

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Zpracování a webová prezentace dat z meteostanice

Processing and web presentation of data from weather station

Diplomová práce

Autor: **Bc. Petr Holub**
Vedoucí práce: doc. Ing. Josef Chaloupka, Ph.D.
Konzultant: Ing. Ondřej Hnilička

V Liberci 2012

Originální zadání práce

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Tímto bych chtěl poděkovat doc. Ing. Josefovi Chaloupkovi, Ph.D. a Ing. Ondřeji Hniličkovi za jejich rady, trpělivost a vedení při realizaci diplomové práce. Dále bych chtěl poděkovat Ing. Vladimírovi Mocovi, který mi umožnil přístup k meteorologické stanici v Kateřinkách. Další komu chci touto cestou poděkovat je osazenstvo PCB labu za rychlou a kvalitní výrobu desky plošných spojů pro I/O modul.

Abstrakt

Práce se zabývá zpracováním a webovou prezentací naměřených dat z meteostanice.

Za tímto účelem byl navržen a vytvořen I/O modul s mikropočítačem, který umožní pomocí vestavěného A/D převodníku průběžně digitalizovat výstupní hodnoty z meteorologických čidel, spínat vyhřívání čidel náchylných na mráz a digitalizované hodnoty posílat k dalšímu zpracování do PC, ke kterému je I/O modul připojen do USB portu přes emulovaný sériový port.

Pro PC byla vytvořena aplikace obsluhující sériový port a ukládající přijatá data do webové MySQL databáze. Na PC je i webový server s podporou PHP, pro který byla vytvořena webová aplikace, pomocí níž jsou naměřená meteorologická data uložená v databázi prezentována na internetu.

I/O modul nemusí být připojen přímo k PC s webovým serverem, protože program ukládající přijatá data je schopen přijatá data ukládat přes internet i do vzdálené MySQL databáze.

Klíčová slova: meteorologie, meteostanice, zpracování dat, webová prezentace.

Abstract

The diploma thesis deals in weather station system, that measure and calculate received data, and in presentation the data on the website page.

It was necessary to design and make I/O module with microprocessor which uses built-in A/D converters to continuously digitize the analog output values from meteorological sensors, activate the heating of sensors sensitive to cold and send the digitized data for further processing to the computer . The I/O module is connected to PC via USB port as a emulated serial port.

On PC is running application operating with serial port and saving received data into a web MySQL database. There is also web server with installed PHP plugin. Web application, which is running on PHP script, is processing the measured data and presentation the results saved in database on the web page.

The I/O module may not be directly connected to the computer with the web server. The application , operating with serial port, can save the measured data over the internet into the database too.

Keywords: meteorology, weather station, data processing, web presentation.

Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
Abstract.....	6
Seznam symbolů, zkratk a termínů.....	9
Úvod.....	10
1 Teoretická část.....	11
1.1 Meteostanice.....	11
1.1.1 Meteorologická čidla.....	11
1.1.2 Možnosti řešení.....	12
2 Realizace modulu s mikro počítačem.....	16
2.1 Hardwarové řešení.....	16
2.1.1 Zdrojová část.....	16
2.1.2 Mikro počítač AT89C51CC03.....	17
2.1.3 Převodník FT232RL.....	18
2.1.4 Galvanické oddělení ADuM 3160.....	19
2.3 Návrh DPS.....	19
2.4 Oživení modulu.....	20
2.5 Program pro mikro počítač.....	21
2.5.1 Úkol programu.....	21
2.5.2 Vývojový diagram.....	21
2.5.3 Nastavení SFR.....	22
2.6 Naprogramování mikro počítače.....	26
3 Realizace zpracování dat.....	27
3.1 Obsluha sériové linky.....	27
3.1.1 Vývojový diagram pro obsluhu sériové linky.....	28
3.1.2 Popis metod pro obsluhu sériové linky.....	28
3.2 Obsluha databáze.....	29
3.2.1 Vývojový diagram pro obsluhu databáze.....	30
3.2.2 Popis metod pro obsluhu databáze.....	30
4 Realizace prezentace dat.....	32
4.1 Použité technologie.....	33
4.2 Popis realizace.....	34
4.2.1 Návrh prezenčního rozhraní.....	34
4.2.2 Návrh databáze.....	35
4.2.3 Skript PHP na serveru – aktuální hodnota.....	36
4.2.4 Skript PHP na serveru – historie hodnot.....	38
4.2.5 Skript PHP na serveru – export dat do XLS.....	38

4.2.6 AJAX v prezenčním rozhraní.....	40
4.2.7 API v prezenčním rozhraní.....	42
5 Instalace meteostanice.....	44
6 Závěr.....	45
Použitá literatura.....	46
Seznam obrázků.....	47
Seznam tabulek.....	47
Přílohy.....	48
Schéma I/O modulu.....	49
DPS I/O modulu.....	51
Fotodokumentace.....	52
Přílohy v elektronické podobě – CD.....	55

Seznam symbolů, zkratek a termínů

AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface (rozhraní pro programování aplikací)
A/D	Analog/Digital
CMOS	Complementary Metal-Oxide-Semiconductor (doplňující se kov-oxid-polovodič)
DPS	Deska plošných spojů
GND	Zemnicí potenciál napájecího napětí
HTML	HyperText Markup Language (značkovací jazyk pro strukturování textu)
I/O	Input/Output
JDBC	Java Database Connectivity
JRE	Java Runtime Environment (Java běhové prostředí)
LED	Light Emitting Diode (svítivá dioda)
LGPL	Lesser General Public License (licence svobodného softwaru)
PC	Personal Computer (osobní počítač)
PHP	Hypertext Preprocessor
SFR	Special Function Register (speciální funkční registry)
SMD	Surface Mount Device (součástka pro povrchovou montáž)
TUL	Technická univerzita v Liberci
UART	Universal Asynchronous Receiver / Transmitter (univerzální asynchronní přijímač / vysílač)
USB	Universal Serial Bus (univerzální sériová sběrnice)
Vcc	Kladný potenciál napájecího napětí
Vss	Záporný potenciál napájecího napětí
XML	Extensible Markup Language (rozšiřitelný značkovací jazyk)

Úvod

Hlavním cílem této diplomové práce je modernizace hardwarového a softwarového vybavení meteorologické stanice umístěné v Liberci v Kateřinkách a následná prezentace naměřených dat pomocí webového prostředí.

Prvním cílem práce je navrhnout a vytvořit modul s mikropočítačem. Modul bude v nastavených okamžicích digitalizovat analogové signály z meteorologických čidel. Modul bude přes UART předávat digitalizované hodnoty do PC ve známém tvaru. K PC bude modul připojen přes emulovaný sériový port do USB. Nový modul nahradí zastaralý modul ANDI 08, který je připojen k PC přes COM port, který se na dnešních PC vyskytuje jen zřídka a navíc ANDI 08 odesílá hodnoty do PC v neznámém tvaru a tak je nemožné hodnoty z ANDI 08 dále zpracovávat.

Druhým cílem je vytvořit v jazyce C51 a nahrát do mikropočítače řídicí program, který bude mít za úkol přepínat mezi jednotlivými signály z meteorologických sond a reagovat na komunikaci s PC, ke kterému je připojen.

Třetím cílem je vytvořit aplikaci v jazyce JAVA pro PC, která bude obsluhovat emulovaný sériový port pro příjem naměřených hodnot a následně je pomocí JDBC ukládat do MySQL databáze pro jejich následnou prezentaci. Aplikace bude tvořena v jazyce JAVA, aby bylo docíleno kompatibilitosti s více operačními systémy.

Čtvrtým cílem je vytvoření webové aplikace, která bude přehlednou formou prezentovat naměřená data uložená v databázi. A to nejen hodnoty aktuální, ale i předešlou historii naměřených dat. Aby bylo docíleno co největší dynamičnosti webové aplikace, tak bude tvořena ve skriptovacím jazyce PHP v kombinaci s technologií AJAX. Díky tomu bude webová aplikace prezentovat aktuální hodnoty aniž by se musela na klientské straně načítat znovu celá stránka.

Posledním cílem je úspěšné nainstalování nových hardwarových a softwarových prostředků a tím nahradit stávající prostředky v meteostanici v Kateřinkách.

1 Teoretická část

1.1 Meteostanice

Meteostanice je zařízení, které naměřenými hodnotami uživatele informuje o aktuálním počasí a lze na základě naměřených hodnot i předpovídat vývoj budoucího počasí. Naměřené hodnoty nemusí sloužit jen k informaci o počasí, protože údaje o teplotě, tlaku a vlhkosti vzduchu jsou významné i v mnohých technických odvětvích, protože jsou to veličiny, které dokáží ovlivnit vlastnosti technických zařízení. Rychlost větru, směr větru a srážky jsou zase zajímavé pro letectví. Z toho je vidět, že meteostanice mají dnes širokou oblast působnosti.

Meteostanice se skládá z meteorologických čidel, vyhodnocovací jednotky a ze zobrazovacího prostředí, které zobrazuje naměřené hodnoty. Dále se teoretická část práce zabývá jen meteorologickými sondami, protože vyhodnocovací jednotka a zobrazovací prostředí je praktickým cílem této práce.

1.1.1 Meteorologická čidla

Meteostanice je vybavena sedmi meteorologickými čidly pro měření meteorologických veličin.

Sluneční záření

Pro měření intenzity slunečního záření se používá čidlo RS 81-I. Měření je založeno na principu teplotní difference vzniklé slunečním zářením na černé a bílé ploše. Na připevněných termočláncích vzniká napětí, které je po zesílení přímo závislé na intenzitě slunečního záření. Výstupem čidla je proudový signál 4-20mA. Sluneční záření se uvádí ve wattech na metr čtverečný (W/m^2).

Rychlost a směr větru

Pro měření rychlosti a směru větru se používá čidlo WS 12-H. Pro měření rychlosti větru má rotační lopatkový kříž a pro měření směru větru otočnou lopatkovou směrovku. Snímání otáček kříže a polohy natočení směrovky je prováděno optoelektronicky a k dalšímu zpracování je předáváno v podobě frekvenčně modulovaného signálu do vyhodnocovací desky WO 21-I. Vyhodnocovací deska přijímaný signál vyhodnotí a dál předává dva samostatné proudové signály 4-20mA. Rychlost větru se uvádí v metrech za sekundu (m/s) a směr větru ve stupních počínaje od severu a zvyšujícím se po směru hodinových ručiček ($^{\circ}$).

Vlhkost vzduchu

Vlhkost vzduchu je měřena kapacitním čidlem HS 52-I, které je určeno k měření relativní vlhkosti s vyloučením rušivých povětrnostních vlivů. Relativní vlhkost je poměrem okamžitého množství vodních par ve vzduchu a množstvím par, které by měl vzduch při stejném tlaku a teplotě při plném nasycení. Proto se relativní vlhkost vzduchu uvádí v procentech (%). Čidlo má proudový výstup 4-20mA.

Teplota

K měření teploty je použito čidlo TS 42-I. Měřicím elementem čidla je platinový teploměr a součástí čidla je převodník odpor-proud. Výstup čidla je tedy proudový signál 4-20mA. Pro vyloučení chyb čidla vlivem sluneční radiace je čidlo vybaveno radiačním krytem. Teplota vzduchu se uvádí ve stupních Celsia (°C).

Tlak

K měření tlaku je použito čidlo PS 71-I, které je určeno k měření atmosferického tlaku ve volném prostředí. Výstup čidla je proudový signál 0-20mA. Vzhledem k tomu, že se stoupající výškou atmosferický tlak klesá, se provádí korekce čidla na hladinu moře, aby byly hodnoty porovnatelné pro různé lokality. Atmosferický tlak se uvádí v hektopascalech (hPa).

Srážky

K měření srážek je použit srážkoměr NTM 500, který pracuje na vážkovém principu. Jako vlastní snímací prvek srážkoměru je použito jazýčkové relé ovládané permanentním magnetem. Výstup srážkoměru je tedy pulsní. Srážkoměr je použitelný celoročně, protože je vybaven vyhříváním s bimetalovým spínačem.

1.1.2 Možnosti řešení

Vzhledem k tomu, že zpracovaná data z meteostanice mají být prezentována na webu, nabízí se několik alternativ, jak se dostat od analogových signálů z meteorologických čidel přes web až ke klientskému PC.

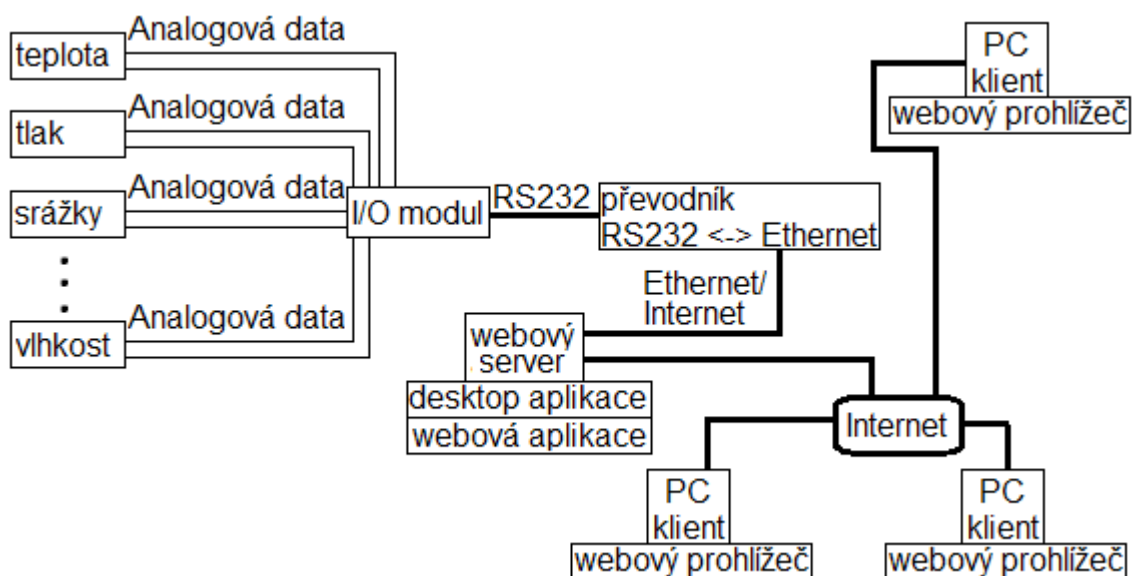
Nejprve je potřeba analogové signály převést na signály digitální a odeslat je do PC. K tomu lze využít univerzální komerční I/O modul s A/D převodníkem, který jde propojit s osobním počítačem, jako je například Quido RS 10/1 z produktové řady Quido od tuzemské společnosti Papouch s. r. o., kde jsou striktně určené vstupy

a výstupy. Vhodnější ale je vyrobit si vlastní I/O modul řízený mikropočítačem, který bude na míru k meteostanici a zároveň si díky mikropočítači zachová určitou univerzálnost pro případné využití v jiné aplikaci pracující s analogovými signály. Vlastní I/O modul s mikropočítačem lze dovybavit i dalším převodníkem, díky kterému se bude moci modul propojit přes emulovanou sériovou linku do USB portu PC.

Dále je nutné digitální signály zpracovat a převést na konkrétní hodnoty měřených veličin. K tomu se přímo nabízí, aby se o to starala aplikace v PC, která bude obsluhovat emulovanou sériovou linku. Aplikace tedy po přijetí digitálních signálů signály převede pomocí převodních charakteristik od jednotlivých čidel na konkrétní naměřenou hodnotu, která bude již čitelná pro každého člověka. Aplikace bude naměřené hodnoty rovnou i ukládat do databáze na webovém serveru, kde bude i webová aplikace starající se o webovou prezentaci naměřených dat z meteostanice.

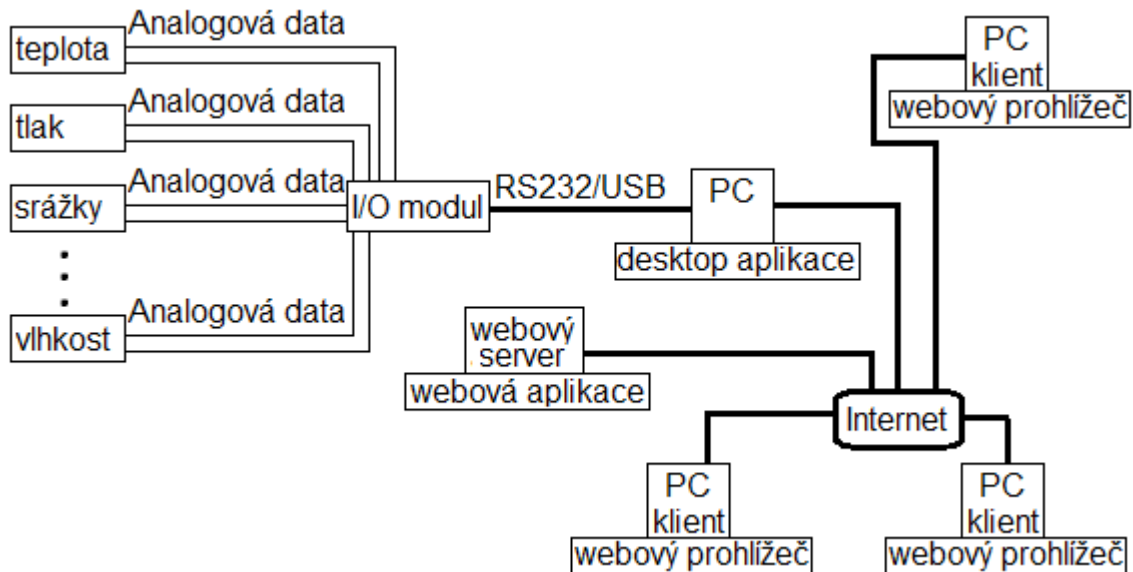
Prostor pro různé způsoby řešení zůstává mezi I/O modulem a webovým serverem.

Prvním příkladem realizace je Distributed Presentation, kdy je veškerá zátěž na serveru. I/O modul bude vybaven převodníkem sériové linky RS232 na Ethernet a komunikace obslužné aplikace s I/O modulem bude probíhat vzdáleně přes internet a to umožní, aby obslužná aplikace byla nainstalována na stejném PC jako webový server s webovou aplikací starající se o prezentaci dat. Zároveň však nebude nutné, aby PC bylo ve stejné lokalitě jako meteostanice.



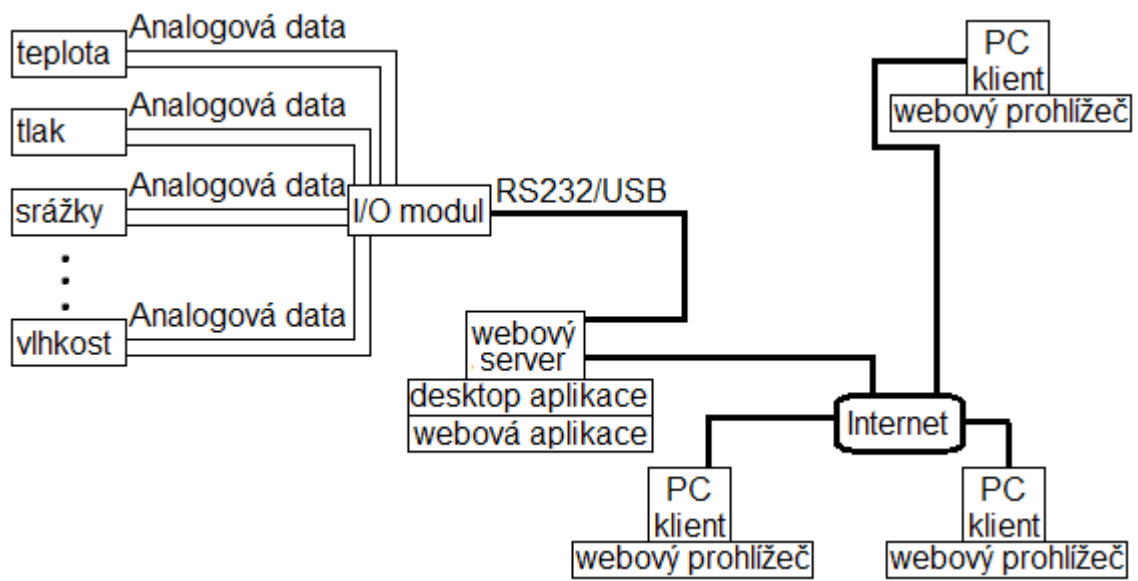
Obrázek 1.1: Meteostanice jako Distributed Presentation se vzdáleným serverem.

V dalším případě se jedná o Remote Data Management a použije se v případě, kdy je u meteostanice PC, ale není dostatečně výkonné pro běh webového serveru nebo není připojeno k internetu dostatečně dimenzovanou linkou. I/O modul lze připojit přímo k tomuto PC, ve kterém bude obslužná aplikace. Obslužná aplikace potom data bude ukládat do databáze na webovém serveru vzdáleně přes internet. PC a internetové připojení tak bude vytíženo minimálně, protože bude data odesílat jen jednou do databáze na webovém serveru a ne několikanásobně všem připojeným klientům.



Obrázek 1.2: Meteostanice jako Remote Data Management

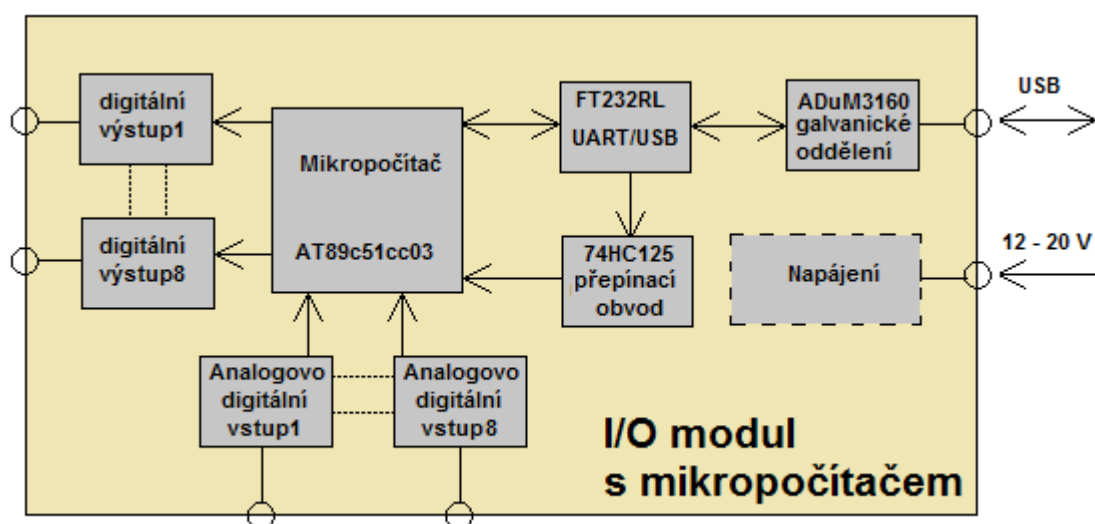
V případě, že je u meteostanice dostatečně výkonné PC, aby na něm šel provozovat webový server a je připojen k internetu dostatečně dimenzovanou linkou lze I/O modul připojit přímo k tomuto PC na kterém bude nainstalována obslužná aplikace i webový server s webovou aplikací a klienti se budou přes internet připojovat přímo k PC u meteostanice. Toto řešení je nejjednodušší a nejlevnější, protože není potřeba převodník sériové linky RS232 na Ethernet a postačí jedno PC, které bude zároveň i webovým serverem. V tomto případě se jedná opět o Distributed Presentation.



Obrázek 1.3: *Meteostanice jako Distributed Presentation s lokálním serverem.*

2 Realizace modulu s mikropočítačem

Tato část práce se zabývá konkrétním řešením I/O modulu s mikropočítačem. Základem modulu je mikropočítač, a protože modul musí zpracovávat analogové signály, tak byl vybrán mikropočítač od společnosti ATMEL AT89c51cc03, který disponuje osmikanálovým A/D převodníkem s rozlišením 8 bitů.



Obrázek 2.1: Blokové schéma I/O modulu s mikropočítačem.

Na obrázku 2.1 je blokové zapojení celého modulu. Z blokového zapojení je patrné, že modul disponuje osmicími vstupů, které lze použít pro analogové i pro digitální signály. Dále je modul vybaven osmicími digitálními výstupy, které lze také použít i jako digitální vstupy. Zda se jedná o vstupy nebo výstupy se řeší řídicím programem mikropočítače. Celý modul je napájen externím zdrojem a je galvanicky oddělen od PC, ke kterému je připojen přes emulovaný sériový port do USB.

2.1 Hardwarové řešení

Schéma zapojení bylo realizováno v návrhovém softwaru EAGLE 5.11. Návrhový list se zapojením je rozdělen do jednotlivých funkčních částí. Každá část obsahuje soubor součástek, které k sobě patří.

2.1.1 Zdrojová část

Modul je osazen svorkovnicí, ke které je připojen externí zdroj stejnosměrného napětí v rozsahu 12-20 V. Za svorkovnicí je ochranná dioda a elektrolytický filtrační

kondenzátor. Vyfiltrované napětí je usměřováno lineárním napěťovým stabilizátorem 7805, který má na výstup napětí o hodnotě 5 V, které je použito k napájení celého modulu.

2.1.2 Mikropočítač AT89C51CC03

Mikropočítač AT89C51CC03 byl vybrán proto, že je založen na jádře známé a léty prověřené architektury 80C51. I přes to, že je založen na staré architektuře 80C51 jedná se o moderní mikropočítač, který má přímo na čipu programovou paměť o velikosti 64 kB + 2 kB pro Bootloader, který slouží pro přímé nahrávání programu přes plně duplexní UART. K nahrávání programu lze ale použít i grafický bootloader FLIP od společnosti ATMEL. Dále je mikropočítač vybaven A/D převodníkem s rozlišením 8 bitů, který má 8 multiplexovaných vstupů a již zmiňovaný UART. Mezi další potřebné vlastnosti patří tři 16 bitové čítače/časovače. To byl výčet nadstandardních vlastností mikropočítače, které jsou potřeba pro realizaci modulu. Mezi další zajímavé vlastnosti kterými disponuje, ale nebudou využity patří například možnost komunikace po CAN sběrnici, PWM (pulsně šířková modulace), rozhraní SPI pro komunikaci s dalšími mikropočítači atd. Podrobný popis mikropočítače lze nalézt v technickém listě mikropočítače.[2]

Schéma zapojení mikropočítače vychází z doporučených zapojení uvedených v dokumentaci k mikropočítači.[2] Důležité je zapojit všechny piny označené Vcc ke kladnému potenciálu napájení a všechny piny Vss k zemnímu potenciálu napájení, který je značen jako GND a nespoléhat na vnitřní propojení těchto pinů.

Pomocí pinů 1 (VAGND) a 2 (VAREF) se nastavuje referenční napětí pro A/D převodník, které má být v rozsahu 2,5-3 V. Pin 1 je tady připojen ke GND a pin 2 je zapojen k odporovému děliči napětí, který vytváří kladné napětí 3 V.

Piny 3-10 (Port 1) jsou vstupy pro A/D převodník, tak jsou připojeny ke šroubovacím svorkovnicím, ke kterým se připojují výstupy meteorologických čidel.

Pin 11 (EA) nastavuje zda je pro program použita vnitřní (log. „1“) nebo vnější (log. „0“) programová paměť. Protože vnitřní programová paměť je dostačující je pin 11 připojen k Vcc (log. „1“).

Piny 12 (RXD) a 13 (TXD) slouží k sériové komunikaci UART a proto jsou připojeny k převodníku FR232RL (viz kapitola 2.1.3).

Piny 30-37 (Port 0) jsou připojeny přes rezistorovou síť 8 x 10kΩ k Vcc, aby bylo zajištěno, že je na výstupech portu P0 při nečinnosti log. „1“. Port P0 tedy slouží jako výstupy pro případné spínání doplňkových obvodů. Spínání je prováděno log. „0“.

Pin 38 (PSEN) slouží k odemčení programové paměti pro přepsání. Pin 38 je zatěžovací odpor 10 kΩ připojen k Vcc (log. „1“), takže je programová paměť uzamčena a nelze jí přepsat. Při programování mikropočítače, je však pin 38 připojen pomocí prepínacího obvodu (74HC125) ke GND (log. „0“) a programová paměť se odemkne a lze do mikropočítače nahrát nový program.

Piny 40 (XTAL1) a 41 (XTAL2) jsou přes keramické kondenzátory připojeny ke GND a mezi tyto piny je připojen krystal o kmitočtu 20 MHz, který udává taktovací kmitočet mikropočítači.

Pin 44 (RESET) slouží k resetování mikropočítače. Připnutím pinu 44 k Vcc (log. „1“) po dobu dvou strojových cyklů se mikropočítač resetuje. Pin 44 je vnitřně přidržen zatěžovacím odporem ke GND (log. „0“), takže již není potřeba ho zatěžovat externě. Mikropočítač se po připojení k napájení resetuje díky kondenzátoru připojeném mezi pinem 44 a Vcc. Kondenzátor pin 44 potřebně dlouhou dobu propojuje s Vcc (log. „1“). Pin 44 je zapotřebí propojit s Vcc (log. „1“) i během programování, a proto je pin 44 připojen i na prepínací obvod (74HC125), který během programování mikropočítače připojí pin 44 k Vcc (log. „1“)

2.1.3 Převodník FT232RL

Vzhledem k ubývajícím sériovým portům u PC je převodník FT232RL čím dál častěji používaným převodníkem. Jedná se totiž o obousměrný převodník z USB na UART. V modulu je převodník zapojen jako obousměrný převodník USB na RS232. V programovacím režimu se pomocí převodníku z PC do mikropočítače nahrává řídicí program a v pracovním režimu přes převodník probíhá obousměrná komunikace mezi mikropočítačem a PC.

Z převodníku FT232RL jsou brány i ovládací signály pro obvod 74HC125, který přepíná mikropočítač mezi programovacím a provozním režimem.

Schéma zapojení převodníku vychází z doporučeného zapojení uvedeném v technickém listu převodníku.[3] Doporučené zapojení je doplněno o společné

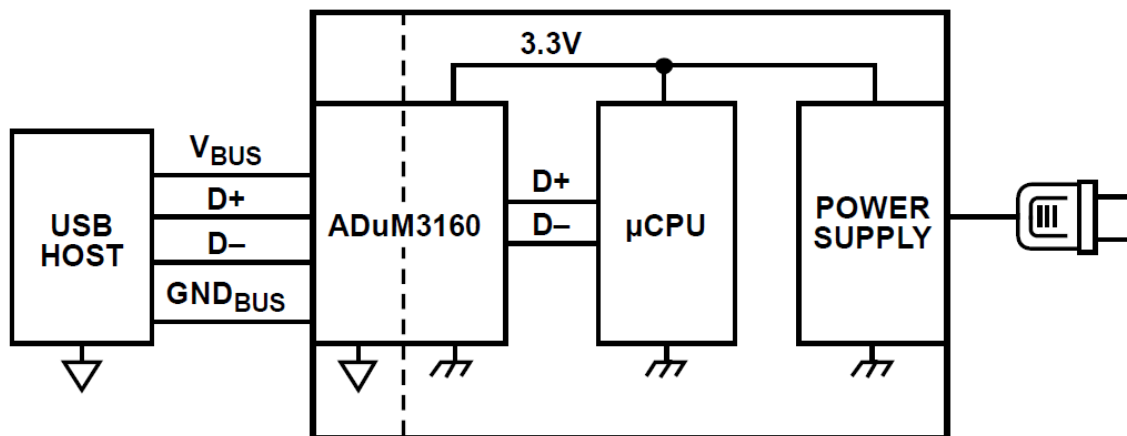
propojení signálů DCD, DSR a DTR a jejich použití k ovládní přepínacího obvodu 74HC125. Dále jsou vzájemně propojeny signály RTS a CTS a použity jako další ovládací signál přepínacího obvodu 74HC125.

2.1.4 Galvanické oddělení ADuM 3160

Aby bylo PC, ke kterému je modul připojen chráněno před nestandardními výboji elektrické energie, tak je modul od PC galvanicky oddělen. Ke galvanickému oddělení je použit integrovaný obvod ADuM 3160.

ADuM 3160 je oddělovač pro USB port využívající k oddělení kombinaci technologie CMOS a transformátorů se vzduchovým jádrem. ADuM 3160 je schopen odolávat přepětí až do 2,5 kV.

Schéma zapojení galvanického oddělení vchází z popisu uvedeného v technickém listu integrovaného obvodu ADuM 3160[4], které je dáno celkem jednoznačně a jedinou možností volby je nastavení přenosové rychlosti oddělovače. A to jestli se jedná o pomalou verzi USB 1.1 (Low-Speed) nebo o rychlou verzi USB 2.0 (Full-Speed). Vzhledem k tomu, že USB je zpětně kompatibilní byla vybrána vyšší rychlost určená pro USB 2.0.



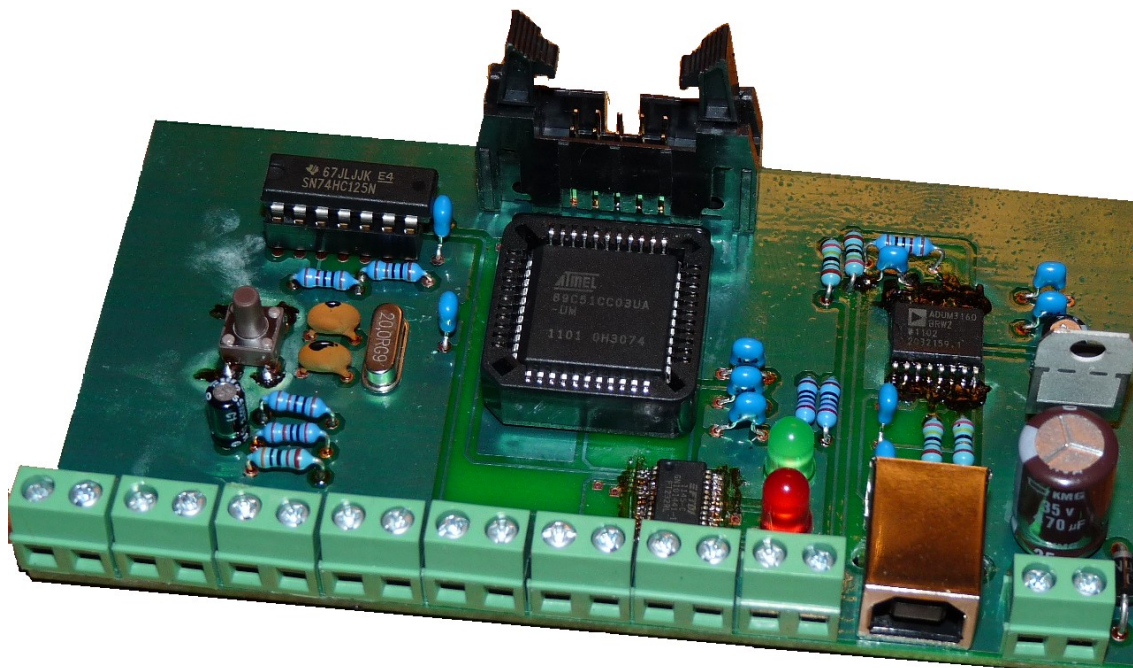
Obrázek 2.2: Blokové zapojení galvanického oddělení.[4]

2.3 Návrh DPS

Před návrhem DPS bylo nutné rozvrhnout umístění připojovacích svorkovnic pro připojení jednotlivých čidel a napájení a umístění USB konektoru pro propojovací kabel z PC, aby se vše připojovalo na okraji DPS. Celý modul je složen z jedné oboustranné DPS, která je osazena všemi součástkami.

Návrh DPS byl prováděn stejně jako návrh schématu v návrhovém softwaru EAGLE 5.11. Při návrhu musela být brána v potaz dostupnost pouzder jednotlivých součástek, proto jsou v návrhu SMD součástky pro povrchovou montáž i běžné součástky určené pro montáž do vrtaných otvorů. Dále musely být brány v potaz technické možnosti návrhového softwaru a laboratoře pro výrobu DPS.

Konečným výsledkem je tedy dvouvrstvá deska plošných spojů o celkových rozměrech 118 x 64 mm. Motiv DPS je vyobrazen v příloze.



Obrázek 2.3: Vyrobený I/O modul s mikropočítačem.

2.4 Oživení modulu

Modul by po částech testován. Nejprve byl modul připojen k napájení a následně byla napájecí část modulu proměřena a shledána funkční a bezpečnou. To umožnilo bezpečné připojení modulu k PC. Po připojení modulu k PC se v PC automaticky nainstaloval emulovaný sériový port, čímž se ověřila funkčnost převodníku FT232RL a galvanického oddělení ADuM 3160. K poslednímu testu funkčnosti byl využit program FLIP, pomocí kterého byl mikropočítač přepnut do programovacího režimu a byl do mikropočítače úspěšně nahrán prázdný program. Po tomto testu byl modul prohlášen po hardwarové stránce za plně funkční a tudíž nebyla v návrhu schématu a DPS odhalena žádná chyba mající vliv na funkci modulu.

2.5 Program pro mikropočítač

Program pro mikropočítač je napsán v programovacím jazyce C51, což je odnož programovacího jazyku C určená pro programování mikropočítačů založených již na zmíněném jádře 80C51. K napsání a kompilaci programu do strojového kódu ve formátu HEX přijatelného pro mikropočítač bylo využito vývojového prostředí Keil uVision3 s integrovaným kompilátorem.

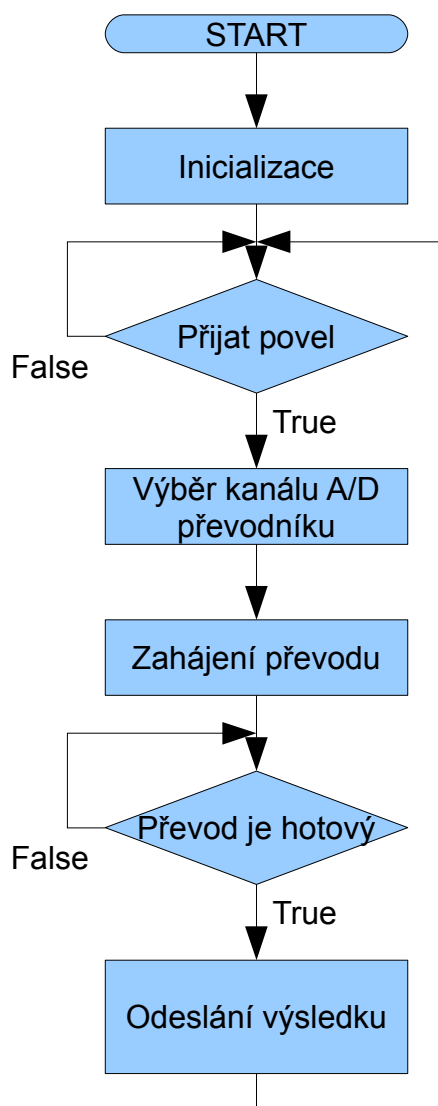
Jazyk C51 byl zvolen proto, že se jedná o vyšší programovací jazyk, který na rozdíl od jazyku symbolických adres, který je nižším programovacím jazykem, umožňuje přehlednější a rychlejší programování

2.5.1 Úkol programu

Program pro mikropočítač určuje jak se má mikropočítač chovat. Od mikropočítače se očekává, že po přijetí povelu po sériové lince na tento povel patřičně odpoví. Tyto povely jsou ve formě jednotlivých číselných znaků 0 až 7. Po přijetí znaku 0 mikropočítač reaguje tak, že přečte hodnotu na vstupu A/D převodníku na vstupním kanále 0 a hodnotu odešle přes sériovou linku do PC. Při přijetí povelu 1 provede to samé pro kanál 1 atd. Proto je nutné aby program pro mikropočítač dokázal obsluhovat komunikaci na sériové lince a zároveň dokázal obsluhovat vnitřní A/D převodník mikropočítače.

2.5.2 Vývojový diagram

Pro návrh a popis programu je nejlepší použít vývojový diagram, který přehlednou formou v několika grafických blocích znázorní funkci programu. Při programování se potom řeší postupně jednotlivé bloky.



Obrázek 2.4: Vývojový diagram programu pro mikročítač.

2.5.3 Nastavení SFR

SFR jsou speciální funkční registry, pomocí nichž se řídí a nastavují vnitřní obvody mikročítače jako je sériová linka, časovače, A/D převodník a podobně.

ADCF – A/D Configuration

Port 1 je po restartu mikročítače nastaven jako vstupně výstupní pro digitální signály. Pomocí osmibitového SFR ADCF lze definovat, které piny Portu 1 budou použity jako vstupy pro analogové signály do A/D převodníku. Konkrétní piny Portu 1 se nastavují jako vstupy pro analogové signály do A/D převodníku nastavením log. „1“ na konkrétním bitu ADCF.

Například $ADCF = 11111111b$; nastaví všechny piny Portu 1 jako vstupy pro analogové signály do A/D převodníku

Tabulka 2.1: Význam bitů v SFR ADCF.

	7	6	5	4	3	2	1	0
ADCF	Port 1.7	Port 1.6	Port 1.5	Port 1.4	Port 1.3	Port 1.2	Port 1.1	Port 1.0

ADCON – A/D Control

SFR ADCON je registr módů a řízení A/D převodníku. Práci s ním nejlépe popíše následující tabulka.

Tabulka 2.2: Význam bitů v SFR ADCON.

číslo bitu	bit	význam bitu
7	-	Nevyužit.
6	PSIDLE	Nastavení módu.
5	ADEN	Povolení A/D převodníku.
4	ADEOC	Signalizace ukončení převodu. Vyvolá přerušení.
3	ADSST	Spuštění A/D převodu.
2	SCH2	Výběr kanálu.
1	SCH1	Výběr kanálu.
0	SCH0	Výběr kanálu.

Při Používání A/D převodníku je potřeba nejprve A/D převodník povolit:

$ADCON = 00100000b$; Povolení A/D převodníku.

Když je A/D převodník povolen, tak se vybere konkrétní kanál a spustí se A/D převod:

$ADCON = 00000110b$; Vybrání kanálu 6.

$ADCON = 00001000b$; Spuštění převodu.

K výběru kanálu A/D převodníku se používají tři nejnižší bity ze SFR ADCON, kterými jsou SCH0, SCH1 a SCH2. Výběr konkrétních bitů popisuje následující tabulka.

Tabulka 2.3: Výběr kanálu A/D převodníku.

SCH2	SCH1	SCH0	Vybraný kanál
0	0	0	kanál 0
0	0	1	kanál 1
0	1	0	kanál 2
0	1	1	kanál 3
1	0	0	kanál 4
1	0	1	kanál 5
1	1	0	kanál 6
1	1	1	kanál 7

O úplnosti A/D převodu nás informuje přerušení číslo 9 vyvolané A/D převodníkem. V přerušení je potřeba softwarově vynulovat bit ADEOC který je v SFR ADCON na pozici 4 a značí ukončení A/D převodu:

ADCON |= 00010000;

Vynulování bitu ADEOC.

ADDH – A/D Data High Byte

Po ukončení A/D převodu je při použití osmibitového módu převedená hodnota uložena v osmibitovém SFR ADDH.

Tabulka 2.4: Význam bitů v SFR ADDH.

	7	6	5	4	3	2	1	0
ADDH	ADAT7	ADAT6	ADAT5	ADAT4	ADAT3	ADAT2	ADAT1	ADAT0

RCAP2H(L) – Timer/Counter 2 Reload/Capture High(Low) byte

SFR RCAP2H a RCAP2L se používají k nastavení konstanty pro šestnáctibitový časovač 2, kterou musí časovač načítat než dojde k přetečení. Časovač 2 se používá k nastavení přenosové rychlosti sériové linky. Přenosová rychlost sériové linky se totiž odvíjí od četnosti přetečení časovače 2. Přenosová rychlost se nastavuje dle vzorce 2.1.

$$\text{Přenosová rychlost} = \left(\frac{2^{SMOD}}{32} \right) \cdot \left(\frac{f_{osc}}{12 \cdot (65536 - [RCAP2H, RCAP2L])} \right) \quad (\text{Vzorec 2.1})$$

Tabulka 2.5: Význam bitů v SFR RCAP2H a RCAP2L.

SFR	číslo bitu	význam bitu
RCAP2H	7 – 0	Vyšší bity časovače 2.
RCAP2L	7 – 0	Nižší bity časovače 2.

T2CON – Timer 2 Control

K nastavení a ovládání časovače 2 se používá SFR T2CON. Aby se mohl časovač 2 využít k nastavení přenosové rychlosti sériové linky je potřeba nastavit bity RCLK a TCLK a následně časovač 2 spustit nastavením bitu TR2.

T2CON |= 00110000b; Nastavení bitů RCLK a TCLK

TR2 = 1b; Spuštění časovače 2.

K jednotlivým bitům se dá přistupovat i přímo, takže ke spuštění a zastavování časovače 2 je vhodné pro přehlednost nastavovat bit TR2 přímo.

Tabulka 2.6: Význam bitů v SFR T2CON.

číslo bitu	bit	význam bitu
7	TF2	Signalizace přetečení časovače 2. Vyvolá přerušení.
6	EXF2	Signalizace externího přerušení pro časovač 2.
5	RCLK	Nastavení zda bude časovač 2 použit pro příjem po sériové lince.
4	TCLK	Nastavení zda bude časovač 2 použit pro vysílání po sériové lince.
3	EXEN2	Povolení externího přerušení pro časovač 2.
2	TR2	Spuštění časovače 2.
1	C/T2	Volba činnosti mezi čítačem a časovačem
0	CP/RL2	Nastavení zda časovač reaguje na přetečení nebo externí vstup.

SCON – Serial Control

Sériová linka u mikropočítače se nastavuje registrem módů a řízení, což je SFR SCON. Nejvýhodnější mód sériové linky je mód 1, který je obousměrný asynchronní s nastavitelnou přenosovou rychlostí. Mód 1 se nastavuje bity SM0 a SM1. Aby se dala sériová linka využít obousměrně, tak je nutné ještě povolit příjem přes sériovou linku. SFR SCON se tedy nastaví takto:

SCON |= 01010000b; Nastavení modu 1 a povolení příjmu.

Tabulka 2.7: Význam bitů v SFR SCON.

číslo bitu	bit	význam bitu
7	SM0	Konfigurační bit určující společně s bitem SM1 mód sériové linky.
6	SM1	Konfigurační bit určující společně s bitem SM0 mód sériové linky.
5	SM2	Bit povolující vytvoření víceprocesorové komunikace.
4	REN	Bit povolující příjem přes sériovou linku.
3	TB8	Devátý datový bit pro vysílání
2	RB8	Devátý datový bit pro příjem.
1	TI	Příznak prázdného posuvného registru, používaného při vysílání.
0	RI	Příznak přijatých platných dat.

2.6 Naprogramování mikropočítače

Mikropočítač se naprogramuje vytvořeným programem zkompilevaným do formátu HEX. Program se do mikropočítače nahrává sériově, přes sériovou linku pomocí bootladeru umístěného ve vnitřní programové flash paměti mikropočítače. Při programování přes bootlader je nutné zajistit hardwarovou nebo softwarovou aktivaci/deaktivaci bootladeru, která je popsána v technickém listu mikropočítače.[2] Pokud se bootlader po naprogramování nedeaktivuje, tak po restartu mikropočítače nebude probíhat program nahraný od adresy 0000h, ale od adresy FC00h, kde je nahrán bootlader.

K programování mikropočítače lze použít i grafický bootlader FLIP (Flexible In-System Programmer) od společnosti ATMEL. Pomocí FLIPu lze programovat většinu procesorů společnosti ATMEL vybavených rozhraním UART, CAN nebo USB. I při použití FLIPu je nutné deaktivovat vnitřní bootlader mikropočítače a to tak, že se ve FLIPu odškrtně BLJB (BootLoader Jump Bit), který tím deaktivuje vnitřní bootlader mikropočítače a po restartu mikropočítače probíhá nahraný program od adresy 0000h.

3 Realizace zpracování dat

Hodnoty odeslané z I/O modulu do PC je potřeba dále zpracovat a uložit do databáze. Přijaté hodnoty jsou jen zdigitalizované signály z meteorologických čidel, a proto se musí přijaté hodnoty převést pomocí převodních charakteristik jednotlivých čidel na čitelné hodnoty odpovídajících rozsahů.

Převedené hodnoty je potom možné uložit do databáze kde budou přístupné pro webovou prezentaci.

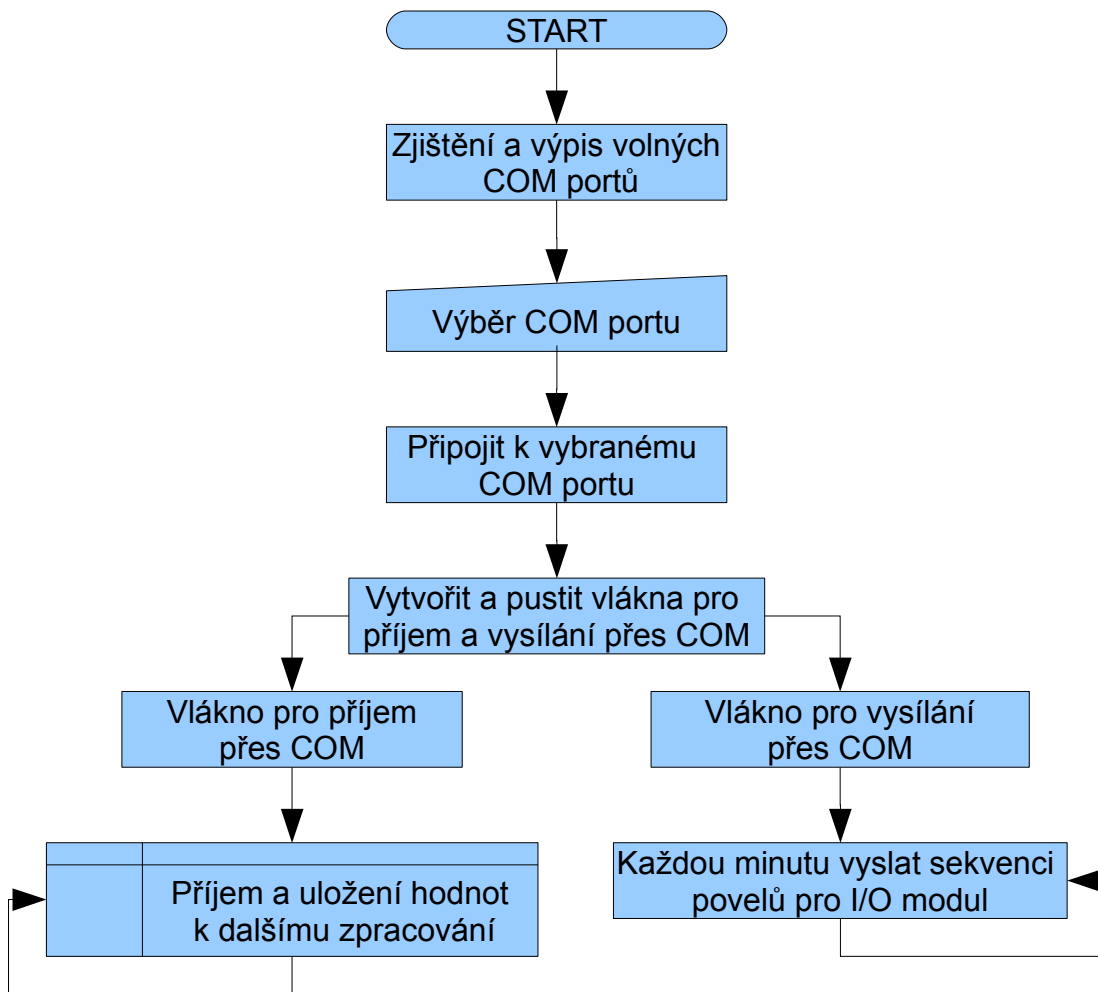
Za tímto účelem byla napsána aplikace pro PC, ke kterému je připojený I/O modul. Aplikace je napsána v objektově orientovaném programovacím jazyce Java ve vývojovém prostředí NetBeans IDE 7.0.1. Java je multiplatformním jazykem a tak po překompilování zdrojových kódů je možné program pustit na libovolné platformě.

3.1 Obsluha sériové linky

Pro programování aplikací v Javě pracujících se sériovou linku je potřeba použít rozšiřující aplikační rozhraní. Oracle však ukončil vývoj Java Communication API pro operační systém Windows, které umožňovalo obsluhu sériové linky. Proto je doporučeno používat komunitní implementaci Java Communication API, kterým je projekt RXTX. RXTX je šířen pod licencí LGPL, takže je možné RXTX linkovat programem, který může být jak svobodným tak i proprietárním.[6]

Při používání RXTX se používají stejné příkazy a stejně deklarované metody jako při používání již nepodporovaného Java Communication API, a tak lze při programování použít původní dokumentaci od Oracle.[7]

3.1.1 Vývojový diagram pro obsluhu sériové linky



Obrázek 3.1: Vývojový diagram pro obsluhu sériové linky.

3.1.2 Popis metod pro obsluhu sériové linky

Aby se aplikace mohla připojit k sériovému portu, musí se nejdříve zjistit dostupnost sériových portů v PC. K tomu slouží metoda `getAvailableSerialPorts()`, která jako výsledek vrátí asociativní pole ve kterém je seznam všech sériových portů a zda jsou obsazené nebo volné. Volné sériové porty, ke kterým se lze připojit se vypíší do roletového ComboBoxu, a dá se tím na výběr uživateli, ke kterému sériovému portu se chce připojit, aby tím navázal komunikaci s I/O modulem.

Po uživatelském výběru volného sériového portu se metoda `connect(String portName)` připojí k vybranému sériovému portu. Vzhledem k tomu, že komunikace přes sériovou linku je asynchronní událost, je nutné, aby příjem

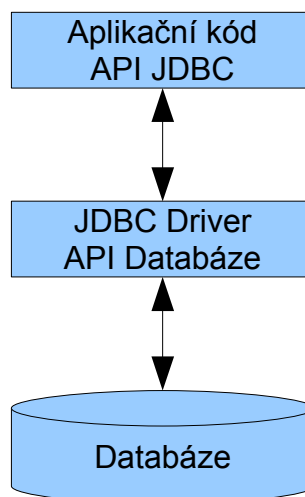
dat a vysílání dat probíhalo každé v samostatném vlákne. Metoda `connect(String portName)` tedy po připojení k sériovému portu vytvoří a spustí dvě nová vlákna ve třídách `SerialReader` a `SerialWriter`.

V metodě `run()` nového vlákna pro vysílání přes sériovou linku ve třídě `SerialWriter` probíhá v minutových intervalech vysílání žádostí přes sériovou linku do I/O modulu o aktuální hodnoty na výstupech meteorologických čidel. Ve vlákne pro příjem přes sériovou linku ve třídě `SerialReader` metoda `run()` přes sériovou linku tyto hodnoty přijímá a předává k dalšímu zpracování metodě `insertIntoTable(int veličina, int hodnota)` ve třídě `ServerComm`.

3.2 Obsluha databáze

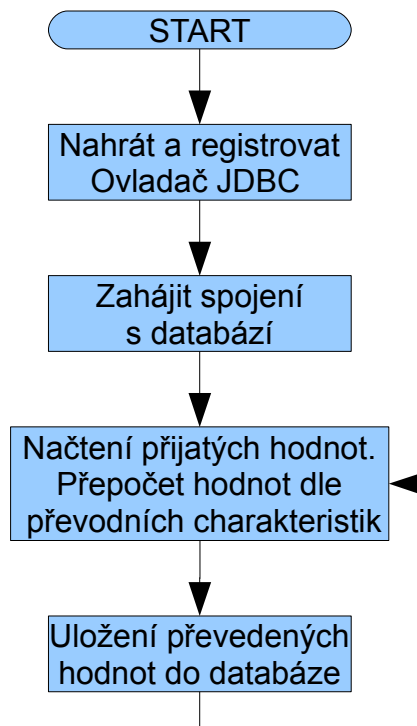
Data z meteorologických čidel jsou prezentována ve webové aplikaci, a tak je vhodné přijatá data zpracovat a uložit do databáze, kde budou pro webovou aplikaci dostupné. Konkrétně se jedná o databázi typu MySQL.

Java používá pro práci s databázemi aplikační rozhraní Java Database Connectivity. JDBC je základním rozhraním pro unifikovaný přístup k databázím. JDBC překládá aplikační kód do nativních volání dané databáze. Programátor je tak odstíněn od specifikací dané databáze a stačí mu znalosti jednotného rozhraní JDBC.[8]



Obrázek 3.2: Znárodnění architektury JDBC

3.2.1 Vývojový diagram pro obsluhu databáze



Obrázek 3.3: Vývojový diagram pro obsluhu databáze.

3.2.2 Popis metod pro obsluhu databáze

Veškeré operace prováděné nad databázemi v Javě musí být ošetřeny použitím výjimek. K ošetření výjimek v Javě se používá konstrukce `try-catch-finally`.

```
Try{  
    //chráněný blok kódu  
}catch(typ výjimky){  
    //ošetření výjimky  
}finally{  
    //blok kódu provedený bez ohledu na vznik výjimky  
    //u práce s databází se zde uvolňují použité prostředky  
}
```

Před provedením prvních operací nad databází je potřeba nahrát a registrovat ovladač JDBC. To se provede zavoláním statické metody `Class.forName("com.mysql.jdbc.Driver");`. Tato metoda musí být ošetřena výjimkou `ClassNotFoundException`, což je výjimka kontrolující zda systém danou třídu dokázal najít. Registrace JDBC ovladače je vázána na třídu `DriverManager`. `DriverManager` je třída obsahující metodami pro práci s JDBC.

Po nahrání a registraci JDBC ovladače je možné navázat spojení s databází. Metodu `getConnection` pro navázání spojení s databází nám poskytne již zmíněná třída `DriverManager`. Spojení s konkrétní databází se identifikuje pomocí databázového URL. Databázové URL vypadá dle specifikace takto:

```
jdbc:mysql://host:port/databáze.
```

Pokud dojde k autorizaci k získání spojení, tak metoda `getConnection()` vrátí požadované spojení, přes které je možné provádět všechny operace nad připojenou databází. Provádění operací nad připojenou databází nám umožní třída `Statement`, kterou nám poskytne objekt `Connection`.

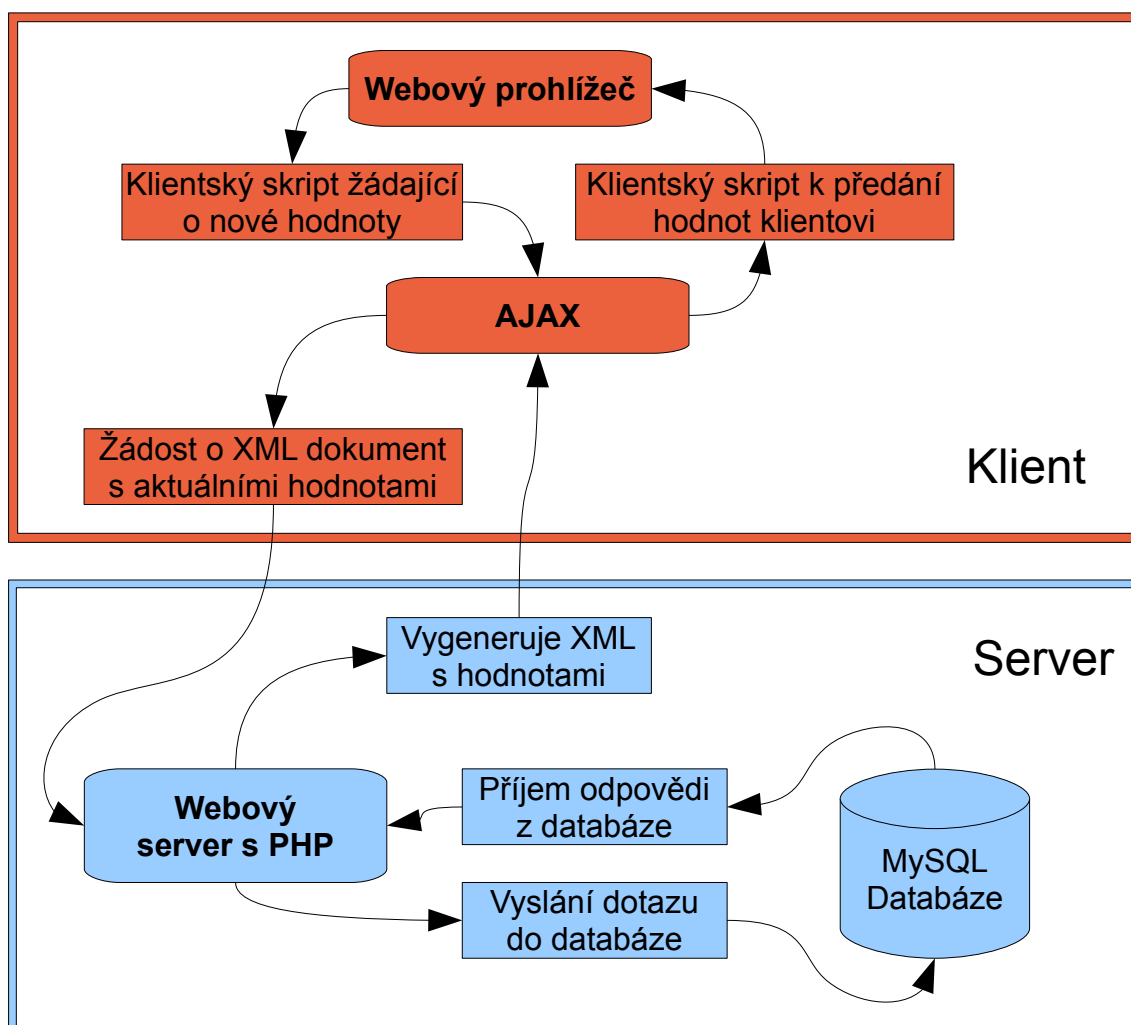
Instance třídy `Statement` nám potom slouží k posílání SQL příkazů připojené databázi. Třída `Statement` nabízí velké množství metod, ale vzhledem k tomu, že je potřeba do databáze jen ukládat data, tak stačí použít metodu `executeUpdate(sql)`, kde `sql` je proměnná typu `String` a obsahuje SQL dotaz pro uložení hodnot do databáze.

Tvar SQL dotazu vytvoří ve třídě `ServerComm` metoda `insertIntoTable(int veličina, int hodnota)` potom co se na základě převodních charakteristik meteorologických čidel přepočítají přijímané hodnoty do čitelného tvaru dle rozsahů měřicích čidel.

4 Realizace prezentace dat

Naměřené hodnoty z meteorologických čidel jsou prezentovány na internetu. K internetové prezentaci je vytvořena webová aplikace využívající moderní dynamické technologie jako je AJAX, JavaScript a PHP a statické HTML.

Technologie AJAX je využita ke změně obsahu webové stránky bez nutnosti jejího znovunačítání, což uživateli zpříjemní sledování naměřených dat bez problikávání celé webové stránky. JavaScript je využit k tvorbě měnících se grafických částí webové stránky a technologie PHP je využito ke komunikaci s MySQL databází, ve které jsou uložena naměřená data. Statické HTML vytváří kostru celé webové stránky, do které jsou vloženy jednotlivé dynamické technologie.



Obrázek 4.1: Schématické znázornění funkce webové aplikace.

4.1 Použité technologie

Statické HTML

HTML je značkovací jazyk pro strukturování textu. Pomocí HTML se nemusí ovšem strukturovat jen text, ale umožňuje strukturovat i grafické prvky, a tak je HTML vhodné pro tvorbu statických webových stránek, nebo pro rozvržení vzhledu webových stránek s dynamickým obsahem. Ke strukturování se používají párové tagy, mezi které se uzavírají textové, grafické nebo dynamické elementy, které mají být daným párem tagů ovlivněny.

PHP

PHP je interpretovaný skriptovací jazyk pro programování dynamických částí webových stránek. Při použití PHP se skripty provádějí na straně serveru a klient obdrží jen výsledek provedené operace. PHP tedy zatěžuje klientské PC jen minimálně, protože u klienta probíhá jen prezentační část a aplikační a datovou část má na starosti server. Klient tedy ve svém webovém prohlížeči nemá vůbec k PHP skriptům přístup, protože obdrží jen výsledek určený k prezentaci.

JavaScript

JavaScript je interpretovaný objektově orientovaný skriptovací jazyk. Při použití JavaScriptu se skripty provádějí na straně klienta, takže klientské PC je zatíženo jak aplikační tak prezentační částí vykonávání skriptu. JavaScript se používá ve spojení s webovými aplikacemi k tvorbě dynamicky se chovajících grafických prvků, jako jsou tlačítka nebo ke tvorbě animací. Vzhledem k tomu, že skript probíhá na straně klienta, tak z bezpečnostních důvodů není možné pomocí JavaScriptu pracovat se soubory a databázemi a proto je v těchto případech JavaScript doplňován skriptovacím jazykem PHP.

AJAX

AJAX je technologie kombinující asynchronní JavaScript a XML a umožňuje tvorbu dynamických webových aplikací, které mění svůj obsah bez toho, aby docházelo k jejímu znovunačtení. Technologie AJAX používá JavaScript pro zobrazování dynamických změn a tyto změny jsou potom prezentovány technologií HTML. Aby technologie AJAX běžící na straně klienta mohla komunikovat se serverem a se skripty běžícími na straně serveru používá ke komunikaci mezi klientem a serverem

formát údajů XML. Technologie AJAX jako taková není žádnou technologií, ale jen vhodnou kombinací technologií jiných.[9]

MySQL

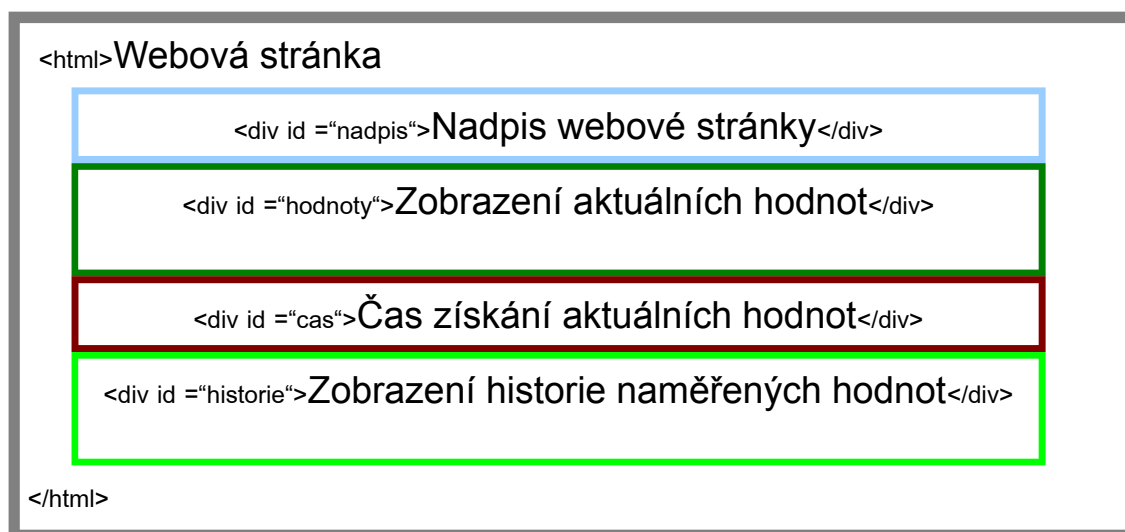
MySQL je multiplatformní databázový systém. Komunikace s ním probíhá pomocí jazyka SQL. MySQL je optimalizován na rychlost, což přináší některá zjednodušení systému. I přes tato zjednodušení je MySQL dostatečně použitelné pro většinu aplikací. Hlavní výhodou MySQL je jeho bezplatná licence, což z něj dělá jeden z nejrozšířenějších databázových systémů.

4.2 Popis realizace

4.2.1 Návrh prezenčního rozhraní

Při tvorbě webové aplikace byla tedy nejprve navržena kostra webové stránky použitím statického HTML, pomocí něhož se rozvrhl vzhled prezenční webové stránky. K rozvržení vzhledu prezenční webové stránky byla použita technika DIV layout. Jedná se o rozdělení webové stránky na blokové elementy. Rozdělení na blokové elementy typu DIV bylo provedeno s ohledem na použití technologie AJAX, protože technologie AJAX je schopna dynamicky měnit obsah konkrétních blokových elementů.

Webová stránka je rozdělena do čtyřech vodorovných blokových elementů. Webová stránka se dá tedy dělit do čtyřech pod sebe rozdělených částí.



Obrázek 4.2: Rozložení vzhledu webové aplikace.

První DIV identifikovaný jako *nadpis* obsahuje jen textový element ohraničený

dalším párovým HTML tagem <h1> značící nadpis první úrovně. Tento textový element má za úkol uživatele informovat o jakou mateostanici se jedná.

Druhý DIV identifikovaný jako *hodnoty* obsahuje šest gauge metrů pro zobrazování aktuálních hodnot jednotlivých veličin a jeden růžicový kompas pro zobrazení aktuálního směru větru. Hodnoty aktuálních veličin jsou získávány z MySQL databáze PHP skriptem a tnechnologií AJAX předávány klientovi.

Třetí DIV identifikovaný jako *čas* obsahuje čas naměření aktuálních hodnot. Čas je získáván z databáze společně s aktuálními hodnotami.

Čtvrtý DIV identifikovaný jako *historie* obsahuje několik záložek. Pod každou záložkou je graf zobrazující historii naměřených hodnot pro jednotlivé veličiny. Další záložka obsahuje stránku s PHP skriptem, který umožňuje exportovat naměřené hodnoty do souboru typu XLS. Se souborem typu XLS klient může pracovat nebo si hodnoty archivovat na svém lokálním disku. Koncepce vzhledu webové aplikace umožňuje přidávat další záložky s libovolnými informacemi nebo skripty aniž by se tím narušil vzhled webové aplikace.

4.2.2 Návrh databáze

Databáze musela být navržena již před tvorbou aplikace pro webovou prezentaci, protože je databáze využita již aplikací pro obsluhu sériové linka a obsluhu databáze, která do databáze ukládá zpracovaná data.

Při návrhu databáze se běžně používá relační schéma k odstranění funkčních závislostí a struktura databáze se potom zaznamenává pomocí E-R modelu, který zachycuje vazby mezi tabulkami. Databáze pro meteostanici ovšem nemá příliš mnoho atributů a většina atributů má stejnou váhu, takže je možné databázi navrhnout i bez odstraňování funkčních závislostí a s jedním integritním omezením a to, že kombinace data a času je dostatečným identifikátorem pro jednotlivé záznamy.

Databáze bude obsahovat jednu tabulku s názvem *meteo*. Obsah tabulky *meteo* znázorňuje tabulka 4.1.

Tabulka 4.1: Rozvržení databázové tabulky meteo.

atribut	typ	popis
date	date	Datum naměření veličiny. S atributem time tvoří primární klíč
time	time	Čas naměření veličiny. S atributem time tvoří primární klíč

radiation	float	Hodnota slunečního záření.
windspeed	float	Hodnota rychlosti větru.
winddirection	float	Hodnota směru větru.
humidity	float	Hodnota vlhkosti vzduchu.
temperature	float	Hodnota teploty vzduchu.
pressure	float	Hodnota atmosferického tlaku.
rainfall	float	Hodnota srážek.

4.2.3 Skript PHP na serveru – aktuální hodnota

Ve webové aplikaci pro prezentaci dat z meteostanice je PHP skript použit ke komunikaci s databází MySQL a ke generování XML dokumentu obsahující poslední naměřené hodnoty z meteostanice, se kterým dále na straně klienta pracuje technologie AJAX. Vygenerování XML dokumentu je nutné proto, že PHP skript probíhá na serveru, ale XML dokument vygenerovaný PHP skriptem se vygeneruje u klienta a tak se naměřené hodnoty přes XML dokument předají na stranu klienta, kde se provádějí skripty technologie AJAX.

PHP skript k navázání spojení s databází používá funkci `mysql_connect()`, po úspěšném navázání spojení s databázovým serverem se použije funkce `mysql_select_db()` k vybrání konkrétní databáze.

Po vybrání databáze lze posílat na databázi SQL dotazy. K výběru záznamů z databáze se používá SQL klauzule `SELECT`. K odeslání SQL dotazu na vybranou databázi slouží v PHP funkce `mysql_query()`.

Pro zobrazení aktuálních hodnot je potřeba získat poslední naměřené hodnoty, což je poslední záznam v databázi. V SQL není klauzule pro získání posledního záznamu z tabulky, ale jsou dva způsoby jak tento záznam získat aniž bychom musely načítat celou tabulku. Prvním způsobem je použití SQL dotazu pro zjištění kolik je v tabulce záznamů a následně použít výběrový dotaz doplněný SQL klauzulí `LIMIT`, která má dva argumenty. První argument určuje od kterého záznamu se má z tabulky vybírat a druhým argument určuje kolik záznamů se má vybrat.

```
$pocet_radku = SELECT COUNT(*) FROM meteo;
$posledni_zaznam = SELECT * FROM meteo LIMIT $pocet_radku, 1;
```

Druhou možností je použít výběrový dotaz doplněný třídící SQL klauzulí

ORDER BY DESC, kterou záznamy seřídíme sestupně podle data a času, čímž získáme nejnovější záznam na první pozici a SQL klauzulí LIMIT vybereme jen první záznam.

```
$posledni_zaznam = SELECT * FROM meteo ORDER BY date, time  
DESC LIMIT 1, 1;
```

Ač se zdá druhý způsob jednodušší, vzhledem k tomu, že používá jen jeden SQL dotaz, tak byl použit první způsob, protože hodnota získaná při zjišťování počtu řádků je dále používána při generování XML dokumentu k identifikaci.

Po získání posledního záznamu je záznam potřeba rozpársovat. Rozpársování odpovědi na výběrový dotaz se provede PHP funkcí `mysql_fetch_array(odpověd, způsob indexace)`. Funkce `mysql_fetch_array` odpověď rozděljuje na jednotlivé záznamy a každý záznam rozdělí do pole podle atributů. Lze zvolit jestli se jedná o pole asociativní, indexované nebo kombinované.

Nejvýhodnější je použít asociativní pole, protože umožňuje přistupovat k jednotlivým hodnotám přes název atributu a nemusí se znát žádné indexy. K rozdělení do asociativního pole se použije argument `MYSQL_ASSOC`.

Z rozpársovaných hodnot se vygeneruje XML dokument obsahující jednotlivé hodnoty. Aby PHP skript mohl XML dokument vygenerovat musí kromě XML struktury obsahovat i hlavičku XML souboru, která interpret informuje o tom, že se jedná o XML dokument a ne o PHP skript. Hlavička pro PHP generující XML dokument je ve tvaru:

```
header("Content-Type: text/xml; charset=utf-8");
```

Struktura XML dokumentu je potom ve tvaru:

```
<?xml version="1.0" ?>  
<root>  
  <DATA id="'.$pocet_radku.'">  
    <DATE>datum</DATE>  
    <TIMEE>čas</TIMEE>  
    <RADIATION>sluneční záření</RADIATION>  
    <WINDSPEED>rychlost větru</WINDSPEED>  
    <WINDDIRECTION>směr větru</WINDDIRECTION>  
    <HUMIDITY>vlhkost vzduchu</HUMIDITY>  
    <TEMPERATUR>teplota vzduchu</TEMPERATUR>  
    <PRESSURE>atmosferický tlak</PRESSURE>
```

```
<RAINFALL>srážky</RAINFALL>
</DATA>
</root>
```

4.2.4 Skript PHP na serveru – historie hodnot

Každá měřená veličina má svůj vlastní PHP skript pro získávání historických hodnot z MySQL databáze. PHP Skripty se liší jen v získávané veličině.

PHP skript navazuje spojení a vybírá databázi stejným způsobem jako PHP skript pro získávání aktuálních hodnot. Rozdíl je pouze ve výběrovém SQL dotazu, protože v tomto případě jsou potřeba všechny hodnoty od jedné veličiny u které chceme zobrazit historická data a ne jen poslední hodnoty všech veličin.

Aby se jednotlivé hodnoty od dané veličiny daly přiřadit i ke konkrétnímu datu a času a tím je zobrazit v grafu, tak jsou potřeba navíc ještě všechny záznamy data a času. Výsledný výběrový SQL dotaz má tedy tvar:

```
$zaznamy = SELECT date, time, veličina FROM meteo
```

Po získání odpovědi je potřeba odpověď opět rozpársovat funkcí `mysql_fetch_array` na jednotlivé záznamy a atributy.

Vzhledem k tomu, že v tomto případě je potřeba datum a čas mít pro další použití ve formátu uplynulých sekund od 1. 1. 1970 je potřeba získaný datum a čas přetypovat z datového typu `String` ve kterém MySQL všechny hodnoty vrací na datový typ `time`. Tohoto přetypování se docílí funkcí `strToTime()`. V případě že se v PHP skriptu používá funkce `strToTime()`, tak je vhodné ve skriptu uvést v jakém časovém pásmu je umístěn server na kterém PHP skript běží. Časové pásmo umístění serveru se určí funkcí `date_default_timezone_set('Europe/Prague')`.

Rozpársované hodnoty se potom předávají JavaScriptu, který z hodnot vykreslí graf. Jedná se o synchronní JavaScript a tak není nutné hodnoty předávat přes XML dokument jako je tomu u technologie AJAX.

4.2.5 Skript PHP na serveru – export dat do XLS

Aby uživatelé mohli pracovat s naměřenými hodnotami, tak je vytvořen PHP skript, který uživateli umožňuje naměřená data exportovat do XLS dokumentu, se kterým se dá pracovat v běžných kancelářských balících jako je Microsoft Office,

Open Office apod.

Při vytváření XLS dokumentu je důležité hodnoty z databáze vhodně rozdělit do jednotlivých buněk XLS dokumentu. Kancelářské balíky dokáží rozpoznat několik způsobů oddělení buněk.

První možností rozdělení hodnot do buněk je oddělovat jednotlivé atributy středníkem (;) a jednotlivé záznamy odřádkovat.

```
date;time;radiation;windspeed;winddirection;temperature;...  
datum1;čas1;záření1;rychlostvětru1;směr větru1;teplota1;...  
datum2;čas2;záření2;rychlostvětru2;směr větru2;teplota2;...
```

Druhým způsobem je vytvořit HTML stránku, která obsahuje tabulku vytvořenou pomocí HTML tagů, kde jsou jednotlivé záznamy rozděleny do řádků párovým tagem <tr> a jednotlivé atributy do sloupců tagem <td> a uložit HTML stránku jako XLS dokument.

```
<table>  
  <tr>  
    <td>date</td><td>time</td><td>radiation</td>...  
  </tr>  
  <tr>  
    <td>datum1</td><td>čas1</td><td>záření1</td>...  
  </tr>  
  <tr>  
    <td>datum2</td><td>čas2</td><td>záření2</td>...  
  </tr>  
</table>
```

Oba předchozí způsoby vyžadují před uživatelským otevřením XLS dokumentu potvrzení, že program ve kterém dokument otevíráme správně rozpoznal způsob rozdělení jednotlivých buněk. To může být ovšem pro uživatele matoucí a proto byl použit způsob třetí, který žádné potvrzení nevyžaduje, protože jsou použity pro oddělení datové pakety dle specifikace XLS dokumentu. Tento způsob je náročnější, ale výsledek exportu je bezproblémový a uživatelsky nejpřívětivější. Celá specifikace je dostupná na webových stránkách společnosti Microsoft.[10]

Datový paket značící začátek XLS dokumentu:

(0x809, 0x08, 0x00, 0x10, 0x00, 0x00)

Datový paket značící konec XLS dokumentu:

(0x0A, 0x00)

Datový paket značící umístění a obsah jednotlivých buněk XLS dokumentu:

(0x204, 8 + velikost buňky, číslo řádku, číslo sloupce, 0x00, velikost buňky, obsah buňky)

PHP skript pro export dat do XLS tedy naváže spojení a vybere databázi jako PHP skripty předchozí. Výběrový SQL dotaz je uživatelsky modifikovatelný pomocí přehledných zaškrtačacích boxů, kde si uživatel vybírá které veličiny chce exportovat a v textových boxech si vybírá časový rozsah exportovaných hodnot. Výběrový SQL dotaz je následně odeslán na databázi a odpověď se opět rozdělí na jednotlivé záznamy a atributy funkcí `mysql_fetch_array`.

Export dat do XSL dokumentu (Microsoft Excel, OpenOffice Calc, LibreOffice Calc)

Teplota:

Sluneční záření:

Vlhkost vzduchu:

Směr větru:

Rychlost větru:

Tlak:

Srážky:

Datum od:

Datum do:

Obrázek 4.3: Uživatelské modifikování výběrového SQL dotazu.

Při zpracování odpovědi se zaznamenává o kolikátý záznam a o kolikátý atribut se jedná, aby se dalo určit umístění budoucí buňky, tak jak to vyžaduje datový paket pro označení jednotlivých buněk. Doplněny jsou samozřejmě také pakety pro označení začátku a konce XLS dokumentu.

Po zpracování celé odpovědi a vygenerování XLS dokumentu dojde k automatickému stahování XLS dokumentu na lokální disk uživatele.

4.2.6 AJAX v prezenčním rozhraní

Technologie AJAX potřebuje ke své existenci vytvořit objekt `XMLHttpRequest` pro přenos informací ze serveru na pozadí prezenčního rozhraní. Prezenční rozhraní je proto potřeba doplnit o JavaScriptové funkce, které se o vytvoření objektu `XMLHttpRequest` starají.

Ve webovém prohlížeči Internet Explorer se objekt `XMLHttpRequest` implementuje jako objekt typu `ActiveX`, kdežto ve všech ostatních webových prohlížečích jako je Mozilla Firefox, Safari, Opera a Google Chrome se implementuje jako nativní objekt JavaScriptu. Z toho důvodu musí JavaScript zjistit jakým způsobem má objekt `XMLHttpRequest` vytvořit.

Jedním způsobem při volbě způsobu vytvoření objektu `XMLHttpRequest` je zjistit v jakém typu webového prohlížeče se JavaScript vykonává. To je ale poněkud neuniverzální řešení, protože s nástupem nových webových prohlížečů by toto řešení nemuselo fungovat. Jednodušším způsobem je použití podmínky, zda JavaScript dokáže vytvořit JavaScriptový objekt `XMLHttpRequest`. Pokud JavaScript tento objekt vytvořit nedokáže, tak se vytvoří objekt `Microsoft.XMLHttp` typu `ActiveX`, čímž se pokryjí požadavky všech současných i budoucích webových prohlížečů podporujících JavaScript.

```
if (window.XMLHttpRequest)
    {rReq = new XMLHttpRequest();}
else if (window.ActiveXObject)
    {rReq = new ActiveXObject("Microsoft.XMLHttp");}
```

Objekt `XMLHttpRequest` obsahuje několik metod a parametrů, které nám umožní asynchronní komunikaci se serverem.

Aby bylo možné asynchronně zobrazovat naměřené hodnoty na webové stránce byl vytvořen PHP skript generující XML dokument (viz. kapitola 4.2.3). Nyní pomocí objektu `XMLHttpRequest` budeme XML dokument zpracovávat. Ke zpracování XML dokumentu použijeme metodu `open` objektu `XMLHttpRequest`.

```
rReq.open("GET, inner/xml.php, true");
```

Kde `GET` je metoda značící, že od serveru požadujeme data. `inner/xml.php`

je URL stránky obsahující skript generující XML dokument a poslední parametr `true` určuje, že se požadavek má vykonat asynchronně.

Pomocí události `readyState` objektu `XMLHttpRequest` lze testovat stav odpovědi metody `open`. Zajímavým stavem je stav 4, který značí, že odpověď metody `open` je kompletní a lze pokračovat ve zpracování XML dokumentu.

Když je odpověď kompletní je možné XML dokument zpracovat atributem `responseXML` a vypársovat z XML dokumentu jednotlivé hodnoty. Vypársování hodnot z XML dokumentu je díky struktuře XML dokumentu jednoduché a tak lze jednoduše přistupovat k jednotlivým položkám.

Zpracování XML dokumentu:

```
var xmlDoc = rReq.responseXML;
```

Načtení obsahu položky DATA, která obsahuje všechny ostatní položky:

```
var uzly = xmlDoc.getElementsByTagName("DATA");
```

Přístupování k jednotlivým položkám, které obsahují naměřené hodnoty:

```
var radiation = uzly[0].getElementsByTagName("RADIATION");  
dateGauge = parseFloat(radiation[0].firstChild.nodeValue);
```

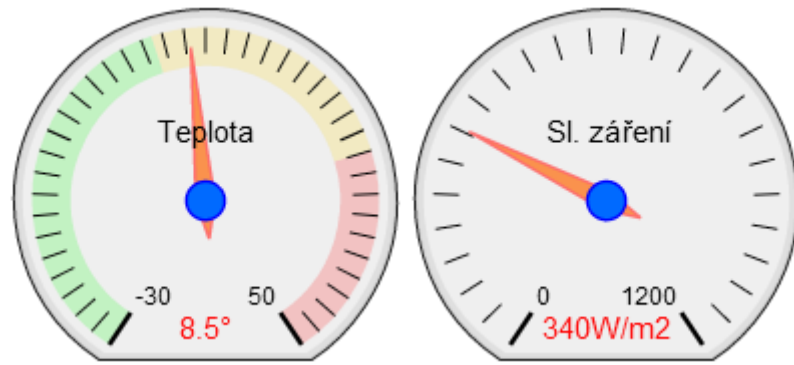
Naměřené hodnoty se takto dostaly z MySQL databáze až do proměnné v JavaScriptu, kde nyní JavaScript s touto hodnotou může pracovat a dynamicky tak měnit obsah prezenčního rozhraní.

4.2.7 API v prezenčním rozhraní

K zobrazování dynamicky měnících se hodnot je použit gauge metr. Za tímto účelem bylo použito API `jsGauge`[11], které je šířeno pod MIT licenci[12]. `JsGauge` využívá vizualizační komponenty JavaScriptu a ve webové prezentaci se vykresluje pomocí elementu `Canvas`.

U `jsGauge` lze pomocí metody `setOptions` měnit základní nastavení gauge metru jako jsou zobrazované jednotky, rozsah, minimální a maximální hodnota a barvy gauge metru.

Metodou `setValue` lze nastavit zobrazovanou hodnotu. Zobrazovaná hodnota je uložena v JavaScriptové proměnné, do které ji ukládá technologie AJAX (viz. kapitola 4.2.6). API `jsGauge` pomocí metody `setValue` v nastavených intervalech kontroluje obsah zmíněné proměnné a při změně jejího obsahu dynamicky překreslí gauge metr.

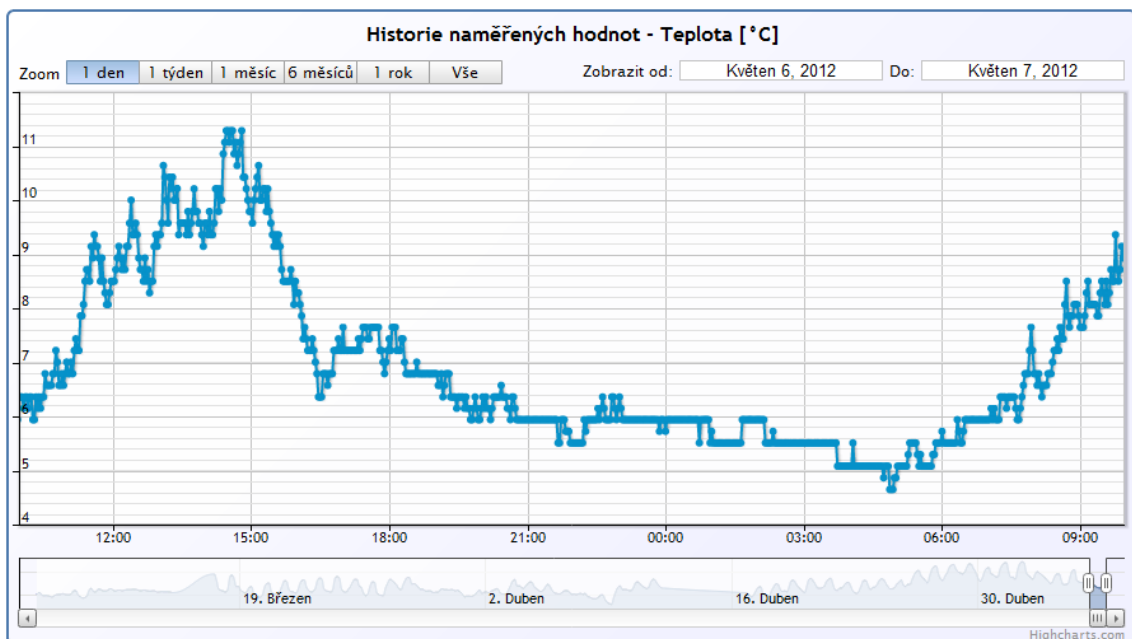


Obrázek 4.4: Vzhled gauge metru.

K zobrazování historických hodnot je použitý graf. K vykreslování grafů je použito API Highstock. API Highstock umožňuje vykreslování grafů s časovou osou. API Highstock je naprogramováno v JavaScriptu a pro nekomerční použití je volně použitelný.

API Highstock obsahuje metody pro úpravu vzhledu grafu. Pomocí metod lze nastavovat maximální přiblížení, přednastavit základní přiblížení jako je přiblížení na jeden den, jeden týden, jeden měsíc a podobně. Další metody se používají k popisům grafu jako je nadpis, jednotky a barvy grafu.

V grafu lze vyhledávat pomocí lišty pod přiblíženou částí grafu, která obsahuje celý graf. Pomocí této lišty je možné nastavovat i rozsah přiblížené části. Vyhledávat v grafu lze i pomocí data zadáním rozsahu.



Obrázek 4.5: Vzhled grafu.

5 Instalace meteostanice

Meteostanice je nainstalována v budově Střední školy strojní, stavební a dopravní v Liberci v Horské ulici 165.

I/O modul je zabudován v konstrukční krabici společně s napájecím zdrojem a vyhodnocovací deskou WO 21-I. Do konstrukční krabice je přiveden více žilový kabel od meteorologických čidel, která jsou umístěna na střeše budovy. Pomocí více žilového kabelu jsou přenášeny od meteorologických čidel signály do I/O modulu a meteorologická čidla jsou tímto kabelem i napájena. Dále je z konstrukční krabice vyveden USB kabel, kterým je I/O modul propojen s PC.

Na připojeném PC je nainstalován databázový server a webový server s podporou PHP. Operačním systémem PC je Microsoft Windows XP. Jako databázový server byl použit server MySQL a webový server je Apache s modulem PHP. Celá konfigurace serveru je tedy typu WAMP.

V databázovém MySQL serveru byla vytvořena databáze `meteo` a v ní tabulka `meteo` obsahující atributy podle tabulky 4.1.

Na webový server byla nahrána webová aplikace pro prezentaci dat z meteostanice.

Na připojené PC bylo nainstalováno JRE, což je běhové prostředí pro aplikace naprogramované v Javě. JRE obsahuje virtuální stroj se zavaděčem, který umožní spuštění aplikace pro obsluhu sériové linky a databáze, která je na PC nahraná a spouští se s parametrem `java`, který umožní zavedení aplikace do virtuálního stroje.

Pomocí všech výše zmíněných hardwarových a softwarových prostředků byla meteostanice úspěšně nainstalována a uvedena do provozu a je nyní dostupná v celosvětové počítačové síti internet.

6 Závěr

Výsledkem diplomové práce je nově vyvinuté plně funkční hardwarové a softwarové vybavení meteostanice v Kateřinkách, které plně odpovídá zadání práce a všem předem zadaným požadavkům.

Vytvořený I/O modul zpracovává analogové signály do digitální podoby přesně podle potřeb a s dostatečným rozlišením. I přesto že I/O modul je konstruován pro vstupní napětí A/D převodníku v rozsahu napětí 0 - 3 V, tak je schopen zpracovat libovolný běžný analogový signál. V případě že se jedná o napět'ový signál většího rozsahu, tak se signál upraví napět'ovým děličem a pro proudové signály se zvolí vhodný rezistor, který se použije jako jednoduchý převodník proud-napětí.

Aplikace v PC pro příjem naměřených hodnot z I/O modulu a jejich ukládání do databáze je napsána v programovacím jazyce JAVA a využívá bezplatně dostupná API pro I/O operace a práce nad databází. I přes to, že I/O operace mnohdy snižují multiplatformnost, tak použité API RXTX je tak pokročilé, že multiplatformnost nijak neomezuje ani při I/O operacích.

Webová aplikace přehlednou formou dynamicky prezentuje nejaktuálnější hodnoty i hodnoty historické, které byly meteostanicí naměřeny.

Hardwarové a softwarové vybavení bylo v Kateřinkách úspěšně nainstalováno a napojeno na soustavu meteorologických čidel a od 7. března 2012 je meteostanice v provozu.

Při napojení na soustavu meteorologických čidel byla zjištěna nefunkčnost několika meteorologických čidel a proto se nefunkční meteorologická čidla budou nahrazovat. I přes to, že se budou meteorologická čidla měnit, tak to nebude mít žádný vliv na současnou funkci I/O modulu ani na softwarové vybavení, jelikož celý systém byl navržen tak, aby mohly být jednotlivé části rychle vyměněny a dle potřeb upraveny.

Použitá literatura

- [1] VEDRAL, J. – FISCHER, J. Elektronické obvody pro měřící techniku. Praha, ČVUT, 2004. (ISBN 80-01-02966-2).
- [2] ATMEL, Datasheet AT89C51CC03. Dostupné z:
<<http://www.atmel.com/Images/doc4182.pdf>>
- [3] FTDI CHIP, Datasheet FT232R. Dostupné z:
<http://www.ftdichip.com/Support/Documents/Datasheets/ICs/DS_FT232R.pdf>
- [4] ANALOG DEVICES, Datasheet ADuM 3160. Dostupné z:
<http://www.analog.com/static/imported-files/data_sheets/ADuM3160.pdf>
- [5] RXTX, Komunitní projekt RXTX.org. Dostupné z: <<http://rxtx.qbang.org>>
- [6] GNU, Lesser General Public License. Dostupné z:
<<http://www.gnu.org/licenses/lgpl.html>>
- [7] ORACLE, Java Communication API. Dostupné z:
<http://docs.oracle.com/cd/E17802_01/products/products/javacomm/reference/api/javax/comm/package-summary.html>
- [8] ORACLE, Java SE Technologies – Database. Dostupné z:
<<http://www.oracle.com/technetwork/java/javase/jdbc/index.html>>
- [9] LACKO, L. *AJAX Hotová řešení*. První vydání. Brno, Computer Press, 2008. (ISBN 978-80-251-2108-5).
- [10] MICROSOFT, Excel 97-2007 Binary File Format (.xls) Specification. Dostupné z:
<[http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/Excel97-2007BinaryFileFormat\(xls\)Specification.pdf](http://download.microsoft.com/download/0/B/E/0BE8BDD7-E5E8-422A-ABFD-4342ED7AD886/Excel97-2007BinaryFileFormat(xls)Specification.pdf)>
- [11] TIHAUAN, D. API jsGauge. Dostupné z: <<http://code.google.com/p/jsgauge/>>
- [12] OSI, The MIT License. Dostupné z:
<<http://www.opensource.org/licenses/mit-license.php>>

Seznam obrázků

Obrázek 1.1: *Meteostanice jako Distributed Presentation se vzdáleným serverem.*

Obrázek 1.2: *Meteostanice jako Remote Data Management.*

Obrázek 1.3: *Meteostanice jako Distributed Presentation s lokálním serverem.*

Obrázek 2.1: *Blokové schéma I/O modulu s mikropočítačem.*

Obrázek 2.2: *Blokové zapojení galvanického oddělení.*

Obrázek 2.3: *Vyrobený I/O modul s mikropočítačem.*

Obrázek 2.4: *Vývojový diagram programu pro mikropočítač.*

Obrázek 3.1: *Vývojový diagram pro obsluhu sériové linky.*

Obrázek 3.2: *Znázornění architektury JDBC.*

Obrázek 3.3: *Vývojový diagram pro obsluhu databáze.*

Obrázek 4.1: *Schématické znázornění funkce webové aplikace.*

Obrázek 4.2: *Rozložení vzhledu webové aplikace.*

Obrázek 4.3: *Uživatelské modifikování výběrového SQL dotazu.*

Obrázek 4.4: *Vzhled gauge metru.*

Obrázek 4.5: *Vzhled grafu.*

Seznam tabulek

Tabulka 2.1: *Význam bitů v SFR ADCF.*

Tabulka 2.2: *Význam bitů v SFR ADCON.*

Tabulka 2.3: *Výběr kanálu A/D převodníku.*

Tabulka 2.4: *Význam bitů v SFR ADDH.*

Tabulka 2.5: *Význam bitů v SFR RCAP2H a RCAP2L.*

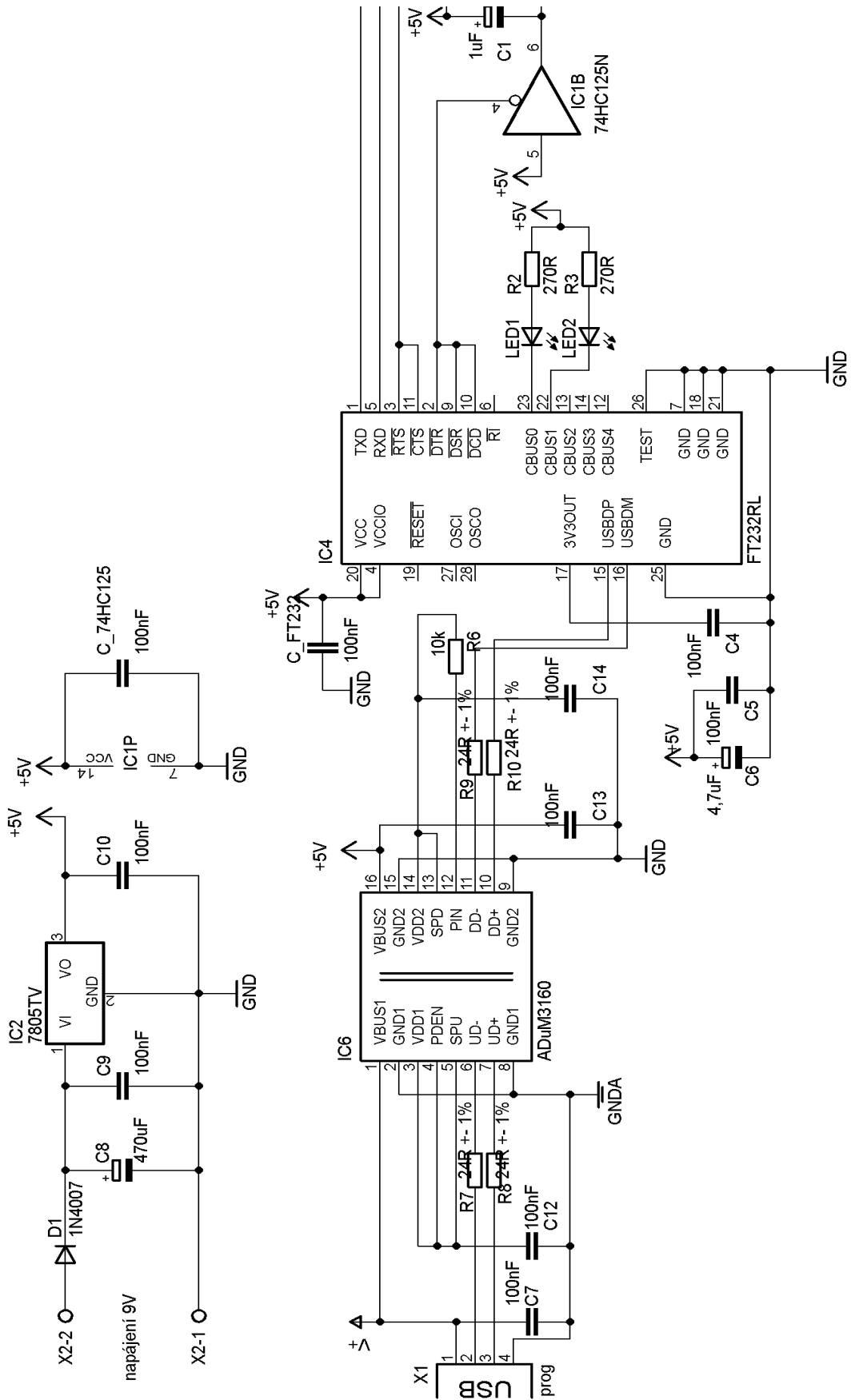
Tabulka 2.6: *Význam bitů v SFR T2CON.*

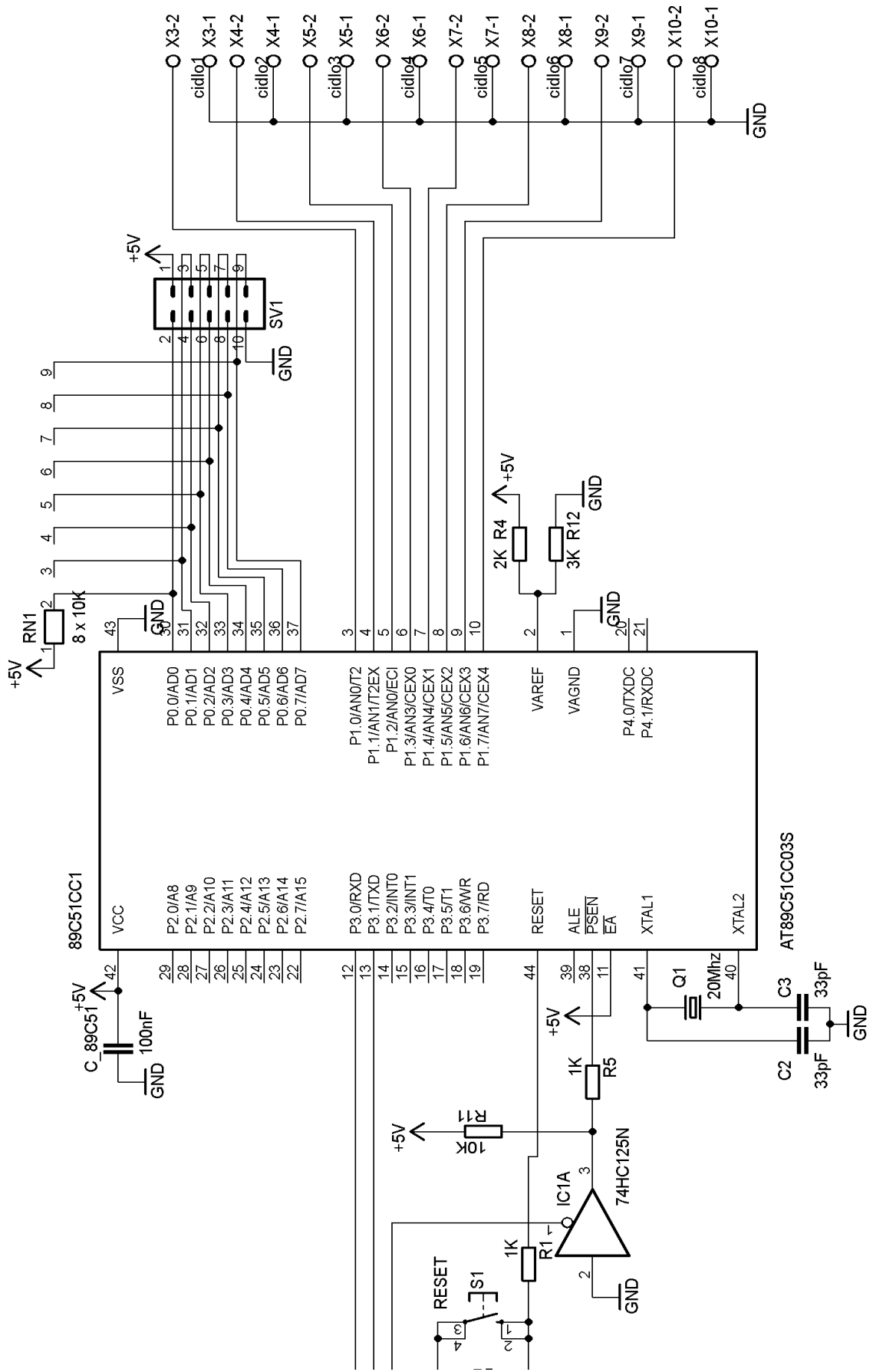
Tabulka 2.7: *Význam bitů v SFR SCON.*

Tabulka 4.1: *Rozvržení databázové tabulky meteo.*

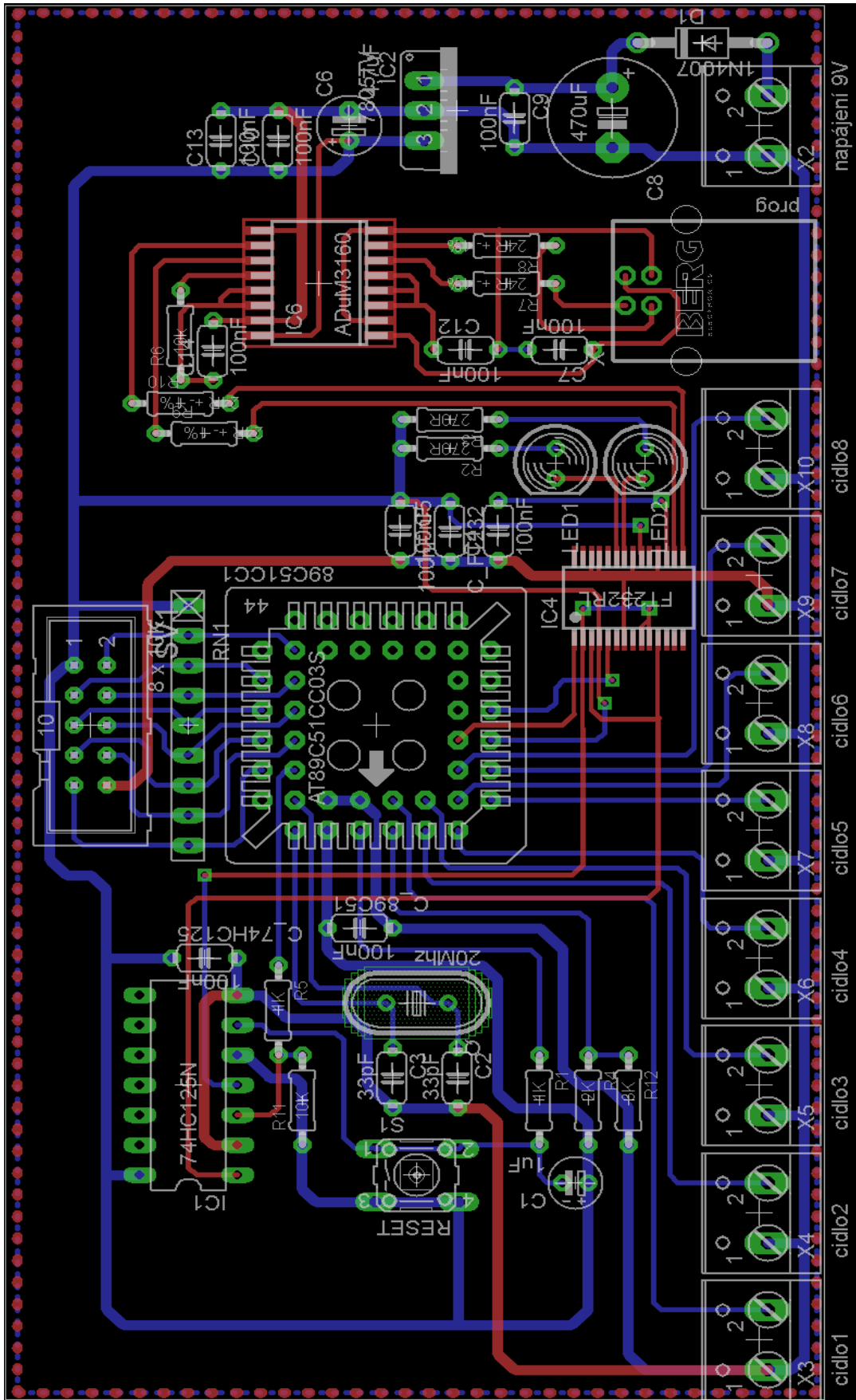
Přílohy

Schéma I/O modulu



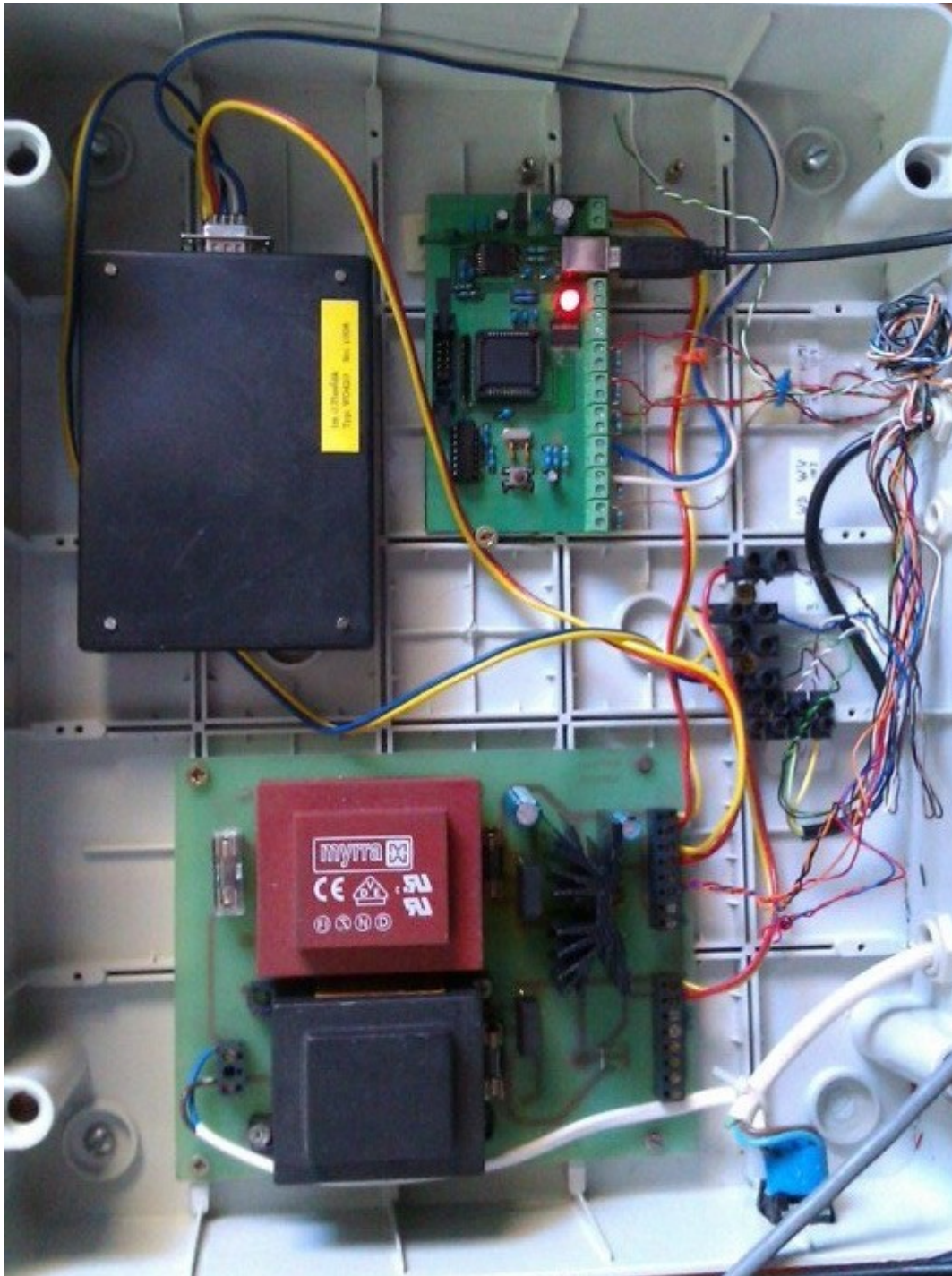


DPS I/O modulu



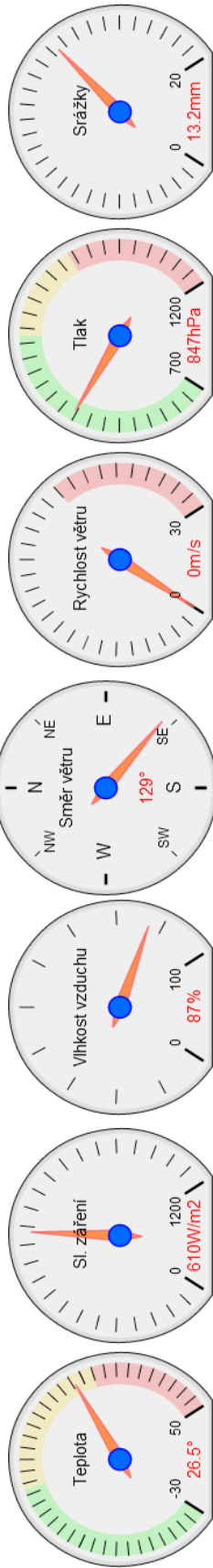
Fotodokumentace

Instalovaný I/O modul.



Vzhled webové aplikace.

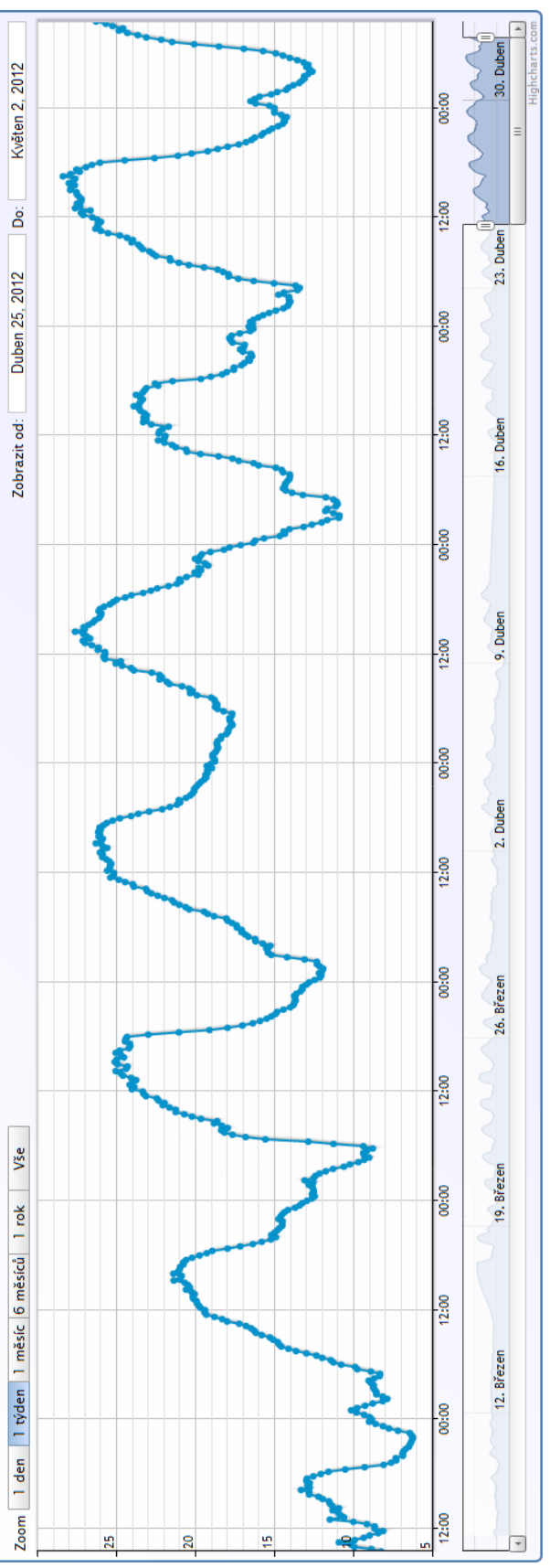
Meteostanice Kateřinky - Liberec



Aktuální hodnota 10:31:47

[Teplota](#) | [Sluneční záření](#) | [Vlhkost vzduchu](#) | [Směr větru](#) | [Rychlost větru](#) | [Tlak](#) | [Srážky](#) | [Export dat do XLS](#) | [Připomínky](#)

Historie naměřených hodnot - Teplota [°C]



Vzhled aplikace pro obsluhu sériové linky a databáze.

Nastavení připojení k MySQL serveru

Host

Port

Databaze

Uživatel

Heslo

Vybrat COM port

Vypis konzole

COM1 je volný

Poslední přijatá data

Přílohy v elektronické podobě – CD

Na přiloženém CD jsou kompletní přílohy práce v elektronické podobě:

- text této diplomové práce ve formátu PDF
- schéma zapojení I/O modulu ve formátu SCH
- návrh DPS I/O modulu ve formátu BRD
- zdrojové kódy aplikace pro obsluhu sériové linky a databáze
- zkompilovaná aplikace pro obsluhu sériové linky a databáze
- Skripty kompletní webové aplikace
- softwarové prostředky pro prohlížení všech elektronických příloh
 - Adobe Reader
 - Eagle
 - NetBeans