

# **Modernizace modelového staničního zabezpečovacího zařízení**

Bc. Jan Esterka

---

Diplomová práce  
2006



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2005/2006

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan ESTERKA**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Modernizace modelového staničního  
zabezpečovacího zařízení**

### Zásady pro vypracování:

Cílem diplomové práce je návrh modernizace staničního zabezpečovacího zařízení modelové železnice na systém Jednotného obslužného pracoviště dle platných sdandardů Českých drah. V praxi bude systém aplikovatelný primárně na dopravní sál Střední odborné školy Edvarda Beneše a Středního odborného učiliště Břeclav a s navrhovanými postupy modulárně přenositelný na jakékoliv modelové zpracování reálné železniční sítě.

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

**Práce s mikrokontroléry ATMEL AT89S8252, Matoušek David, BEN, Praha, 2002**


**Využití rozhraní PC, Burkhard Kainka, BEN, Praha, 1997**

Vedoucí diplomové práce: **prof. Ing. Karel Vlček, CSc.**  
Ústav aplikované informatiky


Datum zadání diplomové práce: **14. února 2006**

Termín odevzdání diplomové práce: **26. května 2006**

Ve Zlíně dne 14. února 2006

  
prof. Ing. Vladimír Vašek, CSc.  
*pověřený děkan*



  
doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Cílem této práce je navrhnout zařízení, které by bylo schopno zajistit moderní přístup k řízení železničního provozu. Zařízení bude po připojení ke stávajícímu zabezpečovacímu zařízení schopno společně s osobním počítačem řídit provoz modelového výukového kolejiště. Řídící software v osobním počítači je závislý na konkrétních podmínkách implementace. Navrhované zařízení by mělo být modulární, aby bylo možné toto zařízení použít i na jiných výukových a modelových železničních kolejištích a zároveň finančně dostupné pro širokou veřejnost, aby se podpořil rozvoj nových technologií řízení. V první části se čtenář seznámí s teorií staničního řídicího systému a používaných elektronických zařízení. Dále bude navrhován řídicí systém včetně obslužných softwarových vybavení. V poslední části jsou uvedeny výkresové dokumentace navrhovaného zařízení.

Klíčová slova: Atmel, GAL, řídicí technika, zabezpečovací zařízení, mikrokontroler

## **ABSTRACT**

The goal of this diploma thesis is design hardware, which can provide modern access to model railway traffic. Hardware will in conjunction with current equipment and personal computer take control of model training yard. Control software in personal computer is dependent of specific terms of implementation. Designed hardware should be modular to use it in others educational and model railway's yards and simultaneously financially available for general public to support expansion of new control technologies. In first part can reader meet theory of station's control system and used electronic hardware. Next, new controlling system with service software will be designed. In last part are all drawing documentation of designed hardware

Keywords: Atmel, GAL, control system, safety technique, microcontroller

Děkuji vedoucímu diplomové práce panu prof. Ing. Karlovi Vlčkovi, CSc. za odbornou pomoc při vypracování projektu. Dále pak panu Bc. Jiřímu Strakovi za poskytnutí vývojového kitu pro ověření správnosti programového kódu mikrokontroleru.

Ve Zlíně

.....

Podpis diplomanta

# OBSAH

<b>ÚVOD</b> .....	<b>8</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 PŘEHLED UVAŽOVANÝCH A POUŽITÝCH TECHNOLOGIÍ</b> .....	<b>11</b>
1.1 MIKROKONTROLERY .....	11
1.1.1 Mikrokontroler PIC .....	12
1.1.2 Vývojové prostředky MPLAB, Asix Up.....	13
1.1.3 Mikrokontroler ATMEL .....	14
1.1.4 Programátor AEC ISP .....	15
1.2 PROGRAMOVATELNÁ LOGICKÁ POLE.....	15
<b>2 SYSTÉMY STANIČNÍCH ZABEZPEČOVACÍCH ZAŘÍZENÍ</b> .....	<b>18</b>
2.1 OBECNÉ POŽADAVKY NA STANIČNÍ ZABEZPEČOVACÍ ZAŘÍZENÍ.....	18
2.2 ZABEZPEČOVACÍ ZAŘÍZENÍ TYPU AŽD-71.....	21
2.3 ZABEZPEČOVACÍ ZAŘÍZENÍ TYPU ETB.....	23
2.4 ZABEZPEČOVACÍ ZAŘÍZENÍ TYPU ESA 11 .....	23
2.5 JEDNOTNÉ OBSLUŽNÉ PRACOVÍŠTĚ (JOP).....	24
<b>3 ROZBOR ŘEŠENÍ</b> .....	<b>27</b>
3.1 ŘÍDÍCÍ JEDNOTKA JOP .....	27
3.1.1 Zdroj hodinového signálu .....	30
3.1.2 Reakční doba .....	30
3.1.3 Přenosová rychlost RS-232 .....	31
3.2 OBSLUŽNÉ PRACOVÍŠTĚ JOP .....	32
<b>II PRAKTICKÁ ČÁST</b> .....	<b>33</b>
<b>4 IMPLEMENTACE</b> .....	<b>34</b>
4.1 ŘÍDÍCÍ JEDNOTKA JOP .....	34
4.1.1 Základní deska .....	34
4.1.2 Spínací modul (WRITER).....	42
4.1.3 Snímací modul (READER).....	45
4.1.4 Programový kód selektoru .....	47
4.1.5 Programový kód mikrokontroleru.....	50
4.2 KOMUNIKAČNÍ KANÁL RS-232 .....	57
4.3 OBSLUŽNÉ PRACOVÍŠTĚ JOP .....	58
<b>5 ROZŠÍŘENÍ ZÁKLADNÍHO ŘEŠENÍ</b> .....	<b>60</b>
5.1 KOMUNIKAČNÍ KANÁL ETHERNET.....	60
5.2 MODUL ŘÍZENÍ TRAKCE.....	61
<b>ZÁVĚR</b> .....	<b>62</b>
<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>63</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK</b> .....	<b>64</b>

<b>SEZNAM OBRÁZKŮ .....</b>	<b>65</b>
<b>SEZNAM TABULEK.....</b>	<b>66</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>67</b>

## ÚVOD

Střední odborná škola Edvarda Beneše a Střední odborné učiliště Břeclav (dále jen SOŠ a SOU Břeclav) má pro výuku železniční problematiky dopravní sál. Tento dopravní sál je sestaven z modelového kolejiště a zabezpečovacího zařízení AŽD, které je používáno ve skutečném provozu Českých drah, a.s. Díky tomuto výukovému prvku jsou absolventi vhodně připraveni do budoucího zaměstnání. V dnešní době již ČD, a.s. provádí modernizaci uvedeného zabezpečovacího zařízení typu AŽD na systém jednotné obsluhy JOP a to převážně na hlavních tratích. Tato skutečnost dala podnět SOŠ a SOU Břeclav, aby zahájila obdobnou modernizaci svého výukového zařízení. Jelikož by modernizace na shodné zařízení používané v praxi byla finančně příliš náročná (více než 1 milion Kč), je cílem použít jiné řešení, které by ovšem nebylo řízením rozeznatelné od implementovaného systému ČD, a.s.

Tato skutečnost již dává vstupní podmínky pro návrh výsledného zařízení. To musí být levné, moderní, modulární a spolehlivé. Všechny uvedené podmínky se tato práce snaží zohlednit a přinést SOŠ a SOU Břeclav řešení, pomocí kterého by svých cílů dosáhla.

Navrhované řešení díky své modularitě nemusí být přímo vázané na simulační kolejiště SOŠ a SOU Břeclav, ale může být použito i v jiných vzdělávacích institucích. Díky své finanční nenáročnosti by si uvedené řízení modelového kolejiště mohl implementovat i fanoušek modelové železnice nebo klub železničních modelářů.

Navrhované zařízení nebude řešit kompletní problematiku řízení železničního provozu modelového kolejiště, ale musí poskytnout základní platformu pro individuální následný vývoj ovládání provozu. Díky modularitě a flexibilitě použitých hardwarových součástí by uvedené zařízení mělo mít možnost poskytnout vývojovou základnu dalším řešitelům nových modulů a tím vylepšení celého systému řízení.

Software je rozdělen na dva základní díly, a to software (firmware), který je obsažen v jednotlivých elektronických součástech a obslužný software na pracovní stanici osobního počítače. Obslužný software je v této práci popisován tak, aby jej bylo možné vytvořit pro jakýkoliv model kolejiště i s minimálními znalostmi programování. Vizualní zpracování aplikace záleží na podmínkách zpracovatele i na věrnosti s reálným řízením dopravního provozu.



Hardware je sestaven z běžně dostupných elektronických součástí, přičemž lze jednoduchými úpravami jeho finanční náročnost snížit, ovšem s ohledem na konkrétní situaci a řízený hardware. Pomocí dalších modulů, které si může jednotlivý implementátor vytvořit individuálně lze docílit řídicího systému simulující komerční systémy v hodnotách několika desítek až stovek tisíc korun.

Cílem celého systému je poskytnutí základního komunikačního nástroje a vývojové platformy pro nahrazení složitých, prostorově obsáhlých, finančně náročných a dnes pomalu nevyhovujících řídicích systémů, které nejsou dostupné široké veřejnosti a jejich implementace je pro odborně zaměřené výukové instituce vysoké finanční zatížení.

Tato diplomová práce může sloužit také jako vhodná ukázka využití mikroprocesorů v řídicí technice, jako návod programování mikrokontrolerů ATMEL a v neposlední řadě jako vstupní platforma pro rozvíjení elektronických a programátorských znalostí mladších generací.

## **I. TEORETICKÁ ČÁST**

## 1 PŘEHLED UVAŽOVANÝCH A POUŽITÝCH TECHNOLOGIÍ

Tato kapitola seznamuje s nástroji a technologiemi, pomocí kterých byla diplomová práce realizována, resp. které byly prostudovány a později nepoužity. Jelikož se jedná o problematiku velice rozsáhlou, bude maximálně zkrácena a doplněna o odkazy na literaturu, kde lze nalézt podrobnosti.

### 1.1 Mikrokontrolery

V dnešní době nalezneme mikrokontrolery prakticky v každém zařízení. Jedná se o integrovaný obvod, který je často součástí vestavného systému a je určen k vykonávání specifických činností. Obvod je samostatný a nezávislý na jakémkoliv dalším obvodu, což právě předurčuje jeho použití v řídicím a/nebo rozhodovacím systému zařízení.

Mikrokontroler představuje spojení mikroprocesoru s vnitřní pamětí a obvody rozhraní. Čip je opatřen poměrně nízkým počtem vývodů, které lze používat k přímému řízení připojených zařízení.

Mezi charakteristické vlastnosti mikrokontrolerů patří:

- obsahují aritmeticko-logickou jednotku (ALU), která je zodpovědná za provádění aritmetických a logických operací. Jedná se například o součet, rozdíl, logický součet, negaci, bitové posuny atd.
- obsahují paměť RAM a ROM, do kterých se ukládá zpracováváný program a dočasné data, které program ke své činnosti potřebuje
- obsahují vstupně/výstupní porty pomocí kterých můžeme ovlivňovat nebo detekovat připojené obvody
- obsahují čítače/časovače, které se využívají k měření frekvence nebo přesnému časování
- jednoduchý (stále převážně osmibitový) počítač v jediném pouzdře
- vyžadují minimum vnějších součástek (nebo dokonce žádné)
- funkci jednotlivých vývodů lze programovat (vstup, výstup, vstup čítače, přerušení ...)

- zpravidla navrženy podle hardwarové architektury, která má oddělenou paměť programu a dat, což zvyšuje bezpečnost programování a výkonnost mikrokontroleru

Existuje několik druhů mikrokontrolerů, ovšem obecně se dají rozdělit do tří základních typů:

- typ 8048, který obsahuje 1KB ROM pro program a 64B RAM pro data
- typ 8051 je dnešním „standardem“
- jednočipy s RISC architekturou (PIC, MC68HCxx, AVR), což jsou mikrokontrolery s redukovanou sadou instrukcí.

Při výběru vhodného mikrokontroleru musíme vzít v úvahu několik aspektů, mezi ně patří zejména:

- dostačující počet vstupně/výstupních portů a čítačů/časovačů
- dostačující paměť programu a dat, při nedostatku musíme navrhované obvody rozšířit o externí paměti
- dostupné zařízení pro programování obsahu mikrokontroleru

Pro návrh zařízení byly brány v úvahu mikrokontrolery firmy ATMEL a firmy Microchip (PIC). Po zhodnocení charakteristik jednotlivých mikrokontrolerů byl zvolen typ AT89S8252 od firmy ATMEL. Hlavní předností tohoto obvodu je mimo dostačující počet vstupně/výstupních portů a rychlost mikrokontroleru právě jednoduchá programovatelnost. Díky jednoduchému programování mohou mikrokontroler použít i uživatelé, kteří nemají v této oblasti žádné zkušenosti anebo není u nich předpoklad dalšího využívání.

### 1.1.1 Mikrokontroler PIC

Společnost Microchip v posledních letech značně rozšířila rodinu mikrokontrolerů řady PIC. Všechny typy se vyznačují příznivou cenou. Převratnou novinkou jsou PIC se zabudovaným rozhraním USB2 umožňující rychlý přenos dat i pro malé aplikace. Kromě mikrokontrolerů PIC firma vyrábí obvody pro RF přenos, řadiče pro spínání velkých proudů a napětí apod.

PIC je obvykle RISC CPU kontrolér s většinou jednocyklovými instrukcemi. Při uvažování o vhodnosti PIC mikrokontroleru byl zvažován obvod PIC16F877 s  $8K \times 14$  bitů FLASH programovatelné paměti,  $368 \times 8$  bitů datové paměti.

Mikrokontroler PIC disponuje širokou nabídkou podpůrných obvodů (Watch Dog timer, sériový port, I<sup>2</sup>C, SPI sběrnice, čítače, časovače, komparátory, PWM generátor, AD převodník), na druhou stranu je složitější jeho programování nedostatkem levných programátorů. V našem případě by také bylo mnoho funkcí tohoto mikrokontroleru nevyužitých a proto je doporučován do jiných aplikací výsledného zařízení.

### 1.1.2 Vývojové prostředky MPLAB, Asix Up

Hlavním kladem tvorby aplikací s PIC jsou dobré vývojové prostředky. Při vývoji lze použít prostředí MPLAB, které je k dispozici ke stažení zdarma na stránkách firmy Microchip, viz [2], spolu se standardním assemblerem pro PIC MPASM.

Při psaní programu v assembleru je doporučeno používání MAKER. Program je pak překládán dvakrát. V první fázi makra generují kód a v druhé se tento kód překládá do své HEX podoby. MAKRA lze použít pro generování tabulek a dlouhých skoků. Nezanedbatelnou výhodou používání MAKER je zlepšení přehlednosti zdrojového kódu. Seznam makropříkazů podporovaných MPASM najdete v helpu aplikace MPLAB pod záložkou MPASM.

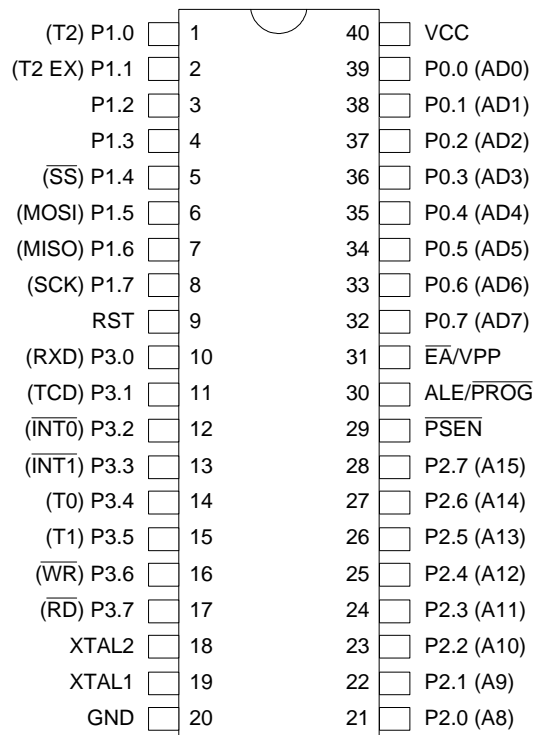


Obr. 1. Programátor Presto

Nahrávání přeloženého programu do mikrokontroleru lze realizovat za pomoci programátoru USB PRESTO od firmy ASIX s.r.o. Programátor PRESTO je ovládán z programu ASIX UP, který volně distribuuje firma ASIX. Programování mikrokontrolerů probíhá

pomocí sběrnice ICSP. Programátor PRESTO umožňuje programování většiny mikrokontrolerů PIC, ATMEL AVR a sériových pamětí. Nevyžaduje externí napájení.

### 1.1.3 Mikrokontroler ATMEL



Obr. 2. Mikrokontroler AT89S8252

Společnost ATMEL se specializuje na integrované obvody různorodého zaměření, od mikrokontrolerů, přes paměti, komunikační a multimediální řešení až po síťové datové prvky. Mezi poslední novinky společnosti ATMEL patří 32bitové RISC mikrokontrolery AVR a specializované mikrokontrolery pro automobilový průmysl.

Mikrokontroler AT89S8252 je MCU z rodiny 8051 s plnou instrukční sadou a jedno až tři cyklovými instrukcemi. Tento typ mikrokontroleru obsahuje 8KB FLASH, 2KB EEPROM, 256 bajtů RAM a 32 vstupně/výstupních linek.

Stejně jako obvody PIC obsahuje mnoho dalších funkcí, např. programovatelný Watch Dog timer, dva datové ukazatele, dva 16ti bitové čítače a časovače, plně duplexní sériový port a ISP. Jeho cena je přijatelnější než cena mikrokontroleru PIC.

Mikrokontroler AT89S8252 lze osadit vnějším oscilačním zdrojem v rozsahu 0-24MHz. Pro aplikaci byl zvolen maximální hodinový kmitočet, tedy krystal o frekvenci 24MHz. Výpočty přenosové rychlosti a doba instrukčního cyklu jsou závislé na hodinovém taktu,

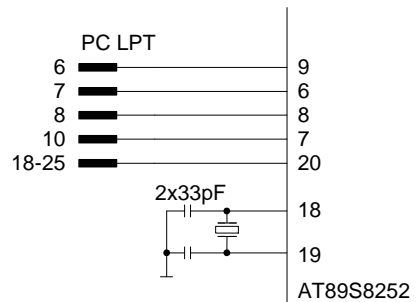
pokud by tedy byl zvolen jiný frekvenční zdroj, nesmí být zapomenuto na změnu rychlostních konstant v programu mikrokontroleru.

#### 1.1.4 Programátor AEC ISP

Pro psaní a ladění kódu bohužel nejsou dostupné kvalitní softwarové simulátory, proto se musí programový kód assembleru, popř. C++, psát v běžně dostupných textových editorech, např. Notepad.

Pro programování mikrokontroleru AT89S8252 lze použít volně šířitelný software AEC\_ISP, dostupný ke stažení na stránkách firmy AEC Electronics Ltd, viz [4]. Jedná se o účelový software pro programování mikrokontrolerů AT89S5x a AT89S8252 pomocí sběrnice ISP. Překladač assembleru lze získat na internetových stránkách výrobce mikrokontroleru, viz [3].

Programátor sestává z D-SUB 25/M konektoru připojeného k paralelnímu portu PC. Mikrokontroler musí být oscilován zdrojem v rozsahu 4 – 25 MHz. Zapojení je uvedeno na Obr. 3.



Obr. 3. Programátor AT89S8252

## 1.2 Programovatelná logická pole

K čemu jsou programovatelná logická pole vhodná? Odpověď je zřejmá již ze samotného názvu, tyto obvody nám dovolují „nadiktovat“ jejich chování. V praxi tím umožní nahrazení několika desítek až tisíců integrovaných obvodů a různých hradel do jediného čipu.

Nejběžnější dělení programovatelných obvodů v současné době je na obvody PLD (Programmable Logic Devices) a FPGA (Field Programmable Gate Arrays). Obvody PLD se dále dělí na obvody SPLD (Simple PLD) – charakteristické tím, že obsahují jedno progra-

movatelné pole, a obvody CPLD (Complex PLD) se strukturou odpovídající několika obvodům SPLD na společném čipu.

Obvody PLD jsou založeny na vyjadřování kombinačních logických funkcí ve tvaru součtu součinů (SOP – Sum Of Products), podobně jako známější paměti PROM. Na rozdíl od těchto pamětí, které mají pevně zapojenou součinnovou část a programovatelné součtové pole, je u obvodů PLD programovatelné součinnové pole a pevná součtová část (struktura PAL) nebo jsou programovatelné obě části (struktura PLA). V nejjednodušším případě odpovídá každé realizované logické funkci v obvodu PLD jedna makrobunčka, takže počet makrobuněk v těchto obvodech dává představu o tom, jak složité zapojení se do nich „vejde“. U jednodušších aplikací není obtížné sestavit potřebný obraz konfigurace programovatelného pole ručně, i když se k tomu v praxi používají návrhové systémy. Obvody SPLD obsahují 8 až 10 makrobuněk, obvody CPLD jich obsahují až několik stovek. Obvody PLD se v současnosti vyrábějí s rozsahem až do zhruba 10 000 ekvivalentních hradel.

Obvody FPGA obsahují pole malých programovatelných buněk, které se pro vytvoření logické funkce musí vhodně propojit. Úloha najít optimální propojení je zde mnohem složitější než u obvodů PLD a pro běžného uživatele to obvykle znamená spolehnout se na funkci návrhového systému, který toto propojení vytvoří automaticky. Rozsah logiky v obvodech FPGA se pohybuje o dva až tři řády výše než u obvodů PLD (miliony i více ekvivalentních hradel).

Návrhové systémy jsou v současnosti nezbytným nástrojem pro práci s programovatelnými obvody. Vstupní údaje (popis vyvíjené konstrukce) je nutné zapsat ve formě, kterou je systém schopen převést na model této konstrukce. Ten je pak možno zpracovávat simulátorem k ověření jeho správnosti, časových parametrů a podobně, dále jej syntetizérem a implementačním programem vložit (implementovat) do cílového programovatelného obvodu.

Zápis vstupních údajů bývá nejčastěji textový – pomocí jazyků HDL (Hardware Description Language) nebo grafický – editory schémat, stavových diagramů a podobně. Grafické systémy vstupu bývají obvykle nepřenositelné na jiný systém, než je ten, v němž byl popis vytvořen. Z tohoto hlediska jsou výhodnější systémy textového vstupu, které jsou ve výrazně větší míře standardizovány. K neznámějším jazykům HDL patří jazyk ABEL a jazyk VHDL. Jazyk ABEL je poměrně jednoduchý, a jeho syntaxe vychází ze struktury obvodů PLD, pro něž je určen. Jazyk VHDL má výrazně vyšší stupeň abstrakce (a také složitosti), což dovoluje i syntézu zaměřenou na obvody FPGA.



V navrhovaném zařízení je použito programovatelné logické pole SPLD, GAL22V8, k omezení počtu integrovaných obvodů na základní desce zařízení. Taktéž se tímto způsobemlepší časové parametry vyhodnocovacích logických obvodů.

## 2 SYSTÉMY STANIČNÍCH ZABEZPEČOVACÍCH ZAŘÍZENÍ

V železniční síti ČD a.s. se v dnešní době používají různé zabezpečovací zařízení. Jejich postupný modernizovaný vývoj nutí výukové a simulační centra ke stálému zdokonalování svých implementovaných zařízení. V následujících odstavcích se seznámíme s nejčastějšími zabezpečovacími zařízeními, které by bylo možné pomocí vhodného obslužného software a řešeného zařízení simulovat, resp. řídit modelovou železniční sítí.

### 2.1 Obecné požadavky na staniční zabezpečovací zařízení

Úkolem staničního zabezpečovacího zařízení (SZZ) je především zajištění bezpečné jízdy vlaků a kolejových vozidel přes výměny a znemožnění střetnutí s jinými jedoucími či stojícími vozidly. Dalším jeho úkolem je návěstění stanovené rychlosti, kterou smí vlak jet, popřípadě určení místa, kde má vlak zastavit.

Aby byla zajištěna bezpečná jízda přes výměny a zabránilo se střetnutí s jinými vozidly, je třeba před postavením návěstidla na povolující návěst splnit tyto základní požadavky:

- a) výměny, které budou při zvolené jízdě pojížděny, musí být ve správné poloze,
- b) výměny nebo jiné zařízení, které tvoří přímou boční ochranu, musí být v odvrátané poloze, aby jízda vlaku nebyla ohrožena z boku jízdou jiných vozidel,
- c) kolej, po níž vlak či posunovaný díl pojedje, a všechny pojížděné výměny musí být volné,
- d) nesmějí být uvolněna návěstidla pro jiné cesty, které danou cestu křížují nebo jinak ohrožují,
- e) jde-li o vjezd vlaku, musí být zaručeno, že odjezdové návěstidlo na konci koleje dává návěst zakazující další jízdu,
- f) musí se kontrolovat, že došly případné souhlasy sousedních obsluh k jízdě vlaku (např. při odjezdu vlaku ze stanice),
- g) musí se kontrolovat, že přejezdové zařízení na komunikacích, které danou jízdě cestu křížují, jsou v činnosti a dávají uživatelům silnic výstrahu,
- h) výměny podle bodů a, b musí být ve správné poloze uzavřeny a znemožněno jejich přestavování tak dlouho, dokud po nich nepřejede celý vlak či sunutý díl.

Dále je třeba, aby staniční zabezpečovací zařízení umožňovalo kdykoli zrušit povolující znak na návěstidle.

Zařízení musí zaručit, že výměny lze přestavovat jen za těchto podmínek:

- a) výměna je volná a nejsou na ní vozidla (s výjimkou podle bodu d),
- b) výměna není uzavřena, tzn. že není použita v některé jízdě,
- c) kdykoli během přestavování výměny musí být možno výměnu vrátit do původní polohy (s výjimkou podle bodu e),
- d) obsadí-li se během přestavování výměna, nesmí být přestavování přerušeno, ale musí se dokončit, aby nenastala vidlicová jízda,
- e) obsadí-li se během přestavování výměna, nesmí být možné přestavování přerušit a vrátit výměnu do původní polohy,
- f) poloha výměny musí odpovídat poloze výměnového řadiče na ovládacím pultu (nesouhlas vzniklý například rozřezem výměny, tj. jejím násilným projetím, musí být okamžitě signalizován).

Staniční zabezpečovací zařízení musí umožňovat obsluhu jednotlivých výhybek i jejich celých skupin současně. Dále musí umožňovat soustředění obsluhy velkého počtu výměn a návěstidel a spolupráci s traťovým zabezpečovacím zařízením.

Ovládání návěstidel, výměn i výkolejek musí být nenáročné na vynaložení síly obsluhy. Obsluze musí být indikovány tyto skutečnosti:

- a) obsazení výměn a staničních kolejí,
- b) poloha výměn a jejich případné uzávorování,
- c) návěstní znaky na návěstidlech,
- d) postavení vlakové cesty a její závěr,
- e) rozřez výměn,
- f) různé další informace, jako stav přejezdových zařízení, traťový souhlas, přiblížení vlaku, provoz na nouzové napájení atd.

Zařízení musí znemožňovat postavení vzájemně se ohrožujících jízdnicích cest. Zařízení musí vyloučit tyto vlakové cesty:

- a) současné vjezdy vlaků na tutéž kolej s výjimkou dopravní koleje rozdělené výměnovou spojkou, která musí být v odvrátané poloze,
- b) současné vjezdy vlaků stejného i opačného směru na různé koleje, jestliže tyto koleje nejsou v celé délce i v pokračování odděleny odvratnými výměnami či výkolejkami. Takovéto vjezdy i při nesplnění odvratu lze dovolit jen tehdy, je-li ohrožené místo, v němž se pokračování vlakových cest stýká, kryto hlavním návěstidlem s návěstí „Stůj“. Přitom tomuto návěstidlu musí předcházet na zábrzdnu vzdálenost předvěst s návěstí „Výstraha“.
- c) současné vjezdy vlaků opačného směru na různé koleje, je-li místo styku pokračování vlakových cest sice kryto podle předcházejícího bodu hlavním návěstidlem, ale předcházející návěstidlo je návěstidlo vjezdové, mezi jehož předvěstí a krajní vjezdovou výhybkou je na délce poloviny zábrzdné vzdálenosti nebo delší spád větší než 8 ‰.
- d) současné vjezdy vlaků opačného směru na různé koleje, je-li místo možného ohrožení sice kryto podle bodu b, ale vjezdová rychlost je vyšší než 40 km/hod. Totéž platí pro současný vjezd vlaku a odjezd druhého vlaku stejným směrem z různých kolejí,
- e) současný vjezd vlaku a odjezd druhého vlaku z jiné koleje stejným směrem, jestliže obě koleje mají společné skupinové návěstidlo. Podobně je nutné posuzovat dovolení současné jízdy vlaku a posunovaného dílu. Vlakové cesty podle bodů c,d lze dovolit, je-li mezi hlavním návěstidlem kryjícím místo ohrožení s tímto místem dodržena předepsaná tzv. prokluzová vzdálenost. Místem možného ohrožení bývá zpravidla námezník.

Zařízení musí vyloučit tyto posunové cesty:

- a) cesty souhlasného i opačného směru, jestliže se stýkají nebo křižují,
- b) cesty opačného směru na stejnou kolej, je-li kratší než 100m,
- c) cesty souhlasného i opačného směru na různé koleje, nejsou-li odděleny v celé délce i ve svém pokračování odvratnými výměnami,
- d) cesty souhlasného i opačného směru na různé koleje podle c, které se ve svém pokračování stýkají, a místo možného ohrožení není kryto nepřenosným návěstidlem s návěstí „Posun zakázán“ nebo „Stůj“.

Jízdní cesta se musí uzavřít před uvolněním příslušného návěstidla, tj. před rozsvícením návěstního znaku povolující jízdu.

Všechny výše uvedené požadavky plní reléové staniční zabezpečovací zařízení elektrickými závislostmi reléových obvodů. Jejich výstavba má své zvláštnosti, vyplývající z toho, že reléové obvody nesmějí při žádné poruše (zkrat na kabelu, přerušení vodiče, proražení usměrňovače apod.) způsobit provozně méně bezpečný stav (poruchou způsobit např. samovolné přestavení výměny pod vozidlem). Proto jsou všechna povolující kritéria vytvořená zásadně obvody na stálý proud, tzn. že relé, které svými zapínacími kontakty hlásí do zařízení např. stav volnosti koleje, je v základní poloze, je-li kolej volná, buzeno. Vjede-li na kolej vlak, relé odpadá a hlásí do zařízení, že je kolej obsazena. Totéž se stane, dojde-li k nějaké poruše v obvodu relé.

Ve všech případech, kdy reléové obvody ovládají výměny, výkolejky, návěstidla, přejímají kontroly o jejich stavu a vytvářejí mezi nimi závislosti, je třeba tyto obvody vytvářet se zřetelem na možné poruchy tak, aby při poruše nedošlo k ohrožení bezpečnosti dopravy.

## 2.2 Zabezpečovací zařízení typu AŽD-71

Reléové staniční zabezpečovací zařízení typu AŽD-71 je v blokovém provedení s výměnami přestavovanými skupinově, u něhož jsou všechny závislosti uskutečněny elektricky. Používá světelných návěstidel a pro kontrolu volnosti kolejí a výhybkových úseků kolejových obvodů.

Zařízení se ovládá z ústředního stavědla, ve kterém je na řídicím stanovišti umístěn řídicí stůl nebo ovládací stupň a kontrolní skříň. Řídicí stůl se navrhuje v malých a středních stanicích, ovládací stoly a kontrolní skříň ve velkých stanicích s rozsáhlým kolejištěm, kde by řídicí stůl musel být značně rozměrný. Všechny ovládací a indikační prvky jsou umístěny na panelu, který je na řídicím stole ve sklopené poloze pod určitým úhlem. Ve velkých stanicích jsou na samostatné nosné konstrukci umístěny pouze indikační prvky a ovládací prvky jsou umístěny na sklopeném panelu na ovládacím stole. Nouzové ovládací prvky se obvykle umísťují mimo řídicí stůl ve skříni s pomocnými tlačítky. Podle rozsahu vlakové dopravy a kolejiště může být na jednom stanovišti ovládacích stolů více s možností přepnutí obsluhy na jeden ovládací stůl.

Jízdní cesty se stavějí postupným stlačením počátečního a koncového tlačítka. Tímto tzv. dvoutlačítkovým způsobem se postaví základní cesta. Má-li být postavena variantní cesta, je nutné po stisknutí počátečního tlačítka stisknout ta variantní tlačítka, která určují zamýšlenou jízdní cestu. Jako poslední se stlačí koncové tlačítko. Jsou-li splněny podmínky pro určenou jízdní cestu, rozsvítí se na příslušných návěstidlech návěst dovolující jízdu.

V reléovém zabezpečovacím zařízení AŽD-71 se používá celkem 12 typů normalizovaných bloků. Jednotlivé typy používaných bloků jsou uvedeny v tab. 1.

Funkční určení bloků	Označení bloků
Závislostní bloky hlavního návěstidla	H
Doplňkový blok vjezdového návěstidla	W
Doplňkový blok odjezdového návěstidla	Q
Rychlostní blok	R
Blok seřadovacího návěstidla mezi výhybkami	A
Blok seřadovacího návěstidla bezvýhybkového úseku	B
Blok seřadovacího návěstidla u kusé koleje	C
Blok bezvýhybkového izolovaného úseku	M
Blok výhybkového izolovaného úseku	S
Blok dopravní koleje	K
Dvojitý výhybkový blok	D
Blok třífázového výměnového přestavníku	Vt

tab. 1. Typy a označení bloků AŽD-71

Staniční zabezpečovací zařízení AŽD-71 je ovládáno pomocí tlačítek na řídicím pultu. V novějších realizacích se implementovala tzv. číslicová volba, kde se jednotlivým tlačítkům přiřadila čísla, která se zadávala zabezpečovacímu zařízení pomocí ovládací skříňky.

### 2.3 Zabezpečovací zařízení typu ETB

Staniční zabezpečovací zařízení ETB vzniklo spojením kladů reléových stavědel (AŽD-71) s komfortem a rozsáhlými komunikačními schopnostmi počítačového ovládání. Toto zabezpečovací zařízení bylo poprvé uvedeno do provozu v roce 1991.

Zabezpečovací zařízení ETB je poloelektronické stavědlo, jehož vnitřní zařízení má část počítačovou a reléovou. Počítačová část je bezpečná a zajišťuje komunikaci s obsluhou, komunikaci s nadřazenými počítačovými systémy a některé vlastní logické funkce. SZZ ETB umožňuje ovládání z více pracovišť, která mohou být rovnocenná nebo vzájemně podřízena. Z každého pracoviště je umožněno řízení omezeného počtu dalších dopraven vybavených vlastním SZZ ETB.

Obslužné pracoviště jsou vybavena ovládáním, které odpovídá základním technickým požadavkům na jednotné obslužné pracoviště ČD (ZTP-JOP). K ovládání všech činností stavědla (tedy včetně nouzových funkcí) se používá trackball a klávesnice, nezřizují se žádná ovládací tlačítka. Zobrazování probíhá až na pěti barevných monitorech (podle rozsahu kolejiště) a na jednom doplňkovém textovém monitoru. Pro identifikaci obsluhy je každé pracoviště vybaveno čtečkou osobní identifikační karty, tak je obsluha umožněna pouze pracovníkům s příslušným oprávněním. Obslužné pracoviště je vystavěno kolem zadávacího počítače (kategorie alespoň PC 486).

Bezpečné počítačové jádro stavědla je tvořeno místní sítí LAN čtyř technologických počítačů a jednoho počítače pro údržbu, které jsou vybaveny speciálním softwarem vyvinutým firmou AŽD. Do této sítě se zapojují zadávací počítače jednotlivých obslužných pracovišť. Toto počítačové jádro řídí činnost dvou větví prováděcích počítačů, které se sdružují po dvojicích do bezpečných prováděcích počítačových uzlů, jež jsou dále vybaveny jednotkami bezpečných a spolehlivých výstupů a bezpečných vstupů. Tyto vstupy a výstupy tvoří rozhraní mezi počítačovou a reléovou částí stavědla. Počítačové uzly jsou po hardwarové i softwarové stránce produktem firmy AŽD.

### 2.4 Zabezpečovací zařízení typu ESA 11

SZZ ESA 11 je nástupcem SZZ ETB, uvedené do provozu v roce 1997, a jedná se již o elektronické stavědlo s reléovým rozhraním k venkovním prvkům zabezpečovacího zařízení. To znamená, že prakticky všechny logické funkce stavědla jsou vykonávány počíta-

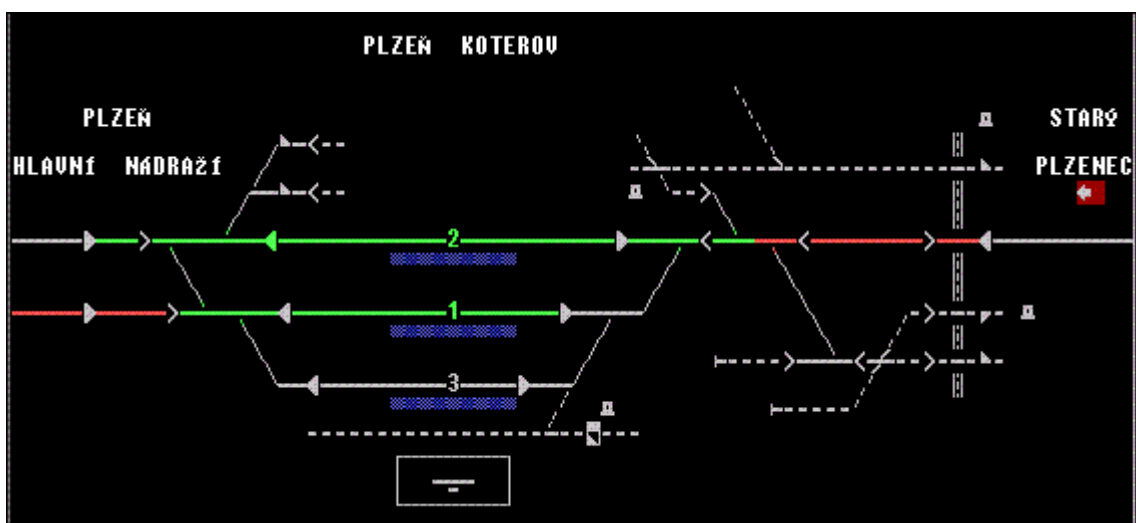
čovou částí, relé jsou použita jako spínače výkonového signálu k návěsným žárovkám, přestavníkům, kolejovým obvodům atd.

Elektronické stavědlo ESA 11 se skládá z těchto základních částí: zadávací počítačové pracoviště, ovládané podle normalizovaného systému ČD (JOP), počítačovou část kde se bezpečným způsobem provádí zadané povely a na reléovou část, která slouží jen jako výkonové převodníky k venkovním prvkům.

Počítačová část představuje vlastní počítačové jádro tvořené sítí LAN, do níž jsou zapojeny čtyři technologické počítače kategorie PC Pentium 150MHz. Programové vybavení obsahuje tzv. úroveň ETB (převzatou ze SZZ ETB) a úroveň ESA (nová část programu, jež vykonává funkce, které u SZZ ETB zajišťovala reléová logická část). Dva osobní počítače jsou vybaveny operačním systémem MS-DOS, program je psán v jazyku Borland C++. Další dva počítače jsou vybaveny operačním systémem WIN NT, program je psán v jazyku Microsoft C++. Koncepce bezpečnosti je založena na dvojnásobném zpracování dat ve dvou počítačových větvích podle dvou různých a nezávislých programů a na neustálé komparaci aktuálních dat mezi oběma aktivními technologickými počítači se zajištěným přechodem do bezpečného stavu při neshodách.

Zadávací úroveň je shodná se zadávací úrovní staničního zabezpečovacího zařízení ETB. Hardwarové vybavení zadávacího pracoviště je shodné s technologickým počítačem a pracují pod operačním systémem MS-DOS, program je psán v jazyce Borland C++.

## 2.5 Jednotné obslužné pracoviště (JOP)



Obr. 4. Příklad zobrazení kolejiště dle ZTP-JOP



Přes obecný nedostatek financí ČD se zabezpečovací systémy na počítačové platformě uplatňují v provozu ČD stále častěji. Je to zejména v důsledku výstavby tranzitních koridorů, ale tyto SZZ se vyskytují i mimo koridorové tratě. Vznikla proto potřeba ovládní těchto, technicky mnohdy velice rozdílných, systémů sjednotit. Bylo však nutné vzít v úvahu i ustanovení vztažných předpisů, ale i hlediska ergonomická aj. Proto byly ze strany provozovatele stanoveny základní technické požadavky na Jednotné obslužné pracoviště (ZTP-JOP), jako rozhraní mezi dopravním zaměstnancem a zabezpečovacím zařízením.

Jako primární médium pro zadávání obslužných úkonů je použita myš, která nezávisle na počtu monitorů umožňuje obsluhu prostřednictvím kurzoru na všech obrazovkách. Pro alfanumerické zadávání je k dispozici klávesnice, která může být využita namísto myši také k pohybu kurzoru. Redundance zadávání pro případy poruch myši je řešena obsluhou příslušných kláves klávesnice.

Pro registraci obslužných úkonů, povinně dokumentovaných událostí, provozních poruch a komentářů se použije elektronické paměťové médium. K identifikaci pracovníka a stupně jeho oprávnění k obsluze, resp. údržbě zabezpečovacího zařízení slouží čtecí zařízení personálních identifikačních karet.

Aktuální stavy zařízení jsou zobrazeny asociativní symbolikou na jednom nebo více barevných monitorech, zobrazujících kolejiště. Počet monitorů je závislý na velikosti obvodu stavědla a typografii kolejiště. Z ergonomických důvodů nemá být překročen počet pěti monitorů pro jedno pracoviště.

V určeném horním rohu se alespoň na jednom monitoru trvale zobrazuje přesný čas. Některé zobrazovací stavy jsou doplněny i akustickými indikacemi. Kromě zobrazení, nutného k ovládní zabezpečovacího zařízení, smí být na monitoru vyhrazena plocha pro komunikaci s informačním systémem, popřípadě může být některý z monitorů vyhrazen pro komunikaci s jinými systémy. Tuto část obslužného pracoviště JOP lze využít při řízení složitějších obvodů rozdělených do několika obslužných pracovišť.

Při pravidelném provozu jsou očekávány stejně krátké reakční doby jako u obslužných zařízení reléových stavědel:

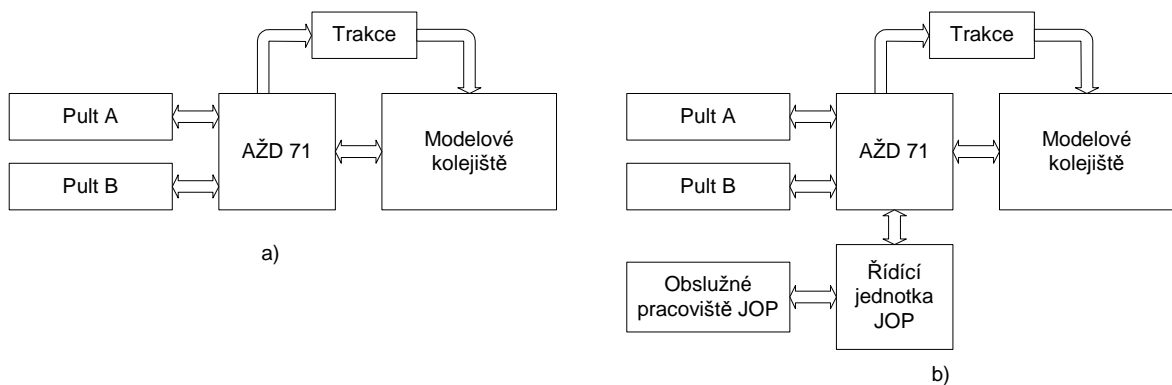
- a) potvrzení obslužných úkonů systémem do 1 sekundy,
- b) změna indikace na základě změny stavu zařízení do 2 sekund.

JOP umožňuje volbu následné jízdni cesty již před projetím kolizní cesty, přičemž následná cesta bude postavena ihned po projetí kolizní jízdni cesty, případně s časovým posunem podle požadavků obsluhy. Povelý pro stavění jízdni cesty přicházejí do zásobníku povelů, kde se řadí do fronty. První povel v pořadí (v případě, že nečeká na splnění blokující podmínky) neprodleně odchází do navazující úrovně a ze zásobníku je vymazán v okamžiku zapevnění volené cesty. Poté se okamžitě na aktuální pozici přesouvá další povel v pořadí. Obsah zásobníku musí být 10 – 15 povelů.

JOP nabízí možnost vytvoření maker povelů pro jednotlivé jízdni cesty, přičemž maximální počet maker je dán velikostí ovládaného obvodu (musí odpovídat minimálně počtu jízdni cesty uvedených v závěrové tabulce). Samozřejmostí je možnost zadání varovných štítků (na kolejové úseky, výhybky apod.), zadání výluk kolejí, lze také zadat textové komentáře (až 50 znaků), které jsou zaznamenávány v elektronickém dopravním deníku.

### 3 ROZBOR ŘEŠENÍ

SOŠ a SOU Břeclav disponuje dopravním sálem vybaveným modelovým kolejištěm řízeným zabezpečovacím zařízením typu AŽD-71 se dvěma obslužnými stanovišti. Cílem navrhovaného systému je vytvořit systém Jednotného obslužného pracoviště (JOP) pro toto modelové kolejiště a zabezpečovací zařízení, jako doplněk k současnému stavu. Po implementaci tím instituce může získat výukový systém pro standardní zabezpečovací zařízení i modernizovaný systém využívaný na tratích ČD.



Obr. 5. Blokové schéma dopravního sálu a) před implementací; b) po implementaci  
 Systém jednotného obslužného pracoviště bude doplňovat současný stav, jak je patrné z Obr. 5. Lze vidět, že navrhovaný systém sestává ze dvou základních částí:

- a) obslužné pracoviště,
- b) řídicí jednotka.

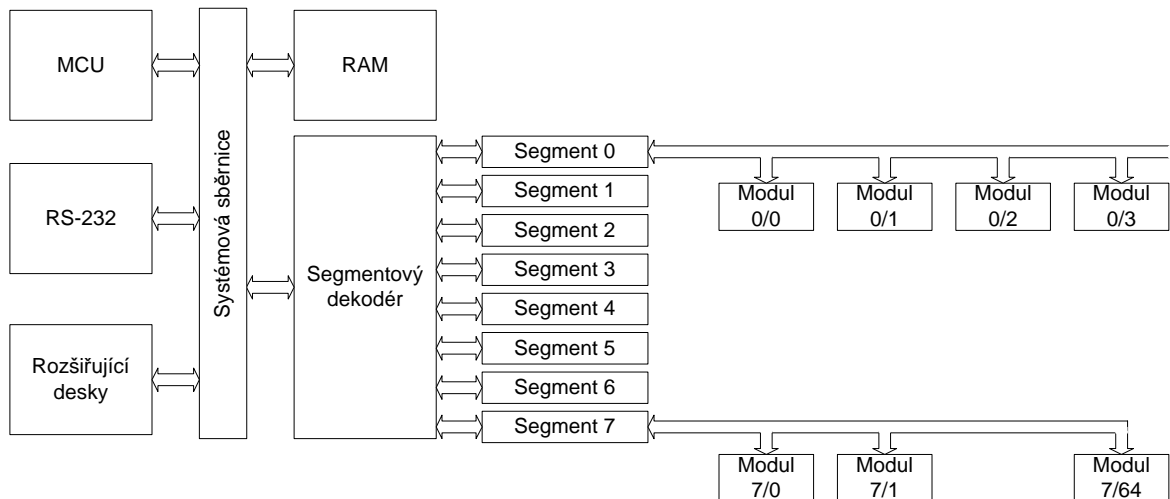
Doplňovaný systém JOP je propojen se současným zabezpečovacím zařízením, které přímo ovládá a sleduje.

#### 3.1 Řídící jednotka JOP

Řídící jednotka JOP je hardwarové řešení prováděcího systému v modulárním provedení, aby nebyla vázána na použité zabezpečovací zařízení, nebo aby umožnila ve spojení s příslušným softwarem obslužného pracoviště JOP plně nahradit modelové staniční zabezpečovací zařízení.

Modularita řídicí jednotky JOP je dána její vnitřní architekturou, uvedenou na Obr. 6. Systémová sběrnice propojuje všechny součásti, se kterými mikrokontroler (MCU) komunikuje. V základním provedení je využita statická paměť RAM, komunikační kanál RS-232 a

segmentový dekodér, který vytváří osm segmentů vnějších modulárních sběrnic. Řešení lze dále rozšiřovat pomocí rozšiřujících desek (podobně jako u PC), které mohou umožnit rozšíření segmentů, jiný komunikační kanál, diagnostické úlohy apod. Úlohou mikrokontroleru je kontrolovat stav modulů připojených k segmentům a porovnávat je s obsahem paměti RAM. Při neshodě si nový stav zapamatuje v paměti RAM a také jej oznámí přes komunikační kanály aplikaci obslužného pracoviště JOP.



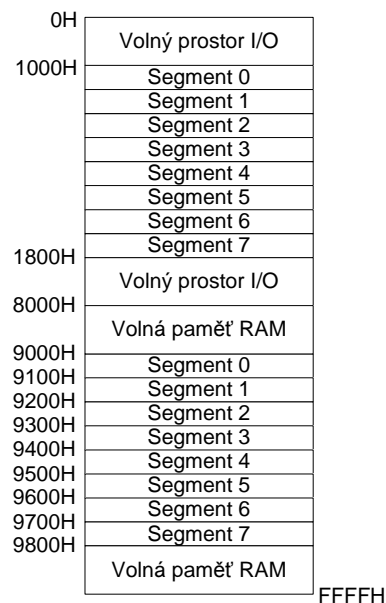
Obr. 6. Blokové schéma řídicí jednotky JOP

Z pohledu mikrokontroleru je paměť RAM a segmentový dekodér chápán jako jedna externí paměť dat. Mikrokontroler AT89S8252 umožňuje adresovat 64KB vnější paměti, proto je vhodné použít paměť 32KB RAM a stejnou velikost určit jednotlivým segmentům, tedy celkový prostor rozdělit na dvě poloviny. Výběr poloviny se může provádět kontrolou logického stavu nejvyššího bitu adresy. Použije-li se některé rozšiřující desky, bude její adresní prostor zasahovat do prostorů segmentových dekodérů. Využitím volných výstupů mikrokontroleru lze ovšem docílit, že se vstupně/výstupní rozšiřující desky adresně nekryjí s vnějším paměťovým prostorem (RAM a segmentový dekodér).

Segmentový dekodér umožňuje vytvoření osmi osmibitových sběrnic. Na sběrnici segmentu je možno připojit různé vstupně/výstupní moduly, které provádí řízení či diagnostiku zabezpečovacího zařízení nebo přímo modelového kolejiště. Díky dvěma směrům (čtení nebo zápis z vnější paměti) lze jeden segment osadit vstupním i výstupním modulem se shodnou adresou v segmentu (jeden pro čtení, druhý pro zápis). Mikrokontroler poté při zápisu do vnější paměti dat zapisuje do výstupních modulů a při čtení z vnější paměti zjiš-

tuje hodnoty ze vstupních modulů. Tímto lze využít celý adresní prostor 32KB ke vstupním i výstupním modulům, tedy max. pro 262 144 vstupů a shodný počet výstupů.

Jelikož rozšiřující desky mohou mít vlastnosti vstupně/výstupních portů a mohou používat pouze osmibitové řadiče, je vnější datová paměť mikrokontroleru rozdělena tak, aby vyšší polovinu zajišťovala externí RAM, viz Obr. 7.



Obr. 7. Rozdělení vnější datové paměti

Z uspořádání vnější paměti dat je možné usoudit, že segmentový dekodér umožňuje adresovat 2KB pro výstupní moduly. Pokud by byl počet nedostačující, lze pomocí dalšího rozšiřujícího modulu připojit více segmentových dekodérů, čímž by se mohlo docílit alokačního prostoru až 32KB pro vstupní a výstupní moduly. Při rozšiřování se ovšem musí vzít v úvahu časové prodlevy mezi jednotlivými čtecími a zapisovacími cykly. Čím větší adresný prostor (vyšší počet vstupů a výstupů) bude sledován, tím delší bude časová prodleva (reakce) detekce změny stavu. Touto problematikou se zabývá kapitola 3.1.2.

V základním provedení systém umožňuje připojit 16 384 vstupů a stejný počet výstupů (max. 32KB = 262 144 vstupů a výstupů). Po prostudování výkresové dokumentace k dopravnímu sálu SOŠ a SOU Břeclav, byl vyvozen závěr, že základní řešení je pro implementaci naddimenzované, tudíž v plném rozsahu použitelné.

### 3.1.1 Zdroj hodinového signálu

Mikrokontroler AT89S8252 umožňuje práci v taktovacím kmitočtu 0 – 24 MHz. Je logické, že čím vyšší bude taktovací kmitočet mikrokontroleru, tím rychleji zpracuje jednotlivé instrukce. Na druhou stranu si můžeme pomoci hodinového signálu vhodně vyřešit časové prodlevy externích obvodů, aniž bychom museli používat čekací smyčky v řídicím software. Každá instrukce mikrokontroleru trvá jinou délkou strojního cyklu, proto je nutné při návrhu hardware a software brát zřetel i na časové prodlevy v jednotlivých opakujících se částech.

Pro realizaci je doporučováno použití krystalu s taktovacím kmitočtem 24MHz. Veškeré výpočty provedené v této diplomové práci budou počítat právě s tímto taktovacím kmitočtem. Při použití jiného hodinového cyklu je nutno tyto výpočty přepočítat, přičemž je nutné si hned na začátku uvědomit, že délka strojního cyklu je 12× větší než délka hodinového cyklu. Pro taktovací kmitočet 24MHz je tedy délka strojního cyklu 0,5μs, viz (1).

$$t_s = \frac{12}{f_{osc}} \quad (1)$$

Délka strojního cyklu nás bude omezovat hlavně při volbě vhodných rychlostí při komunikaci pomocí sériového kanálu v závislosti na délce zpracování přerušení. Taktéž je tato délka důležitá pro určení maximální reakční doby na změnu vstupu, která je sumou délek instrukcí provádějících smyčku kontroly shodnosti RAM a vnějších vstupů.

### 3.1.2 Reakční doba

$$t_R = t_s \cdot \sum t_I \cdot \sum I \quad (2)$$

Jelikož mikrokontroler provádí detekci změny stavu vstupu cyklicky, je dána jeho reakční doba na změnu právě počtem kontrolovaných vstupů a délkou kontrolního kódu ve strojních cyklech. Takto určená reakční doba může být ještě prodloužena externími zdroji přerušení, nebo přerušení od sériové sběrnice mikrokontroleru. Reakční doba pak může být dána vztahem (2), kde  $t_s$  je délka strojního cyklu,  $t_I$  je délka instrukce v kontrolní smyčce a  $I$  je počet kontrolovaných vstupů.

Zvýšíme-li naprogramovanou reakční dobu mikrokontroleru o délku vysílací části po sériovém komunikačním kanálu (zde záleží na rychlosti přenosu a době než dojde k vyslání),

získáme skutečnou reakční dobu řídicí jednotky, která určuje maximální prodlevu mezi provedením změny a oznámením této změny obslužnému pracovišti.

### 3.1.3 Přenosová rychlost RS-232

Sériový kanál RS-232 je sériovou asynchronní linkou, která umožňuje efektivní spojení s osobním počítačem. Přenosová rychlost může být při použití mikrokontroleru AT89S8252 řízena čítačem/časovačem 2, což dovoluje její jemnější nastavení, přičemž ostatní přerušeni a čítače/časovače mohou zůstat nedotčeny, tedy pro následné využití rozšiřujícími deskami.

V režimu generátoru přenosové rychlosti pro sériový kanál pracuje čítač/časovač 2 jako 16bitový časovač s auto-reloadem. Obnovování stavu časovače proběhne automaticky, nemusí být řešeno programově. V tomto režimu čítá časovač ne obvyklou 1/12, ale 1/2 hodinového kmitočtu.

Přenosová rychlost je určena obsahem 16bitového registru RCAP2H, RCAP2L:

$$f_{CC2} = \frac{f_{osc}}{32 \cdot [65536 - (RCAP2H, RCAP2L)]} \quad (3)$$

Hodnoty registru RCAP2 pro vybrané přenosové rychlosti jsou v tab. 2. Realizovat lze pouze případy, kdy je odchylka mikrokontroleru a standardizované přenosové rychlosti v rozsahu  $\pm 5\%$ .

Přenosová rychlost	RCAP2	$f_{CC2}$	Odchylka	Realizace
2400 Bd	65 523	2 396 Bd	-4	Lze
9600 Bd	65 458	9 615 Bd	+15	Lze
19 200 Bd	65 497	19 230 Bd	+30	Lze
38 400 Bd	65 516	37 500 Bd	-900	Lze
57 600 Bd	65 523	57 692 Bd	+92	Lze
115 200 Bd	65 529	107 143 Bd	-8 057	Nelze

tab. 2. Přenosová rychlost a příslušná hodnota registrů RCAP2

### 3.2 Obslužné pracoviště JOP

Úlohou obslužného pracoviště JOP je možnost řízení zabezpečovacího zařízení dle základních technických požadavků na Jednotné obslužné pracoviště, jehož zpracovatelem jsou ČD. Obslužné pracoviště simulátoru JOP je realizováno osobním počítačem PC, který splňuje nároky obslužného software a obsahuje alespoň jeden sériový komunikační port COM. Obslužný software může být realizován v kterémkoliv volně šířitelném programovacím jazyce a může využívat i volně dostupných operačních systémů (FreeDOS, Linux, atd.). Pro finanční nenáročnost celého systému je doporučeno využít volně dostupného operačního systému FreeDOS a obslužný software programovat v některém z volně dostupných programovacích jazyce pro tento operační systém (např. FreePascal).

Pro simulátor JOP lze uvažovat přepínání mezi monitory pomocí klávesnice a na jednom monitoru zobrazovat obsah zvoleného monitoru. Lze tímto docílit simulaci i složitějších řízených obvodů při minimálních nákladech na technické vybavení pracoviště.

Software obslužného pracoviště zajišťuje předávání příkazů řídicí jednotce dle specifikace použitého komunikačního kanálu. Také musí zajistit reakci na příjem příkazů z řídicí jednotky s následným vizuálním zobrazením.



## **II. PRAKTICKÁ ČÁST**

## 4 IMPLEMENTACE

Zadání diplomové práce určovalo charakter výsledného modelu. Byla zvolena klasická technologie součástek DIP, jelikož takto osazené moduly si mohou vyrobit i amatéři. Umístění integrovaných obvodů do patic umožňuje opravovat závady vzniklé během provozu.

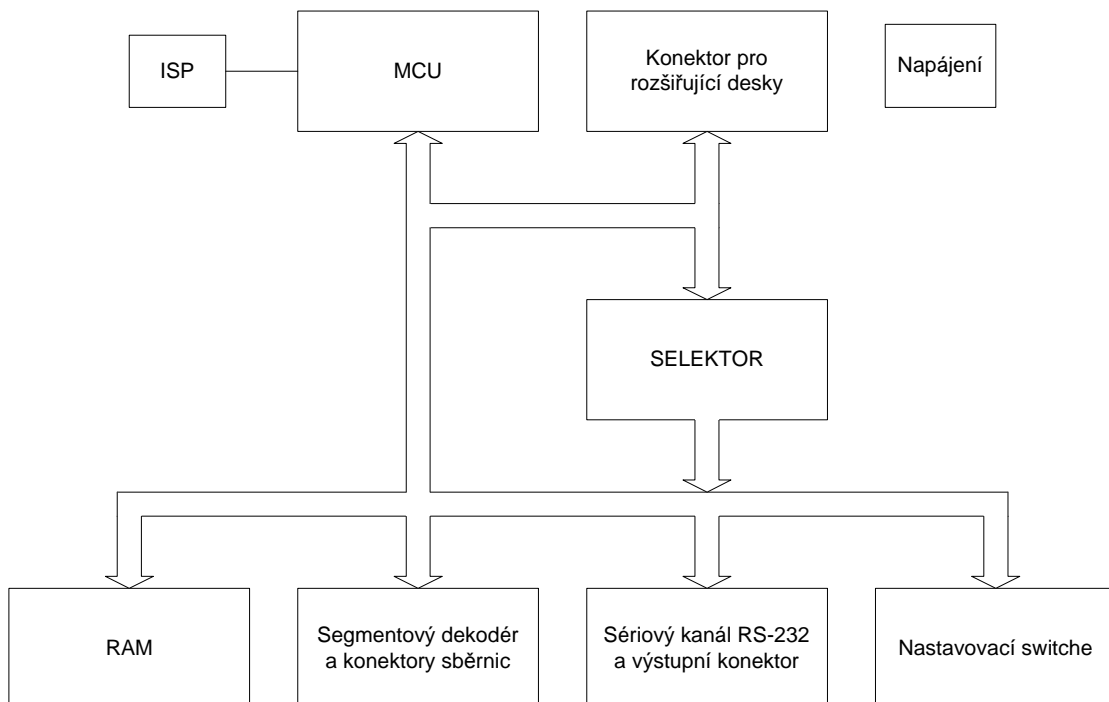
### 4.1 Řídící jednotka JOP

Řídící jednotka je rozdělena na tři části:

- Základní desku obsahující mikrokontroler, paměť, nastavovací obvody a komunikační kanál RS-232,
- Vstupní modul zajišťující čtení logického stavu vstupních linek,
- Výstupní modul zajišťující spínání malých proudů a napětí.

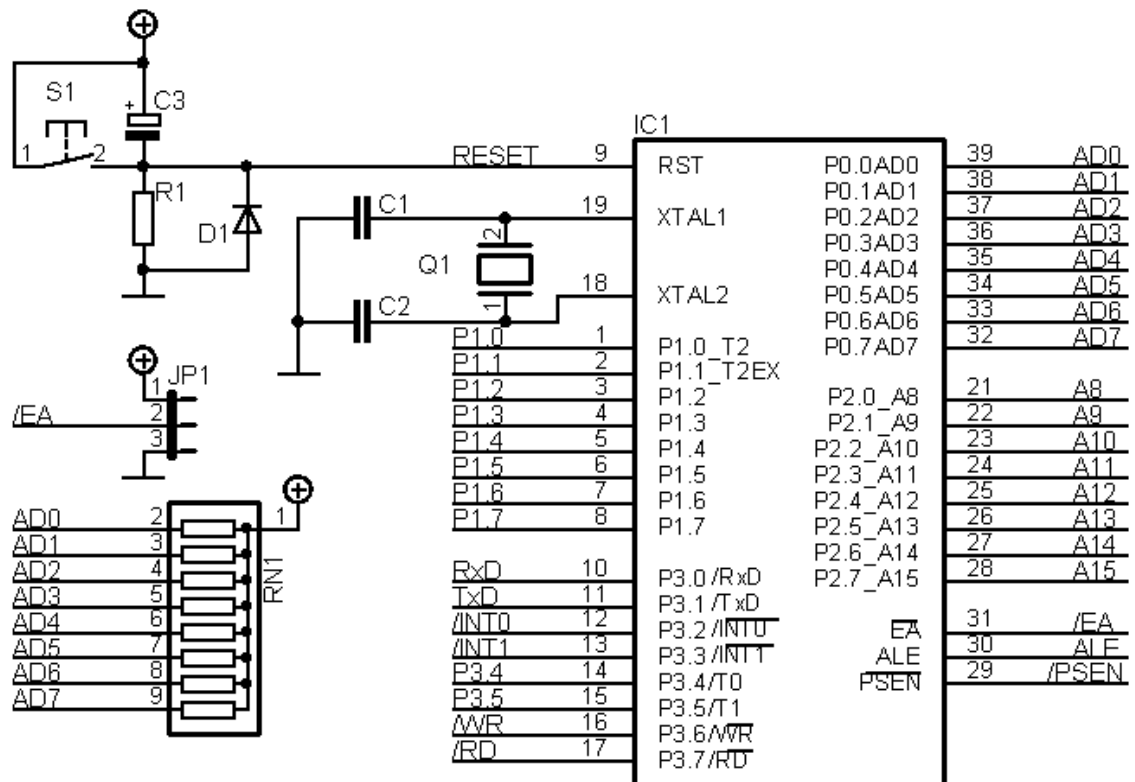
Dodatečně lze řídicí jednotku rozšiřovat o doplňující desky, které mohou zajišťovat nové komunikační kanály nebo rozšířit vlastnosti řídicího systému.

#### 4.1.1 Základní deska



Obr. 8. Blokové schéma základní desky řídicí jednotky JOP

Základní deska je jádrem celého řídicího systému. Obsahuje mikrokontroler a všechny důležité podpůrné obvody. Blokové schéma zapojení základní desky je uvedeno na Obr. 8. Systémová sběrnice propojuje všechny bloky a za pomoci selektoru vybírá obvod, se kterým aktuálně mikrokontroler komunikuje. Systém tímto umožňuje mít několik zařízení se shodnou 16bitovou adresou (např. výstupní I/O obvody a nastavovací switche). Toto rozšíření je prováděno pomocí plně nevyužitého portu mikrokontroleru.

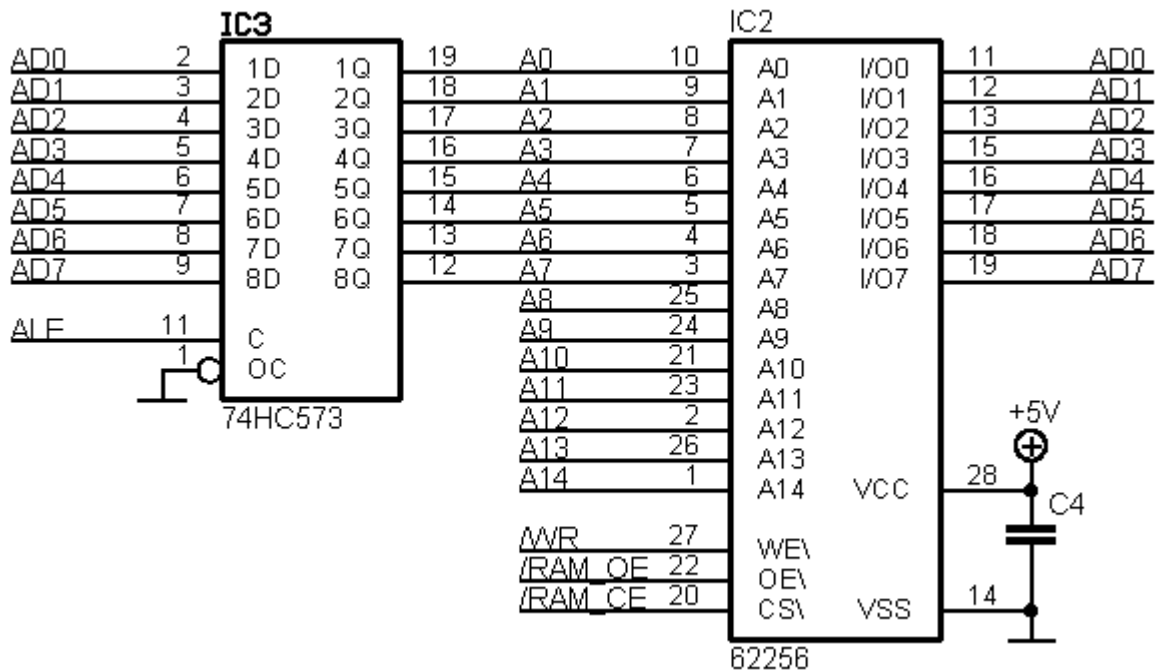


Obr. 9. Zapojení mikrokontroleru

Propojení systémové sběrnice s mikrokontrolerem (IC1) a jeho základních obvodů je na Obr. 9. Zdrojem hodinového taktu mikrokontroleru je krystal Q1 s frekvencí 24MHz a dva kondenzátory (30pF). Tento zdroj hodinového signálu je nutný pro vlastní práci mikrokontroleru, tedy i při jeho programování. Není tedy nijak odpojován. Resetování mikrokontroleru umožňuje sepnutí tlačítka S1, který přivede na resetování vstup mikrokontroleru log. 1. Reset mikrokontroleru je tvořen RC obvodem D1 a C3. Kondenzátor C3 resetuje procesor při zapnutí napájení. Dioda D1 slouží k vybití C3 při resetu zařízení krátkým vypnutím napájecího napětí.

Jumper JP1 slouží k určení paměti programu. Jsou-li propojeny piny 1-2, mikrokontroler spustí program z vnitřní paměti Flash. Jsou-li propojeny piny 2-3, provádí mikrokontroler program z vnější paměti programu. Tento jumper se může použít při vývoji software nebo diagnostice, kde lze spustit externí program z připojené vnější paměti. Odporová síť RN1 (47 k $\Omega$ ) tvoří zdvihací odpory portu P0.

Port P0 je používán pro multiplexování adresové a datové sběrnice při přístupu k vnější programové nebo datové paměti. Port P2 je používán k předání vyššího bajtu adresy. Zapojení vnější datové paměti RAM lze vidět na Obr. 10. Adresové vodiče mezi mikrokontrolerem a pamětí lze „zpreházet“. Není to na škodu, protože se čte a zapisuje stejným způsobem. Tento postup umožňuje mnohem jednodušší návrh plošného spoje. Podobně je možno zpreházet i datovou sběrnici.

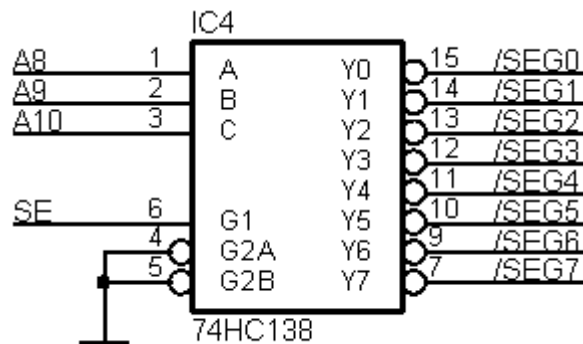


Obr. 10. Připojení vnější paměti dat

Pro demultiplexování adresy a dat z portu P0 mikrokontroleru je použit transparentní Latch 74HC573. Ten je řízen výstupem mikrokontroleru (ALE). Průběh signálů při čtení či zápisu dat z vnější paměti dat je uveden v [1].

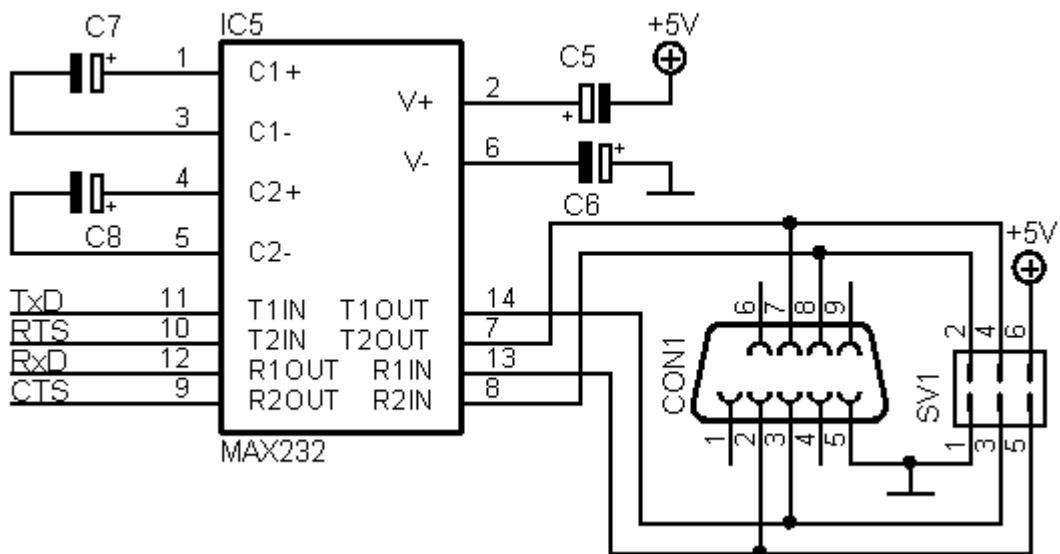
Segmentový dekodér je vytvořen dekodérem 74HC138, jehož zapojení je na Obr. 11. Dekodér snímá dolní tři bity horní poloviny adresy a aktivuje jeden ze segmentů. Jelikož má dekodér invertující výstupy, je zvolený segment aktivní při log. 0. Díky tomuto musí být před vlastní segmentovou sběrnici tyto signály ještě invertovány dvojicí integrovaných

obvodů 74LS04. Aby byla zajištěna aktivace ve správném adresním rozsahu, musí selektor generovat povolující signál SE (Segment Enable).

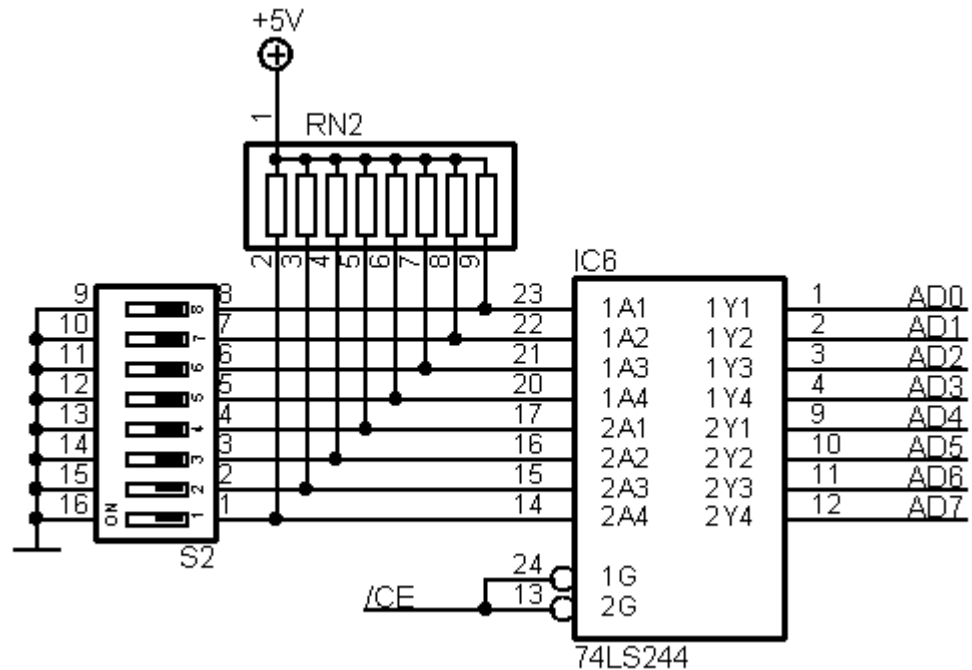


Obr. 11. Segmentový dekodér

Sériový kanál RS-232 je tvořen obligátním obvodem MAX232 v typickém zapojení, viz Obr. 12. Pokud se využijí linky RTS a CTS (např. napojením na P3.4 a P3.5), lze vytvořit jednoduchý hardwarový handshake. Způsob vysílání a přijímání znaků je pak jiná, než v této práci popisovaná. Rychlost komunikace sériového kanálu je volena přepínači v nastavovacích obvodech. Mikrokontroler ve svém software zajistí příslušnou rychlost pomocí interních časovačů. Sériový kanál je vyveden ze základní desky konektorem typu D-SUB 9F. Pokud bude uvažován sériový komunikační kanál k jiným účelům než pro spolupráci s vnějším zařízením (např. pro komunikaci s druhým systémem, resp. mikrokontrolerem), je také na základní desce interní konektor se šesti vývody.



Obr. 12. Komunikační kanál RS-232



Obr. 13. Připojení nastavovacích přepínačů

Základní deska umožňuje pozměnit nastavení systému bez nutnosti přehrávání řídicího software v mikrokontroleru. Schéma zapojení nastavovacích přepínačů je na Obr. 13. Při čtení nastavovacích přepínačů je jejich obsah přenášen na systémovou sběrnici pomocí Latche 74LS244. Ten je aktivován signálem Config Enable (/CE), který je aktivní v log. 0 a pochází opět ze selektoru. Jsou-li přepínače v rozpojeném stavu, dostává mikrokontroler signály v úrovni log. 1. Je to dáno zdvihacím odporovým polem RN2. Přepínací pole S2 lze nahradit polem osmi jumperů. Význam jednotlivých přepínačů určuje tab. 3.

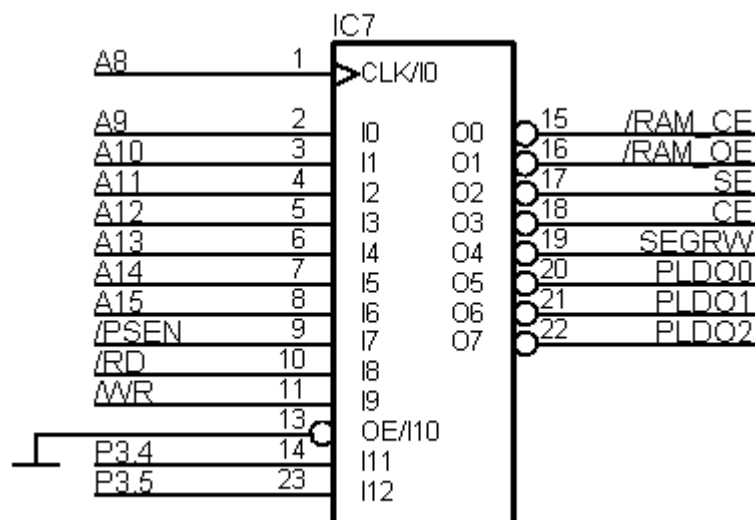
Číslo přepínače	Význam	
1	Zapnutí/vypnutí komunikace kanálem RS-232	
2,3	Výběr rychlosti komunikačního kanálu RS-232	
Pin 2	Pin 3	Rychlost
Off	Off	9 600 Bd
On	Off	19 200 Bd
Off	On	38 400 Bd
On	On	57 600 Bd

4	Neukládat stav vstupů do paměti RAM, ale přímo oznamovat do komunikačních kanálů
5-8	Rezervováno pro software využívající rozšiřující desky

tab. 3. Významy konfiguračních přepínačů

Selektorem si mikrokontroler volí, s kterou skupinou prvků bude komunikovat. Jelikož může být tento prvek využit i jednoduchými rozšiřujícími deskami, bylo zvoleno programovatelné logické pole. Zapojení tohoto programovatelného pole je vidět na Obr. 14.

Selektor disponuje rozhodováním nad celou horní polovinou adresy (port P2 mikrokontroleru) a některými řídicími vstupy. Cílem selektoru je vybrat aktivní obvody, pro které jsou platné data na portu P0 mikrokontroleru. Ze všech osmi výstupů má programovatelné pole zapojeno na základní desce pouze pět (pro RAM, segmenty a nastavení). Ostatní výstupy jsou pro přídatné desky. Toto řešení umožní efektivněji využít programovatelné pole v závislosti na dalších použitých přídatných deskách. Ty nemusí obsahovat selektovací obvody, ale mohou využít jednoho ze tří výstupů programovatelného pole umístěného na základní desce.



Obr. 14. Zapojení vývodů selektoru

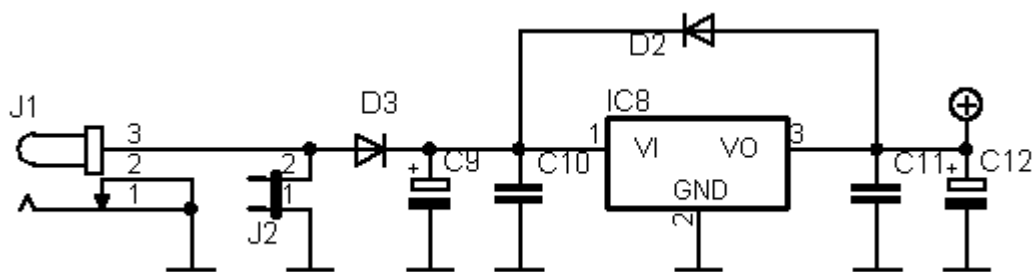
Může nastat situace, že toto programovatelné pole nebude dostačující, pak lze mezi jednotlivé přídatné moduly vložit další programovatelné logické pole, které by generovalo nové řídicí signály pro následující přídatné desky. Jelikož je systémová sběrnice jednoduchou formou několikažilové vedení, lze ji taktéž rozvětvit, ovšem za předpokladu nepřetížení

výstupních portů mikrokontroleru. Za takovým účelem je vhodné systémovou sběrnici doplnit o obousměrné budiče, které by zajistili spolehlivost na delším nebo rozvětveném vedení.

Řídící software pro programovatelné logické pole je odvozeno podle potřeb celé sestavy modulů. Pro základní sestavu bez přídatných desek pro spínací a čtecí moduly je tento software uveden v kapitole 4.1.4.

Napájecí zdroj integrovaný na základní desce vychází z klasického zapojení stabilizátoru 7805, viz Obr. 15. Napájecí napětí z „adaptéru do zdi“ je vedeno přes konektor J1 na diodu D3 (libovolná malá usměrňovací) chrání celou desku před přepólováním na filtrační elektrolytický kondenzátor C9. Konektorem J2 (pájecí piny) lze přivést napájecí napětí z výkonnějšího zdroje s tím, že se dá vyhnout někdy nevhodným konektorům typu Jack.

Obvod 7805 je v klasickém zapojení, C10 a C11 blokují celé zapojení před záškmity, dioda D2 chrání stabilizátor před poškozením při poklesu vstupního napájecího napětí pod úroveň napětí na výstupu. Kondenzátor C12 slouží jako hlavní výstupní filtrační kondenzátor. Jednotlivé integrované obvody mají mezi vstupy pro napájení a uzemnění malé keramické kondenzátory blokuující napájení proti záškmitům.



Obr. 15. Napájení základní desky

Základní deska komunikuje v jednotlivých segmentech pomocí segmentové sběrnice. Ta je vyvedena pomocí šestnáctipinového konektoru typu WSL. Základní deska obsahuje osm segmentových sběrnic. Každá obsahuje osmibitovou multiplexovanou adresní a datovou sběrnici. K demultiplexování je využíván výstup ALE mikrokontroleru. Moduly připojené k této sběrnici musí splňovat následující podmínky:

- mohou mít vlastní zdroj napájení, ovšem musí být zachována společná zem signálu,
- modul oddělí vnější obvody od řídicích obvodů modulu,



- c) modul musí mít schopnost rozeznat jeho adresu a reagovat pouze v době, kdy je to po něm požadováno pomocí řídicích signálů,
- d) modul nesmí zatěžovat sběrnici více než  $100\mu\text{A}$ .

Konkrétní významy pinů segmentové sběrnice jsou uvedeny v tab. 4.

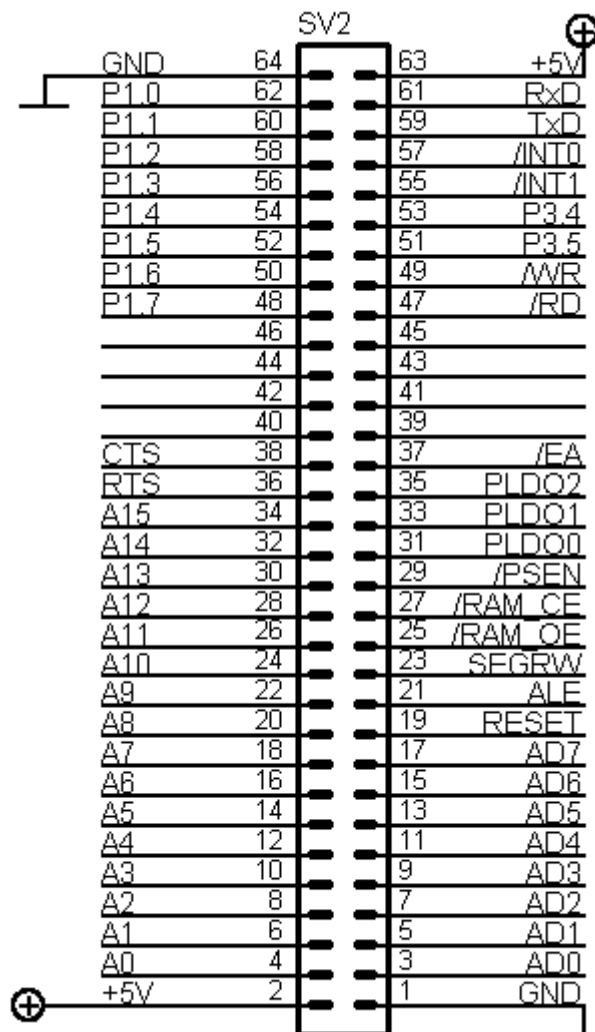
Číslo pinu	Označení	Směr dat	Význam
1	Vcc		Napájení +5V
2 – 9	AD0 – AD7	vstup/výstu p	Multiplexovaná adresa a data
10	RESET	výstup	Reset segmentu
11	ALE	výstup	Signál oddělující adresu od dat
12	SEG	výstup	Výběr aktivního segmentu
13	R/W	výstup	Čtení nebo zápis do adresovaného modulu
14 – 16	SR0 – SR2		Rezervováno pro připojení k rozšiřující desce

*tab. 4. Popis pinů segmentové sběrnice*

Systémová sběrnice je kompletně včetně dalších řídicích signálů přivedena na rozšiřující konektor, který umožňuje připojení rozšiřujících desek. Rozpis vývodů rozšiřujícího konektoru je pro názornost uveden na Obr. 16. Nevyužité piny systémové sběrnice lze využít pro komunikaci mezi jednotlivými rozšiřujícími deskami, popř. pro výstupy navazujícího programovatelného pole.

Kompletní elektronické schéma základní desky je uvedeno v příloze P I.

Finanční náklady na elektronické součástky pro osazení základní desky jsou dle aktuálních cen přibližně 550Kč. Veškeré součástky základní desky jsou doporučovány osazovat do patič DIL, aby se umožnila jednodušší oprava vadné součástky. To platí i pro odporové pole SIL. Kontaktní svorkovnice není nutno osazovat, pokud budou kabelové spoje přímo pájeny.



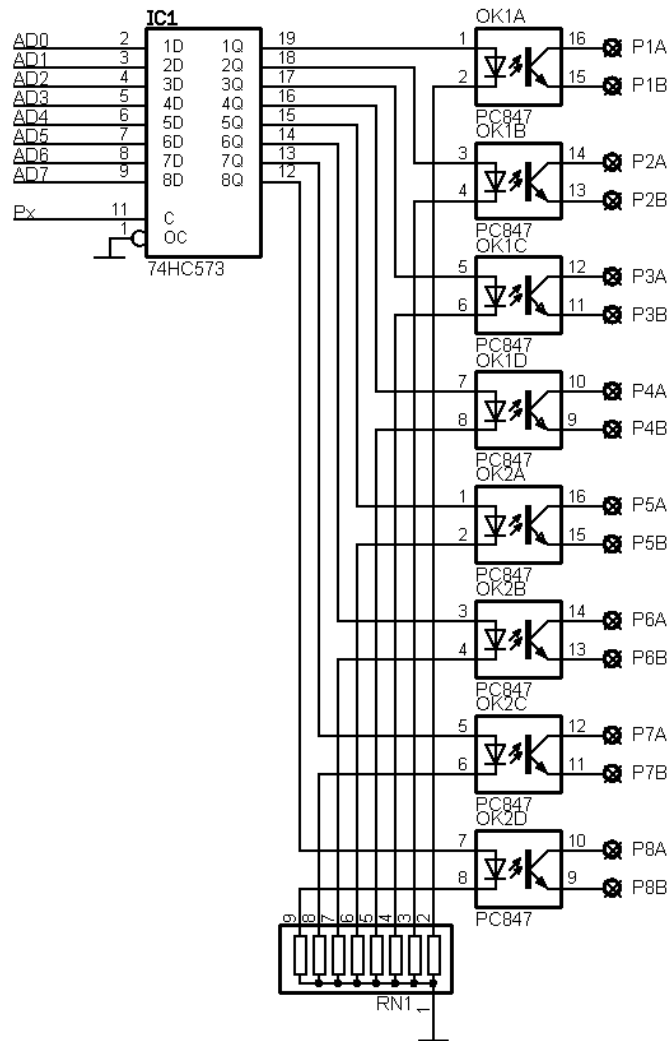
Obr. 16. Zapojení vývodů rozšiřujícího konektoru

#### 4.1.2 Spínací modul (WRITER)

Spínací modul je výstupním modulem, připojeným k segmentové sběrnici. Modul je rozdělen na dvě základní části:

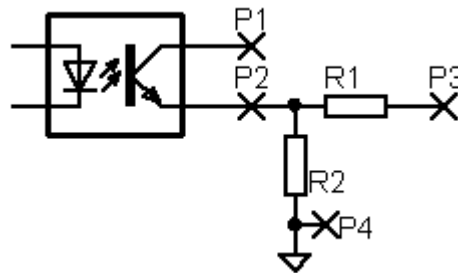
- část spínací,
- část řídicí.

Spínací část zajišťuje bezpečné oddělení provozních napájení a spínání výstupních vodičů a zajišťuje trvalý stav spínacího prvku. Oddělení je realizováno za pomoci optočlenů, paměťová buňka pomocí výstupního Latch. Zapojení 1/8 výstupů modulu je na Obr. 17.



Obr. 17. 8 opticky oddělených spínačů

Latch 74HC573 se chová jako paměťová buňka a v případě aktivního signálu Px přesune nový stav na výstupy. Signál Px se získává z řídicí části a za x se dosazuje číslo spínací větve (1 až 8).



Obr. 18. Zapojení odděleného výstupu

Doporučené zapojení oddělené výstupní části je na Obr. 18. Všechny letovací body jsou na desce plošných spojů doporučeny. Bod P1 je určen pro připojení vstupního spínaného napětí. Budeme-li pracovat se spínacím vedením až v místě určení, můžeme vynechat rezistory a použijeme vývodu P2. Pokud budeme chtít nastavit výchozí napájení již na modulu, je možné jej osadit rezistorovým polem SIL a pro rezistory R2 nebo DIL pro rezistory R1. Hodnoty rezistorů R1 a R2 doporučují v hodnotách  $1K\Omega$  pro R1 a  $10K\Omega$  pro R2.

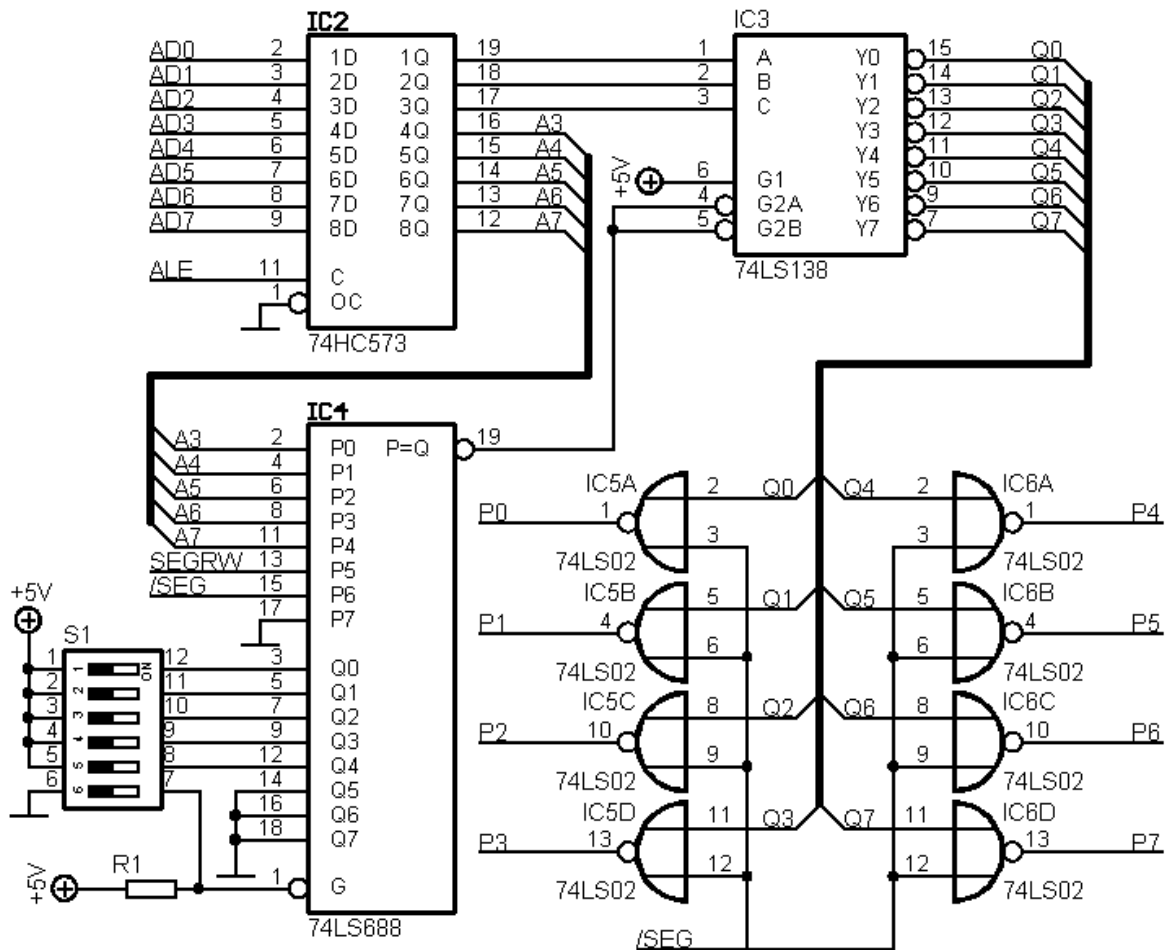
Řídící část je soustava integrovaných a Latch obvodů zajišťující chování modulu jako paměťové buňky. Jelikož by bylo zcela neekonomické vytvářet desítky malých modulů, je spínací modul navržen tak, aby jeho řídicí část byla co nejjednodušší, ovšem co nejefektivnější co do počtu spínaných vstupů, tak do počtu řídicích obvodů.

Na jednom segmentu je umožněno mít až 256 adresovatelných osmibitových buněk. To činí celkem 2048 výstupů. Každý spínací modul umožňuje až 64 výstupů, jelikož zajišťuje 8 adresovatelných osmibitových buněk. Každý výstupní modul lze tedy adresovat vždy v násobku 8 (00h – 07h, 08h – 0Fh, 10h – 17h ..., F8h – FFh). K nastavení detekovaného segmentu slouží DIP switch se šesti přepínači. Pět přepínačů je k nastavení adresy, šestý je pro centrální vypnutí modulu. Toto lze i řešit výměnou DIP switche za jumperové pole.

Na Obr. 19 lze vidět zapojení řídicí části spínacího modulu. Sběrnice s multiplexovanou adresou a daty je demultiplexována obvodem IC2. Výběr počáteční adresy v segmentu provede uživatel přepínači 1 – 5 v poli S1. Přepínačem 6 v tomto poli se zapínají a vypínají následné obvody. Integrovaný obvod IC4 je osmibitový komparátor, který provede kontrolu horní části adresy (A3 – A7), řídicích signálů s nastavením modulu a je-li modul zapnut, nastavuje výstup P=Q na negovanou logickou úroveň bitovému součinu.

Je-li rozsah adresy pro uvedený modul, je do povolovacích bran obvodu IC5 přivedena logická 0, která způsobí shození jedné z linky Q0 až Q7 na nízkou úroveň, dle vybrané adresy. Můžeme tedy stanovit, že aktivní signál Q0 až Q7 je offset od základní adresy modulu. Tyto signály je nutno negovat, jelikož ve spínací části aktivují tyto signály jeden z optických bloků.

Význam a nastavení jednotlivých přepínačů v poli S1 je uvedeno v příloze P II.



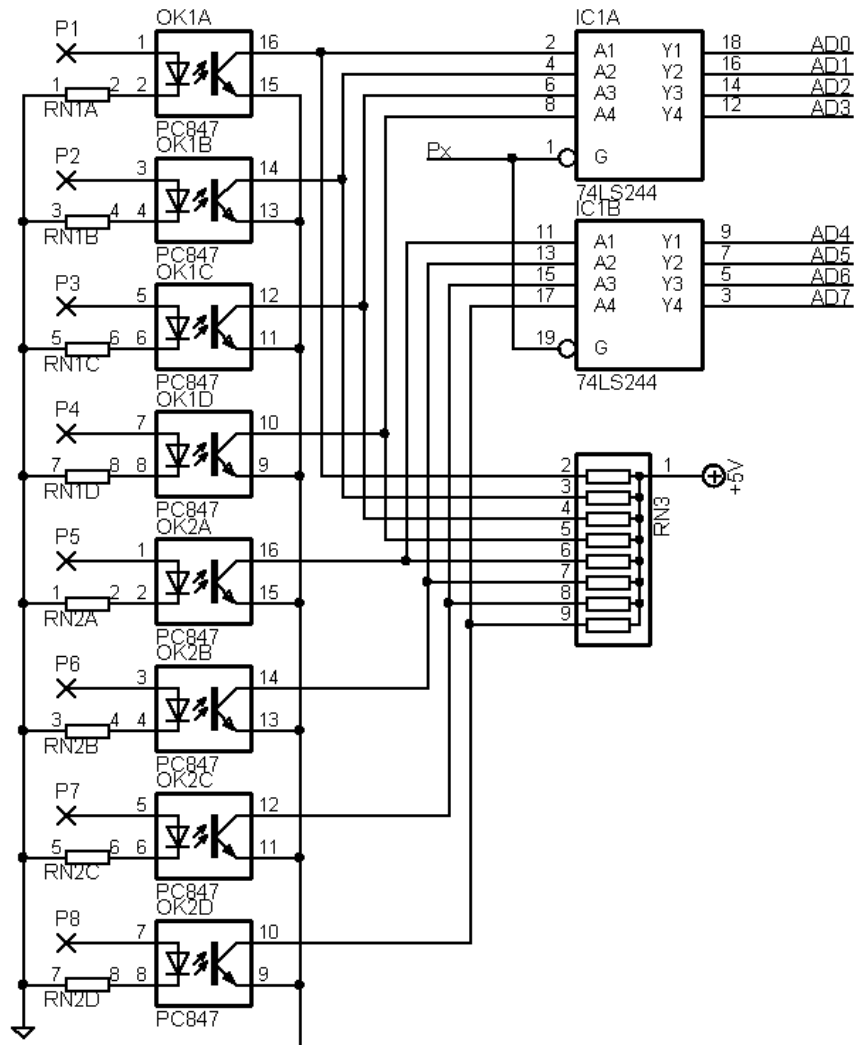
Obr. 19. Řídící část spínacího modulu

### 4.1.3 Snímací modul (READER)

Snímací modul umožňuje snímání stavu jeho vstupů přes segmentovou sběrnici. Princip tohoto modulu je obdobný spínacímu modulu popisovaném v předchozí kapitole. Stejně jako v případě výstupního modulu, je i vstupní navrhován s ohledem na maximální efektivitu modulu. Obsahuje tedy taktéž 64 vstupů, adresuje se pomocí přepínačů a to v násobcích 8 a na segmentovou sběrnici jich můžeme dát až 32, tedy opět 2048 vstupů na segmentu.

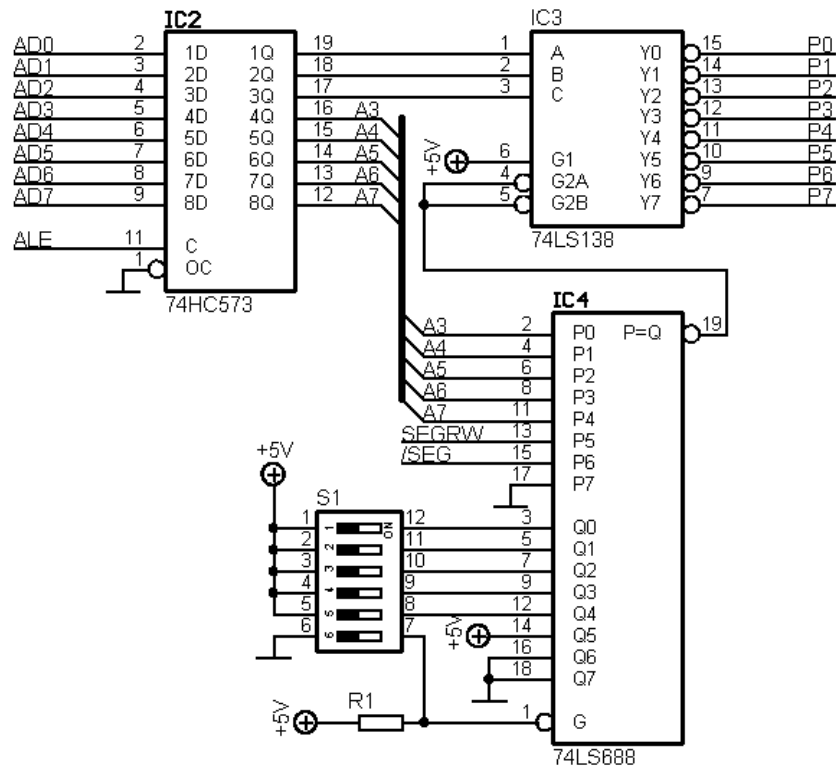
První jeho část, část snímací, je oddělena opět pomocí optočlenů. Schéma je uvedeno na Obr. 20. Odporové pole v provedení DIL (RN1 a RN2) nastavují pracovní jas pro svítivou diodu v optočlenu. Sepne-li tranzistor optočlenu, spojí vstupní signál přímo na zem, čímž získáme logickou 0 na vstupu. Výchozí stav (log.1) přivádíme pomocí pole zdvihacích rezistorů (RN3). Z popisu je zřejmé, že informace předávající do řídicí jednotky jsou invertované. Je-li optočlen sepnut, tedy prochází-li proud vnějším obvodem, je na výstupu

generována logická nula. Pomocí 74LS244 si data v případě potřeby přesuneme na multiplexovanou adresně-datovou sběrnici. Kdy tam stav přesuneme nám oznámí řídicí část modulu pomocí signálu Px, kde x je číslo větve (1 – 8) vstupů modulu.



Obr. 20. Zapojení osmi vstupů

Řídicí část modulu je obdobná řídicí části spínacího modulu. Jedná se totiž o stejný princip, pouze se musí dát pozor na jiné logické návaznosti. Ty jsou dány změnou budiče sběrnice. V zapojení snímacích vstupů má snímací část použity obvody 74LS244, které se aktivují v úrovni log. 0, na rozdíl od předchozího budiče aktivovaném v úrovni log. 1. Proto je v zapojení (Obr. 21) vynechána negace výběrového signálu. Další podstatná změna je u obvodu osmibitového komparátoru, kde zjišťujeme stav signálu SEGRW v úrovni log. 1. Ostatní je shodné se spínacím modulem, popisovaném v předchozí kapitole.

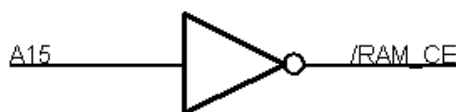


Obr. 21. Řídící část snímacího modulu

#### 4.1.4 Programový kód selektoru

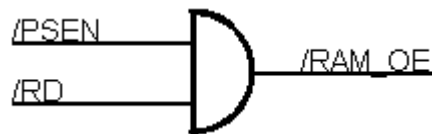
Selektor, jak již bylo popsáno dříve, je tvořen programovatelným logickým polem. Toto logické pole nahrazuje několik různých hradel v jeden obvod. Můžeme tím získat úsporu součástek na desce plošných spojů (např. kvůli jednomu hradlu NAND je zbytečné pořizovat celý integrovaný obvod) a tím i zmenšíme velikost této desky. Programovatelné pole může emulovat mnoho logických funkcí a může být několikrát programováno.

Začněme paměťovým dekodérem. Vstupně/výstupní porty jsou adresovatelné v rozsahu 0x0000 až 0x7FFF a SRAM je adresovatelná od 0x8000 do 0xFFFF. Při převodu do binární podoby bychom zjistili, že můžeme použít vývod mikrokontroleru A15 jako řídicí signál pro výběr typu rozsahu (I/O nebo SRAM). Tímto pro SRAM potřebujeme signál A15 invertovat, abychom získali řídicí signál Chip Select (/CS) pro obvod SRAM a tedy rozsah 0x8000 až 0xFFFF. Tento případ je uveden na Obr. 22.



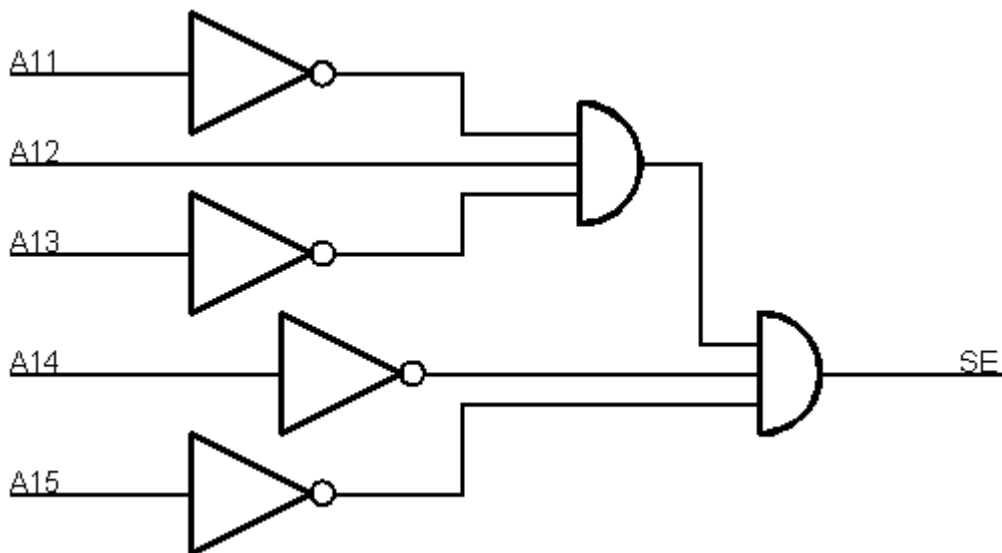
Obr. 22. Selektor - /RAM\_CE

SRAM nebudeme nijak moc zvlášť omezovat, rovnou budeme počítat s možnostmi, které mohou v budoucnu nastat. Co mám na mysli? Paměť SRAM se může používat nejen jako „odkladistiště“ dočasných dat, ale také pro spouštění uživatelských programů. Např. může nastat situace, že si stáhneme z obslužného pracoviště nebo diagnostického zařízení část programu, který chceme spustit. Nejjednodušší je nahrát jej právě do paměti RAM do volného prostoru. Využijeme tedy externí paměť jako paměť programu i dat. Tohoto může být jednoduše docíleno logickou operací AND signálů  $\overline{RD}$  a  $\overline{PSEN}$ , uvedenou na .



Obr. 23. Selektor -  $\overline{RAM\_OE}$

Signál SE (Segment Enable), který selektor generuje se používá k aktivaci výběru jedné z osmi segmentových sběrnic. Tyto sběrnic jsou dostupné na adresách 0x1000 až 0x17FF. Tento rozsah můžeme z dostupných adresových signálů získat pouze tak, že vytvoříme následující podmínku: singály A11, A13 – A15 byly vždy log. 0, A12 vždy log. 1 a na ostatních nižších nezáleží. Elektronicky vytvořená podmínka je na Obr. 24.



Obr. 24. Selektor – SE

Výstupní signál pro načtení konfigurace si umístíme namísto jednoho čtecího segmentu, tedy do prostoru vstupně/výstupních modulů (0x0000 – 0x7FFF) a to na úplný konec, tedy posledních 256 bajtů (0x7F00 – 0x7FFF). Čtením kterékoliv této adresy vždy získáme aktuální stav nastavovacích přepínačů. Požadovaného rozsahu získáme tak, že budou signály



A8 – A14 v log. 1 a A15 v log. 0. Taktéž se jedná pouze o čtení, proto přidáme do podmínky ještě signál /RD, který musí být v log. 0.

Řídící linku pro výběr směru zápisu dat na segmentové sběrnici jsem zvolil následující významy pro stavy signálu SEGRW:

- a) je-li SEGRW = log.1, pak jde o čtení z modulů do základní desky;
- b) je-li SEGRW = log.0, pak jde o zápis do modulů ze základní desky.

Tento řídicí signál lze jednoduše generovat ze signálů /RD a SE, přičemž SEGRW je v log. 1 právě tehdy, pokud /RD je log. 0 a SE je log. 1.

Již tímto výčtem lze uvážit, že použitím programovatelného logického pole lze ušetřit 6 integrovaných obvodů, 13 logických hradel. Zápis logických funkcí v programovatelném logickém poli bude pro překladač v následujícím tvaru (program je průběžně komentován):

Nejdříve se vytvoří definice použitého programovacího pole:

```
1 CHIP mCPU GAL20V8
```

Definice názvů jednotlivých vstupů a výstupů:

```
2 a8=1 a9=2 a10=3 a11=4 a12=5 a13=6 a14=7 a15=8
3 psen=9 rd=10 wr=11 p34=14 p35=23
4 ram_ce=15 ram_oe=16 se=17 ce=18 segrw=19
```

Funkční tabulka obsahuje všechny logické operace, které potřebujeme v programovacím poli provést:

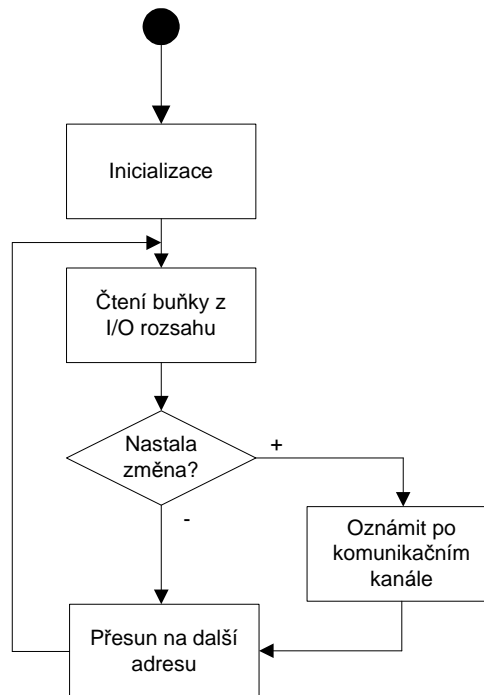
```
5 EQUATIONS
6 ram_ce = /a15
7 ram_oe = rd * psen
8 se = /a11 * a12 * /a13 * /a14 * /a15
9 ce = a8 * a9 * a10 * a11 * a12 * a13 * a14 * /a15 * /rd
10 segrw = rd * se
```

Z uvedeného programu lze poznat, že vstup wr není vůbec použit, stejně jako p34 a p35. To je z důvodu jednodušších budoucích upgrade, kdy může nastat, že budeme tyto řídicí signály na některé logické funkce využívat.

Pro kompilaci funkční tabulky do formátu JEDEC potřebujeme PLD assembler. Ve vyhledávači Gogole.com jsem našel např. OPAL jr 2.0, který je velmi jednoduchý na obsluhu. Pro zápis souboru ve formátu JEDEC používám GAL programátor GALBLAST, viz [5].

#### 4.1.5 Programový kód mikrokontroleru

Do Flash paměti mikrokontroleru AT89S8252 se zapisuje program ve formátu Intel HEX. Tento formát můžeme získat z různých typů překladačů, které podporují mikrokontrolery řady x51. Osobně jsem využil volně dostupný překladač přímo z internetových stránek výrobce mikrokontroleru [6], který překládá programový kód psaný v assembleru.



Obr. 25. Hlavní smyčka aplikace

Program se skládá z hlavní programové smyčky a obsluhy jednotlivých přerušení. Blokové schéma hlavní smyčky aplikace je na Obr. 25. Po úvodním nastavení a inicializaci proměnných aplikace stále dokola zjišťuje aktuální stav všech vstupů a změnu oznámí přes komunikační kanál druhému systému. Nyní si rozeberme základní části programu.

```

1 $NOMOD51
2 $INCLUDE (89S8252.MCU)

```

Nejdříve sdělujeme překladači jaký mikrokontroler jsme si zvolili. Překladač pak ve zvoleném definičním souboru (89S8252.MCU) definuje veškeré dostupné názvy portů a paměťových oblastí. My jsme se pomocí uvedených dvou řádků zvolili mikrokontroler 89S8252.

```

3 DELKABUFFERU EQU 3
4         DSEG
5 BUFFIN: DS DELKABUFFERU      ;vstupní buffer

```

```

6 BUFDEL: DS 1 ;délka do zaplnění bufferu
7 CONFIG: DS 1 ;poslední nastavení
8 BSEG
9 SOUT: DBIT 1 ;zapnutý nebo vypnutý přenos RS-232
10 USERAM: DBIT 1 ;zapisovat do paměti RAM nebo přímo do ka-
nálu?

```

Definice proměnných je vcelku jednoduchá, když člověk ví, jaké a kam je umístí. My jsme využili dva typy proměnných: bajt a bit. Na počátku definujeme hodnotu výrazu DELKABUFFERU, a to proto, abychom při změně definice komunikačního protokolu nemuseli celý programový kód procházet a opravovat všechny místa. Proměnná CONFIG je deklarována spíše pro ukázkou možného pozdějšího využití.

```

11 CSEG AT 0000H
12 AJMP MAIN
13 ORG 0003H
14 RETI
15 ORG 000BH
16 RETI
17 ORG 0013H
18 RETI
19 ORG 001BH
20 RETI
21 ORG 0023H
22 AJMP SERIAL

```

Důležitá je obsluha vstupních vektorů. Zde si hlavně nadefinujeme skoky na místa, ve kterých máme příslušný kód. Vektory jsou v pořadí: hlavní vektor po zapnutí, vnější přerušení 0, čítač/časovač 0, vnější přerušení 1, čítač/časovač 1, sériový kanál nebo SPI. V našem případě chybí ještě obsluha čítače/časovače 2, ovšem ten by zde byl zcela zbytečný, jelikož je využíván ke generování přenosové rychlosti sériového portu. Ve zvoleném režimu nikdy přerušení nespouští.

V následující části programového kódu je uvedeno hlavní tělo. Po připojení mikrokontroleru k napájecímu napětí je spuštěn vektor 0000H, který odkazuje právě na návěští MAIN. Úlohou následujících řádků je zajistit nastavení proměnných na základě vnějšího nastavení přepínačů. Hlavní částí této inicializace je nastavení komunikace sériového kanálu RS-232.

```

23 MAIN: MOV SP,#05FH ;umístnění zásobníku
24 INITCFG: ;čtení konfigurace
25 MOV P2,#7FH ;Nastavíme čtení konfigurace
26 MOV R0,#0 ;první buňka

```

```
27      MOVX A,@R0
28      MOV CONFIG,A          ;uložíme nastavení
29      RLC A                  ;A.7 -> C
30      CPL C                  ;negace C=zapnout komunikaci?
31      MOV SOUT,C            ;nastavíme sériovou komunikaci
32      RLC A
33      JB SOUT,INIT232      ;zapnuta komunikace přes RS-232?
34      RLC A                  ;ignorování nastavení RS-232
35      AJMP AFTER232
36 INIT232:
37      SETB SOUT
38      MOV SCON,#01010000B   ;8bit+1stop-bit
39      JC INIT232A          ;první část přenosové rychlosti
                           ;(A6=off => init232a)
40      RLC A
41      JC INIT232B          ;druhá část při první (a6=on,a5=off
                           ;=>init232b)
42      MOV RCAP2H,#HIGH 65523 ;57600Bd
43      MOV RCAP2L,#LOW 65523
44      AJMP INIT232END
45 INIT232B:
46      MOV RCAP2H,#HIGH 65497 ;19200Bd
47      MOV RCAP2L,#LOW 65497
48      AJMP INIT232END
49 INIT232A:
50      RLC A
51      JC INIT232C          ;druhá část při první (A6=off,A5=off
                           ;=> init232c)
52      MOV RCAP2H,#HIGH 65516 ;38400Bd
53      MOV RCAP2L,#LOW 65516
54      AJMP INIT232END
55 INIT232C:
56      MOV RCAP2H,#HIGH 65458 ;9600Bd
57      MOV RCAP2L,#LOW 65458
58 INIT232END:
59      MOV T2MOD,#0          ;zvolení režimu č/č 2
60      MOV T2CON,#00110100B
61      MOV IE,#10010000B    ;zapnout přerušení
62      MOV R0,#BUFFIN       ;adresa komunikačního bufferu
63      MOV BUFDEL,#DELKABUFFERU ;délka bufferu
64 AFTER232:
65      RLC A                  ;ukládat do RAM?
```

66

MOV USERAM,C

Na řádku 23 si nastavujeme zásobník na adresu 0x5F. Tato adresa je dostatečně daleko od bitově adresovatelného prostoru, který by mohl zásobník při výchozím nastavení ovlivnit. Čtení konfigurace přepínačů je provedeno osmibitovým přesunem kterékoliv adresy. Adresu konfigurace jsme nastavili přenosem na port P2. Pomocí bitové rotace registru akumulátoru přes příznak C získáváme právě toužený nastavovací bit právě do příznaku C. Takovýmto způsobem se s bitovou informací lépe pracuje, než její předchozí zamaskování. Navíc, nám to nezmění obsah registru, kdybychom rotaci udělali osmkrát, měli bychom obsah registru zase zpět. Není-li nastaven přenos pomocí sériového kanálu, je nastavení čítače/časovače 2 plně přeskočeno. Lze jej tedy využít k jiným účelům. V poslední řadě se nastavuje příznak využívání externí paměti dat. Není-li paměť dat zvolena, bude aplikace stále dokola vypisovat aktuální zjišťovaný bajt I/O prostoru na výstupní komunikační kanál, jelikož není s čím porovnávat změny stavu.

Řádky 67 – 90 provádí cyklickou smyčku, procházející celý zvolený adresní prostor a znamenávající aktuální stavy, přičemž podle zvolené konfigurace umožňuje přeskočit práci s vnější pamětí.

```

67 ;*****
68 ;* MAIN LOOP
69 GPRST:  MOV DPTR,#1000H           ;počátek I/O
70 GPLOOP: MOVX A,@DPTR             ;načti stav ze snímače
71         JNB USERAM, NORAM        ;používáme RAM?
72         MOV B,A                   ;ulož do B
73         MOV A,DP0H                ;ulož high byte DPTR
74         PUSH ACC                   ;ulož high byte DPTR
75         ORL A,#10000000B           ;nastav na RAM
76         MOV DP0H,A
77         MOVX A,@DPTR              ;načti stav z RAM
78         XRL A,B                    ;porovnej A (RAM) a B (snímač)
79         JZ GPCONT                  ;není-li změna, pokračuj na GPCONT
80         JNB SOUT,GPCONT            ;nevysíláme změny?
81         ACALL SENDRAMBYTE          ;zaslání bajtu na RS-232
82 GPCONT: POP ACC                   ;obnov zpět na I/O
83         MOV DP0H,A
84         AJMP GPNEXT                ;a jdeme na další
85 NORAM:  JNB SOUT,GPNEXT            ;nevysíláme změny?
86         ACALL SENDRAMBYTE          ;zaslání bajtu na RS-232
87 GPNEXT: INC DPTR                  ;další byte

```

```

88          MOV A,DP0H
89          CJNE A,#00001000B,GPLOOP      ;opakuji GPLOOP dokud není A11=1
90          AJMP GPRST                    ;jinak nový cyklus

```

Funkce pro práci se sériovým kanálem jsou uvedeny na řádcích 91 – 106. První (W232) čeká na dokončení přenosu vysílaného znaku, aby se mohl vysílat další. Mikrokontrolery totiž nemají několika bajtové výstupní buffery, proto je třeba takovou čekací smyčku vytvořit ručně.

Druhá funkce SENDRAMBYTE zajišťuje správné pořadí vyslání tří bajtů komunikačním kanálem RS-232. Při návrhu vnějších obvodů je nutné si uvědomit, že tato část programu může trvat různou dobu v závislosti i např. na aktuálním vytížení systému Obslužného pracoviště.

```

91 ;*****
92 ;* Čekání na ukončení vysílání po RS-232
93 W232:   JNB TI,$
94         CLR TI
95         RET
96
97 ;*****
98 ;* Zaslání DPTR a ACC na RS-232
99 SENDRAMBYTE:
100        MOV SBUF,DP0H          ;pošli HIGH byte
101        ACALL W232
102        MOV SBUF,DP0L          ;pošli LOW byte
103        ACALL W232
104        MOV SBUF,A              ;pošli stav
105        ACALL W232
106        RET

```

Obsluha přerušení od sériového kanálu je uvedena na řádcích 107 – 144. Při přijetí bajtu pomocí sériového kanálu, nebo po vyslání bajtu na sériový kanál, je spuštěn vektor přerušení 023H.

```

107 ;*****
108 ;* Obsluha sériového kanálu RS-232
109 SERIAL: PUSH PSW      ;uložení registrů
110          PUSH ACC
111          JB TI,SERIV    ;test vysílání
112          CLR RI        ;nulování příznaku
113          DEC BUFDEL    ;sníží BUFDEL o 1

```

```

114      MOV @R0,SBUF      ;uložení do bufferu
115      JZ SERIS          ;A=0 po příjmu celého bufferu
116      INC R0           ;posuň pointer v bufferu
117      AJMP SERIV       ;přijímej dál
118 SERIS:  PUSH DP0H     ;uložíme DPTR
119      PUSH DP0L
120      MOV R0,#BUFFIN   ;všechny bajty přijaty
121      MOV A,@R0
122      CJNE A,#00FFH,SER2R ;nejedná-li se o příkaz, zapiš do pa-
                           ;měti
123      INC R0           ;přesuneme pointer na příkaz
124      AJMP CMDS        ;a zpracujeme jej
125 SER2R:  ANL A,#01111111B ;nastav rozmezí cílové buňky do I/O
                           ;prostoru

126      MOV DP0H,A
127      INC R0
128      MOV DP0L,@R0
129      INC R0
130      MOV A,@R0
131      MOVX @DPTR,A     ;uložení hodnoty do I/O prostoru
132      JNB USERAM,SERNOR ;používáme RAM?
133      ORL DP0H,#10000000B ;nastavení bitu pro RAM
134      MOVX @DPTR,A     ;uložení hodnoty do RAM
135      ANL DP0H,#01111111B ;zpětné maskování
136 SERNOR:  MOV SBUF,A   ;odeslání zpět zapsané hodnoty
137      ACALL W232
138 SERIK:   POP DP0L     ;obnovení DPTR
139      POP DP0H
140      MOV R0,#BUFFIN   ;příprav buffer
141      MOV BUFDEL,#DELKABUFFERU ;a jeho délku
142 SERIV:   POP ACC      ;obnovení registrů
143      POP PSW
144      RETI             ;ukončení přerušeni

```

Úlohou tohoto kódu je postupné zaplňování vstupního bufferu v paměti (řádky 113 – 117). Jakmile je buffer zaplněn, dojde k jeho vyhodnocování (řádky 118 – 122). Zde prochází první bajt testem shody s hodnotou 255 (0xFFh). Obsahuje-li tento bajt právě uvedenou hodnotu, jedná se o řídicí příkaz, který je zpracováván v samostatné funkci (CMDS). Od řádku 125 provedeme nejdříve bitové omezení na dolní polovinu adresovacího rozsahu pomocí bitové masky a následně na tuto adresu zapišeme zvolenou hodnotu. Pokud používáme vnější paměť dat, je tato hodnota zapsána také do této paměti pro další zpracování.

Nakonec je zapsaný bajt pro kontrolu vyslán zpět po komunikačním kanálu ovládací jednotce (řádky 136 a 137) a buffer nastaven pro nové přijímání znaků (řádky 140 a 141).

Zpracování řídicích příkazů pro software v řídicí jednotce je ve svém principu jednoduché a přitom plně efektivní pro další rozšiřování. Základní kód pro obsluhu příkazu vidíme na řádcích 145 – 161.

```

145 ;*****
146 ;* Zpracování příkazu pro RS-232
147 CMDS:   MOV A,@R0           ;zjištění příkazu
148         PUSH DP0H
149         PUSH DP0L
150         MOV DPTR,#CMDTAB     ;nastav DPTR na tabulku skoků
151         PUSH A
152         CLR C                ;pro jistotu nulujeme
153         SUBB A,#CMDLEN       ;odpočítáme maximální číslo
154         JC CMDBAD            ;přeskoč provádění, pokud je špatné
155         POP A
156         MOVC A,@A+DPTR       ;načti adresu příkazu
157         JMP @A+DPTR          ;skoč na příkaz
158 CMDBAD: POP A
159 CMDSR:  POP DP0L             ;sem se vrátíme a obnovíme DPTR
160         POP DP0H
161 CMDSK:  AJMP SERIK          ;ukončení obsluhy

```

Programový kód využívá konstrukce instrukce `JMP @A+DPTR`, která provede skok na adresu danou součtem obsahu akumulátoru s registrem `DPTR`. Na řádku 150 nastavíme registr `DPTR` na tabulku vektorů, což je počáteční pozice pro zjištění adresy vektoru.

Ochrana proti nežádoucímu přetečení je realizována konstantou `CMDLEN`, která je definována u tabulky vektorů. Její hodnota musí být vždy větší nebo rovna čísla příkazu. Řádkem 153 provedeme test správnosti čísla příkazu. Pokud je číslo větší než počet příkazů (nastavil se v `SUBB` příznak `C`), tak se následující řádek postará o přeskočení dalšího provádění příkazu. Každé zpracování příkazu se pak zpětně vrací na návěští `CMDSR`.

```

162 ;*****
163 ;* Tabulka skoků příkazů
164 CMDTAB EQU $
165         DB CMD_SOUTE        ;0 = zapnutí vysílání
166         DB CMD_SOUTD        ;1 = vypnutí vysílání
167         DB CMD_DUMP         ;2 = výpis obsahu RAM

```



```
168 CMDTABEND :
169 CMDLEN EQU CMDTABEND-CMDTAB
```

Tabulka seznam vektorů je vložen mezi návěští CMDTAB a CMDTABEND, pomocí kterých se počítá celkový počet příkazů do konstanty CMDLEN. Obsluha ukázkových příkazů je na posledních řádcích souboru:

```
170 CMD_SOUTE :      SETB SOUT
171                  AJMP CMDSR
172 CMD_SOUTD :      CLR SOUT
173                  AJMP CMDSR
174 CMD_DUMP :       PUSH DP0L
175                  PUSH DP0H
176                  MOV DPTR, #9000H
177 CMD_DUMP_LOOP :  MOVX A, @DPTR
178                  ACALL SENDRAMBYTE
179                  INC DPTR
180                  MOV DP0H, A
181                  CJNE A, #10011000B, CMD_DUMP_LOOP
182                  POP DP0H
183                  POP DP0L
184                  AJMP CMDSR
185                  END
```

## 4.2 Komunikační kanál RS-232

Každý komunikační kanál má své vlastní specifické parametry, které určují jakým způsobem se pomocí tohoto kanálu komunikuje s jiným systémem. Sériový kanál je v základním provedení určen ke komunikaci Řídící jednotky JOP s Obslužným pracovištěm JOP. Přes tento komunikační kanál se přenáší všechny změny na snímacích vstupech z řídicí jednotky a příkazy z obslužného pracoviště. Komunikace je elektronicky řízena zabudovanými asynchronními obvody UART na straně Obslužného pracoviště a obvodem MAX232 na straně Řídící jednotky.

Rychlost komunikace se může volit nastavením přepínačů na řídicí jednotce, obslužný program implementovaný na Obslužné pracoviště musí umět touto rychlostí komunikovat.

Přenos probíhá vždy přenesením 3 bajtů v ASCII módu. Význam jednotlivých bajtů lze vidět na Obr. 26. V prvních dvou bajtech se přenáší adresa měněné buňky a třetí bajt určuje její novou hodnotu.

	HI	LO	Value		HI	LO	Value	
	ADRESA		HODNOTA		ADRESA		HODNOTA	

*Obr. 26. Přenos komunikačního kanálu RS-232*

Adresa měněné buňky může být pouze v rozsahu vstupně/výstupních modulů. Pokud bude adresa vyšší, jedná se o příkaz některé z rozšiřujících desek. Základní software řídicí jednotky rozeznává příkaz zadáním hodnoty 0xFF do prvního bajtu. Následující bajt je identifikační číslo příkazu a poslední je parametr pro zvolený příkaz.

Obdobně je tomu i při oznamování obslužnému pracovišti o zjištěné změně snímaného vstupu řídicí jednotky. Nejdříve se zasílá 16 bitů adresy následované novou zjištěnou hodnotou.

### 4.3 Obslužné pracoviště JOP

Obslužné pracoviště JOP je realizováno osobním počítačem PC AT, vybaveným barevným monitorem, klávesnicí a myší. V praxi se běžně používají počítače řady Pentium, ovšem nebude-li počítač provádět zároveň funkci staničního zabezpečovacího zařízení, tzn. nemusí-li kontrolovat závislosti a správnosti volených údajů, může se použít i osobní počítač řady 486.

Operační systém instalovaný na obslužném pracovišti musí být podporován obslužným softwarem, obvykle se volí MS-DOS, lze však zvolit i volně šířitelnou platformu OS DOS. Obslužný program je většinou psán v programovacím jazyce Borland C++ nebo Microsoft C++, lze však pro začínající vývojáře využít také programovacích jazyků Pascal, či Basic.

Jelikož není v některých případech (a u zadavatele tak tomu bylo) problém použít výkonnější osobní počítač, kdy lze využít operační systém Microsoft Windows, budeme se věnovat hlavně této obsluze. Zvolili jsme programovací jazyk C++ Builder od firmy Borland.

Pro programovací jazyk C++ Builder existuje třída TSerial, kterou lze používat k elegantní práci se sériovým kanálem pod Windows. Třída disponuje velkým množstvím vlastností a metod, proto si uvedme pouze nutně potřebné.

Nejdříve je zapotřebí nastavit komunikační rychlost sériového portu. To provedeme změnou vlastnosti `BaudRate`. Pomocí dalších vlastností si nastavíme požadovaný způsob přenosu dat (`Parity`, `StopBits`, `ByteSize` a všechny možné časové `timeouty`). Nesmíme zapomenout na nastavení vstupního a výstupního bufferu pomocí metody `SetupComm`. Pak již jen reagujeme na podněty z grafického prostředí a v případě požadavku na změnu (např. simulace stisku tlačítka) přenášíme příkaz dle specifik komunikačního kanálu RS-232 řídicí jednotce. Data přenášíme po jednotlivých bajtech a to metodou `WriteByte`

Za pomoci časovačů si můžeme průběžně kontrolovat aktuální stav příchozího bufferu sériového portu a postupně načítat příkaz. Při dosažení délky informace (3 bajty) tuto informaci analyzujeme a provádíme patřičně nutné vizuální změny.

Použijeme-li programovací jazyk Pascal nebo Borland C++ pro DOS, ovládneme sériový port pomocí příkazu `PORT`. Podrobnější informace k tomuto řešení komunikace nalezneme v [7].

Propojení obslužného pracoviště JOP s řídicí jednotkou JOP je provedeno třížilovým sériovým kabelem v zapojení tzv. `null-modem`.

## 5 ROZŠÍŘENÍ ZÁKLADNÍHO ŘEŠENÍ

Základní řešení poskytuje vývojovou a funkční platformu pro realizaci simulačního zařízení řídicího zařízení JOP. Jelikož sestává ze tří základních rozšiřitelných modulů, které plně postačují pro základní řízení jakéhokoliv modelového kolejistiště, dalo by se říci, že další rozšiřování není potřebné. Opak je ovšem pravdou, jelikož existuje mnoho dostupných technologií, které se dají dále implementovat. V této části bych rád alespoň nastínil některé z nich.

### 5.1 Komunikační kanál Ethernet

Komunikace obslužného pracoviště a řídicí jednotky je někdy pomocí sériové linky RS-232, resp. RS-485 velmi nevhodné. Přispívá k tomu hlavně náchylnost na okolní rušení, vcelku krátký dosah kabelů a někdy i rychlost komunikace. V dnešní době bychom mohli dvě takové zařízení propojit pomalu kteroukoliv z dostupných technologií, ať už drátem, opticky, rádiovými vlnami nebo za pomoci připojení k síti internet. Každý další způsob komunikace přináší své klady, ovšem i zápory.

Navrhoval bych dále rozšířit takové zařízení o komunikaci v síti Ethernet s podporou protokolů ARP a TCP/IP. Komunikační kanál na bázi Ethernetu umožňuje svou technologií jednodušší dálkové řízení, monitorování, archivaci, diagnostiku a mnoho dalších vlastností. Staniční zabezpečovací zařízení může provádět self-testy s výsledky zobrazovanými jako webová stránka, nebo právě archiv provedených úkonů na jedné intranetové stránce systému.

Modul, který by základní desku rozšířil o Ethernetový komunikační kanál by sestával z čipu firmy Realtek řady 8019. Jedná se o integrovaný obvod umožňující 10Mbps přenos po TP kabelu („kroucená dvoulinka“).

Integrovaný obvod Realtek RTL8019AS umožňuje pracovat bez přídavné EEPROM paměti, kde má uloženu MAC adresu a nastavení I/O adresy, resp. IRQ. Tento obvod lze vcelku jednoduše připojit k osmibitové sběrnici s pevně nastavenou, nebo nastavitelnou adresou v systému. Taktéž volně dostupný ovladač pro osmibitové mikrokontrolery umožňuje jeho implementaci k takovému zařízení.

Modul by se připojoval přes rozšiřující slot a mohl by plně nahradit integrovaný komunikační kanál RS-232, který by se mohl využít ke komunikaci těch zařízení, které nevyžadují

vyšší rychlost, nebo přímo mezi několika osmibitovými řídicími jednotkami zapojenými v kruhové topologii.

## 5.2 Modul řízení trakce

Modul řízení trakce představuje zdánlivě jednoduchý modul pro spínání napětí do jednotlivých kolejových oddílů. Vyzdvihneme ovšem laťku trošku výše. Aby byla simulace plně věrohodná, nesmí jí také chybět určitá logika. Vždyť jen uvažujeme případ, kdy vlaková souprava zastavuje ve stanici. Nikdy to není takový pohyb naznačující „jedu po sem a pak stojím“. Nebo při omezené rychlostní jízdě lze přeci také simulující vlakovou soupravu zpomalit.

Hledal bych využití PWM generátorů, resp. další soustavu mikrokontrolerů, která by zajistila pulsně-šířkový pohon motorků. Tato cesta přináší další možnosti rozvoje zařízení. Existují (i když vcelku drahé) systémy, resp. lokomotivy, které lze řídit pomocí modulovaného signálu v kolejovém obvodu. Nevýhoda těchto systémů je, že nelze kombinovat klasické lokomotivy s těmito typy na jednom kolejišti. Toto je jeden z faktorů, proč je tak malý rozmach těchto „inteligentních souprav“. Výměna celého vozového parku může uživatele přijít na několik desítek tisíc.

Při využití mikrokontrolerové soustavy lze zjistit v kterém obvodu se nachází jaký typ lokomotivy a podle toho stále zasílat do toho obvodu pouze takový signál, kterém daná souprava „rozumí“.

Modul řízení trakce by zcela jistě nebyl jen jeden na rozsáhlejší simulované kolejiště a proto by i pro něj měla segmentová sběrnice své uplatnění.

## ZÁVĚR

Zařízení, které je popsáno v této diplomové práci je všestranně využitelné zařízení, které i když svým prvotním zaměřením je natolik přístupné, že se dá využít prakticky i ke svévolnému řízení a snímání sledovaného zařízení. Díky několika součástkám na jednotlivých modulech není nikterak technicky náročné pro realizaci, díky čemuž je také dostupné i začínajícím.

Prvotní záměr diplomové práce byl navrhnout zařízení, které by umožnilo ve spolupráci se stávajícím zařízením modernější obsluhu stávajícího staničního zabezpečovacího zařízení. Díky své modularitě je dostupný prakticky na jakkoliv velké kolejiště se stovkami řídicích a také vizuálních prvků. Zařízení umožňuje být „roztroušeno“ po celém modelovém kolejišti v osmi směrech díky osmi osmibitovým segmentům. Je také úctyhodný počet vstupních a výstupních bodů v řádu stovek tisíců. Přitom maximální stanovená časová reakční doba dle ZTP je ještě vyšší než čas, který zabere plně obsazenému systému zareagovat na všechny změny.

Náklady na pořízení kompletního zařízení od téměř monopolních společností se pohybují až v desetinásobcích pořizovacích nákladů tohoto zařízení. Cenu obvykle navyšují vývoje, kterých nejsou někteří domácí, resp. zájmoví uživatelé modelových kolejišť, schopni, nebo je nemohou zaplatit. Toto řešení pomáhá postavit základní fungující hardwarový řídicí systém jen s minimem znalostí v oboru. Platformu obslužného software již musí implementátor zvolit sám takovou, která mu bude vyhovovat i z hlediska následných změn a s přihlédnutím na použitý hardware osobního počítače.

Největší výhodou tohoto zařízení je i jeho další oblast využití, a to jako malý vývojový kit procesorů AT89S8252. Může tak pomoci při získávání zkušeností s prací s mikrokontrolery za minimální náklady. Jelikož je základní deska rozšiřitelná o další desky, lze ji rozšířit např. o senzory teplotní, D/A převodníky, budiče dalších sběrnic (např. I<sup>2</sup>C), nebo přímo novým řídicím systémem a pomocí vývojového kitu si ladit software mikrokontroleru.

Již nyní je jasné, že implementace takového systému je vhodná pro různé zaměření a pro navrhovanou cílovou skupinu (zadavatel a zájmové kroužky) je to dobrý a levný odrazový můstek pro další rozvoj a udržení trendu s vývojem staničních zabezpečovacích zařízení.

Realizovaný řídicí software mikrokontroleru vyhoví všem základním implementacím zařízení. Stejně tak software v programovatelné logickém poli PLD.

**SEZNAM POUŽITÉ LITERATURY**

- [1] MATOUŠEK, David. *Práce s mikrokontrolery ATMEL AT89S8252 2. díl*. Praha: BEN – Technická literatura, 2002. 290s ISBN 80-7300-066-0.
- [2] *Data Sheet PIC16F87X 28/40-Pin 8-Bit CMOS FLASH Microcontrollers*. Microchip Technology Incorporated. Printed in the USA. 2001  
„<http://www.microchip.com/>“
- [3] *Data Sheet AT89S8252 8-bit Microcontroller with 8K Bytes Flash*. Atmel Corporation. USA  
„<http://www.atmel.com/>“
- [4] *AEC\_ISP.EXE PC Based AT89S ISP (In System Programming) Software*. AEC Electronics Ltd. New Zéland  
„<http://aec-electronics.co.nz/>“
- [5] *GALBLAST – programátor GAL* [online]. Dostupný z WWW:  
<<http://www.geocities.com/ResearchTriangle/Forum/8070/galblast.htm>>
- [6] *Archiv software společnosti Atmel Corporation k x51*. Dostupný z WWW:  
<[http://www.atmel.com/dyn/products/tools.asp?family\\_id=604](http://www.atmel.com/dyn/products/tools.asp?family_id=604)>
- [7] KAINKA, B. *Využití rozhraní PC*. Praha: HELL, 1998, 133 s.
- [8] *Informační materiály AŽD Praha, s.r.o.*  
<<http://www.azd.cz/>>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

ALU	Aritmetická jednotka mikrokontroleru
AVR	Rodina RISC mikrokontrolerů firmy ATMEL
CPU	Procesor zařízení
ČD	České Dráhy, a.s.
GAL	Generic Array Logic (základní programovatelné pole)
JOP	Jednotné obslužné pracoviště
LATCH	Záchytný obvod
MCU	Mikrokontroler Unit, mikrokontroler
PLD	Programovatelné logické zařízení
PWM	Pulsně šířková modulace
SPI	Seriál P
SZZ	Staniční zabezpečovací zařízení
ZTP-JOP	Základní technické požadavky na Jednotné obslužné pracoviště



**SEZNAM OBRÁZKŮ**

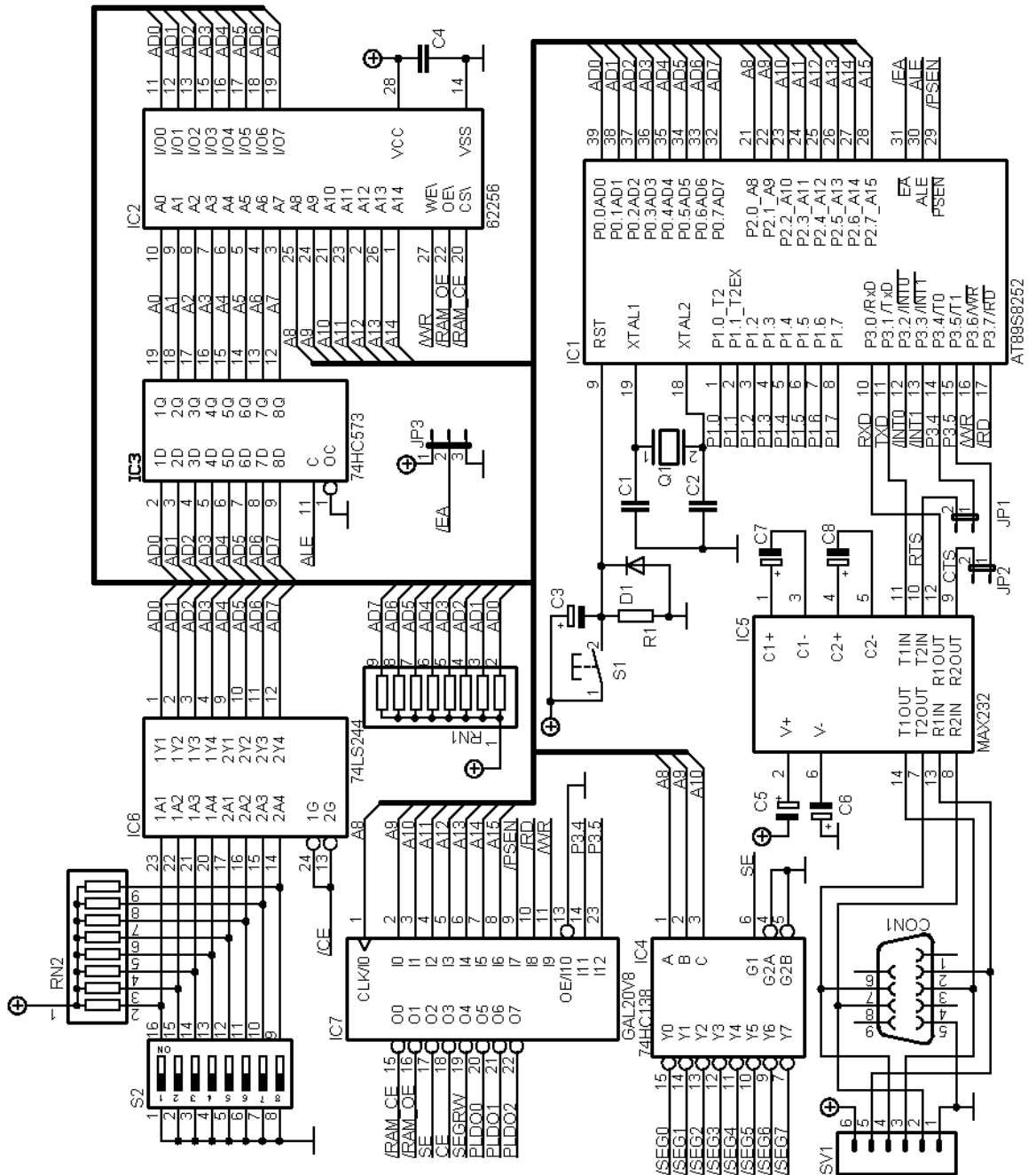
Obr. 1. Programátor Presto .....	13
Obr. 2. Mikrokontroler AT89S8252 .....	14
Obr. 3. Programátor AT89S8252.....	15
Obr. 4. Příklad zobrazení kolejiště dle ZTP-JOP .....	24
Obr. 5. Blokové schéma dopravního sálu a) před implementací; b) po implementaci.....	27
Obr. 6. Blokové schéma řídicí jednotky JOP.....	28
Obr. 7. Rozdělení vnější datové paměti .....	29
Obr. 8. Blokové schéma základní desky řídicí jednotky JOP.....	34
Obr. 9. Zapojení mikrokontroleru.....	35
Obr. 10. Připojení vnější paměti dat .....	36
Obr. 11. Segmentový dekodér .....	37
Obr. 12. Komunikační kanál RS-232.....	37
Obr. 13. Připojení nastavovacích přepínačů .....	38
Obr. 14. Zapojení vývodů selektoru .....	39
Obr. 15. Napájení základní desky .....	40
Obr. 16. Zapojení vývodů rozšiřujícího konektoru .....	42
Obr. 17. 8 opticky oddělených spínačů .....	43
Obr. 18. Zapojení odděleného výstupu.....	43
Obr. 19. Řídicí část spínacího modulu.....	45
Obr. 20. Zapojení osmi vstupů.....	46
Obr. 21. Řídicí část snímacího modulu.....	47
Obr. 22. Selektor - /RAM_CE .....	47
Obr. 23. Selektor - /RAM_OE .....	48
Obr. 24. Selektor – SE .....	48
Obr. 25. Hlavní smyčka aplikace.....	50
Obr. 26. Přenos komunikačního kanálu RS-232.....	58

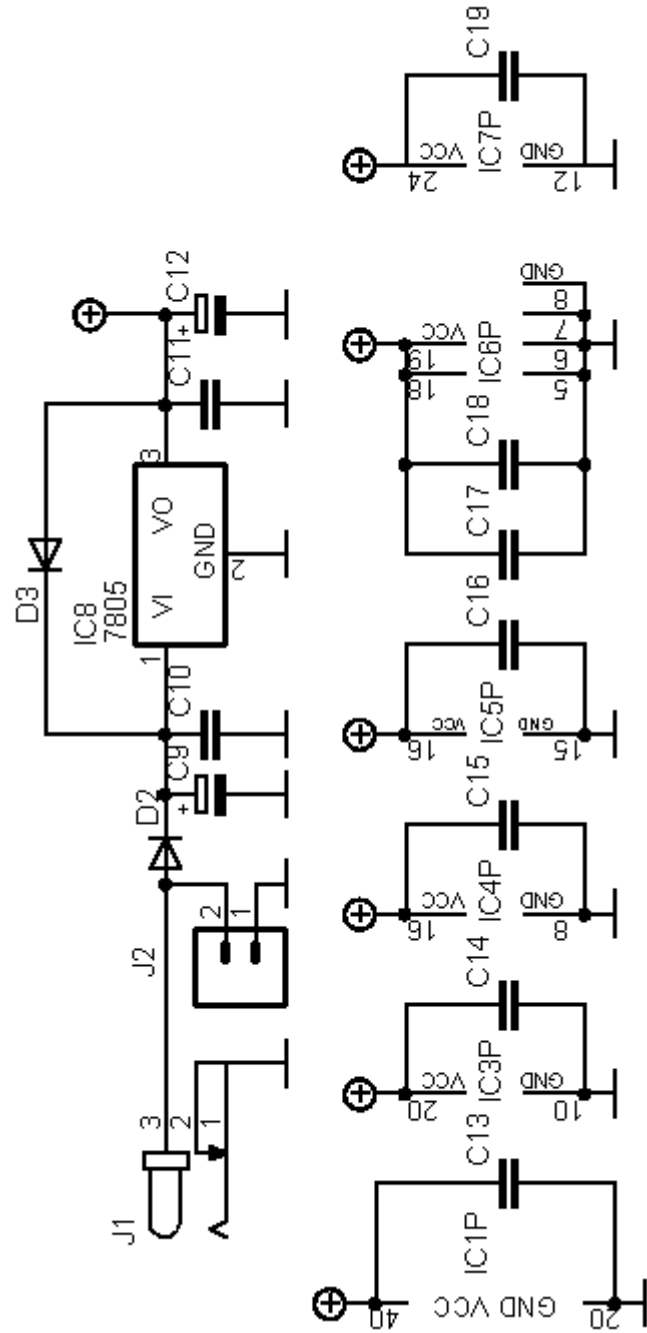
**SEZNAM TABULEK**

tab. 1. Typy a označení bloků AŽD-71 .....	22
tab. 2. Přenosová rychlost a příslušná hodnota registrů RCAP2 .....	31
tab. 3. Významy konfiguračních přepínačů .....	39
tab. 4. Popis pinů segmentové sběrnice .....	41

## SEZNAM PŘÍLOH

- P I Schéma řídicí jednotky
- P II Schéma spínacího modulu
- P III Schéma snímacího modulu





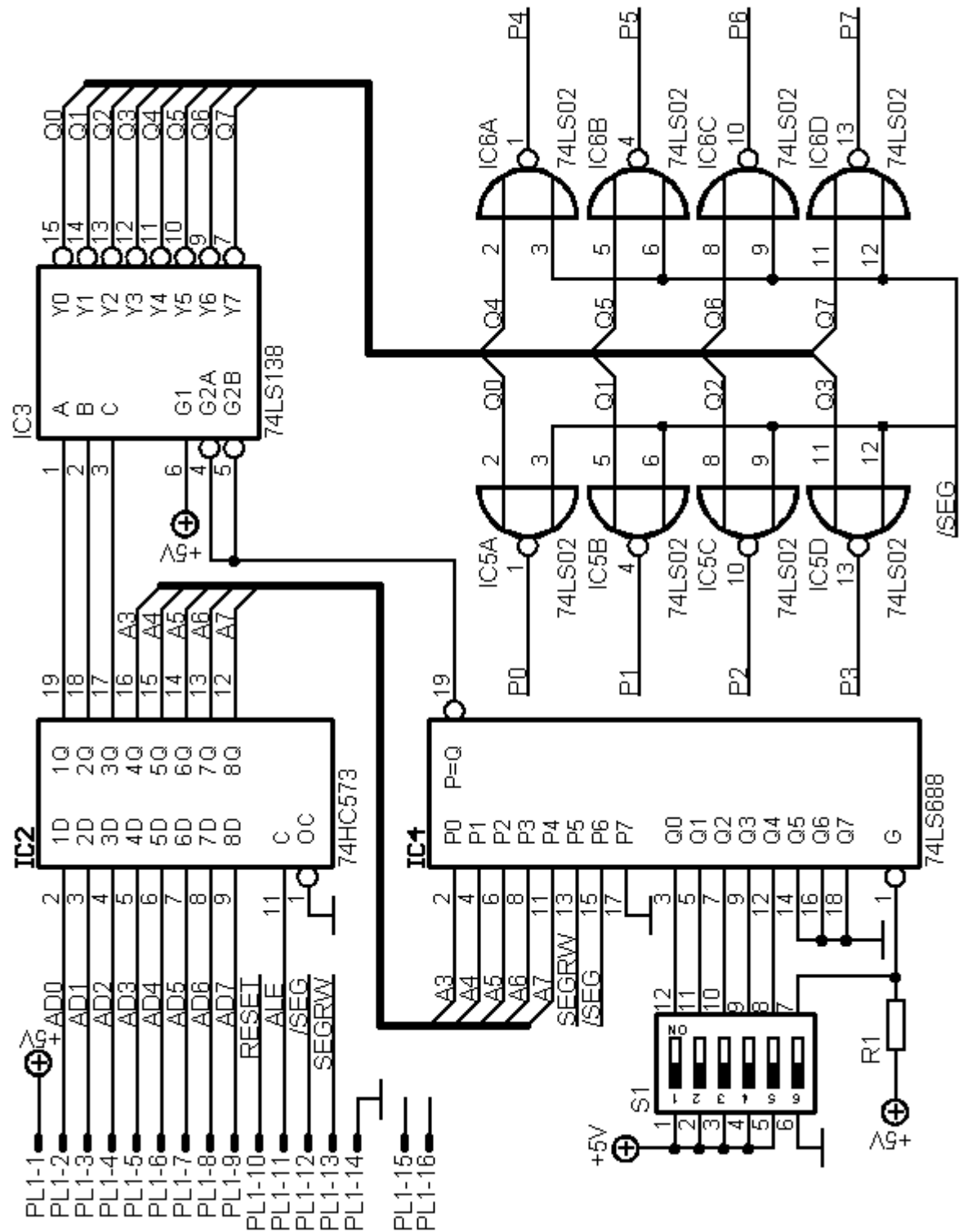
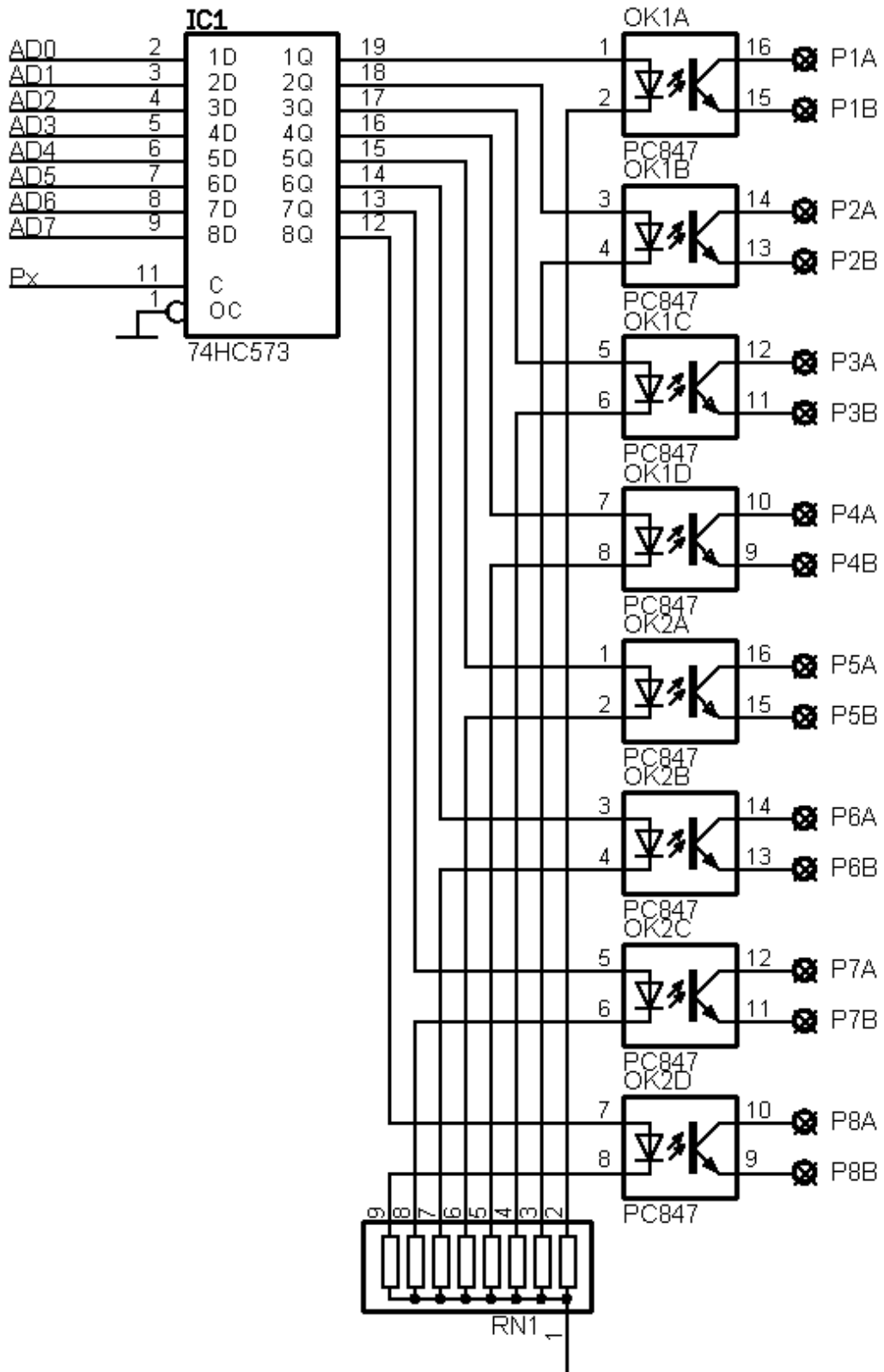


Schéma obsahuje zapojení jednoho optického výstupu. Dalších 7 je identických, rozlišují se pouze signálem Px, kde x je číslo větve.



# PŘÍLOHA P III: SCHEMA SNÍMACÍHO MODULU STRANA 1/2

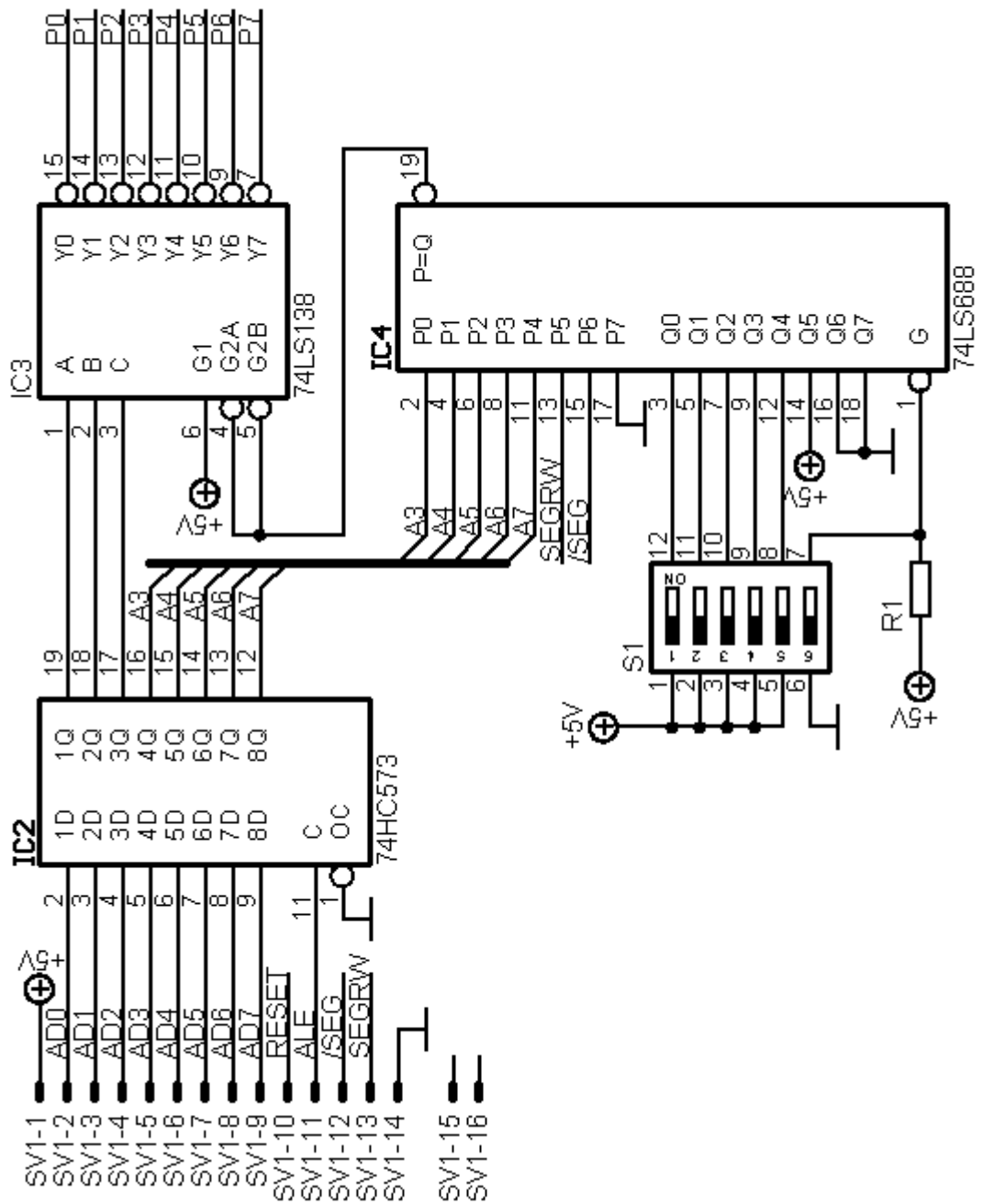




Schéma obsahuje zapojení jednoho optického výstupu. Dalších 7 je identických, rozlišují se pouze signálem Px, kde x je číslo větve.

