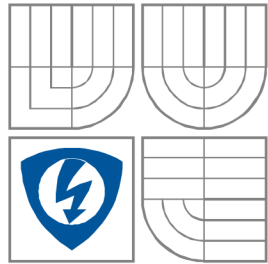


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

SYSTÉM BEZDRÁTOVÉHO PŘENOSU DAT ZE SNÍMAČE Z OTOČNÉ MĚŘICÍ STOLICE

THE SYSTEM OF WIRELESS TRANSMISSION OF DATA FROM SENSORS ON THE ROTARY
TABLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL BADIN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. DAVID SKULA

BRNO 2011

ZADÁNÍ PRÁCE:

Cílem práce je navrhnout systém pro vyčítání dat ze snímače a jejich bezdrátový přenos do nadřazeného systému (PC) pro vyhodnocení. Snímač bude napájen bateriově a umístěn na rotační stoličce. Důraz bude kladen především na komunikační rychlost a malé rozměry zařízení. PC nebude od snímače vzdáleno více než několik metrů. Ke komunikaci se snímačem bude využito mikroprocesoru ATMEGA a komunikačního rozhraní SPI.

Abstrakt

Tato práce obsahuje teoretický i praktický popis bezdrátového komunikačního systému užívajícího bezdrátové komunikační moduly XBee. Práce obsahuje teoretické informace o jednotlivých komponentech zařízení, návrh zařízení a příklad softwarové i hardwarové realizace komunikačního zařízení.

Klíčová slova

Bezdrátová komunikace, komunikační moduly XBee, mikrokontrolér ATMEGA, komunikace po sériovém portu PC

Abstract

In this thesis is a theoretical and a practical description of wireless communication device using wireless communication device XBee. The work contains theoretical information about individual components of the communication device, concept of a communication device and an example of software and hardware implementation of the communication device.

Keywords

Wireless communication, communication device XBee, microcontroller ATMEGA, communication using serial PC port

Bibliografická citace:

BADIN, Pavel. *Systém bezdrátového přenosu dat ze snímače z otočné měřicí stolice*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 38s. Vedoucí bakalářské práce byl Ing. David Skula.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Systém bezdrátového přenosu dat ze snímače z otočné měřicí stolice jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si 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.

V Brně dne: 23. května 2011

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Davidu Skulovi za ochotu a dobrý přístup ke konzultacím, účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 23. května 2011

.....
podpis autora

OBSAH

1. ÚVOD	- 1 -
1.1 Cíl práce.....	- 1 -
1.2 Popis kapitol	- 1 -
2. MIKROKONTROLÉRY ATMEGA	- 2 -
2.1 Základní informace o mikrokontrolérech ATMEGA.....	- 2 -
2.2 Popis jednotlivých částí mikrokontroléru.....	- 2 -
2.3 Popis vstupně výstupních obvodů	- 3 -
2.4 Programovací jazyky pro mikrokontroléry	- 4 -
2.5 Programování paměti mikrokontroléru	- 4 -
3. BEZDRÁTOVÁ KOMUNIKACE	- 5 -
3.1 Druhy bezdrátové komunikace.....	- 5 -
3.2 Modulace signálu.....	- 5 -
3.3 Druhy komunikačních modulů	- 6 -
3.3.1 ZigBee	- 6 -
3.3.2 Bluetooth.....	- 7 -
3.3.3 Wi-Fi	- 7 -
3.3.4 Jiné řešení:.....	- 7 -
3.3.5 Výběr komunikačního modulu.....	- 8 -
4. MODUL XBEE	- 10 -
4.1 Protokol IEEE 802.15.4.....	- 10 -
4.2 Jak pracuje modul XBee.....	- 10 -
4.2.1 Posílání dat	- 10 -
4.2.2 Přijímání dat	- 11 -
4.2.3 Přesun dat mezi mikrokontrolérem a modulem XBee	- 11 -
4.2.4 Command mód	- 12 -
4.2.5 Rozhraní sériového protokolu	- 12 -
4.2.6 Dělení stavů modulu XBee	- 13 -
4.2.7 Pracovní režimy modulu XBee	- 13 -
4.3 Vlastnosti sítě modulu XBee	- 14 -
4.3.1 Charakteristiky sítě DigiMesh.....	- 14 -
4.3.2 Přenos dat	- 14 -
4.3.3 Topologie sítě modulu XBee	- 15 -

4.3.4 Modely sítě modulu XBee	- 16 -
5. ZPŮSOBY A ZABEZPEČENÍ SPRÁVNÉHO PŘENOSU DAT	- 17 -
5.1 Druhy přenosu dat	- 17 -
5.2 Způsoby digitálního přenosu dat	- 17 -
5.2.1 Bitové kódování	- 17 -
5.2.2 Jiné druhy kódování	- 18 -
6. ŘEŠENÍ PROJEKTU	- 20 -
6.1 Teoretický popis řešení.....	- 20 -
6.2 Požadavky na komunikační zařízení	- 21 -
6.3 Zabezpečení správného přenosu dat.....	- 21 -
6.4 Komunikace s moduly Xbee	- 21 -
6.5 Komunikační zařízení na otočné měřicí stoličce	- 22 -
6.5.1 Hardwarové řešení	- 22 -
6.5.2 Softwarové řešení	- 23 -
6.6 Komunikační zařízení připojené k PC.....	- 26 -
6.6.1 Hardwarové řešení	- 26 -
6.6.2 Softwarové řešení	- 26 -
6.7 Ověření funkce bezdrátového přenosu dat	- 28 -
6.8 Jak by se dalo v práci pokračovat.....	- 28 -
ZÁVĚR	- 29 -
LITERATURA.....	- 30 -
SEZNAM PŘÍLOH	- 32 -

SEZNAM OBRÁZKŮ

Obrázek 1 - Architektura mikrokontroléru AVR [1]	- 3 -
Obrázek 2 - Amplitudová modulace [14]	- 6 -
Obrázek 3 - Frekvenční modulace [15]	- 6 -
Obrázek 4 - Model ISO-OSI protokolu XBee [8]	- 10 -
Obrázek 5 - Schéma toku dat uvnitř modulu XBee [4]	- 11 -
Obrázek 6 - Schéma UART přenosu dat [4]	- 12 -
Obrázek 7 - Schéma sítě modulu XBee [4]	- 15 -
Obrázek 8 - Topologie sítě modulu XBee [8]	- 16 -
Obrázek 9 - příklad výpočtu CRC kódu [24]	- 19 -
Obrázek 10 - Schéma funkce zařízení z měřicí stolice	- 20 -
Obrázek 11 - Schéma funkce zařízení připojeného k PC	- 20 -
Obrázek 12: Schéma zapojení komunikačního zařízení	- 23 -
Obrázek 13: Vývojový diagram programu mikrokontroléru	- 24 -
Obrázek 14: Schéma zapojení komunikačního zařízení	- 26 -
Obrázek 15: Ukázka aplikace pro ovládání bezdrátového systému	- 27 -
Obrázek 16: Osazovací plán zařízení na straně PC	- 36 -
Obrázek 17: Osazovací plán zařízení na straně senzoru	- 36 -
Obrázek 18: DPS komunikačního zařízení na straně PC	- 37 -
Obrázek 19: DPS komunikačního zařízení na straně senzoru	- 37 -
Obrázek 20: Fotografie komunikačního zařízení na straně PC	- 38 -
Obrázek 21: Fotografie komunikačního zařízení na straně senzoru	- 38 -

SEZNAM TABULEK

Tabulka 1: Přehled komunikačních modulů [19], [20], [21], [22], [23]..... - 8 -

Tabulka 2: Ukázka vytvoření náhradních dat.....- 25 -

Tabulka 3: Ukázka vytvoření původních dat- 25 -

1. ÚVOD

1.1 CÍL PRÁCE

Cílem práce je vytvořit systém bezdrátového přenosu dat ze snímače z otočné měřicí stolice. Snímač bude napájen bateriově a umístěn na rotační stolici.

Část komunikačního zařízení bude umístěna na otočné měřicí stolici. Tato část by měla mít co nejmenší rozměry. Další požadavek na tuto komunikační část je malá spotřeba. Při malé spotřebě můžeme použít baterii menších rozměrů a nebudeme ji muset často měnit, což nám ušetří provozní náklady. Dále je třeba zajistit aby komunikační zařízení mělo co nejvyšší komunikační rychlost z důvodu rychlého přenosu dat ze snímače do počítače. Na část komunikačního zařízení umístěného u počítače jsou kladeny prakticky stejné nároky, ale tato část bude napájena z USB portu.

Pro ovládání vzdáleného mikrokontroléru a komunikace po sériovém portu bude vytvořena knihovna funkcí obsluhující komunikaci po sériovém portu PC. Také bude vytvořena vizuální aplikace pro testování a ladění softwaru a hardwaru.

1.2 POPIS KAPITOL

Druhá kapitola je věnována základním rysům mikrokontrolérů ATMEGA.

V třetí kapitole jsou shrnuty základní druhy bezdrátové komunikace. Poté je zde popsán výběr základních komunikačních modulů, které bychom mohli použít pro bezdrátovou komunikaci v této práci. V závěru této kapitoly je provedeno srovnání vybraných komunikačních modulů a výběr toho nejefektivnějšího komunikačního modulu pro naši práci.

Ve čtvrté kapitole je popsán standard Zigbee a jsou zde uvedeny základní vlastnosti modulu XBee.

Pátá kapitola je zaměřena na zajištění správného přenosu dat.

Šestá kapitola se zabývá řešením projektu, shrnuje výsledky projektu a popisuje možné další vylepšení této práce.

2. MIKROKONTROLÉRY ATMEGA

Tato kapitola používá zdroje [1], [2].

2.1 ZÁKLADNÍ INFORMACE O MIKROKONTROLÉRECH ATMEGA

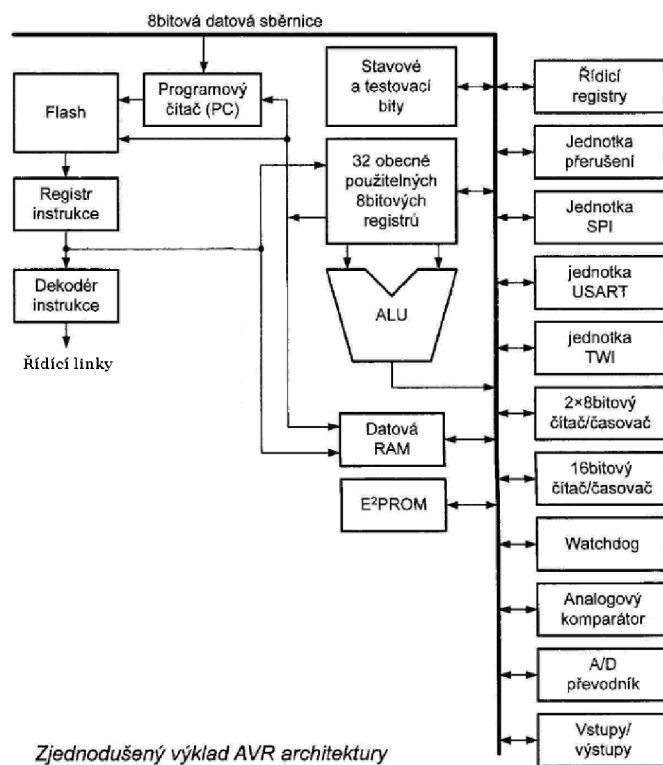
Jedná se o integrovaný obvod obsahující mikro počítač. Mikrokontroléry se vyznačují výbornou spolehlivostí a kompaktností, proto se užívají například v oblasti řízení, regulace a dalších odvětvích průmyslu. Mikrokontrolér ATMEGA je osmibitový mikrokontrolér založený na architektuře RISC. Jeho výpočetní rychlost je dána pracovní frekvencí. Provádí jednu výkonnou instrukci za jeden hodinový cyklus. Při frekvenci 16 MHz tedy dosahuje výpočetní rychlosti 16 MIPS.

2.2 POPIS JEDNOTLIVÝCH ČÁSTÍ MIKROKONTROLÉRU

Mikrokontrolér je čip, na kterém je integrován mikroprocesor, paměti a vstupně výstupní obvody. Mikroprocesor je složitý číslicový obvod, který s pomocí programu do něj vloženého vykonává sled aritmeticko-logických operací.

Mikrokontroléry AVR z hlediska rozdělení paměti užívají harvardskou architekturu – paměť programu je oddělená od paměti pro data. Pro paměť programu se používá paměť typu flash. Paměť flash je obdobou paměti EEPROM, jde mazat elektronickou cestou, je statická a energeticky nezávislá. Pro datovou paměť lze užít paměti SRAM a EEPROM. Paměť SRAM je energeticky závislá, má ale velmi krátkou přístupovou dobu. Proto se v mikrokontrolérech užívá pro uložení hodnot pracovních registrů, I/O registrů a pro uložení proměnných dat programu. Paměť EEPROM je energeticky nezávislá, práce s ní je však pomalá. Užívá se například k uchování parametrů které nastaví uživatel přístroje.

Vstupně výstupní obvody zajišťují různé funkce - komunikaci mikrokontroléru s okolím, čítače, časovače, režim pulsně šířkové modulace, analogově digitální převodník a další. Vybavenost vstupně výstupních obvodů je různá u každého mikrokontroléru.



Obrázek 1 - Architektura mikrokontroléru AVR [1]

2.3 POPIS VSTUPNĚ VÝSTUPNÍCH OBVODŮ

Zde je popis vstupně výstupních obvodů co jsou užity pro tuto práci:

Jednotka přerušení při splnění jistých podmínek přeruší běh hlavního programu a nainicializuje běh programu přerušení. Přerušení mají svou prioritu a nastane-li jich více naráz, bude přednostně obslouženo to s nejvyšší prioritou. Přerušení může být i vnořené. Užívá se hlavně pro obsloužení vstupně výstupních obvodů. Po vykonání programu přerušení pokračuje běh hlavního programu.

Obvod čítač/časovač zvyšuje svoji hodnotu v závislosti na hodinovém kmitočtu mikrokontroléru. To nám umožňuje vyvolávat přerušení ve stanovených intervalech, měřit kmitočet vstupního signálu nebo řídit střidu PWM režimu. Závislost na hodinovém kmitočtu závisí na nastavení předděličky a může být 1:1 až 1:1024. Čítač může být 8-bitový či 16-bitový.

A/D převodník převádí hodnotu analogového signálu na digitální a umožňuje jeho zpracování mikrokontrolérem. Přesnost převodu je závislá na jeho rychlosti, řádově se pohybuje mezi 8 až 10 bity.

Jednotka USART zajišťuje komunikaci mikrokontroléru s ostatními zařízeními. Může fungovat synchronně či asynchronně. Nejdříve je třeba nastavit parametry komunikace – rychlost, paritu, počet stop bitů atd. Poté inicializujeme komunikační jednotku a komunikace

může probíhat. Data se odesílají či přijímají po bytech. Pro odeslání stačí uložit byte do datového registru této jednotky a počkat než jednotka USART odešle data. Přijímání dat probíhá automaticky, po přijetí bytu je nastaven přerušovací bit, který nám oznámí že byl přijat byte dat.

Komunikační obvody SPI a TWI fungují na podobném principu. Jeden mikrokontrolér však musí být nastaven v režimu master a ostatní v režimu slave. Oproti jednotce USART dokáží hardwarově určit adresáta. Komunikace tedy může probíhat po jedné sběrnici mezi více zařízeními.

2.4 PROGRAMOVACÍ JAZYKY PRO MIKROKONTROLÉRY

Programovací jazyky rozdělujeme na nižší a vyšší.

Nižší programovací jazyk je například jazyk symbolických adres. Využívá symbolické reprezentace jednotlivých strojových instrukcí, které jsou třeba k vytvoření strojového kódu pro daný mikroprocesor. Překlad jazyka symbolických adres na strojový kód provádí program zvaný assembler. Nižší programovací jazyk je z hlediska psaní programu složitější, zabírá ale méně místa v paměti programu, dokážeme přesně určit dobu trvání jednotlivých částí programu a program probíhá mnohem efektivněji.

Vyšší programovací jazyk může být například C, C++, Pascal a jiné. Instrukce psané ve vyšším programovacím jazyce jsou překladačem přeloženy na jazyk symbolických adres a ten je posléze přeložen na strojový kód. Výhoda vyššího programovacího jazyka je rychlost vytvoření programu a také jeho přehlednost.

Časově kritické aplikace je tedy vhodné psát v nižším programovacím jazyce, pomalejší aplikace je vhodné psát ve vyšším programovacím jazyce.

2.5 PROGRAMOVÁNÍ PAMĚTI MIKROKONTROLÉRU

Naprogramovat paměť mikrokontroléru lze jak sériově, tak paralelně:

Paralelní programování se užívá méně než sériové programování. Mikrokontrolér musí mít při programování odpojené I/O vývody, proto musí být většinou vyjmut z obvodu ve kterém je zapojen. Pokud si však špatným nastavením mikrokontroléru zablokujeme možnost přenastavit jej (například nastavením externího hodinového zdroje, který nemáme), paralelním programováním lze tyto vývody přenastavit.

Sériové programování užívá několika vývodů (MISO, MOSI, SCK, RESET). Jeho velká výhoda je možnost naprogramovat mikrokontrolér zapojený v obvodu. To umožňuje ladění programu přímo v aplikaci či aktualizaci softwaru.

3. BEZDRÁTOVÁ KOMUNIKACE

Tato kapitola čerpá ze zdrojů [12], [13], [14], [15], [16], [17], [18].

3.1 DRUHY BEZDRÁTOVÉ KOMUNIKACE

Bezdrátová komunikace spočívá ve spojení více objektů jinak než mechanicky. Dle nosného média ji můžeme rozdělit na sonickou, optickou a rádiovou.

1. Sonická komunikace:

verbální komunikace – užití v mezilidské komunikaci

ultrazvuková komunikace – užití v echolokaci či sonografii

2. Optická komunikace:

Infračervené spoje – užití v dálkových ovládaních, v čidlech, ...

Laserové spoje – užití v měřicích aplikacích či optických záznamových médiích - CD, DVD, ...

3. Rádiová komunikace:

Přenos rádiovými vlnami – užití v přenosu informací ve většině bezdrátových aplikací – televizní přenos, telekomunikace, ...

Sonická komunikace se v praxi nepoužívá neboť snadno podléhá rušení okolí a je pomalá.

Optická komunikace nám již z hlediska přenosu informací vyhovuje mnohem více, ke komunikaci však musí být vysílač namířen směrem na přijímač a v optické dráze se nesmějí vyskytovat žádné překážky.

Rádiová komunikace je pro přenos informací nejlepší neboť rádiové vlny dokáží pronikat mechanickými překážkami a vyzařování informací bývá všesměrové. Frekvence užívaných rádiových vln je od 3 Hz (komunikace s ponorkami) po 300GHz (radioastronomie, vysokorychlostní mikrovlnný přenos dat).

3.2 MODULACE SIGNÁLU

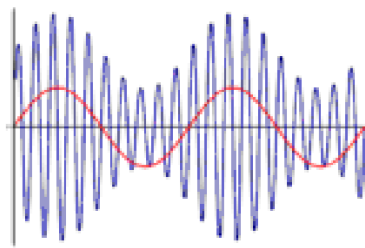
Vysílací frekvence bývá v naprosté většině případů vyšší než frekvence přenášené informace. Je tedy třeba přenášenou informaci intermodulovat do vysílací frekvence.

Definice pojmů: modulační signál je signál, který chceme přenést. Nosný signál je signál, který modulujeme modulačním signálem. Modulovaný signál je výsledný signál po provedení modulace. Demodulace je opak intermodulace – z modulovaného signálu odstraníme nosný signál a zůstane modulační signál.

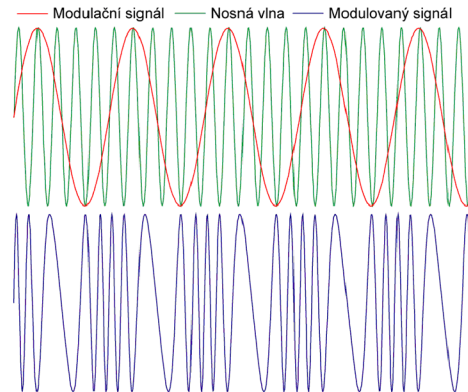
V praxi se nejčastěji používá amplitudová a frekvenční modulace:

Amplitudová modulace nemění fázi ani frekvenci nosné vlny. Vysílač musí obsahovat modulátor, který v závislosti na amplitudě modulačního signálu mění amplitudu nosné vlny. Přijímač musí obsahovat demodulátor, který odstraní nosný signál. Po jeho odstranění zůstane modulační signál. Příklad amplitudové modulace demonstruje Obrázek 2.

Frekvenční modulace mění frekvenci nosného signálu v závislosti na velikosti amplitudy modulačního signálu. Vysílač musí opět obsahovat modulátor a přijímač demodulátor. Příklad frekvenční modulace demonstruje Obrázek 3.



Obrázek 2 - Amplitudová modulace [14]



Obrázek 3 - Frekvenční modulace [15]

3.3 DRUHY KOMUNIKAČNÍCH MODULŮ

Byl proveden průzkum trhu v oblasti bezdrátových modulů použitelných pro řešení dané úlohy. Obecně je možné použít tři hlavní technologie: ZigBee, Bluetooth a Wi-Fi.

3.3.1 ZigBee

ZigBee je bezdrátová komunikační technologie popsána standardem IEEE 802.15.4. Patří do kategorie osobních počítačových sítí, tzv. sítí PAN (Personal Area Network). ZigBee je primárně určen pro průmyslové aplikace kde není třeba velké přenosové rychlosti dat. Technologie ZigBee umožňuje tvorbu rozsáhlých sítí a za cíl si dává nízkou spotřebu energie, vysokou spolehlivost, jednoduchou implementaci a v neposlední řadě i nízkou tržní cenu. Přenosová rychlost se pohybuje od 20 kb/s do 250 kb/s. Vzdálenost mezi komunikačními moduly ve volném prostoru se může pohybovat od 100 m po několik kilometrů. Typické uplatnění ZigBee komunikace je řízení budov, dálkové ovládání zařízení, průmyslová automatizace a další.

3.3.2 Bluetooth

Bluetooth je bezdrátová komunikační technologie sloužící k propojení dvou a více elektrických zařízení. Stejně jako ZigBee spadá do kategorie počítačových sítí PAN. Přenosové rychlosti se liší dle jednotlivých standardů Bluetooth a může se pohybovat od 1 Mb/s po 24 Mb/s. Vzdálenost mezi komunikačními moduly se ve volném prostoru pohybuje od 1 m do 100 m. Jeden řídicí (master) modul dokáže obsloužit simultánně až 7 podřízených (slave) zařízení. Takovéto sítě se mohou dále sdružovat do sítí větších. Typické užití Bluetooth komunikace se uplatňuje mezi telefony, počítačem a tiskárnou atd.

3.3.3 Wi-Fi

Wi-Fi je bezdrátová komunikační technologie definovaná standardem IEEE 802.11. Wi-Fi spadá do kategorie sítí LAN (Local Area Network). Přenosové rychlosti jsou od 2 Mb/s po 54 Mb/s. Vzdálenost mezi komunikujícími zařízeními je obdobná jako u standardu Zigbee. Cíl Wi-Fi je zajistit bezdrátové propojení elektrických zařízení v malých sítích (například firem) a také zajistit jejich připojení k síti internet.

Výhoda předchozích komunikačních modulů je, že vyhodnocení zda-li jsme adresáti přijaté informace nebo ne, za nás obstarává komunikační modul. Komunikační moduly mohou rovněž signál šifrovat, dešifrovat a obstarávat mnohé další funkce.

3.3.4 Jiné řešení:

Jestliže z nějakého důvodu nechceme užívat některý z těchto komunikačních modulů, můžeme si vytvořit vlastní modul. Při jistém zjednodušení můžeme říct, že vysílač se skládá z vysokofrekvenčního oscilátoru, modulátoru, vysokofrekvenčního zesilovače a antény. Modulační signál intermodulujeme spolu s nosným signálem. Ten poté zesílíme VF zesilovačem a signál z něj vyzáříme anténou. Přijímač přijme signál z antény, zesílí jej, demoduluje a výsledný modulační signál pošle na svůj výstup. Další možností je zakoupit integrovaný obvod, který v sobě má zabudovaný vysílač a přijímač.

Nevýhodou těchto dvou řešení je nutnost pro tyto moduly naprogramovat komunikační protokol. Důležité je také vysílat v povoleném frekvenčním pásmu, jinak bychom se mohli dostat do konfliktu s českým telekomunikačním úřadem.

3.3.5 Výběr komunikačního modulu

Pro výběr nejvhodnějšího modulu je zde uvedena Tabulka 1 shrnující nejdůležitější vlastnosti vybraných komunikačních modulů:

Komunikační modul	XBee ZB XB24-Z7CIT-004	XBee – PRO 868 XBP08-DPUIIT-024	CC2430F128RTC (Texas Instruments)	ZyXEL NGB-417N	BTM-112
Druh modulu	ZigBee	ZigBee	ZigBee	Wi-Fi	Bluetooth
Dosah v budově [m]	40	550	-	-	16-35
Dosah venku [m]	120	40 000	-	-	-
Komunikační rychlost [Kbps]	250	24	250	100 000	460
Nosná frekvence [MHz]	2400	868	2400	2400	2400
Klidová spotřeba [mW]	1,25	63	-	6 000	-
Vysílací spotřeba [mA]	35	500	26,7	1000	80
Přijímací spotřeba [mA]	38	65	27		
Cena [Kč]	499	1885	283	754	292

Tabulka 1: Přehled komunikačních modulů [19], [20], [21], [22], [23]

Komunikační moduly XBee od společnosti Digi International se vyznačují dobrou cenou a lze je jednoduše používat. Naproti tomu je dokumentace k modulům XBee většinou příliš stručná a najít potřebné informace je poměrně komplikované.

Produkt ZigBee od firmy Texas Instruments se oproti modulům XBee vyznačuje nižší spotřebou, což bude nejspíše vykompenzováno menším komunikačním dosahem. Dokumentace je jednoduše nalezitelná a je velmi podrobná. Modul ZigBee umožňuje velký rozsah nastavení, což vede k poměrně komplikovanému užívání.

Modul BTM-112 má podobné vlastnosti jako modul XBee – jednoduché užívání, dobrou cenu, stručnou dokumentaci.

Modul Zyxel je beztrátový router. Dokumentace je naprosto jednoduchá a přehledná, ale neříká nic o vnitřní stavbě modulu. Router se nastavuje pomocí internetového prohlížeče přímo z počítače.

Komunikační modul Wi-Fi je určen pro větší sítě, proto by nebylo efektivní nasazovat jej na dvou bodovou komunikaci. V úvahu proto připadají komunikační moduly Bluetooth a ZigBee. Fyzickou velikost mají přibližně stejnou. Modul Bluetooth má oproti modulu ZigBee značně vyšší komunikační rychlost. Rozhodující faktor však byla spotřeba modulů. Zatím co modul Bluetooth vydrží z bateriového napájení asi týden, modul ZigBee při dobrém nastavení i dva roky.

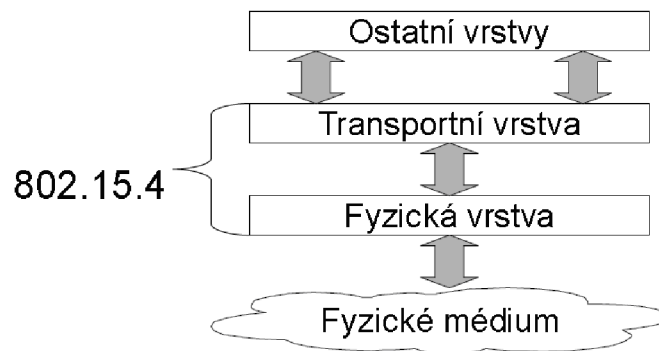
Výběr ZigBee modulů je poměrně velký. Ovšem fakt, že 2 XBee moduly bylo možno si od fakulty VUT zapůjčit, jednoznačně rozhodl pro XBee.

4. MODUL XBEE

Tato kapitola používá zdroje [4], [8].

4.1 PROTOKOL IEEE 802.15.4

Protokol 802.15.4 definuje fyzickou a transportní vrstvu ISO-OSI modelu. Jak jsem již uvedl, mezi hlavní výhody ZigBee modulů patří jednoduchost protokolu, nízká cena, nízká spotřeba elektrické energie a možnost tvorby rozsáhlých sítí. ZigBee se tedy nesnaží působit jako konkurence Wi-Fi či Bluetooth, ale snaží se být jejich doplňkem. Frekvence vysílače se pohybuje v bezlicenčním pásmu. Standard ZigBee dovoluje vysílat v několika různých frekvenčních pásmech. Modul XBee pracuje v jednom celosvětově používaném pásmu 2400 MHz - 2483,5 MHz, které je rozděleno na 16 komunikačních kanálů.



Obrázek 4 - Model ISO-OSI protokolu XBee [8]

4.2 JAK PRACUJE MODUL XBEE

4.2.1 Posílání dat

Data posíláme asynchronní sériovou linkou na pin DIN (Digital Input). Tyto data se ukládají do vyrovnávací paměti (Serial Receiver Bufferu). Pokud je to možné, jsou data ihned preposílána vysílači. Pokud přijímač přijímá stálý tok informací, data nebudou odeslána dokud tento tok neustane.

Modul musí najít adresáta nebo vytvořit cestu sítí k adresátovi. Po odeslání dat má adresát poslat potvrzení že data byla doručena v pořádku. Pokud modul toto potvrzení nedostane, data odesílá znovu. Tento proces zabírá jistý čas navíc a proto se snižuje komunikační rychlost přenosu a data se začnou hromadit ve vyrovnávací paměti. Při přetečení

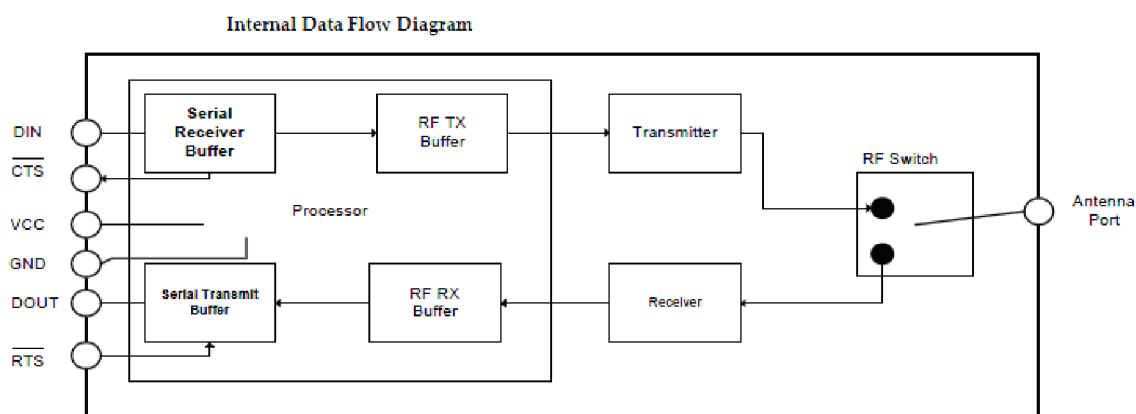
vyrovnávací paměti dojde ke ztrátě posílaných dat. Abychom této ztrátě zabránili, musíme kontrolovat stav pinu CTS (Clear To Send). Pokud je tento bit nastaven, znamená to, že vyrovnávací paměť je plná a nemáme posílat další data. Až se data odešlou, bit je nastaven do logické nuly.

4.2.2 Přijímání dat

Přijátá data jsou ukládána do vyrovnávací paměti (Serial Transmit Bufferu). Data přijímáme asynchronní linkou z pinu DOUT (Digital Output). Data jsou posílána na výstup ihned po jejich přijetí pokud není mikrokontrolérem nastaven bit RTS (Ready To Send). Je-li bit RTS nastaven, modul XBee uchovává data ve vyrovnávací paměti a čeká na nastavení bitu RTS do logické nuly.

Pokud je přijímací rychlost modulu XBee vyšší než rychlost přijímání dat mikrokontrolérem, data se hromadí ve vyrovnávací paměti.

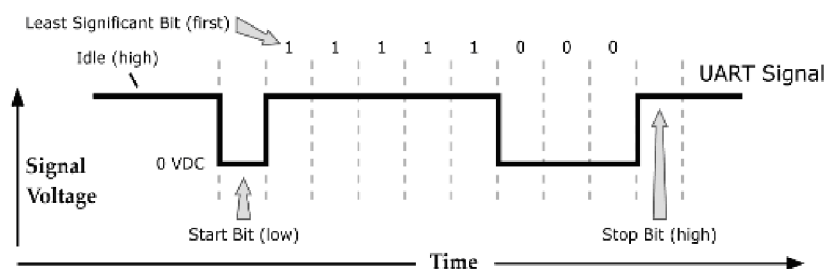
Při přetečení vyrovnávací paměti dojde ke ztrátě právě přijatých dat. Je tedy zajistit dostatečně rychlý přenos dat z modulu XBee do mikrokontroléru. Po přijetí dat posílá přijímací modul vysílacímu modulu zprávu potvrzující správné přijetí dat.



Obrázek 5 - Schéma toku dat uvnitř modulu XBee [4]

4.2.3 Přesun dat mezi mikrokontrolérem a modulem XBee

Data jsou posílána asynchronní sériovou linkou. Defaultní nastavení je následující posloupnost – start bit, 8 bitů nosících informací, stop bit. Toto nastavení lze změnit v nastavení komunikačního modulu. Pokud změníme nastavení, je nutné znát jeho poslední parametry, jinak nebudeme schopni navázat opět komunikaci. Nastavit můžeme přenosovou rychlost, paritu, počet stop bitů a počet bitů nosících informací. Defaultní nastavení ilustruje následující Obrázek 6.



Obrázek 6 - Schéma UART přenosu dat [4]

4.2.4 Command mód

Jak již bylo řečeno, data přijímaná asynchronní sériovou linkou jsou vysílána jinému komunikačnímu modulu. Při komunikaci s modulem užíváme ASCII kódu. Pokud chceme měnit nastavení XBee modulu, musíme vstoupit do command módu. Abychom do něj vstoupili, musíme splnit následující postup:

1. Vyčkat ochranný čas GT (Guard Time), který zabraňuje nechtěnému přístupu do command módu.
2. Pošleme tři ACSII znaky “+++“.
3. Vyčkáme Guard Time.

Poté můžeme posílat modulu příkazy měnící jeho nastavení. Příkazy mají následující tvar: Předpona AT, příkaz, mezera, parametr příkazu, znak CR (Carriage Return). Command mód lze ukončit command příkazem nebo počkáme čas specifikovaný nastavením hodnoty CT (Command Mode Timeout). Po každém command příkazu počkáme na přijetí potvrzení příkazu “OK“. V případě že příkaz nebude přijat správně, přijde odpověď “ERROR“. V případě že posíláme příkaz pro zjištění stavu nějakého parametru, dojde v odpovědi právě tento parametr.

4.2.5 Rozhraní sériového protokolu

Moduly XBee podporují jak transparentní sériové rozhraní, tak API (Application Programming Interface) sériové rozhraní.

V módu transparentního rozhraní se XBee chová jako kabel. Data poslána na pin modulu vysílače DIN jsou vysílána na jiný modul, který je pošle na svůj pin DOUT. Nastavení parametrů modulu se provádí v command módu. Před vstupem do command módu odešle XBee modul všechna data z přijímací vyrovnávací paměti mikrokontroléru.

API rozhraní je alternativou k transparentnímu rozhraní, rozšiřuje však možnosti mikrokontroléru při interakci se sítí. Data vstupující a vystupující z modulu jsou vysílána v rámcích, které obsahují události a operace které se dějí uvnitř modulu.

Rámec odeslaný do modulu musí obsahovat datový rámec (přenášená data) a příkazový rámec (příkazy sloužící k nastavení parametrů modulu). Rámec přijatý z modulu obsahuje datový rámec, příkazový rámec (odpověď na příkazy) a oznámení události (přenáší informace o tom co se děje v modulu, např. reset modulu atd.).

Výhody API rozhraní: API rozhraní dokáže posílat data více modulům bez nutnosti měnit cílovou adresu v command módu, což znatelně urychluje přenos dat. To také znamená že je modul schopen přijímat data z více modulů najednou. Modul dostane informaci o úspěchu či neúspěchu přijetí každého odeslaného paketu. Adresát je schopen identifikovat odesilatele paketu. API rozhraní umožňuje bezdrátové nastavení parametrů modulu

4.2.6 Dělení stavů modulu XBee

Idle mode – stav kdy modul není v command módu a nepřijímá a nevysílá žádná data.

Transmit mode (vysílací mód) – přepnutí nastane při přijmutí dat po asynchronní sériové lince

Recieve mode (přijímací mód) – přepnutí nastane při přijmutí dat skrze rádiové vlny

Command mode (nastavovací mód) – přepnutí nastane při přijmutí sekvence příkazů pro vstupu do command módu

4.2.7 Pracovní režimy modulu XBee

Spící režim (sleep mode): Modul ve spícím režimu není schopen přijímat data prostřednictvím rádiových vln, ani skrz asynchronní sériovou linku. Všechny moduly se probouzí po stejném intervalu, vymění si informace a synchronní zprávy a poté se opět přepnou do režimu spánku.

Normální režim (normal mode): Modul nemůže být ve spícím režimu. Pomáhá však synchronizovat spící moduly.

4.3 VLASTNOSTI SÍTĚ MODULU XBEE

4.3.1 Charakteristiky sítě DigiMesh

- Připojení či odpojení jakéhokoliv modulu nezpůsobí pád sítě
- Není třeba hierarchie modulů v síti
- Mapa sítě není nikde uložena, komunikační cesty mezi moduly jsou tvořeny a užity až v případě jejich potřeby
- Na přijaté příkazy odpovídá pouze adresát
- Správné přijetí dat je potvrzeno zprávou od adresáta vysílači
- Moduly se mohou přepnout do úsporného režimu spánku se synchronizovaným probuzením

4.3.2 Přenos dat

Unicast addressing (vysílání pro jeden modul): Po odeslání dat modul čeká na potvrzení jejich správného přijetí. Čas čekání je dvojnásobek doby cestování dat skrz celou síť. Pokud potvrzení nepřijde, jsou data vyslána znovu. Počet opakování je definován nastavením parametru NR (Network Retries). Pro poslání dat ve formě unicast addressingu je třeba nastavit parametry vysílacího modulu DH (Destination High) a DL (Destination Low) tak aby se rovnaly parametrům SH (Serial High) a SL (Serial Low) adresáta.

Broadcast addressing (vysílání všem modulům): V tomto módu se nepoužívá potvrzení správného přijetí informace. Proto vysílač pošle informaci 4x po sobě. Poslání broadcast zprávy silně sníží přenosovou rychlost sítě, proto by se tento mód adresování měl užívat zřídka.

Route discovery (hledání cesty): Pokud vysílací modul nezná cestu k adresátovi nebo od něj nedostává potvrzení o správném přijetí dat, začne hledat novou cestu skrz síť. Toto hledání začíná broadcastovou zprávou. Moduly přijímající tuto zprávu ji mohou přeposlat dále nebo ji nechají být. Tato zpráva se přeposílá tak dlouho, dokud se nedostane k adresátovi. Adresát zprávu přijme a odešle pro vysílač zprávu udávající cestu k němu. Je možné že bude existovat více funkčních cest. V tom případě vysílací modul vybere cestu s nejlepší kvalitou přenosu.

Routing (směrování): Vysílač posílá zprávu adresátovi skrz síť. Aby zpráva dorazila k adresátovi co nejefektivněji, musí obsahovat adresy modulů, přes které bude zpráva přeposílána. K nalezení nejvhodnější cesty slouží Route discovery.

4.3.3 Topologie sítě modulu XBee

Topologie sítě se skládá ze tří různých zařízení:

Koordinátor:

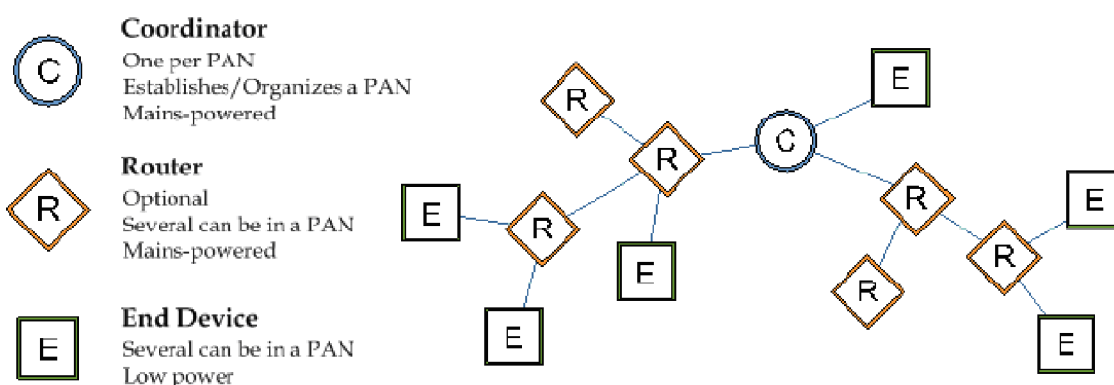
- vybírá komunikační kanál a PAN ID
- povoluje routerům a koncovým zařízením připojit se do sítě
- nemůže být v režimu spánku
- může se účastnit přeposílání dat
- může dočasně ukládat data pro koncová zařízení, která jsou momentálně v režimu spánku

Router:

- než může posílat, přijímat či přeposílat data, musí se připojit do sítě
- po připojení povoluje routerům a koncovým zařízením připojit se do sítě
- po připojení se může účastnit přeposílání dat
- nesmí být v režimu spánku
- může dočasně ukládat data pro koncová zařízení, která jsou momentálně v režimu spánku

Koncové zařízení:

- než může posílat či přijímat data, musí se připojit do sítě
- nemůže povolit ostatním zařízením připojit se do sítě
- data ostatním může posílat pouze přes router či koordinátora
- může být v režimu spánku a tím snížit svou spotřebu, tudíž může být napájen z baterie



Obrázek 7 - Schéma sítě modulu XBee [4]

4.3.4 Modely sítě modulu XBee

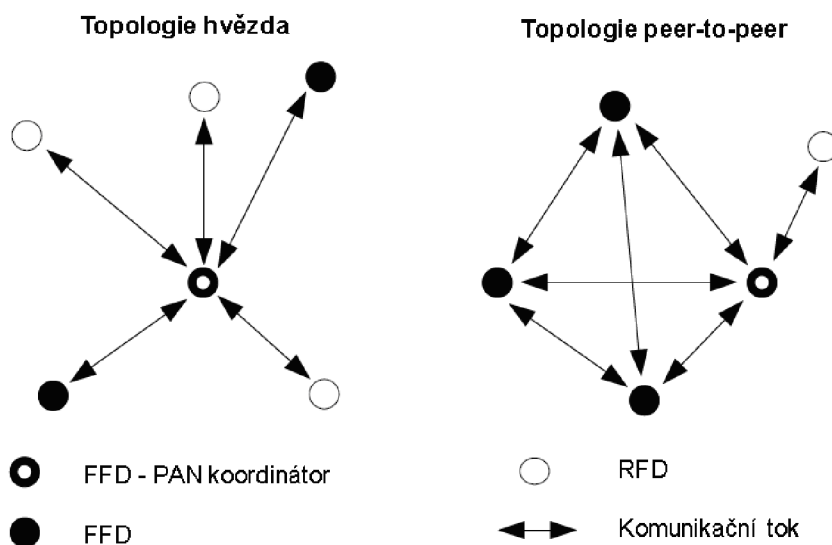
Uzly sítě se dělíme na dva typy:

1. **Full function device (FFD)** – obsahuje plnou implementaci 802.14.5 protokolu. Mohou komunikovat se všemi uzly v síti a mohou se stát koordinátory sítě.
2. **Reduced function device (RFD)** – může komunikovat s FFD, nemůže se však stát koordinátorem sítě. Toto zařízení je jednodušší a levnější.

Síť definovaná protokolem 802.15.4 má dvě topologie – hvězdicovou topologii a peer-to-peer topologii.

Hvězdivová topologie: Tato topologie je výrazně jednodušší než peer-to-peer topologie, poskytuje však mnohem menší komunikační možnosti. Nejdříve se aktivuje hlavní FFD. Najde volné PAN ID, najde ostatní prvky sítě a povolí připojení uzlů. Veškerá komunikace probíhá přes koordinátora. Proto je tento typ komunikace pomalý.

Peer-to-peer topologie: Výhoda této topologie je že rovnocenné uzly mohou mezi sebou vzájemně komunikovat. Koordinátor pouze řídí síť. Tím že komunikace neprobíhá přes koordinátora je komunikace rychlejší, ale složitější.



Obrázek 8 - Topologie sítě modulu XBee [8]

5. ZPŮSOBY A ZABEZPEČENÍ SPRÁVNÉHO PŘENOSU DAT

Tato kapitola čerpá ze zdrojů [3], [9], [24].

5.1 DRUHY PŘENOSU DAT

Analogový přenos dat: Každá úroveň hodnoty přenášených data nese informaci. Proto je analogový přenos dat velmi choulostivý na vnější rušení a na nelineární zkreslení vysílače či přijímače.

Digitální přenos dat: Obsahuje jen dvě úrovně hodnot. Díky tomu není tolik choulostivý na nelineární zkreslení vysílače nebo přijímače. Vnější rušení během bezdrátového přenosu však může způsobit špatný přenos jednoho či více bitů z posílaných dat a znehodnotit tak přenášená data.

5.2 ZPŮSOBY DIGITÁLNÍHO PŘENOSU DAT

Způsob přenosu dat volíme v závislosti na možnosti přenosové cesty a na zajištění správného přenosu dat. K zajištění správného přenosu dat přidáváme k informaci další bity sloužící ke kontrole správné přijetí informace. Tím zpomalujeme rychlost komunikace, protože musíme přenášet nadbytečná data. Nadbytečnost informací se nazývá redundance.

Pro zjištění chyb v přenosu vzniká pouze nízká redundance neboť přidáváme pouze paritní bity či kontrolní součet. Existují algoritmy, které dokáží do určitého procenta chybně přenesených bitů v přenosu opravit chybně přijaté bity. Tyto algoritmy jsou výpočetně náročné a vzniká u nich vysoká redundance.

Dalším důvodem proč přidávat bity navíc může být to, že některé přijímače potřebují dostávat signál s poměrem logických nul a logických jedniček 1:1, aby správně fungovala synchronizace frekvence vysílače a přijímače.

5.2.1 Bitové kódování

Dvoufázové bitové kódování (Manchesterské kódování): Každý bit je vysílán dvakrát – jednou přímo a jednou negovaně. Tím je zajištěn poměr logických jedniček a logických nul 1:1. Komunikační rychlost je ale snížena na polovinu, vzniká redundance o velikosti 100 %. Dekódování signálu je poměrně složité, protože je náročné na synchronizaci.

Bitové kódování 1/3 : 2/3: Na každý bit nesoucí informaci vysíláme 3 bity – nejdříve je vyslána logická 0, poté datový bit a následují logická 1. Délku vyslání těchto tří bitů označíme T. Komunikační rychlost je snížena na třetinu, redundance má velikost 200 %.

Výhodou je, že dekódování signálu je poměrně jednoduché. Přenos bitu je zahájen přechodem signálu do logické 0. Poté vyčkáme $T/2$ času a navzorkujeme signál.

5.2.2 Jiné druhy kódování

Bytové kódování: Toto kódování užívá vyváženého počtu jedniček a nul. V 8 bitech existuje 68 kombinací, které obsahují poměr jedniček a nul 1:1. Bytové kódování uijeme, pokud nám stačí 68 binárních kódů. Při vysílání musíme vlastní data nejdříve zakódovat pomocí převodní tabulky o 68 znacích. Při přijímání je musíme dle stejné tabulky dekódovat.

Kódování FEC: Každý byte je vysílán dvakrát – jednou přímo a jednou negovaně. Při přijetí jsou oba byty porovnány. Je-li jeden po negaci rovný druhému, jsou data přijata správně.

Zabezpečení přenosu paritním bitem: Ke každému bytu je přidán jeden bit, který doplní počet logických jedniček na sudý nebo na lichý. Tento způsob zabezpečení dokáže odhalit chyby pouze v lichém počtu bitů. Vzniknou-li chyby v sudém počtu, nebude chyba přenosu dat odhalena.

Zabezpečení přenosu použitím CRC (Control Redundant Code): Vysílaná data jsou doplněna řetězcem CRC. CRC je založen na dělení v konečném tělese. Jde o množinu polynomů, jejichž koeficienty mohou nabývat pouze hodnot 0 a 1. Výpočet CRC můžeme provést součtem v tělese nebo operací XOR. Tento výpočet však nepozná některé druhy chyb. Lepší možnost odhalení chyby přenosu dat nabízí užití primitivního polynomu.

Jak pracuje kód CRC s primitivním polynomem:

1. Nejdříve si zvolíme klíč (polynom $G(x)$). Tento polynom musí začínat číslem 1.
2. Zkopírujeme data určená k odeslání. Tato data nazveme *odesílaná data*.
3. Za zkopírovaná data přidáme tolik nul, jaká je délka klíče $G(x)$. Tyto data nazveme *CRC data*. Máme tedy řetězec skládající se z části *odesílaná data* a z části *CRC data*.
4. Vezmeme první bit zleva od *odesílaných dat* či od *předešlého výsledku* (viz. dále). Je-li tento bit nulový, provedeme rotaci dat vlevo. Není-li tento bit nulový, provedeme operaci XOR polynomu $G(x)$ a *odesílaných dat* či *předešlého výsledku*. Výsledek uložíme.
5. S výsledkem postupujeme jako u bodu 4 dokud z *odesílaných dat* nebudou samé nuly.
6. Odešleme *odesílaná data*.
7. Až se dokončí přenos odesílaných dat, pošleme za nimi výsledný řetězec *CRC data*.

Postup výpočtu ukazuje následující obrázek. Data k odeslání je řetězec 0101 0111, klíč je řetězec 1 0000 0111. Výsledné řetězce CRC data jsou 1010 0010 a 1001 1000.

MSB první	LSB první
0101011100000000	1110101000000000
-00000000	-100000111
=0101011100000000	=0110100110000000
-100000111	-100000111
=001010110110000000	=001010000100000000
-000000000	-100000111
=000101101100000000	=000010001010000000
-100000111	-000000000
=00000110101011000000	=000010001010000000
-000000000	-100000111
=000001101011000000	=0000000100110000
-100000111	-000000000
=0000001010101100	=0000000010011000
-100000111	-000000000
=00000000101000010	=00000000010011000
-000000000	-000000000
=00000000010100010	=00000000010011000

Obrázek 9 - příklad výpočtu CRC kódu [24]

Přijímač v sobě má klíč jako vysílač. Po přijetí části data k odeslání bude postupovat dle předchozího postupu od bodu 2 do bodu 5. Výsledný řetězec CRC data porovná s přijatým řetězcem CRC data a budou-li oba stejné, jsou přijatá data správná. V opačném případě musí požádat o jejich opětovné poslání.

Použití samoopravného (Hammingova) kódu: Tento kód umožňuje detekovat a opravit jednu chybu v přijatém bytu informací. K jednomu vysílanému bytu přidává 4 paritní bity, které jsou vytvořeny přesně definovaným způsobem. Po přijetí přijímač znovu vygeneruje dle stejného algoritmu 4 paritní bity. Pokud jsou si 4 přijaté a 4 právě vytvořené paritní bity rovny, nenastala chyba při přenosu. Pokud si nejsou rovny, provede se vždy mezi dvěma bity, které mají být navzájem ekvivalentní, operace XOR. Výsledek těchto operací se nazývá syndrom. Podle výsledku syndromu poznáme ve kterém bitu nastala chyba přenosu. Tento bit negujeme a výsledkem je správná informace.

6. ŘEŠENÍ PROJEKTU

6.1 TEORETICKÝ POPIS ŘEŠENÍ

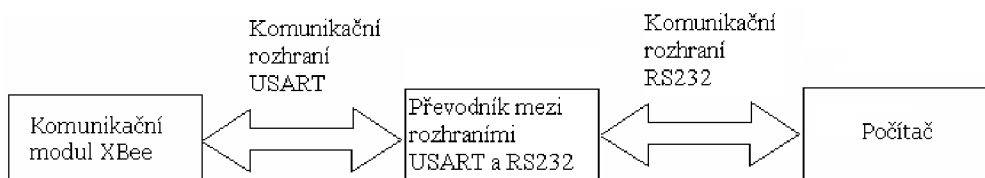
Zjednodušené schéma projektu:

Schéma funkce zařízení připojeného k otočné měřicí stolici:



Obrázek 10 - Schéma funkce zařízení z měřicí stolice

Schéma funkce zařízení připojeného k PC:



Obrázek 11 - Schéma funkce zařízení připojeného k PC

Komunikace zařízení:

Komunikační přístroj z otočné měřicí stolice: Typ snímače může být libovolný, musí s ním být možné komunikovat pomocí rozhraní SPI či USART. Přenos dat mezi mikrokontrolérem ATmega a modulem XBee bude probíhat po asynchronní sériové lince. Pro posílání dat přes bezdrátové moduly XBee bude užito transparentního rozhraní, protože při posílání dat mezi dvěma body má vyšší komunikační rychlost.

Komunikační přístroj připojený k PC: Ovládání modulu a přenos dat bude řídit počítač. Pro komunikaci bylo vzhledem k možnosti přímého ovládání modulu XBee počítačem užito sběrnice RS232. Pro převod mezi komunikačními rozhraními USART a RS232 musíme užít převodník napětových úrovní. V našem případě volíme integrovaný obvod MAX232 od firmy Texas Instruments. Tok informací mezi počítačem a vzdáleným senzorem bude řízen prostřednictvím námi vytvořené aplikace.

6.2 POŽADAVKY NA KOMUNIKAČNÍ ZAŘÍZENÍ

1. Mikrokontrolér připojený ke snímači z otočné měřicí stolice:

- Bude snímat naměřenou hodnotu ze snímače z otočné měřicí stolice po komunikačním rozhraní SPI nebo USART a odesílat ji po asynchronní sériové lince vysílacímu komunikačnímu modulu
- Bude přeposílat data z PC na senzor
- Bude přijímat povely od XBee modulu připojenému k počítači
- Bude možné měnit nastavení jeho periférií

2. Počítač připojený k XBee modulu

- Bude přijímat hodnotu z komunikačního modulu ze snímače z otočné měřicí stolice
- Bude obsahovat základní knihovny pro řízení mikrokontroléru, připojenému ke snímači z otočné měřicí stolice

6.3 ZABEZPEČENÍ SPRÁVNÉHO PŘENOSU DAT

V kapitole 5 byly uvedeny různé varianty zabezpečení přenosu dat proti chybám. Standard ZigBee definuje různé komunikační vrstvy zajišťující adresování, směrování, modulaci signálu atd. Správnost přenosu dat má na starost komunikační vrstva MAC (Media Access Control Layer) standardu ZigBee. Zajišťuje potvrzování zpráv, opakování zpráv v případě jejich neúspěšného přijetí a předchází kolizím v komunikaci mezi jednotlivými moduly. Při užití XBee modulů je tedy vcelku zbytečné užívat algoritmy pro detekci či opravu chyb když je již zajišťuje MAC vrstva standardu ZigBee. Ke ztrátě dat může dojít pouze v případě kdy se nepodaří zprávu doručit několikrát po sobě a překročíme-li limit nastaveného počtu jejího znovudoslání. Tento případ je ale značně nepravděpodobný.

Znalost zabezpečení přenosu dat je ale klíčová při užití levnějších RF zařízení. Tato zařízení nemají žádnou vnitřní logiku, pouze vysílají a přijímají data. Pokud by záleželo na ceně zařízení, museli bychom se postarat o správnost přenosu dat sami. Zde nachází výborné využití CRC kód, popřípadě Hammingův samoopravný kód. Známý jsou i mnohem účinnější samoopravné kódy, bohužel jejich složitost značně přesahuje možnosti této práce. Také bychom museli vytvořit komunikační protokol, který by se postaral o adresování zpráv a který zajistí bezkolizní přístup k vysílači a přijímači.

6.4 KOMUNIKACE S MODULY XBEE

Defaultní nastavení modulů XBee je následující: Komunikační rychlost je 9600 Baudů, paritní bit se nepoužívá a užití kódování znaků je ve formátu ASCII. Při nastavování komunikačních parametrů XBee modulů se velmi osvědčilo ukládat tyto parametry pouze do volatilní paměti. Při chybném nastavení stačí odpojit modul XBee od napájecího napětí a jeho

komunikační parametry budou opět v defaultním nastavení. Pokud bychom parametry komunikace uložili do nevolatilní paměti, existuje velké riziko že při špatném nastavení komunikačních parametrů modulu bychom jej už nedokázali znovu přenastavit. Proto i v praktickém řešení této práce ukládáme změny pouze do volatilní paměti. Další výhodou užití tohoto způsobu nastavení je že si nemusíme pamatovat předchozí nastavení XBee modulu.

6.5 KOMUNIKAČNÍ ZAŘÍZENÍ NA OTOČNÉ MĚŘÍCÍ STOLICI

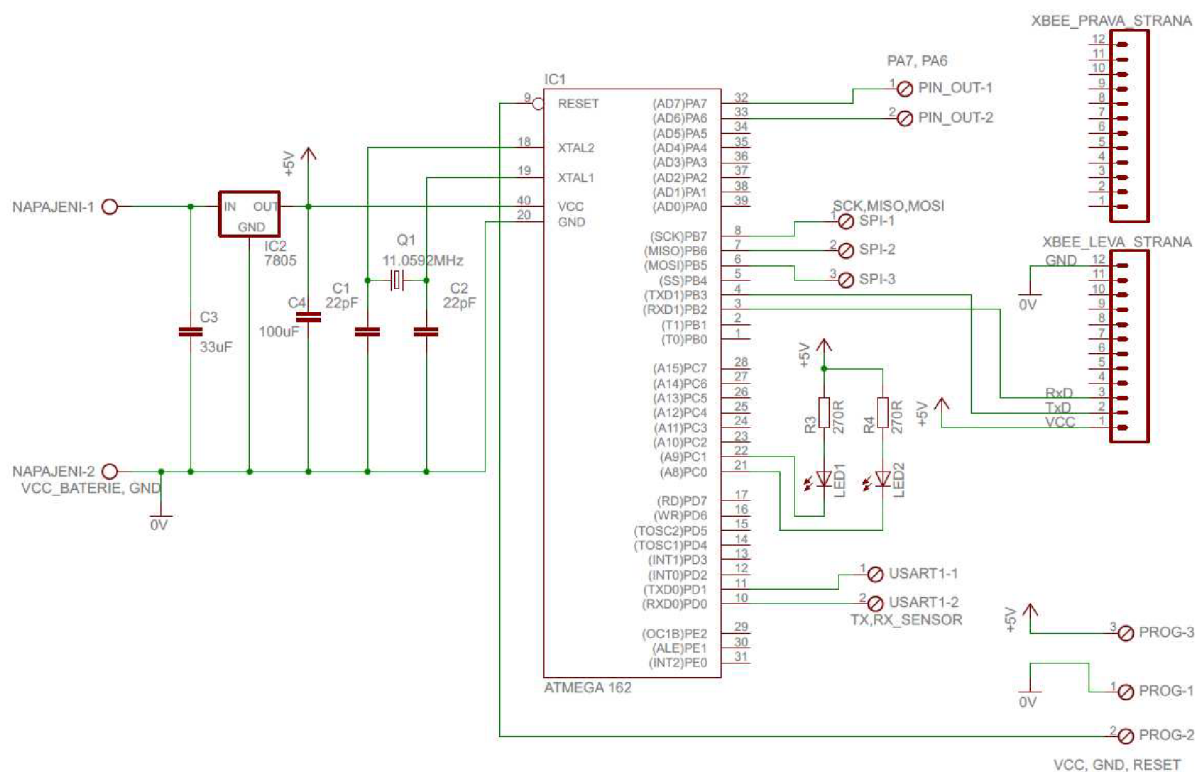
6.5.1 Hardwarové řešení

Pro napájení mikrokontroléru a modulu XBee je třeba stabilizovaný zdroj napětí +5 V. Jelikož velikost napájecího napětí baterie použité v projektu bude jen 6 V, použijeme lineární stabilizátor napětí. Byl zvolen obvod 7805-STM. Tepelný odpor mezi stabilizátorem 7805 a vzduchem je 50 °C/W. Při předpokládaném maximálním proudu 100 mA a ztrátovém napětí 1 V bude ztrátový výkon $P = 0,1 \text{ W}$. Z toho je vidět že pro jeho použití nebude třeba užit chladič. Bylo uvažováno o užití spínaných měničů napětí, ovšem jejich použití by se vyplatilo jen při vyšším napájecím napětí baterie. Použití stabilizátoru pro stálý provoz by bylo vzhledem k velikosti jeho klidového proudu 6 mA nepřijatelné. Ukázalo se však že zařízení bude po dobu provozu neustále komunikovat a po ukončení provozu bude vypnuto. Nízkou spotřebu zařízení v klidovém stavu tedy nemusíme řešit.

Použití stabilizátoru 7805 se může jevit trochu podezřelé, protože je na něm úbytek napětí přibližně 1,5 V. Při napájecím napětí +6 V bude na jeho výstupu asi 4,5 V. Mikrokontrolér je schopen pracovat v rozmezí napájecího napětí od 2,7 V do 5,5 V. U modulu XBee bylo experimentálně ověřeno že začne pracovat již při vstupním napětí 3,4 V. Použití stabilizátoru 7805 je tedy v pořádku.

Pro komunikaci se senzorem a modulem XBee jsme zvolili mikrokontrolér ATMEGA162-16PU protože obsahuje dvě USART periferie. Může tedy komunikovat se senzorem na otočné měřící stoličce a modulem XBee najednou. Také jej lze jednoduše vložit do nepájivého kontaktního pole a otestovat tak všechny funkce zařízení před výrobou DPS. Mikrokontrolér ATMEGA 162 může pracovat s maximální vnitřní frekvencí 16 MHz. Lze k němu tedy připojit krystal o frekvenci způsobující minimální chybu při nastavení komunikačních rychlostí které podporuje modul XBee, tj. krystal o frekvenci 11,059 MHz. Pro možné budoucí ladění programu jsou k mikrokontroléru připojeny LED diody které mohou indikovat vnitřní stav programu mikrokontroléru. Z desky plošných spojů jsou vyvedeny konektory pro připojení senzoru i konektory pro přeprogramování mikrokontroléru přímo v zařízení. Na DPS jsou také vyvedeny dva piny mikrokontroléru rezervovány pro možné budoucí použití. Všechny tyto konektory jsou vidlice se zámkem do DPS s roztečí 2,54 mm. Pro připojení vodiče je třeba užit zásuvku se zámkem na kabel.

Modul XBee i mikrokontrolér ATMEGA jsou zasazeny v patici aby je bylo možné v budoucnu užít i v jiných aplikacích. Schéma zapojení ukazuje Obrázek 12.

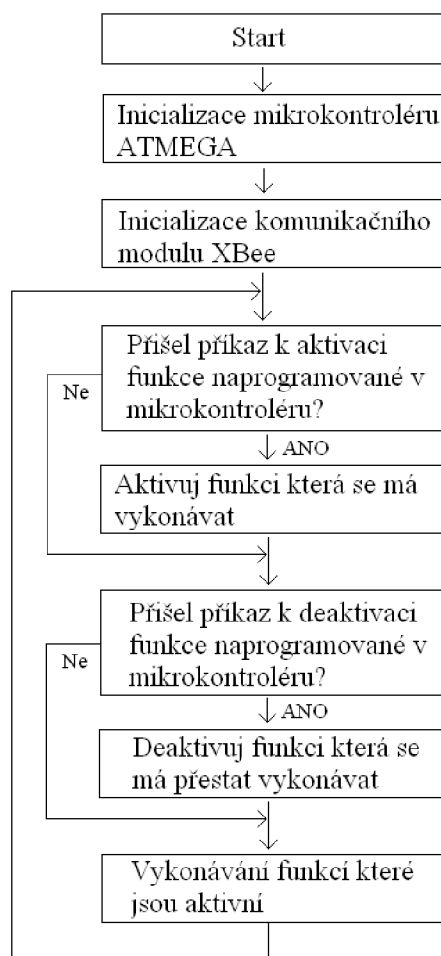


Obrázek 12: Schéma zapojení komunikačního zařízení

6.5.2 Softwarové řešení

Jelikož je komunikační rychlost modulů XBee vzhledem k výpočetnímu výkonu mikrokontroléru pomalá, bude celý projekt možné naprogramovat v jazyce C. Díky tomu bude program maximálně přehledný a dobře naprogramovatelný.

Při startu či resetu mikrokontroléru se nejdříve inicializuje mikrokontrolér a poté proběhne inicializace XBee modulu. Aby program vykonával různé funkce, musejí být nejdříve aktivovány řídicím programem počítače. Program cyklicky vykonává aktivní funkce a neustále ověřuje zda-li nedošel příkaz k aktivaci či deaktivaci nějaké funkce. Pro lepší představu o funkci programu slouží vývojový diagram (Obrázek 13).



Obrázek 13: Vývojový diagram programu mikrokontroléru

Při přijímání dat z počítače musí mikrokontrolér rozlišit data pro senzor a data pro vlastní nastavení. K odlišení dat pro změnu nastavení mikrokontroléru slouží hlavička. Jde o předem definovaný znak po kterém budou následovat řídicí příkazy. V posílaných datech pro senzor se však znak hlavičky nesmí vyskytnout. Vyskytuje-li se znak hlavičky v datech pro senzor, musíme jej nahradit. Nejdříve předem definujeme následující znaky: znak hlavičky = 0xAA, náhradní znak = 0xBB, znaky za náhradní znak = 0xBB a 0x55. Náhradní znak 0xBB nahrazuje znak hlavičky 0xAA a posloupnost znaků 0xBB a 0x55 nahrazuje náhradní znak 0xBB. Data tedy posíláme ve dvou tvarech:

1. Posílání dat pro změnu nastavení mikrokontroléru: Odešleme znak hlavičky 0xAA a za ním řídicí znaky. V mikrokontroléru po přijetí znaku hlavičky víme že další přijatá data budou řídicí příkazy. Délka řídicích příkazů je předem definována.
2. Posílání dat pro senzor: Před odesláním dat pro senzor je třeba zjistit zda-li se v nich nevyskytuje náhradní znak 0xBB. Pokud ano, nahradíme jej posloupností znaků 0xBB a 0x55. Poté zjišťujeme zda-li se v datech pro senzor nevyskytuje znak hlavičky 0xAA. Pokud ano, nahradíme jej náhradním znakem 0xBB. Data v tomto tvaru přijme

mikrokontrolér a před přeposláním těchto dat senzoru je musí vrátit původního stavu. Detekujeme-li náhradní znak 0xBB a za ním jiný znak než 0x55, nahradíme jej znakem 0xAA. Detekujeme-li posloupnost znaků 0xBB a 0x55, nahradíme je znakem 0xBB. Pro lepší pochopení jsou zde uvedeny následující příklady:

Zpracování dat počítačem před jejich odesláním mikrokontroléru		
Řetězec dat pro senzor	Náhrada náhradního znaku 0xBB posloupností náhradních znaků 0xBB, 0x55	Náhrada znaku hlavičky 0xAA náhradním znakem 0xBB. Tyto data odesíláme mikrokontroléru
0xAA, 0x54	0xAA, 0x54	0xBB, 0x54
0xBB, 0x80	0xBB, 0x55, 0x80	0xBB, 0x55, 0x80
0xAA, 0xBB	0xAA, 0xBB, 0x55	0xBB, 0xBB, 0x55

Tabulka 2: Ukázka vytvoření náhradních dat

Zpracování dat mikrokontrolérem před jejich odesláním senzoru		
Data přijatá mikrokontrolérem	Náhrada náhradního znaku 0xBB znakem hlavičky 0xAA	Náhrada posloupnosti náhradních znaků 0xBB a 0x55 znakem 0xBB. Tyto data odesíláme senzoru
0xBB, 0x54	0xAA, 0x54	0xAA, 0x54
0xBB, 0x55, 0x80	0xBB, 0x55, 0x80	0xBB, 0x80
0xBB, 0xBB, 0x55	0xAA, 0xBB, 0x55	0xAA, 0xBB

Tabulka 3: Ukázka vytvoření původních dat

Přijímá-li mikrokontrolér data pro senzor či počítač, ukládá je do zásobníku a poté je teprve odesílá. Zásobníky mají velikost 32 Bytů, je tedy možné do nich posílat data vyšší rychlostí než s jakou je přeposílá. Přetečení zásobníku díky softwarovému ošetření nastat nemůže, ale při zápisu do plného zásobníku přichází uživatel o data.

Mikrokontrolér má naprogramovány následující funkce:

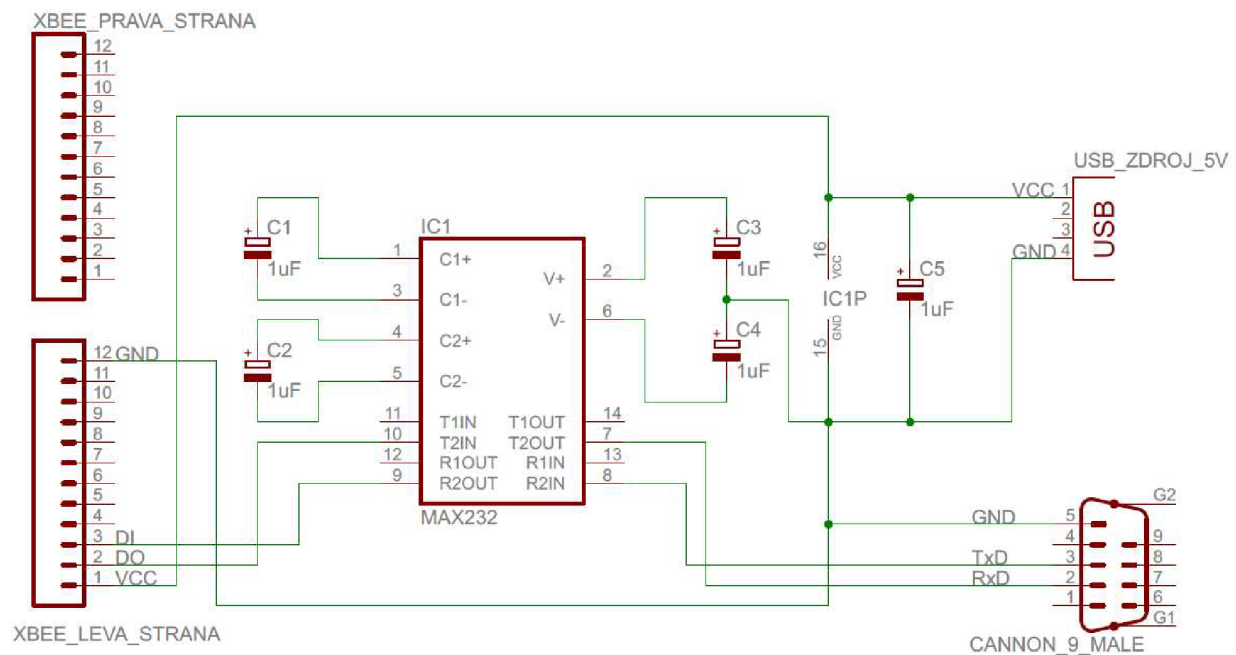
- Přeposílání dat ze senzoru do počítače
- Přeposílání dat z počítače do senzoru
- Změnu nastavení periférií dle pokynů řídicího programu počítače
- Rozsvícení nebo zhasnutí LED diod
- Cyklické posílání ASCII znaků pro otestování správné funkce bezdrátové komunikace

6.6 KOMUNIKAČNÍ ZAŘÍZENÍ PŘIPOJENÉ K PC

6.6.1 Hadrwarové řešení

Pro napájení XBee modulu a převodníku TTL logiky na sériovou sběrnici RS232 je třeba napájecí napětí +5 V. Abychom nemuseli užívat napájení z baterie, zvolili jsme napájení z USB portu. Jeho napětí je právě +5 V a může dodat proud až 100 mA, což nám postačuje.

Sběrnice RS232 užívá napěťové signály ± 10 V. Výhodou převodníku MAX232 je že v sobě obsahuje nábojové pumpy, které si vytvoří potřebné napětí. Schéma zapojení ilustruje následující obrázek:



Obrázek 14: Schéma zapojení komunikačního zařízení

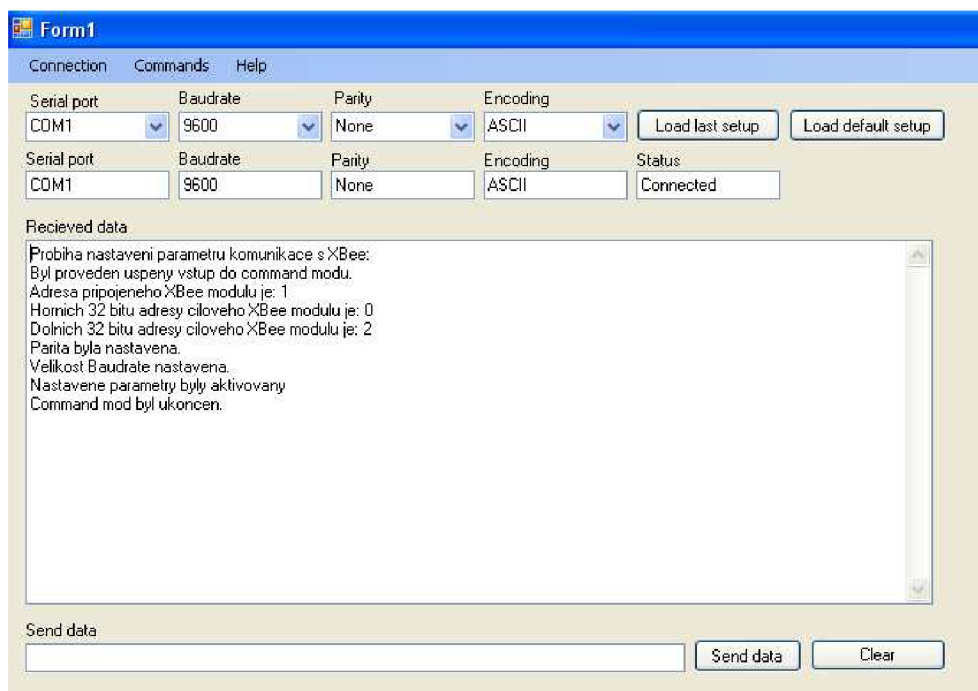
6.6.2 Softwarové řešení

Pro vytvoření aplikace jsme se rozhodovali mezi jazyky C++ a C#. Nakonec jsme zvolili jazyk C# protože sám o sobě obsahuje velké množství již vytvořených metod a také je velmi robustní. Jeho nevýhodou je větší náročnost na hardware počítače. Ovšem u takto jednoduché aplikace by to neměl být problém.

Za úkol bylo vytvořit aplikaci pro testování komunikace se senzorem a knihovnu funkcí pro automatickou komunikaci se senzorem.

Program byl napsán v prostředí Microsoft Visual Studio 2010. Aplikace pro testování komunikace umožňuje uživateli rychle otestovat funkčnost zapojení a komunikaci se senzorem. Po startu aplikace vybereme sériový port na který je připojen modul XBee, nastavíme komunikační parametry a otevřeme spojení. Z aplikace je možné přenastavit vlastnosti komunikačních periférií mikrokontroléru připojeného k senzoru měřící stolice. Po

nastavení komunikačních periférií můžeme příkazem Send data přeposlat libovolný povel senzoru. Odpověď senzoru se zobrazí v okně Recieved data spolu s časem přijetí. Program ukládá všechna přijatá data a také poslení nastavení modulu XBee. Tlačítkem Load last setup se zobrazí poslední aktivované nastavení modulu XBee. Další vlastnosti a funkce programu jsou popsány v helpu programu. Zde je ukázka vytvořené aplikace:



Obrázek 15: Ukázka aplikace pro ovládání bezdrátového systému

Knihovna funkcí obsahuje všechny metody nutné pro ovládání bezdrátového komunikačního systému. Tato knihovna byla napsána aby bylo možné pomocí jiného programu automaticky ovládat bezdrátovou komunikaci. Popis všech metod pro ovládání bezdrátového komunikačního systému se nachází v příloze.

Zdrojové kódy programů pro řízení mikrokontroléru a pro ovládání bezdrátové komunikace z PC jsou poměrně rozsáhlé a vkládání vybraných úseků programového kódu by jejich činnost příliš neosvětlilo. Principy nejdůležitějších funkcí byly vysvětleny v podkapitolách *softwarové řešení* umístěných v této kapitole. Pro simulaci funkcí programu mikrokontroléru je nutné mít nainstalovaný program AVRStudio. Pro spuštění řídicího programu napsaného v jazyce C# je třeba užít operačního systému MS Windows a mít nainstalovanou platformu .NET Framework verze 4. Pro testování funkcí řídicího programu ovládajícím bezdrátový komunikační systém z PC je třeba mít nainstalován program Microsoft Visual Studio 2010.

6.7 OVĚŘENÍ FUNKCE BEZDRÁTOVÉHO PŘENOSU DAT

Zařízení jsme testovali v normálních podmínkách (v obytných prostorech). Abychom ověřili správnost přenosu dat, propojili jsme na zařízení na straně senzoru piny zajišťující příjem dat ze senzoru a vysílání dat na senzor (piny MOSI a MISO, RxD a TxD). Díky tomu jsme data odeslaná na senzor dostali přijaty zpět. I při různých komunikačních rychlostech byl bezdrátový přenos dat naprosto bezchybný.

6.8 JAK BY SE DALO V PRÁCI POKRAČOVAT

Zařízení umožňuje provádět všechny požadované operace dané zadáním práce. Pokud by se ale mělo v práci pokračovat, zde je seznam věcí na kterých by bylo vhodné pracovat:

Zařízení na straně senzoru:

Bylo vhodné užít jiný mikrokontrolér. ATMEGA162 nemá ve svých perifériích integrován A/D převodník. Není tudíž možné měřit velikost napětí baterie a chránit ji tak před příliš velkým vybitím. ATMEGA162 sice obsahuje analogový komparátor. Díky němu by bylo možné měřit alespoň přechod mezi předem definovanými velikostmi napětí. Vývody analogového komparátoru mikrokontroléru ATMEGA162 nešťastně vyvedeny na stejných pinech jako je jedna z komunikačních jednotek USART. Použit integrovaný analogový komparátor není bez přerušení komunikace možné. O něco jednodušší řešení by bylo vytvořit analogový komparátor pomocí operačních zesilovačů a jejich výstup přivést na vyvedené vstupy PA7 či PA6.

Pro delší provoz bez nutnosti dobít baterii by bylo vhodné užít baterii s vyšším napájecím napětím a pro stabilizaci napětí užít spínaného zdroje. Ztráty na spínaném zdroji jsou minimální a tudíž by zařízení mohlo být v provozu déle bez nutnosti dobíjení baterie.

Mohly by být napsány knihovny pro ovládání různých senzorů připojených k mikrokontroléru. V současné době ještě nejsou známy typy senzorů se kterými by mělo zařízení komunikovat.

Zařízení na straně PC:

Místo užití sériové komunikace po rozhraní RS232 by bylo vhodnější užít FTDI převodník mezi USB a TTL logikou. Na plošném spoji by nemusel být obvod MAX232 a pro sériovou komunikaci i napájení by byl použit právě USB port.

ZÁVĚR

V rámci této práce byl vytvořen systém bezdrátového přenosu dat který sbírá data ze senzoru přes rozhraní USART či SPI a přeposílá je do PC. Bezdrátový systém je možno ovládat jak pomocí vizuální aplikace, tak pomocí knihovny funkcí. Přenos dat probíhá v normálních podmínkách prakticky bezchybně. Užitím komunikačních modulů XBee jsme si ušetřili práci s komunikačním protokolem, na druhou stranu se tím zvedla cena celého komunikačního zařízení.

Spotřeba komunikačního zařízení na straně senzoru je přibližně 75 mA, spotřeba komunikačního zařízení na straně PC je asi 50 mA. Rozměry zařízení jsou poměrně malé. Základní požadavky zadání této práce tedy byly splněny.

Standard ZigBee je vhodný pro jednoduchý a nepřiliš rychlý přenos dat v průmyslových aplikacích. Mezi jeho nesporné výhody patří protokol zajišťující správný přenos informací, možnost tvorby rozsáhlých sítí či nízká spotřeba komunikačních zařízení. Mezi jeho nevýhody patří hlavně vyšší pořizovací cena.

V elektronické příloze jsou uvedeny programy pro ovládání vytvořeného bezdrátového systému a veškerá dokumentace o tomto projektu.

LITERATURA

- [1] MATOUŠEK David.: *Práce s mikrokontroléry Atmel AVR ATmega 16 – 4. díl.* 1. vyd. Praha: BEN – technická literatura, 2006. 317 s. ISBN: 80-7300-174-8
- [2] VÁŇA Vladimír.: *Mikrokontroléry Atmel AVR programování v jazyce C.* 1. vyd. Praha: BEN – technická literatura, 2003. 206 s. ISBN: 80-7300-102-0
- [3] HRBÁČEK Jiří.: *Komunikace mikrokontroléru s okolím 1. díl.* 1. vyd. Praha: BEN – technická literatura, 1999. 152 s. ISBN: 80-86056-42-2
- [4] DIGI INTERNATIONAL: *XBee®/XBee-PRO® ZB RF Modules*[online]. Minnetonka, 2010. 155 s.
<http://ftp1.digi.com/support/documentation/90000976_C.pdf>
- [5] ČEJKA Miroslav.: *Elektronické měřicí systémy.* VUT Brno, 2002. 117 s.
- [6] SHARP John.: *Microsoft Visual C# 2010 krok za krokem.* 1. vyd. Brno: Computer PRESS, 2010. 687 s. ISBN: 978-80-251-3147-3
- [7] KADLEC Václav.: *Učíme se programovat v JAZYCE C.* 2. vyd. Brno: CP Books, 2005. 277 s. ISBN: 80-7226-715-9
- [8] ČERNÝ Viktor.: *Správa prvků bezdrátové sítě: diplomová práce.* Praha: ČVUT, katedra počítačů, 2009. 47 s. 0 příl. Vedoucí diplomové práce: Doc. Ing. Jan Janeček, CsC
- [9] HRBÁČEK Jiří.: *Komunikace mikrokontroléru s okolím 2. díl.* 1. vyd. Praha: BEN – technická literatura, 2006. 148 s. ISBN: 80-86056-73-2
- [10] TEXAS INSTRUMENTS: *MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS*[online]. 03.2004. 17 s.
<http://www.gme.cz/_dokumentace/dokumenty/329/329-036/dsh.329-036.1.pdf>
- [11] ATMEL®: *8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash*[online]. 07.2009. 324 s.
<http://www.atmel.com/dyn/resources/prod_documents/doc2513.pdf>
- [12] Příspěvatelé Wikipedie, *Bezdrátová komunikace* [online], Wikipedie: Otevřená encyklopedie, c2011, Datum poslední revize 2. 02. 2011, 14:17 UTC, [citováno 23. 05. 2011]
<http://cs.wikipedia.org/w/index.php?title=Bezdr%C3%A1tov%C3%A1_komunikace&oldid=6475527>

- [13] Příspěvatelé Wikipedie, *Modulace* [online], Wikipedie: Otevřená encyklopedie, c2010, Datum poslední revize 5. 10. 2010, 13:26 UTC, [citováno 23. 05. 2011]
<<http://cs.wikipedia.org/w/index.php?title=Modulace&oldid=5921468>>
- [14] Příspěvatelé Wikipedie, *Amplitudová modulace* [online], Wikipedie: Otevřená encyklopedie, c2011, Datum poslední revize 3. 05. 2011, 22:48 UTC, [citováno 23. 05. 2011]
<http://cs.wikipedia.org/w/index.php?title=Amplitudov%C3%A1_modulace&oldid=6857886>
- [15] Příspěvatelé Wikipedie, *Frekvenční modulace* [online], Wikipedie: Otevřená encyklopedie, c2011, Datum poslední revize 4. 05. 2011, 14:49 UTC, [citováno 23. 05. 2011]
<http://cs.wikipedia.org/w/index.php?title=Frekven%C4%8Dn%C3%AD_modulace&oldid=6860035>
- [16] Příspěvatelé Wikipedie, *Bluetooth* [online], Wikipedie: Otevřená encyklopedie, c2011, Datum poslední revize 13. 05. 2011, 10:02 UTC, [citováno 23. 05. 2011]
<<http://cs.wikipedia.org/w/index.php?title=Bluetooth&oldid=6893028>>
- [17] Příspěvatelé Wikipedie, *ZigBee* [online], Wikipedie: Otevřená encyklopedie, c2010, Datum poslední revize 13. 04. 2010, 10:40 UTC, [citováno 23. 05. 2011]
<<http://cs.wikipedia.org/w/index.php?title=ZigBee&oldid=5218445>>
- [18] Příspěvatelé Wikipedie, *Wi-Fi* [online], Wikipedie: Otevřená encyklopedie, c2011, Datum poslední revize 19. 05. 2011, 10:49 UTC, [citováno 23. 05. 2011]
<<http://cs.wikipedia.org/w/index.php?title=Wi-Fi&oldid=6915194>>
- [19] DIGI INTERNATIONAL: *XBee® & XBee-PRO® ZB* [online], [citováno 23. 05. 2011]
<http://www.digi.com/pdf/ds_xbeezbmodules.pdf>
- [20] DIGI INTERNATIONAL: *XBee-PRO® 868* [online], [citováno 23. 05. 2011]
<http://www.digi.com/pdf/ds_xbeepr868.pdf>
- [21] TEXAS INSTRUMENTS: *CC2430 Data Sheet (rev. 2.1)* [online], [citováno 23. 05. 2011]
<<http://www.farnell.com/datasheets/38167.pdf>>
- [22] ZYXEL: *NBG-417N Wireless N-lite Home Router: User's Guide*[online]. 2. vyd. 4.2009. [citováno 23. 05. 2011]
<ftp://ftp.zyxel.com/NBG-417N/user_guide/NBG-417N_1_Ed1.pdf>
- [23] RAYSON: *Bluetooth® Module BTM-112*[online]. [citováno 23. 05. 2011]
<http://www.sparkfun.com/datasheets/Wireless/Bluetooth/BTM112_wATcommands.pdf>
- [24] Příspěvatelé Wikipedie, *Cyklický redundantní součet* [online], Wikipedie: Otevřená encyklopedie, c2011, Datum poslední revize 4. 01. 2011, 20:02 UTC, [citováno 23. 05. 2011]
<http://cs.wikipedia.org/w/index.php?title=Cyklick%C3%BD_redundantn%C3%AD_sou%C4%8Det&oldid=6297511>

SEZNAM PŘÍLOH

PŘÍLOHY V TÉTO PRÁCI:

- Popis funkcí knihovny pro ovládání komunikačního zařízení
- Osazovací plán
- Schéma DPS
- Fotografie vytvořeného komunikačního zařízení

PŘÍLOHY NA DVD:

- Tato práce v pdf souboru
- Program mikrokontroléru ATMEGA162
- Program pro PC ovládající komunikační systém
- Knihovna funkcí pro ovládání komunikačního systému
- Manuál pro ovládání modulu XBee
- Schéma a deska plošných spojů obou částí komunikačního zařízení vytvořená v programu Eagle

Popis funkcí knihovny pro ovládání komunikačního zařízení:

Pro užití knihovny XBee_library je třeba ji v programu vytvořit příkazem:

```
XBee_Library xBee = new XBee_Library();
```

Zde je popis jednotlivých metod:

```
public void initialization()
```

Tato metoda je klíčová pro správnou funkci ostatních metod. Je třeba ji zavolat před užitím jiných metod zde popsanych. Plní počáteční hodnoty proměnných a vytváří a aktivuje časovač.

```
public string openSerialPort(string portName, Int32 baudRate)
```

Metoda nastavující parametry komunikace po sériovém portu. Je třeba předat jí jméno sériového portu který pro komunikaci uijeme a komunikační rychlost v Baudech. Pro kompatibilitu se zbytkem komunikačního systému jsou definovány tyto komunikační rychlosti: 4800, 9600, 19200, 38400, 57600, 115200. V případě úspěchu vrací funkce string "OK", v případě neúspěchu vrací string "ERROR".

```
public string closeSerialPort()
```

Užitím této metody ukončíme komunikaci po sériovém portu. Sériový port bude uvolněn a může být užíván jiným programem. V případě úspěchu vrací funkce string "OK", v případě neúspěchu vrací string "ERROR".

```
public string LEDDiodesControl(string command)
```

Tato metoda umožňuje rosvícení nebo zhasnutí LED diod na straně komunikačního zařízení u senzoru. Předáme-li jí parametr "TurnOn", diody se rozsvítí. Při předání parametru "TurnOff" diody zhasnou. V případě úspěchu vrací funkce string "OK", v případě neúspěchu vrací string "ERROR".

```
public string SensorUSART_Data(string command)
```

Metoda povolující nebo zakazující přeposílání dat ze senzoru připojenému ke komunikační jednotce USART do PC. Při užití parametru "Start" je přeposílání dat ze senzoru do PC povoleno, při použití parametru "Stop" je přeposílání dat zakázáno. V případě úspěchu vrací funkce string "OK", v případě neúspěchu vrací string "ERROR".

```
public string SensorUSARTBaudrate(Int32 baudRate)
```

Jedná se o metodu měnící komunikační rychlost senzoru s mikrokontrolérem připojenému ke komunikační jednotce USART. Pro kompatibilitu se zbytkem komunikačního systému jsou definovány tyto komunikační rychlosti: 4800, 9600, 19200, 38400, 57600, 115200. V případě úspěchu vrací funkce string "OK", v případě neúspěchu vrací string "ERROR".

```
public string XBeeUSARTBaudrate(Int32 baudRate)
```

Tato metoda mění komunikační rychlost mikrokontroléru s modulem XBee k němu připojeným. Pro kompatibilitu se zbytkem komunikačního systému jsou definovány tyto komunikační rychlosti: 4800, 9600, 19200, 38400, 57600, 115200. V případě úspěchu vrací funkce string "OK", v případě neúspěchu vrací string "ERROR".

```
public string PC_XBeeSetup(Int32 baudRate)
```

Užitím této metody změníte komunikační rychlost mezi modulem XBee a počítačem. Pro kompatibilitu se zbytkem komunikačního systému jsou definovány tyto komunikační rychlosti: 4800, 9600, 19200, 38400, 57600, 115200. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string SPI_Data(string command)
```

Metoda povolující či zakazující přeposílání dat ze senzoru připojenému k SPI komunikační jednotce do PC. Při užití parametru “Start” je přeposílání dat ze senzoru do PC povoleno, při použití parametru “Stop” je přeposílání dat zakázáno. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string SPI_bitSequence(string bitSequence)
```

Tato metoda určuje zda-li vysíláme a přijímáme data v režimu MSB nebo LSB. Užitím parametru “LSB” nastavíme LSB režim, při užití parametru “MSB” užíváme MSB režim. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string SPI_dataAndClock(string CPOL, string CPHA)
```

Metoda určující nastavení fáze hodin a dat při užití SPI rozhraní. První parametr má varianty “CPOL0” a “CPOL1”, druhý parametr má varianty “CPHA0” a “CPHA1”. Existují tedy 4 různá nastavení. Parametry CPOL a CPHA přímo odpovídají parametrům v mikrokontroléru ATMEGA. Pro správné pochopení funkce těchto parametrů se podívejte do datasheetu pro mikrokontroléry ATMEGA. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string SPI_frequency(string frequency)
```

Metoda určující komunikační rychlost komunikace po SPI. Rychlost je dána násobky frekvence hodin mikrokontroléru. Násobky mají samozřejmě velikost menší než 1. Parametry této metody jsou: “F0/4”, “F0/16”, “F0/64”, “F0/128”, kde F0 je frekvence hodin mikrokontroléru. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string loadSPISetupToMicrocontroller()
```

Tato metoda aktivuje nastavení parametrů SPI komunikačního rozhraní. Užití metod SPI_startSendingData, SPI_bitSequence, SPI_dataAndClock, SPI_frequency bez zavolání této metody tedy nemá žádný účinek. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string sendData(string dataToSend)
```

Tato metoda posílá data na senzor. Předávaný parametr dataToSend jsou právě data která budou senzoru přeposlána. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

```
public string recieveStoredData()
```

Metoda vracející přijatá data ze senzoru. Přijatá data se postupně ukládají do zásobníku který má velikost 1024 Bytů. Možnost přetečení zásobníku je softwarově ošetřena, při málo častém čtení dat hrozí jejich ztráta.

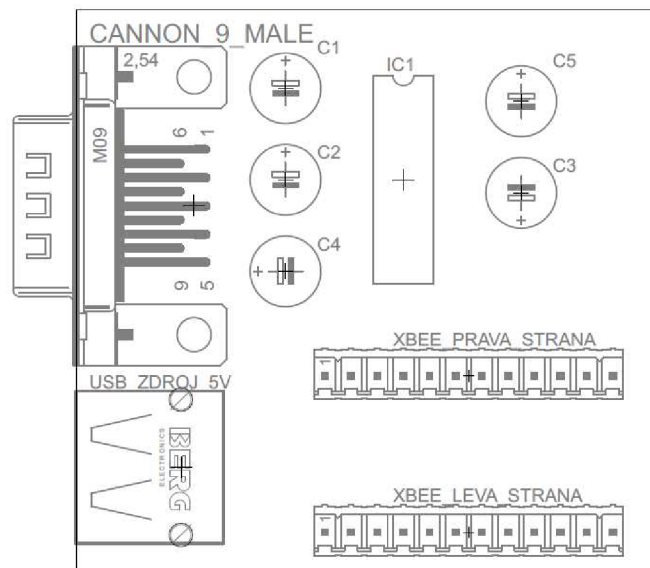
```
public string recieveStoredSystemData()
```

Metoda vracející přijatá data z mikrokontroléru. Data z mikrokontroléru musí mít následující tvar: ve svém prvním znaku mají hlavičku (ASCII znak 0xAA), v druhém znaku velikost řetězce a v ostatních znacích samotnou zprávu. Jelikož momentálně nejsou implementovány žádné funkce díky kterým by mikrokontrolér vracel data jiná než ze senzoru, tato metoda nemá momentálně využití. V případě budoucího rozvoje práce ale může být užita (například pro měření napětí baterie).

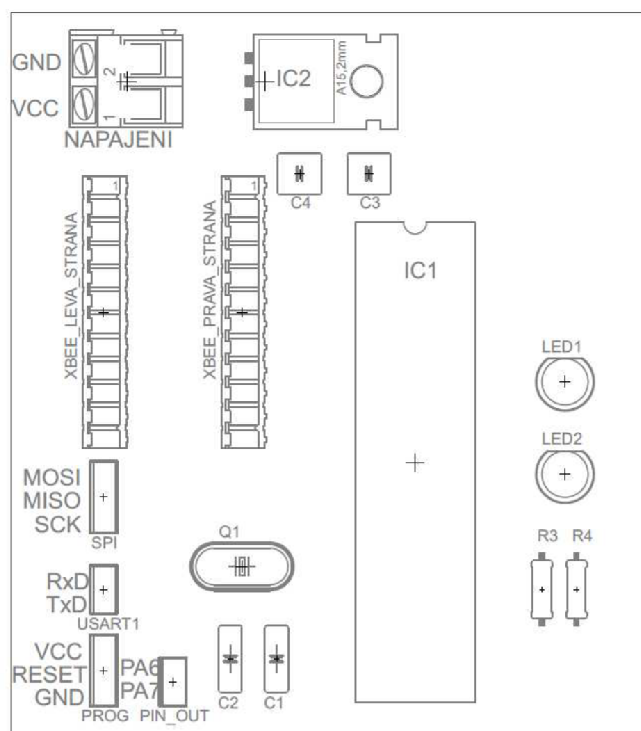
```
public string sendTestData(string command)
```

Tato metoda povoluje nebo zakazuje posílání nekonečného řetězce testovacích dat. Jedná se o cyklické posílání čísel 0-9 v ASCII znacích. Tato metoda slouží pro testování spolehlivosti komunikace mezi XBee moduly. Při užití parametru “Start” začne mikrokontrolér posílat počítači řetězec čísel. Při užití parametru “Stop” mikrokontrolér přestane tato data posílat. V případě úspěchu vrací funkce string “OK”, v případě neúspěchu vrací string “ERROR”.

Osazovací plán:

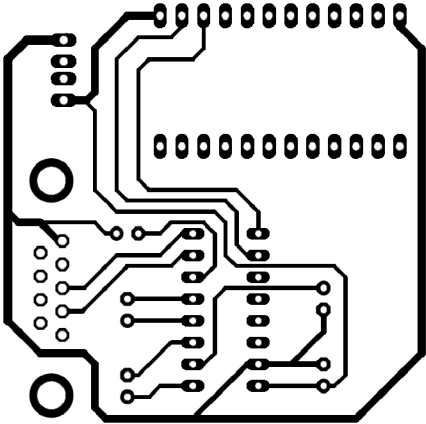


Obrázek 16: Osazovací plán zařízení na straně PC

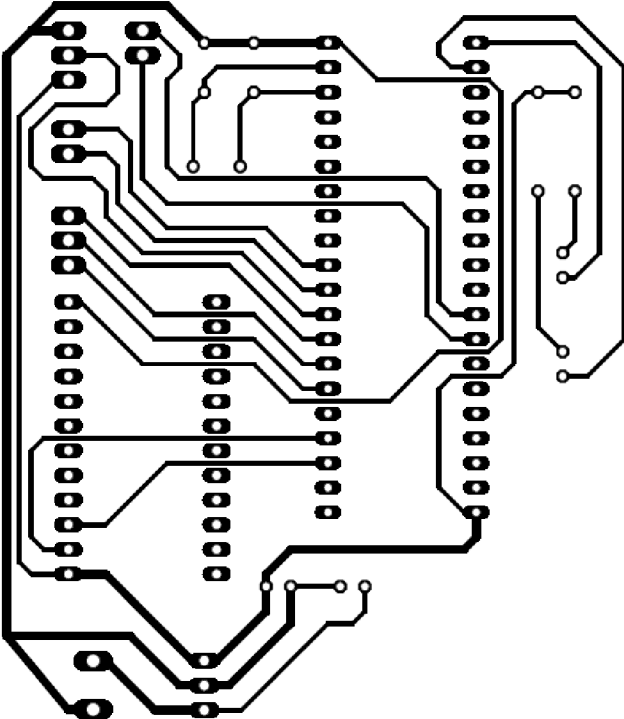


Obrázek 17: Osazovací plán zařízení na straně senzoru

Schéma DPS:

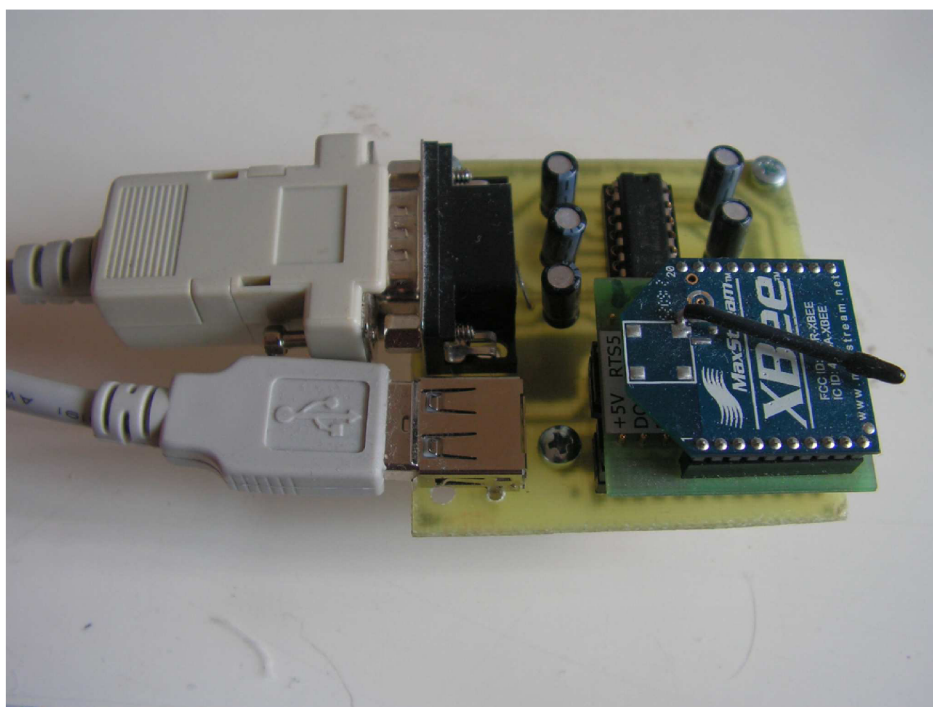


Obrázek 18: DPS komunikačního zařízení na straně PC



Obrázek 19: DPS komunikačního zařízení na straně senzoru

Fotografie komunikačního zařízení:



Obrázek 20: Fotografie komunikačního zařízení na straně PC



Obrázek 21: Fotografie komunikačního zařízení na straně senzoru