



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

ZAŘÍZENÍ PRO MONITOROVÁNÍ KVALITY VZDUCHU

DEVICE FOR MONITORING OF AIR QUALITY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

FILIP JAŠEK

VEDOUcí PRÁCE

SUPERVISOR

Ing. TOMÁŠ GOLDMANN,

BRNO 2018

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2017/2018

Zadání bakalářské práce

Řešitel: **Jašek Filip**

Obor: Informační technologie

Téma: **Zařízení pro monitorování kvality vzduchu**
Device for Monitoring of Air Quality

Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s problematikou monitorování ovzduší. Zjistěte, které senzory se k tomuto monitorování používají.
2. Navrhněte zařízení využívající vhodnou mikrokontrolérovou platformu, které bude sledovat teplotu a monitorovat kvalitu ovzduší. Dále navrhněte přenos dat ze zařízení do mobilního telefonu pomocí technologie Bluetooth.
3. Navržené zařízení realizujte na vybrané platformě. Implementujte pro mikrokontrolér program, který bude zpracovávat data ze senzorů a ukládat je do paměti. Pro čtení dat ze zařízení vytvořte mobilní aplikaci pro operační systém Android.
4. Vytipujte ve městě či jeho okolí několik míst, kde by měla být rozdílná kvalita vzduchu a proveďte experimenty.
5. Zhodnoťte možnosti použití zařízení a navrhněte další rozšíření.

Literatura:

- IKRAM, Jahangir, et al. View: implementing low cost air quality monitoring solution for urban areas. *Environmental Systems Research*, 2012, 1.1: 10.
- KNOX, Andrew, et al. The expanding scope of air pollution monitoring can facilitate sustainable development. *Science of the total environment*, 2013, 448: 189-196.
- KIZEL, Fadi, et al. Node-to-node field calibration of wireless distributed air pollution sensor network. *Environmental Pollution*, 2017.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

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

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

Vedoucí: **Goldmann Tomáš, Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Práce se zabývá návrhem a realizací zařízení pro monitorování kvality vzduchu pomocí vybraných senzorů (RHT03, MQ-2, MQ-7) na mikrokontrolérové platformě STM Nucleo F446RE. V další části je popsán vývoj mobilní aplikace za účelem vyčtení naměřených dat pomocí technologie Bluetooth Low Energy vytvořené pomocí frameworku NativeScript primárně pro platformu Android. Součástí práce je i zkušební provoz měřicího zařízení a vyhodnocení výsledků.

Abstract

The thesis deals with the design and implementation of air quality monitoring device using selected sensors (RHT03, MQ-2, MQ-7) on STM Nucleo F446RE microcontroller platform. The next section describes the development of a mobile application to read out measured data using Bluetooth Low Energy created with the NativeScript framework primarily for the Android platform. Part of the thesis is the test operation of the measuring device and evaluation of the results.

Klíčová slova

Monitorování kvality vzduchu, STM Nucleo, Bluetooth, NativeScript, mobilní aplikace, Android, senzory, mikrokontrolér, nevolatilní paměť.

Keywords

Air quality monitoring, STM Nucleo, Bluetooth, NativeScript, mobile application, Android, sensors, microcontroller, non-volatile memory.

Citace

J AŠEK, Filip. *Zařízení pro monitorování kvality vzduchu*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Tomáš Goldmann,

Zařízení pro monitorování kvality vzduchu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Tomáše Goldmanna a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Filip Jašek
17. května 2018

Poděkování

Děkuji tímto panu Ing. Tomáši Goldmannovi za odbornou pomoc a vedení mé bakalářské práce.

Obsah

1 Úvod	3
2 Měření kvality ovzduší	4
2.1 Dopady znečištění ovzduší na zdraví člověka	4
2.2 Vliv meteorologických podmínek na kvalitu ovzduší	5
2.3 Zdroje znečištění ovzduší	5
2.4 Látky ovlivňující kvalitu vzduchu	6
2.5 Ochrana ovzduší	6
2.6 Prostředky pro měření kvality ovzduší	9
3 Návrh řešení	13
3.1 Návrh měřicího zařízení	13
3.2 Návrh programu zařízení	17
3.3 Návrh mobilní aplikace	18
4 Implementace	26
4.1 Sestava měřicího zařízení	26
4.2 Program měřicího zařízení	28
4.3 Mobilní aplikace	31
4.4 Možnosti dalšího rozšíření zařízení	34
5 Experimenty	36
5.1 Experimentální měření – venkov	36
5.2 Experimentální měření – město	38
6 Závěr	40
Literatura	41
A Snímky obrazovky mobilní aplikace	45

Seznam obrázků

2.1	Měřicí stanice automatizovaného imisního monitoringu v Teplicích.[44]	8
3.1	Vývojová deska STM32 Nucleo-64.[39]	14
3.2	Bluetooth modul HC-05.[19]	17
3.3	Zastoupení jednotlivých verzí androidu k únoru 2018.[3]	21
3.4	Diagram architektury operačního systému Android.[2]	22
3.5	Průběh vyhodnocení objektu promise v jazyce JavaScript.[24]	24
4.1	Zjednodušené schéma zapojení periférií k mikropočítači.	27
4.2	Vytvořené monitorovací zařízení.	28
4.3	Kalibrace senzoru MQ-2.	29
4.4	Ukázka stránky Detail měření.	33
5.1	Experimentální měření během silného deště.	36
5.2	Vývoj hodnot teploty vzduchu na venkově.	37
5.3	Vývoj hodnot relativní vlhkosti vzduchu na venkově.	37
5.4	Vývoj hodnot koncentrace hořlavých plynů (tmavší křivka) a oxidu uhelnatého (světlejší křivka) ve vzduchu na venkově.	38
5.5	Vývoj hodnot teploty vzduchu ve městě.	38
5.6	Vývoj hodnot relativní vlhkosti vzduchu ve městě.	39
5.7	Vývoj hodnot koncentrace hořlavých plynů (tmavší křivka) a oxidu uhelnatého (světlejší křivka) ve vzduchu ve městě.	39

Kapitola 1

Úvod

Jedním z důležitých faktorů ovlivňujících život a zdraví člověka je kvalita prostředí, ve kterém žije, zejména pak kvalita ovzduší. Proto je cílem této bakalářské práce navrhnout, sestavit a naprogramovat zařízení pro monitorování kvality vzduchu.

Začátek práce pojednává obecně o monitorování a ochraně ovzduší, přírodních i antropogenních zdrojích a látkách znečišťujících okolní vzduch a prostředcích pro jejich měření. Další část obsahuje teoretický návrh měřicího zařízení (hardware) na mikrokontrolérové platformě STM32 Nucleo F446RE a jeho programu (software) pro sledování teploty, relativní vlhkosti a koncentrace vybraných látek ovlivňujících kvalitu ovzduší. Součástí návrhu je i ukládání naměřených dat do nevolatilní paměti monitorovacího zařízení. Hlavní částí práce je pak sestavení a popis zařízení a přenos dat z měřicího zařízení do mobilního zařízení pomocí technologie Bluetooth Low Energy.

Součástí práce je také návrh a implementace mobilní aplikace vytvořené pomocí open source frameworku NativeScript pro operační systém Android, která zajišťuje čtení dat ze zařízení, jejich ukládání do paměti a prezentaci naměřených hodnot v tabulce a grafech.

Samostatná kapitola prezentuje experimentální měření na místech s předpokládanou rozdílnou kvalitou vzduchu.

Závěr práce shrnuje získané poznatky, možnosti jejich využití a další možnosti rozšíření měřicího zařízení pro zpřístupnění jím naměřených dat širší veřejnosti s cílem přispět k lepší informovanosti veřejnosti o kvalitě prostředí, ve kterém žijeme.

Kapitola 2

Měření kvality ovzduší

Jako kvalitu vnějšího ovzduší označujeme úroveň znečištění vnějšího ovzduší, která může svými účinky ovlivňovat lidské zdraví, vegetaci, materiály i celé ekosystémy. Tato úroveň znečištění vnějšího ovzduší je způsobena vypouštěním znečišťujících látek z různých zdrojů v důsledku přírodní nebo lidské činnosti (např. doprava, spalování, průmyslová výroba a další). Znečišťující látky jsou po vypuštění ze zdroje přenášeny v atmosféře a mohou tak ovlivňovat kvalitu ovzduší jak v nejbližším okolí samotného zdroje znečištění, tak ve vzdálenějších oblastech. [23]

2.1 Dopady znečištění ovzduší na zdraví člověka

Vliv znečišťujících látek z ovzduší na zdraví člověka závisí nejen na jejich schopnosti působit na zdraví, ale také na velikosti expozice, tedy na tom, po jakou dobu a jak vysoké koncentraci látek jsou lidé vystaveni. Z těchto a dalších informací se sestavuje tzv. *zdravotní riziko*.

Zdravotní riziko vyjadřuje pravděpodobnost změny zdravotního stavu exponovaných osob. Při hodnocení zdravotních rizik se standardně postupuje ve čtyřech následných krocích.

1. Identifikace zdravotní nebezpečnosti – zda je sledovaná látka, faktor nebo komplexní směs schopná vyvolat nežádoucí zdravotní účinek.
2. Odhad dávkové závislosti nežádoucího efektu – jak se intenzita, frekvence nebo pravděpodobnost nežádoucích účinků mění s dávkou sledované látky nebo faktoru.
3. Odhad expozice – do jaké míry je populace vystavena působení sledované látky či faktoru v daném prostředí.
4. Charakterizace rizika – integrace poznatků vyplývajících ze všech výše zmíněných kroků, včetně zvážení všech nejistot, závažnosti i slabých stránek použitých podkladových materiálů. Cílem je dospět, pokud to dostupné informace umožňují, ke kvantitativnímu vyjádření míry konkrétního zdravotního rizika za dané situace.

Jednotlivé látky mají vliv na konkrétní zdravotní problémy. Například působení oxidu dusičitého (NO_2) je spojováno se zvýšením celkové, kardiovaskulární a respirační úmrtnosti. Hlavním účinkem krátkodobého působení vysokých koncentrací NO_2 je nárůst reaktivity dýchacích cest a zvýšený výskyt astmatických obtíží a alergií. U dětí představuje

expozice vyšším koncentracím NO_2 zvýšené riziko respiračních onemocnění v důsledku snížené obranyschopnosti vůči infekci a snížení plicních funkcí. Je ale obtížné až nemožné oddělit účinky dalších, současně působících látek. [22]

2.2 Vliv meteorologických podmínek na kvalitu ovzduší

Je známo, že úroveň znečištění ovzduší významně ovlivňují tzv. meteorologické podmínky rozptylu, které podmiňují šíření znečišťujících látek v atmosféře. Za nejdůležitější a rozhodující meteorologické podmínky jsou považovány směr a rychlost větru (proudění) a teplota vzduchu.

Směr a rychlost větru ovlivňují horizontální šíření a rozptyl znečišťujících látek v atmosféře. Při vyšší rychlosti větru je rozptyl intenzivnější a koncentrace znečišťujících látek nižší. Nicméně při vysokých rychlostech větru může docházet k resuspenzi, tj. k opětovnému zvržení částic po jejich usazení.

Vliv teploty vzduchu na znečištění ovzduší je přímý a nepřímý. Změny teploty s výškou, tzv. vertikální teplotní zvrstvení, podmiňují vertikální stabilitu ovzduší, a tím přímo ovlivňují vertikální šíření a rozptyl znečišťujících látek v atmosféře. Čím je teplotní zvrstvení stabilnější, tj. když teplota vzduchu se zvětšující se výškou nad zemí klesá pomaleji, nemění se, nebo dokonce stoupá, tím hůře se znečišťující látky vertikálně rozptylují. Nejméně příznivou situací je tzv. teplotní inverze, při které je teplota v nižší hladině atmosféry nižší než v hladině vyšší. Teplota vzduchu rovněž může nepřímo, především v topném období, výrazně ovlivňovat množství emisí. Při nižších teplotách se více topí a tím se množství emisí zvyšuje.

Nezanedbatelný vliv na úroveň kvality ovzduší mají i atmosférické srážky, které mohou snižovat imisní úroveň vymýváním znečišťujících látek z ovzduší. Meteorologické podmínky se v teplém a chladném období roku výrazně liší. Nejméně příznivé podmínky jsou během zimních měsíců od prosince do února. V případě látek, jejichž vznik a chování jsou zásadně ovlivňovány intenzitou slunečního záření, jsou naopak příznivé meteorologické podmínky pro jejich tvorbu po dobu letních měsíců. [6]

2.3 Zdroje znečištění ovzduší

Znečištění ovzduší může být způsobeno chemickým, fyzikálním nebo biologickým činitelem, který změní vlastnosti zemské atmosféry.

Antropogenní zdroje

Mezi antropogenní zdroje jsou řazeny veškeré zdroje podmíněné nebo způsobené člověkem. Hlavními antropogenními zdroji jsou: [12]

- spalování fosilních paliv při výrobě elektřiny, v dopravě, průmyslu a v domácnostech
- průmyslové procesy a používání rozpouštědel, například v chemickém průmyslu a při zpracování nerostných surovin
- řízené spalování lesů používané v zemědělství nebo lesnictví
- zpracování odpadu

Přírodní zdroje

Významnými přírodními zdroji jsou: [45]

- kouř a oxid uhelnatý vznikající při lesních požárech
- sopečná aktivita (uvolňují se částice síry, chloru a popela)
- prach z přírodních zdrojů (zejména z oblastí pokrytých řídkou nebo žádnou vegetací, z pouští)
- bioplyn (metan) uvolňovaný živými organizmy při procesu trávení a emise těkavých organických látek z rostlin

2.4 Látky ovlivňující kvalitu vzduchu

Kvalitu vzduchu ovlivňuje mnoho různorodých látek. Doposud nebyla vytvořena jednoznačná kategorizace rizikových složek vzduchu, bývají tedy uváděny bez určení priorit takto: [20]

- SO_2 – **oxid siřičitý** – Pro jeho měření existují moduly se senzorem.
- NO_x – **oxidy dusíku** – Jedná se o oxid dusný (NO) a oxid dusičitý (NO_2). Pro jejich měření existují moduly se senzorem.
- CO – **oxid uhelnatý** – Pro jeho měření rovněž existují moduly se senzorem.
- CO_2 – **oxid uhličitý** – Pro jeho měření rovněž existují moduly se senzorem.
- O_3 – **přízemní ozón** – Pro jeho měření rovněž existují moduly se senzorem.
- CH_2O – **formaldehyd** – Pro jeho měření rovněž existují moduly se senzorem.
- VOC – **těkavé organické látky** – Pro jejich měření rovněž existují moduly se senzorem.
- TZL – **tuhé znečišťující látky** – Měření tuhých látek je složitější, nejčastěji je realizováno pomocí filtrů pevných částic.

2.5 Ochrana ovzduší

Kvalita životního prostředí ovlivňuje životní úroveň člověka, přičemž ovzduší je jedním z nejvýznamnějších faktorů kvality životního prostředí. I proto je měření kvality ovzduší a jeho ochrana stále častěji skloňovaným tématem. V současné době se pro měření kvality ovzduší využívá široká škála prostředků, od levných senzorů až po komplexní měřicí stanice.

Historie ochrany ovzduší na území ČR

Ochrana čistoty ovzduší na území České republiky má mnohem kratší historii než meteorologie a hydrologie. Po druhé světové válce byla hlavním cílem obnova hospodářství a péči o kvalitu ovzduší nebyla věnována výrazná pozornost.

V padesátých letech 20. století byla hlavním garantem sledování imisní situace tzv. hygienická služba. V tomto období se postupně rozvíjel těžký průmysl a výstavba hnědouhelných elektráren v Podkrušnohoří. Na konci padesátých let byl zapojen do omezování znečištění ovzduší i Český hydrometeorologický ústav, a to zejména výzkumem rozptylu znečištění v ovzduší.

V šedesátých letech 20. století vznikl v Praze pod hydrometeorologickým ústavem zvláštní útvar čistoty ovzduší s regionálními pracovišti na pobočkách ústavu.

Dle údajů OSN bylo v roce 1978 Československo na 3. místě nejvyššího zatížení SO_2 v Evropě. Dochází k výraznému poškození smrkových porostů v Krkonoších a Jizerských horách. Zároveň se zvyšuje výskyt alergií a onemocnění dýchacích cest u dětí. Díky nově zaváděným dvojestupňovým odlučovačům se však snižuje prašnost.

V osmdesátých letech dosahuje znečištění ovzduší na území Česka svého maxima. Na zhoršené imisní situaci se podílí i četné inverze. V důsledku masivního poškození a odumírání lesů dochází k jejich velkoplošné těžbě. Veřejnost začíná zvyšovat svůj zájem o vliv ovzduší na lidské zdraví a probíhají demonstrace pro čisté ovzduší. Ke zdrojům znečištění se neřadí jen tepelné elektrárny, ale i lokální topeniště, dálkový přenos škodlivin a postupně se zvyšující doprava.

Změna politické situace a převod ČHMÚ do působnosti nově vzniklého ministerstva životního prostředí v devadesátých letech ještě umocnila všeobecný zájem o kvalitu ovzduší. V této době vzniká na celém území ČR automatizovaná síť sledování kvality ovzduší se zaměřením na škodliviny jako oxid siřičitý, prach, oxidy dusíku a ozon. Významným zlomem koncem devadesátých let bylo odsíření elektráren a pokles emisí prachových částic. Roku 1992 je založen informační systém kvality ovzduší, který však navazuje na starší imisní informační systémy, a tak obsahuje některé záznamy i z roku 1971.

Počátkem první dekády tohoto století byla monitorovací síť komplexně rekonstruována a doplněna o sledování dalších látek, zejména polyaromatických uhlovodíků a postupně i jemných prachových částic. V roce 2012 vstoupil v platnost nový zákon o ochraně ovzduší, který platí až dodnes. [43]

Aktuální stav ochrany ovzduší v ČR

Kvalitu ovzduší lze sledovat pravidelně na území celé ČR prostřednictvím sítě měřicích stanic tzv. *imisního monitoringu*. Imise (škodliviny volně se rozptylující v ovzduší) se nacházejí všude kolem nás a mohou mít škodlivý účinek nejen na zdraví lidí a stav ekosystémů. Důležité je proto provádět jejich monitoring – sledování a vyhodnocování škodlivých látek ovzduší. Znečištěné ovzduší má dle platného zákona č. 201/2012 Sb., o ochraně ovzduší, ve znění pozdějších předpisů, definovány své limity pro stanovené znečišťující látky – tzv. *imisní limity*. Kromě imisních limitů jsou v zákoně definovány i prahové (informativní, regulační a varovné) hodnoty pro vybrané znečišťující látky (jako SO_2 , NO_2 , TZL a O_3), pro které při překročení těchto prahových hodnot platí vyhlášení smogové situace. [43]

Imisním monitoringem v České republice je v současné době pověřen Český hydro-meteorologický ústav (ČHMÚ). Tento ústav provádí monitorování imisí dvěma hlavními způsoby:

- **AIM – automatizovaný imisní monitoring** – měření provádějí měřicí stanice, jako je ta na obrázku 2.1. Tyto stanice zasílají on-line výsledky měření ČHMÚ. Většina stanic je osazena analyzátory na měření koncentrací oxidu siřičitého (SO_2), oxidu dusnatého (NO), oxidu dusičitého (NO_2) a prašného aerosolu (TZL). Na menším počtu stanic jsou stanovovány koncentrace přízemního ozonu (O_3) a oxidu uhelnatého (CO). Vybrané stanice AIM měří i koncentrace některých těkavých organických látek (benzen, toluen, xylen).
- **MIM – manuální imisní monitoring** – měření koncentrací znečišťujících látek probíhá dodatečně v laboratořích ČHMÚ.

Aktuální imisní situaci na vybraných měřicích stanicích lze sledovat on-line, na webovém portálu ČHMÚ¹, kde se data z měřicích stanic každou hodinu aktualizují. [43]



Obrázek 2.1: Měřicí stanice automatizovaného imisního monitoringu v Teplicích.[44]

¹Aktuální imisní situace: <https://bit.ly/1pCmDaB>.

2.6 Prostředky pro měření kvality ovzduší

Kvalitu ovzduší lze měřit mnoha různými zařízeními. Od profesionálních stanic, které používá ČHMÚ, přes malé domácí měřicí stanice, až po mikropočítače se senzory pro měření znečišťujících látek.

Mikropočítače

Mikropočítač je počítač obsahující mikroprocesor, paměť a vstupně/výstupní zařízení, na která je možné připojit celou řadu periférií, často bývají mikropočítače také označovány jako vývojové desky.

Arduino

Arduino patří mezi neznámější vývojové desky vůbec. Vývojové desky Arduino pracují na 5V logice a vyrábí se v několika verzích, z nichž pravděpodobně nejrozšířenější je Arduino UNO opatřená mikroprocesorem ATmega328P s frekvencí hodinového signálu 16MHz. [5]

Raspberry Pi

Tento mikropočítač je jeden z nevykonnějších široce známých mikropočítačů. Například Raspberry Pi 3 model B je vybaven čtyřjádrovým procesorem Broadcom BCM2837 se 64bit architekturou a frekvencí 1,2GHz. Výhodou je i integrovaný Wireless LAN a Bluetooth modul. Přímou na desce jsou navíc LAN, HDMI a čtyři standardní USB konektory. Pro účely této práce však není tak výkonné zařízení potřeba. [33]

STM Nucleo

Arduinu konkurují vývojové desky od firmy STMicroelectronics. Například mikropočítač STM32F4 je osazen procesorem ARM Cortex-M4 s 32bitovou architekturou a frekvencí až 84MHz. Výkonově je tedy mezi výše zmíněnými dvěma mikropočítači. Na rozdíl od Arduina, STM Nucleo využívá 3,3V logiku, avšak procesor umí pracovat i se senzory na Arduino, kterých je obecně větší množství. [39]

Čidla teploty

Teplota vzduchu se v praxi měří třemi hlavními způsoby: pomocí termistoru, termočládku a nebo RTD senzoru. Měření teploty může být použito při hledání korelace mezi znečištěním ovzduší a jeho teplotou. [41]

Termistorová čidla

Termistor je druh rezistoru, jehož odpor závisí na okolní teplotě². Rozlišujeme dva druhy termistorů - tedy i dva druhy termistorových čidel: pozistory (PTC) a negastory (NTC). U PTC termistorů odpor se zvyšující se teplotou roste, u NTC odpor termistoru s rostoucí teplotou naopak klesá. Výhodou termistorových senzorů je rychlá doba odezvy a nízká

²Teplotní závislost platí i u běžných rezistorů. V případě termistorů je však závislost výrazně větší.

cena. Mezi nejvýznamnější nevýhody patří malý rozsah měření (pro měření teploty okolního vzduchu to však nehraje roli) a efekt samoohřívání, který může zkreslovat naměřené hodnoty.[41]

Čidla s termočlánkem

Termočlánek je elektronická součástka využívající termoelektrického jevu a je zdrojem napětí závislého na teplotě. Skládá se ze dvou kovů zapojených sériově dvěma spoji. Mají-li spoje navzájem různou teplotu, vzniká na každém spoji odlišný elektrický potenciál. Výhodou termočlánu je velmi široké rozmezí teplot, které je schopný měřit, navíc není ovlivňovaný samoohříváním. [42]

RTD čidla

Odporová teplotní čidla jsou konstruována z rezistivního materiálu (platina, měď nebo nikl) navinutého na keramické jádro. Rozdíl oproti termistorům spočívá v konstrukci. Termistor je vytvořen z polymerů, kdežto RTD jsou čistě metalické. Výhodou je velmi dobrá opakovatelnost a přesnost. Oproti termistorům mají větší rozsah měření, avšak menší než termočlánu. Největší nevýhodou je vysoká cena. [15]

Čidla relativní vlhkosti vzduchu

Vlhkost vzduchu se zjišťuje nejčastěji měřením kapacity polymeru, která se mění v závislosti na vlhkosti vzduchu. Takový princip využívají i senzory DHT11 a DHT22, které jsou kompatibilní s mikropočítačem STM Nucleo a jsou vhodné pro použití v mé práci. Oba zmíněné senzory obsahují navíc i termistorové (NTC) čidlo teploty okolního vzduchu. [26]

Čidla pro měření plynů v okolním vzduchu

Obsah plynů v okolním vzduchu lze měřit několika způsoby. Ve vestavěných systémech se používají převážně dva druhy senzorů, a to elektrochemické a polovodičové.

Elektrochemické senzory generují signální proud v závislosti na obsahu měřeného plynu. Měřený plyn se po difuzi přes porézní membránu dostane do kontaktu s povrchem čidla a dojde k redoxní reakci. Množství takto vzniklého proudu pak udává koncentraci měřeného plynu v okolním vzduchu. Nevýhodou je možná koroze nebo chemická kontaminace senzoru. Z tohoto důvodu je potřeba senzory periodicky nahrazovat novými (jeden senzor vydrží přibližně 1 až 2 roky). Příkladem elektrochemického senzoru je senzor CM-31910 pro měření oxidu uhelnatého. [9]

Polovodičové senzory využívají chemickou reakci, která nastane při přímém kontaktu plynu s povrchem senzoru. Senzor je nejčastěji tvořen oxidem cínčitým (SnO_2), kterému se při kontaktu s měřeným plynem snižuje elektrický odpor. Polovodičové senzory pro měření koncentrací plynů v okolním vzduchu jsou často schopné měřit koncentrace několika druhů plynu současně. Příkladem polovodičového senzoru je senzor MQ-7 vhodný pro měření koncentrace oxidu uhelnatého v okolním vzduchu nebo MQ-9 vhodný pro měření koncentrace oxidu uhelnatého a metanu v okolí. [36]

Oxid uhelnatý (CO)

Koncentrace oxidu uhelnatého v okolním vzduchu lze měřit již zmiňovaným elektrochemickým senzorem CM-31910, který je však mnohem dražší než alternativní polovodičové senzory MQ-2, MQ-7, MQ-9, MICS-6814 nebo TGS2442.

Oxid uhličitý (CO₂)

Pro měření koncentrace oxidu uhličitého v okolním vzduchu existují rovněž jak elektrochemické, tak polovodičové senzory. Na elektrochemické senzory oxidu uhličitého se specializuje společnost CO2Meter³, jež nabízí několik druhů senzorů, které se liší rozmezím koncentrací oxidu uhličitého ve vzduchu, jež mohou měřit. Alternativní a mnohem levnější polovodičové senzory jsou například MQ-135 nebo MG-811. [25]

Oxidy dusíku

Senzory pro měření koncentrace oxidu dusnatého (NO) jsou poměrně drahé, například senzor 4NO-250 stojí přibližně 234 €. Pro oxid dusičitý (NO₂) jsou senzory mnohem levnější, například MQ-135 nebo MICS-2710. [25]

Oxid siřičitý (SO₂)

Polovodičový senzor vhodný pro měření koncentrací oxidu siřičitého je například MQ-136. Existují také elektrochemické senzory k tomuto účelu, ale jsou opět výrazně dražší a nemají tak velkou životnost. [25]

Přízemní ozon (O₃)

Pro měření koncentrací přízemního ozonu ve vzduchu lze použít polovodičový senzor MQ-131. Vhodné jsou také senzory MICS-2610 a MICS-2614. [25]

Těkavé organické látky

Pro měření koncentrace těkavých organických látek je vhodný například senzor MICS-5524 nebo MQ-9. [36]

Formaldehyd (CH₂O)

Senzor pro měření koncentrace formaldehydu je například SEN0231, jehož cena se pohybuje okolo 50 \$. [10]

Čidla pro měření tuhých znečišťujících látek

Měření obsahu tuhých znečišťujících látek v ovzduší je založeno převážně na dvou principech:

1. Filtry pevných částic, které musejí být periodicky měněny a vyhodnocovány nejčastěji vážením nebo podrobnějším rozbořem v laboratořích.

³Webové stránky společnosti: <https://www.co2meter.com>.

2. Optická čidla, která umožňují měřit hodnoty průběžně. Senzor snímá kolísání přenosu světla, které je přerušováno pohybem částic polétavého prachu světelným paprskem. Takové senzory jsou například SamYoung DSM501 nebo Shinyei_PPD42NS. [\[34\]](#)

Kapitola 3

Návrh řešení

V následující kapitole je popsán teoretický návrh fyzického zařízení pro monitorování kvality vzduchu, jeho programu a také návrh mobilní aplikace pro vyčítání naměřených hodnot ze zařízení pomocí technologie Bluetooth.

3.1 Návrh měřicího zařízení

Základem fyzického zařízení (hardware) pro monitorování kvality vzduchu je dle návrhu vývojová deska STM32 Nucleo-64. Na tuto desku budou připojena čidla, snímající data z okolí (jako například teplotu, vlhkost, výskyt nežádoucích plynů, apod.) a dále Bluetooth modul pro následnou komunikaci s mobilní aplikací pro vyčítání naměřených dat.

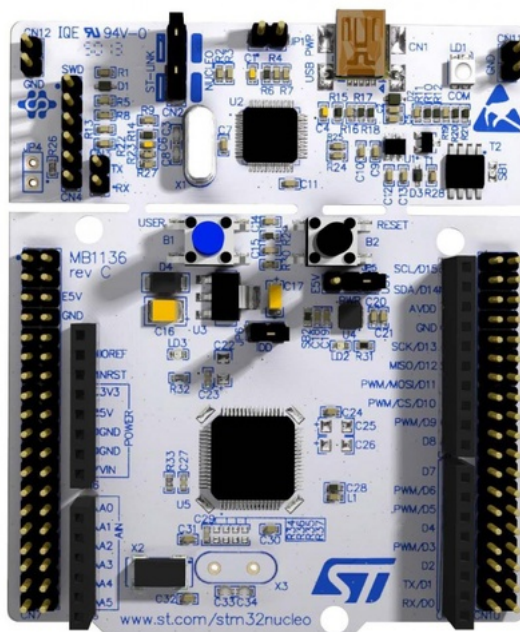
STM32 Nucleo-64

Vývojová deska STM32 Nucleo-64 (obrázek 3.1) je osazena mikrokontrolérem ARM® Cortex®-M4 (celým označením STM32F446RE) s frekvencí 84 MHz. Nucleo-64 dále obsahuje SRAM paměť o velikosti 96 KB a FLASH paměť o velikosti 512 KB. Výhodou této desky je také vestavěný programátor a debugger ST-LINK/V2-1, který umožňuje navíc přístup do zmiňované FLASH paměti přes USB a virtuálním COM portem, rovněž přes rozhraní USB.[38] Vývojová deska je také opatřena dvěma sadami konektorů: *ST morpho* a *Arduino UNO*, a to pro dosažení kompatibility s periferiemi vytvořenými jak pro vývojové desky *ST morpho*, tak pro desky *Arduino UNO*.[37]

Schopnost Nuclea konkurovat Arduinu zvyšují převážně lepší technické parametry - větší SRAM i FLASH paměť, rychlejší takt procesoru, kompatibilita i s periferiemi pro desky *ST morpho*, apod. Mnoho uživatelů navíc osloví také cena, která je u Nuclea bezmála poloviční než u Arduina. [37] [5]

Polovodičový senzor hořlavých plynů MQ-2

Princip senzoru MQ-2 spočívá v proměnné vodivosti, která se mění v závislosti na koncentraci plynů, na které čidlo reaguje. Se zvyšující se koncentrací plynů v okolí se zvyšuje vodivost, tedy snižuje elektrický odpor. K senzoru je zapojen potenciometr (velikost jeho odporu ovlivňuje citlivost senzoru). Celé toto zapojení pak funguje jako dělič napětí a koncentrace hořlavých plynů je určena měřením napětí na vývodech senzoru. Materiál, který na změnu koncentrací plynů reaguje, je oxid cínčitý (SnO_2), který má v čistém vzduchu nižší vodivost.



Obrázek 3.1: Vývojová deska STM32 Nucleo-64.[39]

Senzor MQ-2 je citlivý na LPG, propan (C_3H_8) a vodík (H_2), rovněž může být použit pro detekci metanu (CH_4), oxidu uhelnatého (CO) a jiných hořlavých výparů (alkoholu) či kouře. Senzor reaguje na zmíněné plyny společně (není například možné rozlišit, zda je velká koncentrace propanu nebo vodíku), což může být považováno za nevýhodu.

Mezi výhody senzoru MQ-2 patří dlouhá životnost, nízká pořizovací cena, jednoduché použití, rychlá reakce na změnu koncentrace plynů a dobrá citlivost na hořlavé plyny v širokém rozsahu (je možné citlivost ovlivnit pomocí potenciometru na modulu se senzorem). Rozsah měření je výrobcem stanoven od 300 do 50 000 ppm¹.

Použití senzoru je nejčastěji v zařízeních pro detekci úniku plynů v domácnostech, v průmyslových detektorech hořlavých plynů nebo v přenosných detektorech plynů. [32] [35]

Polovodičový senzor oxidu uhelnatého MQ-7

Senzor MQ-7 je svou konstrukcí i principem podobný předešlému senzoru MQ-2. Senzitivní vrstvu tvoří oxid cínčitý (SnO_2) a společně s předehřívacím odporem je vsazen do pouzdra z nerezové oceli. Senzor MQ-7 má velmi dlouhou životnost, která dosahuje až pěti let provozu. Ke správné funkci potřebuje speciální cyklus napájení předehřívacího odporu (po dobu 60 vteřin napětí 5 V a po dobu 90 vteřin napětí 1,4 V). Rozsah měření senzoru MQ-7 je 20 až 2 000 ppm. Využívá se v domácích nebo průmyslových detektorech oxidu uhelnatého. [16]

¹ppm z anglického „Parts Per Milion“ je jednotka pro měření koncentrací látek. 1% koncentrace látky je rovno 10 000 ppm.

Modul teploty a relativní vlhkosti vzduchu RHT03

RHT03, známý též jako DHT22 nebo AM2302, je digitální modul se snímači relativní vlhkosti a teploty okolního vzduchu, které jsou na tomto modelu teplotně kompenzovány a kalibrovány ve speciálních kalibračních komorách, kde je stanoven kalibrační koeficient. Ten je uložen do OTP paměti modulu RHT03. Součástí modulu je i jednočipový osmibitový procesor, který při měření využívá koeficient z paměti k dosažení co nejpřesnějšího výsledku měření.

Rozsah měření modulu RHT03 pro relativní vlhkost vzduchu je 0 – 100% s odchylkou $\pm 2\%$. Teplotu je schopný modul měřit v rozsahu od -40°C do 80°C s odchylkou menší než $\pm 0,5^{\circ}\text{C}$.

Výhodou modulu RHT03 je nízká cena, malé rozměry, nízká spotřeba energie a jednoduché ovládání, což činí tento modul vhodným pro široké spektrum aplikací. [4]

Bluetooth modul

Modul s rozhraním Bluetooth je možné použít všude tam, kde je z jakýchkoliv důvodů nevhodné používat přenos dat připojeným kabelem. Díky tomuto modulu je možné komunikaci po kabelu nahradit sériovou bezdrátovou komunikací až na vzdálenost deseti metrů v otevřeném prostředí. Přenos dat pomocí technologie Bluetooth však není příliš rychlý. Ve verzi 2.0 je standardem uváděná maximální teoretická přenosová rychlost 3 Mbit/s, ve verzi 4.0 pak 24 Mbit/s. [19]

Technologie Bluetooth

Bluetooth je technologický standard používající rádiové spojení krátkého dosahu namísto kabelů pro propojení elektronických zařízení. Standardy Bluetooth specifikují jednotnou strukturu pro komunikaci mezi širokou škálou elektronických zařízení, a to s minimálním úsilím uživatele. Jinými slovy je možné Bluetooth popsat jako robustní, jednoduchou a ekonomickou technologii, která umožňuje bezdrátový přístup do místních sítí (LAN), veřejných telefonních sítí (PSTN), mobilních sítí a internetu. [14]

Mezi technologie konkurující standardu Bluetooth se řadí zejména WLAN – tedy bezdrátová lokální síť (Wireless Local Area Network). Stejně jako Bluetooth, WLAN dodržuje standard IEEE 802.11², její použití je však v mnoha případech výhodnější, protože umožňuje vyšší přenosové rychlosti na větší vzdálenost. Další konkurenční technologií je IrDA – přenos dat pomocí infračerveného světla (Infrared Data Association). Velkou nevýhodou IrDA je možnost spojení pouze dvou komunikujících zařízení najednou (point-to-point komunikace) a také nutnost přímého vizuálního kontaktu mezi komunikujícími zařízeními. V neposlední řadě můžeme vzpomenout technologii UWB – ultra-širokopásmové rádio (Ultra-Wideband). Tato technologie je využívána zejména pro armádní účely (radary). Umožňuje přenos velkého objemu dat na krátkou vzdálenost, a to s velmi nízkou energetickou spotřebou. [14]

²Více informací o standardu IEEE 802.11 na: <http://www.ieee802.org/11/>.

Bluetooth Low Energy

Standard Bluetooth 4.0 Low Energy³ upravuje klasický Bluetooth ve verzi 2.0. Největší změnou je snížení energetické náročnosti a úprava komunikačních standardů. Řada nových modulů pro práci s Bluetooth podporuje pouze standard Bluetooth LE.

Bluetooth LE rozděluje zařízení na periferie a centrální zařízení. Periferie jsou např. senzory a centrální zařízení je např. mobilní telefon. Periferie mohou pracovat ve dvou módech: přímé spojení s centrálním zařízením nebo všesměrové vysílání, což je jedna z největších změn od klasického Bluetooth 2.0. Periferie může být v jednu chvíli přímo spojena pouze s jedním zařízením. Po připojení k centrálnímu zařízením se pak periferie přestane všesměrově propagovat a jiná centrální zařízení nemají možnost se k periférii připojit, ani ji vidět mezi dostupnými Bluetooth zařízeními. Centrální zařízení k sobě však může mít více připojených periférií zároveň.

Nový standard také definuje dva nové pojmy, a to *služby* a *charakteristiky*. Služby jsou používány k rozdělení dat na logicky asociované kusy (v anglické literatuře „chunks“) a skládají se z kolekce charakteristik. Charakteristiky jsou pak jakési kontejnery uchováující data patřící dané službě. Skládají se minimálně ze dvou atributů, a to deklarace charakteristiky (uchováující metadata, tedy informace o datech) a hodnoty charakteristiky (samotná data). Hlavní vlastnosti charakteristik mohou být následující:

- *read* – pro čtení dat
- *write*, případně *writeWithoutResponse* – pro zápis dat
- *notify* – pro upozornění na změnu dat, ta jsou pak čtena podobně jako u charakteristiky *read*
- *indicate* – podobné jako *notify*, ale s potvrzením o přijetí upozornění na nová data

Každá charakteristika pak může mít více vlastností, např. jedna charakteristika může umožňovat čtení, zápis i upozornění na změny. [1]

Modul HC-05

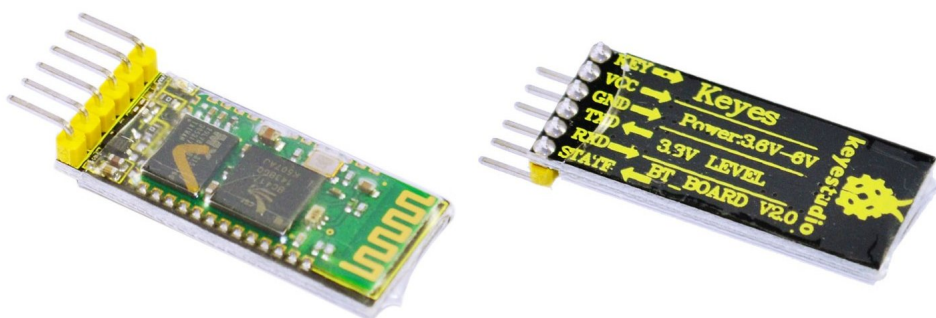
Modul HC-05 využívá klasický Bluetooth, tedy verzi 2.0. Pro ochranu proti prachu a ochranu před nežádoucí statickou elektřinou je celý modul od výrobce Keyestudio (obrázek 3.2) zataven v teplem se smršťující bužírce. Jádrem modulu používá HC-05 slave modul.

Součástí Bluetooth modulu je i LED dioda, která indikuje stav připojení Bluetooth. Rychlé blikání znamená žádné připojení, pomalé blikání indikuje přechod do módu AT (Bluetooth modul obsahuje celkem čtyři AT příkazy – test komunikace, přejmenování, změnu komunikační rychlosti a změnu hesla). Dvojblik diody značí, že je zařízení připojeno přes Bluetooth a port pro komunikaci je otevřený.

Modul vyžaduje napájecí napětí 3,6 až 6V, pro předejití následků změny polarit napájecího napětí obsahuje deska ochrannou diodu. Proud činí cca 30mA bez spárování, 10mA po úspěšném spárování.

Po spárování je možné komunikovat jako po obousměrné (full duplex) sériové lince, přičemž není nutné znát žádné Bluetooth protokoly. Formát dat, který modul podporuje, je 8 bitů pro data, 1 stop bit a je možné navíc nastavit ještě kontrolní bit pro sudou nebo lichou paritu. [19]

³Často zkracovaný na Bluetooth LE nebo jen BLE.



Obrázek 3.2: Bluetooth modul HC-05.[19]

Modul HM-10

Modul HM-10 je svou konstrukcí velmi podobný modulu HC-05, zásadní rozdíl je však v používání nového standardu Bluetooth Low Energy. Nabízí možnost sériové komunikace pomocí technologie Bluetooth. Podobně jako HC-05 lze pomocí AT příkazů měnit komunikační rychlost, nastavovat formát odesílaných dat (kontrolní bity) nebo název, kterým se bude zařízení všesměrově propagovat. [18]

Zařízení nabízí celkem čtyři služby, z toho pouze jedna obsahuje charakteristiku s vlastnostmi *write* a *writeWithoutResponse*, právě tato bude použita pro komunikaci s mobilní aplikací.

3.2 Návrh programu zařízení

Pro programování mikropočítače STM Nucleo F446RE se používá ve světě hardwaru velmi rozšířený imperativní programovací jazyk C.

Návrh programu

Nejprve je třeba nastavit, na kterých analogových a digitálních pinech jsou připojeny periferie (tedy senzory a Bluetooth modul) a určit přenosovou rychlost Bluetooth komunikace na 9600 baudů⁴. Následně bude program v nekonečné smyčce v určitých časových intervalech vyčítat hodnoty ze senzorů. Hodnoty z analogových senzorů MQ-2 a MQ-7 je potřeba vhodně interpretovat, a to vynásobením surové hodnoty ze senzoru kalibrační konstantou (ta bude určena měřením známé koncentrace plynů v okolním vzduchu). Po získání měřených hodnot, budou tyto hodnoty zapsány do paměti.

Program bude čekat na požadavek mobilní aplikace k odeslání naměřených dat. Po jeho příchodu budou data z paměti odeslána mobilní aplikaci. Jakmile bude odesílání dokončeno, budou data z paměti měřicího zařízení odstraněna.

⁴Baud [bd] je jednotka modulační rychlosti, udávající počet změn stavu přenosového média za jednu sekundu.

Zápis do paměti

Aby byla naměřená data zachována i při výpadku napájení, je potřeba data zapisovat do tzv. nevolatilní paměti, tedy do paměti, která uchovává svůj obsah i po odstavení napájení. V mikropočítačích se jako nevolatilní paměti používají nejčastěji paměti EEPROM a Flash. Mikropočítač STM Nucleo F446RE je vybaven Flash pamětí o velikosti 512 KB.

3.3 Návrh mobilní aplikace

Platformou pro mobilní aplikaci bude operační systém Android. Hlavním účelem aplikace bude vyčtení dat z hardwarového zařízení a jejich prohlížení.

Operační systém Android

Android je open-source platforma na bázi Linuxu určená především pro mobilní zařízení, tedy chytré telefony, tablety, navigace, ... V současnosti se stále více přidávají i konvertibilní zařízení, tj. tablety s připojitelnou klávesnicí, které se hodí i k méně náročnému podnikovému nasazení. Platforma Android se objevuje i v televizních přijímačích a různých dalších zařízeních.

Systém Android vyvíjí organizace Open Handset Alliance, jejíž součástí jsou desítky firem včetně těch nejznámějších ve světě mobilních zařízení – Google, HTC, Intel, NVIDIA, Qualcomm, Samsung a další. Jde o jeden z mála operačních systémů, které podporují více platforem, můžeme jej vidět v zařízeních nejrozličnějších značek. To však přináší jednu značnou nevýhodu – chybí optimalizace systému na konkrétní hardwarovou platformu, což je silná zbraň Apple iOS. Android je však multiplatformní s možností přizpůsobení a vytvoření nadstavby (HTC Sense, Samsung TouchWiz). Na druhé straně, když Google vydá aktualizaci systému, ta není hned dostupná pro všechna zařízení, uživatel si musí počkat na konkrétní aktualizaci od výrobce jeho zařízení, takže se u různých zařízeních můžeme setkat s různými verzemi Androidu.

Největší výhodou a zároveň nevýhodou platformy je její otevřenost a možnost úprav, ať už ze strany výrobců konkrétních zařízení nebo jejich uživatelů. Úpravy se netýkají pouze konfigurace či widgetů, ale i firmwaru. Pro Android je k dispozici více aplikací než pro jakoukoliv jinou platformu, mnohé jsou však pochybné kvality, jelikož proces schvalování není tak přísný jako u Windows či iOS. Tablety a telefony s Androidem dodává na trh velké množství firem. Je to zárukou dynamičtějšího vývoje nových zařízení, například v porovnání s Applem, kde celý vývoj hardwaru řídí jedna společnost, a zároveň to představuje problém, protože aplikace běží na přístrojích s různým rozlišením displeje a různým výkonem procesoru a grafické karty. V praxi to znamená různý komfort uživatelského rozhraní. [21]

Historie Androidu a jeho verze

Společnost Android Inc. byla založena čtyřmi zakladateli v kalifornském Palo Alto v říjnu roku 2003. O dva roky později, v srpnu 2005, odkoupila Android Inc. společnost Google. V listopadu 2007 byla založena Open Handset Alliance, která stojí za vývojem Androidu dodnes. V té době vyšel zároveň vývojářský kit SDK. V září 2008 přišel v USA na trh HTC Dream (G1), první chytrý telefon s Androidem 1.0, který měl na trhu inteligentních tele-

fonů zastoupení pouhých 0.5%. V dubnu 2009 spatřil světlo světa první masový Android ve verzi 1.5.

- **1.5 Cupcake** – nabízí podporu virtuálních klávesnic třetích stran s podporou vlastních slovníků, nahrávání a přehrávání videa (formáty MPEG-4 a 3GP), možnost kopírovat a vložit obsah přes schránku, možnost umístit widgety na domácí obrazovku, animace přechodů mezi obrazovkami, ...
- **1.6 Donut** – integrované vyhledávání Google, aplikační Market obohacený o obrázky a hodnocení uživatelů, podpora obrazovek s vyšším rozlišením (WVGA), podpora VPN, ...
- **2.0 / 2.1 Eclair** – nový design uživatelského prostředí, optimalizace výkonu, podpora standardů HTML5, Bluetooth 2.1, podpora velkého množství rozlišení displejů, ...
- **2.2 Froyo** – webový prohlížeč s podporou JavaScriptu, podpora Flash 10, možnost instalovat aplikace i na paměťovou kartu, sdílení kontaktů pomocí Bluetooth, významná optimalizace používání paměti a celkového výkonu, ...
- **2.3 Gingerbread** – vylepšená správa napájení, podpora dalších senzorů (gyroskop, barometr), podpora internetových hovorů (VoIP), vylepšená virtuální klávesnice, ...
- **3.0 Honeycomb** – první verze určená pouze pro tablety, vylepšené uživatelské rozhraní, zjednodušené notifikace, přizpůsobitelná domácí obrazovka, hardwarová akcelerace, podpora USB příslušenství, ...
- **4.0 IceCream Sandwich** – tato verze odstraňuje rozdíly mezi verzemi pro mobilní telefony a pro tablety. Na rozdíl od předchozích verzí obsahuje aplikace na obrazovce uzamčení, nedávno spuštěné aplikace, vytváření adresářů, možnost zrušit notifikace, zastavení aplikací na pozadí, ...
- **4.2 Jelly Bean** – vylepšené notifikace, podpora uživatelských účtů s možností jejich přepínání, widgety na obrazovce uzamčení, vylepšená aktualizace aplikací, výrazné zrychlení vykreslování obrazu, ...
- **4.4 KitKat** – vyšší výkon, vylepšená podpora zařízení s vícejádrovými procesory, rychlejší multitasking, podpora více cloudových služeb, fullscreen režim pro několik aplikací, podpora infračerveného ovládání, inteligentní vypínání nepotřebných procesů na pozadí, ...
- **5.0 / 5.1 Lollipop** – nahrazení virtuálního stroje Dalvik, pod kterým běžely aplikace, novým enginem ART (Android Runtime). Přínosem ART je možnost částečně kompilovat aplikace už při jejich instalaci, což významně urychlí spouštění aplikací a sekundárně prodlouží výdrž baterie, jelikož Dalvik vykonával proces kompilace při každém spuštění aplikace. Notifikace je nově možné prohlížet i na obrazovce uzamčení, inovovaný systém multitaskingu (pomocí „karet“), ... [21]
- **6.0 Marshmallow** – odemykání zařízení pomocí čtečky otisků prstů, uživatel může nově měnit oprávnění aplikací i za jejich běhu, optimalizace spotřeby energie – výdrž baterie se výrazně prodloužila, ... [27]

- **7.0 / 7.1 Nougat** – optimalizace využití paměti a zvýšení úspory baterie, změny v uživatelském rozhraní pro zpřístupnění uživatelům se špatným zrakem, změny v oprávnění pro možnost podnikového využití zařízení, možnost práce s více okny zároveň (split screen), ... [28]
- **8.0 Oreo** – inovace navigace a zadávání na virtuální klávesnici, limit využití výkonu pro aplikace na pozadí za účelem optimalizace výdrže baterie, podpora Bluetooth 5, ... [29]
- **preview P** – v současné době je k dispozici pouze náhled klíčových změn nejnovější verze operačního systému Android. Plánují se vylepšení bezpečnosti jako například omezení přístupu aplikacím na pozadí ke zdrojům (kamera, mikrofon), lokalizace uvnitř budov pomocí Wi-Fi RTT, podpora displeje pokrývajícího celou přední stranu zařízení, ... [30]

O dynamickém vývoji Androidu svědčí i skutečnost, že odstup mezi verzemi je jen několik měsíců. Číslování verzí operačního systému nekoresponduje s číslováním API (číslo API je k jednotlivým verzím přiřazené v tabulce na obrázku 3.3). [21]

Jakou nejstarší verzi podporovat v aplikaci

Doba životnosti mobilního telefonu je maximálně dva až tři roky, podobně je tomu i u tabletů, takže přístrojů se staršími verzemi rapidně ubývá. Společnost Google se neustále snaží zdokonalovat uživatelské rozhraní Androidu. Dnes aplikace využívají nejnovějších prvků uživatelského rozhraní. Pro vývojáře je výhodné, aby nové aplikace vypadaly a fungovaly stejně i na starších verzích Androidu. K tomu je potřeba přidávat do projektů knihovny Android Support Library k dosažení tzv. dopředné kompatibility.[21]

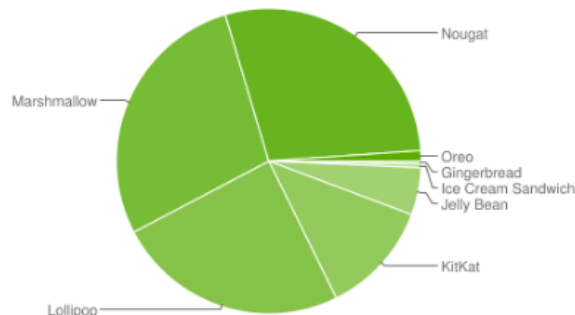
Z grafu na obrázku 3.3 je patrné, že většina uživatelů Androidu má v dnešní době ve svém zařízení verzi 5.0 Lollipop a nebo vyšší. Z tohoto důvodu je dostatečné vytvářet aplikace směřující k uživatelům této nebo vyšší verze. Verze 5.0 je vhodná i pro svoji již zmíněnou změnu výpočetního enginu.

Architektura Androidu

Pro obecný přehled při tvorbě aplikací je vhodné znát základní principy architektury operačního systému Android. Následující popis bude od nejnižší architektonické vrstvy po tu nejvyšší, tak jak je naznačeno i na obrázku 3.4.

Nejnižší vrstvou architektury Androidu je upravené jádro operačního systému Linux – *Linux Kernel*. Úpravy se týkají redukce funkcí a jejich přizpůsobení možnostem mobilních zařízení. Jádro slouží k přímé interakci s hardwarem mobilního zařízení, čímž pro vyšší softwarové vrstvy zajišťuje úplnou abstrakci od hardwaru. Zabezpečuje správu paměti, správu procesů (možnost multiprocessingu – více běžících procesů současně), základní síťovou vrstvu, bezpečnost, vstupně-výstupní operace a správu napájení a ovladače podle hardwarové konfigurace (3G, EDGE, Bluetooth, Wi-Fi, kamera, akcelerometr, ...). Aplikace a služby jsou spouštěny v oddělených procesech, které ale často potřebují vzájemně komunikovat. Na druhé straně představuje možnost přímé komunikace mezi procesy jedno z největších bezpečnostních rizik. Proto je na platformě Android implementován meziprocetový ovladač komunikace a volání metod zvaný *Binder*, který umožňuje více procesům sdílet údaje. Údaje se odevzdávají jako balíčky přes sdílenou paměť. *Binder* mapuje vzájemné reference napříč systémem, takže přesouvané objekty mohou být

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.7%
4.2.x		17	2.6%
4.3		18	0.7%
4.4	KitKat	19	12.0%
5.0	Lollipop	21	5.4%
5.1		22	19.2%
6.0	Marshmallow	23	28.1%
7.0	Nougat	24	22.3%
7.1		25	6.2%
8.0	Oreo	26	0.8%
8.1		27	0.3%



Obrázek 3.3: Zastoupení jednotlivých verzí androidu k únoru 2018.[3]

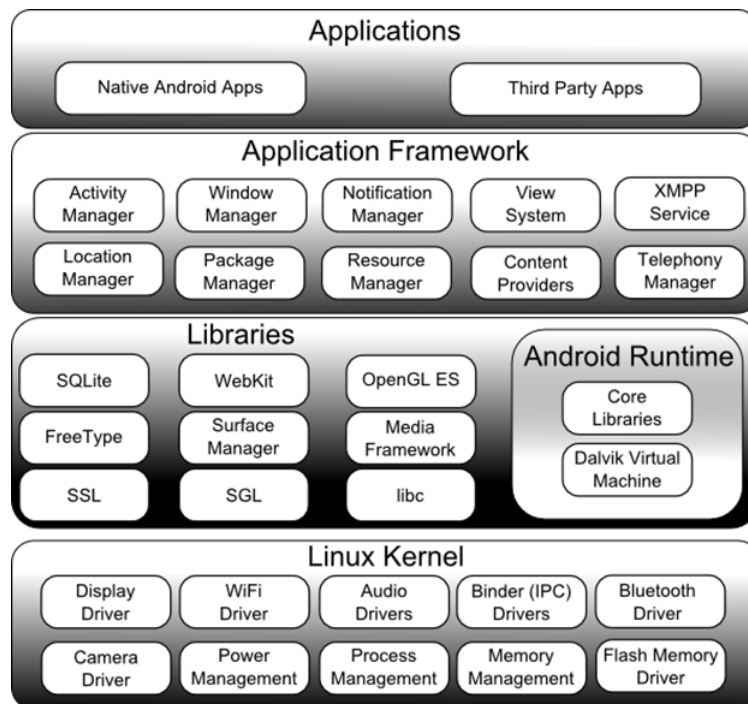
případně identifikovány, vysledovány a zrušeny. Funguje to tak, že se každá aplikace při spuštění zaregistruje ve službě Service Manager. Když potřebuje nějaký proces komunikovat s jiným, osloví Service Manager. Ten poskytne procesu potřebné informace a následně probíhá komunikace přímo mezi procesy, přičemž Binder plní roli zprostředkovatele.

Nad jádrem je situována vrstva knihoven – *Libraries*, které aplikacím poskytují přímý přístup k různým komponentám operačního systému Android. Jsou to nativní⁵ knihovny psané v C/C++. Tvoří mezikomponentu mezi linuxovým jádrem a různými komponentami vyšších vrstev. Knihovny v této vrstvě suplují linuxové funkce, o které bylo jádro operačního systému oprostěno. Do vrstvy knihoven patří například Modul Surface manager, který se stará o funkcionalitu multidotykového displeje, zabezpečuje výslednou kompozici grafického výstupu více aplikací do souvislého toku dat, který směřuje do grafické vyrovnávací paměti a je následně vykreslován na obrazovku. Dalším příkladem je systémová knihovna LibC, která je přizpůsobená použití v mobilních zařízeních, takže obsahuje jen části, které jsou pro Android potřeba.

Následující vrstva – *Android Runtime* – obsahuje sadu základních knihoven. Každá aplikace pro Android je samostatný proces využívající vlastní instanci virtuálního stroje Dalvik⁶. Tento se stará o běh spustitelných souborů s příponou DEX (ty vznikly kompilací ze souborů CLASS a JAR). Dalvik je optimalizován pro mobilní zařízení, bere tedy v úvahu omezení možnosti napájení a podobně. Současně může běžet více instancí virtuálního stroje.

⁵Nativní aplikace se vyznačují velkou schopností přizpůsobit se různým mobilním zařízením.

⁶Dalvik Virtual Machine na mobilních zařízeních je analogií Java Virtual Machine na klasických počítačích, na rozdíl od JVM jsou aplikace spouštěny ve svých procesech.



Obrázek 3.4: Diagram architektury operačního systému Android.[2]

V pořadí „od spodu“ je čtvrtou vrstvou *Application Framework*. Ten obsahuje v aplikacích opakovaně použitelný software (ovládací prvky, ikony, ...). Framework je napsán v Javě a je to nejdůležitější vrstva pro vývojáře aplikací, neboť jim poskytuje základní služby systému pro použití ve svých projektech. Klíčové služby jsou například Activity Manager, Resource Manager, Notification Manager, View System a další. Tato vrstva je navržena tak, aby byly její komponenty snadno použitelné a vyměnitelné uživatelem.

Poslední – nejvyšší – vrstvou architektury operačního systému Android jsou *Aplikace*, například program pro posílání zpráv, kalendář, přehrávač hudby a podobně. Běžný uživatel Androidu interaguje povětšinou pouze s touto vrstvou.[21]

Bezpečnost na platformě Android

Android je open-source platforma, která má robustní bezpečnost a přísnou bezpečnostní politiku. Tento operační systém převzal některé základní mechanismy zabezpečení z Linuxu a byl navržen s vícevrstvou bezpečností, která poskytuje flexibilitu vyžadovanou pro otevřenou platformu.

Důležitým bezpečnostním prvkem je tzv. Sandbox pro izolovaný běh aplikací. Každá aplikace je spouštěna s vlastním přiděleným UID v samostatném procesu. Aplikace dostanou přidělenou část souborového systému, kam mohou zapisovat soukromá data. Od verze 3.0 je navíc možné souborový systém šifrovat, veškerá uživatelská data mohou být zabezpečena v jádře použitím dm-crypt⁷ implementace AES-128.

⁷ dm-crypt je specifický subsystém linuxového jádra, který vytváří transparentní vrstvu pro online šifrování a dešifrování blokových zařízení.[11]

Z pochopitelných důvodů aplikace nemohou přistupovat k některým údajům a komponentám (např. pořizovat fotografie nebo odesílat SMS), pokud k takovému přístupu nemají povolení.

Oprávnění pro aplikace na přístup k různým službám mobilního zařízení, které spadají do *Accessing Protected APIs*, musí být deklarována v manifestu⁸. Tato práva jsou potom vynucována běhovým prostředím systému. Mezi takto chráněné API patří například použití kamery, funkcionality telefonování, síťové konektivity nebo komunikačního rozhraní Bluetooth a podobně. Oprávnění přistupovat k těmto zdrojům uděluje aplikacím sám uživatel při jejich instalaci. [21]

Možnosti vývoje aplikací pro platformu Android

Vyvíjet aplikace pro platformu Android je možné několika prostředky jako například programovací jazyk Python nebo Ruby. Klasický a široce rozšířený způsob tvorby mobilních aplikací je programování v jazyce Java pomocí Android Studia⁹. Mě však oslovila novější technologie NativeScript¹⁰, přinášející možnost snadné rozšiřitelnosti vytvořené aplikace i pro platformu iOS.

NativeScript

NativeScript je open-source framework pro tvorbu mobilních aplikací napříč platformami Android a iOS. Stará se o něj společnost Telerik¹¹. Od ostatních frameworků pro tvorbu mobilních aplikací se NativeScript liší hned v několika aspektech, přičemž nejvýznamnější je skutečnost, že díky NativeScriptu můžeme tvořit nativní aplikace napříč platformami, a to s jednou základnou zdrojového kódu. Dalším aspektem je snadný přechod z jiných programovacích jazyků. Logika aplikace NativeScriptu (back-end) je psaná v jazyce JavaScript nebo TypeScript¹², uživatelské rozhraní (front-end) je definováno ve značkových jazycích XML nebo HTML s použitím kaskádových stylů CSS, což jsou pro velkou část i začínajících programátorů známé technologie.[7]

Velkou výhodou NativeScriptu je také jednoduchá obsluha asynchronních procesů, která je při vývoji aplikací pro mobilní zařízení nezbytná. Protože je v NativeScriptu pro psaní programů použitý JavaScript, je zde výhodné využívat tzv. *promise*. Objekt promise reprezentuje eventuální dokončení (nebo selhání) asynchronní operace a hodnotu jejího výsledku. Umožňuje přidružit tzv. *handler* (obsluhu) k eventuálnímu úspěchu asynchronní operace, nebo k jejímu selhání. Toto u asynchronních funkcí zajišťuje navrácení hodnoty podobně jako u klasických synchronních funkcí, avšak hodnota není navržena ihned, ale její navrácení je přislíbeno do budoucna (odtud název promise).

Promise nabývá jednoho ze tří stavů:

- *pending* – počáteční stav, asynchronní operace ještě není vyhodnocena

⁸Manifest je nutná součást aplikace pro android, kde je vývojář povinen definovat všechny prostředky, které bude aplikace využívat. Je zapsán v souboru `AndroidManifest.xml` v elementu `<uses-permission>`.

⁹Android Studio je oficiální IDE (vývojové prostředí) pro Android, je odvozené z prostředí IntelliJ IDEA. <https://developer.android.com/studio/index.html>

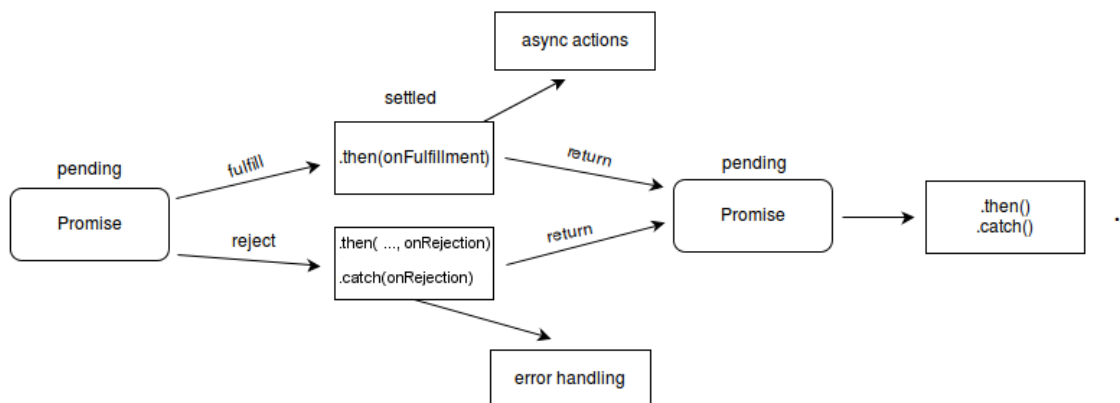
¹⁰Webové stránky NativeScriptu: <https://www.nativescript.org/>

¹¹Webové stránky společnosti Telerik: <https://www.telerik.com/>.

¹²TypeScript je programovací jazyk vyvinutý společností Microsoft, který je velmi podobný JavaScriptu, na rozdíl od něj je však třídí objektové orientovaný a typovaný.[17]

- *fulfilled* – asynchronní operace proběhla úspěšně, je navržena hodnota
- *rejected* – asynchronní operace selhala

Po vyhodnocení asynchronní operace jsou volány přidružené služby z fronty objektu promise, do které byly vloženy metodou `then()`. Průběh vyhodnocení promise je vyobrazen na obrázku 3.5. [24]



Obrázek 3.5: Průběh vyhodnocení objektu promise v jazyce JavaScript.[24]

Architektonický vzor MVC

Architektonický vzor MVC tedy Model-View-Controller¹³ odděluje část programu definující a reprezentující data (například přístupy do databáze) od částí zajišťujících logiku programu a části, která má na starost zobrazení výsledků. MVC tedy striktně odděluje front-end aplikace od back-endu, což zajišťuje vysokou flexibilitu a znovupoužitelnost celého řešení.[31]

Samotný framework NativeScript je na architektonickém vzoru MVC postavený. *Model* v takových aplikacích představuje většinou databáze (případně jinak uložená data) a ji obsluhující kód programu. *View* je grafické uživatelské rozhraní, v NativeScriptu implementované ve značkovacím jazyce XML. A *Controller* tyto dvě části aplikace propojuje a obsahuje logiku celé aplikace.[40]

Aby mohl být využitý celý potenciál architektonického vzoru MVC, používá se v NativeScript aplikacích navázání dat z kontroléru na prvky uživatelského rozhraní, tzv. *data binding*. Právě data binding zajišťuje komunikaci mezi jinak striktně odděleným uživatelským rozhraním a kontrolérem. Objekty navázané na prvky uživatelského rozhraní jsou označovány slovem *Observable* a jsou v nich pozorovány změny. Pozorovatelné objekty jsou instancí třídy *Observable*. Pokud v nich nastanou nějaké změny, je zajištěna aktualizace všech na ně navázaných prvků uživatelského rozhraní. V NativeScript XML se definuje data binding jako jméno pozorovaného objektu ve dvou složených závorkách, např.: `<Label text="{ { note } }"/>`. [40]

¹³V dokumentaci NativeScriptu je tento vzor označován jako MVVM tedy Model-View-ViewModel.

Funkcionality aplikace

Mobilní aplikace má za cíl vyčíst a reprezentovat data nashromážděná měřicím zařízením. Aplikace bude mít čtyři hlavní obrazovky: *seznam záznamů*, *přidání nového záznamu*, *editace názvu a poznámky záznamu* a *detailní zobrazení záznamu*.

V první obrazovce aplikace – seznam záznamů – bude mít uživatel možnost přejít na načtení nového záznamu a přejít na detailní zobrazení vybraného záznamu. Tato obrazovka bude zároveň i domovskou obrazovkou aplikace.

Ve druhé obrazovce budou dvě pole pro vložení textu, do prvního uživatel zadá název nového měření, do druhého pak poznámku. Potvrzením se inicializuje Bluetooth komunikace mezi mobilním telefonem a měřicím zařízením, a toto zařízení odešle naměřená data do telefonu, kde budou uložena do souboru v serializačním formátu JSON¹⁴.

Další obrazovka aplikace – detailní zobrazení záznamu – bude obsahovat grafickou reprezentaci naměřených hodnot v podobě tabulky a grafů. Na této obrazovce bude mít uživatel možnost měření odstranit.

Poslední obrazovka bude sloužit k editaci informací zadaných při čtení dat – název a poznámka k měření.

¹⁴JSON – JavaScript Object Notation – Jednoduchý a široce rozšířený formát pro serializaci dat. (<https://www.json.org/>)

Kapitola 4

Implementace

Implementace měřicího zařízení je rozdělena do tří hlavních částí: sestavení fyzického zařízení ze zvolených modulů, tvorba programu pro obsluhu měřicího zařízení a nakonec tvorba mobilní aplikace pro vyčítání zaznamenaných hodnot a jejich grafickou interpretaci.

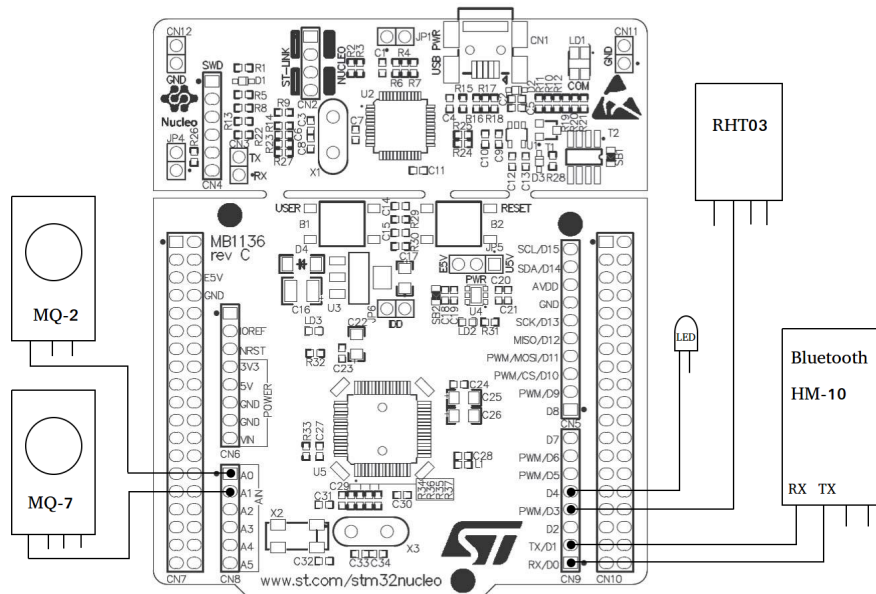
4.1 Sestava měřicího zařízení

Řízení měřicího zařízení zajišťuje mikropočítač STM Nucleo F446RE, který vykonává program zapsaný v jazyce C. Ten obsluhuje tři senzory, které měří čtyři veličiny – teplotu, relativní vlhkost vzduchu a koncentrace hořlavých plynů a oxidu uhelnatého ve vzduchu. Dále je součástí měřicího zařízení Bluetooth LE modul HM-10 a v poslední řadě LED dioda indikující práci s pamětí, při které by nemělo být zařízení odpojeno od napájení, jinak může dojít ke znehodnocení naměřených dat. Zapojení je provedeno na nepájivém propojovacím poli podle zjednodušeného schématu na obrázku 4.1, které zobrazuje pouze datové a řídicí kontakty.

Napájení

Pro napájení měřicího zařízení bylo původně zamýšleno použít modul YwRobot POWER MB V2. Výhodou tohoto modulu je jeho konstrukce navržená tak, aby se dal vložit na nepájivé propojovací pole, které má po svých stranách napájecí vývody, které pak díky tomuto modulu budou napájeny. Navíc si uživatel může volit mezi napájecím napětím 5 V a 3,3 V. Tento modul má však maximální výstupní proud pouze 800 mA. K napájení tohoto modulu byl použit zdroj s maximálním výstupním proudem 850 mA při napětí 9 V. Přestože naměřený proud při zátěži činil 0,51 A, napájecí modul se při tom velmi zahříval, což by zkreslovalo měření teploty.

Řešením je použití výkonnějšího napáječe, které však často mívají vyšší napětí. Pro napájení měřicího zařízení byl tedy použit napáječ dříve určený pro napájení malého notebooku. Jeho maximální výstupní proud jsou 3 A při napětí 12 V. Ke snížení napětí byl použit tzv. step-down měnič s řídicím čipem LM2596, který napětí na vstupu z rozsahu 5 V až 45 V snížil na napětí v rozsahu 3 V až 40 V, a to při maximálním proudu 3 A. Výše výstupního napětí je nastavena víceotáčkovým potenciometrem, pro účely této práce bylo nastaveno napětí 5 V. Tímto napětím jsou pak napájeny všechny senzory i mikropočítač. Pouze Bluetooth modul a ventilátor potřebují napětí 3,3 V, které dodává mikropočítač.



Obrázek 4.1: Zjednodušené schéma zapojení periférií k mikro počítači.

Šasi měřicího zařízení

Celé měřicí zařízení, kromě zdroje napětí, je umístěno v plastové krabici o vnějších rozměrech 70 x 120 x 180 mm, která splňuje stupeň krytí IP 31, zařízení však bylo úspěšně otestováno i za silného svíslého deště (obrázek 5.1).

Na dvou stěnách krabice jsou otvory pro přístup měřeného vzduchu, které jsou překryty stříškou pro dosažení odolnosti vůči dešti. U jedné stěny s otvory jsou umístěny senzory a na protější straně je ventilátor, který zajišťuje výměnu vzduchu uvnitř zařízení.

Na přední stěnu je vyveden konektor pro napájení a LED dioda indikující práci s pamětí.

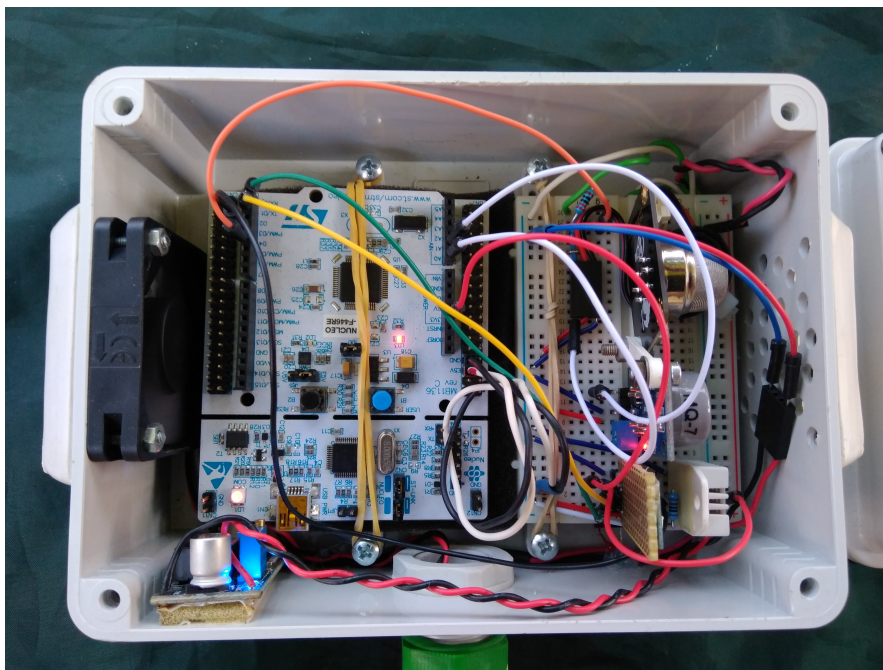
Výsledná sestava

Výsledkem sestavení je monitorovací zařízení (obrázek 4.2) vytvořené na mikrokontrolérové platformě STM Nucleo, které měří, zaznamenává a prostřednictvím technologie Bluetooth odesílá hodnoty vybraných ukazatelů kvality vzduchu do mobilní aplikace.

Kalibrace senzorů

Podle informací uvedených výrobcem je pro správnou funkci potřeba senzory MQ-2 a MQ-7 zkalibrovat. Rovněž senzory, které byly kalibrovány v laboratořích, je vhodné opětovně zkalibrovat s uvážením prostředí, ve kterém budou použity. Může totiž docházet k interferenci s jinými látkami, než které by měl sensor detekovat. Senzory také stárnou a časem mohou měnit své vlastnosti. [13]

U senzoru hořlavých plynů a kouře s označením MQ-2 je kalibrace nezbytná, protože modul s tímto senzorem obsahuje potenciometr, který mění citlivost a tedy i rozsah měření. Kalibrace byla pro účely této práce provedena v domácích podmínkách měřením



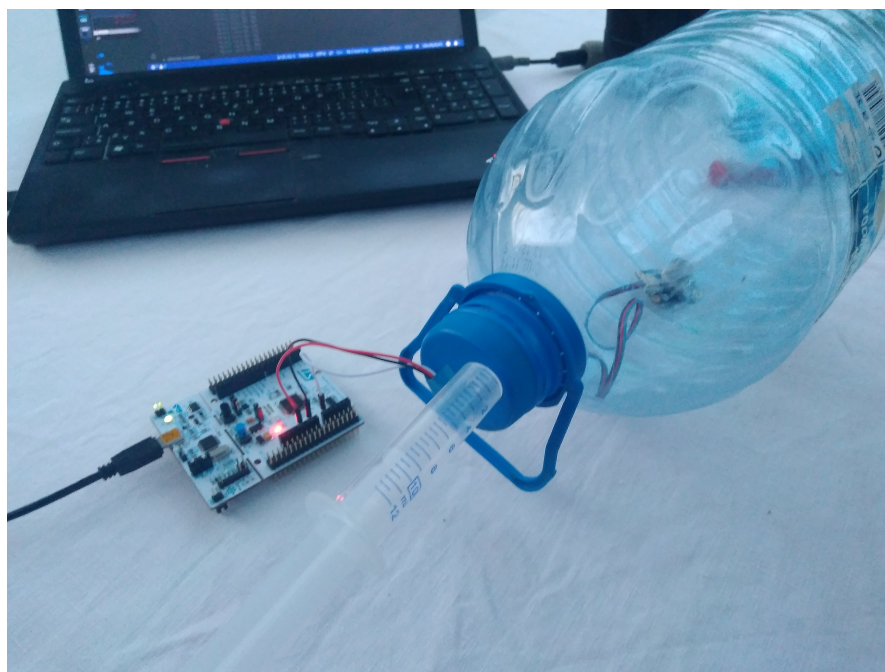
Obrázek 4.2: Vytvořené monitorovací zařízení.

obsahu propanu ve vzduchu v uzavřené pětilitrové lahvi, jak je vidět na obrázku 4.3. Mikropočítač STM Nucleo každou vteřinu snímá data z modulu se senzorem MQ-2, který byl uzavřen v čisté lahvi naplněné okolním vzduchem. Snímané hodnoty mikropočítač odesílá přes USB na sériový monitor do počítače. Senzor byl umístěn do lahve, aby byla zaznamenána analogová hodnota (přepočítané napětí na analogovém výstupu modulu) pro okolní vzduch bez zvýšené koncentrace hořlavých plynů – hodnota byla průměrně 350. Do injekční stříkačky byl nabrán čistý propan a malým otvorem ve víčku byl do lahve přidán 1 ml čistého propanu. V lahvi bylo tedy docíleno koncentrace 200 ppm propanu ve vzduchu. S takovou koncentrací byla snímána analogová hodnota průměrně 950. Následně byl senzor vyjmut, v lahvi byl pomocí kompresoru vyměněn vzduch a senzor byl opět zaveden do lahve – pro ověření, že analogová hodnota 350 je opravdu hodnota okolního vzduchu.

Senzor MQ-7 nakonec nebylo možno podobným způsobem zkalibrovat, protože běžně nelze získat čistý oxid uhelnatý. Pro určování koncentrace byly použity hodnoty z data-sheetu, které by měly odpovídat realitě. Na rozdíl od MQ-2 totiž modul se senzorem nemá na desce žádný potenciometr, který by mohl měnit citlivost a rozsah měření.

4.2 Program měřicího zařízení

Jako vývojové prostředí pro psaní programu měřicího zařízení bylo zvoleno Arduino IDE s knihovnou pro obsluhu mikropočítače STM Nucleo.



Obrázek 4.3: Kalibrace senzoru MQ-2.

Obsluha senzorů

Měřicí zařízení obsahuje tři moduly se senzory, které dohromady měří čtyři veličiny – teplotu a relativní vlhkost vzduchu senzorem RHT03, koncentraci hořlavých plynů v okolním vzduchu senzorem MQ-2 a koncentraci oxidu uhelnatého v okolním vzduchu senzorem MQ-7.

Senzory MQ-2 a MQ-7 jsou analogové, data z nich jsou tedy čtena funkcí `analogRead()`, jejíž jediný parametr je číslo analogového pinu, ze kterého jsou požadována data. Tato funkce vrací hodnotu typu `int` v rozsahu 0 až 1023. Navracená hodnota je přímo úměrná elektrickému napětí (v rozsahu 0 V až 5 V) na analogovém výstupu čteného senzoru. Takto vyčtená data je potřeba vynásobit kalibrační konstantou. Tak je získána hodnota koncentrace měřeného plynu v okolním vzduchu.

Senzor teploty a relativní vlhkosti vzduchu RHT03 je na rozdíl od předchozích dvou digitální. Jeho obsluhu zajišťuje knihovna `DHT.h`. Nejprve je potřeba definovat datový pin senzoru RHT03 a zvolit typ senzoru, přičemž knihovna podporuje dva, a to DHT11 a DHT22. RHT03 je pouze jiné označení senzoru DHT22, pro správnou funkci knihovny je tedy třeba zvolit tento typ. Po tomto nastavení lze snadno získat hodnotu teploty voláním funkce `readTemperature()`. Podobně lze získat také hodnotu relativní vlhkosti vzduchu, a to pomocí funkce `readHumidity()`. Obě zmíněné funkce navrací hodnotu typu `float`.

Práce s nevolatilní pamětí

Ačkoliv je použitý mikrokontrolér vybaven nevolatilní pamětí Flash, pro práci s ní je používána knihovna `EEPROM.h`, která je komunitou programátorů¹ upravena tak, aby emulo-

¹Vývojářů, kteří si zvolili vytvářet projekty pro mikrokontroléry STM ve vývojovém prostředí Arduino IDE, není málo. Tato komunita sdílí své poznatky na diskuzním fóru *Arduino for STM32* ([http:](http://)

vala paměť EEPROM na mikrokontrolérové platformě STM Nucleo. Tato knihovna umožňuje zápis a čtení dat pouze po jednotlivých Bytech (například zapsání nebo přečtení jednoho ASCII znaku). Funkce pro práci s pamětí pak vždy vyžadují číslo čteného nebo zapisovaného Bytu předané parametrem, toto číslo je tedy jakýsi ukazatel do paměti. Ve zdrojovém kódu programu měřicího zařízení tuto funkci plní globální proměnná `eptr`.

Naměřená data jsou zapisována do Flash paměti jako JSON pole objektů, kde každý objekt reprezentuje okamžité hodnoty v čase, kdy byl do paměti zapsán. Je potřeba dodržet validitu tohoto pole, protože mobilní aplikace už data nijak neupravuje a při nedodržení validity odeslaná data odmítne.

Nevolatilní paměť je použita pro případ výpadku nebo odpojení napájecího napětí, je tedy žádoucí, aby po opětovném připojení nově zapisovaná data navázala na data již uložená v paměti a nepřepisovala je. Před spuštěním hlavní smyčky programu je tedy potřeba najít konec zapsaných dat a nastavit na něj ukazatel `eptr`. Pro jednoznačnost je vždy poslední znak zapsaných dat mřížka (`#`). Je-li paměť prázdná, je potřeba do ní zapsat počáteční znak JSON pole (`[`).

Pro usnadnění zápisu dat do paměti byla implementována funkce `appendString()`, která na konec dat v paměti přidá obsah pole znaků předaného parametrem. Dojde-li během zápisu do nevolatilní paměti k odpojení napájení, s vysokou pravděpodobností nebude dodržena JSON validita, čímž budou naměřená data znehodnocena. Aby bylo riziko odpojení napájení v nesprávnou chvíli co možná nejmenší, je práce s pamětí signalizována LED diodou. Pokud tato dioda svítí, neměl by uživatel odpojovat napájení, jinak může přijít o všechna dosud nevyčtená data.

Pro pohodlnější čtení dat byla implementována funkce `readString()`, která přečte data z paměti a uloží je do pole znaků předaného parametrem. Flash paměť je však několikanásobně větší, než je paměť programu. Může tedy nastat situace, kdy data nelze vyčíst najednou, ale je nutné je číst po částech. Proto vyžaduje funkce ještě jeden parametr, který udává, kolik úseků již bylo z paměti vyčteno. Díky tomuto parametru je správně nastaven ukazatel do paměti. Funkce `readString()` vrací číslo 1, pokud ještě v paměti nějaká data zbývají, jakmile je přečtena celá paměť, funkce vrátí hodnotu 0.

Bluetooth komunikace ze strany měřicího zařízení

Díky připojení Bluetooth modulu k pinům TX a RX mikrokontroléru je možné jednoduše číst data pomocí funkce `Serial.read()` a zapisovat data užitím funkce `Serial.print()`.

Pro přijetí požadavku na poslání naměřených dat bylo původně zamýšleno použít přerušení. Nakonec jsem však tuto možnost přehodnotil, protože přerušení by mohlo přijít v době zápisu dat do paměti a tím jej narušit. Pak by nebyla dodržena validita JSON dat a mobilní aplikace by data nepřijala.

O odesílání dat se stará funkce `blSend()`, která v cyklu volá funkci `readString()`. Po vyčtení celé paměti je nastaven ukazatel `eptr` na druhý Byte paměti (protože v prvním Bytu už je zapsán znak začátku JSON pole) a na tento Byte je zapsán ukončující znak `#`.

[//www.stm32duino.com/](http://www.stm32duino.com/)). Na platformě GitHub existuje repozitář *Arduino_STM32* s knihovnami upravenými pro mikrokontroléry STM, mezi kterými je i knihovna EEPROM.h.

4.3 Mobilní aplikace

Pro vývoj mobilní aplikace jsem zvolil open-source framework NativeScript. Zadání mé práce tvorbu aplikace specifikuje pro platformu Android, NativeScript však nabízí možnost aplikaci zkompilovat i pro iOS. Pro definování uživatelského rozhraní jsem zvolil značkovací jazyk XML s využitím kaskádových Sass² stylů. Logika aplikace byla naprogramována v jazyce TypeScript z důvodu možnosti využití výhody statické typové kontroly. TypeScript rovněž nabízí výhody třídního objektově orientovaného programování.

Asynchronní programování aplikací pro Android

Každé aplikaci je přiděleno tzv. hlavní vlákno UI³ sloužící k interakci s uživatelem. Všechny komponenty aplikace jsou vytvořeny ve vlákně UI a interakce s nimi musí probíhat právě z tohoto vlákna.

Všechny dlouhotrvající operace je potřeba přesunout mimo vlákno UI do samostatných vláken. Určujícím faktorem, co je dlouhotrvající operace, je předpokládaný čas jejího trvání 50 až 100 milisekund i na nejpomalejším zařízení.

V mnoha případech je úlohou aplikační logiky v aktivitě aplikace načítat údaje z webové služby nebo ze souboru. Tento proces může trvat několik sekund. Kdyby byly údaje načítány v hlavním vlákně kódu aplikace, ta by po dobu čekání na údaje neodpovídala na podněty uživatele. Pokud by tento proces trval déle, operační systém Android by aplikaci po pěti sekundách násilně ukončil s chybovým hlášením *Application Not Responding*.

Proto je potřeba operace, u kterých se předpokládá, že budou trvat déle, realizovat jako asynchronní, takže nebudou blokovat aplikaci, která je spustila, ale poběží paralelně s ní. [21]

Struktura mobilní aplikace

Framework NativeScript používá moduly, které lze ve svých aplikacích využívat a značně tím zvýšit efektivitu práce. Základem implementované aplikace je modul `nativescript-theme-core` společně s modulem `nativescript-ui-sidedrawer` pro zobrazení postranního menu. Využitím těchto dvou modulů vznikla kostra aplikace, do které byly postupně přidávány obrazovky. Struktura mobilní aplikace vychází z architektonického vzoru MVC.

Obrazovka aplikace je celkem pět, z toho v menu jsou pouze dvě, a to domovská stránka a stránka s informacemi o aplikaci. Každá stránka je umístěna ve vlastní složce a obsahuje jeden XML dokument s definicí uživatelského rozhraní – v architektuře MVC známý jako *View*, jeden tzv. *code behind*, soubor stejného názvu jako XML dokument, avšak s koncovkou `nativescript-bluetooth.ts`. Obsahuje tedy program v TypeScriptu, který zajistí propojení (tzv. *data binding*) *View* s kontrolérem (v terminologii NativeScriptu bývá používán spíše pojem *ViewModel*). Dalším souborem ve složce stránky je její *ViewModel*, tedy zdrojový kód logiky stránky. V poslední řadě může obsahovat každá složka tři soubory – Sass styly upravující vzhled dané stránky, jeden pro zařízení s Androidem, jeden pro zařízení s iOS a jeden společný. Tato možnost je však využita jen u stránky *detail*, a i tam je využit pouze společný Sass styl pro danou stránku. Každá stránka kromě svého obsahu

²Sass – Syntactically Awesome StyleSheets – je nástavba klasických CSS stylů. Přináší řadu výhod, jako je například možnost používat proměnné, vnořená pravidla a v neposlední řadě napomáhá přehlednosti kaskádových stylů. Při kompilaci aplikace jsou Sass styly přeloženy do klasických CSS stylů. [8]

³UI – User Interface znamená uživatelské rozhraní.

zahrnuje také `ActionBar` pro zobrazení nadpisu a odkazu pro zobrazení menu a komponentu `RadSideDrawer` pro zobrazení postranního menu.

Model aplikace (v souvislosti s MVC architekturou) je tvořen JSON soubory uloženými v paměti mobilního zařízení v adresáři `measurements`. Práce se soubory je prováděna za pomoci modulu `file-system` ze souboru modulů `tns-core-modules`. Operační systém Android ukládá soubory aplikací do jim vyhrazených adresářů v paměti zařízení. Modul `nativescript` pro práci se souborovým systémem pak pracuje ve složce vyhrazené aplikaci. Výhodou je, že pokud je aplikace odinstalována, stačí smazat pouze její adresář a všechna data aplikací uložená budou rovněž odstraněna. Každý soubor představuje jedno vyčtené měření z měřicího zařízení a obsahuje jméno měření, poznámku k měření, datum a čas vyčtení a samotné naměřené hodnoty uložené jako pole objektů – každý objekt jsou hodnoty senzorů v určitém okamžiku.

Domovská stránka

První stránka, kterou *router* zobrazí po spuštění aplikace, je domovská stránka, která obsahuje seznam názvů již vyčtených měření (v XML element `<ListView>`). Tento seznam je generovaný z objektu `measurementList` typu `ObservableArray`. Obsah objektu `measurementList` je načten v konstruktoru třídy `HomeViewModel`, který do zmíněného objektu postupně načte obsah JSON souborů ze složky `measurements` v paměti mobilního zařízení. Načítání dat je prováděno asynchronně pomocí objektu `promise`, jak je popsáno v předchozí kapitole.

Domovská obrazovka také obsahuje tlačítko pro vyčtení nového měření, po jeho stisknutí bude uživatel přesměrován na jinou obrazovku.

Detail měření

Stránka detail měření slouží k reprezentaci naměřených dat. Ta jsou zapsána jak v tabulce, tak ve třech grafech (jeden zobrazuje vývoj teploty v čase, druhý vývoj vlhkosti vzduchu a třetí koncentrace plynů v okolním vzduchu). Grafy jsou vykreslovány za pomoci modulu `nativescript-ui-chart`. Při používání tohoto modulu je třeba stanovit typ grafu (výšečový nebo v kartézské soustavě souřadnic – těch modul nabízí celou řadu, včetně například sloupcových nebo spojnicových) a pomocí `data bindingu` navázat na data, která chceme zobrazovat. Pro potřeby aplikace je nejvhodnější graf spojnicový, na vodorovné ose zobrazující index jednotlivých měření v čase, na svislé pak hodnotu měřených veličin.

Na stránku detailu měření se uživatel dostane kliknutím na libovolnou položku seznamu na domovské obrazovce. Díky `data bindingu` je možné získat index prvku, na který uživatel kliknul. K přesměrování na obrazovku s dalšími informacemi o měření lze použít metodu `navigate()` obsaženou v modulu `frame`, která nabízí možnost přesměrovat i s tzv. kontextem, v tomto případě s jedním objektem z pole `measurementList`. `ViewModel` obrazovky s detaily měření obsahuje kromě konstruktoru (ve kterém se vyčte kontext přesměrování a nastaví `data binding`) ještě další dvě metody, a to `delete()` a `edit()`.

První metoda je volána, když uživatel stiskne ikonu popelnice v pravém horním rohu stránky. Odstranění měření je nevratná událost, proto je po uživateli požadováno potvrzení. Po jeho kladném vyhodnocení je z paměti mobilního zařízení smazán JSON soubor obsahující příslušné měření. Nakonec je uživatel přesměrován na domovskou obrazovku, kde je znovu provedeno načtení všech měření z paměti, která už však smazané měření neobsahuje.



Obrázek 4.4: Ukázka stránky Detail měření.

Metoda `edit()` pouze přesměruje uživatele na obrazovku s editací měření, je volána po stisku ikony tužky v horní části obrazovky. Přesměrování je provedeno podobně jako z domovské stránky na stránku s detaily měření, tedy s kontextem.

Editace měření

Na této stránce má uživatel možnost upravit informace o měření, a to jeho název a poznámku. Stránka sestává ze dvou textových polí, ve kterých jsou díky dvoucestnému data bindingu předepsané stávající hodnoty názvu a poznámky, a tlačítka pro uložení úprav. Po stisku tohoto tlačítka je přepsán JSON soubor obsahující upravované měření a uživatel je přesměrován s kontextem upraveného měření na jeho detail.

Načtení nového měření

Svým vzhledem se tato stránka podobá editaci měření – XML soubor je skoro stejný, v pozadí je však mnohem složitější. Kromě načtení názvu a poznámky je potřeba pomocí standardu Bluetooth vyčíst zaznamenané hodnoty z měřicího zařízení.

Pro práci s Bluetooth nabízí NativeScript modul `nativescript-bluetooth`, který má ovšem podporu pouze pro zařízení se standardem Bluetooth Low Energy a dosud není plánována podpora starších verzí technologie Bluetooth. Protože zatím neexistuje žádný podobný NativeScript modul pro klasický Bluetooth, bylo zde potřeba se odchýlit od původního navrhovaného Bluetooth modulu HC-05 (který využívá Bluetooth verze 2.0), místo něj byl použit novější modul HM-10.

Obsluha Bluetooth zařízení je značně časově náročná, například čtení dat může trvat až několik minut. Proto je potřeba provádět veškerou práci s Bluetooth asynchronně, a to v NativeScriptu pomocí již zmiňovaných objektů `promise`. Nejprve je potřeba se ujistit, že

dal uživatel svolení přistupovat k využívání Bluetooth v mobilním zařízení. Toto povolení se žádá buď při instalaci aplikace nebo při prvním pokusu o použití Bluetooth, a to metodou `requestCoarseLocationPermission()`. Pokud uživatel odsouhlasí povolení používat Bluetooth, je tato volba zapamatována a při dalším použití už aplikace o povolení nežádá. Povolení může uživatel kdykoliv upravit v nastavení mobilního zařízení. Poté je ještě potřeba zjistit, zda je Bluetooth zapnutý, a to metodou `isBluetoothEnabled()`. Na zařízeních s operačním systémem Android je možné Bluetooth povolit metodou `enable()`, která po souhlasu uživatele Bluetooth povolí.

Pak už je možné vyhledat dostupná Bluetooth LE zařízení metodou `startScanning()`. Tato metoda nabízí omezit hledání na zařízení s určitou Bluetooth službou. Služba, kterou využívá Bluetooth modul v měřicím zařízení, má UUID `ffe0`, hledání dostupných zařízení je tedy omezeno pouze na zařízení poskytující tuto službu. Vyhledaná zařízení jsou uložena do objektu `devices` typu `ObservableArray`. Tento objekt je čtený v XML souboru a každé nově nalezené zařízení je zobrazeno uživateli, který si z nalezených zařízení vybere to, se kterým se chce spárovat.

Kliknutím na některé zobrazené zařízení je spuštěna funkce `connectDevice()`, která se pokusí spojit s vybraným Bluetooth zařízením pomocí metody `connect()`. Je-li pokus o spárování úspěšný, Bluetooth zařízení odešle mobilní aplikaci JSON soubor s výčtem podporovaných služeb a charakteristik.

Poslední věc ze strany uživatele je vyplnění názvu měření, poznámky a potvrzení kliknutím na tlačítko „Vyčíst“. Tím je spuštěna funkce `add()`, která si nejprve ověří, že je připojeno nějaké Bluetooth zařízení. Poté zpracuje zadané údaje (název a poznámka) a funkcí `writeWithoutResponse()` pošle měřicímu zařízení požadavek, aby odeslalo naměřená data. Bylo třeba využít funkci `writeWithoutResponse()`, protože ta, na rozdíl od funkce `write()`, nečeká na odpověď a uvolní službu pro čtení.

Po odeslání požadavku aplikace pomocí metody `startNotifying()` čeká na příchozí data. Ta jsou datového typu `Uint8Array` a pomocí modulu `TextDecoder` jsou převedena na datový typ `string`. Vyčítaná data mohou být dlouhá natolik, že jsou odeslána v několika dávkách, proto se na příchozí data čeká až do doby, než přijde dávka s posledním znakem `']'`, který značí konec JSON pole s naměřenými hodnotami. Jakmile přijde takto označená poslední dávka dat, je vytvořen a do paměti zařízení uložen JSON soubor obsahující informace o měření. Pokud ukládání souboru proběhlo úspěšně, je spuštěna funkce `disconnect()`, která korektně odpojí Bluetooth zařízení a přesměruje uživatele na hlavní stránku aplikace.

4.4 Možnosti dalšího rozšíření zařízení

Měřicí zařízení je pro účely této bakalářské práce navrženo, sestaveno a naprogramováno jako zkušební přenosné zařízení s možností měření kvality ovzduší na vybraných místech. V možnostech dalšího rozšíření zařízení je však i úprava programu pro využití širší veřejností. A to zejména zapisování naměřených dat do cyklické fronty⁴ a změna vyčítací funkce tak, aby nemazala paměť. Potom by mohly být naměřené hodnoty vyčteny vícekrát, a to i různými uživateli. Měřicí zařízení by pak bylo umístěno na veřejně přístupné místo a naměřená data by si z něj mohl stáhnout kdokoli, kdo by si nainstaloval aplikaci pro vyčítání dat. Podobných měřicích zařízení by pak mohlo být umístěno více – na různých místech.

⁴Cyklická fronta po zaplnění paměti přepisuje novými daty ta nejstarší, která v paměti jsou.

Další možností rozšíření je obohatit měřicí zařízení o další senzory, například senzor MiCS-2710 pro měření koncentrace oxidu dusičitého (NO_2) a jiných znečišťujících látek v okolním vzduchu. Rovněž by bylo vhodné přidat do zařízení RTC⁵ modul, aby bylo měření přesně časově zařazeno. Na druhou stranu by se tím výrazně zvětšil JSON zápis jednoho měření a tím pádem by se paměť zařízení rychleji zaplnila.

Pro prodloužení doby měření je možné měřicí zařízení opatřit externí pamětí, například modulem s Micro SD kartou.

Pro dosažení lepší přenosnosti měřicího zařízení by bylo dobré napájet jej z baterie. To s sebou nese nutnost optimalizovat spotřebu měřicího zařízení (uspávat mikrokontrolér, vyčítání pomocí Bluetooth inicializovat pomocí přerušení). Toto také bylo původně zamýšleno, ale senzory MQ-2 a MQ-7 mají relativně velkou spotřebu energie (každý bezmála jeden Watt) a navíc kvůli nutnosti předejít přehřívání není možné je během čekání mezi intervaly měření odpojovat od napájení. Zařízení by tak vydrželo provoz na baterii jen krátce. Vhodným kompromisem by bylo opatřit zařízení baterií pro případ výpadku napájecího napětí. Další možností rozšíření by také bylo nabíjet baterii solárním panelem, což by zajistilo samostatnost měřicího zařízení bez nutnosti stálého napájení z rozvodné sítě.

Užitečným rozšířením by také byla možnost měnit intervaly mezi zapisovanými měřeními bez zásahu do programu mikropočítače (v současné době je totiž interval definovaný na jednom místě v programu, jeho úprava není složitá, ale vyžaduje připojení mikrokontroléru přes USB k počítači a nahrání upraveného programu).

Díky NativeScriptu je aplikace pro vyčítání naměřených hodnot snadno upravitelná i pro operační systém iOS.

⁵RTC modul– Real Time Clock modul je označení modulu s hodinami reálného času.

Kapitola 5

Experimenty

K otestování správné funkce měřicího zařízení byla provedena dvě dvacetičtyřhodinová měření na místech s předpokládanou rozdílnou hodnotou měřených veličin v ovzduší, a to ve městě a na venkově. Hodnoty byly zaznamenány každou půl hodinu a vývoj shrnují níže uvedené grafy.

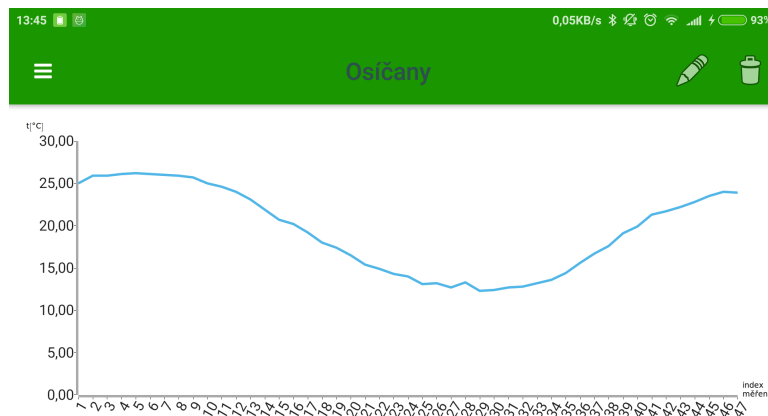


Obrázek 5.1: Experimentální měření během silného deště.

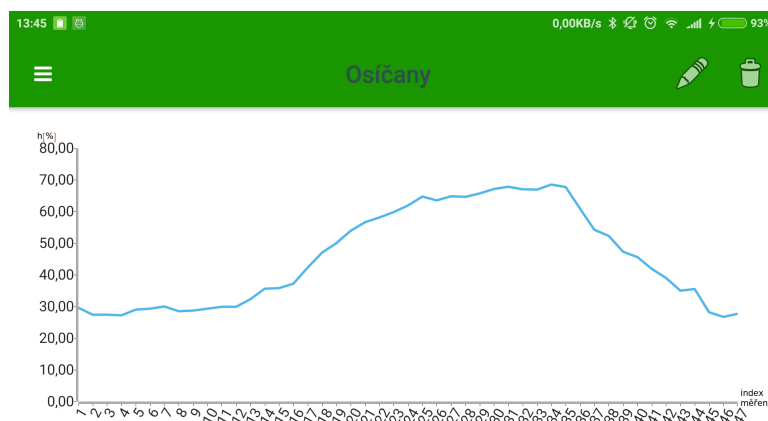
5.1 Experimentální měření – venkov

Hodnoty byly naměřeny ve vesnici Olomouckého kraje na adrese Osícany č.p. 48. Měření bylo provedeno v době od 13. května 2018 13:45 do 14. května 2018 13:45 hodin. Po celou dobu měření byla jasná obloha, měřicí zařízení bylo umístěno ve stínu. Kontrolní měření ukazuje zejména velký rozdíl teploty a relativní vlhkosti vzduchu v denní a noční době.

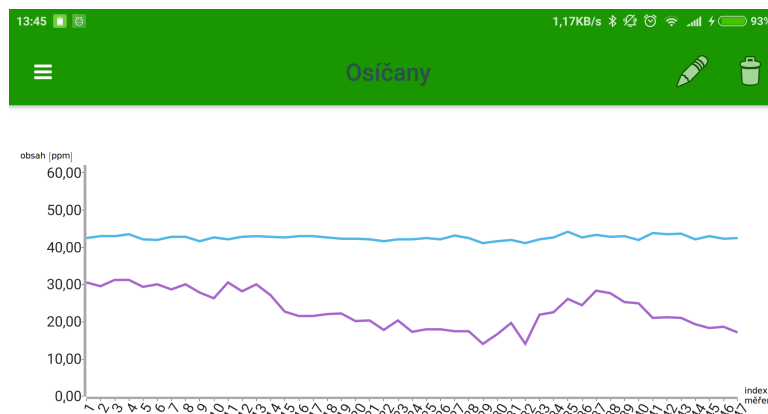
Teplota v noci postupně klesala z 25 °C na cca 14 °C (obrázek 5.2), naopak vlhkost vzduchu v noční době vzrůstala z 30 % na téměř 70 % (obrázek 5.3).



Obrázek 5.2: Vývoj hodnot teploty vzduchu na venkově.



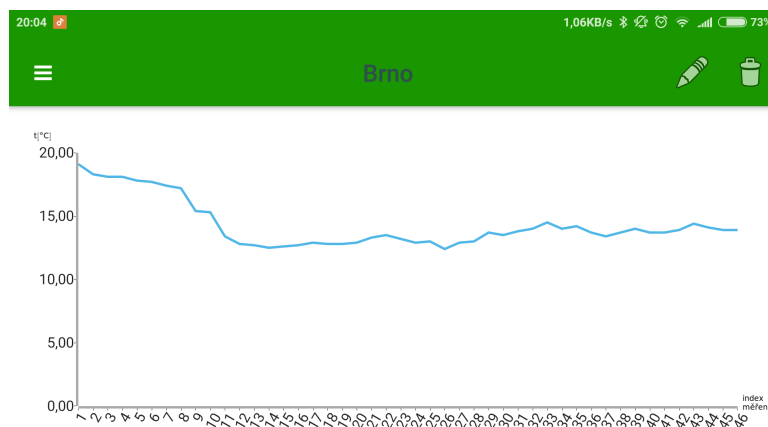
Obrázek 5.3: Vývoj hodnot relativní vlhkosti vzduchu na venkově.



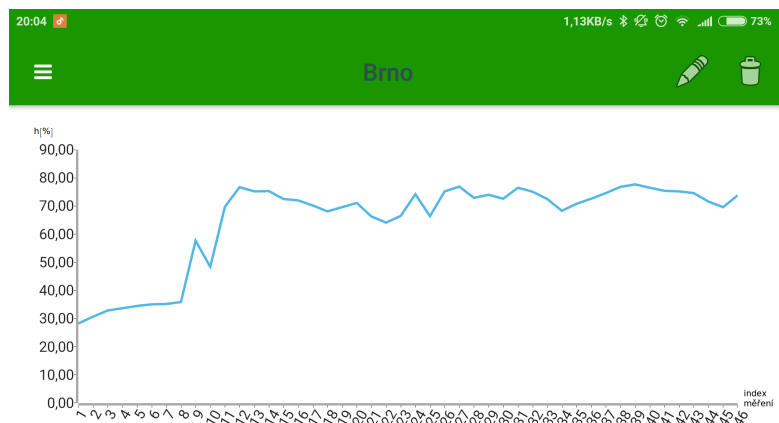
Obrázek 5.4: Vývoj hodnot koncentrace hořlavých plynů (tmavší křivka) a oxidu uhelnatého (světlejší křivka) ve vzduchu na venkově.

5.2 Experimentální měření – město

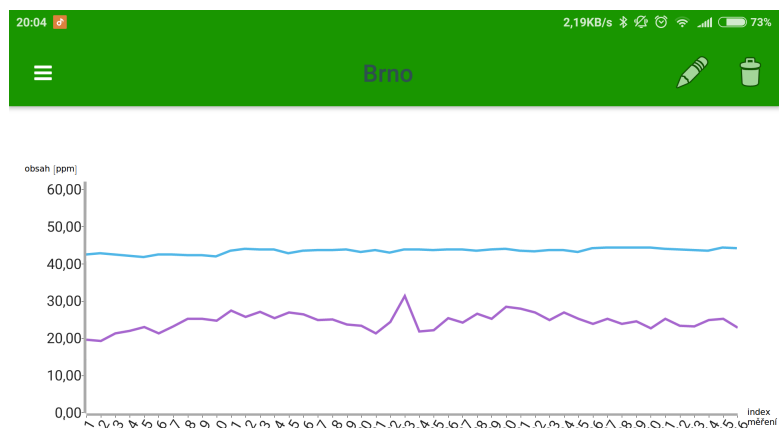
Hodnoty byly naměřeny ve městě Brně na adrese Merhautova 966/90. Měření bylo provedeno v době od 14. května 2018 20:15 do 15. května 2018 20:00 hodin. Po necelých čtyřech hodinách měření začalo silně pršet, což je doprovázeno poklesem teploty (obrázek 5.5) a značným vzrůstem relativní vlhkosti vzduchu (obrázek 5.6).



Obrázek 5.5: Vývoj hodnot teploty vzduchu ve městě.



Obrázek 5.6: Vývoj hodnot relativní vlhkosti vzduchu ve městě.



Obrázek 5.7: Vývoj hodnot koncentrace hořlavých plynů (tmavší křivka) a oxidu uhelnatého (světlejší křivka) ve vzduchu ve městě.

Oproti teoretickému předpokladu čistějšího prostředí na venkově než ve městě byly naměřeny nižší hodnoty hořlavých plynů a oxidu uhelnatého ve městě (obrázek 5.7) oproti venkovu (obrázek 5.4), což lze zdůvodnit pročištěním ovzduší deštěm v průběhu měření.

Kapitola 6

Závěr

V rámci této bakalářské práce bylo navrženo, sestrojeno a popsáno zařízení pro monitorování vybraných hodnot kvality vzduchu.

Zařízení je postaveno na mikrokontrolérové platformě STM Nucleo F446RE, k měření teploty, relativní vlhkosti vzduchu a škodlivých látek jsou využívána čidla RHT03, MQ-2 a MQ-7. Vytvořený software pak zpracovává data z těchto senzorů a ukládá je do nevolatilní paměti, odkud je uživatel může načíst do mobilní aplikace pro zařízení s operačním systémem Android pomocí technologie Bluetooth Low Energy.

Mobilní aplikace pro vyčítání a zobrazení naměřených hodnot byla vytvořena pomocí Frameworku NativeScript a je určena primárně pro operační systém Android. NativeScript však umožňuje aplikaci po drobných úpravách zkompileovat i pro operační systém iOS.

Sestrojené zařízení bylo úspěšně otestováno ve dvacetičtyřhodinovém zkušebním provozu na dvou místech s předpokládanou rozdílnou kvalitou ovzduší a výsledky měření jsou v práci prezentovány v grafech vygenerovaných mobilní aplikací. Výsledky měření ukazují (navzdory teoretickému předpokladu čistšího ovzduší na venkově než ve městě) nižší úroveň monitorovaných znečišťujících plynů ve městě oproti venkovu, a to zřejmě v důsledku pročištění ovzduší dlouhotrvajícím deštěm. To potvrzuje závislost kvality ovzduší na počasí.

V závěru implementační části jsou navržena další možná zdokonalení vytvořeného měřicího zařízení, aby tak našlo uplatnění pro využití širší veřejností.

Literatura

- [1] Alasdair, A.; Don, C.; Sandeep, M.: *Make: Bluetooth: Bluetooth LE Projects with Arduino, Raspberry Pi, and Smartphones*. O'Reilly, 2015, ISBN 1457187094.
- [2] AllAndroi: *Android architecture*. [Online; navštíveno 17.3.2017].
URL <https://allandroi.wordpress.com/2015/01/16/android-architecture/>
- [3] Android: *Distribution dashboard*. [Online; navštíveno 17.3.2017].
URL <https://developer.android.com/about/dashboards/>
- [4] Aosong Electronics Co.: *Digital-output relative humidity & temperature sensor/module DHT22*. [Online; navštíveno 4.4.2018].
URL <https://www.spar/itkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [5] Arduino: *ARDUINO UNO REV3*. [Online; navštíveno 13.1.2018].
URL <https://store.arduino.cc/arduino-uno-rev3>
- [6] Blažek, Z.; aj.: *VLIV METEOROLOGICKÝCH PODMÍNEK NA KVALITU OVZDUŠÍ V PŘEŠHRANIČNÍ OBLASTI SLEZSKA A MORAVY*. [Online; navštíveno 8.12.2017].
URL http://portal.chmi.cz/files/portal/docs/poboc/OS/OCO/pdf_ooco/publikace.pdf
- [7] Branstein, M.; Branstein, N.: *The NativeScript Book*. [Online; navštíveno 21.3.2018].
URL <https://www.nativescript.org/get-the-nativescript-book>
- [8] Catlin, H.; Weizenbaum, N.; Eppstein, C.: *Sass (Syntactically Awesome StyleSheets)*. [Online; navštíveno 5.5.2018].
URL https://sass-lang.com/documentation/file.SASS_REFERENCE.html
- [9] CO2Meter: *Carbon Monoxide Electrochemical Sensor*. [Online; navštíveno 15.1.2018].
URL <http://www.co2meters.com/Documentation/Datasheets/DS-AP-0005-CO-Sensor.pdf>
- [10] DFRobot: *Gravity: Formaldehyde (HCHO) Sensor*. [Online; navštíveno 5.3.2018].
URL <https://www.dfrobot.com/product-1574.html>
- [11] Dočekal, M.: *Správa linuxového serveru: Praxe šifrování s dm-crypt/LUKS*. [Online; navštíveno 20.3.2018].
URL <https://www.linuxexpres.cz/praxe/sprava-linuxoveho-serveru-praxe-sifrovani-s-dm-crypt-luks>
- [12] EAA: *Znečištění ovzduší*. [Online; navštíveno 9.12.2017].
URL <https://www.eea.europa.eu/cs/themes/air/intro>

- [13] Fadi, K.; Yael, E.; Rakefet, S.-N.; aj.: *Node-to-node field calibration of wireless distributed air pollution sensor network*. *Environmental pollution*, ročník 233, 2017.
- [14] Ganguli, M.: *Getting started with Bluetooth*. Cincinnati, Ohio: Premier Press, 2002, ISBN 1-931841-83-7.
- [15] Gates, A.: *Teplotní senzory: Vyberte si správně aneb RTD vs. TC*. [Online; navštíveno 15.1.2018].
URL <http://www.controlengcesko.com/hlavni-menu/artykuly/artykul/article/teplotni-senzory-vyberte-si-spravne-aneb-rtd-vs-tc>
- [16] Hanwei Electronics Co.: *Technical data MQ-7 gas sensor*. [Online; navštíveno 4.4.2018].
URL <https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>
- [17] Jansen, R. H.; Vane, V.; de Wolff, I. G.: *TypeScript: Modern JavaScript Development*. Packt Publishing, 2016, ISBN 978-1-78728-759-4.
- [18] JNHuaMao Technology Company: *Bluetooth 4.0 BLE module Datasheet*. [Online; navštíveno 10.5.2018].
URL <https://bit.ly/2Koepl1>
- [19] Keystudio: *Ks0097 keystudio Bluetooth Transmission Module for Arduino with Bottom HC-05 Master and Slave*. [Online; navštíveno 5.3.2018].
URL http://wiki.keystudio.com/index.php/Ks0097_keyestudio_Bluetooth_Transmission_Module_for_Arduino_with_Bottom_HC-05_Master_and_Slave
- [20] Knoblochová, A.: *Rizikové složky vzduchu*. [Online; navštíveno 10.12.2017].
URL <https://www.airbi.cz/rizikove-slozky-vzduchu.php>
- [21] Lacko, L.: *Vývoj aplikací pro Android*. Brno: Computer Press, první vydání, 2015, ISBN 978-80-251-4347-6.
- [22] Ministerstvo životního prostředí: *Informace o zdravotních rizicích spojených s kvalitou ovzduší v roce 2016*. [Online; navštíveno 15.2.2018].
URL [https://www.mzp.cz/C1257458002F0DC7/cz/zdravotni_dusledky_znecisteni_ovzdusi/\\$FILE/000-Zdravotni_rizika_2016-20180105.pdf](https://www.mzp.cz/C1257458002F0DC7/cz/zdravotni_dusledky_znecisteni_ovzdusi/$FILE/000-Zdravotni_rizika_2016-20180105.pdf)
- [23] Ministerstvo životního prostředí: *Kvalita ovzduší*. [Online; navštíveno 8.12.2017].
URL http://www.env.cz/cz/kvalita_ovzdusi
- [24] Mozilla a samostatní přispěvatelé: *Promise*. [Online; navštíveno 7.5.2018].
URL https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise
- [25] MySensors community: *Air Quality Sensor*. [Online; navštíveno 15.1.2018].
URL <https://forum.mysensors.org/topic/147/air-quality-sensor/45>
- [26] Nedelkovski, D.: *DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino*. [Online; navštíveno 5.3.2018].
URL <https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/>

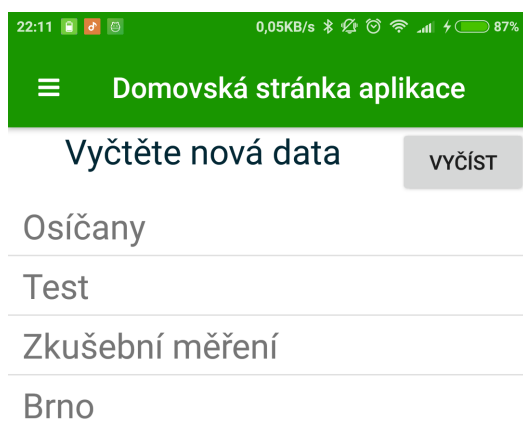
- [27] Open Handset Alliance: *Android 6.0 Changes*. [Online; navštíveno 17.3.2018].
URL <https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html>
- [28] Open Handset Alliance: *Android 7.0 Behavior Changes*. [Online; navštíveno 17.3.2018].
URL <https://developer.android.com/about/versions/nougat/android-7.0-changes.html>
- [29] Open Handset Alliance: *Android 8.0 Behavior Changes*. [Online; navštíveno 17.3.2018].
URL <https://developer.android.com/about/versions/oreo/android-8.0-changes.html>
- [30] Open Handset Alliance: *Android P Features and APIs*. [Online; navštíveno 17.3.2018].
URL <https://developer.android.com/preview/features.html>
- [31] Pecinovský, R.: *Návrhové vzory*. Brno: Computer Press, vyd. 1. vydání, 2007, ISBN 978-80-251-1582-4.
- [32] Pololu: *MQ-2 Semiconductor Sensor for Combustible Gas*. [Online; navštíveno 6.3.2018].
URL <https://www.pololu.com/file/0J309/MQ2.pdf>
- [33] Raspberry Pi Foundation: *RASPBERRY PI 3 MODEL B*. [Online; navštíveno 14.1.2018].
URL <https://www.raspberrypi.org/products/raspberry-pi-3-model-b>
- [34] Seeed community: *Grove - Dust Sensor*. [Online; navštíveno 5.3.2018].
URL http://wiki.seeed.cc/Grove-Dust_Sensor/
- [35] Seeed community: *Grove - Gas Sensor(MQ2)*. [Online; navštíveno 6.3.2018].
URL http://wiki.seeed.cc/Grove-Gas_Sensor-MQ2/
- [36] Seeed community: *Grove - Gas Sensor(MQ9)*. [Online; navštíveno 15.1.2018].
URL http://wiki.seeed.cc/Grove-Gas_Sensor-MQ9
- [37] ST: *NUCLEO-F446RE*. [Online; navštíveno 5.3.2018].
URL http://www.st.com/content/st_com/en/products/evaluation-tools/product-evaluation-tools/mcu-eval-tools/stm32-mcu-eval-tools/stm32-mcu-nucleo/nucleo-f446re.html
- [38] ST: *Overview of the ST-LINK embedded in STM32 MCU Nucleo, Discovery Kits and Eval Boards*. [Online; navštíveno 5.3.2018].
URL http://www.st.com/content/ccc/resource/technical/document/technical_note/group0/30/c8/1d/0f/15/62/46/ef/DM00290229/files/DM00290229.pdf/jcr:content/translations/en.DM00290229.pdf
- [39] ST: *STM32F401RE*. [Online; navštíveno 14.1.2018].
URL <http://www.st.com/en/microcontrollers/stm32f401re.html>
- [40] Telerik: *Observable*. [Online; navštíveno 26.3.2018].
URL <https://docs.nativescript.org/cookbook/data/observable>
- [41] The Engineering ToolBox: *Temperature Sensors - Comparing Types*. [Online; navštíveno 15.1.2018].
URL https://www.engineeringtoolbox.com/temperature-sensors-d_448.html

- [42] The Engineering ToolBox: *Thermocouples - types, principles and temperature ranges*. [Online; navštíveno 15.1.2018].
URL https://www.engineeringtoolbox.com/thermocouples-d_496.html
- [43] Vlčková, R.: *Imisní monitoring – anebo jak se měří imise?* [Online; navštíveno 6.1.2018].
URL http://www.ecmost.cz/img/clanky/imise/imisni_monitoring.pdf
- [44] ČHMÚ: *Instalace zábran a oprava střechy AIM Teplice*. [Online; navštíveno 10.12.2017].
URL <http://www.chmuul.org/aktuality/instalace-zabran-2009/#7>
- [45] Žilová; Stoklasová; Pavlíková: *Znečištění ovzduší*. [Online; navštíveno 9.12.2017].
URL http://ftp.zemedelka-opava.cz/web_skola/znecesteni_ovzdusi.pdf

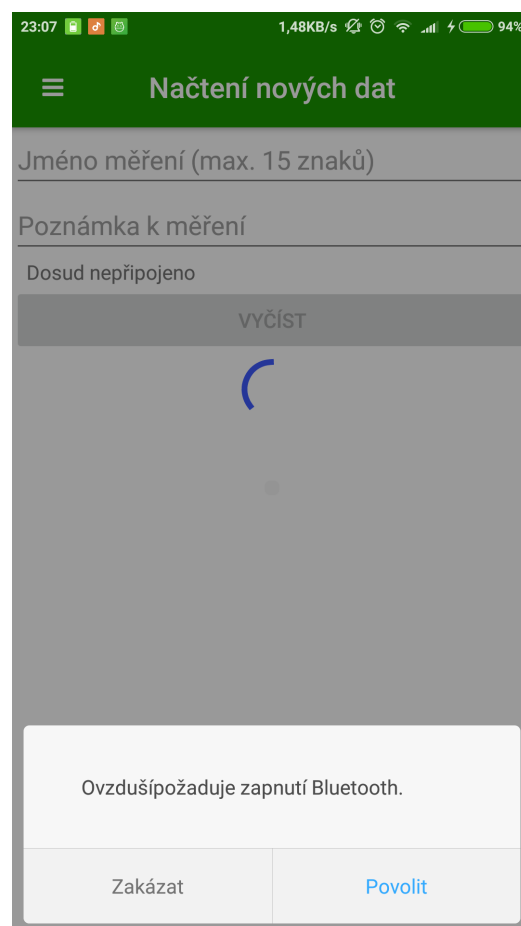
Příloha A

Snímky obrazovky mobilní aplikace

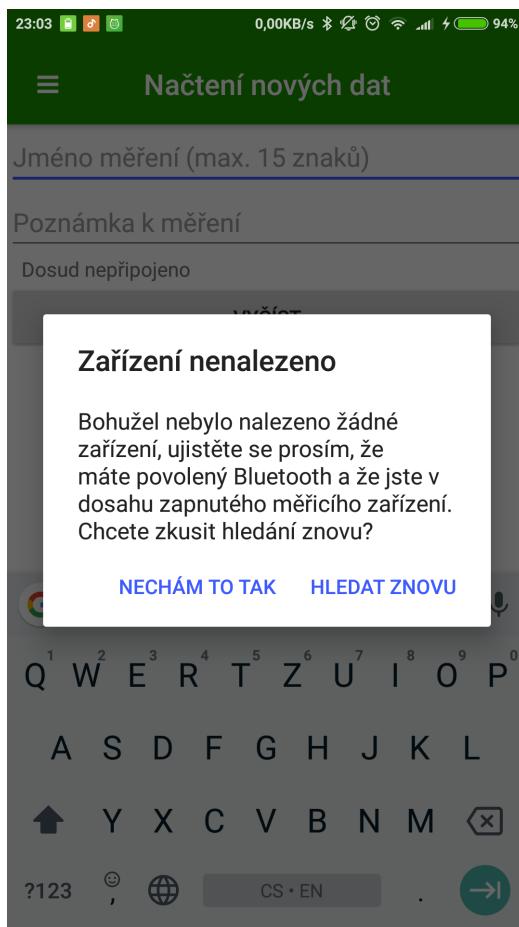
Domovská stránka aplikace



Pokus o vyčtení dat bez povoleného Bluetooth



Měřicí zařízení nebylo nalezeno



Úspěšně připojené měřicí zařízení

