

Fakulta matematiky, fyziky a informatiky UK

Technická dokumentácia

Projekt Serial transformers

Vypracovali : Ivana Bekešová, Andrej Nagy, Zdenko Németh, Mykola Shulhin

Obsah

1.	Katalóg požiadaviek	2
1.1.	Úvod	2
1.2.	Všeobecný popis	3
1.3.	Špecifické požiadavky	4
2.	Návrh.....	7
2.1.	Špecifikácia vonkajších interfejsov	7
2.2.	Dátový model perzistentných údajov pre transformačný modul.....	8
2.3.	Formát súborov pre transformačný modul pre teleskop	10
2.4.	Komunikačné protokoly.....	10
2.5.	Dátový model perzistentných údajov pre univerzálny transformačný modul	11
2.6.	Formát súborov pre univerzálny transformačný modul	12
2.7.	Návrh používateľského rozhrania.....	13
2.8.	Návrh implementácie	15
2.9.	Plán implementácie.....	20
3.	Testovacie scenáre	20
3.1.	Modul teleskop	20
3.2.	Modul universal.....	21

1. Katalóg požiadaviek

1.1. Úvod

1.1.1. Účel katalógu požiadaviek

Tento dokument vznikol v rámci predmetu "Tvorba informačných systémov" v akademickom roku 2024/2025 na Fakulte matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. Dokument opisuje všetku požadovanú funkcionality vyvíjaného projektu "Serial transformers". Je určený pre riešiteľov projektu, zadávateľa a vyučujúceho predmetu "Tvorba informačných systémov" a jeho cieľom je poskytnúť záväznú dohodu o funkčnosti navrhovaného systému medzi zadávateľom a riešiteľmi.

1.1.2. Rozsah produktu

Vyvinutý informačný systém bude pripojený do komunikačného rozhrania medzi riadiacim počítačom a krokovými motormi teleskopu v hvezdárni FMFI UK. Hlavnou úlohou systému bude transformácia sériovej komunikácie tak, aby umožňovala dynamické úpravy rýchlosti motorov teleskopu, čím sa kompenzuje degradácia hardvéru a zabezpečí presnejšie sledovanie nočnej oblohy. Alternatívne systém ponúkne určitú univerzálnu schopnosť prekladania paketov, ktoré bude vykonávané na základe špecifických pravidiel definovaných v konfiguračnom súbore transformácie.

1.1.3. Definície a skratky

- **produkt** - výsledné zariadenie a jeho softvér, spĺňajúce stanovené požiadavky
- **RA** - right ascension, pohyb v smere východ-západ, zemepisná dĺžka
- **DEC** - declination, pohyb v smere sever-juh, zemepisná šírka
- **telegram** - packet komunikácie
- **GUI** - grafické používateľské rozhranie

1.1.4. Referencie

- Github repozitár projektu - [Serial Transformer](#)
- Stránka predmetu - [Tvorba informačných systémov](#)
- Dodatočné materiály od zadávateľa - [Protokol](#)

1.1.5. Prehľad zvyšnej časti dokumentu

V druhej kapitole nazvanej "Všeobecný popis" je poskytnutý základný prehľad o produkte, jeho perspektíve a funkciách, ktoré podporujú efektívnu komunikáciu s teleskopom. Tretia kapitola s názvom "Špecifické požiadavky" detailne popisuje technické a funkčné požiadavky, ktoré musí systém splniť.

1.2. Všeobecný popis

1.2.1. Perspektíva produktu

Hvezdáreň FMFI UK prevádzkuje teleskop, ktorého súčiastky zodpovedné za pohyb časom degradujú, čo vedie k problémom pri snímaní fotografií s dlhšou uzávierkou kvôli pohybujúcim sa hviezdám. Naším cieľom je vyvinúť "blackbox", ktorý bude transformovať komunikáciu medzi riadiacim počítačom a controllerom motorov. Tento systém bude eliminovať chyby spojené s pohybom teleskopu a zabezpečí presné sledovanie hviezdnej oblohy.

Produkt bude fungovať v dvoch režimoch: manuálnom, kde používateľ môže priamo zasahovať do nastavení rýchlosti pohybu motorov, a automatickom, v ktorom bude nainštalovaná kamera na teleskope neustále sledovať pohyb oblohy a prispôbovať rýchlosť motorov podľa aktuálnych podmienok. Týmto prístupom sa zvýši kvalita snímok a efektívnosť pozorovania v hvezdárni.

1.2.2. Funkcie produktu

Produkt bude integrovaný do master-slave komunikácie medzi riadiacim procesorom a modulmi teleskopu. Jeho hlavnou úlohou bude filtrovať príkazy z prebiehajúcej komunikácie, ktoré ovládajú krokové motory pre osy RA a DEC. Na základe týchto filtrov bude systém dynamicky upravovať rýchlosť pohybu jednotlivých krokových motorov, čím umožní teleskopu presne sledovať vybranú časť nočnej oblohy a zachytiť snímku.

Dynamická úprava rýchlosti sa bude realizovať buď manuálne prostredníctvom grafického užívateľského rozhrania, kde používateľ môže zadať zmeny pre motory RA a DEC, alebo automaticky na základe údajov z Autoguider kamery. Táto kamera porovnáva snímky oblohy a zisťuje, ako sa obraz pohybuje, čím prispieva k presnej a efektívnej korekcii polohy teleskopu.

Systém bude evidovať záznamy o svojej činnosti, ktoré bude možné v konfiguračnom programe na PC zobrazovať či už v reálnom čase alebo dodatočne. Úroveň logovania bude konfigurovateľná v programe na PC.

Hardvér a softvér systému bude navrhnutý tak, aby umožňoval záznam (logovanie), filtrovanie (blokovanie časti komunikácie) a transformáciu paketov pre všeobecnú aplikáciu, nielen pre účel riadenia krokových motorov teleskopu. Súčasťou riešenia bude príklad jednoduchkej aplikácie pre logovanie, filtrovanie a transformáciu paketov.

1.2.3. Používateľské role

Používateľmi produktu budú výlučne vedci v hvezdárni FMFI UK, ktorí ho budú používať na prácu s riadením rýchlosti krokových motoroch teleskopu.

1.2.4. Všeobecné obmedzenia

Hlavným obmedzením produktu je rýchlosť a oneskorenie pri prijímaní paketov v komunikácii master-slave.

1.2.5. Predpoklady a závislosti

Potrebným predpokladom je, že externý softvér komunikujúci s autoguider kamerou bude aktívne komunikovať so zariadením na transformáciu paketov cez samostatný komunikačný kanál a posielat' zariadeniu na transformáciu paketov informáciu o aktuálnej chybe pozície teleskopu. Spôsob a formát tejto komunikácie bude jednoznačne definovaná.

1.3 Špecifické požiadavky

1.3.1. Spoločné požiadavky

1. Zariadenie umožní dva druhy funkcionality: 1) transformáciu paketov (pod čo spadá aj filtrovanie), 2) záznam paketov.
 - 1.1. Používateľ vie aktivovať každú funkcionality nezávisle, t.j. iba transformáciu, iba logovanie, alebo súčasnú transformáciu aj logovanie.
2. Funkcionalita záznamu paketov bude ukladať prebiehajúcu komunikáciu do súboru na PC pomocou softvéru na PC.
 - 2.1. ak je aktívna aj transformácia, v súbore budú uložené aj pôvodné pakety prijaté na vstupe aj výsledné pakety odoslané na výstupe a pre každý z nich aj "timestamp", ktorý môže byť potenciálne iný pre vstupné a výstupné pakety.
 - 2.2. ak transformácia nie je aktívna, pakety sú určené prestávkami v komunikácii, každý súvislý úsek komunikácie je považovaný za "paket" a v zázname ho predchádza "timestamp"
3. Transformáciu paketov budú vykonávať transformačné moduly.
4. Systém bude navrhnutý tak, aby pokročilý používateľ vedel pridať nový transformačný modul bez toho, aby musel meniť iné časti kódu.
5. Dodaný systém bude obsahovať dva transformačné moduly - špecifický pre úpravu *right ascension* teleskopu a univerzálny transformujúci pakety podľa regulárneho výrazu.
6. Používateľ bude môcť vybrať z dostupných transformačných modulov, ktorý z nich sa má používať.
7. Zariadenie bude mať softvérový prepínač, ktorým sa bude dať prepnúť do bypass módu, v ktorom sa všetka komunikácia priamo preposiela zo vstupu na výstup. To môže byť užitočné pri ovládaní teleskopu na diaľku vzdialeným pripojením pracovnej plochy.
8. Systém je možné softvérovo "resetnúť" samostatným fyzickým tlačidlom na zariadení i tlačidlom v rozhraní PC softvéru. Reset by nemal spôsobiť skutočný

reset riadiacej jednotky v zariadení, ale vynulovanie všetkých akumulovaných premenných, ktoré transformačné moduly využívajú.

9. PC softvér zobrazuje v akom móde je zariadenie prepnuté a ktorý transformačný modul je zvolený.
10. Všetky nastavenia vrátane nastavení transformačných modulov bude môcť používateľ robiť aj priamym editovaním konfiguračného súboru aj v grafickom rozhraní PC softvéru.
11. Základná konfigurácia sa bude vykonávať v základnom okne rozhrania PC softvéru a každý transformačný modul bude mať svoje vlastné okno (druh okna) pre svoju konfiguráciu a priebežné zobrazovanie svojich špecifických informácií.
12. Systém funguje neprerušovane nezávisle od toho, či je PC softvér spustený a či je zobrazené iba základné okno, alebo aj okno aktívneho transformačného modulu.

1.3.2. Transformačný modul pre teleskop

1. Používateľ bude môcť nastaviť pevnú konštantu zrýchlenia, resp. spomalenia teleskopu, napr. 1.005 znamená, že výsledná priemerná rýchlosť pohybu teleskopu za celý čas merania (ale aj za jeho krátke intervaly) v osi RA má byť o 5 promile rýchlejšia ako by bola bez transformácie.
2. Softvér bude cez svoje API dostupný externému softvéru na programatické stanovenie konštanty zrýchlenia (napr. pre autoguider kameru).
3. Používateľ bude môcť prepnúť medzi manuálne konfigurovanou konštantou (pozri 3.2.1.) a automatickým módom, kde konštanta priebežne pravidelne prichádza napr. z autoguider kamery a neustále sa mení (pozri 3.2.2).
4. Vo všetkých módoch bude "blackbox" transformovať komunikáciu podľa nastavenej konštanty.
5. Modul umožní konfiguráciu spôsobu transformácie paketov:
 - 5.1. "blackbox" okamžite preposiela všetky pakety, ale môže v nich meniť rýchlosť pohybu motora, pričom sa snaží dosiahnuť čo najkratšie (ideálne celkom zanedbateľné) zdržanie,
 - 5.2. "blackbox" nemení pakety, ale môže ich odoslanie (resp. ukončenie odoslania) pozdržať.
 - 5.3. kombinácia 5.1 a 5.2 - t.j. môže aj meniť rýchlosť aj pozdržať odoslanie.
 - 5.4. Rovnako ako 5.3, ale okrem toho môže "blackbox" niektoré pakety celkom vynechať a zároveň na výstup vyslať celkom nové pakety, ktoré neprišli zo vstupu.
6. Modul bude na základe prichádzajúcich a odchádzajúcich paketov neustále počítať pôvodnú určenú akumulovanú priemernú rýchlosť a výslednú akumulovanú priemernú rýchlosť.

7. Ak je spustený PC softvér, rýchlosti počítané podľa 6 sa budú ukladať do samostatného textového súboru pomenovaného podľa dátumu a času začiatku merania v samostatnom priečinku pre záznam rýchlostí teleskopu.
8. Ak je zobrazené okno transformačného modulu teleskopu, rýchlosti počítané podľa 6 sa budú:
 - 8.2. neustále zobrazovať:
 - 8.2.1. ako číselné hodnoty (textové okienko/štítok),
 - 8.2.2. v grafe s dvoma krivkami rôznej farby.
9. Frekvencie zobrazovania a ukladania v (7., 8.) sú rovnaké a táto frekvencia sa dá nastaviť ako desatinné číslo v Hz,
 - 9.1. špeciálna hodnota frekvencie "*" znamená, že sa rýchlosti uložia po každom jednom odoslanom pakete,
 - 9.2. špeciálna hodnota frekvencie "0" znamená, že logovanie rýchlostí a ostatných informácií modulu (pozri nižšie) je celkom vypnuté.
10. V grafickom okne transformačného modulu teleskopu aj v súbore s rýchlosťami (pozri 7. a 8.) sa okrem toho bude zobrazovať/ukladať aj počet paketov,
 - 10.1. v ktorých bola zmenená rýchlosť,
 - 10.2. ktoré boli pozdržané,
 - 10.3. ktoré boli odoslané bez zmeny,
 - 10.4. ktoré boli zadržané (neboli vôbec preposlané),
 - 10.5. ktoré vznikli (neprišli zo vstupu, ale boli poslané na výstup).
11. každý riadok záznamu v súbore s rýchlosťami obsahuje v prvom stĺpci "timestamp" v ms.

1.3.3. Univerzálny transformačný modul

1. Používateľ bude môcť zadať zoznam regulárnych výrazov, ktoré budú rozoberať vstupný prúd bajtov na pakety (telegramy). Zhoda doposiaľ načítanej postupnosti bajtov s ľubovoľným vzorom spôsobí vytvorenie paketu danej kategórie (kategória = index regulárneho výrazu v zozname).
2. Pre každý z regulárnych výrazov v 3.3.1. (pre každú kategóriu paketu) bude používateľ môcť zadať množinu transformácií (replace) regulárnymi nahradzovacími výrazmi ako dvojicami stringov (vyhľadávaný, výsledný), ktoré sa postupne všetky uplatnia na jednotlivé pakety (ak nie je zhoda, paket zostáva rovnaký) danej kategórie.
3. Regulárne výrazy budú môcť byť aplikované na ASCII pakety alebo aj binárne pakety a softvér pre účely transformácie binárnych paketov nejak umožní definovať v regulárnych výrazoch binárne hodnoty vyjadrené v HEXA kódach.

2. Návrh

2.1. Špecifikácia vonkajších interfejsov

Táto kapitola popisuje všetky vonkajšie rozhrania informačného systému, ktoré sú zodpovedné za jeho komunikáciu s ostatnými aplikáciami, súbormi, zariadeniami.

2.1.1. Hardvérové rozhranie

Táto kapitola popisuje použitý hardvér jeho špecifikáciu a účel, pre ktorý bol použitý.

2.1.1.1. Teensy 4.1 Microcontroller

Mikrokontrolér Teensy 4.1 slúži ako hlavná riadiaca jednotka systému, na ktorej beží softvér, ktorý spracúva prichádzajúce dáta a implementuje logiku transformácie a filtrovania paketov. V rámci informačného systému umožňuje transformovať, filtrovať pakety pre ovládanie rýchlosti krokových motorov teleskopu. Prepojenie s riadiacim PC je pomocou USB a s krokovými motormi teleskopu cez sériové rozhrania RS232 a RS485.

Špecifikácia:

- ARM Cortex-M7 at 600MHz
- 1024K RAM (512K is tightly coupled)
- 8 Mbyte Flash (64K reserved for recovery & EEPROM emulation)
- USB Host Port
- 2 chips Plus Program Memory
- 55 Total I/O Pins
- 3 CAN Bus (1 with CAN FD)
- 2 I2S Digital Audio
- 1 S/PDIF Digital Audio
- 1 SDIO (4 bit) native SD
- 3 SPI, all with 16 word FIFO
- 7 Bottom SMT Pad Signals
- 8 Serial ports
- 32 general purpose DMA channels
- 35 PWM pins
- 42 Breadboard Friendly I/O
- 18 analog inputs
- Cryptographic Acceleration
- Random Number Generator
- RTC for date/time
- Programmable FlexIO
- Pixel Processing Pipeline
- Peripheral cross triggering
- 10 / 100 Mbit DP83825 PHY (6 pins)
- microSD Card Socket
- Power On/Off management

2.1.1.2. RS232 na RS485 prevodník

Tento prevodník zabezpečuje transformáciu signálu medzi komunikačnými štandardmi RS232 (Teensy) a RS485 (krokové motory teleskopu).

2.1.2. Softvérové rozhranie

2.1.2.1. PC Softvér

Softvér slúži ako grafické používateľské rozhranie (GUI) pre výber modulu, jeho ovládanie, nastavenie parametrov a sledovanie stavu systému, komunikácie.

Štruktúra transformačného modulu pre teleskop:

- Prepínač režimov – manuálny/automatický/bypass
- Vstupná hodnota – manuálne nastavenie rýchlosti
- Vizualizácia transformácie – graficky znázornená transformácia s možnosťou nastavenia intervalu priemerovania
- Zobrazenie dát o paketoch – informácie o počte so zmenenou rýchlosťou, pozdržaných, nezmenených, zadržaných
- Tlačidlá pre manipuláciu logov – zobrazenie/stiahnutie/zmazanie

2.1.2.2. Komunikácia s Autoguider kamerou

Komunikácia s Autoguider kamerou prebieha cez webový server. Komunikačné rozhranie:

- Kamera odosiela údaje cez protokol HTTP alebo HTTPS na špecifický port (napr. 8080).
- Kamera posíla údaje v JSON formáte prostredníctvom HTTP POST požiadavky na adresu webserveru.
- Kamera odosiela údaje v pravidelných časových intervaloch.
- Webserver prijíma tieto POST požiadavky a zapisuje údaje do interného systému alebo logu.
- Ak webserver zaznamená chybu pri prijímaní údajov, loguje tento incident s časovou pečiatkou pre ďalšie diagnostické účely.

2.2. Dátový model perzistentných údajov pre transformačný modul

V dátovom modeli budú reprezentované údaje uchovávané počas celej prevádzky systému, ktoré sú dôležité pre konfiguráciu, logovanie a transformáciu komunikácie.

2.2.1. Konfiguračné údaje

Konfigurácia je uložená v štruktúrovanom súbore, ktorý obsahuje nasledujúce nastavenia:

- **Mód (mode):** prepína medzi automatickým a manuálnym módom.
- **Transformačný modul:** identifikátor aktívneho transformačného modulu (teleskopický alebo univerzálny).
- **Konštanty zrýchlenia/spomalenia (acceleration/deceleration constants):** nastavuje pevnú konštantu zrýchlenia pre os RA.
- **Bypass mód:** binárna hodnota určujúca, či systém funguje v móde bypass (1 = bypass, 0 = transformácia).
- **Logovanie činnosti:** úroveň logovania (napr. "plné logovanie," "iba chybové hlásenia").

Logovanie paketov: parameter, ktorý určuje, či sa majú logovať prichádzajúce a odchádzajúce pakety. "Áno" pre zapnutie logovania alebo "nie" pre jeho vypnutie.

2.2.2. Údaje o zaznamenaných paketoch

Každý záznam o pakete obsahuje:

- **ID záznamu (record_id):** unikátne ID záznamu.
- **Časová pečiatka (timestamp):** čas príchodu paketu do systému (a čas po transformácii, ak sa transformuje).
- **Originálny paket (original_packet):** surové dáta paketu, ako prišli zo vstupu.
- **Transformovaný paket (transformed_packet):** výsledný paket po aplikovaní transformačných pravidiel (ak je aktívna transformácia).

2.2.3. Údaje o používateľských a automatických akciách/udalostiach

Každá akcia je logovaná s nasledujúcimi atribútmi:

- **Čas akcie (action_timestamp):** čas, kedy bola akcia vykonaná.
- **Typ akcie (action_type):** popis akcie (napr. zmena módu, reset systému, nová konštanta z kamery, nová nakalibrovaná hodnota pre určitú deklináciu a pod.).
- **Parametre akcie (action_parameters):** prípadné dodatočné informácie o akcii, ako napr. hodnota nastavenej konštanty.

2.2.4. Automaticky kalibrované konštanty

- Tabuľka s poslednou najlepšou platnou verziou nájdených automatických konštánt

2.3. Formát súborov pre transformačný modul pre teleskop

Pre komunikáciu a ukladanie dát systém využíva nasledujúce formáty súborov

2.3.1. Konfiguračný súbor

- **Formát:** TXT.
- **Obsah:** Obsahuje konfiguračné údaje, ktoré systém načíta pri štarte, vrátane nastavení pre transformáciu, bypass mód, a konfigurácie transformačných modulov.

2.3.2. Debug-Log súbor

- **Formát:** TXT.
- **Obsah:** Tento log uchováva informácie o vykonaných operáciách systému, ako je nastavenie rýchlosti motorov, aplikované pravidlá pre transformáciu, a akékoľvek chyby alebo výnimky počas behu systému.
- **Štruktúra logu:**
 - Časová pečiatka, Úroveň závažnosti, Správa, Kód chyby

2.3.3. Paket-Log súbor

- **Formát:** TXT.
- **Obsah:** Logy obsahujú zaznamenané údaje o prichádzajúcich a odchádzajúcich paketov, časové pečiatky, typ paketu, záznam o vykonaných transformáciách. Binárne pakety sú logované v ASCII podobe, t.j. bajt s hodnotou 0x14 sa zobrazí ako textový reťazec „14“
- **Štruktúra logu:**
 - Časová pečiatka, Čas zdržania, Originálny paket, Transformovaný paket

2.4. Komunikačné protokoly

Pre komunikáciu v master-slave architektúre systému je potrebný špecifický protokol, ktorý zahŕňa príkazy pre riadenie teleskopu a dynamickú úpravu rýchlosti motorov.

2.4.1. Master-Slave Komunikácia

- **Formát telegramu:** každý telegram obsahuje hlavičku, adresu, príkaz, a checksum.
- **Hlavička (header):** identifikuje začiatok telegramu.

- **Adresa (address):** špecifikuje, či ide o príkaz pre os RA alebo DEC.
- **Príkaz (command):** binárna hodnota určujúca typ akcie (napr. zvýšenie/zníženie rýchlosti).
- **Dátová sekcia (data):** informácie o požadovanej rýchlosti, oneskorení, prípadne ďalších parametroch pre transformáciu.
- **Checksum:** zabezpečuje správnosť telegramu.

2.4.2. API pre Autoguider

Externý softvér komunikuje s transformačným zariadením cez špeciálne definované API. Toto API obsahuje príkazy pre:

- **Dynamickú zmenu konštanty zrýchlenia/spomalenia:**
 - **API príkaz:** SET_ACCEL_CONST
 - **Popis:** Tento príkaz umožňuje zmenu konštanty zrýchlenia alebo spomalenia zariadenia. Hodnota konštanty je zadávaná ako parameter v príkaze a odosiela sa prostredníctvom HTTP požiadavky na API zariadenia.
 - **Formát:**
http://IP_ADRESA/camera_api?passwd=HESLO&accel_const=HODN OTA
 - passwd=HESLO – Heslo pre autentifikáciu prístupu k API.
 - accel_const=HODNOTA – Nová hodnota konštanty zrýchlenia/spomalenia, ktorá bude nastavená na zariadení.
- **Reset zariadenia:**
 - **API príkaz:** RESET
 - **Popis:** Tento príkaz slúži na resetovanie zariadenia, čím sa zariadenie vráti do pôvodného stavu.
 - **Formát:** http://IP_ADRESA/camera_api?passwd=HESLO&reset=1
 - passwd=HESLO – Heslo pre autentifikáciu prístupu k API.
 - reset=1 - indikuje, že zariadenie má byť resetované. Hodnota 1 znamená, že reset bude vykonaný.

2.5. Dátový model perzistentných údajov pre univerzálny transformačný modul

2.5.1. Konfiguračné údaje pre univerzálny modul

Univerzálny modul vyžaduje dodatočné konfiguračné údaje pre pravidlá transformácie:

- **Pravidlá transformácie:** Pole pravidiel, kde každé pravidlo obsahuje:
 - **ID pravidla:** Unikátne identifikátor pravidla.
 - **Vzor (pattern):** Regulárny výraz, ktorý opisuje, aké pakety sa majú transformovať.
 - **Náhrada (replacement):** Text alebo vzor, ktorým sa nahradí zodpovedajúca časť paketu.
 - **Aktivácia pravidla:** Binárna hodnota určujúca, či je pravidlo aktívne (1 = aktívne, 0 = neaktívne).
- **Poradie pravidiel:** Definuje prioritu aplikácie pravidiel, ak ich je viacero.

2.5.2. Údaje o zaznamenaných paketoch pre univerzálny modul

Rovnaké ako pre teleskopický modul:

- **Originálny paket:** Surové dáta paketu.
- **Transformovaný paket:** Paket po aplikácii pravidiel univerzálneho modulu.
- **Aplikované pravidlá:** Zoznam ID pravidiel, ktoré boli aplikované na konkrétny paket.

2.5.3. Logovanie pre univerzálny modul

Rozšírenie logovania pre univerzálny modul:

- **ID aplikovaného pravidla:** Identifikátor pravidla, ktoré bolo použité na transformáciu.

2.6. Formát súborov pre univerzálny transformačný modul

2.6.1. Konfiguračný súbor pre univerzálny modul

- **Formát:** TXT.
- **Obsah:** Obsahuje konfiguračné údaje, ktoré systém načíta pri štarte, vrátane nastavení pre transformáciu, bypass mód a konfigurácie pravidiel transformácie.

2.6.2. Debug-Log súbor pre univerzálny modul

- **Formát:** TXT.

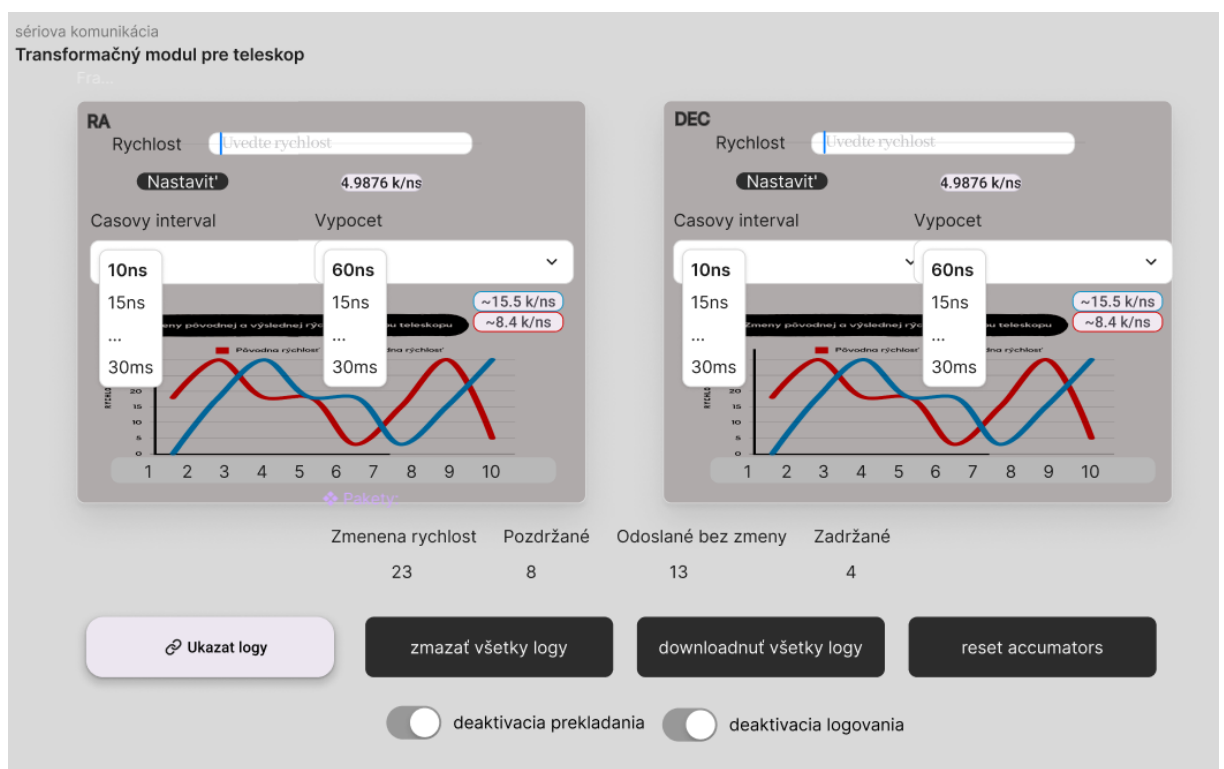
- **Obsah:** Tento log uchováva informácie o vykonaných operáciách systému, ako je aplikovanie nových pravidiel regulárnych výrazov, zmeny konfigurácie, a akékoľvek chyby alebo výnimky počas behu systému.
- **Štruktúra logu:**
 - Časová pečiatka, Úroveň závažnosti, Správa, Kód chyby

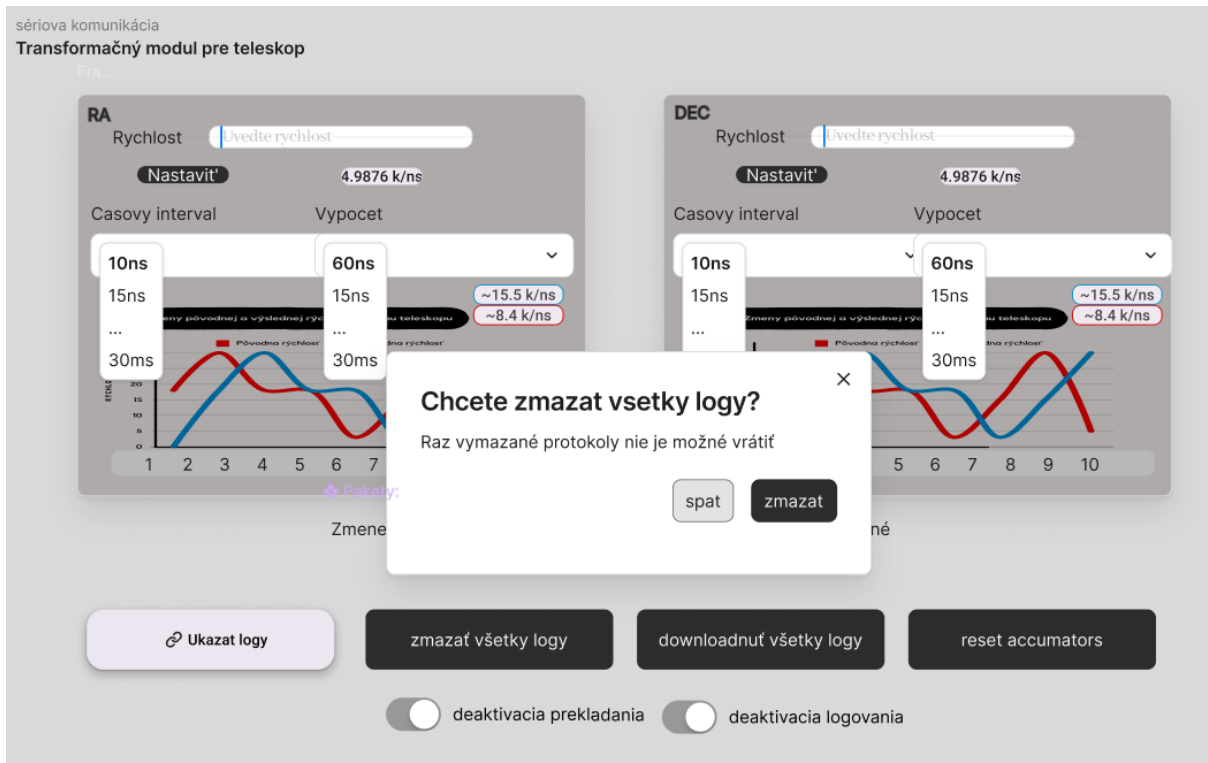
2.6.3. Paket-Log súbor pre univerzálny modul

- **Formát:** TXT.
- **Obsah:** Logy obsahujú zaznamenané údaje o prichádzajúcich a odchádzajúcich paktoch, časové pečiatky, typ paketu, záznam o vykonaných transformáciách.
- **Štruktúra logu:**
- Časová pečiatka, Čas zdržania, Originálny paket, Transformovaný paket, Aplikované regex pravidlo .

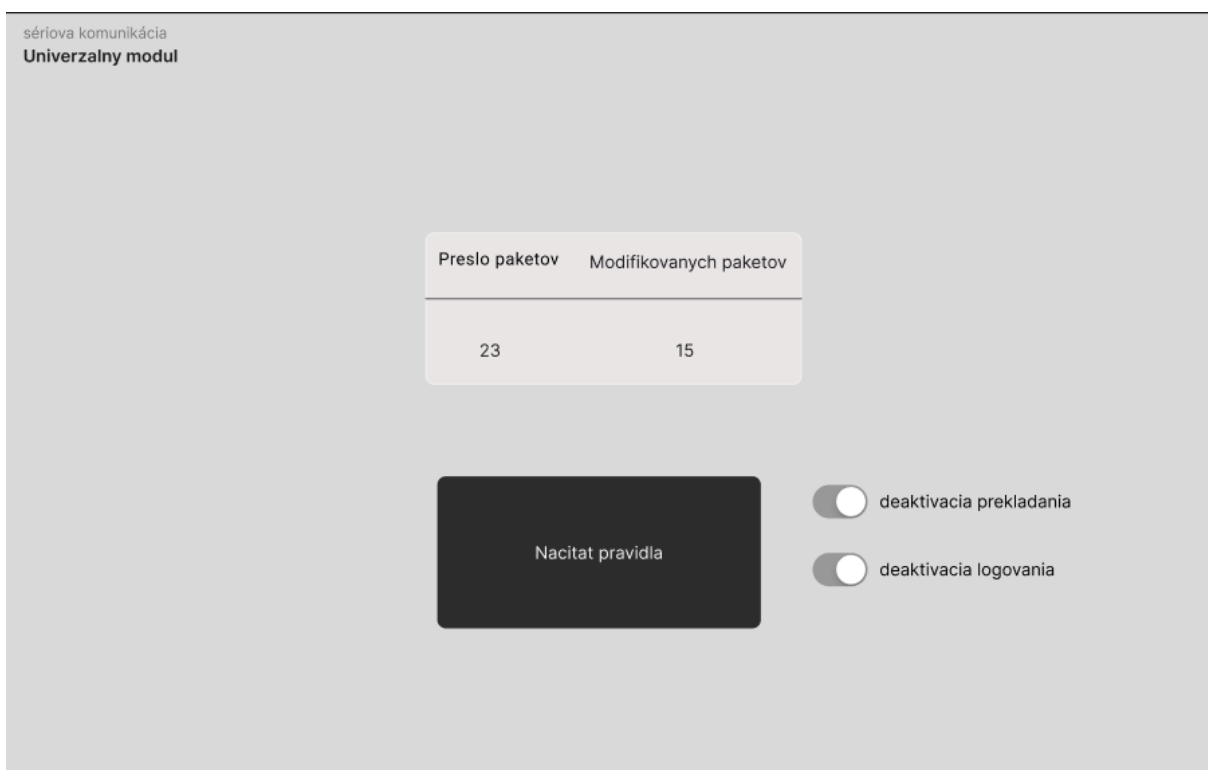
2.7. Návrh používateľského rozhrania

2.7.1. Pre transformačný modul pre teleskop



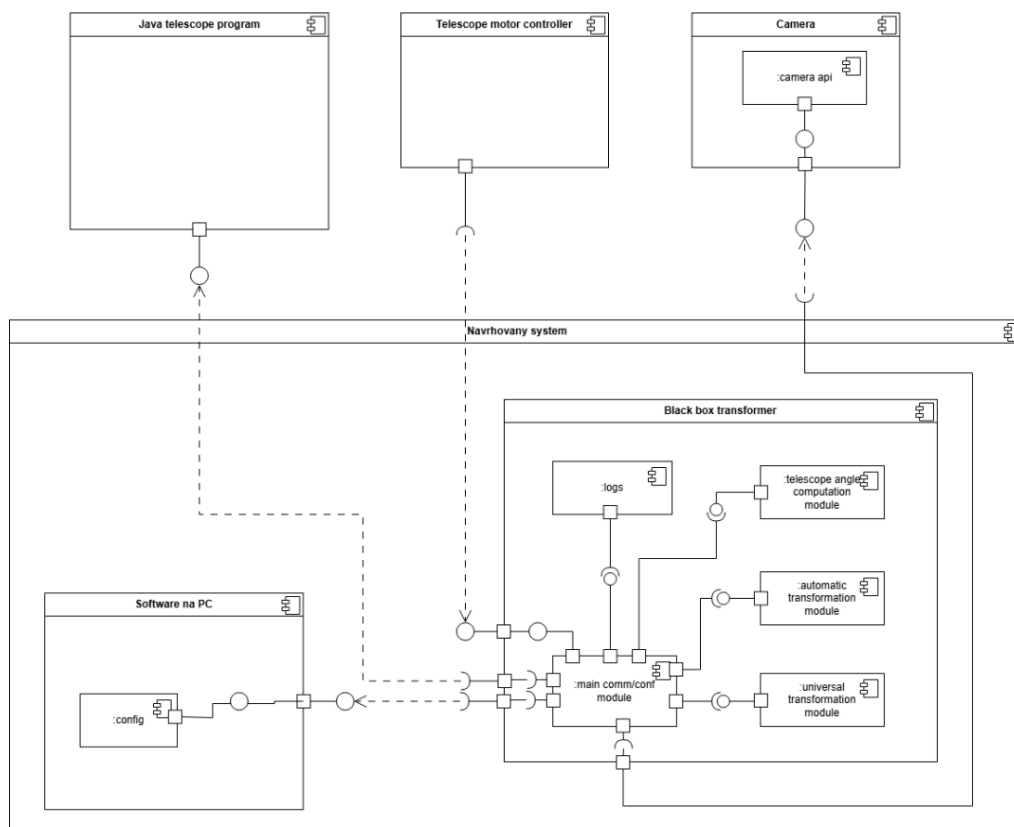


2.7.2. Pre univerzálny transformačný modul



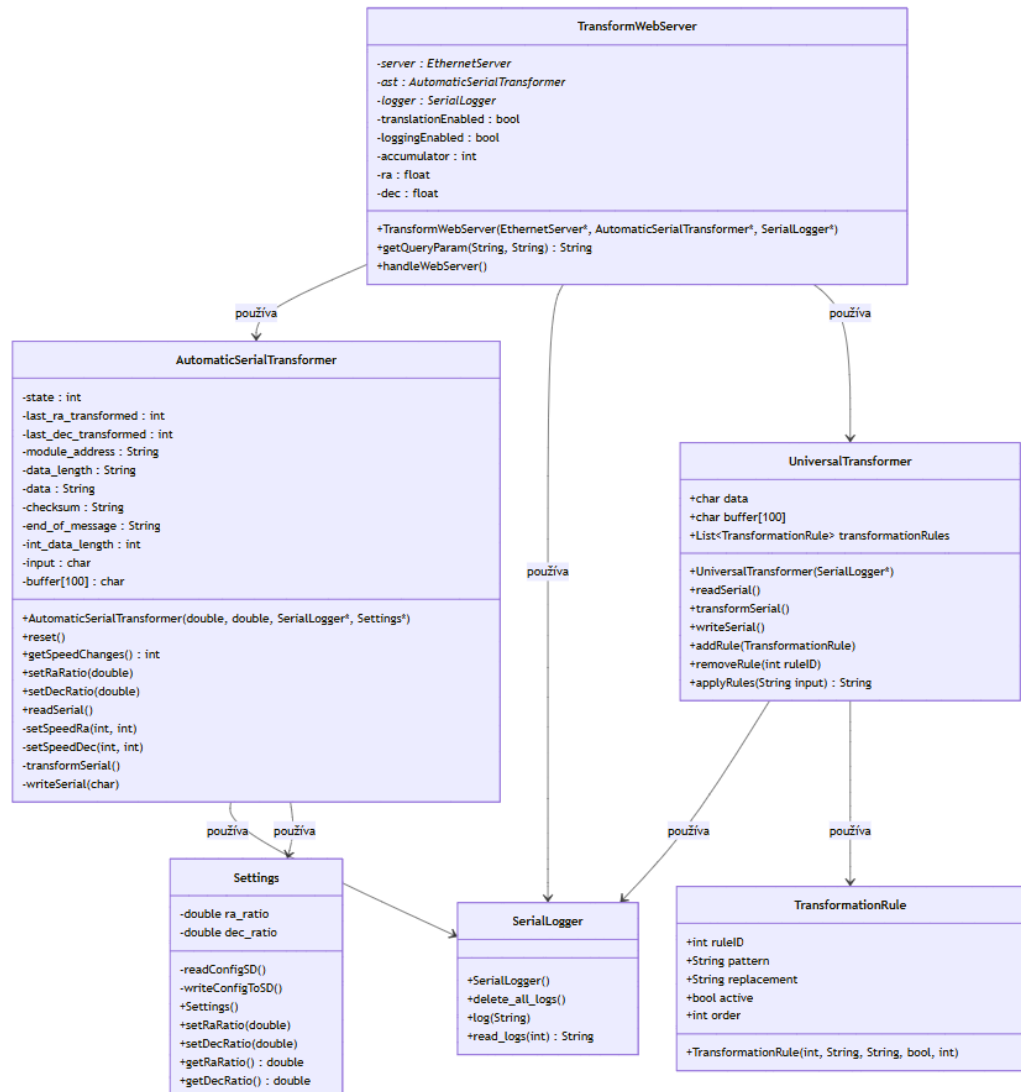
2.8. Návrh implementácie

2.8.1. Diagram komponentov



- Java telescope program - externý softvérový modul, ktorý komunikuje so systémom.
- Telescope motor controller - riadi pohyb teleskopu na základe prijatých príkazov.
- Camera - obsahuje modul, ktorý umožňuje komunikáciu s navrhovaným systémom. Používa sa na snímání obrazu z teleskopu.
- Software na PC - slúži na nastavovanie a správu.
- Black box transformer - obsahuje niekoľko modulov, ktoré vykonávajú rôzne výpočty a transformácie: telescope angle computation module, telescope angle computation module, universal transformation module, logs, main comm/conf module

2.8.2. Triedny diagram



- **TransformWebServer** – Poskytuje webové rozhranie na komunikáciu so systémom. Využíva: **UniversalTransformer** na univerzálne transformácie dát, **SerialLogger** na logovanie.
- **AutomaticSerialTransformer** – Spracováva a transformuje sériové dáta. Využíva: **SerialLogger** na záznam udalostí, **Settings** na čítanie a ukladanie konfigurácie.
- **UniversalTransformer** – Poskytuje univerzálnu transformáciu dát, pričom: Využíva **SerialLogger** na logovanie, Uchováva a spracováva zoznam pravidiel **TransformationRule**
- **TransformationRule** – Definuje jednotlivé pravidlá, ktoré **UniversalTransformer** aplikuje na dáta

- Settings – Uchováva konfiguračné údaje a dokáže ich načítať alebo uložiť na SD kartu.
- SerialLogger – Zabezpečuje logovanie (zápis do súboru, čítanie logov, mazanie logov) a využíva sa všade tam, kde je potrebné zaznamenávať udalosti (AutomaticSerialTransformer, UniversalTransformer, TransformWebServer).

2.8.3. Rozdelenie na časti (moduly) a ich interfejsy

- ```
class TransformWebServer {
public:
 // Konštruktor
 TransformWebServer(EthernetServer* ptr_server,
AutomaticSerialTransformer* ptr_ast, SerialLogger* ptr_logger) {
 }

 // Extrahuje parameter z HTTP requestu
 String getQueryParam(String request, String param) { }

 // Spracováva prichádzajúce HTTP požiadavky
 void handleWebServer() { }
};
```

- ```
class UniversalTransformer {
public:
    // Konštruktor
    UniversalTransformer() { }

    // Číta dáta zo sériového portu
    void readSerial() { }

    // Vykoná transformáciu dát
    void transformSerial() { }

    // Zapíše dáta na sériový port
    void writeSerial() { }
};
```

- ```
class Settings {
public:
 // Konštruktor - načíta
 konfiguráciu zo SD karty
 Settings() { }
 // Nastaví a uloží RA pomer
 void setRaRatio(double ra) { }
 // Nastaví a uloží DEC pomer
 void setDecRatio(double dec) { }
 // Vráti aktuálny RA pomer
 double getRaRatio() { }
 // Vráti aktuálny DEC pomer
 double getDecRatio() { }
};
```

- ```
class SerialLogger {  
public:  
    // Konštruktor  
    SerialLogger() { }  
    // Vymaže všetky logy  
    void delete_all_logs() { }  
    // Zapíše správu do logu  
    void log(String message) { }  
    // Vráti obsah logu  
    String read_logs(int amount) { }  
};
```



```
class AutomaticSerialTransformer {  
public:  
    // Konštruktor  
    AutomaticSerialTransformer(double ra, double dec, SerialLogger*  
ptr_logger, Settings* ptr_settings) { }  
    // Resetuje interné premenné  
    void reset() { }  
    // Vráti počet zmien rýchlosti  
    int getSpeedChanges() { }  
    // Nastaví RA pomer a aktualizuje logy  
    void setRaRatio(double ra) { }  
    // Nastaví DEC pomer a aktualizuje logy  
    void setDecRatio(double dec) { }  
    // Číta dáta zo sériového portu  
    void readSerial() { }  
};
```

2.8.4. Využité technológie

- **Arduino/C++ framework** – pre programovanie mikrokontrolérov, kde sa využíva jazyk C++ a knižnice ako Arduino.h, ktoré umožňujú prácu so sériovou komunikáciou, časom, a pod.
- **Teensy 4.1** – ako hlavný mikrokontrolér, ktorý poskytuje dostatočný výkon a periférie.
- **Serial komunikácia** – na odosielanie a prijímanie dát medzi modulmi.
- **Ethernet a QNEthernet knižnica** – pre implementáciu webového servera, ktorý spracováva HTTP požiadavky a umožňuje interakciu so systémom cez webové rozhranie.
- **SPI a SD knižnica** – na prácu so SD kartou, ktorá slúži na ukladanie konfiguračných údajov a logov.

2.9. Plán implementácie

1. Fáza – Základné moduly a infraštruktúra
 - Najprv sa implementujú základné triedy, ako napríklad SerialLogger a Settings, ktoré sú kľúčové pre logovanie a správu konfigurácie.
2. Fáza – Implementácia transformačných modulov
 - V ďalšej fáze sa vyvíjajú transformačné moduly – AutomaticSerialTransformer a UniversalTransformer.
3. Fáza – Vývoj webového rozhrania
 - Po overení funkčnosti základných a transformačných modulov sa implementuje TransformWebServer, ktorý poskytuje komunikáciu cez HTTP/HTTPS a zabezpečuje integráciu s externými zariadeniami, ako je autoguider či PC softvér.
4. Fáza – Integrácia s externými komponentmi
 - Nasleduje integrácia s externými systémami, ako sú Java telescope program, Telescope motor controller a kamera. Cieľom je zabezpečiť bezproblémovú komunikáciu medzi všetkými modulmi a overiť, že rozhrania medzi komponentmi fungujú korektne.
5. Fáza – Systémové testovanie a ladenie
 - V záverečnej fáze prebehne celková integrácia systému. Systém bude testovaný v reálnych scenároch, čo umožní identifikovať a odstrániť prípadné chyby, optimalizovať výkon a zabezpečiť finálnu spoľahlivosť celého riešenia.

3. Testovacie scenáre

3.1. Modul teleskop

1. akcia: používateľ nastaví mód na Automatický [Nastavenie módu – rozbaľovacia ponuka]

výsledok: v rozhraní bude v rozbaľovacej ponuke pre nastavenie módu vybrané Automatický a teleskop sa bude pohybovať a upravovať rýchlosť automaticky, teleskop nebude reagovať na nastavenie manuálnej konštanty pre zmenu rýchlosti

2. akcia: používateľ nastaví mód na Manuálny [Nastavenie módu – rozbaľovacia ponuka]

výsledok: v rozhraní bude v rozbaľovacej ponuke pre nastavenie módu vybrané Manuálny a teleskop sa bude pohybovať podľa aktuálne nastavenej konštanty

3. akcia: používateľ zadá konštantu zrýchlenia a stlačí tlačidlo [Nastaviť]

výsledok: v rozhraní sa zmení ukazovateľ aktuálne nastavenej rýchlosti a teleskop sa začne pohybovať podľa novej konštanty

4. akcia: používateľ zmení konštantu časového intervalu priemerovania pre graf [Časový interval – rozbaľovacia ponuka]

výsledok: v rozhraní sa bude v rozbaľovacej ponuke pre nastavenie časového intervalu priemerovanie pre graf zobrazovať zvolená konštanta, priebeh grafu sa bude zobrazovať a priemerovať podľa nej

5. akcia: používateľ zmení konstantu časového intervalu výpočtu pre automatickú korekciu [Výpočet – rozbaľovacia ponuka]

výsledok: v rozhraní sa bude v rozbaľovacej ponuke pre nastavenie časového intervalu výpočtu pre automatickú korekciu zobrazovať zvolená konštanta, automatická korekcia sa bude vypočítavať v intervale podľa zvolenej konstanty

6. akcia: používateľ klikne na tlačidlo [Ukázať logy]

výsledok: v rozhraní sa zobrazí nové okno, v ktorom sú zobrazené zaznamenané logy

7. akcia: používateľ klikne na tlačidlo [Zmazať všetky logy]

výsledok: v rozhraní sa zobrazí vyskakovacie okno pre potvrdenie zmazania všetkých logov s možnosťou ich zmazať alebo ponechať

8. akcia: používateľ klikne na tlačidlo [Downloadnuť všetky logy]

výsledok: v rozhraní sa zobrazí vyskakovacie okno na výber miesta uloženia súboru (alebo sa logy automaticky stiahnu do predvoleného adresára), súbor bude vo formáte .txt obsahujúci všetky aktuálne zaznamenané logy

9. akcia: používateľ klikne na tlačidlo [Reset accumators]

výsledok: v rozhraní sa zobrazí vyskakovacie okno pre potvrdenie resetovania všetkých akumulovaných hodnôt s možnosťou ich resetovať alebo ponechať

10. akcia: používateľ klikne na prepínač (zapne) [Deaktivácia prekladania]

výsledok: v rozhraní sa zmení farba prepínača (indikátor pre zapnuté/vypnuté), pakety sa budú preposielať bez úprav

11. akcia: používateľ klikne na prepínač (zapne) [Deaktivácia logovania]

výsledok: v rozhraní sa zmení farba prepínača (indikátor pre zapnuté/vypnuté), logovanie sa zastaví, existujúce logy zostanú zachované

3.2. Modul universal

1. akcia: používateľ klikne na tlačidlo [Nastaviť pravidla]

výsledok: zobrazí sa okno alebo sekcia pre nastavenie pravidiel, kde používateľ môže zadať zoznam regulárnych výrazov a transformácie pre každú kategóriu.

2. akcia: používateľ klikne na prepínač (zapne) [Deaktivácia prekladania]

výsledok: v rozhraní sa zmení farba prepínača (indikátor pre zapnuté/vypnuté), pakety sa budú preposielať bez úprav

3. akcia: používateľ klikne na prepínač (zapne) [Deaktivácia logovania]

výsledok: v rozhraní sa zmení farba prepínača (indikátor pre zapnuté/vypnuté), logovanie sa zastaví, existujúce logy zostanú zachované

4. akcia: používateľ zadá regulárny výraz **^Hello.*\$** pre ASCII paket.

výsledok: modul úspešne identifikuje všetky pakety, ktoré začínajú textom **Hello**.

5. akcia: používateľ pridá transformáciu **Hello** za **Hi** na danú kategóriu.

výsledok: pakety kategórie definovanej výrazom **^Hello.*\$** budú transformované tak, že každé **Hello** bude nahradené za **Hi**.

6. akcia: používateľ zadá regulárny výraz **\x01\x02.*\x03** na binárny paket s HEXA hodnotou.

výsledok: modul úspešne identifikuje všetky binárne pakety, ktoré začínajú sekvenciou **01 02** a obsahujú ľubovoľné dáta ukončené **03**.

7. akcia: používateľ pridá transformáciu **\x01\x02** za **\xFF\xFF** na binárny paket.

výsledok: pakety, ktoré začínajú **01 02**, budú transformované tak, že úvodná sekvencia bude nahradená **FF FF**.

8. akcia: používateľ zadá regulárne výrazy s chybnou syntaxou.

výsledok: modul zistí chybnú syntax, upozorní používateľa a nedovolí uložiť pravidlo.