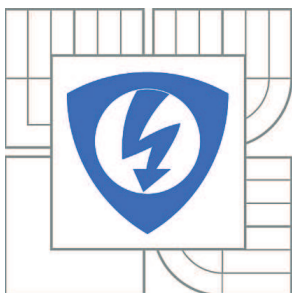


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

APLIKACE PRO SBĚR DAT ZE SÍTĚ BEZDRÁTOVÝCH SENZORŮ RELATIVNÍ VLHKOSTI VZDUCHU

DATA COLLECTING APPLICATION IN WIRELESS SENSOR NETWORK FOR MEASURING
RELATIVE HUMIDITY OF AIR

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

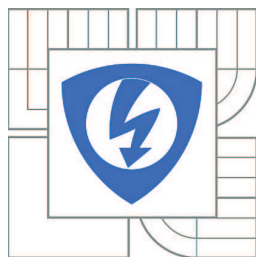
Bc. STANISLAV PROCHÁZKA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. DAVID JAROŠ

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav mikroelektroniky

Diplomová práce

magisterský navazující studijní obor
Mikroelektronika

Student: Bc. Stanislav Procházka

ID: 125295

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Aplikace pro sběr dat ze sítě bezdrátových senzorů relativní vlhkosti vzduchu

POKYNY PRO VYPRACOVÁNÍ:

Zadání diplomové práce se skládá ze dvou částí. První část tvoří návrh aplikace pro sběr dat ze sítě bezdrátových senzorů relativní vlhkosti vzduchu. Druhou částí je návrh a realizace převodníku pro vzájemný převod dat z bezdrátové sítě v pásmu ISM do bezdrátové sítě Wi-Fi.

Navrhněte a implementujte aplikaci s grafickým uživatelským rozhraním pro sběr dat ze sítě bezdrátových senzorů relativní vlhkosti vzduchu. Výsledná aplikace bude určena pro operační systém Android a bude založena na programové platformě Qt. Aplikace bude obsahovat základní správu sítě senzorů a možnost ukládání a grafické interpretace naměřených dat. Aplikace bude data přijímat z bezdrátové sítě Wi-Fi. Dále navrhněte obvodové řešení pro převodník dat mezi sítí bezdrátových senzorů komunikujících v pásmu ISM a řídicí aplikací, která komunikuje prostřednictvím bezdrátové komunikační technologie Wi-Fi. Navržený převodník realizujte. Zdrojové kódy programového vybavení přiložte jako součást práce na CD.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 10.2.2014

Termín odevzdání: 29.5.2014

Vedoucí práce: Ing. David Jaroš

Konzultanti diplomové práce:

prof. Ing. Vladislav Musil, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

ABSTRAKT

Diplomová práce se zabývá návrhem a realizací dvou částí systému pro měření relativní vlhkosti a teploty vzduchu. Systém měření teploty a vlhkosti je tvořen sítí bezdrátových senzorů, převodníkem ISM/Wifi a aplikací pro sběr, vyhodnocování a archivaci naměřených dat. Tato práce se zabývá návrhem a realizací převodníku ISM/Wifi a aplikace pro sběr, vyhodnocování a archivaci naměřených dat. Převodník ISM/Wifi bude schopen komunikovat se senzorovou sítí pomocí komunikačního protokolu SimpliciTI a koncovým zařízením (zařízení s operačním systémem Android) pomocí protokolu IEEE 802.11. Aplikace bude schopna obdržená data vyhodnocovat, ukládat a přehledně prezentovat koncovému uživateli.

KLÍČOVÁ SLOVA

Qt framework, síťová komunikace, Wifi, protokol SimpliciTI

ABSTRACT

The master work is focused on the design and implementation of two parts of the system for measuring temperature and relative humidity. The system for measuring temperature and relative humidity consists of a network of wireless sensors, a ISM/Wifi converter and application which will collect, analyze and archive data. The design and implementation of ISM/Wifi converter as well as the design and implementation of application are described in this work. ISM/Wifi converter will be able to communicate with the network of wireless sensors through ISM/Wifi protocol and end device (the device with Android operating system) by means of IEEE 802.11 protocol. The application will be able to evaluate, to save and to present user the received data very clearly.

KEYWORDS

Qt framework, network communication, Wifi, SimpliciTI protocol

BIBLIOGRAFICKÁ CITACE DÍLA:

PROCHÁZKA, S. *Aplikace pro sběr dat ze sítě bezdrátových senzorů relativní vlhkosti vzduchu*. Brno, 2013. 53 s. Vedoucí diplomové práce Ing. David Jaroš. FEKT VUT v Brně

PROHLÁŠENÍ AUTORA O PŮVODNOSTI DÍLA:

Prohlašuji, že jsem tuto vysokoškolskou kvalifikační práci vypracoval samostatně pod vedením vedoucího diplomové práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení.

V Brně dne 27.5.2014

.....

PODĚKOVÁNÍ:

Děkuji vedoucímu diplomové práce Ing. Davidu Jarošovi za metodické a cíleně orientované vedení při plnění úkolů realizovaných v návaznosti na diplomovou práci.

OBSAH

Obsah	iii
Seznam obrázků	vi
Seznam tabulek	viii
Úvod	1
1 Teoretická část	2
1.1 Volba programové platformy a programovacího jazyka	2
1.1.1 Programovací jazyk C++	2
1.1.2 Qt Framework	3
1.2 Deklarativní jazyk QML	4
1.3 Volba vývojového prostředí.....	5
1.3.1 Visual Studio.....	5
1.3.2 Qt Creator	5
1.3.3 Zhodnocení a výběr vývojového prostředí	5
1.4 Volba komunikačního protokolu	6
1.4.1 Protokol UDP.....	6
1.4.2 Protokol TCP	6
1.4.3 Zhodnocení a výběr protokolu.....	7
1.5 Norma IEEE 802.11.....	7
1.6 Komunikační protokol SimpliciTI.....	8
1.7 Pásmo pro průmyslové, vědecké a lékařské aplikace	9
2 Návrh aplikace	11
2.1 Systém měření teploty a vlhkosti.....	11
2.2 Stanovení požadavků na aplikaci.....	12
2.2.1 Komunikační požadavky	12
2.2.2 Požadavky na správu sensorové sítě.....	12
2.2.3 Požadavky na způsob ukládání naměřených dat	13
2.2.4 Požadavky na grafickou interpretaci dat.....	13
2.2.5 Požadavky na grafické uživatelské rozhraní.....	13

2.3	Návrh aplikace s využitím unifikovaného modelovacího jazyka	13
2.3.1	Diagram případů užití	13
2.3.2	Návrh diagramu tříd	16
3	Programová implementace aplikace	18
3.1	Diagram tříd	18
3.2	Aktivity diagramy hlavních metod	19
3.2.1	Přihlášení senzoru	19
3.2.2	Obdržení naměřených hodnot	21
3.2.3	Přidání senzoru na blokující seznam	21
3.2.4	Odstranění senzoru	22
3.3	Datový rámeček	23
3.4	Realizace grafického uživatelského prostředí	25
3.4.1	Rozvržení obrazovky aplikace	25
3.4.2	Tvorba ikon	26
3.5	Struktura aplikace	27
3.5.1	Hlavní obrazovka	28
3.5.2	Nastavení aplikace	29
3.5.3	Seznamy senzorů	30
3.5.4	Nastavení senzoru	31
3.5.5	Detail senzoru	32
3.5.6	Naměřené hodnoty	33
3.5.7	Statistika naměřených hodnot	34
3.5.8	Grafické zobrazení naměřených hodnot	34
4	Návrh obvodového řešení převodníku ISM/Wifi	38
4.1	Funkce převodníku	38
4.2	Volba vhodných komponent	39
4.2.1	Volba mikroprocesoru	39
4.2.2	Volba obvodu čtečky RFID	39
4.2.3	Volba Wifi modulu	40
5	Obvodové řešení převodníku ISM/Wifi	42
5.1	Obvodové řešení	42
5.2	Popis ovládacích a signalizačních prvků	43
5.3	Popis softwaru	44

5.3.1	Nastavení Wifi modulu.....	45
6	Závěr	46
	Literatura	47
	Seznam symbolů, veličin a zkratk	48
	seznam příloh	49
A	Obvodové řešení převodníku ISM/Wifi	50
B	Deska plošných spojů	51
C	Komunikační protokol	52

SEZNAM OBRÁZKŮ

Obrázek 1 – Mechanismus signálů a slotů v Qt – převzato z [5]	4
Obrázek 2 – Segmentace dat – převzato z [6]	6
Obrázek 3 – TCP segment – převzato z [6]	7
Obrázek 4 – Síťová topologie Simpliciti – převzato z [5].....	8
Obrázek 5 – Systém pro měření teplot.....	11
Obrázek 6 – Schématické značky diagramů případů užití.....	14
Obrázek 7 – Use case diagram aplikace	15
Obrázek 8 – Diagram tříd	16
Obrázek 9 – Diagram tříd realizované aplikace.....	18
Obrázek 10 – Diagram aktivit – přihlášení senzoru	20
Obrázek 11 – Aktivity diagram – Obdržení naměřených hodnot.....	21
Obrázek 12 – Aktivity diagram – přidání senzoru do blokujícího seznamu	22
Obrázek 13 – Aktivity diagram – odstranění senzoru	23
Obrázek 14 – Obecný tvar rámce	23
Obrázek 15 – Přizpůsobení aplikace rozměrům displeje.....	26
Obrázek 16 – Správný návrh grafických prvků - použití jednotky dp (převzato z [12])	27
Obrázek 17 – Špatný návrh grafických prvků - použití jednotky px (převzato z [12])..	27
Obrázek 18 – Hierarchický model rozvržení aplikace.....	27
Obrázek 19 – Úvodní obrazovka aplikace	28
Obrázek 20 – Obrazovka – nastavení aplikace	29
Obrázek 21 – Seznamy senzorů.....	30
Obrázek 22 – Obrazovka – nastavení senzoru	31
Obrázek 23 – Obrazovka – detail senzoru	32
Obrázek 24 – Obrazovka – přehled naměřených hodnot.....	33
Obrázek 25 – Obrazovka – statistika naměřených hodnot	34
Obrázek 26 – Obrazovka – grafické zobrazení hodnot.....	35
Obrázek 27 – Dialogová okna – přidání/odebrání hodnot	36
Obrázek 28 – Dialogová okna – nastavení os	37
Obrázek 29 – Blokové schéma převodníku	38

Obrázek 30 – RFID transpondéry	40
Obrázek 31 – Obvodové řešení převodníku	42
Obrázek 32 – Převodník ISM/Wifi	43

SEZNAM TABULEK

Tabulka 1 – Označení a frekvenční rozsah pásem ISM (převzato z [16]).....	9
Tabulka 2 – Typy přenášených paketů	24
Tabulka 3 – Hodnota a význam bajtu CMD	25

ÚVOD

Diplomová práce se zabývá návrhem a realizací dvou částí systému pro měření relativní vlhkosti a teploty vzduchu. Systém měření relativní vlhkosti a teploty je tvořen sítí bezdrátových senzorů, převodníkem ISM/Wifi a aplikací pro sběr, vyhodnocování a archivaci naměřených dat. Tato práce se zabývá návrhem a realizací převodníku ISM/Wifi a aplikace pro sběr, vyhodnocování a archivaci dat.

Systém pro měření vlhkosti se skládá ze tří částí a to sensorové sítě, převodníku ISM/Wifi a samotné aplikace. Systém bude sloužit pro měření teploty a relativní vlhkosti vzduchu uvnitř budov. Primárně bude určen pro nově zateplené budovy, a to jak rekonstruované, tak novostavby. Výsledky prezentované aplikací by měly sloužit jako zpětná vazba pro stavaře a majitele domů, zdali zateplení domů vedlo k zlepšení klimatu ve sledovacích prostorách. Z naměřených hodnot by mělo být dále čitelné, zda jsou obytné prostory dostatečně a správně větrány. Nesprávné větrání může způsobovat zvyšování relativní vlhkosti vzduchu, to vede ke srážení vodních par a v místě srážení pak k plísním.

Převodník ISM/Wifi bude schopen komunikovat se sensorovou sítí pomocí komunikačního protokolu SimpliciTI a koncovým zařízením (zařízení s operačním systémem Android) pomocí protokolu IEEE 802.11. Funkcí převodníku je pouze data předávat, nijak s nimi nemanipulovat.

Aplikace bude shromažďovat data předložená z převodníku ISM/Wifi, ta bude dále zpracovávat a prezentovat. Dalším úkolem aplikace bude spravování sensorové sítě (přidávání/odebírání senzorů ze systému). Předností aplikace bude její „uživatelsky přívětivý“ design. Prezentace dat bude probíhat ve formě grafů a tabulek, z čehož vyplývá, že data budou pro koncového uživatele lehce srozumitelná a dobře čitelná.

Výsledná aplikace bude určená pro zařízení s operačním systémem Android, z toho plyne, že aplikace je primárně určená pro přenosná zařízení („chytré“ telefony, tablety, atd.). Přenos měřených dat bude probíhat pomocí bezdrátové komunikace v bezlicenčním frekvenčním pásmu ISM (Industrial Scientific and Medical).

Diplomová práce je členěna do několika částí – teoretické části, návrhu programu, programové implementace aplikace, obvodového návrhu převodníku, obvodového řešení převodníku a závěru. V teoretické části se budeme věnovat multiplatformní knihovně Qt, programovacímu jazyku C++, vhodné volbě vývojového prostředí a různým komunikačním protokolům. Část návrh programu se bude zabývat návrhem aplikace pomocí technik UML (Unified Modeling Language). Část obvodová implementace aplikace bude popisovat do jaké míry se realizovaná aplikace odklonila od jejího návrhu, dále zde bude aplikace popsána z pohledu uživatele (popis jednotlivých funkčních bloků). V části obvodového návrhu se budeme věnovat vhodné volbě obvodových prvků a jejich zapojení. Část obvodového řešení převodníku bude zaměřena na popis postupu realizace, realizovaného převodníku a jeho správného pracovního nastavení. V poslední části budou zhodnoceny výsledky, kterých bylo dosaženo.

1 TEORETICKÁ ČÁST

V teoretické části diplomové práce jsou zvoleny vhodné informační technologie, které budou sloužit k realizaci aplikace a převodníku ISM/Wifi. Tyto technologie budou zvoleny na základě stanovených požadavků, které musí budoucí aplikace splňovat. Samozřejmostí je respektování zadání práce. V kapitole se tedy budeme věnovat zvolení vhodného programovacího jazyka a vhodného vývojového prostředí. Teoretická část práce se zabývá pojmy, jako jsou programovací jazyk C++, objektová grafická knihovna Qt, komunikační protokoly, Visual Studio nebo Qt Creator.

1.1 Volba programové platformy a programovacího jazyka

Volbou programové platformy se nemusíme příliš zabývat, jelikož je daná zadáním – jedná se o objektovou grafickou knihovnu Qt. Qt je knihovnou programovacího jazyka C++, existuje však i pro další programovací jazyky jako například Python, C# nebo Java. Ovšem drtivá většina aplikací využívajících tuto grafickou knihovnu jsou psané pomocí programovacího jazyka C++. V našem případě byl také zvolen programovací jazyk C++ s ohledem na množství již existujících funkcí, výkon, kvalitní dokumentace a množství dostupných ukázkových kódů.

1.1.1 Programovací jazyk C++

C++ je objektově orientovaný programovací jazyk, který touto vlastností navazuje na programovací jazyk SIMULA, jež byl prvním objektově orientovaným jazykem vůbec. Programovací jazyk C++ vznikl začátkem 80. let v Belových laboratořích zásluhou Bjarne Stroustrupa a dalších. Jak sám název napovídá, jedná se o rozšíření jazyka C, a to o rysy OOP. Jelikož je jazyk C až na několik výjimek podmnožinou jazyka C++, je každý program napsaný v jazyce C platným programem C++. Zároveň programy psané v C++ mohou využívat knihovny psané v jazyce C. [1]

Výhody použití jazyka C++

- široce používaný jazyk,
- vychází z programovacího jazyka C,
- výhody objektového programování,
- rychlost (jedná se o kompilovaný jazyk),
- široká dostupnost kvalitních překladačů pro různé hardwarové platformy, vývojových a ladících nástrojů.

Nevýhody jazyka C++

- relativní složitost,
- některé knihovny nejsou standardizované,
- změna zdrojového kódu vyžaduje rekompilaci.

1.1.2 Qt Framework

Pro tvorbu grafického uživatelského prostředí v C++ je možné využít hned několik grafických knihoven. Tyto knihovny jsou využívány zejména k realizaci grafického uživatelského rozhraní. Mezi nejznámější patří MFC, což je knihovna od společnosti Microsoft, která slouží k tvorbě programů pro MS Windows. Další možností může být multiplatformní knihovna Qt dostupná pod licencí GPL (General Public Licence) a LGPL (Leader General Public Licence).

Pro naši aplikaci byla zvolena knihovna Qt s ohledem na některé její výhody. Vyniká širokou podporou operačních systémů, open-source licencemi a v neposlední řadě velice kvalitně zpracovanou dokumentací.

Na první verzi knihovny se začalo pracovat již v roce 1991, o tři roky později dva z vývojářů této knihovny založili společnost Trolltech. V roce 2008 byl Trolltech odkoupen společností Nokia a přejmenován na QtSoftware. Ten se staral o další rozvoj knihovny. Původní myšlenka „napiš jednou, sestav kdekoli“ se těší mezi programátory velké oblibě a patří k její bezesporu největší výhodě.

V dnešní době patří Qt k nejpoužívanějším grafickým knihovnám, které běží pod open-source licencí. Je to multiplatformní grafická knihovna, která podporuje jak desktopové (Windows, Linux, OS X), tak i mobilní (od verze 5 podpora systému Android) platformy. Aplikace vytvořené pro grafické uživatelské prostředí používají nativní vzhled operačního systému, takže vyvinuté aplikace se vždy přizpůsobí do používaného prostředí.

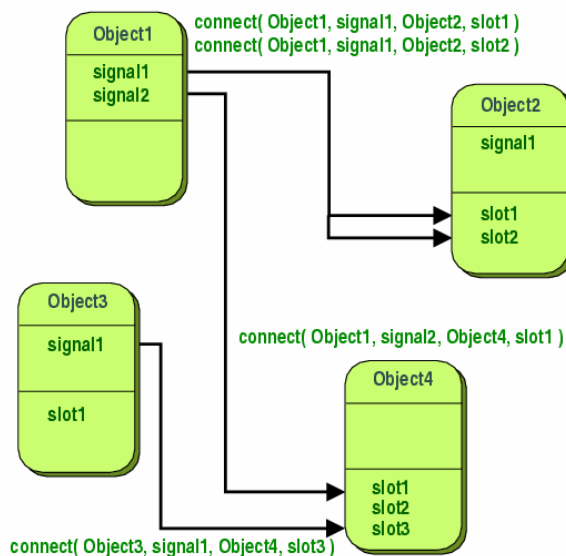
Knihovna Qt obsahuje třídy, které podporují práci s databázemi, QML (Qt Meta Language) soubory, síťovou komunikací a další. Hlavním rozdílem, kterým se Qt odlišuje od ostatních knihoven je využívání mechanismu signálů a slotů. Tento mechanismus slouží pro komunikaci mezi objekty. V programování GUI (Graphical User Interface) často potřebujeme, aby nějaký widget (widget je základním prvkem pro vytvoření uživatelského GUI, widgety lze zobrazovat data a přijímat informace od uživatele) reagoval na podmět jiného widgetu, např. pokud stiskneme tlačítko „Zavřít“, pravděpodobně budeme chtít vyvolat funkci *close()*. K tomuto spojení nám dopomáhá mechanismus signálů a slotů. [11]

Starší knihovny této komunikace dosahují tzv. zpětným voláním, což je ukazatel na metodu objektu, kterou chceme vyvolat po nějaké události jiného objektu. Tento přístup měl dvě nevýhody. Nebyla zde při volání typová kontrola a metody volané pomocí callbacku jsou silně vázané, tzn. volaná metoda musí znát ukazatel na metodu, z které byla vyvolána. Při používání signálů a slotů se tyto dvě nevýhody nevyskytují. [4]

Další výhodou mechanismu signálů a slotů je bezpečná komunikace mezi objekty, každý objekt se nachází ve vlastním vlákne programu. Tyto objekty mohou mít signály, které odkazují na sloty v jiných vláknech, a posílat události objektům, které v těchto vláknech „žijí“. Toto je možné, protože každé vlákno má dovoleno vlastnit svou událostní smyčku. [4]

Pro použití signálů a slotů se definuje spojení signálu se slotem pomocí metody *connect()*, což je přetížená metoda, která se tudíž může volat s různými argumenty. Sloty a signály mohou být využity ve všech objektech, které jsou přímo nebo nepřímo zděděny ze třídy *QObject*. [4]

Signály jsou vysílány objektem, pokud se jeho vnitřní stav změnil takovým způsobem, který je zajímavý pro jeho uživatele nebo vlastníka. Qt widgety mají spoustu předdefinovaných signálů, ale i tak má programátor možnost definovat si signály vlastní. Slot je zcela normální funkce, ke které je připojen určitý signál, který ho volá. Slot však může být volán i standardně bez použití signálu. [4]



Obrázek 1 – Mechanismus signálů a slotů v Qt – převzato z [5]

Na obrázku 1 je zobrazen mechanismus signálů a slotů. Můžeme zde vidět, že je možné přiřadit jeden signál více slotům. Co zde již není zobrazeno, je skutečnost, že k jednomu slotu může být připojeno více signálů, dokonce signál může být připojen k jinému signálu. [4]

1.2 Deklarativní jazyk QML

Qt Modeling Language (dále jen QML) je deklarativní jazyk, který umožňuje tvorbu grafického uživatelského prostředí pomocí jednotlivých grafických komponent. Pomocí QML je možné popsat vztahy mezi jednotlivými komponenty a jejich reakce na vnější i vnitřní podněty. Jedná se o vysoce čitelný jazyk, který podporuje dynamické změny vlastností jednotlivých komponent tak, aby byly použitelné ve více částech uživatelského rozhraní a dokázali se tomuto rozhraní přizpůsobit. Použitím modulu QtQuick může návrhář nebo vývojář snadno vytvořit animační uživatelské prostředí a má možnost připojit knihovny C++. QML, mimo jiné, také podporuje JavaScript.

Díky QML lze snadno oddělit programovou a grafickou část programu, a to se zachováním všech výhod Qt. U menších projektů lze s výhodou použít návrhový program Qt Quick Designer, který je součástí vývojového prostředí Qt Creator.

1.3 Volba vývojového prostředí

Jestliže uvažujeme vývoj aplikace pomocí grafické objektové knihovny Qt a programovacího jazyku C++, naskýtají se hned dvě možnosti kvalitních IDE (Integrated Development Environment). Tímto vývojovým prostředím může být produkt firmy Microsoft – Visual Studio nebo Qt Creator distribuovaný firmou Digia.

1.3.1 Visual Studio

Microsoft Visual Studio je vývojové prostředí, pomocí něhož je možné programovat jak konzolové aplikace, tak aplikace s grafickým výstupem Windows Forms, webové stránky, webové aplikace, atd. Visual Studio primárně nepodporuje žádný programovací jazyk, podpora jazyků je přidávána pomocí tzv. jazykové služby. Do Visual Studia lze implementovat knihovnu Qt. IDE obsahuje editor kódu, který podporuje zvýrazňování syntaxe, automatické doplňování kódů, atd. Dále pak výkonný debugger, který podporuje vkládání break pointů, watche (sledování určité hodnoty), krokování programu, atd. A v neposlední řadě designery jako například WinForm Designer nebo Web Designer. Tyto designery usnadňují práci programátorovi při grafickém návrhu aplikace. Podporují „drag and drop“ čili uživatel si v nabídce vybere nějaký prvek, např. tlačítko, a pouze ho přetáhne do svého návrhu.

1.3.2 Qt Creator

Jak už z názvu vyplývá, Qt Creator je primárně určen pro vývoj aplikací s využitím knihovny Qt a programovacího jazyka C++. Jedná se o multiplatformní IDE, kde je možné vyvíjet aplikace jak konzolové, tak i grafické. Obsahuje pokročilý editor C++ kódu s automatickým doplňováním kódu, nástroje pro správu a budování projektů, vizuální debugger. Další nesporně výbornou vlastností je integrovaná kontextová nápověda. Samozřejmostí je přítomnost Qt Designer, pomocí něhož lze jednoduše metodou „drag and drop“ vytvořit grafický návrh aplikace.

1.3.3 Zhodnocení a výběr vývojového prostředí

Microsoft Visual Studio je nesporně robustnější a v některých ohledech zajisté lepším vývojovým prostředím. Vyniká například výborným debuggerem, možnou implementací jazykových služeb a jednodušším přístupem ke konfiguraci a úpravám projektu, lepším laděním programu. Hlavní nevýhodou je orientace použití finálního programu pouze na operační systém Windows.

Qt Creator je oproti Visual Studiu méně robustní vývojové prostředí, avšak jeho nespornou výhodou je jeho multiplatformnost a orientace pouze na problematiku Qt. S ohledem na tuto skutečnost zde nalezneme lepší podporu Qt objektů (pomoc, navigace, atd.), lepší GUI editor. Mezi nevýhody Qt Creatoru patří špatné ladění programu.

Z porovnání obou IDE jasně vyplývá, že pro naši potřebu bude vhodnější využití Qt Creatoru. Hlavním důvodem je typ koncového zařízení, na které bude aplikace implementována.

1.4 Volba komunikačního protokolu

Tato část práce je věnována výběru vhodného komunikačního protokolu, pomocí kterého bude aplikace komunikovat s převodníkem ISM/Wifi. Jedná se o protokoly transportní vrstvy architektury TCP/IP (Transmission Control Protocol/Internet Protocol) a to protokol TCP (Transmission Control Protocol) a UDP (User Datagram Protocol).

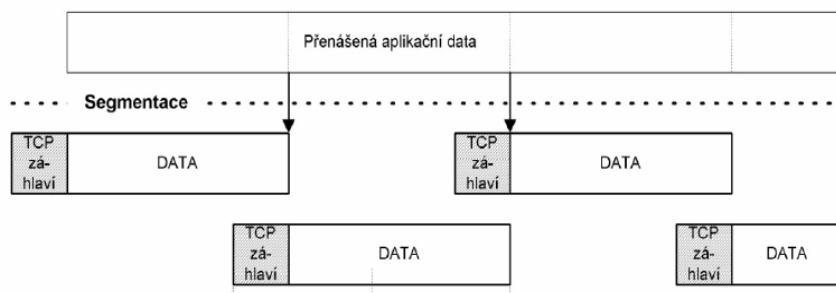
1.4.1 Protokol UDP

Protokol UDP realizuje nespojovou nespolehlivou službu. To znamená, že se před vlastní komunikací nerealizuje žádné ověření připojení, přistupuje se rovnou k odesílání dat. Tato služba odesílá data bez ohledu na to, jestli je někdo očekává. Odesílaná data jsou balena do tzv. UDP datagramů s velmi jednoduchou hlavičkou, která obsahuje pouze zdrojový a cílový port, délku dat a kontrolní součet. Kontrolní součet je však nepovinný údaj. UDP datagramy jsou dále baleny do datagramů IP. Hlavní výhodou této služby je jednoduchost a nízká přenosová režie. [6]

1.4.2 Protokol TCP

Protokol TCP je stejně jako protokol UDP službou transportní vrstvy architektury TCP/IP. Protokol TCP realizuje spojovou spolehlivou službu. To znamená, že před každým odesláním dat je zřízeno vlastní spojení obou komunikujících bodů. Příjemce dat musí potvrdit, že je připraven přijímat data. Po sestavení spojení je možné data odesílat, příjemce musí každý příjem dat potvrdit. Potvrzování dat může probíhat dvěma způsoby, buď potvrzením každého přijatého paketu zvlášť (protokol Stop-and-Wait) nebo potvrzením více paketů najednou, pak se jedná o metodu plovoucího okna (protokol Sliding Window). [6]

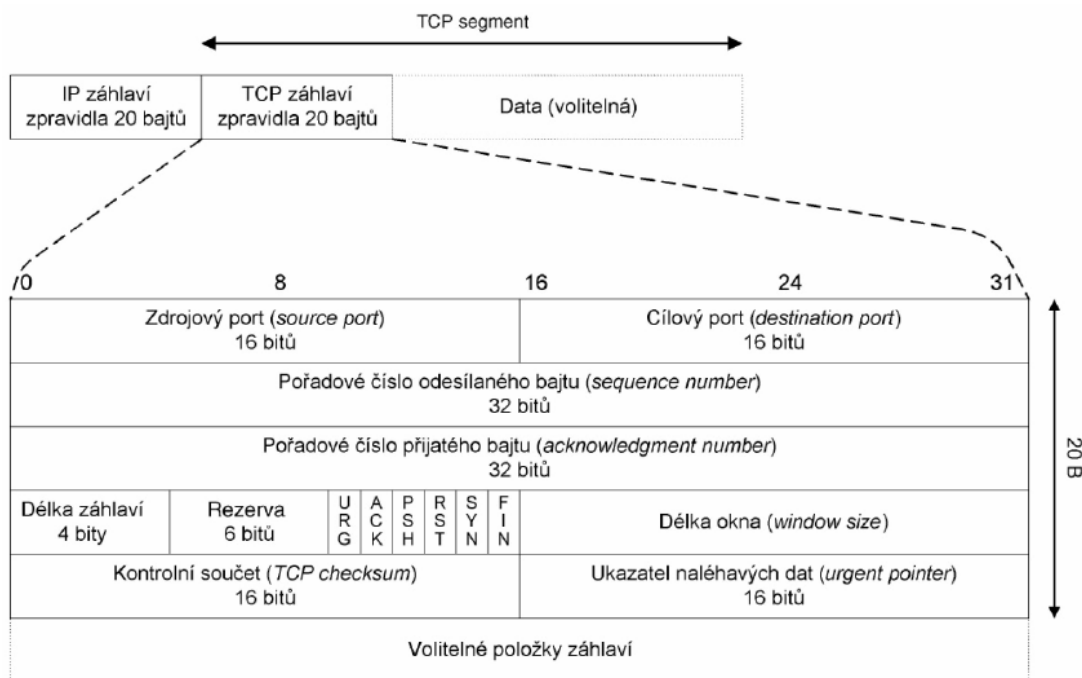
Jelikož transportní vrstva využívá pouze nespolehlivé a nespojové služby síťové vrstvy, probíhá realizace spolehlivé a spojové služby v koncových zařízeních, a to softwarově. [6]



Obrázek 2 – Segmentace dat – převzato z [6]

Pro účely přenosu dat jsou přenášená data dělena do tzv. segmentů TCP, které jsou baleny do IP datagramů, ty obsahují záhlaví a přenášená data. Datagram IP se vkládá do linkového rámce. Pokud by byla velikost datagramu IP větší než velikost linkového rámce, musela by probíhat fragmentace, což zvyšuje režii přenosu dat. Proto je snaha

dělit přenášená data do takových segmentů, aby fragmentace nebyla nutná. Na obrázku 2 je zobrazení dělení přenášených dat do TCP segmentů. [6]



Obrázek 3 – TCP segment – převzato z [6]

Záhlaví TCP segmentu (obrázek 3) obsahuje položky, které musí být povinně vyplněny, a mezi nimi i kontrolní součet. Kontrolní součet je důležitý pro bezpečný přenos dat, počítá se z velikosti TCP segmentu doplněného o pseudozáhlaví. Kontrolní součet vyžaduje sudý počet bajtů, proto v případě lichého počtu bajtů je výsledek doplněn o jeden bajt. Povinné položky záhlaví mají velikost 20 bajtů, za nimi mohou následovat nepovinné položky a to maximálně do velikosti 40 bajtů. [6]

1.4.3 Zhodnocení a výběr protokolu

Po zhodnocení všech výhod a nevýhod obou protokolů je jasné, že pro naši aplikaci je vhodnější použít protokol TCP. Jeho nevýhody, jako větší přenosová režie a relativní složitost, nepředčí jeho spolehlivost při přenášení dat. Aplikace si nemůže dovolit ztrátu dat. Protokol UDP je vhodnější použít u aplikací, kde je vyžadováno zpracování velkého množství dat v reálné čase.

1.5 Norma IEEE 802.11

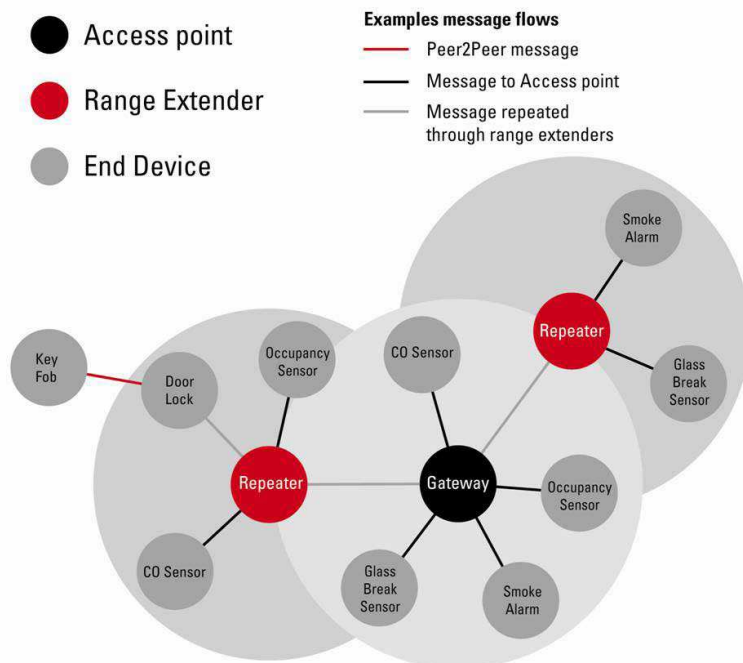
Norma IEEE 802.11 v sobě zahrnuje hned několik standardů, které popisují bezdrátovou komunikaci v počítačových sítích. V původní normě IEEE 802.11 vydané v roce 1997 se hovoří o přenosu dat v bezlicenčních pásmech radiových a infračervených vln 2,4 až 2,4835 GHz a 300 až 428 GHz. Garantovaná rychlost této normy je 1 – 2 Mb/s. Tato rychlost postupem času přestala být dostatečná, proto byly specifikovány další normy jako např. IEEE 802.11b. Tato norma byla schválena v roce

1999, specifikuje pouze jediné pásmo, a to 2,4 GHz, a zvyšuje rychlost na 11 Mb/s. Dnes jsou tyto normy označovány souhrnným názvem Wifi. [6]

Původní použití Wifi se nacházelo ve vzájemném bezdrátovém propojení přenosových zařízení a následném připojení do lokálních (firemních, kancelářských, atd.) sítí. Dnes nachází hlavní využití v možnosti bezdrátového připojení k síti Internet. Dalším využitím může být oblast telemetrie a vzdáleného řízení různých zařízení.

1.6 Komunikační protokol SimpliciTI

SimpliciTI je jednoduchý síťový protokol zaměřený na malé RF sítě (do 100 koncových bodů) navržený společností Texas Instruments. Pro zařízení těchto sítí bývá typické použití baterií jako zdrojů energie. Hlavním požadavkem na tyto zařízení je dlouhá výdrž baterie. S ohledem na tento požadavek je protokol navržen tak, aby co nejméně energeticky zatěžoval systém, proto vykazuje nízkou přenosovou rychlost dat, malý pracovní cyklus a je zde také omezen počet vzájemně komunikujících uzlů, které spolu mohou komunikovat přímo nebo prostřednictvím přístupových bodů, příp. opakovačů. Ve struktuře sítě není nutné používat přístupové body nebo opakovače, ale poskytují některé funkce jako ukládání nebo přeposílání zpráv. SimpliciTI má minimální nároky na mikrokontrolér, to snižuje cenu systémů. Protokol podporuje širokou škálu aplikací, jako jsou bezpečnostní senzory a alarmy (detektory kouře, tříštivého skla, světelné senzory), automatické měřicí systémy (plynoměry, vodoměry), domácí automatizace (garážová vrata). [5]



Obrázek 4 – Síťová topologie SimpliciTi – převzato z [5]

SimpliciTI byl navržen pro snadnou implementaci a nasazení pro několik TI RF platform, jako jsou např. obvody z rodiny MSP430 s nízkou spotřebou. Protokol je

dostupný pod volnou licenci a zdrojové kódy lze získat na webových stránkách TI (Texas Instruments).[5]

Protokol SimpliTI i přes svou jednoduchost a nenáročnost nabízí hned několik topologických variant. Podporuje komunikaci rovný s rovným, dále pak připojení přístupového bodu pro možnosti ukládání a další manipulací s předávanými daty. Případně je možné rozšířit rozsah sítě pomocí opakováčů. Budoucí verze protokolu budou přidávat další sofistikované funkce.

Na obrázku 4 je zobrazen příklad síťové topologie protokolu SimpliTI. Tento systém obsahuje jeden přístupový bod, který má za úkol přijímat data od koncových bodů a opakováčů. Opakovače rozšiřují komunikační rozsah. Můžeme zde pozorovat i typy komunikací (např. rovný s rovným, komunikaci opakováče a přístupového bodu, atd.) mezi jednotlivými body systému.

1.7 Pásmo pro průmyslové, vědecké a lékařské aplikace

ISM (Industrial, Scientist and Medical) je část frekvenčního spektra, která nepodléhá žádným licenčním poplatkům. Jak již z názvu vyplývá, toto pásmo je určeno pro průmyslová, vědecká a zdravotnická využití. V tomto pásmu nemusejí pracovat zařízení využívající pásmo pouze k přenosu informace, typickým příkladem takového zařízení je mikrovlnná trouba. Jelikož je pásmo volné, může mít neomezený počet uživatelů, vzhledem k tomuto faktu mohou být tato pásma značně zarušená.

Tabulka 1 – Označení a frekvenční rozsah pásem ISM (převzato z [16])

Ozn.	Kmitočtové pásmo
<i>a</i>	6 765 – 6 795 kHz
<i>b</i>	13 553 – 13 567 kHz
<i>c</i>	26,957 – 27,283 MHz
<i>d</i>	40,660 – 40,700 MHz
<i>e</i>	138,200 – 138,450 MHz
<i>f</i>	433,050 – 434,790 MHz
<i>f1</i>	433,050 – 434,790 MHz
<i>f2</i>	433,050 – 434,790 MHz
<i>g</i>	868,000 – 868,600 MHz
<i>h</i>	868,700 – 869,200 MHz
<i>i</i>	869,300 – 869,400 MHz
<i>j</i>	869,400 – 869,650 MHz
<i>k</i>	869,700 – 870,000 MHz
<i>l</i>	2 400 – 2 483,5 MHz
<i>m</i>	5 725 – 5 875 MHz
<i>n</i>	24,000 – 24,250 GHz
<i>o</i>	61,0 – 61,5 GHz

V radiofrekvenčním spektru je takových pásem hned několik. Podmínky provozu v těchto volných pásmech na lokální úrovni stanovuje český telekomunikační úřad (licence GL-12/R/2000, případně GL-30/R/2000). Mimo jiné, do tohoto pásma spadají také komunikační standardy, jako jsou Wifi nebo Bluetooth.

Tabulka 1 zobrazuje konkrétní frekvenční pásma, jejich označení a frekvenční rozsah. Pásma f , $f1$, $f2$ mohou sloužit pro přenos hovorových a akustických signálů. Přenos obrazových informací je možný pouze v pásmu l a ostatní pásma mohou být použita pro různé průmyslové, vědecké nebo lékařské účely. Zařízení užívaná v těchto frekvenčních pásmech by měla vyzařovat co nejmenší možné rušení.[16]

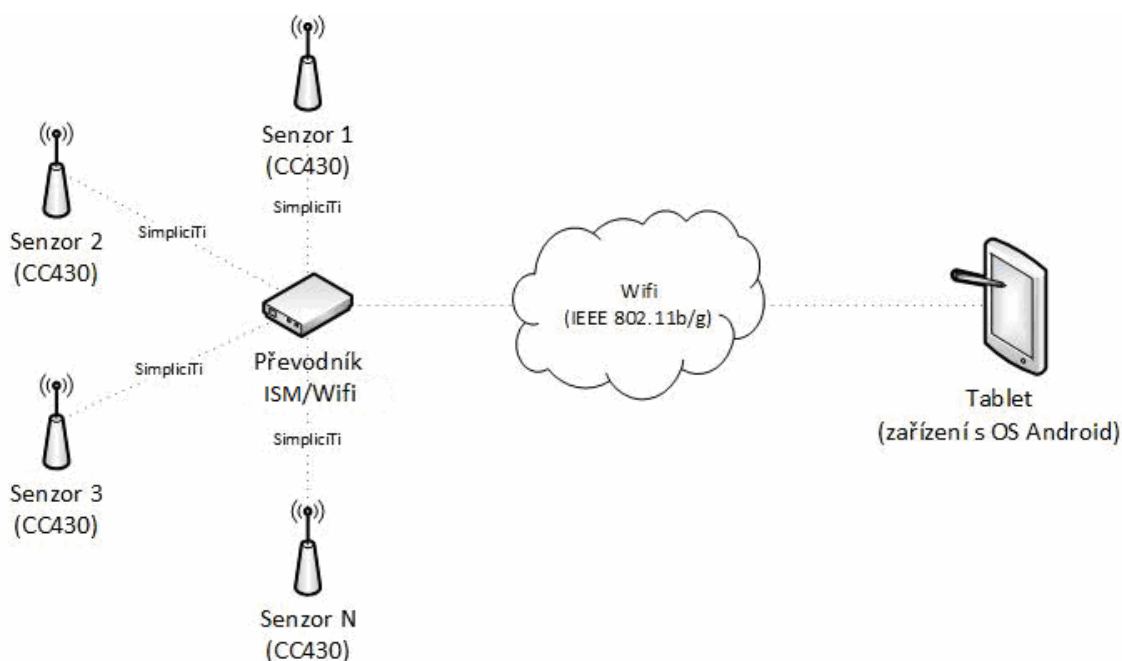
2 NÁVRH APLIKACE

Návrh aplikace pro správu a archivaci dat probíhá dle standardizovaných metod jazyka UML. Modelovací jazyk UML se snaží sjednotit názvosloví a metodiku postupu návrhu aplikací. Výsledkem kvalitního návrhu aplikace by měl být soubor diagramů, které koncovému uživateli osvětlí mechanismy, pomocí kterých bude dosaženo stanovených požadavků.

V jedné části této kapitoly se budeme zabývat celkovým pohledem na systém měření teploty a vlhkosti. Zbytek kapitoly bude věnován pouze návrhu aplikace pro správu a archivaci naměřených dat.

2.1 Systém měření teploty a vlhkosti

Systém měření teploty a relativní vlhkosti ovzduší, jak můžeme vidět na obrázku 5, se skládá ze sensorické sítě, převodníku ISM/Wifi a aplikace pro správu a archivaci dat.



Obrázek 5 – Systém pro měření teplot

Senzorická síť je tvořena jednoduchou hvězdicovou topologií, která je v podobných aplikacích využívána. Jednotlivé senzory obsahují mikrokontroléry řady CC430, které v sobě integrují 1 GHz RF radiové rozhraní a mnoho dalších periférií a rozhraní. Tento mikrokontrolér je využit jak pro koncové zařízení, tak i pro centrální jednotku (převodník ISM/Wifi). Komunikace mezi převodníkem a senzory je tvořena pomocí protokolu SimpliciTI.

Převodník ISM/Wifi distribuuje data obdržaná od jednotlivých senzorů aplikaci pro

správu a archivaci naměřených dat. Další funkcí převodníku je identifikace jednotlivých senzorů. Převodník obsahuje čtečku RFID (Radio Frequency Identification), pomocí které je zjištěn transpondér senzoru a ten odeslán aplikaci, která následně zobrazí uživateli informace o daném senzoru.

Aplikace pro správu a archivaci dat bude určena pro zařízení s operačním systémem Android. Z toho vyplývá, že aplikace bude určena pro přenosná zařízení, jako jsou tablety, „chytré“ telefony, atd.

2.2 Stanovení požadavků na aplikaci

Stanovení požadavků obvykle probíhá komunikací zákazníka, který se poptává po zhotovení konkrétní aplikace, a samotného zhotovitele aplikace. V našem případě probíhalo stanovení požadavků při konzultaci s vedoucím práce.

2.2.1 Komunikační požadavky

Tyto požadavky stanovují nároky na komunikaci mezi převodníkem ISM/Wifi a aplikací pro správu a archivaci dat.

IP adresu TCP serveru bude moci nastavovat uživatel, port bude nastaven pevně. Aplikace bude schopna podávat informace o stavu připojení systému i uživateli (signál *connected*, signál *error*, informace o přenesených datech, pokus o znovunavázání spojení). Komunikace na rozhraní převodník ISM/Wifi – aplikace bude definována komunikačním protokolem a každá úspěšně obdržená zpráva od převodníku musí být aplikací potvrzena.

2.2.2 Požadavky na správu senzorové sítě

Aplikace bude schopna dle nastavených pravidel koordinovat senzorovou síť. Sensory budou děleny do třech seznamů. Trvalý seznam – v tomto seznamu se nacházejí všechny senzory, které byly k aplikaci někdy připojeny, aktivní seznam – senzory z trvalého seznamu, od kterých v poslední době byla přijata data (poslední doba – uživatelem nastavitelný časový údaj), blokující seznam – senzory z trvalého seznamu, od kterých nebudou přijímána žádná data.

Aplikace bude zaznamenávat každý senzor, který kdy byl k aplikaci připojen (trvalý seznam). Uživatel bude moci jakýkoli senzor z tohoto seznamu vymazat, a to včetně jím naměřených hodnot. Naměřené hodnoty konkrétním senzorem budou moci být smazány i samostatně, bez toho aniž by byl senzor odstraněn z trvalého seznamu. Uživatel bude moci blokovat dočasně nebo trvale přidávání nových senzorů a přijímání jejich dat, dále pak přidávat senzory na blokující seznam (naměřené hodnoty od senzorů na blokujícím seznamu nebudou přijímána). Aplikace bude zobrazovat informace o senzoru, který bude identifikován čtečkou RFID.

2.2.3 Požadavky na způsob ukládání naměřených dat

Hlavní úložiště sensorických dat v aplikaci bude reprezentovat datový kontejner, který v sobě ukládá instance jednotlivých senzorů, dále v sobě ukládá datový kontejner pro ukládání jednotlivých zpráv ze senzorů.

Všechny objekty, které v sobě nesou naměřená data nebo data o senzorech, budou serializovatelné. Aplikace před svým ukončením bude ukládat data do binárního souboru pomocí serializace dat, naopak po spuštění aplikace jsou data načtena z binárního souboru pomocí deserializace dat. Aplikace bude umožňovat automatickou zálohu dat v průběhu své činnosti, jako prevenci před ztrátou dat, perioda automatického uložení je uživatelem nastavitelná v minutách, data jsou ukládána do dočasného souboru, který je při skončení aplikace smazán (po vytvoření standardního záznamu).

Aplikace bude dále ukládat potřebná uživatelská nastavení a bude umožňovat ukládání dat do souboru CVS (Comma Separated Value).

2.2.4 Požadavky na grafickou interpretaci dat

Naměřená data budou uživateli prezentována dvěma způsoby, a to tabelárně a graficky. Data vykreslená pomocí tabulky bude možné filtrovat dle různých parametrů (vlhkosti, teploty, napětí akumulátoru, času posledního přijatého paketu, identifikátoru senzoru). Data prezentovaná pomocí grafu bude možné zobrazovat pro více senzorů najednou a v definovaném časovém úseku.

Aplikace bude schopna tvořit přehledové statistiky pro jednotlivé senzory (počet přijatých zpráv, průměrné hodnoty naměřených veličin, atd.)

2.2.5 Požadavky na grafické uživatelské rozhraní

Grafické rozhraní bude adaptovat své prvky na aktuální rozlišení okna, resp. obrazovky. Návrh grafického rozhraní bude respektovat ustálené zvyky v ovládání aplikací s grafickým rozhraním. Ovládání aplikace bude intuitivní a „uživatelsky přívětivé“.

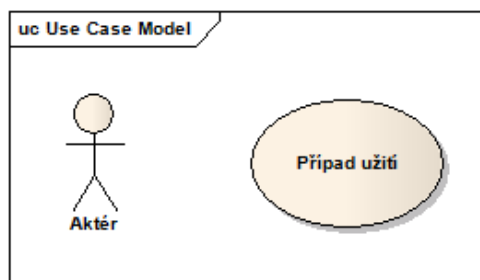
2.3 Návrh aplikace s využitím unifikovaného modelovacího jazyka

Unifikovaný modelovací jazyk (dále jen UML) slouží pro vizuální modelování systémů. Nejčastěji slouží k modelování objektově orientovaných systémů. UML sjednocuje nejlepší modelovací techniky a unifikuje názvosloví a postupy návrhu systémů. Jedním z hlavních úkolů modelovacího jazyka UML je přiblížit funkci a mechanismy, pomocí kterých byla daná funkčnost docílena, koncovému uživateli popřípadě zákazníkovi.

2.3.1 Diagram případů užití

Diagram případů užití se používá k popisu chování systému z hlediska uživatele a zachycuje, které typy uživatelů se systémem pracují a jaké činnosti v rámci systému

vykonávají. Tím vymezuje rozsah vlastností daného systému. Základními prvky diagramů jsou aktér a případ užití (schématické značky jsou zobrazeny na obrázku 6). Aktér specifikuje roli uživatele, který daný systém používá, definuje rozsah jeho působnosti. Do role aktéra může být dosazen uživatel, systém nebo i čas. Případ užití definuje určitou funkcionalitu systému, kterou využívá aktér a která plní určitý cíl. Mezi aktéry a případy užití se vyskytují určité komunikační vztahy – relace, ty znázorňují tok informací. V našem diagramu jsou použity pouze dva typy relací, a to asociace a include. Asociace představuje spojení dvou objektů, relaci include používáme, pokud je případ užití volán z více objektů. [7][8]



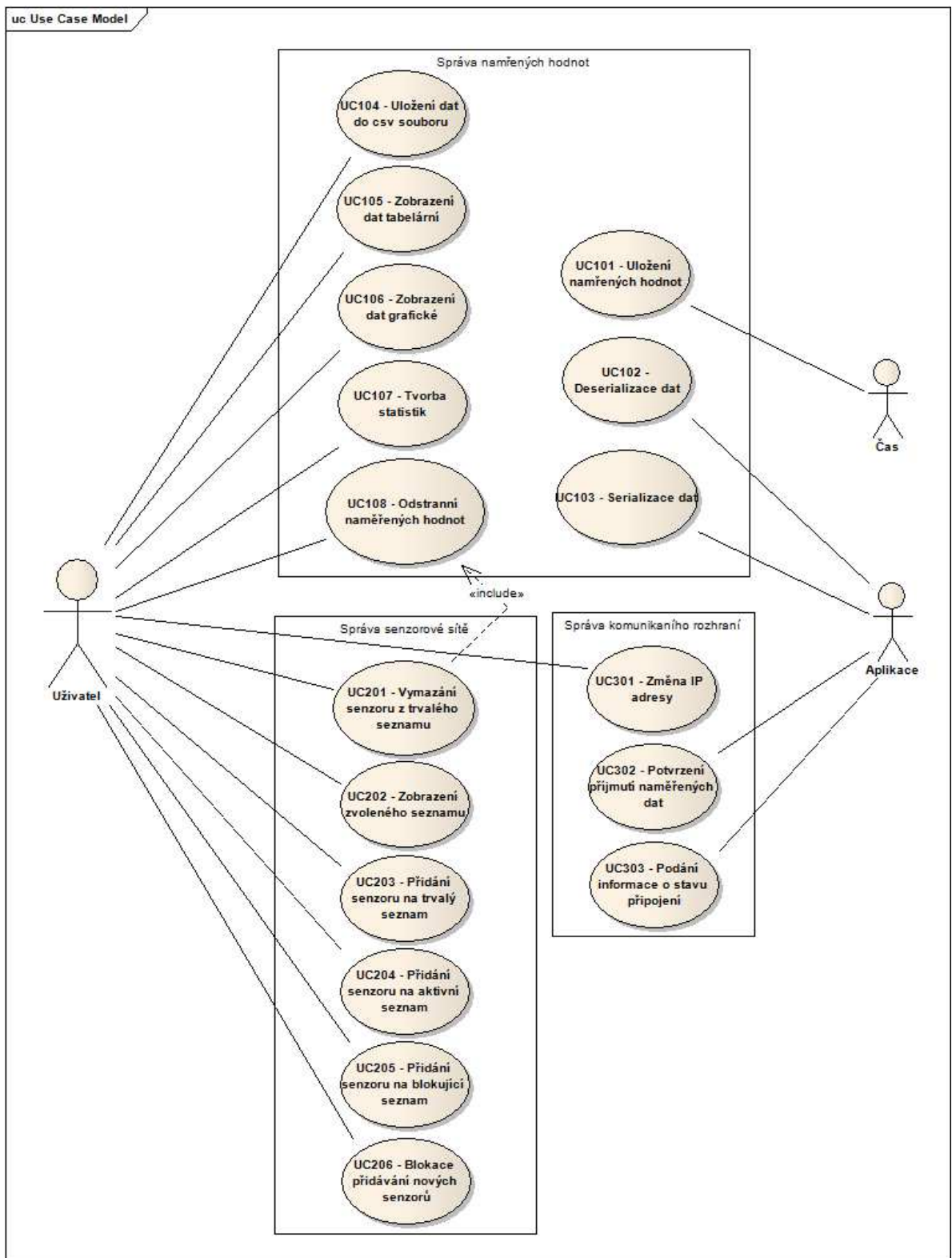
Obrázek 6 – Schématické značky diagramů případů užití

Dle výše stanovených požadavků byl vytvořen diagram případů užití, který je zobrazen na obrázku 7. Diagram přehledně zobrazuje funkcionalitu celé aplikace. Aplikace obsahuje tři aktéry, a to uživatele, aplikaci a čas, a je členěn do tří subjektů: „Správy naměřených hodnot“, „Správa senzorové sítě“ a „Správa komunikačního rozhraní“.

Aktér **uživatel** řídí celou aplikaci, jeho pravomoci zasahují do všech subjektů, ovšem neovládá úplně všechny případy užití. Mezi jeho pravomoci spojené se správou senzorové sítě patří ukládání nových a odstraňování stávajících senzorů, přemísťování senzorů mezi jednotlivými seznamy (aktivní, trvalý, blokový), zobrazování těchto seznamů a blokování dočasného přidávání nových senzorů. Dalšími případy užití, kterými uživatel disponuje, patří do subjektu „Správa naměřených hodnot“. Jsou to zobrazování naměřených hodnot jak ve formě tabelární tak i grafické, odstranění naměřených hodnot konkrétním senzorem, tvorba statistik a ukládání hodnot do csv souboru. Pravomoc aktéra zasahuje i do subjektu „Správa komunikačního rozhraní“, kde může měnit IP adresu TCP serveru, ke kterému se bude zařízení připojovat.

Aktéru **čas** přísluší pouze jediný případ užití, a to ukládání dat do dočasného souboru. Aktér čas představuje uživatelem definovaný časový okamžik, po jehož uplynutí proběhne záloha dat.

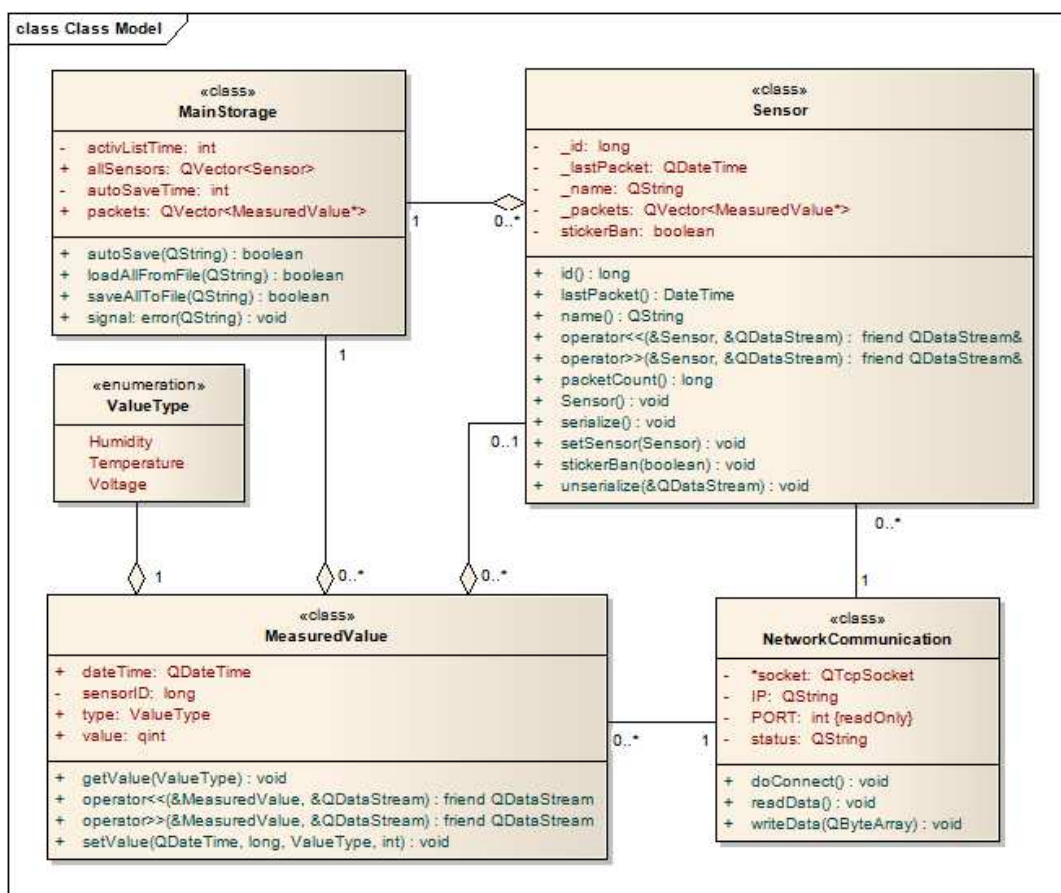
Aktér **aplikace** představuje aplikaci samotnou. Případy užití serializace, resp. deserializace dat, příslušející subjektu „Správa naměřených hodnot“, aplikace automaticky provede po ukončení, resp. spuštění aplikace. Další dva případy užití, které jsou součástí subjektu „Správa komunikačního rozhraní“, aplikace využívá během svého chodu. Případ užití „Podání informace o stavu připojení“ neustále informuje uživatele o stavu komunikace mezi aplikací a převodníkem ISM/Wifi. Případ užití „Potvrzení přijmutí naměřených dat“ informuje uživatele o bezchybném obdržení naměřených hodnot.



Obrázek 7 – Use case diagram aplikace

2.3.2 Návrh diagramu tříd

Modelování diagramu tříd je nejdůležitější částí návrhu programu. Popisuje typy použitých objektů, vztahy mezi nimi, jejich atributy a metody. Při popisování analytických modelů jsou popisovány pouze klíčové třídy softwaru. U tříd jsou často uváděny pouze hlavní atributy a metody, pokud chceme popsat vztahy mezi třídami, nemusejí třídy obsahovat žádné atributy a metody. [7]



Obrázek 8 – Diagram tříd

Diagram tříd aplikace (viz. obrázek 8) obsahuje čtyři hlavní třídy, jsou to *MainStorage*, *Senzor*, *MeasuredValue* a *NetworkCommunication*. Třída *MainStorage*, neboli hlavní úložiště, obsahuje dva kontejnery. Kontejner ukazatelů typu *MeasuredValue* a kontejner ukazatelů typu *Senzor*. Tyto kontejnery představují datová úložiště, kam aplikace ukládá informace o senzorech a naměřených hodnotách. Metody třídy zajišťují nahrání dat, která jsou uložena v textovém souboru, do kontejneru po spuštění aplikace a ukládání dat před ukončením aplikace zpět do textového souboru. Třída se dále stará o automatickou zálohu dat do dočasného souboru – ochrana před ztrátou dat.

Třída *Senzor* definuje vlastnosti senzoru, jako jsou například jeho id, jméno, poslední přijatý paket a kontejner ukazatelů typu *MeasureValue*, ve kterém jsou uloženy všechny naměřené hodnoty konkrétním senzorem. Třída *Senzor* obsahuje metody, které definují, co můžeme z instancí třídy *Senzor* dělat, například změnit, příp. přičíst,

jednotlivé atributy senzoru, serializovat a deserializovat data a další.

MeasureValue představuje třídu, do jejichž instancí se budou ukládat naměřená data. Hlavním atributem, který rozlišuje, zdali jde o hodnotu teploty, relativní vlhkosti vzduchu nebo napětí, je atribut *type*, který je datového typu *ValueType*. Tento datový typ je námi definován a může nabývat hodnot *Humidity*, *Temperature* a *Voltage*. Třída obsahuje i další atributy jako čas měření, id senzoru a samotnou naměřenou hodnotu.

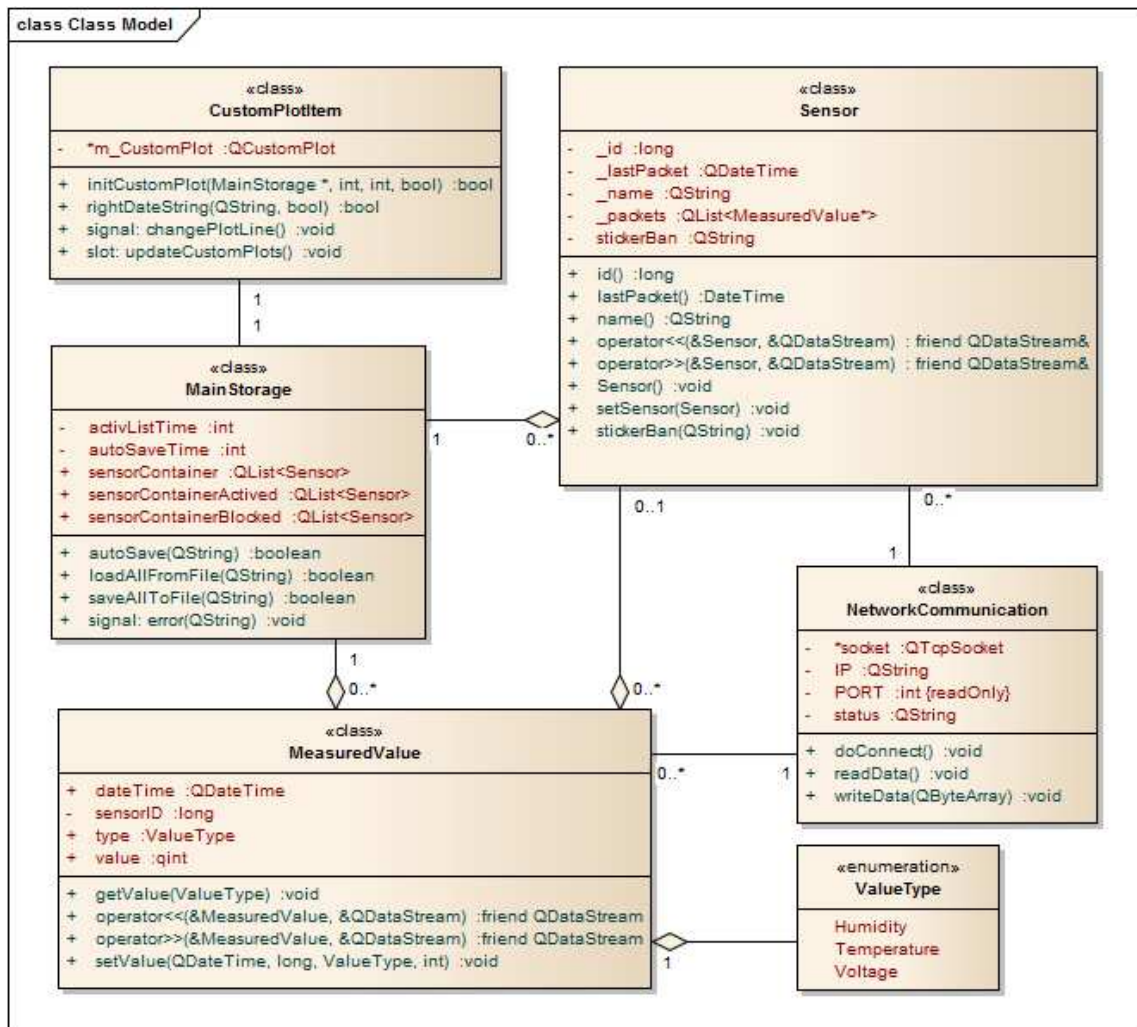
Třída *NetworkCommunication* obstarává komunikaci s převodníkem ISM/Wifi. Tato komunikace probíhá pomocí protokolu TCP. K vytvoření komunikačního rozhraní využíváme třídu *QTcpSocket*, která obsahuje všechny potřebné metody a signály. *NetworkCommunication* obsahuje i další atributy, však pouze jediný může definovat uživatel a to je atribut adresa IP. Hlavními metodami jsou zajisté čtení, odesílání dat a výstavba spojení. Odesílaná a přijímaná data jsou ve formě pole bajtů.

3 PROGRAMOVÁ IMPLEMENTACE APLIKACE

Tato kapitola pojednává o samotné realizaci aplikace pro sběr, vyhodnocování a ukládání naměřených dat ze sítě bezdrátových senzorů. Dále prezentuje uživatelské prostředí aplikace a vysvětluje funkčnost jednotlivých ovládacích prvků.

3.1 Diagram tříd

Diagram tříd zobrazený na obrázku 9 popisuje programovou strukturu realizované aplikace. Diagram v rámci přehlednosti zobrazuje pouze názvy hlavních atributů a metod daných tříd.



Obrázek 9 – Diagram tříd realizované aplikace

Pokud tento model porovnáme s modelem navrženým, zjistíme, že se oba modely od sebe liší pouze minimálně. Na změně struktury reálného diagramu tříd, oproti navrženému, se podílela velkou měrou předešlá neznalost deklarativního jazyka QML, a jeho odlišností, které oproti běžnému Qt Widgets přináší.

Hlavní změnou oproti navrženému diagramu tříd je výskyt nové třídy a to *CustomPlotItem*, pomocí níž aplikace vytváří grafy. Hlavním atributem třídy je ukazatel *m_CustomPlot* typu *QCustomPlot*. Jedná se o Qt C++ widget určený pro tvorbu různých 2D charakteristik, grafů a tabulek. Dokáže vykreslovat charakteristiky v reálném čase a exportovat vytvořené grafy do souborů různých formátů (například JPG, BMP, PDF a dalších). Knihovnu vytvořil Emanuel Eichhammer a pro nekomerční účely je dostupná pod GPL licenci. Knihovna je primárně určená pro desktopové aplikace, proto se při její transformaci na přenosná zařízení setkáváme s několika problémy, jako je například přibližování oblasti grafu. [14]

Další změny v diagramu tříd vycházejí z uplatnění deklarativního jazyka QML k tvorbě grafického uživatelského prostředí. Jedná se o změnu typu datových kontejnerů, kde namísto typu *QVector* byl použit kontejner typu *QList* a rozšíření počtu datových kontejnerů tak, aby jednotlivé typy seznamů senzorů mohly být prezentovány uživateli pomocí objektu *ListView*. Záměna typu *QVector* za *QList* je výhodnější i z hlediska rychlosti přístupu k jednotlivým datům, zatímco *QVector* pro data alokuje paměť v souběžném rámci a přístup k datům uprostřed kontejneru je pomalejší, *QList* využívá k ukládání dat princip rozprostřené paměti.

3.2 Aktivity diagramy hlavních metod

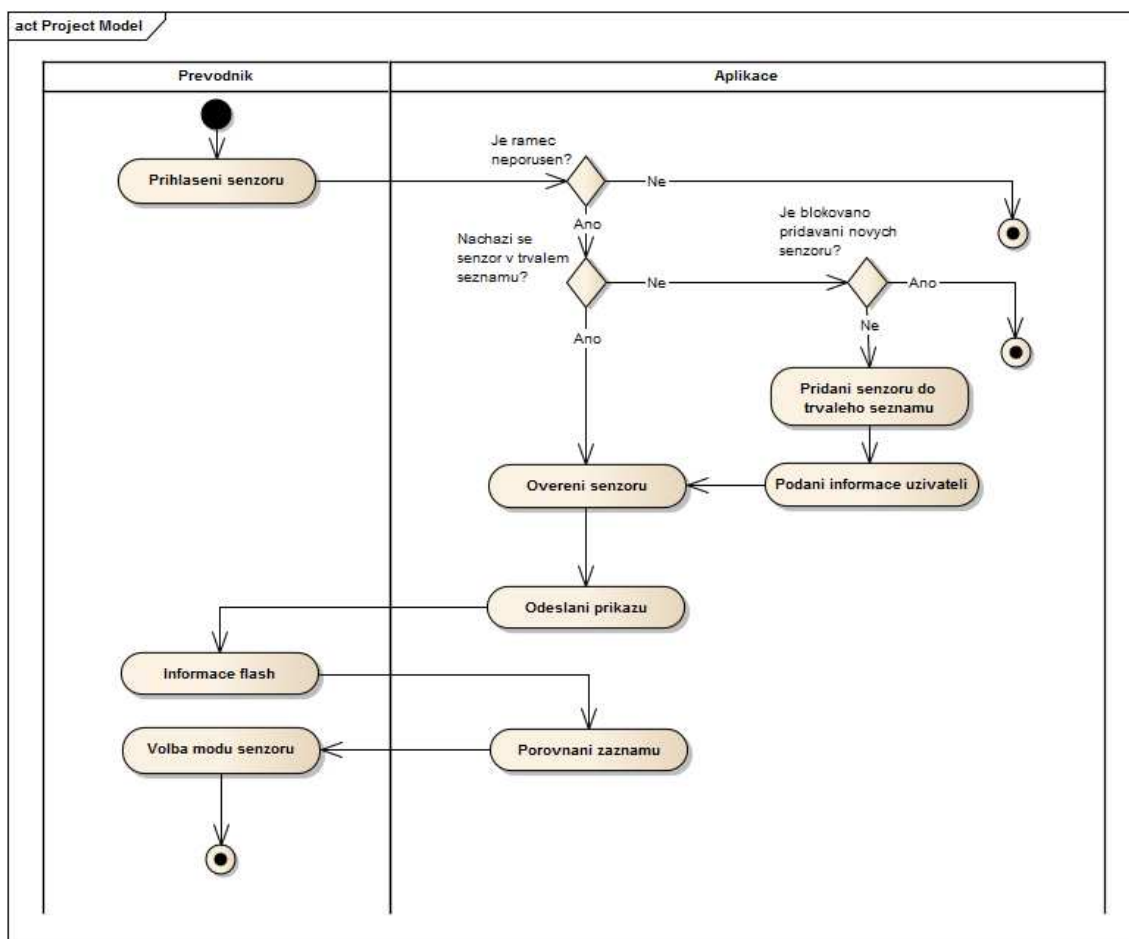
Aktivity diagramy jsou „objektově orientovanými vývojovými diagramy“ [8], které umožňují graficky popsat jednotlivé případy užití, posloupností akcí. Jednotlivé akce a uzly jsou spojeny komunikačními hranami. Lze s nimi popisovat jak procedurální logiku, tak i pracovní postupy a další. Aktivity diagramy byly v UML verze 1 chápány jako jakýsi druh stavových diagramů. Od verze 2 nabyly nové sémantiky a ta zjednodušila a zpřehlednila jejich návrh. [7][8]

3.2.1 Přihlášení senzoru

Jednou z hlavních případů užití, který je popsán diagramem aktivit zobrazeném na obrázku 10 je přihlášení, případně přidání nového senzoru do trvalého seznamu. Aktivita popsaná zmíněným diagramem probíhá následovně: poté, co aplikace obdrží datový rámec od převodníku ISM/Wifi, provede kontrolu tohoto rámce, pokud je obdržený rámec porušen, aktivita končí, v opačném případě pokračuje aktivita dále. Následuje ověření senzoru, zdali se senzor již nachází v trvalém seznamu nebo nikoli. Pokud se senzor v seznamu nenachází, je ověřena další podmínka, jestli je uživatelem povoleno přidávat do trvalého seznamu nové senzory. Pokud je přidávání nových senzorů povoleno, aplikace přidá daný senzor na trvalý seznam, podá informaci uživateli, že byl nový senzor přidán na seznam a aktivita pokračuje dále. Jestliže je uživatelem blokováno přidávání nových senzorů, aplikace ukončí svou aktivitu.

V případě, že příchozí žádost o přihlášení do systému pocházela od senzoru obsaženém v trvalém seznamu nebo od senzoru do tohoto seznamu úspěšně přidaného,

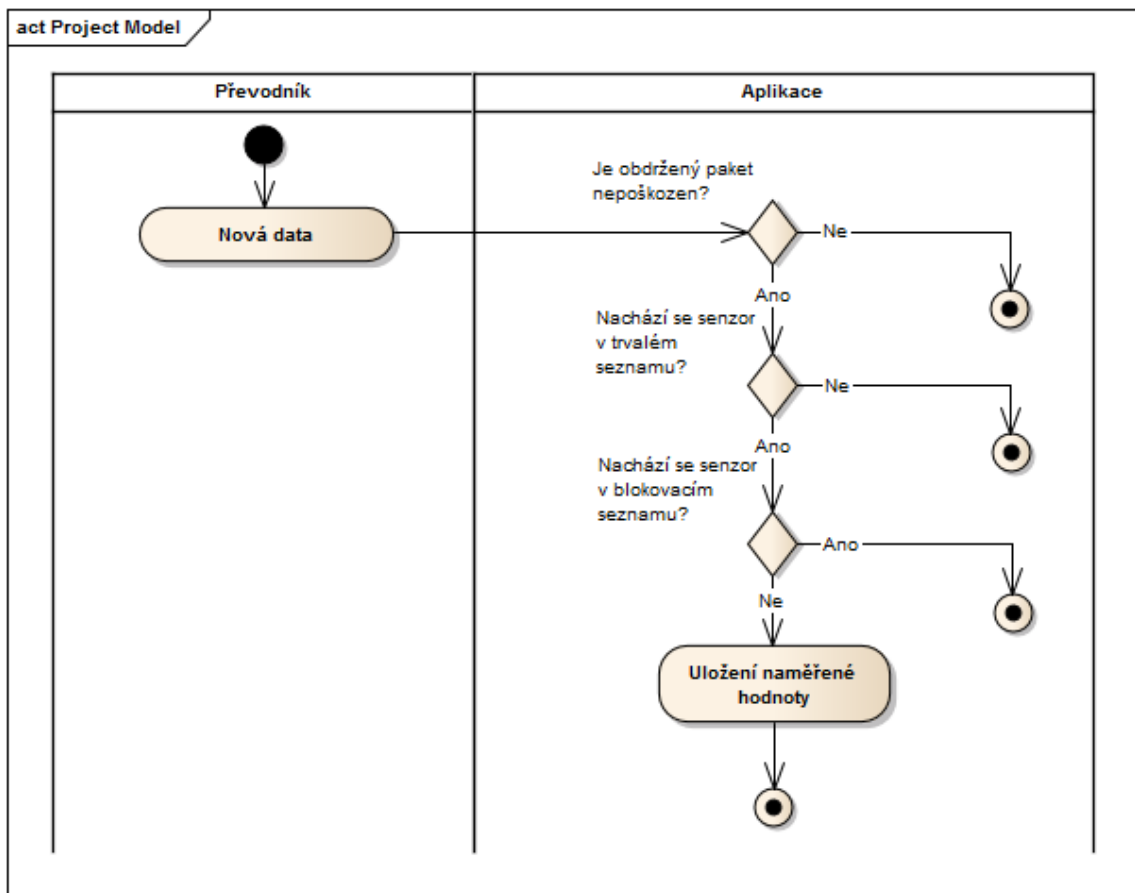
odesílá aplikace převodníku, respektive senzoru, žádost o stavu flash paměti. Po obdržení informační zprávy aplikace zkontroluje, zdali se poslední uložený záznam senzoru shoduje s posledním záznamem v paměti flash a odesílá příkaz, který nastaví senzor do *wireless* modu. Pokud se čas posledního záznamu neshoduje, vysílá aplikace příkaz pro nastavení senzoru do *flash* modu a aktivita je ukončena.



Obrázek 10 – Diagram aktivit – přihlášení senzoru

3.2.2 Obdržení naměřených hodnot

Popis jednotlivých metod probíhá v logickém postupu jejich použití. Jakmile aplikace úspěšně identifikuje, případně uloží nový senzor, očekává jím naměřená data. Po obdržení takových dat nejprve kontroluje obdržení paket, zdali nebyl během přenosu porušen. Následuje kontrola výskytu senzoru v trvalém seznamu, pokud zde senzor není obsažen, jsou obdržená data ignorována. V opačném případě aplikace ověří, zdali se senzor nenachází v blokujícím seznamu, pokud je senzor v tomto seznamu obsažen, naměřená data jsou opět ignorována, pakliže se v tomto seznamu nenachází, jsou obdržená data uložena.



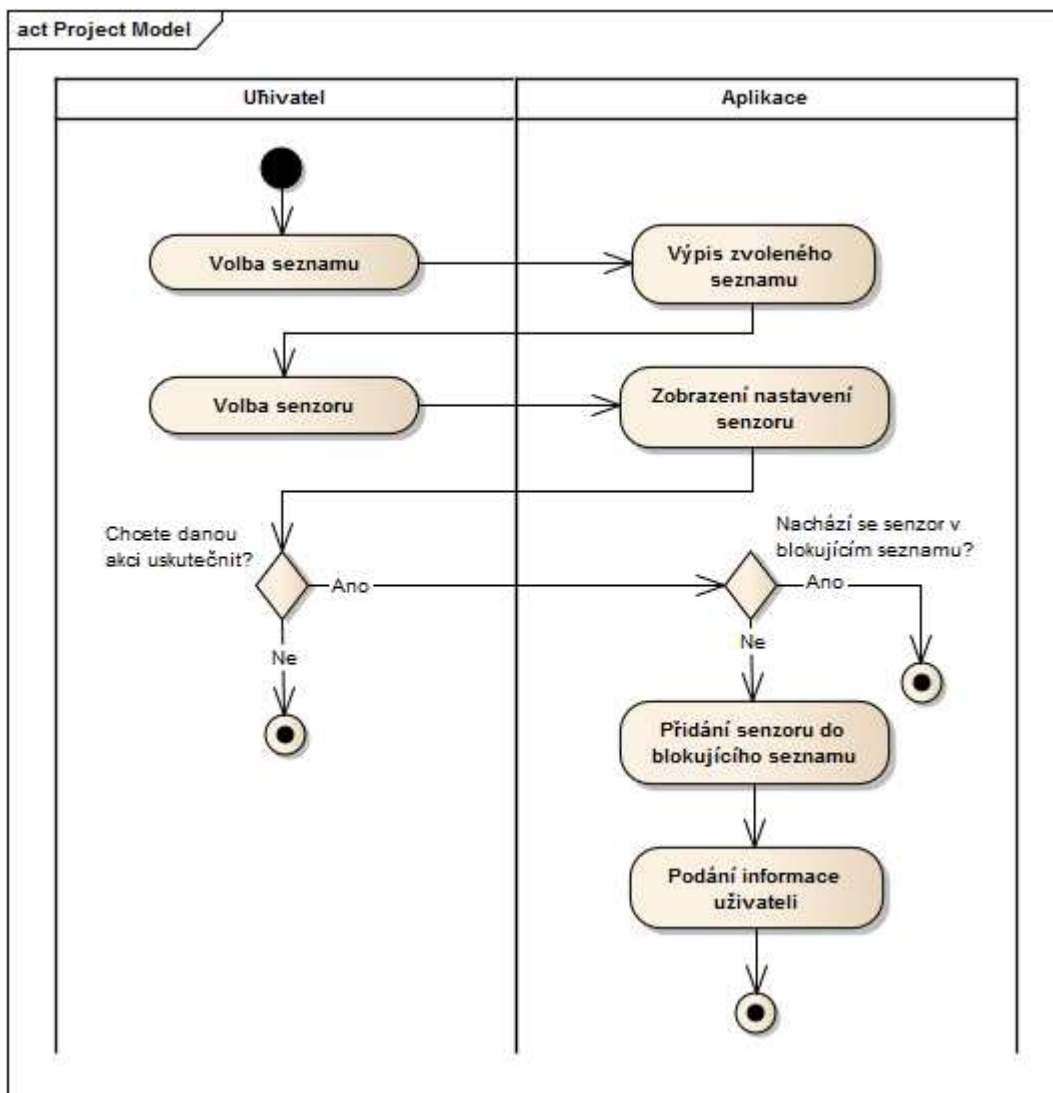
Obrázek 11 – Aktivita diagram – Obdržení naměřených hodnot

3.2.3 Přidání senzoru na blokující seznam

Aplikace nabízí uživateli možnost blokovat příchozí data od jakéhokoli senzoru. Takto označené senzory se nacházejí v blokovacím seznamu. Diagram aktivit zobrazený na obrázku 12 popisuje metodu, která zvolený senzor do tohoto seznamu vloží.

Aktivita vložení senzoru do blokujícího seznamu probíhá v tomto pořadí, uživatel nejprve zvolí typ seznamu. Aplikace mu tento seznam poskytne. Následuje výběr daného senzoru a zobrazení jeho nastavení, kde uživatel zvolí možnost blokování senzoru a potvrdí danou akci. Aplikace následně ověří, zdali se senzor na blokovacím seznamu již nenachází, pokud v seznamu není obsažen, aplikace senzor na tento seznam

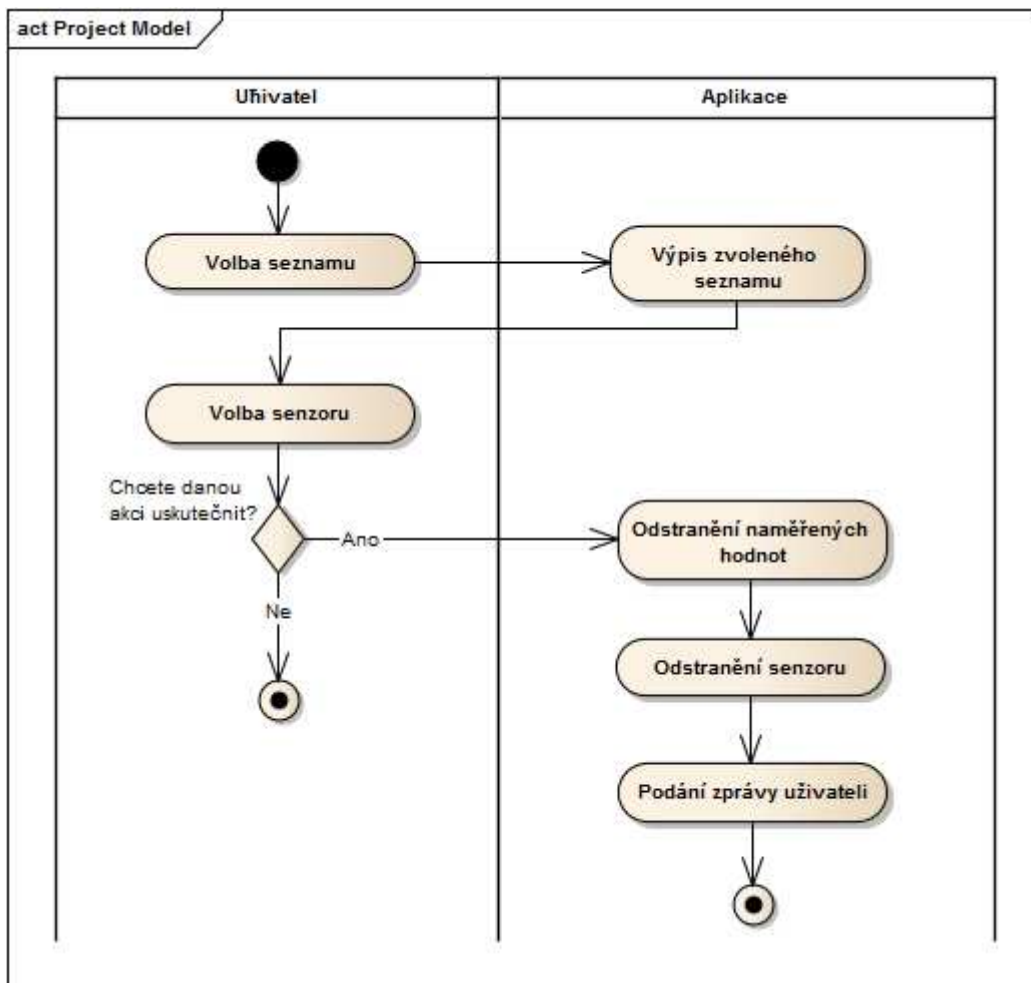
přidá. V opačném případě je aktivita ukončena. Posledním bodem aktivity po úspěšném přidání senzoru do blokovacího seznamu je podání zprávy uživateli a ukončení akce.



Obrázek 12 – Aktivita diagram – přidání senzoru do blokovacího seznamu

3.2.4 Odstranění senzoru

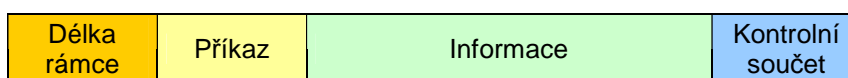
Posloupnost aktivit, která probíhá v případě užití – smazání senzoru z trvalého seznamu, popisuje diagram na obrázku 13. Uživatel volí typ seznamu, aplikace na základě volby uživatele poskytuje daný seznam. Uživatel ze seznamu vybere senzor, který chce smazat, a potvrdí akci. Pokud tuto akci nepotvrdí je celá aktivita ukončena. Aplikace nejprve odstraní hodnoty, které byly senzorem naměřeny, následně odstraní i informace o samotném senzoru. Poté aplikace podá zprávu uživateli o odstranění senzoru z trvalého seznamu. Tento případ užití by měl uživatel používat se zvýšenou pozorností, nýbrž odstranění dat je nevratné.



Obrázek 13 – Aktivity diagram – odstranění senzoru

3.3 Datový rámec

System pro měření teploty a relativní vlhkosti ovzduší využívá hned několik komunikačních protokolů. Na úrovni sensorické sítě je využito komunikačního protokolu SimpliciTI, komunikace mezi převodníkem ISM/Wifi a aplikací pro správu a archivaci dat je zajišťována síťovými protokoly IEEE 802.11. Naším úkolem je definovat podobu datového rámce, kterým si mezi sebou budou předávat zprávy a data převodník a aplikace tak, aby předávání zpráv bylo jednoznačné, bezpečné a efektivní.



Obrázek 14 – Obecný tvar rámce

Definujeme obecnou podobu rámce, pomocí kterého budou předávány konkrétní informace. Formát navrženého rámce je zobrazen na obrázku 14. Rámec obsahuje záhlaví, které udává délku rámce a příkaz, následují přenášená data. Velikost této části rámce je proměnná podle typu přenášených dat a kontrolní součet. Protokol disponuje

ochrannými prvky před ztrátou a poškozením dat. Délka rámce je daná součtem velikosti informace a kontrolního součtu, výsledkem je počet bajtů. Detekce příjmu poškozených dat zajišťuje kontrolní součet CRC, v našem případě se jedná o logický součin součtů všech bajtů rámce mimo bajt CRC s hodnotou 0xFF.

V rámci komunikace mezi převodníkem a aplikací je využíváno hned několik typů datových paketů. Tabulka 2 přehledně zobrazuje jejich podobu, tedy účel využití, typ přenášených dat a velikost v bajtech. Význam bajtů obsažených v hlavičce paketu (LNG, CMD) a posledního bajtu (CRC) je pro všechny typy paketů shodný. Význam ostatních bajtů v pořadí se může měnit.

Tabulka 2 – Typy přenášených paketů

Naměřená data													
0	1	2	3	4	5	6	7	8	9	10	11	12	13
LNG	CMD	ID0	ID1	ID2	ID3	RSSI	LQI	HUM	T0	T1	V0	V1	CRC
Data z flash													
0	1	2	3	4	5	6	7	8	9	10	11	12	13
LNG	CMD	ID0	ID1	ID2	ID3	RSSI	LQI	NO0	NO1	HUM	T0	T1	CRC
Informace o flash													
0	1	2	3	4	5	6	7	8	9	10	11	12	
LNG	CMD	ID0	ID1	ID2	ID3	RSSI	LQI	PRD0	PRD1	CNT0	CNT1	CRC	
Přihlášení do sítě													
0	1	2	3	4	5	6	7	8					
LNG	CMD	ID0	ID1	ID2	ID3	RSSI	LQI	CRC					
Identifikátor RFID													
0	1	2	3	4	5	6							
LNG	CMD	ID0	ID1	ID2	ID3	CRC							
Příkaz vyslaný aplikací převodníku													
0	1	2	3	4	5	6	7	8	9	10			
LNG	CMD	ID0	ID1	ID2	ID3	PRD0	PRD1	CNT0	CNT1	CRC			

LNG – délka paketu

LQI – kvalita spojení

PRD_x – perioda měření

CMD – typ paketu

T_x – teplota

CNT_x – pořadí záznamu

ID_x – identifikátor senzoru

V_x – napětí

NO_x – číslo záznamu

RSSI – síla přijatého signálu

HUM – relativní vlhkost

CRC – kontrolní součet

Přenášené pakety obsahují řídicí bajt, označený jako CMD, jehož hodnota určuje typ přenášených dat, případně typ příkazu. Hodnoty, kterých tento bajt může nabývat, a jejich konkrétní význam je uveden v tabulce 3. Detailní přehled provozu v rámci komunikace senzor-převodník-aplikace a výpis všech typů paketů s popisem významů jednotlivých bajtů je uveden v příloze 3.

Tabulka 3 – Hodnota a význam bajtu CMD

Hodnota	Význam
0x00	žádost o záznam
0x01	žádost o stav flash (perioda, počet záznamů)
0xE0	start wireless_mode
0xF0	start flash_mode
0xAA	přihlášení do sítě
0xCC	adresa RFID
0xBB	záznam z flash
0xDD	naměřená data
0xFF	smazat záznamy na flash

3.4 Realizace grafického uživatelského prostředí

Grafické uživatelské prostředí bylo vytvořeno s využitím QML, viz kapitola Teoretická část. Díky tomuto deklarativnímu jazyku lze jednoduše tvořit dynamické uživatelské prostředí.

Návrh a tvorba grafického uživatelského prostředí se řídila doporučeními uvedenými v [13]. Tyto stránky jsou věnovány vývoji aplikací určených pro operační systém Android, vývojář zde nalezne informace o každé části návrhu programu.

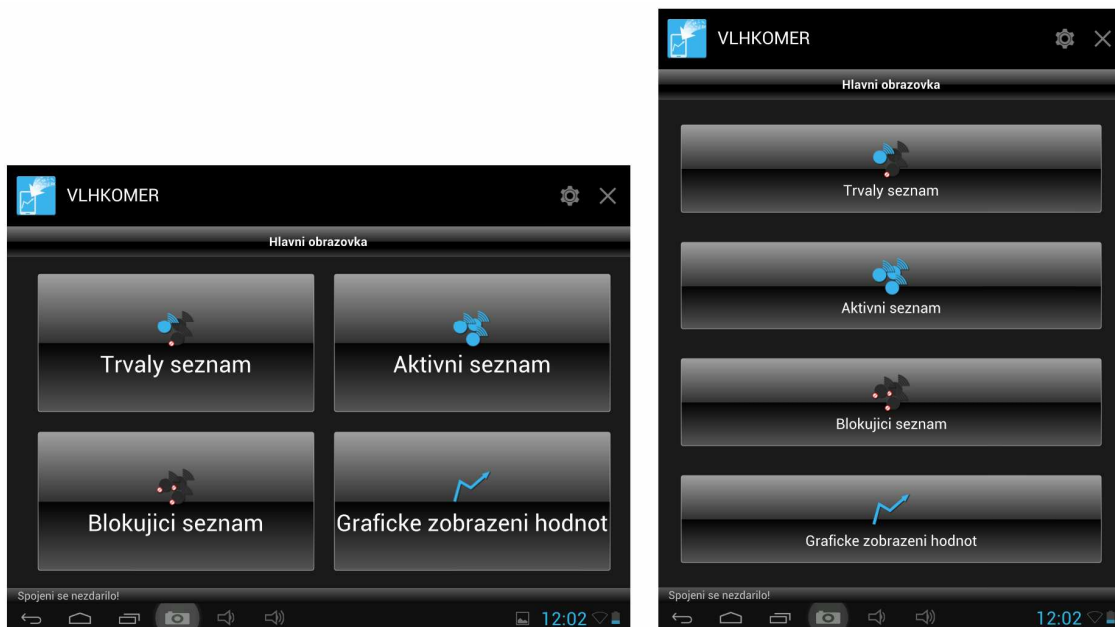
Při návrhu naší aplikace byla věnována největší pozornost přizpůsobení aplikace velikosti a hustotě zobrazovacích bodů obrazovky, proto se v této kapitole zaměřujeme na návrh layoutu a ikon.

3.4.1 Rozvržení obrazovky aplikace

Operační systém Android je v dnešní době velmi rozšířeným operačním systémem, který se vyskytuje na různých přenosných zařízeních mnoha druhů. S tímto souvisí jeden z hlavních problémů návrhu grafického uživatelského prostředí.

Vývojář musí počítat se skutečností, že jím navržená aplikace bude zobrazovaná na zařízeních s různým typem displejů – různé rozměry, velikosti zobrazovaných bodů a jejich hustota. Například pokud by byla navržená aplikace optimalizovaná na velikost displeje small, bude ten samý návrh nasazený na zařízení velikosti xlarge obsahovat spoustu volného místa. Proto se návrhy často optimalizují pro různé velikosti obrazovek. Dle [13] dělíme velikosti obrazovek do několika skupin.

Naše aplikace je navržena a vytvořena tak, aby se byla schopna dynamicky přizpůsobit jakékoli velikosti displeje. S tímto souvisí návrh aplikace, který je navržen tak, aby co nejpřehledněji zobrazoval informace koncovému uživateli. Úvodní obrazovka, jak je zobrazeno na obrázku 15, reaguje na orientaci obrazovky.



Obrázek 15 – Přizpůsobení aplikace rozměrům displeje

3.4.2 Tvorba ikon

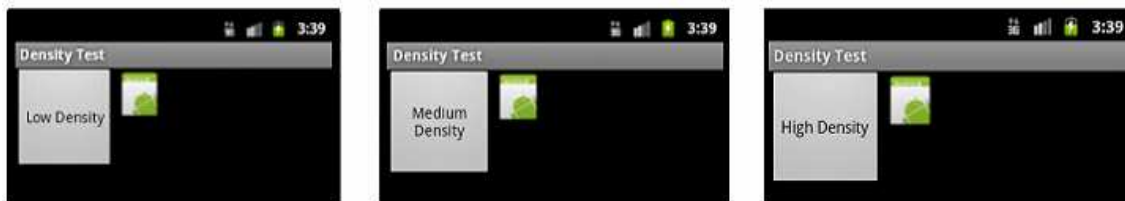
Ikony jsou malé obrázky, které poskytují rychlou a intuitivní orientaci v programu. Reprezentují nějakou akci, stav nebo aplikaci. Ikony by měli být tvořeny pomocí jednoduchých obrázků, které jasně evokují účel použití dané ikony.

Při návrhu vlastní ikony je nutné uvažovat o použití aplikace na různých zařízeních. Z čehož vyplývá, že ikony budou zobrazovány na displejích různých velikostí, rozlišení a hustoty (dpi). Vzhledem k rozmanitosti těchto parametrů jsou obrazovky kategorizovány do několika skupin, zaprvé podle velikosti obrazovky (small, normal, large, xlarge a další) a zadruhé podle hustoty zobrazovacích bodů obrazovky (ldpi, hdpi, xhdpi, xxhdpi a další).

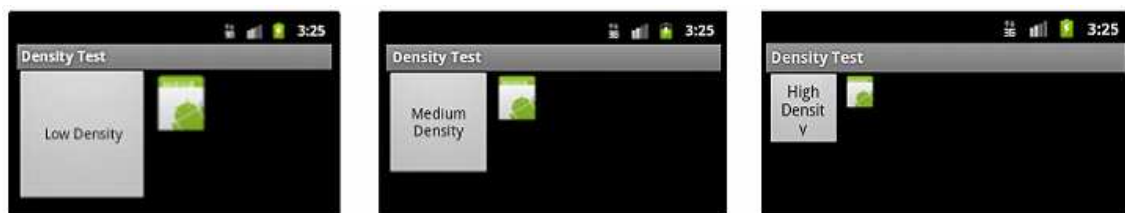
Pokud uvažujeme obrazovku s rozlišením 1280 x 720, bude pixel zobrazovaný touto obrazovkou jistě větší než pixel zobrazený menší obrazovkou se stejným rozlišením. Proto platforma Android zavádí novou jednotku, a to dp, což je virtuální pixel. Ten se vypočítá na základě hustoty obrazovky následovně:

$$px = \frac{dp * dpi}{160} \quad (1)$$

Pokud vytvoříme objekt, jehož velikost bude uvedena v jednotkách dp, nikoli v px, bude se tento objekt zobrazovat na různých obrazovkách podobně. Z tohoto vyplývá návrh ikon pro naši aplikaci, ta počítá s čtyřmi hustotami obrazovek (mdpi, hdpi, xhdpi a xxhdpi), proto byly vytvořeny čtyři sady ikon o velikostech určených konkrétní hustotě obrazovky. Pokud by velikosti jednotlivých grafických objektů byly uváděny pomocí jednotky px, mohlo by se stát, že s měnící se velikostí obrazovky by se snižovala čitelnost těchto objektů. Na obrázku 16 je zobrazen správný a na obrázku 17 chybný návrh aplikace.



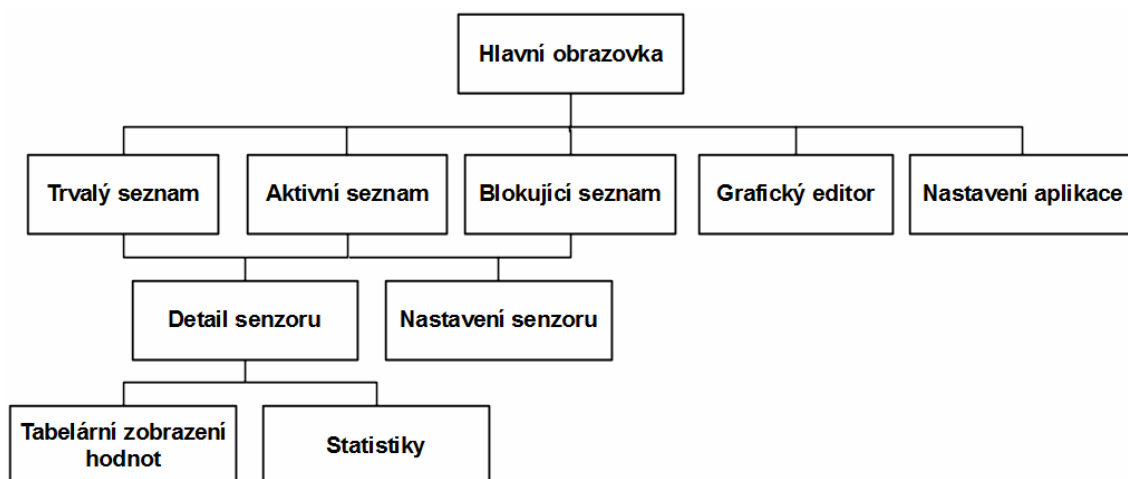
Obrázek 16 – Správný návrh grafických prvků - použití jednotky dp (převzato z [12])



Obrázek 17 – Špatný návrh grafických prvků - použití jednotky px (převzato z [12])

3.5 Struktura aplikace

Tato kapitola je věnována popisu jednotlivých obrazovek aplikace, jejich hierarchií a popisu jednotlivých funkčních bloků. Tuto kapitolu může čtenář chápat jako návod k použití aplikace. Obrázek 18 zobrazuje hierarchický model aplikace, tento model vykresluje hierarchii obrazovek a jejich návaznost. Jednotlivé obrazovky aplikace jsou rozděleny do čtyř úrovní. V nejvyšší úrovni, která je zároveň hlavním rozcestníkem aplikace, je úvodní obrazovka, odtud lze postupně dosáhnout každé úrovně obrazovek. Více k jednotlivým obrazovkám níže.



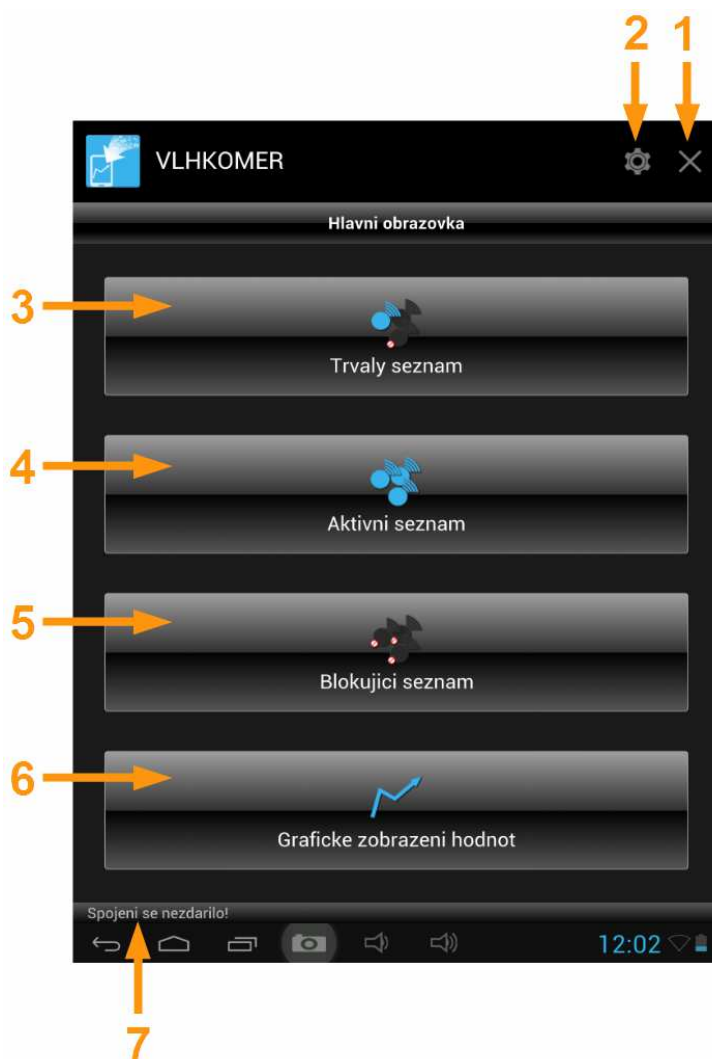
Obrázek 18 – Hierarchický model rozvržení aplikace

3.5.1 Hlavní obrazovka

Po spuštění aplikace je uživateli zobrazena hlavní obrazovka aplikace. Té dominují čtyři navigační tlačítka, pomocí nichž může uživatel volit mezi jednotlivými seznamy senzorů nebo grafickým zobrazením naměřených hodnot.

V pravém horním rohu jsou umístěna tlačítka pro nastavení aplikace (viz. kapitola Nastavení aplikace) a tlačítko pro vypnutí aplikace.

Ve spodní části obrazovky je umístěn informační řádek, pomocí něhož je uživatel informován o různých akcích, které aplikace provedla. Veškerá navigace v aplikaci je řešena tak, aby byla jasně čitelná a intuitivní, proto byla funkčnost většiny tlačítek vyjádřena ikonami namísto popisů.



Obrázek 19 – Úvodní obrazovka aplikace

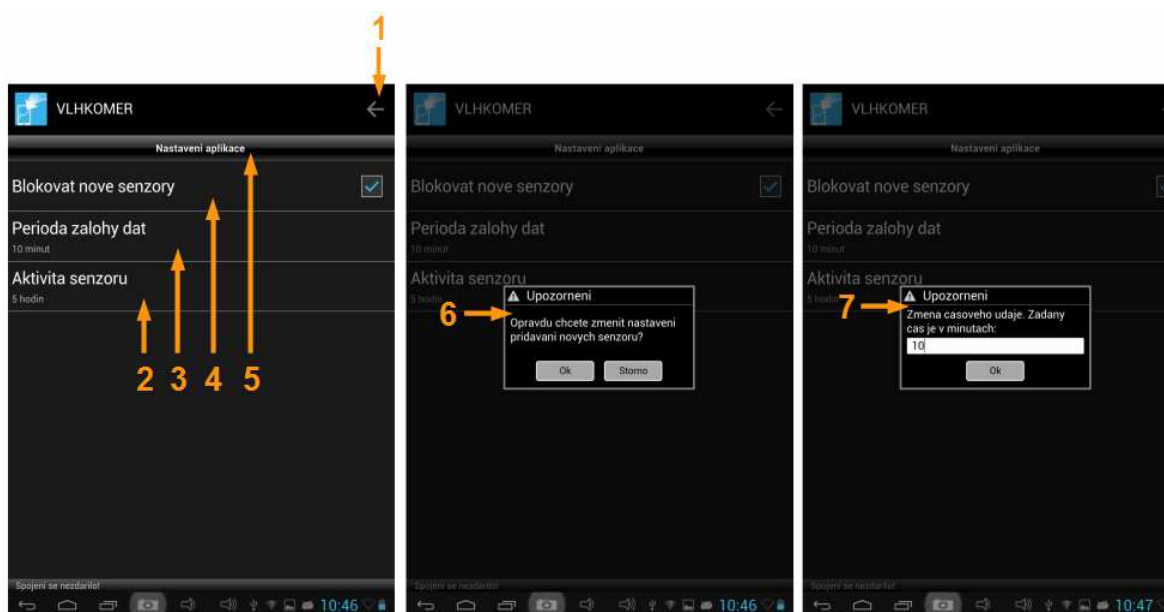
- | | |
|----------------------------|-------------------------------|
| 1 – vypnutí aplikace | 5 – blokující seznam senzorů |
| 2 – nastavení aplikace | 6 – grafické zobrazení hodnot |
| 3 – trvalý seznam senzorů | 7 – informační řádek |
| 4 – aktivní seznam senzorů | |

3.5.2 Nastavení aplikace

Obrazovka nastavení aplikace dává možnost uživateli nastavit několik základních parametrů. Hlavním kritériem návrhu aplikace bylo, aby aplikace a její nastavování bylo pro koncového uživatele jednoduché a srozumitelné. Aplikace tedy bude, i bez nutnosti nastavování jakýchkoli parametrů, spolehlivě fungovat.

V původním návrhu aplikace byla uvažována možnost, že k realizaci spojení mezi aplikací a převodníkem bude muset uživatel zadat IP adresu převodníku. Tato varianta byla v průběhu realizace aplikace zavržena a bylo realizováno automatické vyhledávání IP adresy převodníku. Z tohoto důvodu v tomto okně chybí možnost nastavení IP adresy převodníku.

Obsahem nabídky jsou tři položky. Pomocí první z nich lze zakázat přidávání nových senzorů do databáze aplikace. V základním nastavení je povoleno přidávání nových senzorů. Další položka je nastavení časového okamžiku, po jehož uplynutí jsou naměřená data automaticky zálohována. Tento časový údaj je zadáván v minutách, a pokud uživatel nezmění základní nastavení, tak je tento parametr nastaven na pět minut. Poslední položkou je nastavení dalšího časového údaje, a to nastavení periody, po jejímž uplynutí a zároveň neobdržením žádné naměřené hodnoty senzoru z aktivního seznamu („poslední doba“) bude tento senzor prohlášen za neaktivní. Poslední doba je udávána v hodinách a v základním nastavení je nastavena na jednu hodinu. Nastavení aplikace je ukládáno do binárního souboru obdobně jako informace o jednotlivých senzorech.



Obrázek 20 – Obrazovka – nastavení aplikace

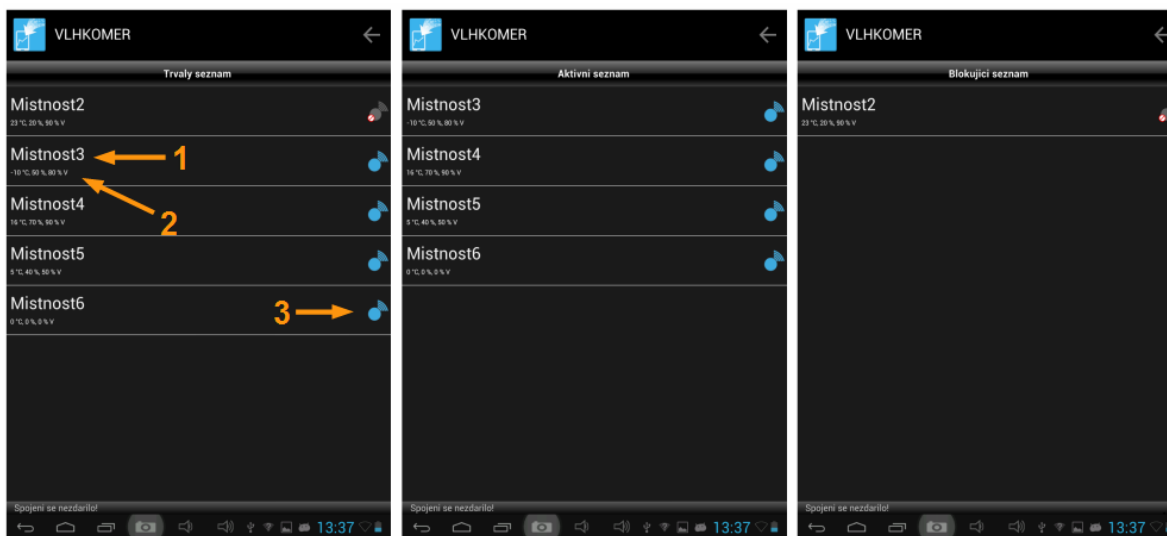
- | | |
|--|---|
| 1 – tlačítko zpět | 5 – navigační pole |
| 2 – perioda aktivity senzoru | 6 – dialogové okno – potvrzení nastavení |
| 3 – zálohování dat (nastavení časového okamžiku) | 7 – dialogové okno – nastavení nové hodnoty |
| 4 – blokování přidávání nových senzorů | |

Na obrázku 20 je zobrazena obrazovka nastavení a dále pak dialogová okna, kde uživatel musí zadat nové parametry a potvrdit jejich změnu. Jsou zde použity dva typy dialogových oken, pro položku „Blokovat nové senzory“ pouze potvrzení změny nastavení, pro zbylé dvě položky obdobné dialogové okno, kde uživatel musí zadat nový parametr a ten potvrdit.

3.5.3 Seznamy senzorů

Aplikace uvažuje několik seznamů senzorů. Rozdělení senzorů do těchto seznamů probíhá dle několika pravidel. V trvalém seznamu se nacházejí všechny senzory, od kterých kdy aplikace obdržela nějaká data. Aktivní seznam obsahuje pouze senzory, od kterých aplikace obdržela naměřená data během „poslední doby“. Posledním seznamem je seznam blokových senzorů. Do tohoto seznamu jsou umístěny senzory, od kterých si uživatel nepřeje přijímat žádné naměřené hodnoty.

Na obrázku 21 jsou zobrazeny jednotlivé seznamy. Hlavními zobrazovanými informacemi o senzoru jsou: jeho název, ten může být například nahrazen umístěním senzoru, a ikona senzoru (umístěná v pravé části řádku), ta přesně vystihuje, v kterém seznamu se sensor nachází. Dalšími zobrazovanými informacemi jsou senzorem změřené poslední hodnoty teploty, vlhkosti a napětí.



Obrázek 21 – Seznamy senzorů

1 – název senzoru

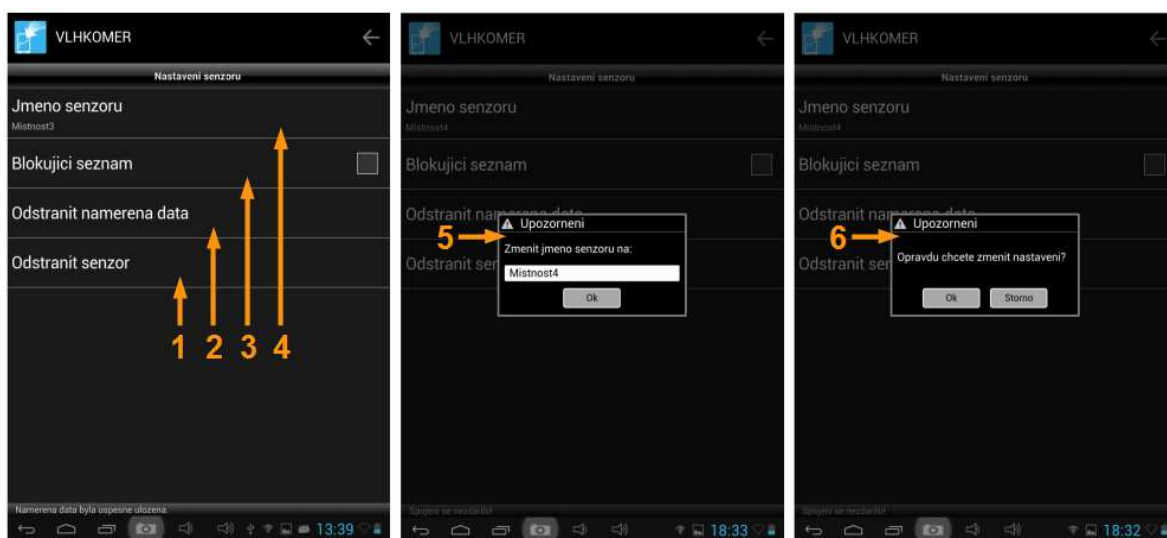
3 – status senzoru

2 – poslední naměřené hodnoty

3.5.4 Nastavení senzoru

Obrazovku nastavení senzoru zobrazí uživatel, pokud stiskne a podrží řádek déle než 800 ms v jakémkoliv ze seznamů. V této nabídce může uživatel definovat hned několik vlastností senzoru.

Aplikace dovoluje uživateli editovat název senzoru, zde je vhodné uvést například umístění senzoru. Vyplnění nového názvu se provede pomocí dialogového okna, které je zobrazeno na obrázku 22, jakož i samotná obrazovka „Nastavení senzoru“. Dalšími možnostmi je přidání/odebrání senzoru z blokovacího seznamu, odstranění naměřených hodnot a odstranění informací o samotném senzoru, použití posledních dvou položek by měl uživatel důkladně zvážit, neboť odstranění dat je nevratné.



Obrázek 22 – Obrazovka – nastavení senzoru

- | | |
|---|--|
| 1 – odstranění senzoru z trvalého seznamu | 4 – změna názvu senzoru |
| 2 – odstranění naměřených dat | 5 – dialogové okno – nastavení nové hodnoty |
| 3 – přidání senzoru na blokovací seznam | 6 – dialogové okno – potvrzení změny nastavení |

3.5.5 Detail senzoru

Po stisknutí konkrétního řádku v jakémkoli ze seznamů jsou zobrazeny detailní informace o zvoleném senzoru. V přehledném listu je uveden název senzoru, id, poslední přijatý paket a informace, zdali se senzor nachází v blokovacím seznamu či nikoli. Seznam obsahuje dva navigační řádky, které slouží k tabelárnímu zobrazení naměřených hodnot a zobrazení statistik (více kapitoly 3.5.6 a 3.5.7). Poslední položkou lze exportovat senzorem naměřená data do souboru CSV.



Obrázek 23 – Obrazovka – detail senzoru

- | | |
|--|---|
| 1 – název senzoru | 5 – navigační řádek – zobrazení naměřených hodnot |
| 2 – id senzoru | 6 – navigační řádek – zobrazení statistik |
| 3 – poslední přijatý paket | 7 – export naměřených dat do cvs |
| 4 – informace o umístění senzoru na blokovacím seznamu | |

3.5.6 Naměřené hodnoty

Naměřené hodnoty zobrazuje aplikace uživateli dvěma způsoby, a to tabelárně (obrázek 24) a, pro větší přehlednost, i graficky (více kapitola 3.5.8). Data zobrazená pomocí tabulky jsou řazena do čtyř sloupců (čas měření, vlhkost, teplota a napětí). Tabulka umožňuje uživateli vzestupné/sestupné řazení dle data měření. Stačí pouze stisk konkrétního nadpisu sloupce.

Čas měření	Vlhkost [N]	Teplota [C]	Napětí [N]
13:33 08.05.2014	50	-10	80
16:43 08.05.2014	40	8	30

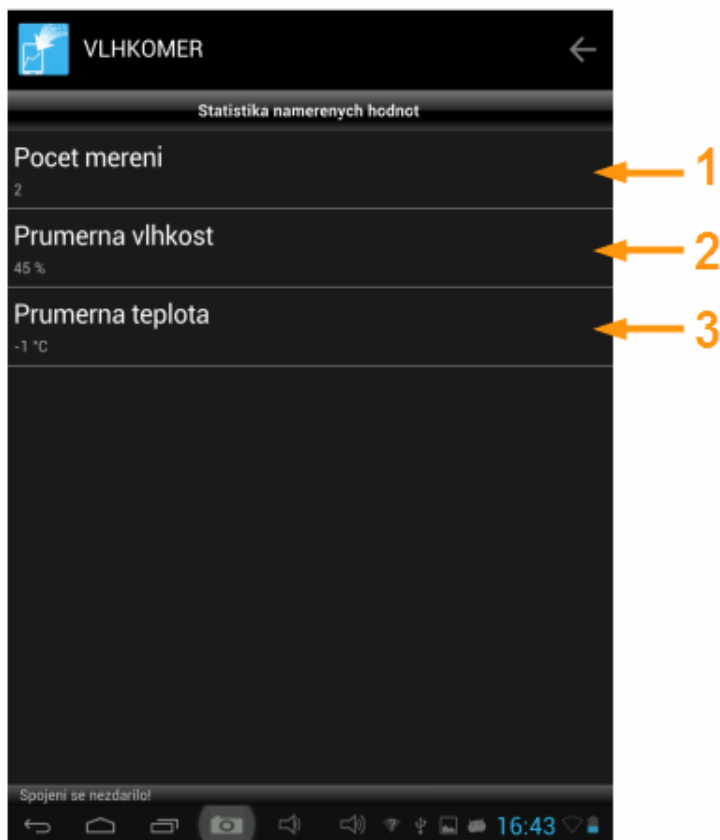
Obrázek 24 – Obrazovka – přehled naměřených hodnot

1 – názvy veličin a jednotky

2 – naměřená data

3.5.7 Statistika naměřených hodnot

Aplikace je schopna uživateli zobrazit některé statistické hodnoty. Jedná se o průměrnou hodnotu vlhkosti, teploty od prvotního započetí měření senzoru a celkový počet obdržených paketů. Pomocí těchto hodnot může uživatel například definovat polohu rosného bodu v prostoru umístění senzoru.



Obrázek 25 – Obrazovka – statistika naměřených hodnot

1 – celkový počet měřících intervalů
2 – průměrná hodnota vlhkosti

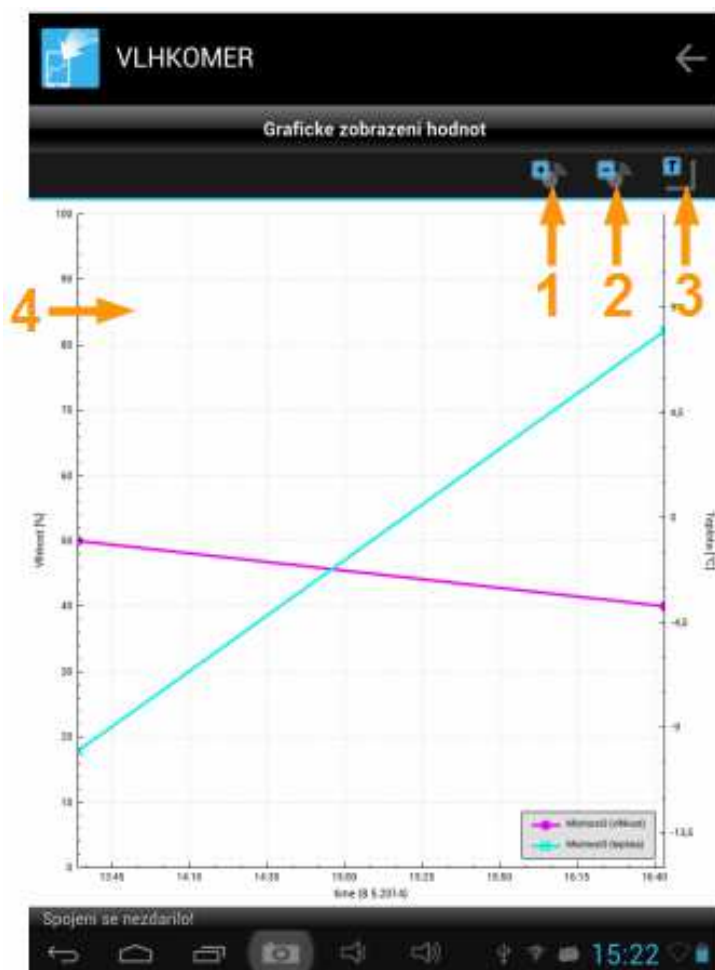
3 – průměrná hodnota teploty

3.5.8 Grafické zobrazení naměřených hodnot

Jedna z předností a zároveň jeden ze způsobů prezentace dat koncovému uživateli je grafické zobrazení dat. Jedná se o jednoduchý grafický editor, který je založen na knihovně *QCustomPlot*, která byla vytvořena pro kvalitní zobrazování 2D výkresů, grafů a tabulek. Jde o velmi kvalitní knihovnu, která je dokonce schopna vykreslovat grafy v reálném čase. Grafický výstup lze exportovat do různých typů souborů jako je například PDF, JPG, BMP a další. Knihovna je dostupná pod licencí GPL. Knihovna je primárně určena pro platformu Qt.

Na obrázku 26 je zobrazeno základní rozložení grafického editoru s ukázkou jednoduchého lineárního průběhu. V horním pravém rohu okna se nacházejí tři hlavní ovládací prvky. Jedná se o přidání/odebrání hodnot z grafu a nastavení os grafu.

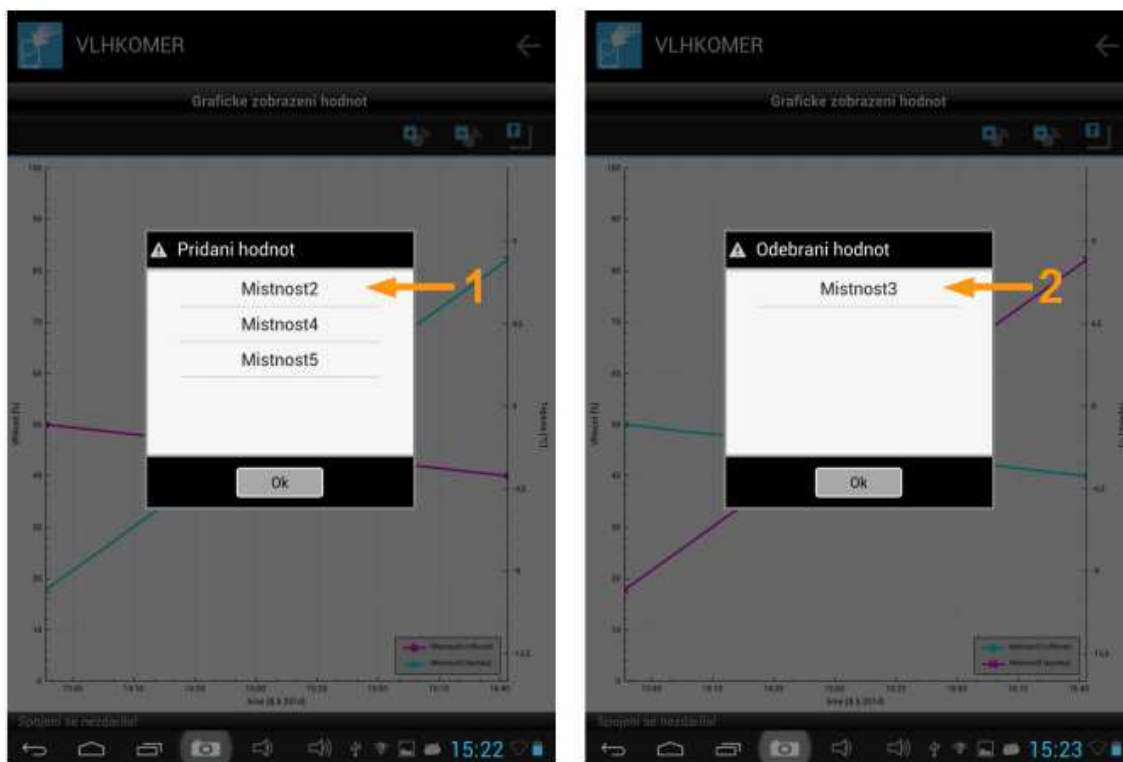
S ohledem na jednoduchou a srozumitelnou prezentaci hodnot uživatelům jsou další vlastnosti grafu přednastaveny. Do grafu může uživatel přidávat neomezený počet průběhů a hodnoty zobrazovat na hlavní a vedlejší ose y. O přehlednost zobrazovaných průběhů se stará automatické přidělování barev jednotlivým průběhům a samozřejmostí je legenda umístěná v pravém dolním rohu grafu.



Obrázek 26 – Obrazovka – grafické zobrazení hodnot

- | | |
|---------------------------------------|------------------------|
| 1 – přidání nového průběhu | 3 – změna nastavení os |
| 2 – odebrání, již zobrazeného průběhu | 4 – oblast grafu |

Přidávání a odebrání průběhů do grafu se děje pomocí dialogových oken zobrazených na obrázku 27. Pro případ přidání nového průběhu je zobrazen seznam senzorů, jejichž hodnoty lze do grafu přidat. Jediným kritériem pro zobrazení senzoru v tomto seznamu je naměření více jak jedné hodnoty teploty, vlhkosti nebo napětí. V druhém případě, odebrání průběhu z grafu (na obrázku 28 vpravo), je zobrazen seznam senzorů, které se v grafu již nacházejí. Po kliknutí na název senzoru je průběh z grafu odstraněn.

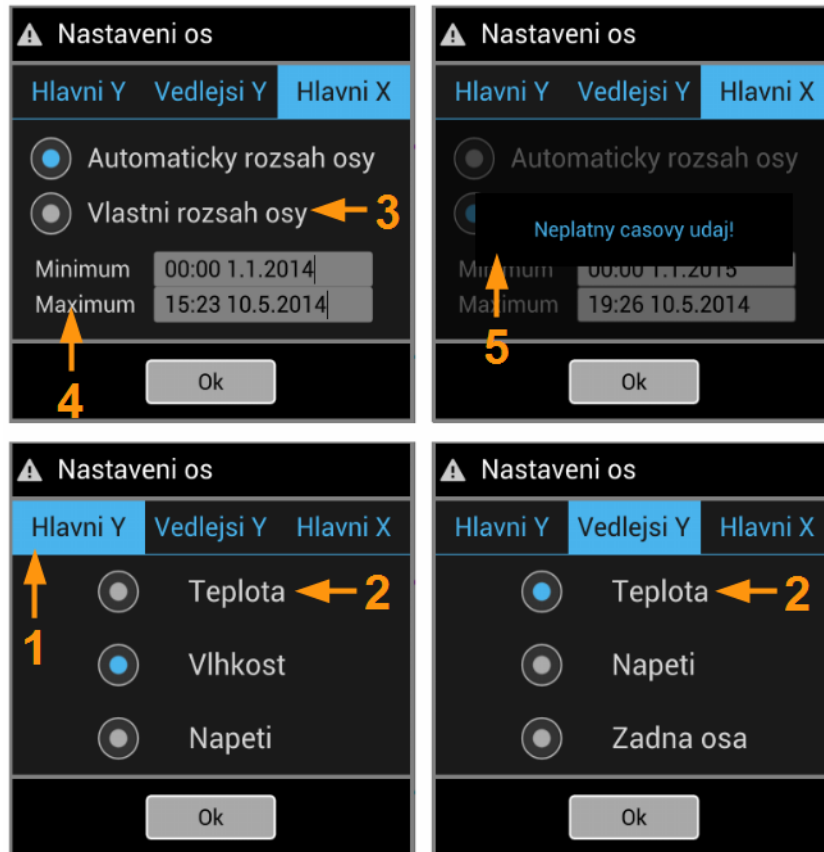


Obrázek 27 – Dialogová okna – přidání/odebrání hodnot

1 – seznam senzorů, jejichž hodnoty lze přidat 2 – seznam, již zobrazených průběhů do grafu

Změna nastavení os se provádí prostřednictvím dialogového okna, které je detailně zobrazeno na obrázku 28. V horní části tohoto okna jsou tři záložky, pomocí nichž může uživatel jednoduše listovat mezi nastavením hlavní/vedlejší osy y a hlavní osou x. V základním nastavení je definována pouze veličina vztažená k hlavní ose y, to však může uživatel jednoduše změnit. Podle nastavení hlavní osy aplikace nabídne veličinu, kterou lze zobrazit na vedlejší ose y, tím je eliminováno zobrazování stejných veličin na obou osách y. Editor podporuje pouze zobrazování hodnot pro hlavní osu x, zde má ale uživatel možnost dvou nastavení rozsahů, a to automatického, který jako minimum zvolí hodnotu, která byla změřena v nejdelším časovém horizontu a jako hodnotu maximální volí časový okamžik poslední změřené hodnoty. Pokud je v grafu umístěno více průběhů pro více senzorů, editor vždy zvolí nejstarší a nejaktuálnější časový okamžik. Druhou možností je volba vlastního rozsahu, tu uživatel musí definovat prostřednictvím textového řetězce. V důsledku toho by mohla být aplikace snadno zranitelná vůči nesprávně zadané hodnotě. Pokud uživatel zadá hodnotu v jiném tvaru, použije jiný než povolený znak (povolené znaky: jakákoli číslice, tečka, dvojtečka a mezera) nebo zamění hodnoty minimum – maximum, ohlásí aplikace uživateli chybový stav. Dialog nelze ukončit, dokud uživatel nevyplní údaje ve správném tvaru nebo dokud nepřepne na možnost automatický rozsah osy.

Ukázku formátu řetězce může uživatel vidět na předem vyplněných hodnotách (tvar řetězce: „hh:mm dd.MM.yyyy“).



Obrázek 28 – Dialogová okna – nastavení os

1 – navigační záložky

2 – možnosti zobrazovaných veličin

3 – volba typu rozsahu

4 – textová pole pro vyplnění textového řetězce

5 – varovná informace

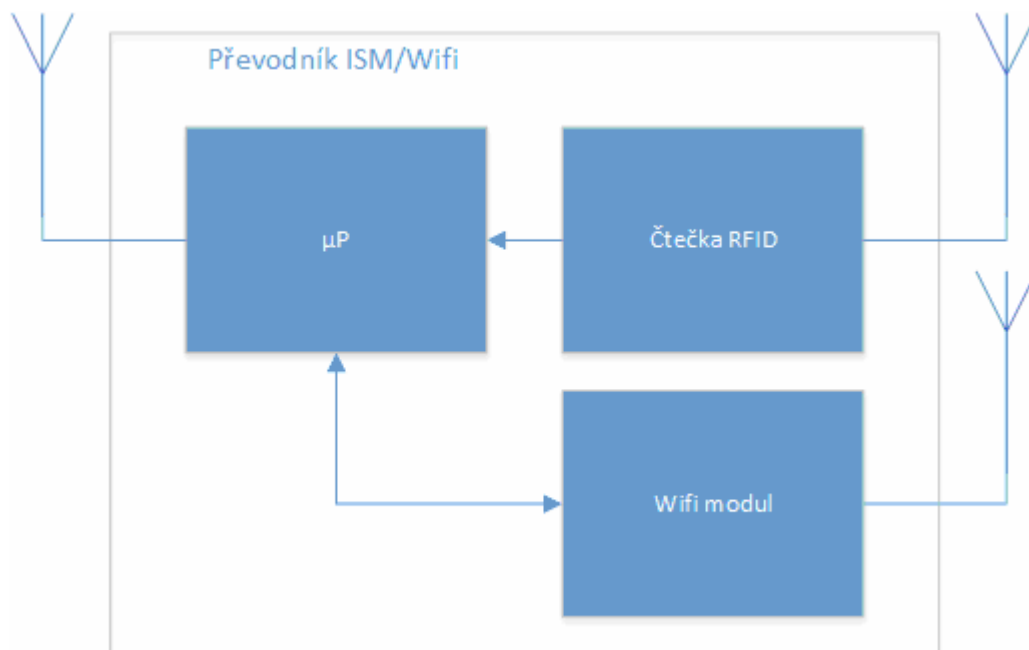
4 NÁVRH OBVODOVÉHO ŘEŠENÍ PŘEVODNÍKU ISM/WIFI

Tato kapitola je věnována návrhu obvodového řešení převodníku ISM/Wifi. V kapitole jsou shrnuty požadavky na převodník a určena jeho funkce. Dále je zde uvedeno funkční zapojení obvodu a zvoleny konkrétní komponenty.

4.1 Funkce převodníku

Převodník ISM/Wifi je nedílnou součástí systému pro měření teploty (náhled na celý systém obrázek 5). Hlavním úkolem převodníku je distribuce naměřených hodnot senzorovou sítí aplikaci pro správu a archivaci dat. Sensory mezi sebou a převodníkem komunikují pomocí protokolu SimpliciTi, což je protokol vytvořený firmou Texas Instrument. Na druhé straně aplikace vyžaduje komunikaci pomocí protokolů, které jsou dnes souhrnně označovány názvem Wifi.

Mimo tento úkol musí převodník navíc ještě plnit funkci čtečky RFID. Čtečka RFID bude využívána systémem k identifikaci rozmístěných senzorů.



Obrázek 29 – Blokové schéma převodníku

Obrázek 29 zobrazuje blokové schéma převodníku. To se skládá ze třech částí, a to mikroprocesoru, který má na starosti komunikaci se senzorovou sítí a zpracování dat z čtečky RFID. Čtečka RFID má za úkol číst výrobní číslo HF transpondéru a to následně předávat mikroprocesoru. Převodník UART/Wifi odesílá předložená data od mikroprocesoru koncové aplikaci.

4.2 Volba vhodných komponent

V této části budou zvoleny vhodné obvodové prvky s ohledem na různé specifikace, které systém pro měření teploty a vlhkosti přináší. Bude se jednat o volbu mikroprocesoru, čtečky RFID a Wifi modulu.

4.2.1 Volba mikroprocesoru

Senzorická síť systému pro měření teploty a relativní vlhkosti vzduchu je tvořena senzory s mikrokontrolery řady CC430, které v sobě integrují 1 GHz RF rádiové rozhraní. Dále pak sériová rozhraní UART (Universal Asynchronous Receiver and Transmitter), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit) a další. Tato rozhraní budou využita ke komunikaci se čtečkou RFID a wifi modulem. Pro rádiový přenos dat je dostupný komunikační protokol SimpliciTI, tento protokol je určen pro použití v bez licenčním pásmu ISM (The Industrial, Scientific and Medical radio bands). Tyto mikrokontrolery jsou díky svým výborným vlastnostem vhodné pro použití jak pro satelitní tak i centrální jednotku. Mikroprocesor je pouze nastaven do role přístupového bodu, opakovače nebo koncového bodu.

V tomto případě je tedy volba mikrokontroleru velmi jednoduchá a spíše již předurčená s ohledem na jejich využití u měřicích senzorů.

4.2.2 Volba obvodu čtečky RFID

Jedním z požadavků na převodník ISM/Wifi je možnost identifikování jednotlivých senzorů. K tomuto účelu bude využita, dnes široce rozšířená technologie, rádio frekvenční identifikace – RFID. Jednotlivé senzory budou opatřeny transpondéry a převodník čtečkou RFID.

Než přistoupíme k samotné volbě konkrétního obvodu čtečky, musíme nejprve určit pásmo, v kterém naše aplikace bude pracovat. Volba frekvence je u systémů využívající RFID klíčová, určuje hned několik omezení. Mezi ně patří například dosah čtení, rychlost čtení informace, použitelnost v různých prostředích, apod. Pro systémy RFID existují čtyři pracovní pásma: LF pásmo 125 – 134 kHz, HF pásmo 13,56 MHz, UHF 860 – 960 MHz a MW 2,4 – 5,8 GHz. [3]

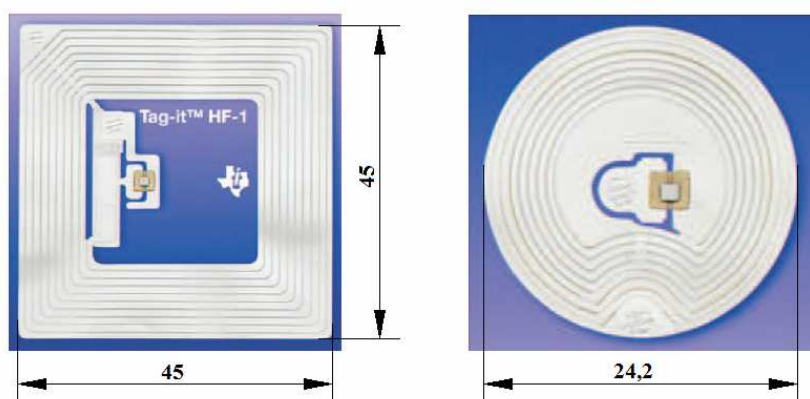
Pásmo **low frequency** se vyznačuje poměrně malým dosahem čtení, méně než 0,5m, a malou rychlostí čtení. Mezi výhody tohoto pásma patří možnost snímání přes kapalinu. [3]

Pásmo **high frequency** má čtecí dosah do 1 m a dostatečnou rychlost čtení. Čtení přes kapalinu je horší než u pásma low frequency. [3]

Pásmo **ultra high frequency** disponuje čtecím rozsahem do jednotek metrů, čtecí rychlost je vyšší než u předcházejících pásem. Nevýhodou tohoto pásma je nemožnost čtení přes kapalinu. [3]

Pásmo **microwave** pracuje v blízkosti pásma využívajícím wifi. Mezi jeho výhody patří vysoká přenosová rychlost a čtecí vzdálenost, často se využívá v kombinaci s aktivními transpondéry pro zvýšení čtecí vzdálenosti až na desítky metrů. Nevýhodou tohoto pásma je snižující se výkon v přítomnosti kovů a kapalin. [3]

Pro naši aplikaci bych volil pásmo HF (High Frequency) s ohledem na uspokojivou čtecí vzdálenost. Čtecí vzdálenosti u ostatních pásem, které se nacházejí v řádu jednotek, u některých dokonce desítek metrů, jsou pro náš systém nepotřebná ba nepraktická. Když uvážíme, že senzory budou umístěny například v různých pokojích bytu, mohla by nastat situace, kdy jsou senzory od sebe vzdáleny méně než 10m. V tomto případě by mohla nastat identifikace obou senzorů najednou a uživatel by nebyl schopen určit, o jaký senzor se jedná. V tomto pohledu by byla nejpříjemnější volba LF (Low Frequency) pásma. LF pásmo je však pro náš systém nepoužitelné, protože se snižující se frekvencí se zvětšuje anténa transpondéru a tím i jeho rozměr (velikost senzoru do 5 cm). Dle mého názoru je vhodným kompromisem volit pásmo HF.



Obrázek 30 – RFID transpondéry

Na obrázku 30 (velikosti uvedené v obr. jsou v milimetrech) jsou zobrazeny transpondéry určené pro HF pásmo. HF pasivní transpondér se skládá z antény, která je tvořena buď měděnou cestou, nebo častěji vodivým tiskem a čipem, který uchovává informaci. RF transpondéry jsou dvojího typu – aktivní a pasivní. Aktivními jsou označovány transpondéry, které mají vlastní zdroj energie a často je do nich možné zapisovat určité informace. Pro náš systém budou stačit pasivní transpondéry, tedy ty bez vlastního zdroje energie a možností pouze čtení informace. Touto volbou ušetříme i nemalé finanční náklady.

Po volbě technologie následuje volba konkrétního obvodu. S ohledem na zvolenou technologii a již dříve zvolený mikroprocesor jsme se orientovali na výrobky firmy Texas Instruments, která nabízí obvody řady RTF796x. Ty jsou určeny pro zvolené pásmo HF. Nespornou výhodou je předurčené použití daného obvodu s mikroprocesory MSP430. Z porovnání cen s cenami obvodů konkurenčních firem vychází, že obvody jsou ve stejné cenové kategorii a spíše levnější.

4.2.3 Volba Wifi modulu

Jedinými požadavky na volbu obvodu, který bude přijímat/odesílat data protokoly souhrnně označované jako wifi, jsou jednoduchá integrace do zapojení převodníku, nízká spotřeba a malé rozměry. S ohledem na tyto požadavky byl zvolen WLAN modul RN-131, který vyniká především svou jednoduchou aplikací do různých obvodů.

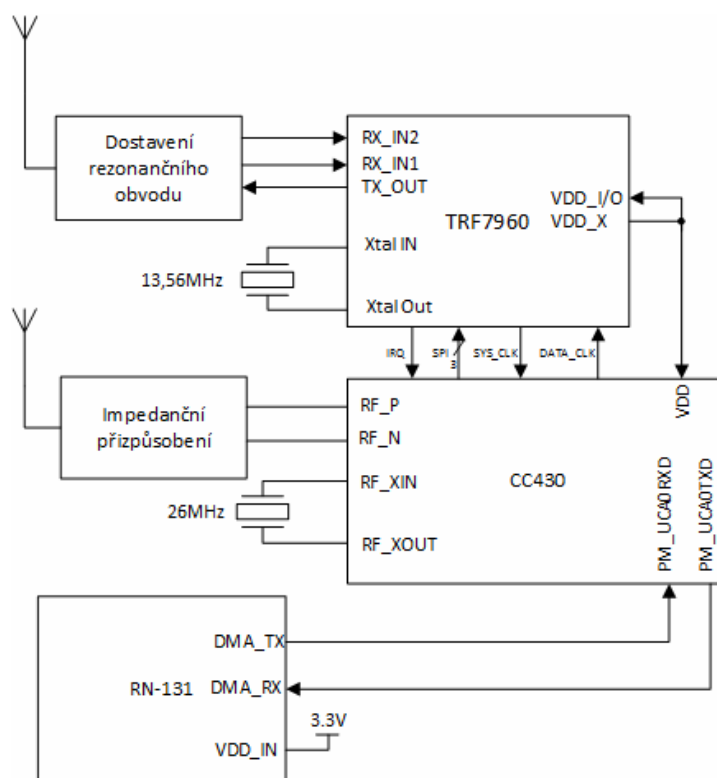
Wifi modul obsahuje 2,4 GHz vysílač/přijmač, procesor, TCP/IP stack, hodiny reálného času, správu napájení. RN-131 je díky svým minimálním rozměrům a extrémně nízkou spotřebou předurčen k použití v přenosných zařízeních. V modulu je přeinstalovaný software, který usnadňuje jeho integraci do obvodu. V nejjednodušším zapojení lze připojit k modulu pouze signály RX, TX a napájení. Když tento modul porovnáme s konkurencí a vezmeme v potaz všechny jeho výhody, tak je i jeho cena výhodná.

5 OBVODOVÉ ŘEŠENÍ PŘEVODNÍKU ISM/WIFI

Tato kapitola pojednává o realizaci převodníku na základě předešlého návrhu. V kapitole je obsažen popis obvodového řešení převodníku, popis zařízení a ovládacích prvků, nastavení wifi modulu a softwaru mikrokontroléru.

5.1 Obvodové řešení

Návrh obvodového řešení převodníku je zobrazen na obrázku 31. Obvodové zapojení obsahuje námi zvolené prvky a jejich zapojení se řídí dle doporučení výrobce.



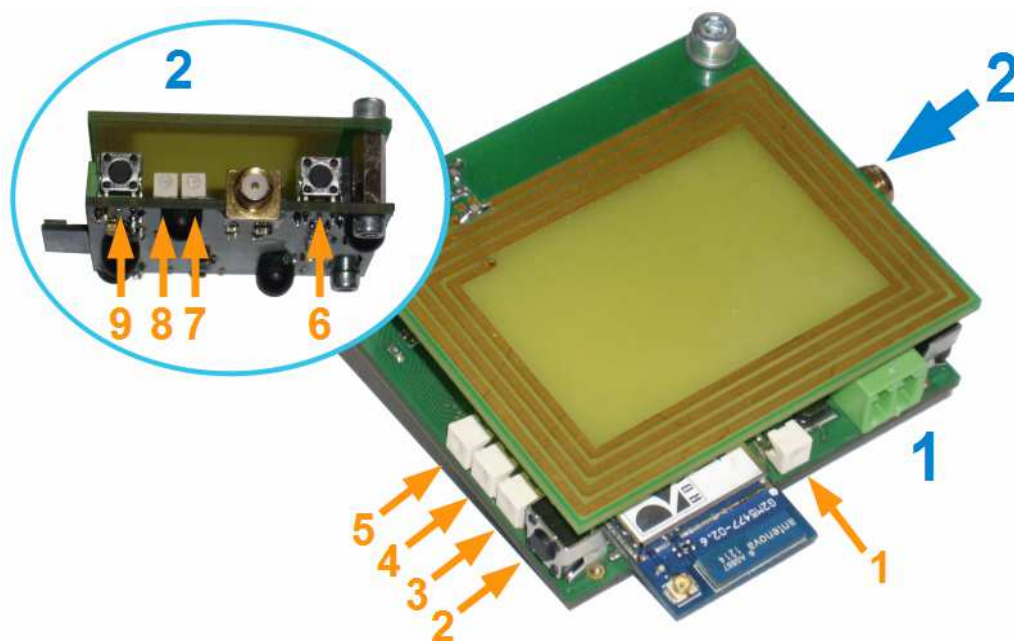
Obrázek 31 – Obvodové řešení převodníku

Mikrokontrolery z rodiny CC430 jsou založeny na architektuře RISC (Reduced instruction set computing). Z hlediska zapouzdření obvodu se jedná o SoC (Systém on Chip) a integruje v sobě radiové RF rozhraní 1 GHz, pomocí něhož převodník komunikuje s koncovými zařízeními (měřícími senzory). Radiové rozhraní MCU je realizováno obvodem CC1101. Tento obvod může pracovat hned s několika nosnými frekvencemi. Na které frekvenci bude obvod pracovat, rozhoduje zapojení několika externích pasivních součástek a nastavení řídicích registrů. Naše aplikace vyžaduje pracovní frekvenci 868 MHz. Přizpůsobení antény je tvořeno rezonančním obvodem,

který nastavuje nosnou referenční frekvenci (tento obvod tvoří L1, L2, L3, L4, C23, C25 a C26), a LC filtrem. LC filtr odstraňuje zkreslení nosné frekvence. Ve schématu je dále naznačeno zapojení krystalového oscilátoru 26MHz, který generuje referenční frekvenci pro syntezátor, hodinový signál pro převodník ADC a digitální část.

Převodník ISM/Wifi obsahuje čtečku RFID, ta je realizována obvodem TRF796, jak je znázorněno na obrázku 22. Ke spojení obvodu RFID čtečky a mikrokontroleru můžeme využít dvou typů rozhraní, sériové nebo paralelní. V našem zapojení byl zvolen sériový typ komunikace, s využitím komunikačního rozhraní SPI. Obvod má dva přijímací vstupy RX_IN1 a RX_IN2. Oba vstupy jsou připojeny na externí filtr, aby bylo zajištěno, že vždy alespoň jeden obdrží modulovaný signál od RF transpondéru. Externí filtr zajišťuje fázový posuv 45° vstupu RX_IN2. Obvod dále obsahuje zapojení krystalového oscilátoru 13,56MHz, který přímo generuje RF signál do výstupního stupně, dále generuje hodinový signál pro digitální části obvodu a hodinový signál SYS_clk, který může být použit pro externí MCU (Microcontroller).

5.2 Popis ovládacích a signalizačních prvků



Obrázek 32 – Převodník ISM/Wifi

- | | |
|---|--|
| 1 – LED1 – signalizace připojení napájení | 6 – rezervováno (v současné verzi nevyužito) |
| 2 – TL1 – ovládání wifi modulu | 7 – rezervováno (v současné verzi nevyužito) |
| 3 – LED3 – signalizace stavu připojení | 8 – signalizace činnosti čtečky RFID |
| 4 – LED4 – signalizace přenosu dat | 9 – rezervováno (v současné verzi nevyužito) |
| 5 – LED5 – signalizace stavu přidružení k bezdrátové síti | |

Tato podkapitola je věnována popisu jednotlivých ovládacích a signalizačních prvků, kterými převodník ISM/Wifi disponuje. Převodník je zobrazen na obrázku 32, zde je převodník zobrazen ze dvou pohledů, tak aby byly přehledně znázorněny všechny ovládací a signalizační prvky.

Pohled 1 je zaměřen na detail wifi modulu, který v realizovaném obvodu zastává funkci TCP serveru. V blízkosti modulu se nacházejí led diody LED3 až LED5 a tlačítko TL1. Tyto diody signalizují status připojení a signalizaci obdržených dat. LED3 (zelená) signalizuje stav TCP/IP připojení, pokud je led ve stavu jedna, jedná se o aktivní připojení TCP klienta (aplikace) k TCP serveru, rychlé blikání této led značí chybnou indikaci ip adresy, pomalé blikání led značí správně indikovanou ip adresu a zároveň nepřipojení žádného TCP klienta (v našem případě aplikace) k převodníku. LED4 (zelená) značí přenos dat TCP klient – server. LED5 (červená) indikuje stav přidružení k bezdrátové síti, pokud je modul přidružen k síti, led se nachází ve stavu jedna, v opačném případě je připojení v pořádku. Tlačítko TL1 slouží k restartování modulu (aktivní stav 0), pokud je tlačítko stlačeno (po dobu 160 μ s) při připojení napájení, modul se přepne do nastavovacího režimu. V případě, že je tlačítko stlačeno po dobu déle než 10s, je modul uveden to továrního nastavení. LED1 indikuje připojení napájení.

5.3 Popis softwaru

Program převodníku ve své nejvyšší úrovni spravuje tři základní úlohy:

- přenos dat mezi senzory a aplikací (převod ISM/Wifi),
- připojování nových klientů (senzorů),
- obsluha a přenos dat z čtečky RFID.

Obsluha prvních dvou úloh je asynchronně řízena z nižších programových vrstev, které jsou implementovány přímo společností Texas Instruments a v projektu jsou využívány jako zásobník protokolu SimpliCI. Tyto programové vrstvy jsou uživatelsky parametrizovány (vývojářem programového vybavení, který těchto vrstev využívá). Mezi parametry, které lze definovat, patří, frekvence komunikačního kanálu, typ síťové komponenty, maximální počet spojení, možnost šifrování dat. Podrobněji se této problematice věnuje kapitola, která se zabývá komunikačním protokolem SimpliCI. Kompletnost a neporušenost dat při přenosu je kontrolována v několika úrovních. Na aplikační vrstvě je kompletnost a neporušenost dat potvrzována bajtem s kontrolním součtem. Na nižších vrstvách jsou aplikovány další kontrolní mechanismy.

Obsluha čtečky RFID se provádí každých 10 ms, čtečka je zapnuta a je kontrolována přítomnost RFID transpondéru, pokud je v její blízkosti je načteno ID a následně předáno je předáno v paketu přes UART wifi modulu a dále distribuováno aplikaci.

Přenos dat z čtečky RFID je následující: poté co čtečka RFID obdrží signál od RFID transpondéru převede bitový dekodér analogový signál na bitové slovo, které je následně uloženo do FIFO registru, ten disponuje velikostí 12 bajtů. Zde jsou dekódovaná data připravena na čtení externím MCU. Pokud je zjištěna jakákoli chyba načtených dat – formát, parita, CRC – je externí systém upozorněn prostřednictvím

přerušení. Informace o typu přerušení je dostupná v IRQ registru. [10]

5.3.1 Nastavení Wifi modulu

Wifi modul RN-131 je schopen samostatně realizovat bezdrátové připojení LAN sítě. Obsahuje TCP/IP stack (kompletní nastavení síťových protokolů) a aplikaci, která uživateli usnadňuje nastavení modulu. Při správném nastavení wifi modulu se po jeho zapnutí automaticky připojí do wlan sítě a odesílá/přijímá sériová data.

Nastavení modulu probíhá v nastavovacím režimu, kde pomocí klíčových slov měníme jednotlivé parametry. Modul může pracovat jako přístupový bod s malou bezdrátovou sítí určenou pro zařízení s operačním systémem android nebo se modul může připojit již ke stávající síti. V našem případě s výhodou využijeme druhé varianty. Před prvním připojením ke konkrétní wlan síti musejí být nastaveny přihlašovací údaje do této sítě a následné nastavení automatického přihlašování modulu do nastavené sítě po restartování modulu. Komunikace modul – mikrokontroler MSP430 probíhá pomocí sériového rozhraní, z tohoto důvodu je nutné nastavit i jeho parametry (například přenosovou rychlost). Potřebná nastavení získaná z [15]:

```
$$$ //aktivování příkazového řádku
set wlan join 1 //nastavení automatického přihlášení po
                spuštění k danému ssid
set wlan ssid nazev //nastavení ssid
set wlan phrase heslo //nastavení hesla (WPA a WPA2)
set wlan key heslo //nastavení hesla (WEP)
set ip dhcp 1 //automatické přidělení ip adresy dhcp
                serveru přístupového bodu ke kterému se
                modul připojuje
set uart baud 115200 //nastavení rychlosti čtení/zápisu dat
set uart tx 1 //povolení portu tx
save //uložení nastavení
reboot //restart modulu
```

6 ZÁVĚR

Diplomová práce se zabývá návrhem a realizací dvou částí systému pro měření relativní vlhkosti a teploty vzduchu uvnitř měřených objektů. Systém se skládá ze senzoričké sítě, převodníku ISM/Wifi a aplikace pro správu, vyhodnocování a archivaci naměřených hodnot. Tato práce se blíže věnuje návrhu a realizaci převodníku ISM/Wifi a aplikace pro správu, vyhodnocování a archivaci naměřených hodnot.

Práce je členěna do šesti oddílů. První část práce je věnována teoretickému rozboru využitých informačních technologií a jejich studiu. Druhá část popisuje postup návrhu aplikace. Programová implementace návrhu programu s popisem výsledné aplikace je obsažena ve třetí části práce. Čtvrtá část se zaměřuje na návrh obvodového řešení převodníku ISM/Wifi. Následující kapitola, tedy pátá, je věnována realizaci převodníku ISM/Wifi. Poslední část hodnotí dosažené výsledky.

V teoretické části byly popsány informační technologie, které systém využívá nebo je pomocí nich tvořen. Kapitola obsahuje informace o grafické knihovně Qt, programovacím jazyku C++, komunikačním protokolu SimpliCI, frekvenčním pásmu ISM a o síťových protokolech souhrnně označovaných jako Wifi. Dále zde bylo zvoleno vhodné vývojové prostředí a síťový komunikační protokol.

Návrh aplikace pro správu, vyhodnocování a archivaci naměřených hodnot byl realizován pomocí modelovacího jazyka UML. S ohledem na doporučené postupy návrhu objektového systému uvedené v [8] byly v této části nejprve stanoveny požadavky na výslednou aplikaci. Tyto požadavky byly předlohou pro tvorbu diagramu případu užití a diagramu tříd.

Programová implementace aplikace byla realizována dle navržených UML diagramů s využitím vývojového prostředí Qt Creator. Výsledná aplikace splňuje stanovené požadavky a je plně funkční. Výsledné uživatelské prostředí je přehledné a uživatelsky přívětivé – toho bylo docíleno pomocí doporučení uvedených v [13].

V části návrhu převodníku ISM/Wifi byly stanoveny funkční požadavky a zvoleny vhodné komponenty k realizaci převodníku, kapitola uvádí zdůvodnění výběru těchto komponent.

V navazující kapitole – Obvodové řešení převodníku ISM/Wifi – je uvedeno realizované obvodové řešení převodníku, popis jednotlivých ovládacích a signalizačních prvků a popis programového vybavení převodníku.

Výstupem této práce je realizovaná aplikace určená pro zařízení s operačním systémem Android a převodník ISM/Wifi. Realizace systému pro měření relativní vlhkosti a teploty vzduchu se řídila uceleným návrhem, který byl vytvořen za použití modelovacího jazyka UML. Diplomová práce obsahuje všechny části vývoje nového zařízení od návrhu až po konečné testování. V rámci práce bylo využito nových informačních technologií, jako je například deklarativní jazyk QML.

LITERATURA

- [1] *Serial C++* [online]. Programujte.com, 2005 [cit. 2013-11-08]. Dostupné z: programujte.com/clanky/39-serial-c/
- [2] SOMMEROVÁ, Martina. Základy RFID technologií. [online]. s. 31 [cit. 2013-11-27]. Dostupné z: rfid.vsb.cz/miranda2/export/sites-root/rfid/cs/okruhy/informace/RFID_pro_Logistickou_akademii.pdf
- [3] WATZKE, David. Grafické programy v Qt 4. [online]. [cit. 2013-12-02]. Dostupné z: www.abclinuxu.cz/serialy/qt-4-psani-graficky-programu
- [4] *Qt Reference Documentation* [online]. Qt-project.org [cit. 2013-12-02]. Dostupné z: doc.qt.digia.com/qt/
- [5] *SimpliciTI* [online]. Texas Instruments [cit. 2013-12-02]. Dostupné z: www.ti.com/simpliciti
- [6] KOLKA, Zdeněk. *Počítačové a komunikační sítě*. Brno, 2007.
- [7] REJNKOVÁ, Petra. Příklady použití diagramů UML 2.0. [online]. [cit. 2013-12-03]. Dostupné z: uml.czweb.org/index.html
- [8] ARLOW a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací*. Brno: Computer Press, a.s., 2007. ISBN 978-80-251-1503-9.
- [9] TEXAS INSTRUMENTS. *MSP430 SoC with RF Core*. USA, 2009. Dostupné z: www.ti.com/lit/ds/symlink/cc430f5137.pdf
- [10] *MULTI-STANDARD FULLY INTEGRATED 13.56-MHZ RFID ANALOG FRONT END AND DATA-FRAMING READER SYSTEM*. USA, 2006. Dostupné z: www.ti.com/lit/ds/slou186f/slou186f.pdf
- [11] BLANCHETTE, Jasmin a Mark SUMMERFIELD. *C++ GUI programming with Qt 4*. USA: Trolltech AS, 2006. ISBN 0-13-187249-4.
- [12] ŠAJBOCH, Antonín. *Rozšíření informačního systému EPI, s.r.o. na mobilní platformu Android*. Trenčín, 2012. Bakalářská práce. Trenčianska univerzita Alexandra Dubčeka v Trenčíně. Vedoucí práce Ing. Jindřich Petrucha, PhD.
- [13] *Android: Developer Resources* [online]. 2014 [cit. 2014-03-17]. Dostupné z: www.android.com/
- [14] EICHHAMMER, Emanuel. Qt Plotting Widget QCustomPlot. [online]. [cit. 2014-05-11]. Dostupné z: www.qcustomplot.com/
- [15] ROVINGnetworks. *WiFly Command Reference, Advanced Features & Applications User's Guide*. USA, 2013. Dostupné z: ww1.microchip.com/downloads/en/DeviceDoc/rn-wiflycr-ug-v1.2r.pdf
- [16] Všeobecné oprávnění č. VO-R/10/08.2005-24 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu. In: www.ctu.cz/1/download/Opatreni%20obecne%20povahy/VO_R_10_08_2005_24.pdf. 2002.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

CSV	Comma Separated Value, ukládání dat do řetězce kde jsou jednotlivá data rozdělována čárkou
IDE	Integrated Development Environment, vývojové prostředí pro usnadnění vývoje aplikací
SoC	System on a Chip, integrovaný obvod, který integruje na jednom čipu více elektronických systémů
MCU	Microcontroller, jednočipový počítač
GUI	Graphical User Interface, je grafické uživatelské prostředí, které umožňuje uživateli ovládat určitý systém pomocí grafických prvků
TCP/IP	Transmission Control Protocol/Internet Protocol, soubor protokolů, které definují komunikaci v počítačových sítích
TCP	Transmission Control Protokol, jeden z hlavních protokolů Internetu, konkrétně protokol transportní vrstvy
UDP	User Datagram Protocol, jeden z hlavních protokolů Internetu, konkrétně protokol transportní vrstvy
UML	Unified Modeling Language, grafický jazyk pro navrhování programových systémů
UART	Universal Asynchronous Receiver and Transmitter, jedná se o rozhraní sériové komunikace
LF	Low Frequency, pásmo dlouhých elektromagnetických vln
HF	High Frequency, pásmo krátkých elektromagnetických vln
SPI	Serial Peripheral Interface, sériové komunikační rozhraní
I2C	Inter-Integrated Circuit, multi-masterová počítačová sériová sběrnice
RFID	Radio Frequency Identification, bezkontaktní identifikace různých předmětů
ISM	Industrial Scientific and Medical, jsou frekvenční pásma pro rádiové vysílání v oborech průmyslu, vědy a zdravotnictví
GPL	General Public Licence, licence pro svobodný software, definuje za jakých podmínek se software může volně šířit
LGPL	Leader General Public License, licence pro svobodný software, definuje za jakých podmínek se software může volně šířit
QML	Qt Meta Language, deklarativní jazyk pro tvorbu uživatelského rozhraní aplikací, využíváný výhradně pro mobilní aplikace
RISC	Reduced Instruction Set Computing, redukováná instruktážní sada

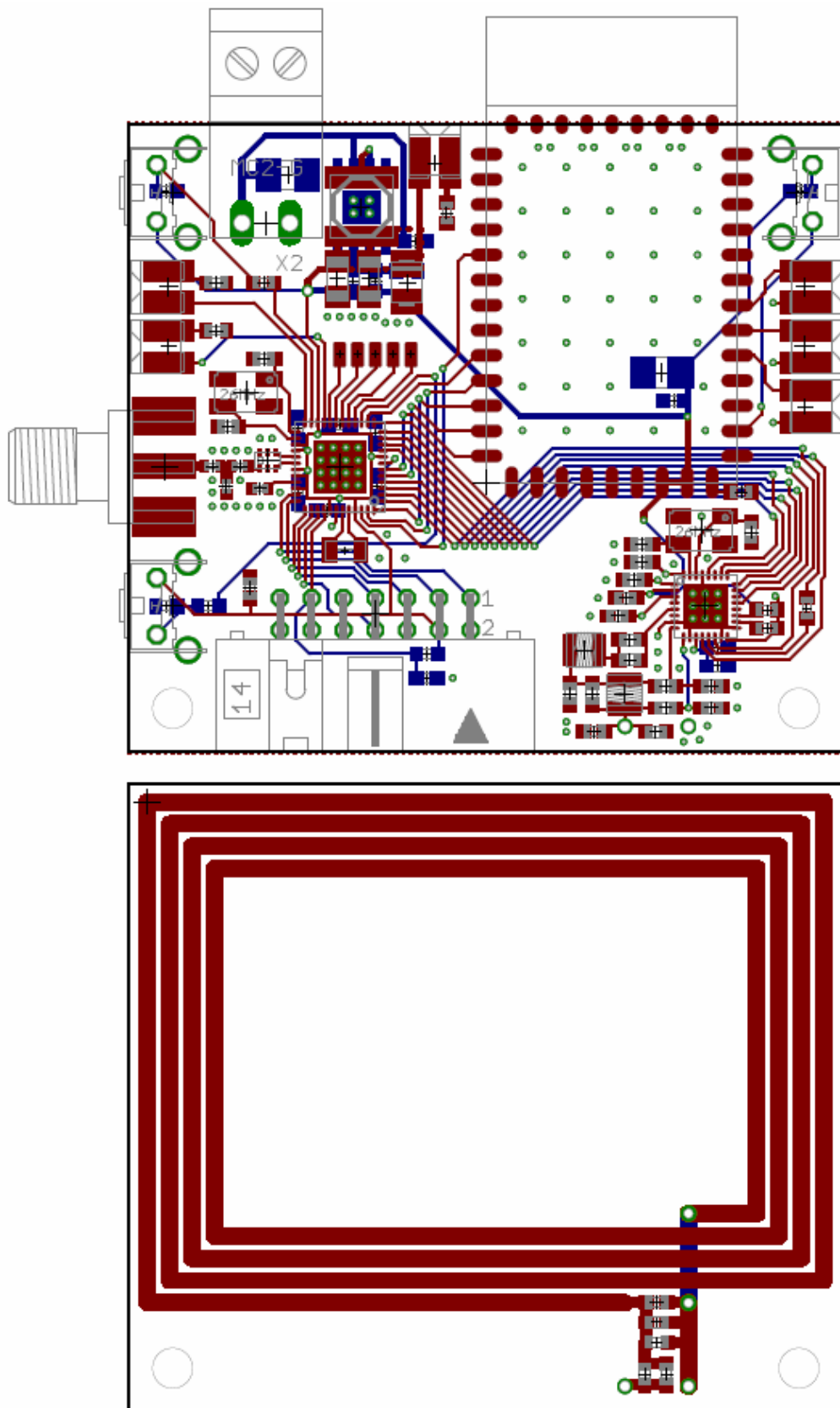
SEZNAM PŘÍLOH

A	Obvodové řešení převodníku ISM/Wifi	50
B	Deska plošných spojů	51
C	Komunikační protokol	52

A OBVODOVÉ ŘEŠENÍ PŘEVODNÍKU ISM/WIFI

Obvodové řešení převodníku je pro svou prostorovou náročnost umístěno ve volných přílohách.

B DESKA PLOŠNÝCH SPOJŮ



C KOMUNIKAČNÍ PROTOKOL

Senzor	Převodník	Aplikace
rozebrání paketu	← 0 1 2 3 4 5 6 LNG CMD PRD0 PRD1 CNT0 CNT1 CRC	Příkaz 0 1 2 3 4 5 6 7 8 9 10 LNG CMD ID0 ID1 ID2 ID3 PRD0 PRD1 CNT0 CNT1 CRC
Naměřená data	→ 0 1 2 3 4 5 6 7 LNG CMD HUM T0 T1 V0 V1 CRC	rozebrání paketu
Data z flash	→ 0 1 2 3 4 5 6 7 LNG CMD NO0 NO1 HUM T0 T1 CRC	rozebrání paketu
Informace o flash	→ 0 1 2 3 4 5 6 7 8 LNG CMD PRD0 PRD1 CNT0 CNT1 CRC	rozebrání paketu
přihlášení do sítě	→ 0 1 2 3 4 5 6 7 8 LNG CMD ID0 ID1 ID2 ID3 RSSI LQI NO0 NO1 HUM T0 T1 CRC	0 1 2 3 4 5 6 7 8 9 10 LNG CMD ID0 ID1 ID2 ID3 PRD0 PRD1 CNT0 CNT1 CRC
0 1 2 LNG CMD CRC	Identifikátor RFID 0 1 2 3 4 5 6 7 8 LNG CMD ID0 ID1 ID2 ID3 RSSI LQI CRC	rozebrání paketu
	→ 0 1 2 3 4 5 6 LNG CMD ID0 ID1 ID2 ID3 CRC	rozebrání paketu

Hodnoty bajtu CMD:

- 0x00 žádost o záznam
- 0x01 žádost o stav flash
- 0xE0 start wireless_mode
- 0xF0 start flash_mode
- 0xAA přihlášení do sítě
- 0xCC adresa RFID
- 0xBB záznam z flash
- 0xDD naměřená data
- 0xFF smazat záznamy na flash

Hodnoty bajtu LNG

Délka paketu - tak jak je přijat.

Hodnoty bajtů IDX

Adresa senzoru (32 bitů)

Hodnoty bajtů NOx

Číslo záznamu

Hodnoty bajtů PRDx

Perioda měření v sekundách

Hodnoty bajtů CNTx

Pořadí záznamu ve flash, který je požadován

Hodnoty bajtu HUM

Relativní vlhkost vzduchu v procentech

Hodnota bajtu Tx

Teplota v desetinných stupně Kvalita spojení

Hodnota bajtu Vx

Napětí v milivoltech

Hodnota bajtu RSSI

Síla přijatého signálu

Hodnota bajtu LQ

Součet všech bajtů paketu vyjma CRC / 0xFF