

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

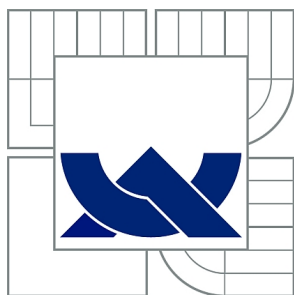
ODOMETRICKÝ MODUL PRO MOBILNÍ ROBOT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

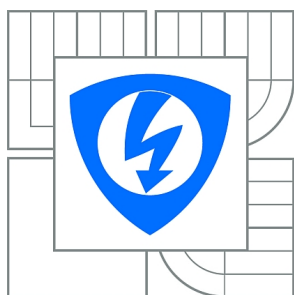
AUTOR PRÁCE
AUTHOR

DANIEL DAVÍDEK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ODOMETRICKÝ MODUL PRO MOBILNÍ ROBOT

ODOMETER MODULE FOR MOBILE ROBOT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

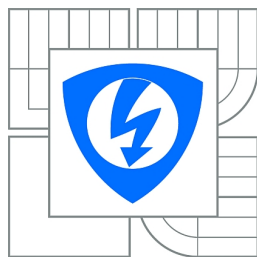
AUTOR PRÁCE
AUTHOR

DANIEL DAVÍDEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ JÍLEK

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student: Daniel Davídek

ID: 136510

Ročník: 3

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Odometrický modul pro mobilní robot

POKYNY PRO VYPRACOVÁNÍ:

Na základě návrhu provedeného v semestrální práci realizujte odometrický modul pro mobilní robot envMap. Osadte a oživte navržené DPS. Vytvořte software pro použitý mikrokontrolér. Zajistěte časovou synchronizaci s daty poskytovanými jinými sebelokalizačními jednotkami. Navrhněte a implementujte odhad chyby lokalizačních souřadnic v reálném čase. Vytvořte vizualizační aplikaci pro PC. Vyhodnoťte parametry realizovaného odometrického modulu. Srovnajte dosažené výsledky s vybranou jinou sebelokalizační metodou.

DOPORUČENÁ LITERATURA:

HOLLAND, John. Designing Autonomous Mobile Robots. : Newnes, c2004. 335 s. ISBN 0-7506-7683-3.

Termín zadání: 11.2.2013

Termín odevzdání: 27.5.2013

Vedoucí práce: Ing. Tomáš Jílek

Konzultanti bakalářské práce:

doc. Ing. Václav Jirsík, CSc.
Předseda oborové rady

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

Hlavním cílem této práce byl návrh a zprovoznění odometrického modulu pro mobilní robot envMap. Modul zpracovává signály z enkodérů a následně vypočítává polohu a natočení robotu spolu s pravděpodobností jeho výskytu. Vypočtená data jsou spolu s časovou značkou získanou z GPS odesílána do nad-systému pomocí sériového rozhraní. K realizaci byla zvolena vývojová deska STM32F407G. Časovače desky jsou nastavitelné do funkce enkodéru, což minimalizuje možnost ztráty informace. Byly navrženy dvě desky plošných spojů: redukce k GPS pro přivedení potřebného signálu 1PPS a nasazovací deska pro STM32F4.

V této práci jsou uvedeny rovnice pro výpočty odometrických informací i pravděpodobnosti výskytu. Pro výpočet pohybu robotu je zvolena rovnice uvažující pouze pohyb po rovinné ploše. Dále je popsána struktura programu pro STM32 a jeho funkčnost. Ve výsledcích je uvedeno kompletní schéma zapojení spolu s popisem obslužného programu a jednotlivých komunikačních zpráv. Závěrem práce jsou popsána a vyhodnocena měření zjišťující parametry robotu a finální odometrické měření porovnané s GPS lokalizací jako referencí.

KLÍČOVÁ SLOVA

odometrie; lokalizace; inkrementální enkodéry; diferenciální podvozek; mobilní robotika; STM32F407G; kvadraturní signál; sériová komunikace; kovarianční matice pravděpodobnosti

ABSTRACT

The main objective of this work was the development and realization of odometric module for mobile robot envMap. The module processes signals from the incremental encoder and then calculates the position and direction of the robot together with the probability of its occurrence. The calculated data with a time stamp obtained from the GPS, are then sent to the above-system via the serial interface.

For the realization the development board STM32F407G was chosen. f this board can be configured to function as an encoder input. This minimizes the possibility of encoder data loss when external interrupts are handled. The STM32F4 will not be any pull-back in the later development as there are plenty of peripherals on STM32F4. They were designed with two printed circuit boards: the reduction for the GPS module and shield-board for STM32F4.

The equations for calculating odometric information and the probability of occurrence are included in this work. To calculate the movement of the robot, the equation considering only the movement on the plane surface is chosen. Later in this thesis there is also description of the source code for the STM32 and its functionality. In the results there is a complete connecting diagram along with the description of the utility program ODOMAT and communication messages. Finally, measurements correcting systematic error in parameters of the robot are described and there is the final odometry measurements compared with GPS localization as ground truth.

KEYWORDS

odometry; localization; incremental encoders; differential drive; mobile robotics; STM32F407G; quadrature signal; serial communication; covariance matrix

DAVÍDEK, Daniel *Odometrický modul pro mobilní robot*: bakalářská práce. Vysoké Učení Technické, Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace, 2013. 78 s. Vedoucí práce byl Ing. Tomáš Jílek,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Odometrický modul pro mobilní robot“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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.

Vysoké Učení Technické, Brno
(podpis autora)

PODĚKOVÁNÍ

Tímto bych rád poděkoval vedoucímu této bakalářské práce panu Ing. Tomáši Jílkovi za odborné vedení, cenné připomínky, konzultace, trpělivost a podnětné návrhy k práci. Dále bych chtěl poděkovat Ing. Miloslavu Richterovi, Ph.D za podnětné rady.

Vysoké Učení Technické, Brno

(podpis autora)

OBSAH

Úvod	13
1 Řešení studentské práce	14
1.1 Teoretický rozbor	14
1.1.1 Odometrie & Dead reckoning	14
1.1.2 Reálný model	19
1.1.3 Vztahy použité pro výpočet v odometrickém modulu	23
1.2 Technická část	24
1.2.1 Vybavení robotu envMap	24
1.2.2 Odometrický modul	26
1.3 Softwarová sekce	28
1.3.1 Vývojové prostředí	28
1.3.2 Členění zdrojových kódů	29
1.3.3 Nastavení periferií	29
1.3.4 Datové struktury	31
1.3.5 Odometrické výpočty	31
1.3.6 Typický programový cyklus Odometrického modulu	31
2 Výsledky studentské práce	34
2.1 Hardware	34
2.1.1 Kompletní schéma zapojení	34
2.1.2 Desky plošných spojů	34
2.2 Software	34
2.2.1 ODOMAT - obslužný a vizualizační program	34
2.2.2 Protokol a zasílané zprávy	35
2.3 Testování	37
2.3.1 Měření převodového poměru	37
2.3.2 Měření vůle v převodech	38
2.3.3 Měření scale-faktoru	38
2.3.4 UMBmark měření	38
2.3.5 Odometrické měření s GPS referencí	40
3 Závěr	44
Literatura	45
Seznam symbolů, veličin a zkratk	49

Seznam příloh	50
A Obsah přiloženého CD	51
A.1 Software	51
B Přílohy Technické části	52
B.1 Enkodéry HEDS 5500	52
B.2 GPS Trimble BX982	53
B.3 Kabeláž	53
C Přílohy Softwarové části	55
C.1 Frekvence jednotlivých komponent	55
C.2 Vývojové prostředí	55
C.2.1 Změny v konfiguračních souborech	56
C.3 Nastavené periferie a piny STM32	57
C.4 Struktura ODOM_robot	57
C.5 Roznásobené maticové výpočty	58
C.6 Program ODOMAT	58
C.7 Komunikační zprávy	58
D Přílohy Výsledků práce	64
D.1 Statistické výpočty	64
D.2 Doplnující informace	64
D.2.1 Zamezení ztrát informací o poloze	64
D.3 Desky plošných spojů	66
D.3.1 Pinout	66
D.3.2 Celkové schéma zapojení	66
D.3.3 Navrhy Desek plošných spojů	66
D.4 Výsledky měření	66
D.4.1 UMBmark	66
D.4.2 GPS referenční měření	66

SEZNAM OBRÁZKŮ

1.1	Podstava a detail kola robotu envMap s přibližnými rozměry	15
1.2	Geometrické znázornění pohybu robotu při konstantních otáčkách kol.	16
1.3	Znázornění geometrických vztahů mezi jednotlivými veličinami	17
1.4	Geometrické znázornění aktualizace pozice a natočení	18
1.5	Vzorkovaná reprezentace důvěry v polohu nesenzorického robotu. Plná čára znázorňuje pravou polohu a vzorky míru důvěry v polohu v různém čase THRUN[23]	20
1.6	Příklad optického disku, kvadrurní dvoukanálový signál a měřitelné charakteristiky signálu z optických enkodérů	25
1.7	Zapojení enkodérů HEDS 5500 spolu s pull-up rezistory[16]	26
2.1	Vizualizace robotu v programu ODOMAT	35
2.2	Náhled na měření UMBmark	40
2.3	Náhled na průřez parkoviště	41
B.1	Ilustrační foto, pasivní-vlečné kolo typu caster[10]	52
B.2	Optický enkodér typu HEDS 5500 osazený na robotu envMap[16]	53
B.3	Spojení GPS a STM32F4 pomocí přímého kabelu	53
B.4	Připojení enkodérů do desky STM32, pomocí čtyřžilového kabelu	54
C.1	Rozhraní programu ODOMAT	59
C.2	Znázornění signálu ATT a 1PPS v časové souslednosti (překresleno) [17]	61
D.1	Vytvořené ošetření nasčítaných hodnot enkodérových timerů při příchozích přerušení	65
D.2	Celkové schéma zapojení	67
D.3	GPS redukce - Logické schéma	68
D.4	STM32F4 násadná deska - Logické schéma	69
D.5	GPS redukce - Návrh desky plošných spojů	70
D.6	STM32F4 násadná deska - Návrh desky plošných spojů	71
D.7	Výsledky simulace měření UMBmark do nichž nebyl započítán E_D	72
D.8	Výsledky simulace měření UMBmark do nichž byl započítán E_D	72
D.9	Geometrické znázornění výpočtu chyby transformace GPS souřadnic	73
D.10	Měření s parametry I na trase A	73
D.11	Měření s parametry II na trase A	74
D.12	Měření s parametry III na trase A	74
D.13	Měření s parametry IV na trase A	75
D.14	Odchylka GPS na trase A	75
D.15	Měření s parametry I na trase B	76
D.16	Měření s parametry II na trase B	76

D.17 Měření s parametry III na trase B	77
D.18 Měření s parametry IV na trase B	77
D.19 Odchylka GPS na trase B	78

SEZNAM TABULEK

1.1	Popis geometrického znázornění pohybu	16
1.2	Popis stavových rovnic systému a měření	21
1.3	Tabulka vybraných parametrů motorů BCI-63.55 [15]	24
1.4	Tabulka stavů výstupních kanálů kvadraturního enkodéru	25
1.5	Tabulka popisu výstupních pinů enkodéru HEDS 5500[16]	26
1.6	Tabulka vypočtených hodnot	28
1.7	Popisy obsahů jednotlivých souborů zdrojových kódů	29
1.8	Nastavení jednotlivých sériových komunikačních kanálů	29
1.9	Způsob vyvolání iterace při přerušení od TIM2 závisí na nastavení parametru <code>iter_type</code> , jenž může nabývat různých hodnot	32
2.1	Druhy zpráv	36
2.2	Popis zprávy ATT	36
2.3	Příklad zprávy ODOMSG	36
2.4	Tabulka vypočtených tiků na otočku kola	38
2.5	Tabulka vypočtených vůlí kol a převodovky [tiků]	38
2.6	Tabulka vypočtených korekčních parametrů UMBmark	40
2.7	Tabulka vypočtených sklonů plochy parkoviště a maximální chyby . .	42
2.8	Tabulka výsledků pro obě trasy	43
2.9	Tabulka korigovaných parametrů robotu	43
B.1	Tabulka vybraných parametrů enkodérů HEDS 5500 [16]	52
B.2	Hodnoty odečtené z osciloskopu na enkodérech při běžných rychlos- tech motoru	52
B.3	Tabulka vybraných parametrů GPS modulu Trimble BDX982 [16] . .	53
B.4	Potřebné datové kabely - propojení pin×pin	54
C.1	Popis jednotlivých prvků struktury <code>ODOM_robot</code>	60
C.2	Vypisování zprávy ODOMSG	62
C.3	Popis zprávy ODOMSG	63
D.1	Přiřazení propojení pinů jednotlivých komponent	66

ÚVOD

Cílem této bakalářské práce bylo navrhnout a realizovat odometrický modul pro mobilní robot envMap. Dále navrhnout a zprovoznit software pro MCU zpracovávající odometrická měření, vypočítávající polohu a pravděpodobnost výskytu v reálném čase a posílat tyto data do počítače kde jsou vizualizována. envMap je dálkově ovládaný mobilní robot, jehož účel je přesně lokalizovat svou polohu v prostoru. Dále má sloužit jako přesný referenční bod pro navigaci a lokalizaci dalších robotů.

Jeho podvozek je složen ze dvou souosých náhonových kol a dvou vlečných kol typu caster¹. Jejich rozmístění je vidět z nákresu 1.1.1.

Jedná se tedy o diferenciální podvozek. Elektrické motory pohánějící skrze převodovku kola jsou opatřeny relativními optickými enkodéry. Tento typ podvozku umožňuje jednoduchý přepočet dráhy ujeté jednotlivými koly na polohu robotu v prostoru.

Pro lokalizaci v prostoru robot dosud využíval jen GPS modul a data z laserového 3D skeneru. Tato práce se zabývá pouze odometrií, ostatní lokalizační prostředky byly použity pouze pro referenční měření.

Pro zpracování odometrických dat je použit externí modul na platformě STM32F4 určený jen pro tuto funkci.

Signály z enkodérů jsou přivedeny do modulu a z nich následně vypočtena informace o pohybu robota v prostoru. Robot je reprezentován natočením (azimutem) a bodem na středu osy náhonových kol promítnutým do roviny země, které se kola dotýkají v „jednom bodě“.

Odometrické výpočty uvažují robot pohybující se na horizontální rovině ploše.

Vypočtené informace o pohybu jsou zasílány pomocí zprávy ODOMSG do vytvořeného obslužného programu ODOMAT. Tyto zprávy kromě lokalizačních dat obsahují také časovou značku synchronizovanou signálem 1PPS², získaným z instalovaného GPS modulu. Časová značka umožňuje při určování polohy porovnat data s ostatními dříve zmíněnými lokalizačními prostředky, jelikož tyto mají stejnou časovou základnu.

¹pasivní kolo upevněné na otočném kloubu, kde osa kloubu neprotíná osu otáčení casteru, což umožňuje pohyb všemi směry se zmenšenou možností prokluzu, ilustrační obrázek viz. B.1

²PPS: pulzů za sekundu – Pulse per second

1 ŘEŠENÍ STUDENTSKÉ PRÁCE

1.1 Teoretický rozbor

1.1.1 Odometrie & Dead reckoning

Již námořníci určovali svou polohu na základě známého kurzu a rychlosti. Tato technika určení nynější pozice na základě matematického výpočtu z předchozí pozice je v anglické literatuře nazývána dead reckoning.¹

Odometrií je někdy nazýváno nejjednodušší určování polohy, kdy se předpokládá že „poloha robotu může být určena ze zařízení nazývaného odometr, které je instalováno na robotu.“ Optické enkodéry připojené k náhonovým kolům nebo motorům jsou jedny z nejběžněji užívaných nástrojů odometrie.²

Náš robot s diferenciálním podvozkem a optickými enkodéry na motorech umožňujícími vypočítat ujetou dráhu obou kol, je matematicky jednoduše reprezentovatelný systémem.

Geometrická reprezentace

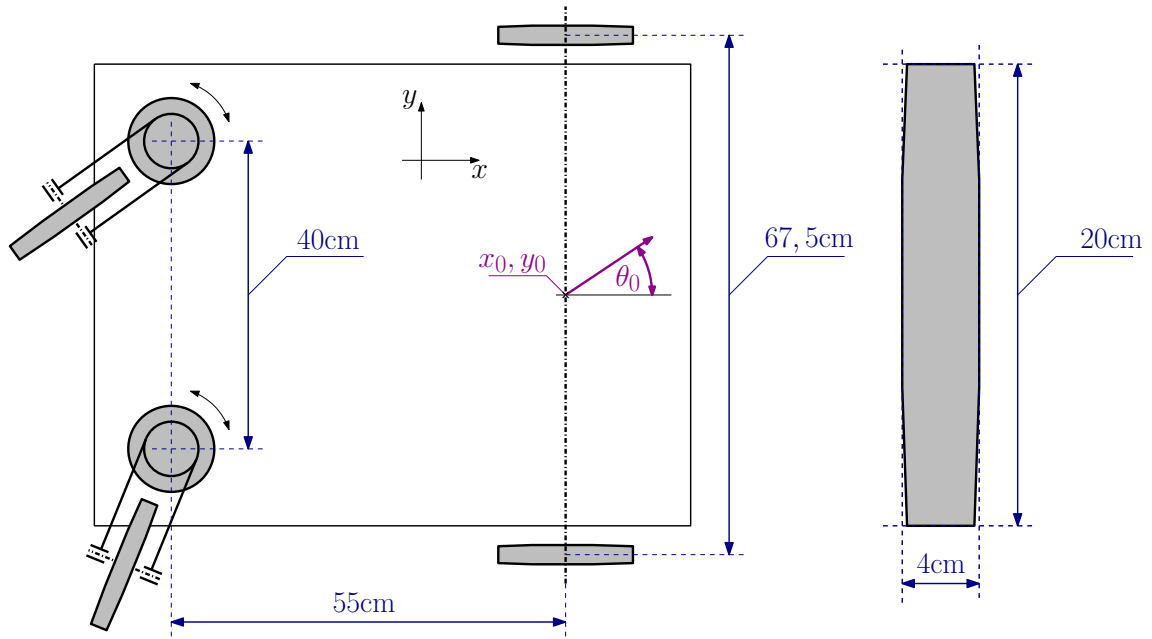
Jak již bylo řečeno robot je reprezentován bodem na středu osy náhonových kol promítnutým do roviny obsahující bodové doteky kol a svým natočením v „globálním“ souřadnicovém systému^{1.2}. Vztahy 1.1, 1.2 a 1.3 popisují ideální případ, kdy kolo má s podložkou dotyk v jednom bodě, nedochází k prokluzům a zrychlení kol je nulové. Podložka je rovná a je umístěna horizontálně.

V našem modelu opisují při pohybu robotu kola křivku, která pro případ zanedbatelně malého nebo nulového zrychlení obou kol odpovídá kružnicím s totožným středem. Pro malá zrychlení jsou křivky kružnicím v přiblížení podobné. V této práci se úhlové zrychlení kol, vzhledem k vysoké frekvenci vzorkování, neuvažuje. V dalším textu a nákresech vystupují proměnné s dolním indexem (0) jako vzorky v kroku (k) a proměnné s indexem (1) jako vzorky v kroku (k+1).

- Pro stejné rychlosti kol je poloměr kružnice nekonečný a tudíž robot jede po přímce.
- Pro různé rychlosti pravého a levého kola opisují trajektorie kružnice se stejným středem a s poloměrem nepřímo úměrným rychlosti otáčení kol. Rychlejší kolo opisuje větší kružnici, protože ujede delší dráhu.

¹ DUNLAP, G.D. a SHUFELDT, H.H., *Dutton's Navigation and Piloting*, s. 557-579. [8]

² BORENSTEIN, J. *"Where am I?"*, s. 13. (překlad autora) [7]



Obr. 1.1: Podstava a detail kola robotu envMap s přibližnými rozměry

$$p_C = \frac{p_R + p_L}{2} \quad (1.1)$$

$$\Delta\theta = \frac{p_R - p_L}{b} \quad (1.2)$$

$$R = \frac{p_L b}{(p_R - p_L)} \quad (1.3)$$

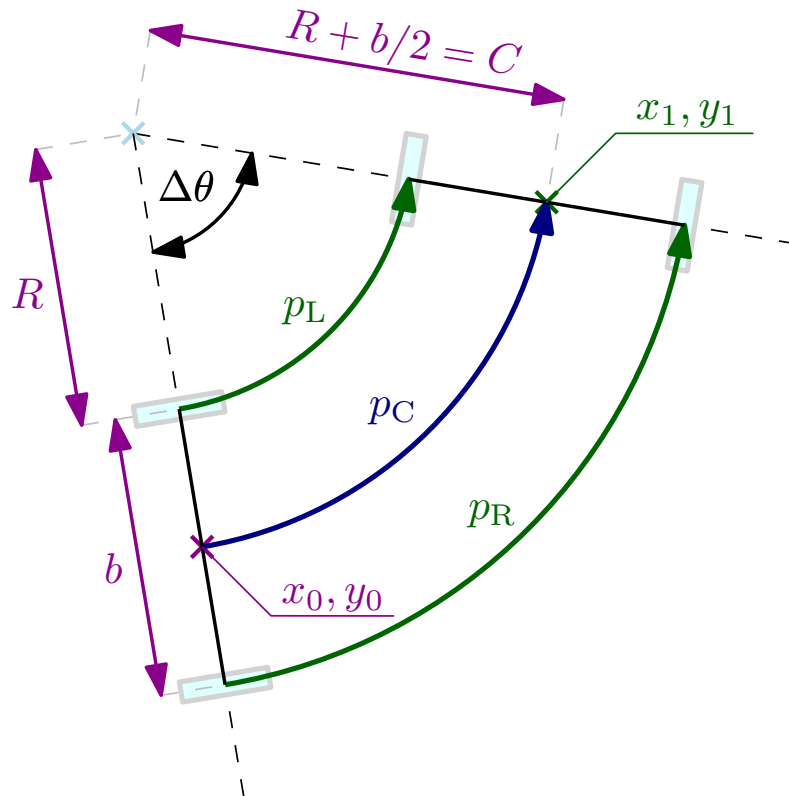
$$C = R + \frac{b}{2} = \frac{p_R + p_L}{2} \frac{b}{p_R - p_L} = \frac{p_C}{\Delta\theta} \quad (1.4)$$

Dráhy ujeté jednotlivými koly

Dráha, kterou kolo ujelo se vypočítává z enkodérů jako poměr příchozích náběžných hran neboli tiků-pulzů ku poměru tiků na jednu otáčku motoru, přepočtené přes poloměr kola a převodový poměr, protože mezi motorem a kolem je převodovka. Pro zjištění převodového poměru bylo provedeno měření.

$$p_{L/R} = a_{L/R} t_{L/R} \quad (1.5)$$

$$a_{L/R} = \frac{2\pi r_{L/R}}{T_{L/R}} \quad (1.6)$$



Obr. 1.2: Geometrické znázornění pohybu robotu při konstantních otáčkách kol.

Tab. 1.1: Popis geometrického znázornění pohybu

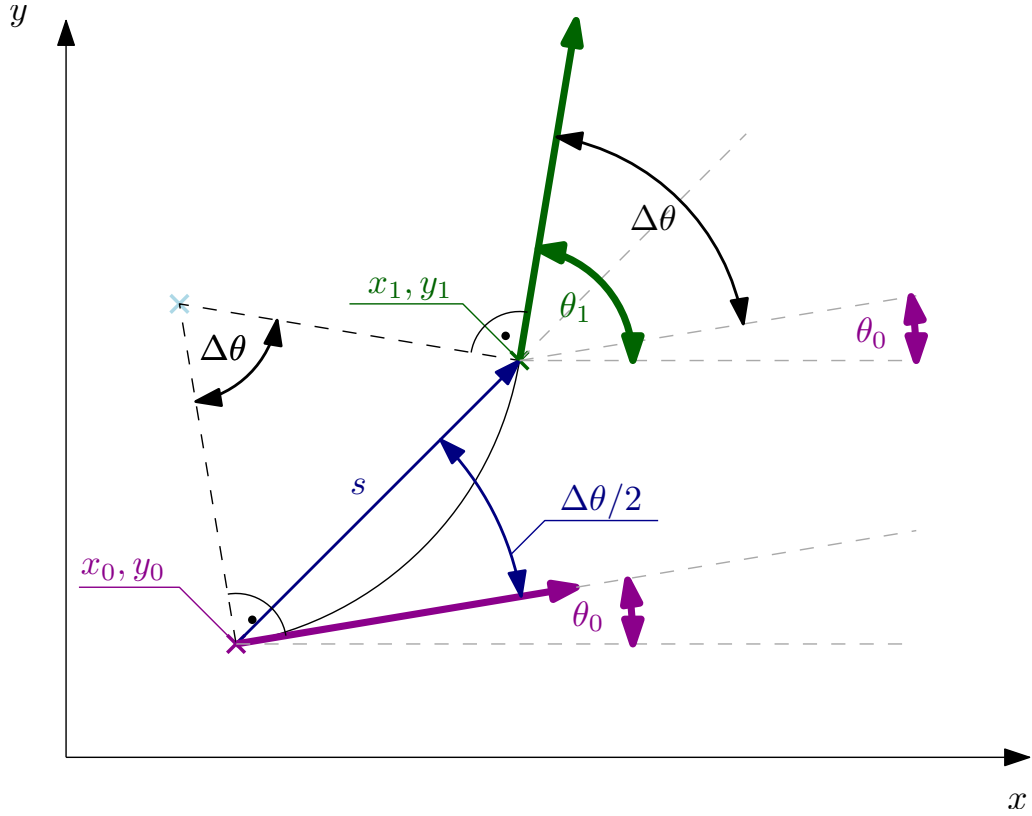
$p_{L/R}$...	kruhovité dráhy ujeté jednotlivými koly
$t_{L/R}$...	převodní poměr mezi motorem a kolem viz 2.3.1
$r_{L/R}$...	poloměr náhonového kola
$T_{L/R}$...	počet tiků enkodéru na jednu otočku kola

Změna orientace robotu

Orientace natočení robotu je vždy tečnou ke kružnici, kterou robot opisuje. Změna úhlu natočení, jak je vidět na nákresu 1.3, odpovídá úhlu ($\Delta\theta$), znázorněnému již na nákresu výše 1.2.

Posunutí v kartézských souřadnicích

Pro výpočet posunu v ose x a y je potřeba znát délku úsečky spojující body $[x_0, y_0]$ a $[x_1, y_1]$. V našem případě odpovídá daná úsečka tětivě kružnice jak je vidět z nákresu 1.4. Délka tětivy je po zjednodušení vidět v rovnici 1.8.



Obr. 1.3: Znázornění geometrických vztahů mezi jednotlivými veličinami

$$s = 2C \sin(\Delta\theta/2) \quad (1.7)$$

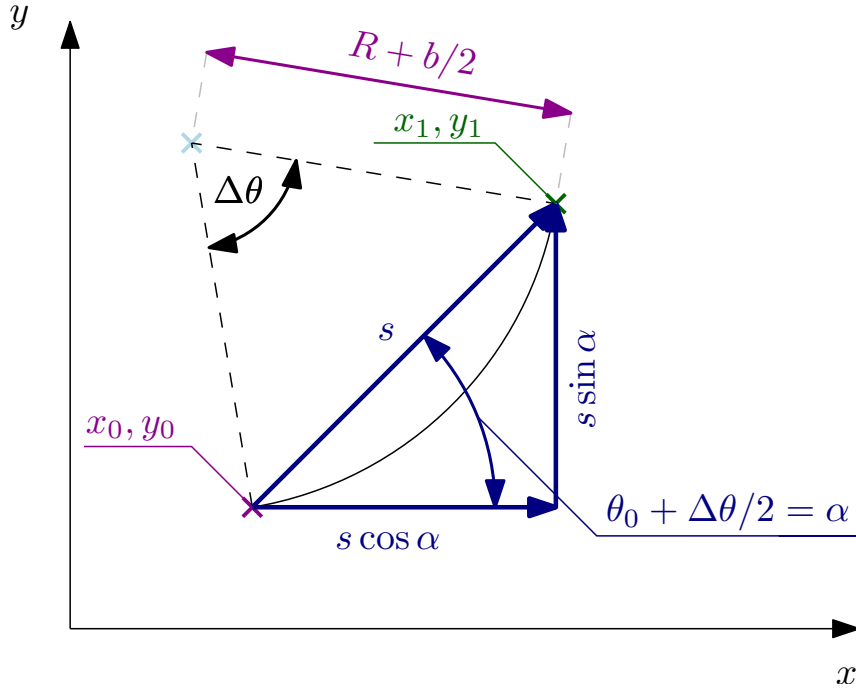
$$s = \frac{p_C}{\Delta\theta/2} \sin(\Delta\theta/2) \quad (1.8)$$

Aktualizace polohy a natočení

Nyní již máme posunutí a změnu natočení naší bodové reprezentace robotu, zbývá jen aktualizovat polohu v kartézských souřadnicích (viz Obráz 1.4/str.18). K tomu využijeme jednoduché geometrické rovnice:

$$\begin{cases} x_1 &= x_0 + s \cos(\theta_0 + \Delta\theta/2) \\ y_1 &= y_0 - s \sin(\theta_0 + \Delta\theta/2) \\ \theta_1 &= \Delta\theta + \theta_0 \end{cases} \quad (1.9)$$

Po dosazení délky tětivy.



Obr. 1.4: Geometrické znázornění aktualizace pozice a natočení

$$\begin{cases} x_1 &= x_0 + \frac{p_C}{\Delta\theta/2} \sin(\Delta\theta/2) \cos(\theta_0 + \Delta\theta/2) \\ y_1 &= y_0 - \frac{p_C}{\Delta\theta/2} \sin(\Delta\theta/2) \sin(\theta_0 + \Delta\theta/2) \\ \theta_1 &= \Delta\theta + \theta_0 \end{cases} \quad (1.10)$$

Součin goniometrických funkcí byl převeden na jejich součet, což je z hlediska náročnosti výpočtu optimálnější. Avšak relativní chyba výpočtu se teoreticky zvýší, protože se sčítají respektive odčítají dvě podobně velká čísla (pro malé $\Delta\theta$) viz [4]. V modulu je proto možné nastavit obě varianty výpočtu.

$$\begin{cases} x_1 &= x_0 + \frac{p_C}{\Delta\theta} \left[\sin(\theta_0 + \Delta\theta) - \sin(\theta_0) \right] \\ y_1 &= y_0 - \frac{p_C}{\Delta\theta} \left[\cos(\theta_0 + \Delta\theta) - \cos(\theta_0) \right] \\ \theta_1 &= \Delta\theta + \theta_0 \end{cases} \quad (1.11)$$

Přímočarý pohyb

Lze vidět že rovnice nedávají smysl pro nulovou změnu úhlu $\Delta\theta$. V případě tohoto stavu se použijí limity těchto rovnic pro $(\Delta\theta \rightarrow 0)$:

$$\begin{cases} x_1 &= x_0 + p_C \cos(\theta_0) \\ y_1 &= y_0 + p_C \sin(\theta_0) \\ \theta_1 &= \theta_0 \end{cases} \quad (1.12)$$

1.1.2 Reálný model

Zatím jsme uvažovali pouze ideální případ, avšak pokud zohledníme i reálné aspekty fyzikální podstaty odometrie zjistíme, že do výpočtů budou zaneseny neurčitosti a chyby. Například kola nemusí být díky nerovnému povrchu ve stálém kontaktu se zemí nebo může dojít nedostatečnou přiléhavostí či smyku k prokluzu. Enkodér prokluzujícího kola zaznamenaná otočky ačkoliv nedochází k opravdovému posunu po podložce, což způsobí chybu ve výpočtu polohy.

Zdroje chyb můžeme rozdělit do dvou hlavních skupin³:

- **Chyby systematické:**

- Nestejné průměry levého a pravého kola
- Nedokonalá kruhovitost kola - nelineární odvalování
- Nepřesná velikost rozestupu kol
- Vyosení kol – osa otáčení není totožná se spojnicí kol
- Konečné rozlišení enkodéru
- Konečná rychlost vzorkování enkodéru

- **Chyby nesystematické:**

- Jízda po nerovném povrchu
- Nepředpokládané překážky
- Prokluz způsobený
 - * nedostatečnou adhezivitou styčných ploch
 - * přílišné zrychlení
 - * smyk při rychlém zatáčení
 - * vnější síly (vítr, překážky)
 - * vnitřní síly (síly způsobené vlečnými koly)
 - * kontakt kola s povrchem není v jednom bodě

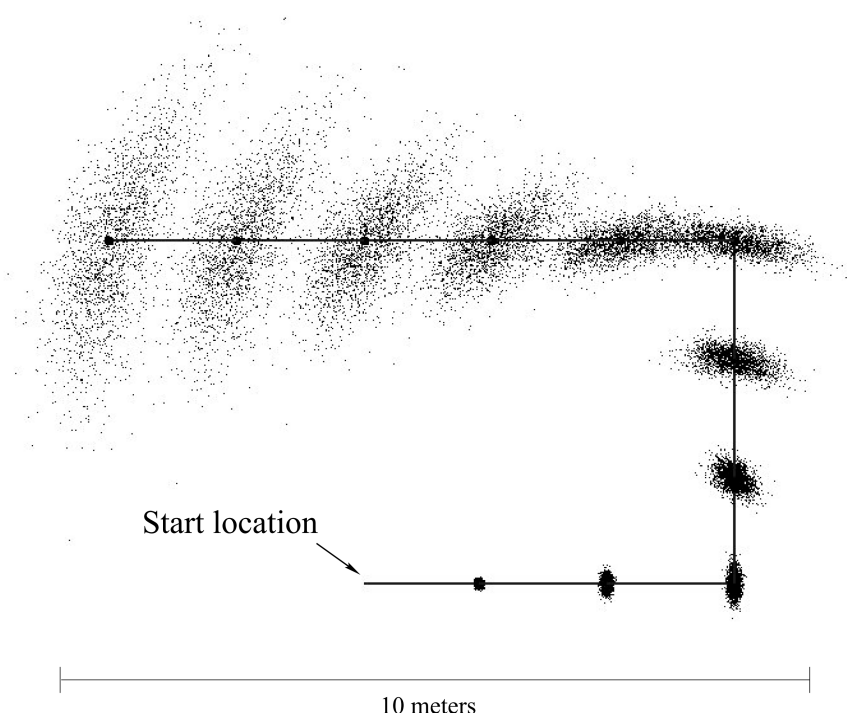
Výpočet pravděpodobnosti

Při výpočtu neurčitostí byl uvažován tzv. Markovský předpokladem jenž uvádí, že následující stav systému je nezávislý na předchozích stavech systému a je závislý pouze na stavu aktuálním.

³ BORENSTEIN, J. "Where am I?", s. 130-131. [7]

Existuje mnoho modelů pro výpočet lokalizační pravděpodobnosti, jejich rozsáhlý přehled je možné nalézt například v KRAMER [21]: „Způsob jak získat co nejlepší data z měření náchylných k chybě, je rozebírán už dlouho. Gaussova metoda nejmenších čtverců z roku 1795 je ta, ze které vychází všechny typy gaussovských pravděpodobnostních modelů. Bayesův filtr je nejobecnějším způsobem rekurzivního výpočtu míry důvěry v daný vzorek, avšak není možné jej v počítači implementovat a proto se musíme poohlížet po jiných filtrech které jsou realizovatelné“ [volný překlad autora].

Mezi hojně využívané způsoby zpracování naměřených hodnot a výpočtu pravděpodobnosti patří Kalmanův filtr a jeho varianty. Dále existují adaptivní filtry, které se dokáží přizpůsobit i změnám systému a náhlému zvýšení šumu. Tyto se dělí na filtry s tzv. děděnou adaptací a filtry s adaptací řízenou. Filtry se zděděnou adaptací většinou aplikují nějaký druh evolučního mechanismu, jenž je nastaven tak že vybírá nejvhodnější parametry systému. Řízené adaptační funkce jsou řízeny zvenci pomocí zpětnovazební smyčky. Tyto filtry většinou uvažují reprezentaci lokalizační pravděpodobnosti pomocí více částic, jejichž počet může v čase růst či klesat dle vývojové funkce a které mohou mít různou „cenu“.



Obr. 1.5: Vzorkovaná reprezentace důvěry v polohu nesenzorického robotu. Plná čára znázorňuje pravou polohu a vzorky míru důvěry v polohu v různém čase THRUN[23]

Vzhledem ke zvolenému MCU a zadání z něhož vyplývá, že lokalizační informace spolu s údajem o nepřesnosti má být zpracovávána a přeposílána do nad-systému

v reálném čase, zavrhl jsem pro přílišnou časovou náročnost při posílání částicové filtry, a pro reprezentaci robotu uvažuji predikční část filtru z rodiny Kalmanových filtrů, viz dále.

Nejefektivnějším lineárním filtrem pro lineární systémy s bílým gaussovým šumem (průměrná hodnota v nule, nekorelovaný) je filtr Kalmanův. Jak je dokázáno v SIMON [22] je Kalman ideálním lineárním filtrem i pro nelineární systémy.

Diskrétní Kalmanův filtr DKF, předpokládá lineární systém popsany rovnicemi 1.13.

$$\begin{cases} \mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_k \\ \mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \end{cases} \quad (1.13)$$

Tab. 1.2: Popis stavových rovnic systému a měření

$\mathbf{x}$...	stavový vektor
$\mathbf{u}$...	vektor vstupních veličin
$\mathbf{z}$...	vektor měřených veličin
$\mathbf{A}$...	systémová matice vnitřních vazeb
$\mathbf{B}$...	matice vazby systému na vstup
$\mathbf{C}$...	matice pozorování
$\mathbf{w}$...	reprezentuje vektor vstupního šumu
$\mathbf{v}$...	reprezentuje vektor šumu měření

Filtr pracuje ve dvou fázích nad pravděpodobnostně reprezentovaným systémem. Ve fázi predikce přepočte aktuální stav systému pomocí rovnic 1.14 a ve fázi aktualizace zahrne do systému nová měření (1.15) [25]

$$\begin{cases} \hat{\mathbf{x}}_k^- = \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{B}_k \mathbf{u}_{k-1} \\ \mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \end{cases} \quad (1.14)$$

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T (\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{V}_k)^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k^- \end{cases} \quad (1.15)$$

Horní index minus značí veličinu před započtením měření z kroku (k) = apriory předpoklad a horní plus nebo vynechaný index značí veličinu která bere v potaz i měření v aktuálním kroku = aposteriory. Stav systému $\mathbf{x}_k$ je zde reprezentován svým průměrem $\hat{\mathbf{x}}_k$ a kovarianční maticí $\mathbf{P}_k$. $\mathbf{K}_k$ je matice zesílení Kalmanova filtru.

Šum je vyjádřen kovariančními maticemi s předpokládaným charakterem bílého nekorelovaného šumu (1.16) viz [24].

$$p(\mathbf{w}_k) = N(0, \mathbf{Q}) \quad (1.16)$$

$$p(\mathbf{v}_k) = N(0, \mathbf{R}) \quad (1.17)$$

Protože náš systém je nelineární, využijeme Extended Kalmanova Filtru - EKF, který modeluje nelineární systém parciálními derivacemi v okolí aktuálního kroku. Nelineární systém ve tvaru

$$\begin{cases} \mathbf{x}_k &= g(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_k \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k \end{cases} \quad (1.18)$$

Kde nelineární funkce g a h reprezentují model systému a měření. Upravené rovnice predikce a aktualizace pro EKF mají tvary (1.19), (1.20).

$$\begin{cases} \hat{\mathbf{x}}_k^- = g(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) \\ \mathbf{P}_k^- = \mathbf{G}_k \mathbf{P}_{k-1} \mathbf{G}_k^T + \mathbf{W}_k \end{cases} \quad (1.19)$$

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k)^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-)) \\ \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \end{cases} \quad (1.20)$$

Kde matice $\mathbf{G}$ je linearizací modelu pomocí parciálních derivací a stejně tak $\mathbf{H}$ je linearizací vektoru měření. Matice jsou definovány dle 1.21 a 1.22.

$$\mathbf{G}_k = \frac{\partial g(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})}{\partial \mathbf{x}_{k-1}} \quad (1.21)$$

$$\mathbf{H}_k = \frac{\partial h(\mathbf{x}_{k-1})}{\partial \mathbf{x}_k} \quad (1.22)$$

Zavedení vstupu enkodérů do modelu systému

Pokud zahrneme do vyjádření vstupů z enkodérů i jejich chybu (1.23), vložíme tuto chybu také do rovnice systému respektive rovnici predikce (1.19) a musíme rovnici systému a predikce upravit ([23], [25], [29]).

$$\begin{cases} \hat{t}_L = t_L \pm \varepsilon_{t_L} \\ \hat{t}_R = t_R \pm \varepsilon_{t_R} \end{cases} \quad (1.23)$$

Nová rovnice systému

$$\mathbf{x}_k = g(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \varepsilon) \quad (1.24)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (1.25)$$

Nová rovnice predikce

$$\begin{cases} \hat{\mathbf{x}}_k^- = p(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, 0) \\ \mathbf{P}_k^- = \mathbf{G}_k \mathbf{P}_{k-1} \mathbf{G}_k^T + \mathbf{D}_k \Gamma_k \mathbf{D}_k^T \end{cases} \quad (1.26)$$

Kde Γ je kovarianční maticí šumu senzorů a $\mathbf{D}_k$ je jakobián mapující odometrický šum do stavového prostoru.

$$\mathbf{D}_k = \frac{\partial p(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, 0)}{\partial \mathbf{u}_{k-1}} \quad (1.27)$$

V našem případě, kdy kromě odometrických měření modul žádná další měření nepřijímá, nám zůstává z Kalmanova filtru pouze fáze predikce. Pro aktualizaci pravděpodobnosti při nepříchových měřeních se za aposteriory pravděpodobnost bere pravděpodobnost apriory [24].

$$\mathbf{P} = \mathbf{P}_k^- \quad (1.28)$$

1.1.3 Vztahy použité pro výpočet v odometrickém modulu

Dosadíme-li za nelineární funkci systému p rovnice vypočtené výše 1.11 ve tvaru sloupcového vektoru a vypočteme parciální derivace jednotlivých matic, dostaneme následující vztahy.

$$\mathbf{G}_k = \begin{pmatrix} 1 & 0 & \frac{p_C}{\Delta\theta} (\cos(\Delta\theta + \theta_k) - \cos(\theta_k)) \\ 0 & 1 & \frac{p_C}{\Delta\theta} (\sin(\Delta\theta + \theta_k) - \sin(\theta_k)) \\ 0 & 0 & 1 \end{pmatrix} \quad (1.29)$$

$$\mathbf{D}_k = \begin{pmatrix} \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{D}_{12} & \mathbf{D}_{22} \end{pmatrix}, \text{ kde} \quad (1.30)$$

$$\begin{cases} \mathbf{D}_{11} = \frac{a_L}{b} (-C \cos(\Delta\theta + \theta_k) + \frac{p_R}{\Delta\theta^2} (\sin(\Delta\theta + \theta_k) - \sin(\theta_1))) \\ \mathbf{D}_{12} = \frac{a_R}{b} (C \cos(\Delta\theta + \theta_k) - \frac{p_L}{\Delta\theta^2} (\sin(\Delta\theta + \theta_k) - \sin(\theta_1))) \\ \mathbf{D}_{12} = \frac{a_R}{b} (-C \sin(\Delta\theta + \theta_k) - \frac{p_R}{\Delta\theta^2} (\cos(\Delta\theta + \theta_k) - \cos(\theta_1))) \\ \mathbf{D}_{22} = \frac{a_R}{b} (C \sin(\Delta\theta + \theta_k) + \frac{p_L}{\Delta\theta^2} (\cos(\Delta\theta + \theta_k) - \cos(\theta_1))) \\ C = \frac{p_C}{\Delta\theta} \end{cases} \quad (1.31)$$

Kovarianční matice šumu senzorů - enkodérů Γ_k přepočítává chybu tiků na 1 otočku kola $\sigma_{T_{L/R}}$ v závislosti na jejich počtu $t_{L/R}$, ku celkovému počtu na jednu otočku $T_{L/R}$.

$$\Gamma_k = \begin{pmatrix} \sigma_{tL}^2 & 0 \\ 0 & \sigma_{tR}^2 \end{pmatrix}, \text{ kde} \quad (1.32)$$

$$\sigma_{tL} = \sigma_{T_L} \frac{t_L}{T_L} \quad (1.33)$$

$$\sigma_{tR} = \sigma_{T_R} \frac{t_R}{T_R} \quad (1.34)$$

1.2 Technická část

1.2.1 Vybavení robotu envMap

Podvozek a kola

Jak již bylo zmíněno, robot envMap má diferenciální podvozek se dvěma aktivními a s dvěma podpěrnými vlečnými koly. Rozměry podstavky jsou vidět na nákresu. (viz Obraz 1.1.1/str.15) Kola jsou vyrobeny z tuhé pryže a tím je zaručen dobrý kontakt a vysoká adhezivita s povrchem.

Převodovka

Mezi motorem a kolem je připojena vícestupňová převodovka do pomala. Pro zjištění převodového poměru bylo provedeno experimentální měření viz. 2.3.1.

Motory

Aktuátory robotu jsou stejnosměrné komutátorové motory typu BCI-63.55 od firmy ebmpapst.

Tab. 1.3: Tabulka vybraných parametrů motorů BCI-63.55 [15]

Nominální napájecí napětí	24 V
Nominální rychlost	3300 ot/min ⁻¹
Nominální moment	270 mNm
Proudový odběr	4,9 A
Výstupní výkon	93 W

Enkodéry

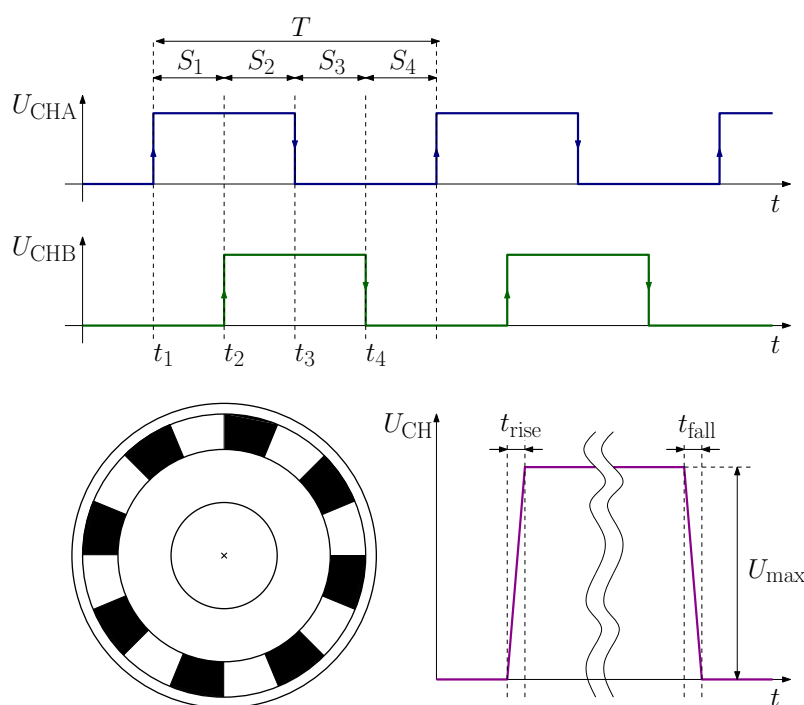
Na levém i pravém motoru je nainstalován optický inkrementální enkodér firmy Avago TECHNOLOGIES.

Oba enkodéry jsou typu ENC HEDS 5500, jejich výstupem je dvoukanálový kvadrurní signál⁴ s 512ti hranami na otáčku.

Každý enkodér obsahuje jednu LED jako zdroj světla, optickou soustavu, kódovací stínítko a integrovaný obvod s fotodetektory a zpracovávajícími jednotkami. Při otáčení motoru stínící optický disk střídavě propouští světlo od diody na fotodetektory, jenž jsou rozmístěny tak, že výsledný signál je kvadrurní. Celkově mohou nastat 4 stavy viz tabulka 1.4:

Tab. 1.4: Tabulka stavů výstupních kanálů kvadrurního enkodéru

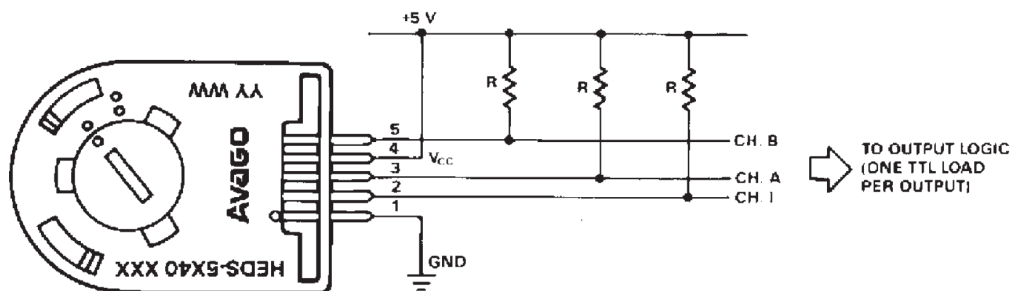
stav	S1	S2	S3	S4
U_{CHA}	H	H	L	L
U_{CHB}	L	H	H	L



Obr. 1.6: Příklad optického disku, kvadrurní dvoukanálový signál a měřitelné charakteristiky signálu z optických enkodérů

⁴Dva signály s 90-ti stupňovým fázovým posunem

Enkodér má pět výstupních pinů, které jsou zapojeny dle obrázku 1.2.1, kdy HEDS 5500 nemají zapojený pin kanálu I.



Obr. 1.7: Zapojení enkodérů HEDS 5500 spolu s pull-up rezistory[16]

Pull-up rezistory 1.2.1 nejsou pro HEDS 5500 nutností, ale jsou doporučovány pro vylepšení náběžných hran a pro vyšší signálové frekvence. [16]

Tab. 1.5: Tabulka popisu výstupních pinů enkodéru HEDS 5500[16]

PIN1	GND	Signálová a napájecí zem
PIN2	NC	Nevyužitý, enkodéry typu HEDS 5500, nemají zapojený kanál I
PIN3	CHA	Výstup prvního kanálu enkodéru
PIN4	VCC	Napájecí napětí enkodéru
PIN5	CHB	Výstup druhého kanálu enkodéru

GPS Modul

K časovému determinizmu odometrických informací bude využíván signál 1PPS, jenž je zarovnán k časové značce (viz obrázek C.7) z instalovaného GPS modulu BD982 firmy Trimble. Na robotu jsou instalovány dvě antény, což umožňuje přesnější zaměření. Při použití externí statické referenční antény je možné přijímat z GPS zprávy obsahující polohu robotu v souřadnicích [East North Up], spolu s azimutem a náklonem. Podrobnější informace v sekci 2.2.2.

1.2.2 Odometrický modul

K realizaci procesní jednotky byla po konzultaci s vedoucím práce vybrána vývojová deska STM32F4DISCOVERY⁵, jenž díky své bohaté výbavě umožňuje pozdější rozšíření funkčnosti. Vybrané parametry desky [11] [12]

⁵konkrétně STM32F407VGT6

- 100 pinové provedení, (2x dvouřadé 25 pinové)
- ST-LINK/V2 debugovací rozhraní zahrnuto přímo v desce
- SMD LED, tlačítka
- USB OTG full-speed rozhraní, micro-AB konektor

V desce je osazen výkonný mikrokontroler z řady Cortex™-M4.

Vybrané parametry procesoru[13]:

- maximální frekvenci jádra 168MHz
- FPU⁶, CMSIS⁷
- 1-Mbyte Flash, 192-Kbyte SRAM
- GPIO piny mají toleranci vstupního napětí 5V (viz referenční manuál) [13]

1PPS

Linka 1PPS je chráněna proti ESD. Výška pulzu je 3.3V a 8μsekund [17]. STM32 signál snadno detekuje, protože frekvence vzorkování splňuje Shannon-Kotelnikovu podmínku pro zachycení signálu^{1.37}.

$$f_{PPS} < 2f_{TIMX} \quad (1.35)$$

$$\frac{1}{8\mu s} < 2 \cdot 84MHz \quad (1.36)$$

$$\frac{1}{8}10^6Hz \ll 168 \cdot 10^6Hz \quad (1.37)$$

Frekvence detekce odpovídá frekvenci enkodérových časovačů TIM1 a TIM8. Více v C.1.

Napájení

Napětí externí baterie, která bude použita pro napájení modulu je 12V. Po připojení do DC-Jacku je pro napájení desky STM32F4 a enkodérů je konvergováno na 5VDC pomocí stabilizátoru L78M05. Maximální proudy jednotlivých komponent jsou zobrazeny v tabulce 1.6.

Celkový odběr jednotlivých komponent nepřekročí maximální výstupní proud ze stabilizátoru (500mA). Do zdířky napájení může být přiveden zdroj stejnosměrného napětí s výstupním napětím v rozmezí (7 ÷ 35)V jenž je schopný dodat proud alespoň 300mA. Tyto hodnoty vycházejí z parametrů stabilizátoru a odběrů jednotlivých komponent spolu s proudovou rezervou. Odometrický modul je možné napájet také skrze USB konektor na desce STM32-Discovery. Tímto jsou napájeny

⁶FPU: jednotka zpracovávající operace nad čísly s desetinou částí – Floating Point Unit

⁷CMSIS: Standart pro softwareové rozhraní mikrokontrolerů typu Cortex – Cortex™ microcontroller software interface standard

Tab. 1.6: Tabulka vypočtených hodnot

Komponenta	Maximální proud [mA]
Enkodéry HEDS 5500	2×40
Procesor STM32F4	117
4×LED	4×20
MAX232	10
I_{celk}	287

i enkodéry, jelikož jsou připojeny na pin 5V a jejich proudový odběr je nižších než dovolených 100mA na jeden Port STM32-Discovery. [16]

$$P_{celk} = UI = 5V \cdot 287mA = 1.435W \quad (1.38)$$

$$P_{dim} = 1.5W \quad (1.39)$$

Převodník musí být dimenzován na celkový příkon alespoň $P_{dim} = 1.5W$. Byl zvolen regulátor napětí L78M05 v pouzdře TO220 s maximálním dodaným proudem 500mA, celková spotřeba je tedy koncipována maximálně na $P_{prikon} = 2.5W$.

Vstupní napětí pinů

Všechny používané piny STM32, ať už komunikační nebo signálové, jsou FT (Fault Tolerant), což mimo jiné značí, že na ně může být přivedeno napětí 5V.

1.3 Softwarová sekce

1.3.1 Vývojové prostředí

Po konzultaci bylo k programování zvoleno freewarové vývojové prostředí CoIDE, jenž je součástí vývojových nástrojů CoCox, které nabízí podporu pro mikrokontroléry typu ARM Cortex. CoIDE se skládá z CoBuilder programovací platformy a dále z CoDebuggeru ladícího nástroje pro ARM Cortex Procesory.

CoBuilder podporuje GCC kompilátor, integruje CoDebugger a umožňuje interní sdílení a integraci zdrojů.

1.3.2 Členění zdrojových kódů

Soubory zdrojových kódů jsou členěny dle 1.7, kdy každý z c-souborů (kromě `main.c`) má svou hlavičku se stejným jménem. Tato je opatřena ochranou proti vícenásobnému vložení pomocí pre-kompilačních příkazů (např. `#ifdef`)

Tab. 1.7: Popisy obsahů jednotlivých souborů zdrojových kódů

<code>main.c</code>	obsahuje funkci <code>main()</code> a definici globálních pointerů na hlavní datové struktury
<code>INIT.h</code> <code>INIT.c</code>	inicializační funkce pro jednotlivé periferie
<code>ODOM.h</code> <code>ODOM.c</code>	definici hlavních datových struktur a funkcí pro odometrické výpočty a vytváření <code>ODOMSG</code>
<code>PERIPH.h</code> <code>PERIPH.c</code>	funkce obsluhy externích přerušení od periférií
<code>SIM.h</code> <code>SIM.c</code>	obsahuje funkce pro simulaci vstupů od enkodérů
<code>syscalls.c</code>	zde je definována funkce <code>_write(..)</code> jenž umožňuje zasílat řetězce tisknuté pomocí funkce <code>printf(..)</code> na definovanou USART linku
<code>stm32f4_conf.c</code>	odkomentované hlavičkové soubory používaných periférií

Ostatní soubory jsou generovány automaticky. Obsahují většinou nastavení registrů a konfiguraci mikrokontroléru.

1.3.3 Nastavení periférií

USART

Pro komunikaci s nadřazeným systémem (USART2) i pro příjem ATT zprávy od GPS (USART3) je sériové rozhraní nastaveno dle tabulky 1.8.

Tab. 1.8: Nastavení jednotlivých sériových komunikačních kanálů

	GPS - UTC message	ODOMSG
Baudrate	9600	19200
HW regulace toku dat	vypnuta	vypnuta
Délka slova	8 bitů	8 bitů
STOP-bit	1	1
Parita	vypnuta	vypnuta

EXTINT

Externí přerušení jsou nastaveny pro příjem signálu 1PPS od GPS a detekci uživatelského tlačítka - vyvolání přerušení na náběžnou hranu a jsou nastaveny funkce obsluhy těchto přerušení.

LED - GPIO

Pro ladící účely mohou být pomocí maker `DEBUG_` nastaveny 4 LED na reprezentaci příchozích tiků atd. Dále čtyři piny slouží pro simulaci signálů od kvadratických enkodérů a jeden k simulaci signálu 1PPS. Podrobnější informace v příloze C.3.

Frekvence oscilátorů mikrokontroléru

Pro co největší rychlost zpracování všech funkcí, byla zvolena frekvence (interního oscilátoru) procesoru 168MHz. Externí oscilátor má definovanou frekvenci 8MHz. Výpočty jednotlivých frekvencí periférií a timerů jsou v příloze C.1.

TIM1 & TIM8 - enkodéry

Pro sběr dat z enkodérových kvadrturních signálů jsou použity timery TIM1 a TIM8 jenž jsou nastaveny na STM32F4 podporovanou funkci Enkodéru. Vypočtená hodnota nasčítaných tiků je nepřetržitě ukládána do jejich shadow-registru. Toto umožňuje načítání otáček i při vyvolání přerušení a proto nedochází ke ztrátě tiků oproti původnímu nastavení, kdy byla detekce tiků nastavena na přerušení vstupními piny, kdy při vyvolání jiného přerušení došlo ke ztrátě informace.

TIM2 - iterační časovač

Pro spouštění iterací - tedy výpočtů mezi jednotlivými zprávami 1PPS, je používán timer TIM2 s nastavitelnou periodou. Při příchodu jeho přerušení na konci periody jsou vyčtena data z enkodérových timerů a je dle nastavení `iter_type` vyvolána iterace, která přepočte údaje o aktuální poloze robotu.

TIM3 - simulace signálu PPS

Pro simulaci příchozích zpráv 1PPS při nepřipojené GPS je možnost spustit timer TIM3 s nastavitelnou frekvencí, na něhož bude modul reagovat stejně jako při příchozí zprávě 1PPS.

1.3.4 Datové struktury

Všechny fyzické parametry robotu (rozměry kol, převodový stupeň ...) jsou spolu s naměřenými a vypočtenými odometrickými informacemi uloženy ve struktuře `ODOM_robot`. Zde se také nachází nastavení pro započítání nového iteračního kroku.

Naposledy odeslaná zpráva nad-systému spolu se zprávou ATT a zprávami PCMSG, jež jsou přijímány od nad-systému, se nacházejí ve struktuře `ODOM_communication`.

1.3.5 Odometrické výpočty

Při vyvolání funkce `ODOM_iteration` je přepočítán počet tiků uložených ve struktuře `ODOM_robot` na celkovou ujetou dráhu jednotlivých kol a dále na pohyb a natočení robotu dle funkcí popsaných v sekci Geometrická reprezentace 1.1.1. Maticový výpočet pravděpodobností byl roznásoben na jednotlivé členy tak, aby ho mohl zpracovat MCU bez použití maticových struktur viz příloha C.5.

1.3.6 Typický programový cyklus Odometrického modulu

Po zkompileování a nahrání nejaktuálnějšího programu se mikrokontrolér naboootuje a vnitřní registry se nastaví do resetových hodnot. Poté se z funkce `Default_Reset_Handler()` zavolá funkce `SystemInit()`, která přenastaví frekvenci procesoru a poté zavolá hlavní funkci `main()`.

Inicializace

Ve funkci `main` se nejdříve zavolají inicializační funkce jež nastaví periferie mikrokontroléru. Vytvoří se instance hlavních programových struktur `rob` a `msg`, ve kterých budou uloženy odometrické a komunikační data. Dva globální ukazatele (`currob`, `currmsg`) odpovídajících typů se nastaví tak, aby ukazovaly na tyto instance. To proto, aby byl zajištěn přístup k datům i z funkcí, do kterých nemůže být předána hodnota (např. obsluha přerušení).

Ve funkcích s předponou `ODOM_INIT_` se inicializují defaultní hodnoty těchto struktur.

Poté následuje hlavní nekonečná supersmyčka programu.

Pokud je makro zapnutý `DEBUG` mód, simulují se ve smyčce pomocí funkcí s předponou `SIM_` signály, jež reprezentují vazbu od enkodérů a na HW tlačítko vývojové desky je nastaveno simulované vyvolání přerušení jako při příchodu signálu 1PPS. Jednotlivé nastavení jsou přístupná po definování příslušných `DEBUG_` maker v hlavičkovém souboru `INIT.h`.

V pozadí už jsou nyní aktivovány obsluhy přerušení, jež obstarávají ostatní funkce programu.

Iterace

Program poté běží v tzv. iteracích, jenž mohou být zapnuty pro rozdělení intervalu mezi jednotlivým odesíláním odometrické zprávy (`odomsg` více v 1.3.6). Pokud je zapnut timer TIM2, jenž obstarává kontrolu pro spouštění iterací, a vyvolá přetečením svou obslužnou rutinu, porovná se na-sčítaný počet tiků s nastavenými hodnotami dle tabulky 1.3.6. Pokud se má vyvolat iterace nejprve se zkopírují aktuálně napočítané hodnoty tiků enkodérů do bufferů pomocí funkce `ODOM_CALC_ticks_and_reset(...)`. Poté se zavolá funkce `ODOM_iteration(...)`, jenž přepočte aktuální lokalizační údaje a inkrementuje čítač iterací.

Tab. 1.9: Způsob vyvolání iterace při přerušení od TIM2 závisí na nastavení parametru `iter_type`, jenž může nabývat různých hodnot

n	TIM2 nemá na vyvolání iterace vliv
t	iterace je vyvolána vždy po uplynutí timeru TIM2
s	iterace je vyvolána po navýšení součtu enkodérových tiků nad mez definovanou v <code>iter_trig</code>
x	iterace je vyvolána po navýšení součtu tiků od jednoho z enkodérů nad mez definovanou v <code>iter_trig</code>

Iterace je vyvolána vždy když by došlo k přetečení 16-ti-bitových enkodérových timerů (TIM1 nebo TIM8), aby nedošlo ke ztrátě informací o otočení kola. Při nynější konfiguraci (kola a převody) by se musel robot pohybovat rychlostí $2.39 \frac{\text{m}}{\text{s}}$, aby došlo k přetečení. To však v běžných podmínkách provozu nenastane.

1PPS

V případě přijetí signálu 1PPS od GPS nebo při vyvolání jeho simulačních protějšků - přerušení od TIM3, zmáčknutí uživatelského tlačítka - modul spustí funkci `ODOM_PPS_came(...)`, do níž je vložen i znak reprezentující původce signálu: v případě že jde o GPS $\Rightarrow$ [G], interrupt od TIM3 $\Rightarrow$ [I] a pokud dojde ke stlačení tlačítka BUT0 $\Rightarrow$ [B]. Tato funkce vyčte hodnoty enkodérů a provede iteraci, poté zkontroluje jestli nebyla přijata zpráva PCMSG z nad-systému a pokud ano tak ji začne zpracovávat. Dále odešle zprávu ODOMSG obsahující lokalizační informace, a pokud je po přečtení PCMSG z minulého kroku požadováno odeslání zprávy ACTUALMSG, odešle ji. Jestliže je nastaveno relativní ukládání polohy, vynuluje uloženou aktuální polohu aby byla v příštím kroce vypočtena poloha relativní k té nynější. Nakonec

vynuluje i čítač iterací a naopak zvýší čítač kroků, jenž udává počet příchodů (i simulovaných) signálu PPS. Podrobnější informace o zprávách jsou v sekci 2.2.2.

Nastavení robotu a způsobů výpočtu - struktura ODOM_robot

Ve struktuře `ODOM_robot` jsou uloženy základní fyzikální parametry robotu, napočítané tiky, data používaná k výpočtům a nastavení typu spouštění iterací. Jednotlivé proměnné struktury jsou detailněji popsány v příloze C.4.

2 VÝSLEDKY STUDENTSKÉ PRÁCE

2.1 Hardware

2.1.1 Kompletní schéma zapojení

Schematické znázornění zapojení je na obrázku D.3.2. Byly navrženy celkem dvě desky plošných spojů. Redukce k GPS a „shield“ k vývojové desce STM32F4.

2.1.2 Desky plošných spojů

Redukce k GPS modulu

Signál potřebných výstupních dat z GPS modulu je vyveden do 3řadého 26ti pinového konektoru Dsub26. K modulu je dodávána redukce, která však neobsahuje signál 1PPS a proto bylo nutné navrhnout redukci, která obsahuje i tento signál. Pro rozměry desky redukce bylo omezující umístění konektoru Dsub26 (DA26) přímo vedle anténních konektorů TNC. Kompromisem pro šířku DPS byla hodnota 5cm.

Pro GPS-redukci byla pro jednoduchost zvolena jednovrstvá deska. Kvůli spletičnosti rozmístění konektorových pinů bylo nutno použít 3 propoje. Spoje diferenčních signálů byly vedeny co nejbližší k sobě. Schéma D.3.3 a maska pro vedení spojů spolu s konektory D.3.3 je uvedena v příloze.

STM32F4Shield

Aby byla zjednodušena instalace propojení, na vývojovou desku byla navrhnutá deska násadná („shield“), která obsahuje krom pinových konektorů také regulátor napětí a převodník napěťových úrovní RS-232 z GPS na TTL logiku pro MCU. Pro svou v mnoha zařízeních ověřenou funkčnost byl zvolen převodník MAX232.

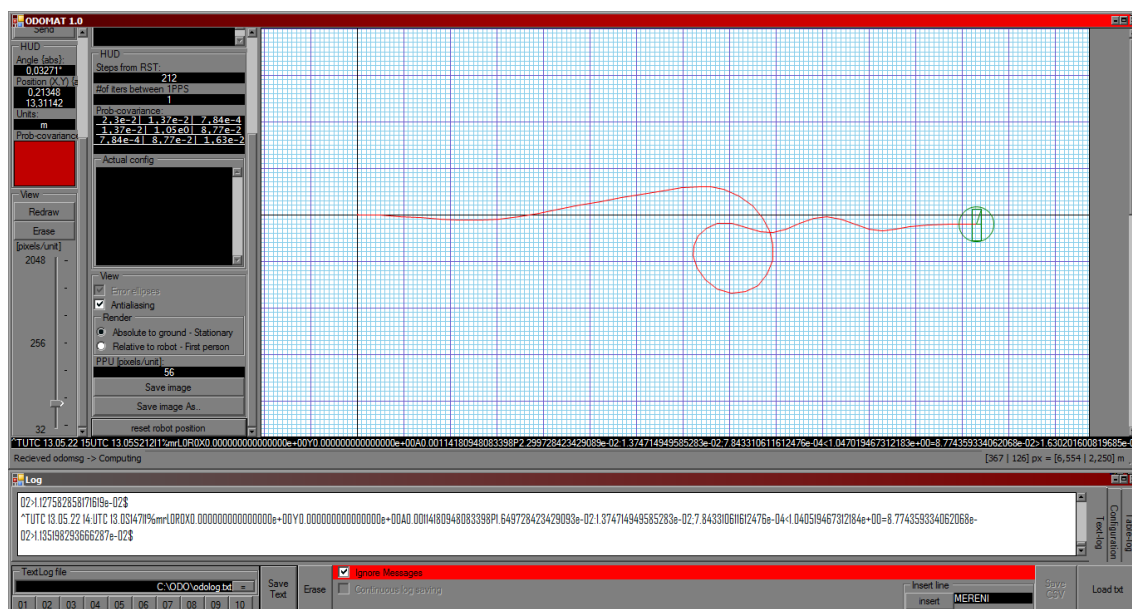
Z GPS je do STM32F4 Shield přiveden i signál RTS (Ready To Send), který však nakonec nebyl pro nadbytečnost implementován. Schéma D.3.3 a DPS D.3.3 je uvedena v příloze.

2.2 Software

2.2.1 ODOMAT - obslužný a vizualizační program

Pro kalibraci, vyčítání hodnot a jejich vizualizaci byl vytvořen program v jazyce C# v prostředí Visual Studio 2010. Po připojení k sériové lince modulu zpracovává, parsuje a loguje zprávy ODOMSG, následně vizualizuje polohu robotu a zakresluje do mapy s mřížkou jeho uraženou trajektorii. Velikost kroku mřížky může být

změněna. Logger umožňuje ukládat zprávy v textovém formátu a také pozastavit příjem zpráv. Je možné zasílat z ODOMATu do modulu zprávy PCMSG, jenž mění nastavení parametrů modulu. Rozhraní je možné lépe vidět na obrázku v příloze C.6.



Obr. 2.1: Vizualizace robotu v programu ODOMAT

2.2.2 Protokol a zasílané zprávy

Jednotlivé druhy zpráv jsou stručně vyjmenovány v tabulce 2.1 a podrobněji popsány dále v textu.

ATT

Pro časový determinizmus lokalizačních zpráv je z GPS přijímána zpráva ASCII časová značka – ASCII time tag (ATT), která je zarovnána k signálu 1PPS. Každou sekundu je z GPS modulu generován puls na lince 1PPS a přibližně 0.5 sekundy před tímto pulzem je na zvoleném portu vyslána zpráva (viz C.7), obsahující textový výpis aktuálního UTC času ve formátu 2.1. Pokud nejsou signály ab známy jsou ve výpisu nahrazeny otazníky ??. Maximální hodnota b je 9 i při vyšším počtu satelitů. Každá ATT zpráva je ukončená kontrolními znaky reprezentujícímu návrat vozíku a posun o řádek (C notace „\r\n“). Počet datových znaků zprávy je vždy 24.

Tab. 2.1: Druhy zpráv

STM32 ⇒ PC	
ODOMSG	lokalizační zpráva obsahující polohu, natočení, pravděpodobnostní matici, počet tiků od poslední iterace ad.
ACTUALMSG	zpráva obsahující aktuální nastavení jednotlivých výpočetních parametrů odometrického modulu
PC ⇒ STM32	
PCMSG	zpráva obsahující příkazy pro STM32
GPS ⇒ STM32	
ATT	UTC čas synchronizovaný signálem 1PPS
GPS ⇒ PC	(=NMEA-0183) pro účely referenčního měření
PTNL,AVR	UTC čas, azimut (yaw) a náklonem (Tilt) robotu
PTNL,VGK	UTC čas a polohou v souřadnicích East North Up od reference

Tab. 2.2: Popis zprávy ATT

UTC yy.mm.dd hh:mm:ss ab (2.1)

UTC ... značí, že se jedná o UTC čas a ne čas v lokaci GPS
yy.mm.dd ... aktuální datum - rok, měsíc a den
hh:mm:ss ... aktuální UTC čas - 24-hodinový formát, minuty, sekundy
a ... číslo celočíselného typu udávajícího typ zaměření GPS (GPS-fix)
b ... počet sledovaných GNSS satelitů

ODOMSG

Zpráva odomsg odesílaná do nad-systému obsahuje zprávu ATT získanou od GPS a další odometrické informace tak jak je naznačeno v 2.2.2.

Tab. 2.3: Příklad zprávy ODOMSG

^TUTC...S114I1%mrL1524R-2982X...Y...A...P...:...;...<...=...>...\$

Ve zprávě je prvními třemi tečkami nahrazena část standardní zprávy ATT. Další vynechávky jsou místo čísel ve formátu X.XXXXXXXXXXXXXX $\pm$ XX

PCMSG

Zpráva je uvozená znakem [^] a ukončená znakem [\$]. Mezi těmito znaky se mohou vyskytovat jednotlivé dvouznakové příkazy, jenž jsou odděleny podtržítkem [_]. Mezi možné příkazy patří [GC] ⇒ Get Config, na tuto zprávu STM reaguje tak, že zašle aktuální nastavení parametrů, zprávou ACTUALMSG. Pokud je PCMSG zaslaná do STM, je zpracována až v následujícím kroku po přijetí PPS signálu. O další krok opožděna je případná odezva v podobě ACTUALMSG. Další možné příkazy jsou předznamenány písmenem [SX] ⇒ Set X, kde za X jsou dosazeny i jiné znaky. Tento typ příkazu slouží k nastavení jednotlivých parametrů odometrického modulu, kdy po příkazu následuje číselná hodnota, na kterou se má parametr nastavit. Všechny příkazy PCMSG jsou dostupné v programu ODOMAT pod tlačítkem „PCMSG Command List“.

ACTUALMSG

Zpráva obsahuje všechna aktuální nastavení odometrických parametrů, jenž jsou uvedena v tabulce C.4. Její formát je podobný formátu zprávy ODOMSG. Zprávu program ODOMAT zpracuje a zobrazí aktuální nastavené parametry desky.

NMEA-0183

Pro tyto zprávy od GPS byl vytvořen parser, který převede data do tvaru vhodného pro vizualizační skript. Další popis v měření 2.3.5 a v dokumentaci [18].

2.3 Testování

Na desce byla provedena řada simulací, jenž dokázali funkční nastavení jednotlivých periférií, správné vyvolávání a reakce na přerušení. Dále byla provedena měření redukující systematickou chybu parametrů a porovnávající odometrii s jiným lokalizačním prostředkem

2.3.1 Měření převodového poměru

Pro zjištění převodového poměru bylo provedeno opakované měření tiků z enkodérů pro každé kolo zvlášť. Každým kolem se provedlo deset otáček a výsledný počet tiků byl zaznamenán. Měření bylo provedeno na obě strany 10krát.

Výsledky Naměřený počet tiků na jednu otočku kola je uveden v tabulce 2.4. Z měření lze dále vyvodit, že převodový poměr převodovky je přibližně 16.76 : 1 do pomala.

Tab. 2.4: Tabulka vypočtených tiků na otočku kola

	T_L	T_R
aritmetický průměr [tiků / 1otáčku]	8582	8581
rozptyl (dle D.3)	4	3

2.3.2 Měření vůle v převodech

Vůle byla měřena tak, že kolem se otočilo na jednu stranu o jednu otočku a poté na druhou s tím, že při první otočce byly zuby převodovky od začátku v záběru.

Výsledky Z tabulky výsledků 2.5 Lze vidět že vůle v levém kole je dvojnásobná oproti kolu pravému.

Tab. 2.5: Tabulka vypočtených vůlí kol a převodovky [tiků]

	Levé kolo	Pravé kolo
Vůle v převodovce [tiků / 1otáčku]	278 ± 14	120 ± 30

2.3.3 Měření scale-faktoru

Pro zmenšení systematické chyby způsobené odlišným nominálním průměrem kol bylo provedeno měření na rovné čáře. Střed mezi koly robotu byl vyznačen ve předu i vzadu aby byl umožněn přímočarý pohyb po rovné podložce. S robotem byla ujeta vzdálenost předem naměřená posuvným metrem a data naměřená odometrií zaznamenána. Bylo provedeno celkem 10 měření.

$$E_S = \frac{L_{\text{real lenght}}}{L_{\text{calc by odometry}}} \quad (2.2)$$

$$r_{L/R} = r_{L/R} E_S \quad (2.3)$$

Rovnice 2.2 vyjadřuje jaký vliv má korekční parametr E_S .

2.3.4 UMBmark měření

Pro zmenšení vlivu systematických chyb způsobených nestejným efektivním poloměrem levého a pravého kola, a odlišnou velikostí efektivní báze od zadaných hodnot,

bylo provedeno měření UMBmark [28]. Na rovné dlážděné podlaze se zanedbatelnými spárami (oproti poloměru kola) byl vyznačen čtverec o straně $L = 5.397\text{m}$ jak je vidět z obrázku 2.3.4. Robot byl umístěn do jednoho z rohů čtverce a jeho souřadnice byly do nulových hodnot. Osa kol byla zarovnána v přímce z jednou stranou čtverce a vyznačený střed osy robotu byl totožný s druhou stranou čtverce. Poté byla s robotem ujeta dráha tak, aby byla jeho osa stále souosa se stranou čtverce. U druhého rohu byl robot zastaven, a otočen na místě o 90deg a následně veden stejným způsobem přes všechny rohy čtverce, dokud nebyl znovu ve výchozí poloze. Poté byly zaznamenány rozdíly polohy v obou osách oproti původním nulovým hodnotám a označeny ex respektive ey . Měření byla provedena 5krát v obou směrech.

UMBmark dokáže vypočítat korekční faktory pro systematickou chybu způsobenou nestejným poloměrem kol E_D a odlišnou velikost báze od její efektivní hodnoty. Rovnice 2.5 až 2.10 popisují jakým způsobem se tato chyba vypočítává z naměřených hodnot. Index CW a CCW značí orientaci po směru a protisměru hodinových ručiček, pokud jsou v indexu obě hodnoty znamenají to že se výpočet provede pro každý směr měření zvlášť.

Při měření na přibližně rovné podlaze, převyšují chyby systematické co do velikosti chyby nesystematické, jak je uvedeno v [28]. A proto výsledky těchto měření mohou být použity pro korekci těchto chyb.

$$\begin{cases} \hat{x}_{CW/CCW} &= \frac{1}{n} \sum_1^n ex_{i,CW/CCW} \\ \hat{y}_{CW/CCW} &= \frac{1}{n} \sum_1^n ey_{i,CW/CCW} \end{cases} \quad (2.4)$$

$$\alpha = \text{průměr} \left(\frac{\hat{x}_{CW} + \hat{x}_{CCW}}{-4L}; \frac{\hat{y}_{CW} - \hat{y}_{CCW}}{-4L} \right) \quad (2.5)$$

$$\beta = \text{průměr} \left(\frac{\hat{x}_{CW} - \hat{x}_{CCW}}{-4L}; \frac{\hat{y}_{CW} + \hat{y}_{CCW}}{-4L} \right) \quad (2.6)$$

$$E_D = \frac{L + b \sin(\beta/2)}{L - b \sin(\beta/2)} \quad (2.7)$$

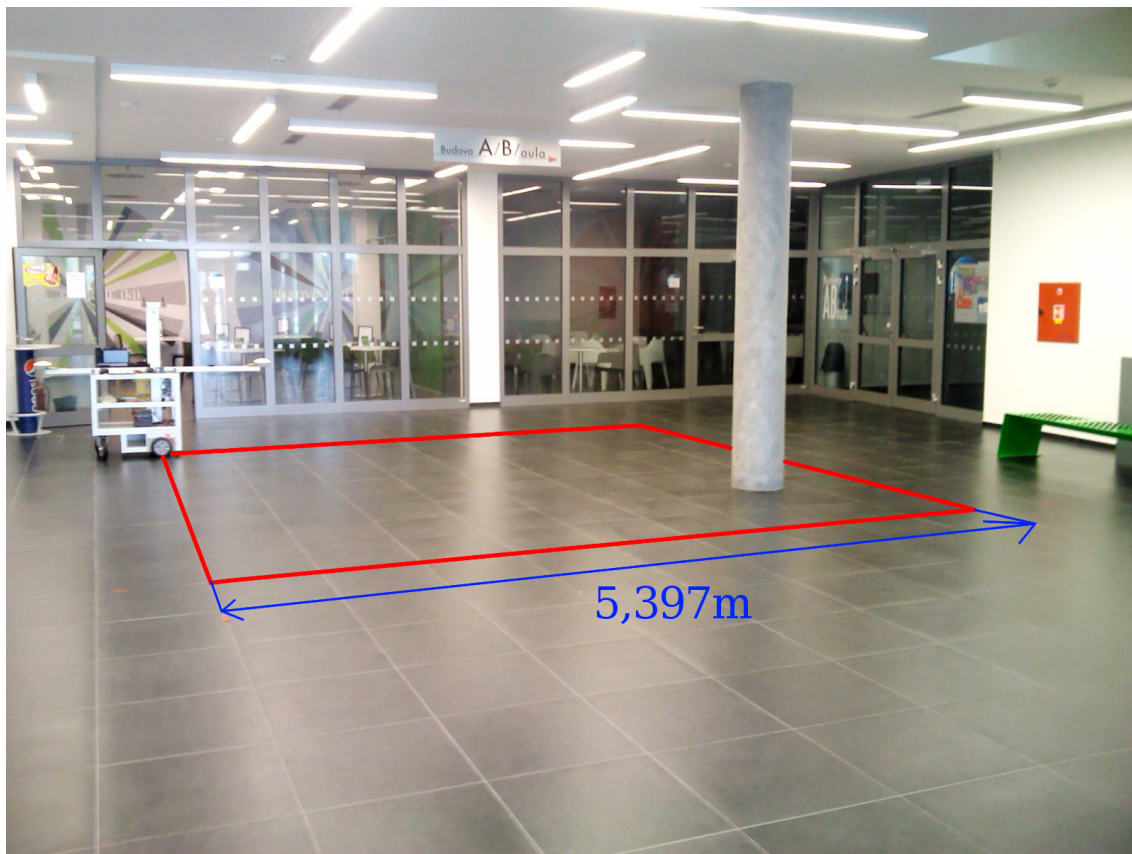
$$c_B = \frac{\pi}{\pi + \alpha} \quad (2.8)$$

$$c_L = \frac{2}{E_D + 1} \quad (2.9)$$

$$c_R = E_D c_L \quad (2.10)$$

Kde L je označena délka hrany čtverce (5.397m); $\hat{x}_{CW/CCW}$ a $\hat{y}_{CW/CCW}$ jsou průměry měření v jednotlivých osách; báze robotu je označena b ; c_L a c_R jsou korekční poměry jimiž jsou přenásobeny poloměry kol.

Pro hodnoty získané měřením byla simulací vypočtena hodnota korekčních veličin.



Obr. 2.2: Náhled na měření UMBmark

Tab. 2.6: Tabulka vypočtených korekčních parametrů UMBmark

	c_B	c_L	c_R
Výpočet se započteným E_D (UMB0.9941)	0.99987	0.99995	1.00005
Výpočet s původním poloměrem (UMB1)	0.99980	0.99995	1.00005

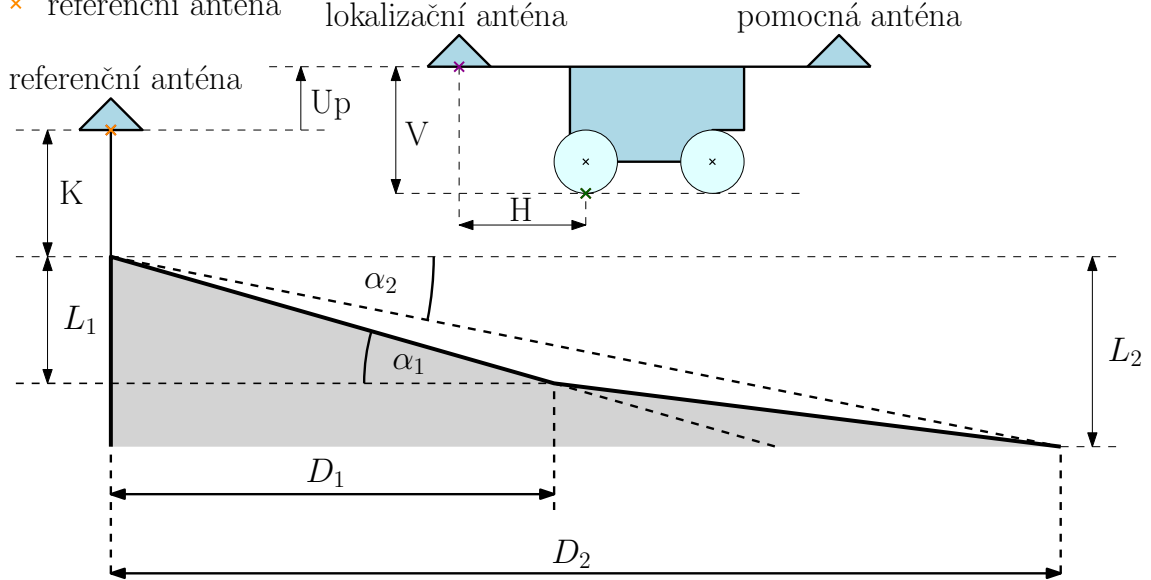
2.3.5 Odometrické měření s GPS referencí

Pro závěrečné zhodnocení byla současně měřena lokalizace pomocí odometrie a lokalizace pomocí GPS modulu instalovaného na robotu. Na parkovišti s nehrbolatým rovným povrchem s konstantním sklonem v jedné ose, byla umístěna stacionární referenční GPS stanice, jež umožnila zasílání vektoru souřadnic přední antény robotu envMap skrze zprávu (NMEA-0183:PTNL,VGK). Tento vektor byl v souřadnicích [East, North, Up], jenž v kartézském systému určují souřadnice vzhledem k Severu, kdy údaj na Z-ové ose odpovídá výšce znázorněné také na obrázku 2.3.5. Zpráva NMEA-0183:PTNL,AVR potom obsahovala azimut (Yaw) robotu v globálních souřad-

nicích a relativní náklon osy antén (Pitch).

Souřadnicové systémy

- × odometrická data
- × přední anténa
- × referenční anténa



Obr. 2.3: Náhled na průřez parkoviště

Podmínky měření Pro změření sklonu povrchu parkoviště byla uložena data z GPS lokalizace, kdy robot byl postaven rovnoběžně z vrstevnicí v různých kolmých vzdálenostech od referenční antény. Poté byl vypočten úhel sklonu ve dvou vzdálenostech. Následně byla provedena dvě referenční odometrická měření, kdy první nepřekročilo vzdálenost D_1 a druhé nepřekročilo vzdálenost D_2 .

$$\alpha_i = \text{atan} \frac{L_i}{D_i} \quad (2.11)$$

$$\Delta_i = L_i \frac{1 - \cos \alpha_i}{\cos \alpha_i} \quad (2.12)$$

Při měření byly zprávy NMEA-0183:PTNL,AVR a NMEA-0183:PTNL,VGK (viz [18]) posílané z GPS několikrát za sekundu logovány do PC. Zároveň byla odometrickým modulem přijímána zpráva ATT obsahující časovou značku. Pro větší frekvenci vzorkování bylo zapnuto vnitřní generování signálu PPS pomocí timeru TIM3 s periodou 0,5 sekundy. ODOMSG zprávy zaznamenávané programem ODOMAT mají ale pouze rozlišení sekund, proto při zpracování bylo ke každému druhému času připočteno 0,5 sekundy. Při zpracování zpráv od GPS byly využity pouze ty které měly stejný UTC čas.

Parsovaná data byla poté zpracována ve skriptu programu MATLAB.

Tab. 2.7: Tabulka vypočtených sklonů plochy parkoviště a maximální chyby

číslo měření	i	1	2
vzdálenost	D_i	1,8295m	3,4685m
referenční výška	L_i	0,0264m	0,0431m
úhel sklonu	α_i	0.8267°	0.7119°
maximální chyba	Δ_i	0.1905mm	0.2675mm

Zpracování měření Při transformaci ze souřadnic GPS na souřadnice odometrické byl zanedbán úhel sklonu parkoviště, protože na měřené délce se projevil méně než chyba GPS lokalizace, která činila přibližně 2 až 4 milimetry. Výpočet chyby transformace je znázorněn na obrázku D.4.2 v příloze. GPS souřadnice přední antény byly pomocí zpráv NMEA-0183 transformovány do souřadnic na středu náhonových kol jak je vidět z obrázku 2.3.5. Úhly na obrázku jsou pouze ilustrační. Poslední operací byla rotace. Úhel o který se data GPS otočily byl vypočítán jako průměr prvních jednotek sekund měření, kdy robot měl relativně rovnou trajektorii.

Odometrický čas byl opatřen offsetem 0,6 sekundy, protože při měření nebyl přítomen synchronizační signál 1PPS. Tento offset byl nastaven tak aby při si odpovídali první data zaznamenávající pohyb od obou lokalizačních prostředků. Následně byly jednotlivé měření přiřazeny k sobě metodou výběru nejbližšího souseda, kdy data GPS která časově nejbližší odpovídala násobkům 0,5 sekundy od počátku pohybu byla přiřazena k odometrickým datům v těchto časech.

Jelikož GPS souřadnice byly vysílány více po skupinách než s konstantní frekvencí, jsou některá odometrická data zatěžkávána chybou tímto vzniklou. Tato odchylka je však ve výsledcích zaznamenána.

Následně byly vizualizovány souřadnice robotu vypočtené oběma lokalizačními prostředky pro různé parametry robotu spolu s elipsou pravděpodobnosti. Tato je vypočítávaná z kovarianční matice pravděpodobnosti $\mathbf{P}_k$. Elipsa v simulaci znázorňuje prostor ve kterém se robot nachází s pravděpodobností 99.73%. Pro vykreslování pravděpodobnostních elips byl použit skript [31], a pro natočení popisů x-ové osy bylo využito skriptu [32].

V grafech byla potom znázorněna relativní odchylka souřadnic a úhlu, spolu s časovou odchylkou GPS. Pro porovnání správnosti lokalizace při jednotlivých nastaveních parametrů (j) byla vypočtena suma prostorové chyby za celkovou dobu pohybu Σ_j . Tato byla následně normalizována chybou měření s neupravenými parametry Σ_1 . Druhým výstupním parametrem je průměrná prostorová chyba (EXY) přes všechny kroky (n). Oba parametry jsou definovány v rovnicích 2.13.

$$e_{x,i} = \text{ODOM}_{x,i} - \text{GPS}_{x,i} \quad (2.13)$$

$$e_{y,i} = \text{ODOM}_{y,i} - \text{GPS}_{y,i} \quad (2.14)$$

$$\Sigma_j = \sum_j^n \sqrt{e_{x,i}^2 + e_{y,i}^2} \quad (2.15)$$

$$Q_j = \frac{\Sigma_j}{\Sigma_1} \quad (2.16)$$

$$EXY_j = \frac{\Sigma_j}{n} \quad (2.17)$$

Výsledky měření Z tabulky výsledků měření vidíme, že nejlepších výsledků bylo na obou trasách dosaženo s nastavením parametrů IV, kdy byla použita jak korekce scale-factor tak korekční faktory vypočtené z metody UMBmark. Tabulka 2.9 udává nově vypočtené parametry robotu s menší systematickou chybou.

Tab. 2.8: Tabulka výsledků pro obě trasy

trasa	set parametrů (j)	I	II	III	IV
A i B	barva	blue	green	red	pink
	E_s	1	1	0.9941	0.9941
	UMBmark	-	UMB1	-	UMB0.9941
A	Q_j	1	1.4704	1.399	0.94033
	EXY_j [m]	0.11833	0.17400	0.16555	0.11127
B	Q_j	1	0.99253	0.98603	0.97835
	EXY_j [m]	0.31069	0.30837	0.30635	0.30396

Tab. 2.9: Tabulka korigovaných parametrů robotu

$r_L = 0,09936 \text{ m}$ $r_R = 0.09941 \text{ m}$ $b = 0,6749 \text{ m}$
--

3 ZÁVĚR

Výsledkem studentské práce je odometrický modul schopný zpracovávat data z inkrementálních enkodérů. Bylo ověřeno, že dokáže přijímat ATT zprávu od GPS. Dále je schopný odesílat zprávy z odometrickými informacemi, informující o absolutní (od zapnutí) nebo relativní poloze robotu. Součástí řešení je také program ODOMAT, jenž umožňuje komunikaci s modulem a nastavení jeho parametrů. Byla provedena měření pro korekci systematických chyb - nestejného poloměru kol a různé efektivní šířky báze od její nominální hodnoty. V tabulce 2.9 jsou uvedeny korigované parametry robotu.

Práce mi pomohla využít a rozšířit své znalosti o mikrokontrolérech a o sériových linkách.

Do budoucna je nutné ověřit či dále upřesnit parametry robotu dalšími měřeními na jiných trasách. Naskýtá se taky možnost tyto parametry vypočítat na počítači z referenčních měření metodou nejmenších čtverců nebo jinou, která by dokázala parametry zpřesnit tak, aby co nejlépe odpovídali realitě. Pro odpovídající výpočet pravděpodobnosti je také nutné správně inicializovat matici pravděpodobnosti. V další fázi vývoje je možné vyzkoušet vliv jednotlivých druhů spouštění iterací i v mezidobí mezi signály 1PPS na celkovou chybu výpočtu. Jako další korekce může být zaveden systém pro připočtení určitého počtu tiků při změně směru, který by korigoval chybu vzniklou vůlí v převodech. Propojit STM s některým z globálních lokalizačních modulů, a využít například Kalmanova filtru pro sjednocení naměřených dat.

V programu ODOMAT může být dále zavedeno ukládání jednotlivých částí zprávy ODOMSG do souboru dle vybraného formátování.

LITERATURA

- [1] LUCAS, G.W. *A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators* [online]. 2000, poslední změna červen 2004 [cit. 2013. 17. 05] . <http://rosum.sourceforge.net/papers/DiffSteer/DiffSteer.html>
- [2] www.SCIO.cz s.r.o. *Statistické pojmy* [online]. 2008 poslední změna 2011 [cit. 2013. 17. 05] http://www.scio.cz/vyzkum/tvorba_testu/statisticke-pojmy/index.asp#rozptyl_smerodatna_odchylka
- [3] VYBÍRAL, B. *ZPRACOVÁNÍ DAT FYZIKÁLNÍCH MĚŘENÍ* [online]. [cit. 2013. 17. 05] <http://fyzikalniolympiada.cz/texty/mereni.pdf>
- [4] LIMPOUCH, Jiří. Šíření chyb ve výpočtech: Zaokrouhlovací chyby. In: *Šíření chyb ve výpočtech: Zaokrouhlovací chyby* [online]. 1999 [cit. 2013-05-26]. Dostupné z: <http://kfe.fjfi.cvut.cz/~limpouch/numet/foluvux/node6.html>
- [5] NOVÁK, Jiří. Základy teorie chyb a zpracování fyzikálních měření. *Základy teorie chyb a zpracování fyzikálních měření* [online]. 2001, č. 1 [cit. 2013-05-26]. Dostupné z: <http://webfyzika.fsv.cvut.cz/PDF/teoriechyb.pdf>
- [6] ČEJKA, Miloslav. Stručný úvod do problematiky nejistot měření. *Stručný úvod do problematiky nejistot měření* [online]. 2007, č. 1 [cit. 2013-05-26]. Dostupné z: <https://www.vutbr.cz/elearning/>
- [7] BORENSTEIN, J. *"Where am I?" Sensors and Methods for Mobile Robot Positioning* [online]. 1996 poslední změna květen 1996 [cit. 2013. 17. 05] <http://www-personal.umich.edu/~johannb/Papers/pos96rep.pdf>
- [8] DUNLAP, G.D. a SHUFELDT, H.H., *Dutton's Navigation and Piloting*, Naval Institute Press, [NON VIDI]
- [9] BOLDIŠ, P. *Bibliografické citace dokumentů podle ČSN ISO 690 a ČSN ISO 690-2* [online]. 2001, poslední aktualizace 11. 11. 2004 [cit. 2013. 17. 05] . Dostupné z URL: <http://www.boldis.cz/citace/citace.html>
- [10] *Swivel Caster* [online]. [cit. 2013. 17. 05] <http://www.gzsealink.com/showpro.asp?ProID=728>
- [11] *STM32 F4 series - marketing slides* [online]. [cit. 2013. 17. 05] http://www.st.com/internet/com/SALES_AND_MARKETING_RESOURCES/MARKETING_PRESENTATIONS/PRODUCT_PRESENTATION/stm32f4_marketing_pres.pdf

- [12] *stm32f4 UM1472, User Manual* [online]. [cit. 2013. 17. 05] http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/DM00037368.pdf
- [13] *stm32f4 reference manual* [online]. [cit. 2013. 17. 05] www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/REFERENCE_MANUAL/DM00031020.pdf
- [14] *stm32f4 datasheet* [online]. [cit. 2013. 17. 05] <http://www.st.com/internet/mcu/subclass/1521.jsp>
- [15] *ebmpapst, bci-63.55 motor reference manual* [online]. [cit. 2013. 17. 05] http://pdf.directindustry.com/pdf/ebm-papst/bci-motors/Show/9357-155506-_9.html
- [16] *Avago TECHNOLOGIES HEDS5500 encoder reference manual* [online]. [cit. 2013. 17. 05] <http://www.avagotech.com/docs/AV02-1046EN>
- [17] *Trimble, gps module BX982 user manual* [online]. [cit. 2013. 17. 05] <http://www.pacificcrest.com/products.php?page=bx982>
- [18] *Trimble, Trimble BD9xx GNSS Receivers Help* [online]. [cit. 2013. 17. 05] http://www.trimble.com/OEM_ReceiverHelp/V4.80/en/default.html#welcome.html
- [19] *Fórum Coocox, 2009 - 2011* [online]. [cit. 2013. 17. 05] <http://www.coocox.org/forum/topic.php?id=346>
- [20] *Dokument, Procedure Call Standard for the ARM Architecture, str.17* [online]. [cit. 2013. 17. 05] http://infocenter.arm.com/help/topic/com.arm.doc.ihl0042d/IHI0042D_aapcs.pdf
- [21] KRAMER, Jeffrey a Abraham KANDEL. On accurate localization and uncertain sensors. *International Journal of Intelligent Systems*. 2012, vol. 27, issue 5, s. 429-456. DOI: 10.1002/int.21530. Dostupné z: <http://doi.wiley.com/10.1002/int.21530>
- [22] SIMON, Dan. *Optimal state estimation Kalman, H[infinity], and nonlinear approaches* [online]. New Jersey: John Wiley, 2006 [cit. 2013-05-23]. ISBN 978-047-0045-343. Dostupné z: <http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471708585.html>

- [23] THRUN, Sebastian. *Probabilistic robotics* [online]. Massachusetts: MIT Press, c2006 [cit. 2013-05-23]. ISBN 0-262-20162-3. Dostupné z: <http://robots.stanford.edu/probabilistic-robotics/fig.pdf>
- [24] KIRIY, Evgeni a Martin BUEHLER. Three-state Extended Kalman Filter for Mobile Robot Localization. *Three-state Extended Kalman Filter for Mobile Robot Localization* [online]. April 12, 2002, č. 1, s. 18 [cit. 2013-05-24]. Dostupné z: http://kom.aau.dk/group/05gr999/reference_material/filtering/eKf-3state.pdf
- [25] SANTANA, André M., Anderson A. S. SOUZA a Ricardo S. BRITTO. Localization of a mobile robot based on odometry and natural landmarks using extended kalman filter. *Localization of a mobile robot based on odometry and natural landmarks using extended kalman filter* [online]. 2008, č. 1 [cit. 2013-05-24]. Dostupné z: <http://repositorio.ufrn.br:8080/jspui/handle/1/6150>
- [26] MURPHY, Kevin P. Gaussians. *Gaussians* [online]. 2006, č. 1, November 24, 2006 [cit. 2013-05-24]. Dostupné z: <http://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall106/reading/gaussians.pdf>
- [27] KOK SENG CHONG a L. KLEEMAN. Accurate odometry and error modelling for a mobile robot. <i>Proceedings of International Conference on Robotics and Automation</i>. IEEE, 1997, inspec5685121, s. 2783-2788. DOI: 10.1109/ROBOT.1997.606708. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=606708>
- [28] BORENSTEIN, J. a L. FENG. UMBmark: A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots. *UMBmark: A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots*. 1994, UM-MEAM-94-22, s. 77. Dostupné z: <http://www-personal.umich.edu/~johannb/Papers/umbmark.pdf>
- [29] LARSEN, Thomas Dall a Karsten Lentfer HANSEN. Design of Kalman Filters for Mobile Robots: Evaluation of the Kinematic and Odometric Approach. *Design of Kalman Filters for Mobile Robots: Evaluation of the Kinematic and Odometric Approach* [online]. 1999, č. 1 [cit. 2013-05-25]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=801027>
- [30] KELLY, Alonzo. Fast and Easy Systematic and Stochastic Odometry Calibration. *Fast and Easy Systematic and Stochastic Odometry Calibration* [online]. 2004, č. 1 [cit. 2013-05-25]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1389908>

- [31] Error_ellipse. In: JOHNSON, A.J. *Error_ellipse* [online]. 2. vyd. 2004 [cit. 2013-05-26]. Dostupné z: <http://www.mathworks.com/matlabcentral/fileexchange/4705-errorellipse>
- [32] Rotate X-axis tick labels. In: TORDOFF, Ben. *Matlab Central* [online]. 2. vyd. 2010, 2013 [cit. 2013-05-26]. Dostupné z: <http://www.mathworks.com/matlabcentral/fileexchange/27812-rotate-x-axis-tick-labels>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ST	mezinárodní firma vyrábějící elektroniku operující hlavně v evropě – STMicroelectronics
MEMS	Mikro elektromechanické systémy – Micro-Electro-Mechanical Systems
GPIO	Vstupy/Výstupy a pro běžné účely – General Purpose Input/Output
USB OTG	USB zařízení umožňující změnu role hosta – Universal Serial Bus On-The-Go
USART	Univerzální synchronní a asynchronní přijmač a vysílač – Universal synchronous asynchronous receiver transmitter
SWD	Sériové drátové debugovací rozhraní – Serial Wire Debug
GNSS	Globální družicový polohovací systém – Global Navigation Satellite System
UTC	Koordinovaný světový čas – Coordinated Universal Time
ATT	ASCII časová značka – ASCII time tag
RTS	Připraven k odeslání – Ready To Send
CMSIS	Standart pro softwareové rozhraní mikrokontrolerů typu Cortex – Cortex™ microcontroller software interface standard
FPU	jednotka zpracovávající operace nad čísly s desetinou částí – Floating Point Unit
GPS	celosvětový (družicový) polohový systém – Global Positioning System
PPS	pulzů za sekundu – Pulse per second
DKF	Diskrétní Kalmanův Filtr – Discrete Kalman Filter
EKF	Rozšířený Kalmanův Filtr – Extended Kalman Filter

SEZNAM PŘÍLOH

A	Obsah přiloženého CD	51
A.1	Software	51
B	Přílohy Technické části	52
B.1	Enkodéry HEDS 5500	52
B.2	GPS Trimble BX982	53
B.3	Kabeláž	53
C	Přílohy Softwarové části	55
C.1	Frekvence jednotlivých komponent	55
C.2	Vývojové prostředí	55
C.2.1	Změny v konfiguračních souborech	56
C.3	Nastavené periferie a piny STM32	57
C.4	Struktura ODOM_robot	57
C.5	Roznásobené maticové výpočty	58
C.6	Program ODOMAT	58
C.7	Komunikační zprávy	58
D	Přílohy Výsledků práce	64
D.1	Statistické výpočty	64
D.2	Doplňující informace	64
D.2.1	Zamezení ztrát informací o poloze	64
D.3	Desky plošných spojů	66
D.3.1	Pinout	66
D.3.2	Celkové schéma zapojení	66
D.3.3	Navrhy Desek plošných spojů	66
D.4	Výsledky měření	66
D.4.1	UMBmark	66
D.4.2	GPS referenční měření	66

A OBSAH PŘÍLOŽENÉHO CD

Naměřené hodnoty

A.1 Software

Vizualizační skript Matlab

Zdrojový kód programu ODOMAT v jazyce C#

Zdrojový kód Firmware pro STM32 v jazyce C pro CoCoX CoIDE

Ostatní pomocné programy

B PŘÍLOHY TECHNICKÉ ČÁSTI



Obr. B.1: Ilustrační foto, pasivní-vlečné kolo typu caster[10]

B.1 Enkodéry HEDS 5500

Tab. B.1: Tabulka vybraných parametrů enkodérů HEDS 5500 [16]

Teoretická $f_{in,max}$	100 kHz
Proud	17mA, max 40mA
Signál	2 hranové signály s 90ti stupňovým offsetem, TTL komptaibilní
Tiků na otáčku	512 (A,B)
$U_{DD} = U_{IN}$	(-0,5 $\Rightarrow$ 7) V
Doporučovaná $R_{pull-up}$	3,2 k Ω – nutné při kabelu delším 4stop

Tab. B.2: Hodnoty odečtené z osciloskopu na enkodérech při běžných rychlostech motoru

t_{rise}	8 μ s
t_{fall}	8 μ s
U_{pk2pk}	5,04V



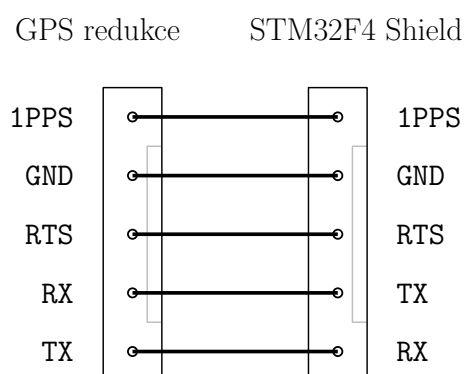
Obr. B.2: Optický enkodér typu HEDS 5500 osazený na robotu envMap[16]

Tab. B.3: Tabulka vybraných parametrů GPS modulu Trimble BDX982 [16]

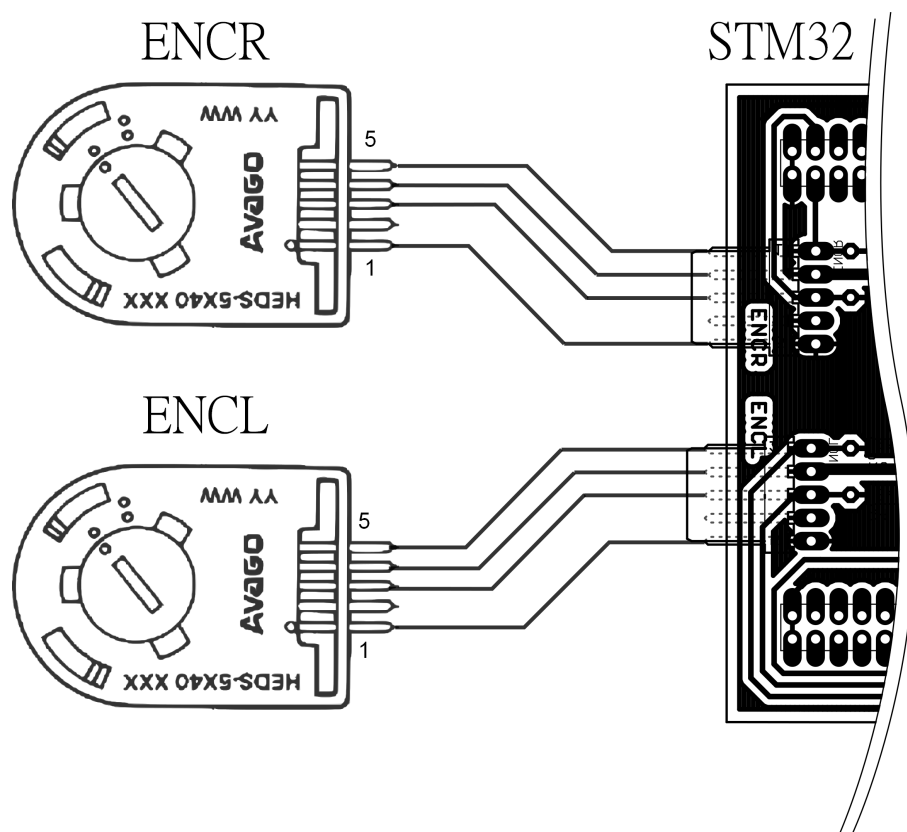
Napájecí napětí	(9 - 28) V DC
Konektory	D-sub DE9, DA26
Rozhraní	1xUSB, 1xCAN, 1xLAN Ethernet, 3xRS232
Maximální baud rate	460800 Bd
Další výstupy	1PPS

B.2 GPS Trimble BX982

B.3 Kabeláž



Obr. B.3: Spojení GPS a STM32F4 pomocí přímého kabelu



Obr. B.4: Připojení enkodérů do desky STM32, pomocí čtyřžilového kabelu

Tab. B.4: Potřebné datové kabely - propojení pin×pin

kabelů×žil	Popis
1×5	Konektory z jsou na shieldu STM rozmístěny tak, že není potřeba použít křížený kabel. Už při použití rovného kabelu je zaručeno připojení TX GPS na RX shieldu a naopak jak je vidět na obrázku B.3.
2×4*	Pro připojení enkodérů HEDS5500 je sice potřeba pouze 4 drátů, ale jelikož je konektor 5-ti pinový, byl stejný zvolen na obou stranách. Kabel samotný však může obsahovat jen 4 žíly.
1×3*	Pro propojení STM s nadsystémem je potřeba RX TX a GND.

C PŘÍLOHY SOFTWAREVÉ ČÁSTI

C.1 Frekvence jednotlivých komponent

Zde je uveden příklad výpočtu frekvence iteračního časovače TIM2, spolu s hodnotami frekvencí jednotlivých komponent.

```
SysClk = 168000000 Hz = 168 MHz
HPRE = 1
HCLK = SysClk / HPRE = 168 MHz
PRE2 = 2
APB2_clock = PCLK2 = HCLK/PRE2 = 168/2 MHz = 84 MHz
PRE1 = 4
APB1_clock = PCLK1 = HCLK/PRE1 = 168/4 MHz = 42 MHz
TIM2CLK = 2 * APB1_clock = 84 MHz (Because PRE1 != 1)
TIM_Prescaler = prescaler register = TIMx_PSC = 42000 - 1
TIM2_counter_clock = TIM2CLK / (TIM_Prescaler + 1) = 2 kHz = 2000 Hz
TIM_Period = auto-reload register = TIMx_ARR = 2000 - 1
TIM2_real_freq = TIM2_counter_clock / (TIM_Period + 1) = 1 Hz
```

kde

SysClk ... Frekvence hlavního jádra

HCLK ... Frekvence hlavní sběrnice AHB

PCLK2 ... Frekvence vysokorychlostní sběrnice APB2

PCLK1 ... Frekvence sběrnice APB1

TIM2CLK ... Frekvence „jádra“ časovače TIM2

TIM2_Real_freq ... Reálná frekvence časovače TIM2, s její hodnotou budou
vyvolávány „overflow“ přerušení

Nastavení prescalerů pro sběrnice tak jak je uvedeno, nastavuje maximální možné frekvence pro jednotlivé sběrnice vývojové desky STM32F4.

C.2 Vývojové prostředí

Po konzultaci bylo k programování zvoleno freeware vývojové prostředí CoIDE, jenž je součástí vývojových nástrojů CoCox, které nabízí podporu pro mikrokontrolery typu ARM Cortex. CoIDE se skládá z CoBuilder programovací platformy a dále z CoDebugger ladícího nástroje pro ARM Cortex Procesory.

CoBuilder podporuje GCC kompilátor, integruje CoDebugger a umožňuje interní sdílení a integraci zdrojů.

CoIde

CoIDE velmi usnadňuje a urychluje vývoj embeded mikroprocesorových aplikací díky podpoře CMSIS. Při vytváření projektu pro STM32F407GT6, byla jednoduše z nabídky vybrána tato vývojová deska a pro přidání potřebných knihoven, stačilo v Repozitáři zatrhnout periferie jenž budou používány, o zbytek se postaralo CoIDE. Muselo být provedeno pár změn v konfiguračních souborech (viz. C.1, C.2 a C.3).

Pro debugging bylo využito rozhraní ST-Link obsažené přímo ve vývojové desce. V nastavení debuggeru byl zvolen protokol SWD a maximální frekvence komunikace 1MHz. IP adresa a port Hosta byla nastavena na 127.0.0.1:2009, Semihosting ani CacheRom nebyly povoleny.

Linker

V Linkeru bylo zvoleno používání základní C-čkové knihovny (`Use base C Library`) nastavení. Přilinkovány byly knihovny potřebné pro matematické výpočty a funkci `printf` a to sice z balíků knihoven `GNU Tools for ARM Embedded Processors` (verze 4.6 2012q4). Byli zvoleny verze zkompileované pro FPU Hard. Přilinkované knihovny: `c`, `g`, `gcc`, `m`.

Ovládací řetězec linkeru byl vygenerován ve tvaru (část cesty k balíku knihoven je nahrazena dvěma tečkami):

```
-mcpu=cortex-m4; -mthumb; -Wl; -nostartfiles; -g; -Map=EXTINT_Flash.map;  
-O0; --gc-sections; -lgcc; -lc; -lm; -L; -lc; -L; -lg;  
-Lc:..\4.6 2012q4\lib\gcc\arm-none-eabi\4.6.2\armv7e-m\fpv\;  
-lgcc; -L; -lm; -L${linkdir}; -T${linkdir}/arm-gcc-link.ld;
```

Kompilér

Kompilér byl nastaven na použití FPU `hard`, která umožňuje rychlejší operace v desetinné čárce, což teoreticky přispívá k více relevantním odometrickým výpočtům, méně zatěžkaných chybou, způsobenou příliš dlouhým krokem iterace. Optimalizace prozatím nebyly zapnuty (`-O0`).

C.2.1 Změny v konfiguračních souborech

Velikost Stacku byla zvětšena, hlavně pro umožnění použití funkce `printf`, editací definovaného makra v souboru `startup_stm32f4xx.c`

```
#define STACK_SIZE 0x00002000 (C.1)
```

Ve stejném souboru byl změněn počáteční ukazatel stacku aby vyhovoval podmínce zarovnání v paměti. Při původní hodnotě byl problém s předáváním float

proměnných do funkcí printf a sprintf.¹

```
(void (*)(void))((unsigned long)pulStack + sizeof(pulStack)), (C.2)
```

Makro definující frekvenci externího oscilátoru v souboru `system_stm32f4xx.c` bylo změněno na správnou hodnotu, protože námi použitá vývojová deska má osazen „nestandardní“ externí oscilátor o frekvenci 8MHz.

```
#define HSE_VALUE ((uint32_t)8000000) (C.3)
```

C.3 Nastavené periferie a piny STM32

Pro přehlednost je zařazena tabulka nastavení periferií.

periferie	zkratka – funkce	přiřazené piny / obslužné rutiny
GPIO	LEDS	pd12,13,14,15
usart2	ODO+PC	tx=pa2, rx=pa3
usart3	GPS	tx=pb10, rx=pb11
EXTI1	příjem PPS	pb1
TIMERS		
TIM1	ENCL	ch1=pe9, ch2=pe11
TIM8	ENCR	ch1=pe6, ch2=pe7
TIM2	iteration	TIM2_IRQn
TIM3	SIM_PPS	TIM3_IRQn
DEBUG		
SIMENC	pins	ENCL=pe0,pe1 ENCR= pe2,pe3
SIMPPS	pins	pe6
SIMPPS	tim	TIM3_IRQn
SIMPPS	btn	pa0

C.4 Struktura ODOM_robot

Sufix `_L` a `_R` značí parametry pro levou a pravou stranu robotu. Všechny proměnné s příponou `_L` existují také ve své pravo-stranné variantě, tyto však byly pro zjednodušení v tabulce vynechány. Znak `[A]` ve druhém sloupci značí, že tento údaj je odesílán ve zprávě `ACTUALMSG`.

¹ viz. Fórum coocox, <http://www.coocox.org/forum/topic.php?id=346> [19] a dokument: Procedure Call Standard for the ARM Architecture, http://infocenter.arm.com/help/topic/com.arm.doc.ih0042d/IHI0042D_aapcs.pdf str.17/33 [20]

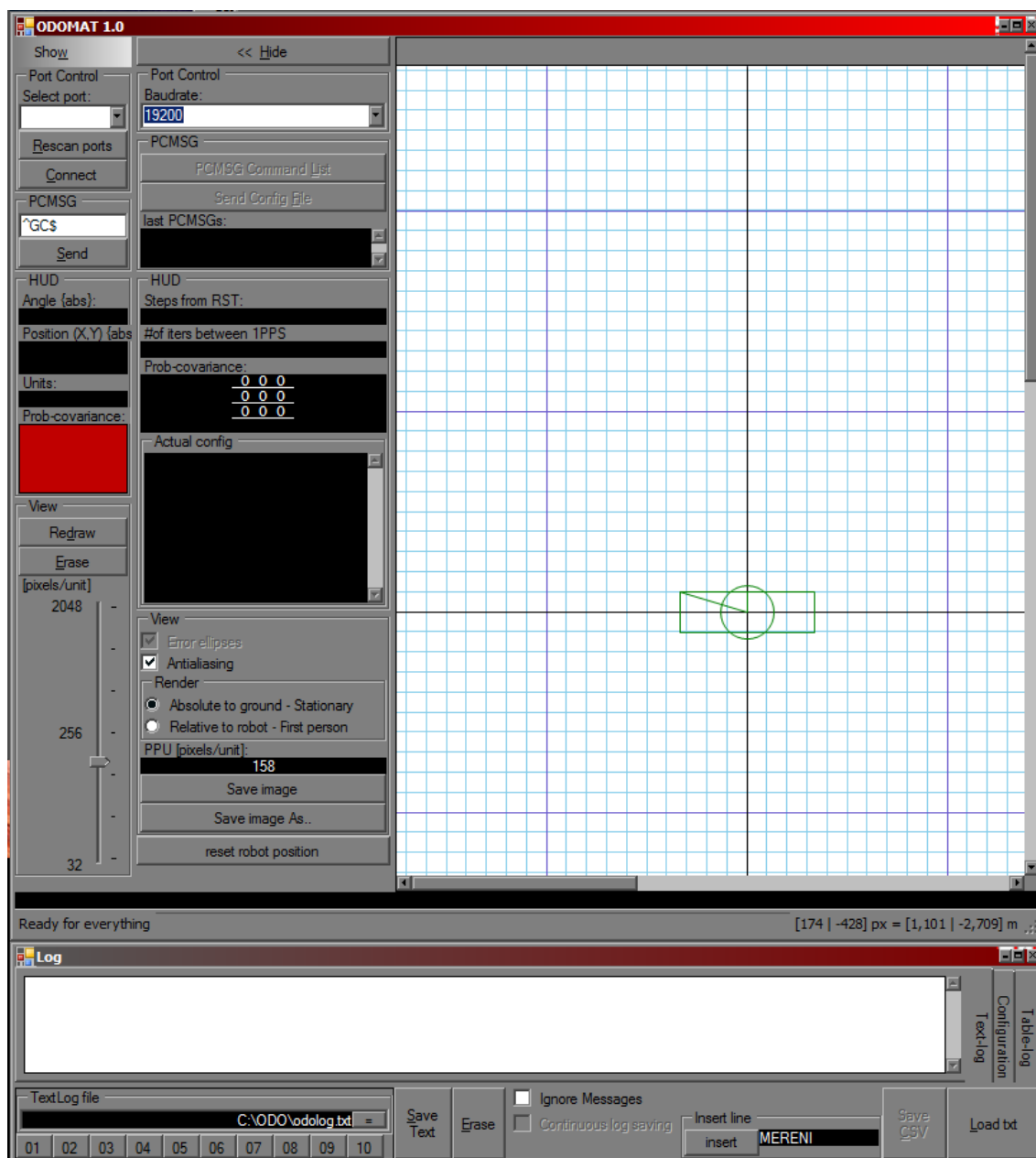
C.5 Roznásobené maticové výpočty

$$\begin{aligned}P11 &= (A13*(A13*P33+P13)+A13*P13+P11) + ((B12*B12)*VR2+(B11*B11)*VL2) \\P12 &= (A13*(A23*P33+P23)+A23*P13+P12) + (B12*B22*VR2+B11*B21*VL2) \\P13 &= (A13*P33+P13) + (B12*B32*VR2+B11*B31*VL2) \\P22 &= (A23*(A23*P33+P23)+A23*P23+P22) + ((B22*B22)*VR2+(B21*B21)*VL2) \\P23 &= (A23*P33+P23) + (B22*B32*VR2+B21*B31*VL2) \\P33 &= (P33) + ((B32*B32)*VR2+(B31*B31)*VL2)\end{aligned}$$

Prvky P_{xx} představují členy matice pravděpodobnosti $\mathbf{P}_k$, prvky A_{xx} členy matice linearizovaného modelu Γ_k , prvky B_{xx} členy matice šumu senzorů $\mathbf{D}_k$ a prvky VL2 a VR2 rozptyly levého a pravého počtu tiků od enkodéru, úměrné aktuálnímu počtu.

C.6 Program ODOMAT

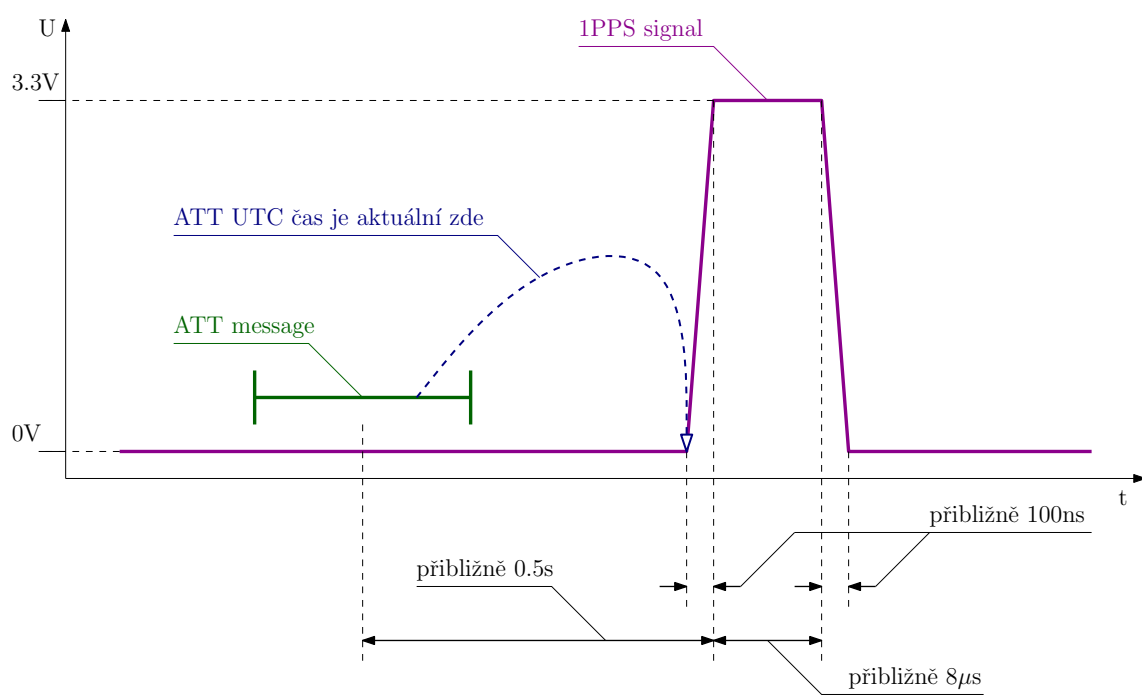
C.7 Komunikační zprávy



Obr. C.1: Rozhraní programu ODOMAT

Tab. C.1: Popis jednotlivých prvků struktury `ODOM_robot`

<code>unit</code>		Zde je nastavena hlavní jednotka vzdálenosti a úhlu a ve které jsou nastaveny fyzikální parametry robotu.
<code>position_store_type</code>		Tato proměnná značí jestli je uložená poloha a natočení robotu po každém odeslání <code>odomsg</code> nastavena na startovní hodnoty. Pokud je zvoleno <code>a</code> – absolutní, poloha se nikdy nerestartuje. Pokud je nastavena na <code>r</code> – relativní je poloha pokaždé reinitializována.
<code>iter_type</code>	A	udává způsob započetí nové iterace
<code>iter_sum</code>	A	ukládá informaci o počtu iteracích provedených mezi jednotlivými odesláními <code>odomsg</code> . Po každém odeslání je nulována.
<code>step_sum</code>	A	počet kroků = přijatých signálů PPS. Nuluje se při restartu
<code>base</code>	A	nastavení rozteče kol, kde se byla změřena vzdálenost středů kol
<code>rL</code>	A	poloměry kol
<code>tL</code>		sem se uloží dosud nasčítané tiky od enkodérů
<code>aL</code>		konstanta úměrnosti mezi počtem tiků a dráhou ujetou kolem
<code>pL</code>		dráha dosud ujetá kolem
<code>TL</code>	A	počet tiků na otáčku kola
<code>tLbuf</code>		buffer pro nasčítané tiky, aby nedošlo k přepsání při vícenásobném ukládání tiků z enkodérů
<code>dth, pC</code>		změna natočení a polohy relativně k poslední hodnotě
<code>x, y, th</code>		aktuální poloha a natočení v kartézské soustavě souřadnic
<code>START_(x,y,z)</code>	A	Startovací poloha do které jsou nastaveny proměnné <code>x</code> , <code>y</code> a <code>th</code> , po odeslání zprávy <code>odomsg</code> , pokud je nastaveno relativní ukládání polohy
<code>calc_prob</code>		Udává jestli se má počítat pravděpodobnost



Obr. C.2: Znázornění signálu ATT a 1PPS v časové souslednosti (překresleno) [17]

Tab. C.2: Vypisování zprávy ODOMSG

```
printf("%c%c%.*s%c%i%c%i%c%c%c%i%c%i%c"
    "%. *e%c%. *e%c%. *g%c%. *e%c%. *e%c%. *e%c%. *e%c%. *e%c%. *e%c\`r\`n",
    '^', // %c
    from, // %c
    ATT_LEN, msg->ATT, // %. *s
    'S', // %c
    rob->step_sum, // %i
    'I', // %c
    rob->iter_sum, // %i
    '%', // %c
    rob->unit, // %c
    rob->position_store_type, // %c
    'L', // %c
    rob->tLbuf, // %i
    'R', // %c
    rob->tRbuf, // %i
    'X', // %c
    DBL_DIG, rob->x, // %. *e
    'Y', // %c
    DBL_DIG, rob->y, // %. *e
    'A', // %c
    DBL_DIG, rob->th, // %. *g
    'P', // %c
    DBL_DIG, rob->P11, // %. *e
    ':', // %c
    DBL_DIG, rob->P12, // %. *e
    ';', // %c
    DBL_DIG, rob->P13, // %. *e
    '<', // %c
    DBL_DIG, rob->P22, // %. *e
    '=', // %c
    DBL_DIG, rob->P23, // %. *e
    '>', // %c
    DBL_DIG, rob->P33, // %. *e
    '$' // %c
```

Tab. C.3: Popis zprávy ODOMSG

ATT_LEN	...	je délka řetězce ATT zprávy – standartně 24 znaků
msg->ATT	...	ATT zprávu uloženou ve struktuře msg
rob->	...	jednotlivé odometrická data viz C.4
DBL_DIG	...	počet vypisovaných číslic za desetinnou čárkou
rob->(x,y,th)	...	vypočtené koncové hodnoty posunutí
	...	a natočení v kartézských souřadnicích,
	...	jenž jsou relativní vůči START souřadnicím, více v C.1

D PŘÍLOHY VÝSLEDKŮ PRÁCE

D.1 Statistické výpočty

Výpočet aritmetického průměru

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n} \quad (\text{D.1})$$

Výpočet směrodatné odchylky

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2} \quad (\text{D.2})$$

Výpočet směrodatné odchylky – vzorek z větší „populace“ možných hodnot

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2} \quad (\text{D.3})$$

Zavedena tzv. Besselova korekce $(n-1)$.

Nejistota nepřímého měření

$$\hat{y} = f(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \quad (\text{D.4})$$

$$u_y = \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial \hat{x}_i} u_{x_i} \right)^2} \quad (\text{D.5})$$

Násobení konstantou

$$\hat{y} = c\hat{x} \quad (\text{D.6})$$

$$\sigma_y = c\sigma_x \quad (\text{D.7})$$

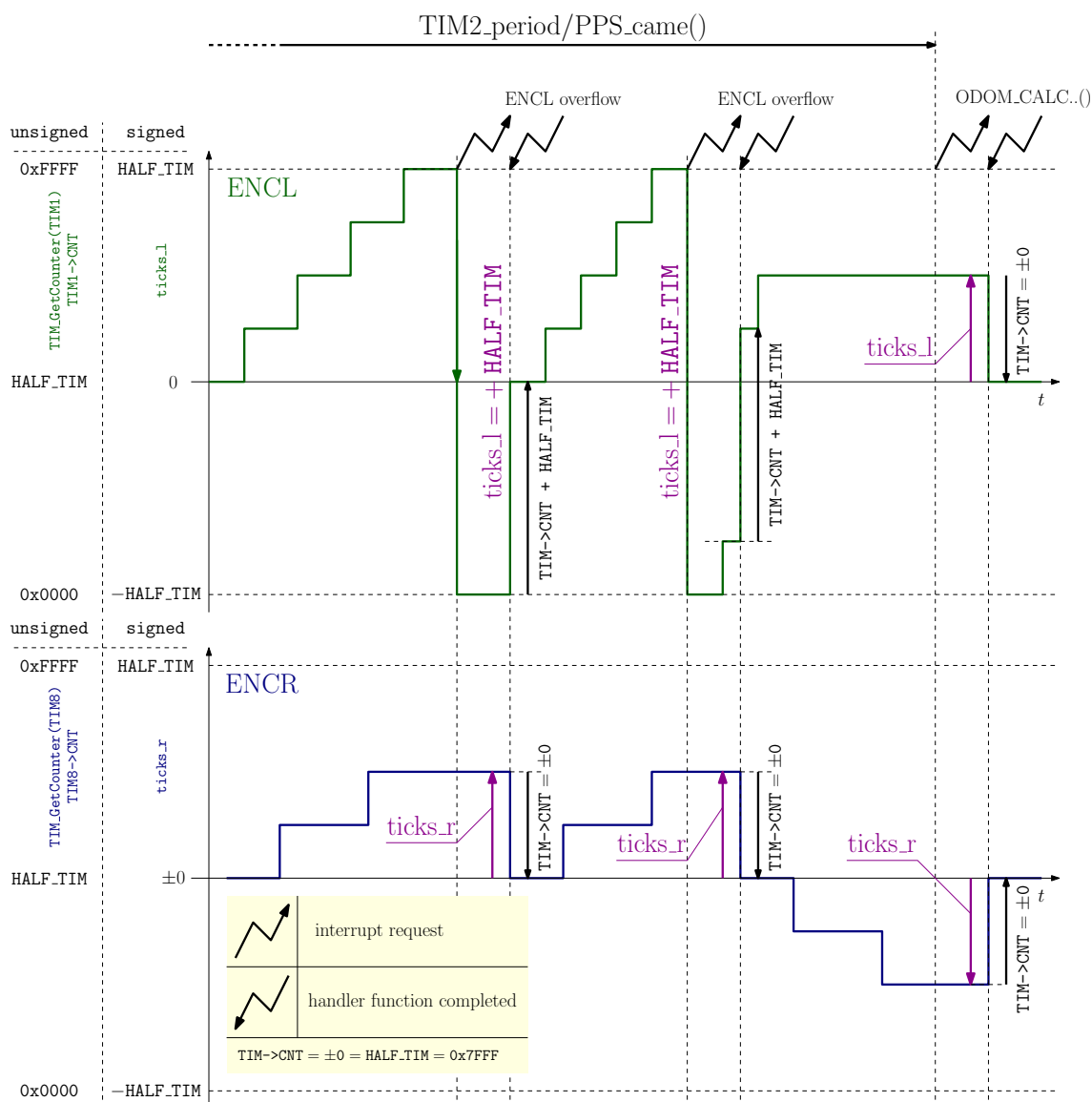
Viz literatura [6] a [5].

D.2 Doplnující informace

D.2.1 Zamezení ztrát informací o poloze

Při přerušení

Časovače TIM1 a TIM8 jsou nastaveny ve funkci enkodérů, což znamená že pokud je vykonáván program a nastane přerušení, není nijak ovlivněno ukládání a vyhodnocování počtu tiků z kvadrurního signálu od enkodérů.



Obr. D.1: Vytvořené ošetření nasčítaných hodnot enkodérových timerů při příchodích přerušních

Přetečení uložené pozice a natočení

Při nastavení odesílání odomsg je praktičtější odesílat relativní posun od START pozice C.1. Při dlouhodobém „ne-nulování“ by mohla být uložená pozice příliš velká pro svůj datový typ, což by teoreticky nakonec vedlo ke ztrátě informace. Restartování pozice také může do budoucna sloužit k nahrávání „absolutní“ pozice získané například z GPS do Startovní pozice. Poté bude v každé odomsg odesílána relativní poloha od Startovní pozice což může sloužit k lokalizaci i mimo dosah GPS.

D.3 Desky plošných spojů

D.3.1 Pinout

Tab. D.1: Přiřazení propojení pinů jednotlivých komponent

GPS		STM32F4	
label	DA26 pin	STM32F4 pin	label
RX1	21	PB10	TX
TX1	12	PB11	RX
RTS1	11	PB14	CTS
1PPS	20	PB1	1PPS

Celkové propojení všech pinů je jasné z Návrhů DPS D.3.3 a D.3.3. Použité piny byly pokud možno voleny tak aby rozmístění na desce nezpůsobovalo zbytečný chaos a aby byly všechny piny FT - tedy odolné 5V napětí.

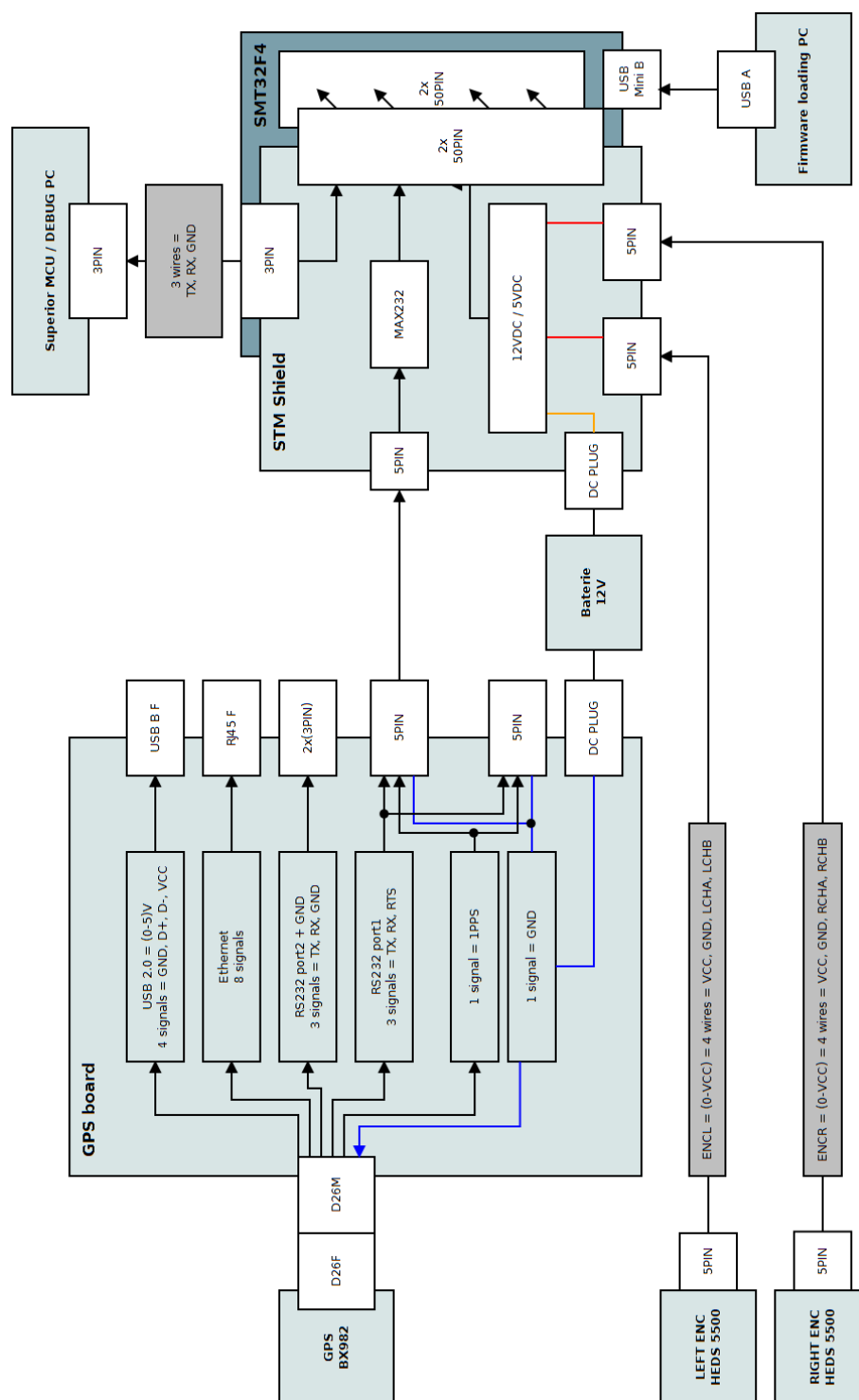
D.3.2 Celkové schéma zapojení

D.3.3 Navrhy Desek plošných spojů

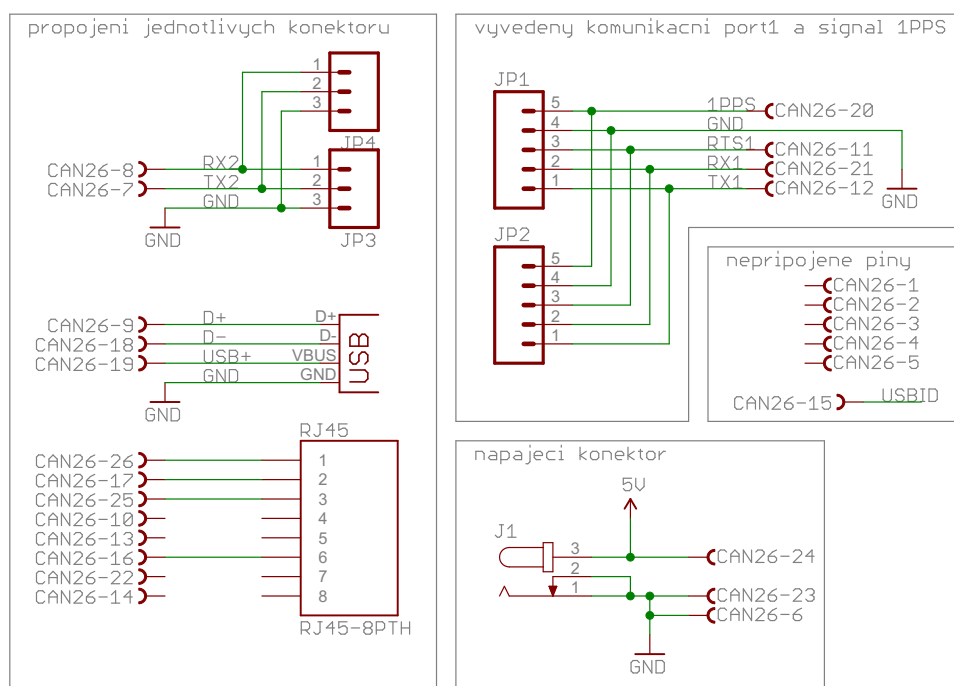
D.4 Výsledky měření

D.4.1 UMBmark

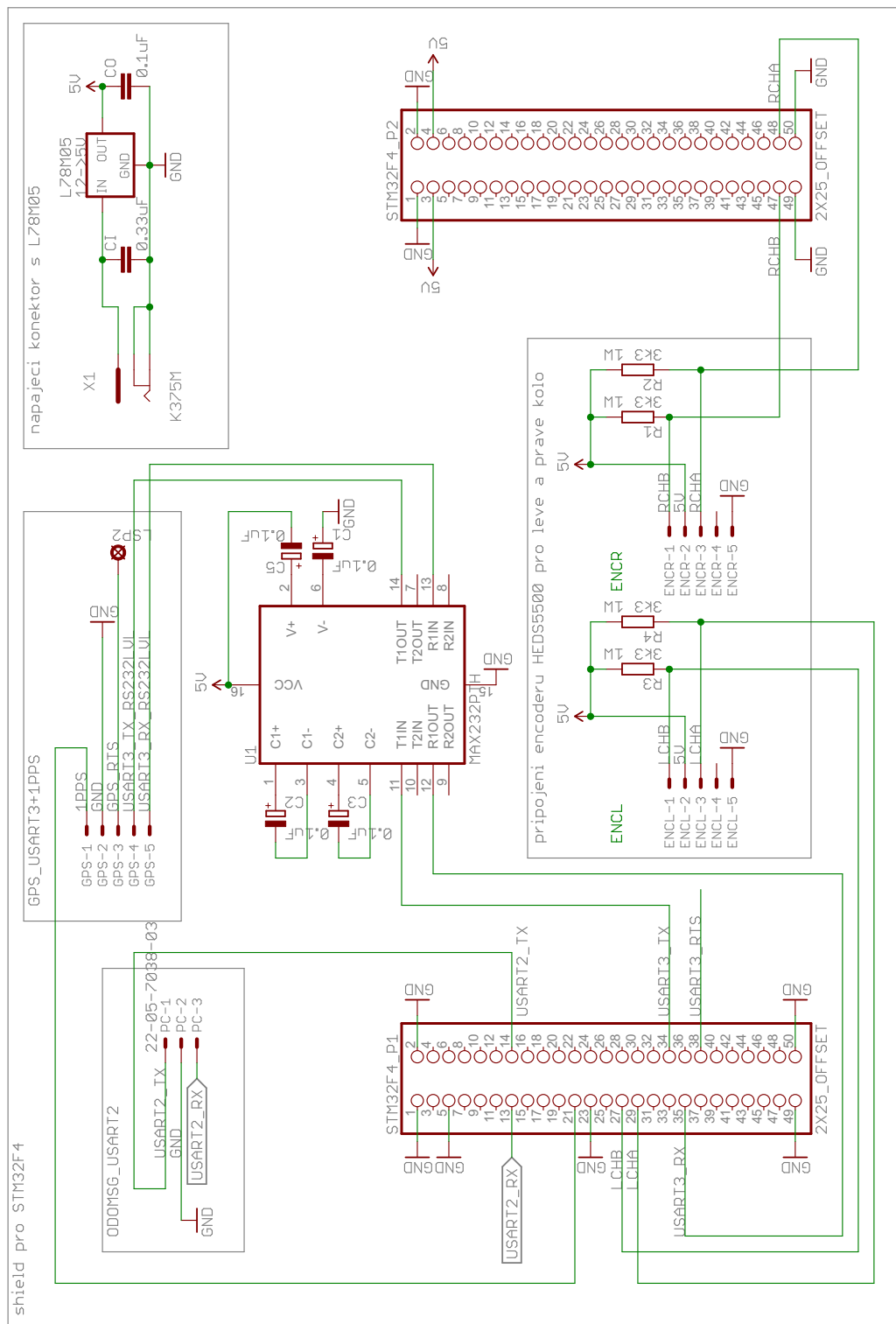
D.4.2 GPS referenční měření

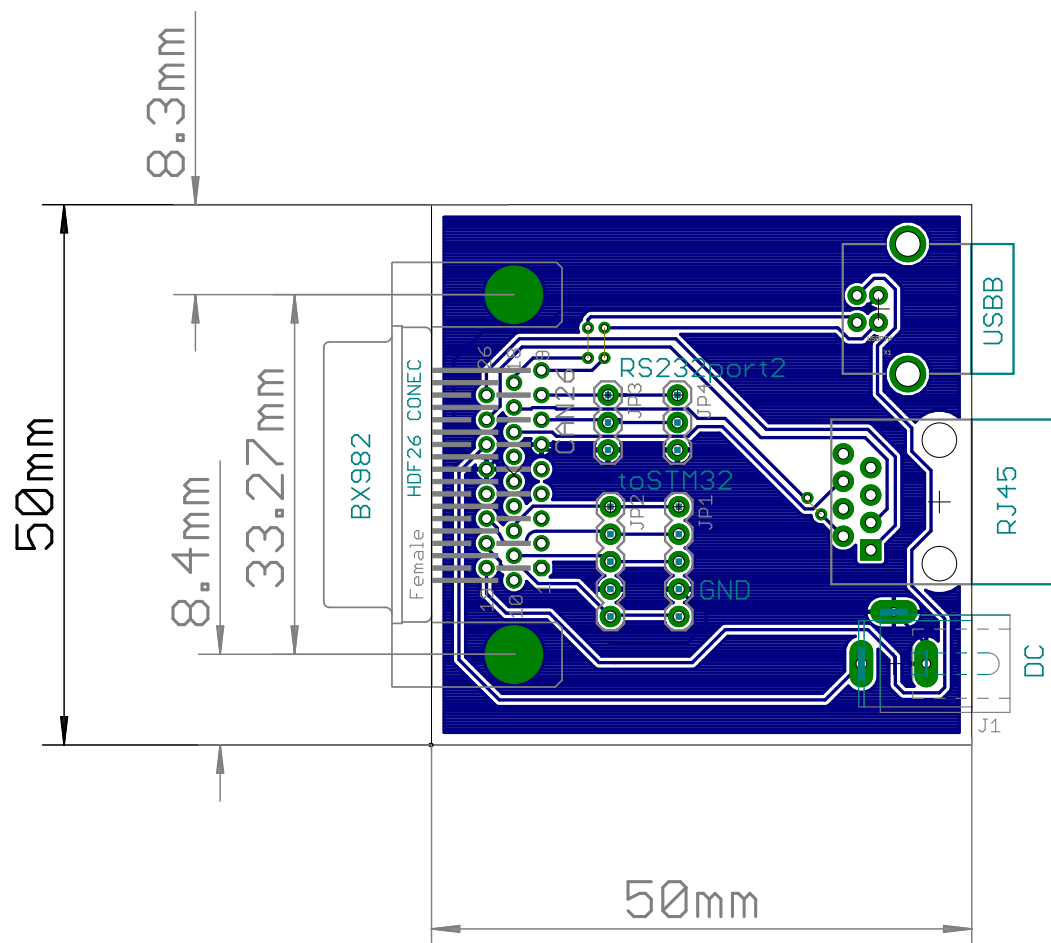


Obr. D.2: Celkové schéma zapojení

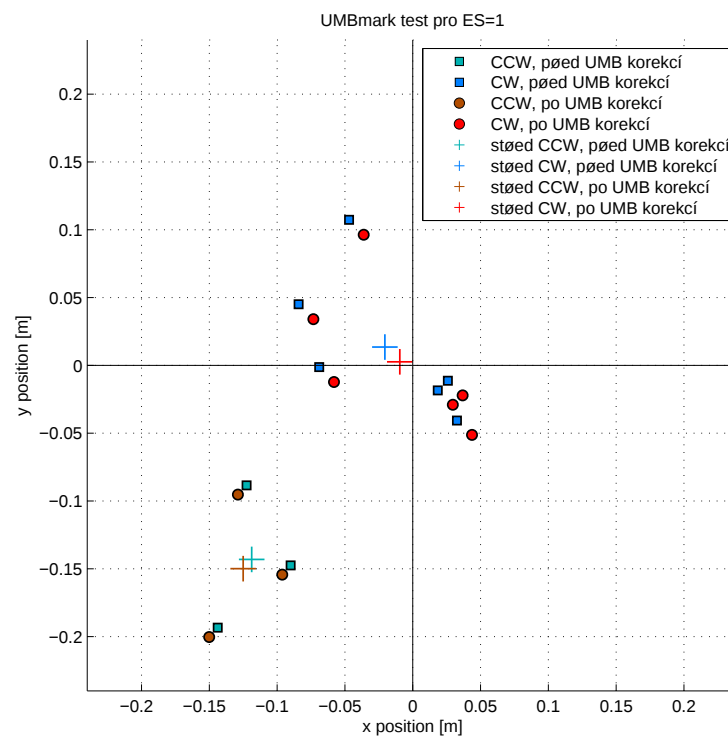


Obr. D.3: GPS redukce - Logické schéma

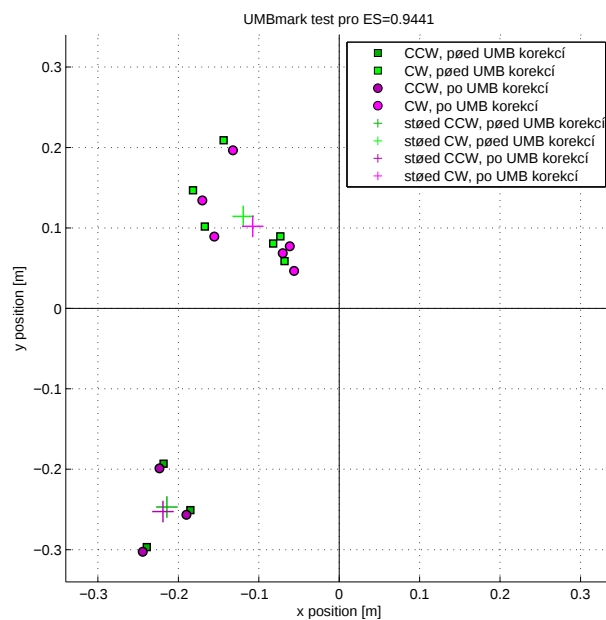




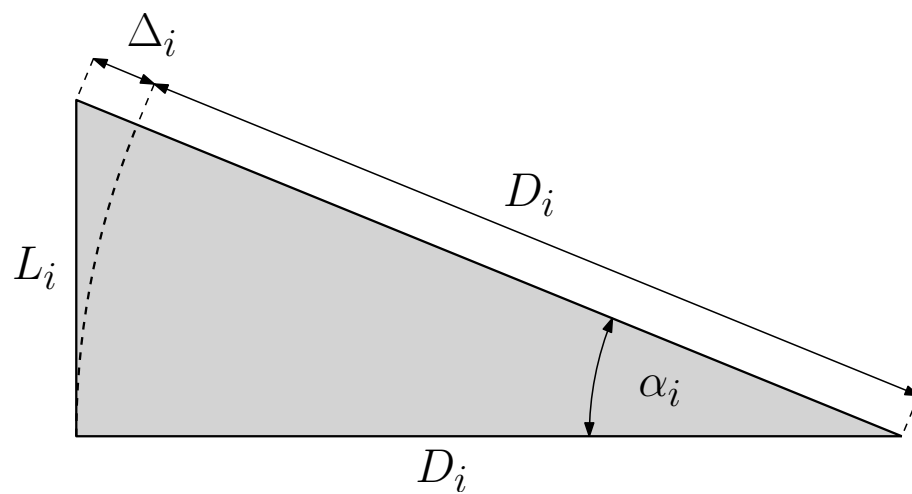
Obr. D.5: GPS redukce - Návrh desky plošných spojů



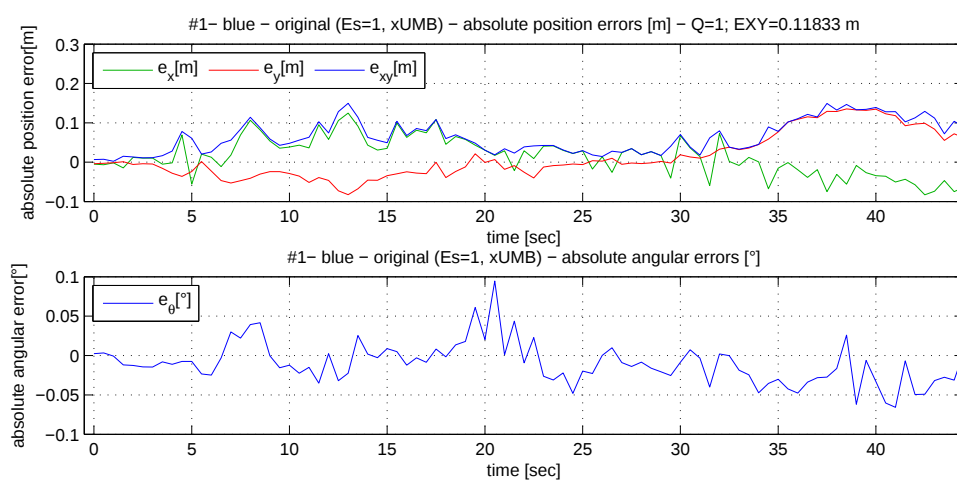
Obr. D.7: Výsledky simulace měření UMBmark do nichž nebyl započítán E_D



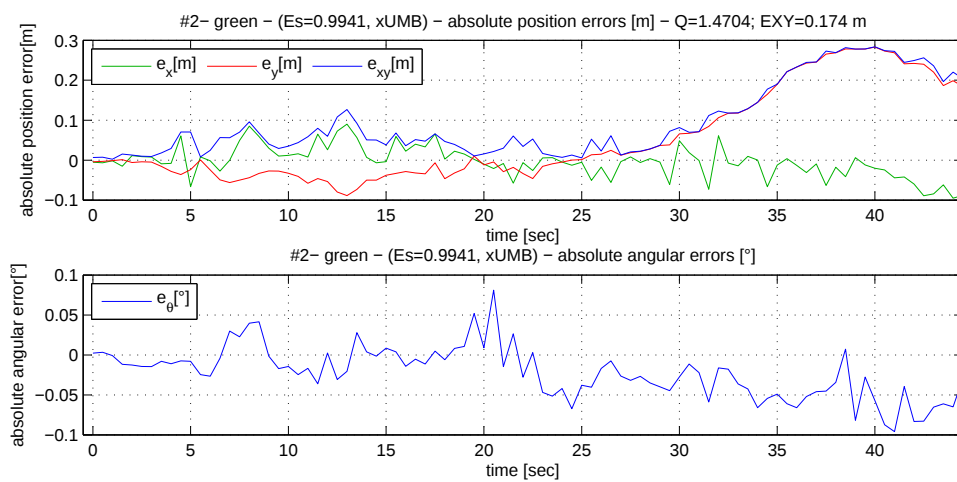
Obr. D.8: Výsledky simulace měření UMBmark do nichž byl započítán E_D



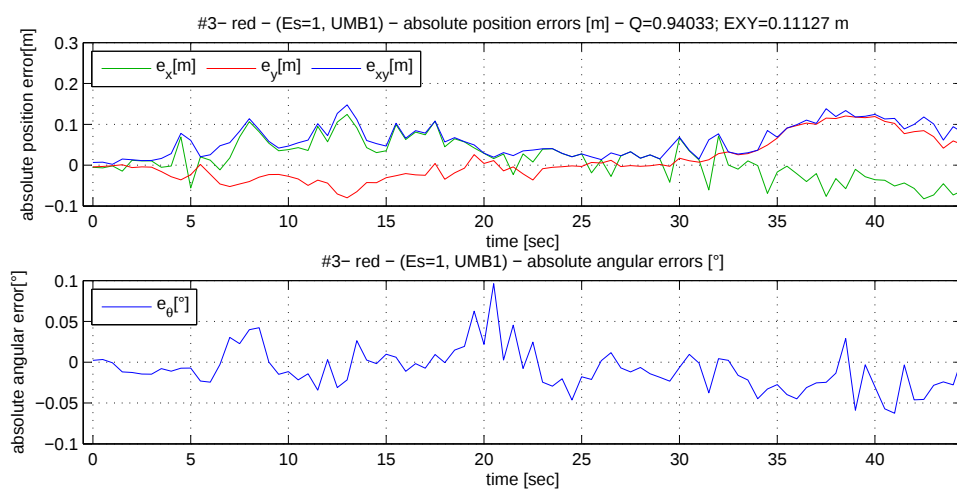
Obr. D.9: Geometrické znázornění výpočtu chyby transformace GPS souřadnic



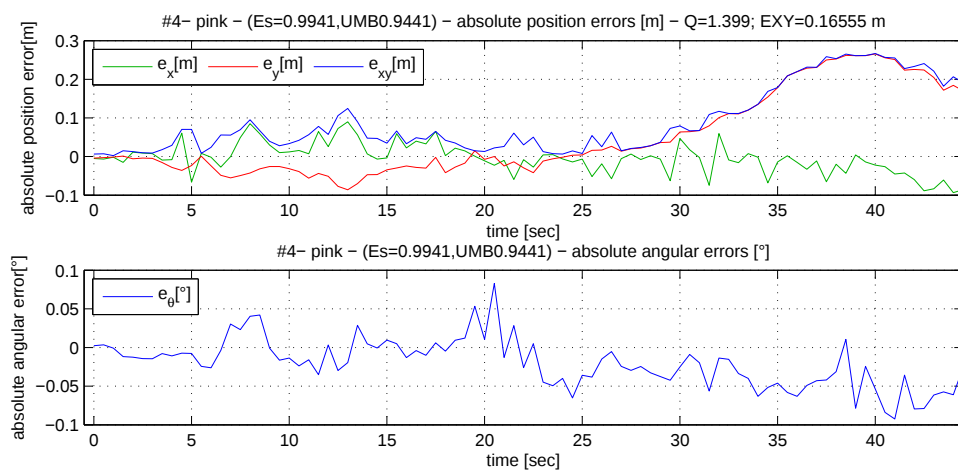
Obr. D.10: Měření s parametry I na trase A



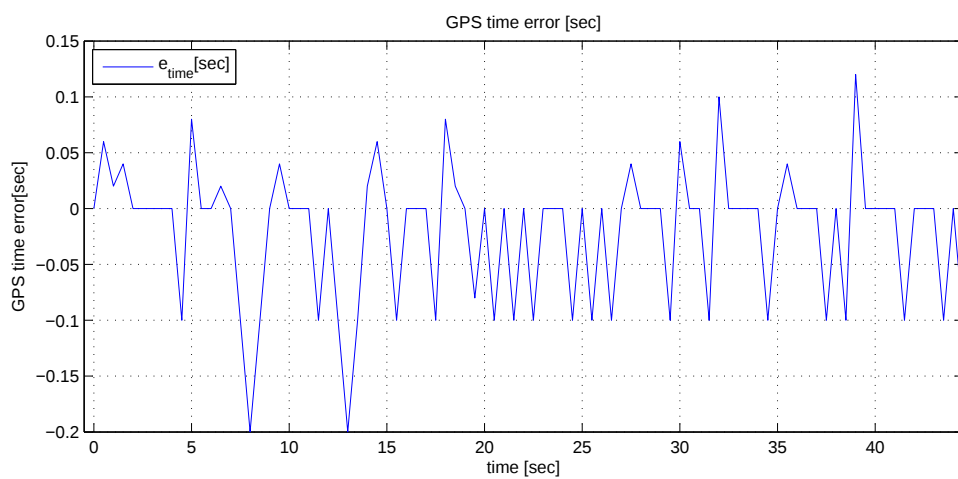
Obr. D.11: Měření s parametry II na trase A



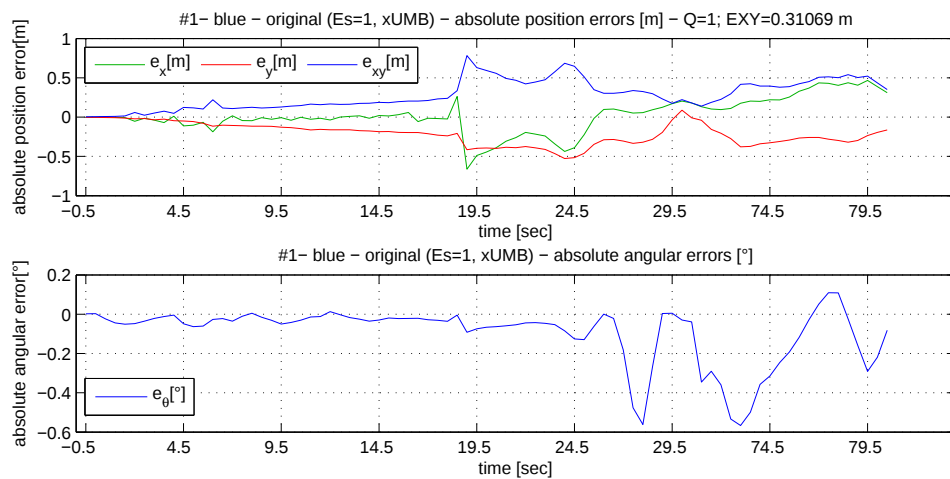
Obr. D.12: Měření s parametry III na trase A



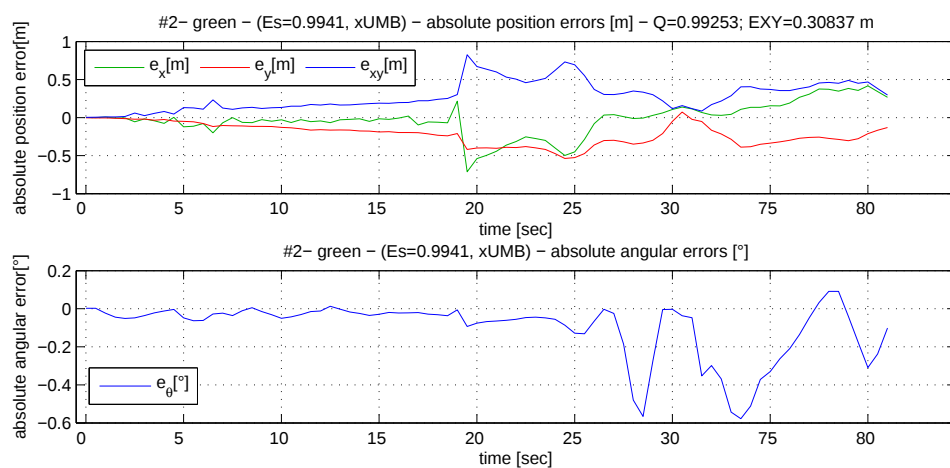
Obr. D.13: Měření s parametry IV na trase A



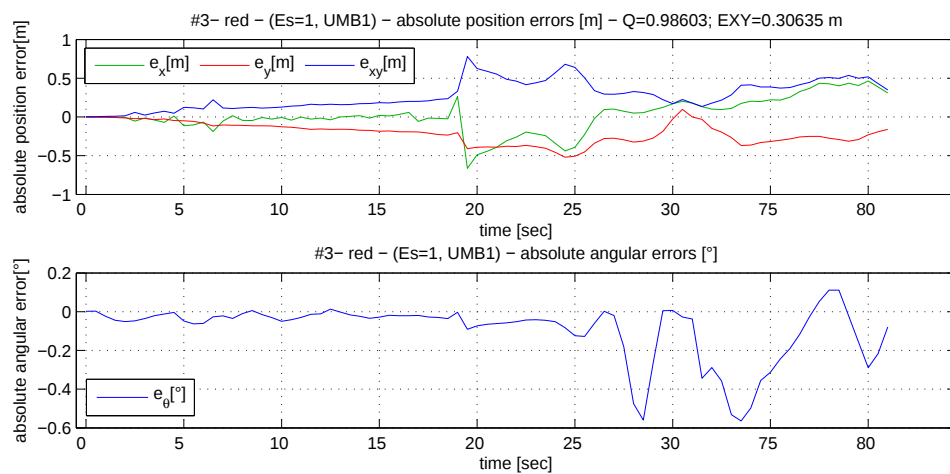
Obr. D.14: Odchylka GPS na trase A



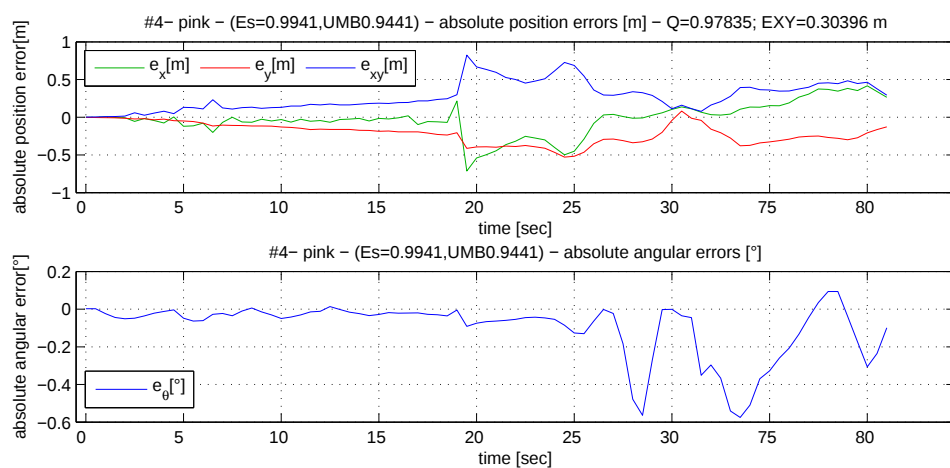
Obr. D.15: Měření s parametry I na trase B



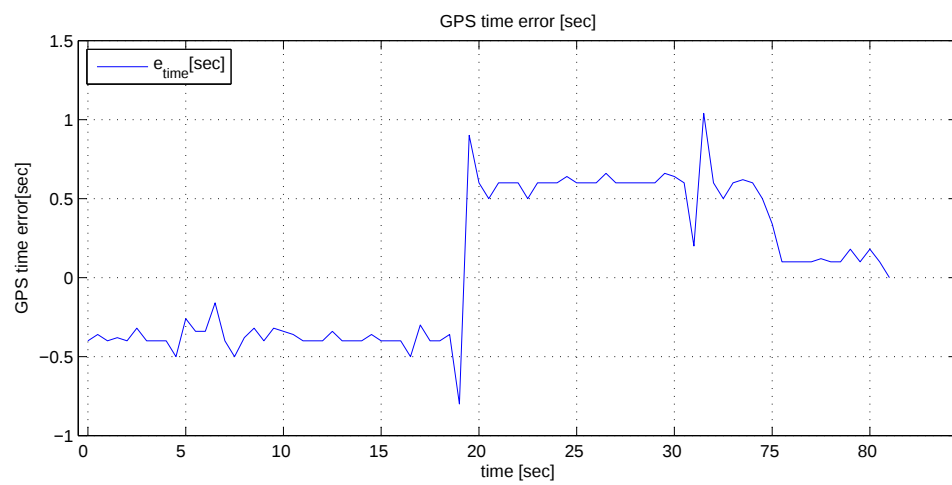
Obr. D.16: Měření s parametry II na trase B



Obr. D.17: Měření s parametry III na trase B



Obr. D.18: Měření s parametry IV na trase B



Obr. D.19: Odchylka GPS na trase B