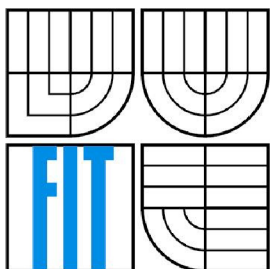




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# DÁLKOVÉ ŘÍZENÍ A MĚŘENÍ POMOCÍ OS LINUX

REMOTE CONTROL AND MEASUREMENT WITH LINUX

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JIŘÍ ŠEVCOVIC

VEDOUČÍ PRÁCE  
SUPERVISOR

ING. JAROSLAV ŠKARVADA

BRNO 2007

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2006/2007

# Zadání bakalářské práce

Řešitel: **Ševcovic Jiří**

Obor: Informační technologie

Téma: **Dálkové řízení a měření pomocí OS Linux**

Kategorie: Počítačová architektura

### Pokyny:

1. Seznamte se s operačním systémem Linux, principy meziprocesové komunikace a způsobem programování sériových portů v tomto OS. Nastudujte HTML, CGI, PHP, protokol I2C. Provéřte čidla dostupná na našem trhu a zvolte vhodná čidla.
2. Navrhněte modulární systém s OS Linux, který přes internet umožní dálkové řízení spotřebičů, monitorování stavu spotřebičů, přenos jednoduchých provozních zpráv, snímání a zaznamenávání historie teploty, snímání úrovně ionizace vzduchu okolního prostředí a případně i dalších veličin dle uvážení.
3. Navržený systém prakticky realizujte.
4. Demonstrujte funkčnost realizovaného systému na vhodných příkladech.

### Literatura:

- dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů zadání 1-2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

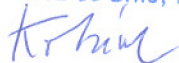
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Škarvada Jaroslav, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačových systémů a sítí  
602 00 Brno, Božetěchova 2



---

doc. Ing. Zdeněk Kotásek, CSc.  
vedoucí ústavu

**LICENČNÍ SMLOUVA  
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Jiří Ševcovic**  
Id studenta: 84077  
Bytem: Sídliště Osvobození 620/7, 682 01 Vyškov  
Narozen: 11. 04. 1985, Vyškov  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1**

**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Dálkové řízení a měření pomocí OS Linux  
Vedoucí/školitel VŠKP: Škarvada Jaroslav, Ing.  
Ústav: Ústav počítačových systémů  
Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

|                    |   |
|--------------------|---|
| tištěné formě      | počet exemplářů: 1                                  |
| elektronické formě | počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD) |



2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## Článek 2 Udělení licenčního oprávnění


1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

  
.....  
Autor



## **Abstrakt**

Cílem práce bylo vytvoření modulárního systému pro dálkové řízení a měření přes operační systém linux. Využita byla sběrnice I<sup>2</sup>C, připojená na sériový port počítače. K řízení zařízení na sběrnici slouží aplikace vytvořená v jazyce C++. Ta komunikuje s webovým rozhraním, přes které ji může uživatel ovládat.

## **Klíčová slova**

Sběrnice, I<sup>2</sup>C, teplotní senzor, sériová komunikace, RS232, LCD

## **Abstract**

The destination of this work was a creation of modular system for a remote control and measurement with Linux operation system. People use an I<sup>2</sup>C bus, connected to serial port of a computer. Application is made with C++ language to perform a remote control of systems. Application communicates with web interface, which is controlled by user.

## **Keywords**

Bus, I<sup>2</sup>C, temperature sensor, serial communication, RS232, LCD

## **Citace**

Jiří Ševcovic: Dálkové řízení a měření pomocí OS Linux, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Dálkové řízení a měření pomocí OS Linux

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jaroslava Škarvady

Další informace mi poskytl Ing. Jan Haša.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jiří Ševcovic  
5.5.2007

## Poděkování

Rád bych poděkoval panu Ing. Jaroslavu Škarvadovi, který byl vedoucím mé bakalářské práce a poskytl mi odbornou pomoc při konzultacích. Dále své rodině a přátelům, kteří mě podporovali při práci.

© Jiří Ševcovic, 2007.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

|  |    |
|--|----|
| Obsah .....  | 1  |
| Úvod .....   | 3  |
| 1 Sběrnice.....                                    | 5  |
| 2 Protokol RS-232.....                             | 7  |
| 3 Sběrnice I <sup>2</sup> C .....                  | 10 |
| 3.1 Obecný popis sběrnice .....                    | 10 |
| 3.2 Protokol sběrnice.....                         | 10 |
| 3.2.1 Zahájení komunikace .....                    | 11 |
| 3.2.2 Ukončení komunikace .....                    | 11 |
| 3.2.3 Zaslání a potvrzení byte .....               | 11 |
| 3.2.4 Princip komunikace .....                     | 11 |
| 3.3 Propojení sběrnice I <sup>2</sup> C a PC ..... | 12 |
| 4 Řídící aplikace .....                            | 14 |
| 4.1 Základní princip .....                         | 14 |
| 4.2 Popis činnosti aplikace .....                  | 14 |
| 4.3 Popis dílčích částí.....                       | 15 |
| 4.3.1 Kalendář.....                                | 15 |
| 4.3.2 Vzdálené řízení .....                        | 16 |
| 4.3.3 Komunikační protokol .....                   | 16 |
| 4.3.4 Databáze a práce s ní.....                   | 17 |
| 4.4 Třída SerialPort .....                         | 19 |
| 4.5 Třída TI2C.....                                | 19 |
| 4.6 Objekty zařízení .....                         | 19 |
| 4.6.1 Objekt TLCD .....                            | 20 |
| 4.6.2 Objekt TTherm.....                           | 20 |
| 5 Webové rozhraní.....                             | 20 |
| 6 Konstrukce modulů a jejich ovládání.....         | 21 |
| 6.1 LCD modul.....                                 | 21 |
| 6.1.1 Zapojení vývodů LCD .....                    | 21 |
| 6.1.2 Instrukce pro řízení LCD .....               | 22 |
| 6.1.3 Propojení s I <sup>2</sup> C sběrnici.....   | 23 |
| 6.2 Teplotní senzor.....                           | 23 |
| 6.2.1 Čidlo KTY81-210 .....                        | 25 |
| 6.2.2 PCF8591 .....                                | 25 |



|   |                                   |    |
|---|-----------------------------------|----|
| 7 | Propojení jednotlivých částí..... | 25 |
| 8 | Závěr .....                       | 26 |
|   | Literatura .....                  | 27 |

# Úvod

Tématem práce je využití I<sup>2</sup>C pro řízení a měření ve spojení s OS Linux. Věnuje se možnosti využití sběrnice s připojenými zařízeními přes sériový port. Důležitou součástí realizace je programové vybavení, které umožňuje transparentní přístup ke sběrnici, snímání a řízení připojených zařízení. Tato aplikace pracuje jako vícevláknová a umožňuje řízení sběrnice pomocí navrženého protokolu I2C přes webové rozhraní. Pro komunikaci se sběrnici přes protokol I2C byl vytvořen klient v jazyce PHP, který komunikuje se serverem, zobrazuje zjištěné informace z připojených zařízení a umožňuje jejich vzdálené ovládání. Využití sběrnice je velice široké. Uplatnění může nalézt například jako systém pro snímání dat ze vzdálených senzorů a jejich prezentace na internetu. Lze ji využít pro laboratorní úlohy při měření a vyhodnocování fyzikálních či chemických jevů, případně pro vytvoření inteligentní domácnosti. Výhodou této sběrnice jsou především nízké náklady na pořízení jednotlivých obvodů a jejich dostupnost na trhu. Tento standard podporuje velké množství výrobců a díky konkurenčnímu boji se ceny pohybují v přijatelné míře. Sběrnice je natolik rozšířená, že někteří návrháři operačních systémů začlenili její podporu do jádra svých systémů. Je tomu tak především u OS Linux.

V první kapitole se budu zabývat úvodem do problematiky sběrnic. Jaké jsou typy, způsoby komunikace a jejich realizace. Dozvíte se výhody a nevýhody jednotlivých řešení a při jakých situacích je vhodné je použít.

Kapitola číslo dva se zabývá komunikací mezi počítačem a perifériemi a rozeberu zde standard RS-232, který je využit pro komunikaci mezi I<sup>2</sup>C a počítačem. V této části se okrajově zmíním o vývoji a různých režimech, které sběrnice umožňuje a hlavně o informacích, nutných k pochopení dané problematiky.

Ve třetí kapitole uvádím hlavní parametry sběrnice I<sup>2</sup>C. Důkladně zde proberu protokol této sběrnice a na konci kapitoly vysvětlím možnosti propojení s počítačem.

Konstrukce programového vybavení, které řídí chod sběrnice je tématem kapitoly číslo 4. Zde se zabývám hlavním řídicím modulem sběrnice, popisem částí ze kterých se skládá aplikace, jejím napojením na internet a propojením s webovým klientem. To považují za nejdůležitější část mé práce.

Návrh webového klienta je dalším zásadním úkolem práce. Z internetové stránky dokážeme ovládat sběrnici, zobrazovat aktuální stavy senzorů a grafy průběžně naměřených hodnot.

Abych mohl prakticky demonstrovat činnost sběrnice, bylo nezbytné vytvořit demonstrativní moduly. Kapitola 6 se stručně zabývá jejich konstrukcí a způsobem ovládání. Neřešil jsem zde do hloubky technické detaily. Ty lze nalézt v příložené dokumentaci použitých obvodů a čidel.

Kapitola 7 stručně popisuje propojení všech komponent sběrnice.

Závěrečná kapitola ukazuje dosažené výsledky a můj přínos do technického řešení uvedeného problému. Uvedl jsem zde nástin další tvůrčí činnosti v této oblasti a poukázal na mé zkušenosti, vědomosti a dovednosti, kterých bylo nutno pro splnění cílů využít.

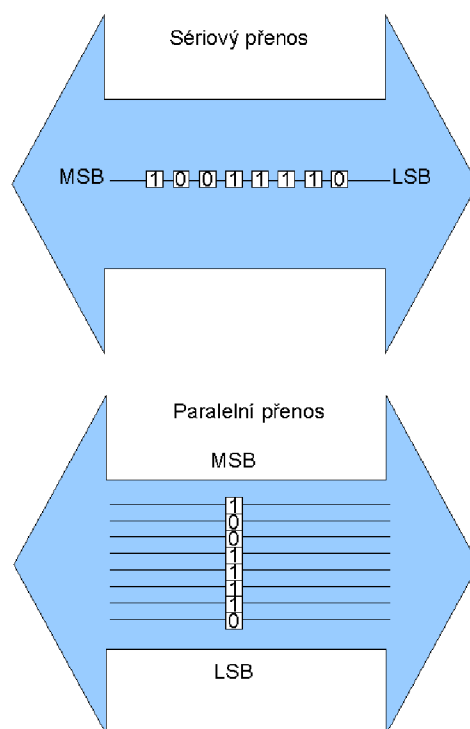


# 1 Sběrnice

Sběrnice je fyzicky skupina vodičů, přes které probíhá komunikace mezi minimálně dvěma zařízeními. Počty vodičů se různí podle typu sběrnice. Záleží především na jakém principu pracují, způsobu přenosu dat a koncovém využití. Při navrhování sběrnice je potřeba analyzovat prostředí ve kterém bude pracovat a přes jaké médium bude komunikovat.

Sběrnice je možné rozdělovat podle mnoha kritérií. Podle způsobu přenosu bitů je dělíme na sériové a paralelní. Pro sériové je typické, že přenos informace se provádí po jednom vodiči a to tak, že jsou postupně za sebou posílány jednotlivé bity. Tato metoda je jednodušší na implementaci, protože není nutné řešit těžkosti spojené s velkým počtem vodičů. U paralelní komunikace se přenáší větší počet bitů současně. Počet přenesených bitů v jednom taktu je dán šířkou sběrnice, kterou určuje počet vodičů. V minulosti se ve většině případů využíval pro sběrnice paralelní přenos, který však v současné době narazil na své fyzikální hranice a není možné dále zvyšovat rychlosti přenosu touto cestou. Nové, vysokorychlostní sběrnice, jsou dnes konstruovány především na principu sériového přenosu.

Problém s přeslechy se kterými jsme se potýkali hlavně u paralelního vedení signálu byl z velké části odstraněn použitím kroucené dvoulinky. Princip tohoto vedení spočívá v použití dvou vodičů pro přenos jednoho signálu, které jsou stočené kolem sebe. Logická hodnota signálu je poté zjištěna z difference napětí mezi vodiči. Tento spoj umožňuje spolehlivý přenos na veliké vzdálenosti s vysokou přenosovou rychlostí. Princip přenosu bytu přes sériovou a paralelní sběrnici je znázorněn na obrázku 1.1.



1.1 Princip přenosu bytu 10011110 přes sériovou a paralelní sběrnici

Podle dalšího kritéria je možné dělit sběrnice na synchronní a asynchronní. Pro synchronní komunikaci je nutné vyhradit jeden kanál pro přenos synchronizačního signálu tzv. hodin. Tento signál určuje, kdy jsou data na sběrnici připravena a mohou být dále zpracována. Interval mezi sousedními nástupnými hranami je dán rychlostí generátoru hodinového signálu. Při asynchronní komunikaci jsou data přenášena po určitých sekvencích. Ty jsou na sběrnici vkládány přesně danou rychlostí, která je nastavena při startovací sekvenci. Každé zařízení připojené ke sběrnici má v sobě zabudovaný přesný oscilátor, díky kterému snímají a posílají data v přesně definovaných intervalech. Asynchronní přenos není příliš vhodný pro velké objemy dat. Jeho výhody můžeme najít při komunikaci na velké vzdálenosti díky úspoře synchronizačního vodiče. Složitost a cena použité elektroniky je rovněž vyšší a navíc rychlost komunikace zpomaluje nutná úvodní synchronizace.

Posledním, pro nás důležitým kritériem, které nás bude zajímat je dělení na sběrnice sdílené a nesdílené. Sdílené sběrnice využívají jednu sadu vodičů pro přenos různých typů informací jako jsou data, adresy, příkazy a stavy. Počet typů informací se může u různých sběrnic lišit. Při sdíleném přenosu informací je nutné implementovat způsob, jakým se budou od sebe jednotlivé typy informací odlišovat. K tomuto se využívají tzv. identifikační signály. U nesdílených sběrnic je pro každý typ informace využívána samostatná sada vodičů.

Podle způsobu arbitráže je možno rozdělit řízení sběrnice na centralizované a decentralizované. Pokud existuje prvek bus master, který řídí sběrnici, jedná se o centralizovaný způsob. Tomuto zapojení se říká též master/slave. Jestliže jsou si všechny zařízení vzájemně rovnocenné, hovoříme o decentralizovaném způsobu řízení. Z pohledu řešené problematiky nás bude zajímat především

sběrnice typu master/slave. U těchto sběrnic musíme rozlišovat mezi dvěma rozdílnými prvky. Master — první ze zmiňovaných má za úkol řídit komunikaci a vznášet dotazy podle toho o jakou informaci usiluje. Plní tedy funkci tzv. nadřízeného. Druhým typem prvku je slave, který odpovídá na dotazy mastera. Master se většinou vyskytuje na sběrnici jeden. Obecně jich ale může být větší počet. Prvků typu slave je obecně libovolný počet. Zde závisí především na vlastnostech dané sběrnice. Občas je potřeba zajistit, aby prvek slave uvědomil o nějaké změně stavu prvek master a ten mohl na tuto událost včas reagovat. K řešení tohoto problému se využívá tzv. přerušování (interrupt), které je většinou vyvoláno změnou logické úrovně na vyhrazeném vodiči. Prvek master musí detekovat změnu stavu a včas reagovat na tuto událost. Většinou se pozastaví ostatní činnost na sběrnici a přerušování se provede prioritně.

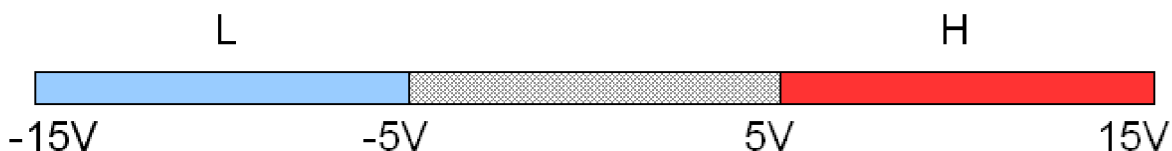
Přenos dat na sběrnici se řídí stanoveným protokolem. Ten určuje jakým způsobem se přenáší data a řeší kolizní situace. Protokol stanoví, jestli se na sběrnici posílá jako první nejvíce významný bit (MSB) nebo nejméně významný bit (LSB). Většina sběrnic má definovaný svůj vlastní protokol. Každý protokol musí definovat minimálně operace pro zahájení a ukončení komunikace, adresaci zařízení a samotný přenos dat a obsahuje také informace o elektrických standardech na kterých sběrnice komunikuje. Protokol musí být striktně dodržován a nesmí být tolerováno jiné chování než nám dovoluje. V této práci nás bude zajímat pouze sběrnice I<sup>2</sup>C a protokol RS-232, kterými se budeme zabývat dále.

## 2 Protokol RS-232

Protokol RS-232 byl jako standard schválen v roce 1969 Electronic Industries Alliance. Využívá se jako komunikační rozhraní osobních počítačů a další elektroniky. V dnešní době nalezneme jeho uplatnění především v průmyslu. V oblasti personálních počítačů je již jeho zlatá éra dávno pryč a je většinou nahrazován rychlejším a modernějším univerzálním sériovým rozhraním USB. Standard definuje synchronní i asynchronní sériovou komunikaci. Více je využívána její asynchronní varianta.

Jako první je na sběrnici poslán nejméně významný bit (LSB). Počet datových bitů je podle standardu volitelný. Nejvíce se využívá počtu 8 tedy jednoho bytu, ale můžeme se i lišit. Pro identifikaci logické úrovně se používá bipolární úroveň napětí. Bipolární znamená že logická „1“ a „0“ je reprezentována kladnou a zápornou hodnotou napájení. Různá zařízení mohou nabývat hodnot napětí  $\pm 5V$ ,  $\pm 10V$ ,  $\pm 12V$  a  $\pm 15V$ . Nejčastěji je využívána varianta  $\pm 12V$ . Napětíové úrovně na sebe přímo nenavazují, ale je mezi nimi zakázaná oblast, která není ani logickou „0“ ani „1“. Příklad rozdělení napětíových úrovní a logických hodnot je na obrázku 2.1.





2.1 Ukázka rozsahu napětí pro výstup z počítače

Na začátku každé asynchronní komunikace musí proběhnout řádná synchronizace. Každý datový byte je uvozen jedním start bitem, který přepne z klidového stavu hodnotu na lince na opačnou. Datové bity jsou ukončeny jedním paritním bitem a pak jedním nebo více stop bity. Po ukončení zasílání této sekvence se linka opět uvedena do klidového stavu.

Maximální délka vodiče pro spolehlivý přenos je 15 metrů nebo délka vodiče při kapacitě 2500pF. Pokud použijeme kvalitní kabeláž, můžeme dosáhnout délky vedení až 50 metrů. K dalšímu zvyšování délky lze využít snížení rychlosti přenosu. Následující tabulka, vypracovaná Texas Instruments, uvádí závislost přenosové rychlosti na délce vedení. Měření probíhalo v laboratorních podmínkách. V praxi by bylo nutné počítat navíc s elektromagnetickým rušením signálu, čímž by se komunikace zpomalila. Tabulka je převzata z materiálu [HWS].

| Baud rate [Bd] | Max length [ft] | Max length [m] |
|----------------|-----------------|----------------|
| 19 200         | 50              | 15             |
| 9 600          | 500             | 150            |
| 4 800          | 1 000           | 300            |
| 2 400          | 3 000           | 900            |

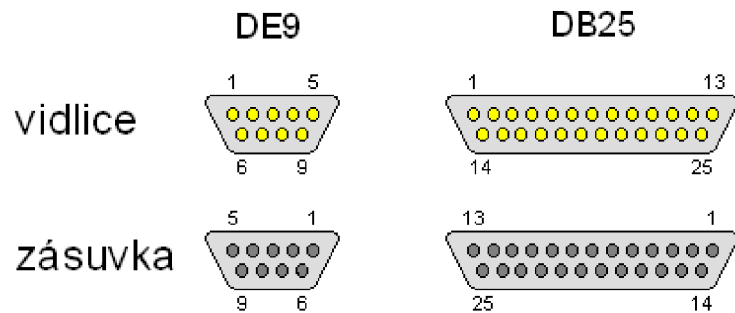
2.2 Závislost přenosové rychlosti na délce vedení

Pro přístup k sériovému rozhraní se v oblasti osobních počítačů používá obvod UART. Dříve byl tento obvod instalován na většinu základních desek. Přístup k sériovému rozhraní je ve většině OS standardně podporován.

V PC se můžeme setkat se dvěma typy konektorů sériového rozhraní. Jsou to typy DE9 a DB25. Popis, označení a funkce vývodů konektoru je v tabulce 2.3 a na následujícím obrázku je vidět rozmístění vývodů na konektorech [SER].

| Název | Směr | DE9 | DB25 | Funkce              |
|-------|------|-----|------|---------------------|
| DCD   | IN   | 1   | 8    | Carrier Detect      |
| RxD   | IN   | 2   | 3    | Receive Data        |
| TxD   | OUT  | 3   | 2    | Transmit Data       |
| DTR   | OUT  | 4   | 20   | Data Terminal Ready |
| GND   |      | 5   | 7    | System Ground       |
| DSR   | IN   | 6   | 6    | Data Set Ready      |
| RTS   | OUT  | 7   | 4    | Request to Send     |
| CTS   | IN   | 8   | 5    | Clear to Send       |
| RING  | IN   | 9   | 22   | Ring Indicator      |

2.3 Zapojení vývodů konektoru DE9 a DB25



2.4 Rozmístění pinů v konektorech vidlice a zásuvka

V této práci je použito propojení přes konektor DE9. Zapojení a použití jednotlivých vývodů je popsáno dále v kapitole o sběrnici I<sup>2</sup>C.

## 3 Sběrnice I<sup>2</sup>C

### 3.1 Obecný popis sběrnice

Tento standard pro komunikaci mezi integrovanými obvody vyvinula firma Philips. Sběrnice I<sup>2</sup>C využívá pro komunikaci dvou vodičové vedení. Prvním z vodičů je linka SDA po které probíhá přenos dat a adresy současně. I<sup>2</sup>C je sériová sdílená sběrnice s obousměrným přenosem. Pro synchronizaci se využívá linka SCL, která udává hodinový signál sběrnice. Tento signál určuje, jak rychle bude komunikace probíhat. Dolní hranice frekvence není určena. Toho můžeme využít pro krokování sběrnice a sledování její činnosti. Horní limit je dán různými režimy přenosu. Režimy a maximální povolené rychlosti jsou v následující tabulce.

| Režim           | Maximální rychlost |
|-----------------|--------------------|
| Standard        | 100 kbit/s         |
| Fast            | 400 kbit/s         |
| High-speed mode | 3,4 Mbit/s         |

3.1 Režim přenosu a jeho maximální rychlost

Sběrnice propojuje prvky typu master s podřízenými slave. Master řídí komunikaci a generuje hodinový signál na SCL. Pro naše účely byla zvolena 5-ti voltová logika. Logická „1“ je v sběrnici reprezentována napětím 5 voltů a logická „0“ napětím 0 voltů. V současné době se vyrábí velké množství různých integrovaných obvodů jako jsou A/D, D/A převodníky, řadiče LED a LCD, EEPROM, RAM, obvody pro audio a video techniku a další.

Všechny obvody I<sup>2</sup>C jsou adresovatelné. Adresa se skládá z části pevné, která je pevně dána výrobcem a části nastavitelné. Pevná část je většinou společná pro jeden druh obvodu. Díky nastavitelné části můžeme mít na sběrnici více obvodů stejného typu.

### 3.2 Protokol sběrnice

Standard definuje pravidla chování na sběrnici [I2C]. Tento protokol musí být striktně dodržován všemi zúčastněnými. Jak již bylo řečeno prvek master řídí komunikaci na sběrnici. Aby toto mohl provádět musely být definovány takové mechanismy, které umožní zahájení a ukončení komunikace s obvodem, adresaci obvodu a přenos dat.

### 3.2.1 Zahájení komunikace

Komunikace na sběrnici je zahájena speciální posloupností stavů na vodičích SDA a SCL. Podmínkou zahájení komunikace je sběrnice v klidovém stavu, tj. vodiče SDA a SCL nastavené na logickou úroveň „1“. V této fázi jsou obvody ve stavu „naslouchání“. Hodnota na SDA nejprve přechází z „1“ → „0“ a následně s určitým zpožděním SCL přechází „1“ → „0“. Po této sekvenci všechny obvody očekávají vystavení adresy cílového zařízení a další komunikaci s ním. Ustálená hodnota SCL na „1“ je od této chvíle chápána jako potvrzení bitu na SDA.

### 3.2.2 Ukončení komunikace

Stejně důležité jako zahájení komunikace je její ukončení. Za korektní ukončení spojení se považuje změna stavu na SCL z „0“ → „1“ a poté s určitým zpožděním změna SDA z „0“ → „1“. Sběrnice se tak uvede do výchozího klidového stavu a je možné zahájit další komunikaci.

### 3.2.3 Zaslání a potvrzení byte

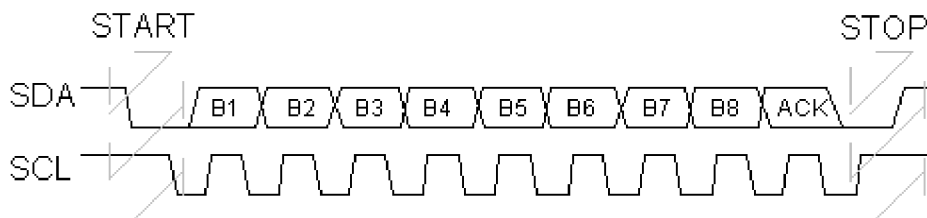
Každý zasláný byte je příjemcem potvrzován (ACK). Děje se tak 9. bitem následujícím za každým bytem. Master před každým 9. bitem nastaví SDA na úroveň logické „1“. Příjemce má poté za úkol stáhnout hodnotu na z „1“ → „0“ jako signál úspěšného doručení bytu. Poté může master pokračovat zasláním dalšího bytu nebo ukončením komunikace.

### 3.2.4 Princip komunikace

Všechny důležité prvky komunikace byly již zmíněny a je tedy čas ukázat jak samotný přenos probíhá. Na začátku je nutné provést zahájení komunikace. Po této sekvenci jsou obvody v pohotovostním režimu a očekávají adresaci cíle. Z tohoto důvodu je prvním bytem po startovací sekvenci vždy adresa cílového zařízení s nímž chceme komunikovat. Pro adresaci se využívá vždy jen horních 7 bitů. První 4 bity obsahují pevnou část a následné 3 uživatelsky nastavitelnou část adresy. Z tohoto důvodu je patrné, že na sběrnici je možno připojit pouze 27 různých obvodů. Osmý nejméně významný bit je informace, zda master hodlá data zasílat nebo přijímat od prvku slavy.

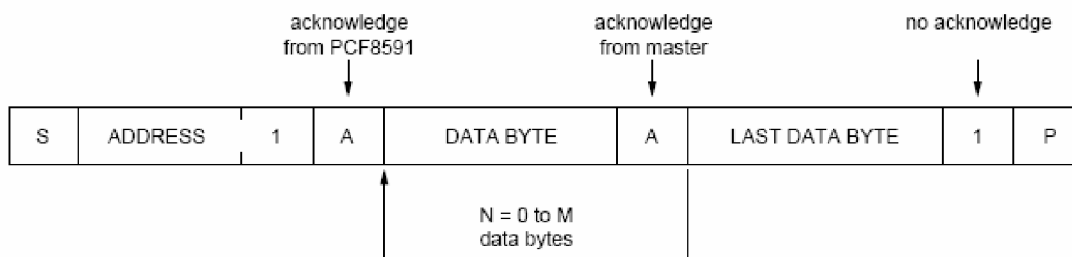
Nyní je již zcela jasné, se kterým obvodem bude master komunikovat. Ostatní obvody mohou během této doby provádět svou vlastní činnost a vyčkávají na signál ukončení předchozí akce a nové adresace zařízení. Po adresovém bytu může následovat další byte. O jaký druh byte se bude jednat, to závisí na konkrétním obvodu. Musíme rozlišovat mezi řídicím a datovým bytem. Řídicí umožňuje konkrétní nastavení obvodu pokud je to potřebné. Zde je nutné nahlédnout do dokumentace ke

konkrétnímu obvodu. Podle nastavení 8. bitu adresového byteu, je datový byte zaslán buďto od prvku master k prvku slave a nebo naopak. Po přenosu všech byteu je komunikace ukončena příslušnou sekvencí. Princip navázání komunikace, přenosu jednoho byte a ukončení spojení je demonstrováno na následujícím obrázku.



3.2 Přenos byte přes sběrnici I<sup>2</sup>C

Pro detailnější představu, jakým způsobem probíhá komunikace po sběrnici, uvedu příklad čtení hodnot z A/D převodníku PCF8591. Obrázek je přebrán z dokumentace výrobce tohoto obvodu. Začátek komunikace je uvozen startovací sekvencí (v obrázku písmeno „S“) a následným zasláním adresy zařízení, kde osmým bitem je 1. Dáváme tak převodníku najevo, že z něj hodláme číst. Obvod potvrdí zahájení komunikace acknowledge. Od tohoto okamžiku začíná převodník zasílat masteru po sběrnici datové byty, odpovídající naměřeným hodnotám analogových vstupů. Pořadí závisí na nastavení. Převodník očekává po každém zaslání datového bytu potvrzení acknowledge masterem a po jeho přijetí pokračuje dalším. Jestliže chceme komunikaci ukončit master místo acknowledge zašle „1“ a stop sekvencí končí (na obrázku písmeno „P“). Sběrnice je opět v klidovém stavu a připravena pro další komunikaci.



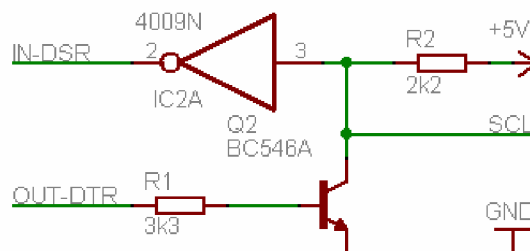
3.3 Princip čtení z A/D převodníku PCF8591

### 3.3 Propojení sběrnice I<sup>2</sup>C a PC

Princip I<sup>2</sup>C sběrnice jsme již probrali a nyní je čas připojit sběrnici k počítači. Rozhraní, přes které přistupujeme ke sběrnici, je navrženo univerzálně. Dnešní moderní počítače již velmi často nebývají vybaveny sériovým portem, proto bylo žádoucí, aby si uživatel mohl zvolit i jiné alternativy. Hlavní modul, který umožňuje komunikaci se sběrnici je vybaven jak sériovým rozhraním, tak dnes již běžně využívaným USB rozhraním. Aby nebyla konstrukce příliš náročná, je při využívání

propojení přes USB využito konvertoru FT232BM, který převádí signál na sériové rozhraní. Díky tomuto obvodu je možno jednoduchým způsobem připojit aplikace navržené pro komunikaci přes sériový port na toto universální sériové rozhraní. Při takovémto připojení se v PC objeví nové virtuální rozhraní, se kterým můžeme komunikovat stejným způsobem jako s klasickým sériovým portem. Je potřeba mít pouze nainstalovaný příslušný ovladač. FT232BM využívá na výstupu 5-ti voltovou logiku a navíc signál invertuje. Tento stav je žádoucí a pro propojení se základním modulem pro sériovou komunikaci je nutné zapojit pouze logický obvod AND.

Jak jsem již v předchozím textu uvedl, budeme pracovat s 5-ti voltovou logikou. Sériová sběrnice ovšem pracuje na úrovních  $\pm 12$  voltů a proto budeme muset napěťové úrovně konvertovat. Nejjednodušším způsobem jak tohoto docílit, je využít obvod MAX 232, který je k tomuto určen. Zapojení je naznačeno v příloze. Obvod MAX 232 má tu vlastnost, že invertuje na výstupu logické hodnoty. Abychom dostali požadované úrovně musíme signál invertovat. Na trhu je celá řada invertorů ze kterých si můžeme vybrat. V naší aplikaci budu pracovat s invertorem HCF4009. Tento invertor má tzv. „tvrdý výstup“, což znamená, že pokud je na výstupu logická 0, je jakoby natvrdo propojen se zemí, když 1 je propojen s napájecím napětím. Toto zapojení při nastavování úrovní na sběrnici není možné. Když např. master uvolňuje sběrnici (nastaví ji na logickou 1), je nutné, aby slave mohl potvrdit přijetí bytu snížením na logickou 0. Invertor je proto možné využít pouze při čtení hodnot ze sběrnice. Na odchozí lince je nezbytné využít zapojení s tzv. pull-up rezistorem. Zapojení je znázorněno na obrázku 3.4. Pro ukázkou je uvedeno zapojení linky SCL. Signály DTR a DSR procházejí přes převodník MAX 232. Pomocí výstupu DTR je nastavována hodnota na SCL. Hodnota se konvertuje přes převodník z  $\pm 12$ V na 0-5V a je veden na bázi tranzistoru Q1. Pokud je na bázi přivedeno napětí, sepne se spínač kolektor  $\rightarrow$  emitor a na SCL je generována logická hodnota 0. Toto zapojení má výstup s tzv. „otevřeným kolektorem“. Pull-up rezistor je v zapojení označen jako R2. Pokud budeme chtít hodnotu ze sběrnice číst, signál prochází přes invertor, kde již nevádí „tvrdý výstup“ a je po konverzi přiveden na vstupní pin DSR sériového portu. Obdobně by vypadalo zapojení pro datovou linku SDA.



3.4 Zapojení s otevřeným kolektorem

Propojení sériového rozhraní a sběrnice je znázorněno v následující tabulce. Bylo nutné dodržet vstupní a výstupní zapojení sériového rozhraní.

| Pin  | Číslo pinu | Zapojení  | Popis                          |
|------|------------|-----------|--------------------------------|
| DTR  | 4          | PC → SCL  | Nastavuje hodnotu na lince SCL |
| DSR  | 6          | PC ← SCL  | Čte hodnoty z linky SCL        |
| RTS  | 7          | PC → SDA  | Nastavuje hodnotu na lince SDA |
| CTS  | 8          | PC ← SDA  | Čte hodnoty z linky SDA        |
| RING | 9          | PC ← INT0 | Stav linky INT0 pro přerušení  |
| DCD  | 1          | PC ← INT1 | Stav linky INT1 pro přerušení  |

3.5 Tabulka propojení vstupů a výstupů sériového portu se sběrnicí

## 4 Řídící aplikace

### 4.1 Základní princip

Úkolem této podkapitoly je popsat základní řídicí prvek sběrnice. Tímto prvkem je server, napsaný v jazyce C++ a pracující pod operačním systémem linux [LS]. Aplikace umožňuje abstraktní přístup ke sběrnicí a její vzdálené řízení přes webové rozhraní. Objektový model byl navržen s ohledem na co největší znovupoužitelnost kódu pro možnost budoucího rozšíření o další moduly. V následujících podkapitolách bude postupně popsána funkce celé aplikace, ukázány dílčí části nezbytné pro její chod a v závěru rozebrány objekty pro připojení konkrétních modulů na sběrnicí.

### 4.2 Popis činnosti aplikace

V následující části objasním možnosti řídicí aplikace. Protože tato aplikace pracuje jako vícevláknová, bylo nutné zajistit dokonalou synchronizaci jednotlivých vláken při přístupu ke sdíleným zdrojům. Využito bylo prostředků pro meziprocesovou komunikaci a synchronizaci, jako jsou např. mutexy [LIN]. Nastavení aplikace je prováděno přes databázový server MySQL. Tento typ serveru byl zvolen pro jeho licenční podmínky a jednoduchou možnost začlenění do programovacího jazyka C++. Ukládání do databáze bylo vybráno na základě požadavku uchovávání pravidelných statistik z některých senzorů, které tuto vlastnost umožňují.

Při spuštění serveru je nutné zadat parametry pro přístup do MySQL databáze. Program se připojí do databáze, kde je seznam jednotlivých entit na sběrnicí, vytvoří potřebné objekty a spustí jejich pravidelně se opakující akce. Pro opakující činnosti, jako např. pravidelné snímání teploty a



uchovávání naměřené statistiky, byla vytvořena konstrukce kalendáře, do kterého se mohou tyto akce zaznamenat a samy se v daném čase aktivují. Při vytváření objektu jsou základní údaje o entitách uchovány v tabulce zařízení. Po této počáteční fázi a inicializaci všech obvodů jsou nastartovány ostatní vlákna aplikace.

Vlákna jsou celkem tři. První z nich je pro obsluhu již zmíněného kalendáře a aktivaci naplánovaných akcí v něm. Detailnější popis je v podkapitole číslo 4.3. Druhé vlákno zajišťuje interakci s webovým rozhraním. Tato aplikace funguje jako server. Přes sockety komunikuje s okolním světem a webový klient má možnost se k aplikaci připojit a přes specifický protokol ovládat jednotlivé činnosti na sběrnici. Poslední vlákno obsluhuje vzniklá přerušení. Některá zařízení mohou generovat přerušení a je nutné, v co nejkratší době, tyto události obsloužit. Aplikace při detekci přerušení dokončí všechny běžící činnosti na sběrnici a identifikuje zařízení, které jej vyvolalo. Patřičnou protiakcí zareaguje a dále pokračuje v běžné činnosti. Přerušení mohou vyvolat např. uživatelé při stisku tlačítka nebo některé senzory reagující na náhlou změnu prostředí.

Uživatel má možnost na tuto univerzální sběrnici připojit libovolné entity komunikující I2C protokolem. Každý prvek musí mít unikátní adresu. Samotné nastavení je prováděno ve specifické tabulce databáze. Jednotlivé parametry a jejich význam je popsán v části 4.3.4.

Celý program běží v nekonečné smyčce. Po stisknutí jakékoliv klávesy se celý server korektně ukončí.

## 4.3 Popis dílčích částí

### 4.3.1 Kalendář

Kalendář je datová struktura, umožňující práci s pravidelně se opakujícími činnostmi. Často používaným příkladem může být např. pravidelné snímání teploty senzorem a uchovávání průběžné statistiky v databázi. S těmito daty můžeme poté dále pracovat, vytvářet grafy, hledat maximální, minimální a průměrné hodnoty měření apod.

Vytvořen je jeden pro celou aplikaci. Jelikož s kalendářem může pracovat více vláken, bylo nutné zamezit jejich současnému přístupu. Využito bylo tzv. mutexu, který je pro tento problém určen. Při volání metody „Activate“ která je dostupná pro všechny zařízení připojené na sběrnici, je do kalendáře uložen záznam o časové aktivaci. Kalendář pracuje v podstatě jako lineární seznam, kde prvky jsou řazeny podle časové značky, kdy se mají aktivovat a podle velikosti priority. Entity, které se mají aktivovat dříve, jsou ukládány před entity s pozdější aktivací. Pokud jsou časy shodné, rozhodování se řídí prioritou. Více prioritní jsou před méně prioritními. Každý záznam v kalendáři rovněž nese informace o identifikaci zařízení a potřebnými parametry pro aktivaci naplánované události.

Aktivace je prováděna specifickým vláknem. Nekonečná smyčka neustále prochází záznamy v kalendáři a pokud narazí na prvek, jehož časová značka je menší nebo rovna aktuálnímu času, zavoláme metodu `PeriodBehavior`, která vykoná požadovanou akci. Ta je specifická pro každé zařízení. Pokud je sběrnice zrovna využívána zařadí se do fronty a čeká na její uvolnění.

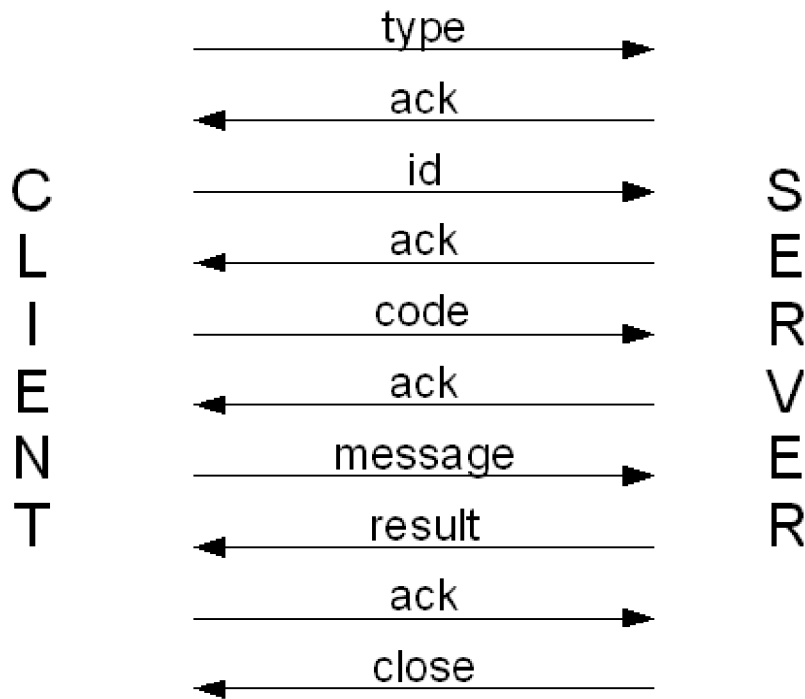
### **4.3.2 Vzdálené řízení**

Jednou ze zásadních vlastností celé aplikace je komunikace s okolním světem. V úvahu připadalo několik ověřených metod, jako jsou pojmenované roury, sdílené soubory, sockety apod. Po prostudování dostupných materiálů [COM] a s ohledem na předchozí zkušenosti byla zvolena forma socketu. Výhodou je rovněž jejich podpora v jazyce php, ve kterém byla plánována konstrukce webového rozhraní.

Komunikace přes socket běží opět ve vlastním vlákně. Klienti se mohou na server připojovat a řídit tak chování celé sběrnice. Ovládání je řešeno podle přesně daných pravidel, která jsou specifikována navrženým komunikačním protokolem. Ten určuje chování a reakce na definované příkazy. Komunikace přes sockety slouží k interakci s uživatelem. Po přijetí potřebných dat pro provedení operace, vyhledá server v tabulce zařízení žádanou entitu a zavolá s příslušnými parametry její metodu `Behavior`. Ta již provede všechny požadované operace. Pokud je výsledkem operace rovněž nějaká hodnota, zasílá server jako odpověď zprávu klientovi. Server neumožňuje souběžné připojení více klientů. Jelikož je aplikace určena pro demonstraci činnosti sběrnice neřešila se otázka zabezpečení.

### **4.3.3 Komunikační protokol**

Aby bylo možné přesně a jasně odlišit jednotlivé příkazy a vnést do komunikace přes sockety nějaký řád, bylo nezbytné vytvořit komunikační protokol. Jeho obsahem jsou smluvené signály, určující počátek a ukončení komunikace, přenos informací mezi klientem a serverem, ale také mechanismy umožňující kontrolu správného vykonání předchozí operace.



4.1 Komunikační protokol

Komunikace se serverem je zahájena zasláním řetězce „type“, který odlišuje typ zařízení, na které hodláme aplikovat danou operaci. Přijetí typu je potvrzeno pomocí „ack“. Jako další hodnota je přeneseno „id“ zařízení. Toto „id“ odpovídá číslu zařízení, které je uloženo v databázi. Umožňuje nám odlišit jednotlivá zařízení stejného typu. Server opět potvrzuje přijetí řetězce. Následuje hodnota „code“, jíž specifikujeme funkci zařízení, které bychom rádi ovládali. Např. u LCD displeje je to zobrazení zvoleného textu, vymazání displeje nebo ovládání zapnutí a vypnutí podsvětlení. Po zaslání potvrzení přijímá server zprávu message, která umožňuje bližší specifikaci požadované operace (např. text, který se má zobrazit na LCD). Nyní server vykoná požadovanou operaci a pokud je jejím výsledkem nějaká zpráva, doručí ji klientovi. Ten potvrdí „ack“ a server ukončuje celou komunikaci zprávou „close“. Pokud provedená operace nevyžaduje žádnou odpověď od klienta, server ukončuje komunikaci rovnou a nezasílá žádný „result“.

#### 4.3.4 Databáze a práce s ní

Počáteční inicializace se řídí hodnotami v databázi MySQL. Nyní se dostáváme do části, která tuto problematiku osvětlí. Abychom mohli do databáze přistupovat a využívat tak její data, je nutné do projektu připojit hlavičkový soubor <mysql.h>. Nyní již můžeme, poměrně transparentním způsobem, manipulovat s požadovanými daty podobně, jak to známe např. z jazyka php. Server se spouští s parametry specifikujícími připojení do databáze I2C. V ní se nalézají 3 tabulky. Postupně se podíváme na struktury těchto tabulek a významy jejich sloupců.

První tabulku, kterou si popíšeme je tabulka „devices“. Ta nese informace o všech zařízeních připojených na sběrnici a je využívána při počáteční inicializaci aplikace. Struktura je znázorněna na následujícím obrázku.

| DEVICES |      |         |           |         |
|---------|------|---------|-----------|---------|
| id      | type | address | actualize | connect |

4.2 Tabulka DEVICES

Prvním atributem je primární klíč „id“. Je unikátní pro každé zařízení a podle něj odlišujeme přístup k jednotlivým elementům. Svou hodnotou odpovídá hodnotě uložené v tabulce zařízení. Druh zařízení je určen záznamem „type“. Jeho hodnoty jsou cizím klíčem do tabulky TYPE. Každý typ má své specifické chování a přistupuje se k němu s určitým způsobem. Podle něj se generuje vizuální podoba stránky klienta a ovládání elementu. Třetím atributem je adresa. Ta musí být opět unikátní, aby nedocházelo ke kolizi na sběrnici. Je to jednobytová hodnota. Skládá se z pevné a nastavitelné části, která je určena pro každý obvod. Tuto hodnotu je nutné nastudovat podle dané dokumentace. Předposledním sloupcem tabulky je hodnota „actualize“. Důležitá je pouze pro entity, které provádí nějakou periodicky se opakující činnost. Pokud nic takového zařízení nepovoluje je nastavena implicitní hodnota na 0. Poslední sloupec „connect“ odpovídá logické hodnotě true/false podle toho zda je daný prvek připojen na sběrnici či nikoliv.

Druhou důležitou tabulkou je tabulka TYPE. Obsahuje označení jednotlivých typů zařízení připojených na sběrnici. Jelikož je se podle typu v aplikaci určuje vytvářený objekt je nutné aby se údaje shodovaly.

| TYPE |      |
|------|------|
| id   | type |

4.3 Tabulka TYPE

Poslední tabulka je určena pro zaznamenávání statistiky z teplotních čidel. Teplota je do nich ukládána pravidelně po určité periodě podle nastavení. Díky této tabulce máme možnost vykreslit graf průběhu teploty, vyhledávat maximum, minimum či průměrné teploty. Můžeme procházet historii apod. Každý záznam je nositelem informace o přesném datu a čase, kdy bylo měření provedeno, identifikačního čísla senzoru a naměřené teplotě. Struktura je ukázána v následující tabulce.

| TEMP |      |        |             |
|------|------|--------|-------------|
| id   | date | device | temperature |

4.4 Tabulka TEMP

## 4.4 Třída SerialPort

Základní třídou, která umožňuje přístup k sériovému portu, je SerialPort. Spojuje všechny metody, které jsou nutné pro ovládání.

Pro otevření portu byla vytvořena metoda OpenSerial. Ta otevře rozhraní pro komunikaci podle zadaného jména a nastaví získaný Handle, tedy jakýsi ukazatel na zařízení. Po zavolání této metody je možno zahájit komunikaci. Pokud již nechceme s portem dále pracovat, můžeme zařízení uzavřít metodou CloseSerial.

Jak je z názvu patrné, metody SetDTR a SetRTS nastavují logickou hodnotu na sběrnici. Pro čtení máme možnost využít metody GetCTS, GetDSR, GetRING a GetDCD. Jména odpovídají pinům sériové sběrnice. K přístupu k sériovému portu je využito funkce ioctl(), která nám umožňuje manipulaci se speciálními soubory, tedy i portem.

## 4.5 Třída TI2C

Je potomkem třídy SerialPort. Zapouzdřuje základní příkazy pro řízení sběrnice I<sup>2</sup>C. Nezbytnými funkcemi je vykonání startovací či ukončující sekvence, zaslání bajtu, nebo její přečtení. Před komunikací po sběrnici je nutné otevřít sériový port a provést inicializaci. Pokud vše proběhlo v pořádku je sběrnice v klidovém stavu, tj. SDA i SCL jsou na úrovních logické 1. Konstrukcí SINGLETON je zajištěno, že objekt umožňující manipulující se sběrnici byl vytvořen pouze jeden. Zabrání se tak kolizi při komunikaci. Jelikož je aplikace vícevláknová a ke sběrnici je možné přistupovat z více směrů, bylo nezbytné zabezpečit ochranu pomocí MUTEXU.

## 4.6 Objekty zařízení

Abychom byli schopni ovládat jednotlivá zařízení na sběrnici bylo nezbytné vytvořit patřičné objekty. Všechny objekty vycházejí ze třídy TElement. Ta sjednocuje všechny společné vlastnosti jako je práce s adresou zařízení, kalendářem apod. Bližší informace o této třídě jsou dostupné v programové dokumentaci. Pro ukázkou práce se sběrnici bylo vytvořeno ovládání pro dva druhy objektů – teplotní senzor a LCD display. Stručně si popíšeme jejich unikátní chování.

## 4.6.1 Objekt TLCD

Při tvorbě této třídy bylo nutné si uvědomit, jaké možnosti ovládání budou dostupné uživateli. Z praktického používání vyplynulo, že bude potřeba zobrazovat text na displeji, zapínat a vypínat jeho podsvětlení. LCD displej je připojen ke sběrnici přes 16bitový expander MCP23016. Vznikla tedy třída TMPC23016 zajišťující tuto komunikaci. Expander zobrazuje binárně na výstupech dva byty. Přes tyto výstupy je připojen LCD displej a díky vlastnímu řadiči umožňuje jeho jednoduché ovládání. Po vytvoření objektu je nutné provést počáteční nastavení řadiče. Zavoláme proto metodu `initLCD()`. Chování je ovládáno metodou `Behavior`, ke které vzdáleně přistupujeme přes sockety.

## 4.6.2 Objekt TTherm

Teplotní čidlo je připojeno přes A/D převodník PCF8591. Podle zadané periody je prováděno pravidelné měření teploty a její ukládání do databáze. Na vyžádání uživatele je možné změřit aktuální teplotu a předat přes socket php klientovi, který ji patřičným způsobem zobrazí. Při měření je nutné každé čidlo zkalibrovat. To se provádí na úrovni serverové aplikace. Přes webové rozhraní je možné kalibraci nastavovat. Přesnost měření odpovídá polovině stupně Celsia. Popis samotného čidla je v části věnující se fyzickému zapojení sběrnice a zařízení na ní připojených.

# 5 Webové rozhraní

Webové rozhraní slouží ke komfortnímu řízení prvků na sběrnici. Každé zařízení má podle typu možnosti nastavení a ovládání. V postranním menu jsou na výběr tři možnosti.

První slouží k ovládání připojených zařízení. Skript podle databáze vygeneruje zobrazení každého objektu a způsob přístupu k jeho možnostem obsluhy. Pro ukázkou jsou vytvořeny dva objekty. Jeden pro řízení zobrazování na LCD a druhý pro snímání teploty z čidla. U teplotního senzoru je navíc možno zobrazit graficky historii měřených teplot a provádět kalibraci.

Druhou záložkou je setup. Slouží k nastavení připojených modulů a přístupu k I2C serveru. Před prvním použitím je nutné zadat číslo portu a IP adresu serveru. Pokud se klientovi nepodaří se na server připojit, je o tomto stavu uživatel informován chybovou hláškou. Nebyla řešena otázka zabezpečené komunikace přes protokol I2C. Ovládání bylo navrženo s ohledem na komfort a jednoduchost práce.

Poslední položka menu informuje o způsobu ovládání přes webové rozhraní.

## 6 Konstrukce modulů a jejich ovládání

Konstrukce jednotlivých modulů je klíčová věc pro chod celé sběrnice. Každé takovéto zařízení musí mít unikátní adresu, aby nedošlo ke kolizi. Při návrhu byl kladen důraz hlavně na dostupné obvody komunikující s protokolem I<sup>2</sup>C. Těch je na trhu opravdu mnoho a proto jsem výběr provedl také s ohledem na jejich cenu. Samotná zapojení jsou ve většině případů velice jednoduchá a nemají žádnou vlastní logiku. Ta byla implementována až do řídicí aplikace na úrovni osobního počítače.

V následujících podkapitolách si postupně popíšeme zapojení modulů pro něž bylo vytvořeno ovládání v aplikaci. Rozebereme si základní chování a jeho vlastnosti. Podrobné informace o jednotlivých obvodech naleznete v dokumentaci výrobce.

### 6.1 LCD modul

Slouží jako externí zobrazovací jednotka. Na trhu jsou k dispozici LCD displeje maticové alfanumerické, grafické a zákaznické.

Alfanumerické mají zobrazovací plochu rozdělenou na řádky a sloupce. Každý prvek takovéto tabulky je maticí bodů, kterou ovládáme zobrazení požadovaného znaku. Tento typ většinou obsahuje logiku pro ulehčení práce s displejem a paměť s používanými alfanumerickými znaky.

Grafické displeje mají plochu složenou z jednotlivých bodů. Matice pokrývá celou viditelnou oblast a můžeme tak ovládat každý bod samostatně. Jsou náročnější na obsluhu než alfanumerické. Můžeme však vykreslovat jakékoliv obrazce.

Zákaznické obsahují speciální motivy podle přání klienta. Jsou určeny do předem dané série výrobků. Příkladem mohou být displeje v mobilních telefonech, hodinkách či kalkulačkách.

Naše aplikace pracuje s alfanumerickým displejem MC1602. Disponuje velikostí 16x2 znaků, má vlastní řadič HD44780, který je charakterizován následujícími vlastnostmi:

- zobrazuje znaky pomocí matice bodů a ovládá je sadou příkazů,
- má rozhraní pro 4 nebo 8-bitovou datovou sběrnici,
- obsahuje paměť a generátor znaků,
- umožňuje uživateli vytvářet vlastní znaky a symboly.

#### 6.1.1 Zapojení vývodů LCD

Chceme-li s displejem komunikovat, je nutné připojit se na jeho rozhraní. Tento typ LCD má celkem 16 vývodů, jejichž funkci si pro snadnější pochopení komunikace objasníme v následující tabulce.



| Číslo vývodu | Název | Popis funkce  |
|--------------|-------|---|
| 1            | Vss   | napájení GND  |
| 2            | Vdd   | napájení +5V  |
| 3            | Vo    | kontrast 0V .. 5V (typ. 0,8V)                           |
| 4            | RS    | registr Select (0 = instrukce, 1 = data)                |
| 5            | R/W   | zápis/čtení z LCD (0 = zápis, 1 = čtení)                |
| 6            | E     | řídící impuls platnosti dat na sběrnici (hrana z 1 → 0) |
| 7            | DB0   | datový bit 0 – LSB                                      |
| 8            | DB1   | datový bit 1  |
| 9            | DB2   | datový bit 2  |
| 10           | DB3   | datový bit 3  |
| 11           | DB4   | datový bit 4  |
| 12           | DB5   | datový bit 5  |
| 13           | DB6   | datový bit 6  |
| 14           | DB7   | datový bit 7 – MSB                                      |
| 15           | A     | podsvícení - anoda LED (+5V)                            |
| 16           | K     | podsvícení - katoda LED (GND)                           |

### 6.1 Zapojení vývodů LCD

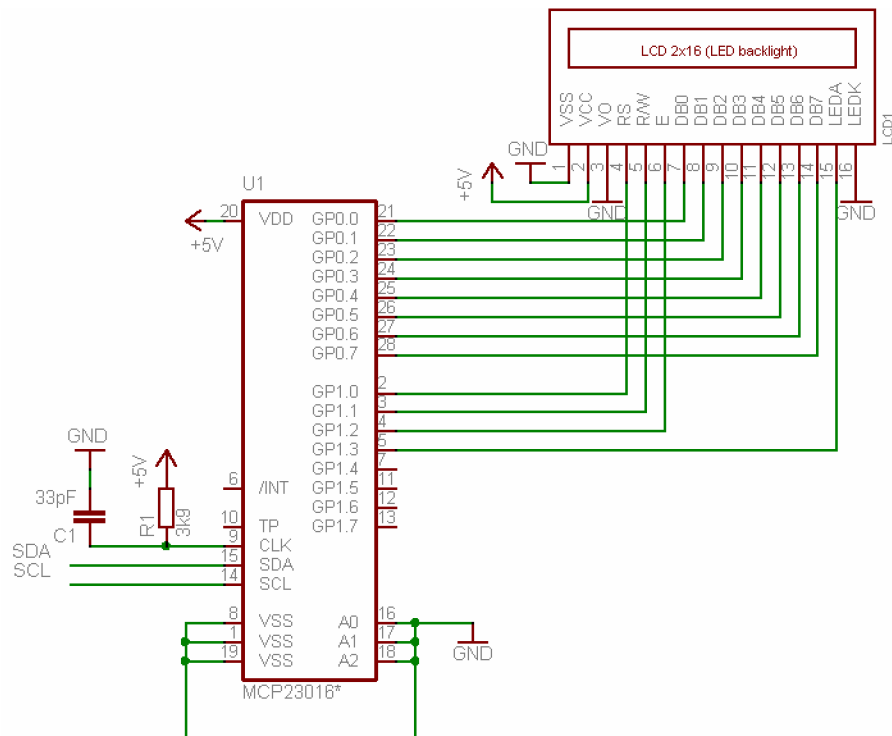
Řadič umožňuje, aby komunikace probíhala po 4 nebo 8-bitové sběrnici. Při 4 bitové komunikaci jsou první 4 bity uzemněny a přenos probíhá pouze po datových vodičích 4 až 7. Každý byte je pak rozdělen na dvě části. Jako první se zašlou vyšší 4 bity a potvrdí sestupnou hranou Enable (vodič č. 6) a pak nižší 4 bity rovněž potvrzené Enable. Komunikace po 4-bitové sběrnici je dvakrát pomalejší než po 8-bitové.

#### 6.1.2 Instrukce pro řízení LCD

Řízení displeje probíhá podle přesně daných instrukcí. Je tak značně ulehčena komunikace s LCD. Instrukční sady se mohou u různých řadičů lišit. Před použitím LCD je třeba seznámit se s příslušnou dokumentací a způsobem zapojení daného modelu. Po zapnutí sestavy je nezbytné nastavit správnou šířku sběrnice a registry, které jsou nutné pro správný chod displeje. Pro získání bližších informací o instrukcích a komunikaci přes řadič HD44780 je vhodným studijním zdrojem [LCD].

### 6.1.3 Propojení s I<sup>2</sup>C sběrnici

Z předchozího rozboru vyplynulo, že displej komunikuje přes 4 nebo 8-bitovou sběrnici. Není proto možné připojit LCD přímo na I<sup>2</sup>C. Jako propojovací člen byl vybrán 16-bitový expander MCP23016. Má dostatečný počet výstupů pro 8-bitovou komunikaci s LCD a jeho řízení.

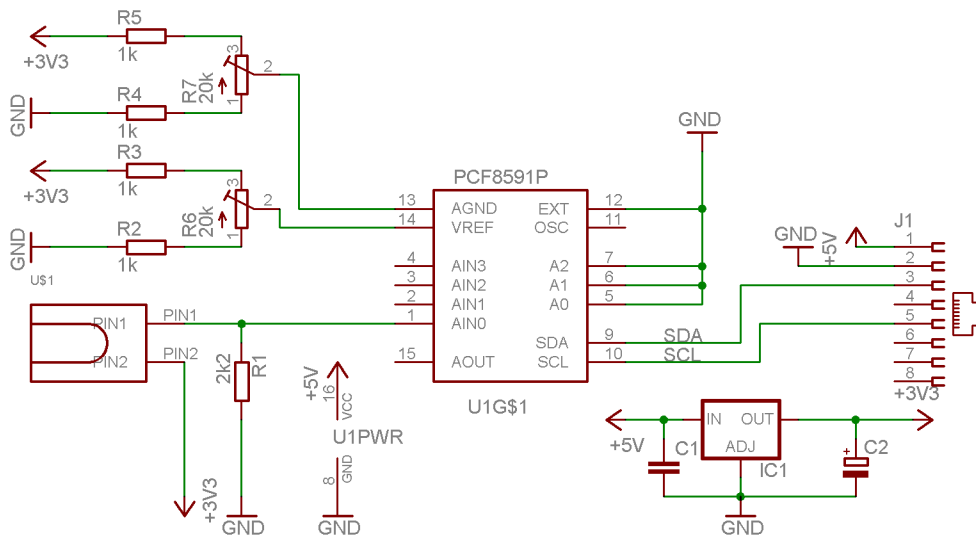


6.2 Schéma propojení expanderu MCP23016 a LCD displeje

Expander může pracovat v režimu čtení nebo zápisu hodnot na vstupně výstupních konektorech. Před komunikací je nutné správně nastavit vnitřní registry. Obvod umožňuje generování přerušení, které je však v našem zapojení nevyužité. Informace, jakým způsobem je možné s expanderem komunikovat, je nutné vyhledat v popisu obvodu dodávaného výrobcem.

## 6.2 Teplotní senzor

Pro účely snímání teploty byl vyvinut speciální senzor. Měření je prováděno pomocí polovodičového křemíkového čidla KTY81-210. Termistor je zapojen v napěťovém děliči a výsledná hodnota odporu je měřena pomocí A/D převodníku. Jako převodník slouží obvod PFC8591.



### 6.3 Zapojení napěťového děliče a A/D převodníku

Abychom mohli dosáhnout přesnosti aspoň 0,5°C, je potřeba nastavit A/D převodník. Přes vodiče přivádíme napájecí napětí 5 voltů. Jelikož s ním pracuje více obvodů, nemůžeme si být jistí jeho přesnou hodnotou, kterou ovšem potřebujeme pro správný výsledek A/D převodu. Využil jsem proto přesného stabilizátoru, který zajistí napětí 3,3 volty.

Výpočtem zjistíme přibližné hodnoty, maxima a minima napětí, které bude možné naměřit na děliči napětí. Pro dělič platí vztah 
$$U = \frac{U_{CC} \times R}{R + R_T}$$
. Po dosazení za  $U_{CC} = 3,3 \text{ V}$ ,  $R = 2200\Omega$  a  $R_T$  podle

dokumentace, pro největší a nejmenší měřitelnou teplotu, získáme interval měřitelného napětí. V mé konstrukci bylo použito napětí  $AGND = 1,3\text{V}$  a  $VREF = 2\text{V}$ , odpovídající teplotnímu rozsahu  $-20^\circ\text{C} - 100^\circ\text{C}$ . K nastavení slouží dva trimry. Využijeme tak celý rozsah převodníku. Dalším krokem, který bylo nutno udělat, je kalibrace čidla. Ta proběhla metodou postupného měření a sestavení polynomu. Voda v nádobě byla uvedena do bodu varu a pomocí laboratorního teploměru a našeho čidla, byla zjištěna binární hodnota A/D převodníku odpovídající změřené teplotě. Do vroucí vody se postupně přimíchávala studená a po ustálení jsem zaznamenával hodnoty na převodníku. Ze získaných hodnot byl sestaven polynom, který je základem funkce výpočtu teploty. Počet měření odpovídá konečné přesnosti polynomu. Tato metoda má tu výhodu, že hodnota přesně odpovídá našemu čidlu. Nyní již stačí dosadit do polynomu hodnotu, získanou A/D převodem a výsledek výpočtu je aktuální teplota. Nepřesnost měření je z důvodu nedostupnosti přesných laboratorních podmínek.

## 6.2.1 Čidlo KTY81-210

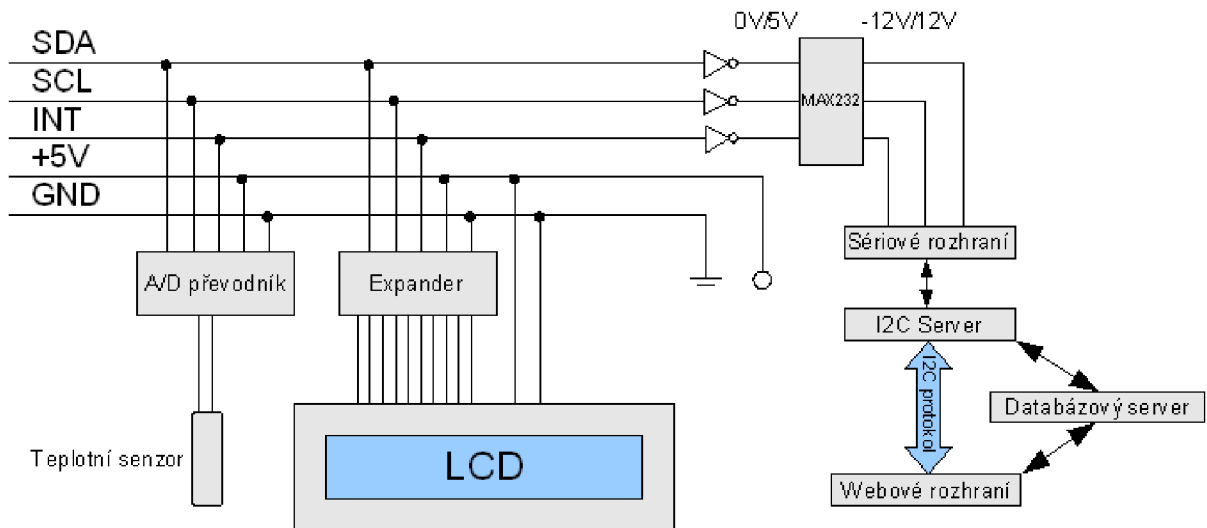
Při výrobě čidla se využívá polovodiče typu N, tedy s dominantní elektronovou vodivostí. Pohyblivost elektronů v krystalové mřížce závisí na teplotě. Při rostoucí teplotě dochází vlivem rozptylu nosičů náboje ke zmenšování pohyblivosti těchto nábojů a díky tomu narůstá odpor čidla. Křemíkové senzory jsou hodné pro měření teplot v rozsahu -50 až 150 °C. Teplotní součinitel odporu je téměř konstantní, jeho střední hodnota se pohybuje kolem 0,01 K<sup>-1</sup>. Výhodou je dobrá dostupnost a přijatelná cena. Čidlo není úplně lineární, ale nelinearitu lze odstranit vhodnými metodami.

## 6.2.2 PCF8591

Teplotní čidlo je k I<sup>2</sup>C sběrnici připojeno přes 8-bitový A/D a D/A převodník. Tento obvod má v sobě implementován jeden D/A výstup a 4 A/D vstupy. Funkce analogových vstupů je programovatelná. Převádí vstupní napětí v intervalu 0 – V<sub>REF</sub> na číselnou hodnotu délky jeden byte. V zapojení s teplotním čidlem pro nás bude důležitá pouze činnost A/D převodníku. Čtení může probíhat dvěma způsoby. Prvním je cyklické čtení ze všech vstupů, kde prvním bytem je vždy hodnota z předchozí komunikace. Tento způsob dokáže uspořit čas, nutný pro navazování komunikace, pokud potřebujeme hodnoty snímat z více vstupů současně. Dalším způsobem je nastavení vstupu pomocí řídicího byteu a čtení hodnoty pouze odtud. Podrobnosti jsou opět dostupné v příslušné dokumentaci.

# 7 Propojení jednotlivých částí

Samotné části systému byly vysvětleny v předchozích kapitolách a nyní si ukážeme jejich propojení do výsledného funkčního systému. Na zjednodušeném obrázku je možno si představit, jak komunikace a řízení probíhá. V popisu začneme webovým rozhraním. Uživatel zadá na rozhraní požadovanou operaci, která je přes I2C protokol doručena aplikačnímu serveru. Ten tuto zprávu dekóduje a spustí akci, která přes sériový port komunikuje s moduly zapojenými na sběrnici. Pokud má operace nějaký výsledek, např. získaná teplota z čidla, zašle server odpověď webovému rozhraní, které ji zpracuje. Provede třeba zobrazení naměřené teploty příslušným způsobem. Na schématu je také naznačena komunikace s databázovým serverem.



7.1 Schéma propojení funkčního systému

## 8 Závěr

Cílem této práce bylo praktické vytvoření sestavy se sběrnici I<sup>2</sup>C, která bude připojitelná k počítači a bude možné ji dálkově ovládat přes webové rozhraní. Byl vytvořen jednoduchý a snadno sestrojitelný adaptér, který má vysokou stabilitu a bezchybný provoz. Na tuto sběrnici je možné zavěsit jakékoliv zařízení pracující na rozhraní I<sup>2</sup>C. Komunikace přes sockety a databázový server umožnila monitoring, ovládaní přes internet a ukládání průběžné statistiky. Tato práce byla velkým přínosem pro pochopení principu fungování sběrnic a konstrukce vícevláknových aplikací. Vznikl komunikační protokol a došlo tak k propojení aplikace a internetového rozhraní. Navržené řešení neslouží pouze jako demonstrace činnosti, ale je možné jeho široké využití při praktickém řešení technických sestav, jejichž provoz má být vzdáleně diagnostikován a řízen. Podle šablony lze snadno doprogramovat ovládání jakéhokoliv modulu. Při tvorbě jsem využil zkušenosti a vědomosti, které jsem nabyl v mnoha předmětech na fakultě, především z oblasti programování a návrhu hardware.

# Literatura

- [HWS] Olmr, V., Půhoný, J.: *HW server představuje - RS-232* [online], HW server, 2003.  
[cit. 4.5.2007], Dostupný z WWW: < <http://rs232.hw.cz/> >
- [I2C] Haša, J. *Sběrnice I<sup>2</sup>C a její využití*, 2003
- [LS] Sweet, M.: *Serial Programming Guide for POSIX Operating Systems* [online], 2002.  
[cit. 4.5.2007], Dostupný z WWW: < <http://www.easysw.com/~mike/serial/serial.html> >
- [SER] Krc, P.: *RS-232 (EIA-232) serial interface pinout* [online], Pinouts.ru team, 2006.  
[cit. 4.5.2007], Dostupný z WWW: < <http://pinouts.ru/SerialPorts/> >
- [COM] Hall, B.: *Beej's Guide to Unix Interprocess Communication* [online], 1997. [cit. 4.5.2007],  
Dostupný z WWW: < <http://www.ecst.csuchico.edu/~beej/guide/ipc/> >
- [LIN] Matys, J.: *Seriál Programování pod Linuxem pro všechny* [online], Internet Info, s.r.o., 2004.  
[cit. 4.5.2007],  
Dostupný z WWW: < <http://www.root.cz/serialy/programovani-pod-linuxem-pro-vsechny/> >
- [LCD] Strolený, J.: *Znakové LCD displeje* [online], DOVEDA BOYS, 2007. [cit. 4.5.2007],  
Dostupný z WWW: < <http://www.cmail.cz/doveda/lcd/> >

## Seznam příloh

Příloha 1. Schéma zapojení základního modulu

Příloha 2. Programová dokumentace

Příloha 3. Zdrojové texty

Příloha 4. DVD



# Příloha 1

## Schéma zapojení základního modulu

