České vysoké učení technické v Praze
Fakulta elektrotechnická

KATEDRA ŘÍDICÍ TECHNIKY

Diplomová Práce

Řídicí systém pro prezentační místnost

Vladimír Kudrna

Vedoucí práce: Ing. František Vaněk

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra řídicí techniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Vladimír Kudrna**

Studijní program: Elektrotechnika a informatika (magisterský), strukturovaný
Obor: Kybernetika a měření, blok KM1 - Řídicí technika

Název tématu: **Řídicí systém pro prezentační místnost**

Pokyny pro vypracování:

1. Prostudujte základní komunikační protokoly pro dataprojektory (RS232, RS485 a IR) a porovnejte jejich možnosti pro řízení (vypnutí, ostření, zoom, přepnutí vstupu, ...).
2. Prostudujte základní komunikační protokoly pro řízení střihacích videomatic.
3. Navrhněte vhodnou strukturu, jak reprezentovat základní prezentační prvky a scénáře (dataprojektor, videomatice, audiomatice, řízení světel, žaluzií, tabulí, ...) v řídicím OS.
4. Navrhněte HW pro řídicí systém, který bude schopen modulárně (RS232/RS485/IR/relé/...) řídit minimálně 8 (lépe 16) vnějších zařízení.
5. Navrhněte vhodný prvek pro vizuální kontrolu řídicího systému (display).
6. Navrhněte způsob, jak řídicí jednotku ovládat přes internet.
7. Dle možnosti realizujte řídicí systém a vyzkoušejte na vybavení místnosti K14/K24/K26.
8. Vytvořte SW, ve kterém bude možné zadat konfiguraci prezentačního HW a scénářů.

Seznam odborné literatury:

Dodá vedoucí práce

Vedoucí: Ing. František Vaněk

Platnost zadání: do konce zimního semestru 2008/2009

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

L.S.

doc. Ing. Boris Šimák, CSc.
děkan

V Praze dne 10. 9. 2007

Poděkování:

Rád bych poděkoval všem lidem, bez kterých by nemohla diplomová práce vzniknout, především však děkuji vedoucímu diplomové práce ing. Františku Vaňkovi za metodické vedení práce, cenné rady a ochotu vždy pomoci.

Prohlášení:

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám žádný důvod proti užití tohoto školního díla ve smyslu 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 12. 5. 2009 .....

podpis

Abstrakt

Moderní učebny díky moderním prezentačním prostředkům velice rozšiřují možnosti výuky, ale také zvyšují nároky na obsluhu díky zvyšujícímu se množství ovládaných přístrojů a také jejich složitějšímu ovládání. Proto jsou hledány způsoby jak ulehčit práci s těmito zařízeními. Práce představuje řídicí systém navržený pro ovládání prezentačních prvků v učebnách katedry řídicí techniky založený na mikrokontroléru AT90USB1287 řady AVR podporující přímé ovládání pomocí klávesnice a LCD i vzdálené ovládání z PC. Systém je schopen řídit pomocí sériového rozhraní a 0/1 výstupů až 16 zařízení. Hlavní část práce je praktického charakteru a představuje převážně tvorbu software pro mikrokontrolér a programu vzdáleného ovládání.

Abstract

Modern classroom thanks to modern presentation instruments extends possibilities of teaching, but increases demands for service because number of device is high. Therefore we are searching for way how to simplify control of this device. This work is presents control system designed to control presentation device in classrooms of Department of Control Engineering, based on microcontroller AT90USB1287. System can be controlled under keyboard or remote control. Control system is able to control up to 16 other devices that are connected via serial port or 0/1 output. The biggest part of diploma is practical work based on programming software for microcontroller and for remote control system.

Obsah

Seznam obrázků **VII**

Seznam tabulek **IX**

Seznam ukázek **IX**

1 Úvod
1.1 Motivace …….. 1
1.2 Cíle práce……. 1

2 Analýza a návrh řešení …….. 2
2.1 Videomatice …….. 2
2.1.1 Karmer…….. 2
2.1.2 Extron …….. 3
2.1.3 Veemux……. 4
2.1.4 Zhodnocení …….. 5
2.2 Projektory …….. 5
2.2.1 Epson …….. 5
2.2.2 Sharp …….. 6
2.2.3 Optoma…….. 7
2.2.4 Panasonic ……... 8
2.2.5 Zhodnocení ……... 8
2.3 Ostatní ovládaná zřízení……... 9
2.4 Mikrokontrolér ……... 9
2.4.1 AT90USB1287 ……... 10
2.5 Ovládání ……... 11
2.6 USB ……... 13
2.6.1 ……... 14
2.6.2……..... 14

3. Realizace
3.1 LCD ……... 16
3.1.1 Hardware …… 16
3.1.2 Software ……. 17
3.2 Klávesnice ……... 19
3.2.1 Hardware ……... 19
3.2.2 Software …….. 20
3.3 Sériová komunikace RS232 …… 21
3.3.1 Hardware …… 21
3.3.2 Software …….. 22
3.4 0/1 výstupy …… 23
3.4.1 Hardware ……. 23
3.4.2 Software …….. 24
3.5 USB …… 25
3.5.1 Hardware …… 25
3.5.2 Software …….. 27
3.6 Program vzdáleného ovládání …… 28

3.7 Celková struktura 30
3.7.1 Hardware 31
3.7.2 Struktura menu 34
3.8 Knihovny ovládaných zařízení 37
3.8.1 Videomatice 37
3.8.2 Projektor 39
3.8.3 Editovatelná zařízení 39

4. Ovládání a testy **41**
4.1 Ovládání 41
4.2 Testování a uvedení doprovodu 41
4.2.1 Periferie a výstupy 41
4.2.2 Testy celku 41

5. Závěr **43**

6. Literatura **44**

A. Obsah CD **46**

Seznam Obrázků

Obrázek 2.1 Doporučené propojení pro videomatici Kramer ……. 2
Obrázek 2.2 Doporučené propojení pro Videomatici extron ……. 4
Obrázek 2.3 Doporučené propojení pro videomatici Veemux ……. 4
Obrázek 2.4 Doporučené propojení počítače s projektorem Sharp pomocí RS-232 …... 6
Obrázek 2.5 USB řadič v mikrokontroléru AT90USB1287 ……. 10
Obrázek 2.6 Blokové schéma - I/O porty AT90USB1287 ……. 11
Obrázek 2.7 Maticová klávesnice ……..……..……. 12
Obrázek 2.8 Mechanická vrstva USB rozhraní ……..…… 14
Obrázek 2.9 Konektor typu A , konektor typu B ……..…… 14

Obrázek 3.1 Propojení LCD ……..……..……. 17
Obrázek 3.2 Čtyřbitová komunikace LCD……..……..…… 18
Obrázek 3.3 Propojení klávesnice ……..……..……. 20
Obrázek 3.4 Propojení AVR s GAL a MAX232 ……..……... 22
Obrázek 3.5 Propojení AVR se vstupy 0/1 ……..……... 24
Obrázek 3.6 Propojení USB s AVR ……..……..……. 25
Obrázek 3.7 Přehled USB Device přerušení……..……... 26
Obrázek 3.8 Přehled USB ENDPOINT přerušení ……..……... 27
Obrázek 3.9 Hlavní menu v programu vzdáleného ovládání ……. 29
Obrázek 3.10 Blokové schéma řídicího systému ……..……... 30
Obrázek 3.11 Schéma řídicího systému ……..……... 31
Obrázek 3.12 Návrh desky plošného spoje…… 32
Obrázek 3.13 Deska řídicího systému ……..……... 33
Obrázek 3.14 Ladící deska ……..……..……. 33
Obrázek 3.15 Diagram znázorňující strukturu menu ……..…… 35

Seznam tabulek

Tabulka 2.1 Struktura příkazů videomatice Kramer 2
Tabulka 2.2 Tabulka příkazů pro projektory Epson 6
Tabulka 2.3 Struktura příkazu projektoru Epson 7
Tabulka 2.4 Struktura odpovědi projektoru Epson 7
Tabulka 2.5 Příklad komunikace projektoru Epson 7
Tabulka 2.6 Struktura příkazu projektoru Optoma 7
Tabulka 2.7 Struktura příkazu s parametrem projektoru Panasonic 8
Tabulka 2.8 Struktura příkazu bez parametru projektoru Panasonic 8
Tabulka 2.9 Chybová hláška projektoru Panasonic 8
Tabulka 2.10 Přehled některých modelů řady AVR..... 9
Tabulka 2.11 Popis zapojení LCD 12
Tabulka 2.12 instrukční sada pro LCD s řadičem HD44870 13

Seznam ukázek

Ukázka 3.1 Zapsání dat to řadiče LCD 18
Ukázka 3.2 Funkce zapisující polovinu dat LCD..... 19
Ukázka 3.3 Nalezení stisknutého tlačítka Klávesnice 20
Ukázka 3.4 Přiřazení pomocného čísla stisknuté klávese 21
Ukázka 3.5 Ukázka programu pro hradlové pole GAL 23
Ukázka 3.6 Nastavení stavů na 1/0 portech 24
Ukázka 3.7 Zpracování USB přerušení 28
Ukázka 3.8 Tvorba menu 34
Ukázka 3.9 Tvorba příkazu pro videomatici 38
Ukázka 3.10 Sekvence pro start Projektoru 39
Ukázka 3.11 Funkce pro start editovatelného zařízení 40

1. Úvod

1.1 Motivace

Učebny na katedře řídicí techniky jsou vybaveny velkým množstvím prezentačních pomůcek (např. projektory a možnost propojit je s různými vstupními zařízeními) a jiných ovládatelných zařízení zvyšující komfort výuky, ale kladoucí stále vyšší nároky na ovládání, proto vznikl tento projekt, který si klade za cíl navrhnout vhodný řídicí systém, jenž by ulehčoval práci s těmito zařízeními díky možnosti je ovládat z jednoho místa a možnosti přednastavit si nejčastější scénáře. Aby bylo možné tento systém navrhnout, je nutné vhodně zvolit základní ovládací prvek, v našem případě se jevil jako vhodný mikrokontrolér, ovšem s dostačujícími parametry. Volbu modelu omezily základní minimální vlastnosti jako je potřeba uchování scénářů v paměti i po tom co bude systém odpojen od napájení, schopnost komunikovat jednoduchým způsobem s PC (pro vzdálené ovládání) a dostatečná paměť pro program.

1.2 Cíle práce

Cílem této diplomové práce je návrh a realizace vhodného řídicího systému splňujícího parametry uvedené v zadání diplomové práce. Ve své podstatě tato práce vyžaduje provést tyto základní úkony:

- Prostudování komunikačních protokolů pro data projektory, videomatice, případně jiná připojitelná zařízení a srovnání jejich možností.
- Návrh vhodného HW pro řídicí systém a v souvislosti s ním vhodný způsob řízení až 16ti výstupů, vhodný prvek pro vizuální kontrolu a způsob propojení s PC.
- Realizace SW řídicího systému včetně vhodných scénářů.

Hlavní váhu práce nese realizační část, která se dělí na dva hlavní úkoly a to návrh vhodného HW splňujícího všechna specifika a tvorbu SW pro zvolený mikrokontrolér. Tvorba SW bude obzvlášť náročná, díky rozsahu a různorodosti ovládaných zařízení. Proto je obzvláště nutné provést důkladnou analýzu a zvolit vhodné hardwarové součásti a programovací nástroje.

2. Analýza a návrh řešení

2.1 Videomatice

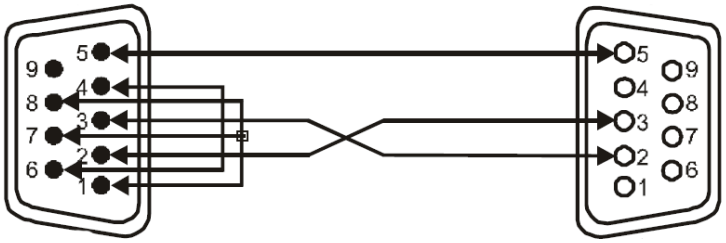
Jako jedno z hlavních připojitelných zařízení je videomatice, mezi hlavní parametry jenž ovlivní náš výběr, patří vlastnosti komunikačního rozhraní, možnosti protokolu a množství funkcí. Proto bylo nutné udělat průzkum videomatic hlavních výrobců.

1.2.1 Kramer

Videomatice Kramer používají pro komunikaci rozhraní RS232 nebo RS485. Pro naše účely je nejvhodnější rozhraní RS232, přes které komunikuje videomatice pomocí protokolu Kramer protokol 2000. Tento protokol používá 4 byte informaci. tyto byte jsou definovány níže.

Propojení a komunikace:

Videomatice se připojuje k ovládacímu zařízení přes konektor RS-232. Jak je vidět na obrázku mezi sebou propojeny piny 2 na 3, 3 na 2 a 5 na 5. Dále jsou mezi sebou propojeny piny 8,7,1 a pin 6 je propojen s pinem 4. Rychlost přenosu dat je 9600 baud, s 8-mi datovými bity, 1 stop bit, žádná parita a řízení toku.



Obrázek 2.1 Doporučené propojení pro videomatici Kramer

Kramer communication protokol 2000:

	CÍL	INSTRUKCE					
0	D	N5	N4	N3	N2	N1	N0
7	6	5	4	3	2	1	0

	VSTUP						
1	I6	I5	I4	I3	I2	I1	I0
7	6	5	4	3	2	1	0

	VÝSTUP						
1	O6	O5	O4	O3	O2	O1	O0
7	6	5	4	3	2	1	0

			ČÍSLO ZAŘÍZENÍ				
1	OVR	X	M4	M3	M2	M1	M0
7	6	5	4	3	2	1	0

Tabulka 2.1 Struktura příkazů videomatice kramer

První byte: bit 7 – definován jako 0.
D – Cíl : 0 – posílání informací z PC do videomatice
1 - posílání informací z videomatice do PC
N5…N0 – Zde je zapsáno číselné označení každé funkce kterou vykoná videomatice.

Druhý byte: bit 7 – definován jako 1.
I6…I7 - Při přepínání tyto bity obsahují číslo vstupu na který je přepínáno.

Třetí byte : bit 7 – definován jako 1.
O6…O7 - Při přepínání tyto bity obsahují číslo výstupu na který je přepínáno.

Čtvrtý byte: Bit 7 – definován jako 1.
Bit 6 - OVR .
Bit 5 – není využitý
M4…M0 – číslo přístroje.

Čtvrtý byte je používán pro adresování při ovládání více přístrojů přes jeden sériový port. V případě že je OVR nastaveno na jedna, tak všechny přístroje akceptují příkaz a odpovída jen jeden označený v sekci M4…M0. V případě že ovládáme jen jeden přístroj je M4…M0 vždy 1.

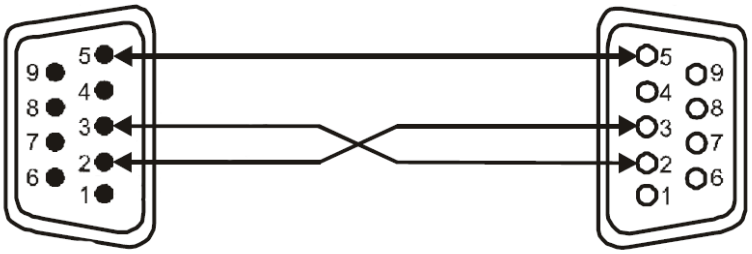
2.1.2 Extron

Videomatice firmy Extron požívají pro externí ovládání také rozhraní RS232 využívající pro komunikaci protokol SIS(Simple Instruction Set), případně lze využít ovládací program pro Windows.

Propojení a komunikace:

Videomatice obsahuje klasický RS-232 konektor, využity jsou však pouze piny 2 (TX), 3 (RX) a 5 (GND). Ostatní piny by měli být z důvodu spolehlivosti odpojeny.

Protokol RS232 je 9600 baud, 8-bit, 1 stop bit, žádná parita a řízení toku.



Obrázek 2.2 Doporučené propojení pro Videomatici extron

SIS protokol:

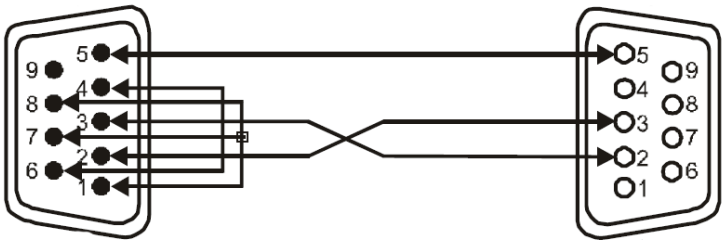
SIS příkazy jdoucí směrem z PC do videomatice se skládají z jednoho nebo více znaků. Není zde vyžadováno žádných speciálních znaků nebo jejich sekvence. Každá videomatice odpovídá na konec SIS příkazu novým řádkem a posunem o řádek (CR/LF). Každý příkaz je reprezentován nějakým znakem a každý znak je převeden na hexadecimální číslo pomocí tabulky ASCII tabulky.

Příklad:
Propojení má znak “*“ a nastavení audio i video má znak “!“. Pro propojení vstupu 1 na výstup 3 tedy použijeme syntaxi “ 1*3! “ Odpověď videomatice do PC poté bude “Out3.In1.All“.

2.1.3 Veemux

Propojení

Videomatice Veemux se připojuje k ovládacímu zařízení přes konektor RS-232 . Jak je vidět na obrázku mezi sebou propojeny piny 2 na 3, 3 na 2 a 5 na 5. Dále jsou mezi sebou propojeny piny 8,7,1 a pin 6 je propojen s pinem 4. Rychlost přenosu dat je 9600 baud, s 8-mi datovými bity, 1 stop bit, žádná parita .



Obrázek 2.3 Doporučené propojení pro videomatici Veemux

Komunikační protokol:

Příkazy do videomatice se skládají z jednoho nebo více znaků. Není zde vyžadováno žádných speciálních znaků nebo jejich sekvence.Všechny příkazy musí být ukončeny <CR>(carriage return) všechny příkazy musí být psány velkými písmeny a všechna čísla nižší než 10 musí mít tvar 0X (např. pro 1 je to 01). Když je příkaz poslán, je celý zopakován s odpovědí od videomatice. Prodleva mezi příkazy musí být minimálně 500ms. Příkazy jsou definovány ve speciální tabulce.

Příklad:

Příkaz :CS SW, IP,OP odpověď: *<CR>

CS – znamená propojení jednoho vstupu k jednomu výstupu
SW – je adresa jednotky
OP - číslo výstupního portu
IP – číslo vstupního portu

2.1.4 Zhodnocení

Všechny videomatice používají rozhraní RS232 a jen Kramer používali navíc RS485, nastavitelnost parametrů je u všech videomatic také na podobné úrovni, však zpracování dokumentace a přehlednost protokolu byla nejlepší u videomatic Kramer, proto se jeví jako nejvhodnější.

1.3Projektory

2.2.1 Epson

Projektory Epson jsou ovládány pomocí protokolu ESC/VP21. Příkazové znaky jsou složeny z ASCII kódu a proto může být projektor jednoduše ovládán. Protože je ESC/VP21 nezávislý protokol, může být použit přes rozhraní Seriál, USB a podobně. Komunikace probíhá tak že PC vyšle nějaký příkaz zakončený ukončujícím znakem CR a čeká na odezvu, ta je buď ":" nebo chybová hláška "ERR"následována ":".

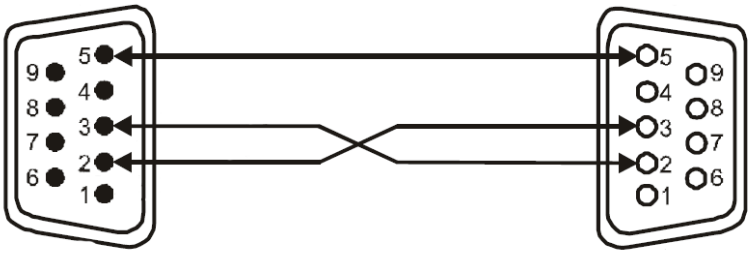
příklad komunikace - : SOURCE 21 (0DH)

Položka		Příkaz
Zapnutí a vypnutí napájení	ON	PWR ON
	OFF	PWR OFF
Volba signálu	Video	SOURCE 41
	S-Video	SOURCE 42
	HDMI (pouze EB-W6)	SOURCE 30
	USB Display	SOURCE 51
	Prezentace (pouze EB-W6)	SOURCE 52
A/V Mute ON/ OFF (Aktivace a deaktivace Vypnutí A/V)	ON	MUTE ON
	OFF	MUTE OFF
Volba Vypnout A/ V	Černá	MSEL 00
	Modrá	MSEL 01
	Logo	MSEL 02
Položka		**Příkaz**
		Port počítače
Volba signálu	Počítač (autom.)	SOURCE 1F
	Počítač	SOURCE 11

Tabulka 2.2: Tabulka příkazů pro projektory Epson [8]

2.2.2 Sharp

Projektory Sharp využívají obvykle rozhraní RS-232, nastavení komunikace je 9600 baud, 8-bit, 1 stop bit, žádná parita a žádné řízení toku. Propojení je na následujícím obrázku.



Obrázek 2.4 Doporučené propojení počítače s projektorem Sharp pomocí RS-232

Příkazy z PC do projektoru jsou složeny ze dvou čtyřbytových částí, první je příkaz a druhá je parametr, ukončení příkazu vytvoříme pomocí návratového znaku (0DH)

C1	C2	C3	C4	P1	P2	P3	P4	Návratový znak (0DH)

Tabulka 2.3 Struktura příkazu.

Odpověď od projektoru je bud OK pro potvrzení vykonání příkazu, nebo ERR pro chybu při vykonávání. Oba příkazy opět následuje návratový znak (0DH).

O	K	Návratový znak (0DH)
E	R	R	Návratový znak (0DH)

Tabulka 2.4 Struktura odpovědi.

Každý projektor může vykonávat skupinu specifických příkazů, jenž jsou uvedeny v tabulkách . Následující příklad ukazuje nastavení jasu na -10.

R	A	B	R	_	-	1	0	(0DH)	<	>	O	K	(0DH)

Tabulka 2.5 Příklad komunikace projektoru.

2.2.3 Optoma

Projektory Optoma využívají rozhraní DIN na RS232, nastavení komunikace je 9600 baud, 8-bit, 1 stop bit, žádná parita a žádné řízení toku. Každý příkaz vysílaný do projektoru začíná znakem “~”, následuje řetězec příkazu, který je malé písmeno a řetězec dat, který je velké písmeno, vše je ukončeno znakem CR. V následující tabulce je zobrazena struktura příkazu. V případě že není příkaz přijat korektně, projektor neodpoví. V případě, že byl přijat korektně projektor odešle odpovídající hodnotu.

	Hlavička	Řetězec příkazu	Řetězec dat	Ukončovací řetězec
HEX	7Eh	příkaz	data	0Dh
ASCII	~	příkaz	data	CR

Tabulka 2.6 Struktura příkazu

Příklad příkazu a odpovědi :
Dotaz na jas : ~q B (CR)
Odpověď: 100

2.2.4 Panasonic

Projektory Panasonic využívají klasické rozhraní RS232, rychlost komunikace 9600 baud, 8bit, 1 stop bit, bez parity a řízení toku. Každý příkaz vysílaný do projektoru začíná a končí speciálním znakem. Jsou dva druhy příkazů a to s parametrem (Tabulka 2.7) a bez parametru.(Tabulka 2.8)

S parametrem

	Hlavička	Řetězec příkazu	Oddělovač	Řetězec dat	Ukončovací řetězec
HEX	02h	příkaz	3Ah	data	03h
ASCII		příkaz	:	data	

Tabulka 2.7 Struktura příkazu s parametrem

Bez parametru

	Hlavička	Řetězec příkazu	Ukončovací řetězec
HEX	02h	příkaz	03h
ASCII		příkaz	

Tabulka 2.8 Struktura příkazu bez parametru

Odpověď v případě že se příkaz proběhl v pořádku je, ve tvaru zopakování příkazu. V případě špatného příkazu, místo zopakování příkazu vyšle chybovou hlášku ERR401.

	Hlavička	Řetězec příkazu					Ukončovací řetězec
HEX	02h	45h	52h	34h	30h	31h	03h
ASCII		E	R	4	0	1	

Tabulka 2.9 Chybová hláška

2.2.5 Zhodnoceni

V učebnách řídicí katedry jsou obvykle projektory Panasonic a Epson, jenž mají protokoly i rozhraní v dostačující kvalitě, a proto bude ovládání omezeno na tyto dva typy projektorů.

1.4 Ostatní ovládaná zařízení

Ostatní zařízení, která bude muset řídicí systém ovládat se dělí na dva druhy podle rozhraní jednodušší je typ který se spíná prostým přivedením úrovně jedna na vstup (např. žaluzie nebo plátno), tento druh zařízení je snadno ovladatelný a implementace do systému nebude obtížná. Další zařízení je například nový projektor nebo videomatice atp. s kterými není v kódu počítáno, přidání těchto zařízení bude nutno ošetřit pomocí editovatelných zařízení, jenž budou mít editovatelné příkazy.

2.4 Mikrokontrolér

Podle nároků kladených na řídicí systém je nutno vybrat platformu, na které bude problém řešen, jako nevhodný se jevil systém založený na osobním počítači PC a to hned z několika důvodů, jeho vysoká cena, vysoká spotřeba, přes vysokou cenu nutnost dořešení vstupů a výstupů. Jako další a nejvhodnější možnost se jeví řešení pomocí mikrokontroléru, zde jsou dva hlavní výrobci a to Mikrochip a jeho PIC a Atmel s řadou 8051 a AVR. Zvolena byla firma Atmel pro zkušenosti s mikrokontroléru řady 8051. Zde však musíme vzít v potaz nároky na vstupně výstupní porty a paměť. Zařízení musí ovládat 10 zařízení na RS232, USB, klávesnici, LCD a cca 6 zařízení ovládaných úrovní a musí být schopno uchovat v paměti scénáře i po vypnutí. Tyto parametry vedly k řadě procesorů AVR od firmy Atmel.

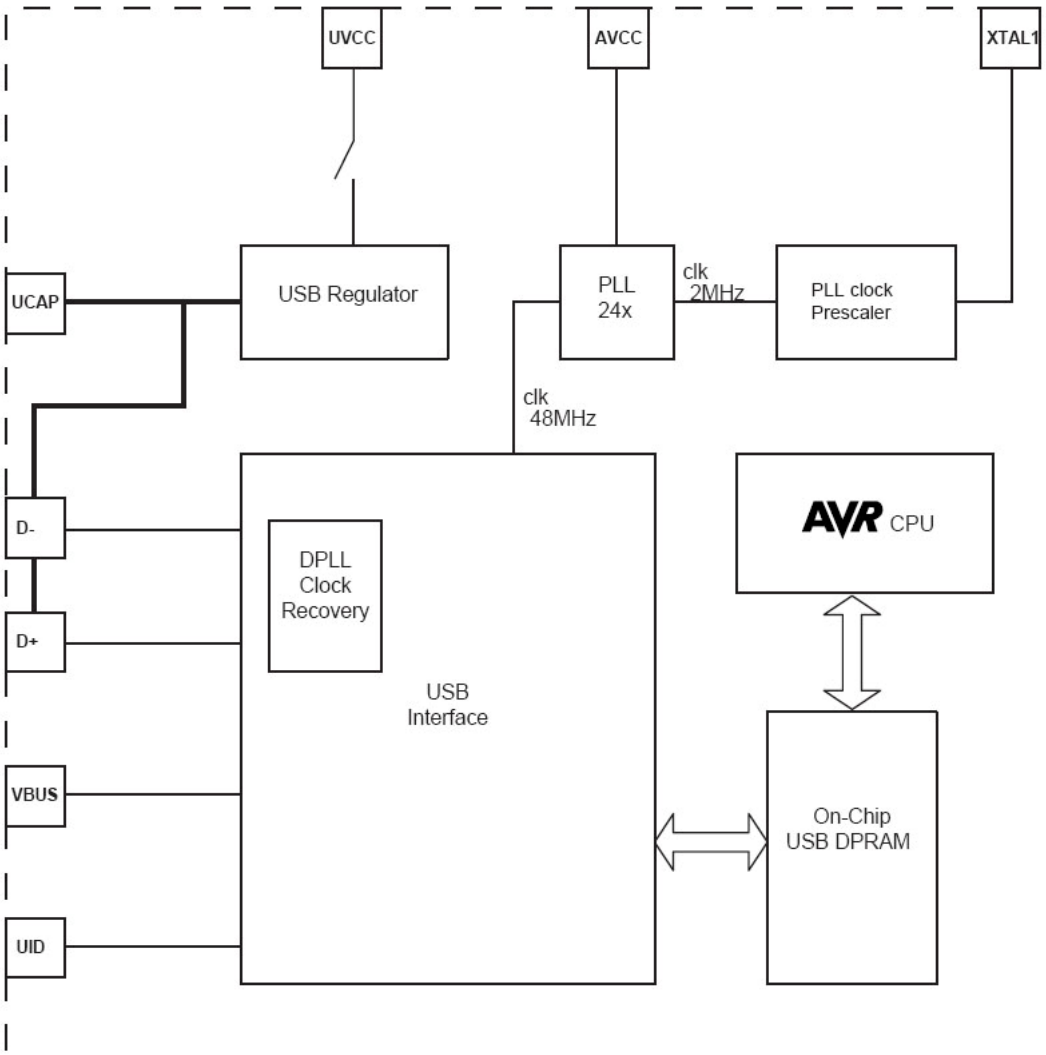
Rodina mikrokontrolérů AVR jsou jednočipové mikrokontroléry, založené na harwardovské architektuře s instrukční sadou RISC (redukovanou). Jejich hlavní výhodou je efektivní instrukční sada, kde převážná většina instrukcí trvá 1 hodinový cyklus, což při frekvenci až 16 Mhz znamená velice vysoký výkon. Na rozdíl od nižší řady má navíc paměť eeprom přepisovatelnou programem schopnou uchovat informaci i po odpojení napájení, kromě standardních pamětí, kterými jsou paměť flash přepisovatelná jen pomocí programátoru, používanou pro samotný program a paměť SRAM, jenž je rychlá paměť používaná při výpočtech. AVR se vyrábí v mnoha modifikacích, počínaje levnou ATiny, draží a vybavenější ATmega až po modely zaměřené na určitou speciální schopnost jako například AT90USB1287, který má implementovanou technologii USB.

MODEL	FLASH	SRAM	EEPROM	I/O piny	Maximální frekvence
AT90S8535	8K	512	512	32	8 MHz
AT90C8534	8K	256	512	7	1.5MHz
AT90S4414	4K	256	256	32	8 MHz (4.0-6.0V)
AT90S8515	8K	512	512	32	8 MHz (4.0-6.0V)
ATmega128	128K	4K	4K	32+8+8	16 MHz
ATmega64	64K	2K	4K	32+8+8	16 MHz
ATmega16	16K	1K	512	35	8MHz
ATtiny10	1K	No	No	5+1+0	16 MHz
ATtiny11	1K	No	No	5+1+0	16 MHz
ATtiny12	1K	No	64	6	8 MHz
ATtiny22	2K	128	128	5	8 MHz
AT90USB1287	128K	8K	4K	32+8+8	16 Mhz
Atmega32	32K	2K	1K	32	

Tabulka 2.10 Přehled některých modelů řady AVR

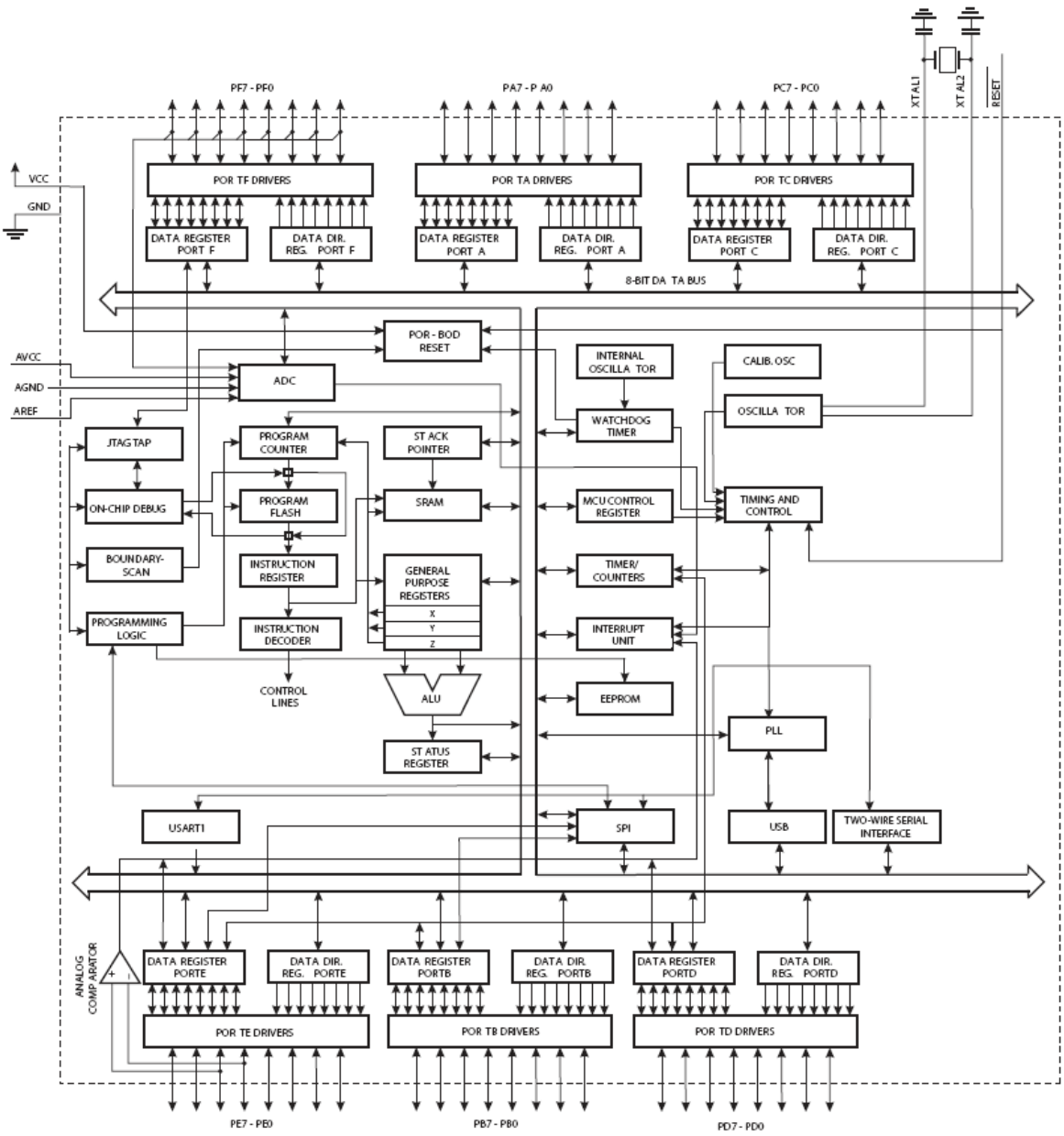
2.4.1 AT90USB1287

Mikrokontrolér AT90USB1287 je z mladší řady mikrokontrolérů AVR (ARM7), jeho hlavní výhodou je že má integrovaný hardwarový USB řadič, k němuž je přistupováno pomocí I/O adresového rozsahu, dále je vybaven pamětí DPRAM do které jsou ukládána přijatá a odeslaná data. USB sběrnice je efektivně řízena pomocí přerušení, které je schopen mikrokontrolér zaznamenat i v úsporném režimu a je tedy možno mikrokontrolér z tohoto režimu probudit.



Obrázek 2.5 USB řadič v mikrokontroléru AT90USB1287 [1]

Dále je mikrokontrolér vybaven 128KB programové paměti flash, 8KB rychlé paměti SRAM a 4KB paměti EEPROM. Instrukční sada obsahuje 135 Instrukcí, z nichž většina trvá jeden hodinový cyklus. Maximální frekvence je 16Mhz, což umožňuje v ideálním případě až 16MIPS. Mikrokontrolér obsahuje také dostatek vstupně/výstupních portů označených v sadách po 8mi zkratkami PA - PF z nichž většina má ještě další funkci.

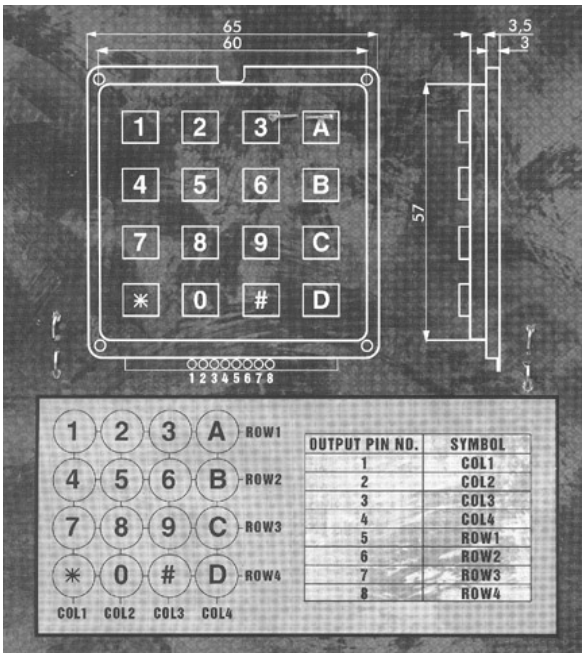


Obrázek 2.6 Blokové schéma - I/O porty AT90USB1287 [1]

2.5 Ovládání

Řídicí systém musí obsahovat dvě úrovně ovládání a to úroveň uživatelskou a úroveň administrátorskou, uživatelská úroveň ovládání musí být dostupná jen přímo z místa, kde je umístěn řídicí systém, zatímco požadavek na administrátorské ovládání je vzdálená kontrola minimálně z místností sousedící s místností ovládanou.

Přímé ovládání se bude realizovat pomocí maticové klávesnice, pro kterou je nutné vytvořit jednoduchý ovladač.



Obrázek2.7 Maticová klávesnice [6]

Pro vizuální kontrolu je použit LCD display, po uvážení byl použit LCD s řadičem HD44870. Tyto displaye se dají poměrně jednoduše ovládat to bud 4bitovou, nebo 8 mi bitovou metodou a jsou za příznivou cenu.

Vývod	Název	Funkce
1	Vss	GND
2	Vcc	napájení +5V
3	Vee, V0	nastavení kontrastu
4	RS	volba mezi 0 - instrukce, 1 - data
5	R/W	volba mezi 0 - zápis, 1 - čtení
6	E	hodinový vstup
7	DB0	data 0
8	DB1	data 1
9	DB2	data 2
10	DB3	data 3
11	DB4	data 4
12	DB5	data 5
13	DB6	data 6
14	DB7	data 7
15	LED+	anoda podsvětlení
16	LED-	katoda podsvětlení

Tabulka 2.11 Popis zapojení LCD

Při osmibitové komunikaci jsou využity všechny datové vodiče. Při čtyřbitové komunikaci jsou datové vodiče DB0-3 obvykle uzemněny a využívány jsou jen vodiče DB4-7 zde inicializace funguje tak že se vyšle nejdříve první polovina inicializačního bytu, čímž se display nastaví na 4 bitovou komunikaci a poté se pošle celý inicializační byt 4- bitovou metodou, která funguje tak že se nejdříve pošle horní polovina bytu a poté se na hodinový vstup nastaví kladná hodnota a poté se pošle spodní polovina bytu a na hodinový vstup se nastaví opačná hodnota. LCD se ovládá pomocí jednoduchých instrukcí (Tabulka2.12) a můžeme ve něm vytvořit až 8 vlastních znaků.

Význam instrukce	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Délka
smaže disp. a nastaví kurzor na začátek	0	0	0	0	0	0	0	0	0	1	1,64 ms
nastaví kurzor na začátek	0	0	0	0	0	0	0	0	1	x	1,64 ms
směr posuvu kurzoru I/D (0=vlevo, 1=vpravo), posuv textu S (0=ne, 1=ano)	0	0	0	0	0	0	0	1	I/D	S	40 us
D - zapne displej, C - zapne kurzor,B - zapne blikání kurzoru	0	0	0	0	0	0	1	D	C	B	40 us
1x posune (S/C=0 kurzor, S/C=1 text) směrem (R/L=0 vlevo, R/L=1 vpravo)	0	0	0	0	0	1	S/C	R/L	x	x	40 us
inicializace: DL=0 4-bit, DL=1 8-bit mód, N=0 jednořádkový, N=1 dvouřádkový disp. F=0 font 5x8, F=1 font 5x10	0	0	0	0	1	DL	N	F	x	x	40 us
přepnutí na zápis do CGRAM	0	0	0	1	adresa v CGRAM						40 us
přepnutí na zápis do DDRAM	0	0	1	adresa v DDRAM							40 us
čtení příznaku BF (BF=0 příjem povolen, BF=1 řadič zaneprázdněn), čtení adresy v DDRAM	0	1	BF	adresa v DDRAM							0 us
zápis dat do CGRAM nebo DDRAM	1	0	data								40 us
čtení dat z CGRAM nebo DDRAM	1	1	data								40 us

Tabulka 2.12 instrukční sada pro LCD s řadičem HD44870

2.6 USB

USB se stala standardem u veškeré spotřební a částečně i průmyslové elektroniky, pro jeho rychlost a možnosti vytlačilo klasické rozhraní RS232 rozhraní, proto je použito i v přípravku pro vzdálené ovládání.

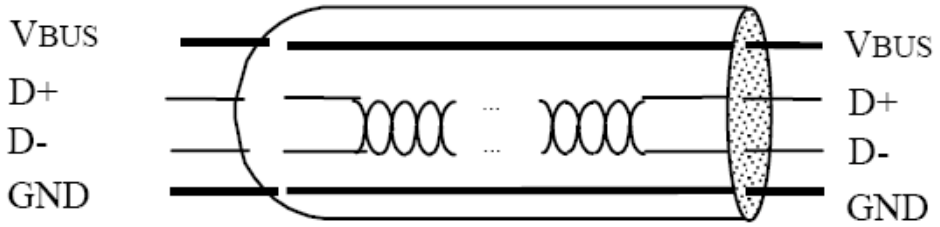
Základní vlastnosti USB rozhraní:

- možnost připojit až 127 zařízení k jednomu kořenovému přepínači
- teoretická rychlost 1,5 Mbaud (low speed) až 450 Mbaud (High speed)
- schopnost napájet připojená zařízení s maximálním proudem 500mA přes vodiče Vbus a GND

- topologie je stromového typu, na každém uzlu je připojen USB hub neboli přepínač, tyto přepínače mohou být propojovány jeden do druhého.
- Transportní vrstva je řešena pomocí roury mezi hostitelem a zařízením, celé přenosné médium je transparentní.

2.6.1 Mechanická část

Diferenciální vedení je realizováno kroucenou dvoulinkou která má ženěný plášť (obrázek 2.8).



Obrázek 2.8 Mechanická vrstva USB rozhraní [14]

Konektory na straně hostitele a zařízení se liší. NA straně hostitele je konektor typu A a na straně zařízení je konektor typu B.(Obrázek 2.9)

Obrázek 2.9 Konektor typu A , konektor typu B [14]

2.6.2 Datová část

Základní přenášené datové struktury jsou pakety, maximální počet bytů v paketu je pro endpoint 0 8,16,32 nebo 64 bytu, ostatní endpointy mohou volit velikost v Endpoint descriptoru. Všechna data jsou tedy rozdělována na pakety a opatřena hlavičkou. Endpoint 0 je obousměrný a obstarává počáteční vyjednávání, další endpointy mohou být nastaveny podle potřeby na příchozí nebo odchozí. Pro potřeby komunikace existuje 16 druhů paketů které jsou rozděleny do čtyřech skupin.

potvrzovací - , NYET, ACK, NAK, STALL

pověřovací - IN, OUT, SETUP, SOF

datové -MDATA, DATA0, DATA1, DATA2

speciální - ERR, PREamble, Split, Ping

Nutné schopnosti zařízení:

- Odeslat Device Descriptor
- Odeslat Configuration Deskriptor
- Odeslat HDI descriptor
- Přijmout příkaz Set Address
- Přijmout příkaz Set Configuration
- Přijmout data pomocí posloupnosti OUT a DATA paketu

3. Realizace

Část Realizace podrobně rozebírá všechny části zařízení, které byly v rámci této DP realizovány. Tato část je rozdělena na několik bodů, bod *Struktura systému* popisuje zařízení jako celek, další body se věnují zvlášť každému připojitelnému zařízení a rozebírá softwarovou a hardwaru část každé připojené periferie.

3.1 LCD

MC2004B je jednoduchý znakový display s řadičem HD-44870 s rozlišením LCD 20x4 znaků. Jeho výhodou je snadná obsluha a například možnost čtení pozice kurzoru. Veškeré obslužné funkce pro řadič obsahuje knihovna lcdovl.h, která obstarává veškerou nízkoúrovňovou komunikaci s displayem a uživateli poskytuje funkce pro inicializaci, základní nízkoúrovňové funkce pro zasílání a čtení dat a vysokoúrovňové funkce, které za využití nízkoúrovňových například nastavují pozici na displeji nebo vypíší znak.

3.1.1 Hardware

Display komunikuje s mikrokontrolérem pomocí 4 bitového interface, sestávající se z těchto vstupů a výstupů:

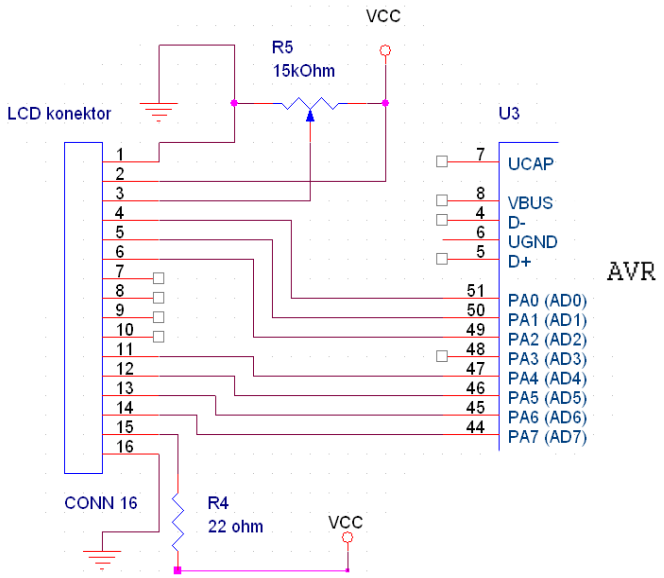
DATA [0-3] – datový vektor, na němž se nastavuje znak nebo příkaz

E – hodinový vstup, který oznamuje, jestli jsou na vstupech platná data

RW – určuje zda jsou na data na výstupech Data [0-3] příkaz nebo znak

RS – přepíná LCD mezi režimem čtení a zápisu

K mikrokontroléru je LCD připojen pomocí vstupně/výstupního portu PA (obrázek 3.1). Dále je připojeno napájení osvětlení a regulace kontrastu displaye, která je řešena pomocí potenciometru.



Obrázek 3.1 Propojení LCD

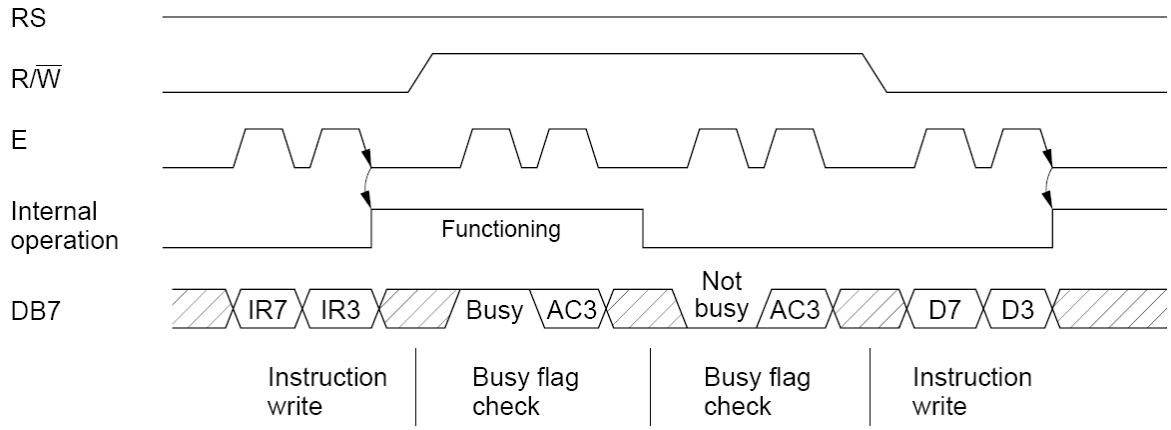
Display umožňuje i čtení aktuální pozice kurzoru,načtené hodnoty jsou však zakódovány jako u dvouřádkového displaye a proto se musí převést na čtyřřádkový. Pozice jsou očíslovány podle následujícího popisu:

1. řádek …. 00h až 13h
2. řádek …. 40h až 53h
3. řádek …. 14h až 27h
4. řádek …. 54h až 67h

3.1.2 Software

Řadič LCD dovoluje použití ve buďto ve čtyřbitovém módu nebo ve osmibitovém módu.U použitého čtyřbitového módu bývá častým problémem inicializace. Po prvním zapnutí napájení je totiž LCD nastaven na 8-bit komunikační mód. První instrukcí (ještě v 8-bit módu) dojde k přepnutí na 4-bit verzi. Vzhledem k tomu, že DB0-DB3 jsou na GND, tak LCD obdrží jen horní část init. bytu, dolní 4 bity jsou 0b0000. Následně, pokud potřebujeme nastavit i spodní čtveřici bitů inicializace, musíme celou inicializaci zopakovat tentokrát již ve 4-bit módu. Od teď už lze komunikovat jen ve 4-bit módu.
Další problém nastane při restartování samotného mikrokontroléru, zatímco LCD zůstává ve čtyřbitovém módu, tak mikrokontrolér se snaží zapsat první instrukci pomocí osmibitového.

Tento problém řeší složitější inicializace pomocí metody lcd_init() kde nejprve nastaví zpět osmibitová inicializace a poté opět čtyřbitová.



Obrázek 3.2 Čtyřbitová komunikace [13]

Vytvořené knihovny jsou inspirovány, jinými, již vytvořenými, nutnost novývh knihoven vzešla z potřeby komunikovat 4bitově a hlavně nutnosti číst pozici lcd, což většina knihoven neumí.Ovládání displaye obstarává několik jednoduchých funkcí, tyto funkce lze rozdělit na nízkoúrovňové a vysokoúrovňové.
Nízkoúrovňové funkce například zapisují lcd_writedata() nebo čtou (lcd_readdata() jeden byte přímo z registrů. Pomocí těchto funkcí jsou v programu pouze nastavovány vlastnosti kurzoru, a načítána pozice kurzoru.

```
void lcd_writedata(unsigned char data)
{
#asm
   cbi  0x02,1          ; nastavíme RD na nulu 0x02 je adresou pro port a u
použitého
   in    r26, 0x01      ; Mikrokonrtoléru
   ori   r26,0xf0 | (1<<0) | (1<<0) | (1<<2) ;nastavení jako výstup
   out   0x01,r26
   in    r27, 0x02
   andi  r27,0xf
   ld    r26,y
#endasm
   lcd_writepul ();          //zapis vyzšši
#asm
   ld    r26,y
   swap  r26
#endasm
   lcd_writepul ();          //write zapis nizsi
#asm
   sbi   0x02,1    ;RD=1
#endasm
}
```

Ukázka 3.1 Zapsání dat to řadiče LCD

V předchozím příkladě byla použita další funkce lcd_writepul(), pomocí které zapisuje půlku byte.

```
static void lcd_writepul(void)
{
#asm
   andi  r26, 0xf0
   or    r26,r27
   out   0x02,r26          ;zapis
   sbi   0x02,2 ;nastavi na  EN 1
   cbi   0x02,2 ;EN=0
#endasm

}
```

Ukázka 3.2 Funkce zapisující polovinu dat

Vysokoúrovňové funkce z velké části využívají nízkoúrovňové. Tyto funkce jsou:

lcd_init() – inicializace displaye, tato funkce nastaví LCD do módu čtyřbitové komunikace, nastaví kurzor na počáteční pozici vymaže display.

lcd_goto(x,y) – přesune kurzor na pozici určenou parametrem x a y, kde x je sloupec a y řádek.

lcd_putchar(char c) – zapíše na display znak uložený v "c" a poskoční na další pozici.

lcd_clear() – vymaže všechny znaky a nastaví kurzor na počáteční pozici

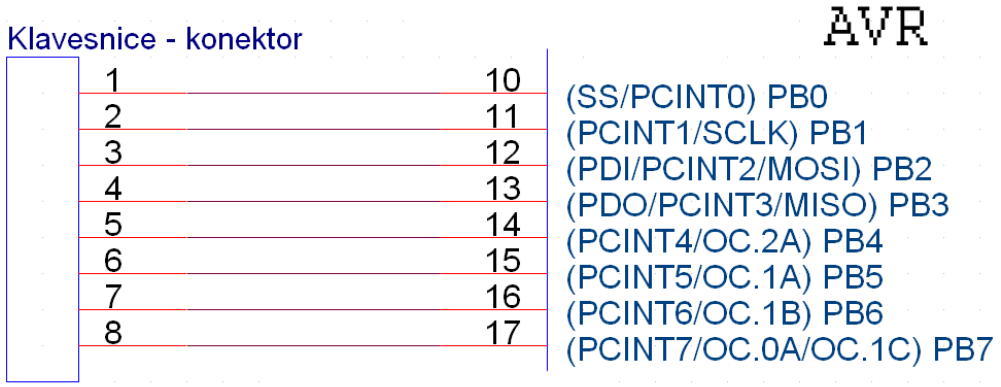
lcd_putfl() – vypíše na display řetězec umístěný ve flash paměti

3.2 Klávesnice:

Použita je standardní maticová klávesnice 4x4, maticová klávesnice byla zvolena díky poměrně jednoduché obsluze a zároveň dostatečnému počtu kláves při ovládání osmi vodiči.

3.2.1 Hardware

Klávesnice je k mikrokontroléru je připojena pomocí osmivodičového rozhraní na port PB mikrokontroléru, tak že 4 horní bity jsou připojeny na řádky a spodní 4 bity jsou připojeny na sloupce. (obrázek 3.3).



Obrázek 3.3 Propojení klávesnice

3.2.2 Software

Funkce pro čtení klávesnice jsou obsaženy v knihovně keyb.h, čtení z maticové klávesnice pomocí osmi vodičů je realizováno, tak že program postupně přivádí na 4 vodiče nastavené jako výstupní jedničku a na vstupních ověřuje, jestli je jednička přítomna (Ukázka 3.3) a pomocí této metody získáme vzájemným násobením číslo, které se pomocí funkce switch(), přiřadí konkrétnímu tlačítku.(Ukázka 3.4)

```
while(1)
{
upperNibble = 0xff;

for(i=0; i<4; i++)
{
delay_ms(5);
KB_PORT_OUT = ~(0x01 << i);
delay_ms(5);
upperNibble = KB_PORT_IN | 0x0f;

if (upperNibble != 0xff)
{
delay_ms(5);
upperNibble = KB_PORT_IN | 0x0f;
if(upperNibble == 0xff) goto OUT;

keyCode = (upperNibble & 0xf0) | (0x0f & ~(0x01 << i));

while (upperNibble != 0xff)
upperNibble = KB_PORT_IN | 0x0f;
```

Ukázka 3.3 Nalezení stisknutého tlačítka

```
switch (keyCode)

 {
case (0xee): keyPressed = '1';
break;
case (0xed): keyPressed = '2';
break;
case (0xeb): keyPressed = '3';
break;
case (0xe7): keyPressed = 'A';
break;
case (0xde): keyPressed = '4';
break;
case (0xdd): keyPressed = '5';
break;
        .
        .
        .
```

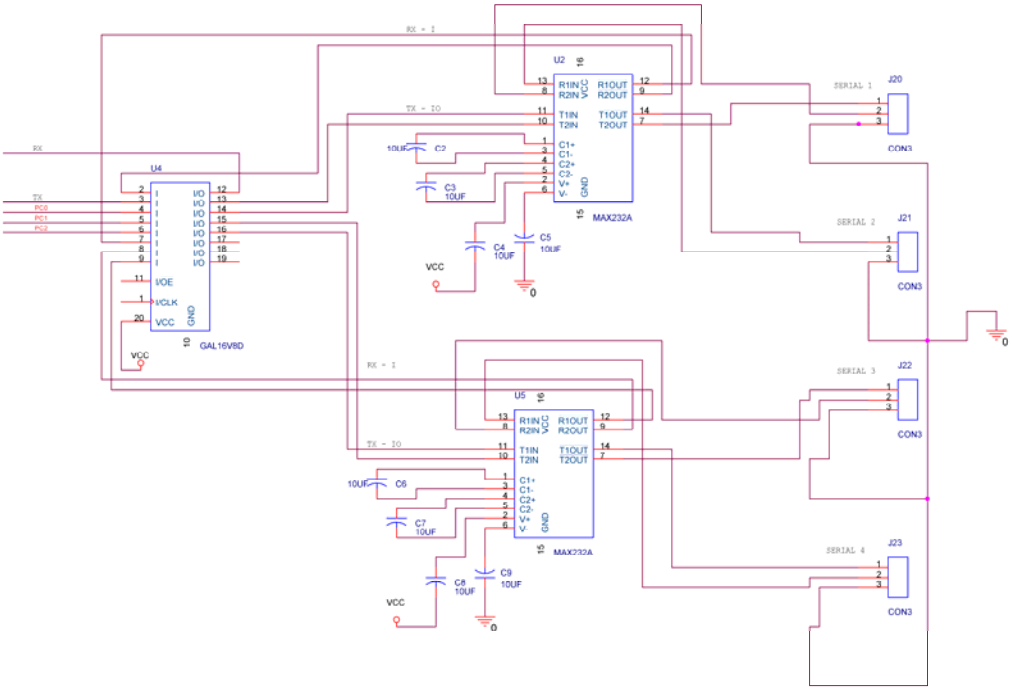
Ukázka 3.4 Přiřazení pomocného čísla stisknuté klávese

3.3 Sériová komunikace RS232

Sériové RS-232 bylo zvoleno jako hlavní komunikační rozhraní, hlavním problémem při realizaci bylo to že použitý mikrokontrolér používá pouze jedno sériové interface. Proto je v návrhu použito programovatelných hradlových polí jenž přepínají jeden sériový port z mikrokontroléru na maximálně 10 zařízení.

3.3.1 Hardware

Propojení sériového rozhraní je řešeno pomocí programovatelných hradel, použity byly tři chipy typu GAL16V8 které jsou dále propojeny na převodník úrovně MAX232 jenž převádí úroveň TTL na RS232. Obvody GAL jsou propojeny každý se sériovým interface na mikrokontroléru, dále na vstup vedou ještě vodiče pomocí nichž se ovládá propojení pole ke konkrétnímu výstupu. Pro tuto funkci musel být vyhrazen celý 8mi vodičový vstupně/výstupní port PC. (Obrázek 3.4) Osm vodičů bylo dostačujících protože první dva obvody GAL propojují vždy čtyři zařízení a tedy potřebují každý tři řídící vodiče a třetí GAL řídí jen dvě zařízení a proto stačí pro jeho ovládání vodiče dva.



Obrázek 3.4 Propojení AVR s GAL a MAX232

3.3.2 Software

Ovládání sériového rozhraní v mikrokontroléru je řešeno pomocí knihovnu io.h. Pro odesílání a přijímání znaků jsou k dispozici následující funkce:

putchar(char c) - vyšle znak uložený v proměnné c na sériové rozhraní .

getchar() - načte znak ze sériového rozhraní a uloží ho do námi zvolené proměnné.

printf(“xx“) – vyšle řetězec znaků napsaný mezi uvozovkami, který je při překladu uložený do paměti flash.

Protože je sériové rozhraní přepínáno až mezi 10 ti zařízeními, pomocí programovatelných hradlových polí GAL, musí být tato pole naprogramována dvě jsou programována, pro řízení pomocí třech vstupů a přepínají mezi čtyřmi zařízeními a třetí je ovládáno dvěma vodiči a řídí jen dva vstupy. Naprogramovány byly pomocí starého, ale dostačujícího programu ORCAD PLD .(Obrázek3.5)

Řídicí signály pro obvody GAL jsou vysílány pomocí portu PC ten je ovládán funkcí gal(int c) Jenž má za úkol v svislosti na zvoleném vývodu nastavit vhodný byte na výstupu PC. Proměnná c obsahuje číslo vývodu sériového rozhraní.

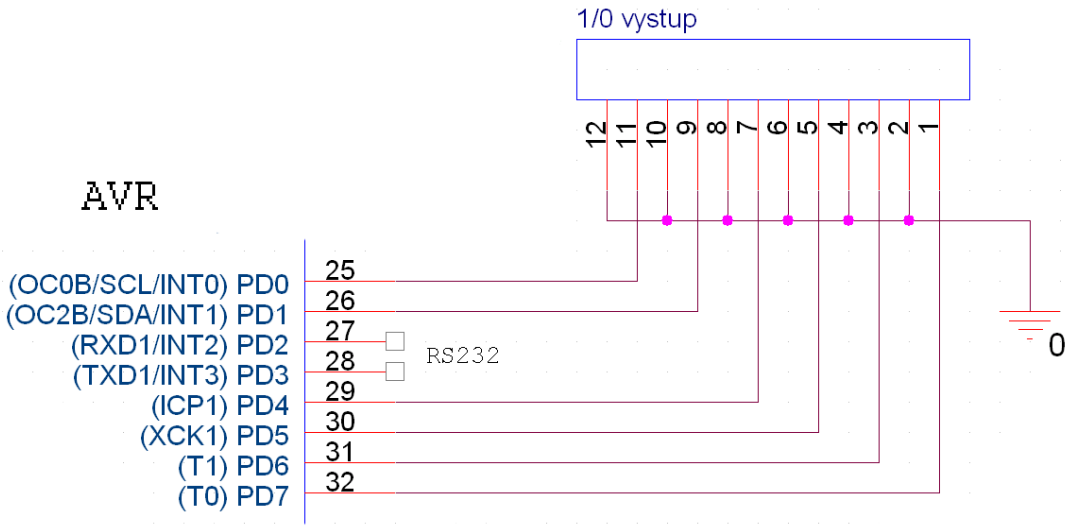
```
GAL16V8

| 2..9:(I[0..7]), 12..16:(O[0..4])

|O1=I1&I2'&I3'&I4
|O0=I0&I2'&I3'&I4

|O0=I5&I2'&I3&I4'
|O2=I0&I2'&I3&I4'

|O0=I6&I2'&I3&I4
|O3=I0&I2'&I3&I4

|O0=I7&I2&I3'&I4'
|O4=I0&I2&I3'&I4'

                    //Kód pro testovací program
|Vectors:
|{
|Display I0,I1,"   ",I2,I3,I4,"   ",I5,I6,I7,"->",O0,O1,O2,O3,O4
|Test I2,I3,I4,I0,I1,I5,I6,I7
|End
|}
```

Ukázka 3.5 Ukázka programu pro hradlové pole GAL

3.4 0/1 výsupy

3.4.1 Hardware

Tyto výstupy jsou určeny pro zařízení, jenž jsou zapínána připojením signálu TTL úrovně high. Výstupy jsou připojeny přímo na vstupně výstupní port PD mikrokontroléru. Pro ovládání šesti zařízení je použito 6 výstupů portu PD, konkrétně výstupy PD -0,1,4,5,6,7. (Obrázek 3.5)



Obrázek 3.5 Propojení AVR se vstupy 0/1

3.4.2 Software

Pro nastavení úrovně high na výstupech 0/1 je použit přímo port PD a proto není třeba vytvářet nové knihovny pro ovládání tohoto výstupu. U mikrokontroléru řady AVR se nastavují porty jako výstupní pomocí příkazu *DDRD.x = 1;* a samotné nastavení úrovně se děje pomocí příkazu *PORTD.x=1;* .(Ukázka 3.6)

```
switch (i){
case(10):{PORTD.7=1;return;}
break;
case(11):{PORTD.6=1;return;}
break;
case(12):{PORTD.6=1;return;}
break;
case(13):{PORTD.4=1;return;}
break;
case(14):{PORTD.1=1;return;}
break;

        .
        .
        .
```

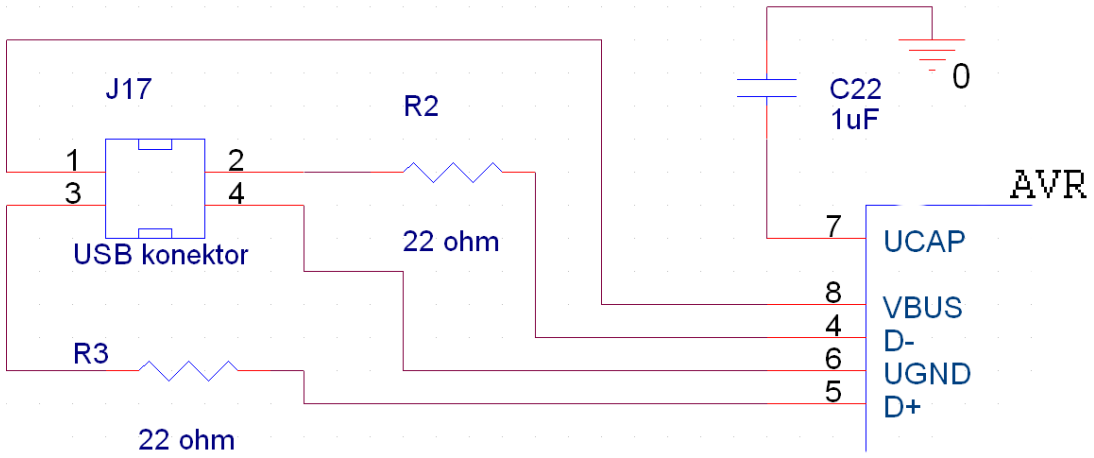
Ukázka 3.6 Nastavení stavů na 1/0 portech

3.5 USB

Pro vzdálené ovládání řídicího systému bylo zvoleno rozhraní USB a s tím související typ mikrokontroléru AT90USB1287, který má USB rozhraní implementováno čímž výrazně zjednoduší programování i připojení přístroje.

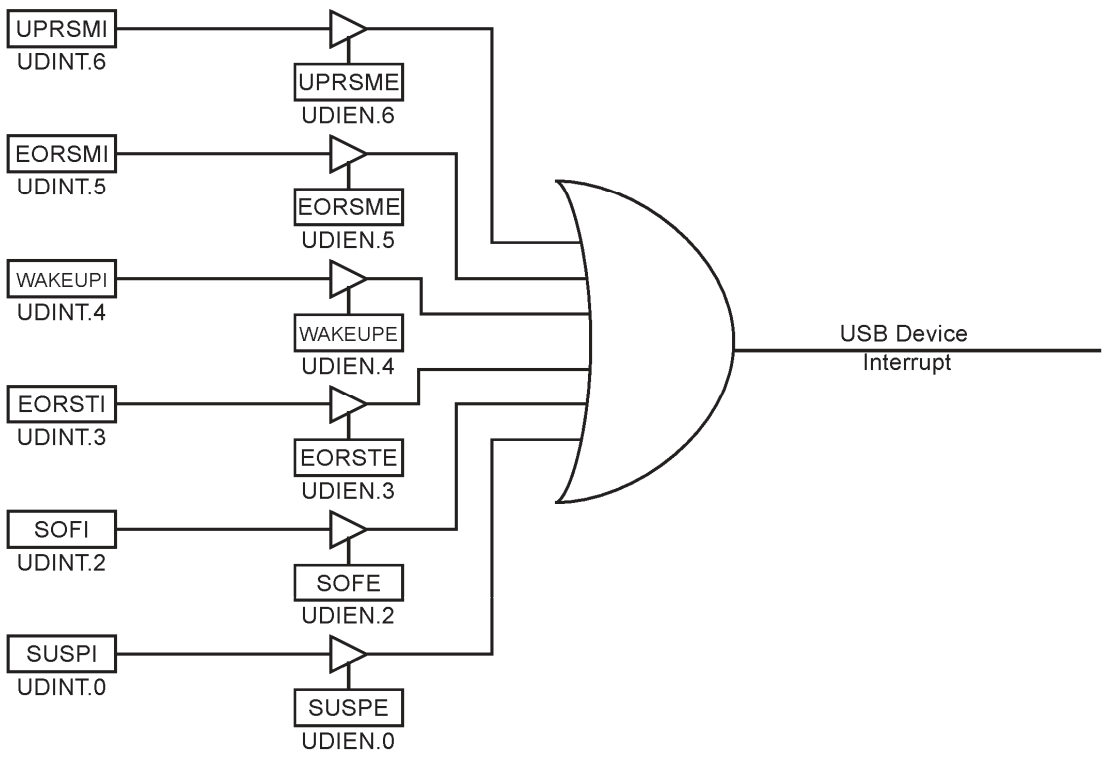
3.5.1 Hardware

Díky zahrnutí USB rozhraní přímo v mikrokontroléru je propojení velice snadné, jen na signálové vodiče D+ a D- je nutno připojit rezistor o hodnotě 22 Ohm. Zvolené propojení je pro typ "Device", pro napájení AVR z cizího zdroje.(Obrázek 3.6)



Obrázek 3.6 Propojení USB s AVR

Použitý mikrokontrolér je vybaven integrovaným hardwarovým USB řadičem. Pro ukládání přijatých a odesílaných dat je používána vlastní DPRAM paměť. Veškeré události na USB sběrnici jsou převedeny na sadu přerušení (Obrázek 3.7 a 3.8), díky čemuž je komunikace obsloužena velmi nízkými časovými prodlevami. Zpracování těchto přerušení umožňuje vyřešit poměrně složitý systém komunikace USB.



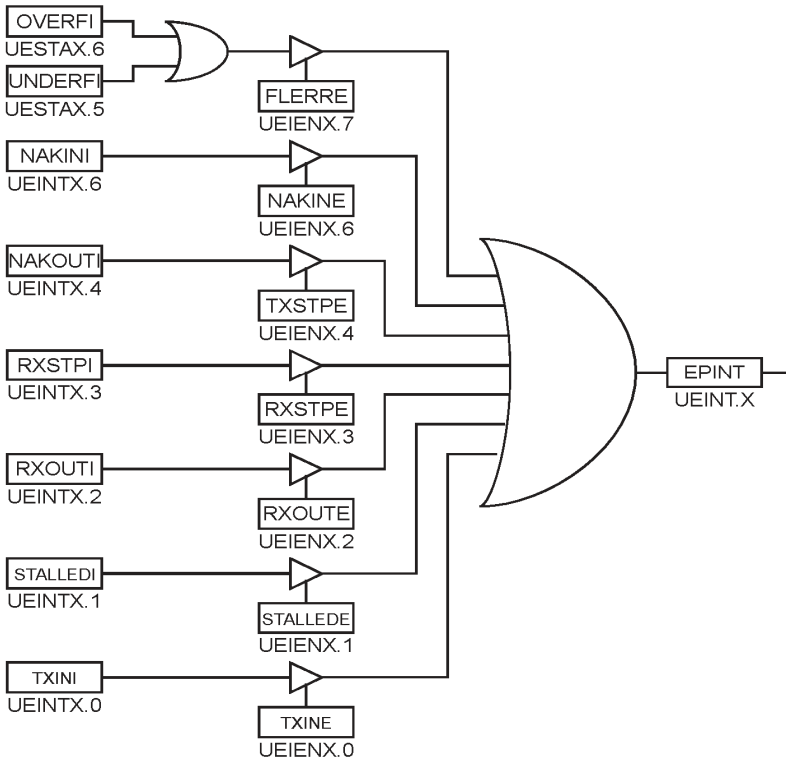
Obrázek 3.7 Přehled USB Device přerušení [1]

Význam nejdůležitějších přerušení :

WAKEUPI - Toto přerušení je aktivováno v případě že je USB kontroler znovu aktivován němčím signálem. Je řízen pomocí WAKEUPE přepínače, který musí být nastaven a hodnotu jedna aby bylo přerušení WAKEUPI aktivní.

EORSTI – (End of reset) Je nastaven v případě že je zaznamenán konec resetu USB kontrolérem. Je povolován pomocí EORSTE přepínače, který musí být nastaven a hodnotu jedna aby bylo přerušení EROSTII aktivní.

SUSPI – Je nastaven v případě že USB odloží idle mód na dobu tří rámců. Pro funkčnost musí být nastaven SUSPE přepínač do polohy 1.



Obrázek 3.8 Přehled USB ENDPOINT přerušení [1]

Význam nejdůležitějších přerušení :

TXINI – Nastaveno na jedničku v případě že aktuální banka je volná a může být naplněna.

RXSTPI – Tento příznak je nastaven v případě že aktuální banka obsahuje nová platná SETUP data. Pro fungování, musí být nastavena

3.5.2 Software

Softwarové ovládání USB je poměrně náročné, USB zařízení se musí nastavit, spojit s hostitelem pomocí endpointu 0 a až nakonec je možno vysílat nebo přijímat data. V programu Codevision je nastavování USB i jeho endpointů řešeno pomocí registrů. Nejdříve je třeba vynulovat některé z registrů a nastavit mikrokontrolér jako zařízení (device).

```
UDCON    = 0x01;        //Detach
USBCON     = 0x20;      //zastavení hodin
OTGIEN     = 0;
UDIEN      = 0;
UHIEN      = 0;
UEIENX     = 0;
UDINT      = 0;
```

```
UPIENX      = 0;
UHWCON    = 0x80;        //nastavené módu zaříyení
```

Následuje povolení některých přerušení a uvedení USB kontrolér do stavu pohotovosti

```
UHWCON   = 0x81;         // zapnutí pad
USBCON   = 0xB0;         //zastavení hodin     .
        .
        .
USBCON   = 0x90;         //zapnutí hodin
        .
        .
UDIEN    = 0x18;         //povolení wake up a reset přerušení
UDCON    = 0;            //Strach

//Následuje nekonečná smyčka doplněná o libovolný kód.
```

Dále je však nutné postarat se o přerušení wake up a end of reset. A o přerušení endpointů, z nichž nejdůležitější je enpoint 0 který vyjednává s hostitelem. Zachycení na endpointech se děje pomocí přerušení skupiny USB_ENDPOINT_PIPE (viz následující příklad).

```
interrupt [USB_ENDPOINT_PIPE] void usb_endp (void)
{

  if (UEINT & 1)             //pro endpoin 0
```

Pomocí endpointu 0 vysílá host žádosti, například o device decriptor

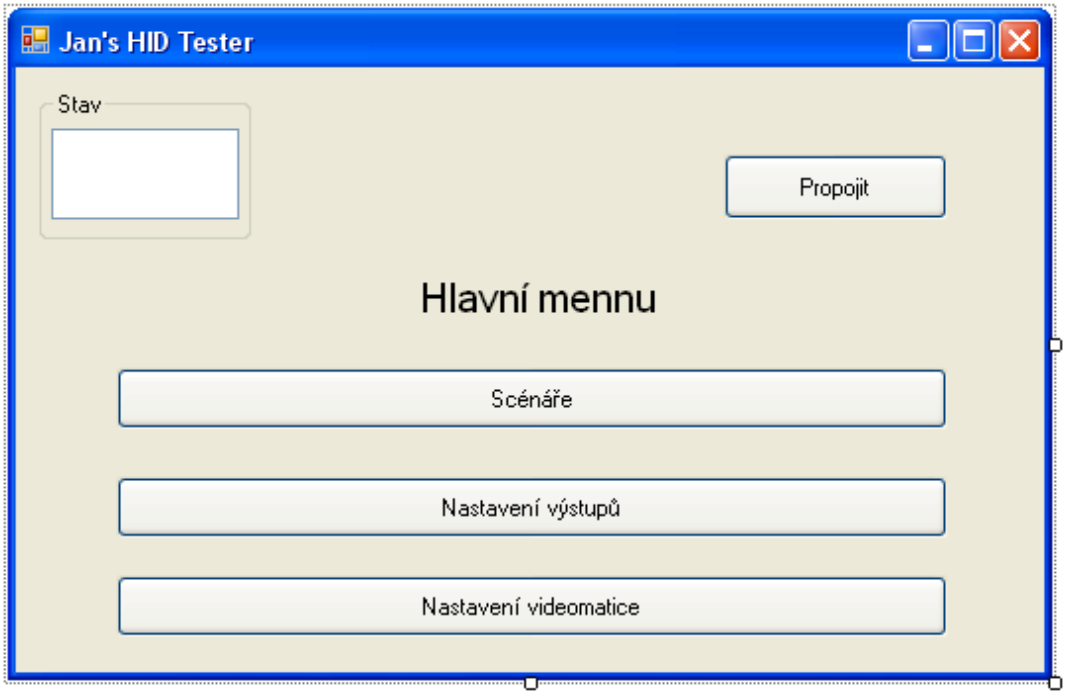
```
if ((Data[0] & 0x60) == 0x00){
     showintser(Data[1]);

     if (Data[1] == 5){

       UDADDR = Data[2];

        UEINTX &= 0xFE;
        while(!(UEINTX & 0x01));

        UDADDR |= 0x80;
     }else if(Data[1] == 6){
                                        // GET Descriptor
          if (Data[3] == 1){            // Device Descriptor
          for (a=0;a<18;a++)
             UEDATX = DEV_Desc[a];
          UEINTX &= 0xFE;
        }else if (Data[3] == 2){

          ……
```

Ukázka 3.7 Zpracování USB přerušení

Poté se podobným způsobem odešle případně i HID descriptor a je následně možno posílat a přijímat klasická data data.

3.6 Program vzdáleného ovládání

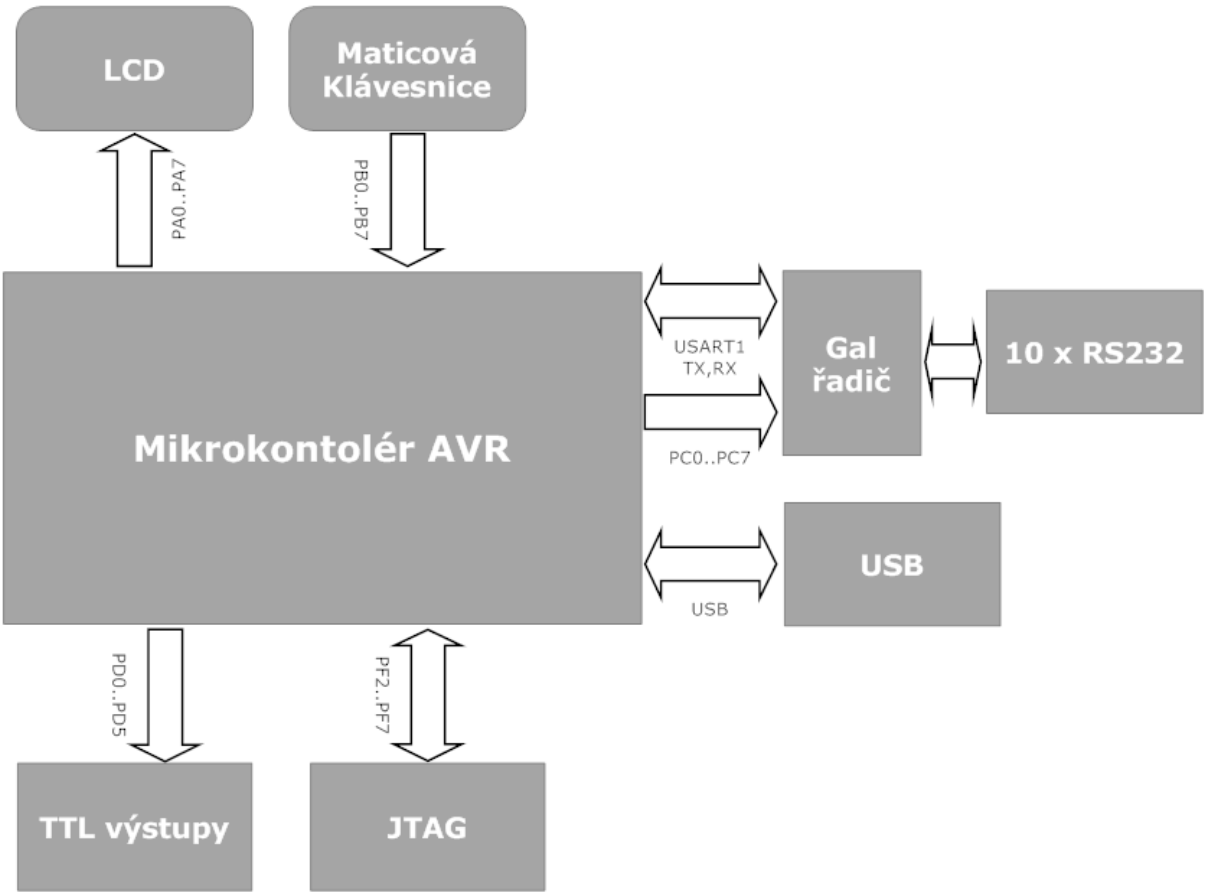
Ovládací program v PC připojeném k řídicímu systému přes USB je řešen stejně jako menu přímého v mikrokontroléru a z důvodu snadnější obsluhy i jednoduššího programování Programem vzdálené obsluhy ovládáme menu přímo menu v mikrokontroléru. Programován je v jazyce C# při využití volně použitelných knihoven.(viz. CD \Csharp) Informace mezi zařízeními jsou vyměňovány pomocí vzájemně posílaných příkazů které vzdálené ovládání informují v jaké úrovni je skutečné menu a směrem k mikročipu jsou posílány příkazy které pro menu simulují vstupy nutné pro další krok.



Obrázek 3.9 Hlavní menu v programu vzdáleného ovládání

3.7 celková struktura

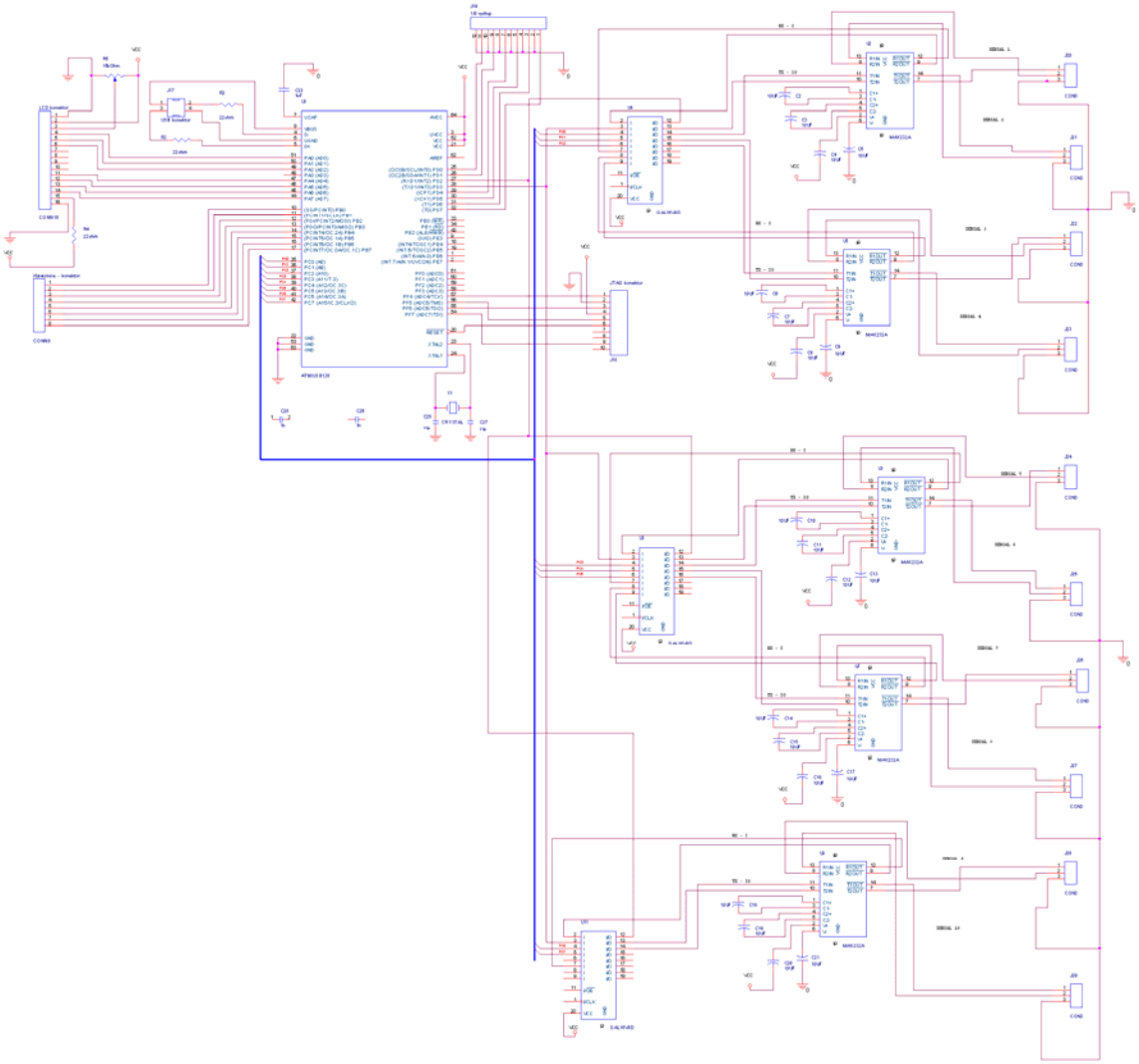
Celková struktura řídicího systému je ovlivněna převážně požadavky zadání diplomové práce, ale také cenou a náročností, kterou řešení obnáší. Jako hlavní řídící jednotka, byl zvolen mikrokontrolér AT90USB1287 řady AVR. Program bude vytvářen v jazyce C převážně v prostředí Codevision. Jako programátor je použito zařízení AVR Dragon, toto zařízení je schopno nižší modely řady AVR i ladit, díky tomu bylo možno některé jednodušší části programu odladit na modelu ATmega32, bohužel však nelze ladit přímo na použitém AT90USB1287 a proto některé části, jako například USB komunikace, musely být programovány bez možnosti ladění.



Obrázek 3.10 Blokové schéma řídicího systému

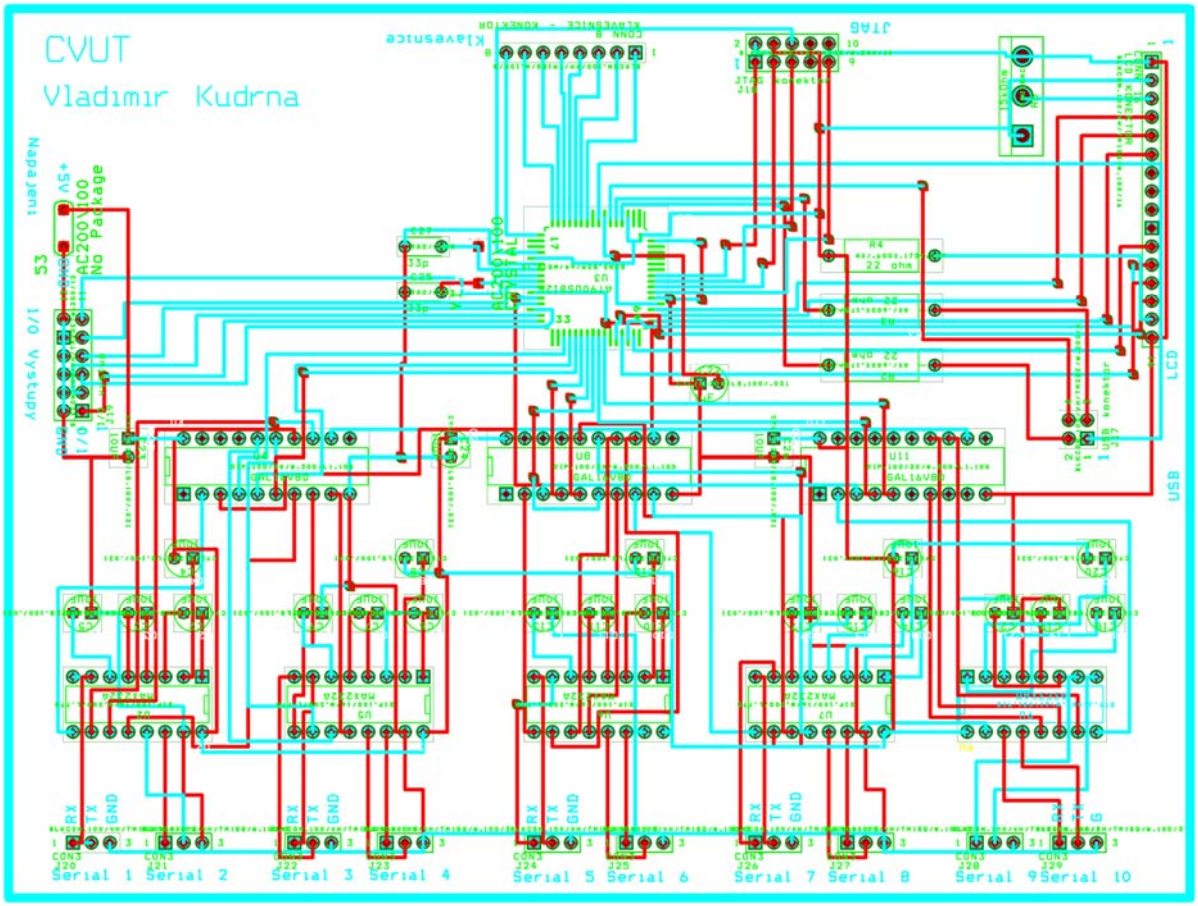
3.7.1 Hardware

Řídicí systém je založeno na mikrokontroléru AT90USB1287 a využívá většinu schopností jeho periferií. Schéma bylo navrženo v programu Capture CIS který je součástí balíku OrCAD (Obrázek 3.11).



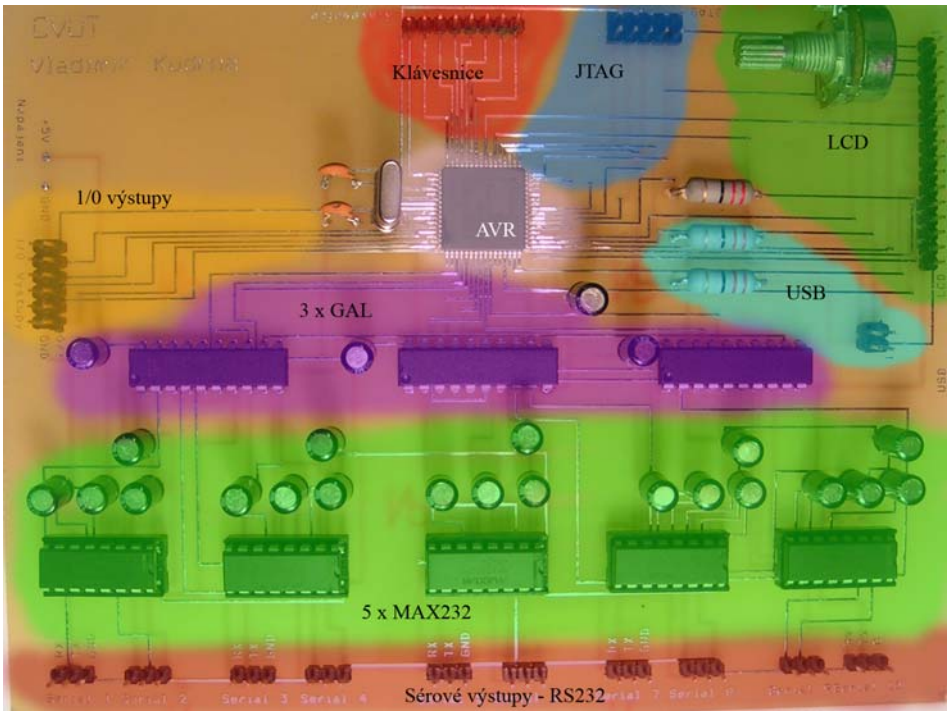
Obrázek 3.11 Schéma řídícího systému

Návrh desky plošného spoje je vytvořen pomocí programu Layout a SPECCTRA, součásti balíku ORCAD. Program SPECCTRA navrhl plošný spoj podle přednastavených parametrů. Desku bylo nutné pro vyšší složitost umístit na dvouvrstvý plošný spoj. (Obrázek 3.12)



Obrázek 3.12 Návrh desky plošného spoje

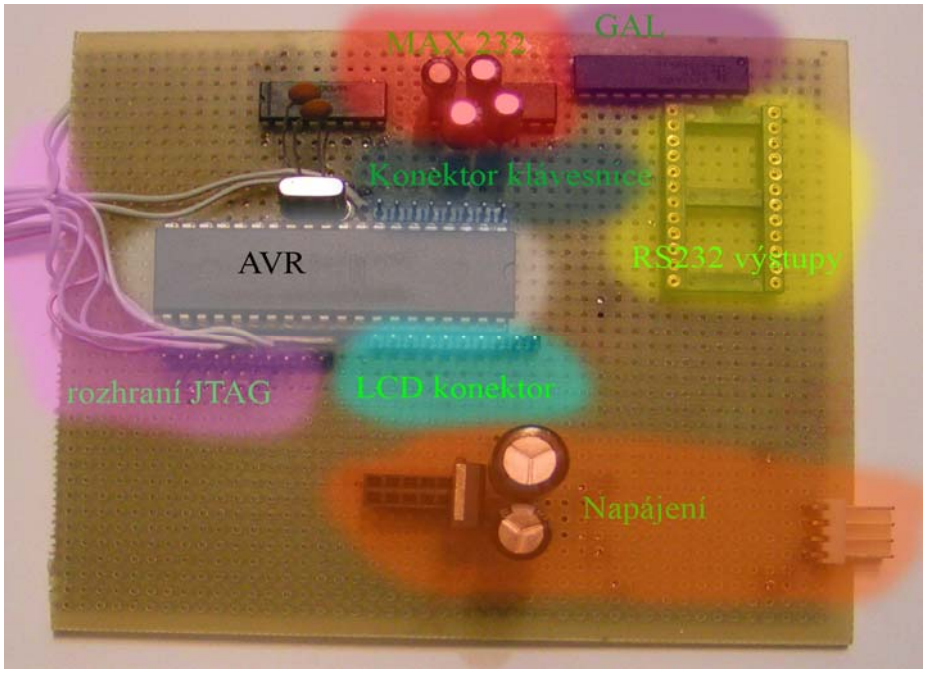
Všechny periferie, jsou propojeny pomocí lištových konektorů (JUMPER konektor) a plochých kabelů. Přípravek je napájen regulovaným napětím 5V. Plošný spoj byl vytvořen na standardní desku s tloušťkou mědí 70µm. (Obrázek 3.13)



Obrázek 3.13 Deska řídicího systému

Ladící přípravek

Pro ladění jednotlivých částí programu, byl před návrhem plošného spoje navržen a postupně dopracováván ladící přípravek. Použit byl mikrokontrolér ATmega32, protože programátor AVR Dragon který byl k dispozici umí ladit pouze nižší modely z řady AVR. Na tomto přípravku, byly odzkoušeny a laděny především knihovny pro klávesnici, LCD hradlová pole GAL a sériový kanál. Schéma plošného spoje odpovídá až na propojení USB a počet sériových konektorů konečnému zapojení.



Obrázek 3.14 Ladící deska

3.7.2 Struktura Menu

Přímé nastavování řídicího systému se provádí pomocí maticové klávesnice a LCD pomocí kterých se orientujeme menu přístroje. Výběr je omezen v hlavním menu na tři základní možnosti. Tyto možnosti jsou :

Scénáře – toto volba vede ke nastavení a uložení scénářů

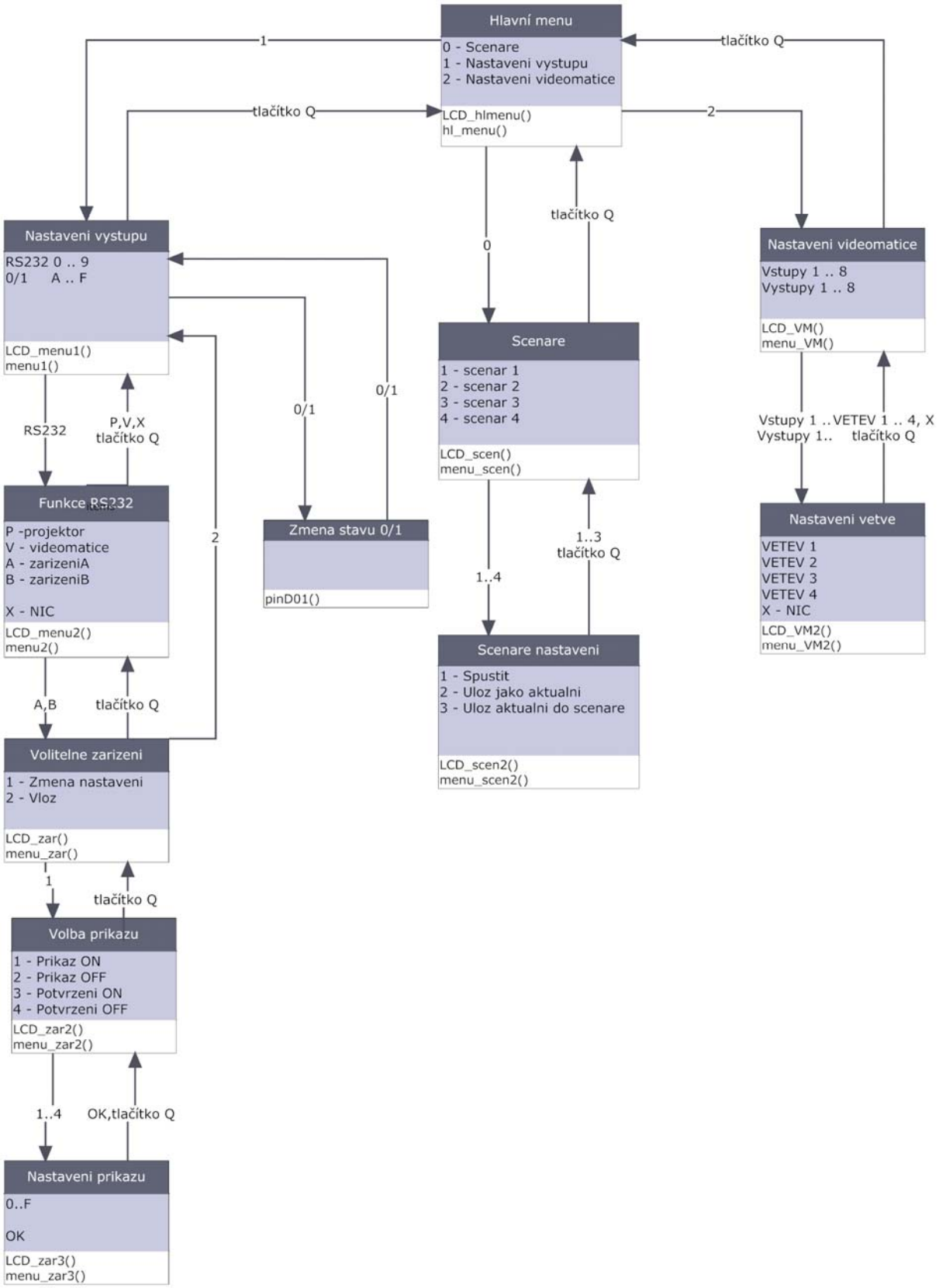
Nastavení výstupů – pomocí této volby jsou nastavovány typy přístrojů připojovaných pomocí sérového rozhraní RS232, jejich funkce a také funkce výstupů 0/1.

Videomatice – Tato volba nastavuje pomocí větví propojení vstupů a výstupů ve videomatici .

Menu je programem implementováno pomocí funkcí jenž představují každé okénko v menu řízené pomocí funkce switch() a pomocných proměnných. Tato metoda spočívá vtom že každý krok v menu má vlastní funkci zobrazující na displayi aktuální nabídku a druhou funkci, která je jakousi maskou pro zobrazený obsah a přiřazuje podle pozice kurzoru na displayi správnou funkci k dané položce v menu (Obrázek 3.15).

```
switch (mm){

    case(3):{

    LCD_hlmenu();
    mm=0xFE;                //přednastavení pomocné proměnné
    while (mm==0xFE){


        ih=nacti_POZ();     // v jiném případě než je zmáčknutí potvrzovací klávesy
        mm=hl_menu(ih);     // když je kurzor vhodném místě, vrací hlmenu hodnotu
                            // hFE
   } }
    break;

     case(0):{              //v případě potvrzení kurzoru na pozici "nastavení vystupu"
     LCD_menu(stavy);       // je nastavena pomocná proměnná na 0 a menu se
     m1=0xFE;               // přesune na
     while (m1==0xFE)       // tuto smyčku
           {
        ik=nacti_POZ();
        m1 = menu1(ik);

     }
    }
      break;
```

Ukázka 3.8 Tvorba menu



Obrázek 3.15 Diagram znázorňující strukturu menu

Ovládání menu

Zvolená metoda ovládání menu je založena na snímání pozice kurzoru z displeje, díky klávesnici je možno se posouvat po zobrazeném menu na displeji a po stisknutí potvrzovacího tlačítka je odeslána pozice displeje. Tato metoda umožňuje nastavovat i poměrně složité parametry zařízení bez větších nároků na klávesnici, avšak je nutné pro každý krok v menu změnit masku snímání pozice kurzoru, která nastavuje, jak bude pozice kurzoru vnímána. Tuto metodu zajišťují tyto funkce:

keyb() – funkce jenž čte stisknutou klávesu

nacti (char c) – funkce která přiřazuje funkci určitým klávesám načtených pomocí funkce keyb() a v případě stisku potvrzovacího tlačítka vrací načtenou pozici.

nacti_poz() - funkce která volá předchozí funkce a podle načtených dat upraví pomocné proměnné pro menu

lcd_read_b0() – načte znak z aktuální pozice

Nastavení výstupů:

Pomocí této části menu je zobrazováno a nastavováno jaké zařízení se připojí na sériové rozhraní RS232. Funkce LCD_menu1() zajišťuje zobrazení informací na display, funkce menu1() díky získané pozici kurzoru na displeji, buď v případě 0/1 výstupů změní stav výstupu v paměti eeprom, nebo nastaví pomocné proměnné přejde do menu funkce RS232 kde se nastavují již konkrétní funkce pro RS232. Změnu stavu výstupů 0/1 zajišťuje funkce pinD01(int i), kde i je pozice měněného výstupu.

Funkce RS232

Menu "Funkce RS232" je zobrazeno díky funkci LCD_menu2, funkce menu2 zajišťuje díky datům, která byla uložena do pomocné proměnné a pozici kurzoru zapsání typu připojeného zařízení do eeprom. Můžeme zvolit funkci projektor, videomatice, nebo zařízení 1 a 2 jenž jsou zařízení definovatelná uživatelem.

Volitelné zařízení

Tato část menu je určena pro nastavení volitelného zařízení, v případě že volby "Změna nastavení" se zobrazí další nabídka kde vybíráme který příkaz chceme editovat. Poté zadáme příkaz a uložíme potvrzením kurzoru na pozici OK. Jednotlivé stupně v menu "Volitelné zařízení " jsou zobrazovány pomocí funkcí LCD_zar(), LCD_zar1(), LCD_zar2() a data jsou zpracovávána pomocí funkcí menu_zar(),menu_zar1(),menu_zar2() z nichž vlastní uložení navolených příkazů do eeprom provádí menu_zar2().

Nastavení Videomatice

Zde je zobrazováno propojení vstupů a výstupů videomatice, pomocí kurzoru je možno zvolit který vsup či výstup bude nastavován. Zobrazení menu zajišťuje funkce LCD_VM(), menu_VM pomocí získané pozice displaye určí který vstup nebo výstup bude nastavován a uloží informaci do pomocné proměnné.

Nastavení větve videomatice

Větve jsou určeny pro propojování vstupů s výstupu, v každé větvi může výt maximálně jeden vstup a až 8 výstupů. Zobrazení menu zajišťuje funkce LCD_VM2. Funkce menu_VM2 ukládá do paměti eeprom správné nastavení videomatice. Nastavení videomatice získáme pomocí čísla vstupu či výstupu poslaného pomocnou proměnnou z menu nastavení videomatice a čísla větve, které bylo získáno pomocí čtení pozice displeje.

Scénáře

Tato část menu umožňuje výběr jednoho ze scénářů. Funkce LCD_scen() zobrazuje možnosti scénáře a funkce menu_scen() předává označení zvoleného scénáře pomocné proměnné.

Nastavení Scénáře

Menu "Nastavení Scénáře" určuje jaká akce se provede se scénářem zvoleným v předchozím kroku menu.
Menu je zobrazeno pomocí funkce LCD_scen2 a pomocí menu_scen2 volá funkce pro vykonání zvolené akce. Volby jsou následující:

Spustit - nastaví zvolený scénář jako aktuální a spustí ho pomocí zavolané funkce menu_proved()

Uložit jako aktuální – zvolený scénář uloží jako aktuální

Uložit aktuální do scénáře – přepíše zvolený scénář aktuálním nastavením

3.8 Knihovny ovládaných zařízení

3.8.1 Videomatice

Knihovna videomatice Kramer má za úkol vytvořit takovou posloupnost příkazů, které ji správně propojí a nastaví. Propojení ve videomatici při každém nastavení jsou nejdříve vymazány, potom jsou načtena data pro zapsání a nakonec poslán příslušný příkaz o délce čtyři byty.

Byte 1 – Určuje směr komunikace, tedy jestli jde o četní nebo zápis a určuje typ příkazu. Pro zápis a přepínání video vstupů je hodnota 1h.

Byte 2 – Nastavuje vstupy videomatice, nevyšší bit je vždy 1 a další bity odpovídají číslu vstupu, který propojuje. Při propojování vstupu 1 bude tedy číslo hodnota tohoto byte 81h.

Byte 3 - Nastavuje vsýtupy videomatice, nevyšší bit je vždy 1 a další bity odpovídají číslu výstupu, který propojuje. Při propojování výstupu 2 bude tedy číslo hodnota tohoto byte 82h.

Byte 4 – Představuje číslo videomatice pro kterou je příkaz posílán,nejvyšší bitje vždy 1 v případě že je ovládána jedna videomatice je hodnota tohoto byte vždy 81.

```
void vmZAPISdo(void){
int byte1, byte2, byte20, byte3, byte4ů
int i, ii;
byte1=0x1;
byte20=0x80;
byte4=0x81;          //nastavení standardních hodnot

vmCTIin();
vmCTIout();          //načtení zapisovaných hodnot

for(i=1;i<5;i++){    //nulování
putchar(byte1);
delay_us(10);
putchar(byte20);
delay_us(10);
byte3=i+0x80;
putchar(byte3);
delay_us(10);
putchar(byte4);
delay_ms(100);
}
 for(ii=pocS1;ii>0;ii--){      //počet nastavovaných vstupů v globální proměnné
for(i=pocSo1;i>0;i--)         // počet nastavovaných výstupů v globální proměnné
{
byte3=spojo1[i-1]+0x80;   //zápis do videomatice
byte2=spoj1[ii-1]+0x80;
putchar(byte1);
delay_us(10);
putchar(byte2);
delay_us(10);
putchar(byte3);
delay_us(10);
putchar(byte4);
delay_us(1000);
}}
```

Ukázka 3.9 Tvorba příkazu pro videomatici

3.8.2 Projektor

Projektor Panasonic je ovládán pomocí krátkých příkazů, z kterých jsou hlavní příkazy pro zapnutí a vypnutí. Díky tomu že projektor po přijetí příkazu potvrzuje, že je příkaz správný tím způsobem, že odesílá stejný příkaz zpět, musíme z data přístroje zachytávat ihned po odeslání. Každý příkaz musí obsahovat startovací byte(02h) příkaz(například PON) a ukončovací byte(03h).

```
void projON(void)
{
int i;
char startB = 0x02;
char endB = 0x03;
char getP='<';
char cc[]={'P','O','N'};
char c[3];                      //tvorba příkazu

lcd_goto(8,0);
putchar(startB) ;
printf(" PON ");
putchar(endB) ;                 //vyslání příkazu +kontroly na lcd

 for(i=0;i<3;i++){
getP=getchar();
lcd_putchar(getP);
c[i]=getP;
}                               //přijetí a kontrola přijatých dat
if(c[i]==cc[i]){lcd_putsf("OK");}
else (lcd_putsf("PWR_ERR"));
}
```

Ukázka 3.10 Sekvence pro start Projektoru

3.8.3 Editovatelná zařízení

V případě editovatelných zařízení bylo počítáno pouze s jednoduššími zařízeními typu projektor. Proto je tvorba příkazů v menu omezena na startovací a ukončovací příkaz a následné potvrzení obou příkazů. Tvorba příkazu je podobná jako u projektorů.

```
void zar1_ON(void)
{
int i;
        ;                       //příkaz je v globální proměnné v eeprom zar_1st

char c[8]={0,0,0,0,0,0,0,0}
 for(i=0;i<lengz1;i++){
```

```
putchar(zar_1st[i]);
}
for(i=0;i<lengz1;i++){
putchar(zar_1st[i]);
}
for(i=0;i<lenpz1;i++){
getP=getchar();
c[i]=getP;
}                                          //přijetí a kontrola přijatých dat
if(c[i]==zar_1stp[i]){lcd_putsf("OK");}
else (lcd_putsf("PWR_ERR"));
}
```

Ukázka 3.11 Funkce pro start editovatelného zařízení

4. Ovládání a testy

4.1 Ovládání

Ovládání přístroje je rozděleno na dvě úrovně a to uživatelská a správcovská, v uživatelské úrovni budou používána pouze 4 tlačítka pro volbu scénáře a vše ostatní bude přednastaveno správcem.

Správce je tedy zodpovědný za správné nastavení a připojení ovládaných zařízení a sestavení scénářů. Pro nastavení musí využívat na rozdíl od uživatele display nebo vzdálené ovládání. Menu je pro obě verze nastaveno stejně a tak je nutné učit se s ním pracovat jen jednou. Manuál pro práci je podrobně popsán v příloze B.

4.2 Testování a uvedení do provozu

Testování probíhalo v provizorním sestavení, protože deska bude v panelu, který ještě není sestaven. Proto jsou veškeré periferie pouze vyvedeny pomocí kabelů mimo desku a zařízení není sestaveno jako celek.

4.2.1 Periferie a výstupy

První testování periférií probíhalo souběžně s vytvářením, ve většině případů již na ladící desce a nevyžadovalo žádné speciální prostředí.

V případě klávesnice nebylo uvedení do provozu doprovázeno žádnými problémy, hlavním úkolem v praxi bylo nastavení čekání, které ovlivňuje nutnou délku stisku a případný problém s falešným vícenásobným stiskem.

V případě LCD bylo uvedení do provozu a testování doprovázeno menšími problémy, když bylo nutno některé pauzy uváděné výrobcem prodloužit cca na dvojnásobek. Dále bohužel použitý řadič vnímá display při čtení pozice kurzoru jako dvouřádkový, což také muselo být zohledněno jak při čtení pozice tak při posunu nahoru a dolů.

Při návrhu sériového rozhraní se vyskytl pouze problém s programováním hradel GAL, do nichž často nešla nahrát vytvořená data. Tento problém zmizel po přehání ovládacího SW programátoru BeeProg a očistění patice.

Problém byl v případě rozhraní USB, které pravděpodobně díky nepřesnému časovému taktu dlouho nekomunikovalo se hostitelem (tedy PC). Tento problém byl vyřešen výměnou mikrokontroléru a následnou výměnou krystalu.

4.2.2 Testy celku

Protože je zařízení určeno k praktickému použití byl testován v podmínkách maximálně podobných konečnému provozu a tedy v učebnách katedry Řídicích systémů.

Pro testování byl sestaven základní scénář obsahující pouze projektor Panasonic a videomatici Kramer. Nebylo možné připojit zařízení pro 0/1 vstupy, protože ještě nebyly dokončeny úpravy místností a přidružených zařízení.

Během testování bylou nutné prodloužit dobu mezi příkazy zasílanými do videomatice a v případě projektoru bylo nutné upravit program pro případ, že je projektor fázi startu nebo ukončování a neodpovídá a celý program se zastavil na této chybě.

Po vyřešení těchto drobných chyb byl řídící systém jako celek funkční a schopný bezproblémového provozu.

5. Závěr

Výsledkem práce je funkční řídicí systém určený pro ovládání prezentační místnosti, který je schopný řídit 10 zařízení připojitelných pomocí sériového rozhraní a 6 zařízení připojitelných pomocí 0/1 rozhraní. Systém umožňuje ovládání dva typy ovládání, přímé a vzdálené, které jsou založeny na stejném menu. V souladu s cíli práce byl navržen vhodný hardware a kód v programovacím jazyce C pro zvolený AVR mikrokontrolér AT90USSB1287. Pro vzdálenou komunikaci byl navržen ovládací program v jazyce C#, za použití volně šířitelných knihoven pro USB. Hlavní váhu práce na sobě nesl program pro mikrokontrolér, který je vytvořen tak, aby byl snadně čitelný a v případě potřeby jednoduše upravitelný.

Dokončením zařízení byly splněny stanovené cíle práce. Zařízení bylo úspěšně otestováno, případné chyby byly odstraněny a jeho funkce jsou dostatečné pro učený provoz.

Literatura

[1] Atmel Corporation. Dokumentace mikrokontroléru AT90USB1287.
(http://www.atmel.com/dyn/products/devices.asp?family_id=607)

[2] Atmel Corporation. .AVR studio – vývojové prostření pro simulaci programů v AVR.
(http://www.atmel.com/dyn/Products/tools_card.asp?tool_id=2725)

[3] HID USB Driver / Library.
(http://www.florian-leitner.de/index.php/2007/08/03/hid-usb-driver-library/#hid-class)

[4] Universal Serial Bus Specification Revision 1.1.
(http://www.usb.org/developers)

[5] Maxim. MAX232 manual.
(http://www.maxim-ic.com/quick_view2.cfm?qv_pk=1798)

[6] Domenatce ke klávesnici F-KV16KEY.
(http://www.gme.cz/cz/index.php?product=637-091)

[7] Lattice. Dokumentace hradlového pole GAL 6V8
(http://www.latiticesemi.com)

[8] Epson. Manuál projektoru Epson EB-W6.
(http://www.epson.com.sg/products/projectors/EPSON_EB-W6.shtml)

[9] Kramer. Manuál videomatice KRAMER. VP-8x8
(http://www.kramerelectronics.com/products/model.asp?pid=387&sf=47)

[10] Panasonic Dokumentace projektoru PANASONIC FV100NT.
(http://panasonic.com/pub/Panasonic/Drivers/PBTS/manuals/OM_PT-FSeriesRS232C.pdf)

[11] Atmel Corporation. Dokumentace programátoru AVR Dragon.
(http://support.atmel.no/knowledgebase/avrstudiohelp/mergedProjects/AVRDragon/AVRDragon.htm)

[12] Extron. Dokumntace videomatice Extron MVX 88 VGA
(http://www.extron.com/product/product.aspx?id=mvx88vga&subtype=136)

[13] Popis funkcí LCD založeném na řadiči HD44780.
(http://hw.cz/novinky/art2727-ovladani-znakovych-lcd-s-radicem-hd44780.html)

[14] Popis rozhraní USB
(http://hw.cz/Teorie-a-praxe/Dokumentace/ART327-USB---Universal-Serial-Bus--- Popis-rozhrani.html)

[15] Veemux. Dokumentace videomatice Veemux SM-8x8-15V-LCD

(http://www.42u.com/nti-matrix-video-switch.htm)

[16] Veemux. Dokumentace videomatice Veemux SM-8x8-15V-LCD
(http://www.42u.com/nti-matrix-video-switch.htm)

[17] Hpinfotech . Dokumentace k porostření Codevision AVR ¨
(http://www.hpinfotech.ro/html/download.htm)

[17] Optoma. Dokumentace Projektoru Optoma EP721
(http://www.optomausa.com/Product_detail.asp?productsubcat=1&productcategory=Business+Data&product_id=337&itemno=EP721)

Obsah CD

Obsah přiloženého CD je rozdělen do X adresářů podle kategorií. Tyto kategorie jsou :

PROGAVR – Zdrojový kód pro mikrokontrolér, napsaný v jazyce C včetně všech potřebných knihoven.

MANUALY – Manuály ke všem použitým přístrojům, programům knihovnám součástkám ve formátu pdf.

CSHARP – Ovládací program pro vzdálené ovládání vytvořený v jazyce C# včetně zdrojového kódu a knihoven.

GAL – Zdrojový kód pro použitá hradlová pole GAL, obsahuje i program free verzi programu OrCAD PLD.

SCHEMA – Obsahuje schéma vytvořené v programu OrCAD Capture i návrh desky plošného spoje a obrázky ve vysokém rozlišení.

OVLA – Manuál pro ovládání řídicího systému obsahující verzi uživatelskou stávcovou, ve formátu pdf.

DIPLOM – Tato diplomová práce ve formátu pdf.