

ŽILINSKÁ UNIVERZITA
V ŽILINE

Fakulta elektrotechniky a informačných technológií

Dispečerské pracovisko modelovej železnice

Diplomová práca

Bc. Dávid Jozef Macko

Študijný program: Riadenie procesov

Študijný odbor: Kybernetika

Školiace pracovisko: Žilinská univerzita v Žiline

Vedúci diplomovej práce: Ing. Peter Nagy, PhD.

Žilina 2024



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta elektrotechniky
a informačných technológií
Katedra radiacích
a informačných systémov

Akademický rok 2023/2024

Evidenčné číslo 28260220242019

ZADANIE DIPLOMOVEJ PRÁCE

Meno študenta: **Bc. Dávid Jozef Macko**

Študijný program: **Riadenie procesov**

Téma diplomovej práce: **Dispečerské pracovisko modelovej železnice**

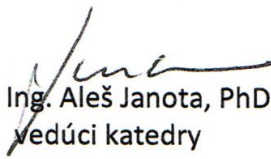
Pokyny pre vypracovanie diplomovej práce:

1. Analýza súčasného stavu riadenia modelovej železnice v laboratóriu KRIS AB 205.
2. Špecifikácia funkčných a technických požiadaviek na dispečerské pracovisko.
3. Návrh algoritmov riadiaceho systému modelového koľajiska a dispečerského pracoviska.
4. Programová implementácia algoritmu vo zvolenom programovom prostredí.
5. Overenie funkčnosti algoritmu a zhodnotenie dosiahnutých výsledkov.

Vedúci diplomovej práce : **Ing. Peter Nagy, PhD.**

Dátum odovzdania diplomovej práce : **30.04.2024**

V Žiline dňa 30.10.2023


prof. Ing. Aleš Janota, PhD.
vedúci katedry

Čestné vyhlásenie

Vyhlasujem, že som zadanú diplomovú prácu vypracoval samostatne, pod odborným vedením vedúceho práce/školiťa a používal som len literatúru uvedenú v práci.

Žilina, 30. apríla 2024

Pod'akovanie

Chcel by som sa poďakovať môjmu školiteľovi, pánovi Ing. Petrovi Nagyovi, PhD. a kolegovi Bc. Jakubovi Krškovi za ich cenné rady, pomoc a pripomienky pri vypracovaní diplomovej práce.

Abstrakt

Cieľom tejto diplomovej práce je návrh a implementácia dispečerského pracoviska pre modelovú železniciu Katedry riadiacich a informačných systémov, Fakulty elektrotechniky a informačných technológií, Žilinskej univerzity v Žiline. Práca sa zaoberá analýzou súčasného stavu koľajiska a implementovaných zabezpečovacích systémov. V úvode práce sú vypracované funkčné a technické požiadavky, na nimi zvolený modelovaný systém dispečerského riadenia. Rozhodli sme sa pre implementáciu modelu systému SIMIS W spoločnosti Siemens Mobility, spolu s obslužným pracoviskom ILTIS-N. Hlavná časť diplomovej práce sa zaoberá návrhom algoritmov riadiaceho systému modelového koľajiska a riadiaceho dispečerského pracoviska. Práca obsahuje aj opis programovej implementácie algoritmov v prostredí Qt Creator, s využitím jazyka Python. V diplomovej práci je obsiahnutý aj opis návrhu komunikačného rozhrania, vytvoreného systému SIMIS W s ďalším systémom ESA-44, už implementovaným na vedľajšej trati modelovej železnice. V závere práce sú zhodnotené výsledky implementácie nového modelu dispečerského pracoviska na modelovej železnici. Modelová železnica by mala slúžiť ako výučbová pomôcka pri štúdiu železničných zabezpečovacích systémov a pre prezentačné účely katedry.

Kľúčové slová: modelová železnica, dispečerské pracovisko, zabezpečovací systém, SIMIS W, ILTIS-N

Abstract

The main aim of this diploma thesis is the design and implementation of a dispatching workstation for a model railway of the Department of Control and Information Systems, Faculty of Electrical Engineering and Information Technology, University of Žilina. The diploma thesis deals with the analysis of the current state of the model railway, and the implemented signalling systems. In the beginning of the thesis, functional and technical requirements for the modelled dispatching control system are elaborated. We have decided to implement the model of the SIMIS W system of Siemens Mobility company, together with the ILTIS-N operator workplace. The main part of the thesis deals with the design of the algorithms of the model railway control system and the control dispatching workstation. The thesis also includes a description of the software implementation of the algorithms in the Qt Creator environment, using the Python programming language. The thesis also includes a description of the design of the communication interface between SIMIS W system and ESA-44 system, already implemented on the model railway. The thesis concludes with an evaluation of the results of the implementation of the new dispatching workstation model on the model railway. The model railway should serve as a teaching aid in the study of railway signalling systems and for presentation purposes of the department.

Keywords: model railway, dispatch system, interlocking system, SIMIS W, ILTIS-N

Obsah

Úvod.....	1
1 Analýza súčasného stavu.....	2
1.1 Trať č.1 Nitra - Chynorany.....	2
1.2 Trať č.2 Radošina - Hlohovec	5
1.2.1 Riadiaci systém.....	7
1.2.2 Hardvérové komponenty.....	7
1.2.3 Doplnenie hardvérových prvkov koľajiska.....	10
1.2.4 Staničné zabezpečovacie zariadenia	11
1.2.5 Traťové zabezpečovacie zariadenia	11
1.2.6 Priecestné zabezpečovacie zariadenia	12
1.2.7 Spôsob riadenia.....	12
1.2.8 Unifikácia riadiacich systémov.....	14
2 Špecifikácia požiadaviek	14
2.1 Požiadavky na vizualizáciu dopravných procesov	14
2.2 Požiadavky na staničné zabezpečovacie systémy	15
2.3 Požiadavky na traťové zabezpečovacie systémy	15
2.4 Požiadavky na priecestné zabezpečovacie systémy.....	16
2.5 Komunikácia modelov ESA-44 a SIMIS W	17
3 Elektronické stavadlo SIMIS W	17
3.1 Riadiaca úroveň.....	18
3.2 Obslužná úroveň.....	19
3.3 Úroveň centrálnych funkcií stavadla	19
3.4 Úroveň riadenia kontroly vonkajších prvkov	20
3.5 Štruktúra softvéru	20
3.6 Princíp činnosti	21
4 Návrh konceptu riadiaceho systému	22
4.1 Koncept riadenia vonkajších prvkov.....	22
4.2 Návrh konceptu komunikačného protokolu.....	22
4.2.1 Návrhy komunikácie	23
4.3 Návrh a realizácia komunikačného protokolu	25
5 Hardvérové komponenty riadiaceho systému.....	26
6 Návrh grafickej aplikácie.....	28
6.1 Jazyk Python	28

6.2	Symboly prvkov	28
6.3	Pracoviská lokálnej a vzdialenej obsluhy	29
7	Programová implementácia funkčných algoritmov	30
7.1	REST API rozhranie	32
7.1.1	Štruktúra zdrojových kódov	33
7.1.2	Funkcie pre zápis do PLC	34
7.1.3	Funkcia pre načítanie dát do PC aplikácie	34
7.1.4	Funkcie pre prácu so zdieľanými dátami	35
7.1.5	Funkcie pre systém ESA-44	36
7.1.6	Arbitráž komunikácie REST API rozhrania s PLC	36
7.2	Traťové zabezpečovacie zariadenia	37
7.3	Priecestné zabezpečovacie zariadenia	39
7.4	Staničné zabezpečovacie zariadenia	40
7.4.1	Funkcia „stavanieCesty“	41
7.4.2	Funkcia „rusenieCesty“	42
7.4.3	Funkcia „rusenieOD“	43
7.4.4	Funkcia „prestavenieVyh“	43
7.4.5	Záverová tabuľka	43
7.5	Riadenie vonkajších prvkov v koľajisku	44
7.5.1	Funkčné bloky riadenia staničných návěstidiel	45
7.5.2	Funkčný blok výmeny	46
7.5.3	Funkčný blok koľajových obvodov	46
7.6	Algoritmy grafického rozhrania obsluhy	47
7.6.1	Súbory zdrojového kódu	48
7.6.2	Väzba medzi grafickým oknom a metódami	52
7.6.3	Viacvláknové spracovanie údajov	53
8	Úprava riadenia trate s modelom ESA-44	54
8.1	Zmena topológie koľajiska	54
8.2	Úprava hardvérovej časti riadiaceho systému	56
8.3	Parametrizácia digitálnych dekodérov	58
8.4	Montáž digitálneho systému riadenia Lenz	59
9	Zhodnotenie výsledkov	60
	Záver	61

Zoznam obrázkov

Obr. 1.1 Schematické znázornenie modelového koľajiska [2].....	2
Obr. 1.2 Situačná schéma stanice Nitra.....	3
Obr. 1.3 Situačná schéma trate Nitra - Lužianky, časť 1	3
Obr. 1.4 Situačná schéma trate Nitra - Lužianky, časť 2	3
Obr. 1.5 Situačná schéma stanice Lužianky, časť 1	4
Obr. 1.6 Situačná schéma stanice Lužianky, časť 2.....	4
Obr. 1.7 Situačná schéma trate Lužianky - Topoľčany.....	4
Obr. 1.8 Situačná schéma výhybne Topoľčany	4
Obr. 1.9 Situačná schéma traťového úseku Topoľčany – Chynorany.....	5
Obr. 1.10 Situačná schéma stanice Chynorany.....	5
Obr. 1.11 Situačná schéma stanice Radošina.....	6
Obr. 1.12 Situačná schéma stanice Zbehy	6
Obr. 1.13 Situačná schéma stanice Hlohovec	6
Obr. 1.14 Situačná schéma trate Radošina - Zbehy	6
Obr. 1.15 Situačná schéma trate Lužianky – Zbehy	6
Obr. 1.16 Situačná schéma trate Zbehy – Hlohovec	7
Obr. 1.17 Prúdový modul Lenz LB 101 [13].....	7
Obr. 1.18 Schéma rozhrania trať - vstup PLC [2]	8
Obr. 1.19 Prispôsobenie výstupu PLC zapojeniu návěstidla [2].....	8
Obr. 1.20 Elektromotorický prestavník Tillig [14]	9
Obr. 1.21 Schéma pripojenia prestavníka k PLC [2].....	9
Obr. 1.22 Schéma pripojenia výstražníka [3]	10
Obr. 1.23 Schéma zapojenia prestavníka Tillig	11
Obr. 1.24 Vizualizácia pre traťové zabezpečovacie zariadenia [2].....	13
Obr. 1.25 Vizualizácia pre priecestné zabezpečovacie zariadenia	13
Obr. 3.1 Štruktúra stavadla SIMIS W s obslužným pracoviskom [1].....	18
Obr. 3.2 Príklad obslužného pracoviska ILTIS-N v stanici Veľké Kostoľany [11].....	18
Obr. 3.3 Počítač SIMIS ECC použitý v ACC komponente [12].....	20
Obr. 3.4 Delenie softvéru stavadla SIMIS W.....	21
Obr. 4.1 Schéma pripojenia dekodérov LS-100 k PLC S7-1200	23
Obr. 4.2 Schéma pripojenia modulu LR-101 k PLC S7-1200.....	24
Obr. 4.3 Príklad URL adresy	26
Obr. 5.1 Architektúra hardvérovej časti riadiaceho systému	27
Obr. 6.1 Ukážka zobrazenia symbolov vybraných prevádzkových stavov výhybky	29

Obr. 6.2 Vizualizácia prevádzkových stavov trojznakového automatického bloku	29
Obr. 7.1 Architektúra softvérovej časti riadiaceho systému	31
Obr. 7.2 Znárodnenie sieťovej komunikácie	32
Obr. 7.3 Funkcia "writePredhl" s arbitrážnymi algoritmi	37
Obr. 7.4 Algoritmus automatického bloku	38
Obr. 7.5 Stavový automat automatického hradla	39
Obr. 7.6 Stavový diagram činnosti priecestia	40
Obr. 7.7 Zápis jazdnej cesty v stanici Radošina	44
Obr. 7.8 Stavový automat funkčného bloku výmeny	46
Obr. 7.9 Príklad jednej sady z funkčného bloku "Tratove_useky"	47
Obr. 7.10 Príklad zoznamu „enum“ pre symbol zriaďovacieho návěstidla	50
Obr. 7.11 Reprezentácia metódy „rusenieCesty“ objektu „Naveštidlo“	51
Obr. 7.12 Príklad prepojenia grafického okna s užívateľskými metódami	52
Obr. 7.13 Porovnanie jednovláknového a viacvláknového spracovania	53
Obr. 8.1 Pôvodný stav medzistaničného úseku Topoľčany – Chynorany [10]	55
Obr. 8.2 Nový stav medzistaničného úseku Topoľčany - Chynorany	55
Obr. 8.3 Pôvodný stav medzistaničného úseku Nitra – Lužianky, časť 1 [15]	55
Obr. 8.4 Pôvodný stav medzistaničného úseku Nitra – Lužianky, časť 2 [15]	55
Obr. 8.5 Nový stav medzistaničného úseku Nitra – Lužianky, časť 1	56
Obr. 8.6 Nový stav medzistaničného úseku Nitra – Lužianky, časť 2	56
Obr. 8.7 Spôsob spájania vybraných LED hlavného návěstidla a predzvesti	57
Obr. 8.8 Situačná schéma pravého zhlavia stanice Lužianky	58
Obr. 9.1 Schéma zapojenia modulu priecestia [3]	67
Obr. 9.2 Schéma pripojenia LED návěstidiel L2, L3 v stn. Nitra	68
Obr. 9.3 Schéma pripojenia LED návěstidiel AB24, AB49	68
Obr. 9.4 Schéma pripojenia LED návěstidla L v stn. Lužianky	68
Obr. 9.5 Schéma pripojenia LED návěstidiel S1, S3, S5 v stn. Lužianky	69
Obr. 9.6 Schéma pripojenia LED návěstidla S2 v stn. Lužianky	69
Obr. 9.7 Schéma pripojenia LED návěstidiel L1, L3 v stn. Lužianky	69
Obr. 9.8 Schéma LED návěstidiel FS, S v stn. Lužianky a L, S vo výh. Topoľčany	70
Obr. 9.9 Schéma pripojenia LED návěstidiel L/PrL v stn. Chynorany	70
Obr. 9.10 Schéma pripojenia LED návěstidla S1 v stn. Chynorany	71
Obr. 9.11 Schéma LED návěstidiel L2, L5 v stn. Lužianky a S2, S3 v stn. Chynorany	71
Obr. 9.12 Príklad pripojenia oddielového návěstidla AB3	71
Obr. 9.13 Komunikačná vrstva PLC	72
Obr. 9.14 Architektúra PLC SIMATIC S7-1200	72

Obr. 9.15 Architektúra vzdialeného ostrovu vstupných a výstupných modulov	73
Obr. 9.16 Architektúra PLC SIMATIC S7-300.....	73
Obr. 9.17 Hlavná obrazovka PC aplikácie vzdialenej obsluhy	76
Obr. 9.18 Hlavná obrazovka PC aplikácie lokálnej obsluhy	77
Obr. 9.19 Zdrojový kód algoritmu riadenia PZZ	79
Obr. 9.20 Zdrojový kód bloku detekcie smeru jazdy koľajového vozidla.....	80
Obr. 9.21 Rozhodovací strom návěstidla S zo stanice Zbehy	81
Obr. 9.22 Príklad rozhodovacieho stromu návěstidla S2 zo stanice Zbehy	82
Obr. 9.23 Sekvenčný diagram automatického hradla	83
Obr. 9.24 Vývojový diagram funkcie „ <i>stavanieCesty</i> “	84
Obr. 9.25 Vývojový diagram funkcie „ <i>rusenieCesty</i> “	85
Obr. 9.26 Vývojový diagram funkcie „ <i>rusenieOD</i> “	86
Obr. 9.27 Vývojový diagram funkcie „ <i>prestavenieVyh</i> “	86
Obr. 9.28 Pripojenie modulov DCC riadenia ku zberniciam	87

Zoznam tabuliek

tab. 2.1 Požiadavky na vizualizáciu dopravných procesov	15
tab. 2.2 Požiadavky na staničné zabezpečovacie systémy	15
tab. 2.3 Požiadavky na traťové zabezpečovacie zariadenia.....	16
tab. 2.4 Požiadavky na priecestné zabezpečovacie zariadenia	16
tab. 4.1 Prepojenie vstupov a výstupov PLC s dekodérmi	24
tab. 5.1 Zoznam použitých modulov pre PLC S7-1200	27
tab. 5.2 Zoznam použitých modulov pre PLC S7-300	27
tab. 7.1 Štruktúra slovníka „ <i>dictStavanieCesty</i> “	41
tab. 7.2 Popis výberovej ponuky pre odchodové návestidlo.....	49
tab. 8.1 Popis registrov 3 – 6 digitálnych dekodérov LS-100	58
tab. 9.1 Popis výberových ponúk návestidiel	74
tab. 9.2 Popis prvkov ponuky pre ostatné elementy vizualizácie	75

Zoznam skratiek

Skratka	Anglický význam	Slovenský význam
ACC	Area Control Component	Komponent riadenia a kontroly vonkajších prvkov
API	Application Programming Interface	Aplikačné programovacie rozhranie
DCC	Digital Command Control	Digitálne riadenie
DPS	Printed Circuit Board	Doska plošných spojov
ECC	Element Control Computer	Objektový ovládací počítač
GVD	Timetable of Train Traffic	Grafikon vlakovej dopravy
HMI	Human Machine Interface	Rozhranie človek-stroj
IIC	Interlocking and Interface Component	Komponent centrálnych funkcií stavadla
JSON	JavaScript Object Notation	Zápis objektu JavaScript
NC	Normally Closed	Normálne zopnutý kontakt
NO	Normally Open	Normálne rozpojený kontakt
OMC	Overhead Management Component	Komponent manažmentu konfigurácie stavadla
PLC	Programmable Logic Computer	Programovateľný logický automat
PZZ	Level crossing system	Priecestné zabezpečovacie zariadenie
REST	Representational State Transfer	Prenos reprezentatívneho stavu
SZZ	Interlocking system	Staničné zabezpečovacie zariadenie
URI	Uniform Resource Identifier	Jednotný identifikátor zdroja
XML	Extensible Markup Language	Rozšíriteľný značkovací jazyk
ŽSR	Railways of the Slovak Republic	Železnice Slovenskej republiky

ÚVOD

Ako tému našej diplomovej práce sme si zvolili návrh dispečerského pracoviska modelovej železnice. Diplomovú prácu sme sa rozhodli spracovať v rámci výstavby modelového koľajiska Katedry riadiacich a informačných systémov, Fakulty elektrotechniky a informačných technológií, Žilinskej univerzity v Žiline, určenom na výučbu predmetov z oblasti riadiacich systémov pre železničnú dopravu.

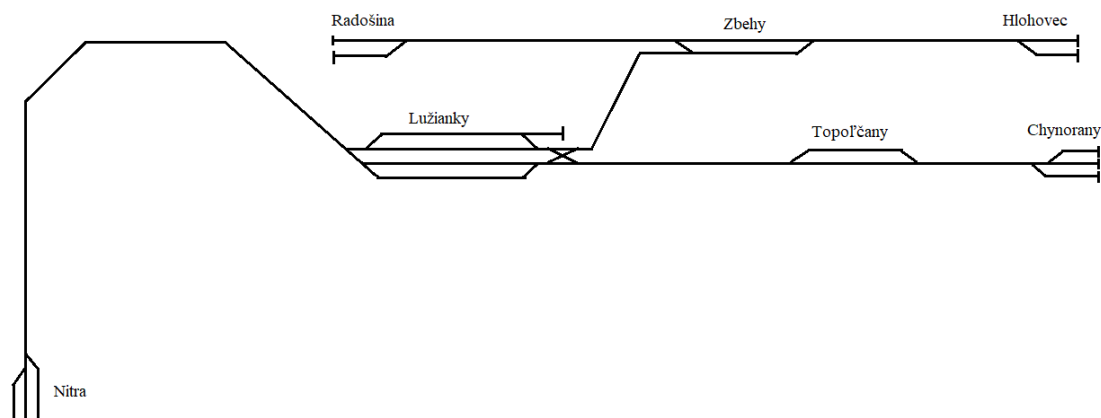
Cieľom práce je návrh a implementácia aplikácií obslužných pracovísk systému ILTIS-N pre vybranú trať modelového koľajiska. Vytvorené aplikácie sú v rozsahu hlavného pracoviska vzdialenej obsluhy a troch pracovísk lokálnej obsluhy, pre každú stanicu ležiacu na trati. V diplomovej práci sme sa zaoberali špecifikáciou funkčných a technických požiadaviek na jednotlivé aplikácie dispečerského pracoviska. Hlavnou časťou práce bol návrh a implementácia funkčných algoritmov obslužných pracovísk modelu systému ILTS-N a algoritmov elektronického stavadla SIMIS W spoločnosti Siemens Mobility.

Modelové koľajisko má slúžiť ako učebná pomôcka vo výučbovom procese, zameranom na železničné zabezpečovacie systémy, pre názorné ukážky spôsobu obsluhy rôznych zabezpečovacích systémov. Okrem toho sa má využívať aj na prezentačné účely katedry. Na modelovej železnici sú implementované rôzne systémy, bežne používané aj v praxi. Na trati sa nachádza model systému ESA-44 s obslužným pracoviskom JOP spoločnosti AŽD, náš model systému SIMIS W s obslužným pracoviskom ILTIS-N, model priecestného zabezpečovacieho zariadenia PZZ-EA a modely traťových zabezpečovacích systémov AB3-74 a AH-88.

1 ANALÝZA SÚČASNÉHO STAVU

V našej diplomovej práci sme sa venovali návrhu a implementácii obslužného pracoviska pre modelovú železnicu v laboratóriu AB 205 Katedry riadiacich a informačných systémov Žilinskej univerzity v Žiline. Modelová železnica sa skladá z dvoch prepojených tratí, demonštrujúcich riadenie dopravy s využitím rôznych typov riadiacich systémov, používaných na tratiach ŽSR. V nasledujúcich podkapitolách je uvedený opis súčasného stavu oboch tratí.

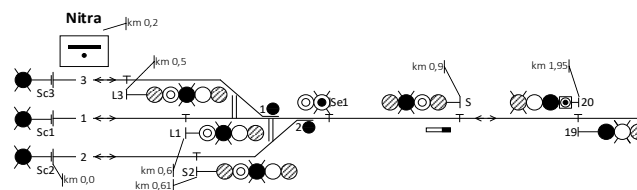
Pohyb vozidiel po trati je riadený digitálnou ústredňou firmy Lenz LZV 100. Tento spôsob riadenia bol zvolený, pretože poskytuje univerzálny a z hľadiska obsluhy jednoduchý spôsob ovládania jazdy dráhových vozidiel na trati. Nie je potrebné v riadiacom systéme implementovať algoritmy určené pre riadenie jazdy vozidiel. Okrem toho je modelové koľajisko verné reálnej prevádzke, kde je jazda vozidiel riadená rušňovodičmi nezávislými od riadiaceho systému. Zároveň firma Lenz vo svojom portfóliu ponúka produkty, umožňujúce riadenie prvkov v modelovom koľajisku (výhybky, návěstidlá) pomocou algoritmov v PC, prostredníctvom komunikačného rozhrania medzi PC a riadiacou DCC ústredňou.



Obr. 1.1 Schematické znázornenie modelového koľajiska [2]

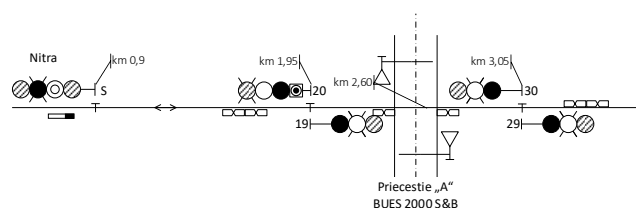
1.1 Trať č.1 Nitra - Chynorany

Trať č.1 pozostáva z troch staníc, výhybne a troch medzistaničných úsekov. Počiatočná stanica Nitra (Obr. 1.2) je navrhnutá ako úvraťová stanica s tromi dopravnými koľajami. Vzhľadom na použitie trojznakového automatického bloku v priliehajúcom medzistaničnom úseku, sú odchodové návěstidlá predzvest'ami prvého oddielového návěstidla. Cez výhybky je do odbočného smeru uvažovaná rýchlosť 40 km/h.

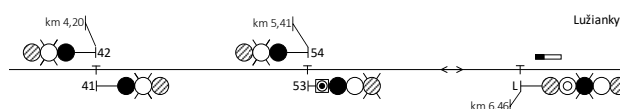


Obr. 1.2 Situačná schéma stanice Nitra

Priliehajúci medzistaničný úsek (Obr. 1.3 a Obr. 1.4) je rozdelený na štyri priestorové oddiely, kryté oddielovými návěstidlami trojznakového automatického bloku. V medzistaničnom úseku sa nachádza úrovňové železničné priecestie A, ktoré má byť v budúcnosti ovládané priecestným zabezpečovacím zariadením BUES 2000 spoločnosti Scheidt & Bachmann, nainštalovaným v laboratóriu.

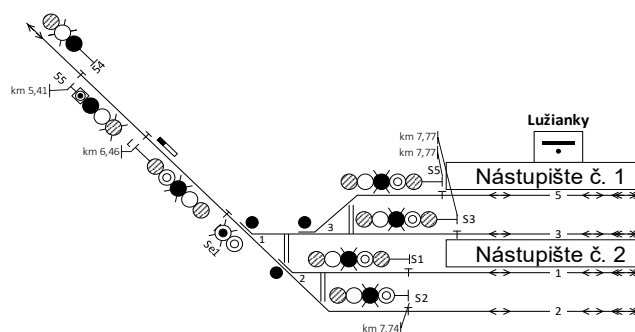


Obr. 1.3 Situačná schéma trate Nitra - Lužianky, časť 1

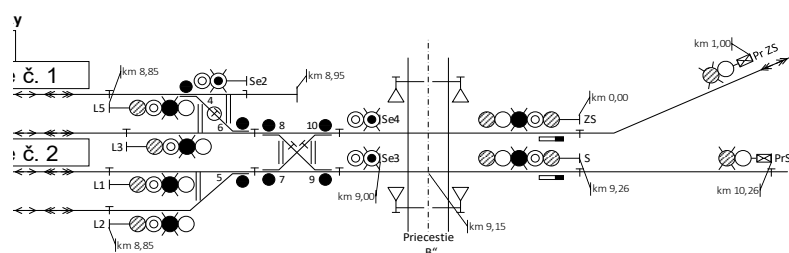


Obr. 1.4 Situačná schéma trate Nitra - Lužianky, časť 2

Nasledujúca stanica Lužianky (Obr. 1.5 a Obr. 1.6) je na celej trati najväčšia. Pozostáva zo štyroch dopravných koľají. Odchodové návěstidlá na ľavom zhlaví sú opäť doplnené horným žltým svetlom, pretože plnia funkciu predzvesti pre prvé oddielové návěstidlo trojznakového automatického bloku. Na pravom zhlaví do stanice ústi krátka prípojná trať z trate č.2 zo stanice Zbehy. Okrem toho sa na zhlaví nachádza aj úrovňové železničné priecestie. V celej stanici je pri jazde do odbočného smeru cez výhybky uvažovaná maximálna rýchlosť 40 km/h.

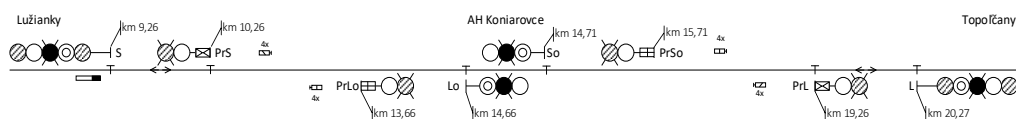


Obr. 1.5 Situačná schéma stanice Lužianky, časť 1



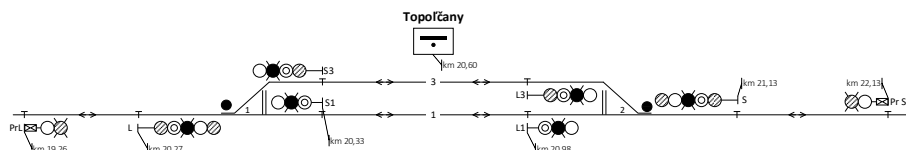
Obr. 1.6 Situačná schéma stanice Lužianky, časť 2

V nasledujúcom medzistaničnom úseku (Obr. 1.7) je realizovaný dvojnakový automatický blok v terminológii ŽSR označované ako automatické hradlo. Medzistaničný úsek je rozdelený na dva priestorové oddiely hradlom Koniarovce. Hlavné návestidlá v danom úseku sú doplnené samostatnými predzvest'ami.



Obr. 1.7 Situačná schéma trate Lužianky - Topoľčany

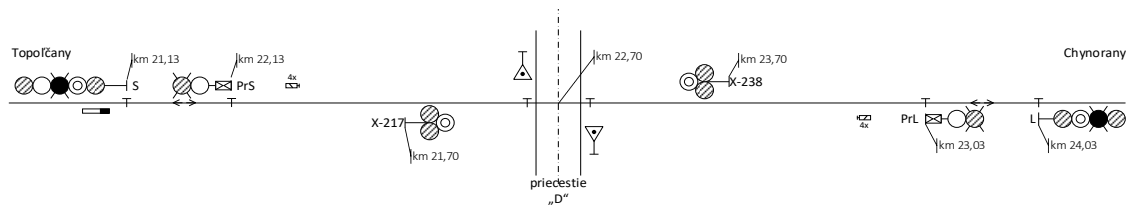
Výhybňa Topoľčany (Obr. 1.8) je tvorená dvomi dopravnými koľajami. Riadenie dopravy v obvode výhybne je zabezpečené vchodovými a odchodovými návestidlami. Nie je tu možné vykonať zabezpečený posun. Pre jazdu do odbočného smeru cez výhybky je uvažovaná maximálna rýchlosť 40 km/h.



Obr. 1.8 Situačná schéma výhybne Topoľčany

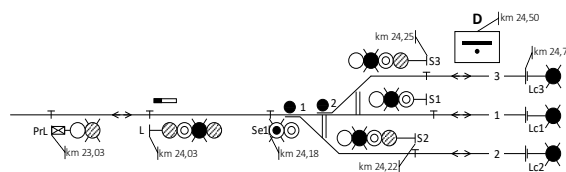
Posledný medzistaničný úsek medzi výhybňou Topoľčany a stanicou Chynorany (Obr. 1.9) obsahuje iba jeden priestorový oddiel. Jazda vlakov v danom úseku je riadená

odchodovými návěstidlami susedných dopravní. V úseku sa nachádza úrovňové železničné priecestie, ktoré je pre oba jazdné smery kryté priecestníkmi.



Obr. 1.9 Situačná schéma traťového úseku Topoľčany – Chynorany

Koncovou stanicou trate je stanica Chynorany (Obr. 1.10). Jedná sa o úvratovú stanicu s tromi dopravnými koľajami. Topologicky je totožná so stanicou Nitra. Aj v tejto stanici je pri jazde do odbočného smeru cez výhybky uvažovaná maximálna rýchlosť 40 km/h.



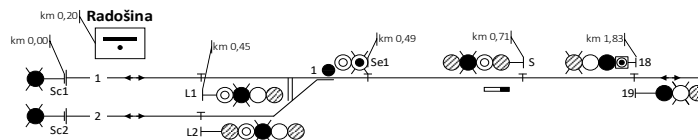
Obr. 1.10 Situačná schéma stanice Chynorany

Návrh riadenia zabezpečovacích systémov pre túto trať bol témou inej diplomovej práce. Prvky v koľajisku v jednotlivých staniach tejto trate sú riadené simulačným modelom elektronického stavadla firmy AŽD, ESA-44, so vzdialenými elektronickými panelmi rozhrania EIP, riadiacimi vonkajšie prvky v jednotlivých staniach. Riadenie vzdialených panelov a zobrazovanie prevádzkovej situácie v riadených staniach trate, je realizované prostredníctvom zadávacieho počítača, z ktorého možno diaľkovo ovládať všetky stanice v riadenej oblasti. Zobrazovanie prevádzkovej situácie a zadávanie príkazov je realizované v súlade s predpisom pre jednotné obslužné pracovisko (JOP), používané ako štandard na tratiach Správy železníc v Českej republike, ale aj v mnohých staniach v sieti ŽSR. Simulačný model použitý v laboratóriu navyše umožňuje aj automatické riadenie jazdy vlakov v riadenej oblasti [15].

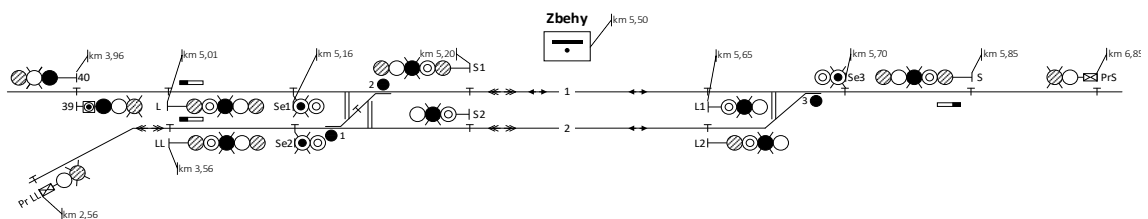
1.2 Trať č.2 Radošina - Hlohovec

Trať č.2 je zadná, kratšia trať, zložená z troch staníc: Radošina, Zbehy a Hlohovec. Stanice Radošina (Obr. 1.11) a Hlohovec (Obr. 1.13) sú koncové úvratové stanice. Obsahujú dve dopravné koľaje ukončené zarážadlami a návěstidlami, trvalo návěstiacimi „Stoj“. Pre jazdu do odbočného smeru cez výhybku je navrhovaná rýchlosť 40 km/h. Odchodové návěstidlá v stanici Radošina sú doplnené horným žltým svetlom, pretože plnia funkciu predzvesti pre prvé oddielové návěstidlo trojznakového automatického bloku. Stanica Zbehy (Obr. 1.12) má dve dopravné koľaje. Pre jazdu

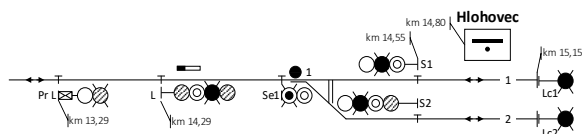
do odbočného smeru cez výhybky je navrhovaná rýchlosť 40 km/h. Do stanice okrem trate č.2 ústi krátka prípojná trať zo stanice Lužianky (Obr. 1.15). Odchodové návěstidlá pre smer do Radošiny sú zároveň predzvesti pre prvé oddielové návěstidlo trojznakového automatického bloku na trati v smere do Radošiny, preto sú doplnené horným žltým svetlom.



Obr. 1.11 Situačná schéma stanice Radošina

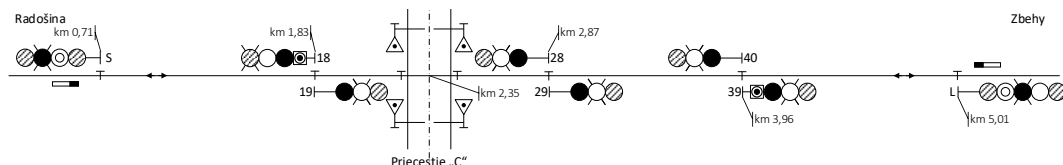


Obr. 1.12 Situačná schéma stanice Zbehy



Obr. 1.13 Situačná schéma stanice Hlohovec

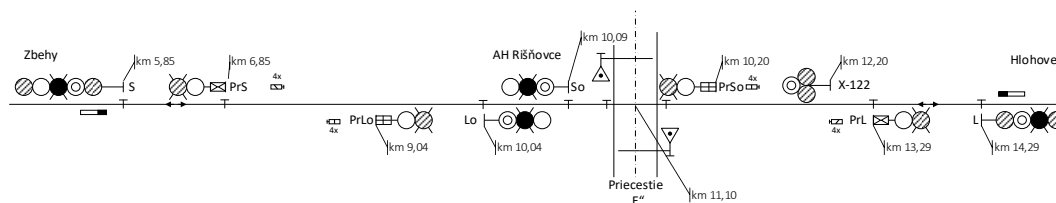
Trať sa skladá z dvoch medzistaničných úsekov. Jazda vlakov v medzistaničnom úseku Radošina – Zbehy (Obr. 1.14) je riadená obojsmerným trojznakovým automatickým blokom. Medzistaničný úsek je rozdelený na štyri priestorové oddiely. Druhý medzistaničný úsek Zbehy – Hlohovec (Obr. 1.16) je vybavený obojsmerným automatickým hradlom. Úsek je rozdelený na dva približne rovnaké priestorové oddiely hradlom Rišňovce. Oddielové návěstidlá na hradle Rišňovce sú voči sebe vzájomne posunuté. V priestore medzi návěstidlami je zriadená ochranná dráha.



Obr. 1.14 Situačná schéma trate Radošina - Zbehy



Obr. 1.15 Situačná schéma trate Lužianky – Zbehy



Obr. 1.16 Situačná schéma trate Zbehy – Hlohovec

V oboch medzistaničných úsekoch je situované jedno úrovňové železničné priecestie. Priecestia sú vybavené svetelným výstražným zariadením s pozitívnou návessou a doplnené polovičnými závorami. Priecestie situované v úseku Radošina – Zbehy, je pre oba dopravné smery kryté oddielovými návěstidlami. Druhé priecestie v úseku Zbehy – Hlohovec je zo smeru od Zbehov kryté oddielovým návěstidlom Lo hradla Rířňovce a zo smeru od Hlohovca kryté priecestníkom X-122 [2].

1.2.1 Riadiaci systém

Základným prvkom hardvérovej časti trate je riadiaci modul (konkrétne PLC SIMATIC S7-300 od spoločnosti Siemens), ktoré v staniciach a medzistaničných úsekoch riadi vonkajšie prvky zabezpečovacích zariadení (prestavníky výmen a svetlá návěstidiel) a sníma ich stavy. Na riadenie a kontrolu železničných priecestí je tiež použité PLC SIMATIC S7-1200 od spoločnosti Siemens. Rozhranie medzi PLC a obsluhou je tvorené HMI panelom MP 277 8'' key. Vzhľadom na viacero bakalárskych prác venujúcich sa danej trati, je vizualizácia rozdelená na dve časti [2][3].

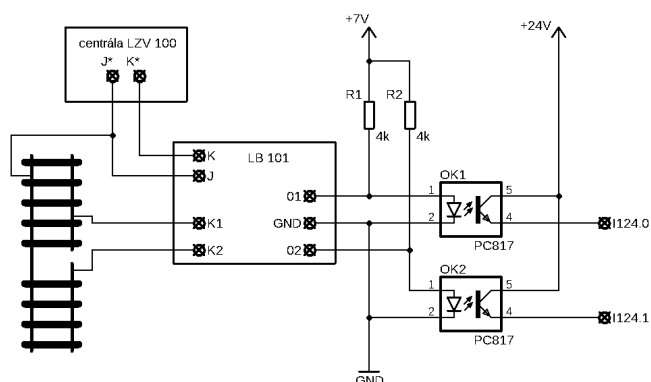
1.2.2 Hardvérové komponenty

Rozhranie medzi PLC a traťou je riešené sériou elektronických komponentov a dosiek plošných spojov. Vstupy do PLC tvoria primárne informácie o obsadenosti koľajových úsekov. Pre zachovanie jednotného spôsobu detekcie prítomnosti koľajových vozidiel na oboch tratiach, bol použitý rovnaký modul ako na trati č. 1. Na tento účel bol použitý hotový modul prúdového detektoru prítomnosti koľajového vozidla v úseku od firmy Lenz LB-101 (Obr. 1.17), ktorého základom je antiparalelné zapojenie diód. Okrem toho slúži aj na napájanie koľajových vozidiel počas jazdy. Rozsah pracovných prúdov sa pohybuje od 1 mA do 3 A.



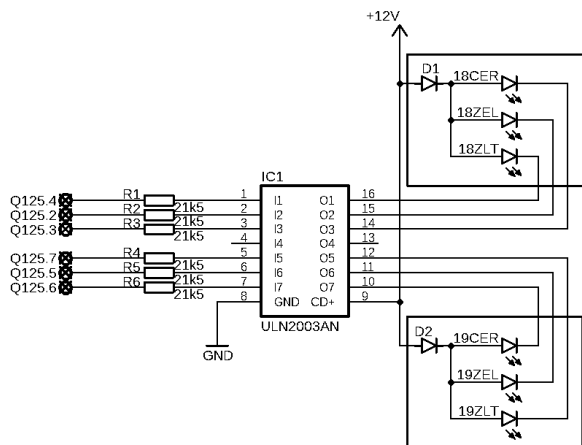
Obr. 1.17 Prúdový modul Lenz LB 101 [13]

Vzhľadom k nekompatibilite výstupných napäťových úrovní modulu LB-101 a napäťových úrovní vstupného modulu PLC (Simatic SM321), bolo nutné použiť napäťový menič (Obr. 1.18). Ten je vytvorený z optočlenu PC 817 v kombinácii s rezistorom pre nastavenie vstupného napätia. Fototranzistor optočlenu je pripojený na vstup PLC. Na strane fotodiódy je pripojený prúdový modul, ktorý v prípade obsadeného koľajového úseku otvorí optočlen na svojom výstupe, čím nastane pokles napätia aj na externom optočlene a vstup PLC sa prepne z log. 1 do log. 0. Vďaka tomu je vytvorená simulácia funkcie paralelného koľajového obvodu – t.j. v stave neprítomnosti vozidla v koľajovom úseku, je na vstupe signál log. 1, v stave prítomnosti koľajového vozidla v úseku, je na vstupe signál log. 0.



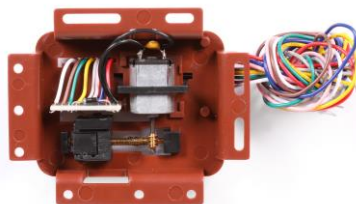
Obr. 1.18 Schéma rozhrania trať - vstup PLC [2]

Výstupy PLC sú určené na ovládanie LED na návestidlách. Na trati sú použité návestidlá od firmy Němeček vo vyhotovení zapojenia so spoločnou anódou. Napájacie napätie je výrobcom stanovené na 12 VDC, čo tiež nie je kompatibilné s výstupným napätím PLC (24 VDC). Pre prispôbenie výstupného napätia PLC požadovanému napájaciemu napätiu návestidiel bolo použité integrované tranzistorové darlingtonové pole ULN2003A vo funkcii invertujúceho spínača (Obr. 1.19). Integrovaný obvod je doplnený vstupným rezistorom pre úpravu signálového napätia na vstupe.



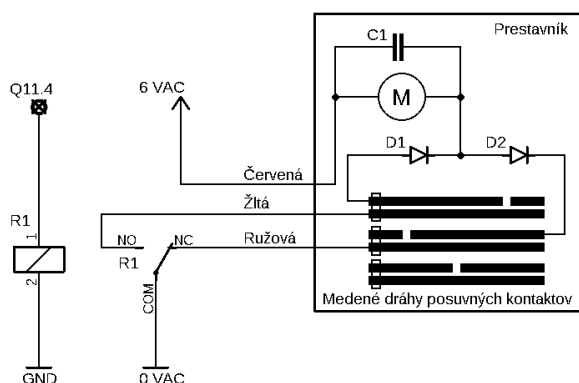
Obr. 1.19 Prispôbenie výstupu PLC zapojeniu návestidla [2]

Okrem návěstidiel bolo nutné pomocou PLC ovládať prestavníky výhybiek v staniciach. Na tento účel bol využitý modul elektromotorického prestavníka firmy Tillig (produktové číslo 86112, Obr. 1.20). Jeho základom je malý DC motor, ktorý prostredníctvom prevodu so závitovou tyčou a pevného pružného drôtu pohybuje jazykmi výmeny.



Obr. 1.20 Elektromotorický prestavník Tillig [14]

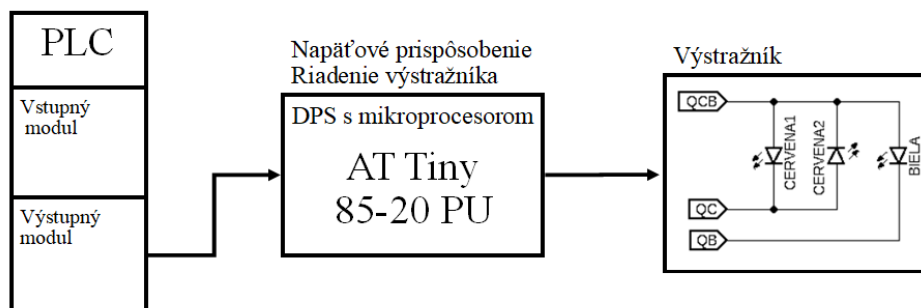
Napájacie napätie prestavníkov bolo na základe odporúčania od výrobcu zvolené 6 VAC. Vzhľadom na rôzne povahy napätí, striedavé pre napájanie prestavníka a jednosmerné z PLC, je do obvodu vložené elektromechanické relé PE14024. Cievka relé je ovládaná napätím z PLC a na NO a NC kontakty sú pripojené vodiče prestavníka tak, aby bol zabezpečený obojsmerný chod. Vzhľadom na konštrukciu prestavníka, ktorá po dosiahnutí koncovej polohy odpája napájacie napätie od motora, nemusel byť tento problém riešený v PLC a mohli byť využité oba kontakty relé (Obr. 1.21).



Obr. 1.21 Schéma pripojenia prestavníka k PLC [2]

Výstražníky firmy Němeček použité na priecestiach, sú vyhotovené pre trojvodičové zapojenie (Obr. 1.22). Vnútorne usporiadanie výstražníka tvorí antiparalelné zapojenie dvoch červených LED medzi spoločným vodičom CB a vodičom C, doplnené jednou bielou LED medzi vodičmi CB a B. Vzhľadom na antiparalelné zapojenie červených LED vo výstražníku, nebolo možné výstražník pripojiť priamo k výstupom PLC. Druhým problémom bolo napäťové prispôsobenie výstražníka a PLC. Bol vytvorený prispôbovací riadiaci obvod. Jeho základom je mikroprocesor AT Tiny 85-20PU. Mikroprocesor zabezpečuje prerušované svietenie výstražných červených LED. Obvod je doplnený napäťovým prispôobením medzi výstupným napätím PLC, napätiami mikroprocesora a LED výstražníka. Pre obe priecestia bola vyhotovená DPS,

obsahujúca spomínaný mikroprocesor spolu s pomocným ovládacím obvodom pre priecestník. Schéma zapojenia DPS sa nachádza v prílohe (Príloha B)

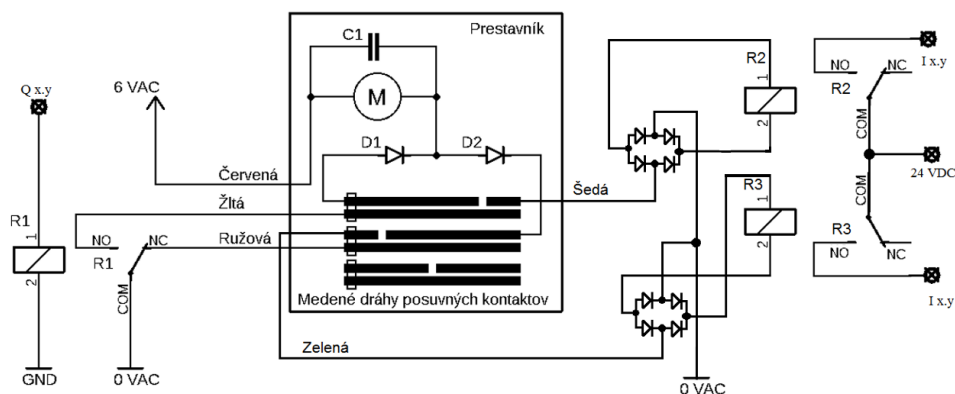


Obr. 1.22 Schéma pripojenia výstražníka [3]

Pohon závor je zabezpečený elektromotorickým prestavníkom Tillig (Obr. 1.20). Prestavník je pod stolom upevnený v zvislej polohe tak, aby bol pohyb bežca na závitovej tyči vedený vertikálne. Ovládanie pohybu závor je riadené pomocou priemyselných relé priamo z PLC. [3].

1.2.3 Doplnenie hardvérových prvkov koľajiska

Pri analýze súčasného stavu koľajiska sme zistili, že nie je navrhnuté snímanie koncových polôh výmen v staniciach. Preto sme navrhli DPS na snímanie koncovej polohy výmen. Použitý prestavník Tillig má inštalované kontakty, umožňujúce detekciu koncovej polohy pohonu prestavníka. Obr. 1.23 znázorňuje vnútornú stavbu medených dráh pre bežce, ktoré sú pevne spriahnuté s prestavnou tyčou. Pri privedení napätia medzi červený a žltý vodič sa prestavná tyč aj s bežcami pohne smerom doprava a prúd preteká cez prvý bežec. Po dosiahnutí koncovej polohy sa bežec dostane na krátku odizolovanú časť medenej dráhy pripojenej k šedému vodiču. Motor prestavníka sa zastaví, vybudí sa relé R2 a na vstup PLC je privedené napätie. V prípade prestavovania do opačnej polohy je napätie privedené medzi červený a ružový vodič, prúd preteká cez druhý bežec. Po dosiahnutí koncovej polohy je motor odpojený od napätia, relé R3 sa vybudí a napätie je privedené na iný vstup PLC.



Obr. 1.23 Schéma zapojenia prestavníka Tillig

Pre krátkosť času sme sa rozhodli pre najjednoduchšie zapojenie spätnej väzby pomocou relé. Použili sme relé, ktorého cievka je určená na spínanie 5 VDC. Vďaka tomu nepotrebujeme riešiť žiadne napäťové prispôsobovanie medzi napätím z prestavníka (6 VAC) a vstupným napätím PLC 24 VDC. Pre správnu funkčnosť je každé relé doplnené mostíkovým usmerňovačom.

1.2.4 Staničné zabezpečovacie zariadenia

Vzhľadom na to, že návrh riadiacich algoritmov pre staničné zabezpečovacie zariadenia nebol obsahom ani jednej z bakalárskych prác [2][3], boli vytvorené iba zjednodušené funkcie v rámci bakalárskej práce, ktorá sa zaoberala traťovými zabezpečovacími zariadeniami. Aktuálne programové vybavenie riadiaceho systému umožňuje v jednotlivých staniách zjednodušené stavenie vchodových ciest, ktoré zahŕňa príkaz na prestavenie prechádzaných výmen do polohy potrebnej pre stavanú vlakovú cestu a rozsvietenie návestného znaku, podľa aktuálne stavanej vlakovej cesty. Algoritmus však nekontroluje splnenie ďalších podmienok na postavenie jazdnej cesty a taktiež neumožňuje stavenie posunových ciest v staniách. V riadení staníc je implementovaný aj zjednodušený algoritmus rušenia vlakovej cesty jazdou vlaku.

1.2.5 Traťové zabezpečovacie zariadenia

V riadiacom systéme sú zahrnuté riadiace algoritmy, na realizáciu vybraných funkcií traťových zabezpečovacích systémov – bezpečné riadenie jazdy následných vlakov a znemožnenie súčasných protismerných jász vlakov po rovnakej traťovej koľaji. Riadiace algoritmy traťových zabezpečovacích zariadení sú v PLC definované formou funkčných blokov. Funkčné bloky môžu byť rozdelené do dvoch skupín na hlavné a podporné. Medzi hlavné patria funkčné bloky oddielových návěstidiel, predzvestí a čítania informácií z koľajových obvodov. Medzi podporné sú zaradené funkčné bloky staníc.

Základným prvkom hlavných funkčných blokov je stavový automat vytvorený tak, aby odrážal funkčné správanie v praxi bežne používaných prvkov zabezpečovacích systémov. Stavový automat je schopný reagovať na simulované poruchy zadané užívateľom v rozsahu vypálených svetiel na konkrétnom návěstidle, zmenou aktuálneho návěstného znaku na menej povoľujúci a v prípade trojznakového automatického bloku aj prenosom červenej na predchádzajúce hlavné návěstidlo.

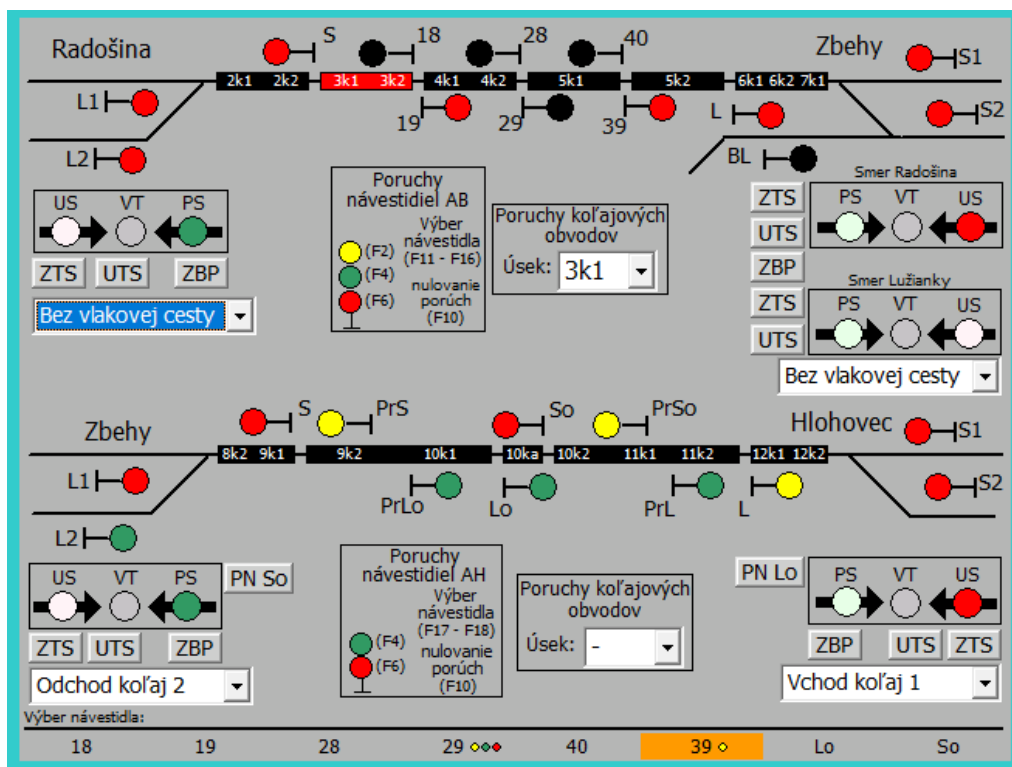
Podporné bloky staníc sú tiež koncipované ako stavové automaty. Nemajú do hĺbky prepracovanú štruktúru. Ich účelom je simulovať funkcie spolupracujúcich staničných zabezpečovacích systémov. Majú v sebe implementované aj algoritmy traťového súhlasu, pre zabezpečenie jazd v medzistaničnom úseku. Pri zmene traťového súhlasu je potrebné, aby stanica bez súhlasu vytvorila žiadosť na udelenie traťového súhlasu, ktorá musí byť potvrdená stanicou odovzdávajúcou traťový súhlas. [2].

1.2.6 Priecestné zabezpečovacie zariadenia

Ovládanie priecestných zabezpečovacích zariadení prebieha v dvoch úrovniach. Prvá úroveň je realizovaná riadiacim algoritmom implementovaným do PLC, spracovávajúcim informácie o obsadenosti koľajových úsekov, tvoriacich približovacie úseky a anulačný úsek priecestia. Spracováva aj povel z HMI pre manuálne ovládanie priecestných zabezpečovacích zariadení. Na základe týchto dát PLC vydáva pokyny na zatvorenie, resp. otvorenie závor, ovláda červené a biele svetlá na výstražníkoch a biele svetlo na priecestníkoch. Okrem toho je v PLC zriadená väzba na traťové zabezpečovacie zariadenia, kvôli krytiu priecestí oddielovými návěstidlami. Druhá úroveň je realizovaná riadiacim algoritmom implementovaným v mikroprocesore AT Tiny, zabezpečujúcim ovládanie svetiel výstražníka vo všetkých prevádzkových stavoch PZZ, na základe riadiacich povelov z prvej úrovne riadenia v PLC [3].

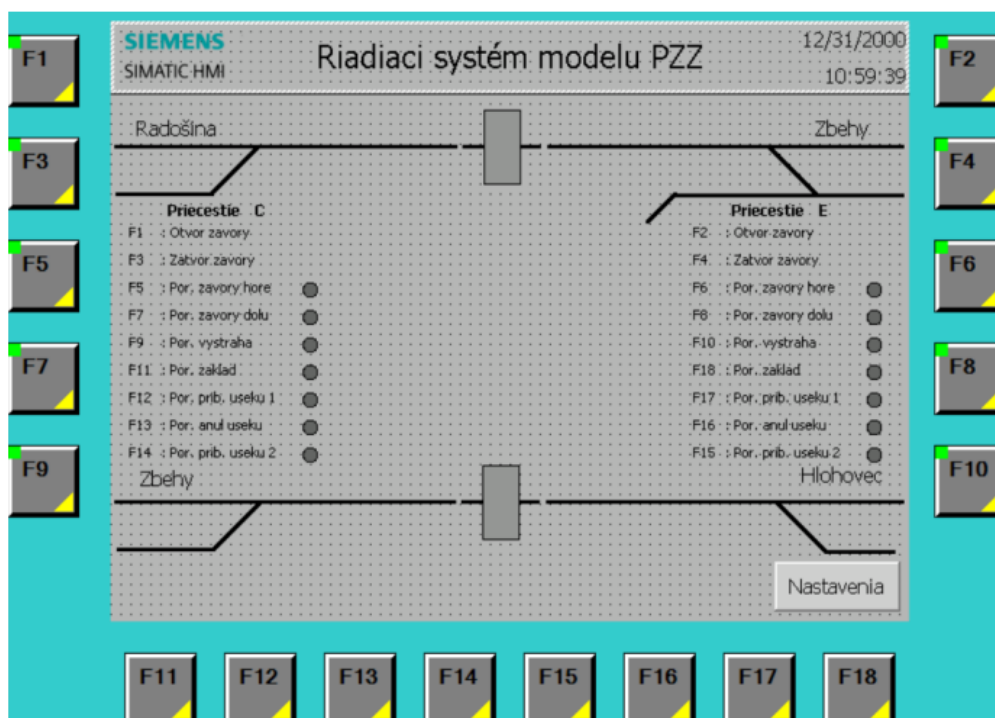
1.2.7 Spôsob riadenia

Ovládanie modelovej železnice je zabezpečené HMI panelom MP 277 8“ key spoločnosti Siemens. Vzhľadom na fakt, že vývoj funkčných algoritmov pre traťové zabezpečovacie zariadenia a priecestné zabezpečovacie zariadenia bol rozdelený na dve samostatné bakalárske práce, ovládanie jednotlivých zariadení nie je unifikované.



Obr. 1.24 Vizualizácia pre traťové zabezpečovacie zariadenia [2]

Jedna z vizualizácií (Obr. 1.24) ponúka možnosti zjednodušeného ovládania staníc v rozsahu potrebnom pre správnu funkčnosť traťových zabezpečovacích zariadení a zobrazenie aktuálneho stavu prvkov v koľajisku (obsadenosť traťových úsekov, návěstný znak na návěstidlách, smer traťového súhlasu, ...).



Obr. 1.25 Vizualizácia pre priecestné zabezpečovacie zariadenia

Druhá vizualizácia (Obr. 1.25) je zameraná na riadenie priecestných zabezpečovacích zariadení. Umožňuje sledovanie prevádzkových stavov priecestí, manuálne ovládanie, konfiguráciu priecestí v rozsahu zmeny sklápacích, anulačných časov a podobne [2][3].

1.2.8 Unifikácia riadiacich systémov

Realizácia algoritmov TZZ a PZZ boli témami dvoch samostatných bakalárskych prác. Aby sa študenti v priebehu riešenia zadania vzájomne neobmedzovali, vedúci oboch prác rozhodol o použití dvoch samostatných PLC na riešenie jednotlivých prác. Momentálne sú na trati č. 2 implementované algoritmy traťových zabezpečovacích systémov riadené z PLC SIMATIC S7-300 a priecestné zabezpečovacie zariadenia riadené z PLC SIMATIC S7-1200. Medzi týmito systémami je zriadená väzba, na zabezpečenie správnej funkčnosti všetkých zariadení. Prípadná unifikácia funkčných algoritmov do jedného PLC by zjednodušila súčasné zapojenie a odbúrala potrebu dodatočných algoritmov pre spracovanie informácií získaných od spolupracujúcich algoritmov. Najlepším riešením by bol presun algoritmov traťových zabezpečovacích zariadení do PLC SIMATIC S7-1200, pretože programová pamäť PLC SIMATIC S7-300 nie je veľká, aby sme do PLC vložili všetky potrebné algoritmy zabezpečovacích systémov pre celú trať.

2 ŠPECIFIKÁCIA POŽIADAVIEK

Pred začiatkom práce na riadiacich algoritmoch sme si definovali požiadavky na pracoviská vzdialenej obsluhy a jednotlivé druhy zabezpečovacích zariadení. Pretože návrh algoritmov pre traťové a priecestné zabezpečovacie zariadenia bol témou bakalárskych prác, pri návrhu požiadaviek na tieto systémy sme vychádzali zo špecifikácií, definovaných v týchto bakalárskych prácach [2][3].

2.1 Požiadavky na vizualizáciu dopravných procesov

Grafické zobrazenie procesov by malo predstavovať zjednodušený model riadiaceho systému ILTIS-N spoločnosti Siemens Mobility. Pri návrhu systému by sme mali vychádzať z aktuálnych predpisov ŽSR D 101 / T 101, najmä príloh 8a, 8b a 8c, špecifikujúcich spôsob obsluhy a použitý katalóg elektronických staničných, traťových a priecestných zabezpečovacích systémov na lokálnej aj centrálnej (dispečerskej) obslužnej úrovni. V tab. 2.1 je obsiahnutý súbor požiadaviek, podľa ktorých sme postupovali pri návrhu a implementácii vizualizácie dopravných procesov [7][8][9].

tab. 2.1 Požiadavky na vizualizáciu dopravných procesov

Ozn.	Požiadavka
F1.1	Grafické rozhranie musí umožňovať základné spôsoby riadenia dopravných procesov
F1.2	Grafické rozhranie musí umožňovať obsluhu jazdných ciest v staniciach
F1.3	Grafické rozhranie musí umožňovať zmenu traťového súhlasu
F1.4	Grafické rozhranie musí umožňovať manuálnu obsluhu priecestí
F1.5	Grafické rozhranie musí zobrazovať aktuálne informácie o riadených prvkoch
T1.1	Grafické rozhranie má byť vypracované ako PC aplikácia
T1.2	Grafické rozhranie má byť spustiteľné z akéhokoľvek PC v laboratóriu

2.2 Požiadavky na staničné zabezpečovacie systémy

Model staničných zabezpečovacích systémov by mal predstavovať zjednodušený variant stavadla SIMIS W spoločnosti Siemens Mobility. Algoritmy tohto modelu by mali byť navrhnuté tak, aby boli splnené všetky podmienky vyplývajúce z použitia staničného zabezpečovacieho zariadenia 3. kategórie podľa zabezpečenia jazdných ciest na základe normy TNŽ 34 2620 [5]. V tab. 2.2 sú uvedené vypracované funkčné požiadavky na staničné zabezpečovacie systémy.

tab. 2.2 Požiadavky na staničné zabezpečovacie systémy

Ozn.	Požiadavka
F2.1	Algoritmy musia umožniť postavenie všetkých ciest definovaných v záverovej tabuľke
F2.2	V základnom stave musí byť na návestidlách zobrazený návestný znak zakazujúci jazdu vlaku, resp. posunujúceho dielu
F2.3	Pri stavaní jazdných ciest musí byť zabezpečená bočná ochrana prestavením potrebných výmen do odvratnej polohy
F2.4	Pri stavaní ciest s rýchlosťou nad 40 km/h musí byť vytvorená ochranná dráha jazdnej cesty
F2.5	Algoritmy musia umožniť zrušenie jazdnej cesty ako celku bez jazdy vlaku
F2.6	Grafické rozhranie musí umožňovať manuálne prestavovanie výmen
F2.7	Algoritmy musia mať zriadenú väzbu na TZZ tak, aby bola zabezpečená predzvestná väzba medzi oddielovými návestidlami automatického bloku a staničnými návestidlami
F2.8	Algoritmy môžu povoliť postavenie odchodovej cesty, iba v prípade voľného traťového oddielu za stanicou a stanica prijala traťový súhlas

2.3 Požiadavky na traťové zabezpečovacie systémy

Na trati č. 2 sa nachádzajú dva medzistaničné úseky, osadené obojsmerným dvojsmerovým automatickým blokom a obojsmerným trojsmerovým automatickým blokom. Podrobná špecifikácia traťových zabezpečovacích systémov sa nachádza

v bakalárskej práci [2]. Funkčné správanie modelov traťových zabezpečovacích systémov by malo vychádzať z požiadaviek normy TNŽ 34 2630 [6]. V tab. 2.3 sú uvedené požiadavky na traťové zabezpečovacie systémy.

tab. 2.3 Požiadavky na traťové zabezpečovacie zariadenia

Ozn.	Požiadavka
F3.1	Systém automatického hradla musí pri poruche návestidla umožniť dať privolávaciu návesť na oddielovom návestidle smerom do vlastnej stanice
F3.2	Po zaznamenaní prejazdu koľajového vozidla okolo hradla musí byť odoslaná odhláška za vlakom späť na predchádzajúce hradlo alebo do stanice
F3.3	Na oddielovom návestidle sa zobrazí povoľujúci návestný znak, iba ak je prijatá odhláška za posledným vlakom a zároveň bola prijatá predhláška pre následný vlak
F3.4	Trojznakový automatický blok musí mať zriadenú predzvestnú väzbu medzi nasledujúcimi hlavnými návestidlami
F3.5	Trojznakový automatický blok musí mať zriadenú predzvestnú väzbu so staničnými návestidlami na príslušnom zhlaví stanice
F3.6	Návestidlá trojznakového automatického bloku platné pre smer, v ktorom nie je udelený traťový súhlas, musia byť zhasnuté
F3.7	Traťový súhlas je možné zmeniť, iba ak je voľný medzistaničný úsek a do úseku nie sú postavené žiadne odchodové cesty

2.4 Požiadavky na priecestné zabezpečovacie systémy

Na našej trati sa nachádzajú dve úrovňové železničné priecestia s aktívnou signalizáciou, doplnené svetelnou výstrahou s polovičnými závorami. Priecestné zabezpečovacie zariadenia by mali byť navrhnuté tak, aby spĺňali požiadavky na priecestné zariadenia 3. kategórie podľa normy STN P 34 2651 [4]. Podrobná špecifikácia priecestných zabezpečovacích zariadení je uvedená v bakalárskej práci [3]. V tab. 2.4 sú uvedené požiadavky na priecestné zabezpečovacie zariadenia.

tab. 2.4 Požiadavky na priecestné zabezpečovacie zariadenia

Ozn.	Požiadavka
F4.1	Svetelná výstraha musí umožniť signalizáciu všetkých prevádzkových stavov priecestia
F4.2	Systémy PZZ musia mať vytvorenú väzbu na TZZ, pokiaľ sa požaduje krytie priecestia oddielovým návestidlom
F4.3	Ak nie je priecestie kryté hlavným návestidlom, musí byť doplnené priecestníkom aby bola zabezpečená signalizácia stavu priecestia rušňovodičovi
F4.4	PZZ musia umožniť manuálnu obsluhu priecestia z najbližšej dopravne
F4.5	Medzi PZZ a grafickou vizualizáciou má byť zriadená väzba pre prenos aktuálneho stavu PZZ

2.5 Komunikácia modelov ESA-44 a SIMIS W

Na modelovom koľajisku sa nachádza krátka spojovacia trať medzi stanicami Lužianky a Zbehy. Stanica Lužianky je riadená systémom ESA-44 a stanica Zbehy je riadená modelom systému SIMIS W. Pre vytvorenie bezpečnej jazdy vlakov v úseku je potrebné, aby oba systémy spolu komunikovali. Preto sme navrhli protokol pre komunikáciu modelov systémov ESA-44 a SIMIS W na modelovom koľajisku.

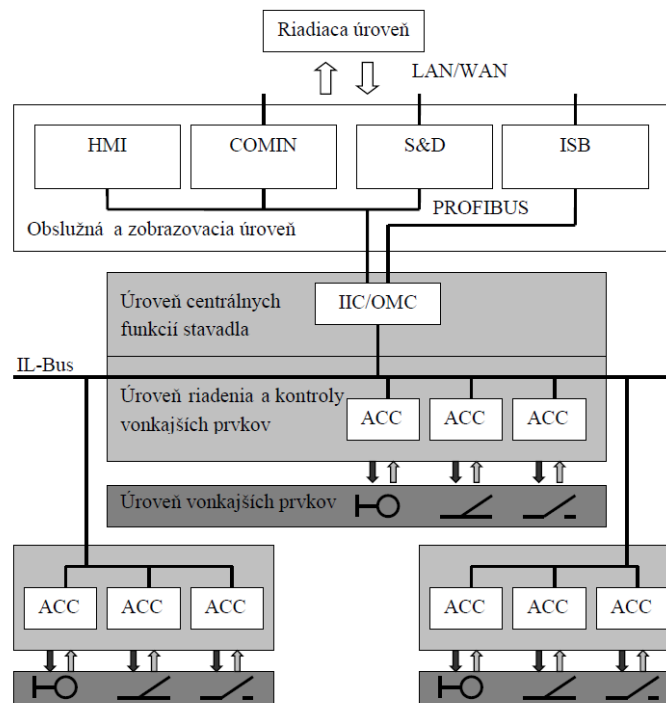
Protokol je navrhnutý tak, aby zabezpečil obojsmerný prenos informácií medzi riadiacimi systémami. Medzi prenášané dáta sú zahrnuté:

1. Informácia o voľnosti medzistaničného úseku
2. Aktuálny smer traťového súhlasu
3. Žiadosť o udelenie traťového súhlasu
4. Informácie týkajúce sa blokovej podmienky
5. Predhláška pred vlakom
6. Odhláška za vlakom
7. Aktuálny návestný znak na vchodovom návestidle

3 ELEKTRONICKÉ STAVADLO SIMIS W

Elektronické Stavadlo SIMIS W od firmy Siemens je celosvetovo rozšírený typ stavadla. Stavadlo SIMIS W spĺňa požiadavky na úroveň integrity bezpečnosti SIL 4 a je typické vysokou pohotovosťou, pretože pracuje v architektúre 2oo3. Medzi jeho charakteristiky patrí hlavne široká možnosť aplikácie stavadla – od malých staníc, až po veľké železničné uzly so stovkami vonkajších prvkov. Umožňuje prepojenie s už existujúcimi zariadeniami a jednoduché úpravy systému. Elektronické stavadlo SIMIS W je doplnené obslužným pracoviskom ILTIS alebo VICOS na lokálnej alebo vzdialenej dispečerskej úrovni. Vnútorňú štruktúru stavadla SIMIS W s obslužným pracoviskom môžeme rozdeliť na niekoľko úrovní:

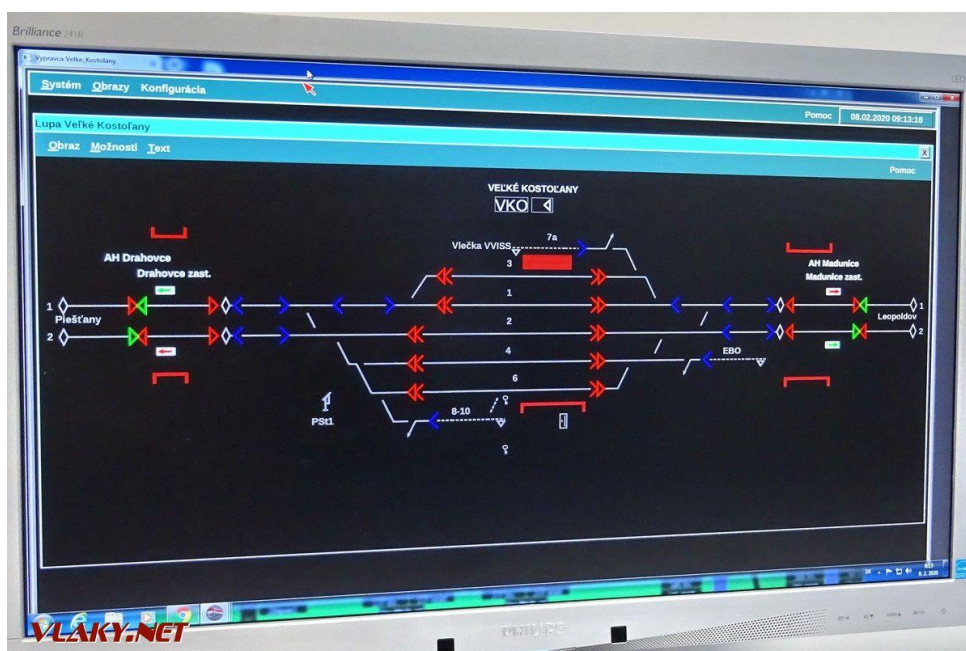
1. riadiaca úroveň,
2. obslužná úroveň,
3. úroveň centrálnych funkcií stavadla,
4. úroveň riadenia a kontroly vonkajších prvkov [1].



Obr. 3.1 Štruktúra stavadla SIMIS W s obslužným pracoviskom [1]

3.1 Riadiaca úroveň

Pod pojmom riadiaca úroveň rozumieme dispečerské pracovisko, komunikujúce s viacerými stavadlami. Posiela im príkazy od obsluhy a obsluha zobrazuje prevádzkové stavy podľa telegramov zo stavadiel. Aktuálne sú na trhu k stavadlu SIMIS W ponúkané dva varianty obslužného pracoviska – ILTIS a VICOS.



Obr. 3.2 Príklad obslužného pracoviska ILTIS-N v stanici Veľké Kostoľany [11]

3.2 Obslužná úroveň

Obslužná úroveň je tvorená lokálnym obslužným pracoviskom, podľa potreby vybaveným jedným alebo viacerými monitormi, myšou a klávesnicou. Obslužné pracovisko je pripojené k riadiacej úrovni COMIN, k servisnému a diagnostickému zariadeniu a zariadeniu na uvedenie do prevádzky ISB. Prvky na tejto vrstve nemusia spĺňať požiadavky na integritu bezpečnosti SIL 4. Komunikácia prvkov je zabezpečená zbernicou Profibus. Pre zaistenie pohotovosti je komunikácia zdvojená (1oo2).

Obslužné pracovisko tvorí rozhranie medzi obslužným pracovníkom a elektronickým stavadlom. Jedno stavadlo môže byť rozdelené medzi viacero obslužných pracovísk. Každé z nich reaguje na každý povel zamestnanca buď akustickou, alebo optickou vizualizáciou zmeny stavu. Hlásenia systému sú prístupné pre všetky pracoviská, ale každé pracovisko selektuje iba preňho relevantné hlásenia.

Servisné a diagnostické zariadenie je reprezentované počítačom, slúžiacim na zber a analýzu stavových informácií zo stavadla. Zabezpečuje archiváciu dát, ukladanie poruchových dát v prípade výpadku elektronického stavadla, listovanie aktívnych porúch a zobrazovanie dokumentácie k údržbe.

Zariadenie na uvedenie do prevádzky je tiež tvorené počítačom. Ako názov napovedá, je určené hlavne na diagnostiku počas testovania stavadla, prototypovanie hlásení systému. V aktívnej prevádzke nie je potrebné, ale dá sa využiť na zber a vyhodnocovanie systémových hlásení [1].

3.3 Úroveň centrálnych funkcií stavadla

Úroveň centrálnych funkcií stavadla sa delí na dva komponenty. Prvý z nich, OMC, spravuje topografické dáta stanice, ktoré má uložené v pamäti. Okrem toho sprostredkováva správu stavadlovej konfigurácie a systémových stavov. Druhý komponent, IIC, spravuje centrálné úlohy stavadla, kontroluje správnosť príkazov z obslužnej úrovne, poskytuje dáta pre úroveň riadenia a kontroly vonkajších prvkov, generuje a kontroluje dáta pre servis a diagnostiku. Komponenty IIC a OMC sa skladajú z počítačov ECC, vyvinutých spoločnosťou Siemens. Počítače ECC sú navrhnuté v architektúre 2oo3 s hardvérovou komparáciou všetkých kanálov. Vstupné dáta sú spracovávané paralelne, spúšťanie mikroprocesorov počítačov je cyklické kontrolným softvérom. Počítače ECC sú doplnené komunikačnými modulmi [1].

3.4 Úroveň riadenia kontroly vonkajších prvkov

Úroveň kontroly a riadenia vonkajších prvkov tvoria ACC komponenty. Každý z nich pokrýva určitú časť vonkajších prvkov stavadla. Od výrobcu majú všetky komponenty rovnaký programový základ. Pri spúšťaní stavadla do prevádzky sa z OMC komponentu prevezmú špecifické dáta, určujúce konkrétny typ komponentov riadených z daného ACC. Informácie získané z vonkajších komponentov sú ďalej odosielané na obslužnú úroveň cez IIC. Na realizáciu ACC komponentov sa tiež používajú počítače ECC. Okrem mikroprocesorových modulov obsahuje počítač ECC aj moduly komunikačnej zbernice IL-Bus a špecifické moduly periférií podľa pripojených vonkajších komponentov [1].



Obr. 3.3 Počítač SIMIS ECC použitý v ACC komponente [12]

3.5 Štruktúra softvéru

Softvérové vybavenie stavadla sa delí na tri úrovne. Základná úroveň obsahuje systémový softvér, tvoriaci jadro stavadla a je nutný pre jeho fungovanie (operačný systém, prerušenia, časovače, ...). Na systémový softvér nadväzuje bázový softvér, obsahujúci špecifické funkcie ako komunikačné protokoly, bezpečnostné funkcie alebo systémová logika. Bázový a systémový softvér je nezávislý od zákazníka a patrí k základnej výbave každého stavadla. Ďalšia úroveň softvérového vybavenia stavadla je definovaná koncovým zákazníkom. Ide o užívateľsky špecifický softvér, vyplývajúci z technických noriem a prevádzkových predpisov daného zákazníka. V tejto úrovni sú implementované všeobecné dopravno-bezpečnostné algoritmy staničných, traťových a priecestných zabezpečovacích systémov. Táto časť je posudzovaná iba raz pre každého zákazníka a následne je použitá vo všetkých jeho aplikáciách stavadla. Poslednou úrovňou je projektovo špecifický softvér, obsahujúci topologické informácie konkrétnej stanice, kde je stavadlo implementované [1].



Obr. 3.4 Delenie softvéru stavadla SIMIS W

3.6 Princíp činnosti

Elektronické stavadlo SIMIS W je založené na základe definovaných oblastí. Každá stanica je rozdelená na viac oblastí, riadených z ACC komponentov. V prípade výpadku niektorého riadiaceho ACC nenastane výpadok ovládania celej stanice, ale len jednej z jej častí, čo stále umožňuje prevádzku stanice aspoň v obmedzenom režime.

Stavadlo SIMIS W pracuje na geografickom princípe. Ku každému vonkajšiemu prvku sa pristupuje ako k jednému objektu. V logike stavadla sú jednotlivé objekty prepojené väzbami na základe geografie koľajiska. V stavadle sú definované logické objekty, každý so svojim špecifickým správaním a vlastnosťami. Jednotlivé objekty počas činnosti navzájom komunikujú prostredníctvom správ.

Po prijatí vopred definovaného príkazu (druh cesty, počiatkový a cieľový objekt) na stavanie jazdnej cesty od obslužného pracoviska, sa zaháji proces jej postavenia. V logike stavadla sa pri stavaní postupuje od cieľového prvku po počiatkový. Najskôr je v IIC príkaz na stavenie skontrolovaný z logického hľadiska či nie je chybné zadanie a obsahuje všetky potrebné informácie. Následne IIC vydá príkaz konkrétnemu ACC na začatie prvej fázy stavania vlakovej cesty. ACC prevedie kontrolu jednotlivých prvkov, napríklad či patria do zadanej vlakovej cesty a môžu vykonávať žiadanú funkciu. Táto kontrola prebieha po prvkoch počínajúc cieľovým objektom. V prípade, že požadovaný objekt nespravuje aktuálne ACC, tak ACC pošle správu so žiadosťou o činnosť (kontrolu, prestavenie, zablokovanie) ďalšiemu ACC podľa topológie stavanej cesty. Ak je odpoveď objektu negatívna, je na IIC odoslaná negatívna väzba a stavenie je prerušené. V prípade pozitívnej odozvy na IIC nasleduje druhá fáza stavania vlakovej cesty, kedy je znovu vydaný povel pre ACC, ale teraz sa už objekty neskúšajú. Ak ACC narazí na objekt, ktorý je potrebné prestaviť, vydá povel na prestavenie objektu. Pokiaľ by sa prestavovaný objekt nenachádzal v pôsobnosti daného ACC, to opäť vydá povel pre príslušné ACC na zmenu stavu (prestavenie) objektu [1].

4 NÁVRH KONCEPTU RIADIACEHO SYSTÉMU

Štruktúra riadenia je navrhnutá tak, že každý z prvkov nachádzajúcich sa v koľajisku má definované funkčné správanie, na základe ktorého dokáže interagovať z nadriadeným systémom. Naša koncepcia ráta s tým, že nadradený systém odosiela jednotlivým prvkom jednoduché správy. Prvok tieto správy spracuje svojimi vnútornými algoritmami a vykoná požadovanú akciu (prestavenie, zmena návestného znaku...).

4.1 Koncept riadenia vonkajších prvkov

Budiacim vstupom výhybky je podnet na prestavenie. Po vybudení má byť výhybka, ak je to možné, prestavená do požadovanej polohy. Na základe spätnej väzby z fyzického zapojenia prestavníka, má výhybka nepretržite informovať nadradený systém o svojej aktuálnej polohe (prestavená v polohe +, prestavená v polohe –).

Budiacimi vstupmi pre návestidlo sú požiadavka na zmenu návestného znaku a stav koľajového úseku za návestidlom. Po vybudení návestidla nadradeným systémom požiadavkou na zmenu návestného znaku, sa vykoná kontrola podmienok na zmenu návestného znaku. V prípade pozitívneho výsledku kontroly, sú na návestidle rozsvietené všetky potrebné svetlá pre zobrazenie požadovaného návestného znaku.

Koľajový úsek nemá žiadne budiace vstupy z pohľadu nadradeného riadiaceho systému. Hlavnou úlohou tohto prvku je nepretržité informovanie nadradeného riadiaceho systému o voľnosti, resp. obsadenosti daného koľajového úseku.

Priecestia sú navrhnuté tak, aby dokázali pracovať aj bez povelov z vyšších riadiacich systémov. Obvod priecestia je tvorený dvomi približovacími a jedným anulačným úsekom. Po obsadení približovacieho úseku, je vydaný príkaz na spustenie svetelnej výstrahy. Po uplynutí sklápacieho času sú spustené závory. Po prejazde koľajového vozidla a uvoľnení anulačného úseku, je vydaný povel na ukončenie výstrahy a zdvihnutie závor. Automatická činnosť priecestí je doplnená možnosťou manuálneho ovládania priecestia pracovníkom v najbližšej obsadenej dopravni.

4.2 Návrh konceptu komunikačného protokolu

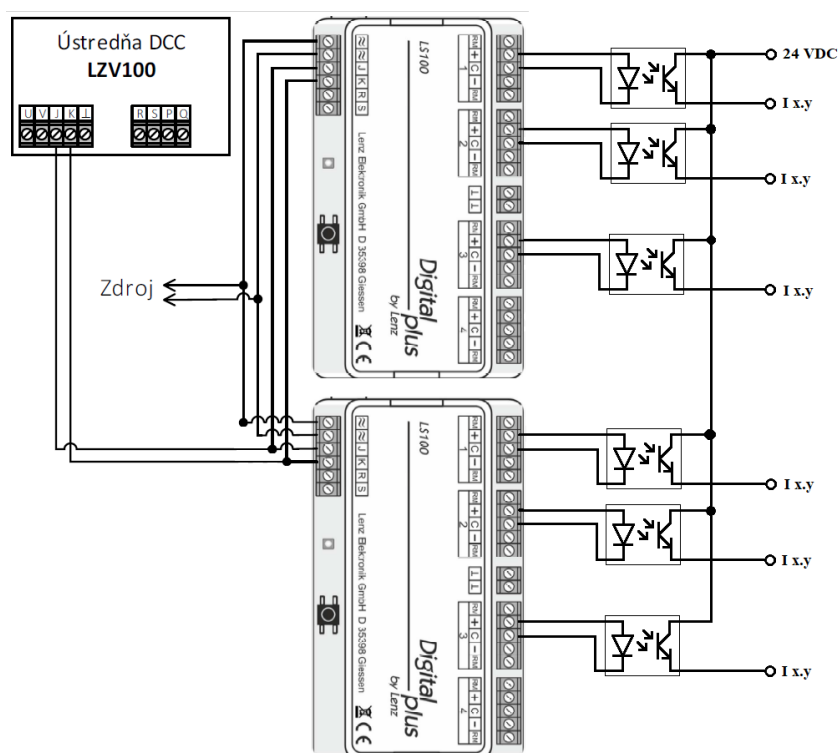
Pre správnu funkčnosť oboch systémov (ESA-44 a ILTIS-N), je nevyhnutné poznať ich aktuálny stav. V prípade výpadku jedného zo systémov je potrebné odovzdať túto informáciu druhému systému, aby bolo zabezpečené blokovanie jazdy vlaku do úseku, v ktorom menované systémy hraničia. Sledovanie aktívneho stavu je zabezpečené cyklickou komunikáciou riadiacich systémov s rozhraním. Od frekvencie odosielaných

správ sú v rozhraní vypočítavané časové pečiatky oboch systémov. V prípade uplynutia viac ako 15 sekúnd od poslednej aktualizácie údajov riadiacim systémom, by mal byť daný systém prehlásený za neaktívny. Druhý systém by mal na základe tejto informácie vyvodiť také opatrenia, aby bolo znemožnené povolenie jazdy vlaku do medzistaničného úseku Lužianky – Zbehy.

4.2.1 Návrhy komunikácie

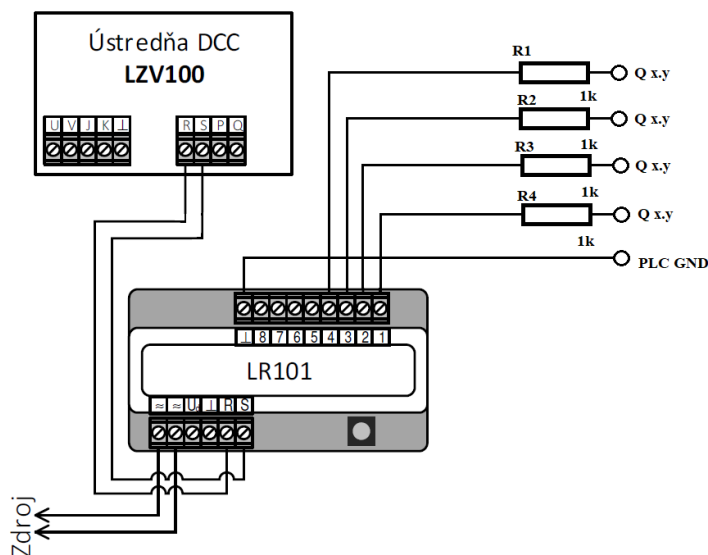
Najjednoduchším spôsobom vytvorenia komunikačnej väzby medzi našimi riadiacimi systémami je využitie hardvérových modulov a princípov, implementovaných na koľajisku. Simulačný model systému ESA-44 je prispôbený tak, aby dokázal komunikovať s digitálnymi modulmi firmy LENZ. Pre riadenie trate č. 1 na modelovom koľajisku boli použité digitálne dekodéry LENZ LS-100. Hlásenie o voľnosti koľajových úsekov je zabezpečené modulmi spätného hlásenia LR-101.

Dekodér LENZ LS-100 poskytuje štyri výstupy vo forme polovodičových relé. Na výstupe relé je spínané napätie 12 VDC pre prestavníky a návěstidlá. Dekodér LS-100 má slúžiť na prenos informácií z ESA-44 do SIMIS W. Vzhľadom na rozdiel napäťových úrovní dekodéra a PLC je potrebné vytvoriť napäťové prispôbenie pomocou optočlenov PC-817, ktoré budú zároveň slúžiť ako galvanické oddelenie oboch riadiacich systémov (Obr. 5.1).



Obr. 4.1 Schéma pripojenia dekodérov LS-100 k PLC S7-1200

Modul spätného hlásenia LR-101 obsahuje osem vstupov, ktoré sú primárne určené pre pripojenie štyroch dvojkanálových prúdových snímačov LB-100 slúžiacich ako detektory voľnosti traťových úsekov. Aj v tomto prípade je potrebné vytvoriť napäťové prispôsobenie medzi výstupným napätím PLC (24 VDC) a vstupným napätím modulu (5 VDC). V tomto prípade postačuje vloženie predradného rezistoru medzi digitálny modul LR-101 a PLC (Obr. 4.2).



Obr. 4.2 Schéma pripojenia modulu LR-101 k PLC S7-1200

Pre realizáciu navrhovaného spôsobu komunikácie sú potrebné dva moduly LS-100 a jeden modul LR-101. Pripojenie jednotlivých vstupov a výstupov modulov s PLC je uvedené v tab. 4.1.

tab. 4.1 Prepojenie vstupov a výstupov PLC s dekodérmi

Signál	Príjemca	Modul	Kontakt
voľnosť medzistaničného úseku	SIMIS W	LS-100.2	1+
žiadosť o TS	SIMIS W	LS-100.1	1+
	ESA-44	LR-101	1
udelenie TS	SIMIS W	LS-100.1	2+
	ESA-44	LR-101	2
smer TS	SIMIS W	LS-100.1	3+
predhláška	SIMIS W	LS-100.2	2+
	ESA-44	LR-101	3
odhláška	SIMIS W	LS-100.2	3+
	ESA-44	LR-101	4

Uvedené riešenie sa ukázalo ako nevyhovujúce, pretože v laboratóriu sa momentálne nenachádzajú žiadne voľné dekodéry, ktoré by sme mohli použiť. Ďalším nedostatkom je potreba doplniť programové vybavenie modelu stavadla ESA-44, ktoré by doplnilo funkcionality komunikácie s iným stavadlom prostredníctvom DCC systému Lenz.

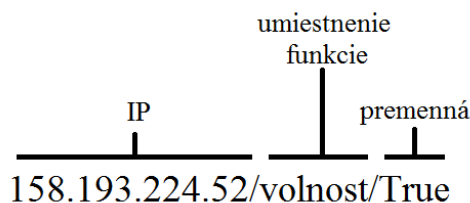
Zároveň sme narazili na problém s kontrolnými časovými pečiatkami systémov. Pri implementácii by bolo potrebné realizovať výpočet časových pečiatok na oboch stranách nezávisle na rôznych strojoch, čo predstavuje riziko v podobe rozdielnych procesorových časov PLC a stolového PC s ESA-44. Mohlo by dôjsť k zlej interpretácii časových pečiatok.

Druhým možným riešením je využitie komunikačného rozhrania REST API medzi PC aplikáciami systému ILTIS-N a PLC SIMATIC S7-1200. Toto riešenie je jednoduchšie z hardvérovej stránky, pretože nie je potrebné navrhnuť nové komunikačné rozhranie. Stolový PC, na ktorom by mal byť spúšťaný systém ESA-44, je už pripojený do LAN siete v laboratóriu. Tento spôsob ponúka aj riešenie problému s prenosom časových pečiatok pri predošlom riešení. Aby sa predišlo kolíziám časových pečiatok pre rôzne procesorové časy strojov, tak sa tieto pečiatky budú vytvárať a spracovávať v rámci REST API rozhrania. Realizácia tohto spôsobu vyžaduje návrh komunikačného protokolu medzi systémom ESA-44 a REST API rozhraním. Po zvážení oboch prístupov, sme sa rozhodli pre druhé riešenie využívajúce už existujúce REST API rozhranie.

4.3 Návrh a realizácia komunikačného protokolu

Základom komunikačného protokolu sú štandardné dopyty typu GET a PUT pre naše REST API rozhranie medzi PC a PLC. Vytvorili sme sadu dopytov dedikovaných pre systém ESA-44 na čítanie a zápis potrebných dát. V tejto sade sa nachádzajú dopyty, umožňujúce systému ESA-44 pracovať s traťovým súhlasom v rozsahu žiadania a udelenia traťového súhlasu a informovať systém SIMIS W o postavenej odchodovej ceste zo stanice Lužianky. Pretože detekciu voľnosti medzistaničného úseku Lužianky – Zbehy obstaráva systém ESA-44, do balíka dopytov sme doplnili dopyt slúžiaci na informovanie systému SIMIS W o stave daného medzistaničného úseku. Tiež sme sem zahrnuli dopyt, poskytujúci aktuálne dáta pre systém ESA-44 v rozsahu informácií o traťovom súhlase, predhlášok a odhlášok zo stanice Zbehy, informácií o blokovej podmienke a informácií o platnosti časovej pečiatky systému SIMIS W.

Základom komunikačného rozhrania sú dopyty vo forme URL adresy, obsahujúcej IP adresu stolového PC, so spusteným REST API rozhraním a umiestnenie volanej funkcie v rozhraní (napríklad pri volaní funkcie typu GET). Podľa potreby môže byť URL adresa doplnená jedným alebo viacerými premennými v prípade, kedy je v rozhraní volaná funkcia typu PUT (pri zápise dát do rozhrania) (Obr. 4.3). Komunikačný protokol obsahuje jednu funkciu, slúžiacu na poskytovanie nových dát systému ESA-44. Táto funkcia je navrhnutá pre cyklické volanie, pretože od frekvencie jej volania sa odvíja výpočet časovej pečiatky systému ESA-44.



Obr. 4.3 Príklad URL adresy

Podrobnejší rozbor jednotlivých dopytov sa nachádza v kapitole 7.1. Kompletná špecifikácia komunikačného protokolu pre systém ESA-44, obsahujúca podrobný opis všetkých funkcií, vnútornú štruktúru dopytov, povolené dátové typy parametrov a opis spôsobu plnenia dát do dopytov, sa nachádza v prílohách na priloženom CD.

5 HARDVÉROVÉ KOMPONENTY RIADIACEHO SYSTÉMU

Po analýze aktuálneho stavu modelového koľajiska sme sa rozhodli zachovať jeho aktuálny stav a naše riešenie naviazať na existujúci hardvér. Pre prenos informácií medzi vonkajšími prvkami trate (návestidlá, výhybky a koľajové obvody) sme využili aktuálne implementované moduly, opísané v kapitole 1.2.2. Okrem toho bola doplnená väzba medzi elektromotorickými prestavníkmi a PLC určená na snímanie koncových polôh výmen.

Základným prvkom riadiacej časti je PLC SIMATIC S7-1200 spoločnosti Siemens. Architektúra tohto PLC je zložená z procesorového modulu CPU 1214C DC/DC/DC doplneného jedným výstupným modulom SM 1222. K PLC je pripojený vzdialený ostrov vstupných a výstupných modulov. Komunikáciu ostrova s PLC zabezpečuje modul ET 200SP pomocou linky Profinet. Pretože nie je možné prepojenie rozhrania koľajových obvodov s PLC S7-1200, zahrnuli sme do hardvérovej výbavy aj PLC S7-300, ktoré sa v súčasnosti na koľajisku nachádza. Informácie z koľajových obvodov budú predspracované v PLC S7-300 a následne odoslané do PLC S7-1200. Pretože komunikačné rozhrania oboch PLC nie sú kompatibilné, rozhodli sme sa pre priame prepojenie výstupov jedného PLC a vstupov druhého PLC tak, ako to bolo realizované v bakalárskej práci [3]. Podrobný zoznam modulov sa nachádza v tab. 5.1 a tab. 5.2. Obr. 5.1 zobrazuje spôsob prepojenia jednotlivých komponentov riadiaceho systému.

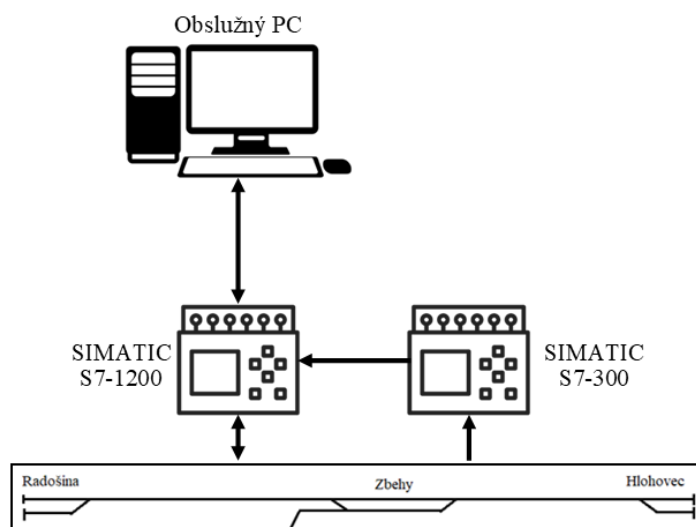
tab. 5.1 Zoznam použitých modulov pre PLC S7-1200

Funkcia modulu	Označenie	Sériové číslo	ks
CPU	CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0	1
komunikačný modul	ET 200SP	6ES7155-6AR00-0AN0	1
vstupný modul	ET 200SP DI 16X24VDC/0,5A	6ES7131-6BH00-0BA0	2
výstupný modul	SM1222	6ES7222-1BH32-0XB0	1
	ET 200SP DQ 16X24VDC/0,5A	6ES7132-6BH00-0AA0	8
päťica s novou potenciálovou skupinou	ET 200SP BaseUnit BU15-16+A0+2D	6ES7193-6BP00-0DA0	1
päťica s rovnakou potenciálovou skupinou		6ES7193-6BP00-0BA0	9

tab. 5.2 Zoznam použitých modulov pre PLC S7-300

Funkcia modulu	Označenie	Sériové číslo	ks
CPU	CPU 313C-2DP	6ES7313-6CG04-0AB0	1
I/O modul	SM 323	6ES7323-1BL00-0AA0	1
výstupný modul	SM 322	6ES7322-1BH01-0AA0	1

Medzi hardvérové komponenty môžeme zaradiť aj stolné počítače typu PC, na ktorých sa spúšťajú aplikácie jednotlivých obslužných pracovísk. Pre tieto účely sme využili počítače, nachádzajúce v laboratóriu AB205. Na prepojenie počítačov s PLC sme využili lokálnu internetovú sieť v laboratóriu. Podrobný opis architektúry oboch PLC sa nachádza v prílohe (Príloha D).



Obr. 5.1 Architektúra hardvérovej časti riadiaceho systému

6 NÁVRH GRAFICKEJ APLIKÁCIE

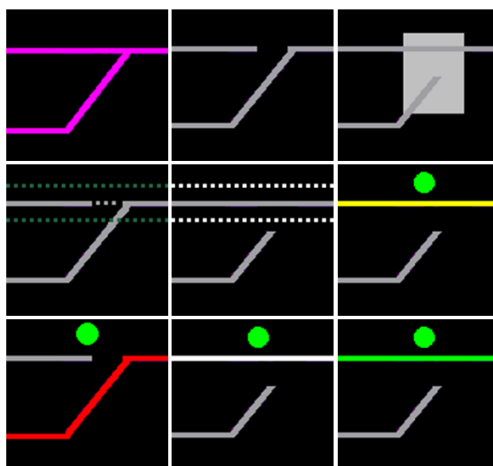
Pre vývoj dispečerskej aplikácie sme sa rozhodli použiť programovací jazyk Python. Vybrali sme si ho pre jeho jednoduchosť pri práci so serverom a prenosom dát medzi klientmi. Okrem toho sú na internete voľne dostupné knižnice, slúžiace na nadviazanie komunikácie medzi PC aplikáciou a PLC firmy Siemens. Pre dizajn a spracovanie funkčných algoritmov sme použili vývojové prostredie Qt Creator. Tento nástroj je pomerne jednoduchý, obsahuje podporu pre PySide6 a návrh grafickej časti aplikácie bol vďaka predpripraveným prvkom intuitívny.

6.1 Jazyk Python

Programovací jazyk Python vznikol v roku 1991. Odvtedy postupne naberá na popularite. Python je primárne určený na vývoj webových aplikácií, hlavne na strane serverov, všeobecného softvéru, matematických aplikácií a skriptovanie. Je vhodný pre prácu s big data, a rýchle prototypovanie. Podporuje rôzne platformy (Windows, MAC, Linux, Raspberry PI), vďaka čomu je softvér ľahko prenositeľný. Oproti iným jazykom (C++, Java, ...) má zjednodušený syntax, ale dlhší vykonávací čas programu. Jeho ďalšou výhodou je jednoduchá implementácia a ovládanie asynchrónnych metód a viacvláknového spracovania [10].

6.2 Symboly prvkov

Dôležitým krokom pri vývoji našej aplikácie bol výber a vytvorenie symbolov zobrazovania jednotlivých prvkov. Symboly prvkov v koľajisku zobrazované na obslužnom pracovisku, sme vytvorili v súlade s požiadavkami predpisu ŽSR D 101 / T 101 príloha 8c [9]. Vyberali sme iba symboly pre prvky, ktoré sa reálne nachádzajú na našej modelovej železnici t.j. symboly pre traťové úseky, výhybky, návěstidlá a prvky spojené s organizáciou dopravných procesov. Na základe veľkosti vizualizácie pre dispečerské pracovisko sme stanovili vhodné rozmery jednotlivých symbolov. Následne sme v grafickom editore vybrané symboly upravili pre použitie v našej vizualizácii (zmena rozmerov, korekcia farieb). Pre každý z prvkov sme vytvorili symboly, potrebné pre správne zobrazenie všetkých prevádzkových stavov. Každý symbol sme uložili ako bitovú mapu s rozlíšením 256 farieb a roztriedili ich pre zjednodušenie prístupu. V adresári aplikácie sme vytvorili adresár so zložkami, obsahujúcimi nami vytvorené symboly. Každý prvok vizualizácie (vchodové návěstidlá, odchodové návěstidlá, zriaďovacie návěstidlá, výhybky, koľaje..) má priradenú jednu zložku so symbolmi.



Obr. 6.1 Ukážka zobrazenia symbolov vybraných prevádzkových stavov výhybky

6.3 Pracoviská lokálnej a vzdialenej obsluhy

Pracoviská lokálnej obsluhy sú pre každú stanicu na našej trati modelované na základe funkčného správania obslužného pracoviska ILTIS-N spoločnosti Siemens Mobility v podobe, v akej je používané v sieti ŽSR. Vzhľadom na veľmi jednoduchú topológiu staníc sú aj jednotlivé aplikácie na prvý pohľad relatívne prázdne a jednoduché. Základným prvkom vizualizácie je reliéf stanice, ktorý obsahuje zobrazenie staničných prvkov (návestidlá, výhybky, koľajové úseky). Vizualizáciu sme doplnili ďalšími prvkami tak, aby čo najviac pripomínala skutočný systém ILTIS-N. V hornej časti obrazovky sme vytvorili výberovú lištu spolu so zobrazením aktuálneho času a dátumu. Do lišty sme pridali aj vlastné prvky slúžiace na nadviazanie komunikácie s REST API rozhraním. Je to kontextové okno, obsahujúce textové pole pre zadanie IP adresy stolového PC, so spusteným REST API rozhraním a tlačidlá „Connect“ a „Disconnect“ na ovládanie komunikácie. Väčšina prvkov v lište nefunguje, pretože určité funkcie (zobrazenie GVD, tlač dokumentov, prihlasovanie obsluhy ...) nie sú v našom modeli potrebné. V dolnej časti sme vytvorili panel správcu hlásení, slúžiaci na zobrazenie chybových a systémových hlásení. Na našej trati sme sa rozhodli modelovať správanie automatického bloku typu AB3-74, v ktorom sú návestidlá platné pre smer, v ktorom nie je udelený traťový súhlas, zhasnuté. Vo vizualizácii sme danú skutočnosť naznačili tak, že zhasnuté oddielové návestidlá sa budú zobrazovať sivou farbou (Obr. 6.2). Ukážka vizualizácií oboch typov pracovísk sa nachádza v prílohe (Príloha F).



Obr. 6.2 Vizualizácia prevádzkových stavov trojznakového automatického bloku

Počas prvotného vývoja aplikácie sme uvažovali rôzne spôsoby ovládania vizualizácie. Jednou z možností bolo využitie modulu na detekciu miesta kliknutia vo vizualizácii.

V tomto prípade by všetky symboly boli definované ako prvky typu „Label“. Vývojové prostredie QT Creator umožňuje do prvku „Label“ vložiť okrem textu aj grafický prvok, čo sa na náš účel hodilo. Nevýhodou takto vytvorenej vizualizácie, je potreba implementovania algoritmu na detekciu miesta kliknutia myšou na obrazovke, ktorý detegované súradnice odovzdá algoritmu vyhodnocujúcemu, ktorý grafický prvok sa nachádza v mieste kliknutia. Tento postup sa nám javil ako príliš zložitý a náročný na výpočtový výkon počítača.

Preto sme hľadali použitie iného spôsobu realizácie obslužného elementu v obraze koľajiska, ktorý by nahradil prvok typu „Label“. Pre naše požiadavky sa ako najvhodnejšia možnosť javilo použitie prvku „Button“. Tento prvok, rovnako ako „Label“, podporuje vloženie grafického obrázku. Zároveň odpadá potreba zložitého algoritmu na detekciu miesta kliknutia, keďže prvok „Button“ po kliknutí vygeneruje správu, ktorú v aplikácii vieme jednoducho prepojiť s výkonnou akciou. Okrem toho signál stlačeného tlačidla vieme použiť na zavolanie konkrétnej metódy.

Vo vizualizácii sa tiež nachádzajú aj iné prvky, ktoré nevykonávajú žiadnu funkciu. Sú to symboly koncov kusých koľají v úvratových staniaciach, názvy staníc alebo prípadné označenie traťových koľají na viackoľajnej trati. Pretože nie je potrebné, aby po kliknutí na symbol bola volaná nejaká metóda, rozhodli sme sa na zobrazenie daného symbolu použiť prvok „Label“.

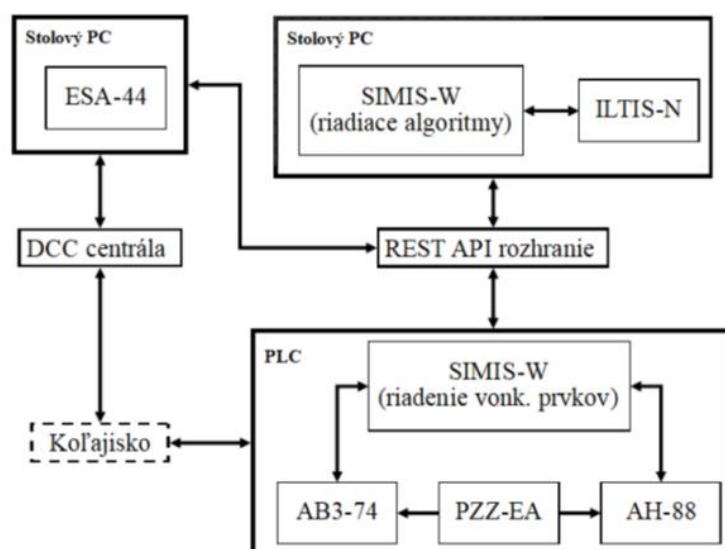
Pretože systém ILTIS N je komplexný nástroj s mnohými funkciami, rozhodli sme sa pre návrh zjednodušeného modelu. Vizualizácia, ako aj jej ovládanie je prispôsobené tak, aby sa náš produkt čo najviac podobal skutočnému systému. Pri návrhu a implementácii sme sa držali platných predpisov ŽSR, ktoré špecifikujú požiadavky na spôsob ovládania a zobrazovania prevádzkových stavov zariadenia [7][8].

7 PROGRAMOVÁ IMPLEMENTÁCIA FUNKČNÝCH ALGORITMOV

Pri návrhu funkčných algoritmov sme zvažovali niekoľko možností architektúry softvéru radiaceho systému. Jednou z možností, ktorými sme sa zaoberali bol presun celej logiky do PC aplikácie a PLC by slúžilo ako rozhranie medzi vonkajšími prvkami a pracoviskom obsluhy. Vzhľadom k faktu, že sme sa rozhodli použiť dve vzájomne prepojené PLC, zhodli sme sa na tom, že logika zabezpečovacích systémov bude rozdelená na niekoľko vrstiev. Najvyššia je vrstva ovládania a zobrazovania prevádzkových stavov, implementovaná v PC aplikáciách. Vrstva riadenia na úrovni zariadenia, realizujúca dopravno-bezpečnostné algoritmy zabezpečovacích systémov, je rozdelená medzi PC

aplikácie a PLC. V PC aplikáciách sú zahrnuté algoritmy staničných zabezpečovacích systémov a v PLC SIMATIC S7-1200 sú zahrnuté algoritmy pre traťové a priecestné zabezpečovacie zariadenia. V PLC sa tiež nachádza vrstva riadenia prvkov staničných zabezpečovacích zariadení, riadiaca prestavníky, svetlá návěstídiel a sníma stavy prestavníkov a koľajových úsekov.

Po návrhu správania zabezpečovacích zariadení podľa vytvorenej špecifikácie sme začali postupne vytvárať jednotlivé algoritmy zabezpečovacích zariadení. Každý algoritmus sme navrhli a otestovali najprv samostatne. Po validácii jednotlivých algoritmov sme medzi nimi vytvorili potrebné dátové prepojenia tak, aby bola zabezpečená spoľahlivá komunikácia jednotlivých druhov zabezpečovacích zariadení. Obr. 7.1 znázorňuje principiálnu schému prepojenia jednotlivých zariadení v nami vytvorenej komunikačnej sieti.



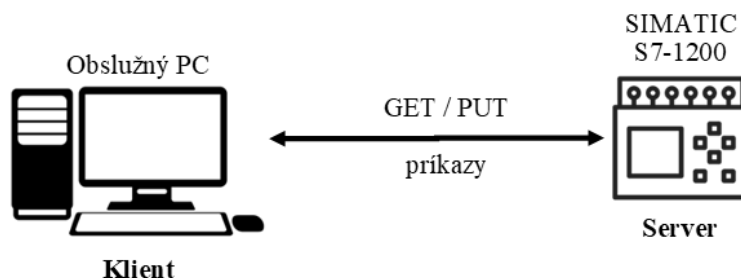
Obr. 7.1 Architektúra softvérovej časti riadiaceho systému

Po implementácii a otestovaní funkčnosti všetkých algoritmov sme sa rozhodli pomocou nástroja „Pyinstaller“ konvertovať zdrojové kódy do jedného spustiteľného .exe súboru. Vďaka tomu sme dosiahli prenositeľnosť jednotlivých aplikácií ILTIS-N medzi stolovými PC v laboratóriu. Odpadla potreba vopred na PC inštalovať Python knižnice, pretože všetky potrebné knižnice pre beh aplikácie sú zbalené spolu so zdrojovými kódmi v .exe súbore. Spolu s týmto súborom je potreba distribuovať aj zložku „img“, obsahujúcu jednotlivé bitové mapy elementov vizualizácie. Tým sme dosiahli splnenie požiadaviek T1.1 a T1 (tab. 2.1).

7.1 REST API rozhranie

Komunikácia medzi jednotlivými PC aplikáciami a PLC je zabezpečená pomocou REST API protokolu. Pre jeho použitie sme sa rozhodli preto, lebo pomocou neho dokážeme veľmi jednoducho čítať dáta z PLC, ako aj dáta odosielať do PLC. Na skratku REST sa často odkazuje ako na komunikačný protokol. V skutočnosti sa jedná o architektonický štýl. Pod API rozumieme súbor pravidiel, ktoré definujú ako zariadenia a aplikácie medzi sebou komunikujú. Pri použití REST API sa na komunikáciu najčastejšie využíva HTTP protokol. Jednotlivé aplikácie klienta a serveru o sebe prakticky nič nevedia, pre spojenie so serverom klientovi stačí poznať jeho adresu. Napriek tomu, že REST API nemá definované vlastné dátové formáty, najčastejšie sa spája s formátmi JSON a XML. Pre komunikáciu používa štyri základné príkazy:

1. GET – slúži na načítanie informácie zo záznamu,
2. POST – slúži na vytvorenie nového záznamu,
3. PUT – slúži na zmenu informácie v zázname,
4. DELETE – zmaže vybraný záznam.



Obr. 7.2 Znáznornenie sieťovej komunikácie

Vzhľadom na fakt, že veľká časť komunikácie prebieha medzi aplikáciou a PLC, architektúru našej siete sme navrhli tak, že PLC sme zadefinovali ako server a grafické aplikácie k nemu pristupujú ako klienti prostredníctvom REST API (Obr. 7.2). Hlavným účelom rozhrania je sprostredkovanie komunikácie medzi stolovými riadiacimi PC aplikáciami a PLC. Zároveň ukladá zdieľané dáta dôležité pre správnu kooperáciu jednotlivých aplikácií. Týka sa to hlavne informácií o postavených odchodových cestách, stavoch traťových súhlasov a spôsobe riadenia staníc (lokálne/diaľkovo). Každý dopyt má priradenú funkciu, vykonávajúcu stanovenú činnosť, napríklad načítanie alebo zápis dát, výpočet parametrov a podobne. Jednotlivé funkcie sme rozdelili do niekoľkých skupín podľa toho, aké majú zameranie [16][17].

7.1.1 Štruktúra zdrojových kódov

Zdrojové kódy REST API komunikačného rozhrania sú rozdelené do dvoch súborov. Prvý, pomenovaný „REST.py“, obsahuje všetky funkcie, potrebné pre zaistenie komunikácie a prenosu dát medzi PLC a PC aplikáciami modelu ILTIS-N. Bližší opis jednotlivých funkcií sa nachádza v kapitolách 7.1.2 až 7.1.6.

Súbor „S71200.py“ obsahuje metódy pracujúce s knižnicou „snap7“. Táto knižnica slúži na nadviazanie komunikácie medzi PC aplikáciou a PLC SIMATIC S7-1200 alebo S7-1500 spoločnosti Siemens. Samotná knižnica ponúka široké spektrum funkcií, ktorými dokážeme čítať a zapisovať rôzne údaje do PLC, od bežných vstupov a výstupov, až po diagnostické dáta. Použitie samotnej knižnice je dosť zložité, vzhľadom na náročnú predprípravu dát na zápis, prípadne spracovanie prijatých dát od PLC. Na internete sa nám podarilo nájsť jednoduchý kód, uľahčujúci prácu s knižnicou „snap7“. Spomenutú triedu sme pomenovali „S71200“. trieda sa nachádza práve v tomto súbore.

Pre naše potreby využívame dve metódy: metódu „getMem“ na čítanie dát a metódu „writeMem“ pre zápis dát. Obe pracujú približne na rovnakom princípe. Vstupnými dátami sú informácie o adrese, s ktorou chceme pracovať v PLC. V prípade zápisu sa medzi vstupnými dátami nachádza aj hodnota, zapisovaná do pamäťového priestoru PLC.

Príklad volania funkcie getMem v prípade, kedy chceme načítať hodnotu výstupného bitu PLC, Q1.4: „S71200.getMem(QX1.4)“. Atribút funkcie obsahuje dodatočnú informáciu o dátovom type, ktorý čítame alebo zapisujeme „X“ použijeme v prípade práce s bitom, „B“ používame pri čítaní celého bajtu a „W“ použijeme pri práci s dátovým typom „word“. Následne si metóda načíta zo zadanej adresy potrebné parametre ako číslo bajtu a bitu, s ktorým chceme pracovať (v našom prípade QX1.4 to bude 1. bajt a 4. bit). V prípade zápisu do PLC nasleduje vloženie nami určenej hodnoty do pamäťového priestoru PLC. V prípade čítania dát metóda načíta dáta z PLC pomocou vstavanej metódy „read_area“ v knižnici „snap7“. Vzhľadom na fakt, že z PLC je možné čítať dáta iba po bajtoch, posledným krokom je úprava získaných dát na formát, ktorý užívateľ požaduje (bit, celé číslo...) [18].

Zdrojový kód obsahuje aj dva súbory „RiseEdge.py“ a „FallEdge.py“, obsahujúce pomocné triedy. Obe triedy obsahujú jednoduchú metódu, slúžiacu pre detekciu prechodov signálov z log. 0 do log. 1, resp. z log. 1 do log. 0.

7.1.2 Funkcie pre zápis do PLC

Do tejto kategórie sme zaradili všetky funkcie, odosielajúce nové dáta do PLC. Funkcia „*writeVyhodka*“ odosiela do PLC pokyny na prestavovanie výmen v staniciach. Jej vstupnými parametrami sú názov výmeny a požadovaný smer prestavenia. Funkcie „*writeNavestidlo*“ a „*writeOchrDr*“ súvisia s riadením návestidiel. Prvá z menovaných funkcií v PLC aktualizuje návestné znaky zobrazené na návestidlách. Druhá poskytuje doplňujúce informácie o tom či je vydaný príkaz na postavenie vchodovej cesty s ochrannou dráhou alebo bez ochrannej dráhy, pre zabezpečenie svietenia správneho návestného znaku na vchodovom návestidle stanice. Úlohou funkcie „*writePredhl*“ je prenos predhlášok zo staníc Hlohovec a Zbehy pre oddielové návestidlá hradla Alekšince. Poslednou funkciou v tejto kategórii je funkcia pre zadávanie príkazov manuálnej obsluhy priecestných zabezpečovacích zariadení „*writePriecestie*“.

7.1.3 Funkcia pre načítanie dát do PC aplikácie

Funkcia „*read*“ je rozsahom najväčšou funkciou celého rozhrania podľa prenášaných informácií. Jej úlohou je zber a spracovanie traťových dát z PLC, dát od systému ESA-44 a zdieľaných dát nachádzajúcich sa v komunikačnom rozhraní.

Načítavanie traťových dát (informácií o voľnosti traťových úsekov) prebieha v niekoľkých krokoch. Najprv sú z PLC načítané surové dáta v celočíselnom formáte „byte“. Pre naše účely však potrebujeme poznať hodnotu každého bitu separátne, preto sú dáta v ďalšom kroku konvertované do dátovej štruktúry „bytearray“. Ide o binárny zápis celého čísla.

Pre zrýchlenie komunikácie sme sa rozhodli dáta načítavať vo formáte „byte“. Pri testovacích pokusoch o načítanie dát z PLC priamo v bitovej podobe sme zistili, že táto metóda trvá priemerne osem až desaťkrát dlhšie, ako pri načítavaní v celočíselnom formáte. Táto skutočnosť sa začala prejavovať ako nepravidelná funkcia aplikácie, s neprijateľne dlhou časovou odozvou medzi zmenou stavu vonkajšieho prvku v modelovom koľajisku a jej zobrazením v obraze koľajiska v PC aplikácii.

Nasleduje spracovanie informácií o voľnosti medzistaničných úsekov. Pre zistenie voľnosti medzistaničných úsekov sú potrebné dáta o voľnosti traťových oddielov a dáta o postavených odchodových cestách do daného úseku. Pre zber a vyhodnocovanie spomenutých dát sme vytvorili algoritmus preklápacieho RS obvodu a pomocný algoritmus pre detekciu nábežnej hrany signálu. Pomocným algoritmom skúmame časový okamih, kedy je úsek po obsadení koľajovým vozidlom opätovne uvoľnený. Po vyhodnotení uvoľnenia medzistaničného úseku, môžeme zrušiť príznak postavenej odchodovej cesty do daného úseku. Princíp funkčnosti preklápacieho RS obvodu

je nasledovný: ak máme informáciu o voľnosti úseku a nie sú postavené žiadne odchodové cesty do tohto úseku, prípadne máme odhlášku za posledným vlakom, môže byť úsek považovaný za voľný. Voľnosť úseku sa ruší jeho obsadením, postavením odchodovej cesty alebo aktívnou predhláškou pred vlakom.

Nakoniec sú zo zdieľaných dát načítané informácie o aktívnych predhláškach, odhláškach a časovej pečiatke systému ESA-44. Všetky nadobudnuté dáta sú zbalené do JSON formátu a ako návratová hodnota funkcie odoslané konkrétnej PC aplikácii. Tá si z dátovej štruktúry importuje iba pre ňu potrebné informácie. Uvažovali sme aj nad vytvorením špecifických správ pre každú aplikáciu, ale z praktických dôvodov sme sa rozhodli pre jednotné riešenie, s ktorým budú schopné pracovať všetky PC aplikácie.

7.1.4 Funkcie pre prácu so zdieľanými dátami

Do tejto skupiny patria funkcie, pracujúce s dátami uloženými v komunikačnom rozhraní. Zdieľané dáta slúžia pre správnu funkčnosť jednotlivých obslužných pracovísk ILTIS-N. Pri návrhu rozhrania sme najprv uvažovali s lokálnym konceptom dát. Všetky dáta týkajúce sa traťových súhlasov, súhlasov na obsluhu a podobne, meniace sa interakciou jednotlivých aplikácií, budú vytvorené lokálne na úrovni aplikácie. REST API rozhranie by slúžilo iba na prenos požiadaviek medzi aplikáciami a dáta by sa museli meniť v komunikujúcich aplikáciách nezávisle na sebe. Toto riešenie prináša potenciálny problém straty synchronizácie dát pri výpadku jednej z aplikácií. Po reštarte by aplikácia prešla do základného stavu, nie nutne korešpondujúceho s aktuálnym stavom na trati. Problém so synchronizáciou sme odstránili centralizovaním dát v komunikačnom rozhraní. Jednotlivé aplikácie menia zdieľané dáta pomocou dopytov typu PUT. Načítavanie nových dát do PC aplikácií je integrované vo funkcii „read“, viď kap. 7.1.3.

Reprezentantmi funkcií pracujúcich so zdieľanými dátami sú funkcie „writeZUS“ a „writeUTS“, slúžiace pre prácu s traťovým súhlasom a funkcia „writeZiadRiad“ spolu s funkciou „writeUdelRiad“, pre zmenu oprávnenia na obsluhu stanice z konkrétneho pracoviska obsluhy (lokálna/diaľková obsluha). Ďalej sem patrí funkcia „writeOdchod“, určená pre zápis príznaku postavenej odchodovej cesty do zdieľaných dát a funkcia „writePredhl“, slúžiaca na odoslanie predhlášky algoritmom TZZ. Túto informáciu zároveň ukladá v REST API rozhraní.

V tejto kategórii sa okrem už spomenutých funkcií nachádzajú ešte funkcie „writeCesta“ a „writeNavesť“, pre synchronizáciu postavených ciest a návestných znakov medzi PC aplikáciami. V prípade, že výpravca v stanici postaví vlakovú cestu, tieto funkcie zabezpečia, aby sa postavená cesta zobrazila aj dispečerovi trate na pracovisku

vzdialenej obsluhy aj napriek tomu, že danú stanicu neriadi. Rovnaká funkcionálnosť je zachovaná tiež v opačnom prípade diaľkovej obsluhy stanice.

7.1.5 Funkcie pre systém ESA-44

Pre vytvorenie komunikačnej väzby medzi modelmi systémov SIMIS W a ESA-44 sme sa rozhodli využiť REST API rozhranie. Táto metóda bola pre nás výhodná, pretože niektoré z potrebných dopytov už boli v rozhraní implementované, napríklad dopyty pre traťový súhlas (viď. kap. 7.4.3). Vďaka tomu stačilo do rozhrania doplniť iba zopár funkcií spracovávajúcich nové dopyty od systému ESA-44. Doplnili sme funkcie „writeCasPec“ a „writeVolnost“. Prvá z týchto funkcií slúži na zaznamenávanie poslednej aktualizácie dát v rozhraní formou časovej pečiatky. Úlohou druhej funkcie je prenos informácie o voľnosti medzistaničného úseku Lužianky – Zbehy. Okrem toho sme do rozhrania doplnili aj funkciu spätného hlásenia pre systém ESA-44 „writeESA“. Táto funkcia zabezpečuje kontrolu platnosti časových pečiatok riadiacich systémov a v návratovej hodnote poskytuje dôležité dáta pre správnu kooperáciu oboch riadiacich systémov. Obsahom spätného hlásenia sú dáta o traťovom súhlase, blokovej podmienke, časovej pečiatke systému SIMIS W a dátach spojených so stavaním odchodovej cesty v stanici Zbehy.

7.1.6 Arbitráž komunikácie REST API rozhrania s PLC

Počas vývoja komunikačného rozhrania sme narazili na problém spracovania dát v PLC. Zistili sme, že PLC SIMATIC S7-1200 dokáže počas behu operačného cyklu, v čase vyhradenom pre komunikáciu, spracovať iba jeden dopyt. V prípade snahy paralelného odosielania dát do PLC z viacerých aplikácií alebo čítania dát z PLC bola komunikácia odmietnutá z dôvodu zahlteného PLC. Vzhľadom na to, že návrh REST API rozhrania počítal so synchronným čítaním dát a asynchronným zápisom dát, bolo potrebné navrhnuť arbitrážny systém, pre zabezpečenie plynulej komunikácie REST API rozhrania s PLC. Toto je potrebné aj pre zaistenie funkčnej komunikácie v prípade, kedy budú spustené viaceré PC aplikácie. Pri rôznych časoch spustenia každej aplikácie by mohla nastať situácia, kedy sa bude viac aplikácií snažiť čítať dáta v rovnakom čase.

Pre zaistenie plynulej komunikácie sme v REST API vytvorili globálne premenné „zapis“ a „citanie“ typu „bool“. Ak je hodnota premenných True, prebieha zápis, prípadne čítanie dát. Ak majú hodnotu False, neprebieha žiadna komunikácia. Tieto premenné sú importované do všetkých funkcií REST API rozhrania, ktoré pracujú s PLC.

Po zavolaní funkcie komunikujúcej s PLC, je na začiatku behu funkcie kontrolovaná voľnosť komunikačného kanála s PLC. Ak v danom čase prebieha iná komunikácia

s PLC, funkcia sa uspí a cyklicky sa bude každých 100 ms dopytovať na voľnosť kanálu. V prípade voľného komunikačného kanála sa vykonajú špecifické príkazy v tele funkcie pre výber adresy dát v PLC, prípadne hodnoty, ktorá ma byť do PLC zapísaná. Nasleduje pokus o odoslanie požiadavky do PLC s využitím štruktúry „try...except“. Vetva „try“ obsahuje príkaz na odoslanie požiadavky do PLC. Ak by bolo odoslanie neúspešné a bola by zaznamenaná chyba typu „RuntimeError“, kanál sa uvoľní pre inú komunikáciu a funkcia sa na 100 ms uspí. Po uplynutí času sa vykoná ďalší pokus o odoslanie požiadavky. V prípade úspešnej komunikácie je beh funkcie ukončený a komunikačný kanál je uvoľnený pre ďalšiu komunikáciu. Obr. 7.3 zobrazuje príklad časti programového kódu funkcie, v ktorej sú implementované arbitrážne algoritmy.

```

260 @app.put('/predhl/{index}/{value}') #metóda pre zápis preshlášok pre AH
261 async def writePredhl(index: str, value: bool):
262     global plc
263     global zapis
264     global citanie
265
266     while zapis and citanie:
267         sleep(0.1)
268
269     if not (zapis and citanie):
270         if index == 'ZBE':
271             address = 'MX29.0'
272
273         elif index == 'HLO':
274             address = 'MX29.1'
275
276         while True: #arbitrážny cyklus
277             zapis = True
278             try:
279                 plc.writeMem(address, value)
280
281             except RuntimeError:
282                 zapis = False
283                 sleep(0.1)
284
285             else:
286                 break
287
288     zapis = False

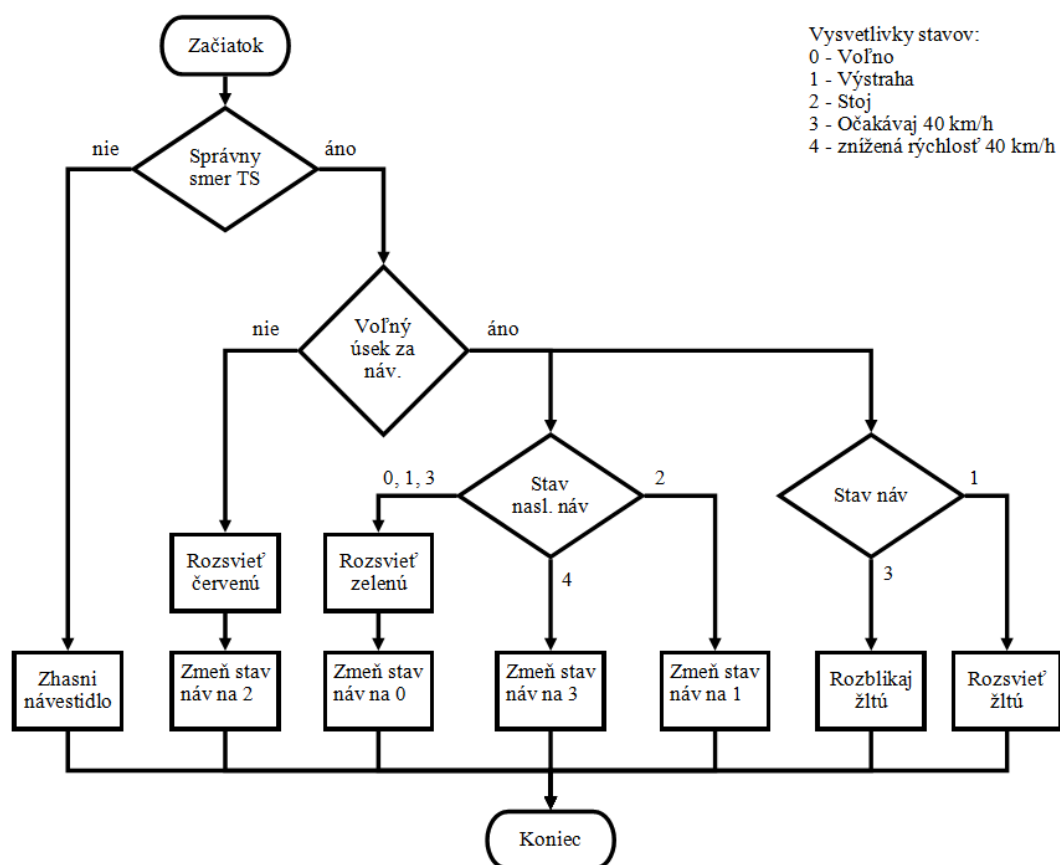
```

Obr. 7.3 Funkcia "writePredhl" s arbitrážnymi algoritmami

7.2 Traťové zabezpečovacie zariadenia

Traťové zabezpečovacie zariadenia sú tvorené dvomi funkčnými blokmi. Prvý zabezpečuje funkcie trojznakového automatického bloku. Tento funkčný blok neobsahuje stavový automat. Návestidlo je riadené sériou podmienok (viď Obr. 7.4), podľa ktorých sa vyberá vhodný návestný znak, pre zobrazenie na konkrétnom návestidle. Medzi skúmané parametre sme zaradili aktuálny smer traťového súhlasu, smer, pre ktorý platí oddielové návestidlo, voľnosť úseku za návestidlom a aktuálne zobrazený návestný znak na nasledujúcom návestidle. Rozhodli sme sa, že na našej trati budeme modelovať správanie automatického bloku typu AB3-74, v ktorom sú

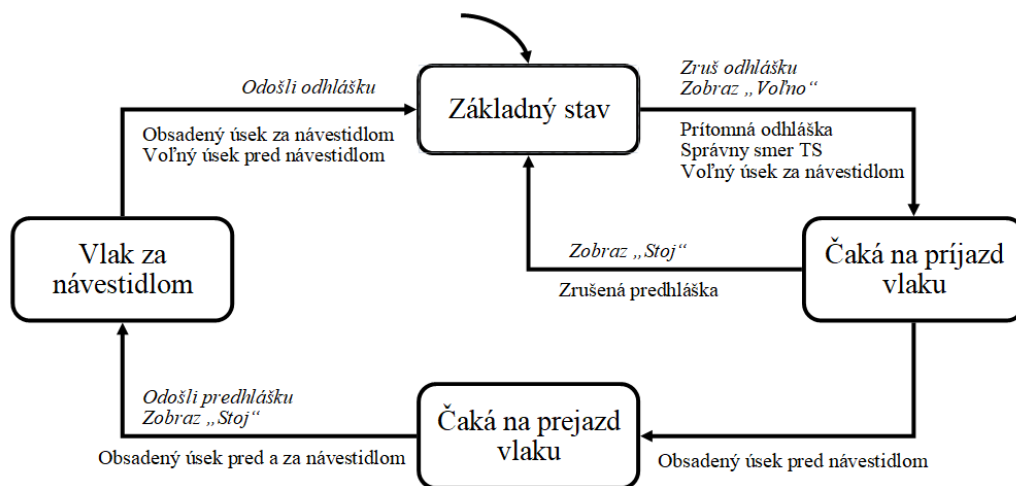
návestidlá platné pre smer, v ktorom nie je udelený traťový súhlas, zhasnuté. Pre túto činnosť využívame kombináciu statického atribútu „smer_Nav“ a premennej „Trat_suhlas“. Atribút „smer_Nav“ je zadany každému návestidlu osobitne a určuje, v akom smere je návestidlo otočené. Pre nepárne návestidlo je hodnota atribútu log. 0, pre párne návestidlo je hodnota atribútu log.1. Ak sa hodnoty atribútov „smer_Nav“ a „Trat_suhlas“ rovnajú, vieme, že návestidlo sa nachádza v smere traťového súhlasu a je potrebné na ňom rozsvietiť príslušnú návesť. Funkčná logika automatického bloku je navrhnutá v zmysle požiadaviek F3.4, F3.5 a F3.6 (tab. 2.3).



Obr. 7.4 Algoritmus automatického bloku

Funkčný blok automatického hradla (viď. Obr. 7.5) je tvorený stavovým automatom. V základnom stave, po prvotnom spustení systému, svieti na návestidle červené svetlo a je aktívna odhláška do stanice. Vstupným podnetom pre vybudenie činnosti, je príjem predhlášky. Následne sa kontrolujú podmienky na zmenu návěsti, t.j. správny smer traťového súhlasu, voľnosť úseku za návestidlom a prítomnosť odhlášky za posledným vlakom. Ak sú podmienky splnené, zruší sa odosielané odhlášky za posledným vlakom a rozsvieti sa povoľujúca návěst' na návestidle. Po postupnom obsadení úsekov pred a za návestidlom, dochádza k zmene návěsti „Voľno“ na „Stoj“ a je odoslaná predhláška na nasledujúce hradlo, prípadne do susednej stanice. Akonáhle je celý vlak v úseku

za návestidlom, prechádza stavový automat do základného stavu a odošle sa odhlášku za vlakom späť na predchádzajúce hradlo, prípadne do počiatočnej stanice. Stavový automat automatického hradla je navrhnutý v zmysle požiadaviek F3.1, F3.2 a F3.3 (tab. 2.3). V prílohách (Príloha I |) sa nachádza sekvenčný diagram činnosti automatického hradla.



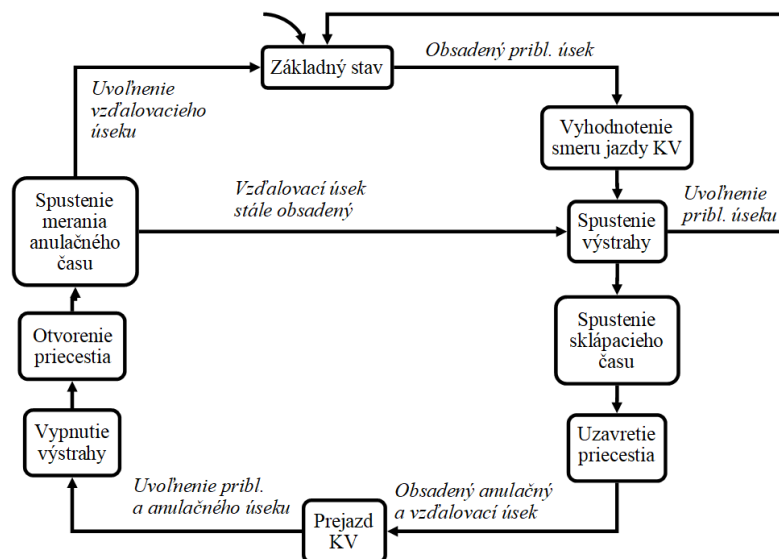
Obr. 7.5 Stavový automat automatického hradla

7.3 Priecestné zabezpečovacie zariadenia

Riadiace algoritmy priecestných zabezpečovacích zariadení sú prebrané z bakalárskej práce iného študenta [3]. Po analýze funkčných algoritmov PZZ sme zhodnotili, že v súčasnom stave vyhovujú požiadavkám F4.1, F4.2, F4.3 a F4.4 (tab. 2.4). Pôvodné algoritmy bolo potrebné upraviť tak, aby vyhovovali našim potrebám. Do funkčnej časti sme nezasahovali, úpravy sa týkali hlavne spôsobu akvizície dát o voľnosti traťových úsekov priecestí. Pôvodné algoritmy obsahovali funkčný blok, konvertujúci informácie o obsadení koľajových úsekov na výstupné dáta týkajúce sa približovacích úsekov a anulačného úseku priecestia. Pretože rovnakú činnosť už vykonáva PLC SIMATIC S7-300, dotýčny funkčný blok stratil význam a mohli sme ho z programu odstrániť. Do funkčného bloku bolo ešte potrebné implementovať väzbu na REST API rozhranie. Implementáciou tejto väzby sme splnili požiadavku F1.5 a F4.5 (tab. 2.1 a tab. 2.4).

V základnom stave je na výstražníkoch zobrazené prerušované biele svetlo. Po obsadení približovacieho úseku koľajovým vozidlom, sa spustí algoritmus pre určenie smeru jazdy koľajového vozidla vo vnorenom funkčnom bloku „*Detekcia_Smeru_Jazdy*“. Začne plynúť výstražný sklápací čas a priecestie prechádza do výstražného stavu. Po uplynutí sklápacieho času, je vydaný povel na zatvorenie závor. Algoritmus ďalej čaká na prejazd koľajového vozidla cez priecestie. Po vyhodnotení prejazdu, je vydaný povel na ukončenie výstrahy, zdvihnutie závor a spustenie merania anulačného času.

Ak sa vzdalovací úsek uvoľní pred vypršaním tohto času, priecestie prechádza do základného stavu. Je vydaný povel na zobrazenie bieleho prerušovaného svetla na výstražníkoch. Ak v čase uplynutia anulačného času je vzdalovací úsek stále obsadený, priecestie prechádza do výstražného stavu (Obr. 7.6).



Obr. 7.6 Stavový diagram činnosti priecestia

Do funkčného bloku sme doplnili jednoduchú spätnú väzbu pre grafické PC aplikácie. Spätnú väzbu tvoria tri premenné, informujúce algoritmy modelu ILTIS-N o aktuálnom stave priecestia (priecestie otvorené, priecestie uzavreté a priecestie v procese sklápania/zdvíhania závor). Tieto informácie sú potrebné pre správne zobrazenie symbolu priecestia v grafickej aplikácii. Okrem toho sme doplnili možnosť manuálneho ovládania priecestia dispečerom, prípadne výpravcom stanice. Algoritmy PZZ sa nachádzajú v prílohách (Príloha G).

7.4 Staničné zabezpečovacie zariadenia

Funkčné algoritmy modelu systému SIMIS W sa nachádzajú v súbore „SZZ.py“. Obsahom súboru je trieda „SZZ“ s funkciami základných operácií v stanici. Patrí sem:

1. Funkcia „stavanieCesty“, slúžiaca na stavanie jazdných ciest
2. Funkcia „rusenieCesty“, slúžiaca na zrušenie neprejdenej jazdnej cesty
3. Funkcia „rusenieOD“, slúžiaca na zrušenie ochrannej dráhy vlakovej cesty
4. Funkcia „prestavenieVyh“, slúžiaca na prestavovanie výmen v stanici

Pre správnu funkčnosť algoritmov, sa v triede „SZZ.py“ ukladajú dáta o type aktuálnej cesty (posunová / vlaková), informácia o aktuálnej stavanej, prípadne rušenej ceste a o potrebe zriadiť ochrannú dráhu. Okrem toho sa tu nachádza slovník s dátami

o priebehu stavania jazdnej cesty. Zjednodušené vývojové diagramy jednotlivých funkcií sa nachádzajú v prílohe (Príloha J).

7.4.1 Funkcia „stavanieCesty“

Pri zavolaní tejto funkcie algoritmami dispečerského pracoviska, si aplikácia načíta dáta o počiatočnom a koncovom návestidle, vybranom obsluhou, počas výberu jazdnej cesty. Funkcia začne prehľadávať záverovú tabuľku, až pokým nenájde zhodu s niektorou definovanou jazdnou cestou. Okrem toho sa skúma či na počiatočnom návestidle nie je aktívna privolávacia návesť. V prípade negatívneho výsledku hľadania je beh funkcie zastavený a je vydaný povel na výpis chybového hlásenia o zle zadanej jazdnej ceste. Vyhľadávacia časť funkcie je navrhnutá v zmysle požiadavky F2.1 (tab. 2.2).

Po nájdení zhody so záverovou tabuľkou funkcia skúma či užívateľ zvolil stavanie cesty s ochrannou dráhou alebo bez nej. Funkcia vykoná kontrolu, že nájdená jazdná cesta v záverovej tabuľke má definovanú ochrannú dráhu. Ak sa zistí, že vybraná cesta je bez ochrannej dráhy a užívateľ ju požaduje (prípadne naopak), funkcia sa vráti na začiatok k prehľadávaniu záverovej tabuľky. Po úspešnom dohľadaní jazdnej cesty v záverovej tabuľke nasleduje hlbšia kontrola jazdnej cesty. Týmto je splnená požiadavka F2.4 (tab. 2.2).

V nasledujúcom cykle funkcie sa skúma či nie sú niektoré z koľajových úsekov jazdnej cesty, prípadne jej ochrannej dráhy, už pod záverom inej jazdnej cesty. Na tieto účely slúži slovník „dictStavanieCesty“, ktorého premenné obsahujú informácie o aktuálnom stave stavania jazdnej cesty. Štruktúra slovníka je naznačená v tab. 7.1.

tab. 7.1 Štruktúra slovníka „dictStavanieCesty“

premenná	vysvetlenie
„stavanie“	informácia o povolení postaviť vybranú jazdnú cestu
„volnosť“	informácia o voľnosti všetkých koľajových úsekov v jazdnej ceste
„vymeny“	informácia o správnej polohe všetkých výmen v jazdnej veste

Ak sú všetky úseky mimo záveru, je v slovníku s dátami o priebehu stavania jazdnej cesty „dictStavanieCesty“ prepísaný parameter „stavanie“ na „True“. Po úspešnom vyhodnotení, pokračuje kontrola voľnosti všetkých úsekov jazdnej cesty a jej ochrannej dráhy. Prebieha rovnakým spôsobom, ako predošlá kontrola. Po vyhodnotení voľnosti úsekov sa v slovníku „dictStavanieCesty“ prepíše parameter „volnosť“ na „True“.

V prípade stavania odchodovej cesty, je potrebné zabezpečiť, aby stanica mala udelený traťový súhlas, a prvý traťový oddiel za stanicou bol voľný. Kontrola traťového súhlasu prebieha už pri výbere cieľového návestidla (kontrola je implementovaná v metóde „akciaNavestidlo“, viď. kap. 7.6.1). Ak by stanica nemala udelený traťový súhlas,

stavanie cesty je automaticky odmietnuté. Kontrola voľnosti prvého traťového oddielu prebieha bezprostredne pred začatím stavania jazdnej cesty (t.j. stav všetkých premenných v slovníku „*dictStavanieCesty*“ je True). Algoritmus zo záverovej tabuľky načíta názov prvého traťového oddielu a dopytuje sa na jeho aktuálny stav. Ak je výsledok kontroly pozitívny (oddiel je voľný), stavanie cesty je zahájené. V prípade negatívneho výsledku kontroly (oddiel je obsadený), stavanie cesty je odmietnuté. Kontrola spomenutých podmienok je v zmysle požiadavky F2.8 (tab. 2.2).

V nasledujúcich cykloch začína stavanie jazdnej cesty. Všetky úseky jazdnej cesty sú prehlásené za vyhradené pre stavanie jazdnej cesty, aby bolo znemožnené postavenie konfliktnej cesty. V tejto fáze sa pri symbole koncového návestidla zobrazí žltý index prebiehajúceho stavania jazdnej cesty. Funkcia vykoná kontrolu požadovanej polohy výmen, ktoré sú súčasťou cesty (prechádzaných a odvrátných). Ak je výmena prestavená do iného smeru, ako je požadovaný pre stavanú cestu, je vydaný povel na jej automatické prestavenie. Funkcia bude cyklicky volaná až do chvíle, kým budú všetky potrebné výmeny prestavené v požadovanom smere pre jazdnú cestu. Následne je v slovníku „*dictStavanieCesty*“ nastavený parameter „*vymeny*“ na hodnotu True. Cyklus určený pre kontrolu požadovanej polohy výmen je navrhnutý v zmysle požiadavky F2.3 (tab. 2.2).

V tejto chvíli je možné pristúpiť k dokončeniu stavania jazdnej cesty. Funkcia cyklicky prechádza koľajové úseky jazdnej cesty a prehlasuje za záverované pre jazdnú cestu. Vo vizualizácii sa pri symbole záverovaných výmen zobrazí zelený kruh a žltý index pri odchodovom návestidle sa zmení na zelený (záverovaná vlaková cesta) alebo biely (záverovaná posunová cesta). Následne je vydaný povel na rozsvietenie povoľujúcej návesti na počiatočnom návestidle. Stavanie jazdnej cesty je ukončené a posledným krokom je nastavenie premenných v slovníku „*dictStavanieCesty*“ na východziu hodnotu „False“, aby boli pripravené vo východzom stave pre stavanie ďalšej vlakovej cesty.

7.4.2 Funkcia „*rusenieCesty*“

Po zavolaní danej funkcie, sa podľa vybraného koncového návestidla prehľadáva záverová tabuľka. Hľadá sa taká cesta, ktorej počiatočné návestidlo má príznak rozsvietennej povoľujúcej návesti. Tiež sa skúma či je postavenie výmen pre danú cestu správne. Ak sú splnené vyššie uvedené podmienky, funkcia prechádza na druhú časť rušenia cesty.

V druhej časti sa pri cieľovom návestidle zmení index na fialový (označenie rušenej jazdnej cesty) a skúma sa, aký typ záveru má jazdná cesta. V prípade predbežného záveru (voľný približovací úsek), sa okamžite pristupuje k rušeniu záveru jazdnej cesty.

V prípade trvalého záveru (obsadený približovací úsek) sa spúšťa časový súbor 3 minúty pre vlakovú cestu alebo 1 minúta pre posunovú cestu. Po uplynutí stanoveného času, je možné pokračovať v rušení cesty.

V tretej časti funkcia cyklicky prechádza všetky úseky jazdnej cesty, ruší príznak záveru daného úseku a vo vizualizácii sa zobrazujú príslušné indikácie. V prípade, že cesta obsahuje aj ochrannú dráhu, všetky úseky ochrannej dráhy sú uvoľnené rovnakým spôsobom. Funkcia je navrhnutá v zmysle požiadavky F2.5 (tab. 2.2).

7.4.3 Funkcia „*ruseenieOD*“

Táto funkcia sa volá počas štandardného rušenia ochrannej dráhy. Po obsadení staničnej koľaje, sa spúšťa časové oneskorenie 30 sekúnd. Počas behu časovača sa očakáva bezpečné zastavenie vlaku pred koncovým návestidlom. Po uplynutí spomenutého času, je volaná funkcia „*ruseenieOD*“, pre zrušenie záveru úsekov ochrannej dráhy rovnako ako v prípade rušenia neprejdenej jazdnej cesty.

7.4.4 Funkcia „*prestavenieVyh*“

Funkcia „*prestavenieVyh*“ slúži na manuálne, prípadne automatické prestavovanie výmen v stanici. Jej vstupným podnetom je informácia o poslednej vybranej výmene. Funkcia preskúma či nie je výmena záverovaná pre jazdnú cestu a či je voľná. Po úspešnej kontrole je do PLC vydaný podnet na prestavenie výmeny do opačnej polohy. Funkcia je navrhnutá v zmysle požiadavky F2.6 (tab. 2.2).

7.4.5 Záverová tabuľka

Záverová tabuľka je vyhotovená špecificky pre každú stanicu na trati. Dáta záverových tabuliek sú uložené v súboroch „*zaverTab.py*“. V súbore sa nachádza zložený slovník. Vonkajšia vrstva slovníka obsahuje zoznam všetkých jazdných ciest v stanici. Meno každej cesty hovorí o tom, na ktorom zhlaví bude daná cesta vytvorená či ide o odchodovú alebo vchodovú cestu, číslo dopravnej koľaje a informáciu o tom či je daná cesta posunová alebo vlaková. Vnútorňá časť zoznamu je špecifická pre každú jazdnú cestu. Na tejto vrstve nájdeme informácie o tom, ktoré návestidlo je počiatočným a koncovým prvkom danej cesty, ako majú byť prestavené výmeny, vylúčené jazdné cesty a zoznam úsekov nachádzajúcich sa v jazdnej ceste, prípadne v ochrannej dráhe.


```

'ZBE_RAD_1v_OD': {
  'start': ['Z_L', 'DZ_L'],
  'stop': ['Z_L1', 'DZ_L1'],
  'ZBE_V1': [1],
  'ZBE_V2': [1],
  'ZBE_V3': [1],
  'DISP_ZBE_V1': [1],
  'DISP_ZBE_V2': [1],
  'DISP_ZBE_V3': [1],
  'Vyluky': ['ZBE_RAD_1v', 'ZBE_RAD_1v_OD', 'ZBE_RAD_2v', 'ZBE_RAD_1o', 'ZBE_RAD_2o', 'ZBE_LUZ_2v_OD',
    'ZBE_k1_Lk', 'ZBE_Lk_k1', 'ZBE_k2_Lk', 'ZBE_Lk_k2',
    'ZBE_HLO_1v', 'ZBE_HLO_1v_OD', 'ZBE_HLO_1o', 'ZBE_HLO_2v', 'ZBE_HLO_2o',
    'ZBE_k1_Sk', 'ZBE_Sk_k1', 'ZBE_k2_Sk', 'ZBE_Sk_k2'],
  'Useky': ['ZBE_Lk', 'ZBE_V1', 'ZBE_k1',
    'DISP_ZBE_Lk', 'DISP_ZBE_V1', 'DISP_ZBE_k1'],
  'UsekyOD': ['ZBE_V3', 'ZBE_Sk',
    'DISP_ZBE_V3', 'DISP_ZBE_Sk'],
  'dopUseky': [],
  'PU': 'RAD_ZBE_TU4',
  '1TU': '0'
},

```

Obr. 7.7 Zápis jazdnej cesty v stanici Radošina

Obr. 7.7 obsahuje príklad jazdnej cesty v stanici Zbehy. Zobrazuje vchodovú cestu na prvú staničnú koľaj zo smeru Radošina, so zriadenou ochrannou dráhou. Počiatočné návestidlo „start“ je vchodové návestidlo L. Koncové návestidlo cesty zapísané v zozname „stop“, je L1. Vo vlakovej ceste sa nachádza výhybková spojka V1-2 a výhybka V3 („ZBE_V1“, „ZBE_V2“, „ZBE_V3“). Všetky výhybky majú byť prestavené do priamej polohy. Nasleduje zápis všetkých vylúčených jazdných ciest „Vyluky“, zápis kontrolovaných úsekov, ktoré sú súčasťou vchodovej cesty „Useky“ a zápis všetkých úsekov, tvoriacich ochrannú dráhu „UsekyOD“. Zoznam „dopUseky“ je relevantný pri posunových cestách, preto je prázdny. Úsek zapísaný ako „PU“ je približovací úsek vchodovej cesty. Položka „1TU“ je relevantná iba pri odchodových cestách. Obsahuje zápis prvého traťového úseku za stanicou.

Vchodová cesta je vyňatá zo záverovej tabuľky pre dispečerské pracovisko, preto sú niektoré prvky zapísané dvakrát v jednotlivých zoznamoch. Jeden element sa nachádza v podrobnom zobrazení a druhý element v prehľadovom zobrazení trate. Príkladom môže byť zápis výhybiek „ZBE_V1“ a „DISP_ZBE_V1“.

7.5 Riadenie vonkajších prvkov v koľajisku

Riadenie vonkajších prvkov v koľajisku je zabezpečené vytvorenými funkčnými blokmi. PLC SIMATIC S7-300 obsahuje funkčný blok, pre vyhodnotenie a spracovanie informácií z jednotlivých koľajových obvodov a ich konverziu na informácie o voľnosti traťových úsekov, sprostredkované pre spolupracujúce PLC SIMATIC S7-1200. V PLC SIMATIC S7-1200 sme vytvorili funkčné bloky pre riadenie prvkov staničných, traťových a priesestných zabezpečovacích systémov.

Riadenie oddielových návěstidiel v medzistaničných úsekoch aj riadenie závor a výstražníkov na priecestiach je spojené s logikou daných zabezpečovacích zariadení vo funkčných blokoch príslušných zabezpečovacích systémov. Preto sme pre tieto prvky nevytvárali dedikované funkčné bloky. Funkčné bloky pre riadenie vonkajších prvkov bolo nutné vyhotoviť pre staničné zabezpečovacie zariadenia, pretože logika SZZ sa nachádza v PC aplikácii obslužného pracoviska. Vytvorením týchto funkčných blokov sme zjednodušili a zefektívniili komunikáciu medzi PC a PLC. Nie je nutné, aby PC aplikácia riadila jednotlivé svetlá na návěstidlách a výmeny v staniách. Namiesto toho do PLC odošle iba informáciu o aktuálnom type návěstného znaku pre konkrétne návěstidlo alebo smer prestavenia konkrétnej výmeny. Zvyšok je spracovávaný v rámci funkčných blokov vonkajších prvkov SZZ.

7.5.1 Funkčné bloky riadenia staničných návěstidiel

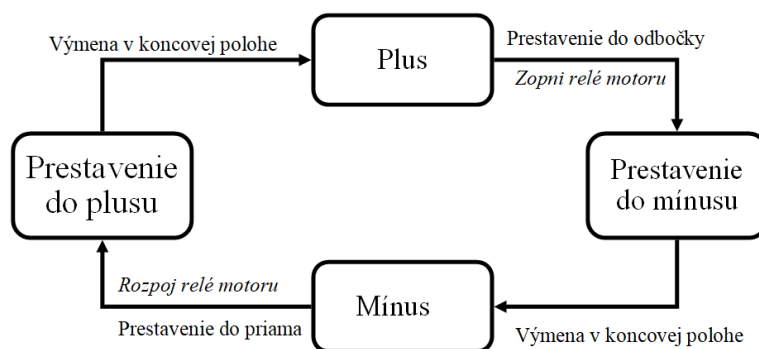
Pre každú stanicu sme vytvorili dedikovaný funkčný blok, definujúci správanie návěstidiel danej stanice. Daný funkčný blok nám poskytuje aj informáciu o voľnosti príľahlého medzistaničného úseku. Vstupným podnetom pre každé návěstidlo je informácia od systému SIMIS W. Informácia pre návěstidlo je dátového typu „Byte“ a môže nadobudnúť štyri hodnoty:

- A. 0 – Stoj
- B. 1 – Voľno (akákoľvek návěst dovoľujúca jazdu vlaku)
- C. 2 – Posun
- D. 3 – Privolávací návěst

Ďalším vstupným podnetom je spätné hlásenie polohy výmen v úseku za návěstidlom. Pre zriadenie predzvestnej väzby medzi vchodovým a odchodovým návěstidlom, je pre vchodové návěstidlo potrebné poznať aj stav odchodových návěstidiel. Podľa toho rozhodovací strom vyberie správnu návěst pre zobrazenie. V stanici Radošina a Zbehy sme navyše zriadili predzvestnú väzbu medzi odchodovými návěstidlami a prvým oddielovým návěstidlom automatického bloku. V staniách Zbehy a Hlohovec sme vytvorili väzbu s oddielovým návěstidlom automatického hradla v smere do konkrétnej stanice (Zbehy – So, Hlohovec - Lo) pre správny prenos odhlášky. Funkčné bloky staničných návěstidiel boli navrhnuté v zmysle požiadavky F2.2 a F2.7 (tab. 2.2). Názorná ukážka vytvorených rozhodovacích stromov vchodového a odchodového návěstidla v stanici Zbehy sú v prílohe (Príloha H I).

7.5.2 Funkčný blok výmeny

Funkčný blok výmeny je tvorený stavovým automatom tvoreným štyrmi stavmi. Dva stabilné, kedy sa výmena nachádza v koncových polohách a dva prechodné, predstavujúce pohyb výmeny počas prestavovania. Vstupmi stavového automatu sú podnety na prestavenie výmeny. Výstupom stavového automatu je pokyn pre relé motoru prestavníka. Po zapnutí systému, sa automaticky výmeny prestavia do priamej polohy (poloha plus) a stavový automat sa nachádza vo východnom stave „Plus“. Ak sa aktivuje podnet na prestavenie výmeny do odbočného smeru, vydá sa povel na zopnutie relé motoru prestavníka a stavový automat prechádza do stavu „*Pohyb do mínusu*“. Podmienkou na opustenie tohto stavu je aktivácia koncového spínača prestavnej tyče. Vtedy sa deaktivuje vstupný podnet na prestavenie výmeny, zmení sa stav premennej spätného hlásenia „*smer*“ pre riadiace algoritmy SZZ a automat prechádza do stavu „Mínus“. Jazyky výmeny sú teraz prestavené pre jazdu do odbočného smeru. Prestavovanie výmeny späť do priamej polohy prebieha rovnako, ale pri prestavovaní automat prechádza do stavu „*Pohyb do plusu*“ a pre relé motoru prestavníka je odoslaný povel na rozpojenie (viď. Obr. 7.8). Ak by bol vydaný povel na prestavenie výmeny do smeru, v ktorom sa už nachádza, algoritmus vyhodnotí prestavovanie ako nepredmetné a vydaný povel sa zahodí.

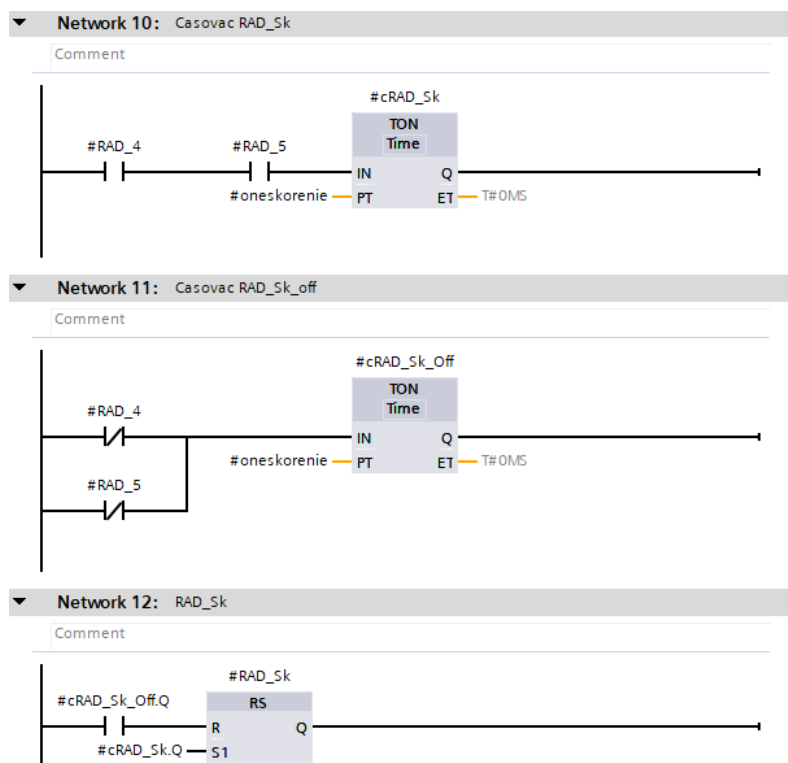


Obr. 7.8 Stavový automat funkčného bloku výmeny

7.5.3 Funkčný blok koľajových obvodov

Funkčný blok „*Tratove_useky*“ je zložený zo sád časovačov a preklápacích RS obvodov pre každý traťový úsek. V jednej sade sa nachádza preklápací RS obvod, nesúci informáciu o voľnosti daného úseku a dva časovače, slúžiace pre oneskorenie reakcie klopného obvodu. Obr. 7.9 je príkladom jednej sady, konkrétne pre traťový úsek „Sk“ na zhlaví stanice Radošina. Vstupnými podnetmi sú informácie z koľajových úsekov „*RAD_4*“ a „*RAD_5*“. Ak sú oba koľajové úseky voľné (t.j. majú hodnotu log. 1) dlhšie, ako je stanovená časová konštanta oneskorenia „oneskorenie“ (250 ms), traťový úsek je prehlásený za voľný a premenná „*RAD_Sk*“ je nastavená na log. 1. V opačnom

prípade, ak je aspoň jeden z koľajových obvodov obsadený (t.j. má hodnotu log. 0) dlhšie ako stanovený čas, premenná „RAD_Sk“ je nastavená na hodnotu log. 0.



Obr. 7.9 Príklad jednej sady z funkčného bloku "Tratove_useky"

Časovače, tvoriace oneskorenie reakcie preklápacieho RS obvodu, slúžia na „znecitlivenie“ PLC počas trvania prechodových javov v detektoroch. Ihneď po obsadení izolovaného úseku koľajovým vozidlom dochádza k rôznym prúdovým výkyvom, ovplyvňujúcim zozbierané dáta. Počas prechodových javov môže stav izolovaného úseku oscilovať, preto je nutné v tomto čase nebrať do úvahy vstupné dáta do PLC.

7.6 Algoritmy grafického rozhrania obsluhy

Základnými funkciami vrstvy ovládania a zobrazovania je zobrazenie aktuálneho stavu vonkajších prvkov na trati a umožnenie obsluhu zadávať príkazy pre obsluhu vonkajších prvkov a dopravných procesov. Vzhľadom na veľké množstvo dát, ktoré je potrebné spracovať a uchovávať, sme sa rozhodli pre objektový prístup k vývoju aplikácie. Základnými prvkami aplikácie sú objekty jednotlivých vonkajších prvkov. Vytvorili sme tri základné triedy pre výhybky, návěstidlá a koľajové obvody. Základnou súčasťou každej triedy je konštruktor, pre tvorbu objektov. Aby bol zdrojový kód prehľadný, rozdelili sme ho na viacero častí a pre každú z nich sme vytvorili nový súbor zdrojového kódu pre Python s príponou .py. Prepojenie jednotlivých súborov a funkčnosť celej aplikácie sme zaistili importovaním potrebných súborov do hlavného súboru príkazom „import“.

7.6.1 Súbory zdrojového kódu

Pre zachovanie prehľadnosti sme sa rozhodli zdrojový kód funkčných algoritmov rozdeliť do niekoľkých súborov. Každý súbor predstavuje jednu triedu, prípadne ucelenú časť algoritmov. Štruktúra zdrojového kódu je nasledovná:

1. „app.py“
 - a. „ico.py“
 - b. „ui_form.py“
 - c. „usek.py“
 - d. „navestidlo.py“
 - e. „priecestie.py“
 - f. „vyhybka.py“
 - g. „SZZ.py“ (opis v kap. 7.4)
 - I. „zaverTab.py“ (opis v kap. 7.4.5)
 - h. „tratSuhlas.py“
 - i. „riadenieObsluhy.py“

Súbor „app.py“ obsahuje hlavnú časť zdrojového kódu aplikácie. Tvorí základ nášho modelu systému ILTIS-N. Súbor sa skladá z viacerých tried. Hlavnou triedou je trieda „App“, obsahujúca kompletnú logiku grafickej vizualizácie. Medzi najdôležitejšie funkcie sa radí funkcia „update“, ktorej úlohou je cyklická kontrola všetkých objektov vizualizácie (koľajové úseky, návestidlá, výhybky, ...) a aktualizovať ich výzor podľa stavu ich vnútorných premenných.

Funkcia „clickObjekt“ spracováva podnety obsluhy, pri kliknutí na objekt vizualizácie, resp. pri výbere objektu na obsluhu. Jej vstupnými parametrami je index a typ objektu, vybraný obsluhou. Zabezpečuje riadenie prístupu k objektom vo vizualizácii tak, aby nebolo možné obsluhovať dané objekty vo chvíli, kedy pracovisko nemá súhlas na obsluhu. Tiež zabezpečuje zobrazenie správneho výberového menu, pre prácu s označeným objektom vizualizácie. Ak obsluha z ponúknutého menu vyberie niektorú z akcií, je zavolaná príslušná funkcia pre spracovanie dopytu.

Príkladom môže byť funkcia „akciaNavestidlo“, volaná po výbere návestidla pre obsluhu. Táto funkcia vyhodnocuje výber počiatočného a koncového návestidla pre postavenie jazdnej cesty. Telo funkcie tvorí štruktúra rozhodovacieho stromu „if ... else“. Vstupnými podnetmi funkcie sú index vybraného úkonu z ponuky a identifikátor návestidla. Ako príklad je v tab. 7.2 uvedený opis výberového menu pre odchodové návestidlo. Kompletný opis všetkých výberových ponúk sa nachádza v prílohe (Príloha E I).

tab. 7.2 Popis výberovej ponuky pre odchodové návestidlo

Položka ponuky	Vysvetlenie	Index položky
VLAK	stavanie vlakovej cesty	1
POSUN	stavanie posunovej cesty	2
PN	aktivácia privolávacej návesti	3
ZR JC	zrušenie jazdnej cesty	4
VOLNO	manuálne zadanie „Voľno“ na návestidle	5
STOJ	manuálne zadanie „Stoj“ na návestidle	6

Funkcia tiež kontroluje správnosť výberu obsluhovaného objektu. Systém napríklad znemožní obsluhu vybrať koncové návestidlá kusých koľají, prípadne symboly fiktívnych zriaďovacích návestidiel ako počiatočné návestidlo jazdnej cesty. Prípadne pri výbere koncového návestidla odchodovej vlakovej cesty, funkcia kontroluje správny smer traťového súhlasu a prítomnosť odhlášky za posledným vlakom. Okrem už spomínaných funkcií sa tu nachádzajú aj iné, zabezpečujúce ostatné aspekty funkčného správania modelu ILTIS-N:

2. „*akciaPriecestie*“ slúži na manuálne otvorenie a zatvorenie priecestí
3. „*akciaTS*“ slúži na spracovanie podnetov pre zmenu traťového súhlasu (návrh funkcie spĺňa požiadavky F1.3 a F3.7 (tab. 2.1 a tab. 2.3))
4. „*akciaStanica*“ spracováva podnety pre zmenu súhlasu na obsluhu pracoviska
5. „*vyberCestu*“, „*postavCestu*“ a „*ukonciStavanie*“ slúžia na spracovanie podnetov pre algoritmy SZZ, volajú funkcie uvedené v kap. 7.4 (návrh funkcie spĺňa požiadavku F1.2 (tab. 2.1))
6. „*prikazDoPLC*“ slúži na vytvorenie a odoslanie podnetov pre algoritmy TZZ a PZZ nachádzajúcich sa v PLC a pre komunikáciu s REST API rozhraním
7. Iné funkcie zabezpečujúce funkčné správanie grafickej vizualizácie:
 - a. „*popUp*“ slúži na otvorenie a zatvorenie okna konfigurácie REST API rozhrania
 - b. „*citajAdresu*“ slúži na načítanie IP adresy REST API rozhrania
 - c. „*zahajPripojenie*“ a „*ukonciPripojenie*“ slúži na otvorenie a ukončenie pripojenia PC aplikácie k REST API rozhraniu
 - d. „*aktualizaciaCasu*“ slúži na výpis aktuálneho času a dátumu na obrazovke systému ILTIS-N
 - e. „*update*“ na pre aktualizáciu výzoru jednotlivých symbolov vo vizualizácii
 - f. „*clickObjekt*“ slúži na spracovanie kliknutia na objekt vo vizualizácii
 - g. „*comboShowHide*“ slúži na zobrazenie a skrytie správnej výberovej ponuky pre prvok vybraný obsluhou
 - h. „*lupa*“ slúži na prepínanie medzi prehľadovým zobrazením celej trate a podrobným zobrazením stanice na pracovisku vzdialenej obsluhy

- i. „vypisHlasenia“ slúži na výpis prevádzkového hlásenia pri chybnéj obsluhu pracoviska
- j. „quit“ slúži a zatvorenie PC aplikácie pracoviska

Spomenuté funkcie boli vypracované v súlade s požiadavkou F1.1 (tab. 2.1).

V súbore „ico.py“ sa nachádzajú algoritmy, spracovávajúce bitové mapy elementov vizualizácie. Nachádzajú sa tu zoznamy typu „enum“ (Obr. 7.10), pre každý element v grafickej aplikácii (návestidlá, výhybky a koľajové úseky), uchovávajúce špecifické súbory bitových máp pre jednotlivé elementy vizualizácie a metóda „create“. Táto metóda pre každý symbol zavolá vstavanú metódu „addFile“ nad objektom „QIcon“, čím z našej bitovej mapy vytvorí použiteľnú ikonu, priradenú objektom „Button“ a „Label“ v grafickom okne aplikácie.

```
class NavZriadL(enum.Enum):
    STOJ = 'icon_zr_stojL' #posun zakázaný
    STOJ_V = 'icon_zr_stojL_v' #posun zakázaný, výber návestidla
    STOJ_OBS = 'icon_zr_stojL_obs' #posun zakázaný, obsadený úsek pred
    STOJ_OBS_V = 'icon_zr_stojL_obs_v' #posun zakázaný, výber návestidla, obsadený úsek pred
    STOJ_STAVANIE_START = 'icon_zr_stojL' #posun zakázaný, výber posunovej cesty
    STOJ_STAVANIE_START_OBS = 'icon_zr_stojL_obs' #posun zakázaný, výber posunovej cesty
    STOJ_VC = 'icon_zr_stojL_vc' #posun zakázaný, úsek pred návestidlom vo vlakovej ceste
    STOJ_PC = 'icon_zr_stojL_pc' #posun zakázaný, úsek pred návestidlov v posunovej ceste
    STOJ_OD = 'icon_zr_stojL_od' #posun zakázaný, úsek pred návestidlov v ochrannnej dráhe
    STOJ_ZAVER_STAVANIE = 'icon_zr_stojL_zaver_stavanie' #posun zakázaný, žltý index cesty
    STOJ_ZAVER_PC = 'icon_zr_stojL_zaver_pc' #posun zakázaný, biely index cesty
    STOJ_ZAVER_PC_VYB = 'icon_zr_stojL_zaver_pc_v' #posun zakázaný, biely index cesty, výber návestidla
    RUSENIE = 'icon_zr_stojL_rusenie_cesty' #posun zakázaný, fialový index cesty
    POSUN = 'icon_zr_posunL' #posun
    POSUN_V = 'icon_zr_posunL_v' #posun, výber návestidla
    POSUN_OBS = 'icon_zr_posunL_obs' #posun, obsadený úsek pred návestidlom
    POSUN_V_OBS = 'icon_zr_posunL_v_obs' #posun, obsadený úsek pred návestidlom, výber návestidla
```

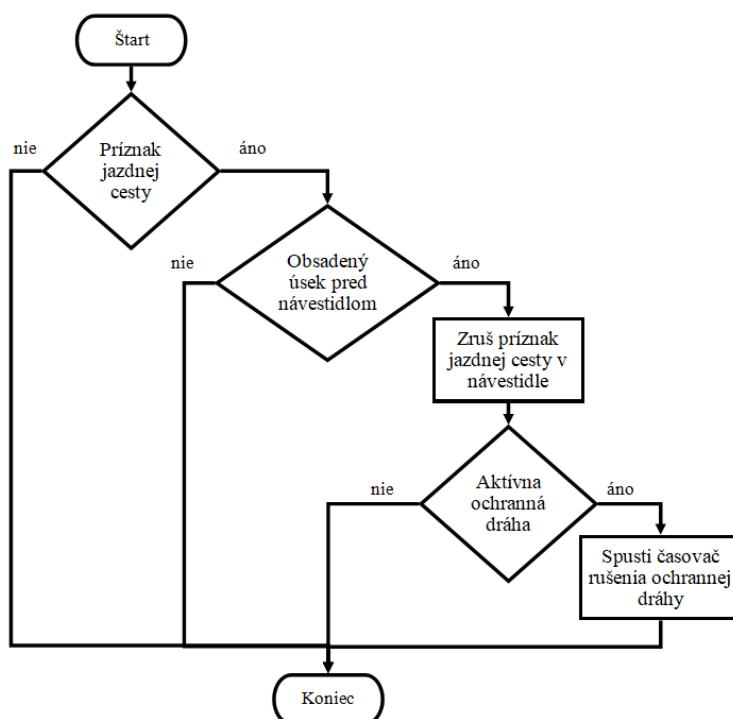
Obr. 7.10 Príklad zoznamu „enum“ pre symbol zriaďovacieho návestidla

Súbor „ui_form.py“ je vytváraný pomocou modulu PySide6 dostupného v jazyku Python. Vo vývojovom prostredí Qt Creator sme v dizajnovej časti vytvorili okno grafickej aplikácie. Rozmiestnili sme v ňom jednotlivé prvky typu „Button“ a „Label“ tak, aby sme vytvorili reliéf trate č.2 a každému z nich sme priradili bitovú mapu, pre základné zobrazenie po spustení aplikácie. Zdrojový kód ku grafickej časti aplikácie sa vytvára automaticky v súbore „form.ui“. V tomto súbore je vo formáte XML definované zobrazenie grafickej aplikácie. Obsahuje názvy vkladáných prvkov, ich vlastnosti, polohu prvkov v okne aplikácie a pod. Tento súbor nie je možné priamo upravovať v prostredí Qt Creator a nie je spustiteľný, preto je potrebné pred spustením aplikácie konvertovať súbor „form.ui“ pomocou modulu PySide6 na spustiteľný Python súbor (v našom prípade „ui_form.py“).

Súbory tried „usek.py“, „navestidlo.py“, „priecestie.py“ a „vyhybka.py“ obsahujú rovnomenné triedy objektov, predstavujúce základné železničné prvky. Každá z tried

obsahuje metódy pre prácu s objektmi danej triedy. Najdôležitejšou z nich je metóda „*update*“, slúžiaca na zmenu zobrazovaných symbolov jednotlivých elementov v grafickom okne. Väčšina atribútov objektu nesie informácie o aktuálnom stave objektu (smer prestavenia, obsadenosť úseku, návěstný znak..), na základe ktorých rozhodovací strom v tejto metóde určí správny symbol elementu vo vizualizácii. Po výbere správneho symbolu sa zavolaním vstavanej metódy „*setIcon*“ nad prvkom vizualizácie priradenom danému objektu, zmení zobrazovaný symbol. Pre uchovanie informácií o dostupných symboloch konkrétneho elementu sme vytvorili potrebný počet zoznamov typu „*enum*“. V nich sú definované statické zoznamy všetkých možných symbolov pre daný element.

Okrem toho triedy obsahujú aj špecifické funkcie, slúžiace na úpravu atribútov jednotlivých objektov v rôznych prevádzkových situáciách. Príkladom môže byť metóda „*rusenieCesty*“ triedy „*Navestdidlo*“. Jej primárnou úlohou je úprava atribútov objektu návěstidla tak, aby vo vizualizácii prebehla zmena návěstného znaku pri prejazde koľajového vozidla okolo návěstidla. Okrem toho zabezpečuje aj postupné rušenie záveru koľajových úsekov jazdnej cesty pri prejazde koľajového vozidla úsekmi danej jazdnej cesty (Obr. 7.11).



Obr. 7.11 Reprezentácia metódy „*rusenieCesty*“ objektu „*Navestdidlo*“

Pre jednoduchú orientáciu sme objekty týchto tried uložili do slovníkov. Kedykoľvek potrebujeme pristupovať k metódam niektorého z objektov, vyhľadáme ho v slovníku a určíme, ktorá metóda sa má vykonať. Napríklad objekty triedy „*Navestdidlo*“ sú uložené v slovníku „*dictNav*“ a objekty triedy „*Usek*“ a „*Vyhybka*“ sú v druhom, spoločnom

slovníku „dictUpdate“. Rozhodli sme sa tak, pretože pri aktualizácii dát obsadenosti koľajových úsekov je pre nás praktickejšie mať všetky objekty, ktorých sa koľajové úseky priamo týkajú v jednom slovníku. Objekty triedy „Navestidlo“ k týmto dátam prístupujú nepriamo, preto môžu byť vo vlastnom slovníku.

Súbory „tratSuhlas.py“ a „riadenieObsluhy.py“ obsahujú pomocné triedy, vykonávajúce pomocné funkcie, pre správnu kooperáciu jednotlivých PC aplikácií modelu ILTIS-N. Tieto triedy spracovávajú dáta o traťových súhlasoch medzi stanicami na trati a informácie súvisiace s riadením obsluhy z pracovísk lokálnej obsluhy.

7.6.2 Väzba medzi grafickým oknom a metódami

Nami vytvorená aplikácia v prostredí Qt Creator, po spustení vytvorí grafické okno, podľa nami definovaných parametrov iba raz a následne cyklicky volá funkciu, aktualizujúcu grafické okno podľa dát nadobudnutých v poslednom cykle aplikácie. V prípade, že chceme vytvoriť väzbu medzi prvkom v grafickom zobrazení („Button“, „Label“, „Combo box“...), je nutné v hlavnej metóde „__main__“ definovať signály podobné prerušeniam v programe. Každý z prvkov má definované signály, meniace sa podľa stavu prvku. Napríklad prvok „Button“ má definované signály, nesúce informáciu o tom či bolo tlačidlo stlačené, aktivované, pustené a podobne. Všetky tieto signály dokážu byť spojené s vstavanou metódou „connect“, ktorá po zavolaní spustí vopred definovanú užívateľskú metódu s definovanými parametrami.

```
501 widget.ui.RAD_zr_do_st_odZ.clicked.connect(lambda: widget.nav_click(1))
502 widget.ui.RAD_S.clicked.connect(lambda: widget.nav_click(3))
503 widget.ui.RAD_L1.clicked.connect(lambda: widget.nav_click(4))
504 widget.ui.RAD_L2.clicked.connect(lambda: widget.nav_click(5))
505 widget.ui.RAD_fik_S.clicked.connect(lambda: widget.nav_click(6))
506 widget.ui.RAD_fik_l8.clicked.connect(lambda: widget.nav_click(7))
507 widget.ui.RAD_1k_fik.clicked.connect(lambda: widget.nav_click(8))
508 widget.ui.RAD_2k_fik.clicked.connect(lambda: widget.nav_click(9))
```

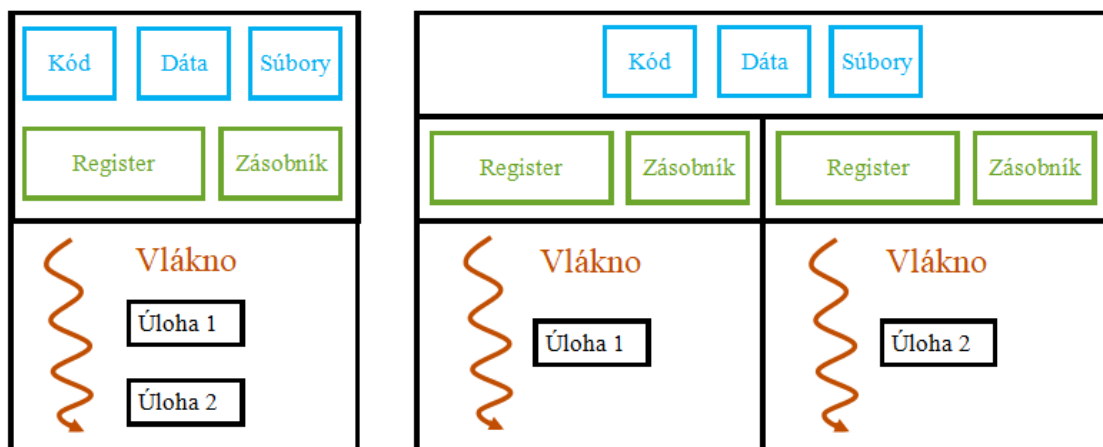
Obr. 7.12 Príklad prepojenia grafického okna s užívateľskými metódami

Obr. 7.12 je ukážkou vytvorených väzieb medzi elementami návestidiel grafického okna a užívateľskými metódami lokálneho obslužného pracoviska pre stanicu Radošina. Pre ilustráciu funkčnosti prepojenia bližšie popíšeme príkaz na riadku 502. Začiatok príkazu, „widget.ui.“, odkazuje na objekt triedy „App“, resp. grafické okno. Nasleduje meno konkrétneho elementu („RAD_S“), ktorému je daný príkaz priradený. Ďalej sa v príkaze nachádza definícia signálu, spúšťajúceho dané prepojenie („clicked“), konkrétne po kliknutí na element. Koniec príkazu obsahuje volanie už spomínanej metódy „connect“, ktorej sme v parametri definovali spúšťanú užívateľskú metódu „widget.nav_click(6)“. Volanie užívateľskej metódy obsahuje aj vstupné parametre pre danú metódu. Funkcia „lambda“, ktorá sa v príkaze vyskytuje, slúži na to, aby bolo

možné vložiť do parametra metódy volanie inej metódy. Keby sme ju v našom prípade nepoužili, metóda „*widget.nav_click()*“ by bola volaná aj v iných prípadoch, kedy to je nežiadúce, nielen po kliknutí na priradený element v grafickom okne aplikácie.

7.6.3 Viacvláknové spracovanie údajov

Pri návrhu programového kódu, sme sa dostali do bodu, kedy sme potrebovali čítať dáta z väčšieho množstva koľajových obvodov. Tieto dáta bolo potrebné spracovať a uložiť do príslušných objektov a následne aktualizovať grafické okno aplikácie. Počiatočná koncepcia počítala s jednovláknovým spracovaním aplikácie. Pri prvotnom testovaní funkcií sme narazili na problém, súvisiaci s týmto prístupom. Po načítaní a počas spracovávania dát, nebolo grafické okno aplikácie programovo obsluhované a v dôsledku toho nebola možná obsluha žiadneho z elementov. To obsluhu navodzovalo pocit nefunkčnosti programu, tzv. zamrznutia okna. Tento nedostatok sa prejavoval v každom cykle načítavania údajov. Najefektívnejším riešením sa nám javilo vytvoriť viac vlákien a beh aplikácie rozdeliť medzi jednotlivé vlákna tak, aby sme dosiahli paralelné spracovanie údajov (Obr. 7.13).



Obr. 7.13 Porovnanie jednovláknového a viacvláknového spracovania

Na tento účel sme využili knižnicu „*QThread*“ dostupnú v balíčku PySide6 pre Python. V programe sme vytvorili triedu „*dataUpdate*“, obsahujúcu algoritmy potrebné pre spracovanie a zápis dát z PLC a REST API rozhrania do vytvorených objektov našich tried. V tejto triede sa nachádza už spomínaný slovník, s uloženými objektmi. Po spustení programu a úspešnom pripojení aplikácie k PLC, sa spúšťa druhé vlákno, s cyklicky spúšťanou metódou „*run*“ triedy „*dataUpdate*“. Metóda po spustení odošle dopyt do PLC. PLC v odpovedi vráti potrebné dáta vo formáte JSON. Metóda následne rozbalí obdržané dáta. Pomocou cyklov „*for*“ ich roztriedi medzi jednotlivé objekty všetkých tried. Po ukončení triedenia a zápisu dát, metóda odošle do hlavného vlákna

aplikácie správu o ukončení práce v danom cykle. V hlavnom vlákne sú v triede „App“ vybudená metóda „update“, určená pre aktualizáciu stavu objektov všetkých tried.

Vedľajšiemu vláknu sme nepriradili žiadnu periódu, s ktorou by sa mala volať metóda „run“ triedy „dataUpdate“. To znamená, že čítanie a zápis dát je prakticky neustály. Počas testovania sme merali čas vykonávania spomínanej metódy „run“ od odoslania dopytu do PLC, až po odoslanie správy o ukončení čítania do hlavného vlákna. Táto hodnota je pre každú z aplikácií špecifická. Záleží hlavne od počtu aktualizovaných objektov. Uvedené merania sme vykonali v aplikácii pre lokálne obslužné pracovisko stanice Zbehy, ktorá je najväčšou stanicou na trati. Perióda, čítania a zápisu dát priemerne trvala 1,2 sekundy. Prevádzkový výkon vlákna je podľa nás postačujúci, pre splnenie požiadavky F1.5 (tab. 2.1).

8 ÚPRAVA RIADENIA TRATE S MODELOM ESA-44

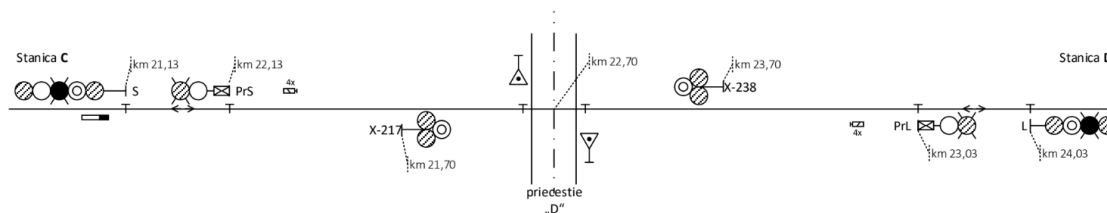
Návrh riadenia modelovej železnice systémom ESA-44 bol témou inej diplomovej práce [15]. Vzhľadom na to, že práca bola vypracovávaná v období distančného vzdelávania, nebolo možné venovať sa praktickej časti v rozsahu stavby koľajiska a implementácie systému v laboratóriu AB205. Preto sa stavba trate zlúčila s bakalárskymi prácami na danom koľajisku a s našou diplomovou prácou.

8.1 Zmena topológie koľajiska

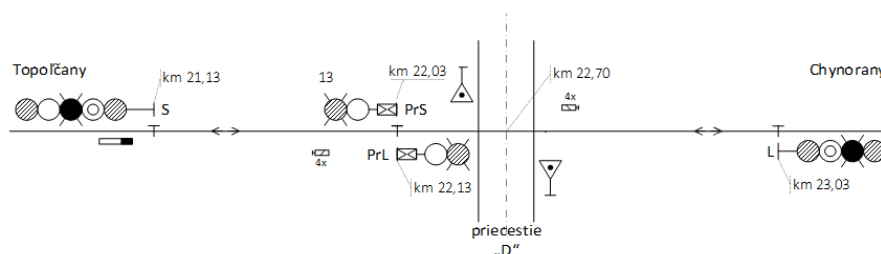
Autorom diplomovej práce [15] zaoberajúcej sa modelom systému ESA-44, bol vytvorený a navrhnutý riadiaci softvér pre trať Nitra – Chynorany v laboratóriu AB205, ako aj schémy zapojenia digitálneho riadenia trate (pripojenie dekodérov, vonkajších prvkov). Tiež bol vyhotovený adresný softvér, umožňujúci komunikáciu modelu systému ESA-44 a digitálnej centrály Lenz LZV-100.

Po ukončení pokladania koľají na modelovom koľajisku, sme zistili nezrovnalosti medzi fyzickou topológiou a projektovanou topológiou trate. V realite nebolo možné vytvoriť projektovaný návrh, pretože na stole by nezostal dostatok miesta pre osadenie všetkých návěstidiel a vytvorenie izolovaných úsekov. Tento problém vznikol v medzistaničnom úseku Topoľčany – Chynorany. Pôvodný návrh počíta z piatimi úsekmi (viď. Obr. 8.1). Fyzická dĺžka miesta vyhradeného pre realizáciu medzistaničného úseku na stole neumožňovala vytvorenie všetkých izolovaných úsekov. Po osadení vchodových návěstidiel a ich predzvestí, sa obe predzvesti nachádzali v jednom bode na koľajisku a nezostal medzi nimi priestor pre priecestie. Po konzultácii s autorom diplomovej práce [15], sme v úseku zmenili počet izolovaných úsekov na dva. Priecestie „D“ bolo

implementované do úseku medzi predzvest' PrL a vchodové návestidlo L v stanici Chynorany (viď. Obr. 8.2).

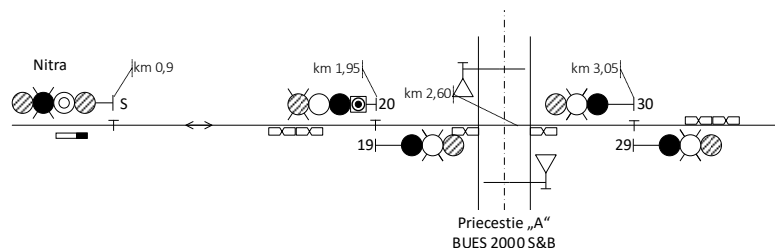


Obr. 8.1 Pôvodný stav medzistaničného úseku Topoľčany – Chynorany [10]

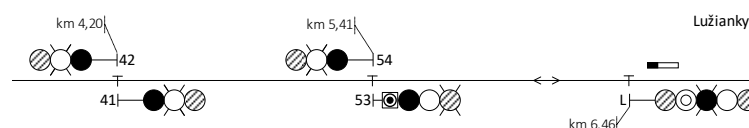


Obr. 8.2 Nový stav medzistaničného úseku Topoľčany - Chynorany

Pri analýze návrhu pripojenia LED návestidiel k dekodérom LS-100 sme zistili, že pôvodný návrh prevzatý z diplomovej práce [15], počíta s použitím celkom 56 ks dekodérov LS-100. V laboratóriu AB205 sa v súčasnom stave nachádza iba 44 ks dekodérov LS-100. Z tohto dôvodu bolo potrebné upraviť návrh pripojenia tak, aby vyhovoval použitiu 44 ks dekodérov.



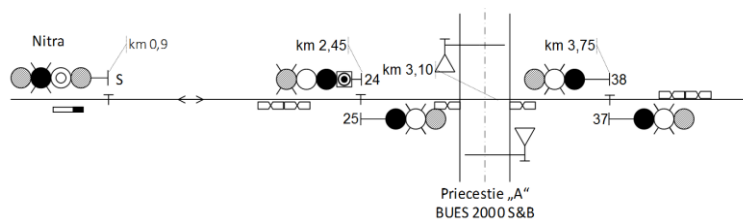
Obr. 8.3 Pôvodný stav medzistaničného úseku Nitra – Lužianky, časť 1 [15]



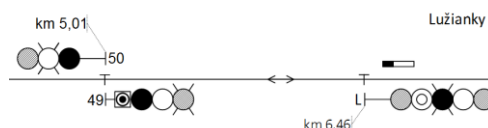
Obr. 8.4 Pôvodný stav medzistaničného úseku Nitra – Lužianky, časť 2 [15]

Jednou z úprav bola zmena počtu medzistaničných úsekov trojznakového obojsmerného automatického bloku v úseku Nitra – Lužianky. V medzistaničnom úseku sa celkovo nachádzalo päť koľajových úsekov (Obr. 8.3 a Obr. 8.4). V novom návrhu

topológie sme na trati navrhli vytvorenie troch medzistaničných úsekov tak, aby bola stále možná demonštrácia princípov funkčnosti trojznakového automatického bloku (Obr. 8.5 a Obr. 8.6).



Obr. 8.5 Nový stav medzistaničného úseku Nitra – Lužianky, časť 1



Obr. 8.6 Nový stav medzistaničného úseku Nitra – Lužianky, časť 2

Po zmene fyzického rozloženia úsekov a návěstidiel na trati, bolo potrebné tieto zmeny implementovať aj do softvéru systému ESA-44. Nanešťastie sme nemali prístup ku konfiguračnému softvéru CAS spoločnosti AŽD, pre konfiguráciu systémov ESA-44 a JOP. S autorom pôvodnej diplomovej práce sme sa po konzultácii dohodli, že úpravu softvéru systému ESA-44 vykoná on podľa našich požiadaviek.

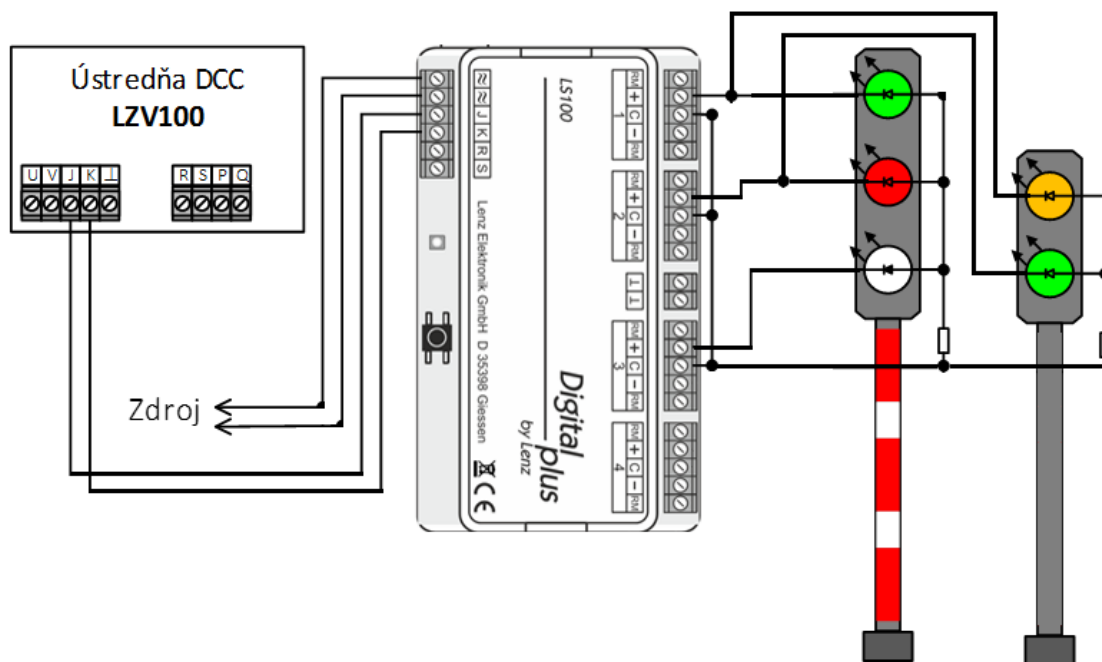
Pre zapracovanie novej topológie trate Nitra – Chynorany, bolo potrebné vytvoriť nový predpis pripájania vonkajších prvkov (návestidlá, výmeny, koľajové obvody) k digitálnym modulom Lenz LS-100 a LR-101. Podľa vytvoreného predpisu sme pokračovali parametrizovaním adries a výstupov digitálnych modulov. Pri parametrizácii sme postupovali podľa návodov k digitálnej centrále LZV-100 a diaľkovému ovládaniu LH- 100, ako aj podľa pôvodnej diplomovej práce [15]. Kompletný prehľad parametrov jednotlivých digitálnych modulov sa nachádza v prílohách na CD.

8.2 Úprava hardvérovej časti riadiaceho systému

Ako už je vyššie uvedené, bolo potrebné upraviť návrh pripojenia LED návěstidiel k dekodérom Lenz LS-100. Zmenou topológie medzistaničného úseku Nitra – Lužianky, sa nám podarilo znížiť počet potrebných dekodérov na 53 ks, čo je stále veľa. Rozhodli sme sa pre ďalší krok, ktorým bolo zlučovanie vybraných LED návěstidiel predzvestí a hlavných návěstí. Táto úprava sa týkala hlavne vchodových a odchodových návěstidiel, oddielových návěstidiel automatického hradla a samostatných predzvestí.

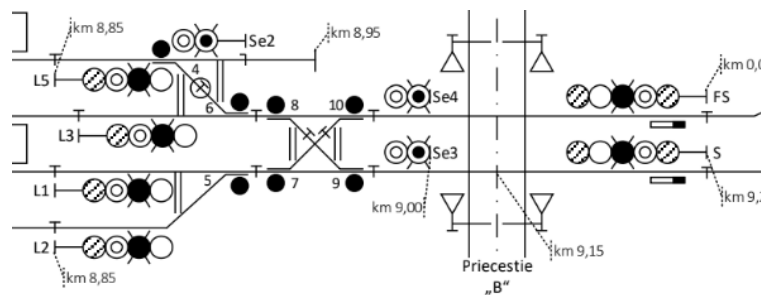
Vo všetkých staniciach na trati je uvažovaná rýchlosť na výhybkách pri jazde do odbočného smeru 40 km/h. Vďaka tomuto predpokladu sme mohli pri všetkých odchodových návěstidlách, na ktorých sa vyžaduje návštenie zníženej rýchlosti v obode výhybiek za návěstidlom, pripojiť dolné žlté svetlo k zelenému a hornému žltému svetlu.

Ako príklad sme uviedli situáciu zo stanice Lužianky. Pri stavaní odchodovej cesty z piatej koľaje smerom na ľavé zhlavie (viď. Obr. 8.8), vlaková cesta vždy prechádza cez odbočný smer výhybkovej spojky V4-V6, t.j. vždy bude na návěstidle spolu s povoluujúcou návěstou návštená aj znížená rýchlosť 40 km/h. Preto môžeme pripojiť zelené a dolné žlté svetlo návěstidla L5 na jeden výstup dekodéra LS-100. Rovnakým spôsobom dokážeme pripojiť žlté svetlo samostatnej predzvesti k červenému svetlu oddielového návěstidla a zelené svetlo predzvesti k zelenému svetlu oddielového návěstidla, čím ušetríme dva výstupy dekodéra (Obr. 8.7).



Obr. 8.7 Spôsob spájania vybraných LED hlavného návěstidla a predzvesti

Pre dosiahnutie stanoveného počtu 44 ks dekodérov, sme boli nútení v stanici Nitra pripojiť biele svetlá odchodových návěstidiel L1 – L3 iba k trvalo spínaným výstupom dekodérov, čo znemožnilo aktiváciu privolávacej návesti na spomínaných návěstidlách. V stanici sme pri poruche návěstidiel uvažovali vypravovanie vlakov iba na ručnú privolávaciu návseť a písomný rozkaz.



Obr. 8.8 Situačná schéma pravého zhlavia stanice Lužianky

Aby sme predišli nechceným skratom a problémom pri kombinovaní jednej LED na rôzne výstupy, rozhodli sme sa použiť usmerňovacie diódy rady 1N4007. Použili sme ich pri LED pripojených ku dvom výstupom dekodéra (jeden výstup pre trvalé svietenie, druhý výstup pre prerušované svietenie) a v prípadoch kedy je dolná žltá LED pripojená k výstupu pre zelenú LED, ale aj k výstupu pre hornú žltú LED. Principiálna schéma pripojenia usmerňovacích diód medzi LED návestidlo a výstupy dekodéra spolu so schémami všetkých takto upravených návestidiel sú uvedené v prílohe (Príloha C I).

8.3 Parametrizácia digitálnych dekodérov

Po vyhotovení nového návrhu parametrizácie digitálnych modulov, sme v každom digitálnom dekodéri LS-100 a LR-101 nakonfigurovali naše nové parametre. V moduloch sme zmenili komunikačnú adresu dekodéra a parametre správania jednotlivých výstupov.

Adresa digitálneho dekodéra LS-100 je uložená v registri č.1 a môže byť priradená v rozsahu 1 – 256. Registre 3 – 6 obsahujú informácie o vlastnostiach všetkých štyroch výstupov. Hodnota týchto registrov sa pohybuje v rozsahu 0 – 47. Bližšie roztriedenie a opis jednotlivých hodnôt je v tab. 8.1.

tab. 8.1 Popis registrov 3 – 6 digitálnych dekodérov LS-100

Rozsah	Význam
0 – 15	impulzné spínanie (dĺžka impulzu v rozsahu 0,1 s – 15 s)
32	trvalé zopnutie výstupu
33 – 47	spínanie s definovanou frekvenciou v rozsahu 0,5 Hz – 4 Hz

Všade kde sme potrebovali trvalé spínanie LED návestidiel, sme do registrov zapisovali hodnotu 32. Pre ovládanie prestavníkov sme nastavili impulzné spínanie s dĺžkou impulzu 0,8 s (hodnota v registri = 5). Pre prerušované svietenie LED na návestidlách sme zvolili spínanie s definovanou frekvenciou 1 Hz (hodnota v registri = 45).

Digitálne moduly LR-101 majú obdobné registre ako moduly LS-100. Register 1 obsahuje adresu modulu v rozsahu 1 – 256. Registre 11 – 18 obsahujú parametre jednotlivých vstupov modulu. Do týchto registrov vieme zadať hodnotu oneskorenia

reakcie na zmenu na vstupe modulu v rozsahu 10 ms – 2,55 s. Výrobcom nastavená hodnota registrov je 50, čo sa rovná oneskoreniu 500 ms. Túto hodnotu sme nemenili, v module LR-101 sme nastavili iba jeho adresu v registri 1.

Programovanie prebiehalo v súlade s návodom k digitálnej ústredni LVZ100 a opisom programovania v diplomovej práci [15]. Digitálny modul sme pripojili na programovaciu zbernicu centrály a s pomocou ovládača LH100 sme zadávali definované hodnoty do registrov digitálnych modulov.

8.4 Montáž digitálneho systému riadenia Lenz

Po vyhotovení potrebných úprav trate a radiaceho softvéru, sme pokračovali montážou a pripojením komponentov digitálneho riadenia of firmy Lenz na modelové koľajisko. Niektoré komponenty, ako motorické prestavníky výmen, prúdové detektory a vstupné digitálne dekodéry LR-101 boli na trati už implementované. Našou úlohou bola montáž výstupných digitálnych dekodérov LS-100 pre ovládanie výhybiek a svetiel na návestidlách. Pokračovali sme osadením napájacích transformátorov, ktoré sme rovnomerne rozmiestnili pozdĺž celej trate. Posledným krokom bolo prepojenie digitálnych dekodérov s centrálou pomocou zberníc. Celkovo sa na trati nachádzajú tri zbernice:

1. Dátová zbernica – prenos informácií pre vonkajšie prvky a hnacie koľajové vozidlá, tvorená červeným a modrým vodičom, pripájaným k terminálom J a K
2. Zbernica spätného hlásenia – hlásenie o voľnosti koľajových úsekov a polohy výmen, tvorená zeleným a žltým vodičom, pripájaným k terminálom R a S
3. Napájacia zbernica – napájanie z transformátorov 15 VAC, tvorená hnedým a šedým vodičom, pripájaným k terminálom U a V

Schéma pripojenia zberníc systému digitálneho riadenia sa nachádza v prílohách (Príloha K I).

9 ZHODNOTENIE VÝSLEDKOV

Po úspešnej implementácii a otestovaní funkčných algoritmov riadiacich modelovú železniciu, sme sa rozhodli pre vytvorenie názornej video ukážky hotového dispečerského pracoviska pre trať č.2, Radošina – Hlohovec. Ukážka sa nachádza na priloženom CD.

Vo videu je zachytené pracovisko vzdialenej obsluhy pre trať Radošina – Hlohovec. Počas prehrávania videa sa v spodnej časti obrazovky budú postupne zobrazovať zábery z modelovej železnice. Video sme sa rozhodli spomaliť, pretože jazda vlaku v reálnom čase prebehla rýchlo a video bolo neprehľadné.

Dispečer na začiatku otvorí podrobné zobrazenie stanice Zbehy. Pokračuje zmenou smeru traťového súhlasu medzi stanicami Zbehy a Hlohovec, aby mohol byť z Hlohovca vypravený vlak. Následne je postavená odchodová cesta zo stanice Hlohovec a dochádza k odoslaniu predhlášky na hradlo Rišňovce. Dispečer pokračuje postavením priebežnej vlakovej cesty v stanici Zbehy po prvej traťovej koľaji. Postavenie vchodovej cesty prebieha v prehľadovom zobrazení trate a postavenie odchodovej cesty prebieha v podrobnom zobrazení stanice. Po príjazde vlaku do stanice Zbehy dispečer ďalej stavia vchodovú cestu na druhú traťovú koľaj v stanici Radošina, kde vlak ukončí svoju jazdu.

Opis reálnych záberov modelovej železnice:

1. Záber na zhlavie stanice Hlohovec, kde je vidno prestavenie výmeny a zmena návestného znaku na odchodovom návestidle
2. Záber na hradlo Rišňovce pri prijatí predhlášky pred vlakom a prejazd vlaku hradlom
3. Záber na stanicu Zbehy. Na vchodovom návestidle dochádza k zmene návestných znakov zo „Stoj“ na „Rýchlosť 40 km/h a výstraha“. Po postavení celej priebežnej vlakovej cesty je sa na vchodovom návestidle zobrazí znak „Voľno“
4. Záber na medzistaničný úsek Radošina – Zbehy osadený modelom systému AB3-74
5. Záber na priecestie s implementovaným modelom systému PZZ-EA. Sú zachytené všetky stavy priecestia (základný stav, výstražný stav, pohyb závor, anulačný stav)

ZÁVER

Výsledkom našej diplomovej práce je funkčný model dispečerského pracoviska modelovej železnice Katedry riadiacich a informačných systémov, Fakulty elektrotechniky a informačných technológií, Žilinskej univerzity v Žiline. Na začiatku sme zhodnotili aktuálny stav zabezpečovacích systémov implementovaných na modelovej železnici. Implementácia týchto systémov bola témou bakalárskych prác riešených v minulých rokoch. Po zhodnotení stavu modelového koľajiska, sme sa rozhodli už implementované systémy zachovať pre ďalšie použitie.

Vytvorili sme špecifikáciu funkčných a technických požiadaviek na dispečerské pracovisko, ako aj na staničné zabezpečovacie systémy, ktoré na modelovej železnici chýbali. Rozhodli sme sa pre návrh a implementáciu systému SIMIS W s obslužným pracoviskom

ILTIS-N spoločnosti Siemens Mobility.

Hlavnou časťou diplomovej práce bol návrh a implementácia funkčných algoritmov pre aplikácie obslužných pracovísk a riadiaceho systému modelovej železnice. Prvú časť funkčných algoritmov sme vypracovali pre PC aplikácie v jazyku Python. Druhá časť algoritmov bola implementovaná do PLC SIMATIC S7-1200 a S7-300 formou priečkovej logiky a funkčných blokov.

Na záver sme nami vytvorené algoritmy implementovali a otestovali na modelovej železnici. Pretože sa na koľajisku nachádza druhá trať, riadená systémom ESA-44, spojená s našou traťou, vytvorili sme špecifikáciu komunikačného protokolu medzi systémami ESA-44 a SIMIS W. Komunikačný protokol sme po otestovaní implementovali do softvérovej výbavy koľajiska.

Návrh a implementácia systému ESA-44 boli témou inej diplomovej práce. Nanešťastie nikdy nedošlo k fyzickému nasadeniu systému na modelovej železnici. Nad rámec našej diplomovej práce sme sa rozhodli dokončiť implementáciu modelu systému ESA-44 na modelovej železnici. Fyzicky sme vybudovali trať č.1 Nitra – Chynorany. Na trať sme umiestnili a k digitálnym dekodérom pripojili prestavníky, návestidlá a koľajové obvody. Zapojili sme zbernice DCC systému a parametrizovali použité dekodéry. Nakoniec sme upravili adresný softvér pre model systému ESA-44.

Zoznam použitej literatúry

- [1] ZAHRADNÍK, J. – RÁSTOČNÝ, K. 2005. *Aplikácie zabezpečovacích systémov*. Žilina: EDIS vydavateľstvo Žilinskej univerzity, 2005. 226 s. ISBN 80-8070-546-1.
- [2] MACKO, D. 2021. *Model traťového zabezpečovacieho zariadenia riadený PLC*. [bakalárska práca]. Žilina: Žilinská univerzita v Žiline, Fakulta FEIT, Katedra radiacií a informačných systémov, 2021. 79s.
- [3] CHLEBNÍČAN, I. 2021. *Model priecestného zabezpečovacieho zariadenia riadený PLC*. [bakalárska práca]. Žilina: Žilinská univerzita v Žiline, Fakulta FEIT, Katedra radiacií a informačných systémov, 2021. 86 s.
- [4] STN P 34 2651: 1999, *Železničné priecestné zariadenia*.
- [5] TNŽ 34 2620: 2000, *Predpisy pre železničné staničné zabezpečovacie zariadenia*.
- [6] TNŽ 34 2630: 1999, *Predpisy pre železničné traťové zabezpečovacie zariadenia*.
- [7] ŽSR D 101/ T 101 Príloha 8a: 2009, *Obsluha elektronického stavadla typu SIMIS W. Popis činnosti*.
- [8] ŽSR D 101/T 101 Príloha 8b: 2009, *Obsluha elektronického stavadla typu SIMIS W. Spôsob obsluhy*.
- [9] ŽSR D 101/T 101 Príloha 8c: 2009, *Obsluha elektronického stavadla typu SIMIS W. Symboly zobrazovania*.
- [10] COURSEA. 2023. *What is Python Used For? A Beginner's Guide* [online]. [cit. 8-10-2023].
Dostupné na: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>
- [11] ENGLER, M. 2020. *Simulátor ŽSR na SPŠ dopravnej v Trnave* [online]. [cit. 16-4-2024]. Dostupné na: <https://www.vlaky.net/galerie/7119/simulator-zsr-na-sps-dopravnej-v-trnave/>
- [12] SIEMENS AG. 2014. *Trackguard Simis W electronic interlocking* [online].
Dostupné na:
https://assets.new.siemens.com/siemens/assets/api/uuid:dbc68ea043c3c3ebe7b aa037a4542a8c4af0fa4f/sic_00021364.pdf
- [13] *Modul LB 101*. [online]. Dostupné na:
https://ttmodelar.eu/index.php?route=product/product&product_id=3052

- [14] *Elektromotorický prestavník Tillig 86112*. [online]. Dostupné na:
https://www.ewmb.de/product_info.php?info=p2000104717_tillig-86112-h0-motorischer-unterflurweichenantrieb.html
- [15] HRUŠKA, M. 2020. *Riadenie modelového koľajiska systémom ESA 44*. [diplomová práca]. Žilina: Žilinská univerzita v Žiline, Fakulta FEIT, Katedra radiacií a informačných systémov, 2020. 106 s.
- [16] ERIC, 2024. *REST API: Your guide to getting Started Quickly* [online]. [cit. 12-02-2024]. Dostupné na: [REST API: Your Guide to Getting Started Quickly \(stackify.com\)](https://stackify.com/rest-api-your-guide-to-getting-started-quickly/)
- [17] *What is a REST API* [online]. [cit. 12-02-2024]. Dostupné na: [What is a REST API? | IBM](https://www.ibm.com/cloud/what-is-a-rest-api/)
- [18] 2014. *Raspberry Pi getting data from a S7-1200 PLC* [online]. [cit. 15-10-2023]. Dostupné na: <https://simplyautomationized.blogspot.com/2014/12/raspberry-pi-getting-data-from-s7-1200.html>

Prílohy

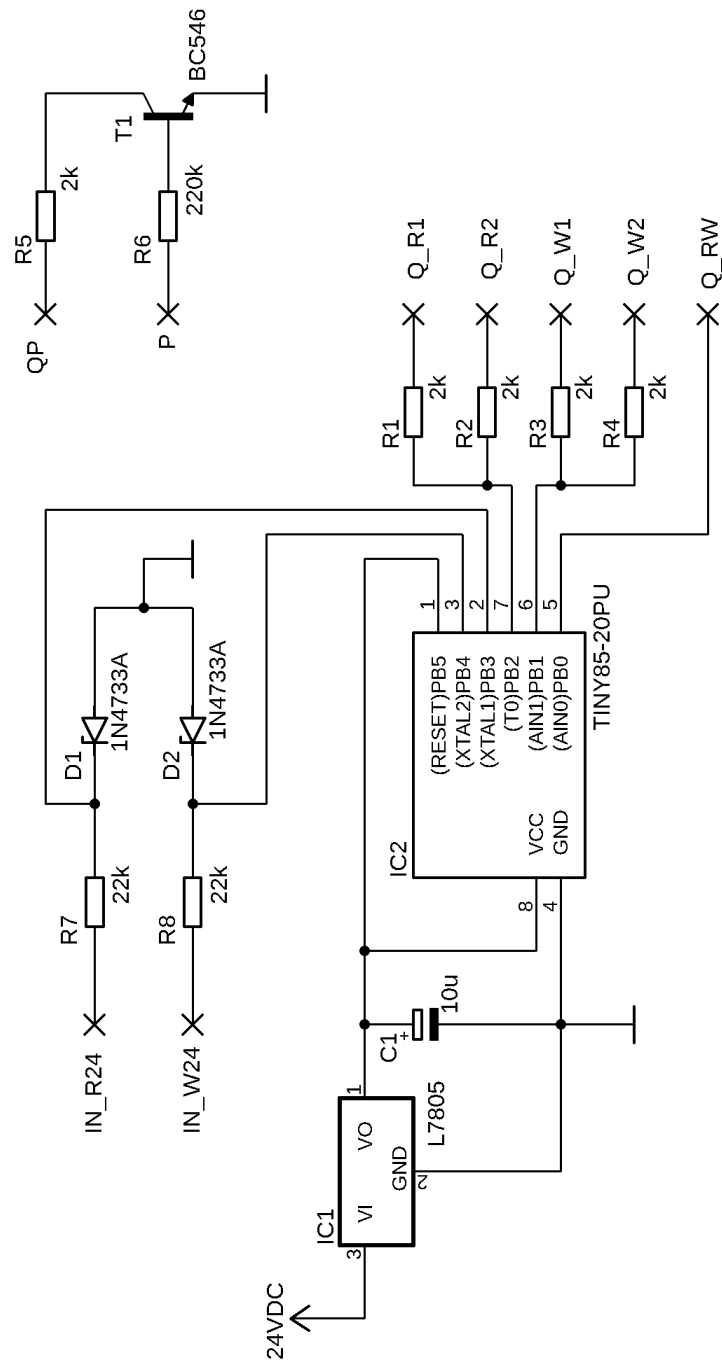
Zoznam príloh

Príloha A	Zoznam súborov na CD.....	66
Príloha B	Schéma zapojenia modulu priecestia.....	67
Príloha C	Schémy pripojenia usmerňovacích diód k návestidlám	68
Príloha D	Detailná topológia vrstvy PLC	72
Príloha E	Opis výberových ponúk prvkov.....	74
Príloha F	Hlavná obrazovka modelu ILTIS-N	76
Príloha G	Funkčné algoritmy PZZ.....	78
Príloha H	Príklady rozhodovacích stromov staničných návestidiel.....	81
Príloha I	Sekvenčný diagram automatického hradla	83
Príloha J	Vývojové diagramy systému SIMIS W	84
Príloha K	Pripojenie digitálneho riadenia trate riadenej systémom ESA-44.....	87

Príloha A | Zoznam súborov na CD

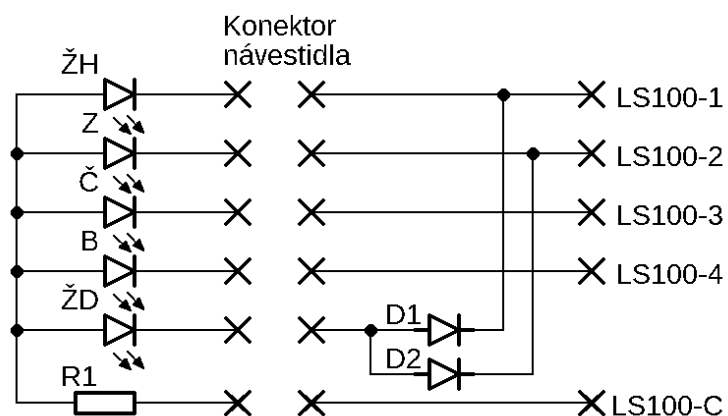
- pdf súbor textovej časti diplomovej práce „Návrh dispečerského pracoviska modelovej železnice“
- pdf súbor „Návod na obsluhu modelovej železnice“
- zložka „Aplikácie“ obsahujúca spustiteľné exe súbory jednotlivých pracovísk obsluhy spolu so zložkou „img“, obsahujúcou bitové mapy vizualizácie
- zložka „DP_Macko_2024“ obsahujúca PLC projekt v prostredí TiaPortal v.16
- zložka „Zdrojové kódy“ obsahujúca zdrojové kódy jednotlivých pracovísk obsluhy
- mp4 súbor „ILTIS“ s ukážkou modelovej železnice a pracoviska vzdialenej obsluhy
- xlsx súbor „Parametrizácia modulov Lenz“, ktorý obsahuje nastavené parametre digitálnych modulov LS-100 a LR-101 na trati č.1
- pdf súbor „Špecifikácia REST API“, obsahujúci detailný rozbor komunikačného protokolu pre systém ESA-44

Príloha B | Schéma zapojenia modulu priecestia

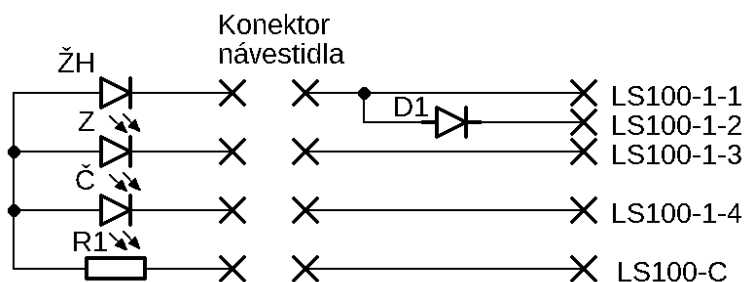


Obr. 9.1 Schéma zapojenia modulu priecestia [3]

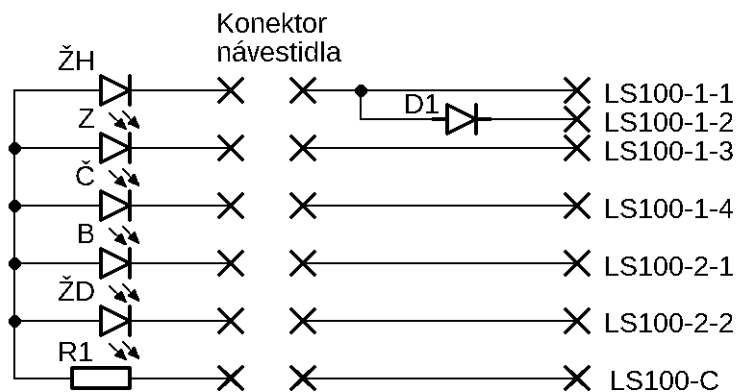
Príloha C | Schémy pripojenia usmerňovacích diód k návestidlám



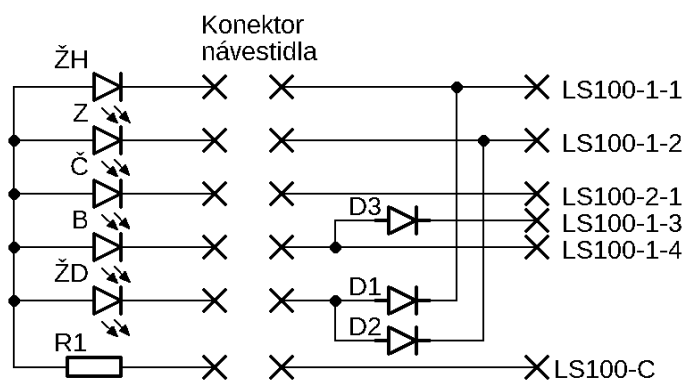
Obr. 9.2 Schéma pripojenia LED návestidiel L2, L3 v stn. Nitra



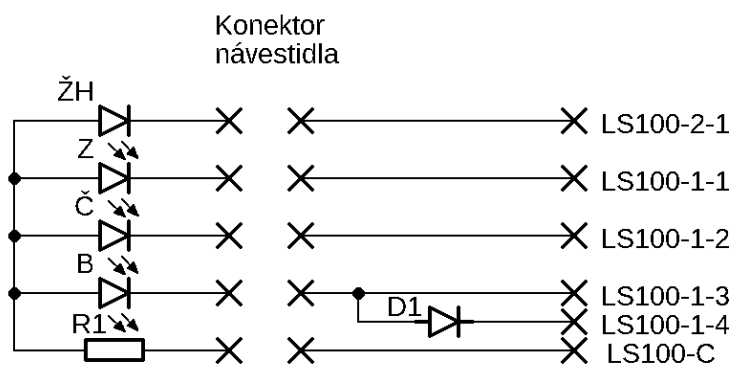
Obr. 9.3 Schéma pripojenia LED návestidiel AB24, AB49



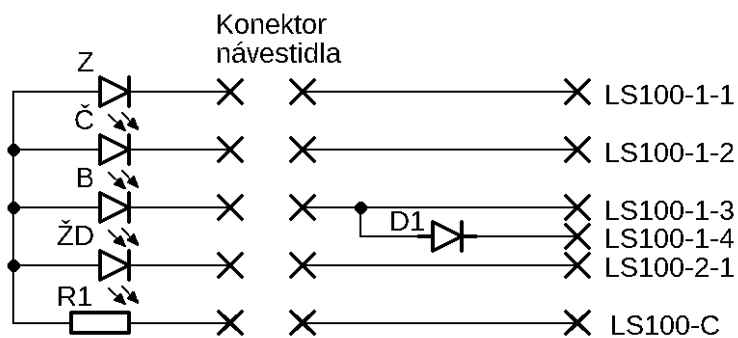
Obr. 9.4 Schéma pripojenia LED návestidla L v stn. Lužianky



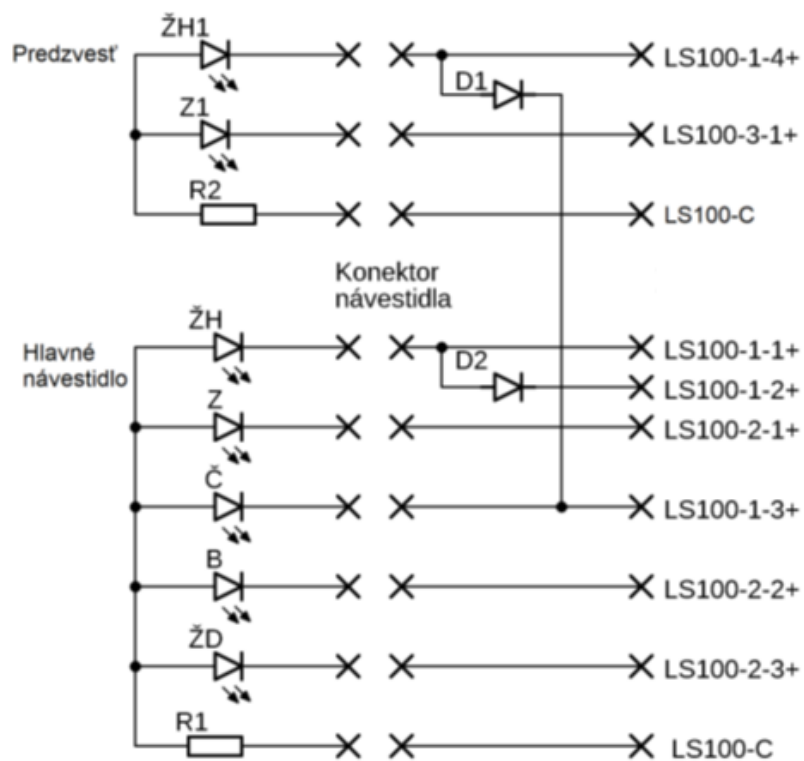
Obr. 9.5 Schéma pripojenia LED návěstidiel S1, S3, S5 v stn. Lužianky



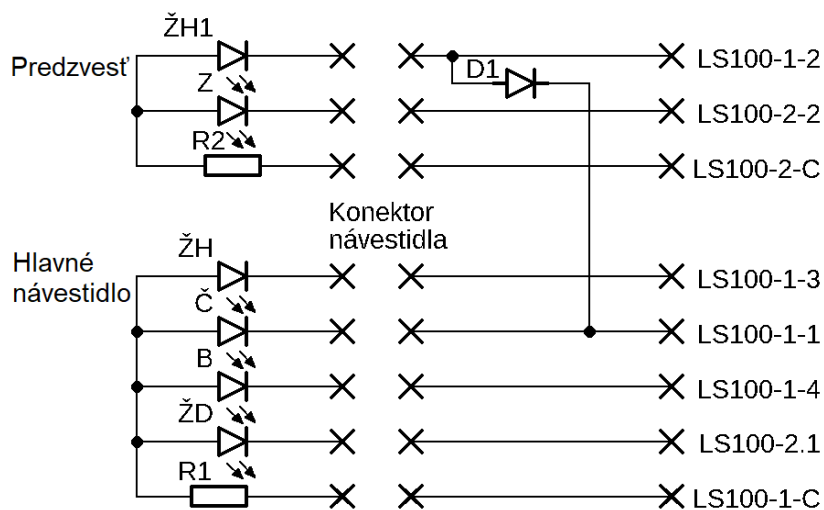
Obr. 9.6 Schéma pripojenia LED návěstidla S2 v stn. Lužianky



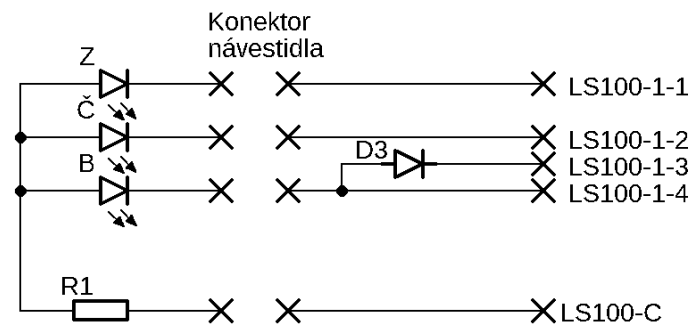
Obr. 9.7 Schéma pripojenia LED návěstidiel L1, L3 v stn. Lužianky



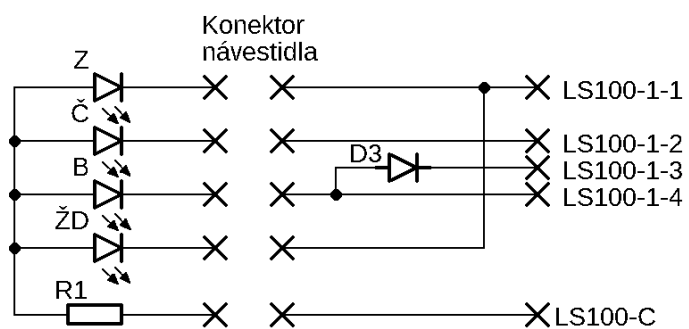
Obr. 9.8 Schéma LED návestidiel FS, S v stn. Lužianky a L, S vo výh. Topoľčany



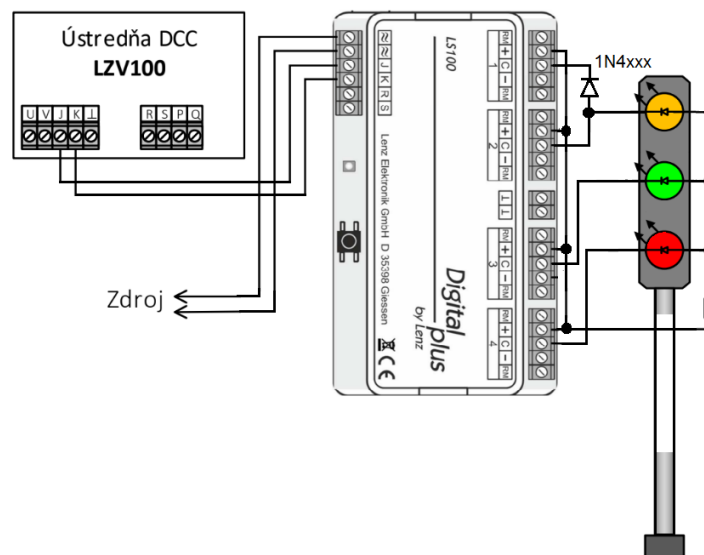
Obr. 9.9 Schéma pripojenia LED návestidiel L/PrL v stn. Chynorany



Obr. 9.10 Schéma pripojenia LED návestidla S1 v stn. Chynorany

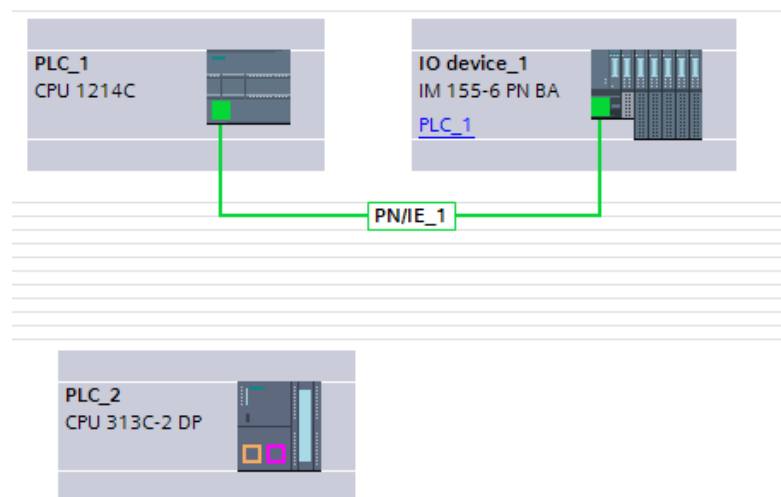


Obr. 9.11 Schéma LED návestidiel L2, L5 v stn. Lužianky a S2, S3 v stn. Chynorany

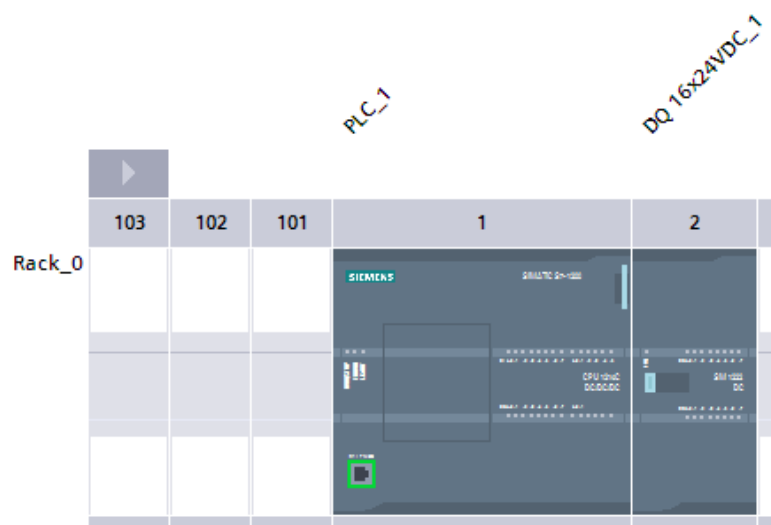


Obr. 9.12 Príklad pripojenia oddielového návestidla AB3

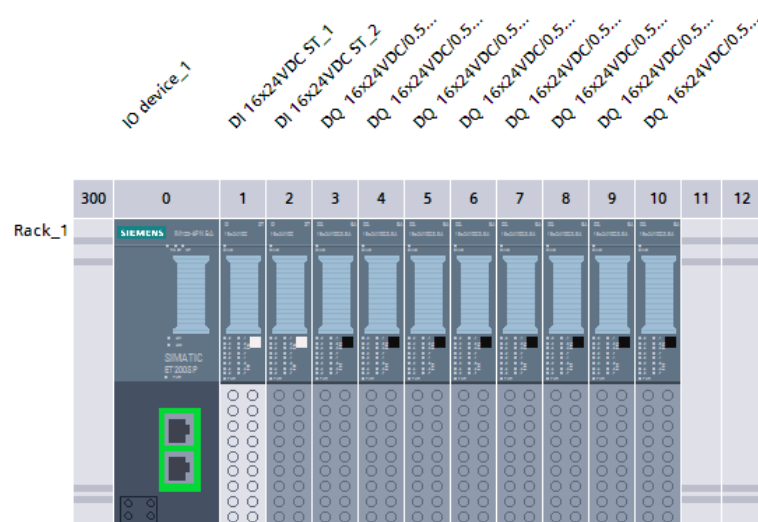
Príloha D | Detailná topológia vrstvy PLC



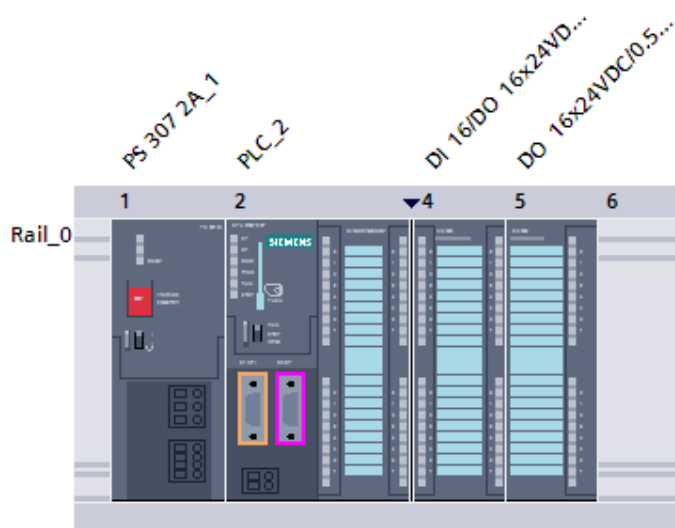
Obr. 9.13 Komunikačná vrstva PLC



Obr. 9.14 Architektúra PLC SIMATIC S7-1200



Obr. 9.15 Architektúra vzdialeného ostrovu vstupných a výstupných modulov



Obr. 9.16 Architektúra PLC SIMATIC S7-300

Príloha E | Opis výberových ponúk prvkov

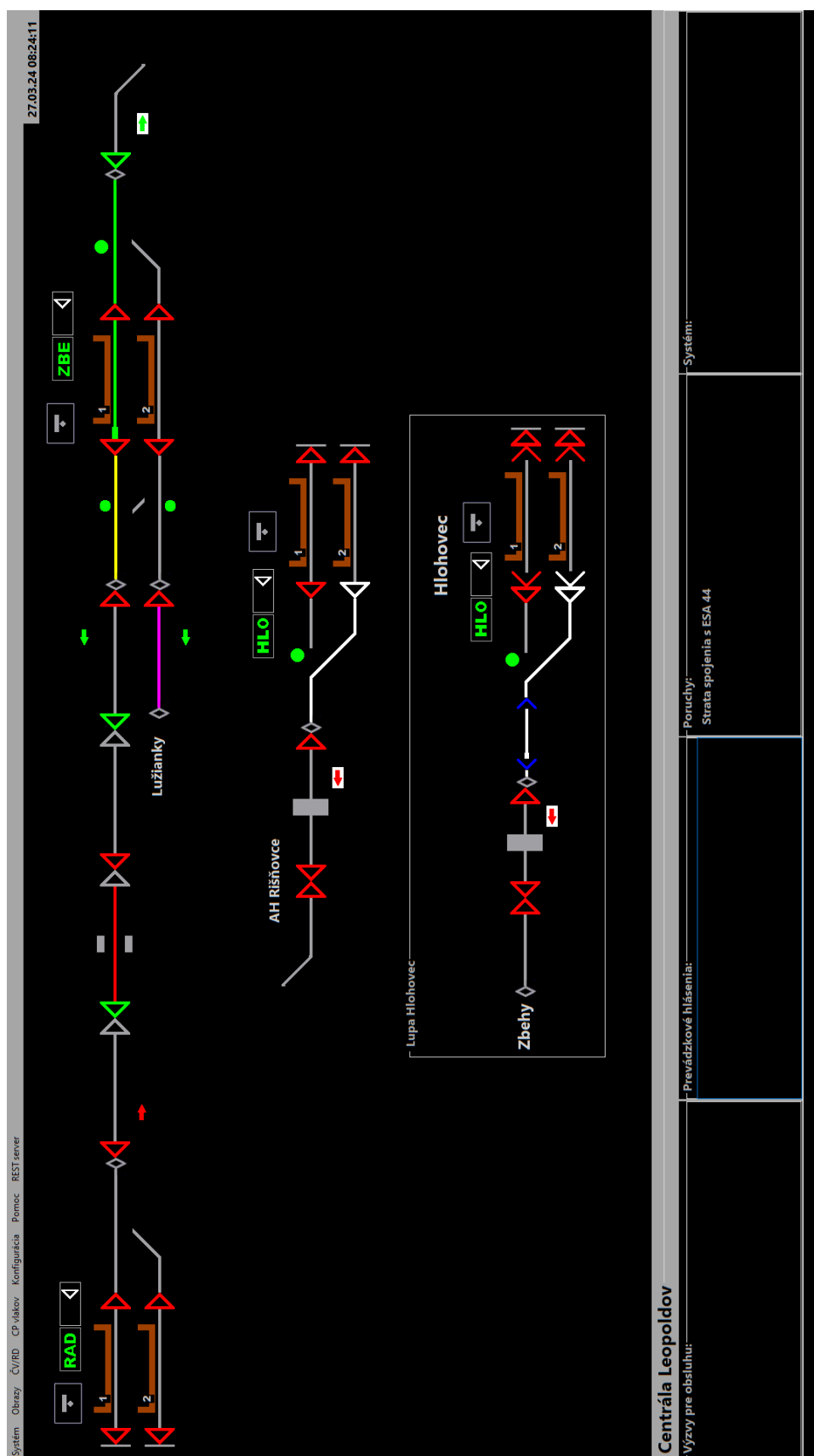
tab. 9.1 Popis výberových ponúk návěstídiel

Pre symbol hlavného návěstidla	
VLAK	stavanie vlakovej cesty
PN	privolávacia návesť
ZR JC	zrušenie jazdnej cesty***
VOLNO	manuálne zadanie povoľujúcej návěsti*
STOJ	manuálne zadanie zakazujúcej návěsti*
Pre symbol hlavného návěstidla zlúčeného so zriaďovacím návěstidlom	
VLAK	stavanie vlakovej cesty
POSUN	stavanie posunovej cesty
PN	privolávacia návesť
ZR JC	zrušenie jazdnej cesty
VOLNO	manuálne zadanie povoľujúcej návěsti*
STOJ	manuálne zadanie zakazujúcej návěsti*
Pre symbol zriaďovacieho návěstidla	
POSUN	stavanie posunovej cesty
ZR JC	zrušenie jazdnej cesty
VOLNO	manuálne zadanie povoľujúcej návěsti*
STOJ	manuálne zadanie zakazujúcej návěsti*
Pre symbol fiktívneho hlavného návěstidla	
ZR JC	zrušenie jazdnej cesty
Pre symbol hlavného návěstidla / zlúčeného návěstidla počas výberu cieľa jazdnej cesty	
PVC 40	postavenie vchodovej vlakovej cesty s obmedzenou rýchlosťou
PVC TR-1	postavenie vchodovej vlakovej cesty s ochrannou dráhou triedy 1
POS-K	postavenie posunovej cesty
Pre symbol zriaďovacieho návěstidla počas výberu cieľa jazdnej cesty	
POS-K	postavenie posunovej cesty
Pre symbol fiktívneho hlavného návěstidla počas výberu cieľa jazdnej cesty	
POC	postavenie odchodovej cesty
Pre symbol oddielového návěstidla	
PN	privolávacia návesť
VOLNO	manuálne zadanie povoľujúcej návěsti**
STOJ	manuálne zadanie zakazujúcej návěsti**
	* len ak je návěstidlo počiatočným návěstidlom postavennej jazdnej cesty ** iba ak je na návěstidle prijatá predhláška *** len ak je k návěstidlu postavená jazdná cesta

tab. 9.2 Popis prvkov ponuky pre ostatné elementy vizualizácie

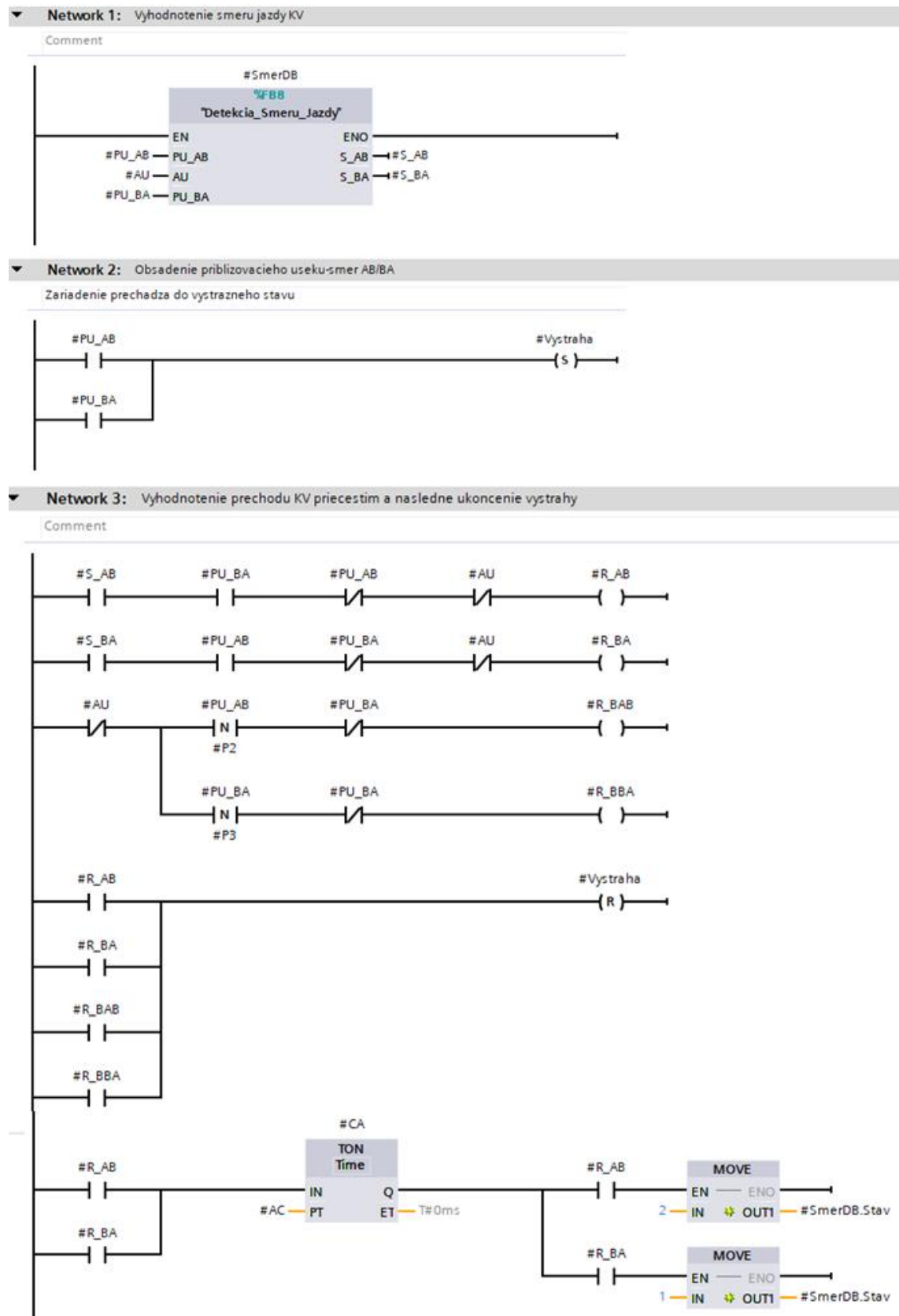
Pre symbol traťového súhlasu	
ZUS	žiadosť o traťový súhlas
ZR ZUS	zrušenie žiadosti o traťový súhlas
UTS	udelenie traťového súhlasu
Pre symbol priecestia	
ZATVOR	manuálne zatvorenie priecestia
OTVOR	manuálne otvorenie priecestia
Pre symbol výhybky	
P	manuálne prestavenie výmeny
Pre symbol názvu stanice v dispečerskom pracovisku	
CENTR	žiadosť o prevzatie riadenia stanice do diaľkovej obsluhy
LOKAL	potvrdenie žiadosti o prevzatie stanice do lokálnej obsluhy
Pre symbol názvu stanice v pracovisku lokálnej obsluhy	
LOKAL	žiadosť o prevzatie riadenia stanice do lokálnej obsluhy
CENTR	potvrdenie žiadosti o prevzatie stanice do diaľkovej obsluhy

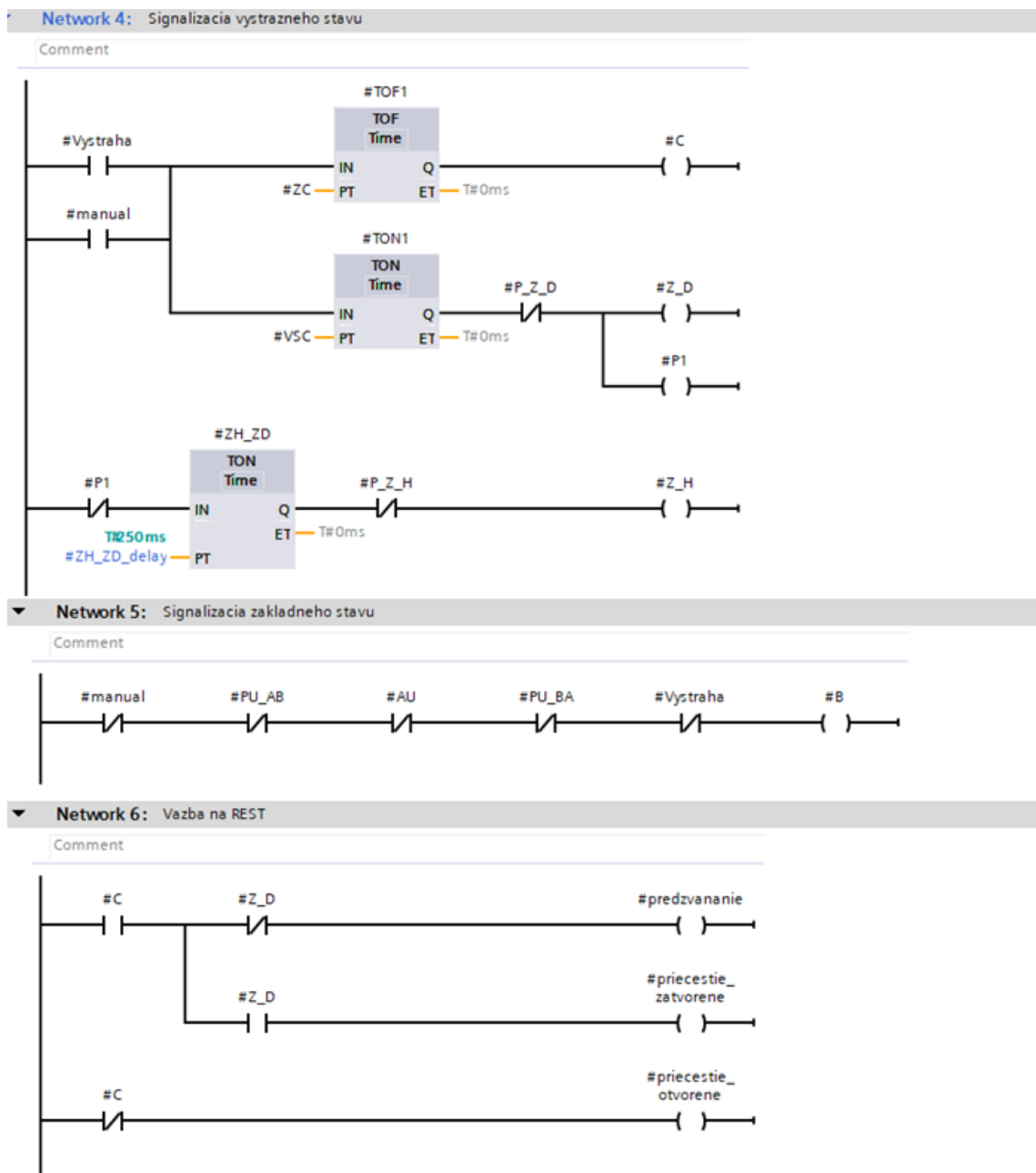
Príloha F | Hlavná obrazovka modelu ILTIS-N



Obr. 9.17 Hlavná obrazovka PC aplikácie vzdialenej obsluhy

Príloha G | Funkčné algoritmy PZZ





Obr. 9.19 Zdrojový kód algoritmu riadenia PZZ

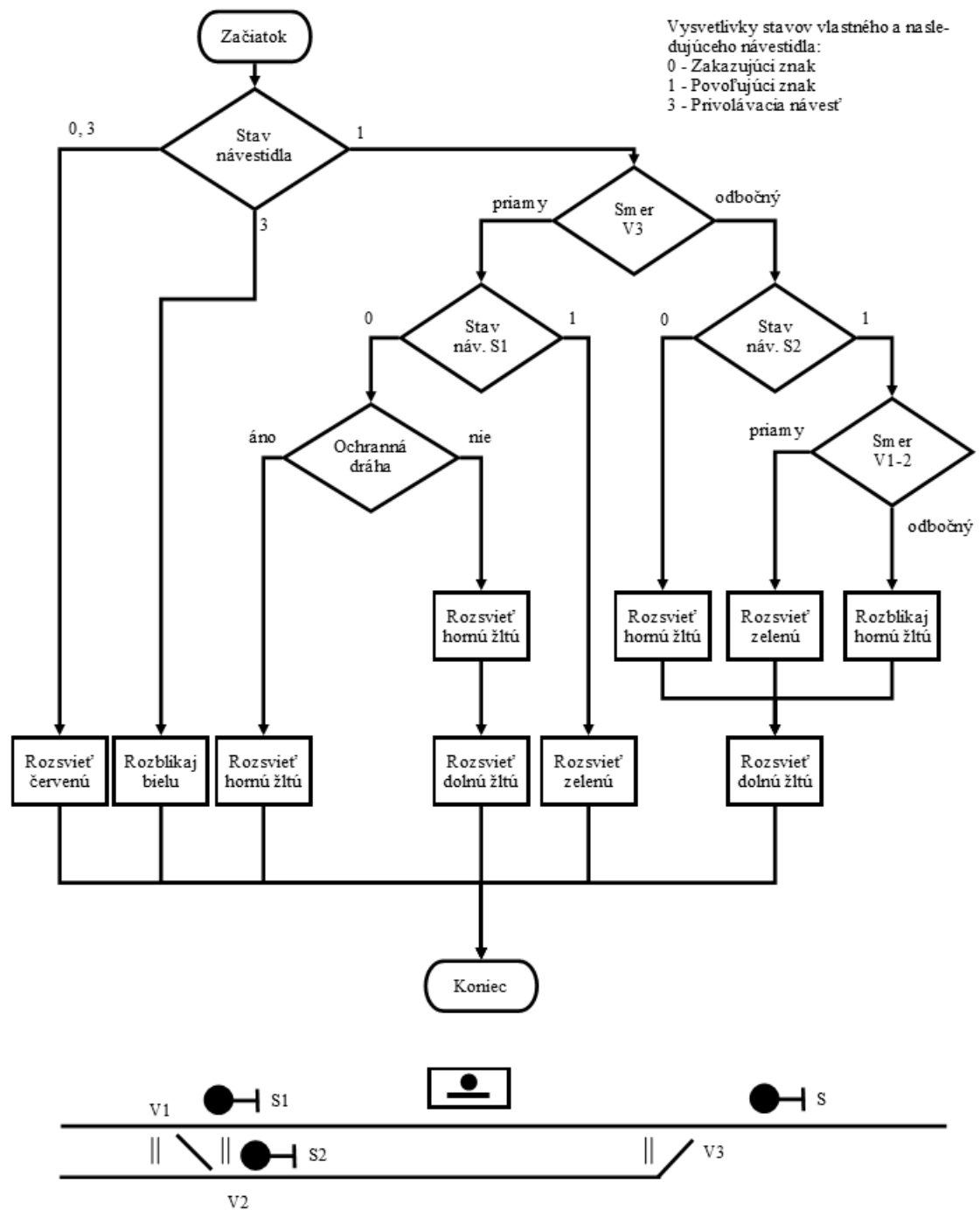
```

1 CASE #Stav OF
2     0:
3         #S_AB := FALSE;
4         #S_BA := FALSE;
5
6         IF ((#PU_AB) AND NOT (#AU) AND NOT (#PU_BA)) THEN
7             #Stav := 1;
8         END_IF;
9
10        IF (NOT (#PU_AB) AND NOT (#AU) AND (#PU_BA)) THEN
11            #Stav := 2;
12        END_IF;
13
14        1:
15            #S_AB := TRUE;
16            #S_BA := FALSE;
17
18            IF (NOT (#PU_AB) AND NOT (#AU) AND NOT (#PU_BA)) THEN
19                #Stav := 0;
20            END_IF;
21
22        2:
23            #S_AB := FALSE;
24            #S_BA := TRUE;
25
26            IF (NOT (#PU_AB) AND NOT (#AU) AND NOT (#PU_BA)) THEN
27                #Stav := 0;
28            END_IF;
29 END_CASE;
30

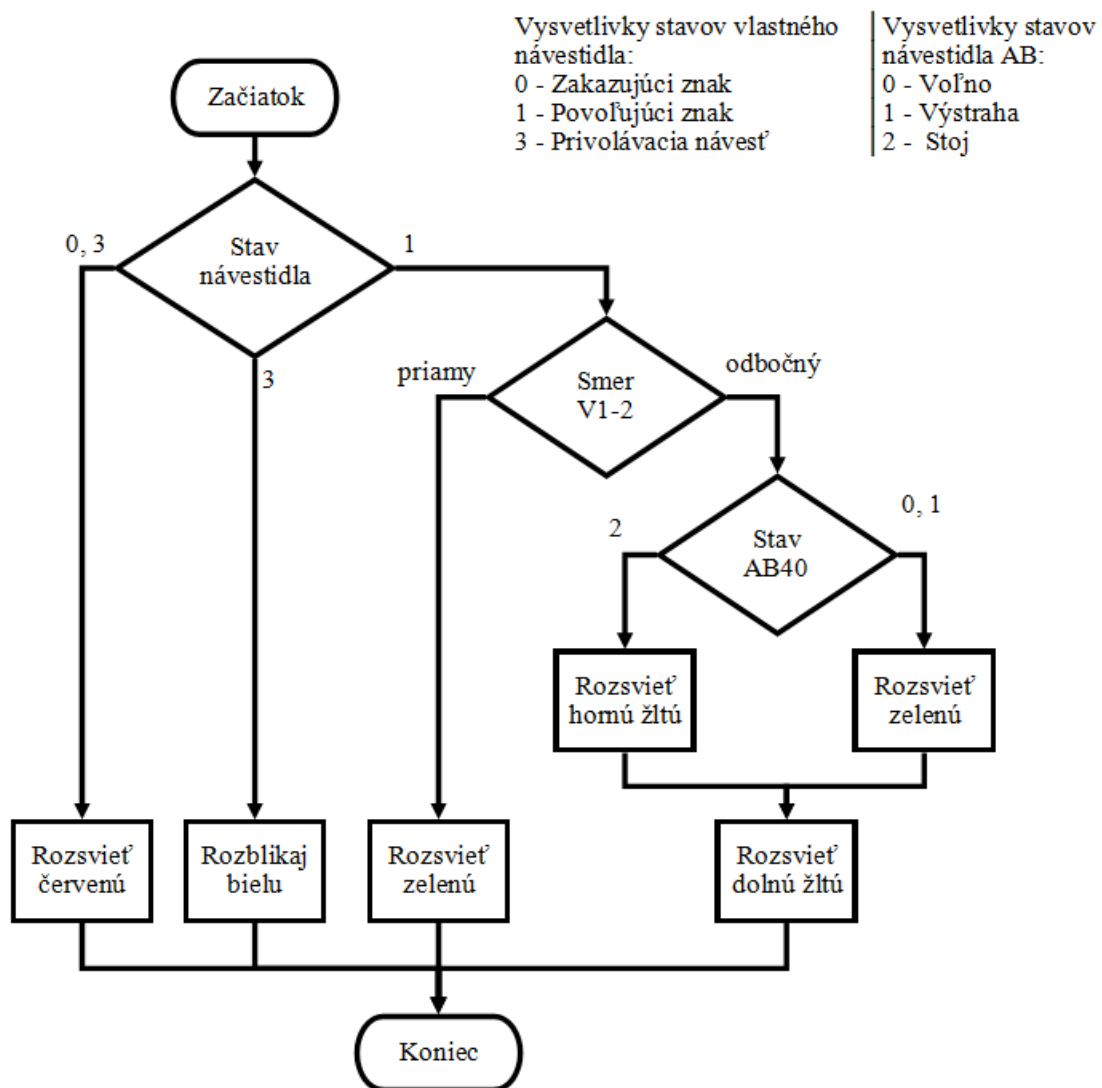
```

Obr. 9.20 Zdrojový kód bloku detekcie smeru jazdy koľajového vozidla

Príloha H | Príklady rozhodovacích stromov staničných návěstidiel

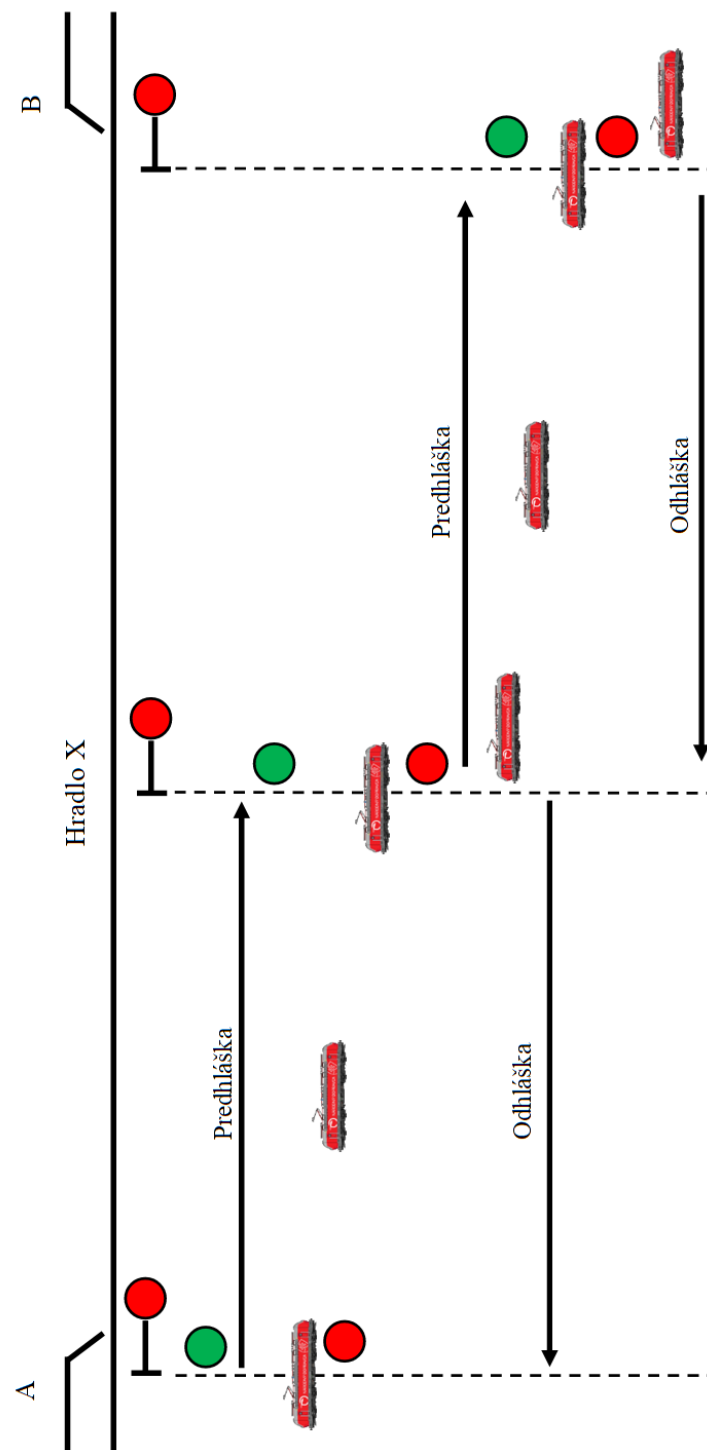


Obr. 9.21 Rozhodovací strom návěstidla S zo stanice Zbehy



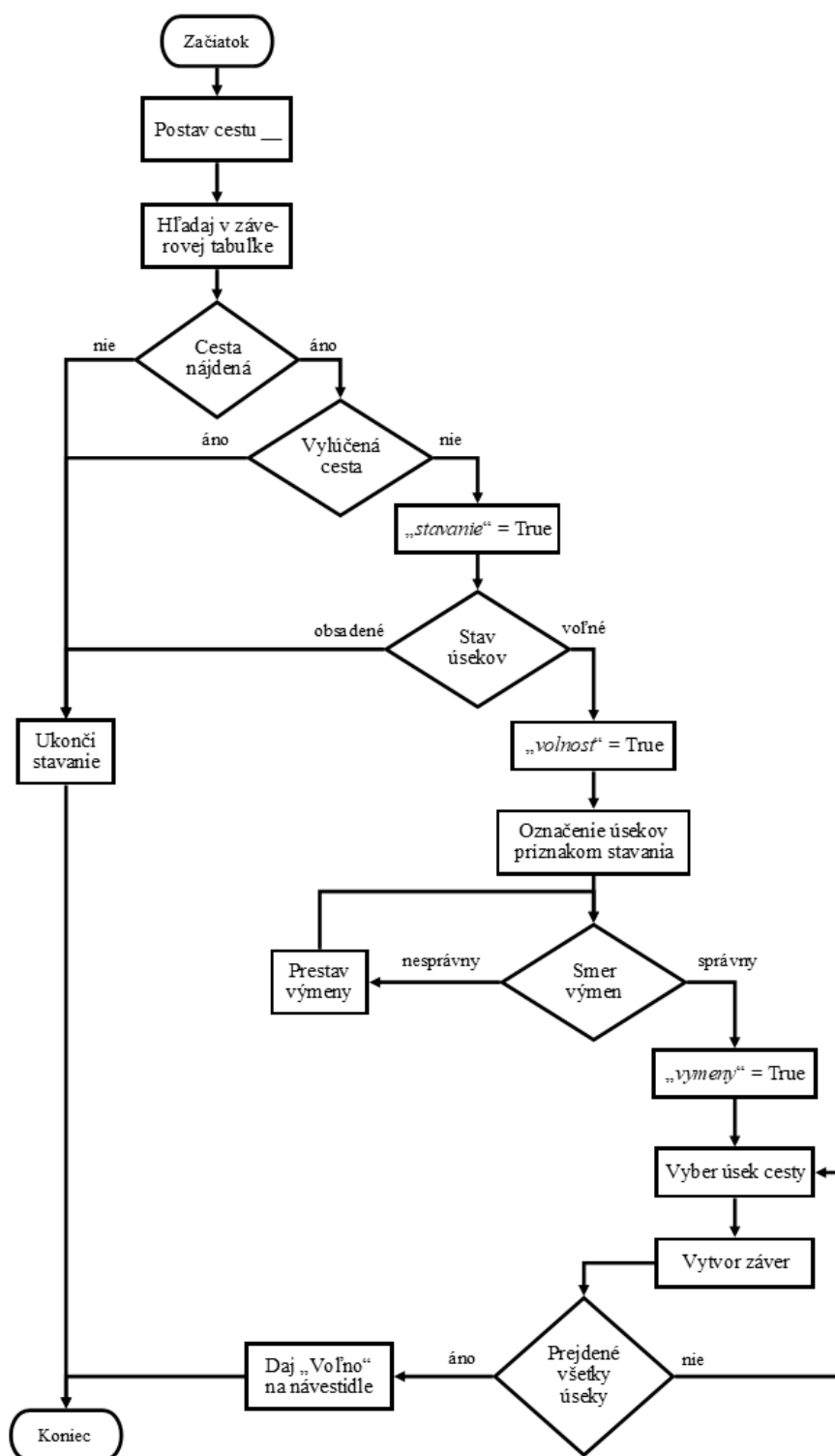
Obr. 9.22 Príklad rozhodovacieho stromu návěstidla S2 zo stanice Zbehy

Príloha I | Sekvenčný diagram automatického hradla

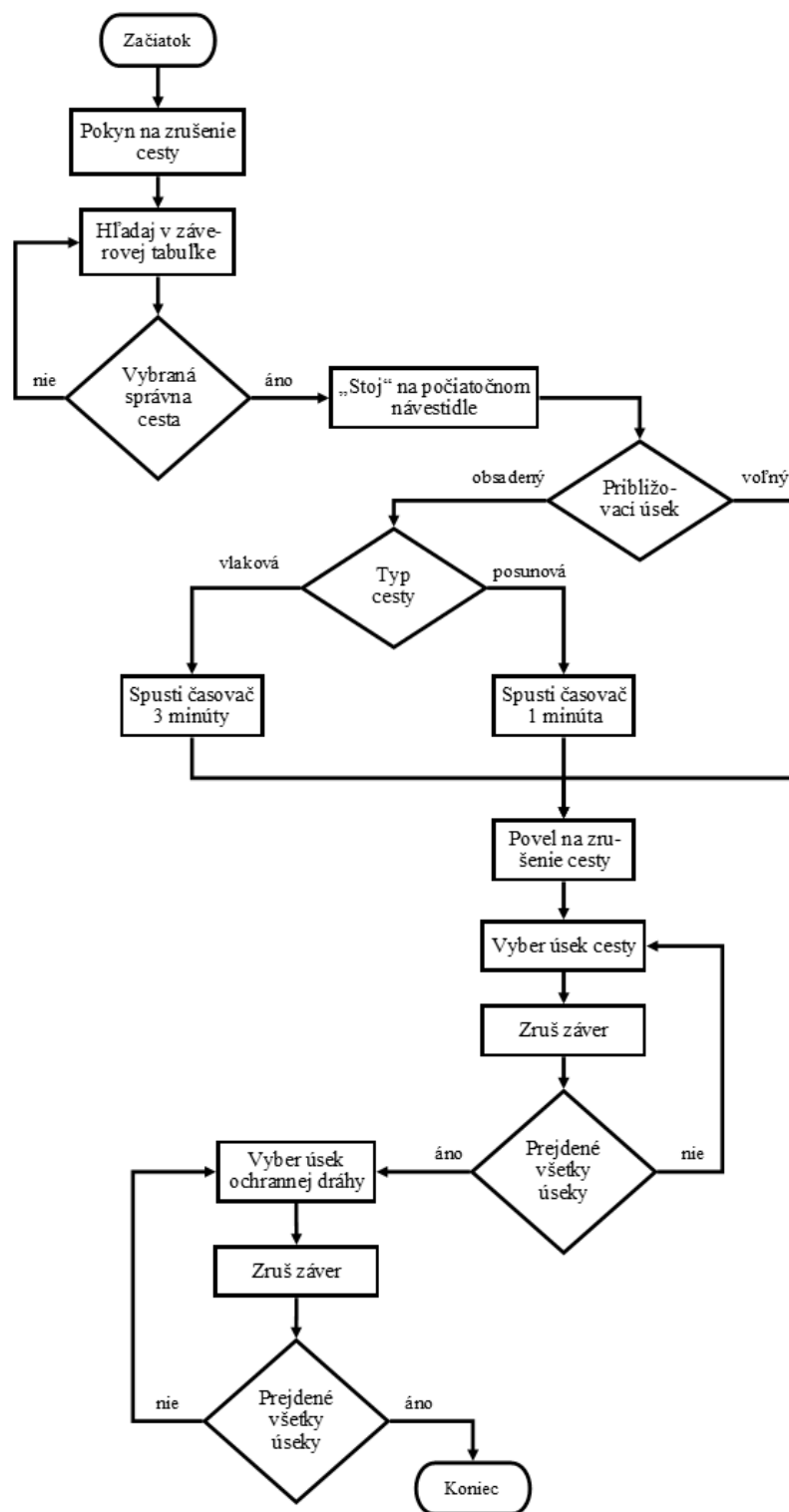


Obr. 9.23 Sekvenčný diagram automatického hradla

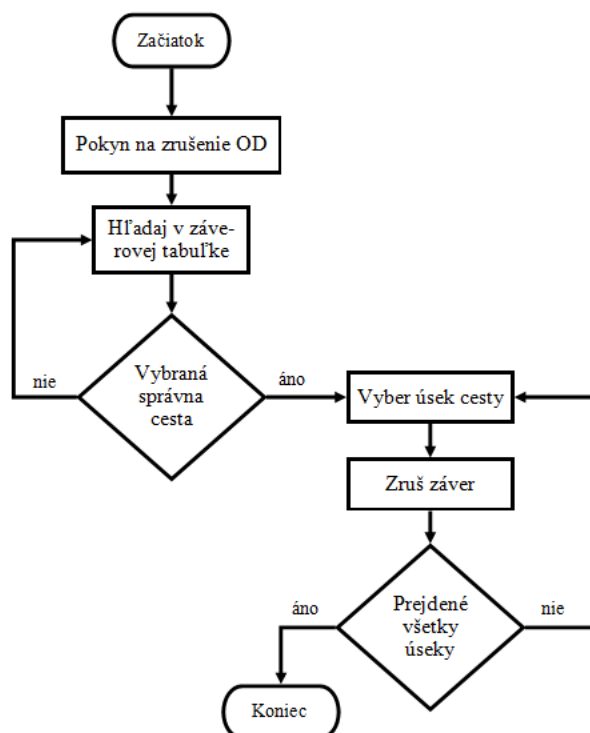
Príloha J | Vývojové diagramy systému SIMIS W



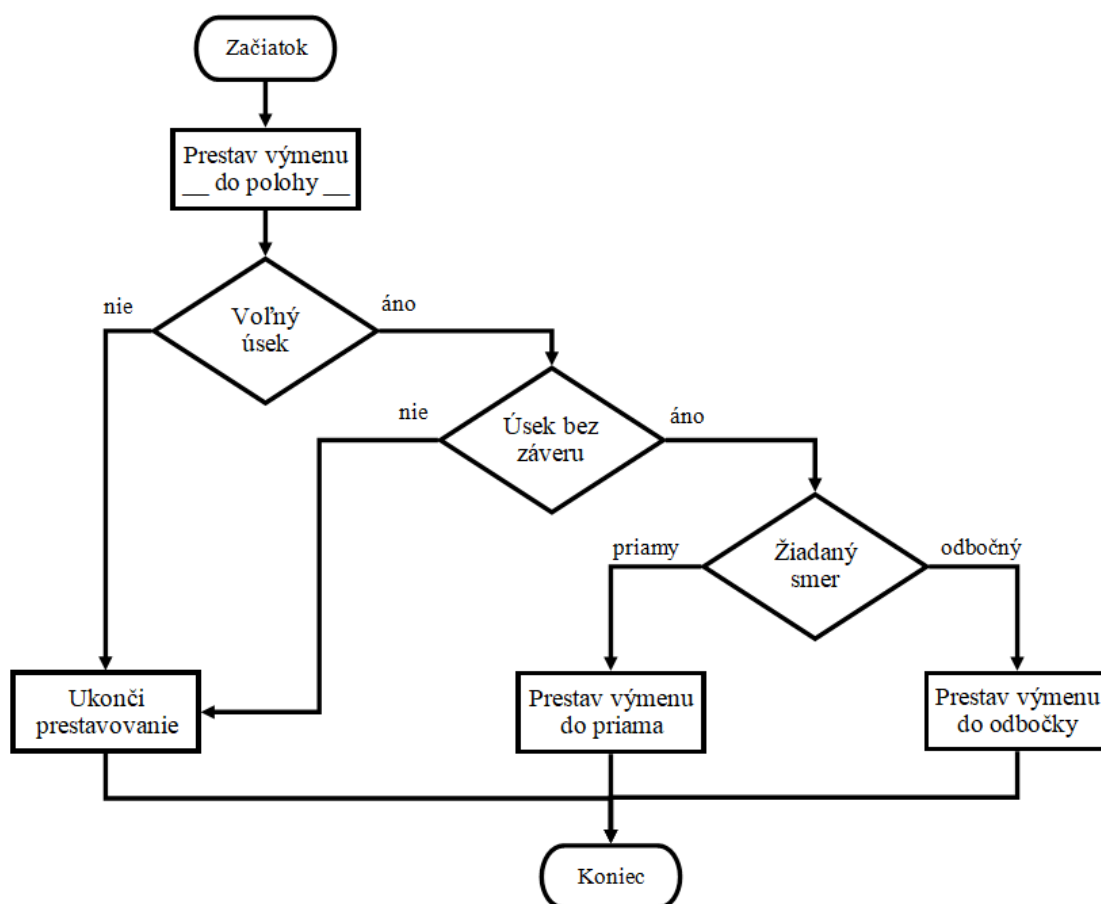
Obr. 9.24 Vývojový diagram funkcie „stavanieCesty“



Obr. 9.25 Vývojový diagram funkcie „rusenieCesty“

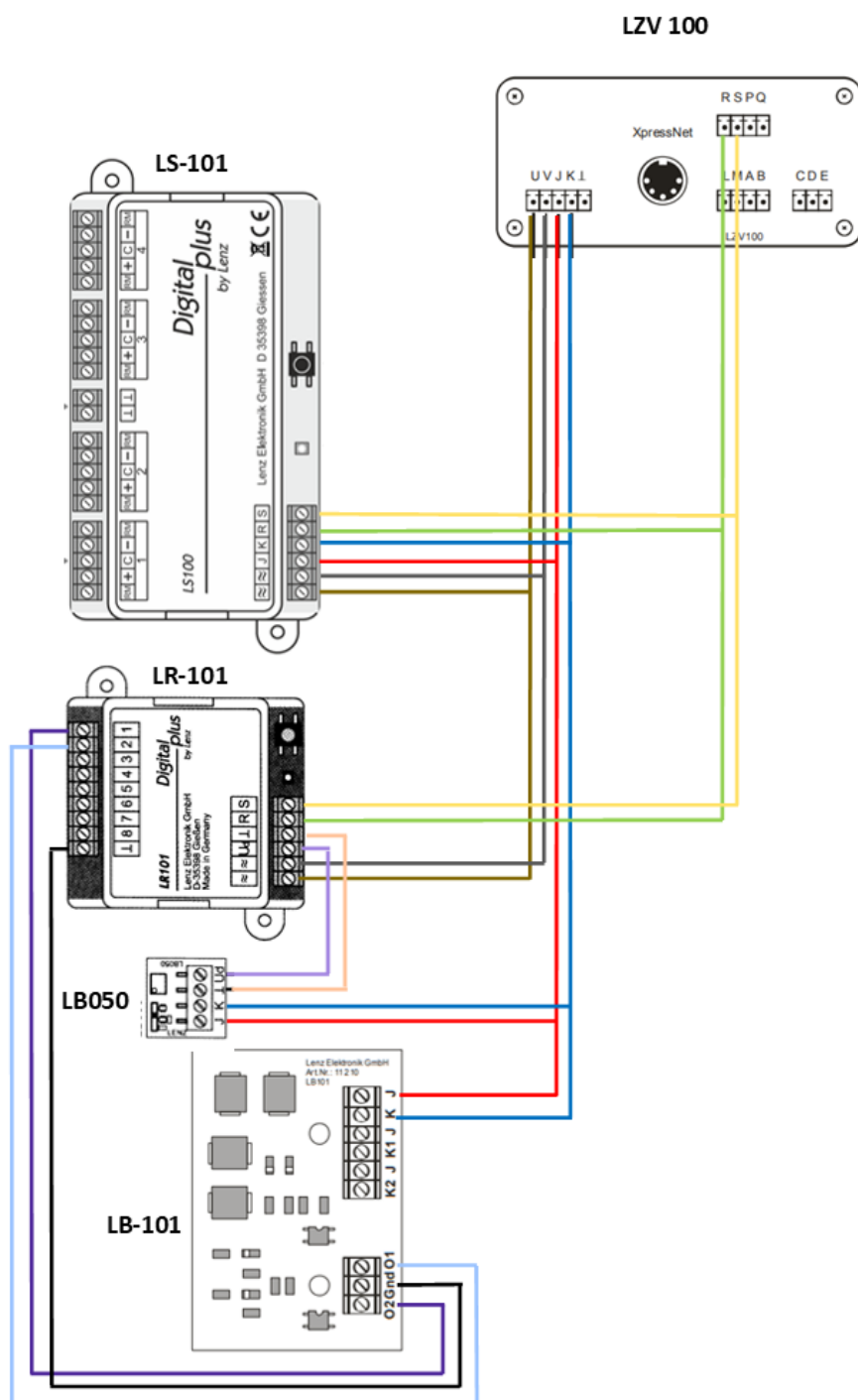


Obr. 9.26 Vývojový diagram funkcie „*rusenieOD*“



Obr. 9.27 Vývojový diagram funkcie „*prestavenieVyh*“

Príloha K | Pripojenie digitálneho riadenia trate riadenej systémom ESA-44



Obr. 9.28 Pripojenie modulov DCC riadenia ku zberniciam