



Středoškolská technika 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

INTELIGENTNÍ METEOROLOGICKÁ STANICE

Patrik Bachan

SPŠE Ječná
Ječná 30, Praha 2

Abstrakt

Tento dokument popisuje celý vývoj a stavbu vlastní meteorologické stanice s důrazem na otevřenost softwaru i hardwaru. Je zde zahrnut vývoj elektroniky, včetně výběru senzorů a kompletního osazení, návrh a výroba krabičky pro umístění stanice do exteriéru. Dále je vysvětlena funkce jednotlivých komponent a napsaného softwaru.

Obsah

1 Úvod	2
2 Atmosférické jevy a jejich vliv	3
2.1 Funkce meteorologických stanic	3
3 Návrh	4
3.1 Řídící mikrokontrolér	4
3.1.1 Programování bootloaderu	5
3.1.2 Nahrávání programu	5
3.2 Tvorba prototypu	5
3.2.1 obslužný program	5
3.3 Senzory	7
3.3.1 Senzor vlhkosti - DHT11	7
3.3.2 Senzor teploty a tlaku - BMP085	9
3.3.3 kalibrace	10
3.4 Napájení	10
3.5 Krabíčka	10
4 Stavba	16
4.1 Výroba PCB	16
4.2 osazení PCB	17
4.3 tepelná izolace elektroniky	17
4.4 umístění v krabíčce	19
4.5 Odstranění chyb	19
5 Server	21
5.1 Získávání dat z meteostanice	22
5.2 Úložiště dat	22
5.3 Zpracování dat	22
5.3.1 Tvorba grafů	23
5.3.2 Četnost měření	24
5.4 webová aplikace	24
6 Závěr	26
6.1 Použitý software	26

1 Úvod

Jako téma maturitní práce jsem si zvolil výrobu meteorologické stanice s webovým přístupem. Mým cílem bylo vytvořit komplexní zařízení navržené zcela od základu. Zvolil jsem maximální otevřenost projektu, celý je open-source.

Toto téma mě jsem si vybrat, protože mám již delší zkušenost s programováním ať už 8 bitových nebo 32 bitových mikrokontrolérů v jazyce C a také v programování aplikací pro PC v jazyce python. Tento projekt zahrnuje obojí. Pro mojí meteorologickou stanici jsem se rozhodl vytvořit elektronickou část zcela podle sebe bez použití již hotových zařízení vyjma případných modulů pro jednodušší sestavení.

2 Atmosférické jevy a jejich vliv

Atmosféra obklopuje celý Zemský povrch a tak ovlivňuje vše na něm. Aktuální stav ovzduší se nazývá počasí, což je souhrn všech atmosférických jevů v daném místě v určitém čase. Mezi nejvýznamnější patří teplota vzduchu, atmosférický tlak, relativní vlhkost, rychlost a směr větru, informace o srážkách a další.

Dnes již není problém tyto jevy sledovat elektronicky a to i na vlastní meteorologické stanici.

2.1 Funkce meteorologických stanic

Běžné komerčně prodávané meteorologické stanice nabízejí především měření atmosférického tlaku, teploty vzduchu a relativní vlhkosti. V případě dražších zařízení jsou funkce rozšířené například o měření rychlosti a směru větru, prašnosti, množství srážek, intenzity slunečního svitu a další. Tato zařízení se obvykle skládají ze dvou částí. Ta, kterou člověk běžně vidí, je zobrazovací jednotka umístěná někde v interiéru. Podle vybavenosti se může dozvědět aktuální změřené hodnoty, jejich vývoj v posledních dnech. Některé meteorologické stanice dokonce nabízejí možnost predikce počasí. Druhá část stanice je umístěna venku, kde získává informace o aktuálním stavu měřených veličin. Zde jsou umístěny hlavní senzory. Přenos informace do zobrazovací jednotky je řešen po metalických vodičích nebo bezdrátově, záleží na požadavcích na umístění.

Tyto komerčně vyráběné stanice disponují určitou sadou senzorů, která se nedá rozšiřovat. Pokud již výrobce umožňuje rozšíření, dodává pouze vlastní moduly se senzory.

Stanice, kterou tvořím já, má zcela otevřený zdrojový kód i veškerou dokumentaci. Není tedy složité implementovat podporu pro další senzory a tak získat další informace o počasí i jiných jevech.

3 Návrh

Na začátku jsem stál před volbou co vše má moje meteorologická stanice měřit. Po několika úvahách jsem se rozhodl pro měření základních veličin a to: teploty tlaku a relativní vlhkosti. Ovšem celý návrh stanice je tvořen s ohledem na plánované rozšíření o další senzory.

Funkci webového serveru zastává nízko nákladový minipočítač RaspberryPi. Vybral jsem jej pro malou spotřebu a dobrou dostupnost. Jako dlouhodobé úložiště naměřených hodnot slouží mysql databáze běžící právě na tomto minipočítači. Grafy z naměřených hodnot se nejprve vykreslovaly na straně prohlížeče (klienta) pomocí javascriptu. Problém nastal na zařízeních s nízkým výpočetním výkonem, kde toto vykreslení trvalo i desítky sekund. Nakonec jsem tedy zvolil řešení, že jsou na straně serveru vytvořeny grafy v podobě obrázků a ty jsou zobrazovány i na těchto zařízeních bez problémů.

3.1 Řídící mikrokontrolér

Celá meteostanice potřebuje pro svůj chod mikrokontrolér, který na požadavek získá ze senzorů aktuální hodnoty a pošle je po sériové lince. Veškerou činností mikrokontroléru je komunikace se senzory, provedení přepočtu hodnot, sestavení jednoduchého rámce a odeslání. Pro takto jednoduchý účel zcela postačí 8 bitový mikrokontrolér. Jelikož mám zkušenosti s mikrokontroléry od firmy Atmel, konkrétně s řadou ATmega, zvolil jsem mě osvědčený typ ATmega8.

Tabulka 3.1: Parametry mikrokontroléru

Napájecí napětí	4.5V až 5.5V
Maximální taktovací frekvence	16MHz
Velikost FLASH	8kB
Velikost RAM	1kB

3.1.1 Programování bootloaderu

Pro jednodušší nahrávání nových programů, jsem nahrál do mikrokontroléru bootloader. To je speciální program, který umožňuje zapisovat programy do FLASH paměti přes jiné rozhraní než SPI, kde je vyžadován ISP programátor. Nový program lze nahrát přes stejnou sériovou linku, kterou se běžně posílají naměřené hodnoty. Z důvodu kompatibility jsem vybral bootloader, který využívají vývojové desky arduino. Výhodou je podpora v širokém spektru nahrávacího softwaru. Kvůli změně desky, je stavová LED umístěna na jiném GPIO pinu mikrokontroléru. V nastavení bootloaderu byl tedy překonfigurován tento pin a zároveň zkrácena doba po kterou čeká bootloader na příjem příkazu pro nahrávání nového programu od počítače do paměti FLASH, tím se urychlil start stanice do pohotovostního stavu.

3.1.2 Nahrávání programu

Pro nahrání nového programu se musí mikrokontrolér resetovat. Toho lze dosáhnout odpojením a opětovným připojením napájení meteostanice. Druhou možností je poslat meteostanici příkaz 'R', který ji resetuje softwarově. Při každém startu čeká bootloader několik sekund a pokud nepřijde příkaz pro nahrávání nového programu, je spuštěn aktuální program.

V době kdy bootloader čeká na příkaz a příjem nového programu, je potřeba spustit nahrávání nového programu, typicky z počítače. K tomu lze využít třeba program avrdude.

Takto lze nahrát novou verzi programu do meteostanice.

3.2 Tvorba prototypu

Předtím, než bylo možné navrhnout plošný spoj, krabičku a napsat celý řídicí program, vytvořil jsem prototyp na vyzkoušení komunikace mikrokontroléru s počítačem a senzory.

3.2.1 obslužný program

Program pro mikrokontrolér je napsán v jazyce C. Pro definice registrů, kompilaci a linkování je použit GNU toolchain pro AVR mikrokontroléry.

Celý program je napsán od základu s výjimkou použití externí knihovny pro práci s rozhraním i2c. Mikrokontrolér má nyní připojený kombinovaný senzor tlaku a teploty na i2c sběrnici. Senzor relativní vlhkosti je připojený přímo na GPIO pin, který musí být řízen softwarově.

Jakmile je k meteostanici připojeno napájení, hardware mikrokontroléru počká na jeho ustálení a resetuje jej. Následujících několik sekund čeká

na případné nahrávání nového programu. Pokud zařízení na druhém konci sériové linky nepošle příslušný příkaz, skončí čekání bootloaderu a přejde se na začátek programu.

běh programu

Nejdříve jsou nastaveny všechny periferie, tj. sériová komunikace, i2c sběrnice a jsou nastaveny příslušné GPIO piny. Běžný program s nezakompilovaným ladicím výstupem pouze nahlásí po sériové lince úspěšný start. Při startu jsou zároveň staženy kalibrační konstanty z kombinovaného senzoru teploty a tlaku. Pokud vše proběhne bez chyby, je zapnuto přerušování příchozích dat na sériové lince a mikrokontrolér je uveden do úsporného stavu.

Jakýkoliv příchozí byte na sériové lince zapříčiní skok na obsluhu přerušování. Přijatý znak je vyhodnocen. Není-li znak 'Q' ani 'R' je odeslána odpověď s informací o možných požadavcích tj. R nebo Q a mikrokontrolér je opět uveden do úsporného stavu.

Pokud je přijatý znak 'R', mikrokontrolér jej vyhodnotí jako požadavek na reset. Jelikož samotný mikrokontrolér nemá přímou možnost softwarově vyvolaného resetu, je zapnut watchdog, což je periferie, která umožňuje resetovat mikrokontrolér v případě, že jej software nenuluje. Po přetečení watchdogu je mikrokontrolér resetován a začíná opět znova celou startovací sekvencí včetně čekání na příjem nového programu.

V případě, že je přijatým znakem 'Q', je zahájen sběr dat ze senzorů. Jsou získána data o teplotě tlaku i vlhkosti. Následně je vypočtena teplota a kalibrační konstanty pro tlak. Ze získaných hodnot je vypočten tlak. Hodnotu vlhkosti není nutno nijak převádět. Všechny získané hodnoty jsou odeslány po sériové lince v předem určené formě jednoduchého paketu. Mikrokontrolér po ukončení přenosu přechází do úsporného stavu a čeká na další požadavky.

formát paketu

Hodnoty posílané po sériové sběrnici jsou posílány v ASCII kódu. Formát zprávy jsem navrhl tak, aby bylo možné rozeznat stanici, veličinu, hodnotu a konec se začátkem. Jednotlivé části jsou od sebe odděleny čárkou, Za celým paketem následuje zalomení řádku.

Příklad přijaté zprávy z meteostanice:
!,MULTISNS1,T0,12,P0,97679,H0,56,\$

Tabulka 3.2: popis částí paketu

!	uvozující znak
MULTISNS1	jméno stanice
T0	teplotní senzor 0
12	teplota (12°C)
P0	senzor tlaku 0
97679	nekorigovaná hodnota tlaku v desetinách pa
H0	senzor vlhkosti 0
56	relativní vlhkost v %
\$	ukončovací znak

3.3 Senzory

Řídící mikrokontrolér má periferie pro komunikaci po sběrnici i2c, SPI, UART, ale disponuje pouze 10 bitovým AD převodníkem. Aby bylo možné zachovat jednoduchost a nízkou cenu stanice, volil jsem digitální senzory. Takové senzory mají již integrovaný vlastní AD převodník a jsou i často zkalibrovány už od výrobce. Další výhodou digitálních senzorů je možnost umístit jich více na jednu sběrnici.

3.3.1 Senzor vlhkosti - DHT11

Pro měření relativní vlhkosti jsem vybral senzor s označením DHT11. Jeho výhodou je především cena a dostupnost. Samotný senzor dokáže měřit teplotu a relativní vlhkost. Pro komunikaci využívá vlastní protokol, který je zdokumentován v katalogovém listu senzoru.

Tabulka 3.3: Parametry

Relativní vlhkost	20-80% ±5%
Rozlišení	1%
Napájecí napětí	3V až 5.5V
Proudový odběr při měření	0.5mA až 2.5mA
Proudový odběr v klidu	100 μ A až 150 μ A
Maximální frekvence čtení hodnot	1Hz

protokol

Senzor DHT11 využívá komunikační protokol, který není běžný jako třeba i2c nebo SPI. Pro komunikaci je využit jediný datový vodič. Nejdříve musí

dát mikrokontrolér najevo, že začíná komunikaci. Poté pošle senzor změřená data a následně informuje mikrokontrolér o konci přenosu.

Klidový stav na datovém vodiči je logická jednička tj. 5V, ta je zajištěna pull-up rezistorem. Chce-li mikrokontrolér získat data ze senzoru, musí nejdříve stáhnout napětí na datovém vodiči k 0 voltům na dobu větší než 18 ms. Poté mikrokontrolér uvolní datový vodič a napětí stoupne zpět na 5V. Za 20 až 40 μs senzor uzemní datový vodič na dobu 80 μs . Na dalších 80 μs senzor uvolní datovou linku a nakonec ji opět uzemní. Touto sekvencí je uvozena komunikace mezi mikrokontrolérem a senzorem. Zároveň je díky odpovědi ověřena přítomnost senzoru. viz obrázek 3.4

Za uvozoací sekvencí následuje přenos dvou bajtů hodnoty vlhkosti, dvou bajtů hodnoty teploty a poslední bajt tvoří kontrolní součet.

Na obrázku 3.5 ve časový diagram, kde je porovnána logická nula a jednička. Když senzor posílá logickou nulu, uzemní datový vodič na dobu 50 μs a následně ji uvolní na 26 až 28 μs . Sekvence pro logickou jedničku se liší pouze dobou uvolnění datového vodiče na dobu 70 μs . Pro jednoduchost rozpoznání hodnoty mikrokontrolér čeká na náběžnou hranu, počká 50 μs a zjistí stav datového vodiče. Je-li na něm logická nula pak má přenášený bit hodnotu 0, je-li na datovém vodiči zjištěna logická jednička, je hodnota přenášeného bitu 1. Těchto bitů je celkem přeneseno $5 \times 8 = 40$ tj. zmíněných 5 bajtů.

Celá komunikace je znázorněna na obrázku 3.6 Po odeslání pěti bajtů senzor zakončí komunikaci uzemněním datového vodiče na 54 μs a uvolněním. Senzor přejde do úsporného stavu a čeká na další požadavek od mikrokontroléru.

Kvůli potřebě přesného časování jsem vytvořil programovou smyčku, u které je známá doba běhu. Časovače mikrokontroléru jsem nevyužil kvůli možnému upotřebení pro další senzory. Zároveň v době příjmu dat ze senzoru nemusí mikrokontrolér provádět jinou činnost, programová smyčka v tomto případě nepřináší žádnou nevýhodu.

Listing 3.1: Funkce programu mikrokontroléru, která čte data ze senzoru DHT11

```
void DHT(uint8_t bit)
{
    uint8_t i, j;

    // "say hello"
    DHT_START(1<<bit);
    // expect ACK
    while(DHTI&(1<<bit)); // wait for pull down
    while(!(DHTI&(1<<bit))); // wait for pull up
    while(DHTI&(1<<bit)); // wait for pull down
    // receive 40(1<<bit)s
```

```

for(j=0;j <5;j++)
{
  response[j]=0;//erase old value
  for(i=0;i <8;i++)
  {
    response[j]=response[j]<<1;//shift left
    while(!(DHTI&(1<<bit)));//wait for pull up
    delay_p(50);
    if(DHTI&(1<<bit))//i still pulled up>longer pulse>LOG1
    {
      response[j]|=1;//set lowest (1<<bit)
      while(DHTI&(1<<bit));//wait for pull down
    }
    //else //already pulled down > shorter pulse > LOG0
    // zero is hopefully there (default)
  }
}
//link is down
//end event check
while(!(DHTI&(1<<bit)));//wait for pull up
}

```

3.3.2 Senzor teploty a tlaku - BMP085

Jako senzor tlaku jsem zvolil obvod BMP085, což je digitální senzor atmosférického tlaku kombinovaný s měřením teploty. Senzor komunikuje po sběrnici i2c.

Senzor je dostupný pouze v SMD pouzdru. Abych nemusel toto pouzdro pájet, zakoupil jsem již osazený modul s tímto senzorem. Na jeho plošném spoji je zároveň osazen 3.3 voltový stabilizátor a převodník napěťových úrovní, takže je možné se senzorem komunikovat na 5 voltové TTL. Modul má na kolíkovou lištu vyvedené napájení stabilizátoru, datový a hodinový signál sběrnice i2c, resetovací vstup senzoru a signál indikující konec měření. Lze jej tedy jednoduše zasadit do jiné desky plošných spojů.

Tabulka 3.4: Parametry

Rozsah měřeného tlaku	300hPa až 1100hPa
Napájecí napětí	1.8V až 3.6V
Standardní doba měření	7.5 ms
Frekvence hodinového signálu i2c	až 3.4MHz

3.3.3 kalibrace

Samotný senzor je již od výrobce zkalibrován. V paměti má uložené veškeré kalibrační konstanty. Před měřením je nejprve potřeba tyto hodnoty stáhnout do mikrokontroléru, aby je mohl využít k přepočtu hodnot získaných při měření. Tyto kalibrační konstanty se nemění a není nutné je tedy stahovat před každým měřením. V katalogovém listu není vysvětlen konkrétní význam konstant, výrobce uvádí potřebné výpočty pro převod dat ze senzoru na hodnotu tlaku.

3.4 Napájení

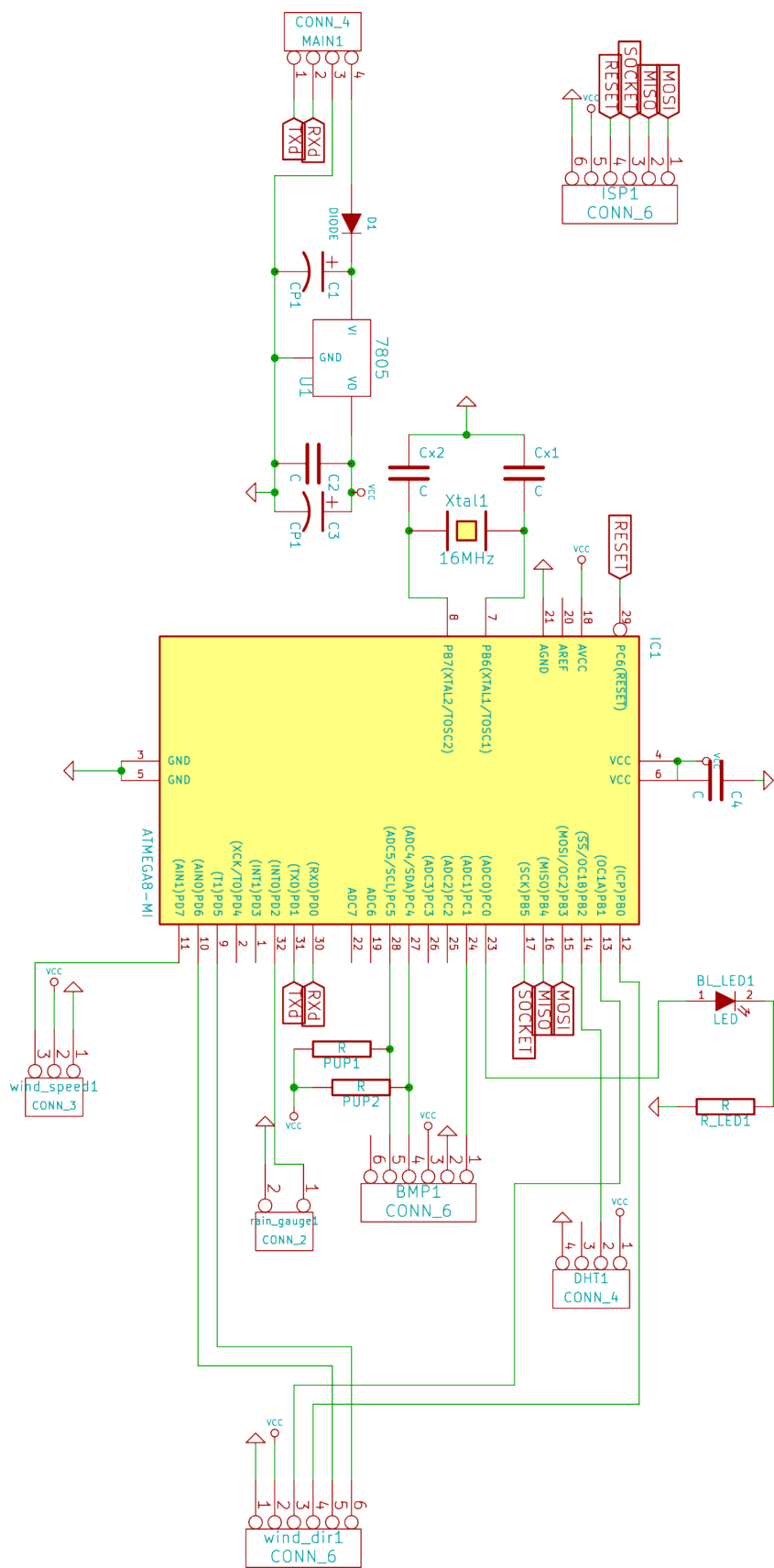
Již od prvotního návrhu byla stanice plánována pro externí umístění zatímco počítač shromažďující data bude umístěn v interiéru. Pro možné umístění stanice do vzdálenosti desítek metrů je napájecí napětí stabilizováno až u mikrokontroléru kvůli úbytku na přívodním kabelu. Je tedy možné stanici napájet libovolným stejnosměrným zdrojem od 7V do 30V s minimálním proudem 100mA.

3.5 Krabička

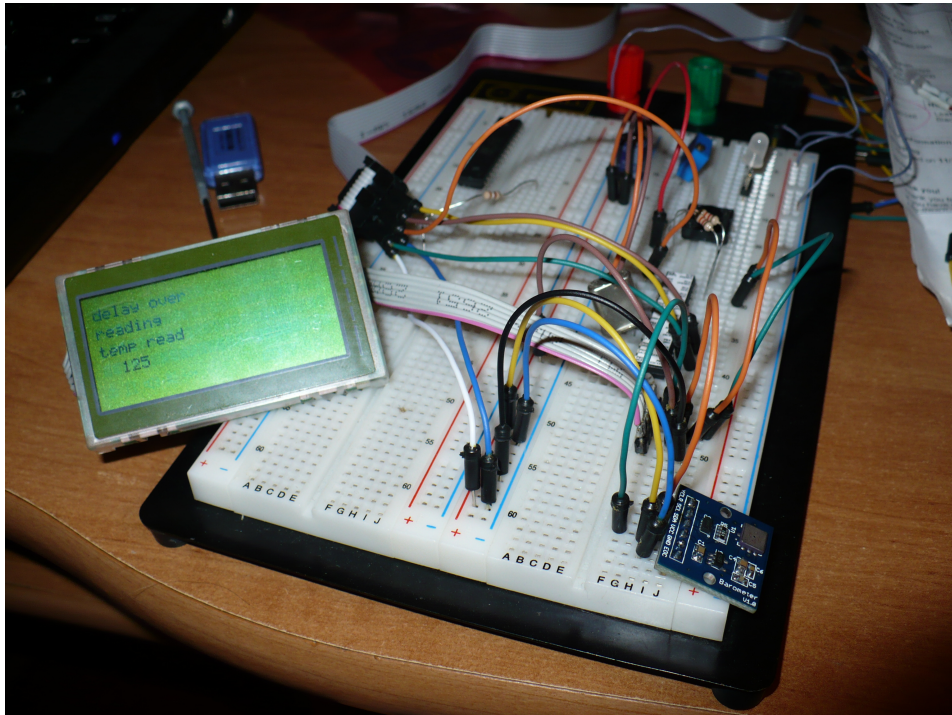
Pro venkovní použití stanice bylo nutné umístit elektroniku do krabičky. Elektronika musí být chráněna před přímým kontaktem s vodou, ale zároveň musí kolem senzorů volně proudit vzduch z okolí. Inspiroval jsem se z designů komerčně vyráběných a navrhl vlastní.

Díly na krabičku jsem vytiskl na naší 3D tiskárně, kterou jsme se spolužákem postavili právě pro takovéto využití. Lze tak kdykoliv dotisknout další díly pro rozšiřující senzory, nebo uzpůsobit montáž celé krabičky. Technologie tisku, kterou používá tiskárna má jistá omezení. Proto musela být krabička tisknuta po dílech, což se nakonec ukázalo velkou výhodou. Pro symetrické působení proudícího vzduchu je krabička kulatá. Tato volba s sebou nese jisté komplikace. Zejména je tím tvar PCB řídicí elektroniky, který musel vyhovovat vnitřním rozměrům krabičky.

Design je vytvořený v openscadu, což je 3D modelovací (CAD) program, kde se design objektu vytváří psaním kódu se syntaxí podobnou jazyku C. Celý design je vytvořený s ohledem na modulárnost. Není tedy problém změnit jakékoliv parametry krabičky. Spojení všech dílů zajišťují ocelové dráty, na které jsou jednotlivá patra nasunuta. Je tedy možné kdykoliv doplnit patra pro další senzory, nebo například vyměnit vrchní díl za jiný s otvorem pro měření intenzity osvětlení.

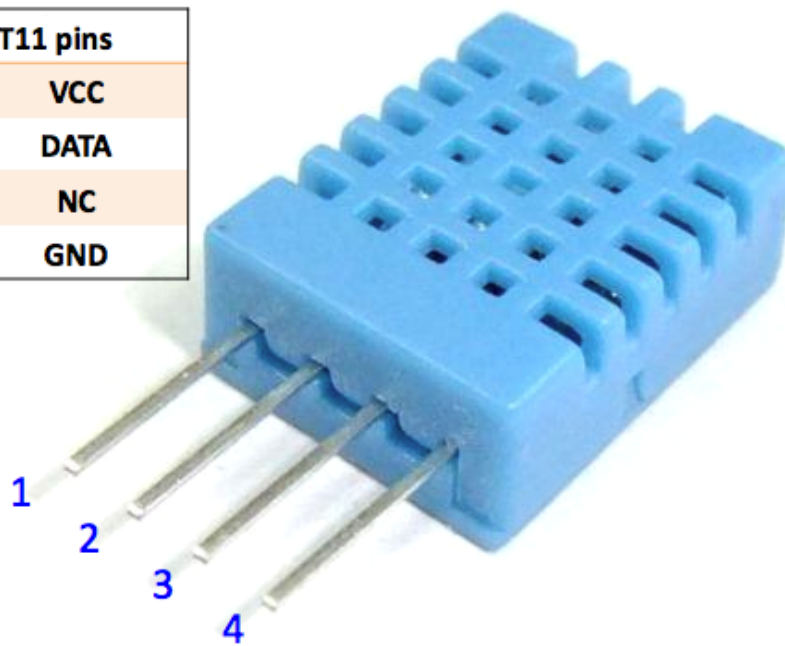


Obrázek 3.1: Výsledné schéma

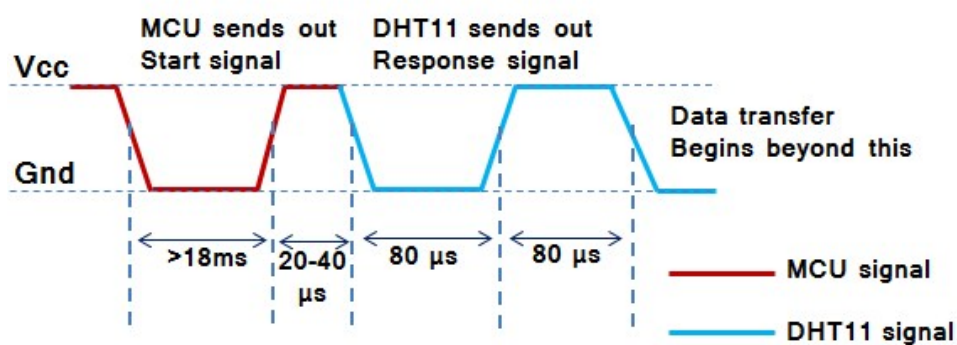


Obrázek 3.2: testování i2c komunikace na nepájivém poli

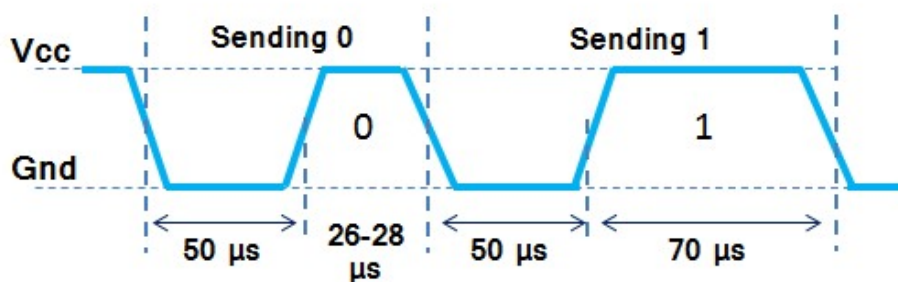
DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



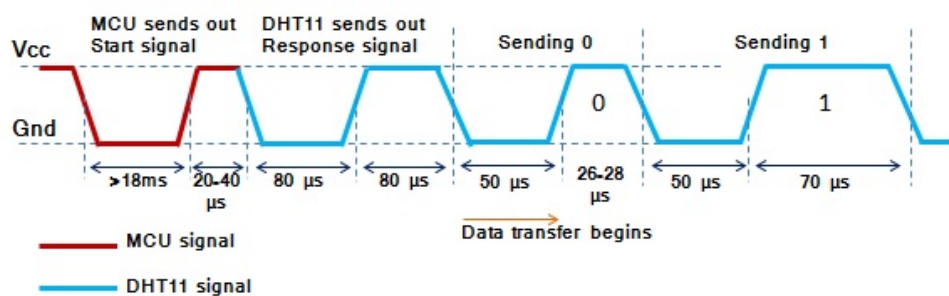
Obrázek 3.3: Pouzdro a vývody senzoru



Obrázek 3.4: Uvození komunikace se senzorem



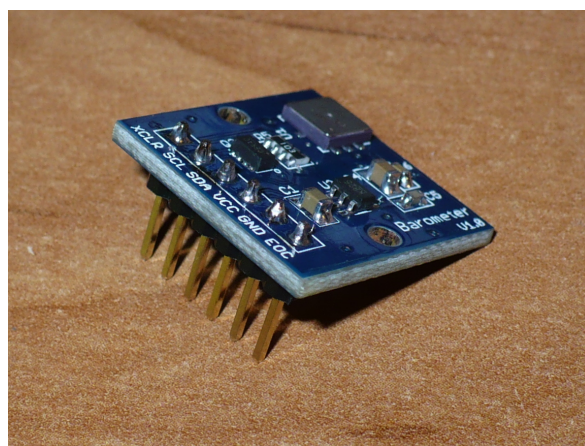
Obrázek 3.5: Rozlišení logické nuly a jedničky



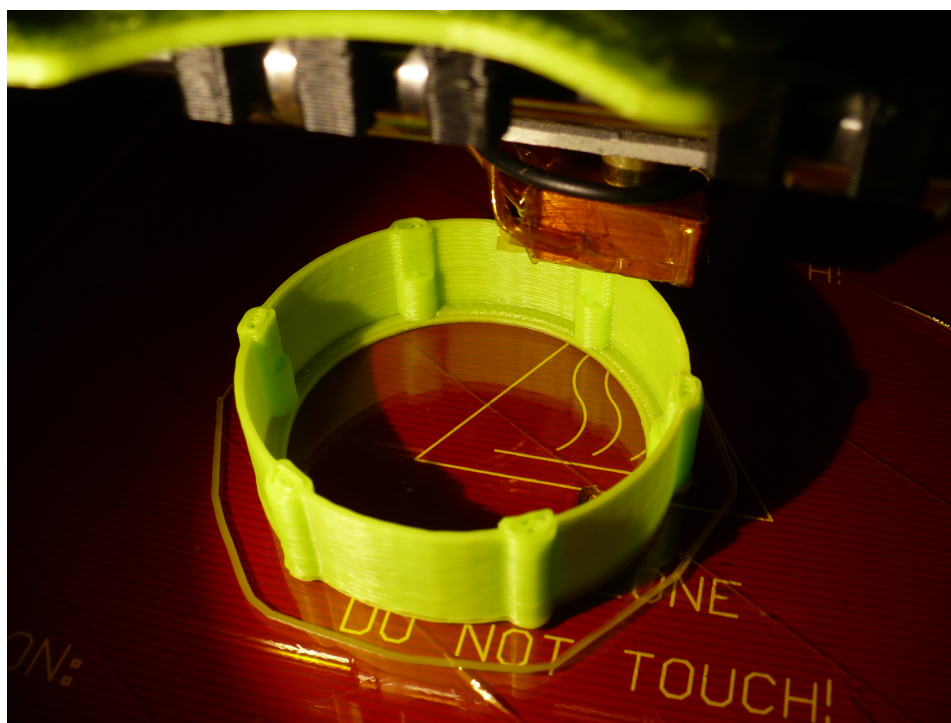
Obrázek 3.6: Start a přenos dat



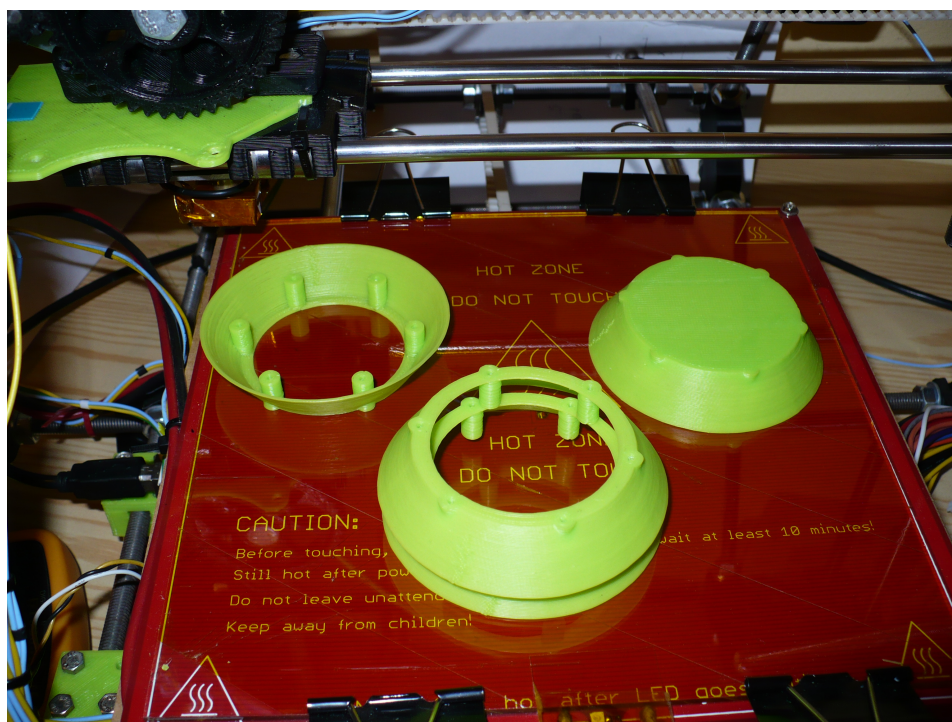
Obrázek 3.7: Senzor BMP085



Obrázek 3.8: Modul se senzorem



Obrázek 3.9: Tisk krabičky

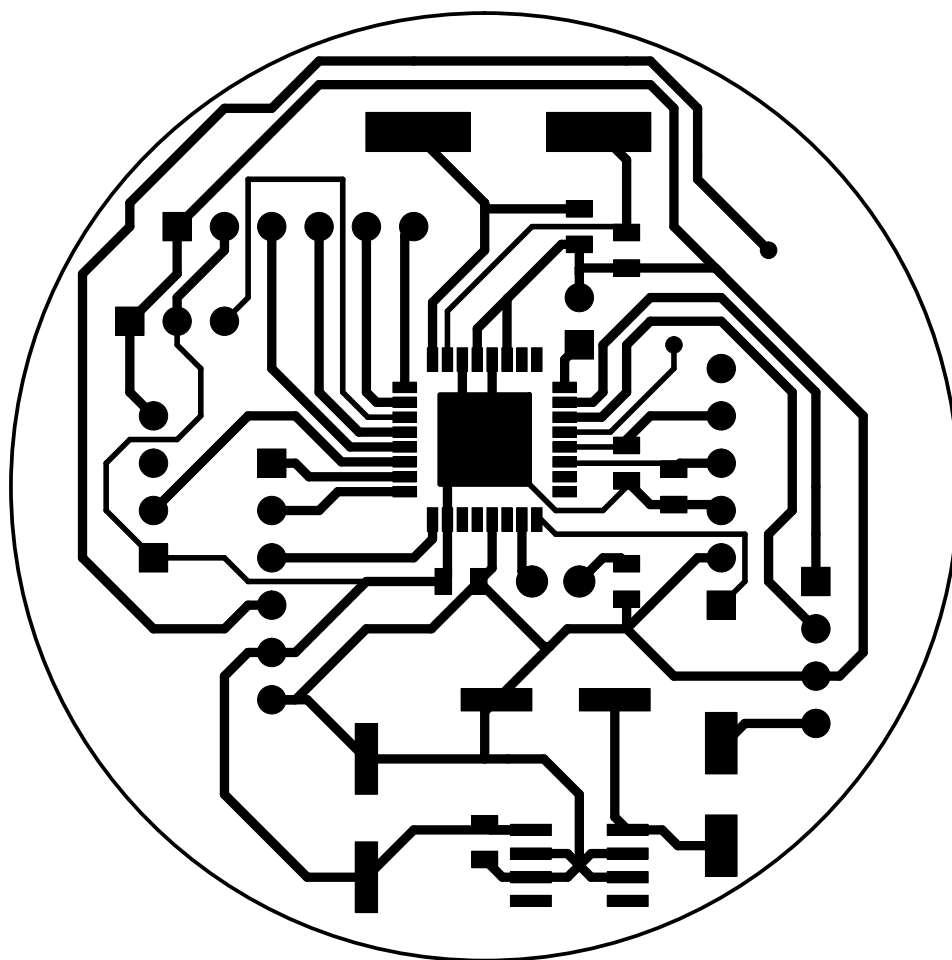


Obrázek 3.10: Přehled dílů

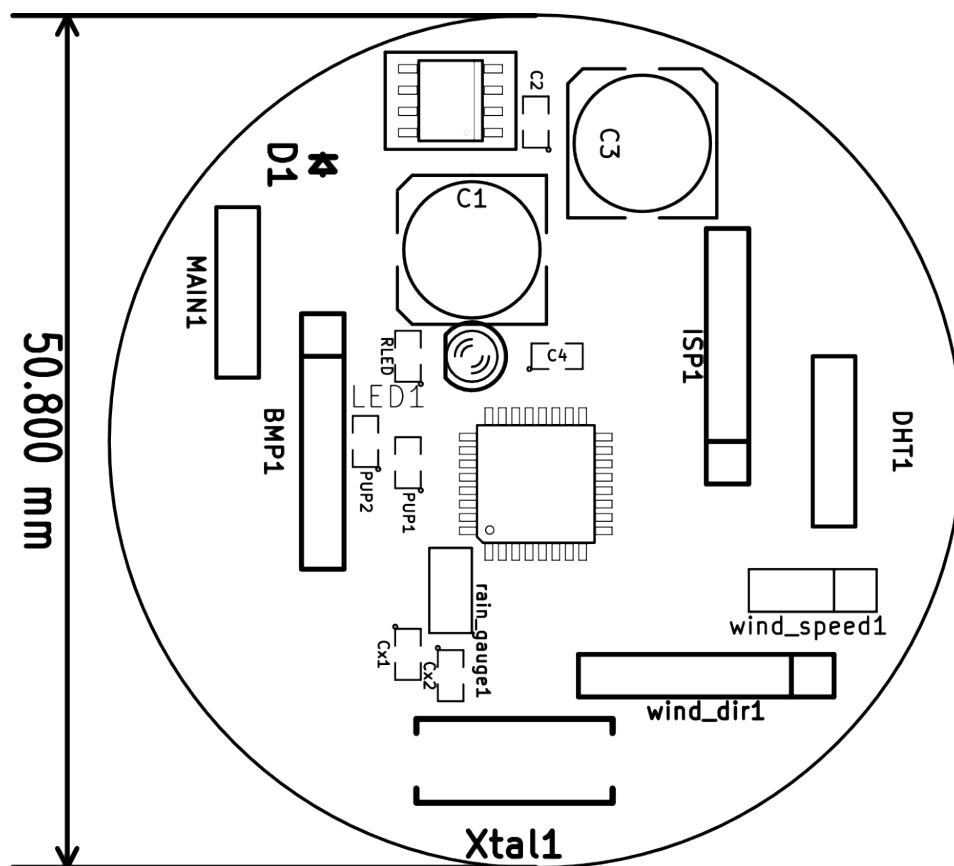
4 Stavba

4.1 Výroba PCB

Jelikož jsem počítal s výrobou desky plošných spojů ve škole, navrhl jsem pouze jednostrannou desku. Kvůli omezenému prostoru uvnitř krabičky meteostanice jsem volil co největší množství součástek v SMD pouzdře.



Obrázek 4.1: plošný spoj



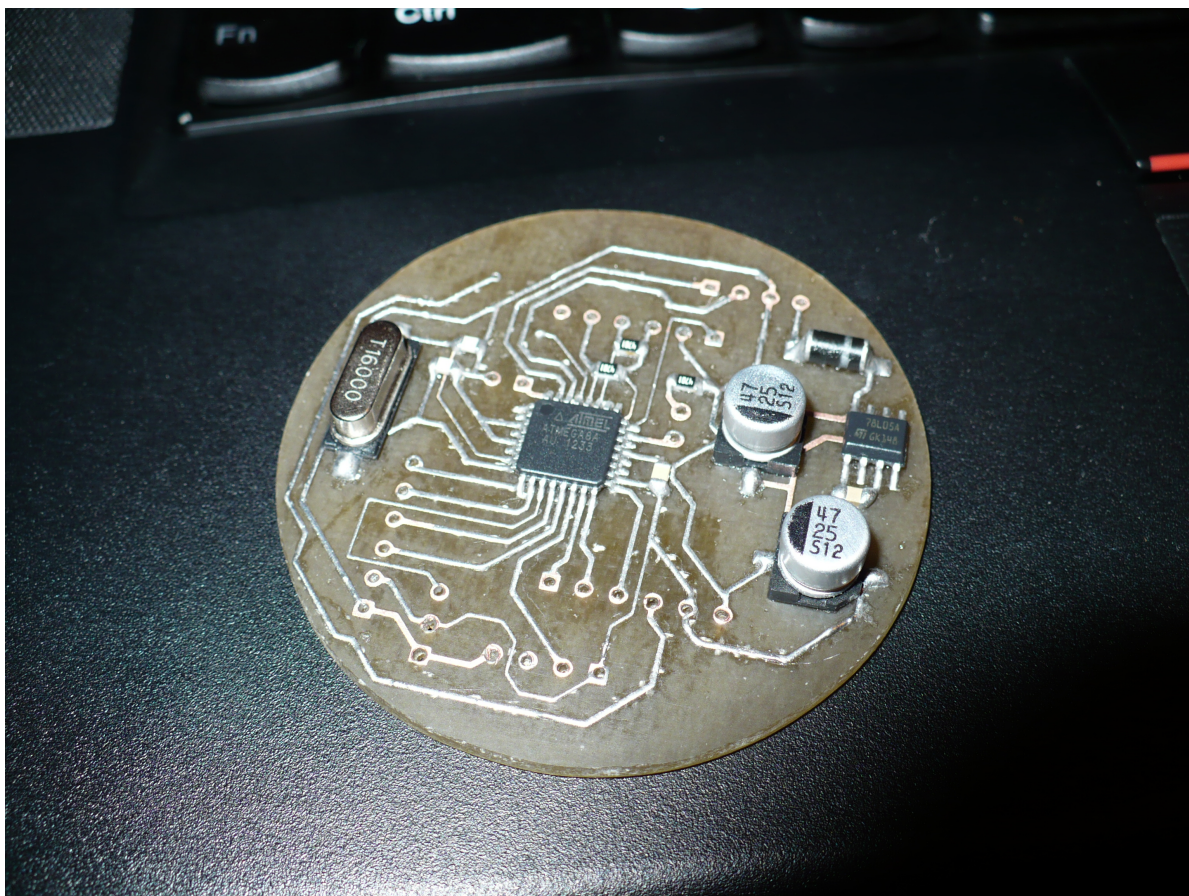
Obrázek 4.2: rozmístění součástek

4.2 osazení PCB

Desku plošných spojů jsem očistil technickým lihem od nečistot. Nejdřív jsem osadil mikrokontrolér, který má pouzdro s největším počtem vývodů. Potom jsem osadil pull-up rezistory na i2c sběrnici a rezistor k indikační LED. Poté jsem osadil blokovací kondenzátor na napájení mikrokontroléru. Následoval stabilizátor s vyhlazovacími kondenzátory a diodou pro ochranu proti přepólování napájení. Jako poslední jsem osadil krystal s kondenzátory sloužící jako zdroj hodinového signálu pro mikrokontrolér. Nakonec jsem osadil dutinkové lišty, které se částečně zasunuly do vyvrtaných děr ze strany spojů a neprocházely tak skrz desku.

4.3 tepelná izolace elektroniky

Každá součástka, kterou protéká elektrický proud produkuje teplo. I když samotný modul meteostanice obsahuje minimální množství komponent,



Obrázek 4.3: Osazování desky řídicí elektroniky, uprostřed mikrokontrolér

stále produkuje malé množství tepla. Abych předešel ovlivňování měření tímto odpadním teplem z elektroniky, vložil jsem mezi PCB a senzory několik vrstev tepelně izolačních materiálů.

Jako první, nejtlustší vrstva je použita pěna již vytvarovaná do plátu, který jsem ořízl do kulatého tvaru odpovídajícímu vnitřním rozměrům krabičky. Druhá izolační vrstva je tvořena kaptonovou páskou, což je páska z materiálu na bázi polyamidu, odolávající velkým teplotním rozdílům a širokému spektru záření. Pro meteostanici má největší přínos její tepelně izolační schopnost. Páska také částečně brání případnému průniku vody k řídicí elektronice. Poslední vrstva je polystyren, který má také velmi dobré izolační vlastnosti.

Všemi vrstvami prochází dva otvory pro konektory senzorů. Samotné senzory jsou pak tepelně izolované od elektroniky a zároveň vystaveny přímému působení okolních jevů.



Obrázek 4.4: Tepelná izolace mezi elektronikou a senzory

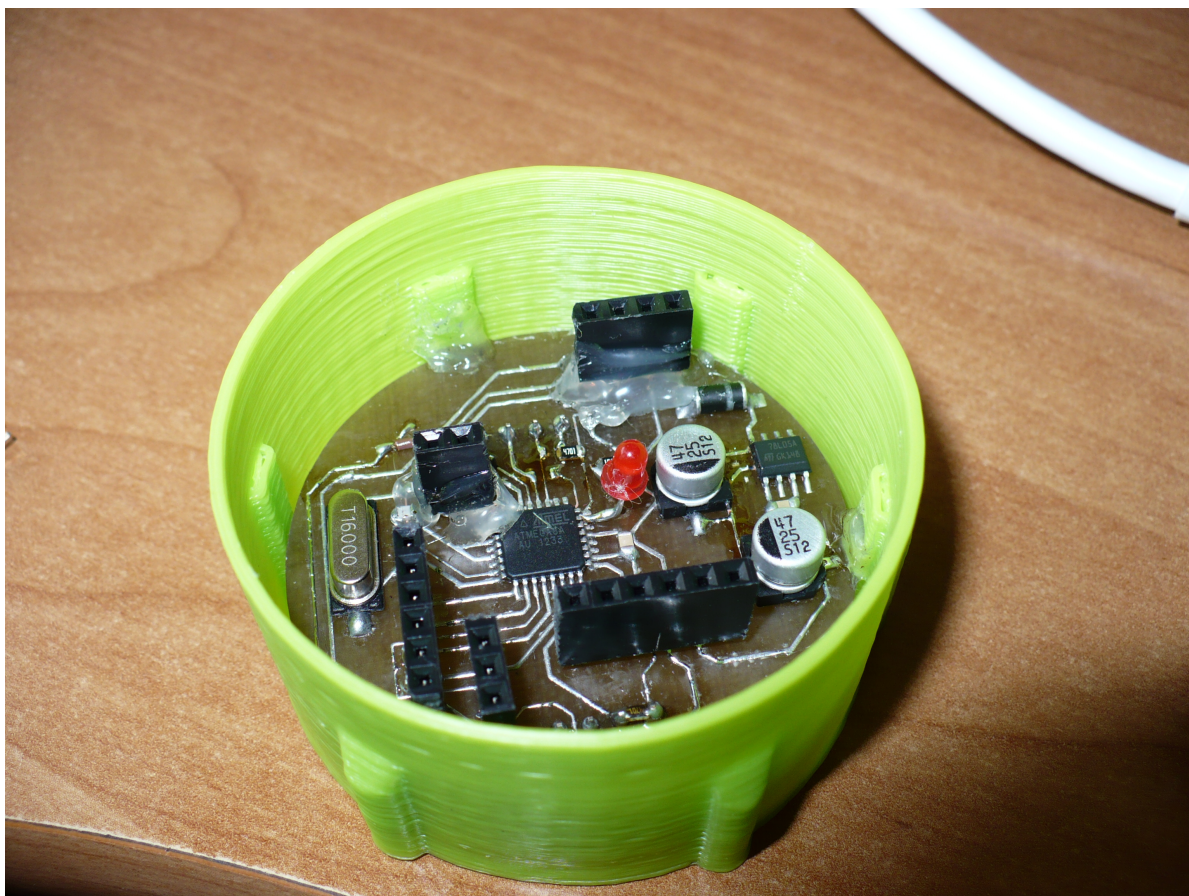
4.4 umístění v krabičce

Deska plošných spojů elektroniky má kulatý tvar, který přesně zapadá do vnitřního prostoru krabičky meteostanice. Uvnitř je zalepena tavným lepidlem pro případné vyjmutí bez poškození krabičky.

4.5 Odstranění chyb

Při návrhu desky plošných spojů jsem si neuvědomil, že dutinkové lišty budou umístěné ze stejné strany jako měděné cestičky. Jelikož tyto lišty držely pouze na nich, hrozilo poškození cestiček. Abych předešel těmto problémům, připevnil jsem dutinkové lišty k desce tavným lepidlem. Takto zafixované již netahají při mechanickém namáhání přímo za měď.

Při návrhu desky plošných spojů jsem automaticky umístil na i2c sběrnici pull-up rezistory. Při prvním testování se mi nedařilo navázat komunikaci se senzorem. Po proměření jsem zjistil, že odpory jsou nejspíš již na desce



Obrázek 4.5: Řídící elektronika zasazená do krabičky

senzoru a mikrokontrolér tak nebyl schopen stáhnout napětí na datovém ani hodinovém vodiči k nule, což je startovací sekvence na i2c sběrnici. Po odstranění rezistorů z desky řídicí elektroniky již bylo možné komunikovat se senzorem bez problémů.

5 Server

Webový server Nginx běží na minipočítači RaspberryPi. Vyhnul jsem se často využívanému serveru Apache kvůli funkcím, které ani zdaleka nevyužiji a zbytečně by zabíral místo na už tak malém úložném prostoru. Jelikož jsou stránky meteostanice statické, nemusí server na každý dotaz generovat stránku znovu, ale pošle přímo soubor webové stránky.



Obrázek 5.1: Raspberry Pi sloužící jako webový server

5.1 Získávání dat z meteostanice

Na RaspberryPi běží v pozadí cron, což je daemon, který podle nastavení spouští programy v určené časy.

Aktuální nastavení:

```
*/10 * * * * /home/pi/weather_station/run.sh  
1-9,11-19,21-29,31-39,41-49,51-59 * * * * python3 \  
/home/pi/weather_station/grabber_daemon.py jsonly
```

První řádek určuje, že bude každých 10 minut spuštěn script run.sh . Ten dále spouští script v pythonu, který komunikuje s meteostanicí. Výstup je směrován do souborů logu. Po proběhnutí script run.sh zkontroluje složku s log soubory a smaže více jak den staré. Získané hodnoty jsou uloženy do databáze a jsou překresleny grafy.

Druhý řádek nastavuje spuštění scriptu komunikujícího s meteostanicí. Spouští se každou minutu vyjma celých desítek. Slouží pouze pro aktualizaci hodnot na stránce, hodnoty nejsou uloženy do databáze.

5.2 Úložiště dat

Pro dlouhodobé uložení dat a snadný přístup k nim, jsem vybral mysql databázi. Lze k ní přistupovat přímo ze scriptu v pythonu nebo z php. Lze v ní přesně nastavit datové typy pro jednotlivé hodnoty, pak není využíváno nadbytečné místo v paměti. Zároveň je získávání i zápis dat do databáze velmi rychlý. Pomocí jednoduchých dotazů lze zpětně dohledávat maxima či minima změřených hodnot, mazat staré výsledky nebo zanechávat jen ty něčím význačné.

5.3 Zpracování dat

Samotné získávání dat z meteostanice provádí script v pythonu nazvaný grabber_daemon.py Ten podle zadaných parametrů může přistupovat do databáze, generovat grafy, získávat data z meteostanice nebo aktualizovat hodnoty na webové stránce. Bez parametrů si script vyžádá data od meteostanice, uloží je do databáze, vyžádá si posledních N výsledků a z nich vytvoří nové grafy.

Výpis nápovědy:

```
./grabber_daemon.py help  
Welcome sky people!,
```


no date, but will be recorded -----^^

Pokud je nalezena chyba v návaznosti, je automaticky přidán datum s přesným časem do grafu k příslušné hodnotě. Zároveň je vyznačen tento špatný časový interval v grafech červeným svislým pruhem.

5.3.2 Četnost měření

Nyní mám je v cronu nastaveno, aby se z meteostanice každých 10 minut získaly hodnoty a došlo k jejich uložení a překreslení grafů. Každou minutu jsou navíc aktualizovány hodnoty na webové stránce v hlavičce.

Obě tato nastavení lze změnit přepsáním příslušných pravidel v cronu. Je potom nutné nastavit stejnou hodnotu ve scriptu jenž získává z meteostanice hodnoty a obstarává další úkony. Jinak by nedocházelo ke správné detekci časových kroků vzorků a mohly by být všechny rozpoznány jako chybné. Je podmínkou aby byl interval celočíselným dělitelem 60 minut.

5.4 webová aplikace

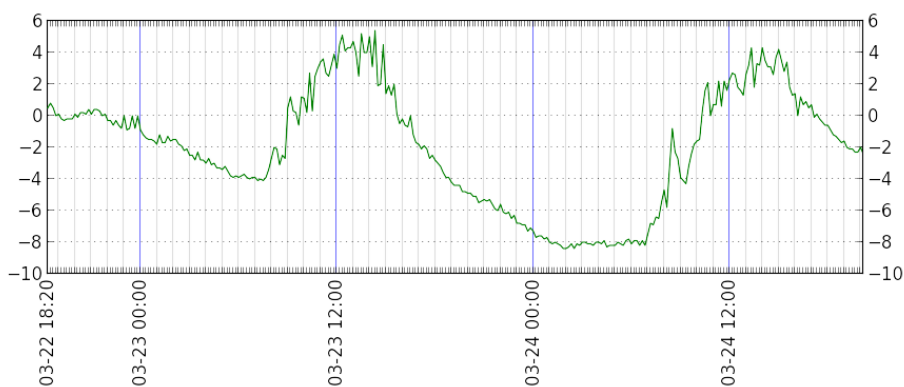
Celá webová aplikace se skládá z několika stránek napsaných v HTML, CSS a JavaScriptu. HTML stránka zůstává statická a o dynamické změny se stará kód v JavaScriptu. Ten si při každém novém načtení stránky stáhne soubor s proměnnými ve kterých jsou uloženy hodnoty z posledního měření a vypíše je do hlavičky stránky.

-2.9 °C 1021 hPa 48 %

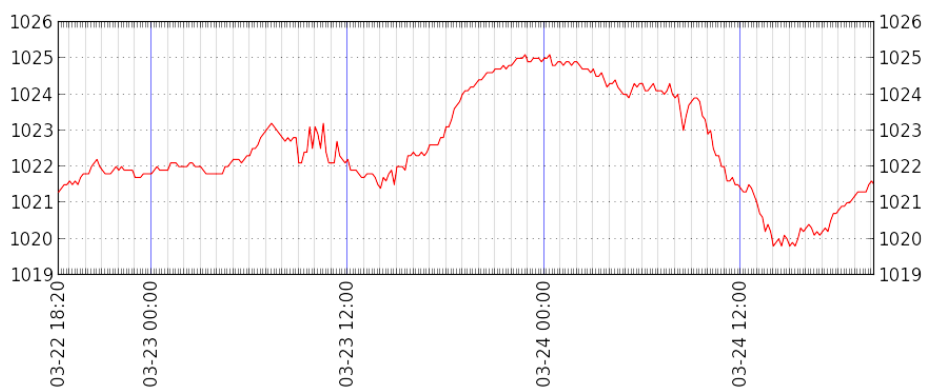
24.03. 2013 20:16

[Hlavní strana](#) [O projektu](#) [Kontakt](#)

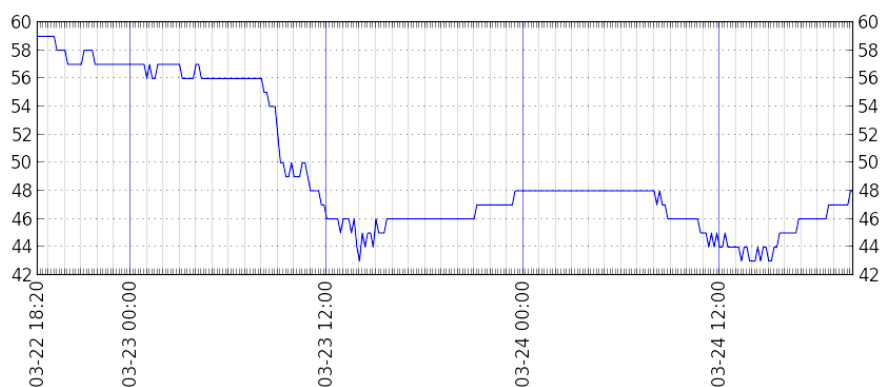
Teplota posledních dnů v °C



Tlak v posledních dnech v hPa



Relativní vlhkost v posledních dnech v %



Data z meteorologické stanice jsou stahována každou minutu a zobrazována v hlavičce. Grafy jsou aktualizovány každých 10 minut. Červeně označené oblasti indikují vynechaná měření.

By Patrik Bachan, CONTACT, opensource project

Obrázek 5.2: Webová stránka meteorologické stanice
25

6 Závěr

Vyrobená meteorologická stanice měří již několik týdnů. Během tohoto období jsem zaznamenal několik výpadků. Zatím z neznámé příčiny se stalo RaspberryPi nedostupné přes síť, přestalo spouštět script na získávání dat a docházelo k masivnímu přístupu k úložnému prostoru. Jelikož tento výpadek nastal pouze jednou, nesnažil jsem se zatím najít chybu. Jiné výpadky byly způsobeny problémy s napájením, kde došlo k rozpojení přírodních vodičů.

Při měření dochází občas ke špatnému změření teploty, kdy meteostanice vrátí nulovou hodnotu i v případě, je-li venku odlišná teplota. Tuto chybu se mi nepodařilo zcela eliminovat. Omezil jsem ji přidáním časovou prodlevou, kdy čeká mikrokontrolér na změření hodnot od senzoru. V blízké době počítám s přidáním samostatného teplotního čidla, od kterého si slibuji větší spolehlivost. Měření tlaku i relativní vlhkosti probíhá bez problémů. Po odladění těchto problémů mám v plánu kontaktovat naše bytové družstvo, zda-li by měli zájem umístit stanici na střechu domu. Pokud bude tento návrh přijat, plánuji zhotovit ultrazvukový snímač směru a rychlosti větru na principu měření rychlosti zvuku. Dále bych chtěl publikovat článek o této meteostanici na webu oficiálním webu raspberrypi.org.

6.1 Použitý software

Vývojové prostředí Geany
avrdude pro nahrávání programů do mikrokontroléru
AVR GNU toolchain pro kompilaci kódu pro mikrokontrolér

Distribuce linuxu Raspbian
Běžové prostředí python3 a knihovny pro obslužný script stanice.
Webový server Nginx
SQL databáze MySQL

\LaTeX sazeč textu
TexWorks frontend pro \LaTeX

Literatura

- [1] *Katalogový list mikrokontroléru*
http://www.atmel.com/Images/Atmel-2486-8-bit-AVR-microcontroller-ATmega8-L_datasheet.pdf
- [2] *Katalogový list senzoru DHT11*
<http://www.micro4you.com/files/sensor/DHT11.pdf>
- [3] *Katalogový list senzoru BMP085*
<http://www.daedalus.ei.tum.de/attachments/article/60/BST-BMP085-DS000-05%20Drucksensor.pdf>
- [4] *Dokumentace protokolu senzoru DHT11*
<http://embedded-lab.com/blog/?p=4333>
- [5] *Web knihovny OurSQL pro přístup k mysql databázi z pythonu*
<https://launchpad.net/oursql>