



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

MĚŘENÍ POLOHY UVNITŘ BUDOV POMOCÍ INERCIÁLNÍ JEDNOTKY

INDOOR POSITIONING BASED ON INERTIAL MEASUREMENT UNIT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Marek Coufal

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Král, Ph.D.

BRNO 2024

Bakalářská práce

bakalářský studijní program **Elektronika a komunikační technologie**

Ústav radioelektroniky

Student: Marek Coufal

ID: 240598

Ročník: 3

Akademický rok: 2023/24

NÁZEV TÉMATU:

Měření polohy uvnitř budov pomocí inerciální jednotky

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s dostupnými jednotkami pro měření inerciální a neinerciální polohy. Navrhněte a realizujte vhodný hardware samostatné bezdrátové jednotky, která bude umět zaznamenávat interně své údaje do paměti a na konci měření tato data přenést do počítače.

Realizujte ovládací SW a FW a skripty v Matlabu nebo Pythonu pro určení polohy uvnitř budov za pomoci GPS. Funkci demonstруйте. Všechny programy realizujte tak, aby byly vhodné a použitelné v rámci laboratorní úlohy předmětu MPC-RAR. Navrhněte koncept této laboratorní úlohy.

DOPORUČENÁ LITERATURA:

- [1] MALÝ, Martin. Hradla, volty, jednočipy: Úvod do bastlení. Praha: CZ.NIC, 2017. ISBN 978-80-88168-26-3.
- [2] RITCHIE, Dennis M., KERNIGHAN, Brian W. The C Programming Language, Second Edition. New Jersey: Prentice Hall, 1988. ISBN 9780133086249.

Termín zadání: 16.2.2024

Termín odevzdání: 27.5.2024

Vedoucí práce: Ing. Jan Král, Ph.D.

doc. Ing. Lucie Hudcová, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá inerciální navigací pro použití uvnitř budov. Je rozebrán algoritmus zpracování dat akcelerometrů a gyroskopů a jejich zpracování pomocí pohybových rovnic. Jsou také definovány různé vztažné soustavy pro potřeby manipulace s vektorovými fyzikálními veličinami a jejich převody a rotace mezi soustavami. Práce se také věnuje návrhu hardwaru inerciální jednotky, minimálním požadavkům pro rychlý záznam dat. Jsou věnovány kapitoly samotnému návrhu desky plošných spojů a realizaci zařízení, následnému vývoji firmwaru a softwaru na přenos a zpracování dat.

KLÍČOVÁ SLOVA

IMU, inerciální navigace, poloha, orientace, připevněný, vztažné soustavy, Euler, gravitace, gyroskop, akcelerometr, magnetometr, GNSS, STM32, ESP32, MATLAB, kalibrace

ABSTRACT

This bachelor's thesis deals with inertial navigation intended for indoor usage. The algorithm for processing accelerometer and gyro data is described along with their position in equations of motion. Various coordinate systems are defined for ease of manipulation of vector quantities, including conversions and rotations between frames. The thesis also addresses the design of the hardware for the inertial unit, including the minimum requirements for fast data recording. Some chapters are dedicated to the design of the printed circuit board and the implementation of the device, followed by the development of firmware and software for data logging and processing.

KEYWORDS

IMU, inertial navigation, position, orientation, strapdown, coordinate systems, Euler, gravity, gyro, accelerometer, magnetometer, GNSS, STM32, ESP32, MATLAB, calibration

COUFAL, Marek. *Měření polohy uvnitř budov pomocí inerciální jednotky* . Bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2024. Vedoucí práce: Ing. Jan Král, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Marek Coufal
VUT ID autora:	240598
Typ práce:	Bakalářská práce
Akademický rok:	2023/24
Téma závěrečné práce:	Měření polohy uvnitř budov pomocí inerciální jednotky

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

* Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Janu Královi, Ph.D. za odborné vedení, konzultace, trpělivost, vstřícnost a podnětné návrhy k práci.

Obsah

Úvod	11
1 Algoritmus inerciální navigace	12
1.1 Využití inerciální navigace	12
1.2 Princip fungování inerciální navigace	12
1.2.1 Zavedení vztažných soustav	13
1.3 Vztažné soustavy a rotace Země	13
1.4 Tíhové pole Země	14
1.4.1 Gravitační modely Země	15
1.5 Rotace měření akcelerometru	16
1.6 Diagram algoritmu inerciální navigace	17
2 Inerciální senzory	18
2.1 MEMS akcelerometry	18
2.2 MEMS gyroskopy	19
3 Hardware inerciální jednotky	20
3.1 Akcelerometr a gyroskop	21
3.2 Magnetometr	23
3.3 GNSS	24
3.4 Paměť	25
3.5 Uživatelské rozhraní	25
3.6 Napájení	26
3.7 Hlavní procesor	27
3.8 ESP32	28
3.9 Testování s vývojovými stavebnicemi	29
4 Realizace hardwaru	30
4.1 Konstrukce	31
4.2 Ověření základních funkcí hardwaru jednotky	33
5 Firmware inerciální jednotky	35
5.1 HAL	35
5.2 FreeRTOS	35
5.3 Vývojové diagramy firmwaru	36
5.3.1 KeepaliveTask	37
5.3.2 hubTask	37
5.3.3 powerTask	38

5.3.4	gpsTask	38
5.3.5	lsmTask	39
5.3.6	mpuTask	40
5.3.7	adisTask	40
5.3.8	loggerTask	41
5.3.9	oledTask	42
5.4	Kalibrace IMU	44
5.5	Převod dat do CSV souboru	47
6	Zpracování dat	48
6.1	Výpočet trajektorie pomocí pohybových rovnic	48
6.2	Vliv kalibrace senzorů	50
6.3	Fúze dat s GNSS	51
7	Koncepce laboratorní úlohy	53
7.1	Analýza využití GNSS, dostupnost signálu a obsluha	53
7.2	Princip fungování IMU, vliv kalibrace a určení náklonu a polohy inerciální jednotkou	54
	Závěr	55
	Literatura	56
	Seznam symbolů a zkratk	60
	Seznam příloh	62
A	Schéma zapojení inerciální jednotky	63
B	Výkres DPS	70
B.1	Pohled osazení součástek	70
B.2	Vrchní vrstva mědi DPS	71
B.3	Vnitřní vrstva mědi DPS In1	72
B.4	Vnitřní vrstva mědi DPS In2	73
B.5	Spodní vrstva mědi DPS	74
C	Obsah elektronické přílohy	75

Seznam obrázků

1.1	Výškový reliéf geoidu. [7]	16
1.2	Blokové schéma algoritmu strapdown inerciální navigace, převzato z [1] [2].	17
2.1	Struktura MEMS akcelerometru. [8]	18
2.2	Mikroskopický snímek MEMS akcelerometru. [9]	19
2.3	Struktura MEMS gyroskopu. [8]	19
3.1	Blokové schéma inerciální jednotky.	20
3.2	IMU verze ADIS16505-2/PCBZ.	23
3.3	Fotografie grafického OLED displeje.	26
3.4	Testovací přípravek s vývojovými deskami.	29
4.1	3D model navržené DPS.	30
4.2	Izolační podložka IMU.	32
4.3	3D model krabičky.	32
4.4	Fotografie zařízení.	33
4.5	Dodatečná oprava RTC oscilátoru.	34
5.1	Konfigurace MCU v STM32CubeIDE.	36
5.2	Vývojový diagram KeepaliveTask.	37
5.3	Vývojový diagram HubTask.	37
5.4	Vývojový diagram PowerTask.	38
5.5	Vývojový diagram GpsTask.	39
5.6	Vývojový diagram LsmTask.	39
5.7	Vývojový diagram MpuTask.	40
5.8	Vývojový diagram AdisTask.	41
5.9	Vývojový diagram LoggerTask.	42
5.10	Vývojový diagram OledTask.	43
5.11	GUI stavových informací jednotlivých senzorů.	44
5.12	GUI záznamu dat.	44
5.13	GUI kalibrační procedury.	45
6.1	Záznam dat IMU ADIS16505.	49
6.2	Výsledky po první integraci.	49
6.3	Odhad trajektorie.	50
6.4	Porovnání senzorů s kalibrací a bez kalibrace.	51
6.5	Naměřená GNSS data.	52
6.6	Odhad trajektorie z fúze dat GNSS a IMU.	52
7.1	Obslužná aplikace u-center.	53

Seznam tabulek

3.1	Porovnání základních parametrů gyroskopů. [11] [13]	22
3.2	Podporované družicové systémy. [16]	24
3.3	Odhad celkového datového toku pro záznam dat.	25
3.4	Odhad spotřeby proudu 3,3V větve.	27
3.5	Minimální požadavky na periferie mikroprocesoru.	28
4.1	Signálové uspořádání vrstev na DPS.	31
4.2	Uspořádání měděných a izolačních vrstev DPS JLC04161H-7628. . . .	31

Úvod

Tato práce se zabývá poměrně komplexní problematikou inerciální navigace pro použití uvnitř budov z údajů o lineárním zrychlení z akcelerometru a úhlové rychlosti z gyroskopu pro omezení potřeby neustálé dostupnosti signálu z globálních navigačních systémů.

Jsou popsány algoritmy pro zpracování dat z šestiosé inerciální jednotky s nehybně umístěnými gyroskopy a akcelerometry typu mikro-elektromechanických systémů (*Micro-ElectroMechanical Systems*, MEMS), což umožňuje značnou miniaturizaci a snížení ceny, v porovnání s osvědčenými systémy využívající velké a složité mechanické konstrukce gimbalů a gyroskopů. Přestože se přesnost senzorů stále zlepšuje, i díky malé odchylce měřených dat může rychle narůstat chyba odhadované polohy díky integraci úhlové rychlosti a dvojitě integraci lineárního zrychlení. Z tohoto důvodu je jednotka opatřena i magnetometrem a modulem globálního družicového polohového systému (*Global navigation satellite system*, GNSS) pro možnost fúze dalších dat pro co největší zmenšení chyby.

Jsou také věnovány kapitoly samotnému návrhu inerciální jednotky, minimálním požadavkům na komunikace a jejich počtům a rychlostem, způsobu zaznamenávání dat a celkovému blokovému konceptu. Je popsán postup konstrukce zařízení a jeho oživení.

Další kapitoly se věnují vývoji firmwaru pro procesor z řady STM32 a jednotlivým funkcím stavových automatů vláken operačního systému reálného času. Jednotka umožňuje přenos naměřených dat do počítače, kde se pomocí vytvořeného skriptu převedou do čitelné podoby.

Předposlední kapitola je věnována zpracování dat, popisuje problémy nepřesnosti senzorů a úspěšnosti navržené kalibrační procedury. Je také otestována fúze dat z jiných senzorů. Poslední část práce je věnována návrhu využití zařízení v laboratorní úloze.

1 Algoritmus inerciální navigace

1.1 Využití inerciální navigace

Možnost navigace a znalost polohy je pro lidstvo již dlouhou dobu důležitá jak v průmyslu, tak i v každodenním životě. Pravděpodobně nejrozšířenějším druhem navigace jsou tzv. globální navigační systémy, například globální polohový systém (*Global Positioning System*, GPS). Ovšem pro některé aplikace nemusí být použití GNSS, ať už z politických či technických důvodů ideální. Pro navigaci v oblasti letectví a námořnictví se začala inerciální navigace využívat kolem roku 1960 a je využívána dodnes. [1]

Díky stále přesnějším a levnějším inerciálním sensorům se rozšiřují možnosti využití inerciální navigace i v běžných průmyslových aplikacích, například v oblastech robotiky, automobilové techniky, nebo i pro údržbu podzemních infrastruktur, mapování kanalizací a další. [1] Tato práce se zabývá využitím inerciální navigace pro účely určení polohy uvnitř budov a fúzí dat z dalších sensorů.

1.2 Princip fungování inerciální navigace

Inerciální navigační systémy pracují na principu nepřímého měření z dat, které poskytuje akcelerometr a gyroskop. Akcelerometry poskytují informaci o lineárním zrychlení v prostoru pomocí měření síly F na definovanou jednotku hmotnosti m a pomocí druhého Newtonova zákona určí zrychlení a [1]

$$a = \frac{F}{m}. \quad (1.1)$$

Síla F představuje síly působící na senzor vůči jeho tělu ve volném pádu, skládá se tedy ze statické (tíhové) a dynamické síly způsobené zrychlením vůči Zemi. [1] Z tohoto důvodu, pokud je akcelerometr v klidu na povrchu Země, změří zrychlení o velikosti zhruba $9,81 \text{ m}\cdot\text{s}^{-2}$.

Akcelerometry zpravidla měří hodnoty lineárního zrychlení ve třech navzájem pravouhlých osách. Znalostí počáteční rychlosti $v(t_0)$ a polohy $x(t_0)$ v čase t_0 můžeme pomocí zrychlení a v časech $s > t_0$ určit rychlost $v(t)$ a následně polohu $x(t)$ pomocí dvou integrací [2]

$$v(t) = v(t_0) + \int_{t_0}^t a(s) ds, \quad (1.2)$$

$$x(t) = x(t_0) + \int_{t_0}^t v(s) ds. \quad (1.3)$$

Aby bylo možné s inerciální jednotkou volně pohybovat v prostoru, je potřeba kromě znalosti polohy měřit, nebo kompenzovat její natočení. Jednou z možností jak kompenzovat rotaci jednotky je připevněním akcelerometrů na gimbal, který bude udržovat jejich natočení vůči Zemi konstantní. Tohoto principu se často využívá v letectví, zejména kvůli jejich vysoké přesnosti, ovšem velkou nevýhodou bývá mechanická složitost a velikost. [2] [3]

Druhou možností, jak kompenzovat natočení je měřit jeho úhel a následně zrychlení z akcelerometru rotovat vůči referenčnímu systému. [2] [3] K tomuto účelu slouží gyroskopy, které měří úhlovou rychlost ω otáčení jednotky kolem osy. Podobně jako se zrychlením u akcelerometru, znalostí počátečního úhlu $\varphi(t_0)$ v čase t_0 můžeme pomocí úhlové rychlosti ω v časech $s > t_0$ určit úhel natočení $\varphi(t)$, ovšem tentokrát pouze jednou integrací

$$\varphi(t) = \varphi(t_0) + \int_{t_0}^t \omega(s) ds. \quad (1.4)$$

Díky tomu můžou být gyroskopy a akcelerometry nepohyblivě připevněny na mechanickou konstrukci. Jde o tzv. „Strapdown“ typ inerciální navigace.

1.2.1 Zavedení vztažných soustav

Pro účely přehlednosti a exaktnosti bývá v oblastech inerciální navigace zavedeno několik kartézských vztažných soustav. Každá soustava je ortogonální a pravotočivá. [4] [1]

- Inertial frame (i-frame) má počátek ve středu Země. Její osy jsou pevné vůči nepohybujícím se hvězdám. Osa z_i prochází zemskou osou.
- Earth frame (e-frame) má také počátek ve středu Země, její osy jsou pevně vztažené vůči Zemi, tedy rotují kolem i-frame. Osa z_e prochází zemskou osou.
- Navigation frame (n-frame) má počátek ve výchozím bodě navigace. Osy jsou natočené ve směrech sever, východ, dolů (*North East Down*, NED).
- Body frame (b-frame) má počátek v inerciální jednotce a její osy jsou natočené ve směrech náklonu, stáčení a vychýlení jednotky.

S následnými měřeními a vypočtenými daty je často manipulováno jako s vektory označenými indexy odpovídajícím soustavě, ke které jsou vztaženy (i, e, n, b).

1.3 Vztažné soustavy a rotace Země

Pokud budeme chtít provozovat navigaci vztaženou k pevnému počátečnímu bodu v prostoru (i-frame), bude možné používat klasické pohybové rovnice, tedy pro zrychlení \vec{a}_i , polohu \vec{r} a čas t platí [1]

$$\vec{a}_i = \left. \frac{d^2 \vec{r}}{dt^2} \right|_i \quad (1.5)$$

a obdobně pro rychlost \vec{v}_i platí

$$\vec{v}_i = \left. \frac{d\vec{r}}{dt} \right|_i. \quad (1.6)$$

Ovšem v mnoha případech použití inerciální navigace potřebujeme jako počátek zvolit nějaký bod na Zemi, která se neustále otáčí kolem své osy, v tomto případě tedy budeme využívat e-frame. Pokud máme data pouze ze vztažné soustavy i-frame, bude potřeba započíst úhlovou rychlost Země $\vec{\omega}_{ie}$ [1]

$$\vec{\omega}_{ie} = \begin{pmatrix} 0 \\ 0 \\ \Omega \end{pmatrix}, \quad (1.7)$$

kde $\Omega = 7,292 \times 10^{-5} \text{ rad}\cdot\text{s}^{-1}$ je rychlost rotace Země kolem své osy, vztaženo vůči nejbližší hvězdě (Slunci). Poté můžeme rychlost \vec{v}_e v e-framu definovat jako [1] [2]

$$\vec{v}_e = \left. \frac{d\vec{r}}{dt} \right|_e = \vec{v}_i - \vec{\omega}_{ie} \times \vec{r}. \quad (1.8)$$

S přepočty mezi i a e frame je potřeba nakládat zejména při navigaci letadel a raket, kdy je potřeba započíst i rotaci Země, vzhledem k dlouhým úsekům času i vzdálenosti. V této práci se ovšem zabýváme navigací na malé vzdálenosti, kdy doba měření může být řádově v jednotkách minut a vzdálenost ve stovkách metrů. Vliv rotace Země tedy bude velmi malý.

1.4 Tíhové pole Země

Abychom mohli popsat chování jednotky v i-frame, označíme dynamickou sílu způsobenou zrychlením na jednotku hmotnosti jako tzv. „specifickou sílu“ \vec{f} představující sílu na jednotku hmotnosti, tedy samotné zrychlení. S její pomocí můžeme rovnici 1.5 upravit na [1] [2]

$$\vec{a}_i = \left. \frac{d^2 \vec{r}}{dt^2} \right|_i = \vec{f} + \vec{g}, \quad (1.9)$$

kde \vec{g} je tíhové zrychlení pole Země. Z tohoto poznatku vyplývá, že pro správné fungování inerciální navigace je potřeba přesně znát velikost a směr \vec{g} .

Vzhledem k tomu, že Země: [5]

- rotuje kolem své osy,

- není koule,
- není homogenní,

není možné považovat velikost tíhového zrychlení za konstantní a jeho směr vždy do středu Země.

Rotace Země kolem své osy vyvolává dostředivé zrychlení.¹ Toto zrychlení má vždy kolmý směr k ose otáčení Země. Vektorovým součtem s gravitačním zrychlením \vec{a}_g , které má vždy směr do středu Země dostaneme tíhové zrychlení \vec{g} , proto se v závislosti na zeměpisné šířce tíhové zrychlení odchyluje od středu Země. [5]

Pro kompenzaci rotace Země bychom mohli vyjádřit tíhové zrychlení \vec{g} v závislosti na gravitačním zrychlení \vec{a}_g a následně využít v rovnici 1.9 [1]

$$\vec{g} = \vec{a}_g - \vec{\omega}_{ie} \times (\vec{\omega}_{ie} \times \vec{r}). \quad (1.10)$$

Díky tomu, že Země není koule, ale přibližně elipsoid, jsou body na pólech blíže středu Země než body na rovníku, to tedy ovlivňuje velikost gravitačního zrychlení. Tento vliv by šel analyticky vypočítat, pokud bychom předpokládali, že Země je elipsoid a zanedbali vliv reliéfu povrchu Země. [5]

Vzhledem k tomu, že tato práce se zabývá zpracováním dat z inerciální navigace až tzv. offline, tedy ne v reálném čase, můžeme pro řešení problémů s nekonzistentním tíhovým polem Země použít některý z dostupných gravitačních modelů Země (*Earth Gravitational Model*, EGM).

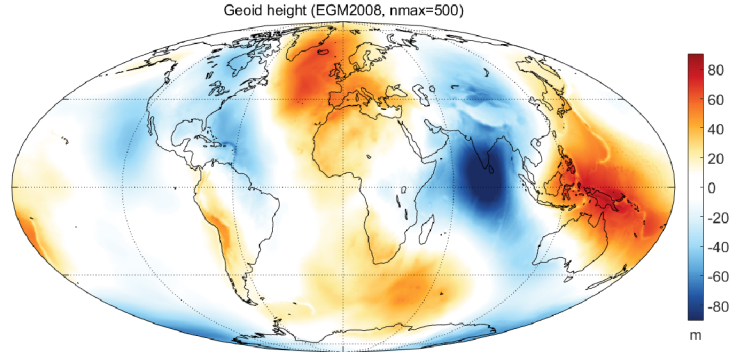
1.4.1 Gravitační modely Země

EGM jsou modely popisující geoid Země s velmi širokou škálou použití v oblastech fyziky, geodézie, oceánografie, navigace a dalších. Spravuje je americká geoprostorová agentura (*National Geospatial- Intelligence Agency*, NGA) a jsou volně dostupné. Nejnovější veřejně dostupný EGM je z roku 2008 (EGM2008), který byl vytvořen na základě několika pozemních a výškových měření. [6]

Vytvořená mapa s výškou geoidu modelu EGM2008 publikovaného NGA je znázorněna na obrázku 1.1

Takovýto model je poté možné použít například pomocí funkcí z Matlab Aerospace Toolbox, který vypočítá vektor gravitačního zrychlení a_g se správným směrem i velikostí na základě dodaný souřadnic systému WGS84. Poté pomocí rovnice 1.10 můžeme vypočítat tíhové zrychlení.

¹Všude na Zemi, vyjma samotných zeměpisných pólů



Obr. 1.1: Výškový reliéf geoidu. [7]

1.5 Rotace měření akcelerometru

Vzhledem k tomu, že se tato práce zabývá strapdown systémem inerciální navigace, jsou data akcelerometru vztažena k soustavě inerciální jednotky (tedy b-frame). Pro převedení dat do vztažné soustavy i-frame je potřeba rotovat vektor specifické síly \vec{f} maticí směrových kosínů \mathbf{C}_b^i . Tato rotace by se dala popsat jako 3 po sobě jdoucí natočení o tzv. Eulerovy úhly vůči referenčním osám (např. pro tento případ i-frame). Tyto úhly označíme ϕ , θ a ψ . [1]

- Rotaci o úhel ϕ kolem osy x přiřadíme rotační matici \mathbf{C}_x [1]

$$\mathbf{C}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}. \quad (1.11)$$

- Rotaci o úhel θ kolem osy y přiřadíme rotační matici \mathbf{C}_y [1]

$$\mathbf{C}_y = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}. \quad (1.12)$$

- Rotaci o úhel ψ kolem osy z přiřadíme rotační matici \mathbf{C}_z [1]

$$\mathbf{C}_z = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.13)$$

Součinem těchto tří matic získáme matici směrových kosínů \mathbf{C}_b^i pro převod z b-frame na i-frame [1]

$$\mathbf{C}_b^i = (\mathbf{C}_i^b)^T = (\mathbf{C}_x \mathbf{C}_y \mathbf{C}_z)^T = \mathbf{C}_z^T \mathbf{C}_y^T \mathbf{C}_x^T, \quad (1.14)$$

$$\mathbf{C}_b^i = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix}. \quad (1.15)$$

Toto poté můžeme použít k rotaci samotného vektoru specifické síly \vec{f} z b-frame na i-frame

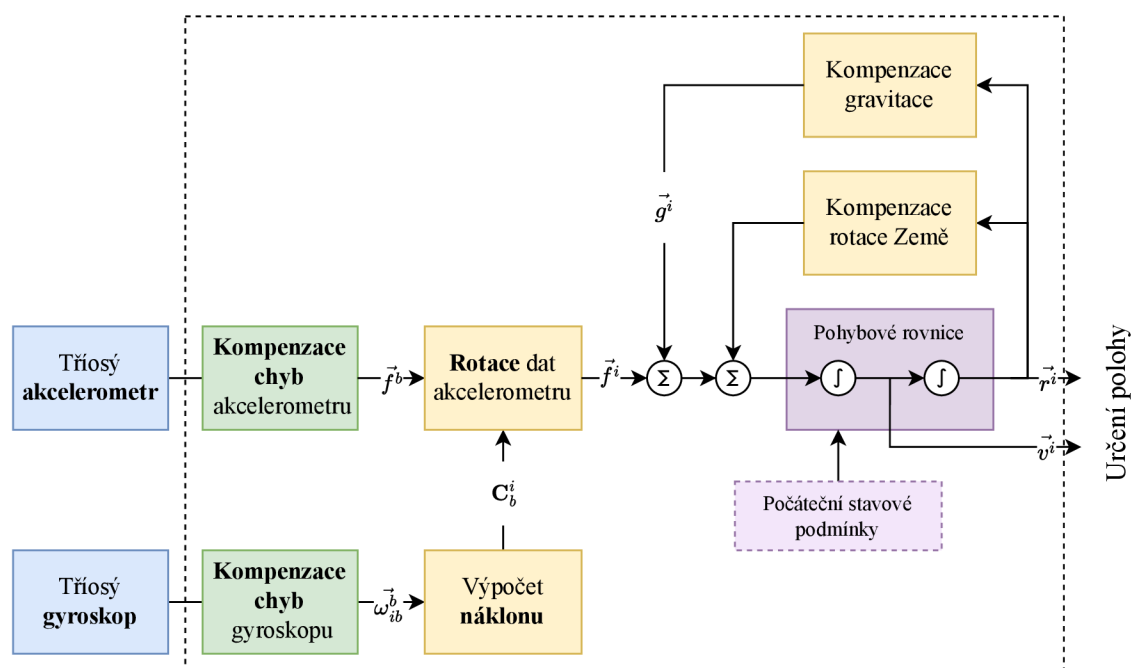
$$\vec{f}^i = \mathbf{C}_b^i \vec{f}^b. \quad (1.16)$$

Kombinací s rovnicí 1.9 získáme rovnici popisující pohyb

$$\vec{a}_i = \left. \frac{d^2 \vec{r}^i}{dt^2} \right|_i = \vec{f}^i + \vec{g}^i = \mathbf{C}_b^i \vec{f}^b + \vec{g}^i. \quad (1.17)$$

1.6 Diagram algoritmu inerciální navigace

Pro přehlednější znázornění algoritmu základní formy inerciální navigace pro 6 stupňů volnosti (*Degrees of Freedom*, DoF) inerciální měřicí jednotky (*Inertial Measurement Unit*, IMU) byl vytvořen blokový diagram (obrázek 1.2).



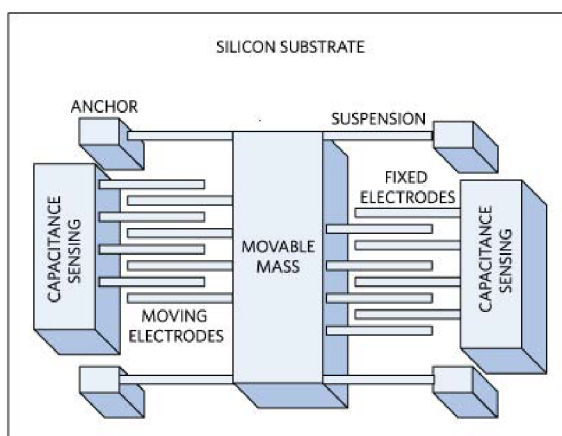
Obr. 1.2: Blokové schéma algoritmu strapdown inerciální navigace, převzato z [1] [2].

2 Inerciální senzory

Ke sběru dat pro účely inerciální navigace se jako první začaly používat a stále používají např. v letectví složité, velké a drahé mechanické přístroje se setrvačníky poháněnými motory využívající gyroskopický efekt. Jejich natočení os vůči referenčnímu rámu je poté možné měřit například laserem, či jiným způsobem. Tyto senzory jsou díky jejich přesnosti stále využívány v aplikacích, kde je kladen důraz na opravdu velkou spolehlivost a přesnost. Ovšem díky zlepšujícím se technologiím se začínají hojně využívat i MEMS IMU v oblastech inerciální navigace, zejména tam, kde nejsou kladeny velmi přísné požadavky na přesnost a spolehlivost, jako je například navigace uvnitř budov. Díky MEMS technologiím je možné sestavit celou inerciální jednotku bez jakýchkoliv pohyblivých částí, velmi malou a poměrně levnou. [1] [2]

2.1 MEMS akcelerometry

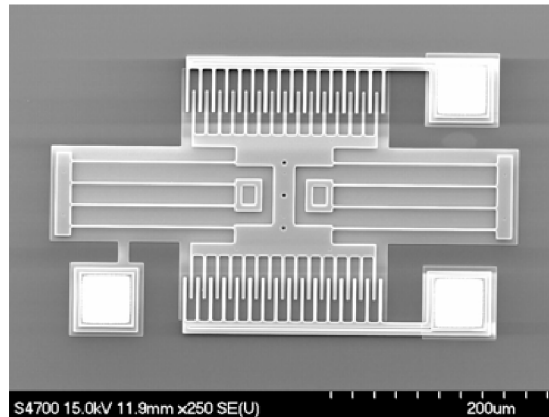
Akcelerometr měří lineární zrychlení nepřímo díky druhému Newtonovu zákonu (rovnice 1.1). Přímo na struktuře křemíku je těleso známé hmotnosti, kterému je umožněn pružný pohyb v nějaké z os. Pokud na akcelerometr působí zrychlení, je těleso vychylováno silou, která je následně měřena. Tato síla je často měřena jako změna kapacity mechanické struktury znázorněné na obrázku 2.1 a 2.2, převedena na napětí, zesílena, vyfiltrována a následně převedena analogově-digitálním převodníkem (*Analog to Digital Converter*, ADC). Tyto struktury jsou následně uspořádány ortogonálně, aby vytvořily tříosý akcelerometr. [8]



Obr. 2.1: Struktura MEMS akcelerometru. [8]

Za povšimnutí stojí rozdílné parametry např. biasu pro různé osy IMU v tabulce 3.1, to může být způsobené vrstvením struktury během výroby čipu, kdy dvě z os

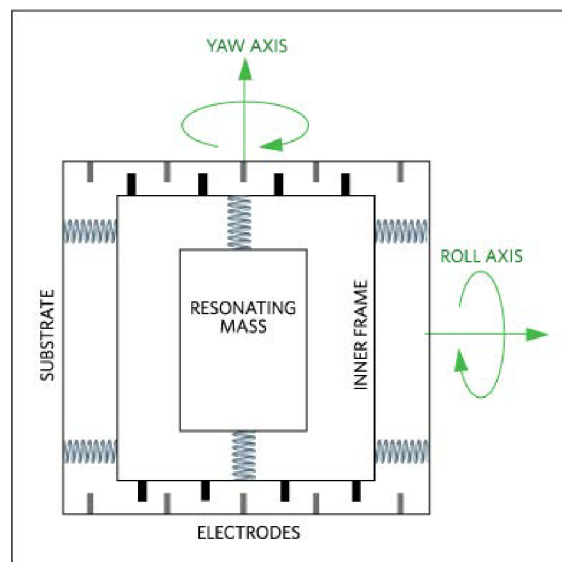
akcelerometru mají osy vychýlení hmoty rovnoběžnou se směry vrstev, zatímco třetí osa je na tyto vrstvy kolmá.



Obr. 2.2: Mikroskopický snímek MEMS akcelerometru. [9]

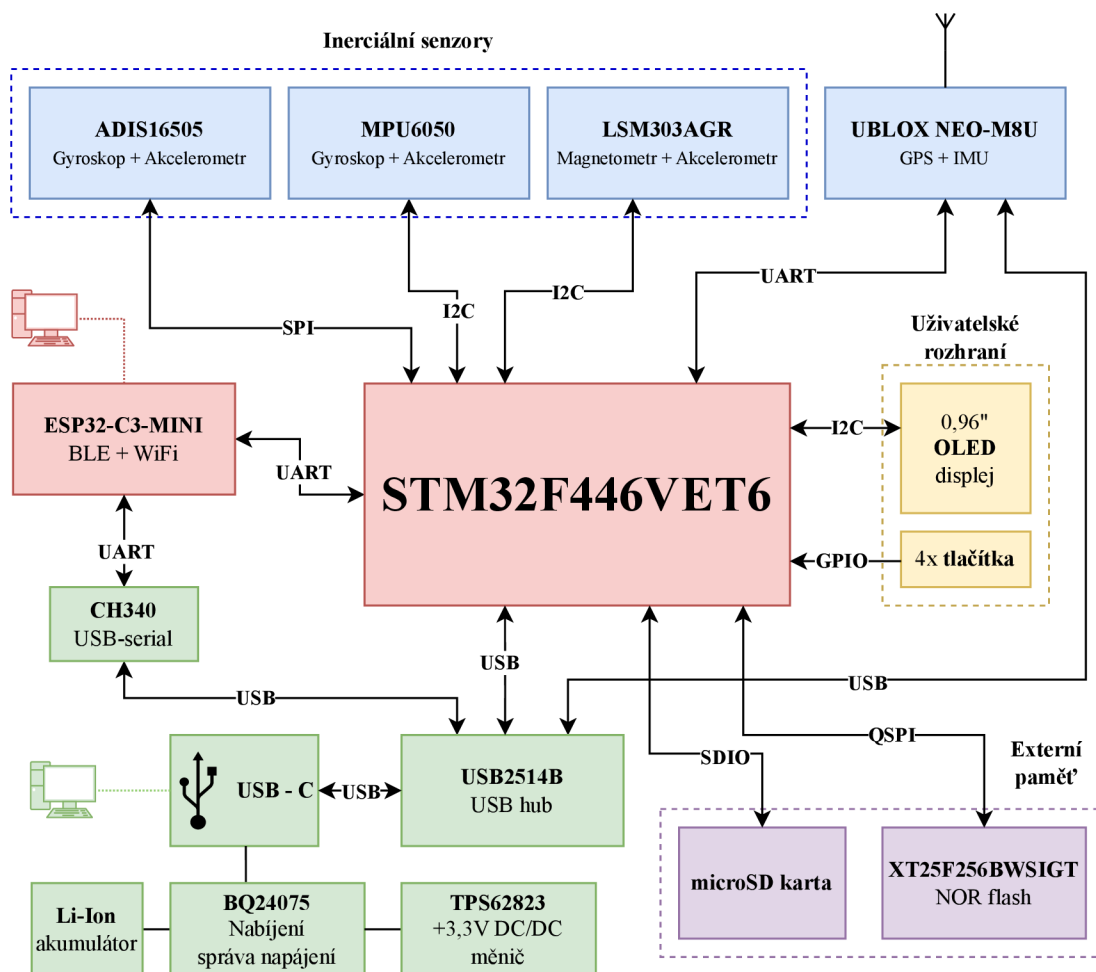
2.2 MEMS gyroskopy

Gyroskopy jsou senzory, které měří úhlovou rychlost. Uspořádání mechanické struktury (na obrázku 2.3) je rozdílné v porovnání s akcelerometry. Zde je tělesu definované hmotnosti umožněno kmitat v jedné z os. Při působení úhlové rychlosti na gyroskop je vnitřní struktura díky Coriolisově síle vychýlena, což způsobí změnu kapacity. Ta je dále zpracována obdobně jako v akcelerometru (kapitola 2.1). [1] [8]



Obr. 2.3: Struktura MEMS gyroskopu. [8]

3 Hardware inerciální jednotky



Obr. 3.1: Blokové schéma inerciální jednotky.

Hardware inerciální jednotky je realizován tak, aby umožňoval zaznamenávat hodnoty změřené inerciálními senzory a poskytovat dohromady devítiosá data (akcelerometr, gyroskop a magnetometr). Jednotka také obsahuje GPS modul s vestavěným IMU, jehož použití by mohlo být vhodné například v prostorech s alespoň částečným pokrytím signálu GPS.

Naměřená data je možné uložit do externí NOR Flash paměti připojené k mikrokontroléru (*Microcontroller Unit*, MCU), popřípadě lze využít i kartu typu microSD. Konektor univerzální sériové sběrnice (*Universal Serial Bus*, USB) typu C umožňuje nabíjení vestavěného Li-Ion akumulátoru jednotky a komunikaci mezi PC a ESP32, GPS modulem a hlavním MCU skrze vestavěný USB rozbočovač. K přenosu dat pro jejich následné zpracování v PC primárně slouží USB rozhraní, ale zařízení disponuje i bezdrátovým modulem ESP32-C3, umožňující komunikaci přes WiFi, nebo Bluetooth.

Pro jednoduchou volnost pohybu je jednotka napájena jedním Li-Ion akumulátorem velikosti 18650, při záznamu dat tedy nebude potřeba externího zdroje energie. Grafický organický LED (*Organic Light-Emitting Diode*, OLED) displej a 4 tlačítka slouží jako uživatelské rozhraní při používání jednotky.

3.1 Akcelerometr a gyroskop

Jednotka obsahuje dvě šestiosá IMU (gyroskop s akcelerometrem) rozdílných parametrů a řádově rozdílné ceny. Takto odlišné součástky byly vybrány proto, aby bylo možné porovnat vliv přesnosti, šumu, biasu a driftu senzorů na následně zpracovaná data. V tabulce 3.1 jsou porovnány důležité parametry senzorů MPU6050 a ADIS16505-2. Pro účely inerciální navigace je důležitý zejména nízký bias a drift senzorů, aby při integraci dat k vyhodnocení polohy nebyla integrována i driftová chyba, což má za výsledek velmi nepřesné zpracování hodnot. [10]

Integrovaný obvod MPU6050 je standardní šestiosé MEMS IMU, vhodné mimo jiné pro použití v mobilních zařízeních a dalších podobných aplikacích. Jeho vnitřní gyroskop a akcelerometr má softwarově přepínatelné rozsahy měřených veličin. Kromě inerciálních senzorů má i vestavěný signálový procesor pro fúzi a filtrování dat přímo v integrovaném obvodu. Tato funkce může být vhodná pro odlehčení výpočetního výkonu hlavního procesoru, ovšem pro účely této práce nebude signálový procesor využit, jelikož se měřená data budou zpracovávat až po jejich naměření v PC, ne v reálném čase. Vzorkovací frekvence gyroskopu je 8 kHz a akcelerometru 1 kHz, oba senzory mají 16bitové rozlišení. [11]

MPU6050 disponuje rozhraním mezi-obvodové komunikace (*Inter-Integrated Circuit*, I2C) s maximální frekvencí hodinového signálu 400 kHz. [11] Pokud bychom chtěli vyčítat ze senzoru data při maximální možné vzorkovací frekvenci, byla by potřeba minimální přenosová rychlost sběrnice

$$f_{\text{mathrmclk}} = 3 \text{ osy} \cdot (f_{\text{gyro}} + f_{\text{acc}}) \cdot (16\text{bitů} + 2 \cdot \text{ACK}) = 3 \cdot (8000 + 1000) \cdot (16 + 2) = 486 \text{ kHz.} \quad (3.1)$$

Při vyčítání dat o maximální vzorkovací frekvenci jsme omezeni samotným I2C rozhraním senzoru (využití maximální vzorkovací frekvence je teoreticky možné krátkodobě, pomocí interního 1kB FIFO zásobníku). [11]

Jelikož pro účely inerciální navigace stačí vzorkovací frekvence dat v řádu stovek hertz [12], tak není tato limitace omezující. Senzor je propojen s hlavním MCU přes I2C sběrnici s frekvencí hodinového signálu 400 kHz a není sdílena s žádným jiným zařízením, aby bylo možné, v případě potřeby, využít maximální dostupný potenciál senzoru (i přestože je reálná potřeba vzorkovací frekvence nižší).

Tab. 3.1: Porovnání základních parametrů gyroskopů. [11] [13]

Model IMU	MPU6050	ADIS16505-2	jednotka
Parametry gyroskopů			
Dynamický rozsah	programovatelný, ±250, ±500, ±1000, ±2000	±500	°/s
Citlivost ¹	65, 5	2621440	LSB/(°/s)
Bias v ose x a z	±20	±0, 14	°/s
Bias v ose y	±20	±1, 4	°/s
Efektivní hodnota hustoty šumu při 10Hz pro osy x a y	0,005	0,0043	°/s/√Hz
Efektivní hodnota hustoty šumu při 10Hz pro osu z	0,005	0,0034	°/s/√Hz
Parametry akcelerometrů			
Dynamický rozsah	programovatelný, ±19, 6, ±39, 2, ±78, 4, ±156, 8	±78, 4	m/s ²
Citlivost ²	418	26756268	LSB/(m/s ²)
Bias v ose x a y	±0, 491	±0, 0196	m/s ²
Bias v ose z	±0, 785	±0, 0196	m/s ²
Efektivní hodnota hustoty šumu při 10Hz pro osy x a y	3924	167	μm/s ² /√Hz
Efektivní hodnota hustoty šumu při 10Hz pro osu z	3924	243	μm/s ² /√Hz

Integrovaný obvod ADIS16505-2 je precizní šestiosé MEMS IMU, vhodné pro použití v průmyslových a navigačních aplikacích s poměrně nízkým driftem a vysokou přesností. Na rozdíl od MPU6050 nemá přepínatelný dynamický rozsah, je fixně daný variantou součástky. Vzorkovací frekvence gyroskopu i akcelerometru je 2 kHz, oba senzory mají 32bitové rozlišení. S hlavním MCU komunikuje přes sběrnici sériového periferního rozhraní (*Serial Peripheral Interface*, SPI) s maximální frekvencí hodinového signálu 2,1 MHz. [13] Pokud budeme chtít vyčítat data ze senzoru při maximální možné vzorkovací frekvenci, bude potřeba minimální přenosová rychlost

¹Pro porovnání citlivosti byl vybrán dynamický rozsah 500 °/s senzoru MPU6050 pro možnost porovnání hodnoty s druhým senzorem

²Pro porovnání citlivosti byl vybrán dynamický rozsah 78,4 m/s² senzoru MPU6050 pro možnost porovnání hodnoty s druhým senzorem

sběrnice

$$f_{\text{clk}} = 3 \text{ osy} \cdot (f_{\text{gyro}} + f_{\text{acc}}) \cdot 32 \text{ bitů} = 3 \cdot (2000 + 2000) \cdot 32 = 384 \text{ kHz.} \quad (3.2)$$

Nejsme tedy omezeni maximálními frekvencemi hodinového signálu a můžeme teoreticky využívat senzor i při nejvyšší možné rychlosti.

Výrobce prodává tento obvod ve variantě stopinového pouzdra matice kuliček (*Ball Grid Array*, BGA) čipu, ale i jako vývojovou desku osazenou senzorem a kolíkovou lištou (obrázek 3.2) pro jednodušší práci s osazením desky plošných spojů (DPS). [13] Hardware jednotky byl navržen tak, aby bylo možné využít jak samotný BGA čip, tak i hotový modul s konektorem.



Obr. 3.2: IMU verze ADIS16505-2/PCBZ.

3.2 Magnetometr

Vzhledem k tomu, že výběr komerčně dostupných devítiosých senzorů (akcelerometr, gyroskop a magnetometr) je značně omezený, popřípadě součástky prodávané jako devítiosé IMU jsou ve skutečnosti moduly více součástí na jedné desce, tak je ve výsledném obvodovém zapojení použit senzor magnetické indukce jakožto samostatná součástka.

Přestože fúze dat z magnetometru může mít pozitivní dopady na zmenšení chyby trajektorie [14], jeho použití uvnitř budov je značně omezené vzhledem k jednoduše ovlivnitelnosti měření blízkými feromagnetickými látkami, silovými rozvody elektřiny a pod. Proto nebyly na výběr magnetometru kladeny vysoké požadavky a slouží spíše pro porovnání vlivu přítomnosti / absence naměřených dat z tohoto senzoru.

K tomuto účelu byl vybrán běžně dostupný obvod LSM303AGR, který kromě magnetometru v pouzdře obsahuje i akcelerometr, ten ovšem nebude pro potřeby práce využit, jelikož tuto funkci obstarávají akcelerometr a gyroskop z kapitoly 3.1.

Magnetometr komunikuje s hlavním MCU přes sběrnici I2C s maximální vzorkovací frekvencí 150 Hz, dynamickým rozsahem $\pm 4,915$ mT a 16bitovým rozlišením. [15]

3.3 GNSS

Zajímavou a uživatelsky přívětivou kombinaci GNSS a inerciální navigace poskytuje například firma u-blox s řadou modulů podporující funkci „dead reckoning“. Jedná se o navigační moduly s vestavěným IMU, určené zejména do oblasti automobilového průmyslu. Jejich typický příklad použití, dle výrobce, je navigace aut, kdy při běžném provozu je zafixovaný signál z GNSS a při výpadku signálu (vjezd do garáže, tunelu apod.) je navigace modulem stále poskytována na základě dat z IMU. [16]

Navigační modul u-blox NEO-M8U byl vybrán a implementován do obvodového zapojení inerciální navigační jednotky. Výrobce udává, že modul zvládne odhadovat polohu po ztrátě signálu GNSS po dobu 60 s s typickou odchylkou 10 % trajektorie. Dále také modul při zapnutí odpovídající funkce umí využít interní IMU ke zvýšení maximální rychlosti aktualizace polohy až na 30 Hz. Jeho využití v rámci této práce může být různé, například pro navigaci v místech s alespoň částečným pokrytím signálu GNSS. [16]

NEO-M8U umí využívat všechny světové navigační systémy (uvedeny v tabulce 3.2) Tento modul komunikuje s hlavním MCU přes sběrnici univerzálního asynchronního

Tab. 3.2: Podporované družicové systémy. [16]

GNSS systém	Pásmo	Frekvence (MHz)
GPS	L1C/A	1575,42
GLONASS	L1OF	1602
BeiDou	B1	1561,098
Galileo	E1-B/C	1575,42

přijímače-vysílače (*Universal Asynchronous Receiver-Transmitter*, UART), pomocí standardizovaných příkazů americké organizace námořní elektroniky (*National Marine Electronics Association*, NMEA) příkazů v textové podobě, nebo pomocí binárního protokolu *UBlox message - zpráva formátu UBLOX* (UBX), který je specifikován výrobcem. Použití protokolu NMEA je omezené pouze na standardní funkce

GNSS modulů, pokud chceme využít speciálních funkcí, například inerciální navigace, je nutné použít proprietární protokol UBX. [16] NEO-M8U také disponuje USB portem, skrz který je možné modul ovládat a konfigurovat pomocí PC aplikace výrobce. Tento port je připojen na integrovaný USB rozbočovač a lze jej využít například pro vývojové účely.

3.4 Paměť

Tab. 3.3: Odhad celkového datového toku pro záznam dat.

Senzor	Odhadovaný datový tok
ADIS16505-2	375 kbit/s
MPU-6050	422 kbit/s
LSM303AGR	7 kbit/s
NEO-M8U	1 kbit/s
Celkem	805 kbit/s (0,1MB/s)

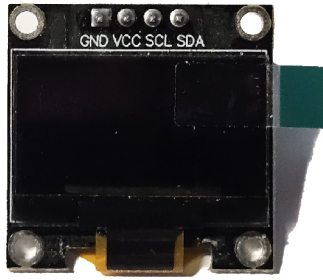
V případě, že bychom chtěli zaznamenávat data ze všech senzorů při jejich maximálních vzorkovacích frekvencích, nebude množství změřených dat zanedbatelné. V tabulce 3.3 je hrubý odhad potřebné rychlosti záznamu dat pro tento krajní případ. Pokud bude měření trvat např. 2 minuty, vygenerujeme dohromady 12 MB dat, což převyšuje velikost paměti většiny dostupných MCU.

Z tohoto důvodu je v obvodovém zapojení inerciální jednotky implementována 32MB NOR Flash paměť, propojená s hlavním MCU přes sběrnici QUADSPI s maximální možnou hodinovou frekvencí 120 MHz. Měla by tedy být pro potřeby této aplikace z pohledu velikosti paměti a datového toku dostačující. [17]

Kromě výše popsání Flash paměti jednotka obsahuje i slot na microSD kartu, která by z uživatelského hlediska mohla být jednodušší k použití, ovšem při zápisu může latence SD karty být (krátkodobě) až stovky milisekund. [18] To by mohlo znemožnit jejím použití v případě, že by hlavní MCU měl nedostatek volné paměti RAM pro krátkodobé uchování dat, proto bude o její využití rozhodnuto až v následujících kapitolách.

3.5 Uživatelské rozhraní

Pro ovládání uživatelem disponuje jednotka grafickým OLED displejem s úhlopříčkou 0,96 palce a rozlišením 128 × 64 pixelů, který je připojený přes sběrnici I2C.



Obr. 3.3: Fotografie grafického OLED displeje.

Společně s 4 tlačítky by měl poskytnout dostatečně univerzální a pohodlné uživatelské rozhraní.

3.6 Napájení

Inerciální jednotka je napájena z jednoho Li-Ion akumulátoru velikosti 18650. Nabíjení je realizováno obvodem BQ24075RGT, který monitoruje nabíjecí odebraný proud jednotkou. Proud, kterým je nabíjen akumulátor je regulován tak, aby nepřekročil maximální hranici 900 mA z USB portu. [19]

Všechny součásti inerciální jednotky (až na obvod reálného času (*Real Time Clock*, RTC) a zálohovací registry hlavního MCU a GPS modulu) jsou napájeny skrz DC/DC měnič z výstupního vývodu tohoto nabíjecího obvodu. V případě, že je připojena jednotka do USB a nabíjí se, na výstupním pinu nabíjecího obvodu je napájecí napětí USB portu. Díky tomu nedochází k velkým ztrátám pokud je jednotka zapnuta a nabíjí se zároveň. Jestliže je USB odpojeno, skrz interní tranzistor je jednotka napájena z akumulátoru. [19]

Nabíjecí obvod také umožňuje kompletní odpojení napájení jednotky přes jeden z vývodů. Toho je využito pro ochranu akumulátoru proti podvybití pomocí zapojení S/R klopného obvodu na napájení USB a jednoho z výstupů procesoru. Napětí akumulátoru je měřeno pomocí ADC mikrokontroléru. Jestliže klesne pod definovanou úroveň, pomocí pulzu bude celý obvod odpojen od napájení až do té doby, dokud uživatel znovu nepřipojí jednotku do USB portu.

Vzhledem k většímu počtu součástek není odebraný proud z 3,3V napájecí větve malý (zhruba 0,5 A, viz. tabulka 3.4). Budeme-li uvažovat rozsah výstupního napětí nabíjecího obvodu 3,5 V (vybitý akumulátor) až 5 V (zařízení připojené do USB) zjistíme, že pro napájení 3,3V větve není vhodný lineární regulátor, zejména kvůli vysokému ztrátovému výkonu. Ten je v krajním případě

$$P_{\text{ztrátový}} = (U_{\text{USB}} - U_{\text{IO}}) \cdot I_{\text{IO}} = (5 - 3,3) \cdot 0,5 = 0,85 \text{ W.} \quad (3.3)$$

Tab. 3.4: Odhad spotřeby proudu 3,3V větve.

Součástka	Odhadovaný proud (mA)
STM32F446	50
ESP32	150
USB2514	135
ADIS16505	50
NEO-M8U	30
OLED displej	10
microSD karta	50
Celkem	475

Proto byl na napájení hlavní 3,3V větve vybrán spínaný regulátor TPS62823. Jedná se o buck (snižující) měnič s integrovaným výkonovým tranzistorem pracujícím na frekvenci 2,2 MHz. Díky vyšší spínací frekvenci je možné využít menší komponenty, zejména cívku a filtrační kondenzátory na výstup, ovšem je potřeba dodržet doporučená pravidla při návrhu desky pro omezení rušení a velkých proudových smyček. Rozsah napájecího napětí čipu je 2,4 až 5 V, maximální výstupní proud 3 A. [20]

Minimální napětí, na které můžeme nechat akumulátor vybit je dáno odpory přechodů Drain-Source vnitřních tranzistorů nabíjecího obvodu, DC/DC měniče a stejnosměrným odporem cívky. V tomto případě bude regulátor pracovat v módu s minimální střídou. [20] Toto napětí je

$$\begin{aligned}
 U_{\text{batMin}} &= U_{\text{out}} + I_{\text{out}} \cdot (R_{\text{DS}(\text{charge})} + R_{\text{DS}(\text{conv})} + R_{\text{DC}(\text{L})}) = \\
 &= 3,3 + 0,5 \cdot (0,05 + 0,026 + 0,014) = 3,345 \text{ V}
 \end{aligned}
 \tag{3.4}$$

3.7 Hlavní procesor

Požadavky na výběr hlavního procesoru byly z velké části dané počtem a druhem potřebných periférií, které jsou popsány v tabulce 3.5. Dále byly z podskupiny procesorů disponujících všemi perifériemi z tabulky 3.5 vybrány takové, které mají velikost vnitřní FLASH paměti alespoň 512 kB, abychom nebyli při vývoji Firmware jednotky omezeni velikostí programu. Pouzdra procesorů byla vybrána taková, aby se s nimi dalo jednoduše pracovat, z toho důvodu byla vyloučena pouzdra typu BGA. V neposlední řadě byla zvažována i dostupnost vybíraných procesorů u nejobvyklejších distributorů elektronických součástek, aby bylo možné v případě potřeby výrobu jednotky opakovat.

Tab. 3.5: Minimální požadavky na periferie mikroprocesoru.

Druh periferie	Minimální požadovaný počet	Použití periferie
I2C	3	OLED displej, LSM303AGR, MPU6050, USB2514B
SPI	1	ADIS16505
UART	2	NEO-M8U, ESP32
QUADSPI	1	NOR FLASH paměť
SDIO	1	microSD karta
ADC	1	měření napětí akumulátoru

Na základě těchto požadavků byl vybrán mikrokontrolér *STM32F446VET6*. Jedná se o 32bitový ARM Cortex-M4 procesor z portfolia „high performance“ mikrokontrolérů výrobce STMicroelectronics. Splňuje všechny výše zmíněné minimální požadavky, v obvodovém zapojení byla použita i USB periferie procesoru, která může mít různá využití. Procesor obsahuje 512 kB paměti Flash a 128 kB paměti RAM, maximální hodinová frekvence je 180 MHz a disponuje matematickým koprocесorem pro operace s plovoucí desetinou čárkou. Vzhledem k počtu univerzálních vstupně/výstupních pinů (*General Purpose Input/ Output*, GPIO) v zapojení inerciální jednotky byla vybrána varianta procesoru v pouzdře LQFP100. [21]

3.8 ESP32

Pro splnění požadavků zadání práce je potřeba, aby mohla inerciální jednotka komunikovat bezdrátově s PC zpracovávajícím data. Pro tento úkol byl vybrán bezdrátový modul ESP32-C3-Mini. Jedná se o jeden z novějších produktů portfolia bezdrátových modulů firmy Espressif. Podporuje standard WiFi 802.11 b/g/n a Bluetooth LE 5. [22]

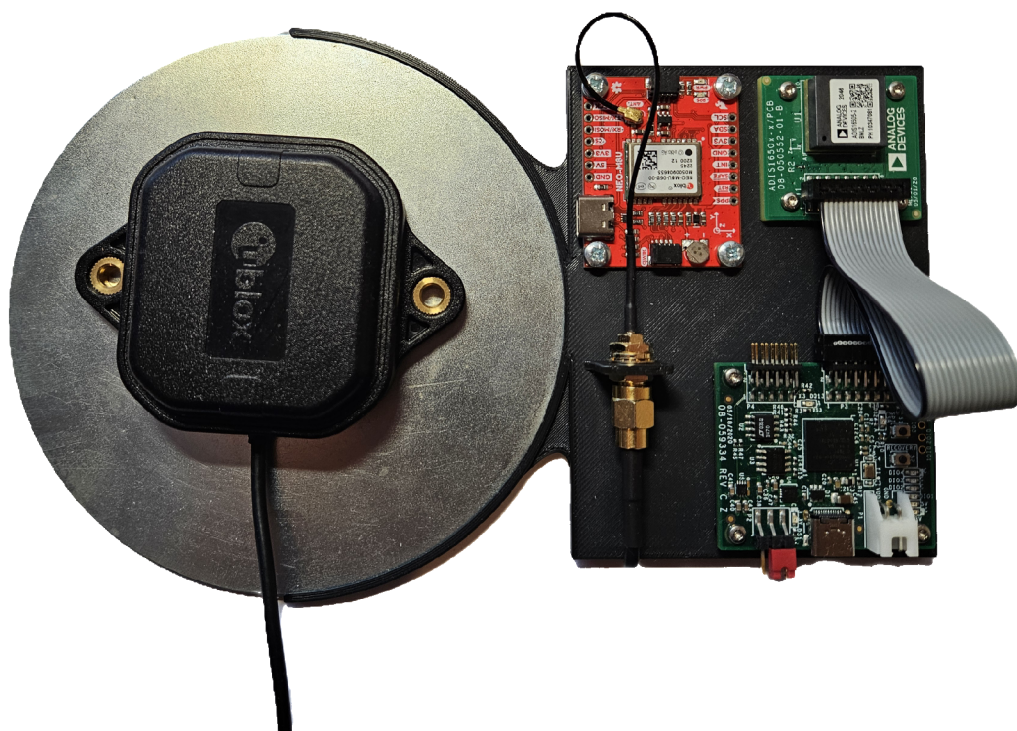
Tento modul je v obvodovém zapojení použit čistě jako bezdrátové rozhraní, neobsahuje žádné další GPIO kromě 2 UART sběrnic. První sběrnice UART je připojena pomocí USB-serial převodníku CH340 na USB rozbočovač v inerciální jednotce. Toto rozhraní slouží pro nahrávání, popřípadě aktualizaci vestavěného AT firmwaru výrobce. V případě, že by poskytovaný firmware výrobce nedostačoval, nebo nebyl vhodný pro potřeby naší aplikace, bude možné pomocí tohoto rozhraní nahrát vlastní obslužný firmware pro ESP32.

Druhá sběrnice UART je připojena k hlavnímu MCU inerciální jednotky. Kromě standardních pinů Rx a Tx jsou propojeny i piny pro řízení toku, které by bylo možné použít na zjednodušení časování komunikace.

3.9 Testování s vývojovými stavebnicemi

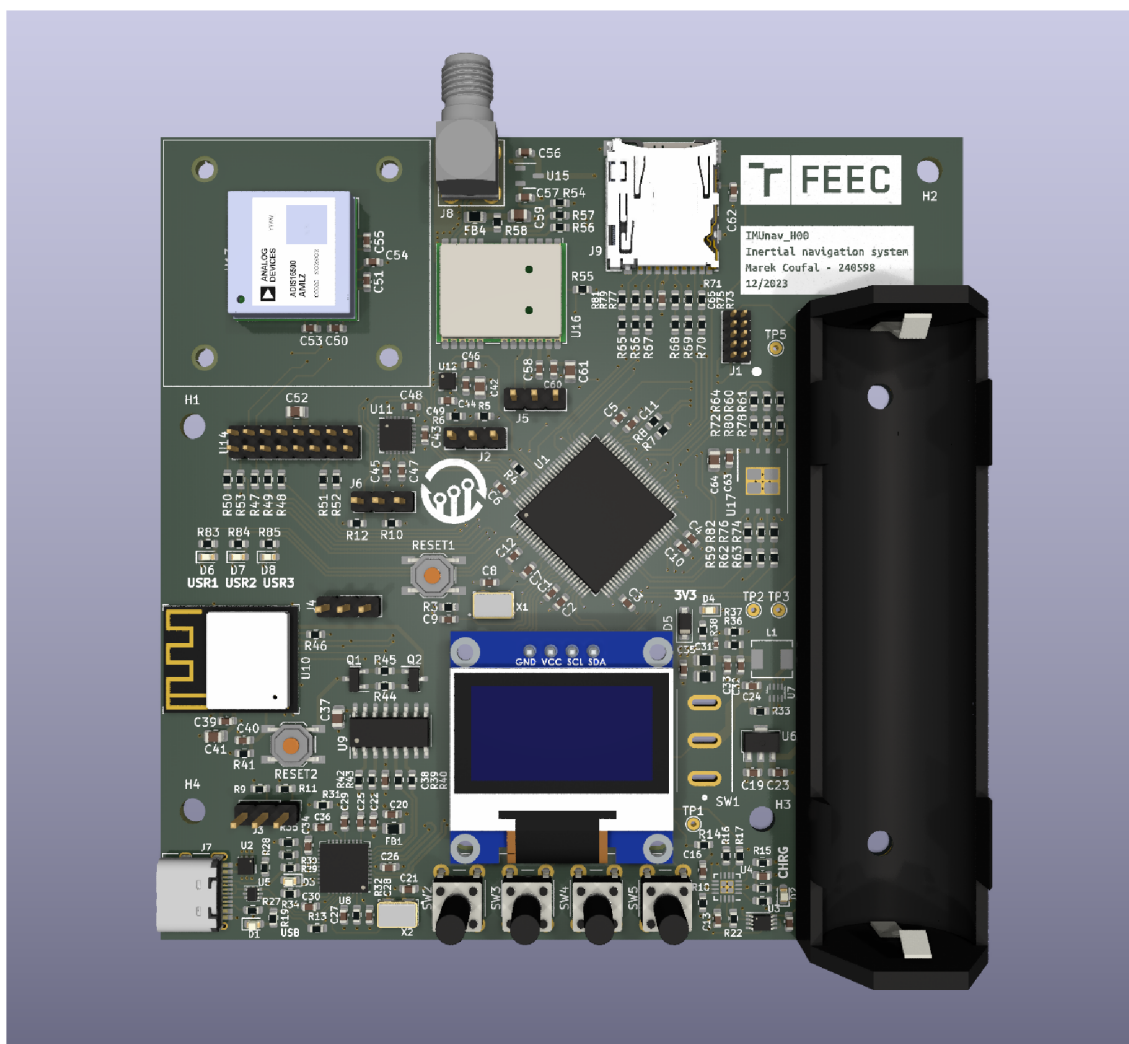
Pro účely vyzkoušení fúze dat z GNSS modulu a inerciálních senzorů byl navržen 3D tištěný držák (na obrázku 3.4) pro upevnění vývojových stavebnic osazených NEO-M8U a ADIS16505 s potřebnými periferními obvody na připojení k PC přes USB. Pomocí skriptů v Pythonu je možné ukládat data z obou senzorů do csv souborů a ty následně spojit.

Vzhledem k asynchronnosti USB komunikace bylo složité udržet definované vzorkovací kmitočty, popřípadě vzorkovat data z GNSS a IMU zároveň. Z tohoto důvodu byly desky použity na základní otestování a rozsáhlejší zpracování dat bude provedeno až s vlastní deskou hardwaru popsánem v kapitole 4.



Obr. 3.4: Testovací přípravek s vývojovými deskami.

4 Realizace hardwaru



Obr. 4.1: 3D model navržené DPS.

Schéma i DPS byly navrženy v programu KiCad. V příloze A je schéma inerciální jednotky rozdělené do několika logických bloků. Příloha B.1 obsahuje pohled na osazení součástek vrchní vrstvy a přílohy B.2 až B.5 obsahují nákres jednotlivých vrstev mědi. Na obrázku 4.1 je vygenerovaný 3D model DPS. Celý projekt programu KiCad je také dostupný v elektronické příloze.

Inerciální jednotka je realizována jako čtyřvrstvá deska plošných spojů o velikosti 100×100 mm s uspořádáním vrstev popsánem v tabulce 4.1.

Typ a tloušťka substrátu DPS byla vybrána v konfiguraci JLC04161H-7628, jejich mechanické uspořádání a dielektrické vlastnosti jsou popsány v tabulce 4.2. Pomocí kalkulačky výrobce byly vypočteny potřebné hodnoty požadovaných šířek a mezer spojů mikropáskového vedení pro impedanci 50Ω a 90Ω . Hodnoty pro ve-

Tab. 4.1: Signálové uspořádání vrstev na DPS.

Vrstva mědi	Využití
Horní	Vysokorychlostní signály
1. vnitřní	Společná zem
2. vnitřní	Napájení
Dolní	Signály

dení o impedanci $50\ \Omega$ byly použity při návrhu cest mezi GPS modulem a anténou subminiaturní verze A (*Sub- Miniature version A*, SMA). Šířka a vzdálenost diferenciálního páru o impedanci $90\ \Omega$ byla použita při návrhu USB části zapojení.

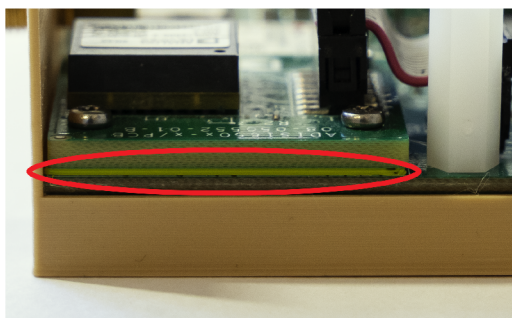
Tab. 4.2: Uspořádání měděných a izolačních vrstev DPS JLC04161H-7628.

Typ materiálu	Tloušťka (mm)	Relativní permitivita $\epsilon_r(-)$
Vrchní vrstva mědi	0,0350	1
Prepreg 7628	0,2104	4,4
1. vnitřní vrstva mědi	0,0152	1
Jádro	1,065	4,6
2. vnitřní vrstva mědi	0,0152	1
Prepreg 7628	0,2104	4,4
Spodní vrstva mědi	0,0350	1

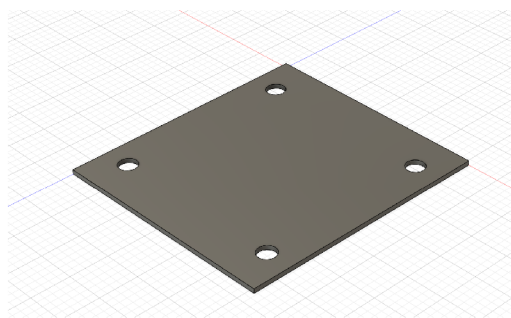
Většina pouzder pasivních součástek byla vybrána o velikosti 0603, což by mělo poskytnout dostatečný kompromis mezi velikostí výsledné desky a možností ruční výměny součástky pro případné opravy na prvním prototypu. Prototypová deska je také opatřena měřicími body na napájecích větvích a konektorovými hřebínky na digitálních komunikacích pro možnost připojení osciloskopu, nebo logického analyzátoru na odposlouchávání komunikace mezi MCU a jednotlivými senzory.

4.1 Konstrukce

Všechny SMD součástky, kromě GNSS a IMU modulu byly osazeny strojově, zbylé součástky a konstrukční prvky ručně. Vzhledem k tomu, že deska byla navržena pro možnost výběru použití BGA, nebo PCB varianty ADIS16505, tak v případě použití hotového modulu je potřeba použít izolační desku, která zamezí nežádoucím zkratům mezi odkrytými ploškami neosazeného BGA pouzdra a PCB varianty inerciálního

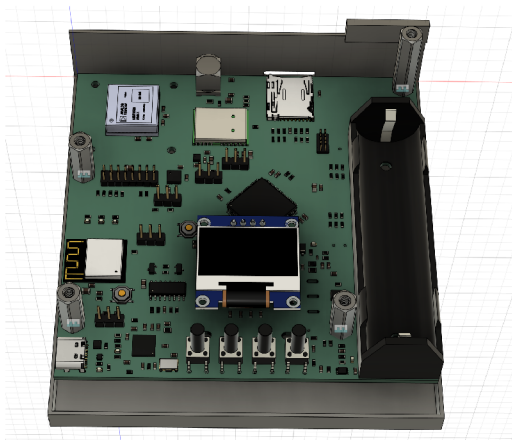


(a) Fotografie umístěného izolátoru.

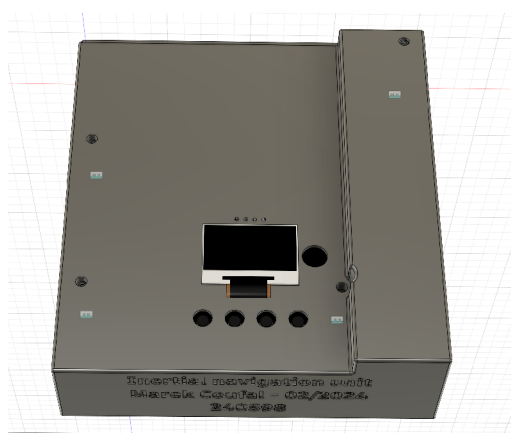


(b) 3D model izolátoru.

Obr. 4.2: Izolační podložka IMU.



(a) 3D model krabičky bez víka.



(b) 3D model krabičky s víkem.

Obr. 4.3: 3D model krabičky.

modulu. K tomuto účelu byl navržen jednoduchý 3D tištěný izolační prvek, který je vložen mezi jednotlivé desky, znázorněn na obrázku 4.2.

Obdobné izolační podložky byly také navrženy a použity při montáži distančních sloupků držící OLED displej.

Dále byla navržena a vyrobena 3D tištěná montážní krabička (obrázek 4.3) pro ochranu citlivých komponent zařízení při běžném užívání a manipulaci. Veškeré modely v této práci jsou navrženy pomocí 3D počítačem podporovanému projektovacímu (*Computer-Aided Design, CAD*) programu Fusion 360 od firmy Autodesk. Ve spodní části krabičky jsou přidány závitové vložky, které byly teplem vlisovány do plastového dílu pomocí mikropájky. DPS je připevněna k tomuto dílu pomocí čtveřice nylonových distančních sloupků, které zároveň slouží jako vzpěry pro vrchní díl krabice. Sestavené zařízení je na obrázku 4.4.



(a) Spodní díl krabičky s deskou.



(b) Hotová sestava.

Obr. 4.4: Fotografie zařízení.

4.2 Ověření základních funkcí hardwaru jednotky

Při prvním zapnutí desky byl k napájení použit laboratorní zdroj s proudovým omezením místo Li-Ion akumulátoru, aby v případě chyby v zapojení byla co nejvíce omezena šance poškození součástek. Zařízení při použití napájecího napětí 4 V odebíralo proud zhruba 198 mA, což je v tolerovaných mezích. Dále bylo zkontrolováno, zdali 3,3V snižující měnič, který napájí všechny citlivé komponenty, pracuje správně. Bylo změřeno napětí 3,313 V na jeho výstupu, je tedy v pořádku.

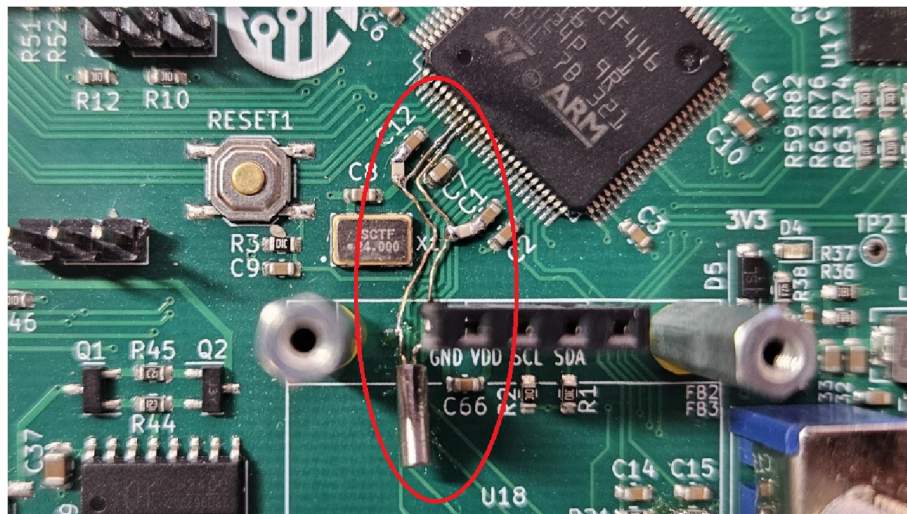
Jako další důležitý blok byla zkontrolována funkčnost S/R klopného obvodu, který poskytuje ochranu proti podvybití, popsanou v kapitole 3.6. K tomuto byl vytvořen jednoduchý firmware procesoru, který průběžně měří napětí akumulátoru pomocí ADC a v případě, že klesne pod úroveň 3,5 V přepne pin na vstupu KO do stavu logické jedničky. Tento stav byl simulován postupným snižováním výstupního napětí laboratorního zdroje. K vypnutí zařízení došlo při 3,54 V a ve vypnutém stavu zařízení odebíralo proud 19,73 μ A, což je spotřeba samotných klopných obvodů a RTC zálohovacích registrů MCU a GNSS modulu. Při tomto napětí zbývá v akumulátoru zhruba 10 % energie [23], můžeme tedy vypočítat jak dlouho je možné ponechat zařízení softwarově vypnuté se sepnutým hlavním vypínačem, aniž by došlo k degradaci akumulátoru

$$t = \frac{C \cdot 0,1}{I} = \frac{2,9 \cdot 0,1}{19,73 \cdot 10^{-6}} = 1,68 \text{ roku}, \quad (4.1)$$

kde C je kapacita akumulátoru v ampérhodinách a I je proudový odběr vypnutého zařízení.

Při připojení zařízení ke zdroji pomocí USB bylo napájení obnoveno, což je žádoucí. Také byla zkontrolována proudová spotřeba při nabíjení akumulátoru pomocí orientačního USB měřicího přístroje TC66C, ta byla 751 mA v případě, že je zařízení zapnuté a akumulátor se nabíjí.

Další části zařízení jsou úzce vázány na software, jejich funkčnost byla tedy postupně testována při vývoji. Později byl odhalen drobný nedostatek hardwarového návrhu, a to v oblasti RTC periferie hlavního procesoru. Ta slouží k udržení aktuálního času, který je možné synchronizovat například pomocí GNSS. Z tohoto důvodu výrobce MCU umožňuje napájet periferii pomocí separátního pinu *VBAT* který je využíván při vypnutém hlavním napájení a má velice malý proudový odběr, jehož zdrojem je lineární napěťový regulátor s malým klidovým proudem připojeným přímo k akumulátoru. Vzhledem k tomu, že tato funkcionality není pro aplikaci inerciální navigace klíčová a jedná se pouze o možnost zvýšení pohodlí uživatele, tak nebyly kladeny vysoké nároky na její přesnost, proto nebyl zapojen externí nízkofrekvenční krystalový oscilátor *LSE* s domněním, že bude postačovat pouze interní oscilátor *LSI RC*. Ukázalo se ovšem, že z napájecí domény *VBAT* je poskytováno napájení pouze externímu nízkofrekvenčnímu oscilátoru, nikoliv internímu [21]. Z tohoto důvodu byl dodatečně přidán 32,768kHz krystal, společně se zatěžovacími kondenzátory (viz. obrázek 4.5), který byl po otestování připevněn lepidlem.



Obr. 4.5: Dodatečná oprava RTC oscilátoru.

5 Firmware inerciální jednotky

Firmware hlavního MCU byl vyvíjen pomocí volně dostupného vývojového prostředí (*Integrated Development Environment*, IDE) poskytovaného výrobcem - *STM32CubeIDE*. Jedná se o nástroj určený pro práci s jazyky C/C++, GCC kompilátorem, založeném na Eclipse [24]. Zároveň poskytuje grafické rozhraní pro konfiguraci a generování knihoven vrstvy abstrakce hardwaru (*Hardware Abstraction Layer*, HAL), možnost použití operačního systému reálného času (*Real Time Operating System*, RTOS) a ladicí prostředí.

V této práci byly použity poskytované knihovny HAL a FreeRTOS pro ulehčení a urychlení vývoje firmwaru. Jejich použití často s sebou nese nevýhody, jako je například horší využití paměti, nebo výpočetního výkonu, z tohoto důvodu byl zvolen takový MCU, aby měl dostatečné rezervy pro jejich použití.

5.1 HAL

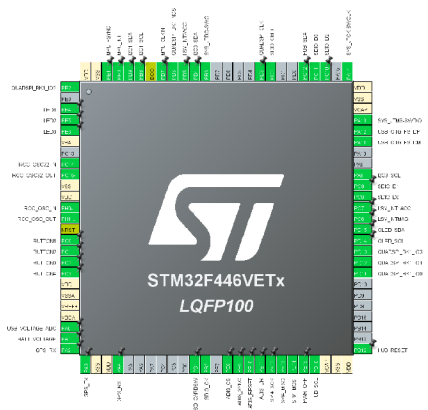
Generování kódu HAL v *STM32CubeIDE* je možné pomocí grafického rozhraní, které poskytuje uživateli možnost nastavení jednotlivých pinů, periférií, komunikačních rozhraní a vnitřních hodin (obrázek 5.1). Vygenerované knihovny následně umožňují uživateli pracovat s MCU s jistou mírou abstrakce, například není nutné znát a pracovat s názvy jednotlivých registrů. Typickým příkladem mohou být komunikační sběrnice (SPI, I2C . . .), pro které jsou dostupné obslužné funkce na čtení a vysílání dat, jak v blokujícím režimu, tak i v neblokujícím (například pomocí přímého vstupu do paměti (*Direct Memory Acces*, DMA)). [24]

Dále je možné pomocí stejného grafického rozhraní importovat rozšiřující softwarové balíčky, i když už se nejedná přímo o HAL. V této práci byly použity *FATFS* pro manipulaci se soubory na microSD kartě, *FreeRTOS* jakožto jeden z dostupných RTOS a *USB_DEVICE* pro práci s USB rozhraním třídy *Mass Storage Class* - *třída paměťového média* (MSC).

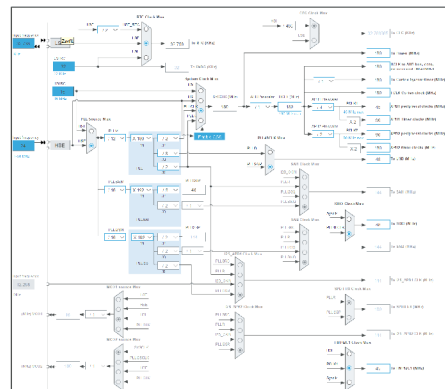
5.2 FreeRTOS

V této aplikaci je potřeba vyčítat, převádět a zapisovat data z několika různých senzorů, které nemají přesně stejný hodinový signál, zároveň obsluhovat grafické uživatelské rozhraní (*Graphical User Interface*, GUI) a provádět záznam dat. Pro potřeby synchronizace několika úloh, které nemají stejné periody, nebo například čekají na vstup od uživatele se hodí RTOS.

Byla vybrána jedna z variant operačních systémů reálného času, a to FreeRTOS. Jedná se o jednoduchý open-source systém, který je hojně využíván ve vestavěných



(a) Konfigurace pinů.



(b) Konfigurace hodin.

Obr. 5.1: Konfigurace MCU v STM32CubeIDE.

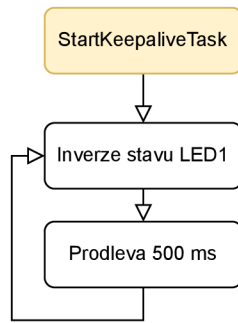
aplikacích. Umožňuje aplikaci virtuálně rozdělit na několik samostatných vláken (tzv. *tasků*) s různými prioritami. Časování tasků je možné například pomocí neblokujících prodlev, nebo semaforů. FreeRTOS také plní funkci správy a alokace paměti. Předávání informací mezi jednotlivými tasky se provádí pomocí tzv. *Queues*, které představují zásobníky typu první dovnitř, první ven (*First In First Out*, FIFO) s nastavitelnou délkou fronty a velikostí jednotlivých dat. Díky tomu je možné se vyvarovat použití globálních proměnných. [25]

Práce s FreeRTOS je v STM32CubeIDE zjednodušená také díky poměrně dobré možnosti ladit aplikace pomocí již vestavěného RTOS-aware debuggeru, díky kterému můžeme například analyzovat využití paměti jednotlivých tasků, využití času, nebo kontrolovat stavy semaforů a velikost obsazených Queues.

5.3 Vývojové diagramy firmwaru

Popsat chování a funkcionalitu firmwaru této aplikace dohromady by bylo poměrně nepřehledné. Proto budou jednotlivé funkce rozděleny do několika samostatných logických bloků, kde každý blok reprezentuje jeden task operačního systému.

Diagramy byly vytvořeny s úmyslem co nejlépe a nejjednodušeji reprezentovat chování jednotlivých tasků, proto i ty představují jistou míru abstrakce a neobsahují velké množství detailů, jako jsou různé prodlevy, obsluhy pinů, kontroly časovačů a podobně, aby byly lépe čitelné. Kompletní zdrojový kód je dostupný v elektronické příloze práce.

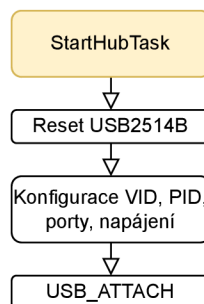


Obr. 5.2: Vývojový diagram KeepaliveTask.

5.3.1 KeepaliveTask

Jedná se o úlohu s nastavenou nejnižší prioritou. Slouží pouze pro ladicí účely a umožňuje jednoduchou a rychlou reprezentaci stavu systému pomocí blikající LED, zdali je spouštěn i task s nejnižší prioritou. V případě, že by LED přestala blikat, znamená to, že buď nějaký z vyšších tasků využívá výpočetní čas natolik, že se již nespustí úlohy s nižší prioritou, nebo došlo k chybě systému (například problémy přístupu do paměti)

5.3.2 hubTask



Obr. 5.3: Vývojový diagram HubTask.

Tato úloha vykonává funkce pouze při zapnutí zařízení, a to konfiguraci a sepnutí vestavěného USB rozbočovače. Do něj jsou nahrána konfigurační data pomocí sběrnice I2C, jako je například identifikace výrobce (*Vendor ID*, VID), identifikace produktu (*Product ID*, PID), nastavení napájecího režimu a nastavení jednotlivých portů. U tohoto rozbočovače jsou zapnuty pouze využívané porty, aby byla snížena spotřeba zařízení. Následně jsou registry rozbočovače přepnuty do režimu pouze pro čtení a USB rozhraní zapnuto. Při běhu zařízení již konfigurace zůstává stejná a task je neaktivní.

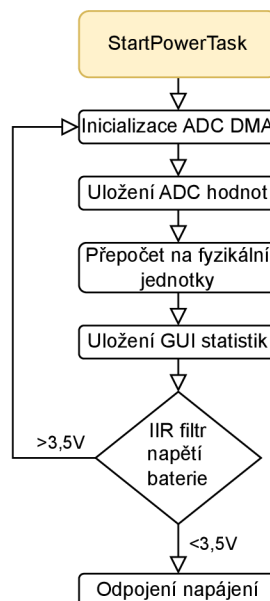
5.3.3 powerTask

V tomto tasku jsou periodicky měřena všechna analogová napětí pomocí ADC procesoru, například napětí zdroje, USB portu, akumulátoru, ale i teplota procesoru. Tyto stavové veličiny jsou zobrazovány pomocí GUI. Čtené hodnoty napětí akumulátoru jsou průměrovány pomocí pohyblivého exponenciálního filtru, který je možný zapsat pomocí rovnice

$$y[n] = \alpha \cdot x[n] + (1 - \alpha) \cdot y[n - 1], \quad (5.1)$$

kde $x[n]$ je přečtená hodnota napětí, $y[n]$ vyfiltrovaná hodnota napětí, $y[n - 1]$ výsledek vyfiltrované hodnoty napětí z předešlého cyklu a α je nastavitelný koeficient odezvy filtru. Experimentálně bylo odzkoušeno, že vhodných výsledků filtrace šumu je možné dosáhnout s $\alpha = 0,3$. Tento typ filtru byl zvolen zejména pro jeho jednoduchost a úsporné využití paměti.

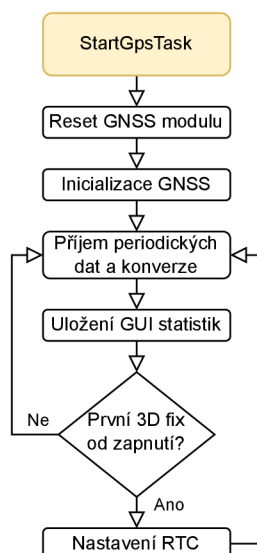
V případě, že klesne napětí akumulátoru pod hranici 3,5 V je zařízení vypnuto překlopením S/R klopného obvodu, který je zmiňovaný v kapitole 4 a tím je dosažena ochrana akumulátoru proti podvybití. Poté je možné zařízení znova zapnout pouze připojením do USB nabíječky.



Obr. 5.4: Vývojový diagram PowerTask.

5.3.4 gpsTask

Tato úloha se stará o periodické zpracování příchozích dat z GNSS modulu pomocí UBX zpráv. Tato data jsou přijímány pomocí sběrnice UART a je využito DMA

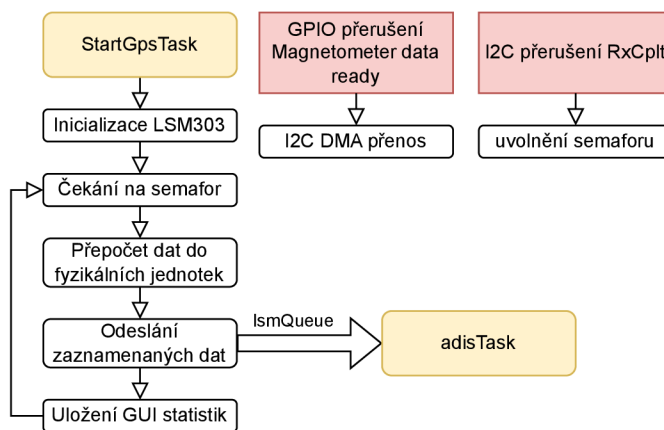


Obr. 5.5: Vývojový diagram GpsTask.

přenosu, následně jsou převedena do čitelné podoby (datum, čas, zeměpisná šířka, délka. . .) pomocí knihovny GNSS parseru [26].

V této úloze je také kontrolováno, kdy dojde k prvnímu přesnému určení polohy fixací na satelity a je aktualizován čas a datum do vnitřního RTC.

5.3.5 IsmTask

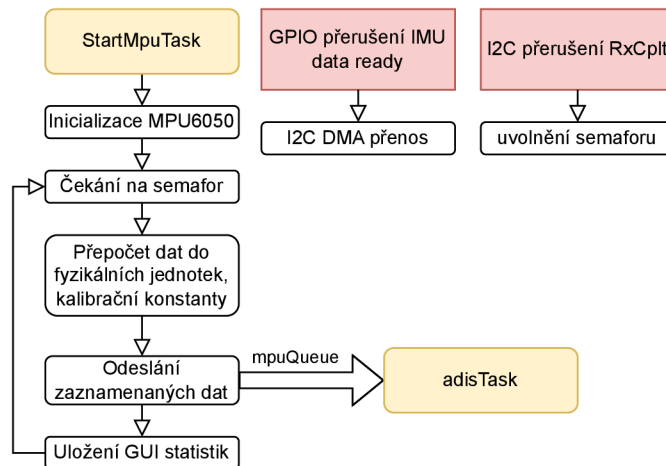


Obr. 5.6: Vývojový diagram LsmTask.

V tomto tasku jsou periodicky vyčítána data o magnetickém poli z elektronického kompasu LSM303. Ten je při zapnutí zresetován a následně inicializován rozsah, rozlišení a vzorkovací frekvence na 100 Hz. Po navzorkování dat magnetometr změní stav na svém pinu signalizujícím konec vzorkování. V ten okamžik je započat DMA přenos pomocí sběrnice I2C a jakmile jsou data vyčtena, převedou se do fyzikálních

jednotek a odešlou se k záznamu. Využití DMA a přerušení data ready pinu senzoru nám umožňuje využívat plně neblokující kód, který má nízké využití výpočetního času MCU.

5.3.6 mpuTask



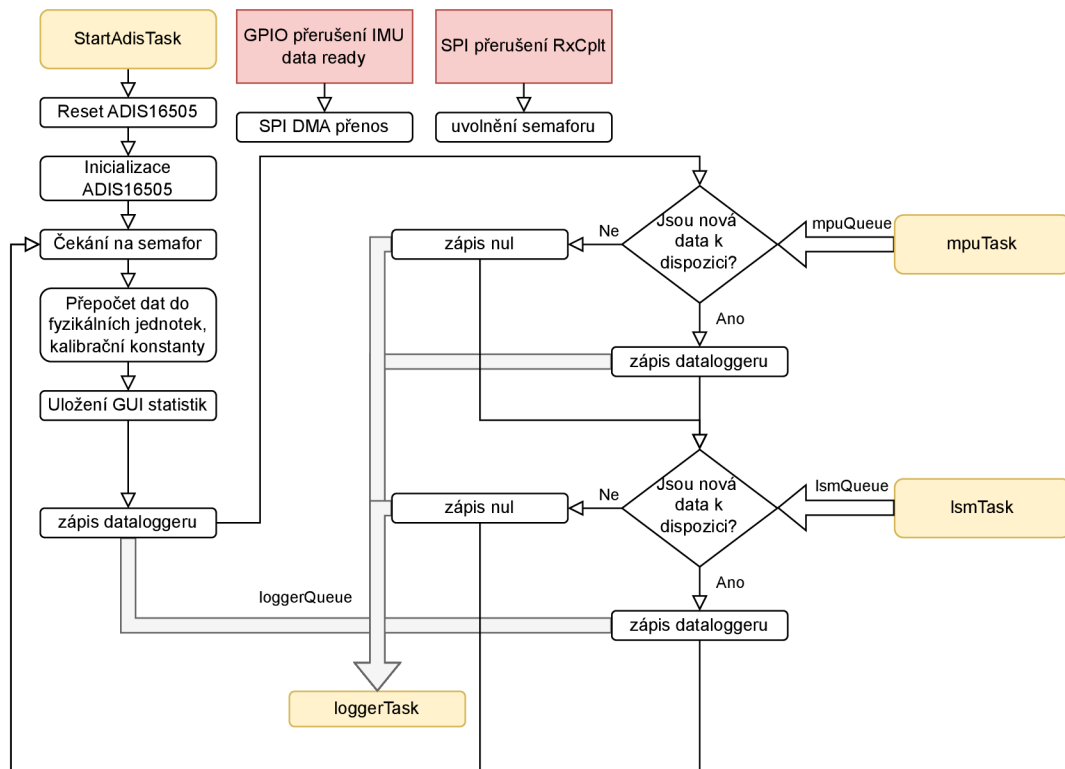
Obr. 5.7: Vývojový diagram MpuTask.

Tato úloha je velice obdobná předchozí z kapitoly 5.3.5. Zde jsou čtena a převáděna data z šestiosého IMU MPU6050 s frekvencí 400 Hz.

5.3.7 adisTask

Tento task, obdobně jako předchozí, obstarává inicializaci, vyčítání a konverzi dat z šestiosého IMU ADIS16505 se vzorkovací frekvencí 400 Hz. Komunikace se senzorem probíhá přes sběrnici SPI a stejně jako u ostatních sensorů, je využíváno přerušení a DMA přenosů k vytvoření neblokujícího kódu. Vzhledem k vyššímu rozlišení senzoru v porovnání s ostatními je pro ukládání hodnot převedených do jednotek SI využíván datový typ `double` místo `float`.

Vzorkování dat z ADIS16505 zároveň slouží jako synchronizace k zarovnání řádků dat ve výsledném souboru naměřených dat, tedy v případě, že jsou dostupná nová data z LSM303, nebo MPU6050, jsou přidána do celkové záznamové datové struktury, v opačném případě jsou na jejich odpovídající místa zapsány nuly. V nejhorším možném případě, tedy že data z ostatních sensorů jsou vzorkována těsně po navzorkování z ADIS16505 je jejich zpoždění rovno délce jedné periody celkové vzorkovací frekvence, tedy 2,5 ms.



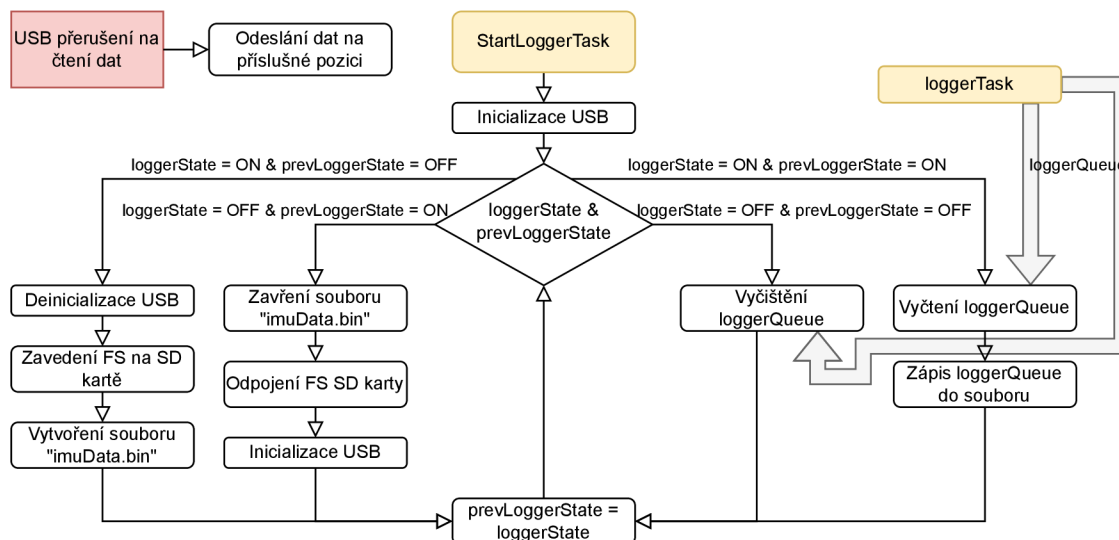
Obr. 5.8: Vývojový diagram AdisTask.

5.3.8 loggerTask

Tato úloha provádí samotné ukládání dat. Testováním bylo zjištěno, že při použití kvalitnějších microSD karet (v zařízení je použita *SAMSUNG 64GB EVO PLUS*) nedochází k náhodným delším prodlevám při zápisu. Zároveň v porovnání s NOR FLASH pamětí poskytuje řádově větší úložiště, rychlejší zápis bloku dat, není potřeba mazat každý sektor paměti před zápisem a také je jednodušší implementace souborového systému. Z těchto důvodů bylo rozhodnuto o využití SD karty k ukládání měřených dat.

Stavový automat v tomto tasku lze rozdělit do čtyř stavů: záznam dat, kdy jsou data průběžně ukládána, vypnutý záznam, kdy jsou data zahazována a začátek a konec záznamu, kdy dochází k inicializacím USB a souborových systémů. V průběhu záznamu je USB rozhraní vypnuté, aby nebylo možné zasahovat do souboru připojeným PC když je zrovna do něj zapisováno zařízením.

Samotný přenos naměřených dat do počítače je realizován přes USB třídy vysokokapacitního úložiště. Připojený PC při potřebě čtení dat spustí přerušení na přenos s konkrétní adresou. Jelikož je na microSD kartě implementován souborový systém FatFS, je v běžných operačních systémech reprezentován jako obyčejné externí úložiště.



Obr. 5.9: Vývojový diagram LoggerTask.

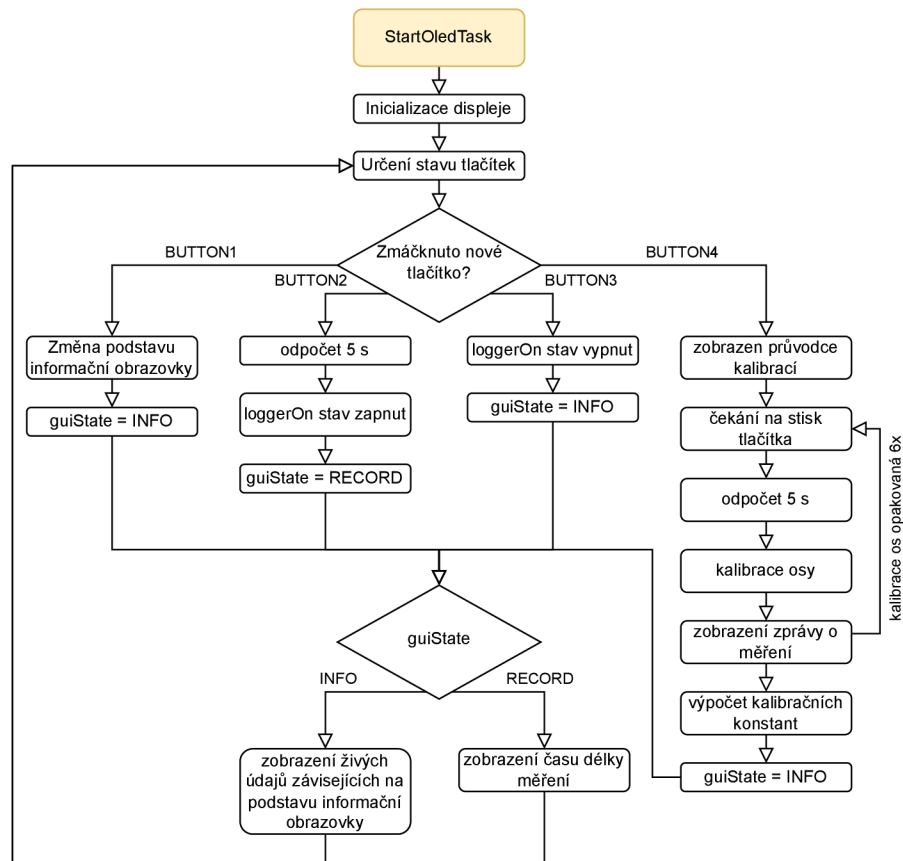
Jednotka je také vybavena WiFi modulem pro bezdrátovou komunikaci, který by bylo možné využít pro bezdrátový přenos dat. Zatím byl ovšem implementován pouze přenos pomocí USB, jelikož je hlavní MCU propojen s bezdrátovým rozhraním přes sběrnici UART s maximální rychlostí komunikace 500 kBaud/s, a přenos dat při delším měření by byl poměrně dlouhý.

5.3.9 oledTask

Tento task obstarává obsluhu uživatelského rozhraní pomocí grafického displeje a čtveřice tlačítek. Při každém cyklu je porovnán předchozí a stávající stav tlačítek. Jestliže došlo k nějakému stisknutí, změní se náležitě zobrazované informace a chování zařízení. Displej komunikuje s MCU přes sběrnici I2C a k ovládání byla použita knihovna řadiče SSD1306 [27]. Funkce jednotlivých tlačítek jsou reprezentovány ikonami ohraničenými rámečkem na spodní straně displeje.

Stav zařízení by se dal zjednodušeně znázornit třemi možnostmi:

- Výchozí obrazovka stavových informací (obrázek 5.11) - pomocí ní je možné odečítat okamžité hodnoty veličin měřených senzory a zařízením. Při zapnutí je zobrazena výchozí domovská obrazovka, na které jsou přítomny provozní informace, jako je napětí akumulátoru, teplota procesoru, nebo čas uložený v RTC. Stisky tlačítka 1 (reprezentováno ikonou HOME na displeji) je možné měnit zobrazované informace na jednotlivé IMU, elektronický kompas a GNSS modul. Při změně obrazovky se také náležitě obmění levá spodní ikona reprezentující tlačítka 1. Všechny sensorové veličiny (lineární zrychlení, úhlová rychlost, velikost magnetické indukce) jsou již převedeny a zobrazeny v jednotkách



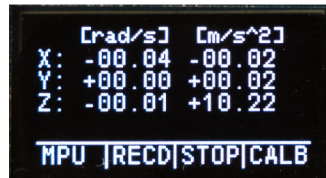
Obr. 5.10: Vývojový diagram OledTask.

z mezinárodní soustavy jednotek.

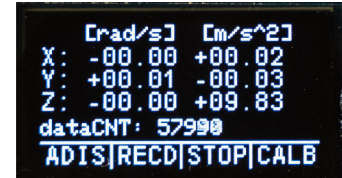
- Záznam dat (obrázek 5.12) - Ten je možné aktivovat stiskem tlačítka 2, kterému náleží ikona RECD. Po jeho stisknutí je na displeji zobrazen pětisekundový odpočet, který slouží k tomu, aby uživatel umístil jednotku do nehybného stavu (i samotný zákmit ze stisku tlačítka může velice ovlivnit měření) a až poté je zapnut záznam dat. V průběhu záznamu je uživateli zobrazována délka zaznamenaných dat (v sekundách) a není nijak omezena. Záznam dat je poté možné přerušit stiskem tlačítka 3, kterému odpovídá ikona STOP.
- Kalibrace (obrázek 5.13) - toto menu slouží jako průvodce kalibrační procedurou. Aktivuje se stiskem tlačítka 4, kterému odpovídá ikona CALB. Následně je uživateli zobrazen dvoustránkový průvodce kalibrační procedurou, který popisuje jednotlivé kroky potřebné k úspěšnému provedení kalibrace. Poté je uživatel vyzván k umístění jednotky na jednu z 6 hran a po pětisekundovém odpočtu je změřeno 50 vzorků lineárního zrychlení a úhlové rychlosti z obou IMU a vypočítán jejich průměr. Tento postup se opakuje šestkrát s každou hranou zařízení. Následně jsou z měření vypočteny kalibrační konstanty pomocí algoritmu popsáném v kapitole 5.4.



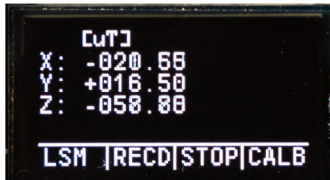
(a) Domovská obrazovka.



(b) Hodnoty MPU6050.



(c) Hodnoty ADIS16505.

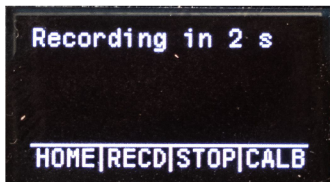


(d) Hodnoty LSM303.

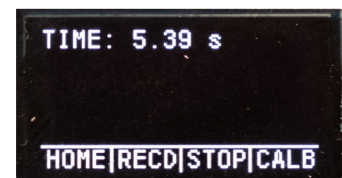


(e) Hodnoty GNSS.

Obr. 5.11: GUI stavových informací jednotlivých senzorů.



(a) Odpočet startu.



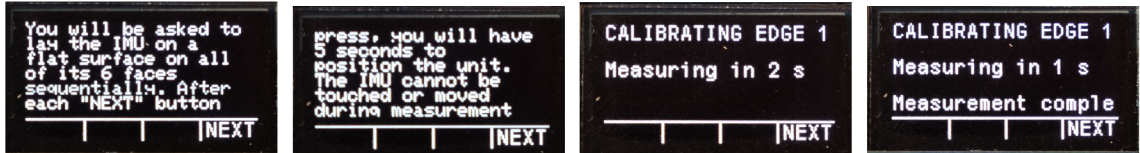
(b) Délka záznamu dat.

Obr. 5.12: GUI záznamu dat.

5.4 Kalibrace IMU

Akcelerometry a gyroskopy jsou poměrně citlivé součástky. Přestože většina i těch levnějších IMU jsou v továrně výrobcem kalibrovány, jejich vlastnosti se můžou změnit například procesem pájení, změnou teploty, nebo mechanickým pnutím v desce plošných spojů. Drtivá většina popsaných kalibračních metod akcelerometrů se zaměřuje na korekci lineární křivkou, jedná se tedy o výpočet konstant k a q v rovnici $y = k \cdot x + q$, kde x je měřená hodnota senzorem, k a q jsou kalibrační konstanty a y je výsledná veličina po kalibraci. Konstantu k můžeme označit jako gain a konstantu q jako offset. Většinou se dostupné metody kalibrací používají i pro kompenzaci odchylek natočení os senzorů vůči desce, nebo nějakému jinému referenčnímu prvku zařízení (například hrany krabice), které můžou vzniknout nepřesným pájením, nebo montáží. Tyto metody vyžadují buď přípravky, které nám umožní otáčet se zařízením o přesně definovaný úhel v pravouhlých osách, nebo alespoň vyžadují přesné natáčení zařízení do poloh, kde dvě z os měří nulové zrychlení. [28] [29]

V této práci ovšem není až tak důležité kompenzovat natočení os senzorů, jelikož je zařízení určené pro manipulaci v rukách, jde nám tedy pouze o výpočet konstant



(a) Průvodce 1. (b) Průvodce 2. (c) Odpočet měření. (d) Hotový záznam.

Obr. 5.13: GUI kalibrační procedury.

k a q , aby bylo měřeno zrychlení správné velikosti. Výroba výše zmíněných mechanických přípravků, které by byly dostatečně přesné a opakovatelné, by byla náročná a nesouvisející s povahou této práce. Díky tomu, že nepotřebujeme kompenzovat natočení os, můžeme vycházet z dostupných kalibračních procedur a implementovat jejich podstatu v upravené podobě:

Kalibrační procedura vychází ze znalosti velikosti tíhového zrychlení. Budeme-li mít akcelerometr nehybný, tak v jakékoliv jeho pozici by velikost vektoru zrychlení měla být rovna tíhovému zrychlení. To můžeme zapsat pomocí rovnice 5.2.

$$|\vec{a}| = \sqrt{|\vec{a}_x|^2 + |\vec{a}_y|^2 + |\vec{a}_z|^2} = |\vec{g}|. \quad (5.2)$$

Za velikosti zrychlení dosadíme měřené hodnoty akcelerometrem, korigované lineární křivkou. Dostaneme rovnici 5.3.

$$|\vec{g}| = \sqrt{(k_x \cdot a_x + q_x)^2 + (k_y \cdot a_y + q_y)^2 + (k_z \cdot a_z + q_z)^2}. \quad (5.3)$$

Cílem je tedy určit konstanty k a q . K tomu je potřeba vyřešit nelineární rovnici o šesti neznámých. K řešení provedeme 6 měření a vytvoříme soustavu 6 nelineárních rovnic o 6 neznámých z rovnice 5.3. Teoreticky nezáleží na natočení zařízení v průběhu 6 zmiňovaných měření, jelikož je počítána velikost vektoru zrychlení ze tří ortogonálních os senzoru, ovšem nejlepších výsledků dosáhneme, když natočíme zařízení na všechny ze 6 stran krabičky zařízení (pravý bok, levý bok, spodní strana, horní strana, přední strana a zadní strana), jelikož v těchto případech budou měřena maxima a minima zrychlení pro všechny osy. Pokud bychom totiž provedli 6 měření v úplně stejné poloze zařízení, může se stát, že použitá numerická metoda řešení rovnic konverguje k řešení typu všechny gainy budou nulové a offset v jedné ose bude roven velikosti tíhového zrychlení. Toto řešení by bylo matematicky správné, ale nedávalo by smysl.

Jednou z možností, jak kalibraci provést by bylo tzv. "offline", tedy provést kalibrační měření a následně data přenést do počítače a kalibrační konstanty vypočítat například pomocí MATLABu. To by ovšem omezilo univerzálnost vyvíjeného zařízení, také by byla kalibrace poměrně složitá a nemotorná z pohledu uživatele. Jako

pohodlnější řešení byla zvolena implementace metody nejmenších čtverců pomocí funkce `fsolve` ve spojení s *MATLAB Coder*.

Nejdříve byla vytvořena následující funkce, která slouží k určení všech 6 kalibračních konstant akcelerometru. Tyto kalibrační konstanty jsou uloženy do výstupního vektoru `x`. Vstupními argumenty je matice `accel(6,3)`, která obsahuje naměřené zrychlení a `gravityScalar`, což je velikost tíhového zrychlení.

```
1 function [x,fval] = AccelCalSolver(accel, gravityScalar) %#codegen
2 % The directive %#codegen indicates that the function
3 % is intended for code generation
4 arguments
5     accel (6,3) double
6     gravityScalar (1,1) double
7 end
8
9
10 fun = @(x)gravityFun(x, accel, gravityScalar);%gravity function handler
11 x0 = [1 0 1 0 1 0]; % Initial point - gain=1, offset=0
12 options = optimoptions('fsolve','Algorithm','levenberg-marquardt','Display',
13     'off');
14 [x,fval] = fsolve(fun, x0, options);
15 end
```

Kořeny rovnice jsou vyřešeny metodou nejmenších čtverců s výchozími body `gain = 1` a `offset = 0`. Samotný tvar matice je definován v samostatném souboru a obsahuje šest rovnic 5.3 v upraveném tvaru:

```
1 function F = gravityFun(x, accel, gravityScalar)
2 F = zeros(6,1); % Allocate return array
3
4
5 F(1) = (accel(1,1)*x(1) + x(2))^2 + (accel(1,2)*x(3) + x(4))^2 + (accel
6     (1,3)*x(5) + x(6))^2 - gravityScalar^2;
7 F(2) = (accel(2,1)*x(1) + x(2))^2 + (accel(2,2)*x(3) + x(4))^2 + (accel
8     (2,3)*x(5) + x(6))^2 - gravityScalar^2;
9 F(3) = (accel(3,1)*x(1) + x(2))^2 + (accel(3,2)*x(3) + x(4))^2 + (accel
10    (3,3)*x(5) + x(6))^2 - gravityScalar^2;
11 F(4) = (accel(4,1)*x(1) + x(2))^2 + (accel(4,2)*x(3) + x(4))^2 + (accel
12    (4,3)*x(5) + x(6))^2 - gravityScalar^2;
13 F(5) = (accel(5,1)*x(1) + x(2))^2 + (accel(5,2)*x(3) + x(4))^2 + (accel
14    (5,3)*x(5) + x(6))^2 - gravityScalar^2;
15 F(6) = (accel(6,1)*x(1) + x(2))^2 + (accel(6,2)*x(3) + x(4))^2 + (accel
16    (6,3)*x(5) + x(6))^2 - gravityScalar^2;
17 end
```

Také byla vytvořena testovací funkce na její ověření. Všechny zdrojové kódy jsou také dostupné v elektronické příloze. MATLAB umožňuje vytvořit zdrojový kód v jazyce C k těmto funkcím pomocí aplikace Coder. Optimalizace byla zvolena pro procesory ARM a vygenerované soubory použity a implementovány ve firmwaru zařízení.

Kalibrace gyroskopů lineární křivkou by byla bohužel složitější a již by vyžadovala nějaký přesný mechanický přípravek na natáčení zařízení o přesně definovaný úhel. Z tohoto důvodu je u gyroskopů kompenzován pouze jejich offset. Ten je měřen ve stejnou dobu při všech 6 měření akcelerometru. Následně je vypočten aritmetický průměr všech 6 měření pro každou osu gyroskopu zvlášť, tato hodnota představuje offset senzoru.

5.5 Převod dat do CSV souboru

Žádaným výstupem ze zařízení jsou naměřená data v jednoduše čitelném a univerzálním typu souboru, například čárkou oddělené hodnoty (*Comma Separated Value*, CSV), ve kterém jsou naměřené veličiny uloženy v textové podobě. MCU k záznamu dat a výpočtu používá standardní datové typy, jako je `int_8`, `uint_16`, `float`, `double` a podobně. Pokud bychom přímo v zařízení převáděli tato čísla na text, tak se násobně zvětší nároky na rychlost a velikost ukládání dat. Například datový typ `int_8` je v binární podobě reprezentován pouze jedním bytem, zatímco v textové podobě čtyřmi bajty (např. +127). Z tohoto důvodu jsou data na SD kartu ukládána v binární podobě a po přenosu do PC převedena do CSV souboru pomocí skriptu napsaném v jazyce Python, který je dostupný v elektronické příloze.

Tento skript má v sobě definované pořadí jednotlivých datových typů ukládané struktury proměnných. Také bylo potřeba přidat zarovnávací (padding) bity, které kompilátor automaticky přidává na zarovnání proměnných v paměti pro správné fungování MCU. Toto zarovnání je definováno v sestavovacím souboru (linker file) a pro uživatelskou oblast RAM je nastaveno na 64 bitů.

V případě, že senzory s nižší vzorkovací frekvencí nenaměřily nová data, zapíší na všechna odpovídající místa nuly. Skript proto kontroluje tato místa a v případě, že struktura dat senzoru obsahuje samé nuly, nahradí je prázdným místem.

Natočení os jednotlivých senzorů v zařízení se mezi sebou liší, proto jsou upravena tak, aby odpovídala referenčním osám ADIS16505, které jsou v pořadí ENU (East, North, Up).

6 Zpracování dat

Vytvořené zařízení inerciální jednotky poskytuje poměrně velké množství dat, které je možné různě zpracovávat. Kromě samotných dat z dvou IMU také zaznamenává měřené hodnoty z elektronického kompasu a GNSS modulu. V průběhu vypracování práce se ukázalo, že zpracovat data tak, abychom dosáhli uspokojivých výsledků, není vůbec jednoduché a zasloužilo by si samo o sobě rozsah další práce. V této kapitole budou popsány zejména dva postupy zpracování měření, a to výpočet trajektorie čistě z pohybových rovnic a fúze dat s GNSS. Později bylo také zjištěno, že navigace čistě z inerciálních dat nedosahuje uspokojivých výsledků a je zapotřebí korekce dat z GNSS, který má ovšem často ve vnitřních prostorech velmi špatné pokrytí.

Všechny níže popsané skripty byly vytvořené v prostředí MATLAB R2023b, jelikož je v něm manipulace s vektory a maticemi jednoduchá. Také můžeme využít již hotových modelů chování senzorů a implementovaných filtrů z Navigation Toolboxu, který je stále poměrně rozsáhle rozšiřován. V elektronické příloze jsou kromě skriptů dostupná i vzorová naměřená data, převedena do formátu csv, které je možné použít k experimentování. Obsahují například chůzi napříč budovou a místností, jízdu autem na venkovním prostranství a měření jednotky v klidu.

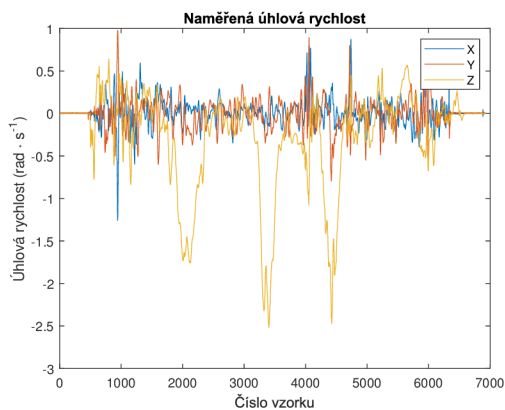
6.1 Výpočet trajektorie pomocí pohybových rovnic

Jedná se o základní způsob zpracování dat, který byl popsán v kapitole 1, které lze nejlépe vystihnout obrázkem 1.2. Nejdříve je vypočteno z prvního vzorku zrychlení natočení celého zařízení vůči zemi, tedy směr tíhového vektoru. Následně jsou data z gyroskopu integrována numerickou lichoběžníkovou metodou, abychom získali změnu orientace zařízení, ke kterému je přičten výchozí stav.

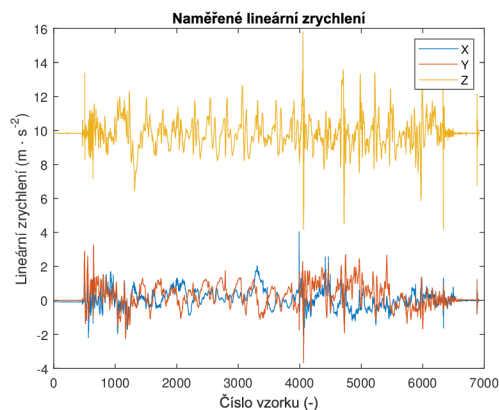
Z těchto úhlů natočení je vypočtena rotační matice, pomocí které otočíme měřená data zrychlení z body framu do earth framu. Z tohoto rotovaného zrychlení je následně odečten vektor tíhového zrychlení a pomocí dvou dalších numerických integrací otočeného zrychlení vypočteme odhad trajektorie.

Pro demonstraci tohoto postupu byla vytvořena vzorová data chůze ve tvaru obdélníku o rozměrech 3,5 a 4 m. Na začátku a konci těchto měření byla jednotka položena nehybně na stole.

Na obrázku 6.1 jsou vidět nezpracovaná zaznamenaná data. Úhlová rychlost v ose Z obsahuje výrazné špičky, jedná se o otáčení v rozích pomyslného obdélníku. Na datech akcelerometru je vidět, že celý čas měření byla jednotka převážně ve vodorovné pozici, jelikož lineární zrychlení v ose Z představuje tíhové zrychlení. Zákmity ve zrychlení jsou způsobeny kroky chůze.

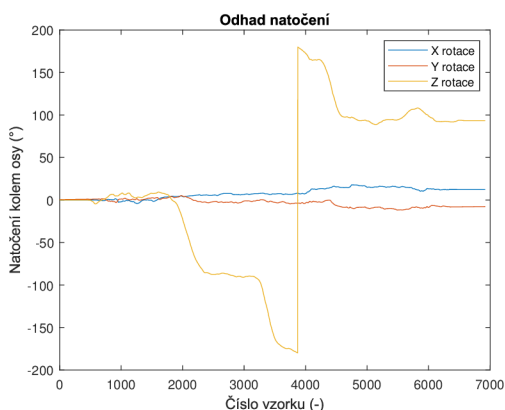


(a) Data z akcelerometru.

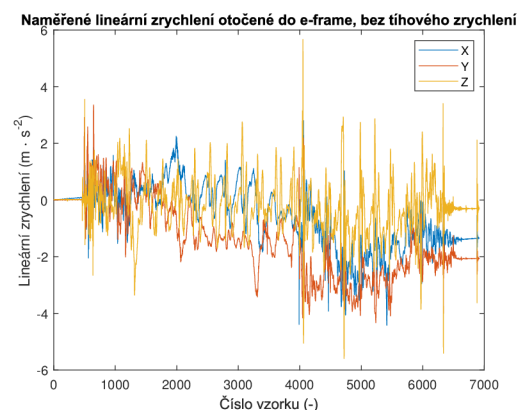


(b) Data z gyroskopu.

Obr. 6.1: Záznam dat IMU ADIS16505.



(a) Natočení jednotky.



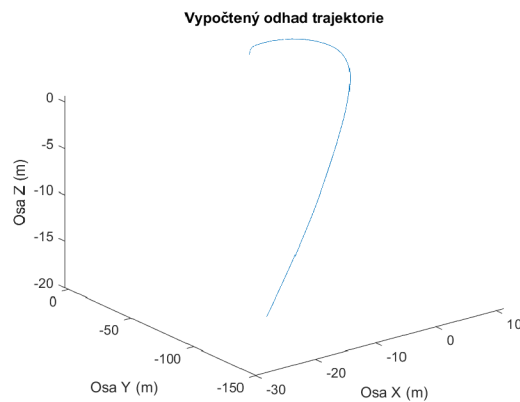
(b) Zrychlení v e-frame bez tíhového.

Obr. 6.2: Výsledky po první integraci.

Po integraci dat úhlové rychlosti můžeme určit natočení jednotky (znázorněno na obrázku 6.2a) a rotovat vektory zrychlení do e-frame. Po odečtení tíhového zrychlení z osy Z dostaneme data na obrázku 6.2b.

Na obou grafech z obrázku 6.2 je patrně vidět problém čistě inerciální navigace, kde na konci měření není zrychlení, nebo natočení v osách X a Y nulové. To je způsobené kumulováním chyby při integraci dat. Toto bývá označováno jako gyro drift. Často jsou chyby potlačovány použitím kalmanova filtru, kde je drift gyroskopu kompenzován dlouhodobým měřením směru tíhového zrychlení, což je vhodné pro případ, že by naším cílem bylo určit pouze náklon v prostoru. Ovšem vzhledem k tomu, že použití filtrů přináší časovou prodlevu v ustálení měření orientace, tak jsou data akcelerometru stále krátkodobě natačena ve špatném směru, což po následné integraci na rychlost a trajektorii stále přináší velkou chybu.

Po následných dvou numerických integracích zrychlení můžeme výslednou trajektorii reprezentovat v 3D prostoru obrázkem 6.3, kde je v počátku mírně vidět tendence chůze v přímce, ale poté nastává integrací chyby úplné znehodnocení dat.



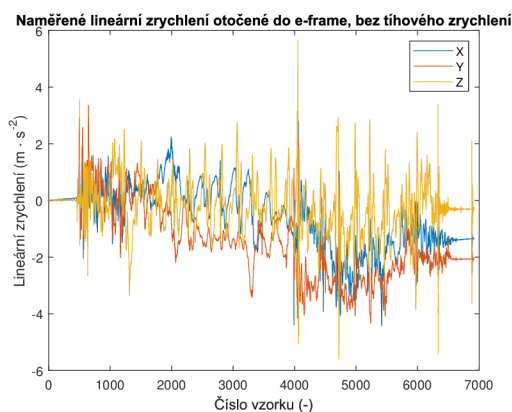
Obr. 6.3: Odhad trajektorie.

6.2 Vliv kalibrace senzorů

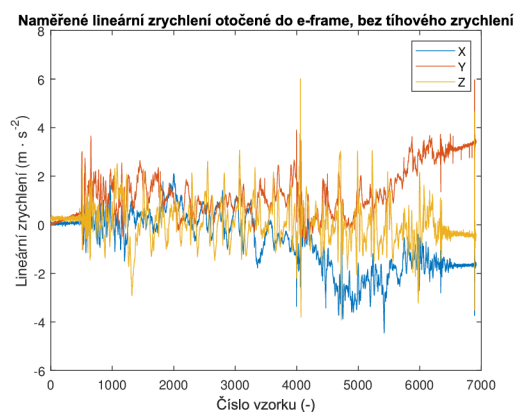
Za krátkou zmínku stojí i porovnání vlivu kalibrace senzorů na výsledná data. Na to můžeme použít skript z kapitoly 6.1. Přestože odhad trajektorie nemá moc velkou vypovídající hodnotu, tak přesnost senzorů můžeme alespoň částečně posoudit například tím, jak moc se změřené zrychlení v e-framu s odečteným tíhovým zrychlením na konci měření, když je zařízení v klidu, odchyluje od nuly. Byla provedena dvě obdobná měření při chůzi s jednotkou trajektorie stejného tvaru z předchozí kapitoly, bez kalibrace a s kalibrací.

Na obrázku 6.4 můžeme vidět porovnání dvou měření se zpracovanými daty z obou IMU. ADIS16505 dosahuje na konci měření maximální odchylku zrychlení zhruba 2 m/s^2 s i bez kalibrace. To odpovídá tomu, že se jedná o poměrně přesný a nákladný senzor. Po kalibraci jsou totiž konstanty gainu pro všechny osy akcelerometru většinou řádově rovny 0,99990 a lepší, není jej tedy potřeba téměř kalibrovat. Situace je ovšem jiná u IMU MPU6050, které bez kalibrace dosahuje dvojnásobné odchylky od ADIS16505, zatímco po kalibraci je jeho odchylka zhruba srovnatelná s přesnějším senzorem.

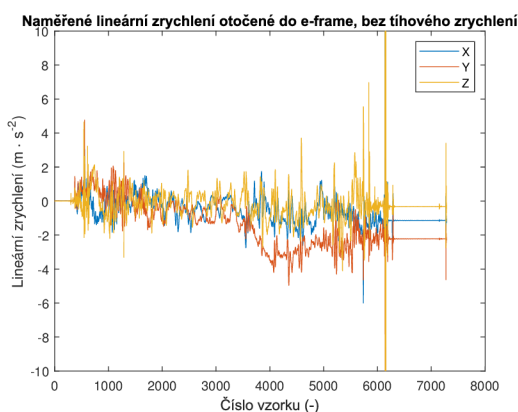
Z tohoto můžeme usoudit, že i takto poměrně jednoduchá metoda kalibrace má dost pozitivní dopady na výsledky měření levnějších IMU.



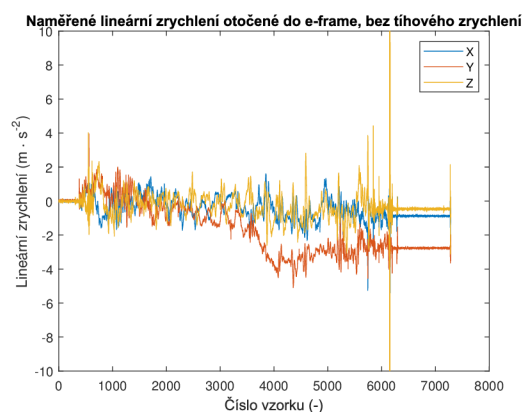
(a) ADIS16505 bez kalibrace.



(b) MPU6050 bez kalibrace.



(c) ADIS16505 s kalibrací.



(d) MPU6050 s kalibrací.

Obr. 6.4: Porovnání senzorů s kalibrací a bez kalibrace.

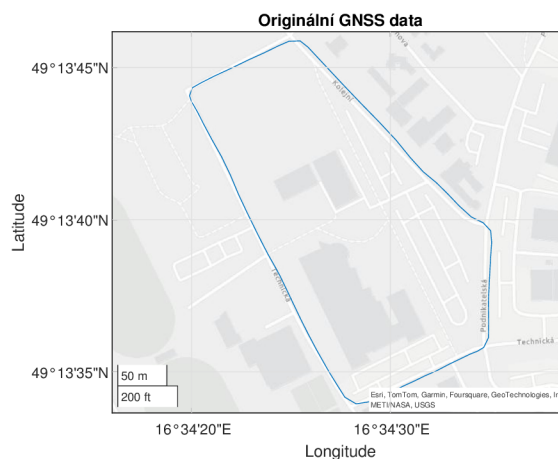
6.3 Fúze dat s GNSS

Na fúzi inerciálních dat byl použit filtr s připravenými modely senzorů akcelerometru, gyroskopu, magnetometru a GNSS. Na odhad polohy využívá rozšířený kalmanův filtr s 22 prvkovým stavovým vektorem, který kromě pozice sleduje i natočení, rychlost, biasy akcelerometru, gyroskopu a magnetometru a odhad magnetického pole Země. [30]

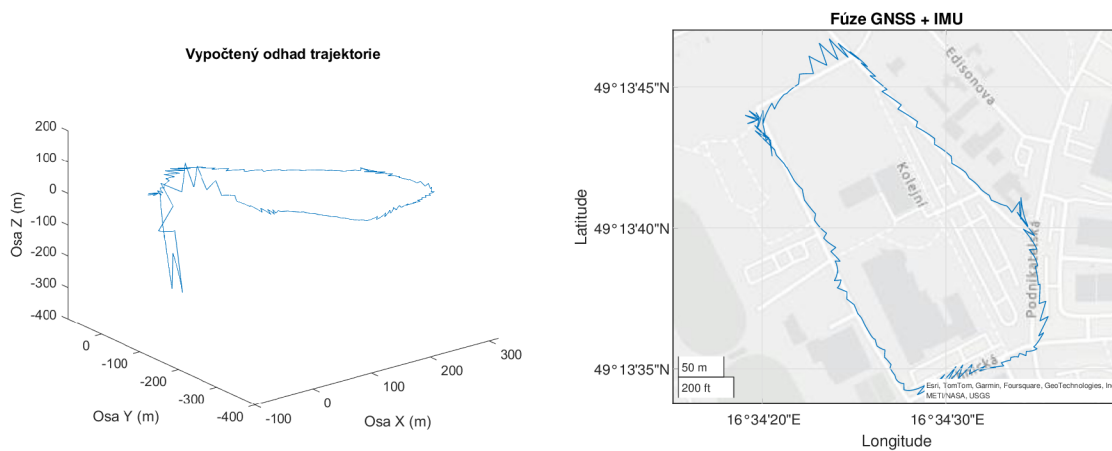
Pro zpracování dat byl použit vzorek měření jízdy autem ve venkovním prostředí, kde je příjem GNSS signálu spolehlivější. Na obrázku 6.5 jsou zobrazena naměřená data polohy GNSS přijímače, na obrázku 6.6 je zobrazen odhad polohy z fúze GNSS a IMU dat pomocí filtru `insFilterMARG`. Jak je vidět, tak výsledky nejsou optimální, zejména ve výchozí pozici měření, kde dochází k ustálení filtru.

Přesnost výsledků filtru je poměrně závislá na správném nastavení kovarianční matice 22 prvkového stavového vektoru a nastavení rozptylu šumu a biasu jednotlivých senzorů. Zde byly použity pouze odhady a výchozí hodnoty těchto veličin. Pro

správné fungování bude pravděpodobně potřeba tyto hodnoty přesně vyčíslit, což umožňují například některé ladicí funkce Navigation toolboxu, ty ovšem využívají buď modelů konkrétních použitých součástek, nebo měření se zařízením, se kterým bylo pohybováno po přesně definované trajektorii, což vzhledem k časové náročnosti takového měření a předchozích částí této práce nebylo realizováno.



Obr. 6.5: Naměřená GNSS data.



(a) Odhad trajektorie v 3D prostoru.

(b) Odhad trajektorie na mapě.

Obr. 6.6: Odhad trajektorie z fúze dat GNSS a IMU.

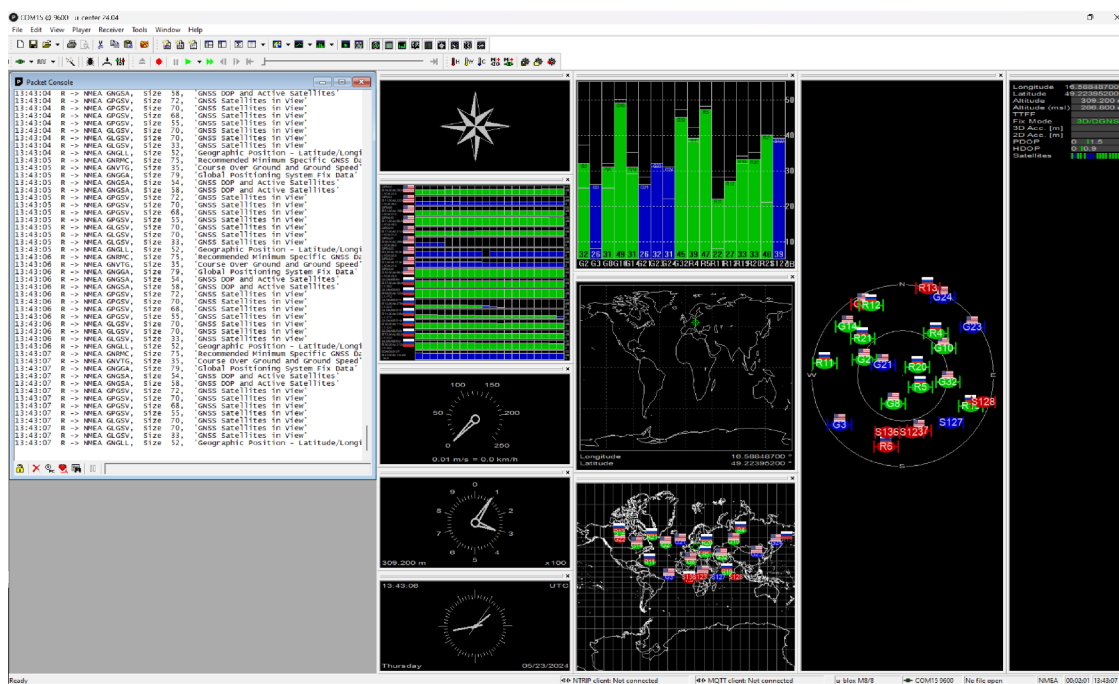
7 Koncepce laboratorní úlohy

Vzhledem k poměrně velké univerzálnosti vytvořeného hardwaru a obslužných skriptů jsou možnosti využití tohoto zařízení poměrně velké. Jako koncepci laboratorní úlohy můžeme zmínit dvě následující.

7.1 Analýza využití GNSS, dostupnost signálu a obsluha

Tato laboratorní úloha by se mohla týkat analýzy dostupnosti a využití GNSS signálů uvnitř a vně budov, obsluhy a fungování navigačních modulů.

Základní informace o stavu GNSS modulu v zařízení je možné zobrazovat na displeji jednotky. Také díky tomu, že je zapojený USB port GNSS modulu NEO-M8U do interního rozbočovače zařízení, lze využít například jakýkoliv terminál sériového rozhraní, nebo obslužnou aplikaci u-center od výrobce U-blox pro zobrazení dostupných satelitů navigačních systémů a jejich síly signálu, zaznamenat časovou závislost počtu satelitů od doby zapnutí zařízení, nebo faktory ovlivňující kvalitu GNSS signálu (překážky, atmosferické podmínky...). Použitý GNSS modul podporuje všechny světové navigační systémy, lze tedy zaznamenat i počet dostupných satelitů jednotlivých systémů (GPS, BeiDou, Glonass, Galileo). Laboratorní úloha se také může věnovat dekódování NMEA zpráv, nebo určení přesnosti změřené polohy.



Obr. 7.1: Obslužná aplikace u-center.

7.2 Princip fungování IMU, vliv kalibrace a určení náklonu a polohy inerciální jednotkou

Tato laboratorní úloha může být zaměřena na popsání principu fungování IMU, jaké veličiny lze měřit pomocí inerciálních senzorů, vliv a způsob kalibrace, zobrazení naměřených dat a jednotlivých dílčích veličin (zrychlení, orientace, rychlost, úhel), faktory ovlivňující nepřesnost výsledku a fúze dat s ostatními senzory.

Pomocí zobrazených veličin na displeji lze zobrazit všechny měřené údaje zařízením. Můžeme jednoduše znázornit měření tíhového pole akcelerometrem v případě, že je jednotka v klidu. Následně lze pohybem se zařízením reprezentovat změny úhlové rychlosti a zrychlení. Také je možné výpočtem velikosti vektoru tíhového zrychlení otestovat přesnost senzorů před a po kalibraci, popřípadě i kalibrační konstanty vypočítat ručně pomocí dodaného kalibračního skriptu v MATLABu. Dále je možné reprezentovat z naměřených dat jednotlivé veličiny po integraci z čistě inerciální navigace a problémy spjaté s její nepřesností. Také můžeme využít skriptu na fúzi dat a otestovat její fungování, na to by bylo ovšem vhodné ještě zpracování dat vylepšit, abychom dosáhli uspokojivých výsledků.

Závěr

V rámci bakalářské práce byla popsána kinematika pohybu a nakládání s veličinami změřenými IMU pro potřeby výpočtu polohy. Také bylo definováno několik vztažných soustav a postupy pro převod mezi nimi. Je rozebráno tíhové pole Země, gravitační modely a jejich význam v inerciální navigaci.

Byl popsán funkční princip IMU a společně s GNSS modulem s možností inerciální navigace byly vyzkoušeny a otestovány pomocí běžně dostupných vývojových stavebnic.

Práce se také věnuje návrhu obvodového zapojení inerciální jednotky, definováním minimálních požadavků na hlavní MCU tak, abychom nebyli v budoucnu omezeni některým z rozhodnutí při návrhu hardwaru. Inerciální jednotka byla osazena i jinými senzory než gyroskopy a akcelerometry pro možnou sensorickou fúzi. Dále byl popsán návrh samotné DPS v programu KiCad, jejíž výkresy a schéma jsou v příloze. Je popsána konstrukce celého zařízení, byla vytvořena 3D tištěná krabička a postup oživení.

Další část práce se věnuje vývoji firmwaru pro MCU a použitým nástrojům. Funkční bloky firmwaru byly rozloženy do několika stavových automatů pro větší přehlednost a jejich funkcionality popsána. Také byl vytvořen skript pro převod binárních dat do jednoduše čitelného souborového formátu. Předposlední kapitola práce je věnována vytvořeným skriptům na zpracování dat z čistě inerciálních senzorů a jeho omezením a vlivu kalibrace na výsledek. Také byla otestována fúze dat z GNSS modulu, u které je pravděpodobně možné dosáhnout řádově lepších výsledků navazujícím výzkumem. V elektronické příloze jsou také vzorová data z různých měření pro případ, že by se někdo chtěl zabývat jejich zpracováním bez toho, aby měl zařízení fyzicky u sebe.

V poslední kapitole jsou krátce shrnuty dvě možné koncepce využití zařízení pro laboratorní úlohu, ovšem vzhledem k univerzálnosti vytvořeného zařízení se můžou najít i jiné oblasti jeho využití.

Literatura

- [1] TITTERTON, D. H. a WESTON, J. L. *Strapdown inertial navigation technology*. Second edition. Progress in astronautics and aeronautics, 207. Reston, VA: American Institute of Aeronautics and Astronautics, c2004. ISBN 1-56347-693-2.
- [2] GREWAL, Mohinder S.; ANDREWS, Angus P. a BARTONE, Chris. *Global navigation satellite systems, inertial navigation, and integration*. Third edition. Hoboken, New Jersey: John Wiley, 2013. ISBN 978-1-118-44700-0.
- [3] POLÁK, Luboš. *Navigační jednotka s modifikovanou soustavou akcelerometrů*. Online, Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2018. Dostupné z: <http://hdl.handle.net/10467/74430>. [cit. 2023-12-25].
- [4] PEKÁREK, David. *Přesné měření vlastní trajektorie vozidla*. Online, Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta elektrotechnická, 2020. Dostupné z: <http://hdl.handle.net/10467/90221>. [cit. 2023-12-25].
- [5] HALLIDAY, David; RESNICK, Robert a WALKER, Jearl. *Fyzika: vysokoškolská učebnice obecné fyziky*. Překlady vysokoškolských učebnic. Brno: VUTIUM, 2000. ISBN 80-214-1869-9.
- [6] PAVLIS, Nikolaos K.; HOLMES, Simon A.; KENYON, Steve C. a FACTOR, John K. The development and evaluation of the Earth Gravitational Model 2008 (EGM2008). Online. *Journal of Geophysical Research: Solid Earth*. 2012, roč. 117, č. B4, s. 1-38. ISSN 0148-0227. Dostupné z: <https://doi.org/10.1029/2011JB008916>. [cit. 2023-12-26].
- [7] BEZDĚK, Aleš a SEBERA, Josef. Matlab script for 3D visualizing geodata on a rotating globe. Online. *Computers and Geosciences*. 2013, roč. 56, s. 127-130. ISSN 00983004. Dostupné z: <https://doi.org/10.1016/j.cageo.2013.03.007>. [cit. 2023-12-26].
- [8] DADAFSHAR, Majid. *APPLICATION NOTE 5830: ACCELEROMETER AND GYROSCOPES SENSORS: OPERATION, SENSING, AND APPLICATIONS*. Online aplikační poznámka. 2014. Dostupné z: <https://pdfserv.maximintegrated.com/en/an/AN5830.pdf>. [cit. 2023-12-27].

- [9] MNX. *What is MemS Technology?* Online. MEMS and Nanotechnology Exchange. Dostupné z: <https://www.mems-exchange.org/MEMS/what-is.html>. [cit. 2023-12-27].
- [10] BLOCHER, Lukas; MAYER, Wolfram; ARENA, Marco; RADOVIC, Dusan; HILLER, Tobias et al. Purely Inertial Navigation with a Low-Cost MEMS Sensor Array. Online. In: *2021 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL)*. IEEE, 2021, s. 1-4. ISBN 978-1-7281-5099-4. Dostupné z: <https://doi.org/10.1109/INERTIAL51137.2021.9430468>. [cit. 2023-12-09].
- [11] TDK INVENSENSE. *MPU6050: Product specification*. Online katalogový list. 3.4. 2013. Dostupné z: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. [cit. 2023-12-12].
- [12] WEI, Y. a LI, Y. IMPACT OF SENSOR DATA SAMPLING RATE IN GNSS/INS INTEGRATED NAVIGATION WITH VARIOUS SENSOR GRADES. Online. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2022, roč. XLVI-3/W1-2022, s. 205-211. ISSN 2194-9034. Dostupné z: <https://doi.org/10.5194/isprs-archives-XLVI-3-W1-2022-205-2022>. [cit. 2023-12-16].
- [13] ANALOG DEVICES. *ADIS16505: Precision, Miniature MEMS IMU*. Online katalogový list. C. 2020. Dostupné z: <https://www.analog.com/media/en/technical-documentation/data-sheets/adis16505.pdf>. [cit. 2023-12-12].
- [14] TKHORENKO, M. Yu.; PAVLOV, B. V.; KARSHAKOV, E. V. a VOLKOVITSKY, A. K. On integration of a strapdown inertial navigation system with modern magnetic sensors. Online. In: *2018 25th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*. IEEE, 2018, s. 1-4. ISBN 978-5-91995-057-8. Dostupné z: <https://doi.org/10.23919/ICINS.2018.8405845>. [cit. 2023-12-17].
- [15] ST MICROELECTRONICS. *LSM303AGR: Ultracompact high-performance eCompass module*. Online katalogový list. 11. 2022. Dostupné z: <https://www.st.com/resource/en/datasheet/lsm303agr.pdf>. [cit. 2023-12-12].
- [16] U-BLOX. *NEO-M8U: u-blox M8 untethered dead reckoning module including 3D inertial sensors*. Online katalogový list. R13. 2022. Dostupné z: https://content.u-blox.com/sites/default/files/NEO-M8U_DataSheet_UBX-15015679.pdf. [cit. 2023-12-17].

- [17] XTX TECHNOLOGY LIMITED. *XT25F256BWSIGT: Quad IO Serial NOR Flash Datasheet*. Online katalogový list. 2020. Dostupné z: <http://www.xtxtech.com/download/?Aid=287>. [cit. 2023-12-17].
- [18] KRÄWINKEL, R.W. *The effect of writing and transmitting SD card data on the consistency of SD card write performance*. Online, bakalářská. Enschede, Holandsko: University of Twente, 2020. Dostupné z: http://essay.utwente.nl/82256/1/Krawinkel_BA_EEMCS.pdf. [cit. 2023-12-17].
- [19] TEXAS INSTRUMENTS. *BQ2407x: Standalone 1-Cell 1.5-A Linear Battery Chargers with Power Path*. Online katalogový list. N. 2021. Dostupné z: <https://www.ti.com/lit/gpn/bq24075>. [cit. 2023-12-21].
- [20] TEXAS INSTRUMENTS. *TPS6282x: 5.5-V, 1-A, 2-A, 3-A Step-Down Converter Family with 1% Accuracy*. Online katalogový list. C. 2019. Dostupné z: <https://www.ti.com/lit/gpn/tps62823>. [cit. 2023-12-21].
- [21] *STM32F446xC/E: Arm® Cortex®-M4 32-bit MCU+FPU, 225 DMIPS, up to 512 KB Flash/128+4 KB RAM, USB OTG HS/FS, seventeen TIMs, three ADCs and twenty communication interfaces*. Online katalogový list. 10. 2021. Dostupné z: <https://www.st.com/resource/en/datasheet/stm32f446ve.pdf>. [cit. 2023-12-14].
- [22] ESPRESSIF SYSTEMS. *ESP32C3MINI1: Smallsized 2.4 GHz WiFi (802.11 b/g/n) and Bluetooth® 5 module*. Online katalogový list. 2022. Dostupné z: https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1_datasheet_en.pdf. [cit. 2023-12-21].
- [23] CHERUIYOT, Fabian; SEGERA, Davies a OSORIO DE LA ROSA, Edith. A Master-Slave Salp Swarm Algorithm Optimizer for Hybrid Energy Storage System Control Strategy in Electric Vehicles. Online. *Journal of Energy*. 2022, roč. 2022, s. 1-20. ISSN 2314-615X. Dostupné z: <https://doi.org/10.1155/2022/1648433>. [cit. 2024-05-13].
- [24] ST MICROELECTRONICS. *UM2609: STM32CubeIDE user guide*. Online uživatelský návod. 11. 2024. Dostupné z: https://www.st.com/resource/en/user_manual/um2609-stm32cubeide-user-guide-stmicroelectronics.pdf. [cit. 2024-05-14].
- [25] ZHU, Ming-Yuan. *Understanding FreeRTOS: A Requirement Analysis*. Online. ResearchGate, 2011. Dostupné z: <https://doi.org/10.13140/RG.2.2.12419.09767>. [cit. 2024-05-14].

- [26] SIMPLEMENTHOD. *STM32 library with DMA support for u-blox devices supporting Global Navigation Satellite Systems and UBX standard*. Online. Neznaležený vydavatel. 2021. Dostupné z: <https://github.com/SimpleMethod/STM32-GNSS>. [cit. 2024-05-16].
- [27] ALEKSEEV, Aleksander. *STM32 library for working with OLEDs based on SSD1306*. Online. GitHub. 2024. Dostupné z: <https://github.com/afiskon/stm32-ssd1306>. [cit. 2024-05-16].
- [28] ST MICROELECTRONICS. *AN4508: Parameters and calibration of a low-g 3-axis accelerometer*. Online aplikační poznámka. 2014. Dostupné z: https://www.st.com/resource/en/application_note/an4508-parameters-and-calibration-of-a-lowg-3axis-accelerometer-stmicroelectronics.pdf. [cit. 2023-05-20].
- [29] FREESCALE SEMICONDUCTOR. *AN4399: High-Precision Calibration of a Three-Axis Accelerometer*. Online aplikační poznámka. 2. 2015. Dostupné z: https://www.nxp.com/docs/en/application_note/AN4399.pdf. [cit. 2023-05-20].
- [30] MATHWORKS. *Navigation Toolbox User's Guide*. Online. 24.1. 2024. Dostupné z: https://www.mathworks.com/help/pdf_doc/nav/nav_ug.pdf. [cit. 2024-05-22].

Seznam symbolů a zkratek

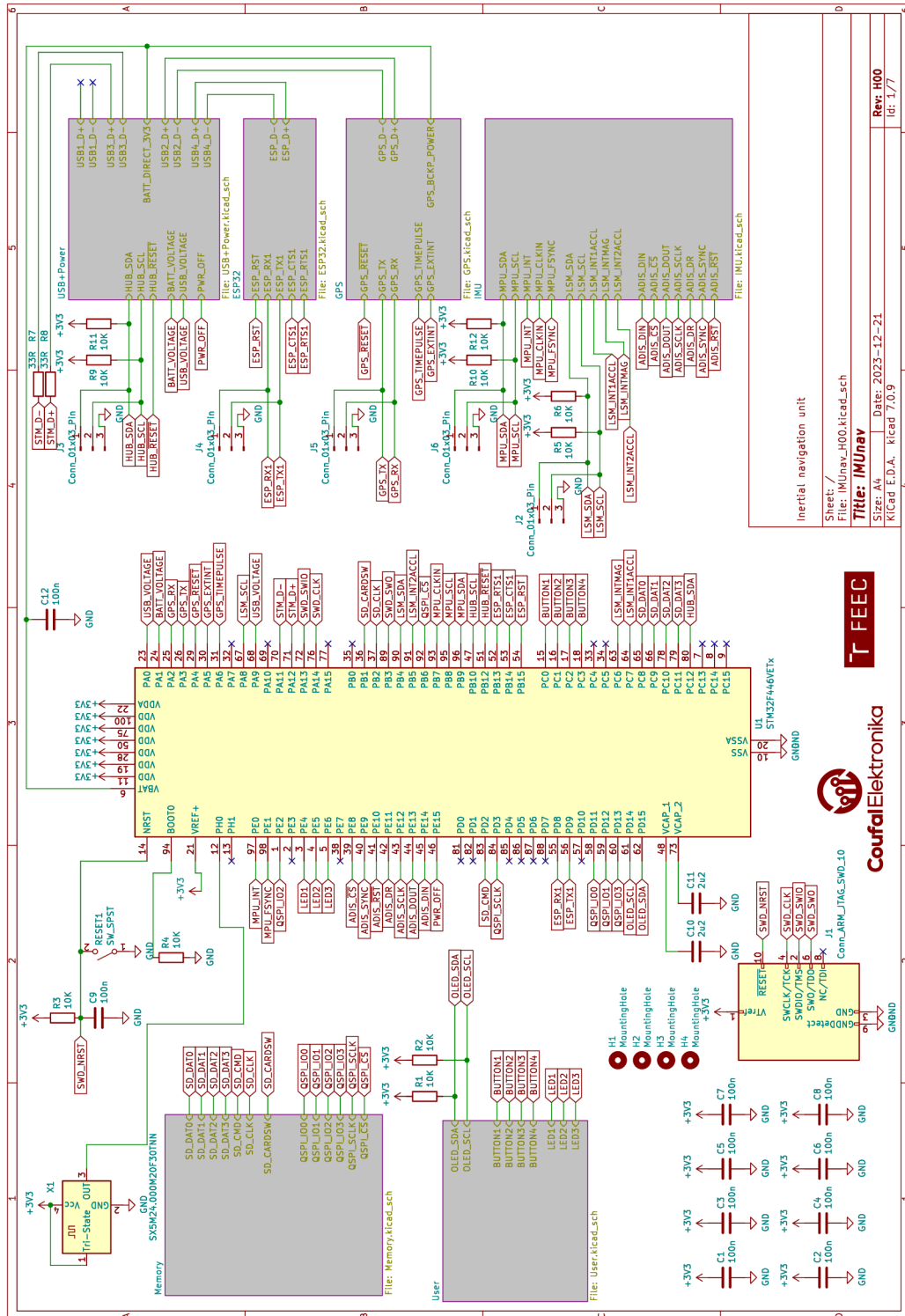
ADC	Analog to Digital Convertor - analogově-digitální převodník
BGA	Ball Grid Array - pouzdro matice kuliček
CAD	Computer-Aided Design - počítačem podporované projektování
CSV	Comma Separated Value - čárkou oddělené hodnoty
DMA	Direct Memory Acces - přímý přístup k paměti
DoF	Degrees of Freedom - stupně volnosti
DPS	Deska Plošných Spojů
EGM	Earth Gravitational Model - gravitační model Země
FIFO	First In First Out - první dovnitř, první ven
GNSS	Global navigation satellite system - globální družicový polohový systém
GPIO	General Purpose Input/Output - univerzální vstupní/výstupní pin
GPS	Global Positioning System - globální polohový systém
GUI	Graphical User Interface - grafické uživatelské rozhraní
HAL	Hardware Abstraction Layer - vrstva abstrakce hardwaru
I2C	Inter-Integrated Circuit - mezi obvodová komunikace
IDE	Integrated Development Enviroment - vývojové prostředí
IMU	Inertial Measurement Unit - měřicí inerciální jednotka
IMU	Inertial Measurement Unit - inerciální měřicí jednotka
MCU	Microcontroller Unit - mikrokontrolér
MEMS	Micro-ElectroMechanical Systems - mikro-elektromechanické systémy
MSC	Mass Storage Class - třída paměťového média
NED	North East Down - sever východ dolů
NGA	National Geospatial-Intelligence Agency - americká geoprostorová agentura

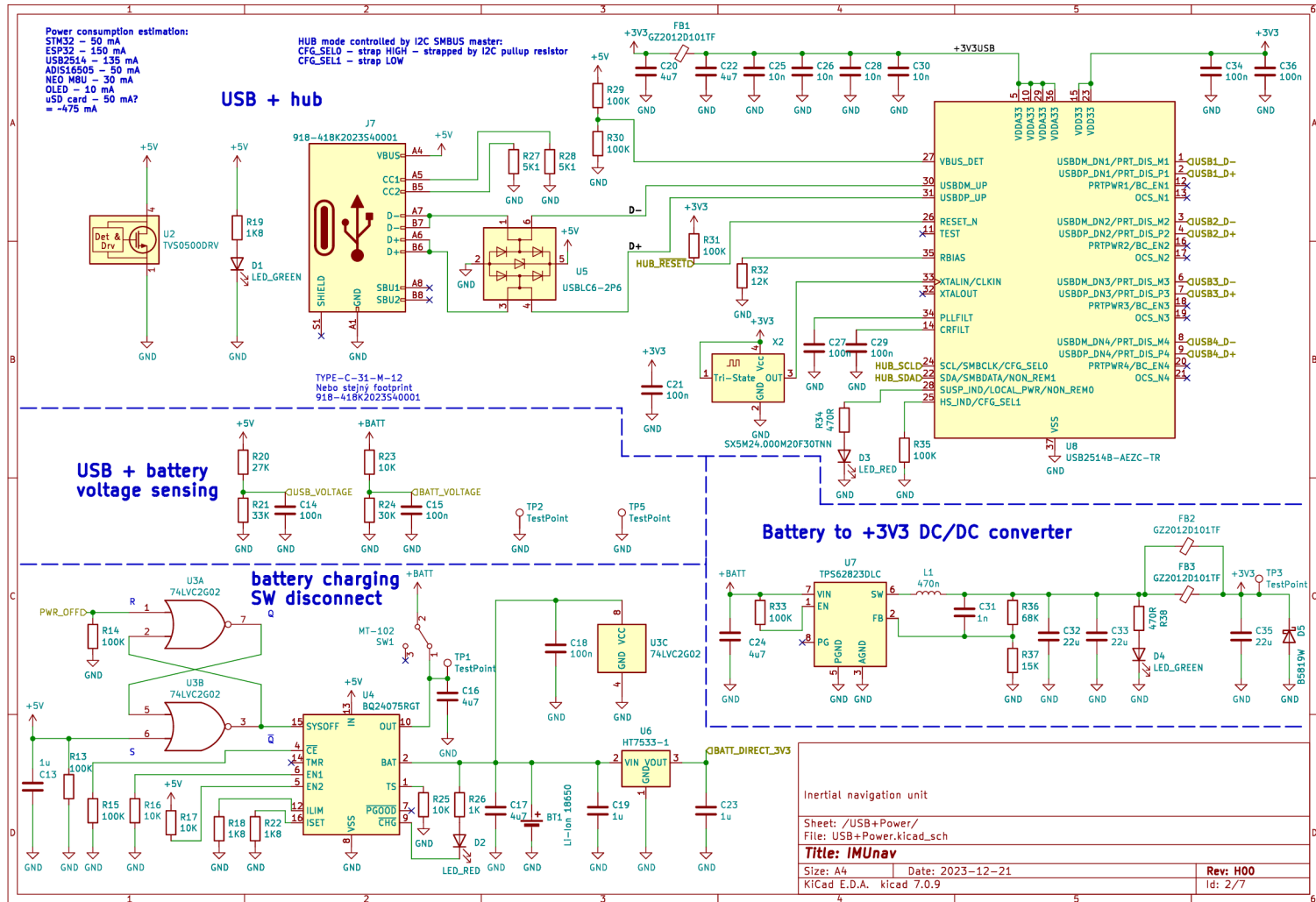
NMEA	National Marine Electronics Association - americká organizace námořní elektroniky
OLED	Organic Light-Emitting Diode - organický LED
PID	Product ID - identifikace produktu
RAM	Random Access Memory - paměť pro náhodný přístup
RTC	Real Time Clock - obvod reálného času
RTOS	Real Time Operating System - Operační systém reálného času
SMA	SubMiniature version A - subminiaturní verze A
SPI	Serial Peripheral Interface - sériové periferní rozhraní
UART	Universal Asynchronous Receiver-Transmitter - univerzální asynchronní přijímač-vysílač
UBX	UBloX message - zpráva formátu UBLOX
USB	Universal Serial Bus - univerzální sériová sběrnice
VID	Vendor ID - identifikace výrobce

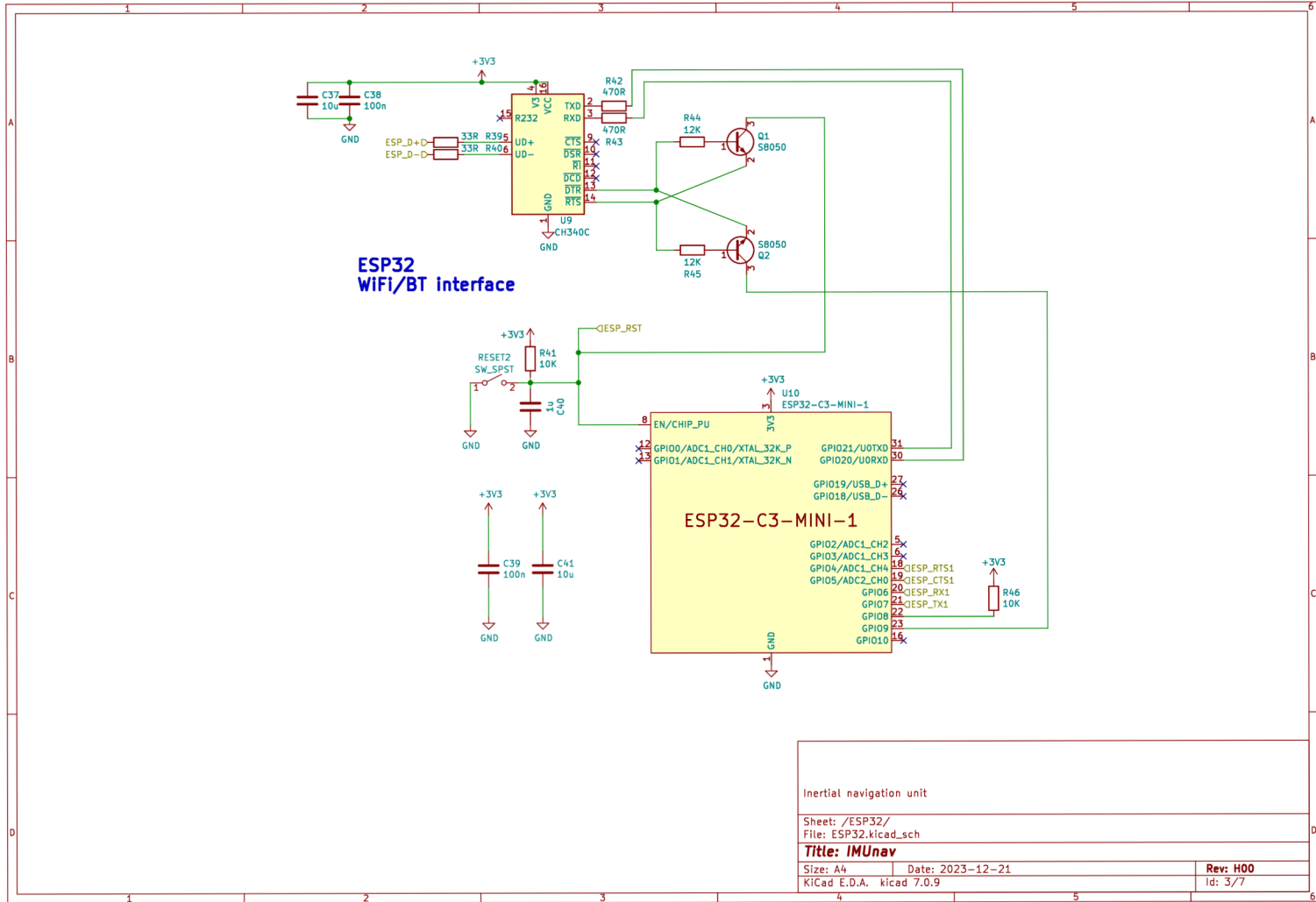
Seznam příloh

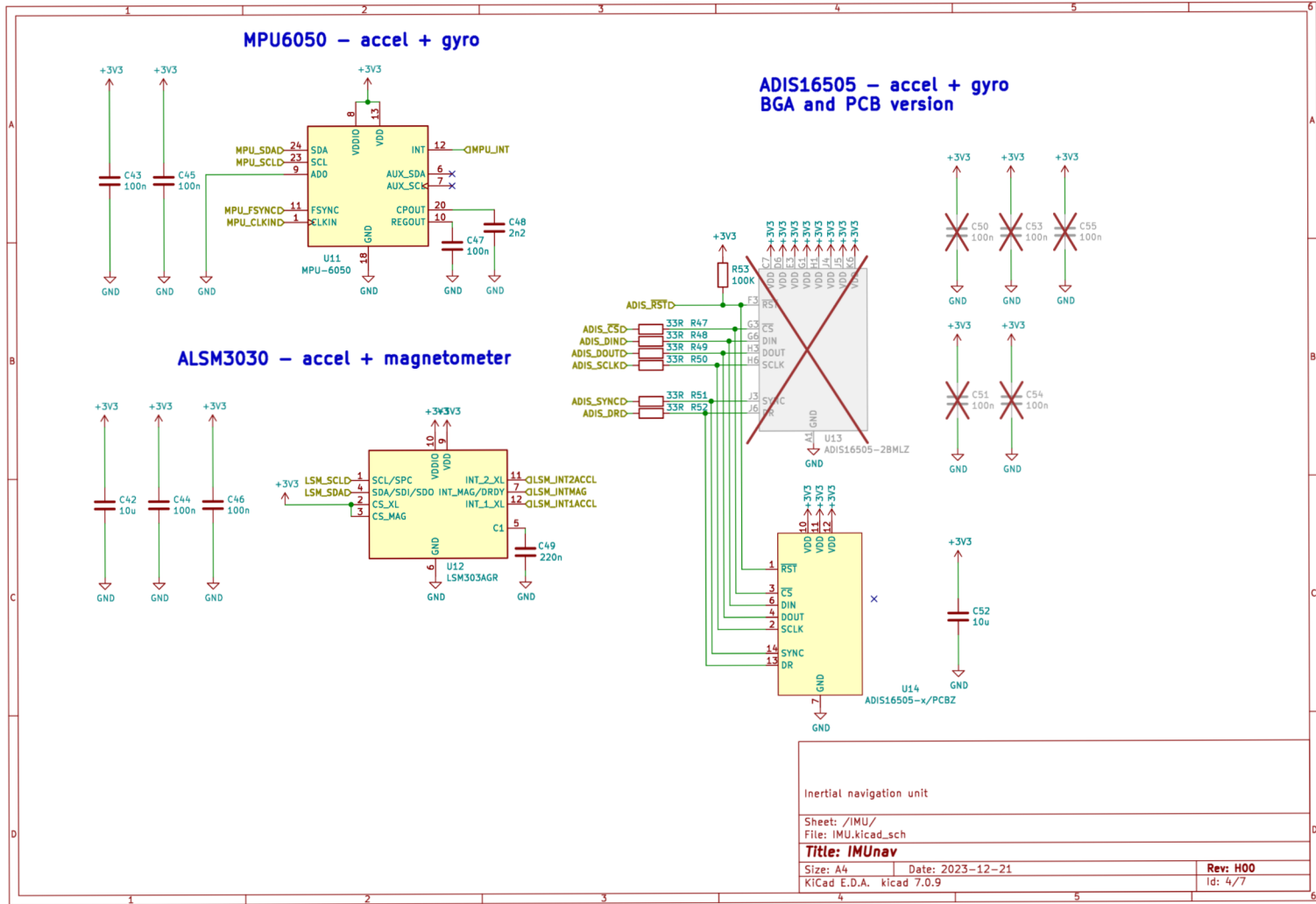
A	Schéma zapojení inerciální jednotky	63
B	Výkres DPS	70
B.1	Pohled osazení součástek	70
B.2	Vrchní vrstva mědi DPS	71
B.3	Vnitřní vrstva mědi DPS In1	72
B.4	Vnitřní vrstva mědi DPS In2	73
B.5	Spodní vrstva mědi DPS	74
C	Obsah elektronické přílohy	75

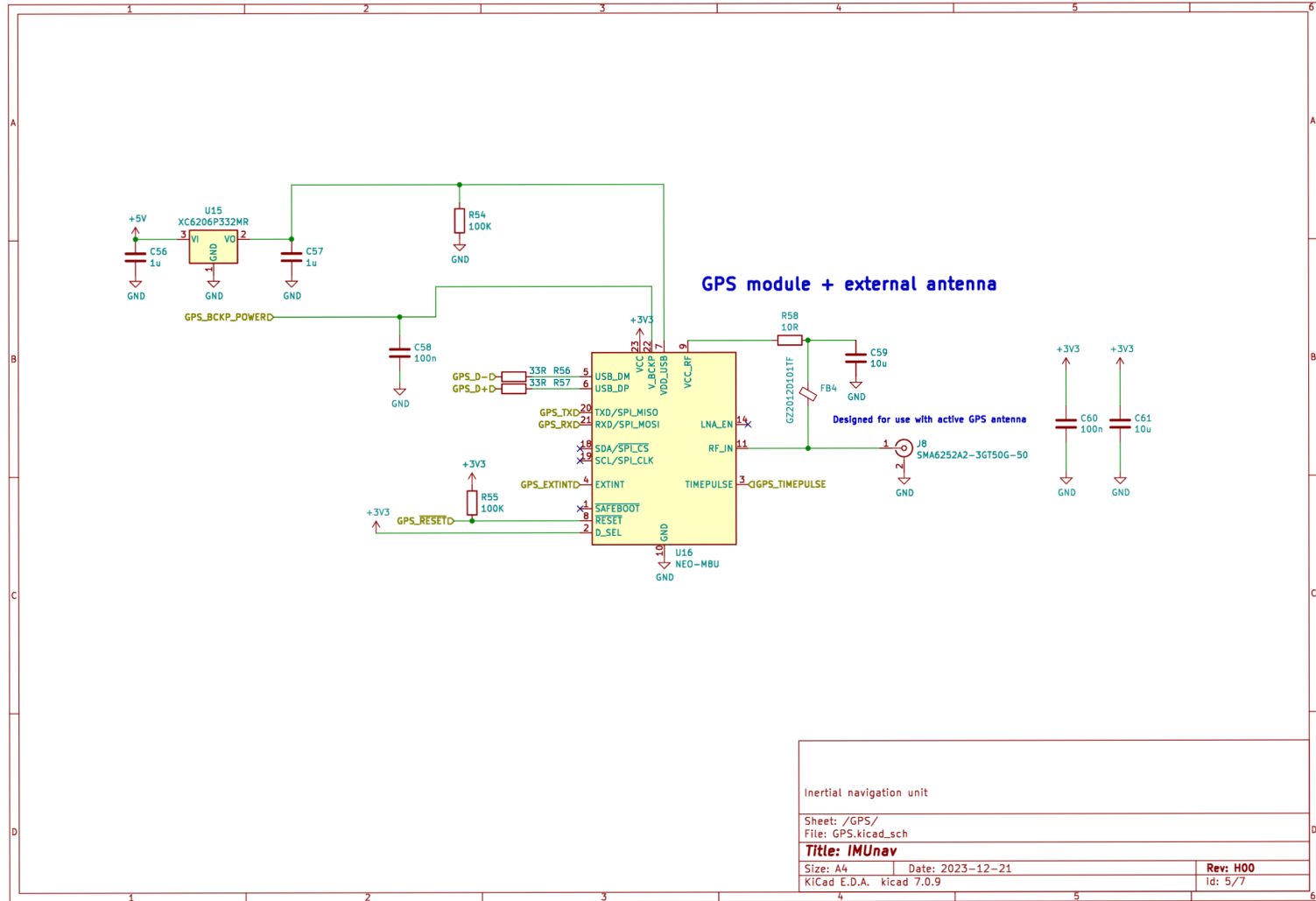
A Schéma zapojení inerciální jednotky

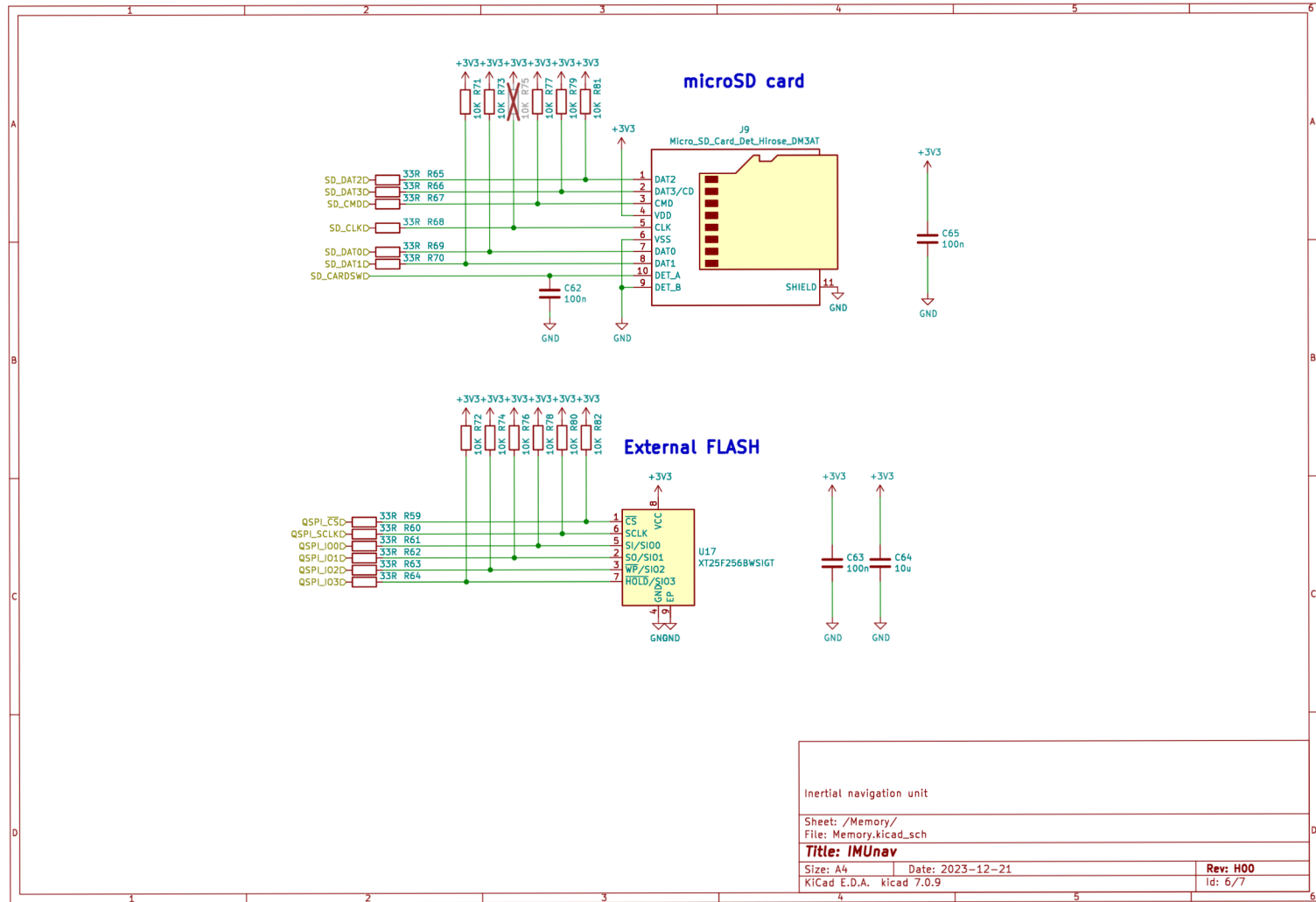




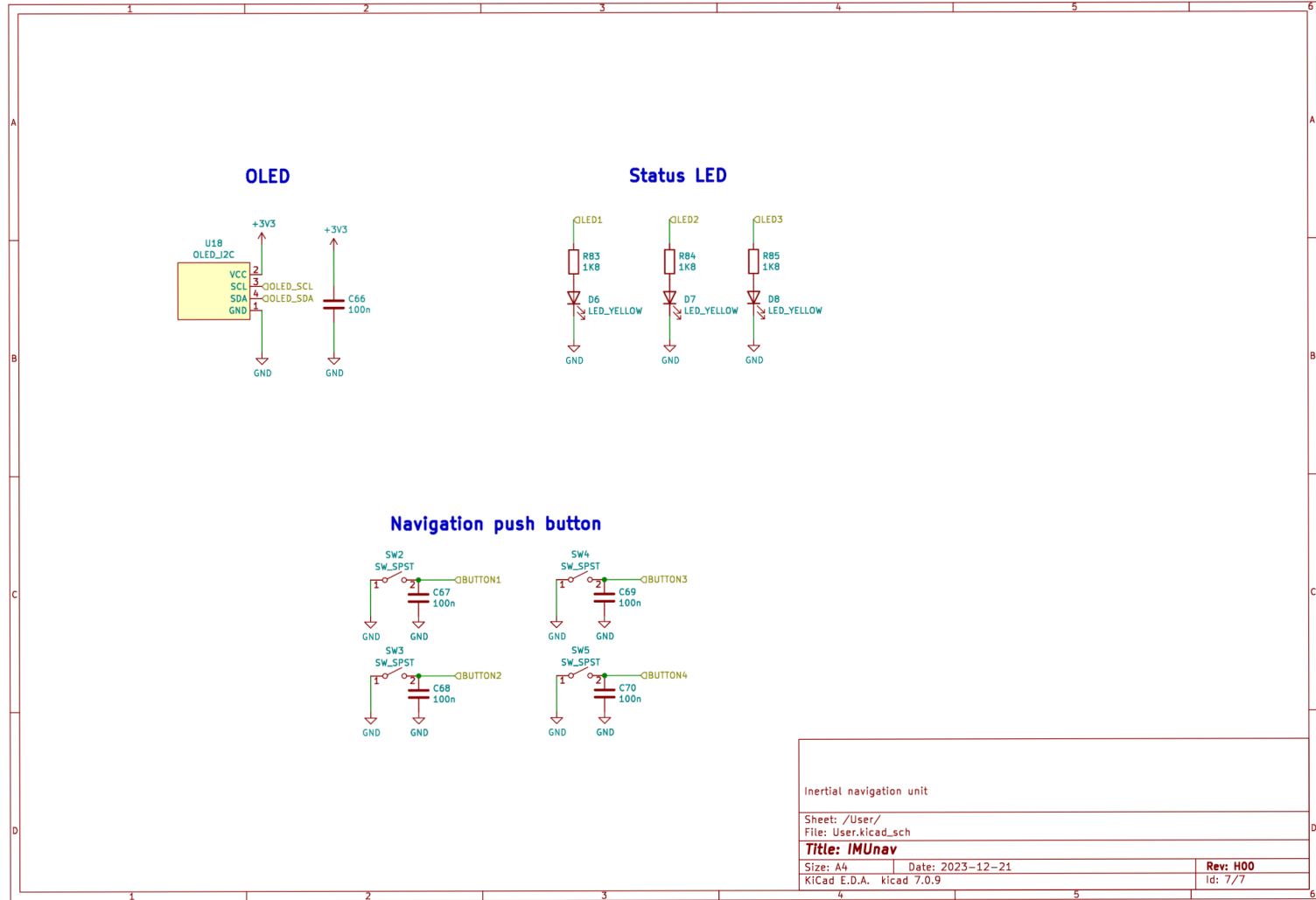






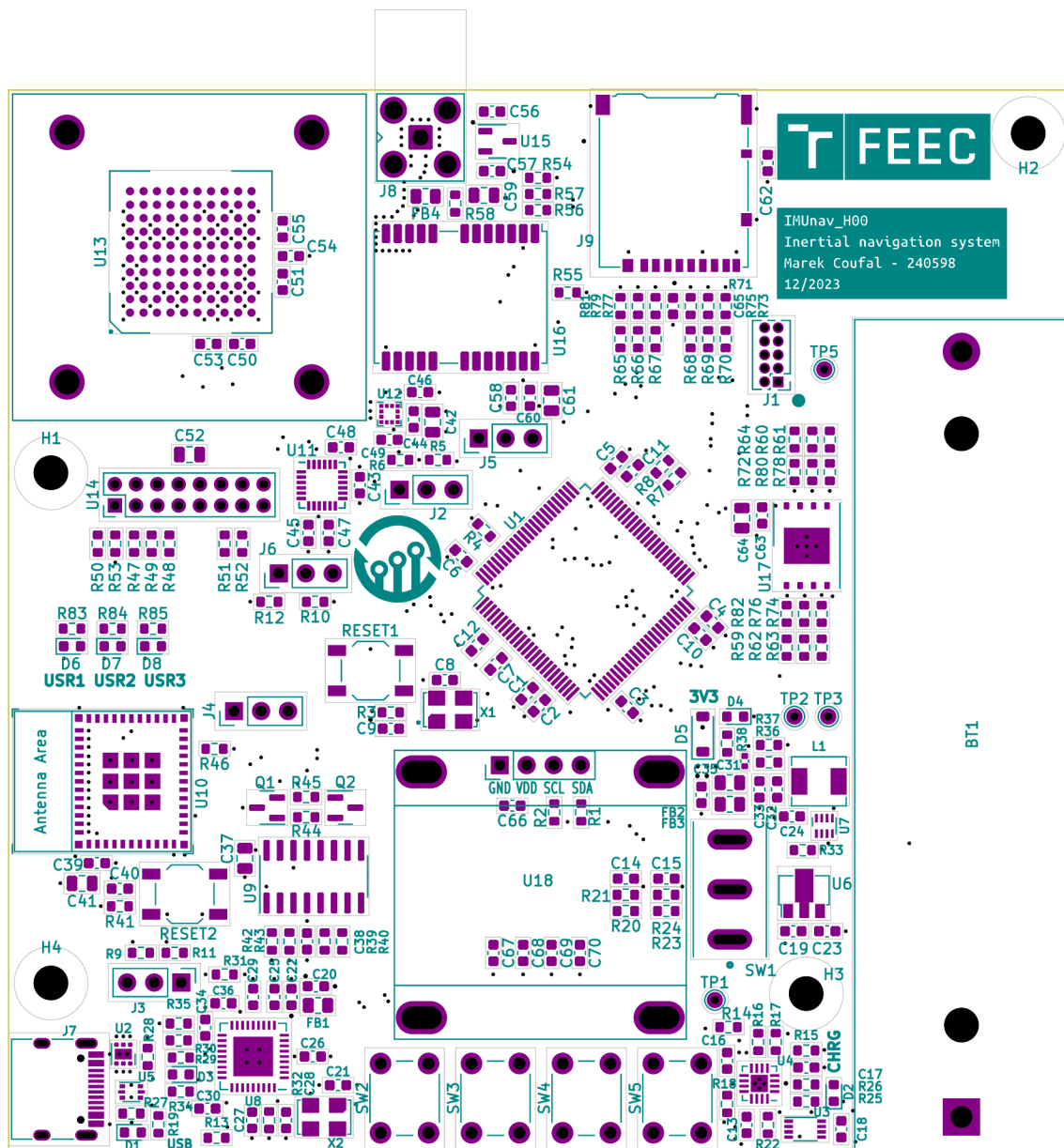


Inertial navigation unit		
Sheet: /Memory/		
File: Memory.kicad_sch		
Title: IMUnav		
Size: A4	Date: 2023-12-21	Rev: H00
KICad E.D.A. kicad 7.0.9		Id: 6/7

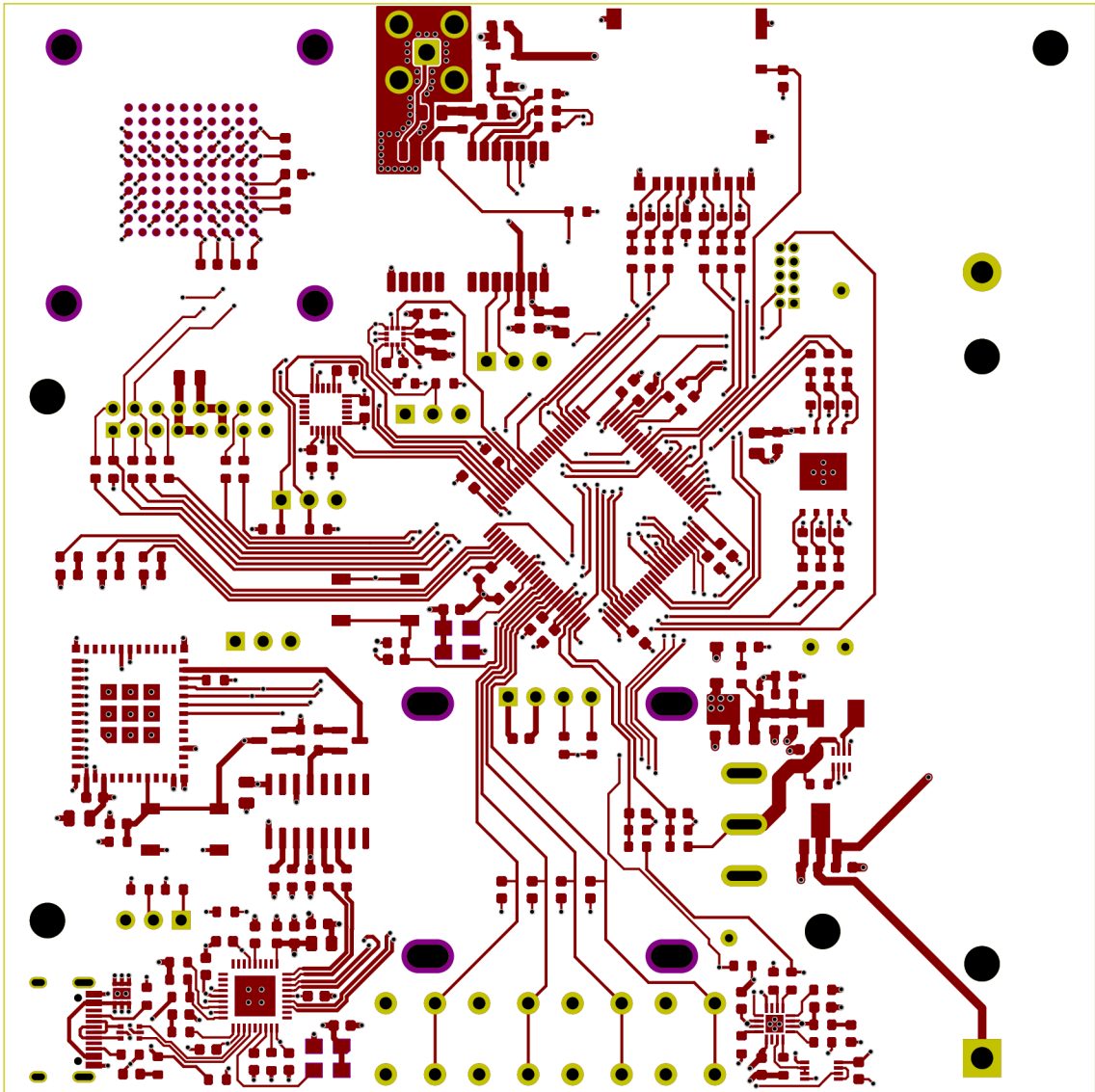


B Výkres DPS

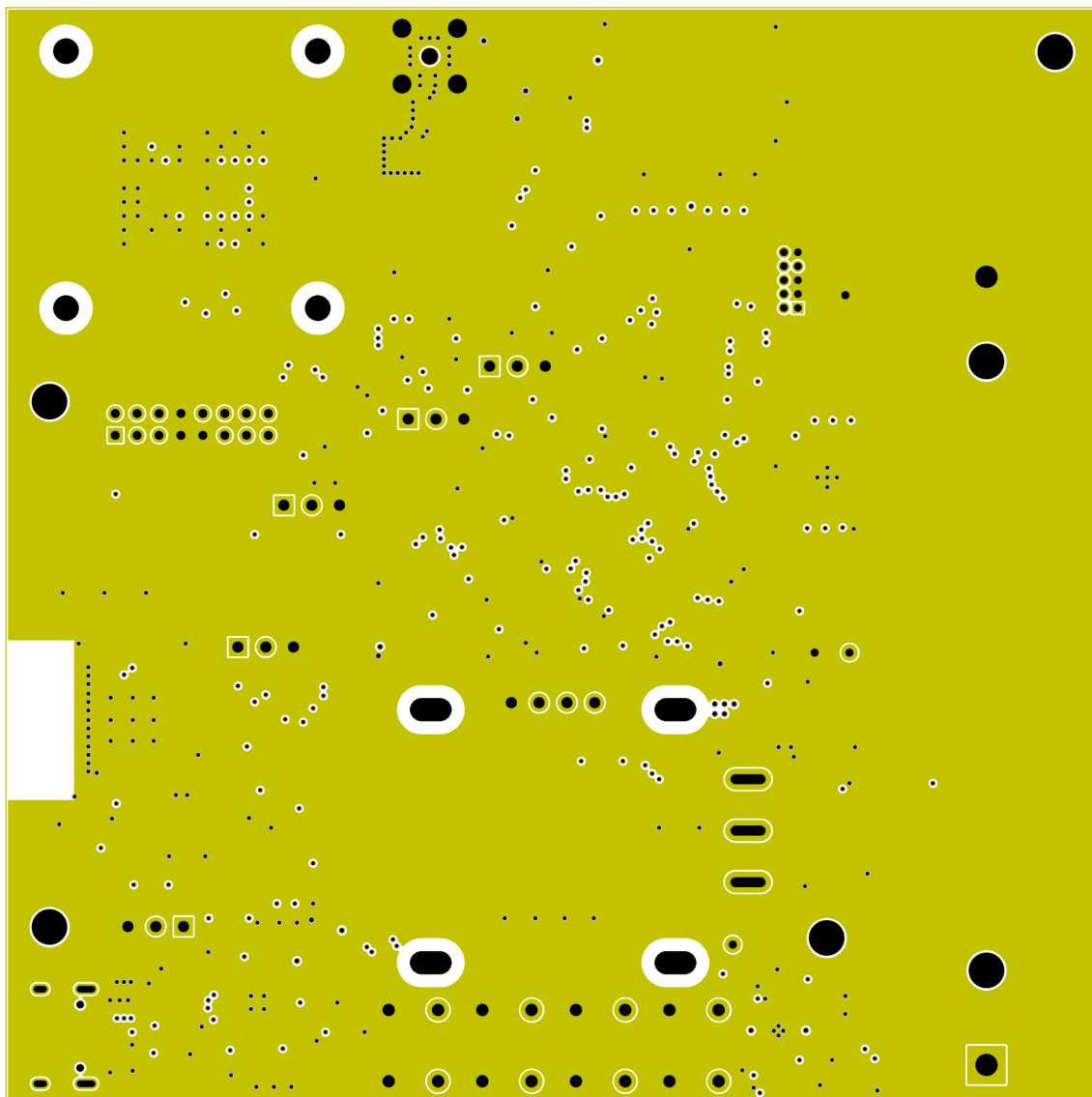
B.1 Pohled osazení součástek



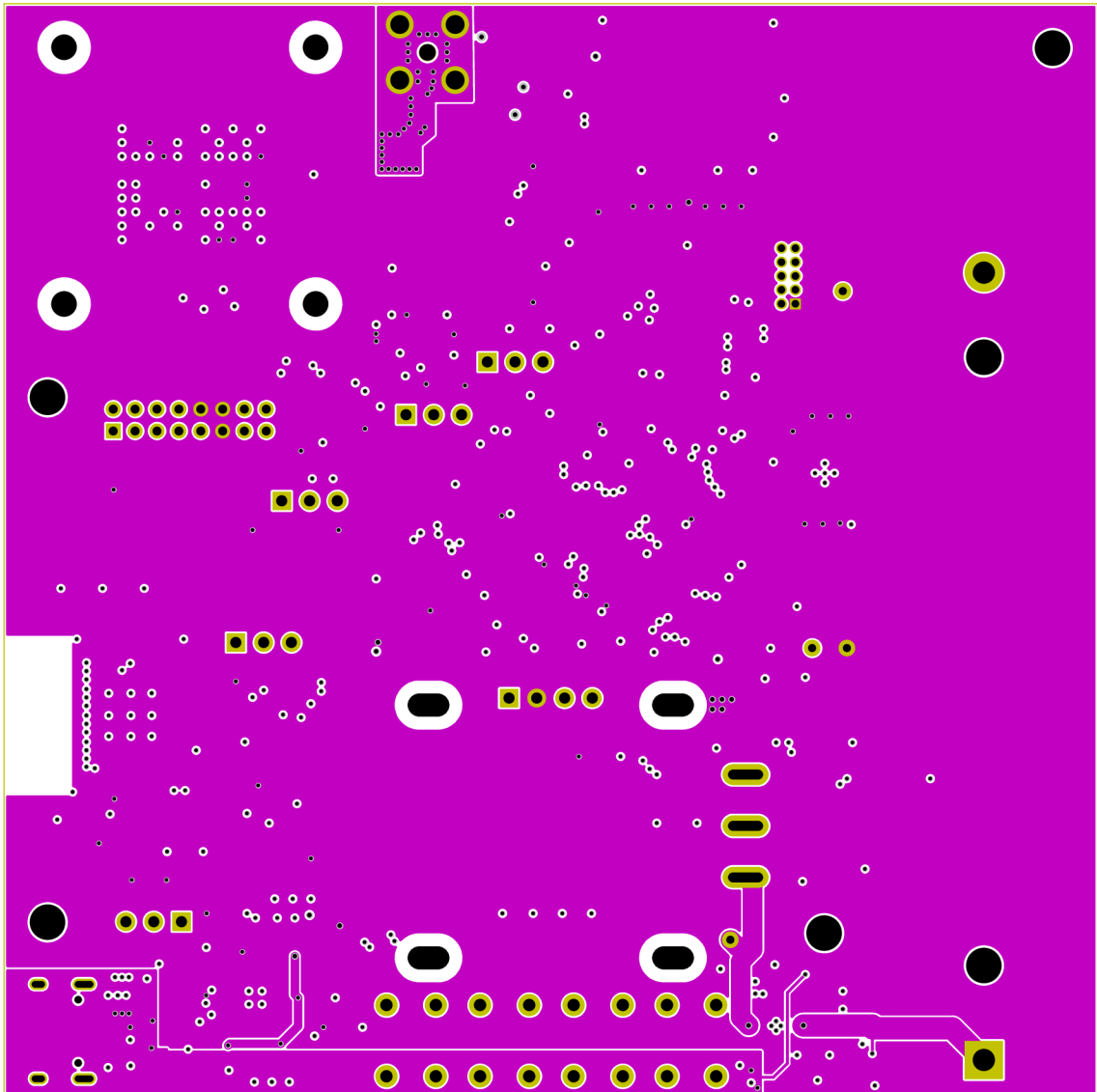
B.2 Vrchní vrstva mědi DPS



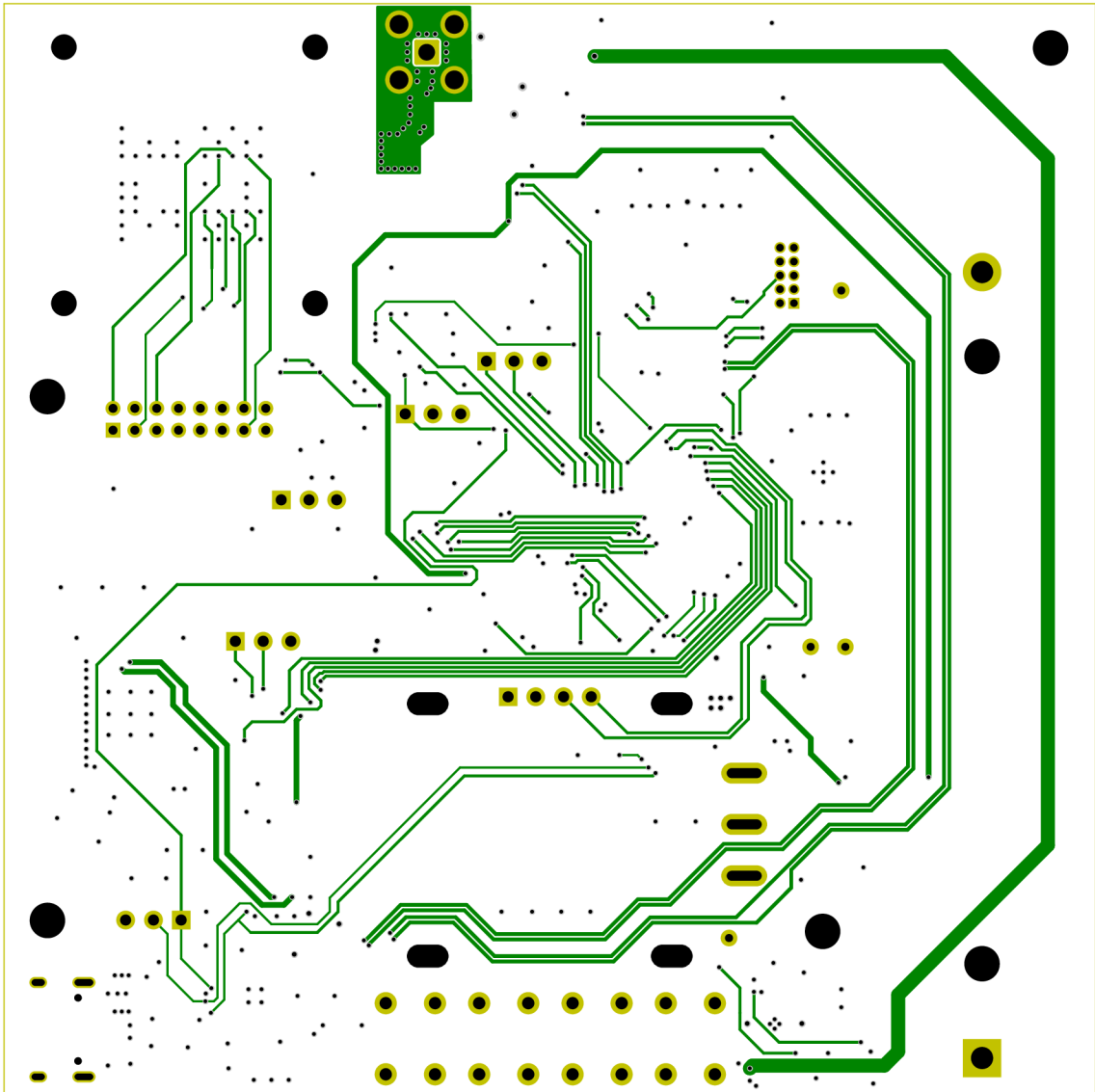
B.3 Vnitřní vrstva mědi DPS In1



B.4 Vnitřní vrstva mědi DPS In2



B.5 Spodní vrstva mědi DPS



C Obsah elektronické přílohy

— Firmware	
— IMUnav_STM_F00	složka projektu Firmwaru MCU
— Core	
— AccelCalSolver	vygenerované soubory kalibrační procedury
— ...	
— Inc	hlavičkové soubory
— ...	
— Src	zdrojové soubory
— ...	
— Debug	
— ...	
— Drivers	HAL knihovny
— ...	
— FATFS	knihovna souborového systému
— ...	
— Middlewares	zdrojové soubory pro FatFS, USB a FreeRTOS
— ...	
— Release	
— ...	
— USB_DEVICE	knihovna pro USB zařízení
— ...	
— IMUnav_STM_F00 Debug.launch	
— IMUnav_STM_F00.ioc	
— STM32F446VETX_FLASH.ld	linker skript pro FLASH
— STM32F446VETX_RAM.ld	linker skript pro RAM
— Hardware	
— IMUnav-libra	knihovna součástek a loga
— ...	
— IMUnav_H00	složka projektu DPS
— bom	
— ibom.html	seznam součástek
— production	
— ...	
— ESP32.kicad_sch	list schématu
— GPS.kicad_sch	list schématu
— IMU.kicad_sch	list schématu
— IMU.kicad_sch-bak	
— IMUnav_H00.kicad_dru	
— IMUnav_H00.kicad_pcb	deska
— IMUnav_H00.kicad_prl	
— IMUnav_H00.kicad_pro	
— IMUnav_H00.kicad_sch	list schématu
— IMUnav_H00.kicad_sch-bak	

- IMUnav_H00.step.....3D model desky
- Memory.kicad_sch..... list schématu
- USB+Power.kicad_sch.....list schématu
- User.kicad_sch.....list schématu
- fp-info-cache
- fp-lib-table
- sym-lib-table

Software

- Matlab.....složka vytvořených MATLAB skriptů a vzorových dat
 - C generator složka kalibrační procedury
 - ...
 - IMUonlyNavigation.m.....skript pro výpočet trajektorie pomocí pohybových rovnic
 - chuzePoByte.csv.....vzorová data
 - chuzeVenkuCALB.csv vzorová data
 - ctverecChuze.csv.....vzorová data
 - ctverecChuze2.csv.....vzorová data
 - ctverecChuze3CALB.csv vzorová data
 - ctverecRychlaChuze.csv.....vzorová data
 - imuAndGnssInsfilterMARG.m..skript pro fúzi dat pomocí Navigation Toolboxu
 - koleckoKolemFEKTUCALB.csv vzorová data
 - tiltedStationary.csv.....vzorová data
- Python
 - IMUDATA.BIN.....vzorová binární data
 - binDecoder.py skript pro převod binárních dat na CSV