

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra elektromechaniky a výkonové elektroniky

BAKALÁŘSKÁ PRÁCE

Bezdrátový teploměr komunikující s PC

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin JURÍK**
Osobní číslo: **E12B0219P**
Studijní program: **B2644 Aplikovaná elektrotechnika**
Studijní obor: **Aplikovaná elektrotechnika**
Název tématu: **Bezdrátový teploměr komunikující s PC**
Zadávající katedra: **Katedra elektromechaniky a výkonové elektroniky**

Zásady pro vypracování:

1. Popište základní možnosti řešení bezdrátového teploměru, výtiskujte vhodné teplotní čidlo a způsob bezdrátové komunikace.
2. Navrhněte obvodové zapojení a zrealizujte jej.
3. Vytvořte obslužný SW do PC.



Rozsah grafických prací: podle doporučení vedoucího

Rozsah pracovní zprávy: 20 - 30 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce:

Ing. Jiří Žahour

Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **15. října 2014**

Termín odevzdání bakalářské práce: **8. června 2015**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Prof. Ing. Václav Kůs, CSc.
vedoucí katedry

V Plzni dne 15. října 2014

Abstrakt

Předkládaná bakalářská práce je zaměřena na měření teploty a její následné bezdrátové odeslání na server. Na serveru je následně zpracována, uložena a zobrazena na internetovou stránku. V první části práce jsou popsány různé možnosti řešení celého projektu. Ve druhé kapitole je detailně vysvětlena každá aplikovaná možnost. Závěrem jsou rozebrány další možné vylepšení celého projektu.

Klíčová slova

Měření teploty, bezdrátová komunikace, ATmega88PV, Nrf24l01+, server, Java, JSON, MySQL

Abstract

This bachelor thesis is focused on measuring temperature. The measured temperature is wirelessly sent to server, where it is saved into a database. User can read measured values at website. In the first part are described possible opportunities of whole project. In the second part are described and explained methods which were used in developed device. At the end is a list of possible improvements.

Key words

Measuring temperature, wireless communication, ATmega88PV, Nrf24l01+, server, Java, JSON, MySQL

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 5.6.2015

Martin Juřík

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Jiřímu Žahourovi za cenné profesionální rady, připomínky a metodické vedení práce.

Dále bych rád poděkoval lidem, kterých si velmi vážím. Jsou jimi Vojtěch Kadlec a Václav Král. Patří jim mé poděkování především za předané zkušenosti, rady a věcné připomínky při řešení projektu.

Obsah

| | |
|---|-----------|
| OBSAH | 8 |
| ÚVOD | 10 |
| 1 TEORETICKÝ ROZBOR | 11 |
| 1.1 CELKOVÝ KONCEPT | 11 |
| 1.2 VOLBA MIKROKONTROLÉRU | 11 |
| 1.3 VÝBĚR TEPLOTNÍHO ČIDLA | 13 |
| 1.4 VÝBĚR KOMUNIKAČNÍHO MODULU | 14 |
| 1.5 VÝBĚR BATERIE..... | 15 |
| 1.6 SERVER..... | 16 |
| 1.6.1 Zařízení přijímající data | 16 |
| 1.6.2 Obslužný program..... | 19 |
| 1.6.3 Internetové stránky | 20 |
| 2 PRAKTICKÁ REALIZACE | 23 |
| 2.1 ZAŘÍZENÍ REALIZUJÍCÍ MĚŘENÍ TEPLoty | 23 |
| 2.1.1 Mikrokontrolér a program | 24 |
| 2.1.2 Teplotní čidlo | 30 |
| 2.1.3 Komunikační modul | 32 |
| 2.1.4 Baterie..... | 33 |
| 2.1.5 Celková spotřeba energie | 33 |
| 2.2 SERVER..... | 37 |
| 2.2.1 Serverový přijímač | 38 |
| 2.2.2 Ovládací program..... | 39 |
| 2.2.3 Internetové stránky | 43 |
| 2.3 KRABÍČKA | 46 |
| 2.4 APLIKACE PRO ANDROID | 48 |
| 3 DALŠÍ MOŽNÉ VYLEPŠENÍ | 48 |
| 3.1 MĚŘENÍ TLAKU | 48 |
| 3.2 WINDOWS APLIKACE | 49 |
| 3.3 DOMÁCÍ STANICE ZOBRAZUJÍCÍ TEPLotu A DENNÍ TEPLotní EXTRÉMY | 49 |
| 3.4 GENEROVÁNÍ POLYNOMU | 49 |
| 4 ZÁVĚR | 50 |
| PŘÍLOHY | I |

Seznam obrázků

| | |
|---|----|
| OBRÁZEK 1: ZJEDNODUŠENÉ ZNÁZORNĚNÍ CELÉHO KONCEPTU | 11 |
| OBRÁZEK 2: SPOTŘEBA MIKROKONTROLÉRU ATMEGA88PV | 34 |
| OBRÁZEK 3: SPOTŘEBA KOMUNIKAČNÍHO MODULU NRF24L01+ | 34 |
| OBRÁZEK 4: SPOTŘEBA TEPLOTNÍHO ČIDLA SI7053 | 35 |
| OBRÁZEK 5: ZNÁZORNĚNÍ ČASOVÉ SOUSLEDNOSTI ZAPÍNÁNÍ JEDNOTLIVÝCH PRVKŮ. | 37 |
| OBRÁZEK 6: SPOTŘEBA CELÉHO ZAŘÍZENÍ | 37 |
| OBRÁZEK 7: NOVÉ TAGY V HTML5 | 43 |
| OBRÁZEK 8: LEGENDA Z INTERNETOVÝCH STRÁNEK - ZNÁZORNĚNÍ POUŽITÍ TAGU CANVAS. | 44 |
| OBRÁZEK 9: FAVICON A TITULEK STRÁNKY | 45 |
| OBRÁZEK 10: ZOBRAZENÍ GRAFU. | 45 |
| OBRÁZEK 11: SCHÉMA ZAŘÍZENÍ MĚŘÍCÍ TEPLOTU. | I |
| OBRÁZEK 12: SCHÉMA SERVEROVÉHO PŘIJÍMAČE. | II |

Úvod

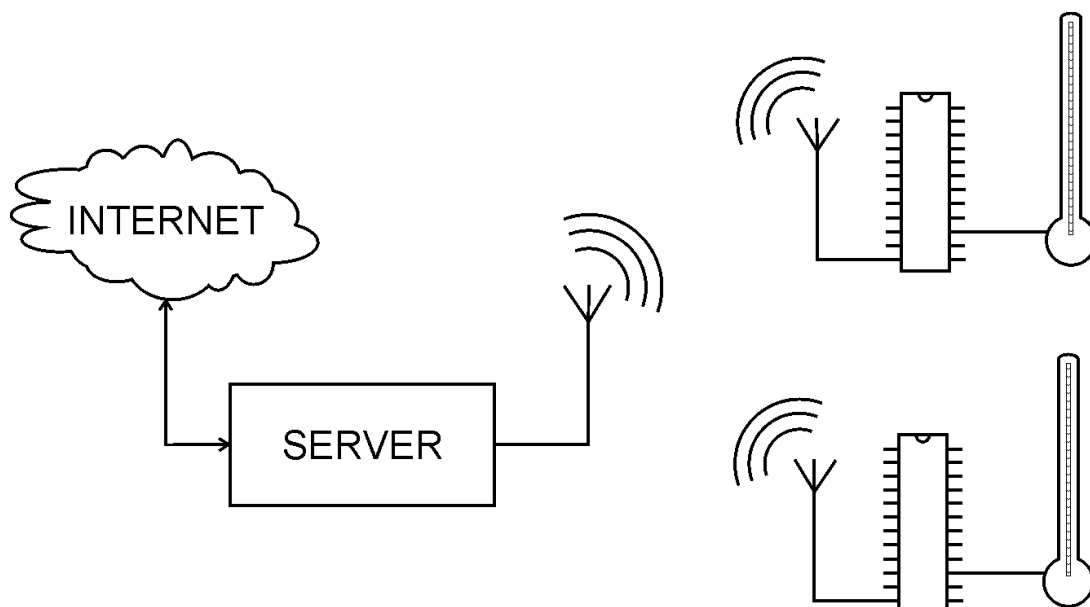
Práce se zabývá bezdrátovým sledováním teploty. Snahou není pouze zobrazovat aktuální teplotu, ale také teplotu monitorovat dlouhodoběji. Uživateli jsou následně data zobrazena pomocí internetových stránek a aplikace do mobilních telefonů.

V práci byly rozebrány základní možné součástky pro návrh jak zařízení měřící teplotu, tak zařízení přijímající data. Snahou bylo kompletní text rozdělit do logických a ucelených bloků pro snadnější orientaci v textu. První část se zabývá obecným popisem problematiky a způsoby řešení. V druhé části je detailně popsána zhotovená realizace a v poslední části jsou navržena další možné vylepšení.

1 Teoretický rozbor

1.1 Celkový koncept

Vysílač teploty, umístěný v dosahu přijímače (serveru), několikrát za minutu vyšle data s aktuální teplotou. Na straně serveru bude zařízení připojené do USB, které data přijme a pouze je přepoše přes USB do serveru. V serveru je program, který přijatá data zpracuje a uloží do databáze. Uživatel si může tuto databázi procházet na internetových stránkách a prohlížet si naměřená data z minulosti v podobě grafů. *Obrázek 1* znázorňuje zjednodušené schéma celého systému. Pro projekt byl vytvořen název **Meteobug**.



Obrázek 1: Zjednodušené znázornění celého konceptu

1.2 Volba mikrokontroléru

Mikrokontrolér je základním prvkem celého zařízení. Musí zajistit změření teploty a následný bezchybný přenos dat k serveru. Důležitým úkolem je také relativně přesné načasování následnosti úkonů. Pokud by nebyla zajištěna souslednost úkonů, mohly by některé součástky být aktivní i v době, kdy pro ně nejsou připravena vstupní data. Většinu úkonů, které musí mikrokontrolér vykonat, je možné vytvořit pomocí programu. Jsou ovšem části, které ovlivnit již nelze. V této práci je kladen velký důraz na spotřebu a ta je již dána výrobou a architekturou. U některých mikrokontrolérů je možné využít dodatečných funkcí,

na které se daný výrobce specializuje. Jsou to například funkce, které, i v době spánku mikrokontroléru, umí přesouvat data v paměti nebo například velmi značné snížení spotřeby snížením frekvence při samotném běhu. Je obvyklé, že výrobci, zabývající se nízkoodběrovými mikrokontroléry, mají na svých internetových stránkách uvedeny postupy jak dosáhnout co nejnižší spotřeby.

Je vhodné, aby vybraný mikrokontrolér disponoval základními sběrnicemi, jako je SPI, I²C či UART pro usnadnění komunikace se součástkami zajišťujícími měření teploty a bezdrátový přenos dat. Není to nutnou podmínkou, protože vše lze implementovat do programu. Má to ovšem nevýhodu v delší době vykonávání programu a díky tomu mnohem vyšší spotřebě energie. Tento problém není ovšem nutné řešit, protože těmito základními periferiemi disponují všechny moderní mikrokontroléry.

Po prostudování současných možností trhu bylo zjištěno, že většina společností je na tom se spotřebou velmi podobně. Na základě tohoto faktu bylo rozhodnuto, že je lepší prohloubit zkušenosti s mikrokontroléry od společnosti, se kterou již byly získány zkušenosti na předchozích projektech než se od základu naučit s mikrokontroléry od společnosti jiné a dosáhnout stejného výsledku. Díky tomu mohou být získané znalosti ohledně úspory energie snadno aplikovány na projekty již dokončené a na projekty plánované do budoucna. Vybranou společností je Atmel. Hlavními faktory pro výběr byla spotřeba a zkušenosti s danými mikrokontroléry. Jelikož má ale tato společnost velmi široké portfolio výrobků, bylo nutné vybrat ten nejvhodnější. Každý z uvažovaných je zástupcem jedné kategorie mikrokontrolérů vyráběné touto společností. Zde je výčet prvků, které je možné použít:

- ATtiny43U
- ATmega88PV
- ATXmega16A4
- SAM¹ C21E15A

Mikrokontroléry jsou v seznamu seřazeny podle náročnosti aplikace, do které by měly být nasazeny, od nejnižší po nejvyšší. První tři zmíněné mají 8mi bitovou vnitřní sběrnici, zatímco poslední disponuje 32 bitovou sběrnicí. Všechny zmíněné mikrokontroléry obsahují veškeré základní komunikační sběrnice.

¹ SAM = Smart ARM-based MCUs

Prvními dvěma zmíněnými jsou mikrokontroléry určené pro jednodušší aplikace. I náročnost programování je nižší. Důvodem jsou jasně dané cesty, které Atmel vytvořil a těžko se modifikují. Je to například vidět na spánkových režimech. Bylo vytvořeno několik spánkových režimů, kde každý má jasně dané možnosti. Pokud je požadován neustálý běh velmi specifické periferie, může nastat situace, kdy není možné použít režimu spánku, protože by daná periferie byla zakázána. Pokud je ovšem jeden ze spánkových režimů přesně tím požadovaným, tak je programování velmi usnadněno.

Třetí mikrokontrolér je z řady zvané „Xmega“, tedy vylepšená verze základní řady „mega“. Jsou to 8mi bitové mikrokontroléry s velkým množstvím periférií pro velmi složité činnosti (velmi rychlý AD převodník, jednotka AES pro šifrování apod.). Celá tato řada byla vytvořena jako jakýsi mezikrok mezi klasickými 8mi bitovými a 32 bitovými mikrokontroléry. Ovšem ani u této řady nejsou zanedbány režimy spánku a nízká spotřeba v době nečinnosti.

Poslední je mikrokontrolér s jádrem ARM a šířkou sběrnice 32 bitů. Je absolutní novinkou uvedenou na trh v květnu 2015 od společnosti Atmel. Disponuje velkým množstvím pokročilých funkcí v komunikaci a zároveň ve spánkových režimech. Do takto nových mikrokontrolérů již vývojáři firmy Atmel nenavrhují periferie pro jednotlivé sběrnice, ale moduly zvané Sercom, které lze nastavit na jakoukoliv běžnou sběrnici. Byly navrženy rovnou 4 tyto moduly. Spánkové režimy se programují odlišným způsobem než u prvních tří zmíněných mikrokontrolérů. U tohoto typu je možné povolit několik desítek nastavení spojených se snižováním spotřeby. Jsou jimi například zákaz periférií, povolení hodinových signálů různých vnitřních sběrnic atd.

Jelikož poslední dva mikrokontroléry jsou připraveny na takto složité operace, odpovídá tomu i jejich zvýšená cena. Na základě tohoto faktu o nich již dále nebylo uvažováno a byly brány v potaz pouze první dva.

1.3 Výběr teplotního čidla

Prvotním požadavkem na teplotní čidlo byla pouze spotřeba, protože byla předpokládána relativně vysoká přesnost čidel. Přesnost je akceptovatelná do $\pm 1^\circ\text{C}$, protože je zbytečné mít

informaci o aktuální venkovní teplotě, která je příliš zkreslená nepřesností čidla. Bohužel bylo zjištěno, že některá používaná nízkoodběrová čidla tento parametr nejsou schopna splnit.

Tab. 1: soupis teplotních čidel

| | TMP101 | SHT25 | Si705X | AD7415 |
|--------------------------|-------------------------|---------------------------|----------------------------|---------------------------|
| Odběr ve spánku | 100nA | 150 nA | 60 nA | 800 nA |
| Odběr při měření | 45 μ A | 300 μ A | 90 μ A | 1,2 mA |
| Doba měření ² | 320 ms | 17 ms | 2,4 ms | 29 μ s |
| Přesnost ³ | $\pm 2^{\circ}\text{C}$ | $\pm 0,7^{\circ}\text{C}$ | $\pm 0, X^{\circ}\text{C}$ | $\pm 1,5^{\circ}\text{C}$ |

V tabulce zmíněné čidlo Si705X se vyrábí v několika variantách, kde X reprezentuje přesnost čidla. Například varianta Si7053 má přesnost $0,3^{\circ}\text{C}$. Dostupné varianty jsou 7053, 7054, 7055 a 7050 pro přesnost $\pm 1^{\circ}\text{C}$.

Z tabulky je patrné, že jediná dvě čidla, vyhovující podmínce přesnosti, jsou SHT25 a Si705X. Pravděpodobně je to dáno tím, že obě čidla jsou výrobcem kalibrována. Čidlo Si705X je v letošním roce (2015) vydanou novinkou od společnosti Silicon Labs. To je možná i jedním z důvodů, proč má lepší parametry než výrobky konkurence. Senzory SHT25 umožňují měřit i vlhkost ovzduší. V rozhodování mezi těmito dvěma čidly mohla rozhodnout už jen cena a i v tomto parametru jsou čidla Si705X lepšími.

1.4 Výběr komunikačního modulu

Nejdůležitějším prvkem z hlediska spotřeby je komunikační modul, protože v okamžiku vysílání je v něm spotřebováno několikanásobně více energie než ve všech ostatních součástkách. Tento fakt byl empiricky ověřen a i s časovými průběhy je popsán v sekci 2.1.5 .

Tab. 2: Soupis komunikačních modulů

| | Mrf89xa | Nrf24l01+ | Mrf24j40 |
|-----------------------|-------------|-------------|-----------|
| Frekvence | 868 MHz | 2,4 GHz | 2,4 GHz |
| Spotřeba při vysílání | 25 mA | 11,3 mA | 23 mA |
| Spotřeba při příjmu | 3 mA | 13,5 mA | 19 mA |
| Spotřeba ve spánku | 0,1 μ A | 0,9 μ A | 2 μ A |

² Pro 12-ti bitové rozlišení

³ Maximální přesnost pro rozsah -20°C až $+ 70^{\circ}\text{C}$

Modul Mrf24j40 (v pravé části Tab. 2) má z vybíraných modulů nejnepříznivější parametry pro aplikaci s nízkým odběrem. Má ovšem velkou výhodu v kompatibilitě s velmi rozšířenými moduly ZigBee a MiWi protokolem. Je tedy možné ho snadno aplikovat na již vytvořenou síť z jiných modulů. Bohužel jeho vysoké energetické požadavky jsou pro účely této práce neakceptovatelné.

Dále bylo rozhodováno mezi Mrf89xa a Nrf24l01+. Energetické požadavky jsou ve výsledku velmi podobné, ale nejdůležitějším rozdílem je komunikační frekvence modulů. Obecně vzato je nižší frekvence výhodou, protože signál má lepší prostupnost překážkami. Byl ovšem zjištěn problém v tom, že většina bezdrátových modulů (vyjma zařízení pracujících na principu výměny optických signálů) v domácnosti komunikuje právě na frekvenci 868 MHz. Pokud je takových zařízení v domácnosti větší množství, tak je velmi pravděpodobné, že dané pásmo bude velmi zarušené a teplotní senzor se nebude moci spojit se serverem. Pásmo kolem frekvence 2,4 Ghz je velmi používané pro domácí WiFi sítě. Rušení komunikace je tedy také poměrně silné. Výhodu to má ovšem v tom, že každý WiFi přístupový bod ze samotného principu vytváří v komunikaci časové mezery, aby umožnil komunikovat ostatním přístupovým bodům na stejné frekvenci. V těchto okamžicích moduly Nrf24l01+ mohou uskutečnit svou komunikaci.

1.5 Výběr baterie

Mezi uvažované baterie patří: CR2032, Eneloop baterie od společnosti Panasonic a LS 14250 od společnosti Saft. Shrnutí základních parametrů je znázorněno v následující tabulce.

Tab. 3: Soupis baterií

| | CR2032 | Eneloop | LS 14250 |
|------------------|--------------------|--------------------|----------------------|
| Napětí [V] | 3 | 1,2 | 3,6 |
| Kapacita [mAh] | 225 | 750 | 1200 |
| Pouzdro | Knoflíková baterie | AAA | ½ AA |
| Cena v roce 2015 | 25 Kč ⁴ | 60 Kč ⁵ | 135 Kč ⁶ |
| Typ baterie | Lithiová | Ni-Mh | Li-SoCl ₂ |
| Samovybíjení | 1% za rok | 3% za rok | 1% za rok |

⁴<http://www.gme.cz/baterie-lithiova-knoflikova-gp-cr2032-3v-210mah-lithiova-p542-047>

⁵http://www.tme.eu/cz/details/accu-r3_800hr/akumulatory/panasonic/#t47d5b5f34f5ae3c9a76ac41dc4153d03

⁶<http://www.tme.eu/cz/details/saft-ls14250/baterie/saft/ls-14250/#t79ceee9b79e11ffc7df5ecfaa108552e>

Již z názvu baterie CR2032 lze poznat několik údajů. Písmena CR znamenají, že je baterie vyrobena z lithia. První dvojčíslí značí průměr baterie v milimetrech a druhé dvojčíslí výšku v desetinách milimetrů. V tomto případě má tedy baterie průměr 20mm a výšku 3,2mm. Výhodou oproti ostatním zmíněným je malá velikost a nízká cena. Nevýhodou je velmi nízká kapacita a s tím spojená jejich častá výměna. Další nevýhodou je napěťový pokles v závislosti na teplotě. Již při teplotě -10°C je pokles napětí (v závislosti na zátěži) pod kritickou hranici 1,9V. Pod touto hranicí již zařízení není schopné spolehlivě pracovat. Pro vnitřní použití se to nemusí brát v potaz, protože se nepředpokládá pokles teploty na tak nízkou úroveň. Hlavním cílem této práce je měřit teplotu venkovního prostředí a v zeměpisných šířkách střední Evropy je nutné počítat i s teplotami mnohem nižšími než -10°C .

Baterie Eneloop jsou velmi kvalitními bateriemi pro zařízení s nízkým odběrem. Jejich velkou výhodou je možnost opětovného nabití. Jejich nevýhodou je ovšem nižší napětí, protože každý článek má pouze 1,2V. Tím pádem by bylo nutné použít minimálně 2 kusy a tím by se razantně zvýšila velikost. O této možnosti by se dalo uvažovat v případě značného zvýšení kapacity na úkor prostoru. Bohužel kapacita by nebyla zvýšena z důvodu principu sériového spojení článků, kde se sice sčítají napětí, ale kapacita zůstane shodná jako kapacita jednoho článku. Tím pádem by se zařízení stalo velkým s nepříliš velkou kapacitou.

Poslední zmíněné baterie (LS 14250) mají výhodu oproti konkurenci ve velmi nízkém samovybíjení. Se samovybíjením 1% kapacity za rok je možné mít zajištěné napájení pro danou aplikaci na několik let. Další výhodou je dlouhodobá stabilita napětí. Velkou nevýhodou je cena a nemožnost nabití. Baterie této velikosti není příliš používána, avšak pro účely této práce je vhodná. Pro snazší představu, pouzdro má polovinu rozměru klasické tužkové baterie (AA).

1.6 Server

1.6.1 Zařízení přijímající data

Základními prvky vytvořeného zařízení jsou modul pro bezdrátové přijímání, mikrokontrolér dekódující data z přijímače a integrovaný obvod, který umožní mikrokontroléru komunikovat se serverem přes rozhraní USB.

Bezdrátový modul:

Jako bezdrátový modul je zapotřebí použít stejný typ, který je použit ve vysílacím modulu. Není to podmínkou funkčnosti, ale najít modul se stejnými parametry od jiné firmy je poměrně komplikované a časově velmi náročné na zprovoznění. Je tudíž jednodušší zvolit stejný model.

Mikrokontrolér:

Na mikrokontrolér nejsou kladeny žádné vysoké nároky. Musí mít jednu komunikační sběrnici umožňující komunikaci s bezdrátovým modulem (obvykle SPI nebo I²C). Pro komunikaci s počítačem bylo hned na začátku zvoleno rozhraní USB, kvůli velmi snadnému použití na jakémkoliv počítači. Jelikož cílovou skupinou jsou lidé vlastníci jednodeskový počítač⁷ (dále je používané označení minipočítač), mohlo být využito některé z obvyklých sběrnic, které jsou vyvedeny na většině z nich (UART, SPI, I²C) a zároveň na většině mikrokontrolérů. Toto řešení má ovšem velkou nevýhodu ve velmi špatné přenositelnosti kódu, poněvadž v každém minipočítači se na tyto sběrnice přistupuje jiným způsobem. Pro každý minipočítač je upraveno několik distribucí operačního systému Linux a každá verze má jinak zpracované ovládání těchto sběrnic. Další nevýhodou tohoto řešení je, že pokud uživatelem není člověk vlastníci minipočítač, ale klasický server, tak by tento uživatel neměl možnost toto zařízení připojit. Proto byla tato možnost již na začátku tvoření konceptu zavrhnuta a bylo rovnou rozhodnuto, že zařízení musí být připojitelné přes USB.

Dalším důležitým rozhodnutím bylo, zda-li má mít mikrokontrolér přímo integrovanou periférii USB umožňující mikrokontrolér připojit přímo na datové vodiče této sběrnice nebo jestli je vhodnější využít externího obvodu zajišťujícího převod některé z obvyklých sběrnic levnějších mikrokontrolérů na USB. Možnosti použití integrovaného obvodu jsou popsány v sekci *Převodník komunikačních sběrnic*. Bylo využito externího obvodu a důvody jsou popsány v sekci 2.2.1.

Na mikrokontrolér nejsou kladeny žádné vysoké požadavky z hlediska výkonu, protože v této aplikaci je jeho úkolem pouze čekat na přijatá data z přijímacího modulu, která si následně přečte a přepoše po jiné sběrnici do jiného zařízení. Z hlediska úspory času je

⁷ Veškeré potřebné součásti pro chod jsou umístěny na jednom plošném spoji. Nejrozšířenějším je RaspberryPi.

vhodné použít mikrokontrolér od stejné společnosti, která byla vybrána pro zařízení zajišťující měření teploty. Není to samozřejmě nutnou podmínkou pro chod, ale velmi to usnadní práci, protože na obou stranách je použita komunikace s bezdrátovým modulem a je zbytečné tento kód psát dvakrát. Jelikož ve vysílacím modulu je hlavním bodem nízká spotřeba, je nutné vybrat mikrokontrolér (obvykle) ve vyšší cenové kategorii, která umožňuje využít veškerý potenciál dané společnosti v nízkoodběrových aplikacích. Na straně serveru je spotřebě dána velmi nízká priorita z hlediska výběru součástek a tudíž je možné zvolit levnější alternativu v rámci stejné mikroprocesorové řady.

Převodník komunikačních sběrnic:

Pokud by nebyl použit mikrokontrolér s periferií USB, bylo by nutné použít převodník některé ze sběrnic, jejichž periferii vybraný mikrokontrolér obsahuje. Obvykle se používají převodníky UART-USB, protože součástí sběrnice UART je vodič, po kterém jsou vysílána data a druhý po kterém jsou přijímána. Z toho vyplývá mírná analogie s celým převodníkem tohoto typu, kde mikrokontrolér vyšle data a ony se objeví v USB přijímači. V obráceném směru to funguje analogicky. Počítač vyšle data po USB a ta se objeví v přijímacím bufferu mikrokontroléru. V případě použití jiných sběrnic by funkčnost byla samozřejmě zachována. Poté komunikace pracuje tak, že si integrovaný obvod vynutí obsluhu od mikrokontroléru, který si následně data musí vyčíst. Kvůli zjednodušení a zkušenostem s těmito obvody bylo rozhodnuto, že dále bude rozvíjena jen myšlenka převodníku USB-UART. V této kategorii je možné použít například následující obvody:

- 1) **FT231X** – Obvod realizující převod sběrnice USB na UART. Napájecí napětí je 5V. Obsahuje 4 vstupně/výstupní programovatelné piny. V základu je na těchto pinech například naprogramována signalizace přijetí a odeslání datového paketu. Podpora standardu USB 2.0. Samozřejmostí je i podpora veškerých nastavitelných možností sběrnice UART (rychlost, parita, stop bity atd.).
- 2) **MCP2200** – Obvod se srovnatelnými parametry jako FT231X. Jediným rozdílem je, že obvody FT jsou na trhu již velmi dlouho a díky tomu jsou jejich ovladače rozšířené napříč všemi operačními systémy.

Oba zmíněné obvody vyhovují veškerým požadavkům a dokonce velké množství požadavků několikanásobně převyšují. Proto byl vybrán již odzkoušený obvod FT231X a jeho aplikování bylo popsáno v sekci 2.2.1 .

1.6.2 Obslužný program

Základní nutností u návrhu softwaru je stanovit si na začátku co od tohoto programu očekáváme a vytvořit si nějakou základní koncepci. V následující části jsou shrnuty požadavky a způsoby řešení tohoto projektu.

Požadované funkce:

- Příjem dat z USB
- Zpracování přijatých dat
- Zápis dat do Systému řízení báze dat (dále používán ustálený pojem – databáze)
- Příprava dat pro internetové stránky

Z hlediska celkové koncepce bylo důležité rozhodnout, zda je lepší na každou část využít samostatný skript, který bude vykonávat jen tu jednu konkrétní funkci, nebo vytvořit program, který bude obsluhovat veškeré funkce.

Použití skriptu na každou funkci má tu výhodu, že potenciální chyba nečekaně nezastaví celý program, ale pouze tento skript a nenaruší tak další běh celého procesu. Případná chyba se může projevit například nezapsáním teploty do databáze nebo ztrátou komunikace s přijímacím modulem. Má to ovšem nevýhodu v necelistvosti kódu.

Použití jednoho programu má tu nevýhodu, že v případě neošetření některého chybového stavu se může celý program nečekaně ukončit bez jakéhokoliv varování. Velkou výhodou je veškerý kód na jednom místě a snazší následná editace.

Výběr programovacího jazyka:

- **Bash:** Velkou výhodou tohoto jazyka je, že je obsažen v každém počítači s operačním systémem Linux. To se ale zároveň stává i nevýhodou v případě připojení k počítači s jiným operačním systémem. Bash je skriptovací jazyk, kterým se přímo vykonávají systémové příkazy operačního systému Linux. Pro programátora, který v tomto jazyce nikdy nedělal, může být velmi obtížné něco vytvořit, protože má velmi přísnou syntaxi a celý program je tvořen o něco náročněji než v jiných jazycích. Další nevýhodou je nepřítomnost velkého množství obvyklých funkcí, které programátorovi usnadní práci a tím pádem je nutné si prakticky veškerý kód sestavit pomocí elementárních funkcí vlastními silami.

- **PHP:** Skriptovací jazyk PHP je rozšířeným jazykem vytvořeným primárně na tvorbu internetových stránek. Je v něm napsána valná většina nejrozšířenějších redakčních systémů (WordPress, Joomla, Drupal ...), nejrozšířenější systém pro správu databázových systémů (phpMyAdmin), výukový portál Moodle nebo částečně i nejrozšířenější sociální síť současnosti Facebook. Neznamená to ovšem, že v tomto jazyce nelze vytvořit nic jiného než internetovou stránku.
- **C:** Tento jazyk je zajímavou možností, protože pokud bude program přeložen přímo na serveru, na kterém se následně bude vykonávat, tak bude mnohem více využívat potenciál procesoru díky přímému využívání jeho nativních funkcí. Nevýhodou je, že je poměrně obtížné naprogramovat tak složité operace jakými jsou například práce s databází.
- **Java:** Za největší výhodu je považována multiplatformost⁸. Další výhodou je velká rozšířenost mezi programátory, kteří již vytvořili velké množství knihoven. Za výhodu se dá považovat i objektivnost, ale to již záleží na programátorovi tvořící aplikaci, zda-li to považuje za výhodu či nevýhodu.

1.6.3 Internetové stránky

Základním předpokladem internetových stránek bylo jejich vytvoření v jazyce HTML5. Z toho vyplývá možné použití jazyků HTML a JavaScript. Hlavním důvodem byla snadná přenositelnost kódu. Alternativou bylo použití skriptovacího jazyku PHP, ale to by přinášelo nutnost instalace PHP na server.

U internetových stránek je základem mít uživatelsky příjemný vzhled. Pro tuto práci byl vytvořen tmavý vzhled se žlutými prvky zajišťující dostatečný kontrast pro výraznou rozdílnost podkladu a prvků.

Přenos dat ze serveru k uživateli:

Jelikož byla jako úložiště dat zvolena MySQL databáze, tak nebylo tak triviální načítat naměřená data pouze pomocí HTML5. Jazyk HTML5 je složen z klasického HTML, který sám o sobě neumožňuje sečíst dvě čísla, natož načíst si část databáze, a JavaScriptu, který sice tyto věci umožňuje, ale je to skriptovací jazyk, který běží pouze u uživatele v prohlížeči. Musí

⁸ Pokud je programovací jazyk multiplatformní, lze ho spustit v jakémkoliv operačním systému.

se tedy zvolit nějaký "prostředník", který prohlížeči data předá. Pro tyto účely se používají soubory typu JSON, které jsou vygenerovanou tabulkou dat. U tohoto způsobu je komplikací, že tento soubor musí být přegenerován s každou změnou v databázi. Což se může při větším provozu stát problémem.

Další možností je skript běžící přímo na serveru, který zobrazí uložená data pouze v případě, že si je koncový uživatel vyžádá. Internetové stránky napsané v jazyce PHP pracují přesně na tomto principu. Skript uživateli vygeneruje HTML stránku z vnitřních dat serveru (části databází, soubory atd.) a tato stránka je následně uživateli zobrazena v jeho prohlížeči. Tato možnost se zdá být velmi výhodná, ale pro účely této práce není úplně vhodná. Je to dáno tím, že potenciální majitel tohoto výrobku používá jako server minipočítač s nepříliš velkým výpočetním výkonem. Čidlem, kterým je měřena teplota jednou za minutu, je vygenerováno 44640 vzorků za měsíc, nemluvě o ročním grafu či o možné přítomnosti několika čidel. Základním pravidlem u tvorby internetových stránek je, aby uživatel nebyl odrazen příliš dlouhou dobou čekání při načítání. A právě kvůli tomu byla tato možnost shledána jako nevyhovující.

Bezpečnost:

Důležitým parametrem při komunikaci uživatele se serverem je také bezpečnost. V tomto případě není nutné řešit bezpečnost přenosu informací od uživatele k serveru, protože uživatel neposílá serveru žádné citlivé informace a případný útočník se v nejhorším případě dozví, jaká byla u uživatele na zahradě před dvěma minutami teplota. Toto ani není považováno za bezpečnostní riziko, protože si to případný útočník může přečíst na stejných internetových stránkách jako uživatel. Za bezpečnostní riziko by se dal považovat průnik útočníka do skriptu běžícího na serveru. Základními metodami útoku jsou cross-site-scripting (XSS) a SQL Injection. Tuto problematiku je nutné řešit, pokud si interakce uživatele vyžádá nějaké nové spojení se serverem a server musí dekodovat, co si uživatel přeje a poslat mu nová data. XSS je způsob útoku, kdy útočník donutí server, aby spustil skript umístěný někde na internetu. Útočníkův skript má obvykle za úkol změnit zdrojový kód napadeného skriptu, aby byl na internetových stránkách zobrazován jiný obsah.

➤ Příklad neošetřené SQL Injection

Představme si situaci, kdy by uživatel v tomto projektu měl možnosti si na internetových stránkách zadat datum z minulosti, pro které by chtěl detailně zobrazit

průběh teplot. Případný útočník si nejprve zjistí, pomocí jakého jazyka jsou dané stránky vytvořeny a jaká databáze je použita. Dále se seznámí se základní syntaxí příkazů umožňující přístup do databází v daném jazyce. Pak už jen stačí do daného políčka, které očekává datum v nějakém formátu, napsat ukončení příkazu. Za to se dá volně pokračovat dalším příkazem. V tomto případě by útočník například nahradil veškeré naměřené teploty konstantní hodnotou.

Neošetřený kód na straně serveru by vypadal následovně:

```
vyhledej_datum_v_databázi ( datum_co_uživatel_zadal );
```

Případný útočník by mohl například zadat:

```
5.listopadu ); nastav_vše_v_databázi_na ( 0°C
```

Výsledek by poté vypadal následovně: (podtržená část je útočnickův kód)

```
vyhledej_datum_v_databázi(5.listopadu);  
nastav_vše_v_databázi_na(0°C);
```

Nejdříve se provede vyhledání data a poté se vše úspěšně přepíše.

Veškeré použité příkazy nejsou obsaženy v žádném programovacím jazyce. Byly vymyšleny pro názorné předvedení principu útoku.

Na útoky podobného typu jsou v každém jazyce ,ve kterém je možné vytvořit internetové stránky, základní příkazy chránící proti většině těchto útoků. Obvykle tyto příkazy odstraní speciální znaky, které se používají v příkazech spojených s prací s databází. Ochranu proti vniknutí na server je velmi důležité řešit a chránit se proti ní.

Grafy:

Jedním ze zásadních prvků internetových stránek jsou grafy zobrazující teplotu. Jsou obecně dva způsoby pro zobrazení grafů uživateli. První možností je vygenerovat obrázek s motivem grafu, který se následně uživateli zobrazí. Pomocí tohoto systému je například tvořen systém Cacti určený na monitorování počítačových sítí. Má to ovšem velkou nevýhodu v nízké interaktivitě. Druhým způsobem je použití JavaScriptu. Pro vykreslení grafu je stažen ze serveru výpis teplot, který se následně vykreslí přímo u uživatele v prohlížeči. Jelikož je interaktivita základním předpokladem pro komfort uživatele, byla tato cesta zvolena jako vhodná k použití. Je ovšem velké množství knihoven, které jsou schopny dané grafy vykreslit. Zde je krátký výčet základních knihoven:

- **Google Chart API** – Velmi využívaná možnost velkým množstvím programátorů. Pokud se spokojíme se vzhledem grafů, tak je práce skoro hotova. Stačí pouze grafu podstrčit data a vše se samo vykreslí. Výhodou je možná propojitelnost s dalšími API od společnosti Google.
- **Metrics Graphics** – Knihovna má značné možnosti v zobrazení grafů. Zprvu byl tento projekt na této knihovně vyvíjen. Po velmi zdoluhavých úpravách knihovny (snaha o korektní zobrazení hodin v denním grafu) bylo zvoleno jiné vykreslovací jádro grafů a tím bylo poslední zmíněné.
- **HighCharts** – Knihovna s nejširšími možnostmi ze všech zmíněných. Velkou výhodou je velká rozšířenost mezi programátory, čímž je zajištěna velká podpora. Možných funkcí je velká řada a je zde zmíněno jen pár základních. Knihovna umožňuje vykreslení několika grafů se shodným nastavením, vytváření plovoucí legendy ke grafům, animace při načítání, aktualizace dat v grafu a má mnoho dalších praktických funkcí.

2 Praktická realizace

V této části bude popsán způsob výsledné realizace a důvody volby jednotlivých komponent.

2.1 Zařízení realizující měření teploty

Schéma + plošný spoj:

Kompletní text této části je úzce spjat se schématem z *Přílohy*.

Ze všech možností uvedených v části *Teoretický rozbor* byly vybrány následující součástky:

- Mikrokontrolér – ATmega88PV
- Bezdrátový komunikační modul – Nrf24101+
- Teplotní senzor – Si7053

Zapojení samotného mikrokontroléru není nijak specifické. Napájení mikrokontroléru je přímo u napájecích pinů vyhlazeno pomocí blokovacího kondenzátoru C1 s hodnotou 100nF. Výrobce doporučuje připojení resetovacího pinu přes pull-up rezistor k napájecímu napětí.

V tomto zapojení se jedná o rezistor R1 s hodnotou 10k Ω . Je zde vyvedena sběrnice SPI včetně resetovacího pinu pro naprogramování mikrokontroléru. Další částí je vyvedení sběrnice I²C (mikrokontrolér - piny PC4 a PC5), kde má každý vodič vlastní pull-up rezistor s hodnotou 10k Ω (R2 a R3). Tato hodnota vychází z katalogového listu použitého teplotního čidla Si7053 (viz zdroj [9]). Pro bezdrátový komunikační modul je použita sběrnice SPI, dva vodiče jako chip-select a pin modulu vyvolávající přerušení, který je připojen na pin PD3 umožňující vytvoření přerušení programu v případě změny napěťové úrovně. Ještě se v zapojení vyskytují kondenzátory C2 a C3 s hodnotou 100nF, kde první zmíněný by měl být umístěn co nejbližší napájecím pinům teplotního čidla a druhý u bezdrátového modulu Nrf24l01+. Ve schématu i na plošném spoji je zakreslen krystal Q1 s kondenzátory C98 a C99 s hodnotou 22pF. Tuto část není nutné implementovat. Je zde zakreslena pro případný požadavek přesnějšího časování než zajišťuje vnitřní 8mi MHz oscilátor.

2.1.1 Mikrokontrolér a program

Volba mikrokontroléru je nejdůležitější částí celého zařízení, protože se od něj odvíjí celé programování trvající nejdelší část doby vývoje. Z výběru, zmíněného v sekci 1.2, byl nakonec vybrán mikrokontrolér ATmega88PV. Jedná se o novou generaci velmi používaného ATmega8. Novinkou u této generace je například snížená spotřeba, přidání několika dalších režimů spánku, rozšíření čítače v kontrolním obvodu watchdog a mnoho dalšího. Již z názvu mikrokontroléru se dají poznat některé základní rysy:

- ATmega – Mikrokontroléry RISC s 8mi bitovou sběrnicí.
- 88 – Nová generace ATmega8. Číslo 8 značí velikosti paměti typu Flash, do které se nahrává program.
- PV – Snížená frekvence na rozsah 0-10MHz a snížené operační napětí na rozsah 1,8-5,5V. Obojí je zároveň vhodné pro snížení spotřeby.

Vybraný mikrokontrolér disponuje velkým množstvím periférií, které v projektu nejsou využity. Jsou jimi například AD převodníky, čítače, časovače, podpora Qtouch⁹, analogový komparátor, USART apod.

⁹ Qtouch je systém vyvinutý společností Atmel pro ovládání tlačítek založených na principu změny kapacity v důsledku dotyku.

Výrobce umístil do mikrokontrolérů použité generace registr, ve kterém je možné nepoužité periferie zakázat. Tento registr se jmenuje PRR (Power reduction register). Periferie jsou zakázány vypnutím hodinového signálu, který do nich přichází. Tím se sníží spotřeba a zároveň jsou data uchována až do výpadku napájení či opětovného povolení hodinového signálu. Uchování dat ovšem není zaručeno a výrobce doporučuje kompletní reinicializaci periferie v případě opětovného použití. Zmíněný registr má následující složení bitů:

Tab. 4: Power reduction register

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------------|---------------|---------------|---|---------------|--------------|-----------------|--------------|
| PRTWI | PRTIM2 | PRTIM0 | - | PRTIM1 | PRSPI | PRUSART0 | PRADC |

A jejich význam je následující: (PR v názvu každého znamená Power Reduction)

- PRTWI – vypnutí TWI (periferie podobná I²C)
- PRTIM2, PRTIM0, PRTIM1 – vypnutí čítačů 2, 0 a 1
- PRSPI – vypnutí sběrnice SPI
- PRUSART0 – vypnutí komunikační periferie USART
- PRADC – vypnutí AD převodníků a analogových komparátorů

Program:

Pro snadnější práci s periferiemi a zařízeními byl pro každou část sepsán hlavičkový soubor obsahující definice s nastavením a hlavičky funkcí. Ke každému hlavičkovému souboru (kromě souboru s protokolem) byl vytvořen soubor s příponou „*.c“, pomocí kterých jsou dodeklarována těla funkcí. Jsou to tedy soubory i2c.h , spi.h pro periferie a nrf.h pro komunikaci s modulem Nrf24l01+. Veškeré zdrojové kódy jsou umístěny na přiloženém CD ve složce Teplotni_senzor/Program.

Program se dá rozdělit do dvou hlavních částí. První částí je prvotní inicializace, která se provede pouze v případě výměny baterie či při případném resetování mikrokontroléru v důsledku chybového stavu vyvolaného kontrolními obvody. Druhou částí je nekonečná smyčka, která má za úkol obstarat měření a odesílání teplot.

Před vysvětlením jednotlivých částí programu je na místě vysvětlit způsob komunikace a předání dat mezi dvěma zařízeními v tomto projektu. Komunikační obvod Nrf24l01+ využívá svůj vlastní protokol na posílání zpráv, který obsahuje datovou část, CRC, adresy atd. V případě bezchybného přenosu se datová část dá vyčíst z komunikačního modulu na straně

přijímače. V oné datové části byl vytvořen vlastní protokol, který by měl být popsán před dalším popisem programu.

Protokol byl sestaven v jazyce C jako struktura ve formě bitového pole. Tento fakt umožnil, aby vysílaná data nebyla upravitelná po bytech¹⁰ ale po bitech či skupinách bitů. Celý paket (tím je myšlen paket datové části a ne veškeré informace přenášené mezi komunikačními moduly) je dlouhý 8 bytů. Pro názornost je zde uvedena definice typu, pomocí kterého je paket vytvářen.

```
typedef union protocol {
    struct {
        unsigned char receiver_address_category    : 4;
        unsigned char receiver_address_id         : 4; // 1B

        unsigned char transmitter_address_category : 4;
        unsigned char transmitter_address_id      : 4; // 1B

        union state {                               // 2B
            struct {
                char message_type    : 4; // 16 types
                char SW_version      : 3; // 8 versions
                char device_status   : 1;
                char unused          : 8;
            };
            short status;
        } state;
        signed long data; // 4B
    };
    unsigned char buffer[8];
} protocol;
```

Ve výše uvedené definici je možné si všimnout několika náležitostí. Celý datový typ je unionem se dvěma prvky – strukturou a polem bytů. Zmíněné pole je zde pouze pro usnadnění práce s touto proměnnou v dalších funkcích. Samotná struktura umožňuje pohodlné nastavení jednotlivých částí paketu.

V prvním bytu se nastaví informace o zařízení, pro které je paket určen. Tento byte nemá v této práci příliš význam, ale má velký význam v dalším rozšiřování bezdrátové sítě. Návrh této sítě pracuje na principu existence jednoho hlavního prvku (pro tento účel bude využita serverová část tohoto projektu), který vytváří spojovací bod pro všechna zařízení v síti. Pokud

¹⁰ Byte = skupina 8mi bitů

tedy zařízení A chce vyslat signál do zařízení B, musí zpráva projít vždy přes hlavní zařízení, které díky tomu může monitorovat stav celé sítě. Tento způsob komunikace umožňuje snadnou kontrolu nad celou sítí, protože je vše ovládáno kontrolním programem v serveru. Tento program je možné velmi pohodlně a vzdáleně měnit. Pokud se ovšem na tento byte podíváme detailněji, je možné si všimnout rozdělení tohoto bytu na dvojici proměnných, každou o velikosti 4 bity. Tento byte je tedy rozdělen na kategorii a identifikátor (ID) zařízení. Důvodem je možné budoucí rozšíření, ve kterém je plánováno větší množství zařízení a znalost tohoto bytu tak dává možnost snadné identifikace zařízení. Pro tento účel byly vytvořeny dvě základní kategorie – `category_system` (0) a `category_thermometer` (1). Celý systém by měl v budoucnu posloužit jako základní stavební kámen inteligentní domácnosti. Je možné přidávat například kategorie na světla, senzory atd. V každé kategorii jsou zařízení s unikátním ID (unikátní pro danou kategorii). V tomto projektu je komunikace pouze jednosměrná a probíhá pouze ze strany čidel k serveru, proto mají všechna čidla nastaveny tento byte na `category_server` (0) a `device_server` (0). Rozdělení zařízení do jednotlivých kategorií umožňuje na straně serveru poslat zprávu celé kategorii (inspirací byl broadcast v počítačových sítích).

Druhý byte je analogický s tím prvním. Jedinou změnou je, že zde jsou informace o zařízení, které zprávu vysílá. Definice kategorie a zařízení umožňuje velmi jednoduché přidání dalších teplotních čidel. Stačí pouze vytvořit unikátní ID pro kategorii s teploměry a na straně serveru následně proběhne jednoduchá detekce určující, se kterým zařízením je právě komunikováno.

Následující 2 byty jsou pro definování stavu zařízení. Jsou zde tedy umístěny informace o verzi softwaru, informace zda-li jsou přenášena data chybovým hlášením či platnými daty a pak je zde několik nevyužitých bitů, které budou do jisté míry v budoucím rozšiřování zapotřebí.

Posledními čtyřmi byty je konkrétní přenášená zpráva, která je specifická pro každé zařízení. V tomto projektu se přenáší znaménkový celočíselný typ `signed long` o velikosti 4 byty. Zároveň je ale požadavkem přenášet teplotu na 2 desetinná místa. Tudíž se přenáší stonásobek skutečné hodnoty. Pro teplotu $-21,76^{\circ}\text{C}$ se tedy přenesou číslo -2176 , které se na straně serveru zpracuje a uloží již ve správném formátu.

Program – prvotní inicializace:

V této části je nutné nastavit mikrokontrolér tak, aby nemusel některé základní nastavení vykonávat v případě každého oběhu nekonečné smyčky. Jedná se tedy o nastavení parametrů zprávy, které jsou neměnné (informace o přijímači a vysílači). Dále výrobce doporučuje zapnout všechny nevyužité piny jako vstupní a zapnout na všech pinech vnitřní pull-up rezistory. Následně se inicializuje sběrnice SPI a samotný modul Nrf24l01+. V této inicializaci se nastaví funkce jednotlivých vstupně výstupních pinů, nastavení rychlostí atp. Dále se nastavuje registr pro snížení spotřeby PRR (popsán v části 2.1.1). Provede se zde příkaz na uspání komunikačního modulu, aby v případě prvního běhu nebyl aktivní až do prvního okamžiku odeslání, po kterém je uspán. A celá inicializační část je zakončena globálním povolením přerušeni příkazem `sei()`; pro možné využití přerušeni od kontrolního obvodu watchdog.

Program – nekonečná smyčka:

Na začátku nekonečné smyčky se provede inicializace watchdogu pomocí funkce `wd_init()`. V této funkci se zapne povolení přerušeni od watchdogu v okamžiku přetečení jeho vnitřního čítače. Zakáže se zde schopnost resetu programu v okamžiku přetečení a nastaví se délka čítače (změna periody přetečení). V průběhu programování nastal problém s korektním nastavením z důvodu ochrany této periferie výrobcem. Cílem této ochrany je zajistit, aby „bloudící“ kód náhodně nezměnil nastavení tohoto kontrolního obvodu. Je tedy nutné vykonat dvě specifické operace ve velmi rychlém sledu. Zápisem do kontrolního registru se mikrokontroléru oznámí, že se bude upravovat nastavení této periferie a poté má program 4 hodinové cykly na to, aby ho nastavil. Jinak se snaha o nastavení ignoruje a požadované nastavení se neprovede. Tato inicializace nám zajistí, že každých 8 sekund (maximální nastavitelná doba) se mikrokontrolér dostane do přerušeni generovaného watchdogem. Jelikož není požadováno, aby byla teplota měřena každých 8 sekund, tak je v tomto přerušeni pouze zvětšována hodnota proměnné `time`, která znamená počet průchodů tímto přerušením, tedy počet 8mi sekundových cyklů bez vyslání dat. Případný uživatel si před naprogramováním může definovat hodnotu, po které se mají data změřit a poslat. Výchozí hodnotou je 32 sekund a v případě změny stačí pouze upravit `RELOAD_TIME` v hlavním souboru. Program je schopný tuto akci vykonat pouze jednou za 8 sekund, s čímž se musí logicky počítat a hodnotu nastavit na základě toho. Vykona se to buď v nastavený čas nebo v nejbližší vyšší násobek 8mi. Maximální nastavitelná hodnota je dána maximálním

rozsahem proměnné `time` (255) vynásobeno 8mi, což znamená 34 minut. Proměnná `time` musí být nadeklarována jako `volatile`, aby se při každém průchodu přerušením znovu načetla její aktuální hodnota.

Po inicializaci `watchdogu` se zkontroluje stav proměnné a ověří se, zda nenastal čas vyslat teplotu. Pokud je tato podmínka splněna, je vynulována proměnná `time`, změřena teplota, jsou vyslána data a znovu uspán komunikační modul.

Měření teploty je vykonáno externí součástkou Si7053 popsanou v sekci 2.1.2. V této sekci je popsán i obslužný program pro mikrokontrolér. Stejně tak použití a způsob vyslání dat je popsáno v sekci 2.1.3 o komunikačním modulu.

Za podmínkovým blokem je umístěna část zajišťující maximální snížení spotřeby v okamžiku, kdy není vyžadován běh procesoru. Režim spánku (z anglického `sleep mode`) je stav mikrokontroléru, při kterém je vypnut přívod hodinového signálu k některým částem a tím snížena spotřeba. Prvním krokem je nastavení požadovaného režimu. Režimů je poměrně značné množství a každý se používá v jiné aplikaci. Základními režimy jsou režimy pro eliminaci rušení (při převodu AD převodníku) způsobené vnitřním hodinovým rozvodem a operacemi v mikrokontroléru. Dalšími jsou režimy `standby` pro snížení odběru, ale se zajištěním některých základních funkcí zajišťující rychlý start po probuzení. Pokud je mikrokontroléru vypnut hlavní oscilátor, tak probuzení trvá delší dobu, protože je nutné počkat, dokud se hodinové impulsy neustálí na konstantní frekvenci. V režimech `standby` jsou oscilátory povoleny. Jako poslední jsou režimy nejhlubšího spánku – `power_down` a `power_save`. Tyto režimy jsou schopné vypnout kompletně celý mikrokontrolér. Mezi těmito dvěma bylo vybíráno a jediný rozdíl je, že v režimu `power_save` může běžet jeden z čítačů/časovačů v asynchronním módu. Tato funkce nebyla od programu požadována a tím pádem byl ze všech uvedených režimů vybrán režim `power-down`, který vypne hodinové impulsy od jádra procesoru přes paměť programu až ke vstupně-výstupním pinům. Tím deaktivuje jakoukoliv možnost vykonávání programu. Z tohoto režimu může být mikrokontrolér „vzbuzen“ externím přerušením (změna napěťové úrovně na konkrétních

pinech), shodou adresy I²C (v případě, že je mikrokontrolér nastaven jako slave¹¹) a watchdogem.

Pro aplikování režimů spánku v programu bylo využito oficiální knihovny od firmy Atmel pro tento typ mikrokontrolérů. Tato knihovna se nachází v základních knihovnách Atmel Studia¹². Pro použití dané knihovny je zapotřebí oznámit překladači, že danou knihovnu chceme využít typickou direktivou preprocesoru `#include`. Pro použitý typ mikrokontroléru vypadá příkaz následovně: `#include <avr/sleep.h>`. Poté stačí v samotném programu použít následující sled příkazů:

```
set_sleep_mode(SLEEP_MODE_PWR_DOWN);
sleep_bod_disable();
sleep_mode();
```

První příkaz nastaví požadovaný režim. Druhý zakáže kontrolní obvod Brown-out detection, který při poklesu napájecího napětí provede reset mikrokontroléru. Tento kontrolní obvod není v práci použit a v případě nepoužití je doporučeno ho přímo zakázat pro maximální snížení odběru. Třetí a zároveň poslední příkaz celé smyčky zajistí, aby se vstoupilo do režimu spánku a v případě přerušení od watchdogu se tento režim řádně ukončil, aby bylo možné v programu dále pokračovat.

2.1.2 Teplotní čidlo

Vybrané teplotní čidlo Si705X disponuje skutečně výjimečnými parametry pro nízkoodběrové aplikace. Jeho proud v režimu spánku je dle katalogového listu 60nA. V praxi nebylo možné tuto hodnotu ověřit, což je dále popsáno v sekci 2.1.5. Pro účely této práce bylo vybráno nejlepší možné čidlo z této řady a to Si7073 s přesností $\pm 0,3^{\circ}\text{C}$.

Zapojení teplotního čidla je již popsáno v kapitole 2.1 a její sekci *Schéma + plošný spoj*. Pro komunikaci s čidlem je použita sběrnice I²C. Adresa čidla je přímo určená křemíkovým čipem a je neměnná s hodnotou 0x40 v hexadecimálním tvaru. Pevná adresa má výhodu ve snazším návrhu, kde není nutné nijak adresu určit zapojením. Má ovšem i nevýhodu

¹¹ Výraz pro zařízení na sběrnici, které komunikuje pouze v případě, že je mu dán prostor pro komunikaci od zařízení kontrolující danou sběrnici (označováno jako master).

¹² Atmel Studio – volně dostupný program vydaný společností Atmel pro programování mikrokontrolérů od stejnojmenné společnosti.

v možnosti užití pouze jediného čidla na jedné I²C sběrnici. Čidlo je schopné pracovat ve dvou základních režimech:

- 1) Hold master mode – Režim, ve kterém je celá sběrnice rezervována pro komunikaci s teplotním čidlem. Po vyslání příkazu značícího začátek měření je nutné stále měnit napěťovou úroveň vodiče vedoucího hodinový signál. V okamžiku dokončení převodu začne čidlo po jednotlivých bitech vysílat data na sběrnici, kde jsou zaznamenávána mikrokontrolérem.
- 2) No hold master mode – V tomto režimu je vyslán příkaz pro měření. Poté je možné vykonávat jinou činnost a po nějaké době si mikrokontrolér vyčte změřený údaj. Rozpoznání, zda-li se převod ukončil nebo jestli je stále aktivní je patrné z odpovědi teplotního čidla. Pokud převod není ještě zcela hotov, tak čidlo nereaguje na žádný podnět ze strany mikrokontroléru. Pokud je již převod dokončen, tak čidlo zareaguje na výzvu ze strany mikrokontroléru platným bitem acknowledge¹³.

Výběr požadovaného režimu se provede vysláním rozdílného příkazu. Pro první režim se začne měřit po vyslání příkazu 0xE3 a pro druhý režim je použit příkaz 0xF3. Pro účely této práce byl vybrán režim č.2 v mírně modifikované podobě, při které je čidlo neustále dotazováno, zda-li již převod dokončilo, a v okamžiku dokončení převodu se začnou přijatá data ukládat do proměnných v programu. Před začátkem programování bylo počítáno s možností vyslání příkazu pro začátek měření, následné usnutí mikrokontroléru. Po probuzení by byla data vyčtena a zpracována. Bohužel nejkratší doba, po kterou je mikrokontrolér schopen být v režimu spánku je 16ms a to je přibližně trojnásobek průměrné doby převodu. Změřená doba převodu je patrná z grafu – viz *Obrázek 4*.

Data vyslaná z teplotního čidla nejsou ve formátu, ze kterého by šla rovnou a jednoduše přečíst teplota. Je nutné vykonat přepoččet podle následujícího vzorce:

$$teplota[{}^{\circ}C] = \frac{175,72 * data}{65536} - 46,85$$

Výsledná teplota je poté uložena do paketu jako stonásobek skutečné hodnoty a odeslána jako celočíselný datový typ `signed long`.

¹³Stav, při kterém si přijímač vynutí dominantním stavem na sběrnici. Díky tomuto stavu dokáže vysílač zjistit, jestli data k některému zařízení skutečně dorazila.

2.1.3 Komunikační modul

Jako komunikační modul byl zvolen Nrf24l01+, kvůli snadné aplikaci a nízkému odběru v režimu spánku. Výhodou je i zabezpečení komunikace z hlediska detekce chyb. Samotný modul umí rozpoznat, jestli data byla přijata v pořádku či ne. Na straně přijímače umí rozpoznat korektnost dat podle CRC a podle toho dále jedná. V případě korektního přenosu informuje vysílač, že paket byl přijat v pořádku a informuje o novém paketu ovládací prvek (mikrokontrolér) pomocí přerušení. V případě chybného přenosu nepošle přijímač acknowledge bit a pro vysílač to znamená, že paket nedorazil. Pro správné přijetí acknowledge bitu je nutné mít nastavenou jednu z přijímacích adres na stejnou hodnotu jako je hodnota zařízení, kterému je zpráva určena.

Tento modul má i další výhody z hlediska usnadnění práce. Základem je zmiňovaný acknowledge, který napomůže korektnosti přijatých dat. Další výhodou je kupříkladu možnost „auto-retransmit“, která zajistí, že v případě chyby přenosu je celá zpráva odeslána znovu. Počet pokusů a čas mezi nimi je nastavitelný pomocí vnitřních registrů modulu. V tomto projektu byl počet nastaven na 16 a čas mezi nimi na 1,5ms. Dále se dá nastavit kanál přenosu na hodnotu 0-127. Reálná frekvence přenosu se poté vypočítá ze vztahu: $2400\text{MHz} + \text{kanál} * 1\text{MHz}$. Kvůli legislativě je ovšem možné použít pouze prvních 84 kanálů (kanály 0-83), protože volné pásmo je 2400MHz – 2483,5MHz. Vysílání na kanálech 84-127 není v České republice povoleno, protože to již patří mimo pásmo určené k volnému provozu a je protiprávní. Pro správný chod je nutné nastavit shodný kanál na obou zařízeních.

Moduly Nrf24l01+ jsou adresovatelné. To znamená, že každý modul má programovatelnou adresu pro vyslání zprávy a adresu pro příjem. Adres pro příjem může být až 6. Modul má také několik přijímacích bufferů pro případ, že přijde zpráva ze dvou adres v jeden okamžik. Délka adresy je nastavitelná v rozsahu 3-5 bytů. Pro vyslání zprávy byla vytvořena funkce `nrf_send_message()`, která jako parametry očekává adresu, buffer s daty a počet, kolikrát se má zkusit opětovné odeslání dat v případě nezdařilého přenosu. Tato funkce a mnoho dalších lze nalézt v souborech `nrf.c` a `nrh.h` na přiloženém CD ve složce `Teplotni_senzor/Program/meteobug/lib`.

2.1.4 Baterie

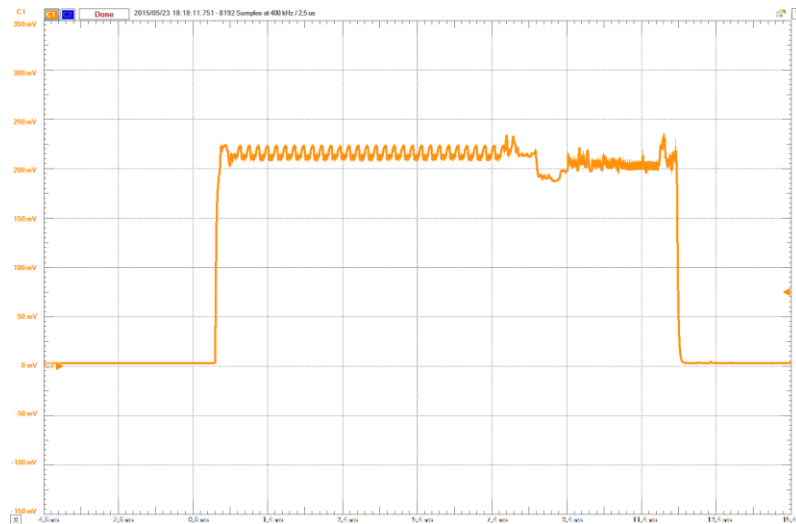
Ze všech možností zmíněných v kategorii 1.5 byly vybrány baterie LS 14250 pro své výborné vlastnosti. Ve všech ohledech jsou lepší než ostatní uvažované baterie. Za největší výhodu se dá považovat velmi nízké samovybíjení a fakt, že napětí na baterii s časem a odběrem příliš neklesá. Tím pádem může zařízení pracovat až do posledního okamžiku před vybitím. Je to zároveň ale i velkou nevýhodou. Méně kvalitním bateriím napětí s časem klesá a díky tomu je možné včas detekovat neodvratné vybití. Díky této detekci je možné uživatele upozornit. Upozorněn může být emailem, SMS zprávou či jiným informačním kanálem. Stačí to pouze implementovat do programu běžícího na serveru.

Pokles napětí se projeví se zatížením. Je ovšem závislý především na teplotě a odebíraném proudu. Avšak i pro okamžik odeslání dat, kdy je proud 14,4mA by napětí mělo být stále dostatečné pro chod. Pokud budeme vycházet z extrémních podmínek pro provoz baterie, tak například pro teplotu -40°C a odebíraný proud 15mA je napětí přibližně 2,6V. Všechny použité součástky jsou schopné pracovat od 1,9V, takže je zde ještě určitá rezerva. Vypsání hodnoty bylo zjištěno v katalogovém listu (viz zdroj [13]) k daným bateriím.

Zařízení splňuje i podmínku maximálního odebíraného proudu z baterie, který je 35 mA.

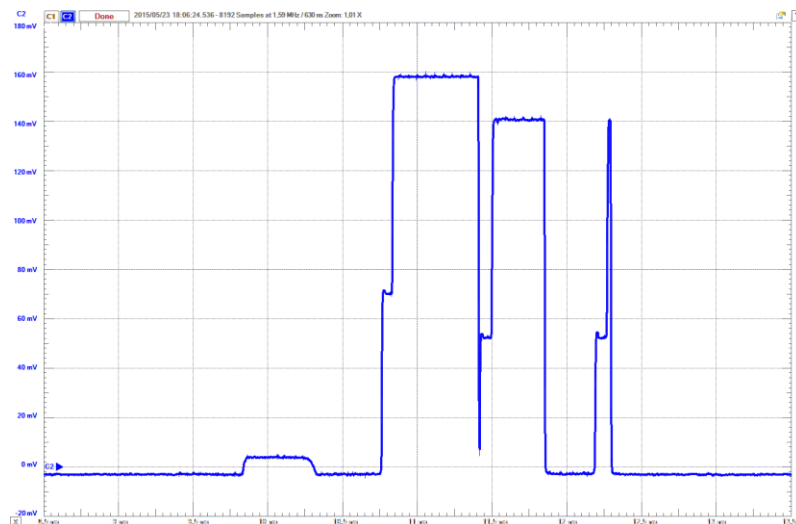
2.1.5 Celková spotřeba energie

Spotřeba energie vychází ze změřených a vypočtených hodnot. U baterie byla použita hodnota z katalogového listu, protože není možné změřit kapacitu této baterie bez jejího kompletního vybití. Na určení spotřeby energie je nutné znát proud, který do jednotlivých součástek vtéká. Jelikož není možné měřit přímo proud, tak je nutné měřit napětíový úbytek na rezistoru, vytvořený právě tímto proudem. Tato metoda spočívá v tom, že je připojen sériově rezistor mezi napájecí pin součástky a rozvod napájecího napětí. Pomocí úbytku lze následně vypočítat proud tekoucí do součástky pomocí Ohmova zákona. Problémem je zvolit vhodnou hodnotu rezistoru. Pro příliš velkou hodnotu se při špičkovém odběru na něm vytvoří velké napětí a způsobí to nesprávnou funkci součástky. Z důvodu nízkého napětí součástka následně nemusí pracovat správně a tím pádem je měření neplatné. Pro každou součástku musela být zvolena nejvhodnější hodnota rezistoru, aby byla zachována funkčnost a úbytek byl co největší. O co největší úbytek se snažíme z důvodu eliminace rušení a chyb tím vzniklých. Hodnoty spotřeby pro jednotlivé součástky jsou tedy následující:

Mikrokontrolér ATmega88PV:

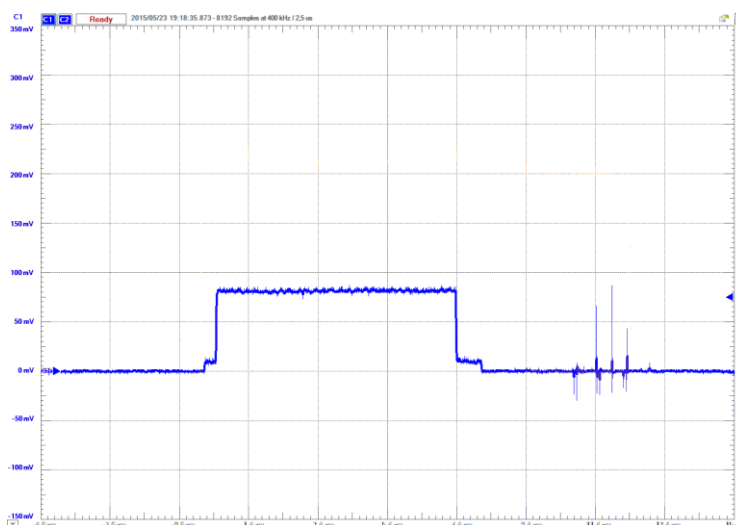
Obrázek 2: Spotřeba mikrokontroléru ATmega88PV

Délka impulsu je 12,33ms a jeho průměrný úbytek napětí je 215mV. Pro měření byl použit rezistor 330,5 Ω . Po použití Ohmova zákona je tedy odebíraný proud 651 μA . Pro měření odběru v době spánku byl použit rezistor 9,96 k Ω a byla změřena hodnota 48,6 mV. Z toho byl vypočten proud 4,88 μA .

Nrf24101+:

Obrázek 3: Spotřeba komunikačního modulu Nrf24101+

Délka impulsu je 1,094 ms, úbytek odpovídá 157 mV a pro použitý rezistor 10,9 Ω vychází proudový odběr pro okamžik vysílání 14,4 mA. Okamžik spánku byl měřen rezistorem 220 k Ω a byla změřena hodnota 161,8 mV. Z toho byl vypočten odběr 0,73 μA .

Teplotní čidlo:

Obrázek 4: Spotřeba teplotního čidla Si7053

Odběr teplotního čidla Si7053 je skutečně tak malý, jak bylo předpokládáno. Úbytek v době impulsu je 72mV. Pro použitý rezistor 683 Ω vychází proud 105 μ A. Délka impulsu je 6,91ms z důvodu použití 14-ti bitového rozlišení převodu. V době spánku byl odběr menší, než byla měřitelná úroveň. V dalších výpočtech je tedy považován za téměř nulový.

Celková spotřeba:

V tabulce je souhrn jednotlivých součástí a informací o nich.

Tab. 5: Přehled součástí a jejich odběrů.

| | ATmega88PV | Nrf24101+ | Si7053 |
|-----------------|--------------|--------------|-------------|
| Délka pulsu | 12,33 ms | 1,094 ms | 6,91 ms |
| Špičkový proud | 651 μ A | 14,4 mA | 105 μ A |
| Odběr ve spánku | 4,88 μ A | 0,73 μ A | 0 |

Důležité informace pro výpočty:

Perioda měření: 32 s

Doba spánku: 32s – 12,33ms

Kapacita baterie: 1,2 Ah = 1,2 * 10⁶ μ Ah

Výpočty:

$$\text{Průměrný proud při měření} = \frac{651 \cdot 12,33 + 14400 \cdot 1,094 + 105 \cdot 6,91}{12,33} = 1987,5 \mu\text{A}$$

$$\text{Průměrný proud při spánku} = 4,88 + 0,73 = 5,61 \mu\text{A}$$

$$\text{Průměrný proud za celou periodu měření} = \frac{1987,5 \cdot 12,33 + 5,61 \cdot (32\,000 - 12,33)}{32\,000} = 6,3736 \mu\text{A}$$

$$\text{Doba funkčnosti} = \frac{1,2 \cdot 10^6}{6,3736} = 188276 \text{ hod} = 7845 \text{ dnů} = 21,49 \text{ let}$$

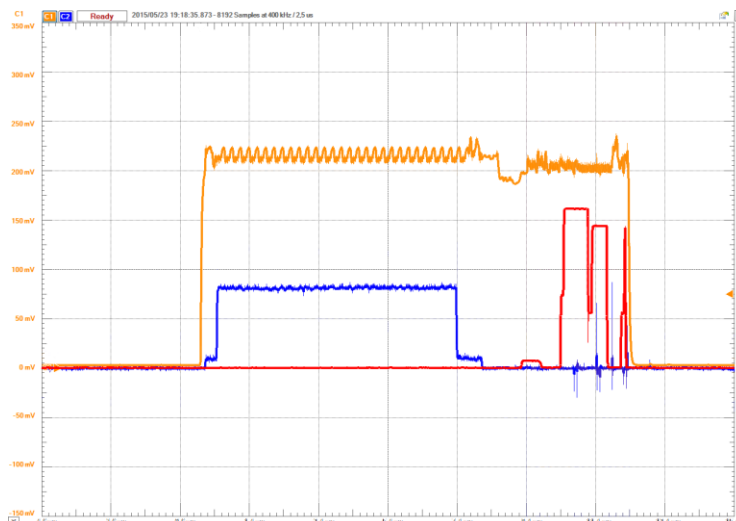
Jelikož doba funkčnosti vyšla přibližně na 21 let, je vhodné zpětně započítat samovybíjení baterie. Kapacita baterie bude tedy ve výpočtu snížena o 21%, protože samovybíjení u baterie LS14250 je 1% za rok.

$$\text{Doba funkčnosti se započteným samovybíjením} = 16,977 \text{ let}$$

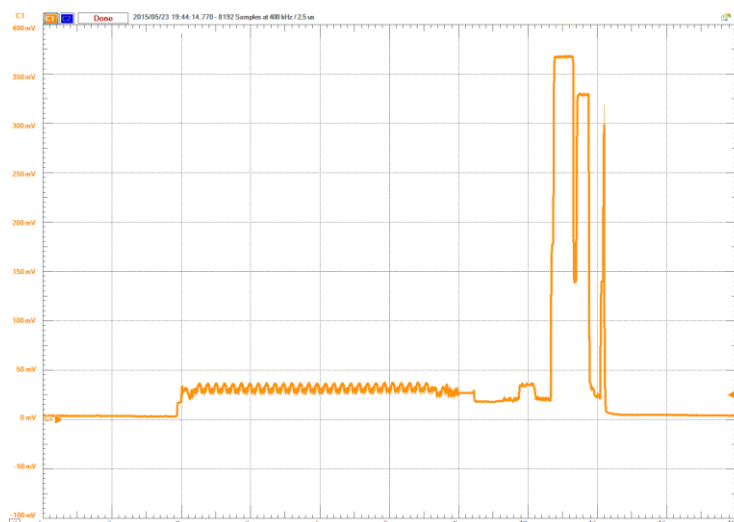
Vypočtená hodnota je pouze teoretická. Veškeré hodnoty, se kterými bylo počítáno, platí pouze pro jasně dané laboratorní podmínky. Například s teplotou velmi kolísá kapacita akumulátoru. Samovybíjení je započteno velmi jednoduchým způsobem.

Obrázek 5 velmi přehledně znázorňuje spínání jednotlivých prvků. Oranžová křivka znázorňuje mikrokontrolér, modrá křivka značí teplotní čidlo a červená křivka bezdrátový komunikační modul. Je nezbytně nutné si uvědomit jeden důležitý fakt. Aby bylo možné zobrazit časování jednotlivých prvků v jednom přehledném grafu, bylo zapotřebí upravit měřítko pro každý prvek.

Pokud je požadováno stejné měřítko, tak celkový odběr zařízení (měřen přímo odběr proudu z baterie) je vyobrazen v grafu – viz *Obrázek 6*. Z tohoto obrázku je patrný markantní rozdíl v odběrech, kde komunikační modul několikanásobně převyšuje odběr všech zbylých součástí. Tento fakt je v souladu s teoretickými předpoklady.



Obrázek 5: Znárodnění časové souslednosti zapínání jednotlivých prvků.
POZOR ! Měřítka je pro každý kanál rozdílné !!!



Obrázek 6: Spotřeba celého zařízení

2.2 Server

Jako server je možné využít jakýkoliv počítač. Samotné zařízení nemá žádnou paměť na přijatá data a tím je sběr dat možný pouze v případě, že je počítač zapnut. Díky nízké spotřebě je doporučeno použít některý z minipočítačů. Jsou to zařízení, na kterých běží mírně modifikovaný operační systém a je možné na něm provádět většinu funkcí jako na běžném počítači. Je to samozřejmě limitováno operačním výkonem a velikostí operační paměti. Cena se pohybuje maximálně do 1000 Kč a odběr je nejvýše 5W. Jako vhodný minipočítač bylo vybráno pcDuino1, které disponuje architekturou procesoru ARM Cortex A8, 1GB operační paměti, ethernetovým portem s rychlostí 100Mbps, třemi USB porty a HDMI výstupem.

2.2.1 Serverový přijímač

Na serverový přijímač nebyly kladeny vysoké nároky z hlediska spotřeby ani z hlediska výpočetního výkonu. Má totiž za úkol pouze přeposílat data přijatá z modulu Nrf24l01+ přes UART do serveru, ke kterému je připojen. Serverový přijímač má za úkol plnit pouze tento jednoduchý úkol. Tento způsob byl vybrán především kvůli velké náročnosti vzdáleného přeprogramování zařízení a na druhé straně kvůli velmi jednoduché aktualizaci ovládacího programu běžícího přímo v serveru.

Schéma + plošný spoj:

Realizace je patrná ze schématu vyobrazeném v příloze. Veškeré výrobní podklady jsou umístěné na CD ve složce Server/Plosny_spoj. Celé zapojení je řízeno mikrokontrolérem ATmega8, který je přes SPI připojen k programovacímu konektoru a k modulu Nrf24l01+. Byla zde použita jiná verze modulu než u teplotního čidla. Tato verze je vybavena externí anténou a zajišťuje několikanásobně delší dosah. Větší anténa by měla pomoci v citlivosti přijímače a tím by měl být i zkvalitněn přenos od vysílačů s nízkou kvalitou signálu. Dalším významným prvkem je integrovaný obvod realizující převod sběrnice UART na USB. Je jím obvod FT231X, jehož funkce jsou již zmíněny v sekci 1.6.1. Zapojení integrovaného obvodu bylo inspirováno katalogovým listem, v němž se nachází doporučené zapojení pro tento účel. K mikrokontroléru jsou ještě připojeny dvě luminiscenční diody. D1 signalizuje příjem nové zprávy ze serveru a D2 signalizuje přijetí nového paketu z komunikačního modulu. K oběma diodám jsou sériově připojeny rezistory R1 a R2 s hodnotou 150 Ω , které snižují úroveň proudu protékajícího diodami. Dalším významným prvkem je stabilizátor LF33 (a jeho dva kondenzátory C6 a C7) snižující napětí na 3,3V. Je zde umístěn kvůli modulu Nrf24l01+ jehož napájecí napětí smí být nanejvýše 3,6V. Přímo u napájecích pinů modulu by měly být umístěny kondenzátory C4 a C5 pro eliminaci poklesu napětí při proudových špičkách v době vysílání. Kondenzátory C1, C2, C3 s hodnotou 100nF jsou blokovacími kondenzátory a měly by být umístěny co nejbližší integrovaným obvodům. K mikrokontroléru je ještě připojen krystal s hodnotou 14,7456 MHz a kondenzátory C98 a C99. Krystal není nutné osazovat a je zde pouze pro případ, že by přenosová rychlost sběrnice UART 4800 Bd nebyla dostatečná. Pro vyšší rychlost již není vnitřní zdroj hodinových impulsů (8 MHz) správně dělitelný a tudíž nelze použít. Posledními nezmíněnými jsou rezistor R3 a kondenzátor C8 pro správný chod mikrokontroléru.

Při oživování zařízení byl odhalen problém s velkým přechodovým odporem na kontaktech komunikačního modulu a bylo za potřebí, přímo na modul připájet dodatečný kondenzátor, aby byla zajištěna stabilita napětí. Tento kondenzátor má hodnotu 100uF a není umístěn ve schématu, protože se nenachází na plošném spoji ale přímo na modulu. Bez tohoto kondenzátoru se modul při každém pokusu vyslat data restartoval. Tím nebyla data odeslána a vnitřní nastavení modulu bylo nastaveno na tovární hodnoty. Mikrokontrolér se o tomto jevu nedozvěděl, a tudíž neprovedl reinicializaci modulu. Modul tím pádem následně přijímal na frekvencích určených výrobou, předpokládal jiné komunikační rychlosti atp.

Program:

Mikrokontrolér nemá za úkol nic jiného než kontrolovat stav komunikačního modulu. V případě přijetí nové zprávy se v registru zvaném „status“ změní bit zvaný „data_ready“. V okamžiku nastavení tohoto bitu si mikrokontrolér začne vyčítat zprávu z přijímacího bufferu modulu. Data jsou následně přeposlána přes sběrnici UART do serveru. Jelikož je využívána periferie UART, tak byl pro tento účel vytvořen hlavičkový soubor `usart.h` a k tomu odpovídající `usart.c`. Dále zde byly využity knihovny vytvořené pro modul s teplotním čidlem na ovládání spi a pro komunikaci s komunikačním modulem Nrf24l01+.

Program je částečně připraven na budoucí rozšíření. Bylo zprovozněno přerušení od periferie UART navázané na příjem nových dat. Na toto místo přijde v budoucnu uložení přijatých dat a nastavení proměnné symbolizující, uložení nových dat k bezdrátovému odeslání. V hlavní smyčce se poté zkontroluje, zda-li nejsou připravena nová data a pokud ano, tak se odešlou. Není vhodné dát odesílací funkci přímo do přerušení, protože může nastat několik konfliktních stavů. Například pokud by přerušení přišlo v okamžiku vyčítání zprávy z modulu, tak by nedošlo ke korektnímu přečtení a mohlo by to způsobit následné škody v obslužném programu.

2.2.2 Ovládací program

Celý obslužný program byl napsán v jazyce Java. V době této práce byla nejaktuálnější verze Java 8. Tento jazyk byl vybrán kvůli tomu, že je objektový, multiplatformní a také kvůli zkušenostem spojeným s tímto jazykem. Objektivnost umožňuje vytvořit si třídu „Teploměr“, od které může být vytvořen libovolný počet instancí a nad všemi vytvořenými instancemi volat globální funkce, které vykonají nějaký sled úkonů. Tento přínos bude mít větší uplatnění

v budoucnu, kdy bude do celého systému přidána například kontrola světel a poté se jedním příkazem všechna světla zhasnou.

Struktura a samotný chod je následující. Na začátku hlavního programu se nejprve vyčtou data z konfiguračního souboru `Server.conf`. Ukázkový vzhled je na přiloženém CD ve složce `Server/Ridici_program`. Tento soubor musí být umístěn v aktuálním adresáři odkud je program spouštěn. Z tohoto souboru se vyčte port, na kterém se má komunikovat a následně se program pokusí otevřít port. Pokud bylo otevření úspěšné, program pokračuje dále. Pokud ovšem dojde k chybě při otvírání, program nahlásí chybu a vypíše seznam dostupných portů. Pokud se povedlo vytvořit instanci portu, tak se nad tímto portem vytvoří `EventListener`¹⁴, který zareaguje při každém přijatém znaku. Přijatá zpráva je předána jako parametr konstruktoru třídy `Device`, která symbolizuje obecné zařízení. Konstruktor zprávu zpracuje a určí, který typ zařízení to je. V případě, že s daným zařízením již bylo v minulosti komunikováno, tak se přistoupí k instanci daného zařízení. V opačném případě se vytvoří nová instance již konkrétního typu zařízení a uloží se do seznamu známých zařízení. Pokud je zařízením teplotní čidlo, vytvoří se instance třídy `Thermometer` a jako parametr konstruktoru je mu předána instance zařízení, která byla vytvořena při přijetí zprávy.

Třída `Device` sama o sobě nemá příliš mnoho funkcí. Má ovšem základ, který musí mít všechny zařízení v celém systému. Velmi důležitou metodou je metoda `doAction`. Nachází se ve třídě `Device` a musí být přepsána každým potomkem¹⁵. Je to z toho důvodu, že poté je tato metoda volána nad obecným zařízením uloženým v seznamu známých zařízení. Je tím umožněno mít seznam obecných zařízení a nad nimi volat tuto obecnou metodu. Metoda na přijetí znaku tím pádem nemusí znát totožnost konkrétního zařízení a zavolá jeho metodu `doAction`. Tato metoda v každém typu zařízení vykoná přesně to, co se od daného zařízení očekává. Pro teploměr je to například akce „zapsání teplot do databáze“.

Jednotlivé třídy zastupující reálné zařízení musí být tedy potomkem `Device`.

¹⁴ Funkce, která se zavolá v okamžiku události na kterou je navázána. Analogií je mikroprocesorové přerušení.

¹⁵ Třída využívající základní znak objektového

Pro názornost metoda `doAction` ve třídě `Thermometer` vykonává následující sled příkazů. Zobrazí teplotu v terminálu¹⁶. Dále se zjistí rozdíl dvou po sobě přijatých teplot. Pokud je rozdíl mezi nimi větší jak 10°C, tak jsou obě hodnoty prohlášeny za neplatné. Zjištění rozdílu je zde implementováno kvůli maximální eliminaci chyb ze strany čidla, rušení nebo jiného neočekávaného zdroje chyb. Pokud byl rozdíl teplot vyhodnocen jako reálný, vykoná se následující blok příkazů. Prvním je vytvoření průměru ze dvou posledních údajů a ten je poté uložen do databáze. Následně jsou smazána stará a již neplatná data z jednotlivých tabulek databáze a jako poslední se vygenerují soubory JSON se seznamem teplot a časů, které jsou následně staženy uživatelem při načtení internetových stránek.

Pro chod musí být nainstalována Java8, což se dá provést tímto postupem:

- 1) Stažení instalačního balíku ze stránek společnosti Oracle. V době psaní práce je tento balík dostupný na adrese

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

- 2) Následné rozbalení pomocí příkazu:

```
sudo tar zxvf jdk-8-linux-arm-vfp-hflt.tar.gz -C /opt
```

- 3) Nastavení Javy8 jako výchozí verze Javy pro spouštění aplikací pomocí:

```
sudo update-alternatives --install /usr/bin/javac javac /opt/jdk1.8.0/bin/javac 1
```

```
sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0/bin/java 1
```

```
sudo update-alternatives --config javac
```

```
sudo update-alternatives --config java
```

- 4) Na závěr je možné si zkontrolovat správnost pomocí příkazů:

```
java -version a javac -version
```

Databáze:

Pro uložení dat je využita databáze MySQL. Byly zde vytvořeny 3 základní tabulky pro ukládání teplot a jedna pro nastavení teplotních čidel. Tabulky pro ukládání teplot jsou identické, avšak každá pro jiný časový úsek. Podle časového úseku jsou i analogicky pojmenovány – *day*, *month* a *year*. Všechny tyto tabulky obsahují sloupec *id* pro přesnou identifikaci záznamu. Tento sloupec je nastaven jako primární klíč a zároveň je na něm nastaven `auto_increment`, což je funkce zajišťující automatické zvyšování hodnoty pro každý

¹⁶ Někdy také označováno jako konzole a v operačním systému Windows jako příkazový řádek.

nový prvek. Druhým sloupcem je identifikátor teplotního čidla. Dalším sloupcem je samotná teplota a posledním je čas.

Tabulka s nastavením se jmenuje *settings* a obsahuje informace o jednotlivých čidlech. Obsahuje 3 sloupce, kde prvním je *thermometer_id*. Tento identifikátor musí být stejný, jako je naprogramovaný identifikátor v konkrétním čidle. Zároveň je tento sloupec označen jako primární, takže není umožněno vytvořit dva záznamy se stejným identifikátorem. Druhým sloupcem je název čidla, který se bude zobrazovat na internetových stránkách a posledním sloupcem je čas obnovy. S posledním parametrem je počítáno do budoucna pro přesnější výpočet správnosti derivace dvou po sobě jdoucích vzorků.

Při vytváření této práce se vyskytl problém s rozdílností časového pásma databáze a aktuální časovou zónou. Pro správný chod je nutné mít korektně nastaveny časová pásma, protože jinak se v grafu zobrazují chybné časy. Přenastavení je nutné vykonat pod uživatelem s nejvyššími právy v databázovém systému a provede se v terminálu následujícími příkazy:

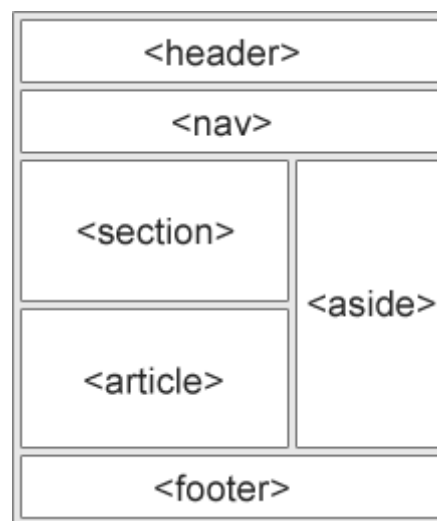
```
mysql -uroot -p          (pro uživatele root)
poté je nutné zadat heslo
SET GLOBAL time_zone = 'Europe/Prague';
exit
```

Na CD lze nalézt soubor s ukázkovou databází s názvem *meteobug*., který stačí pouze naimportovat do vytvořené databáze v databázovém systému a vše nastavit v konfiguračním souboru *Server.conf*. Je zapotřebí nastavit jméno databáze, uživatele + heslo pro přístup, a umístění databáze. Názvy parametrů pro nastavení jsou následující: *DATABASE*, *DATABASE_USER*, *DATABASE_PASSWORD* a *DATABASE_SERVER*.

V souboru *Server.conf* je nutné nastavit ještě důležitou položku *WEB_DIRECTORY*. Tato položka označuje cestu k adresáři internetových stránek. Bez nastavení této cesty nebude program fungovat a v případě nastavení špatné cesty budou soubory s aktuálními teplotami generovány do úplně jiných složek, a tudíž nebudou na internetu zobrazovány. Je nutné nastavit kořenový adresář, do kterého jsou nakopírovány všechny soubory z CD ze složky *Server/Internetove_stranky*.

2.2.3 Internetové stránky

Internetové stránky byly napsány v jazyce HTML5 jak již bylo zmíněno v části 1.6.3 . Velkým rozdílem oproti starším verzím HTML a HTML5 je přítomnost nových tagů, které umožňují přehlednější vzhled kódu, ze kterého se stránky v prohlížeči vykreslují. Tyto tagy byly přidány díky informacím získaným společnostmi Google, která při indexaci miliónů stránek napříč celým internetem zjistila velkou míru opakování identifikátorů tagů „div“ v podobných částech internetových stránek.



Obrázek 7: Nové tagy v HTML5¹⁷

Starým způsobem by se například hlavička stránky vytvářela následovně:

```
<div id="header"> meteobug </div>
```

Využití nových tagů vypadá v kódu následovně:

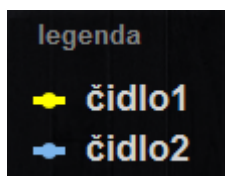
```
<header> meteobug </header>
```

A k tomu odpovídající CSS :

```
header {
  font-family: 'Lato', sans-serif;
  font-size: 11vw;
  height: 14vw;
}
```

¹⁷ http://www.w3schools.com/html/img_sem_elements.gif

Zmíněné tagy (viz *Obrázek 7*) nejsou jedinou novinkou v HTML5. Byly zde přidány tagy na přehrávání videa a audia (<video>, <audio>), geolokaci, vykreslování grafiky a mnoho dalšího. Jako jedna z využívaných novinek je také tag <canvas>, který je využíván pro kreslení grafiky. Tento tag vytvoří grafické plátno, do kterého je poté možné kreslit pomocí JavaScriptu. Ve starších verzích HTML je mnohem komplikovanější vytvořit podobnou grafiku. V této práci byl tento tag využit na velmi drobné znázornění barvy grafu v legendě (viz *Obrázek 8*).



Obrázek 8: Legenda z internetových stránek - znázornění použití tagu canvas.

Dalšími zajímavostmi při tvoření stránek bylo například implementování nestandardního fontu v části, kde se zobrazuje aktuální hodnota, plovoucí menu v případě posunutí pohledu do části, kde menu již není viditelné apod.

Grafy:

Nejmimozantnější částí internetových stránek jsou ovšem grafy. Jsou zde umístěny 3 grafy, na kterých se zobrazují hodnoty denní, měsíční a roční z každého čidla. Na každý z těchto grafů je možné se pohodlně přesunout kliknutím na daný typ v menu. Pro vykreslení byla použita JavaScriptová knihovna HighCharts. Pro vykreslení grafu je nutné si vytvořit objekt Graf, kterému je jako parametr předáno požadované nastavení a parametry. V nastaveních se například objevují prvky: zdroj dat pro vykreslení, barva, nastavení os, typ grafu a mnoho dalšího. V praxi to poté vypadá následovně:

```
var chart_day = new Highcharts.Chart(options_day);
```

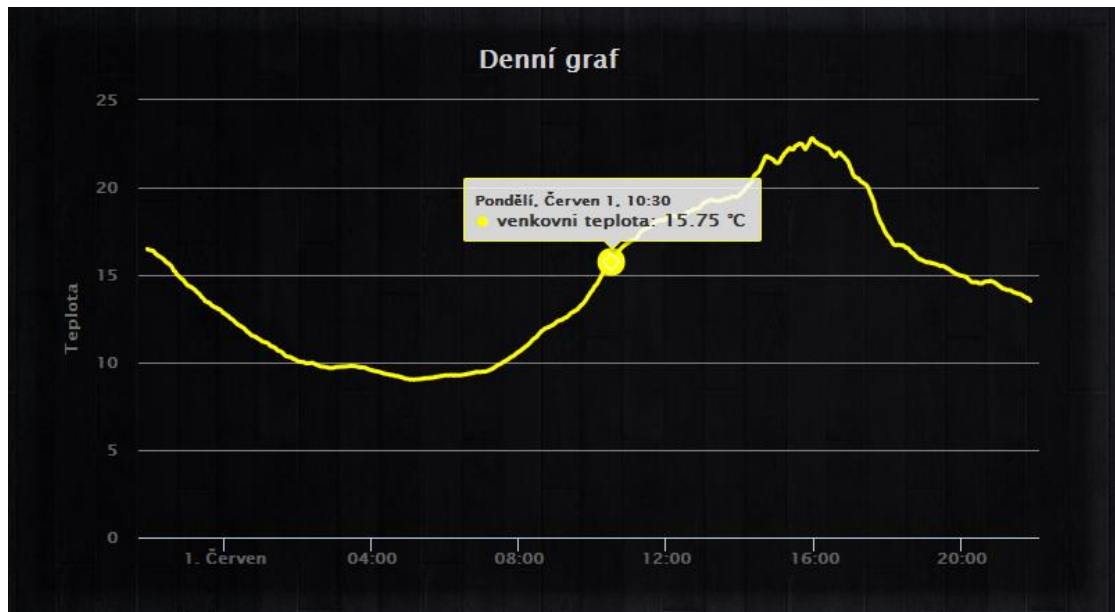
Roční graf se po základním načtení nenačítá z důvodu úspory přenesených dat. Je ovšem možné si ho zobrazit kliknutím na tlačítko „Načíst roční graf“.

Pro příjemný pocit uživatele byla velmi důležitá implementace zobrazení teploty na aktuální pozici kurzoru v grafu. Zajímavou informací totiž není pouze aktuální hodnota, ale jsou jimi i informace o teplotách z minulosti (viz *Obrázek 10*). Dalšími zajímavostmi jsou například: přibližování grafu tažením kurzoru, automatická aktualizace hodnot na pozadí,

vypnutí zobrazování jednotlivých čidel kliknutím na příslušné jméno čidla v legendě. Jako posledním prvkem pro zkvalitnění pocitu uživatele bylo vytvoření faviconu. Je to drobný obrázek zobrazující se v liště otevřených panelů v prohlížeči.



Obrázek 9: Favicon a titulek stránky.



Obrázek 10: Zobrazení grafu.

Zmenšení souborů:

Těsně před dokončením tvorby internetových stránek je vždy vhodné vytvořit zmenšenou verzi¹⁸ všech JavaScriptových souborů a souborů kaskádních stylů. V tomto případě se tedy jednalo o soubory: *highcharts.js*, *style.css*. Tento proces má velký význam pro snížení velikosti souborů, které se následně stahují ke koncovému uživateli. Zmenšení souborů je vykonáno smazáním mezer a speciálních znaků jako jsou odřádkování či tabulátor. Na rychlost vykonávání programu to nemá vliv. Naopak na rychlost načítání to má vliv markantní. Při pozdější aktualizaci zdrojových souborů je nutné vždy vytvořit novou zmenšenou verzi. V tabulce *Tab. 6* je znázorněno zmenšení velikosti použitých souborů.

¹⁸ V anglické terminologii je zmenšený soubor označován jako minify od čehož je odvozen i pozměněný název, který po zmenšení obsahuje slovo ".min.".

Tab. 6: Zmenšení souborů.

| | Původní velikost | Velikost po zmenšení |
|---------------|------------------|----------------------|
| highcharts.js | 488 kB | 175 kB |
| style.css | 4,1 kB | 2,1 kB |

Pro vygenerování je možné využít nějaký online generátor, program třetích stran nebo přímo doplněk do vývojového prostředí. Vzniklé soubory mají typicky upravený název. Soubory vzniklé se tedy jmenují *highcharts.min.js* a *style.min.css*.

Doména:

Jak již bylo zmíněno v konceptu, pro celý projekt byl vytvořen název meteobug a od toho se odvíjí i název domény, která byla pro tento účel zřízena. Původně bylo cílem získat národní doménu (meteobug.cz). Bohužel to ale není zdarma a každoroční náklady na provoz nejsou akceptovatelné. Stejně tak tomu bylo i s evropskou doménou (meteobug.eu), kde jsou náklady sice nižší, ale stále zatěžují majitele. Obě tyto domény byly v době tvorby projektu k dispozici. Nakonec byla tedy vybrána bezplatná doména, na které lze internetové stránky momentálně najít. Získaná doména tedy vypadá následovně: <http://meteobug.tk>. Na tuto doménu je navázána i aplikace pro Android, která si právě z této domény stahuje aktuální data.

2.3 Krabíčka

Celé zařízení bylo zapotřebí umístit do krabíčky odolné proti kapající vodě. Možností bylo koupit již hotovou krabíčku v některém z mnoha obchodů. Koupená krabíčka by ovšem nesplňovala veškeré požadavky, které jsou:

- odolnost proti kapající vodě
- větrací otvor pro zajištění shodné teploty uvnitř i vně
- otvor pro snadnou montáž pomocí hřebíku, vrutu či šroubu
- nenápadnou barvu, aby zařízení nebudilo na první pohled pozornost

Žádná z nalezených krabíček nesplňovala všechny požadavky zároveň. Vždy existoval alespoň jeden nesplněný bod. Tento fakt byl bohužel neakceptovatelný, a tudíž byla zvolena jiná možnost. Tou je nakreslit si krabíčku vlastní a vytisknout si jí na 3D tiskárně. Bylo ovšem těžké splnit požadavek odolnosti proti kapající vodě a zároveň větracího otvoru. Tento

problém nebylo jednoduché vyřešit a trval asi nejdéle z celého procesu vymýšlení krabičky. Nakonec bylo nalezeno zajímavé řešení. U některých nalezených krabiček vyráběných profesionálními výrobci bylo nalezeno, že je jasně udaná poloha, v jaké je zajištěna odolnost proti kapalně vodě. To bylo určité usnadnění, které bylo aplikováno i u této krabičky a tudíž je jasně dána taková poloha, že plocha větracího otvoru musí být ve spodní části krabičky rovnoběžně se zemským povrchem (viz *Přílohy*). Díky zvýšenému okraji je zajištěno nezatékání vody z vrchní a boční strany. Víčko krabičky chrání vnitřní prostor proti dešti ze třetí boční strany a má přesah, který zajistí nezatékání vody ze spodní strany. Zároveň má víčko na konci, u kterého se předpokládá odkapávání vody, ostrý roh, aby bylo zajištěno skutečné odkápnutí kapky.

Větrací otvor je sice potřeba pro shodnou teplotu ovzduší jak uvnitř, tak vně krabičky. Zároveň to usnadňuje rozšíření měření například i vzdušné vlhkosti či tlaku, které by bylo v uzavřené krabičce velmi komplikované. Bohužel to ovšem přináší i nevýhodu, že se do krabičky dostává drobný hmyz, který není úplně vhodným prostředím pro elektroniku. Na základě této nevýhody byla v krabičce (v oblasti větracího otvoru) zhotovena drážka pro zasunutí mřížky. Tato mřížka byla také nakreslena v programu SolidEdge (stejně tak i zbytek krabičky) a počítalo se také s tiskem na 3D tiskárně. Bohužel byly požadovány tak malé rozměry, že by byl výrobek velmi křehký a ve výsledku velmi těžko realizovatelný. Proto byla zvolena cesta kovového sítky, které má mnohem vyšší pevnost i menší otvory. Větrací otvor velmi napomůže i problému s kondenzací vody uvnitř krabičky.

Fakt, že krabička byla vytvořena přímo na míru a přesně dle požadavků umožnilo přidat v průběhu tvorby různé další vylepšení, které by splnilo jen malé množství krabiček. Je to například drážka pro přesné umístění plošného spoje, přesné rozměry vnitřku krabičky v závislosti na velikosti elektroniky nebo také otvor pro případné přichycení víčka, pokud by na jeho udržení nestačila jeho vlastní váha. Pro sundání víčka v předepsané poloze je nutné působit silou proti zemské přitažlivosti čímž je zajištěna základní ochrana proti samovolnému otevření.

Výkresy krabičky s okótovanými základními rozměry lze nalézt v příloze a veškeré potřebné výrobní podklady včetně krátkého videa znázorňujícího sestavení lze nalézt na přiloženém CD.

2.4 Aplikace pro Android

Aplikace pro mobilní telefon je v dnešní době základem pro úspěch celého projektu. Pro uživatele je složitější podívat se na internetové stránky. Byla vybrána platforma Android, která má v aktuálním roce (2015) stále více než 50-ti procentní zastoupení na trhu. Je mnohem jednodušší a pohodlnější kliknout na malou ikonku, díky které se otevře aplikace zobrazující teplotu, než vyhledávat stránku v internetovém prohlížeči. V základní verzi je vše realizováno pomocí klasického aplikačního rozhraní, které si stáhne soubor s poslední naměřenou teplotou a tu následně zobrazí.

Tvorba této aplikace má velký potenciál a bylo by zajímavé jí rozšířit o další vylepšení. Prvním vylepšením by mohlo být stažení JSONu s nastavením a zobrazení u teplot také název čidel. Uživateli by se poté zobrazilo několik hodnot i s popisem. Například „22,5°C - skleník, 24°C - terasa“. Druhým vylepšením je vytvoření widgetu (plovoucí okénko přímo na ploše každého Android zařízení), kde by se neustále zobrazovala aktuální teplota. Dalšími vylepšeními jsou možnost upozornění při překročení nějaké kritické úrovně nebo například zobrazování grafů teplot. Posledním a nejdůležitějším rozšířením pro tuto aplikaci je migrace i na jiné platformy, jejichž uživatelé by mohli mít rovněž zájem sledovat teplotu pomocí mobilní aplikace.

3 Další možné vylepšení

Vylepšení, kterými je možné zlepšit tento projekt je samozřejmě obrovské množství. Dále jsou zmíněny jen základní vylepšení, o kterých bylo v průběhu tvorby nějak uvažováno, že by se daly do projektu zapracovat již při tvorbě nebo v budoucnu.

3.1 Měření tlaku

Prvním vylepšením je měření atmosférického tlaku, ze kterého se dá následně předpovědět déšť. Tato část byla zprvu považována za automatickou součást elektroniky, ale s postupem času to bylo spíše posouzeno jako součást, která bohužel odebírá příliš velké množství energie ve srovnání s tím, kolik zajímavých informací to přinese. Je to samozřejmě velmi subjektivní pohled, protože pro někoho to může být nutná součást meteostanice. V základní verzi to ovšem není zapracováno. Případné rozšíření o tuto funkci by nemělo být příliš složité, protože je na to připravena jak krabička, tak byl zvolen mikrokontrolér

s dostatečným počtem potřebných periférií a na plošném spoji by se dozajista také našlo místo. Uvažovanou součástí pro měření atmosférického tlaku je BMP-180.

3.2 Windows aplikace

Dalším zpříjemněním uživatelského dojmu z aplikace je rozšíření pro Windows. Cílová skupina verzí operačních systémů je od Windows 7 po novější. V případě Windows 7 by se jednalo pravděpodobně o tzv. „Windows gadget“ což je neustále otevřená aplikace na uživatelské ploše. Nejpoužívanějším gadgetem jsou hodiny a zobrazení výkonu počítače. Pro novější verze operačního systému Windows by bylo zajímavější vytvořit aplikaci schopnou pracovat hlavně v podobě dlaždice. Je to zajímavý směr i díky faktu, že by uživatel měl být v budoucnu schopen používat stejnou aplikaci jak v mobilním zařízení s operačním systémem Windows Phone, tak na klasické verzi pro stolní počítače. Tím by odpadl vývoj dvou aplikací a ušetřený čas by se mohl věnovat vylepšení společné aplikace.

3.3 Domácí stanice zobrazující teplotu a denní teplotní extrémy

Jako dalším zvažovaným vylepšením celého projektu je domácí stanice, která by zobrazovala aktuální teplotu a extrémní teploty za posledních 24 hodin. První zvažovanou možností bylo jednoduchý zobrazovač pomocí sedmi-segmentových zobrazovačů pro zobrazování pouze aktuální teploty. Po delší úvaze bylo ovšem toto možné vylepšení posunuto na vyšší úroveň a byl dodán požadavek teplotních extrémů, popřípadě denního grafu. Na zobrazení tolika informací je již zapotřebí použít větší display.

Jako nejsnazší cesta se momentálně zdá být doladění aplikace pro Android a zobrazování všech údajů na tabletu.

3.4 Generování polynomu

Toto vylepšení bude možná nutné aplikovat v případě velké zátěže serveru při generování souborů s hodnotami. Druhým okamžikem, kdy bude vyžadováno toto vylepšení je v okamžiku, kdy soubory JSON budou neúnosně velké a uživatel by tak musel dlouze čekat na stažení a zobrazení. Uživatel by si ve výsledku nestáhl seznam naměřených teplot, ale kořeny polynomu popisující stejnou křivku. Generování polynomů je také poměrně náročným výpočtem, ale pokud bude vygenerován s dostatečnou přesností, tak určuje i směr průběhu

teplot. Tím pádem díky tomu můžeme předpovědět teplotu na pár minut dopředu a díky tomu stačí, když se polynom vygeneruje pouze několikrát do hodiny.

Zjištění kořenů polynomu z tabulky hodnot je možné řešit Gaussovou eliminací matic.

4 Závěr

Hlavní myšlenkou této bakalářské práce bylo vybrat vhodné součástky a navrhnout zařízení pro bezdrátové měření teploty. Bezdrátový přenos byl zvolen kvůli snadnému umístění na jakémkoliv místě v dosahu přijímače a kvůli eliminaci tepelného ovlivnění jinými objekty. Navrhnuté zapojení bylo následně naprogramováno a odzkoušeno. Na straně přijímače bylo vytvořeno přijímací zařízení, které data přepośle do serveru přes rozhraní USB. V serveru jsou data zpracována a uložena do databáze.

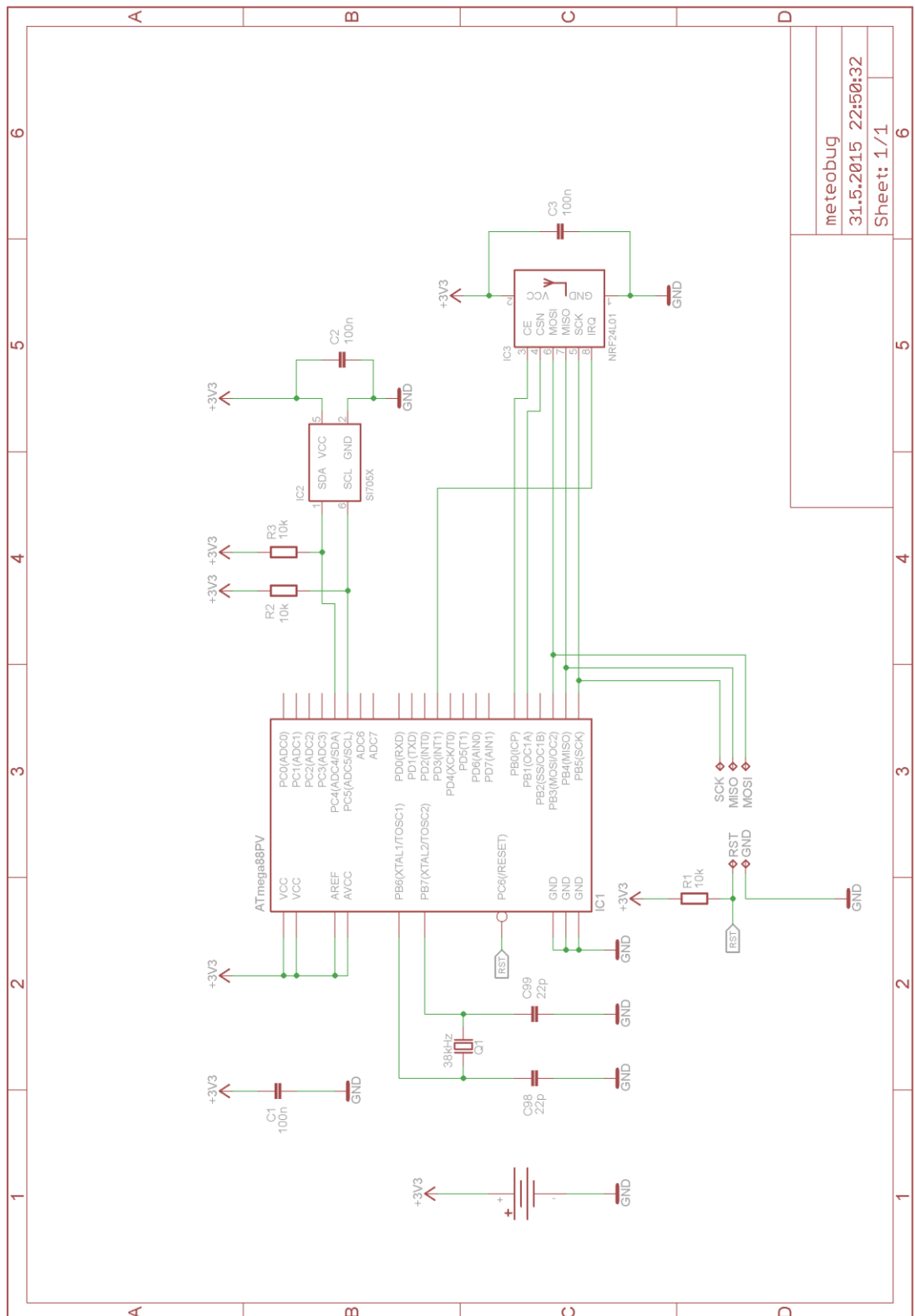
Pro uživatele bylo vytvořeno několik způsobů jak si změřené teploty zobrazit. Byly vytvořeny internetové stránky pro zobrazení aktuálních teplot i teplot z minulosti v podobě grafů. Druhým způsobem je aplikace pro mobilní operační systém Android, kde si uživatel může zobrazit aktuální teplotu.

Cílem projektu bylo, aby zařízení mělo co nejmenší proudový odběr kvůli dlouhé životnosti bez nutnosti časté výměny akumulátoru. Proudový odběr byl změřen a následně z toho byla vypočítána předpokládaná doba funkce bez výměny akumulátoru na 16 let. Tato doba je platná pro laboratorní podmínky, avšak i přesto byl tento výsledek shledán za vyhovující.

Seznam literatury a informačních zdrojů

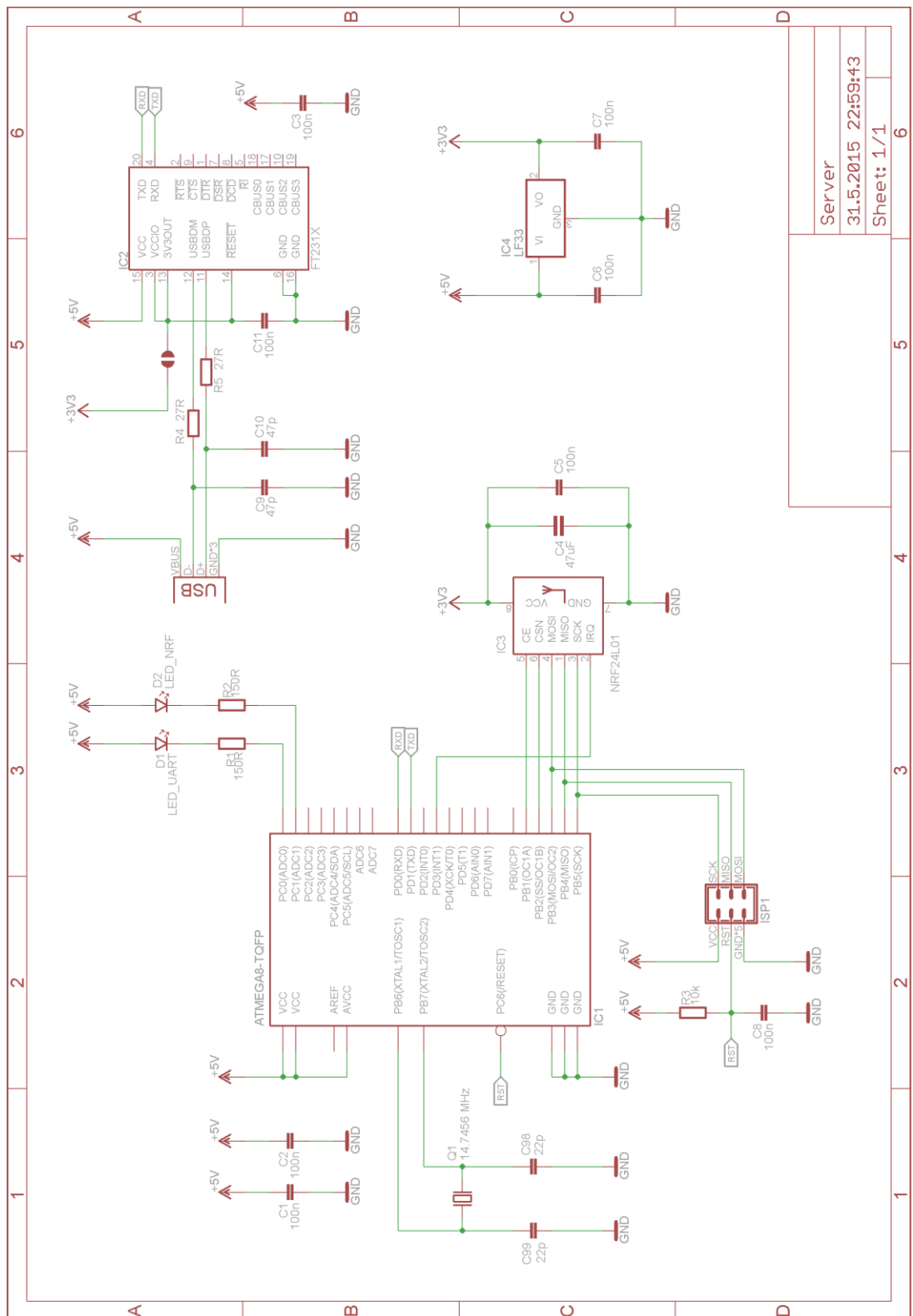
- [1] Atmel. Datasheet ATmega8A. [online]. Poslední změna 02/2013. [Cit 2.6.2015]. Dostupné z: http://www.atmel.com/images/atmel-8159-8-bit-avr-microcontroller-atmega8a_datasheet.pdf
- [2] Atmel. Datasheet ATtiny43U. [online]. Poslední změna 02/2012. [Cit 2.6.2015]. Dostupné z: <http://www.atmel.com/images/doc8048.pdf>
- [3] Atmel. Datasheet ATmega88PV. [online]. Poslední změna 06/2011. [Cit 2.6.2015]. Dostupné z: <http://www.atmel.com/images/8025s.pdf>
- [4] Atmel. Datasheet ATXmega16A4. [online]. Poslední změna 06/2013. [Cit 2.6.2015]. Dostupné z: http://www.atmel.com/Images/Atmel-8069-8-and-16-bit-AVR-AMEGA-A4-Microcontrollers_Datasheet.pdf
- [5] Atmel. Datasheet SAMC21E15A. [online]. Poslední změna 05/2015. [Cit 2.6.2015]. Dostupné z: http://www.atmel.com/Images/Atmel-42365-SAMC21_Datasheet.pdf
- [6] Analog Devices. Datasheet AD7415. [online]. Verze F. [Cit 2.6.2015]. Dostupné z: http://www.analog.com/media/en/technical-documentation/datasheets/AD7414_7415.pdf
- [7] Texas Instruments. Datasheet TMP101. [online]. Poslední změna 03/2015. [Cit 2.6.2015]. Dostupné z: <http://www.ti.com/lit/ds/symlink/tmp101.pdf>
- [8] Sensirion. Datasheet SHT25. [online]. Poslední změna 05/2014. [Cit 2.6.2015]. Dostupné z: http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT25_Datasheet_V3.pdf
- [9] Silicon Labs. Datasheet Si7053. [online]. Poslední změna 02/2015. [Cit 2.6.2015]. Dostupné z: <https://www.silabs.com/Support%20Documents/TechnicalDocs/Si7050-3-4-5-A20.pdf>
- [10] FTDI chip. Datasheet FT231X. [online]. Poslední změna 02/2013. [Cit 2.6.2015]. Dostupné z: http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT231X.pdf
- [11] Energizer. Datasheet CR2030. [online]. [Cit 2.6.2015]. Dostupné z: <http://data.energizer.com/PDFs/cr2032.pdf>
- [12] Panasonic. Eneloop. [online]. [Cit 2.6.2015]. Dostupné z: <http://panasonic.net/energy/battery/us/eneloop/lineup/index.html>
- [13] Saft. Datasheet LS14250. [online]. Poslední změna 09/2009. [Cit 2.6.2015]. Dostupné z: http://www.saftbatteries.com/force_download/LS14250.pdf
- [14] Nordic semiconductor. Datasheet Nrf24l01+. [online]. Verze: 2. [Cit 2.6.2015]. Dostupné z: http://www.nordicsemi.com/eng/nordic/download_resource/8041/1/57152017
- [15] Microchip. Datasheet Mrf89xa. [online]. Poslední změna 11/2011. [Cit 2.6.2015]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/70622C.pdf>
- [16] Microchip. Datasheet Mrf24j40. [online]. Poslední změna 08/2010. [Cit 2.6.2015]. Dostupné z: <http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>

Přílohy



meteobug
31.5.2015 22:50:32
Sheet: 1/1

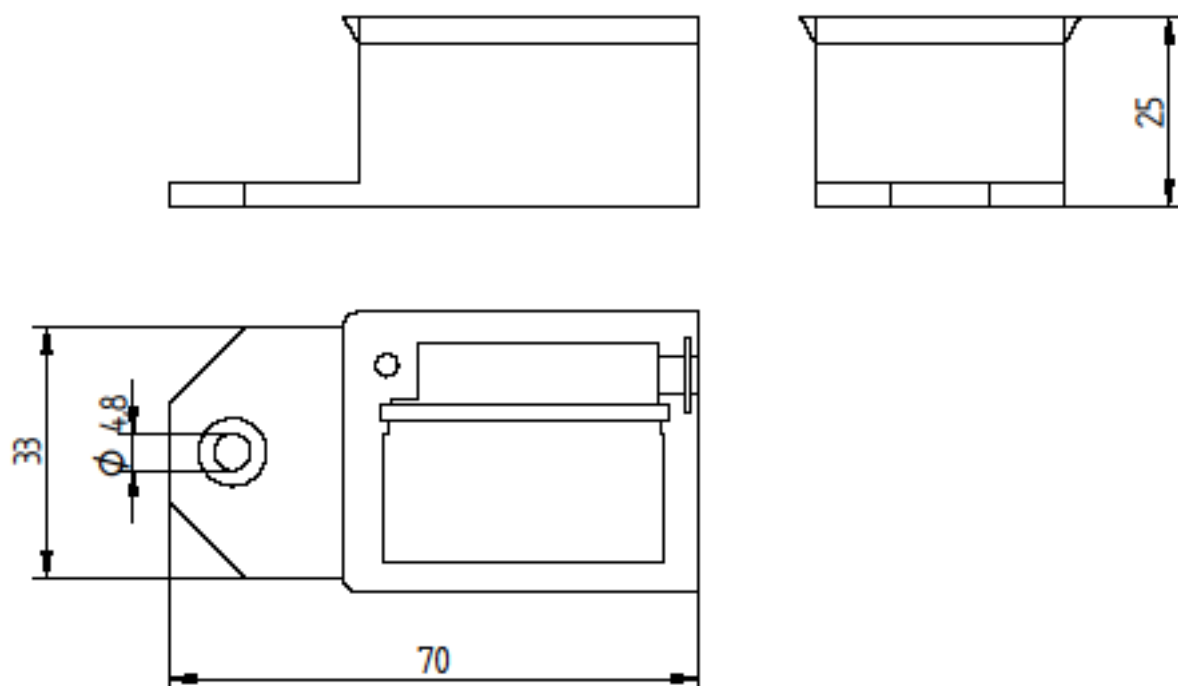
Obrázek 11: Schéma zařízení měřící teplotu.



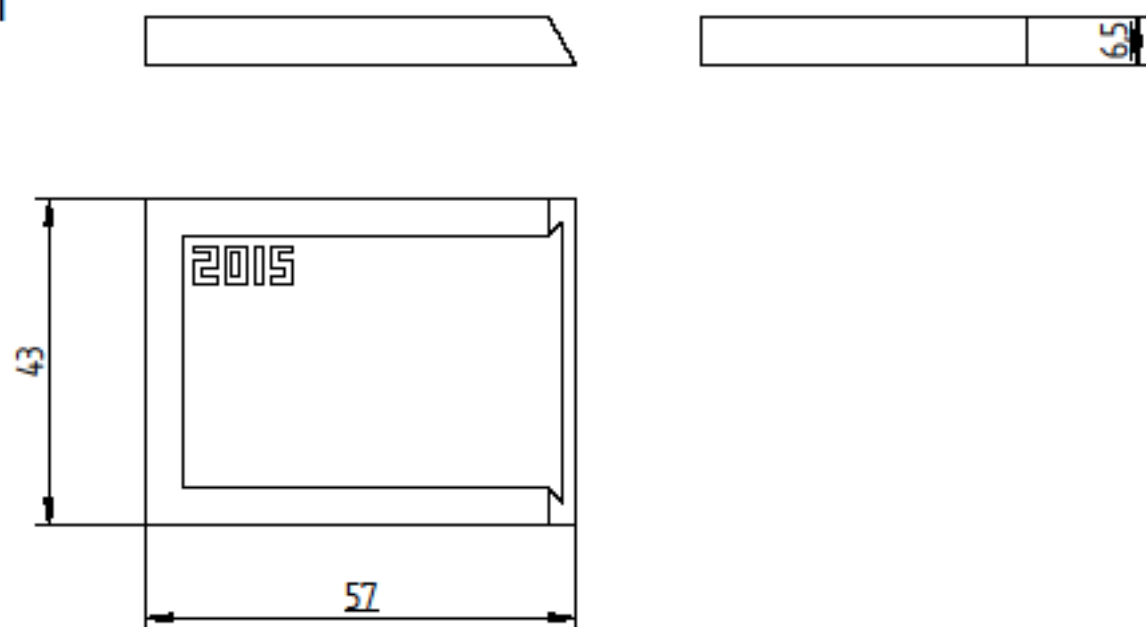
| |
|--------------------|
| Server |
| 31.5.2015 22:59:43 |
| Sheet: 1/1 |

Obrázek 12: Schéma serverového přijímače.

Krabička 1:1



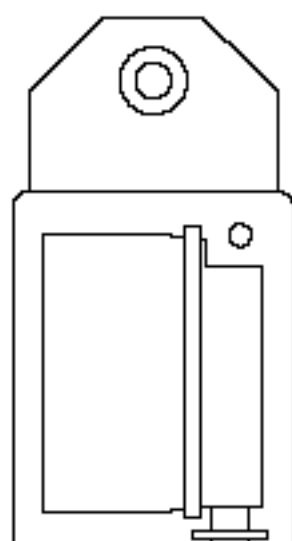
Víko 1:1



| | | | | |
|-------------------------|-------|----------|----------------------------|------------|
| | JMÉNO | DATUM | Solid Edge | |
| KRESLIL | MJ | 06/01/15 | NÁZEV | meteobug |
| KONTROLOVAL | | | NÁZEV SOUBORU: Vykres1.dft | |
| KÓTY JSOU V MILIMETRECH | | | MĚŘÍTKO: | HMOTNOST: |
| | | | | LIST 1 Z 2 |

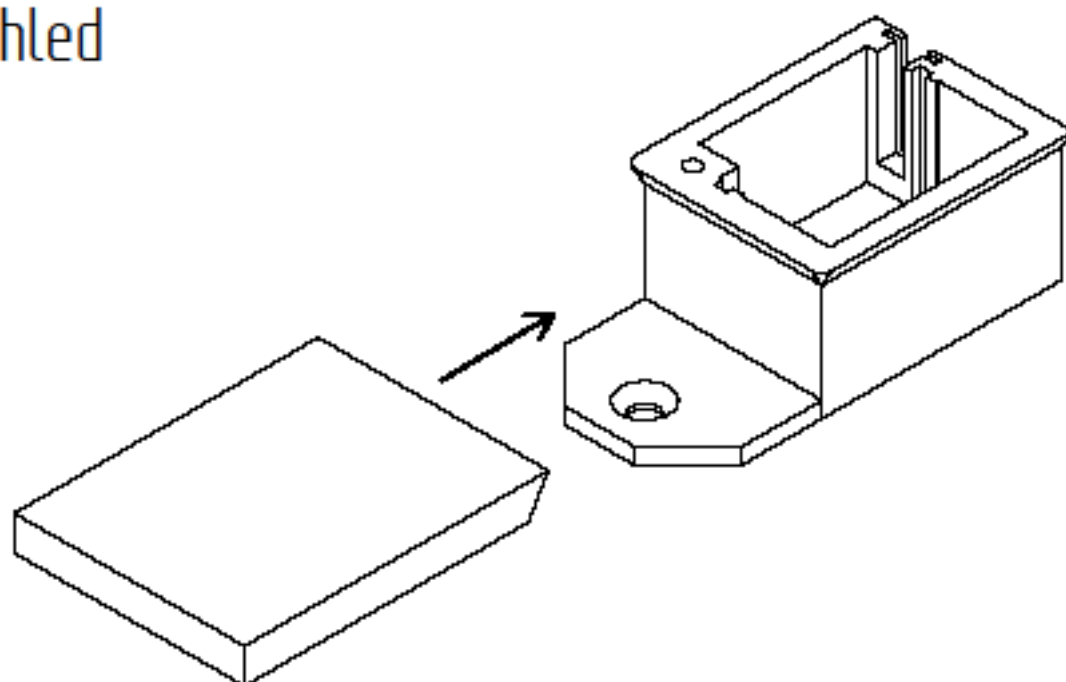
SOLID EDGE ACADEMIC COPY

Předepsaná poloha



Zemský povrch

3D náhled



| | | | | |
|-------------------------|-------|----------|----------------------------|------------|
| | JMÉNO | DATUM | Solid Edge | |
| KRESLIL | MJ | 06/01/15 | NÁZEV | meteobug |
| KONTROLOVAL | | | NÁZEV SOUBORU: Vykres1.dft | |
| KÓTY JSOU V MILIMETRECH | | | MĚŘÍTKO: | HMOTNOST: |
| | | | | LIST 2 Z 2 |

SOLID EDGE ACADEMIC COPY