

**ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA ELEKTROTECHNICKÁ**

**KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACE**

# **BAKALÁŘSKÁ PRÁCE**

**Aplikace pro vzdálenou konfiguraci a ovládání polohy  
satelitních antén na palubě lodí**

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
Fakulta elektrotechnická  
Akademický rok: 2016/2017

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lev SHASHORIN**  
Osobní číslo: **E14B0163P**  
Studijní program: **B2612 Elektrotechnika a informatika**  
Studijní obor: **Elektronika a telekomunikace**  
Název tématu: **Aplikace pro vzdálenou konfiguraci a ovládání polohy satelitních antén na palubě lodí**  
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

**Z á s a d y p r o v y p r a c o v á n í :**

1. Prostudujte technologii satelitních komunikačních systémů pro pohyblivé dopravní prostředky.
2. Prostudujte možnosti a dostupné aplikace pro vzdálené ovládání satelitních systémů.
3. Vyberte vhodný komunikační protokol a navrhnete vhodnou strukturu aplikace pro vzdálené ovládání satelitního systému.
4. Vytvořte grafickou řídicí aplikaci na zvolené platformě.
5. Ověřte funkci Vámi vytvořené aplikace, zhodnoťte její přínos a porovnejte s dostupnými komerčními aplikacemi.

Rozsah grafických prací: podle doporučení vedoucího

Rozsah kvalifikační práce: 30 - 40 stran

Forma zpracování bakalářské práce: tištěná/elektronická

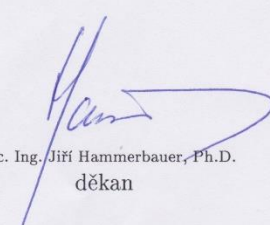
Seznam odborné literatury:

1. Fuchs, Karl. Antenna Integration: The Open-AMIP Standard For SATCOM [online]. MilsatMagazine (c) 2013 [cit. 11.4.2016] Dostupné z: <http://www.milsatmagazine.com/story.php?number=1483485170>
2. Lyova Satcom & Wireless Solutions [online] (c) 2016 [cit. 11.4.2016] Dostupné z: <http://www.lyova.eu/home/index.html>
3. Navisystem [online] (c) 2016 [cit. 11.4.2016] Dostupné z: <http://www.navisystem.it/eng/prodotti.html>
4. Windows Dev Center [online] (c) 2016 [cit. 11.4.2016] Dostupné z: <https://dev.windows.com>
5. Developer Centrum Xamarin [online] (c) 2016 [cit. 11.4.2016] Dostupné z: <http://developer.xamarin.com>

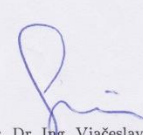
Vedoucí bakalářské práce: **Ing. Petr Kropík, Ph.D.**  
Katedra teoretické elektrotechniky

Datum zadání bakalářské práce: **14. října 2016**

Termín odevzdání bakalářské práce: **8. června 2017**

  
Doc. Ing. Jiří Hammerbauer, Ph.D.  
děkan



  
Doc. Dr. Ing. Vjačeslav Georgiev  
vedoucí katedry

V Plzni dne 14. října 2016

## **Abstrakt**

Cílem této práce je navrhnout takový software, který umožní dálkové ovládaní datovými komunikačními prostředky přes satelitní antény, pro jednoduchou údržbu systému a navazování telekomunikačních spojení i za nepříznivých podmínek. Software je napsán v programovacím jazyce Java a celá komunikace v síti probíhá pomocí protokolu openAMIP.

## **Klíčová slova**

Java, SQL, VSAT, protokol OpenAMIP, VSAT anténa, satelitní modem.

## **Abstract**

The aim of this work is to design software that allows remote control via data communication means over satellite antennas, for easy system maintenance and establishing telecommunication connections even under unfavorable conditions. The software is written in the Java programming language and the whole network communication is running via openAMIP protocol.

## **Key words**

Java, SQL, VSAT, protocol OpenAMIP, VSAT antenna, satellite modem.

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské/diplomové práce, je legální.

.....

podpis

V Plzni dne 8.6.2017

Lev Shashorin

## **Poděkování**

Tímto bych rád poděkoval firmě LYOVA s.r.o. za ideu této práce a poskytnuti potřebných zařízení a nástrojů.

# Obsah

<b>OBSAH</b> .....	<b>8</b>
<b>SEZNAM SYMBOLŮ A ZKRATEK</b> .....	<b>9</b>
<b>ÚVOD</b> .....	<b>10</b>
<b>1 HARDWARE</b> .....	<b>11</b>
1.1 SATELITE ROUTER.....	11
1.2 ANTÉNA.....	13
1.2.1 VSAT.....	13
1.2.2 Námořní VSAT.....	14
1.3 OPENAMIP.....	16
<b>2 SOFTWARE</b> .....	<b>18</b>
2.1 JAVA.....	18
2.2 NÁVRH APLIKACE.....	20
2.3 KNIHOVNY.....	22
2.3.1 SQL.....	22
2.3.2 Serial Port.....	23
2.4 ZABEZPEČENÍ HESLA.....	25
2.4.1 SHA.....	25
2.4.2 Salt.....	26
<b>3 TESTOVÁNÍ</b> .....	<b>28</b>
<b>4 DALŠÍ MOŽNOSTI VÝVOJE</b> .....	<b>31</b>
<b>ZÁVĚR</b> .....	<b>32</b>
<b>SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ</b> .....	<b>33</b>
<b>PŘÍLOHY</b> .....	<b>1</b>
<b>OBSAH PŘILOŽENÉHO CD</b> .....	<b>1</b>



## **Seznam symbolů a zkratek**

ACU.....	Antena control unit
AMIP .....	Antenna to Modem Interface Protocol
BUC .....	Block upconverter
IDU .....	In Door Unit
LNB .....	Low-noise block converter
MD.....	Message Digest
ODU .....	Out Door Unit
RX.....	Receive Data
SHA .....	Secure Hash Algorithm
TDMA .....	Time Division Multiple Access
TX.....	Transmit Data
VSAT.....	Very Small Aperture Terminal
VPN .....	Virtual private network

## Úvod

V dnešní době vybudovat síť na bázi VSAT není problém. Výrobci satelitních zařízení nabízí širokou nabídku různých VSAT antén a satelitních modemů. Každá anténa má své funkce, své algoritmy zaměřování na satelit a své protokoly pro komunikaci. Stejně jako antény, satelitní modemy mají své různé a unikátní parametry.

Takový velký počet různých zařízení zjednodušuje vybudování sítí VSAT pro různé požadavky zákazníka. Ale taková rozmanitost může být i problémem, v případě, kdy se na jednom dopravním prostředku nachází několik VSAT terminálů různých výrobců. Každý takový terminál má své parametry a protokoly pro komunikaci, kterým jiný terminál nemusí rozumět. To je velkým problémem při vytváření univerzálního systému ovládání satelitních antén. Existuje jeden protokol, který je přítomen ve většině VSAT zařízení, a to protokol openAMIP, který byl navržen v roce 2006 firmou iDirect.

Firma LYOVA s.r.o. měla zájem o průzkum protokolu openAMIP, který je zabudován ve většině zařízení, a na jeho základě vytvořit aplikaci která by mohla řídit kteroukoliv anténu s kterýmkoliv modemem.

Cílem této práce je ukázat možnost řízení počítačem jak satelitního modemu, tak i antény. Pro vytvoření aplikace byl využit programovací jazyk Java a některé rozšiřující knihovny.

# 1 Hardware

## 1.1 Satellite Router

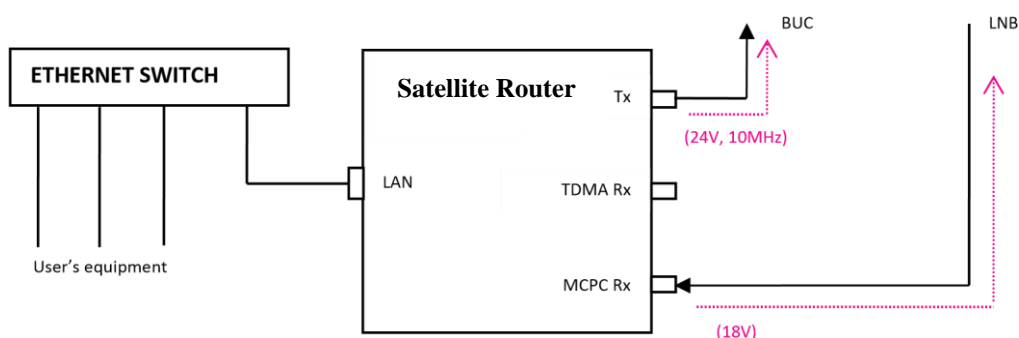
Satelitní router je univerzální hardware, který se dá řídit pomocí zabudovaného softwaru pomocí počítače. V závislosti na aktivovaných možnostech softwaru, router může pracovat v různých režimech a provádět různé role v satelitní síti. Většinou takový router obsahuje:

- Napájení (DC IN)
- Konektor LAN
- USB (konzole)
- Vstup frekvenčního měniče vysokorychlostního demodulátoru (SCPC RX)
- Vstup frekvenčního měniče paketového demodulátoru (TDMA RX)
- Výstup modulátoru (TX OUT)

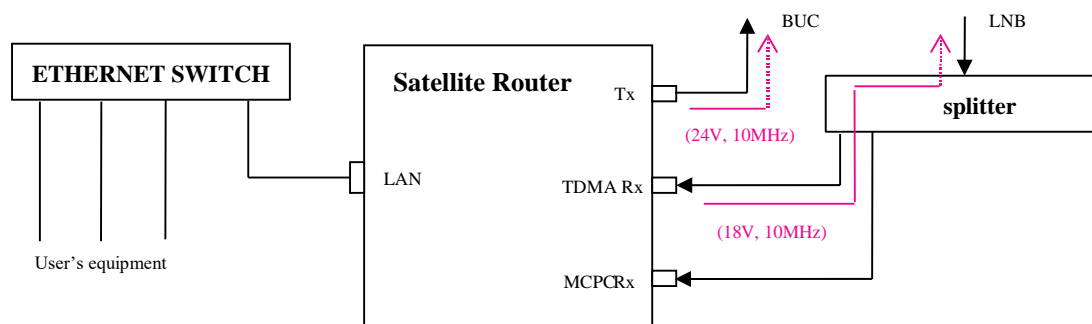
SCPC-RX je jeden ze dvou vstupních signálů z přijímací antény satelitního konvertoru (LNB). Tento vstup je určen pro příjem trvalý nosné (SCPC/MCPC) v DVB-S a DVB-S2 formátu ze satelitu. Délka a kvalita kabelu může mít vliv na kvalitu a schopnost přijímat signály.

TDMA-RX je druhý vstupní signál z přijímací antény satelitního konvertoru (LNB). Tento vstup slouží pro příjem paketu (TDMA) ve správném formátu přeměrovaných ze satelitu.

Modulátor TX OUT je obvykle kompatibilní s většinou satelitních vysílačů / konvertorů (BUC). Modulátor je připojen přímo ke konektoru frekvenčního měniče vysílače.



Obrázek 1: Příklad připojení routeru v režimu SCPC [1]



Obrázek 2: Příklad připojení routeru v režimu TDM / TDMA [1]

Typicky je směrovač připojen k jednotlivým vysokofrekvenčním zařízením (ODU). V takovém uspořádání výstup modulátoru připojen k výkonovému zesilovači BUC a vstup vysokorychlostního demodulátoru připojen do LNB. Uživatelské zařízení by mělo být připojeno ke směrovači přes ethernetový přepínač nebo rozbočovač. Pokud je zapotřebí pouze jedno zařízení, může být připojeno přímo do konektoru LAN.

V režimu TDM / TDMA vstupy vysokorychlostního a paketového demodulátorů musí být připojeny k rozbočovači LNB přes frekvenční měnič. Rozdělovač musí zprostředkovat přenos DC napětí pro napájení LNB a referenčního signálu o 10 MHz, protože na terminálu TDM/TDMA obvykle používá nízko šumový konvertor LNB PLL s externím referenčním signálem. [1][2]

## 1.2 Anténa

### 1.2.1 VSAT

VSAT (Very Small Aperture Terminal) – satelitní pozemská stanice (terminál) s malým průměrem parabolické antény. VSAT vznikl jako experimentální síť na Aljašce kde antény pracovali v C-pásmu. Tyto sítě se ukázaly jako velmi účinné, a tím začal jejich rozvoj.

Když VSAT přešel na Ku pásmo započal nový boom VSAT-systému. Nyní ve světě pracuje více než jeden milion VSAT stanic. Jsou určeny pro vysokorychlostní přístup k Internetu, IP telefonii a vzájemné propojení vzdálených objektů.

VSAT stanice mají průměr antény, který není větší než 2,4 metrů. Výkon vysílače by obvykle neměl přesáhnout 2 Watty. Centrální satelitní pozemská stanice je určena především pro regulaci VSAT sítě (organizace a konfiguraci komunikačních kanálů) a spojení pozemních komunikačních kanálů a veřejných sítí. Satelit-retranslátor (komunikační satelit) přijímá rádiový signál z pozemní stanice, zesiluje ho, moduluje na jinou nosnou frekvenci a přenáší směrem k zemskému povrchu, kde další pozemská stanice přijímá signál, pokud ví, jaká je frekvence. VSAT satelit se pohybuje na geostacionární oběžné dráze ve výšce kolem 36.000 kilometrů. Satelit nachází stále na stejném místě vzhledem k Zemi, což umožňuje použití jednoduchého ovládání antény, protože anténa nemusí sledovat komunikační satelit.

VSAT se skládá ze dvou hlavních částí, ODU (OutDoorUnit) – externí jednotka, tj. anténa a vysílač, typicky 1-2 Wattu a IDU (InDoorUnit) – vnitřní jednotka nebo satelitní modem. ODU – externí jednotka je umístěna v ohnisku antény, který vysílá a přijímá modulované rádiové signály přes satelit. Struktura zahrnuje polovodičový zesilovač ODU (SSPB, BUC), konvertor s malým šumem (LNB) a polarizační volič (OMT).



Obrázek 3: VSAT anténa

Velikost satelitní antény VSAT závisí na několika faktorech:

- Geografická poloha účastnické stanice
- Požadovanou rychlost v opačném kanálu
- Klimatické pásmo, ve kterém je účastnická stanice nachází.

U VSAT jsou používána C, Ku a Ka pásma. C-pásmové antény obvykle mají průměr 2,5 – 3,5 metru, takže formálně to nejsou VSAT antény. Ka-pásmové antény jsou nejmenší. V současné době probíhá přechod na vysokofrekvenční Ka-pásmo. Antény se stále zmenšují a zlevňují, proto se antény VSAT se stávají ve světě rozšířenější. [3][4]

Hlavní výrobci VSAT ve světě:

- Advantech Wireless (Kanada)
- Hughes Network System (USA)
- Gilat (Izrael) – SkyEdge
- ViaSat (USA)
- iDirect(USA)
- NDSatCom (Německo)
- Иcrap (Rusko)
- Newtec(Belgie)
- ComTech

### 1.2.2 Námořní VSAT

Satelitní anténa, která je instalovaná na loď, umožňuje využití telefonů a internetu uživatelem i během plavby. Nicméně, taková VSAT anténa je konstrukčně složitější než stacionární nebo automobilové VSAT antény, vzhledem k potřebě pro přesné směřování antény ke geostacionární družici v nepříznivých podmínkách a pohybu lodí. Pro zajištění stability, anténa využívá složitý polohovací systém, do kterého patří tři gyroskopy, které poskytují stabilizaci ve třech osách



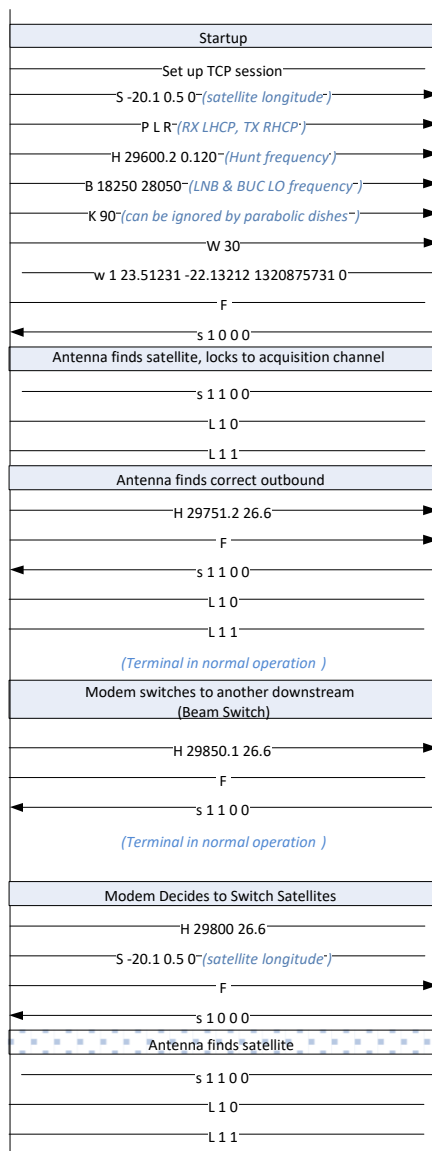
Obrázek 4: Námořní VSAT [12]

(automobilový VSAT stabilizuje pouze dvě osy). Pro ochranu před vodou se používá voděodolný kryt, který propouští radio-frekvenční signály. Taková složitá stabilizace dělá námořní VSAT 200krát dražší než stacionární VSAT a pětkrát dražší než automobilový VSAT.

I když je anténa technologicky složitá, je velice populární. Obecně platí, že VSAT mobilní trh rychle roste, v současné době – až o 50% ročně. Neustále se zvyšuje rychlost přístupu k internetu, aktuálně 8 Mbit/s pro příchozí komunikaci a 2 Mbit/s – odchozí.

Je třeba poznamenat, že ve srovnání s komunikací prostřednictvím satelitního telefonu, námořní VSAT většinou vítězí. Náklady na jednu minutu telefonního hovoru u VSAT technologie jsou přibližně 10krát nižší a cena jednoho megabytu – až 100krát nižší. Jak je vidět k zajištění připojení účastníků na námořních lodích, VSAT skoro nemá žádnou alternativní technologii, která by byla tak rychlá a pohodlná k používání. [3]

### 1.3 OpenAMIP



Obrázek 5: Ukázka komunikace antény a modemu pomocí openAMIP [5]

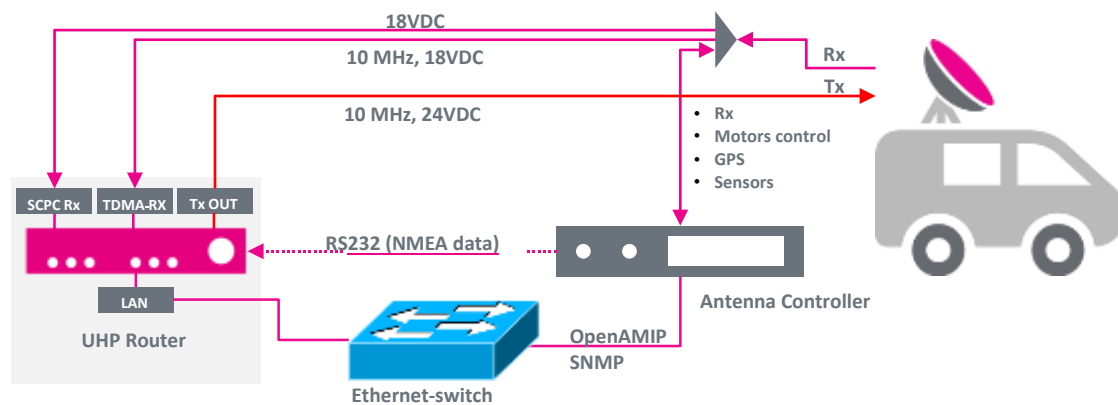
Jedním z hlavních problémů rozšíření VSAT u námořnictva bylo automatické přepínání mezi satelity v průběhu pohybu lodi. Na rozdíl od systému Inmarsat, který tento princip měl zabudovaný od začátku, VSAT systém nebyl původně určen pro použití na moři. Dnes na trhu VSAT postupně vítězí díky stálým tarifům, neomezenému připojení a vysokorychlostnímu přenosu dat. V současné době na celém světě existuje asi 10.000 lodí, které jsou vybaveny VSAT. Nicméně, tento problém nebyl vyřešen.

Řešení nabídla společnost iDirect, která zavedla speciální protokol OpenAMIP pro zprostředkování informací mezi anténami a satelitním modemem. Od té doby se VSAT stal hlavním systémem pro satelitní komunikaci v námořní dopravě. Hlavní funkcí VSAT systému je poskytnout vysokorychlostní přenos dat a telekomunikační služby prostřednictvím satelitu. [5]

Pomocí OpenAMIP, VSAT lze využít nejen na lodích, ale v podstatě na kterémkoli dopravním prostředku. Protokol zahrnuje komunikaci mezi kontrolérem antény a satelitním modemem. OpenAMIP umožňuje modemu řídit kontrolér při hledání požadovaného satelitu. Další důležitá funkce OpenAMIP umožňuje těmto dvěma zařízeními vyměňovat si informace, které jsou nezbytné pro vytvoření a udržení satelitního kanálu.



OpenAMIP přenáší zprávy v podobě ASCII znaků. Jediným účelem protokolu je synchronizovat anténu a modem pro automatickou volbu satelitu. [2][5]



Obrázek 6: Schéma připojení antény a modemu pomocí openAMIP [2]

## 2 Software

### 2.1 Java

„Java je objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems a představila 23. května 1995. Jde o jeden z nejpoužívanějších programovacích jazyků na světě. Podle TIOBE indexu je Java nejpopulárnější programovací jazyk. Díky své přenositelnosti je používán pro programy, které mají pracovat na různých systémech počínaje čipovými kartami (platforma JavaCard), přes mobilní telefony a různá zabudovaná zařízení (platforma Java ME), aplikace pro desktopové počítače (platforma Java SE) až po rozsáhlé distribuované systémy pracující na řadě spolupracujících počítačů rozprostřené po celém světě (platforma Java EE). Tyto technologie se jako celek nazývají platforma Java. Dne 8. května 2007 Sun uvolnil zdrojové kódy Javy (cca 2,5 miliónů řádků kódu) a Java bude dále vyvíjena jako open source.“ [10] – Wikipedie

Programovací jazyk Java byl vybrán pro vytvoření programu z několika důvodů. Jedním z nich je schopnost snadno ovládat více vláken. Virtuální stroj Java je optimalizován pro multi-jádrové stroje, proto není problém řídit stovky paralelně spouštěných vláken. Díky tomu vývojář může psát kód na svém notebooku, a pak spustit aplikaci na serveru, kde využije plnou sílu několik jader procesoru.

Dalším důvodem k použití Javy je kompatibilita mezi platformami. Nebyl to první jazyk pro psaní aplikací pro různé platformy, ale v současné době je nejpopulárnější. To neznamená, že Java má plnou kompatibilitu mezi různými systémy – chybějící knihovny nebo nekompatibilní verze knihoven snadno zastaví běh kódu. Nemůžete vzít kód desktopové aplikace a spustit ji přímo na svém telefonu s Java ME. Ale přes to, vývojář může snadno napsat kód na svém počítači, přenést a spustit ho buď na telefonu, serveru nebo mikrokontroleru a pokud při kompilaci budou k dispozici potřebné knihovny, kód bude pracovat.

Třetí příčina je rozšíření Javy na mikročipech. Java nikdy nebyl populární nástroj pro vývoj desktopových aplikací, na rozdíl od mobilního segmentu. Platforma Android je celá

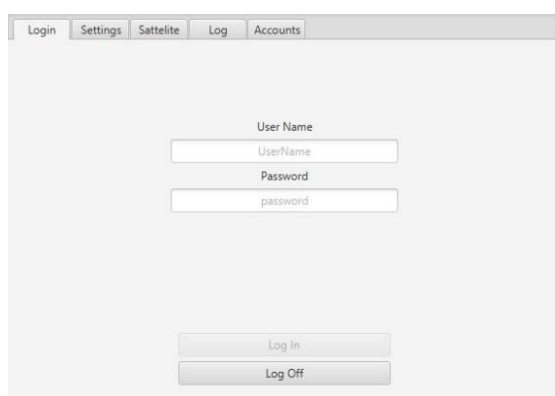
postavena na technologii Java. Zjednodušené verze jazyka a virtuálního stroje, jako Java ME, Java Embedded, Java Card jsou používány v mnoha jednoduchých zařízeních buď s nebo bez displeje (např. čipy na bankovních kartách), jejichž počet po celém světě je v řádu milionů.

Poslední výhodou pro napsání programu v Javě, že je *open source*. Firma Sun a nyní Oracle, byla vždy jedním z lídrů v komunitě *open source*. Díky tomu lze využít různých knihoven a projektů, které byly napsány Java programátory v rámci svobodných a otevřených licencí.

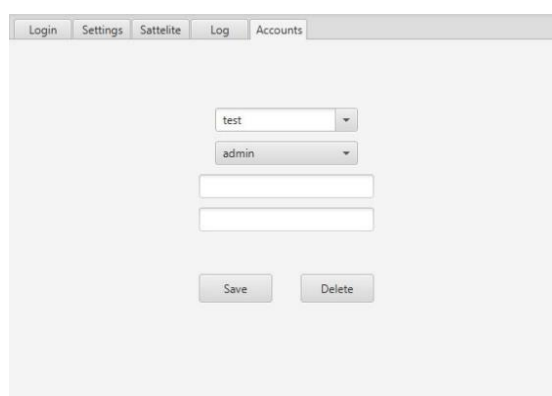
Existují i další výhody jako je například kompilace kódu pomocí virtuálního stroje, který je napsán v jiném jazyce (firma Microsoft si půjčila tuto myšlenku při vývoji C#, ale takový trik lze provést jenom na virtuálním stroji, který pracuje pouze v operačním systému Windows). Tyto výhody nebyly zatím přímo využity při tvorbě programu, ale umožní snadné budoucí rozšíření.

## 2.2 Návrh aplikace

Aplikace se skládá z osmi tříd. První z nich *Main* slouží jenom k spuštění programu a nastavení rozměru zobrazovacího okna. Ve třídě *MainController* probíhá veškerá práce s grafickým prostředím aplikace a veškeré požadavky. Pokud uživatel zmáčkne tlačítko, kód pro jeho vykonání se nachází zde. V této třídě jsou inicializovány všechny operátory pro všechny grafické objekty, jako jsou například tlačítka nebo pole pro zadávání hodnot. Dál se ve třídě nachází ještě vnořená třída *SQL*, která umožňuje pracovat s databází *SQL*. Je vnořena, protože pracuje se stejnými daty, jako třída *MainController*. Třída *SQL* má nejen funkce pro čtení, uložení a mazání informací, ale také umí šifrovat hesla a pak porovnávat s údaji v databázi pro vyhodnocování oprávnění uživatelů (více v kapitole 2.4).

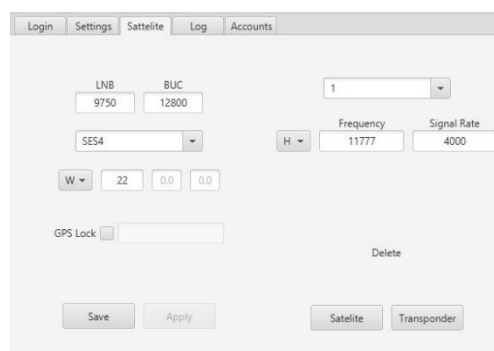


Obrázek 8: Přihlašovací okno



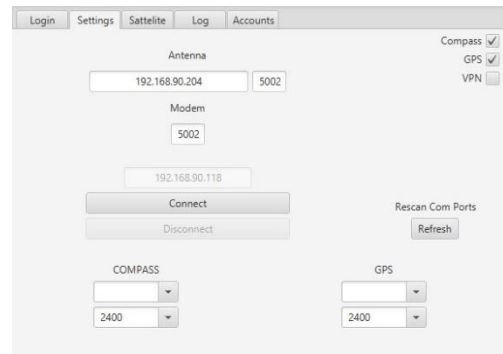
Obrázek 7: Okno pro vytvoření nového uživatele

*SQL* databáze obsahuje informace o satelitech, které může vytvářet uživatel. V databázi jsou uloženy dvě tabulky: jedna pro uložení informace o poloze satelitů, druhá pro uložení počtu a dat transpondérů. Každý transpondér ví, ke kterému přísluší satelitu. Uživatel může sám vytvořit, upravit nebo pouze volit (pokud je typ účtu *user*) data která potřebuje. Pokud uživatel chce, aby se anténa zaměřila na zvolený satelit, musí zmáčknout tlačítko *Apply* a všechny potřebné příkazy se provedou automaticky. To lze provést jen, když je spojení mezi anténou a modemem už vytvořeno.

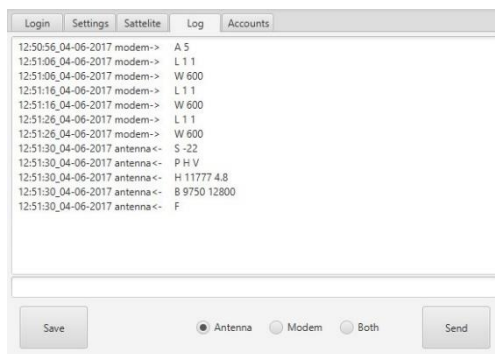


Obrázek 9: Nastavení satelitu/transpondéru

Dále se ve třídě *MainController* spustí vlákna *GenealThread* a *VPN*. *VPN* vlákno se spouští, když začne běh programu, aby se k aplikaci šlo připojit pomocí *VPN* (vlákno *VPN2*). *GenealThread* vlákno řídí veškerou komunikaci v síti pro potřeby vytvoření a udržení satelitního spoje. Toto vlákno spustí jen tehdy, když uživatel zadá příslušné parametry (např. IP adresu antény a port pro modem) a stiskne tlačítko *Connect*. Externě lze připojit kompas a GPS pomocí COM port. To je výhodné, pokud chceme provozovat systém na velké lodi která má přesnější navigační zařízení. Když program začne běžet, automaticky proskenuje všechny vstupy, uloží adresy do tabulky ze které lze volit COM port příslušného zařízení.



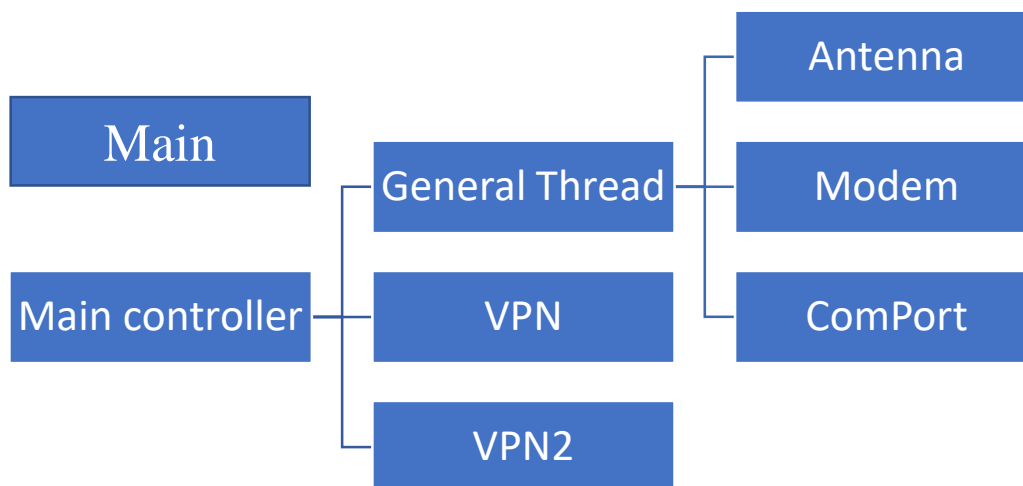
Obrázek 10: Okno pro nastavení síťových parametrů



Obrázek 11: Okno komunikace

Když vlákno *GenealThread* snaží navazovat spojení, spustí ještě dvě vlákna: *Antenna* a *Modem*. Jak je vidět z názvu, tato dvě vlákna pracují přímo s těmito zařízeními, dle kterých mají názvy. Vytváří vstupní a výstupní kanál, po kterých se předávají data z modemu do antény a opačně. Tato data zaznamenají do logu diagnostického okna, kde je vidět veškerá komunikace a lze ji také

řídít pomocí příkazového řádku. Pokud jsou zaškrtnuty položky *GPS* nebo *Compass*, spustí se příslušné vlákno pro obsluhu těchto zařízení.



Obrázek 12: Diagram programu (viz příloha)

## 2.3 Knihovny

### 2.3.1 SQL

Pro ukládání potřebných dat např. hesla nebo souřadnic satelitu, program využívá databáze SQL. Pro práci s SQL databází program používá externí knihovnu `sqlite-jdbc-3.16.1.jar`.

Program se připojí k databázi pomocí příkazu:

```
connection = DriverManager.getConnection("jdbc:sqlite:SQL.db");
statement = connection.createStatement();
```

kde `dbc:sqlite:SQL.db` je cesta k databázi. Pokud databáze je přímo v adresáři, stačí napsat jen její jméno. Databáze SQL může obsahovat několik tabulek. Program má celkem čtyři tabulky: `LNB_BUC`, `login`, `satelites` a `transponders`.

Tabulka `LNB_BUC` obsahuje parametry zařízení LNB a BUC. Tyto hodnoty se mohou změnit jen když změníme celou anténu.

V tabulce `login` jsou uloženy přihlašovací údaje uživatelů, typ účtu (admin nebo běžný účet) a tzv. `sůl` (viz 2.4.1).

Tabulka `satelites` obsahuje souřadnice, které budou poslány do antény pro detekování satelitu.

Poslední tabulka `transponders` obsahuje údaje, ke kterému satelitu zvolený transpondér přísluší a vlastní údaje pro navázání spojení.

Program se připojí k potřebné tabulce pomocí příkazu `SELECT`:

```
resultSet = statement.executeQuery("Select LNB, BUC from LNB_BUC");
```

Změna hodnot se dá provést příkazem *UPDATE*:

```
preparedStatement = connection.prepareStatement("UPDATE LNB_BUC SET LNB = ?,BUC = ?");

    preparedStatement.setString(1, textLNB.getText());
    preparedStatement.setString(2, textBUC.getText());
    preparedStatement.executeUpdate();
```

„preparedStatement.setString(1, textLNB.getText())” příkaz pro změnu hodnoty v proměnné LNB. Příkaz „preparedStatement.executeUpdate()” je pro uložení změn v tabulce.

Pro vytvoření nových hodnot existuje příkaz *INSERT*:

```
preparedStatement = connection.prepareStatement("INSERT INTO
sattelites(Sattelite,EW,coordinates,angle,velocity) VALUES(?,?,?,?,?)");

    preparedStatement.setString(1, cbSatelliteName.getEditor().getText());
    preparedStatement.setString(2, choiceBoxEW.getValue().toString());
    preparedStatement.setString(3, textSatelliteCoordinates1.getText());
    preparedStatement.setString(4, textSatelliteCoordinates2.getText());
    preparedStatement.setString(5, textSatelliteCoordinates3.getText());
    preparedStatement.executeUpdate();
```

Příkaz *DELETE* pro mazání hodnot:

```
preparedStatement = connection.prepareStatement("DELETE FROM sattelites WHERE Sattelite = ?");

    preparedStatement.setString(1, id);
    preparedStatement.executeUpdate();
```

Potřebný řádek najdeme pomocí *WHERE*.

### 2.3.2 Serial Port

K programu lze připojit externí periférii např. GPS nebo kompas. Program umí zobrazovat souřadnice, které posílá zmíněné zařízení a následně uložit do souboru pro diagnostické účely. Pro práci se sériovým portem je použita externí knihovna RXTXcomm.jar. Pomocí příkazu „serialPort = (SerialPort) portId.open(name, TIMEOUT)“ se dá nastavit jméno

COM Portu a čas, jak dlouho se bude program pokoušet připojit se k zvolenému COM Portu.

```
try {
    serialPort = (SerialPort) portId.open(name, TIMEOUT);
} catch (PortInUseException ex) {
    Logger.getLogger(ComPort.class.getName()).log(Level.SEVERE, null, ex);
    return false;
}

serialPort.notifyOnDataAvailable(true);

try {
    serialPort.setSerialPortParams(speed, DATABITS_8, STOPBITS_1, PARITY_NONE);
    serialPort.setFlowControlMode(FLOWCONTROL_NONE);
} catch (UnsupportedCommOperationException ex) {
    Logger.getLogger(ComPort.class.getName()).log(Level.SEVERE, null, ex);
}

try {
    inputStream = serialPort.getInputStream();
} catch (IOException ex) {
    Logger.getLogger(ComPort.class.getName()).log(Level.SEVERE, null, ex);
}
```

Po uplynutí času program skočí do výjimky. Pomoci příkazu „serialPort.setSerialPortParams(x1, x2, x3, x4)“ se dají nastavit parametry spojení, kde

x1 je rychlost připojení,

x2 je počet datových bitů,

x3 je počet stop bitů,

x4 je používaná parita.

Příkazem „serialPort.setFlowControlMode(FLOWCONTROL)“ nastavuje se zbývající parametr flowcontrol. Ukázka kódu níže demonstruje celou funkci pro připojení k sériovému portu.



## 2.4 Zabezpečení hesla

Pro zabránění neoprávněného přístupu k aplikaci je možné zahešovat hesla uživatelů. Kryptografická hašovací funkce musí splňovat tři podmínky:

- Nevratnost

Pro danou hesh funkci je obtížné najít jinou kombinaci pro dosažení stejného hash kódu.

- Odolnost ke kolizím I druhu

Pro zprávu M musí být obtížné systematicky najít jinou zprávu N takovou, že  $H(N) = H(M)$ .

- Odolnost ke kolizím II druhu

Musí být obtížné systematicky najít dvě zprávy (M, M') které mají stejný hash.

Pod hašováním rozumíme konverzi vstupních dat libovolné délky do výstupního bitového řetězce pevné délky. Takovou konverzi využíváme pro ověření uživatelů nebo pro výpočet kontrolních sum souboru. Existuje celá řada algoritmů hašování: MD5, SHA, bcrypt.

### 2.4.1 SHA

Algoritmus Heshování SHA-1 (Secure Hash Algorithm Version 1 — Bezpečný algoritmus hešování, verze 1) byl vyvinut v roce 1995. Některé nedostatky SHA-1 byly nalezeny v roce 2005. Pro vstupní zprávy libovolné délky (až 2 exabytů) algoritmus generuje 160 bitovou hodnotu hash.

Algoritmus SHA-2 – rodina kryptografických algoritmů SHA-224, SHA-256, SHA-384, SHA-512, SHA-512 / 256 a SHA-512/224, které vypracovala Národní bezpečnostní agentura USA (NSA) a zveřejnil Národní institut standardů a technologie (NIST). [6][7]

Hash se dá považovat za bezpečný, pokud splňuje dvě kritéria:

- Odolnost proti útokům (Slovníkový útok, Rainbow tables, kolize)
- Nemožnost najít stejná hesla různých uživatelů pomocí hash kódu

První bod lze splnit, pokud budeme používat moderní algoritmy hašování jako například SHA-512. V Javě se používá konstruktor třídy *MessageDigest*, do kterého lze zadat požadovaný algoritmus (MD5, SHA-1, SHA-256, SHA-512). Do metody *update* se předává pole bitů pro generování hashu. Ukázka níže demonstruje použití generování hashe pomocí SHA-512:

```
public String cryptPassword(String password, byte[] salt) throws NoSuchAlgorithmException,
    UnsupportedEncodingException {

    MessageDigest md;
    md = MessageDigest.getInstance("SHA-512");
    md.update(salt);
    byte crypt[] = md.digest(password.getBytes("UTF8"));
    for (int i = 0; i < crypt.length; i++) {
        result += Integer.toHexString((0x000000ff & crypt[i] | 0xfffff00).substring(6));
    }
    return result;
}
```

Pro splnění druhé podmínky k heslu před hešováním připišeme sekvenci náhodných dat (tzv. salt).

## 2.4.2 Salt

Salt (česky – sůl) je zapotřebí, když dva uživatelé mají stejné heslo např. ABC. Po přidání soli (*ABCsul1* a *ABCsul2*) budou mít různou hash funkci. Salt – je náhodná sekvence dat, která se přidá k heslu. Existují různé algoritmy pro generování soli, od úplně náhodných proměnných, generovaných pomocí náhodných čísel, do prokládání bitů tak, aby sekvence byla co nejvíce prostřídána jedničkou a nulou. Sůl je taky výhodné použít proti slovníkovému útoku.



Obrázek 13: Ukázka získání hesla admin pomocí online decryptoru.



Obrázek 14: Ukázka odolnosti hash kódu se soli

Aby hash kód bylo těžké odhadnout, sůl se píše za heslem. Pokud by to tak nebylo, pachatelům útoku usnadníme práci, protože stačí spočítat hash soli a pak lze vyrobit  $hash(sůl)+hash(heslo)$  skoro stejnou rychlostí jako kdyby bylo použito jenom heslo bez soli.

Níže je uvedena ukázka výroby soli.

```
private byte[] getSalt() throws NoSuchAlgorithmException {
    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
    byte[] salt = new byte[16];
    sr.nextBytes(salt);
    return salt;
}
```

### 3 Testování

Testování bylo provedeno na dvou VSAT terminálech. První terminál – anténa COBHAM EXPLORER 7100 a satelitní modem UHP-1000. Druhý terminál – anténa NAVISYSTEM VSAT95HT a satelitní modem UHP-1000. Počítač s testovací aplikací byl připojen k terminálu přes switch.



Obrázek 15: VSAT95HT



Obrázek 16: EXPLORER 7100

1. Ověření funkčnosti antény bez použití testovací aplikace ve standardním režimu s použitím profilu nastavení ACU (zaměření na satelit SES4 22W). Výsledek: anténa se zaměřila na satelit, modem přijímá požadovanou frekvenci.
2. Příprava testovací aplikace k testům. Na počítače použita IP adresa pro síť VSAT terminálu. V okně aplikace uvedena IP adresa ACU a modemu.
3. Počítač s testovací aplikací ke zkouškám připraven.
4. Ověření funkčnosti antény s použitím protokolu openAMIP.
  - a. Při provádění testu s použitím antény EXPLORER 7100, v ACU, v profilu nastavení bylo povoleno ovládání od modemu iDirect (protokol openAMIP).
  - b. Při provádění testu s použitím antény VSAT95HT v ACU byl zapnut režim ovládání pomocí protokolu openAMIP.

Výsledek: anténa v režimu vyhledání, aplikace dostala informaci o statusu antény (není zaměřena a režim Tx zakázán – s 0 0) a souřadnicích (w 1 23.51231-22.13212 0). Stejně tak byla přijata informace od modemu (není požadovaná frekvence a režim Tx vypnut – L 0 0).

- c. V programu, v okně *Nastavení satelitu/transpondéru* byl vybrán satelit SAT1 (SES4 22W) a odeslání těchto dat bylo provedeno pomocí zmáčknutí tlačítka *Apply*.

Výsledek: v době vyhledávání satelitu aplikace přijímala informaci o statusu antény (není zaměřena a režim Tx zakázán – s 0 0) a souřadnicích (w 1 23.51231-22.13212 0). Stejně tak byla přijata informace od modemu (není požadovaná frekvence a režim Tx vypnut – L 0 0).

Anténa našla a zaměřila se na satelit SAT1 (SES4 22W). Aplikace přijala informaci o statusu antény (zaměřeno, režim Tx povolen – s 1 1) a souřadnice (w 1 23.51231-22.13212 0). Informace přicházela i od modemu (požadovaná frekvence je přítomna, režim Tx zapnut – L 1 1)

- d. V programu, v okně *Nastavení satelitu/transpondéru* byl vybrán satelit SAT2 (AZERSPACE1) a odeslání těchto dat bylo provedeno pomocí stisku tlačítka *Apply*.

Výsledek: v době vyhledávání satelitu aplikace přijímala informaci o statusu antény (není zaměřena a režim Tx zakázán – s 0 0) a souřadnicích (w 1 23.51231-22.13212 0). Stejně tak byla přijata informace od modemu (není požadovaná frekvence a režim Tx vypnut – L 0 0).

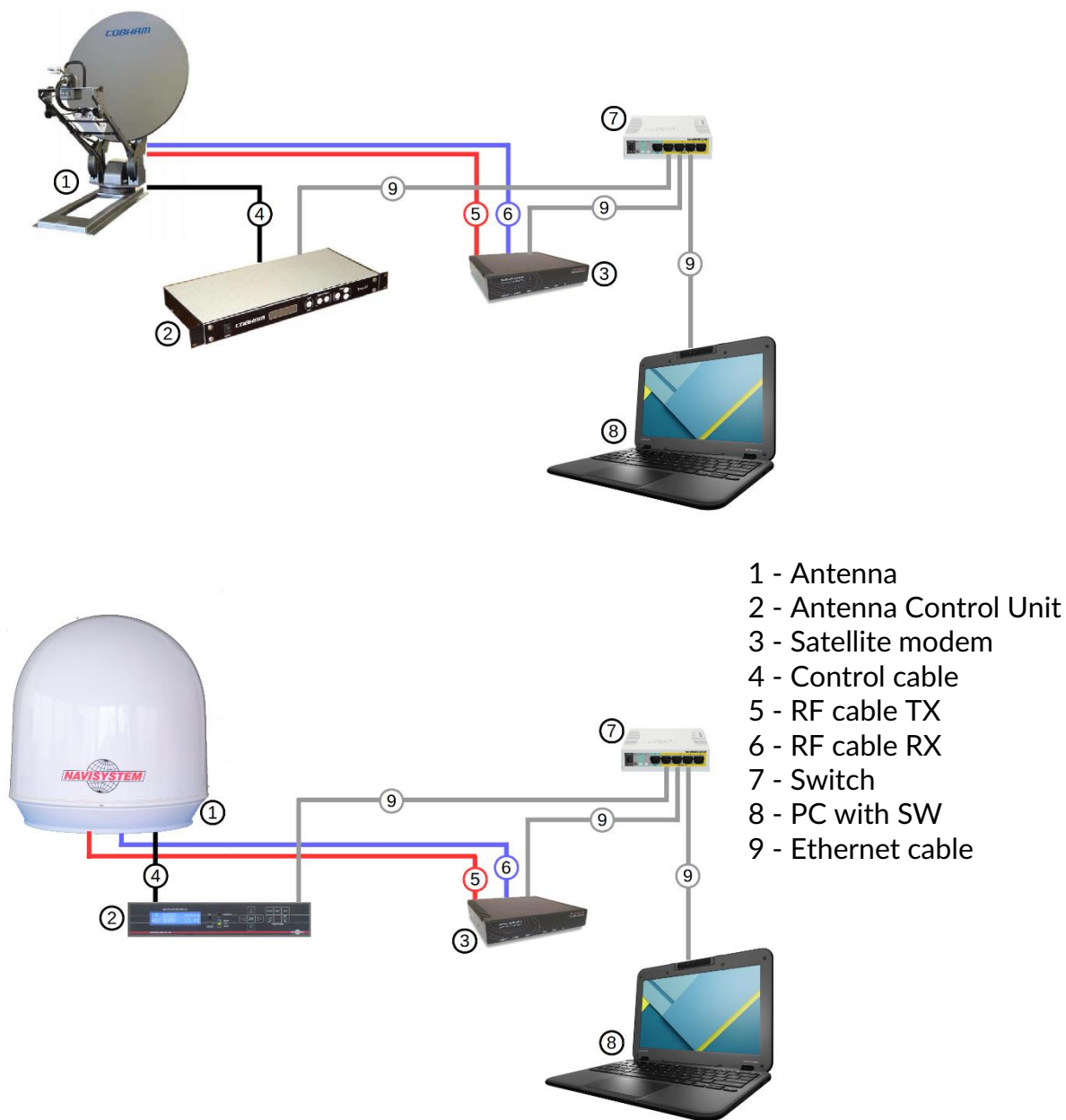
Anténa našla a zaměřila se na satelit SAT2 (AZERSPACE1). Aplikace přijala informaci o statusu antény (zaměřeno, režim Tx povolen – s 1 1) a souřadnice (w 1 23.51231-22.13212 0). Informace přicházela i od modemu (požadovaná frekvence je přítomna, režim Tx zapnut – L 1 1).

5. Tento postup byl opakován několikrát pro ověření funkčnosti programu. Bylo ověřeno přepínání z jednoho satelitu na druhý.

Výsledek: anténa zaměřovala požadovaný satelit při každém požadavku z programu.

6. Ověření přijetí doplňující navigační informace přes COM porty. V okně programu *Nastavení síťových funkcí* byly nastaveny parametry COM portu (číslo COM portu, rychlost).

Výsledek: aplikace přijímala informace z obou COM portů.



Obrázek 17: Schéma zapojení VSAT terminálu [12]

## 4 Další možnosti vývoje

To, že je aplikace napsaná v jazyce Java, dává široké možnosti pro rozšíření programu. Lze napsat další užitečné moduly programu, například automatické přepínání satelitu.

Další možností zlepšení programu, je úprava jeho vizuální části. Z důvodu rozsahu programu a omezeného času v rámci bakalářské práce, nebylo zatím řešeno otevírání aplikace do režimu celé obrazovky.

Jiná možnost zlepšení je přenesení aplikace na přenosné zařízení. I když na každé větší lodi je vždy k dispozici počítač, na kterém lze spustit řídicí aplikaci, u jachet a automobilů to může být problém. Z toho důvodu by bylo výhodné kód přenést buď na telefon nebo Raspberry Pi aby majitelé těchto dopravních prostředků nemuseli používat desktopové počítače.

V aplikaci lze také vytvořit doplňující profily pro konfiguraci a řízení několika antén, a to najednou či zvlášť.

Aplikace je připravena tak, že má velký potenciál pro přidávání dalších požadovaných funkcí.

## **Závěr**

I když v dnešní době poptávka po systému VSAT neustále roste, stále je problémem přepínání satelitu. Tento problém brání dalšímu rozšíření tohoto systému.

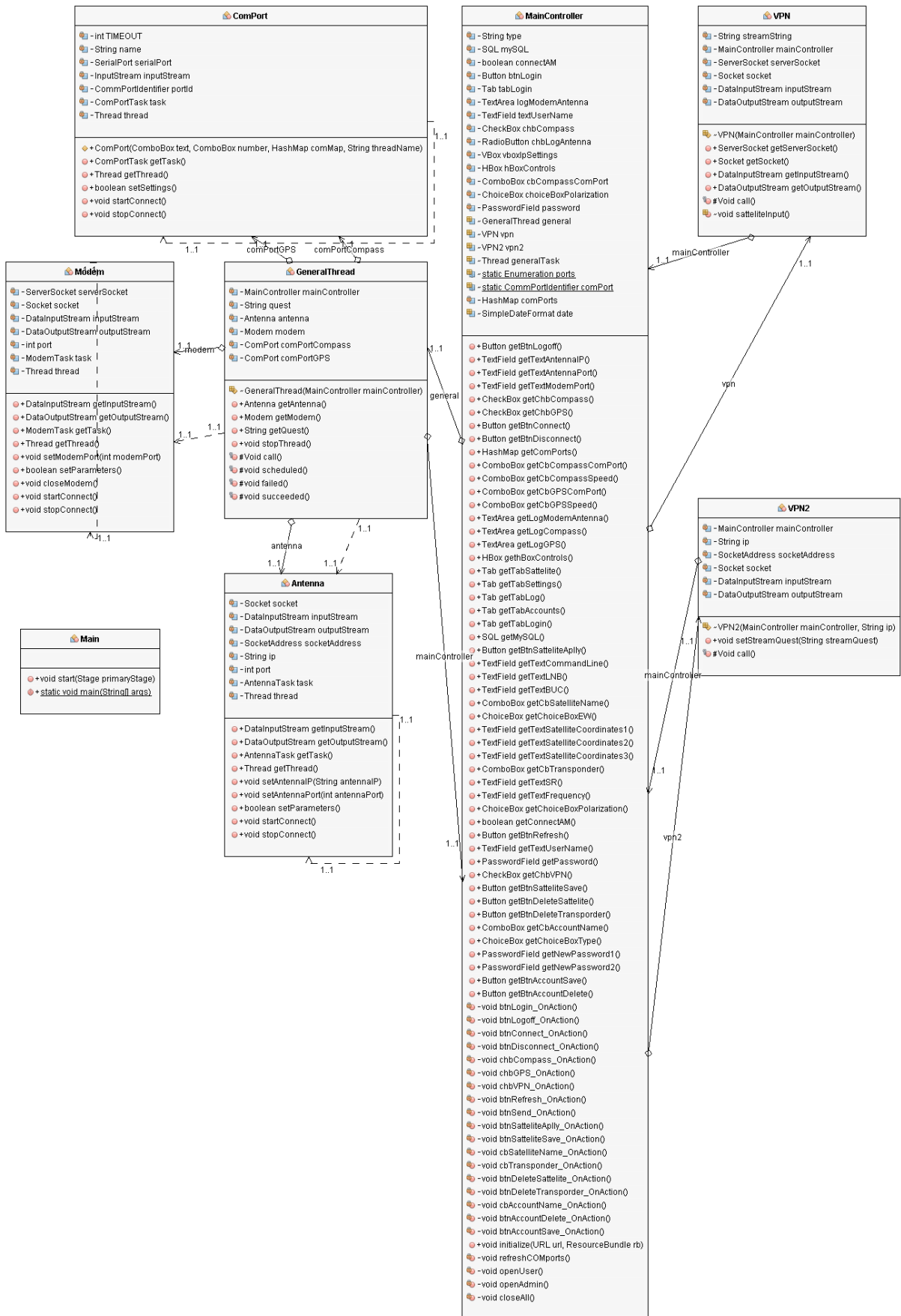
V této bakalářské práci bylo ukázáno, že pomocí openAMIP lze řídit satelitní systémy, a to nejen námořní. Jak je vidět, lze řídit celý systém i v případě automobilového VSAT. V této oblasti může být takové řízení ještě mnohem užitečnější. Na rozdíl od lodí, kde existuje člověk, který ovládá telekomunikační systémy, potřebné pro udržení spojení s ostatním světem, v osádce automobilu takový člověk většinou není. Například rallye tým v pouštní oblasti potřebující odvysílat např. video reportáž o postupu svého týmu, může s výhodou využít takový VSAT systém, který bude nenáročný na ovládání a půjde z něho odesílat velké objemy dat.



## Seznam literatury a informačních zdrojů

- [1] SATELLITE ROUTER UHP-1000 GENERAL DESCRIPTION AND INSTALLATION GUIDE, 2015, RELEASE 3.2.
- [2] UHP SCPC MODEM USER MANUAL, 2015, RELEASE 3.2.
- [3] VSAT 95 HT Instruction and instalation manual, 2010, release 6.0.
- [4] TracLRI Live Remote Interface Installation & Operations Reference Guide,2014, v.1.2.
- [5] Specification: OpenAMIP V1.7 ICD,2012, v.4.4.
- [6] Paul Mutton, One million SSL certificates still using “insecure” SHA-1 algorithm, 19.10, 2015, Dostupné z: <https://news.netcraft.com/archives/2015/10/19/one-million-ssl-certificates-still-using-insecure-sha-1-algorithm.html>.
- [7] Marc Stevens, Pierre Karpman and Thomas Peyrin, Freestart collision for full SHA-1,2015, Dostupné z: <https://eprint.iacr.org/2015/967.pdf>.
- [8] OpenAMIP Solution Overview, Dostupné z: <http://www.idirect.net/~/media/Files/Solution%20Overviews/OpenAMIPSolution-Overview-2014.pdf>
- [9] Karl Fuchs, Antenna Integration: The Open-AMIP Standard For SATCOM, 2013, Dostupné z: <http://www.satmilmagazine.com/story.php?number=1483485170>
- [10] Java, 2017, Dostupné z: [https://cs.wikipedia.org/wiki/Java\\_\(programovací\\_jazyk\)](https://cs.wikipedia.org/wiki/Java_(programovací_jazyk))
- [11] K. Sierra, B. Bates, Head First Java,2. vydání, ЭКСМО, 2012, ISBN: 978-5-699-54574-2.
- [12] Poskytnuto firmou Lyova s.r.o.

# Přílohy



## Obsah příloženého CD

- text..... text práce
  - BP.pdf ..... text práce ve formátu PDF
- program..... zdrojové kódy
  - viaAMIPv0.6 ..... kód programu
    - SQL ..... databáze
    - lib..... externí knihovny
      - RXTXcomm.jar ..... knihovna pro práce s COM porty
      - sqlite-jdbc-3.16.1.jar ..... knihovna pro práce s SQL