

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH  
PEDAGOGICKÁ FAKULTA  
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

ZÍSKÁVÁNÍ DAT Z MĚŘÍCÍHO PŘÍSTROJE  
A PUBLIKOVÁNÍ VÝSLEDKŮ NA WEBU

MAREK SYSEL

VEDOUcí BAKALÁŘSKÉ PRÁCE: ING. LADISLAV BERÁNEK, CSC., MBA

ČESKÉ BUDĚJOVICE 2008

## Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě, elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne 25.dubna 2008

## Anotace

Tato bakalářská práce se zabývá získáváním dat z měřicího přístroje, zkonstruovaného za účelem poskytnutí ukázky měření směru a rychlosti větru a teploty vzduchu a prezentováním získaných výsledků na webové stránce.

Hlavním cílem předkládané bakalářské práce bylo objasnit problematiku získávání dat, a také poskytnout ucelené informace z oblasti měření neelektrických veličin které mohou sloužit při ukázkách měření v laboratorních cvičení z fyziky nebo elektroniky.

Práce je rozdělena do dvou částí. V první části je uvedena teorie o měření neelektrických veličin a jejich snímačích. Následuje kapitola popisující zpracování naměřených dat a kapitola zabývající se přenosem informace. V druhé praktické části se práce věnuje popisu fungování měřicího přístroje a koncepcí návrhu softwaru, který zajišťuje přenos dat z měřicího přístroje až po zobrazení výsledku na internetu.

## Abstract

This bachelor thesis deals with obtaining data from a measuring instrument, which was created to demonstrate wind direction, wind speed and air temperature measuring, and presenting its results on a web site.

The main aim of the thesis was to illustrate problems of data obtaining and also to provide complex information on measuring non-electric quantities which can be used for laboratory practice of physics and electronics.

The thesis is divided into two parts. The first one presents a theory of measuring non-electric quantities and their sensors. It is followed by two chapters describing procedure of data processing and data transfer. The practical part explains how the measuring instrument works and presents a suggestion for its software which enables data transfer from the measuring instrument to the internet web site.

## Poděkování

Děkuji vedoucímu mé bakalářské práce Ing. Ladislavu Beránkovi za připomínky, odborné konzultace a vstřícnost nejen při psaní této práce ale i během mého studia.

# Obsah

1. Úvod.....	8
2. Teorie měření neelektrických veličin .....	10
2.1. Měřicí soustava .....	10
2.2. Vlastnosti měřicí soustavy .....	11
2.2.1. Statické vlastnosti .....	11
2.2.2. Dynamické vlastnosti .....	15
2.3. Snímače neelektrických veličin.....	18
2.3.1. Teplotní senzor SMT-160.....	18
2.3.2. Optoelektronické snímače:.....	18
3. Přenos informace .....	21
3.1. Sériový přenos .....	21
3.1.1. Sériový asynchronní přenos .....	21
3.1.2. Sériový synchronní přenos.....	22
3.2. Standard RS-232.....	23
3.2.1. USB port.....	23
3.2.2. Převodník USB/RS-232 .....	24
4. Zpracování dat .....	25
4.1. Proud dat ( streaming).....	25
4.2. Komprese dat .....	27
4.2.1. Bezeztrátová komprese .....	28
4.3. Grafické znázornění průběhu dat. ....	29
4.3.1. Popis knihovny JpGraph .....	30
5. Měřicí zařízení .....	32
5.1. Popis měřícího zařízení .....	33
5.2. Řídící jednotka .....	33

5.3.	Měřicí jednotka.....	36
5.3.1.	Teplotní senzor.....	36
5.3.2.	Senzor rychlosti.....	37
5.3.3.	Optický senzor polohy.....	40
6.	Software řídicí jednotky.....	42
6.1.	Deklarace proměnných.....	42
6.2.	Inicializace.....	42
6.3.	Hlavní běh programu.....	42
6.4.	Důležité části programu.....	44
7.	Software pro načítání dat do počítače a přenos na internet.....	48
7.1.	Program meteoK3.....	49
7.1.1.	Konfigurace.....	49
7.2.	Běh aplikace.....	50
7.3.	Přenos dat na internet.....	52
8.	Zobrazování výsledků na webové stránce.....	55
8.1.	Požadavky pro běh webových stránek.....	55
8.2.	Struktura webových stránek.....	56
8.2.1.	Rozvržení stránky.....	56
8.3.	Skripty PHP.....	57
9.	Závěr.....	60
	Literatura.....	62
	Příloha.....	64

## 1. Úvod

Měření neelektrických veličin jako je například teplota, proudění vzduchu nebo srážková činnost, je velice zajímavým procesem při němž se využívá poznatků z oblasti fyziky, elektroniky a informatiky. Tato problematika dala impuls k tomu aby autor této práce využil současných znalostí, které získal nejen studiem na vysoké škole ale i v průběhu středoškolského studia. Již tehdy se také zrodila touha porozumět mechanismu, který je schopen převádět tyto veličiny do digitální podoby. Pro porozumění celé mechaniky byl zkonstruován amatérský anemometr, jehož úkolem je měřit tři fyzikální veličiny - směr větru, rychlost větru a teplotu vzduchu. Celý přístroj je určen výhradně pro pokusy a experimenty a neklade si nároky být nasazen do reálného prostředí a dosahovat kvalit profesionálních anemometrů. Od popisovaného přístroje tedy nelze očekávat profesionální výsledky. Pro hodnocení funkčnosti přístroje a dosahovaných výsledků je nezbytné brát v úvahu fakt, že vznikl v amatérských podmínkách, čemuž odpovídá celá konstrukce a zvolený materiál.

V průběhu testování funkčnosti přístroje vyvstala potřeba zobrazovat změřená data nejen na displeji přístroje, ale také na počítači, a následně je publikovat na webu. Cílem předkládané bakalářské práce je nejen popsat problematiku získávání dat z měřícího přístroje a prezentace výsledků na webu, ale zároveň se pokusit rozšířit poznatky v oblasti amatérské měřící techniky. Práce může být přínosem pro pedagogy, kteří ji využijí v rámci laboratorních cvičení z fyziky, ale rovněž pro studenty nebo pro elektromechaniky-amatéry. Autorovou snahou je také poskytnout praktické informace a rady, které získal v průběhu programování uživatelské aplikace a realizací webových stránek, na němž jsou výsledky zobrazeny.



Práce je rozdělena do dvou částí. V teoretické části jsou uvedeny způsoby měření neelektrických veličin, senzory snímačů těchto veličin, základy teorie o přenosu informace a zpracování dat. Praktická část obsahuje popis měřícího zařízení, princip fungování přenosu dat do počítače, software pro načítání dat z počítače a software pro prezentaci výsledků na webové stránce.

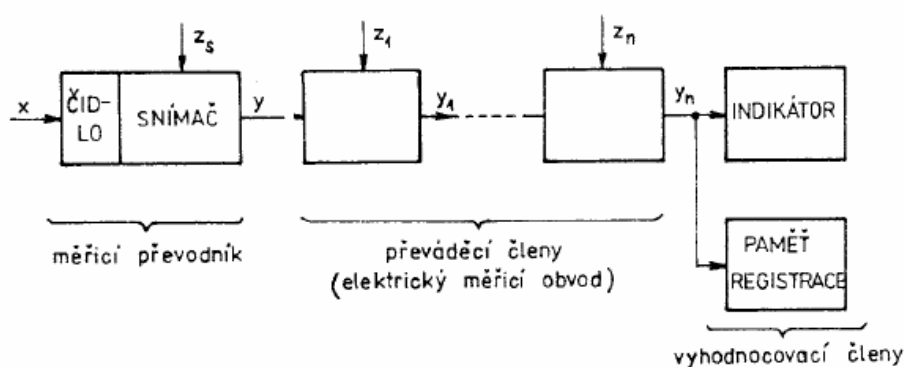
## 2. Teorie měření neelektrických veličin

### 2.1. Měřicí soustava

Měřicí soustava (měřicí řetězec) je soubor měřicích členů (jednotek) účelně uspořádaných tak, aby bylo možné splnit požadovaný úkol měření, tj. získat informaci o velikosti fyzikální veličiny na měřeném objektu. Nejdůležitějším členem měřicího řetězce obrázek (2.1) je snímač, jehož první část označovaná jako čidlo je v přímém styku s měřeným objektem a přijímá od něj energii. Druhou část tvoří elektrický měřicí obvod (EMO).

Je důležité si uvědomit, že při každém měření dochází k odčerpání části energie z měřeného objektu, tj. objekt je vždy měřením rušen a teoreticky nelze dosáhnout měření bez chyby. Výstupní veličina čidla je zpravidla neelektrická (např. mechanický pohyb) a může být u složitějších snímačů ještě několikrát transformována na jiné neelektrické veličiny uvnitř snímače [1].

Výstupní elektrická veličina snímače je dále zpracována v elektrickém měřicím obvodu na tvar a velikost požadovanou pro vyhodnocení. Elektrický měřicí obvod (EMO) je složen z převáděcích členů jako jsou zesilovače, generátory, můstkové obvody, filtry, počítačící obvody, atd).



Obrázek 2.1: *Blokové schéma měřicího řetězce*

Výstupní veličina z těchto členů je zpracována vyhodnocovacími prvky na formu přístupnou lidskému vnímání. Typickými vyhodnocovacími členy pro analogový údaj jsou ručkové měřicí přístroje, pro číslicový údaj číslicové displeje (svítící segmenty, kapalně krystaly). Důležitým členem vyhodnocovacího zařízení je paměť, zajišťující uchování informace o hodnotě měřené veličiny po určitou dobu. Pro analogový signál se jako paměť užívá zapisovačů nebo měřicích magnetofonů, pro číslicové signály polovodičových pamětí, magnetických nebo optických disků [6].

## **2.2. Vlastnosti měřicí soustavy**

Jak již bylo řečeno, nejdůležitějším členem je snímač, protože určuje vlastnosti celého řetězce. Je to zejména proto, že chyby vzniklé ve snímači buď nelze odstranit vůbec, nebo jen velmi obtížně. K posouzení vhodnosti snímače k danému účelu je nutné znát statické a dynamické vlastnosti snímače [1].

### **2.2.1. Statické vlastnosti**

Statické vlastnosti měřicího řetězce popisují jeho chování v časově ustáleném stavu a jsou dány statickými vlastnostmi jednotlivých členů. Jsou popsány buď algebraickými rovnicemi nebo častěji zobrazeny statickými charakteristikami.

#### **Statická charakteristika**

Je závislost výstupního elektrického signálu na měřené neelektrické veličině a je nezávislá na čase a platí vztah:

$$y = f(x) \tag{1}$$

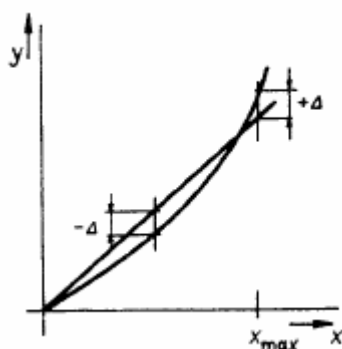
kde **y** je výstupní veličina

**x** je vstupní veličina

V nejjednodušším a často žádaném případě platí lineární vztah  $y = Kx$ , kde  $K$  je konstanta přenosu (zesílení) měřicího členu. Pro obecnou funkční závislost definujeme “konstantu” přenosu z přírůstků  $\Delta x$ ,  $\Delta y$  a tedy obecně je  $K$  funkcí vstupní veličiny  $x$  [6].

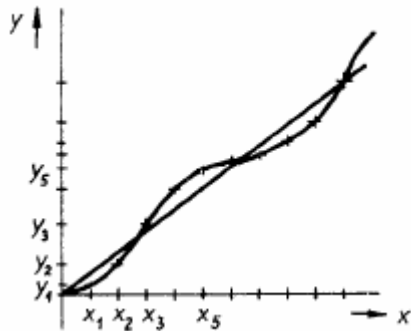
$$K = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{df(x)}{dx} \quad (2)$$

Chybou linearity většinou rozumíme odchylku skutečné charakteristiky od ideální přímkové charakteristiky. Skládá-li se řetězec z většího počtu členů s lineárními statickými charakteristikami, bude výsledná charakteristika dána výsledným zesílením vypočteným z blokového schématu. Např.: při sériovém řazení členů bude výsledné zesílení dáno součinem všech zesílení. Je-li charakteristika snímače nelineární, měla by se linearizovat použitím korekčních členů, nebo zvolením náhradní lineární charakteristiky.



Obrázek 2.2: Náhradní charakteristika

Jeli funkce  $y = f(x)$  měřicího řetězce složitá a je-li tato funkce měřením zjištěna, je výhodné zvolit za náhradní charakteristiku *empirickou regresní funkci* obrázek (2.3) získanou výpočtem metodou nejmenších čtverců. Pro zjištění hodnoty  $K$  v případě nejjednodušší regresní funkce je  $y = Kx$  [6].



Obrázek 2.3: *Lineární regresní funkce*

### Citlivost měřícího členu

Je schopnost přístroje reagovat za stanovených pracovních podmínek na změnu hodnoty měřené veličiny. Stanovené pracovní podmínky jsou dané určitou hodnotou nebo tolerančním polem hodnot ovlivňujících veličinu, jako je např. teplota okolí, tlak, vlhkost.

Citlivost se vyjadřuje výrazem, který je dán podílem změny údaje přístroje  $\Delta y$ , ku změně měřené veličiny  $\Delta x$ , která tuto změnu způsobila [6].

$$c = \frac{\Delta y}{\Delta x} \quad (3)$$

Je-li známa statická charakteristika  $y = f(x)$  je průběh citlivosti v celém rozsahu měřené veličiny dán vztahem:

$$c = \frac{dy}{dx} \quad (4)$$

Vztah platí pouze pro příslušnou hodnotu měřené veličiny, která musí být uvedena a která je počáteční hodnotou uvažované změny měřené veličiny.

V případě nelineární charakteristiky však platí, že pro každý bod charakteristiky není citlivost konstantní, ale je úměrná měřené veličině. Řada přístrojů má nelineární charakteristiku danou rovnicí:

$$y = x_0 + k \cdot x^2, \text{ tedy } c = 2k \cdot x \quad (5)$$

*Prahová citlivost* je nejmenší změna měřené veličiny, která již vyvolá změnu výstupní veličiny. Je vlastně projevem *pohyblivosti* měřícího přístroje.

Převrácená hodnota citlivosti se nazývá *konstanta přístroje*  $K = 1/c$ . V praxi je dána počtem jednotek měřené veličiny na jeden dílek stupnice. Stanovuje se obvykle jako podíl rozsahu stupnice ku počtu dílků. Správnou hodnotu měřené veličiny určíme jako součin počtu dílků  $n$  a konstanty přístroje [1].

### **Rozsah měřícího členu**

Rozsah přístroje udává v jakém rozmezí měřených hodnot je přístroj použitelný. Rozlišuje se rozsah přístroje - *ukazovací*, daný krajními hodnotami měřené veličiny vyznačenými na stupnici a *měřící*, což je ta část stupnice, ve které není údaj přístroje zatížen chybou, než je chyba dovolená. Ta je dána třídou přesnosti přístroje. Z toho vyplývá, že měřící rozsah analogového přístroje je menší než ukazovací. Pro digitální přístroje je měřící i ukazovací rozsah dán nejvyšším zobrazitelným číslem na displeji.

### **Přesnost měřícího členu**

Je schopnost měřící soustavy udávat naměřenou hodnotu měřené veličiny s dovolenou celkovou chybou. Přesnost měřící soustavy je definována krajní chybou  $\Delta_K$ , která je určena trojnásobkem hodnoty směrodatné odchylky  $\sigma$ .

$$\Delta_K = s \cdot \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\Delta y_i)^2} \quad (6)$$

kde: **n** je počet opakovaných měření

$\Delta y_i$  je absolutní chyba

**s** je násobek

*Absolutní chyba* měřicí soustavy  $\Delta y_i$  je rovna rozdílu údaje přístroje  $y_i$  a aritmetického průměru z naměřených hodnot  $y_p$ . Hodnoty aritmetického průměru se stanoví opakovaným měřením pro určitý bod měřícího rozsahu soustavy [1].

$$\Delta y_i = y_i - y_p \quad (7)$$

Relativní chyba je dána poměrem absolutní chyby ku maximální hodnotě měřícího rozsahu.

$$\delta = \frac{\Delta y_i}{y_{\max}} \quad (8)$$

*Třída přesnosti* klasifikuje měřicí soustavu z hlediska chyb přesnosti v celém měřícím rozsahu soustavy. Třída přesnosti (i) se udává poměrem krajní chyb  $\Delta y_{\max}$  ku maximální hodnotě měřícího rozsahu [6].

$$i = \frac{\Delta_K}{y_{\max}} \cdot 100 \quad (9)$$

### 2.2.2. Dynamické vlastnosti

Při měření neelektrických veličin, jejichž hodnota se rychle mění s časem, je nutné znát dynamické vlastnosti měřicí soustavy, abychom obdrželi správnou naměřenou hodnotu [1].

Soustava s nevhodnými dynamickými vlastnostmi nebude měřit správně, protože údaj soustavy se bude opožďovat za změnou neelektrické veličiny. Vztah mezi výstupní veličinou  $y$  a vstupní veličinou  $x$  v přechodovém stavu můžeme obvykle vyjádřit lineární diferenciální rovnicí s konstantními koeficienty.

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 y'(t) + a_0 y(t) = x(t) \quad (10)$$

Kde exponent v závorce znamená řád derivace.

Například termočlánek se chová jako statická soustava 0. řádu. Jeho statické i dynamické chování je popsáno rovnicí  $a_0 \cdot y = x$ . Takový přístroj se z hlediska dynamiky chová ideálně.

#### **Dynamické vlastnosti měřicích přístrojů charakterizuje:**

- jejich dynamická charakteristika,
- čas, za který dosáhne dynamická chyba určité hodnoty

Při vyšetřování dynamických vlastností přístroje se sleduje jeho odezva, tj. závislost údaje přístroje na čase vyvolané skokovou změnou měřené veličiny  $x(t)$ . Průběh výstupní veličiny v závislosti na čase při skokové změně neelektrické veličiny se nazývá přechodová charakteristika. Frekvenční charakteristikou nazýváme průběh harmonické změny neelektrické veličiny v závislosti na kruhové frekvenci. Tyto charakteristiky se mohou určit pouze tehdy, pokud jsou známy příslušné diferenciální rovnice [6].

#### **Přechodová charakteristika**

Většina měřicích soustav má přechodovou charakteristiku popsanou rovnicí prvního řádu jako např.: skleněný rtuťový teploměr. Jeho rovnice vypadá takto.

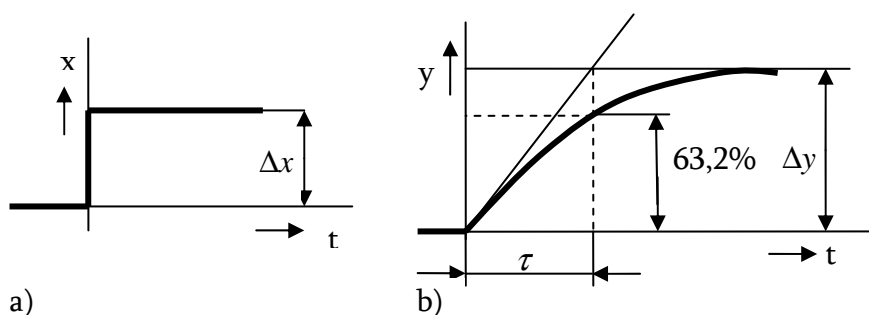
$$a_1 y'(t) + a_0 y(t) = x \quad (11)$$



Přechodová funkce:

$$y = \frac{1}{a_0} \left(1 - e^{-\frac{t}{\tau}}\right) \quad (12)$$

Na obrázku (2.3 a) je nakreslen průběh skokové změny  $\Delta x$  neelektrické veličiny. Velikost tohoto skoku se považuje za jednotku a vyjadřuje se v jednotkách vstupní veličiny, nebo v procentech rozsahu. Na obrázku (2.3 b) je průběh výstupní veličiny  $y$ , kde  $\tau = a_1 / a_0$  je časová konstanta, která udává za jakou dobu dosáhne výstupní veličina 63,2 % z konečné hodnoty při změně vstupní veličiny skokem [1].



Obrázek 2.3: Přechodová charakteristika měřící soustavy.

### Dynamické chyby

Dynamická chyba vyjadřuje rozdíl mezi údajem přístroje a správnou hodnotou měřené veličiny v přechodovém stavu. V ustáleném stavu dynamická chybu vymizí. Je to chyba systematická a můžeme ji udávat jako absolutní dynamickou chybu  $e_d$ . Z toho vyplývá, že dynamické chyby jsou funkcí dynamických vlastností přístroje a času. V jednoduchých případech můžeme časovou závislost vypočítat. U přístroje který je z hlediska dynamiky soustavou 1. řádu vypočítáme dynamickou chybu z odezvy na změnu vstupní měřené veličiny jednotkovým skokem.

Často jsou u přístrojů vyžadovány hodnoty v časech  $t_{50}$ ,  $t_{95}$  a  $t_{99}$  za který jeho údaj dosáhne 50%, 95% a 99 % ustálené hodnoty měřené veličiny, tedy bude mít 50ti 5ti či 1% dynamickou chybu. Znalost dynamických chyb je důležitá při měření rychle se měnících veličin a u měření, která se v pravidelných intervalech opakují [6].

### **2.3. Snímače neelektrických veličin**

Vzhledem k rozsahu práce a k jejímu zaměření, bude v následující kapitole uvedena teorie těch snímačů, které jsou použity v měřicím přístroji. Jedná se o odporový teplotní snímač teploty a o optoelektronické snímače.

#### **2.3.1. Teplotní senzor SMT-160**

Senzor SMT-160 s PWM<sup>1</sup> výstupem pracuje jako převodník *teplota/střída (duty cycle, DUTY)*. Jeho výstupem je TTL signál, jehož šířka pulsu je přímo úměrná teplotě. Zpracování takového signálu bývá jednodušší než zpracování klasické analogové informace (napětí, proud, odpor). Při zpracování výsledků procesorem lze použít vlastní časovací obvody procesoru, zatímco u klasických převodníků je třeba použít další obvody (A/D převodník) [12].

#### **2.3.2. Optoelektronické snímače:**

U těchto detektorů dochází k přímé interakci mezi hmotou a fotony dopadajícího záření. Detektory je možno rozdělit na detektory využívající vnější fotoelektrický jev (fotonky a fotonásobiče) a detektory využívající vnitřní fotoelektrický jev (fotorezistory, fotodiody, fototranzistory apod.).

---

<sup>1</sup> (PWM = pulse-width modulation = pulsně-šířková modulace)

Nejdůležitějšími parametry optoelektronických detektorů jsou:

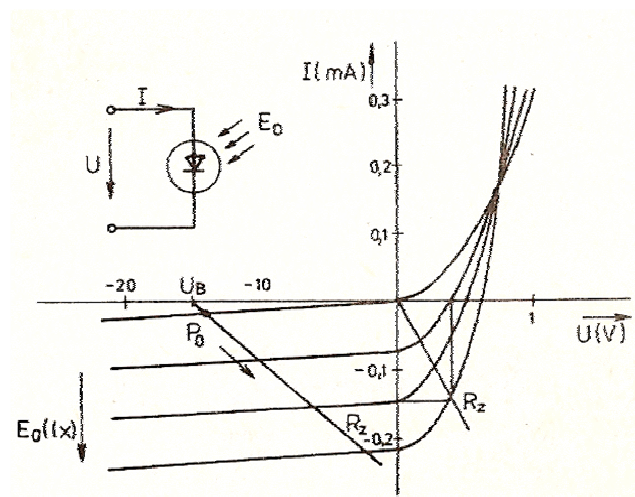
- spektrální citlivost
- šumové vlastnosti
- dynamické vlastnosti
- převodní charakteristika

Při detekci slabých signálů je rozhodující vlastní šum detektoru, který omezuje jeho využitelnost. Udává se jako prahový zářivý tok, nebo také jako minimální detekovatelný výkon [7].

Dynamické vlastnosti detektorů jsou velmi důležité pro většinu aplikací. Udávají se nejčastěji v podobě přechodové charakteristiky, tj. odezvy na jednotkový skok zářivého toku. Prakticky představují rychlost snímače.

Převodní charakteristiky detektorů uvádějí závislost citlivosti na velikosti dopadajícího zářivého toku. Při malých hodnotách toků bývá citlivost konstantní, při větších hodnotách ozáření citlivost zpravidla klesá. U snímačů různých neelektrických veličin se využívá zejména detektorů s vnitřním fotoelektrickým jevem. Do této skupiny patří zejména fotoorezistory, fotodiody a fototranzistory. Jako zdroje záření se využívají též polovodičové lasery.

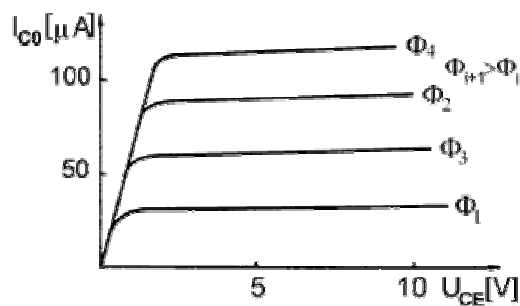
Fotodiody jsou nelineární, nesouměrný jednobran, řízený dopadajícím zářivým tokem. V prvním kvadrantu jsou VA charakteristiky nejhustší, fotodiody jsou zde neméně citlivé na světlo a tato oblast VA charakteristiky diody se nevyužívá. Ve čtvrtém kvadrantu se fotodiody chovají jako zdroj elektrické energie, energie záření se mění přímo na energii elektrickou - jedná se o tzv. hradlový, nebo fotovoltaický režim. Ve třetím kvadrantu jsou charakteristiky rovnoběžné - jedná se o tzv. odporový režim. Fotodiody sloužící k detekci optických signálů mají nastaven pracovní bod do tohoto kvadrantu [7].



Obrázek 2.4: VA charakteristika fotodiody v závislosti na osvětlení

Spektrální charakteristika opět závisí na použitém materiálu. Fotodiody všeobecně mají velkou citlivost, malý proud za tmy, dlouhodobou stálost a jsou poměrně rychlé (do 100 kHz).

Fototranzistory jsou konstrukčně uspořádány tak, aby maximum záření bylo absorbováno v prostoru báze. VA charakteristiky fototranzistoru jsou obdobné jako u normálního tranzistoru, parametrem je zde zářivý tok. Citlivost je větší než u fotodiody, ale rychlost je však poněkud menší [5].



Obrázek 2.5: VA charakteristika fototranzistoru pro různé hodnoty

### **3. Přenos informace**

Existují tři druhy přenosu informací. Prvním z nich je použití metalických datových kabelů, druhý využívá optických vláken a třetí rádiového přenosu [2]. Výhodou prvního připojení je především snazší implementace a realizace přenosu. Použití optických vláken vyžaduje především drahou techniku a používá se hlavně ve velkých datových sítích. Poslední spojení je v dnešní době velice rozšířené, ale přináší s sebou složitější ovládání přenosu. Lze se s ním setkat hlavně tam, kde se preferuje mobilita systému a elegantnost spojení. Následující část práce se bude věnovat prvnímu způsobu přenosu, protože byl použit v popisovaném měřicím zařízení.

#### **3.1. Sériový přenos**

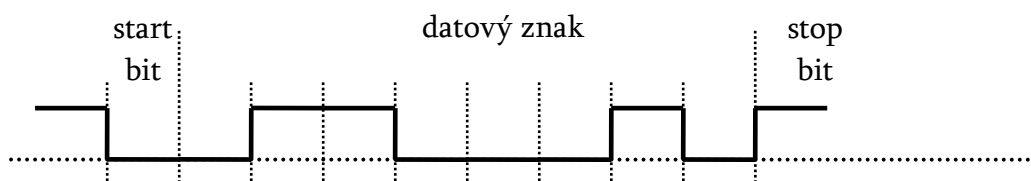
Tato kapitola se zabývá sériovým přenosem, paralelní komunikace se skoro nepoužívá a všechna novější rozhraní, které dnešní přístroje nabízejí, fungují právě na principu sériové komunikace. Sériový přenos využívá v daném čase pouze jediný komunikační kanál. Základní přenášenou entitou je jeden znak, který obsahuje jednotlivé datové bity. Komunikace probíhá způsobem posílání jednotlivých bitů znaku po jednom vodiči. Posílá se postupně jeden bit za druhým jako v jedné lince ( sérii ). [3]

##### **3.1.1. Sériový asynchronní přenos**

Při asynchronním přenosu musí mít vysílací a přijímací strana shodně nastavený formát přenosu dat (počet bitů ) a přenosovou rychlost. Ta se volí z řady 110, 300, 600, 1200, 2400, 4800, 9600 nebo 19200 bit/s. Někdy se uvádí přenosová rychlost v jednotkách  $Bd=1\text{bit/s}$  [3].

Rozpoznání znaku je zabezpečeno úvodním start-bitem a ukončujícím stop-bitem. Znak tedy začíná start-bitem, následuje 5-8 datových bitů, případný paritní

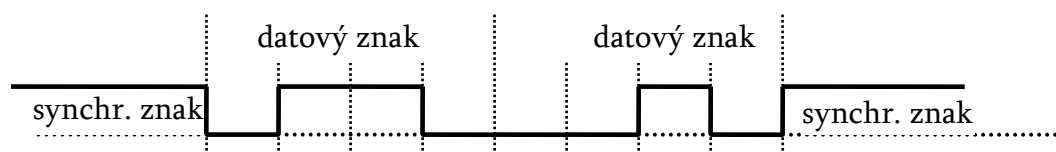
bit a jeden nebo dva stop-bity ( celkem tedy 10 -14 bitů). Nejmenší přenášenou informací je jeden znak. Pokud je mezi znaky delší mezera, je vyplněna stop-bitem. Nejčastěji používaný formát je start-bit, 8 datových bitů a stop-bit. Asynchronní přenos je velmi rozšířen díky své jednoduchosti a možnosti použít minimální počet vodičů [2].



Obrázek 3.1: *Průběh asynchronního přenosu*

### 3.1.2. Sériový synchronní přenos

Při tomto přenosu se synchronizuje celý znak a odpadá tak nutnost posílat společně se znakem start a stop bit. Přenos je zahájen synchronizačním znakem, potom následuje blok dat a ukončen je opět synchronizačním znakem. Tedy jednotlivé znaky dat na sebe navazují bezprostředně za sebou.



Obrázek 3.2: *Průběh synchronního přenosu*

Synchronizační znaky se mohou generovat hardwarově nebo softwarově. Při hardwarovém způsobu se používají generátory synchronizačních znaků a v druhém případě se proces generování znaků nazývá *handshaking*.

Zapojení konektoru je tedy rozšířeno o synchronizační vodič. Výhodou tohoto přenosu je odolnost proti rušení a při stejné rychlosti se přenesou větší počet informací [2].

### **3.2. Standard RS-232**

Sériový port počítače využívá standardu RS-232. Tento standard umožňuje propojení dvou zařízení a následnou komunikaci mezi sebou. Počet datových bitů je volitelný, obvykle se používá 8 bitů, lze se ale také setkat se 7 nebo 9 bity. Každý bit nese informaci v podobě stavu logická „0“ nebo „1“. Tento stav je reprezentován pomocí dvou možných úrovní napětí. V případě stavu log.1 odpovídá napěťová úroveň hodnotám -3 až -15V a ve stavu log.0 se hodnoty úrovní pohybují v rozmezí mezi +3 až +15V. Dále je ve standardu definováno také 20 signálů rozhraní a přiřazení ke konkrétní pozici v konektoru s 25 kontakty. Pro komunikaci s počítačem se ale používá výhradně devítipinového konektoru, v němž je využíváno devět základních signálů[3].

V této práci je použito nejjednoduššího zapojení a to třívodičové, kde jsou využity signály: RxD pro příjem, TxD pro vysílání a společná zem GND. Standard RS-232 dále definuje synchronní a asynchronní sériovou komunikaci pro přenos dat, o níž bylo pojednáno výše [2].

#### **3.2.1. USB port**

Většina dnešních počítačů je vybavena rozhraním USB ( Universal serial bus ), které posílá data také pomocí sériové komunikace. Avšak toto rozhraní bylo vyvinuto tak, aby se k němu mohlo připojit více zařízení na větší vzdálenost

(několik desítek metrů) a také byla zvýšena celková přenosová rychlost. Rozlišují se tyto rychlosti:

- **Low speed** - max. 1.5 Mbit/s
- **Full speed** - max. 12 Mbit/s
- **High speed** - max. 480 Mbit/s

Komunikace probíhá odlišně než tomu bylo u rozhraní RS-232. Počítač nejdříve vyšle tzv. *token paket*, který popisuje typ a směr výměny dat, dále adresu USB zařízení a číslo koncové jednotky. Potom zařízení, které má vysílat data, vyšle datový paket, nebo signalizuje, že žádná data nejsou k dispozici. Příjímá strana nakonec vyšle *handshake* paket, kterým informuje o úspěšnosti přenosu. Existují zde dva typy přenosového modelu - tok dat (stream), využívající izochronní přenos dat v reálném čase, a zpráva (message), využívající asynchronní přenos [2].

### 3.2.2. Převodník USB/RS-232

Výše uvedená rozhraní charakterizuje jedna společná vlastnost a tou je sériový přenos dat. Protože byla jednotlivá rozhraní vyvinuta v odlišném časovém období, vyvstal problém jejich vzájemné kompatibility. Aby starší měřicí technika mohla být využívána i v současnosti, objevují se na trhu převodníky mezi rozhraním USB a RS-232. Tím je zajištěna možnost připojit starší měřicí zařízení komunikující po sériové lince k dnešním počítačům, ačkoli neobsahují sériový port COM. Tyto převodníky jsou běžně dostupné v každém obchodě s výpočetní technikou nebo elektronikou.



## 4. Zpracování dat

Tato kapitola se zabývá teorií zpracování dat v takovém rozsahu který vyžaduje praktická část. Obsahuje kapitoly zabývající se přenosem digitalizovaného signálu, kompresí dat a zobrazením.

### 4.1. Proud dat ( streaming)

Pod pojmem streaming je obecně chápán libovolný přenos digitalizovaného signálu v reálném čase. Přenos v reálném čase znamená, že uživatel na konci přenosového řetězce je schopen přijímat přicházející data průběžně, nikoli až po odeslání celého záznamu [9].

Proudy dat se objevují v mnoha aplikačních oblastech, jako např. provoz Internetu, monitorování a analýza proudů dat získaných pomocí senzorů, (geografická data, medicínská data, meteorologická data), zpracování záznamů transakcí, nebo vývoj cen na burze. Z aplikací vyplývá, že proudy dat lze rozdělit do dvou základních kategorií: *proudy transakční dat* a *proudy naměřených dat*.

Data těchto aplikací, představují nepřetržitý, v čase se měnící proud dat, který je sice možné uložit, v tradičních databázích, ovšem ne vždy uspokojivým způsobem. Vždy se setkáme s proudy dat, které buď není nutné ukládat do databázi, nebo to dokonce není pro jejich rozsah ani fyzicky možné. Tradiční databáze nejsou stavěny na rychle se měnící proudy. Jejich úkolem je data shromažďovat a v daných časových okamžicích vyhodnocovat dotazy nad aktuálním stavem databáze. Získané odpovědi jsou úplné a přesné, nejsou zde dovoleny žádné aproximace, které jsou pro proudy dat často nutné, protože dotazy mohou být *dlouhotrvající* (vyhodnocují se ve velkém časovém úseku) nebo dokonce *nepřetržité* (do dotazu vstupuje nepřetržitě proud dat) [8].

Mezi novými ITC technologiemi, zaujímá v kontextu nových požadavků na databáze, čelné místo levná mikrosenzorová technika, která umožňuje většině senzorů podávat zprávy ve formě proudu dat v reálném čase. Zprávy obsahují informace o hodnotách jejich atributů, což jsou např. fyzikální veličiny jako je teplota, tlak, nebo pozice umístění (např. pomocí globálního pozičního systému).

Pojem senzor je v problematice proudu dat chápán jako zařízení s vlastním energetickým zdrojem poskytující nepřetržité, možná nekonečné proudy dat. Sensory mohou být instruovány dělat periodická měření a provádět předzpracování dat [8]. Mezi jejich typické úkoly patří operace

- pošli data nebo kombinaci dat (průměr, součet apod.) do nějakého vybraného uzlu,
- ulož data (která mohou být zpracována později v čase),
- pošli zprávu jako reakci na nějakou pozorovanou událost.

Výhodou proudu dat je, že čtení je možné začít již několik sekund po navázání spojení. Nevýhodou naopak je požadavek na stabilní propustnost komunikačního protokolu, aplikace klienta si sice může uchovávat data na několik sekund dopředu, tím se ale zvyšuje celkové zpoždění přenosu [8]. Technicky je přenos zajišťován třemi články:

**Encoder:** Jedná se o mikroprocesor (počítač), který signál převádí do podoby akceptovatelné serveru.

**Server:** Poskytuje kontinuální zdroj dat spolu se synchronizací jednotlivých složek (audio, video). V případě multibitrate vysílání (vysílání na více rychlostech) zodpovídá také za vybrání patřičného proudu pro danou šířku pásma. Server může vysílat ze dvou médií - proud dat z

encoderu (používá se při on-line vysílání), nebo soubor uložený na serveru (při módu Video on Demand).

**Klient:** Je software, který je spuštěn u uživatele. Jeho účelem je přijímat data ze sítě a vytvářet z nich výslednou podobu pro zobrazení [9].

## 4.2. Kompresie dat

Kompresie dat je speciální postup při ukládání nebo transportu dat. Úkolem komprese je zmenšit datový tok nebo zmenšit potřebu zdrojů při ukládání informací. Obecně se jedná o snahu zmenšit velikost datových souborů, což je výhodné např. pro jejich archivaci nebo při přenosu přes síť s omezenou rychlostí (snížení doby nutné pro přenos). Kompresie může být nutná při omezené datové propustnosti, např. mobilní telefon komprimuje hovor pro přenos GSM sítí [14].

Zvláštními postupy ( kódováním ), které je dané zvoleným kompresním algoritmem, se ze souboru odstraňují redundantní (nadbytečné) informace, snižuje se tím tak entropie dat<sup>2</sup>. Kompresy dat lze rozdělit do dvou základních kategorií:

*Ztrátová komprese* při kompresi jsou některé informace nenávratně ztraceny a nelze je zpět rekonstruovat. Používá se tam, kde je možné ztrátu některých informací tolerovat a kde nevýhoda určitého zkreslení je bohatě vyvážena velmi významným zmenšením souboru. Používá se pro kompresi zvuku a obrazu (videa), při jejichž vnímání si člověk chybějících údajů nevšimne nebo si je dokáže domyslet (do určité míry).

*Bezeztrátová komprese* obvykle není tak účinná jako ztrátová komprese dat. Velkou výhodou je, že komprimovaný soubor lze opačným postupem

---

<sup>2</sup> je obecně veličina udávající míru neuspořádanosti zkoumaného systému nebo také míru neurčitosti daného procesu.

rekonstruovat do původní podoby. To je nutná podmínka při přenášení počítačových dat, výsledků měření, textu apod., kde by ztráta i jediného znaku mohla znamenat nenávratné poškození souboru [14].

*Kompresní poměr* je podíl velikosti původních dat ku velikosti komprimovaných dat. Například při kompresi 10MB souboru do 2MB je poměr  $10/2 = 5$ . Kompresní poměr je ovlivněn volbou kompresního algoritmu i typem komprimovaných dat. **Úspora místa** se vyjadřuje jako opačný poměr, například  $1 - 2/10 = 0,8$  (tj. 80% úspora). Například nekomprimované skladby na Audio CD mají datový tok přibližně 1,35 Mb/s, zatímco komprimované zvukové soubory mají typicky 128 Kb/s. Kompresní poměr je tedy asi 11:1, a úspora datového toku přibližně 90 %. Jedná se pochopitelně o ztrátovou kompresi, pro bezztrátovou kompresi jsou (pro stejný typ dat) typické poměry do 2 : 1. V případě této komprese nelze dosáhnout vyššího poměru, než je míra entropie komprimovaných dat [14].

#### **4.2.1. Bezeztrátová komprese**

Bezeztrátová komprese je jeden ze dvou základních přístupů ke kompresi dat. Jedná se o algoritmy, které dovolují přesnou zpětnou rekonstrukci komprimovaných dat, na rozdíl od ztrátové komprese, kde to možné není. Používá se všude tam, kde je důležité, aby originální data a data po dekompresi komprimovaného souboru byla totožná - např. komprese textů nebo komprese čehokoli, kde je nepřijatelná i sebemenší ztráta kvality.

Existuje mnoho různých formátů, které využívají bezztrátové komprese. Například velmi populární ZIP, se kterým je možné pracovat téměř na jakékoli platformě. Dalším velmi oblíbeným formátem je RAR, tento formát je však komerční a neexistují nástroje pro vytváření archivů RAR na jiné platformě než na platformě MS Windows. Mezi opensourcovými komunitami jsou oblíbeny

formáty gzip a bzip2, jejichž algoritmy mají tu výhodu, že nejsou patentované. Dalším méně známým, ale velice schopným kompresním algoritmem je LZMA, který je užíván programem 7-Zip [14].

### 4.3. Grafické znázornění průběhu dat.

Měřili-se funkční závislost fyzikálních veličin, využívá-se pro přehlednější zpracování této funkční závislosti grafického vyjádření. Uvažované fyzikální veličiny označíme  $x$ ,  $y$  a jejich neznámou funkční závislost  $y = f(x)$ . Při měření se získá  $n$  dvojic odpovídající hodnotám  $[x_1, y_1], [x_2, y_2] \dots [x_n, y_n]$ .

Ke grafickému znázornění se nejčastěji užívá pravoúhlý souřadnicový systém, ale existují i jiné, například polární nebo prostorový. V této práci je používán pouze pravoúhlý (kartézský) v němž každé dvojici odpovídá jeden bod grafu a spojením těchto bodů od 1 do  $n$  vznikne požadovaný průběh [4].

Pro prezentaci výsledného grafu v digitální podobě se využívá softwaru, který celý proces generování zajišťuje. Software může být napsán v libovolném programovacím jazyce a jeho výstupem je požadovaný průběh naměřených hodnot reprezentovaný vhodným grafickým objektem. Podle požadavků prostředí ve kterém se bude výstupní graf zobrazovat volíme druh programovacího jazyka. Požadavky této práce určily volbu některého ze skriptovacích jazyků umožňující interaktivní spolupráci s sítí internet. Byl zvolen jazyk PHP, což je univerzální skriptovací jazyk fungující nezávisle na použité platformě. Skripty jsou prováděny na straně serveru a k uživateli se dostane jen výsledek. Pro kreslení základních grafických objektů poskytuje jazyk knihovnu *GD* ale pro komplexnější řešení je dobré využít dalších rozšíření [4].

### 4.3.1. Popis knihovny JpGraph

JpGraph je knihovna pro programovací jazyk PHP (4.0.2 a vyšší) obsahující sadu tříd, které umožňují generování různých typů grafů za použití co nejmenšího množství kódu. Pomocí JpGraph je možné vytvářet velké množství různorodých grafů, od základních sloupcových, bodových, křivkových, koláčových až po polární, tří-dimenzionální, mapové nebo obrázkové. Knihovna také umožňuje zvolit různé druhy stupnic na souřadnicových osách, na výběr jsou lineární, logaritmické, celočíselné nebo vlastní [11].

Podporována je i technologie CSIM (Client Side Image Maps), dovolující definovat aktivní místa uvnitř grafu, které lze využívat, prostřednictvím odkazů, k popisování vybraných údajů. Kromě vlastního *cache* systému, který dokáže redukovat počet přístupů na server, obsahuje také podporu pro grafické vyhlazování a definování vlastního písma [13].

#### Práce s knihovnou.

JpGraph je založen na funkcích dostupných v prostředí jazyka PHP za pomoci grafické knihovny GD. V závislosti na použité verzi knihovny jsou podporovány následující grafické formáty JPEG, PNG a GIF. Architektura knihovny je podrobněji znázorněna na obrázku (4.1). Místo pevně dané funkcionality bylo vytvořeno plně rozšiřitelného návrhu a tím je knihovna rozdělena na jádro a rozšiřitelné části. Jádro představuje základ pro samotné kreslení a zajišťuje celý běh knihovny včetně chybových hlášení. Jednotlivé metody rozšiřitelných částí potom představují různé typy grafů.. Jednotlivá rozšíření jsou pak pojmenována tak, aby jejich název vystihoval typ grafu, který zastupují. Nappříklad *jpgraph\_bar.php* reprezentuje modul pro kreslení sloupcových grafů. Právě tato modularita celého systému napomáhá k snížení délky celého kódu.

Další charakteristikou knihovny je poskytnout práci v kompletně objektově orientované struktuře a tím si udržet konzistenci API rozhraní mezi rozdílnými grafickými objekty. Často se používá metoda, která nastavuje všem grafickým objektům jejich textové vlastnosti. Například metoda *SetFont()* nastavuje styl fontu na daném objektu [13].

V JpGraph knihovně obsahují objekty mnoho vlastností, které dále upravují zase své další parametry. Vytváří se tím tak nepohodlně dlouhé volání, které vypadá například takto: *\$graph->xaxis->scale->ticks->Set(...);*

Řešením je volání metod nastavující jednotlivé vlastnosti objektů a nepřístupovat tak k vlastnostem přímo. Tím je myšleno například následující volání , které plní stejnou funkci jako předchozí zápis [13].

*\$graph->xaxis->SetTicks(...);*

## 5. Měřicí zařízení

Předmětem této kapitoly bude popis funkce měřícího přístroje, který slouží k demonstraci měření neelektrických veličin. Přístroj napodobuje chování anemometru a díky tomu je pochopení celého principu měření snazší. Jak bylo uvedeno již v úvodu přístroj byl vyroben před pěti lety a byl sestaven z materiálů, které měl autor v té době k dispozici. Proto je celá konstrukce přístroje určena pouze pro názornou ukázkou funkčnosti a o nasazení do reálného prostředí autor neuvažuje.

V průběhu psaní této práce byly zjištěny nedostatky v oblasti vypočítávání, vyhodnocování a přenosu dat. Měřicí zařízení také neobsahuje všechny členy blokového schématu standardní měřící soustavy, jak bylo uvedeno v teoretické části. viz obrázek (2.1). Tyto problémy jsou dané dobou vývoje celého projektu a cílem této práce tedy není jejich náprava, ale využití stávajícího přístroje k popisu měření neelektrických veličin. Autor proto nezasahoval do opravy výše zmíněných nedostatků, ale vzal celý přístroj v takovém stavu, v jakém byl vyroben.



Obrázek 5.1: *Měřicí jednotka*



Obrázek 5.2: *Řídící jednotka*



## 5.1. Popis měřícího zařízení

Celý přístroj se skládá ze dvou částí. První část je měřící jednotka viz obrázek(5.2), která měří povětrnostní vlivy podnebí a teplotu vzduchu. Druhá část, *řídící jednotka*, slouží k sběru digitálních signálů, které vyhodnocuje a dále zobrazuje na LCD panelu. Jednotka také odesílá data do počítače k dalšímu zpracování.

## 5.2. Řídící jednotka

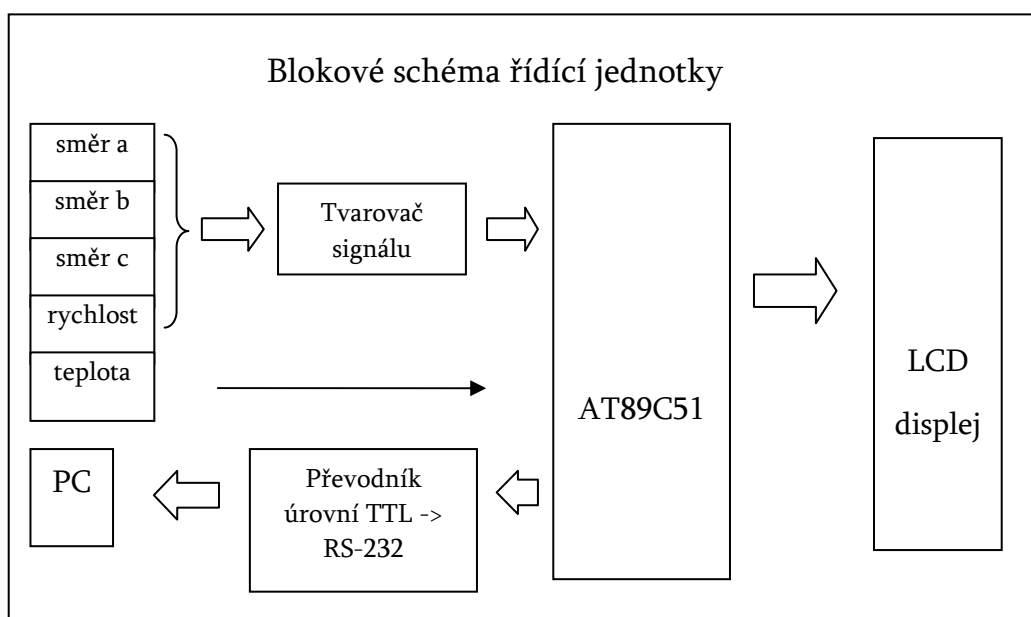
Srdcem řídící jednotky je mikroprocesor od firmy Atmel AT89C51, který obsahuje čtyři vstupně-výstupní porty. Přestože procesor již není nejmodernější, pro účel práce plně postačuje, a autor s ním měl v předchozích letech dobré zkušenosti. Obsahuje řadu přerušovacích vstupů, vstupy vnějších čítačů a vnitřní časovače. V neposlední řadě disponuje dostatečnou pamětí pro obslužný program a podporou pro sériovou komunikaci.

Vstupními daty procesoru jsou signály z měřící jednotky, které mají podobu číslicového výstupu, velice podobného hodinovému signálu. Proto jsou připojeny přímo na piny jednotlivých portů mikroprocesoru, kde jsou snadno měřitelné. Pokud by přístroj poskytoval signál s analogovým výstupem, musela by řídící jednotka obsahovat ještě A/D převodník.

Protože se výstupní hodinový signál z měřící jednotky přenáší po datovém kabelu, dlouhém čtyři metry, dochází k jeho zkreslení a zeslabení. Proto je důležité před zpracováním signál zesílit a vytvarovat ho tak, aby byl procesorem snadno měřitelný. K tomuto účelu slouží Schmittův klopný obvod 74HC14, na jehož vstupu je jeden signál ze snímače rychlosti větru a tři signály ze snímače směru větru. Poslední signál ze snímače teploty nebylo potřeba tvarovat ani zesilovat.

Velkou výhodou mikroprocesoru je tedy podpora sériového kanálu. Jedná se o plně duplexní kanál umožňující pracovat v asynchronním nebo synchronním režimu. Procesor je také vybaven registry, které nastavují způsob přenášení dat sériové lince, a sériovým bufferem, z kterého můžeme číst nebo zapisovat znaky. Protože všechny integrované obvody pracují s logickými úrovněmi TTL, je důležité pro správnou komunikaci po lince převést tyto úrovně na hodnoty rozhraní RS-232. Celý převod zajišťuje integrovaný obvod MAX232, který obsahuje dva vstupy pro převod z rozhraní RS-232 do TTL a dva vstupy pro převod z rozhraní TTL do RS-232.

K řídicí jednotce je také připojený displej, který slouží pro zobrazení naměřených dat. Autor využil alfanumerický display 2 x 16 znaků od firmy HITACHI HD44780. Na každém řádku lze tedy zobrazit 16 znaků. Každý znak je složen z 35 bodů organizovaných do matice 5x7, takže se můžou zobrazovat i vlastní znaky, jako jsou ikony apod. Pomocí vestavěného řadiče displeje se mohou měnit různé vlastnosti zobrazování, například velikost písma, font písma, pohyb kurzoru, posun textu vpravo či vlevo.

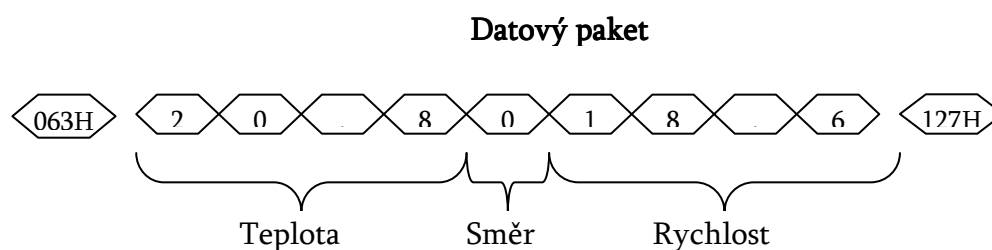


Obrázek 5.3: *Blokové schéma řídicí jednotky*

Změřené a zpracované veličiny posílá řídicí jednotka do počítače přes rozhraní RS-232, kde se využívá 8bitový UART režim (Universal Asynchronous Receiver-Transmitter).

### Rozbor datového paketu

Posílá se vždy sekvence devíti znaků jdoucích hned po sobě, přičemž začátek této sekvence je uvozen znakem 63H a konec znakem 127H. Tyto znaky slouží pro snadné rozpoznání celé věty, kterou dále zpracovává počítač. Podle obrázku (4.3) je vidět, že se nejdříve posílá informace o teplotě, potom následuje jeden znak směru větru a poslední čtyři znaky patří rychlosti.



Obrázek 5.4: *Datový paket přenášených dat*

V případě teploty a rychlosti představují jednotlivé čtyřčlenné sekvence znaků reálnou hodnotu neelektrické veličiny ale informace o směr větru je kódována hodnotou v intervalu 0 až 8 z čehož není patrné o jaký směr se jedná. Proto musí dojít k převodu hodnoty na konkrétní údaj, prostřednictvím pravidel vyplívajících z tabulky (5.2). Podle obrázku tedy odpovídá sekvence znaků teploty hodnotě 20.8, znak směru kóduje hodnotu 0 (0 je ekvivalentní hodnota pro severní světovou stranu) a rychlost hodnotu 18.6. O napájení řídicí jednotky se stará stabilizované napětí 5V z externího adaptéru.

### 5.3. Měřicí jednotka

Druhá část přístroje má za úkol měřit následující veličiny: směr a rychlost větru a teplotu vzduchu. Všechny tři snímače jsou zabudovány v jedné konstrukci a snímané hodnoty jsou vedeny osmi-žilovým kabelem do řídicí jednotky.

Za měřicí soustavu je považován celý přístroj, který se skládá ze třech měřících členů. Každý člen (senzor) obsahuje čidla, která jsou v přímém styku s měřenou veličinou. Dále následuje popis principu fungování jednotlivých členů měřicí soustavy.

#### 5.3.1. Teplotní senzor

Pro snímání teploty vzduchu byl použit senzor od firmy Smartec SMT160-30. S tímto čidlem měl autor již menší zkušenosti a také jeho cena byla v té době velice přijatelná. Vzhledem k tomu, že celá koncepce zařízení je určena pro experimenty a pokusy ve vnitřních prostorech, bylo vybráno pouzdro TO-92 housing, které je určeno právě do vnitřního prostředí. Pro účely testování funkčnosti i mimo místnost, bylo nutné zapouzdřit celý senzor do kovové schránky a následně jej zalít silikonovým tmelem. Celý senzor tvoří tři kontakty, dva pro napájení a jeden pro výstupní signál, jehož střídu udává vnitřní převodník teplota/střída.

#### Vlastnosti senzoru podle katalogového listu:

- napájecí napětí 4,75 až 7,2 V
- teplotní rozsah (-45 až 130) °C,
- celková absolutní chyba  $\pm 0,7$  °C,
- přesnost lepší než 0,05 °C, dosažitelné rozlišení závisí na přesnosti měření střídy výstupního signálu.

Formát výstupního signálu je hodinový a nikoli číslicový, a proto se musí ještě měřit a vypočítávat z něj velikost střídavy. Měření je zajištěno jedním ze vstupů procesoru, který měří délku úrovně log“1“ ku celkové periodě T. Výsledná neelektrická veličina je dána vztahem:

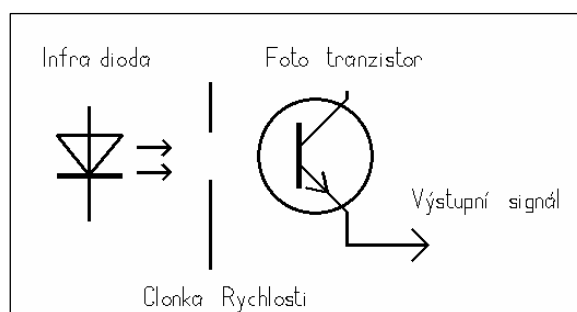
$$t = \frac{D/C - 0,32}{0,0047} \quad (13)$$

$D/C$  je poměr úrovně log“1“ k délce periody T.

$t$  je výsledná teplota ve stupních.

### 5.3.2. Senzor rychlosti

Činnost senzoru je založena na principu fotočlánku. Ten je tvořen fototranzistorem a ozařovací infračervenou diodou. Výstupní signál je odebrán z kolektoru fototranzistoru a je dále veden do vyhodnocovacího členu řídicí jednotky.



Obrázek 5.5: Fotočlánek v senzoru rychlosti

#### Popis principu činnosti

Mechanická konstrukce se skládá z hřídelky, která je uložena ve valivých ložiscích, a ty jsou zapouzdřeny v silonovém náboji. Na jednom konci hřídelky jsou upevněny lopatky a na druhém je upevněna clonka ve tvaru kolečka. Podle toho jestli vítr pohybuje s lopatkami, clonka propouští nebo nepropouští záření na fototranzistor, a ten buď spíná nebo rozpíná obvod stejnosměrného napětí 5V.

Fotočlánek tedy generuje signál, jehož stavy se opakují s proměnnou frekvencí. V sepnutém stavu představuje fototranzistor hodnotu úrovně signálu v log „0“ a v rozepnutém stavu hodnotu v úrovni log „1“ . Takto vytvořený signál je velice podobný hodinovému a měří se počet jednotlivých stavů, tj. impulsů nikoli frekvence.

### Ocejchování senzoru

Pro správné určení hodnoty měřené veličiny musí být senzor ocejchován. Proces cejchování vyžaduje dvě měřící soustavy, první soustava musí být přesně kalibrována a druhá je ta, která se cejchuje. Na obě soustavy se nechá působit měřená neelektrická veličina a při změnách této veličiny se zaznamenává aktuální hodnota výstupního signálu cejchované soustavy k odpovídající hodnotě kalibrované soustavy.

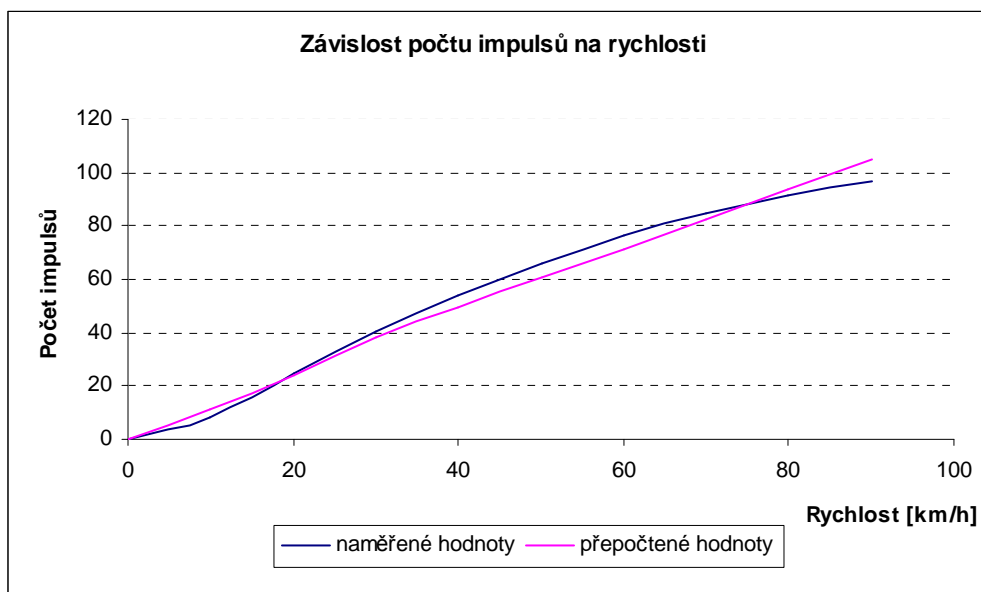
V případě cejchování senzoru rychlosti byl za kalibrovanou soustavu zvolen automobil a měřenou neelektrickou veličinou bylo proudění vzduchu okolo automobilu. Následně byla provedena série jízd při různých rychlostech a zaznamenány odpovídající údaje. Výsledkem celé činnosti je tabulka (4.1) naměřených hodnot a následné stanovení vlastností senzoru.

*Tabulka 5.1: Naměřené hodnoty*

Počet Impulsů	0	10	20	30	40	50	60	70	80	90
Rychlost [ km/h]	0	8	25	41	54	66	76	85	92	97

Na základě tabulky byl vytvořen graf, ze kterého pomocí lineární regresní metody vznikla rovnice, představující vztah pro výpočet výsledné neelektrické veličiny ( A je počet impulsů).

$$Y = -11 + 1,98 * A - 0,0087A^2 \quad (14)$$



Obrázek 5.6: Graf počtu impulsů a rychlosti

### Vlastnosti senzoru

- napájecí napětí 4,75 až 7,2 V
- rozsah hodnot (0 - 99.9) km/h
- celková absolutní chyba ± 15,7 km/h

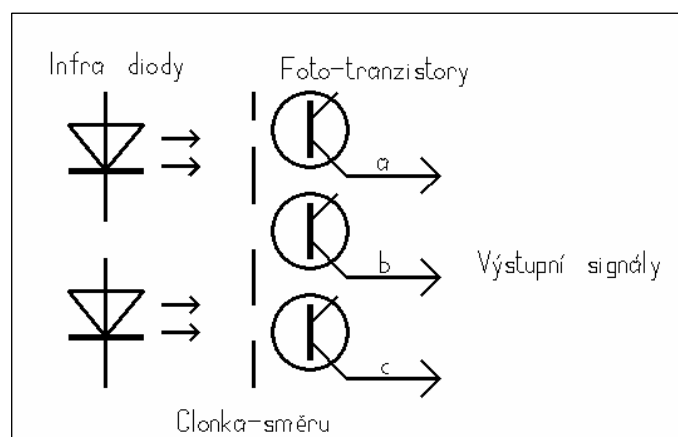
Kvůli použité metodě cejchování a také konstrukci snímače bylo dosaženo hrubých hodnot vlastností senzoru. Mezi vlivy, které způsobily tyto nedostatky, patří celková nepřesnost tachometru automobilu, nesouměrné proudění vzduchu a špatný návrh konstrukce snímače. Avšak v době vývoje senzoru neměl autor patřičné vybavení, aby mohlo být provedeno kvalitnější cejchování. Vzhledem

k tomu, že se práce nezabývá konstrukcí ani výrobou senzorů, může se pro orientační měření sensor s těmito vlastnostmi použít.

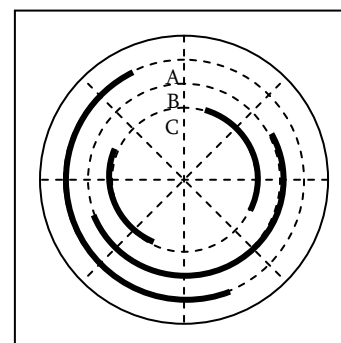
### 5.3.3. Optický sensor polohy

Stejně jako v případě senzoru rychlosti je i zde využíváno fotočlátku, který obsahuje tři fototranzistory a dvě ozařovací infračervené diody. Sensor rozlišuje celkem osm směrů s krokem 45°, výsledný úhel natočení představuje tří-bitové číslo v binární soustavě, které je v řídicí jednotce následně přeloženo na jeden z osmi základních směrů. Sensor funguje tedy jako kódovač osmi stavů do tří-bitového čísla.

Principem činnosti je pohyb vějíčky, která na společné hřídeli s clonkou propouští světelný paprsek na fotosnímač. Výsledná hodnota veličiny je určena třemi signály, které nesou danou informaci. Návrh clonky je znázorněn na obrázku (5.8), kde jsou černě zobrazeny vyfrézované otvory.



Obrázek 5.7: fotočlánek senzoru



Obrázek 5.8: Návrh clonky



### Počáteční nastavení senzoru

K počátečnímu nastavení senzoru bylo použito přesného kompasu, který se natočil tak, aby určoval některou ze světových stran (např. sever). Potom se vedle kompasu umístil nastavovaný senzor, kterému byla natočena vějíčka, aby s ní spojená clonka propouštěla světlo na fotosnímače podle tabulky (5.2). Severní světové straně odpovídá úhel natočení vějíčky  $0^\circ$  a výstupní hodnota senzoru je 000b. Tímto procesem byla nastavena výchozí poloha senzoru, který byl následně zafixován v konstrukci přístroje.

### Vlastnosti senzoru

- napájecí napětí 4,75 až 7,2 V
- rozlišovací schopnost 8 pozic
- krok  $45^\circ$
- rozsah (0 až  $360^\circ$ )
- celková absolutní chyba  $\pm 11^\circ$

Výslednou celkovou chybu ovlivňuje hlavně kvalita provedení clonky, kde záleží na přesném vyfrézování otvoru propouštějící záření na fotosnímač.

Tabulka 5.2: *Tabulka tří-bitového kódovače*

úhel [°]	0	45	90	135	180	225	270	315
signál A	0	0	0	0	1	1	1	1
signál B	0	0	1	1	1	1	0	0
signál C	0	1	1	0	0	1	1	0

## **6. Software řídicí jednotky**

Pro správnou funkci celého přístroje je nezbytně nutný software, který řídí všechny obvody a provádí výpočet ze změřených dat. Software tak dává hardwaru jeho funkci a zprostředkovává komunikaci mezi uživatelem a přístrojem. Software v řídicí jednotce je uložen v paměti v mikroprocesoru a je napsán v jazyce assembler. Účelem této kapitoly je pomocí vývojových diagramů principiálně naznačit fungování hlavních částí programu (způsob vyhodnocování signálů).

### **6.1. Deklarace proměnných**

První částí programu je deklarace paměťových míst proměnných, které jsou v programu využívány, a také deklarace obslužných rutin vnějších přerušení a časovačů. Po deklaraci rutin následuje jejich implementace. V programu se dále využívá dvou časovačů TR0 a TR1. První z nich je určen pro měření rychlosti a druhý slouží k sériové komunikaci. Dále je použito vnější přerušení EX0 a jedno přerušení od sériového kanálu .

### **6.2. Inicializace**

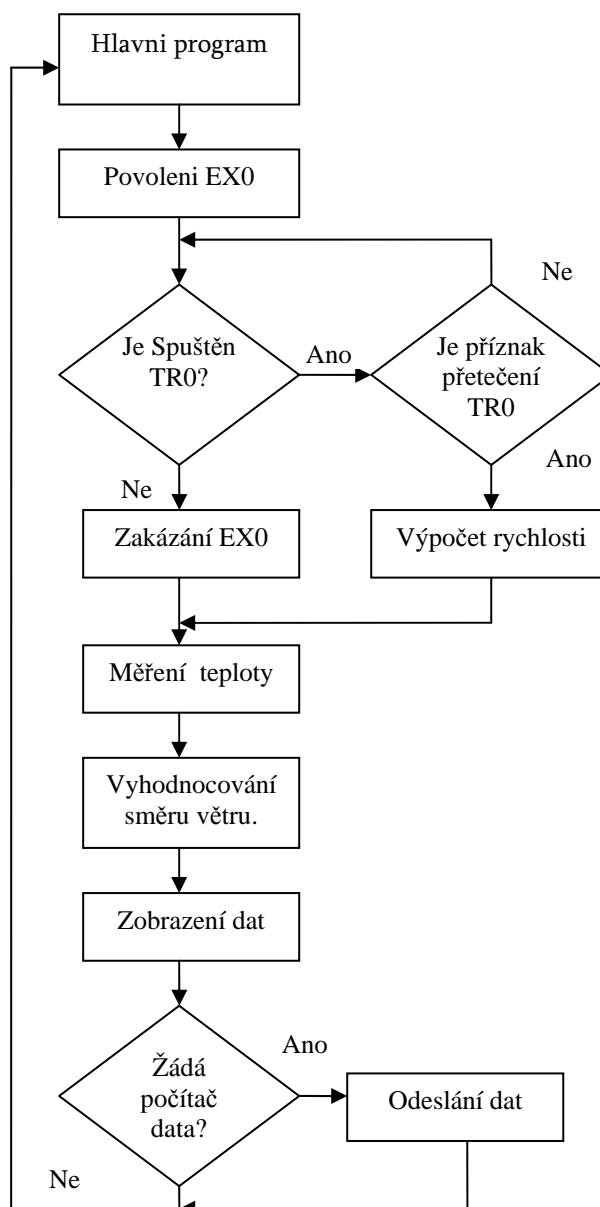
V této části se nastavují registry pro správnou činnost přerušení, časovačů a sériové komunikace. Také se nuluje paměť a zároveň se nastavují počáteční hodnoty proměnných. Dále se volají uživatelské podprogramy pro inicializaci periférií, například LCD displej, obvod reálného času nebo externí paměť. Celá inicializace se provede pouze jednou a to hned při spuštění.

### **6.3. Hlavní běh programu**

Načítání dat ze senzorů probíhá v hlavní programové smyčce, která se neustále opakuje a díky tomu jsou zobrazovaná data na LCD displeji stále aktuální. Celá činnost smyčky je znázorněna na obrázku (5.3).

V prvním kroku se provádí testování časovače TR0, který mohl být spuštěn vnějším přerušením (EX0) od senzoru rychlosti. Proto je hned na začátku toto přerušení povoleno. Pokud je senzor v činnosti, začne se provádět samotné měření, které trvá 2s. Po uplynutí této doby se vnější přerušení zakáže a provádí se výpočet rychlosti. Pokud senzor v činnosti nebyl, dosadí se za výsledek nulové hodnoty a opět se přerušení EX0 zakáže.

Dalším krokem je měření teploty, jehož doba je přesně stanovena na dobu jedné sekundy. Po tuto dobu musí být zakázána všechna přerušení, která by mohla narušit průběh měření. Po skončení měření jsou všechna přerušení opět povolena, až na vnější přerušení EX0, které by mohlo způsobovat komplikace v dalších částech programu. Následně se provede výpočet teploty. Pro určení směru větru nejsou vyžadována žádná měření ani výpočty, pouze se provádí vyhodnocení aktuálního stavu senzoru z kterého se získá požadovaná hodnota. Před



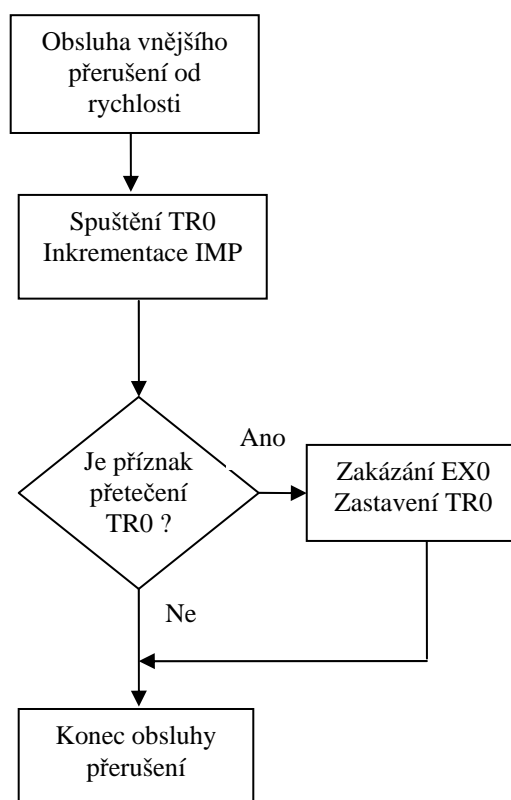
Obrázek 6.1: *Vývojový diagram hlavní programové smyčky*

koncem programu jsou získaná data odeslána zobrazovací jednotce a v závěrečné fázi probíhá kontrola příznaku *žádosti o data* do počítače. Pokud je příznak nastaven, odešlou se data po sériové lince do počítače. Celá smyčka je tímto u konce a začíná se opět prvním krokem.

## 6.4. Důležité části programu

### Obsluha vnějšího přerušení od senzoru rychlosti

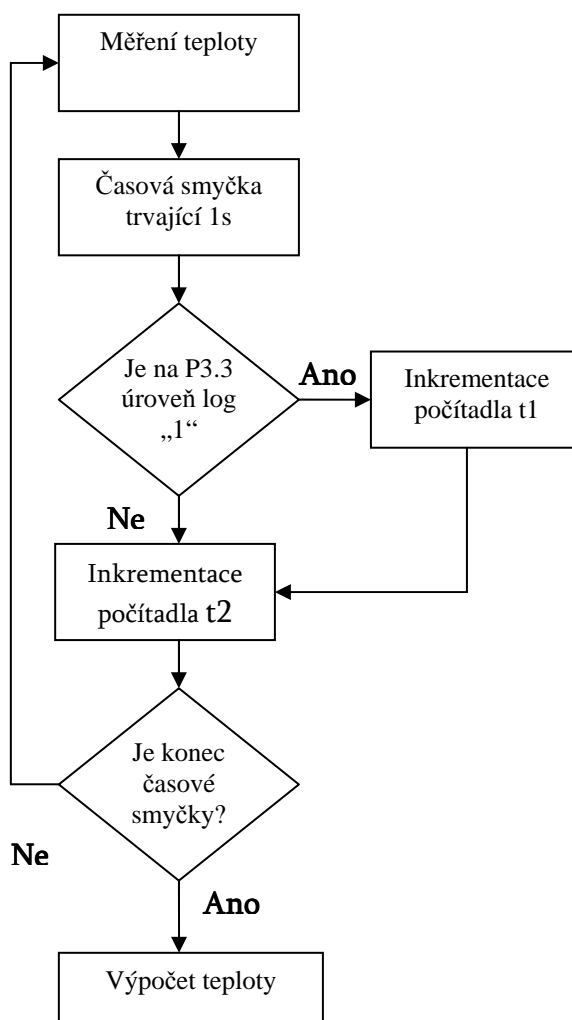
Jako první se měří data ze senzoru rychlosti, jehož výstupní signál je přiveden na pin vnějšího přerušení procesoru (bit P3.2). Pokaždé, když se na tomto bitu objeví sestupná hrana signálu, vykoná se obsluha přerušení, která inkrementuje počítadlo impulsů IMP, a zároveň spouští časovač TR0, který udává dobu měření. Po uplynutí doby měření se zakáže další přerušení a provádí se výpočet podle vzorce (14). Neznámý člen  $A$  ve vzorci představuje právě změřený počet impulsů. Celkovou dobu, po kterou se měří rychlost, určuje předem nastavený časovač. Tato doba musí být rovna době, která byla použita při cejchování senzoru, jinak by vypočtený výsledek nebyl správný. Vnější přerušení je povoleno zase až na začátku hlavní smyčky.



Obrázek 6.2: Vývojový diagram obsluhy vnějšího přerušení

## Měření teploty

Pro měření teploty je nutné vypočítat střidu signálu. Nejdříve je zapotřebí zjistit velikost doby  $t_1$ , kdy se měřený signál nachází ve stavu úrovně  $\log\text{"1"}$ , a doby  $t_2$ , kdy je měřený signál ve stavu  $\log\text{"0"}$ . Střida signálu je potom dána poměrem času  $t_1$  ku součtu časů  $t_1$  a  $t_2$  (celkové době). Pro celkovou dobu měření nelze použít další časovač, protože je používán pro výpočet rychlosti. Časování tedy zajišťuje cyklus s předem danou délkou trvání. V těle cyklu se zjišťuje stav pinu procesoru P3.3, na který je přiveden měřený signál. Podle stavu se inkrementuje počítadlo doby  $t_1$  nebo  $t_2$ . Důležité při tomto měření je fakt, že celková délka cyklu není stejná jako součet dob  $t_1$  a  $t_2$ . Pokud by tento předpoklad nebyl dodržen, nebylo by měření přesné a výsledná teplota by nebyla správná. Výpočet teploty se provádí podle vzorce (13).



Obrázek 6.3: Vývojový diagram měření teploty

## Vyhodnocování směru větru

Získání informace o směru větru probíhá na základě vyhodnocování třech signálů, které jsou přivedeny na port P1. Hodnota portu (8bitů) se načte do pomocné proměnné, z níž jsou důležité pouze první tři bity. Pomocí bitových operací se získá tříbitové binární číslo, ze kterého se za použití převodní tabulky (5.2) určí světová strana. Výsledná hodnota se porovná se starou hodnotou, a pokud se shodují, ponechá se původní údaj. V opačném případě je starý údaj nahrazen novým. Takto vyhodnocený údaj musí projít převodem, který byl pospán výše, a který se provádí až na straně obslužné aplikace v počítači. Je to z důvodu snazší implementace kódu a menšímu počtu znaků v posílaném datovém paketu.

## Posílání a zobrazování dat

Po změření a vykonání všech výpočtů se odesílají hodnoty po sériové lince a zobrazují se na displeji. Posílání hodnot do počítače se provádí vždy po dokončení měření všech veličin. Datový paket, který se posílá do počítače, je ve tvaru znázorněném na obrázku (5.3). Neobsahuje žádné kontrolní součty a jeho přenos není zabezpečen, protože pro účely práce nebylo třeba dodržet tyto požadavky. Je pouze uvozen zahajovacím a ukončovacím znakem, aby bylo možné rozpoznat konec a začátek celé sekvence. Formát dat je téměř identický s formátem zobrazovaných dat na displeji. Liší se pouze v tom, že se hodnoty změřených veličin posílají bez jednotky, a hodnota směru větru je reprezentována číslem, nikoli písmenem.

Proces posílání dat v hlavní programové smyčce připomíná problematiku streamingu sensorových dat. V tomto případě je za *encoder* považována řídicí jednotka, která převádí informace z měřících členů na signál srozumitelný serveru (počítači). Počítač je tedy chápán jako server, vysílající proud dat z encodéru

přímo klientovi, který data přijímá a vytváří z nich konečnou podobu. Za klienta se může považovat obslužná aplikace meteoK3, která zobrazuje uživateli výsledné hodnoty.

V případě že by server nebyl k dispozici, přejímá jeho činnost *encoder* (řídící jednotka), která funguje zároveň i jako server. Klientem by v takovém případě byl podprogram obsluhující zobrazovací modul řídící jednotky. Hodnoty se zobrazují v následujícím sledu: název veličiny, hodnota a jednotka. Grafický výstup tedy vypadá následovně:

*Teplota 25.6 °C,*

*Rychlost 7.8 km/h Směr JV*

### **Výpočty**

Pro výpočty hodnot teploty a rychlosti byly implementovány podprogramy, které danou fyzikální veličinu přepočítávají podle vzorců, vycházejících z vlastností senzorů. Tyto vzorce bylo nutné upravit tak, aby jejich výpočet vyžadoval co nejmenší počet matematických operací. Tím, výpočet zabral procesoru méně času a tak došlo k úspoře paměti programu a snazší implementaci. Každý krok navíc by zabral určité místo v programu, ale například při převedení veškerého dělení na násobení vynechal kompilátor dělicí rutiny a ušetřil další místo v paměti. Pro přesnější výsledek byly použity matematické rutiny pracující s 16-bitovým číslem, přestože procesor umí pracovat pouze s 8-bitovým číslem. Tyto rutiny byly uvolněny výrobcem procesoru a jsou součástí této práce.

## **7. Software pro načítání dat do počítače a přenos na internet**

V této kapitole bude popsána aplikace, která se stará o to, aby změřená data byla ukládána do počítače a zobrazena v grafickém uživatelském prostředí programu meteok3. Tato aplikace také zajišťuje přenos dat na webový server, který potom generuje webovou stránku s aktuálními výsledky. Celá aplikace byla vyvíjena tak, aby mohla být společně s měřícím přístrojem přenosná, a aby mohla být spuštěna na více počítačích bez nutnosti webového serveru. Nejedná se tedy o serverovou aplikaci, která by nepřetržitě zaznamenávala data a vyhodnocovala je. Aplikace by měla být spíše doplňkem měřícího přístroje, který byl popsán výše. Jelikož přístroj pouze demonstruje činnost anemometru a poskytovaná data jsou orientační, aplikace nijak nezajišťuje bezpečnost přenosu dat, vyhodnocování chyb, kalibraci nebo jejich archivaci. Je navržena tak, aby společně s přístrojem přehledně poskytovala aktuální hodnoty, které mohou být okamžitě vyhodnoceny nebo odeslány na server.

Za účelem tvorby aplikace bylo využito vývojového prostředí Microsoft Visual Studio 2005, protože poskytuje výbornou podporu k obsluze sériového portu. Pro správné fungování je důležité mít na uživatelském počítači nainstalovanou podporu běhového prostředí .NET framework 2.0, které sdružuje všechny důležité knihovny pro běh aplikace. Výstupem aplikace je XML soubor obsahující změřená data.

Počítač na kterém bude spuštěn program meteok3, musí být připojen k síti internet, aby aplikace mohla spouštět obslužné skripty a posílat data na webový server. Skripty jsou napsány pomocí skriptovacího jazyka PHP, který byl vyvinutý především pro tvorbu dynamických webových stránek, ale hodí se i na tvorbu samostatných skriptů sloužících k různým účelům. První skript na serveru se stará o



načítání dat z datového souboru do databáze MySQL, druhý potom data z databázového zdroje přehledně zobrazuje v HTML stránce. Pro vzájemnou komunikaci mezi aplikací a serverem je využíváno HTTP protokolu a pro přenos datového souboru je použit FTP protokol.

## 7.1. Program meteoK3

Program pro načítání dat z měřícího přístroje splňuje tyto požadavky:

- načítání znaků ze sériového kanálu,
- konfigurace uživatelské aplikace,
- zobrazování naměřených výsledků,
- ukládání dat do výstupního souboru,
- přenos dat na server v pravidelném intervalu.

### 7.1.1. Konfigurace

Každá aplikace má určitá nastavení. V tomto případě se jedná o nastavení sériového kanálu, nastavení parametrů FTP přenosu a nastavení parametrů měřených veličin. K zapamatování celého nastavení by mohla aplikace ukládat údaje do konfiguračního souboru nebo do registrů. S tím vyvstává ale problém se správou těchto souborů. *Visual Studio* pro tyto účely nabízí podporu ve formě silně typované třídy, dostupné v podobě `Projekt.Properties.Settings`, která zajišťuje veškerou správu a činnost uživatelského nastavení. Konfigurační soubor se ukládá do uživatelského profilu, ze kterého jsou konfigurační data při opětovném spuštění aplikace načtena. Aplikace umožňuje nastavovat následující parametry:

a) pro sériovou komunikaci:

- název sériového portu, na kterém je zařízení připojeno,
- rychlost komunikace,
- uvozovací znaky datového paketu.

b) pro přenos dat na internet:

- adresu FTP serveru,
- uživatelské jméno,
- uživatelské heslo.

c) měřené veličiny:

- název veličiny,
- jednotka,
- počet posílaných znaků, které nesou informaci o veličině,
- povolení zobrazování.

Vlastnosti potřebné pro nastavení měřených veličin vychází z rozboru datového paketu, popsaného na obrázku (4.3).

## 7.2. Běh aplikace

Hned po spuštění se aplikace pokouší nahrát konfigurační data ze souboru. V případě, že se konfigurační data nepodaří získat, nastaví se výchozí hodnoty. Program se může nacházet ve dvou stavech. První stav „stop“ nepřijímá ani nevyhodnocuje data, ale umožňuje provádět nastavení celé konfigurace. Tím je dosaženo toho, že za běhu programu nedojde například ke změně parametrů sériové komunikace. Druhým stavem je stav „start“, kdy program přijímá znaky na sériové lince a provádí jejich následné zpracování a ukládání do souboru. V tomto stavu se také provádí zobrazování a odesílání dat na server.

Obsluha sériového portu je realizována prostřednictvím vestavěné komponenty *SerialPort*, která obsahuje metody a události pro práci se sériovým přenosem. Kontrolka je standardní součástí vývojového prostředí a díky ní lze pohodlně realizovat nastavování parametrů sériového kanálu, povolovat čtení/zápis příchozích/odchozích znaků a provádět tak další zpracování.

## Načítání a zobrazení dat

Celé zpracování probíhá následovně. Pokud se program nachází ve stavu „start“, odešle měřicímu přístroji žádost o data. Ten, pokud jsou data dostupná, odešle celý datový paket (obrázek 4.3) počítači a pokračuje v činnosti. Počítač načte datový paket a rozdělí jej na jednotlivé veličiny podle předem nadefinovaných vlastností, nastavených v konfiguraci. Data jsou nejdříve otestována na rovnost celkového počtu přijatých znaků ku celkovému počtu očekávaných znaků. Pokud počty nesouhlasí, program dále nepokračuje a žádá o data znovu. V opačném případě jsou získané hodnoty převedeny na primitivní typy jako je *String* nebo *Integer*. Zobrazení dat se provádí okamžitě po úspěšném načtení a otestování. Výsledné hodnoty jsou znázorněny jak v textové, tak v grafické podobě. Pro grafickou podobu autor navrhl tři vlastní uživatelské kontrolky, s kterými se dobře pracuje díky jejich vlastnostem.

## Zpracování a ukládání dat

Výše popsané načítání se provádí v krátkých, opakujících se intervalech, a proto je důležité před uložením data zpracovat. Provádí se tak vždy v celou hodinu, kdy je z načtených dat vypočítáván průměr, a v případě směru větru se za výsledný směr považuje nejčtenější hodnota. Data jsou ukládána do XML souboru, kde každá veličina má následující strukturu:

```
<velicina>  
  <Datum>31/03/2008</Datum>  
  <Cas>16:35</Cas>  
  <Nazev>Teplota</nazev>  
  <Hodnota>26.3</hodnota>  
  <Jednotka>°C</jednotka>  
</velicina>
```

Díky značkám XML jazyka, které datový soubor obsahuje, je jeho pozdější zpracování snazší a pohodlnější. Může se také provádět validace celého souboru pomocí tzv. XSD šablony, ale v tomto případě by to bylo zbytečné. Data nejsou tak rozsáhlá a složitě strukturovaná, aby vyžadovala tuto validaci. Typem jednotlivých hodnot je typ *String*.

### **7.3. Přenos dat na internet**

V této části již program více spolupracuje s webovým serverem, který zajišťuje prezentaci naměřených výsledků. Hlavním úkolem je uložit data z datového souboru vytvořeného programem meteoK3 do databázové struktury. Protože se data nenachází na databázovém serveru, je nejdříve celý datový soubor přenesen pomocí protokolu FTP na příslušný server, kde jsou z něj data nahrána do databáze. Přenos dat je realizován pomocí vestavěné třídy *WebClient* z balíku .NET, a to konkrétně ze jmenného prostoru *System.NET*. Třídě stačí nastavit parametry připojení jako je jméno, heslo, adresa, a pomocí vhodné metody odeslat datový soubor na server.

#### **Kopírování dat do databáze**

Jak už bylo zmíněno výše, o kopírování dat do databáze se stará skript umístěný na serveru. Tento skript je součástí předkládané práce a stará se o načtení dat ze souboru, následné rozdělení a uložení jednotlivých hodnot veličin do databáze. Celý mechanismus byl realizován pomocí *XMLParseru*, což představuje nejlepší variantu získávání dat ze souboru. Spouštění skriptu se provádí voláním z aplikace meteoK3, která k tomuto účelu využívá HTTP dotaz. To znamená, že skript již musí být fyzicky přítomen na serveru a umístěn v určité adresářové struktuře. Pro práci s HTTP dotazem je používána třída *HttpWebRequest*, která obsahuje potřebné metody.

Po nakopírování dat do databáze se datový soubor ze serveru smaže a následně proběhne kontrola již uložených dat. Kontrola se provádí databázovým dotazem, který ověřuje stáří jednotlivých záznamů tabulky delších než jeden týden. Pokud taková data existují, vytvoří archivní soubory pomocí kompresní metody LZW a data z databáze smaže. Data jsou tedy v databázi archivována po dobu jednoho týdne, a pokud by bylo potřeba zpracovávat údaje ještě starší než týden je možné implementovat skript který by archivní soubory zpracovával.

### **Struktura databáze**

Pro prezentaci hodnot na webové stránce je důležité zvolit vhodnou formu ukládání dat do SQL databáze. Pomocí SQL dotazů lze totiž snadno načítat data podle požadovaných parametrů. Proto je důležitým krokem návrh vhodné struktury databázové tabulky. Jednotlivé veličiny, v daném případě teplota, rychlost a směr, představují samostatné řádky tabulky. Sloupce tabulky jsou určeny vlastnostmi veličin (datum, čas, název, hodnota atd.). Díky jednoduché struktuře lze v SQL databázi snadno vytvořit požadovanou tabulku. Při návrhu tabulky je důležité brát ohled na způsob, jakým budou data načítána, a na základě toho vhodně zvolit typ jednotlivých sloupců. Databáze podporuje několik typů, z nichž každý se hodí pro různé účely, například typ pro textové hodnoty nebo typ pro číselné hodnoty. K uchování časových nebo datumových údajů se používá tzv. *timestamp*, který představuje 32-bitové číslo, udávající počet sekund od data 1. 1. 1970. Díky funkcím jazyka PHP lze takto typovaný údaj libovolně převádět do různých datumových a časových formátů.

Hodnoty z datového souboru jsou ukládány vždy v každou celou hodinu, a tak je počet záznamů v tabulce malý. Díky tomu se snadno provádí výpočet maximální, minimální nebo průměrné hodnoty. Pro jednodušší získávání posledních změřených hodnot se používá pomocná tabulka, která má shodnou

strukturu s tabulkou popsanou výše. Jejím obsahem jsou vždy poslední aktuálně změřené hodnoty. Její Plnění provádí je prováděno procesem, který může být spuštěna nezávisle na pravidelné hodinové aktualizaci.

Důležitou součástí návrhů tabulek je také návrh indexu. Indexy mohou být vytvořeny na jednom nebo více sloupcích. Slouží především k urychlení načítání dat podle kritérií daného sloupce, k hledání minima, maxima atp.

## 8. Zobrazování výsledků na webové stránce

V této kapitole je popsáno prezentování výsledků na webové stránce. Nejdříve jsou uvedena kritéria a požadavky na *Webhosting*, na kterém budou stránky s prezentací běžet. V další části bude nastíněn mechanismus generování grafu podle vstupních parametrů, a v poslední části kapitoly budou představeny náhledy stránek a grafů. Vzhledem ke krátké době vývoje těchto stránek je prezentační rozsah velice jednoduchý a stránky jsou pořád ve vývoji.

### 8.1. Požadavky pro běh webových stránek

Pro provoz stránek je nutné zřídit účet na některém z dostupných webhostingových serverů. Nároky na poskytovatele webového prostoru nejsou velké a v současné době je všichni splňují. V případě předkládané bakalářské práce byl vybrán server <http://molbud.cz> a na něm zřízen účet pro webovou prezentaci naměřených dat. Stránky běží na adrese <http://meteo.molbud.cz> od 31. 3. 2008.

#### Požadavky pro provoz stránek jsou následující:

- webový server s podporou PHP 5.0
- rozšíření PHP o knihovnu GD2 pro manipulaci s obrázky
- databázový server MySQL
- prostor 50MB

Dalším požadavkem, který tolik nesouvisí s provozem webového serveru je přídatná knihovna JpGraph do distribuce PHP. Tato knihovna není standardní součástí jazyka PHP a je uvolněna pod licencí QPL 1.0. Pro její správnou činnost je vyžadována také podpora knihovny GD2.

## 8.2. Struktura webových stránek

K návrhu struktury a vzhledu webových stránek je nutné přistupovat z mnoha hledisek. Na návrhu se obvykle podílí dodavatel obsahu spolu s grafickým návrhářem. Grafický návrh putuje k programátorovi webových stránek, který převede vzhled do HTML (případně dalších technologií, v současné době hlavně CSS), a nasadí ho například na určitý redakční systém. Nakonec zbývá systém naplnit obsahem, otestovat a vystavit na web. Při návrhu stránek pro zobrazování měřených hodnot teploty, rychlosti a směru větru bylo stanoveno několik základních požadavků na vzhled a obsah.

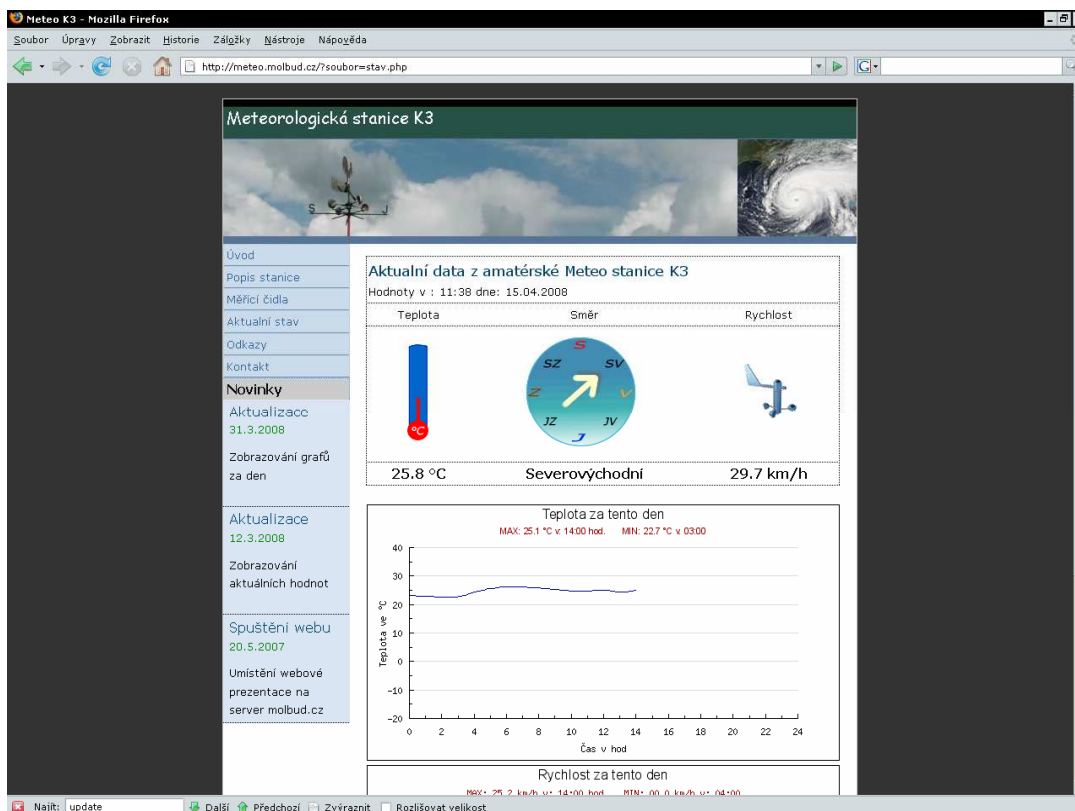
- jednoduché rozvržení stránky,
- jednoduchý a přehledný vzhled,
- výrazné textové zobrazení aktuálních hodnot,
- grafický průběh hodnot aktuálního dne.

### 8.2.1. Rozvržení stránky

Rozvržení stránky je podobné jako používá mnoho dalších webových stránek. Nejdominantnějším prvkem stránky je prostřední horní část, ve které jsou textově i graficky zobrazeny aktuální hodnoty. V levé části je klasické navigační menu a pod ní další různé informace, jako jsou rekordní hodnoty teploty nebo aktuality vztahující se k obsahu stránek.

Hlavní část stránek tvoří blok umístěný uprostřed, na němž jsou zobrazeny průběhy teploty vzduchu a rychlosti větru za aktuální den. Rozvržení je ukázáno na obrázku (8.1).





Obrázek 8.1: Rozvržení stránky

### 8.3. Skripty PHP

Pro zobrazení úvodní stránky je volán soubor *index.php*, který zahrnuje celou funkci webu, tvoří pomyslný programový rozcestník stránek a zajišťuje také připojení externího souboru se styly. Hlavní skript, který zobrazuje aktuální hodnoty, se jmenuje *stav.php*. Obsahuje několik databázových dotazů pro získání požadovaných hodnot z databázové tabulky a zároveň spouští generování dvou obrázků představujících grafické zobrazení průběhu naměřených hodnot za aktuální den.

## Generování obrázku

Proces generování zajišťují skripty `teplota.php` a `rychlost.php`. Každý obrázek v HTML jazyce musí být uvozen nepárovou značkou `<img>`, v jejímž těle se nachází atribut `src`, který určuje cestu ke zdroji obrázku. Jelikož obrázek je výstupem výše uvedených souborů, je do tohoto atributu dosazena cesta právě k těmto souborům. Nepárový tag v HTML jazyce pro výsledný obrázek vypadá takto:

```

```

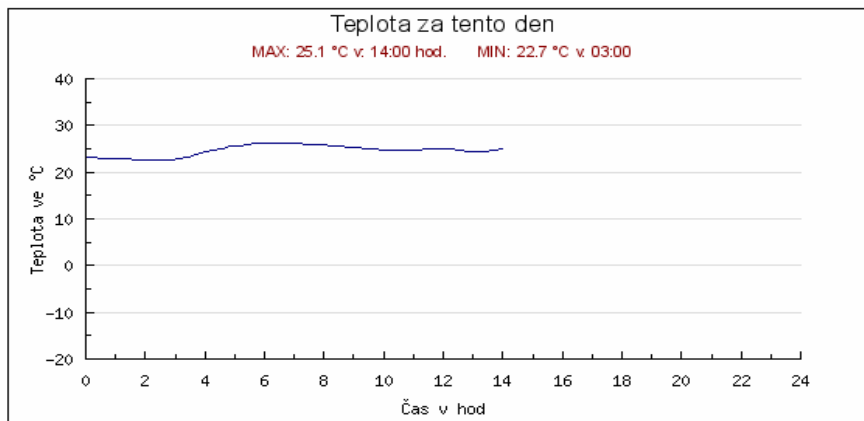
## Soubor `teplota.php`

Vstupními údaji pro výsledný graf jsou data z databáze. Nejdříve se provádí databázový dotaz, který požaduje data pro danou veličinu za celý aktuální den. Potom se volají PHP funkce z knihovny JpGraph, které nastavují další parametry nutné k vygenerování obrázku, například různé konstanty pro stanovení hodnot na osách  $x$  a  $y$ , popisky os, jejich umístění, styl fontu, vykreslování hlavní a vedlejší mřížky atd.

Dále se určuje typ grafu, nastavuje se rozsah os  $x$  a  $y$  a rozteč mřížky, která může být nastavena programově nebo automaticky. Současně se také nastavuje parametr, který udává typ výstupního souboru. Na výběr jsou tyto formáty: JPG, PNG nebo GIF. V další fázi je vytvořeno pomyslné plátno s prázdným obrázkem, na které je postupně zaznamenáván průběh.

Při kreslení je možno využít několik metod pro odstranění šumu tak, aby výsledná spojnice mezi jednotlivými body byla pokud možno hladká. Ovšem s tímto požadavkem vyvstává problém se zobrazováním náhlých a rychlých změn hodnot. V poslední fázi je do obrázku přidán popisek s názvem zobrazované

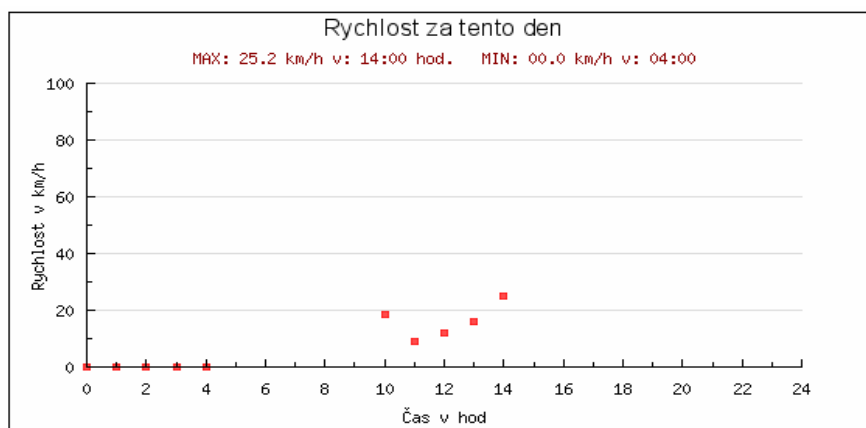
veličiny a také jsou doplněny údaje o minimální a maximální hodnotě. Vše je vidět na obrázku (8.2).



Obrázek 8.2: *Graf teploty*

### Soubor rychlost.php

Funguje velice podobně jako soubor teplota.php až na drobné odlišnosti ve způsobu zobrazování. Ty se projevují při kreslení průběhu grafu rychlosti, u níž je větší pravděpodobnost náhlých a rychlých změn hodnot. Z toho důvodu se používá bodový graf, aby průběh neznázorňoval špatné hodnoty. Výstupem je například obrázek(8.3)



Obrázek 8.3: *Graf rychlosti*

## 9. Závěr

Úkolem této bakalářské práce bylo prozkoumat problematiku získávání dat z přístroje, který je schopen měřit neelektrické veličiny, a dále prezentovat zpracované výsledky na internetové stránce. Práce přináší nejen teoretické poznatky o měření těchto veličin a principech fungování snímačů, které jsou pro taková měření využívána, ale rovněž popisuje způsob přenosu informace z měřicího přístroje do počítače, včetně přenosu dat na internet. Součástí práce je také názorná ukázka toho jakým způsobem probíhají jednotlivá měření vybraných veličin (směru, rychlosti a teploty vzduchu) za pomoci konkrétního přístroje (anemometru) určeného k demonstraci.

Na základě zmíněných hledisek proběhla analýza měřicího přístroje zkonstruovaného pro laboratorní a experimentální účely, na kterém bylo možné celou problematiku získávání dat předvést. Tím se skloubila praktická část s teoretickou, která se k danému tématu vztahovala. Hlavní náplní analýzy přístroje bylo pochopení principu činnosti jednotlivých senzorů, vyhodnocení signálů a určení jejich vlastností. V důsledku toho byl upraven stávající software přístroje tak, aby správně přenášel data do počítače a byl vytvořen lepší formát posílaných dat pro samotné načítání.

Samotné měřicí zařízení nemůže konkurovat komerčním anemometrům, jejichž přednostmi jsou menší hmotnost, lepší mobilita díky bezdrátovému spojení senzorů, a poskytování přesnějších hodnot měření. Ale přístroj je určen k lepšímu pochopení dané problematiky a je schopen reálně napodobovat chování anemometru. Využití celého přístroje je spatřována například v praktických cvičeních z fyziky nebo elektroniky. Pro zkvalitnění činnosti přístroje by bylo vhodné opatřit jej například pamětí pro dlouhodobější uchovávání dat bez nutnosti připojovat počítač, nahradit starší rozhraní RS-232 modernějším USB

portem, vyřešit bezpečnost přenosu dat mezi řídicí a měřící jednotkou, a celkově zlepšit parametry všech senzorů.

I přes zmíněné nedostatky však předkládaná práce dosáhla vytyčeného cíle a o jeho úspěšném zvládnutí svědčí funkční webové stránky umístěné na adrese <http://meteo.molbud.cz>, na kterých si je možné prohlédnout aktuální změřená data. Tato data poskytuje webové stránce jednoduchá aplikace napsaná v jazyce C#, která je schopna běžet na většině současných platforem podporujících technologii .NET 2.0.

# Literatura

- [1] Obrazová, Hana. *Elektrická měření neelektrických veličin*, 2.kap., *Základy teorie měření*. 3. přepr. vyd. Praha : Ediční středisko ČVUT, 1989. ISBN80-01-00136-9
- [2] Kocourek, Petr. *Přenos Informace* . 1. vyd. Praha : Vydavatelství ČVUT, 2004. 164s. ISBN80-01-02892-5
- [3] Haasz V., Roztočil J., Novák J. *Číslicové měřicí systémy*. 2. přepr. vyd. Praha : Vydavatelství ČVUT, 2000. 315 s. ISBN80-01-02219-6
- [4] Vybíral, Bohumil. *Zpracování dat fyzikálních měření, Studijní text pro řešitele FO, studující fyziku na UHK a ostatní zájemce o fyziku* ,[online]. c 2007. [citováno 15.4.2008]. Dostupný z WWW: <<http://fo.cuni.cz/texty/mereni.pdf>>
- [5] Fyzikální praktikum na MFF UK: *Charakteristiky optoelektronických součástek*, [online]. c 2002. [citováno 15.4.2008]. Dostupný z WWW: <[http://physics.mff.cuni.cz/vyuka/zfp/mereni\\_305.pdf](http://physics.mff.cuni.cz/vyuka/zfp/mereni_305.pdf)>
- [6] Karas, Ondřej. [online]. *Senzory a snímače, studijní materiály z přednášek*. 1.kap. c 2007 [citováno 2.4.2008]. Dostupný z WWW: <<http://www.kvetakov.net/down.php?file=../UEIKMT/2/ls/ISAME/MTVKap1.pdf>>
- [7] SPŠTaSOUT, Třebíč. *Snímače neelektrických veličin: studijní materiály pro LABView*. [online]. c 2006. [citováno 10.04.2008]. Dostupný z WWW: <<http://www.spstlabview.wz.cz/snimace.doc>>
- [8] Pokorný J., Snášel V. *Zpracování proudů dat* . Praha: MFF UK, Ostrava: VŠB TU. [online]. c 2006. [citováno 15.04.2008]. Dostupný z WWW: <<http://www.ksi.mff.cuni.cz/~pokorny/papers/DATAKON06.pdf>>
- [9] Mallat J. *Co je co v IT > streaming*. [online]. c 2004. [citováno 15.04.2008]. Dostupný z WWW:<<http://hps.mallat.cz/view.php?cisloclanku=2004042001>>
- [10] *Simple Serial Communication with Microsoft Visual C# Express*. [online]. Práce s COM portem. Dostupný z WWW: <<http://csharp.simpleserial.com/>>

- [11] *on-line JpGraph* . [online]. c 2008. [citováno 12.04.2008].  
Dostupný z WWW: <<http://www.aditus.nu/jpgraph/jpdownload.php>>
- [12] Fakulta elektrotechnická. Praha: ČVUT. *Polovodičové teploměry : úlohy ze cvičení*. [online]. c 2003. [citováno 13.04.2008]. Dostupný z WWW:  
<<http://measure.feld.cvut.cz/groups/edu/sz/pdfs/uloha1.pdf>>
- [13] Pešout, Martin, *Modul pro implementaci slučování grafů: bakalářská práce*.  
Brno: Fakulta informatiky MU, 2006. 25s. [online]. [citováno 19.04.2008 ]  
Dostupný z WWW: <[http://is.muni.cz/th/99232/fi\\_b/](http://is.muni.cz/th/99232/fi_b/)>
- [14] Wikipedice.cz, *Komprese dat*. [online]. 4. února 2008 v 10:45  
[citováno 19.04.2008] Dostupný z WWW:  
<[http://cs.wikipedia.org/wiki/Komprese\\_dat](http://cs.wikipedia.org/wiki/Komprese_dat)>

## **Přílohy**

Vytvořený software doplňující funkce měřícího přístroje je přiložen na datovém nosiči CD-ROM, který je součástí tištěné verze bakalářské práce, archivované na PF JCU.