

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra mikroelektroniky



**Hlavní jednotka pro sběr informací
ze ZigBee senzorické sítě**

květen 2014

Bakalant: Tadeáš Holler
Vedoucí práce: Ing. Vladimír Janíček, Ph.D

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra mikroelektroniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **HOLLER Tadeáš**

Studijní program: Komunikace, multimédia a elektronika
Obor: Aplikovaná elektronika

Název tématu: **Hlavní jednotka pro sběr informací ze ZigBee senzorické sítě**

Pokyny pro vypracování:

- 1) Prostudujte literaturu týkající se sběru dat ze senzorových sítí.
- 2) Navrhněte obvodové zapojení hlavní jednotky pro sběr dat ze senzorické sítě..
- 3) Předpokládejte komunikaci pomocí ZigBee protokolu.
- 4) Naprogramujte řídicí jednotku a vytvořte podpůrný program pro PC.
- 5) Zhotovte prototyp a ověřte jeho funkci.
- 6) Vyhodnoťte výsledky z ověřovacího měření.
- 7) Vytvořte obslužný manuál.

Seznam odborné literatury:

- 1)Gislason D. - Zigbee wireless networking, ISBN 780750685979
- 2)Farahani S, - ZigBee wireless networks and transceivers ISBN 9780750683937
- 3)Kainka B. - USB:měření, řízení a regulace pomocí sběrnice USB, ISBN 8073000733
- 4)Matoušek D. - USB prakticky s obvody FTDI (1. díl), ISBN 8073001039

Vedoucí: **Ing. Vladimír Janíček, Ph.D.**

Platnost zadání: 31. 8. 2015

L.S.

Prof. Ing. Miroslav Husák, CSc.
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 23. 1. 2014

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a konzultanta a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum 23. 5. 2014

.....

podpis bakalanta

Anotace:

Tato bakalářská práce se zabývá návrhem hlavní jednotky pro sběr dat ze ZigBee senzorické sítě, výběrem vhodných komponentů pro její realizaci, a dále vytvoření pravidel pro komunikaci v senzorické síti, vytvořením obslužného manuálu pro vyvinutý program a následné změření pokrytí s ohledem na použitelnost v praxi a porovnání s komerčními produkty

Abstract:

This bachelor thesis deals with the design main unit for data capture from ZigBee Based sensors net, selecting optional components for its implementation. Furthermore create a rules for communication sensors net, to crate utility manual for to develop program and subsequently measuring coverage with regarding to availability in practice and comparison with commercial products.

Poděkování

Rád bych poděkoval Ing. Vladimíru Janíčkoví Ph.D., za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce.

Seznam použitých zkratk

μC.....	Mikrokontrolér
ACK.....	Acknowledgement
AES.....	Advanced Encryption Standard
ALU.....	Arithmetic Logic Unit
APDU.....	Aplikační Protokol Datových Jednotek
APL.....	Application Layers
APS.....	Aplikační Podvrstva
CCA.....	Clear Channel Assesment
CDC.....	Communications Device Class
CISC.....	Complex Instruction Set Computer
CRC.....	Cyclic Redundancy Check
CSMA/CA.....	Carrier Sense Multiple Access with Collision Avoidance
DSSS.....	Direct Sequence Spread Spectrum
EOP.....	End of Packet
FFD.....	Full Function Device
H.....	Hight
HID.....	Human Interface Device
I/O.....	Input/Output
ID.....	Identification
IEEE.....	Institute of Electrical and Electronics Engineers
INT.....	Interrupt
ISM.....	Industrial, Scientific and Medical
L.....	Low
LED.....	Light-Emitting Diode
MAC.....	Media Access Control
NAK.....	Negative Acknowledgment
NIB.....	NWK Information Base
NLDE.....	Network Layer Data Entity)
NLME.....	Network Layer Management Entity
NRZI.....	Non Return To Zero Inverted
NWK.....	Network layers

O-QPSK.....Offset Quadrature Phase-Shift Keying
PAN.....Personal Area Network
PC.....Personal Computer
PHY.....Physical Layer
PID.....Process Identifier
RAM.....Random Access Memory
RFD.....Reduced Function Device
RISC.....Reduced Instruction Set Computing
ROM.....Read-Only Memory
RTTC.....Real-Time Clock and Calendar
SAP.....Service Access Point
SD.....Secure Digital
SPI.....Serial Peripheral Interface
USART.....Universal Synchronous/Asynchronous Receiver and Transmitter
USB.....Universal Serial Bus
VID.....Vendor Identifier
WIFI.....Wireless Fidelity
ZDO.....ZigBee Device Object

Obsah

1	Úvod	1
2	Průzkum trhu	2
3	Struktura stanice	4
3.1	Napájecí část	4
3.2	Vysílací a přijímací část	4
3.3	Sběrníková část	4
3.4	Řídící část	4
4	Teoretická část	5
4.1	Mikrokontrolér	5
4.1.1	Základní struktura	5
4.1.2	Von Neumannova architektura	6
4.1.3	Harvardská architektura	6
4.1.4	RICS a CISC	7
4.2	USB	7
4.2.1	Základní vlastnosti USB	7
4.2.2	Detekce rychlosti	8
4.2.3	Možnosti napájení	8
4.2.4	Napěťové úrovně	9
4.2.5	Kódování	9
4.2.6	Synchronizace	9
4.2.7	Přenos dat	10
4.2.8	Pakety	10
4.2.9	Token paket	11
4.2.10	Datový paket	12
4.2.11	SOF paket	13
4.2.12	Handshake paket	13
4.2.13	Režim reset, resume a suspend	14
4.3	ZigBee a jeho vlastnosti	14
4.3.1	ZigBee Aliance	14
4.3.2	Základní specifikace	15
4.4	Struktura komunikačního standardu	16
4.4.1	IEEE 802.15.4	16
4.4.2	Detekce volného kanálu	17
4.4.3	MAC	17
4.4.4	CSMA/CA	17
4.4.5	Síťová vrstva (NWK)	18
4.4.6	ZDO	20
4.4.7	Uživatelské aplikační objekty	20
4.4.8	Topologie sítě	21
4.4.9	Zabezpečení	22
5	Výběr součástek	23
5.1	Výběr mikrokontroléru	23
5.1.1	PIC 18F27J53	23
5.1.2	Základní specifikace	23
5.1.3	USB v PIC 18F27J53	24
5.2	Výběr ZigBee čipu/modulu	24
5.2.1	MRF24J40	24

5.2.2	MRF24J40MA	25
5.2.3	MRF24J40MB	25
5.2.4	MRF24J40MC	26
5.2.5	Konečný výběr ZigBee modulu	26
5.3	Napájení hlavní stanice	26
6	Obvodový návrh a fyzická realizace	27
6.1	Blokové schéma zapojení	27
6.2	Obvodové zapojení	27
6.3	Fyzická realizace prototypu	28
7	Způsob komunikace.....	30
7.1	Komunikace mezi sběrnou jednotkou a čidly	30
7.1.1	Přidávání nového čidla.....	30
7.1.2	Standardní vysílání	30
7.1.3	Vypnutí čidla.....	30
7.2	Struktura námi posílaného paketu	31
8	Firmware pro řídicí jednotku.....	33
8.1	Firmware pro obsluhu USB	33
8.1.1	HID firmware.....	33
8.2	Firmware pro obsluhu Zigbee modulu	34
8.2.1	SPI komunikace	34
8.2.2	Inicializace Zigbee	35
8.2.3	Nastavení vysílacího výkonu	37
8.2.4	Nastavení filtru a přijímacího módu	38
8.2.5	Přijímání a odesílání	38
8.3	Kompletní firmware přijímací jednotky.....	39
9	Podpůrný software pro PC	41
9.1	Obsluha USB.....	41
9.2	Vzhled hlavního formuláře	41
9.2.1	Vytvoření nového nebo načtení stávajícího souboru	42
9.2.2	Přidání nového čidla	43
9.2.3	Zpracování přijatých dat	45
9.2.4	Smazání čidla	46
9.2.5	Grafy	46
9.2.6	Manuál pro obsluhu programu.....	47
9.2.7	Kapacita čidel	47
10	Měření dosahu čidel.....	48
10.1	Měření v patře.....	48
10.2	Měření skrz patra	50
10.3	Měření na volném prostranství	52
10.4	Závěr z měření	52
11	Ekonomické zhodnocení.....	53
12	Závěr	55
Použitá literatura		
Příloha A – Schéma zapojení		
Příloha B – Topografie desky plošných spojů		
Příloha C – Osazovací výkres		
Příloha D – Vyhotovená deska plošných spojů		
Příloha E – Seznam použitých součástek		
Příloha F – Fotodokumentace		
Příloha G – CD		

1 Úvod

Tato práce se zabývá problematikou bezdrátových sensorových sítí. V dnešní době úspory energie a vysokého komfortu je potřeba měřit veličiny jako je teplota, vlhkost, osvětlení a podobné veličiny. Sensorické sítě se liší od sebe především použitou technologií jako je bezdrátová technologie, nebo sběrníková tj. po kabelu a použitelností, jako například pro domácnost či průmysl. Dále se kladou požadavky, aby zařízení bylo co nejjednodušší a také bezúdržbové. Pro domácí použití jsou tyto sensorové sítě především zvyšování komfortu uživatele. Využívají se zde například pro udržování stálé teploty v místnosti, nebo snížení energetické náročnosti budovy. V průmyslu jsou tyto sensorové sítě nepostradatelné, a to například v chemickém průmyslu kde chemické děje probíhají pod neustálým dohledem teplot.

Cílem práce je navrhnout zapojení pro hlavní sběrnou jednotku, jež komunikuje s čidly pomocí bezdrátové technologie Zigbee, vybrat vhodné komponenty a provést její fyzickou realizaci. Dále je zapotřebí vyvinout pro tuto jednotku příslušný firmware a podpůrný program, jenž bude spuštěn pod operačním systémem Windows. Tento program by měl být uživatelsky jednoduchý a změřená data snadno reprezentovatelná.

2 Průzkum trhu

Jelikož je ZigBee technologie mladá a příliš nezaběhnutá, příliš výrobců se s ní nezaobírá. Nejspíš je to tím , že uživatelé ještě o ni moc nevědí a nemají s ní zkušenosti. Jednou z větších firem, která se toto technologií zaobírá, je firma Legrand. Ta nabízí produkty pro regulaci osvětlení, teploty, detektory pohybu, nebo alarmy. Podle toho co jsem se dočetl z jejich katalogového listu, jsou všechna jejich čidla bateriově napájené a aktoři jsou napájené za síť. Ovládací dotykový panel s 3,5“ palcovou obrazovkou, přes který lze spouštět různé scénáře osvětlení, ovládat žaluzie, teploty a řídit zvukový rozvod, přijde na 12317 Kč v brutto ceně, viz **Obrázek 2.1**. Ovladač pro žaluzie nabízí Legrand za 892 Kč v brutto ceně.



Obrázek 2.1: Ovládací jednotka s panelem od firmy Legrand [16]

Dále jsem zjistil, že společnost Ravica nabízí řízení domácnosti pomocí ZigBee protokolu. V jejím portfoliu je hlavní jednotka s připojením přes USB na vložení SD karty a napájením ze sítě za 1041,97 \$, avšak žádné bližší informace jsem neobjevil. Dále nabízí spínané zásuvky, měření spotřeby a především, čím se podobá naší práci, senzorem teploty osvětlení a pohybu. Čidlo s těmito třemi měřenými veličinami stojí 214,59 \$, viz **Obrázek 2.2**. Zda je Ravica výrobcem, či pouze dodavatelem, jsem se nikde nedozvěděl.



Obrázek 2.2: Hlavní jednotka a čidlo od společnosti Ravica [17]

Produkt, který se nejvíce přibližuje mé práci je XStick od společnosti Digi. Tento adaptér do počítače slouží, k připojení do ZigBee sítě. Jeho cena činí 49 \$. Zařízení má interní anténu a s touto anténou výrobce uvádí, že jeho dosah je 15-20m v budově a 40-50m venku. Dále je možno od firmy Digi koupit senzor, jenž měří tři veličiny, a to teplotu, pohyb, a osvětlení. K tomuto senzoru se dá připojit pomocí výše zmíněného adaptéru do počítače. Tento senzor se dá pořídit za 109 \$. Viz **Obrázek 2.3**.



Obrázek 2.3: XStick a senzor od firmy Digi [18]

3 Struktura stanice

Hlavní stanice se skládá z jednotlivých částí, které plní svou funkci. Každý z bloků má jinou funkci a bloky dohromady fungují jako celek. Pro plnou funkčnost jednotky jsou zapotřebí následující části:

3.1 Napájecí část

Hlavní stanice bude neustále čerpat energii ze sítě, proto zde nejsou kladeny takové nároky na hardware napájení, jako pro jednotlivá čidla, která bývají napájena bateriově. Napájecí část bude obsahovat bloky, jež dokážou dodat jednotce potřebné napětí a potřebný proud.

3.2 Vysílací a přijímací část

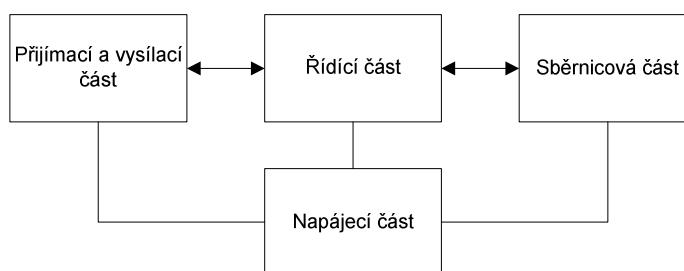
Vysílací a přijímací část se stará o to, aby data byla bezproblémově přenášena, tj. aby se dodržoval použitý komunikační standard. Tato část musí poskytnout určitý výkon, aby se dosáhlo požadovaných výsledků na přenos.

3.3 Sběrnicevá část

Tato část zabezpečuje komunikaci mezi jednotkou a nadřazeným ovládacím prvkem (PC, PLC, jiná ovládací jednotka). Zde záleží, k jakému nadřazenému prvku bude jednotka připojena, podle toho se bude vybírat druh sběrnice, po které bude komunikovat.

3.4 Řídící část

Řídící částí se rozumí část, jež mezi sebou propojí přijímací část a sběrniceovou část. Zde je kladen důraz na paměť a rychlost prvku, který se bude používat, aby nebrzdil v přenosu dat. Proto je zde důležité zvolit ten správný, jinak jednotka nedokáže stíhat komunikovat se všemi připojenými čidly. Viz **Obrázek 3.4**.



Obrázek 3.4: Struktura hlavní stanice

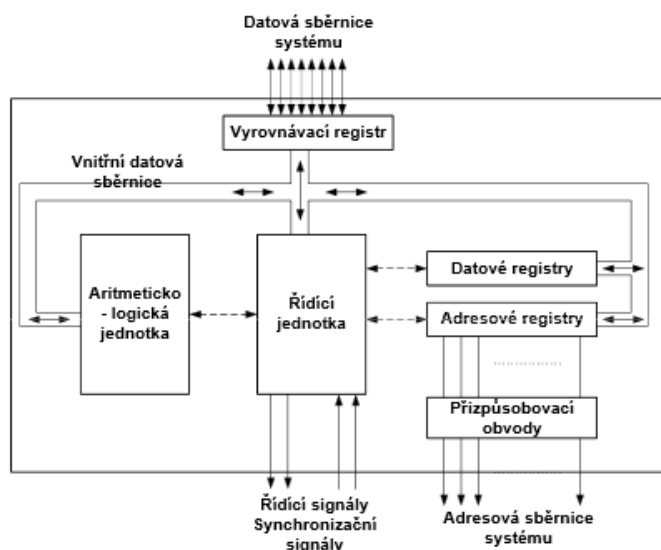
4 Teoretická část

4.1 Mikrokontrolér

Mikrokontrolér neboli jednočipový počítač je součástí obsahující kompletní mikro počítač. Jejich největší využití je v oboru regulace a řízení. V jednom čipu je obsaženo vše, aby obsáhnul celou aplikaci bez připojení podpurných obvodů. Jedná se o A/D převodníky D/A převodníky, časovače, komunikační sběrnice jako je USB, SPI, USART, paměti jako Flash EEPROM či ROM a také operační paměť RAM.

4.1.1 Základní struktura

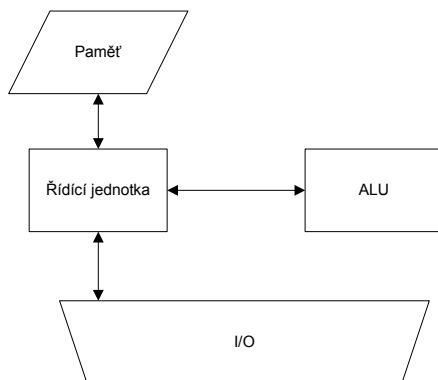
Základní struktura se skládá z ALU, která vykonává aritmetické a logické operace. Taktovací kmitočet zde dosahuje hodnota až stovek MHz a šířka slova až 64 bitů. Dále z operační paměti typu RAM, jejíž velikost se pohybuje v desítkách kB. μC , by se neobešel bez programové paměti. Tato paměti může být typu Flash či EEPROM dosahující až jednotek MB. Nutným prvkem je zde oscilátor ať interní RC, či externí krystalový. Nedílnou součástí je vstupní a výstupní rozhraní neboli I/O. Viz **Obrázek 4.1.** [20]



Obrázek 4.1: Obecné schéma mikroprocesoru [19]

4.1.2 Von Neumannova architektura

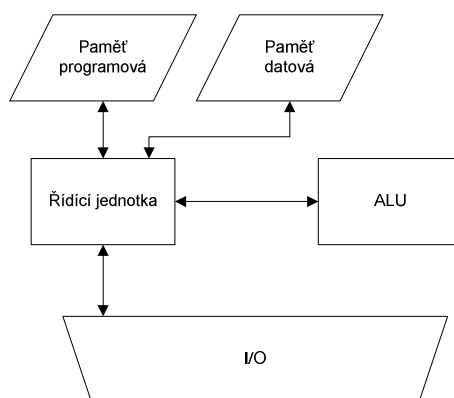
Tato architektura popisuje základní bloky mikropočítače, jež má společnou paměť pro data a instrukce. Proto je i zpracování sekvenční. Tuto architekturu využívají především počítače typu x86 (PC). Viz **Obrázek 4.2**.



Obrázek 4.2: Obecné schéma Von Neumannovy architektury [19]

4.1.3 Harvardská architektura

Tato architektura navazuje na architekturu Von Neumannovu, odstraňuje však některé nedostatky, jako je přepsání řídicího programu. Tento fakt upravuje díky rozdělení paměti na programovou a datovou. Takto jak využívá i dvě sběrnice a vzájemný přístup jak k datům, tak instrukcím. Viz **Obrázek 4.3**. Takovouto architekturu má většina mikrokontrolérů.



Obrázek 4.3: Obecné schéma Harvardské architektury [19]

4.1.4 RICS a CISC

Mikrokontroléry se také dělí podle instrukční sady.

RICS

Tato zkratka v překladu znamená počítač s redukovanou instrukční sadou. Instrukce obsahuje jednoduché instrukce, jejich délka provádění je vždy pouze jeden cyklus. Tyto instrukce jsou implementovány na procesoru, dochází tak ke zvýšení rychlosti jejich provádění. Registry jsou pouze víceúčelové a využívá se řízení instrukcí. Tuto sadu využívají například mikrokontroléry PIC.

CISC

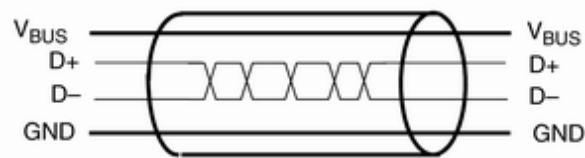
Tato zkratka označuje procesory s velkou sadou instrukcí a malým počtem registrů. Na rozdíl od RICS jsou jejich instrukce různě dlouhé a jejich vykonávání trvá různě dlouhou dobu. Tato sada je využívána procesory postavené na architektuře Intel x86. [20]

4.2 USB

USB je v této době nejrozšířenější standard pro připojování periférií k osobním počítačům. Důvodem je jednoduchost připojení, vysoká přenosová rychlost a také možnost přímého napájení (není třeba externího napájení).

4.2.1 Základní vlastnosti USB

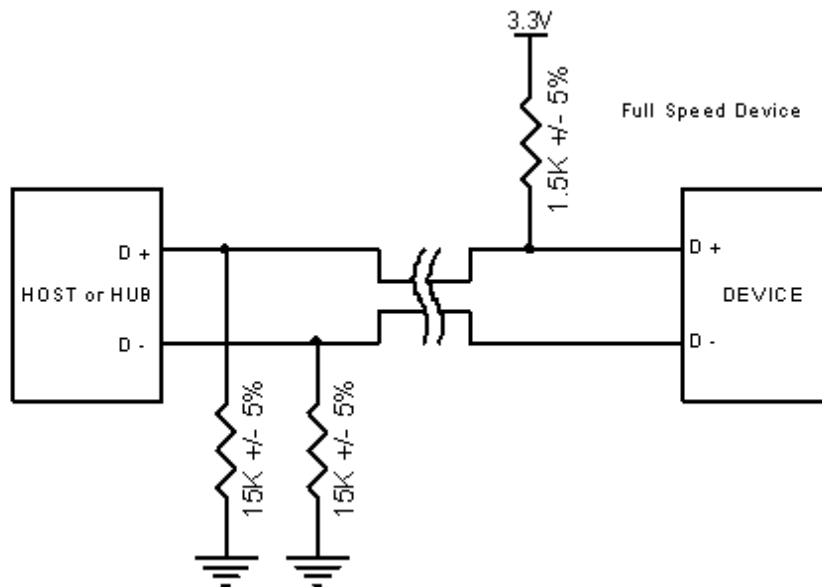
Jedná se o strukturu Master – Slave, kde je pouze jeden Master, ostatní zařízení jsou Slave. USB zařízení se propojují pomocí čtyřvodičového kabelu, kde dva slouží pro napájení a další dva D+ D- jsou určeny pro sériový poloduplexní přenos dat s využitím diferenčního kódování. Viz **Obrázek 4.4**. Přenášené bity se kódují pomocí metody NRZI s vkládanými bity (bit stuffing) pro synchronizaci. Na straně HOSTU (například počítač nebo rozbočovač) jsou na D+ a D- připnuté rezistory o hodnotě 15 k Ω na zem, tudíž jsou oba datové vodiče na úrovni L.[2]



Obrázek 4.4: Struktura USB kabelu [2]

4.2.2 Detekce rychlosti

Výše zmíněný μC umožňuje komunikaci pomocí Low Speed (1,5 Mbps) a Full Speed (12 Mbps). Detekce rychlosti na sběrnici po připojení zařízení se provádí pomocí tzv. pull-up rezistoru (rezistor připojen jedním koncem na napájecí napětí 3,3V). Tento rezistor o hodnotě 1,5 k Ω je připojen buď na D+, zařízení pracuje na normální rychlosti (Full Speed), nebo na D- zařízení pracuje na snížené rychlosti (Low Speed). Viz **Obrázek 4.5**. Z důvodu diferenčního kódování se po detekci rychlosti musí rezistor odpojit, jinak by linka nebyla vyvážená.[2]



Obrázek 4.5: Detekce rychlosti na sběrnici pro Full Speed [2]

4.2.3 Možnosti napájení

Připojení umožňuje napájení +5 V. Odebíraný proud může bez problému dosahovat hodnoty až 100 mA bez dalších opatření. Pokud by zařízení potřebovalo odebírat větší proud, musí si o něj zažádat u Host zařízení. Po takovém to ohlášení je k dispozici až 500 mA. Signálové úrovně na datových vodičích D+ a D- dosahují napěťových úrovní 0 V / 3,3 V.[4]

4.2.4 Napěťové úrovně

Napěťové úrovně pro logické hodnoty L (nízká) a H (vysoká) se podle použití přenosové rychlosti liší. Při použití Low Speed je logická hodnota L stanovena v rozmezí 0,0 V až 0,3 V a hodnota H v rozsahu 2,8 V až 3,6 V. Pro Full Speed logické úrovně L napětí dosahuje hodnot v rozmezí -10 mV až 10 mV a logické úroveň H v rozmezí 360 mV až 440 mV. [2]

4.2.5 Kódování

Výše zmíněné úrovně H a L přímo nerepresentují přenášené bity. USB komunikace používá upravené kódování NRZI. Z důvodu opačné polariry přenášených dat při nízké rychlosti oproti datům přenášených rychlostí plnou, si zavedeme stavy J a K. Reprezentace bitu při přenosu je následující: pro bit s nulovou hodnotou se změní stav z J na K či naopak z K na J, zatímco pro bit s logickou hodnotou jedna se stav linky nezmění (zůstává ve stavu, v jakém byla). Viz **Tabulka 4.1**. [3]

Tabulka 4.1: Stavy J a K na sběrnici při různých rychlostech [2]

Stav	Režim Low speed	Režim Full speed
J	Diferenciální ,0'	Diferenciální ,1'
K	Diferenciální ,1'	Diferenciální ,0'

Jelikož je sběrnice diferenciálního typu, lze v jednom okamžiku zakódovat jeden ze čtyř stavů: Viz **Tabulka 4.2**.

Tabulka 4.2: Stavy na sběrnici a jimi příslušné úrovně na vodičích [2]

Stav sběrnice	Úroveň na vodiči D+	Úroveň na vodiči D-	Poznámka
Diferenciální ,1'	H	L	
Diferenciální ,0'	L	H	
Single Ended Zero (SE0)	L	L	konec paketu, reset atd.
Single Ended One (SE1)	H	H	nepovoleno

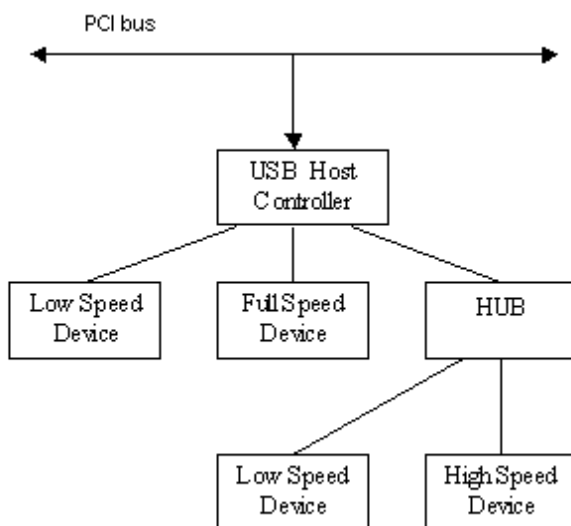
4.2.6 Synchronizace

Synchronizace se provádí takzvaným bit stuffing, který je reprezentován logickou hodnotou nula. Tento bit je použit, pokud se na sběrnici objeví datový tok s šesti za sebou jdoucími logickými hodnotami jedna, tj. stav sběrnice se šestkrát

za sebou nezměnil. Stuffing bit vynutí změnu stavu na lince. Tento impuls se dá využít pro synchronizaci zařízení.[3]

4.2.7 Přenos dat

Topologie USB sběrnice je stromová s jedním kořenovým uzlem, viz **Obrázek 4.6**. Kořenový uzel adresuje jednotlivá připojená zařízení a řídí přenos dat. Jelikož sběrnice používá ke komunikaci pouze dva datové vodiče, datový proud neobsahuje jen užitečná data, nýbrž i příkazy, které posílá host připojeným zařízením, a také zařízení posílají hostu informace o stavu přenosu dat, vysílání, příjmu atd.[3]



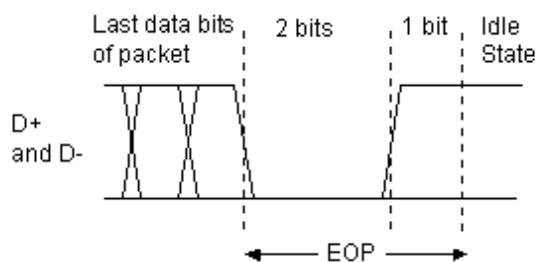
Obrázek 4.6: Ukázka stromové topologie na sběrnici USB [2]

4.2.8 Pakety

V komunikaci se vyskytují tři druhy paketu: token paket s příkazem, který vysílá kořenový uzel, jenž řídí komunikaci, následuje datový paket a komunikaci ukončuje takzvaný handshake paket, který nese informaci o přenosu (potvrzení či chyba přenosu).

Každý paket je uváděn synchronizačním bytem o bitové hodnotě „00000001“. Pro přenosy o vysoké rychlosti má tento synchronizační byt velikost 32 bitů.

Přenos celého paketu ukončuje sekvence EOP. Fyzicky na sběrnici je tato sekvence realizována následovně: na dobu přenosu dvou bitů se nastaví stav SE0, dále následuje přechod do stavu J na dobu minimálně jednoho bitu. V takovém to stavu může určitý úsek sběrnice setrvat libovolně dlouhou dobu. Viz **Obrázek 4.7**. [3]



Obrázek 4.7: Fyzická reprezentace EOP sekvence a přechod do stavu idle [3]

4.2.9 Token paket

Tento paket slouží k adresaci zařízení, specifikaci koncového bodu (logická adresa v rámci zvoleného zařízení – používá se tam, kde mají zařízení více funkcí například zápis a vyčítání dat). Token paket má vždy délku 24 bitů, jenž má následující strukturu, viz **Tabulka 4.3**.

Tabulka 4.3: Složení Token paketu [3]

Označení	Počet bitů	Význam
Sync	8	synchronizační sekvence
PID	8	identifikace typu paketu
ADDR	7	adresa zařízení 1–127
ENDP	4	koncový bod v rámci zvoleného zařízení
CRC5	5	kontrolní součet
EOP		konec paketu

PID – Identifikátor typu paketu, jenž je reprezentován čtyřbitovým číslem, které je zapsáno v osmi bitech dvakrát (přímo a inverzně). Tímto způsobem je ošetřena detekce chyb a odpadá vkládání dodatečného bitu (stuffing bit), viz **Tabulka 4.4**

Tabulka 4.4: Stavy PID a jejich bitová reprezentace [3]

PID	Bitová hodnota	Význam
OUT	0001	přenos dat z kořenového uzlu do zařízení (zápis)
IN	1001	přenos dat ze zařízení do kořenového uzlu (čtení)
SETUP	1101	inicializace zařízení (speciální datový paket o délce osmi bytů)

- ADDR – Adresa zařízení, se kterým chce Host navázat spojení. Jak je vidět, je možné adresovat 127 zařízení na jeden segment USB. Adresa 0 je rezervována pro zařízení, kterým ještě nebyla přiřazena adresa.
- ENDP – Takzvaný Endpoint, kterých může být adresováno na koncových zařízeních až 16. Například pevný disk připojený přes USB, jenž je rozdělen na dva oddíly, bude mít minimálně 4 adresované endpointy (zápis dat pro první oddíl, vyčítání dat pro první oddíl, zápis dat pro druhý oddíl, vyčítání dat pro druhý oddíl). Tyto kódy přiřazuje především výrobce.
- CRC5 – Pětibitový kontrolní součet.
- EOP – Konec paketu, jehož princip je vysvětlen výše.[3]

4.2.10 Datový paket

Následuje po Token paketu, jeho přenos probíhá buď z Hostu do zařízení či naopak. Tento paket má podobnou strukturu jako Token paket. Na rozdíl o něj nemá pevně danou délku. Jako každý paket, i tento začíná synchronizačním bytem, viz **Tabulka 4.5.**[3]

Tabulka 4.5: Složení datového paketu [3]

Označení	Počet bitů	Význam
Sync	8	synchronizační sekvence
PID	8	identifikace typu paketu
DATA0/DATA1/DA-TA2/MDATA	max. 8192	přenášená data
CRC16	16	kontrolní součet
EOP		konec paketu

Nejdůležitější částí tohoto paketu jsou datové pakety DATAx. Jednotlivé datové pakety jsou potvrzeny tzv. handsahke paketem typu ACK.

- DATA0 a DATA1 – Pro zařízení pracující v Low Speed a Full Speed. V režimu normální rychlosti dosahují data délky až 1023 bytů, kdežto u pomalé rychlosti pouhých 8-mi bytů (zde je velmi neefektivně využita přenosová kapacita).

- DATA2 a MDATA – Ve standardu 2.0, kde tyto typy umožňují přenášet data rozdělená do bloků až o velikosti 1kB, viz **Tabulka 4.6.** [3]

Tabulka 4.6: Možné hodnoty PID u datového paketu a jejich bitová reprezentace [3]

PID	Bitová hodnota	Význam
DATA0	0011	data o délce 0–1023 bytů (pro low speed 8 bytů)
DATA1	1011	data o délce 0–1023 bytů (pro low speed 8 bytů)
DATA2	0111	použito pro vysokorychlostní přenosy
MDATA	1111	použito pro vysokorychlostní přenosy

4.2.11 SOF paket

SOF paket neboli Start of Frame se využívá pro synchronizaci dat při izochronním přenosu. Uspořádání paketu je založeno na struktuře Token paketu, až na to, že místo adresy a čísla endpointu je přenášena 11-ti bitová hodnota počítadla, viz **Tabulka 4.7.** Vyslání paketu se provádí každou milisekundu. Po překročení maximální hodnoty počítadla (hodnota 2047) se znovu čítá od nuly. [3]

Tabulka 4.7: Složení SOF paketu [3]

Označení	Počet bitů	Význam
Sync	8	synchronizační sekvence
PID	8	binární hodnota 0101 PID paketu typu Start of Frame
frame#	11	aktuální hodnota počítadla
CRC5	5	kontrolní součet
EOP		konec paketu

4.2.12 Handshake paket

Tato skupina paketů slouží pro oznámení stavu příjemce odesílateli dat. Paket se odesílá ihned po přijmutí dat. Paket se skládá ze synchronizačního bajtu, PID identifikace typu a EOP.

Základními typy jsou ACK (potvrzení příjmu/vysílání) a NAK (příjem/vyslání se nezdařilo). Další typy handshake paketů, viz **Tabulka 4.8.** [3]

Tabulka 4.8: Typy handshake paketu a jejich bitová reprezentace [3]

PID	Bitová hodnota	Význam
ACK	0010	datový paket byl v pořádku přečten
NAK	1010	zařízení nemohlo data přijmout či odvíšlat
STALL	1110	zařízení je ve stavu trvalé chyby, čeká se na reset od kořenového uzlu
NYET	0110	USB 2.0 – rozdělená transakce (více bloků dat) nebyla ještě dokončena
ERR	1100	rozdělená transakce (více bloků dat) nebyla dokončena z důvodu chyby

4.2.13 Režim reset, resume a suspend

Kořenový uzel může zařízení přepnout do stavu idle (zařízení naslouchá na sběrnici a čeká na data), po určité době je však nahrazen režimem suspend. Do tohoto režimu přechází, pokud režim idle trvá minimálně 3ms.

Probuzení z režimu suspend se nazývá režim resume. Probuzení se provádí následovně: nejdříve se na obou signálových vodičích změní napětí, tento děj by měl trvat minimálně 20 ms, poté řídicí uzel po dobu trvání dvou bitů nastaví stav SE0, následuje změna na stav J. V tuto chvíli je zařízení probuzeno a schopno příjmu dat.

Reset komunikační části zařízení se provádí nastavením stavu SE0 po dobu minimálně 10ms, poté se komunikace restartuje a po uplynutí několika milisekund je zařízení schopno komunikovat. [3]

4.3 ZigBee a jeho vlastnosti

4.3.1 ZigBee Aliance

Jde o společenství firem, které vzniklo roku 1997. V roce 2003 byl jimi vyvíjený standard IEEE 802.15.4 (ZigBee) uveden na trh. V dnešní době je ve společenství přes 300 firem a převážně firem zaměřujících se na automatizaci například: Omron, Honeywell, Siemens, také Texas Instruments, Analog Device, Cisto Systems, Farnell, Microchip atd. Úkolem aliance je testovat a vydávat nové revize standardů, také jejich certifikace. Pokud nějaká firma chce nabízet své produkty se ZigBee technologií, musí se stát členem ZigBee Aliance.

4.3.2 Základní specifikace

Bezdrátový standard IEEE 802.15.4 je spravován ZigBee Aliancí. Byl vyvinut, aby zaplnil mezeru mezi technologiemi Bluetooth a WIFI, viz **Tabulka 4.9**. Největší mezera je v průmyslových bezdrátových sítích a senzorové technice. Zde může zastávat sériového přenosu RS-232 nebo RS-485. ZigBee pracuje v bezlicenčních pásmech, přičemž přenosová rychlost se pohybuje od 20kb/s až po 250kb/s, a to podle pásma.

- pásmo ISM 2.4 GHz, 16 kanálů, přenosová rychlost 250kb/s, definováno celosvětově
- pásmo 915 MHz, 10 kanálů, přenosová rychlost 40kb/s, definováno pro americký kontinent
- pásmo 868 MHz, 1 kanál, přenosová rychlost 20kb/s, definováno pro Evropu

Nízké přenosové rychlosti poskytují vyšší odolnost proti rušení. Velké úspory energie u bateriově napájených zařízení se dá dosáhnout funkcí tzv. „uspání“ čidla. To se dá využít, když není potřeba, aby čidlo snímalo, tak se jednoduše „uspí“ a když jej je potřeba, tak se probudí. Dalším účelem bylo ZigBee aplikovat do málo výkonových 8–mi bitových mikrokontrolérů, proto je komunikační protokol jednoduchý a díky tomu nezabere více než 30 kB programové paměti. Protokol se skládá ze tří základních vrstev, ale tyto vrstvy více analyzují ve struktuře komunikačního standardu.[5]

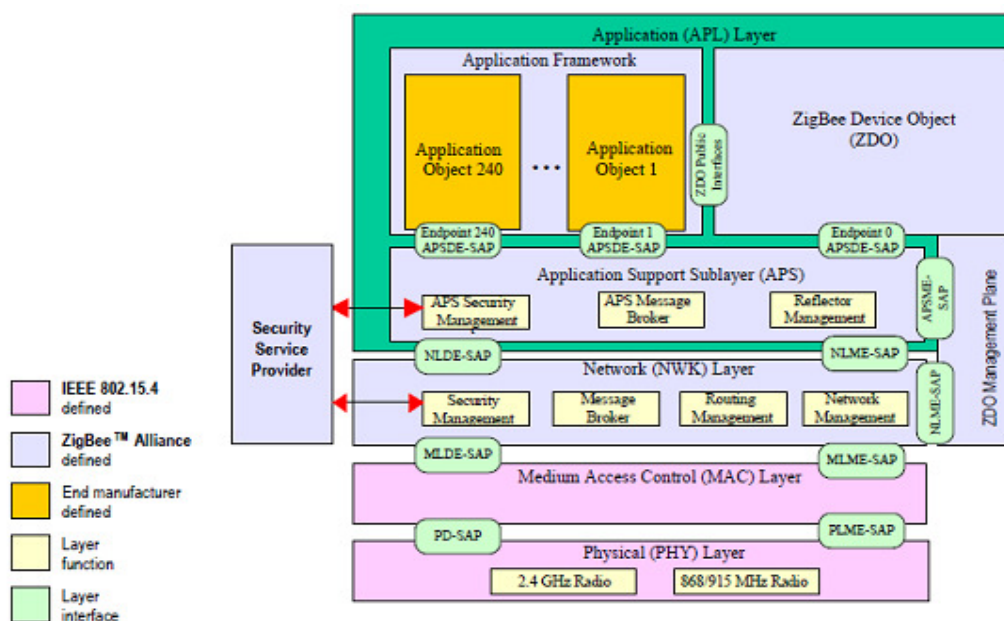
Tabulka 4.9: Porovnání základních parametrů používaných standardů pro bezdrátovou komunikaci [5]

Obchodní jméno Standard	GPRS/GSM 1xRTT/CDMA	Wi-Fi™ 802.11b	Bluetooth™ 802.15.1	ZigBee™ 802.15.4
Aplikační zaměření	Široké oblasti Hlas & Data	Web, Email, Video	Náhrada za kabel	Monitorování & Řízení
Systémové zdroje (paměť)	16MB a více	1MB a více	250KB a více	4KB - 32KB
Životnost baterií (dny)	1-7	0.5 - 5	1 - 7	100 – 1 000 i více
Max. velikost sítě (počet uzlů/sít')	1	32	7	65 000 (příp. až 2 ⁶⁴)
Přenosová rychlost (Kb/s)	64 – 128	11 000	720	20 - 250
Komunikační dosah (m)	1 000 i více	1 - 100	1 - 10	1 – 100
Výhody	Dosažitelnost, Kvalita	Rychlost, Flexibilita	Cena, Jednoduchost	Spolehlivost, Výkon/Cena

4.4 Struktura komunikačního standardu

Jak již bylo zmíněno, komunikační protokol se skládá ze tří základních vrstev, vrstvy standardu IEEE 802.15.4, síťové (NWK) a aplikační (APL), viz. **Obrázek 4.8**.

Vrstva N poskytuje služby vrstvě N+1 a sama využívá služeb jí poskytovaných vrstvou N-1. Pro vzájemnou výměnu dat při vertikální komunikaci je nutno definovat rozhraní mezi jednotlivými vrstvami. Toto rozhraní se nazývá SAP, přístupový bod ke službě. Důležitou vlastností každého SAP je schopnost poskytovat služby většímu počtu vrstev stejné úrovně, tj. paralelně a nezávisle více instancím vrstvy (N+1).[6]



Copyright © 2007 ZigBee Standards Organization. All rights reserved.

Obrázek 4.8: Model komunikačního protokolu ZigBee [6]

4.4.1 IEEE 802.15.4

První nejnižší vrstva je IEEE 802.15.4-2003. Standard definuje dvě spodní vrstvy: Fyzickou vrstvu (PHY) a vrstvu MAC. Fyzická vrstva může pracovat na dvou rozdílných frekvencích, a to na 868/915 MHz a 2,4 GHz. Frekvence 868 MHz je určena pro Evropu a 915 MHz pro Spojené státy americké a Austrálii, 2,4 GHz je určena globálně. Pro přenos se datový signál moduluje metodou O-QPSK (pro kmitočet 2,4 GHz) nebo o BPSK (pro ostatní kmitočty). Přenáší se prostřednictvím DSSS, podobně jako technologie WIFI.

4.4.2 Detekce volného kanálu

Detekce volného kanálu CCA je vyvolána na požadavek vyšší MAC vrstvy. CCA je rovněž součástí managementu fyzické vrstvy (NLME).

Fyzická vrstva provede CCA jednou ze tří dostupných módů:

1. Detekce energie nad prahovou úroveň – pokud fyzická vrstva zjistí, že naměřená hodnota je mimo povolený dosah, nahlásí vyšší vrstvě, že medium je obsazené.
2. Detekce nosné vlny – V tomto případě fyzickou vrstvu nezajímá, zda-li je naměřená hodnota vyšší, nebo nižší než stanovená mez. Pokud detekuje nějaký signál, je kanál označen jako obsazený.
3. Detekce nosné vlny s energií vyšší než prahová úroveň – umožňuje jemnější řízení požadavku MAC vrstvy. Můžeme si vybrat z logické kombinace obou předchozích metod.[5]

4.4.3 MAC

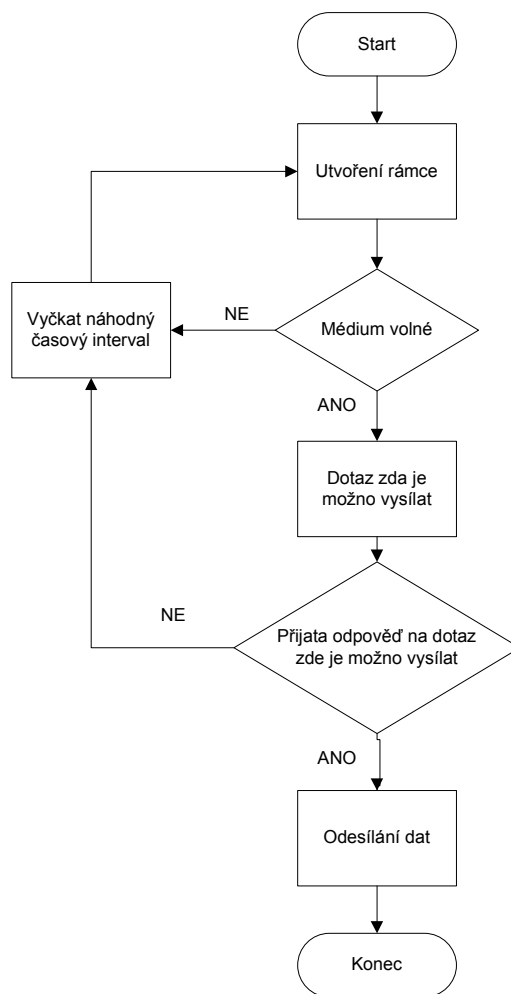
MAC vrstva řídí přístup k rádiovému kanálu, který využívá metody CSMA/CA. Také definuje komunikaci mezi jednotlivými zařízeními prostřednictvím rámců. Konkrétně jsou definovány čtyři typy komunikačních rámců využívané buď pro přenos užitečných datových informací, nebo k režijním účelům souvisejícím se sestavením, správou a řízením sítě:

- Data Frame – rámeček pro přenos užitečné informace pro všechny datové přenosy
- Acknowledgement Frame – rámeček pro přenos potvrzovací informace a je využitelný pouze na úrovni MAC pro potvrzovanou komunikaci
- MAC Command Frame – rámeček k centralizované konfiguraci, nastavení a řízení klientských zařízení v síti
- Beacon Frame – rámeček k synchronizaci zařízení v síti. Je využíván hlavně při konfiguraci sítě v módu, v němž umožňuje uvádění klientských zařízení do spánkových režimů s extrémně sníženou spotřebou.[5]

4.4.4 CSMA/CA

Tento protokol patří do třídy protokolů, jež se označuje jako metoda s vícenásobným přístupem a nasloucháním nosné frekvence. Rozdíl v CSMA/CA a CSMA/CD je v tom, že CA nezajišťuje výskyt kolizí, tudíž je odeslaný paket ztracen a musí se odeslat znovu. Fungování této metody nejlépe vysvětlí

Obrázek 4.9.



Obrázek 4.9: Diagram způsobu přístupu ke kanálu u CSMA/CA metody

4.4.5 Síťová vrstva (NWK)

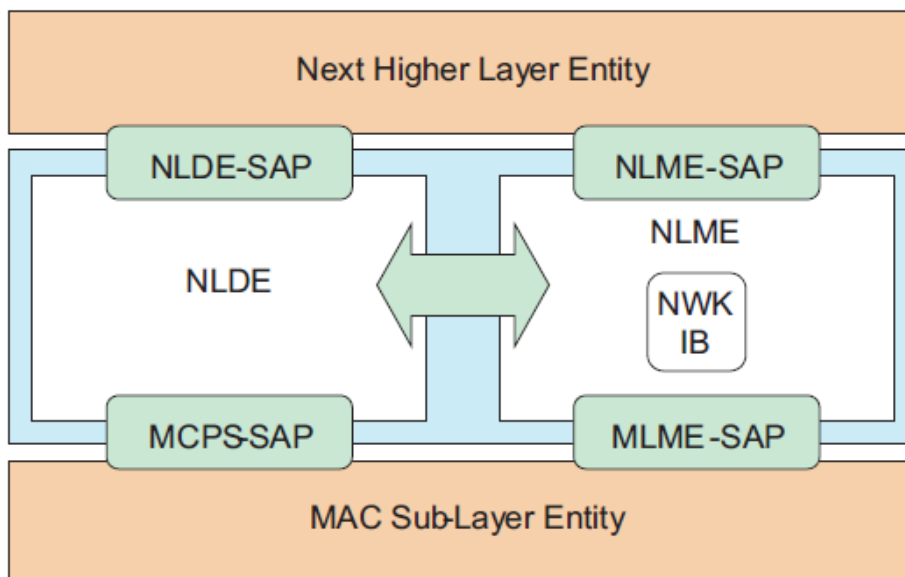
Síťová vrstva poskytuje funkcionalitu pro zajištění správné funkce podvrstvy IEEE 802.15.4-2003 MAC a zároveň poskytuje služby aplikační vrstvě. Síťová vrstva obsahuje dvě služby, datovou NLDE a řídicí NLME, viz **Obrázek 4.10**. Tyto dvě služby komunikují také mezi sebou. Dále síťová vrstva obsahuje tzv. NIB, zde jsou obsaženy informace nezbytné pro správu sítě.

NLDE

Poskytuje službu k přenosu dat mezi jednotlivými jednotkami, které musí být v jedné síti. Vytváří takzvaný APDU. Zajišťuje správnou cestu přenosu buď přímo na koncovou jednotku nebo pomocí směrovačů. Dále se stará o zabezpečení přenosu.

NLME

Tato služba spravuje: Nastavení nových zařízení, spouštění sítě, routing, záznam cesty přenosu a zefektivnit přenos, adresovat zařízení v síti, připojení a odpojení sítě (schopnost připojit se k jiné síti), bezpečnostní ochrana.



Obrázek 4.10: Vnitřní komunikace mezi službami [6]

Na síťové úrovni ZigBee rozlišuje tři duhy uzlů:

- síťový koordinátor, který je typu FFD
- síťový směrovač, také typu FFD
- koncové zařízení, obvykle typu RFD. Toto není daným pravidlem, ale je to dáno z hlediska ekonomického a úspory energie.[6]

Aplikační vrstva (APL)

Aplikační vrstva se skládá ze čtyř částí:

- uživatelského programu
- aplikační pomocné podvrstvy
- ZDO
- Uživatelských aplikačních objektů

Aplikační podvrstva (APS)

Aplikační podvrstva (APS) je vrstva, která má na starosti párovací (Binding) tabulky, které obsahují záznam o propojení dvou zařízení v závislosti na jejich parametrech, jejich potřebách a podporovaných službách. Díky párovacím tabulkám

přeposílá zprávy dalším zařízením. Díky tomuto způsobu ZigBee podporuje multiskokovou komunikaci.

4.4.6 ZDO

ZDO definují přesnou roli zařízení v síti. Definuje, zda je zařízení Koordinátor, Router či Koncové zařízení. Dále vytváří a odpovídá na žádosti o spojení mezi zařízeními, zřizuje zabezpečené spojení mezi dvěma prvky sítě, také zajišťuje hledání v síti a spravuje používané zabezpečovací klíče.

Koordinátor

Zajišťuje založení, nastavení a start sítě. Koordinátor vybírá komunikační kanál, na kterém mohou zařízení komunikovat, nastavuje PANID sítě, přijímá požadavky na připojení a v případě šifrování, sdílí zabezpečovací klíče. V síti může existovat pouze jeden koordinátor, jelikož je klíčovým prvkem, na kterém stojí celá síť.

Router

Zajišťuje rozšíření sítě na vyšší počet prvků. V síti může být více zařízení typu router. Router by neměl být stejně jako koordinátor napájen z baterií, jelikož na něm závisí celá větev sítě.

Koncový prvek

Koncový prvek sítě. Typický koncový prvek je čidlo teploty, senzor pohybu apod. End-Device je jediným možným prvkem v ZigBee síti, který může být napájen pouze bateriově.

4.4.7 Uživatelské aplikační objekty

Uživatelské aplikační objekty implementují do sítě konkrétní požadavky pro aplikaci. Tyto požadavky se nazývají ZigBee Aplikační profily. Každý profil definuje typy použitých zařízení, typy zpráv a jejich formáty. Každý profil má vlastní 16-ti bitové identifikační číslo, které přiděluje ZigBee Alliance. Aplikační profil je dohoda o formátech zpráv a akcích, které umožní zařízením od různých výrobců komunikovat mezi sebou a tím zajistit kompatibilitu. Každé zařízení před vydáním na trh musí vyhovovat předem danému profilu. ZigBee aliance definuje dva druhy profilů: průmyslový specifický a veřejný.

Veřejné profily pokrývají základní využití technologie v různých oborech. Specifikace si definuje výrobce sám, avšak profil musí projít schválením. V současné době jsou k dispozici následující ZigBee profily, uváděné originálními názvy:

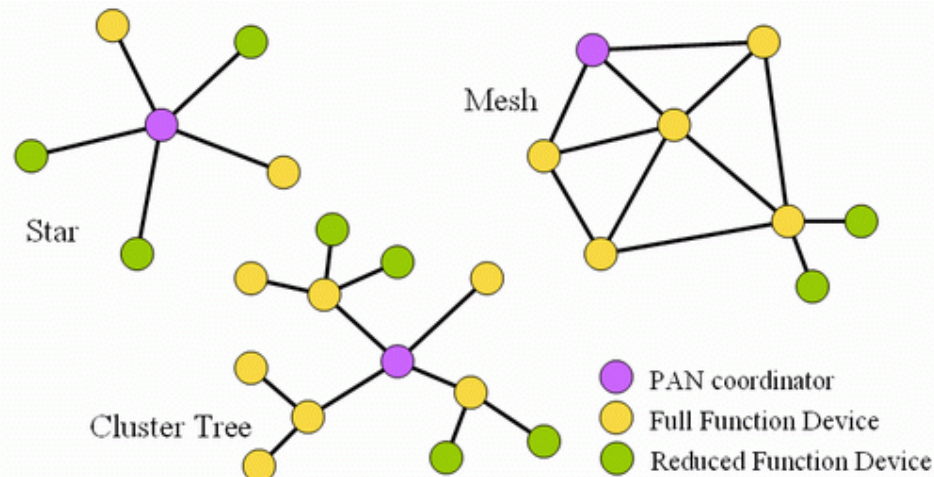
- ZigBee Building Automation (pro komerční použití v ovládání budov)
- ZigBee Remote Control (náhrada za infračervených dálkových ovladačů)
- ZigBee Smart Energy (vysoce úsporné projekty)
- ZigBee Health Care (pro nekritické služby ve zdravotnictví a v oblasti fitness či péči o chronicky nemocné. Profil vyhovuje standardu ISO/IEEE 11073. Typická zařízení jsou měřiče cukru či informativní EKG.)
- ZigBee Home Automation (domácí automatizaci dostupné pro inteligentní dům jako je automatické rolování žaluzií, osvětlení, otevírání garážových vrat atd.)
- ZigBee Input Device (využití ZigBee technologie v ovládacích prvcích pro osobní počítače)
- ZigBee Telecom Services (pro služby typu placení či lokalizace pomocí mobilních telefonů v budovách. Předpokládá se, že telefon bude mít speciální ZigBee SIM kartu.)
- ZigBee 3D Sync (pro bezdrátové brýle pro sledování 3D videa)
- ZigBee Retail Services (chytré monitorování, kontroly a automatizace objednáni a doručení zboží)[6]

4.4.8 Topologie sítě

Jedná se o síť typu PAN. Standard podporuje tři různé síťové topologie viz **Obrázek 4.11**.

- topologie hvězdicová – má jeden centrální řídicí uzlem (koordinátor sítě), který řídí celou strukturu a přiděluje jednotlivým zařízením PANID. Nevýhodou této topologie je, že všechna komunikace proudí přes jeden uzel. Při jeho výpadku je nefunkční celá síť.
- stromová struktura – vychází hvězdicové topologie. Je rozšířena o směrovače, které umožňují zvětšit vzdálenost mezi koordinátorem a koncovým zařízením, a to tím, že koordinátor „najde“ cestu přes směrovače k určenému koncovému zařízením. Při výpadku jednoho ze směrovačů nepřestává fungovat celá síť, ale pouze jen část, kterou nefunkční směrovač obsluhuje.

- Mesh struktura – u předchozích dvou zapojení koordinátor sítě, je zodpovědný za spuštění a nastavení sítě, tj. nadefinování adres a nastavení směrovačů. U topologie typu mesh, je umožněna komunikace typu peer-to-peer. Zde vznikají tzv. redundantní spoje a koordinátor není zbytečně zatížen. Síť není tak náchylná na výpadky jako hvězdicová a stromová, při výpadku některého ze směrovačů se cesta přizpůsobí (pokud to jde) přes jiné směrovače.[5] [15]



Obrázek 4.11: Příklady topologií se standardem ZigBee [5]

4.4.9 Zabezpečení

Bezpečnostní nástroje v rámci sítě nezaručují jen ochranu vlastních dat v datových rámcích, ale i příslušné nástroje jsou aplikovány i k zjištění MAC řidičích, signálních a potvrzovacích rámců. Jako základ se používá 64 bitový nebo 128 bitový kryptografický algoritmus AES. MAC vrstva nařídí vlastní bezpečnostní procesy. Řízení procesu náleží síťové vrstvě (NWK), která nastavuje klíče a zabezpečení. MAC vrstva získá klíč od NWK a aplikuje jej na rámce. Podle hlavičky MAC se pozná, zda je bezpečnostní mechanismus aplikován. Data jsou zakódována a doplněna podle zvolené úrovně dalšími bezpečnostními nástroji, které generují příslušné doplňkové rámce a data čím se mění rámcová číslovací sekvence. Takto dochází k většímu celkovému zabezpečení přenosu dat.[6]

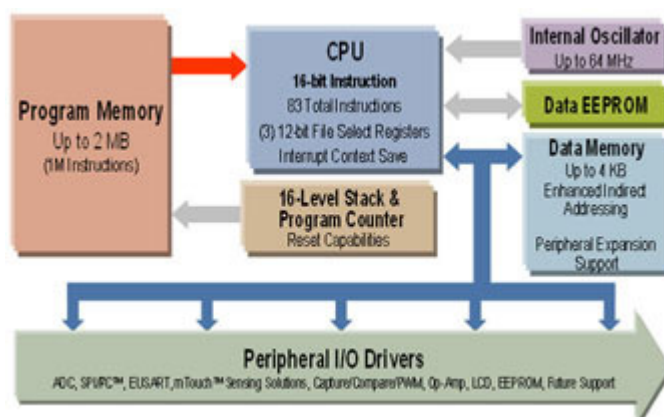
5 Výběr součástek

5.1 Výběr mikrokontroléru

Na trhu je dosti na výběr tohoto produktu. Každý výrobce mikrokontrolérů má široké zastoupení pro všemožné aplikace. Také se liší v počtech vstupů, frekvenci, šířce sběrnice, periferních obvodech. Každý výrobce má jinou instrukční sadu, proto je na uživateli, s jakou instrukční sadou umí pracovat. Já osobně mám nejvíce zkušeností s instrukční sadou pro μC od Microchipu, proto jsem vybíral z jejich portfolia.

5.1.1 PIC 18F27J53

Jedná se o 8-mi bitový mikrokontrolér, z rodiny PIC 18. Disponuje USB v2.0, dále úspornou technologií nanoWatt XPL, která je pro hlavní jednotku nepodstatná z důvodu permanentního napájení z USB. Tento typ disponuje 28 vývody, výrobce vyrábí i typ s 44, ale pro naši aplikaci by byly zbytečné a nevyužité. Vnitřní struktura viz **Obrázek 5.1.**[1]



Obrázek 5.1: Vnitřní uspořádání mikrokontroléru PIC řady 18 [12]

5.1.2 Základní specifikace

Hlavní důvodem výběru tohoto μC byla integrovaná podpora USB v2.0, kde jsou datové vodiče D+ a D- z USB připojeny přímo na vstupní porty, aniž by bylo zapotřebí dalšího obvodu. Dalším požadavkem bylo, aby disponoval SPI komunikací pro připojení ZigBee modulu. Lze jej napájet od hodnoty 2,0 V po 3,6 V, ovšem výrobce doporučuje napájení 3,3 V, aby nevznikaly problémy při komunikaci USB. Oscilátor μC se dá použít jako vnitřní, tak vnější. V mém případě je využito 16 MHz krystalového externího oscilátoru. [1]

5.1.3 USB v PIC 18F27J53

Firma Microchip má pro své mikrokontroléry podpůrný firmware pro USB sběrnici, jenž je bezplatně dostupný na jejich stránkách. Firmware stačí vzít a upravit si jej pro své účely. Nejvíce nás bude zajímat, jaký typ přenosu budeme využívat, jinak řečeno jaký typ endpointu. Microchip má již předkonfigurované tyto typy:

- HID – human interface device
- CDC – virtuální sériový port
- Mass Storage Class
- Microchip Custom Driver

Pro moji aplikaci se nejvíce hodí CDC, který se jeví jako nejjednodušší možnost týkající se komunikace a vývoje podpůrného programu na vyčítání dat na PC.

5.2 Výběr ZigBee čipu/modulu

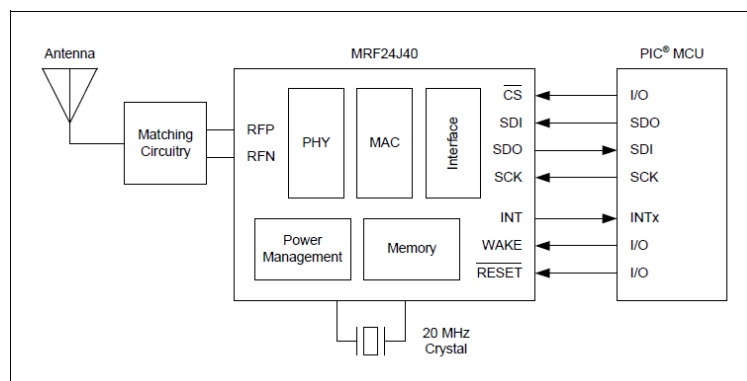
Na trhu se vyskytuje spousta výrobců, kteří vyrábějí jak čipy, tak celé moduly pro ZigBee komunikaci. Patří mezi ně firmy jako Texas Instruments, Amel, Freescale, Microchip a spousta dalších.

Při výběru jsem hleděl na cenu, spotřebu a především na podporu. Jelikož jsem již vybral mikrokontrolér od firmy Microchip, zaměřil jsem se na jejich portfolio i při výběru ZigBee čipu/modulu, a to z důvodu bezproblémového spojení s μC , podpory a především zkušenostmi s produkty od Microchipu.

5.2.1 MRF24J40

MRF24J40 je čip z portfolia společnosti Microchip. Již má v sobě implementovány všechny potřebné náležitosti pro komunikaci pomocí IEEE 802.15.4. Komunikace mezi čipem a μC probíhá pomocí čtyřvodičového sériové periferní rozhraní SPI. Nastavení požadovaných vlastností a akcí se provádějí nastavováním jednotlivých registrů v čipu. Napájení čipu se pohybuje v rozmezí 2,4 V až 3,6 V, avšak výrobce doporučuje 3,3 V. Blokové uspořádání viz **Obrázek 5.2**.

Data lze přenášet rychlostí 250 kbps. Microchip také vyrábí moduly s použitím tohoto čipu. Výhodou modulu je, že se hardwarově nemusí navrhovat vysokofrekvenční přizpůsobení a přepínací obvod mezi vysíláním a přijímáním (tzv. balun). Jedná se o tři různé varianty modulů, viz **Obrázek 5.3 a 5.4**. [7]



Obrázek 5.2: Blokové uspořádání MRF24J40 [7]

5.2.2 MRF24J40MA

Tento modul je první ze tří variant. Modul nese čip MRF24J40, diskrétní součástky pro funkčnost čipu, balun a vysokofrekvenční anténu. Tento model se hodí díky své spotřebě (přijímání 19 mA, vysílání 23 mA a spánek 2 μ A) do bateriově napájených zařízení. S jeho spotřebou souvisí i vysílací výkon, který je dán pouze výkonem výstupní části čipu, tudíž není moc velký. Výrobce uvádí dosah až 120 m na přímou viditelnost. [8]



Obrázek 5.3: MRF24J40MA [13]

5.2.3 MRF24J40MB

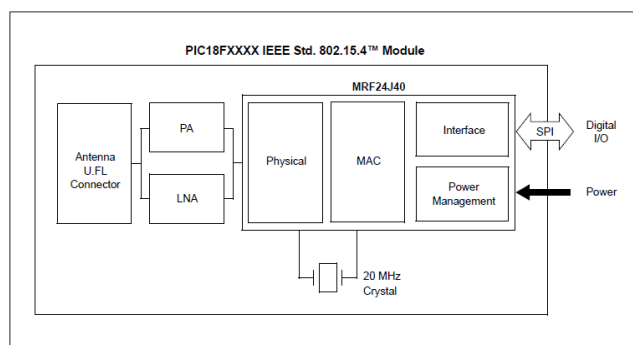
Modul této varianty se od první liší tím, že má zabudovaný zesilovač. Z toho plyne, že spotřeba bude i větší (přijímání 25 mA, vysílání 130 mA a spánek 5 μ A), z toho důvodu i větší dosah, údaj udávaný výrobcem je 1200 m na přímou viditelnost.[9]



Obrázek 5.4: MRF24J40MB (vlevo) a MRF24J40MC (vpravo) [14]

5.2.4 MRF24J40MC

Poslední ze tří variant má stejně jako varianta MB zabudovaný zesilovač, ale jeho výstup není přiveden na interní anténu, zde je zakončena Ultra Miniature Coaxial (U.FL) 50 Ω konektorem pro připojení externí antény, viz **Obrázek 5.5**. Napájecí proudy jsou následující: přijímání 25 mA, vysílání 120 mA a spánek 12 μ A. Dosah uváděný výrobcem je totožný s variantou MB, avšak neuvádí při použití jakého typu antény.[10]



Obrázek 5.5: Vnitřní uspořádání modulu MRF24J40MC [10]

5.2.5 Konečný výběr ZigBee modulu

Jako nejvhodnější kandidát se jeví modul MRF24J40MC. Je vybaven zesilovačem a konektorem pro připojení externí antény. Modul MB není také špatný, ale pro hlavní jednotku je nevhodný z důvodu interní antény, která by mohla být zastíněna při vložení do krabičky.

5.3 Napájení hlavní stanice

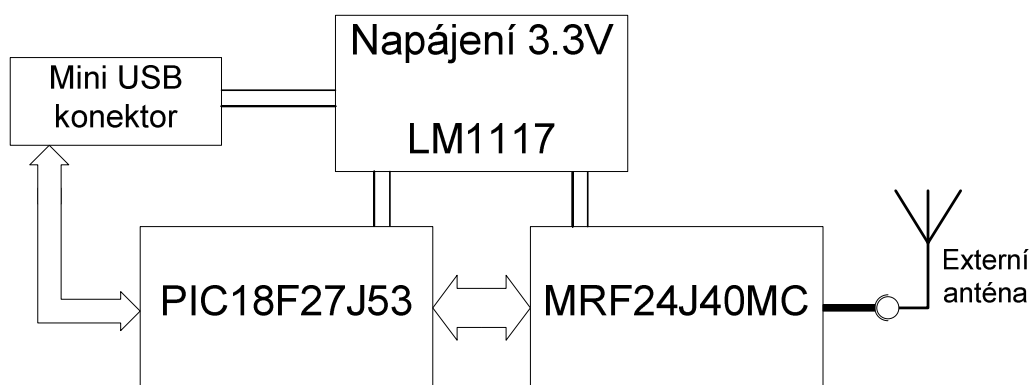
Stanice bude připojena k PC pomocí USB, jež má, jak již bylo zmíněno, výše napětí +5 V a dokáže dodat proud až o hodnotě 500 mA. Napětí 5 V není pro moji aplikaci příliš vhodné, proto je zapotřebí jej upravit. Jelikož stanice bude neustále připojena ke zdroji energie, nemusíme řešit žádné speciální úsporné obvody pro úpravu napětí. Postačí obyčejný lineární regulátor, který na výstupu dodá 3,3 V. Dále je potřeba, aby proud dodaný do obvodu byl větší jako proud potřebný pro chod μ C a ZigBee modulu, tj. přibližně 25 mA pro mikrokontrolér a 120 mA pro ZigBee modul (při vysílání). Pro tuto aplikaci jsem vybral obvod LM1117 od firmy Texas Instruments, jež dokáže dodat do obvodu až 800 mA, což výrazně převyšuje požadavky na proud. [11]

6 Obvodový návrh a fyzická realizace

Veškerý návrh jsem dělal v návrhovém softwaru EAGLE 6.5.0 Light.

6.1 Blokové schéma zapojení

Srdcem celé hlavní jednotky je PIC18F27J53, který komunikuje po USB sběrnici s PC. Také je spojen přes SPI rozhraní s modulem MRF24J40MC, jenž vysílá/přijímá data. ZigBee modul MRF24J40MC disponuje konektorem, ke kterému se dá připojit externí anténa s impedancí 50 Ω . Nejvhodnější pro tuto aplikaci je všesměrová anténa. Mikrokontrolér společně s modulem je napájen přes 3,3 voltový stabilizátor, který získává energii z USB. Viz **Obrázek 6.1**.



Obrázek 6.1: Blokové schéma přijímací stanice

6.2 Obvodové zapojení

Při návrhu obvodového zapojení jsem se řídil doporučených zapojení od výrobců a mými vlastními zkušenostmi získaných během studií.

Jak jsem již uvedl, pro napájení celé stanice je použit obvod LM1117. Tento obvod bere napájení z USB B mini konektoru, přes který je připojen k počítači. Mezi USB konektorem a vstupem stabilizátoru je koráلكový ferit, jenž zabraňuje indukci vzruchů na kabelu, kterým je připojen k počítači. Na vstupu a výstupu stabilizátoru jsem použil tantalové kondenzátory o hodnotě 470 μF a blokační kondenzátory. Mikrokontrolér jsem osadil křemíkovým krystalem o frekvenci 16 MHz. Blokační kondenzátory jsem použil o hodnotě, kterou doporučuje výrobce a zapojil je podle doporučeného zapojení. Vstup MCLR jsem připojil přes odpory R1 a R2 mezi těmito odpory je připojen blokační kondenzátor, jenž je připojen na zem. Dle mojí zkušenosti jsem ještě umístil tlačítko mezi MCLR pin a zem, a to z důvodu, že MCLR slouží jako

hardwarový reset a pro restart je nutné jej přes tlačítko uzemnit. Z tohoto důvodu jsem ještě připojil diodu mezi odpor R2 a vstupní pin MCRL. Na pinu RA0 se nachází uzemňování pro signalizační LED diodu. LED dioda je připojena přes odpor na napájecí napětí. Datové vodiče z USB konektoru jsem přímo připojil na piny D+ a D-. Žádné další součástky nejsou potřeba pro komunikaci USB. Mikrokontrolér má již v sobě zabudovaný modul pro komunikaci po USB sběrnici. Zigbee modul jsem připojil k mikrokontroléru pomocí SPI sběrnice, tato sběrnice má na jasně dané piny SDI, SDO, SCK. Další piny jsem zapojil dle vlastního uvážení. Samozřejmě bylo zapojit blokační kondenzátor mezi napájení a zem. Dále jsem musel počítat s faktem, že se bude muset nějak programovat. Z tohoto důvodu jsem musel vyvést napájení, zem a následující piny MCLR, PGD, PGC na hřebínkové piny. Jaké a na jakou pozici je vyvést, jsem se informoval v manuálu příslušného programátoru, který jsem se rozhodl používat pro naprogramování mikrokontroléru. Pro připojení antény jsem použil propojovací kabel z IPX konektoru (na ZigBee modulu) na SMA konektor, jenž je umístěn na krabičce. Viz **Příloha A – Schéma zapojení**.

6.3 Fyzická realizace prototypu

Při fyzické realizaci jsem postupoval následovně. Nejdříve jsem si v návrhovém softwaru nahrubo poskládal veškeré součástky a odhadl potřebnou velikost plošného spoje. Plošný spoj jsem uvažoval dvouvrstvý. Následně jsem vybral krabičku. Krabičku jsem vybíral podle odhadnutých rozměrů plošného spoje. Vybral jsem krabičku FR-55-25-50-ME od Fischer elektronik. Tato krabička vyhověla svými rozměry a designově se hodila k použité technologii Zigbee. Výrobci propagují žlutou či oranžovou barvu kvůli slovu Bee¹. Dále vyhovovala tím, že je celohliníková kvůli odstínění a také drážkám, které jsou uvnitř pro zaaretování plošného spoje v krabičce.

Jelikož jsem již znal rozměry, mohl jsem navrhnout plošný spoj. Vycházel jsem z předpokladu, že plošný spoj bude dvouvrstvý. Při tvorbě jsem se držel zásad, jako například, aby součástky byly uspořádány, vysokofrekvenční obvody obklopovala měděná vrstva, jež je spojena se zemí, blokační kondenzátory byly co nejbližší součástce, pro kterou jsou. Výsledný návrh plošného spoje viz **Příloha B – Motivy desky plošných spojů**.

¹ Bee v překladu znamená včela

Následně jsem nechal zhotovit tuto desku u firmy PragoBoard , s.r.o., a to jako dvouvrstvou desku s nepájivou maskou na obou stranách, potiskem na vrchní straně a chemickým zlacením. Výroba desky trvala přibližně týden. Viz **Příloha D – vyhotovená deska plošného spoje**.

Poté jsem desku osadil s pomocí vybavení školy, tj. hrotovou mikropájecí stanicí a horkovzdušnou pájecí stanicí. Při celém vývojovém a konstrukčním procesu jsem nezaznamenal žádné potíže.

Takto osazenou desku jsem vložil do krabičky a následně upravil přední panel krabičky pro konektor USB, signalizační diodu a tlačítko reset. Následovalo upravení zadního panelu krabičky, tj. vyvrtání otvoru a usazení SMA konektoru pro externí anténu. Viz **Příloha F – Fotodokumentace**.

7 Způsob komunikace

7.1 Komunikace mezi sběrnou jednotkou a čidly

Na pravidle komunikace jsem se musel domluvit s kolegou, který vyvíjí čidlo. Jelikož jsou čidla náchylná na energetickou spotřebu a po většinu běhu jsou čidla v režimu spánku a mají vypnutý modul na přijímání, nebyl způsob, jakým by hlavní jednotky začínala komunikace s čidly. Proto komunikace navazují čidla a stanice naslouchá. Přijímací stanice většinu času naslouchá, kromě případu, kdy čidlu odesílá informace. Toto je učiněno, aby se mohla přidávat nová čidla za běhu sítě. Pravidla pro komunikaci jsou následující:

7.1.1 Přidávání nového čidla

Nové čidlo se přihlásí do sítě pomocí paketu, jehož uváděcí znak obsahuje požadavek na přidání. Pokud jej stanice zachytí, odešle čidlu potřebná data o nastavení sítě a především, kdy má čidlo vysílat. Poté se čidlo ubere do režimu spánku a je nedostupné.

7.1.2 Standardní vysílání

Po prvotní nastavení čidla je čidlo v režimu standardním, jinak řečeno má nastavené parametry pro komunikaci v síti. Hlavní stanice stále naslouchá, a když čidlo začne vysílat ve svůj daný čas, jeho data přijme, zpracuje a pošle mu zpět potřebná data pro plynulý běh sítě. Za potřebná data se považuje aktuální čas pro synchronizaci, kanál, interval a čas komunikace.

7.1.3 Vypnutí čidla

Dále jsme se dohodli, že bude možno vypnout čidlo na dálku. Pokud se uživatel rozhodne na dálku vypnout čidlo, hlavní stanice čidlu při první možnosti komunikace pošle paket se sekvencí na vypnutí.

7.2 Struktura námi posílaného paketu

Kvůli jednoznačnosti posílaných dat jsme se museli domluvit na společné struktuře odesílaných dat. Námi zvolená struktura je následující. Každý datový paket bude ukončen následující znakovou kombinací */.

Data odesílaná z čidla mají takovou tu strukturu:

Byte	0	1	2	3	4
Druh informace	Stav jednotky	ID jednotky 1/2	ID jednotky 2/2	Typ jednotky	Teplota 1/2
	Nová = 0x31			Teploměr = 0x01	
	Známa = 0x30			Vlhkoměr = 0x02	

5	6	7	8	9	10	11	12	13	14
Teplota 2/2	Vlhkost 1/2	Vlhkost 2/2	rezerva			Stav baterie	Vysílací čas	Interval měření	

15	16	17	18	19	20	21	23	24
Rok	Měsíc	Den	Hodina	Minuta	Sekunda	Vysílací kanál	Ukončovací znaky */	

V takovém to uspořádání se data posílání do hlavní jednotky, zde se tyto data pomocí ukončovacích znaků zkontrolují, posunou se o dvě místa a na nultý bajt se запиše hodnota 0x20. Na první bajt se запиše délka rámce včetně ukončovacích znaků.

Struktura rámce odesílání z počítače do hlavní jednotky vypadají následovně:

Byte	0	1	2	3
Druh informace	Uváděcí znak	Délka rámce	Znak pro nastavení jednotky	ID jednotky 1/2
			Čekej na příjem = 0x42	
			Reset = 0x52	
			Stav jednotky 0x53	
			Nastav jednotku = 0x4E	

4	5	6	7	8	9	10	11	12
ID jednotky 2/2	Vysílací čas	Interval měření	Rok	Měsíc	Den	Hodina	Minuta	Sekunda

13	14	15	16
Kanál, na kterém CHCEME VYSÍLAT	Kanál, na KTERÉM SE VYSÍLÁ	Ukončovací znaky */	

Takto načtený rámeček je v hlavní jednotce překontrolován, pokud proběhne vše v pořádku, nahraje se do odesílací paměti pro Zigbee modul bez prvních dvou bajtů.

Rezerva v rámci je určena do budoucna, kdyby se přidávala další čidla s odlišnou funkcí tak, aby se nemusela měnit nebo přeorganizovat struktura paketu.

8 Firmware pro řídicí jednotku

Pro vývoj firmwaru pro PIC 18f27j53 jsem si zvolil software od firmy MikroElektronika MikroE. V tomto softwaru se programuje pomocí jazyka C#. Jeho velkou výhodou je podpora ve stylu knihoven, tudíž ušetří práci s věcmi, které již před vámi někdo vymyslel a vy je nemusíme objevovat znovu. Využil jsem například knihovny Delay – pro zpoždění, SPI- pro SPI komunikaci, USB – komunikace USB a další. Dále jsem využil již napsaného firmwaru pro jednotky s modulem MR24J40MA. Firmware měl za funkci navázat spojení s jedním přijímačem a odeslat mu hodnotu čítače a tu zobrazit na LED diodách. Z tohoto programu jsem využil počáteční nastavení modulu MRF24J40 a odesílání a přijímání paketů.

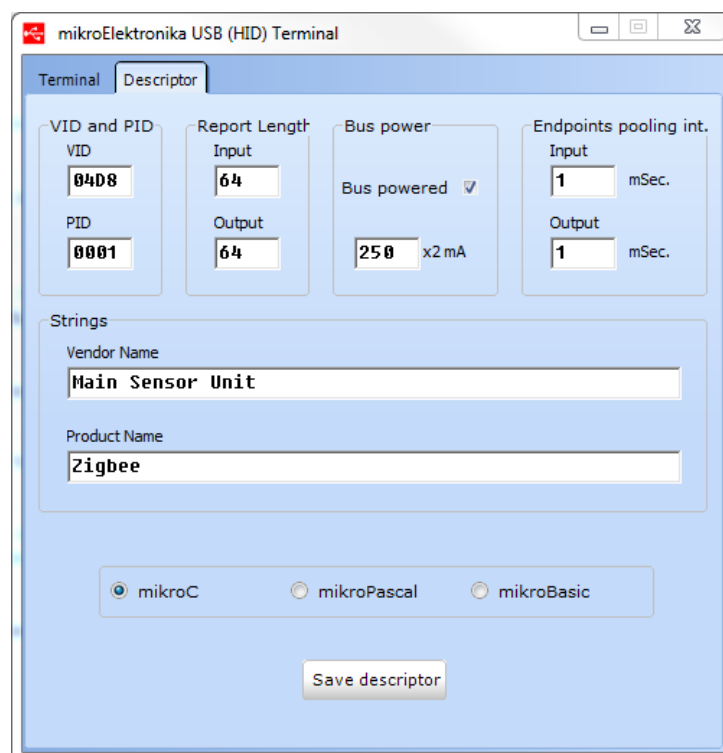
8.1 Firmware pro obsluhu USB

V kapitole 2.6 USB v PIC18F27J53 jsem se zmínil o tom, že pro moji aplikaci se hodí CDC. S touto myšlenkou jsem také začal vyvíjet firmware, avšak po několika dnech jsem došel k názoru, že CDC (virtuální sériový port) není ten správný směr, a to z důvodu následujícího. Při snaze načíst celý poslaný řetězec a porovnat ho se tento úkon nepodařilo provést. Pokaždé se podařilo porovnat jeden znak nebo pouze poslední. Zda byla chyba v mém programu či procedura CDC nedokázala načíst celý řetězec, se mi nepodařilo zjistit. Proto jsem od této způsobu komunikace upustil a nahradil jsem ji HID (Human Interface Device). Nejenže tento způsob dosahuje vyšších přenosových rychlostí, ale po podrobnějším přezkoumání jsem došel k závěru, že i práce s ním bude jednodušší.

8.1.1 *HID firmware*

Pro komunikace pomocí USB sběrnice využívám knihovny v softwaru MicroC. Jednou ze základních procedur je *USB_Interrupt_Proc*, tento proces slouží k obsluze USB při různých událostech a volá se přes přerušení. Další procesy, které využívám, jsou *HID_Read* a *HID_Write*. Pokud jsou data přijata, vrátí procedura *HID_Read* hodnotu rozdílnou od nuly. Přijatá data jsou obsažena v USB bufferu mikrokontroléru. Každá řada PIC má různé adresy, odkud tento buffer začíná. U mého PICu začíná na adrese 0x60. Od této adresy také vyčítám přijatá data. S procesem *HID_Write* se pracuje tak, že mu musíme dát dvě vstupní hodnoty, a ty jsou: pole dat, které chceme

poslat, délka pole dat. Tyto funkce využívám v hlavním programu, ovšem aby komunikace po USB sběrnici fungovala, musí projekt obsahovat ještě soubor *USBdsc.c*. Tento soubor se dá vygenerovat pomocí nástroje HID terminál (v softwaru MicroC) v záložce *deskriptor*, viz **Obrázek 8.1**. Zde je potřeba nastavit VID a PID, které chceme, aby zařízení mělo, Vendor a Product Name a také délka paketu. Dále se zde dá nastavit maximální odebíraný proud, v mém případě je nastaveno 500 mA. Je zřejmé, že soubor *USBdsc.c* bude obsahovat tyto náležitosti, které jsou pro inicializaci USB komunikaci nezbytné. Pro VID hodnotu jsem použil sublicencovanou hodnotu od Microchipu 0x04D8. PID hodnou jsem si zvolil 0x0001.



Obrázek 8.1: HID terminál společně s tabulkou generující *USBdsc.c* soubor

8.2 Firmware pro obsluhu Zigbee modulu

8.2.1 *SPI komunikace*

Jak jsem výše zmínil, Zigbee modul komunikuje přes SPI sběrnici. Pro SPI komunikaci jsem využil knihovny SPI. Zde jsou tři důležité procesy, jeden z nich je *SPI_Int*, ten slouží pro prvotní inicializace sběrnice. Dále je tu *SPI_Read* a *SPI_Write*.

Například pro zápis:

```
void write_short(short int address, short int data_r)
{
    CS = 0;
    address = ((address << 1) & 0b01111111) | 0x01; // Maska pro zápis
    SPII_Write(address); // adresovaný registr
    SPII_Write(data_r); // zapisovaná data
    CS = 1;
}
```

8.2.2 Inicializace Zigbee

Prvotní inicializace probíhá následujícím způsobem, a to tak že se modul nejprve softwarově a hardwarově resetuje. Prvním krokem je nastavení zapnutí modulu. To se provádí následovně: nastaví se polarita probouzacího signálu a povolí se I/O pin pro probuzení a do registru *WAKECON(0x22<7>)* se zapíše 1. Druhým krokem je nastavení adresy jednotky, a to jak dlouhé, tak krátké, také je zapotřebí nastavit PAN ID. Následujícím krokem je nastavení samotných vlastností modulu. Po důmyslném zvážení jsem se rozhodl, že komunikace bude probíhat pomocí nonbeacon komunikace, tj. nebude komunikovat pomocí super rámce, ale běžným rámcem. Inicializace pro nonbeacon komunikace je následující:

- nastaví se potřebné registry, např. pro správu sítě, nastavení oscilátoru atd. (podrobnější popis nastavení registru viz příloha **Firmware**)
- vybere se jeden ze tří módů CCA metody, v mém případě mód 1.
- vybere se jeden ze dvou možných způsobu měření RRSI, já využívám druhý způsob, a to při přijímání dat modul vypočte RSSI a zařadí je na konec paketu posílaný do mikrokontroléru.
- povolí se přerušení
- nastaví se vysílací kanál
- provede se reset jednotky (reset je nutno provádět při každé změně kanálu)

Příklad používaného kódu pro výběr RSSI:

```
void set_RSSI_mode(short int RSSI_mode)
{
    short int temp = 0;
    switch(RSSI_mode)
    {
        case 1: {
            temp = read_ZIGBEE_short(BBREG6);
            temp = temp | 0x80; // 0x80 maska pro RSSI1 mód
            write_ZIGBEE_short(BBREG6, temp);
        }
        break;
        case 2:
            write_ZIGBEE_short(BBREG6, 0x40); // 0x40
            hodnota pro RSSI2 mode
            break;
    }
}
```

Předchozí proces se dá použít pro všechny zařízení jak pro koordinátory, směrovače, tak pro samotná čidla.

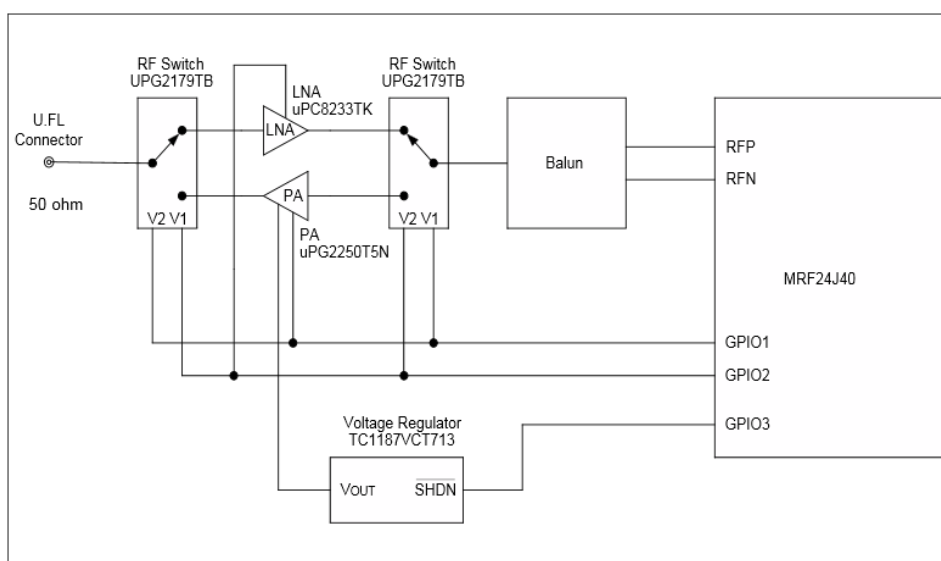
Dále je nutno jednotku nadefinovat jako koordinátor sítě a to následovně:

```
void nonbeacon_coordinator_device()
{
    short int temp = 0;
    temp = read_ZIGBEE_short(RXMCR);
    temp = temp | 0x04; // 0x04 maska pro koordinátor
    write_ZIGBEE_short(RXMCR, temp); // zápis hodnoty
    temp = read_ZIGBEE_short(TXMCR);
    temp = temp & 0xDF; // 0xDF pro nastavení CSMA-CA módu
    write_ZIGBEE_short(TXMCR, temp);
    write_ZIGBEE_short(ORDER, 0xFF); // nastavení hodnot BO, SO na 15
}
```

Po tomto kódu je zařízení nastaveno jako koordinátor sítě.

8.2.3 Nastavení vysílacího výkonu

Je nutné nastavit vysílací výkon. Ten se dá regulovat od -26,4 dBm až do 18,5 dBm, tato hodnota se zapisuje do registru $RFCON3(0x203<7:3>)$. V mém případě jsem výkon nastavil na maximální možný. Zde jsem narazil na problém, a to v podobě nízkého vysílacího výkonu, přestože jsem nastavil vysílací výkon na maximum. Vysílací jednotka měla pouze dosah cca 2 metry. Po podrobnějším prostudování datasheetů od všech třech modulů MRF24J40 a různých diskuzích na internetu jsem došel k závěru, že je vypnut zesilovač PA, který zesiluje výstupní signál. Jelikož jsem vycházel z programu napsaného pro MRF24J40MA, který nemá žádný zesilovač, tak kód neobsahoval potřebné nastavení registrů pro zapnutí zesilovače. Jak je vidět z **Obrázku 8.2**, tak je zesilovač napájen z regulátoru napětí, jenž spíná binární výstup GPIO3. Aby zesilovač fungoval, je nutné sepnout tento regulátor. Pouze modul verze MC využívá GPIO3, ani verze MB, která má také zesilovač nevyužívá GPIO3, dokonce ani nový model MD/ME nevyužívá regulátor a je ovládán přímo z GPIO0. Sepnutí regulátoru, tj. aktivace GPIO3 se dělá následovně. Po zapsání hodnoty vysílacího výkonu do registru $RFCON3(0x203)$ je nutné zapsat hodnotu 0x0F do registru $TESTMODE(0x22F)$, dále na čtvrtý bit registru $TRISGPIO(0x34)$ zapsat 1 (GPIO3 jako výstupní). Poslední krok je nastavení hodnoty 1 na GPIO3, to se dělá pomocí $GPIO(0x33<3>)$. Po těchto úkonech je regulátor sepnut a výstupní zesilovač zapnut.



Obrázek 8.2: Vnitřní zapojení výstupní části MRF24J40MC

8.2.4 Nastavení filtru a přijímacího módu

Poslednímu kroky při inicializaci jsou nastavení filtru a přijímacího módu. Filtr má čtyři možné varianty: všechny data, instrukční data, užitečná data, bacen frame. Já jsem zvolil variantu první, tj. všechny data. Přijímací mód má tři varianty:

- normální – akceptuje pouze pakety s dobrým CRC splňující požadavky IEEE 802.15.4
- error – akceptuje pakety buď s dobrým, nebo špatným CRC
- volný – akceptuje všechny pakety

V mém případě jsem zvolil první variantu, tj. normální.

Po všech těchto úkonech se na WAKE pin nastaví 1, vyčká 5 milisekund a zařízení je připraveno přijímat a odesílat.

8.2.5 Přijímání a odesílání

Odesílání se provádí pomocí procedury `write_TX_normal_FIFO`. Tato procedura má vstupní parametr délku posílaných dat. Jak jdou data za sebou, popisuje **Obrázek 8.3**. Adresy, hodnoty, data se postupně přiřazují do pole, které se poté pošle na jasně určenou adresu pro normální, tj. nezabezpečený přenos. Adresa pro tento přenos začíná na 0x00 a končí na 0x07F (128 bajtů). Délka hlavičky a délka dat musí být v přijímači a vysílači nastavena stejně, jinak se data nepřenesou správně!

0	1	2	3	4	5	6
Délka hlavičky	Délka hlavičky + délka dat	0x01	0x88	sekvenční číslo	Cílové PAN ID1/2	Cílové PAN ID 2/2
7	8	9	10	11	12	n
Adresa přijímače 1/2	Adresa přijímače 2/2	Místní PAN ID1/2	Místní PAN ID 2/2	Adresa zdroje 1/2	Adresa zdroje 2/2	Data

Obrázek 8.3: Pořadí dat v rámci

Poté co jsou data přesunuta z mikrokontroléru do paměti modulu, následuje nastavení, zda požadujeme potvrzovací rámec či ne. V mém případě jsem zvolil postup, aby přijímač posílal potvrzovací rámec, dále jsem nastavil procedurou `set_no_encrypt`, aby rámec nebyl kódován. V tuto chvíli je možné odeslat paket, a to tak, že do registru `TXCON(0x1B<0>)` se zapíše 1, tím se paket začne posílat. Dále se už o nic nestaráme,

všechny potřebná náležitosti si modul MRF24J40MC obstarává sám. Po úspěšném odeslání se hodnota v *TXCON(0x1B<0>)* automaticky vynuluje.

Přijíatá data se načítají pomocí procedury *read_RX_FIFO*, která vrací hodnotu délky přijatých dat. Ale nejdříve je potřeba, aby mikrokontrolér věděl, že má data z paměti modulu vyčítat. To se provádí pomocí přerušení. Pin INT z modulu je připojen na vstup RB0, zde je softwarově testováno, zda nenastalo přerušení. Toto přerušení se dá nastavit v modulu pro různé činnosti, v mém případě je nastaveno pouze na přijímání dat. Poté co modul přijme data, oznámí to přes INT a v tuto chvíli je volána procedura *read_RX_FIFO*. Po zavolání této procedury se pošle hodna pro zastavení přijímání a vyčtou se přijatá data, s těmi se dále pracuje. V tomto procesu se také vyčítá hodnota RSSI a LQI. Vyčítá se od adresy 0x300 do 0x38F. Všechna data a hodnoty se uloží do pole znaků *DATA_RX*. S tímto polem se dále pracuje. Po načtení všech dat se znovu pošle do modulu příkaz, který odblokuje příjem a modul je chopen znovu přijímat data.

8.3 Kompletní firmware přijímací jednotky

Předchozí části jsem musel spojit dohromady, aby vytvořily celkový firmware a jednotka pracovala správně. Zvolil jsem cestu přes přerušení, a to z důvodu neperiodicity toku dat od čidel a z počítače. Přerušení se jeví pro tento problém jako nejvhodnější možnost. Nejdříve se vykoná jedna funkce a mezitím může přijít další požadavek, který se zapamatuje a po skončení děje se plynule přejde na další děj, který byl vyvolán od dalšího zdroje. Například se může stát, že budeme přidávat nové čidlo do sítě a zároveň nám bude posílat stávající čidlo sítě informace. Při této situaci se nejdříve pošlou informace pro nové čidlo a po skončení této procedury se plynule přejde k vyčtení a následnému zpracování dat, které nám poslalo stávající čidlo.

Při programování jsem narazil na problém s přístupem k procedurám z funkce *main* a z *interrupt*, proto jsem musel veškerý kód obsáhnout pouze ve funkci *interrupt*. Ve funkci *main* musím uměle dávat flagbit od INT2, který fyzicky nevyužívám, do hodnoty 1, Ten vyvolá přerušení, ve kterém se provede prvotní inicializace a nastavení modulu. Takto jsem obešel problém s dvojitým přístupem k procedurám. Dále mám ve funkci *main* proceduru *HID_Enable*, která se musí volat před prvotním užíváním USB komunikace. Dále je zde jen nekonečný cyklus. Zbylý kód je obsažen

ve funkci *interrupt*. Funkce *interrupt* reaguje na tři přerušení a to od USB, softwarově přerušení od modulu při příjmu dat a uměle vytvořeným INT2 pro prvotní inicializaci. Prvotní inicializace už není zapotřebí dále rozebírat je rozebrána v kapitole **7.2.2 Inicializace Zigbee**. Ted se soustředím na přerušení od modulu, tj. přijímání dat.

Pokud přijde přerušení od Zigbee modulu přečte se hodnota registru INSTAT z modulu, tím se tento registr vynuluje, dále se zavolá výše zmíněná funkce *read_RX_FIFO*. Do USB bufferu se nahrají úvodní znaky pro komunikace se softwarem, za těmito znaky následují data, která se pomocí cyklu *for* nahrají do bufferu. Po této činnosti následuje odeslání dat do počítače pomocí USB sběrnice. Následně se musí vynulovat flag bit, který toto přerušení způsobil.

Třetím a posledním přerušením je přerušení od USB, zde přerušení nastává, pokud jsou přijata nějaká data. Nejprve se zavolá procedura *USB_Interrupt_Pro*, následuje podmínka pro čtení a ta, že pokud je proces *HID_Read()* rozdílný od, nuly začnou se porovnávat přijatá data, ty se zpracují a podle hodnot se provedou následující činnosti. Pokud je přijata hodnota:

0x20 – nejprve se porovná kanál, na kterém chceme vysílat s kanálem aktuálním a popřípadě se provede změna kanálu. Následuje nahrání dat do DATA_TX a zavolání procedury *write_TX_normal_FIFO*, která odešle data patřičnému zařízení.

0x21 – zde se provádí pouze přeladění na jiný kanál a to z důvodu, aby se dalo přidat nové čidlo, které vysílá na defaultním kanálu.

9 Podpůrný software pro PC

Podpůrný program pro sběr dat jsem vyvíjel v programu Visual Studio 2013 od společnosti Microsoft. Tento program jsem si vybral z toho důvodu, že lidé, kteří v mém okolí programují, používají tento program a pochvalují si ho. Rozhodl jsem se pro formulářovou aplikaci psanou v jazyce C#. Jelikož jsem s tímto vývojovým prostředím dělal poprvé, začátky byly velmi krušné, nakonec jsem všechny problémy zdolal a vytvořil jsem funkční program pro sběr dat z čidel.

9.1 Obsluha USB

Abych byl schopný načítat data z jednotky, musel jsem se porozhlédnout po nástroji, který by to umožňoval. Po chvilce brouzdání na internetu jsem našel nástroj *microgeniosUsbHID_port*. Na internetu bylo i instruktážní video jak s tímto nástrojem pracovat. Nejprve je nutné vědět VID a PID svého zařízení, se kterým chcete pracovat. Po aktivaci formuláře by se toto zařízení mělo načíst, proto příkaz s VID a PID je nutno umístit právě zde. Nástroj obsahuje procedury pro zjištění stavu zařízení (připojeno nebo odpojeno) i za běhu, dále pro čtení a zápis. Po přijetí dat se s těmito daty dále pracuje. V prvním kroku se zkontroluje, zda přijatý paket je uváděn hodnotou 0x20 a následuje kontrola délky. Dále se volá procedura, která dále s těmito daty pracuje.

Ukázka kódu pro kontrolu přijatých dat.

```
if (DataRead[0] == 0x20)
{
    int delka_paketu = (int)DataRead[1];
    if (DataRead[delka_paketu-2] == '*')
    {
        if (DataRead[delka_paketu-1] == '/')
        {
            Zpracovani_dat(DataRead);
        }
    }
}
```

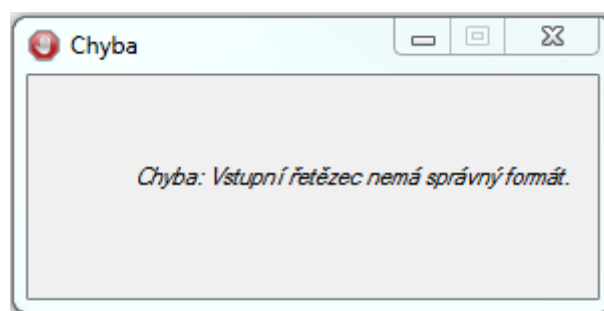
9.2 Vzhled hlavního formuláře

Hlavní formulář se skládá ze tří základních částí. V horní části je menu, vedle menu je popisek, který nám signalizuje připojené či odpojené zařízení. Dále je zde tlačítko pro smazání čidla. V pravého horním rohu je aktuální čas. V horní polovině

formuláře je tabulka, ve které se zobrazují aktuální čidla s posledními naměřenými hodnoty. Ve spodní polovině se nachází tabulka s naměřenými daty pro uživatelem zvolené čidlo.

9.2.1 Vytvoření nového nebo načtení stávajícího souboru

Pokud se uživatel rozhodne založit nové měření, nejprve si musí zvolit, na jakém kanálu chce, aby síť komunikovala. Dále se vytvoří textový soubor se jménem, jak ho uživatel pojmenuje, je na něm. V tomto souboru jsou obsaženy všechny cesty k potřebným datům. Při vytvoření nového měření se vygenerují dva textové soubory a jejich cesta se uloží do již výše zmíněného textového souboru. Jeden z vygenerovaných souborů je *nazevuživateleID.txt*, v tomto souboru se ukládají použitá ID čísla přidávaných jednotek, aby nevznikla kolize dvou stejných ID, číslo je možno zadávat od hodnoty 01 do 98. hodnota 00 je rezervována pro defaultně nastavená čidla, která žádají o přiřazení do sítě. Hodnota 99 je rezervována pro hlavní stanici. Druhým vygenerovaným souborem je *nazevuživateleINT.txt*, zde se ukládají použité časy vysílání, aby nedošlo, ke kolizi, a ztrátě dat při příjmu. Ve skutečnosti ani ke ztrátě dat z čidel dojít nemůže z důvodu CSMA/CA přístupu. Další možnost je načíst data z již předešlých měření, zde stačí vybrat kořenový textový soubor, z něhož si program vyčte potřebné odkazy na data. Při načítání dat z již existujícího měření se načtou do tabulky poslední změřené hodnoty čidel. Veškeré procesy pro načítání a práci se soubory jsou ošetřeny, a pokud dojde k nějakým potížím, program o tom uživatele upozorní, viz Obrázek 9.1.



Obrázek 9.1: Chybová hláška při potížích s načítáním souboru

9.2.2 Přidání nového čidla

Každou půl minutu vyše program příkaz pro přeladění na defaultní kanál, kde se po dobu dvou sekund naslouchá. Periodicitu nám zajišťuje nástroj *Timer*. Přidání nové čidla se hlásí kódem 0x31 na třetím bajtu. Pokud se tato hodnota přečte, následují procesy pro přidání nového čidla. Nejdříve se čidlu odešle signál, že byl jeho požadavek přijat, to způsobí, že čidlo již nevysílá a čeká na naše instrukce. Dále se načte soubor *nazevuzivateleID.txt*, ze kterého se zjistí použitá ID. Procesy zjistí, jaké nejnižší volné číslo je možno použít a tuto hodnotu také odešle do přidávacího formuláře, kde se zobrazí.

Ukázka kódu pro výběr nejnižšího možného ID.

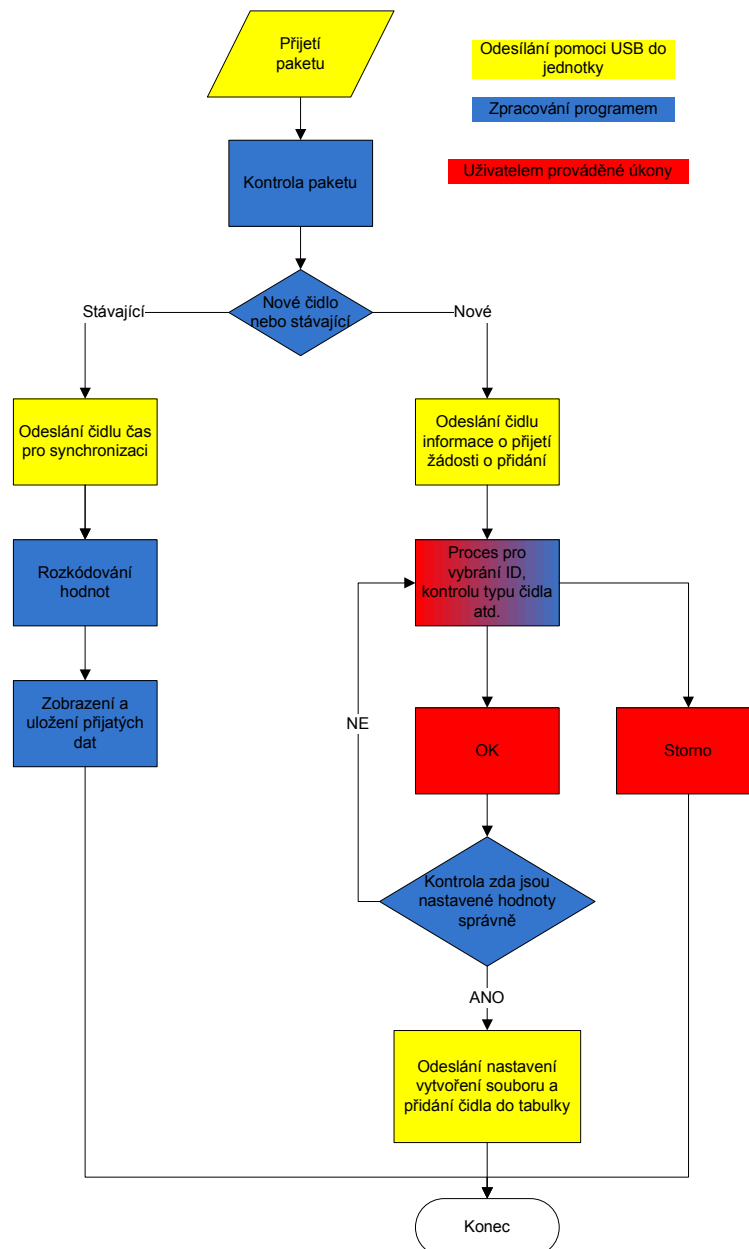
```
for (int cisloj = 1; cisloj < 100; cisloj++)
{
    string ID_cislo = "";
    if (cisloj < 10)
    {
        ID_cislo += "0"; // kvůli dvojnakovému ID
    }
    ID_cislo += cisloj;
    if (sLine == ID_cislo)
    {
        pouzita_ID[cisloj] = true; pokud je číslo v seznamu označí se jako true
        cisloj = 100;
    }
}
```

Toto číslo společně s údaji, co nám poslalo čidlo, tj. druh čidla, jsou tyto hodnoty předány druhému formuláři, který se zobrazí pro přidání nového čidla. Díky druhému formuláři může hlavní program běžet dál a tak pádem přijímat a vyhodnocovat data od již známých čidel a tak pádem si nové čidlo můžeme přidat v klidu. Formulář pro přidání čidla obsahuje *TextBox* pro jeho pojmenování, *MaskedTextBox*, kde je zadána maska 00, která udává, že zadaná hodnota musí být dvouznaková a pouze číslice. Dále je zde *Label*, který zobrazuje funkci čidla, tento text mu předává první formulář, poslední částí je zde nástroj *NumericUpDown*. Zde si uživatel nastavuje, po jakém intervalu bude čidlo odesílat informace. Pokud uživatel nevyplní kolonku pro název čidla, program ho nepustí dál a nahlásí chybu. Kolonka s ID číslem je již předvyplněna od programu, který nabízí nejnižší volnou hodnotu. Tu si uživatel může změnit, pokud ji zadá špatně, nebo zadá již obsazené ID, program nahlásí chybu. Popis druhu čidla je pouze informativní a uživatel jej nemůže měnit. Interval začíná na 1 minutě s krokem 1 minuta. Maximální možný interval je nastaven na 10 minut. Pokud je vše v pořádku a uživatel zmáčkne tlačítko OK, zobrazí se uživateli dialogové

okno pro uložení souboru, do kterého budou zapisována data z čidla. Soubor si uživatel může pojmenovat podle svého. Po uložení se zavře dialogové a formulářové okno a do ID souboru se zapíše použité ID, do INT souboru čas vysílání. Do hlavního souboru se zapíše cesta k souboru pro ukládání hodnot z měření. Nový soubor, který se vytvoří pro ukládání dat z jednotky, obsahuje hlavičku, kterou generuje program. Jeho struktura je vidět na **Obrázku 9.2**. Do jednotky se odešlou informace pro nastavení čidla. Informace, jež se odesílají do jednotky, jsou následující: ID, kanál, interval měření, čas vysílání a především aktuální čas pro synchronizaci. Do hlavního formuláře se předají zpět informace o čidlu a přidá se řádek do tabulky čidel. Ovšem pokud uživatel zmáčkne Storno, formulář se zavře a čidlo se nepřidá a odešle se příkaz čidlu, aby se uspalo. Celý rozhodovací proces přidání čidla včetně zpracování dat je vyobrazen na **Obrázku 9.3**.

```
01|01|Teplotní čidlo|2
Teplota [°C] |Vlhkost [%]|RSSI [dB]|Kvalita signálu|Stav baterie [%]|Interval [min]|Frekvence|čas |datum
```

Obrázek 9.2: Vygenerovaná hlavička pro nové čidlo



Obrázek 9.3 : Proces pro zpracování přijatých dat

9.2.3 Zpracování přijatých dat

Po přijmutí paketu a následné kontrole se odešle čidlu paket s vysílacím kanálem, periodou měření a především s aktuálním časem pro synchronizaci. Tyto informace si odešlou a následuje proces zpracování dat. Hodnoty jako změřená teplota, čas a datum, síla a kvalita signálu potřebují převod. Změřená teplota se posílá tak, jak se změřila. Její rozkódování se provádí až v počítači, a to z důvodu úspory místa v datovém rámci. Hodnoty síly signálu se převádí z hexadecimální hodnoty na hodnotu dBm, kvalita signálu se převádí na slovní hodnotu jako dobrá či špatná. Čas a datum potřebují konverzi, jelikož jsou ve formátu pro mikrokontrolér a jeho

RTCC. Po těchto konverzích a úpravách se data uloží do souboru k příslušnému čidlu a jejich hodnota se zobrazí v tabulce v hlavním formuláři.

Kód pro převod dvoubajtové hodnoty teploty na číslo.

```
if ((data[7] & (0xF0)) >= 1)
{
teplota += "-"; //pokud je teplota záporná

data[7] = (byte)(int)~data[7];
data[6] = (byte)(int)(~data[6]+1
}
double setiny = ((data[6] & 0x0F) * 625); //vyčtení setin
data[7] &= 0x07; // bitová maska pro spodní čtveřici bitů
data[7] <<= 4; //bitový posuv
data[6] >>= 4;
data[6] &= 0x0F;
int desetiny =( data[6] | data[7] ); // vyčtení desítek
double tepl = (double)desetiny + (setiny / 10000);
teplota += tepl; // uložení teploty do stringu
```

Data, která se ukládají do souboru, se dávají do jednoho řetězce, který se poté zapíše do souboru, jednotlivá data jsou od sebe oddělena znakem „|“. Tento znak jsem zvolil kvůli tomu, aby se s daty dalo pracovat například v Excelu nebo jiných tabulkových procesorech.

9.2.4 Smazání čidla

Pokud se uživatel rozhodne smazat čidlo, může to udělat pomocí tlačítka smazat, nebo v menu nástroje smazat čidlo, a to tak, že vybere v tabulce řádek s čidlem, které chce smazat a klikne na jednu z předešlých možností. Proces, který obsahuje program, nejprve načte název a ID čidla z tabulky. Poté načte hlavní soubor a do mezipaměti si uloží jednotlivé odkazy na data, ty začne postupně otevírat a srovnávat hlavičky souborů s načtenými daty z tabulky, pokud nastane shoda, smaže odkaz na tyto dat v hlavním souboru, poté přicházejí na řadu ID. Načtené ID se porovnává s ID.txt souborem, a když dojde ke shodě, toto ID se smaže. Stejně probíhá proces pro mazání času odesílání v souboru INT. Následně se čidlu odešle příkaz pro restart a uspání. Tento způsob jsem zvolil, aby se zachovala data z měření pro pozdější zpracování.

9.2.5 Grafy

Aplikace je schopna zobrazovat graf závislosti teploty na čase, kdy hodnota byla pořízena. Grafy se vyobrazují pomocí nástroje Chart. Tento nástroj složí pro zobrazení grafů. V mém programu nástroj bere data z uložených dat v textovém

souboru. Nejprve uživatel vybere řádek s čidlem, které chce zobrazit a dvakrát na ni poklepe myší. Proces pro načtení je podobný jako u funkce mazání. Nejprve se načte hlavní soubor s odkazy na data a poté se porovnávají hlavičky jednotlivých souborů s daty vyčtenými z řádku tabulky. Poté co dojde ke shodě, se data ze souboru vyčtou a zobrazí v grafu. Po každém dvojkliku se data z grafu odstraní a nahradí se uživatelem vybranými daty, tj. zobrazený průběh je pouze jeden.

9.2.6 Manuál pro obsluhu programu

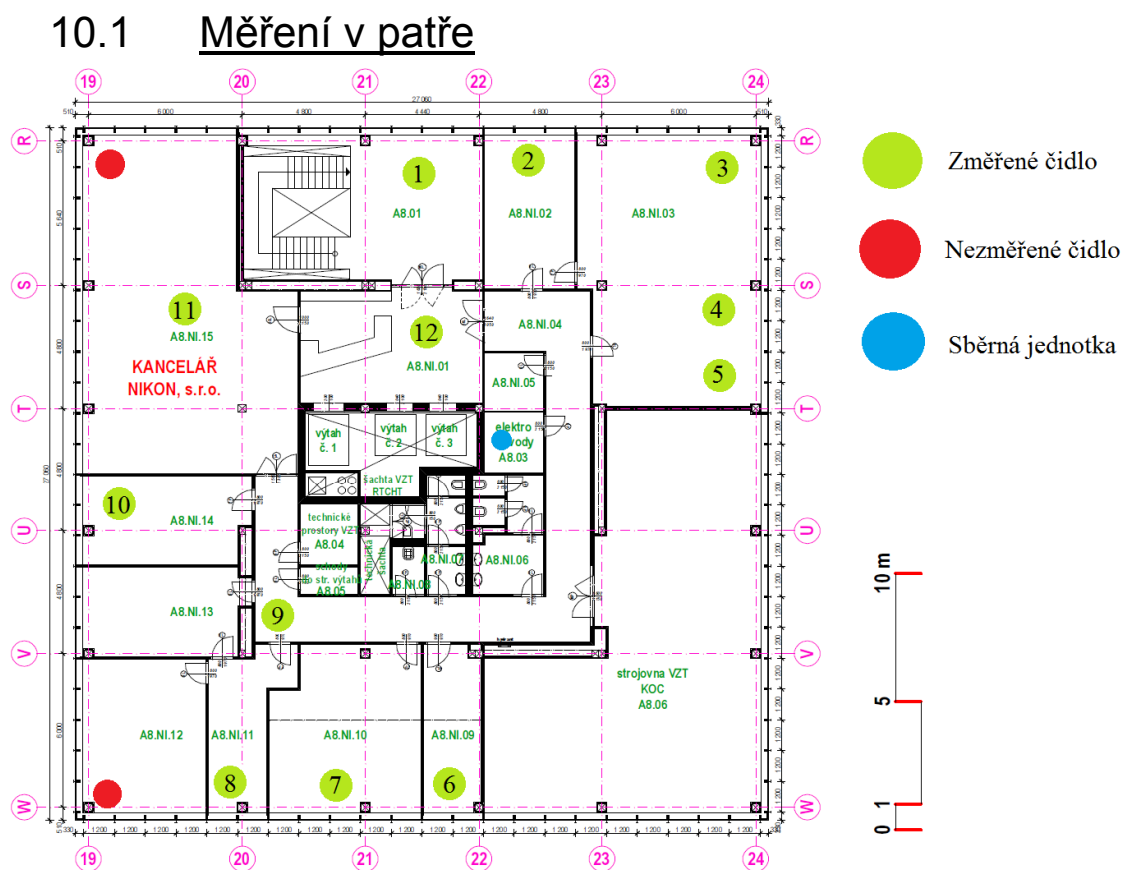
Napsal jsem jednoduchý obslužný manuál k programu, aby obsluha věděla, jak má pracovat s programem. V tomto manuálu jsou napsány postupy, jak zakládat či otevírat měření a také, jak přidávat a nastavovat čidla. Samozřejmě nechybí popis, jak pracovat s přijatými daty. Manuál k nahlédnutí viz **Příloha G - Návod pro podpůrný program.**

9.2.7 Kapacita čidel

Program, jenž jsem vytvořil, má možnou kapacitu 28 čidel. To je učiněno tím, že každé čidlo má přiděleno vysílací čas, ve kterém vysílá. Čidla mají mezi sebou časový rozestup 2 sekundy, aby nedošlo ke kolizi. Dále je potřeba, aby jednotka přelad'ovala na defaultní kanál, na jež vysílají čidla, která žádají o přidání. Toto přelad'ování se provádí dvakrát za minutu a jeden interval trvá dvě sekundy.

10 Měření dosahu čidel

Měření jsme prováděli v budově Palác Kodaňská Office Center, kde by se později tato sensorová síť mohla nasazovat. Měřili jsme v budově A v 8. patře. Budova je železobetonová, ovšem příčky mezi kanceláři jsou sádkartonové o tloušťce 20 mm. Hlavní stanice byla umístěna v technické místnosti, která je skoro ve středu budovy. Hlavní stanice byla osazena všesměrovou anténou o zisku 2 dBi. Měřili jsme sílu a kvalitu přijímaného signálu od čidla. A to z toho důvodu, že čidlo má mnohem menší vysílací výkon než hlavní stanice. Naměřené hodnoty jsme brali od hlavní stanice z rámce, který je posílán do počítače. Měření si stanice provádí sama, tudíž jsme do jejího měření nemohli zasahovat.



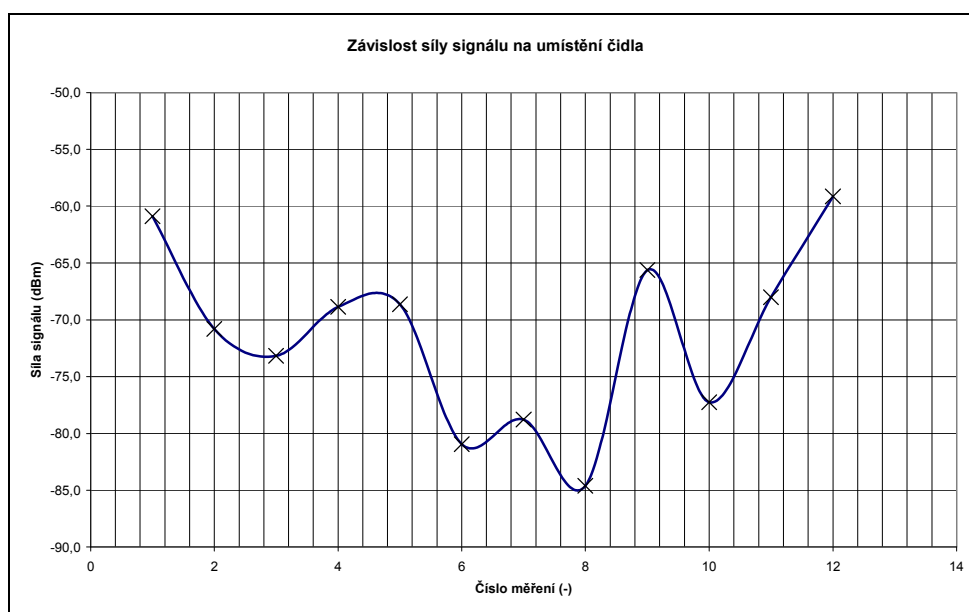
Obrázek 10.1: Plán budovy, ve které proběhlo měření

Na **Obrázku 10.1** jsou umístěné barevné tečky s čísly. Zeleně vyznačené polohy čidel se podařilo změřit. Červeně vyznačené jsou polohy čidel, které se nepodařilo změřit. Síla signálu se měří v rozsahu od 0 (nejhorší kvalita) do 255(nejlepší kvalita).

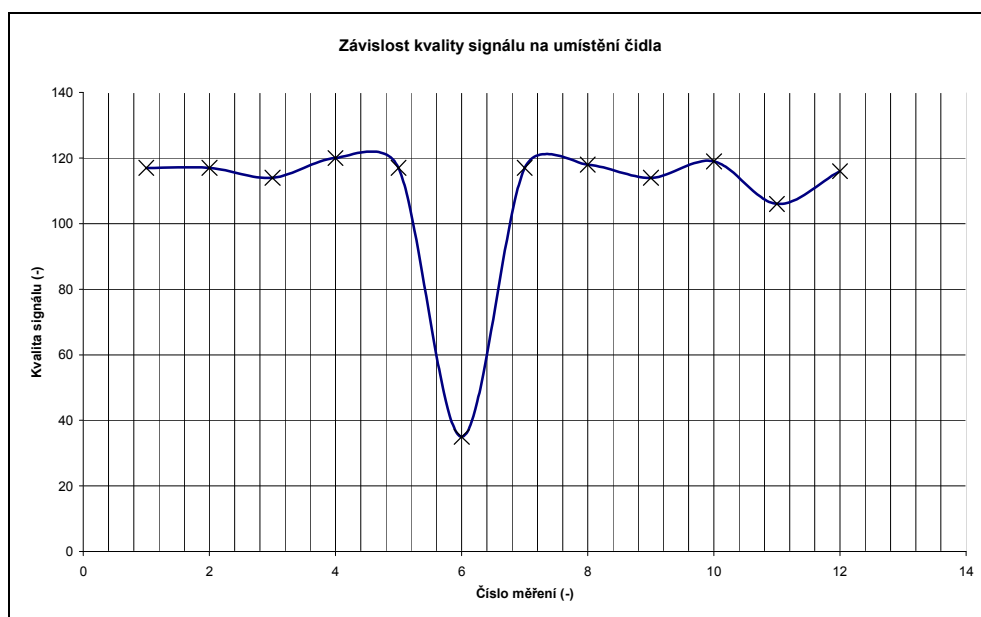
Tabulka 10.1 Měření pro podlaží

Měření v budově A 8. podlaží		
Číslo měření	Síla signálu (-)	Kvalita signálu (dBm)
1	117	-60,9
2	117	-70,8
3	114	-73,2
4	120	-68,9
5	117	-68,6
6	35	-80,9
7	117	-78,8
8	118	-84,6
9	114	-65,6
10	119	-77,3
11	106	-68,0
12	116	-59,2

Změřené hodnoty zanesené v **Tabulce 10.1** jsou přepočítány z hexadecimálních hodnot, které obsahoval přijatý paket. Přijatý paket, který byl přijat měl strukturu, kterou čidlo používá pro vyslání požadavku o přijetí do sítě.



Obrázek 10.2 : Graf síly signálu pro podlaží



Obrázek 10.3 : Graf kvality signálu pro podlaží

Z grafů je vidět, že signál se vzdáleností klesá, viz **Obrázek 10.2 a 10.3**. Výkyv v kvalitě signálu může být způsoben tím, že vedle místnosti, ve které se nacházela pozice 6, je strojovna vzduchotechniky a mohlo zde docházet k mnohačetným odrazům od plechové konstrukce vzduchotechniky.

10.2 Měření skrz patra

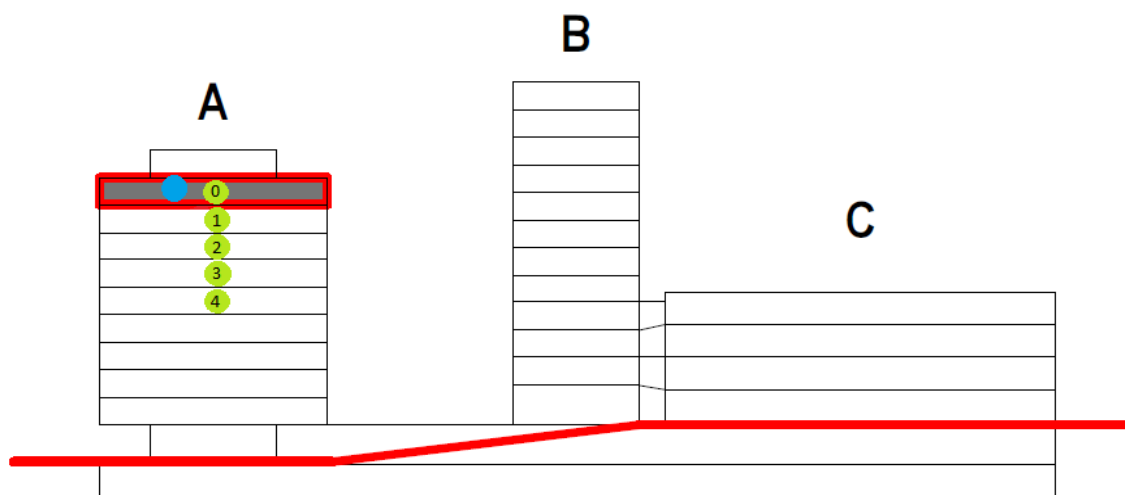


schéma řezu

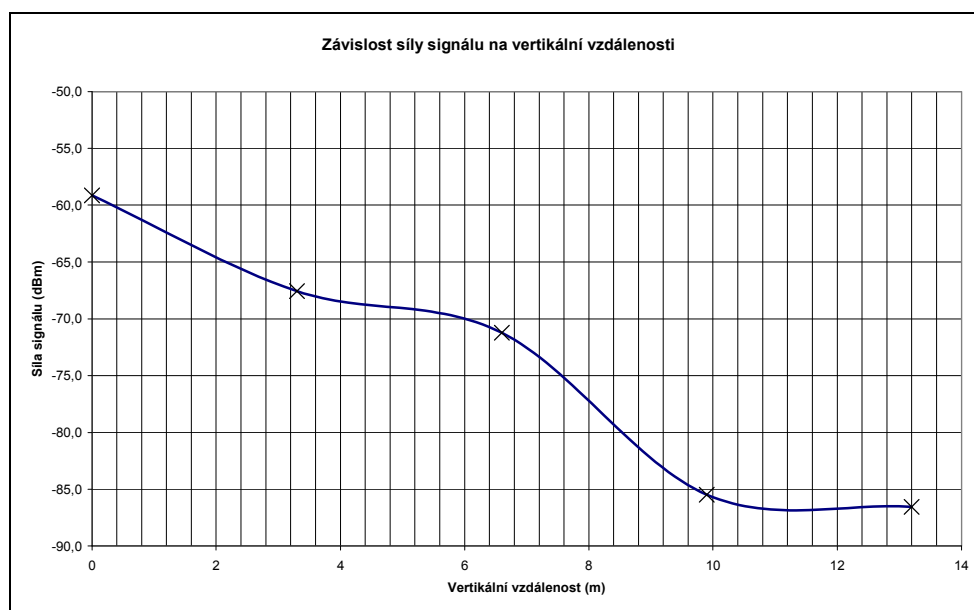
Obrázek 10.4: Plán řezu budovy, ve které proběhlo měření

Dále jsme měření prováděli přes patra, viz **Obrázek 10.4**. Jelikož jsme byli v nejvyšším patře, mohli jsme jít pouze dolů. V každém patře jsme provedli jedno měření, a to v místech mezi sektory **21, 22 a S, T** (viz **obrázek 10.1: plánec budovy**).

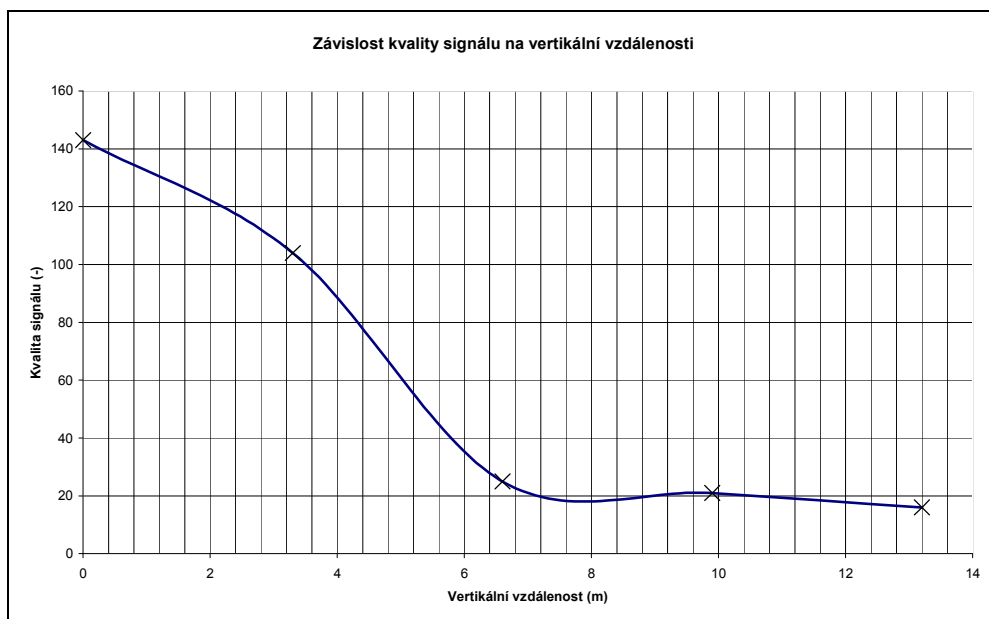
Tabulka 10.2: Hodnoty pro vertikální měření

Měření v budově A skrz patra					
Číslo měření	Výška patra	Rozdíl výšky	Patro	Kvalita signálu	Síla signálu
	(m)	(m)	(-)	(-)	(dBm)
0	23,1	0	8	143	-59,2
1	19,8	3,3	7	104	-67,6
2	16,5	6,6	6	25	-71,2
3	13,2	9,9	5	21	-85,5
4	9,9	13,2	4	16	-86,5

Z **Tabulky 10.2** je patrné, že maximální změřená vzdálenost přes patra je cca 13,2 m (v určité výšce byla anténa jednotky). V nižších patrech jsme již signál nezachytili.



Obrázek 10.5: Graf kvality signálu pro vertikální vzdálenost



Obrázek 10.6: Graf síly signálu pro vertikální vzdálenost

Z grafů je vidět, že síla téměř lineárně klesá se vzdáleností až do 10 metrů, poté je hodnota konstantní a velmi nízká, tudíž záleží na náhodě, zda se signál přijme. Kvalita signálu je ve srovnání se silou signálu horší. Její hodnoty celkem dosti klesají. To bude nejspíše způsobeno železobetonovou konstrukcí. Viz **Obrázek 10.5 a 10.6**.

10.3 Měření na volném prostranství

Dále jsme měřili dosah čidel na volném prostranství, tj. na přímou viditelnost. Měřili jsme za slunného dne v odpoledních hodinách při teplotě cca 20 °C. Tentokrát jsme neměřily více poloh a nesestavovali jsme grafy v závislosti na vzdálenosti, ale změřili jsme maximální vzdálenost, při které přijímač zachytí data od čidla. Výsledek z tohoto měření je, že na volném prostranství je dosah signálu 85-90 m.

10.4 Závěr z měření

Ze změřených dat jsme usoudili, že by tato senzorická síť v těchto podmínkách obstála. Dosažené hodnoty vzdálenosti celkem odpovídají hodnotám uváděných v technických listech modulů. Pro zlepšení příjmu, aby se pokrylo kompletně celé patro, by pomohla anténa s větším ziskem a lepší umístění jednotky. Co se týče vertikálního dosahu, tak předčil naše očekávání.

11 Ekonomické zhodnocení

Ekonomická stránka je nedílnou součástí vývoje, obzvlášť v této době, kdy každý hledí nejprve na cenu. V uvedených cenách není uveden poplatek za dopravu. V **Tabulce 11.1** jsou uvedeny všechny ceny za použité součástky.

Výroba jedné desky plošného spoje vyjde cca na 200 Kč. Celková část za materiál činí 983,05 Kč. V této ceně není zahrnuta cena antény a cena přípojovacího kabelu.

Tabulka 11.1: Ceny jednotlivých součástek

Označení	Hodnota	Cena (Kč)
Tantalový kondenzátor	470 μ F/6,3V	22,52
Tantalový kondenzátor	470 μ F/6,3 V	22,52
Keramický kondenzátor	100 nF	0,17
Keramický kondenzátor	100 nF	0,17
Keramický kondenzátor	18 pF	0,25
Keramický kondenzátor	100 nF	0,17
Tantalový kondenzátor	18 pF	0,25
Keramický kondenzátor	10 μ F/ 6,3 V	5,74
Keramický kondenzátor	100 nF	0,17
Keramický kondenzátor	100 nF	0,17
Keramický kondenzátor	100 nF	0,17
Dioda pínací	1N4148	0,44
Feritový korálek	-	3,8
PIC18F27J53	-	74,64
LM1117	-	12,18
Hřebínkový konektor 5x	-	1,06
LED	SMD zelená	2,69
MRF24J40MC	-	343,19
Krystal	16 Mhz	12,97
Rezistor	10 k Ω	0,21
Rezistor	470 Ω	0,21
Rezistor	150 Ω	0,21
Spínač	1 polohový spínač	4,9
Konektor	Mini-USB	27,56
Krabička	FR-55-35-50-ME	224,11
Světlovod	-	22,58
	Celkem	783,05

Tabulka 11.2 porovnává ceny podobných výrobků na trhu jako je má přijímací stanice. Některé ceny jsou přepočítány z amerického dolaru na koruny, při aktuálním

kurzu 20 Kč za \$. Jak je vidět, vlastní vývoj vyšel levněji než nabízené produkty kromě USBStick od společnosti Digi, kde se ceny shodují. Ovšem v cenách výrobců je zahrnuta i podpora produktu a příslušný software. Přesto je podle mého názoru cena za mnou vyvíjený prototyp uspokojivá.

Tabulka 11.2: Porovnání cen produktů na trhu s prototypem mé jednotky

Výrobce	popis jednotky	Cena (Kč)
Legrand	Ovládací jednotka s dotykovým panelem	12317
Ravica	Hlavní jednotka do zásuvky na SD kartu či USB	20839
Digi	USB Stick	980
Mnou vyvíjená jednotka přes USB		983

12 Závěr

Závěrem své bakalářské práce bych chtěl zhodnotit dosažené výsledky. Zadané požadavky byly splněny ve všech směrech. Díky vhodně zvolené kombinaci obvodového zapojení jsem dosáhl pozitivních výsledků v oblasti konstrukce, a to díky podpoře komponentů, jež jsme v zapojení použil. To se také promítlo do softwarové problematiky, kde mi tato podpora ulehčila mnoho práce. Desénový vzhled práce je obstojný a dle mého názoru i zdařilý. Změřené hodnoty dosahu signálu jsou uspokojivé a v praxi použitelné. Co se týče ekonomického zhodnocení, tak jsem byl mile překvapen, že cena za prototyp byla příznivá s porovnáním s komerčními produkty.

Měření, které bylo prováděno v budově Palác Kodaňská Office Center, nebylo zvoleno bezdůvodně. V budoucnu je možnost zde tuto síť nasadit do praxe, a to i s centrálním velínem pro celou budovu.

Co se týče firmwaru jednotky, zde bylo možné některé věci vylepšit. Například program, jenž je v počítači a ovládá tuto jednotku, by se dal nahrát do této jednotky a jednotka by byla naprosto samostatná. Také by se dala rozšířit sběrnice. A to o RS 485 s možností Modbus protokolu pro připojení k PLC. Dále by se jednotka mohla obohatit o připojení k internetu a přes web server a webový přístup ukládat data do databází, aby měl uživatel online přístup.

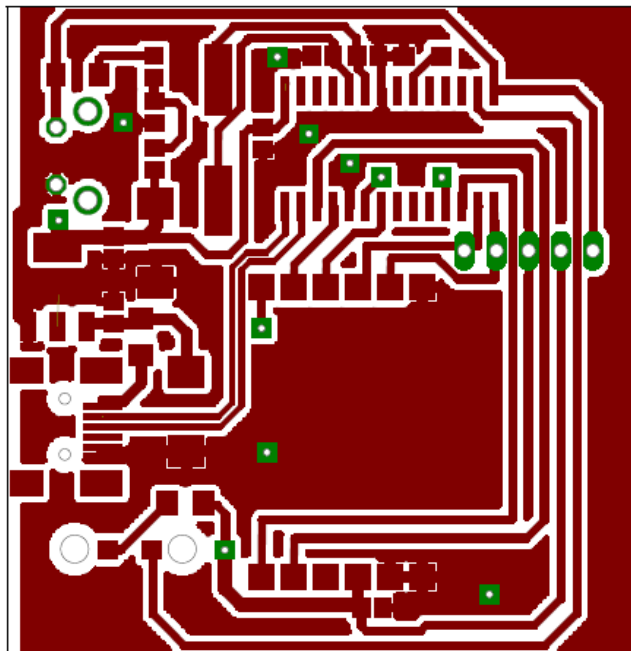
Software v počítači by se dal také zefektivnit. Například pomocí redukce časů k přístupu vysílání, které přiřazuje jednotka čidlům. Zde by se dala diference mezi dvěma vysílacími časy zmenšit, tím by se dalo přidat více čidel do sítě.

Použitá literatura a zdroje

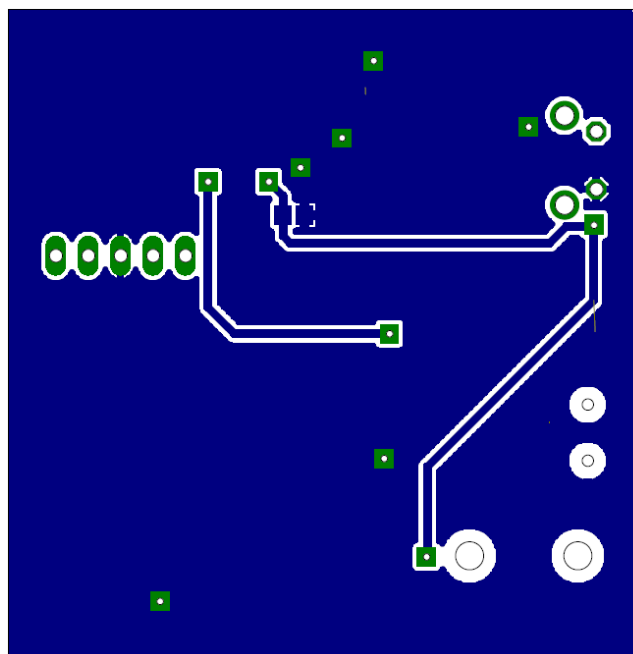
- [1] – Data sheet PIC 18F27J53 dostupný z:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39964B.pdf>
- [2] – TIŠNOVSKÝ, Pavel. Komunikační protokol universální sériové sběrnice. ROOT.CZ. *Root.cz: Periferie* [online]. 22. 1. 2009 0:00 [cit. 2014-02-01]. Dostupné z: <http://www.root.cz/clanky/komunikacni-protokol-universalni-seriove-sbornice/#ic=serial-box&icc=text-title>
- [3] – TIŠNOVSKÝ, Pavel. Přenos dat po universální sériové sběrnici. ROOT.CZ. *Root.cz: Periferie* [online]. 29. 1. 2009 0:00 [cit. 2014-02-01]. Dostupné z: <http://www.root.cz/clanky/prenos-dat-po-universalni-seriove-sbornici/#ic=serial-box&icc=text-title>
- [4] – KAINKA, Burkhard. *USB. Měření, řízení a regulace pomocí sběrnice USB*. 1. vyd. Praha: BEN, 2002, 247 s. ISBN 80-730-0073-3.
- [5] – VOJÁČEK, Antonín. ZigBee - novinka na poli bezdrátové komunikace. *HW server s.r.o.: Rozhraní* [online]. 8. Červen 2005 [cit. 2014-02-01]. Dostupné z: <http://www.hw.cz/navrh-obvodu/rozhrani/zigbee-novinka-na-poli-bezdratove-komunikace.html>
- [6] – ZigBee Specification - Document 053474r17 Dostupné z: <http://www.zigbee.org/>
- [7] – Data sheet MRF24J40 dostupný z:
<http://ww1.microchip.com/downloads/en/DeviceDoc/39776C.pdf>
- [8] – Data sheet MRF24J40MA dostupný z:
<http://ww1.microchip.com/downloads/en/DeviceDoc/70329b.pdf>
- [9] – Data sheet MRF24J40MB dostupný z:
<http://ww1.microchip.com/downloads/en/DeviceDoc/70599B.pdf>
- [10] – Data sheet MRF24J40MC dostupný z:
<http://ww1.microchip.com/downloads/en/DeviceDoc/75002A.pdf>
- [11] – Data sheet LM1117 dostupný z: <http://www.ti.com/lit/ds/symlink/lm1117-n.pdf>
- [12] – Microchip; Dostupné z: <https://www.microchip.com/pagehandler/en-us/family/8bit/architecture/pic18.html>
- [13] – Katalogový list Silaca dostupný z: <http://www.silica.com/product/mrf24j40xx-transceiver-module.html>
- [14] – Katalogový list TME dostupný z: http://www.tme.eu/html/EN/microchip-wireless-communication-modules/ramka_6994_EN_pelny.html

- [15] ZELINKA, Tomáš a Miroslav SVÍTEK. *Telekomunikační řešení pro informačnsystémy síťových odvětví*. 1. vyd. Praha: Grada, 2009, 218 s. ISBN 978-80-247j-3232-9.
- [16] Technický katalog Céliane Dostupný z:
http://www.legrand.cz/userfiles/file/katalogy/byvanie/Celiane_katalog_2012.pdf
- [17] Nabídka Ravica dostupná z: <http://store.ravica.com/searchresults.asp?cat=108>
- [18] XBee® Drop-in Networking Accessories User's Guide Dostupný z:
http://ftpl.digi.com/support/documentation/90000891_G.pdf
- [19] Prezentace Mikrokontroléry dostupná z:
http://www.umel.feec.vutbr.cz/bdom/predn/Mikrokontrolery_1.pdf
- [20] JMATOUŠEK, David. *C pro mikrokontroléry PIC: práce s PIC18F452 a PIC18F1220 v jazyce C*. 1. vyd. Praha: BEN - technická literatura, 2011, 367 s. μ C. ISBN 978-80-7300-413-2.

Příloha B – Motivy desky plošných spojů

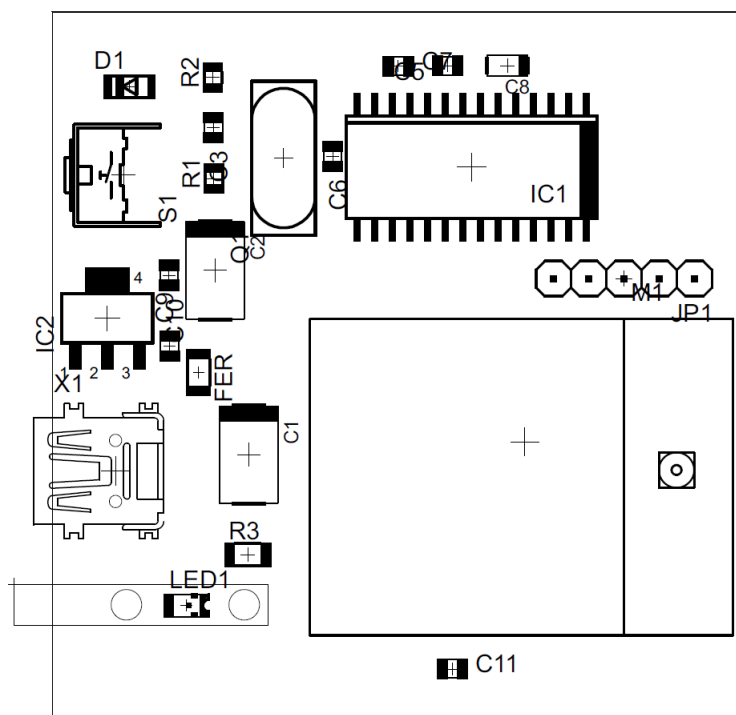


Obrázek B.1: Návrh plošného spoje - vrchní strana

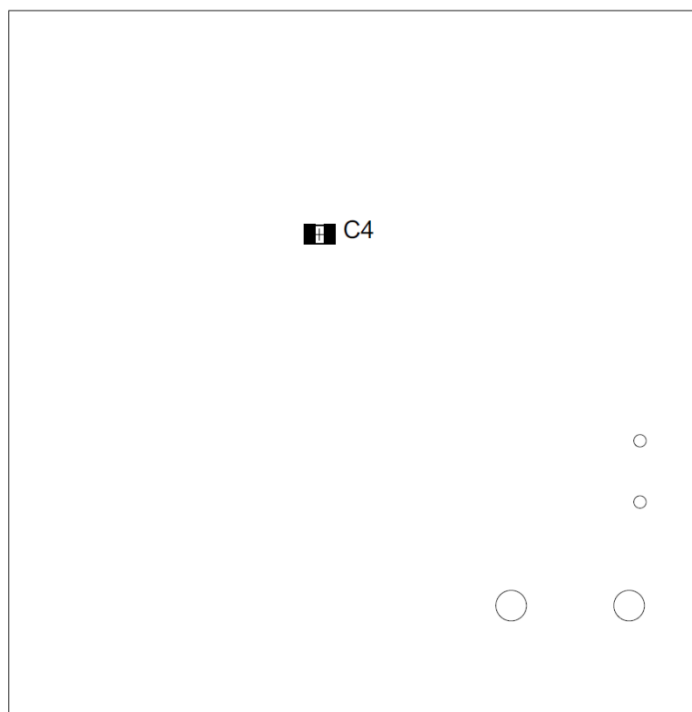


Obrázek B.2: Návrh plošného spoje - spodní strana

Příloha C – Osazovací výkres

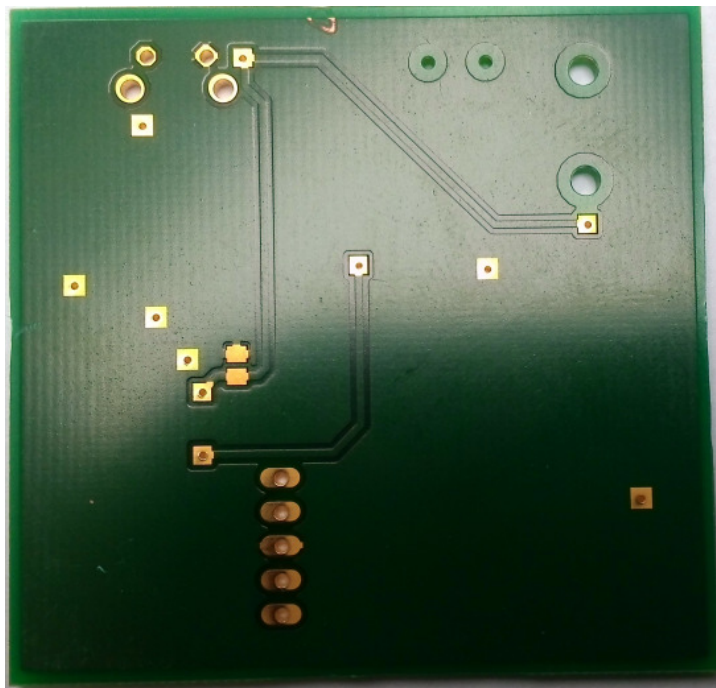


Obrázek C.1: Osazovací výkres - spodní strana

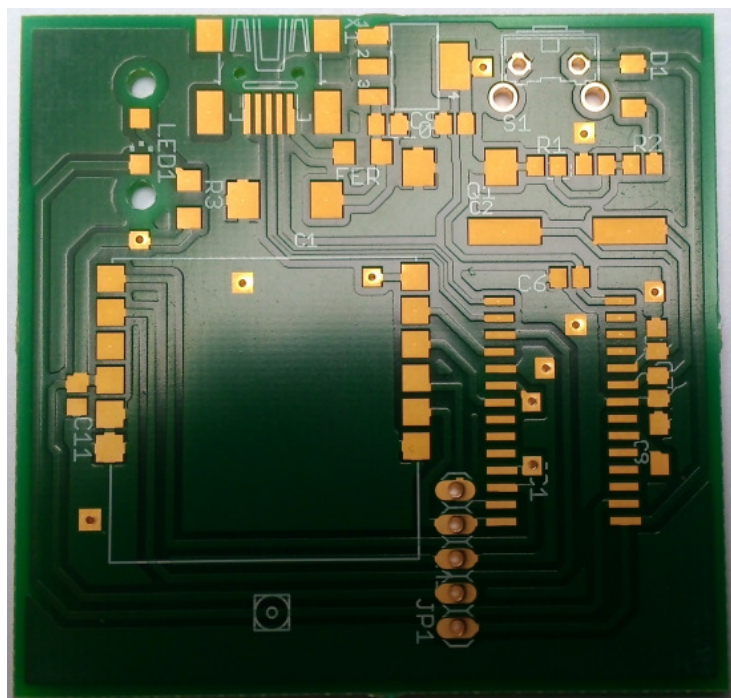


Obrázek C.2 Osazovací výkres - spodní strana

Příloha D – Vyhotovená deska plošných spojů



Obrázek D.1: Zhotovená deska plošného spoje - vrchní strana



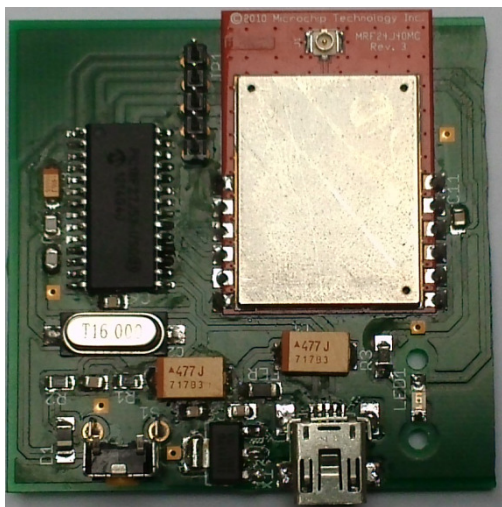
Obrázek D.2: Zhotovená deska plošného spoje - spodní strana

Příloha E – Seznam použitých součástek

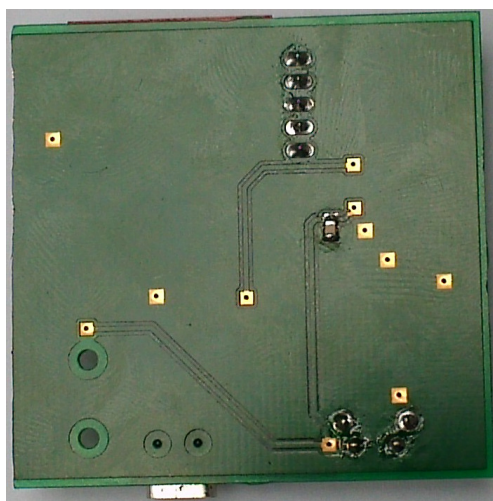
Tabulka E.1: Seznam použitých součástek

Označení	Hodnota
C1	470 μ F/6,3V
C2	470 μ F/6,3 V
C3	100 nF
C4	100 nF
C5	18 pF
C6	100 nF
C7	18 pF
C8	10 μ F/ 6,3 V
C9	100 nF
C10	100 nF
C11	100 nF
D1	1N4148
FER	Feritový korálek
IC1	PIC18F27J53
IC2	LM1117
JP1	Hřebínkový konektor 5x
LED1	SMD zelená
M1	MRF24J40MC
Q1	16 MHz
R1	10 k Ω
R2	470 Ω
R3	150 Ω
S1	1 polohový spínač
X1	Mini-USB

Příloha F – Fotodokumentace.



Obrázek F.1: Zhotovený prototyp - vrchní strana



Obrázek F.2: Zhotovený prototyp - spodní strana



Obrázek F.2: Finální podoba



Obrázek F.4: Finální podoba včetně kabelu