

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE



Katedra elektrotechniky a automatizace

Vývoj inteligentního senzoru relativní vlhkosti vzduchu
diplomová práce

Vedoucí diplomové práce: prof. Ing. Zdeněk Bohuslávek, CSc.

Diplomant: Bc. Ondřej Tesař

PRAHA 2009

Vysoká škola: Česká zemědělská univerzita v Praze	Fakulta: technická
Katedra: elektrotechniky a automatizace	Akademický rok: 2007/2008

ZADÁNÍ DIPLOMOVÉ PRÁCE

Diplomant: **Ondřej Tesař**

Studijní obor: **Informační a řídicí technika v APK**

Název práce: **Vývoj inteligentního senzoru relativní vlhkosti vzduchu**

Zásady pro vypracování:

Cíl práce: **Zpracování návrhu tzv. inteligentního (procesorem řízeného) psychrometrického vlhkoměru vzduchu a jeho praktické ověření na funkčním modelu.**

Osnova práce:

- Přehled používaných metod pro měření vlhkosti vzduchu. Charakteristické vlastnosti vybraných metod a technických prostředků.
- Charakteristika vlastností a stavby inteligentních (smart) senzorů.
- Charakteristika vlastností embedded mikrokontrolerů, přehled vybraných typů mikroprocesorů užívaných pro vestavbu do senzorů a přístrojů.
- Návrh a realizace řídicí části technologického psychrometrického vlhkoměru vzduchu.
- Návrh a odladění software mikroprocesoru.
- Testování vlastností návrhu s použitím dodaného ovládacího programu pro PC,
- Zhodnocení a závěr.

Metodika práce:

- Zpracování literárních podkladů a konzultace u firem k bodům osnovy a), b) a c).
- Zjištění vlastností dodaného HW klasického technologického psychrometrického vlhkoměru a provedení koncepčního návrh HW řídicí části vlhkoměru.
- Provedení detailního návrhu řešení funkčního modelu zařízení podle bodu osnovy d) a návrh řešení SW podle bodu e).
- Odladění SW, provedení laboratorních zařízení a jejich vyhodnocení.

Rozsah práce: 50-60 stran textu včetně obrázků, grafů a tabulek

Seznam doporučené odborné literatury:

1. Ďaďo S., Kreidel: Sensory a měřicí obvody, skripta ČVUT, Praha, 1996.
2. Gajski, D.: Specification and Design of Embedded Systems. Eastbourne, 2002.
3. Přístrojové aplikace mikroprocesorů / Jiří Podlešák -- 1. vyd. -- Praha : ČVUT, 1999
4. Haasz, V., Roztočil, J.: Měřicí systémy. ČVUT FEL, 1999, 55s.

Vedoucí diplomové práce: prof. Ing. Zdeněk Bohuslávka, CSc.

Datum zadání diplomové práce: 10.12. 2007

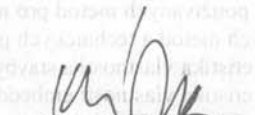
Termín odevzdání diplomové práce: 30. 4. 2009





prof. Ing. Karel Pokorný, CSc., dr.h.c.

vedoucí katedry



prof. Ing. Jiří Klíma, CSc.

děkan

V Praze dne 21.1.2008

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, pod vedením prof. Ing. Zdeňka Bohuslávka, CSc. Další informace mi poskytl Ing. Petr Mík z firmy Asix. Použité literární prameny jsou uvedeny v příloženém seznamu použité literatury.

Dále prohlašuji, že souhlasím s půjčováním a zveřejňováním své diplomové práce nebo její části se souhlasem katedry.

V Praze dne 30. dubna 2009

Podpis:

Poděkování

Děkuji rodině a přátelům za podporu.

Abstrakt: Práce popisuje vývoj konstrukce inteligentního senzoru vlhkosti, kterým je doplněn modul pro psychrometrické měření relativní vlhkosti vzduchu. Také obsahuje přehled několika profesionálních zařízení vhodných pro tyto aplikace. Shrnuje požadavky kladené na smart senzory relativní vlhkosti, uvádí příklady jejich použití a principy funkce jednotlivých typů čidel. Popisuje několik průmyslových standardů, používaných pro přenos naměřených hodnot mezi jednotlivými senzory. V druhé části se věnuje sestavení jednoduché úlohy, tzv. smart senzoru. Jedná se o elektronický modul, který je osazen mikroprocesorem. Vyhodnocuje teplotu suchého a mokrého teploměru, ze které vypočítává hodnotu relativní vlhkosti vzduchu. Tyto údaje zobrazuje na LCD displeji.

Klíčová slova: psychrometr, teplota suchého teploměru, teplota mokrého teploměru, měření vlhkosti.

The development of smart sensor of the air humidity

Summary: This thesis describes the development of relative humidity smart sensor construction which can be connected to a psychrometric measuring module. It also includes an overview of several professional devices which are suitable for this kind of application. It summarises the requirements for relative humidity smart sensors and the principles of converting humidity to electrical values. It describes several industrial standards, used for transmitting measured values from individual sensors. The smart sensor project is described in the second part; it is an electronic module, based on microprocessor ATmega8. The dry and wet bulb temperature is evaluated and relative humidity is also calculated. These data are displayed on a LCD display.

Key words: psychrometer, dry bulb temperature, wet bulb temperature, humidity measurement.

Obsah:

1. Úvod.....	1
2. Přehled používaných metod měření vlhkosti vzduchu	2
2.1 Metody měření.....	4
2.1.1 Mechanický (vlasový) vlhkoměr.....	4
2.1.2 Gravimetrické (sorpční) měření vlhkosti.....	5
2.1.3 Zrcadlové senzory teploty rosného bodu - kondenzační vlhkoměry	6
2.1.4 Elektrolytický vlhkoměr.....	8
2.1.5 Absorpce záření	10
2.1.6 Odporové senzory vlhkosti.....	11
2.1.7 Kapacitní senzory vlhkosti	13
2.1.8 Vlhkoměr s vyhřívanými termistory	16
2.1.9 Psychrometr.....	17
2.1.10 Kalibrace vlhkoměrů	19
3. Charakteristika vlastností a stavby inteligentních (smart) senzorů.....	20
3.1 Charakteristika vlastností inteligentních senzorů.....	20
3.2 Přehled vybraných typů dostupných smart senzorů vlhkosti	21
3.2.1 Inteligentní senzor THT – Vlhkoměr s teploměrem pro RS485	21
3.2.2 Comet - Snímač teploty, vlhkosti s výstupem RS485, RS232	22
3.2.3 Snímač vlhkosti Profess	23
3.2.4 Dewtron smart senzor.....	24
3.2.5 Smart senzor EE31	25
3.2.6 Psychrometr DSL12	26
3.2.7 Průmyslový psychrometr Hygrophil h	27
4. Charakteristika vlastností embedded mikrokontrolerů.....	28
4.1 Přehled vybraných typů mikroprocesorů užívaných pro vestavbu do senzorů a přístrojů	28
5. Návrh a realizace řídicí části technologického psychrometrického vlhkoměru vzduchu..	29
5.1 Obecná koncepce.....	29
5.2 Návrh napájecího zdroje.....	30
5.3 Výběr mikroprocesoru.....	30
5.4 Metody zjištění relativní vlhkosti pomocí teplotní difference.....	31
5.4.1 Z psychrometrické tabulky	31
5.4.2 Výpočtem	31
5.4.3 Z grafu	33
5.5 Popis měření	34
5.5.1 měření A/D převodníkem:	34
5.5.2 Schéma zapojení.....	35
5.5.3 Seznam součástí.....	35
5.5.4 Mechanické provedení.....	36
5.5.5 Mechanické provedení.....	36
6. Návrh a odladění software mikroprocesoru.....	37
6.1 Obsluha A/D převodníků.....	37
6.2 Prováděné výpočty	37
6.3 Obsluha LCD displeje	38
6.4 Obsluha sériové linky.....	38
6.5 Programování mikroprocesoru	39
7. Testování vlastností návrhu	40
7.1 Dosažené parametry	40

7.2	Rychlost změny měření	40
7.3	Měření vlhkosti.....	41
8.	Zhodnocení	42
8.1	Zhodnocení vlastností návrhu.....	42
8.2	Ekonomické zhodnocení návrhu	43
8.3	Soupis nákladů součástí	43
9.	Závěr	44
10.	Použitá literatura.....	45
10.1	Internetové zdroje:.....	45
10.2	Publikace	47
10.3	Seznam použitých zkratk	48
10.4	Seznam obrázků.....	49
10.5	Seznam tabulek.....	50
10.6	Seznam grafů	50
11.	Přílohy	1
11.1	Výkres plošného spoje.....	1
11.2	Výkres osazení plošného spoje.....	2
11.3	HX Diagram	3
11.4	Výpis programu	4

Cíl práce: Zpracování návrhu tzv. inteligentního (procesorem řízeného) psychrometrického vlhkoměru vzduchu a jeho praktické ověření na funkčním modelu.

Osnova práce:

- a) Přehled používaných metod pro měření vlhkosti vzduchu. Charakteristické vlastnosti vybraných metod a technických prostředků.
- b) Charakteristika vlastností a stavby inteligentních (smart) senzorů.
- c) Charakteristika vlastností embedded mikrokontrolerů, přehled vybraných typů mikroprocesorů užívaných pro vestavbu do senzorů a přístrojů.
- d) Návrh a realizace řídicí části technologického psychrometrického vlhkoměru vzduchu.
- e) Návrh a odladění software mikroprocesoru.
- f) Testování vlastností návrhu s použitím dodaného ovládacího programu pro PC.
- g) Zhodnocení a závěr.

Metodika práce:

- 1) Zpracování literárních podkladů a konzultace u firem k bodům osnovy a), b) a c).
- 2) Zjištění vlastností dodaného HW klasického technologického psychrometrického vlhkoměru a provedení koncepčního návrhu HW řídicí části vlhkoměru.
- 3) Provedení detailního návrhu řešení funkčního modelu zařízení podle bodu osnovy d) a návrh řešení SW podle bodu e).
- 4) Odladění SW, provedení laboratorních zařízení a jejich vyhodnocení.

1. Úvod

Měření vlhkosti je vedle měření teploty dalším z důležitých měření v mnoha technologických procesech. Pro rychlá orientační jednorázová měření vlhkosti se nejčastěji používají sondy s kapacitními čidly. Pokud ovšem potřebujeme měřit vlhkost v udírnách, sušárnách a podobných prostorech s vysokým množstvím nečistot v ovzduší, může být psychrometrické měření vhodná volba. Vykazuje totiž dobrou odolnost vůči mechanickým a chemickým nečistotám v měřeném prostředí.

Psychrometrický způsob měření vlhkosti vzduchu používá dva teploměry, tzv. suchý a mokrý. Povrch mokrého je vlhčen vodou a oba jsou stejnou měrou ofukovány proudem vzduchu, jedná se o tzv. Assmanův psychrometr. Toto uspořádání způsobuje, že teplota suchého a mokrého teploměru není stejná. Tento rozdíl teplot je při daném tlaku úměrný pouze relativní vlhkosti vzduchu okolního prostředí a nezávislý na okolní teplotě.

Navržené zařízení využívá pro měření relativní vlhkosti vzduchu psychrometrický senzor vlhkosti RTU-01-T, jehož výrobcem byl Státní statek Přeštice – přidružená výroba Stod. Tento modul je doplněn o zobrazování hodnot teploty suchého a mokrého teploměru a výsledné hodnoty relativní vlhkosti vzduchu. K zobrazení je použit dvouřádkový LCD displej. Dále je možnost předávání naměřených hodnot pomocí sériové linky pro případné další zpracování.

Měřicí rozsah vlhkoměru je dán teplotním rozsahem použitých teploměrů. V tomto případě je jejich rozsah 0–102,4 °C. Tento rozsah odpovídá výstupnímu proudu proudové smyčky v rozsahu 0–20 mA. Principiálně je možno psychrometricky měřit vlhkost vzduchu i v případě záporných teplot, v případě tohoto přístroje tomu tak není, protože jsme omezeni rozsahem použitých teploměrů.

2. Přehled používaných metod měření vlhkosti vzduchu

Vlhký vzduch se skládá ze směsi tvořené suchým vzduchem a vodní párou. V případě, že je v této směsi voda ve stavu plynném, jedná se o směs homogenní, což je soustava látek, které všechny tvoří jednu fázi. Vlhký vzduch ovšem může také tvořit soustavu heterogenní, což je různorodá soustava látek několika fází. V tom případě je voda obsažena z části ve stavu páry a z části ve stavu kapalném – tj. vodní kapky, déšť či mrholení. Další možností je, že voda je obsažena ve stavu páry a částečně ve stavu tuhém, tj. v podobě ledových krystalků, sněhu nebo krup. A poslední možností, která může nastat, je kombinace stavu páry a částečně kapalného i pevného stavu, tj. kombinace obou výše uvedených možností [1].

Pro vyjádření množství vodních par ve vzduchu se používá několik charakteristik. Hmotnost vodní páry obsažené v jednotce objemu vzduchu označujeme jako absolutní vlhkost:

$$\rho = \frac{m_v}{V} = \rho_v \quad [\text{kg/m}^3; \text{g/m}^3]$$

Jednotkou je gram na krychlový metr $[\text{g/m}^3]$. Což je vlastně hustota vodních par. Množství vodních par, které je za daných podmínek vzduch schopen pojmout, je omezené. Při nasycení vzduchu vlhkostí již další vlhkost nepřijímá. Konkrétní hodnota závisí kromě tlaku vzduchu hlavně na teplotě. S rostoucí teplotou roste schopnost pojmout větší objem vodních par.

Relativní vlhkost vzduchu udává, jaký je poměr mezi maximálně možným (nasyceným) m_v'' a skutečným obsahem vody m_v ve vzduchu. Uvádí se v procentech a označuje %RH nebo %RV.

$$\varphi = \frac{m_v}{m_v''} = \frac{\rho_v}{\rho_v''} = \frac{P_v}{P_v''} \quad [-] \quad \varphi[\%] = 100 \cdot \varphi[-]$$

Značení " se používá pro veličiny vztahující se k vodním parám ve stavu nasyceném.

Ve vzduchotechnických výpočtech se používá měrná vlhkost, což je hmotnost vodních par vztažená na hmotnost suchého vzduchu:

$$x = \frac{m_v}{m_d} \quad [\text{kg/kg s.v.}; \text{g/kg s.v.}]$$

Další veličina popisující stav vlhkosti vzduchu je teplota rosného bodu. Při dosažení této teploty je vzduch právě nasycen a vodní pára v něm obsažená začíná kondenzovat. Rosný bod je charakterizován teplotou, udávanou ve [°C] nebo [K].

Jak suchý, tak vlhký vzduch lze s jeho vlastnostmi v jistém zjednodušení považovat za plyny blížíící se ideálnímu plynu. Tedy pro ně platí stavová rovnice plynu a Daltonův zákon. Ten říká, že ve směsi plynů se každá složka chová tak, jako by daný prostor za stejných podmínek vyplňovala sama a to při svém dílčím (parciálním) tlaku. Dále z Daltonova zákona vyplývá, že celkový tlak směsi je dán součtem parciálních tlaků všech složek, tj. že tlak vlhkého vzduchu je roven součtu parciálního tlaku suchého vzduchu p_a a parciálního tlaku vodních par p_v . [13]

Další veličinou je teplota mokrého teploměru, což je teplota mezního adiabatického ochlazení (tj. ochlazení na stav sytosti odpařováním vody bez přívodu či odvodu tepla).

Hodnoty absolutní vlhkosti vzduchu pro stav nasycení vodní parou, tj. 100 %RH při vybraných teplotách je v Tabulka 1 podle [2].

Tabulka 1 Absolutní vlhkost vzduchu pro 100 %RH a vybrané teploty

Teplota (°C)	Vlhkost (g/m³)
-100	0,000 018
-80	0,000 6
-60	0,011
-40	0,120
-20	0,888
0	4,87
10	9,44
20	17,4
25	23,1
30	30,5
40	51,3
60	130
80	292
100	591

2.1 Metody měření

2.1.1 Mechanický (vlasový) vlhkoměr

Některé organické látky, jako např. lidské vlasy, koňské žíně, hedvábí a podle [2] i některé syntetické materiály jako např. nylon či celuloid, jsou schopné přijmout do své struktury určitý objem vody, jehož množství je přímo úměrné relativní vlhkosti vzduchu. Tato schopnost má vliv na změnu rozměru v závislosti na množství přijaté vody. Změna délky je mechanicky převedena na pohyb ručičky nebo zapisovacího mechanismu pro dlouhodobé sledování průběhu vlhkosti. Používá se i provedení s možností vyhodnocování dosažení mezního stavu. Tento stav je vyhodnocován elektronicky, případně optoelektronicky.

Pro vlastní měření se nejčastěji používají odmaštěné lidské vlasy. Podle [3] se vlasy při změně relativní vlhkosti vzduchu od 0 do 100 % prodlouží o 2,5 %. Měření je nelineární, popsáno vztahem:

$$t = kT \log \varphi$$

kde k je konstanta, T absolutní teplota [K], φ relativní vlhkost v % [10]. V praxi se 15 cm dlouhé vlasy při 0 % vlhkosti vzduchu při změně vlhkosti na 100 % prodlouží asi o 4 mm. Stupnici vlhkoměru lze linearizovat vhodnou mechanickou úpravou, nebo převedením na elektrický signál, a ten dále zpracovat. K výhodám vlasového vlhkoměru (viz Obrázek 1) patří jednoduchost jeho konstrukce a schopnost pracovat i při teplotách pod bodem mrazu. Mezi nevýhody pak malá mechanická odolnost, citlivost na proudění měřeného plynu a omezení maximální teplotou přibližně do 80 °C.

Obrázek 1 Vlasový vlhkoměr

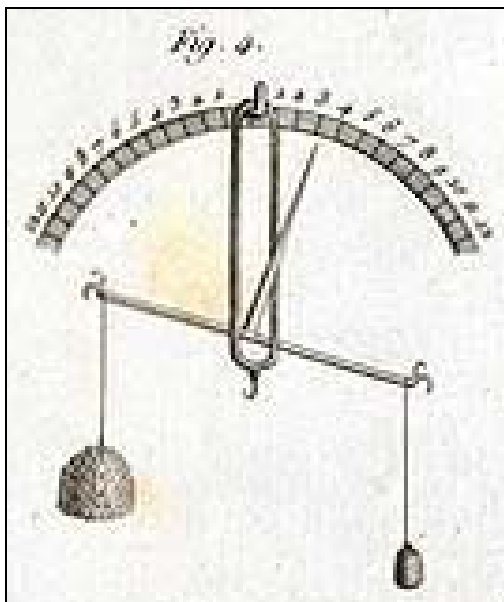


2.1.2 Gravimetrické (sorpční) měření vlhkosti

Gravimetrie je jedna z velmi přesných metod pro měření vlhkosti. Princip metody spočívá v absorpci vzdušné vlhkosti obsažené ve známém objemu plynu do vhodného materiálu. Následně se zjišťuje přírůstek hmotnosti. Měření je náročné časově i na použité přístrojové vybavení. Z těchto důvodů se běžně pro účely provozního měření nepoužívá. Slouží především pro ověřování přístrojů pro měření vlhkosti založených na jiných principech a v laboratorních podmínkách.

Jako první zkonstruoval vlhkoměr využívající tohoto principu Leonardo da Vinci (1452-1590) mezi léty 1480 a 1486. Jeho vlhkoměr využíval existenci hygroskopických látek, jako např. jedlá sůl – chlorid sodný, která pohlcuje vzdušnou vlhkost. S rostoucím množstvím látky, kterou vzorek pojal, rostla i jeho hmotnost. Na druhé misce rovnoramenných vah byl umístěn vzorek z materiálu, který hygroskopický není. V suchém prostředí byly váhy seřízeny na nulovou výchylku. Při měření, s rostoucí vlhkostí rostla i hmotnost hygroskopické látky a tak se při změně vlhkosti vzduchu měnila i výchylka na stupnici rovnoramenných vah, viz. Obrázek 2. [4]

Obrázek 2 Gravimetrický vlhkoměr [14]

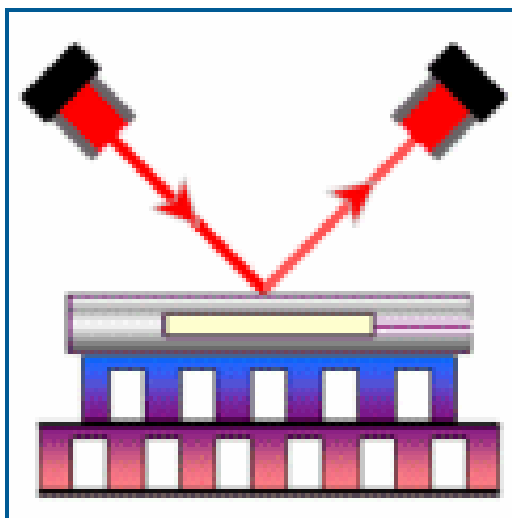


2.1.3 Zrcadlové senzory teploty rosného bodu - kondenzační vlhkoměry

Jedná se o nepřímé měření vlhkosti pomocí zjištění teploty rosného bodu. Rosný bod je teplota, při které je vzduch při daném tlaku maximálně nasycen vodními parami. Pokud totiž klesne teplota pod tuto hodnotu, nadbytečná část vodní páry zkondenzuje. Termín rosný bod zde ovšem vyjadřuje jednak skutečně teplotu rosného bodu, tzv. dew point – DP, ale také teplotu bodu ojínění, tzv. frost point – FP.

Měřicí metoda spočívá v termoelektrickém ochlazování lesklé kovové plochy (zrcátka) do té doby, než se na ní začne srážet vzdušná vlhkost ve formě kapiček. Odrazivost zrcátka se tímto sníží a tento okamžik se opticky vyhodnotí. Zároveň se změří teplota zrcátka. Kondenzační plocha se poté začne mírně zahřívat, dokud na ní zkondenzovaná vlhkost nezmizí. Tento okamžik je opět opticky vyhodnocen a střední hodnota z těchto dvou měření je považována za teplotu rosného bodu. Uspořádání viz Obrázek 3

Obrázek 3 Princip optického měření rosného bodu



Jiným principem pro vyhodnocování vzniku kondenzované vlhkosti je použití krystalového rezonátoru. Jeho rezonanční frekvence je ovlivněna vodní párou zkondenzovanou na jeho povrchu. Resonanční frekvence je následně elektronicky vyhodnocována.

Vztah pro teplotou rosného bodu jako závislost mezi teplotou vzduchu a poměrnou vlhkostí je uveden jako upravená Clausius-Clapeyronova rovnice:

$$E = E_0 \exp\left(-\frac{L}{R} \left(\frac{1}{T} - \frac{1}{273,16}\right)\right)$$

Např. podle [14]: kde:

„ E značí parciální tlak nasycené vodní páry při teplotě T (v Kelvinech)

L je skupenské teplo vypařování

R je měrná plynová konstanta vodní páry

E_0 je parciální tlak nasycené vodní páry při teplotě 273,16 K

Teplotu rosného bodu získáme tak, že zjištěnou hodnotu E vynásobíme poměrnou vlhkostí (např. při 43 % vynásobíme číslo E hodnotou 0,43), tzn. hodnota E nyní vystupuje jako parciální tlak nenasyčené vodní páry. Tuto hodnotu zpětně dosadíme do této rovnice a vypočteme novou hodnotu T , kterou pak můžeme interpretovat jako teplotu rosného bodu.“

Kondenzační vlhkoměry jsou dlouhodobě stabilní, dobře odolávají chemikáliím v měřeném prostředí a jsou přesnými měřicími přístroji. Lepší výsledky poskytuje pouze gravimetrická metoda, která ovšem není příliš vhodná pro běžná a opakovaná měření. Nevýhodou je citlivost na míru znečištění zrcátka. Pokud je toto znečišťování postupné, lze s ním počítat a elektronicky ho kompenzovat.

2.1.4 Elektrolytický vlhkoměr

Elektrolytické vlhkoměry vyhodnocují vzdušnou vlhkost okolního prostředí, která reaguje ve speciálním roztoku s elektrolyticky vylučovanou chemickou látkou. Elektrický náboj prošlý senzorem v tomto uspořádání je úměrný absolutní vlhkosti vzduchu.

Při styku plynu, jehož vlhkost potřebujeme měřit, s hygroskopickou látkou (tj. látkou schopnou pohlcovat a udržovat vlhkost) z něj tato látka přijímá vlhkost tak dlouho, dokud nevznikne roztok o takové koncentraci, při níž je za dané teploty parciální tlak vodních par v roztoku roven parciálnímu tlaku vodní páry plynu v okolním prostředí. Při zmenšení vlhkosti okolního prostředí se voda z tohoto roztoku odpařuje do té doby, dokud není opět dosaženo rovnováhy. Jako hygroskopické látky se používá různých roztoků solí, např. LiCl. Roztok je pak elektricky vodivý a jeho elektrická vodivost se mění v závislosti na obsahu absorbované vody. Vlastní senzor se skládá z nevodivého základního materiálu na kterém jsou vytvořeny dvě kovové elektrody viz Obrázek 4.

Obrázek 4 Uspořádání elektrod elektrolytického vlhkoměru [15]

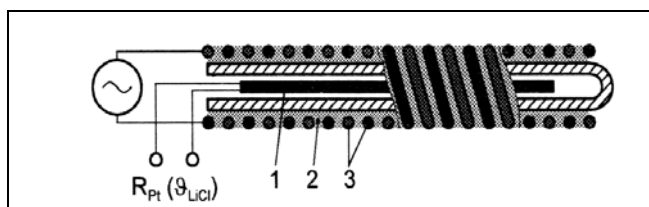


Vlhkostí způsobené změny vodivosti hygroskopické látky, která je nanášena mezi těmito elektrodami, vedou ke změnám elektrického proudu procházejícího senzorem. Aby se zabránilo elektrolytickému rozkladu aktivní látky senzoru, používá se pro měření střídavého napětí. Proud protékající senzorem je úměrný vlhkosti plynu, který je měřen. Vlastní senzor vlhkosti bývá teplotně stabilizován, protože případné změny teploty prostředí ovlivňují rovnovážný stav vlhkosti hygroskopické látky.

Další variantou tohoto principu je systém se skleněnou trubičkou z tenkého skla, která je obalena skelnou tkaninou, viz. Obrázek 5. Ta je nasycena vodným roztokem LiCl a také obsahuje dvě platinové elektrody. Vnitřkem trubičky prochází odporový teploměr. Platinové elektrody jsou napájeny ze zdroje střídavého proudu. Elektrický proud procházející elektrodami zahřívá elektrolyt, čímž dochází k částečnému odpařování vody v něm obsažené. Při nárůstu vlhkosti vzduchu v prostředí okolo snímače, dojde ke zvětšení vodivosti elektrolytu, protože pohltí více vzdušné vlhkosti. Dochází k nárůstu elektrického proudu procházejícího elektrolytem a ten se zahřívá. Tím je stále udržována rovnovážná teplota elektrolytu, která tak je závislá pouze na parciálním tlaku, který má vodní pára v okolním vzduchu. Odečítání hodnoty je ve °C rosné teploty odporového teploměru uvnitř trubičky, nebo v absolutní vlhkosti, případně v parciálním tlaku vodní páry. Pro měření relativní vlhkosti vzduchu elektrolytickým vlhkoměrem je potřeba měřit ještě kromě rovnovážné teploty i teplotu vzduchu. K tomu je doplněn další teploměr.

Elektrolytické vlhkoměry jsou přesné a reagují rychle na změny vlhkosti, ustáleného stavu dosahují přibližně za 30 s. Tyto vlhkoměry je možno použít i pro měření v systémech stlačeného vzduchu. Nevýhodou tohoto systému je časově omezená životnost aktivní vrstvy, protože dochází k jejímu zanesení nečistotami.

Obrázek 5 Elektrolytický vlhkoměr [27]

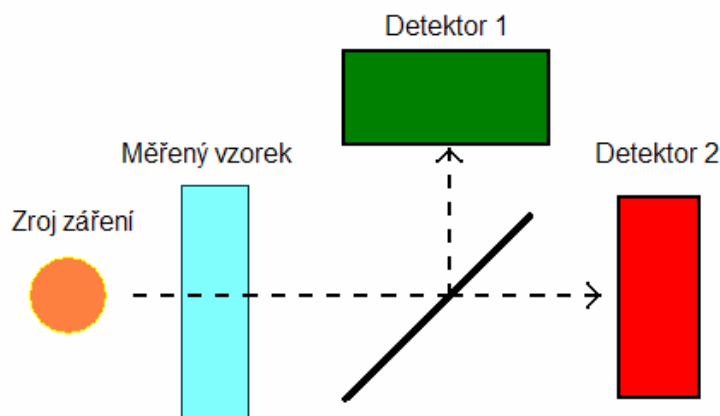


2.1.5 Absorpce záření

Pro zvláštní účely se používají vlhkoměry vyhodnocující absorpci ultrafialového nebo infračerveného záření vodní parou. Výstupní signál je úměrný absolutní vlhkosti plynu a předností je velmi krátká doba odezvy (10–100 ms) a schopnost měřit i velmi malé koncentrace. Zdrojem ultrafialového záření jsou výbojky plněné vodíkem, argonem nebo kryptonem či žhavené zdroje pro infračervenou oblast spektra.

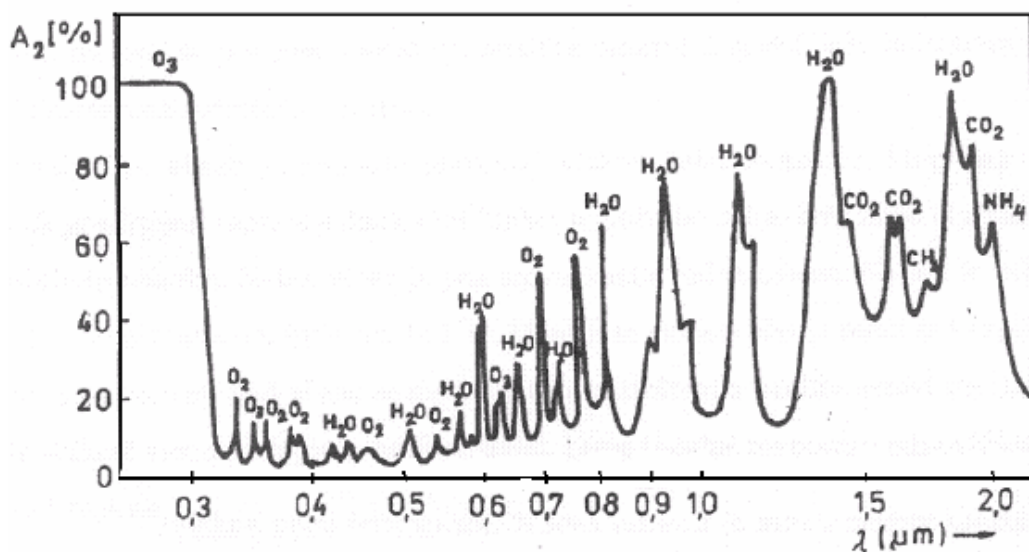
Při jednom z uspořádání prochází část záření měřeným prostředím a část jde přímo na referenční detektor, viz Obrázek 6

Obrázek 6 Uspořádání detektorů záření



Z poměru signálů na obou detektorech je vypočtena vlhkost. Volba vlnové délky je závislá na zvoleném zdroji záření. Závislost útlumu atmosféry na zvolené vlnové délce ukazuje Obrázek 7.

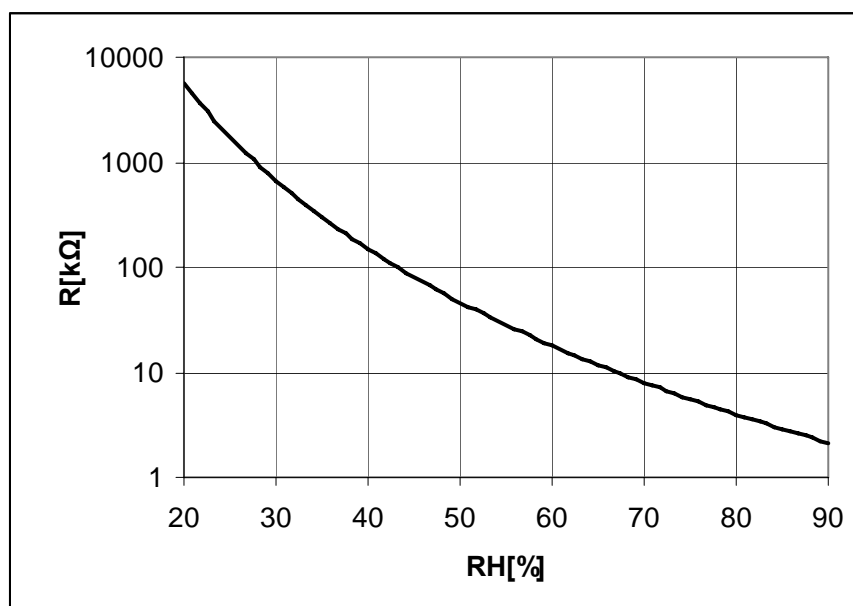
Obrázek 7 Absorpce optického záření v atmosféře [16]



2.1.6 Odporové senzory vlhkosti

U odporových senzorů vlhkosti se využívá změny vodivosti některých hygroskopických materiálů jako např. vodivých polymerů nebo solí. Typický průběh změny hodnoty odporu v závislosti na vlhkosti je inverzní exponenciála, viz Graf 1.

Graf 1 Typický průběh závislosti hodnoty odporu na RH



Senzory používají elektrody z ušlechtilých kovů, které jsou fotoelektricky nanášeny na základní materiál, kterým je sklo nebo plast. Jiné provedení používá tenkých drátků, které jsou na základní materiál navinuty bifilárně. Čas odezvy je pro většinu odporových senzorů v řádu desítek vteřin. Rozsah hodnot odporu je typicky 1 kΩ – 100 MΩ. Pro vyhodnocování odporu se používá střídavého napětí v rozsahu 10 Hz – 30 kHz bez stejnosměrné složky, jako ochrany proti elektrolýze aktivní vrstvy senzoru. Výsledný proud je usměrněn a dále zesílen, linearizován či digitalizován A/D převodníkem. Výhodou odporových vlhkoměrů je jejich zaměnitelnost s chybou přibližně $\pm 2\%$ RH. Díky tomu není potřeba kalibrovat každý senzor jednotlivě. Příklad odporového senzoru vlhkosti ukazuje Obrázek 8.

Obrázek 8 Odporový senzor vlhkosti Samyoung[2]

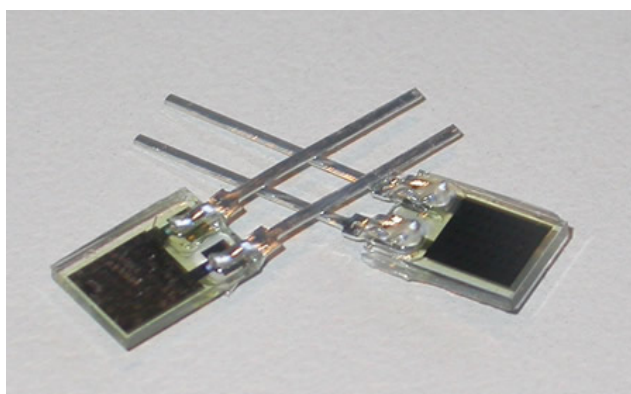


Pokud dojde k pokrytí senzoru vrstvou zkondenzované vody, případně je do vody ponořen, trvá přibližně 15 min. (při 30 %RH) než opět dojde k obnovení měřicí schopnosti. Odporové senzory vlhkosti jsou rozměrově malé, přesné (desetiny %RH), záměnné, dlouhodobě stabilní a s nízkou cenou. Mezi jejich nevýhody patří citlivost na orosení a výkyvy teploty, které je potřeba odstranit tepelnou kompenzací.

2.1.7 Kapacitní senzory vlhkosti

Také kapacitní čidla využívají změny množství vody absorbované v polymerních materiálech. Měřenou veličinou je zde změna kapacity kondenzátoru, jehož dielektrikum je tvořeno polymerem. Jedna z elektrod je pórovitá, což umožňuje okolnímu vzduchu kontakt s vlastním polymerním dielektrikem. I když je množství absorbované vody poměrně malé, jsou díky její dostatečně velké dielektrické konstantě změny kapacity měřitelné: řádově činí 0,1 % z celkové kapacity na každé % RH. Tyto senzory se vyrábějí také v provedení s integrovaným zpracováním signálu. Zde je potom výstupem místo změny kapacity přímo elektrické napětí, nebo digitální rozhraní. Kapacitní senzory se vyznačují poměrně malou závislostí údaje na teplotě, dobrou odolností vůči kondenzaci vody na jejich povrchu, dobrou odezvou v řádu desítek sekund, přesností v jednotkách %RH. Odolnost proti chemikáliím a vyšším teplotám je také dobrá. Díky malým rozměrům, robustnímu provedení a nízké ceně jsou to dnes nejrozšířenější senzory vlhkosti viz Obrázek 9.

Obrázek 9 Senzory vlhkosti Smartec [5]



Jako příklad uvedme senzor firmy Smartec s označením HS07 [5]. Jedná se o senzor relativní vlhkosti, který je tvořen kondenzátorem s kapacitou, která roste úměrně množství molekul vody, jež jsou absorbovány v dielektriku kondenzátoru. Senzor, který je vyroben technologií výroby polovodičů, je nanesen na skleněném podkladu. Vývody kondenzátoru jsou provedeny pozlacenými přívodními vodiči. Aktivní dielektrikum je pokryto polymery, které chrání jádro senzoru před polarizací, proděravěním vrstvy a zvyšují odolnost senzoru před vnějšími vlivy okolního prostředí. Díky tomu je možno nasazení tohoto senzoru i v náročných podmínkách vnějšího prostředí. Při relativní vlhkosti 55 % a teplotě 25 °C je hodnota kapacity 330 pF. Změna kapacity je rovna 0.6 pF / % relativní vlhkosti (tj. 60 pF v rozsahu 0–100 % relativní vlhkosti), 1 – 1000 kHz, do maximální teploty 85 °C.

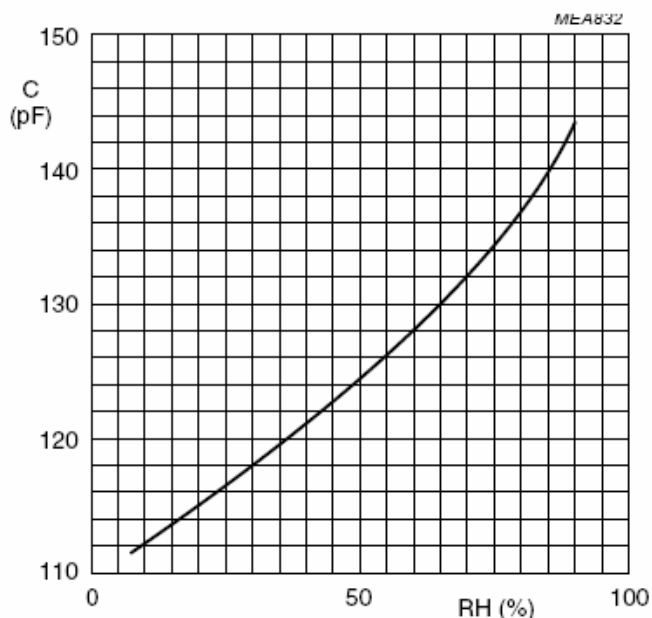
Dalším příkladem je kapacitní senzor vzdušné vlhkosti, který se skládá z bezindukční fólie, která je z obou stran pokryta vrstvou zlata. Dielektrická konstanta fólie se mění v závislosti na relativní vlhkosti okolního prostředí. Podle hodnoty kapacity je určována výsledná hodnota vlhkosti. Fólie je přitisknuta mezi dvěma pružnými kontakty a sestavena v robustním plastovém krytu, odolávající namáhání až 10 N, Obrázek 10.

Obrázek 10 Senzor vlhkosti Vishay [6]



Charakteristika (Graf 2) senzoru není ovlivněna vodou, která by případně mohla kondenzovat na fólii senzoru. Senzor by neměl být vystavován působení par chlóru a acetonu. [6]

Graf 2 Závislost kapacity na vlhkosti vzduchu



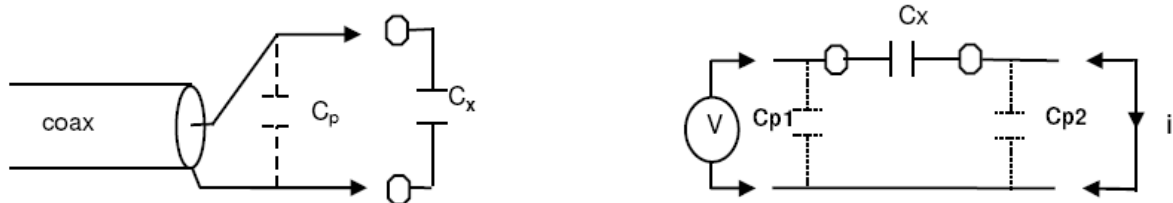
Změnu kapacity úměrnou relativní vlhkosti je možno vyhodnotit analogově nebo digitálně. Analogové vyhodnocení představuje levnou a jednoduchou variantu, která postačuje pro orientační měření. Bývá realizována jako převodník frekvence na napětí. Změna kapacity kondenzátoru způsobuje změny frekvence obvodu oscilátoru, a ta je převáděna na výstupní

stejnoseměrné napětí. Jako základ pro tento typ převodníku se nabízí obvod NE555, který je pro toto použití vhodný jak obvodovými parametry, tak příznivou cenou [5].

Při potřebě přesnějšího vyhodnocení např. pro laboratorní měření je vhodné použít číslicové vyhodnocení. Jedna z možností je použití specializovaných obvodů – Universal Transducer Interface UTI [9].

Univerzální obvod rozhraní (UTI) je převodník výstupního signálu senzoru na časový interval. Jeho podstatou je oscilátor s modulací periody. Připojení senzorů k UTI je většinou přímé a nevyžaduje žádné další podpurné elektronické obvody. Pro dosažení vysoké přesnosti měření umožňuje obvod připojení senzoru více vodiči, nebo zapojení dalších kompenzačních kondenzátorů, viz Obrázek 11.

Obrázek 11 Porovnání klasické a UTI metod měření [9]

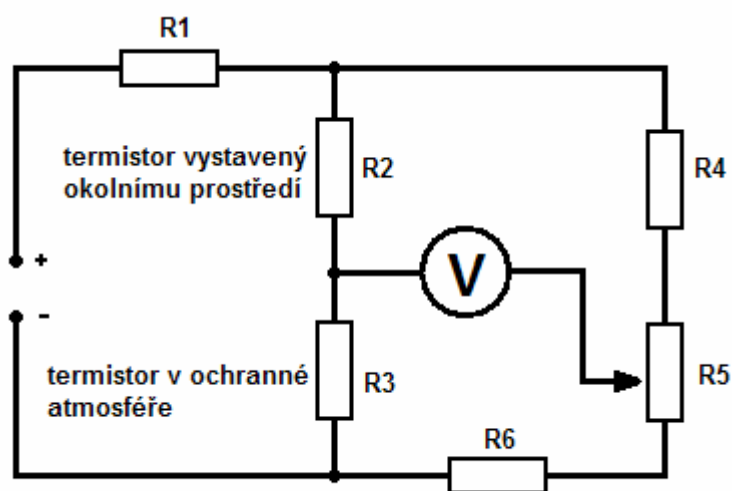


Obvody UTI mohou pracovat v několika různých pracovních režimech. Vysoké přesnosti lze dosáhnout díky použití vícevodičového připojení senzorů a použitím referenčních měřicích prvků, které mají shodné parametry jako měřený senzor. Uváděná rozlišovací schopnost je až 1 fF při měření kapacity o velikosti 2 pF s přívodními vodiči o délce několika metrů.

2.1.8 Vlhkoměr s vyhříváními termistory

Tato metoda využívá závislosti tepelné vodivosti vzduchu na jeho vlhkosti. V senzoru je umístěna dvojice stejných termistorů. Jeden z dvojice je hermeticky uzavřen v pouzdře vyplněném suchým dusíkem, druhý je umístěn tak, aby byl přístupný okolnímu prostředí. Takto do série zapojené termistory tvoří jednu větev můstku. Ve druhé větvi jsou zapojeny pevné rezistory a trimr k nastavení nulového proudu tekoucího můstkem ve vyváženém stavu, viz Obrázek 12.

Obrázek 12 Zapojení termistorů



Průchodem elektrického proudu o velikosti přibližně 100 mA se termistory zahřívají. Dosažená teplota závisí na stupni jejich ochlazování, tj. na tepelné vodivosti okolního plynu a je přímo úměrná obsahu vody v tomto plynu. Můstek se vynuluje při umístění senzoru v suchém vzduchu a přítomnost vodní páry způsobí jeho rozvážení. Signál senzoru je úměrný absolutní vlhkosti. Jeho přednostmi jsou schopnost pracovat při vysokých teplotách (až 200 °C) a chemická odolnost daná uzavřeným zapouzdřením, viz Obrázek 13.

Obrázek 13 Vyhřívání termistory [17]



2.1.9 Psychrometr

Název psychrometr pochází ze spojení slov psýchro, znamenající „studený, zmrzlý“, a métron, znamenající „měřit“.

Psychrometr pracuje na principu měření psychrometrické diference. Jedná se o rozdíl teplot mezi dvěma teploměry, tzv. suchým a mokrým. Suchý teploměr je vystaven okolnímu prostředí. Aktivní část mokrého teploměru je vlhčena vodou. Spotřeba tepla na vypařování vody z povrchu mokrého teploměru snižuje jeho teplotu a způsobuje teplotní diferenci, která je úměrná relativní vlhkosti vzduchu. Čím je měřený vzduch sušší, tím intenzivnější je vypařování vody a tím větší je i pokles teploty. Rozlišujeme dvě základní uspořádání, tzv. Augustův a Assmanův psychrometr.

U uspořádání Augustova, které ukazuje Obrázek 14, nejsou teploměry ofukovány proudem vzduchu. Toto uspořádání je z hlediska konstrukce nejjednodušší. Při měření je potřeba použít tabulky odpovídající tomuto typu vlhkoměru, v případě použití výpočtu odpovídající konstantu A .

Obrázek 14 Augustův psychrometr [24]



Pro dobrou přesnost měření musí být zajištěno, že se okolní vzduch nebude pohybovat. Pokud by docházelo k proudění okolního vzduchu, bude docházet ke změnám v rychlosti vypařování, které popisuje psychrometrická konstanta A . Tím může dojít ke zhoršení přesnosti měření.

Psychrometr v Assmanově provedení se také skládá z dvojice teploměrů, z nichž jeden je vlhčen vodu a druhý měří okolní prostředí přímo. Dále je doplněn ventilátorem umístěným v horní části přístroje, viz Obrázek 15. Ten zajišťuje rovnoměrné proudění okolního vzduchu kolem teploměrů. Minimální rychlost proudění by měla být 2,5 m/s. Vyšší rychlost proudění urychluje odezvu senzoru na měření. Běžně je doba potřebná pro ustálení naměřené hodnoty v řádu jednotek minut.

Obrázek 15 Assmanův psychrometr [24]



Obecně je přesnost každého psychrometru závislá na změnách okolního atmosférického tlaku. Při běžném měření to v praxi problém nebývá. Jistou roli to ovšem může hrát při měření ve vyšších nadmořských výškách.

Používáme-li psychrometr pro měření relativní vlhkosti vzduchu při teplotě pod 0 °C, pak voda tzv. vlhkého teploměru mrzne v led. V tom případě jsou psychrometrické konstanty pro různá proudění vzduchu menší než konstanty udávané pro vodu za stejných podmínek. Po dosažení odpovídajících hodnot je možno psychrometrem měřit i za těchto podmínek.

2.1.10 Kalibrace vlhkoměrů

Při kalibraci vlhkoměrů je zapotřebí dodržet po celou dobu kalibrace dobrou stabilitu všech parametrů, které měření ovlivňují. Jedná se o teplotu a vlhkost vzduchu, rychlost jeho proudění a tlak vzduchu. Jednou možností je umístění senzoru, který má být kalibrován, do klimaticky stabilní komory. A následné porovnání naměřených hodnot s jiným senzorem, který je brán za kalibrační standard.

Jinou možností je využít vlastností některých nasycených roztoků solí. Relativní vlhkost v prostředí nad jejich volnou hladinou je v uzavřeném prostoru a po ustálení závislá pouze na teplotě, viz Tabulka 2.

Tabulka 2 Příklady kalibračních roztoků [5], [7], [8]

Sůl (nasycený vodný roztok)		RH (%) při 20 °C	RH (%) při 25 °C
Chlorid lithný	LiCl	12.0	11.3
Chlorid hořečnatý	MgCl ₂	33.1	32.8
Dusičnan hořečnatý	Mg(NO ₃) ₂	55.0	53.0
Chlorid sodný	NaCl	75.5	75.3
Chlorid draselný	KCl	85.1	84.3
Síran draselný	K ₂ SO ₄	97.6	97.3

Nasycený roztok je vytvořen tak, že cca 3 mm vysokou vrstvou soli se pokryje dno nádoby, ve které se bude kalibrace provádět. Poté se přidává voda do té doby, dokud se sůl rozpouští. Roztoky musí být ve skleněných nádobách, které je možno neprodyšně uzavřít. Kalibrovaný senzor se umístí do prostoru nad hladinou roztoku a nádoba se uzavře. Po ustálení prostředí, které trvá asi 30 min. u rozměrově malých senzorů a 60 min. u větších, je možno odečíst naměřenou hodnotu. Po celou dobu je nutno přesně dodržovat teplotu na předepsané hodnotě.

3. Charakteristika vlastností a stavby inteligentních (smart) senzorů

3.1 Charakteristika vlastností inteligentních senzorů

Inteligentní senzor, neboli smart senzor, je prvek, který obsahuje všechny potřebné prvky pro měření, zpracování, analýzu a vhodnou reprezentaci měřené veličiny (teploty, vlhkosti, tlaku, průtoku, pH, mechanického napětí, atd.) včetně čidla v jednom kompaktním provedení, viz Obrázek 16. Tato celá soustava se nazývá měřicí řetězec.

Obrázek 16 Obecné schéma smart senzoru

Čidlo senzoru	Měřicí obvody a zesilovače	Obvody pro autokalibraci a kompenzaci	A/D převod	Uživatelské rozhraní	Komunikace
Čidlo senzoru				Aplikace	
Čidlo senzoru				Ukládání dat	

Použití smart senzorů zjednodušuje začlenění vlastních snímačů do aplikací. Např. měření teploty vyžaduje použití převodníku teploty na napětí, pro reprezentaci naměřených dat ve stupních Celsia nebo Fahrenheita. Inteligentní senzor teploty obsahuje vestavěný modul, který provádí převod a poskytuje naměřenou hodnotu teploty dostupnou ze síťového rozhraní. Kromě toho také smart senzory často obsahují rozhraní pro přímé zobrazování naměřené veličiny. Pojmeme inteligentní senzor se označují i takové senzory, jejichž část je z nějakého důvodu oddělena od zbytku měřicích elektronických obvodů. Tato situace nastává v případech, kdy samotné čidlo je vystaveno působení nepříznivých vlivů prostředí. Jedná se většinou o vysoký tlak, teplotu či chemickou agresivitu. Pokud pro toto uspořádání není zvláštní důvod, snažíme se mu vyhnout. Dlouhé přívodní vodiče totiž téměř vždy představují vznik parazitních kapacit a impedancí. To buď vede ke snížené přesnosti měření vlivem rušení na signálovém vedení, nebo ke zvýšení ekonomických nákladů vynaložených na složitější obvodová řešení potřebná pro udržení parametrů na dané výši. Vhodné může být použití optických vláken realizujících toto propojení.

3.2 Přehled vybraných typů dostupných smart senzorů vlhkosti

3.2.1 Inteligentní senzor THT – Vlhkoměr s teploměrem pro RS485

THT – Vlhkoměr s teploměrem pro RS485 je smart senzor od firmy Papouch s.r.o. Toto zařízení obsahuje digitální vlhkoměr, teploměr a výpočet teploty rosného bodu. Poskytuje výstup hodnoty vlhkosti přímo v jednotkách procent RH a teplotu ve stupních Celsia. Pro začlenění do nadřazených komunikačních systémů je vybaven rozhraním linky RS485. Více senzorů lze snadno propojit na jedno společné vedení o maximální délce až 1,2 km. Vzhledem k malé spotřebě lze vytvořit řetězec těchto zařízení propojených jedinou čtyřvodičovou sběrnici, která obsahuje jeden pár vodičů pro přenos komunikačních signálů RS485, další pár je využit pro napájení. Senzor je možno napájet nestabilizovaným napětím. Celý tento řetězec je možno připojit společně k nadřazenému systému, většinou platformy PC, zajišťující sběr a zpracování naměřených údajů. Pro komunikaci je použit vlastní protokol firmy, který je pojmenován Spinel nebo standardní MODBUS RTU. Senzor je v provedení s krytím IP65 použitelný přímo i pro venkovní prostředí, viz

Obrázek 17.

Pro měření vlhkosti je použit polymerní senzor s přesností $\pm 2\%$ RH. Ten je umístěn pod plastovou sítkou zabezpečující mechanickou ochranu. Rozsah teplot, ve kterých tento senzor může měřit je -40 až $+123,8$ °C, rozsah vlhkosti je 0% až 100% RH. Z těchto naměřených hodnot je i vypočítávána hodnota rosného bodu.

Obrázek 17 THT - Vlhkoměr s teploměrem pro RS485 [18]



3.2.2 Comet - Snímač teploty, vlhkosti s výstupem RS485, RS232

Jedná se o smart senzor firmy Comet, obsahující čidla teploty, relativní vlhkosti a atmosférického tlaku vzduchu. Naměřené veličiny jsou převáděny na číslicový sériový výstup s rozhraním RS485 nebo RS232. Výstup RS485 je galvanicky oddělen od napájecích obvodů. Tento smart senzor podporuje použití následujících komunikačních protokolů: ModBus RTU, ADAM od firmy Advantech a ARION firmy AMIT. Kromě načítání naměřených hodnot lze po sériové lince měnit některá nastavení přístroje. Pracuje vždy v podřízeném režimu, což znamená, že naměřené hodnoty odesílá pouze na dotaz nadřízeného systému. Každý smart senzor má přiřazenou adresu v rozsahu 1 až 255, kterou se prokazuje při komunikaci.

Toto zařízení přímo obsahuje velký dvouřádkový displej pro zobrazování teploty, vlhkosti, tlaku a dalších vypočtených hodnot v místě měření. V nastavení lze displej vypnout. Pro snímání vlhkosti je použito polymerní čidlo vlhkosti, které je odolné vůči kondenzaci vody na jeho povrchu. Předpokládá se měření v neagresivním prostředí. Volitelně je dostupné i čidlo vlhkosti a teploty pro prostředí s tlakem do 20 barů. Z naměřených údajů jsou vypočítávány další veličiny: teplota rosného bodu, absolutní vlhkost, měrná vlhkost, směšovací poměr a specifická entalpie. Pro napájení je nutno použít stejnosměrné napětí v rozsahu 9 až 30 V, příkon smart senzorů je přibližně 0,5 W. Maximální rozsah měřených teplot je -30 až +80 °C. Snímač je v provedení s krytím IP65, pro použití ve venkovním prostředí, viz Obrázek 18.

Obrázek 18 Smart senzor Comet



3.2.3 Snímač vlhkosti Profess

Snímače vlhkosti Profess jsou smart senzory umožňující měření a zobrazování hodnot teploty a vlhkosti. Relativní vlhkost je zde měřena pomocí polymerového kondenzátoru, jehož kapacita se v závislosti na vlhkosti vzduchu mění. Ze změny kapacity je vypočtena hodnota relativní vlhkosti a převedena na signál proudové smyčky v rozsahu proudu 4–20 mA. Čidla teploty a vlhkosti jsou umístěny v ochranné trubce, vybavené filtrem pro ochranu proti vnějším nečistotám, viz Obrázek 19.

Na těle senzoru je umístěn čtyřmístný LCD displej, na kterém je zobrazována měřená hodnota relativní vlhkosti a hodnota teploty, kterou lze zobrazit jak ve °C tak ve °F. Tlačítka vedle displeje lze vyvolat údaje pro maximální a minimální hodnoty. Pro napájení je vyžadován zdroj stejnosměrného napětí 12 až 30 V. Jmenovitý rozsah je 0–100 % RH, s odolností proti kondenzaci. Linearita je v tomto rozsahu ± 1 % RH, hystereze ± 2 % RH a doba reakce 45 s. Teplotní rozsah pro měření je -40 až 100 °C. Výstupem je proudová smyčka s úrovněmi 4–20 mA.

Obrázek 19 Smart senzor Profess [20]



3.2.4 Dewtron smart senzor

Dewtron je smart senzor firmy Pertec. Je navržen pro kombinované měření teploty a relativní vlhkosti. Rozsah měření je 0–100 % RH s rozlišovací schopností 0,03 % pro relativní vlhkost a -40 až +120 °C s rozlišovací schopností 0,01 °C. Přesnost je ± 2 % RH v rozsahu 10–90 % RH. Pro teplotu je přesnost $\pm 0,4$ °C v rozsahu 5–40 °C a ± 1 °C v rozsahu -20–70 °C. Vlastní čidlo teploty a vlhkosti je odděleně od měřicí elektroniky, se kterou je spojeno 1 nebo 2 m dlouhým vodičem. Odezva na měření je výrobcem udávána menší než 4 s. Pro napájení je vyžadován zdroj se stejnosměrným napětím 10–36 V nebo střídavým 8–26 V. Příkon zařízení je 1,5 W. Zařízení je v provedení s krytím IP67, viz Obrázek 20.

Výstupem je digitální rozhraní RS232 a RS485 a dvojice relé zatížitelných 2 A/120 V s programovatelnou úrovní sepnutí při dosažení mezní hodnoty.

Obrázek 20 Smart senzor Dewtron [21]



Konzultace k těmto smart senzorům proběhla v průběhu veletrhů Ampér 2007 a 2009. Některé údaje byly dodatečně doplněny písemně.

3.2.5 Smart senzor EE31

Smart senzor EE31 slouží k měření relativní vlhkosti a teploty, jeho výstup lze připojit do sítě standardu Ethernet. Z naměřených údajů počítá i další charakteristiky vlhkosti vzduchu, jako je absolutní vlhkost, teplota orosení, teplota ojinění, teplota mokrého teploměru, specifická entalpie, tlak vodních par a směšovací poměr. Měřicí rozsah pro relativní vlhkost je 0–100 % RH, pro teplotu -40–180 °C s dosaženou přesností ± 2 % RH pro relativní vlhkost a $\pm 0,2$ °C pro měření teploty. Výstupem jsou jednak analogové linky 0–5 V, 0–10 V, 0–20 mA, 4–20 mA, dále digitální sériové rozhraní: RS232C, RS485 a jednak rozhraní Ethernet. Zařízení vyžaduje pro napájení napětí 24 V AC nebo DC. Pro různá prostředí je k dispozici několik variant, odlišujících se provedením snímače. Příkladem je provedení pro montáž na zeď či varianta pro měření v prostředích s tlakem až 15 bar při použití externích čidel, jak je uvádí Obrázek 21.

Obrázek 21 Smart senzor EE31 [22]



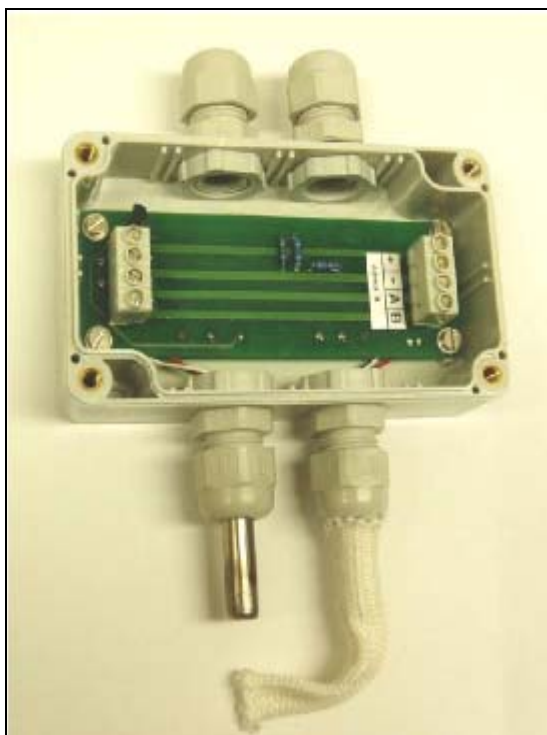
Díky rozhraní Ethernet může být tento smart senzor snadno začleněn do podnikové datové infrastruktury. Pro vzdálené sledování, ukládání a analýzu naměřených hodnot lze využít dodávaný SW nebo použít vlastních databázových nadstaveb.

3.2.6 Psychrometr DSL12

Přístroj DSL12 je psychrometrický senzor vlhkosti, připojitelný k nadřazenému systému. Je vhodný pro použití v průmyslovém prostředí. Tento psychrometr převádí teplotu ze dvou teplotních čidel v rozsahu -55 až 125 °C. Jako vlastní senzory teploty jsou zde použity obvody DALAS DS18B2. Naměřené hodnoty jsou převedeny na číslo v ASCII formátu, tj. dále se přenáší ve tvaru čistě textového řetěze. K nadřazenému systému je připojen pomocí sběrnice RS485. Přenosovou rychlost lze měnit v rozsahu 600–19200 Bd. Maximální délka sběrnice je 1200 m. Propojení je provedeno čtyřvodičovým kabelem, ve kterém je umístěn pár vodičů pro datovou linku a další pár pro napájecí napětí, které může být v rozsahu 7–29 V DC, nestabilizované s ochranou proti přepólování. Odběr proudu je 3 mA.

Při připojení více senzorů je každý z nich adresovatelný až do počtu 31 modulů. Měření teploty je prováděno s rozlišením 0,01 °C, při přesnosti $\pm 0,5$ °C v rozsahu -10 až +85 °C, jinde ± 2 °C. Přesnost získané relativní vlhkosti je 5 %. Po zahájení měření je třeba vyčkat přibližně 20 min. na dokonalé navlhčení knotu a ustálení teplot. Krytí přístroje je IP65, viz Obrázek 22.

Obrázek 22 Psychrometrický senzor DSL 12 [26]



3.2.7 Průmyslový psychrometr Hygrophil h

Tento průmyslový psychrometr (viz Obrázek 23) je určen pro kontinuální měření vlhkosti do teplot +1000 °C. Je navržen pro použití v agresivním prostředí, jako např. olejové páry, plyny rozpustné ve vodě, rozpouštědla, agresivní kyseliny či prach. Obsahuje vyhřívaný filtr měřeného plynu, který je periodicky regenerován pomocí stlačeného vzduchu, viz [25].

Kontinuální měření rosného bodu lze využít pro optimalizaci teploty odpadních plynů např. v komínových systémech tak, aby se předešlo nežádoucí kondenzaci, způsobující korozi, špatné funkci odlučovačů popílku, znehodnocení textilních filtrů apod.

Předpokládané oblasti použití jsou sušící zařízení, měření emisí v teplárnách, elektrárnách, spalovnách odpadů apod.

Obrázek 23 Průmyslový psychrometr [25]



4. Charakteristika vlastností embedded mikrokontrolerů

4.1 Přehled vybraných typů mikroprocesorů užívaných pro vestavbu do senzorů a přístrojů

V současné době se pro použití v podobných aplikacích používají některé z následujících řad mikroprocesorů: PIC, AVR, ARM. První dvě skupiny tvoří 8bitové RISCové mikroprocesory. Procesory PIC používají méně instrukcí a mají odlišně uspořádanou paměť a přístup k ní. Většina z nich jsou 8bitové, jsou však i verze 16 a 32bitové. Z hlediska použitelnosti nejsou mezi těmito skupinami mikroprocesorů větší rozdíly. Stejně tak v dalších parametrech, jako je vybavenost, obvody rozhraní (A/D převodníky, sériové rozhraní), rychlost, spotřeba, dostupnost podkladů pro práci s nimi, dosahují srovnatelných parametrů. Přiklonění se k jedné či druhé skupině je tak většinou dáno zvykem vývojáře.

ARM je oproti prvním dvěma skupinám označení 32bitových RISCových mikroprocesorů. Tyto procesory jsou výkonnější, vybavené větším množstvím paměti pro program i data. Díky energeticky úspornému návrhu jsou procesory ARM používány hlavně v odvětví mobilních aplikací spotřební elektroniky, kde je požadován vysoký výpočetní výkon a malá spotřeba.

Pro vyhodnocování signálů od některých speciálních čidel používají speciální zákaznické obvody. Nemusí se vždy jednat o mikroprocesory, spíše jde o obvody, které přizpůsobují jednotlivá rozhraní a připravují data pro další zpracování. Příkladem mohou být např. obvody UTI firmy Smartec, podrobněji popsané v kapitole 2.1.7.

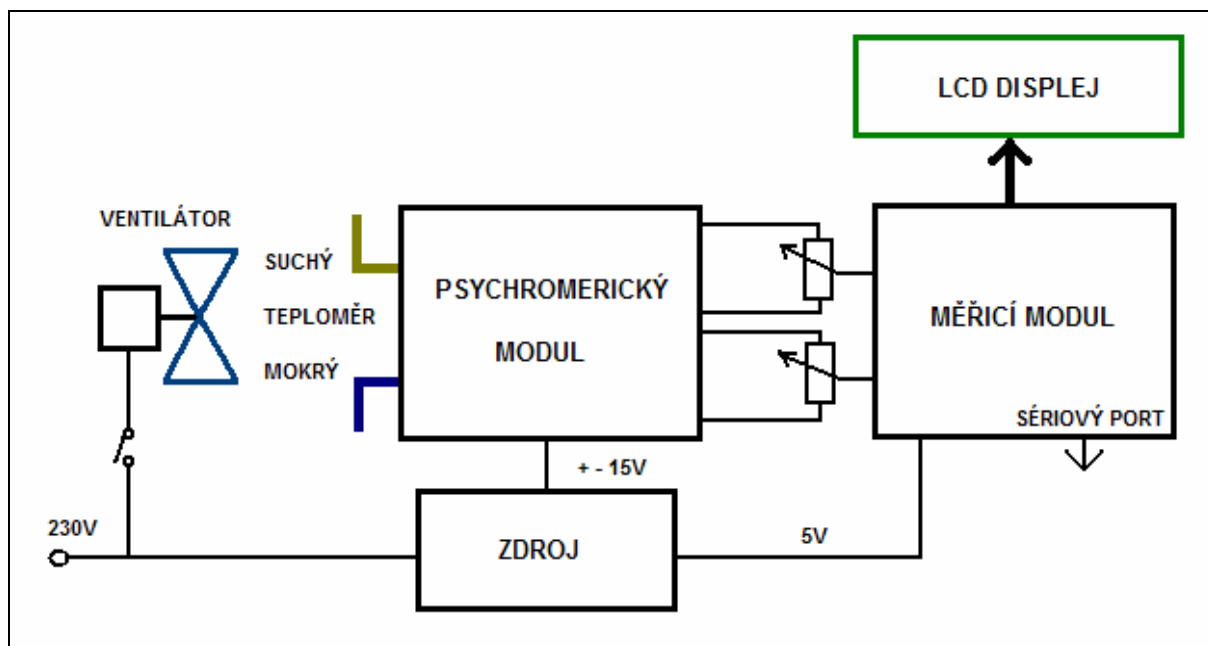
Při návrhu inteligentního senzoru relativní vlhkosti vzduchu jsem použil mikroprocesor ATmega8. Zdůvodnění tohoto výběru je v kapitole 5.3.

5. Návrh a realizace řídicí části technologického psychrometrického vlhkoměru vzduchu

5.1 Obecná koncepce

Návrh smart senzoru používá jako základ senzor vlhkosti RTU-01-T, jehož výrobcem byl Státní statek Přeštice – přidružená výroba Stod. Tento modul je doplněn o zobrazování a předávání naměřených hodnot vlhkosti pro další zpracování. Napájecí napětí tohoto modulu je ± 15 V. Teplotní rozsah je 0–102,4 °C. Modul vlhkoměru se skládá z několika částí. Teplota je snímána dvěma teploměry. Mokrý teploměr je vlhčen bavlněným knotem, který jej spojuje se zásobní nádobkou na vodu. Suchý teploměr je vyveden z pouzdra přístroje na opačné straně. Obě dvě teploměrná čidla jsou vystavena rovnoměrnému proudění vzduchu z ventilátoru, viz Obrázek 24.

Obrázek 24 Blokové schéma smart senzoru



Výstupem obou teploměrů je údaj převedený na proudové smyčky 4–20 mA, které budeme dále vyhodnocovat. Proudové smyčky jsou přijímány dvěma odporovými trimry s hodnotou 250 Ω . Paralelně k nim jsou připojeny vstupy A/D převodníku. Mikroprocesor cyklicky obsluhuje oba převodníky a z naměřených hodnot napětí vypočítává teploty a vlhkost. Hodnoty teploty naměřené suchým a mokrým teploměrem zobrazuje na LCD displeji, stejně tak i vypočítanou hodnotu relativní vlhkosti.

5.2 Návrh napájecího zdroje

Psychrometrický senzor vlhkosti RTU-01-T vyžaduje pro svoji vlastní funkci stejnosměrné symetrické stabilizované napětí ± 15 V. Elektronické vyhodnocovací obvody, obvod displeje a mikroprocesoru pracují se stejnosměrným napájecím napětím 5 V. Na zdroji je dostupné i výstupní napětí 3,3 V pro připojení modulu pro komunikaci přes sériové rozhraní.

Zdroj je napájen ze síťového transformátoru o příkonu 5 VA, jehož sekundární obvod je tvořen dvěma vinutími. Napětí je dvojcestně usměrněno a přivedeno na filtrační kondenzátory o kapacitě 2200 μ F/25 V. Dále jsou připojeny v kladné i záporné větvi monolitické stabilizátory s pevným výstupním napětím ± 15 V. Z napětí +15 V je získáno dalším monolitickým usměrňovačem stabilizované napětí +5 V, a z tohoto napětí dále opět monolitickým usměrňovačem 3,3 V. Proudová zatížitelnost lineárních stabilizátorů napětí pro ± 15 V a +5 V je 1 A. Takováto hodnota proudu může být dosažena pouze při impulsním charakteru odběru, kdy je potřebný náboj poskytován z filtračních kondenzátorů. Pro ustálenou hodnotu proudu je omezující celkový příkon transformátoru, který je ovšem dostatečný, jak je uvedeno dále.

Odběr RTU-01-T je v kladné větvi 36,5 mA a v záporné 12,34 mA.

5.3 Výběr mikroprocesoru

Při výběru vhodného mikroprocesoru bylo klíčové vhodné přizpůsobení vstupů A/D převodníku mikroprocesoru výstupním měřicím obvodům psychrometrického senzoru RTU-01-T. Vzhledem k absenci dokumentace k RTU-01-T jsem předpokládal, že obě výstupní proudové smyčky jsou striktně oddělené od zemního potenciálu, tj. že jsou tzv. plovoucí. Proto jsem k jejich korektnímu měření vybíral A/D převodník s možností diferenciálního měření. Ten jsem objevil u mikroprocesoru ATmega32. Provedení v pouzdře TQFP jsem nemohl použít, pro nedostatečné technické vybavení (pájecí horkovzdušná stanice). I případný návrh plošného spoje by vycházel složitější. Proto jsem zvolil provedení v pouzdře PDIP. Po sestavení a zkoušce měření se ovšem ukázalo, že možnost diferenciálního měření mezi jednotlivými vstupy A/D převodníku existuje pouze u provedení v pouzdře TQFP. Po zjištění, že plovoucí vstupy nejsou nutnou podmínkou pro správné měření, jsem se vrátil k mikroprocesoru ATmega8, který je dostatečný jak počtem a provedením vstupů, tak i dosaženou rychlostí. Je použit interní oscilátor s frekvencí 8 MHz.

5.4 Metody zjištění relativní vlhkosti pomocí teplotní difference

5.4.1 Z psychrometrické tabulky

Psychrometrické tabulky se nejčastěji používají v meteorologii pro vyhodnocování psychrometrických měření při prostém odečtu naměřených hodnot z teploměrů. Jsou sestaveny na základě vztahů uvedených v kapitole 5.4.2. Jejich výhodou je snadné použití, které je vhodné zvláště při odečítání několika opakujících se hodnot. Tato metoda předpokládá, že atmosférický tlak se v místě měření výrazně nemění.

Stejným principem je možno provádět zpracování i při použití mikroprocesorů. Ve vnitřní paměti mikroprocesoru jsou uloženy hodnoty tvořící tuto tabulku. Naměřené hodnoty teplot suchého a mokrého teploměru jsou určujícími klíči pro výběr odpovídající hodnoty relativní vlhkosti. V případě, že pro danou kombinaci hodnot mokré a suché teploty není požadovaná hodnota uložena, získá se aproximací dvou sousedních hodnot. Nároky na velikost obsazené paměti závisí na tom, jak jemné dělení teplot se použije při sestavení tabulky.

5.4.2 Výpočtem

Pokud se dá očekávat, že během měření může docházet k velkým změnám atmosférického tlaku, např. přetlak ve vzduchotechnickém kanále, ve kterém je měření prováděno, je vhodnější výpočet přímo podle Sprungova vztahu. Totéž platí, pokud je kladen velký důraz na přesnost měření. Tento vztah je uveden např. v [22].

$$\varphi = \left[\frac{p''_{pm}}{p''_p} - \frac{A \cdot p_b}{p''_p} \cdot (t_0 - t_m) \right] \cdot 100$$

Kde:

Φ je relativní vlhkost v % RH

p_b je barometrický tlak vzduchu v Pa

t_0 je teplota suchého teploměru

t_m je teplota mokrého teploměru

p''_{pm} je tlak syté vodní páry odpovídající teplotě t_m

p''_p je tlak syté vodní páry odpovídající teplotě t_0

A je psychrometrická konstanta (0.000662)

Další možností je pro zjednodušení použít aproximaci, která umožňuje převod mezi teplotou suchého a mokrého teploměru pro získání relativní vlhkosti. Tato aproximace poskytuje dobré výsledky, pokud jsou teploty měřeny s přesností minimálně ± 1 °C a relativní vlhkost se pohybuje nad 50 % RH. [22]

$$tm = ts - \frac{(100 - \varphi)}{5}$$

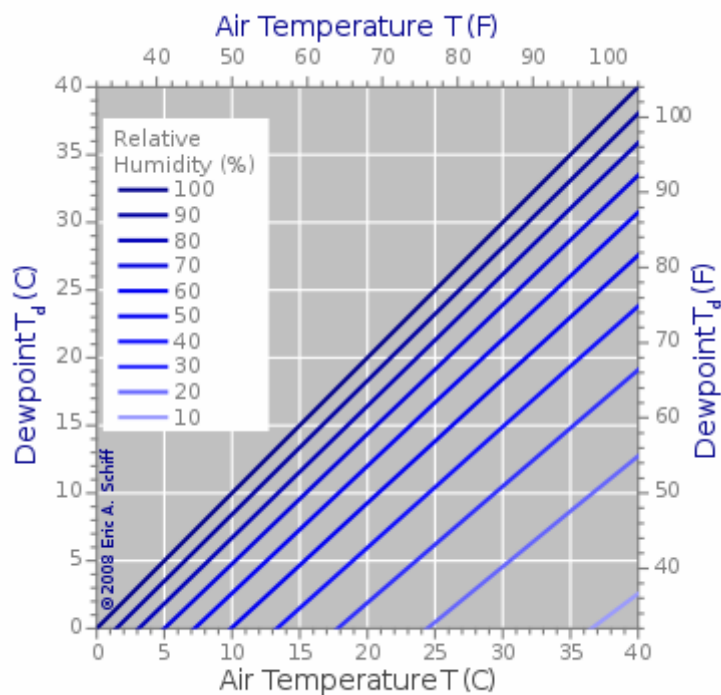
neboli:

$$\varphi = 100 - 5(ts - tm)$$

kde: φ je relativní vlhkost v % RH, tm a ts teplota mokrého a suchého teploměru ve °C.

To odpovídá pravidlu, že na každý stupeň teplotní difference klesá relativní vlhkost o 5 %, od počáteční hodnoty 100 % RH. Tento výpočet je použit pro výpočet v tomto návrhu. Grafické znázornění je na Graf 6 podle [23].

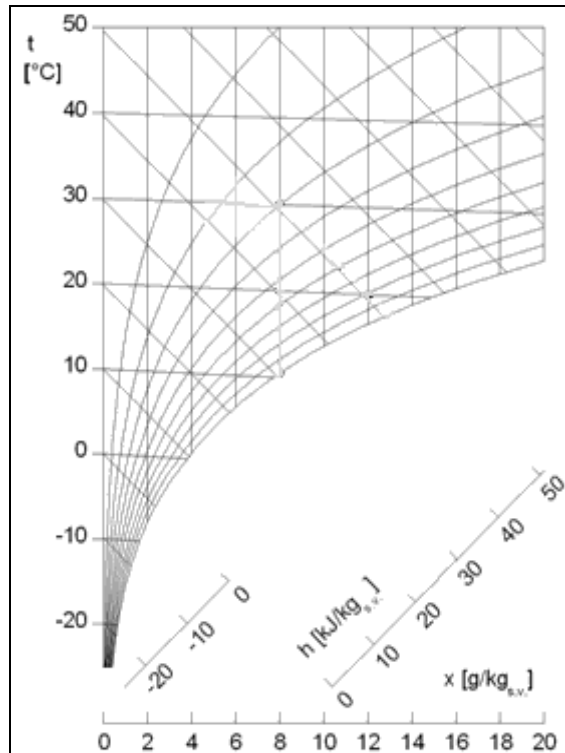
Graf 3 reprezentace vztahu závislosti RH



5.4.3 Z grafu

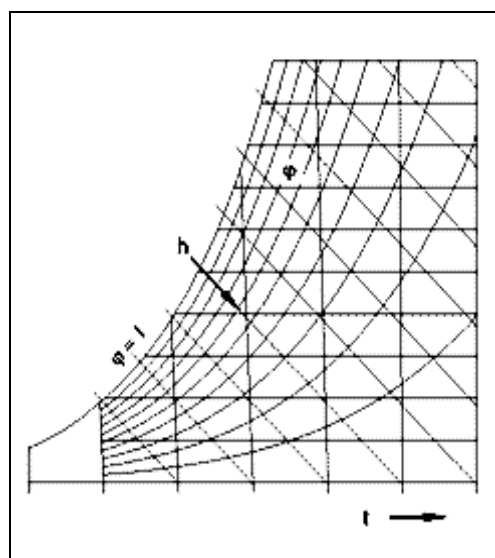
Grafickou interpretací závislosti mezi relativní vlhkostí vzduchu, teplotami suchého a mokrého teploměru a atmosférickým tlakem lze vyčíst z Molliérova diagramu viz Graf 4. Popisuje vzájemnou závislost těchto veličin při izobarických dějích.

Graf 4 Molliéřův diagram



Také se používá modifikace tohoto diagramu, která se nazývá psychrometrický diagram, viz Graf 5. Z Molliérova diagramu vznikne překlopením a otočením.

Graf 5 Psychrometrický diagram

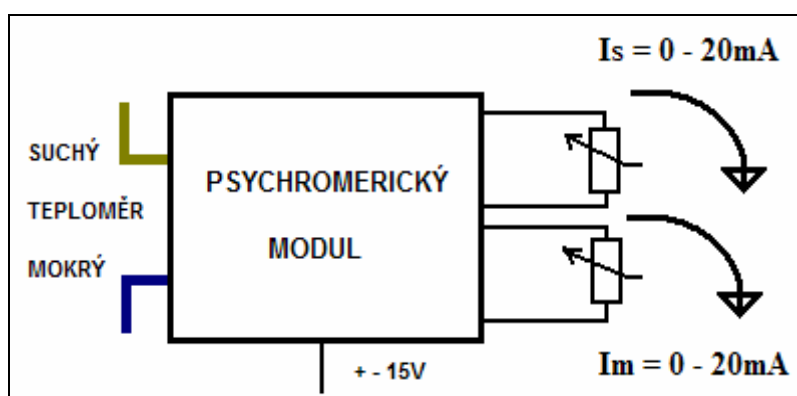


5.5 Popis měření

Do každého z obvodu proudové smyčky, viz Obrázek 25, pro výstup naměřené hodnoty suchého i mokrého teploměru je sériově vřazen odporový trimr s hodnotou 250 Ω . Tím je procházející proud převeden na hodnotu napětí, kterou přivádíme na vstup A/D převodníku. Hodnota proudu 0 mA odpovídá hodnotě 0 V. Maximální velikost proudu v proudové smyčce je rovna 20 mA. Pro zvolenou hodnotu odporového trimru 250 Ω je maximální hodnota napětí na odporovém trimru rovna:

$$0,02 \cdot 250 = 5[A, \Omega, V]$$

Obrázek 25 Výstupní proudové smyčky



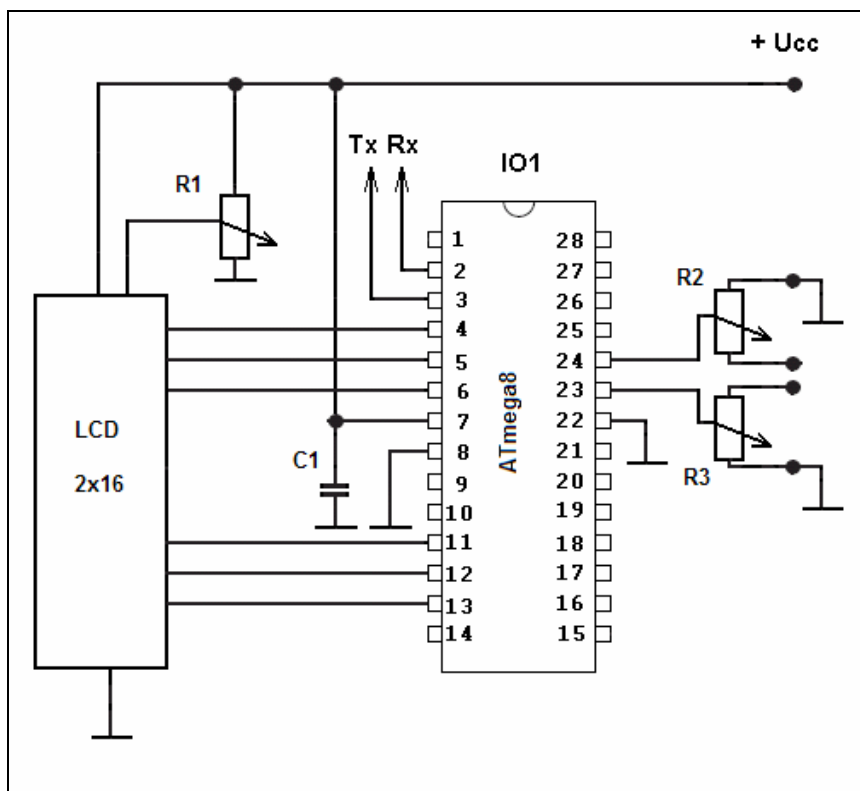
5.5.1 měření A/D převodníkem:

Použitý A/D převodník interpretuje měřené napětí v rozsahu 0–5 V jako číslo o hodnotě 0–1024. Kde 0 V odpovídá hodnotě 0 a 5 V odpovídá hodnotě 1024. Podle kalibračního údaje na štítku senzoru proud 20 mA odpovídá teplotě 102,4 °C. Z toho vyplývá, že změna o jeden dílek z převodníku odpovídá změně 0,1 °C skutečné teploty. Takto převedená hodnota slouží pro další výpočty.

5.5.2 Schéma zapojení

Schéma zapojení je uvedeno na Obrázek 26. Odporový trimr R1 slouží k nastavení kontrastu u LCD displeje. Doporučená hodnota je $5\text{k}\Omega$. Kondenzátor C1=100nF je filtrační, zapojený co nejbližše vývodům mikroprocesoru. Odporové trimry R2 a R3 slouží ke kalibraci měření teploty.

Obrázek 26 Schéma zapojení



5.5.3 Seznam součástek

- IO1 ATmega 8
- LCD LCD displej 2x16 znaků, MC1602E-SYL/H
- R1 odporový trimr $5\text{k}\Omega$
- R2,R3 odporový trimr 250Ω
- C1 100nF

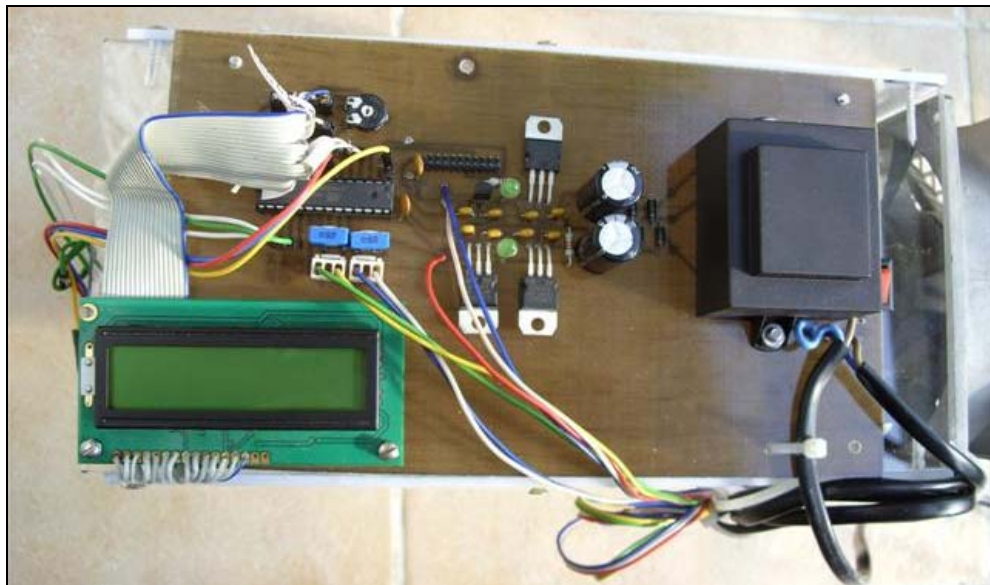
5.5.4 Mechanické provedení

Výkres desky plošného spoje je v příloze 11.1 a výkres osazení součástek je uveden v příloze 11.2.

5.5.5 Mechanické provedení

Elektronika senzoru vlhkosti je umístěna na jednostranné desce plošného spoje. Na stejné desce je současně umístěn i síťový transformátor a stabilizovaný zdroj. Vlastní kryt psychrometru je zhotoven z plastových desek. Takto chrání čidla teploty před tepelným zářením z okolí a zároveň usměrňují proud vzduchu k ventilátoru. Na horní straně tohoto krytu je umístěna deska s elektronikou, viz Obrázek 27 . Toto uspořádání je vhodné spíše pro vývoj zařízení. Při nasazení v praxi by bylo vhodnější desku s elektronikou umístit zvlášť, do vlastní krabičky s dostatečným krytím, minimálně IP54.

Obrázek 27 Deska elektroniky senzoru



6. Návrh a odladění software mikroprocesoru

Software popisující funkci mikroprocesoru se skládá z bloků popsaných v následujících kapitolách. Program běží v nekonečné smyčce, ve které sekvenčně vykonává následující funkce.

6.1 Obsluha A/D převodníků

Prvním je obsluha A/D převodníků, které vyhodnocují vstupní napětí. Jsou využity dva ze šesti dostupných multiplexovaných vstupů A/D převodníku. Následně je spuštěn proces měření. Po čase potřebném pro převod je přečtena hodnota napětí a uložena do proměnné v programu.

6.2 Prováděné výpočty

V bloku výpočtů se naměřené hodnoty teploty převádí na desetinné číslo a zároveň se zde provádí výpočet hodnoty relativní vlhkosti.

Postup výpočtu teploty suchého teploměru:

$$TS4 = t_sucha / 1000$$

$$t_sucha = t_sucha \text{ MOD } 1000$$

$$TS3 = t_sucha / 100$$

$$t_sucha = t_sucha \text{ MOD } 100$$

$$TS2 = t_sucha / 10$$

$$t_sucha = t_sucha \text{ MOD } 10$$

$$TS1 = t_sucha$$

Postup pro oddělení jednotlivých desetinných řádů spočívá v dělení základem tohoto řádu. Tím je získán počet jednotek daného řádu. Následně se provede celočíselné dělení a se zjištěným zbytkem po dělení se postup opakuje pro základ o desítku nižší.

Postup výpočtu teploty mokrého teploměru:

$$TM4 = t_mokra / 1000$$

$$t_mokra = t_mokra \text{ MOD } 1000$$

$$TM3 = t_mokra / 100$$

$$t_mokra = t_mokra \text{ MOD } 100$$

$$TM2 = t_mokra / 10$$

$t_mokra = t_mokra \text{ MOD } 10$

$TM1 = t_mokra$

Postup výpočtu relativní vlhkosti:

$vlhkost = 1000 - 8 * (t_sucha - t_mokra)$

$vlhkost3 = vlhkost / 100$

$vlhkost = vlhkost \text{ MOD } 100$

$vlhkost2 = vlhkost / 10$

$vlhkost = vlhkost \text{ MOD } 10$

$vlhkost1 = vlhkost$

6.3 Obsluha LCD displeje

Zobrazování naměřených hodnot na LCD displeji 2x16 znaků obsluhuje podprogram pro práci s displejem. Nejdříve se provádí inicializace displeje. Potom je vypsán ten text, který se během celého měření nemění. Jedná se o označení měřených veličin a jejich jednotky. Poté je kurzor umístěn na pozici, na kterou má být vypsána příslušná proměnná, např. ta, která označuje desítky stupňů u teploty suchého teploměru. Dále je kurzor přesunut na další pozici a vypsána další proměnná – jednotky stupňů a stejným způsobem i desetiny stupně. Takto se postupuje i při obsluze druhého řádku displeje.

Zvolené uspořádání rozložení textu na displeji vzniklo ze snahy umístit na displej všechny tři zobrazované veličiny. Proto např. znak „%“ relativní vlhkosti nebylo možno umístit bezprostředně za hodnotu zobrazované veličiny.

6.4 Obsluha sériové linky

V tomto posledním bloku programu je vypočítaná hodnota odeslána na výstup sériové linky. Parametry přenosu jsou následující: přenosová rychlost 9600 kb/s, řízení toku žádné – asynchronní přenos, start bit, znaky posílány jako ASCII. V tomto modu provozu není vyžadováno potvrzování přijímané datové zprávy.

6.5 Programování mikroprocesoru

Pro programování mikroprocesoru bylo využito sériové rozhraní ISP, které umožňuje jejich programování, i když jsou přímo zapojeny v navrhovaném obvodu. Díky tomu je ladění aplikace poměrně operativní a provedené změny je možno hned testovat. Pro tvorbu programu bylo využito prostředí Flowcode, doplněné příkazy jazyka C. Pro nahrávání programu do paměti mikroprocesoru byla použita aplikace ASIX UP. Zdrojový kód programu je uveden v příloze 11.4.

Jako programátor bylo použito zařízení PRESTO, od firmy ASIX.

7. Testování vlastností návrhu

7.1 Dosažené parametry

Pracovní rozsahy:

Rozsah teplot: 0–102,4 °C

Rozsah vlhkosti: 0–99,9 % RH

Spotřeba zařízení:

Proud odebíraný vyhodnocovací elektronikou a psychrometrickým senzorem: 48 mA

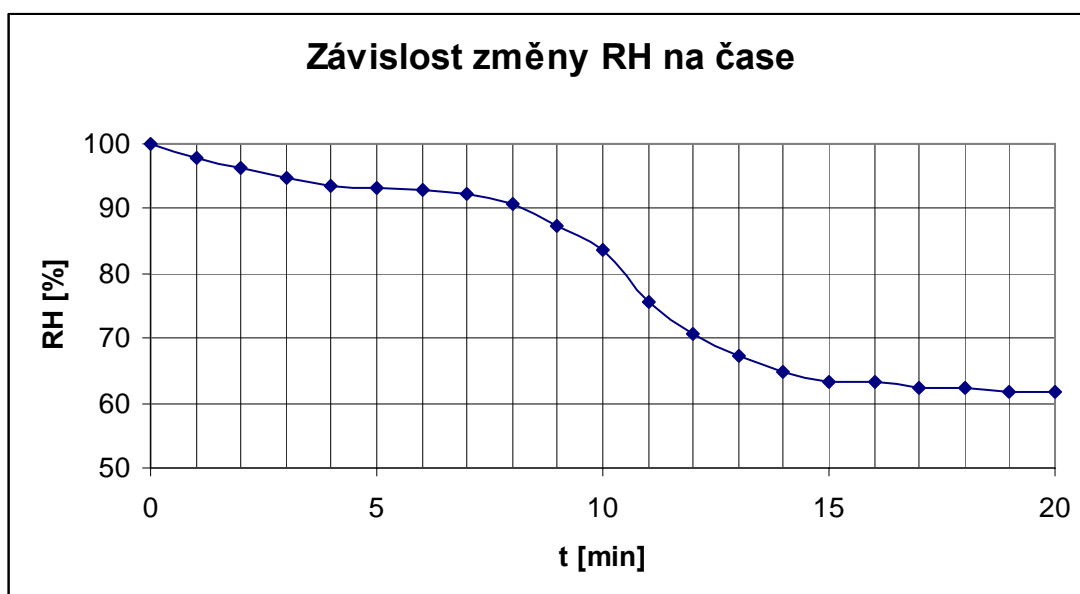
Proud odebíraný ventilátorem senzoru: 112 mA

Spotřeba vody činí 53 ml/den při 64 % RH a průměrné teplotě 21 °C.

7.2 Rychlost změny měření

Odezvu senzoru na změnu vlhkosti vzduchu zachycuje Graf 6. Z počátku je změna jen velmi pozvolná. Následně, patrně vlivem zvýšeného odparu vyvolaného prouděním vzduchu, dochází k rychlejšímu sestupu naměřené vlhkosti a k ustálení dochází přibližně za 15-20 min. Při kontinuálním měření to není na překážku, pokud není vyžadováno hlídání rychlých a náhlých výkyvů vlhkosti v měřeném prostředí. Pro orientační měření je takovýto dosažený čas velmi dlouhý a činí měření velmi náročným, ne-li nepoužitelným.

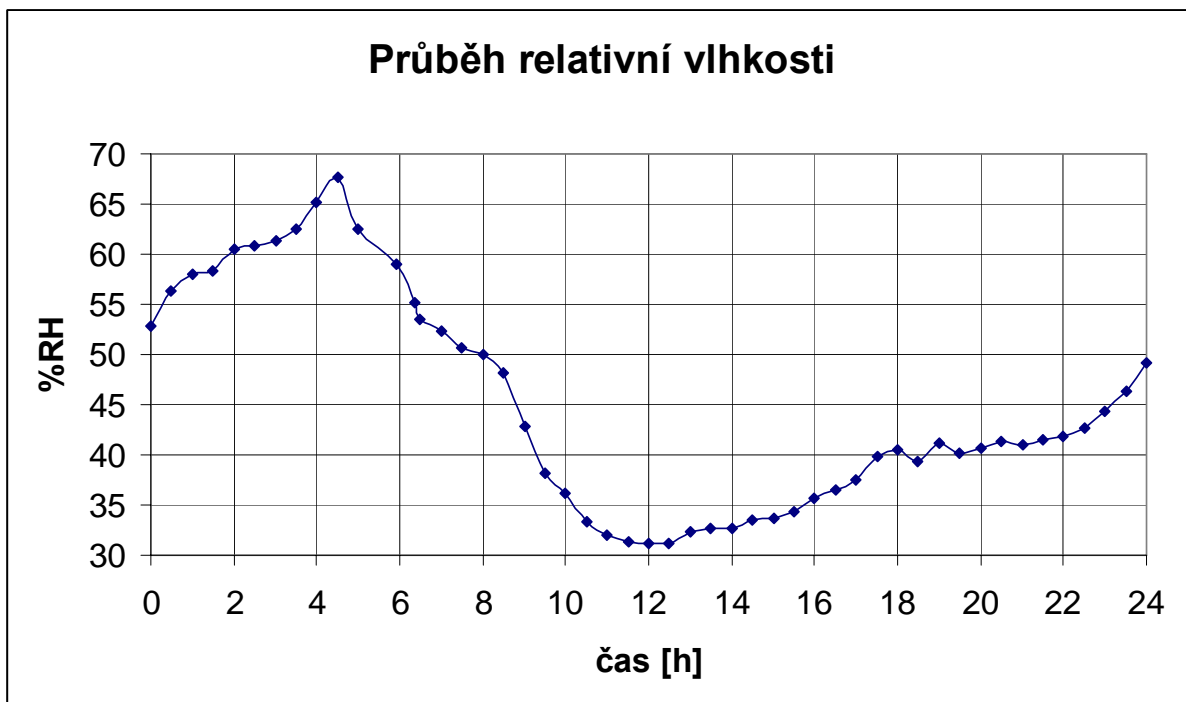
Graf 6 Závislost změny RH na čase



7.3 Měření vlhkosti

Bylo provedeno měření relativní vlhkosti vzduchu. Grafické zpracování průběhu naměřených hodnot ukazuje Graf 7.

Graf 7 Průběh relativní vlhkosti



Při dlouhodobém měření se neprojevuje nijak rušivě poměrně dlouhá odezva senzoru na změnu relativní vlhkosti. Pro nasazení při kontinuálním měření je možno senzor použít s dobrými výsledky. Více je o vlastnostech senzoru popsáno v kapitole 8.

8. Zhodnocení

8.1 Zhodnocení vlastností návrhu

Psychrometr byl navržen podle požadavků uvedených v zadání. Byla zjištěna rozhraní psychrometrického modulu RTU-01-T a navržen odpovídající zdroj napětí, nutný pro správnou funkci modulu. Dále byl zvolen mikroprocesor ATmega8. Disponuje A/D převodníkem s potřebným počtem vstupů a rychlostí pro tento typ vstupních signálů. Má dostatečný výkon pro provádění potřebných výpočtů a sériové rozhraní pro komunikaci s okolními systémy.

Rychlost aktualizace hodnot na LCD displeji je 5 měření za sekundu. Doba ustálení hodnoty při změně vlhkosti je kolem 10min., jak ukazuje Graf 6. Z tohoto měření vyplývá i necitlivost na krátké výkyvy relativní vlhkosti.

Jak je popsáno v kapitole 2.1.9, potřebuje tento typ psychrometru pro svoji funkci mít zaručené proudění okolního vzduchu rychlostí větší, než 2,5m/s. U tohoto psychrometrického senzoru je to zajištěno ventilátorem napájeným síťovým napětím. Zde použitý ventilátor je poměrně silným zdrojem hluku.

Jeho odběr 112mA tj. přibližně 25W je při dlouhodobém provozu příčinou vysoké spotřeby. Odběr vlastního psychrometru s vyhodnocovacím modulem je méně jak poloviční. Řešením je buď tento ventilátor nahradit novějším s nižší spotřebou a hlukem, nebo zajistit proudění vzduchu psychrometrem jiným způsobem.

Nabízí se vhodné umístění do proudu vzduchu v kanálu vzduchotechniky. Zde je ovšem tlak různý od atmosférického, tudíž je potřeba korigovat hodnoty na tlak v místě měření. Pro běžné použití je možnost ventilátor periodicky spínat před každým měřením, což musí být provedeno s dostatečným předstihem.

8.2 Ekonomické zhodnocení návrhu

Základem celého zařízení je modul RTU-01-T. Podle získaných informací již tento modul není na trhu dostupný, jeho pořizovací cena tedy není známa. Lze stanovit cenu vyhodnocovacího modulu. Ta by se skládala z ceny součástek, ceny vhodného krycího boxu, ceny práce výroby a vývoje, případně dalších nákladů jako např. logistické náklady na obaly, přepravu a marketing. To vše doplněno o požadovaný zisk.

Přesně lze stanovit pouze cenu použitých součástek, uvedenou v kapitole 8.3 a přibližně odhadnout počet hodin na vývoj a výrobu prototypu. Ta je odhadována na minimálně 300h.

8.3 Soupis nákladů součástek

Tabulka 3 Ceny součástek

Součástka	Cena za kus [Kč/ks.]	Počet ks.	Cena [Kč]
ATmega8	34	1	34
LCD displej, 2x16	125	1	125
LM7805	8	1	8
LM7815	6	1	6
LM7915	8	1	8
Trimr 250Ω	10	2	20
Trimr 5k	10	1	10
Transformátor	95	1	95
Kondenzátor 2200uF	15	2	30
Kondenzátor 100nF	2	10	20
Konektory			50
DPS	250	1	250
Celková cena:			656

9. Závěr

Tato práce popisuje konstrukci modulu, který slouží k vyhodnocení relativní vlhkosti vzduchu. Je koncipován jako doplnění již existujícího psychrometrického senzoru. Psychrometrický senzor měří teplotu suchého a mokrého teploměru. Naměřené hodnoty předává vyhodnocovacímu modulu, který z jejich rozdílu počítá relativní vlhkost vzduchu. Hodnoty obou teplot a relativní vlhkosti jsou zobrazeny na LCD displeji.

Základem vyhodnocovacího modulu, který jsem navrhl a vytvořil, je mikroprocesor ATmega8, který digitalizuje naměřené hodnoty obou teplot. Dále provádí veškeré výpočty a komunikuje s displejem. Připravena je i možnost budoucího rozšíření o odesílání naměřených dat po sériové lince k dalšímu zpracování

Pracovní rozsah teplot je dán použitým psychrometrem. Rozsah 0-102,4°C je pro většinu aplikací dostatečný. Rychlost zobrazování dat na displeji je také dobrá, činí 5 měření za sekundu. Naopak rychlost vyhodnocení změny vlhkosti trvá 5-10min. Tato poměrně dlouhá prodleva je dána dobou ustálení psychrometru, což vyplývá z jeho principu. Z tohoto důvodu se toto zařízení hodí spíše pro kontinuální měření. Také je potřeba počítat s tím, že nebude schopno zachytit velmi krátké a prudké změny relativní vlhkosti. Ty se ovšem běžně při měření relativní vlhkosti nevyskytují, jak je patrné z naměřených průběhů v kapitole 7.

Předpokládaná oblast použití je v prostorách s vysokým obsahem nečistot v ovzduší, jako jsou např. sušárny a sklady. Zde se projeví výhoda psychrometru, kterou je robustnost jeho konstrukce.

Cena komponent použitých v návrhu je 660Kč. K tomu by bylo zapotřebí připočítat cenu vlastního psychrometru. Dále pak zohlednit čas věnovaný vývoji a výrobě, požadovaný zisk a případně další položky, jako jsou náklady na propagaci a přepravu.

Do budoucna by bylo možno využít rozhraní pro komunikaci s tímto smart senzorem. Případným doplněním konstrukce o čidlo tlaku vzduchu by bylo možno měření dále zpřesnit.

10. Použitá literatura

10.1 Internetové zdroje:

- [1] **ADAMOVSKEÝ Radomír. NEUBERGER Pavel** *Termomechanika II* 2003
- [2] **HANZAL, RNDr. Josef.** Vlhkoměry. *Hw.cz*. [Online] 10. 1 2004. [Citace: 28.2.2009.] <<http://hw.cz/Teorie-a-praxe/Dokumentace/ART1022-Vlhkomery.html>>.
- [3] **NEUSTUPA, Zdeněk.** TECHNICKÉ PROSTŘEDKY ASŘ. [Online] [Citace: 28.2.2009.] <<http://hgf.vsb.cz/neu10/studium/tpa/tpahtml/skripta.htm>>.
- [4] **Vlhkoměr.** *Quido.cz*. [Online] [Citace: 4. 3. 2009.] <<http://www.quido.cz/objevy/vlhkomer.htm>>.
- [5] **Senzor vlhkosti HS07.** *www.smartec.fr* [Online] [Citace: 7. 3. 2009.] <<http://www.smartec.fr/intl/cz/humidity.php>>.
- [6] **Senzor vlhkosti Vishay** *www.vishay.com* [Online] [Citace: 7. 3. 2009.] <<http://www.vishay.com/docs/29001/23226919.pdf>>.
- [7] **Relative humidity above saturated salt solutions at various temperatures.** *http://www.natmus.dk* [Online] [Citace: 16. 3. 2009.] <<http://www.natmus.dk/cons/tp/satslt/satsalt.htm>>
- [8] **Humidity Calibration.** *http://www.thesolar.biz* [Online] [Citace: 16. 3. 2009.] <http://www.thesolar.biz/Relative_Humidity_Calibration_Kit.htm>
- [9] **Univerzální obvod rozhraní Smartec UTI** *www.smartec.fr* [Online] [Citace: 20. 3. 2009.] <<http://www.smartec.fr/intl/cz/uti.php>>.
- [10] **Technické prostředky automatizace.** *http://hgf.vsb.cz* [Online] [Citace: 22. 3. 2009.] <<http://hgf.vsb.cz/neu10/studium/tpa/sylaby/TPA.htm>>.
- [11] **MAREŠ, Ing. Luděk.** Vlhkost vzduchu a její měření. *http://tzb-info.cz* [Online] 13.3.2006. [Citace: 22. 3. 2009.] <<http://vetrani.tzb-info.cz/t.py?t=2&i=3137&h=244&pl=47>>.
- [12] **Psychrometrické funkce** [Online] [Citace: 9. 4. 2009.] <<http://www.filco.cz/index.php?node=40&frommenu=1>>
- [13] **Meteorology** Institute and Museum of History of Science, [Online] [Citace: 9. 4. 2009.] <<http://www.imss.fi.it/presale/eprsal10.html>>
- [14] **Theoretical Meteorology** [Online] [Citace: 9. 4. 2009.] <<http://mysite.du.edu/~etuttle/weather/atphys.htm>>
- [15] **Elektrochemické metody založené na vyčerpávající elektrolýze** [Online] [Citace: 9. 4. 2009.] <http://tomcat.prf.jcu.cz/sima/analyticka_chemie/elektrob.htm>

- [16] **Měření vlhkosti** [Online] [Citace: 12. 4. 2009.]
<<http://www.micro.feld.cvut.cz/home/X34SES/cviceni/Navody%20na%20cviceni/07.Mereni%20vlhkosti.pdf>>
- [17] **Thermal-conductive type gas sensors** [Online] [Citace: 12. 4. 2009.]
< <http://www.nemoto.co.jp/en/products/images/nap-21a.jpg> >
- [18] **THT - Vlhkoměr s teploměrem pro** [Online] [Citace: 12. 4. 2009.]
< http://www.papouch.com/shop/scripts/_detail.asp?katcislo=0301>
- [19] **Snímače teploty, vlhkosti, atmosférického tlaku s výstupem RS485, RS232**
[Online] [Citace: 12. 4. 2009.]
< <http://www.cometsystem.cz/snimace-teploty-vlhkosti-seriove.htm#pTypes> >
- [20] **Snímače vlhkosti** [Online] [Citace: 12. 4. 2009.]
< <http://www.profess.cz/elektrochem.php>>
- [21] **Pertec Dewtron** [Online] [Citace: 12. 4. 2009.]
< http://www.pertec.it/english/Dewtron_technical_sheet.htm >
- [22] **Lawrence M. G**, The relationship between relative humidity and the dew point temperature in moist air: A simple conversion and applications [Online] [Citace: 15. 4. 2009.]
< http://www.reachinformation.com/define/Dew_point.aspx>
- [23] **Dew point** [Online] [Citace: 15. 4. 2009.]
< <http://upload.wikimedia.org/wikipedia/commons/2/2f/Dewpoint-RH.svg> >
- [24] **Seznam přístrojů pro praktickou zk.**[Online] [Citace: 15. 4. 2009.]
< <http://zoohygiene.xf.cz/pristroje.htm> >
- [25] **Vysokoteplotní psychrometr Hygrophil h** [Online] [Citace: 15. 4. 2009.]
<<http://www.ekotechnika.com/produkt/vysokoteplotni-psychrometr-hygrophil-h>>
- [26] **Psychrometr DSL12** [Online] [Citace: 15. 4. 2009.]
< <http://www.rawet.cz/cz/tepl/dsl12.php?odkaz=tepl14>>

10.2 Publikace

[27] ĎAĎO, S., KREIDL, M. *Senzory a měřicí obvody*. Praha: ČVUT, 1999.

ISBN 80-01-02057-6

[28] DANIEL, D., GAJSKI, FRANK, VAHID, SANJIV, NARAYN, JIE GONG
Specification And Design Of Embedded Systems California 2002 ISBN 0-13-150731-1

[29] MATOUŠEK, D. *Práce s mikrokontroléry ATMEL AVR ATmega16*, Praha 2006

ISBN 80-7300-174-8

[30] ČERNÝ, F, *Měření vlhkosti / Psychrometrie/* Praha 1951

10.3 Seznam použitých zkratek

UTI	Universal Transducer Interface, univerzální obvod rozhraní
DP	dew point, rosný bod
FP	frost point, bod ojmění
RH	relative humidity, relativní vlhkost
SW	software
HW	hardware
LiCl	chlorid lithný
A/D	analogově digitální převodník
AC	alternating current - střídavý proud
DC	direct current – stejnosměrný proud
RS232C	standart pro sériovou komunikaci
RS485	standart pro průmyslovou poloduplexní sériovou komunikaci
VA	voltampér
ATmega8	mikroprocesor Atmel
ATmega32	mikroprocesor Atmel
PDIP	Plastic Dual Inline Package, typ pouzdra integrovaného obvodu
TQFP	Thin Quad Flat Pack, typ pouzdra integrovaného obvodu
RTU-01-T	označení psychrometrického čidla vlhkosti
LCD	liquid crystal display, displej s tekutými krystaly
ASCII	American Standard Code for Information Interchange, kódování znaků
ISP	In System Programmable, rozhraní pro programování mikroprocesorů
DPS	deska plošného spoje
PIC	označení skupiny mikroprocesorů výrobce Microchip
AVR	Advanced Virtual RISC
ARM	Advanced RISC Machina

10.4 Seznam obrázků

Obrázek 1 Vlasový vlhkoměr	4
Obrázek 2 Gravimetrický vlhkoměr [14]	5
Obrázek 3 Princip optického měření rosného bodu	6
Obrázek 4 Uspořádání elektrod elektrolytického vlhkoměru [15]	8
Obrázek 5 Elektrolytický vlhkoměr [27]	9
Obrázek 6 Uspořádání detektorů záření	10
Obrázek 7 Absorpce optického záření v atmosféře [16]	10
Obrázek 8 Odporový senzor vlhkosti Samyoung[2]	12
Obrázek 9 Sensory vlhkosti Smartec [5]	13
Obrázek 10 Senzor vlhkosti Vishay [6]	14
Obrázek 11 Porovnání klasické a UTI metod měření [9]	15
Obrázek 12 Zapojení termistorů	16
Obrázek 13 Vyhřívané termistory [17]	16
Obrázek 14 Augustův psychrometr [24]	17
Obrázek 15 Assmanův psychrometr [24]	18
Obrázek 16 Obecné schéma smart senzoru	20
Obrázek 17 THT - Vlhkoměr s teploměrem pro RS485 [18]	21
Obrázek 18 Smart senzor Comet	22
Obrázek 19 Smart senzor Profess [20]	23
Obrázek 20 Smart senzor Dewtron [21]	24
Obrázek 21 Smart senzor EE31 [22]	25
Obrázek 22 Psychrometrický senzor DSL 12 [26]	26
Obrázek 23 Průmyslový psychrometr [25]	27
Obrázek 24 Blokové schéma smart senzoru	29
Obrázek 25 Výstupní proudové smyčky	34
Obrázek 26 Schéma zapojení	35
Obrázek 27 Deska elektroniky senzoru	36

10.5 Seznam tabulek

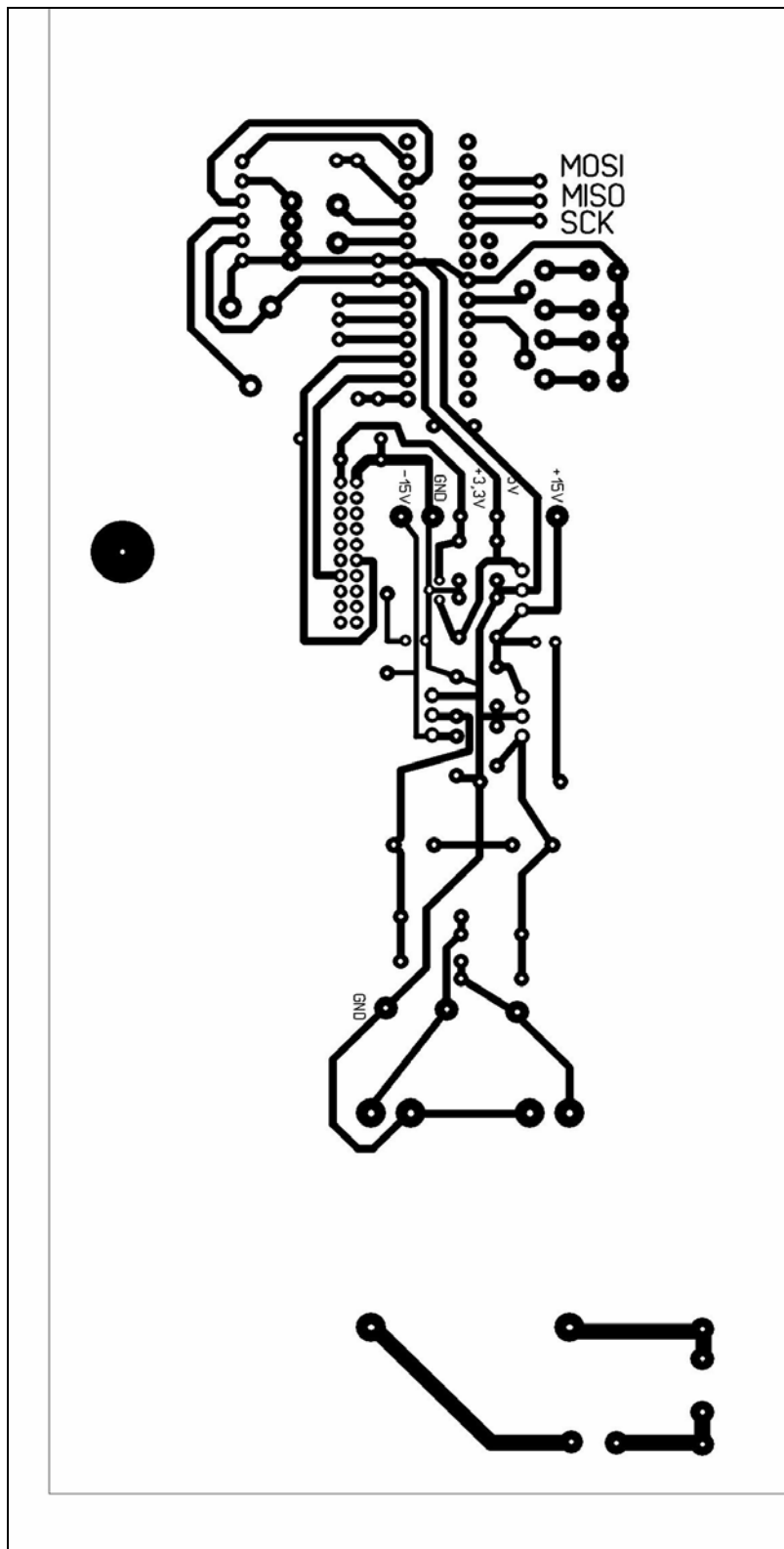
Tabulka 1 Absolutní vlhkost vzduchu pro 100 %RH a vybrané teploty	3
Tabulka 2 Příklady kalibračních roztoků [5], [7], [8].....	19
tabulka 3 Ceny součástek	43

10.6 Seznam grafů

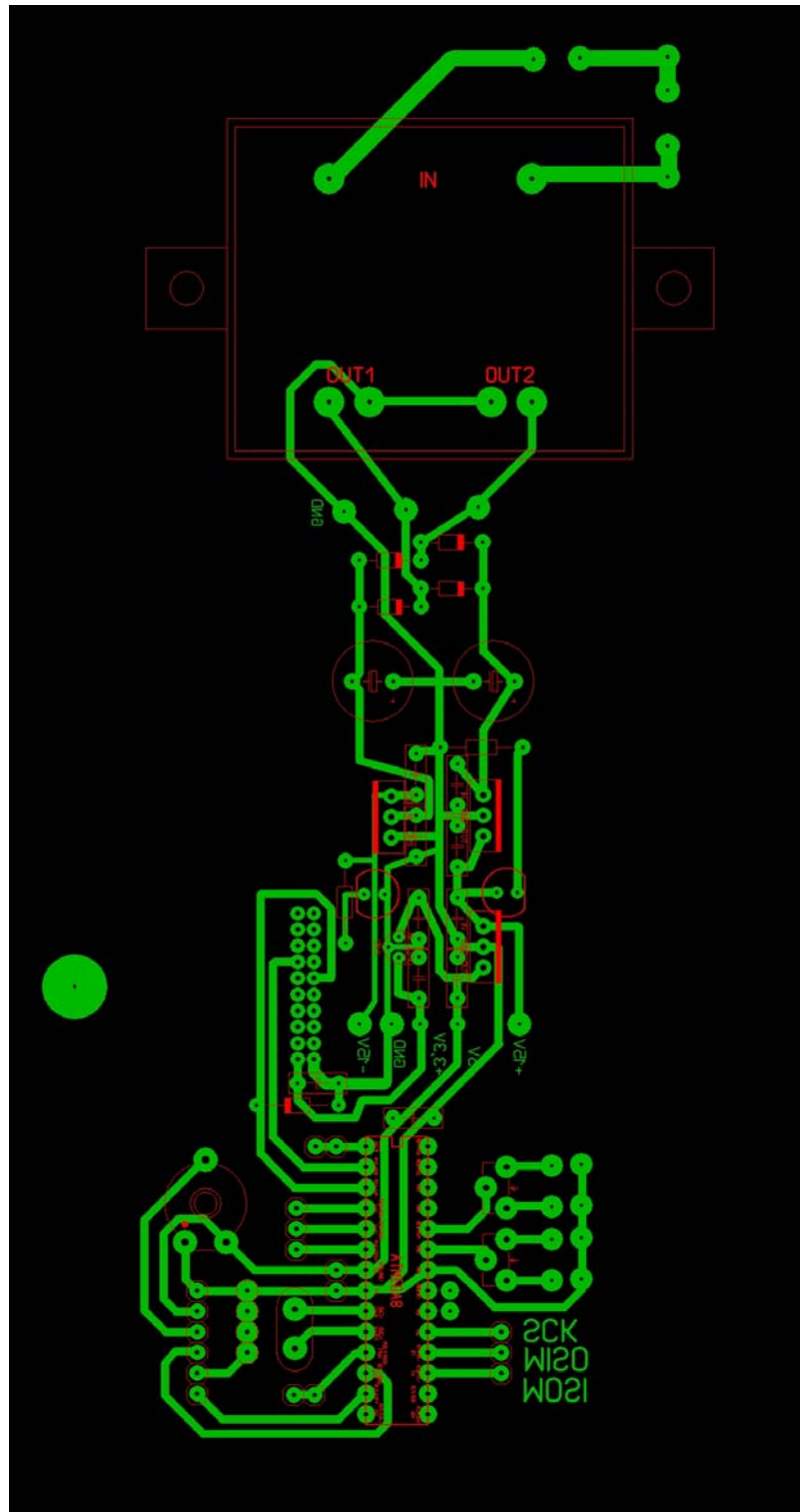
Graf 1 Typický průběh závislosti hodnoty odporu na RH.....	11
Graf 2 Závislost kapacity na vlhkosti vzduchu	14
Graf 3 reprezentace vztahu závislosti RH	32
Graf 4 Molliérův diagram.....	33
Graf 5 Psychrometrický diagram.....	33
Graf 6 Závislost změny RH na čase	40
Graf 7 Průběh relativní vlhkosti	41

11. Přílohy

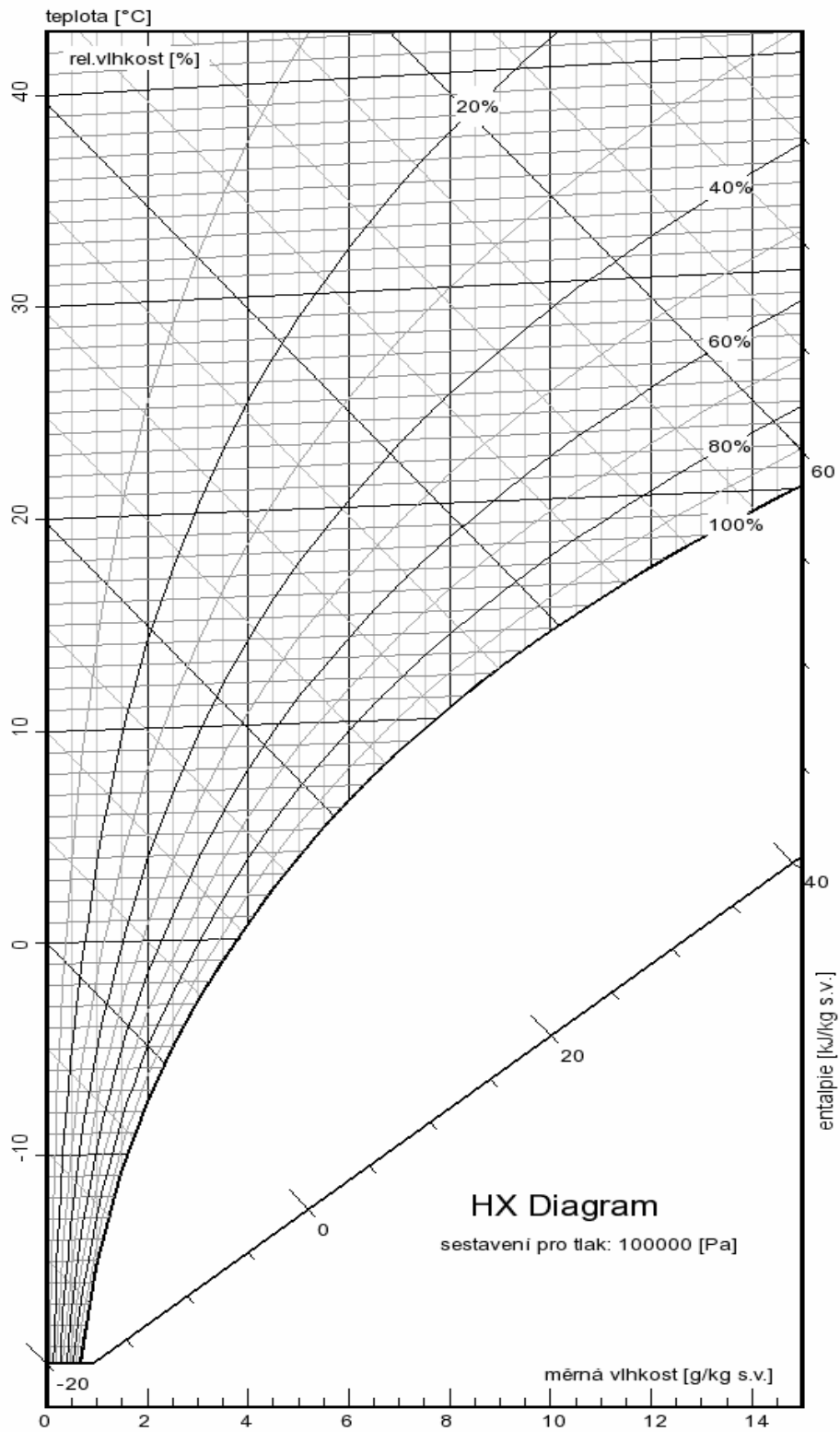
11.1 Výkres plošného spoje



11.2 Výkres osazení plošného spoje



11.3 HX Diagram



Zdroj: <http://qpro.cz/Vypoctynawebu/Hx/diagram.php>

11.4 Vypis programu

```
/**
**
** Date:          Saturday, April 11, 2009 15:00:11
**
**
#define MX_AVR
//Defines for microcontroller
#define MX_AVR
#define MX_EE
#define MX_EE_SIZE 512
#define MX_SPI
#define MX_SPI_B
#define MX_SPI_SDI 4
#define MX_SPI_SDO 3
#define MX_SPI_SCK 5
#define MX_UART
#define MX_UART_D
#define MX_UART_TX 1
#define MX_UART_RX 0
#define MX_MI2C

//Functions
#define F_CPU 2000000UL
#include <avr\io.h>
#include <avr\interrupt.h>
#include <avr\eeeprom.h>
#include <MX_util\delay.h>
#include <MX_util\bit_cmds.h>

//Configuration data
#pragma DATA 0x0, 0xdf
#pragma DATA 0x1, 0xff

//Internal functions
#include "C:\Program Files\Matrix
Multimedia\Flowcode_AVR\FCD\intern
als.h"

//Macro function declarations
void FCM_LCDzobrazeni();
void FCM_Posilani_na_RS232();

//Variable declarations
volatile char FCV_TS2;
volatile char FCV_TS3;
volatile char FCV_TS4;
volatile short FCV_T_MOKRA;
volatile short FCV_T_S;
volatile short FCV_T_SUCHA;
volatile char FCV_VLHKOST1;
volatile char FCV_TM1;
volatile char FCV_VLHKOST2;
volatile char FCV_TM2;
volatile char FCV_VLHKOST3;
volatile char FCV_TM3;

volatile char FCV_TM4;
volatile short FCV_VLHKOST;
volatile short FCV_T_M;
volatile char FCV_TS1;

//LCDDisplay0: //Macro function
declarations

static void
FCD_LCDDisplay0_GetDefines();
static void
FCD_LCDDisplay0_Start();
static void
FCD_LCDDisplay0_Clear();
static void
FCD_LCDDisplay0_PrintASCII(char
Character);
static void
FCD_LCDDisplay0_Command(char in);
static void
FCD_LCDDisplay0_RawSend(char in,
char mask);
static void
FCD_LCDDisplay0_Cursor(char x,
char y);
static void
FCD_LCDDisplay0_PrintNumber(short
Number);
static void
FCD_LCDDisplay0_PrintString(char*
String, char MSZ_String);

//ADC0: //Macro function
declarations

static void FCD_ADC0_SampleADC();
static char FCD_ADC0_ReadAsByte();
static short FCD_ADC0_ReadAsInt();

//ADC1: //Macro function
declarations

static void FCD_ADC1_SampleADC();
static char FCD_ADC1_ReadAsByte();
static short FCD_ADC1_ReadAsInt();

//RS2320: //Macro function
declarations

static void
FCD_RS2320_GetDefines();
```

```

static void                                     //Call Component Macro
FCD_RS2320_SendRS232Char(char                   //Call Component Macro:
nChar);                                        LCDDisplay(0)::Cursor(3, 0)
static char                                     FCD_LCDDisplay0_Cursor(3,
FCD_RS2320_ReceiveRS232Char(char              0);
nTimeout);

//Supplementary defines                        //Call Component Macro
//Macro implementations                       //Call Component Macro:
void FCM_LCDzobrazeni()                       LCDDisplay(0)::PrintASCII(" . ")
{                                              FCD_LCDDisplay0_PrintASCII('
//Call Component Macro                        ');
//Call Component Macro:                      FCD_LCDDisplay0_PrintASCII('
LCDDisplay(0)::Cursor(0, 0)                  ');
FCD_LCDDisplay0_Cursor(0,                    FCD_LCDDisplay0_PrintASCII('
0);                                           .');
//Call Component Macro                       FCD_LCDDisplay0_PrintASCII('
//Call Component Macro:                      ');
LCDDisplay(0)::PrintASCII("Ts: .           LCDDisplay(0)::Cursor(3, 0)
C RH: . ")                                    FCD_LCDDisplay0_Cursor(3,
T');                                          0);
FCD_LCDDisplay0_PrintASCII('               //Call Component Macro
s');                                         //Call Component Macro:
FCD_LCDDisplay0_PrintASCII('               LCDDisplay(0)::PrintNumber(TS3)
:');                                         FCD_LCDDisplay0_PrintNumber(
FCD_LCDDisplay0_PrintASCII('               FCV_TS3);
');                                          //Call Component Macro
FCD_LCDDisplay0_PrintASCII('               //Call Component Macro:
');                                         LCDDisplay(0)::Cursor(4, 0)
FCD_LCDDisplay0_PrintASCII('               FCD_LCDDisplay0_Cursor(4,
.);                                         0);
FCD_LCDDisplay0_PrintASCII('               //Call Component Macro
');                                         //Call Component Macro:
FCD_LCDDisplay0_PrintASCII('               LCDDisplay(0)::PrintNumber(TS2)
C');                                         FCD_LCDDisplay0_PrintNumber(
FCD_LCDDisplay0_PrintASCII('               FCV_TS2);
');                                         //Call Component Macro
FCD_LCDDisplay0_PrintASCII('               //Call Component Macro:
R');                                         LCDDisplay(0)::Cursor(6, 0)
FCD_LCDDisplay0_PrintASCII('               FCD_LCDDisplay0_Cursor(6,
H');                                         0);
FCD_LCDDisplay0_PrintASCII('               //Call Component Macro
:');                                         //Call Component Macro:
FCD_LCDDisplay0_PrintASCII('               LCDDisplay(0)::PrintNumber(TS1)
');                                         FCD_LCDDisplay0_PrintNumber(
FCD_LCDDisplay0_PrintASCII('               FCV_TS1);
.);                                         //Call Component Macro
FCD_LCDDisplay0_PrintASCII('               //Call Component Macro:
');                                         LCDDisplay(0)::Cursor(12, 0)
//Comment:                                    FCD_LCDDisplay0_Cursor(12,
//Napis v hornim                             0);
//radku displeje                            //Call Component Macro

```



```

//Call Component Macro:
LCDDisplay(0)::PrintASCII(" . ")
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
. ');
FCD_LCDDisplay0_PrintASCII('
');

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Cursor(12, 0)
FCD_LCDDisplay0_Cursor(12,
0);

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::PrintNumber(vlhkost
3)
FCD_LCDDisplay0_PrintNumber(
FCV_VLHKOST3);

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Cursor(13, 0)
FCD_LCDDisplay0_Cursor(13,
0);

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::PrintNumber(vlhkost
2)
FCD_LCDDisplay0_PrintNumber(
FCV_VLHKOST2);

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Cursor(15, 0)
FCD_LCDDisplay0_Cursor(15,
0);

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::PrintNumber(vlhkost
1)
FCD_LCDDisplay0_PrintNumber(
FCV_VLHKOST1);

//Comment:
//Jdeme na
//druhy radek

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Cursor(0, 1)
FCD_LCDDisplay0_Cursor(0,
1);

//Call Component Macro

```

```

//Call Component Macro:
LCDDisplay(0)::PrintASCII("Tv: .
C
%")
FCD_LCDDisplay0_PrintASCII('
T');
FCD_LCDDisplay0_PrintASCII('
v');
FCD_LCDDisplay0_PrintASCII('
:');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
. ');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
C');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
%');

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Cursor(3, 1)
FCD_LCDDisplay0_Cursor(3,
1);

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::PrintASCII(" . ")
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
');
FCD_LCDDisplay0_PrintASCII('
. ');
FCD_LCDDisplay0_PrintASCII('
');

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Cursor(3, 1)
FCD_LCDDisplay0_Cursor(3,
1);

```

```

        //Call Component Macro
        //Call Component Macro:
LCDDisplay(0)::PrintNumber(TM3)
        FCD_LCDDisplay0_PrintNumber(
FCV_TM3);

        //Call Component Macro
        //Call Component Macro:
LCDDisplay(0)::Cursor(4, 1)
        FCD_LCDDisplay0_Cursor(4,
1);

        //Call Component Macro
        //Call Component Macro:
LCDDisplay(0)::PrintNumber(TM2)
        FCD_LCDDisplay0_PrintNumber(
FCV_TM2);

        //Call Component Macro
        //Call Component Macro:
LCDDisplay(0)::Cursor(6, 1)
        FCD_LCDDisplay0_Cursor(6,
1);

        //Call Component Macro
        //Call Component Macro:
LCDDisplay(0)::PrintNumber(TM1)
        FCD_LCDDisplay0_PrintNumber(
FCV_TM1);
}

void FCM_Posilani_na_RS232()
{

        //Call Component Macro
        //Call Component Macro:
RS232(0)::SendRS232Char(vlhkost)
        FCD_RS2320_SendRS232Char(FCV
_VLHKOST);

        //Delay
        //Delay: 1 ms
        delay_ms(1);
}

//LCDDisplay0: //Macro
implementations

static void
FCD_LCDDisplay0_GetDefines()
{

        } //Dummy end of function
to allow defines to be added
correctly

        //component connections
#define

```

```

AVR_LCD_5834798_portd
#define LCD_5834798_PORT
portd
#define LCD_5834798_TRIS
trisd

#define LCD_5834798_BIT0
2
#define LCD_5834798_BIT1
3
#define LCD_5834798_BIT2
4
#define LCD_5834798_BIT3
5
#define LCD_5834798_RS
6
#define LCD_5834798_E
7

#define LCD_5834798_DELAY
delay_us(100)

#ifdef AVR_LCD_5834798_porta
#define LCD_5834798_AVRPIN
PIN_A
#define LCD_5834798_AVRDDR
DDR_A
#define
LCD_5834798_AVRPORT PORTA
#endif
#ifdef AVR_LCD_5834798_portb
#define LCD_5834798_AVRPIN
PIN_B
#define LCD_5834798_AVRDDR
DDR_B
#define
LCD_5834798_AVRPORT PORTB
#endif
#ifdef AVR_LCD_5834798_portc
#define LCD_5834798_AVRPIN
PIN_C
#define LCD_5834798_AVRDDR
DDR_C
#define
LCD_5834798_AVRPORT PORTC
#endif
#ifdef AVR_LCD_5834798_portd
#define LCD_5834798_AVRPIN
PIN_D
#define LCD_5834798_AVRDDR
DDR_D
#define
LCD_5834798_AVRPORT PORTD
#endif

//internal function
prototypes
void
LCD_5834798_RawSend(char nIn, char
nMask);

```

```

//internal function
implementations
void
LCD_5834798_RawSend(char nIn, char
nMask)
{
    unsigned char pt;
    unsigned char outVal;

    outVal =
LCD_5834798_AVRPIN;
    pt = ((nIn >> 4) &
0x0f);

    clear_bit(outVal,
LCD_5834798_BIT0);
    clear_bit(outVal,
LCD_5834798_BIT1);
    clear_bit(outVal,
LCD_5834798_BIT2);
    clear_bit(outVal,
LCD_5834798_BIT3);
    clear_bit(outVal,
LCD_5834798_RS);

    if (pt & 0x01)
        set_bit(outVal,
LCD_5834798_BIT0);
    if (pt & 0x02)
        set_bit(outVal,
LCD_5834798_BIT1);
    if (pt & 0x04)
        set_bit(outVal,
LCD_5834798_BIT2);
    if (pt & 0x08)
        set_bit(outVal,
LCD_5834798_BIT3);
    if (nMask)
        set_bit(outVal,
LCD_5834798_RS);

    LCD_5834798_AVRPORT =
outVal;
    LCD_5834798_DELAY;
    set_bit
(LCD_5834798_AVRPORT,
LCD_5834798_E);
    LCD_5834798_DELAY;
    clear_bit
(LCD_5834798_AVRPORT,
LCD_5834798_E);
    LCD_5834798_DELAY;

    outVal =
LCD_5834798_AVRPIN;
    pt = (nIn & 0x0f);

    clear_bit(outVal,
LCD_5834798_BIT0);
    clear_bit(outVal,
LCD_5834798_BIT1);

```

```

        clear_bit(outVal,
LCD_5834798_BIT2);
        clear_bit(outVal,
LCD_5834798_BIT3);
        clear_bit(outVal,
LCD_5834798_RS);

        if (pt & 0x01)
            set_bit(outVal,
LCD_5834798_BIT0);
        if (pt & 0x02)
            set_bit(outVal,
LCD_5834798_BIT1);
        if (pt & 0x04)
            set_bit(outVal,
LCD_5834798_BIT2);
        if (pt & 0x08)
            set_bit(outVal,
LCD_5834798_BIT3);
        if (nMask)
            set_bit(outVal,
LCD_5834798_RS);

        LCD_5834798_AVRPORT =
outVal;
        LCD_5834798_DELAY;
        set_bit
(LCD_5834798_AVRPORT,
LCD_5834798_E);
        LCD_5834798_DELAY;
        clear_bit
(LCD_5834798_AVRPORT,
LCD_5834798_E);
        LCD_5834798_DELAY;
    }

    // Dummy function to close
the defines section off
    void
LCD_5834798_Dummy_Function();
    void
LCD_5834798_Dummy_Function()
    {
    }

    static void
FCD_LCDDisplay0_Start()
    {

        set_bit(LCD_5834798_AVRPORT,
LCD_5834798_E);

        set_bit(LCD_5834798_AVRDDR,
LCD_5834798_BIT0);

        set_bit(LCD_5834798_AVRDDR,
LCD_5834798_BIT1);

        set_bit(LCD_5834798_AVRDDR,

```

```

LCD_5834798_BIT2);

    set_bit(LCD_5834798_AVRDDR,
LCD_5834798_BIT3);

    set_bit(LCD_5834798_AVRDDR,
LCD_5834798_RS);

    set_bit(LCD_5834798_AVRDDR,
LCD_5834798_E);

        delay_ms(12);

LCD_5834798_RawSend(0x33,
0);
        delay_ms(2);

LCD_5834798_RawSend(0x32,
0);
        delay_ms(2);

LCD_5834798_RawSend(0x2c,
0);
        delay_ms(2);

LCD_5834798_RawSend(0x06,
0);
        delay_ms(2);

LCD_5834798_RawSend(0x0c,
0);
        delay_ms(2);

        //clear the display

LCD_5834798_RawSend(0x01,
0);
        delay_ms(2);

LCD_5834798_RawSend(0x02,
0);
        delay_ms(2);

}

static void
FCD_LCDDisplay0_Clear()
{

LCD_5834798_RawSend(0x01,
0);
        delay_ms(2);

LCD_5834798_RawSend(0x02,
0);
        delay_ms(2);

}

```

```

static void
FCD_LCDDisplay0_PrintASCII(char
Character)
{

LCD_5834798_RawSend(Characte
r, 0x10);

}

static void
FCD_LCDDisplay0_Command(char in)
{

LCD_5834798_RawSend(in,
0);
        delay_ms(2);

}

static void
FCD_LCDDisplay0_RawSend(char in,
char mask)
{
        //Error Reading Code For LCD
Display::Macro_RawSend}

static void
FCD_LCDDisplay0_Cursor(char x,
char y)
{

        #if (2 == 1)
                y=0x80;
        #endif

        #if (2 == 2)
                if (y==0)
                        y=0x80;
                else
                        y=0xc0;
        #endif

        #if (2 == 4)
                if (y==0)
                        y=0x80;
                else if (y==1)
                        y=0xc0;
                else if (y==2)
                        y=0x94;
                else
                        y=0xd4;
        #endif

LCD_5834798_RawSend(y+x, 0);
        delay_ms(2);

}

```

```

static void
FCD_LCDDisplay0_PrintNumber(short
Number)
{
    short tmp_int;
    char tmp_byte;
    if (Number < 0)
    {
        LCD_5834798_RawSend('-',
0x10);
        Number = 0 -
Number;
    }
    tmp_int = Number;
    if (Number >= 10000)
    {
        tmp_byte =
tmp_int / 10000;
        LCD_5834798_RawSend('0' +
tmp_byte, 0x10);
        while (tmp_byte >
0)
        {
            tmp_int =
tmp_int - 10000;
            tmp_byte--;
        }
        if (Number >= 1000)
        {
            tmp_byte =
tmp_int / 1000;
            LCD_5834798_RawSend('0' +
tmp_byte, 0x10);
            while (tmp_byte >
0)
            {
                tmp_int =
tmp_int - 1000;
                tmp_byte--;
            }
            if (Number >= 100)
            {
                tmp_byte =
tmp_int / 100;
                LCD_5834798_RawSend('0' +
tmp_byte, 0x10);
                while (tmp_byte >
0)
                {
                    tmp_int =

```

```

tmp_int - 100;
                tmp_byte--;
            }
        }
        if (Number >= 10)
        {
            tmp_byte =
tmp_int / 10;
            LCD_5834798_RawSend('0' +
tmp_byte, 0x10);
            while (tmp_byte >
0)
            {
                tmp_int =
tmp_int - 10;
                tmp_byte--;
            }
            LCD_5834798_RawSend('0'
+ tmp_int, 0x10);
        }
    }
    static void
FCD_LCDDisplay0_PrintString(char*
String, char MSZ_String)
    {
        char idx;
        for (idx=0;
idx<MSZ_String; idx++)
        {
            if
(String[idx]==0)
            {
                break;
            }
            LCD_5834798_RawSend(String[i
dx], 0x10);
        }
    }
    //ADC0: //Macro implementations
    static void FCD_ADC0_SampleADC()
    {
        //This is from the FCD
file...
        char ddc;
        ddc = DDRC;
        DDRC &= ~(1 << 0);
        ADCSRA = (1 << ADEN) | 0x05;
        ADMUX = 0x60 | 0;
        delay_10us(2);
        ADCSRA |= (1 << ADSC);
        while(ADCSRA & (1 << ADSC));
        DDRC = ddc;
    }

```

```

}
static char FCD_ADC0_ReadAsByte()
{
    //This is from the FCD
file...
    return ADCH;
}

static short FCD_ADC0_ReadAsInt()
{
    //This is from the FCD
file...
    int iRetVal;
    iRetVal = (ADCL >> 6);
    iRetVal |= (ADCH << 2);
    return (iRetVal);
}

//ADC1: //Macro implementations

static void FCD_ADC1_SampleADC()
{
    //This is from the FCD
file...
    char ddc;
    ddc = DDRC;
    DDRC &= ~(1 << 1);
    ADCSRA = (1 << ADEN) | 0x05;
    ADMUX = 0x60 | 1;
    delay_10us(2);
    ADCSRA |= (1 << ADSC);
    while(ADCSRA & (1 << ADSC));
    DDRC = ddc;
}

static char FCD_ADC1_ReadAsByte()
{
    //This is from the FCD
file...
    return ADCH;
}

static short FCD_ADC1_ReadAsInt()
{
    //This is from the FCD
file...
    int iRetVal;
    iRetVal = (ADCL >> 6);
    iRetVal |= (ADCH << 2);
    return (iRetVal);
}

//RS2320: //Macro implementations

static void
FCD_RS2320_GetDefines()
{
    } //Dummy end of function
to allow defines to be added
correctly

//component connections

#ifdef MX_UART_ID
#define UDR          UDR0
#define UBRRH
UBRR0H
#define UBRL
UBRR0L
#define UCSRC
UCSR0C
#define UCSRB
UCSR0B
#define UCSRA
UCSR0A

#define UMSEL1
UMSEL01
#define UMSEL0
UMSEL00
#define UPM1          UPM01
#define UPM0          UPM00
#define USBS          USBS0
#define UCSZ1
UCSZ01
#define UCSZ0
UCSZ00
#define UCPOL
UCPOL0

#define RXCIE
RXCIE0
#define TXCIE
TXCIE0
#define UDRIE
UDRIE0
#define RXEN          RXEN0
#define TXEN          TXEN0
#define UCSZ2
UCSZ02
#define RXB8          RXB80

#define RXC          RXC0
#define TXC          TXC0
#define UDRE          UDRE0
#define FE          FE0
#define DOR          DOR0
#define UPE
UPE0

#define U2X          U2X0
#define MPCM          MPCM0
#endif
}

```

```

        #ifndef USR
            #define UCSRA USR
            #define UCSRB UCR
        #endif

        #if (2 == 0)
            #define
FC_RS232_RTS_PORT PORTA
            #define
FC_RS232_RTS_DDR DDRA
            #endif
            #if (2 == 1)
                #define
FC_RS232_RTS_PORT PORTB
                #define
FC_RS232_RTS_DDR DDRB
                #endif
            #if (2 == 2)
                #define
FC_RS232_RTS_PORT PORTC
                #define
FC_RS232_RTS_DDR DDRC
                #endif
            #if (2 == 3)
                #define
FC_RS232_RTS_PORT PORTD
                #define
FC_RS232_RTS_DDR DDRD
                #endif

            #if (2 == 0)
                #define
FC_RS232_CTS_PORT PINA
                #define
FC_RS232_CTS_DDR DDRA
                #endif
            #if (2 == 1)
                #define
FC_RS232_CTS_PORT PINB
                #define
FC_RS232_CTS_DDR DDRB
                #endif
            #if (2 == 2)
                #define
FC_RS232_CTS_PORT PINC
                #define
FC_RS232_CTS_DDR DDRC
                #endif
            #if (2 == 3)
                #define
FC_RS232_CTS_PORT PIND
                #define
FC_RS232_CTS_DDR DDRD
                #endif

            #define FC_RS232_RTS_PIN
0
            #define FC_RS232_CTS_PIN
4

            #if (0 == 1)
                #define
FC_RS232_HARDWARE
                #endif

                #if (0 == 1)
                    #define FC_RS232_DEBUG
                #endif

                #if (0 == 1)
                    #define FC_RS232_ECHO
                #endif

                    #define
fc_rsSTATUS_LOOP 0
                    #define
fc_rsSTATUS_TIMEOUT 1
                    #define
fc_rsSTATUS_RXBYTE 2

                    #define
FC_RS232_TXSTA_VAL 4
                    #define
FC_RS232_SPBRG_VAL 129

                    // Dummy function to close
the defines section off
                    void RS232_Dummy_Function();
                    void RS232_Dummy_Function()
                    {

                    }

                    static void
FCD_RS2320_SendRS232Char(char
nChar)
                    {
                        while ((test_bit(UCSRA,
UDRE)) == 0); // wait for
tx ready

                        #ifndef FC_RS232_HARDWARE
                            //wait until CTS is low
                            while
(test_bit(FC_RS232_CTS_PORT,
FC_RS232_CTS_PIN));
                        #endif

                        UDR = nChar;
                    }

                    static char
FCD_RS2320_ReceiveRS232Char(char
nTimeout)
                    {
                        char delay1 = 0;
                        char delay2 = 0;
                        char dummy = 0;
                        char retVal = 255;
                        char bWaitForever = 0;
                        char rxStatus =

```

```

fc_rsSTATUS_LOOP;

    #ifdef FC_RS232_HARDWARE
        //ready to accept data

        clear_bit(FC_RS232_RTS_PORT,
FC_RS232_RTS_PIN);
        #endif

        if (nTimeout == 255)
        {
            bWaitForever = 1;
        }

        while (rxStatus ==
fc_rsSTATUS_LOOP)
        {
            if
(test_bit(UCSRA, RXC))
            {
                //received
                a Char
                rxStatus =
fc_rsSTATUS_RXBYTE;
            } else {
                if
(bWaitForever == 0)
                {
                    //don't wait forever, so do
                    timeout thing...
                    if
(nTimeout == 0)
                    {
                        rxStatus =
fc_rsSTATUS_TIMEOUT;
                    }
                }
            }
        }
    else {
        //decrement timeout
        delay1--;

        if (delay1 == 0)
        {
            nTimeout--;
        }
    }
}

    if (rxStatus ==
fc_rsSTATUS_RXBYTE)
    {
        if(test_bit(UCSRA, FE))
    }

    {
        dummy =
UDR;
    } else {
        if
(test_bit(UCSRA, DOR))
        {
            dummy
= UDR;
        } else {
            //no
            error, so rx byte is valid
            retVal = UDR;
        }
    }
    #ifdef
FC_RS232_ECHO
        FCD_RS2320_SendRS232Char(ret
Val);
    #endif
}
}

    #ifdef FC_RS232_HARDWARE
        //not ready to accept
        data
        set_bit(FC_RS232_RTS_PORT,
FC_RS232_RTS_PIN);
    #endif
        return (retVal);
    }
}

//Supplementary implementations

int main()
{
    //Ten samy program jako LCD
    a ADC, ale veci pro zobrazovani na
    LCD jsou nasypany do makra, takže
    je to prehlednejsi

    //Initialisation
    MCUCSR=0x00;
    WDTCR=0x10;

    UCSRA = 0x00;
    //
    clear all UCSRA flags

    #ifdef UCSRC
        #ifdef MX_UART_UCSRC
            UCSRC = ((1 <<
UCSZ1) | (1 << UCSZ0));
            // configuration

```



```

for 8 data bits (Individual
UCSRCx)
    #else
        UCSRC = ((1 <<
URSEL) | (1 << UCSZ1) | (1 <<
UCSZ0)); // configuration for 8
data bits (Combined UBRRHx/UCSRCx)
    #endif
    #endif

    #ifdef UBRR
        UBRR =
FC_RS232_SPBRG_VAL;
        // same baudrate
calculation
    #else
        if (FC_RS232_TXSTA_VAL
& 0x04) //Test PIC
BRGH setting
        {
            UBRR1 =
FC_RS232_SPBRG_VAL;
            // same baudrate calculation
            UBRRH = 0x00;

            // as PIC
        }
        else
        {
            UBRR1 =
FC_RS232_SPBRG_VAL << 2; //
PIC baudrate calculation
            UBRRH =
FC_RS232_SPBRG_VAL >> 6; // x4
        }
    #endif
        set_bit(UCSRB, RXEN); //
enable rx
        set_bit(UCSRB, TXEN); //
enable tx

    #ifdef FC_RS232_HARDWARE

        clear_bit(FC_RS232_CTS_DDR,
FC_RS232_CTS_PIN); //CTS is an
input

        set_bit(FC_RS232_RTS_DDR,
FC_RS232_RTS_PIN); //RTS is
an output

        set_bit(FC_RS232_RTS_PORT,
FC_RS232_RTS_PIN); //not
ready to accept data
    #endif

    //Interrupt initialisation
code

//Call Component Macro
//Call Component Macro:
LCDDisplay(0)::Start
FCD_LCDDisplay0_Start();

//Loop
//Loop: While 1
while( 1 )
{
    //Call Component Macro
    //Call Component Macro:
ADC(0)::SampleADC
    FCD_ADC0_SampleADC();

    //Call Component Macro
    //Call Component Macro:
t_sucha=ADC(0)::ReadAsInt
    FCV_T_SUCHA =
FCD_ADC0_ReadAsInt();

    //Call Component Macro
    //Call Component Macro:
ADC(1)::SampleADC
    FCD_ADC1_SampleADC();

    //Call Component Macro
    //Call Component Macro:
t_mokra=ADC(1)::ReadAsInt
    FCV_T_MOKRA =
FCD_ADC1_ReadAsInt();

    //Calculation
    //Calculation:
    // vlhkost = 1000 - 8
    * ( t_sucha - t_mokra )
    // vlhkost3 = vlhkost
    / 100
    // vlhkost = vlhkost
    MOD 100
    // vlhkost2 = vlhkost
    / 10
    // vlhkost = vlhkost
    MOD 10
    // vlhkost1 = vlhkost
    FCV_VLHKOST = 1000 - 8
    * ( FCV_T_SUCHA - FCV_T_MOKRA ) ;
    FCV_VLHKOST3 =
FCV_VLHKOST / 100 ;
    FCV_VLHKOST =
FCV_VLHKOST % 100 ;
    FCV_VLHKOST2 =
FCV_VLHKOST / 10 ;
    FCV_VLHKOST =
FCV_VLHKOST % 10 ;
    FCV_VLHKOST1 =
FCV_VLHKOST ;
}

```

```

//Call Macro
//Call Macro:
deset.mist //Vypocet mokrych LCDzobrazeni
//Calculation: FCM_LCDzobrazeni();
// TM4 = t_mokra /
1000 // t_mokra = t_mokra //Call Macro
MOD 1000 // TM3 = t_mokra / 100 //Call Macro:
// t_mokra = t_mokra Posilani_na_RS232
MOD 100 // TM2 = t_mokra / 10 FCM_Posilani_na_RS232();
// t_mokra = t_mokra
MOD 10 // TM1 = t_mokra //Delay
FCV_TM4 = FCV_T_MOKRA //Delay: 200 ms
/ 1000 ; delay_ms(200);
FCV_T_MOKRA = FCV_T_MOKRA % 1000 ; }
FCV_TM3 = FCV_T_MOKRA
/ 100 ;
FCV_T_MOKRA = FCV_T_MOKRA % 100 ; mainendloop: goto
FCV_TM2 = FCV_T_MOKRA mainendloop;
/ 10 ; FCV_T_MOKRA = }
FCV_T_MOKRA % 10 ; FCV_TM1 = FCV_T_MOKRA
;

deset.mist //Vypocet suchych
//Calculation:
// TS4 = t_sucha /
1000 // t_sucha = t_sucha
MOD 1000 // TS3 = t_sucha / 100
// t_sucha = t_sucha
MOD 100 // TS2 = t_sucha / 10
// t_sucha = t_sucha
MOD 10 // TS1 = t_sucha
FCV_TS4 = FCV_T_SUCHA
/ 1000 ;
FCV_T_SUCHA = FCV_T_SUCHA % 1000 ;
FCV_TS3 = FCV_T_SUCHA
/ 100 ;
FCV_T_SUCHA = FCV_T_SUCHA % 100 ;
FCV_TS2 = FCV_T_SUCHA
/ 10 ;
FCV_T_SUCHA = FCV_T_SUCHA % 10 ;
FCV_TS1 = FCV_T_SUCHA
;

```