniFi (produktová řada Ubiquiti Networks)
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network (Virtuální síť)
VPN	Virtual Private Network (Virtuální privátní síť)
W3C	World Wide Web Consortium
WAAS	Wide Area Augmentation System
Wi-Fi	Wireless Fidelity (Bezdrátová síť)
WLAN	Wireless Local Area Network (Bezdrátová místní síť)
WPA2	Wi-Fi Protected Access 2
XML	eXtensible Markup Language

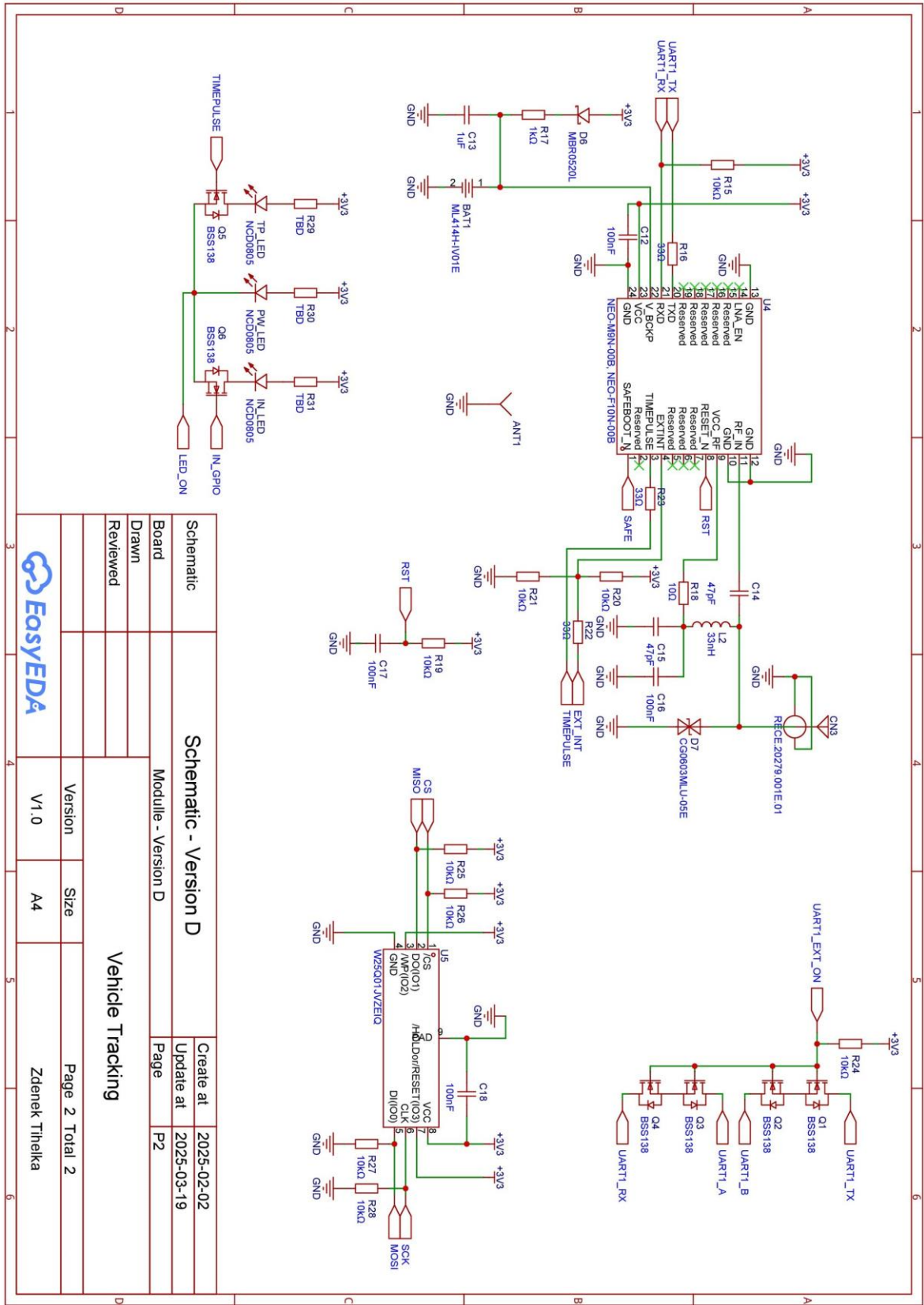
SEZNAM PŘÍLOH

- Příloha P 1: Schéma obvodu 1. část
- Příloha P 2: Schéma obvodu 2. část
- Příloha P 3: Software palubní jednotky
- Příloha P 4: Adresářová struktura software palubní jednotky
- Příloha P 5: Software serverové části
- Příloha P 6: Adresářová struktura software serverové části
- Příloha P 7: Živé zobrazení polohy vozidel
- Příloha P 8: Sumarizační report za jednotlivé dny
- Příloha P 9: Report jednotlivých jízd vozidel

PŘÍLOHA P 1: SCHÉMA OBVODU 1. ČÁST



PŘÍLOHA P 2: SCHÉMA OBVODU 2. ČÁST



Schematic		Schematic - Version D		Create at	2025-02-02
Board		Module - Version D		Update at	2025-03-19
Drawn				Page	P2
Reviewed					
		Vehicle Tracking			
		Version	Size	Page 2 Total 2	
		V1.0	A4	Zdenek Tihelka	



PŘÍLOHA P 3: SOFTWARE PALUBNÍ JEDNOTKY

Programovací jazyk	Verze
C++ (Arduino IDE)	2.3.6

Rozšíření	Verze
ESP32 (Espressif)	3.2.0

Knihovny	Verze	Popis funkce knihovny
<Arduino.h>	*	Základní funkce Arduino
<ArduinoJson.h>	7.4.1	Manipulace a zpracování JSON dat
<freertos/FreeRTOS.h>	ESP32	Základní funkcionalita operačního systému FreeRTOS
<freertos/task.h>	ESP32	Správa úloh (tasků) ve FreeRTOS
<freertos/semphr.h>	ESP32	Podpora synchronizačních primitiv (mutexy, semaforey)
<freertos/event_groups.h>	ESP32	Implementace skupin událostí pro synchronizaci úloh
<Preferences.h>	ESP32	Ukládání a načítání trvalých dat do vlastní flash paměti
<LittleFS.h>	ESP32	Souborový systém pro vlastní flash Paměť
<SPI.h>	*	Podpora sériového SPI rozhraní
<WiFi.h>	ESP32	Správa připojení WiFi sítě
<SparkFun_u-blox_GNSS_v3.h>	3.1.9	Knihovna pro práci s GNSS moduly U-blox
<esp_mac.h>	ESP32	Funkce pro práci s MAC adresou
<esp_sleep.h>	ESP32	Podpora režimů hlubokého spánku a úspory energie
<map>	STL	Asociativní kontejnery typu klíč-hodnota
<vector>	STL	Dynamické pole s automatickým přizpůsobením velikosti
<algorithm>	STL	Standardní algoritmy (řazení, hledání, porovnávání)
<tuple>	STL	Podpora práce s datovou strukturou uspořádané n-tice
STL - základní C++, * - základní Arduino, ESP32 - rozšíření ESP32 x.x.x - externí knihovna		

PŘÍLOHA P 4: ADRESÁŘOVÁ STRUKTURA SOFTWARE PALUBNÍ JEDNOTKY

sw_vehicle	
globals.cpp	- společné či nezařazené funkce (konfigurace, logování)
globals.h	
gnss_data.cpp	- funkce pro práci s GNSS (Read-GNSS-Task)
gnss_data.h	
nor_flash.cpp	- funkce pro práci s NOR FLASH
nor_flash.h	
started_up_monitor.cpp	- funkce pro sledování STARTED UP (Started-Up-Monitor-Task)
started_up_monitor.h	
upload_to_server.cpp	- funkce pro odesílání dat na server (Live-Upload-Task, Save-Upload-Task)
upload_to_server.h	
vehicle_tracking.ino	- spouštěcí část programu (Setup, inicializace RTOS úloh)
web_server.cpp	- funkce pro zobrazení logu klientovi (Web-Server-Task)
web_server.h	
wifi_monitor.cpp	- funkce pro obsluhu Wi-Fi (WiFi-Monitor-Task)
wifi_monitor.h	
data	
ap_list.txt	- seznam známých BSSID (pojmenování AP)
config.txt	- konfigurační soubor
README.md	- informace k programu

(celý kód programu je umístěn jako samostatná příloha diplomové práce)

PŘÍLOHA P 5: SOFTWARE SERVEROVÉ ČÁSTI

Programovací jazyk	Verze
Python	3.12.6

Knihovny	Verze	Popis funkce knihovny
http.server	*	Implementuje jednoduchý HTTP server
socketserver	*	Umožňuje vytvářet síťové servery a zpracovávat požadavky klientů
socket	*	Zajišťuje komunikaci systému na nízké úrovni
time	*	Poskytuje práci s časem
os	*	Umožňuje komunikaci s operačním systémem
json	*	Zajišťuje práci s datovým formátem JSON
numpy as np	2.1.2	Podpora rychlých matematických operací a polí
threading	*	Umožňuje práci s vlákny (paralelní běh úloh)
pytz	2024.2	Práce s časovými pásmy
pyodbc	5.2.0	Umožňuje přístup a práci s databázemi (MS SQL)
PIL (Image)	11.0.0	Zajišťuje práci s obrázky (čtení, úpravy a ukládání obrázků)
datetime (datetime)	*	Poskytuje nástroje pro manipulaci s datem a časem
* - základní Python, x.x.x / x.x - externí knihovna		

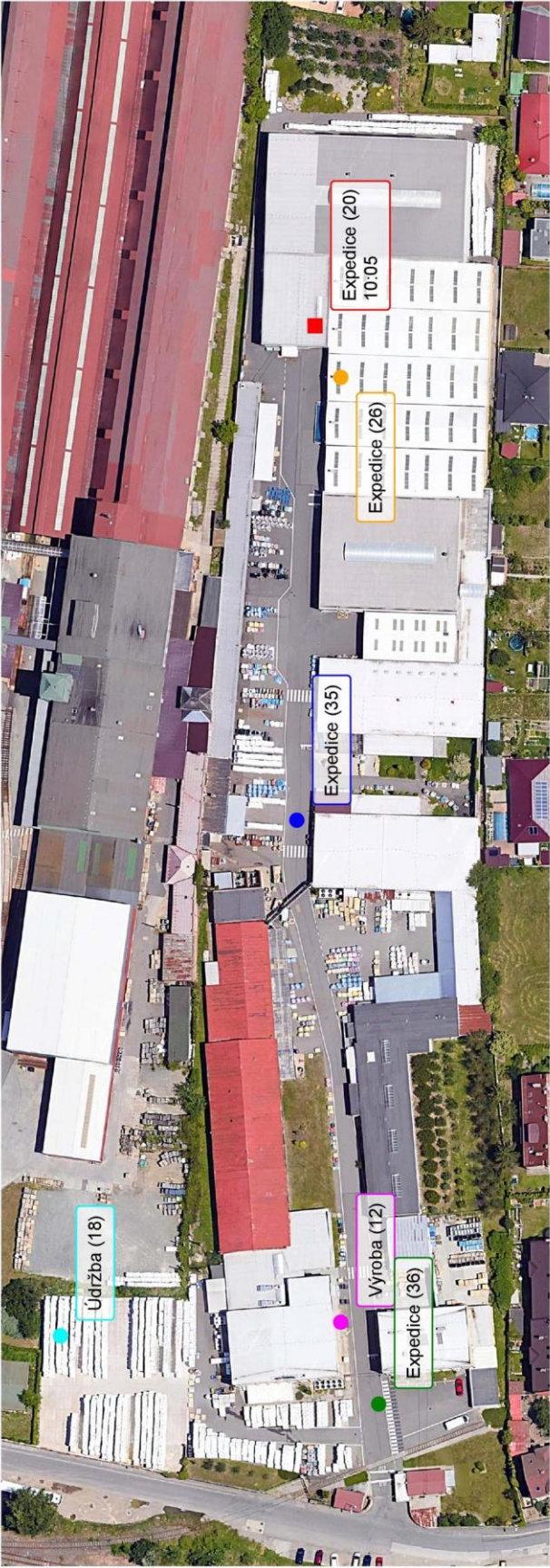
PŘÍLOHA P 6: ADRESÁŘOVÁ STRUKTURA SOFTWARE SERVEROVÉ ČÁSTI

sw_server	
main.py	- spouštěcí část programu
README.md	- informace k programu
data	
known_points.txt	- známé body souřadnic a absolutní pozice na mapě areálu
img	
mapa.jpg	- mapa areálu
package	
common.py	- sdílená data napříč programem
cte_template.py	- report dotazy CTE pro MS SQL
database.py	- dynamické funkce pro práci s MS SQL (SELECT, UPDATE, atp.)
globals.py	- společné či nezařazené funkce
html_template.py	- šablony html stránek pro klienta
http_handlers.py	- dynamické funkce volané z main.py handleru pro POST, GET (API, obsluha klienta, zpracování dat a ukládání dat do databáze)
__init__.py	- struktura projektu

(celý kód programu je umístěn jako samostatná příloha diplomové práce)

PŘÍLOHA P 7: ŽIVÉ ZOBRAZENÍ POLOHY VOZIDEL

Vehicle Tracking



Reports

Forced Remove Unknown

Forced Remove All

Vehicle Name	Mac Address	Hostname	IP Address	BSSID	Channel	RSSI	Satellites	Location At Local Time	Latitude	Longitude	Speed [m/s]	Idle Time
Výroba (12)	40:4c:ca:4f:91:40	10tm-gps-4f9140	172.22.111.60	Trasa - SH vstup linky (WF_22)	1	-56	30	2025-05-20 13:25:48	49.580793	17.288466	0.32	0:00
Expedice (20)	40:4c:ca:4e:68:7c	10tm-gps-4e687c	172.22.111.148	ea:b3:70:bd:22:a7	1	-45	21	2025-05-20 13:15:52	49.582042	17.291604	0.09	10:05
Expedice (35)	40:4c:ca:4f:91:54	10tm-gps-4f9154	172.22.111.87	ea:b3:70:bd:60:72	11	-61	27	2025-05-20 13:25:47	49.581483	17.289988	5.26	0:00
Expedice (36)	40:4c:ca:4f:7f:1c	10tm-gps-4f7f1c	172.22.111.18	ea:b3:70:bd:61:f9	1	-55	30	2025-05-20 13:25:45	49.580618	17.288272	1.93	0:00
Expedice (26)	40:4c:ca:4f:7f:14	10tm-gps-4f7f14	172.22.111.12	ea:b3:70:bd:65:95	6	-56	16	2025-05-20 13:25:46	49.581925	17.291484	2.16	0:00
Údržba (18)	40:4c:ca:4f:91:c8	10tm-gps-4f91c8	172.22.111.120	ea:b3:70:bd:61:17	11	-70	32	2025-05-20 13:25:48	49.581352	17.287910	1.26	0:00

PŘÍLOHA P 8: SUMARIZAČNÍ REPORT ZA JEDNOTLIVÉ DNY

Dates Trips Summary

Month: 2025-05, Time Range: 00:00:00 - 23:59:59

Vehicles: ■ Expedice (20), ■ Expedice (35), ■ Expedice (36), ■ Expedice (26), ■ Údržba (18), ■ Výroba (12)

Date	Count	Engine Time	Idle Time	Off Time	Distance [m (km)]
Σ	2150	244:05:02	77:57:34	1511:03:12	1260962 (1261.0)
2025-05-01	71	5:14:56	1:20:53	42:45:02	31794 (31.8)
2025-05-02	88	15:50:15	4:43:16	104:09:40	84223 (84.2)
2025-05-03	78	4:48:45	1:10:55	43:41:05	29665 (29.7)
2025-05-04	66	4:30:20	53:52	43:46:19	26905 (26.9)
2025-05-05	192	20:46:33	6:43:17	123:13:21	103836 (103.8)
2025-05-06	172	20:35:47	6:46:35	123:24:07	102378 (102.4)
2025-05-07	195	20:32:42	6:52:14	124:08:33	111746 (111.7)
2025-05-08	100	7:52:59	1:49:18	64:06:58	49440 (49.4)
2025-05-09	129	17:56:08	5:31:40	102:27:48	97225 (97.2)
2025-05-10	44	3:48:31	53:40	20:11:28	22122 (22.1)
2025-05-11	52	2:40:57	21:53	21:19:02	20073 (20.1)
2025-05-12	147	21:10:19	7:30:49	122:49:35	101712 (101.7)
2025-05-13	173	22:12:05	7:07:54	121:47:49	106537 (106.5)
2025-05-14	191	21:30:16	7:28:04	122:29:38	104497 (104.5)
2025-05-15	183	20:26:11	6:55:43	123:57:45	106657 (106.7)
2025-05-16	152	24:43:39	9:44:24	119:16:15	104067 (104.1)
2025-05-17	62	5:13:45	1:05:58	42:46:13	34107 (34.1)
2025-05-18	55	4:10:54	57:09	44:42:34	23978 (24.0)

PŘÍLOHA P 9: REPORT JEDNOTLIVÝCH JÍZD VOZIDEL

Vehicles Trips Detail

Month: 2025-05, Time Range: 08:40:00 - 09:30:00

Vehicles: ■ Výroba (12)

Vehicle Name	Date	Count	Number	Start	End	Engine Time	Idle Time	Off Time	Distance [m (km)]
■ Výroba (12)	Σ					1:12:41	16:00	6:17:19	7315 (7.3)
	2025-05-03	2	Σ			2:42	0:58	47:18	280 (0.3)
								14:36	
			1.	08:54:36	08:55:14	0:38	0:18	15:07	18 (0.0)
			2.	09:10:21	09:12:25	2:04	0:40	17:35	262 (0.3)
	2025-05-04	2	Σ			17:36	3:26	32:24	1338 (1.3)
								1:22	
			1.	08:41:22	08:49:23	8:01	1:03	31:01	617 (0.6)
			2.	09:20:24	09:29:59	9:35	2:23	0:01	721 (0.7)
	2025-05-05	1	Σ			4:27	0:37	45:33	292 (0.3)
								16:51	
			1.	08:56:51	09:01:18	4:27	0:37	28:42	292 (0.3)
	2025-05-07	6	Σ			6:45	2:38	43:15	605 (0.6)
								1:58	
			1.	08:41:58	08:45:26	3:28	1:48	2:54	274 (0.3)
			2.	08:48:20	08:48:51	0:31	0:07	5:25	70 (0.1)
			3.	08:54:16	08:55:29	1:13	0:07	0:24	147 (0.1)
			4.	08:55:53	08:56:18	0:25	0:07	11:58	33 (0.0)
			5.	09:08:16	09:08:48	0:32	0:12	8:55	55 (0.1)
			6.	09:17:43	09:18:19	0:36	0:17	11:41	26 (0.0)
	2025-05-10	3	Σ			24:55	6:36	25:05	2381 (2.4)
								4:13	
			1.	08:44:13	08:48:14	4:01	0:33	6:22	469 (0.5)
			2.	08:54:36	09:10:32	15:56	5:04	4:36	1364 (1.4)
			3.	09:15:08	09:20:06	4:58	0:59	9:54	548 (0.5)
	2025-05-13	1	Σ			3:41	0:20	46:19	537 (0.5)
								35:58	
			1.	09:15:58	09:19:39	3:41	0:20	10:21	537 (0.5)
	2025-05-15	1	Σ			0:46	0:12	49:14	36 (0.0)
								18:21	
			1.	08:58:21	08:59:07	0:46	0:12	30:53	36 (0.0)
	2025-05-17	3	Σ			9:21	0:53	40:39	1265 (1.3)
								21:22	
			1.	09:01:22	09:04:40	3:18	0:17	4:36	532 (0.5)
			2.	09:09:16	09:14:46	5:30	0:28	5:24	667 (0.7)
			3.	09:20:10	09:20:43	0:33	0:08	9:17	66 (0.1)
	2025-05-18	2	Σ			2:28	0:20	47:32	581 (0.6)
								6:54	
			1.	08:46:54	08:48:00	1:06	0:08	8:52	369 (0.4)
			2.	08:56:52	08:58:14	1:22	0:12	31:46	212 (0.2)