Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Emulátor letadlového odpovídače SSR

David Moravčík

Diplomová práce

2018

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2016/2017**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. David Moravčík**
Osobní číslo:	**I15221**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Emulátor letadlového odpovídače SSR**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Cílem diplomové práce bude vytvořit softwarový emulátor letadlového odpovídače.
V teoretické části práce budou popsány základní principy sekundární radiolokace s důrazem na roli odpovídačů. Požaduje se, aby emulátor byl řízen podle scénáře. Výstupem emulátoru bude datová struktura. Požaduje se vytvoření návrhového modelu softwarového emulátoru odpovídače.
V praktické části bude realizován emulátor odpovídače.Emulátor bude realizován v jazyku Java.

Rozsah grafických prací:

Rozsah pracovní zprávy: **50-60**

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

Seznam odborné literatury:

***BEAM Walter R. Systems Engineering - Architecture and Design, Mcgraw-Hill, ISBN-13: 978-0070042599**
***OTERO, Carlos E. ,Software Engineering Design Theory and Practice, , ISBN 978-1-4398-5168-5**
***ECKEL, Bruce. Myslíme v jazyku Java: knihovna programátora. Praha: Grada, 2000, 431 s. ISBN 80-247-9010-6.**
***ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. Vyd. 1. Překlad Bogdan Kiszka. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.**
***PECINOVSKÝ, Rudolf. Návrhové vzory: 33 vzorových postupů pro objektové programování. Computer Press, 2007. ISBN 978-80-251-1582-4.**
***ANNEX 10 volume 4: Airbor ne Collision A v oidance System (A CAS) Manual. ICAO. [online]. [cit. 2014-10-14]. Dostupné z: http://www.icao.int/Meetings/anconf12/Document%20Archive/9863_cons_en.pdf**

Vedoucí diplomové práce: **Ing. Karel Šimerda**
Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2016**
Termín odevzdání diplomové práce: **17. května 2017**

Univerzita Pardubice Česká republika

L.S.

Ing. Zdeněk Němec, Ph.D.
děkan

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2016

Prohlášení autora

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 18. 5. 2018

David Moravčík

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat vedoucímu této diplomové práce, panu Ing. Karlu Šimerdovi za všechny návrhy, připomínky a čas, který mi a mé práci věnoval, a podklady, které mi pro vypracování poskytnul.

ANOTACE

Cílem diplomové práce je návrh a tvorba softwarového emulátoru letadlového odpovídače SSR (Secondary Surveillance Radar). V teoretické části budou představeny základní principy sekundární radiolokace včetně role odpovídačů. V praktické části bude realizován softwarový emulátor odpovídače SSR v jazyce Java.

KLÍČOVÁ SLOVA

odpovídač, sekundární radiolokace, emulátor, Java

TITLE

Aircraft transponder emulator.

ANNOTATION

This thesis is dedicated to design and implementation of software emulator of SSR (Secondary Surveillance Radar) transponder for aircraft. The theoretical part will be dedicated to introduction of base principles of secondary radiolocation and the role of transponders. For the practical part, the SSR transponder emulator will be implemented in Java programing language.

KEYWORDS

transponder, secondary radiolocation, emulator, Java

OBSAH

0 Úvod 15

1 Základní principy a rozdělení radarů 17

1.1 Oblasti uplatnění radarů 17

1.1.1 Navigace 17

1.1.2 Ostraha 17

1.1.3 Detekce přírodních objektů 17

1.1.4 Speciální aplikace 17

1.2 Typy radarů a jejich aplikace 18

1.2.1 Primární radar 18

1.2.2 Sekundární radar 18

1.2.3 Pasivní radar 19

2 Sekundární radary 21

2.1 Historie 21

2.2 Princip 21

2.3 Zjištění polohy cíle 22

2.3.1 Měření radiální vzdálenosti cíle 22

2.3.2 Měření azimutu cíle 23

2.3.3 Měření výšky cíle 25

2.3.4 Měření rychlosti cíle 25

2.4 Používané signály a módy 26

2.4.1 Mód 3 / A 26

2.4.2 Mód C 28

2.4.3 Mód S 30

2.5 Problémy systému 32

2.5.1 Hlavní zdroje problémů 32

2.5.2 Překrývání odpovědí (garbling) 33

2.5.3 Chybné vyhodnocení částí dvou odpovědí jako platný cíl 34
3 Systém ADS-B....35
3.1 GNSS....36
3.2 Extended Squitter (ES)....36
3.3 Formát ADS-B rámce 37
3.3.1 ICAO adresa 38
3.3.2 Type Code (TC)....38
3.3.3 Součet kontrolní parity (CRC)....39
3.4 Identifikace letadla (Aircraft identification) 39
3.4.1 Četnost vysílání....41
3.5 Poloha ve vzduchu (Airborne position) 41
3.5.1 Navigation Integrity Category (NIC)....42
3.5.2 Surveillance status 43
3.5.3 Kódování výšky 44
3.5.4 Kódování polohy....45
3.5.5 CPR kódování....45
3.5.6 CPR systém souřadnic 46
3.5.7 Četnost vysílání....54
3.6 Rychlost ve vzduchu (Airborne velocity) 54
3.6.1 Subtype 55
3.6.2 Navigation Accuracy Category (NAC)....56
3.6.3 Kódování rychlosti vůči zemskému povrchu (Ground Speed)....56
3.6.4 Kódování letové rychlosti (Airspeed)....58
3.6.5 Kódování vertikální rychlosti (rychlost stoupání / klesání)....60
3.6.6 Rozdíl oproti barometrické výšce (pole S_DIFF a DIFF) 61
3.6.7 Četnost vysílání....62
3.7 Poloha na zemi (Surface Position) 62

3.7.1 Navigation Integrity Category (NIC) 62
3.7.2 Informace o pohybu (Movement) 63
3.7.3 Kódování směru 64
3.7.4 Kódování polohy na zemi 65
3.7.5 Četnost vysílání 65
3.8 Útoky na systém ADS-B 65
3.8.1 Pasivní útoky 65
3.8.2 Aktivní útoky 66
3.8.3 Vybrané aktivní útoky 67
4 Implementace 70
4.1 Platforma 70
4.1.1 Java 70
4.1.2 JavaFX 71
4.1.3 NetBeans IDE 72
4.1.4 XML 72
4.1.5 JAXB 73
4.2 Struktura aplikace 77
4.2.1 Balíček `utils` 77
4.2.2 Balíček `adsb` 77
4.2.3 Balíček `aircraft` 81
4.2.4 Balíček `responder` 86
4.2.5 Balíček `simulation` 89
4.2.6 Balíček `controllers` 89
4.2.7 Balíček `fxutils` 91
4.3 Princip generování chyby určení polohy a rychlosti 91
4.3.1 Chyba určení polohy 91
4.3.2 Chyba určení rychlosti 92

4.3.3 Implementace generování chyb polohy a rychlosti 92
5 ZÁVĚR 95
6 Použitá literatura 97
7 Přílohy 99

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1 – Základní typy aktivních radarů 18
Obrázek 2 – Kombinovaná anténa primárního a sekundárního radaru 19
Obrázek 3 – Princip SSR 22
Obrázek 4 – Směrová charakteristika hlavní a potlačovací antény 24
Obrázek 5 – Průběh dotazu v módu 3 / A 27
Obrázek 6 – Formát odpovědí v módu 3 / A 28
Obrázek 7 – Průběh dotazu v módu C 29
Obrázek 8 – Průběh odpovědi v módu C 29
Obrázek 9 – Průběh adresného dotazu v módu S 30
Obrázek 10 – Průběh všeobecného dotazu v módu S 31
Obrázek 11 – Průběh odpovědi v módu S 32
Obrázek 12 – Překrytí odpovědí v módech A, 3 nebo C 33
Obrázek 13 – Validní odpovědi neexistujících cílů (Fantom) 34
Obrázek 14 – Systém ADS-B 35
Obrázek 15 – Princip určení polohy pomocí GNSS 36
Obrázek 16 – Short a Extended Squitter [zdroj: vlastní] 37
Obrázek 17 – Příklad ADS-B rámce 38
Obrázek 18 – Registrační značka natištěná na trupu letadla 39
Obrázek 19 – Příklad ADS-B zprávy pro identifikaci cíle (datová část) 40
Obrázek 20 – Zóny zeměpisné šířky (červená: sudé, modrá: liché) 47
Obrázek 21 – Indexování zón zeměpisné šířky 48
Obrázek 22 – Velkosti sudých a lichých zón zeměpisné šířky 48
Obrázek 23 – Vztah mezi sloty a zónami 49
Obrázek 24 – Příklad $\boldsymbol{YZi}$ pro zeměpisnou šířku 43,054°N 49
Obrázek 25 – Zóny zeměpisné délky (červená: sudé, modrá: liché) 50
Obrázek 26 – Redukce počtu zón zeměpisné délky směrem k pólům (pouze sudé zóny) 51
Obrázek 27 – Příklad $\boldsymbol{XZi}$ pro zeměpisnou délku 76,06°W 52
Obrázek 28 – Příklad zobrazení neexistujícího cíle řídícím letového provozu 68
Obrázek 29 – Příklad zobrazení 100 neexistujících cílů řídícím letového provozu 68
Obrázek 30 – Diagram tříd: Modul ADS-B 78
Obrázek 31 – Diagram tříd: Modul Aircraft 82
Obrázek 32 – Přiklad bodů trasy letadla a aktuální poloha 84

Obrázek 33 – Diagram tříd: Modul Responder 86
Obrázek 34 – GNSS: Skutečná a zjištěná poloha 91
Obrázek 35 – Graf: 1000 náhodně vygenerovaných hodnot 93
Obrázek 36 – Hlavní okno 102
Obrázek 37 – Hlavní nástrojová lišta 103
Obrázek 38 – Panel entit 104
Obrázek 39 – Panel výstupu 105
Obrázek 40 – Panel TCP serveru 105
Obrázek 41 – Panel podrobností o letadle 106
Obrázek 42 – Dialogové okno pro editaci letadla 107
Obrázek 43 – Dialogové okno pro editaci samostatných zpráv 109

Tabulka 1 – Použité konvence a styly 16
Tabulka 2 – Seznam módů sekundárního radaru 26
Tabulka 3 – Speciální kódy módu 3 / A 26
Tabulka 4 – Formát ADS-B rámce 37
Tabulka 5 – Příklad zápisů 24 bitové ICAO adresy 38
Tabulka 6 – Typy ADS-B zpráv podle hodnoty Type Code (TC) 39
Tabulka 7 – Formát ADS-B zprávy pro identifikaci letadla (pouze datová část) 40
Tabulka 8 – Formát ADS-B zprávy pro polohu ve vzduchu (pouze datová část) 42
Tabulka 9 – Kódování NIC 43
Tabulka 10 – Kódy pro pole SS (Surveillance status) 44
Tabulka 11 – Počet bitů nezbytných k zakódování polohy s a bez CPR 45
Tabulka 12 – Formát ADS-B zprávy pro rychlost ve vzduchu (pouze datová část) 55
Tabulka 13 – Kódy pro pole ST (Subtype) 56
Tabulka 14 – Kódy pro pole NAC 56
Tabulka 15 – Nastavení bitů S_EW a S_NS 57
Tabulka 16 – Příklad kódů pro pole V_EW nebo V_NS 58
Tabulka 17 – Příklad kódů pro pole HDG 59
Tabulka 18 – Příklad kódů pro pole AS 59
Tabulka 19 – Příklad kódů pro pole VR 60
Tabulka 20 – Příklad kódů pro pole DIFF 61
Tabulka 21 – Formát ADS-B zprávy pro polohu na zemi (pouze datová část) 62

Tabulka 22 – Kódování NIC (ADS-B zpráva o poloze na zemi) 63
Tabulka 23 – Rozsahy kódů pro pole MOV a jejich význam.....63
Tabulka 24 – Příklad kódů pro pole HDG (ADS-B zpráva o poloze na zemi) 64
Tabulka 25 – Znaky a kódy pro volací znak identifikační ADS-B zprávy.....100
Tabulka 26 – Kódy pro pole Emitter Category (EC).....101

SEZNAM ZKRATEK A ZNAČEK

SSR	Secondary Surveillance Radar (Sekundární přehledový radar)
ICAO	International Civil Aviation Organization
ADS-B	Automatic Dependent Surveillance – Broadcast
CPR	Compact Position Reporting
CRC	Cyclic Redundancy Check
XML	Extended Markup Language
UML	Unified Modeling Language
IDE	Integrated Development Environment
JVM	Java Virtual Machine
GUI	Graphic User Interface
CSS	Cascading Style Sheets
OOP	Objektově Orientované Programování
TCP	Transmission Control Protocol
GPS	Globální Poziční Systém
GNSS	Global Navigation Satellite System
CPR	Compact Position Reporting
HAE	Height Above Ellipsoid

0 ÚVOD

Tato diplomová práce se zaobírá problematikou letadlových odpovídačů, které hrají klíčovou roli v rámci systému sekundární radiolokace v letecké dopravě. Sekundární radar je zařízení, které vysílá signál a místo jeho odrazu přijímá a zpracovává odpověď vyslanou cílem. Nedílnou součástí sekundárního radaru je tedy i palubní část, odpovídač (transpondér), který je umístěn na palubách civilních i vojenských letadel.

V úvodní části bude provedeno porovnání principů primární a sekundární radiolokace. Budou představeny základní principy sekundárního přehledového radaru SSR (Secondary Surveillance Radar). Dále zde bude popsána role palubního odpovídače při identifikaci a lokalizaci cíle, včetně získávání dalších informací jako rychlost a nadmořská výška cíle.

Následující část bude věnována kódování a formátu zpráv, pomocí kterých palubní odpovídače s pozemní stanicí komunikují. Bude představen způsob jednoznačné adresace odpovídače, způsob kódování polohy, rychlosti a volacího znaku pro identifikaci letadla (callsign). Tyto zprávy budou výstupem softwarového emulátoru realizovaném v praktické části diplomové práce.

Závěrečná část práce bude zaměřena na praktický návrh softwarového emulátoru palubního odpovídače na platformě Java. Vstupem emulátoru bude scénář ve formátu XML popisující trajektorii a rychlost cíle, včetně dalších údajů jako jednoznačná adresa a volací znak. Výstupem emulátoru budou zprávy v binární podobě definované specifikací ADS-B. Součástí emulátoru bude také grafické uživatelské rozhraní (GUI) realizované s pomocí frameworku JavaFX a TCP server umožňující distribuci výstupních zpráv jiným procesům. Při implementaci emulátoru bude kladen důraz na dodržování základních principů objektově orientovaného návrhu (OOP) a možnost dalšího rozšiřování funkcionalit aplikace.

Použité konvence a styly

Následující tabulka (Tabulka 1) definuje význam formátování použitého v rámci této diplomové práce.

Tabulka 1 – Použité konvence a styly

Styl	Význam	Příklad
Normální styl, velká písmena	Zkratky a označení.	ADS-B
Kurzíva	Zkrácené názvy jednotek.	*m/s*, *GHz*
Neproporcionální písmo s pevnou šířkou	Programový kód nebo části kódu, názvy metod, parametrů, tříd a balíčků použité v textu.	`IAdsbPayload`

1 ZÁKLADNÍ PRINCIPY A ROZDĚLENÍ RADARŮ

Slovo radar vzniklo jako zkratka anglického výrazu „RAdio Detection And Ranging“, což v doslovném překladu znamená rádiové odhalování a určování vzdálenosti. Radar je tedy zařízení určené k detekci vzdálených předmětů za pomoci rádiových vln, k měření jejich vzdálenosti a případně i polohy.

Radiolokace je činnost využívající radar pro zjišťování přítomnosti a polohy objektů, tzv. cílů. V civilním letectví, přesněji ve službách řízení letového provozu, jsou předmětem našeho zájmu letadla. Radiolokací tedy rozumíme zjišťování objektů v prostoru a určování parametrů jejich pohybu a případně dalších informací prostřednictvím elektromagnetických vln.

(BEZOUŠEK, a další, 2004 str. 1)

1.1 Oblasti uplatnění radarů

Tato práce je zaměřena především na radary využívané v civilním letectví za účelem řízení letového provozu.

1.1.1 Navigace

- Pozemní radary pro řízení letového provozu – ATC (Air Traffic Control)
- Palubní letecké radary (zejména pro vojenská letadla)
- Palubní lodní radary
- Pobřežní radary pro kontrolu a řízení lodního provozu

1.1.2 Ostraha

- Ochrana vzdušného prostoru (protivzdušná obrana – PVO)
- Ochrana uzavřených prostor proti vniknutí (poplachové systémy)

1.1.3 Detekce přírodních objektů

- Meteorologické radary
- Mapování povrchu země
- Geologický průzkum

1.1.4 Speciální aplikace

- Výškoměry, Hladinoměry
- Čidla pro automatizaci průmyslových procesů
- Proti srážkové radary v automobilech
- Radary pro měření rychlosti vozidel

- Vojenské aplikace

(BEZOUŠEK, a další, 2004 stránky 1–2)

1.2 Typy radarů a jejich aplikace

Radary, které samy vyzařují své rádiové vlny nazýváme radary aktivní. Aktivní radary se dále dle principu funkce dělí na radary primární a radary sekundární (viz. Obrázek 1). Další skupinou jsou radary pasivní, které pouze přijímají elektromagnetické vlny vyzařované jinými objekty.

Obrázek 1 – Základní typy aktivních radarů
[zdroj: (BEZOUŠEK, a další, 2004 str. 3)]

1.2.1 Primární radar

Primární radar vysílá elektromagnetickou vlnu, která se šíří okolním prostorem směrem od antény radaru. Při střetu s jakýmkoliv objektem se část vyzářené energie odrazí zpět k anténě radaru, kde je zachycena a vyhodnocena. Ze zpoždění mezi vyslaným a přijatým signálem lze určit vzdálenost objektu. Na základě aktuálního natočení (směru) antény lze zjistit polohu cíle vůči radaru. Na základě těchto dvou údajů a znalosti polohy radaru lze velmi přesně určit souřadnice zachyceného objektu. Z Dopplerova posuvu kmitočtu přijatého signálu lze stanovit vzájemnou radiální rychlost radaru a cíle.

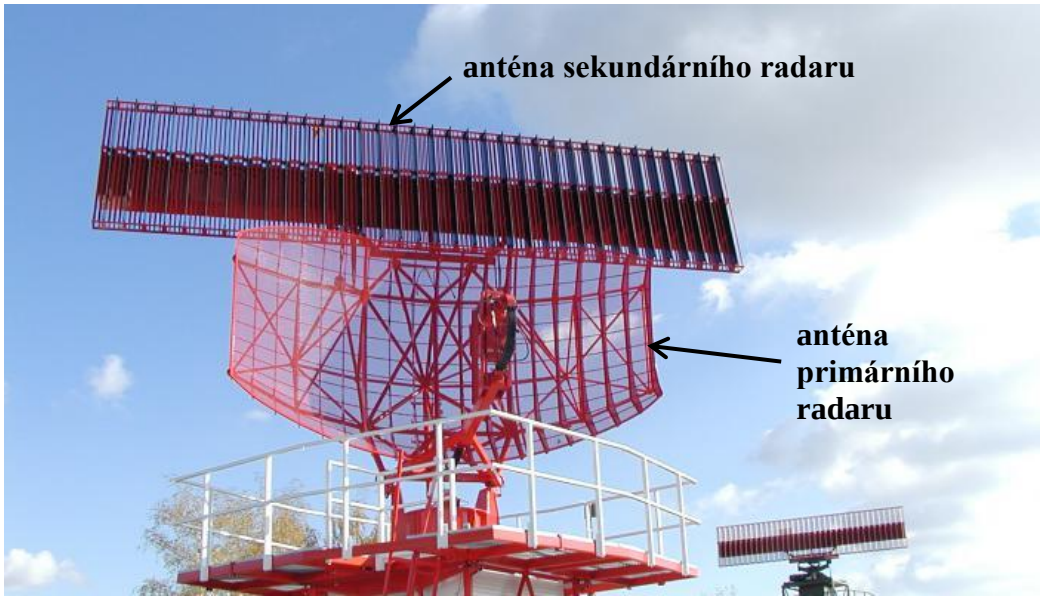
Informací, kterou nelze získat využitím primárního radaru, je jednoznačná identifikace zachyceného objektu. Z tohoto důvodu jsou pro účely řízení letového provozu využívány také radary sekundární. V letectví se primární radar využívá zejména pro vojenské účely k ostraze vzdušného prostoru a pro odhalování cílů, které na dotazy sekundárního radaru neodpoví.

1.2.2 Sekundární radar

Sekundární radar stejně jako primární vysílá elektromagnetickou vlnu, která se šíří okolním prostorem směrem od antény radaru. Na rozdíl od primárního radaru však místo odražené vlny přijímá signál vyslaný zachyceným cílem. Tento způsob radiolokace vyžaduje, aby měl zachycený cíl na palubě aktivní zařízení nazývané odpovídač (angl. transponder). Ten zachytí

signál vyslaný radarem (dotazovačem) a po definované prodlevě odešle definovanou odpověď, která je přijata dotazovačem a vyhodnocena. Tímto způsobem lze mezi radarovou stanicí a cílem přenášet libovolné informace, jakými jsou například jednoznačná identifikace cíle a údaje palubních přístrojů jako GPS souřadnice a nadmořská výška.

Na rozdíl od primárních radarů schopných detekovat a zobrazit všechny objekty[1], jelikož všechny objekty do jisté míry odráží elektromagnetické vlny, sekundární radary mohou detekovat pouze spolupracující objekty vybavené odpovídači. Pokud je odpovídač na palubě cíle nefunkční nebo vypnutý, je takovýto cíl pro systém sekundární radaru neviditelný. Z tohoto důvodu se pro řízení letového provozu využívají oba systémy. Anténa sekundárního radaru bývá v praxi často umístěna přímo nad anténou radaru primárního, kde se otáčí souběžně.



Obrázek 2 – Kombinovaná anténa primárního a sekundárního radaru
[zdroj: (TECH ANNOUNCER, 2017)]

1.2.3 Pasivní radar

Pasivní radary samy o sobě žádné elektromagnetické vlny nevysílají, pouze zachytávají signály vytvářené jinými objekty. Zdroje těchto signálů mohou být umístěny v samotných objektech, nebo jsou vytvářeny jiným externím zařízením, a od cílového objektu se pouze odrážejí.

Pasivní radary mohou k určování polohy objektu využívat metodu časoměrnou, směroměrnou nebo dopplerovskou, případně kombinace předešlých. Časoměrné systémy se skládají z několika přijímačů, rozmístěných v dostatečné vzdálenosti od sebe. Každý přijímač měří okamžik příchozího signálu a s využitím triangulace lze vypočítat polohu zdroje. Směroměrné

[1] Výjimku tvoří letouny využívající technologii Stealth

systémy využívají směrových vlastností (natočení) přijímacích antén pro určení směru příchodu signálu. Při větším počtu takovýchto zařízení lze z geometrického rozložení stanic určit polohu sledovaného objektu. Dopplerovské systému vyhodnocují Dopplerův posun kmitočtu v důsledku pohybu sledovaného objektu.

Všeobecnou výhodou pasivních radarů je absence vlastního vysílače, který bývá obvykle velmi drahou a energeticky náročnou součástí aktivního radaru. Důsledkem toho je, že jsou pasivní zařízení často mnohem levnější než srovnatelná zařízení aktivní. Další výhodou především ve vojenské sféře je výrazně vyšší obtížnost vypátrat a zasáhnou takovéto systémy. Nevýhodou těchto systémů je fakt, že musí spoléhat na externí, často neoptimální signály a z toho vyplívají zhoršené parametry jako přesnost a četnost zachycení signálu, nebo nutnost náročnějšího zpracování signálu.

(BEZOUŠEK, a další, 2004 stránky 3–4)

2 SEKUNDÁRNÍ RADARY

2.1 Historie

Po 2. světové válce tvořila rostoucí hustota civilního letového provozu zásadní problém. Organizace řídící letový provoz proto hledaly nástroj umožňující tento růst zvládnout. Klíčovým požadavkem byla kromě lokalizace cíle i jeho identifikace. Vzniklo několik verzí sekundárního radaru, které na přelomu 60. let 20. století vyústili v normu popisující sekundární přehledový radar (SSR). Tuto normu vytvořila a dodnes spravuje mezinárodní organizace pro civilní letectví ICAO. V zemích bývalého sovětského bloku byly souběžně využívány podobné, ale nekompatibilní systémy až do 80. let 20. století.

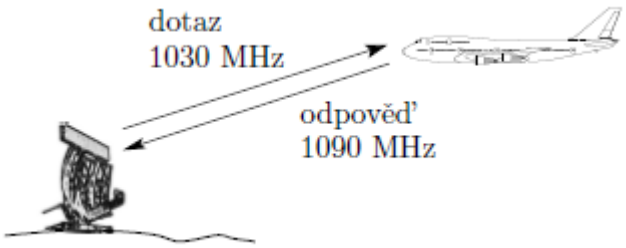
(BEZOUŠEK, a další, 2004 str. 175)

2.2 Princip

Sekundární radar je radiolokační zařízení, které vysílá dotazy a zpracovává odpovědi zaslané odpovídači, které dotaz zachytili. Sekundární radar je tedy schopen zachytit pouze objekty vybavené aktivním prvkem (odpovídačem), který je schopen dotaz zachytit, a korektním způsobem na něj odpovědět. Systém sekundárního radaru se tedy skládá ze 2 částí. 1. částí je takzvaný dotazovač (interrogator) umístěný na radarové stanici. 2. částí je potom odpovídač (transponder) umístěný na palubě letadla. Sekundární radary jsou často využívány jako doplněk radarů primárních a jejich anténa bývá umístěna nad „zrcadlem“ antény primárního radaru. Největšího rozšíření dosáhly sekundární radary v oblasti řízení letového provozu.

Zásadní výhodou sekundárních radarů je příznivější výkonová bilance, vyšší dosah (až 500 *km*), nezávislost funkce na efektivní odrazové ploše cíle, a především potom možností přenášet jakékoliv informace mezi radarem a cílem včetně informace o identitě cíle, která byla prvotním účelem zavedení sekundárních radarů.

Systém SSR pracuje na dvojici pevně určených frekvencí v pásmu L (1 *GHz* – 2 *GHz*). Dotazy vysílá dotazovač na frekvenci 1030 *MHz*. Odpovídače využívají pro vyslání odpovědi frekvenci 1090 *MHz* (viz. Obrázek 3). Signály jsou vysílány s vertikální polarizací.



Obrázek 3 – Princip SSR
[zdroj: (BEZOUŠEK, a další, 2004 str. 176)]

Palubní odpovídač neustále přijímá signál na frekvenci 1030 *MHz* a pokud v něm rozpozná dotaz, odpoví na něj signálem obsahujícím požadovaný údaj na frekvenci 1090 *MHz*. Formát dotazu i odpovědi, jejich frekvence, kódování přenášené informace a další parametry jsou definovány normou SSR.

Fakt, že všechny dotazovače sdílí pro vysílání dotazů jediný frekvenční kanál a všechny odpovídače také vysílají odpovědi na jediné frekvenci, přináší jisté problémy. Část problémů lze řešit poměrně snadno, jako například potlačení odpovědí vyvolaných dotazem jiného radaru a přijatých postranními laloky antény. Další problémy, jakým je například překrytí dvou odpovědí (garbling – viz. kapitola 2.5.2), si vyžadují poměrně náročná řešení, která nejsou vždy spolehlivá.

(BEZOUŠEK, a další, 2004 stránky 175–176)

2.3 Zjištění polohy cíle

Pro určení geografické polohy cíle pomocí sekundárního radaru je třeba znalost následujících informací.

- Radiální vzdálenost cíle od radarové stanice.
- Azimut cíle.
- Nadmořská výška cíle.
- Geografickou polohu radarové stanice.

2.3.1 Měření radiální vzdálenosti cíle

Výpočet vzdálenosti cíle vychází z naměřeného zpoždění mezi náběžnou hranou posledního vyslaného pulzu dotazu a náběžnou hranou prvního přijatého pulzu odpovědi. Změřený čas obsahuje dobu zpoždění odpovídače, která je definována normou jako 3 μs, a dvojnásobek doby šíření signálu od radaru k cíli (tj. doba šíření signálu od radaru k cíli a zpět).

Radiální vzdálenost cíle se následně určí vztahem:

$$s = \frac{(t_p - t_v) \; - \; 3\,\mu s}{2} * c$$

Kde s je radiální vzdálenost cíle, t_p je čas příjmu náběžné hrany prvního impulzu, t_v je čas vyslání náběžné hrany posledního impulzu a c je rychlost šíření signálu (rychlost světla).

Přesnost

Na chybě měření vzdálenosti se podílí zejména tolerance zpoždění odpovědi odpovídače a chyba určení času příchodu odpovědi. Přesnost zpoždění odpovídače je dána normou a je ±0,5 μs. Této toleranci odpovídá chyba vzdálenosti ±75 m. Na přesnost měření času má největší vliv odstup signálu od šumu a šířka pásma přijímaného signálu. Pro celkovou chybu měření lze očekávat dominantní vliv tolerance zpoždění odpovědi.

Rozlišovací schopnost

Dva cíle lze s jistotou rozlišit pouze tehdy, jestliže se jejich odpovědi přijaté radarem nepřekrývají. Protože odpovědi v módu A a C trvají 20,3 μs, je takto definovaná rozlišovací schopnost 3,05 km. Pokud je vzdálenost cílů od sebe menší, dochází k překrytí odpovědi (garbling – viz. kapitola 2.5.2). Existují možnosti, jak dvojici překrytých signálu vyhodnotit (tzv. degarbling), avšak nejsou zcela spolehlivé.

Ve skutečnosti není u SSR rozlišovací schopnost tak významná jako u primárních radarů. Pro rozlišení cílů se zde využívá odlišných odpovědí na dotaz identity (mód A), případně adresy dotazů a odpovědí v módu S.

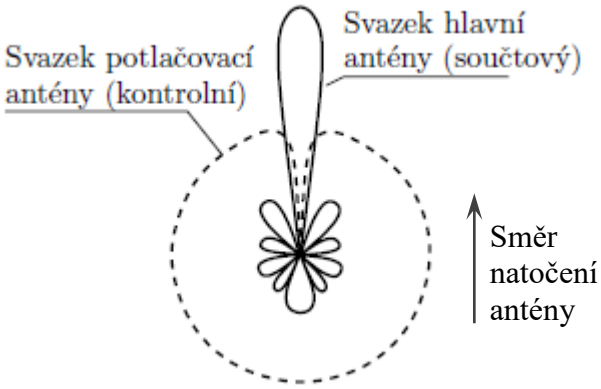
(BEZOUŠEK, a další, 2004 str. 179)

2.3.2 Měření azimutu cíle

K měření azimutu cíle se využívá směrová charakteristika (úhel natočení) antény. Odpovídač není často v hustém provozu schopen zpracovat všechny dotazy a odpovídá střídavě na dotazy několika dotazovačů. Proto je nutné využívat metod, které jsou odolné vůči chybějícím odpovědím i kolísání výkonu vysílače.

U sekundárních radarů je významný problém vysílání a příjmu postranními svazky antény. To může způsobit nejednoznačnost určení azimutu cíle a vyvolat odpovědi odpovídačů nacházejících se mimo oblast okamžitého zájmu (hlavní svazek antény). Tomuto zbytečnému zatížení celého systému se zamezí konstrukcí potlačovací antény se širokým diagramem pokrývajícím postranní svazky hlavní antény (viz. Obrázek 4). Zisk z této potlačovací antény

ve směru postranních laloků hlavní antény je vyšší než úroveň postranních laloků hlavní antény. V požadovaném směru musí být signál vysílaný hlavní anténou minimálně o 9 *dB* vyšší než úroveň signálu antény potlačovací.



Obrázek 4 – Směrová charakteristika hlavní a potlačovací antény
[zdroj: (BEZOUŠEK, a další, 2004 str. 180)]

Při vysílání dotazu je část pulzů vyslána hlavní a část potlačovací anténou. Odpovídač tak rozpozná, zda přijal dotaz z hlavního nebo postranního svazku. Na dotazy přijaté z postranního svazku neodpovídá. Porovnáním amplitudy signálů přijatých hlavní a potlačovací anténou lze eliminovat odpovědi zachycené postranními laloky hlavní antény dotazovače. To se uplatní především pro potlačení odpovědí na dotazy jiných dotazovačů.

Protože poměr amplitud signálu, přijatého hlavní a potlačovací anténou nezávisí na výkonu odpovídače, je takové určení azimutu tolerantní ke změnám výkonu odpovědi. Již ze dvou přijatých odpovědí lze teoreticky vypočítat polohu odpovídače. Tento postup však selhává, pokud je zachycena pouze jediná odpověď od daného cíle.

Alternativou pro určení azimutu je použití monopulzu, kdy je signál přijímán současně dvěma stejnými anténami a směr jeho příchodu se stanoví porovnáním signálů přijatých těmito anténami. Touto metodou lze azimut cíle určit z jediné přijaté odpovědi.

Rozlišovací schopnost

Přesnost a rozlišovací schopnost při určení azimutu cíle je u sekundárních radarů obvykle horší než u radarů primárních. Je to způsobeno tím, že sekundární radary pracují na nižší frekvenci, a mají tak při srovnatelné velikosti antén podstatně větší šířku hlavního svazku.

(BEZOUŠEK, a další, 2004 stránky 179–180)

2.3.3 Měření výšky cíle

Sekundární radar neumožňuje přímé měření výšky cíle. K určení výšky se využívá údaj z barometrického výškoměru na palubě letadla. Tento údaj je přenášen v odpovědi na dotaz módu C, nebo jako součást jedné z možných zpráv módu S. Jelikož je údaj měřen na palubě letadla, nezávisí jeho přesnost na vzdálenost letadla od radaru.

Výška letadla se pro řízení letového provozu udává nad hladinou moře, z historických důvodů ve stopách. Často se pro označení výšky využívá pojem letové hladiny (FL – Flight Level). Každá letová hladina odpovídá 100 stopám (př. FL110 = 11000 stop).

Nadmořská výška letadla je určována z barometrického tlaku podle rovnice:

$$H = (18400 + 67t_m) * \ln\left(\frac{p_0}{p}\right)$$

Kde H je nadmořská výška letadla v metrech, t_m je střední teplota masy vzduchu, p_0 je referenční tlak vzduch přepočítaný na hladinu moře a p je tlak vzduchu v bodě měření.

Střední hodnota tlaku vzduchu na mořské hladině je 101,325 kPa a podle meteorologické situace kolísá ±5 %. Těmto změnám tlaku odpovídá změna nadmořské výšky přibližně o 400 m. Na naměřený údaj výšky má dále vliv proměnná teplota.

Za účelem zvýšení přesnosti určení výšky v kritických fázích letu (vzlet / přistání) se do výšky 6000 stop (1828,8 m) používá jako reference aktuální tlak v oblasti oznámený řízením letového provozu. Při letu na vyšších letových hladinách se používá standardní referenční tlak 101,325 kPa. Důvodem tohoto postupu je minimalizace rizika kolize dvou letadel s různým nastavením referenčního tlaku. Přestože je v tomto případě chyba naměřené výšky vyšší, je tato chyba u všech letadel používajících daný referenční tlak v dané oblasti do značné míry stejná, a lze se tedy spolehnout na to, že se letadla letící v různých letových hladinách nestřetnou. Dále se tímto opatřením usnadňují dlouhé lety, kdy není nutné časté nastavování referenčního tlaku.

(BEZOUŠEK, a další, 2004 stránky 180–181)

2.3.4 Měření rychlosti cíle

Sekundární radar nemůže určit rychlost cíle podle fáze přijatého signálu, protože vysílaný a přijímaný signál jsou nekoherentní, odvozeny od dvou různých místních oscilátorů. Rychlost cíle je možné určit pouze nepřímo, výpočtem rozdílu poloh naměřených v jednotlivých

otáčkách antény. V módu S je pak možné přenášet údaj rychlosti naměřené palubními přístroji letadla jako součást jedné z možných odpovědí.

(BEZOUŠEK, a další, 2004 str. 181)

2.4 Používané signály a módy

Norma ICAO annex 10 definuje několik formátů dotazů, které odpovídají tzv. módům. Odpovídač na ně odpovídá zprávou s daty požadovanými příslušným módem (viz. Tabulka 2). Módy pro civilní využití v systému SSR se značí písmeny A, B, C, D a S. Módy pro vojenské využití se označují čísly 1–4 (připravuje se mód 5).

Tabulka 2 – Seznam módů sekundárního radaru

Mód	Popis
1	IFF (Identification, friend or foe)
2	IFF, Individuální kódy
4	IFF, Kryptomód
3 / A	číslo letu
B	nevyužívá se (docházelo k záměnám s módy A a C)
C	barometrická výška letu
D	nevyužívá se (docházelo k záměnám s módy A a C)
S	adresný mód s přenosem vyžádaných dat

(BEZOUŠEK, a další, 2004 str. 181)

Pro identifikaci vlastní/cizí (IFF) se dnes u vojenských prostředků NATO používá téměř výlučně mód 4. Žádný z nižších módů nesplňuje požadavky pro obtížnou napodobitelnost cizími prostředky. Pro zvýšení spolehlivosti systému IFF připravuje NATO nový mód 5.

(BEZOUŠEK, a další, 2004 stránky 181–182)

2.4.1 Mód 3 / A

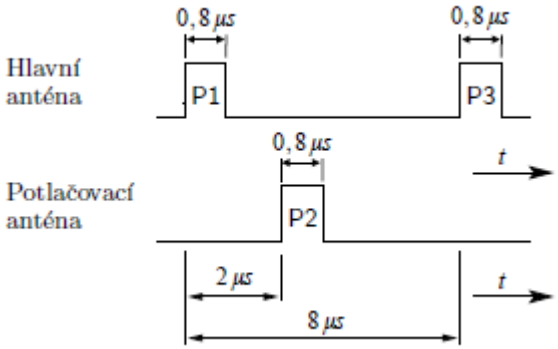
Civilní mód A a vojenský mód 3 se využívá pro zjištění čísla letu. Číslo letu je tvořeno kombinací čtyř oktalových číslic (0–7777_o, 4096 kombinací). Pro lety nad Evropou je číslo letu přiřazováno centrálně z Bruselu organizací Eurocontrol. Existuje několik kódů, které mají podle normy SSR speciální význam a nelze je přiřadit cíli (viz. Tabulka 3).

Tabulka 3 – Speciální kódy módu 3 / A

Kód (oktalově)	Význam
7500	nezákonné vměšování (únos)
7600	ztráta spojení
7700	stav nouze
7777	testovací cíl

(BEZOUŠEK, a další, 2004 str. 182)

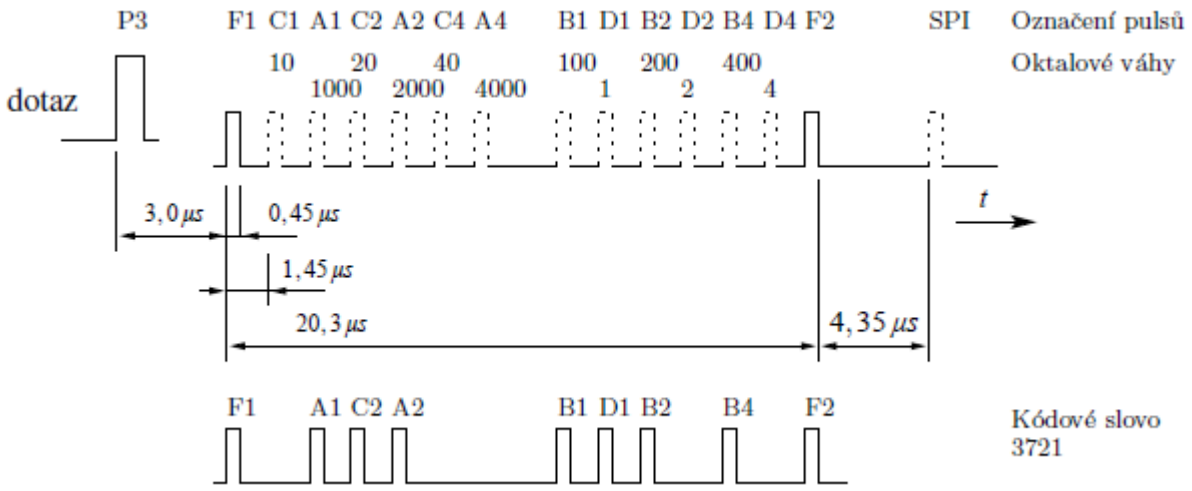
Dotaz módu 3 / A je neadresný a odpovídají na něj všechny cíle ve směru hlavního svazku antény radaru (viz. Obrázek 4). Dotaz tvoří 3 pulzy jednotné šířky 0,8 μs. Krajní pulzy (P1 a P3) jsou vysílány hlavní anténou a prostřední (P2) anténou potlačovací (viz. Obrázek 5).



Obrázek 5 – Průběh dotazu v módu 3 / A
[zdroj: (BEZOUŠEK, a další, 2004 str. 182)]

V případě, že je dotaz odpovídačem přijat z hlavního svazku hlavní antény dotazovače, mají pulzy P1 a P3 větší amplitudu něž pulz P2. Pokud je úroveň pulzů P1 a P3 alespoň 9 *dB* nad úrovní pulzu P2, odpovídač podle normy na dotaz odpovědět musí. Z tohoto důvodu je ve směru hlavního svazku dotazovače požadována úroveň vyzářeného výkony pro pulzy P1 a P3 minimálně o 9 *dB* větší. Je-li dotaz přijat s úrovní pulzu P2 vyšší než úroveň pulzů P1 a P3, potom dotaz pochází z vedlejšího laloku hlavní antény a odpovídač na něj odpovědět nesmí.

Odpověď módu A se skládá z 12 významových bitů (4 × 3 bity) s přiděleným kódem letu. Zpráva začíná krajním pulzem F1, poté následuje polovina pozic pulzů významových bitů. Následuje jedna volná pozice a zbývající významové bity. Každá zpráva je zakončena zakončovacím pulzem F2 (viz. Obrázek 6). K přenosu informace je využito amplitudové modulace, kde logické jedničce odpovídá pulz a logické nule odpovídá jeho absence.



Obrázek 6 – Formát odpovědí v módu 3 / A
[zdroj: (BEZOUŠEK, a další, 2004 str. 183)]

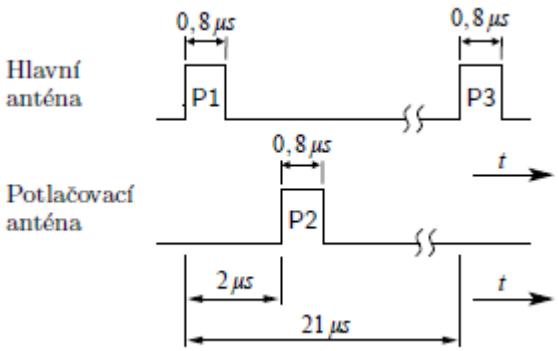
Pulz SPI (Special Pilot Identification) za přenášeným kódem je zapínán pilotem na žádost pozemního řídícího. Po jeho aktivaci je tento pulz vysílán po dobu 20 sekund za každou odpovědí módu A. Při zobrazení je pro tento cíl použita speciální značka a pozemní obsluha si tak může ověřit, s kterým letadlem hovoří.

(BEZOUŠEK, a další, 2004 stránky 182–183)

2.4.2 Mód C

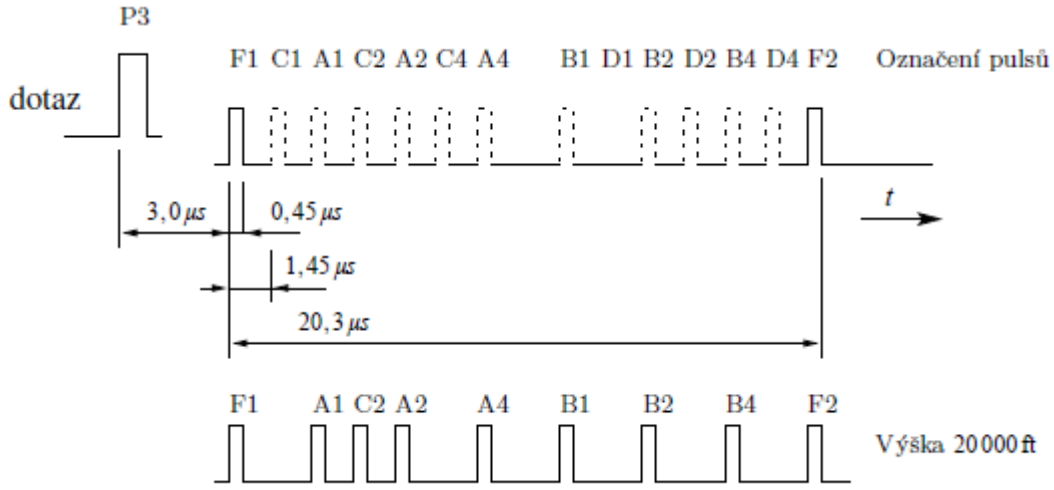
Mód C se používá pro přenášení informace o výšce letadla. Z historických důvodů se údaj o výšce přenáší ve stopách (angl. feet [*ft*], 1 $ft \approx 0{,}3048\ m$). Tento údaj je získán z barometrického výškoměru, který je umístěn na palubě letadla a je zaokrouhlen na stovky stop. Přenášená hodnota ve stovkách stop se nachází mezi -1000 *ft* (-304,8 *m*) a 121000 *ft* (36880,8 *m*).

Dotaz módu C má stejný charakter jako dotaz v módu 3 / A, liší se pouze časovým odstupem pulzů P1 a P3 (viz. Obrázek 7). Stejně jako u módu A je potlačení postranních laloků hlavní antény vysílán potlačovací anténou pulz P2.



Obrázek 7 – Průběh dotazu v módu C
[zdroj: (BEZOUŠEK, a další, 2004 str. 184)]

Odpověď módu C má stejný charakter jako odpověď módu A (viz. Obrázek 8), využívá se však pouze 11 z 12 dostupných významových bitů (bit D1 je trvale nastaven na 0).



Obrázek 8 – Průběh odpovědi v módu C
[zdroj: (BEZOUŠEK, a další, 2004 str. 184)]

Sekundární radar provádí dle poměru amplitud přijatých hlavní a potlačovací anténou potlačení odpovědí přijatých postranními laloky hlavní antény stejným způsobem jako u módu 3 / A.

Naměřená barometrická výška je kvantována s krokem 100 *ft* a poté kódována do binárního slova. Binární kód popisující danou výšku je definován normou. Tabulku všech definovaných kódů lze nalézt v dokumentu *ICAO annex 10 – volume IV* v příloze kapitoly 3 (SSR automatic pressure-altitude transmission code). (ICAO, 2007)

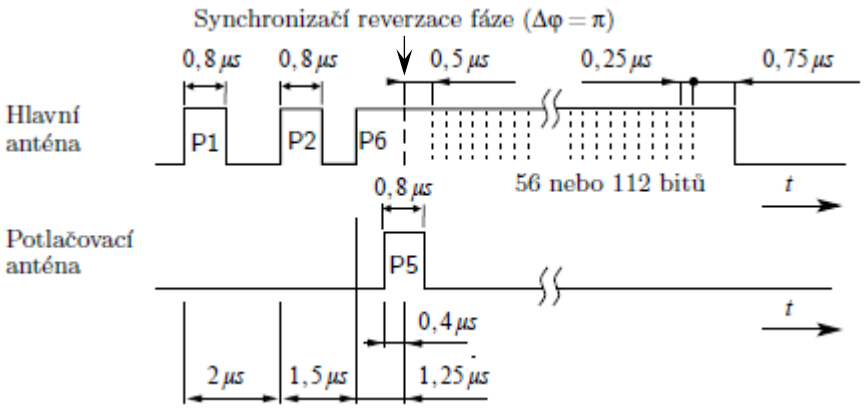
V praktickém provozu sekundární radar pravidelně střídá vysílání dotazů módu A a C (např. AACAAC…). Celková četnost vysílání dotazů je omezena normou na 450 *Hz*.

(BEZOUŠEK, a další, 2004 stránky 183–184)

2.4.3 Mód S

Dotazy v předchozích módech A a C jsou neadresné (odpovídají na ně všechny odpovídače, který daný dotaz zachytí), což vede při rostoucí hustotě letového provozu k přetěžování sdíleného kanálu. Z tohoto důvodu došlo k definování nového módu S (S = Selektivní). Mód S dále přináší mechanizmus kontrolních součtů pro kontrolu konzistence přijaté odpovědi.

Adresný dotaz v módu S má charakter datové zprávy a je vysílán na stejné frekvenci jako dotazy v módech A a C. Průběh dotazu je navržen tak, aby jej starší odpovídače, které mód S nepodporují, vyhodnotily jako neplatný a dotaz ignorovali. Průběh signálu tvoří preambule a dlouhý pulz obsahující datovou zprávu kódovanou pomocí binární fázové modulace. Adresný dotaz vyvolá odpověď odpovídače v módu S i pokud je letadlo na zemi, kdy na dotazy v módech A a C nereaguje.



Obrázek 9 – Průběh adresného dotazu v módu S
[zdroj: (BEZOUŠEK, a další, 2004 str. 184)]

Preambuli dotazu tvoří dvojice pulzů o délce 0,8 μs vzdáleny 2 μs od sebe. Tato dvojice pulzů je vysílána hlavní anténou dotazovače se stejnou amplitudou. Vzdálenost těchto pulzů odpovídá vzdálenosti mezi prvním pulzem vysílaným hlavní anténou a potlačovacím pulzem u módů A a C. Protože potlačovací pulz P2 musí mít u módů A a C ve směru hlavního svazku antény amplitudu alespoň o 9 *dB* vyšší, vyhodnotí starší odpovídače takový dotaz jako neplatný a neodpovídají na něj.

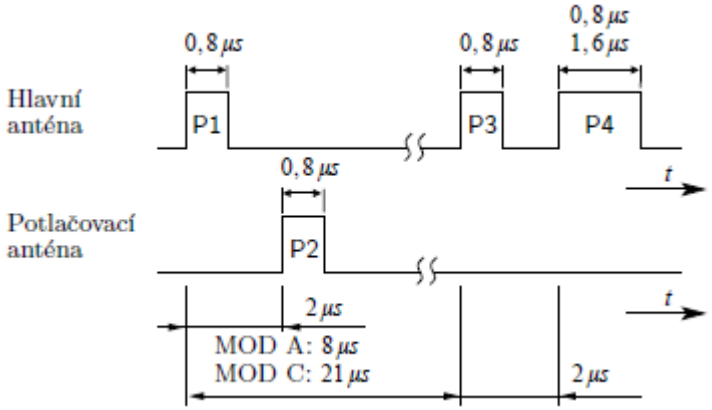
Délka pulzu s daty odpovídá velikosti přenášené zprávy (16,25 μs pro 56 bitů nebo 30,25 μs pro 112 bitů). Za účelem synchronizace odpovídače se ve vzdálenosti 1,25 μs od náběžné hrany datového pulzu P6 nachází referenční fázová změna. Po 0,5 μs od této referenční změny následují samotné bity přenášené zprávy. Logická 1 je reprezentována změnou fáze o 180°, logické 0 odpovídá absence změny.

Datová část dotazu v módu S začíná polem popisujícím typ zprávy. Další bity slouží k pro obsah samotné zprávy. Posledních 24 bitů je vždy vyhrazeno pro součet kontrolní parity (modifikovaný cyklický kód) a pro adresu dotazovaného odpovídače. Po přijetí dotazu odečte odpovídač od posledních 24 bitů svoji adresu a je-li zpráva adresována jemu, a je přijata bez chyb, je výsledek 0.

Potlačení postranních laloků hlavní antény dotazovače je v módu S zajištěno potlačovacím pulzem P5 (viz. Obrázek 9), který je vysílán potlačovací anténou (viz. Obrázek 4). Tento pulz je vysílán v době synchronizační reverzace fáze. Je-li úroveň signálu přijímaného odpovídačem z hlavní antény vyšší, potlačovací pulz se výrazně neuplatní, odpovídač se synchronizuje a na dotaz odpoví. Pokud je dotaz přijat z postranního laloku hlavní antény dotazovače, je potlačovací pulz P5 silnější a znemožní vyhodnocení synchronizační změny fáze. Na takovýto dotaz odpovídač neodpoví.

Klíčovou podmínkou pro možnost vysílání adresných dotazů módu S je získání adres odpovídačů, které se pohybují ve vzdálenosti přístrojového dosahu sekundárního radaru. Adresy je možné získat zachycením zprávy samovolně vysílané každým odpovídačem s periodou 1 sekundy, nebo zasláním tzv. všeobecného dotazu.

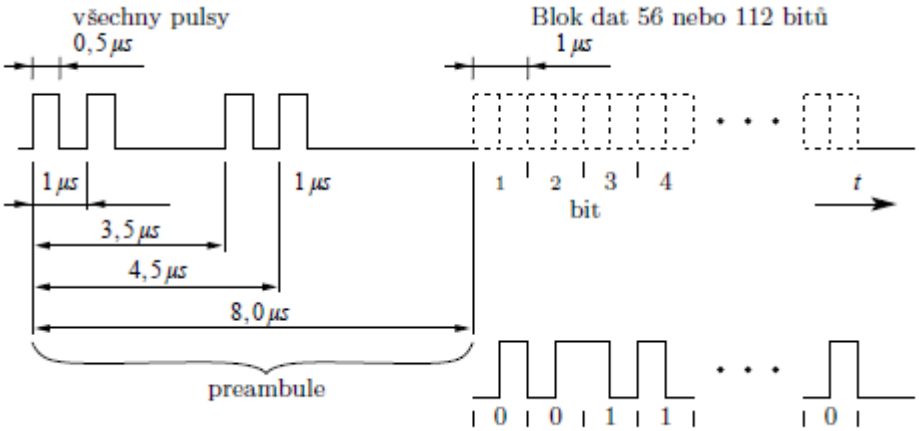
Všeobecný dotaz je speciálním typem dotazu módu S. Jedná se v podstatě o neadresný dotaz všem odpovídačům (tzv. broadcast). Na rozdíl od adresného dotazu dokáží na všeobecný dotaz odpovědět i odpovídače nepodporující mód S. Jejich odpověď potom odpovídá módu A nebo C v závislosti na podobě preambule všeobecného dotazu. Pokud odpovídač podporuje mód S, vyslání odpovědi v módu A / C se po přijetí pulzu P4 potlačí.



Obrázek 10 – Průběh všeobecného dotazu v módu S
[zdroj: (BEZOUŠEK, a další, 2004 str. 186)]

Všeobecný dotaz se dále dělí na 2 podtypy v závislosti na délce pulzu P4. Pokud je délka pulzu P4 = 1,6 μs, odpovídá odpovídač podporující mód S zprávou obsahující jeho adresu. Druhý podtyp s délkou pulzu P4 = 0,8 μs nevyvolá u odpovídače podporujícího mód S žádnou odpověď. Tento formát je určen pro systém TCAS, a využívá se také pro možnost dotázat se pouze transpondérů nepodporujících mód S.

Odpověď v módu S má stejně jako dotaz charakter datové zprávy. Pro přenos odpovědi se používá frekvence 1090 MHz stejně jako u odpovědí v módech A a C. Průběh signálu se skládá z preambule a datové zprávy. Jednotlivé bity datové zprávy jsou na rozdíl od dotazu kódovány pomocí pulzní polohové modulace (viz. Obrázek 11). Způsob potlačení odpovědí přijatých postranními laloky hlavní antény je shodný s módy A a C.



Obrázek 11 – Průběh odpovědi v módu S
[zdroj: (BEZOUŠEK, a další, 2004 str. 186)]

(BEZOUŠEK, a další, 2004 stránky 184–186)

2.5 Problémy systému

Nezbytnou podmínkou pro globální zavedení systému SSR byla snadnost jeho realizace. Obecná jednoduchost systému však v praxi přináší řadu problémů. Vylepšením systému dnes brání především jeho globální rozšíření, nutnost zasáhnou do všech odpovídačů a zajištění zpětné kompatibility.

2.5.1 Hlavní zdroje problémů

- Sdílení jediného frekvenčního kanálu (1030 *MHz* dotazy, 1090 *MHz* odpovědi).
- Relativně velká délka zprávy.
- Nemožnost určit, zda přijatá zpráva patří k vyslanému dotazu.
- Příjem dvojice odpovědí, které se překrývají (garbling).

- Nesprávné vyhodnocení hranic zpráv.
- Vliv vícecestného šíření.

(BEZOUŠEK, a další, 2004 str. 186)

2.5.2 Překrývání odpovědí (garbling)

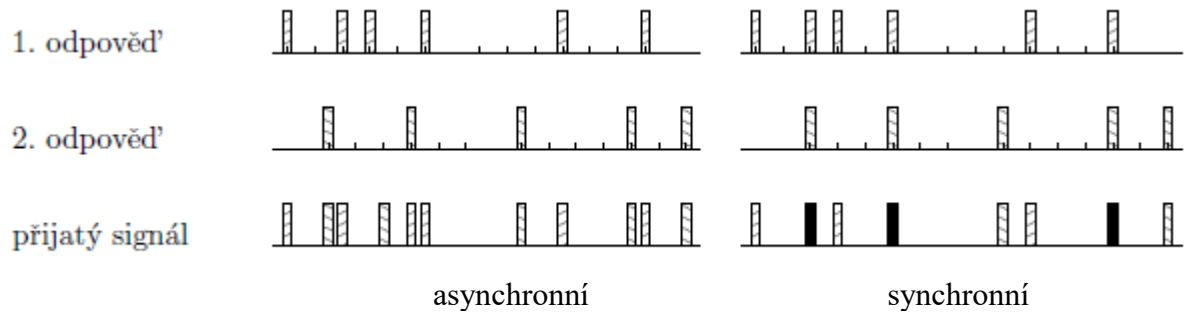
Překrývání odpovědí nastává v případě, že se ve směru hlavního svazku antény nachází dvě letadla s rozdílem šikmé vzdálenosti menší než 3,7 *km*. Délka odpovědí v módu 3 / A a C je 20,3 μs, čemuž odpovídá vzdálenost 3,7 *km*.

Asynchronní překrytí:

Nastává v případě, kdy nedojde k splynutí pulzů obou odpovědí. Vzájemnou polohu obou zpráv lze určit podle prvního a posledního přijatého pulzu. Je-li vzdálenost bezprostředně sousedících pulzů větší než šířka pulzu (0,45 μs), lze pulzy přiřadit k první nebo druhé zprávě přímo. Pokud se pulzy částečně překrývají, použijí se pro přiřazení k jedné zprávě náběžné hrany a k druhé zprávě hrany sestupné. Volba mezi náběžnou a sestupnou hranou se provádí tak, aby vzdálenost sousedních poloh vzniklých rastrů byla větší než délka pulzu.

Synchronní překrytí:

Nastává v případě, kdy je vzájemné zpoždění zpráv násobkem rozteče rastru. V tomto případě nelze přiřadit pulzy k jedné či druhé zprávě a zprávy musí být potlačeny.



Obrázek 12 – Překrytí odpovědí v módech A, 3 nebo C
[zdroj: (BEZOUŠEK, a další, 2004 str. 187)]

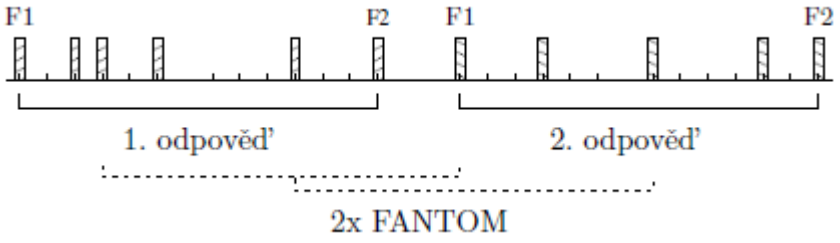
Výskyt překrytí odpovědí je velmi nepříjemný v situaci, kdy se opakuje po několik otáček radaru. Tento jev často nastává při sestupu na přistávací dráhu u velkých letišť, kdy letí velký počet letadel v jedné sestupové ose vzhledem k radaru, který je většinou umístěn v areálu letiště.

V módu S lze díky adresnosti dotazů výrazně omezit překrývaní odpovědí. V případě přijetí částečně překryté zprávy ji lze s využitím kontrolního součtu v jistém rozsahu opravit.

(BEZOUŠEK, a další, 2004 str. 187)

2.5.3 Chybné vyhodnocení částí dvou odpovědí jako platný cíl

Absence mechanizmu pro kontrolu integrity dat může v módech 3 / A a C způsobit „poskládání“ validní zprávy z pulzů vyslaných různými odpovídači. V případě vyslání dvojice nepřekrývajících se zpráv různými odpovídači, jejichž vzájemné zpoždění je násobkem vzdálenosti pulzů (1,45 μs) a mezera mezi nimi je menší než délka zprávy, mohou některé pulzy vytvořit dvojici se vzdáleností 20,3 μs. Tato dvojice pulzů se následně chová jako hraniční pulzy F1 a F2 (viz. Obrázek 6) odpovědi neexistujícího cíle. Tento neexistující cíle se často označuje jako „Fantom“.



Obrázek 13 – Validní odpovědi neexistujících cílů (Fantom)
[zdroj: (BEZOUŠEK, a další, 2004 str. 188)]

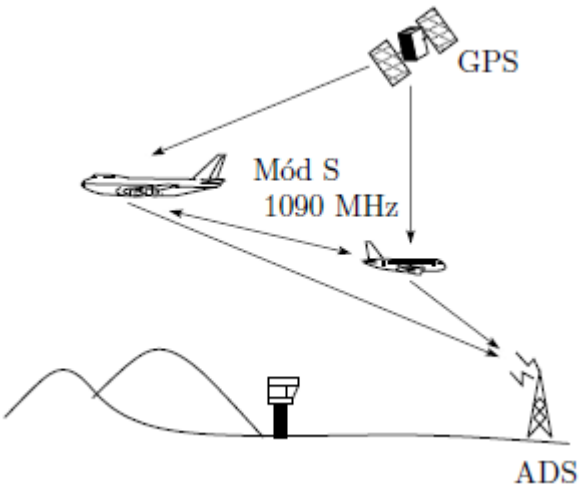
Tento problém v módu S odpadá díky přítomnosti kontrolního součtu pro ověření integrity dat. Pravděpodobnost poskládání zprávy s validním součtem kontrolní parity je zanedbatelná.

(BEZOUŠEK, a další, 2004 str. 187)

3 SYSTÉM ADS-B

ADS-B (Automatic Dependent Surveillance – Broadcast) je technologie pro jednosměrný přenos dat z letadlového odpovídače. Jedná se o všesměrové vysílání (broadcast), což znamená, že zprávu jediného odpovídače zachytí a přijme více přijímačů. Palubní část systému ADS-B (transpondér) přebírá údaje o poloze letadla z navigačních přístrojů a všesměrově je vysílá. K přenosu informace lze využít mód S sekundárního radaru nebo digitální radiovou linku (VDL). Vysílaní těchto zpráv je automatické a nezávislé na vstupu z okolí. To je zásadní rozdíl oproti systému SSR, kde transpondér pouze odpovídá na dotazy z pozemních radarových stanic.

Tyto zprávy mohou kromě pozemního řízení letového provozu přijímat i okolní letadla (viz. Obrázek 14). Letadla mohou přijaté údaje o poloze cílů porovnat s vlastní polohou a v případě zjištění možnosti kolize mohou informovat posádku nezávisle na pozemní infrastruktuře. Díky využití údajů získaných na palubě pomocí systému GNSS je poloha určena s vysokou přesností a nezávisí na vzdálenosti cíle od místa příjmu informace.



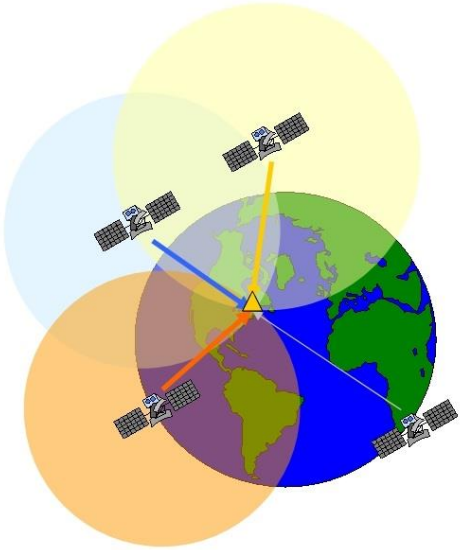
Obrázek 14 – Systém ADS-B
[zdroj: (BEZOUŠEK, a další, 2004 str. 189)]

Formát ADS-B zpráv je veřejně dostupný a popsaný v dokumentu „Minimum Operational Performance Standards of airborne equipment for ADS-B“ od EUROCAE (European Organisation for Civil Aviation Equipment). Na tento systém je zaměřen softwarový emulátor transpondéru realizovaný v praktické části diplomové práce.

(BEZOUŠEK, a další, 2004 str. 189), (EUROCAE, 2009)

3.1 GNSS

GNSS je zkratkou z anglického „Global Navigation Satellite Systém“, což v překladu znamená globální satelitní navigační systém. Satelitní systémy využívají k určení polohy cíle na povrchu, nebo blízko povrchu země metodu triangulace (viz. Obrázek 15). Vzdálenost od satelitu je určena časem, který zabere rádiovému signálu cesta od satelitu k přijímači. Vzdálenost se určí vynásobením doby putování signálu rychlostí světla (cca. 300000 *km/s*). Pro přesné určení polohy v trojrozměrném prostoru je potřeba signál alespoň ze 4 satelitů.



Obrázek 15 – Princip určení polohy pomocí GNSS
[zdroj: (DHEERAJ, © 2015)]

Mezi systémy GNSS se řadí dominantní Americký systém GPS (Global Position System), Evropský systém Galileo a Ruský GLONASS.

(LECHNER, a další, © 2000)

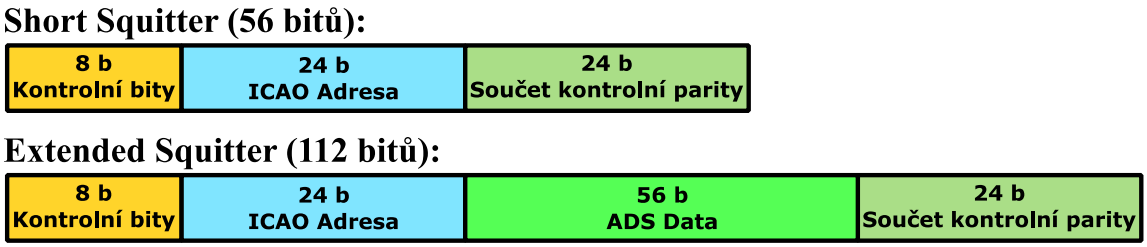
3.2 Extended Squitter (ES)

Slovo „squitter“ označuje periodické vysílání informací o letadle (tzv. „squit“) za pomoci transpondéru v módu S nezávisle na požadavku od pozemního dotazovače. Vypuštěním potřeby dotazů a následného čekání na odpověď dochází k zvýšení schopnosti systému sledovat velké množství cílů.

U základního „squitu“ v módu S zasílá transpondér pouze základní informace o identitě a barometrické výšce letadla, které musí být zpracovány pozemním systémem pro řízení letového provozu. Ten je následně musí porovnat s informacemi získanými z radaru a odvodí pozici letadla, rychlost, směr letu, rychlost stoupání / klesání atd.

Podle nového konceptu ADS-B jsou tyto údaje získány přímo na palubě letadla z navigačního systému GNSS a následně odeslány alespoň jednou za sekundu za pomoci tzv. „extended squitteru“ na frekvenci 1090 *MHz*. Tento přístup umožňuje okolním pozemním řídícím centrům a letadlům sledovat pohyb daného cíle s výrazně vyšší přesností.

„Extended squitter“ je formát schopný přenášet výrazně větší objem dat než základní „short squit“ v módu S (viz. Obrázek 16). Oba formáty je možné přenášet stejným způsobem jako odpověď na dotaz v módu S (viz. Obrázek 11).



Obrázek 16 – Short a Extended Squitter
[zdroj: vlastní]

(GARMIN, © 1996-2017)

3.3 Formát ADS-B rámce

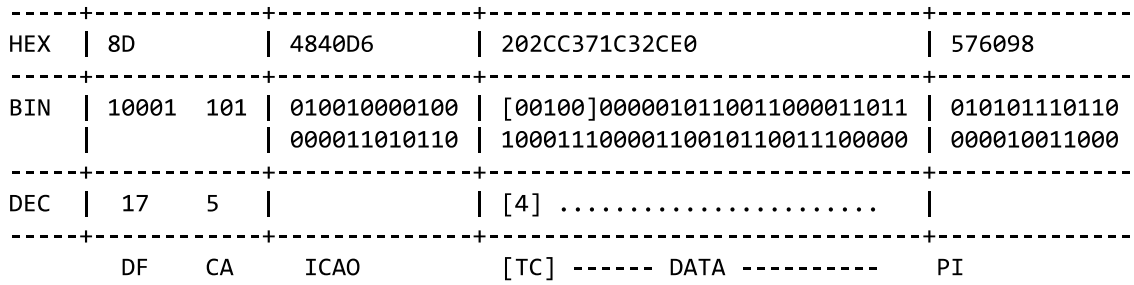
Každá ADS-B zpráva je 112 bitů dlouhá a skládá se z 5 částí popsané v následující tabulce.

Tabulka 4 – Formát ADS-B rámce

Počet bitů	Bity	Zkratka	Název
5	1–5	DF	Downlink formát (17 pro ADS-B)
3	6–8	CA	Capability
24	9–32	ICAO	ICAO adresa letadla
56	33–88	DATA	Data (payload)
	[33–37]	[TC]	Type Code
24	89–112	PI	Součet kontrolní parity (CRC)

(SUN, © 2017)

Každá ADS-B zpráva vyslaná transpondérem v módu S musí začínat Downlink Formátem 17 (10001 binárně) pro prvních 5 bitů. Následující bity 6–8 se využívají jako dodatečný identifikátor, který má různý význam pro specifické typy ADS-B zpráv. Následuje unikátní 24 bitová ICAO adresa transpondéru. Následujících 56 bitů slouží k přenosu samotného obsahu zprávy (dat). Typ zprávy určuje pole Type Code (TC), které je součástí dat (bity 33–37). Posledních 24 bitů slouží k přenosu kontrolní parity pro kontrolu integrity přijaté zprávy a případné opravy.



Obrázek 17 – Příklad ADS-B rámce
[zdroj: (SUN, © 2017)]

(SUN, © 2017)

3.3.1 ICAO adresa

Při registraci letadla u národní asociace je transpondéru přidělena unikátní 24 bitová ICAO adresa, která se stává součástí certifikátu o registraci. Tato adresa se obvykle nemění, kromě případu, kdy dochází k přesunu transpondéru do jiného letadla nebo když dojde ke změně země, ve které je letadlo registrováno. Za přidělování rozsahů adres jednotlivým zemím zodpovídá mezinárodní organizace pro civilní letectví (International Civil Aviation Organization – ICAO).

Celkem je dostupných 16 777 214 ($2^{24} - 2$) unikátních 24 bitových ICAO adres. Adresy skládající se z 24 nul a 24 jedniček jsou rezervované a přidělovat se nesmí. Adresy lze zapsat hexadecimálně, binárně, nebo oktálově (viz. Tabulka 5).

Tabulka 5 – Příklad zápisů 24 bitové ICAO adresy

Formát	Zápis
HEX	AC82EC
BIN	101011001000001011101100
OCT	53101354

(ICAO, 2011), (EUROCONTROL, © 2001-2011)

3.3.2 Type Code (TC)

Pro identifikaci informace, která je přenášena v datové části se využívá tzv. Type Code (TC). TC tvoří prvních 5 bitů v datovém bloku ADS-B rámce (bity 33–37) a je společný pro všechny typy ADS-B zpráv.

Význam jednotlivých rozsahů je popsán v následující tabulce (Tabulka 6):

Tabulka 6 – Typy ADS-B zpráv podle hodnoty Type Code (TC)

TC	Význam
1–4	Identifikace letadla
5–8	Pozice na zemi
9–18	Pozice ve vzduchu (barometrická výška)
19	Rychlost ve vzduchu
20–22	Pozice ve vzduchu (GNSS výška)
23–30	Rezervováno
31	Operační stav letadla

(SUN, © 2017)

3.3.3 Součet kontrolní parity (CRC)

Teoreticky může jakákoli kombinace 112 bitů dát dohromady zdánlivě validní ADS-B zprávu. V případě, že by při přenosu jednoho nebo více bitů došlo k chybě, mohlo by dojít k nesprávné interpretaci takovéto zprávy. Z tohoto důvodu se na konec každé zprávy přidávají tzv. paritní bity (posledních 24 bitů), s jejíchž pomocí dokáže přijímací strana rozpoznat, zdali došlo při přenosu zprávy k chybám, a příslušně na to zareagovat (např. opravit chybné bity).

3.4 Identifikace letadla (Aircraft identification)

ADS-B zpráva s TC v rozsahu 1–4 slouží k přenosu tzv. volacího znaku letadla (angl. callsign). Jeli dostupný letový plán, použije se jako volací znak identifikátor letu použitý v letovém plánu (např. číslo letu). V opačném případě se jako volací znak používá registrační značka letadla.

Obrázek 18 – Registrační značka natištěná na trupu letadla
[zdroj: en.wikipedia.org/wiki/SkyWest_Airlines]

Volací znak letadla se skládá z 8 znaků, každý zakódovaný pomocí šesti bitů na pozicích 9 až 56 v datovém bloku ADS-B zprávy (bity 41–88). Seznam všech použitelných znaků a jejich kódů je dostupný v příloze A. Následující tabulka (Tabulka 7) definuje všechna bitová pole v

datové části identifikační ADS-B zprávy, včetně jejich umístění v rámci datové části a celé ADS-B zprávy.

Tabulka 7 – Formát ADS-B zprávy pro identifikaci letadla (pouze datová část)

Bity v datové části	Bity v ADS-B zprávě	Zkratka pole	Obsah pole
1–5	33–37	TC	Type Code (1–4)
6–8	38–40	EC	Emitter Category
9–14	41–46	C1	1. znak
15–20	47–52	C2	2. znak
21–26	53–58	C3	3. znak
27–32	59–64	C4	4. znak
33–38	65–70	C5	5. znak
39–44	71–76	C6	6. znak
45–50	77–82	C7	7. znak
51–56	83–88	C8	8. znak

(SUN, © 2017)

Příklad zakódování volacího znaku KLM1023_ viz. Obrázek 19.

```
HEX:  20        | 2CC371C32CE0
BIN:  00100 000 | 001011 001100 001101 110001 110000 110010 110011 100000
DEC:      4   0 |   11     12     13     49     48     50     51     32
ZNAK:    TC  EC |   K      L      M      1      0      2      3      _
```

Obrázek 19 – Příklad ADS-B zprávy pro identifikaci cíle (datová část)
[zdroj: (SUN, © 2017)]

Kromě volacího znaku přenáší identifikační ADS-B zpráva také informace o typu cíle. Tato informace je zakódována pomocí kombinace pole TC (bity 33–37) a EC (bity 38–40). Hodnota pole TC určuje tzv. set, podle kterého lze určit na základě hodnoty pole EC typ cíle. Tabulka kódů pro pole EC (Emitter Category) je dostupná v příloze B.

V případě, kdy je informace o typu stroje nedostupná nebo nedefinovaná, nastaví se hodnota pole EC na nulu. Na hodnotě v poli TC v tomto případě nezáleží.

Volací znak přenášený v ADS-B zprávě musí mít přesně 8 znaků. V případě, kdy je požadovaný identifikátor kratší, použijí se místo chybějících znaků znaky podtržítka (_).

(EUROCAE, 2009 stránky 83–85)

3.4.1 Četnost vysílání

ADS-B zprávy pro identifikaci letadla jsou standardně transpondérem vysílány s náhodnou prodlevou z intervalu od 4,8 do 5,2 sekund mezi zprávami a s četností 1 zprávy za 5 sekund v případě, kdy transpondér vysílá ADS-B zprávy o poloze ve vzduchu nebo ADS-B zprávy o poloze na zemi s vysokou četností.

V případě, kdy transpondér vysílá ADS-B zprávy o poloze na zemi s nízkou četností, jsou ADS-B zprávy pro identifikaci letadla standardně vysílány s náhodnou prodlevou z intervalu od 9,8 do 10,2 sekund mezi zprávami a s četností 1 zprávy za 10 sekund.

V případě, kdy nejsou vysílány žádné ADS-B zprávy o poloze, vysílá standardně transpondér ADS-B zprávy pro identifikaci letadla s náhodnou prodlevou z intervalu od 4,8 do 5,2 sekund mezi zprávami a s četností 1 zprávy za 5 sekund.

(EUROCAE, 2009 str. 149)

3.5 Poloha ve vzduchu (Airborne position)

ADS-B zpráva popisující polohu cíle ve vzduchu v geodetických souřadnicích má pole TC nastaveno na hodnoty 9–18, 20–22 nebo 0 a to následujícím způsobem.

- Jeli dostupný údaj polohy ze systému GNSS a údaj výšky z barometrického výškoměru, bude pole TC nastaveno na hodnotu z rozsahu 9–18 podle získané úrovně NIC.
- Není-li dostupný údaj výšky z barometrického výškoměru, použije se údaj výšky ze systému GNSS a pole TC bude nastaveno na hodnotu z rozsahu 20–22 podle získané úrovně NIC.
- V případě, kdy není dostupný údaj polohy ze systému GNSS, nastaví se pole TC na hodnotu 0 a pole pro přenos zeměpisné šířky a délky (LAT-CPR a LON-CPR) jsou nastavena na 0. Jeli dostupný údaj výšky z barometrického výškoměru, je pole ALT nastaveno na příslušnou hodnotu, v opačném případě je nastaveno na hodnotu 0.
- Jeli dostupný údaj polohy ze systému GNSS, ale údaj výšky není dostupný z barometrického výškoměru ani systému GNSS, je pole TC nastaveno na hodnotu z rozsahu 9–18 podle získané úrovně NIC a pole ALT je nastaveno na hodnotu 0.

(EUROCAE, 2009 str. 34)

Tabulka 8 – Formát ADS-B zprávy pro polohu ve vzduchu (pouze datová část)

Bity v datové části	Bity v ADS-B zprávě	Zkratka pole	Obsah pole
1–5	33–37	TC	Type Code (0, 9–18, 20–22)
6–7	38–39	SS	Surveillance status
8	40	NIC-B	NIC Supplement-B
9–20	41–52	ALT	Výška (Altitude)
21	53	T	Time bit
22	54	F	Sudý / Lichý rámec
23–39	55–71	LAT-CPR	Zeměpisná šířka ve formátu CPR
40–56	72–88	LON-CPR	Zeměpisná délka ve formátu CPR

(SUN, © 2017)

3.5.1 Navigation Integrity Category (NIC)

Parametr navigační integrity NIC označuje míru důvěryhodnosti údaje polohy ze systému GNSS a je nezbytný pro zajištění bezpečných rozestupů mezi letadly. Na základě tohoto parametru je vytýčeno maximální horizontální ohraničení cíle (kruh o poloměru R_C, angl. Horizontal Radius of Containment) okolo získané polohy, definující bezpečnou vzdálenost odstupu pro ostatní letadla. Obecně platí, že čím vyšší NIC, tím je přesnost určení polohy cíle vyšší a hodnota maximálního poloměru horizontálního ohraničení cíle je nižší. Parametr R_C je odvozen z informací obdržených společně s polohou ze systému GNSS a je zakódován v ADS-B zprávě jako kombinace polí TC a NIC-B.

(ITT, 2009 str. 17)

Následující tabulka (Tabulka 9) definuje jednotlivé úrovně NIC a způsob jejich kódování v ADS-B zprávě o poloze ve vzduchu.

Tabulka 9 – Kódování NIC

NIC úroveň	Maximální poloměr horizontálního kruhového ohraničení cíle (R_C)	Hodnoty polí v ADS-B zprávě (dekadicky)		NIC-A[2]
		TC	NIC-B	
Údaj výšky z barometrického výškoměru:				
11	$R_c < 7{,}5\ m$	9	0	0
10	$R_c < 25\ m$	10	0	0
9	$R_c < 75\ m$	11	1	1
8	$R_c < 185{,}2\ m\ (0{,}1\ NM)$		0	0
7	$R_c < 370{,}4\ m\ (0{,}2\ NM)$	12	0	0
6	$R_c < 555{,}6\ m\ (0{,}3\ NM)$	13	1	0
	$R_c < 925\ m\ (0{,}5\ NM)$		0	0
	$R_c < 1111{,}2\ m\ (0{,}6\ NM)$		1	1
5	$R_c < 1852\ m\ (1\ NM)$	14	0	0
4	$R_c < 3{,}704\ km\ (2\ NM)$	15	0	0
3	$R_c < 7{,}408\ km\ (4\ NM)$	16	1	1
2	$R_c < 14{,}816\ km\ (8\ NM)$		0	0
1	$R_c < 37{,}04\ km\ (20\ NM)$	17	0	0
0	$R_c \geq 37{,}04\ km\ (20\ NM)$ nebo neznámé	18	0	0
Údaj výšky ze systému GNSS:				
11	$R_c < 7{,}5\ m$	20	0	0
10	$R_c < 25\ m$	21	0	0
0	$R_c \geq 25\ m$ nebo neznámé	22	0	0

(EUROCAE, 2009 str. 51)

V případě, kdy je parametr NIC nedostupný, nastaví se příslušná pole tak, aby indikovaly NIC úroveň 0.

3.5.2 Surveillance status

Dvoubitové pole „Surveillance status“ (SS) je určeno pro indikaci stavu cíle obdobně jako speciální kódy v módu A (viz. kapitola 2.4.1). Jednotlivé kódy a jejich význam jsou popsány v následující tabulce (Tabulka 10).

[2] Parametr NIC-A je přenášen odděleně v ADS-B zprávě o operačním stavu letadla (TC = 31). (SUN, © 2017)

Tabulka 10 – Kódy pro pole SS (Surveillance status)

Kód		Význam
Binárně	Dekadicky	
00	0	Žádná informace o stavu.
01	1	Trvalý stav ohrožení (Nouzový stav).
10	2	Dočasný stav ohrožení.
11	3	Speciální identifikace pozice – SPI (viz. kapitola 2.4.1, Obrázek 6).

(EUROCAE, 2009 str. 57)

3.5.3 Kódování výšky

Výška naměřená na palubě letadla je zakódována pomocí 12 bitů (bity 9–20 v datové části a 41–52 v celé zprávě). V případě, kdy je dostupný validní údaj o výšce z barometrického výškoměru, je TC nastaveno na hodnotu z rozsahu 9–18. Pro kódování barometrické výšky je validní pouze měření se standardním referenčním tlakem 101,325 kPa, nikoliv se seřízeným lokálním tlakem (viz. kapitola 2.3.3). Pokud je dostupný (validní) pouze údaj výšky získaný ze systému GNSS (GNSS výška nad elipsoidem), je TC nastaveno na hodnotu z rozsahu 20–22. Způsob kódování popsaný níže je u obou typů výšky totožný.

Bit 16 (bit 48 v celé zprávě) je takzvaný Q-bit, který určuje granularitu zakódovaného údaje výšky. Pokud je výška kódována v násobcích 25 stop je Q-bit nastaven na 1 a v případě, kdy je výška kódována v násobcích 100 stop je Q-bit nastaven na 0. Ke zakódovaní samotné výšky se tedy využívá 11 bitů, a to konkrétně datové bity 9–15 a 17–20 (41–47 a 49–52 v celé zprávě), kde bit 9 je nejvýznamnější a bit 20 nejméně významný.

Minimální kódovatelná výška odpovídající kódu 0 je -1000 stop (přibližně 304,8 m). Maximální kódovatelná výška závisí na nastavení Q-bitu a odpovídá kódu 2047. Q-bit = 1 lze využít k zakódování výšky v násobcích 25 stop s rozsahem -1000 až 50175 stop (přibližně -304,8 až 15293,34 m) a s přesností ±12,5 stop. Q-bit = 0 lze využít k zakódování výšky v násobcích 100 stop s rozsahem -1000 až 203700 stop (přibližně -304,8 až 62087,76 m) a s přesností ±50 stop. Kódování s Q-bit = 1 je tedy preferované do výšky 50175 stop díky nižší ztrátě přesnosti.

(EUROCAE, 2009 stránky 58–59)

Příklad postupu kódování výšky:

1. Z barometrického výškoměru nebo systému GNSS je získán validní údaj výšky 32000 stop.
2. $32000 < 50175$, použije se tedy kódování v násobcích 25 stop s Q-bit = 1.
3. Pomocí vztahu: $kód = (H + 1000) \div Q$, kde H je naměřená výška ve stopách a Q je zvolená granularita (25 nebo 100 stop), je získán kód pro danou výšku (v tomto případě $kód = 1320$).

4. Získaný kód je reprezentován binárně jako: `10100101000`
5. Mezi bity 7 a 8 je vložen Q-bit (v tomto případě **1**).
6. Výsledný binární kód pro danou výšku tedy vypadá takto: `10100101**1**1000`

3.5.4 Kódování polohy

Pro kódování polohy slouží bit F a pole LAT-CPR a LON-CPR (bity 22–56 v datové části a 54–88 v celé zprávě). Pole F (bit 22) slouží pro indikaci, zda se jedná o sudý, nebo lichý rámec v pořadí odvysílaných zpráv o poloze. Jeho hodnota je 0, jedná-li se o sudý rámec a 1, jedná-li se o rámec lichý. Tato informace je nezbytná při následném kódování zeměpisné šířky a délky. Následují pole LAT-CPR a LON-CPR, které slouží k zakódování zeměpisné šířky a délky ve formátu CPR popsaném v následující kapitole.

3.5.5 CPR kódování

Formát CPR (Compact Position Reporting) byl vyvinut za účelem redukce množství bitů nezbytných pro zakódování dostatečně přesné polohy v geodetických souřadnicích. Rozlišení polohy je pro ADS-B přibližně 5,1 *m* ve vzduchu a 1,3 *m* pro pozemní cíle. Obvod země je přibližně 40000 *km* což pro rozlišení polohy ve vzduchu 5,1 *m* dává přibližně 7800000 možných poloh, které je třeba vyjádřit rozlišným kódem.

Tabulka 11 – Počet bitů nezbytných k zakódování polohy s a bez CPR

Typ polohy	Bitové pole	Nezbytný počet bitů		Bity ušetřené díky CPR
		Bez CPR	S CPR	
Poloha ve vzduchu	Zeměpisná šířka	22	17	5
	Zeměpisná délka	23	17	6
	F-bit	0	1	-1
	Celkem	45	35	10
Poloha na zemi	Zeměpisná šířka	24	17	7
	Zeměpisná délka	25	17	8
	F-bit	0	1	-1
	Celkem	49	35	14

(MARSHALL, 2009 str. 2)

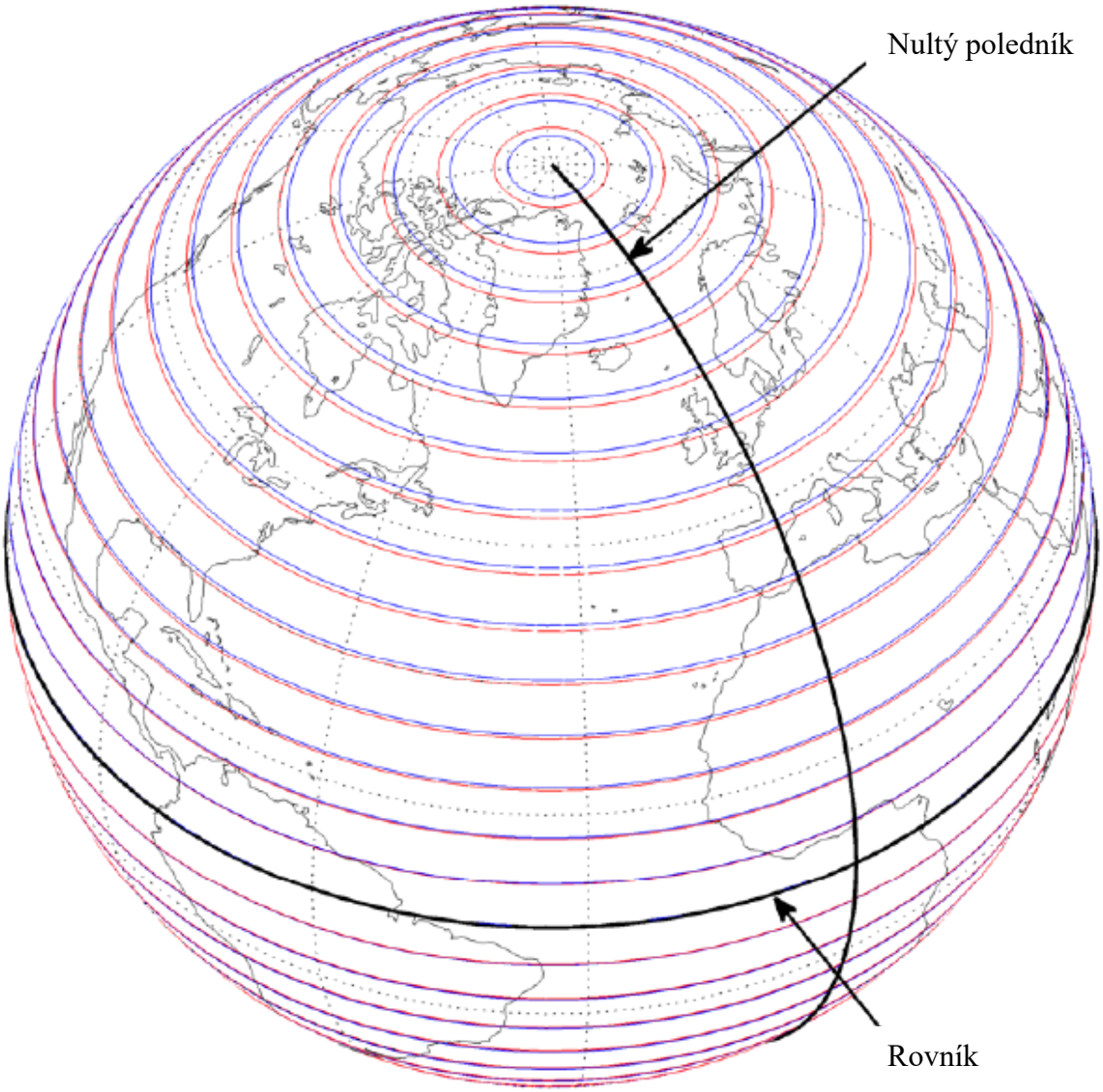
CPR ušetří 10 bitů na zprávu o poloze pro vzdušné cíle a 14 bitů na zprávu pro cíle na zemi. Zprávy o poloze jsou za normálních podmínek vysílány dvakrát za sekundu. CPR tedy ušetří 20 *b/s* pro cíle ve vzduchu a 28 *b/s* pro cíle na zemi. Ze 112 bitů v ADS-B zprávě je pouze 56 vyhrazeno pro přenos dat a z toho je prvních 5 bitů vyhrazeno pro pole TC (Type Code). Pro přenos užitečné informace tedy zbývá pouze 51 bitů na zprávu. Maximální povolený počet vyslaných ADS-B zpráv za sekundu jedním cílem je 6,2 zpráv za sekundu. To znamená, že

maximální přenosová rychlost pro užitečná data je 316 *b/s* $(6{,}2 * 51)$. CPR tedy uspoří 6–9 % přenosové kapacity pro ostatní využití.

(MARSHALL, 2009 str. 2)

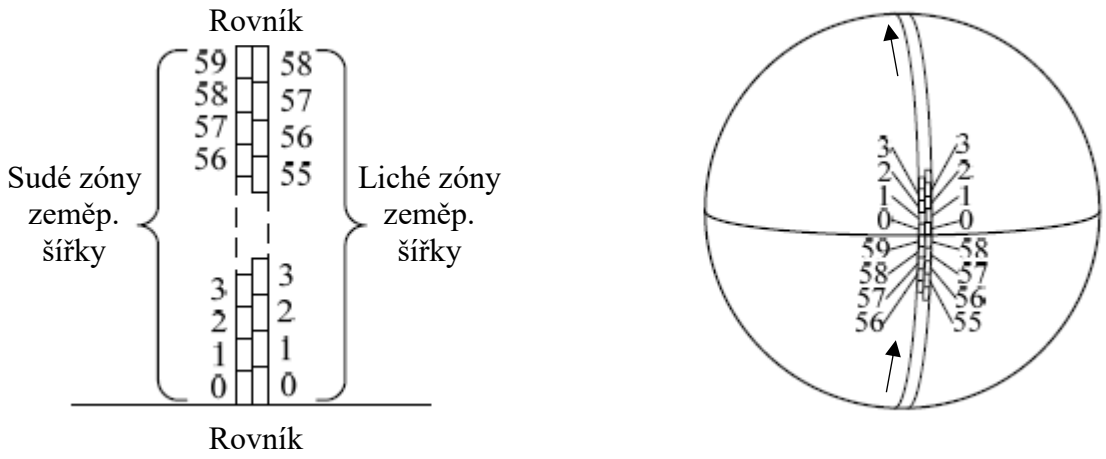
3.5.6 CPR systém souřadnic

O CPR lze uvažovat jako o souřadnicovém systému a sadě transformačních algoritmů pro převod mezi CPR a geodetickým souřadnicovým systémem. V souřadnicovém systému CPR je země rozdělena na zóny zeměpisné šířky, začínající na rovníku a směřující k pólům, a zóny zeměpisné délky, začínající nulovým poledníkem a směřující okolo země. Zóny zeměpisné šířky jsou přibližně 360 *NM* (666,72 *km*) na výšku (ve směru jih – sever) a zóny zeměpisné délky jsou přibližně 360 *NM* (666,72 *km*) na šířku (ve směru západ – východ). Množství zón zeměpisné délky se směrem od rovníku k pólům snižuje, čímž se zajistí konzistentní šířka zón. Existují dva typy zón, které se mírně odlišují svou velikostí. Sudé zóny pro sudý rámec (F-bit = 0) a liché pro lichý rámec (F-bit = 1).



Obrázek 20 – Zóny zeměpisné šířky (červená: sudé, modrá: liché)
[zdroj: (MARSHALL, 2009 str. 3)]

Každá zóna zeměpisné šířky je identifikována indexem od 0 do 59 pro sudé zóny (60 zón) a do 58 pro zóny liché (59 zón). Indexování začíná 0 pro 1. zónu severně od rovníku a pokračuje směrem k severnímu pólu a následně od jižního pólu zpět k rovníku, kde končí posledním indexem (58 nebo 59) pro 1. zónu jižně od rovníku (viz. Obrázek 21).

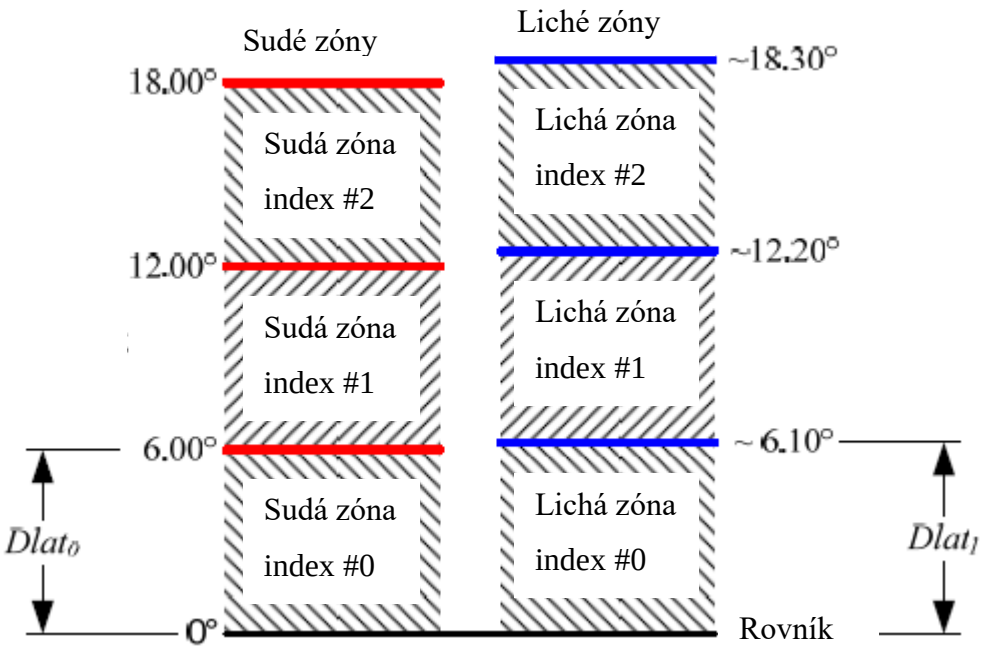


Obrázek 21 – Indexování zón zeměpisné šířky
[zdroj: (MARSHALL, 2009 str. 5)]

Zóny zeměpisné šířky jsou symetrické kolem rovníku. Výška každé zóny ve stupních je označována parametrem $Dlat_i$ a je dána vztahem:

$$Dlat_i = \frac{360°}{4 * NZ - i}$$

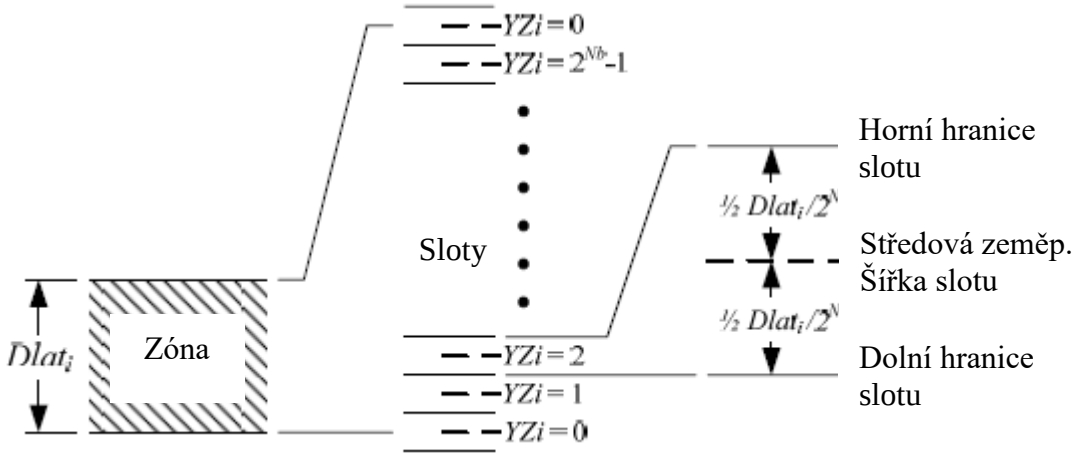
Kde $i = 0$ pro sudý typ zón a $i = 1$ pro lichý typ. Parametr NZ značí počet zón na kvadrant země který je 15. To dává výšku 6° (ve směru jih – sever) pro sudé zóny a přibližně 6,1° pro zóny liché.



Obrázek 22 – Velkosti sudých a lichých zón zeměpisné šířky
[zdroj: (MARSHALL, 2009 str. 6)]

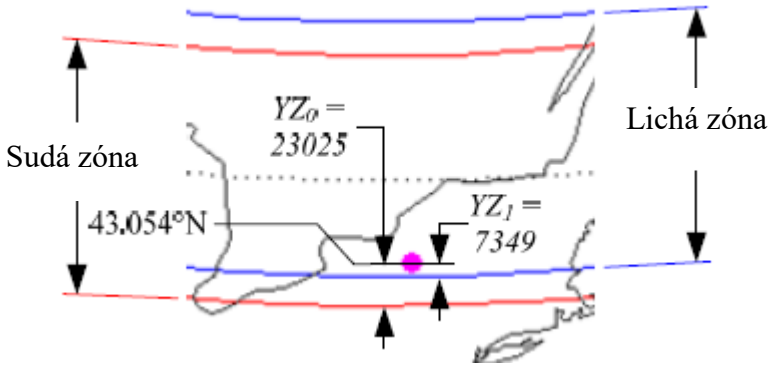
Zóny zeměpisné šířky jsou dále rozděleny do identických slotů, které reprezentují jednotlivé kódovatelné polohy. V každé zóně je celkem 2^{Nb} slotů, kde Nb je počet bitů dostupný pro zakódování zeměpisné šířky, což v případě polohy ve vzduchu ($Nb = 17$) dává 131072 slotů na zónu. Každý slot je identifikován pořadovým číslem YZ_i (indexem) vzhledem k jižnímu okraji zóny. Příznak i indikuje, vztahuje-li se hodnota YZ_i k sudé či liché zóně.

Při dekódování převede CPR algoritmus číslo slotu na zeměpisnou šířku uprostřed daného slotu. Velikost slotu tedy udává rozlišení zakódované zeměpisné šířky (přibližně 5,10–5,18 m pro polohu ve vzduchu).



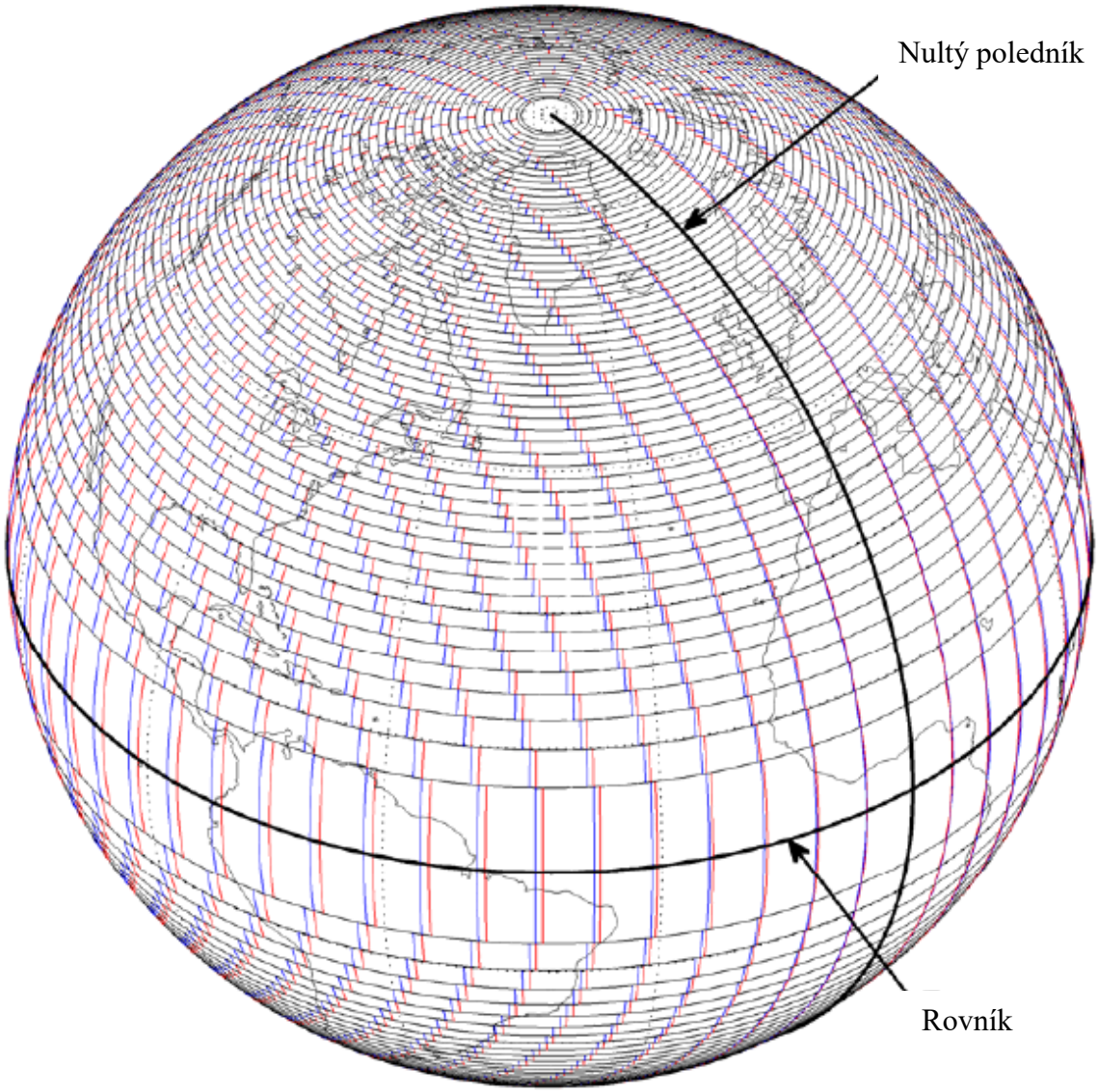
Obrázek 23 – Vztah mezi sloty a zónami
[zdroj: (MARSHALL, 2009 str. 7)]

CPR kódovací algoritmus převede zeměpisnou šířku na index slotu YZ_i, který je přenášen v poli LAT-CPR zprávy o poloze (viz. Tabulka 8). Na obrázku (Obrázek 24) je příklad indexů slotů YZ_i sudé a liché zóny pro zeměpisnou šířku 43,054°N. Bod zájmu je na obrázku zvýrazněn růžově. Index zóny, který není třeba přenášet (je na přijímací straně odvozen z indexů slotů přenesených v sudé a liché zprávě), je v tomto případě 7.



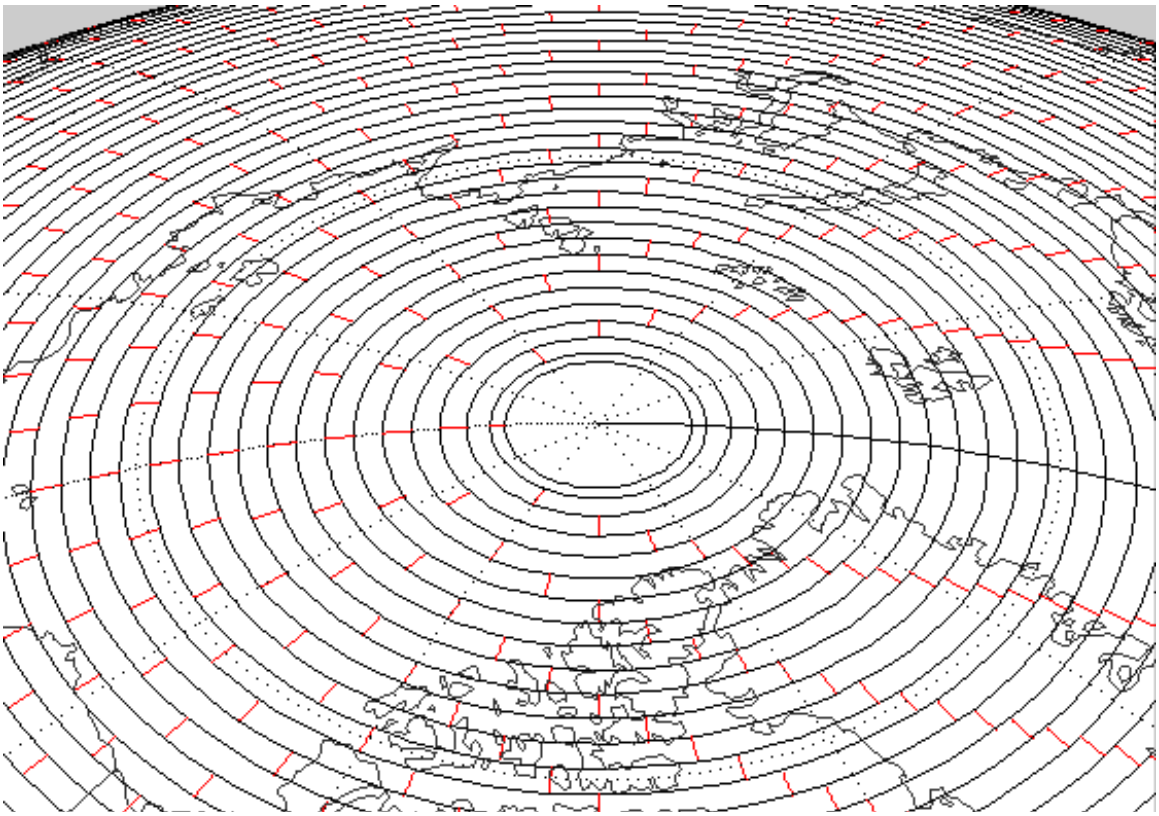
Obrázek 24 – Příklad YZ_i pro zeměpisnou šířku 43,054°N
[zdroj: (MARSHALL, 2009 str. 7)]

Indexování zón zeměpisné délky probíhá velmi obdobným způsobem. V tomto případě je index 0 první zóna východně od nultého poledníku a index se zvyšuje směrem na východ kolem celé země. Počet slotů na každou zónu a jejich interpretace je shodná se zónami zeměpisné šířky.



Obrázek 25 – Zóny zeměpisné délky (červená: sudé, modrá: liché)
[zdroj: (MARSHALL, 2009 str. 4)]

Hlavním rozdílem je celkový počet zón zeměpisné délky obíhajících zemi, který závisí na zeměpisné šířce. Je to způsobeno tím, že se směrem k pólům celkový obvod země při zachování konstantní zeměpisné šířky snižuje. Počet slotů na zónu je vždy konstantní (2^{Nb}), daný počtem bitů vyhrazených pro přenos informace o zeměpisné délce, a pro zachování konstantního rozlišení (šířky slotu v metrech) je nezbytné udržovat šířku (ve směru západ – východ) zón zeměpisné délky co nejkonstantnější.



Obrázek 26 – Redukce počtu zón zeměpisné délky směrem k pólům (pouze sudé zóny)
[zdroj: (MARSHALL, 2009 str. 8)]

Na obrázku (Obrázek 26) je vidět snižující se počet zón zeměpisné délky směrem k severnímu pólu. Parametr, určující počet sudých zón zeměpisné délky pro danou zeměpisnou šířku se označuje jako NL. Zeměpisné šířky, na kterých dochází ke změně NL, které jsou na obrázku (Obrázek 26) vyznačeny černými kružnicemi, se nazývají „NL přechodové zeměpisné šířky“. NL přechodové zeměpisné šířky nesouvisí se zónami zeměpisné šířky.

Pro výpočet parametru NL se používá tzv. „obnovená zeměpisná šířka“ $Rlat_i$. Tento parametr je získán konvertováním indexu slotu zeměpisné šířky YZ_i na zeměpisnou šířku uprostřed daného slotu (viz. Obrázek 23). Výpočet parametru NL se řídí následujícími pravidly:

- Pro $Rlat_i = 0°$ (rovník), $NL = 59$
- Pro $Rlat_i = 87°$ nebo $Rlat_i = -87°$, $NL = 2$
- Pro $Rlat_i > 87°$ nebo $Rlat_i < -87°$, $NL = 1$
- Pro všechny ostatní $Rlat_i$ je NL dán vztahem:

$$NL(Rlat_i) = floor\left(2\pi * \left[arccos\left(1 - \frac{1 - cos\left(\frac{\pi}{2 * NZ}\right)}{cos^2\left(\frac{\pi}{180°} * |Rlat_i|\right)}\right)\right]^{-1}\right)$$

Jak je již zřejmé, je hodnota parametru NL nepřímo úměrná absolutní hodnotě obnovené zeměpisné šířky ($Rlat_i$). S rostoucí absolutní hodnotou zeměpisné šířky tedy počet zón zeměpisné délky (NL) klesá.

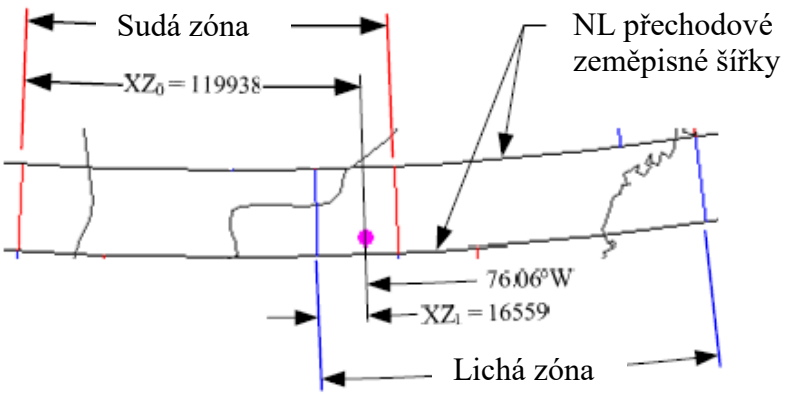
Úhlová šířka zóny zeměpisné délky závisí na hodnotě NL a jedná-li so o sudou nebo lichou zprávu. Parametr NL udává počet zón zeměpisné délky na dané zeměpisné šířce pro sudé ADS-B zprávy o poloze (sudé zóny). V případě liché zprávy je počet zón zeměpisné délky o jednu nižší, než je hodnota parametru NL a to kromě oblastí pólů, kde je počet zón zeměpisné délky pro sudé i liché zprávy roven jedné. Úhlová šířka zóny zeměpisné délky $Dlon_i$ ve stupních je potom dána vztahem:

$$Dlon_i = \begin{cases} \dfrac{360°}{NL(Rlat_i) - i}, & pro\ NL(Rlat_i) - i > 0 \\ 360° \quad , & pro\ NL(Rlat_i) - i = 0 \end{cases}$$

Kde $i = 0$ pro sudou zprávu a $i = 1$ pro zprávu lichou.

Zóny zeměpisné délky jsou rozděleny do slotů stejným způsobem jako zóny zeměpisné šířky s tím rozdílem, že indexování slotů začíná indexem 0 na západním okraji zóny a končí indexem $2^{Nb} - 1$ na východním okraji zóny. Index slotu sudé zóny se označuje jako XZ_0 a index slotu liché zóny jako XZ_1 (obecně XZ_i). CPR kódovací algoritmus převede zeměpisnou délku na index slotu XZ_i, který je přenášen v poli LON-CPR zprávy o poloze (viz. Tabulka 8). Index zóny opět není třeba přenášet, protože je na přijímací straně odvozen z indexů slotů získaných ze sudé a liché zprávy.

Na obrázku (Obrázek 27) je příklad indexů slotů XZ_i sudé a liché zóny pro zeměpisnou délku 76,06°W. Bod zájmu je na obrázku zvýrazněn růžově.



Obrázek 27 – Příklad XZ_i pro zeměpisnou délku 76,06°W
[zdroj: (MARSHALL, 2009 str. 10)]

Postup kódování polohy pomocí CPR:

1. Výpočet velikost zóny zeměpisné šířky $Dlat_i$ pomocí vztahu:

$$Dlat_i = \frac{360°}{4 * NZ - i}$$

Kde $NZ = 15$ a $i = 0$ pro sudou zprávu nebo $i = 1$ pro zprávu lichou.

2. Převod vstupní zeměpisné šířky na index slotu YZ_i pomocí vztahu:

$$YZ_i = floor\left(2^{Nb} * \frac{MOD(lat, Dlat_i)}{Dlat_i} + \frac{1}{2}\right)$$

Kde Nb je počet bitů pro přenos zakódované zeměpisné šířky (17 pro polohu ve vzduchu) a lat je zeměpisná šířka získaná ze systému GNSS v rozsahu -90° až 90°.

3. Výpočet obnovené zeměpisné šířky $Rlat_i$ pomocí vztahu:

$$Rlat_i = Dlat_i\left(\frac{YZ_i}{2^{Nb}} + floor\left(\frac{lat}{Dlat_i}\right)\right)$$

Protože v kroku 2. probíhá zaokrouhlování, hodnota $Rlat_i$ nebude zpravidla přesně odpovídat zeměpisné šířce naměřené systémem GNSS.

4. Výpočet počtu zón zeměpisné délky pro danou zeměpisnou šířku (parametr NL) pomocí vztahu:

$$NL(Rlat_i) = floor\left(2\pi * \left[arccos\left(1 - \frac{1 - cos\left(\frac{\pi}{2 * NZ}\right)}{cos^2\left(\frac{\pi}{180°} * |Rlat_i|\right)}\right)\right]^{-1}\right)$$

Parametr NL se vždy vztahuje k obnovené zeměpisné šířce $Rlat_i$.

5. Výpočet velikost zóny zeměpisné délky $Dlon_i$ pomocí vztahu:

$$Dlon_i = \begin{cases} \frac{360°}{NL(Rlat_i) - i}, & pro\ NL(Rlat_i) - i > 0 \\ 360° \quad , & pro\ NL(Rlat_i) - i = 0 \end{cases}$$

Kde $i = 0$ pro sudou zprávu nebo $i = 1$ pro zprávu lichou.

6. Převod vstupní zeměpisné délky na index slotu XZ_i pomocí vztahu:

$$XZ_i = floor\left(2^{Nb} * \frac{MOD(lon, Dlon_i)}{Dlon_i} + \frac{1}{2}\right)$$

Kde Nb je počet bitů pro přenos zakódované zeměpisné délky (17 pro polohu ve vzduchu) a lon je zeměpisná délka získaná ze systému GNSS v rozsahu -180° až 180°.

Následuje vložení získaných indexů YZ_i a XZ_i do příslušných polí v ADS-B zprávě o poloze. Pole LAT-CPR je nastaveno na hodnotu YZ_i, pole LON-CPR na hodnotu XZ_i a pole F je nastaveno na hodnotu i indikující sudou nebo lichou zprávu (viz. Tabulka 8).

(MARSHALL, 2009 stránky 2–12)

3.5.7 Četnost vysílání

ADS-B zprávy o poloze ve vzduchu jsou standardně transpondérem vysílány s náhodnou prodlevou z intervalu od 0,4 do 0,6 sekund mezi zprávami a s četností 2 zpráv za sekundu. Zprávy jsou vysílány pouze v případě, nachází-li se letadlo ve vzduchu.

(EUROCAE, 2009 str. 148)

3.6 Rychlost ve vzduchu (Airborne velocity)

ADS-B zpráva popisující rychlost cíle ve vzduchu je identifikována nastavením pole TC na hodnotu 19. Existují dva základní typy zpráv o rychlosti, které se liší údajem rychlosti, který je přenášen. Tyto dva typy jsou dále rozděleny podle přenášené rychlosti na nadzvukové a podzvukové. Jednotlivé typy zpráv o rychlosti lze rozlišit pomocí pole „Subtype“ (ST).

Tabulka 12 – Formát ADS-B zprávy pro rychlost ve vzduchu (pouze datová část)

Bity v datové části	Bity v ADS-B zprávě	Zkratka pole	Obsah pole
1–5	33–37	TC	Type Code (19)
6–8	38–40	ST	Subtype
9	41	IC	Intent Change Flag
10	42	RESV-A	Reserved-A (Nastavit na 0)
11–13	43–45	NAC	Velocity uncertainty
14	46	S_EW	Směr východ – západ (pro ST = 1, 2)
		S_HDG	Dostupnost údaje směru (pro ST = 3, 4)
15–24	47–56	V_EW	Rychlost východ – západ (pro ST = 1, 2)
		HDG	Směr / Azimut (pro ST = 3, 4)
25	57	S_NS	Směr sever – jih (pro ST = 1, 2)
		AS-T	Typ rychlosti letu (pro ST = 3, 4)
26–35	58–67	V_NS	Rychlost sever – jih (pro ST = 1, 2)
		AS	Rychlost letu (pro ST = 3, 4)
36	68	VR_SRC	Zdroj vertikální rychlosti
37	69	S_VR	Vertikální směr (Klesání / Stoupání)
38–46	70–78	VR	Vertikální rychlost (rychlost klesání nebo stoupání)
47–48	79–80	RESV-B	Reserved-B (Nastavit na 0)
49	81	S_DIFF	Rozdíl oproti barometrické výšce (příznak)
50–56	82–88	DIFF	Rozdíl oproti barometrické výšce

(EUROCAE, 2009 str. 86)

3.6.1 Subtype

Pole ST (bity 6–8 v datové části) určuje typ údaje rychlosti, který je přenášen v dané zprávě. Je-li dostupný údaj o rychlosti vůči zemskému povrchu (tzv. Ground Speed) ze systému GNSS je použit tento údaj. V případě, kdy není dostupný údaj o rychlosti vůči zemskému povrchu, lze využít údaj o rychlosti letu naměřený palubními přístroji letadla (tzv. Airspeed).

Pomocí pole ST se dále rozlišuje tzv. nadzvuková a podzvuková zpráva o rychlosti. Jedná se především o rozlišení zakódovaného údaje o rychlosti. Při použití podzvukového typu je přenesený údaj rychlosti přesnější, ale maximální rychlost, kterou lze tímto typem přenést je 1022 uzlů (1892,7 *km/h*).

V případě rychlosti vůči zemskému povrchu se na nadzvukový typ přechází v případě, kdy dílčí rychlost ve směru východ – západ nebo ve směru sever – jih překročí 1022 uzlů. Přechod zpět na podzvukový typ se provádí, klesne-li rychlost v obou směrech pod 1000 uzlů. V případě rychlosti letu (Airspeed) se na nadzvukový typ přechází, stoupne-li naměřená rychlost letu nad

1022 uzlů. Zpět na podzvukový typ se přechází v případě, kdy naměřená rychlost letu klesne pod 1000 uzlů.

Kódy pro pole ST jsou uvedeny v následující tabulce.

Tabulka 13 – Kódy pro pole ST (Subtype)

Subtype (kód)		Obsah zprávy
Binárně	Dekadicky	
000	0	Rezervováno
001	1	Rychlost vůči zemskému povrchu (Ground Speed), podzvukový typ
010	2	Rychlost vůči zemskému povrchu (Ground Speed), nadzvukový typ
011	3	Letová rychlost (Airspeed) a směr, podzvukový typ
100	4	Letová rychlost (Airspeed) a směr, nadzvukový typ
101–111	5–7	Rezervováno

(EUROCAE, 2009 str. 88)

3.6.2 Navigation Accuracy Category (NAC)

Pole NAC (bity 11–13 v datové části) udává maximální odchylku (chybu) naměřené horizontální rychlosti s pravděpodobností 95 %. Tento údaj je společně s horizontální rychlostí získán ze systému GNSS. Následující tabulka obsahuje definované úrovně NAC a příslušné maximální odchylky od naměřené rychlosti v metrech za sekundu.

Tabulka 14 – Kódy pro pole NAC

NAC (kód)		Maximální odchylka (chyba) naměřené horizontální rychlosti.
Binárně	Dekadicky	
000	0	Neznámá nebo $\geq 10\ m/s$
001	1	$< 10\ m/s$
010	2	$< 3\ m/s$
011	3	$< 1\ m/s$
100	4	$< 0{,}3\ m/s$

(EUROCAE, 2009 str. 89)

3.6.3 Kódování rychlosti vůči zemskému povrchu (Ground Speed)

V případě, kdy je dostupný údaj horizontální rychlosti ze systému GNSS, je tento údaj systémem ADS-B preferovaný. Údaj rychlosti je v tomto případě rozdělen na rychlost ve směru východ – západ a rychlost ve směru sever – jih. Pole ST je v tomto případě nastaveno na hodnotu 1 nebo 2 (viz. kapitola 3.6.1).

Pro získání dílčích rychlostí lze využít vztahu:

$$v_{NS} = \cos(HDG) * GS$$

$$v_{EW} = \sin(HDG) * GS$$

Kde v_{NS} je rychlost ve směru sever – jih, v_{EW} je rychlost ve směru východ – západ, GS je naměřená horizontální rychlost a HDG je směr letu.

Pro zakódování těchto dílčích rychlostí jsou využívána pole S_EW, V_EW, S_NS a V_NS (viz. Tabulka 12). Pole S_EW a V_EW slouží pro přenos rychlosti ve směru východ – západ a pole S_NS a V_NS slouží pro přenos rychlosti ve směru sever – jih. Údaj o směru letu není třeba přenášet, neboť je na přijímací straně odvozen z obou dílčích rychlostí. Samotné dílčí rychlosti jsou přenášeny pomocí 10 bitů jako absolutní (nezáporná) hodnota a pro indikaci směru (znaménka) se využívají bity S_EW a S_NS. Tabulka 15 definuje nastavení bitů S_EW a S_NS.

Tabulka 15 – Nastavení bitů S_EW a S_NS

Nastavení bitu S_EW nebo S_NS	Směr dílčí rychlosti	
	Bit: S_EW (rychlost východ – západ)	Bit: S_NS (rychlost sever – jih)
0	Východ	Sever
1	Západ	Jih

(EUROCAE, 2009 stránky 90–91)

Pro zakódování samotných dílčích rychlostí slouží 10 bitová pole V_EW a V_NS. Dílčí rychlosti jsou přenášeny v uzlech (1 uzel ≈ 1,852 *km/h*). Nastavení pole V_EW nebo V_NS na nulu indikuje nedostupnost údaje rychlosti v tomto směru, nikoliv rychlost 0 uzlů. Kódování rychlosti začíná od hodnoty 1, která reprezentuje rychlost 0 uzlů v daném směru. Způsob kódování dílčí rychlosti je stejný jak pro směr východ – západ tak i pro směr sever – jih, liší se však rozlišením, a to podle toho, jedná-li se o podzvukový nebo nadzvukový typ zprávy o rychlosti (viz. kapitola 3.6.1). V případě podzvukového typu reprezentuje zvýšení hodnoty pole S_EW nebo S_NS o 1 nárůst rychlosti o 1 uzel v příslušném směru. V případě nadzvukového typu pak reprezentuje zvýšení hodnoty kódu o 1 nárůst rychlosti o 4 uzly.

Následující tabulka (Tabulka 16) obsahuje příklady kódu pro vybrané dílčí rychlosti jak ve směru východ – západ tak i pro směr sever – jih, a to v podzvukové i nadzvukové variantě.

Tabulka 16 – Příklad kódů pro pole V_EW nebo V_NS

Kód pole V_EW nebo V_NS		Dílčí horizontální rychlost v daném směru v uzlech	
Binárně (10 bitů)	Dekadicky	Podzvukový typ (ST = 1)	Nadzvukový typ (ST = 2)
00 0000 0000	0	Informace o horizontální rychlosti v daném směru nedostupná.	
00 0000 0001	1	0 uzlů	
00 0000 0010	2	1 uzel	4 uzly
00 0000 0011	3	2 uzly	8 uzlů
***	***	***	***
11 1111 1110	1022	1021 uzlů	4084 uzlů
11 1111 1111	1023	> 1021,5 uzlů	> 4086 uzlů

(EUROCAE, 2009 stránky 90–91, 95–96)

3.6.4 Kódování letové rychlosti (Airspeed)

Letová rychlost naměřená palubními přístroji letadla je přenášena v případě, kdy není údaj rychlosti vůči zemskému povrchu (Ground Speed) ze systému GNSS dostupný. Pole ST je v tomto případě nastaveno na 3 nebo 4 podle toho, jedná-li se o podzvukový nebo nadzvukový typ (viz kapitola 3.6.1). V případě rychlosti letu není přenášená rychlost rozdělena na dílčí rychlosti ve směru východ – západ a sever – jih. Údaj rychlosti je přenášen tak, jak je naměřen palubními přístroji v poli AS (10 bitů). Pole HDG (10 bitů) pak slouží pro přenos informace o směru letu, která je v tomto případě nezbytná.

Dostupnost údaje o směru letu je indikována nastavením bitu S_HDG. V případě, kdy zpráva obsahuje validní informaci o směru letu je bit S_HDG nastaven na 1. V případě nedostupnosti informace o směru letu je pole S_HDG nastaveno na 0. Pro zakódování informace o směru letu pak slouží následující pole HDG. Údaj o směru letu je měřen ve stupních od 0° do 360° ve směru hodinových ručiček počínaje skutečným nebo magnetickým severem[3]. Kód pro pole HDG lze potom získat pomocí následujícího vztahu:

$$HDG = round\left(\frac{S}{360} * 1024\right)$$

Kde HDG je kód pro daný směr v desítkové soustavě, S je naměřený směr ve stupních a funkce $round(x)$ zaokrouhlí výslednou hodnotu na nejbližší celé číslo. Pro směry blížící se k 360° bude pak výsledná hodnota $HDG = 1024$ přesahovat maximální hodnotu kterou lze zakódovat pomocí 10 bitů. V tomto případě lze výsledek nahradit hodnotou 0, neboť pro směry blížící se

[3] Informace o tom, zda je jako reference využíván skutečný nebo magnetický sever je přenášena odděleně v ADS-B zprávě o operačním stavu letadla (TC = 31). (EUROCAE, 2009 str. 99)

k 360° platí, že se zároveň přibližují ke směru 0°. V následující tabulce (Tabulka 17) jsou příklady kódů pro vybrané směry.

Tabulka 17 – Příklad kódů pro pole HDG

Kód pole HDG		Směr letu ve stupních
Binárně (10 bitů)	Dekadicky	
00 0000 0000	0	0°
00 0000 0001	1	0,3515625°
00 0000 0010	2	0,703125°
00 0000 0011	3	1,0546875°
***	***	***
01 1111 1111	511	179,6484375°
10 0000 0000	512	180°
10 0000 0001	513	180,3515625°
10 0000 0010	514	180,703125°
***	***	***
11 1111 1110	1022	359,296875°
11 1111 1111	1023	359,6484375°

(EUROCAE, 2009 str. 99)

Bit AS-T slouží pro indikaci typu letové rychlosti která je ve zprávě přenášena. Jedná-li se o indikovanou letovou rychlost (IAS) je tento bit nastaven na 0. V případě, kdy je přenášena skutečná rychlost letu (TAS) je tento bit nastaven na 1.

Pro přenos samotné rychlosti pak slouží pole AS (10 bitů). Kódování rychlosti probíhá stejným způsobem jako kódování dílčích rychlostí v případě, kdy je dostupný údaj rychlosti vůči zemskému povrchu (viz. kapitola 3.6.3). Následující tabulka (Tabulka 18) obsahuje příklady kódů pro vybrané rychlosti v podzvukové i nadzvukové variantě.

Tabulka 18 – Příklad kódů pro pole AS

Kód pole AS		Letová rychlost v uzlech	
Binárně (10 bitů)	Dekadicky	Podzvukový typ (ST = 3)	Nadzvukový typ (ST = 4)
00 0000 0000	0	Informace o letové rychlosti nedostupná.	
00 0000 0001	1	0 uzlů	
00 0000 0010	2	1 uzel	4 uzly
00 0000 0011	3	2 uzly	8 uzlů
***	***	***	***
11 1111 1110	1022	1021 uzlů	4084 uzlů
11 1111 1111	1023	> 1021,5 uzlů	> 4086 uzlů

(EUROCAE, 2009 stránky 98–103)

3.6.5 Kódování vertikální rychlosti (rychlost stoupání / klesání)

Pro přenos informace o rychlosti stoupání nebo klesání jsou vyhrazena pole VR_SRC, S_VR a VR (viz. Tabulka 12). Bit VR-SRC indikuje zdroj údaje o rychlosti stoupání, respektive klesání. Je-li zdrojem údaje vertikální rychlosti systém GNSS, je tento bit nastaven na 0. Pochází-li údaj vertikální rychlosti z barometrického výškoměru, je tento bit nastaven na 1.

Bit S_VR indikuje směr vertikálního pohybu, tedy jedná-li se o klesání nebo stoupání. V případě stoupání je tento bit nastaven na 0, v opačném případě je nastaven na 1.

Samotná vertikální rychlost je přenášena v poli VR (9 bitů). Vertikální rychlost je přenášena ve stopách za minutu a rozlišení zakódované vertikální rychlosti je 64 stop za minutu (19,5 *m/s*). Kódování vertikální rychlosti začíná od hodnoty pole VR = 1, která reprezentuje vertikální rychlost 0 stop za minutu. Nastavení pole VR na nulu pak znamená, že je informace o vertikální rychlosti nedostupná. Kód pole VR pro danou vertikální rychlost lze získat aplikací následujícího vztahu:

$$VR = round\left(\frac{|VR_s|}{64}\right) + 1$$

Kde VR je kód pro danou vertikální rychlost v desítkové soustavě, $|VR_s|$ je absolutní hodnota naměřené vertikální rychlosti ve stopách za minutu a funkce $round(x)$ zaokrouhlí výslednou hodnotu na nejbližší celé číslo. Absolutní hodnota je zde neboť informace o tom, zda se jedná o klesání nebo stoupání, je přenášena samostatně pomocí bitu S_VR. Výslednou hodnotu VR je jako poslední krok nutno omezit na maximální hodnotu 511 (9 bitů) pro případ, kdy vertikální rychlost překročí maximální přenositelnou rychlost 32608 stop za minutu. Následující tabulka (Tabulka 19) obsahuje příklady kódů pro vybrané vertikální rychlosti.

Tabulka 19 – Příklad kódů pro pole VR

Kód pole VR		Absolutní vertikální rychlost ve stopách za minutu
Binárně (9 bitů)	Dekadicky	
0 0000 0000	0	Informace o vertikální rychlosti nedostupná.
0 0000 0001	1	0 stop za minutu
0 0000 0010	2	64 stop za minutu
0 0000 0011	3	128 stop za minutu
***	***	***
1 1111 1110	510	32576 stop za minutu
1 1111 1111	511	> 32608 stop za minutu

(EUROCAE, 2009 stránky 91–92)

3.6.6 Rozdíl oproti barometrické výšce (pole S_DIFF a DIFF)

Pole S_DIFF a DIFF slouží pro přenos informace o rozdílu mezi geometrickou výškou nad elipsoidem (HAE) naměřenou systémem GNSS a výškou naměřenou barometrickým výškoměrem na palubě letadla v případě, kdy jsou tyto oba údaje dostupné a validní. Pokud jeden z údajů nebo oba údaje o výšce nejsou dostupné, je pole DIFF nastaveno na 0.

Bit S_DIFF indikuje který z naměřených údajů výšky je vyšší. V případě, kdy je údaj geometrické výšky naměřený pomocí GNSS vyšší než údaj výšky naměřený barometrickým výškoměrem, je tento bit nastaven na 0. V opačném případě je bit S_DIFF nastaven na 1.

Pole DIFF (7 bitů) poté reprezentuje absolutní hodnotu rozdílu mezi geometrickou a barometrickou výškou. Rozdíl výšky je přenášen ve stopách s rozlišením 25 stop (7,62 m) a začíná hodnotou pole DIFF = 1. Nastavení pole DIFF na hodnotu 0 značí nedostupnost tohoto údaje. Kód pro pole DIFF lze za předpokladu dostupnosti obou údajů výšky získat aplikací následujícího vztahu:

$$DIFF = round\left(\frac{|H_{GNSS} - H_{Bar}|}{25}\right) + 1$$

Kde $DIFF$ je kód pro daný rozdíl výšek v desítkové soustavě, H_{GNSS} je údaj výšky ze systému GNSS ve stopách, H_{Bar} je údaj výšky z barometrického výškoměru ve stopách a funkce $round(x)$ zaokrouhlí výslednou hodnotu na nejbližší celé číslo. Výslednou hodnotu $DIFF$ je jako poslední krok nutné omezit na maximální hodnotu 127 (7 bitů) pro případ, kdy rozdíl výšek překročí maximální přenositelnou hodnotu 3137,5 stop. Následující tabulka (Tabulka 20) obsahuje příklady kódů pro vybrané rozdíly výšek.

Tabulka 20 – Příklad kódů pro pole DIFF

Kód pole DIFF		Absolutní rozdíl mezi výškou naměřenou systémem GNSS a výškou naměřenou barometrickým výškoměrem ve stopách
Binárně (7 bitů)	Dekadicky	
000 0000	0	Informace o rozdílu výšek není dostupná.
000 0001	1	0 stop
000 0010	2	25 stop
000 0011	3	50 stop
***	***	***
111 1110	126	3125 stop
111 1111	127	> 3137,5 stop

(EUROCAE, 2009 str. 93)

3.6.7 Četnost vysílání

ADS-B zprávy o rychlosti ve vzduchu jsou standardně transpondérem vysílány s náhodnou prodlevou z intervalu od 0,4 do 0,6 sekund mezi zprávami a s četností 2 zpráv za sekundu. Zprávy jsou vysílány pouze v případě, nachází-li se letadlo ve vzduchu.

(EUROCAE, 2009 str. 149)

3.7 Poloha na zemi (Surface Position)

ADS-B zpráva popisující polohu cíle na zemi v geodetických souřadnicích má pole TC nastaveno na hodnoty 5–8 nebo 0 a to následujícím způsobem.

- Jeli dostupný údaj polohy ze systému GNSS je pole TC nastaveno na hodnotu z intervalu 5–8 podle získané hodnoty NIC.
- V případě, kdy není dostupný údaj polohy ze systému GNSS, nastaví se pole TC na hodnotu 0 a pole pro přenos zeměpisné šířky a délky (LAT-CPR a LON-CPR) jsou nastavena na 0.

Tabulka 21 – Formát ADS-B zprávy pro polohu na zemi (pouze datová část)

Bity v datové části	Bity v ADS-B zprávě	Zkratka pole	Obsah pole
1–5	33–37	TC	Type Code (0, 5–8)
6–12	38–44	MOV	Informace o pohybu (Movement)
13	45	S_HDG	Dostupnost údaje o směru
14–20	46–52	HDG	Směr / Azimut
21	53	T	Time bit
22	54	F	Sudý / Lichý rámec
23–39	55–71	LAT-CPR	Zeměpisná šířka ve formátu CPR
40–56	72–88	LON-CPR	Zeměpisná délka ve formátu CPR

(EUROCAE, 2009 stránky 69–70)

3.7.1 Navigation Integrity Category (NIC)

Parametr navigační integrity NIC funguje stejným způsobem jako pro zprávy o poloze ve vzduchu (viz. kapitola 3.5.1), liší se však definovanými úrovněmi R_C a jejich kódováním.

Následující tabulka (Tabulka 22) definuje jednotlivé úrovně NIC a způsob jejich kódování v ADS-B zprávě o poloze na zemi.

Tabulka 22 – Kódování NIC (ADS-B zpráva o poloze na zemi)

NIC úroveň	Maximální poloměr horizontálního kruhového ohraničení cíle (R_C)	Nastavení pole TC	NIC-A[4]	NIC-C[5]
11	$R_c < 7{,}5\ m$	5	0	0
10	$R_c < 25\ m$	6	0	0
9	$R_c < 75\ m$	7	1	0
8	$R_c < 185{,}2\ m\ (0{,}1\ NM)$		0	0
7	$R_c < 370{,}4\ m\ (0{,}2\ NM)$	8	1	1
6	$R_c < 555{,}6\ m\ (0{,}3\ NM)$		1	0
	$R_c < 1111{,}2\ m\ (0{,}6\ NM)$		0	1
0	$R_c \geq 1111{,}2\ m\ (0{,}6\ NM)$ nebo neznámé		0	0

(EUROCAE, 2009 str. 51)

3.7.2 Informace o pohybu (Movement)

Pole MOV (bity 6–12 v datové části) slouží především pro zakódování informace o rychlosti pohybu na zemi. Kódování rychlosti probíhá obdobným způsobem jako u rychlosti ve vzduchu (viz. kapitoly 3.6.3 a 3.6.4) s tím rozdílem, že rozlišení (kvantizační krok) není konstantní, ale s rostoucí rychlostí rozlišení klesá (kvantizační krok roste).

Tabulka 23 – Rozsahy kódů pro pole MOV a jejich význam

Kód (dekadicky)	Význam (rozsah rychlostí)	Rozlišení[6] (kvantizační krok)
0	Informace o pohybu není dostupná	–
1	Letadlo stojí (**GS** = 0 *km/h*)	–
2	0 *km/h* < **GS** ≤ 0,2315 *km/h* (0,125 uzlu)	–
3–8	0,2315 *km/h* (0,125 uzlu) < **GS** ≤ 1,852 *km/h* (1 uzel)	0,2700833 *km/h*
9–12	1,852 *km/h* (1 uzel) < **GS** ≤ 3,704 *km/h* (2 uzly)	0,463 *km/h* (0,25 uzlu)
13–38	3,704 *km/h* (2 uzly) < **GS** ≤ 27,78 *km/h* (15 uzlů)	0,926 *km/h* (0,50 uzlu)
39–93	27,78 *km/h* (15 uzlů) < **GS** ≤ 129,64 *km/h* (70 uzlů)	1,852 *km/h* (1 uzel)
94–108	129,64 *km/h* (70 uzlů) < **GS** ≤ 185,2 *km/h* (100 uzlů)	3,704 *km/h* (2 uzly)
109–123	185,2 *km/h* (100 uzlů) < **GS** ≤ 324,1 *km/h* (175 uzlů)	9,26 *km/h* (5 uzlů)
124[7]	324,1 *km/h* (175 uzlů) < **GS**	–
125–127	Rezervováno	–

GS = rychlost vůči zemskému povrchu. (EUROCAE, 2009 str. 72)

[4] Parametr NIC-A je přenášen odděleně v ADS-B zprávě o operačním stavu letadla (TC = 31). (SUN, © 2017)
[5] Parametr NIC-C je přenášen odděleně v ADS-B zprávě o operačním stavu letadla (TC = 31). (SUN, © 2017)
[6] Změna rychlosti vůči zemskému povrchu, která odpovídá změně kódu o 1 v daném rozsahu rychlostí.
[7] Použije se vždy když pozemní rychlost přesahuje 324,1 *km/h*.

3.7.3 Kódování směru

Pro zakódování informace o směru (pohybu) letadla slouží pole HDG (bity 14–20 v datové části) a bit S_HDG (bit 13 v datové části). Dostupnost údaje o směru je indikována nastavením bitu S_HDG. V případě, kdy zpráva obsahuje validní informaci o směru je bit S_HDG nastaven na 1. V případě nedostupnosti validní informace o směru je pole S_HDG nastaveno na 0.

Pro zakódování informace o směru pak slouží následující pole HDG. Údaj o směru je měřen ve stupních od 0° do 360° ve směru hodinových ručiček počínaje skutečným nebo magnetickým severem[8]. Kódování směru probíhá obdobným způsobem jako u kódování letové rychlosti (viz. kapitola 3.6.4). Jediným rozdílem je nižší počet bitů dostupných pro přenesení informace o směru, z čehož plyne nižší rozlišení přenesené informace. Kód pro daný směr v tomto případě získáme pomocí vztahu:

$$HDG = round\left(\frac{S}{360} * 128\right)$$

Kde HDG je kód pro daný směr v desítkové soustavě, S je naměřený směr ve stupních a funkce $round(x)$ zaokrouhlí výslednou hodnotu na nejbližší celé číslo. Pro směry blížící se k 360° bude pak výsledná hodnota $HDG = 128$ přesahovat maximální hodnotu kterou lze zakódovat pomocí 7 bitů. V tomto případě lze výsledek nahradit hodnotou 0, neboť pro směry blížící se k 360° platí, že se zároveň přibližují ke směru 0°. V následující tabulce (Tabulka 24) jsou příklady kódů pro vybrané směry.

Tabulka 24 – Příklad kódů pro pole HDG (ADS-B zpráva o poloze na zemi)

Kód pole HDG		Směr ve stupních
Binárně (7 bitů)	Dekadicky	
000 0000	0	0°
000 0001	1	2,8125°
000 0010	2	5,6250°
***	***	***
011 1111	63	177,1875°
100 0000	64	180°
100 0001	65	182,8125°
***	***	***
111 1111	127	357,1875°

(EUROCAE, 2009 str. 73)

[8] Informace o tom, zda je jako reference využíván skutečný nebo magnetický sever je přenášena odděleně v ADS-B zprávě o operačním stavu letadla (TC = 31). (EUROCAE, 2009 str. 74)

3.7.4 Kódování polohy na zemi

Pro kódování polohy na zemi slouží bit F a pole LAT-CPR a LON-CPR (bity 22–56 v datové části a 54–88 v celé zprávě). Kódování polohy na zemi probíhá stejným způsobem jako kódování polohy ve vzduchu (viz. kapitoly 3.5.4, 3.5.5 a 3.5.6).

3.7.5 Četnost vysílání

ADS-B zprávy o poloze na zemi jsou vysílány ve 2 režimech. V režimu s vysokou četností vysílání jsou standardně transpondérem vysílány s náhodnou prodlevou z intervalu od 0,4 do 0,6 sekund mezi zprávami a s četností 2 zpráv za sekundu. V režimu s nízkou četností vysílání jsou standardně transpondérem vysílány s náhodnou prodlevou z intervalu od 4,8 do 5,2 sekund mezi zprávami a s četností 1 zprávy za 5 sekund.

Do režimu s nízkou četností vysílání přechází transpondér v případě, kdy se poloha letadla za posledních 30 sekund nezměnila o více jak 10 metrů. Při přechodu do režimu s nízkou frekvencí vysílání je aktuální poloha uložena do paměti. Zpět do režimu s vysokou frekvencí vysílání přechází transpondér v případě, kdy se aktuální poloha liší o více jak 10 metrů od polohy uložené při přechodu do režimu s nízkou frekvencí vysílání.

V případě nedostupnosti informace o poloze ze systému GNSS je jako výchozí použit režim s vysokou četností vysílání.

(EUROCAE, 2009 str. 148)

3.8 Útoky na systém ADS-B

Protokol ADS-B postrádá jakékoliv kryptografické a autentizační mechanismy pro zajištění důvěrnosti vysílaných zpráv. Volně dostupné specifikace systému ADS-B společně s rádiovou technologií založenou na všesměrovém vysílání činí tento systém velmi zranitelný vůči potenciálním útočníkům. ADS-B zprávy mohou být velmi jednoduše zachyceny, modifikovány, nebo smazány útočníkem s patřičným rádiovým vybavením. Útoky na systém ADS-B se dělí na 2 základní typy. Na útoky pasivní a aktivní.

3.8.1 Pasivní útoky

Jelikož jsou ADS-B zprávy transpondérem vysílány všesměrově a v nezašifrované podobě, mohou být velice snadno zachyceny a dekódovány kýmkoliv s rádiovým přijímačem naslouchajícím na frekvenci 1090 *MHz*. Tímto způsobem může útočník získat unikátní identifikátory letadel v dosahu společně s jejich přesnou polohou. Pasivní útok obvykle nepředstavuje žádnou přímou hrozbu pro bezpečnost letového provozu a existují i služby na

internetu poskytující veřejnosti vizualizovaná ADS-B data, jakou je například server *flightradar24.com*. Pasivní útok však bývá prvním krokem pro zahájení útoku aktivního, jelikož spousta aktivních útoků závisí na znalosti informací získaných pasivním odposloucháváním.

(SCHÄFER, a další, 2013)

3.8.2 Aktivní útoky

Aktivní útoky mohou znamenat vážné ohrožení bezpečnosti letového provozu. Při aktivním útoku se útočník snaží narušit fungování systému ADS-B vysíláním vlastních falešných zpráv, nebo smazáním, popřípadě modifikací zpráv existujících.

Vysílání falešných zpráv

Pro vysílání falešných zpráv potřebuje útočník vysílač schopný generovat korektně modulovaný signál ve validním formátu ADS-B zpráv (viz. kapitoly 3.4–3.7). V případě ADS-B přijímačů využívajících všesměrové antény mohou být falešné zprávy vysílány z jakéhokoliv umístění, jelikož přijímač není schopen rozpoznat falešné zprávy o poloze na základě úhlu natočení antény. I v případě směrových antén mohou být falešné zprávy o poloze zachyceny díky výrazným postranním lalokům, které jsou pro tyto antény typické. Tento postup útoku je nejjednodušší, jelikož na rozdíl od následujících postupů nevyžaduje žádnou synchronizaci a přesné časování vysílaného signálu.

Mazání zpráv

Tento typ útoku lze realizovat dvěma postupy. Prvním postupem je destruktivní rušení, při kterém se útočník snaží eliminovat legitimní signál tím, že bude vysílat inverzní signál legitimního. Jelikož signál zachycený přijímačem je superpozicí těchto dvou signálů, bude výsledný signál smazaný nebo silně narušený. Tento typ rušení vyžaduje velmi přesné časování a synchronizaci, která je těžko dosažitelná pro pohybující se letadla.

Druhým, výrazně jednodušším postupem je konstruktivní rušení. Při tomto postupu se útočník snaží zvýšit počet chybných bitů u legitimních ADS-B zpráv. Součet kontrolní parity, který je součástí každé ADS-B zprávy, umožnuje opravit maximálně 5 chybných bitů. Zprávy s větším počtem chybných bitů již nejsou opravitelné a jsou přijímačem zahozeny. Podmínkou pro konstruktivní rušení pozemní stanice *G* je následující vztah:

$$\frac{P_{PG}}{P_{AG} + N_G} < \beta$$

Kde P_{PG} je výkon legitimního signálu zachycený pozemní stanicí G, P_{AG} je zachycený signál útočníka pozemní stanicí G, N_G je úroveň šumového pozadí na pozemní stanici G a β je minimální odstup signálu od šumu nezbytný pro korektní dekódování zpráv.

Výše popsaný postup způsobí smazání všech zpráv v dosahu vysílače útočníka. S přesným časováním se však útočník dokáže zaměřit i na konkrétní zprávy od konkrétního cíle.

Modifikace zpráv

Cílem tohoto útoku je modifikace legitimních ADS-B zpráv v průběhu jejich vysílání. Existují dva různé postupy umožňující manipulaci se zprávami. Prvním technikou je tzv. zastínění (overshadowing). Při tomto postupu vysílá útočník signál s dostatečně vyšším výkonem, než je výkon legitimního signálu, že na přijímací straně se originální signál jeví jako šum.

Druhou technikou modifikace zpráv je tzv. převracení bitů (bit-flipping). Při tomto postupu se útočník snaží napodobit originální signál a modifikuje pouze specifické bity. Tento postup vyžaduje velmi přesné časování a synchronizaci, která je těžko dosažitelná pro pohybující se letadla.

(SCHÄFER, a další, 2013)

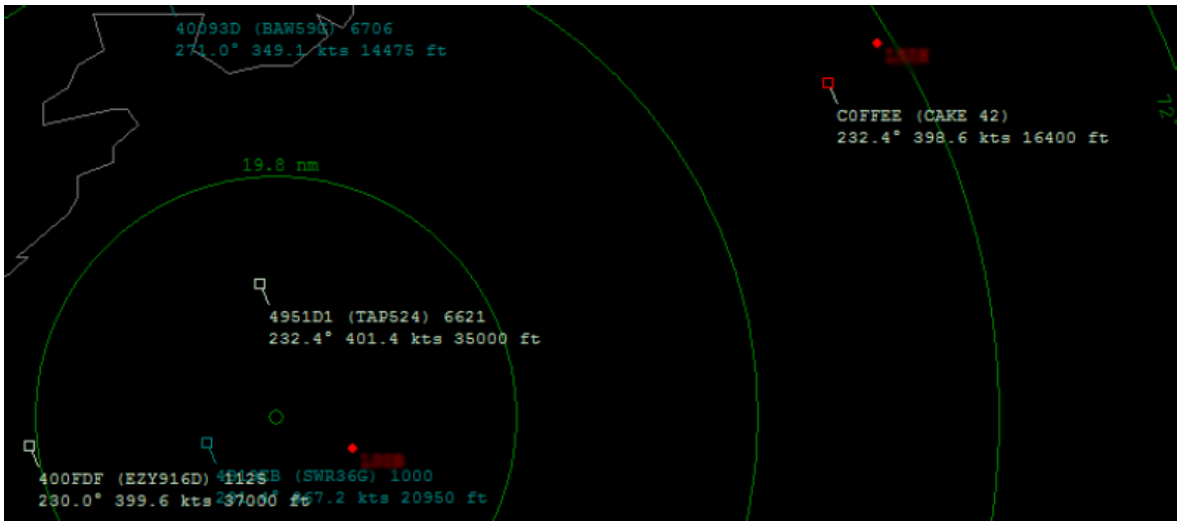
3.8.3 Vybrané aktivní útoky

Vložení neexistujícího letadla (Ghost Aircraft Injection)

Tento útok se zakládá na vysílání falešných ADS-B zpráv. Útočník při něm vysílá ADS-B zprávy neexistujícího cíle na sdíleném ADS-B kanálu. Falešný cíl by měl mít realistické vlastnosti (pozici, rychlost, ID), aby byl nerozpoznatelný od legitimních cílů bez dodatečných informačních zdrojů. Falešný cíl může být vložen jako letící, nebo jako rolující na runway. To může vést ke zmatení řídících letového provozu a společně se sníženou viditelností ke zbytečným zamítnutím odletu / přistání nebo změnám výšky a kurzu okolních letadel.

Systémy pro přecházení kolizí založené na ADS-B pak poskytují útočníkům jednoduchou možnost interference s okolními letadly. Při zhoršené viditelnosti se piloti primárně spoléhají na informace získané z přístrojů. Vložení neexistujícího cíle pak může vést piloty k nežádoucím změnám kurzu, rychlosti nebo výšky.

Tento útok sice přímo havárii nezpůsobí, výsledný zmatek a rozhodnutí pilotů a řídících však může ke vzniku takovéto situace vést.



Obrázek 28 – Příklad zobrazení neexistujícího cíle řídícím letového provozu (ICAO adresa fiktivního cíle: C0FFEE)
[zdroj: (SCHÄFER, a další, 2013 str. 10)]

Záplava neexistujících cílů (Ghost Aircraft Flooding)

Stejný postup jako v předchozím případě vložení neexistujícího letadla s tím rozdílem, že útočník generuje velké množství neexistujících cílů za účelem zahlcení řídících letového provozu. Jedná se o útok typu DoS (Denial of Service).

Obrázek 29 – Příklad zobrazení 100 neexistujících cílů řídícím letového provozu
[zdroj: (SCHÄFER, a další, 2013 str. 11)]

Modifikace virtuální trajektorie

Tento útok se zaměřuje na modifikaci trajektorie existujícího cíle, který vysílá korektní ADS-B zprávy o poloze. Tento útok může být realizován smazáním zpráv o poloze od konkrétního cíle současně s vložením vlastního falešného cíle se stejnou ICAO adresou, nebo přímo modifikací zpráv o poloze existujícího cíle. Bez včasného odhalení může tento útok vést k nesprávným instrukcím od pozemních řídících a opožděným reakcím systému varování před kolizí spoléhajících na data získaná pomocí ADS-B.

Falešný poplach

Obdobně jako u modifikace virtuální trajektorie se útočník snaží modifikovat ADS-B zprávy existujícího cíle. ADS-B poskytuje mechanismy pro indikaci nouzového stavu a nezákonných akcí jako např. únos. Takovýto útok může vyvolat zmatení a zaměřit pozornost zodpovědných osob na napadený cíl. To může vést k zamítnutí přistání a sankcím proti aeroliniím, které provozují napadený stroj. Detekce tohoto útoku bývá velmi obtížná, jelikož v případě únosu je rádiová komunikace považována za nedůvěryhodnou.

Rušení pozemní stanice

Tento útok se zakládá na nepřetržitém mazání zpráv v dosahu pozemní stanice řízení letového provozu. Tento typ útoku je známý a velmi jednoduchý na realizaci. Zasažená stanice řízení letového provozu poté musí přejít na jiné, méně efektivní metody dohledu a řízení. Náhlý výpadek dohledu a systémů varování před kolizí může mít obzvlášť v oblastech s vysokým provozem (např. mezinárodní letiště) fatální následky. Tento útok může být obzvlášť nebezpečný v kombinaci s rušením hlasové rádiové komunikace mezi piloty a řízením letového provozu.

Zmizení letadla

Tento útok se zakládá na nepřetržitém mazání zpráv od konkrétního cíle. Tímto způsobem zabrání útočník detekci cíle pozemní stanicí založenou na ADS-B a okolními letadly. Na rozdíl od předchozího případu je tento útok výrazně nenápadnější, jelikož absence jednoho letadla bude spíše považována za selhání palubních systémů (avioniky) daného stroje. Tento útok s velkou pravděpodobností způsobí mimořádné přistání napadeného letadla za účelem bezpečnostní kontroly a v jeho průběhu ochromí systémy založené na ADS-B, jakým je například systém varování před kolizí.

(SCHÄFER, a další, 2013)

4 IMPLEMENTACE

Následující kapitola se věnuje popisu implementace a struktury prototypu aplikace emulátoru transpondéru ADS-B. V úvodu bude představena zvolená platforma, nad kterou je aplikace implementována. Dále zde budou rozvedeny jednotlivé moduly a způsob jejich spolupráce.

4.1 Platforma

Aplikace je postavena na platformě Java SE[9] ve verzi 8. K vývoji grafického uživatelského rozhraní je využit Framework JavaFX, který je součástí platformy Java SE. Aplikace je vytvořena pomocí open source vývojového prostředí NetBeans IDE. Simulační scénář je exportován do souboru ve formátu XML pomocí frameworku JAXB.

4.1.1 Java

Java je jedním z nejrozšířenějších programovacích jazyků na světě. Jeho hlavní předností je snadná přenositelnost na různé systémy disponující platformou Java bez nutnosti program kompilovat zvlášť pro každý takovýto systém.

Jazyk Java vznikl v roce 1995 pod společností Sun Microsystems. Syntaxe jazyka vycházela zejména z jazyků C a C++, které byly v této době nejrozšířenější. Hlavním příslibem bylo heslo „Write Once, Run Anywhere“ (napiš jednou, spusť kdekoliv) symbolizující univerzálnost jazyka pro všechny podporované platformy. V roce 2009 byla společnost Sun Microsystems koupena společností Oracle Corporation, která s vývojem Javy pokračuje.

(BINSTOCK, 2015), (BYOUS, 1998)

Java na rozdíl od svých předchůdců používá automatickou správu paměti. Programátor tedy rozhoduje pouze o tom, kdy bude daný objekt vytvořen, a musí zajistit, že na tento objekt bude neustále odkazovat alespoň jedna reference. Objekt, na který neodkazuje žádná reference, je považován za nedostupný a je pomocí tzv. garbage colectoru z paměti odstraněn. V jazyce Java tedy neexistuje destruktor. Destruktor je programová konstrukce podobná konstruktoru sloužící k uvolnění paměti při odstraňování objektů, jakou můžeme najít například v jazyce C++. Tento přístup výrazně snižuje možnost výskytu chyb způsobujících únik paměti.

(Oracle, © 1994)

[9] SE = standardní edice (angl. Standard Edition)

Implementace platformy Java je v současnosti rozdělena do dvou distribucí. První distribucí je Java Runtime Environment (JRE), která obsahuje část platformy Java SE potřebnou pro běh programů napsaných v jazyce Java (tj. JVM). Tato distribuce je určena především pro koncové uživatele. Druhou distribucí je Java Development Kit (JDK) určená pro softwarové vývojáře, která navíc obsahuje vývojové nástroje jako je Java kompilátor, Javadoc a debugger.

Při kompilaci programu v jazyce Java se místo strojového kódu pro konkrétní systém vytváří pouze tzv. mezikód (bajtkód). Tento formát je nezávislý na architektuře počítače nebo zařízení. Program pak lze spustit na libovolném zařízení, které má k dispozici tzv. virtuální stroj Javy (JVM – Java Virtual Machine). JVM byl později doplněn technologií JIT (Just In Time compilation), která mezikód za běhu programu dynamicky kompilují do strojového kódu pro daný systém, a přidává řadu dalších optimalizací. To umožňuje programům napsaným v jazyce Java se výkonem vyrovnat programům kompilovaným přímo do strojového kódu.

(ROUSE, 2005)

Výsledkem kompilace programu v jazyce Java je soubor s příponou „jar“. Tento soubor je archiv obsahující zkompilované třídy a jejich metadata společně s dalšími zdroji, které tyto třídy využívají (např. obrázky, zvuky, styly atd.). Archiv může být spustitelnou aplikací, nebo ho lze využít jako softwarovou knihovnu pro jiný program. Pro spuštění souboru s příponou „jar“ musí být na klientovi nainstalována platforma JRE ve verzi stejné nebo vyšší, než je verze JDK, kterou byl daný archiv zkompilován. Platforma JRE je zpětně kompatibilní.

4.1.2 JavaFX

JavaFX je softwarová platforma postavená na bázi platformy Java sloužící k tvorbě aplikací s grafickým uživatelským rozhraním. Platforma JavaFX vznikla za účelem nahradit zastaralý Swing jako standardní knihovnu pro tvorbu GUI aplikací. Platforma byla navržena s důrazem na oddělení aplikační logiky od uživatelského rozhraní. Tím se zajistí vyšší přehlednost a flexibilita zdrojového kódu.

Ke specifikaci vzhledu jednotlivých elementů využívá platforma JavaFX standard CSS ve verzi 2.1 s rozšířeními pro podporu specifických vlastností platformy JavaFX. Pro odlišení vlastností určených pro elementy JavaFX se v CSS používá přípona „`-fx-`“.

(Oracle, © 2008)

Za účelem návrhu GUI byl vytvořen značkovací jazyk FXML na bázi jazyka XML. FXML poskytuje praktickou alternativu k tvorbě GUI přímo v kódu aplikační logiky. Výhodou FXML

je zejména jeho hierarchické uspořádání, které více odpovídá struktuře objektů reprezentujících jednotlivé grafické prvky uživatelského rozhraní. Od verze Java SE 7u6 je knihovna JavaFX součástí platformy Java SE.

(BROWN, 2011)

4.1.3 NetBeans IDE

NetBeans IDE je svobodné, zdarma distribuované integrované vývojové prostředí, které v současné době vlastní a sponzoruje firma Oracle Corporation. Prostředí NetBeans je vytvořeno v jazyce Java. Díky své modulární softwarové architektuře umožňuje prostředí NetBeans programování v celé řadě programovacích jazyků (patří mezi ně např. Java, C++, C#, PHP, …) a rozšíření svých funkcí prostřednictvím tzv. pluginů.

Alternativy:

- Eclipse – Open source prostředí, původně vyvíjené spol. IBM.
- IntelliJ IDEA – Komerční prostředí od společnosti JetBrains.
- JDeveloper – Volně šiřitelné prostředí vyvíjené spol. Oracle Corporation.

Prostředí NetBeans IDE je dostupné pod svobodnou licencí GNU GPL v2, nebo pod licencí CDDL (Common Development and Distribution License).

4.1.4 XML

XML (Extensible Markup Language) je flexibilní značkovací jazyk který byl vyvinut a standardizován konsorciem W3C. Jazyk XML je určen především pro výměnu dat mezi aplikacemi a publikování dokumentů. Data ve formátu XML jsou uložena v čitelné textové podobě a organizována do hierarchické struktury.

Základem struktury XML jsou tzv. elementy. Každý element je definován pomocí počáteční a koncové značky. Počáteční značka je tvořena názvem elementu uzavřeným ve špičatých závorkách. Název elementu nesmí obsahovat mezery.

Příklad počáteční značky:

```
<Element>
```

Koncová značka vypadá stejně, jen je před názvem elementu, který ukončuje navíc znak lomítko (/).

Příklad koncové značky:

```
</Element>
```

Cokoli se nachází mezi těmito značkami, je považováno za potomka daného elementu. Potomkem můžou být textová data nebo jiný element. Každý element XML dokumentu musí mít jak počáteční, tak koncovou značku. Výjimkou jsou prázdné elementy, které lze zapsat zkráceně přidáním znaku lomítko (/) do počáteční značky před ukončovací závorku.

Příklad prázdného elementu:

```
<PrázdnýElement />
```

Elementy se stejným názvem a předkem se mohou opakovat. Výjimkou je tzv. kořenový element, tedy element, který nemá žádného předka. Kořenový element je přímým nebo nepřímým předkem všech elementů v daném dokumentu.

Každý element může mít definované atributy. Atributy se zapisují do počáteční značky elementu. Syntaxe zápisu atributů je název atributu, následovaný znakem rovno (=), následovaným hodnotou atributu v uvozovkách. Jednotlivé atributy s mezi sebou oddělují mezerami. Pro názvy atributů platí stejná pravidla jako pro názvy elementů.

Příklad prázdného elementu s atributy:

```
<Element atribut1="0.5" atribut2="textová hodnota" />
```

Na začátku XML dokumentu se nachází speciální element nazývaný XML deklarace. Ten zpravidla obsahuje atribut *version*, definující verzi standardu XML, a atribut *encoding* udávající použité kódování.

Syntaxe XML deklarace:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Příklad validního XML dokumentu:

```
<?xml version="1.0" encoding="UTF-8" ?>
<KořenovýElement>
    <Element atribut="prázdný element" />
    <Element>
        <Potomek atribut="hodnota" />
    </Element>
    <Element atribut="40" />
</KořenovýElement>
```

Jazyk XML rozlišuje velká a malá písmena.

(BRAY, a další, 2008)

4.1.5 JAXB

Java Architecture for XML Binding (JAXB) je softwarový framework pro mapování Java tříd na XML struktury. Základními funkcionalitami knihovny JAXB je serializace (marshaling)

Java objektů do XML podoby a deserializace (unmarshaling) XML struktur zpět na Java objekty bez nutnosti implementovat specializované serializační a deserializační rutiny pro jednotlivé třídy. Mapování tříd a jejich atributů je realizováno pomocí Java anotací.

Framework JAXB vyžaduje, aby veškeré mapované třídy disponovaly bezparametrickým konstruktorem. Tento konstruktor může být definován jako privátní. Framework JAXB je součástí platformy Java SE.

Příklady použití vybraných anotací

Anotace `@XmlRootElement`

Tato anotace definuje danou třídu jako kořenový element XML souboru. Jméno elementu je odvozeno z názvu třídy, nebo jej lze specifikovat pomocí parametru `name`.

Příklad anotace třídy *SimulationScenario* jako kořenový element:

```
@XmlRootElement(name = "SimulationScenario")
public class SimulationScenario {
```

Anotace `@XmlElement`

Tato anotace mapuje daný atribut třídy jako XML element. Jméno elementu je odvozeno z názvu atributu třídy, nebo jej lze specifikovat pomocí parametru `name`. Parametr `required` značí, zdali je výskyt elementu v XML souboru povinný.

Příklad mapování atributu `scenario` na povinný XML element `AircraftScenario`.

```
@XmlElement(name = "AircraftScenario", required = true)
private AircraftScenario scenario;
```

Anotací `@XmlElement` lze anotovat přímo atributy nebo jejich „get“ metody. Pro anotaci privátních atributů je nutno opatřit danou třídu anotací:

```
@XmlAccessorType(XmlAccessType.FIELD)
```

Anotace `@XmlAttribute`

Tato anotace mapuje daný atribut třídy jako XML atribut nadřazeného elementu, kterým je daná třída. Jméno XML atributu je odvozeno z názvu atributu, nebo jej lze specifikovat pomocí parametru `name`. Parametr `required` značí, zdali je výskyt atributu v daném elementu povinný.

Příklad mapování třídního atributu `icao` jako povinný XML atribut nadřazeného elementu:

```
@XmlAttribute(required = true)
private int icao;
```

Anotace `@XmlElementWrapper`

Tato anotace se obvykle využívá společně s anotací `@XmlElement` a to v případě, kdy mapujeme kolekci objektů. Použití této anotace způsobí „zabalení“ elementů vzniklých mapováním dané kolekce do elementu s názvem specifikovaným parametrem `name`.

Příklad zabalení XML elementů `waypoint` vzniklých mapováním kolekce `waypoints` do XML elementu `waypoints`:

```
@XmlElement(name = "Waypoint", required = true)
@XmlElementWrapper(name = "Waypoints", required = true)
private List<Waypoint> waypoints = new ArrayList<>();
```

Výsledné XML potom může vypadat následovně:

```
<Waypoints>
    <Waypoint />
    <Waypoint />
    <Waypoint />
</Waypoints>
```

Anotace `@XmlElements`

Tato anotace umožňuje definovat více anotací `@XmlElement` a používá se pro mapování polymorfních kolekcí. V jednotlivých anotacích `@XmlElement` se pomocí parametru `type` definují konkrétní třídy, které se mohou v dané kolekci vyskytovat, a pomocí parametru `name` lze specifikovat název XML elementu na který budou tyto konkrétní třídy mapovány.

Na následujícím příkladu je kód mapující polymorfní kolekci `messages`, která může obsahovat konkrétní typy `AirbornePositionMessage`, `IdentificationMessage` a `AirborneVelocityMessage`. Všechny elementy vzniklé mapováním kolekce `messages` jsou „zabaleny“ do elementu `Messages`, jehož výskyt je v nadřazeném elementu povinný.

```
@XmlElements({
    @XmlElement(name = "AirbornePositionMessage", type =
AirbornePositionMessage.class),
    @XmlElement(name = "AirborneVelocityMessage", type =
AirborneVelocityMessage.class),
    @XmlElement(name = "IdentificationMessage", type =
IdentificationMessage.class)
})
@XmlElementWrapper(name = "StandaloneMessages", required = true)
private List<StandaloneMessage> messages;
```

Výsledné XML potom může vypadat následovně:

```
<StandaloneMessages>
    <AirbornePositionMessage />
    <AirbornePositionMessage />
    <IdentificationMessage />
    <AirborneVelocityMessage />
</StandaloneMessages>
```

Anotace `@XmlTransient`

Tato anotace způsobí, že bude daný atribut třídy nebo třída při mapování ignorována a ve výsledném XML dokumentu se nevyskytuje.

Metoda `afterUnmarshal`

Je-li v mapované třídě definována metoda s následující signaturou:

```
private void afterUnmarshal(Unmarshaller unmarshaller, Object parent)
```

Je tato metoda automaticky zavolána vždy, když je dokončena deserializace instance dané třídy. Tuto metodu lze poté využít k dodatečné inicializaci atributů dané instance. Důležitou vlastností této metody je, že je volána až po načtení všech atributů, které jsou mapovány v XML dokumentu. Parametr `parent` obsahuje referenci na instanci nadřazené třídy mapované na rodičovský XML element. V případě, kdy je metoda `afterUnmarshal` definována v kořenové třídě (opatřené anotací `@XmlRootElement`) odkazuje parametr `parent` na hodnotu `null`. Tato metoda může být definována jako privátní.

Příklad serializace pomocí frameworku JAXB

Následující kód provede serializaci instance třídy `AircraftScenario` pojmenovanou `aircraftScenario` jako kořenový element XML dokumentu a výsledný dokument zapíše do souboru `file` (instance třídy `File` odkazující na platný soubor).

```
JAXBContext context = JAXBContext.newInstance(AircraftScenario.class);
Marshaller m = context.createMarshaller();
m.marshal(aircraftScenario, new FileWriter(file));
```

Příklad deserializace pomocí frameworku JAXB

Následující kód deserializuje instanci třídy `AircraftScenario` z XML souboru `file`.

```
JAXBContext context = JAXBContext.newInstance(AircraftScenario.class);
Unmarshaller um = context.createUnmarshaller();
AircraftScenario aircraftScenario =
   (AircraftScenario) um.unmarshal(new FileReader(file));
```

4.2 Struktura aplikace

Následující kapitola popisuje strukturu prototypu softwarového emulátoru transpondéru. Jsou zde představeny jednotlivé balíčky (moduly) a vybrané třídy, a je zde představena jejich funkcionalita.

Jedinou třídou v kořenovém balíčku `ssrt` je třída `MainApp`. Tato třída definuje metodu `start`, která je vstupním bodem do aplikace a slouží především k inicializaci JavaFX frameworku. Třída `MainApp` dále definuje metodu `main`, která zajišťuje zpětnou kompatibilitu s IDE, která plně nepodporují JavaFX. Dále jsou zde definovány globální konstanty využívané v rámci celé aplikace.

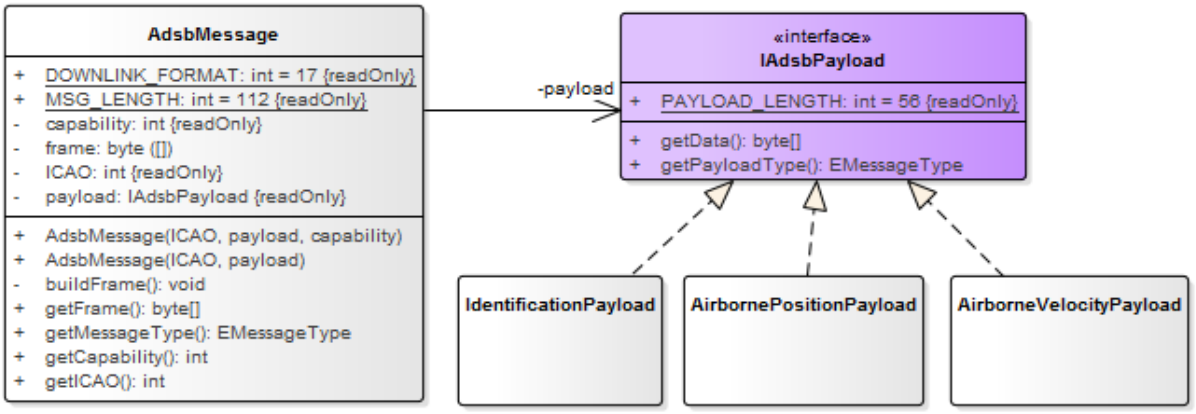
4.2.1 Balíček `utils`

Tento balíček obsahuje pomocné třídy využívané v rámci celé aplikace. Třída `Conversions` obsahuje veřejné statické metody pro konverzi mezi jednotkami. Třída `AirborneNic` slouží k určení parametru Type Code (TC) a parametru NIC supplement-B (NIC-B) pro ADS-B zprávy o poloze ve vzduchu (viz. kapitola 3.5) a definuje statickou tovární metodu `calculateTypeCodeAndNicSb` a veřejné konstanty `typeCode` a `nicSb`, do kterých jsou uloženy výsledné parametry.

Třída `SimpleJaxbSerializer` je pomocná třída sloužící pro serializaci scénáře do souboru XML a deserializaci scénáře ze souboru XML pomocí frameworku JAXB. Poslední třídou v tomto balíčku je třída `Utils` obsahující pomocné statické metody používané v rámci celé aplikace.

4.2.2 Balíček `adsb`

Tento balíček obsahuje třídy definující samotné ADS-B zprávy. Balíček obsahuje třídu `AdsbMessage` reprezentující společnou část ADS-B zpráv a rozhraní `IAdsbPayload` které implementují třídy reprezentující datovou část ADS-B zpráv. Dále se zde nachází balíček `payloads` obsahující třídy implementující rozhraní `IAdsbPayload` a balíček `enums` obsahující enumerace, které souvisí s obsahem ADS-B zpráv. Následující obrázek (Obrázek 30) obsahuje diagram tříd popisující vztah mezi třídou `AdsbMessage` a rozhraním `IAdsbPayload`.



Obrázek 30 – Diagram tříd: Modul ADS-B
[zdroj: vlastní]

Konstruktor třídy `AdsbMessage` má následující parametry:

1. ICAO adresa jako celé číslo typu integer. Pro převod z hexadecimálního zápisu lze využít metody `icaoToInt` z třídy `Utils`.
2. Instance třídy implementující rozhraní `IAdsbPayload` reprezentující konkrétní datovou část ADS-B zprávy.
3. Volitelně parametr `capability` (integer od 0 do 7).

Důležitou metodou třídy `AdsbMessage` je metoda `getFrame`, která vrací pole bytů reprezentující ADS-B rámec v binární podobě. Sestavení tohoto pole bytů probíhá již při vytváření instance třídy `AdsbMessage` a následná změna parametrů včetně obsahu datové části již není možná. Při vytváření instance zároveň probíhá výpočet kontrolního součtu CRC, který tvoří posledních 24 bitů ADS-B zprávy. Pro výpočet kontrolního součtu slouží metoda `calcParity` z třídy `Utils`, která jako parametr přijímá pole bytů obsahující sestavenou ADS-B zprávu bez posledních 24 bitů a vrací pole 3 bytů (24 bitů) obsahujících CRC pro danou zprávu.

Třídy reprezentující datovou část ADS-B zprávy implementují rozhraní `IAdsbPayload`, které předepisuje metodu `getData` a `getMessageType`. Metoda `getData` musí při zavolání vrátit pole 7 bytů reprezentující datovou část ADS-B rámce.

Pro sestavení pole bytů jak ve třídě `AdsbMessage`, tak ve třídách reprezentující datovou část zprávy se využívají následující metody z třídy `Utils`:

- `toBinary` – Převede hodnotu typu integer na pole typu boolean o velikosti specifikované druhým parametrem reprezentující danou hodnotu v binární podobě (posloupnost hodnot `true` a `false`).
- `concatArrays` – Pomocná metoda slouží pro spojení více polí do jednoho v pořadí, v jakém jsou jednotlivá pole specifikována jako parametry.
- `booleansToBytes` – Převede pole typu boolean na pole bytů. Převáděné pole typu boolean musí mít velikost, která je beze zbytku dělitelná 8.
- `calcParity` – Vypočítá 24 bitové CRC z pole bytů dodaného jako parametry. Vrací pole 3 bytů reprezentující součet kontrolní parity pro danou zprávu.

Třída `IdentificatioPayload`

Tato třída reprezentuje datovou část ADS-B zprávy pro identifikaci letadla (viz. kapitola 3.4). V konstruktoru se očekává pole 8 instancí enumerace `ECharactor` a jedna instance enumerace `EEmitterCategory`.

Enumerace `ECharactor` obsahuje výčet všech znaků použitelných v ADS-B zprávě pro identifikaci letadla společně s příslušným číselným kódem daného znaku. Číselný kód daného znaku lze získat pomocí metody `getCode` nad instancí `ECharactor`. Pro převedení textového řetězce na pole typu `ECharactor` lze využít statickou metodu `toCharactors` enumerace `ECharactor`.

Třída `AirbornePositionPayload`

Tato třída reprezentuje datovou část ADS-B zprávy pro polohu ve vzduchu (viz. kapitola 3.5). V konstruktoru se očekávají následující parametry v daném pořadí:

1. `latitude` – Zeměpisná šířka v rozsahu od -90 do 90.
2. `longitude` – Zeměpisná délka v rozsahu od -180 do 180.
3. `altitude` – Nadmořská výška v metrech.
4. `oddFrame` – `True`, jedná-li se o lichý rámec. `False`, jedná-li se o sudý rámec.
5. `ss` – Kód pro Surveillance Status. Kódy s popisem se nachází v enumeraci `ESurveillanceStatus`. Definován je rozsah od 0 do 3.
6. `nicSb` – `True`, pokud je nastaven NIC supplement-B.
7. `time` – `True`, pokud je nastaven Time bit.
8. `qBit` – `True`, je-li použito kvantování 25 stop pro kódování výšky (Q-Bit = 1), v opačném případě se použije kvantování 100 stop (Q-Bit = 0).

9. `typeCode` – Hodnota Type Code (TC), která společně s NIC supplement-B určuje maximální poloměr horizontálního ohraničení R_c.

Pro zjištění TC a NIC supplement-B na základě známého R_c lze využít třídu `AirborneNic` z balíčku `utils`. Tato třída obsahuje tovární metodu `calculateTypeCodeAndNicSb`, která jako 1. parametr přijímá instanci enumerace *EAltitudeSource*, určující zdroj údaje o výšce, a 2. parametrem je maximální poloměr horizontálního ohraničení R_c. Tato metoda vrací instanci třídy `AirborneNic` obsahující konstanty `typeCode` (kód pro pole TC) a `nicSb` (nastavení bitu NIC supplement-B).

Třída `AirborneVelocityPayload`

Tato třída reprezentuje datovou část ADS-B zprávy pro rychlost ve vzduchu (viz. kapitola 3.6). Tato třída má 2 varianty konstruktoru. 1. je varianta pro rychlost vůči zemskému povrchu (Ground Speed) a 2. varianta je pro letovou rychlost (Airspeed).

V 1. variantě konstruktoru se očekávají následující parametry v daném pořadí:

1. `northSouthVelocityMs` – Rychlost ve směru sever – jih v metrech za sekundu. Záporná hodnota indikuje rychlost v opačném směru. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).
2. `eastWestVelocityMs` – Rychlost ve směru východ – západ v metrech za sekundu. Záporná hodnota indikuje rychlost v opačném směru. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).
3. `verticalSpeedMs` – Rychlost stoupání v metrech za sekundu. Záporná hodnota indikuje rychlost klesání. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).
4. `subtype` – Instance enumerace `ESubtype` indikující nastavení pole subtype (ST) a určující kvantovací krok pro kódování rychlosti. V této variantě konstruktoru by se měl předávat pouze subtype 1 (Ground Speed subsonic) a 2 (Ground Speed supersonic).
5. `NAC` – Kód pro Navigation Accuracy Category (NAC). Definovány jsou hodnoty od 0 do 4 (viz. Tabulka 14 – Kódy pro pole NAC).
6. `verticalRateSource` – `False`, pochází-li údaj vertikální rychlosti ze systému GNSS. `True`, pochází-li údaj vertikální rychlosti z barometrického výškoměru.
7. `intentChangeFlag` – Nastavení bitu Intent Change Flag (IC).
8. `diffFromBaroAlt` – Rozdíl mezi geometrickou výškou nad elipsoidem (HAE) naměřenou systémem GNSS a výškou naměřenou barometrickým výškoměrem v metrech. Kladná

hodnota znamená že údaj geometrické výšky je vyšší než údaj výšky z barometrického výškoměru. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).

Dílčí rychlosti ve směru sever – jih a východ – západ lze získat pomocí vztahů:

$$v_{NS} = \cos(HDG) * GS$$

$$v_{EW} = \sin(HDG) * GS$$

Kde v_{NS} je rychlost ve směru sever – jih, v_{EW} je rychlost ve směru východ – západ, GS je naměřená horizontální rychlost a HDG je směr letu.

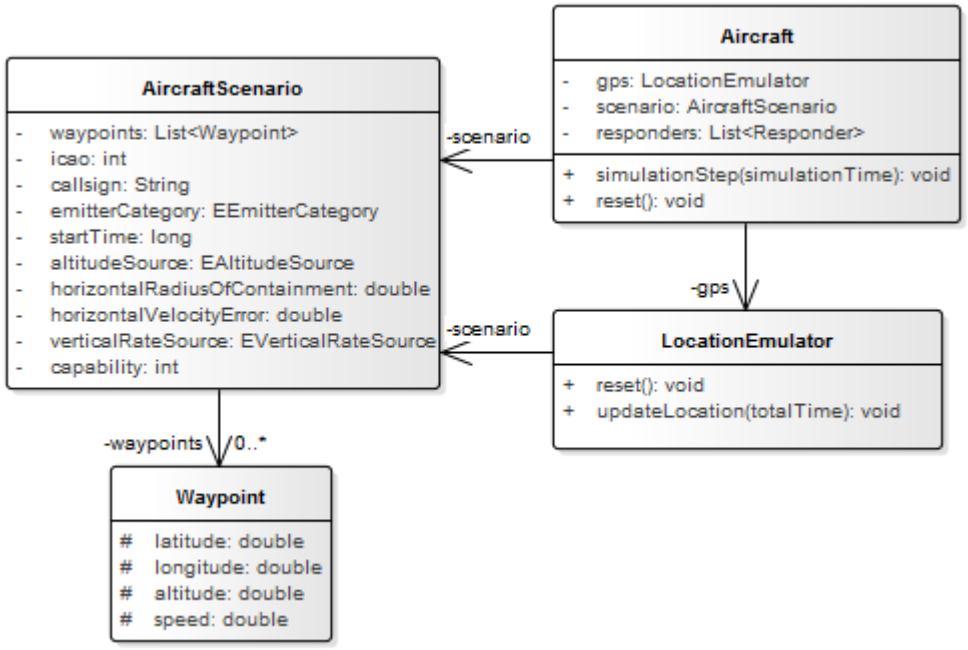
V 2. variantě konstruktoru se očekávají následující parametry v daném pořadí:

1. `airspeed` – Letová rychlost v metrech za sekundu. Tento parametr musí být nezáporný. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).
2. `heading` – Směr letu ve stupních od 0 do 360 podle směru hodinových ručiček počínaje severem.
3. `verticalSpeedMs` – Rychlost stoupání v metrech za sekundu. Záporná hodnota indikuje rychlost klesání. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).
4. `asT` – `False`, pokud přenášený údaj rychlosti je indikovaná letová rychlost (IAS). `True`, pokud přenášený údaj rychlosti je reálná letová rychlost (TAS).
5. `subtype` – Instance enumerace `ESubtype` indikující nastavení pole subtype (ST) a určující kvantovací krok pro kódování rychlosti. V této variantě konstruktoru by se měl předávat pouze subtype 1 (Ground Speed subsonic) a 2 (Ground Speed supersonic).
6. `NAC` – Kód pro Navigation Accuracy Category (NAC). Definovány jsou hodnoty od 0 do 4 (viz. Tabulka 14 – Kódy pro pole NAC).
7. `verticalRateSource` – `False`, pochází-li údaj vertikální rychlosti ze systému GNSS. `True`, pochází-li údaj vertikální rychlosti z barometrického výškoměru.
8. `intentChangeFlag` – Nastavení bitu Intent Change Flag (IC).
9. `diffFromBaroAlt` – Rozdíl mezi geometrickou výškou nad elipsoidem (HAE) naměřenou systémem GNSS a výškou naměřenou barometrickým výškoměrem v metrech. Kladná hodnota znamená že údaj geometrické výšky je vyšší než údaj výšky z barometrického výškoměru. Nedostupnost tohoto údaje lze nastavit předáním hodnoty `NaN` (`Double.NaN`).

4.2.3 Balíček `aircraft`

Tento balíček obsahuje třídy, které zajišťují simulaci pohybu letadla v čase. Instance třídy `Aircraft` reprezentuje konkrétní letadlo. Každá instance třídy `Aircraft` obsahuje instanci

třídy `LocationEmulator`, která zajišťuje simulaci pohybu po zadané trajektorii a instanci třídy `AircraftScenario`, která definuje trajektorii pohybu a další parametry daného letadla. Dále instance třídy `Aircraft` obsahuje polymorfní kolekci instancí potomků třídy `Responder`, které reprezentují transpondér ASD-B a je jim věnována následující kapitola. Následující diagram tříd znázorňuje vztahy mezi třídami v balíčku `aircraft`.



Obrázek 31 – Diagram tříd: Modul Aircraft
[zdroj: vlastní]

Třída `AircraftScenario`

Tato třída reprezentuje simulační scénář daného letadla (instance třídy `Aircraft`). Kromě trasy letadla se zde nastavují i další parametry ovlivňující chování letadla během simulace a podobu generovaných ADS-B zpráv.

Atributy třídy `AircraftScenario`:

- `waypoints` – Kolekce instancí třídy `Waypoint` reprezentující body trasy pohybu letadla. Pro správnou funkci simulátoru musí být definovány alespoň 2 body trasy.
- `icao` – ICAO adresa jako celé číslo typu integer. Pro převod z hexadecimálního zápisu lze využít metody `icaoToInt` z třídy `Utils`.
- `callsign` – Volací znak letadla. Textový řetězec by měl mít přesně 8 znaků ze sady definované v enumeraci `ECharactor`. Velká a malá písmena se nerozlišují.
- `emitterCategory` – Instance enumerace `EEmitterCategory` určující typ stroje.
- `startTime` – Simulační čas počátku aktivity (spuštění) daného letadla v milisekundách.

- `altitudeSource` – Instance enumerace `EAltitudeSource` určující zdroj informace o výšce.
- `horizontalRadiusOfContainment` – Maximální poloměr horizontálního ohraničení (R_c).
- `horizontalVelocityError` – Maximální odchylka (chyba) naměřené horizontální rychlosti s pravděpodobností 95 %.
- `geometricAltitudeError` – Maximální odchylka (chyba) naměřené geometrické (GNSS) výšky s pravděpodobností 95 %.
- `barometricAltitudeError` – Maximální odchylka (chyba) naměřené barometrické výšky s pravděpodobností 95 %.
- `verticalRateSource` – Instance enumerace `EVerticalRateSource` určující zdroj informace o rychlosti stoupání / klesání.
- `capability` – Kód pro pole capability (celé číslo od 0 do 7).

Třída `Waypoint`

Tato třída reprezentuje body trasy pohybu letadla. Každý bod trasy je definován následujícími atributy:

- `latitude` – Zeměpisná šířka ve stupních od -90° do 90°.
- `longitude` – Zeměpisná délka ve stupních od -180° do 180°.
- `altitude` – Nadmořská výška v tomto bodě v metrech.
- `speed` – Rychlost pohybu z tohoto bodu v kilometrech za hodinu.

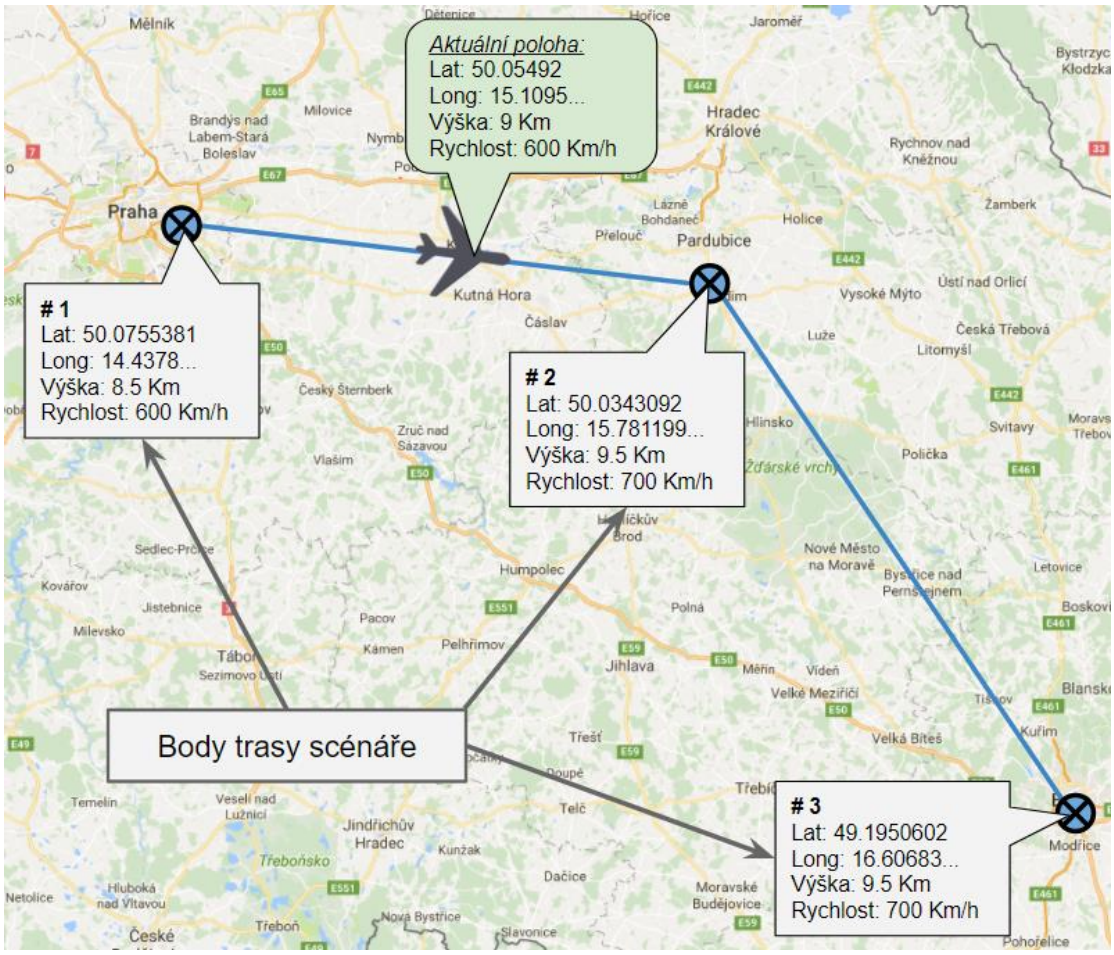
Třída `LocationEmulator`

Tato třída zajišťuje evoluci pohybu letadla (cíle) po zadané trajektorii v průběhu simulace. Simulace pohybu letadla je v implementovaném prototypu značně zjednodušena. Letadla se pohybují po přímých trajektoriích mezi zadanými body trasy. Rychlost pohybu se mění skokově na každém bodu trasy podle hodnoty parametru `speed` daného bodu.

Důležitou metodou této třídy je metoda `updateLocation`, která při zavolání aktualizuje parametry na hodnotu v čase daném parametrem `totalTime`[10] v milisekundách. Při každém zavolání metody `updateLocation` je aktualizována aktuální poloha, výška, rychlost a směr pohybu letadla. Pro výpočet nové polohy v geodetických souřadnicích je využita open source knihovna Geodesy[11]. Další důležitou metodou je metoda `reset`, která nastaví evoluci pohybu

[10] Tento parametr je vypočítán jako rozdíl simulačního času a atributu `startTime` ve scénáři daného letadla.
[11] http://www.gavaghan.org/blog/free-source-code/geodesy-library-vincentys-formula-java/

letadla na její počátek a aktualizuje aktuální polohu, výšku, rychlost a směr na hodnoty odpovídající počátku evoluce. Následující obrázek (Obrázek 32) znázorňuje trasu letadla danou scénářem a jeho aktuální polohu v průběhu simulace danou aktuálním stavem instance třídy `LocationEmulator`.



Obrázek 32 – Přiklad bodů trasy letadla a aktuální poloha
[zdroj: vlastní]

Další vybrané metody třídy `LocationEmulator`:

- `getTotalDistance` – Vrátí celkovou délku trasy v metrech.
- `getCurrentLocation` – Vrátí aktuální polohu a výšku letadla.
- `getCurrentSpeedMs` – Vrátí aktuální rychlost letadla v metrech za sekundu.
- `getCurrentGroundSpeedMs` – Vrátí aktuální rychlost letadla vůči zemskému povrchu v metrech za sekundu.
- `getCurrentVerticalSpeedMs` – Vrátí aktuální rychlost stoupání / klesání letadla v metrech za sekundu.

- `getDistanceTravelled` – Vrátí dosavadní překonanou vzdálenost v metrech.
- `getCurrentBearing` – Vrátí aktuální směr letu ve stupních v rozsahu od 0° do 360° podle směru hodinových ručiček počínaje severem.

Třída `Aircraft`

Třída `Aircraft` zapouzdřuje třídy `AircraftScenario` a `LocationEmulator` a pro přístup k aktuálním informacím jako jsou poloha, různé typy rychlostí a další parametry se doporučuje využívat metody této třídy. Důležitou metodou je metoda `simulationStep`, která zajistí aktualizaci polohy letadla a případné vyslání ADS-B zpráv, a metoda `reset`, která zajistí nastavení všech parametrů na počátek evoluce. Metoda `simulationStep` přijímá jako parametr aktuální simulační čas v milisekundách.

Další vybrané metody třídy `Aircraft`:

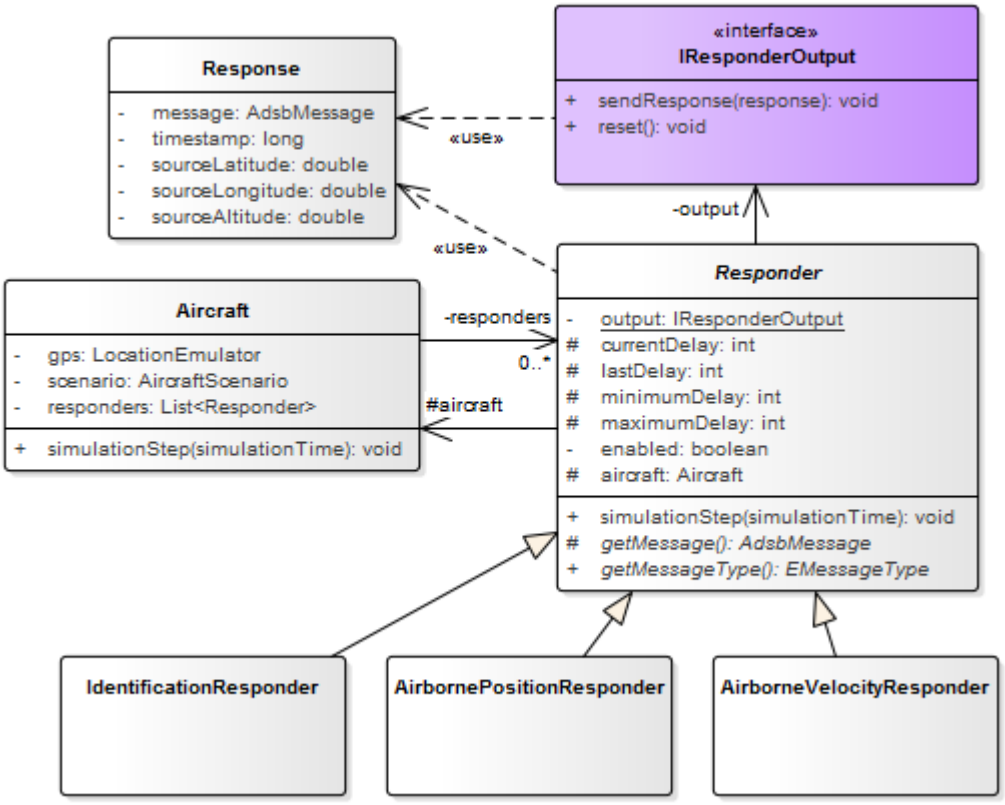
- `getAircraftStatus` – Vrátí instanci enumerace `EAircraftStatus`, která značí stav letadla v rámci simulace. Metoda přijímá jako parametr aktuální simulační čas v milisekundách. Definovány jsou následující stavy:
  - `WAIT` – Letadlo čeká na aktivaci v případě, kdy je hodnota atributu `startTime` ve scénáři daného letadla (instance `AircraftScenario`) vyšší než aktuální simulační čas. V tomto stavu nejsou vysílány žádné ADS-B zprávy.
  - `FINISHED` – Letadlo dokončilo zadanou trasu (byl dosažen poslední bod trasy). V tomto stavu nejsou vysílány žádné ADS-B zprávy.
  - `ACTIVE` – Letadlo je aktuálně v pohybu po zadané trase a aktivně vysílá ADS-B zprávy, jsou-li povoleny.
- `getLatitude` – Vrátí aktuální zeměpisnou šířku ve stupních od -90° do 90°.
- `getLongitude` – Vrátí aktuální zeměpisnou délku ve stupních od -180° do 180°.
- `getAltitude` – Vrátí aktuální výšku v metrech.
- `getSpeedMs` – Vrátí aktuální rychlost letadla v metrech za sekundu.
- `getGroundSpeedMs` – Vrátí aktuální rychlost letadla vůči zemskému povrchu v metrech za sekundu.
- `getVerticalSpeedMs` – Vrátí aktuální rychlost stoupání / klesání letadla v metrech za sekundu.
- `setScenario` – Nastaví scénář pro dané letadlo a inicializuje evoluci na jeho počátek. Jako parametr přijímá instanci třídy `AircraftScenario`.
- `getScenario` – Vrátí aktuální scénář daného letadla.

- `getBearing` – Vrátí aktuální směr letu ve stupních v rozsahu od 0° do 360° podle směru hodinových ručiček počínaje severem.
- `getIcao` – Získá ICAO adresu daného letadla jako celé číslo typu integer. Pro převod na hexadecimální textový řetězec lze využít metody `icaoToHexString` ve třídě `Utils`.
- `getAircraftIdentification` – Vrátí volací znak daného letadla jako textový řetězec.

Dále třída `Aircraft` zapouzdřuje kolekci potomků abstraktní třídy `Responder`, kterým je věnována následující kapitola.

4.2.4 Balíček `responder`

Třídy v tomto balíčku mají na starost generování ADS-B zpráv v průběhu simulace. Abstraktní třída `Responder` slouží jako předek pro všechny konkrétní respondéry, které se specializují na generování konkrétního typu ADS-B zpráv. Každé letadlo (instance třídy `Aircraft`) může vlastnit několik těchto respondérů. Následující obrázek (Obrázek 33) obsahuje diagram tříd popisující vztah mezi třídami `Aircraft`, `Responder` a rozhraním `IResponderOutput`.



Obrázek 33 – Diagram tříd: Modul Responder
[zdroj: vlastní]

Třída `Responder` poskytuje základní funkcionalitu pro konkrétní potomky této třídy. Třída definuje metodu `simulationStep`, která je volána z instance třídy `Aircraft` vlastnící

danou instanci třídy `Responder` při každém simulačním kroku. Tato metoda zajišťuje časování vysílání ADS-B zpráv podle nastavení minimální a maximální prodlevy mezi vysíláním pomocí parametrů `minimumDelay` a `maximumDelay`. Metoda přijímá jako parametr aktuální simulační čas v milisekundách.

Atribut `enabled` třídy `Responder` určuje, zdali je vysílání pro danou instanci povoleno. V případě, že je nastaven na hodnotu `false`, neprobíhá při zavolání metody `simulationStep` vysílání zpráv ani odpočítávání prodlevy do dalšího vysílání.

V případě, kdy je atribut `enabled` nastaven na hodnotu `true`, a hodnota atributu `currentDelay` je při zavolání metody `simulationStep` nižší než 0, je volána metoda `sendResponse` statického atributu `output` (instance třídy implementující rozhraní `IResponderOutput`). Tato metoda přijímá jako parametr instanci třídy `Response`. Třída `Response` zapouzdřuje ADS-B zprávu společně s dalšími informacemi a je blíže popsána v následující kapitole. Příslušná ADS-B zpráva (instance třídy `AdsbMessage`) je získána zavoláním abstraktní metody `getMessage`, kterou implementují jednotliví potomci třídy `Responder`. Hodnota atributu `currentDelay` je při každém volání metody `simulationStep` snížena o délku simulačního kroku.

Třídy dědící abstraktní třídu `Responder` musí implementovat následující metody:

- `getMessage` – Sestaví a vrátí příslušnou ADS-B zprávu (instanci třídy `AdsbMessage`).
- `getMessageType` – Vrátí typ ADS-B zpráv (instance enumerace `EMessageType`) na které je daný potomek zaměřen.
- `getDefaultMaximumDelay` – Vrátí výchozí maximální prodlevu mezi zprávami daného typu.
- `getDefaultMinimumDelay` – Vrátí výchozí minimální prodlevu mezi zprávami daného typu.

Statický atribut `output` třídy `Responder` reprezentuje sdílený kanál pro komunikaci systému ADS-B. Všechny instance potomků třídy `Responder` používají pro vysílání zpráv jednu konkrétní instanci třídy implementující rozhraní `IResponderOutput`.

Všechny třídy implementující rozhraní `IResponderOutput` musí implementovat metodu `sendResponse`, která jako jediný parametr obdrží instanci třídy `Response` zapouzdřující ADS-B zprávu. V této metodě lze poté zpracovat výstupní ADS-B zprávu (např. vypsat na

obrazovku). Dále musí implementovat metodu `reset`, která je volána při resetu simulace do výchozího (počátečního) stavu. V této metodě je poté možno implementovat rutiny, které se spustí při resetování simulace (např. vyprázdnit okno výstupu).

Třída `Response`

Tato třída slouží k zapouzdření ADS-B zpráv společně s dalšími informacemi. Třída `Response` definuje následující atributy:

1. `message` – Instance třídy `AdsbMessage` reprezentující sestavenou ADS-B zprávu.
2. `timestamp` – Simulační čas při vytvoření zprávy v milisekundách.
3. `sourceLatitude` – Skutečná zeměpisná šířka cíle v době vyslání zprávy.
4. `sourceLongitude` – Skutečná zeměpisná délka cíle v době vyslání zprávy.
5. `sourceAltitude` – Skutečná nadmořská výška cíle v době vyslání zprávy.

Pořadí v seznamu odpovídá pořadí parametrů v konstruktoru třídy `Response`.

Dále tato třída definuje anotace nezbytné pro serializaci zpráv do formátu XML. Výsledné XML poté obsahuje jediný element `Message`, který obsahuje danou ADS-B zprávu jako hexadecimální řetězec. Element `Message` dále definuje následující atributy:

- `timestamp` – Simulační čas při vytvoření zprávy v nanosekundách.
- `ICAO` – ICAO adresa cíle v hexadecimální podobě.
- `sourceLatitude` – Skutečná zeměpisná šířka cíle v době vyslání zprávy.
- `sourceLongitude` – Skutečná zeměpisná délka cíle v době vyslání zprávy.
- `sourceAltitude` – Skutečná nadmořská výška cíle v době vyslání zprávy.
- `messageType` – Typ ADS-B zprávy.

Třída `StandaloneMessage`

Abstraktní třída `StandaloneMessage` z balíčku `responder` je velmi podobná třídě `Responder`, slouží však k vysílání samostatných ADS-B zpráv v nastaveném simulačním čase, které nejsou vázány na simulovaný cíl (letadlo). Atribut `sendTime` určuje simulační čas v milisekundách, ve kterém má být zpráva vysílána. Zbylé atributy reprezentují nastavení společných parametrů pro danou ADS-B zprávu (nezávislých na typu ADS-B zprávy).

Na rozdíl od třídy `Responder` zde chybí metoda `simulationStep`. V případě, kdy aktuální simulační čas překročí hodnotu atributu `sendTime` dané instance třídy `StandaloneMessage` je simulátorem volána metoda `sendResponse`. Tato metoda

následně volá metodu `sendResponse` statického atributu `output` (instance třídy implementující rozhraní `IResponderOutput`), která jako parametr přijímá zapouzdřenou ADS-B zprávu (instanci třídy `Response`). Potomci třídy `StandaloneMessage` vždy implementují metodu `getMessage` a `getMessageType` stejným způsobem jako u abstraktní třídy `Responder`. Potomci třídy `StandaloneMessage` dále definují atributy, podle kterých jsou nastaveny parametry konkrétního typu ADS-B zprávy.

4.2.5 Balíček `simulation`

Balíček `simulation` obsahuje rozhraní `ISimulation`, které definuje základní metody pro ovládání simulace. Dále rozhraní definuje konstantu `TIME_STEP`, která určuje simulační krok v milisekundách.

Metody rozhraní `ISimulation` a jejich význam:

- `getSimulationTime` – Vrátí aktuální simulační čas v milisekundách.
- `start` – Spustí simulaci.
- `stop` – Pozastaví simulaci a zachová aktuální stav. V simulaci lze pokračovat pomocí metody `start`.
- `reset` – Pozastaví a resetuje simulaci na počáteční stav (simulační čas 0 *ms*).
- `isRunning` – Vrátí `true` pokud simulace probíhá, nebo `false` je-li pozastavena.

Dále tento balíček obsahuje třídu `SimulationScenario`, jejíž účelem je zapouzdřovat kompletní scénář simulace. Při ukládání scénáře do souboru XML je vytvořena instance této třídy obsahující seznam entit (letadel a samostatných zpráv) a příp. další parametry simulátoru. Tato instance je následně pomocí JAXB serializována do formátu XML, který představuje kompletní scénář simulace.

4.2.6 Balíček `controllers`

Tento balíček obsahuje třídy obsluhující především uživatelské rozhraní (tzv. kontroléry). Výjimkou je třída `SimulationController`, která navíc implementuje rozhraní `ISimulation` a zajišťuje ovládání a běh simulace.

Třída `SimulationController`

Tato třída implementuje metody rozhraní `ISimulation` tak, jak bylo popsáno v přechozí kapitole. V průběhu simulace je za pomoci časovače periodicky volána metoda `simulationStep`. Tato metoda projde cyklem seznam všech letadel (instance třídy

`Aircraft`) a pro každé vyvolá metodu `simulationStep` dané instance. Zároveň testuje, zdali dané letadlo dokončilo celou svou trasu. Dále vybere ze zásobníku samostatných zpráv (instance třídy `StandaloneMessage`) všechny zprávy, jejichž atribut `sendTime` je nižší nebo roven aktuálnímu simulačnímu času. Pokud již všechna letadla dokončila svou trasu a v zásobníku nezbývají žádné samostatné zprávy, je pomocí metody *stop* simulace pozastavena. V opačném případě je navýšen aktuální simulační čas o hodnotu simulačního kroku, který je definován konstantou `TIME_STEP` v rozhraní `ISimulation` (viz. kapitola 4.2.5).

Metoda *reset* nastaví simulační čas na 0, následně projde cyklem všechna letadla a zavolá jejich instanční metody *reset*, které nastaví jejich evoluci na počátek. Dále je zde volána metoda *reset* sdíleného výstupního kanálu implementujícího rozhraní `IResponderOutput`.

Třída `MainController`

Tato třída má na starost obsluhu hlavního menu aplikace. Také jsou zde implementovány rutiny pro uložení a načtení simulačního scénáře do souboru XML. Při spuštění aplikace je zde také načten výchozí simulační scénář.

Třída `OutputController`

Tato třída implementuje rozhraní `IResponderOutput` a má na starost okno výstupu. Při zavolání metody `sendResponse` v průběhu simulace jsou zapouzdřené ADS-B zprávy (instance třídy `Response`) ukládány a zobrazeny ve výstupní tabulce. Metoda `reset` zajistí vyprázdnění výstupní tabulky. Zároveň je zde implementována rutina pro uložení výstupních zpráv do textového souboru s možností uložení ve formátu XML.

Třída `TcpServerController`

Tato třída spravuje server pro vysílání výstupních zpráv pomocí protokolu TCP. Stejně jako v předchozím případě tato třída implementuje rozhraní `IResponderOutput`. Při zavolání metody `sendResponse` v průběhu simulace jsou ADS-B zprávy vysílány v binární nebo XML podobě všem připojeným klientům.

Třída `AircraftScenarioController`

Tato třída obsluhuje dialogové okno pro vytváření a editaci scénáře letadla. Nastavují se zde všechny atributy instance třídy `AircraftScenario`.

Třída `StandaloneMessageController`

Tato třída obsluhuje dialogové okno pro vytváření a editaci samostatných zpráv. Nastavují se zde všechny atributy instancí potomků třídy `StandaloneMessage`.

Třída `TransponderController`

Tato třída má na starost okno pro správu letadla. Dají se zde spravovat všechny atributy letadla a jeho respondérů (instance potomků třídy `Responder`), které lze měnit v průběhu simulace. Je zde možné měnit časování pro jednotlivé respondéry. Zároveň zde lze daný respondér zcela vypnout, což potlačí generování daného typu ADS-B zpráv pro dané letadlo.

4.2.7 Balíček `fxutils`

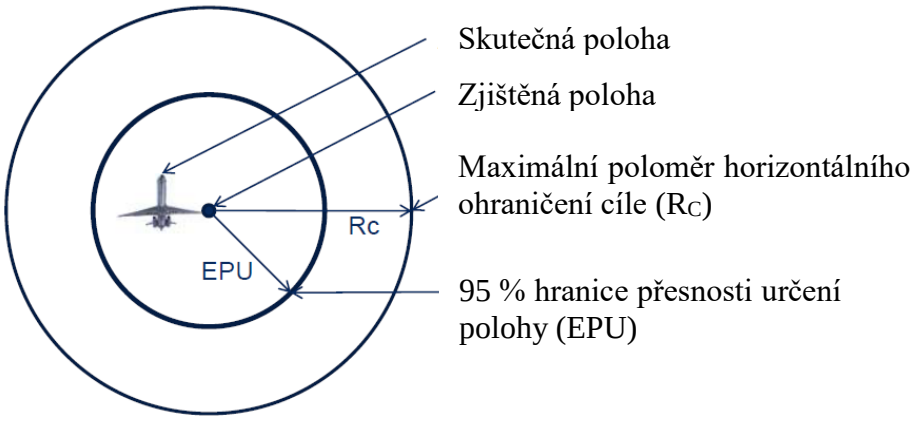
Tento balíček obsahuje pomocné třídy, které souvisejí s frameworkem JavaFX a zaměřují se především na uživatelské rozhraní. Pro fungování simulace nejsou nezbytné.

4.3 Princip generování chyby určení polohy a rychlosti

Informace získané ze systému GNSS jsou v reálném světě zatíženy určitou chybou. Tato chyba závisí především na počtu aktuálně dostupných satelitů, atmosférických podmínkách, ale i na systému GNSS který je využíván (GPS, Galileo, …).

4.3.1 Chyba určení polohy

Chyba určení polohy je udávána parametrem EPU (Estimated Position Uncertainty) také známý pod názvem HOFM (Horizontal Figure of Merit). Tento parametr je získán ze systému GNSS a udává poloměr kruhu v metrech, ve kterém se nachází skutečná poloha s pravděpodobností 95 %. Hodnota tohoto parametru je vždy nezáporná. Hodnota parametru EPU je polovinou poloměru horizontálního ohraničení (R_C) pro Navigation Integrity Category (viz. kapitola 3.5.1). Následující obrázek (Obrázek 34) znázorňuje vztah skutečné polohy a polohy zjištěné systémem GNSS společně s kruhovým ohraničením reprezentujícím parametry EPU a R_C.



Obrázek 34 – GNSS: Skutečná a zjištěná poloha
[zdroj: (ITT, 2009)]

(ITT, 2009)

4.3.2 Chyba určení rychlosti

Velikost chyby určení horizontální rychlosti je udávána parametrem Horizontal Velocity Error, který je měřen v metrech za sekundu. Skutečná horizontální rychlost se poté z 95 % pravděpodobností nachází v intervalu $\langle V_M - HVE, V_M + HVE \rangle$, kde V_M je zjištěná (naměřená) horizontální rychlost a HVE je hodnota parametru Horizontal Velocity Error. Hodnota tohoto parametru je vždy nezáporná.

4.3.3 Implementace generování chyb polohy a rychlosti

Jak je zřejmé s předchozích kapitol, určují jednotlivé parametry nepřesnosti interval, ve kterém se s pravděpodobností 95 % nachází skutečná hodnota dané veličiny, a to ať už se jedná o jednorozměrnou veličinu, jakou je například rychlost, nebo dvourozměrnou veličinu, jakou je poloha, kde je interval reprezentován kruhem o poloměru daném velikostí chyby.

Nabízí se tedy jako první možnost generovat náhodnou chybu podle Gaussova rozdělení se střední hodnotou $\mu = 0$ a směrodatnou odchylkou $\sigma = 0{,}5$, které s pravděpodobností 95 % vygeneruje náhodnou hodnotu v rozsahu od -1 do 1. Následně stačí vygenerovanou hodnotu vynásobit parametrem určujícím chybu dané veličiny.

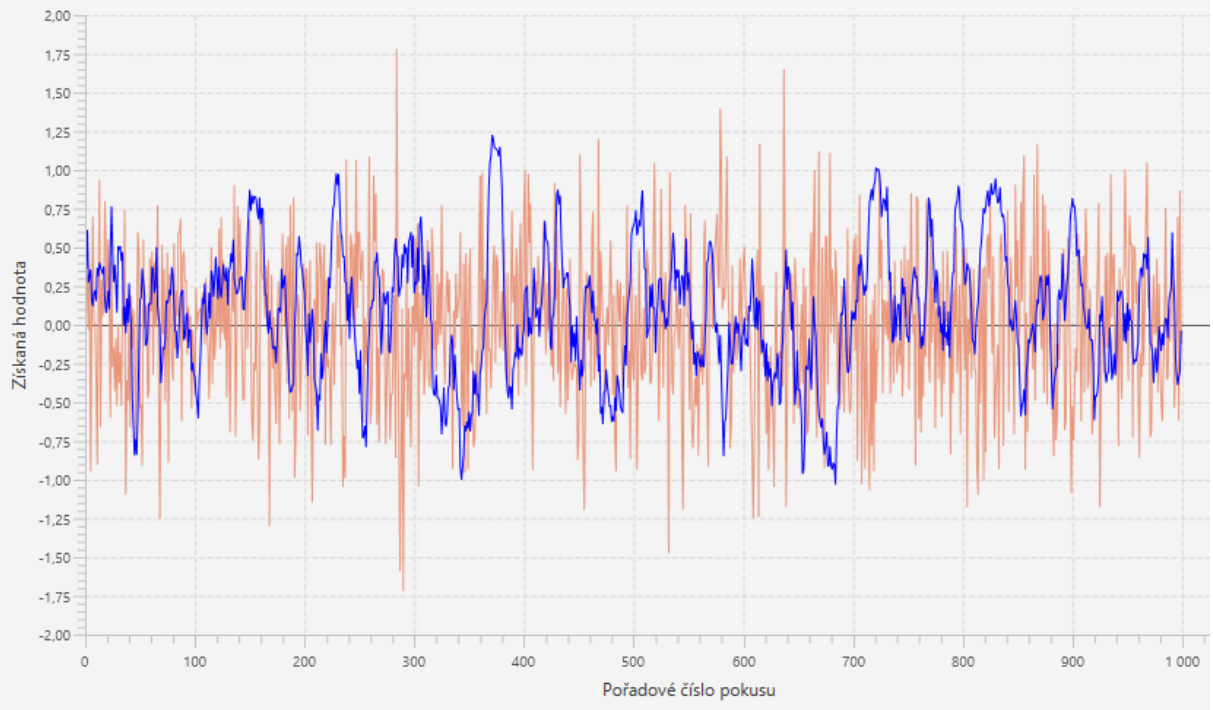
Při bližším pohledu na hodnoty naměřené reálnými přístroji (např. GPS přijímačem) však vyplývá, že hodnota naměřená v krátkém časovém intervalu od předchozího měření má velkou pravděpodobnost ležet v blízkosti hodnoty získané v předchozím měření. Chyba reálných měření tedy vykazuje jistou míru autokorelace v čase.

Pro tento účel byla implementována třída `AutocorrelatedGaussianGenerator` (generátor podle Gaussova rozdělení s autokorelací) v balíčku `utils`, která slouží pro generování náhodných hodnot závislých na hodnotách generovaných v minulosti. Při voláním metody `nextValue` je ve výchozím nastavení vygenerováno reálné číslo, které s pravděpodobností 95 % leží v intervalu od -1 do 1 a závisí na přechozích hodnotách vrácených touto metodou. Následně stačí vygenerovanou hodnotu vynásobit parametrem určujícím chybu dané veličiny.

Pro generování náhodné hodnoty je využita metoda `nextSkewedBoundedDouble`, která generuje náhodné hodnoty podle sešikmeného Gaussova rozdělen. Míra sešikmení je dána parametrem `bias`, do kterého je poslána hodnota vrácená předchozím voláním metody `nextValue`. Dále je na výslednou hodnotu vrácenou metodou `nextValue` aplikován klouzavý průměr, který má ve výchozím nastavení velikost posuvného okna 5. Za účelem

ustálení generovaných hodnot je při vytvoření instance této třídy do zásobníku před generováno tolik hodnot, jaká je nastavená velikost posuvného okna pro klouzavý průměr (parametr konstruktoru `slidingAvgWindowSize`).

Následující obrázek (Obrázek 35) znázorňuje průběh 1000 náhodně generovaných hodnot pomocí standardního Gaussova rozdělení $N(0;\ 0{,}5)$ bez autokorelace a pomocí instance třídy `AutocorrelatedGaussianGenerator` s autokorelací.



Obrázek 35 – Graf: 1000 náhodně vygenerovaných hodnot
Oranžová – Standardní Gaussovo rozdělení
Modrá – Implementovaný generátor s autokorelací
[zdroj: vlastní]

Pro získání informací se simulovanou chybou slouží specifické metody třídy `Aircraft` opatřené příponou „`WithError`“ v názvu metody. Každé volání takovéto metody vrátí hodnotu aktuální příslušného parametru zatíženou náhodnou chybou. Velikost této chyby je závislá na nastavení parametrů určujících maximální odchylku od skutečné hodnoty pro dané letadlo a na velikostech chyb získaných předcházejícím voláním dané metody.

Pro zanesení chyby polohy je využito 2 generátorů s autokorelací (2 instance třídy `AutocorrelatedGaussianGenerator`). Výstup 1. generátoru je násoben 180° a definuje směr nové (chybové) polohy oproti skutečné. Výstup 2. generátoru je násoben polovinou parametru R_C (viz. kapitola 3.5.1) nastaveném pro dané letadlo a určuje vzdálenost v metrech mezi novou (chybovou) polohou a skutečnou polohou. Nová (chybová) poloha je

poté vypočítána pomocí metody `calculateEndingGlobalCoordinates` z knihovny Geodesy, která na základě známé skutečné polohy, směru a vzdálenosti v metrech vrátí novou polohu se zanesenou chybou. Pro generování chyby polohy je nezbytné nastavit pro dané letadlo parametr R_C (Horizontal Radius of Containment) na nenulovou hodnotu (ve výchozím stavu je nulový).

Pro zanesení chyby horizontální rychlosti stačí 1 generátor s autokorelací. Výstup tohoto generátoru je násoben parametrem Horizontal Velocity Error (HVE) nastaveným pro dané letadlo a výsledkem je relativní chyba horizontální rychlosti v metrech za sekundu. Následně se daná relativní chyba přičte k přesné horizontální rychlosti získané ze simulátoru.

Pro chybu výšky jsou vytvořeny 2 generátory s autokorelací, a to zvlášť pro systém GNSS a pro barometrický výškoměr. Parametry maximální chyby určení výšky (s pravděpodobností 95%) pomocí systému GNSS a barometrického výškoměru lze nastavit ve scénáři daného letadla. Stejně jako v předchozím případě je výstup daného generátoru vynásoben příslušným parametrem maximální chyby a výsledkem je relativní chyba výšky v metrech. Ta je následně přičtena k přesnému údaji nadmořské výšky získané ze simulátoru. V případě, že je jako zdroj informace o výšce pro dané letadlo nastaven barometrický výškoměr, je rozdíl obou chybou zatížených údajů použit pro pole DIFF ve zprávě o rychlosti ve vzduchu (viz. kapitola 3.6.6).

5 ZÁVĚR

V teoretické části práce bylo provedeno porovnání primárních a sekundárních radarů. Tato poměrně rozsáhlá část byla zpracována na žádost vedoucího diplomové práce, a to z důvodu seznámení se s problematikou radiolokace, která nebyla naplní studia informačních technologií.

Byly zde podrobněji představeny principy sekundární radiolokace, především potom mód S, který je využíván systémem ADS-B. V následující kapitole byl podrobněji představen systém ADS-B, na který se zaměřuje softwarový simulátor realizovaný v praktické části práce.

Předpokládá se, že systém ADS-B jednou nahradí radar jako primární způsob řízení letového provozu. Systém ADS-B poskytuje oproti klasickému radaru řadu výhod, jakou je například schopnost ostatních letadel disponujících transpondérem ADS-B zachytávat zprávy vysílané ostatními letadly a mít tak přehled o okolním provozu bez nutnosti spoléhat se na pozemní řídící centra. To výrazně zvyšuje výkonost systému pro předcházení kolizím (TCAS) a obecně přispívá k bezpečnosti letového provozu.

Systém ADS-B má však i své slabiny. Tou nejvýznamnější slabinou je absence jakýchkoliv kryptografických a autentizačních mechanismů pro zajištění důvěrnosti vysílaných zpráv. To činí tento systém zranitelným vůči cíleným útokům podrobněji popsaných v kapitole 3.8. Tento nedostatek byl jednou z hlavních motivací pro implementaci simulátoru.

V praktické části práce byl implementován prototyp softwarového emulátoru transpondéru ADS-B na platformě Java. Při vývoji jádra simulátoru byly dodržovány principy OOP s důrazem na snadnou rozšiřitelnost o další typy ADS-B zpráv. V současné verzi simulátor podporuje ADS-B zprávy pro identifikaci, polohu ve vzduchu a rychlost ve vzduchu. Pro obsluhu simulátoru bylo vytvořeno grafické uživatelské rozhraní pomocí frameworku JavaFX.

Vstupem simulátoru je scénář, který obsahuje množinu definovaných letadel společně s jejich trasami pohybu a množinu samostatných zpráv. Simulátor tedy podporuje 2 typy entit. První jsou letadla, která se během simulace pohybují po zadaných trajektoriích a generují příslušné ADS-B zprávy. Druhou jsou samostatné zprávy, které jednorázově generují ADS-B zprávu podle zadání. Scénář je možné exportovat jako soubor ve formátu XML.

Výstupem simulátoru jsou jednotlivé ADS-B zprávy jako binární data ve formátu popsaném v kapitole 3. V současné verzi lze výstupní zprávy zobrazit, uložit jako textový nebo XML soubor a zasílat ostatním aplikacím pomocí protokolu TCP.

Možné cíle pro další vývoj:

- Doplnit podporu zbývajících typů ADS-B zpráv.
- Implementovat realističtější pohyb simulovaných letadel po zadaných trasách.
- Možnost simulace pozemních cílů (letadla na zemi).
- Vizuální zadávání trasy letadla pomocí mapy.

6 POUŽITÁ LITERATURA

ARLOW, Jim a NEUSTADT, Ila. 2007. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky.* [překl.] Bogdan Kiszka. Brno : Computer Press, 2007. str. 567. ISBN 978-80-251-1503-9.

BEZOUŠEK, Pavel a ŠEDIVÝ, Pavel. 2004. *Radarová technika.* Praha : Vydavatelství ČVUT, 2004. ISBN 978-80-01-03036-3.

BINSTOCK, Andrew. 2015. Java's 20 Years Of Innovation. *Forbes.* [Online] 20. 5 2015. [Citace: 6. 9 2017.] https://www.forbes.com/sites/oracle/2015/05/20/javas-20-years-of-innovation.

BRAY, Tim, a další. 2008. Extensible Markup Language (XML) 1.0 (Fifth Edition). *W3C Recommendation.* [Online] 26. 11 2008. [Citace: 22. 9 2017.] https://www.w3.org/TR/xml/.

BROWN, Greg. 2011. Introducing FXML: A Markup Language for JavaFX. *FX Experience.* [Online] 15. 8 2011. [Citace: 26. 4 2015.] http://fxexperience.com/wp-content/uploads/2011/08/Introducing-FXML.pdf.

BYOUS, Jon. 1998. JAVA TECHNOLOGY: THE EARLY YEARS. *Wayback Machine.* [Online] 1998. [Citace: 6. 9 2017.] http://gcc.uni-paderborn.de/www/wi/wi2/wi2_lit.nsf/0/abf8d70f07c12eb3c1256de900638899/$FILE/Java%20Technology%20-%20An%20early%20history.pdf.

DHEERAJ, Thedijje. © 2015. HOW GPS WORKS. *INTELLIGENT COMPUTING.* [Online] © 2015. [Citace: 3. 5 2018.] https://www.inteligentcomp.com/2012/02/how-gps-works.html.

EUROCAE. 2009. *Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B).* Malakoff, France : EUROCAE, 2009.

EUROCONTROL. © 2001-2011. Mode S Technical Overview. *Wayback Machine.* [Online] © 2001-2011. [Citace: 17. 7 2017.] https://web.archive.org/web/20130331041052/http://www.eurocontrol.int/msa/public/standard_page/modes_tech_overview.html.

GARMIN. © 1996-2017. What do you get by extending your squit? *ADS-B Academy.* [Online] © 1996-2017. [Citace: 14. 7 2017.] https://www.garmin.com/us/intheair/ads-b/squit.

ICAO. 2011. Aircraft Addressing System. *KLOTH.NET.* [Online] 23. 6 2011. [Citace: 17. 7 2017.] http://www.kloth.net/radio/icao24alloc.php.

—. 2007. Surveillance and Collision Avoidance Systems. *Aeronautical Telecommunications.* 4. Montreal : International Civil Aviation Organization, 2007, Sv. 4.

ITT. 2009. ADS-B Deployment in the United States. *ION Rocky Mountain Section.* [Online] 11. 12 2009. [Citace: 21. 7 2017.] http://rms.ion.org/wp-content/uploads/2010/01/ADS-B_Briefing-20091210.pdf.

LECHNER, Wolfgang a BAUMANN, Stefan. © 2000. Global navigation satellite systems. *Computers and Electronics in Agriculture.* Linden, Germany : Elsevier Science B.V., © 2000.

MARSHALL, Al. 2009. An Expanded Description of the CPR Algorithm. East Syracuse, New York : Sensis Corporation, 2009.

Oracle. © 2008. JavaFX CSS Reference Guide. *Oracle.* [Online] © 2008. [Citace: 6. 9 2017.] https://docs.oracle.com/javafx/2/api/javafx/scene/doc-files/cssref.html.

—. © 1994. Understanding Memory Management. *Oracle.* [Online] © 1994. [Citace: 6. 9 2017.] https://docs.oracle.com/cd/E13150_01/jrockit_jvm/jrockit/geninfo/diagnos/garbage_collect.html.

ROUSE, Margaret. 2005. just-in-time compiler (JIT). *TheServerSide.* [Online] Květen 2005. [Citace: 6. 9 2017.] http://www.theserverside.com/definition/just-in-time-compiler-JIT.

SCHÄFER, Matthias, LENDERS, Vincent a MARTINOVIC, Ivan. 2013. Experimental Analysis of Attacks on Next Generation Air Traffic Communication. *Applied Cryptography and Network Security.* Berlin : Springer, 2013. Sv. 7954. ISBN: 978-3-642-38979-5.

SUN, Junzi. © 2017. *ADS-B Decoding Guide.* [Online] TuDelft, © 2017. [Citace: 15. 7 2017.] http://adsb-decode-guide.readthedocs.io/en/latest/.

TECH ANNOUNCER. 2017. Global Medium Range Surveillance Radar Sales Market 2017 – Thales, Airbus, IAI, SRC, CETC. *Tech Announcer - Technology News and Reviews.* [Online] 31. 8 2017. [Citace: 3. 5 2018.] https://techannouncer.com/global-medium-range-surveillance-radar-sales-market-2017-thales-airbus-iai-src-cetc/.

7 PŘÍLOHY

Příloha A – Tabulka znaků a jejich kódů pro volací znak identifikační ADS-B zprávy.....100

Příloha B – Tabulka kódů pro pole Emitter Category (EC) 101

Příloha C – Návod k obsluze 102

Příloha A – *Tabulka znaků a jejich kódů pro volací znak identifikační ADS-B zprávy*

Následující tabulka obsahuje 6 bitové kódy pro každý znak použitelný v ADS-B zprávě pro identifikaci letadla (volací znak). Volací znak letadla / letu se skládá přesně z 8 znaků uvedených v následující tabulce.

Tabulka 25 – Znaky a kódy pro volací znak identifikační ADS-B zprávy

Znak	Kód	
	Dekadicky	Binárně (6 bitů)
A	1	000001
B	2	000010
C	3	000011
D	4	000100
E	5	000101
F	6	000110
G	7	000111
H	8	001000
I	9	001001
J	10	001010
K	11	001011
L	12	001100
M	13	001101
N	14	001110
O	15	001111
P	16	010000
Q	17	010001
R	18	010010
S	19	010011
T	20	010100
U	21	010101
V	22	010110
W	23	010111
X	24	011000
Y	25	011001
Z	26	011010
0	48	110000
1	49	110001
2	50	110010
3	51	110011
4	52	110100
5	53	110101
6	54	110110
7	55	110111
8	56	111000
9	57	111001
_	32	100000

Příloha B – *Tabulka kódů pro pole Emitter Category (EC)*

Jednotlivé „sety“ jsou identifikovány hodnotou v poli TC (Type Code)

Tabulka 26 – Kódy pro pole Emitter Category (EC)

Set	TC	EC	Význam
A	4	0	Informace o kategorii nedostupná
		1	Lehký stroj (<7030 *kg*)
		2	Malý stroj (7030 až 34019 *kg*)
		3	Velký stroj (34019 až 136077 *kg*)
		4	Velký stroj, vysoký vortex (např. B-757)
		5	Těžký stroj (>136077 *kg*)
		6	Vysokovýkonný stroj (>5 g akcelerace a rychlost > 400 uzlů)
		7	Rotorcraft (např. helikoptéra)
B	3	0	Informace o kategorii nedostupná
		1	Kluzák
		2	Lehčí než vzduch (např. balón, vzducholoď)
		3	Parašutista
		4	Ultralight / hang-glider / paraglider
		5	Rezervováno
		6	Bezpilotní letoun (UAV)
		7	Trans atmosférické vozidlo (např. raketoplán)
C	2	0	Informace o kategorii nedostupná
		1	Pozemní záchranné vozidlo
		2	Pozemní servisní vozidlo
		3	Bodová překážka (např. uvázaný balón)
		4	Shluková překážka
		5	Liniová překážka
		6–7	Rezervováno
D	1	0	Informace o kategorii nedostupná
		1–7	Rezervováno

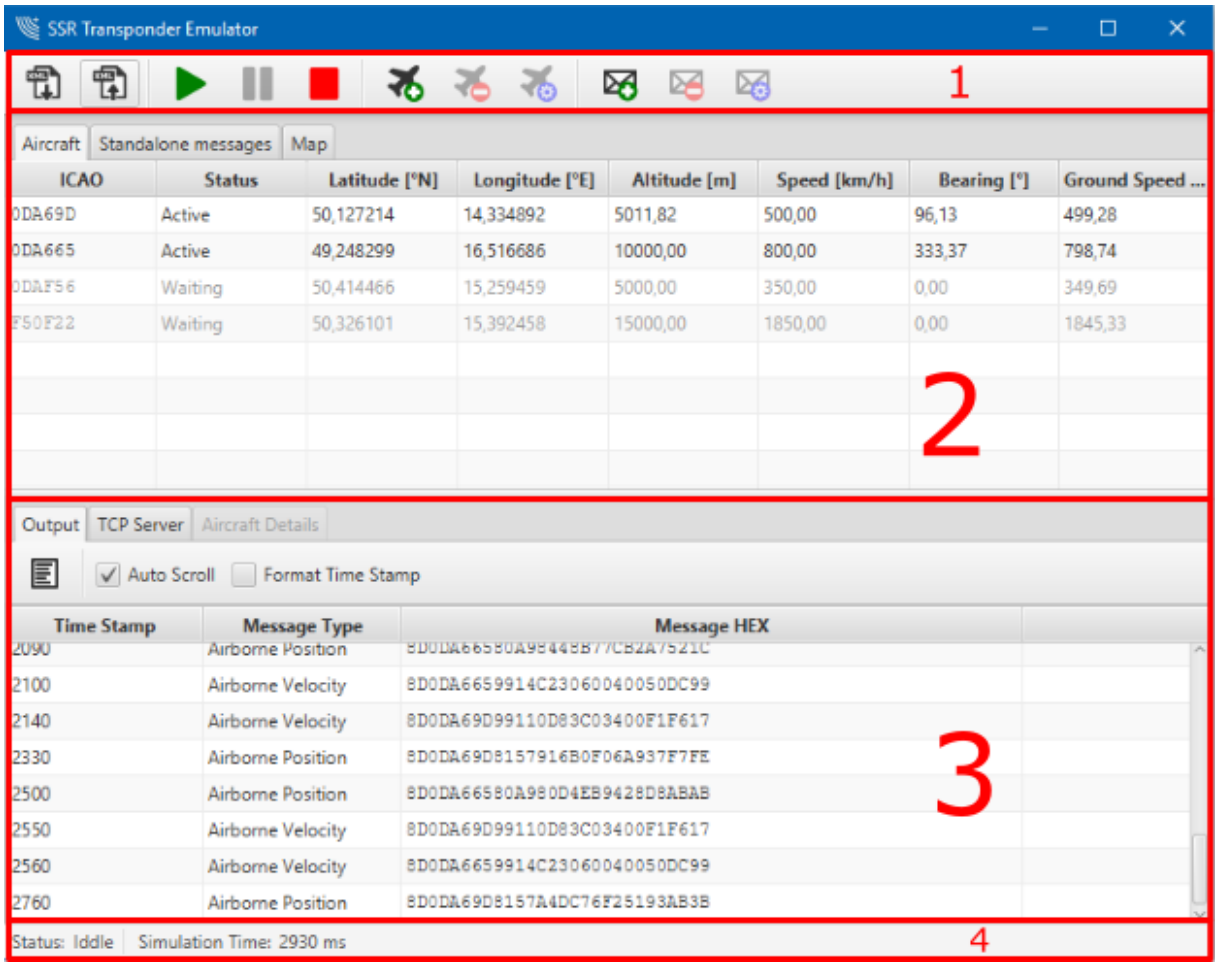
(EUROCAE, 2009 str. 84)

Příloha C – *Návod k obsluze*

Aplikace simulátoru vyžaduje nainstalované prostředí Java Runtime Environment (JRE) ve verzi alespoň 8u121 (1.8_121) nebo vyšší. Dále se doporučuje obrazovka alespoň s rozlišením 800 na 600 pixelů a 4 GB operační paměti.

Uživatelské rozhraní – Hlavní okno

Výchozí okno, které se zobrazí po spuštění aplikace simulátoru.



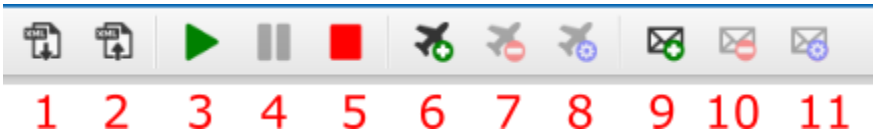
Obrázek 36 – Hlavní okno
[zdroj: vlastní]

1. Hlavní nástrojová lišta poskytuje základní nástroje pro ovládání simulátoru.
2. Panel entit zobrazuje veškerá letadla a naplánované samostatné zprávy ve scénáři a jejich aktuální stav během simulace. Dále je zde záložka pro zobrazení mapy.
3. Panel výstupů a detailů.
4. Stavový řádek zobrazuje aktuální simulační čas a stav simulátoru.

Jednotlivé panely jsou podrobněji popsány v následujících kapitolách.

Hlavní nástrojová lišta

Hlavní nástrojová lišta poskytuje základní nástroje pro ovládání simulátoru.



Obrázek 37 – Hlavní nástrojová lišta
[zdroj: vlastní]

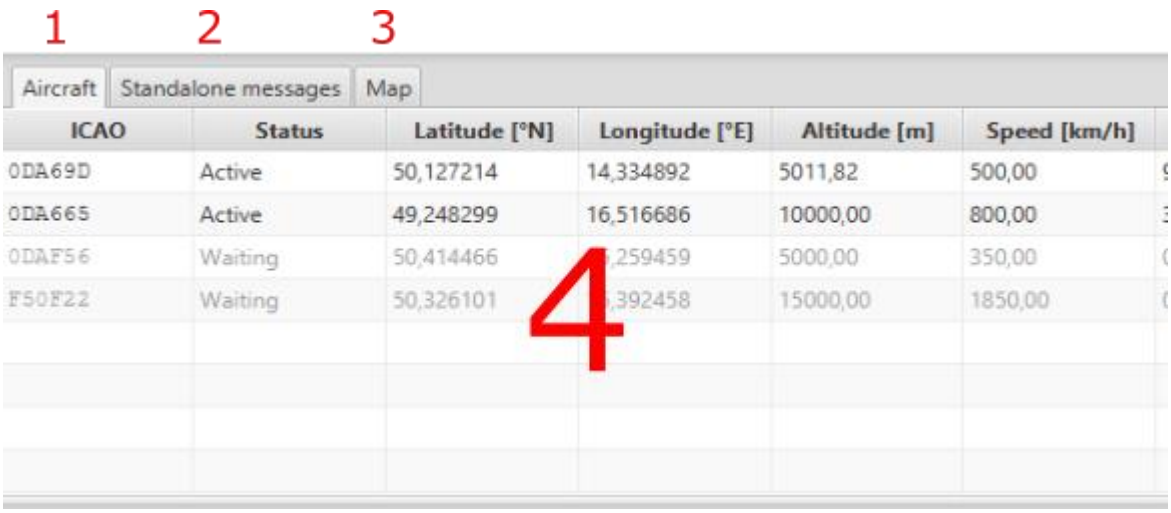
1. Načíst (importovat) simulační scénář ze souboru XML.
2. Uložit (exportovat) simulační scénář do souboru XML.
3. Spustí běh simulace.
4. Pozastaví běh simulace.
5. Resetuje simulaci na počáteční stav. Vyprázdní okno výstupu.
6. Vyvolá dialog pro přidání nového letadla do scénáře simulátoru.
7. Odebere vybrané letadlo ze simulátoru. Lze použít i klávesu DELETE.
8. Vyvolá dialog pro editaci vybraného letadla a jeho scénáře. Lze využít i klávesu ENTER nebo dvojité kliknutí na vybrané letadlo v tabulce letadel.
9. Vyvolá dialog pro přidání nové samostatné zprávy do scénáře simulátoru.
10. Odebere vybranou samostatnou zprávu ze simulátoru. Lze použít i klávesu DELETE.
11. Vyvolá dialog pro editaci vybrané zprávy. Lze využít i klávesu ENTER nebo dvojité kliknutí na vybranou zprávu v tabulce samostatných zpráv.

Panel entit

Zobrazuje všechny entity (letadla a samostatné zprávy) v simulátoru. Pomocí tohoto rozhraní lze přidávat, editovat a mazat jednotlivé entity. Pro letadla dále zobrazuje aktuální informace o jejich poloze, rychlosti atd. které se aktualizují v průběhu simulace.

Některé detaily lze skrýt pomocí kontextového menu, které lze vyvolat stisknutím pravého tlačítka myši nad tabulkou. Pomocí kontextového menu lze také duplikovat vybrané letadlo nebo samostatnou zprávu.

Dále je zde záložka pro zobrazení mapy, na které se zobrazí simulovaná letadla. Zobrazují se pouze aktivní letadla a identifikují se pomocí ICAO adres. Mapa je poskytována službou Google Maps a ke svému zobrazení vyžaduje připojení k Internetu.



Obrázek 38 – Panel entit
[zdroj: vlastní]

1. Záložka letadel. Zobrazí tabulku všech letadel v simulátoru a jejich stav během simulace.
2. Záložka samostatných zpráv. Zobrazí tabulku všech naplánovaných samostatných zpráv v simulátoru.
3. Záložka mapy. Na mapě se zobrazují všechna aktivní letadla v průběhu simulace. (Zobrazení mapy vyžaduje připojení k Internetu.)
4. Aktuální záložka zobrazuje tabulku všech letadel v simulátoru.

Panel výstupů a detailů

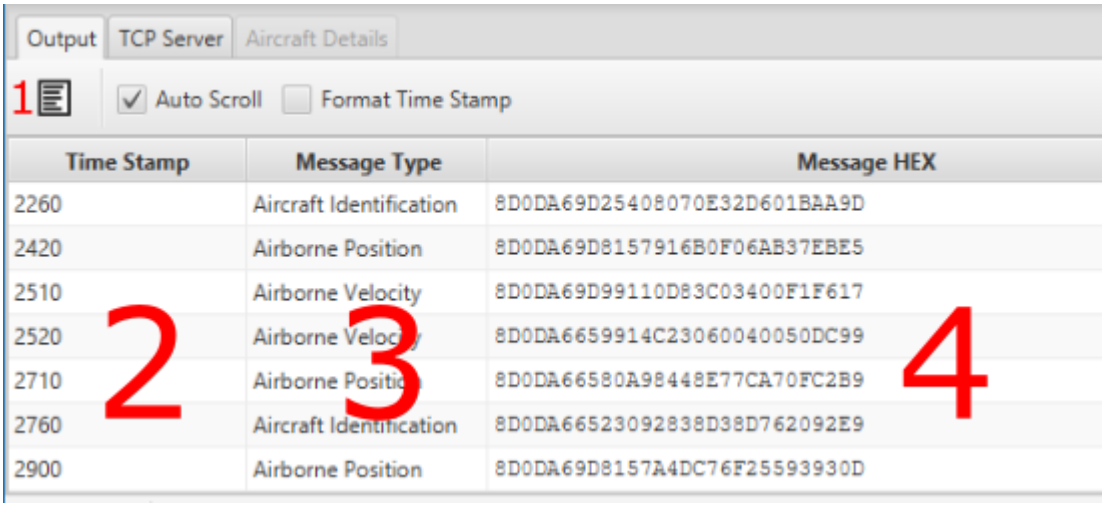
Panel se nachází ve spodní části hlavního okna a obsahuje následující záložky:

- Output – Zobrazí panel výstupu.
- TCP Server – Zobrazí panel TCP serveru.
- Aircraft Details – Zobrazí panel podrobností o letadle.

Jednotlivé panely záložek jsou podrobněji popsány v následujících kapitolách.

Panel výstupu

Zobrazuje ADS-B zprávy tak, jak jsou generovány v průběhu simulace. Pomocí kontextového menu nad vybranou zprávou lze její obsah v hexadecimální podobě zkopírovat do schránky.

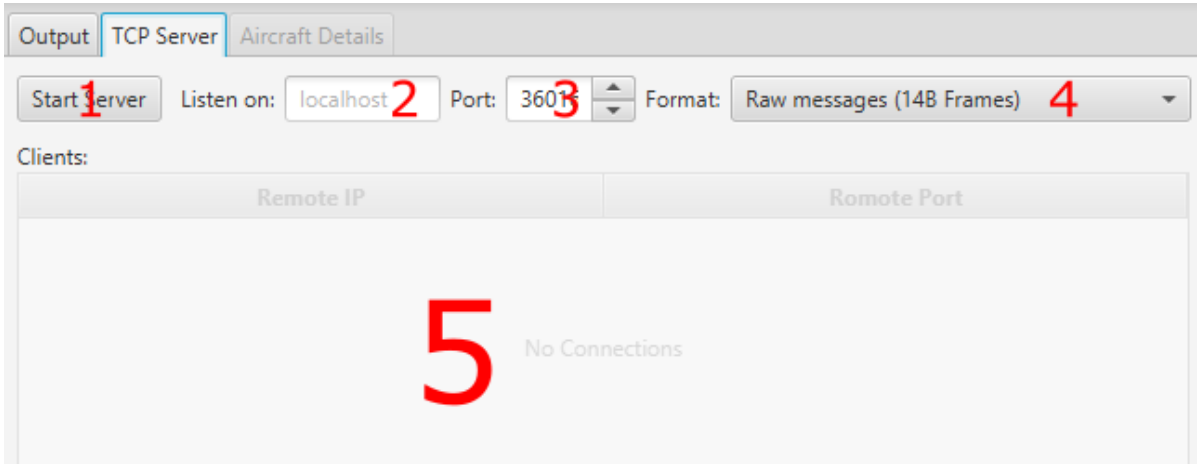


Obrázek 39 – Panel výstupu
[zdroj: vlastní]

1. Uloží ADS-B zprávy na výstupu do textového souboru. Možnost uložit ve formátu XML.
2. Časové razítko značí simulační čas, ve kterém byla daná zpráva vyslána.
3. Sloupec označuje typ ADS-B zprávy.
4. Sloupec obsahuje ADS-B zprávu v hexadecimálním formátu.

Panel TCP serveru

Tento panel poskytuje rozhraní pro obsluhu TCP serveru. Pokud je spuštěn, jsou ADS-B zprávy vysílány všem připojeným klientům tak, jak jsou generovány simulátorem. Jednotlivé zprávy jsou vysílány jako pole 14 bytů nebo ve formátu XML v textové podobě s kódováním UTF-8.

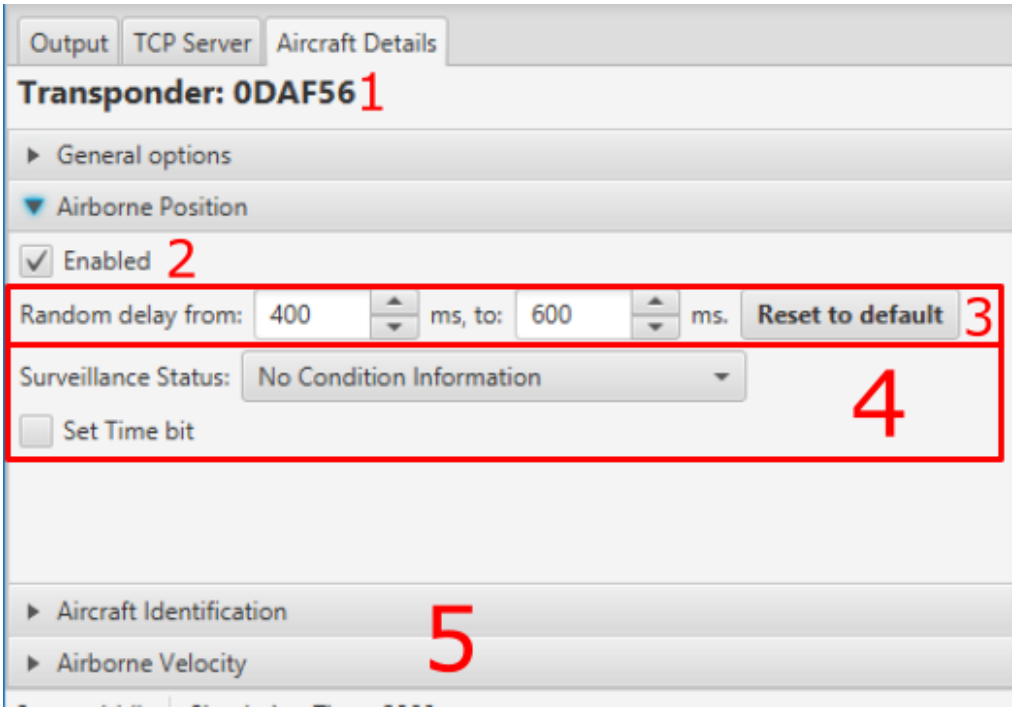


Obrázek 40 – Panel TCP serveru
[zdroj: vlastní]

1. Spustí nebo zastaví TCP server.
2. Adresa, na které má server naslouchat. Lze využít doménové jméno, IPv4 nebo IPv6 adresu. Ve výchozím stavu je použito doménové jméno „localhost“.
3. Port, na kterém má server naslouchat.
4. Výběr výstupního formátu (pole bytů nebo XML). Formát XML je podrobněji popsán v kapitole 4.2.4 – Třída `Response`. Jednotlivé zprávy jsou od sebe odděleny znakem nového řádku „`\n`“.
5. Zde se zobrazují aktuálně připojení klienti. Pomocí kontextového menu nad vybraným klientem lze dané spojení ukončit.

Panel podrobností o letadle

Tento panel je dostupný pouze je-li v tabulce letadel vybráno letadlo a slouží k nastavování parametrů transpondéru daného letadla, a to i za běhu simulace. Záložka „General options“ obsahuje společné parametry pro dané letadlo, které lze měnit v průběhu simulace. Zbylé záložky obsahují nastavení pro specifické typy ADS-B zpráv. V každé této záložce lze najít checkbox „Enabled“, pomocí kterého lze potlačit generování daného typu ADS-B zpráv pro dané letadlo. Dále zde lze změnit časování daného typu ADS-B zpráv změnou minimální a maximální prodlevy v milisekundách mezi vysíláním zpráv daného typu.



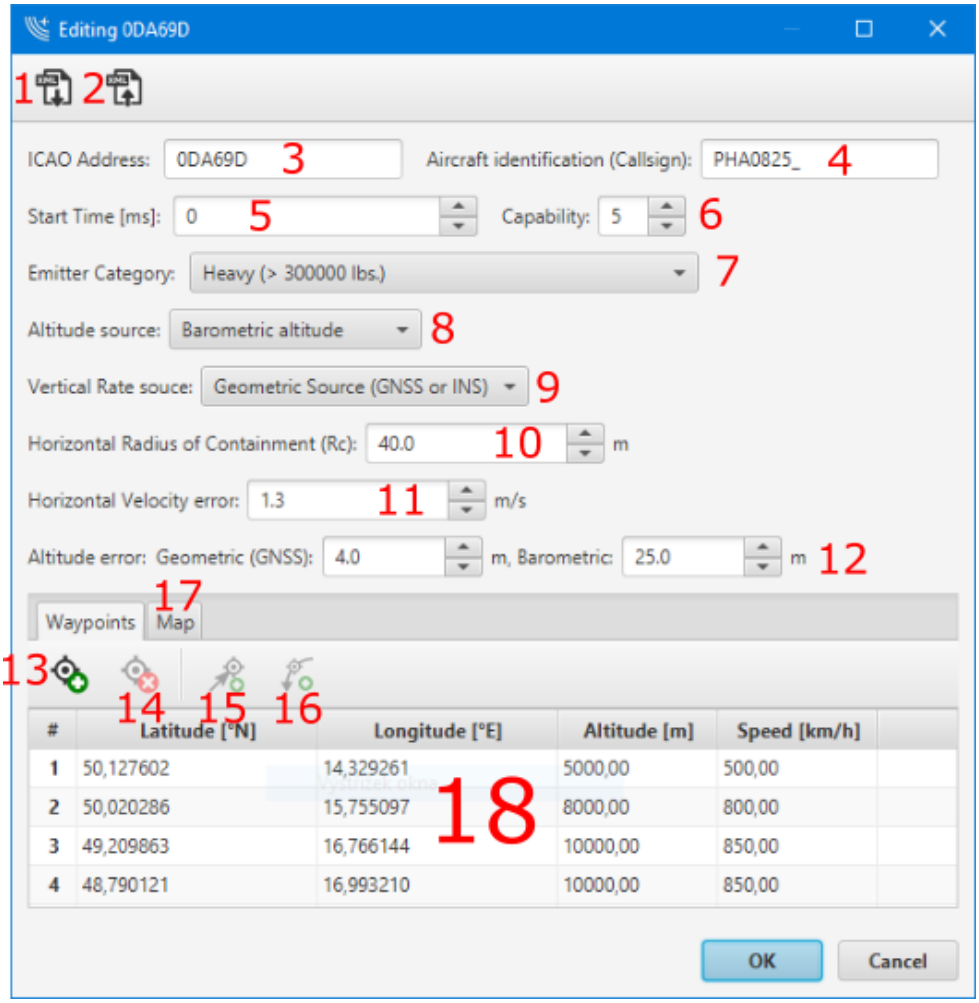
Obrázek 41 – Panel podrobností o letadle
[zdroj: vlastní]

1. ICAO adresa daného transpondéru v hexadecimální podobě.

2. Pokud není vybráno, způsobí potlačení generování daného typu ADS-B zpráv pro dané letadlo.
3. Změna časování (minimální a maximální prodlevy mezi zprávami) pro daný typ ADS-B zpráv.
4. Nastavení specifická pro ADS-B zprávy o poloze. (Liší se pro ostatní typy ADS-B zpráv.)
5. Záložky obsahují nastavení pro ostatní podporované typy ADS-B zpráv.

Dialog pro editaci letadla

Toto dialogové okno slouží pro editaci parametrů a trasy existujícího, nebo definování nového letadla. Pro potvrzení změn je třeba, aby všechna pole obsahovala validní hodnoty daných parametrů a následně potvrdit dialogové okno tlačítkem OK.



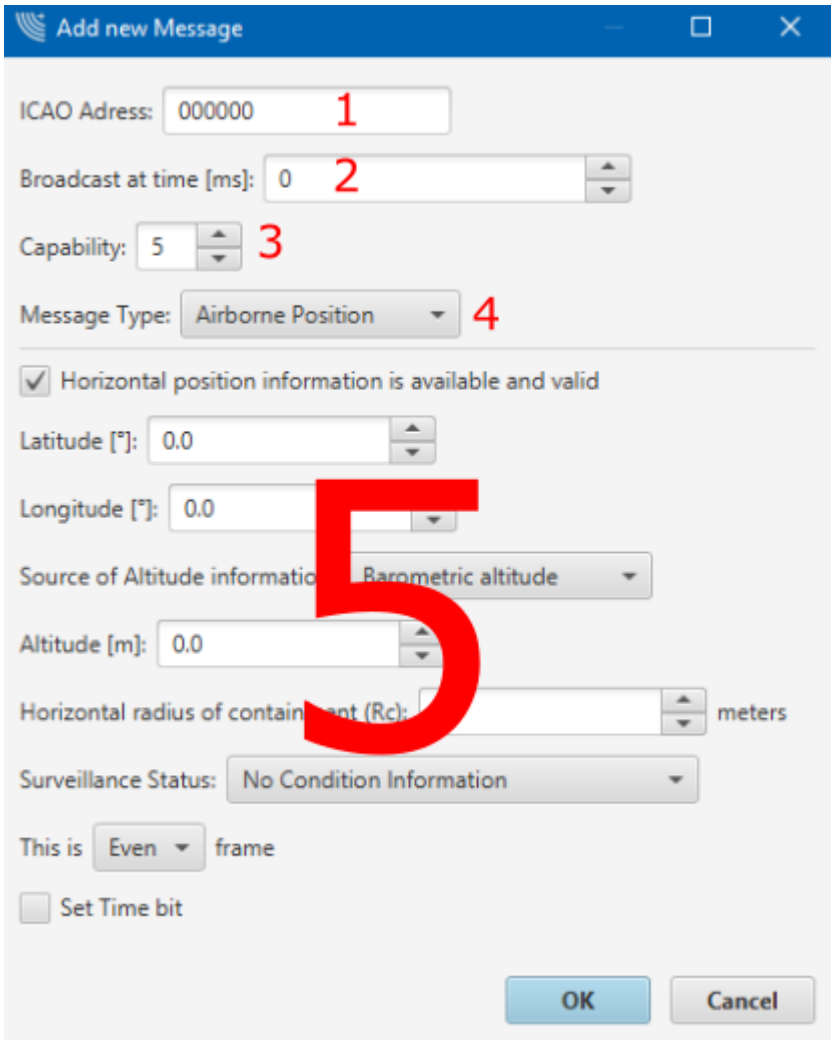
Obrázek 42 – Dialogové okno pro editaci letadla
[zdroj: vlastní]

1. Načíst (importovat) scénář letadla ze souboru XML.
2. Uložit (exportovat) scénář letadla do souboru XML.
3. ICAO adresa v hexadecimální podobě. Velká a malá písmena se nerozlišují.

4. Volací znak daného letadla. Musí mít přesně 8 znaků a může obsahovat pouze znaky definované v tabulce přílohy A (Tabulka 25). Velká a malá písmena se nerozlišují.
5. Simulační čas v milisekundách, ve kterém letadlo začne vykonávat zadanou trasu a vysílat ADS-B zprávy.
6. Nastavení parametru „Capability“ pro vysílané ADS-B zprávy.
7. Nastavení parametru „Emitter Category“ pro vysílané ADS-B zprávy.
8. Nastavení zdroje informace o výšce.
9. Nastavení zdroje informace o vertikální rychlosti.
10. Nastavení maximálního poloměru horizontálního ohraničení (R_c) pro dané letadlo.
11. Nastavení maximální chyby měření horizontální rychlosti pro dané letadlo.
12. Nastavení maximální chyby určení výšky (s pravděpodobností 95%) pro systém GNSS a barometrický výškoměr daného letadla.
13. Přidat nový bod trasy letadla.
14. Odebrat vybraný bod trasy letadla.
15. Vygenerovat nový bod trasy na základě zadání směru, vzdálenosti a úhlu stoupání, popř. klesání. Nejprve je třeba vybrat referenční bod trasy z již vytvořených. Nový bod bude přidán jako následující vybranému referenčnímu bodu.
16. Nástroj pro vytvoření pozvolného obratu. Na základě zadání úhlu obratu a poloměru vytvoří aproximaci obratu pomocí zadaného počtu bodů trasy. Nejprve je třeba vybrat referenční bod trasy z již vytvořených. Nové body trasy budou přidány jako následující vybranému referenčnímu bodu.
17. Zobrazí trasu letadla na mapě. (Vyžaduje připojení k Internetu.)
18. Tabulka bodů trasy letadla. Jednotlivé hodnoty lze editovat dvojitým kliknutím na danou buňku. Editaci buňky je možné zrušit stisknutím klávesy Escape.
    a. Latitude – Zeměpisná šířka ve stupních od -90 do 90.
    b. Longitude – Zeměpisná délka ve stupních od -180 do 180.
    c. Altitude – Nadmořská výška v metrech.
    d. Speed – Rychlost pohybu z daného bodu v kilometrech za hodinu.

Dialog pro editaci samostatných zpráv

Toto dialogové okno slouží pro editaci parametrů nebo definici nové samostatné zprávy. Pro potvrzení změn je třeba, aby všechna pole obsahovala validní hodnoty daných parametrů a následně potvrdit dialogové okno tlačítkem OK.



Obrázek 43 – Dialogové okno pro editaci samostatných zpráv
[zdroj: vlastní]

1. ICAO adresa v hexadecimální podobě. Velká a malá písmena se nerozlišují.
2. Simulační čas v milisekundách, ve kterém bude vygenerována ADS-B zpráva podle zadání.
3. Nastavení parametru „Capability“ pro danou zprávu.
4. Typ vygenerované ADS-B zprávy.
5. Spodní část dialogového okna slouží pro nastavení specifických parametrů pro daný typ zprávy. Ovládací prvky ve spodní části se liší podle nastaveného typu ADS-B zprávy.