

Obsah

Obsah

Seznam ilustrací.....	iii
Seznam tabulek.....	iv
1 Analýza jednoduchých přístupových systémů.....	v
2 Specifikace systému.....	viii
3 Popis komponent systému.....	ix
3.1 RFID.....	x
3.1.1 Technologie MIFARE.....	xi
4 Implementace hardware.....	xiii
4.1 Bezkontaktní čtečka.....	xiii
4.2 Elektrický otvírač dveří.....	xiv
4.3 Připojení dveřních elektronik k řídicímu počítači.....	xiv
4.4 Dveřní elektronika.....	xv
4.4.1 Napájecí obvod.....	xvi
4.4.2 Ovládání elektrického zámku.....	xvi
4.4.3 Připojení převodníku linky RS232/TTL.....	xvii
4.4.4 Připojení převodníku linky RS485/TTL.....	xviii
4.4.5 Signály BEEP, LED.....	xix
4.4.6 Mikroprocesor.....	xix
4.5 Návrh DPS.....	xx
4.6 Řídicí počítač.....	xxi
4.7 Interakční diagram.....	xxii
4.8 Komunikační protokol.....	xxiv
4.8.1 Význam instrukcí.....	xxv
4.9 Firmware dveřních elektronik.....	xxix
4.9.1 Základní struktura firmwaru.....	xxx
4.9.2 Inicializace.....	xxx
4.9.2.1 Konfigurace režimu práce mikroprocesoru.....	xxxii
4.9.2.2 Vypnutí nadbytečných částí hardware.....	xxxii
4.9.2.3 Nastavení priorit pro přerušení.....	xxxii
4.9.2.4 Nastavení interního časovače.....	xxxii
4.9.2.5 Konfigurace softwarového UART.....	xxxii
4.9.2.6 Konfigurace a inicializace LED a BEEP signálů u čtečky.....	xxxii
4.9.2.7 Konfigurace hardwarového EUSART.....	xxxii
4.9.3 Obsluha přerušení.....	xxxiii
4.9.3.1 Detekce přechodu high-low.....	xxxiii
4.9.3.2 Příjem dat na EUSART.....	xxxiii
4.9.3.3 Interní časovač Timer2.....	xxxiv
4.9.3.4 Hlavní cyklus.....	xxxv
4.9.3.5 Testování příznaku reset.....	xxxvi
4.9.3.6 Test RS232.....	xxxvi
4.9.3.7 Test RS485.....	xxxvii
4.9.4 Upload firmwaru a debugging.....	xxxvii
5 Ovládací software řídicího počítače.....	xxxix
5.1 Obsluha sériového portu.....	xxxix
5.1.1 Vývojový diagram.....	xl

5.1.2	Formát IPC zprávy.....	xlii
5.2	Autorizační program.....	xlii
5.2.1	Struktura aplikace.....	xlii
5.2.2	Načtení konfigurace a inicializace.....	xliv
5.2.2.1	Registrace shutdown funkce.....	xliv
5.2.2.2	Nastavení obsluhy pro zpracování signálů.....	xlv
5.2.2.3	Inicializace autorizačních tříd.....	xlv
5.2.2.4	Inicializace observerů.....	xlv
5.2.3	Obsluha hlavní smyčky.....	xlv
5.2.3.1	Autorizační mechanismus.....	xlvi
5.3	Testovací rozhraní.....	xlvii
6	Uvedení systému do provozu.....	xlviii
6.1	Instalace operačního systému.....	xlviii

Seznam ilustrací

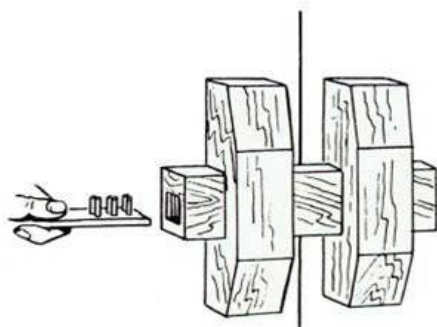
Obrázek 1: Egyptské závory.....	v
Obrázek 2: Různé provedení tagů.....	vi
Obrázek 3: Zámek s integrovanou čtečkou otisku prstu.....	vii
Obrázek 4: Schéma zapojení komponent systému.....	ix
Obrázek 5: Schéma transponderu.....	xi
Obrázek 6: Blokové schéma karty MIFARE.....	xii
Obrázek 7: MIFARE - umístění sériového čísla.....	xii
Obrázek 8: Čtečka Proximity MF7.....	xiii
Obrázek 9: Asynchronní přenos - struktura znaku.....	xiv
Obrázek 10: Elektrický zámek.....	xiv
Obrázek 11: RS485 zapojení prvků.....	xv
Obrázek 12: Blokové schéma dveřní elektroniky.....	xv
Obrázek 13: Napájecí obvod.....	xvi
Obrázek 14: Schéma ovládní relé.....	xvi
Obrázek 15: Schéma zapojení převodníku RS232/TTL.....	xvii
Obrázek 16: Záznam s osciloskopu.....	xviii
Obrázek 17: Implementace datového výstupu MF7.....	xviii
Obrázek 18: Schéma převodníku RS485/TTL.....	xix
Obrázek 19: Schéma zapojení mikroprocesoru.....	xx
Obrázek 20: DPS dveřní elektroniky.....	xxi
Obrázek 21: Rozmístění komponent v ZD3000 skříni.....	xxii
Obrázek 22: Interakční diagram.....	xxiii
Obrázek 23: Struktura programu mikroprocesoru.....	xxx
Obrázek 24: Detekce náběžné hrany.....	xxxiii
Obrázek 25: Příjem znaku na hw UART.....	xxxiv
Obrázek 26: Obsluha časovače Timer2.....	xxxv
Obrázek 27: Hlavní smyčka programu pro mikroprocesor.....	xxxvi
Obrázek 28: Test přijatých dat z RS232.....	xxxvii
Obrázek 29: MP-LAB vývojové prostředí.....	xxxviii
Obrázek 30: Schéma komunikace modulů.....	xxxix
Obrázek 31: Rozhraní Serial_Communication.....	xl
Obrázek 32: Rozhraní Ipc_Message.....	xl
Obrázek 33: Struktura obslužného programu pro sériový port.....	xli
Obrázek 34: Vývojové prostředí Zend Studio.....	xliii
Obrázek 35: Struktura autorizační aplikace.....	xliv
Obrázek 36: Příjem zprávy.....	xlvi
Obrázek 37: Struktura tabulky autorizační databáze.....	xlvi

Seznam tabulek

Tabulka 1: Porovnání RFID technologií.....	10
Tabulka 2: Převodní tabulka ASCII HEX formátu.....	13
Tabulka 3: Popis instrukcí protokolu.....	25
Tabulka 4: Popis odpovědí protokolu.....	25

1 Analýza jednoduchých přístupových systémů

Potřeby ochránit svůj majetek, respektive dovolit vstoupit do objektu pouze důvěryhodným osobám, sahají až do pravěku. Nejjednodušším způsobem bylo uschování majetku na neobvyklé a neznámé místo. S postupem času, kdy se společnost vyvíjela a lidé začínají rozvíjet nejen o své fyzické potřeby, se tato metoda stává neúčinná, vzhledem k snadnému zapomenutí místa úschovy a také lidé přestávají žít kočovný život a dochází k budování stálých příbytků, které je potřeba zabezpečit proti nezvaným hostům. Způsob realizace závisel na společenském postavení a technologických znalostech. V Indii se používalo ke střežení pokladů krokodýlů a pouze osoba znalá omamných látek měla po chvíli volnou cestu. Staří Řekové při odchodu z domu vázali na dveře složité uzly a pouze autor je mohl rozvázat v rozumné době. Čistě mechanickému provedení používali Egypťané, tzv. egyptské dřevěné závory.



Obrázek 1: Egyptské závory

Po zasunutí správného klíče se nadzvedly západky do horní úrovně a závora se mohla vytáhnout. Tento princip se zachoval i do současnosti a dnes je znám s různými obměnami jako cylindrická vložka.

Po staletí tento princip přístupového systému vyhovoval, neboť neexistovala alternativa, která by nevyžadovala složité a často nespolehlivé mechanické provedení. Hlavní problém spočívá v nutnosti mít fyzicky u sebe příslušný klíč. Při jeho ztrátě je následně nutné vyměnit z bezpečnostních důvodů všechny zámky, do kterých pasoval. Zároveň evidence uživatelů v případě velkých objektů není nejjednodušší. Na trhu existují produkty s generálním klíčem, který pasuje do více zámků a navíc má každý zámek svůj unikátní klíč. Ač se snaží tyto systémy ulehčit správu, platí u nich všechny nevýhody, které byly zmíněny dříve.

V současné době se problémy spojené se správou a údržbou systémů snaží řešit elektronické zámky. Hlavní myšlenka ovšem zůstává stejná. Při úspěšné autorizaci, buďto mechanickým klíčem, či jiným zařízením, je povolen přístup.

Z hlediska správy lze přístupové systémy rozdělit na

- Centrálně řízené
- Autonomní
- Kombinované

Centrálně ovládané zámky jsou vhodné zejména ve velkých organizacích nebo při vysokém počtu nasazení. Řízená správa umožňuje rychle a efektivně reagovat na změny, např. při vysoké fluktuaci zaměstnanců. Rovněž, lze při ztrátě autorizačního klíče snadno zablokovat přístup, pouhým odstraněním z centrálně uložené evidence. Nevýhody tohoto systému je technická složitost zařízení zajišťující ovládnání a ověřování, údržba komunikačních linek a zajištění školené obsluhy. Vyšší pořizovací cenu lze vykompenzovat integrací do firemního účetnictví, kdy lze sledovat docházku pracovníků a následně automaticky zpracovat mzdové podklady.

Autonomně řízené zámky fungují zcela nezávisle a ke svému provozu potřebují pouze energetický zdroj. Jsou vhodné v prostředí, kde se neočekává velké nasazení. Výhoda oproti klasickým zámekům se projeví v případě ztráty klíče. Elektronický zámek není třeba vyměnit, pouze se vymaže záznam z vnitřní paměti. Snaha výrobců je stavět tato zařízení s minimálními náklady, a proto jsou většinou omezeny pouze pro několik desítek klíčů v závislosti na volbě technologie autorizace.

Kombinované zámky sdružují výhody předchozích řešení a odstraňují jejich technické nedostatky. Centrálně řízené zámky se stávají v případě výpadku komunikace nefunkčními a autonomní zámky jsou omezeny kapacitně. Kombinované zámky se primárně chovají stejně jako centrálně řízené, pouze v případě výpadku komunikace, či jiné neočekávané situace se přepnou do nezávislého režimu. Zde je uložena sada pravidel pro nouzový režim a jejich používání záleží na specifických podmínkách uživatelů. Např. lze stanovit použití autorizačních klíčů, které byly použity nejčastěji nebo lze definovat sadu záložních klíčů.

Při volbě elektronického zámku lze vybírat z několika různých technologií. Každá má své specifika a její nasazení záleží čistě na preferencích uživatele. Zámky lze rozdělit podle několika kritérií

Vyžadující externí klíč

- kontaktní magnetické karty
- bezkontaktní čipové karty
- rádiové a IR vysílače

Klíč je součástí nositele

- biometrické
 - snímače otisku prstu
 - snímače oční duhovky
- číslíkové kódové

Elektronické zámky vyžadující externí klíč nepřinášejí až tak uživatelský komfort v lidské zapomnětlivosti, ale zahrnují v sobě výhody jednodušší správy klíčů a bezpečnosti, která byla u mechanickým zámeků historicky překonávána. Design klíčů se liší od požadavků uživatele a stejně jako Římští občané, kteří nosily tógy bez kapes, mohou být navrhnuty jako okrasný šperk.

Rozsah aplikací, kde lze použít klíče je neomezený. Dalo by se říci, že slovo klíč ve významu odemykání zámeků ztrácí na významu. Například bezkontaktní čipové karty lze využít i jako elektronickou peněženku při platbě v obchodě, nálepky s integrovaným obvodem a pasivním

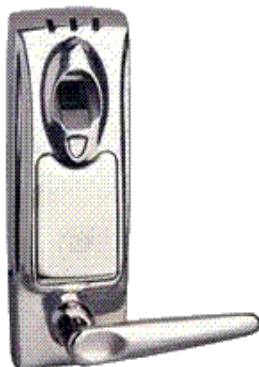


Obrázek 2: Různá provedení tagů

napájením jsou využívány k ochraně a snadné evidenci zboží.

Bezklíčové systémy obvykle používají jedinečnou vlastnost osoby, která se autorizuje, popřípadě se

využívá paměťového efektu. Oproti předchozímu použití zde odpadá nutnost nosit klíče stále při sobě, neboť jsou často nedílnou součástí uživatelů a ztráta nosiče klíče bývá bolestivá. Historicky je nejpoužívanější číselný kód. Uživatel se autorizuje vyťukáním čísel na klávesnici, případně hesla. Jde o poměrně jednoduchý systém, ovšem spoléhající se na lidskou paměť. Naproti tomu biometrické systémy, pomineme-li problémy s ochranou soukromí, využívají unikátní lidské znaky, které jsou neměnné od narození. Na trhu jsou dostupné dva produkty. Snímač oční duhovky a otisku prstu. S rostoucí výkonností výpočetní techniky a mikroprocesorových čipů se biometrické systémy přesouvají ze sci-fi filmů a tajných vládních agentur do komerčního prostředí. Autorizace pomocí otisku prstů našla uplatnění nejenom v přístupových systémech. Snímače lze například nalézt na novějších notebookech, kde nahrazují vstupní hesla.



Obrázek 3: Zámek s integrovanou čtečkou otisku prstu

2 Specifikace systému

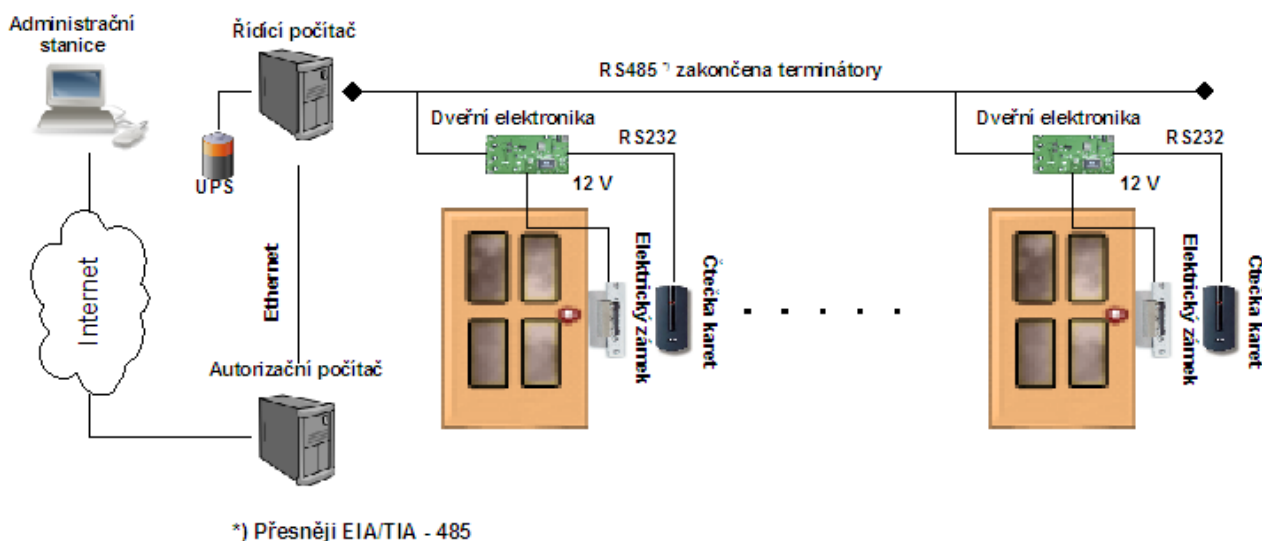
Při návrhu přístupového systému jsem musel vycházet ze současné situace a jelikož hlavním cílem práce je vytvořit nenákladnou alternativu komerčních přístupových systémů, museli být při rozhodování o výběru technologie vzaty z ekonomického hlediska tato omezení:

- Čtečky bezkontaktních karet standardu MIFARE,
 - každý student ČVUT a i jiných Vysokých škol (VŠE, UK) obdrží při zápisu svoji kartu, kterou má dispozici během studia. Není nutné investovat finanční prostředky na nákup nových karet.
- Řídící a autorizační počítač vycházejí ze specifikace IBM PC,
 - s miniaturizací součástek je dnes možné sehnat kompletně vybavený osobní počítač o velikosti 17 x 17 cm. Cena zařízení se pohybuje vzhledem k vysoké produkci komponent do 2000 Kč. Na trhu se vyskytují jednoúčelové desky, ale cena se pohybuje o řád více než klasická PC.
- Ovládací software využívající open-source produktů,
- Dveřní elektronika vybavená jednočipovým mikroprocesorem,
 - jedno z řešení přístupového systému je napojení čteček pomocí převodníků rozhraní na řídicí počítač. V případě výpadku počítače, popř. spojení je však systém nefunkční, jelikož čtečky postrádají logiku a cena převodníku je mnohonásobně vyšší, než mikroprocesorové řešení.
- Sdílená sběrníková topologie pro dveřní elektroniky,
 - umožňuje připojování dveřních elektronik na stávající systém bez nadměrné instalace kabeláže. Každá elektronika zároveň může sledovat stav na lince a dle toho reagovat na situaci.

Z hlediska správy bude přístupový systém kombinovaný. Každý elektrický zámek bude mít definovanou sadu klíčů pro případ výpadku komunikace s řídicím počítačem. Tyto klíče bude možno zadávat z řídicího počítače.

3 Popis komponent systému

Schéma propojení komponent je zobrazeno na obr. 4.



Obrázek 4: Schéma zapojení komponent systému

- **Administrativní stanice**
Pomocí této stanice se definují povolené klíče a časové zóny vstupů. Uživatel se připojí programem s grafickým rozhraním k autorizačnímu počítači, jehož pomocí nastavuje potřebné hodnoty.
- **Autorizační počítač**
Tato stanice udržuje informace o klíčích a časových zónách. Údaje jsou uloženy v databázi a lze je měnit po přihlášení z administrativní stanice.
- **Řídicí počítač**
Jedná se o nejdůležitější součást systému. Tento počítač zajišťuje komunikaci mezi autorizačním počítačem a dveřními elektronikami. Zároveň napájí elektrické zámky stejnosměrným napětím.
- **Dveřní elektronika**
Tato deska plošných spojů osazená mikroprocesorem posílá na řídicí počítači klíče, které získá ze čtečky bezkontaktních karet. Po úspěšné autorizaci ovládá připojený elektrický zámek.
- **Čtečka karet**
Čtečka bezkontaktních karet MIFARE po přiložení karty přečte klíč uložený na kartě a odešle ho do dveřní elektroniky.
- **Elektrický zámek**
Zámek po připojení stejnosměrného napětí vybudí v cívce magnetické pole, které přitáhne kotvu a umožní se tak otevření dveří. Zámek je ovládán dveřní elektronikou.

3.1 RFID

V současné době došlo k velkému nárůstu používání bezdrátových technologií. Přechod od stacionárních k mobilním zařízením je podpořen miniaturizací komponent a rozvojem mikropočítačových zařízení, které svou výkonností splňují vysoké požadavky kladené na bezpečnost, ale zároveň na snadnou implementaci. Zároveň s rozvojem hardwaru, vzniká zde prostor pro aplikace, které byly dříve omezeny nedostatečným technologickým vývojem. S rostoucí oblibou GSM mobilních zařízení vyvstává otázka univerzálního zařízení, které by dokázalo splnit veškeré požadavky na finanční, tak i bezpečnostní aplikace.

Na trhu s RFID existují de facto dva standardy.

1) NF

Nízko frekvenční RFID jsou typicky definovány rádiovou frekvencí transponderu (viz níže) pracujícího v rozsahu 100 KHz až 140 KHz. Nejčastěji jde o frekvenci 125 KHz. LF produkty jsou méně citlivé na blízké kovy než zařízení operující na vyšších frekvencích. Na trhu se již vyskytují řadu let a v současné době se nejeví, že by mohli být nahrazeny VF systémy, pro svou rozšířenost a robustnost.

2) VF

Vysoko frekvenční RFID dnes pracují na frekvenci 13,56 MHz. Hlavními producenty jsou Philips Semiconductor (MIFARE) a Texas Instruments (Tag-it). Oba systémy byly donedávna zcela nekompatibilní a až v roce 2000 došlo k vytvoření standardů ISO 15693 „Vicinity” a ISO 14443 „Proximity”. VF RFID pracují na principu magnetic wave coupling principles, při němž je časově proměnný proud transformován na magnetické pole. V transponderu, vyskytující se v tomto poli, je indukován proud, který napájí čip.

Následující tabulka porovnává vlastnosti jednotlivých technologií

	Technologie	
	NF	VF
Čtecí vzdálenost	Do 50 cm	Do 10 cm
Odolnost proti rušení	vyšší	nižší
Rychlost přenosu dat		
Antikolizní mechanismus	není	implementován
Bezpečnost	nízká	vysoká
Náklady na transponder	nižší	vyšší

Tabulka 1: Porovnání RFID technologií

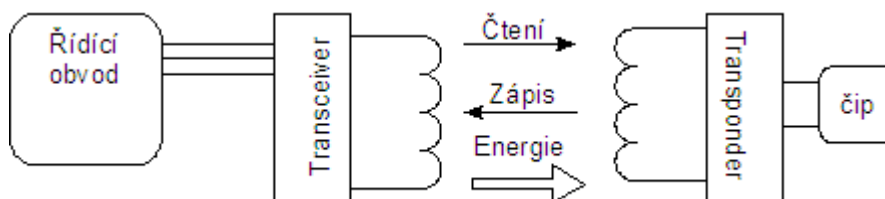
Porovnáním jednotlivých technologií a požadavků na autorizační mechanismus, vychází NF transpondery jako optimální řešení. Nastává zde ale kolize s cílovou skupinou uživatelů, kteří jsou již vybaveni VF transpondery. Nákupem odlišné technologie by se nepřiměřeně zvýšila cena implementace.

Základní RFID systém se skládá ze tří komponent

- antény
- vysílač/přijímač (transceiver)
- transponder (RF tag)

Oba systémy v případě bezkontaktních karet fungují na stejném principu, liší se pouze užitnými vlastnostmi vyplývající z fyzikální podstaty přenosu informací.

Transceiver vysílá radiový signál, který aktivuje tag a následně lze z něj číst, popř. zapisovat data.



Obrázek 5: Schéma transponderu

Obvykle je anténa spojena s transceiverem a dekodérem. Takovéto zařízení má název bezkontaktní čtečka RF tagů. Čtečka neustále vysílá radiový signál a pokud se objeví tag v zóně jejího dosahu, dekodují se data z čipu a jsou následně poslána k následnému zpracování.

Pasivní RFID tagy pracují bez externího zdroje napájení a získávají energii generovanou čtečkou. Pasivní tagy jsou následkem toho lehčí než aktivní, levnější a nabízejí skoro neomezenou životnost. Nevýhoda spočívá v nižší čtecí vzdálenosti a obvykle bývají read-only s výrobcem předdefinovanými daty.

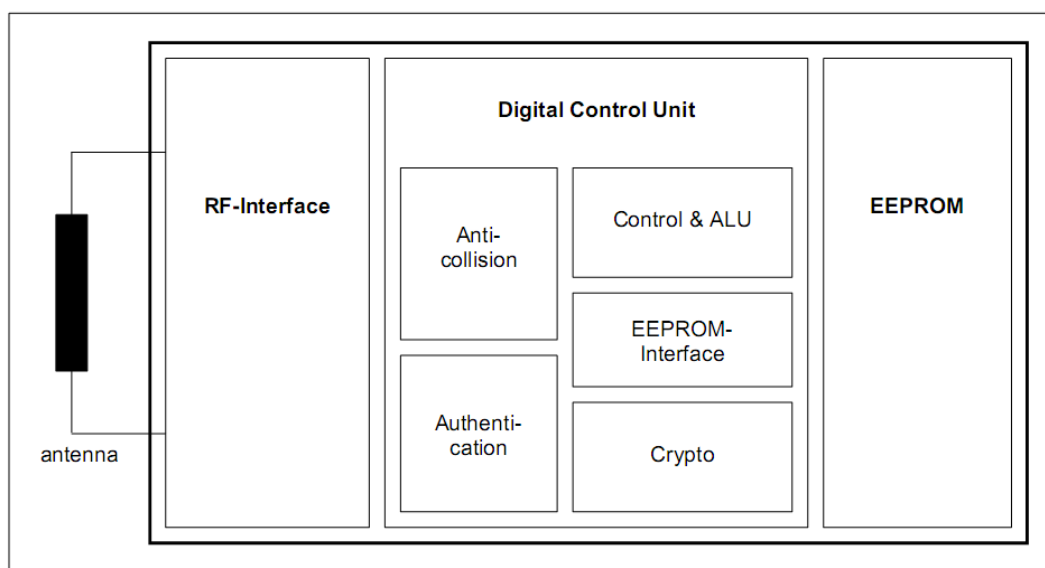
Hlavní výhodou všech RFID systémů je bezkontaktnost mezi piny tagu a čtecím zařízením, což se projeví zvýšenou životností karet a není třeba přímá viditelnost mezi čtečkou, jak je tomu u čárových kódů nebo magnetických karet. Naproti tomu, vyšší cena implementace tagů brání většímu rozšíření na trhu. Ale tato nevýhoda by se měla v brzké době odstranit. V laboratořích se testují postupy, kdy bude možné tag vytisknout pomocí speciální náplně, podobně, jako čárové kódy.

3.1.1 Technologie MIFARE

Bezkontaktní karty typu MIFARE 1 S50 jsou speciálně designovány, aby splnily požadavky na zabezpečení platebních transakcí. Mohou být použity ve veřejné dopravě místo klasických jízdenek nebo jako elektronická peněženka. MIFARE® karty mají typicky velikost kreditní karty a nemusejí být vyndány z peněženky během transakce, dokonce ani v případě, že jsou zde přítomny mince.

Mezi hlavní bezpečnostní prvky patří:

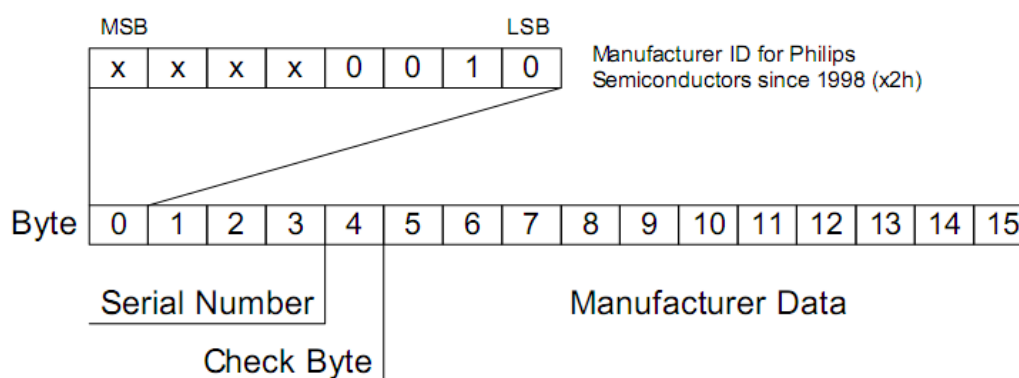
- Proměnlivá tři průchodová autentizace dle standardu ISO/IEC 15706-2
- Šifrování bezdrátové komunikace s ochranou proti zpětnému opakování
- Individuální klíče na sektory, lze tak podporovat více aplikací
- Unikátní sériové číslo nastavené výrobcem karty
- Antikolizní mechanismus - schopnost ovládat více karet vyskytujících se v aktivní zóně čtečky
- Transportní klíč



Obrázek 6: Blokové schéma karty MIFARE

Technická data:

- Frekvence - 13.56MHz
- Komunikační rychlost až 106 KBaudů
- Antikolizní mechanismus: Možnost obsloužit více karet nacházející se současně ve čtecím poli
- Operační vzdálenost až do 10 cm
- 8 kBit EEPROM paměť
- Uchování dat 10 let.
- Zapisovací četnost 100.000 cyklů



Obrázek 7: MIFARE - umístění sériového čísla

Kompletní popis vlastností karty je přiložen na doprovodném CD. Vzhledem k omezením uvedená v kapitole 3. *Specifikace systému* je pro autorizaci uživatele použito pouze sériové číslo karty. Tento způsob autorizace je v současné době nevyhovující. S patřičným technickým zázemím a vybavením lze vyrobit kartu s libovolným sériovým číslem.

4 Implementace hardware

4.1 Bezkontaktní čtečka

Bezkontaktní čtečka Proximity MF7 je připojena k dveřní elektronice sériovou linkou RS-232C. Po přiložení čipové karty je přečteno její sériové číslo a asynchronně odesláno po lince.



Obrázek 8: Čtečka Proximity MF7

Sériové číslo, alias klíč, je tvořen osmi byty zakódovanými v ASCII HEX formátu. Byte je rozdělen na horní a dolní 4 bitová slova, která se převedou na odpovídající vyjádření čísel a znaků. Režie přenosu navýšená o 100% je kompenzována snadným propojením na terminály, které umí zobrazovat pouze ASCII znaky.

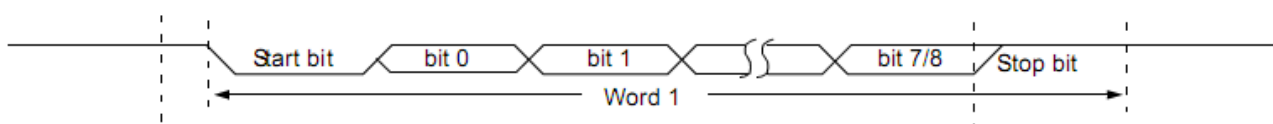
4 bitové slovo				HEX	ASCII
0	0	0	0	30	0
0	0	0	1	31	1
0	0	1	0	32	2
0	0	1	1	33	3
0	1	0	0	34	4
0	1	0	1	35	5
0	1	1	0	36	6
0	1	1	1	37	7
1	0	0	0	38	8
1	0	0	1	39	9
1	0	1	0	41	A
1	0	1	1	42	B
1	1	0	0	43	C
1	1	0	1	44	D
1	1	1	0	45	E
1	1	1	1	46	F

Tabulka 2: Převodní tabulka ASCII HEX formátu

Rozhraní RS-232C bylo původně určeno pro komunikaci mezi dvěma zařízeními až na vzdálenost 15 m. Pro větší odolnost proti rušení se pro přenos používá vyšších napětí než TTL logika a přenos informací může probíhat synchronně nebo asynchronně. Toto rozhraní mělo být v současnosti nahrazeno USB, ale pro svou jednoduchost implementace se hojně využívá nejen v embedded zařízeních, ale i v moderních PC.

Standard definuje asymetrické vysílače a přijímače, které přenášejí signál relativně ke společnému vodiči (GND). U rozhraní není implementováno galvanické oddělení propojovaných zařízení, proto je důležité dbát na zemnicí smyčky. Napětí v rozsahu -12 V až -3 V odpovídá logické jedničce a napětí od +3 V do +12 V logické nule. Bližší popis standardu je umístěn na internetové adrese [1].

Při asynchronním přenosu musí být na straně vysílače i přijímače definovány stejné parametry přenosu. Čtečka MF7 je nastavena na 9600 baud, 8 bitů na paket, bez využití parity a 1 stop bit.



Obrázek 9: Asynchronní přenos - struktura znaku

Čtečka obsahuje navíc dva kontrolní vodiče, externí bzučák a LED. Propojením specifického vodiče na GND se aktivuje jeho funkce.

4.2 Elektrický otvírač dveří

Elektrický zámek po připojení stejnosměrného napětí vygeneruje v cívce magnetické pole, které přitáhne ocelovou kotvu držící mechanický blokováč. Konstrukce zámku umožňuje blokovat otevření dveří bez přiloženého napětí. Další varianta, bezpečnostní, pracuje na opačném principu. Pokud je napětí přerušeno zámek se přepne automaticky do stavu odblokováno. Tento typ se používá u nouzových východů a bývá napojen na EPS.



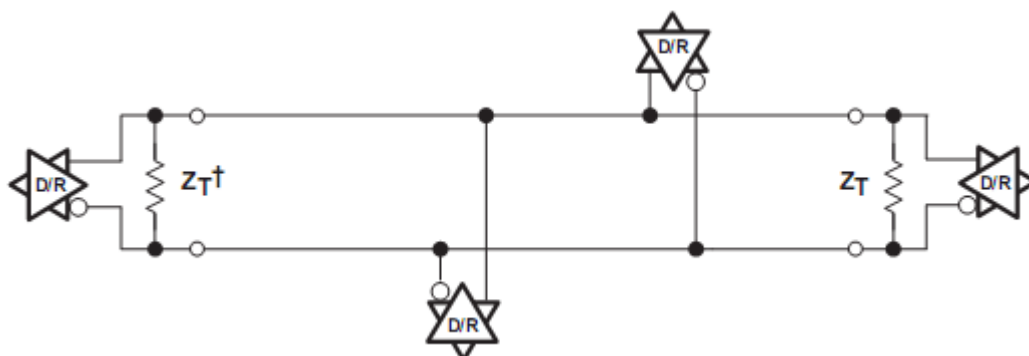
Obrázek 10: Elektrický zámek

Důležitým parametrem při výběru zámku je proud procházející cívkou při sepnutí. U rozvodů s centrálním napájením, kdy kabeláž vede stovky metrů dochází příliš k velkým ztrátám vedení. Výborné vlastnosti má zámek BEFO Profi od společnosti Bera, s.r.o. Výrobce deklaruje proud 220-230 mA při 12 V.

4.3 Připojení dveřních elektronik k řídicímu počítači

Sběrnice RS485 nabízí robustní řešení pro přenos dat na dlouhé vzdálenosti i v silně zarušeném prostředí. Na rozdíl od RS232, logická úroveň se zjišťuje diferenciálním vyhodnocením napětí mezi dvěma vodiči. Toto řešení umožňuje komunikovat až na vzdálenost 1200 m a rychlosti 10 Mbit/s na stíněné kroucené dvojlinky zakončené impedančním přizpůsobením.

Norma umožňuje připojit na společnou sběrnici (společná pro příjem i vysílání dat) až 32 zařízení. Vedení musí být na obou koncích impedančně přizpůsobeno, aby nedocházelo k odrazům. Typická hodnota rezistorů používaných jako terminátory je 120 Ohm. Zároveň se doporučuje definovat klidovou úroveň napětí, v případech, kdy žádné zařízení nevysílá. Každé zařízení se může stát kdykoliv řídicím (MASTER). Aby nedocházelo ke kolizi dat na lince, musí být v každém zařízení použito řízení ovládání přístupu na společnou komunikační linku RS485.



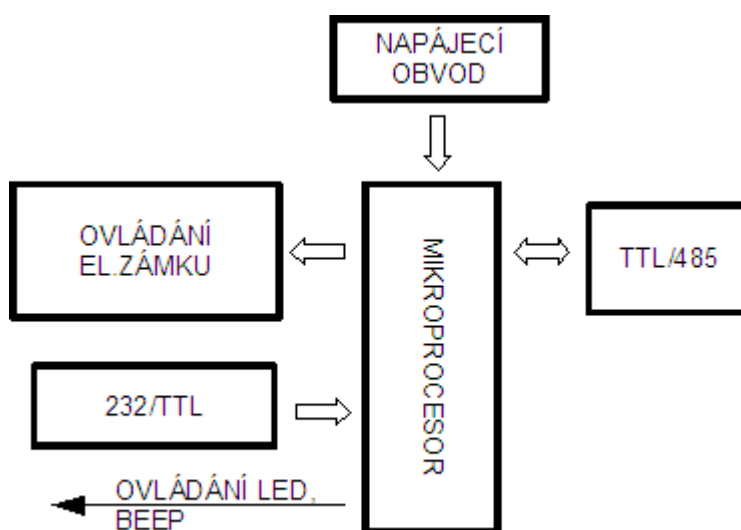
Obrázek 11: RS485 zapojení prvků

V praxi se obvykle nevyužívá velkých přenosových rychlostí, a tak může být topologie odlišná od doporučeného standardu, například je možno při rychlosti 9600 Baudů použít zapojení do hvězdy a dokonce nezapojit impedanční přizpůsobení. Přesto bude komunikace probíhat spolehlivě.

Bližší popis standardu je umístěn na internetové adrese [2].

4.4 Dveřní elektronika

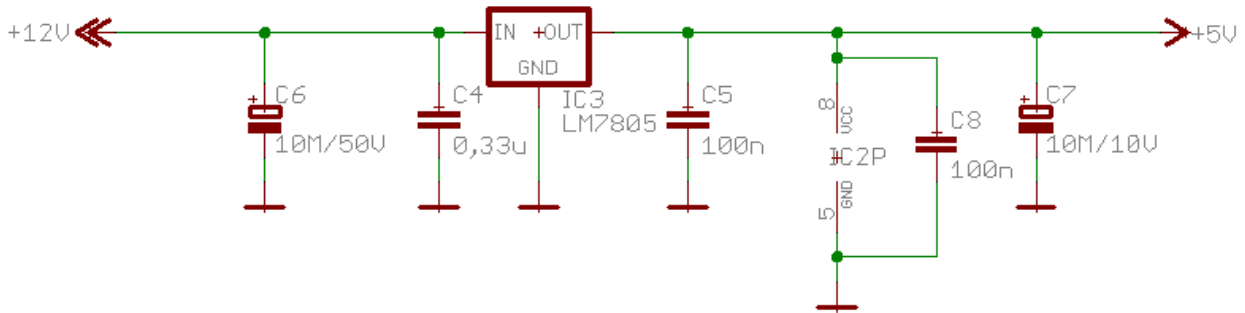
Zjednodušené schéma je zobrazeno na obr. 12. Základním blokem je mikroprocesor, který komunikuje s perifériemi a ovládá je podle pokynů řídicího počítače.



Obrázek 12: Blokové schéma dveřní elektroniky

4.4.1 Napájecí obvod

Elektronika je napájena z přívodu stejnosměrného napětí 12V, které zároveň napájí i elektrický zámek. Vstup je chráněn proti zkratu tavnou pojistkou dimenzovanou na maximální proud 1,6A. Ochranu proti přepólování elektronických součástek zajišťuje v sérii zapojená dioda 1N4005. Všechny součástky, použité v návrhu vyžadují stabilizované stejnosměrné napětí 5 V, zajištěné stabilizátorem L78M05. Zapojené vyhlazovací kondenzátory odpovídají doporučeným hodnotám z datasheetu součástky.

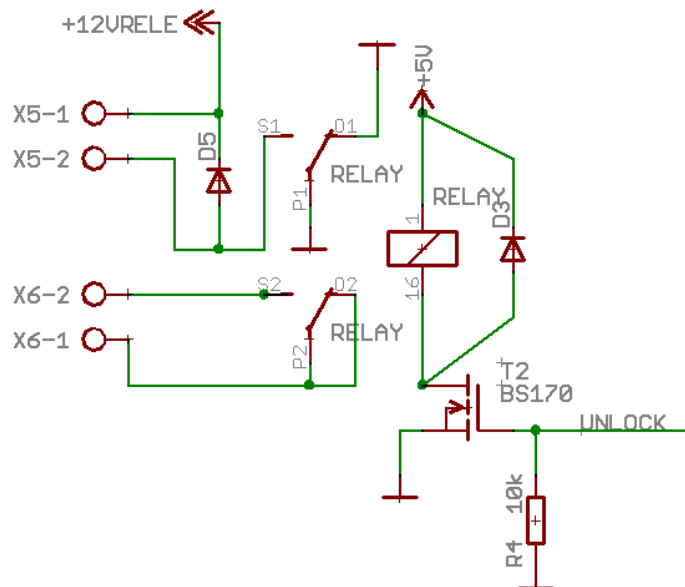


Obrázek 13: Napájecí obvod

Správná funkce napájení je signalizována diodou.

4.4.2 Ovládání elektrického zámku

Napájení pro zámek bylo původně zajišťováno výkonovým spínacím tranzistorem. Toto řešení odstraňuje nutnost použití mechanických součástek, ale má některé nevýhody. Není zajištěno galvanické oddělení na vstupu k mikroprocesoru a nelze provést integraci do systémů, kde je zámek již napájen vlastním zdrojem. Z těchto důvodů byla zvolena varianta nahradit tranzistor dvou kontaktním přepínacím relé. Spínání relé zajišťuje N-kanálový MOSFET tranzistor, jehož gate je spojen s výstupem mikroprocesoru.

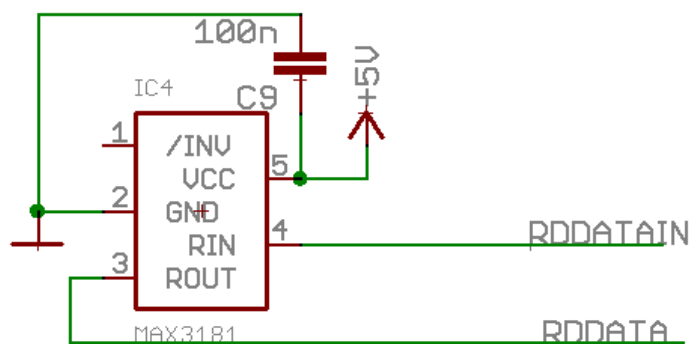


Obrázek 14: Schéma ovládání relé

Cívka relé i vývody k elektrickému zámku jsou opatřeny antiparalelně zapojenou diodou, která u tranzistoru zabrání výskytu zvýšeného napětí, jenž vznikne při vypnutí proudu do cívky a u relé jiskření na kontaktech.

4.4.3 Připojení převodníku linky RS232/TTL

Vstupy mikroprocesory jsou navrženy na 5V TTL logiku, proto je nutné použít převodník rozhraní. Běžně dostupné převodníky obsahují více vstupních linek pro převod a zároveň opačný převod TTL/RS232, který vyžaduje pro svou funkci připojení vysokokapacitních kondenzátorů ve funkci nábojové pumpy. Naštěstí firma MAXIM vyrábí řadu převodníků (MAX3181), které obsahují pouze jeden vstup. Nevýhoda této součástky je její nedostupnost na českém trhu v menším množství než 1000 kusů. Tato nevýhoda je kompenzována možností nechat si zaslat od firmy MAXIM vzorky až do 30 kusů na osobu.

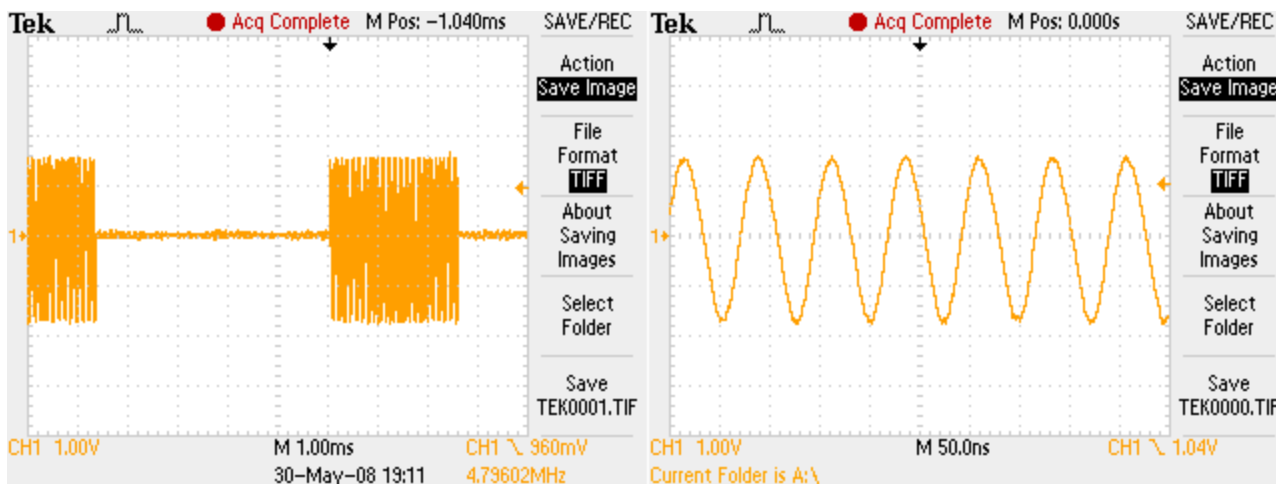


Obrázek 15: Schéma zapojení převodníku RS232/TTL

K napájecím vývodům je připojen blokovací 100nF keramický SMD kondenzátor.

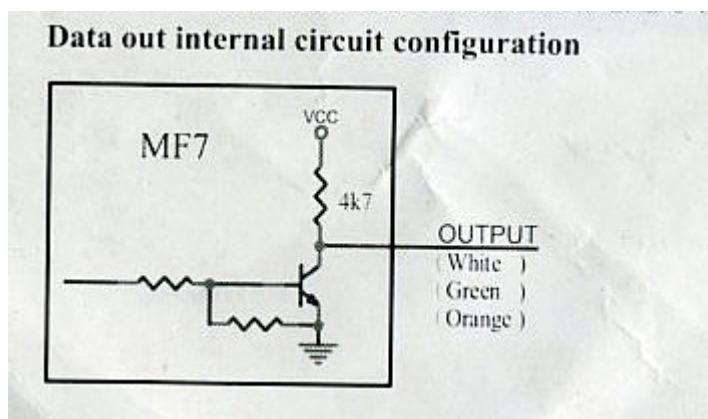
Analýzou signálu použité čtečky jsem zjistil, že ač se v technické specifikaci hovoří o RS-232 výstupu, ve skutečnosti je výstup TTL, kde logická 0 je 5V a 1 0V a jiné napěťové úrovně se zde nevyskytují. Použití převodníku by teoreticky nemuselo být třeba, ale zvolil jsem jeho použití z těchto důvodů

- ochrana vstupů proti elektrostatickému výboji do 15kV
- univerzálnost, lze připojit libovolnou čtečku s RS-232 výstupem
- vyhlazení náběžných hran signálu



Obrázek 16: Záznam s osciloskopu

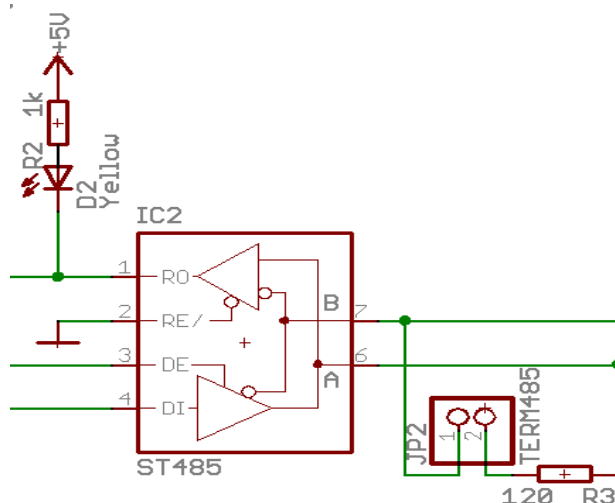
Při prvotních testech a bez použití převodníku docházelo při přímém připojení výstupu čtečky na vstup mikroprocesoru k samovolnému spouštění rutiny obsluhující sériový port. Kontrola softwaru, ani on-chip debug nástroje neobjevily příčinu spouštění. Až detailní pohled na datový výstup odhalil důvod neobvyklého chování. Každých 4,6 ms je generováno sinusové rušení o frekvenci 15,56 MHz a amplitudě 1,5V. Tato frekvence je shodná s frekvencí, kterou používají RFID tagy. S použitím převodníku se podařilo eliminovat náhodné spouštění rutiny. Dle technické specifikace má MAX3181 vstupní odpor 5k Ohm, zatímco čtečka používá výstup pull-up odporem 4,7k Ohm. Rušení je sníženo na tomto děliči na polovinu a již nezpůsobuje nestabilní chování, neboť se dostane mimo rozhodovací úroveň převodníku.



Obrázek 17: Implementace datového výstupu MF7

4.4.4 Připojení převodníku linky RS485/TTL

Podobně jako v případě zapojení převodníku pro RS232, je nutné použít speciální obvod zajišťující konverzi signálu. Převodník ST485bn od firmy STMicroelectronics splňuje veškeré požadavky kladené na funkčnost. Jako bonus umožňuje zdvojnásobit počet připojených zařízení na jedné lince oproti normě. ST485 umožňuje obousměrný přenos, který je řízen signály RE a DE.



Obrázek 18: Schéma převodníku RS485/TTL

V této aplikaci je signál \overline{RE} trvale nastaven na log 1. Veškerý přenos, i z vlastního výstupního obvodu, je směřován zpět na vstup. Tato implementace umožňuje zapojení v režimu multimaster s náhodným přístupem ke sběrnici. Kolize se detekují porovnáním vstupu/výstupu a je nasnadě využití zjednodušených algoritmů založených na CSMA/CD.

Příjem dat je indikován diodou, zapojenou na TTL výstup převodníku (pin RO). Ten je v klidovém stavu nastaven na log 0 (5V). Pokud dojde ke změně logické hodnoty (napěťové úrovně), začne diodou procházet proud. Jeho hodnota je snížena předřadným odporem, aby nedošlo k ovlivnění rozhodovací úrovně na vstupu mikroprocesoru. Terminační odpor 120 Ohm se zapojí jumperem. Klidový stav na lince definuje vždy rozhraní, které je napojené na řídicí počítač.

4.4.5 Signály BEEP, LED

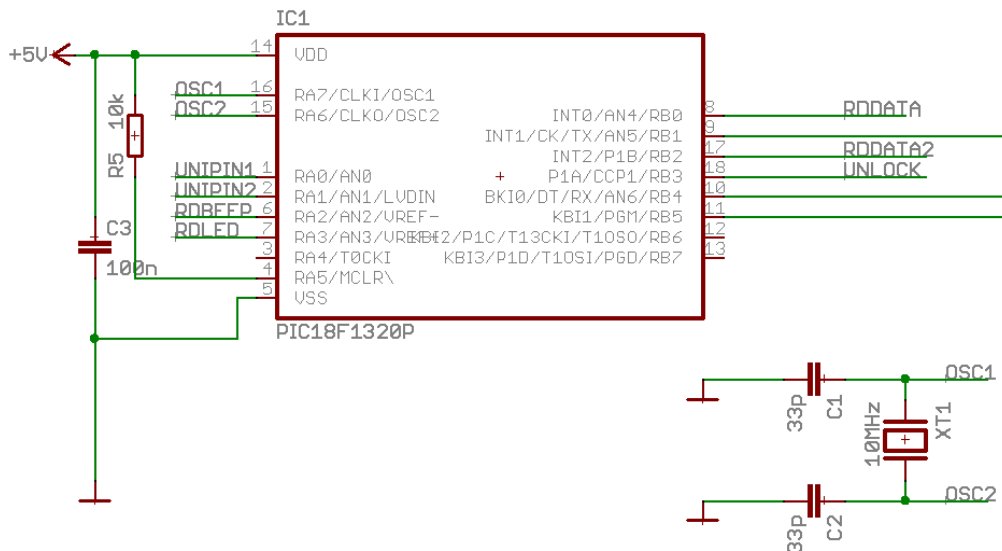
Čtečka MF7 obsahuje dva vodiče, kterými se ovládá integrovaný piezoelektrický bzučák (BEEP) a rozsvícení červené diody. Pokud je aktivován bzučák, čtečka nepřijímá identifikátory z karet. Této vlastnosti je využito při signalizaci otevření dveří. LED je aktivována v případě aktivovaného zámku nebo při selhání komunikace s řídicím počítačem. Vodiče jsou připojeny přímo na vstupy mikroprocesoru. Jejich aktivace se provádí nastavením nulového potenciálu na vstupech, resp. jsou vodiče spojené se zemí.

4.4.6 Mikroprocesor

Základem veškeré logiky na dveřní elektronice je mikroprocesor PIC18F1320 (DIP18) od firmy Microchip [3]. Jedná se o 8-bitový, nízkopříkonový procesor s bohatou výbavou integrovaných periférií.

Výběr právě tohoto typu byl založen na následujících vlastnostech:

- Vynikající vývojové prostředí MPLAB
- Kvalitní překladače jazyka C - Hi-tech
- Výuka v předmětech na ČVUT-FEL
- Zabudované debugovací nástroje přímo v procesoru
- Rozsáhlá softwarová podpora nejen od výrobce [4]
- Používání ve firmě IMA



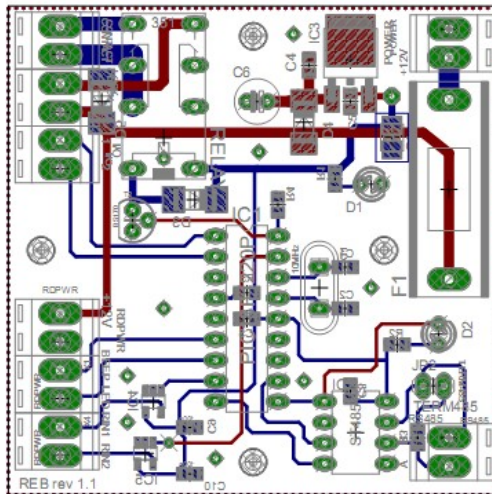
Obrázek 19: Schéma zapojení mikroprocesoru

Zapojení je provedeno podle doporučených hodnot výrobce. Čip je umístěn v patici (DIP18), lze ho jednoduše vyjmout, naprogramovat, nebo případně vyměnit poškozený kus.

Procesor nevyžaduje při připojení periférií žádné úpravy, pokud jsou TTL kompatibilní. Na určitých vstupních pinech je použit Schmitt trigger buffer, pomocí něhož je snadné implementovat a naprogramovat softwarový UART bez použití pollingu. Taktovací frekvence 10MHz je postačující pro chod aplikace.

4.5 Návrh DPS

Při návrhu desky plošných spojů je důležité dbát na doporučené postupy rozmisťování součástek a vedení spojů, viz publikace [5]. Návrh jsem provedl v programu Eagle od společnosti Cadsoft [6]. Tento software je velmi intuitivní a začínající návrhář je schopen během několika dnů naučit se základy ovládání a vytvářet vlastní knihovny součástek. Pro nekomerční účely je poskytován zdarma, s omezením na maximální velikost DPS 10x10 cm. Výrobu plošného spoje jsem přenechal odborné firmě Pragoboard [7], jelikož nabízí přímo plugin do programu Eagle na vytvoření požadovaných exportů. Výsledné desky jsou na profesionální úrovni za velmi přijatelnou cenu. Co považuji za nevýhodu, je příplatek za DPS menší než 1 dm². Naštěstí lze použít trik, že v jednom exportním souboru zkopírujeme vedle sebe, případně i nad, jednotlivé desky a ponecháme mezi nimi vrstvu DIMENSION. Společnost za takovýto soubor nepočítá příplatky za poddimenzování a navíc rozdělí desky podle zvolené vrstvy.



Obrázek 20: DPS dveřní elektroniky

4.6 Řídící počítač

Strategii při výběru technologie jsem se snažil volit tak, aby vynaložené úsilí bylo co nejmenší a zároveň vyvážené efektivitou provedení. Na současném trhu lze nalézt již plnohodnotné počítače za přiměřeně nízkou cenu. Integrace komponent je dostatečná a pro svůj provoz nepotřebují žádné přídatné periferie a výpočetní výkon je vhodný i pro běh moderních serverových operačních systémů.

Koncepci jsem se snažil pojmout více profesionálně. Umístěním počítače do ATX skříně svádí k použití nestandardních metod. Jako kryt jsem zvolil kovovou skříň ZD3000 od firmy ELSO. Svou velikostí postačuje pro počítač formátu mini-ITX a napájecí zdroj.

Základní desku tvoří Little Valley BLKD201GLYL s procesorem Intel Celeron 1.0 GHz. Na této desce je až na paměť vše integrováno, obsahuje až zbytečně periférií, např. audio. Tato „nevýhoda“ oproti účelovým deskám je vyvážena cenou, přibližně 1500 Kč v maloobchodním řetězci.

Operační systém jsem zvolil Ubuntu 8.04 server edition. Ačkoliv se nejedná o minidistribuci (vyžaduje cca 1GB volného prostoru), obsahuje velký počet předpřipravených aplikací a tím značně ulehčí problémy s kompilací programů. Operační systém je uložen na USB flash disku, neboť klasické rotační disky jsou z mechanického hlediska nespolehlivé. V nedávné době velmi výrazně klesla cena flash disků, 2GB lze sehnat do 500 Kč.

Připojení řídicího počítače ke sběrnici je realizováno USB – RS485/422 převodníkem od společnosti Papouch [8]. Po připojení do USB je operačním systémem automaticky vytvořeno zařízení „/dev/ttyUSB0“ s identickými vlastnostmi jako standardní sériový port (obdoba virtuálního COM portu z OS MS Windows). Přepínání směru toku dat je prováděno automaticky a bez prodlevy.

Na převodníku je řada přepínačů, které nastavují pracovní režim. Podrobnější informace se lze dočíst v technické dokumentaci k produktu. V této aplikaci je využito nastavení režimu RS485 a odporů definujících klidový stav na lince.

Pro snadnější připojení ke zdroji napětí a sběrnici byl navrhnut DPS. Obsahuje Molexový konektor pro připojení napájení elektrických zámků a dveřních elektronik, dva mikropsínače PWR a RESET a WAGO kompatibilní konektory pro připojení datového kabelu.



Obrázek 21: Rozmístění komponent v ZD3000 skříni

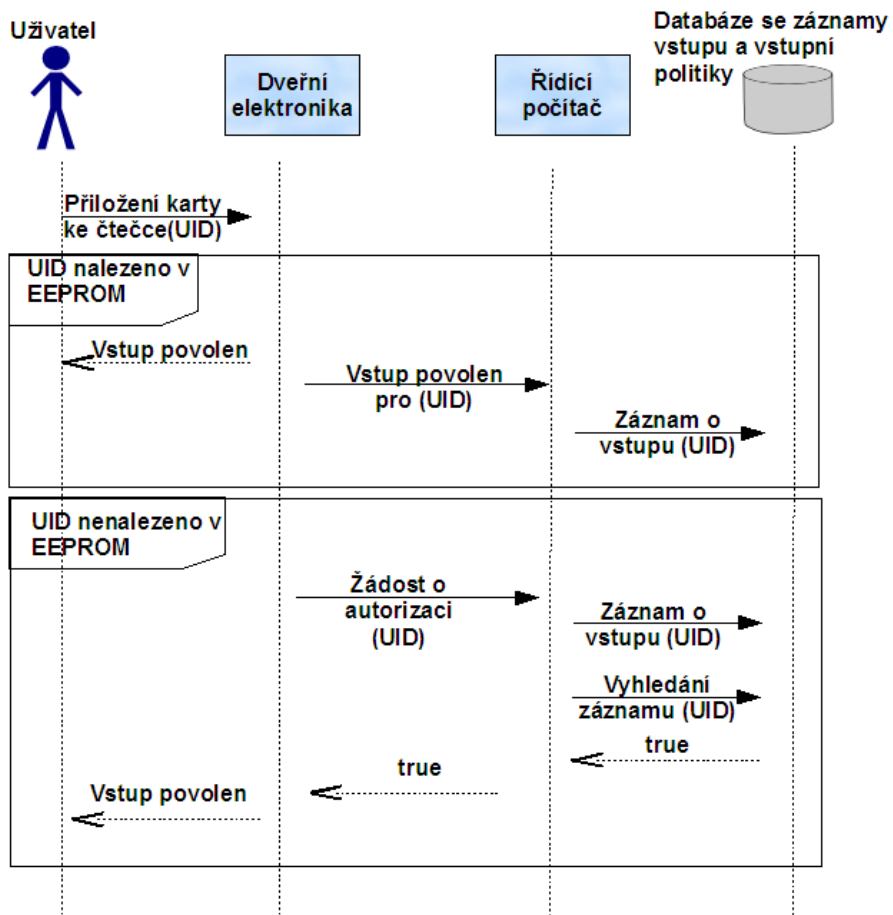
Implementace softwaru řídicího počítače

Návrh software je rozdělen do dvou skupin.

- Software pro mikroprocesor v dveřní elektronice
- Software pro řídicí počítač

4.7 Interakční diagram

Vzájemný vztah mezi oběma částmi znázorňuje interakční diagram. Komunikace mezi částmi je realizována protokolem.



Obrázek 22: Interakční diagram

4.8 Komunikační protokol

Komunikační protokol mezi dveřní elektronikou a řídicím počítačem původně vychází z protokolu Spinel používaný firmou Papouch ve svých produktech. V průběhu vývoje protokolu došlo ke změnám, které postupně vedli k odchýlení se od předpisu. Snažil jsem se zachovat doporučené oblasti příkazů a potvrzení. Data jsou přenášena v rámcích (paketech) s definovaným začátkem a koncem a pevně stanovenou délkou. Lze tak vypočítat při znalosti přenosové rychlosti maximální čas pro příjem odpovědi a případně detekovat chybu.

Formát rámce

PRE FRM ADR INST/ACK DATA CR

PRE.....Prefix - slouží k detekci začátku (synchronizaci) rámce (02H).
FRM.....Číslo formátu rámce (27H).
ADR.....Adresa zařízení.
INST/ACK.....Instrukce pro zařízení/odpověď
DATA.....Data závislá na instrukci, případně typu odpovědi
CR.....Zakončovací znak rámce (03H).

Formát 27

Formát 27 je binární protokol s osmibitovou adresou. Instrukce jsou rozděleny na dotaz a odpověď.

Dotaz:

PRE FRM ADR INST DATA CR

Odpověď:

PRE FRM ADR ACK DATA CR

Adresa zařízení (ADR)

Adresa tvoří jeden byte, který jednoznačně určuje konkrétní zařízení mezi ostatními na jedné komunikační lince. Zařízení toto číslo vždy používá pro svou identifikaci v odpovědích na dotazy z nadřazeného systému.

Adresa FFH je „broadcast“. Instrukce s touto adresou jsou určeny pro všechna připojená zařízení. Na tyto instrukce se neodpovídá. Adresy začínají od 30 H (ASCII „0“).

Kód instrukce (INST)

Kód instrukce pro příslušného zařízení. Tento znak rozlišuje jednotlivé příkazy pro zařízení. Je-li přijata platná instrukce (souhlasí ADR mimo adresy FFH), zařízení na takovou instrukci musí odpovědět. Je-li kód instrukce neznámý, zařízení odpoví ACK „Neznámá instrukce“.

Název instrukce	Hodnota	Význam
INSTR_SET_PINS	20H	Nastavení výstupních portů
INSTR_OPEN_DOOR	21H	Otevření dveří
INSTR_READ_PINS	30H	Čtení stavu výstupních portů
INSTR_REQ_DATA	51H	Žádost o číslo karty
INSTR_SET_EEPROM	E2H	Nastavení EEPROM paměti
INSTR_RESET	E3H	Reset zařízení
INSTR_ENABLE_CONFIG	E4H	Povolení konfigurace
INSTR_READ_STATUS	F1H	Stav komunikačních linek a zařízení
INSTR_READ_EEPROM	F2H	Čtení z EEPROM
INSTR_READ_FIRMWARE	F3H	Verze firmware

Tabulka 3: Popis instrukcí protokolu

Kód odpovědi (ACK)

Kód odpovědi příslušného zařízení na instrukci.

Název odpovědi	Hodnota	Význam
ACK_OK	30H	Instrukce proběhla v pořádku
ACK_ERR	31H	Instrukci nelze provést
ACK_UNKNOWN_INSTRUCTION	32H	Neznámá instrukce
ACK_ACCESS_DENIED	34H	Konfigurace není povolena
ACK_DEVICE_ERR	35H	Chybový stav zařízení
ACK_NO_DATA	36H	Zařízení neobsahuje žádné data
ACK_CARD_IN_EEPROM	37H	Číslo karty se shoduje s uloženým v EEPROM
ACK_CARD	38H	Číslo karty

Tabulka 4: Popis odpovědí protokolu

4.8.1 Význam instrukcí

Nastavení výstupních portů

Popis: Instrukce umožní nastavit jednotlivé piny mikroprocesoru na definovanou úroveň.

Dotaz: 20H (pin | hodnota)

Odpověď: 30H

34H konfigurace nebyla povolena

Legenda:

pin Horní 4 bity v byte. Pozice určuje identifikaci pinu

1000 - nevyužito

0100 - MF7LED_NEG – ovládání rozsvícení diody u čtečky karet

0010 - BEEP_NEG – ovládání bzučáku u čtečky karet

0001 - UNLOCKED – ovládání elektrického zámku
hodnota ... Dolní 4 bity v byte pozičně odpovídají nastavení pinů.

Příklad: Otevření dveří na zařízení s adresou 30H.
11H 00010001b
02H 27H 30H 20H 11H 00H 00H 00H 00H 00H 00H 00H 03H
Nastavení všech pinů na log 1 na zařízení s adresou 30H.
71H01110001b
02H 27H 30H 20H 71H 00H 00H 00H 00H 00H 00H 00H 03H

Poznámka: U pinů zakončených na _NEG je provedena negace hodnoty.

Otevření dveří

Popis: Otevření dveří sepnutím relé.
Dotaz: 21H
Odpověď: 30H

Příklad: Otevření dveří na zařízení s adresou 30H.
02H 27H 30H 21H 00H 00H 00H 00H 00H 00H 00H 00H 03H

Poznámka: Doba sepnutého relé je konfigurovatelná ve zdrojovém souboru firmware.

Čtení stavu výstupních portů

Popis: Instrukce umožní zjistit stav jednotlivých pinů mikroprocesoru.
Dotaz: 30H
Odpověď: 30H (0000| hodnota)
Legenda:

hodnota ... Dolní 4 bity v byte. Pozice určuje identifikaci pinu
1000 - nevyužito
0100 - MF7LED_NEG – rozsvícení diody u čtečky karet
0010 - BEEP_NEG – bzučák u čtečky karet
0001 - UNLOCKED – elektrický zámek

Příklad: Dotaz na zařízení s adresou 30H.
02H 27H 30H 30H 30H 00H 00H 00H 00H 00H 00H 00H 03H
Odpověď
02H 27H 30H 20H 01H 00H 00H 00H 00H 00H 00H 00H 03H
01H 00000001b – dveře jsou otevřeny

Poznámka: U pinů zakončených na _NEG je provedena negace hodnoty.

Žádost o číslo karty

Popis: Instrukce umožní zjistit, zda byla přiložena karta ke čtečce karet.
Dotaz: 51H
Odpověď: 36H zařízení nemá žádná data k dispozici
37H (číslo karty 8B) číslo karty bylo nalezeno v interní EEPROM
38H (číslo karty 8B) číslo karty nebylo nalezeno v interní EEPROM, žádost o
 autorizaci

Příklad: Dotaz na zařízení s adresou 30H.
02H 27H 30H 51H 00H 00H 00H 00H 00H 00H 00H 00H 03H
Odpověď, číslo karty v ASCII HEX formátu 92AEBF3C
02H 27H 30H 37H 39H 32H 41H 45H 42H 46H 33H 43H 00H 03H

Poznámka: Pokud zařízení vrátí odpověď ACK_CARD_IN_EEPROM, provede se interně zároveň instrukce INSTR_OPEN_DOOR. Nadřazený systém je pouze informován o této skutečnosti.

Nastavení EEPROM paměti

Popis: Instrukce umožní zapsat do interní EEPROM čísla karet.

Dotaz: E2H (pozice) (číslo karty 8B)

Odpověď: 30H
31H pozice není v povoleném limitu
34H konfigurace nebyla povolena

Legenda:

pozice ...určuje pozici uložení záznamu v EEPROM v rozsahu 0 – 7.
hodnota je udána v ASCII HEX formátu, tj. 30H - 37H

číslo karty ... číslo musí být uvedeno v ASCII HEX formátu

Příklad: Zápis do EEPROM, pozice 6, číslo karty 92AEBF3C, na zařízení s adresou 30H.
02H 27H 30H 36H E2H 39H 32H 41H 45H 42H 46H 33H 43H 03H

Reset zařízení

Popis: Instrukce provede teplý restart mikroprocesoru.

Dotaz: E3H

Odpověď: 30H

Příklad: Reset na zařízení s adresou 30H.
02H 27H 30H E3H 00H 00H 00H 00H 00H 00H 00H 00H 03H

Poznámka: Další instrukce smí následovat až po 500 milisekundách. Obsah veškerých proměnných kromě EEPROM dat je vymazán.

Povolení konfigurace

Popis: Instrukce nastaví příznak povolení konfigurace zařízení. Systém dvouúrovňové konfigurace předchází možné chybné konfiguraci při výskytu chybě na komunikačním kanálu.

Dotaz: E4H

Odpověď: 30H

Poznámka: Každá instrukce mimo této ruší příznak povolení konfigurace, proto musí každá konfigurační instrukce následovat bezprostředně po INSTR_ENABLE_CONFIG.

Stav komunikačních linek a zařízení

Popis: Instrukce umožní zjistit stav komunikačních linek RS232 a RS485.

Dotaz: F1H

Odpověď: 30H (RS485) (RS232) (zařízení)

Legenda:

RS485 ...	X	X	RXOVR	FE	RXERR	RXOVR	RXBE	RXBF
-----------	---	---	-------	----	-------	-------	------	------

bit 7 – 6 Reserved
bit 5 RXOVR:
Přeplnění hardwarového EUSART bufferu
bit 4 FE:
Format Error, přijat neplatný paket
bit 3 RXERR:
Přijat neplatný znak
bit 2 RXOVR:
Přeplnění softwarového bufferu
bit 1 RXBE:
Prázdný softwarový buffer
bit 0 RXBF:
Softwarový buffer zaplněn

RS232 ...	X	X	X	X	RXERR	RXOVR	RXBE	RXBF
-----------	---	---	---	---	-------	-------	------	------

bit 7 – 4 Reserved
bit 3 FE:
Format Error, přijat neplatný paket
bit 2 RXOVR:
Přeplnění softwarového bufferu
bit 1 RXBE:
Prázdný softwarový buffer
bit 0 RXBF:
Softwarový buffer zaplněn

Zařízení ...	X	X	X	X	X	X	X	CFAIL
--------------	---	---	---	---	---	---	---	-------

bit 7 – 1 Reserved
bit 0 CFAIL:
Komunikace s nadřazeným systémem byla nedostupná

Příklad: Stav zařízení s adresou 30H.

02H 27H 30H F1H 00H 00H 00H 00H 00H 00H 00H 00H 00H 03H

Odpověď:

02H 27H 30H 02H 01H 00H 00H 00H 00H 00H 00H 00H 00H 03H

RS485... softwarový buffer je prázdný (02H)

RS232... softwarový buffer je plný (01H), další přijatý znak bude ztracen

Zařízení... nebyl ztracen kontakt s nadřazeným systémem

Čtení z EEPROM

Popis: Instrukce umožní přečíst čísla karet uložených v interní EEPROM .
Dotaz: E3H (pozice)
Odpověď: 30H (pozice) (číslo karty 8B)
31H pozice není v povoleném limitu

Legenda:
pozice ... určuje pozici uložení záznamu v EEPROM v rozsahu 0 – 7.
hodnota je udána v ASCII HEX formátu, tj. 30H - 37H
číslo karty ... číslo musí být uvedeno v ASCII HEX formátu

Příklad: Čtení z EEPROM, pozice 6 na zařízení s adresou 30H.
02H 27H 30H E3H 00H 00H 00H 00H 00H 00H 00H 00H 03H
Odpověď
02H 27H 30H 30H E2H 39H 32H 41H 45H 42H 46H 33H 43H 03H
Na pozici 6 bylo uloženo číslo karty 92AEBF3C.

Verze firmware

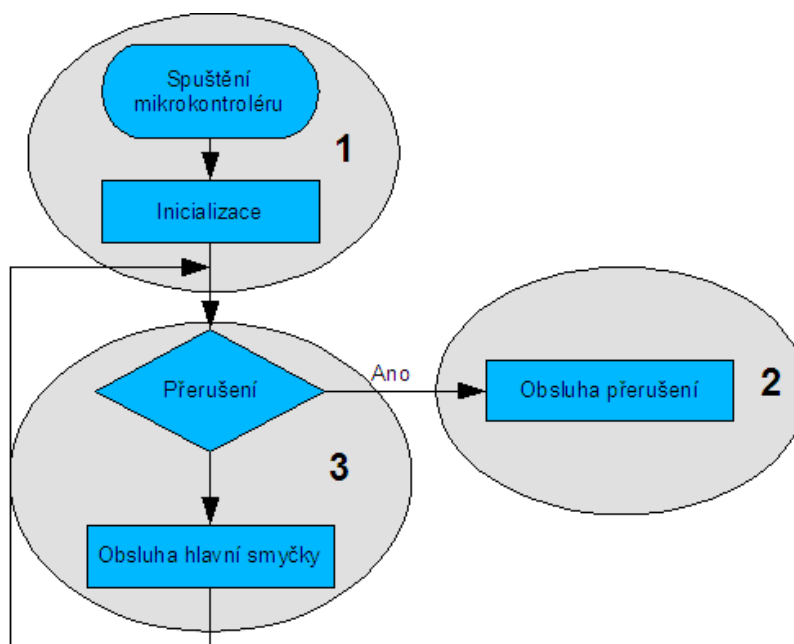
Popis: Zařízení vrátí verzi firmwaru
Dotaz: F3H
Odpověď: 30H (verze 9B)

Příklad: Verze firmware na zařízení s adresou 30H.
02H 27H 30H F3H 00H 00H 00H 00H 00H 00H 00H 00H 03H
Odpověď:
02H 27H 30H 35H 61H 32H 37H 76H 31H 2EH 30H 00H 00H 03H
Verze firmware(vyjádřeno v ASCII) „5a27v1.0“

4.9 Firmware dveřních elektronik

Dveřní elektronika se ve své podstatě chová pouze jako most mezi rozhraním čtečky a řídicího počítače, navíc s možností ovládání zámku(relé). V první fázi vývoje jsem uvažoval o použití assembleru, jelikož mikroprocesor má k dispozici pouze 256B RAM. Nakonec jsem se ale rozhodl pro jazyk C. Po vyzkoušení trial-verze překladače od firmy HI-TECH jsem porovnáním testovacího vygenerovaného kódu zjistil rozdíl nárůstu pouze o 10% oproti mému řešení.

4.9.1 Základní struktura firmwaru



Obrázek 23: Struktura programu mikroprocesoru

Běh programu by se dal rozdělit na 3 části podle procesů a úloh které řeší.

Část 1 - INICIALIZACE

V této části dojde ke spuštění mikroprocesoru a proběhne proces inicializace mikroprocesoru. Nastaví se směry všech vstupně/výstupních bran, provede se nastavení potřebných přerušení a jejich priorita, nastavení obvodů časovačů a taktovací děličku sériového portu. Zároveň je provedena inicializace proměnných.

Část 2 – OBSLUHA PŘERUŠENÍ

Tato část zpracovává asynchronní události, které se vyskytnou v průběhu chodu aplikace. Přerušení je rozděleno do dvou částí, s vyšší a nižší prioritou. Vyšší priorita je přiřazena události zpracování dat od čtečky karet, neboť softwarová obsluha je časově závislá a vyžaduje deterministický přístup. Nižší prioritu má obsluha dat příjmu instrukcí z řídicího počítače a interní časovací obvod, jelikož ke své funkci využívají pomocný hardware implementovaný přímo na čipu.

Část 3 – HLAVNÍ CYKLUS

Po inicializaci se program dostane do části 3 a zacyklí se do nekonečné smyčky, ve které je připraven vykonávat podprogramy podle instrukcí přijatých přerušením v části 2.

4.9.2 Inicializace

Hned po spuštění mikroprocesoru se nakonfigurují registry. Konfigurace probíhá v následující fázi:

- Konfigurace režimu práce mikroprocesoru
- Vypnutí nadbytečných částí hardware
- Nastavení priorit pro přerušení
- Nastavení interního časovače
- Povolení globálního přerušení

V druhé části inicializace jsou nastaveny parametry komunikačních linek a vstupně/výstupních pinů

- Konfigurace softwarového UART
- Konfigurace a inicializace LED a BEEP signálů u čtečky
- Konfigurace hardwarového EUSART

4.9.2.1 Konfigurace režimu práce mikroprocesoru

Režim práce se řídí pomocí konfiguračních bitů uložených na adresách 3000000H-3FFFFFFH. Jsou tedy mimo adresní prostor a jejich nastavení se provádí při uploadu firmwaru. Detailní popis významu lze nalézt v technické dokumentaci k mikroprocesoru [9]. Překladač od firmy HI-TECH má interně zabudovanou podporu pro konfiguraci startovacích podmínek a zavádí pomocné makro `__CONFIG(Konfigurační registr, Typ)`, které se postará o příslušné nastavení.

V programu jsem použil následující nastavení:

```
__CONFIG(1, HS)
__CONFIG(2, PWRTEN & BOREN & WDTON & WDTPS512)
__CONFIG(3, MCLR);
__CONFIG(4, LVPDIS & DEBUGDIS & STVREN);
__CONFIG(5, UNPROTECT);
__CONFIG(6, WRTEEN);
__CONFIG(7, TRU);
```

HS	Mikroprocesor je taktován vysokofrekvenčním oscilátorem
PWRTEN	Vykonávání instrukcí je opožděno z důvodu stabilizace napětí
BOREN	Při poklesu napětí dojde k restartu
WDTON	Watchdog timer, pokud není nulován časovač do 2 s, provede se restart
MCLR	Externí reset povolen
LVPDIS	Nízkonapěťové programování nepovoleno
DEBUGDIS	Debugovací režim vypnut
STVREN	Ochrana přetečení zásobníku zapnuta
UNPROTECT, TRU	Ochrana přečtení a zápisu do paměti vypnuta

4.9.2.2 Vypnutí nadbytečných částí hardware

Vypnutím nepotřebných funkčních bloků se dosáhne úspora energie a je možné použít piny přiřazené pro tyto funkce na běžné digitální I/O funkce. V této aplikaci není použit integrovaný AD převodník a komparátor.

4.9.2.3 Nastavení priorit pro přerušení

Mikroprocesor umožňuje určitému typu přerušení nastavit prioritu provádění. Pokud se vyskytnou obě přerušení v jeden okamžik, s vyšší prioritou má přednost. Podobně, pokud se nacházíme v obsluze přerušení s nižší prioritou, je při výskytu přerušení s vyšší prioritou zavolána jeho obsluha.

4.9.2.4 Nastavení interního časovače

Interní časovací obvod Timer2 je využit jako čítač a při přetečení hodnoty je generováno přerušení. Nastavení se dělí od taktovacího kmitočtu procesoru a je následně upraveno pomocí předděličky

na generování přerušeni každých 20 milisekund. V obsluze přerušeni je využita lokální statická proměnná. Její inkrementací lze dosáhnout po padesátém cyklu přibližný takt jedné sekundy.

4.9.2.5 Konfigurace softwarového UART

PIC181320 obsahuje pouze jeden hardwarový univerzální asynchronní port, který je využit pro příjem dat z řídicího počítače. Pro příjem dat ze čtečky musela být napsána softwarová rutina, která implementuje podobné chování. Oproti hardwarovému řešení má řadu nevýhod, např. omezení samplování vzorku pouze na jeden, závislost na aktuální zátěži mikroprocesoru.

Čtečka vysílá RS232 data s následujícími parametry 9600, N, 8, 1 (baud, parita, data bit, stop bit), každý bit má délku $\frac{1}{9600} \approx 104 \mu s$ a časování vzorků je provedeno pomocí interního časovacího obvodu Timer1. Hodnoty pro časovač jsou počítány podle následujícího vzorce:

$$NEXT_{BIT} = \frac{XTAL_{FREQ}/4}{9600}$$

$$WAIT_{BIT} = NEXT_{BIT} + \frac{XTAL_{FREQ}/16}{9600} - INTLATENCY$$

INTLATENCY...prodleva při přerušení

4.9.2.6 Konfigurace a inicializace LED a BEEP signálů u čtečky

Konfigurace pinů v mikroprocesoru se zajišťuje pomocí tzv. TRISA a TRIXB registrů. Jednotlivé bity odpovídají režimu pinu. Výrobce zvolil mnemotechnickou pomůcku a uložením logické 1 se nastaví pinu funkce „vstupní“, podle počátečního anglického písmena „Input“. Logická 0 pak znamená režim výstupu „Output“. Např. TRISA(5) = 0 nastaví šestý pin na bráně A do režimu výstupu.

Signály LED a BEEP se aktivují spojením se signálem GND. Piny jsou tedy nastaveny do režimu „O“. Nastavením na logickou 0 se pin propojí s GND a signál je aktivován.

4.9.2.7 Konfigurace hardwarového EUSART

Integrovaný UART umožňuje příjem a vysílání dat bez účasti mikroprocesoru. Obsahuje vlastní časovací obvod, obvody pro detekci start a stop bitu a každý bit je vzorkován 3krát, přičemž logická hodnota je spočtena z majoritní části. Úspěšný přenos/příjem je indikován přerušením.

Spojení s řídicím počítačem probíhá s následujícími parametry 19200, N, 8, 1 (baud, parita, data bit, stop bit).

Konfigurace rychlosti přenosu se řídí hodnotou v SPBRG registru a je odvozena od taktovací frekvence. Hodnota je počítána podle následujícího vzorce:

$$SPBRG_{VAL} = \frac{XTAL_{FREQ}/19200}{4} - 1$$

Jelikož musí být výsledkem celé číslo, je důležité zkontrolovat odchylku od požadované rychlosti. Pro zachování stability je důležité, aby odchylka nebyla více než 1%.

$$Reálná\ rychlost = \frac{XTAL_{FREQ}}{4 * (SPBRG_{VAL} + 1)}$$

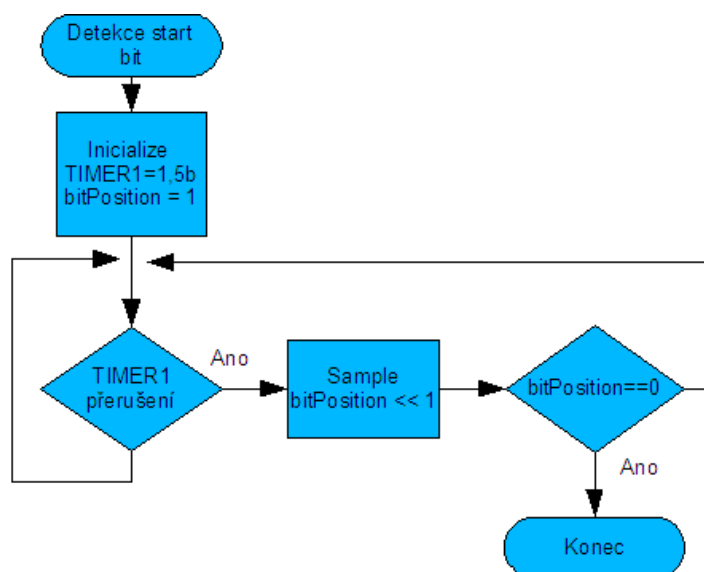
$$Odchylka = \frac{Skutečná\ rychlost - Požadovaná\ rychlost}{Požadovaná\ rychlost} * 100$$

4.9.3 Obsluha přerušení

Program využívá tyto přerušení

- Vysoká priorita
 - detekce přechodu high-low a interní časovač Timer1
- Nízká priorita
 - příjem dat na EUSART
 - interní časovač Timer2

4.9.3.1 Detekce přechodu high-low



Obrázek 24: Detekce náběžné hrany

Tento typ přerušení je využit u implementace softwarového UARTu. Využívá se principu, kdy dle normy musí být linka v klidu ve stavu logické 1 a asynchronní přenos je zahájen přechodem na logickou 0 (start bit). S použitím převodníku se přechod projeví změnou napětí z 5V na 0V. Změna stavu je detekována sestupnou hranou na pinu se zapojeným Schmitt trigger bufferem. Po detekci je generováno přerušení a spuštěn čítač s inicializační hodnotou na délku 1,5 bitu. Vzorkování končí po osmé iteraci.

Výsledný přijatý byte je tvořen vzorky samplovanými po bitové délce. Standard RS232 vyžaduje, aby LSB (Least Significant Bit) byl vyslán na linku první. Tohoto využívá výše zmíněný algoritmus. Při inicializaci je statická proměnná bitPosition nastavena na 00000001b. Hodnota vzorku se ukládá na pozici, kterou určuje log 1. Po odebrání vzorku je proveden logický posun doleva, značící novou pozici. Po osmé iteraci je log 1 „vystrčena“ a bitPosition obsahuje nulu. Tento stav definuje odebrání 8mi vzorků, tedy 1 byte.

Výše uvedený postup je náhrada za známé makro v C

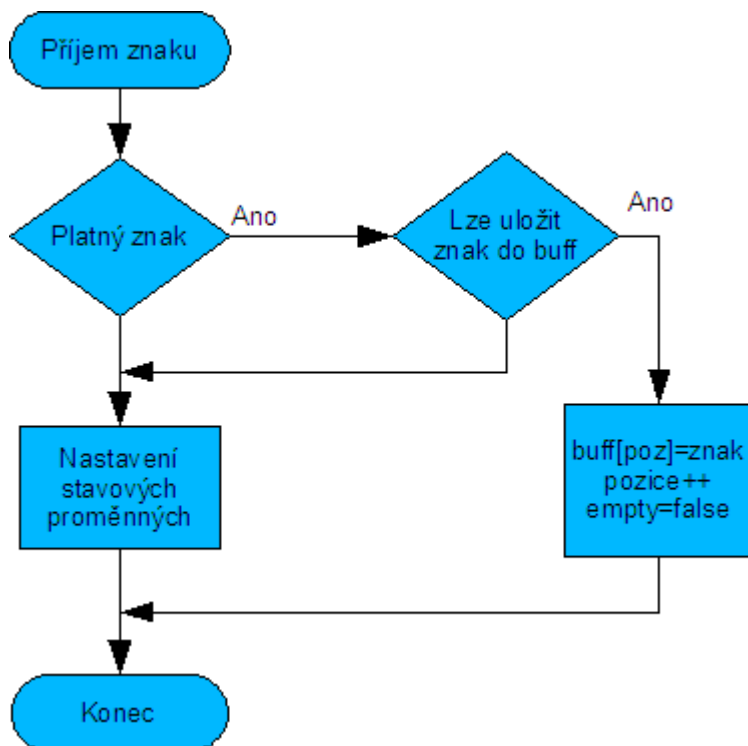
```
#define BIT_SET(var;bitno) ((var) |= 1 << (bitno)).
```

Porovnáním vygenerovaných asm instrukcí jsem použitím složitějšího postupu ušetřil přibližně 60% z celkové délky 50 instrukčních slov pro tuto funkci.

4.9.3.2 Příjem dat na EUSART

Přijatý platný byte se dočasně uloží do vyrovnávacího jednobytového registru a vygeneruje se

přerušení. Obslužná rutina vyjme znak a uloží ho do kruhového bufferu a případně podle příznakových registrů modifikuje stavové proměnné o komunikační lince. Oproti softwarové implementaci je zde možno detekovat poškozené rámce. Ostatní operace jsou stejné.

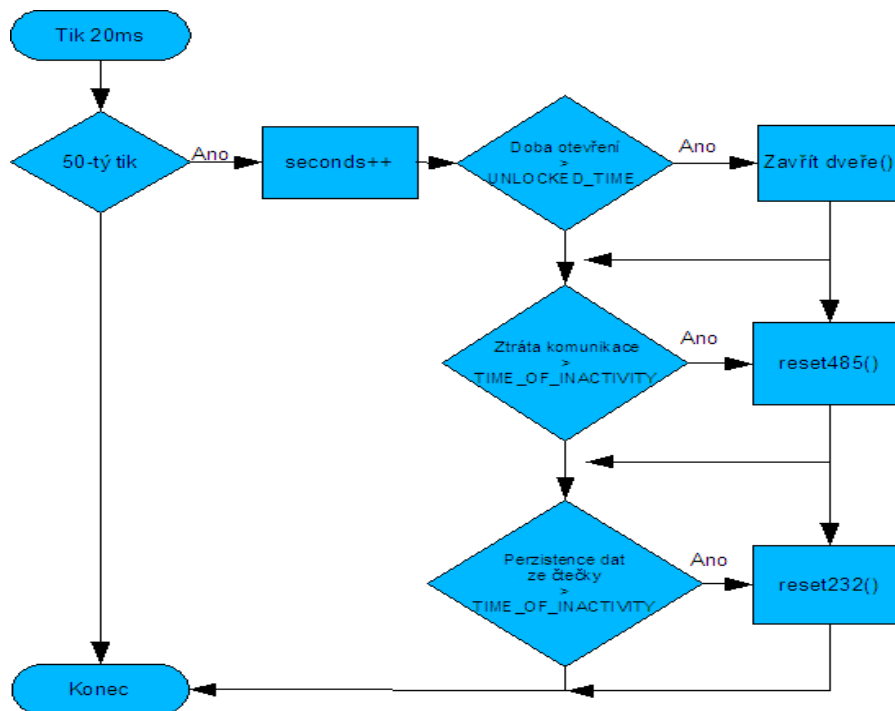


Obrázek 25: Přijem znaku na hw UART

4.9.3.3 Interní časovač Timer2

Přerušení je generováno každých 20 milisekund. Obslužná rutina využívá statickou proměnnou, s každým přerušením jí inkrementuje a otestuje, zda není rovna padesáti. Pokud ano, inkrementuje globální proměnnou seconds, která uchovává počet sekund od spuštění mikroprocesoru. Přetečení počtu sekund v této aplikaci není ošetřeno, neboť všechny požadavky vyžadující časový rozdíl nepřesáhnou 254s. Nakonec je zavolána funkce timer1Sec() v níž se zjišťuje

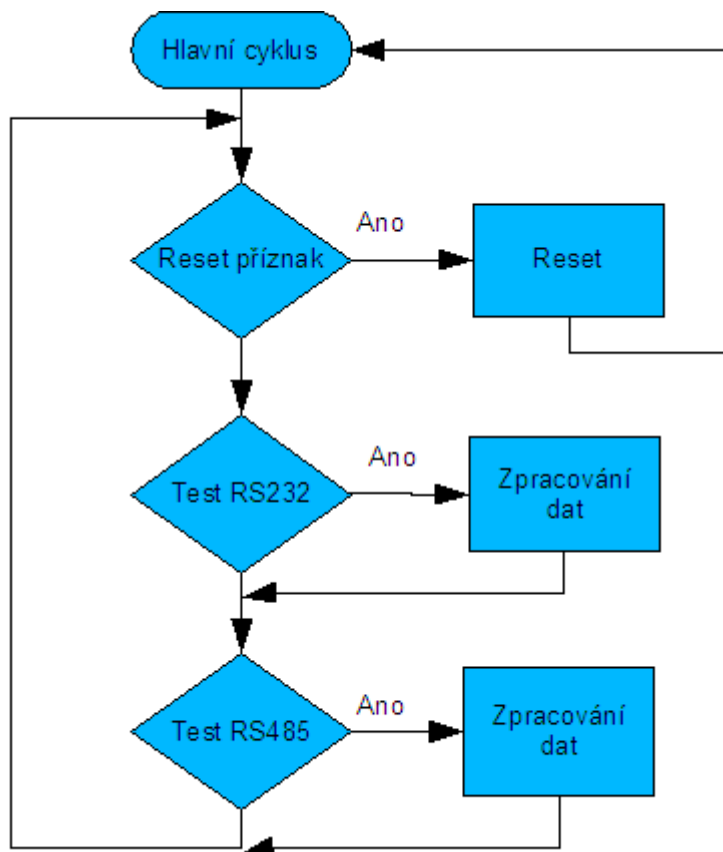
- doba otevření dveří nepřesáhla UNLOCKED_TIME
- výpadek komunikace s řídicím počítačem není delší než TIME_OF_INACTIVITY
- doba platnosti dat ze čtečky karet není delší než TIME_OF_INACTIVITY



Obrázek 26: Obsluha časovače Timer2

4.9.3.4 Hlavní cyklus

Po inicializaci se hlavní program zacyklí do nekonečné smyčky a testuje proměnné, zda se má provádět určitá úloha. Proměnné, které mohou měněny být nezávisle na běhu programu (paměťově mapované registry, globální proměnné v přerušovacích obsluhách) by měly obsahovat identifikátor *volatile*. Kompilátor u těchto proměnných nebude provádět optimalizaci a vždy načte hodnotu z paměti.



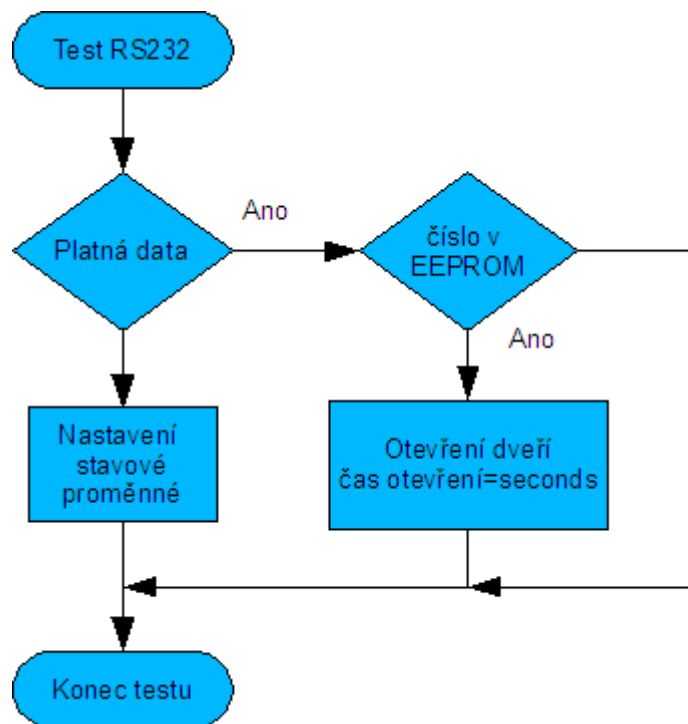
Obrázek 27: Hlavní smyčka programu pro mikroprocesor

4.9.3.5 Testování příznaku reset

Tento test je pozůstatek z původního firmwaru, který byl psán pro mikroprocesor PIC16F628A. Tento mikrokontrolér neměl implementovanu instrukci RST a reset se prováděl skokem na adresu 000000H. Provést „reset“ z funkce obsluhující příkazy z řídicího počítače nebylo možné. Každé volání funkce znamenalo uložení návratové adresy na omezený hardwarový zásobník. Pokud by se provedlo několik resetů za sebou, došlo by k podtečení a program by zkolaboval. Současný mikroprocesor již instrukci obsahuje a její chování je obdobné teplému restartu u PC.

4.9.3.6 Test RS232

Podmínkou úspěšného testu je odpovídající počet znaků v RS232 kruhovém bufferu a nepřítomnosti jiného čísla karty v bufferu. Počet znaků je dán protokolem čtečky. Následně jsou data postupně extrahována a je prováděna kontrola na dodržení platnosti protokolu. Číslo karty je porovnáno s hodnotami uloženými v EEPROM a v případě shody jsou neprodleně otevřeny dveře.



Obrázek 28: Test přijatých dat z RS232

4.9.3.7 Test RS485

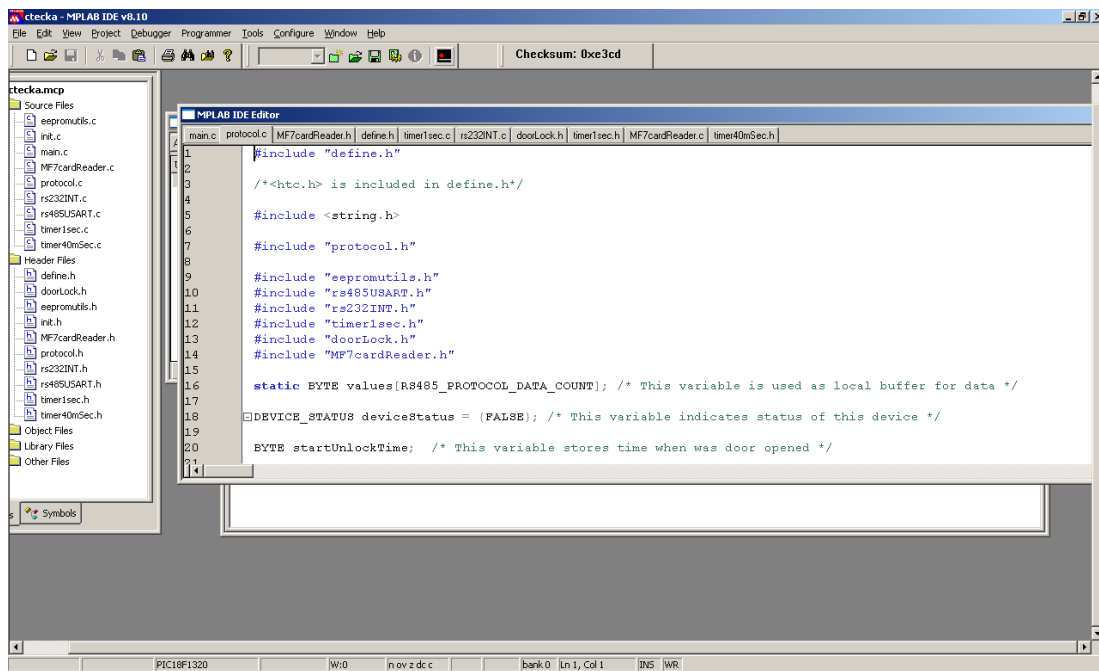
Podmínka je obdobná jako v předešlém odstavci. Testuje se ale pouze počet znaků v bufferu. Po ověření integrity se přechází ke zpracování dat z protokolu pokud souhlasí přijatá adresa s lokálně nastavenou, případně se jedná o broadcast.

4.9.4 Upload firmwaru a debugging

K nahrání firmwaru do mikroprocesoru PIC18 postačí i jednoduchý, ručně sestavený programátor, jehož zapojení lze najít na Internetu, případně lze koupit univerzálnější programátor Presto od firmy ASIX [10]. Sofistikovanější pomoc nabízí programátor a debugger ICD2 přímo od firmy Microchip. Má výrazný kulatý tvar a proto se mu někdy říká také puk. Se spojením s prostředím MP-LAB lze ladit a krokovat aplikaci na osazené DPS, pokud se dodrží doporučené zapojení pinů.

Sestavení(build) programu probíhá zcela transparentně a pokud se ve zdrojových kódech nevyskytne sémantická chyba, je vygenerován obraz(image) v Intel HEX formátu, případně lze vhodným nastavením zvolit jiný typ. Tento postup je v případě využití externího programátoru. V případě ICD2 se program nahraje pouze výběrem z menu.

MP-LAB zároveň obsahuje simulátor pro vybraný typ mikrokontroléru. Vývojář má k dispozici spolu s ICD2 mocné prostředí na vytváření robustních aplikací.



Obrázek 29: MP-LAB vývojové prostředí

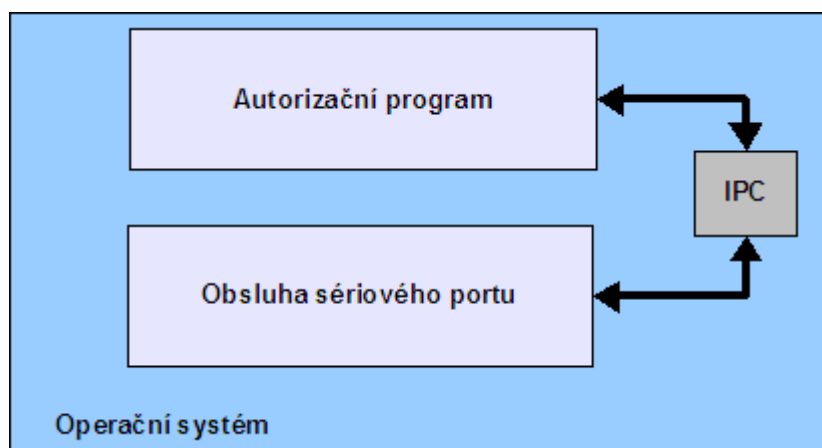
5 Ovládací software řídicího počítače

Ačkoliv jsou dveřní elektroniky schopné samostatně operovat až s 8 uloženými kartami v EEPROM, pro nasazení s větším počtem uživatelů je toto řešení nereálné. Centrálně řízené systémy nabízejí mnohé výhody, jak z technického, tak z uživatelského hlediska.

Aplikace se skládá ze dvou modulů:

- Obsluha sériového portu
- Autorizační software

Moduly mezi sebou komunikují pomocí IPC zpráv (Inter-Process Communication). Mechanismus zasílání zpráv patří mezi základní postupy výměny dat mezi různými procesy. V Linuxovém prostředí a v některých skoro operačních systémech (např. MS Windows) se navíc využívají metody sdílené paměti, semaforey, signály aj. pro komunikaci mezi procesy. Zasílání zpráv jsem zvolil z důvodu jednoduchosti programování a pro aplikaci jsou postačující. Samotný přístup na periférie je ponecháno čistě v režii operačního systému.

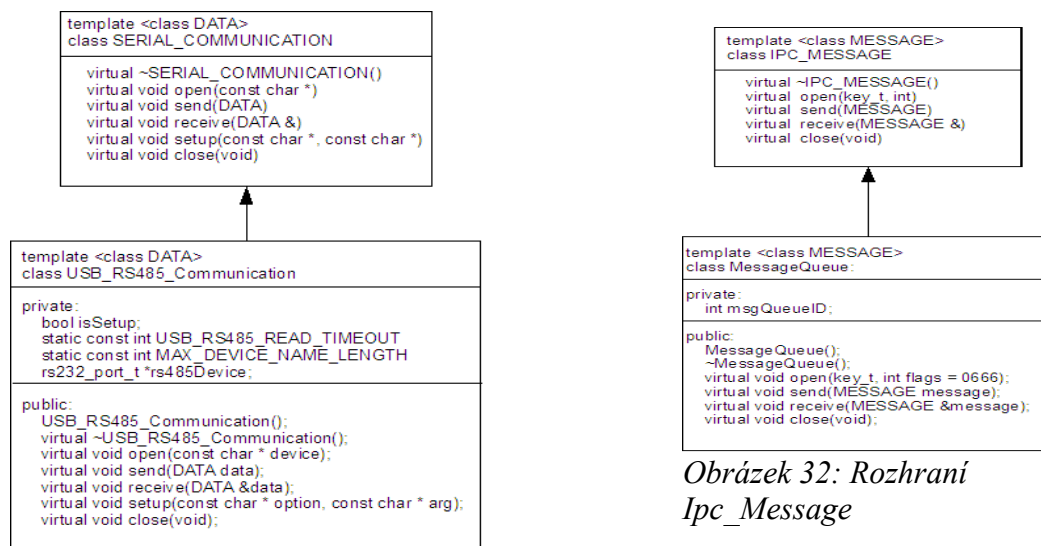


Obrázek 30: Schéma komunikace modulů

5.1 Obsluha sériového portu

Ovládací program je napsán v jazyce C++ s podporou STL, která umožňuje zajistit protokolovou nezávislost. Původní verze napsaná v C byla silně implementačně závislá a každá změna vyžadovala přepsání logiky aplikace.

Parametrické třídy(strukтуры) MESSAGE a DATA jsou již implementačně závislé.



Obrázek 31: Rozhraní Serial_Communication

Obrázek 32: Rozhraní Ipc_Message

Aplikace používá pro logování open-source knihovnu Log for C++. Log4cpp je knihovna C++ tříd pro flexibilní logování do souborů, syslogu, IDSA popř. jiných míst. Navazuje na úspěšný projekt Log4j Java a snaží se implementovat podobné API volání funkcí. Zdrojové kódy a dokumentaci lze nalézt na oficiálních stránkách [11].

Aplikace je psaná pomocí IDE Netbeans s podporou C/C++ [12]. Toto prostředí automaticky generuje makefile soubor a není nutné ho mít při sestavení k dispozici. Kompilaci lze provést standardními nástroji pod OS Linux (make, gcc). Při linkování je důležité přidat parametr „--static“. Tento parametr zajistí statické linkování knihoven přímo do těla programu.

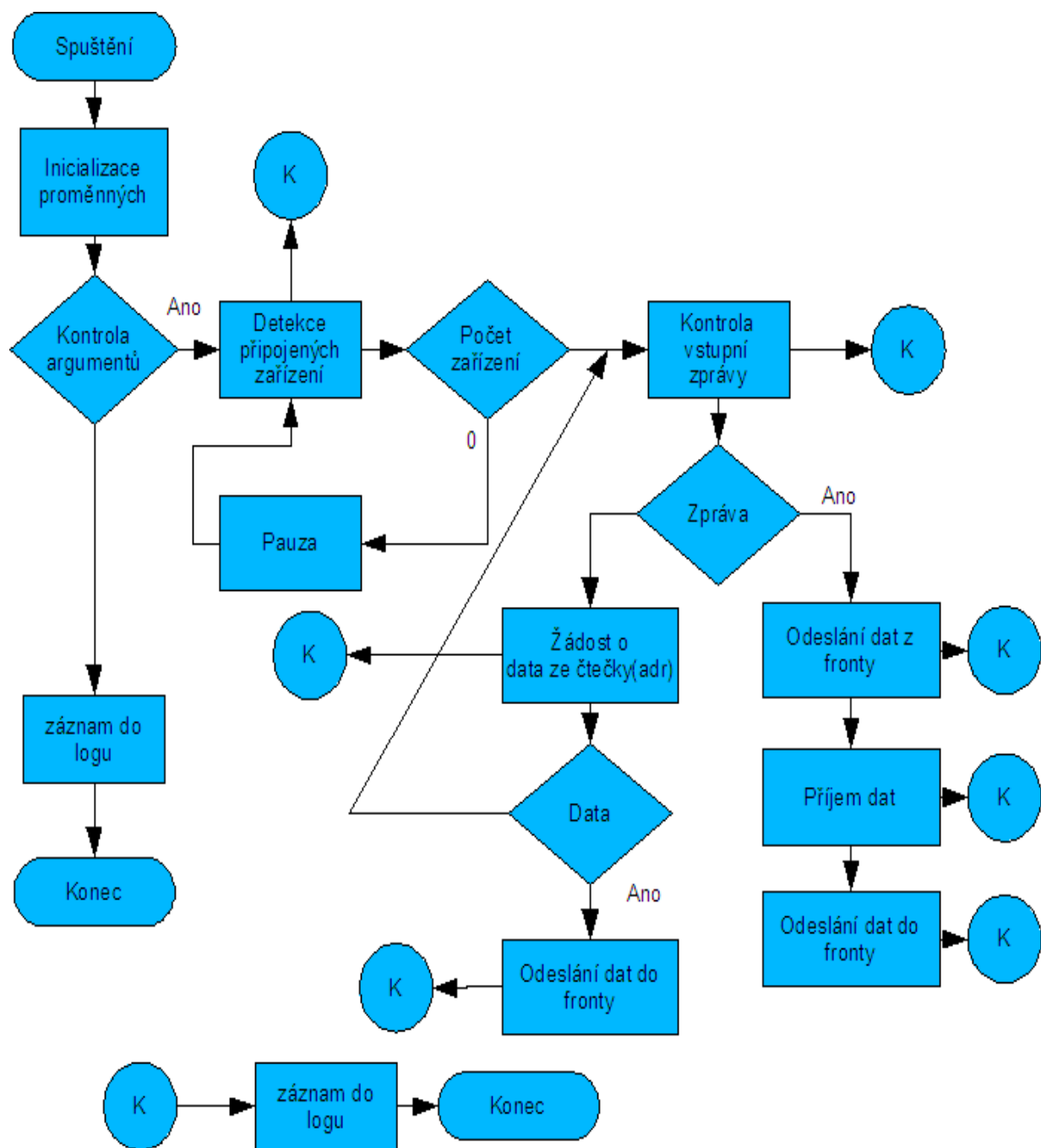
Spuštění aplikace je zajišťováno Autorizačním programem. Ke svému běhu potřebuje předpřipravené prostředí, tj. existující sériový port a inicializované fronty zpráv. Informace jsou předávány pomocí argumentů s následující syntaxí:

program -i vstupní fronta zpráv -o výstupní fronta zpráv -d „název zařízení“

Syntaxe je důsledně kontrolována a v případě chyby je program ukončen. Obsluha chyb je zajišťována pomocí vyjímek (Exceptions), které jsou logovány do souboru. Kritické chyby, např. selhání čtení z portu, nepovolený zápis do fronty zpráv znamenají ukončení programu s návratovým kódem EXIT_FAILURE.

5.1.1 Vývojový diagram

Program se po spuštění snaží automaticky zmapovat zařízení připojená na sběrnici. Vysílá na celý adresní rozsah žádost o poskytnutí čísla firmwaru (instrukce INSTR_READ_FIRMWARE). Pokud neobdrží žádnou odpověď (čekání na vstupní data z portu vyprší), uvede se do režimu spánku a po definované době se znovu spustí detekce zařízení. V opačném případě se program zacyklí do nekonečné smyčky a začne nejdříve přijímat zprávy z fronty zpráv. Následně postupně vysílá na všechny detekované zařízení žádost o data (instrukce INSTR_REQ_DATA). Zařízení, které se odebere ze sběrnice za běhu zařízení způsobí po vypršení maximálního počtu chybových stavů novou detekci připojených zařízení.



Obrázek 33: Struktura obslužného programu pro sériový port

5.1.2 Formát IPC zprávy

Zprávu tvoří 11 bytů dlouhý řetězec a tvoří podmnožinu komunikačního protokolu.

Význam pozic v řetězci:

0	adresa zařízení
1	instrukce/odpověď
2-10	data

Spolu se zprávou je posílán také její typ. Přijímací aplikace si zaregistruje u operačního systému typ, který jí zajímá a ostatní jsou ignorovány.

MSGTYPE_ALL 0	Každá zpráva je přijata
MSGTYPE_PHP 8	Typ zpráv pro autorizační software

5.2 Autorizační program

Autorizační aplikace zastává nejdůležitější funkci řídicího počítače. Na základě přístupových práv rozhoduje, zda má daný uživatel přístup do místnosti v daný časový okamžik. Zároveň informuje o událostech, pokud se některý uživatel snaží vstoupit do jemu nepřístupných prostor. Jde o nejkompaktnější součást celého přístupového systému, proto jsem jejímu vývoji věnoval zvláštní pozornost.

Aplikace je napsána v programovacím jazyku PHP verze 5. Tento interpret, ačkoliv je mnohými odsuzován pro svoji koncepci, umožňuje rychlý vývoj aplikací s využitím hotových frameworků a rozsáhlé dokumentace. Původně byl tento jazyk navrhnut pro webové prostředí na psaní jednoduchých CGI skriptů, ale postupem času se z něj stává plnohodnotný jazyk, syntaxí připomínající Javu se zachováním svých specifíků. V Linuxovém prostředí, pro které byl primárně určen, je možno využít všech výhod poskytovaných operačním systémem, neboť interpret je napsán v jazyce C. Ty části kódu, které spotřebovávají mnoho systémových prostředků v PHP lze jednoduše a efektivně napsat v C a do PHP přiložit jako rozšíření (extension).

Jazyk umožňuje psát v procedurálním i objektovém stylu, případně oba kombinovat. Pro aplikaci jsem zvolil objektový styl, pouze v ojedinělých případech jsou využity procedurální principy. V dalším textu budu používat následující dělení:

- program, aplikace – souhrn zdrojových programů
- skript – jednotlivý zdrojový soubor s příkazy v jazyce PHP

5.2.1 Struktura aplikace

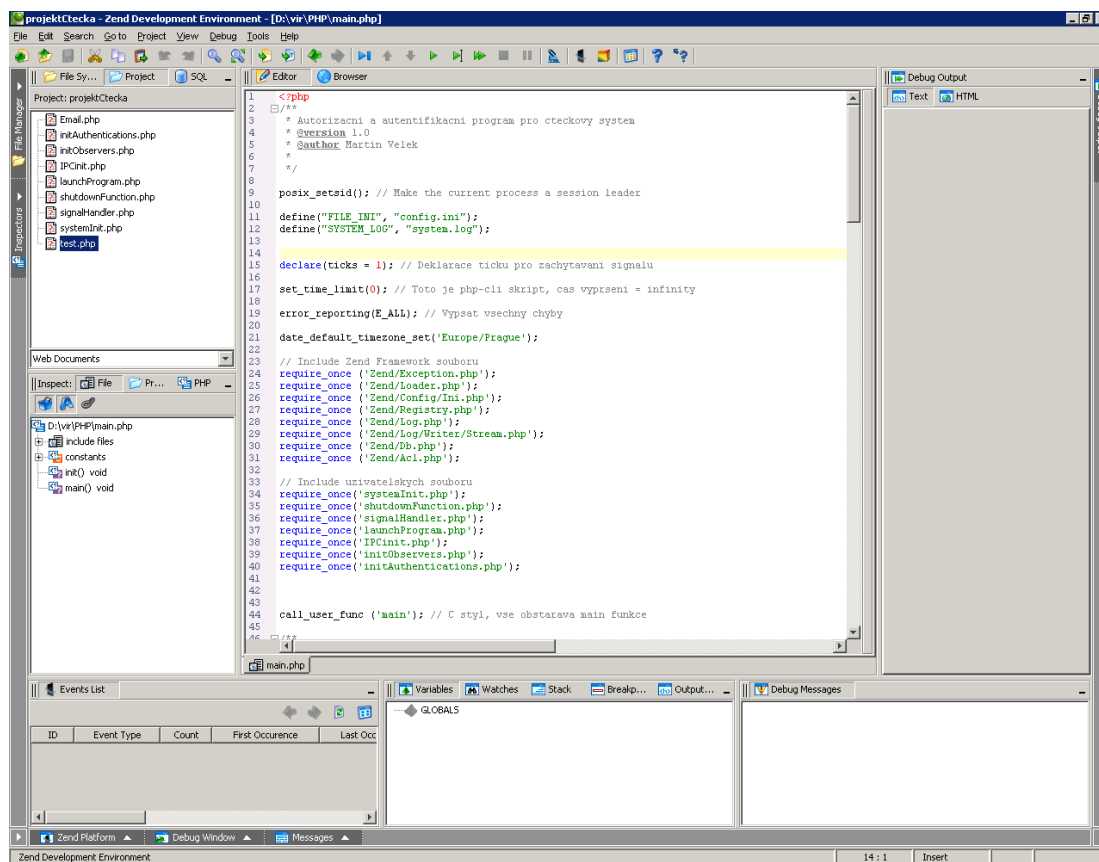
Program je po spuštění vykonáván interpretem řádek po řádku. Pokud není příkaz definicí funkce nebo třídy je vykonán. V rozsáhlé aplikaci může vzniknout nepřesnost, který zdrojový soubor je výchozí. Z tohoto důvodu jsou podprogramy psány ve formě definic tříd a funkcí. V aplikaci existuje pouze jeden hlavní skript (main.php), který naimportuje ostatní.

Start aplikace je zajištěn voláním funkce `call_user_func ('main')`, podobné chování lze nalézt v jazyce C, kde je funkce `main()` vstupním bodem programu.

Při návrhu aplikace jsem využil existujícího open-source frameworku od společnosti Zend [13]. Tento framework je postaven na čistě objektových rysech a implementuje řadu vlastností známých z jiných jazyků, například návrhové vzory (design patterns). Dalším pomocným nástrojem při vývoji

aplikaci je Zend Studio IDE. Toto prostředí se svými funkcemi neliší např. od Eclipse, navíc společnost Zend začala vyvíjet modul přímo do prostředí Eclipse.

Rozdělení aplikace na funkční bloky je velmi podobný firmwaru u mikroprocesoru. V celé práci



Obrázek 34: Vývojové prostředí Zend Studio

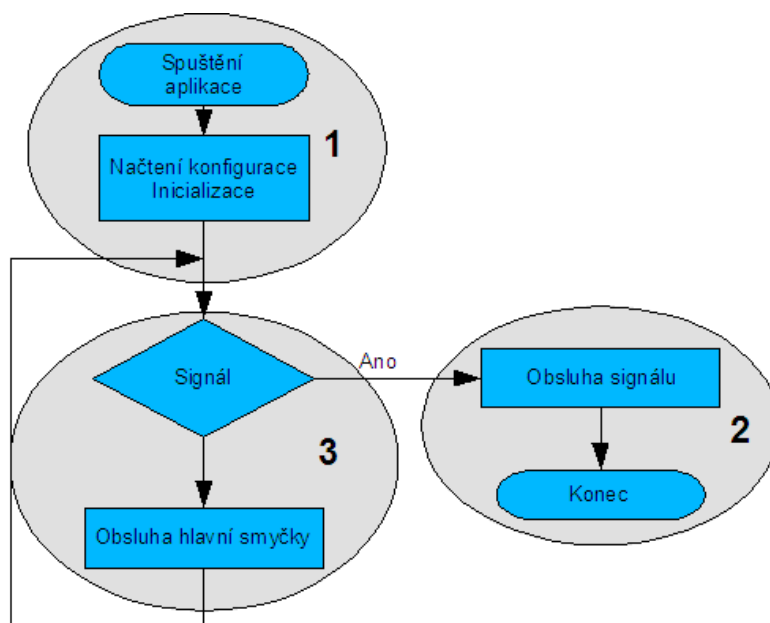
jsem se snažil zachovat jednotnou formu, aby byla zachována přehlednost a jednoduchost.

Z frameworku jsou využity tyto komponenty:

- **Zend_Config**
Zjednodušuje přístup ke konfiguračním údajům uvnitř aplikace. Konfigurační data mohou pocházet z různých zdrojů, podporujících hierarchické uspořádání. V současné době je implementována podpora pro INI formát a XML.
- **Zend_Exceptions**
Všechny výjimky vyvolané Zend frameworkem jsou odvozeny od základní třídy Zend_Exceptions. Aby je bylo možné odchytit, je definovat tuto třídu.
- **Zend_Loader**
Komponenta umožňující dynamicky nahrát definici třídy. Je výhodná v případech, kdy je název třídy uložen v proměnné.
- **Zend_Log**
Je komponenta pro zjednodušené a všeobecné logování. Podporuje řadu možností cílového logování, formátování a filtrování zpráv.

➤ Zend_Registry

Je kontejner pro uchovávání objektů a proměnných. Uložení do registrů je možno zpřístupnit objekty a proměnné ve všech částech aplikace. Tento mechanismus nahrazuje použití globálních proměnných.



Obrázek 35: Struktura autorizační aplikace

5.2.2 Načtení konfigurace a inicializace

Konfigurace je uložena v textovém souboru, cesta se jménem je definována v hlavním skriptu. Data jsou uložena v sekcích a přístup k nim je umožněn pomocí vnořených třídních proměnných. Tato metoda je založena na možnosti použít objekt jako třídní proměnnou. Operátor „->“ zajišťuje přístup k proměnným a konfigurační hodnotu lze získat následujícím zřetězením \$konfiguracniObjekt->NazevSekceObjekt->Nazev.

Obecný formát souboru je následující:

```
[Název_Sekce]  
Název = hodnota
```

Inicializace má za úkol připravit systémové a pomocné proměnné nutné pro běh aplikace. Probíhá v následujících krocích:

- Registrace shutdown funkce
- Inicializace fronty zpráv
- Nastavení obsluhy pro zpracování signálů
- Spuštění externího programu pro obsluhu sériového portu
- Inicializace autorizačních tříd
- Inicializace observerů
- Inicializace mail klienta

5.2.2.1 Registrace shutdown funkce

Shutdown funkce je zavolána v případě ukončení programu voláním funkce `exit()` nebo ukončením

zpracování skriptu. Jejím úkolem je uzavřít všechna otevřená spojení, ukončit podružný program a pomocí emailové zprávy informovat o ukončení aplikace.

5.2.2.2 Nastavení obsluhy pro zpracování signálů

Operační systém Linux může zaslat procesu zprávu ve formě signálu. Každý signál má číselné vyjádření a má speciální význam. Programátor může rozhodnout, zda bude program tyto signály zachycovat a zpracovávat. Některé signály, např. SIGKILL nelze zachytit a znamená násilné ukončení procesu ze strany operačního systému. Autorizační program zachytává dva typy signálů:

- SIGCHLD - oznámení o ukončení externího programu
- SIGTERM - ukončení aplikace na žádost uživatele

5.2.2.3 Inicializace autorizačních tříd

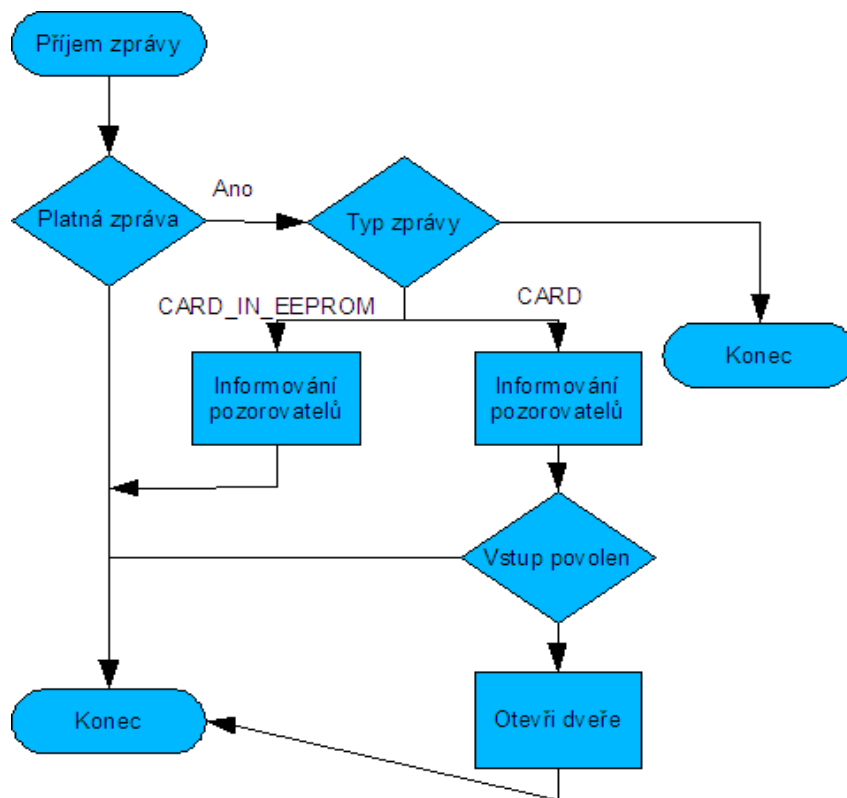
Autorizaci uživatelů lze provádět oproti řadě zdrojů, databáze, LDAP server, textový soubor. V této části se nahrávají moduly autorizace na základě konfiguračního souboru.

5.2.2.4 Inicializace observerů

Observer je typ návrhového vzoru. Každá událost může být zaslána zaregistrovaným pozorovatelům. Obecně se jedná o způsob jak nezávislý objekt informuje závislé objekty o událostech, které je mohou ovlivnit aniž by znaly vnitřní strukturu. V této aplikaci se pozoruje událost vstupu uživatele do dveří.

5.2.3 Obsluha hlavní smyčky

Po inicializaci přechází program do nekonečné smyčky a čeká na zprávy od externího programu. Čekání je pasivní, tzn. aplikace je interně uspána a nevyužívá procesorový čas. Při příjmu zprávy je probuzena a začne ji zpracovávat.



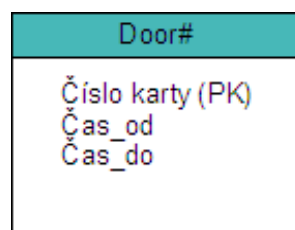
Obrázek 36: Příjem zprávy

5.2.3.1 Autorizační mechanismus

Jednotlivé autorizační třídy implementují rozhraní Authentication s metodu *bool isAllowed(\$cardID,\$door)*. Prvním parametrem je textový řetězec s číslem karty v ASCII HEX formátu, druhý číslo dveří. Návratovou hodnotou je typ Boolean (true/false), který je rozhodující při povolení vstupu.

Seznam autorizačních tříd je uveden v konfiguračním souboru a třídy jsou v případě nenalezení záznamu procházeny postupně, jak byly registrovány. Pokud není ani v jedné z nich povolen přístup, vrací se hodnota *false*.

V testovací aplikaci je navržena třída Authentication_DB autorizující se vůči lokální databázi. Každé zařízení má v databázi přidruženou tabulku s názvem Door(číslo dveří) a strukturou zobrazenou na obr. 37.



Obrázek 37: Struktura tabulky autorizační databáze

5.3 Testovací rozhraní

Pro ověření správné funkčnosti systému byla vytvořena v jazyce PHP utilita, simulující režim autorizační aplikace a umožňuje posílat libovolné instrukce na jednotlivá zařízení. Zároveň je tak prověřena funkce výměny zpráv mezi procesy.

Utilita se ovládá pomocí příkazové řádky se stejnými parametry jako formát IPC zprávy. V adresáři je nutné mít aktuální kopii konfiguračního souboru pro autorizační aplikaci.

Syntaxe: `> php5 Tester.php adresa_zařizeni kód_instrukce d0 d1 d2 d3 d4 d5 d6 d7 d8`

<i>adresa_zařizeni</i>	viz ADR z popisu protokolu
<i>kód_instrukce</i>	viz INSTR z popisu protokolu
<i>d0 – d8</i>	data odpovídající kódu instrukce

Parametry se zadávají v hexadecimálním tvaru

Příklad: Žádost o otevření dveří na zařízení s adresou 31H
`> php5 Testerp.php 31 21 00 00 00 00 00 00 00 00 00`
`> Odpoved: 31 ACK_OK 00 00 00 00 00 00 00 00 00`

6 Uvedení systému do provozu

Při instalaci kabeláže je u sběrnice RS485 důležité správné zapojení vodičů A a B. Pokud je třeba napájet dveřní elektroniky a zámky z centrálního zdroje, lze s výhodou využít EZS kabely od společnosti IMEXTrade [14]. Nabízí stíněný kabel s označením typu 2x0.75+4x0.22, který má 2 páry datových vodičů a jeden pár pro napájení koncových zařízení.

Instalace softwaru na řídicí počítač vyžaduje částečnou znalost operačního systému Linux. V následujících krocích budu popisovat pouze dílčí kroky, přesné detaily postupu lze nalézt na Internetu v sekcích podpory, či v diskuzních fórech.

6.1 Instalace operačního systému

Instalace Ubuntu 8.04 na USB flash disk se neliší od běžné instalace na pevný disk. Je však třeba počítat s většími prodlevami a při rozdělování oddílů a jejich formátování je dobré nastavit atribut „noatime“ a nežurnálovací souborový systém.

Po instalaci a restartování by se měla provést aktualizace seznamu balíčků příkazem „apt-get update“ a provést upgrade zastaralých částí „apt-get upgrade“.