



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra měření**

Diplomová práce

Softwarový nástroj na návrh komunikačního protokolu družicového přístroje podle ECSS-E-ST-70-41C

Bc. Hana Pikulíková
Letectví a kosmonautika

Leden 2023

Vedoucí práce: Prof. RNDr. René Hudec, CSc.



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Pikulíková** Jméno: **Hana** Osobní číslo: **466020**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra měření**
Studijní program: **Letectví a kosmonautika**
Studijní obor: **Avionika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Softwarový nástroj na návrh komunikačního protokolu družicového přístroje podle ECSS-E-ST-70-41C

Název diplomové práce anglicky:

Software Tool for Design of Satellite Payload Communication Compliant with ECSS-E-ST-70-41C

Pokyny pro vypracování:

Navrhněte a realizujte softwarový nástroj v programovacím jazyce C#, ve kterém bude možné plně specifikovat komunikační protokol s obecným přístrojem na palubě družice a následně i s přístrojem komunikovat. Volba komunikačního protokolu musí být v souladu se standardem ECSS-E-ST-70-41C a musí být možné specifikovat služby typu č. 1, 3, 5, 6, 9 a 17. Zprávy přenášené v rámci uvedených služeb jsou ve formátu CCSDS paketů. Při popisu uvedených služeb se zaměřte na určení typů parametrů v závislosti na typu služby. Na službě typu č. 6 (služba správy paměti) popište navíc formát komunikačního protokolu ASN.1/ACN. Popište navržený softwarový nástroj, který bude mimo jiné splňovat následující vlastnosti (TC = telecommand, TM = telemetry):

- lze specifikovat výše uvedené služby;
 - grafické rozhraní, které umožní správné zadávání syntaxe ASN.1/ACN;
 - uložení/načtení do/z formátu ASN.1/ACN;
 - grafické rozhraní, kterým lze specifikovat různé množství testovacích příkazů – TC;
 - grafické rozhraní, kterým lze zobrazit obsah přijatých dat – TM;
 - generování dokumentace k navrženému komunikačnímu protokolu;
 - spuštění programu asn1scc (ze sady volně dostupných nástrojů ESA - TASTE) pro vygenerování zdrojových kódů v jazyce Python, které slouží pro sestavování a dekodování paketů.
 - vygenerování čitelného zdrojového kódu pro odeslání TC a přijetí TM v jazyce Python, který bude i samostatně spustitelný. Následujícím postupem ověřte navržený softwarový nástroj pro komunikaci s přístrojem Space Dosimetry System Demonstrator (2SD), který má sériovou linku a reprezentuje přístroj použitý na družici VZLUSAT-2:
 - načtením existujících souborů 2SD (ASN.1/ACN) s předem ručně upravenou službou typu č. 6, v které odstraňte 10 TC a TM;
 - grafickým nástrojem doplňte odstraněné TC a TM ve službě typu č. 6;
 - pro všechny nově zadané TC nadefinujte i vstupní testovací hodnoty;
 - nástrojem vygenerujte zdrojový kód, který bude rovnou spuštěn;
 - přijaté odpovědi na jednotlivé příkazy zobrazte v nástroji, kde se uživateli zobrazí již dekodované hodnoty.
- Doporučte další postup pro rozvoj vytvořeného softwarového nástroje.

Seznam doporučené literatury:

- 1) Larson W. J., Wertz, J. R., Space Mission Analysis and Design, Microcosm Inc., Netherlands, 3rd ed. , 1999, ISBN 1-881883-10-8
- 2) Fortescue P., Swinerd G., Stark J., Spacecraft Systems Engineering, 4th edition, John Wiley & Sons, Inc., UK, 2011, ISBN 978-0-470-75012-4
- 3) ECSS-E-ST-40C. Space Engineering: Software. Noordwijk: ESA Requirements and Standards Division, 2009.
- 4) ECSS-M-ST-40C. Space Engineering: Configuration and Information Management. Noordwijk: ESA Requirements and Standards Division, 2009.
- 5) ECSS-Q-ST-80C. Space Product Assurance: Software Product Assurance. Noordwijk: ESA Requirements and Standards Division, 2009.
- 6) ECSS-E-ST-70-41C. Space Engineering: Telemetry and telecommand packet utilization. Noordwijk: ESA Requirements and Standards Division, 2016.
- 7) ECSS-E-HB-40A. Space Engineering: Software Engineering Handbook. Noordwijk: ESA Requirements and Standards Division, 2013.

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. RNDr. René Hudec, CSc. katedra radioelektroniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **07.02.2022** Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce:
do konce letního semestru 2022/2023

prof. RNDr. René Hudec, CSc.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomantka bere na vědomí, že je povinna vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

Datum převzetí zadání

Podpis studentky

Poděkování / Prohlášení

Ráda bych poděkovala vedoucímu diplomové práce panu Prof. RNDr. René Hudcovi, CSc. a konzultantovi práce panu Ing. Petru Česákovi za odborné rady, podmětné návrhy a konzultace. Nakonec bych chtěla poděkovat rodině a příteli za jejich podporu a trpělivost při celém studiu.

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 10. 1. 2023

.....

Abstrakt / Abstract

Diplomová práce se zabývá návrhem a realizací softwarového nástroje v programovacím jazyce C#, kterým je možné specifikovat komunikační protokoly s přístrojem na palubě družice a následně i s přístrojem komunikovat. Práce se zabývá problematikou definování komunikačních protokolů v souladu se standardem ECSS-E-ST-70-41C. Funkčnost navržené aplikace je ověřena s přístrojem Space Dosimetry System, který reprezentuje přístroj na družici VZLUSAT-2.

Klíčová slova: ECSS-E-ST-70-41C, ASN.1, ACN, Služba, Podslužba, TC, TM

Vedoucí práce: Prof. RNDr. René Hudec, CSc.

The diploma thesis deals with the design and implementation of a software tool in the C# programming language, with which it is possible to specify communication protocols with the device on board the satellite and subsequently to communicate with the device. The thesis deals with the issue of defining communication protocols in accordance with the ECSS-E-ST-70-41C standard. The functionality of the proposed application is verified with the Space Dosimetry System device, which represents the device on the VZLUSAT-2 satellite.

Keywords: ECSS-E-ST-70-41C, ASN.1, ACN, Service, Subservice, TC, TM

Title translation: Software Tool for Design of Satellite Payload Communication Compliant with ECSS-E-ST-70-41C

Obsah /

1 Úvod	1
2 Satelity	3
2.1 Orbita satelitů	3
2.1.1 Vynesení satelitu na oběžnou dráhu	4
2.1.2 Typy orbitů	4
2.2 Funkce umělých satelitů	5
3 Komunikace s kosmickým pro- středkem	6
3.1 Systém Command and Data Handling (C&DH)	6
3.1.1 Funkce systému C&DH, TC a TM komunikace	6
4 Zavedení normalizace a stan- dardizace ve vesmírném sektoru ..	8
4.1 Standardizace vesmírného sektoru v Evropě	8
4.1.1 Evropská vesmírná agentura (ESA)	8
4.1.2 European Cooperation for Space Standardi- zation (ECSS)	8
4.2 Standardy pro paketovou ko- munikaci	9
4.2.1 Standard ECSS-E-ST- 70-41C	9
5 Tvorba komunikačních zpráv	19
5.1 Abstract Syntax Notation One (ASN.1)	19
5.1.1 ACN	20
6 Použité prostředky	23
6.1 Knihovna tříd Windows Forms	23
6.1.1 Důvod vybrání	23
6.1.2 Vytvoření formuláře	23
6.2 Programovací framework a jazyky	24
6.2.1 C #	24
6.2.2 Python	25
6.2.3 NET framework	25
6.3 Generování dokumentace	25
6.3.1 Nástroj LaTeX	25
6.4 Nástroj Taste tools od ESA ...	25
6.4.1 Funkce nástroje ESA- TASTE	26
6.5 Kompilátor ASN1SCC	26
7 Návrh aplikace	27
7.1 Architektura aplikace	27
7.1.1 Požadavky na funkce aplikace	27
7.1.2 Rozvržení aplikace	29
7.2 Název a ikona aplikace	34
7.3 Tvorba uživatelského rozhra- ní (UI)	34
7.3.1 Zásady zahrnuté v UI aplikace	34
7.3.2 Výběr barev aplikace ...	34
8 Implementace aplikace	36
8.1 Hlavní formulář	36
8.1.1 Načtení existujících souborů ASN.1 a ACN ..	36
8.1.2 Vytvoření souborů ASN.1 a ACN	37
8.1.3 Zápis do souborů ASN.1 a ACN	37
8.1.4 Panel správy paketů TC a TM	38
8.2 Formulář výběru typu služby a podslužby	39
8.3 Formulář specifikace infor- mací protokolu	40
8.4 Formulář volby názvu datové kombinace	42
8.5 Formulář zobrazení informa- cí TC a TM paketů	42
8.6 Dokumentace	43
8.7 Formulář komunikace se sa- telitem	44
8.8 Formulář specifikace hodnot dat TC paketů	46
8.9 Formulář volby IP adresy a portu	46
8.9.1 Komunikace se zaříze- ním	47
8.9.2 Vizualizace TM odpo- vědí přijatých od TCP serveru	49
9 Testování aplikace	51
9.1 Přístroj testování	51
9.1.1 VZLUSAT-2	51

9.1.2	Softwarový nástroj Space Dosimetry Sys- tem Demonstrator (2SD)	52
9.2	Testování aplikace	52
9.2.1	Testovací scénáře	52
9.3	Výsledky testování	53
9.3.1	Načtení souborů	53
9.3.2	Úprava paketů TC a TM	53
9.3.3	Nadefinování testova- cích hodnot TC paketů ..	54
9.3.4	Vygenerování zdrojové- ho kódu aplikací a jeho následné spuštění	55
9.3.5	Dekódování přijatých zpráv a zobrazování v aplikaci	56
9.4	Nápady na vylepšení	59
10	Závěr	60
	Literatura	62
A	Seznam zkratk	65
B	Elektronické přílohy	66

Tabulky / Obrázky

<p>2.1. Parametry orbity4</p> <p>4.1. Standardní typy služeb 11</p> <p>4.2. Podslužby služby Ověření požadavku (Request verification) - Přijímání a podávání zpráv (Acceptance and reporting) 12</p> <p>4.3. Podslužby služby Ověření požadavku (Request verification) - Hlášení exekuce (Execution reporting) 12</p> <p>4.4. Podslužby služby Ověření požadavku (Request verification) - Směrování a hlášení (Routing and reporting) 13</p> <p>4.5. Podslužby služby Provoz a zjištění stavu zařízení (Housekeeping) - Hlášení provozu (Housekeeping reporting) 13</p> <p>4.6. Podslužby služby Provoz a zjištění stavu zařízení (Housekeeping) - Diagnostické hlášení (Diagnostic reporting) 14</p> <p>4.7. Podslužby služby Provoz a zjištění stavu zařízení (Housekeeping) - Konfigurace funkčního hlášení parametrů (Parameter functional reporting configuration) 15</p> <p>4.8. Podslužby služby Hlášení událostí (Event reporting) - Hlášení událostí (Event reporting) 16</p> <p>4.9. Podslužby služby Správa paměti (Memory management) - Správa paměti nezpracovaných dat (Raw data memory management) 16</p> <p>4.10. Podslužby služby Správa paměti (Memory management) - Správa paměti strukturovaných dat (Structured data memory management) 17</p>	<p>2.1. Parametry orbity3</p> <p>2.2. Výšková klasifikace geocentrických oběžných drah5</p> <p>3.1. Komunikace mezi vesmírným a pozemním segmentem6</p> <p>4.1. Komunikace mezi vesmírným a pozemním segmentem PUS service systému9</p> <p>6.1. Externí nástroje aplikace 24</p> <p>6.2. Diagram použití kompilátoru ASN1SCC 26</p> <p>7.1. Funkcionality aplikace 28</p> <p>7.2. Diagram propojení formulářů aplikace 30</p> <p>7.3. Externí části programu 33</p> <p>7.4. a) Logo ikony aplikace b) Logo zobrazené v horní liště aplikace 34</p> <p>7.5. a) Barva #4C6173 b) Barva #FFFFFFE1 c) Barva #D9D1BA 35</p> <p>8.1. Hlavní formulář 36</p> <p>8.2. Panel správy paketů TC 39</p> <p>8.3. Panel správy paketů TM 39</p> <p>8.4. Formulář výběru typu služby a podslužby 40</p> <p>8.5. Formulář specifikace informací protokolu při definování dat 41</p> <p>8.6. Formulář specifikace informací protokolu po nadefinování obsahu paketu 41</p> <p>8.7. Formulář volby názvu datové kombinace 42</p> <p>8.8. Formulář zobrazení informací TC paketu 43</p> <p>8.9. Dokumentace 44</p> <p>8.10. Formulář komunikace se satelitem 45</p> <p>8.11. Formát uložených testovacích hodnot TC paketů do textového souboru 45</p> <p>8.12. Formulář specifikace hodnot dat TC paketů 46</p> <p>8.13. Formulář volby IP adresy a portu 47</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4.11.	Podslužby služby Správa paměti (Memory management) - Společná správa paměti (Common memory management)	17
4.12.	Podslužby služby Správa paměti (Memory management) - Konfigurace paměti (Memory configuration)	17
4.13.	Podslužby služby Správa času (Time management) - Kontrola správy času (Time reporting control)	17
4.14.	Podslužby služby Správa času (Time management) - Správa času (Time reporting) .	18
4.15.	Podslužby služby Test (test) - Test (Test)	18
5.1.	Simple typy	19
5.2.	Structured typy	20
5.3.	Encoding hodnoty	21
5.4.	Endianness hodnoty	21
9.1.	Testovací scénář	53
9.2.	Testování hodnoty T-TC-6-2-MemLoad	54
9.3.	Testování hodnoty T-TC-6-4-MemDump	54
9.4.	Testování hodnoty T-TC-6-8-MemCheck	54
9.5.	Testování hodnoty T-TC-6-134-FlashMassErase	54
9.6.	Testování hodnoty T-TC-6-136-BankSwapEnable	54
9.7.	Testování hodnoty T-TC-6-138-BankSwap	55
9.8.	Testování hodnoty T-TC-6-144-SysMemParamSet	55
9.9.	Testování hodnoty T-TC-6-146-MemCheckStart	55
9.10.	Testování hodnoty T-TC-6-148-MemCopyStart	55
9.11.	Testování hodnoty T-TC-6-160-SysMemParamRequest	55
9.12.	Odpověď T-TM-6-3-MemLoadResponse	56
8.14.	Diagram metod a jejich spojení python kódu komunikace	48
8.15.	Formát uložených dekódovaných TM paketů v textovém souboru	50
9.1.	VZLUSAT-2	51
9.2.	Space Dosimetry System Demonstrator (2SD)	52
9.3.	Ukázka python kódu pro komunikaci se zařízením	56
9.4.	Vizualizace přijatých odpovědí ve formě TM paketů	58

9.13.	Odpověď T-TM-6-5- MemDumpResponse	57
9.14.	Odpověď T-TM-6-9- MemCheckResponse	57
9.15.	Odpověď T-TM-6-135- FlashMassEraseResponse	57
9.16.	Odpověď T-TM-6-137- BankSwapEnableResponse.....	57
9.17.	Odpověď T-TM-6-145- SysMemParamSetResponsee...	57
9.18.	Odpověď T-TM-6-147- MemCheckStartResponse	57
9.19.	Odpověď T-TM-6-149- MemCopyStartResponse	58
9.20.	Odpověď T-TM-6-161- SysMemParamResponse	58
B.1.	Elektronické přílohy	66

Kapitola 1

Úvod

Satelitem se rozumí objekt (měsíc, planeta či stroj) ve vesmíru, který obíhá kolem většího objektu (planety nebo hvězdy). Satelity se dělí podle způsobu vzniku na dva druhy, na přirozené a umělé. Přirozené vznikly bez lidské pomoci. Můžeme uvést například Měsíc obíhající kolem Země. Přirozených satelitů je celá řada už jen proto, že kolem většiny planet obíhá alespoň jeden měsíc. Naopak umělé satelity vytvořili lidé a mohli tak ovlivnit jejich funkci a vlastnosti.

Aby mohly satelity vykonávat své funkce, můžeme jmenovat odvětví telekomunikací, navigace či průzkumy vesmíru a Země, musí s nimi navázat pozemní stanice kontakt. V minulosti se komunikovalo se satelitem hlavně pomocí hardwaru odečítáním analogových kanálů a stav byl vyčítán z registrů nebo z relé, které byly dále vzorkovány podle vzorů a umístěny fixně do formátu telemetrie. V osmdesátých letech se začalo přistupovat k používání softwaru na palubě satelitů. Software musel být však stále kontrolován z pozemní stanice, což stále nevedlo k plné flexibilitě a autonomii komunikace. Pro tyto účely je paluba satelitů nyní vybavena počítačem Command and Data Handling (C&DH), který zajišťuje přijímání a odesílání zpráv.[1] Pro zprávy, které jsou satelitu vysílány, se používají protokoly telecomand (TC) a naopak pro zprávy, které se přijímají se používají protokoly telemetry (TM). Protokoly zajišťují obousměrný komunikační tok s vesmírným prostředkem. Protokoly se používají za účelem monitorování a řízení satelitu.

Aby bylo zajištěno jednotnosti a spolehlivosti technologií, návrhových a konstrukčních metod, začala se i ve vesmírném sektoru používat standardizace. V Evropě se za standardizaci zasadila Evropská vesmírná agentura (ESA), s Eurospace vytvořila European Cooperation for Space Standardization (ECSS). S doporučením Meziagenturního poradního výboru pro vesmírné datové systémy (Consultative Committee for Space Data Systems = CCSDS) vznikly standardy pro paketovou komunikaci - telemetrie a telecomand, jako například standard ECSS-E-ST-70-41C. Standard se zabývá použitím telecomand a telemetries paketů pro vzdálené monitorování a řízení subsystémů a přístrojů užitečného zatížení (payloads) na kosmických lodích. Tvorba zpráv je velmi náročný proces, kdy je nutné obsah a formát zpráv zcela definovat. Za tímto účelem byl vytvořen standard a jazyk ASN.1, který určuje obsah zprávy. K jazyku ASN.1 vznikl doprovodný jazyk ACN, který k obsahu zpráv definuje jejich formát.

Celá realizace komunikace byla pro programátory velmi náročným a zdlouhavým procesem, jelikož musel být nejprve nastudován standard, který je velmi obsáhlý a samotné komunikační zprávy musely být psány ručně, což vedlo k chybám. Tyto problémy měly často za následek zpoždování projektů a velkou náročnost projektové realizace. Za účelem pomoci a eliminace chyb vznikla aplikace, jejímž návrhem, implementací a otestováním se diplomová práce zabývá.

Úkolem diplomové práce je seznámit se s problematikou specifikace komunikačních protokolů, které slouží ke komunikaci s obecným přístrojem na palubě družice. Dále je cílem navrhnout a realizovat softwarový nástroj v programovacím jazyce C#, kterým bude možné komunikační protokoly specifikovat a s přístrojem na palubě družice i ko-

munikovat. Dalším cílem je navržení propojení softwarového nástroje s kompilátorem `asn1sc` ze sady nástrojů ESA-TASTE a vygenerovat zdrojové kódy v jazyce Python pro sestavení a dekodování protokolů TC a TM. Navrhnout čitelný zdrojový kód v jazyce Python a otestovat funkčnost softwarového nástroje.

Při návrhu softwarového nástroje byl kladen důraz především na jeho funkčnost, protože takový nástroj nejen ulehčí definování komunikačních protokolů a propojení s přístroji na palubě družice, ale i zamezí vzniku chyb při ručním vytváření definic komunikačních protokolů.

Kapitola 2

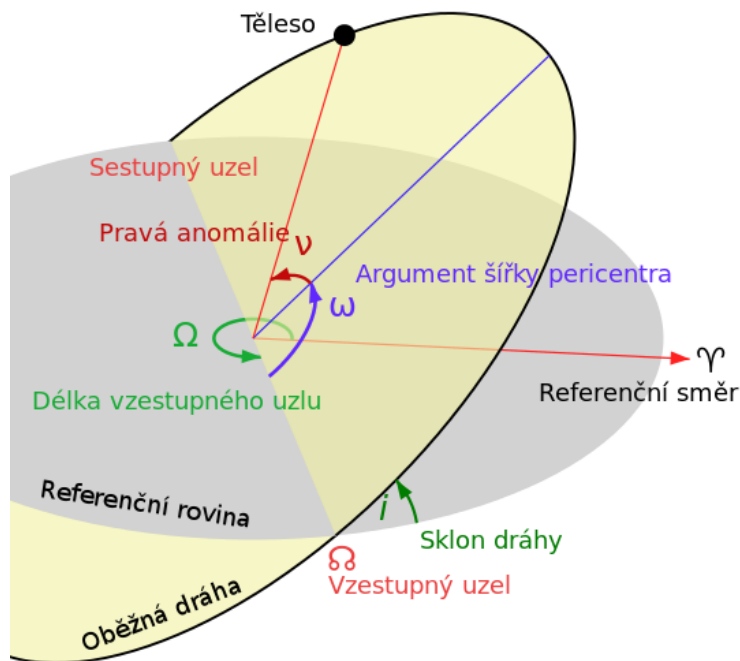
Satelity

Satelitem se rozumí objekt (měsíc, planeta či stroj) ve vesmíru, který obíhá kolem většího objektu (planety nebo hvězdy). Satelity dělíme na dva druhy, podle způsobu vzniku, na přirozené a umělé.

Umělé satelity začaly vznikat až v polovině 20. století. První byla družice Sputnik, která byla na eliptickou nízkou oběžnou dráhu (LEO) vypuštěna 4. října 1957 v rámci sovětského vesmírného programu. Satelit byl v provozu tři týdny, než došlo k vybití baterií.[2] Umělé i přírodní satelity obíhají kolem centrálního tělesa po drahách zvaných orbitách.

2.1 Orbíta satelitů

Orbíta je zakřivená dráha, po které objekt ve vesmíru obíhá kolem jiného objektu vlivem gravitace a dostatečné hybnosti. Zkoumání orbit započal již v 17. století Johannes Kepler. Satelity jsou na své dráhy vynášeny kvůli různým účelům, jmenujme například navigaci či zajištění telekomunikací. Z toho důvodu potřebují být vyneseny i do rozdílné výšky. Za tímto účelem se začaly rozlišovat orbity kolem Země, které se liší svými parametry, a jsou zobrazeny na obrázku 2.1 s vysvětlením v tabulce 2.1. [3]



Obrázek 2.1. Parametry orbity[4]

Symbol	Název	Vysvětlení
a	velká poloosa	vyjadřuje střední vzdálenost tělesa od těžiště oběžné dráhy
e	excentricita dráhy	představuje poměr velikostí
i	sklon dráhy	vyjadřuje úhel svírající rovinou oběžné dráhy a rovinou souřadné dráhy tělesa
Ω	délka vzestupného úhlu	vyjadřuje rotaci roviny orbity a referenční osy
ω	argument šířky pericentra	vyjadřuje úhel svírající průvodičem tělesa ve vzestupném úhlu a směrem průvodiče tělesa v pericentru své dráhy
ν	pravá anomálie	poloha satelitu na orbitě

Tabulka 2.1. Parametry orbity[5]

2.1.1 Vynesení satelitu na oběžnou dráhu

Satelit na oběžnou dráhu vynáší nosná raketa. Její navrhnutí závisí především na hmotnosti satelitu a na druhu orbity, na kterou bude satelit umístěn. Aby byl satelit umístěn přesně na svou oběžnou dráhu, je raketa řízena pomocí inerciálního naváděcího systému (inertial guidance system = IGS).

IGS monitoruje polohu, rychlost a zrychlení rakety. Tyto údaje jsou získávány z gyroskopů a akcelerometrů a jsou vyhodnocovány a předávány navigačnímu systému rakety. Směrování os gyroskopů zůstává nezměněno a na jejich stabilní platformě jsou umístěny akcelerometry, které zaznamenávají zrychlení ve všech třech osách.

2.1.2 Typy orbitů

Satelity se umísťují nejčastěji na jednu z definovaných a používaných oběžných drah kolem Země, či mohou být vyslány na meziplanetární cestu.

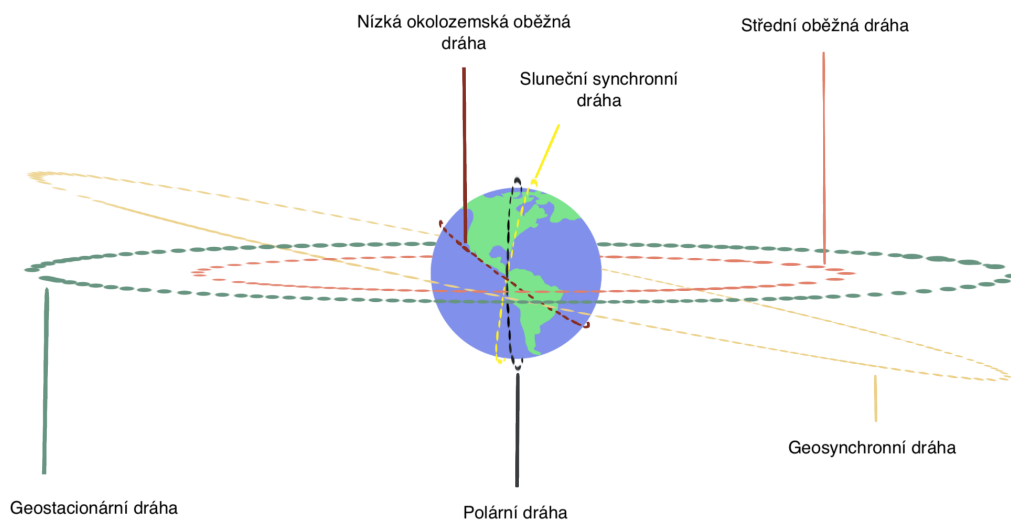
Výšková klasifikace geocentrických oběžných drah

- Geostacionární dráha (GEO) - satelity obíhají nad rovníkem Země. Doba oběhu je 23 hodin 56 minut a 4 sekundy, proto se může zdát, že se satelity nepohybují. Tato orbita se používá pro telekomunikační a pozorovací satelity. Geostacionární orbita se nachází ve výšce 42 164 km od středu Země.[6]
- Nízká okolozemská oběžná dráha Země (LEO) - Nízká oběžná dráha se nachází ve výšce do 1 000 km. Tato dráha je využívána pro dálkový průzkum Země a pro telekomunikace. Satelity se zde pohybují rychlostí okolo 7,8 km za sekundu. Oblet Země tak satelitu trvá asi 90 minut. Při takto velké rychlosti oběhu se pro telekomunikační účely využívá kombinace či konstelace více satelitů, protože by použití pouze jednoho satelitu vyžadovalo velkého úsilí.
- Střední oběžná dráha Země (MEO) - Střední oběžná dráha se nachází mezi GEO a LEO oběžnou dráhou. Tato dráha je využívána satelity pro navigaci. Jsou zde umístěny satelity amerického globálního navigačního systému (GPS) a evropského systému Galileo.
- Polární a sluneční synchronní dráha (SSO) - Polární oběžná dráha svírá úhel do 30 stupňů od zemských pólů. Sluneční synchronní dráhy jsou zvláštním druhem polárních drah a jsou synchronní se Sluncem, proto obíhají kolem Země vždy ve stejný

čas. Tyto dráhy se používají pro meteorologické satelity a pro satelity, které měří atmosférické podmínky a dlouhodobě pozorují Zemi. Satelity se nejčastěji pohybují na polárních drahách ze severu na jih a procházejí v blízkosti zemských pólů. Polární dráhy jsou v nadmořské výšce mezi 200 a 1000 km a proto se řadí mezi LEO dráhy.[7]

- Přenosové dráhy a geostacionární přenosová dráha (GTO) - přenosové dráhy slouží k přechodu satelitů mezi drahami. K dosažení konečné dráhy GEO je využívána nejběžněji používaná geostacionární přenosová dráha.[8]
- Lagrangeovy body (L-body) - Lagrangeovy body soustavy Slunce-Země se nacházejí ve vesmíru daleko (přes milion kilometrů) od zemského povrchu. V Lagrangeových bodech se přitažlivé síly a odstředivé síly působící na malé těleso vyrovnají a umožní tak stabilní oběh malých objektů, satelitů. Nejběžněji používanými body jsou L1 a L2 ve vzdálenosti asi 1,5 milionu km od Země.[8]

Výše popsaná klasifikace geocentrických oběžných drah je vyobrazena na obrázku 2.2.



Obrázek 2.2. Výšková klasifikace geocentrických oběžných drah

2.2 Funkce umělých satelitů

Jak již bylo zmíněno, satelity většinou obíhají planety nebo hvězdy po orbitách, které jsou v případě satelitů obíhající naší planetu od povrchu Země vzdáleny většinou méně než 1000 km. Tento pohled na zemský povrch satelitům umožňuje velký prostor pro zaznamenání a tím získání velkého množství informací najednou. Kromě pohledu na Zem mají také výhodu pro snímání vesmíru, protože na rozdíl od snímání ze zemského povrchu se pohybují nad zemskou atmosférou, která absorbuje například UV či rentgenové záření.[9] Funkce satelitů jsou nejvíce využívány za účely telekomunikace, navigace, meteorologie a vědeckého snímání Země.[10]

Kapitola 3

Komunikace s kosmickým prostředkem

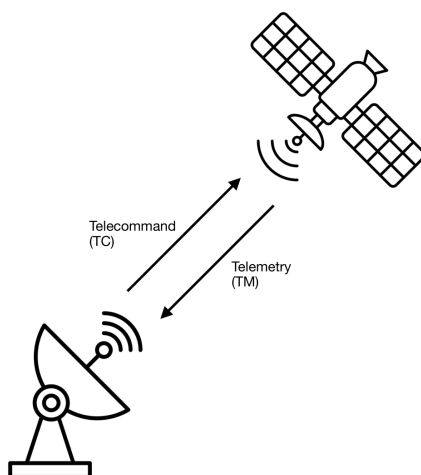
Složitější kosmické prostředky potřebují udržovat kontakt s pozemní stanicí, která zajišťuje především řízení. Za účelem komunikace je na palubě kosmického prostředku systém Command and Data Handling (C&DH), který zajišťuje správu zpráv ve formě paketů.

3.1 Systém Command and Data Handling (C&DH)

Systém Command and Data Handling (C&DH) na kosmickém prostředku zajišťuje správu dat z hlediska odeslání a jejich přijetí. Command and Data Handling Unit (C&DH Unit) se také stará o vzájemnou komunikaci přístrojů na kosmickém prostředku.[1]

3.1.1 Funkce systému C&DH, TC a TM komunikace

Systém C&DH je připojen k jednotce pro bezdrátový přenos dat. Komunikační protokoly, které se používají, komunikují přes space link. Data, která jsou vysílána, používají protokoly telemetry telecommand (TC) a (TM). Tím protokoly zajišťují obousměrný komunikační tok s kosmickým prostředkem. Tento komunikační tok je vyobrazen na obrázku 3.1. TC zprávy systém C&DH přijme, dekóduje a nasměruje k systémům, kterým jsou zprávy určeny. TC se dělí hlavně na přímé příkazy kosmického prostředku pro rekonfiguraci a příkazy specifické pro aplikaci.



Obrázek 3.1. Komunikace mezi vesmírným a pozemním segmentem

Naopak TM zprávy systém C&DH shromažďuje ze systémů na vesmírné stanici a kóduje je pro přenos na zemskou stanici. TM se rozděluje na údaje o kosmickém prostředí, o oběžné dráze (poloze), o payloads (užitečných zatíženích), o stavu příjmu telecommand (CLCW) a o datech výpisu paměti.

Kapitola 4

Zavedení normalizace a standardizace ve vesmírném sektoru

Standardy jsou důležité k zajištění bezpečnosti a spolehlivosti technologií, materiálů, návrhových a konstrukčních metod. Používání standardů je nyní ve vývoji vesmírných technologií vyžadováno. Výhodou jejich používání je zajištění kompatibility technologií od různých výrobců.

4.1 Standardizace vesmírného sektoru v Evropě

Standardizací vesmírného sektoru v Evropě se zabývá Evropská vesmírná agentura (ESA), která společně s dalšími vesmírnými agenturami a organizací Eurospace vytvořila European Cooperation for Space Standardization (ECSS), která má za cíl vytvořit a nastavit normy pro technologie ve vesmírném sektoru.[11]

4.1.1 Evropská vesmírná agentura (ESA)

O založení vesmírné agentury začala Evropa uvažovat po konci druhé světové války. Inženýři z Evropy do té doby museli pracovat ve space průmyslu Spojených států amerických nebo Sovětského svazu. S nápadem založit vesmírnou agenturu přišli za vládami evropských států v roce 1958 Pierre Auger a Edoardo Amaldi. V roce 1964 založili European Launch Development Organisation (ESRO). Ta měla za úkol provádět výzkum, stavět vesmírné lodě a nosné rakety.[12]

ESA vznikla sloučením předem zmíněné ESRO a ELDO (European Launcher Development Organisation). Agentura měla pouze 10 zakládajících členských států. Nyní má ESA 22 členských států. Jednotlivá centra lze najít ve více evropských státech, ale hlavní sídlo je v Paříži.[13]

4.1.2 European Cooperation for Space Standardization (ECSS)

ECSS vznikla s cílem nastavit a vytvořit normy pro technologie, které jsou používány ve vesmírném segmentu. Na tvorbě norem a vývoji vesmírných technologií se podílí významné evropské firmy, jmenujme například ESA a organizaci Eurospace. ECSS vznikla v roce 1993 na podnět organizace Eurospace. Začala vesmírné technologie normalizovat a přispěla k eliminaci problémů při plnění harmonogramů.[14] Oficiálně byla Evropskou vesmírnou agenturou přijata 23. června 1994 rezolucí ESA/C/CXIII/Res.1.

ECSS standardizuje oblasti zajištění produktů (product assurance), udržitelnosti (space sustainability), managementu a inženýrství. Standardizace probíhá třemi různými typy dokumentů - standardy, příručkami (handbooks) a technickými memorandy (technical memoranda). Dva poslední zmíněné dokumenty neslouží k normování technologií, ale tvoří pouze doplňující informace. Handbooks a technická memoranda se používají pouze v oblastech inženýrství a zajištění produktů. Nyní má ECSS 140 aktivních standardů.[15]

4.2 Standardy pro paketovou komunikaci

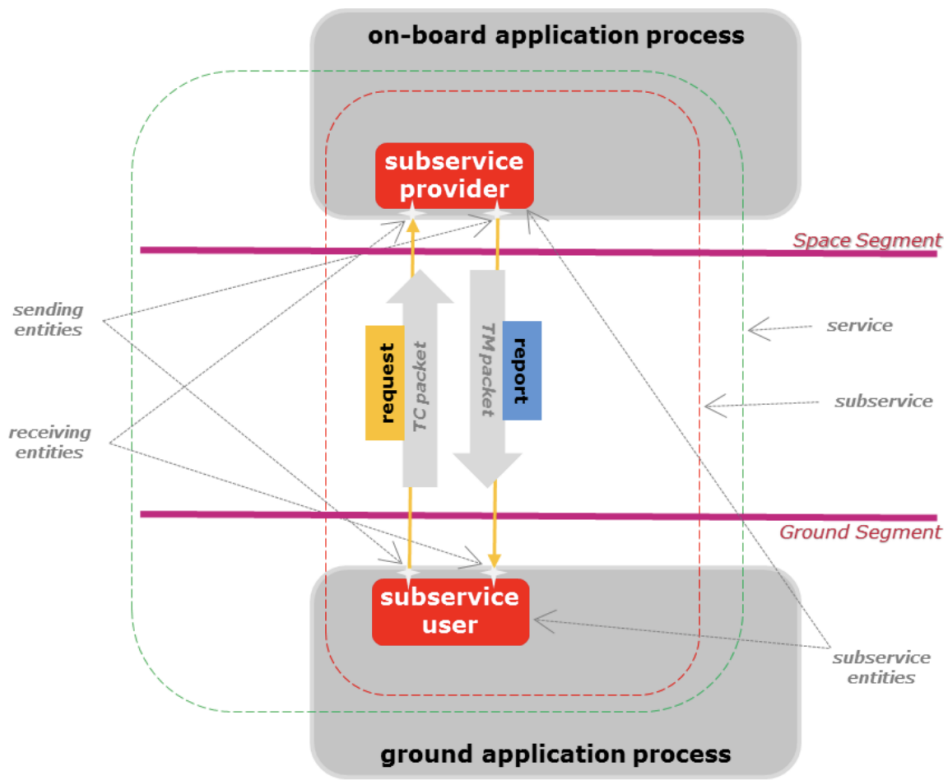
ESA s doporučením Meziagenturního poradního výboru pro vesmírné datové systémy (Consultative Committee for Space Data Systems = CCSDS) vyvinula standardy pro paketovou komunikaci - telecommand a telemetry.

4.2.1 Standard ECSS-E-ST-70-41C

Standard ECSS-E-ST-70-41C se zabývá použitím telecommand a telemetry paketů pro vzdálené monitorování a řízení subsystémů a užitečných zařízení (payloads) na kosmických prostředcích. Pro konkrétní mise lze tento standard po úpravách použít.

Standard zavádí koncept modelu služeb a podslužeb (service a subservice) standardu Packet Utilization Standard (PUS). Zabývá se službami a jim příslušnými podslužbami, které splňují provozní požadavky na monitorování a řízení kosmického prostředku. Specifikuje obsah paketů telecommands (požadavky) a telemetries (přenos zpráv). Komunikace je zobrazena na obrázku 4.1. Telecommand ani telemetry pakety neobsahují zvukové či obrazové informace.

Každá služba se zabývá jinými úkoly a má na starost úkoly různé složitosti. Proto obsahuje minimální soubory informací a dále lze doplnit volitelné/rozšiřitelné informace. Standard se zabývá hlavně požadavky na kosmický prostředek, ale i požadavky na pozemní segment. Z těchto požadavků lze především odvodit, jak by mělo probíhat testování nebo provoz kosmického prostředku.[16]



Obrázek 4.1. Komunikace mezi vesmírným a pozemním segmentem PUS service systému[16]

Služba 17 - Test (Test)

Testovací služba umožňuje spustit testovací funkce na palubě satelitu a jejich výsledky pomocí TM zpráv zaslat pozemní stanici. Jeden z testů je například test “Jsi naživu?”.

ID	Název typu služby
1	Ověření požadavku (Request verification)
2	Přístup k zařízení (Device access)
3	Provoz a zjištění stavu zařízení (Housekeeping)
4	Hlášení statistik parametrů (Parameter statistics reporting)
5	Hlášení událostí (Event reporting)
6	Správa paměti (Memory management)
7	Rezervováno (Reserved)
8	Správa funkcí (Function management)
9	Správa času (Time management)
10	Rezervováno (Reserved)
11	Časové plánování (Time-based scheduling)
12	Monitorování na palubě kosmického prostředku (On-board monitoring)
13	Přenos velkých paketů (Large packet transfer)
14	Ovládání předávání v reálném čase (Real-time forwarding control)
15	Skladování a vyhledávání dat na palubě kosmického prostředku (On-board storage and retrieval)
16	Rezervováno (Reserved)
17	Test (Test)
18	Ovládání procedur na palubě kosmického prostředku (On-board control procedure)
19	Akce událostí (Event-action)
20	Správa parametrů (Parameter management)
21	Sekvenování požadavků (Request sequencing)
22	Plánování na základě pozice (Position-based scheduling)
23	Správa souborů (File management)

Tabulka 4.1. Typy služeb

Podslužba (Subservice)

Podslužba je základní složka služby složená z přesně jednoho poskytovatele dílčích služeb (providera) a souvisejících uživatelů (users) dílčích podslužeb, kteří komunikují prostřednictvím vyhrazených sad zpráv. Má na starost provádění některých funkcí systému. Při specifikaci podslužby musí být deklarováno, zda je povoleno více realizací tohoto typu v rámci jedné služby.

Tento standard používá dva typy identifikačních mechanismů:

- jména pro lidskou komunikaci
- identifikátory pro komunikaci s kosmickou lodí

Poskytovatel dílčích služeb (provider) je obvykle hostován aplikačními procesy na vesmírném zařízení a uživatel obsluhuje pozemní stanici. Standard tyto role a vztahy neomezuje.

Standard též nespecifikuje, jak jsou požadavky, pokyny, zprávy a oznámení implementovány na palubě kosmického prostředku nebo na pozemní stanici. Rovněž nespecifikuje, kdo je zodpovědný za kódování a dekodování paketů telemetrie a paketů telecommand.

Aplikační proces může být implementován v softwaru, firmwaru nebo hardwaru. Počítač kosmického prostředku může hostit jeden nebo více aplikačních procesů nebo naopak může být aplikační proces distribuován mezi dva nebo více palubních počítačů.

Informace jakožto zpráva, je přenášena v nezměněné podobě přenosovým protokolem, který spojuje uživatele dílčích služeb a poskytovatele dílčích služeb.

Zpráva zaslaná uživatelem poskytovateli za účelem vyvolání provedení palubních činností se nazývá request (požadavek). Každý požadavek obsahuje jednu nebo více instrukcí pro každou aktivitu, která se má provést. Zpráva zaslaná poskytovatelem uživateli se nazývá report.

Přehled podslužeb nejvíce využívaných služeb s ID 1, 3, 5, 6, 9 a 17, je uveden v tabulkách 4.2 až 4.15.

ID	Název podslužby	Protokol
1	Zpráva o ověření úspěšného přijetí (Successful acceptance verification report) - Přijímání a podávání zpráv	TM
2	Zpráva o ověření neúspěšného přijetí (Failed acceptance verification report)	TM

Tabulka 4.2. Podslužby služby Ověření požadavku (Request verification) - Přijímání a podávání zpráv (Acceptance and reporting)

ID	Název podslužby	Protokol
3	Zpráva ověřování úspěšného zahájení provedení (Successful start of execution verification report)	TM
4	Zpráva o ověření spuštění se nezdařila (Failed start of execution verification report)	TM
5	Úspěšný průběh zprávy o ověření provedení (Successful progress of execution verification report)	TM
6	Neúspěšný průběh zprávy o ověření provedení (Failed progress of execution verification report)	TM
7	Úspěšné dokončení zprávy o ověření provedení (Successful completion of execution verification report)	TM
8	Neúspěšné dokončení zprávy o ověření provedení (Failed completion of execution verification report)	TM

Tabulka 4.3. Podslužby služby Ověření požadavku (Request verification) - Hlášení exekuce (Execution reporting)

ID	Název podslužby	Protokol
10	Zpráva o ověření směrování se nezdařila (Failed routing verification report)	TM

Tabulka 4.4. Podslužby služby Ověření požadavku (Request verification) - Směrování a hlášení (Routing and reporting)

ID	Název podslužby	Protokol
1	Hlášení provozu vytvořené struktury sestavy parametrů (Create a housekeeping parameter report structure)	TC
3	Hlášení provozu o odstranění struktury sestavy parametrů zprávy (Delete housekeeping parameter report structures)	TC
5	Hlášení provozu o povolení pravidelného generování zpráv o parametrech údržby (Enable the periodic generation of housekeeping parameter reports)	TC
6	Hlášení provozu o zakázání pravidelného generování zpráv o parametrech provozu (Disable the periodic generation of housekeeping parameter reports)	TC
9	Hlášení provozu o struktuře sestavy parametrů sestav (Report housekeeping parameter report structures)	TC
10	Hlášení provozu o struktuře sestavy parametrů (Housekeeping parameter report structure report)	TM
25	Hlášení o parametrech provozu (Housekeeping parameter report)	TM
27	Hlášení provozu o vygenerování jednorázové zprávy pro struktury zpráv o provozních parametrech (Generate a one shot report for housekeeping parameter report structures)	TC
29	Hlášení provozu o připojení parametrů ke struktuře sestavy parametrů zprávy (Append parameters to a housekeeping parameter report structure)	TC
31	Hlášení provozu o úpravě intervalu sběru struktur sestavy parametrů zprávy (Modify the collection interval of housekeeping parameter report structures)	TC
33	Hlášení provozu o vlastnostech periodického generování struktur sestavy parametrů zprávy (Report the periodic generation properties of housekeeping parameter report structures)	TC
35	Hlášení provozu o údržbě parametrů periodické zprávy o vlastnostech generování (Housekeeping parameter report periodic generation properties report)	TM

Tabulka 4.5. Podslužby služby Provoz a zjištění stavu zařízení (Housekeeping) - Hlášení provozu (Housekeeping reporting)

ID	Název podslužby	Protokol
2	Vytvoření struktury sestavy diagnostických parametrů (Create a diagnostic parameter report structure)	TC
4	Hlášení o odstranění struktury zpráv o diagnostických parametrech (Delete diagnostic parameter report structures)	TC
7	Hlášení o povolení pravidelného generování zpráv o diagnostických parametrech (Enable the periodic generation of diagnostic parameter reports)	TC
8	Hlášení o zakázání pravidelného generování zpráv o diagnostických parametrech (Disable the periodic generation of diagnostic parameter reports)	TC
11	Hlášení o struktuře zpráv o diagnostických parametrech (Report diagnostic parameter report structures)	TC
12	Hlášení o zprávě o struktuře zprávy o diagnostických parametrech (Diagnostic parameter report structure report)	TM
26	Hlášení o zprávě o diagnostických parametrech (Diagnostic parameter report)	TM
28	Hlášení o vygenerování jednorázové zprávy pro struktury zpráv o diagnostických parametrech (Generate a one shot report for diagnostic parameter report structures)	TC
30	Hlášení o připojení parametrů ke struktuře sestavy diagnostických parametrů (Append parameters to a diagnostic parameter report structure)	TC
32	Hlášení o úpravách intervalu sběru struktur zpráv o diagnostických parametrech (Modify the collection interval of diagnostic parameter report structures)	TC
34	Hlášení o reportování vlastností periodického generování struktur zpráv diagnostických parametrů (Report the periodic generation properties of diagnostic parameter report structures)	TC
36	Hlášení o diagnostických parametrech periodické zprávy o vlastnostech generování (Diagnostic parameter report periodic generation properties report)	TM

Tabulka 4.6. Podslužby služby Provoz a zjištění stavu zařízení (Housekeeping) - Diagnostické hlášení (Diagnostic reporting)

ID	Název podslužby	Protokol
37	Použití konfigurace sestavování funkčních parametrů (Apply parameter functional reporting configurations)	TC
38	Vytvoření definice parametrického funkčního výkaznictví (Create a parameter functional reporting definition)	TC
39	Odstranění definice funkčních sestav parametrů (Delete parameter functional reporting definitions)	TC
40	Nadefinování funkčních sestav parametrů sestav (Report parameter functional reporting definitions)	TC
41	Hlášení o definici funkce parametru (Parameter functional reporting definition report)	TM
42	Přidávání definice sestavy parametrů do definice sestavování funkčních parametrů (Add parameter report definitions to a parameter functional reporting definition)	TC
43	Odebírání definice sestavy parametrů z definice sestavování funkčních parametrů (Remove parameter report definitions from a parameter functional reporting definition)	TC
44	Upravování vlastnosti periodického generování definic sestavy parametrů definice funkčních sestav parametrů (Modify the periodic generation properties of parameter report definitions of a parameter functional reporting definition)	TC

Tabulka 4.7. Podslužby služby Provoz a zjištění stavu zařízení (Housekeeping) - Konfigurace funkčního hlášení parametrů (Parameter functional reporting configuration)

ID	Název podslužby	Protokol
1	Informativní zpráva o akci (Informative event report)	TM
2	Zpráva o anomálii nízké závažnosti (Low severity anomaly report)	TM
3	Zpráva o anomálii střední závažnosti (Medium severity anomaly report)	TM
4	Zpráva o anomálii s vysokou závažností (High severity anomaly report)	TM
5	Povolit vytváření sestav definic událostí (Enable the report generation of event definitions)	TC
6	Zakázat generování sestav definic událostí (Disable the report generation of event definitions)	TC
7	Nahlásit seznam definic zakázaných událostí (Report the list of disabled event definitions)	TC
8	Zakázaná sestava seznamu definic událostí (Disabled event definitions list report)	TM

Tabulka 4.8. Podslužby služby Hlášení událostí (Event reporting) - Hlášení událostí (Event reporting)

ID	Název podslužby	Protokol
2	Načíst oblasti s nezpracovanými daty paměti (Load raw memory data areas)	TC
5	Vypsát nezpracovaná data z paměti (Dump raw memory data)	TC
6	Zpráva o vyřazených nezpracovaných datech paměti (Dumped raw memory data report)	TM
9	Zkontrolování nezpracovaných dat paměti (Check raw memory data)	TC
10	Zpráva o zkontrolovaných nezpracovaných datech paměti (Checked raw memory data report)	TM
11	Načtení nezpracované oblasti atomických dat paměti v nepřerušitelné transakci (Load a raw memory atomic data area in a non-interruptible transaction)	TC
19	Načtení oblasti nezpracovaných dat paměti odkazem (Load raw memory data areas by reference)	TC
20	Vypsání oblasti nezpracovaných dat paměti do souboru (Dump raw memory data areas to file)	TC

Tabulka 4.9. Podslužby služby Správa paměti (Memory management) - Správa paměti nezpracovaných dat (Raw data memory management)

ID	Název podslužby	Protokol
1	Načíst data paměti objektů (Load object memory data)	TC
3	Vypsát data paměti objektů (Dump object memory data)	TC
4	Zpráva s daty paměti objektů (Dumped object memory data report)	TM
7	Zkontrolování dat paměti objektů (Check object memory data)	TC
8	Zpráva s daty zkontrolované paměti objektů (Checked object memory data report)	TM
17	Zkontrolování objektu paměti objektu (Check an object memory object)	TC
18	Zpráva o objektu zkontrolované paměti objektů (Checked object memory object report)	TM
21	Načtení datové oblasti paměti objektů odkazem (Load object memory data areas by reference)	TC
22	Vypsání datové oblasti paměti objektů do souboru (Dump object memory data areas to file)	TC

Tabulka 4.10. Podslužby služby Správa paměti (Memory management) - Správa paměti strukturovaných dat (Structured data memory management)

ID	Název podslužby	Protokol
12	Zrušení všech výpisů paměti (Abort all memory dumps)	TC

Tabulka 4.11. Podslužby služby Správa paměti (Memory management) - Společná správa paměti (Common memory management)

ID	Název podslužby	Protokol
13	Povolení čištění paměti (Enable the scrubbing of a memory)	TC
14	Zakázání čištění paměti (Disable the scrubbing of a memory)	TC
15	Povolení ochrany paměti proti zápisu (Enable the write protection of a memory)	TC
16	Vypnutí ochrany paměti proti zápisu (Disable the write protection of a memory)	TC

Tabulka 4.12. Podslužby služby Správa paměti (Memory management) - Konfigurace paměti (Memory configuration)

ID	Název podslužby	Protokol
1	Nastavení rychlosti generování časové zprávy (Set the time report generation rate)	TC

Tabulka 4.13. Podslužby služby Správa času (Time management) - Kontrola správy času (Time reporting control)

ID	Název podslužby	Protokol
2	CUC zpráva o čase (CUC time report)	TM
3	CDS zpráva o čase (CDS time report)	TM

Tabulka 4.14. Podslužby služby Správa času (Time management) - Správa času (Time reporting)

ID	Název podslužby	Protokol
1	Provedení testu spojení “Je naživu?“ (Perform an are-you-alive connection test)	TC
2	Zpráva o testu připojení “Je naživu?“ (Are-you-alive connection test report)	TM
3	Provedení testu připojení na desce (Perform an on-board connection test)	TC
4	Zpráva o testu palubního připojení (On-board connection test report)	TM

Tabulka 4.15. Podslužby služby Test (Test) - Test (Test)

Kapitola 5

Tvorba komunikačních zpráv

Obsah zpráv je nedílnou součástí komunikace s vesmírnými prostředky. Pro jejich zpracování, zakódování a dekodování je potřeba znát také jejich formát. Za tímto účelem se začal používat jazyk ASN.1, který definuje obsah zpráv a jazyk ACN, který doplňuje ASN.1 a určuje formát zpráv.

5.1 Abstract Syntax Notation One (ASN.1)

Abstract Syntax Notation One (ASN.1) je komplementární jazyk, taktéž standard ITU-T/ISO, který byl zaveden a definován v roce 1984 International Telecommunication Union Telecommunication Standardization Sector (ITU-T). Zabývá se formálním vyjádřením zpráv, které si rozhraní předávají.

Jazyk popisuje datové typy a rozsahy dat, které jsou formulovány, tedy není uvedeno konkrétní kódování. Používá se hojně v letectví, počítačových sítích či telekomunikacích. Jazyk je nezávislý na programovacím jazyce. Jeho podstatou je popis datových struktur, které vyjadřují zprávy, tedy zápis bitů a bajtů zprávy.[20]

ASN.1 specifikuje metody kódování, jmenujme například:

- Basic Encoding Rules (BER) - dokument X.690
- Canonical Encoding Rules (CER) - modifikace BER
- Distinguished Encoding Rules (DER) – modifikace BER
- Packed Encoding Rules (PER) – dokument X.691
- XML Encoding Rules (XER) – dokument X.693, X.694

Vlastnosti ASN.1

Vlastnost - Simple Type

Simple typy	Popis
INTEGER	Reprezentuje celá čísla.
BOOLEAN	Reprezentuje logické hodnoty pravda (true) a nepravda (false). Nepravda (false) je reprezentována hodnotou 0 a pravda (true) hodnotou 1.
OCTET STRING	Reprezentuje řetězec bajtů neboli řetězec, který se skládá ze sekvencí osmi bitů.
IA5String	Reprezentuje omezenou množinu znaků podle normy ISO 646.
NumericString	Reprezentuje řetězec čísel od 0-9 bez omezení.
BIT STRING	Reprezentuje datovou strukturu pole bitů.
REAL	Reprezentuje čísla racionální (zapsána ve zlomku) a iracionální (ostatní), či na algebraická a transcendentní.
ENUMERATED	Reprezentuje výčet/sadu hodnot.

Tabulka 5.1. Simple typy

Vlastnost - Structured Type

Structured type	Popis
SEQUENCE	Zastupuje uspořádanou kolekci proměnných různého datového typu.
SEQUENCE OF	Zastupuje uspořádanou kolekci proměnných stejného datového typu.
SET	Zastupuje neuspořádanou kolekci proměnných různého datového typu.
SET OF	Zastupuje neuspořádanou kolekci proměnných stejného datového typu.
CHOICE	Zastupuje kolekci odlišných datových typů, ze kterých se může vybrat pouze jedna možnost.
SELECTION	Vybere typ komponenty z nabídky CHOICE.
ANY	Aplikace určí typ sama.

Tabulka 5.2. Structured typy

5.1.1 ACN

Jazyk ACN umožňuje definovat bitové kódování zpráv, jehož struktura je přehledná a člověkem čitelná a je označován doprovodným jazykem ASN.1. Je to zjednodušená verze ECN (Encoding Control Notation). Příkazy se vždy vztahují na stejnojmenný soubor ASN.1 k datům/typům definovaným v ASN.1. Soubor ASN.1 zůstane tedy čistý od samostatného kódování.[21]

Vlastnosti jsou přiřazeny v hranatých závorkách. Pokud nejsou vlastnosti v hranatých závorkách doplněny, vlastnosti se buď zdědí od svých základních typů nebo se vyplní automaticky.

Vlastnosti ACN

Vlastnost - size

Kontroluje velikost kódovaných informací.

- Fixní forma (Fixed form) - size x, kde x je kladné číslo udávané délky uživatelem.
- Proměnlivá délka (Variable size with length specified in external field) - je stejná jako v případě fixní formy s rozdílem, že se délka nastavuje na základě gramatiky ACN, tedy není nastavena uživatelem.

Vlastnost - encoding

Vlastnost lze použít pouze na typ INTEGER, ENUMERATED a REAL, IA5String a Numeric string. Hodnoty, které může vlastnost encoding nabývat, jsou uvedeny v tabulce 5.3.

Encoding hodnoty	Použití na typy ASN.1	Popis
post-int	INTEGER, ENUMERATED	Povoleny jsou pouze kladná celá čísla. U kompilace záporného čísla je ohlášena chyba.
twos-complement	INTEGER, ENUMERATED	Reprezentace dvojkového doplňku.
ASCII	INTEGER, ENUMERATED, IA5String, NumericString	Při kódování je nejprve zakódováno ASCII znaménko, až poté čísla.
BCD	INTEGER, ENUMERATED	Povoleny jsou pouze kladná celá čísla. U kompilace záporného čísla je ohlášena chyba.
IEEE754-1985-32	REAL	Reprezentace je dána stejnojmenným standardem.
IEEE754-1985-64	REAL	Reprezentace je dána stejnojmenným standardem.

Tabulka 5.3. Encoding hodnoty

Vlastnost - endianness

Vlastnost lze též použít na typ INTEGER, ENUMERATED a REAL. Navíc musí být size definována na fixní hodnotu. Nejčastěji, když se používá vlastnost endianness, je size rovna 16, 32 či 64 bitům. Hodnoty, které může endianness nabývat, jsou uvedeny v tabulce 5.4.

Endianness hodnoty	Použití na typy ASN.1	Popis
big	INTEGER, ENUMERATED, REAL	Určuje pořadí sekvence bajtů. Nejvýznamnější bajt se uloží s nejmenší paměťovou adresou.
little	INTEGER, ENUMERATED, REAL	Určuje pořadí sekvence bajtů. Nejvýznamnější bajt se uloží s největší paměťovou adresou.

Tabulka 5.4. Endianness hodnoty

Vlastnost - align-to-next

Vlastnost určuje, že budou data kódována na začátku následujícího:

- Slova
align-to-next word – 16 bits
- Bajtu
align-to-next byte - 8 bits

Vlastnost - encode-values

Vlastnost lze použít pouze na typ ENUMERATED, který je kódován výčtem nebo indexy kódovaných hodnot, takto [encode-values].

Vlastnost - true-value, false-value

Vlastnost lze použít pouze na typ BOOLEAN. Použitím určíme, jaká hodnota bude přiřazena, pokud bude vlastnost:

- TRUE - true-value '111'B
- FALSE - false-value '0'B

Vlastnost present-when

Používá se při použití structured type SEQUENCE a CHOICE, takto:
[present-when serviceId==6 subserviceId==3]

Vlastnost determinant

Vlastnost se používá pro jednodušší definování kódování, externím výčtem a pak jen odkazem na název externího výčtu.

Kapitola 6

Použité prostředky

Hlavním výstupem práce je aplikace, která vznikla za použití vývojového prostředí Visual Studio Community Edition (IDE). Pro aplikaci byla využita šablona projektu Windows Forms v .NET framework v programovacím jazyce C#. Aplikace automaticky generuje dokumentaci za použití nástroje LaTeX a pro komunikaci se zařízením generuje soubor v jazyce Python a automaticky jej spustí, navíc využívá i pomocných textových dokumentů. Nesmíme opomenout, že aplikace načítá vygenerované python soubory z kompilátoru ASN1SCC.

6.1 Knihovna tříd Windows Forms

Windows Forms je bezplatná knihovna tříd pro tvorbu grafického rozhraní (GUI), která je součástí Microsoft .NET Framework. Již na konferenci Microsoft Connect byl představen koncept Windows Forms jako open source projekt GitHubu pod licencí MIT, která udává pouze malá omezení, avšak plné znění licence musí být dodáno společně se SW. Windows Forms byly představeny v roce 2001 s první verzí Visual Studio a .NET. Díky návrháři a široké nabídce komponent - ovládacích prvků a funkcí si získal oblibu vývojářů, kteří nemuseli mít pokročilou znalost programování, aby vytvořili aplikaci. Ovládací prvky Windows Forms se skládají z návrháře ovládacích prvků, kde uživatel navrhuje vzhled formulářů aplikace a kódu, který dodává prvkům funkcionalitu.

6.1.1 Důvod vybrání

Důvodem vybrání je skutečnost, že vytvořením nového formuláře jsou jeho soubory přidány do adresářové struktury aplikace. Přehlednost formulářů umožňuje širší větvení a zanořování aplikace. Výhodou je i možnost výběru z více programovacích jazyků z frameworku .NET mimo jiné C#. Windows Forms umožňuje intuitivní tvorbu aplikace, neboli drag-and-drop návrh komponent aplikace.

6.1.2 Vytvoření formuláře

Použité komponenty jsou instancemi třídy formuláře, ve kterém jsou umístěny a jejich funkčnost umožňují metody, které jsou k prvku přiřazeny a tím umožňují uživateli interakci s komponentou. Sada nástrojů Windows Forms umožňuje procházení souborů, správu databází či přístup na webové stránky. Každá komponenta má seznam vlastností, které si může vývojář modifikovat, jako například můžeme upravit vlastnosti písma, umístění atd.

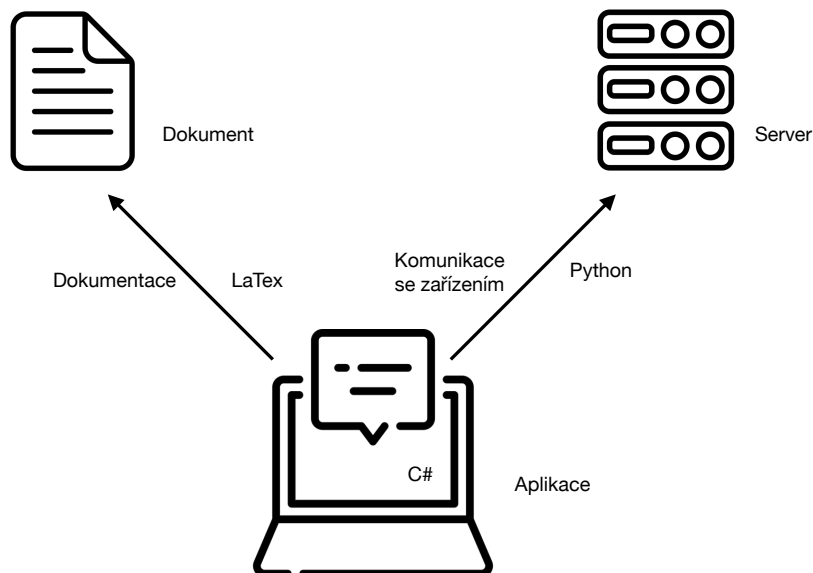
Vytvořením nového projektu Windows Forms vznikne přednastavený formulář.[22] Spolu s formulářem se vývojáři zobrazí okno Windows Forms Designeru, které slouží k umístování (drag-and-drop) a přidávání ze seznamu ovládacích prvků do daného vytvořeného formuláře.

Vytvořením každého formuláře se vytvoří taktéž dva soubory:

- `Jméno.Designer.cs` slouží k inicializaci použitých komponent a k definici jeho vlastností pomocí tečkové notace. Všechny prvky jsou nastaveny na `private`, tedy je může využívat pouze z formuláře, kde jsou vytvořeny.
- `Jméno.cs` definuje interakci prvků formuláře.

6.2 Programovací framework a jazyky

Obsahem této části práce bude popis a odůvodnění výběru programovacích jazyků C#, Python a programovací frameworku .NET pro realizaci aplikace. Externí nástroje aplikace jsou zobrazeny na obrázku 6.1.



Obrázek 6.1. Externí nástroje aplikace

6.2.1 C

Jazyk C# vznikl v roce 1999 společností Microsoft. Je objektově orientovaným, open source a moderním jazykem pro psaní aplikací běžících ve frameworku .NET. Součástí jazyka je řada komponent. Jak již v názvu může být patrné, má své základy v jazyce C.

Výhodami C# jazyka jsou:

- Díky kompilaci do “mezikódu” CLI (Common Intermediate Language) odhalíme dříve chyby.
- Široký výběr z knihoven, šablon a datových struktur pro psaní aplikací.
- Jazyk je srozumitelný - při psaní je využíváno metod a tříd.
- Uvolňuje paměť objektů - pokud se objekt nevyužívá či jsou nedostupné.
- Možnost zpracování výjimek.
- Návratová hodnota metod může být `null`.

■ 6.2.2 Python

Python je interpretovaný, objektově orientovaný a vysokoúrovňový programovací jazyk, který se těší velké oblibě a vývoji. Je používán například pro vývoj umělé inteligence, testingu, machine learningu a v mnoha dalších oblastech. Syntaxe jazyku Python je lehce pochopitelná a čitelná. Jazyk byl vybrán kvůli kompatibilitě s vygenerovanými soubory z kompilátoru ASN1SCC, které dále zpracovává a využívá ke komunikaci se zařízením.

■ 6.2.3 NET framework

.NET je open source a cross-platform framework pro vývoj aplikací, který se začal vyvíjet v roce 1990 pod jménem Next Generation Windows Services (NGWS). První verze .NET 1.0 byla uvedena v roce 2001 a první verze .NET framework v roce 2002. Aplikace, které jsou pomocí .NET psány, vznikají v programovacím jazyku C#, který byl vysvětlen v kapitole 6.2.1, F# a Visual Basic. Kromě aplikací se používá také pro vývoj webů, her či IoT. Pro spouštění kódů je využíváno Common Language Runtime (CLR), což je součástí jazykové infrastruktury CLI.

■ 6.3 Generování dokumentace

Dokumentace by měla být uživateli vygenerována v takovém formátu, aby si ji mohl zobrazit bez omezení na operačním systému počítače, kde bude aplikace používána. Tento cíl splňuje softwarový nástroj pro sazbu dokumentů LaTeX.

■ 6.3.1 Nástroj LaTeX

LaTeX je systémovým nástrojem pro tvorbu dokumentů, který byl vytvořen Leslie Lamportem na počátku 80. let. LaTeX umožňuje uživateli sepsat požadavky na formátování a rozvržení dokumentu. Pro celou přípravu dokumentu uživatel používá pouze psaných příkazů či značek ve formě textu. Po sestavení výstupního souboru je uživateli vytvořen soubor PDF.

Výhodami nástroje a důvod proč byl vybrán je kompatibilita dokumentu napříč operačními systémy. Uživatel není omezen, jestli je soubor typu Word, LibreOffice či Pages. Dále může být zmíněno ještě upravování formátování a struktury po definování napříč celým dokumentem a lehká tvorba tabulek. Tyto výhody byly brány v potaz hlavně z důvodu automatického generování dokumentace, aby byla snáze generovaná a lehce spustitelná.

■ 6.4 Nástroj Taste tools od ESA

Vývojové prostředí The ASSERT Set of Tools for Engineering (TASTE) vznikl na základě iniciativy ESA v roce 2008 a vychází z programu FP6. Nástroj je využíván pro návrh malých a středně velkých vestavěných systémů v reálném čase.[23] Zaměřuje se na automatizaci prací, při kterých snadno vznikají chyby znesnadňující integraci kódů do systému. Při návrhu vychází z technologií AADL, ASN.1 a SDL. Výhodou prostředí je snadné integrování částí kódů, které vznikaly ručně nebo pomocí nástrojů MATLAB Simulink nebo například SCADE.

Když jsou nástroje předány pečlivě vybrané vlastnosti, začne TASTE pracovat na vytváření binárních souborů, se kterými může vývojář dále pracovat pomocí programů v jazyce Python.

6.4.1 Funkce nástroje ESA-TASTE

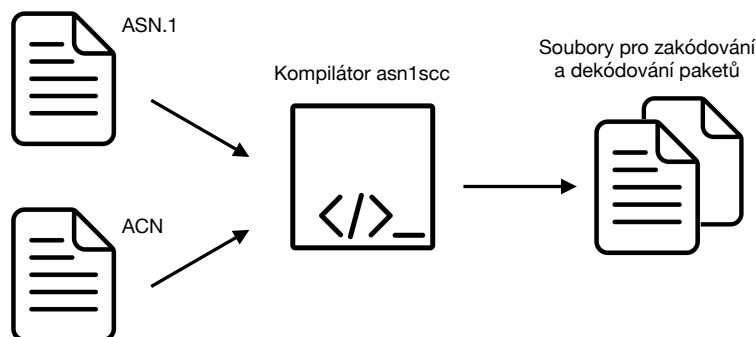
Jak již bylo zmíněno, TASTE se používá ve vestavěných (embedded) systémech, které potřebují pro vývoj softwaru odborné znalosti vývojářů. Je to především z těchto důvodů:

- Vestavěné systémy obsahují aplikace z různých oblastí a jsou vyvíjeny různými vývojáři.
- Mají heterogenní HW.
- Mají autonomní charakter a pracují s rozmanitými systémy.

6.5 Kompilátor ASN1SCC

Nástroj je open-source kompilátor pro vestavěné platformy kritické z hlediska bezpečnosti, který byl vyvinut společností Semantix, dnes Neuropublic a je plně financován Evropskou kosmickou agenturou. Byl naprogramován v jazyce F#.

Pro každý vstupní soubor ASN.1 je načten i doprovodný soubor ACN, na jejichž základě je kompilátor schopný vygenerovat soubory pro zakódování a dekodování paketů. Kompilátor využívá malou paměť a náročnost na CPU. Ač nástroj spadá pod licenci GPL, všechny kódy a soubory, které s jeho použitím vzniknou a operace během runtime nástroje, nepodléhají licenčním pravidlům.[24] Diagram použití kompilátoru je na obrázku 6.2.



Obrázek 6.2. Diagram použití kompilátoru ASN1SCC

Kapitola 7

Návrh aplikace

Hlavním cílem této práce je navrhnout aplikaci v programovacím jazyce C#, který by měl sloužit k specifikaci komunikačních protokolů TC a TM, které umožňují komunikovat s přístroji na palubě družice a také jejich odpověď přijímat. Důraz je kladen na specifikaci protokolů, které by měly být v souladu se standardem ECSS-E-ST-70-41C. Aplikace slouží k eliminaci chyb syntaxe, které se dnes běžně dějí při ručním zápisu protokolů. Také bude mít možnost načíst existující soubory a upravovat či přidávat další služby protokolu.

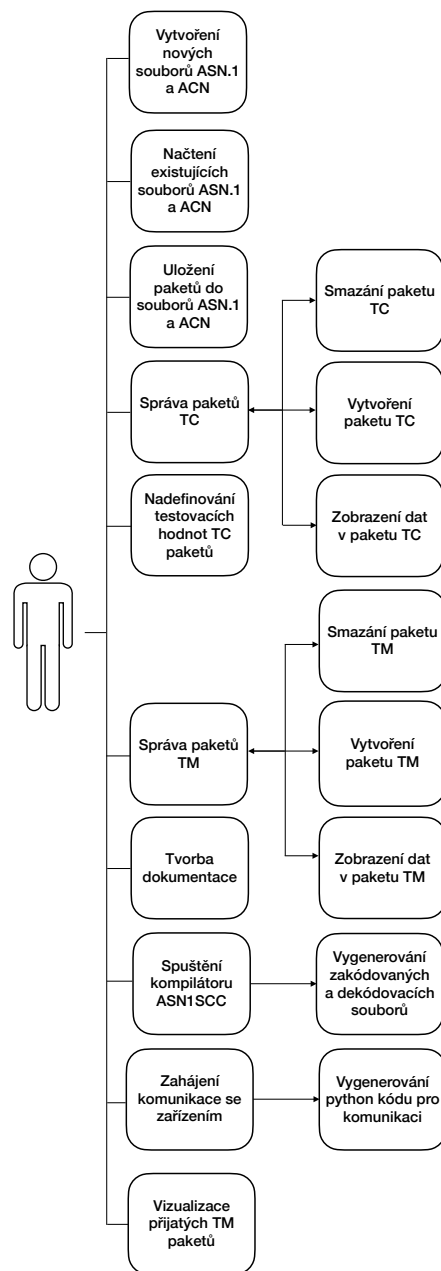
7.1 Architektura aplikace

Aplikace je navrhována tak, aby její ovládání bylo pro uživatele jednoduché a přehledné. Je dbáno na její funkční požadavky, ale i na případná budoucí vylepšení aplikace.

7.1.1 Požadavky na funkce aplikace

Požadavky, které musí aplikace splňovat a pro která je nutno navrhnout rozvržení aplikace, jsou zobrazeny na obrázku 7.1.

- Aplikace by měla umožňovat specifikovat služby typu 1, 3, 5, 6, 9 a 17 protokolů TM a TC.
- Specifikované služby TC a TM by měla aplikace vizualizovat.
- Aplikace by měla specifikované služby uložit do komunikačního protokolu ASN.1 a ACN.
- Komunikační protokoly formátu ASN.1/ACN by mělo být možné do aplikace načíst a dále upravovat.
- K navrženým komunikačním protokolům z aplikace vygenerovat jejich dokumentaci.
- Aplikace by měla mít funkci spustit program kompilátoru ASN1SCC nástroje ESA-TASTE.
- Aplikace by měla vygenerovat a spustit zdrojový kód pro odeslání TC a přijetí TM v programovacím jazyce Python.
- Uživateli by mělo být umožněno nadefinovat vstupní testovací hodnoty TC.
- V aplikaci by měla být vizualizace odeslaných a přijatých dat.



Obrázek 7.1. Funkcionality aplikace

Po spuštění aplikace

Po spuštění aplikace by se uživateli měl zobrazit úvodní a výchozí formulář, který by obsahoval popis použití aplikace. Na tomto formuláři bude uživateli zobrazeno logo a název aplikace společně s jménem jejího autora. Hlavní formulář má v horní části zobrazeno menu aplikace.

Návrh menu (GUI)

Pro vytvoření řídicího elementu aplikace, tedy menu, bylo rozhodnuto kvůli přehlednosti všech hlavních činností, které aplikace umožňuje. Pro menu aplikace bude vybráno rozbalovací menu v horní části hlavního formuláře, tak aby bylo uživateli dostupné a viditelné. Hlavní nabídky menu jsou uživateli zobrazeny pro celou dobu použití aplikace.

Taktéž bude možno použít známé klávesové zkratky na klávesnici. Jedná se o zkratky pro tvorbu nových souborů formátů ASN.1 a ACN, které vytvoříme použitím kombinace kláves Ctrl+N, otevření nebo též načtení obsahu souboru ASN.1 a ACN pomocí kombinace kláves Ctrl+O a také ukládání těchto souborů kombinací kláves Ctrl+S.

Některé prvky v menu budou mít funkci rozbalovací nabídky a ty budou obsahovat možnosti pro správu souborů, protokolů, tvorbu dokumentace a propojení s nástrojem kompilátoru ASN1SCC.

Návrh ovládání aplikace

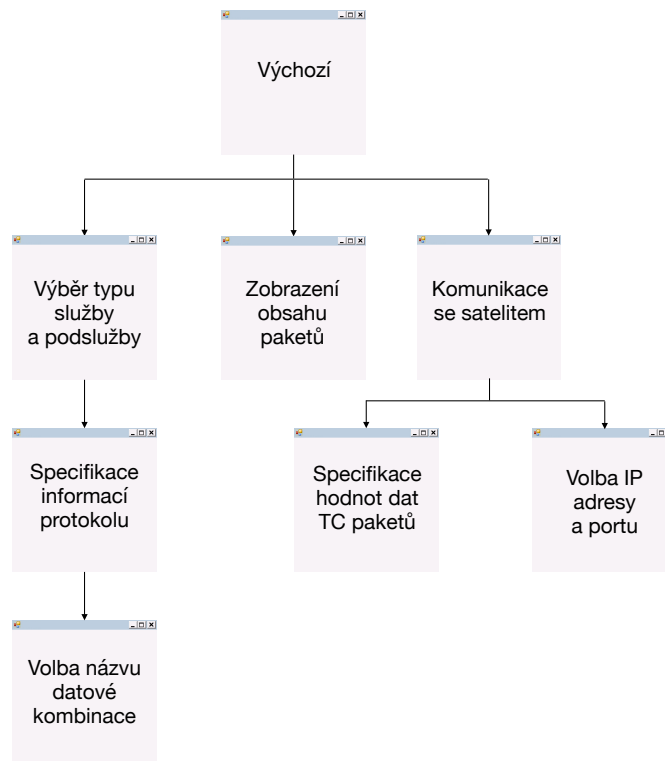
Aplikace se bude ovládat pomocí komponenty tlačítka. Tlačítko bude sloužit pro přechod mezi formuláři činností, které jsou uvedené v menu a pro ovládání prvků např. smazání služby ze seznamu již nadefinovaných služeb.

7.1.2 Rozvržení aplikace

Návrh mapy aplikace je nedílným procesem na počátku jejího vývoje. Je zapotřebí rozvrhnout strukturu a možné interakce mezi formuláři. Pokud při vývoji nastane chyba v interakci komponenty, lze ji v mapě aplikace snáze nalézt a odstranit. Rozvržení aplikace je na obrázku 7.2.

Typy formulářů, které aplikace bude dle návrhu obsahovat

- Hlavní formulář - úvodní formulář aplikace, který se uživateli zobrazí s účelem a funkcemi, které aplikace nabízí. Uživatel má možnost volit činnosti aplikace z menu. Z hlavního formuláře uživatel vytvoří nový soubor, načte existující či vytvoří protokol TC/TM.
- Formulář výběru typu služby a podslužby - formulář slouží pouze jako spojovník hlavního formuláře s formulářem s definicí informací, které chce uživatel do protokolu TC/TM uložit.
- Formulář specifikace informací protokolu - v tomto formuláři uživatel zvolí rozsah a obsah dat, které chce přenést protokolem. Po uložení formuláře je uživatel nasměrován na hlavní formulář aplikace.
- Formulář volby názvu datové kombinace - pokud jsou data zakódována kombinací, která nebyla ještě k zakódování dat použita, uživatel je vyzván k zadání jejich názvu.
- Formulář zobrazení obsahu komunikačního protokolu - tento formulář zobrazí uživateli veškeré nadefinované informace o komunikačním paketu.
- Formulář komunikace se satelitem - uživatel ve formuláři bude moci zvolit konkrétní TC zprávy, které se budou satelitu zasílat, bude zde vizualizace TC a TM zpráv.
- Formulář specifikace hodnot dat TC paketů - tento formulář slouží k nadefinování vstupních testovacích hodnot TC protokolů.
- Formulář volby portu a IP adresy - aby byla komunikace se zařízením započata, uživatel si ve formuláři navolí potřebné informace síťové komunikace.



Obrázek 7.2. Diagram propojení formulářů aplikace

Komponenty použité v aplikaci

- Panel
- Label - nadpis
- Button - tlačítko
- ListBox - ovládací prvek zobrazující seznam položek
- TextBox - ovládací prvek umožňující uživateli zadávat text
- ComboBox - ovládací prvek umožňující uživateli výběr položek ze seznamu hodnot
- DataGridView - ovládací prvek používající tabulku sloužící k zobrazení dat

Návrh nezbytných funkcí formulářů:

Po vydefinování požadavků, které by měla aplikace zahrnout, bude dále rozebrán návrh funkcí a jejich zakomponování do formulářů.

1. Správa protokolů TC a TM

Po vybrání položky menu Protocols se uživateli zobrazí nabídka TC a TM. Uživatel stiskne nabídku TC, zobrazí se stránka pro správu TC protokolů, uživatel tedy zůstane stále ve výchozím formuláři. Stránka formuláře bude obsahovat dva seznamy, kde uživatel uvidí jeden seznam pro definované TC protokoly a druhý seznam pro protokoly TC, které bude chtít uživatel převést do jazyka ASN.1 a ACN. Pro vybrání TC protokolů z prvního pole do druhého bude mít uživatel tlačítko.

Přidání, smazání, zobrazení obsahu paketů a definování vstupních testovacích hodnot bude řešeno pomocí tlačítka, kdy pro případ přidání, definování vstupních testovacích hodnot a zobrazení obsahu budou tlačítka delegovat uživatele na formuláře pro to určené.

Stránka pro správu TM protokolů bude mít stejný vzhled, nebude však mít tlačítko pro definování vstupních testovacích hodnot.

2. Přidání nového TC a TM paketu

Pro přidání paketu využije uživatel tlačítko pro přidání, které ho deleguje na nový formulář. Ve formuláři budou textová pole pro vyplnění informací o typu služby, podslužby a jejich názvu. Po vyplnění veškerých polí uživatel stiskne tlačítko. Tento formulář bude stejný pro pakety TC i TM.

3. Nadefinování informací paketu protokolu TC a TM

Informace o paketu se musí definovat u každého protokolu. Pro větší přehlednost bude pro tuto činnost vytvořen nový formulář. Pole formuláře by měly obsáhnout vlastnosti jazyka ASN.1 a ACN, které může uživatel v aplikaci využít. Po vyplnění informací jednomu datu paketu uživatel tyto informace uloží stisknutím tlačítka. Pokud uživatel zadal kombinaci informací, která ještě nebyla použita, je vyzván novým formulářem pro zadání jejího názvu.

Názvy dat, které si pro daný paket uživatel nadefinoval, budou zobrazena v seznamu v pravé části formuláře. Po nadefinování všech dat, která chce mít uživatel v paketu obsažena, potvrdí jejich uložení stisknutím tlačítka **Save**. Tento formulář bude shodný pro protokoly TC i TM.

4. Volba názvu kombinace definovaných informací dat

Uživatel bude vyzván pomocí nového formuláře, který uživatele informuje o svém účelu. Pro jeho odeslání bude nutné vyplnit textové pole názvu a stisknout tlačítko **Save**. Tento formulář bude shodný pro protokoly TC i TM.

5. Zobrazení obsahu paketu

Formulář bude sloužit k zobrazení obsahu paketu, výčtu obsažených dat a jejich kódování. Pro zobrazení využijeme tabulku, kde budou jednotlivá data vkládána do sloupců.

6. Definování vstupních testovacích hodnot

Výběr paketů, které budeme odesílat v rámci komunikace satelitu, zahájení komunikace a zobrazení obsahu přijatých paketů bude uživatel obsluhovat v rámci jednoho formuláře.

V levé části formuláře bude seznam s protokoly TC, jejichž strukturu si uživatel nadefinoval. Pro zobrazení nadefinovaných hodnot paketů a přijatých paketů bude využito dvou tabulek pro přehledně umístěných nad sebou. Komunikace se zařízením bude zahájena stisknutím tlačítka. Definování vstupních testovacích hodnot bude probíhat v rámci jiného formuláře.

Formulář pro nadefinování testovacích hodnot bude obsahovat pouze pole se seznamem nabídky názvů jednotlivých dat v paketu a textové pole pro zadání hodnoty uživatelem.

7. Zakódování a dekodování zpráv

Soubory pro zakódování a dekodování paketů budou pocházet od kompilátoru ASN1SCC, které budou v jazyce Python. Aplikace bude generovat kód pro zakódování a dekodování, dle zadaných testovacích hodnot TC paketů, kde využije právě soubory z kompilátoru.

8. Komunikace se zařízením

Aby byl uživatel propojen se zařízením, bude nutno zadat doplňující údaje, které umožní propojení se zařízením. Své údaje potvrdí stisknutím tlačítka. Údaje, které bude uživatel muset zadat, budou záviset na výběru přístupu k propojení a komunikaci se zařízením. Máme na výběr ze dvou nejvíce používaných přístupů a to ze sériové a síťové komunikace.

Popis nejvíce používaných přístupů ke komunikaci se zařízením:

■ Sériová komunikace

Sériovou komunikací je označována komunikace mezi zařízeními, která umožňuje přenos informací. Jedná se o nejrozšířenější přístup komunikace zařízení, která má jasně daná pravidla mezi zařízením odesílatele a příjemce. Informace, které jsou sériovou komunikací přenášeny, jsou závislé na změně elektrického napětí nebo proudu, tedy jsou v binární formě 1 (představuje 5V) a 0 (představuje 0V). Informace zasílá bit po bitu pomocí komunikačního kanálu, který tvoří přijímače (receiver) a vysílače (transmitter).

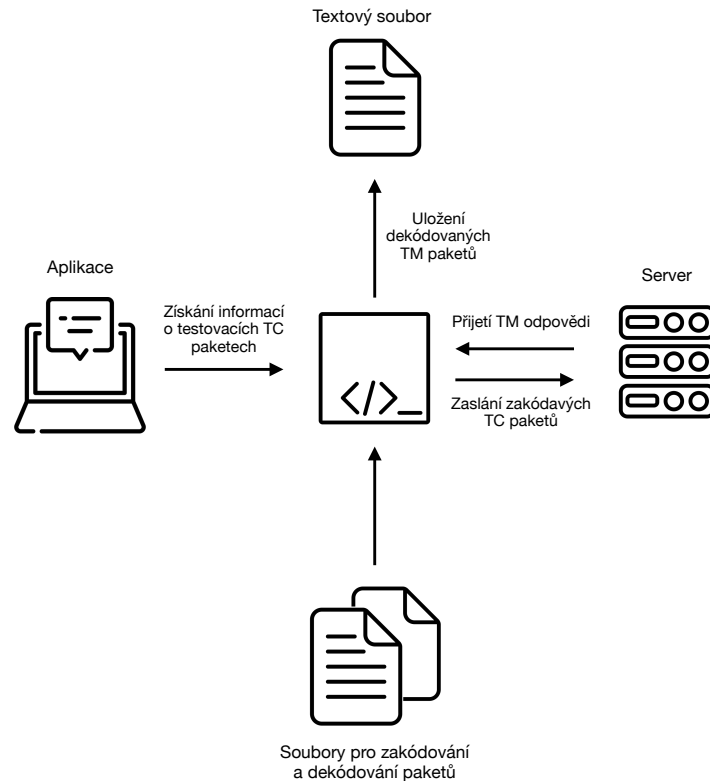
Komunikace je velmi ovlivněna parametry datového kabelu, který umožňuje propojení mezi zařízeními a vzdáleností mezi zařízeními. Uživatel by pro vytvoření spojení zadával port a přenosovou rychlost (baud rate).

■ Síťová komunikace - Transmission Control Protocol (TCP) komunikace

Transmission Control Protocol (TCP) je standardem a protokolem, který je používán pro síťovou komunikaci mezi zařízeními. Spojení, které je díky TCP, je navázáno, je udržováno než skončí výměna informací mezi klientem a serverem. Pokud nastane chyba během přenosu informací, TCP na ní upozorní. Díky vlastnostem TCP pro přesný a spolehlivý přenos informací a umožnění testování aplikace bez

nutnosti přítomnosti u zařízení, byl protokol TCP vybrán pro testování aplikace. Pro klienta je nutno znát IP adresu serveru a příslušný port.

Jak již bylo zmíněno výše, pro komunikaci se zařízením je aplikací automaticky generován kód v jazyce Python, který načítá soubory pro zakódování a dekodování paketů, které byly vygenerované kompilátorem ASN1SCC a jsou rovněž v jazyce Python. Program je rozdělen do více metod, externí části programu jsou na obrázku 7.3.



Obrázek 7.3. Externí části programu

Z aplikace budou načteny informace o TC paketech a testovacích hodnotách, které uživatel chce zařízení zaslat. Díky kódu pro zakódování a dekodování paketu z kompilátoru ASN1SCC je aplikace schopna zpracovat a automaticky generovat příkazy pro definici paketu. Metody pro jeho zakódování a dekodování, které ještě doplní aplikace o výpočet Cyklického redundantního součtu (CRC), který kontroluje, zda jsou data v paketu kompletní a nedošlo při zasílání satelitu k nějaké chybě přenosu.

Na základě výběru TCP komunikace dále python program definuje navázání spojení s TCP serverem, zaslání dotazu - paketu, získání odpovědi a ukončení spojení.

9. Dokumentace

Aplikace bude generovat automaticky dokumentaci obsahující informace o všech TC a TM paketech, které budou v aplikaci nadefinovány. Uživatel bude mít informace o obsahu paketu, jeho kódování a pakety budou přehledně rozděleny podle zařazení do TC a TM paketů.

7.2 Název a ikona aplikace

Ikona slouží ke zvýšení zapamatování aplikace uživatelem. Logo bylo vytvořeno v Adobe Creatice Cloud Expresss. Hlavním cílem byla jeho jednoduchost a také aby vystihovalo smysl aplikace. Pro aplikaci byla navržena dvě loga, logo pro ikonu aplikace a logo zobrazené na horní liště Windows Forms. K návrhu názvu bylo přistupováno stejným způsobem a bylo do loga taktéž zakomponováno. Název aplikace je Spacee talks. Výsledná loga jsou zobrazena na obrázcích 7.4.



Obrázek 7.4. a) Logo ikony aplikace b) Logo zobrazené v horní liště aplikace

7.3 Tvorba uživatelského rozhraní (UI)

Hlavním kritériem při tvorbě UI aplikace je navrhnout UX i UI, aby byla jejich architektura pro uživatele pokud možno srozumitelná, intuitivní a snadno použitelná.

7.3.1 Zásady zahrnuté v UI aplikace

- **Známost** - používat prvky, se kterými se uživatel ve svém životě při používání aplikací setkal.
- **Jasnost** - pomocí chybových textových dialogů dovysvětlit uživateli význam prvků a jejich předpokládaných vstupů.
- **Konzistence** - používat stejné vlastnosti komponent a tím docílit jednotného vizuálního vjemu z aplikace.
- **Efektivita** - ušetřit časovou náročnost ručního zápisu služeb do formátu ASN.1/ACN.
- **Estetika** - nehledět pouze na funkčnost aplikace, ale i zpříjemnit používání aplikace uživateli.

7.3.2 Výběr barev aplikace

Návrh barev, které bude aplikace využívat, je nedílnou součástí při tvorbě UI. U uživatele spoluvytváří dobrý dojem při používání aplikace. Pro aplikaci volím maximálně čtyři barvy - jednu dominantní, která by měla být oproti ostatním výrazná. Při volbě bude dbáno na to, aby zbylé barvy byly kontrastní. Barvy, které budou pro aplikaci zvoleny, jsou zobrazeny na obrázku 7.5.

Rozdělení barev

- Panel Menu - barva panelu menu bude zvolena dominantní, shodná s tou, která je použita v logu aplikace, s hexadecimálním kódem #4C6173.
- Pozadí - pro pozadí aplikace jsem vybrala barvu z nabídky SystemColors z nabídky Windows Forms s názvem Info s hexadecimálním kódem #FFFFFFE1.
- Komponenty - barva #4C6173 je v aplikaci hlavně využívána pro tlačítko.
- Text Panel Menu - volím barvu, která je použita v logu. Barva má kód #D9D1BA.
- Text - barva textu aplikace je kontrastní k světlému pozadí. Použiji stejnou barvu s označením #D9D1BA, jako je vybrána pro text panel menu. Sjednotí se tím vjem z aplikace.



Obrázek 7.5. a) Barva #4C6173 b) Barva #FFFFFFE1 c) Barva #D9D1BA

Kapitola 8

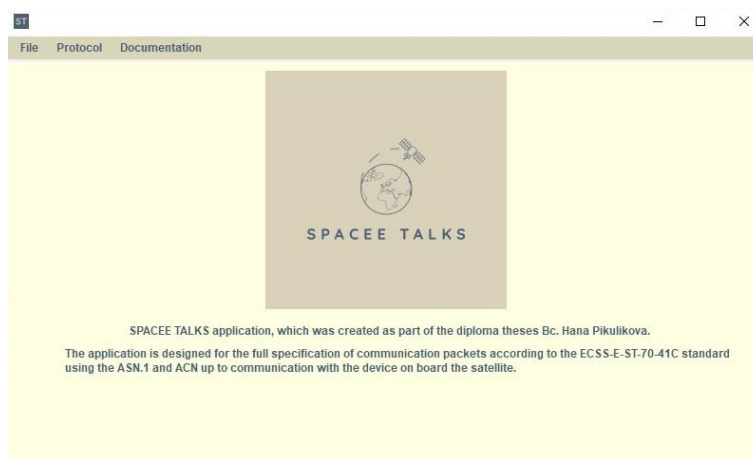
Implementace aplikace

V kapitole bude popsána implementace aplikace. Popis bude vycházet z návrhu rozložení aplikace. V závěru práce bude provedeno otestování aplikace a bude zhodnocena její funkčnost a nedostatky, které byly při implementaci řešeny. Dle zjištěných nedostatků bude navrženo její budoucí vylepšení.

8.1 Hlavní formulář

Po otevření aplikace se uživateli zobrazí hlavní formulář s MenuStrip menu v horní části formuláře. Hlavní formulář slouží jako rozcestníkový formulář aplikace. Formulář umožňuje využít všech funkcí a procesů aplikace, je tedy vždy otevřen.

Menu obsahuje kontextové nabídky **File**, **Protocols** a **Documentation**. Záložka menu **File** obsahuje nabídku **New**, **Open** a **Save**. Záložka **Protocols** zobrazuje nabídku **TM** a **TC**. Poslední záložka **Documentation** obsahuje nabídku **Create**. Formulář aplikace při spuštění představí uživateli aplikaci a její funkce. Hlavní formulář je zobrazen na obrázku 8.1.



Obrázek 8.1. Hlavní formulář

8.1.1 Načtení existujících souborů ASN.1 a ACN

Struktura souborů v jazycích ASN.1 a ACN je stanovena, proto bylo cílem rozparsovat soubory pomocí rozpoznání slov/entit, která jsou v jazycích definována například simple types či structured types. Celé možnosti, které mohou nastat, byly rozepsány v kapitole 5. K ukládání dat bylo využito slovníku těchto entit.

Na spuštění metody, která soubory načte, se uživatel dostane pomocí kliknutí na tlačítko **Load** pod kontextovou nabídkou **File** panel menu v horní části formuláře. K načtení dat ze souborů ASN.1 a ACN je využito control prvku **OpenFileDialog**, který je schopný otevřít okno průzkumníka souborů uživatele. Na rozdíl od ostatních nástrojů tvorby aplikace se nemusí u dialogu určovat jejich vizuální vlastnosti.

Pro otevření souborů jen nutného typu ASN.1 a ACN bylo využito vlastnosti dialogu Filter, kde jsme definovali přípustné typy souborů:

```
openFileDialog.Filter = "asn files |*.asn|acn files |*.acn";
```

Jelikož bude čtení dat probíhat z obou souborů ASN.1 i ACN, postačí získání cesty z dialogu okna jen k jednomu souboru, u které je odstraněna přípona souboru a může aplikace tedy spárovat název dokumentu jak s typem souboru ASN.1 a ACN.

K cestě je poté připojena přípona souboru, ze kterého bude aplikace data číst. Ze souboru jsou načteny všechny řádky do listu, který je následně převeden do pole. Z pole jsou odstraněny prázdné prvky, které v souboru představovaly prázdné řádky.

Po načtení souborů ASN.1 a ACN do slovníku aplikace zkontroluje, zda již není použita datová kombinace či definovaný paket. V případě, že je použit, je zamezeno uložení dat ze souborů do struktury aplikace a uživatel je na tuto skutečnost upozorněn MessageBoxem s hláškou: “An error occurred while loading the .asn or .acn file. Some definition of packets or data definitions is already defined in application. Edit the file and try again.”

Pokud proběhlo bez chyb, jsou pakety rozříděny do seznamů (ListBox) definovaných TC a TM paketů, které nalezne uživatel pod kontextovou nabídkou menu **Protocols**. V případě, kdy aplikace při načítání dokumentu narazí na chybu syntaxe, uživatel bude upozorněn pomocí MessageBoxu s hláškou: “An error occurred while loading the .asn or .acn file. Check that the file has the correct syntax and try again.”

8.1.2 Vytvoření souborů ASN.1 a ACN

Na spuštění metody, která soubory vytvoří, se uživatel dostane pomocí stisknutí tlačítka **New** pod kontextovou nabídkou **File** panel menu v horní části formuláře. K vytvoření nových souborů využívá aplikace control prvku **SaveFileDialog**, který otevře okno průzkumníka souborů uživatele. Pro vytvoření souborů pouze námi nutného typu ASN.1 a ACN bylo využito vlastnosti dialogu Filter, kde byly předem předefinovány přípony souborů, které může uživatel vytvořit.

Uživatel pojmenuje pouze jeden soubor s příponou “.asn” či “.acn”. Pomocí metody **Split()** získáme cestu k souboru bez přípony, která bude poté doplněna a bude tedy umožněno vytvořit jak soubor .asn tak i .acn. Jestliže při vytváření dokumentů nastane chyba, uživatel na ni bude upozorněn MessageBoxem s hláškou: “An error occurred while creating the .asn or .acn file. Try again.”

8.1.3 Zápis do souborů ASN.1 a ACN

Na spuštění metody, která soubory načte, se uživatel dostane pomocí stisknutí tlačítka **Save** pod kontextovou nabídkou **File** panel menu v horní části formuláře. Jelikož jsou data pro zápis do souborů ASN.1 a ACN již předpřipravená ve slovnících nedefinovaných dat a použitých datových kombinací, zápis bude probíhat dle pravidel známé struktury souborů.

K uložení dat do souborů aplikace využívá control prvek **SaveFileDialog**, který otevře okno průzkumníka souborů uživatele. Pro otevření souborů pouze námi nutného typu ASN.1 a ACN bylo využito vlastnosti dialogu Filter.

Uživatel zvolí jen jeden soubor pomocí průzkumníka nástrojů, ze kterého se získá cesta a název souboru, tedy bez přípony typu souboru. Z tohoto oddělení je získána cesta, která bude shodná pro oba soubory ASN.1 a ACN a ke kterým bude doplněna přípona obou souborů podle toho, do kterého budeme právě zapisovat.

Pro zápis bylo použito třídy `StreamWriter`, kterou bylo nutné nejprve deklarovat pomocí kódu:

```
StreamWriter swASN = new StreamWriter(filePathASN);
```

Pro zápis obsahu souborů byla použita metoda `StreamWriter` `swASN.WriteLine()`, tedy zápis po řádcích. Do souborů si uživatel může vybrat protokoly, které jsou definovány v aplikaci pomocí odděleného seznamu `To transfer`. Zápis využívá pravidel syntaxe ASN.1 a ACN, která je popsána v kapitole 5. Po ukončení zápisu do souboru byl zastaven `StreamWriter` metodou `Close()`. Pokud při vytváření dokumentů vznikne chyba, je na ni uživatel upozorněn `MessageBoxem` s hláškou: “An error occurred while creating the .asn or .acn file. Try again.”

8.1.4 Panel správy paketů TC a TM

Na formulář pro výběr typu služby a podslužby se dostaneme pomocí kliknutí na položku kontextové nabídky `Protocols` - `TM` či `TC` panel menu v horní části formuláře.

Uživateli se zobrazí control prvek panel pro definování příslušných protokolů. Panel seskupuje jiné ovládací komponenty jako například nadpisy (labels), tlačítka či seznamy. Panel se zobrazí díky jeho vlastnosti `Visible=true`, ostatní panely jsou schované. Funkce tlačítka budou popsány dle jejich funkcí.

Panel protokolu `TC` se skládá ze stejných komponent a jejich funkce je identická jako pro protokol `TM` kromě tlačítka `Test values` pro definování testovacích hodnot. Panel správy paketů `TC` je zobrazen na obrázku 8.2 a panel správy paketů `TM` je zobrazen na obrázku 8.3. Pro interakci s tlačítky byla použita událost `click event handler`.

Button Add

Tlačítko vytvoří a zobrazí nový formulář výběru typu služby a podslužby. Podrobněji bude formulář výběru typu služby a podslužby popsán níže.

Pro formulář výběru typu služby a podslužby je předdefinován pouze obsah pole se seznamem nabídky typu služeb z hlavního formuláře, který se doplní obsahem podle typu paketu.

Button Delete

Tlačítko odstraní právě vybranou položku v seznamu nadefinovaných protokolů. Zároveň ji odstraní i ze slovníku definovaných paketů.

Button Show

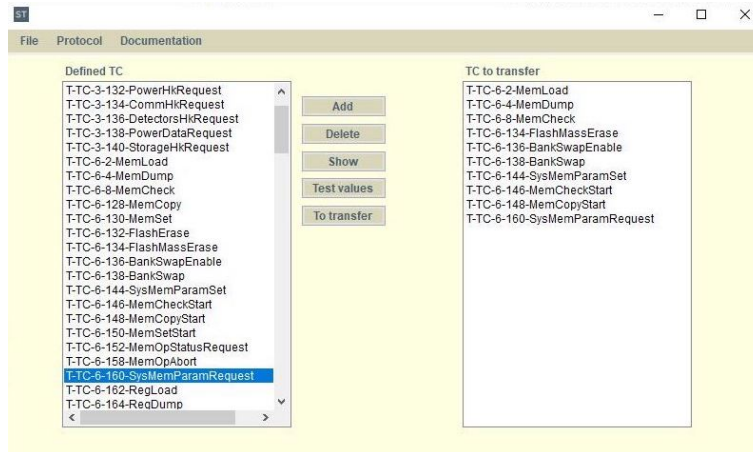
Tlačítko otevře formulář zobrazení obsahu paketů, který zobrazí obsah paketu i s jeho kódováním. Informaci o paketu, jehož obsah má zobrazit, přebírá formulář z vybrané položky v seznamu. Podrobněji bude formulář popsán dále.

Button Test values

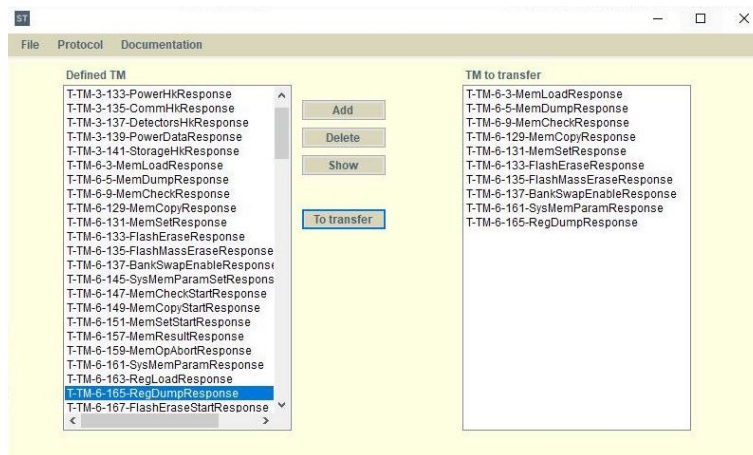
Tlačítko otevře formulář pro přehled `TC` komunikačních paketů, které může uživatel zaslat satelitu a tedy určit pro ně testovací hodnoty a navázat komunikaci se zařízením. Formulář zobrazuje i přehled přijatých odpovědí od satelitu.

Button To transfer

Tlačítko zjistí, zda je zvolená položka již zobrazena v seznamu protokolů pro zápis do souborů `ASN.1` a `ACN`. Pokud není položka zobrazena, přidá se protokol v seznamu pro zápis. V seznamu se zobrazí název protokolu pomocí jeho názvu, který se získává ze slovníku definovaných paketů v aplikaci.



Obrázek 8.2. Panel správy paketů TC



Obrázek 8.3. Panel správy paketů TM

8.2 Formulář výběru typu služby a podslužby

Na formulář pro výběr typu služby a podslužby se dostane uživatel pomocí stisknutí tlačítka Add z panelu pod kontextovou nabídkou **Protocols** - TM či TC z panel menu v horní části formuláře.

Obsah pole se seznamem nabídky (comboBoxu) pro výběr typu služby je definován, jak je popsáno výše, již v hlavním formuláři. Aby byl přehled možných podslužeb pro definování uživateli co možná nejpřehlednější a nevznikly zbytečné chyby při výběru podslužby, který pro vybranou službu neexistuje, je pole se seznamem nabídky podslužby definované až na základě vybrané služby. Při definování nabídky výběru je vycházeno ze standardu ECSS-E-ST-70-41C. Jejich přehled je v kapitole 4.2.1. Formulář výběru typu služby a podslužby je zobrazen na obrázku 8.4.

Pokud by uživatel chtěl definovat vlastní specifický typ služby, může použít nabídku ze seznamu pole služby “Other”. Uživateli se zviditelní textové pole (textBox) pro zadání vlastního čísla služby a podslužby s příslušnými nadpisy. Pole se seznamem nabídky podslužby se sama předvyplní na jedinou možnost, která při definování vlastní specifické služby nastane, na možnost “Other”.

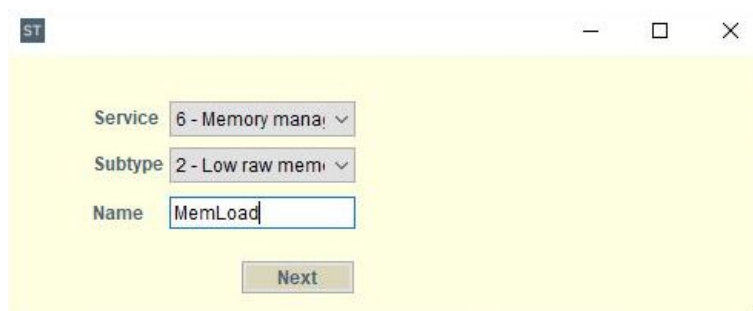
Při nastavení vlastností pro pole se seznamem nabídky bylo znemožněno vpisování informací a jen uživatel může vybírat z kolekce jejich možností nastavením. Tato úprava byla definována ve stylu pole se seznamem nabídky výběrem DropDownStyle.

Stisknutím tlačítka **Next** se zobrazí uživateli formulář specifikace informací protokolu na základě vybraného typu služby a podslužby. Informace o zvoleném výběru z pole se seznamem nabídky zjistíme přes jeho vlastnost `Text`. Z vyplněných možností formuláře se vytváří název protokolu.

Jestliže však uživatel zvolil definování vlastní specifické služby, nestačí přistupovat k vybraným možnostem přes vlastnost `Text`. Musí se vycházet z obsahu textových polí služby a podslužby pomocí vlastnosti `Text`. Pokud je název paketu již v aplikaci použit, je na to uživatel upozorněn `MessageBoxem` s hláškou: “The packet name matches a packet that is already defined.” Textové pole názvu paketu se vymaže a uživatel musí zadat nový název. Pokud se název neshoduje, zobrazí se nový formulář pro specifikaci informací protokolu. Pro zobrazení musíme nejprve formulář inicializovat a zobrazit ho jako modální dialogové okno pomocí kódu:

```
FormOther formOther = new FormOther();
formOther.ShowDialog();
```

Aby nebyl uživatel přehlčen formuláři aplikace při pŕchodu zadávání informací k novým protokolŕm, bude se po vyplnění a uložení informací formulář zavírat. Aplikace si zkontroluje, zda je příslušný formulář otevřen a pomocí vlastnosti `Hide()` formulář uzavře.



Obrázek 8.4. Formulář výběru typu služby a podslužby

8.3 Formulář specifikace informací protokolu

Na formulář pro zadání bližších informací protokolu se dostaneme pomocí stisknutí tlačítka **Next** z formuláře výběru typu služby a podslužby.

Ve formuláři bude mít uživatel jako jedinou předem danou informaci název protokolu, který si v předešlém formuláři vybral.

Value of simple type se liší dle druhu zvoleného Simple typu. Pokud uživatel zvolí číselný datový typ `INTEGER` či `REAL`, zviditelní se textová pole s příslušnými nadpisy pro zadání spodní a horní hranice rozsahu, který může Simple type nabývat. Jestliže uživatel zvolí hodnotu Simple type výčtového typu, zviditelní se textové pole s příslušnými nadpisy pro jejich zadání. Položky výčtu by měly být odděleny “,”.

Pokud uživatel zvolí Simple type `SEQUENCE OF`, zviditelní se uživateli pole se seznamem nabídky, do kterého se automaticky doplní výběr z definovaných datových kombinací a také se zviditelní textová pole pro zadání spodní a horní hranice rozsahu. Formulář specifikace informací protokolu při definování dat je zobrazen na obrázku 8.5.

Obrázek 8.5. Formulář specifikace informací protokolu při definování dat

Uživatel uloží data pomocí stisknutí tlačítka **Next data**. Data, která uživatel definoval, se uloží do slovníku, který by měl sloužit k lepší manipulaci s entitami. Pole se seznamem nabídky i textová pole se před definováním dalších dat vymažou.

Do vlastností zapisujeme informace ve tvaru, které vychází ze syntaxe ASN.1 a ACN. Pokud je hodnota pole se seznamem nabídky pro zadání vlastnosti Endian rovna “big”, v jazyce ACN se nezapisuje, proto ho do pomocného slovníku neuložíme a string bude tedy prázdný.

Po uložení a zpracování informací se jednotlivá definovaná data zobrazí v seznamu pod klíčem Key, tedy jejich názvy.

Jestliže kombinace datových informací nebyla zatím použita, je uživatel vyzván, aby zadal této kombinaci název. Uživateli se zobrazí formulář volby názvu datové kombinace, který bude popsán níže.

Formulář volby názvu datové kombinace se uživateli po uložení automaticky zavře, aby nebyl uživatel zahlcen velkým počtem otevřených formulářů. Jestliže datová kombinace již existuje, doplní název aplikace automaticky. Pokud uživatel již nedefinoval všechna data, která by měl protokol obsahovat stiskne tlačítko **Create**. Formulář specifikace informací protokolu po nadefinování obsahu paketu je zobrazen na obrázku 8.6.

Obrázek 8.6. Formulář specifikace informací protokolu po nadefinování obsahu paketu

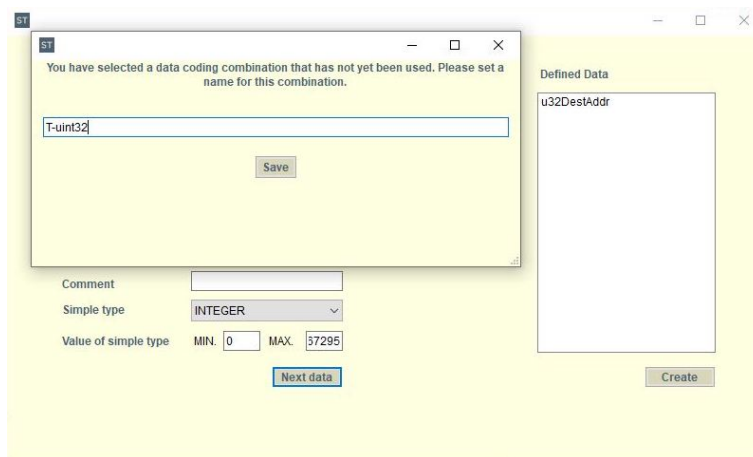
Tlačítko **Create** zpracuje data ze seznamu a aplikace je přidá do slovníku nadefinovaných datových kombinací a paketů.

Pro přidání do seznamu nadefinovaných protokolů se zavolá metoda `UpdateListBox()` hlavního formuláře. Po přidání do slovníku nadefinovaných TC a TM paketů se formulář zavře.

Metoda `UpdateListBox()` zjistí poslední prvek ve slovníku nadefinovaných protokolů a dle klíče (Key) slovníku klíč prvku zobrazí v seznamu paketů TC či TM. Uživateli se zobrazí v seznamu název protokolu.

8.4 Formulář volby názvu datové kombinace

Na formulář pro zadání názvu datové kombinace se dostane uživatel pomocí stisknutí tlačítka `Next` z formuláře specifikace informací protokolu. Formulář uživatele vyzve k zadání názvu nové datové kombinace textem, který zobrazí nadpis: “You have selected a data coding combination that has not yet been used. Please set a name for this combination.” Formulář volby názvu datové kombinace je zobrazen na obrázku 8.7.



Obrázek 8.7. Formulář volby názvu datové kombinace

Uživatel zadá název do textového pole a uloží ho stisknutím tlačítka `OK`. Název z textového pole se uloží do `public static string`. Pokud je datová kombinace již definována, doplní se automaticky. Aby neměl uživatel otevřeno více formulářů než je třeba, formulář se uzavře.

8.5 Formulář zobrazení informací TC a TM paketů

Na formulář zobrazení informací k TC a TM paketů se uživatel dostane z hlavního formuláře pomocí stisknutí tlačítka `Show`. Základní informace, jako jsou název paketu, jeho structured type a informace o jeho službě a podslužbě, jsou zobrazeny využitím textu. Pro vizualizaci obsahu paketů i s jeho kódováním bylo využito tabulky (`dataGridView`). Formulář zobrazení informací TC je zobrazen na obrázku 8.8.

Níže je uveden kód prvního řádku tabulky, který definuje jména sloupců tabulky:

```
dataGridViewShow.ColumnCount = 8;
dataGridViewShow.Columns[0].Name = "Name";
dataGridViewShow.Columns[1].Name = "Comment";
dataGridViewShow.Columns[2].Name = "Type";
dataGridViewShow.Columns[3].Name = "Simple Type";

dataGridViewShow.Columns[4].Name = "Constraint";
dataGridViewShow.Columns[5].Name = "Size [bits]";
dataGridViewShow.Columns[6].Name = "Encoding";
dataGridViewShow.Columns[7].Name = "Endianness";
```

Name: T-TC-6-2-MemLoad
Structured type: SEQUENCE
Present: serviceld = 6 AND subserVICeld = 2

Data:

Name	Comment	Type	Simple Type	Constraint	Size [bits]	Encoding
u32DestAddr		T-uint32	INTEGER	(0..4294967295)	32	pos-int
u8Length		T-uint8	INTEGER	(0..255)	8	pos-int
au8Data		SEQUENCE SIZ...			u8Length	

Obrázek 8.8. Formulář zobrazení informací TC paketu

8.6 Dokumentace

Kliknutím tlačítka **Create** z rozbalené nabídky menu **Documentation** na vrchní liště hlavního formuláře, se uživateli otevře `saveFileDialog` okno, kde si uživatel může zadat cestu uložení a název dokumentu.

Pro usnadnění výběru správného typu souboru bylo využito vlastnosti dialogu `Filter`, kde byl definován `LaTeX`. Následně aplikace vygeneruje `.tex` soubor (dokument `LaTeX`) jak již bylo popsáno v kapitole 6.3, který si uživatel sestaví.

Obsah dokumentu spojuje strukturu `ASN.1` s prvky zobrazení, které bylo využito ve formuláři zobrazení informací TC a TM paketů. Pakety, ke kterým bude dokumentace vytvořena, vycházejí ze seznamu TC a TM **To transfer**.

Pokud při tvorbě dokumentace nastane chyba, je na ni uživatel upozorněn `MessageBoxem` s hláškou: "An error occurred while creating the `.tex` document. Try to generate the documentation again." Příklad obsahu dokumentace je zobrazen na obrázku 8.9.

1 OVERVIEW

The documentation was generated by the SPACEE TALKS application, which was created as part of the diploma theses Bc. Hana Pikulikova. The application is used for the full specification of communication packets according to the ECSS-E-ST-70-41C standard using the ASN.1 and ACN languages up to communication with the device on board the satellite. The communication protocols that are used to communicate with the satellite communicate via the space link. The data that is transmitted uses the telecommand (TC) and telemetry (TM) protocols. These protocols provide a two-way communication flow with the satellite.

2 TC PROTOCOL

TC messages are received by the system, decoded and directed to the systems to which the messages are intended. TCs are mainly divided into direct spacecraft reconfiguration commands and application-specific commands.

2.1 TC PACKETS

ACN parameter	Type	Simple Type	Constraint	Size[bits]	Encoding	Endianness
serviceld	T-uint8	INTEGER	(0..255)	8	pos-int	big
subserviceld	T-uint8	INTEGER	(0..255)	8	pos-int	big

Name	Present	Type
tc -6-2-MemLoad	Present: serviceld = 6 AND subserviceld = 2	T-TC-6-2-MemLoad
tc -6-4-MemDump	Present: serviceld = 6 AND subserviceld = 4	T-TC-6-4-MemDump
tc -6-8-MemCheck	Present: serviceld = 6 AND subserviceld = 8	T-TC-6-8-MemCheck
tc -6-134-FlashMassErase	Present: serviceld = 6 AND subserviceld = 134	T-TC-6-134-FlashMassErase
tc -6-136-BankSwapEnable	Present: serviceld = 6 AND subserviceld = 136	T-TC-6-136-BankSwapEnable
tc -6-138-BankSwap	Present: serviceld = 6 AND subserviceld = 138	T-TC-6-138-BankSwap
tc -6-144-SysMemParamSet	Present: serviceld = 6 AND subserviceld = 144	T-TC-6-144-SysMemParamSet
tc -6-146-MemCheckStart	Present: serviceld = 6 AND subserviceld = 146	T-TC-6-146-MemCheckStart
tc -6-148-MemCopyStart	Present: serviceld = 6 AND subserviceld = 148	T-TC-6-148-MemCopyStart
tc -6-160-SysMemParamRequest	Present: serviceld = 6 AND subserviceld = 160	T-TC-6-160-SysMemParamRequest

2.1.1 T-TC-6-2-MEMLOAD

Structure type: SEQUENCE

Present: serviceld = 6 AND subserviceld = 2

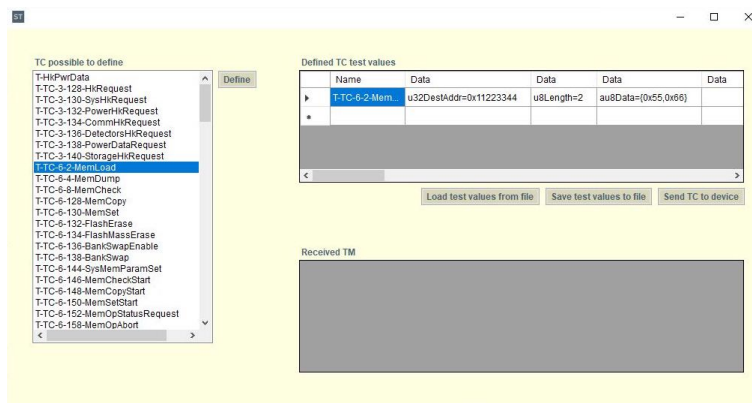
Name	Type	Simple Type	Constraint	Size[bits]	Encoding	Endianness
u32DestAddr	T-uint32	INTEGER	(0..4294967295)	32	pos-int	little
u8Length	T-uint8	INTEGER	(0..255)	8	pos-int	big
au8Data	SEQUENCE SIZE(0..129) OF T-uint8					

Name	Comment
u32DestAddr	
-u8Length	
au8Data	

Obrázek 8.9. Dokumentace

8.7 Formulář komunikace se satelitem

Kliknutím na tlačítko **Test Values** z hlavního formuláře se uživateli otevře formulář komunikace se satelitem. Seznam formuláře se při otevření doplní nadefinovanými TC pakety. Aby uživatel definoval testovací hodnoty pro daný paket, vybere paket ze seznamu a stiskne tlačítko **Define**. Touto akcí se uživateli otevře formulář specifikace hodnot dat TC paketů, který bude popsán níže. Nadefinované testovací hodnoty paketu i s názvem paketu se zobrazí v aplikaci do tabulky testovacích hodnot TC paketů pomocí zavolání metody `UpdateDataGrid()` z formuláře specifikace hodnot dat TC paketů. Počet sloupců vychází z nejvyššího počtu dat co se v paketech, definovaných v aplikaci vyskytují. Formulář komunikace se satelitem je zobrazen na obrázku 8.10.



Obrázek 8.10. Formulář komunikace se satelitem

Nadefinované testovací hodnoty si může uživatel uložit do .txt souboru pomocí stisknutí tlačítka **Save test values to file**. K uložení dat je využit control prvek aplikace `saveFileDialog`, který otevře průzkumníka souborů uživatele. Aby uložil uživatel testovací hodnoty do požadovaného formátu, bylo použito vlastnosti dialogu `Filter`, kde je definován typ souboru .txt. Jednotlivé řádky tabulky jsou načítány, parsovány a ukládány do formátu, který je zobrazen na obrázku 8.11. Pokud při ukládání souboru dojde k chybě, je na ni uživatel upozorněn `MessageBoxem` s hláškou: “An error occurred while creating the file. Try again.”

```

T-TC-6-2-MemLoad | u32DestAddr=0x11223344; u8Length=2; au8Data={0x55, 0x66};
T-TC-6-4-MemDump | u32SrcAddr=0x00; u8Length=0x01;
T-TC-6-8-MemCheck | u32DestAddr=0x55; u32Length=0x66;
T-TC-6-134-FlashMassErase | eFlashId=0x01; u8MemId=0x02;
T-TC-6-136-BankSwapEnable | u16UnlockKey=0x02;
T-TC-6-138-BankSwap |
T-TC-6-144-SysMemParamSet | u16ParamId=0x03; u32ParamValue=0x06;
T-TC-6-146-MemCheckStart |
T-TC-6-148-MemCopyStart |
T-TC-6-160-SysMemParamRequest | u16ParamId=0x02;

```

Obrázek 8.11. Formát uložených testovacích hodnot TC paketů do textového souboru

Uživatel nemusí testovací hodnoty v aplikaci definovat a jen si je načte taktéž z .txt souboru kliknutím na tlačítko **Load test values from file**. Uživateli však nebude zaručena správná komunikace se satelitem. K načtení dat ze souborů využívá aplikace control prvek `OpenFileDialog`, který je schopný otevřít okno průzkumníka souborů uživatele. Pro otevření souboru jen nutného typu .txt je docíleno vlastností dialogu `Filter`, kde je definován přípustný typ souboru .txt. Jednotlivé testovací hodnoty TC paketů jsou čteny ze souboru po řádcích a parsovány do jednotlivých sloupců dat.

Formát souboru se očekává shodný, jaký je použit při ukládání testovacích hodnot TC paketů do souboru. Pokud při čtení souboru dojde k chybě, je na ni uživatel upozorněn `MessageBoxem` s hláškou: “An error occurred while loading document. Please controll format of each line: Name—data=value,data=value.”

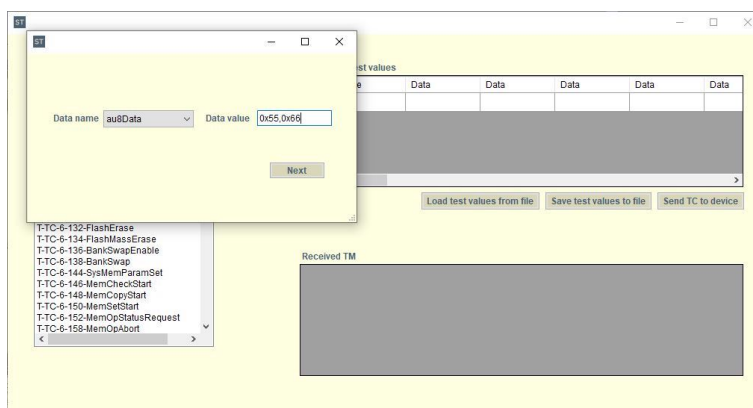
Pro zahájení komunikace se satelitem uživatel stiskne tlačítko **SendTC to device**. Stisknutí otevře formulář volby IP adresy a portu. Ve formuláři bude uživateli po skončení komunikace se satelitem zobrazena odpověď satelitu, tedy obsah TM paketu, kterou zajišťuje formulář volby IP adresy a portu.

8.8 Formulář specifikace hodnot dat TC paketů

Na formulář pro specifikaci hodnot dat TC paketů se uživatel dostane z formuláře komunikace se satelitem pomocí kliknutí na tlačítko **Define**. TC paket, pro který chce uživatel definovat hodnoty dat, vybere ze seznamu definovaných TC v aplikaci.

Formulář specifikace hodnot testovacích dat TC paketů se skládá z pole se seznamem nabídky názvu dat a textového pole specifikace hodnot dat. Obsah pole se seznamem nabídky se naplní automaticky názvy dat, které vychází ze specifikovaných TC paketů v aplikaci. Pokud paket obsahuje data a jejich délku, je název Length (délka) dle syntaxe dle ASN.1 doplněna o dvě pomlčky. Tyto pomlčky se uživateli do pole se seznamem nabídky oddělí a vidí tedy jen název Length.

Hodnota dat je uložena mezi složené závorky, aby byly později v zápisu rozeznány a jednotlivé hodnoty jsou uživatelem odděleny čárkou. Doplněnou hodnotu uživatel ve formuláři uloží pomocí tlačítka **Next** a z pole se seznamem nabídky názvu dat pro definování se název vymaže, aby nedošlo k duplicitě definování. Po definování všech dat je formulář automaticky zavřen, aby neměl uživatel otevřeno více formulářů než je třeba. Data i s hodnotami se uživateli zobrazí pomocí metody UpdateDataGrid() formuláře komunikace se satelitem. Formulář specifikace hodnot dat TC paketů je zobrazen na obrázku 8.12.

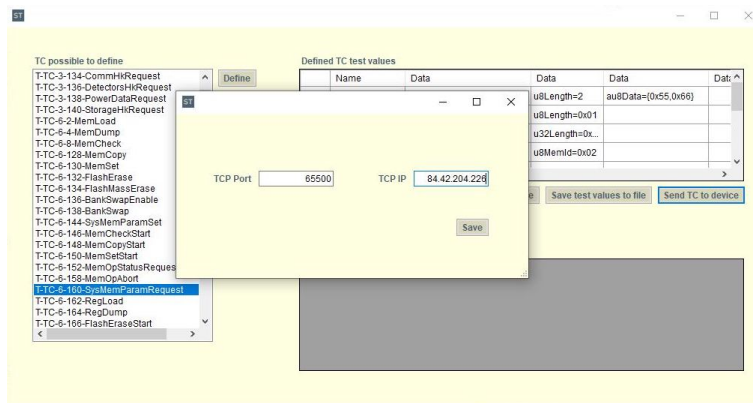


Obrázek 8.12. Formulář specifikace hodnot dat TC paketů

8.9 Formulář volby IP adresy a portu

Na formulář pro zahájení komunikace a nastavení nezbytných informací pro její uskutečnění se uživatel dostane z formuláře komunikace se satelitem pomocí stisknutí tlačítka **Send TC to device**.

Pro komunikaci mezi aplikací a TCP serverem, je nutné definovat TCP port a TCP IP adresu. Formulář volby IP adresy a portu se tedy skládá z textového pole pro specifikaci portu a textového pole specifikace IP adresy. Potvrzení zadaných hodnot proběhne kliknutím na tlačítko **Save** a je zahájena série metod spojená s komunikací s TCP serverem. Formulář volby IP adresy a portu je zobrazen na obrázku 8.13.



Obrázek 8.13. Formulář volby IP adresy a portu

8.9.1 Komunikace se zařízením

Komunikace je uskutečněna python kódem, který aplikace automaticky vygeneruje a spustí. Informace o serveru, tedy jeho IP adresa a portu, přebírá kód od uživatele z formuláře volby IP adresy a portu. Serveru jsou zasílány zakódované TC pakety, které uživatel v aplikaci nadefinoval a informace o jeho hodnotách dat jsou uloženy v tabulkách ve formuláři komunikace se satelitem.

Bohužel nebylo docíleno spuštění programu kompilátoru ASN1SCC aplikací, neboť veřejnosti dostupný kompilátor vygeneruje zdrojové kódy pouze v jazyce C. Avšak firma esc Aerospace poskytla k vypracování práce upravený kompilátor ASN1SCC, kterým byly vygenerovány soubory pro zakódování a dekodování paketů TC a TM, které jsou aplikací načteny pomocí OpenFileDialogu. Je tedy docíleno ochrany duševního vlastnictví firmy ohledně upraveného kompilátoru.

Za pomoci napsaných příkazů v terminálu, kde byl spuštěn kompilátor, byly získány kódy pro zakódování a dekodování paketů. Využité příkazy:

- Příkaz, který generuje adresář SRC a ICD, do kterého se zadávají soubory ASN.1 a ACN:.

```
asn1sc --c-lang --acn-enc --out SRC --icdacn ICD/TMTC1.html iov.acn
iov.acn
```

- Příkaz, který generuje adresář PY

```
asn2dataModel -toPython -o PY iov.acn
```

- Příkaz pro kompilaci pod OS Linux:

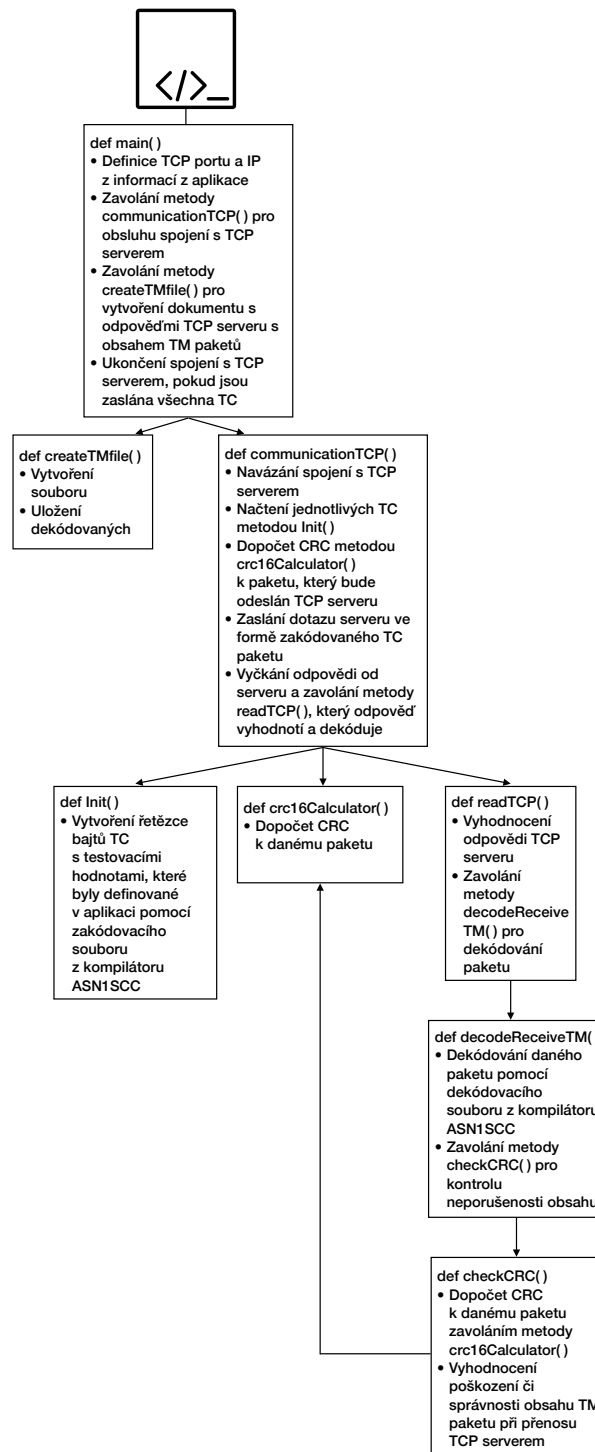
```
cd SRC && gcc -Wall -Wno-maybe-uninitialized *.c -o main
cd PY && make -f ./Makefile.python
```

- Příkaz pro kompilaci pod OS Windows pro vznik DLL:

```
použít CodeBlock, mingw
-std=c99
-static
```

Proces použití kompilátoru ASN1SCC je zobrazen na obrázku 6.2. Po načtení kódu pro zakódování a dekodování paketů uživatel pomocí saveFileDialogu pojmenuje python kód pro komunikaci s TCP serverem. Aby byl docílen bezchybný průběh kódu, musí být python kód umístěn ve stejné složce jako soubory kompilátoru ASN1SCC, tím však

není uživatel zatížen a cesta, kde bude kód komunikace uložen je převzata z OpenFileDialog načtením souborů kompilátoru ASN1SCC. Diagram metod a propojení metod generovaného kódu komunikace je zobrazen na obrázku 8.14.



Obrázek 8.14. Diagram metod a jejich propojení python kódu komunikace

Jak již bylo zmíněno, TC pakety, které budou serveru zasílány, jsou definovány v tabulce nadefinovaných hodnot TC paketů. Metoda `Init()` načte postupně řádky tabulky,

které rozparsuje a vygeneruje příkazy, které vycházejí z definice zakódování TC paketu z kompilátoru ASN1SCC. Pakety jsou zakódovány pomocí metod získaných z kompilátoru ASN1SCC. Příklad příkazů pro zakódování obsahu paketu:

```
myTC = T_TC()
myTC.sentData.kind.Set(DV.tc_6_2_MemLoad_PRESENT)
myTC.sentData.tc_6_2_MemLoad.u32DestAddr.Set(0x11223344)
myTC.sentData.tc_6_2_MemLoad.au8Data.SetLength(2)
myTC.sentData.tc_6_2_MemLoad.au8Data[0].Set(0x55)
myTC.sentData.tc_6_2_MemLoad.au8Data[1].Set(0x66)
```

Zprávy jsou satelitu posílány ve formě TC paketů, jehož hlavička obsahuje typ služby a podslužby dle účelu, který zpráva má, jak bylo popsáno v kapitole 3. Každý paket obsahuje Cyklický redundantní součet (CRC), který zajišťuje informaci o spávnosti přenosu dat. Pokud se vypočtená hodnota CRC s hodnotou CRC neshoduje, došlo při přenosu k poškození obsahu paketu, co může značit, že se změnila hodnota dat i jejich význam. CRC se vypočítává ze zakódovaného TC, tedy byte stringu metodou `crc16Calculator`. Vypočtená hodnota je k bajt stringu připojena až před samotným odesláním paketu serveru. Jak již název naznačuje, jde o výpočet posledních dvou bajtů komunikačního paketu a vychází z výpočtu polynomu CRC-16-CCITT.

$$x^{16} + x^{12} + x^5 + 1$$

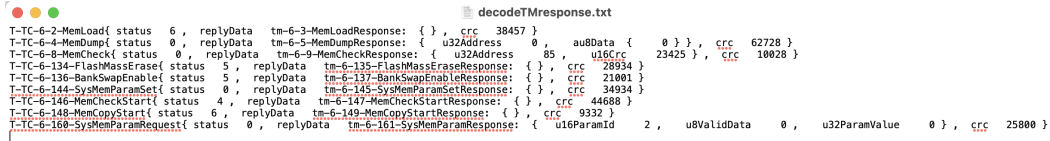
Při příjmu zprávy satelitem je hodnota CRC zkontrolována s hodnotou dopočtenou z přijatého bajt stringu také metodou `crc16Calculator` a je vyhodnoceno, zda se shoduje nebo vznikla chyba. Pokud se hodnota CRC shoduje, je paket dekodován pomocí dekodovacích metod získaných z kompilátoru ASN1SCC. Informace z dekodovaného paketu je uložena do souboru, který je uložen do textového souboru, který je umístěn do stejné složky jako kód pro komunikaci se zařízením. Struktura dekodované odpovědi vychází z metody `PrintAll()` kompilátoru ASN1SCC, do které byly vkládány bajt stringy TM paketu. Odpovědi paketů jsou doplněny aplikací i o informaci k jakému dotazu - TC paketu se vztahuje.

Po uložení kódu komunikace se serverem je kód aplikací automaticky spuštěn za pomoci knihovny IronPython. Umístění kódu je v aplikaci již uloženo a tak je tedy docíleno plně automatizace.

8.9.2 Vizualizace TM odpovědí přijatých od TCP serveru

Po skončení komunikace se zařízením je aplikací automaticky načten soubor s dekodovanými hodnotami TM odpovědí. Umístění souboru je taktéž v aplikaci uloženo a proces je tedy plně automatický.

Dekódovaná odpověď je z textového souboru načítána po řádcích a rozparsována na jednotlivá data s jejich hodnotami, které uživateli vizualizuje v aplikaci v tabulce vizualizace přijatých TM odpovědí formuláře komunikace se satelitem. Ukázka formátu .txt souboru, ze kterých jsou dekodované odpovědi ve formě TM paketů je zobrazena na obrázku 8.15.



```

T-TC-6-2-MemLoad{ status 6, replyData tm-6-3-MemLoadResponse: { }, crc 38457 }
T-TC-6-4-MemDump{ status 0, replyData tm-6-5-MemDumpResponse: { u32Address 0, au8Data { 0 } }, crc 62728 }
T-TC-6-8-MemCheck{ status 0, replyData tm-6-9-MemCheckResponse: { u32Address 85, u16Crc 23425 }, crc 10028 }
T-TC-6-134-FlashMassErase{ status 5, replyData tm-6-135-FlashMassEraseResponse: { }, crc 28934 }
T-TC-6-136-BankSwapEnable{ status 5, replyData tm-6-137-BankSwapEnableResponse: { }, crc 21001 }
T-TC-6-144-SysMemParamSet{ status 0, replyData tm-6-145-SysMemParamSetResponse: { }, crc 34934 }
T-TC-6-146-MemCheckStart{ status 4, replyData tm-6-147-MemCheckStartResponse: { }, crc 44688 }
T-TC-6-148-MemCopyStart{ status 6, replyData tm-6-149-MemCopyStartResponse: { }, crc 9332 }
T-TC-6-160-SysMemParamRequest{ status 0, replyData tm-6-161-SysMemParamResponse: { u16ParamId 2, u8ValidData 0, u32ParamValue 0 }, crc 25800 }

```

Obrázek 8.15. Formát uložených dekodovaných TM paketů v textovém souboru

Po skončení všech metod spojených s komunikací s TCP serverem je formulář volby IP adresy a portu automaticky zavřen.

Kapitola 9

Testování aplikace

Testování aplikace proběhlo za pomoci firmy esc Aerospace, která poskytla přístup ke komunikaci se Softwarovým nástrojem Space Dosimetry System Demonstrator (2SD).

9.1 Přístroj testování

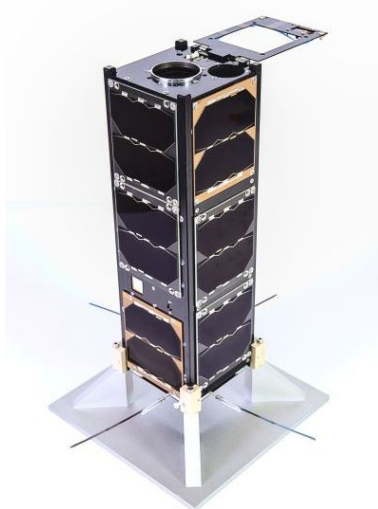
Dne 13. ledna 2022 byla na oběžnou dráhu vypuštěna česká technologická nanodružice VZLUSAT-2[25], jejíž součástí byl i Space Dosimetry System Demonstrator (2SD). Jedná se o pětivrstvý pixelový detektor kosmického záření.

9.1.1 VZLUSAT-2

Hlavním cílem mise je otestovat nové technologie pro budoucí vesmírné mise. Kromě zcela nově vyvinutých zařízení je zde i druhá generace zařízení, která se testovala již v rámci VZLUSAT-1. Nanodružice je sestavena na základě koncepce satelitu Cube-Sat.[26] Jedná se o malý satelit, který je složený ze tří krychlí, kde rozměry jedné krychle jsou 10 cm x 10 cm x 10 cm. Hmotnost VZLUSAT-2 nepřevyšuje 3,9 kg.[27]

VZLUSAT-2, který je vyobrazen na obrázku 9.1, se pohybuje ve výšce 500-600 km nad zemským povrchem s délkou periody oběhu 95 minut. Na svou oběžnou dráhu byla vynesena raketou Falcon 9 od společnosti Space X.

Se zařízením 2SD nese satelit i zařízení vesmírný rentgenový detektor, 2 kamery pro sledování Země, rentgenový sluneční senzor, detektor gama záblesků a senzor pro zaznamenání odplynění vodní páry.[28]

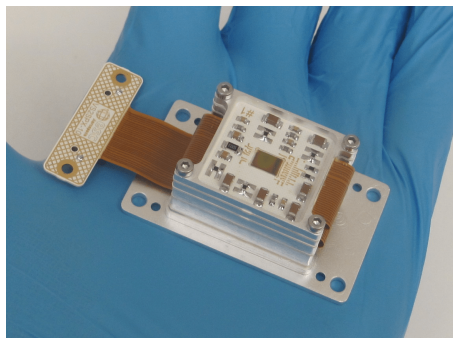


Obrázek 9.1. VZLUSAT-2[29]

■ 9.1.2 Softwarový nástroj Space Dosimetry System Demonstrator (2SD)

Space Dosimetry System Demonstrator je zařízení vyobrazené na obrázku 9.2, které detekuje kosmické záření a vyhodnotí složení záření na elektrony, protony a těžké ionty. Přínosem detekce záření by byla ochrana citlivých technologií umístěných na družici tím, že by se družice natáčela a tím chránila nejvíce citlivé technologie. Přístroj se skládá ze 2 detektorů záření Soft X-ray Monitor (SXM) a SpacePix Radiation Monitor (SXRМ). SXRМ obsahuje 5 detektorů SpacePix2. Detektory SpacePix2 zkoumají radiaci ve vesmíru, jelikož takové záření způsobuje poškození elektronických prvků systémů.

Výhodou přístroje 2SD je jeho malá velikost, hmotnost a spotřeba. Přístroj funguje na čipu SpacePix2. Provozní teplota zařízení je -40 až $+65$ °. Komunikace se zařízením probíhá pomocí sériové linky.[30] Přístroj 2SD je zobrazen na obrázku 9.2



Obrázek 9.2. Space Dosimetry System Demonstrator (2SD)[30]

■ 9.2 Testování aplikace

Hlavním cílem testování vzniklého softwarového nástroje je ověření jeho kvality a funkčnosti, než bude distribuován uživatelům. Testování probíhá zpravidla pro celou množinu procesů, které aplikace umožňuje. Výsledkem testování jsou zpravidla i nalezené problémy - chyby aplikace. Pro účel aplikace Spacee talks použijeme manuální testování, které umožní zkontrolovat i UI aplikace a díky malé velikosti aplikace zcela postačí.

Testování se provádí na základě vytvořeného testovacího scénáře s předem připravenými pomocnými testovacími soubory, jako jsou například vstupy aplikace.

■ 9.2.1 Testovací scénáře

Testovací scénáře jsou vytvářeny vývojáři pro ověření funkčnosti aplikace před jejím předáním koncovým uživatelům. Scénář je psán v logicky navazujících se krocích s očekávanými výstupy a vstupy jednotlivých kroků. V tabulce 9.1 jsou testovací scénáře i s předpokládanými výsledky.

Akce	Data	Předpokládaný výsledek
Načtení souborů	soubory ASN.1 a ACN	Načtení struktury paketů a jejich informací do aplikace.
Úprava paketů TC a TM		Odstranění a přidání paketů TC a TM v aplikaci.
Nadefinování testovacích hodnot TC paketů		V aplikaci nadefinování testovacích hodnot u zvolených TC paketů.
Vygenerování zdrojového kódu aplikací a jeho následné spuštění	soubory z kompilátoru ASN1SCC	Aplikace bude schopna načíst kódovací soubor z ASN1SCC kompilátoru a vygenerovat zdrojový kód v jazyce Python pro zakódování TC paketů a spojení s testovacím zařízením, který i spustí.
Dekódování přijatých zpráv a zobrazování v aplikaci	soubory z kompilátoru ASN1SCC	Vygenerovaný kód v jazyce Python použije dekodovací soubor z kompilátoru ASN1SCC k dekodování TM paketů, přijme TM pakety a následně je aplikace zobrazí.

Tabulka 9.1. Testovací scénář

9.3 Výsledky testování

Testování aplikace proběhlo za použití souborů, které poskytla firma esc Aerospace. Pro testování byly předem nadefinovány testovací scénáře s nutnými vstupy a předpokládanými výsledky, které jsou uvedeny v tabulce 9.1. Aplikace byla testována na zařízení s operačním systémem Windows x64 s grafickým adaptérem Intel(R) HD Graphics 3000 a procesorem Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz.

9.3.1 Načtení souborů

Do aplikace byly načteny soubory iov.asn a iov.acn, které byly poskytnuty firmou esc Aerospace. Načtené pakety se správně načetly a rozřadily podle typu TC a TM. Počty paketů TC a TM byly shodné s nadefinovanými počty v souborech. Pomocí tlačítka Show bylo zkontrolováno jejich obsah.

9.3.2 Úprava paketů TC a TM

V aplikaci bylo náhodně vybráno 10 TC a TM paketů ze služby 6, které jsou uvedeny v tabulkách. Tyto pakety byly odstraněny tlačítkem **Delete**. Pro přidání paketů bylo použito tlačítko **Add**, avšak bylo zjištěno, že poskytnuté soubory nejsou nadefinovány dle standardu ECSS-E-ST-70-41C a tedy pakety nelze přidat pomocí výběru podslužby služby 6 z nabídky comboBoxů. Pakety se přidaly pomocí volby služby "Other", která je dle standardu určena pro služby s ID > 128. Tímto postupem bylo přidáno všech 10 TC a TM paketů ze služby 6.

9.3.3 Nadefinování testovacích hodnot TC paketů

Testovací hodnoty byly přidány u zvolených TC paketů, které byly do aplikace přidány a jsou uvedeny v kapitole 9.3.2. Testovací hodnoty, které byly vybrány, jsou v tabulkách 9.2 až 9.11.

T-TC-6-2-MemLoad

Název dat	Hodnota dat
u32DestAddr	0x11223344
u8Length	2
au8Data	0x55,0x66

Tabulka 9.2. Testování hodnoty T-TC-6-2-MemLoad

T-TC-6-4-MemDump

Název dat	Hodnota dat
u32SrcAddr	0x00
u8Length	0x01

Tabulka 9.3. Testování hodnoty T-TC-6-4-MemDump

T-TC-6-8-MemCheck

Název dat	Hodnota dat
u32DestAddr	0x55
u32Length	0x66

Tabulka 9.4. Testování hodnoty T-TC-6-8-MemCheck

T-TC-6-134-FlashMassErase

Název dat	Hodnota dat
eFlashId	0x01
u8MemId	0x02

Tabulka 9.5. Testování hodnoty T-TC-6-134-FlashMassErase

T-TC-6-136-BankSwapEnable

Název dat	Hodnota dat
u16UnlockKey	0x02

Tabulka 9.6. Testování hodnoty T-TC-6-136-BankSwapEnable

T-TC-6-138-BankSwap

Název dat	Hodnota dat
-	-

Tabulka 9.7. Testování hodnoty T-TC-6-138-BankSwap**T-TC-6-144-SysMemParamSet**

Název dat	Hodnota dat
u16ParamId	0x03
u32ParamValue	0x06

Tabulka 9.8. Testování hodnoty T-TC-6-144-SysMemParamSet**T-TC-6-146-MemCheckStart**

Název dat	Hodnota dat
-	-

Tabulka 9.9. Testování hodnoty T-TC-6-146-MemCheckStart**T-TC-6-148-MemCopyStart**

Název dat	Hodnota dat
-	-

Tabulka 9.10. Testování hodnoty T-TC-6-148-MemCopyStart**T-TC-6-160-SysMemParamRequest**

Název dat	Hodnota dat
u16ParamId	0x02

Tabulka 9.11. Testování hodnoty T-TC-6-160-SysMemParamRequest

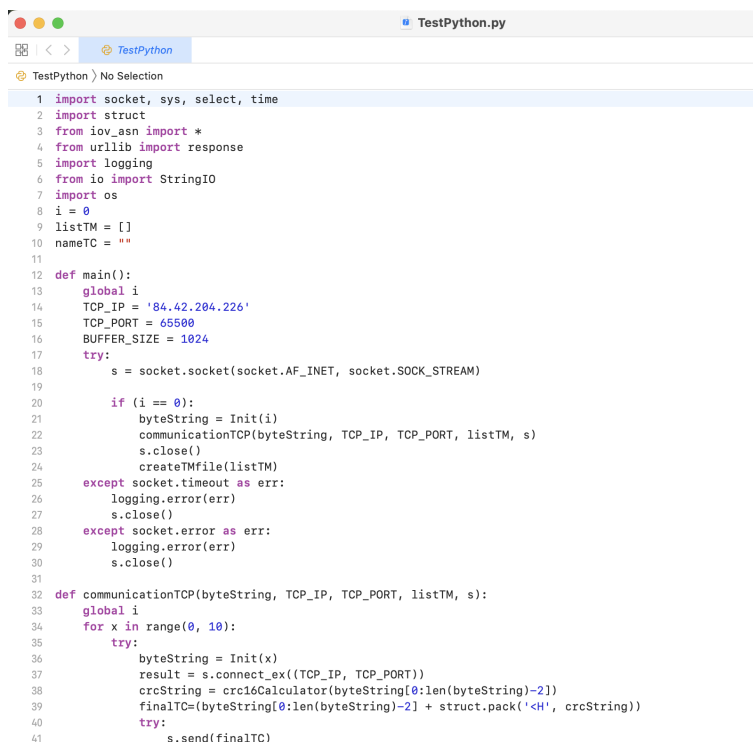
9.3.4 Vygenerování zdrojového kódu aplikací a jeho následné spuštění

Pro vygenerování zdrojového python kódu bylo stisknuto tlačítko `Send TC to device`, který uložil testovací hodnoty do souboru a otevřel formulář pro specifikaci IP adresy a portu TCP serveru. Tyto údaje byly získány rovněž od firmy esc Aerospace, která

softwarový nástroj Space Dosimetry System Demonstrator (2SD) k serveru připojila a definovala pro server i IP adresy, ze kterých se lze k zařízení připojovat.

```
TCP_IP = '84.42.204.226'
TCP_PORT = 65500
```

Aplikace po nadefinování vygenerovala python kód, jehož ukázka je na obrázku 9.3, automaticky ho spustila, přijaté odpovědi ve formě TM paketů dekodovala a uložila do souboru `decodeTMresponse.txt`.



```
1 import socket, sys, select, time
2 import struct
3 from io_ash import *
4 from urllib import response
5 import logging
6 from io import StringIO
7 import os
8 i = 0
9 listTM = []
10 nameTC = ""
11
12 def main():
13     global i
14     TCP_IP = '84.42.204.226'
15     TCP_PORT = 65500
16     BUFFER_SIZE = 1024
17     try:
18         s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
19
20         if (i == 0):
21             byteString = Inet(i)
22             communicationTCP(byteString, TCP_IP, TCP_PORT, listTM, s)
23             s.close()
24             createTMfile(listTM)
25         except socket.timeout as err:
26             logging.error(err)
27             s.close()
28         except socket.error as err:
29             logging.error(err)
30             s.close()
31
32 def communicationTCP(byteString, TCP_IP, TCP_PORT, listTM, s):
33     global i
34     for x in range(0, 10):
35         try:
36             byteString = Inet(x)
37             result = s.connect_ex((TCP_IP, TCP_PORT))
38             crcString = crc16Calculator(byteString[0:len(byteString)-2])
39             finalTC=(byteString[0:len(byteString)-2] + struct.pack('<H', crcString))
40             try:
41                 s.send(finalTC)
```

Obrázek 9.3. Ukázka python kódu pro komunikaci se zařízením

9.3.5 Dekódování přijatých zpráv a zobrazování v aplikaci

Jelikož je kód pro komunikaci se zařízením navrhnut tak, že přijaté odpovědi ve formě TM paketů dekoduje a uloží do souboru, jehož název je v aplikaci uložen, jsou odpovědi v aplikaci hned po skončení komunikace se zařízením zobrazeny. Obsah dekodovaných TM paketů i s TC pakety, na které reagují, jsou uvedeny v tabulkách 9.12 až 9.20.

Odpověď na T-TC-6-2-MemLoad: T-TM-6-3-MemLoadResponse

Název dat	Hodnota dat
-	-

Tabulka 9.12. Odpověď T-TM-6-3-MemLoadResponse

Odpověď na T-TC-6-4-MemDump: T-TM-6-5-MemDumpResponse

Název dat	Hodnota dat
u32Address	0
au8Data	0

Tabulka 9.13. Odpověď T-TM-6-5-MemDumpResponse**Odpověď na T-TC-6-8-MemCheck: T-TM-6-9-MemCheckResponse**

Název dat	Hodnota dat
u32Address	85
u16Crc	23425

Tabulka 9.14. Odpověď T-TM-6-9-MemCheckResponse**Odpověď na T-TC-6-134-FlashMassErase: T-TM-6-135-FlashMassEraseResponse**

Název dat	Hodnota dat
-	-

Tabulka 9.15. Odpověď T-TM-6-135-FlashMassEraseResponse**Odpověď na T-TC-6-136-BankSwapEnable: T-TM-6-137-BankSwapEnableResponse**

Název dat	Hodnota dat
-	-

Tabulka 9.16. Odpověď T-TM-6-137-BankSwapEnableResponse**Odpověď na T-TC-6-144-SysMemParamSet: T-TM-6-145-SysMemParamSetResponse**

Název dat	Hodnota dat
-	-

Tabulka 9.17. Odpověď T-TM-6-145-SysMemParamSetResponse**Odpověď na T-TC-6-146-MemCheckStart: T-TM-6-147-MemCheckStartResponse**

Název dat	Hodnota dat
-	-

Tabulka 9.18. Odpověď T-TM-6-147-MemCheckStartResponse

Odpověď na T-TC-6-148-MemCopyStart: T-TM-6-149-MemCopyStartResponse

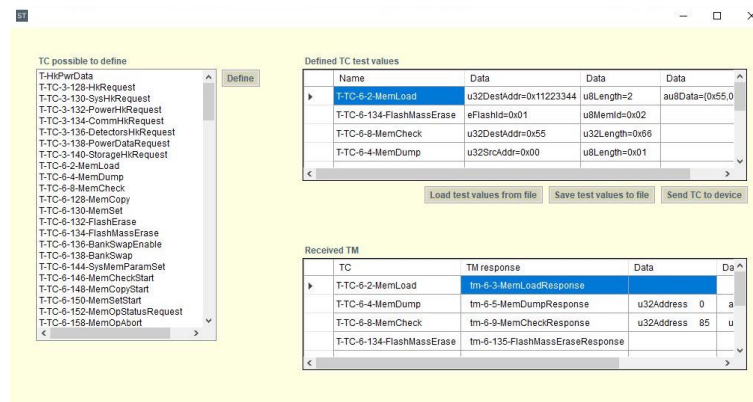
Název dat	Hodnota dat
-	-

Tabulka 9.19. Odpověď T-TM-6-149-MemCopyStartResponse**Odpověď na T-TC-6-160-SysMemParamRequest: T-TM-6-161-SysMemParamResponse**

Název dat	Hodnota dat
u16ParamId	2
u8ValidData	0
u32ParamValue	0

Tabulka 9.20. Odpověď T-TM-6-161-SysMemParamResponse

Z tabulek je patrné, že pro jeden TC paket, konkrétně T-TC-6-138-BankSwap nebyla přijata odpověď. Ze znalostí komunikace se satelitem bylo předpokládáno, že odpověď na tento TC paket by byl TM paket T-TM-6-139 a provedla jsem tedy kontrolu, zda bylo TM paketu nadefinováno v souborech iov.asn a iov.acn. Tento paket nadefinován v souborech nebyl. Zde mohla vzniknout tedy událost, že se přijatý TM paket nepodařilo dekodovat či se od TCP serveru nedostalo odpovědi. Vizualizace přijatých odpovědí ve formě TM paketů je zobrazena na obrázku 9.4.

**Obrázek 9.4.** Vizualizace přijatých odpovědí ve formě TM paketů

9.4 Nápady na vylepšení

Po otestování aplikace bylo provedeno zhodnocení celkového fungování aplikace a byly navrženy změny, které mohou přinést vylepšení uživatelskému dojmu a ovládání aplikace.

- Kontrola zadaných hodnot v textových polích, jestli uživatel zadává vstupní hodnotu například ve formátu string či integer.
- Omezení přechodu na další formulář, pokud uživatel nevyplní všechna textová pole či pole se seznamem nabídky v současném formuláři.
- Kontrola simple typu a jeho velikosti při zadávání testovacích hodnot TC paketu.
- Pokud by se od zařízení nezískalo odpovědi ve formě TM paketu či se paket nepodařilo dekodovat, aplikace by tuto skutečnost uchovála a zobrazila uživateli.

Kapitola 10

Závěr

Cílem práce bylo seznámit se s problematikou související s komunikací satelitů. Dále bylo cílem navrhnout softwarový nástroj v programovacím jazyce C#, ve kterém bude možné plně specifikovat komunikační protokol s obecným přístrojem na palubě družice a následně i s přístrojem komunikovat.

Diplomová práce byla řešena postupně ve třech stěžejních krocích. Přípravná etapa se zakládala na studiu odborných podkladů a výběru nástrojů, které budou využívány. Realizační etapa se zakládala především na programování aplikace a návrhu UX aplikace. Programování aplikace bylo rozděleno do čtyř stěžejních funkcionalit týkajících se správy paketů, tvorby dokumentace, komunikace se zařízením a následného dekódování přijatých paketů. Funkcionality byly programovány často i paralelně. Poslední etapa zahrnovala ladění aplikace a její testování. V závěru diplomové práce bylo provedeno vyhodnocení výsledků testování.

První část diplomové práce se zabývá teoretickými znalostmi o zavedení pojmu satelitu a jeho komunikaci. Jsou zde vymezeny pojmy telecommand a telemetry, tedy směřování toku zpráv. Dále se práce věnuje standardizaci v rámci komunikace satelitů. Druhou oblastí, kterou se praktická část diplomové práce zabývá, je jazyk ASN.1 a ACN s jejich složitou syntaxí. Tato oblast zahrnuje velké množství pojmů a důležitých prvků, které musí komunikace obsahovat a splňovat. V rámci první části práce jsem se setkala s malým množstvím informací týkajících se komunikace satelitů, stejně jako s chybějícími materiály v českém jazyce.

Praktická část diplomové práce se skládá z UX návrhu aplikace, její implementace a následného testování aplikace. Samotnému návrhu a tvorbě aplikace byla věnována velká část práce. Při návrhu aplikace byl kladen důraz na její přehlednost, funkčnost, intuitivnost jejího použití a zamezení chyb, které při zadávání syntaxe paketů mohou nastat. Pro tvorbu aplikace bylo vybráno Windows Forms v programovacím jazyce C#, kde se aplikace skládá z jednotlivých na sebe navazujících formulářů. Pro plnou funkčnost využívá aplikace jazyk Python pro komunikaci se zařízením a dokumentaci generuje za použití nástroje LaTeX. Aplikace je podporována na zařízeních s operačním systémem Windows 8 a novějším, kde je nainstalován .NET Desktop Runtime 6.0.12 (x64). Pro využití všech funkcí, které aplikace umožňuje, je doporučeno stáhnutí LaTeX editoru a programovacího jazyka Python verze 3.

Při tvorbě aplikace bylo nutné vyřešit několik výzev, které souvisely s dodržováním syntaxe ASN.1 a ACN, s propojením a komunikací se zařízením. Bylo navrženo řešení aplikace, které omezí výskyt chybných vstupů a tím umožní komunikaci se satelitem i uživateli s méně pokročilou znalostí standardizace a syntaxe, kterou komunikace se satelity přináší.

V závěru práce bylo provedeno testování aplikace a její zhodnocení. Testy byly provedeny se zařízením Space Dosimetry System Demonstrator (2SD), který reprezentuje přístroj použitý na družici VZLUSAT-2 a byl poskytnut firmou esc Aerospace. Zařízení bylo firmou připojeno na TCP server. Při testech bylo využito všech funkcí aplikace a tím byl i zhodnocen i přínos aplikace. Na základě testování bylo navrženo zlepšení

aplikace. Testování bylo provedeno na souborech, které pro testování nadefinovala firma esc Aerospace. Na počátku testování bylo zjištěno, že poskytnuté soubory a zařízení nejsou nadefinovány dle standardu ECSS-E-ST-70-41C a tím vzniklo riziko nesprávného fungování aplikace. Avšak díky funkci definování služeb s ID > 128 bylo zajištěno, že je možno definovat i služby, které nejsou výhradně dle standardu. Navržená aplikace byla úspěšně otestována a cíle diplomové práce bylo dosaženo.

Pro lepší použití aplikace a zamezení možných chyb, které mohou během definování paketů vzniknout, bych doporučila možné kontroly zadaných hodnot do textových polí a polí se seznamem nabídky dle uživatelem definovaného typu vstupní hodnoty. Také by do aplikace mohla být přidána kontrola zadávaných testovacích hodnot TC paketů s vlastnostmi, které jsou pro data v aplikaci nadefinovány. Dalším vylepšením by mohlo být omezení přechodu mezi formuláři aplikace, pokud uživatel nevyplní všechna potřebná pole, která formulář obsahuje.

Přínosem práce je vznik komplexní aplikace Spacee talks, která pokrývá tvorbu komunikačních zpráv, přiřazení testovacích hodnot TC paketů, vytvoření dokumentace k nadefinovaným paketům, navázání komunikace se zařízením, které je připojeno k TCP serveru a vizualizaci odpovědí na zaslané dotazy. Všechny funkcionality umožňuje aplikace provádět jednoduše, ulehčí velmi náročný a zdlouhavý proces komunikace se satelitem. Díky možnosti ukládání do textových souborů se stává i velice efektivní, protože lze proces přerušit a opět na něj navázat.

Literatura

- [1] *ESA - Telemetry & Telecommand* [online]. [2022-04-10]. Dostupné z: https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Computers_and_Data_Handling/Telemetry_Telecommand.
- [2] BECEDAS, Jonathan a DEMYANOV, Vladislav. *Satellites Missions and Technologies for Geosciences*. London: IntechOpen, 2020. ISBN 978-1-78985-996-6.
- [3] MAINI, Kumar Anil a AGRAWAL, Varsha. *Satellite Technology: Principles and Applications*. 1st edition. John Wiley & Sons Inc., 2007. ISBN 9780470033357.
- [4] *LEO, GTO, SSO a další – čím se liší různé orbity? – ElonX* [online]. [cit. 2022-11-01]. Dostupné z: <https://www.elonx.cz/leo-gto-ss0-a-dalsi-cim-se-lisi-ruzne-orbity/>.
- [5] KARTTUNEN, Hannu, KRÖGER, Pekka, OJA, Heikki, POUTANEN, Markku a DONNER, Karl Johan. *Fundamental Astronomy*. 6th edition. Berlin: Springer, 2017. ISBN 978-3-662-53044-3.
- [6] *Catalog of Earth Satellite Orbits* [online]. [2022-04-10]. Dostupné z: <https://earthobservatory.nasa.gov/features/OrbitsCatalog#:~:text=There%20are%20essentially%20three%20types,farthest%20away%20from%20the%20surface..>
- [7] *ESA - Types of orbits* [online]. [2022-04-10]. Dostupné z: https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits.
- [8] MONTENBRUCK, Oliver a GILL, Eberhard. *Satellite Orbits: Models, Methods, and Applications*. 1st edition. Springer, 2000. ISBN 978-3540672807.
- [9] LARSON, J. Wiley a WERTZ, R. James. *Space Mission Analysis and Design*. 3rd ed. Netherlands: Microcosm Inc., 1999. ISBN 1-881883-10-8.
- [10] FORTESCUE, Peter, SWINERD, Graham a STARK, John. *Spacecraft Systems Engineering*. 4th edition. UK: John Wiley & Sons, Inc., 2011. ISBN 978-0-470-75012-4.
- [11] *A New Approach to European Space Standards* [online]. [2022-04-10]. Dostupné z: [https://www.esa.int/esapub/bulletin/bullet81/krie81.htm#:~:text=Introduction-,The%20European%20Cooperation%20for%20Space%20Standardization%20\(ECSS\)%20is%20an%20initiative,states%20and%20their%20space%20industry..](https://www.esa.int/esapub/bulletin/bullet81/krie81.htm#:~:text=Introduction-,The%20European%20Cooperation%20for%20Space%20Standardization%20(ECSS)%20is%20an%20initiative,states%20and%20their%20space%20industry..)
- [12] *ESA - History of Europe in space* [online]. [2022-04-10]. Dostupné z: https://www.esa.int/About_Us/ESA_history/History_of_Europe_in_space.
- [13] *ESA - ESA facts* [online]. [2022-04-10]. Dostupné z: https://www.esa.int/About_Us/Corporate_news/ESA_facts.
- [14] *ECSS - A Single Set of European Space Standards — European Cooperation for Space Standardization* [online]. [2022-04-10]. Dostupné z: <https://ecss.nl/home/ecss-a-single-set-of-european-space-standards/>.

- [15] *Active Standards — European Cooperation for Space Standardization* [online]. [2022-04-10]. Dostupné z: <https://ecss.nl/standards/active-standards/>.
- [16] *ECSS-E-ST-70-41C. Space Engineering: Telemetry and telecommand packet utilization*[online]. Noordwijk: ESA Requirements and Standards Division, 2016. [cit. 2022-04-06]. Dostupné z: <https://ecss.nl/standard/ecss-e-st-70-41c-space-engineering-telemetry-and-telecommand-packet-utilization-15-april-2016/>.
- [17] *Satellite Navigation - GPS - Space Segment — Federal Aviation Administration* [online]. [2022-04-10]. Dostupné z: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/spacesegments.
- [18] *asics of Space Flight - Solar System Exploration: NASA Science* [online]. [2022-04-10]. Dostupné z: <https://solarsystem.nasa.gov/basics/chapter9-1/>.
- [19] *ESA - Ground Segment overview* [online]. [2022-04-10]. Dostupné z: https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Ground_Segment_overview.
- [20] DUBUISSON, Olivier. *ASN.1 Communication Between Heterogeneous Systems*. Morgan Kaufmann, 2000. ISBN 978-0126333619.
- [21] *Technical topic: ASN.1 - An introduction to ACN - TASTE* [online]. [2022-04-10]. Dostupné z: https://taste.tuxfamily.org/wiki/index.php?title=Technical_topic:_ASN.1_-_An_introduction_to_ACN.
- [22] SELLS, Chris, 2003. *Windows Forms Programming in C#*. Addison-Wesley Professional. ISBN 9780321116208.
- [23] *TASTE* [online]. [2022-04-10]. Dostupné z: <https://essr.esa.int/project/taste>.
- [24] *Technical topic: ASN1SCC - ESA's ASN.1 Compiler for safety-critical embedded platforms - TASTE* [online]. [2022-04-10]. Dostupné z: https://taste.tuxfamily.org/wiki/index.php?title=Technical_topic:_ASN1SCC_-_ESA%27s_ASN.1_Compiler_for_safety-critical_embedded_platforms.
- [25] *Raketa Falcon 9 vynese ve čtvrtek 13. ledna 2022 detektor Fakulty jaderné a fyzikálně inženýrské na oběžnou dráhu - Zpravodajský servis - České vysoké učení technické v Praze* [online]. [2022-02-12]. Dostupné z: <https://aktualne.cvut.cz/aktuality/20220112-raketa-falcon-9-vynese-ve-ctvrtek-13-ledna-2022-detektor-fakulty-jaderne-a>.
- [26] HUDEC, René et al. VZLUSAT-1: Nanosatellite with miniature lobster eye X-ray telescope and qualification of the radiation shielding composite for space application. *Acta Astronautica*. 2017, **140**(11), 96-104. ISSN 0094-5765.
- [27] *CubeSats Overview — NASA* [online]. [2022-04-10]. Dostupné z: https://www.nasa.gov/mission_pages/cubesats/overview.
- [28] *VZLUSAT2 — VZLÚ... partner přemyslu* [online]. [2022-04-10]. Dostupné z: <https://www.vzlu.cz/vzlusat2/?lang=en>.
- [29] *VZLUSAT-2 - Czech Space Portal* [online]. [cit. 2022-12-02]. Dostupné z: <https://www.czechspaceportal.cz/databaze/rejstrik-kosmickykh-objektu/vzlusat-2/>.

- [30] *Solutions and Innovation – esc Aerospace s.r.o, Inc* [online]. [cit. 2020-10-02]. Dostupné z:
<https://www.esc-aerospace.com/sro/solutions-and-innovation/>.

Příloha A

Seznam zkratk

AADL	■	Architecture Analysis and Design Language
ASN.1	■	Abstract Syntax Notation One
CCSDS	■	Meziagenturní poradní výbor pro vesmírné datové systémy
CLI	■	Common Intermediate Language
CLR	■	Common Language Runtime
CPU	■	Centrální procesorová jednotka
CRC	■	Cyklický redundantní součet
C&DH	■	Command and Data Handling
ECSS	■	Cooperation for Space Standardization
ELDO	■	European Launcher Development Organisation
ESA	■	Evropská vesmírná agentura
ESRO	■	European Launch Development Organisation
GEO	■	Geostacionární dráha
GPL	■	GNU General Public licence
GSP	■	Globální navigační systém
GTO	■	Geostacionární přenosová dráha
GUI	■	Grafické uživatelské rozhraní
HW	■	Hardware
ID	■	Identifikátor
IDE	■	Vývojové prostředí
IGS	■	Inerciální naváděcí systém
ISO	■	Mezinárodní organizace pro normalizaci
ITU-T	■	Union Telecommunication Standardization Sector
LEO	■	Nízká oběžná dráha
MEO	■	Střední oběžná dráha
NGWS	■	Next Generation Windows Services
OBC	■	On-Board Computer
PUS	■	Packet Utilization Standard
SDL	■	Specification and Description Language
SSO	■	Sluneční synchronní dráha
SW	■	Software
TASTE	■	The ASSERT Set of Tools for Engineering
TC	■	Telecommand
TCP	■	Transmission Control Protocol
TM	■	Telemetry
UI	■	User Interface
uPER	■	Unaligned Packed Encoding Rules
UX	■	User Experience
2SD	■	Space Dosimetry System Demonstrator

Příloha B

Elektronické přílohy

Program aplikace SPACEE TALKS	Složka SpaceTalksApp
Kód aplikace SPACEE TALKS	Složka SpaceeTalksCode
Testované soubory ASN.1 a ACN	iov.asn a iov.acn
Poskytnuté soubory z kompilátoru ASN1SCC	Složka iov

Tabulka B.1. Elektronické přílohy