

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## ELEKTRONICKÝ KOMPAS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR ŠPAŇHEL

BRNO 2010



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## **ELEKTRONICKÝ KOMPAS**

ELECTRONIC COMPASS

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**PETR ŠPAŇHEL**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. VÁCLAV ŠIMEK**

BRNO 2010

## Zadání bakalářské práce

Řešitel: **Špaňhel Petr**

Obor: Informační technologie

Téma: **Elektronický kompas**  
**Electronic Compass**

Kategorie: Vestavěné systémy

Pokyny:

1. Prostudujte možnosti elektronické detekce magnetického pole, tlaku vzduchu a úhlové orientace vůči zemskému povrchu. Připravte krátkou studii na uvedené téma.
2. Navrhněte koncepci elektronického kompasu v podobě samostatně pracujícího modulu. Zvažte možnosti propojení s externími zařízeními (např. platformou FITkit).
3. V návrhovém systému Eagle nebo KICAD proveďte realizaci desky plošných spojů uvažovaného modulu.
4. Vytvořte obslužný firmware pro MCU tak, aby se na LCD zobrazovaly údaje o aktuální orientaci.
5. Vhodným způsobem demonstруйте funkčnost navrženého systému.
6. Diskutujte dosažené výsledky a zvažte případná rozšíření.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1-3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

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

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

Vedoucí: **Šimek Václav, Ing.**, UPSY FIT VUT

Datum zadání: 1. listopadu 2009

Datum odevzdání: 19. května 2010

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



---

doc. Ing. Zdeněk Kotásek, CSc.  
vedoucí ústavu

## Abstrakt

Tato bakalářská práce se zabývá návrhem a realizací zařízení pro měření teploty, tlaku vzduchu, úhlové orientace vůči zemskému povrchu a jejich zobrazení na grafickém displeji. V úvodní kapitole bude čtenáři nastíněna koncepce systému, který musí řešit úlohu od načtení hodnoty ze senzorů, až po grafické zobrazení této hodnoty na displeji. V druhé kapitole je popsán návrh zařízení pro měření teploty, tlaku vzduchu, úhlové orientace vůči zemskému povrchu a jeho software. V následující části čtenář najde popis softwaru sloužícímu k zobrazování naměřených dat na grafickém displeji. Předposlední část je zaměřena na ovládání daného zařízení a jeho uživatelské rozhraní. V závěrečné části jsou zhodnoceny dosažené výsledky práce.

## Abstract

This bachelor's thesis deals with a design and construction of equipment that should offer an easy approach to measurement of temperature, air pressure and angular orientation to the Earth surface. Subsequently, this set of information will be shown in a concise way on a graphic LCD display. The introductory chapter provides a general outline to the hardware conception, which primarily involves reading of data from sensors and their graphical visualization. Details on control firmware aspects are included in the following two chapters. Fourth chapter is aimed at device operation and its user interface. Finally, evaluation of the results and conclusion is provided.

## Klíčová slova

elektronický kompas, kompas, snímač, grafický lcd displej, měření teploty, měření tlaku, měření úhlové orientace, magnetické pole Země, atmel avr

## Keywords

electronic compass, compass, scanner, graphic lcd display, temperature measurement, pressure measurement, measurement of angular orientation, the Earth's magnetic field, Atmel AVR

## Citace

Petr Špaňhel: Elektronický kompas, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Elektronický kompas

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing, Václava Šimka

.....

Petr Špaňhel  
18. května 2010

## Poděkování

Děkuji vedoucímu bakalářské práce Ing. Václavu Šimkovi, za jeho rady a čas, který této bakalářské práci věnoval.

© Petr Špaňhel, 2010.

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

# Obsah

<b>1 Úvod</b>	<b>3</b>
1.1 Detekce neelektrických veličin . . . . .	3
1.1.1 Magnetické pole Země . . . . .	3
1.1.2 Úhlová orientace vůči zemskému povrchu . . . . .	4
1.1.3 Tlak vzduchu . . . . .	4
1.1.4 Teplota vzduchu . . . . .	5
1.2 Návrh systému . . . . .	6
1.2.1 Načítač . . . . .	7
1.2.2 Displej . . . . .	7
1.2.3 Snímače . . . . .	8
1.2.4 Napájecí obvod . . . . .	8
1.2.5 Ovládání . . . . .	8
1.2.6 Komunikace . . . . .	8
<b>2 Návrh hardware</b>	<b>9</b>
2.1 Načítač . . . . .	9
2.1.1 Atmel AVR ATmega . . . . .	9
2.2 Displej . . . . .	13
2.2.1 Modul displeje ATM12864D . . . . .	13
2.3 Snímače . . . . .	16
2.3.1 I <sup>2</sup> C . . . . .	16
2.3.2 Teploty . . . . .	18
2.3.3 Tlaku . . . . .	20
2.3.4 Úhlové orientace . . . . .	22
2.4 Ovládání . . . . .	23
2.5 Napájecí obvod . . . . .	24
2.6 Deska plošných spojů . . . . .	24
<b>3 Software měřicího zařízení</b>	<b>26</b>
3.1 Struktura zdrojových souborů a klíčové funkce . . . . .	26
3.1.1 config.h . . . . .	26
3.1.2 tables.h . . . . .	27
3.1.3 main.c . . . . .	27
3.1.4 lcd.c . . . . .	27
3.1.5 grafic.c . . . . .	28
3.1.6 i2c.c . . . . .	28
3.1.7 senzors.c . . . . .	29
3.1.8 uart.c . . . . .	29

3.1.9	delay.c . . . . .	30
3.1.10	menu.c . . . . .	30
3.1.11	initialize.c . . . . .	30
3.2	Běh programu . . . . .	30
<b>4</b>	<b>Uživatelské rozhraní</b>	<b>32</b>
4.1	Zobrazení . . . . .	32
4.2	Menu . . . . .	32
<b>5</b>	<b>Závěr</b>	<b>34</b>
5.1	Dosažené výsledky . . . . .	34
5.2	Možné vylepšení . . . . .	35
5.3	Shrnutí . . . . .	35
	<b>Použitá literatura</b>	<b>37</b>
	<b>Seznam použitých zkratk</b>	<b>38</b>
<b>A</b>	<b>Podklady pro výrobu měřícího zařízení</b>	<b>41</b>
<b>B</b>	<b>Fotografie výsledného zařízení</b>	<b>44</b>
<b>C</b>	<b>Ukázka komunikace</b>	<b>46</b>
<b>D</b>	<b>CD disk</b>	<b>47</b>

# Kapitola 1

## Úvod

Cílem této kapitoly bude seznámení čtenáře s možnostmi elektronické detekce magnetického pole, tlaku vzduchu, teploty vzduchu a úhlové orientace vůči zemskému povrchu. V druhé části kapitoly pak bude nastíněno obecné řešení systému. Dále pak čtenáře odkazují na seznam použitých zkratk, kde případně najdou vysvětlené cizí pojmy, které se nachází níže v textu.

### 1.1 Detekce neelektrických veličin

V zadání mé bakalářské práce bylo prostudovat možné řešení při detekci různých fyzikálních jevů. Pro každý jev musíme zjistit, jakou veličinu budeme měřit a poté vybrat vhodný převodník této veličiny na určitou elektrickou veličinu (nejčastěji se jedna o napětí). Poté tuto elektrickou veličinu změřit a převést na požadovanou hodnotu.

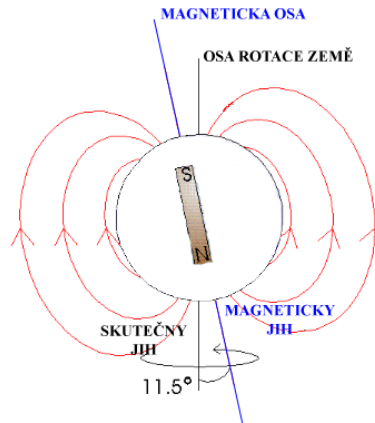
#### 1.1.1 Magnetické pole Země

Protože se Země točí kolem své osy, tekutá vrstva vnějšího jádra umožňuje plášti a pevné kůře, aby se otáčely relativně rychleji než vnitřní jádro. Elektronů v jádře proto vůči elektronům v plášti a kůře vykonávají pohyb. Právě tento pohyb elektronů tvoří jakési přírodní dynamo a výsledek je magnetické pole podobné tomu, které vyvolá elektrická cívka. Magnetická osa Země se od zemské geografické osy odchyluje asi o 11 stupňů a magnetické póly se neshodují s geografickým severním a jižním pólem. Magnetická osa Země svůj úhel neustále mění, ale dlouhou dobu, několik desítek tisíc let, je jejich průměrná vzájemná poloha stabilizována. viz obrázek 1.1.

Tvar magnetického pole Země (*magnetosféry*) silně ovlivňuje sluneční vítr, který na straně bližší k Slunci působí tlakem asi  $1,7 \text{ nPa}$ . Tím zatlačuje magnetosféru do vzdálenosti asi 10 zemských průměrů (asi  $60000 \text{ km}$ ). Na odvrácené straně je magnetosféra prodloužená do chvostu sahajícího daleko za oběžnou dráhu Měsíce.

Magnetosféra je důležitým ochranným faktorem pro biosféru. Magnetosféra Země nedovoluje elektricky nabitým částicím slunečního větru, dostat se k povrchu. Nabité částice (protony a elektrony, které jsou vystřelovány ze Slunce) musí při svém pohybu sledovat siločáry magnetického pole. Plní tedy ochrannou funkci, bez které by život na Zemi nebyl možný. Tyto částice proudí v horních vrstvách atmosféry a při zemi způsobují malá kolísání magnetického pole. Některá kolísání jsou pravidelná - např. denní (noc a den), některá občasná - např. magnetické bouře. V této části bylo čerpáno z [11], [10].





Obrázek 1.1: Magnetické pole Země

### 1.1.2 Úhlová orientace vůči zemskému povrchu

Při úhlové orientaci vůči zemskému povrchu je zapotřebí znát magnetické pole Země v daném místě (viz. sekce 1.1.1). Znalost tohoto pole nám pomůže při určení polohy magnetického severu Země (azimut). Dále pak je zapotřebí znát hodnotu deklinace a inklinace v daném místě na Zemi. Zde popíšu tyto výrazy:

- **Azimut**  $\alpha$  - Úhel mezi magnetickým severem Země a hlavním směrem ( $\mathbf{X}$ ). Magnetický sever je směr ( $\mathbf{H}_{eh}$ ) k severnímu magnetickému pólu Země. Pokud budeme  $\mathbf{H}_{eh}$  považovat jen za velikosti horizontální složky ( $\mathbf{B}_z$ ) zemského magnetického pole, můžeme azimut vyjádřit takto 1.1:

$$\alpha = \arctan \frac{H_{ey}}{H_{ex}} \quad (1.1)$$

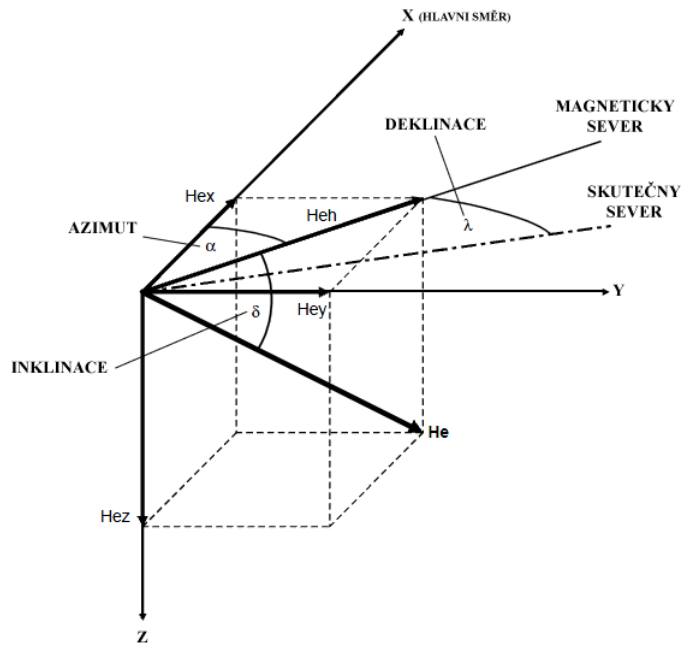
- **Deklinace**  $\delta$  - Úhel mezi zeměpisným a magnetickým severem. Tato odchylka se v různých částech Země mění.
- **Inklinace**  $\gamma$  - Úhel mezi vektorem síly magnetického pole Země s vodorovnou rovinou.

Všechny tyto síly jsou zakresleny na obrázku 1.2.

Převzato z [11].

### 1.1.3 Tlak vzduchu

V mé bakalářské práci se budu zabývat pouze atmosferickým tlakem. Jestliže bude později použit výraz tlak vzduchu, vždy se bude jednat o atmosferický tlak vzduchu. Tlak vzduchu je síla, která působí v daném místě atmosféry kolmo na libovolně orientovanou plochu jednotkové velikosti ( $1 \text{ m}^2$ ), je vyvolán tíhou vzduchového sloupce sahajícího od hladiny, ve které se tlak zjišťuje, až k horní hranici atmosféry. Tlak vzduchu se měří v Pascalech (Pa), případně jeho násobky (kPa, MPa). Na velikost atmosferického tlaku má vliv teplota vzduchu, obsah vodní páry v atmosféře, nadmořská výška a zeměpisná šířka. Pro vzájemné porovnávání se používá tlak redukovaný na hladinu moře. Pro tuto hodnotu je ovšem zapotřebí znát nadmořskou výšku v daném místě. Tento tlak je průměrně 1013,25 hPa (tzv. normální atmosferický tlak).



Obrázek 1.2: Vektor magnetické pole Země

### Závislost tlaku vzduchu na výšce a teplotě

Tlak vzduchu neubývá s výškou lineárně. Stoupáme-li do výšky o stejné úseky (aritmetickou řadou), klesá tlak vzduchu řadou geometrickou. Přibližně po každých  $5,5 \text{ km}$  stoupaní se sníží tlak na polovinu své předchozí hodnoty. O kolik metrů musíme vystoupat, aby tlak vzduchu poklesl o jeden hektopascal ( $hPa$ ) vyjadřuje **vertikální barometrický stupeň**. kolik metrů musíme vystoupat, aby tlak vzduchu poklesl o jeden hektopascal ( $hPa$ ) vyjadřuje vertikální barometrický stupeň. V nížinách klesá tlak vzduchu o jeden hektopascal ( $hPa$ ) přibližně po osmi metrech stoupaní, naopak ve výšce  $7300 \text{ m}$  je pokles o jeden hektopascal ( $hPa$ ) přibližně po  $20 \text{ m}$  stoupaní. Bohužel tlak vzduchu není závislý jenom na nadmořské výšce ale i na teplotě vzduchu. Pro výpočet tlaku vzduchu v různých nadmořských výškách existuje poměrně složitý vzorec. Byly proto vypracovány zjednodušené vzorce, jejichž nedílnou součástí je konstanta, která má pro každou teplotu vzduchu jinou hodnotu. Jako příklad takového zjednodušeného vzorce uvedu **Babinetovu rovnici 1.2**.

$$Tlak = Konstanta * (t1 - t2) * (t1 + t2)^{-1} hPa \quad (1.2)$$

**t1** - tlak vzduchu v nižší nadmořské výšce, **t2** - tlak vzduchu ve vyšší nadmořské výšce  
**Konstanta:**  $0^\circ\text{C} - 15980$ ,  $-20^\circ\text{C} - 14810$ ,  $+40^\circ\text{C} - 18330$

#### 1.1.4 Teplota vzduchu

Termodynamická teplota je základní fyzikální veličinou jednotek SI. Její základní jednotka je kelvin (K). V našich zeměpisných podmínkách se ale spíše setkáme s vedlejší jednotkou - stupeň celsia ( $^\circ\text{C}$ ). Nejnižší možnou teplotou je absolutní nula ( $0 \text{ K}$ ,  $-273,15^\circ\text{C}$ ).

Teplotou vzduchu se v meteorologii a klimatologii rozumí jeho **teplota v zastíněném prostředí ve výšce  $2 \text{ m}$  nad zemským povrchem změřená v meteorologické budce**.

Dále meteorologové zjišťují minimální teplotu vzduchu, která se měří 5 *cm* nad zemským povrchem. V České republice zaznamenáváme nejvyšší hodnoty teploty půdy přibližně ve 13*h*, nejvyšší hodnoty teploty vzduchu kolem 14*h*. Minimální teploty jsou v půdě i vzduchu zaznamenávány přibližně 1 *h* před východem Slunce. S vyšší nadmořskou výškou je chod tepla opožďován. V 1000 *m* se můžeme setkat s výkyvy maximálně 1 až 2 °C, mezi 2000 *m* až 5000 *m* jsou výkyvy 0,5 až 1 °C. Při výškách 5000 *m* režim ovládaný energetickými výměnami mezi zemským povrchem a atmosférou již nezasahuje. Zde jsou teplotní změny způsobovány v důsledku horizontální výměny vzduchových hmot.

### Měření teploty vzduchu

Způsobů, jak zjistit teplotu je mnoho, ale pro potřeby měření teploty vzduchu v číslicové technice se používá změna elektrického odporu vodičů nebo polovodičů při různých teplotách. Odpor vodičů se vzrůstající teplotou stoupá, zatímco odpor polovodičů se vzrůstající teplotou klesá. Tuto vlastnost můžeme například pomocí Ohmovy metody 1.3 měřit a podle výsledného odporu stanovit teplotu.

$$R_x = \frac{U_x}{I_x} [\Omega] \quad (1.3)$$

$U_x$  - Úbytek napětí na měřeném odporu ,  $I_x$  - Proud protékající měřeným odporem  
V této části bylo čerpáno z [9].

## 1.2 Návrh systému

Ze zadání bakalářské práce vyplývá, že celý systém elektronického kompasu musí pracovat v podobě samostatně pracujícího modulu. Po úvaze nad celým systémem jsem se ho rozhodl rozdělit do více celků. Z důvodu možné výměny jednotlivých celků za novější, přesnější, méně energeticky náročné atd. Tímto zároveň umožním případné dodělání dalších funkcí kompasu.

Prvním a hlavní celek (dále v textu jen načítač) musí umět zpracovávat naměřená data, poté je zobrazit a mít schopnost reagovat na uživatelské vstupy.

Druhým celkem je zobrazovací jednotka, jejímž úkolem je zobrazit data uživateli. V další části mé práce budu mluvit o této části jako o displeji.

Třetí celek slouží k možnosti uživatelského ovládní systému.

Předposledním celkem jsou samostatné snímače. Snímače zajišťují převody jednotlivých fyzikálních veličin, na veličiny schopné měřit načítačem.

Posledním a neméně důležitým celkem je napájecí obvod sloužící k dodávce energie do každého z celků.

Z toho poznatku jsem celý systém rozdělil na pět částí. Jednotlivé části podrobněji popíšu v následujících kapitolách. Jednoduchou konfiguraci takového systému můžeme vidět na obrázku 1.3.

Na obrázku můžeme vidět, že základním celkem je načítač, který přijímá data od snímačů a tyto data zobrazuje na displeji. Dále je pak znázorněn komunikace s ostatními zařízeními. Napájecí celek dodává potřebnou energii do všech obvodů.



Obrázek 1.3: Návrh systému

### 1.2.1 Načítač

Z požadavků načítače vyplývá, že bude muset řídit celý systém. Z tohoto důvodu jsem se rozhodl, že se bude jednat o mikrokontrolér. Mikrokontroléry, jinak též označované jako jednočipové mikropočítače, obsahují v jediném pouzdře všechny podstatné části mikropočítače:

- Řadič a aritmetickou jednotku. Podle typu mikrokontroléru se používá délka slova 4, 8, 16 nebo 32 bitů.
- Paměť programu. Paměť programu je buď typu **EPROM** nebo **Flash**.
- Paměť dat typu R/W, někdy doplněnou i nevolatilní (paměť uchovává svůj obsah i po odpojení napájecího napětí) paměť **EEPROM**.
- Periferní obvody pro vstup a výstup dat.

### 1.2.2 Displej

Zobrazení našich naměřených dat je z hlediska uživatele nejdůležitější. Možností jak zobrazit naměřená data je několik. Podle mého úsudku jsem se rozhodl o členění do těchto stupňů:

- **Pomocí diod** (stupeň 1.) - Způsob, kdy data jsou například pomocí led diod zobrazována přímo v binárním kódu, je pro uživatele velice náročný. Vyžaduje to znalost binárního kódu a i při této znalosti je zobrazování velice nepřehledné. Výhodou je, že načítač spotřebuje pro toto zobrazení nejméně svých prostředků. Toto zobrazení je vhodné spíše při vývoji nebo při testování.
- **Znakový displej** (stupeň 2.) - Displej umožňující vypisovat znaky je mnohem uživatelsky přívětivější. Uživateli stačí znát pouze jazyk, ve kterém se data zobrazují. Pokud se použije tento displej již se zabudovaným řadičem, je jeho ovládání snadné a toto řešení nepotřebuje mnoho prostředků načítače.
- **Grafický displej** (stupeň 3.)- Displej umožňující ovládat všechny body displeje. Z tohoto vyplývá, že je možné zobrazit i grafické objekty. Z pohledu uživatele nejkomfortnější řešení. Některá data mohou být převedena na grafické objekty, které mají největší vypovídací hodnotu. Nevýhoda grafického displeje spočívá ve velké spotřebě prostředků načítače pro zobrazovaná data. Část těchto prostředků je ušetřena, pokud má grafický displej v sobě implementován řadič. Stále se však jedná o nejnáročnější zobrazení dat z těchto bodů.

Jak z předchozích bodů vyplývá, čím více chceme uživateli usnadnit čtení naměřených hodnot, tím více spotřebujeme prostředků pro zobrazení. Vždy se musíme rozhodnout, o kolik bude přínos přechodu na další stupeň zobrazení přínosnější, zda se nebude jednat jen o plýtvání prostředky načítače.

### 1.2.3 Snímače

Obecně můžeme snímače neelektrických veličin rozdělit z hlediska výstupu dat na analogové a digitální. Obě kategorie mají své přednosti i nevýhody. Pro svou další práci jsem se rozhodl, že budu uvažovat ve svém systému pouze s digitálními a to z několika důvodů:

- Digitální snímače vychází z výroby již předkalibrované a při výměně snímače za jiný není zpravidla nutno provádět novou recalibraci, protože jednotlivé kusy jsou svými parametry totožné a každý snímač při nahrání stejných parametrů vykazuje stejné vlastnosti.
- Další výhodou je rychlá diagnostika vadného snímače. Snímače jsou spojeny komunikačním rozhraním s načítačem a ten pomocí jejich stavů dokáže zjistit, zda nedošlo k poruše senzoru.
- Oproti analogovým snímačům není potřeba analogově digitální převodník, který by danou hodnotu převáděl na informace čitelné pro načítač.
- Možnost připojení více senzorů na jednu komunikační sběrnici. Při použití sběrnice, která to umožňuje.
- Většina ze senzorů obsahuje korekční algoritmy, které upraví výstupní hodnotu podle dalších vnějších vlivů (například vliv teploty).

Jako jistou nevýhodu u digitálních senzorů považují jejich cenu, která je oproti analogovým vyšší. Nicméně většina digitálních snímačů má analogové snímače doplněné o zabudovaný analogově digitální převodník, obvody vnitřní kalibrace a obvody komunikačního rozhraní. Tím se rozdíl ceny vyrovnává.

### 1.2.4 Napájecí obvod

Napájení všech ostatních částí systému zajistí jedna baterie. Toto napětí bude usměrněno na hodnotu všech okolních částí systému. Jako další možnost napájení je použití externího zdroje.

### 1.2.5 Ovládání

Jako ovládání jsem navrhl tři tlačítka, které budou obstarávat veškeré uživatelské vstupy. První tlačítko zastává funkci vstup do menu a potvrzování jednotlivých operací. Další tlačítka zajišťují vybrání jednotlivých akcí v menu a veškeré změny v operacích.

### 1.2.6 Komunikace

Ze zadání vyplývá, že měřicí zařízení musí být schopno komunikovat se svým okolím. Měřicí zařízení bude vysílat svá naměřená data. Dále bude schopno přijmout jednoduchý příkaz.

## Kapitola 2

# Návrh hardware

V této části se čtenář dozví o volbě hardwaru, který může reprezentovat navržený systém. V každé sekci bude uvedena i možná náhrada za jiný hardware, který by umožňoval stejnou funkci.

### 2.1 Načítač

Jak již bylo řečeno v kapitole 1.2.1 načítač bude realizován pomocí mikrokontroléru. V dnešní době se zabývá výrobou mikrokontroléru mnoho firem. Mezi nejznámější bych uvedl (Atmel, Mikrochip, Freescale, Motorola ...). Všichni tyto výrobci nabízí mikrokontrolér, který by umožňoval řídit uvedený systém. Pro hardwarovou reprezentaci načítače jsem se rozhodl pro firmu Atmel, konkrétně pro rodinu mikrokontrolérů AVR. S těmito mikrokontroléry pracuji již sedmý rok a vždy se podařilo najít řešení pro různé aplikace, které jsem vyvíjel.

#### 2.1.1 Atmel AVR ATmega

Zakladateli této technologie jsou dva studenti z Norského technického institutu - Alf-Egil Bogen a Vegard Wollan. Podle jejich jmen vznikla právě zkratka **AVR**, oficiálně ale znamená *Advanced Virtual RISC*.

**AVR** procesory jsou **RISC** procesory postavené na Harvardské architektuře (mají oddělenou paměť pro program a data). Skládají se z 32 stejných 8-mi bitových registrů, které mohou obsahovat jak data, tak i adresy. Jsou přímo propojeny s **ALU**, proto je **ALU** schopna provést jednu operaci v jednom hodinovém cyklu. Posledních 6 registrů je možno také po párech využít jako ukazatele pro nepřímé adresování do paměti. Hlavní vlastnosti **RISC** procesorů:

- Redukovaná sada instrukcí
- Obsahuje převážně jednoduché instrukce
- Délka provádění jedné instrukce je vždy jeden cyklus
- Délka (počet bitů) všech instrukcí je stejná
- Nepřímé adresování s dekrementací ukazatele po zpracování instrukce
- Mikroinstrukce jsou hardwarově implementovány na procesoru

- Využívá se zde techniky řetězení instrukcí

Většina instrukcí z instrukčního souboru je jednoslovných (16 bitových), pouze 4 instrukce jsou dvouslovné (32 bitové). To umožňuje rychlejší vykonávání instrukcí a také dovoluje jednodušší strukturu mikroprocesorového řadiče. Většina instrukcí umožňuje přímý jednocyklový přístup do všech registrů. Zpracování operandů probíhá následovně: Oba operandy se během taktu převedou z pracovních registrů, uskuteční se operace a výsledek se uloží do registru. Vždy se jedná o operace registr-registr, na kterých je **AVR** technologie založená. Toto má ale i několik záporů. Pouze některé instrukce umožňují pracovat s přímými datovými typy a to pouze s omezeným počtem registrů. Ostatní instrukce mohou pracovat pouze s registry. Výkon těchto mikropočítačů se blíží k 1 miliónu instrukcí za sekundu na 1 MHz, což je způsobeno vykonáváním jedné instrukce v jednom hodinovém cyklu.

Architektura **AVR** umožňuje 5 adresovacích módů:

- Přímé adresování
- Nepřímé adresování s 6ti-bitovým posunem
- Nepřímé adresování
- Nepřímé adresování s dekrementací ukazatele před zpravováním instrukce
- Nepřímé adresování s dekrementací ukazatele po zpravování instrukce

Mikropočítače Atmel Avr disponují třemi druhy pamětí.

- Paměť programu **FLASH**
- Paměť dat (přechodných) **RAM(RAM)**
- Paměť dat **EEPROM**

Paměť **RAM** (SRAM) slouží k uchovávání přechodných dat. Je rozdělena na několik desítek registrů (bajtů). Tato data mohou být pouze pracovní a nebo také řídicí. Většina registrů ovládá periferie a nebo samotný chod procesoru. Jiné zase informují o stavu, v jakém se procesor nachází.

Paměť programu, jak už název napovídá, slouží k uložení programu a nebo konstant, které v programu využíváme. Tato data se ukládají pouze za použití programátoru.

Paměť **EEPROM** slouží pro uchování dat, o která nechceme po vypnutí napájení přijít stejně jako je tomu například u **RAM**. Příkladem mohou být konfigurační data anebo kód v případě kódového zámku.

Současné **AVR** mají mnoho funkcí. Kompletní jejich výčet najdeme v [1]. Zde zmíním jenom některé.

- Multifunkční, obousměrné vstupně / výstupní obecné porty s konfigurovatelnými, zabudovanými rezistory
- Více vnitřních oscilátorů, včetně RC oscilátoru bez externích částí. Díky této vlastnosti se již nemusí zapojovat vnější RC oscilátor, ale při programování lze volit mezi přednastavenými frekvencemi (1,2,4,8) Hz.
- Vnitřní programovatelná Flash paměť až 256 KB

- Vnitřní **EEPROM** paměť až 4KB
- 8bitové a 16ti-bitové časovače
- Obsahuje různá sériová rozhraní viz. dále

## I<sup>2</sup>C

Rozborem této sběrnice se budu zabývat v kapitole o senzorech [2.3.1](#).

## UART / USART

Pro možnost propojení s externími zařízeními jsem se rozhodl použít sériovou linku **UART**. **UART** je obvod schopný obousměrné komunikace rychlostmi jednotek až desítek kbaud. Umožňuje asynchronní přenos s možností nastavení 5 až 9 datových bitů, nastavení počtu stop-bitů a parity. Umožňuje samozřejmě také multiprocesorovou komunikaci. Tato periferie nám umožní komunikaci s dalšími zařízeními, jelikož je často podporována různými výrobci mnoha zařízeních. Dnes již na trhu nalezneme převodníky **UART** ⇒ **USB**, tímto umožníme komunikaci s naprostou většinou osobních počítačů. Stejná situace nastane i s ostatními zařízeními.

Jednotka **USART** je složena ze tří hlavních bloků - vysílače, přijímače a generátoru hodin. Generátor hodin slouží pro generování synchronizačních signálů.

Vysílač a přijímač obsahují „buffery“ pro vysílání a příjem bajtů. Je-li obvod správně nastaven, je možné odvyšlat bajt pouhým zápisem do „bufferu“. Po odvyšlání celého bajtu je nastaven příznak **TXC** v registru **UCSRA**. Při příjmu je nastaven bit **RXC** registru **UCSRA**. Musíme ale dávat pozor na to, že příznaky jsou čitelné pouze jednou. Po přečtení jsou automaticky nulovány.

## Režimy

Jednotka **UART** může pracovat celkem ve čtyřech režimech:

- 1. asynchronní režim
- 2. asynchronní režim s dvojnásobnou rychlostí
- 3. synchronní „master“
- 4. synchronní „slave“

Pro komunikaci s externími zařízeními byl nastaven první režim. Tento režim není tolik závislý na přesnosti jako druhý režim a jelikož nebudeme posílat velké objemy dat, bohatě nám pro tento projekt postačí.

## Čítače / Časovače

Čítače a časovače jsou velmi důležitou součástí vestavěných systémů. Pokud jsou dobře nastaveny mohou obstarávat části programu, které je nutné provádět pravidelně aniž by zatěžovali hlavní část programu.

Procesory **ATMEGA** obsahují celkem tři čítače/časovače. Dva osmibitové, jeden šestnáctibitový. Vstupní signály mohou být předděleny předřazenou předděličkou a to v hodnotách 1, 8, 64, 256, 1024. Jedná se o samostatné části procesoru, nezávislé na probíhajících



instrukcích, které jsou schopny zvětšovat svůj obsah o 1, případně svůj obsah zmenšovat o tuto hodnotu. Každý blok má svoji specifickou funkci:

### Čítač

Čítá impulzy z vnějšího zdroje. Pravidelným čtením a nulováním (například za jednu vteřinu) můžeme měřit frekvenci tohoto signálu. Další možnosti je odměření času načítáním vnějších pravidelných impulzů.

### Časovač

Načítá frekvenci, která je určena vnitřním zdrojem hodinových impulzů a případnou předděličkou zařazenou do cesty. Tento způsob použití se nejčastěji využívá pro časování pravidelných intervalů.

**Input Capture** - slouží pro uložení stavu časovače / čítače. Jeho použití je ale u mikroprocesoru **ATMEGA16** omezeno na časovač 1. Pouze tato periferie je vybavena vstupem **ICP**, který umožňuje uložení stavu časovače.

**Output Capture** - tento komparátor porovnává registr časovače se srovnávacím registrem. Při shodě změní výstup OC svůj stav podle daného nastavení.

### Přerušeni

Přerušeni je reakce procesoru na určitou vnitřní nebo vnější událost. Při zachycení takovéto události je běh programu pozastaven, dojde k vykonání obsluhy přerušeni. Pokud tato obsluha skončí, dojde k obnovení běhu programu. Událostí, které lze sledovat je mnoho. Zde zmíním jenom ty hlavní - přetečení čítače/časovače, přijmutí bajtu po sériové lince, vnější přerušeni atd. Při vyvolání přerušeni program uloží adresu posledně provedené instrukce, skočí na adresu paměti programu, která je určena právě pro tento druh přerušeni. Adresa kam se skočí je uložena ve vektoru přerušeni. Na této adrese bývá téměř vždy umístěn skok na obslužnou rutinu. Pouze velice krátké rutiny (v řádech instrukcí) mohou být umístěny přímo na adrese, kam ukazuje vektor přerušeni. Po vykonání obslužné rutiny přerušeni se program vrátí na naposledy vykonávanou instrukci před příchodem přerušeni a program pokračuje.

### Porty

Mikrokontroléry Atmel **AVR** obsahují celkem čtyři multifunkční, obousměrné vstupně / výstupní obecné porty s konfigurovatelnými, zabudovanými rezistory. Těchto portů má celkem Porty jsou značeny písmeny A-D. Každý port je 8-mi bitový a po jeho nastavení ho můžeme ovládat. Většina portů má i alternativní funkce (vnější přerušeni, **ISP** programování, **UART** atd.). Jako první krok před použitím portu je nutné nastavit pomocí registru **DDR A-D** směr využití portu (logickou „1“ nastavíme v případě výstupu). Při čtení dat z portů se využívá registr **PIN A-D** a při zápisu dat **PORT A-D**. Při čtení z portů můžeme také povolit pullup rezistor. Tento rezistor nám usnadní práci například při čtení zda nebylo stisknuto tlačítko, které je zapojena na zem. Připojení rezistoru provedeme pomocí registru **PIN A-D** nastavením příslušného bitu do logické „1“.



- **R/W** - Bude se jednat o čtení nebo zápis.
- **E** - Hodinový signál sběrnice.
- **CS1** - Volba levé stany displeje .
- **CS2** - Volba pravé stany displeje.
- **RST** - Reset displeje.

Napájecí napětí displeje je 5V. Napětím na vstupu  $V_0$  lze měnit kontrast displeje. Nakonec dva vodiče pro osvětlení displeje. Viz. tabulka s čísly pinů modulu a jejich významem 2.1. Jelikož se jedná se o čistě grafický **LCD** displej, tak znakovou sadu je nutné vytvořit externě. Řešení tohoto problému se budu věnovat v následujících kapitolách.

Číslo	Název	Popis
1	$V_{SS}$	Napájení - zem
2	$V_{DD}$	Napájení - 5V
3	$V_0$	Nastavení kontrastu
4	D/I	Výběr mezi přenosem dat(log. „1“) nebo instrukcí (log.„0“)
5	R/W	Výběr mezi čtením dat(log. „1“) nebo zápisem (log.„0“)
6	E	Hodinový signál
7	DB0	Datový bit 0
8	DB1	Datový bit 1
9	DB2	Datový bit 2
10	DB3	Datový bit 3
11	DB4	Datový bit 4
12	DB5	Datový bit 5
13	DB6	Datový bit 6
14	DB7	Datový bit 7
15	CS1	Výběr levého řadiče (log. „1“)
16	CS2	Výběr pravého řadiče (log. „1“)
17	RST	Reset modulu (log. „0“) nebo normální funkce (log. „1“)
18	$V_{EE}$	Výstup záporného napětí (cca. -12V)
19	A	Anoda led diod pro podsvícení
20	K	Katoda led diod pro podsvícení

Tabulka 2.1: Popis jednotlivých vývodu modulu grafického displeje

### Komunikace s displejem

Celý displej je rozdělen na dvě části (64x64 pixelů). Každá část obsahuje vlastní řadič. Výběr těchto částí je pomocí řídicích signálů **CS1** a **CS2**. Všechny ostatní signály jsou vyvedeny do obou částí displeje. Pokud je aktivní signál **CS1** (log „1“), pracuje s levou stranou displeje. Pro pravou část displeje je zapotřebí aktivovat řídicí signál **CS2**. Displej komunikuje po 8-bitové paralelní sběrnici. Podle řídicích signálů **D/I** a **R/W** určíme, zda jsou na sběrnici data nebo instrukce, případně jestli se jedná o zápis nebo čtení. Záporným pulzem na vývodu **RST** se **LCD** vypne a vynulují se interní registry. Obsah obrazové paměti se nezmění, takže po zapnutí **LCD** instrukcí se objeví původní obraz. Tento vývod jsem se rozhodl nepřipojovat, a proto byl připojen na napájecí napětí.

Při zápisu instrukce nebo dat je byte ze sběrnice načten při sestupné hraně kladného impulzu na vývodu **E**, který by měl trvat alespoň 450ns. Stavové slovo obsahující "busy" příznak, stav resetu a zapnuto/vypnuto se čte následovně:

- 1. - Vývod **R/W** na logickou "1".
- 2. - Vývod **D/I** na logickou "0".
- 3. - Vývod **E** na logickou "1".
- 4. - Nyní se na sběrnici objeví stavové slovo. Význam jeho jednotlivých bitů je uveden v tabulce instrukcí (2.2.)
- 5. - Vývod **E** na logickou "0".

Při čtení dat se postupuje stejně, pouze vývod **D/I** musí být nastaven na log.1 a před vlastním čtením je třeba vygenerovat na **E** jeden kladný pulz.

Po připojení napájecího napětí není možné ihned vysílat instrukce. Je třeba generovat časovou prodlevu nebo neustále číst příznaky "busy" a "reset", dokud se nedostanou do logické "0".

### Instrukce displeje

Po instrukcích zápisu a čtení dat se automaticky zvyšuje hodnota čítače adresy v **X**. Po posledním (63) sloupci se začíná opět zleva (0). Stránky se automaticky neinkrementují. Registr pro posuv prvního řádku lze využít pro vertikální posouvání obsahu **LCD**.

Instrukce	D/I	R/W	DB 7-0							
Zapnutí displeje	0	0	0	0	1	1	1	1	1	1
Vypnutí displeje	0	0	0	0	1	1	1	1	1	0
Nastaví stránku paměti (0-7)	0	0	1	0	1	1	1	Stránka		
Nastaví pozici X (0-63) sloupce	0	0	0	1	Sloupec					
Nastaví pozici Y (0-63) řádky	0	0	1	1	Řádek					
Zápis Dat	1	0	Data							
Čtení dat	1	1	Data							
Stavové slovo displeje	0	1	<b>A</b>	0	<b>B</b>	<b>C</b>	0	0	0	0
A=0 Řadič zaneprázdněn, A=1 Řadič připraven B=0 LCD vypnut, B=1 LCD zapnut C=0 Probíhá reset, C=1 Reset ukončen										

Tabulka 2.2: Tabulka s instrukcemi displeje.

### Možná náhrada za ATM12864D

Bolymin - BG12864A

Podrobnější popis ovladače je uveden v [16]. V této části byly získány znalosti z [3] a [8].

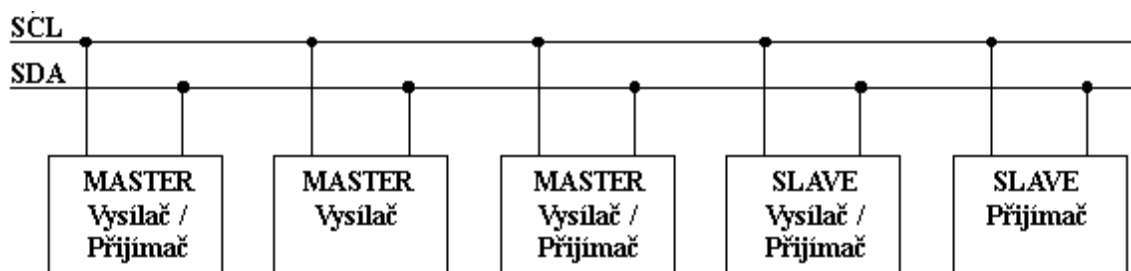
## 2.3 Snímače

Jako komunikační rozhraní jsem se rozhodl u všech senzorů použít sběrnici I<sup>2</sup>C, která umožňuje připojení více senzorů na jednu sběrnici a některé další výhody, které budou popsány níže v textu.

### 2.3.1 I<sup>2</sup>C

I<sup>2</sup>C bus je zkratka, která vznikla z **IIC** bus, tedy *Internal Integrated Circuit Bus*. Jak již název napovídá, jedná se o interní datovou sběrnici sloužící pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. Vyvinula ji firma Philips přibližně před 30 lety a od té doby prošla několika vylepšeními. V dnešní době tento způsob „komunikace“ podporuje řada integrovaných obvodů nejen firmy Philips. Jedná se především o mikrokontroléry, sériové paměti, inteligentní **LCD**, audio a video obvody, a/d a d/a převodníky a některé další digitálně řízené obvody. Hlavní výhodou je, že obousměrný přenos probíhá pouze po dvou vodičích - data **SDA** (*serial data*) a hodiny **SCL** (*serial clock*). To především u mikrokontrolérů výrazně optimalizuje nároky na počet „vstupně-výstupních“ pinů a celkově zjednodušuje výsledné zapojení. Na jednu sběrnici může být připojeno více integrovaných obvodů. V základní verzi jsou obvody adresovány 7bitově a v rozšířené verzi 10bitově. To umožňuje připojení 128, respektive 1024 čipů s různou adresou, na jednu společnou sběrnici. V praxi jsou tato čísla však podstatně nižší, protože adresa čipu většinou nelze určit plnými 7 (10), bity ale třeba jen třemi. Někdy nelze určit vůbec a je dána napevno pro daný typ čipu - takových čipů tedy na jedné sběrnici nemůže být více než jeden. Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100 *kHz*. Ve vylepšených verzích to může být 400 *kHz* nebo 1MHz, ale ne všechny integrované obvody tuto verzi podporují. Rychlost přenosu pak musí být přizpůsobena pochopitelně „nejpomalejšímu“ čipu na sběrnici. Oba vodiče musí být implicitně v logické jedničce a to je zajištěno „pull-up“ rezistory. Jejich odpory mají hodnotu v rádech jednotek (*kΩ*). Čím je vyšší komunikační frekvence, tím musí být nižší hodnoty těchto odporů. Pro 100kHz postačuje 4,7 *kΩ*.

Jeden z integrovaných obvodů (většinou mikrokontrolér) je nastaven jako „master“ a všechny ostatní obvody jsou „slave“. Obvody se dají zapojit i jako tzv. „multi-master“, kdy je čipů „master“ několik [2.2](#).

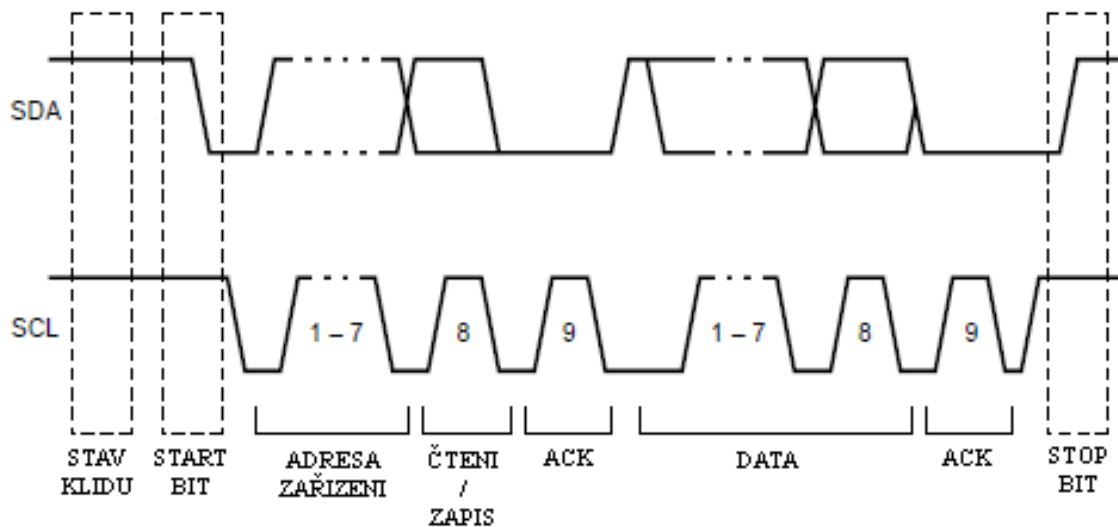


Obrázek 2.2: Možné zapojení různých zařízení

Zařízení typu „master“ při jakémkoli přenosu generuje hodinový signál na vodiči **SCL**. Když jeden čip vysílá, přijímají všechny ostatní a pouze podle adresy určují, zda jsou data určena jim. Čip, který chce vyslat/přijmout data musí nejprve definovat adresu čipu, s kterým chce komunikovat a zda půjde o příjem nebo vysílání - tedy o čtení nebo zápis. To

určuje **R/W** (čtení/zápis) bit, který je součástí adresy. Zde máme jednotlivé fáze přenosu, které mohou nastat při komunikaci na sběrnici:

- **stav klidu** - Je zajištěn logickými jedničkami na obou vodičích, „master“ tedy ne-generuje hodinový signál a neprobíhá žádný přenos. Logické jedničky jsou na obou vodičích zajištěny pull-up rezistory (rezistory mezi vodičem a napájecím napětím), takže klidový stav nastane, i pokud jsou výstupy obvodu „master“ ve stavu vysoké impedance (tedy odpojeny).
- **start bit** - Zahajuje přenos nebo jeho další část. Je vygenerován tak, že se změní úroveň **SDA** z „1“ na „0“, zatímco je **SCL** v logické „1“.
- **stop bit** - Ukončuje přenos. Je vygenerován podobně jako start bit. Logická úroveň **SDA** se změní z „0“ na „1“, zatímco je **SCL** v logické „1“. Stop bit může být generován pouze po „nepotvrzení přenosu“, tedy pouze po přijmutí Ack v logické „1“. (viz níže)
- **přenos dat/adresy** - Data jsou přenášena po 1 bajtu, tedy 8-mi po sobě jdoucích bitů, od nejvyššího po nejnižší. Při přenosu dat se může logická úroveň na **SDA** měnit pouze pokud je **SCL** v logické „0“. Při každém pulzu na **SCL** je přenesen jeden bit.
- **potvrzující bit Ack (acknowledge)** - Tento bit slouží k potvrzení správného přijmutí dat. Ack bit se odesílá stejným způsobem, jako by se odesílal devátý bit dat, ale s tím rozdílem, že ho generuje čip, který přijímal (příjímač) a nikoliv ten, který data odesílal. Pokud přenos proběhl v pořádku, tak odešle logickou „0“. Logická 0 potvrzujícího bitu znamená rovněž to, že je příjímač připraven na příjem dalšího bajtu, který následuje okamžitě po něm při dalším pulzu na **SCL**. Pokud přenos selhal, odešle logickou „1“ nebo pokud má dojít k ukončení přenosu, tak „neodešle nic“. Pull-up rezistor pak zajistí, že bude na **SDA** logická „1“ a Ack bit (v logické „0“) odešle vysílač. Časový průběh komunikace můžeme vidět na obrázku 2.3.



Obrázek 2.3: Časový průběh logických úrovní na vodičích SDA a SCL při jednotlivých fázích přenosu

Podrobnější popis sběrnice je uveden v [13].

### 2.3.2 Teploty

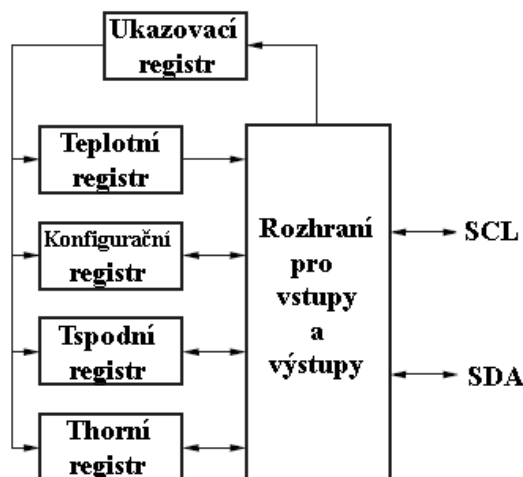
Pro snímání teploty jsem si vybral digitální teplotní senzor **TMP102** 2.4 od firmy Texas Instruments. Tento senzor vyniká svým teplotním rozsahem, který se pohybuje v rozmezí  $-40\text{ }^{\circ}\text{C}$  -  $125\text{ }^{\circ}\text{C}$ , s možností čtení teploty po  $0,0625\text{ }^{\circ}\text{C}$ . Dále disponuje digitálním výstupem s rozhraním I<sup>2</sup>C, přes který je možné celý senzor ovládat. Nízkou spotřebou, maximálně  $10\text{ }\mu\text{A}$  při aktivním stavu. Celý senzor může pracovat v 12ti nebo 13ti bitovém režimu, podle nastavení. Výstupem, který upozorňuje, že teplota je mimo rozsah.



Obrázek 2.4: Senzor TMP102 převzato z [14]

### Komunikace

Komunikace přes digitální rozhraní je realizována pomocí 5 registrů (viz obrázek 2.5). V prvním kroku je zapotřebí v ukazovacím registru nastavit bity P0 a P1 (viz tabulka 2.3) podle toho, který registr budeme používat (viz tabulka 2.4). Po provedení tohoto nastavení již můžeme číst nebo zapisovat do námi zvoleného registru.



Obrázek 2.5: Rozložení registrů

P7	P6	P5	P4	P3	P2	P1	P0
0	0	0	0	0	0	Registry	

Tabulka 2.3: Ukazovací registr - 8 bitů

V tomto systému registry Tspodní a Thorní nebudeme používat. Tyto registry slouží k hlídání, zda se teplota nedostala mimo horní, respektive dolní mez. Proto se dále zaměřím pouze na konfigurační registr a teplotní registr.

P1	P0	Registr	Čtení	Zápis	Délka
0	0	Teplotní registr	ANO	NE	16 bitů
0	1	Konfigurační registr	ANO	ANO	16 bitů
1	0	Tspodní registr	ANO	ANO	16 bitů
1	1	Thorní registr	ANO	ANO	16 bitů

Tabulka 2.4: Adresy ukazovacího registru

### Konfigurační registr

Konfigurační registr nám slouží k nastavování senzoru podle našich požadavků. Jedná se o 16ti bitový registr, kde nastavením jednotlivých bitů dosáhneme požadované funkčnosti. Pro tento systém nám vyhovují parametry nastavené přímo z výroby. Základní parametry jsou:

- 12ti bitový datový formát.
- Zjišťování teploty 4x za vteřinu.
- Výstup při chybě.

### Teplotní registr

V teplotním registru se nachází samotná teplota. Tento registr je opět 16ti bitový. První bajt bude celočíselná hodnota teploty. U druhého bajtu je zapotřebí provést korekci posledních 4 bitů z důvodu, že máme nastaven 12ti bitový datový formát. Pokud je teplota záporná (nejvyšší bit roven log. „1“), je nutné udělat dvojkový doplněk této hodnoty. Tabulka 2.5 s příklady některých teplot a jejich binárních reprezentací.

Teplota [°C]	Digitální výstup [binárně]
100	0110 0100 0000
0	0000 0000 0000
-25	1110 0111 0000

Tabulka 2.5: Příklad teplot

### Možná náhrada za TMP102

National Semiconductor - LM75CIM-5

Další případné nastavení nalezneme v [17].



### 2.3.3 Tlaku

Jako snímač atmosferického tlaku jsem použil úplnou novinku na trhu. Jedná se o senzor **MPL115A 2.6** firmy Freescale Semiconductor. Tento senzor je v české republice velice těžko k dostání, proto byl objednáán z německa. Tato nevýhoda je zapříčiněna krátkou dobou prodeje na trhu. Tento senzor se zatím prodává jen jako vývojový kit, to ale mému projektu nevádí. **MPL115A** je absolutní tlakový senzor s digitálním výstupem pomocí I<sup>2</sup>C pro nízkonákladové aplikace. Velkou výhodou tohoto senzoru jsou malé rozměry (5 x 3 x 1,2 mm), jeho minimální odběr proudu, který je maximálně 5  $\mu A$ . Rozsah senzoru je od 50 do 115 kPa. Tento rozsah nám plně dostačuje pro měření atmosferického tlaku. Integrovaný analogově digitální převodník nám poskytuje digitální hodnotu tlaku, teploty a kompenzační složky vhodné pro výpočet absolutního tlaku s přesností až 1 kPa. Hodnota tlaku a teploty je uložena na 10 bitech.



Obrázek 2.6: Kit KITMPL115A2I2C převzato z [6]

### Komunikace

Komunikace s tímto senzorem je opět pomocí sběrnice I<sup>2</sup>C. Skládá se ze tří základních částí:

- 1. Poslání příkazu pro začátek měření atmosferického tlaku a teploty.
- 2. Přečtení hodnot se senzoru
- 3. Výpočet absolutního tlaku

#### Část 1.

V této části senzoru pošleme příkaz pro zahájení měření.

- 0x10 Zahájení měření pouze tlaku.
- 0x11 Zahájení měření pouze teploty.
- 0x12 Zahájení měření tlaku i teplotu.

Poté je zapotřebí nejméně 30 milisekund vyčkat, než dojde k výpočtu.

#### Část 2.

V této části přečteme všechny hodnoty senzoru (viz. tabulka 2.6).

Adresa	Název	Velikost [bit]	Popis
0	Padc	8	Tlak z MPL115A - horní byte
1	Padc	2	Tlak z MPL115A - dolní byte
2	Tadc	8	Teplota z MPL115A - horní byte
3	Tadc	2	Teplota z MPL115A - dolní byte
4	A0	8	Offset koeficient tlaku - horní byte
5	A0	8	Tlak z MPL115A - dolní byte
6	B1	8	Koeficient citlivosti - horní byte
7	B1	8	Tlak z MPL115A - dolní byte
8	B2	8	1. offset koeficientu teploty - horní byte
9	B2	8	1. offset koeficientu teploty - dolní byte
10	C12	8	Součinitel pro koeficient citlivosti - horní byte
11	C12	8	Součinitel pro koeficient citlivosti - dolní byte
12	C11	8	Koeficient tlaku linearity 2. řádu - horní byte
13	C11	8	Koeficient tlaku linearity 2. řádu - dolní byte
14	C22	8	2. offset koeficientu teploty - horní byte
15	C22	8	2. offset koeficientu teploty - dolní byte

Tabulka 2.6: Vnitřní hodnoty senzoru

### Část 3.

Zde musíme provést výpočet absolutního tlaku a to podle rovnice 2.1.

$$P_{comp} = a_0 + (b_1 + c_{11} * P_{adc} + c_{12} * T_{adc}) * P_{adc} + (b_2 + c_{22} * T_{adc}) * T_{adc} \quad (2.1)$$

Poté převedeme na dekadickou hodnotu:

$$decP_{comp} = ((65.0/1023.0) * P_{comp}) + 50 \quad (2.2)$$

Vysvětlení konstant

- 65 - Rozdíl největšího a nejmenšího tlaku (115 kPa - 50 kPa)
- 1023 - Velikost čítače analogově digitálního převodníku
- 50 - Počáteční tlak (senzor měří od 50 kPa)

Tímto dostaneme výsledný absolutní atmosferický tlak

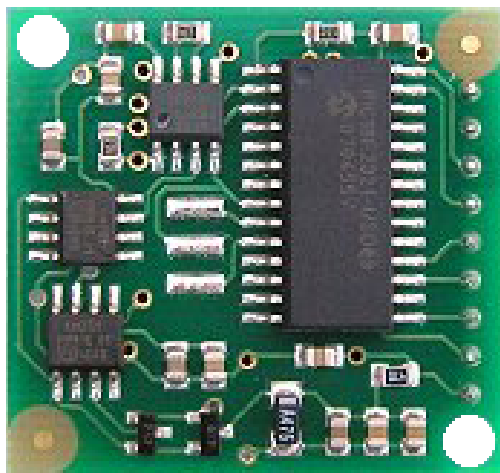
### Možná náhrada za CMPS03

Freescale Semiconductor - MPXY8000

Více informací nalezneme v [7].

### 2.3.4 Úhlové orientace

Pro úhlovou orientaci vůči zemskému povrchu byly použity dva magnetorezistivní senzory **KMZ51** od firmy Philips. Tento senzor má dostatečnou citlivost, aby dokázal snímat magnetické pole Země. Ze signálu dvou navzájem kolmých senzorů se vypočítává směr horizontální složky geomagnetického pole (viz kapitola 1.1.1). Tyto senzory se nachází v modulu **CMPS03 2.7** od firmy Devantech. Spolu s nimi se na modulu nachází i mikrokontrolér, který se stará o výpočet azimutu a komunikaci s okolím pomocí sběrnice I<sup>2</sup>C. Dále tento modul umožňuje i číst hodnotu azimutu pomocí pulzně šířkové modulace.



Obrázek 2.7: Senzor CMPS03 převzato z [19]

#### Komunikace

Modul komunikuje pomocí sběrnice I<sup>2</sup>C. Obsahuje registry, ze kterých je možno číst nebo zapisovat. Obsah jednotlivých registrů je uveden v tabulce 2.7. Azimut lze z těchto registrů získat v rozlišení na jeden nebo dva bajty. Tímto určíme přesnost měření. Pokud zvolíme jeden bajt přesnost bude:

$$presnost1b = \frac{360}{255} = 1,411 \quad (2.3)$$

pokud zvolíme dva bajty:

$$presnost2b = \frac{360}{3599} = 0,1 \quad (2.4)$$

#### Možná náhrada za CMPS03

HONEYWELL S&C - ASDXAVX010NG2A5

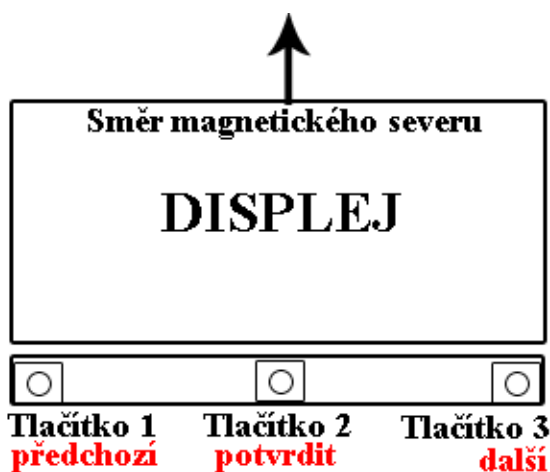
V sekci bylo čerpáno z [15]

Adresa	Popis	Čtení	Zápis
0	Verze software	ANO	NE
1	Azimut (0-255)	ANO	NE
2	Azimut horní byte (0-3599)	ANO	NE
3	Azimut dolní byte (0-3599)	ANO	NE
4-11	Interní testy	ANO	ANO
12-14	Příkazy modulu	ANO	ANO
15	Kalibrace modulu nebo změna adresy	ANO	ANO

Tabulka 2.7: Registry modulu kompasu CMPS03

## 2.4 Ovládání

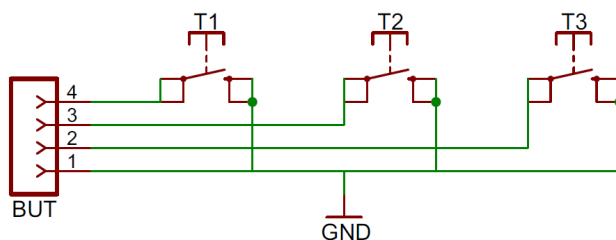
Ovládání tohoto modulu je realizováno pomocí tří tlačítek, které jsou v řadě vedle sebe [2.8](#). Tyto tlačítka jsou umístěny pod displejem a umožňují jednoduché ovládání celého modulu.



Obrázek 2.8: Rozvržení tlačítek a určení směru magnetického severu.

### Schéma zapojení

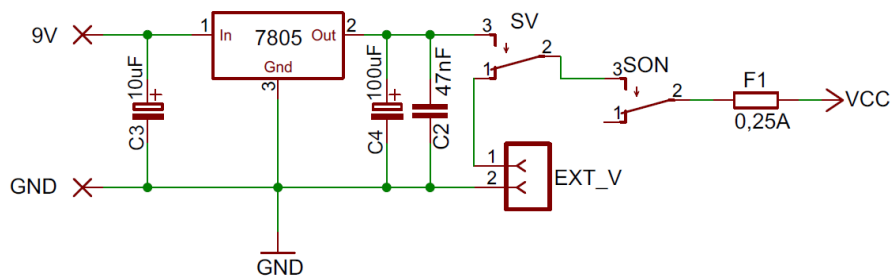
Schéma zapojení tlačítek je velice jednoduché. Můžeme ho vidět na obrázku [2.9](#). Všechny tlačítka mají svůj výstup a jsou připojena proti napájecí zemi.



Obrázek 2.9: Schéma zapojení tlačítek

## 2.5 Napájecí obvod

Jak už bylo zmíněno v 1.2.4, napájení obvodu je realizováno 9V baterií, která je usměrněna pomocí integrovaného obvodu **7805** na hodnotu 5V. Toto napětí je dále rozvedeno do všech částí měřícího zařízení. Dále je pak možné využít externí zdroj napájení, který musí mít hodnotu 5V. Pro volbu mezi napájením z baterie nebo externím zdrojem slouží přepínač. Další přepínač je určen pro zapnutí nebo vypnutí celého měřícího zařízení. Celý napájecí obvod je jištěn pojistkou 250 mA, která v případě poruchy nebo zkratu zajistí, aby nedošlo ke zničení celého měřícího zařízení. Schéma zapojení napájecího obvodu můžeme vidět na obrázku 2.10. V této části byly použity znalosti z [12].



Obrázek 2.10: Schéma zapojení napájecího obvodu

## 2.6 Deska plošných spojů

Při navrhování desky plošných spojů jsem vycházel z vytvoření dvou vrstev nad sebou, kdy první je modul displeje a pod ním je vrstva s ostatními částmi měřícího zařízení. Z tohoto důvodu musela velikost jednotlivých vrstev sobě odpovídat. Jejich jednotný rozměr je 70 x 93 mm. Dále bylo nutné umístit díry pro distanční sloupky přesně na sebe, aby bylo možné jejich spojení. Jednotlivé vrstvy jsou k sobě přišroubovány přes distanční sloupky o velikosti 12 mm. Abych dále dodržel možnost měnění jednotlivých senzorů, displeje a načítače, jsou tyto části upevněny do patice. Modul pro měření úhlové orientace vůči zemskému povrchu 2.3.4 musí být umístěn tak, aby magnetický server směřoval horní stranou displeje od uživatele (viz. obrázek 2.8). Možnost výměny jednotlivých částí tak umožní v případě poruchy nebo vylepšení jejich snadnou výměnu a celé měřící zařízení může pracovat dále. Pro připojení dalších případných senzorů nebo jiných zařízení komunikujících po sběrnici I<sup>2</sup>C, je tato sběrnice vyvedena pomocí pinů spolu s napájecím napětím. Pro možnost programovat načítač přímo v měřícím zařízení, je vyvedeno rozhraní pro sériové programování **ISP**, které implementoval Atmel hlavně do rodiny procesorů **AVR**. Vstup/Výstupní části měřícího zařízení:

- **CMPS03** - Patice pro připojení modulu kompasu 2.3.4
- **TMP102** - Patice pro připojení modulu teplotního čidla 2.3.2
- **MPL115AI** - Patice pro připojení modulu pro měření atmosferického tlaku (komunikace) 2.3.3
- **MPL115AV** - Patice pro připojení modulu pro měření atmosferického tlaku (napájení) 2.3.3

- **LCD** - Patice pro připojení modulu displeje [2.2.1](#)
- **ATMEGA** - Patice pro připojení modulu displeje [2.1.1](#)
- **ISP** - Rozhraní pro sériové programování načítače [2.1.1](#)
- **I2C 1-3** - Možnost připojení externích zařízení na sběrnici I<sup>2</sup>C (3x)
- **BUT** - Patice pro připojení tlačítek [2.4](#)
- **EXTV** - Možnost připojení externího napájení [2.5](#)
- **EXTD** - Odesílání dat

Podklady pro výrobu desky plošných spojů nalezneme v příloze [A](#).

## Kapitola 3

# Software měřicího zařízení

Samotný hardware měřicího zařízení udává jeho možnosti, ale funkce mu dodá jeho software. Cílem této kapitoly je tento software popsat a tím objasnit jednotlivé funkce měřicího zařízení.

Celý program procesoru je napsán v programovacím jazyku C. Tento okomentovaný program je součástí příloh bakalářské práce. Při vývoji byl program vzhledem k nutnosti komunikace s perifériemi laděn a testován v reálném čase přímo v měřicím zařízení. Ke kompilaci jsem použil volně dostupný kompilátor **AVR-GCC** [18]. Celý software byl vyvíjen v **AVR Studiu 4.18** [2], který umožňuje vytváření programových projektů, které zapouzdřují jeho jednotlivé části (soubory s programovými kódy, informace o projektu, textové soubory, atd.).

Celý software měřicího zařízení je umístěn v hlavní části systému načítači **1.2.1**. Načítač je řešen pomocí mikrokontroléru viz. **2.1.1**. Momentálně umístěný mikrokontrolér **Atmega AVR 16-PU**. Celý systém byl odskoušen i s mikrokontrolérem **Atmega AVR 32-PU** se stejnou funkčností.

### 3.1 Struktura zdrojových souboru a klíčové funkce

V této části kapitoly popíší strukturu zdrojových souborů a klíčové funkce. Jsou popisovány jen základní vlastnosti funkcí, celkový jejich výpis není z důvodu jejich obsáhlosti uváděn. Pro podrobný popis, typy parametru a návratových hodnot, odkazují čtenáře na přílohy.

#### 3.1.1 config.h

Tento soubor obsahuje veškeré nastavení programu. Jedná se vždy o konstanty, které jsou používány v celém programu. Nesprávným nastavením těchto hodnot může dojít k úplné nefunkčnosti měřicího zařízení.

#### Možnosti nastavení

- Definice velikosti **LCD**
- Definice připojení modulu displeje k mikropočítači
- Definice konstant příkazů pro modul displeje
- Definice připojení tlačítek k mikropočítači

- Definice připojení led diod k mikropočítači
- Definice konstant pro nastavení modulu pro měření teploty **TMP102**
- Definice I<sup>2</sup>C adres všech připojených senzorů
- Definice nastavení pro I<sup>2</sup>C sběrnici
- Definice znaků a výpisů
- Definice grafického rozložení menu
- Definice nastavení sériové linky **UARTU**
- Definice chyb pro jednotlivé senzory
- Definice velikosti jednotlivých „bufferu“

### 3.1.2 tables.h

Definice tabulek, které jsou potřeba při běhu programu. Tyto tabulky jsou speciálně ukládány v paměti **ROM** a pro jejich čtení jsou využívány instrukce pro přístup do **ROM** paměti, aby tyto tabulky nebyly zbytečně kopírovány při inicializaci mikrokontroléru z **ROM** paměti do **RAM**. Tímto způsobem ušetříme mnoho paměti **RAM**. Pro tyto účely nám v překladači **AVR-GCC** slouží knihovna `pgmspace.h`. Jsou zde uloženy dvě tabulky:

- **font** - Definice jednotlivých znaků pro textový režim displeje
- **logo** - Obrázek s logem projektu o velikosti 128x64 bitů.

### 3.1.3 main.c

Hlavní funkce celé softwarové části. Stará se o prvotní inicializaci všech částí. Poté v nekonečném cyklu zjišťuje hodnoty na senzorech a zajišťuje jejich výpis. V nekonečné smyčce také dochází ke zjišťování, zda nebylo stisknuto tlačítko pro vstup do menu a případná obsluha v menu.

### 3.1.4 lcd.c

Funkce pro práci s řadičem **KS0108** grafického displeje. Funkce implementují i znakový režim, který pro modul displeje 2.2.1 nebyl hardwarově implementován. V tomto režimu má displej 7 řádků. Velikost písmen je 5x7 bodů.

#### **Funkce ks0108\_Initialize(void)**

Funkce inicializuje displej.

#### **Funkce ks0108\_WriteData(data)**

Funkce zapíše data na displej.

#### **Funkce ks0108\_WriteCommand(prikaz,radic)**

Funkce pošle do řadiče příkaz.



### **Funkce ks0108\_WriteChar(znak,invert)**

Funkce na displeji zobrazí znak.

### **Funkce ks0108\_SetPixel(x,y,barva)**

Funkce zobrazí na displeji bod (x,y).

### **3.1.5 grafic.c**

Funkce pro zobrazení základních grafických entit na displeji s řadičem **KS0108**: úsečka, kruh, obdélník a kompas. Nejprve vykreslí oba boky obdélníku a poté obě základny.

### **Bresenhamův algoritmus**

Tento algoritmus byl objeven v šedesátých letech minulého století panem Bresenhamem. Bresenhamův algoritmus spočívá v tom, že se při rasterizaci nachází body ležící nejbližší skutečné úsečce pouze pomocí celočíselné aritmetiky. Jako příklad postupu uvažujme úsečku kdy levý bod leží níže než pravý. Pro nakreslení levého koncového bodu úsečky je třeba zvolit, zda další bod vpravo musí být vykreslen na pozici se stejnou souřadnicí y nebo se souřadnicí o jedničku větší. Pixel, který vybereme je ten, jehož souřadnice y se více blíží skutečné hodnotě y na dané úsečce.

### **Funkce ks0108\_Rectangle(x,y,b,a,barva)**

Funkce vykreslí na displej obdélník s délkami stran (a,b). Jeho levý roh bude na bodě (x,y).

### **ks0108\_Line(x1,y1,x2,y2,barva)**

Funkce vykreslí úsečku pomocí „Bresenhamova algoritmu“. (x1,y1) první bod úsečky, (x2,y2) druhý bod úsečky.

### **Funkce ks0108\_Circle(cx,cy,r)**

Funkce vykreslí kruh pomocí algoritmu midpoint se středem (cx,cy) a poloměrem r. Při výpočtu bodů na kružnici nám stačí nalézt souřadnice oblouku 45 stupňů, neboť všechny další body dostaneme záměnou souřadnic a změnou znamének. Princip zjištění bodu je opět na základě Bresenhamova algoritmu pro úsečky. Jsme ve výchozím pixelu a ptáme se, kam teď můžu jít a testujeme, která pozice je blíž vykreslované kružnici. Výhodou tohoto postupu je opět rychlost, neboť vždy pracujeme pouze s celými čísly.

### **3.1.6 i2c.c**

Funkce pro práci s I<sup>2</sup>C sběrnici.

### **Funkce i2c\_init()**

Funkce, která inicializuje sběrnici I<sup>2</sup>C. Nastavuje rychlost na 100 kHz

### **Funkce `i2c_start(adresa)`**

Funkce zahájí komunikaci s měřícím zařízením na sběrnici I<sup>2</sup>C. Pokud se bude jednat o čtení posílá pouze I<sup>2</sup>C adresu zařízení, pokud se jedná o zápis je nutné k adrese přidat jedničku

### **Funkce `i2c_stop()`**

Funkce, která ukončí komunikaci na sběrnici I<sup>2</sup>C.

### **Funkce `i2c_write(data)`**

Funkce posle byte na sběrnici I<sup>2</sup>C.

### **Funkce `i2c_readAck()` a `i2c_readNAk()`**

Funkce přijme bajt ze sběrnice I<sup>2</sup>C pokud se jedná o Ack bude následovat další bajt. Nak přijme pouze jeden bajt.

### **3.1.7 `senzors.c`**

Funkce pro práci se senzory **TMP102**, **CMPS03**, **MPL115A2T1**.

### **Funkce `(Kompas, Tlak, Tmp102)_print_data(temp)`**

Funkce zobrazí hodnotu ze sensorů.

### **Funkce `(Kompas, Tlak, Tmp102)_get_data(alti)`**

Funkce přečte data ze sensorů, zpracuje je a uloží.

### **Funkce `print_azimut(azi,azi_l)`**

Funkce zobrazí šipku v kompasu.

### **Funkce `lad_vypis(zarizeni,p_reg,s_reg,text)`**

Funkce pro výpis registru zařízení připojených na I<sup>2</sup>C sběrnici.

### **Funkce `calibration()`**

Funkce pro kalibraci kompasu.

### **Funkce `check_senzors()`**

Funkce, která vypíše čísla všech připojených zařízení na sběrnici I<sup>2</sup>C.

### **3.1.8 `uart.c`**

Funkce pro ovládání sériové linky. Ukázkou komunikace po sériové lince s osobním počítačem nalezneme v [C.1](#).

### **Funkce `uart_init()`**

Funkce inicializuje sériovou linku pro asynchronní přenos s volitelnou rychlostí o rozsahu 5400-19200 kbaud. Rámec dat: 8 datových, 1 stop bit, bez parity.

### **Funkce `uart_putc(data)`**

Funkce pošle data pomocí sériové linky.

### **Přerušování `USART_RXC_vect`**

Pomocí tohoto přerušování měřicí zařízení sleduje komunikaci na sériové lince. Pokud přijde příkaz, okamžitě tento příkaz provede.

### **Funkce `check_command()`**

Funkce pro dekódování příkazů. Každý příkaz musí končit speciálním znakem %.

### **3.1.9 `delay.c`**

Funkce s definicí zpoždění.

### **Funkce `delay_but(num)`**

Funkce čeká po dobu  $\text{num} * 1000$  na stisknuté tlačítko.

### **Funkce `wait_but_press(but)`**

Funkce čeká, dokud není stisknuto tlačítko. Po stisknutí vrací jeho číslo. Dále funkce umožňuje čekat pouze na určité tlačítko nebo na všechny tlačítka.

### **3.1.10 `menu.c`**

Funkce pro práci v menu

### **Funkce `menu()`**

Funkce, která zobrazí menu a spravuje všechny akce v něm. Během menu dochází ke zastavení činnosti kompasu.

### **3.1.11 `initialize.c`**

Funkce které inicializují všechny komponenty.

## **3.2 Běh programu**

Celý běh programu můžeme rozdělit do čtyřech základních částí. Těmto částem se budu věnovat dále.

## Inicializace

Jako první po spuštění programu je zapotřebí inicializovat jednotlivé komponenty. Pokud by tento krok nenastal, jednotlivé komponenty by nefungovali. Nejprve se nastaví porty, které slouží pro výstup připojených na led diody. Poté se nastaví vstupní porty pro tlačítka. Jako další se nastaví hodnoty časovače pro přerušení programu na jednu za vteřinu a dojde k vynulování času. Od této doby běží časovač. Následuje posláni příkazu do displeje o zapnutí. Po potvrzení zapnutí displeje dojde k nastavení I<sup>2</sup>C sběrnice. Posledním krokem se nastaví sériová linka pro externí komunikaci. Tímto krokem je ukončena inicializace programu.

## Načtení a zpracování dat se senzorů

Po inicializaci se vstoupí do nekonečné smyčky, která po jedné vteřině posílá žádosti do všech senzorů o nové hodnoty. Výjimku tvoří tlakový senzor. Hodnotu tohoto senzoru není zapotřebí zjišťovat tak často, proto je jeho hodnota zjišťována po delším intervalu. Při jednotlivém čtení hodnot dojde i k jejím zpracování a upravení přesnosti na desetiny.

## Zobrazení výsledků

Poté, co jsou zpracovány hodnoty, dojde k zápisu na displej. U výpisu hodnoty azimutu navíc dojde k překreslení šipky v grafickém prvku kompasu. Při prvních pokusech se celý kompas překreslil, toto řešení ovšem mělo za následek, že kompas nepříjemně blikal vlivem pomalé frekvence. Pro vyřešení tohoto problému jsem přestal mazat celý kompas, ale mažu pouze předchozí šipku. Pro toto řešení je nutno si pamatovat předchozí hodnotu azimutu, nicméně problém byl tímto odstraněn.

## Čekání na vstup do menu

V této části zjišťujeme přibližně jednu vteřinu, zda nebylo stisknuto tlačítko pro vstup do menu. Pokud bylo, tak vstoupíme do menu a přestane obsluha senzorů. V této části programu čekáme na výběr položky od uživatele a dochází k přerušení od časovače. Nedojde-li ke stisknutí tlačítka, program pokračuje opět načítáním hodnot.

## Přerušení

V programu jsou použity obsluhy celkem dvou přerušení. První přerušení je od časovače. Toto přerušení je periodicky voláno s frekvencí 1 Hz. V jeho obsluze dojde k zvýšení hodnoty času o jednu vteřinu. Pro nastavení hodnoty časovače jsem použil vzorec 3.1.

$$\text{Hodnota} = \frac{F_{cu}}{DF_{cu}} = \frac{8000000}{256} = 31250 \quad (3.1)$$

**F<sub>cu</sub>** - Frekvence mikropočítače , **DF<sub>cu</sub>** - Nastavená dělička

Druhé přerušení slouží k přijímání znaků přes sériovou linku. Toto přerušení se zavolá vždy, když se na přijímací straně sériové linky objeví znak. Tento znak je uložen do kruhového „bufferu“. Pokud je v „bufferu“ příkaz, tak se vykoná a „buffer“ se vymaže. Příkaz se pozná podle speciálního znaku na konci %.

## Kapitola 4

# Uživatelské rozhraní

V této kapitole čtenáře seznámím s volbou uživatelského rozhraní.

### 4.1 Zobrazení

Při volbě uživatelského rozhraní byl největší důraz kladen na přehlednost a jednoduchost. Všechny naměřené hodnoty jsou zobrazovány v textové podobě na pravé části displeje. Na displeji jsou zobrazovány tyto hodnoty:

- **time** - Hodnota času od spuštění měření času.
- **azim** - Hodnota azimutu s přesností na desetiny stupně.
- **temp** - Hodnota teploty s přesností na desetiny stupně.
- **alti** - Hodnota absolutního atmosferického tlaku v hektopascalech.

Na Levé straně displeje se nachází grafické zobrazení kompasu, kde hodnotu azimutu ukazuje šipka. Tato šipka ukazuje momentální směr natočení měřícího zařízení vůči zemskému povrchu. Obrázek zobrazení hlavní obrazovky v měřícím zařízení nalezneme v [B.3](#).

### 4.2 Menu

V menu nalezneme celkem šest položek. Vybráním některé z nich se dostaneme do jednotlivých akcí v menu. Po ukončení akce se opět vrátíme na základní obrazovku. Ukázka obrazovky menu [B.4](#)

#### LOGO

V této akci se na displej zobrazí logo projektu. Obrázek můžeme vidět v [B.5](#).

#### TIME

Nastavení času na výchozí hodnotu (00:00:00).

## **SHOW**

Vypíše registry všech připojených zařízení. Registry jsou vypisovány po bajtech. První se zobrazí registry teplotního senzoru, po potvrzení se objeví tlakového čidla a nakonec modulu kompasu.

## **SEAR**

Vyhledá na I<sup>2</sup>C sběrnici všechna připojená zařízení a vypíše jejich adresu.

## **CALI**

Proběhne kalibrace kompasu. Pro úspěšnou kalibraci je zapotřebí znát polohu všech čtyř světových stran. Zadávání je rozděleno do čtyř kroků. Mezi každým krokem je přibližně dvousekundová mezera pro uložení směru.

- V prvním kroku je zapotřebí namířit měřící zařízení na sever a stisknout potvrzovací tlačítko.
- Druhý krok je namíření měřícího zařízení na východ.
- Třetí krok je namíření měřícího zařízení na Jih.
- V posledním kroku namíříme měřící zařízení na západ.

Celé zadávání doprovází pokyny na displeji. Je důležité mířit přesně na světové strany, jinak hrozí špatné spočítání hodnot a tím nefunkčnost kompasu. Dále je důležité, aby se celý proces kalibrace dokončil, jinak opět hrozí nefunkčnost kompasu.

## **BACK**

Návrat na základní obrazovku.

# Kapitola 5

## Závěr

### 5.1 Dosažené výsledky

Na základě návrhu v druhé a třetí kapitole byl sestaven funkční vzorek měřícího zařízení pro měření teploty vzduchu, tlaku vzduchu a úhlové orientaci vůči zemskému povrchu. Jeho hlavní parametry shrnuje následující přehled:

- Stejnosměrné napájecí napětí 9 V - baterie, případně 5 V externí zdroj.
- Proudový odběr ze zdroje 9 V - přibližně 80 mA při vypnutém podsvícení a 100 mA se zapnutým podsvícením displeje.
- Možnost připojení dalších senzorů.
- Vestavěné senzory pro měření tlaku vzduchu, teploty vzduchu a senzor pro měření úhlové orientace vůči zemskému povrchu.
- Schopnost posílat naměřená data dalším zařízením.
- Grafický displej o velikosti 128x64 bodů. S grafickým i textovým zobrazením naměřených hodnot.
- Možnost kalibrace kompasu.

Měřící zařízení je možno softwarově rozšířit o podporu dalších senzorů, pracujících na sběrnici I<sup>2</sup>C. Těchto senzorů je možné připojit až 124 přičemž se nesmí jejich adresy shodovat. Měřící zařízení je vybaveno programovacím vstupem pomocí rozhraní **ISP**. Tímto se stává celé měřící zařízení ještě více univerzálnější.

Jako nevýhodu zařazení bych viděl malou výdrž baterie, která vydrží napájet celé měřící zařízení přibližně 2 hodiny. Tento problém by vyřešila baterie s vyšší kapacitou a přejítí na nižší napájecí napětí všech částí měřícího zařízení.

#### Softwarová část systému

Jak již jsem zmínil v úvodu, celý systém byl vyvíjen poměrně dlouhou dobu a postupem času byl notně vylepšován. V jakémkoliv programu je možno udělat mnoho chyb. Během vývoje bylo mnoho chyb odhaleno, další ale mohly zůstat opomenuty, či mohly být přidány nové. Jelikož se jedná o funkční systém, budu v jeho vývoji nebo alespoň opravě chyb co nejdélejší dobu pokračovat. V aktuální verzi s reprezentací načítače pomocí mikrokontroléru

**Amtel AVR ATmega 16** jsem se dostal na 92 % zaplnění paměti pro program. Z tohoto důvodu je zapotřebí při případném rozšiřování systému přejít na nový mikrokontrolér, který vlastní větší paměť. Například **Amtel AVR ATmega 32**, se kterým byl celý systém také testován.

## 5.2 Možné vylepšení

Jako možné vylepšení měřicího zařízení a tedy i další cestu vývoje bych viděl softwarovou implementaci pro více senzorů nejrůznějších fyzikálních veličin. Připojení modulu **GPS** navigace pro ještě lepší orientaci v prostoru. Dále pak možnost ukládání naměřených hodnot pro pozdější analýzu. Námětů na zlepšování měřicího zařízení je mnoho a doufám, že se mi nějaké podaří zrealizovat.

## 5.3 Shrnutí

Výsledkem této práce měl být návrh systému pro měření teploty vzduchu, tlaku vzduchu, úhlové orientaci vůči zemskému povrchu. Úvodní kapitola vytýčila směr návrhu, který měl vést k vytvoření systému splňujícího požadavky zadání. V druhé a třetí kapitole byl popsán návrh hardwaru a softwaru měřicího zařízení pro požadované veličiny. Podle návrhu byl zkonstruován funkční vzorek hardwaru, na němž byla ověřena jeho funkce. Měřicí zařízení plně komunikuje se všemi senzory a zobrazuje jejich naměřenou hodnotu. Měřicí zařízení nemůže konkurovat komerčně vyráběným zařízením, které mají nižší hmotnost, velikost, spotřebu atd. Nicméně to ani nebylo účelem návrhu. Na fotky celého měřicího zařízení se můžeme podívat v příloze [B.1](#) a [B.2](#).



# Literatura

- [1] Atmel: Atmel AVR.  
URL <http://www.atmel.com/products/AVR/>
- [2] AVR Studio: AVR Studio. WWW Stránky.  
URL [http://www.atmel.com/dyn/products/tools\\_card\\_v2.asp?tool\\_id=2725&source=redirect](http://www.atmel.com/dyn/products/tools_card_v2.asp?tool_id=2725&source=redirect)
- [3] David, M.: *Práce s inteligentními displeji LCD*. BEN - technická literatura, 2006, iSBN 80-7300-121-7.
- [4] David, M.: *Práce s mikrokontroléry Atmel AVR*. BEN - technická literatura, 2006, iSBN 80-7300-209-4.
- [5] David, M.: *Práce s mikrokontroléry ATMEL AVR - ATmega16*. BEN - technická literatura, 2006, iSBN 80-7300-174-8.
- [6] EBV Elektronik: EBV Elektronik.  
URL [www.ebv.com/typo3temp/pics/313336e702.jpg](http://www.ebv.com/typo3temp/pics/313336e702.jpg)
- [7] Freescale Semiconductor: How to Implement the Freescale MPL115A Digital Barometer. [online], 2009-07-05 [cit. 2010-04-03].  
URL [www.freescale.com/files/sensors/doc/app\\_note/AN3785.pdf](http://www.freescale.com/files/sensors/doc/app_note/AN3785.pdf)
- [8] Hebei I. T. (Shanghai) Co., Ltd.: ATM12864D Liquid Crystal Display Module. [online], 2003.  
URL [www.hebeiltd.com.cn/lcm.datasheet/ATM12864D.pdf](http://www.hebeiltd.com.cn/lcm.datasheet/ATM12864D.pdf)
- [9] Jan, B.; Jaroslav, K.: *Jak vzniká počasí*. Karolinum, 2005, iSBN 80-246-1002-7.
- [10] Josip, K.: *Velká encyklopedie vesmíru*. Academia, 2002, iSBN 80-200-0906-X.
- [11] Karban, P.; Šafář, D.; Nováček, P.; aj.: Měření horizontální složky magnetické indukce Země. [online], 2007.  
URL [http://home.zcu.cz/~karban/teaching/te2/mag\\_indukce/mag\\_indukce\\_zeme.pdf](http://home.zcu.cz/~karban/teaching/te2/mag_indukce/mag_indukce_zeme.pdf)
- [12] ON Semiconductor: MC7800, MC7800A, NCV7805. [online].  
URL [http://www.gme.cz/\\_dokumentace/dokumenty/934/934-051/dsh.934-051.1.pdf](http://www.gme.cz/_dokumentace/dokumenty/934/934-051/dsh.934-051.1.pdf)
- [13] Philips Semiconductors: THE I2C-BUS SPECIFICATION. [online], 2000-01-01 [cit. 2010-03-03].  
URL [http://www.nxp.com/acrobat\\_download2/literature/9398/39340011.pdf](http://www.nxp.com/acrobat_download2/literature/9398/39340011.pdf)

- [14] PV Electronic: PV Electronic.  
URL <http://www.pvelectronic.inshop.cz/inshop/catalogue/products/pictures/09418-02.jpg>
- [15] Robot Electronics: CMPS03 - Compass Module. [WWW Stránky], 2006-04-13 [cit. 2010-03-03].  
URL <http://www.robot-electronics.co.uk/htm/cmeps3tech.htm>
- [16] Samsung Electronic: 64 CH SEGMENT DRIVER FOR DOT MATRIX LCD. [online], 2000-06-01 [cit. 2010-03-03].  
URL <http://www.datasheetcatalog.org/datasheet/SamsungElectronic/mXyzytqy.pdf>
- [17] Texas Instruments: Low Power Digital Temperature Sensor With Two-Wire Serial Interface in SOT563. [online], 2008-10-01 [cit. 2010-04-03].  
URL <http://focus.ti.com/lit/ds/symlink/tmp102.pdf>
- [18] WINAVR: C compiler for Atmel AVR. WWW Stránky.  
URL <http://winavr.sourceforge.net/>
- [19] Zagros Robotics: Zagros Robotics.  
URL <https://www.zagrosrobotics.com/images/cmeps03a.jpeg>

# Seznam použitých zkratek

$\alpha$ ( $^{\circ}$ )	azimut, stupeň
$C$ ( $F$ )	kapacita, Farad
$f$ ( $Hz$ )	frekvence, Herz
$I$ ( $A$ )	proud, Ampér
$p$ ( $Pa$ )	tlak, Pascal
$R$ ( $\Omega$ )	odpor, Ohm
$t$ ( $s$ )	čas, sekunda
$T$ ( $K, ^{\circ}C$ )	teplota, Kelvin, stupěn Celsia
$U$ ( $V$ )	napetí, Volt
<b>ADC</b>	<i>Analog to Digital</i> Převodník spojitého signálu na diskrétní
<b>ALU</b>	<i>Arithmetic Logic Unit</i> Aritmeticko logická jednotka
<b>AVR</b>	<i>Advanced Virtual RISC</i> Označení pro rodinu 8-mi bitových mikročipů typu RISC
<b>DAC</b>	<i>Digital to Analog</i> Převodník diskrétního signálu na spojitý
<b>DMA</b>	<i>Direct Memory Access</i> Přímí přístup do paměti
<b>DRAM</b>	<i>Dynamic Random Access Memory</i> Dynamická paměť typu RAM
<b>EEPROM</b>	<i>Electrically Erasable Programmable Read-Only Memory</i> Elektricky mazatelná paměť typu ROM
<b>EPROM</b>	<i>Erasable Programmable Read-Only Memory</i> Uv záření mazatelná paměť typu ROM
<b>FLASH</b>	<i>Flash</i> Elektricky programovatelná paměť typu RAM.
<b>GPS</b>	<i>Global Positioning System</i> Polohový družicový systém
<b>I2C (IIC)</b>	<i>Inter-Integrated Circuit</i> Multi-masterová sériová sběrnice
<b>ISP</b>	<i>In System Programming</i> Programování uvnitř obvodu
<b>LCD</b>	<i>Liquid crystal display</i> Displej z tekutých krystalů
<b>RAM</b>	<i>Random Access Memory</i> Paměť u které lze číst i zapisovat

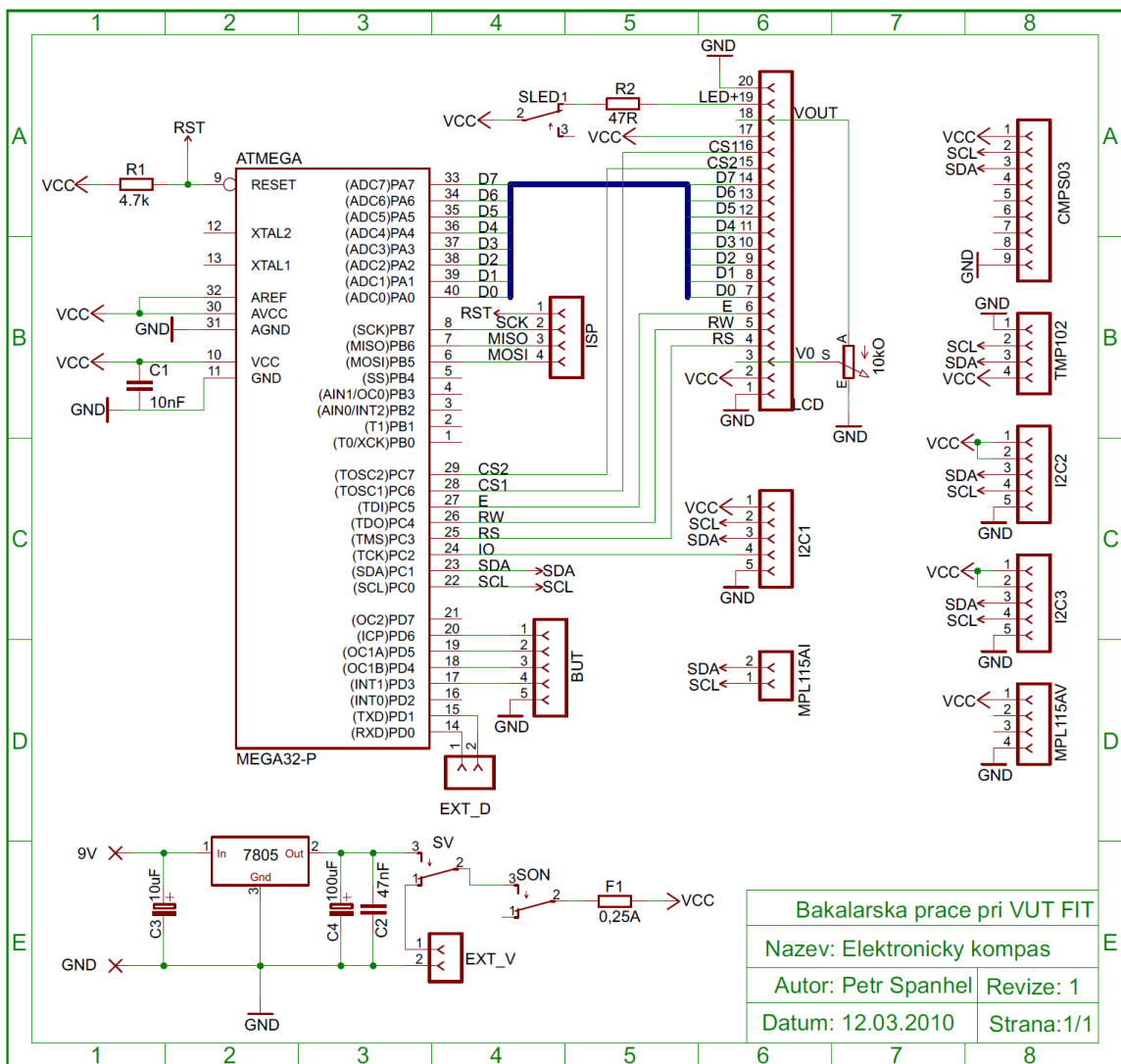
<b>RISC</b>	<i>Reduced Instruction Set Computer</i> Redukovaná instrukční sada
<b>ROM</b>	<i>Read Only Memory</i> Paměť ze které jde pouze číst
<b>SCA</b>	<i>Serial Clock</i> Vodič sběrnice I2C po kterém se posílají data
<b>SI</b>	<i>Systeme International d'Unités</i> Mezinárodně domluvená soustava jednotek
<b>SDA</b>	<i>Serial Data</i> Vodič sběrnice I2C po kterém se vysílá hodinový signál
<b>SRAM</b>	<i>Static Random Access Memory</i> Statická paměť typu RAM
<b>UART</b>	<i>Universal Asynchronous Receiver Transmitter</i> Univerzální asynchronní seriová linka
<b>USART</b>	<i>Universal Synchronous Asynchronous Receiver Transmitter</i> Univerzální synchronní nebo asynchronní seriová linka
<b>USB</b>	<i>Universal Serial Bus</i> Univerzální sériová sběrnice
<i>MASTER</i>	Zahajuje a ukončuje komunikaci
<i>SLAVE</i>	Zařízení odpovídá na dotazy masteru
<i>BUFFER</i>	Vyrovnávací paměť mezi částmi s rozdílnou rychlostí zpracování dat
<i>RECEIVER</i>	Část systému, která přijímá data
<i>TRANSMITTER</i>	Část systému, která vysílá data
<i>PULL-UP</i>	Rezistor zapojený jedním koncem na napájecí napětí a druhým koncem jdoucí k zemi
<i>R/W</i>	Volba mezi čtením nebo zápisem
<i>D/I</i>	Volby mezi daty nebo instrukcí
<i>Logická úroveň</i>	Stav logické „0“ odpovídá nižšímu napětí na výstupu logického členu nežli stav logické „1“.

# Seznam příloh

- A Podklady pro výrobu měřícího zařízení
- B Fotografie výsledného zařízení
- C Ukázka komunikace
- D Dvd disk

# Příloha A

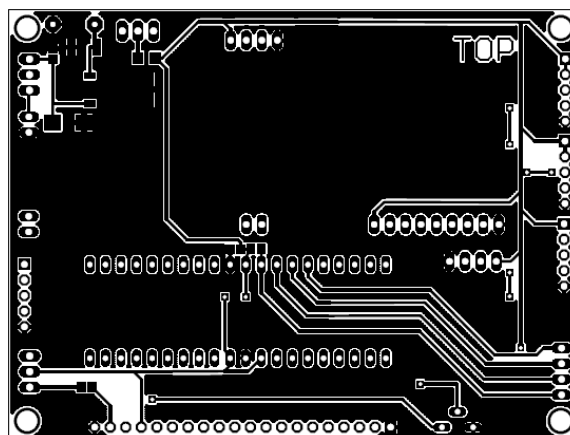
## Podklady pro výrobu měřícího zařízení



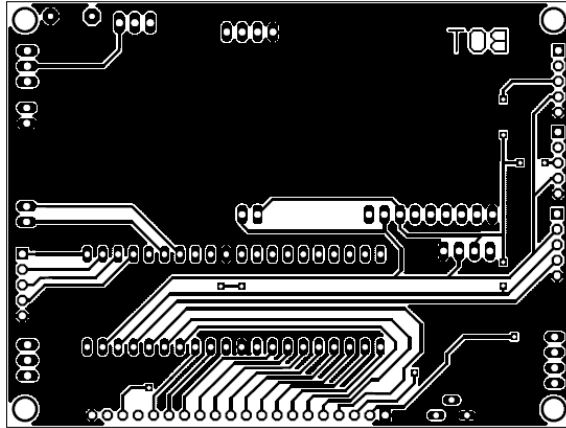
Obrázek A.1: Schéma zapojení

ATMEGA		ATmega16-16PU	DIL40
7805	1 A	Stabilizátor na 5V	D2PAK
MODUL1		Modul kompasu CMPS03	
MODUL2		Kit KITMPL115A2I2C	
MODUL3		TMP102	
C1	10 nF	Keramický kondenzátor	SMD0805
C2	47 nF	Keramický kondenzátor	SMD0805
C3	10 $\mu$ F	Elektrolytický kondenzátor	B
C4	100 $\mu$ F	Elektrolytický kondenzátor	C
R1	4,7 k $\Omega$	Rezistor	R0805
R2	47 $\Omega$	Rezistor	R0805
CON	10 k $\Omega$	Trimr	64Y
F1	0,25 A	Pojistka	R1206
SLED		Miniaturní posuvný spínač	TL3XPO
SON		Miniaturní posuvný spínač	TL3XPO
SV		Miniaturní posuvný spínač	TL3XPO
BUT	5 pinů	Dutinková lišta jednořadá	
CMPS03	9 pinů	Dutinková lišta jednořadá	
LCD	20 pinů	Dutinková lišta jednořadá	
MPL115AI	2 piny	Dutinková lišta jednořadá	
MPL115AV	4 piny	Dutinková lišta jednořadá	
TMP102	4 piny	Dutinková lišta jednořadá	
EXT_D	2 piny	Konektorový kolík jednořadý 90 stupňů	
EXT_V	2 piny	Konektorový kolík jednořadý 90 stupňů	
I2C1	5 pinů	Konektorový kolík jednořadý 90 stupňů	
I2C2	5 pinů	Konektorový kolík jednořadý 90 stupňů	
I2C3	5 pinů	Konektorový kolík jednořadý 90 stupňů	
ISP	4 piny	Konektorový kolík jednořadý 90 stupňů	
DIST	12 mm	Distanční sloupek - matice x šroub o velikosti 12 mm	4x
MATIC	3 mm	Matice pro distanční sloupek velikost 3 mm	4x
SROUB	3 mm	Šroub pro distanční sloupek velikost 3 mm	4x

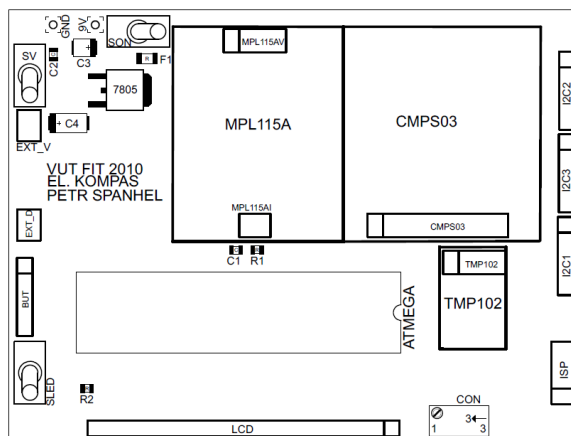
Tabulka A.1: Seznam součástek



Obrázek A.2: Plošný spoj - vrchní strana



Obrázek A.3: Plošný spoj - spodní strana

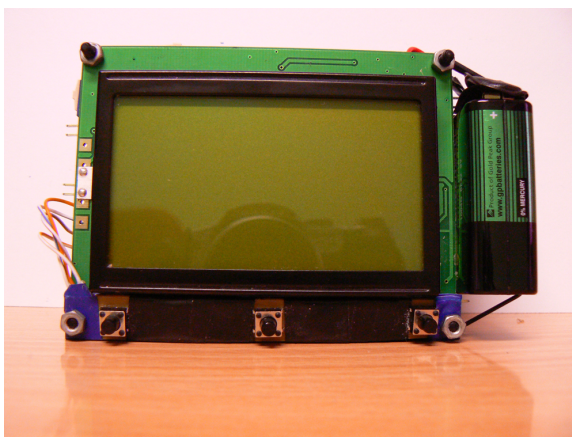


Obrázek A.4: Plošný spoj - Osazení součástek

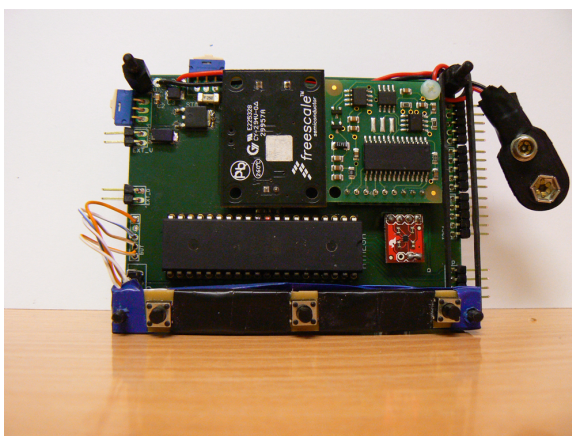


## Příloha B

# Fotografie výsledného zařízení



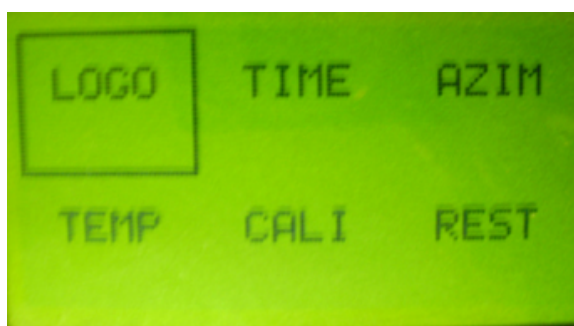
Obrázek B.1: Kompletní zařízení



Obrázek B.2: Spodní vrstva



Obrázek B.3: Hlavní obrazovka



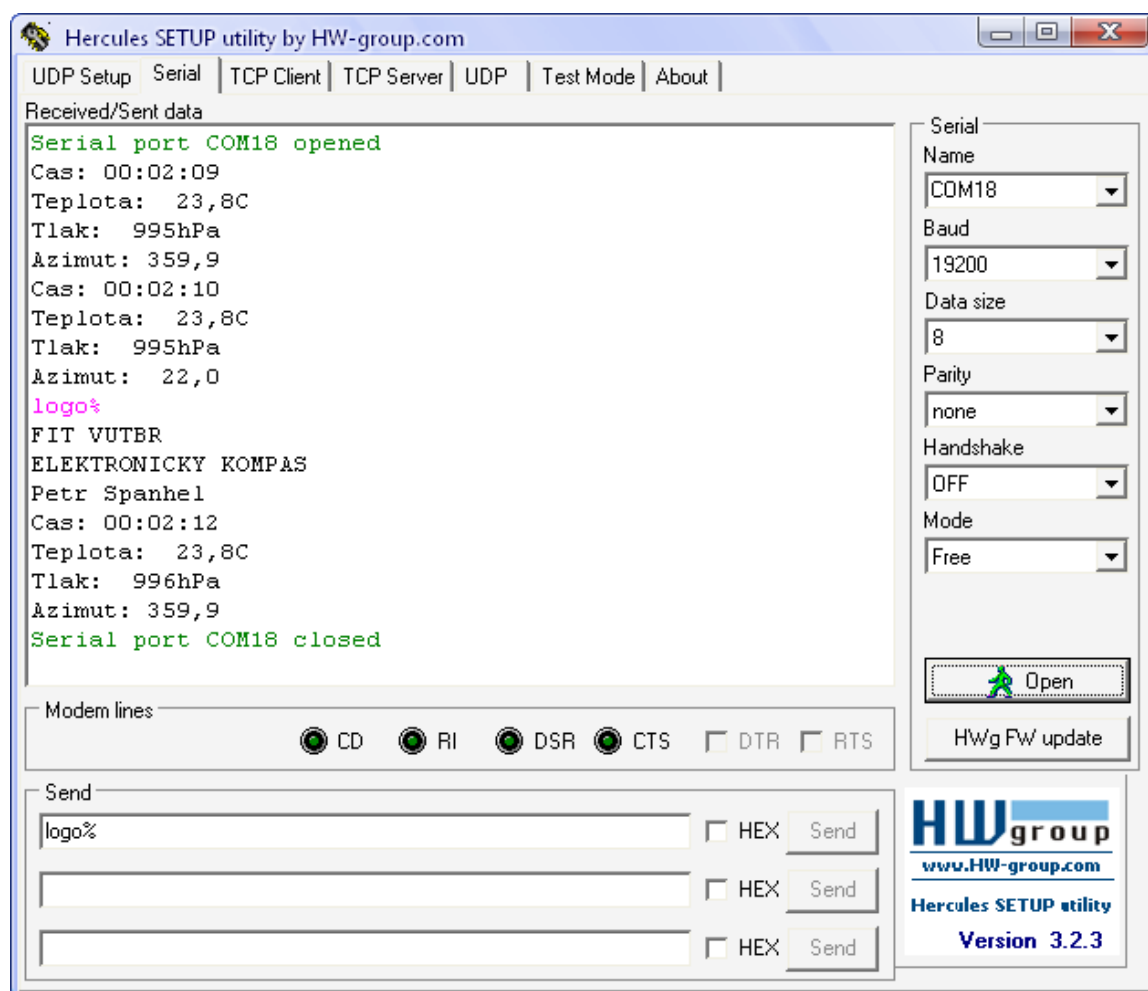
Obrázek B.4: Obrazovka s menu



Obrázek B.5: Logo projektu

## Příloha C

### Ukázka komunikace



Obrázek C.1: Ukázka komunikace s osobním počítačem

# Příloha D

## CD disk

Obsahuje elektronickou verzi písemné zprávy, celý projekt se zdrojovými kódy programu, dokumentaci k softwarové části projektu a podklady pro výrobu desky plošných spojů.