

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra telekomunikační techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Jan Říha**

Studijní program: Komunikace, multimédia a elektronika
Obor: Síťové a informační technologie

Název tématu: **Měřicí systém s bezdrátovým přenosem dat pomocí Wi-Fi modulu**

Pokyny pro vypracování:

1. Seznamte se s problematikou programování mikrokontrolérů pro účely obsluhy digitálních a analogových senzorů. Seznamte se s problematikou dostupných Wi-Fi modulů pro přenos sensorových dat a jejich následné zpracování.
2. Na základě získaných poznatků navrhnete a realizujete zařízení pro bezdrátový přenos sensorových dat jako je teplota, vlhkost, apod. K bezdrátovému přenosu použijte vhodný Wi-Fi modul a navrhnete webovou aplikaci pro uchování a prezentaci měřených dat.
3. Sestavte měřicí a demonstrační pracoviště pro ověření funkčnosti. Proveďte měření a kriticky zhodnoťte parametry vašeho řešení. Navrhnete další možná vylepšení.

Seznam odborné literatury:

- [1] Mann, B.: *C pro mikrokontroléry* (1.vydání). Praha 2003, Nakladatelství BEN - technická literatura, ISBN: 80-7300-077-6.
- [2] Hac, A.: *Wireless sensor network designs*. John Wiley & Sons, Ltd, 2003. ISBN: 0-470-86736-1.
- [3] Malý, M.: *ESP8266 – wifi za pár korun (doslova)*. retrocip.cz, říjen 2014, Dostupné na: <http://retrocip.cz/esp8266-wifi-za-par-korun-doslova/> [on-line]
- [4] Vedral, J., Fischer, J.: *Elektronické obvody pro měřicí techniku*. Vydavatelství ČVUT, Praha 2004, ISBN: 80-01-02966-2.

Vedoucí: Ing. Adam Bouřa, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017



prof. Ing. Boris Šimák, CSc.
vedoucí katedry

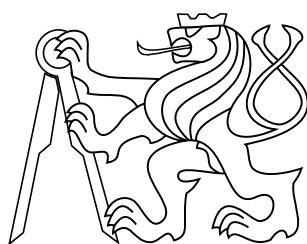
prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 21. 12. 2015

bakalářská práce

Měřicí systém s bezdrátovým přenosem dat pomocí Wi-Fi modulu

Jan Říha



Květen 2016

Ing. Adam Bouřa, PhD.

České vysoké učení technické v Praze
Fakulta elektrotechnická, katedra telekomunikační techniky

Poděkování

Děkuji panu Ing. Adamu Bouřovi, Ph.D. za ochotu, pomoc a rady při zodpovězení mých dotazů spojených s realizací projektu bakalářské práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Abstrakt

Cílem práce bylo vytvořit aplikaci pro měření meteorologických dat, jako například teploty, vlhkosti, či prachových částic. Zařízení se sestává ze senzorů, přípravku Arduino NANO, Wi-Fi modulu, displeje a webové aplikace na zobrazení naměřených dat. Naměřená data je možné zobrazit v grafu.

Klíčová slova

Měřicí systém; Senzor; Teploměr; Arduino; ESP8266; Wi-Fi; Aplikace

Abstrakt

The goal of the project was to create an application for measuring meteorological data, such as temperature, humidity, or dust particles. The device is composed of sensor, Arduino NANO chip, Wi-Fi module, display and a website used for displaying the measured data. The measured data can be shown as a graph.

Keywords

Measuring system; Sensor; Thermometer; Arduino; ESP8266; Wi-Fi; Application

Obsah

1. Úvod	1
2. Motivace	2
3. Cíl práce	3
4. Teoretický rozbor	4
4.1. Pevné částice	4
4.1.1. Zdroje pevných částic	4
Sekundární prašnost	4
Zdravotní vlivy	4
Vliv velikosti částic	5
Složení částic	5
Částice v České republice	5
4.2. Embedded systémy	5
4.2.1. Komponenty vestavěného systému	6
Hardware	6
Hlavní software aplikace	6
Operační systém	6
4.2.2. Klasifikace vestavěných systémů	6
Vestavěné systémy malého rozsahu	6
Vestavěné systémy středního rozsahu	6
Sofistikované vestavěné systémy	6
4.3. Arduino	6
4.3.1. Hlavní výhody	6
4.3.2. Arduino Hardware	7
4.3.3. Arduino Software	7
4.4. Displej	7
4.4.1. Komunikační sběrnice I ² C	7
4.5. Senzory	8
4.5.1. Digitální senzor	8
Komunikační sběrnice 1-Wire	8
4.5.2. Analogový senzor	8
4.6. Wi-Fi modul	9
4.6.1. Napájení	9
4.7. PHP, server, webová stránka	9
5. Návrh a realizace vlastního řešení	10
5.1. Zapojení	10
5.1.1. Displej	10
5.1.2. Digitální senzor	10
5.1.3. Analogový senzor	10
5.1.4. Wi-Fi modul	10
5.1.5. Návrh a tvorba tištěného spoje	10
5.2. Programování hardwarové části	11
5.2.1. Displej	11
5.2.2. Digitální senzor	13
5.2.3. Analogový senzor	13

5.2.4. Wi-Fi modul	13
Odesílání dat na server	14
5.3. Programování serverové části	15
5.3.1. Vytvoření tabulky	15
5.3.2. Vytvoření příkazů na ukládání dat	15
5.3.3. Webová stránka na zobrazení grafů	16
6. Testování	18
6.1. Bez PCB	18
6.2. S PCB	18
6.3. WEB	18
7. Návrh dalších vylepšení	20
8. Závěr	21
Přílohy	
A. CD	22
Literatura	23

Zkratky

Wi-Fi	Bezdrátová technologie pro připojení k internetu
kit	Vývojová deska či stavebnice
OS	Operační systém
PWM	Pulse-width-modulation; pulzně šířková modulace
A/D	Analogově-digitální
ID	Unikátní identifikační číslo
ROM	Read-Only-Memory; Paměť pouze pro čtení
I/O	Input/Output; Vstupně-výstupní
SSID	Název sítě Wi-Fi
PCB	Plošný spoj, deska plošných spojů
IR	Infra-Red; Infračervený

1. Úvod

V dnešní, rychle se rozvíjející společnosti je kvalita ovzduší často diskutované téma. Lidé se snaží méně zatěžovat životní prostředí a především ovzduší používáním ekologičtějších zařízení a k přepravě využívat ve městech městskou hromadnou dopravu. Výrobci automobilů instalují do výfukových potrubí různé komponenty na čištění výfukových splodin, jako například katalyzátory, či filtry pevných částic.

2. Motivace

Původně bylo zamýšleno vytvořit systém na snímání kvality ovzduší, konkrétně měření obsahu prachových částic v ovzduší. Lidé by tak pouhým zadáním adresy do internetového prohlížeče měli možnost zjistit čistotu ovzduší v okolí Prahy 4 - Krče. Podle grafu s daty za poslední dobu by lidé mohli snáze rozhodnout, kdy větrat, či sportovat, aby se jim lépe dýchalo. Po krátkém bádání však bylo zjištěno, že kvalitní snímač prachových částic je finančně velice nákladný, a proto byly pro demonstraci funkce systému zvoleny finančně dostupnější senzory teploty. Co se zapojení, či naprogramování týče, není v senzorech příliš velký rozdíl, rozdíl bude hlavně ve využitelnosti dat získávaných z takto modifikovaného projektu.

3. Cíl práce

Cílem praktické části je vytvoření zařízení pro snímání neelektrických, meteorologických veličin, např. teploty ovzduší a její následné zveřejňování na webu. Jedná se o jednoduché zapojení přípravku Arduino NANO, LCD panelu, digitálních a analogových senzorů měřících teplotu a Wi-Fi modulu, který zprostředkuje přenos dat na server. Přípravek načítá hodnoty z čidel a posílá je na internet. Součástí projektu je také PHP server, který načítá teploty z měřicího systému, hodnoty ukládá do tabulky a obsah tabulky zobrazuje v grafu. Graf by měl být snadno ovladatelný a mělo by být možné zobrazit teplotu z vybraného časového intervalu.

4. Teoretický rozbor

V části Teoretický rozbor jsou informace související s projektem a veškeré moduly, které systém obsahuje. Dále jsou zde popsány způsoby připojení a použití těchto periférií.

4.1. Pevné částice

Pevné částice jsou drobné částice miniaturních rozměrů s takovou hmotností, že mohou být unášeny vzduchem. Jejich zvýšená koncentrace může způsobovat vážné zdravotní komplikace. Podílí se na důležitých atmosferických dějích jako např. vznik vodních srážek a ovlivňují teplotní bilanci Země [1].

4.1.1. Zdroje pevných částic

Zdrojů pevných částic je mnoho, patří mezi ně výfukové splodiny diesellových automobilů bez katalyzátorů, otěry z brzdového obložení automobilů, tepelné elektrárny, těžba kamení, stavby, ve velké míře zemědělství a také přírodní vlivy, jako např. výbuch sopky, lesní požáry, apod. [2]

Sekundární prašnost

Na znečištění se také podílí tzv. sekundární prašnost. Jedná se o proces zvíření prachových částic, které se již na zemském povrchu usadily a jsou znovu rozvířeny například díky automobilismu. Sekundární prašnost může být hlavní příčinou přítomnosti pevných částic ve městech. [3]

Na vzniku sekundární prašnosti se podílí pohyb stavebních automobilů po nezpevněných cestách, těžba, či provoz cementáren a skládek. Dále se na vzniku podílí přírodní vlivy, především vítr. [3, 4]

Sekundární prašnost úzce souvisí s vlhkostí povrchu, na kterém je prach usazen. S rostoucí vlhkostí klesají předpoklady k jejímu vzniku. Velice pomáhá kropení komunikací zejména v letních měsících. [3]

Zdravotní vlivy

Vdechování částic má negativní následky především na kardiovaskulární a plicní systém. [1] Účinek částic na lidský organismus souvisí s dobou, jakou je organismus částicím vystaven.

Při krátkodobé expozici může docházet k zánětlivým onemocněním plic, častějším návštěvám nemocnice a vyšší spotřebě léků. [5]

Při dlouhodobé expozici dochází ke snížení funkčnosti plic, chorobám dolních cest dýchacích, chronickému onemocnění plic a tím zkrácení předpokládané délky života.

Vliv velikosti částic

Podle toho, jak jsou částice velké se odvíjí, jak hluboko do organismu jsou schopné proniknout. Větší částice se zachytí na chloupkách v nose a nezpůsobí větší komplikace [1]. Částice menší, než $10\ \mu\text{m}$ pronikají do dolních cest dýchacích [6]. Menší, než $2,5\ \mu\text{m}$ se mohou usazovat v průduškách, menší, než $1\ \mu\text{m}$ mohou pronikat do plicních sklípků nebo až do krve [7].

Složení částic

Pevné částice často obsahují popílek, saze a horniny. Mohou obsahovat polyaromatické uhlovodíky, polychlorované bifenyly, pesticidy a těžké kovy [8]. Složení částic se liší podle způsobu jejich vzniku [4].

Částice v České republice

Holandská agentura pro hodnocení životního prostředí provedla porovnání ovzduší asi ve třiceti velkých evropských městech a z hlediska znečištění prachem vyhodnotila jako nejhorší město Prahu [9].

Limity koncentrace byly v roce 2008 v Česku překračovány v oblastech, kde žije přibližně 15 % obyvatel. Nejvyšší koncentrace byla v průměru naměřena na Ostravsku a Karvinsku [10].

4.2. Embedded systémy

Vestavěné systémy jsou systémy, které jsou součástí většiny zařízení. Tyto systémy musí plnit svojí úlohu bez zásahu člověka po delší dobu. To je důvod, proč se při jejich výrobě dbá na nižší energetickou spotřebu a především spolehlivost. Vestavěné systémy jsou běžnou součástí života každého z nás bez toho, abychom jsme si to uvědomovali. Můžeme je nalézt například v malých domácích elektrospotřebičích (mikrovlňná trouba, zvukové systémy, televize, apod.), jsou součástí vybavení automobilů, ale také jsou využívány velkými korporacemi v síťových systémech a dalších produktech. Příkladů vestavěných systémů je nespočetné množství.

V posledních letech došlo k velkému pokroku vestavěných systémů, které jsou díky technickému pokroku stále modernější a sofistikovanější. Tato skutečnost je zapříčiněna vyšší úrovní zpracování hardwarových prvků a čipů.

Vestavěný systém je jednoúčelový počítačový systém cíleně vytvořený pro konkrétní aplikaci nebo produkt. Vestavěný systém můžeme definovat jako část počítačového hardwaru s vlastním softwarem, který je vložen v jednom z jeho nejdůležitějších komponent. Vestavěný systém však může být taktéž nezávislý systém anebo pouhá součást jiného většího systému. Jedná se o software, který je zpravidla vložený do paměti pouze pro čtení a nepotřebuje žádnou další paměť.

Vestavěný systém má software navržený tak, aby zohledňoval tři hlavní omezení:

- Dostupnou paměť systému,
- dostupnou rychlost procesoru,
- potřebu limitovat ztrátový výkon.

4.2.1. Komponenty vestavěného systému

Hardware

Hlavní software aplikace

Vykonává současně řadu úloh

Operační systém

Běží v reálném čase (Real Time Operating System - RTOS), dohlíží na software aplikace a zabezpečuje kontrolu toho, jestli systém pracuje správně a podle stanoveného časového plánu. RTOS definuje jak systém funguje, dohlíží na následnost úloh, které vykonává systém a plánuje práci systému tak, aby předešel jakémukoliv zpoždění.

4.2.2. Klasifikace vestavěných systémů

Vestavěné systémy malého rozsahu

Systém navržený jedním, osmi nebo šestnácti bitovým mikroprocesorem, má hardware a software nízké složitosti.

Vestavěné systémy středního rozsahu

Systém navržený více šestnácti nebo třiceti dvěma bitovými mikroprocesory, má hardware a software střední složitosti.

Sofistikované vestavěné systémy

Systém s výraznou složitostí hardwaru a softwaru, nezbytnou součástí je konfigurovatelný procesor.

4.3. Arduino

Arduino je otevřená elektronická platforma, založená na uživatelsky jednoduchém hardware a software. Jedná se o jednoduchý kit, který obsahuje 30 vstupně-výstupních konektorů, ke kterým lze připojit například senzory teploty, vlhkosti, vzdálenosti nebo dorazové spínače. Dále např.: LED diody, LC displeje, krokové motorky, relé apod. Lze jím tedy nakonfigurovat nepřeberné množství projektů od měřicích stanic přes 3D tiskárny až po inteligentní budovy.

4.3.1. Hlavní výhody

- Jednoduché programování,
- jednoduché zapojení,
- nízká cena oproti jiným kitům,
- velké množství návodů,
- uživatelská komunita,
- platformní nezávislost (Windows/Linux/MacOS/...).

Arduino se vyrábí v několika modelech. Pro tento projekt byl použit model Arduino NANO.

4.3.2. Arduino Hardware

Arduino je levný, robustní vývojový kit založený na mikroprocesoru ATmega328. Má třicet digitálních vstupně-výstupních pinů (z toho osm s podporou PWM) a šest analogových vstupů. Návrhy plošných spojů jsou k dispozici pod licencí Attribution ShareAlike 2.5.

Celý Arduino kit pracuje na napěťové úrovni 5 V. [11]

4.3.3. Arduino Software

Arduino je programováno jednoduchým programovacím jazykem (podobný Wiring), který obsahuje knihovny k většině zařízení a je tak velice jednoduché připravit naprogramovat. Naprogramovaný kód lze přímo z prostředí nahrát na vývojovou desku. Je k dispozici ve verzích pro Windows, Mac OS X a Linux. [12]

4.4. Displej

V tomto projektu byl použit univerzální, modře podsvícený displej, který má 16 znaků v každé ze dvou řádek. Displej je vybaven převodníkem na sběrnici I²C, který je zapojen pouhými dvěma konektory. Tím je zredukován počet použitých výstupů z kitu, aby jich zbylo dostatečné množství pro další periferie. Displej je napájen napětím 5 V, které je dodáváno Arduino kitem.

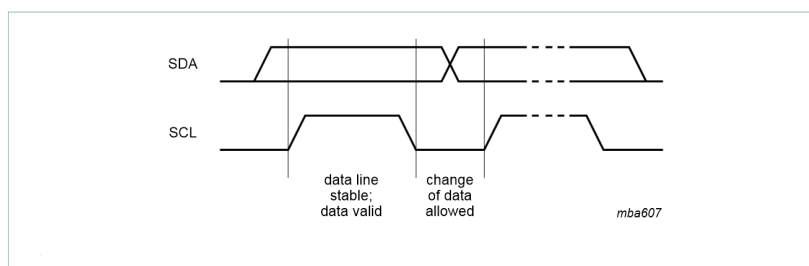
4.4.1. Komunikační sběrnice I²C

I²C (nesprávně I2C) je sběrnice navržena firmou Philips, která pomocí 2 drátů (SDA-serial data line a SCL-serial clock line) umožňuje připojit několik zařízení. Sériová, 8-bitová, obousměrná datová komunikace může dosahovat rychlosti až 100 kbit/s ve Standard-módu, až 400 kbit/s ve Fast-módu, až 1 Mbit/s ve Fast-módu Plus (Fm+), nebo až 3,4 Mbit/s v High-speed módu. Ultra Fast-mód je jednosměrný mód s rychlostí přenosu dat rychlostí až 5 Mbit/s. [13]

Příklad komunikace po I²C sběrnici:

Stavy START a STOP

Všechny přenosy začínají stavem START (S) a jsou ukončeny stavem STOP (P) (Obrázek 1).



Obrázek 1. Datový přenos na I²C sběrnici

HIGH -> LOW přechod na SDA lince když SCL je HIGH definuje stav START.
LOW -> HIGH přechod na SDA lince když SCL je HIGH definuje stav STOP.

4.5. Senzory

V projektu bylo původně zamýšleno použití jednoho senzoru teplotního a druhého na snímání prachových částic. Po zjištění, že kvalitní snímače prachových částic jsou velice nákladné, bylo od tohoto typu senzoru upuštěno. Nyní je připojen jeden digitální snímač teploty a druhý analogový. Jakýkoli další senzor je nadále možno kdykoli připojit a použít na snímání dat. V následujících podkapitolách jsou popsány použité senzory.

4.5.1. Digitální senzor

Digitální teplotní senzor je teplotní snímač, který rovnou převádí analogové informace ze samotného snímání do digitální formy. Zde byl konkrétně použit digitální senzor firmy Dallas DS18B20. Do Arduina tedy tečou přesné informace v digitální podobě po další sběrnici - 1-Wire.

Komunikační sběrnice 1-Wire

1-Wire [14] je - jak už vypovídá její název - sběrnice komunikující pouze jedinou linkou. Jedná se o obousměrný, sériový signalizační protokol, který napájí a ovládá zařízení přes jedinou linku na větší vzdálenost, než I²C. Komunikace probíhá na dvou základních rychlostech:

- Standardní 15,4 kbit/s,
- přetížené 125 kbit/s.

Každé zařízení dostává při výrobě přiřazené, nezměnitelné ID. Komunikace po sběrnici probíhá na třech úrovních:

- Restart/synchronizace zařízení,
- výběr zařízení-příkaz na úrovni ROM,
- funkce zařízení-příkaz na úrovni zařízení.

4.5.2. Analogový senzor

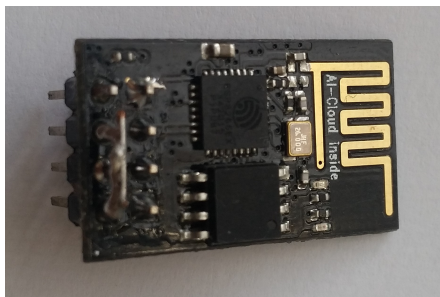
Dalším použitým senzorem je analogový, jehož výstupem jsou data v analogové podobě. Je tedy potřeba použít ještě A/D převodník, kterým bude analogová hodnota převedena do digitální podoby. Existuje zde tak závislost na použitém napájení a také na použitém A/D převodníku. Zde byl použit vestavěný A/D převodník obsažený přímo ve vývojovém kitu Arduino. Výstupem převodníku je celé číslo v rozmezí od 0 do 1023. Toto číslo je potřeba zkalibrovat pomocí prvního, přesného digitálního senzoru. Postup se nazývá proložení přímkou a je to postup, kdy se obě čidla umístí do stejného prostředí se stejnou teplotou a odečte se hodnota na každém z čidel. Poté jsou senzory spolu umístěny do jiného prostředí o jiné teplotě a znovu se provede odečet hodnot teploty. Z těchto čtyř údajů je spočítán výraz, který uvádí převod mezi digitální a analogovou hodnotou ze senzorů. V tomto konkrétním případě se jedná o výraz:

$$\text{Digitální hodnota} = (\text{Analogová hodnota} \times 0,404375) - 224,441$$

Model použitého senzoru je LM335 TO92 od výrobce STMicroelectronics.

4.6. Wi-Fi modul

Dále byl použit Wi-Fi modul s označením ESP8266-01. (Dále jen Wi-Fi modul nebo ESP) (Obrázek 2) Jedná se o velice univerzální zařízení, které je schopno připojit se k Wi-Fi síti, může být i hot-spotem, či serverem. Také podporuje šifrované, zabezpečené připojení WPA2-PSK.



Obrázek 2. Wi-Fi modul ESP8266-01

4.6.1. Napájení

Malou překážkou při zapojování ESP modulu bylo napájení. Wi-Fi modul pracuje, na rozdíl od Arduino kitu, na napětí 3.3V. Tímto napětím je tedy napájen a zároveň na tomto napětí také pracují jeho komunikační piny. Aby nedošlo k poškození modulu, bylo nutné použít stabilizovaný zdroj 3.3V na napájení a odporový dělič na snížení napětí v komunikačním obvodu.

4.7. PHP, server, webová stránka

Nakonec byla vytvořena webová stránka, na které jsou teploty ze senzorů zobrazeny. V pozadí těchto stránek je PHP server, kde je uložen příkaz GET na ukládání naměřených dat a tabulka, kam jsou data ukládána.

5. Návrh a realizace vlastního řešení

V této kapitole je blíže popsána realizace s každou komponentou systému. Společně komponenty tvoří funkční měřicí systém komunikující technologií Wi-Fi.

5.1. Zapojení

Nejdříve byly všechny periferie připojeny k Arduino kitu pomocí drátových propojek. Jak již bylo řečeno, Arduino NANO má třicet I/O pinů. Hlavní jsou digitální a analogové. Dále jsou přítomny piny na Reset, sériovou komunikaci (piny Tx a Rx) apod. Ve většině případů nezáleží na tom, který pin je zrovna použit, mikroprocesor na Arduino desce automaticky příslušný port namapuje. Toto však neplatí pro následující komponentu.

5.1.1. Displej

Piny A4 a A5 jsou určeny jako jediné také k připojení rozhraní I²C. Na tyto dva piny jsou tedy připojeny datové piny displeje. Displej je dále připojen na napájení (napětí 5 V) a zem (GND).

5.1.2. Digitální senzor

Dále byl připojen digitální senzor. Jedná se o tři piny, první je napájecí (napětí 5 V), druhý je připojen na zem (GND) a třetí na digitální pin D3, kde je snímána jeho teplota. Mezi napájecí a snímací pin je umístěn rezistor (odpor 4,7 kΩ) dle datasheetu [15].

5.1.3. Analogový senzor

Druhý, analogový senzor má podobné zapojení, liší se jen v pinu, kterým je snímána teplota. Senzor je zapojen na analogový pin A0.

5.1.4. Wi-Fi modul

Nakonec byl připojen Wi-Fi modul ESP8266-01. Nejdříve muselo být vyřešeno napájení. K tomuto účelu byl zvolen stabilizovaný zdroj napětí 3,3 V model TS1117BCW33 SOT223 od výrobce TAIWAN SEMICONDUCTOR, který je umístěn na tištěném spoji. Napájení modulu je vyřešeno připojením ke stabilizovanému zdroji a zemi (GND). Dále, aby nedošlo k poškození modulu, byl přijímací pin modulu k vysílacímu pinu Arduina připojen přes odporový dělič. Vysílací pin modulu je k přijímacímu pinu Arduina připojen přímo.

5.1.5. Návrh a tvorba tištěného spoje

Všechna zařízení obsažená v měřicím systému je z estetických i bezpečnostních důvodů třeba umístit na PCB. Protože deska není snadno modifikovatelná, byla vytvořena až poté, kdy byly zprovozněny všechny periferie systému. Navržena byla v počítačovém

programu Eagle. Podle návrhu bylo možné kyselinou vyleptat vodivé cesty v měděné fólii, nalepené na izolační sklolaminátové desce. Následně byla deska v místech průniku součástek provrtána speciálními vrtáky na PCB, protože běžné vrtáky by se o sklolaminát ztupily. Nakonec byly na PCB umístěny a přiletovány všechny součástky.

5.2. Programování hardwarové části

Programování Arduino kitu usnadňují zpřístupněné knihovny.

5.2.1. Displej

Existuje mnoho způsobů programování univerzálního, dvouřádkového, šestnácti znakového displeje. Zde byla zvolena metoda použití knihovny `LiquidCrystal_I2C`, která byla ke stažení na stránkách e-shopu [16], kde byl display zakoupen. Knihovna obsahuje vše potřebné k uvedení zařízení do chodu. Knihovnu stačí zahrnout do projektu a vytvořit její instanci. Po vytvoření instance třídy už je výpis na displej jednoduše uskutečnitelný příkazy:

```
lcd.setCursor(0, 0);
lcd.print("text co chcete vypsát");
```

kde první příkaz nastavuje, na které pozici na displeji začne výpis a druhý zajišťuje výpis samotný. Během programování byl zjištěn problém s vypsáním znaku °. Knihovna `LiquidCrystal_I2C.h` totiž obsahuje pouze alfanumerické znaky anglické a japonské abecedy (Obrázek 3). Tento znak bylo potřeba vytvořit ručně. Při bližším zkoumání bylo zjištěno, že lze nadefinovat až šest vlastních znaků pomocí příkazu:

```
lcd.createChar (0, degree);
```

kde `degree` je pole bytů:

```
byte degree[8] = {
  B00110,
  B01001,
  B01001,
  B00110,
  B00000,
  B00000,
  B00000,
  B00000
};
```

Toto pole představuje jeden znak na displeji. Pozice se znakem 0 nebo 1 představuje jeden pixel ze znaku na displeji. Znak 0 značí, že pixel bude zhasnutý, znak 1, že rozsvícený.

5. Návrh a realizace vlastního řešení

Pokud jsou v zápisu znaky 0 nahrazeny mezerami, je názorněji vidět, že pole tvoří na displeji znak °.

```
byte degree[8] = {
  B 11 ,
  B 1 1,
  B 1 1,
  B 11 ,
  B    ,
  B    ,
  B    ,
  B    ,
};
```

Výpis samotného znaku je proveden příkazem:

```
lcd.print(char(0));
```

Teplota na displeji je aktualizována v každém cyklu Arduino kitu, tedy přibližně každé 4/5 sekundy.

Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 bits	Upper 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	1	2	3	4	5	6	7	8	9	A	B	C	D
xxxx0001	(2)			!	"	#	\$	%	&	'	()	*	+	,	-	.
xxxx0010	(3)			"	#	\$	%	&	'	()	*	+	,	-	.	0
xxxx0011	(4)			#	\$	%	&	'	()	*	+	,	-	.	0	1
xxxx0100	(5)			\$	%	&	'	()	*	+	,	-	.	0	1	2
xxxx0101	(6)			%	&	'	()	*	+	,	-	.	0	1	2	3
xxxx0110	(7)			&	'	()	*	+	,	-	.	0	1	2	3	4
xxxx0111	(8)			'	()	*	+	,	-	.	0	1	2	3	4	5
xxxx1000	(1)			()	*	+	,	-	.	0	1	2	3	4	5	6
xxxx1001	(2))	*	+	,	-	.	0	1	2	3	4	5	6	7
xxxx1010	(3)			*	+	,	-	.	0	1	2	3	4	5	6	7	8
xxxx1011	(4)			+	,	-	.	0	1	2	3	4	5	6	7	8	9
xxxx1100	(5)			,	-	.	0	1	2	3	4	5	6	7	8	9	A
xxxx1101	(6)			-	.	0	1	2	3	4	5	6	7	8	9	A	B
xxxx1110	(7)			.	0	1	2	3	4	5	6	7	8	9	A	B	C
xxxx1111	(8)			0	1	2	3	4	5	6	7	8	9	A	B	C	D

Note: The user can specify any pattern for character-generator RAM.

Obrázek 3. Tabulka znaků

5.2.2. Digitální senzor

Pro zprovoznění digitálního čidla byly použity následující knihovny a příkazy nalezené na e-shopu [17], kde byl senzor zakoupen:

```
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 3 /*-(Connect to Pin 3 )-*/

OneWire ourWire(ONE_WIRE_BUS);

DallasTemperature sensors(&ourWire);
```

Když je již vše nadeklarováno, pro přečtení hodnoty teploty ze senzoru slouží příkaz:

```
sensors.getTempCByIndex(0);
```

Přečtená hodnota je zkalibrována výrobcem a není ji tak potřeba nijak upravovat.

5.2.3. Analogový senzor

Pro analogový senzor není třeba žádná knihovna, protože jeho výstupní hodnota je analogová a Arduino kit má vestavěný A/D převodník. Příkazem:

```
analogRead(A0);
```

je ze vstupu A0 přečtena analogová hodnota reprezentovaná jako číselná hodnota v rozmezí 0 až 1023. Hodnotu bylo třeba upravit (zkalibrovat) tak, aby odpovídala skutečné teplotě. Postup se nazývá proložení přímkou a jeho výsledkem je faktor, kterým je nutno analogovou hodnotu vynásobit. Následně pak bude vypočtená hodnota odpovídat reálné teplotě. Konkrétně se jedná o výraz:

$$\text{Digitální hodnota} = (\text{Analogová hodnota} \times 0,404375) - 224,441$$

Nyní již lze teplotu vypsát na displej.

5.2.4. Wi-Fi modul

Tento modul komunikuje přes sériovou linku pomocí AT příkazů (Obrázek 4). Při programování a testování bylo však zjištěno, že nelze použít sériovou linku, která je obsažena v Arduino kitu. Aby bylo možné použít sériovou linku z Arduina, muselo by se jednat o jiný model, např. MEGA. V modelu NANO však existují i další možnosti, jako například zde použitá Softwarová sériová linka (SoftwareSerial). Ta se na pinech 10 a 11 definuje pomocí příkazu:

```
SoftwareSerial softSerial(10, 11);
```

a následně spouští na rychlosti 115200 boudů příkazem:

```
softSerial.begin(115200);
```

5. Návrh a realizace vlastního řešení

AT COMMAND	DESCRIPTION	PARAMETERS	EXAMPLE	QUERY
AT+RST	Reset the module		AT+RST	
AT+CWMODE=<mode>	Client / Access Point mode	1 = Client 2 = AP 3 = Client & AP	AT+CWMODE=3	
AT+CWLAP	List available access points		AT+CWLAP	AT+CWLAP
AT+CWJAP=<SSID>,<PWD>	Connect to an access point	<SSID> = SSID of access point to join <PWD> = Password of access point	AT+CWJAP="MYNetwork" "Pass1234"	AT+CWJAP?
AT+CWQAP	Disconnect from access point		AT+CWQAP	AT+CWQAP=?
AT+CWSAP=<SSID>,<PWD>,<CH>,<ENC>	Set parameters as an access point	<SSID> = SSID of access point <PWD> = Password of access point <CH> = 802.11 channel number (1 to 7) <ENC> = Encryption (0 = no encryption, other values not confirmed working)	AT+CWSAP="MYNetwork" "Pass1234" 6.1	AT+CWSAP?
AT+CIPSTATUS	Get the current connection status		AT+CIPSTATUS	AT+CIPSTATUS
AT+CIPMUX=<MUX>	Set single/multiple connection mode	<MUX> Connection mode: 0 = Single 1 = Multiple	AT+CIPMUX=0	AT+CIPMUX?
AT+CIPSTART=<TYPE>,<URL>,<PORT> OR AT+CIPSTART=<ID>,<TYPE>,<URL>,<PORT>	Enable a UDP or TCP connection	CIPMUX=0: <TYPE> = TCP or UDP <URL> = URL or IP address of server to connect to <PORT> = Port number of server CIPMUX=1: <ID> = Connection ID (0 to 4) <TYPE> = TCP or UDP <URL> = URL or IP address of server to connect to <PORT> = Port number of server	AT+CIPSTART="TCP","WWW.GOOGLE.CO.UK",80	AT+CIPSTART=?
AT+CIPSEND=<LEN> OR AP+CIPSEND=<ID>,<LEN>	Send data to connected server	CIPMUX=0: <LEN> = Number of bytes to send CIPMUX=1: <ID> = Connection ID (0 to 4) <LEN> = Number of bytes to send	AT+CIPSEND=24 <body>Hello World</body>	AT+CIPCLOSE=?
AT+CIPCLOSE	Close UDP or TCP connection		AT+CIPCLOSE	AT+CIPCLOSE=?
AT+CIFSR	Get the IP address		AT+CIFSR=?	AT+CIFSR=?
AT+CIPSERVER=<MODE>,<PORT>	Set as TCP server	<MODE> = Open/Close server mode: 0 = Close 1 = Open	AT+CIPSERVER=1,80	

Obrázek 4. AT příkazy

Tato rychlost není zvolena náhodně, jedná se o výchozí rychlost, na které ESP modul komunikuje. SoftwareSerial však není na této rychlosti dostatečně stabilní, proto je AT příkazem rychlost modulu snížena:

```
softSerial.println("AT+CIOBAUD=9600");
```

Snížena musí být také rychlost softwarové sériové linky.

```
softSerial.begin(9600);
```

Dále je modul nastaven do módu klient pomocí AT příkazu:

```
softSerial.println("AT+CWMODE=1");
```

a připojen k Wi-Fi síti pomocí SSID a hesla:

```
softSerial.println("AT+CWJAP=\"Nazev_site\", \"heslo_site\"");
```

Odesílání dat na server

ESP modul před odesláním dat vyžaduje, aby nejdříve bylo známo, jakou délku budou odesílaná data mít, proto je zde vytvořena proměnná *sget*, která obsahuje odesílaná data a její délka je změřena příkazem:

```
sget.length();
```

obsaženém ve třetím příkazu níže. Posílání dat na server je uskutečněno pomocí pěti (AT) příkazů:

- `softSerial.print("AT+CIPSTART=\"TCP\", \"www.light-web.cz\", 80\r\n");`
- `sget = "GET /teploty/?text=" + (String)digitalni_hodnota + "&id_temp=1 HTTP/1.1\r\nHost: light-web.cz\r\n\r\n";`
- `softSerial.print("AT+CIPSEND=" + (String)sget.length() + "\r\n");`
- `softSerial.print(sget);`
- `softSerial.print("AT+CIPCLOSE");`

Systém pracuje následujícím způsobem: nejprve je ESP modul pomocí prvního příkazu připojen k danému serveru, poté jsou druhým příkazem sestavena data, která budou odeslána, třetím příkazem je ESP modulu řečeno, jakou délku budou odesílaná data mít, čtvrtým jsou data odeslána a posledním příkazem je ukončena komunikace. V podstatě se uskuteční to samé, jako kdyby byla do internetového prohlížeče vložena webová adresa:

```
http://www.light-web.cz/teploty/?text=teplota&id_temp=1
```

Komunikace není nijak šifrována, použitím tohoto příkazu by mohla být do systému úmyslně zanášena chybná data.

Proces odeslání se opakuje pro každý senzor zvlášť, obě hodnoty jsou odeslány každý 300. cyklus kitu Arduino, to přibližně odpovídá čtyřem minutám.

5.3. Programování serverové části

Naprogramování serveru lze rozdělit na tyto tři hlavní části:

- Vytvoření tabulky na ukládání dat s hodnotami naměřených teplot,
- vytvoření příkazů na ukládání dat odesílaných ze vzdáleného přípravku,
- naprogramování webové stránky na zobrazení dat v přehledném grafu.

Příkazy na dotazování databáze a další použité příkazy jsou k nahlédnutí v příloze v adresáři teploty.

5.3.1. Vytvoření tabulky

Na serveru je nejprve nutné vytvořit tabulku, do které bude možné ukládat data s hodnotami:

```
CREATE TABLE 'text' (
  'id' int(255) unsigned NOT NULL AUTO_INCREMENT,
  'text' varchar(255) DEFAULT NULL,
  'id_temp' int(255) DEFAULT NULL,
  'time' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
  ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5.3.2. Vytvoření příkazů na ukládání dat

Připojení k databázi se provede následujícími příkazy:

```
//ip adresa mysql, jmeno, heslo, database
$mysqli = mysqli_connect("localhost", "40247_admin",
  "heslo", "light-web_cz_teploty");
```

Načtení hodnot do proměnných se provede příkazy:

```
$text = $_GET['text'];
$id_temp = $_GET['id_temp'];
```

5. Návrh a realizace vlastního řešení

Uložení hodnot do tabulky je nakonec provedeno příkazem:

```
var_dump(mysqli_query($mysqli, "
    INSERT INTO 'text'
    (text, id_temp)
    VALUES ('".$text."', '".$id_temp."')
    "));
```

5.3.3. Webová stránka na zobrazení grafů

Na obrázku 5 lze vidět graf s online teplotami.

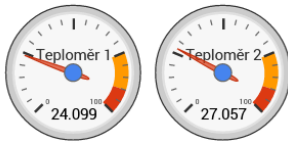
```
//Graf
google.charts.setOnLoadCallback(drawChart2);
function drawChart2() {
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'X');
    data.addColumn('number', 'Teploměr 1');
    data.addColumn('number', 'Teploměr 2');

    data.addRows([
        <?php echo $tempController->getHtmlByIdTemperature(); ?>
    ]);

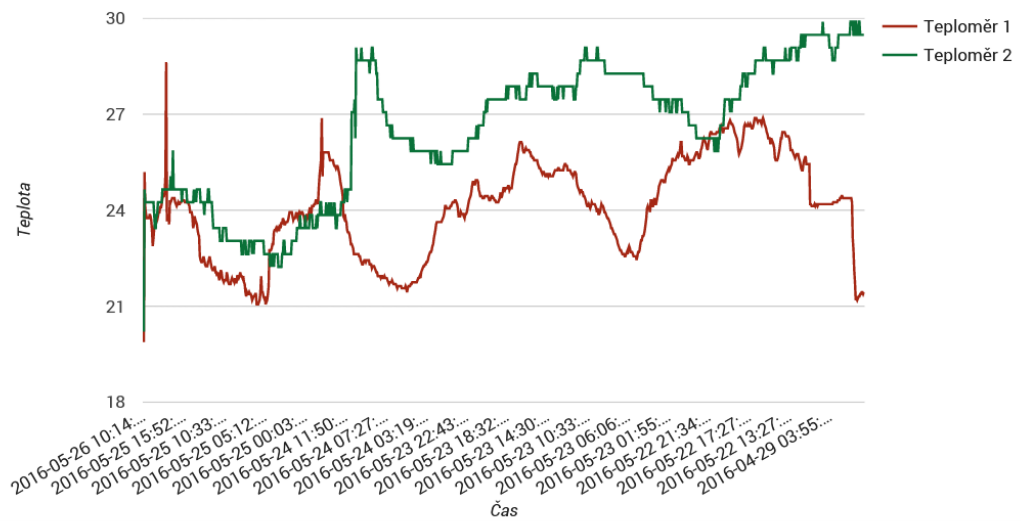
    var options = {
        hAxis: {
            title: 'Čas'
        },
        vAxis: {
            title: 'Teplota'
        },
        colors: ['#a52714', '#097138']
    };

    //Vykreslení grafu
    var chart = new google.visualization.LineChart
    (document.getElementById('columnchart_material'));
    chart.draw(data, options);
}
```

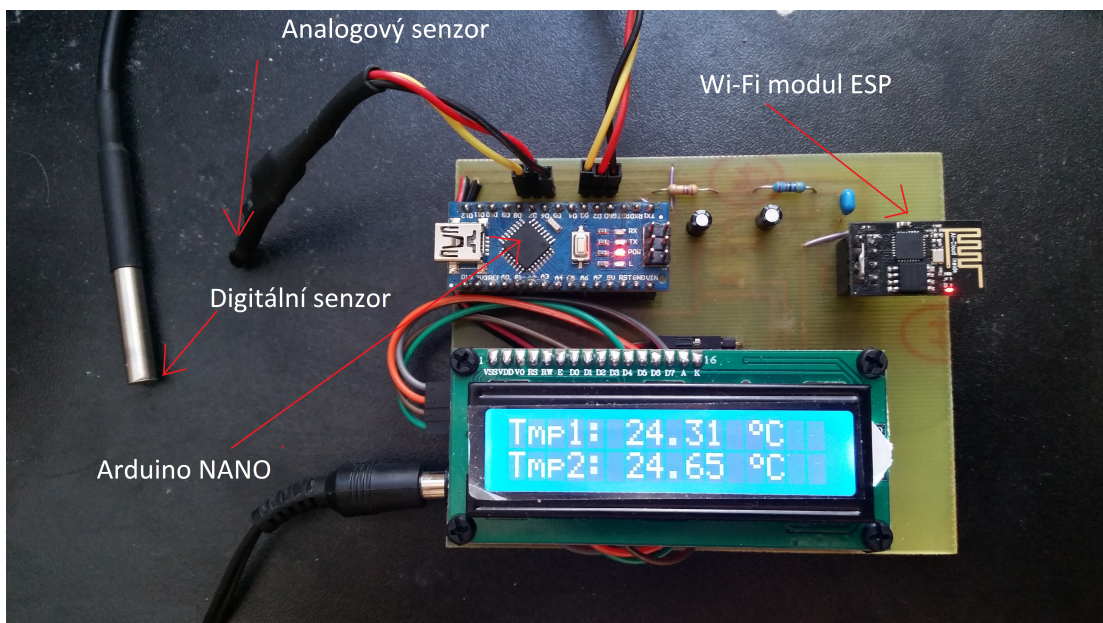
Průměrná teplota na teploměrech



Teploty v grafu



Obrázek 5. Graf s teplotami



Obrázek 6. Měřicí systém

6. Testování

Při testování byla použita domácí Wi-Fi síť se zabezpečením WPA2-PSK a šifrováním WEP.

6.1. Bez PCB

Testování probíhalo zejména se zapojením bez PCB, protože v té době nebyla ještě deska vyrobena. Zařízení je dle mého názoru velice stabilní, při každém zapnutí se automaticky připojí k nedefinované síti Wi-Fi, začne zobrazovat aktuální teploty na displej a pravidelně odesílá změřené hodnoty na server. V tomto zapojení byl celý systém napájen z počítačového zdroje. Bylo tak učiněno z toho důvodu, že bylo potřeba napájet dvěma různými napětími. Napětím 5 V a 3,3 V. Počítačový zdroj produkuje obě tato napětí. Při jednom z prvních "ostrých" měření během první hodiny začal displej zobrazovat nesmyslné kombinace znaků. Zařízení bylo resetováno a problém se již po dobu 4 dnů neobjevil. Není známo, čím problém vznikl, případné pátrání by však nezajistilo zjištění příčiny chybného stavu systému.

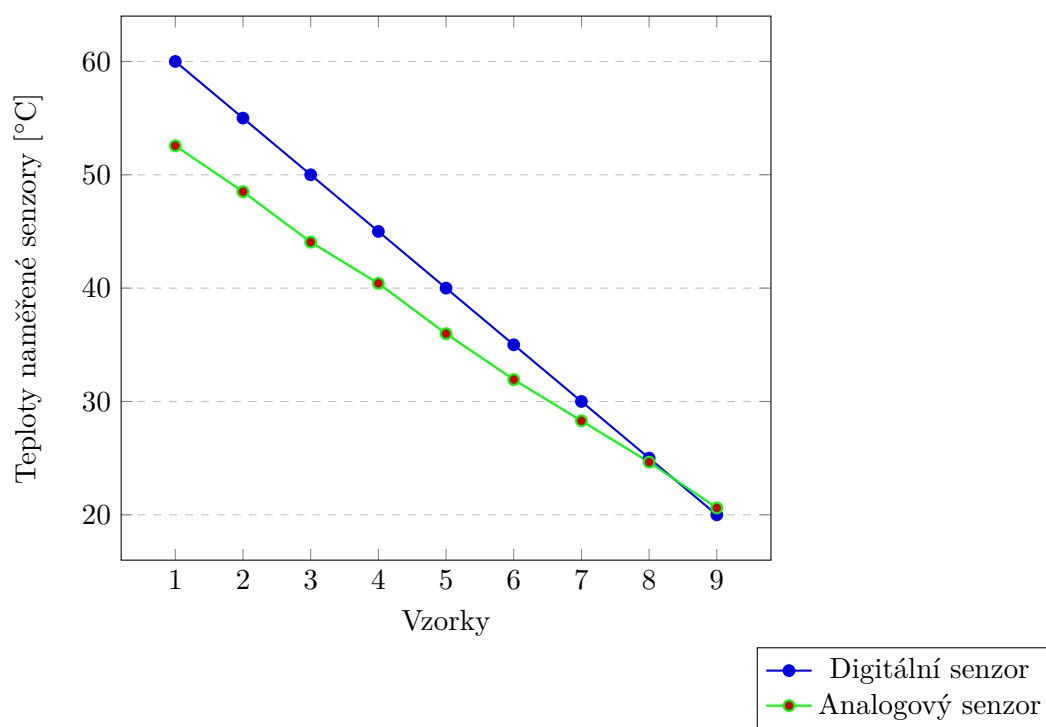
6.2. S PCB

Když byl celý systém přemístěn na PCB, vše bylo funkční na stejné úrovni. Deska tiskárných spojů obsahuje stabilizovaný zdroj napětí 3,3 V. Počítačový zdroj proto není již třeba. K napájení slouží trafo-adaptér produkující napětí 10 V. Systém je nyní kompaktní. Navzdory tomu, že Wi-Fi modul ESP8266-01 obsahuje pouze malou, integrovanou anténu a domácí router značky Ubee od poskytovatele UPC je málo kvalitní a obsahuje také pouze integrovanou anténu, je Wi-Fi modul schopen odesílat data skrze dvě panelové zdi bytu na vzdálenost přibližně sedmi metrů, tedy za podmínek, které nezvládá leckterý notebook. Během testování byla pomocí IR modelářského teploměru měřena teplota všech zařízení. Displej nevykazoval žádné známky zahřívání, stabilizátor napětí 3,3 V měl teplotu přibližně 36 °C, Wi-Fi modul měl na čipu teplotu přibližně 40 °C. Arduino mělo na mikrokontrolleru teplotu přibližně 35 °C, na stabilizátoru napětí 5 V přibližně 60 °C. Tyto teploty zařízení žádným způsobem neohrožují.

Přenos digitálního senzoru je předepsaná výrobcem ($\pm 0,5$ °C) a nelze jinak ovlivnit. Analogové senzory ovlivnit lze. Jak je patrné z Obrázku 7, mezi procesem prokládání přímkou a měřením nastala pravděpodobně změna napájení díky umístění systému na PCB, která způsobila nesprávné měření analogového senzoru.

6.3. WEB

Webové rozhraní je funkční bez problému. Není závislé na odesílání měřených dat, oba systémy se navzájem nijak negativně neovlivňují. Naměřená data jsou přístupná na adrese: <http://www.light-web.cz/teploty/chart.php>



Obrázek 7. Porovnání naměřených hodnot obou senzorů

7. Návrh dalších vylepšení

V první řadě by bylo vhodné naprogramovat Arduino pomocí "stavů". Nyní je programováno pouze jako sled příkazů a kdyby nastala nějaká neočekávaná situace (ztráta dosahu Wi-Fi sítě), není Arduino momentálně schopno tuto situaci řešit a je nutné kit restartovat. Taktéž není schopno ověřit, zda byla data poslána úspěšně nebo jestli odeslání z jakéhokoliv důvodu selhalo.

Dalším krokem by bylo správné zkalibrování analogového senzoru.

Dále by bylo vhodné vylepšit komunikaci mezi Arduinem a ESP modulem. Ovládat ESP modul je možné bez problémů, odpovědi však jsou téměř nečitelné. Nelze tak přechytit případné chybové kódy.

Bylo by také možné rozšířit funkce systému například: o vyhledání Wi-Fi sítí, výběr sítě ze seznamu a zadání, popřípadě zapamatování hesla k síti. Nyní je konfigurace sítě možná pouze pomocí počítače přímo v kódu celého programu.

Samozřejmostí je přidání jakéhokoliv senzoru, konkrétně například senzoru vlhkosti, prachových částic, světla, deště, či větru. Zařízení by tak mělo mnohem širší využití.

Také by bylo možné použít GSM modul namísto Wi-Fi modulu, byla by tak odstraněna závislost zařízení na připojení k internetu a bylo by možné s použitím akumulátoru měřit kdekoliv, kde je v dosahu signál mobilních sítí.

Co se týká webové části, nebylo dosaženo všech cílů. Nepodařilo se vytvořit takový graf, kde by bylo možné nastavit příslušný rozsah časové osy. Graf je momentálně nastavený tak, že se zhušťuje s každým přidaným záznamem. Tento problém by bylo jistě nutné vyřešit.

8. Závěr

V této bakařské práci se střetáváme s tematikou embeded systémů, měření veličin pomocí sensorů a odesíláním dat prostřednictvím Wi-Fi. Cílem práce bylo sestavit a naprogramovat zařízení, které mělo snímat vlastnosti ovzduší. K dosažení tohoto cíle bylo nutné se seznámit s problematikou programování Arduina, výběrem a naprogramováním sensorů, ovládáním ESP modulu pomocí AT příkazů a programováním v PHP, což podrobněji zmiňuje teoretická část. Poté bylo přistoupeno ke konkrétním praktickým a programovacím metodám, které jsou popsány v praktické části.

Původní záměr vytvoření systému na snímání prachových částic z ovzduší pro okolí Prahy 4-Krče může být přínosem pro velké množství místních obyvatel. Tento fakt byl pro mne velkou motivací, protože sestavením zařízení bylo ověřeno, že systém může být sestaven a naprogramován. Nyní ve své podstatě k uskutečnění tohoto cíle chybí pouze již zmíněný sensor na prachové částice. Do systému je v principu možné přidat jakýkoliv další sensor.

Největší překážkou pro mne bylo programování webu a použití jazyka PHP. Nikdy předtím jsem se totiž s tímto nesetkal. Důsledkem toho je také ne zcela splněný cíl s grafickým zobrazováním získávaných dat. Co se týká překonaných překážek, největším problémem pro mne bylo zprovoznění ESP modulu, zjištění, že je k tomu zapotřebí softwarová sériová linka, správná formulace AT příkazů a nakonec již zmíněný postup s délkou odesílaných dat.

Zadaný úkol byl splněn kromě jediného dílčího cíle, tím je ovládání rozsahu grafu.

Příloha A.

CD

Seznam souborů na CD:

riha_jan.pdf
teploty.zip
arduino.zip

Literatura

- [1] CENIA a Ministerstvo životního prostředí České republiky. *Integrovaný registr znečišťování - IRZ [online]*. [cit. 2016-01-02]. URL: <http://www.irz.cz/node/85>.
- [2] Miroslav Šuta. *Nízkoemisní zóny – diskriminace smradochů pro čistější ovzduší měst [online]*. [cit. 2016-01-03]. 2009. URL: <http://suta.blog.respekt.cz/c/67356/Nizkoemisni-zony-diskriminace-smradochu-pro-cistejsi-ovzdusi-mest.html>.
- [3] *Předpoklady výskytu zvýšené sekundární prašnosti [online]*. [cit. 2016-01-05]. 2001-12. URL: http://envis.praha-mesto.cz/rocenky/DZ_00/pril_practexty/BK14/SekPrasnost.pdf.
- [4] Státní zdravotní ústav. *Suspendované částice (aerosol) [online]*. [cit. 2016-01-05]. 2006. URL: http://www.szu.cz/uploads/documents/chzp/ovzdusi/dokumenty_zdravi/susp_castice.pdf.
- [5] Miroslav Šuta. *Účinky výfukových plynů z automobilů na lidské zdraví*. 1. vydání. Děti Země, 2008. ISBN: 80-86678-10-5.
- [6] CENIA a Ministerstvo životního prostředí České republiky. *Integrovaný registr znečišťování - IRZ [online]*. [cit. 2016-01-06]. 2014. URL: <http://www.irz.cz/node/179>.
- [7] Ufrieg. *Definition of ultrafine particles and why it is important to measure them [online]*. [cit. 2016-01-06]. URL: <http://www.ufireg-central.eu/index.php/about-the-topic01>.
- [8] Bc. Linda Landlová. “STUDIUM PERZISTENTNÍCH ORGANICKÝCH POLUTANTŮ VÁZANÝCH NA PRACHOVÉ ČÁSTICE V ATMOSFÉŘE – JEJICH DISTRIBUCE, OSUD A EFEKT”. Dipl. MASARYKOVA UNIVERZITA, PŘÍRODOVĚDECKÁ FAKULTA, 2008. URL: http://is.muni.cz/th/106244/prif_m/diplomova_prace-Linda_Landlova_1.pdf.
- [9] Miroslav Šuta. *Polétavý prach nám zkracuje život. Jak moc a proč? [online]*. [cit. 2016-01-12]. 2006. URL: <http://www.blisty.cz/art/28008.html>.
- [10] *Ročenka 1998 [online]*. [cit. 2016-01-20]. 1999-07. URL: http://czechweather.eu/uoco/isko/groc/gr98cz/kap_0223/kap_0223.htm.
- [11] pighixx.com. *Picture NANO pinout [online]*. [cit. 2016-05-02]. 2014. URL: <http://pighixx.com/nanopdf.pdf>.
- [12] CzechDUINO.cz. *co je to Arduino [online]*. [cit. 2016-05-02]. 2012. URL: <http://czechduino.cz/?co-je-to-arduino,29>.
- [13] © NXP Semiconductors N.V. *I2C-bus specification and user manual [online]*. [cit. 2016-05-02]. 2014. URL: http://www.nxp.com/documents/user_manual/UM10204.pdf.
- [14] MAXIM Integrated. *1-Wire® Tutorial [online]*. [cit. 2016-05-02]. 2014. URL: <https://www.maximintegrated.com/en/products/1-wire/flash/overview/index.cfm>.

Literatura

- [15] Dallas Semiconductor. *DS18B20 Programmable Resolution 1-Wire® Digital Thermometer [online]*. [cit. 2016-03-15]. URL: <https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/DS18B20.pdf>.
- [16] *Internetový obchod Dexhal.cz, LCD 1602 znakový displej modrý s I2C // Arduino [online]*. [cit. 2016-03-15]. URL: http://dexhal.cz/displeje/535-lcd-1602-znakovy-displej-modry-s-i2c-arduino_el-at-ds-16x2bl.html#ke_stazeni.
- [17] *Internetový obchod Dexhal.cz, Vodotěsný teplotní senzor DS18B20 Arduino Raspberry [online]*. [cit. 2016-03-15]. URL: http://dexhal.cz/meteorologicke-veliciny/296-vodotesny-teplotni-senzor-ds18b20-arduino-raspberry_el-at-sethp-2.html#ke_stazeni.