Katedra informatiky Přírodovědecká fakulta Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Řízení modelového kolejiště

Digitalizace analogového ovládání



2018 Vedoucí práce: Mgr. Petr Krajča, Ph.D. Mgr. Jaromír Loun Studijní obor: Aplikovaná informatika, kombinovaná forma

Bibliografické údaje

Autor:	Mgr. Jaromír Loun					
Název práce:	Řízení modelového kolejiště (Digitalizace analogového ovládání)					
Typ práce:	bakalářská práce					
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci					
Rok obhajoby:	2018					
Studijní obor:	Aplikovaná informatika, kombinovaná forma					
Vedoucí práce:	Mgr. Petr Krajča, Ph.D.					
Počet stran:	64					
Přílohy:	1 CD/DVD					
Jazyk práce:	český					

Bibliograhic info

Author:	Mgr. Jaromír Loun
Title:	Model track management (Digitalization of analog control)
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2018
Study field:	Applied Computer Science, combined form
Supervisor:	Mgr. Petr Krajča, Ph.D.
Page count:	64
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Tato bakalářská práce se zabývá řízením modelového kolejiště. Obsahuje návrh modulů pro ovládání analogových součástí kolejiště s použitím levných běžně dostupných součástek. Práce se dále zabývá vytvořením komunikačního protokolu mezi počítačem a ovládacími moduly. Součástí práce je i sestavení aplikace pro řízení modelového kolejiště, která využívá moduly realizované dle návrhu a využívá všechny možnosti navrženého protokolu.

Synopsis

This bachelor thesis deals with the management of a model track. It includes a concept discribing modules for the control of analog track parts with the use of cheap, commonly available components. The thesis includes a communication protocol between the computer and the control modules. A part of the thesis is also the construction of an application for the control of the model track which uses the modules built according to the concept and uses all the possibilities of the proposed protocol.

Klíčová slova: modelová železnice; řízení kolejiště; mikroprocesor; ATmega8

Keywords: model railway; track control; microprocessor; ATmega8

Děkuji vedoucímu bakalářské práce Mgr. Petru Krajčovi, Ph.D. za vstřícný přístup a cenné rady a připomínky, které mi při zpracování mé bakalářské práce poskytl. Dále děkuji také Mgr. Michalu Dudkovi za uvedení do problematiky mikroprocesorů a odborné rady při návrhu ovládacích modulů. Také děkuji své rodině za podporu a pomoc během studia a zpracování bakalářské práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvo	d	8
2	Náv	rh ovládání kolejiště	9
	2.1	Výchozí stav	9
	2.2	Vymezení ovládaných prvků	9
	2.3	Koncept modulů	9
	2.4	Realizace ovládacích modulů	10
	2.5	Komunikace modulů po sběrnici I^2C	13
	2.6	Řídící modul pro komunikaci s počítačem	13
		2.6.1 Komunikační protokol pro rozhraní USART/RS232	14
		2.6.2 Komunikační protokol pro rozhraní I^2C	19
	2.7	Modul pro ovládání diod	21
	2.8	Modul pro ovládání serv	22
	2.9	Modul pro ovládání diod a serv	23
	2.10	Modul pro ovládání kolejových úseků	24
	2.11	Modul pro ovládání stejnosměrných motorů	25
3	Pro	gramování mikroprocesorů Atmel AVR ATmega8	27
4	Apli	ikace pro ovládání kolejiště	29
	4.1	Hlavní okno aplikace	30
	4.2	Menu Soubor	30
	4.3	Menu Nastavení	31
		4.3.1 Zobrazit systémová hlášení	31
		4.3.2 Vždy nahoře	31
		4.3.3 Nastavení sériového portu	31
		4.3.4 Definované konfigurace diody	32
		4.3.5 Definované konfigurace serva	33
		4.3.6 Definované konfigurace motoru	34
	4.4	Menu Ovládací moduly	35
		4.4.1 Ovládací moduly	35
		4.4.2 Výstupy pro diody	37
		4.4.3 Výstupy pro serva	37
		4.4.4 Výstupy pro kolejové úseky	38
		4.4.5 Výstupy pro motory	39
		4.4.6 Vstupy pro tlačítka	39
		4.4.7 Vstupy pro enkodéry	40
		4.4.8 Vstupy pro ampérmetry	40
		4.4.9 Vstupy pro ovladače motorů	40
	4.5	Menu Modelované prvky	41
	-	4.5.1 Diody	41
		4.5.2 Serva	42
		4.5.3 Kolejové úseky	42

		4.5.4	Motory	42
		4.5.5	Tlačítka	43
		4.5.6	Enkodéry	43
		4.5.7	Ampérmetry	44
		4.5.8	Ovladače motorů	44
		4.5.9	Budovy, oblasti	45
		4.5.10	Výhybky	45
	4.6	Menu	Kolejiště	46
		4.6.1	Události kolejiště	46
		4.6.2	Automatické změny	47
		4.6.3	Modelový čas	48
		4.6.4	Plán kolejiště	49
Zá	věr			54
Co	onclu	isions		55
\mathbf{A}	Příl	oha –	schéma řídícího modulu	56
в	Příl	oha –	schéma ovládacího modulu D	57
\mathbf{C}	Příl	oha –	schéma ovládacího modulu ST	58
D	Příl	oha –	schéma ovládacího modulu SDT	59
\mathbf{E}	Příl	oha –	schéma ovládacího modulu U	60
\mathbf{F}	Příl	oha –	schéma ovládacího modulu MTEA	61
\mathbf{G}	Obs	ah při	loženého CD	62
Lit	erat	ura		63

Seznam obrázků

1	Schéma ovládání.
2	Realizace řídícího modulu pro komunikaci s počítačem 14
3	Realizace modulu pro ovládání diod
4	Realizace modulu pro ovládání serv
5	Realizace modulu pro ovládání serv a diod
6	Realizace modulu pro ovládání kolejových úseků
7	Realizace modulu pro ovládání stejnosměrných motorů 25
8	Hlavní okno aplikace
9	Nastavení sériového portu
10	Definované konfigurace diody
11	Definované konfigurace serva
12	Definované konfigurace motoru
13	Ovládací moduly
14	Výstupy pro diody
15	Výstupy pro serva
16	Výstupy pro kolejové úseky
17	Výstupy pro motory
18	Vstupy pro tlačítka
19	Vstupy pro enkodéry 40
20	Vstupy pro ampérmetry 40
21	Vstupy pro ovladače motorů
22	Diody 41
23	Serva
24	Kolejové úseky
25	Motory
26	Tlačítka
27	Enkodéry
28	Ampérmetry
29	Ovladače motorů
30	Budovy, oblasti
31	Výhybky
32	Události kolejiště
33	Automatické změny
34	Modelový čas
35	Plán kolejiště
36	Schéma řídícího modulu pro komunikaci s počítačem
37	Schéma modulu pro ovládání diod
38	Schéma modulu pro ovládání serv
39	Schéma modulu pro ovládání serv a diod
40	Schéma modulu pro ovládání kolejových úseků 60
41	Schéma modulu pro ovládání stejnosměrných motorů 61

Seznam tabulek

1	Přehled ovládacích modulů	12
2	Nastavení rychlosti sériového přenosu RS232	14
3	Zprávy pro testování ovládacích modulů	15
4	Zprávy pro nastavení ovládacích modulů	16
5	Zprávy pro zjištění hodnot z ovládacích modulů	17
6	Zpráva pro změnu adresy ovládacího modulu	17
7	Zprávy pro PC	19
8	Zprávy I^2C pro ovládací moduly	21

Seznam zdrojových kódů

1	Příklad programu	pro mikroprocesor	ATmega8			27
---	------------------	-------------------	---------	--	--	----

1 Úvod

Mnoho modelových kolejišť bylo vytvořeno v době, kdy použití počítače pro jejich řízení nebylo možné. Modeláři dodnes používají analogová ovládání na různém stupni složitosti. S narůstajícím množstvím ovládaných prvků se zapojení značně komplikuje.

Modeláři, kteří se rozhodnou v dnešní době postavit nové kolejiště, si mohou vybrat z několika ucelených systémů digitálních ovládání kolejiště, jejichž základ tvoří centrální řídící jednotka¹ (např. LENZ LZV100 [4]), ke které je připojeno napájení kolejí a další moduly pro ovládání prvků kolejiště (výhybky, semafory, osvětlení, ovladače lokomotiv, ...). Některé centrální řídící jednotky dovolují i ovládání kolejiště pomocí počítače, mobilního telefonu nebo tabletu (např. ROCO Z21 [5]). Tyto systémy nabízí modelářům jednodušší elektrické zapojení a věrnější simulaci reálných dějů, avšak představují nemalé finanční náklady.

Cílem práce je ukázat, jak je možné realizovat počítačem podporované ovládání kolejiště svépomocí za použití běžně dostupných elektronických součástek na úrovni srovnatelné s komerčně dostupnými systémy bez použití centrální řídící jednotky.

 $^{^1{\}rm Centrální řídící jednotka tvoří základ digitálního řízení. Na základě pokynů z ovladačů nebo z počítače generuje příkazy pro řízení lokomotiv a příslušenství.$

2 Návrh ovládání kolejiště

2.1 Výchozí stav

Výchozím stavem modelového kolejiště pro tuto bakalářskou práci je soubor ovládaných prvků (lokomotiv, výhybek, osvětlení staveb, spínačů a tlačítek) v čistě analogovém provedení. Pro provoz kolejiště jsou typické činnosti:

- rozsvícení nebo zhasnutí osvětlení budovy
- změna světel na semaforu
- přestavění výhybky směr rovně nebo do odbočky
- otevření vrat nebo brány
- spuštění nebo zvednutí závory na přejezdu
- změna polohy mechanického návěstidla
- ovládání lokomotiv směr vpřed nebo vzad a rychlost
- přepínání izolovaného úseku kolejí k určitému ovladači lokomotivy
- stisk tlačítka např. pro sepnuté relé, aby se oddělený úsek kolejí připojil ke zdroji napájení
- přepnutí vypínače např. pro rozsvícení nebo zhasnutí osvětlení budovy
- pootočení knoflíku na ovladači např. pro změnu rychlosti lokomotivy

2.2 Vymezení ovládaných prvků

Činnosti uvedené v kapitole 2.1 lze realizovat pomocí několika málo prvků:

- svítivá dioda světla na semaforu, osvětlení budovy
- modelářské servo přestavění výhybky, otevření vrat nebo brány, spuštění nebo zvednutí závory na přejezdu, změna polohy mechanického návěstidla
- motorek ovládání lokomotiv směr vpřed nebo vzad a rychlost
- relé přepínání izolovaného úseku kolejí k určitému ovladači lokomotivy
- tlačítko nebo vypínač pokyn obsluhy nebo signál čidla, které signalizuje koncovou polohu ramene serva
- $\bullet\,$ inkrementální enkodér² po
otočení knoflíku na ovladači např. pro změnu rychlosti lokomotivy

2.3 Koncept modulů

Návrh ovládání kolejiště vychází z předpokladu, že se počet ovládaných prvků v kolejišti může měnit. Proto jsou pro jednotlivé analogové prvky dle typických

²Inkrementální enkodér při otáčení generuje pulsy, jejichž počet odpovídá změně natočení, neříká nic o přesné pozici. Z generovaných pulzů lze určit směr otáčení. Inkrementální enkodér se používá např. pro scrollovací kolečko myši.

situací vytvořeny ovládací moduly, které se v kolejišti mohou vyskytnout vícenásobně podle počtu prvků v konkrétním kolejišti. Viz tabulka 1 na straně 12. Moduly na základě pokynů z počítače mění stav ovládaných prvků nebo naopak informují počítač o změně, která nastala.

Při navrhování modulu jsem zohlednil několik hledisek:

- sestavení modulu by mělo být finančně nenáročné
- modul by měl být sestaven z běžně dostupných součástek
- modul by měl být sestaven z několika málo součástek, aby jej i laik mohl vyrobit v amatérských podmínkách
- modul by měl ke komunikaci s počítačem využít již existující standardní rozhraní počítače
- komunikace s moduly by měla probíhat pomocí standardních sběrnic a protokolů
- modul by měl obsluhu ovládaných prvků zajistit sám bez přímé účasti počítače jen na základě jeho pokynů, aby při nárůstu počtu ovládaných prvků nedošlo k výraznému zhoršení odezvy, z čehož vyplývá, že by měl modul obsahovat vlastní mikroprocesor
- všechny moduly by měly mít stejný mikroprocesor, aby komunikace byla co nejjednodušší, z čehož vyplývá, že mikroprocesor by měl umožnit ovládání všech prvků uvedených v kapitole 2.1

Na základě těchto priorit byl jako základ modulu vybrán Mikroprocesor Atmel AVR ATmega8 [2], protože:

- patří k nejlevnějším na trhu
- jde o procesor, který je v prodejnách běžně skladem
- výrobce poskytuje výbornou dokumentaci s řadou příkladů
- obsahuje jednotku USART, které je určena pro komunikace s počítačem
- obsahuje jednotku I²C [3], která je určena pro vzájemnou komunikaci mikroprocesorů
- obsahuje několik vstupně výstupních portů pro tlačítka a diody
- obsahuje jednotku PWM pro řízení stejnosměrných motorů
- napájení mikroprocesoru je +5 V s nízkou spotřebou proudu

Koncept ovládacích modulů jsem doplnil o jeden řídící modul, který bude po sériové lince RS232 komunikovat s počítačem a pomocí sběrnice I^2C (někdy bývá označována TWI) s ovládacími moduly. Viz obrázek 1.

2.4 Realizace ovládacích modulů

Základ řídícího modulu i ovládacího modulu tvoří osmibitový mikroprocesor Atmel ATmega8 s RISC architekturou, který má k dispozici 8 kB FLASH paměti pro program a 1 kB RAM paměti a 512 B EEPROM paměti pro data. Taktovací



Obrázek 1: Schéma ovládání.

frekvence mikroprocesoru je pomocí externího krystalu na všech modulech nastavena na maximální možnou hodnotu 16 MHz. Použito je provedení pouzdra DIP28.

Mikroprocesor nabízí 23 pinů pro komunikaci s okolím. V zapojeních, která jsou navržena pro ovládání kolejiště, jsou vždy dva piny (XTAL1, XTAL2) obsazeny externím krystalem. Dva piny (SCL, SDA) jsou vyhrazeny pro komunikaci s řídícím modulem. Na ovládacích modulech pro tlačítka nebo enkodéry je jeden pin použit pro signalizaci řídícímu modulu, že došlo ke změně na vstupech. Pro připojení ovládaných prvků zbývá 18 nebo 17 pinů. Protože je proudový odběr z mikroprocesoru omezen na maximálně celkových 200 mA, jsou některé ovládané prvky k pinům připojovány pomocí tranzistorových polí ULN2004A [9]. Řízení stejnosměrného motorku je realizováno pomocí H-můstku L298N [10]. Uvedené součástky jsou běžně dostupné na našem trhu.

Mikroprocesory na modulech jsou připojeny ke zdroji stabilizovaného napětí +5 V. H-můstky jsou napájeny ze zdroje stabilizovaného napětí +12 V. K napájení ostatních částí ovládacích modulů (diody, serva a relé) je možné použít napětí z rozhraní I²C. Vhodnější však je využít samostatný zdroj stabilizovaného napětí +5 V. Výběr způsobu napájení se provádí jumperem na ovládacím modulu (JP₂ na modulu D – viz obrázek 37, JP₁₃ na modulu ST – viz obrázek 38, JP₁₀ na modulu SDT – viz obrázek 39 a JP₂ na modulu U – viz obrázek 40).

Elektrická schémata ovládacích modulů a řídícího modulu jsem nakreslil po-

Тур	Použití v kolejišti	Ovládané prvky
Master	• komunikace mezi počítačem a ovlá- dacími moduly	
D	 osvětlení budov světelná návěstidla signalizace na ovládacím panelu 	• 18 LED
ST	 přestavení výhybky se zpětnou sig- nalizací polohy výhybky pohyb závor a blikání výstražných světel na přejezdu pohyb ramene jeřábu 	 6 modelářských serv 11 logických vstupů
U	 přepnutí kolejového úseku k urči- tému ovladači 	• 9 kolejových úseků přepínaných mezi třemi vstupy
MTEA	 ovládání lokomotivy pohyb lopatek větrného nebo vodního mlýnu 	 2 ovladače motoru po- mocí PWM 5 logických vstupů 2 rotační enkodéry
SDT	viz typ D a ST	 2 modelářská serva 11 LED 4 logické vstupy

Tabulka 1: Přehled ovládacích modulů

mocí programu KiCad EDA [18]. Součástky jsem rozmístil na univerzální pájivá pole a napájel běžnou transformátorovou páječkou. Při stavbě ovládacích modulů i řídícího modulu a jejich oživování jsem vycházel z prací [1] magistra Michala Dudky.

Navrhl jsem několik typů ovládacích modulů podle typických zapojení v kolejišti – viz tabulka1

2.5 Komunikace modulů po sběrnici I²C

Pro vzájemnou komunikaci mezi mikroprocesory jsem zvolil sběrnici I²C [3], jejíž podpora je implementována do ATmega8 výrobcem. Z možných konfigurací I²C sítě jsem vybral variantu, kdy komunikaci řídí pevně jeden mikroprocesor a k němu lze připojit až 127 dalších mikroprocesorů (případně jiných zařízení s podporou I²C sběrnice) s jedinečnými adresami v rozsahu 1 až 127. Propojení mikroprocesorů lze uspořádat do stromu. Mikroprocesor řídícího modulu je nastaven do módu master, ovládací moduly mají mikroprocesor vždy nastaven do módu slave.

Každý ovládací modul obsahuje právě jeden mikroprocesor, adresa mikroprocesoru tedy odpovídá adrese modulu. Po nahrátí softwaru do mikroprocesoru má modul přiřazenu adresu 127. Před připojením dalšího modulu do I²C sítě je nutné tuto adresu pomocí vytvořené aplikace Ovládání kolejiště změnit na hodnotu z intervalu <1;126> – viz kapitola 4.4.1. Tento princip umožňuje libovolnou kombinaci modulů v I²C síti.

Rychlost přenosu řídí SCL signál řídícího modulu, jeho frekvence je pevně nastavena na 100 kHz (maximální povolená frekvence je 400 kHz).

2.6 Řídící modul pro komunikaci s počítačem

Modul zprostředkovává komunikaci mezi počítačem a ovládacími moduly. Sám přímo nespolupracuje s žádným ovládaným prvkem v kolejišti. Jeho schéma je na obrázku 36.

Pro komunikaci s počítačem je využita jednotka USART, která je implementována v mikroprocesoru ATmega8 výrobcem. Z ní jsou použity pouze linky Rx a Tx. Logické úrovně jednotky USART 0 V a +5 V (respektive 0 V a U_{cc}) jsou pomocí obvodu MAX232 [8] upraveny na hodnoty používané ve standardním sériovém portu počítače +10 V a -10 V. Sériový port RS232 byl na starších počítačích běžný, dnes však většinou chybí a je potřeba použít USB adaptér pro RS232 rozhraní.

Rychlost přenosu po sériové lince lze nakonfigurovat pomocí jumperů JP_0 , JP_1 a JP_2 – viz schéma 36 na straně 56. Ostatní parametry jsou nastaveny pevně: 8 bitů, 1 stop bit, bez handshakingu a bez parity.

JP ₀	JP ₁	JP_2	Rychlost přenosu
			v kilobitech za sekundu
			4 800
Х			9 600
	Х		14 400
Х	Х		19 200
		Х	28 800
Х		Х	38 400
	Х	Х	57 600
Х	Х	Х	76 800

Tabulka 2: Nastavení rychlosti sériového přenosu RS232



Obrázek 2: Realizace řídícího modulu pro komunikaci s počítačem.

2.6.1 Komunikační protokol pro rozhraní USART/RS232

Komunikace po rozhraní USART/RS232 probíhá mezi počítačem a řídícím modulem. Z počítače jsou odesílány zprávy o maximální délce 30 bytů zakončené znakem n. Zprávy obsahují požadavky na otestování ovládacích modulů, nastavení hodnot na ovládacích modulech nebo zjištění hodnot z ovládacích modulů, viz tabulky 3, 4 a 5. První dva byty určují typ požadavku, druhé dva byty určují adresu modulu v I²C síti. Po odeslání zprávy do řídícího modulu je zasílání dalších zpráv do řídícího modulu blokováno, dokud řídící modul nezašle signál o uvolnění sériové linky.

Řídící modul zprávu přijme, vyhodnotí, předá přes rozhraní I^2C (viz kapitola 2.6.2) ovládacímu modulu s požadovanou adresou a počká na odpověď

ovládacího modulu. Zpět do počítače jsou po rozhraní USART/RS232 odesílány zprávy zakončené ;\n. První dva byty určují typ odpovědi řídícího modulu. Zprávy jsou popsány v tabulce 7. Za odpovědí následuje po rozhraní USAR-T/RS232 ještě zpráva o uvolnění sériové linky.

Například zpráva pro otestování řídícího modulu má tvar: m_00\n

Odpověď na výše uvedený požadavek má tvar: TD:typ=Master.006;\n

Po odpovědi následuje signál o uvolnění linky: SL1; n

Typ zprávy	Formát	$Popis \qquad (\check{\mathtt{R}}\mathtt{M}=\check{\mathtt{r}}idici \; modul; \; \mathtt{OM}=ovl\check{\mathtt{a}}daci \; modul)$				
Test ŘM	m_00\n	ŘM vrátí do počítače svůj typ a verzi.				
Test OM	$m_A_0A_1 \setminus n$	ŘM předá do OM s adresou A_0A_1 požadavek				
	např. m_10\n	na vrácení hodnot, které jej popisují (typ,				
		verze, adresa a počty jednotlivých ovláda-				
		ných prvků).				
		Vrácené hodnoty ŘM předá do počítače.				
Test všech	ma\n	ŘM postupně předá OM s adresou v inter-				
OM		valu <1; 127> požadavek na vrácení hodnot,				
		které je popisují.				
		ŘM předá do počítače postupně přijaté od-				
		povědi, signál o uvolnění linky zašle až				
		po předání poslední odpovědi.				

Tabulka 3: Zprávy pro testování ovládacích modulů

zprávyKMNastavenímfA ₀ A ₁ F ₀ F ₁ \nnapř.mf10C8\nPWMzádavek na nastavení frekvence PWMsignálusignálu pro výstupy motorů na hodnotuNastaveníd_A ₀ A ₁ I ₀ I ₁ S ₀ \nMastaveníd_A ₀ A ₁ I ₀ I ₁ S ₀ \nNastavenídsS ₀ M ₀ M ₁ N ₀ N ₁ V ₀ V ₁ D ₀ \nMastavenídsS ₀ M ₀ M ₁ N ₀ N ₁ V ₀ V ₁ D ₀ \nMastavenídsS ₀ M ₀ M ₁ N ₀ N ₁ V ₀ V ₁ D ₀ \nMastavenís_A ₀ A ₁ I ₀ I ₁ T ₀ T ₁ P ₀ P ₁ \nMastavenís_A ₀ A ₁ I ₀ I ₁ T ₀ T ₁ P ₀ P ₁ \nMastavenís_A ₀ A ₁ I ₀ I ₁ M ₀ M ₁ \nnapř.s_10010065\ns_10010065\nsi ID I ₀ I ₁ na novou hodnotu P ₀ P ₁ za čas T ₀ T ₁ desetin sekundy.NastavenísiA ₀ A ₁ I ₀ I ₁ M ₀ M ₁ \nnapř.si D I ₀ I ₁ na novou hodnotu P ₀ P ₁ za čas T ₀ T ₁ desetin sekundy.NastavenísiA ₀ A ₁ I ₀ I ₁ M ₀ M ₁ \nnapř.si 1D I ₀ I ₁ na novou hodnotuservasaA ₀ A ₁ I ₀ I ₁ M ₀ M ₁ \nRaximálnínapř.např.si 100185\npolohysatueníservasaA ₀ A ₁ I ₀ I ₁ M ₀ M ₁ \nRaxievenísaA ₀ A ₁ I ₀ I ₁ M ₀ M ₁ \nnapř.si 100185\nzapisovánínapř.např.si 20001kolejovéhosatueníservasa 300185\nNastavenísaA ₀ A ₁ I ₀ I ₁ C ₀ \nRM předá do OM s adresou A ₀ A ₁ požadavek na nastavení maximální krajní polohyservasz 300010\nkolejovéhosa 203001\n <th>Тур</th> <th>Formát</th> <th>Popis (ŘM = řídící modul; OM = ovládací modul)</th>	Тур	Formát	Popis (ŘM = řídící modul; OM = ovládací modul)
Nastavení frekvence PWM signálumfA ₀ A ₁ F ₀ F ₁ \n napž.mf10C8\n mapž.mf10C8\n mapž.mf10C8\n mapž.mf10C8\n mapž.mf10C8\n mapž.d_400db\nŘM předá do OM s adresou A_0A_1 požadavek, ab pro výstup motorů na hodnotu F_0F_1 .Nastavení diodyd_A_0A_1I_0I_3O_N napž.d_400db\nŘM předá do OM s adresou A_0A_1 požadavek, aby pro výstup pro diodu s ID I_0I_1 přijal S_0 stylů.Nastavení diodydsS0_00_M1_N0_N1_V0_V1_D0_N advek, aby pro výstup pro diodu s ID I_0I_1 přijal S_0 stylů.Nastavení polohy servas_A_0A_1I_0I_1T_0T_1P_0P1_N napž. s_10010065\nŘM předá do OM s adresou A_0A_1 požadavek na nastavení výstup pro serva s $ID I_0I_1$ na novou hodnotu P_0P_1 za čas T_0T_1 desetin sekundy.Nastavení polohy servasiA_0A_1I_0I_1M_0M_1 n napž. si100185\n napž. sal00185\nŘM předá do OM s adresou A_0A_1 požadavek na nastavení minimální krajní polohy serva sID I_0I_1 na novou hodnotu M_0M_1 .Nastavení servaszA_0A_1I_0I_1M_0M_1 n napž. sal00185\n n napž. szl0011\n polohyŘM předá do OM s adresou A_0A_1 požadavek na nastavení minimální krajní polohy serva s ID I_0I_1 na novou hodnotu servaNastavení servaszA_0A_1I_0I_2O_N napž. szl0011\n pidek do OM s adresou A_0A_1 požadavek na zapisování poslední polohy serva sID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisování napž. u_20030\n napž. u_20030\n napž. os30010\nŘM předá do OM s adresou A_0A_1 požadavek na nastavení směru otáčení napž. os30010\n zádavek na nastavení směru otáčení motoruNastavení usekuo_A0A_1I_0I_1T_0T_1S_0S_1\n napž. os30010\nŘM předá do OM s adresou A_0A	zprávy		
frekvence PWM signálunapř. mf10C8\n žadavek na nastavení frekvence PWM signálu pro výstupy motorů na hodnotu F_0F_1 .Nastavení diodyd_ $\Delta_0A_1I_0T_1S_0$ \n např. d_400db\nRM předá do OM s adresou A_0A_1 požadavek, aby pro výstup pro diodu s ID I_0I_1 přijal S_0 stylů.Nastavení diodydSS_0M_0M_1N_0N_1V_0V_D_0\n např. d_401007640\nRM předá do OM s adresou A_0A_1 požadavek, aby pro výstup pro diodu s ID I_0I_1 přijal styl S_0 s parametry M_0M_1 , N_0N_1, V_0V_1 a D_0 .Nastavení polohys_A_0A_1I_0I_1T_0T_1P_0P_1\n např.RM předá do OM s adresou A_0A_1 požadavek na nastavení výstup pro serva s 1D I_0I_1 na novou hodnotu P_0P_1 za ča T_0T_1 desetin sekundy.Nastavení servasiA_0A_1I_0I_M_0M_1\n např. si100185\n polohyRM předá do OM s adresou A_0A_1 požadavek na nastavení minimální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení servasaA_0A_1I_0I_M_0M_1\n např. sz1001185\n polohyRM předá do OM s adresou A_0A_1 požadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení servaszA_0A_1I_0I_1Z_0\n např. sz10011\n poslední polohyRM předá do OM s adresou A_0A_1 požadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) např. os30010\n tákavek na připojení kolejového úseku s ID I_0I_1 na hodnotu S_0 .Nastavení vetkuosA_0A_1I_0I_1S_0\n např. os30010\n tákavek na nastavení směru otáčení mot toru s ID I_0I_1 na hodnotu S_0 .Nastavení otáčení motoruo_A_0A_1I_0I_1T_0T_S_0S_1\n např. os30010\n tadavek na	Nastavení	$mfA_0A_1F_0F_1 \setminus n$	ŘM předá do OM s adresou A_0A_1 po-
PWM signálusignálu pro výstupy motorů na hodnotu F_0F_1 .Nastavení diody $d_{A0}A_1I_0I_1S_0 n$ např. $d_400db n$ ŘM předá do OM s adresou A_0A_1 po- žadavek, aby pro výstup pro diodu s ID I_0I_1 přijal S_0 stylů.Nastavení diodydsS_0M_0M_1N_0N_1V_0V_1D_0 n ds01007640 nŘM předá do OM s adresou A_0A_1 poža- davek, aby pro výstup pro diodu s ID I_0I_1 přijal S_0 stylů.Nastavení polohys_A_0A_1I_0I_1T_0T_1P_0P_1 n např.ŘM předá do OM s adresou A_0A_1 poža- žadavek na nastavení výstup uro serva s ID I_0I_1 na novou hodnotu P_0P_1 za čas T_0T_1 desetin sekundy.Nastavení servasiA_0A_1I_0I_1M_0M_1 n např. si100185 n polohyŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu servaNastavení servasiA_0A_1I_0I_M_1 n např. si100185 n polohyŘM předá do OM s adresou A_0A_1 poža- davek na nastavení maximální krajní po- lohy serva s ID I_0I_1 na novou hodnotu servaNastavení servasizA_0A_1I_0I_M_1 n např. sil00185 n polohyŘM předá do OM s adresou A_0A_1 poža- davek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu serva sid J_0I_1 lo paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- servaNastavení vsetvau_A_0A_1I_0I_N_0 n např. u_20030 n úšekuŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID D_0 .Nastavení vsetvaosA_0A_1I_0I_N_0 n např. os30010 n až. os30010 n apř. os30010 n adavek na nastavení směru otáčení mo- toru s ID I_0I_1	frekvence	např. mf10C8\n	žadavek na nastavení frekvence PWM
$\begin{array}{l c c c c c c c c c c c c c c c c c c c$	PWM		signálu pro výstupy motorů na hodnotu
Nastavení diody $d_{A0}A_1I_0I_1S_0 \backslash n$ napž. $d_400db \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek, aby pro výstup pro diodu s ID I_0I_1 přijal S_0 stylů.Nastavení diody $dsS_0M_0M_1N_0N_V0_{V1}D_0 \backslash n$ ŘM předá do OM s adresou A_0A_1 poža- styluNastavení diody $ds01007640 \backslash n$ I_0I_1 přijal styl S_0 s parametry M_0M_1 , N_0N_1 , V_0V_1 a D_0 .Nastavení serva $s_A_0A_1I_0I_1T_0T_1P_0P_1 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení výstupu pro serva s s_10010065 \backslash nNastavení serva $siA_0A_1I_0I_1M_0M_1 \backslash n$ např. $si100185 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu P_0P_1 za ča davek na nastavení maximální krajní polohyNastavení serva $saA_0A_1I_0I_1M_0M_1 \backslash n$ např. $sa100185 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení serva $szA_0A_1I_0I_1Z_0 \backslash n$ n apř. $sz10011 \land n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení velu $o_A_0A_1I_0I_1S_0 \backslash n$ např. $o_30010 \land n$ ŘM předá do OM s adresou A_0A_1 poža- davek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení velu $o_A_0A_1I_0I_1S_0 \backslash n$ např. $o_30010 \backslash n$ ŘM předá do OM s adresou A_0A_1 poža- davek na nastavení směru otáčení mo- totačení mo- totácení	signálu		F_0F_1 .
$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	Nastavení	$d_A_0A_1I_0I_1S_0 n$	ŘM předá do OM s adresou A_0A_1 po-
Image:	diody	např. d_400db\n	žadavek, aby pro výstup pro diodu s ID
Nastavení styludsS $_0M_0M_1N_0N_1V_0V_1D_0 \setminus n$ apř.ŘM předá do OM s adresou A_0A_1 poža- davek, aby pro výstup pro diodu s ID I_0I_1 přijal styl S_0 s parametry M_0M_1 , N_0N_1, V_0V_1 a D_0 .Nastavení $s_A_0A_1I_0I_1T_0T_1P_0P_1 \setminus n$ např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení výstupu pro serva s $s_110010065 \setminus n$ Nastavení $siA_0A_1I_0I_1T_0T_1P_0P_1 \setminus n$ např.ŘM předá do OM s adresou A_0A_1 poža- davek na nastavení výstupu pro serva s T_0T_1 desetin sekundy.Nastavení $siA_0A_1I_0I_1M_0M_1 \setminus n$ např. si100185 \nŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $saA_0A_1I_0I_1M_0M_1 \setminus n$ maximální např. sa100185 \nŘM předá do OM s adresou A_0A_1 poža- davek na nastavení maximální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $saA_0A_1I_0I_1M_0M_1 \setminus n$ maximální např. sz10011 \nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisování např. u_20030 \nNastavení $o_A_0A_1I_0I_1O_1 \cap n$ např. os30010 \nŘM předá do OM s adresou A_0A_1 poža- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1 \cap n$ rychlosti např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.			$I_0 I_1$ přijal S_0 stylů.
stylunapř.davek, aby pro výstup pro diodu s ID I_0I_1 přijal styl S_0 s parametry M_0M_1 , N_0N_1 , V_0V_1 a D_0 .Nastavení $s_A_0A_1I_0I_1T_0T_1P_0P_1 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení výstupu pro serva s s_10010065 \nNastavení $siA_0A_1I_0I_1M_0M_1 \backslash n$ např. si100185 \nŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální např. si100185 \nNastavení $siA_0A_1I_0I_1M_0M_1 \backslash n$ maximálníŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastavenísaA_0A_1I_0I_1M_0M_1 \backslash n maximální např. sa100185 \backslash nŘM předá do OM s adresou A_0A_1 poža- davek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu servaNastaveníszA_0A_1I_0I_1Z_0 \backslash n např. sz10011 \backslash nŘM předá do OM s adresou A_0A_1 poža žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisování např. u_2003 \n např. u_2003 \nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ha hodnotu S_0 .Nastavenío_A_0A_1I_0I_1S_0 \backslash n např. os30010 \backslash n týchlosti např.ŘM předá do OM s adresou A_0A_1 poža davek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1 \backslash n tádevek na nastavení směru otáčení mo- toru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	Nastavení	$\texttt{dsS}_0\texttt{M}_0\texttt{M}_1\texttt{N}_0\texttt{N}_1\texttt{V}_0\texttt{V}_1\texttt{D}_0\backslash\texttt{n}$	ŘM předá do OM s adresou A_0A_1 poža-
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	stylu	např.	davek, aby pro výstup pro diodu s ID
Nastavení polohy $s_A_0A_1I_0I_1T_0T_1P_0P_1 \backslash n$ Apř.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení výstupu pro serva s ID I_0I_1 na novou hodnotu P_0P_1 za čas T_0T_1 desetin sekundy.Nastavení $siA_0A_1I_0I_1M_0M_1 \backslash n$ např. $si100185 \backslash n$ ŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $siA_0A_1I_0I_1M_0M_1 \backslash n$ např. $si100185 \backslash n$ ŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $saA_0A_1I_0I_1M_0M_1 \backslash n$ maximální např. $sa100185 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $szA_0A_1I_0I_1Z_0 \backslash n$ např. $sz10011 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení $u_A_0A_1I_0I_1O_0 \backslash n$ např. $u_20030 \backslash n$ ŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $o_A_0A_1I_0I_1S_0 \backslash n$ např. $os30010 \backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Notoru $o_A_0A_1I_0I_1T_0T_1S_0S_1 \backslash n$ řadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	diody	ds01007640\n	$I_0 I_1$ přijal styl S_0 s parametry $M_0 M_1$,
Nastavení polohy $s_A0A_1I_0I_1T_0T_1P_0P_1 \backslash nApř.ŘM předá do OM s adresou A_0A_1 po-žadavek na nastavení výstupu pro servas ID I_0I_1 na novou hodnotu P_0P_1 za časT_0T_1 desetin sekundy.NastaveníminimálnípolohysiA_0A_1I_0I_1M_0M_1 \backslash nnapř. sil00185 \backslash nŘM předá do OM s adresou A_0A_1 poža-davek na nastavení minimální krajní po-lohy serva s ID I_0I_1 na novou hodnotuM_0M_1.NastavenísatavenípolohysaA_0A_1I_0I_1M_0M_1 \backslash nnapř. sal00185 \backslash nŘM předá do OM s adresou A_0A_1 po-žadavek na nastavení maximální krajnípolohy serva s ID I_0I_1 na novou hodnotuM_0M_1.NastaveníservasaA_0A_1I_0I_1M_0M_1 \land nnapř. sal00185 \backslash nŘM předá do OM s adresou A_0A_1 po-žadavek na nastavení maximální krajnípolohy serva s ID I_0I_1 na novou hodnotuM_0M_1.NastaveníservaszA_0A_1I_0I_1Z_0 \land nnapř. szl0011 \nnapř. szl0011 \nnapř. szl0011 \nnapř. szl0011 \nnapř. u_20030 \nnapř. u_20030 \nmeřuŘM předá do OM s adresou A_0A_1 poža-davek na připojení kolejového úseku s IDI_0I_1 ke vstupu ovladače s ID O_0.Nastavenívsekuo_A_0A_1I_0I_1S_0 \land nnapř. os30010 \nndředá do OM s adresou A_0A_1 po-žadavek na nastavení směru otáčení mo-toru s ID I_0I_1 na hodnotu S_0.Nastavenívschostinapř.o_A_0A_1I_0I_1S_0S_1 \land NAM předá do OM s adresou A_0A_1 po-žadavek na nastavení směru otáčení mo-toru s ID I_0I_1 na stupeň S_0S_1 za časmotoru$			$N_0 N_1, V_0 V_1 a D_0.$
polohy servanapř. s_10010065\nžadavek na nastavení výstupu pro serva s ID I_0I_1 na novou hodnotu P_0P_1 za čas T_0T_1 desetin sekundy.NastavenísiA_0A_1I_0I_1M_0M_1\n např. si100185\nŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastavenísaA_0A_1I_0I_1M_0M_1\n např. sa100185\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohyNastavenísaA_0A_1I_0I_1M_0M_1\n např. sa100185\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastaveníszA_0A_1I_0I_1Z_0\n např. sz10011\nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0N např. u_2003\n davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníoSA_0A_1I_0I_S_N např. os30010\n vádavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A0A_1I_0I_T_0T_S_0S_1\n např.Nastavenío_A0A_1I_0I_T_0T_S_0S_1\n radavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	Nastavení	$s_A_0A_1I_0I_1T_0T_1P_0P_1\setminus n$	ŘM předá do OM s adresou A_0A_1 po-
serva $s_10010065\n$ s ID I_0I_1 na novou hodnotu P_0P_1 za čas T_0T_1 desetin sekundy.Nastavení $siA_0A_1I_0I_1M_0M_1\n$ ŘM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $saA_0A_1I_0I_1M_0M_1\n$ ŘM předá do OM s adresou A_0A_1 požadavek na nastavení maximální např. sa100185\nNastavení $saA_0A_1I_0I_1M_0M_1\n$ ŘM předá do OM s adresou A_0A_1 požadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $szA_0A_1I_0I_1Z_0\n$ ŘM předá do OM s adresou A_0A_1 požadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení $u_A0A_1I_0I_1O_n\n$ ŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0\n$ mapř. $os30010\n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- totačeníNastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1\n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- totačeníNastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1\n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	polohy	např.	žadavek na nastavení výstupu pro serva
Nastavení $SiA_0A_1I_0I_1M_0M_1 \setminus n$ $\bar{R}M$ předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $SaA_0A_1I_0I_1M_0M_1 \setminus n$ $\bar{R}M$ předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohyNastavení $SaA_0A_1I_0I_1M_0M_1 \setminus n$ $\bar{R}M$ předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu servaNastavení $SzA_0A_1I_0I_1Z_0 \setminus n$ $\bar{R}M$ předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení $u_A_0A_1I_0I_1O_1 \cap$ úseku $\bar{M}M$ předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení $u_A_0A_1I_0I_1O_1 \cap$ např. $u_20030 \setminus n$ $\bar{M}M$ předá do OM s adresou A_0A_1 po- žadavek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0 \cap$ např. $os30010 \setminus n$ $\bar{M}M$ předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1 \setminus n$ např. $\bar{M}M$ předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení toru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	serva	s_10010065\n	s ID I_0I_1 na novou hodnotu P_0P_1 za čas
Nastavení $siA_0A_1I_0I_1M_0M_1$ /nRM předá do OM s adresou A_0A_1 poža- davek na nastavení minimální krajní po- lohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $saA_0A_1I_0I_1M_0M_1$ /n např. sa100185\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $saA_0A_1I_0I_1M_0M_1$ /n např. sa100185\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení $szA_0A_1I_0I_1Z_0$ /n např. sz10011\nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení $u_A_0A_1I_0I_1O_n$ např. u_20030 \nŘM předá do OM s adresou A_0A_1 po- žadavek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0$ \n např. $os30010$ \nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1$ \n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na stupeň S_0S_1 za čas motoru			$T_0 T_1$ desetin sekundy.
$\begin{array}{llllllllllllllllllllllllllllllllllll$	Nastavení	$siA_0A_1I_0I_1M_0M_1 n$	RM předá do OM s adresou A_0A_1 poža-
polohy servalohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastavenísaA_0A_1I_0I_1M_0M_1 \n např. sa100185\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastaveníszA_0A_1I_0I_1Z_0 \n např. sz10011\nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0(n) např. u_20030 \n isekuŘM předá do OM s adresou A_0A_1 po- žadavek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníosA_0A_1I_0I_3(n) např. os30010 \n vádavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1 \n např.Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1 \n rychlostiŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	minimální	např. sil00185\n	davek na nastavení minimální krajní po-
serva M_0M_1 .NastavenísaA_0A_1I_0I_1M_0M_1\n např. sa100185\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastaveníszA_0A_1I_0I_1Z_0\n např. sz10011\nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0(n) např. u_20030\nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníosA_0A_1I_0I_0(n) např. os30010\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_T_T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoruNastavenío_A_0A_0I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	polohy		lohy serva s ID I_0I_1 na novou hodnotu
NastavenísaA_0A_1I_0I_1M_0M_1 \n např. sa100185 \nRM předá do OM s adresou A_0A_1 po- žadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastaveníszA_0A_1I_0I_1Z_0 \n např. sz10011 \nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0(n např. u_20030 \nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0(n např. u_20030 \nŘM předá do OM s adresou A_0A_1 po- žadavek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníosA_0A_1I_0I_1S_0(n např. os30010 \nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1 \n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	serva		M_0M_1 .
maximálnínapř. sa100185\n polohyžadavek na nastavení maximální krajní polohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .NastaveníszA_0A_1I_0I_2_0\n např. sz10011\nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0(n) např. u_20030\nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníosA_0A_1I_0I_S_0(n) např. os30010\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	Nastavení	$saA_0A_1I_0I_1M_0M_1 n$	RM předá do OM s adresou A_0A_1 po-
polohy servapolohy serva s ID I_0I_1 na novou hodnotu M_0M_1 .Nastavení zapisování např. sz10011\nŘM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení u_A0A1I0I100\n např. u_20030\nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení úsekuosA_0A_1I_0I_1S_0\n n např. os30010\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení otáčení otáčenío_A_0A_1I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	maximální	např. sa100185\n	žadavek na nastavení maximální krajní
serva M_0M_1 .NastaveníszA_0A_1I_0I_1Z_0/nŘM předá do OM s adresou A_0A_1 po-zapisovánínapř. sz10011/nžadavek na zapisování poslední polohyposledníserva s ID I_0I_1 do paměti EEPROM mi-polohyserva s ID I_0I_1 do paměti EEPROM mi-servazapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_0(nNastavenínapř. u_20030/núsekudavek na připojení kolejového úseku s IDnapř. os30010/nŘM předá do OM s adresou A_0A_1 po-směrunapř. os30010/nnapř. os30010/nžadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1/nNastavenío_A_0A_1I_0I_1T_0T_1S_0S_1/nNastavenío_300164ed/notáčenímotoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru	polohy		polohy serva s ID I_0I_1 na novou hodnotu
Nastavení zapisování $szA_0A_1I_0I_1Z_0\backslash n$ např. $sz10011\backslash n$ RM předá do OM s adresou A_0A_1 po- žadavek na zapisování poslední polohy serva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení úseku $u_A_0A_1I_0I_1O_0\backslash n$ např. $u_20030\backslash n$ ŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení úseku $osA_0A_1I_0I_1S_0\backslash n$ např. $os30010\backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení otáčení motoru $o_A_0A_1I_0I_1T_0T_1S_0S_1\backslash n$ např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	serva		M_0M_1 .
zapisovánínapř. sz10011\nžadavek na zapisování poslední polohyposlednísz10011\nžadavek na zapisování poslední polohypolohyserva s ID I_0I_1 do paměti EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_1O_\n např. u_20030\nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníosA_0A_1I_0I_1S_0\n např. os30010\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoruNastavenío_A_0A_1I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas motoru s ID I_0T_1 desetin sekundy.	Nastavení	$szA_0A_1I_0I_1Z_0 \setminus n$	RM předá do OM s adresou A_0A_1 po-
poslední polohyserva s ID I_0I_1 do pamětí EEPROM mi- kroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastavení úsekuu_A_0A_1I_0I_1O_\n např. u_20030\nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení otáčení motoruosA_0A_1I_0I_S_0\n např. os30010\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení otáčení otáčení otáčení otáčení otáčenío_A_0A_1I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	zapisování	např. sz10011\n	žadavek na zapisování poslední polohy
polohykroprocesoru; Z_0 nabývá hodnoty 0 (ne- zapisovat) nebo 1 (zapisovat).Nastaveníu_A_0A_1I_0I_1O_\n např. u_20030\nŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .NastaveníosA_0A_1I_0I_1S_0\n např. os30010\nŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavenío_A_0A_1I_0I_1T_0T_1S_0S_1\n např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	poslední		serva s ID I_0I_1 do paměti EEPROM mi-
servazapisovat) nebo 1 (zapisovat).Nastavení $u_A_0A_1I_0I_1O_n$ ŘM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0 \setminus n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1 \setminus n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoruNastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1 \setminus n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	polohy		kroprocesoru; Z_0 nabývá hodnoty 0 (ne-
Nastavení $u_{A_0A_1I_0I_1O_n}$ RM předá do OM s adresou A_0A_1 poža- davek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0 \ n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na připojení kolejového úseku s ID I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0 \ n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1 \ např.$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoruNastavení $o_300164ed \ n$ motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	serva		zapisovat) nebo 1 (zapisovat).
kolejovéhonapř. u_20030\ndavek na připojení kolejového úseku s IDúseku I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $osA_0A_1I_0I_1S_0\backslash n$ ŘM předá do OM s adresou A_0A_1 po-směrunapř. $os30010\backslash n$ žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1\backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na stupeň S_0S_1 za čas motoruNastavení $o_300164ed\backslash n$ motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	Nastavení	$u_A_0A_1I_0I_1O_n$	RM předá do OM s adresou A_0A_1 poža-
úseku I_0I_1 ke vstupu ovladače s ID O_0 .Nastavení $0 \le A_0A_1I_0I_1S_0 \setminus n$ ŘM předá do OM s adresou A_0A_1 po-směrunapř. $0 \le 30010 \setminus n$ žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .notoru $0 _ A_0A_1I_0I_1T_0T_1S_0S_1 \setminus n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na stupeň S_0S_1 za čas motoruNastavení $0_ 300164 ed \setminus n$ motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	kolejového	např. u_20030\n	davek na připojení kolejového úseku s ID
Nastavení $osA_0A_1I_0I_1S_0\backslash n$ např. $os30010\backslash n$ RM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .motoruNastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1\backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1\backslash n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	úseku	<u> </u>	I_0I_1 ke vstupu ovladače s ID O_0 .
směru otáčení motorunapř. os30010\n žadavek na nastavení směru otáčení mo- toru s ID I_0I_1 na hodnotu S_0 .Nastavení rychlosti otáčení motoru $\circ_A_0A_1I_0I_1T_0T_1S_0S_1\backslash n$ např.ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	Nastavení	$osA_0A_1I_0I_1S_0 n$	RM předá do OM s adresou A_0A_1 po-
otáčení motorutoru s ID I_0I_1 na hodnotu S_0 .Nastavení rychlosti $o_A_0A_1I_0I_1T_0T_1S_0S_1 \setminus n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	směru	např. os30010\n	žadavek na nastavení směru otáčení mo-
motoruMastavení $o_A_0A_1I_0I_1T_0T_1S_0S_1\n$ ŘM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	otáčení		toru s ID I_0I_1 na hodnotu S_0 .
Nastavení rychlosti $o_A_0A_1I_0I_1T_0T_1S_0S_1 \setminus n$ RM předá do OM s adresou A_0A_1 po- žadavek na nastavení rychlosti otáčení motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	motoru		×
rychlostinapř.žadavek na nastavení rychlosti otáčeníotáčení o_300164 ed\nmotoru s ID I_0I_1 na stupeň S_0S_1 za časmotoru T_0T_1 desetin sekundy.	Nastavení	$o_A_0A_1I_0I_1T_0T_1S_0S_1 n$	RM předá do OM s adresou A_0A_1 po-
otáčení o_300164ed\n motoru s ID I_0I_1 na stupeň S_0S_1 za čas T_0T_1 desetin sekundy.	rychlosti	např.	žadavek na nastavení rychlosti otáčení
motoru $T_0 T_1$ desetin sekundy.	otáčení	o_300164ed\n	motoru s ID I_0I_1 na stupeň S_0S_1 za čas
	motoru		$T_0 T_1$ desetin sekundy.

Ta	bulk	a 4:	Zprá	vy pr	o nast	avení (ovlác	lacích	modu	lů
----	------	------	------	-------	--------	---------	-------	--------	------	----

Typ zprávy	Formát	$Popis \qquad (\check{\mathtt{R}}\mathtt{M}=\check{\mathtt{r}}\check{\mathtt{id}}\check{\mathtt{ic}}\check{\mathtt{modul}}; \mathtt{OM}=\mathtt{ovl}\check{\mathtt{adac}}\check{\mathtt{modul}})$
Zjištění	v_A ₀ A ₁ I ₀ I ₁ \n	ŘM předá do OM s adresou A_0A_1 požadavek
stavu	např.	na zjištění stavu vstupu pro tlačítko s ID $I_0 I_1$.
vstupu	v_1000\n	Vrácenou hodnotu ŘM předá do počítače.
Zjištění	$mvA_0A_1 \setminus n$	ŘM předá do OM s adresou A_0A_1 požadavek
stavu	např. mv10\n	na zjištění stavu všech vstupů pro tlačítka.
vstupů		Vrácenou hodnotu, která obsahuje stavy
		vstupů souhrnně, ŘM předá do počítače.
Zjištění	$meA_0A_1 \setminus n$	ŘM předá do OM s adresou A_0A_1 požadavek
hodnot	např. me10\n	na zjištění hodnot všech enkodérů.
enkodérů		ŘM vrácené hodnoty složí do jedné zprávy a
		předá je souhrnně do počítače.
Zjištění	$ouA_0A_1 \setminus n$	ŘM předá do OM s adresou A_0A_1 požadavek
hodnot	např. oul0\n	na zjištění hodnot všech ampérmetrů motorů.
ampérmetrů		ŘM vrácené hodnoty složí do jedné zprávy a
motorů		předá je souhrnně do počítače.

Tabulka 5: Zprávy pro zjištění hodnot z ovládacích modulů

Typ zprávy	Formát	Popis (ŘM = řídící modul; OM = ovládací modul)
Změna	$mxA_0A_1A_2A_3 n$	ŘM předá do OM s adresou A_0A_1 požadavek
adresy OM	např.	na změnu jeho adresy na hodnotu A_2A_3 .
	mx5051\n	
T	1 11 0 77 /	× 1 1/1 /1 1 1

Tabulka 6: Zpráva pro změnu adresy ovládacího modulu

Typ zprávy	Popis (ŘM = řídící modul; OM = ovládací modul)
Odpověď	TD: <r<sub>0>; \n</r<sub>
na test	např. TD:Master.006;\n
řídícího	
modulu	Odpověď obsahuje verzi řídícího modulu v řetězci r_0 .

Odpověď	$TDA_0A_1: < r_0 > : T_0T_1E_0E_1A_0A_1O_0O_1D_0D_1S_0S_1U_0U_1M_0M_1E_0E_1: \n$
na test	
ovládacího	např TD10.D 001.000000000000000000000000000000
modulu	
modulu	Odpověď obsahuje adresu 4.4. ovládacího modulu jeho
	vorzi r_{2} a počot vetupů a výstupů
	Význam pozic:
	T T požet vetupů pro tlažítka
	$\Gamma_0 \Gamma_1 = \text{pocer vstupu pro tractica}$
	$E_0 E_1 = \text{pocer vstupu pro ekodery}$
	A_0A_1 – pocet vstupu pro amperimetry
	O_0O_1 – pocet vstupu pro ovladace motoru
	D_0D_1 – počet vystupu pro dlody
	S_0S_1 – počet výstupů pro serva
	U_0U_1 – počet výstupů pro kolejové úseky
	M_0M_1 – počet výstupů pro motory
	F_0F_1 – frekvence PWM pro motory.
Přijetí stavu	$HVA_0A_1I_0I_1H_0; \n$
tlačítka	např. HV20011;\n
	Odpověď obsahuje stav ${\cal H}_0$ vstupu pro tlačítko s ID ${\cal I}_0{\cal I}_1$
	na OM s adresou A_0A_1 .
Přijetí stavu	$HLA_0A_1; n$
tlačítek	např. HL20;\n
	Odpověď obsahuje stav všech vstupů pro tlačítko na OM s ad-
	resou A_0A_1 .
Přijetí hodnot	$HEA_0A_1; n$
enkodérů	např. HE30;\n
	Odpověď obsahuje přírůstky všech enkodérů na OM s adresou
	A_0A_1 .
Přijetí hodnot	$ADA_0A_1\ldots; n$
ampérmetrů	$např. AD40: \n$
anip of moor a	
	Odpověď obsahuje hodnoty všech ampérmetrů na OM s ad-
	resource observations in statistically been competitional and the statistical resource A_0A_1 .
Přijetí signálu	SL1:\n
o uvolnění	
sériové linky	ŘM předá informaci, že je možné posílat do ŘM další zprávy
Přijetí stavu tlačítka Přijetí stavu tlačítek Přijetí hodnot enkodérů Přijetí hodnot ampérmetrů Přijetí signálu o uvolnění sériové linky	$\begin{array}{l} O_0O_1 & - \operatorname{počet} vstupů pro ovladače motoru\\ D_0D_1 & - \operatorname{počet} výstupů pro diody\\ S_0S_1 & - \operatorname{počet} výstupů pro serva\\ U_0U_1 & - \operatorname{počet} výstupů pro kolejové úseky\\ M_0M_1 & - \operatorname{počet} výstupů pro motory\\ F_0F_1 & - \operatorname{frekvence} PWM pro motory.\\ HVA_0A_1I_0I_1H_0; \n \\ např. HV20011; \n \\ Odpověď obsahuje stav H_0 vstupu pro tlačítko s ID I_0I_1na OM s adresou A_0A_1.HLA_0A_1; \n \\ např. HL20; \n \\ Odpověď obsahuje stav všech vstupů pro tlačítko na OM s adresou A_0A_1.HEA_0A_1; \n \\ např. HE30; \n \\ Odpověď obsahuje přírůstky všech enkodérů na OM s adresou A_0A_1.SL1; \n \\ ŘM předá informaci, že je možné posílat do ŘM další zprávy. \\ \end{array}$

Přijetí signálu	SV1;\n
o neošetřené	
změně někte-	ŘM předá informaci, že na některém OM došlo ke změně
rého vstupu	stavu vstupu pro tlačítko nebo hodnoty enkodéru.
Přijetí signálu	SV0;\n
o ošetření	
změn vstupů	ŘM předá informaci, že byly ze všech OM přečteny změny
	stavu vstupů pro tlačítko i změny hodnot enkodérů.
-	

Tabulka 7: Zprávy pro PC

2.6.2 Komunikační protokol pro rozhraní I²C

Komunikace po rozhraní I²C [3] slouží k přenosu zpráv mezi řídícím modulem a ovládacími moduly. Komunikace tam i zpět probíhá po jednotlivých bytech a jejich příjem je vždy příjímajícím modulem potvrzen. Iniciátorem komunikace je vždy řídící modul, který předá typ požadavku do ovládacího modulu v jednom bytu, tzn. odešle jeden znak – viz tabulka 8. Po typu požadavku většinou následuje byt s id vstupu nebo výstupu. V dalších bytech jsou hodnoty parametrů z intervalu <0; 250>. Zbývající hodnoty z intevalu <251; 255> jsou vyhrazeny pro přenos speciálních příznaků, což zatím nebylo použito.

Signalizace změny stavu vstupu tlačítka nebo enkodéru z ovládacího modulu na řídící modul je realizována elektronickou cestou na jeden pin řídícího modulu, jehož hodnotu si řídící modul neustále testuje. Pin řídícího modulu je společný pro všechny ovládací moduly, tzn. řídící modul přesně neví, na kterém modulu ke změně došlo. Pokud se objeví signál, že došlo ke změně, řídící modul předá informaci aplikaci v počítači. Aplikace začne cyklicky vyčítat z ovládacích modulů aktuální stav všech tlačítek a enkodérů v I²C síti. Ovládací modul po vyčtení stavu tlačítek a enkodérů zruší elektronickou signalizaci řídícímu modulu. Cyklické vyčítání probíhá tak dlouho, dokud se signalizace změny neukončí po vyčtení všech změněných hodnot ze všech modulů.

Typ zprávy		Požadavek	Odpověď		
Test	m	Požadavek nemá žádné pa-	Odpověď obsahuje typ		
ovládacího		rametry.	modulu a počty vstupů		
modulu			nebo výstupů jednotlivých		
			typů.		
Nastavení	d	Parametrem je id výstupu	Bez odpovědi.		
diody		pro diodu, počet přenáše-			
		ných části konfigurace di-			
		ody a pak čtveřice bytů			
		pro každou část konfigu-			
		race diody.			

Nastavení	S	Parametrem je id výstupu	Bez odpovědi.
polohy		pro servo, čas a poloha.	-
serva			
Nastavení	i	Parametrem je id výstupu	Bez odpovědi.
minimální		pro servo a posun mini-	
polohy		mální polohy.	
serva			
Nastavení	a	Parametrem je id výstupu	Bez odpovědi.
maximální		pro servo a posun maxi-	-
polohy		mální polohy.	
serva		1 0	
Nastavení	z	Parametrem je id výstupu	Bez odpovědi.
zapisování		pro servo a 0 – nezapisovat	-
poslední		nebo 1 – zapisovat.	
polohy		Ĩ	
serva do			
EEPROM			
Přepnutí	u	Parametrem je id výstupu	Bez odpovědi.
kolejového		pro kolejový úsek a identi-	-
úseku		fikace vstupu.	
Změna	0	Parametrem je id výstupu	Bez odpovědi.
rychlosti		pro motor, čas a stupeň	
otáčení		rychlosti.	
motoru			
Změna	r	Parametrem je id výstupu	Bez odpovědi.
směru		pro motor a identifikace	
otáčení		směru.	
motoru			
Nastavení	f	Parametrem je hodnota,	Bez odpovědi.
frekvence		která po přičtení 50 vy-	
PWM		jadřuje požadovanou frek-	
signálu		venci.	
pro motory			
Změna I ² C	х	Parametrem je nová ad-	Bez odpovědi.
adresy		resa.	
modulu			
Zjištění	V	Parametrem je id vstupu	Odpovědí je jeden byte s
stavu		pro tlačítko.	hodnotou $0 =$ rozepnuto
vstupu			nebo $1 = $ sepnuto.
pro tlačítko			

Zjištění	1	Bez parametru.	Hodnoty 0 = rozepnuto
stavu			nebo $1 =$ sepnuto jsou
vstupů			v pořadí vstupů vkládány
pro tlačítka			na pozice jednoho bytu
			(popř. několika), který byl
			předen vynulován, odpo-
			vědí je jeden (nebo ně-
			kolik) byt s kumulovanou
			hodnotou pro osm vstupů.
Zjištění	W	Bez parametru.	Hodnota na každém
hodnot			vstupu je převedena
proudů			do dvoubytového nezna-
na modulu			ménkového čísla, odpovědí
			je několik dvojic bytů
			podle počtu ampérmetrů.
Zjištění	е	Bez parametru.	Hodnota přírůstku na kaž-
přírůstků			dém vstupu je vyjádřena
enkodérů			jednobytovým číslem, kde
na modulu			hodnota 127 odpovídá nu-
			lovému přírůstku, odpo-
			vědí je několik bytů podle
			počtu enkodérů.

Tabulka 8: Zprávy I²C pro ovládací moduly

2.7 Modul pro ovládání diod

Modul ovládá 18 výstupů. Jeho schéma je na obrázku 37. Všechny výstupy slouží k připojení svítivých diod. Ke každému pinu mikroprocesoru pro diodu je připojen ochranný odpor 470 Ω a trimr 3,3 k Ω , kterým se nastaví maximální jas diody.

Pro každý výstup je možné programově nastavit zdánlivou intenzitu svitu diody rychlým střídáním zhasnutého a rozsvíceného stavu. K intenzitě je možné nastavit délku trvání. Celkově lze pro každou diodu definovat až 14 stavů popsaných intenzitou a délkou trvání. Ke každému stavu je ještě přidána informace, který z 14 stavů následuje po jeho skončení. Lze tedy realizovat trvalý svit diody určitou intenzitou, blikání mezi dvěma (nebo více) intenzitami nebo simulovat postupné rozsvěcení diody. Ve výchozím nastavení je dioda trvale zhasnuta.



Obrázek 3: Realizace modulu pro ovládání diod.

2.8 Modul pro ovládání serv

Modul ovládá 6 výstupů a 11 vstupů. Jeho schéma je na obrázku 38. Výstupy slouží k připojení modelářských serv s napájením 5 V. Mezi výstupní piny mikroprocesoru a serva je vložen tranzistorový můstek ULN2004A. Na vstupy mohou být připojeny přepínače nebo tlačítka. U vstupů jsou použity vnitřní pullup rezistory. Aby se omezily zákmity při sepnutí nebo rozepnutí tlačítka, jsou na vstupy připojeny filtrační kondenzátory 100 nF.

Pro každé servo je možné programově nastavit rychlost pohybu a polohu, do které se má přepnout. Výchozí je poloha uprostřed.

Ovládací modul testuje nepřetržitě stavy vstupů. Pokud nastane změna, předá řídícímu modulu informaci, že došlo ke změně. Tuto informaci pak řídící modul předá do PC. Aplikace zařídí vyčtení změn vstupů.



Obrázek 4: Realizace modulu pro ovládání serv.

2.9 Modul pro ovládání diod a serv

Modul ovládá 13 výstupů a 4 vstupy. Jeho schéma je na obrázku 39. Dva výstupy slouží k připojení modelářských serv s napájením 5 V. Jedenáct výstupů slouží pro ovládání diod. Na vstupy mohou být připojeny přepínače nebo tlačítka.

Popis zapojení a programování ovládacího modulu odpovídá popisu v kapitole 2.7 na straně 21 a kapitole 2.8 na straně 22.

Modul je kombinací modulu D a ST. Motivem k jeho návrhu je obsloužení malé železniční vlečky, kde se nachází většinou maximálně dvě výhybky a několik světel.



Obrázek 5: Realizace modulu pro ovládání serv a diod.

2.10 Modul pro ovládání kolejových úseků

Modul ovládá 9 výstupů pro kolejové úseky. Jeho schéma je na obrázku 40. Výstup je v každém okamžiku přepojen k jednomu ze tří vstupů. Výběr vstupu řídí pro jeden výstup dvě relé, která jsou k pinu mikroprocesoru připojena přes tranzistorový můstek ULN2004A. Výchozí je propojení se vstupem A.

To umožňuje rozdělit kolejiště na tři oblasti a v každé lokomotivu řídit nezávislým zdrojem napájení. Napájecí napětí lokomotivního motorku je max. 12 V, ale použitá relé mohou spínat až 125 V.

Předpokládané uplatnění:

- 1. tři analogové zdroje umožní nezávislé ovládání až tří analogových lokomotiv
- 2. dva analogové zdroje a jeden digitání umožní nezávislé ovládání dvou analogových lokomotiv a několika digitálních lokomotiv



Obrázek 6: Realizace modulu pro ovládání kolejových úseků.

2.11 Modul pro ovládání stejnosměrných motorů

Modul ovládá 2 výstupy a 9 vstupů. Jeho schéma je na obrázku 41. Dva vstupy slouží pro připojení enkodérů. Na zbývající vstupy mohou být připojeny přepínače nebo tlačítka.



Obrázek 7: Realizace modulu pro ovládání stejnosměrných motorů. Rychlost otáčení motoru je regulována pulzně šířkovou modulací (PWM).

Modul na základě přijatých zpráv mění šířku impulzu od 0 % do 100 % v 250 krocích. Výchozí hodnota je 0 %. Modul na základě přijatých zpráv mění také směr otáčení motoru změnou polarity výstupu. V diskuzích se modeláři neshodují na ideální frekvenci PWM signálu pro motorky v lokomotivách, proto je modul navržen tak, že na základě přijatých zpráv umí frekvenci nastavit v rozsahu od 50 Hz do 300 Hz s rozlišením 1 Hz. Výchozí hodnota je 50 Hz.

Pro napájení motorů je k modulu nutné připojit samostatné usměrněné napájení 12 V.

Modul je přednostně určen k ovládání analogových lokomotiv, ale může být použit např. k pohonu ramen větrného mlýna, vodního kola, atd. Na výstupy se připojí stejnosměrné motory pro napájení 12 V přímo nebo (pokud mají sloužit k ovládání lokomotiv na kolejišti) se výstupy propojí se vstupy na modulu ovládání kolejových úseků – viz kapitola 2.10.

Ovládání kolejiště čistě pomocí počítače je možné, ale nutí modeláře sledovat místo dění na modelu monitor počítače a nedovoluje volný pohyb kolem kolejiště. Proto byl modul navržen tak, aby umožnil připojení dvou ovladačů pro řízení rychlosti a směru otáčení motorku.

3 Programování mikroprocesorů Atmel AVR ATmega8

Programy pro mikroprocesory Atmel ATmega8 jsou napsány v jazyce C ve vývojovém prostředí Atmel Studio. Při psaní kódu je třeba počítat s omezenou velikostí paměti 8 kB FLASH pro kód a 1 kB RAM pro data. Jinak mají programy běžnou strukturu používanou v jazyce C – viz příklad.

Aby mikroprocesor podle programu pracoval správně, je nutné v oblasti pro direktivy definovat, na jaké frekvenci pracuje.

Typickým rysem programů pro mikroprocesory je rozdělení funkce main na oblast, která se vykoná jen jednou, a oblast, která se má neustále opakovat. V oblasti, která se vykoná jen jednou, jsou deklarovány lokální proměnné a je provedeno počáteční nastavení mikroprocesoru. Druhá oblast je implementována nekonečnou smyčkou while (1). Pro procesor je možné definovat obsluhy několika druhů přerušení.

K nastavení a ovládání jednotlivých částí mikroprocesoru jsou určeny registry, kterých je velké množství – viz manuál [2]. Hodnota registru se mění prostým zápisem do registru. Stav jednotlivých částí mikroprocesoru se zjišťuje čtení hodnot z registrů.

Příklad programu, který bliká diodou připojenou na výstupní pin mikroprocesoru:

```
1 #define F_CPU 16000000 // definice frekvence procesoru
2 #include <util/delay.h> // knihovna pro zpoždění
3 #include <avr/io.h> // knihovna pro input/output
4 #include <stdint.h> // knihovna pro standardní typy
5
6
   int main (void)
7
   {
8 // kód, který se vykoná jednou
9 DDRA = 0xFF; // nastavení PORTU A jako výstup
10 PORTA = 0 \times FF;
                              // zhasnutí všech LED diod PORTU A
11
12 while(1)
13
   {
14 // kód, který se neustále opakuje
15 PORTA |= (1<<PORTA0); // zhasnutí LED diod
                              // zpoždění 50ms
16 _delay_ms(50);
17 PORTA &= ~(1<<PORTA0); // zhasnutí LED diod
18 dolou ma(50); // zhasnutí LED diod</pre>
                              // zpoždění 50ms
18 _delay_ms(50);
19 }
20 return 0;
21
   }
```

Zdrojový kód 1: Příklad programu pro mikroprocesor ATmega8

Program je nutné přeložit do strojového kódu a teprve v této podobě nahrát

do paměti mikroprocesoru. Kompilace do strojového kódu je prováděna programem AVRdude [6]. Kompilované programy jsou z PC do mikroprocesoru přenášeny programátorem USBasp [7].

4 Aplikace pro ovládání kolejiště

Aplikace **Ovládání kolejiště** je napsána v jazyce C# ve vývojovém prostředí MS Visual Studio a je určena pro operační systém Microsoft Windows.

Aby bylo možné kolejiště ovládat, je třeba začít od nastavení parametrů sériové linky. Po navázání komunikace s řídícím modulem uživatel musí postupně připojovat ovládací moduly. Pak je potřeba ke vstupům/výstupům ovládacích modulů přiřadit ovládané prvky, ze kterých se sestavuje plán kolejiště. Na závěr uživatel definuje události, ke kterým může v kolejišti dojít, a automatické změny, které mají při výskytu události proběhnout. V tomto pořadí jsou uspořádány položky menu hlavního okna aplikace. Jednotlivé položky menu jsou dále popsány v samostatných kapitolách.

Objekty, se kterými aplikace pracuje, mají některé společné vlastnosti:

- ID celé číslo, jedinečné pro určitý typu objektu
- Název stručný název objektu
- Poznámka libovolný text
- Barva barva, kterou jsou podbarveny identifikační údaje v tabulce a podklady symbolů prvků v plánu kolejiště.

Při změně hodnot těchto vlastností dojde v celé aplikaci k jejich aktualizaci v zobrazených tabulkách a rozbalovacích seznamech, které je obsahují. Změny jsou automaticky promítnuty i do plánu kolejiště.

Všechna okna aplikace jsou otevírána jako samostatná nazávislá (nemodální) okna, aby mohla být bez problémů libovolně rozmístěna na ploše monitoru (popř. více monitorů). K zavření oken je kromě standardní klávesové zkratky ALT+F4 doplněna i klávesová zkratka CTRL+F4.

V levé části oken, která umožňují editovat seznamy objektů, je formulář pro změnu vlastností objektu a v pravé části tabulka objektů.

Součástí formuláře jsou standardně tlačítka:

- Nový vytvoření nového záznamu
- Kopie duplikace záznamu s přidělením nového ID
- Uložit změny uložení změněných hodnot
- Smazat smazání záznamu, před smazáním jsou prováděny kontroly, jestli není v některém z objektů použita vazba na mazaný objekt
- Nápověda zobrazení příslušné kapitoly nápovědy.

Speciální tlačítka u některých formulářů budou popsána samostatně v dalších kapitolách.

Poklepáním na záhlaví sloupce tabulky dojde k setřídění tabulky dle hodnot tohoto sloupce vzestupně nebo sestupně. Tabulka vždy může být setříděna jen podle hodnot jednoho sloupce.

Tabulku je možné filtrovat pomocí řádku nad tabulkou. Filtrování je prováděno pouze ve sloupcích s řetězci. Pokud je v poli filtru nad sloupcem tabulky zadán řetězec, pak se v tabulce zobrazí jen záznamy, které v příslušném sloupci obsahují hledaný řetězec.

Všechny požadavky na komunikaci s kolejištěm jsou v aplikaci zapisovány do fronty zpráv pro řídící jednotku. Z fronty jsou postupně přenášeny³ do řídící jednotky.

4.1 Hlavní okno aplikace

赋 <mark>Ovlád</mark> a	ání kolejiště				_		×
Soubor	<u>N</u> astavení	<u>O</u> vládací moduly	<u>M</u> odelované prvky	Kolejiště		Nápo	ověda
O ▼ Vlast	nosti sériovéh	io portu			15.03.2	018 12:00):00 .::

Obrázek 8: Hlavní okno aplikace

Hlavní okno aplikace obsahuje menu a stavový řádek. Položky menu budou popsány v samostatných kapitolách.

Stavový řádek v levé části obsahuje informace o stavu a základních parametrech sériového portu, pomocí kterého aplikace komunikuje s řídícím modulem. Klepnutím na ikonu v levé části stavového řádu lze otevřít/zavřít sériový port.

V pravé části stavového řádku je zobrazen modelový čas, který modeláři používají při ježdění podle grafikonu a jsou podle něj spouštěny plánované změny (viz kapitola 4.6.2). Klepnutím na zobrazený čas dojde k jeho spuštění nebo pozastavení. Klepnutím na ikonu před zobrazeným časem dojde k zastavení času nebo jeho spuštění od počáteční hodnoty – viz kapitola 4.6.3.

4.2 Menu Soubor

Položky menu *Soubor* poskytují běžné funkce pro uložení konfigurace kolejiště a opětovné načtení ze souboru ve formátu XML.

- Nová konfigurace
- Otevřít

³Požadavky zapsané ve frontě zpráv pro řídící modul jsou odesílány přes sériovou linku časovačem na pozadí aplikace. Fronta je obsluhována modifikovanou metodou FIFO, kdy lze provést zápis do fronty i na její začátek v případě, že se má požadavek vykonat co nejdříve. Odeslané zprávy jsou z fronty odmazávány.

- Uložit
- Uložit jako

Při uložení konfigurace kolejiště jsou zapsány i poslední polohy a velikosti všech oken aplikace.

4.3 Menu Nastavení

Položky tohoto menu slouží k nastavení vlastností hlavního okna, parametrů sériové linky a základních konfigurací diod, serv a motorů.

4.3.1 Zobrazit systémová hlášení

Položka menu *Zobrazit systémová hlášení* je typu přepínač. Umožňuje v hlavním okně aplikace zviditelnit:

- tabulku, která zobrazuje aktuální obsah fronty zpráv pro řídící modul; tabulku je možné filtrovat
- textové pole, které obsahuje poslední zprávu odeslanou do řídícího modulu
- textové pole, které obsahuje zprávy přicházející z řídícího modulu
- stavový řádek s informací o nutnosti vyčíst hodnoty vstupů pro tlačítka nebo enkodéry.

Tyto informace mají smysl pouze při ladění programu, ve výchozím stavu nejsou zobrazeny.

4.3.2 Vždy nahoře

Položka menu *Vždy nahoře* je typu přepínač. Zajistí, že hlavní okno aplikace zůstane nad všemi ostatními okny.

4.3.3 Nastavení sériového portu

Výběrem této položky menu dojde k zobrazení okna *Nastavení sériového portu*. Okno obsahuje formulář s parametry přenosu po sériové lince: *port, rychlost přenosu, počet bitů, stop bity, handshaking* a *parita*.

Při spuštění aplikace jsou načteny dostupné sériové porty. Uživatel z nich pomocí formuláře vybere jeden, ke kterému je připojen řídící modul. Podle nastavení jumperů na řídícím modulu (viz tabulka 2 na straně 14) ještě vybere rychlost přenosu. Zbývající parametry jsou nastaveny aplikací tak, aby odpovídaly neměnným hodnotám řídícího modulu.

Tlačítko *Otevřít port* naváže spojení s řídícím modulem. Port je možné otevřít také pomocí stavového řádku hlavního okna. K otevření portu může dojít automaticky i při načtení XML souboru s konfigurací kolejiště.

Ukončení spojení umožňuje tlačítko *Zavřít port*. Port je také možné zavřít pomocí stavového řádku hlavního okna.

📟 Nastavení sériového portu 🛛 🗙						
Port	COM3	\sim				
Rychlost	76800	\sim				
Počet bitů	8	\sim				
Stop bity	One	\sim				
HandShaking	None	\sim				
Parita	None	\sim				
Otevřít port	Zavřít port					
Načten í seriových portů						
😵 Nápověda						

Obrázek 9: Nastavení sériového portu

Je-li přidán sériový port (např. pomocí USB redukce) až po spuštění programu, lze seznam sériových portů aktualizovat pomocí tlačítka *Načtení sério*vých portů.

4.3.4 Definované konfigurace diody

Pro každý výstup diody v ovládacím modulu běží čítač. Pomocí konfigurace diody lze nastavit, kolik kroků čítače bude dioda svítit nebo bude zhasnuta. Konfigurace diody se skládá ze 14-ti částí konfigurace diody. Část konfigurace diody má čtyři položky:

- Délka malé smyčky celé číslo z intervalu <0; 250>
- Nula v malé smyčce celé číslo z intervalu <0; délka malé smyčky>
- Délka velké smyčky celé číslo z intervalu <0; 250>
- Následující část celé číslo z intervalu <0; 13>.

Hodnoty *Délka malé smyčky* a *Nula v malé smyčce* určují poměr mezi zhasnutou a rozsvícenou diodou. Pokud jsou nastavovány nízké hodnoty, pak je střídání tak rychlé, že oko nerozliší jednotlivé stavy a vnímá nastavení jako změnu intenzity svícení diody. Hodnota *Délka velké smyčky* určuje počet opakování malé smyčky. Po dokončení velké smyčky dojde k přechodu na další část konfigurace diody. Která to bude, to určuje hodnota *Následující část*.

Takto je možné nastavit různou intenzitu svícení diody buďto trvale nebo ji pomocí přechodů na jinou část konfigurace měnit. Cyklické střídání dvou částí konfigurace je využito např. pro blikání světel na přejezdu. Konfigurace diody do jisté míry dovoluje simulovat i pozvolné rozsvícení žárovky.

🛉 Definované konfigurace diody				- 🗆	×
🕂 Nová 🗙 Smazat					
Kopie 😵 Nápověda		Název	Poznámka	Počet	Ī
5		10201		stylů	
Název	 0	Zhasnuto		2	
Blikání 1:1 100%	1	Rozsvíceno 25%		2	
Poznámka	2	Rozsvíceno 50%		2	
	3	Rozsvíceno 75%		2	
Délka Nulav Délka	4	Rozsvíceno 100%		2	
Císlo malé malé veľké Další stvlu smvčky smvčce smvčky stvl	5	Blikání 1:1 100%		3	
	6	Rozsvíceno 11%		8	
	7	Pomalé přidávání jasu.		12	
	8	Kopie Rozsvíceno 11%		8	
2. 16 0 100 0 0	9	Kopie Kopie Rozsvíceno 11%		8	
3. 16 🜩 0 🜩 100 🜩 0 🜩	10	Rychlé přidávání jasu.		12	
4. 16 0 100 0 0					
5. 16 + 0 + 100 + 0 +					
6. 16 ♀ 0 ♀ 100 ♀ 0 ♀					
7. 16 \$ 0 \$ 100 \$ 0 \$					
11. 16 ♀ 0 ♀ 100 ♀ 0 ♀					
12. 16 🜩 0 🜩 100 🜩 0 🜩					
13. 16 🜩 0 🜩 100 🜩 0 🜩					
Barva 📔 Uložit změny					

Obrázek 10: Definované konfigurace diody

Při nastavování konfigurace diody je třeba si uvědomit, že dle doporučeného zapojení LED výrobcem mikroprocesoru dioda svítí, pokud je na výstupu logická nula. Tedy pro trvalé svícení diody nastavíme např. délka malé smyčky = 16, nula v malé smyčce = 16, délka velké smyčky = 200, následující část = 0 a uložíme konfiguraci pod názvem např. Rozsvíceno 100%. Pro zhasnutou diodu v konfiguraci změníme jen parametr nula v malé smyčce = 0.

Pro blikání diody potřebujeme nastavit dvě části, které se střídají. Konfigurace první by mohla být např. délka malé smyčky = 60, nula v malé smyčce = 0, délka velké smyčky = 200, následující část = 1 a druhá potom délka malé smyčky = 60, nula v malé smyčce = 60, délka velké smyčky = 200, následující část = 0.

4.3.5 Definované konfigurace serva

Konfigurace serva má tři položky:

👔 Definované konfigurace serv						_		\times
+ Nová X Smazat								
Id		ID	Název	Poznámka	Čas [s]	Poloha	Zapisovat poslední polohu	
l Názov		0	Krajní poloha 0	bez časového upřesnění	0	0	Ano	
Krainí poloha 250	▶	1	Krajní poloha 250	čas přechodu 5 s	5.0	250	Ano	
Poznámka								
čas přechodu 5 s								
Čas Poloha								
5.0 🜩 250 🜩								
ZapisovatPosledniPolohu								
Barva 📔 Uložit změny								

Obrázek 11: Definované konfigurace serva

- $\check{C}as$ desetinné číslo z intervalu <0; 25,0> vyjadřuje čas v sekundách
- Poloha celé číslo z intervalu <0; 250>
- Zapisovat poslední polohu může nabývat hodnoty Ne nebo Ano.

Modelářská serva většinou natáčí rameno v rozsahu přibližně 180°. Pomocí hodnoty *Poloha* je 180° rozděleno na 250 dílů, tj. jeden díl odpovídá 0,72°. Přetočení serva z 0° na 180° trvá dle údajů výrobce 0,3 s. To je ale pro modelářské účely často velmi rychle. Pomocí hodnoty *Čas* lze rychlost přetáčení zpomalit. Hodnota určuje čas v sekundách, za který by se rameno serva přetočilo o celých 180°. Toto zpomalení dovoluje např. simulovat pomalé zvedání závor na přejezdu.

Po připojení ovládacího modulu ke zdroji napájení je servo přetočeno do výchozí polohy s hodnotou 125. To by však znamenalo, že se všechna serva vždy po zapnutí zdroje napájení přetočí do této polohy. Logičtější je, aby servo zůstalo v poloze, ve které skončilo naposledy při předchozím užívání. K tomu je určen parametr Zapisovat poslední polohu. Je-li nastaven na Ano, bude požadovaná koncová poloha serva vždy uložena do paměti EEPROM v mikroprocesoru na ovládacím modulu a bude tedy výchozí po příštím spuštění.

4.3.6 Definované konfigurace motoru

Konfigurace motoru má tři položky:

- Cas desetinné číslo z intervalu <0; 25,0> vyjadřující čas v sekundách
- Stupeň celé číslo z intervalu <0; 250> vyjadřující rychlost otáčení
- Směr motoru může nabývat hodnoty Tam nebo Zpět.

Rychlost je řízena modulací šířky pulzu (PWM). Pomocí hodnoty *Stupeň* je šířka pulzu rozdělena na 250 dílů, tj. pomyslně na 250 rychlostních stupňů. Změna rychlosti se projeví okamžitě. To ale neodpovídá skutečnému provozu, kde se výrazně projevuje setrvačnost. Pomocí hodnoty *Čas* lze změnu rychlosti

👔 Definované konfigurace motoru					_		\times
+ Nová X Smazat							
Kopie 😵 Nápověda	ID	Název	Poznámka	Čas [s]	Stupeň	Směr	
Id	0	Rychlost 100		5.0	100	Zpět	
Název Rychlost 100 Poznámka							
Čas Stupeň 5.0 ↓ 100 ↓ Směr Zpět ~ Barva ↓ Uložit změny							

Obrázek 12: Definované konfigurace motoru

otáčení zpomalit. Hodnota určuje čas v sekundách, za který by došlo ke změně rychlosti ze stupně 0 na stupeň 250. Toto zpomalení dovoluje např. simulovat plynulý rozjezd a brzdění lokomotivy.

Směr otáčení motorku (např. směr jízdy lokomotivy) je dán polaritou zdroje, ke kterému je motor připojen. Polaritu zdroje řídí hodnota *Směr motoru*.

4.4 Menu Ovládací moduly

V kapitole je popsána část aplikace, která umožňuje zobrazení a otestování vstupů a výstupů ovládacích modulů v I²C síti. Připojené ovládací moduly obsahují pevně daný počet vstupů a výstupů. Tlačítka pro přidání, kopírování a mazání objektů nemají smysl, s výjimkou mazání již nepoužívaných modulů v tabulce ovládacích modulů.

Tlačítka pro testování vstupů/výstupů jsou dostupná jen v případě, že je otevřena komunikace přes sériový port.

Tabulka pro vstupy/výstupy vždy zobrazuje objekty vybraného typu ze všech ovládacích modulů.

4.4.1 Ovládací moduly

Ovládací modul je v I²C síti jednoznačně určen jedinečnou adresou (celé číslo z intervalu $\langle 1; 127 \rangle$) – viz kapitola 2.5. Každý ovládací modul má ve výchozím nastavení adresu 127, ovládací moduly je tedy nutné připojovat do I²C sítě po jednom a novému modulu vždy změnit adresu na číslo z intervalu $\langle 1; 126 \rangle$. Modul si novou adresu zapíše do EEPROM paměti. Postup můžeme shrnout do kroků:

- připojit nový modul při vypnutém napájení
- zapnout napájení

- otevřít okno Ovládací moduly
- stisknout tlačítko Test modulů řídící modul odešle na všechny adresy z intervalu <1; 127> požadavek, aby modul nahlásil svůj typ a počet ovládaných prvků
- postavit se v tabulce modulů na řádek s novým modulem s adresou 127
- zaškrtnout položku formuláře Změnit adresu
- přepsat adresu 127 na jinou
- stisknout tlačítko *Uložit změny*
- stisknout tlačítko $Test\ modulů-v$ tabulce nyní bude modul s novou adresou ve stavu Aktivní i s adresou 127 ve stavu Neaktivní
- postavit se v tabulce modulů na řádek s modulem s adresou 127 ve stavu Neaktivní



• stisknout tlačítko *Smazat*.

Obrázek 13: Ovládací moduly

Uživatel má možnost jednotlivé moduly pojmenovat, připsat poznámku a přiřadit barvu. Protože je frekvence PWM modulace pro motory společná pro všechny výstupy v rámci jednoho modulu, je tato hodnota nastavována jako atribut ovládacího modulu. Atribut je dostupný pouze pro moduly s výstupy pro motor.

Dalším důležitým atributem je *Stav*, který má po spuštění aplikace hodnotu Neznámý. K jeho změně dojde až po testu ovládacího modulu. Test vybraného modulu lze spustit tlačítkem *Test modulu*. Tlačítkem *Test modulů* je spuštěna procedura, která postupně odešle řídícímu modulu zprávy s požadavkem na test všech 127 ovládacích modulů. Dostane-li řídící modul odpověď od ovládacího modulu, přepošle ji aplikaci. Aplikace zprávu dekóduje a nastaví příslušné atributy:

- *Typ modulu* zkratka složená z písmen, která symbolizují typ vstupů nebo výstupů na ovládacím modulu
- Počet výstupů pro diody
- Počet výstupů pro serva

- Počet výstupů pro kolejové úseky
- Počet výstupů pro motory
- Frekvence PWM pro motory
- Počet vstupů pro tlačítka
- Počet vstupů pro enkodéry
- Počet vstupů pro ampérmetry motorů
- Počet vstupů pro ovladače motorů
- Stav může nabývat hodnoty Neznámý, Neaktivní nebo Aktivní.

Při dekódování zprávy také zkontroluje, zda je k ovládacímu modulu vygenerován správný počet vstupů a výstupů od každého typu. Zároveň dojde ke změně stavu modulu na hodnotu Aktivní. Pokud řídící modul odpověď od ovládacího modulu nedostane, zašle aplikaci zprávu, v níž se vyskytuje řetězec neodpovida. Aplikace nastaví stav ovládacího modulu na Neaktivní.

Tlačítko *Přehled prvků* zobrazí v dialogovém okně stručný přehled ovládaných prvků pro vybraný modul.

Tlačítko *Test řídícího modulu* odešle požadavek, aby řídící modul nahlásil svůj typ. Ten je pak zobrazen v poli pod tlačítkem *Test řídícího modulu*.

🛉 Výstupy pro diody							_		×
😵 Nápověda									
ID Adresa 3 48		ID	Název	Poznámka	Konfigurace diody	Ovládací modul	Тур	Stav	^
Název		7	Výstup pro diodu			16: Dekodér SD.002	SDT.0	Aktivní	
Výstup pro diodu		8	Výstup pro diodu			16: Dekodér SD.002	SDT.0	Aktivní	1_
Poznamka		9	Výstup pro diodu			16: Dekodér SD.002	SDT.0	Aktivní	
		10	Výstup pro diodu			16: Dekodér SD.002	SDT.0	Aktivní	
Barva 🔚 Uložit změny		0	Výstup pro diodu horní věž		2: Rozsvíceno 50%	48: Výtopna a vodárna	D.001	Aktivní	
		1	Výstup pro diodu		1: Rozsvíceno 25%	48: Výtopna a vodárna	D.001	Aktivní	1-
Version i diody		2	Výstup pro diodu		4: Rozsvíceno 100%	48: Výtopna a vodárna	D.001	Aktivní	1
5: Bikání 1:1 100%	•	3	Výstup pro diodu		5: Blikání 1:1 100%	48: Výtopna a vodárna	D.001	Aktivní	1
3. Direatt 1.1 100%		4	Výstup pro diodu			48: Výtopna a vodárna	D.001	Aktivní	1
		5	Výstup pro diodu			48: Výtopna a vodárna	D.001	Aktivní	~

4.4.2 Výstupy pro diody

Obrázek 14: Výstupy pro diody

Formulář obsahuje tlačítko *Testování diody*, které zpřístupní nabídku definovaných konfigurací diody. Vybraná konfigurace je po každé změně zaslána ovládacímu modulu pro vybraný výstup diody.

4.4.3 Výstupy pro serva

Okno Výstupy pro serva obsahuje ve formuláři položky Poloha vlevo \pm a Poloha vpravo \pm , které umožňují nastavit krajní polohy serva.

Dle manuálu k servu generují ovládací moduly pro servo signál s frekvencí 50 Hz (šířkou 20 ms). Je -li v signálu pulz 5 V s šířkou $\sim 1,5$ ms, mělo by být servo ve střední poloze. Při pulzu 5 V s šířkou ~ 1 ms by se mělo servo natočit o 90° vlevo, při pulzu 5 V s šířkou ~ 2 ms by se mělo servo natočit o 90° vpravo. Při testu serva však došlo k natáčení jen o 45°. Uvedené dvě položky slouží k úpravě délky pulzu pro krajní polohy. Jsou odeslány do ovládacího modulu při stisku tlačítka *Uložit změny* a jsou zapsány do EEPROM paměti mikroprocesoru, aby mohly být nastaveny hned po oživení modulu.

💡 Výstupy pro servo													- 1	×
😵 Nápověda			1											
ID Adresa 5 64 Název		ID	Název	Poznámka	Poloha vlevo ±	Poloha vpravo ±	Čas [s]	Poloha	Zapisovat poslední polohu	Konfigurace serva	Ovládací modul	Тур	Stav	T
Servo test zobrazení		0	Servo pro levé křídlo vrat	Poznámka k servu	-60	60	0	0	Ano	0: Krajní p	16: Dekodér SD.002	SDT.0	Aktivní	
Poznámka		1	Servo		0	0	5.0	250	Ne	1: Krajní p	16: Dekodér SD.002	SDT.0	Aktivní	
		0	Servo		0	0	0	112	Ne		64: Východní zhlaví	ST.002	Aktivní	
Poloha vlevo ± Poloha vpravo ±		1	Servo		0	0	0	125	Ne		64: Východní zhlaví	ST.002	Aktivní	
00		2	Servo		0	0	0	125	Ne		64: Východní zhlaví	ST.002	Aktivní	
Barva 🔚 Uložit změny		3	Servo		34	-16	0	125	Ne		64: Východní zhlaví	ST.002	Aktivní	
		4	Servo		0	0	0	125	Ne		64: Východní zhlaví	ST.002	Aktivní	
👋 Testování serva	•	5	Servo test zobrazení		0	0	2,5	188	Ano		64: Východní zhlaví	ST.002	Aktivní	
Definované konfigurace														
bez konfigurace serva V														
Čas Poloha														
Zapisovat poslední polohu														

Obrázek 15: Výstupy pro serva

Formulář obsahuje tlačítko *Testování serva*, které zpřístupní nabídku definovaných konfigurací serva. Vybraná konfigurace je po každé změně zaslána ovládacímu modulu pro vybraný výstup serva. Zpřístupněny jsou i jednotlivé části konfigurace serva (*Čas, Poloha* a *Zapisovat poslední polohu*). Jejich změna se také ihned odešle na vybraný výstup serva.

4.4.4 Výstupy pro kolejové úseky

Výstupy pro úseky							-	- 🗆	
😵 Nápověda									T
ID Adresa 5 80 Název		ID	Název	Poznámka	Propojeno se vstupem	Ovládací modul	Тур	Stav	Ī
Přepínač pro kolej 5a		0	nevyužito		A	80: Kolejové úseky	U.001	Aktivní	
Poznámka		1	Přepínač pro kolej 5b		С	80: Kolejové úseky	U.001	Aktivní	
		2	Přepínač pro kolej 3		В	80: Kolejové úseky	U.001	Aktivní	
Barva 📔 Uložit změny		3	Přepínač pro kolej 1		A	80: Kolejové úseky	U.001	Aktivní	
		4	Přepínač pro kolej 2		С	80: Kolejové úseky	U.001	Aktivní	
🧠 Testování úseku	•	5	Přepínač pro kolej 5a		С	80: Kolejové úseky	U.001	Aktivní	
Propojit se vstupem řídícího modulu		6	Přepínač pro kolej 4		A	80: Kolejové úseky	U.001	Aktivní	
c ~		7	nezanojeno		A	80: Kolejové úseky	U 001	Aktivní	

Obrázek 16: Výstupy pro kolejové úseky

Formulář obsahuje tlačítko *Testování úseku*, které zpřístupní nabídku vstupu, se kterým má být vybraný výstup úseku propojen.

😹 Výstupy pro motor											-		×
Nápověda				1									
ID Adresa 1 32		ID	Název	Poznámka	Citlivost	Směr	Čas	Stupeň	Konfigurace motoru	Ovládací modul	Тур	Stav	٦
Název		0	PWM ovladač motoru	В	1	Zpět	5.0	100	0: Nově de	32: Ovladače motorů	MTEA	Aktivní	
Poznámka	•	1	PWM ovladač motoru	С	2	Tam	2,1	28		32: Ovladače motorů	MTEA	Aktivní	
C													
Sestování motoru													
Definované konfigurace													
bez konfigurace motoru 🗸 🗸													
Čas Citlivost													
2,1 - 2 -													
Směr Poloha													
500 (F11)													

4.4.5 Výstupy pro motory



Formulář obsahuje tlačítko *Testování motoru*, které zpřístupní nabídku definovaných konfigurací motoru. Vybraná konfigurace je po každé změně zaslána ovládacímu modulu pro vybraný výstup motoru. Zpřístupněny jsou i jednotlivé části konfigurace motoru (*Čas, Stupeň, Směr*). Jejich změna se také ihned odešle na vybraný výstup motoru.

4.4.6 Vstupy pro tlačítka

Vstupy pro tlačitka									×
😵 Nápověda									
ID Adresa		ID	Název	Poznámka	Stav	Ovládací modul	Тур	Stav	
Název	•	0	Kon. ohl.: levé křídlo vrat v		Sepnutý	16: Dekodér SD.002	SDT.002	Aktivní	
Kon. ohl.: levé křídlo vrat výtopny - otevře		1	Kon. ohl.: levé křídlo vrat v		Rozepnutý	16: Dekodér SD.002	SDT.002	Aktivní	
Poznámka		2	Kon. ohl.: pravé křídlo vrat		Rozepnutý	16: Dekodér SD.002	SDT.002	Aktivní	
		3	Kon. ohl.: levé křídlo vrat v		Sepnutý	16: Dekodér SD.002	SDT.002	Aktivní	
Barva 🔚 Uložit změny		0	Tlačítko enkodéru - ovlada		Rozepnutý	32: Ovladače motorů	MTEA.004	Aktivní	
		1	Černé tlačítko - ovladač 0		Rozepnutý	32: Ovladače motorů	MTEA.004	Aktivní	
👞 Testování 🛛 👞 Testování		2	Tlačítko enkodéru - ovlada		Rozepnutý	32: Ovladače motorů	MTEA.004	Aktivní	
vstupu vstupů		3	Černé tlačítko - ovladač 1		Rozepnutý	32: Ovladače motorů	MTEA.004	Aktivní	~

Obrázek 18: Vstupy pro tlačítka

Formulář obsahuje tlačítko *Testování vstupu*, které řídícímu modulu odešle požadavek, aby zjistil stav vybraného vstupu tlačítka. Tlačítko *Testování vstupů*

odešle řídícímu modulu požadavek, aby zjistil stav všech vstupů tlačítka na ovládacím modulu, ke kterému je vybraný vstup tlačítka připojen.

🛓 Vstupy pro enkodéry								_		×
😵 Nápověda										
ID Adresa	Г		ID	Název	Poznámka	Přírůstek	Ovládací modul	Тур	Stav	
Název		•	0	Vstup pro enkodér 0		0	Ovladače motorů	MTEA.004	Aktivní	
Vstup pro enkodér 0			1	Vstup pro enkodér		0	Ovladače motorů	MTEA.004	Aktivní	
Poznámka Barva Barva Uložít změny										
Sestování enkodéru Testování enkodérů										

4.4.7 Vstupy pro enkodéry



Formulář obsahuje tlačítko *Testování enkodéru*, které řídícímu modulu odešle požadavek, aby zjistil přírůstek vybraného vstupu enkodéru. Tlačítko *Testování enkodérů* odešle řídícímu modulu požadavek, aby zjistil přírůstky všech vstupů enkodéru na ovládacím modulu, ke kterému je vybraný vstup enkodéru připojen.

4.4.8 Vstupy pro ampérmetry

ID Adresa 0 32	 Čas měření	Proud			
ID Adresa 0 32 ID Název Poznámka 0 Vstup pro ampérmetr	Čas měření	Proud	A 111 A 11		
0 Vstup pro ampémetr		11000	Ovladaci modul	Тур	Stav
Nazev	01.01.2000 00:00:00	0	31: Ovladače motorů	MVEA.004	Neaktivn
Vstup pro ampérmetr 1 Vstup pro ampérmetr	01.01.2000 00:00:00	0	31: Ovladače motorů	MVEA.004	Neaktivn
Poznámka 0 Vstup pro ampérmetr	19.03.2018 23:05:49	158	32: Ovladače motorů	MTEA.004	Aktivní
1 Vstup pro ampérmetr	19.03.2018 23:05:49	1	32: Ovladače motorů	MTEA.004	Aktivní

Obrázek 20: Vstupy pro ampérmetry

Formulář obsahuje tlačítko *Testování ampérmetrů modulu*, které řídícímu modulu odešle požadavek, aby zjistil proudy změřené ampérmetry na modulu, ke kterému je vybraný ampérmetr připojen. Tlačítko *Testování ampérmetrů modulů* odešle řídícímu modulu postupně požadavky pro všechny moduly s ampérmetry, aby vrátily proudy změřené všemi svými ampérmetry.

4.4.9 Vstupy pro ovladače motorů

😵 Nápověda							
ID Adresa		ID	Název	Poznámka	Ovládací modul	Тур	Stav
Název	•	0	Vstup - ovladač motoru		80: Kolejové úseky	U.001	Aktivní
Vstup - ovladač motoru		1	Vstup - ovladač motoru		80: Kolejové úseky	U.001	Aktivní
Poznámka		2	Vstup - ovladač motoru		80: Kolejové úseky	U.001	Aktivní

Obrázek 21: Vstupy pro ovladače motorů

4.5 Menu Modelované prvky

V kapitole je popsána část aplikace, která umožňuje evidenci prvků existujících v kolejišti a jejich propojení se vstupy a výstupy ovládacích modulů v I²C síti. Seznamy mohou obsahovat i prvky, které nemají přiřazen existující vstup nebo výstup – tzv. "virtuální prvky". Virtuální prvky ani existovat fyzicky nemusí, mohou být využity jen pro ovládání v plánu kolejiště. V plánu kolejiště lze zobrazit bez rozdílu všechny modelované prvky.

Položky menu *Modelované prvky* z větší části přirozeně kopírují položky menu *Ovládací moduly* – např. položka *Diody* odpovídá položce *Výstup pro diody*. Modelované prvky mohou však být i složitější a mohou využívat více vstupů a výstupů – např. *Výhybky*.

4.5.1 Diody

Uživatel ve formuláři okna vybírá výstup ovládacího modulu, ke kterému je dioda připojena. V tabulce je vedle názvu diody a přiřazeného výstup pro diodu uvedena i poslední definovaná konfigurace, která byla odeslána do ovládacího modulu.

Pokud chce uživatel v plánu kolejiště ovládat diodu poklepáním na její symbol, musí k diodě přiřadit tlačítko (popř. přepínač - viz kapitola 4.5.5).

🛉 Diody							- 0	×
🕂 Nová 🗙 Smazat								
Kopie 😵 Nápověda		ID	Název	Poznámka	Tlačítko	Výstup pro diodu	Definovaná	^
	•	0	Vodáma homí věž 1		0: Světla - vodáma - homí v	48.0: Výstup pro diodu h	2: Rozsvíceno 5	
Název		1	Vodáma homí věž 2			48.1: Výstup pro diodu	1: Rozsvíceno 2	
Vodáma homí věž 1		2	Výtopna - osvětlení vpředu			48.12: Výstup pro diodu		
Poznámka		3	Výtopna - osvětlení vpředu			48.13: Výstup pro diodu		
		4	Výtopna - osvětlení vzadu v			48.11: Výstup pro diodu		
Výstup pro diodu		5	Výtopna - osvětlení vzadu v			48.10: Výstup pro diodu		
48.0: Výstup pro diodu horní věž vodár ~		6	Výtopna - vrata		2: Světla ve výtopně	48.5: Výstup pro diodu		
		7	Vodárna vnitřní prostor			48.2: Výstup pro diodu	4: Rozsvíceno 1	
u. sveta - vodama - nomi vez u		8	Vodáma vchod			48.3: Výstup pro diodu	5: Blikání 1:1 100%	
Barva 📙 Uložit změny		9	Wtonna - vchod u vrat			48.6: Výstup pro diodu		¥

Obrázek 22: Diody

4.5.2 Serva

Uživatel ve formuláři okna vybírá výstup ovládacího modulu, ke kterému je servo připojeno. V tabulce je vedle názvu serva a přiřazeného výstup pro servo uvedena i poslední definovaná konfigurace, která byla odeslána do ovládacího modulu.

🚏 Serva						- 🗆 ×
+ Nové X Smazat						
Generation Sector Sect		ID	Název	Poznámka	Výstup pro servo	Definovaná konfigurace
0	•	0	Výhybka 1		64.4: Servo	
Název		1	Levé křídlo vrat výtopny		16.0: Servo pro levé křídlo	0: Krajní poloha 0
Výhybka 1		2	Servo test zobrazen í		64.5: Servo test zobrazen í	1: Krajní poloha
Poznámka						
Výstup pro servo						
64.4: Servo 🗸						
Barva 📔 Uložit změny						

Obrázek 23: Serva

4.5.3 Kolejové úseky

Uživatel ve formuláři okna vybírá výstup ovládacího modulu, ke kterému je kolejový úsek připojen. V tabulce je uvedeno, se kterým vstupem ovladače motoru je kolejový úsek aktuálně propojen.

Kolejové úseky						_	
+ Nový X Smazat							
ID	ID	*	Název	Poznámka	Výstup pro kolejový úsek	Propojen i se vstupem	^
U Název	4		k3a		80.2: Přepínač pro kolej 3	В	
k1a	5		k3b		80.2: Přepínač pro kolej 3	В	
Poznámka	6		k4a		80.6: Přepínač pro kolej 4	A	
	7		k4b		80.6: Přepínač pro kolej 4	A	
Výstup pro kolejový úsek	8		k5a		80.5: Přepínač pro kolej 5a	С	
80.3: Přep ínač pro kolej 1 🛛 🗸 🗸	9		k5b		80.1: Přepínač pro kolej 5b	С	
Barva 📔 Uložit změny	10		k5c		80.1: Přepínač pro kolej 5b	С	~

Obrázek 24: Kolejové úseky

4.5.4 Motory

Uživatel ve formuláři okna vybírá výstup ovládacího modulu, ke kterému je motor připojen. V tabulce je vedle názvu motoru a přiřazeného výstup pro motor uvedena i poslední definovaná konfigurace, která byla odeslána do ovládacího modulu.

融 Motory						- 0	×
+ Nový X Smazat							
ID		ID	Název	Poznámka	Výstup pro motor	Definovaná konfigurace	
1		0	Motor lokomotivy (A)		32.0: PWM ovladač motoru	0: Nově definova	
Název	•	1	Motor lokomotivy (B)		32.1: PWM ovladač motoru		
Motor lokomotivy (B) Poznámka							
Výstup motoru							
32.1: PWM ovladač motoru V Barva 🔐 Uložit změny							

Obrázek 25: Motory

4.5.5 Tlačítka

Uživatel ve formuláři okna vybírá vstup ovládacího modulu, ke kterému je tlačítko připojeno. Ke vstupu může být připojeno tlačítko nebo přepínač, což je vyjádřeno hodnotou položky *Typ*. V tabulce je vedle názvu tlačítka a přiřazeného vstupu pro tlačítko uveden i aktuální stav tlačítka a (pokud má tlačítko přiřazen vstup) aktuální stav vstupu pro tlačítko.

3 Tlačítka								- 0	×
+ Nové 🗙 Smazat									
🙀 Kopie 😵 Nápověda		ID	Název	Poznámka	Тур	Stav	Vstup pro tlačítko	Stav vstupu	^
ID	•	2	Světla ve výtopně	Vypínač	Přep in	Sepnuto	64.0: Vstup	Sepnutý	
2		3	Tlačítko enkodéru - ovlada		Tlačítko	Rozepnuto	32.0: Tlačítko enkodéru - o	Rozepnutý	
Název		4	Čemé tlačítko - ovladač 0		Tlačítko	Rozepnuto	32.1: Černé tlačítko - ovlad	Rozepnutý	
Svetla ve vytopne		5	Tlačítko enkodéru - ovlada		Tlačítko	Rozepnuto	32.2: Tlačítko enkodéru - o	Rozepnutý	
Poznamka Wypinač		6	Černé tlačítko - ovladač 1		Tlačítko	Rozepnuto	32.3: Černé tlačítko - ovlad	Rozepnutý	
Tvp tlačítka		7	Červené tlačtko - ovladač 0		Tlačítko	Rozepnuto	32.4: Červené tlačtko - ovla	Rozepnutý	
Přepínač V		0	Světla - vodárna - horní věž 0		Přepín	Neznámý			
Vstup pro tlačítko		1	Světla - vodárna - horní věž 1		Přep in	Sepnuto	64.1: Vstup	Sepnutý	
64.0: Vstup ~		8	Světla - vodárna - vnitřní pr		Přep in	Neznámý			
Barva 📔 Uložit změny		9	Světla - vodáma - vchod		Přep in	Neznámý			-

Obrázek 26: Tlačítka

4.5.6 Enkodéry

Uživatel ve formuláři okna vybírá vstup ovládacího modulu, ke kterému je enkodér připojen. V tabulce je vedle názvu enkodéru a přiřazeného vstup pro enkodér uveden aktuální přírůstek nezpracovaný v automatických změnách. Po zpracování přírůstku se jeho hodnota nuluje – viz kapitola 4.6.2.

						- 0
+ Nový X Smazat						
斗 Kopie 🧳 Nápověda		ID	Název	Poznámka	Přírůstek	Vstup pro enkodér
ID	•	0	Enkodér 0		0	32.0: Vstup pro enkodér 0
0		1	Enkodér 1		17	32.1: Vstup pro enkodér
Nazev Enkodér 0						
Poznámka						
Vstup pro enkodér						
32.0: Vstup pro enkodér 0 🛛 🗸 🗸						
Barva 🔚 Uložit změny						



4.5.7 Ampérmetry

Uživatel ve formuláři okna vybírá vstup ovládacího modulu, ke kterému je ampérmetr připojen. V tabulce je vedle názvu ampérmetru a přiřazeného vstup pro ampérmetr uvedena aktuální hodnota proudu v miliampérech.

🏲 Ampérmetry							_		×
+ Nový X Smazat									
Kopie 😵 Nápověda		ID	Název	Poznámka	Čas měření	Proud	Vstup pro ampérmetr		
ID	▶.	0	Ampérmetr motoru 0		19.03.2018 23:05:49	158	32.0: Vstup pro ampér	netr	
0		1	Ampérmetr motoru 1		19.03.2018 23:05:49	1	32.1: Vstup pro ampér	netr	
Nazev Ampérmetr motoru 0 Poznámka									
Vstup pro ampérmetr 32.0: Vstup pro ampérmetr									
Barva Uložit změny									

Obrázek 28: Ampérmetry

4.5.8 Ovladače motorů

Uživatel ve formuláři okna vybírá vstup ovládacího modulu, ke kterému je ovladač motoru připojen.

🔁 Ovladače motoru					- 0	×
+ Nový X Smazat						
Kopie & Napoveda		ID	Název	Poznámka	Vstup pro ovladač motoru	
ID	•	0	Ovladač motoru		80.0: Vstup - ovladač motoru	
0		1	Ovladač motoru		80.1: Vstup - ovladač motoru	
Název		2	Ovladač motoru		80.2: Vstup - ovladač motoru	
Poznámka						
Vstup pro ovladač motoru						
Barva						

Obrázek 29: Ovladače motorů

4.5.9 Budovy, oblasti

Budovy nebo oblasti kolejiště nemají vazbu k vstupům nebo výstupům ovládacích modulů, slouží pouze pro zobrazení v plánu kolejiště.

î Budovy, oblasti				_	×
+ Nová X Smazat					
斗 Kopie 😵 Nápověda		ID	Název	Poznámka	^
ID	•	0	Výtopna 1		
0		1	Vodáma		
Název		2	Nádražní budova		
Vytopna I Poznámka		3	Obytná budova		
		4	Skladiště 1		
		5	Skladiště 2		
Barva 📊 Uložit změny		6	Zastřešená plocha		~

Obrázek 30: Budovy, oblasti

4.5.10 Výhybky

Výhybka je část kolejí, tedy stejně jako kolejový úsek by měla být přiřazena k výstupu pro kolejový úsek. Pro zobrazení v plánu kolejiště je důležité, jestli kolej z přímého směru odbočuje vpravo nebo vlevo. To je vyjádřeno hodnotou položky Typ.

Pokud chce uživatel v plánu kolejiště ovládat (přestavit) výhybku poklepáním na její symbol, musí k výhybce přiřadit jeden přepínač nebo dvě tlačítka. Jedno tlačítko pro směr rovně a druhé pro směr do odbočky. V případě přepínače zůstane položka formuláře *Tlačítko odbočení* bez tlačítka a přepínač bude vybrán v položce formuláře *Tlačítko rovně*. Stav přepínače *Rozepnuto* odpovídá přestavění do polohy *Rovně*, stav *Sepnuto* odpovídá přestavění do polohy *Do odbočky*.

💃 Výhybky									-	
+ Nová 🗙 Smazat							1			
Ropie Nápověda		ID	Název	Poznámka	Тур	Stav výhybky	Tlačítko rovně	Tlačítko odbočení	Výstup pro kolejový úsek	Propojen i se vstupem
0	•	0	v1		Pravá	Neurčitý	22: Přepínač pro v1		80.1: Přepínač pro kolej 5b	С
v1		1	v2		Levá	Neurčitý	21: Přepínač pro v2		80.0: nevyužito	A
Poznámka		2	v3		Pravá	Neurčitý	20: Přep ínač pro v3		80.0: nevyužito	A
		3	v4		Pravá	Neurčitý	19: Přepínač pro v4		80.0: nevyužito	A
Тур		4	v5		Levá	Neurčitý	18: Přepínač pro v5		80.0: nevyužito	A
Pravá ~		5	v6		Pravá	Neurčitý	17: Přepínač pro v6		80.7: nezapojeno	A
Tlačítko rovně		6	v7		Levá	Neurčitý	16: Přepínač pro v7		80.7: nezapojeno	A
22: Přepínač pro v1 🗸 🗸		7	v8		Pravá	Neurčitý	15: Přepínač pro v8		80.7: nezapojeno	A
Tlačítko odbočení		8	v9		Levá	Rovně	14: Přepínač pro v9		80.7: nezapojeno	A
bez tlačítka 🗸 🗸		9	v10		Pravá	Neurčitý	23: Přepínač pro v10		80.7: nezapojeno	A
Výstup pro kolejový úsek		10	v11		Levá	Neurčitý	24. Přepínač pro v11		80.6: Přepínač pro kolej 4	A
80.1: Přepínač pro kolej 5b V		11	Testovací výhybka		Levá	Neurčitý			80.3: Přepínač pro kolej 1	A
Barva 📙 Uložit změny							1			

Obrázek 31: Výhybky

4.6 Menu Kolejiště

V kapitole je popsána část aplikace, která umožňuje z dříve popsaných objektů sestavit plán kolejiště. Plán kolejiště slouží nejen k zobrazení stavů objektů, ale také k ovládání kolejiště. Ovládání kolejiště lze automatizovat pomocí automatických změn, které reagují na události v kolejišti, a pomocí plánovaných změn na základě modelového času.

🗯 Události kolejiště										- 0	×
+ Nová X Smazat											
ID		ID	Název	Poznámka	Tlačítko	Ze stavu	Do stavu	Enkodér	Start modelového času	Stop modelového času	^
Nizov	•	0	Přepnut í světla tlačítkem		0: Světla - vo	Rozepnuto	Sepnuto		Ne	Ne	
Přeprutí světla tlačítkem		1	Přepnut í světla tlačítkem		0: Světla - vo	Sepnuto	Rozepnuto		Ne	Ne	
Poznámka		2	Automatická změna		2: Světla ve	Rozepnuto	Sepnuto		Ne	Ne	
		6	Automatická změna		2: Světla ve	Sepnuto	Rozepnuto		Ne	Ne	
Tlačítko Enkodér		10	Automatická změna		2: Světla ve	Rozepnuto	Sepnuto		Ne	Ne	
		11	Automatická změna		2: Světla ve	Sepnuto	Rozepnuto		Ne	Ne	
0: Světla - vodáma - homí věž 0 V		12	Výhybka 1 - rovně		4: Čemé tlač	Rozepnuto	Sepnuto		Ne	Ne	
ze stavu Rozepnuto 🗸		13	Výhybka 1 - do oblouku		3: Tlačítko e	Rozepnuto	Sepnuto		Ne	Ne	
do stavu Sepnuto 🗸		14	Automatická změna			Neznámý	Neznámý	0: Enkodér 0	Ne	Ne	
		15	Automatická změna		4: Čemé tlač	Rozepnuto	Sepnuto		Ne	Ne	
Start modelového času		16	Přepnut í světla tlačítkem		1: Světla - vo	Rozepnuto	Sepnuto		Ne	Ne	
Stop modelového času		17	Přepnut í světla tlačítkem		1: Světla - vo…	Sepnuto	Rozepnuto		Ne	Ne	
Barva 🔚 Uložit změny		18	Přepnut í světla tlačítkem		8: Světla - vo	Rozepnuto	Sepnuto		Ne	Ne	
		19	Přepnut í světla tlačítkem		8: Světla - vo	Sepnuto	Rozepnuto		Ne	Ne	
🇠 Test		20	Přepnut í světla tlačítkem		9: Světla - vo	Rozepnuto	Sepnuto		Ne	Ne	~

4.6.1 Události kolejiště

Obrázek 32: Události kolejiště

Aplikace rozpoznává čtyři typy událostí:

- Start modelového času
- Stop modelového času

- Změna stavu tlačítka
 - pro tento typ události je důležité správně nastavit hodnotu starého stavu i nového stavu tlačítka
- Přírůstek enkodéru.

Mohou být definovány dvě různé události stejného typu se stejnými parametry.

4.6.2 Automatické změny

Nová automatická změna má stav Pozastavená, tzn. aplikace na ni nereaguje. Aby se automatická změna prováděla, je nutné nastavit její stav na hodnotu Aktivní a vybrat událost kolejiště, na kterou má změna reagovat, nebo zpoždění od startu modelového času, po jehož uplynutí má ke změně dojít. Změna mění konfiguraci vybraného objektu (diody, serva, ...) nebo dává pokyn ke změření proudu ampérmetrem.

Automatické změny									-		×
🕂 Nová 🗙 Smazat	-										_
🕞 Kopie 🔗 Nápověda							-				믊
	ID	Název	Poznámka	Stav	Událost kolejiště	Dioda	Servo	Kol. úsek	Motor	Výhybka	
	13	Výhybka 1 - do oblouku		Povole	13: Výhybka 1 - do		0: Výhyb				
Název	14	Automatická změna		Povole	14: Automatická zm				0: Motor		
Přeprutí světla tlačítkem	15	Automatická změna		Povole	15: Automatická zm				0: Motor		
Poznámka	16	Přepnut í světla tlačítkem		Povole	16: Přepnutí světla	1: Vodár					
	17	Přepnut í světla tlačítkem		Povole	17: Přepnutí světla	1: Vodár					
Star Baudané	18	Přepnut í světla tlačítkem		Povole	18: Přepnutí světla	7: Vodár					
Lidžlosti kolajičtě	19	Přepnut í světla tlačítkem		Povole	19: Přepnutí světla	7: Vodár					
0: Přeprut í světla tlačítkem	20	Přepnut í světla tlačítkem		Povole	20: Přepnutí světla	8: Vodár					
	21	Přepnut i světla tlačítkem		Povole	21: Přepnutí světla	8: Vodár					
Dioda Servo Kolejový úsek Moto	22	Automatická změna		Povole	22: Automatická zm		2: Servo				
9: Výtopna - vchod u vrat 🗸 🗸	23	Automatická změna		Povole	23: Automatická zm		2: Servo				
Konfigurace 2: Rozsvíceno 50% ~	24	Automatická změna		Povole	4: Tl.ovl.černé S->R	9: Výtop					
Zpoždění před nastavením 0,0 🜩	25	Automatická změna		Povole	3: Tl.ovl.enk R->S	9: Výtop					
	26	Kol1a - vstup A		Povole	5: Tlačítko Kol1 - v			0: k1a			
	27	Kol1a - vstup B		Povole	7: Tlačítko Kol1 - v			0: k1a			
	28	Kol1a - vstup C		Povole	8: Tlačítko Kol1 - v			0: k1a			
	29	Kol1b - vstup A		Povole	5: Tlačítko Kol1 - v			1: k1b			
	30	Kol1b - vstup B		Povole	7: Tlačítko Kol1 - v			1: k1b			
	31	Kol1b - vstup C		Povole	8: Tlačítko Kol1 - v			1: k1b			
Barva 🔚 Uložit změny	32	Kopie Přepnutí světla tlačít		Povole	0: Přepnutí světla tl	0: Vodár					
	33	Přepnutí v9 rovně		Povole	24: Přepnutí přepí					8: v9	
🇠 Test	34	Přepnutí v9 do odbočky		Povole	9: Přepnutí přepín					8: v9	~

Obrázek 33: Automatické změny

Požadavek na změnu konfigurace prvku v kolejišti nebo změření proudu je zařazen do fronty zpráv pro řídící jednotku. Při zápisu do fronty je vypočítán plánovaný čas odeslání do řídící jednotky tak, že je k aktuální hodnotě modelového času přičteno zpoždění automatické změny pro vybraný prvek.

Atributy, které mohou být měněny dle třídy prvku:

- Dioda
 - definovaná konfigurace diody
 - zpoždění před nastavením

- Servo
 - definovaná konfigurace serva
 - čas
 - citlivost
 - změna polohy o
 - zpoždění před nastavením
- Kolejový úsek
 - propojení se vstupem
 - zpoždění před nastavením
- Motor
 - definovaná konfigurace motoru
 - -čas
 - citlivost
 - změna polohy o
 - změna směru
 - zpoždění před nastavením
- Výhybka
 - stav
 - zpoždění před nastavením
- Ampérmetr
 - provést měření
 - zpoždění před nastavením.

Automatická změna může obsahovat zároveň změny konfigurace pro jeden prvek od každého typu.

Aplikace v určený čas odešle zprávu z fronty přes sériový port do řídícího modulu. Je-li změna provedena na základě zpoždění po startu modelového času (plánovaná změna), je u ní nastaven atribut *Spustěna časovačem* na Ano, aby nedocházelo k opakovanému spuštění. Při startu modelového času je u všech změn atribut *Spustěna časovačem* nastaven na Ne.

4.6.3 Modelový čas

Modelový čas⁴ je zobrazován ve stavovém řádku hlavního okna a okna s plánem kolejiště. Možnosti ovládání modelového času pomocí stavového řádku jsou popsány v kapitole 4 na straně 29.

⁴Počítání modelového času provádí časovač, který běží na pozadí aplikace. Stará se jen o připočítávání času a provádění automatických změn na základě definovaného zpoždění od počátku modelového času – viz kapitola 4.6.2.

🔲 M	_		\times
	20:00	:00	
Start	Přeruš	šení	Stop
Počátečn	n <mark>í čas</mark>		
02.03.20	18 20.00	0.00	*
Poměr ke	skuteč	nému ča	asu
1:3			\sim
Formát			
HH:mm:s	s		\sim
	💡 Nápo	ověda	

Obrázek 34: Modelový čas

Okno *Modelový čas* umožňuje nastavit počátek modelového času a poměr, kterým je běh modelového času urychlován vzhledem k běžnému času. Toho se využívá při ježdění podle grafikonu.

Automatické změny jsou prováděny pouze v případě, že modelový čas běží.

4.6.4 Plán kolejiště

Plán kolejiště umožňuje vytvořit schéma kolejiště z panelů se symboly modelovaných prvků. Nový panel je vložen do středu plánu po klepnutí levým tlačítkem myši na ikonu v nástrojové liště. Pro každý typ prvku (dioda, servo, ...) je jedna ikona. Panelu se symbolem prvku se přiřadí konkrétní ovládaný prvek kolejiště (dioda, servo, ...) výběrem objektu z rozbalovacího seznamu v levé části okna. Barva panelu odpovídá barvě, kterou uživatel objektu nastavil.

Vybraný panel je označen černými čtverečky v rozích. Pomocí klávesy CTRL a levého tlačítka myši jde označit více panelů naráz. Označené panely lze duplikovat, přesouvat myší i klávesami UP, DOWN, LEFT a RIGHT nebo měnit velikost klávesovými zkratkami SHIFT+UP, SHIFT+DOWN, SHIFT+LEFT a SHIFT+RIGHT. Na panelech pro diodu, kolejový úsek a výhybku lze symboly otáčet klávesovými zkratkami ALT+LEFT a ALT+RIGHT v rozsahu od 0° po 360°. Klávesou DELETE lze panely smazat.

V nástrojové liště je tlačítko *Jen ovládání* typu přepínač. Je-li tlačítko sepnuto, nelze s panely pohybovat myší.

Každý panel v plánu kolejiště má definováno kontextové menu s položkami:

- $P\check{r}en\acute{est} bli\check{z}$ posune panel s kontextovým menu blíž k uživateli
- Přenést do popředí posune panel s kontextovým menu nejblíž k uživateli
- Odsunout dál posune panel s kontextovým menu dál od uživatele
- Odsunout na pozadí posune panel s kontextovým menu nejdál od uživatele



Obrázek 35: Plán kolejiště

- Zarovnat nahoře vybrané panely se zarovnají horním okrajem dle panelu s kontextovým menu
- Zarovnat dole vybrané panely se zarovnají dolním okrajem dle panelu s kontextovým menu
- Zarovnat vlevo vybrané panely se zarovnají levým okrajem dle panelu s kontextovým menu
- Zarovnat vpravo vybrané panely se zarovnají pravým okrajem dle panelu s kontextovým menu.

Možnosti ovládání prvků kolejiště levým tlačítkem myši:

- poklepání na tlačítko je simulován stisk tlačítka
- poklepání na přepínač dojde ke změně stavu přepínače
- poklepání na diodu dojde ke změně stavu přepínače, který je s diodou svázán
- poklepání na výhybku
 - je-li k výhybce navázán přepínač, potom dojde ke změně jeho stavu
 - jsou-li navázána dvě tlačítka
 - $\ast\,$ je-li výhybka ve stavu Rovně, potom je simulován stisk tlačítka do odbočky
 - * jinak je simulován stisk tlačítka rovně.

Význam symbolů v plánu kolejiště:

• Panel pro diodu





0: Nádraží budova

– oblast s názvem Nádraží budova.

Závěr

Zadání bakalářské práce jsem ve všech bodech splnil. Ovládací moduly pro analogové prvky kolejiště jsem dle vlastního návrhu sestavil z běžně dostupných součástek zcela svépomocí. Navržený komunikační protokol umožňuje obousměrnou komunikaci mezi ovládacími moduly a počítačem. Funkce centrální řídící jednotky z komerčních systémů zcela převzal počítač.

Nyní mnou navržený systém umožňuje ovládat osvětlení budov, řídit signalizaci světelných návěstidel, přestavovat výhybky, zvedat závory a měnit směr a rychlost jízdy lokomotiv. Dále systém reaguje na stisk tlačítka nebo přepnutí přepínače. Lze jej v budoucnu doplnit o nové moduly podle specifických požadavků modelářské praxe. Přestože jsem použil nejnižší model z řady mikroprocesorů ATmega, nebyla jeho paměťová kapacita zcela vyčerpána a dovoluje rozšíření komunikačního protokolu.

Vytvořená aplikace umožňuje přímé ovládání i jednoduchou automatizaci kolejiště. Pro využití na setkáních modelářů, kde by každý modelář řídil svou část kolejiště, by bylo dobré v další verzi doplnit synchronizaci modelového času mezi počítači pomocí počítačové sítě.

Výhodou mnou navrženého systému oproti komerčním systémům je snadná změna chování ovládaných prvků. Veškeré parametry pro chování ovládaného prvku (např. počáteční a koncová poloha serva nebo rychlost pohybu serva) jsou zasílány z aplikace a není nutný žádný zásah do softwaru nebo hardwaru modulu.

Conclusions

I have fulfilled my bachelor thesis assignment at all points. The control modules for the analogue track elements are self-assembling from custom-made components according to their own design. The proposed communication protocol allows bidirectional communication between the control modules and the computer. The function of the central control unit from commercial systems has completely taken over the computer.

Now, the system designed by me lets you control building lighting, control signal lights, adjust switches, lift barriers, and change direction and speed of locomotives. Furthermore, the system responds to the push of a button or switch the switch. It can be supplemented in the future by new modules according to the specific requirements of modeling practice. Although I used the lowest model of the ATmega microprocessor series, its memory capacity was not fully exhausted and allowed to extend the communication protocol.

The created application enables direct control and simple automation of the track. For use on model modelers meetings where each modeler would manage his part of the track, it would be good to add a synchronization of model time between computers using a computer network.

The advantage of designing me in comparison with commercial systems is the easy change in the behavior of the controlled elements. All parameters for controlling the actuator (eg start and end position of the servo or servo speed) are sent from the application and no intervention is required in the software or hardware of the module.



A Příloha – schéma řídícího modulu

Obrázek 36: Schéma řídícího modulu pro komunikaci s počítačem.



B Příloha – schéma ovládacího modulu D

Obrázek 37: Schéma modulu pro ovládání diod.



C Příloha – schéma ovládacího modulu ST

Obrázek 38: Schéma modulu pro ovládání serv.



D Příloha – schéma ovládacího modulu SDT

Obrázek 39: Schéma modulu pro ovládání serv a diod.



E Příloha – schéma ovládacího modulu U

Obrázek 40: Schéma modulu pro ovládání kolejových úseků.



F Příloha – schéma ovládacího modulu MTEA

Obrázek 41: Schéma modulu pro ovládání stejnosměrných motorů.

G Obsah přiloženého CD

bin/

Instalátor SETUP.EXE programu spustitelný přímo z CD.

doc/

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

src/

Kompletní zdrojové texty programu OVLÁDÁNÍ KOLEJIŠTĚ pro bezproblémové vytvoření spustitelných verzí programu (v ZIP archivu).

readme.txt

Instrukce pro instalaci a spuštění programu OVLÁDÁNÍ KOLEJIŠTĚ, včetně požadavků pro jeho provoz.

Navíc CD obsahuje:

data/

Ukázková a testovací data použitá pro potřeby obhajoby práce.

literature/

Některé položky literatury odkazované z dokumentace práce.

U veškerých odjinud převzatých materiálů obsažených na CD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru readme.txt.

Literatura

- DUDKA, Michal: AVR ATmega Elektronická publikace.
 Dostupné z: http://www.tajned.cz/category/programming/mikrokontrolery/avratmega/
- [2] MICROCHIP: Atmel® AVR® ATmega Elektronická publikace. Dostupné z: https://www.microchip.com/wwwproducts/en/ATmega8
- [3] NXP Semiconductors l²C-bus specification and user manual Elektronická publikace.
 Dostupné z: http://www.nxp.com/documents/user_manual/UM10204.pdf
- [4] LENZ Digital plus Zentrale LZV100 Elektronická publikace.
 Dostupné z: http://www.lenz-elektronik.de/digitalplus-zentrale-lzv100.php
- [5] ROCO Z21 model railway control system Elektronická publikace. Dostupné z: http://www.z21.eu/en/What-is-Z21/General-information
- [6] GROUP ID: #3904 AVR Downloader/UploaDEr Summary Elektronická publikace.
 Dostupné z: http://savannah.nongnu.org/projects/avrdude
- [7] FISCHL, Thomas USBasp USB programmer for Atmel AVR controllers Elektronická publikace.
 Dostupné z: http://www.fischl.de/usbasp/
- [8] TEXAS INSTRUMENTS Max232 Texas Instruments Elektronická publikace. Dostupné z: http://www.ti.com/lit/ds/symlink/max232.pdf
- [9] TEXAS INSTRUMENTS ULN2004A Texas Instruments Elektronická publikace.
 Dostupné z: http://www.ti.com/product/ULN2004A/technicaldocuments
- [10] STMICROELECTRONICS L298N Dual Full Bridge Driver STMicroelectronics Elektronická publikace.
 Dostupné z: http://www.st.com/content/st_com/en/products/motordrivers/brushed-dc-motor-drivers/l298.html
- [11] TOWERPRO SG90 digital servo Elektronická publikace. Dostupné z: http://www.towerpro.com.tw/product/sg90-7/
- [12] ZUSKA, Vladimír. Modely železnice v terénu : stavíme modelové kolejiště.
 Vyd. 1. Praha: Nakladatelství dopravy a spojů, 1988. 129 s · il.
- [13] PINTA, Martin Analog x digitál Elektronická publikace.
 Dostupné z: http://www.lokopin.wz.cz/digital/analog_digital.htm

- [14] STÁREK, Zbyněk. Železniční modelářství pro každého.
 Vyd. 2. Praha: Computer Press, 2004. 209 s. ISBN 80-7226-740-X · il.
- [15] STÁREK, Zbyněk. Modelová železnice: od historie modelů po digitální ovládání kolejiště.
 1. vyd. Brno: CPress, 2013, 220 s. ISBN 978-80-251-2199-3.
- [16] Elektronické doplňky modelového kolejiště.
 Valašské Meziříčí: Malá železnice, [1997]. 60 s. Malá železnice; 6. ISBN 80-900126-3-9.
- [17] ZAORAL, Zdenek. Automatizace modelové železnice. Praha: Mladá fronta, 1988.
- [18] STAMBAUGH, Wayne, Jean-Pierre Charras a Dick Hollenbeck KiCad EDA Elektronická publikace. Dostupné z: http://kicad-pcb.org/