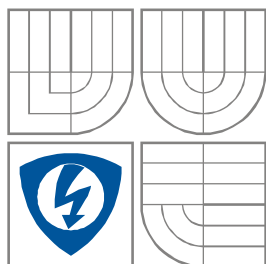


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ  
ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

## VÝŠKOMĚR PRO RC MODELY LETADEL ALTIMETER FOR RC AIRCRAFT MODELS

BAKALÁŘSKÁ PRÁCE

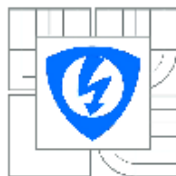
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

Milan Kotulek

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. Jan Prokopec, Ph.D.



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNE

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

## Bakalářská práce

bakalářský studijní obor  
Elektronika a sdělovací technika

**Student:** Milan Kotulek  
**Ročník:** 3

**ID:** 125250  
**Akademický rok:** 2012/2013

### NÁZEV TÉMATU:

**Výškoměr pro RC modely letadel**

### POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti realizace výškoměru pro RC modely letadel. Navrhněte koncepci zařízení pro záznam letové polohy RC modelu. Při návrhu minimalizujte rozměry zařízení a proudový odběr. Připravte software pro zařízení a koncept software pro komunikaci s PC pro stahování údajů o letu.

Realizujte navržené zařízení, vytvořte software pro komunikaci s PC. Ověřte funkci zařízení a sestavte podrobnou dokumentaci.

### DOPORUČENÁ LITERATURA:

[1] BURKHARD, M. C pro mikrokontroléry. Praha: BEN - technická literatura, 2003.

[2] FRÝZA, T., FEDRA, Z., ŠEBESTA, J. Mikroprocesorová technika. Počítačová cvičení. Elektronické skriptum. Brno: FEKT VUT v Brně, 2009.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 31.5.2013

**Vedoucí práce:** Ing. Jan Prokopec, Ph.D.

**Konzultanti bakalářské práce:**

**prof. Dr. Ing. Zbyněk Raida**  
Předseda oborové rady

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Bakalářská práce popisuje zařízení pro měření a záznam letové polohy RC modelu, které se skládá z mikrokontroléru a tlakového čidla. To měří hodnotu atmosférického tlaku a tyto údaje jsou přepočítávány mikrokontrolérem na konkrétní výšku. Vše je během letu ukládáno do paměti a následně může být přeneseno kabelem do PC.

## **KLÍČOVÁ SLOVA**

Atmosférický tlak, výškoměr, Atmega, EEprom, TWI

## **ABSTRACT**

Bachelor thesis describes a device for measuring and recording flight position RC model, which consist of a microcontroller and a pressure sensor. It measures values of atmospheric pressure and these data are convert by microcontroller to specific height. All is during flight saved to memory and later can be transferred to PC.

## **KEYWORDS**

Atmospheric pressure, altimeter, Atmega, EEprom, TWI

KOTULEK, M. *Výškoměr pro RC modely letadel*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 31 s., 8 s. příloh. Vedoucí bakalářské práce Ing. Jan Prokopec, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Výškoměr pro RC modely letadel jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce Ing. Janu Prokopcovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne .....

.....

(podpis autora)

# OBSAH

<b>Seznam obrázků</b>	<b>viii</b>
<b>Seznam tabulek</b>	<b>ix</b>
<b>Úvod</b>	<b>1</b>
<b>1 Výškoměr</b>	<b>2</b>
1.1 Atmosférický tlak .....	2
<b>2 praktické řešení</b>	<b>5</b>
2.1 Komunikace přes TWI.....	5
2.2 Tlakové čidlo .....	6
2.2.1 Kalibrační koeficienty.....	7
2.2.2 Měření teploty čidlem.....	8
2.2.3 Měření tlaku čidlem .....	8
2.2.4 Průběh komunikace.....	9
2.2.5 Čtení výsledků .....	9
2.3 EEPROM .....	11
2.3.1 Komunikace po TWI s pamětí .....	12
2.4 Mikrokontrolér.....	13
2.4.1 Spotřeba .....	14
2.5 Převodník.....	15
2.5.1 UART.....	15
<b>3 Zapojení</b>	<b>18</b>
<b>4 Program</b>	<b>19</b>
4.1 Program pro MCU .....	19
4.1.1 Programová část měření.....	25
<b>5 Výsledky měření</b>	<b>28</b>
<b>6 Závěr</b>	<b>29</b>
<b>Literatura</b>	<b>30</b>

<b>Seznam symbolů, veličin a zkratk</b>	<b>31</b>
<b>Seznam příloh</b>	<b>32</b>

# SEZNAM OBRÁZKŮ

Obr. 1.1:	Průběh změny tlaku s rostoucí nadmořskou výškou [10].	3
Obr. 2.1:	Blokové schéma výškoměru.	5
Obr. 2.2:	Komunikační protokol TWI [3].	6
Obr. 2.3:	Typické zapojení snímače tlaku BMP085 [10].	7
Obr. 2.4:	Prvotní komunikace s čidlem [10].	9
Obr. 2.5:	Čtení výsledků měření [10].	9
Obr. 2.6:	Algoritmus přepočtu na skutečnou teplotu a tlak [10].	10
Obr. 2.7:	Blokové schéma paměti [6].	11
Obr. 2.8:	Komunikace s pamětí při zápisu [6].	12
Obr. 2.9:	Komunikace s pamětí při čtení uložených dat [6].	12
Obr. 2.10:	Pouzdro DIL28 mikrokontroléru ATmega168 s vývody [1].	13
Obr. 2.11:	Zapojení převodníku FT232RL [7].	15
Obr. 2.12:	Zjednodušené znázornění jednotky UART [2].	16
Obr. 2.13:	Struktura rámce pro asynchronní přenos dat v módu 7E2 [5].	17
Obr. 4.1:	Úvod vývojového diagramu.	19
Obr. 4.2:	Vývojový diagram části programu při připojení k PC.	20
Obr. 4.3:	Vývojový diagram části programu pro měření.	24
Obr. 6.1:	Celkové obvodové zapojení výškoměru.	33
Obr. 6.2:	Ukázka programu pro PC.	35
Obr. 6.3:	Závislost proudového odběru v aktivním režimu [1].	36
Obr. 6.4:	Závislost proudového odběru při snížené spotřebě módu Idle [1].	36



# SEZNAM TABULEK

Tab. 1.1:	Závislost barického stupně udávajícího nutné zvýšení výšky v metrech pro pokles tlaku o 1hPa [3].	3
Tab. 2.1:	Kalibrační koeficienty [10].	7
Tab. 2.2:	Hodnoty jimiž se v registru 0xF4 nastuje oblast zájmu měření [10].	8
Tab. 2.3:	Použitelné módy měření [10].	8
Tab. 2.4:	Použitelné módy pro uspání mikrokontroléru [1].	14
Tab. 4.1:	Registr TWCR.	21
Tab. 4.2:	Nastavení rychlosti Baud rate [bit/s] [1].	22
Tab. 4.3:	Registr UCSRnB.	22
Tab. 4.4:	Registr UCSRnC.	23

# ÚVOD

Cílem této práce je realizace výškoměru pro RC modely letadel. Zařízení využívá ke zjištění aktuální výšky měření atmosferického tlaku, který je se změnou výšky pevně spjat. Tlakové čidlo jenž měří tyto změny je ovládáno mikrokontrolérem.

Výsledky jsou během letu ukádány do paměti. Po přistání mohou být k staženy do počítače a dále vytvořeným programem upraveny, archivovány či zobrazeny v grafu.

Při návrhu bylo dbáno na hledisko co nejmenších rozměrů pro snadné zabudování do RC modelu. Software mikrokontroléru obsahuje části pro minimalizaci proudového odběru a tím prodloužení doby po kterou je zařízení schopno záznamu údajů o letu.

# 1 VÝŠKOMĚŘ

Na výpočet výšky v které se pohybuje letadlo se používá několik typů výškoměrů. Nejběžnějším a v letecké dopravě využívaným je tlakový (barometrický). Lze však použít i laser, radiolokátor, GPS(Global Positioning System), atd.

Zařízení využívá tlakového snímače kvůli jeho jednoduchosti a nižším nárokům na napájení. U ostatních systémů by se vyskytli nejrůznější problémy při jejich implementaci. Ať už problém s měřením v zalesněném terénu (laser) nebo větší proudový odběr u GPS.

## 1.1 Atmosférický tlak

Atmosféra Země je vzdušný obal mající svou hmotnost. Na základě normální hustoty má  $1 \text{ m}^3$  vzduchu hmotnost asi 1,29 kg. Hmotnost zemské atmosféry je přibližně 5,157 tisíců tun. Tak velká hmotnost vzduchu působí svou tíhovou silou kolmo na libovolně orientovanou plochu na Zemi a způsobuje tlak. Tento tlak označujeme jako atmosférický (barometrický) a vzniká tíhou svislého sloupce vzduchu sahajícího od zemského povrchu až po horní vrstvy atmosféry v několika stech kilometrech. Jednotkou tlaku je pascal (Pa) v meteorologii se nejčastěji používá jeho násobek hektopascal (hPa = 100 Pa) [4]. Pro výpočet tlaku ve výšce  $h$  lze využít následující rovnici [12]

$$p = p_0 \times \exp\left(\frac{g_0 \times M \times (h_1 - h_2)}{R \times T_b}\right), \quad (1.1)$$

kde  $p$  je tlak ve výšce  $\Delta h$  [Pa],  $p_0$  tlak referenční [Pa],  $g$  je tíhové zrychlení = 9.8 [m/s<sup>2</sup>],  $M$  molární hmotnost vzduchu v atmosféře = 0.0289644 [kg/mol],  $h_2$  je výška v místě měření referenčního tlaku [m],  $h_1$  je výška od hladiny referenčního tlaku [m],  $R$  molární plynová konstanta = 8.31432 [N·m/(mol·K)],  $T$  je teplota [K]

Tlakový výškoměr je barometr měřící jakou silou působí sloupec vzduchu nad ním. Naměřený tlak se převádí na základě matematického modelu standardní atmosféry jenž popisuje barometrická rovnice a výsledkem je výška v metrech. Atmosférický tlak se výrazně mění s počasím v daném místě a době. Jeho závislost na počasí se odstraňuje tím, že se na barometru nastavuje referenční tlak. Na základě zadání rozdílných referenčních hodnot tlaku, se dostane rozdílných hodnot výšky [13].

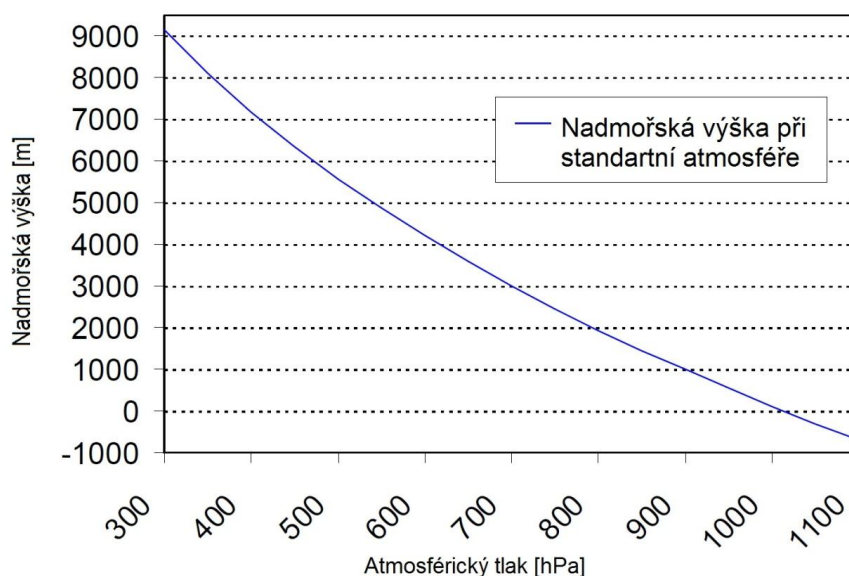
- První způsob využívá jako referenční hodnotu tlak v místě startu a výškoměr tak zobrazí výšku nad letištěm.
- Dalším způsobem je znalost přesné hodnoty nadmořské výšky v místě startu, kterou pak zobrazuje i výškoměr v době startu.
- Ve skutečném leteckém provozu, kde letadlo prolétá místa s rozdílným atmosférickým tlakem se jako referenční hodnota nastavuje 1013,25 hPa (hodnota tlaku na hladině moře udávaná Standardním modelem atmosféry). Údaj prezentovaný výškoměrem pak neudává skutečnou výšku v které se stroj nachází, ale dovoluje zachovat vertikální rozestupy mezi jednotlivými

letadly jenž se pohybují v tzv. letových hladinách.

Standartní model atmosféry je zjednodušením atmosféry a říká jak se mění hustota, viskozita, tlak a teplota s rostoucí výškou. Přesnější vyjádření udává Barický stupeň, který se mění s rostoucí výškou (klesajícím tlakem) a s tím spojeným poklesem teploty. Ta se s rostoucí výškou mění o 0,65°C každých 100 metrů. Z tabulky 1.1 je patrné, že při změně tlaku o 1 hPa se posuneme o 8,6 m výšky což odpovídá změně 11,63 Pa/m. Průběh změny atmosférického tlaku s rostoucí výškou ukazuje obrázek 1.1.

Úpravou rovnice (1.1) lze získat rovnici, díky níž se při znalosti tlaku v daném místě vypočítá výška nad místem s referenčním tlakem  $p_0$ , jak ukazuje následující vztah [10]

$$v\acute{y}\text{ška} = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5,255}}\right) \quad (1.2)$$



Obr. 1.1: Průběh změny tlaku s rostoucí nadmořskou výškou [10].

Tab. 1.1: Závislost barického stupně udávajícího nutné zvýšení výšky v metrech pro pokles tlaku o 1hPa [3].

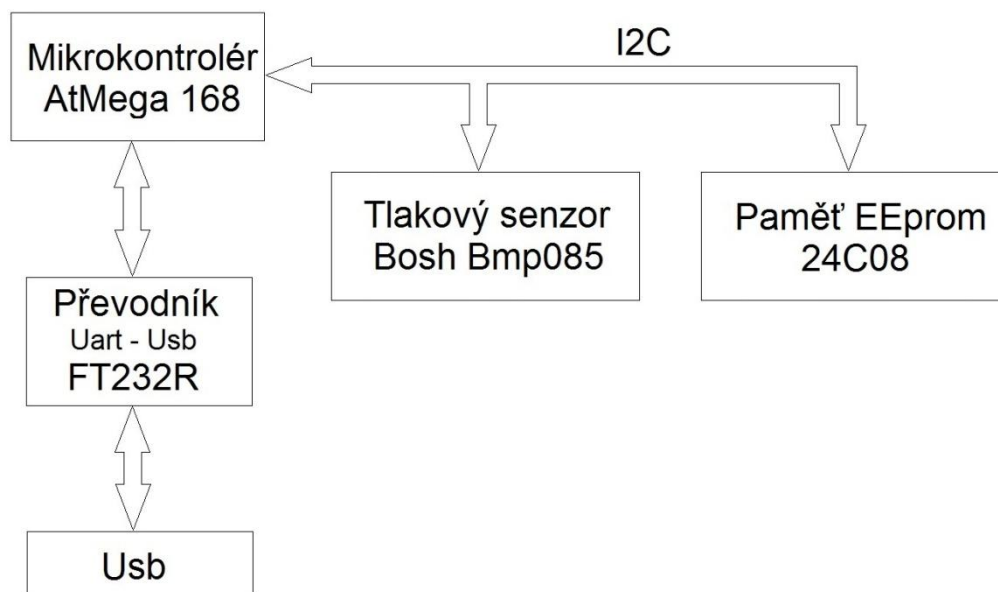
Barický stupeň = dz/dp [m/hPa]					
Tlak [hPa]	Teplota [°C]				
	-40 °C	-20 °C	-0 °C	+20 °C	+40 °C
1000	6,8	7,4	8	8,6	9,2
800	8,5	9,3	9,8	10,7	11,4
600	11,4	12,3	13,3	14,3	15,2
400	17,1	18,6	20	21,4	22,9
100	67,2	73,2	80	86,4	92,8

Ať již zařízení využije kteroukoli ze zmíněných tří variant pro zobrazení výšky, vždy tím ovlivní jen hodnotu na startu. Výška nad zemí se principiálně měří vždy stejně. Na začátku letu zařízení změří tlak v místě startu. Tuto hodnotu bude dále brát jako referenční hodnotu tlaku  $p_0$ . Při letu se budou cyklicky měřit hodnoty tlaku. Ty se porovnájí s referenčním a dostane se tak rozdíl v pascálech.

Změna tlaku o jeden metr není pevně daná hodnota, ale závisí na tlaku v daném místě jak ukazuje tabulka 1.1. Střední nadmořská výška České republiky je 450 m n. m. a maximální hodnota tlaku zaznamenaná na našem území je 1055,4 hPa. Minimem je hodnota 970,1 hPa [4]. Díky těmto hodnotám je zřejmé, že změna tlaku o jeden metr kolísá v rozmezí 11,63 Pa/m (pro 1000 hPa) až 10,3 Pa/m (pro 900 hPa). Na základě měření referenčního tlaku je nejlepší vybrat konkrétní hodnotu. Dělením tímto číslem se dosáhne výsledné výšky nad místem vzletu v metrech. Tento přepočet bude platit do několika set metrů nad zemí kde už by se projevila změna teploty vzduchu a kde by se zároveň již výrazněji změnil okolní tlak.

## 2 PRAKTICKÉ ŘEŠENÍ

Zařízení je reprezentováno blokovým schématem na obrázku 2.1. Samotné měření zajišťuje tlakové čidlo jenž ovládá mikrokontrolér. Jejich vzájemná komunikace je zajištěna sériovou sběrnicí TWI. Její rychlost je omezena nejpomalejším členem, v tomto případě paměti, rychlost komunikace tak může dosáhnout maxima 400 kHz.



Obr. 2.1: Blokové schéma výškoměru.

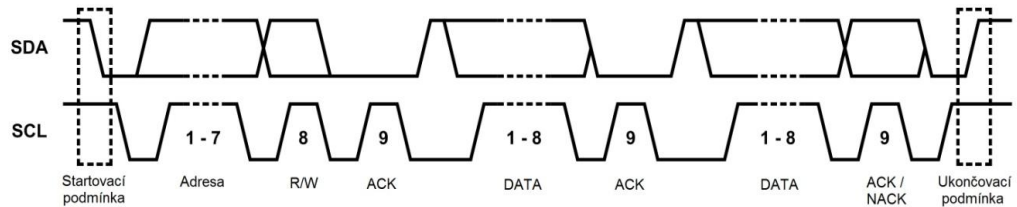
### 2.1 Komunikace přes TWI

Pro komunikaci s okolím využívá čidlo sběrnice TWI. Sběrnice obsahuje dva vodiče. SDA se používá pro přenos užitečných dat a vodič SCL pro hodinový signál. Oba vodiče se připojují k napájecímu napětí pull-up rezistory, typicky 4,7 k $\Omega$ , které zajistí vysohou úroveň signálu v klidovém stavu. Zařízení připojená na sběrnici mají každé svou jedinečnou adresu. Počet zařízení, která lze připojit je omezen pouze kapacitou sběrnice, typicky 400 pF.

Komunikační protokol TWI má přesně specifikované podmínky jak mají vypadat signály pro start, stop i pro přenos binárních dat. Jeho průběh je ukázán na obrázku 2.2. Startovací podmínka vypadá tak, že SCL je ve vysoké úrovni a signál SDA přejde z vysoké na nízkou úroveň neboli jde o sestupnou hranu. Ukončovací podmínka se pozná nástupnou hranou signálu SDA při SCL ve vysoké hodnotě. Tyto podmínky jsou jediné, které se mění při aktivním hodinovém pulsu, kdy je SCL ve vysoké úrovni. Datové signály musí být v této době stabilní.

Komunikační protokol definuje dva typy rámců: adresní a datové. Po startovací podmínce se nejdříve posílá adresa slave zařízení (7 bit), a bit R/W pro výběr zda-li se má číst a nebo zapisovat. Jakmile zařízení rozpozná, že se jedná o jeho adresu potvrdí

tuto skutečnost stažením signálu SDA na nízkou úroveň, neboli pošle potvrzovací bit ACK. Tím byl vyslán adresní rámeček a mohou se vysílat data. Datový rámeček je tvořen osmi bity a jeho přijetí je taktéž potvrzováno bitem ACK [5].



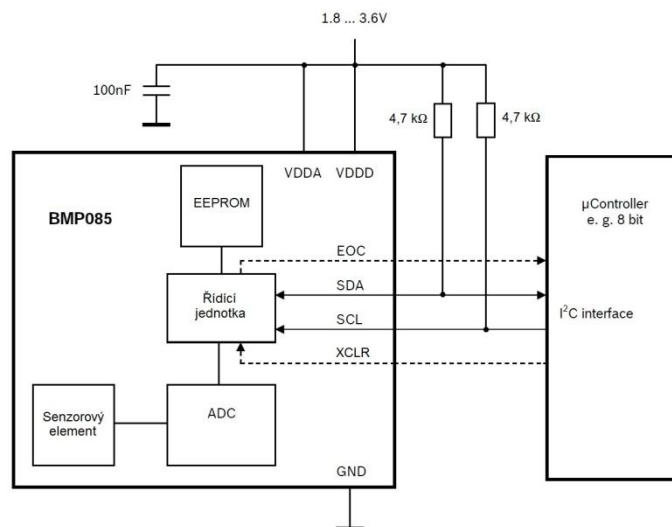
Obr. 2.2: Komunikační protokol TWI [3].

## 2.2 Tlakové čidlo

Popisované zařízení pro měření výšky využívá změny tlaku. Pro měření bylo vybráno čidlo Bmp085 jenž je schopné měřit hodnoty tlaku v rozmezí 300 až 1100 hPa. V absolutních hodnotách je tlak měřen s přesností na 2,5 hPa. Důležitější je zde však rozlišovací schopnost, která je udávána na 1 Pa. Naměřený tlak obsahuje šum, který je v nejlepším případě udáván výrobcem na 0,03 hPa = 0,25 m.

Tento senzor je založen na piezo-rezistivní MEMS technologii ( miniaturní elektro-mechanický systém) integrující mechanické elementy, senzory, akční členy a elektroniku na křemíkovém substrátu [9]. Napájecí napětí je 2,5 V až 3,6 V. Čidlo je schopno pracovat celkem ve čtyřech režimech přesnosti. Každý z nich má jiný proudový odběr, rychlost konverze výsledku a přesnost měření. Kromě měření tlaku je čidlo schopno i měření teploty. Průměrná proudová spotřeba je dle výrobce v nejpřesnějším režimu 12  $\mu$ A. Při konverzi výsledné hodnoty dosahuje odběr špičky 650  $\mu$ A. V klidovém stavu při čekání na měření je udávaná hodnota spotřeby 0,1  $\mu$ A.

Čidlo má digitální výstup a komunikuje výhradně po sběrnici TWI. Jeho adresa při zápisu je 0xEE a při čtení 0xEF. Dokáže po této sběrnici komunikovat rychlostí až 3,4 Mbit/sec při fast a high-speed módu. Jak je vidět na obrázku 2.3 obsahuje i vstup XCLR, kterým lze provést reset obvodu, čítače a inicializaci interních registrů pulsem trvajícím minimálně 1  $\mu$ s. Tento vstup se při nevyužití nemusí připojovat a má vlastní pull-up rezistor. Dalším pinem čidla je výstup EOC díky němuž může čidlo informovat o ukončení měření. Informace se projeví logickou jedničkou v případě ukončení a logickou nulou je-li proces stále v běhu. Doba měření je proměnná v závislosti na zvoleném režimu.



Obr. 2.3: Typické zapojení snímače tlaku BMP085 [10].

## 2.2.1 Kalibrační koeficienty

Aby čidlo měřilo tlak s udávanou přesností je výrobcem kalibrováno. Celkem jedenáct kalibračních koeficientů je uloženo do paměti čidla. Každý z nich má velikost 16 bitů a je tedy rozdělen na dva bajty a uložen na dvou po sobě následujících adresách, přičemž první z nich je na adrese 0xAA. Při výpočtu hodnoty tlaku z bezrozměrné hodnoty, kterou měří čidlo, se tyto kalibrační koeficienty zahrnou.

Pro jejich zjištění se musí navázat komunikace s čidlem, kdy se postupně vyšle adresa čidla s bitem pro nastavení zápisu. Následuje adresa registru v němž jsou kalibrační koeficienty uloženy a po znovuposlání start podmínky tentokrát s bitem pro čtení čidlo pošle obsah paměti na dané adrese. Pokud mikrokontrolér odpoví po jejím přijetí potvrzení ACK, pak čidlo posílá obsah sousední buňky paměti. Obsah paměti a tedy kalibrační koeficienty jsou vypsány v tabulce 2.1 níže.

Tab. 2.1: Kalibrační koeficienty [10].

Načtené kalibrační koeficienty z paměti EEPROM čidla BMP085					
AC1	0xAA	0xAB	16 bit	AC1	8684
AC2	0xAC	0xAD	16 bit	AC2	-1265
AC3	0xAE	0xAF	16 bit	AC3	-14355
AC4	0xB0	0xB1	16 bit	AC4	33691
AC5	0xB2	0xB3	16 bit	AC5	24697
AC6	0xB4	0xB5	16 bit	AC6	23798
B1	0xB6	0xB7	16 bit	B1	5498
B2	0xB8	0xB9	16 bit	B2	69
MB	0xBA	0xBB	16 bit	MB	32768
MC	0xBC	0xBD	16 bit	MC	54481
MD	0xBE	0xBF	16 bit	MD	2432



## 2.2.2 Měření teploty čidlem

Při měření posílá MCU příkazy čidlu, které vrací výsledky jenž se kalibračními koeficienty přepočítají a výsledkem je hodnota tlaku v Pascalech a teploty v desetinách stupně Celsia.

Jako první je potřeba změřit hodnotu okolní teploty jenž má na měření tlaku vliv. Pošle se startovací podmínka, adresa čidla s příznakem zápisu a adresa registru 0xF4. Zápisem do tohoto registru se čidlu říká co se měří. Hodnoty které se zapisují jsou popsány v tabulce 2.2. Posláním hodnoty 0x2E se nastaví měření teploty a ukončí se komunikace. Pokud by byl vyžíván pin EOC mohlo by se na základě jeho stavu rozhodovat kdy je možné číst výsledek měření. Bez využití tohoto pinu se počká po výrobcem udávanou maximální dobu nutnou k jednotlivým měřením. Po jejím uplynutí se posílá startovací podmínka a adresa zařízení. Nyní však následuje adresa registru 0xF6 v němž je uložen horní bajt výsledku. Po opětovném poslání startovací podmínky s adresou zařízení a příznakem pro čtení čidlo pošle výsledek, který má celkem 16 (19) bitů.

Tab. 2.2: Hodnoty jimiž se v registru 0xF4 nastuje oblast zájmu měření [10].

Měří se	Hodnota uložená do registru ( adresa registru 0xF4 )	Maximální doba konverze [ms]
Teplota	0x02	4,5
Tlak (osrs = 0 )	0x34	4,5
Tlak (osrs = 1 )	0x74	7,5
Tlak (osrs = 2 )	0xB4	13,5
Tlak (osrs = 3 )	0xF4	25,5

## 2.2.3 Měření tlaku čidlem

Měření tlaku lze provádět ve čtyřech režimech. Jak ukazuje tabulka 2.3 jejich rozdíl je v proudovém odběru, době potřebné k dosažení výsledku a šumu v něm obsaženém jenž je odstraňován měřením více vzorků tlaku.

Tab. 2.3: Použitelné módy měření [10].

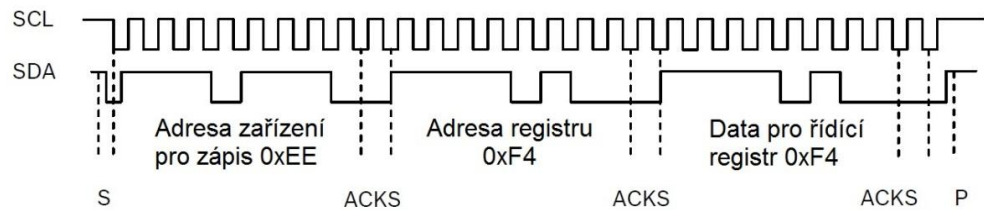
Mód	Parametr osrs	Vnitřní počet vzorků	Průměrný proud při 1 vzorku/s typ. [μA]	RMS hodnota rušení typ. [hPa]	RMS hodnota rušení typ. [hPa]
Nízká spotřeba	0	1	3	0,06	0,5
Standartní	1	2	5	0,05	0,4
Vysoké rozlišení	2	4	7	0,04	0,3
Extra vysoké rozlišení	3	8	12	0,03	0,25

Při výběru režimu měření by bylo nejlepší znát konkrétní model letadla a na základě vyzkoušení vybrat nutné minimální rozlišení. S touto volbou úzce souvisí volba rychlosti opakování jednotlivých měření. Ani toto rozhodnutí není ideální pro každý stroj. Jsou totiž modely akrobatické, civilní či kluzáky a každý z těchto typů dosahuje různých rychlostí. Nelze tak dopředu říci jaký interval mezi měřeními je nejvhodnější pro získání aspoň trochu spojitě křivky výškového záznamu letu.

Sestavené zařízení používá nejpřesnější možný režim a interval mezi měřeními je jedna sekunda.

## 2.2.4 Průběh komunikace

Časový průběh komunikace s čidlem po staru měření tlaku je shodný s měřením teploty až na údaj zapsaný do registru 0xF4 jak ukazuje tabulka 2.2. Celý průběh je ukázán na obrázku 2.4. Po startovací podmínce pošle mikrokontrolér adresu zařízení a adresu registru 0xF4. Následuje hodnota 0xF4 nastavující následující měření tlaku v nejpřesnějším režimu. Čidlo BMP085 posílá potvrzovací bit ACKS po každých 8 bitech, přijmulo-li je. Mikrokontrolér ukončí komunikaci po přijetí posledního potvrzovacího bitu čidla.

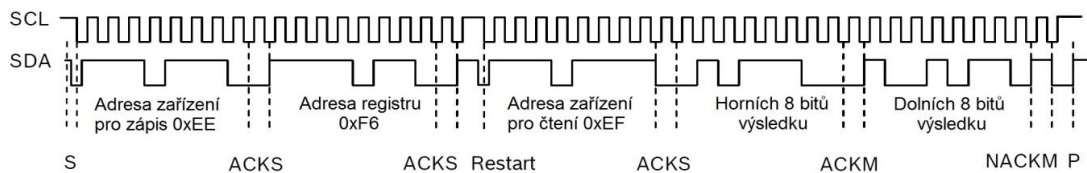


Obr. 2.4: Prvotní komunikace s čidlem [10].

## 2.2.5 Čtení výsledků

Výsledná data teploty jsou v bitovém slově o délce 16 bitů, tlaková informace je taktéž šestnácti bitová a při použití nejpřesnějšího módu má 19bitů.

Po zahájení komunikace MCU posílá adresu zařízení a adresu registru 0xF6, který obsahuje horních 8 bitů výsledku měření. Následuje restart a adresa čidla tentokrát však s příznakem čtení. Čidlo poté pošle prvních 8 nejvýznamnějších bitů, což tentokrát potvrzuje MCU. Následně posílá čidlo 8 nižších bitů a protože je nastaven nejpřesnější režim čidlo posílá dalších 8 bitů. Užitečnými z posledního bajtu jsou však pouze 3 nejvýznamnější bity. Nyní lze komunikaci ukončit.



Obr. 2.5: Čtení výsledků měření [10].

Získané výsledky jsou bezrozměrná čísla. Je třeba je přepočítat na základě znalosti teploty v době měření a kalibračních koeficientů. Konkrétní výpočet jakým se toho dosahuje je ukázán v katalogovém listu následujícím obrázkem.

Výpočet skutečné teploty
$X1 = (UT - AC6) * AC5 / 2^{15}$
$X2 = MC * 2^{11} / (X1 + MD)$
$B5 = X1 + X2$
$T = (B5 + 8) / 2^4$

Výpočet skutečného tlaku
$B6 = B5 - 4000$
$X1 = (B2 * (B6 * B6 / 2^{12})) / 2^{11}$
$X2 = AC2 * B6 / 2^{11}$
$X3 = X1 + X2$
$B3 = ((AC1 * 4 + X3) << oss + 2) / 4$
$X1 = AC3 * B6 / 2^{13}$
$X2 = (B1 * (B6 * B6 / 2^{12})) / 2^{16}$
$X3 = ((X1 + X2) + 2) / 2^2$
$B4 = AC4 * (\text{unsigned long})(X3 + 32768) / 2^{15}$
$B7 = ((\text{unsigned long})UP - B3) * (50000 >> oss$
$\text{if } (B7 < 0x80000000) \{ p = (B7 * 2) / B4 \}$
$\text{else } \{ p = (B7 / B4) * 2 \}$
$X1 = (p / 2^8) * (p / 2^8)$
$X1 = (X1 * 3038) / 2^{16}$
$X2 = (-7357 * p) / 2^{16}$
$p = p + (X1 + X2 + 3791) / 2^4$

Obr. 2.6: Algoritmus přepočtu na skutečnou teplotu a tlak [10].

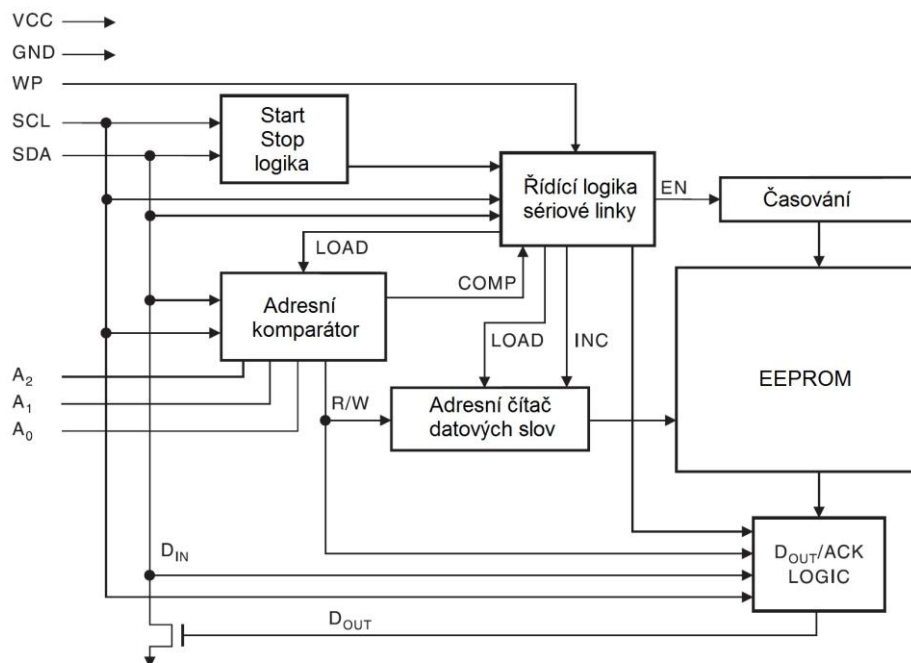
## 2.3 EEPROM

Změřená data je třeba uložit do doby než model dosedne zpět na zem a je možno je stáhnout do PC. K tomuto účelu je zde paměť EEPROM. Hodnoty naměřené čidlem jsou stejně jako tlak z nich vypočteny uloženy v proměnných o velikosti čtyř bajtů (32 bitů). Měří-li se tedy s periodou jedné vteřiny a ukládány jsou jen výsledky, pak pro minutu letu je potřeba alespoň 1920 bitů. Byla vybrána paměť od firmy Microchip, která vyrábí paměti EEPROM komunikující po sběrnici TWI ve velikostech od 128 bitů po 1 Mbit. Při velikosti paměti 1 Mbit (220 bitů) lze zaznamenat až 546 minut záznamu.

Samotná paměť je k desce plošného spoje připojen přes patičku a díky jednotnému pouzdru ji lze kdykoliv vyměnit za větší či menší dle potřeby. Její piny včetně vnitřní blokové struktury jsou vidět na obrázku 2.7. Kromě pinů pro TWI má ještě ochranu proti zápisu (WP), která se dá nastavit vysokou úrovní. Dále piny označené jako A0, A1, A2, které jsou součástí adresy a jejich účel se projeví při více pamětech zapojených na jedné sběrnici.

Na desce osazená paměť 24C08 má následující parametry:

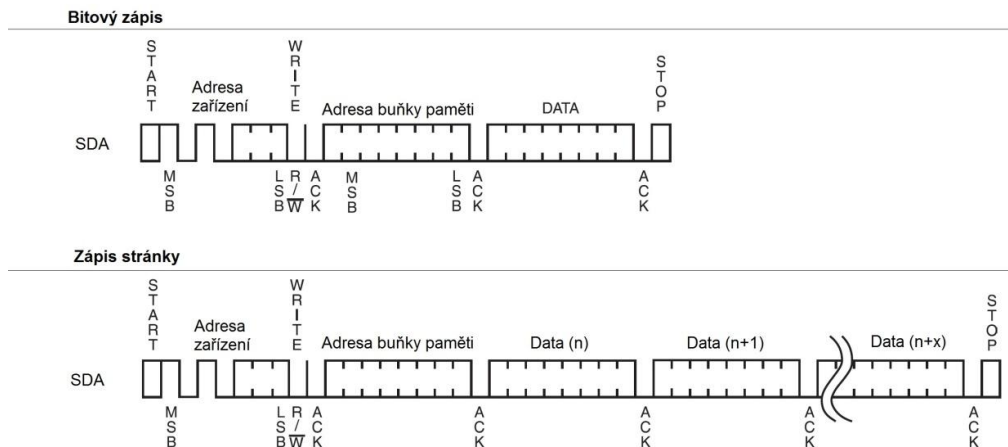
- Velikost paměti: 8 kbit =  $1024 \times 8 = 8192$  bitů
- Vnitřní konfigurace: 16 bytová stránka
- Možnosti komunikace: sériová sběrnice TWI
- Maximální frekvence: 400 kHz
- Napájecí napětí: 2.5 až 5.5 V
- Typ pouzdra: DIP8
- Počet pinů: 8
- Provozní rozsah teplot:  $-40^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$



Obr. 2.7: Blokové schéma paměti [6].

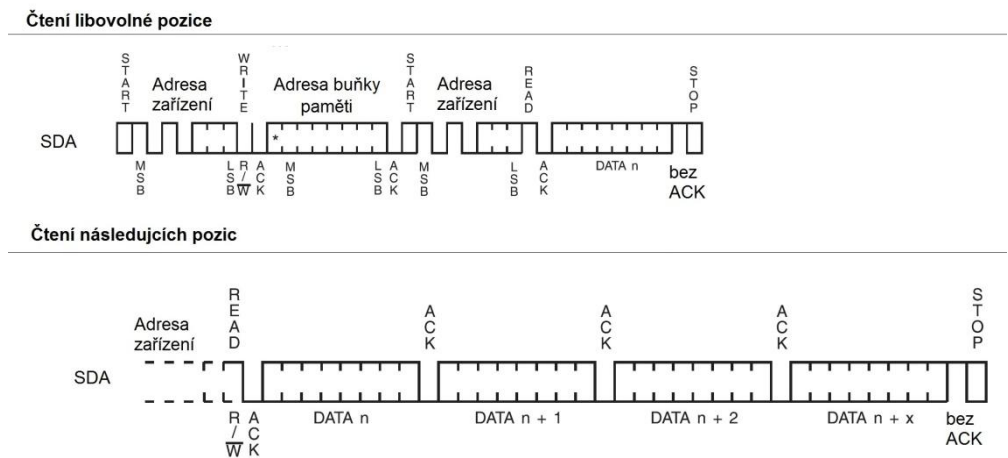
### 2.3.1 Komunikace po TWI s pamětí

Adresa paměti se skládá z pevně daných prvních 4 bitů, 3 bitů nastavených piny A0, A1, A2 na nulu a bitu R/W. Adresa má tedy tvar 0xA0 pro zápis a 0xA1 pro čtení. Časový průběh zápisu je vidět na obrázku 2.8. Po navázání spojení s pamětí se vysílá vyšší a nižší bajt adresy na kterou mají být data ukládána. Následují samotná data. Paměť obsahuje čítač do něž byla uložena adresa buňky pro uložení dat, který se nemaže, ale naopak se automaticky inkrementuje a ukazuje na další buňku. V případě, že MCU komunikaci neukončí po vyslání prvního bajtu je možné poslat další bajt, který bude zapsán na následující adresu a takto postupovat až s 16 bajty.



Obr. 2.8: Komunikace s pamětí při zápisu [6].

Obdobný postup jako pro zápis je použit při čtení uložených dat na obrázku 2.9. Nejdříve MCU posílá adresu paměti a příznak zápisu. Následuje adresa paměťové buňky kterou chce získat. Po restartu komunikace se posílá adresa paměti, tentokrát s nastaveným příznakem čtení. Paměť pošle požadovaná data a MCU nemusí potvrzovat přijetí a komunikaci ukončí. Může však také přijetí potvrdit a paměť posílá obsah následující buňky. Takto lze přečíst obsah celé paměti.



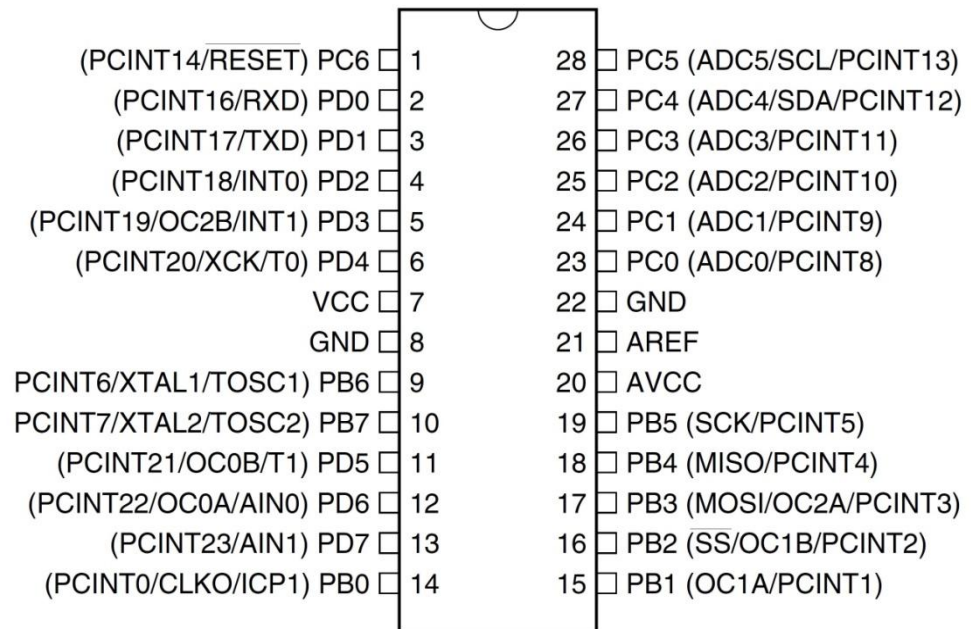
Obr. 2.9: Komunikace s pamětí při čtení uložených dat [6].

## 2.4 Mikrokontrolér

Pro řízení výškoměru byl zvolen mikrokontrolér (MCU) od firmy Atmel. Konkrétní model byl vybrán na základě několika požadavků. Prvním bylo napájecí napětí 3,3 V z důvodu kompatibility s čidlem, jenž je taktéž napájeno tímto napětím. Sběrnice přes kterou poté obě součástky spolu komunikují má tak tedy logické úrovně jimž rozumí. Dalším požadavkem byla schopnost komunikovat přes rozhraní TWI a UART. Posledním kritériem bylo pouzdro DIP, jenž se dá vložit do patice. Tímto výběrem byl jako nejvhodnější mikrokontrolér vybrán ATmega168-20PU.

Jedná se o 8bitový mikrokontrolér a některé z jeho vlastností jsou vypsány níže:

- počet I/O pinů: 28
- Velikost programové paměti: 16 KB
- Velikost paměti EEprom: 512 Bajtů
- Velikost paměti RAM: 1 KB
- CPU rychlost: 0 - 10 MHz
- Typ oscilátoru: Externí, Interní
- Počet časovačů: 3
- Napájecí napětí: 2.7V to 5.5 V
- Provozní rozsah teplot: -40°C to +85 °C
- počet pinů: 32
- Možnosti komunikace: TWI, Serial, SPI, USART, 2-Wire
- Pouzdro DIL28



Obr. 2.10: Pouzdro DIL28 mikrokontroléru ATmega168 s vývody [1].

## 2.4.1 Spotřeba

Proudový odběr mikrokontroléru jde ovlivnit vypnutím jen jeho částí nastavením registru Power reduction (PRR). Ten umožňuje vypnout jednotky TWI, UART, SPI, převodník A/D a čítače/časovače.

Další možností je využít některý z celkem pěti módů na snížení proudové spotřeby mikrokontroléru, které jsou ukázány v tabulce 2.4. Při jejich využití jsou vypnuty hodinové signály pro jednotlivé části MCU nebo je přímo vypnut i zdroj signálu. Poté je tedy je nutno připojit externí krystal. Při jejich aktivaci je MCU schopen pouze čekat na přerušení a to buď interní od časovače či převodníku, nebo externí například rozpoznáním své adresy na sběrnici TWI.

Zařízení využije módu snížené spotřeby pouze ve chvílích mezi jednotlivými měřeními. Přerušením které ho probudí je přetečení časovače. Nejlepší volbou je využití módu s nejméně funkčními částmi. Jelikož však zařízení nemá vnější krystal není možné použít Standby ani Power-save a down mód. Využívá se tedy mód Idle, jenž má dostupný zdroj hodinového signálu a zároveň časovač nutný k probuzení.

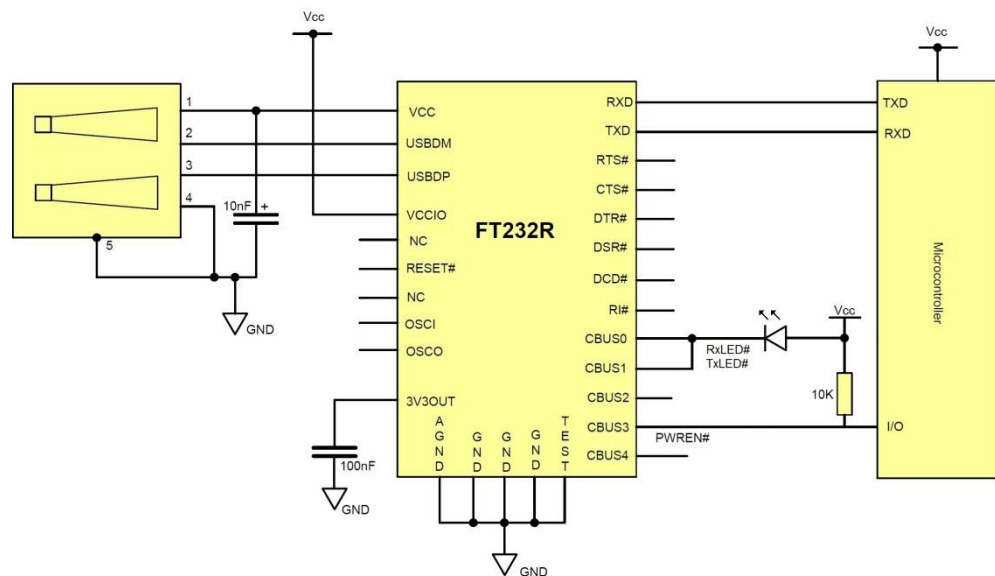
Tab. 2.4: Použitelné módy pro usnutí mikrokontroléru [1].

Mód spánku	Části s aktivním hod. sig.					Oscilátor		Zdroje probuzení					
	clk <sub>CPU</sub>	clk <sub>Flash</sub>	clk <sub>IO</sub>	clk <sub>ADC</sub>	clk <sub>ASY</sub>	Hlavní zdroj hod. signálu	Časovač osc povolen	INT1, INTO změna pinu	Shoda TWI adresy	Časovač2	SPM/EEPRO M	ADC	WDT
Idle			x	x	x	x	x	x	x	x	x	x	x
ADC redukce šumu				x	x	x	x	x	x	x	x	x	x
Power-down								x	x				x
Power-save					x		x	x	x	x			x
Standby						x		x	x				x

Spotřeba MCU je podle katalogového listu výrobce v aktivním stavu při napájení 3,3 V a s interním oscilátorem 8 MHz rovna 3 mA. Při stejných podmínkách, ale v režimu snížené spotřeby Idle klesne odběr na 0,7 mA jak ukazuje obrázek 6.4 v příloze. Tuto hodnotu však ještě mírně navyšuje ponechání v chodu některých částí MCU jako TWI či Uart nastavením registru PRR.

## 2.5 Převodník

Poslední zásadní část výškoměru je převodník zprostředkovávající komunikaci mezi MCU a PC. Vhodnou součástí je FT232RL která, jak je vidět na obrázku 2.12, k propojení na MCU využívá rozhraní UART a veškerá přijatá data pouze přeposílá na rozhraní USB. Převodník umožňuje při komunikaci s MCU využití hardwarového řízení toku piny RTS, CTS, DTR, DSR, DCD a RI. Piny CBUS jsou programovatelné. Bylo však využito jejich továrního nastavení, kdy piny CBUS0 a CBUS1 svou nízkou úroveň rozsvítí k nim připojenou ledku a informují tak o vysílání resp. přijímání dat přes USB. U pinu CBUS3 je využívána funkce PWREN, jež nízkou úroveň informuje o připojení zařízení k PC. V opačném případě je na tomto pinu díky pull-up rezistoru vysoká úroveň.



Obr. 2.11: Zapojení převodníku FT232RL [7].

### 2.5.1 UART

Neboli univerzální asynchronní přijímač a vysílač. Základní úlohou jednotky UART je převzetí datového ( nbitového ) slova od MCU v paralelní podobě a vyslání takového slova po sériové lince. Data se v takovém případě přivedou do registru UDR a jsou v závislosti na zvoleném způsobu komunikace dolněna o start bit, paritními bity a stop bitem stop bity. Následně po přesunu do registru TxDR jsou postupně vysílána.

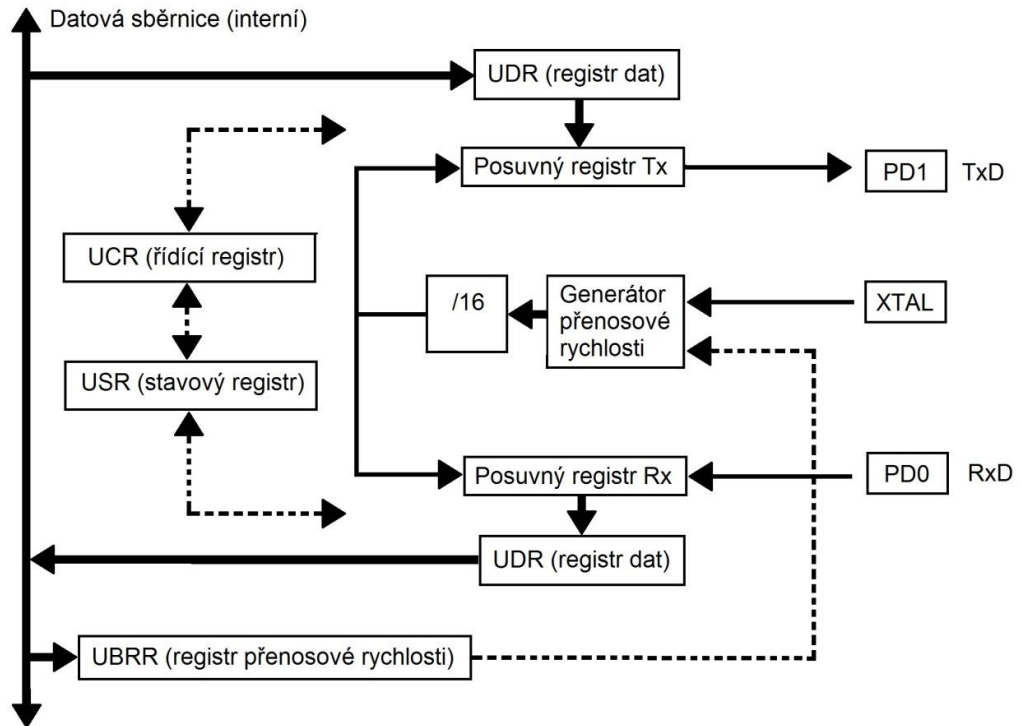
Při příjmu dat je nejprve rozpoznána změna logické úrovně na lince a následně jsou tyto změny načítány do posuvného registru RxDR. Posléze jsou tyto data předána mikrokontroléru.

Protože na přijímač se přivádí jen datová linka a nikoliv bitový hodinový signál, hovoří se o asynchronním přenosu. Vysílání i přijímání je řízeno vnitřním hodinovým signálem, který je typicky 16x vyšší než symbolová rychlost dat na lince. Tímto signálem je taktéž vzorkován stav linky a sestupná hrana start bitu je tak rozpoznána nejhůře s přesností 1/16 doby bitu. Po jejím zachycení se čeká 8 period než se znovu



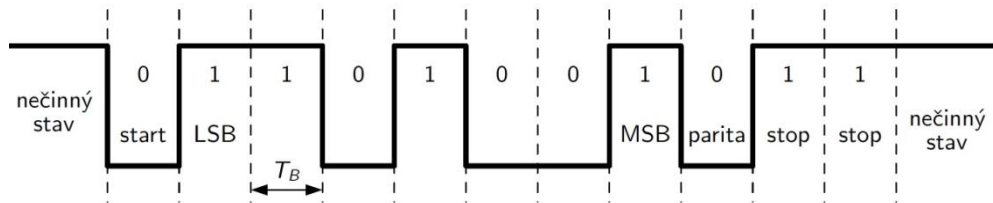
otestuje nešlo-li o falešný impuls. Pokud ne, je stav linky zjišťován každých 16 period hodinového signálu, tedy přesně uprostřed doby trvání bitu. [5]

Jak je vidět na obrázku 2.13 tak UART obsahuje i registr pro nastavení přenosové rychlosti UBRR a několik řídicích a stavových UCSR. Ty jsou nastavovány na základě událostí v přenášené komunikaci. Předávají tak informaci o dokončení vysílání, přijímání dat či že datový registr je již prázdný a může opět přijmout data a vysílat. Dále se lze dozvědět, že přijmutá data jsou v pořádku nebo obsahují chybu rámce nebo parity.



Obr. 2.12: Zjednodušené znázornění jednotky UART [2].

Rámec dat vysílaných při sériovém přenosu jednotkou UART je ukázána na obrázku 2.14. Přenos začíná start bitem. Ten je charakterizován přechodem z klidového stavu HIGH na stav LOW. Samotná data se přenášejí od bitu s nejnižší vahou (LSB - Least Significant Bit) přičemž jejich počet lze nastavit od pěti po osm. Každý bit je na lince udržován po přesně dobu závislou od nastavené symbolové rychlosti. Pro kontrolu zda nedošlo při přenosu informace k chybě se ke každému rámci připojují paritní bity. Paritu lze nastavit na lichou nebo sudou, ale i žádnou nepřipojit. Na konec rámce se připojují stop bity a to buď jeden nebo dva, které mají vždy stav HIGH.



Obr. 2.13: Struktura rámce pro asynchronní přenos dat v módu 7E2 [5].

**Symbolová rychlost:** Udává možný počet změn stavu přenášeného datového signálu za sekundu. Může být označena i jako rychlost modulační. Uvádí se v jednotkách Baud [ Bd ].

**Bitová rychlost:** Udává frekvenci s jakou jsou přenášeny bity za sekundu a uvádí se v jednotkách bit/s nebo bps.

Pokud jedna signálová změna zahrnuje pouze jeden bit, pak se přenosová rychlost v bit/s rovná v Baud/s [5].

### 3 ZAPOJENÍ

Celé zařízení obsahuje tři důležité součástky. O řízení se stará mikrokontrolér Atmega 168. Součástí díky níž je výškoměr použitelný je tlakové čidlo Bmp085 od firmy Bosh. Poslední důležitou částí je převodník FT232R zprostředkovávající naměřené výsledky počítači.

Čidlo je vyráběno v bezvývodovém pouzdře a pro jeho náročnost při zapojení bylo tedy zakoupeno již připojené k desce plošného spoje jenž obsahuje pull-up rezistory pro sběrnici TWI a má vývody všech pinů. Nejjednodušším způsobem jak umožnit komunikaci mezi jednotlivými bloky na sběrnici TWI je, mají-li stejná napájecí napětí. Jelikož maximální napájecí napětí čidla je 3,6V, byla jeho hodnota pro celé zařízení zvolena 3,3V. Tuto hodnotu napětí udržuje stabilizátor LF33CV. Jeho zapojení stejně jako zapojení celého výškoměru je uvedeno na obrázku 6.1 v příloze. Na vstupu je opatřen vypínačem umožňujícím přivést napětí z baterie nebo v případě připojení k PC z rozhraní USB.

Mikrokontrolér ovládá paměť EEprom a čidlo prostřednictvím TWI jenž je vyvedena na 4. a 5. bitu portu C. Na ten jsou taktéž připojeny zbylé dva vodiče čidla umožňující jeho restart a informující o ukončení měření. Na portu B je k mikrokontroléru připojen šestipinový konektor ISP umožňující připojení programátoru. Vzhledem k tomu, že na výstupu programátoru je napětí 5 V neslučující se s omezeními pro čidlo bylo napájení připojeno přes vypínač. Díky němu lze čidlo odpojit a napájet zařízení z programátoru, nebo odpojit programátor a umožnit měření čidlu.

Na portu D je připojena poslední důležitá součástka a to převodník. Ke komunikaci s MCU využívá rozhraní UART obsahující v tomto případě dva vodiče připojené k RxD pro příjem a TxD pro vysílání. Převodník jako jediný není napájen přes stabilizátor, ale přímo z USB pěti volty. Díky tomu musí být na jeho vstup VCCIO přivedeno napětí 3,3V. To mu umožňuje komunikovat ve správných logických úrovních. Mikrokontrolér má připojen i resetovací vstup a to jak k programátoru tak k tlačítku.

Celkem zapojení obsahuje tři Led diody. První indikující probíhající měření, další přenosť přes Usb a poslední je navržena pro upozornění na aktivní reset u převodníku.

Celé zapojení včetně desky plošného spoje bylo navrženo v programu Eagle.

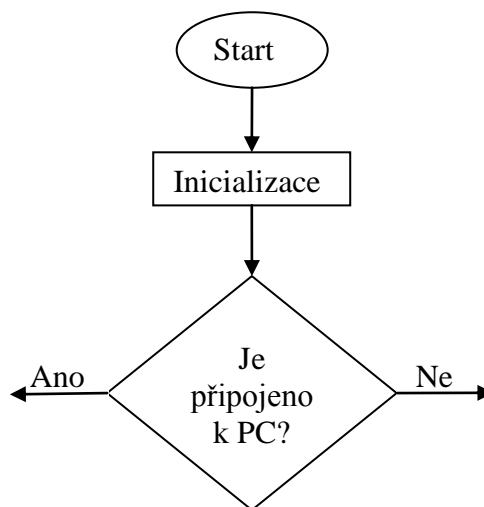
## 4 PROGRAM

Pro obsluhu zařízení byly vytvořeny dva programy. První z nich řídí funkce mikrokontroléru a zajišťuje tak samotné měření a zaznamenání dat o tlaku a teplotě v průběhu letu. Druhý vytvořený program slouží ke zpracování záznamů z letu v počítači.

### 4.1 Program pro MCU

Atmega je dodávána s přednastaveným zdrojem hodinového signálu od interního oscilátoru, jehož frekvence je od výrobce přednastavena na hodnotu 8 MHz  $\pm 10\%$ . Změnou registru OSCCAL lze dosáhnout přesnosti  $\pm 2\%$ . Taktéž je však nastavena programovatelná propojka (tzv. Fuse bit), která nastavuje předděličku oscilátoru na hodnotu 8. Tím snižuje frekvenci hodinového signálu na 1MHz.

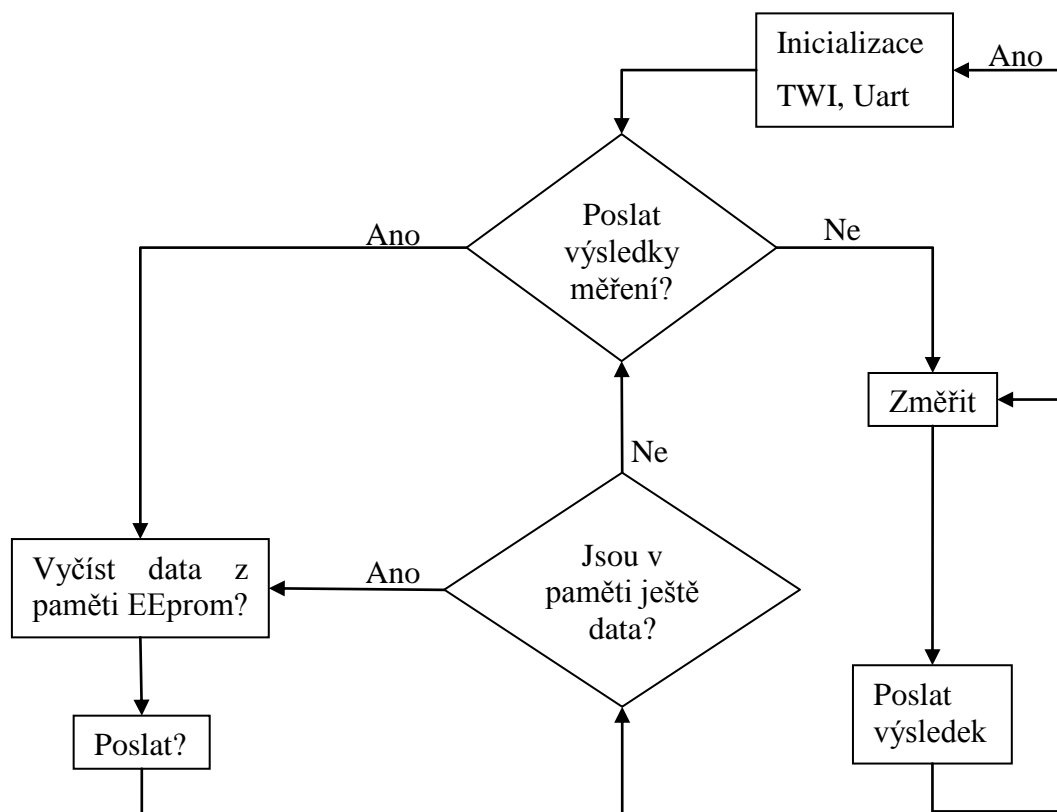
Prvním krokem při vykonávání programu je nastavení předděličky hodinového signálu na hodnotu 1 a tím vrácení frekvence CPU na hodnotu 8MHz, což je potřeba pro nastavení dostatečné (nezáporné) hodnoty do registru TWBR při inicializaci TWI. Toho se dosáhne vepsáním nuly do registru CLKPR.



Obr. 4.1: Úvod vývojového diagramu.

Jako další se nastavují vstupně/výstupní porty. Toho se hned vzápětí využívá, jelikož port D je nastaven jako vstup a na jeho čtvrtý pin je přiveden signál PWREN z převodníku. Ten svou nízkou úrovní informuje, zda je zařízení připojeno k PC, čímž se vykonávání programu rozdělí na dvě větve (viz. obr. 4.1). Výše popsané je ukázáno na následující části kódu funkce main.

V první větvi programu (viz. obr. 4.2) kdy je zařízení připojeno se inicializuje komunikace po sběrnicích TWI a Uart. Následně se vyčká přijetí povelu od PC zda se mohou poslat již změřené hodnoty. Druhou možností je měření úplně nových hodnot, které se budou průběžně posílat v nekonečné smyčce. Tato možnost je zde spíše kvůli testování, kdy je možno okamžitě vidět jaké hodnoty čidlo udává a jak reaguje na konkrétní změny výšky.



Obr. 4.2: Vývojový diagram části programu při připojení k PC.

Při inicializaci TWI je nutné znát frekvenci hodinového signálu. Zařízení využívá pouze interní oscilátor mikrokontroléru a jak již bylo zmíněno jeho hodnota je nastavena na 8 MHz.

Impulsy hodinového signálu jsou přenášeny vodičem SCL. Pro nastavení jejich četnosti na hodnotu 100 kHz se využije vztahu (4.1) respektive jeho úpravou se dostane vztah (4.2). Díky tomu se vypočítá hodnota registru TWBR ( TWI bit rate registr ). Ten má být podle katalogového listu alespoň 10, doporučuje se ještě vyšší. Z tohoto požadavku plyne potřeba frekvence oscilátoru vyšší než 1 MHz a využití tak maximální možné hodnoty. Při jeho zvýšení pouze na 4 MHz se nesplní doporučení aby hodnota TWBR byla alespoň 10, má-li být frekvence SCL 100 kHz. Hodnota předděličky je nastavena na 1, aby ještě více nesnižovala kmitočet SCL.

$$SCL\ frekvence = \frac{CPU\ frekvence\ hodinového\ signálu}{16+2(TWBR) \times (hodnota\ předděličky)} \quad (4.1) [1]$$

$$TWBR = \left( \frac{F_{CPU}}{SCL_{CLOCK}} - 16 \right) \div 2 = \left( \frac{8 \times 10^6}{100 \times 10^3} - 16 \right) \div 2 = 32 \quad (4.2)$$

Při samotné komunikaci jednotky se prakticky využívají pouze dva registry. Prvním je datový TWDR jenž obsahuje data která jsou přijímána/vysílána ( tím je myšlena i adresa volané součástky ) a druhým je registr řídicí TWCR.

Tab. 4.1: Registr TWCR.

TWI (I2C) řídicí registr - TWCR									
Bit	7	6	5	4	3	2	1	0	
	TWCR								
Čtení/zápis	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Počáteční hodnota	0	0	0	0	0	0	0	0	

Bit 7 - TWINT - Po dokončení operace je nastaven, čímž se zjistí úspěšné provedení operace.

Bit 6 - TWEA - Využije se při příjmu dat, kdy jeho nastavením umožníme vyslání potvrzovacího bitu ACK na linku.

Bit 5 - TWSTA - Po nastavení vysílá starovací podmínku.

Bit 4 - TWSTO - Po nastavení vyšle ukončovací podmínku.

Bit 2 - TWEN - Aktivuje TWI. Po jeho nastavení převezme TWI kontrolu nad piny SCL a SDA.

Zde je ukázána funkce pro posílání dat využívaná mikrokontrolérem řídicím výškoměř:

```
void twi_write( char data ) {
    TWDR = data ;          // načtení datového paketu do datového registru
    TWCR = (1<<TWINT) | (1<<TWEN) ;    // vyslání datového paketu
                                        // TWINT...TWI Interrupt Flag
                                        // TWEN....TWI Enable Bit
    while( !( TWCR & (1<<TWINT)) ) ;
        // čekání na dokončení operace, tj. TWINT=1
}
```

Při inicializaci jednotky UART se taktéž využije několik registrů. Pro nastavení přenosové rychlosti (baud rate = bit/s) je určen UBRRn. Následující tabulka ukazuje jak se vypočítá obsah registru UBRR. Nastavení bitu U2Xn v registru UCSRnA se používá pro zdvojnásobení přenosové rychlosti.

Tab. 4.2: Nastavení rychlosti Baud rate [bit/s] [1].

Komunikační mód	Rovnice pro výpočet přenosové rychlosti	Rovnice pro výpočet hodnoty registru UBRRn
Asynchroní, Normální (U2Xn = 0)	$BAUD = \frac{f_{osc}}{16 \times (UBRRn + 1)}$	$UBRRn = \frac{f_{osc}}{16 \times BAUD} - 1$

Ukázka inicializace jednotky Uart:

```
uart_init(207); //9600 bit/s = 51 = 0011 0011
                //4800 bit/s = 103 = 0110 0111
                //2400 bit/s = 207 = 1100 1111

void uart_init(unsigned int ubrr)
{
    /* Nastavení přenosové rychlosti */
    UBRR0H = (unsigned char) (ubrr>>8);
    UBRR0L = (unsigned char) ubrr;

    UCSR0A = (0<<U2X0);
    /* Povolení vysílání a příjmu */
    UCSR0B = (1<<RXEN0) | (1<<TXEN0);

    /* Nastavení formátu rámce: 8N2 */
    UCSR0C = (1<<USBS0) | (3<<UCSZ00);
}
```

Na dalších řádcích kódu je nastavován registr UCSR0B. Využívá se jeho 4. a 3. bitu které povolují vysílání a příjem. Dále umožňuje povolení tvorby přerušování při příjmu a odovysílání dat případně je-li prázdný datový registr.

Tab. 4.3: Registr UCSRnB.

**USART řídicí a stavový registr B - UCSRnB**

Bit	7	6	5	4	3	2	1	0	
	<b>RXCIE<sub>n</sub></b>	<b>TXCIE<sub>n</sub></b>	<b>UDRIE<sub>n</sub></b>	<b>RXEN<sub>n</sub></b>	<b>TXEN<sub>n</sub></b>	<b>UCSZ<sub>n2</sub></b>	<b>RXB8<sub>n</sub></b>	<b>TXB8<sub>n</sub></b>	UCSRnB
Čtení/zápis	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Počáteční hodnota	0	0	0	0	0	0	0	0	

Tab. 4.4: Registr UCSRnC.

**USART řídicí s stavový registr C - UCSRnC**

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Čtení/zápis	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Počáteční hodnota	0	0	0	0	0	1	1	0	

- Bit 7..6 - UMSEL - vybírá mód komunikace; v programu použit asynchronní UART (00)
- Bit 5..4 - UPM - nastavují typ použité parity; při nevyužití se nastavují jako nulové
- Bit 3 - USBS - tento bit specifikuje kolik se použije stop-bitů; program používá dva (1)
- Bit 2..1 - UCSZ - nastavuje jak velký rámec dat se posílá; v programu se využívá 8 bitů (011)

Při nastavení formátu rámce lze vybrat z možností:

- 1 stop bit
- 5, 6, 7, 8 nebo 9 datových bitů
- paritu lze nastavit žádnou, sudou nebo lichou
- počet stop bitů 1 nebo 2

Pro samotné vysílání dat se využívá funkce `uart_transmit`. Nejdříve se čeká až je bit UDREO registru UCSR0A prázdný čímž se zjistí, že v datovém registru už nejsou data. Následně se do něj uloží nová a jednotka Uart je již sama vyšle. Při příjmu je rozdíl pouze v tom, že se data do paměti UDR nevkládají, ale naopak se vyčtou a funkce je vrací jako návratovou hodnotu.

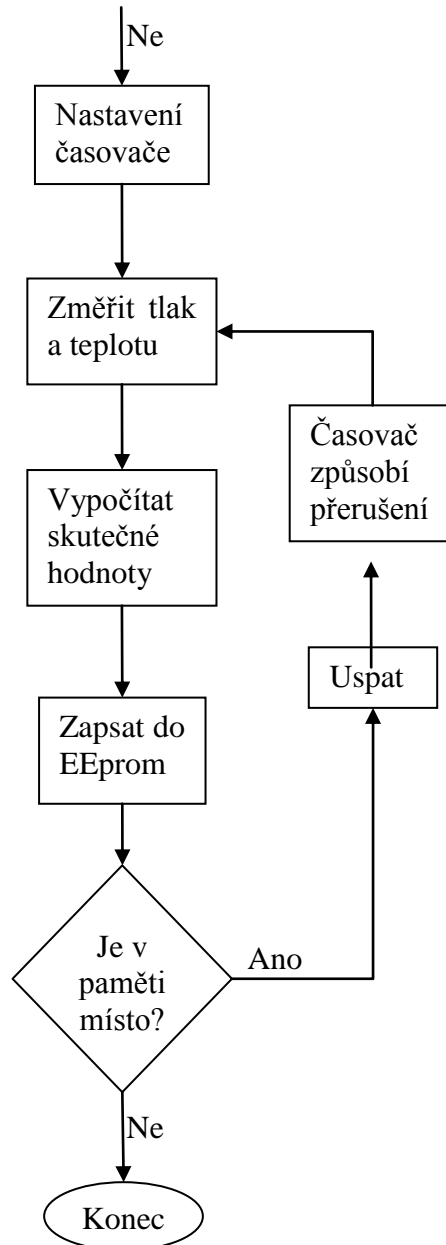


Druhá větev programu je určena pro samotné měření. Jako první se tak nastavuje časovač sloužící k probuzení zařízení.

```

TCNT2=0x00;
TIMSK2=(1<<TOIE2); //vybere přerušení od časovače 2
sei(); //povolí přerušení
TCCR2B = 7; //nastaví časovač 2 s předděličkou 1024

```



Obr. 4.3: Vývojový diagram části programu pro měření.

Následuje samotné měření, výpočet skutečné hodnoty a zápis do paměti. V dalším kroku se vyhodnotí je-li ještě kam zapisovat. Pokud ano, zařízení je usnáno a vyčká tak do dalšího měření.

Jako další je nastaven mód snížené spotřeby a je aktivován. Vše se provádí v registru SMCR. Pro uvedení MCU do stavu snížené spotřeby lze taktéž využít knihovnu sleep.h a k nastavení využít jejich instrukcí.

```
#include <avr/sleep.h>

/* výběr módu IDLE */
set_sleep_mode(SLEEP_MODE_IDLE);
/* usnutí zařízení */
sleep_mode();
```

K probuzení je využíván časovač 2 jenž je uveden v tabulce 2.4 jako možný zdroj přerušení. Jelikož je však pouze osmibitový dosáhne se i s nastavenou předděličkou na maximum jeho přetečení po 33 ms. Z toho důvodu je výše zmíněné usnutí prováděno v cyklu for celkem třicetkrát aby se dosáhlo času 1 s po který výškoměr vyčkává na další měření.

Výpočet doby přetečení časovače [5]

$$t_{ovf} = \frac{1}{f_{CPU}} \times N \times 2^n, \quad (4.3)$$

kde N je hodnota předděličky (1024) a n je počet bitů časovače (8).

#### 4.1.1 Programová část měření

V této kapitole je na několika částech kódu ukázáno jak je prováděno samotné měření.

Pro správný výpočet skutečných hodnot tlaku se používá aktuální hodnota teploty, proto je třeba ji taktéž změřit. Následující část kódu provádí měření tlaku. Jak již bylo popsáno v kapitole 2.2.2, rozdíl mezi měřením teploty a tlaku spočívá v rozdílném zápisu do registru 0xF4. Změnou této hodnoty lze i přenastavit přesnost měření (osrs) čidla jak je ukázáno v Tab 2.2.

```
/*-----MERENI_CIDLA_BMP085-----TLAK-----*/
void Bmp_mereniPzap(char reg ,short EEadresa)
{
    unsigned long pres=0;

    twi_start();
    twi_address_w( 0xEE );           //adresa zarizeni+WRITE => 0xEE
    twi_write( 0xF4 );               // adresa registru 0xF4
    twi_write( reg );                // 0x34 - meri tlak      osrs =0
                                    // 0x74 - meri tlak      osrs =1
                                    // 0xB4 - meri tlak      osrs =2
                                    // 0xF4 - meri tlak      osrs =3

    twi_stop();
    _delay_ms( 30 );
    //-----cteni_vysledku_mereni-----
    twi_start();
    twi_address_w( 0xEE );           //adresa zarizeni+WRITE - EE
    twi_write( 0xF6 );               //0xF6 - MSB vysledku
```

```

twi_start();
twi_address_w( 0xEF );           //adresa zarizeni+Read -EF
twi_readA();                     // vysledek z registru F6-MSB
                                pres=TWDR;
                                pres<<=8;

if(reg==0xF4)                  // Pri nastaveni max rozliseni ma vysledek 19
                                // bitu v 3 registrech - 2 zbývají
{ twi_readA();
  pres|=TWDR;
  pres<<=8;

  twi_readNA();                 // vysledek z registru F7-LSB
  pres|=TWDR;
  pres>>=5;
}
else                          // U výsledku s 16 bity zbývá načíst pouze 8
{ twi_readNA();                 // vysledek z registru F7-LSB
  pres|=TWDR;
}

twi_stop();

switch (reg)
{
  case 0x34:
    pres=vypP(0,pres);          // Vypočítá tlak v Pa
    EEzapis24c ( EEadresa, pres); /*Uloží výsledek (pres)
                                do paměti na adresu ( EEadresa)*/
  break;
  case 0x74:
    pres=vypP(0,pres);          // Vypočítá tlak v Pa
    EEzapis24c ( EEadresa, pres); /*Uloží výsledek (pres)
                                do paměti na adresu ( EEadresa)*/
  break;
  case 0xB4:
    pres=vypP(0,pres);          // Vypočítá tlak v Pa
    EEzapis24c ( EEadresa, pres); /*Uloží výsledek (pres)
                                do paměti na adresu ( EEadresa)*/
  break;
  case 0xF4:
    pres=vypP(3,pres);          // Vypočítá tlak v Pa
    EEzapis24c ( EEadresa, pres); /*Uloží výsledek (pres)
                                do paměti na adresu ( EEadresa)*/
  break;
}
return;
}

```

Jak je vidět, MCU posílá adresu registru 0xF4 načez posílá co chce změřit. Po restartu komunikace se pošle adresa na registr 0xF6 obsahující vyšších 8 bitů výsledku. Posláním potvrzení v podobě ACK bitu čidlo pošle nižších 8 bitů. Pro případ, kdy je vybráno měření s nejvyšší přesností má výsledek 19 bitů. zbývá tak načíst 3 bity jež jsou uloženy na následující adrese a čidlo je pošle odpoví-li MCU i napodruhé bitem ACK. K rozhodnutí zda je potřeba odpovídata i napodruhé se využívá podmínky `if`, která na základě vybrané přesnosti měření rozhodne zda se čte i výsledek uložený na adrese 0xF8.

Podmínka switch se taktéž rozhoduje na základě zvolené přesnosti měření o tom, jak se vypočítá skutečná hodnota tlaku. Na obrázku 2.6 je ukázáno jak se těchto hodnot dá dosáhnout. Výsledek se zapisuje do paměti na adresu jenž byla celé funkci předána jako argument a její hodnota je nastavována funkcí main.

V případě zjišťování kalibračních koeficientů čidla se namísto registru 0xF4 posílá adresa registru v němž je uložen zjišťovaný koeficient. Jejich adresy jsou uvedeny v tabulce 2.1.

Na příkladech kódu je vidět, že data se posílají po 8 bitech a při proměnné větší jak char je nutné přijatá data bitově posunout aby se významnější byty dostali na svá místa. Poslední důležitou funkcí programu je zápis do paměti, kde se taktéž musí vysílaná data rozdělit. Jelikož jsme z měření dostali 16 (19) bitovou hodnotu, kterou jsme pak dále přepočítali, tak ve výsledku je tlak vyjádřen 32 bitovou proměnnou. Do paměti však lze zapisovat jen po osmi bitech a proto každou naměřenou hodnotu je potřeba rozdělit na 4 části - po bajtech.

Při zápisu se po adrese pro uložení prvního bajtu, posílají samotná data. Ty jsou uložena v proměnné typu char a je tak do ní uloženo pouze nejnižších 8 bitů, jenž jsou následně poslány. Pro poslání vyššího bajtu jsou data bitově posunuta a opět uložena do proměnné char a poslána. Paměť je automaticky ukládá na adresy po sobě následující.

K tomuto byla vytvořena funkce EEzapis24c. Její vstupní parametr iEEadresa obsahuje adresu buňky v paměti kam se naměřený tlak *tempo* uloží.

```
void EEzapis24c ( int iEEadresa, long tempo)
{
  char prom;
  prom=tempo;
  twi_start();
  twi_address_w( 0xA0 );           //adresa zarizeni+WRITE - A0

  twi_write( iEEadresa );         //adresa slova - kam se zapise
  twi_write( prom );              // zapis dat - bit 1.
  tempo>>=8;
  prom=tempo;
  twi_write( prom );             // zapis dat - bit 2.
  tempo>>=8;
  prom=tempo;
  twi_write( prom );            // zapis dat - bit 3.
  tempo>>=8;
  prom=tempo;
  twi_write( prom );           // zapis dat - bit 4.

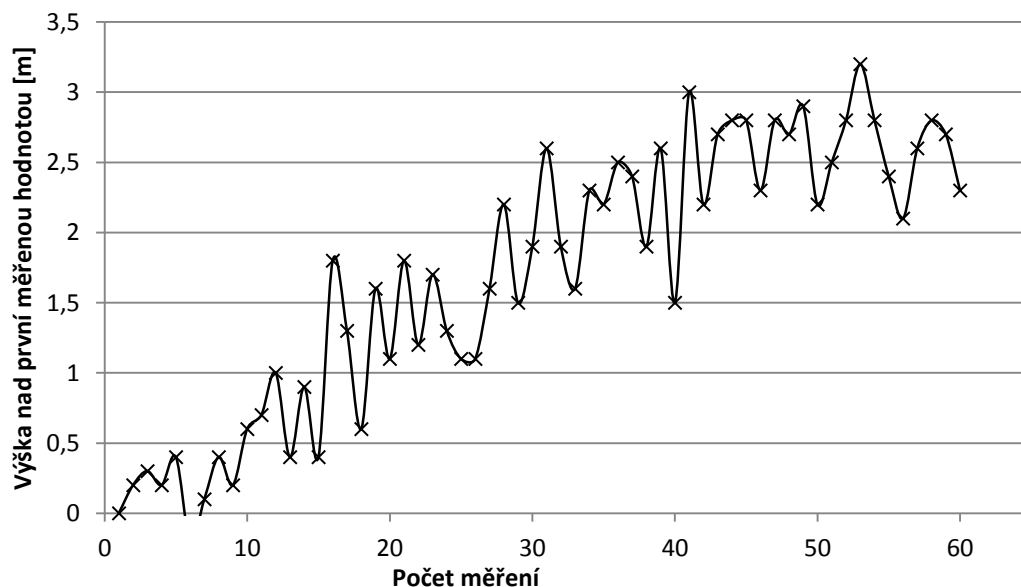
  twi_stop();
  return;
}
```

## 5 VÝSLEDKY MĚŘENÍ

Pro zpracování výsledků měření byl napsán program jehož ukázka je na obrázku 6.2. Je díky němu možno načíst celou paměť výškoměru kliknutím na tlačítko Načíst. Přijmutá data se vypíše do sloupce tlak. Do vedlejšího sloupce se vepíše výsledky přepočtu rozdílu tlaku mezi jednotlivými měřeními na metry. Údaje o výšce se dále použijí k vytvoření grafu. Je zde i možnost měnit měřítko osy výšky. Zbylá tlačítka ještě umožňují jednotlivé měření a nebo spuštění měření které zaplní celou paměť výškoměru. Přijaté údaje se dají též uložit.

Bylo zjištěno, že změřený tlak se neustále mění. Položí-li se čidlo a nechá se měřit, pak výsledky tlaku budou neustále kolísat v rozmezí až  $\pm 30$  Pa. Mnohem častěji se však udrží v rozmezí  $\pm 10$  Pa (1 m). Nedá se počítat, že se změří tlak v jednom místě a pak v jiném a výsledek je rozdíl v metrech. Při měření je třeba změřit několik hodnot na obou místech, čím více tím lépe, a jejich střední hodnoty použít jako ty které se budou porovnávat.

Následující graf jasně ilustruje výsledky jenž dává tlakové čidlo. Při jeho měření byl výškoměr položen na podlaze a postupně plynule zvedán až ke stropu. Vezme-li se jako výsledná hodnota výška 2,5 m pak čidlo měří naprosto přesně.



## 6 ZÁVĚR

Cílem této bakalářské práce bylo vytvořit výškoměr pro RC model schopný zaznamenat údaje o letu.

Zařízení bylo navrženo, zrealizováno a ověřena jeho funkčnost. Je řízeno mikrokontrolérem a pro určení výšky se využívá tlakové čidlo.

Při návrhu bylo dbáno na požadavek minimálních rozměrů, ale možnost ke zmenšení zde je. Stejně tak požadavek na minimální proudový odběr je zohledněn. Použité čidlo má proudový odběr při měření jen  $10 \mu\text{A}$  a spotřeba mikrokontroléru byla zredukována využitím módu snížené spotřeby. Zde je však prostor pro zlepšení při použití externího krystalu lze využít ještě účinnější mód.

Data je možno po skončení letu stáhnout do počítače přes kabel usb. K jejich zpracování byl napsán program, který je prezentuje i ve formě grafu.

Změřená data jsou v rámci zaměření přístroje dostatečně přesná. Měříme-li výšky v řádech desítek metrů nad zemí není nutné znát hodnoty v přesnostech desítek centimetrů. Používal-li by se však takovýto výškoměr k měření přesnějších údajů, bylo by nutné výsledná data jenž zařízení dává zpracovávat pokročilejším způsobem než je pouhé vypočtení rozdílu dvou hodnot. Například i pouhé průměrování dvou po sobě jdoucích hodnot by mohlo stačit k získání hladkých křivek.

# LITERATURA

- [1] Atmel Corporation. *ATmega168-20PU* [online]. 2013 [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.atmel.com/devices/ATMEGA168.aspx>>
- [2] Burkhard, Mann. *C pro mikrokontroléry*. 1. české vydání. Praha: BEN, 2003. 280s. ISBN 80-7300-077-6
- [3] Balon. *Meteorologie: Teplota, tlak a vlhkost vzduchu* [online]. 2013 [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.balon.cz/balonove-letani/pro-piloty-a-posadky/ucebni-materialy>>
- [4] Fyzmatik. *Atmosférický tlak* [online]. Vytvořeno 27. května 2009 [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://fyzmatik.pise.cz/903-atmosfericky-tlak.html>>
- [5] Frýza, Tomáš. *Mikroprocesorová technika a embedded systémy* [online]. Verze vydaná 16. 12. 20011 [cit. 2013-05-01]. Dostupné na World Wide Web: <[http://www.urel.feec.vutbr.cz/~fryza/downloads/mpt\\_prednasky\\_2011-12-16.pdf](http://www.urel.feec.vutbr.cz/~fryza/downloads/mpt_prednasky_2011-12-16.pdf)>
- [6] GM electronic. *24C08* [online]. [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.gme.cz/pameti-EEPROM-seriove/24c08-p414-017/>>
- [7] GM electronic. *FT232RL* [online]. [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.gme.cz/i-o-obvody-multifunkcni-periferie/ft232r1-p959-303/>>
- [8] GM electronic. *LF33CV* [online]. [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.gme.cz/linearni-regulatory-napeti-pevne/lf33cv-p330-165/>>
- [9] Němec, Miloš. *MEMS technologie* [online]. Vytvořeno 4.5.2007 [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.milosnemoc.cz/clanek.php?id=101>>
- [10] Pvelectronic. *BMP085* [online]. [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.pvelectronic.eu/Senzory-Mereni-Zaznam/Tlak/Barometric-Pressure-Sensor-BMP085-Breakout.html>>
- [11] Vernier. *Změna tlaku s výškou* [online]. [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://www.vernier.cz/download/namety/zmena-tlaku-s-vyskou.pdf>>
- [12] Wikipedie. *Barometrická formule* [online]. Poslední editace 16. 4. 2013 [cit. 2013-05-01]. Dostupné na World Wide Web: <[http://en.wikipedia.org/wiki/Barometric\\_formula](http://en.wikipedia.org/wiki/Barometric_formula)>
- [13] Wikipedie. *Výškoměr* [online]. Poslední editace 17. 5. 2013 [cit. 2013-05-01]. Dostupné na World Wide Web: <<http://cs.wikipedia.org/wiki/V%C3%BD%C5%A1kom%C4%9Br>>

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

$p$	Atmosférický tlak v místě měření
$p_0$	Atmosférický tlak v referenčním místě
MCU	Microcontroller unit, mikrokontroler
TWI	Two Wire Interface, dvou vodičové propojení
SDA	Serial Data Line, sériová datová linka
SCL	Serial Clock Line, sériová hodinová linka
ACK	Acknowledge, potvrzení
UART	Universal Asynchronous Receiver and Transmitter, univerzální asynchronní přijímač a vysílač
Eeprom	Electrically Erasable Programmable Read-Only Memory, elektricky mazatelná a programovatelná paměť
USB	Universal Serial Bus, univerzální sériová sběrnice
High	Vysoká logická úroveň signálu, logická 1
Low	Nízká logická úroveň signálu, logická 0
CPU	Central Processing Unit, centrální výpočetní jednotka

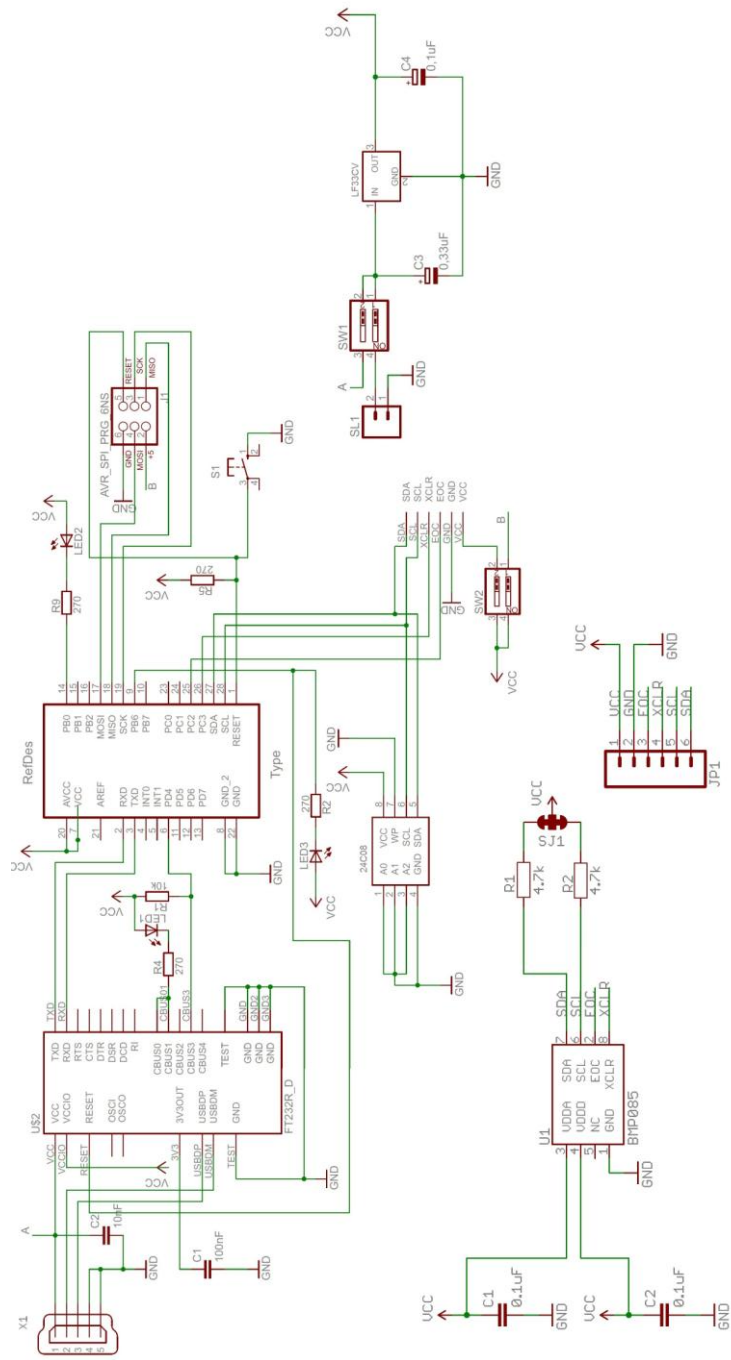


# SEZNAM PŘÍLOH

<b>A</b>	<b>Návrh zařízení</b>	<b>33</b>
A.1	Obvodové zapojení .....	33
A.2	Deska plošného spoje – top (strana součástek).....	34
A.3	Deska plošného spoje – bottom (strana spojů) .....	34
<b>B</b>	<b>Program pro zpracování výsledků měření</b>	<b>35</b>
<b>C</b>	<b>Spotřeba mikrokontroléru</b>	<b>36</b>
<b>D</b>	<b>Seznam součástek</b>	<b>37</b>

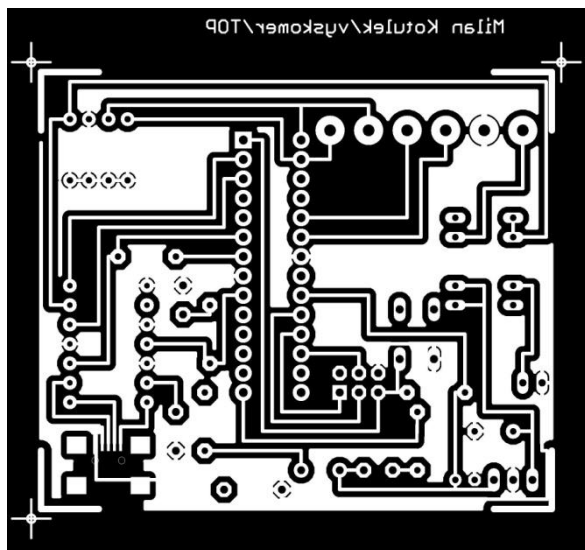
# A NÁVRH ZAŘÍZENÍ

## A.1 Obvodové zapojení



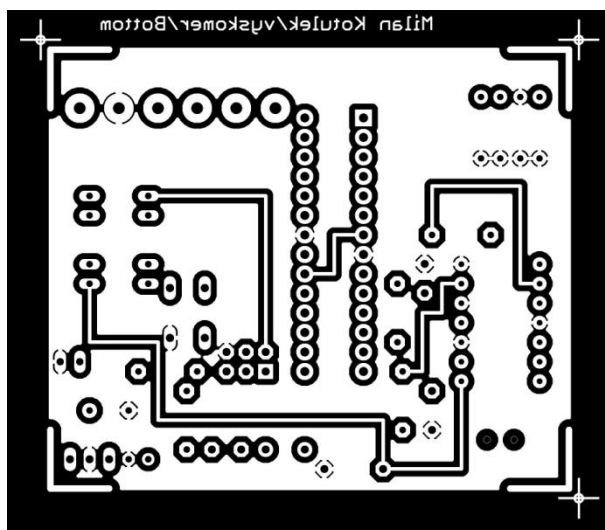
Obr. 6.1: Celkové obvodové zapojení výškoměru.

## A.2 Deska plošného spoje – top (strana součástek)



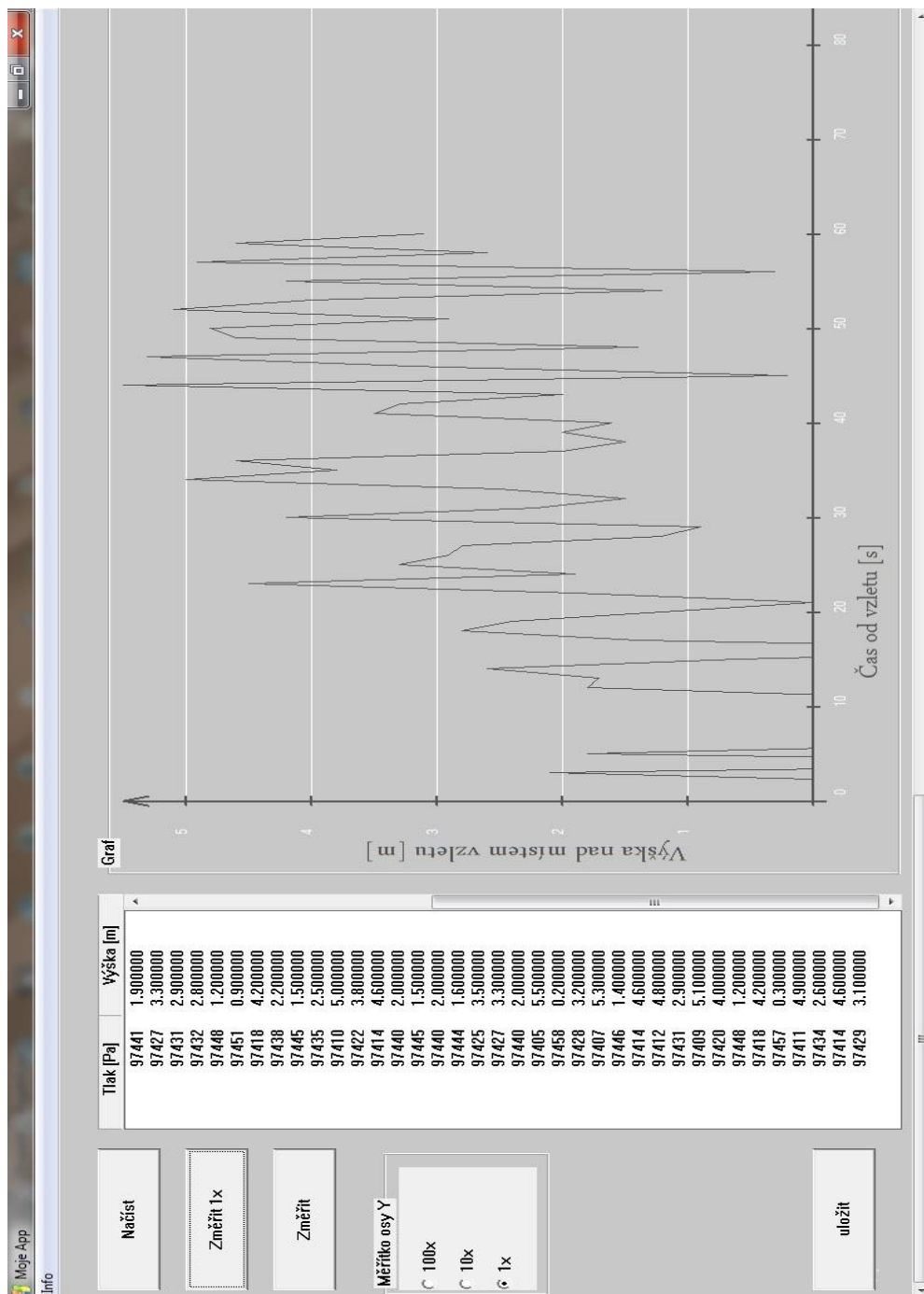
Rozměr desky 65 x 55 [mm], měřítko M1:1

## A.3 Deska plošného spoje – bottom (strana spojů)



Rozměr desky 65 x 55 [mm], měřítko M1:1

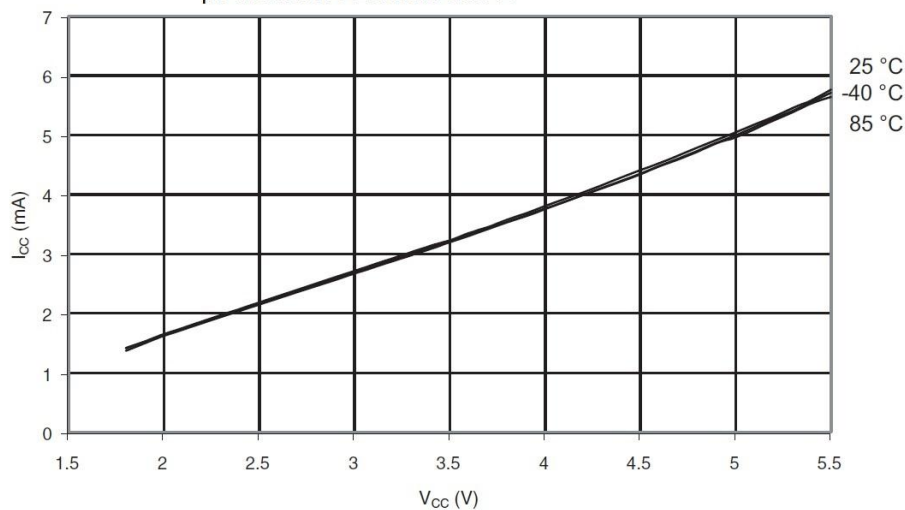
## B PROGRAM PRO ZPRACOVÁNÍ VÝSLEDKŮ MĚŘENÍ



Obr. 6.2: Ukázka programu pro PC.

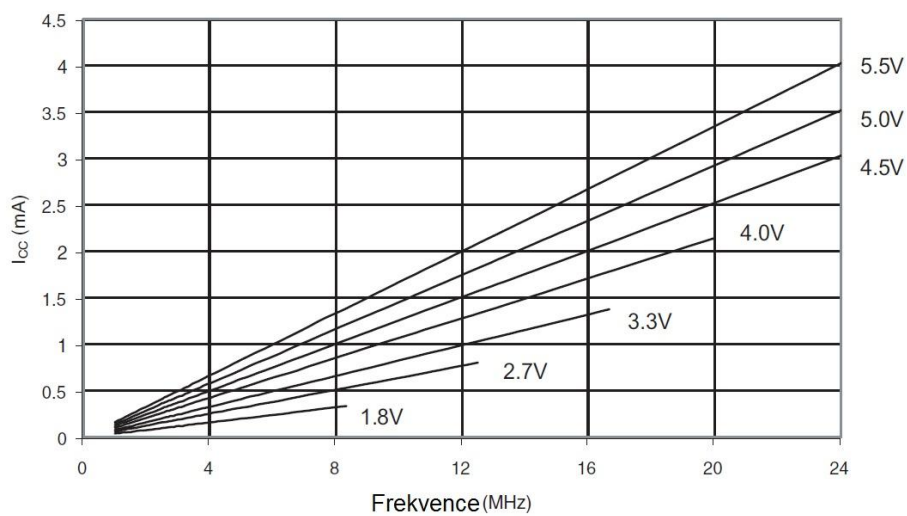
## C SPOTŘEBA MIKROKONTROLÉRU

**Závislost proudového odběru na napájecím napětí v aktivním režimu  
při interním oscilátoru 8MHz**



Obr. 6.3: Závislost proudového odběru v aktivním režimu [1].

**Závislost proudového odběru při Idle módu**



Obr. 6.4: Závislost proudového odběru při snížené spotřebe módu Idle [1].

## D SEZNAM SOUČÁSTEK

Označení	Hodnota	Pouzdro	Popis
C1	10nF /0,01uF	5 mm	Keramický kondenzátor
C2	100nF /0,1uF		Elektrolytický kondenzátor
C3	0,33uF	5 x 11 mm	Elektrolytický kondenzátor
C4	0,1uF	4 x 7 mm	Elektrolytický kondenzátor
LED	3mm 3x		Svítivá - zelená
R1	270 $\Omega$ 3x	R0206	Metalizovaný rezistor
R2	10k $\Omega$	R0207	Metalizovaný rezistor
R3	4k7 $\Omega$ 2x	R0207	Metalizovaný rezistor
FT232RL	FT232RL	SSOP	Převodník USB - UART
Bmp085	Bmp085	LCC8	Tlakové čidlo
24C08	24C08	DIP8	Sériová paměť Eeprom
Atmega	ATMEGA168-20PU	DIL28	Mikrokontrolér
S1	TD-03XG SMD	6 x 6 x 3,5 mm	Mikrospínač SMD
Mini USB	USB Mini B SMD	USB Mini B	USB konektor
LF33CV	3,3 V / 0,5 A	TO220	Lineární regulátor napětí 3,3V/0,5A
Avr.SPI	MLW06G	2x3 kontakty	Konektor pro ploché kabely do DPS přímý
SL1	DCI 006-PI	klips baterie 9V	Napájecí konektor
SW1,SW2	DIP 02	DIP	DIP spínač do DPS