



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV MIKROELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF MICROELECTRONICS

UNIVERZÁLNÍ MODUL PRO PŘENOS NAMĚŘENÝCH DAT PROSTŘEDNICTVÍM WIFI

UNIVERSAL MODULE FOR SENSORIC DATA INTERCHANGE VIA WIFI

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JURAJ VITEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. LADISLAV MACHÁŇ

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav mikroelektroniky

Diplomová práce

magisterský navazující studijní obor
Mikroelektronika

Student: Bc. Juraj Vitek

ID: 125700

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Univerzální modul pro přenos naměřených dat prostřednictvím WiFi

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a realizujte univerzální modul, který umožní bezdrátový přenos dat prostřednictvím sítě WiFi. Modul vybavte vstupními branami, které umožní připojit jak senzory s analogovým výstupem, tak senzory typu SMART s výstupem digitálním. Modul by měl rovněž obsahovat brány výstupní, kvůli možnosti vzdálené konfigurace měřicích podmínek. Vstupní i výstupní brány musí být tolerantní k 5 V logice. Součástí práce je rovněž programové vybavení, které umožní jednoduchou, uživatelsky přívětivou konfiguraci modulu. Pro demonstraci funkce modulu vyberte dva různé druhy senzorů teploty (analogový, digitální), pro které modul nakonfigurujte.

DOPORUČENÁ LITERATURA:

Dle pokynů vedoucího práce

Termín zadání: 10.2.2014

Termín odevzdání: 29.5.2014

Vedoucí práce: Ing. Ladislav Macháň

Konzultanti diplomové práce:

prof. Ing. Vladislav Musil, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

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

Abstrakt

Táto práca opisuje problematiku prenosu dát pomocou bezdrôtového prenosu. Rozoberá teoreticky problematiku počítačových sietí, základne prenosové protokoly a ich použitie. Ďalej sú opísané štandardy a mechanizmy bezdrôtového prenosu a podrobne popísaný postup prenosu dát pomocou WiFi siete. V praktickej časti je opísaný prvotný prototyp a základné funkcie prototypu ako aj jednotlivých komponentov.

Kľúčové slová

Siete, bezdrôtová komunikácia, mikrokontrolér, dáta, bit

Abstract

This work describes the problems of data transmission via wireless networks. We discuss the theoretical background of computer networks, the base transmission protocols and their use. We also describe standards and mechanisms for wireless transmission and described in detail the procedure of data transfer via WiFi network. In practical part we describe prototype and the basic functions of the prototype its components.

Key words

Networks, wireless communication, mikrokontroler, data bit

VITEK, J. *Univerzální modul pro přenos naměřených dat prostřednictvím WiFi*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 57 s. Vedoucí diplomové práce Ing. Ladislav Macháň.

Prehlásenie autora o pôvode diela

Prehlasujem, že svoju diplomovú prácu na tému „Univerzální modul pro přenos naměřených dat prostřednictvím WiFi“ som vypracoval samostatne pod vedením vedúceho semestrálneho projektu a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej prehlasujem, že v súvislosti s vytvorením tohto projektu som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovení § 152 trestného zákona č. 140/1961 Sb.

V Brne dňa 29.5.2014

.....

podpis autora

PodĎakovanie:

Ďakujem vedúcemu diplomovej práce Ing. Ladislavovi Macháňovi za metodické a cielene orientované vedenie pri plnení úloh spojených s diplomovou prácou. Ďalej ďakujem spolupracujúcej firme Honeywell, sol. s.r.o, za poskytnutie priestoru k realizácii experimentálnych prác a vývojovému pracovníkovi tejto firmy Ing. Ondřejovi Pavelkovi za poskytnutú metodickú pomoc a odborné rady.

Obsah

1	Úvod	6
2	Teoretický úvod	6
2.1	Siete	6
2.2	Architektúra lokálnych sietí IEEE 802	9
2.3	Bezdrôtové siete	15
2.4	Štandard IEEE 802.11	16
2.5	Štandard IEEE 802.11b	20
3	Praktická časť	23
3.1	Použité súčiastky	23
3.1.1	WiFi modul: RN-171-XV 802.11 b	23
3.1.2	Vývojová doska EvB 5.1	24
3.1.3	Mikrokontroler ATmega 32:	26
3.1.4	Prevodník napät'ových úrovní TXB0106	26
3.2	Návrh hardwarového konceptu pre modul WiFly	27
3.3	Popis softwarového konceptu	28
3.3.1	Popis rozhrania UART:	28
3.4	Inicializácia modulu	31
3.4.1	Zabezpečenie bezdrôtovej komunikácie	34
3.5	Vytvorenie počítačovej aplikácie	35
3.5.1	Nadviazanie TCP komunikácie	36
3.5.2	Popis záložky TCP-IP console	39
3.5.3	Popis záložky I/O config	40
3.5.4	Popis záložky Monitor	41
3.6	Popis komunikácie a algoritmov pre čítanie a zápis	42
3.6.1	Tlačidlo Send	42
3.6.2	Tlačidlo Read config	43
3.6.3	Tlačidlo Write config	45
3.6.4	Tlačidlo pre zápis výstupu	47
3.6.5	Monitorovanie dát	49
4	Záver	52
5	Zoznam použitých literárnych zdrojov	53
6	Zoznam Obrázkov	54
7	Zoznam príloh	55
8	Prílohy	55

1 ÚVOD

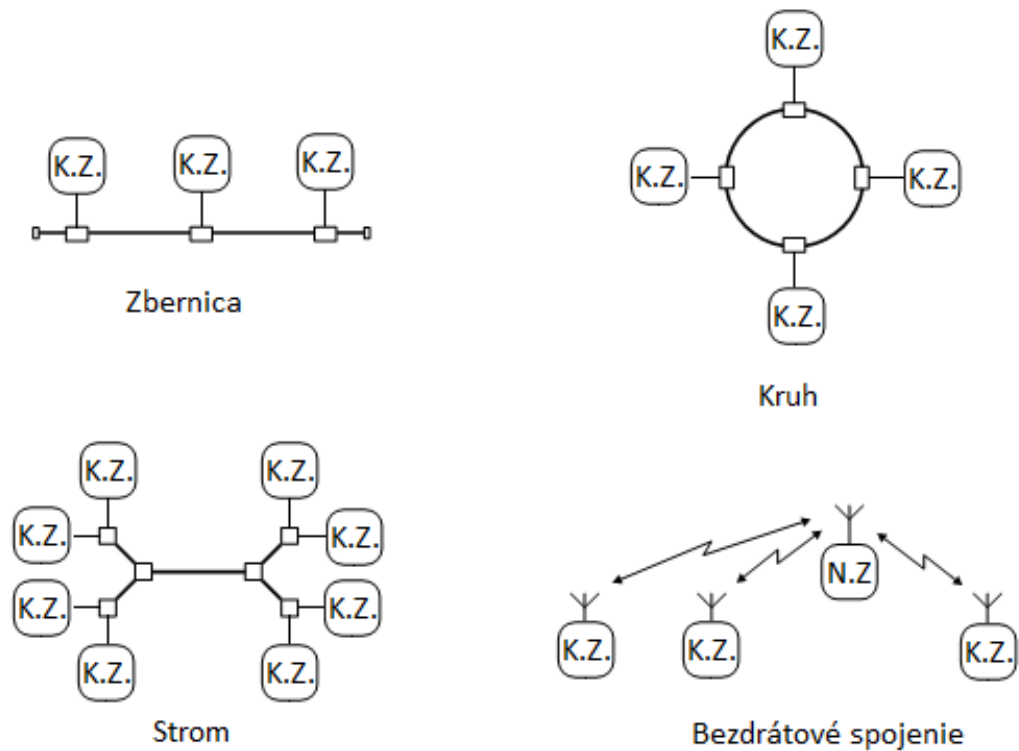
Práca opisuje problematiku prenosu dát pomocou bezdrôtového prenosu. Na danom projekte sa spolupracuje s externou firmou preto je tento prenos dát obmedzený najmä na WiFi komunikáciu a konkrétne na bezdrôtový modul RN-171-XV 802.11 b. Pre tento modul je potrebné vytvoriť základné nástroje pre vývoj a vybrať vhodný hardware. Je nutné taktiež pochopiť základy sieťovej komunikácie a najmä bezdrôtového prenosu dát. Vstupné dáta ako aj konkrétne použitie nie sú zatiaľ presne špecifikované a preto je potrebné hardware voliť tak, aby bol kompatibilný s čo možno najuniverzálnejší pre rôzne senzory, prípadne digitálny signál. V priemysle je možné použiť rôzne senzory pre snímanie rôznych elektrických a neelektrických veličín, s rôznymi výstupmi, ako napríklad analógový výstup, digitálny, prúdová slučka prípadne frekvenčný výstup. Preto sa v tejto práci zameriam najmä na univerzálnosť riešenia.

2 TEORETICKÝ ÚVOD

V tejto kapitole sa budem venovať samotnej sieťovej topológii a možnostiam prenosu dát pomocou bezdrôtovej komunikácie a s tým súvisiacimi štandardami a normami.

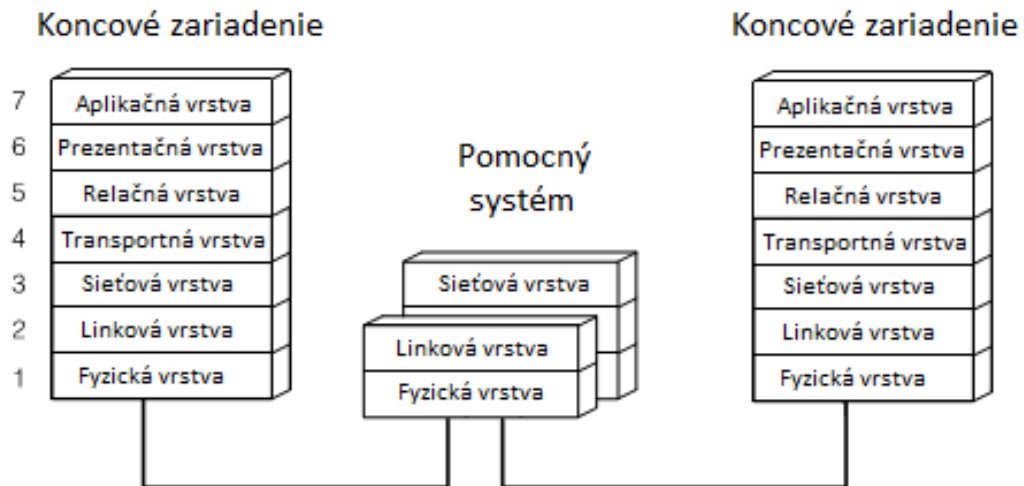
2.1 Siete

Počítačové siete (ďalej skrátene len siete) možno definovať ako entitu niekoľkých zariadení ktoré sa navzájom snažia vymieňať informácie. Vo svete počítačovej techniky ide prevažne o zariadenia schopné komunikovať pomocou štandardu pre súčasné ethernetové siete. Rozličné možnosti prepojenia jednotlivých zariadení vytvára topológiu, ktorej zložitosť závisí od obsahu prvkov. Topológia siete zachytáva reálne zapojenie. Topológie sietí sa opierajú o prechod správ (paketov) a jeho postupný posun po jednotlivých uzloch siete. Klasické miestne siete využívajú prepojenia na komunikačné stanice spoločným vodičom, signál vyslaný jedným zariadením je prijatý ostatnými zariadeniami na rovnakom vodiči. Tento typ spojenia sa označuje ako "broadcast" siete. Voľba správnej topológie má veľký vplyv na vlastnosti siete ako napríklad: možnosť a schopnosť pridávať nové zariadenia do siete, možnosť modifikácie, spoľahlivosť, odolnosť voči výpadkom, výkonnosť využitie, prenosová kapacita, spomalenie správ. Najčastejšie sa používajú topológie typu zbernica, kruh, strom a bezdrôtové spojenie čo je principiálne načrtnuté na Obr.2.1. [1]



Obr. 2.1: Ukážka topológií sietí [1]

Pre komunikáciu medzi zariadeniami je potreba definovať protokol pre komunikáciu. V súčasnej dobe je pre počítačové siete definované niekoľko štandardov, tieto štandardy sú definované organizáciami ako IEEE (Institute of Electrical and Electronics Engineers), ETSI (European Telecommunications Standards Institute), ITU-T (International Telecommunication Union - Telecommunication Standardization Sector), ANSI (American National Standards Institute) a ISO (International Organization for Standardization). Do týchto skupín spadá štandard ATM (Asynchronous Transfer Mode) pre hlavné siete a definujú aj architektúru bezdrôtových sietí. Protokol pre komunikáciu medzi počítačmi sa vytvoril na začiatku osemdesiatich rokov minulého storočia a stal sa štandardom v dnešných sieťach. Jedná sa o model rozdeľujúci jednotlivé funkčné vrstvy, kde každá vyššia vrstva rozširuje možnosti predošlej vrstvy. Jedná sa o ISO/OSI (ISO Open Systems Interconnection) model. Architektúra modelu je zobrazená na Obr.2.2. [1]



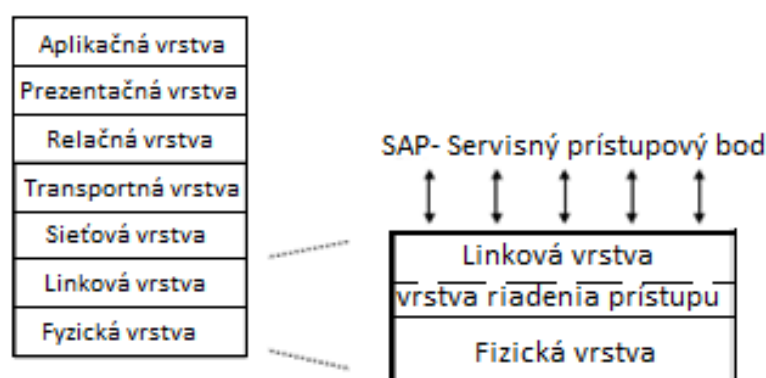
Obr. 2.2: ISO/OSI model [1]

- **Fyzická vrstva** (Physical Layer) definuje fyzické prepojenie medzi prvkami siete. Zaoberá sa konektormi, elektrickými vlastnosťami (napät'ové úrovne, pôsob kódovania modulácie) a spôsob pripojenia prvku k prenosovému médiu.
- **Linková vrstva** (Data Link Layer) definuje pravidla pre prijímanie presne definovaných blokov dát, a kontrolujú prijaté dáta pred chybami prenosu. V prípade, že je na jednej úrovni viacero zariadení je linková vrstva zodpovedná za filtráciu primaných správ.
- **Sieťová vrstva** (Network Layer) definuje spôsob, akým si sieť posielajú správy, ako si je jednotlivé prvky siete posielajú správy od odosielateľa k adresátovi, pričom využívajú sieťovú adresáciu jednotlivých prvkov.
- **Transportná vrstva** (Transport Layer) zabezpečuje viacero paralelných komunikácií na jednom zariadení, vytvára dočasnú komunikačnú linku medzi prvkami. Na tejto úrovni sa jednotlivé pakety rozkladajú na správy a prijaté správy skladajú späť na pakety.
- **Relačná vrstva** (Session Layer) vytvára komunikačný kanál a zabezpečuje aplikačnú úroveň jednotlivých programov.
- **Prezentačná vrstva** (Presentation Layer) transformuje prenášané dáta a to tým spôsobom že vytvára ich kódovanie a kompresiu.
- **Aplikačná vrstva** (Application Layer) jedná sa o vrstvu aplikačných programov a prehliadačov ktoré túto vrstvu využívajú. [1]

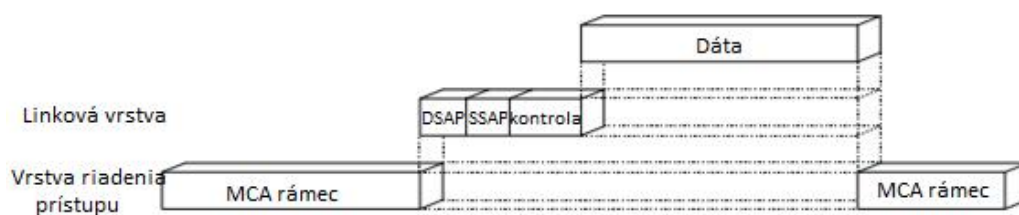
2.2 Architektúra lokálnych sietí IEEE 802

Po vytvorení ISO/OSI modelu nastalo úsilie o vytvorenie štandardu pre sieťovú komunikáciu. Organizácia IEEE vytvorila univerzálny štandard IEEE 802 pre lokálne dátové komunikácie. V roku 1983 sa definícia štandardu obmedzila na technológiu ethernetu. Štandard sa zaoberá najmä tromi najnižšími vrstvami OSI modelu vrstvu fyzickú, linkovú a čiastočne sieťovou. [1]

Doporučene IEEE 802 člení ISO model odlišne. Vytvára vrstvu fyzickú, ktorá definuje média, konektory, signály, vrstva riadenia prístupu ku komunikačnému kanálu MAC (Medium Access Control). Táto adresa predchádza chybám. Ďalšou vrstvou je Linková vrstva LLC (Logical Link Control). Táto vrstva umožňuje viacero komunikácii na jednom kanáli. Vnútorňá štruktúra je zobrazená na Obr.2.3, a príslušný rámec na Obr.2.4. [1]



Obr. 2.3: ISO model podľa štandardu IEEE 802 [1]



Obr. 2.4: Štruktúra rámcov IEEE 802 [1]

Odporúčenie IEEE 802.1 zastrešuje ostatné odporúčenia, definuje MAC adresy a prepojenia v lokálnej sieti. [1]

Odporúčenie IEEE 802.2 definuje linkovú vrstvu a služby ktoré daná sieť obsahuje. jedná sa o dva základné druhy služieb, o nepotvrzovanú datagramovú službu (Connectionless Service) a potvrzovanou datagramovou službu. [1]

Odporúčenie IEEE 802.3, 802.4, 802.5, 802.6, 802.11, 802.12, 802.14, 802.15, 802.16 a 802.17 popisujú fyzickú vrstvu a prístup k médiu pre lokálnu sieť rôzneho typu ako napríklad: pre zbernicové lokálne siete, kruhové lokálne siete IBM Token Ring, bezdrôtové

siete WLAN, širokopásmové siete, osobné bezdrôtové siete a iné. [1]

Odporúčenie IEEE 802.7, 802.8, 802.9 a 802.10 sa zaoberá širokopásmovými kanálmi, optickými vláknami, ochrany dát a bezpečnosti v lokálnych sieťach. [1]

Pochopenie IP adresy:

IP adresa je adresa používaná za účelom jednoznačnej identifikácii zariadení v IP sieti. Adresa sa skladá z 32 bitov, ktoré reprezentujú sieťovú časť a užívateľa. Informácia o zložení IP adresy je uložená v maske podsiete. IP adresa je zložená s 32 bitov, ktoré sú rozdelené do štyroch oktetov (1 oktet = 8 bitov). Jednotlivé oktety sa prezentujú v desiatkovej sústave a sú delené bodkou a preto je IP adresa vyjadrením v desatinnom tvare (napríklad 172.16.81.100). [1], [2]

Príklad prevodu binárneho oktetu do desiatkového čísla: každý binárny byt v desiatkovej sústave nadobúda hodnotu 2^n pričom n označuje pozíciu daného bitu. N nadobúda hodnoty od 0 do 7 a poradie je určené od najmenej významného bitu (z pravej strany). [1], [2]

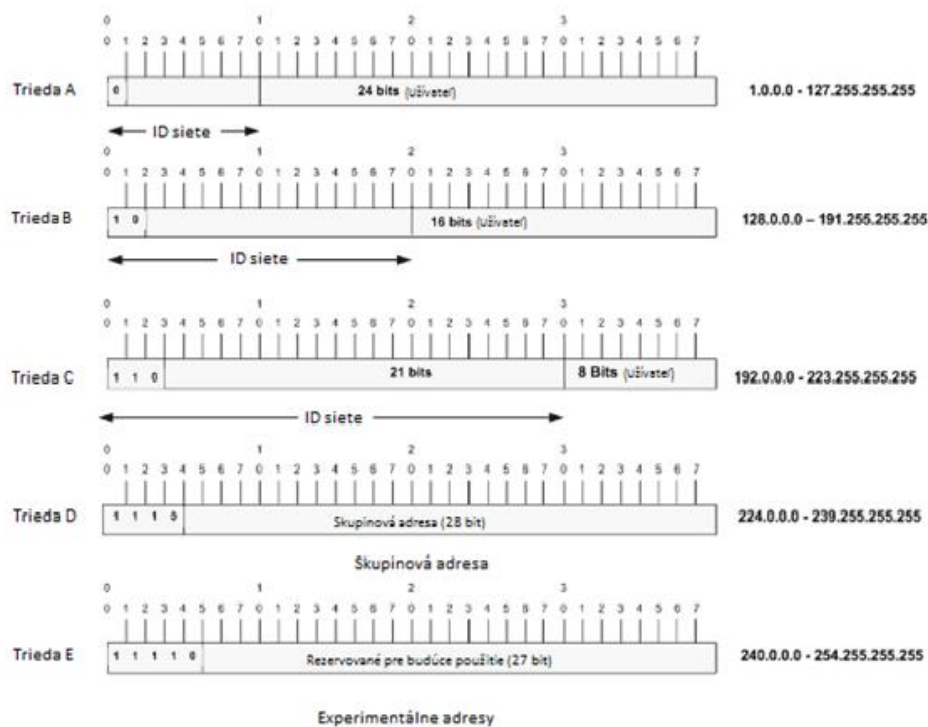
11	1	1	1	1	1	1	
128	64	32	16	8	4	2	1

(128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255)

Príklad IP adresy vyjadrenej v desiatkovej a binárnej sústave.

10.	1.	23.	19	(v desiatkovej sústave)
00001010.	00000001.	00010111.	00010011	(v binárnej sústave)

Tieto adresy je možno rozdeliť do podsietí tak, aby mohli adresovať ako malé, tak aj veľké siete nakoľko každý prvok v sieti musí mať pridelenú vlastnú unikátnu adresu. Pre klasifikáciu sietí podľa rozsahu je možné siete rozdeliť na 5 základných tried od A až po E. Obrázok ilustruje možné použitie tried sietí. Tieto triedy sú zobrazené na Obr.2.5. [1], [2]



Obr. 2.5: Triedy IP adres [2]

V triede A určuje prvý oktet sieť a nasledovné oktety určujú koncové zariadenie v sieti. V tomto prípade sa jedná o adresný priestor pre zariadenia až 24 bitov. Tento adresný priestor je možné použiť pre sieť v rozsahu až do 16777214 koncových zariadení. V triede B sa jedná o prvé dva oktety určujúce sieťovú časť adresy. Adresný priestor pre zariadenia je 16 bitov čo umožňuje pripojiť až a 65534 koncových zariadení. Podobná situácia nastáva v triede C, kde prvé 3 oktety definujú sieť a posledný oktet definuje adresný priestor pre zariadenia. V tomto prípade je na sieť možné pripojiť až 254 koncových zariadení. [1], [2]

Sieťové masky označujú ktorá časť IP adresy je určená pre adresovanie siete a koncového užívateľa. Pre sieťové triedy A , B a C existujú predvolené masky:

Trieda A : 255.0.0.0

Trieda B : 255.255.0.0

Trieda C : 255.255.255.0

Príklad IP adresy triedy A , ktorá nebola podsieťovaná má adresu : 8.20.15.1 255.0.0.0. K jednotlivým bitom IP adresy prislúchajú bity masky podsiete. Pokiaľ uvažujeme o týchto dvojiciach bytov nájdeme dve skupiny, a to skupinu ktorej maska podsiete je vždy nastavená na 1 a na dvojice bitov, ktorých maska podsiete je vždy nastavená na 0. V prípade že maska podsiete je nastavená na 1 odpovedajúce bity IP adresy vyjadrujú adresu siete. V opačnom prípade sa jedná o adresu zariadenia. [1], [2]

Príklad:

8.20.15.1 = 00001000.00010100.00001111.00000001

255.0.0.0 = 11111111.00000000.00000000.00000000

Adresa siete | adresa zariadenia

Adresa siete = 00001000 = 8

adresa zariadenia = 00010100.00001111.00000001 = 20.15.1

Podsiet'ovanie:

Podsiet'ovanie umožňuje vytvoriť viac logických sietí, ktoré existujú v rámci jednej siete triedy A, B, alebo C. Idea delenia siete na menšie skupiny vychádza z nutnosti jedinečnej adresy siete a organizácie pripojených zariadení na rovnakej logickej úrovni. Pri podsiet'ovaní má každá dielčia sieť unikátnu adresu a navzájom tvoria hierarchiu menších sietí. Aby toto bolo možné je potreba rozšíriť masku siete niekoľkými bitmi na úkor adresy zariadenia. Takto je možné vytvoriť unikátny názov siete. Napríklad adresa triedy C 204.17.5.0 s maskou podsiete 255.255.255.0 môže nadobúdať nasledovné podsiete[1], [2]

204.17.5.0	- 11001100.	00010001.	00000101.	00000000
255.255.255.224	- 11111111.	11111111.	11111111.	11100000

Po rozšírení masky na 255.255.255.224, sme rozšírili adresu siete o tri bity. Tieto bity možno nazvať aj adresa podsiete nakoľko spadajú do pôvodnej siete. S adresou podsiete o veľkosti tri bity je možné vytvoriť 8 podsietí pričom každá podsieť môže obsahovať až 30 koncových zariadení (ostávajúcich 5 bitov je možné adresovať ďalších 32 zariadení, ale v sieti sa vyskytujú dve rezervované adresy ako adresa siete a broadcast adresa) [1], [2]

204.17.5.0	255.255.255.224	rozsah adries koncových zariadení 1 do 30
204.17.5.32	255.255.255.224	rozsah adries koncových zariadení 33 - 62
204.17.5.64	255.255.255.224	rozsah adries koncových zariadení 65-94
204.17.5.96	255.255.255.224	rozsah adries koncových zariadení 97-126
204.17.5.128.	255.255.255.224	rozsah adries koncových zariadení 129-158
204.17.5.160	255.255.255.224	rozsah adries koncových zariadení 161-190
204.17.5.192	255.255.255.224	rozsah adries koncových zariadení 193-222
204.17.5.224	255.255.255.224	rozsah adries koncových zariadení 225-254

Ako je zrejmé z príkladu, maska podsiete 255.255.255.224 je väčšia než štandardná maska siete triedy C. Obsahuje 27 bitov a preto môže byť tiež označovaná ako /27.

Vzhľadom k tomu, že teraz používame 4 bity pre adresu podsiete, máme iba štyri bity vyhradené pre adresy hostiteľa. Takže v tomto prípade môžeme mať až 16 podsietí, z ktorých každá môže mať maximálne 14 koncových zariadení. Na základe tejto skutočnosti je jasné, že čím je väčšia adresa siete je možné vytvoriť viacero podsietí s menej koncovými zariadeniami. [1], [2]

Dynamické pridelovanie IP adries (DHCP- Dynamic Host Configuration):

Pri dynamickom pridelovaní adries sa predpokladá že koncové zariadenie nemá nakonfigurovanú žiadnu IP adresu, a pre pripojenie do siete je potrebné, aby si adresu vyžiadal zo serveru adries. Tento proces definuje protokol DHCP ktorý poskytuje mechanizmus, prostredníctvom ktorého počítačov, ktoré používajú TCP / IP protokol môžu prijať konfiguračné nastavenia automaticky prostredníctvom siete. Tento protokol je založený na architektúre klient - server, kde klient je koncové zariadenie, a DHCP server je zariadenie schopné na základe požiadavky spoľahlivo priradiť konfiguračné nastavenie. Pri nadväzovaní komunikácie je potrebné definovať základné príkazy pre server a koncové zariadenie. Príkazy sú zhrnuté v nasledovnej tabuľke. [1], [3]

Tab. 1: Tabuľka DHCP správ

Kód	Správa
0x01	DHCPDISCOVER
0x02	DHCPOFFER
0x03	DHCPREQUEST
0x04	DHCPDECLINE
0x05	DHCPACK
0x06	DHCPNAK
0x07	DHCPRELEASE
0x08	DHCPINFORM

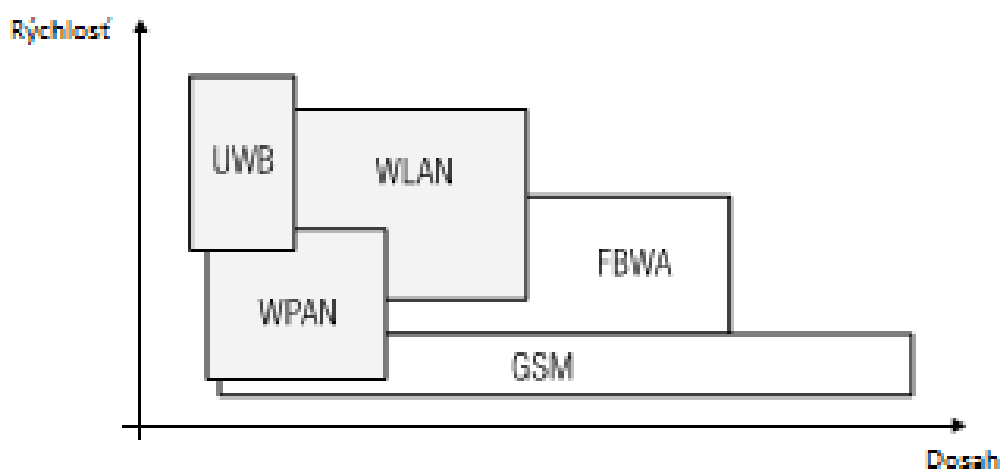
- DHCPDISCOVER: Po inicializácii zariadenia bez konfiguračných nastavení vyšle správu DHCPDISCOVER na lokálnej sieti. Zariadenie nemalo žiadne predchádzajúce nastavenia a preto túto správu posielala všetkým zariadeniam v jeho lokálnej sieti. Ako cieľovú IP adresu používa adresu 255.255.255.255 a ako zdrojovú adresu 0.0.0.0. Ak existuje DHCP server na tejto lokálnej sieti, server vyšle správu DHCPOFFER.
- DHCPOFFER: DHCP server, ktorý prijme správu DHCPDISCOVER obvykle reaguje správou DHCPOFFER. Pokiaľ koncové zariadenie prijme túto správu

prejde do režimu výberu adresy. DHCPOFFER obsahuje počiatočné informácie o konfigurácii zariadenia –IP adresa, maska podsiete a predvolená brána nepatrí k základným údajom. Ďalšie voliteľné argumenty správy sú napríklad doba platnosti IP adresy, čas obnovy a iné. DHCP server pošle DHCPOFFER všetkým zariadeniam na lokálnej sieti, nakoľko zariadenie nemá zatiaľ priradenú IP adresu, ale pre identifikáciu priloží hardwarovú adresu zariadenia.

- DHCPREQUEST: Potom, čo koncové zariadenie dostane DHCPOFFER, odpovie správou DHCPREQUEST, čo naznačuje, že chce prijať parametre zo správy DHCPOFFER a zariadenie začne využívať dočasnú adresu. Klient môže dostať viac DHCPOFFER správ. Klient si vyberie jednu z DHCPOFFER správ a reaguje iba na ňu. DHCPREQUEST je vysielaný pre všetky zariadenia v sieti. Aj napriek tomu, že koncové zariadenie dostalo IP adresu, odošle správu DHCPREQUEST z IP adresy 0.0.0.0 .
- DHCPACK: Potom ako server dostane DHCPREQUEST správu, potvrdí predošlú konfiguráciu DHCPACK správou, čím je dokončená inicializácia. DHCPACK správa má zdrojovú adresu IP serveru, a cieľová adresa je opäť vysielaná všetkým zariadeniam a obsahuje všetky parametre, ktoré klient požadoval v DHCPREQUEST správe. Potom, čo klient dostane DHCPACK správu, použije IP adresu ktorá mu bola pridelená. Medzitým, DHCP server uloží záznam týkajúci sa použitia IP adresy a informácie o zariadení (fyzická adresa).
- DHCPNAK: Ak server nie je schopný vyhovieť správe DHCPREQUEST odpovie správou DHCPNAK. Keď zariadenie dostane správu DHCPNAK, alebo nedostane odpoveď na správu DHCPREQUEST, zariadenie znovu začne žiadať o pripojenie správou DHCPREQUEST. Ak zariadenie neuspeje najmenej štyrikrát, tak počas nasledovných šesťdesiatich sekúnd sa o to nebude pokúšať.
- DHCPDECLINE: Ak zariadenie dostane DHCPACK a bude vykonať záverečnú kontrolu parametrov. Zariadenie zašle ARP správu (Address Resolution Protocol) s IP adresou uvedenou v DHCPACK. Pokiaľ zariadenie dostane odpoveď na ARP, je zrejme že adresa sa už používa a zariadenie je povinné poslať správu DHCPDECLINE na server a reštartujte proces konfigurácie.
- DHCPINFORM: Ak klient zariadenie má nakonfigurovanú IP adresu s inak ako pomocou dynamického pripojenia (najčastejšie ručne) je možné pomocou správy DHCPINFORM ďalšie miestne konfiguračné parametre no bez zmeny IP adresy zariadenia.
- DHCPRELEASE: Pokiaľ zariadenie s pridelenou IP adresou sa rozhodne túto adresu už naďalej nepoužívať, je možné zaslať na server správu DHCPRELEASE, čím server uvoľní danú IP adresu pre ďalšie použitie. [1], [3]

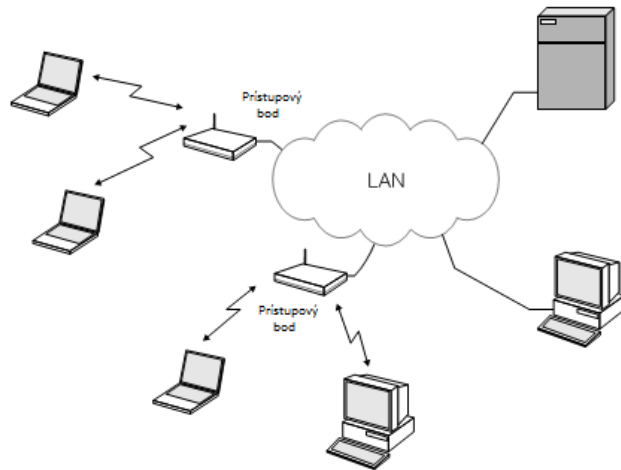
2.3 Bezdrôtové siete

V dnešnej dobe je bezdrôtová komunikácia veľmi rozšírená a je súčasťou nášho každodenného života. Jasným dôkazom je používanie GSM sietí pre naše mobilné telefóny. Aj napriek tomu že bezdrôtové siete boli vyvinuté pre prenos analógového signálu pre Bellov telefón v dnešných dňoch je možné spoľahlivo pomocou bezdrôtových sietí prenášať aj digitálne informácie. Pre dosiahnutie väčšieho objemu prenášaných dát je možné použiť siete označované ako WLAN (Wireless Local Area Network), bezdrôtová lokálna sieť, alebo WPAN (Wireless Personal Area Networks), prípadne FBWA (Fixed Broadband Wireless Access). Jednotlivé technológie sa líšia vzájomne od seba rýchlosťou prenosu dát a dosahom, čo jasne zobrazuje Obr.2.6. [1]



Obr. 2.6: Graf závislosti dosahu a rýchlosti pripojenia na použitej technológii [1]

Technológia lokálnych bezdrôtových sietí sietí WLAN: jedná sa o siete ktorých hlavnou úlohou je zabezpečiť prístup k službám severu (najčastejšie prístup k internetu) no pomocou bezdrôtového spojenia. Prístup k bezdrôtovej sieti sa opiera o pevné pripojenie na základnú stanicu z ktorej zariadenie potrebuje získať prístup. Dosah základných staníc je limitovaný technológiou, a pre pokrytie väčšieho územia je potrebné použiť viacero základných staníc. Na obrázku Obr.2.7 je možno vidieť pripojenie realizované pomocou dvoch základných staníc pripojených bezdrôtovo k serveru. [1]



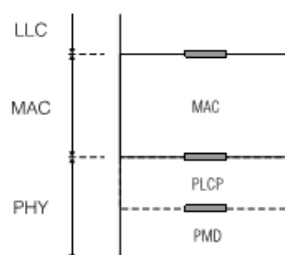
Obr. 2.7: Infraštruktúra bezdrôtovej siete [1]

V dnešných dňoch je táto schéma pripojenia známa ako WiFi. Toto rozhranie je veľmi rozšírené a populárne. Tento typ prenosu dát spadá do odporúčenia IEEE 802.11b, a to najmä vďaka protokolu WECA (Wireless Ethernet Compatibility Alliance), ktorý má na starosti iniciatívu pracovnej skupiny. [1]

Budovanie sieťovej topológie je náročný proces a preto je možné pre jednoduchšie aplikácie a menší dosah využívať iný druh sietí. Tieto siete nevyužívajú žiadnu sieťovú topológiu a označujú sa ad-hoc siete prípadne sa často označujú ako WPAN (Wireless Personal Area Networks). V tomto režime môžu pracovať aj siete WLAN ale väčšinou sa jedná o siete typu Bluetooth. Ich výhodou je že topológia siete je vytvorená dynamicky. Na obrázku je možno vidieť použitie siete WPAN s tromi zariadeniami. [1]

2.4 Štandard IEEE 802.11

Štandard IEEE 802.11 sa vyvinul pre prenos pásma 2.400-2.4835 GHz a infračerveného svetla. Prvá verzia štandardu vyšla v roku 1997. Toto pásmo pracuje v oblasti, kde je časté úzkopásmové rušenie (na tejto frekvencii pracujú aj mikrovlnné rúry). Štandard definuje prenos dát najmä na nižších úrovniach ako je fyzická a linková vrstva, čo je zobrazené aj na Obr.2.8. [1]

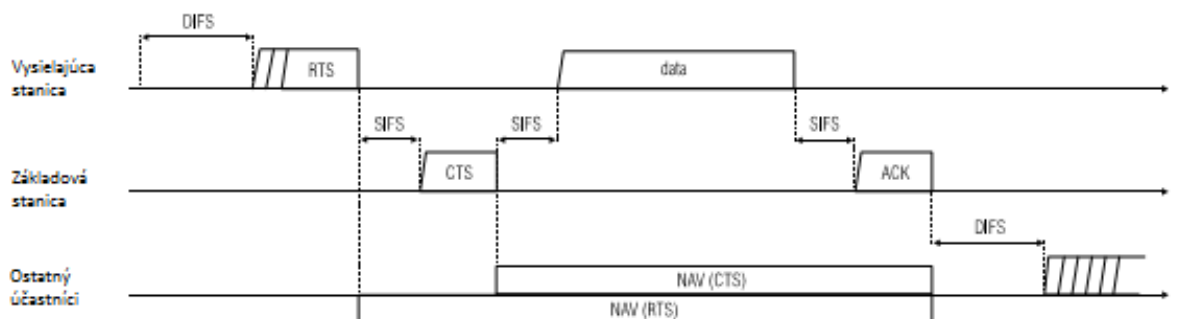


Obr. 2.8: Rozdiel štruktúry ISO/IEEE 802.11 modelu [1]

Fyzická vrstva je delená na vrstvu PLCP (Physical Layer Convergence Protocol), ktorá je závislá na použitom prenosovom médiu a vrstvu PMD (Physical Media Dependent), ktorá je závislá na prenášaných dátach. Linková vrstva zahrňuje prístupovú metódu MAC (Medium Access Control). Nakoľko pri bezdrôtovom prenose je väčšia pravdepodobnosť chyby bol vyvinutý ďalší protokol LLC (Logical Link Control), ktorý dopĺňa funkciu MAC. Nakoľko táto technológia je vysielaná vzduchom je potrebné prenášané dáta príslušne kódovať. Štandard definuje aj bezpečnosť a kódovanie príslušných vrstiev. [1]

Riadenie prístupu– MAC:

Vo všetkých bezdrôtových sieťach je potrebné zaistiť bezkolízny prístup k prenosovému médiu. Jedná sa o mechanizmus ktorý musí zaistiť, že v jednom okamihu vysielajú iba jedna stanica týmto mechanizmom je MAC (Medium Access Control). Najčastejším problémom kolízií dochádza v okamihu ak zariadenie, ktoré sa chystá vysielat' dáta nemá úplné informácie o prevádzke linky. Metóda riadenia prístupu sa zakladá na požiadavke na pridelenie kanálu, ktorý stanica vyšle základovej stanici, a táto požiadavka musí byť schválená pred vysielaním. Ak nastane kolízia a dve a viacej staníc vysielajú dáta zaraz je veľmi pravdepodobné, že žiadne z dát, ktoré boli vyslané nebudú správne, nakoľko počas prenosu nastane interferencia signálov. Výsledkom je, že základná stanica neodpovie a po uplynutí času je potrebné pokúsiť sa poslať dané dáta znovu. [1]

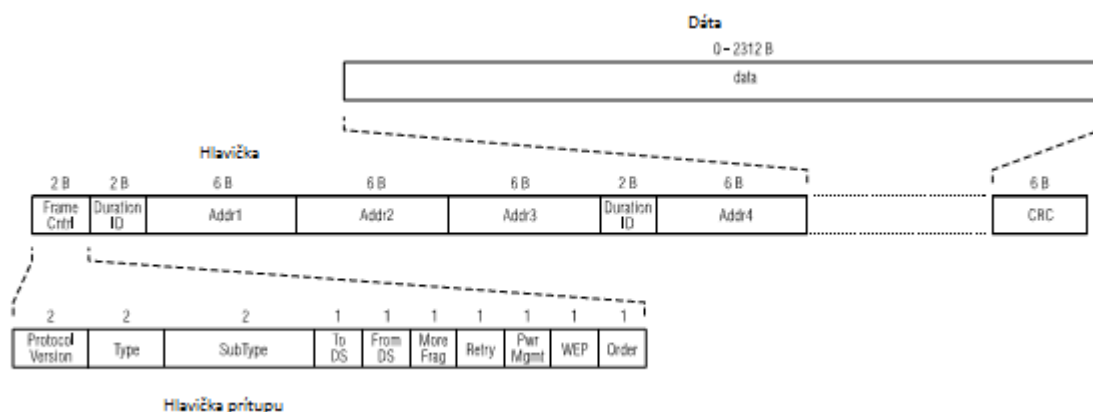


Obr. 2.9: Prístupová metóda IEEE 802.11 [1]

Stanica, ktorá sa chystá vysielat' očakáva voľný prenosový kanál a vyšle príkaz RTS (Request To Send), tento príkaz obsahuje aj dobu rezervácie kanálu. Následne základová stanica potvrdí prenájom kanálu príkazom CTS (Clear To Send), v ktorom pošle čas prenájmu stanice. Následne sú všetci účastníci informovaní o obsadenosti kanálu príkazom NAV (Network Allocation Vector). Ako je možno vidieť na obrázku, vysielanie jednotlivých dát nie je súvislé a nastávajú tu prestávky pri komunikácii. SIFS (Short InterFrame Space) dáva možnosť stanici potvrdiť prijatý rámec bez žiadosti o zmenu kanálu. DIFS (DCF InterFrame Space) doluje spoluprácu staníc bez základovej stanice pri distribuovanom riadení prístupu DCF (Distributed Coordination Function). Rámec tejto prístupovej metódy je zobrazený na Obr.2.9. [1]

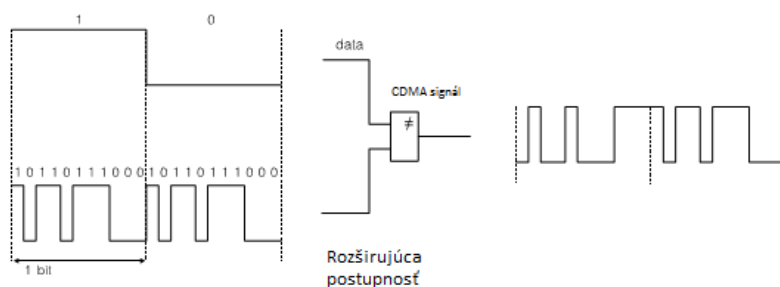
Rámce MAC a ich formáty:

Rámec MAC pre bezdrôtovú sieť je nevyhnutne zložitejší než rámec inej lokálnej siete. Medzi najväčšie rozdiely patria štyri adresné polia, definujúce priamu komunikáciu medzi koncovými stanicami, komunikáciu sprostredkovanú základnou stanicou a komunikáciu sprostredkovanú dvojicou základných staníc v distribučnej sieti. Tieto adresné polia sú definované bitmi ToDS a FromDS v hlavičke rámca. Štruktúra rámcov je vždy rovnaká až na dátovú časť, občas si stanice vymieňajú nielen IP pakety ale aj riadiace dáta ako napríklad a rámce pre správu podporujúce mechanizmy autorizácie, kódovanie dát, časové synchronizácie a iné. V poli Frame Control sú uložené informácie o riadení výkonu stanice a fragmentácii dát. Príklad MAC rámca je možno vidieť na Obr.2.10. [1]



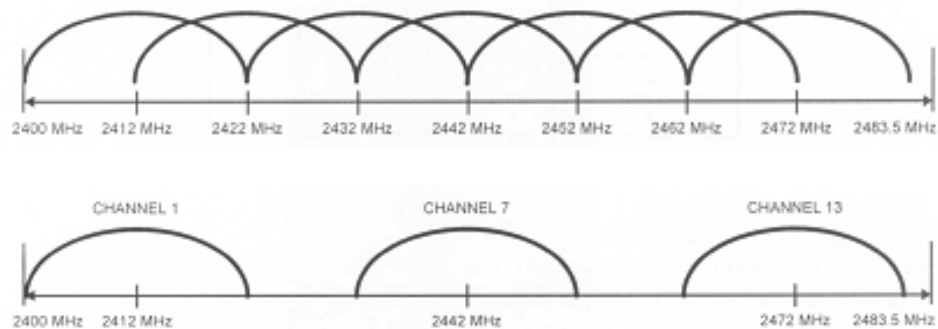
Obr. 2.10: Rámec formátu MAC [1]

DSSS – Táto metóda sa zakladá na sekvencii prenášaných dát, pričom jednotlivé bity sú vzájomne inverzné. Tieto dáta majú za úlohu rozšíriť frekvenčný rozsah prenášaného signálu. Rozloženie tejto postupnosti je pseudonáhodné, bez znalosti postupnosti signál pripomína šum, ale po modulácii signálu výsledný signál pokrýva široké pásmo kmitočtov. Pri dostatočne veľkých rozširujúcich postupnostiach je možný prenos viacerých kanáloch na jednom frekvenčnom pásme. Z praxe však vyplýva že prenosová rýchlosť bude obmedzovaná dĺžkou tejto sekvencie. Základná sekvencia a jej nasledovné spracovanie je zobrazené na Obr.2.11. [1]



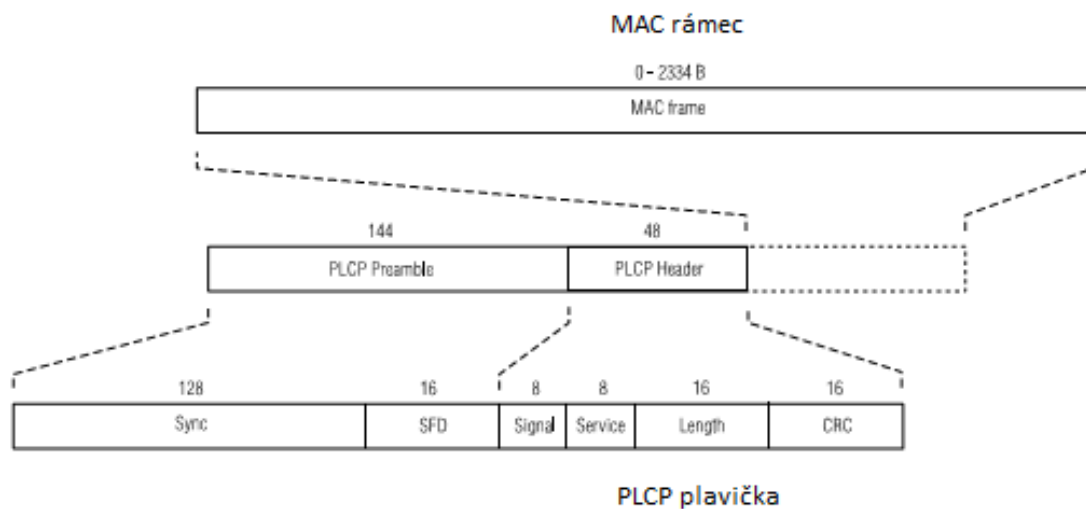
Obr. 2.11: Spôsob rozšírenia vysielanej frekvencie [1]

Pásmo 2,4 GHz poskytuje sedem čiastočne prekrývajúcich sa kanálov, alebo tri oddelené kanály, ako je možno vidieť aj na Obr.2.12.



Obr. 2.12: Jednotlivé kanáli v pásme 2.4GHz [1]

Pomocou rozširovacieho kódu nie je možné viacnásobne využiť kanál, ale je jednoducho možné na strane prijímača určiť začiatky jednotlivých bitov a pokúsiť sa opraviť poškodené dáta. Preambula je kódovaná pomocou „scramblu“ (funkciu si vysvetlíme neskôr), a skladá sa so samých jednotiek. Za preambulou nasleduje SFD (Start Frame Delimiter), čo je šestnásťbitová postupnosť, bitovou synchronizáciou a začiatkom rámca PLCP. Štruktúra hlavičky je odlišná: Signál určuje prenosovú rýchlosť (1Mb/s alebo 2Mb/s). Štandard definuje súlad so štandardom IEEE 802.11 a pole Length definuje dĺžku prenášaných dát. Spôsob výslednej modulácie závisí na prenášanej rýchlosti. Pre prenosovú rýchlosť 1 Mb/s je používaná dvojstavová fázová modulácia DBPSK, pre 2 Mb/s využíva štvorstavovú fázovú moduláciu DQPSK. Pri prenose hlavičky je použitá vždy nižšia prenosová rýchlosť 1 Mb/s. Rámec PLCP DSSS je zobrazený na Obr.2.13. [1]



Obr. 2.13: Štruktúra rámca PLCP DSSS [1]

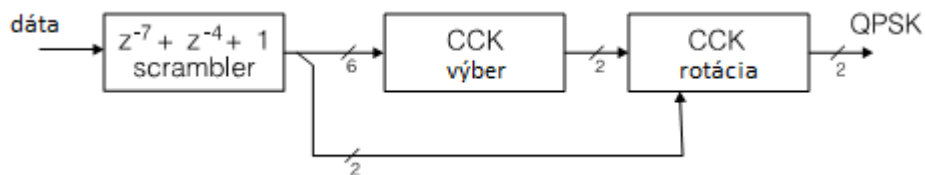
2.5 Štandard IEEE 802.11b

Táto technológia vychádza z princípu kódovo rozprestretého pásma DSS. Pôvodný štandard definuje prenosovú rýchlosť 1Mb/s a 2Mb/s. Táto prenosová rýchlosť, s ohľadom na prenosovú rýchlosť bežnej lokálnej siete, bola nedostatočná. Pri prenose o rýchlosti 1Mb/s je rozprestieracia sekvencia tvorená postupnosťou 10110111000 (Barkerov kód), táto sekvencia je generovaná rýchlosťou 11MHz, a následne sčítaná s dátovým signálom. Tento signál je použitý ako modulačný signál pre fázovú moduláciu BPSK (Binary Phase Shift Keying). Pri zvýšení prenosovej rýchlosti na 2Mb/s je rozprestieracia sekvencia generovaná frekvenciou 22MHz. Po sčítaní s dátovým signálom je tento signál zdroj pre štvorstavový fázový modulátor QPSK (Quaternary Phase Shift Keying). Možnosti modulácie sú zobrazené na Obr.2.14. [1]



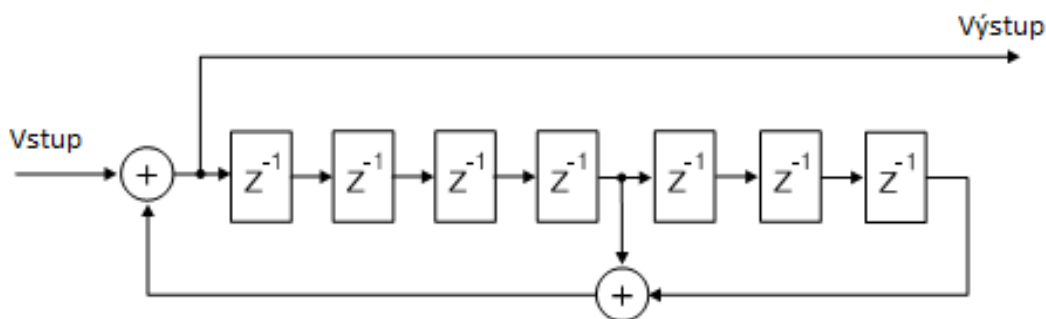
Obr. 2.14: Dvojstavová (BPSK) a štvorstavová (QPSK) modulácia [1]

Pri použití jedného rozprestieracieho kódu je možné dosiahnuť vysokej citlivosti, ale je zrejmé že na jednej frekvencii môže vysielat' iba jedno zariadenie, na druhej strane výhodou bola jednoduchosť prijímaču. Pre zvýšenie prenosovej rýchlosti bola rozprestieracia sekvencia nahradená efektívnejším mechanizmom CCK (Complementary Code Keying). Tento mechanizmus vyvinuli Lucent Technologies a Harris Semiconductor. Osembitová informácia je rozdelená na dva a šesť bitov. Šesť bitov si následne vyberie náhodnú zo šesťdesiatich štyroch sekvencií o dĺžke ôsmich štvorhodnotových symbolov, a následne sú sčítané. Ostávajúce dva bity si vyberajú jednu zo štyroch pootočených základných sekvencií, následne sú tieto signály sčítané a výstupný signál je použitý pre riadenie modulátoru QPSK. Základný princíp je načrtnutý na Obr.2.15. [1]



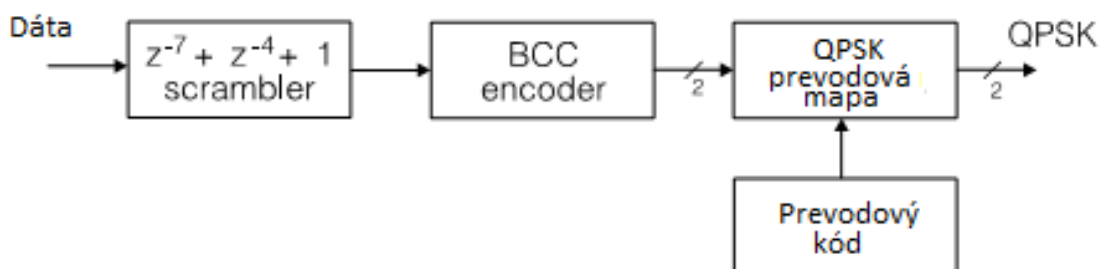
Obr. 2.15: Štruktúra modulátoru CCK [1]

Tento mechanizmus dosahuje výborné výsledky preto, že prenášaný signál je blízky náhodnej postupnosti. Pre dosiahnutie signálu pripomínajúceho náhodný signál je použitie scrambleru. Jedná sa o jednoduché zariadenie, ktorého úlohou je danú správu zakódovať do takej podoby aby tento signál bolo následne možné dekódovať a aby vyhovoval požiadavkám náhodného signálu. Schéma scrambleru je načrtnutá na Obr.2.16. [1]



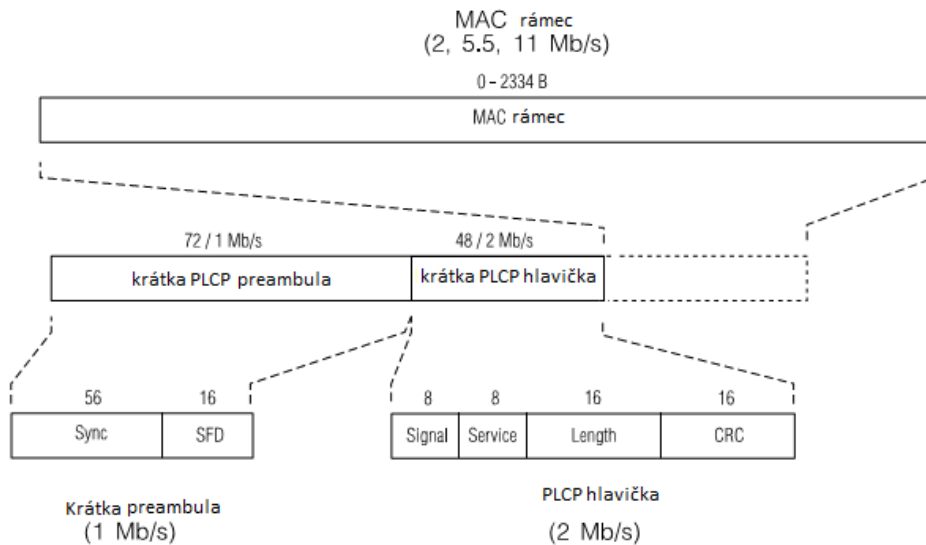
Obr. 2.16: Scrambler $z^{-7} + z^{-4} + 1$ [1]

Ďalšou cestou ako dosiahnuť vyšších prenosových rýchlostí je PBCC (Packet Binary Convolutional Coding). Nevýhodou je zložitejší dekodér ale jedná sa o spoľahlivejší prenos dát. Signál zakódovaný scramblerom vedie do kóderu BCC, z ktorého výstup je dvojbytový signál pre riadenie QPSK modulátoru. QPSK modulátor je však súčasne riadený pseudonáhodným signálom, ktorý posúva fázu modulačného signálu. Kompletná bloková schéma PBCC modulátoru je zobrazená na Obr.2.17. [1]



Obr. 2.17: Štruktúra modulátoru PBCC [1]

Dáta putujú cez skrambler do BCC kóderu ktorý generuje dvojbitový výstupný signál pre riadenie QPSK modulátoru. Týmto spôsobom je možné dosiahnuť prenosových rýchlostí 11Mb/s, no v najnovších aplikáciách prenosová rýchlosť dosahuje až 22Mb/s, pri použití BPSK modulácie je možné dosiahnuť prenosovú rýchlosť až 5.5Mb/s. Nakoľko prenosová rýchlosť narástla rapídne, bolo potrebné upraviť štruktúru vysielaných rámcov. Jednalo sa o úpravy za účelom zrýchlenia komunikácie. Štandardná synchronizačná tabuľka o dĺžke 144 bitov sa skrátila na 72 no stále sa vysiela prenosovou rýchlosťou 1Mb/s. Nasledovná 48 bitová hlavička rámcov sa vysiela na prenosovej rýchlosti 2Mb/s a vlastné dáta sú vysielané prenosovou rýchlosťou aká je definovaná v poli Signál (teda 5.5, 11, alebo 22Mb/s). Príslušná hlavička je zobrazená na Obr.2.18. [1]



Obr. 2.18: Štruktúra rámcu IEEE 802.11b [1]

3 PRAKTICKÁ ČASŤ

Táto kapitola obsahuje moje návrhy hardwarového aj softwarového riešenia. Okrem iného obsahuje aj detailnú komunikáciu samotného modulu, čím ozrejmuje prípadne vysvetľuje funkciu samotného modulu.

3.1 Použité súčiastky

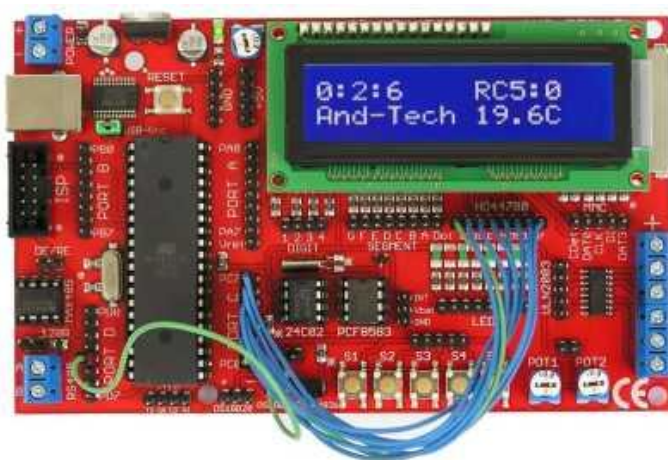
Pri hardwarovom riešení bolo potrebné zaručiť kompatibilitu medzi jednotlivými komponentmi. Na základe požiadavkou z firmy Honeywell návrh všetkých komponentov bol viazaný na WiFi modul RN-171 a vývojovú dosku EvB 5.1.

3.1.1 WiFi modul: RN-171-XV 802.11 b

Pre svoju aplikáciu používam už existujúci modul WiFly. Tento modul som si nevybral náhodou, ale bol zadaný firmou. Jadro modulu tvorí protokol Roving Networks' robust RN-171 pre WiFi moduly. RN - XV. Vstavaná architektúra modulu RN - XV od firmy Pramencov Networks umožňuje pripojenie Wi - Fi pomocou 802.11 b/g. Modul RN - 171 WiFly obsahuje 802.11 b/g vysielač, 32 bitový SPARC procesor, TCP/IP zásobník, hodiny reálneho času, šifrovací akcelerátor, riadenie spotreby a analógové snímacie rozhranie. Modul ponúka aj ďalšie funkcie prostredníctvom programovateľných rozhraní GPIO a ADC. ADC vstup je vlastne analógový vstup samotného modulu a poskytuje štrnásť-bitové rozlíšenie vstupného napätia. Vstupy GPIO sú digitálne vstupy prípadne výstupy, ktoré môžeme konfigurovať. Ďalšou výhodou je veľmi nízky príkon modulu. V režime spánku je pre jeho napájanie potrebný prúd len 4uA. Pri prenose dát modul odoberá napájací prúd 180mA pri vyžarovacej intenzite 10dBm. Modul je schopný nastavovať vyžiarený výkon antény v rozsahu od 0dBm do 12dBm. Modul s nainštalovaným softwarom zjednoduší integráciu a minimalizovať dobu vývoja aplikácie. V najjednoduchšej konfigurácii hardvér vyžaduje iba štyri vodiče (PWR, TX, RX a GND) pre vytvorenie bezdrôtového dátového pripojenia. Pre komunikáciu s mikrokontrolerom modul využíva komunikačné rozhranie UART. Tento modul podporuje väčšinu štandardných prenosových rýchlostí komunikačného rozhrania UART (2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600) no pokiaľ si túto hodnotu nezmeníme je implicitne nastavená na 9600Bd. Daný modul je schopný využívať už definované sieťové štandardy IEEE802.11, WPA a WPA2-PSK. Jedná sa o šifrovacie štandardy a pokiaľ sú v module povolené tak je samotný bezdrôtový prenos automaticky šifrovaný [4]

3.1.2 Vývojová doska EvB 5.1

Konštrukcia celej dosky umožňuje veľmi flexibilné zapojenie a konštrukciu projektov. Všetky periférie sú súčasťou dosky a je len na užívateľovi či ich pre návrh vlastného riešenia využije alebo nie. Ich vstupy alebo výstupy sú vyvedené na samostatné vstupno-výstupné brány, rovnako ako nepoužité brány mikroprocesoru, takže je možné dané periférie pripojiť na ľubovoľné vývody procesoru a ďalej s nimi pracovať. Mikrokontrolér má pevne pripojený iba kryštálový oscilátor 16MHz, tlačidlo RESET, Pull-Up rezistory pre I2C zbernicu (hodnota 10k Ω) a napájanie. Ostatné (voľné) porty sú voľne dostupné na prepojovacích vývodoch. Daná vývojová doska je vyobrazená na Obr.3.1. [5], [6]



Obr. 3.1: Vývojová doska EvB 5.1 [6]

Vlastnosti a osadenie:

- procesor AVR ATmega32 kryštál 16MHz
- obvod reálneho času PCF8583
- pamäť EEPROM AT24C02
- infračervený prijímač TSOP4836
- teplotný senzor DS18B20
- prevodník zbernic RS485/RS232 - SN75176BP
- päťica pre kartu MMC/SD
- 5 tlačidiel
- 8 indikačných LED diód
- 2 potenciometre pre nastavenie napätia
- 4 x sedem segmentový LED zobrazovač
- 5 x výkonový výstup s otvoreným kolektorom ULN2003
- podsvietený displej LCD 2x16 znakov
- USB konektor
- ISP programovací konektor

Periférie a ich zapojenie:

Jadro vývojovej dosky tvorí mikrokontroler ATmega 32. Mikrokontrolér je napevno taktovaný hodinovým kryštálom 16MHz, s možnosťou využitia interného RC oscilátoru 8MHz. Použitie zbernice I2C: v časti mikrokontroleru sú k signálom SDA a SCL pripojené Pull-Up rezistormi, nie je potrebné pripojovať externé rezistory, preto je možné pre pripojenie externých obvodov cez I2C použiť vývody SDA a SCL na portu C tak, ako odporúča výrobca. [5], [6]

Pre meranie teploty je možnosť použiť senzor od Maxim-Dallas s digitálnym výstupom – DS18B20, ktorý má svoj digitálny výstup na spoločnom konektore s výstupom od prijímača infračerveného signálu TSOP4836, pracujúcom na štandarde IR vysielania 36 kHz. [5], [6]

Pre interakciu s okolím je vývojová doska vybavený piatimi tlačidlami a piatimi výkonovými výstupy. Tlačidlá sa zapojujú priamo na vstupy mikroprocesoru, v ktorom je potrebné nastaviť vstupný port s Pull-Up rezistormi a tlačidlo, tak je možné priamo pripojiť medzi vstupnú bránu a tlačidlo, ktoré spína nízku napäťovú úroveň na vstup mikrokontroleru. Preto sú všetky tlačidlá automaticky spojené so zemou. Výkonové výstupy sú realizované cez skrutkovaciu svorkovnicu obvodom ULN2003, pri ktorom sa využíva päť kanálov. Šiesta svorka je spojená priamo s kladnou napájacou svorkou. [5], [6]

Pre zobrazenie výstupov môžeme využiť tri typy optických indikátorov. Je k dispozícii osem zeleno/červených samostatných LED diód. Tieto diódy sú zapojené cez rezistor na päťvoltovú úroveň a preto je pre rozsvietenie pripojiť diódu na zem napájania. Ďalšia možnosť je využiť LCD display s možnosťou riadenia jasou. Poslednou možnosťou zobrazenia údajov je sedem segmentový displej so 4 pozíciami. Užívateľ si môže sám vybrať či použije sedem segmentový alebo LCD display ale nemôže použiť oba naraz. [5], [6]

Pre pripojenie vývojovej dosky k ďalším perifériám pomocou dátových zberníc, alebo obvodov zaisťujúcich komunikáciu, obsahuje doska rozhranie RS485, realizované budičom SN75176BP a rozhranie USB so známym čipom FT232RL, ktorý sa chová ako virtuálny sériový port. Pomocou tohto rozhrania je možné využívať sériovú linku RS232 a pri použití bootloderu je možné aj programovať procesor. Rozhranie RS485 je na doske vybavené voliteľným rezistorom 120Ω ako zakončenie zbernice a voliteľné je možné priamo pripojiť k mikrokontroléru signál DE/RE pre riadenie toku dát (port D, signál PD2). [5], [6]

3.1.3 Mikrokontroler ATmega 32:

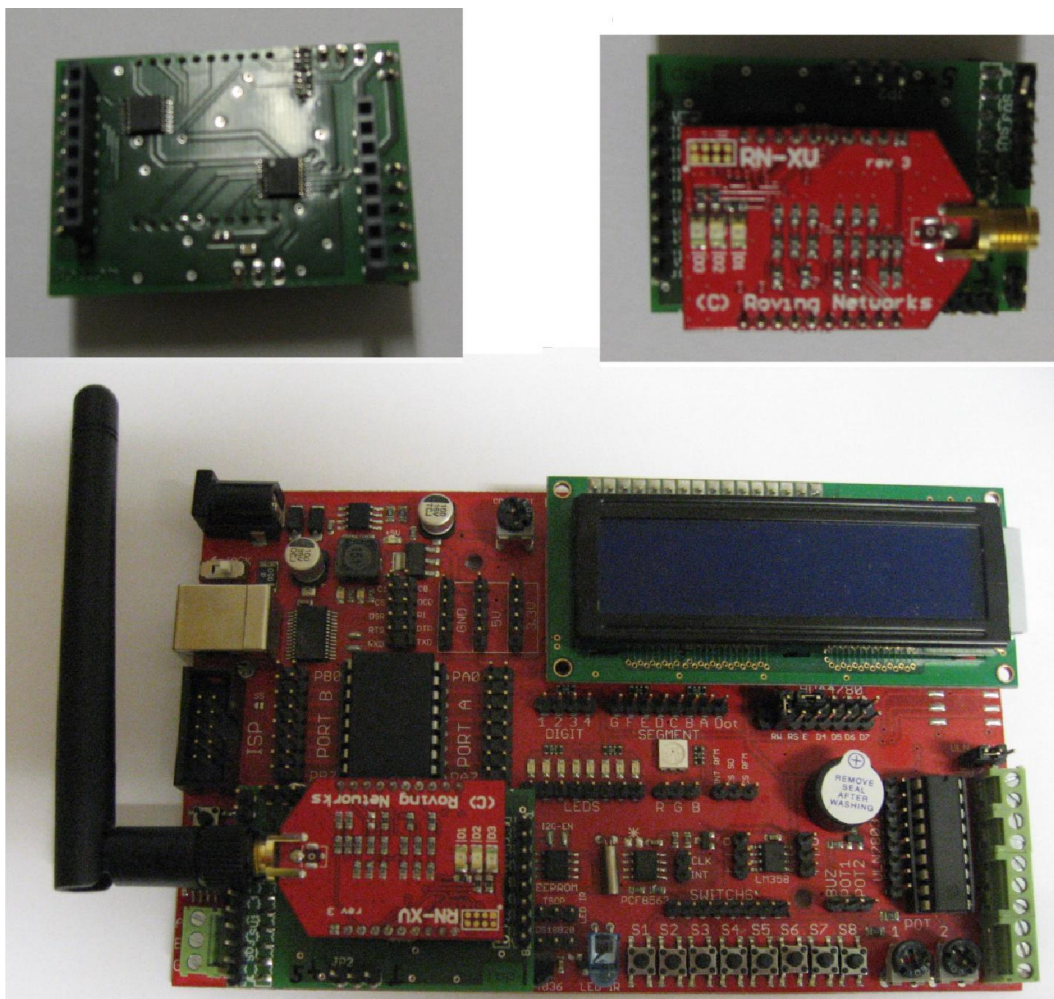
Jedná sa o 8-bitový mikrokontrolér s architektúrou RISC. Kontrolér obsahuje 131 inštrukcií pričom je možné väčšinu z nich vykonávať v jednom hodinovom cykle. Pre obecné použitie kontrolér obsahuje 32×8-Bit univerzálnych registrov, ECI pamäte programu, 1024 bytov dát v pamäti EEPROM, ECI pamäť 2KB, interné domény z LAN, ECI SRAM. Kontrolér obsahuje zbernicu ISP s možnosťou bootovania. Kontrolér obsahuje moduly pre komunikáciu UART, master/slave SPI. Modul obsahuje 32 vodičov ktoré je možné nastaviť ako vstupné alebo výstupné porty. Modul obsahuje 8 desaťbitových ADC prevodníkov, 4 PWN moduly, 3x8 bitový časovač a jeden 16 bitový časovač. Pre analógový signál je na čipe integrovaný analógový komparátor. Ako zdroj hodinového signálu je možné použiť interný kalibrovaný kryštál, interný RC generátor, externý kryštál až do frekvencie 16 MHz, externý RC oscilátor. Kontrolér sa môže nachádzať v 6 módoch počas jeho behu: Idle (aktívny), ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby. Napájanie kontroléru je v rozsahu 2,7 až 5,5V. prúdová spotreba je závislá na móde v ktorom sa nachádza. V aktívnom móde modul je potreba napájať 0,6mA, kľudovom móde 0,2mA. Pre ochranu duševného vlastníctva je možnosť uzamknúť software v kontroléry. [7]

3.1.4 Prevodník napät'ových úrovní TXB0106

Nakoľko štandardné napájanie mikrokontroleru je 5V, a maximálne napájanie WiFly modulu je 3.3V je potrebné zabezpečiť bezpečné prevedenie napät'ových úrovní medzi jednotlivými vstupmi a výstupmi. Nakoľko je užívateľsky možné jednotlivé porty nastavovať aj ako vstupné a aj ako výstupné, smer prenosu jednotlivých dát je vopred neurčiteľný. Väčšina bežne predávaných napät'ových prevodníkov využíva pre riadenie smeru prenosu dát vývod označovaný ako vývod smeru toku dát. Nakoľko by tento vývod bolo potrebné riadiť jednou vstupno-výstupnou bránou z WiFly je tento koncept neefektívny. Preto je pre moje účely vhodný prevodník napät'ových úrovní TXB0106, lebo interne obsahuje dva antyparalelné buffre ktoré tú napájané rozličným napätím. V prípade že zo strany A príde vysoká napät'ová úroveň prevodník nastaví stranu B na vysokú napät'ovú úroveň taktiež. Vetva B je napájaná rozdielnym napätím a preto je jeho napät'ová úroveň odlišná. Podobný proces nastane aj v opačnom prípade, ak príde ku zmene napätia na strane B. Samotný modul je navrhnutý tak, aby ak by prišlo ku kolízii (obe strany prevodníku sú pripojené na vstupné porty, prípadne obe strany prevodníku sú pripojené na výstupné porty určitej logickej úrovni). Nepríde síce ku spoľahlivému prenosu dát, nakoľko to ani nie je v týchto prípadoch možné, ale nepríde ani ku znehodnoteniu či zničeniu samotného prevodníku. Modul dokáže prekladať napät'ové úrovne v rozsahu 1,25-3.6V pre port A a 1,65-5,5V pre port B. Prevodník je taktiež vybavený vstupom OE, ktorý sa používa pre nastavenie všetkých portov do stavu vysokej impedancie. Táto funkcia sa používa najmä ak je potrebné zaučiť odpojenie vstupov a výstupov po pripojení napätia. Rýchlosť prevodu napät'ovej úrovni je závislá na vstupnom a výstupnom napätí, ale pre túto aplikáciu sa jedná o prevod z 3,3V na 5V je doba prevodu je maximálne 4ns v oboch smeroch. [8]

3.2 Návrh hardwarového konceptu pre modul WiFly

Hardwarový koncept bol viazaný na vývojovú dosku EvB 5.1. Toto riešenie bolo zadané z firmy a prinieslo viacero komplikácií. V prvom rade bolo potrebné všetky vstupno-výstupné porty pripojiť na jednotlivé porty procesoru so zaručeným prevodom napät'ových úrovní. K tom slúži prevodník úrovní TXB0106. Ďalším obmedzením bolo samotné rozloženie súčiastok a vzhľad finálnej dosky. Limitácie vznikali najmä z rozloženia portov na samotnej vývojovej doske. Nakoľko je pre firmu nepodstatná schéma zapojenia, nebudem v mojej diplomovej práci uvádzať vzhľad samotnej dosky plošného spoja (Obr. 3.2).



Obr. 3.2: Vzhľad finálnej dosky

Pre jednotlivé senzorké vstupy je potrebné použiť delič. Je použitá sieť rezistorov hodnoty $10\text{K}\Omega$. Tieto odpory sú zapojené ako delič vstupného napätia na polovicu. Ďalšou pasívnou súčiastkou je rezistor $10\text{K}\Omega$ pripojený k vstupu reset. Tento rezistor zabezpečuje neustále vysokú úroveň na tomto vstupe a nakoľko je tento vstup negovaný, modul nespadne do hardwarového reštartu. Na doske je taktiež kondenzátor o kapacite 100nF . Tento kondenzátor slúži ako filter napájacieho napätia.

3.3 Popis softwarového konceptu

Pre overenie možností modulu som sa rozhodol vytvoriť aplikáciu pre monitorovanie a nastavovanie jednotlivých periférií samotného modulu. Z možností samotného modulu je zrejmé, že je veľmi výhodné využívať už vytvorené služby, ktoré modul ponúka. Pre moje účely využívam internú funkcionálnosť TCP serveru. Pomocou tohto rozhrania dokážem vysielat' dáta, ktoré modul reprezentuje na dátovom rozhraní UART.

3.3.1 Popis rozhrania UART:

Synchronne/asynchronne sériové rozhranie USART (*Universal Synchronous/Asynchronous Receiver and Transmitter*). Jedná sa o zariadenie pre sériovú komunikáciu, ktoré je možné nastaviť buď na synchronný (napríklad zbernica SPI) alebo asynchronný prenos (napríklad linky RS232 alebo 485). [7], [9],[10]

Asynchronný prenos

UART vysielala dáta obvykle na vývode označeným obvykle ako TX (transmit) a prijíma na vývode označenom RX (receive). Počiatočná úroveň signálu je log. 1 (pokiaľ neprebíha vysielanie ani príjem). Zahájenie vysielania sa vykoná zmenou hodnoty signálu na log. 0 po dobu jedného bitu (tzv. start-bit). Následne sa následne je posiadaný najnižší dátový bit, posledný a nasledujúce dátové bity až po najvýznamnejší dátový bit, po jeho vyslaní nasleduje stop bit, ktorý má opäť úroveň log. 1. Po odvysielaní stop-bitu môže začať prenos ďalšieho bajtu. [7], [9],[10]

Tab 2: prenášané dáta

bit:	1	2	3	4	5	6	7	8	9	10	11
funkcia:	štart bit	5 až 8 dátových bitov								stop bity	
popis:	štart	Dáta 0	Dáta 1	Dáta 2	Dáta 3	Dáta 4	Dáta 5	Dáta 6	Dáta 7	Stop	

Módy USARTu:

USART je možné konfigurovať v nasledovných módoch:

Asynchronný (full duplex) – V tomto módu je nakonfigurovaný ako úplne duplexný asynchronný systém, ktorý môže komunikovať s perifériami ako sú napr. CRT terminály, osobné počítače (PC) atd.. [7], [9],[10]

Synchronný (half duplex) – Druhá možnosť je nakonfigurovať USART ako „poloduplexný“ synchronný systém, ktorý môže komunikovať napr. s perifériami ako sú A/D a D/A prevodníky, sériová EEPROM atd. Tento synchronný mód môžeme nastaviť ako - Master alebo Slave. [7], [9],[10]

Synchronný mód používa hodinovú a dátovú linku, ale v asynchronnom nie je tento hodinový signál použitý. Jeden vývod je použitý pre vysielanie a jeden pre príjem. Obe

operácie môžu prebiehať nezávisle na sebe. A môžu dokonca prebiehať zároveň – preto je možné hovoriť o asynchrónnom móde ako o plne duplexnom. [7], [9],[10]

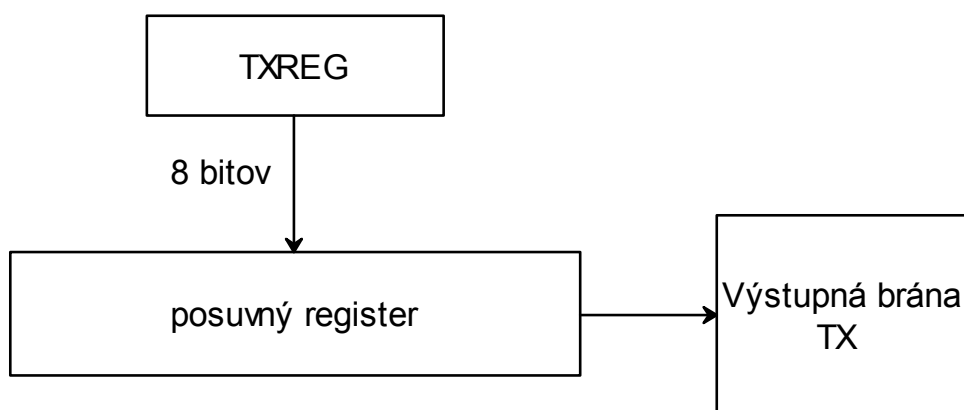
Vysielanie bajtu:

Universal Asynchronous Receiver/Transmitter (UART) je určený pre prenos jednotlivých bitov sekvenčným spôsobom. Na dešifrovanie správy, zariadenie na príjem znova zostaví jednotlivé signály do kompletných bajtov. [7], [9],[10]

UART rozhranie nie je schopné priamo prijímať externé signály použité medzi rôznymi zariadeniami. Samostatné zariadenie musí byť schopné prevodu logických úrovní z UART do externých signalizačných úrovní. Externý signál môže byť z mnohých rôznych podobách, a to podľa štandardov napätia RS - 232 , RS - 422 - 485. Pre špeciálne systémy je možné použiť dokonca aj optické vlákno. [7], [9],[10]

Komunikácia môže byť simplex (iba v jednom smere, bez rezervy na prijímacom zariadení odosielať informácie späť do vysielacieho zariadenia), plne duplexná (oba prístroje posielajú a prijímajú súčasne) alebo poloduplexná (zariadenie sa mení buď na vysielateľ alebo prijímač). [7], [9],[10]

Štart bit signalizuje prijímač, ktorý inicializuje vysielanie. Nasleduje päť až osem bitov, v závislosti na nastavení, pričom obvykle dáta reprezentujú jeden znak. Ak je použitý paritný bit, je s pravidla vysielaný vo všetkých vysielaných bajtoch. Nasledujúci jeden alebo dva bity sú vždy pre ukončenie prenosu (logická '1 '), tzv. stop bit. Tieto signalizujú ukončenie vysielania. Pretože štart bit je logická nula (0) a stop bit je logická jednotka (1) vždy existujú aspoň dve zmeny logickej úrovni o čas vysielania dát. Popis vysielania je zobrazený na Obr.3.3. [7], [9],[10]



Obr. 3.3: Schéma vysielania bajtu [10]

Pre použitie v mikrokontroleri je postup nasledovný:

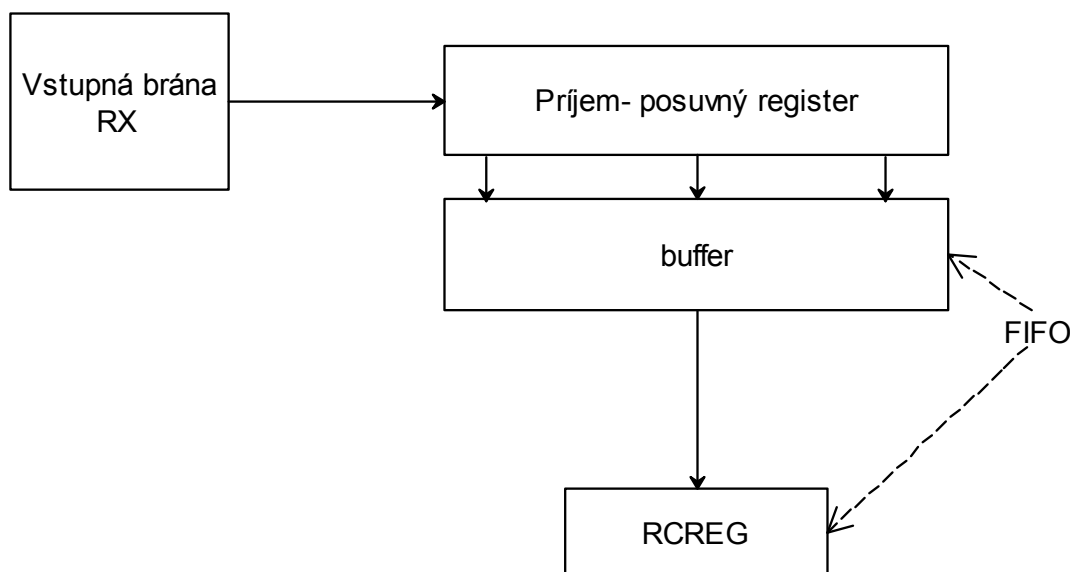
Vysielané dáta sa zapisujú do registra TXREG, odkiaľ sú následne predoslané do registra na vysielanie (Transmit Shift Register). Následne je potrebné zahájiť komunikáciu a predoslať informáciu pomocou vývodov RX, TX a pre zahájenie je použitý štart bit. Pre vysielanie

nasledovných dát je potrebné iba naplniť daný register TXREG, čo výrazne zefektívňuje komunikáciu. [7], [9],[10]

Prijem bajtu:

Pre prijatie dát pomocou zbernice UART je veľmi dôležitá časová synchronizácia. Interný hodinový signál je typicky osem násobok bit rate prenosu. Vstupný signál je kontrolovaný pri každom takte interného hodinového signálu, pričom predpokladá na začiatku prenosu štart bit. Ak štart bit trvá najmenej polovicu hodinových cyklov je vyhodnotený ako platný a následne je zahájený prenos. Ak tomu tak nie je, považuje sa to šum ktorý je ignorovaný. Po uplynutí tejto doby je čítaný nasledujúci bit, pričom načítaný znak je následne uložený do registra. Po požadovanom počte prijatých bitov (5-8 bitov, typicky), je obsah posuvného registra k dispozícii, je nastavený príznak označujúci, prijatie nových dát a môžu tiež nastať požiadavky na prerušenie. [7], [9],[10]

Komunikácia UART nemá v princípe spoločný hodinový signál, na rozdiel od komunikačného signálu. Typicky, synchronizácia je potrebná pre interné hodiny a vykonáva sa na dátových linkách, tento signál nie je považovaný za falošný impulz. Informácia o časovači je potrebná pre prijímacie zariadenie, nakoľko môže kalibrovať nepresnosť hodinového signálu na strane vysielača. Jednoduchšie zariadenia nevykonávajú synchronizáciu pomocou dátových liniek. Miesto toho vyhľadávajú zostupnú hranu štart bitu, na základe tejto informácie vypočítajú čas potrebný pre prijatie jedného bitu a následné dátové bity snímajú v predpokladanom čase (predpokladajú čítaný bit v prostriedku vysielaného bitu). Tento systém je možné použiť ak je zaručená dostatočná presnosť oboch hodinových signálov. Proces prijímania dát sa nachádza na Obr.3.4. [7], [9],[10]



Obr. 3.4: Schéma prijímania bajtu [10]

Pre detekciu štart bitu na vývode RX, je potreba dáta bit po bite presunúť do vstupného registra pre príjem (receive shift register). Po prijatí posledného bitu sa skontroluje stop bit a nahromadené dáta sa presunú do registra bufferu a v prípade prijatého stop bitu aj do RCREG. Oba tieto elementy sa skladajú z pamäte FIFO (first in first out) a spracovávajú sa v poradí v akom boli prijaté. [7], [9],[10]

Modul WiFly má dva režimy: režim prenosu dát a príkazového režimu. V režime prenosu dát, modul môže prijímať prichádzajúce dáta alebo vysielat' dáta. Ak chcete nakonfigurovať parametre samotného modulu, alebo zobraziť aktuálnu konfiguráciu, je nutné vstúpiť do príkazového režimu.

Vstup to príkazového režimu:

V predvolenom nastavení, je modul v režime prenosu dát okamžite po zapnutí. Ak modul prijme sekvenciu znakov „\$\$\$“ spôsobí, že modul vstúpiť do príkazového režimu. Tento reťazec nesmie obsahovať žiadne iné znaky alebo medzery medzi znakmi. Modul odpovie „CMTD“ čím signalizuje, že sa nachádza v príkazovom režime . Potom, čo vstúpime do príkazového režimu, môžeme využívať interné príkazy modulu pre jeho konfiguráciu prípadne zisťovanie súčasnej konfigurácie. Každý príkaz musí končiť ukončením riadku <CR> . Väčšina platných príkazy vráti reťazec „AOK“, tie neplatné vrátia „ERR“. Ak chcete opustiť príkazový režim, je nutné poslať reťazec „exit <CR>“ . Modul odpovie reťazcom „EXIT“ , čo znamená , že má vystúpiť z príkazového režimu a do režimu prenosu dát.

Môžete posilať príkazy modulu cez UART alebo cez vzdialené pomocou telnet pripojenia. Pri použití UART rozhrania, nastavenie komunikácie by malo zodpovedať uloženým nastaveniam WiFly modulu. Predvolené nastavenie prenosu je 9600Bd (bit/s), 8 bitov, bez parity, 1 stop bit a hardvérové riadenie toku zakázané.

3.4 Inicializácia modulu

Pre nadviazanie samotnej komunikácie je potrebné zabezpečiť sieťové prepojenie samotného modulu a PC. Je dôležité, aby bežný užívateľ bol schopný bez problémov nadviazať sieťové prepojenie. Preto som volil koncept vytvorenia prístupového miesta, ktoré vytvára WiFi sieť a obsahuje službu dynamického pridelenia IP adries. Z užívateľského hľadiska je proces pripojenia do správnej siete jednoduchý: vyhladá bezdátové siete v dosahu a následne sa k príslušnej sieti pripojí. V skutočnosti užívateľ pošle požiadavku na priradenie IP adresy v príslušnej sieti, a samotný modul zabezpečí výber a priradenie IP adresy užívateľovi. Tento proces nám zabezpečí užívateľský príjemné pripojenie do siete. Pre dosiahnutie popisovaného chovania je modul potrebné najskôr nakonfigurovať. Prvým krokom pri konfigurácii je overenie prípadne nahranie správnej verzii operačného systému pre samotný modul. Výrobca uvádza, že každý modul obsahuje dve verzie softwaru, ktoré sa líšia vo funkcionalite na ktorú sú určené. V našom prípade modul obsahoval software 2.36 a 2.45 z 14.9.2012. Verzia 2.36 je určená pre pripojenie do existujúcej siete. Pre moje účely som využil verziu softwaru 2.45. Táto verzia softwaru je schopná vytvoriť prístupové miesto

a zabezpečiť automatické priradenie IP adresy. Pre konfiguráciu modulu je potrebné komunikovať s modulom pomocou podporovaných komunikačných rozhraní. Pre moje účely som volil hardwarové riešenie pomocou rozhrania UART. Pre nahranie správnej verzie softwaru som volil nasledovný sled príkazov:

Príkaz:	Popis príkazu:	Odpoveď modulu:
<u>\$\$\$</u>	Vstúpenie do príkazového režimu	CMD
<u>ls<CR></u>	Tento príkaz zobrazí súbory nahrané v internej pamäti modulu	FL#SIZFLAGS 2203WiFly_EZX-2.36 22123wps_app 34213WiFly_EZX-2.45 55110config 200Free,Boot=34,Backup=2
<u>load WiFly_EZX-2.45<CR></u>	Nahranie príslušnej verzii softwaru	AOK
<u>save <CR></u>	Uloženie aktuálnych nastavení	Storinginconfig
<u>Reboot<CR></u>	Riadený reštart modulu. Tento príkaz je potrebný nakoľko až po opätovnom štarte modulu sa prejaví nakonfigurované zmeny.	WiFly Ver 2.45, 09-27-2012 on RN-171 MAC Addr=00:06:66:71:6b:f5 *READY*

Po nahraní správnej verzii softwaru je potrebné nakonfigurovať samotný modul. Pre túto konfiguráciu som volil sled nasledovných príkazov:

Príkaz:	Popis príkazu:
<u>\$\$\$</u>	Vstúpenie do príkazového módu
<u>set wlan channel 11<CR></u>	Voľba kanálu určeného pre vytvorenie komunikácie. Bez konfigurácie je táto hodnota nastavená na 0, čo znamená že je vybraný voľný kanál na základe predvolenej tabuľky
<u>set wlan ssid „MENO“<CR></u>	Nastavenie mena vytvorenej siete
<u>set ip a 192.168.1.1<CR></u>	Nastavenie IP adresy na 192.168.1.1 bez konfigurácie je táto adresa nastavená na 0.0.0.0
<u>set ip net 255.255.255.0<CR></u>	Nastavenie masky podsiete. Bez konfigurácie je maska podsiete rovnaká ako je v príklade nastavená.

set wlan join 7<CR>

Tento parameter nastavuje postup ako má samotný modul postupovať po nadviazaní spojenia s iným zariadením. Je možné voliť z nasledujúcich parametrov:

- 0- Manuálna asociácia siete
- 1- Snaha o vytvorenie spojenia so zariadením typu prístupové miesto, názov siete musí byť zhodný s SSID
- 2- Snaha o vytvorenie spojenia so všetkými zariadením typu prístupové miesto, názov siete nemusí byť zhodný s SSID
- 3- Rezervované
- 4- Vytvorenie prístupového miesta s nastaveným menom, IP adresou a maskou podsiete. Je potrebné vopred nastavenie komunikačného kanálu
- 7- Vytvorenie prístupového miesta s nastaveným menom, IP adresou a maskou podsiete. Tento typ pripojenia je doporučený pre našu verziu softwaru

set ip dhcp 4<CR>

Nastavenie služby DHCP. Opäť je možné vybrať z nasledovných parametrov:

- 0- Použitie statickej IP adresy
- 1- Získavanie IP adresy z prístupového miesta
- 2- Automatické získavanie IP pre hoc mód
- 3- Mód zachytávania IP adres. Modul využije naposledy použitú IP adresu pokiaľ jej platnosť ešte nevypršala
- 4- Vytvorenie DHCP serveru pre prístupové miesto

save <CR>

Uloženie aktuálnych nastavení

Repot<CR>

Riadený reštart modulu. Tento príkaz je potrebný nakoľko až po opätovnom štarte modulu sa prejaví nakonfigurované zmeny.

3.4.1 Zabezpečenie bezdrôtovej komunikácie

Samotný modul je schopný šifrovanie komunikáciu. Pre to nám slúži nasledovný príkaz:

Príkaz:

Set wlan auth X<CR>

Popis príkazu:

Tento príkaz nastavuje spôsob šifrovania komunikácie. Namiesto X je potrebné použiť jednu z uvedených možností:

- 0- Bez šifrovania
- 1- WEP šifrovanie
- 2- WPA1 šifrovanie
- 3- Miešané WPA1 a WPA2-PSK
- 4- WPA2 šifrovanie
- 5- Nepoužitý
- 6- Iba pripojenie do existujúcej siete

WEP šifrovanie

WEP je šifrovací algoritmus postavený na štandarde 802.11. Šifrovanie WEP používa šifru s 40 - alebo 104 - bitovým kľúčom a 24 - bitovým inicializačným vektor. Podľa normy, WEP používa algoritmus RC4. RC4 je symetrický algoritmus, pretože používa rovnaký kľúč pre šifrovanie a dešifrovanie dát. Ak je povolené šifrovanie WEP, obe strany ktoré komunikujú majú kľúč. Keď stanica prijme dáta, ktorý nie sú kódované s príslušným kľúčom, dáta sú vyhlásené za neplatné. WEP je možné použiť predovšetkým pre domáce kancelárie alebo malé kancelárie, ktoré nevyžadujú veľmi silné zabezpečenie. WEP je implementované v hardvéri a preto toto šifrovanie má minimálny vplyv na rýchlosť a objem prenesených dát. Medzi nevýhody WEP patrí to, že samotný kľúč je vysielaný druhému zariadeniu a tieto dáta je možné zachytiť a tým je možné prezradiť šifrovací kľúč. Tento kľúč je pri vysielaní nešifrovaný.[11]

WPA šifrovanie

Šifrovanie WEP môže byť nedostatočné a preto je možné používať aj WPA šifrovanie. Tento štandard poskytuje ochranu proti nabúraniu. Tento systém je založený na aktuálnom overení EAP/802.1x a využíva dynamické správy kľúčov, a tak zabezpečuje lepšie šifrovanie. Potom, čo klient autentifikuje svoje pripojenie k serveru zaeviduje žiadosť na WPA kľúč, následne príde k dohodnutiu šifrovania medzi klientom a serverom. WPA je lepšie zabezpečený a je určený pre zabezpečenia na podnikovej úrovni, WPA poskytuje tiež Pre-Shared Key (Prednastavený šifrovací kľúč) verziu (WPA-PSK), ktorý je určený pre použitie v malej kancelárii, alebo domácej bezdrôtovej sieti.[12]

WPA2 šifrovanie

WPA2 patrí k ďalšej generácii WiFi zabezpečenia. WPA2 patrí ku štandardom IEEE 802.11i. WPA2 implementuje šifrovací algoritmus s využitím režimu CCMP. Šifrovanie spočíva v algoritme ktorý šifruje 128-bitové bloky dát súčasne s 128-bitovým šifrovacím kľúčom. CCMP algoritmus vytvára správu integrity kódu (MIC), ktorá zaisťuje overovanie pôvodu dát a integritu dát pre bezdrôtové rámce. WPA2 ponúka vyššiu úroveň zabezpečenia než WPA, pretože ponúka silnejšie šifrovanie a to je protokol, ktorý vyhodnocuje integritu dočasného šifrovacieho kľúča (Temporal Key Integrity Protocol-TKIP). TKIP je šifrovací algoritmus, ktorý používa WPA. WPA2 vytvára nové kľúče pre každé pripojenie. Šifrovacie kľúče, ktoré sa používajú pre každého klienta v sieti sú jedinečné a špecifické pre daného klienta. Každý balík dát, ktorý ich poslal vzduchom je šifrovaná pomocou unikátneho kľúča. Bezpečnosť s použitím nového a unikátneho šifrovacieho kľúča, pretože nie je možné akýkoľvek šifrovací kľúč opätovne použiť. Väčšina odborníkov tvrdí, že WPA je bezpečný pokiaľ nie je dešifrovaný TKIP, no je doporučený prechod na WPA 2 čo najskôr.[13]

WPA kombinované s WPA2 šifrovaním

WPA bol predstavený vo WiFi zariadeniach v roku 2003. WPA2 bol zavedený po dvoch rokoch a to v roku 2004. Je nutné, aby boli výrobky certifikované so šifrovaním WPA2 kompatibilné s výrobkami, ktoré sú certifikované pre šifrovanie WPA. WPA a WPA2 poskytujú vysokú úroveň zabezpečenia pre koncových užívateľov a správcu siete. Oba štandardy sú určené pre osobné a podnikové siete. Oba štandardy využívajú režim Pre-Shared Key (PSK) pre overovanie. WPA obsahuje podobné riziko ako WEP šifrovanie, preto je určený pre malé kancelárie. WPA používa TKIP (Temporal Key Integrity Protocol-TKIP) algoritmus pre šifrovanie. WPA2 je ďalšia generácia WiFi zabezpečenia. Patrí do štandardu IEEE 802.11i. Národný inštitút pre štandardy a technológie (NIST) odporúča AES šifrovací algoritmus. Medzi hlavné výhody patrí dynamická obsluha jednotlivých užívateľov, pre priradenie šifrovacích kľúčov, odstránenie administratívnej záťaže a bezpečnostných požiadavkou súvisiacich s šifrovaním. S 802.1X protokol používa na overovanie, ako sú prihlasovacie heslá správ, ktoré nikdy nie sú vyslané bez šifrovania cez bezdrôtové médium. Kým typy overovanie 802.1X poskytuje silnú autentizáciu bezdrôtových sietí, je potreba tento štandard rozšíriť o TKIP alebo AES šifrovanie. Ďalšou výhodou overovanie 802.1X je centralizovaná správa pre skupiny užívateľov bezdrôtovej siete.[14]

3.5 Vytvorenie počítačovej aplikácie

Pre overenie použitia modulu v reálnej aplikácii som sa rozhodol vytvoriť počítačovú aplikáciu, ktorá bude schopná konfigurovať samotný modul. Následne je možné monitorovať jednotlivé vstupné brány prípadne nastavovať hodnotu výstupných brán. Všetky operácie sa vykonávajú v príkazovom móde samotného modulu. Tento mód je náročnejší na objem komunikácie a odozvu a preto lepšie testuje možnosti samotného modulu. Väčšina aplikácii bude využívať modul v dátovom móde.

3.5.1 Nadviazanie TCP komunikácie

Pre vytvorenie komunikácie som volil použitie „zásuvných modulov“ (socket) vytvorených pre Microsoft Visual C++. Samotný názov nám napovedá ako tieto softwarové moduly pracujú a čo a ako fungujú. Pre vytvorenie spojenia s vzdialeným počítačom prípadne zariadením vytvoríme tzv. „vlákno“. Je to symbolický názov pre spojenie medzi počítačom a vzdialeným počítačom. Tieto vlákna sú spojené s jednotlivými modulmi. Pre nadviazanie spojenia je potrebné, aby oba počítače prípadne zariadenia boli schopné prijímať rovnaké komunikačné vlákno. Tieto vlákna komunikujú na určitých portoch. Tento port je jedinečný pre každý komunikačný kanál a predstavuje jedinečné označenie daného komunikačného kanálu. Vďaka tomu je možné na jednom počítači nadviazať viacero paralelných komunikácií. Pre sieťovú komunikáciu je vytvorené niekoľko portov s rezervovaným použitím (port 7- ping, port 13- čas, port 15- status siete, port 23 vzdialený prístup,...). Aj napriek tomu je možné použiť mnoho ďalších portov, ktorým nebola priradená žiadna služba a sú vhodné pre nadviazanie komunikácie. Vo všeobecnosti platí, že porty v rozsahu od 1000 až 6535 je možno používať pre vytvorenie komunikácie. Pre jednoduchšie pochopenie uvediem príklad:

Internet Explorer a iné prehliadače využívajú port 80. Tento port využívajú ako na prijímanie tak na vysielanie dát. Pre posielanie mailov na vzdialený poštový server využívame port 25. Pokiaľ prijmeme mail tak na načítanie mailu je najčastejšie vyžívaný port 110. [15]

Na predošlých príkladoch je zrejmé, ako niektoré základné služby fungujú. Pre komunikáciu je potrebné definovať aj samotný koncový počítač prípadne zariadenie s ktorým chceme nadviazať spojenie. Preto je potrebné poznať taktiež IP adresu tohto zariadenia. IP adresa je identifikačné číslo ktoré je priradené ku každému počítaču prípadne zariadeniu v sieti. Jeho tvar sa skladá zo štyroch decimálnych čísel v rozsahu od 0 do 255 oddelenými bodkami. Pre nadviazanie spojenia s iným počítačom, prípadne serverom je potrebné aby sme zadávali jedinečnú IP adresu miesta ku ktorému sa chceme pripojiť pomocou siete. To je potrebné aj pri prehliadaní internetových stránok, kde je potrebné poznať IP adresu serveru ku ktorému sa plánujeme pripojiť. Je zrejmé, že zadávanie IP adresy každej webovej stránky je nezmyselné a ľudia nie sú schopní si tieto adresy zapamätať. Preto boli tieto adresy nahradené doménovými menami. Preto je možné pristúpiť ku vzdialenému serveru zadaním napríklad www.google.com ktoré sú vlastne náhradou IP adresy. Prehliadač túto IP adresu získava z tabuľky prípadne je vyžiadaná od vzdialeného serveru. Po obdržaní tejto IP adresy prehliadač adresuje svoju komunikáciu na adresu samotného serveru alebo koncového užívateľa alebo zariadenie.[15]

Pre naše účely sa tento proces vykonáva bez toho, aby sme tento mechanizmus museli programovať, pokiaľ použijeme už spomínané zásuvné moduly. Samotný modul je definovaný ako binárne kompatibilné sieťové programovacie rozhranie pre systém Microsoft Windows. Windows moduly sú založené na implementácii systému UNIX a sú špecifikované štandardom BSD. Windows moduly umožňujú aplikáciám komunikovať cez akékoľvek

sieťové rozhranie , ktoré zodpovedá rozhrania API Windows modulu . Pre systémy založené na Win32 , Windows moduly zaisťujú spoľahlivé komunikačné vlákno. Modul sa z hľadiska komunikácie javí ako koncové zariadenie, cez ktoré je Windows schopný prijímať a vysielat' dáta cez sieť. V súčasnej dobe je možné využívať dva typy modulov:

- stream modul: tento modul sa dá využívať pre dátový tok.

- datagramový modul: tento modul je zodpovedný za prenos dát, no nie je zaručené že dáta budú prijaté/ vyslané prípadne či budú vyslané/prijaté v správnom poradí.[15]

Oba typy modulov sú schopné vysielat' a prijímať dáta súčasne. Pre pripojenie modulov do samotného návrhu aplikácie je potrebe v inicializácii pridať nasledovné:

```
using System::Net::Sockets::TcpClient;
using System::Net::Sockets::NetworkStream;
```

tieto direktívy nám sprístupnia moduly potrebné pre nadviazanie spojenia, vysielanie dát, príjem dát, uzatvorenie spojenia na rozhraní TCP. Následne je potrebné pokúsiť sa nadviazať spojenie. Pre toto je potrebné poznať klientovu IP adresu a port na ktorom bude komunikácia prebiehať. Trieda TcpClient (String, int32) vytvorí inštanciu triedy TcpClient. Následne je potreba vytvoriť dátový prúd. Na to slúži trieda NetworkStream. Pre vytvorenie príslušného dátového prúdu trieda TcpClient obsahuje prístupovú metódu GetStream. Na základe týchto znalostí je možné vytvoriť spojenie pomocou TCP. Nasledovný príklad ukazuje možnosť nadviazania spojenia:

```
// vytvorenie premenných s IP adresou a portom
String^ server="192.168.1.1";
Int32 port = 13000;

// nadviazanie spojenia s názvom client, triedy TcpClient
TcpClient^ client = gcnew TcpClient( server,port );

// vytvorenie dátového prúdu s názvom stream
NetworkStream^ stream = client->GetStream();
```

Po nadviazaní spojenia a vytvorení dátového prúdu je možné začať čítať a vysielat' dáta. Uvediem jednoduchý príklad čítania dát. Pre čítanie je potreba overiť či je možné čítať dáta v danom momente. Na to nám slúži verejná vlastnosť CanRead, ktorá nám vráti logickú hodnotu. Pre samotné čítanie dát je potrebné použiť prístupovú metódu Read. Táto metóda vyžaduje tri parametre, a to názov reťazca znakov, adresu prvého načítaného bitu a maximálny index pola.[15]

```
if (stream ->CanRead )
{
    array<Byte>^ myReadBuffer = gcnew array<Byte>(1024);
    int numberOfBytesRead = 0;
    // čítanie prijatej správy
    numberOfBytesRead =stream->Read( myReadBuffer,0,myReadBuffer->Length );
}
```

```
}
```

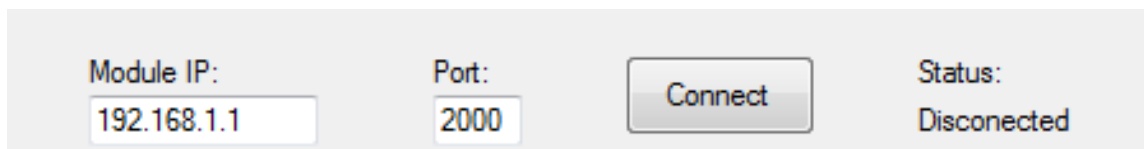
Pri vysielaní správy je situácia veľmi podobná. Na overenie či v danom momente je možné čítať dáta slúži verejná vlastnosť `CanWrite`. Na zápis dát je možno použiť prístupovú metódu `Write`. Táto metóda taktiež obsahuje tri argumenty a to názov reťazca vysielaných znakov, index prvého vysielaného znaku a index posledného vysielaného znaku.[15]

```
if (stream ->CanWrite )
{
    array<Byte>^ myWriteBuffer=Encoding::ASCII->GetBytes( "Message" );
    stream ->Write( myWriteBuffer, 0, myWriteBuffer->Length );
}
```

Pre uzavretie komunikácie je potrebné použiť verejnú metódu `Close`. Príklad použitia:

```
client->Close();
```

na základe týchto principiálnych príkladov sme schopný používať zásuvné moduly bez toho, aby sme potrebovali ďalšie znalosti o danom protokole. V samotnej aplikácii je vytvorenie a uzavretie komunikácie vyriešené pomocou nasledovných nástrojov:

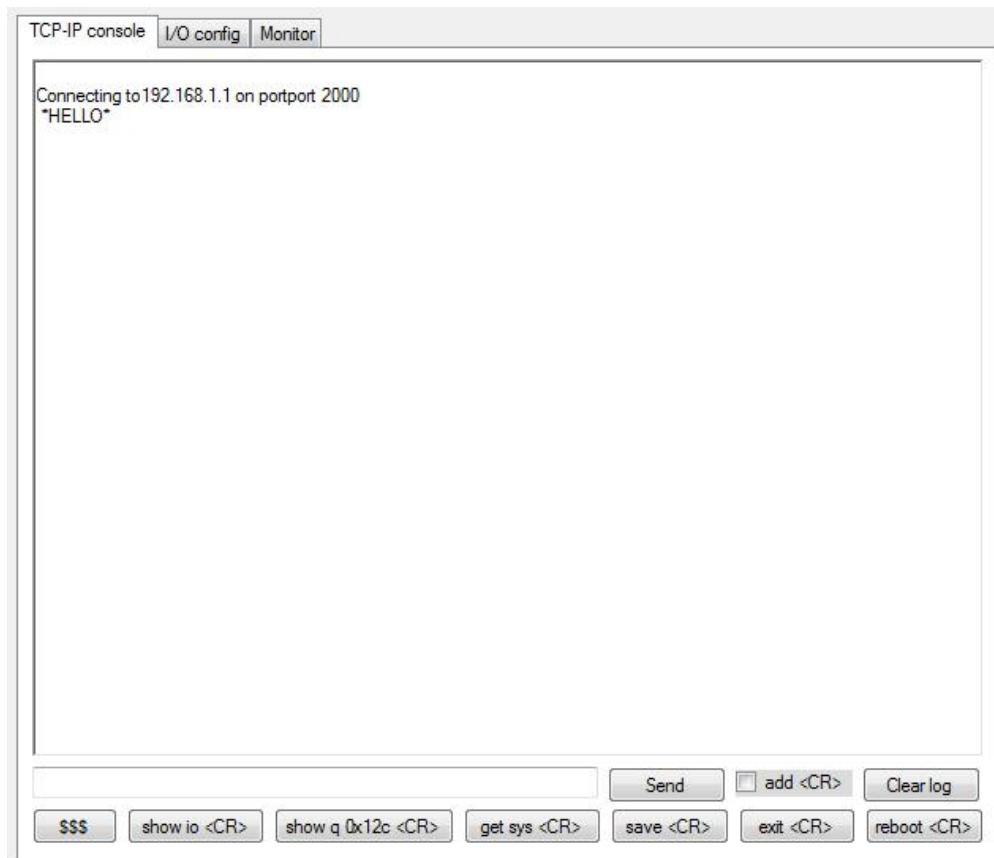


Obr. 3.5: Náhľad na možnosti pripojenia a odpojenia

Ako je vidieť na obrázku, užívateľ môže zadať IP adresu a port na ktorý sa chce pripojiť. Zásuvný modul dovoľuje do okna s titulkom `Module IP` zadať taktiež doménové meno vzdialeného serveru. Pomocou týchto vstupných dát je možné pripojiť sa aj k iným zariadeniam nie len k mnou použitému Wifly modulu. Po zvolení tlačidla `Connect` program sa pokúsi o nadviazanie spojenia. Je jasné, že sa to nemusí podariť ale aplikácia sa o to opakovane pokúsi, a po uplynutí interne definovaného času, sa v konzolovom okne objaví správa „Unable to connect...“. V opačnom prípade, ak príde k nadviazaniu spojenia, príde k zmene statusu na „Connected“ a popis a funkcia tlačidla sa zmenia na `Close`. Po stlačení tohto tlačidla príde ku zatvoreniu komunikácie.[15]

3.5.2 Popis záložky TCP-IP console

Táto záložka nám slúži pre simuláciu konzoly do ktorej je možno vkladať ľubovoľný text. Pomocou tlačidla Send je text odovzdaný na funkcii pre vysielanie dát. Pomocou tejto konzoly je možné komunikovať aj s inými sieťovými zariadeniami nie len s nami využívaným modulom. Na obrázku je zobrazený náhľad na možnosti vysielania po pripojení.

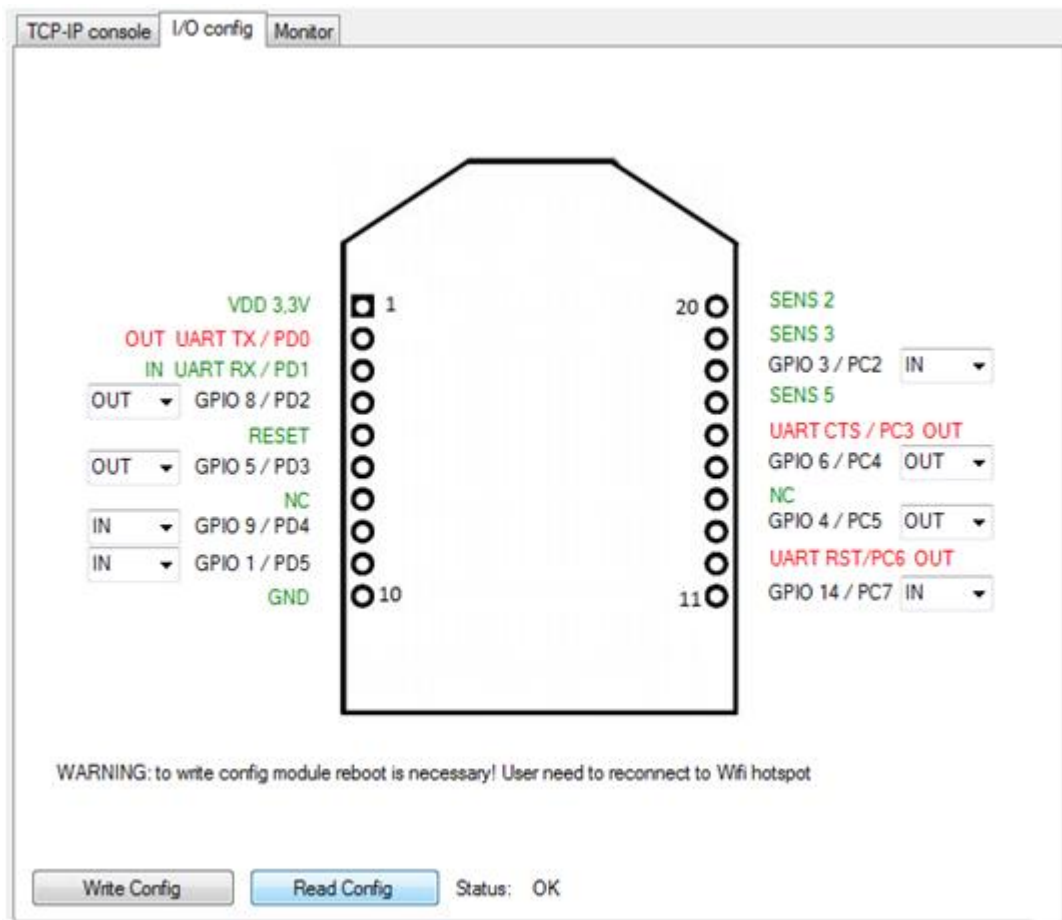


Obr. 3.6: Náhľad na záložku TCP-IP console

Ako je vidieť na obrázku do poľa konzoly sú vypisované aj systémové správy. Pre jednoduchosť a rýchlejšie používanie WiFly modulu pre ktorý je táto aplikácia určená je tu možno pomocou ostatných tlačidiel posielat' niektoré základné príkazy. Pri posielaní vlastného textu je dobré si všimnúť, že vedľa tlačidla Send je zaškrťavacie okno s nápisom add<CR>. Ak je táto položka zaškrtnutá je k textu na odoslanie pripojené zalomenie riadku a to znaky $x0D$ čo je znak návratu vozíka a $x0A$, čo je znak nového riadku. Táto možnosť je tu pre samotný WiFly modul. Pre to aby modul mohol správne prečítať príkaz, je potrebné aby, prijal príkaz aj so zalomením riadku, v opačnom prípade modul daný reťazec nevyhodnotí ako príkaz.

3.5.3 Popis záložky I/O config

Modul obsahuje vstupno-výstupné porty ktoré je možno užívateľsky nastavovať. Pre jednoduchú a názornú kontrolu nastavení nám slúži táto záložka. Na obrázku je vidieť príklad nastavení.

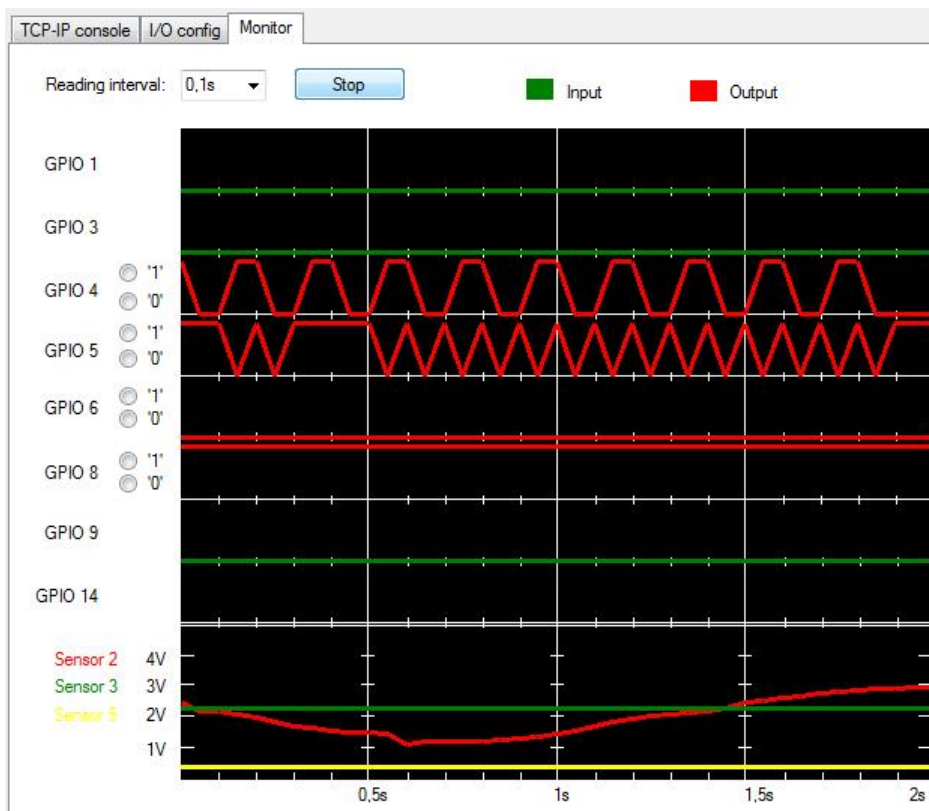


Obr. 3.7: Náklad na záložku I/O config

Pre čítanie aktuálnych nastavení je vytvorené tlačidlo Read Config. Po stlačení tlačidla sa nám nastavia jednotlivé roletové možnosti podľa aktuálneho nastavenia modulu. Je zrejme že všetky pozície nie sú označené, prípadne nie je možnosť si ich meniť. Niektoré vývody sú potrebné pre samotnú funkciu modulu (VDD, RESET, GND), iné vyžadujú odlišné hardwarové riešenie (SENS 2, SENS 3, SENS 5) a niektoré sú určené pre rýchlejšiu a rozširujúcu komunikáciu samotného modulu a to o UART komunikáciu (UART RX, UART TX, UART CST, UART RST). Užívateľ okrem čítania samotných nastavení je schopný tieto nastavenia meniť. Po nastavení konfigurácie aká vyhovuje užívateľovi je potrebné zvoliť tlačidlo Write Config. Tým sa vykoná sled príkazov potrebný pre nastavenie vstupno-výstupných portov, tak ako si to užívateľ želá. Pozor! Po zatlačení tohto tlačidla je potrebné modul reštartovať, čím príde k uzavretiu komunikačného kanálu. Podrobnejšie bude tento proces vysvetlený neskôr.

3.5.4 Popis záložky Monitor

Pre nastavovanie a prezentáciu meraných dát som volil jednoduché a užívateľsky príjemné grafické rozhranie. V tejto záložke užívateľ nastavuje časovú základňu pomocou rolety a po stlačení tlačidla Start program overí nastavenie jednotlivých portov a na základe toho priradí farby ku grafom. Pri výstupných portoch je možné nastavovať aj logickú úroveň. V tejto záložke je zobrazený aj analógový senzor. Príklad možných výstupov je na obrázku.



Obr. 3.8: Náklad na záložku Monitor

Po spustení merania sa overí nastavenie jednotlivých brán a na základe toho sa volí farba pre vykresľovanie a aj možnosť nastavovania logickej úrovne. Táto komunikácia bude vysvetlená neskôr. Pre voľbu časovej základne je možné vybrať z troch možností a to 1s, 0,5s a 0,1s. pre meranie je využitý časovač, ktorý je vždy nastavený na polovicu času ako je zvolený a preto počas doby jednej časovej základne je signál meraný dva krát. Senzorické vstupy sú merané, prevedené na decimálnu hodnotu a následne zobrazené tak, aby zodpovedali päť-voltovému rozsahu. Tieto grafy sú vzájomne odlišené farbou jednotlivých senzorických vstupov. Ako je zrejme pre získanie týchto dát, prípadne zápis výstupných dát, je potrebné použiť interné príkazy WiFly modulu. Je to aj z toho dôvodu, aby sme overili čo najpresnejšie jeho reálne vlastnosti. Táto komunikácia je náročnejšia a bude vysvetlená v neskôr.

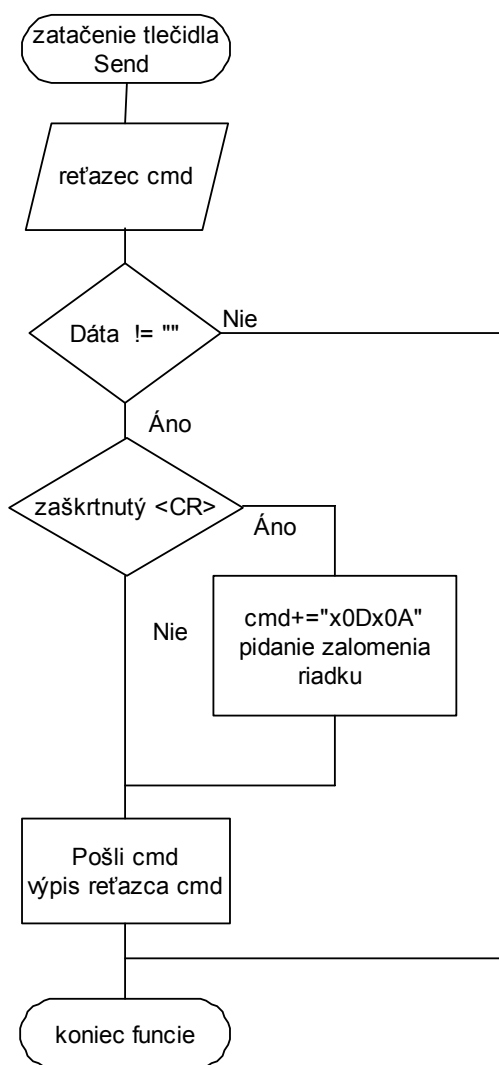
Pre zastavenie monitorovania dát sa tlačidlo Start zmení na tlačidlo Stop. Po stlačení tohto tlačidla sa monitorovanie zastaví, ale namerané dáta budú stále zobrazené v grafe. Ku

zmazaniu príde až po opätovnom spustení monitorovania.

3.6 Popis komunikácie a algoritmov pre čítanie a zápis

3.6.1 Tlačidlo Send

Algoritmus zodpovedný za toto tlačidlo je pomerne jednoduchý. Z užívateľského hľadiska sa jedná o jednoduché zasielanie dát. Pre účely tejto práce tlačidlo Send použijem najmä ako príklad jednoduchého vývojového diagramu, ktorý podrobne vysvetlím a v zložitejších vývojových diagramoch sa budem o to menej venovať jednoduchším detailom. Na obrázku je možno vidieť vývojový diagram zodpovedný za tlačidlo Send.



Obr. 3.9: Vývojový diagram tlačidla Send

Na tomto vývojovom diagrame je vidieť algoritmus po stlačení. Po začiatku funkcie je vytvorená premenná typu reťazec a názvom cmd (String[^] v Microsoft Visual C++). Následne si funkcia kontroluje či existujú dáta na odoslanie. Tieto dáta by sa mali nachádzať v textovom poli naľavo od tlačidla Send. Pokiaľ je textové pole prázdne ukončí sa daná

funkcia (možnosť Nie). Pokiaľ textové pole je nerovné prázdnej reťazcu prichádza k testovaniu zaškrťavacieho okna <CR>. Pokiaľ je dané okno zaškrtnuté ku reťazcu je pripočítané zalomenie riadku (konkrétne hodnoty x0D a x0A). Následne po oboch možnostiach je reťazec cmd vyslaný pomocou TCP spojenia a zároveň vypísaný do konzolového okna. Po tejto operácii sa ukončí táto funkcia.

3.6.2 Tlačidlo Read config

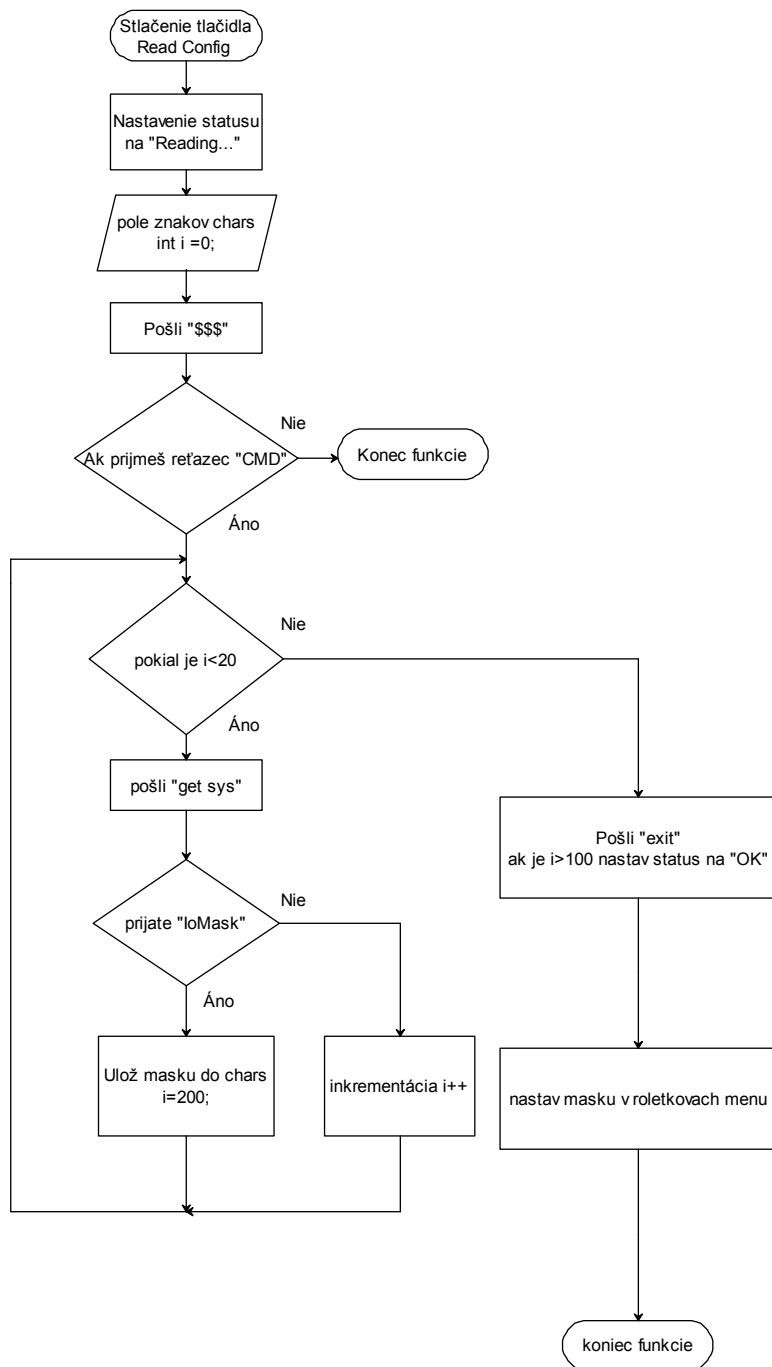
Toto tlačidlo je zodpovedné za načítanie aktuálnej konfigurácie modulu a pre to je potrebné previesť sled nasledovných príkazov aby prišlo k výmene dát medzi aplikáciou a modulom:

Príkaz:	Popis príkazu:	Odpoveď modulu:
<u>\$\$\$</u>	vstup do príkazového módu	CMD
<u>get sys<CR></u>	vyžiadanie systémových informácií. Vďaka tomuto príkazu je možné obdržať niektoré systémové nastavenia, medzi ktoré patrí aj nastavenie masky vstupno-výstupných brán. Táto hodnota je vyčíslená v položke IoMask, kde je reprezentovaná hexadecimálnou hodnotou. V tomto prípade logická nula znamená že sa jedná o vstup a logická jednotka znamená výstup	getsys SleepTmr=0 WakeTmr=0 Trigger=0x1 Autoconn=0 IoFunc=0x0 IoMask=0x21b0 IoValu=0x0 DebugReg=0x0 PrintLvl=0x1 <2.45>
<u>exit <CR></u>	opustenie príkazového módu	EXIT

Ako je vidieť je potrebné vyčítať vstupno-výstupnú masku. V tomto prípade je maska na module nastavená na hodnotu 0x21b0. Túto hodnotu je potrebné previesť do binárneho tvaru a na základe týchto hodnôt je potrebné vyhodnotiť či sa jedná o vstup alebo výstup. Vstupe odpovedá logická hodnota nula a logická hodnota jedna zodpovedá výstupu.

Podobne pracuje aj samotný program. V tomto prípade je pred samotnou komunikáciou potrebné nastaviť status na čítanie a vytvoriť pomocné premenné. Pre moju aplikáciu používam pomocnú premennú *i* ktorá mi slúži na predčasné ukončenie slučky. Následne po týchto krokoch je vyslaný reťazec \$\$\$\$. Nakoľko je nutné, aby sa užívateľ dostal do príkazového módu kontrolujem či sa tak stalo. Pokiaľ nie, predpokladám že nastala chyba pri komunikácii a túto funkciu ukončím so zmenou statusu na neúspešné čítanie. Pokiaľ ale vstúpim do príkazového módu, začnem cyklicky vysielat' žiadosť na aktuálne nastavenie. Túto žiadosť pošlem maximálne dvadsať krát. Je to z toho dôvodu, že odpoveď modulu je pomerne dlhá a nie vždy je prijatá ako jedna správa. Môže nastať situácia, že je zaslaná žiadosť na konfiguráciu, aplikácia prijme dáta, ale v tento okamih neprijme celú správu a pri overovaní prijatej správy, táto informácia ešte nebola prijatá. Preto je treba proces zopakovať,

tak aby prišlo ku prijatiu kompletnej správy. Pokiaľ je správa prijatá správne, nastavím pomocnú premennú na vysokú úroveň, aby prišlo k ukončeniu slučky. Ak na druhú stranu správa nie je prijatá správne, inkrementujem pomocnú premennú, čím zaistím taktiež ukončenie tejto slučky no s maximálnym počtom opakovaní uvedených v podmienke. V tomto prípade je to maximálne dvadsať pokusov. Následne po ukončení slučky je potrebné opustiť príkazový mód. Následne príde k prepočítaniu získaných dát tak, aby sme boli schopný určiť či sa jedná o vstup alebo výstup. Program sám nastaví jednotlivé roletové menu tak, aby užívateľ názorne videl nastavenie modulu. Tento algoritmus je zobrazený na nasledovnom vývojovom diagrame.



Obr. 3.10: Vývojový diagram tlačidla Read config

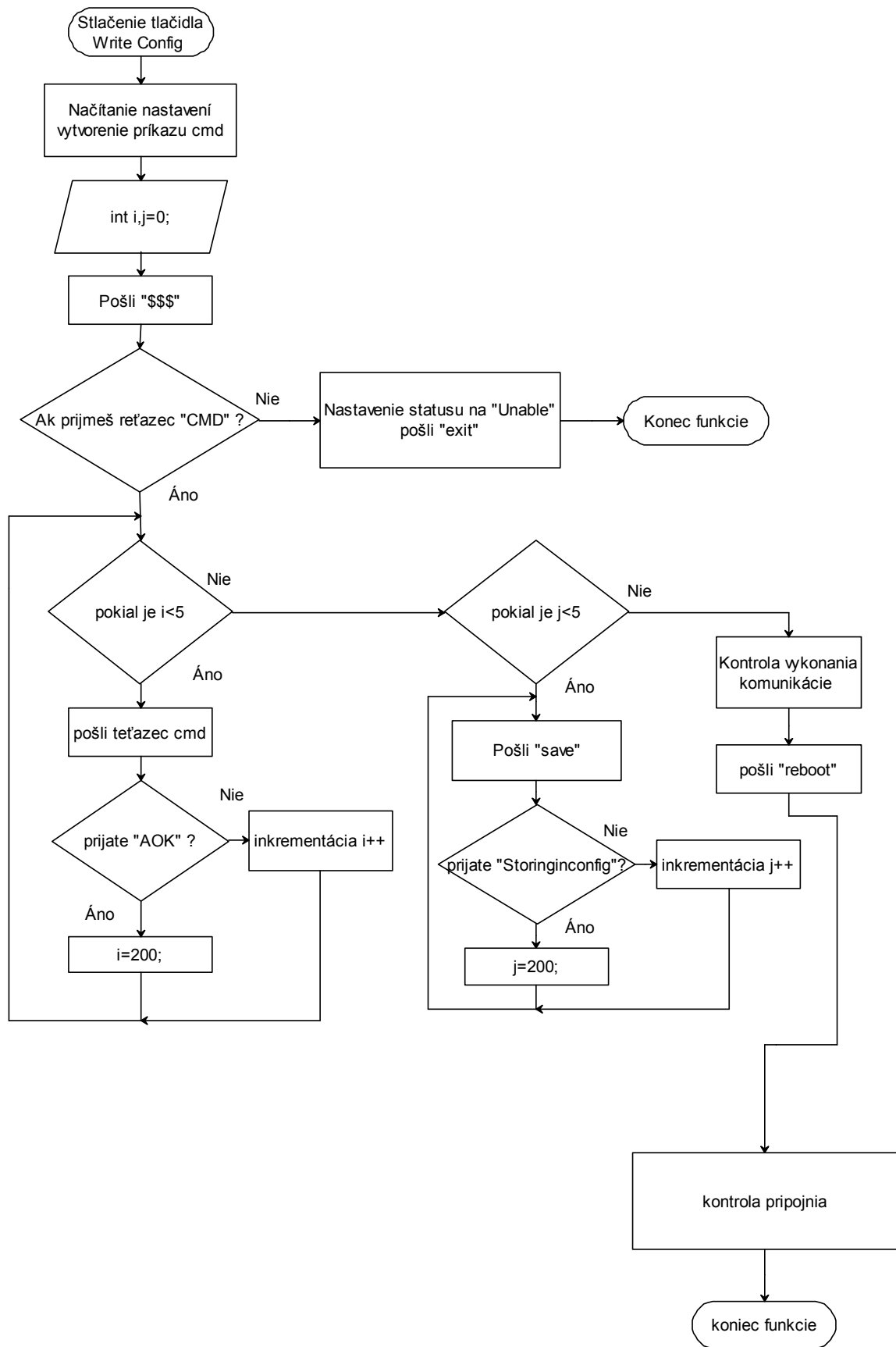
3.6.3 Tlačidlo Write config

Toto tlačidlo je zodpovedné za nastavenie novej konfigurácie modulu čo je náročnejšie a komplikovanejšie v ukladaní a následnom reštarte modulu. Pre samotné nastavenie je potrebné vykonať nasledovný sled príkazov

Príkaz:	Popis príkazu:	Odpoveď modulu:
<u>\$\$\$</u>	vstup do príkazového módu	CMD
<u>set sys mask 0x21f0<CR></u>	Tento príkaz nastaví príslušnú masku pre vstupné a výstupné porty	setsysmask0x21f0 AOK <2.45>
<u>Save<CR></u>	Uloženie nastavenia.	Save Storinginconfig <2.45>
<u>reboot<CR></u>	Riadený reštart modulu. Tento príkaz je potrebný nakoľko až po opätovnom štarte modulu sa prejavia nakonfigurované zmeny	reboot

Ako je možno vidieť z ukážky, je potrebné nielen aktuálne nastavenia len uložiť ale aj modul reštartovať. V príklade nastavujem výstupnú masku na hodnotu 21f0, ktorá nastaví porty 4,5,6,7,10 a 13 ako výstupné, všetky ostatné porty budú nastavené ako vstupné.

V programe je tento algoritmus podobný ako v predchádzajúcom prípade no s istými zmenami. Po štarte funkcie je potrebné načítať nastavenie užívateľa a následne prepočítať aktuálnu masku. Tieto hodnoty ukladám do premennej cmd ktorá sa skladá z reťazca *set sys mask 0x* a vypočítanej masky. Následne je potrebné vytvoriť dve pomocné premenné pre dve slučky programu. Jedna slučka je zodpovedná za nastavenie samotnej masky a druhá za uloženie nastavenia. Pre prípad, ak by prišlo ku chybe na komunikácii a modul by neodpovedal správne, vykoná sa daný príkaz opätovne, no maximálne päť krát. Skrátenie tejto slučky je podobné ako v predchádzajúcom prípade. Následne sa kontroluje či daný proces prebehol v poriadku a pokiaľ tomu tak je, vyšle sa nasledovný príkaz pre reštart. Tento príkaz je pre naše účely nevýhodný nakoľko modul sa na krátky čas od siete odpojí a samotný operačný systém stratí spojenie s modulom. Preto je potrebné overenie pripojenia s modulom. Väčšinou príde k ukončeniu komunikácie a je potrebné opätovné pripojenie k modulu. Z užívateľského hľadiska je proces opätovného pripojenia o niečo komplikovanejší. Operačný systém nezaznamená zmenu v pripojení a bude sa hlásiť ako neustále pripojený, no samotná aplikácia nebude schopná sa opäť k modulu pripojiť. Je to z toho dôvodu, že samotný modul stratí informáciu o tom že je pripojený v sieti a po reštarte sa k nej opäť nepripojí. Preto je potrebné, aj v danej aplikácii overiť či je dané pripojenie ešte stále aktívne. Tento algoritmus je uvedený v nasledovnom vývojovom diagrame.



Obr. 3.11: Vývojový diagram tlačidla Write config

3.6.4 Tlačidlo pre zápis výstupu

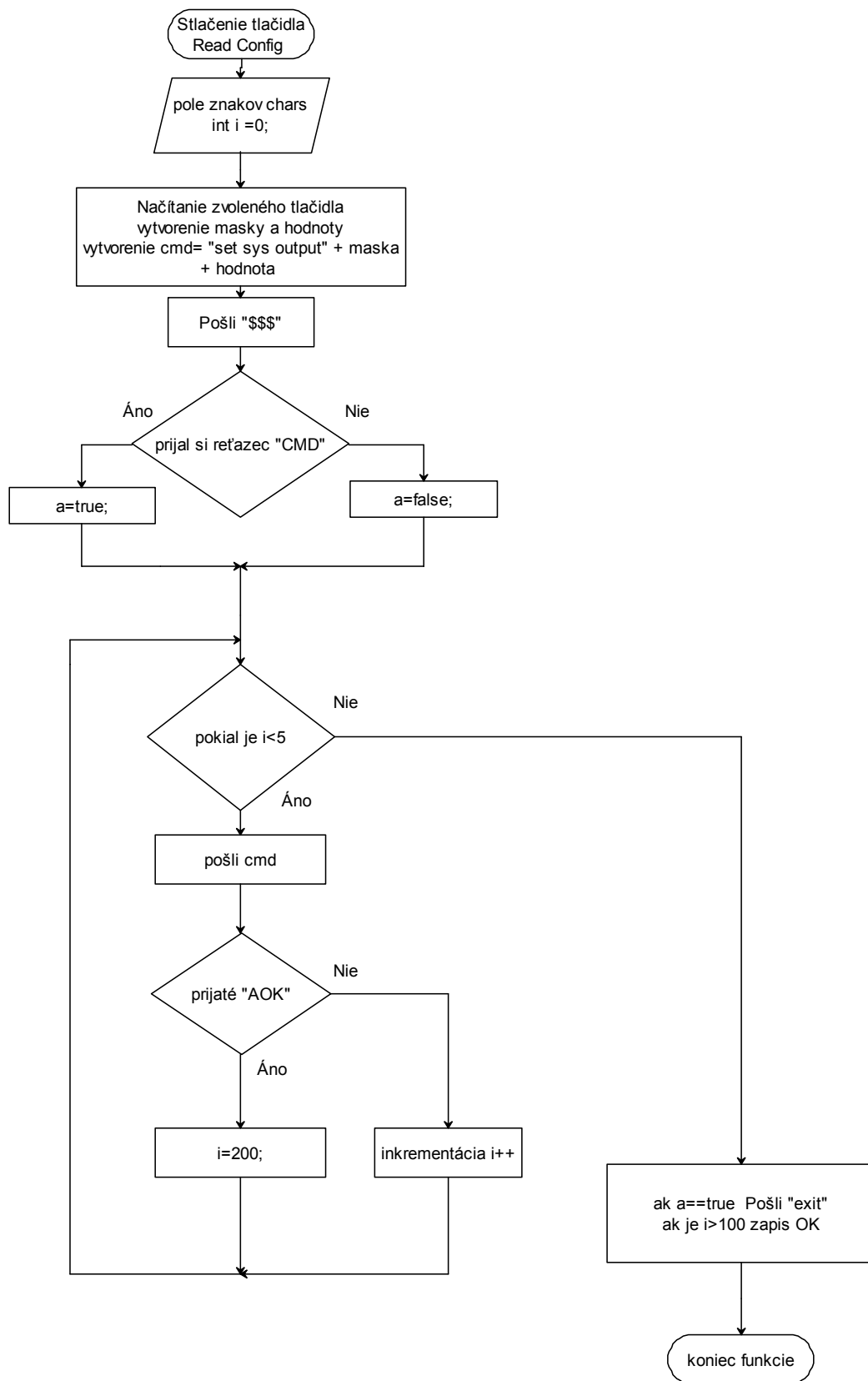
Pomocou sledu nasledovných príkazov je možné nastaviť výstupnú hodnotu na jednotlivých portoch modulu.

Príkaz:	Popis príkazu:	Odpoveď modulu:
<u>\$\$\$</u>	vstup do príkazového módu	CMD
<u>set sys output 0x0100 0x0100<CR></u>	tento príkaz nastavuje výstup na určitú hodnotu podľa dvoch hexadecimálnych hodnôt. Prvá hodnota určuje hodnotu na danom porte (0 odpovedá logickej nule 1 logickej jednotke) a druhá určuje port podľa masky	setsysoutput0x0100 0x0100 AOK <2.45>
<u>exit <CR></u>	opustenie príkazového módu	EXIT

Tento sled príkazov je vykonaný rýchlo a pomerne jednoducho. Zdanlivo sa aj v programe jedná o jednoduchý algoritmus no z praktických skúseností vyplynulo, že tomu tak nie je.

Algoritmus je veľmi podobný ako v predchádzajúcich príkladoch. Inicializujem pomocnú premennú *i* pre ukončenie slučky zápisu a booleovskú premennú *a*. Po tom príde k vytvoreniu reťazca cmd podľa stlačeného tlačidla a požadovanej hodnoty. Následne sa vyšle povel pre vstup do príkazového módu. Je skontrolované či prišlo ku vstupu do príkazového módu. Tento údaj sa odpamätá v premennej *a*. Následne sa vykoná slučka pre zápis už vytvoreného príkazu a po úspešnom vykonaní je kontrolované či prišlo k vykonaniu a v prípade ak nastal vstup do príkazového módu je vyslaný príkaz pre opustenie príkazového módu. Následne je ukončená funkcia pre zápis.

Je jasné, že je kontrolovanie či prichádza k vstupu do príkazového módu alebo nie, má isté opodstatnenie. To je z toho dôvodu, že tlačidlá pre zápis výstupných hodnôt sú viditeľné až po tom, ako príde k monitorovaniu dát. Preto je možné, že príde ku kolízii medzi algoritmom pre monitorovanie dát a zápis. Pokiaľ by neprichádzalo ku tejto kontrole, prichádzalo by, ako ku strate monitorovaných dát, a neprebehlo by úspešne zapísanie výstupných dát.



Obr. 3.12: Algoritmus zápisu výstupov

3.6.5 Monitorovanie dát

Po nastavení správnej časovej základne a stlačení tlačidla Start prebehne nasledovný sled príkazov

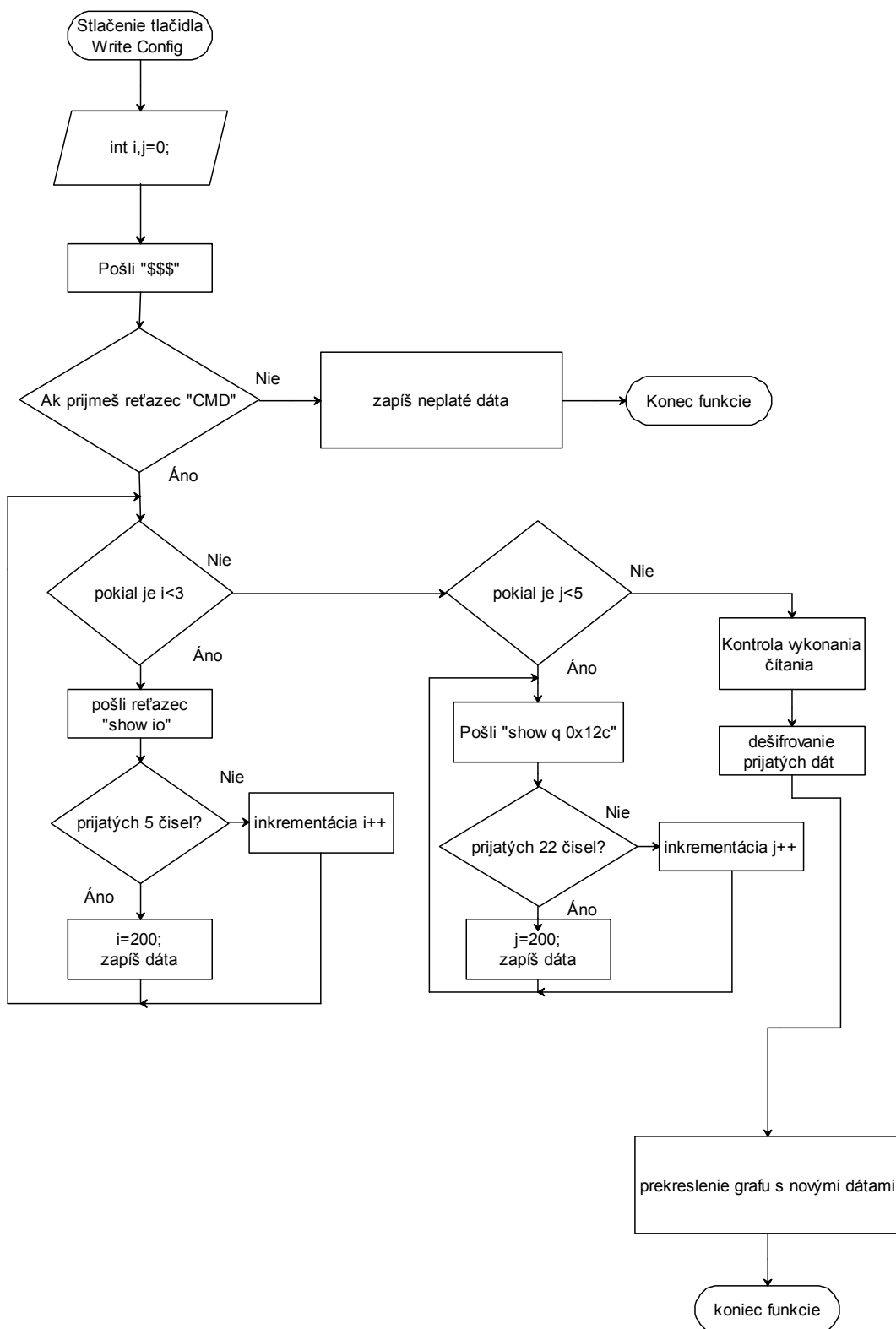
Príkaz:	Popis príkazu:	Odpoveď modulu:
<u>\$\$\$</u>	vstup do príkazového módu	CMD
<u>show io<CR></u>	Vyčítanie hodnoty vstupných a výstupných portov. Tento príkaz vráti hodnotu odpovedajúcu všetkým portom ako vstupným tak výstupným. Najvyšší bit je vždy jedna.	showio 8500 <2.45> show q 0x12c showq0x12c
<u>show q 0x12c<CR></u>	Vyčítanie viacerých senzorov naraz. Maska pre vyčítanie je 2c. Príkaz vždy začína hodnotou 0x1XX kde XX je maska meraných senzorov	showq0x12c 823d64,84398a,806d38, <2.45>
<u>exit<CR></u>	opustenie príkazového módu	EXIT

Tento sled príkazov vyčíta ako hodnotu na digitálnych vstupoch tak aj na senzorech. Ako je možné vidieť na príklade digitálny vstup vracia štvorbajtovú hodnotu pričom najvyšší bit je vždy nastavený na jednotku. V prípade senzoru je hardwarovo delené vstupné napätie na polovicu čo pri 5V na vstupe odpovedá maximálnej hodnote 2,5V na vstupe WiFly modulu. Pri tomto napätí A/D prevodník načíta hodnotu v dekadickom tvare maximálne 400000. Ako je vidieť na príklade každá hodnota analógového vstupu začína znakom 8 a je oddelená čiarkou.

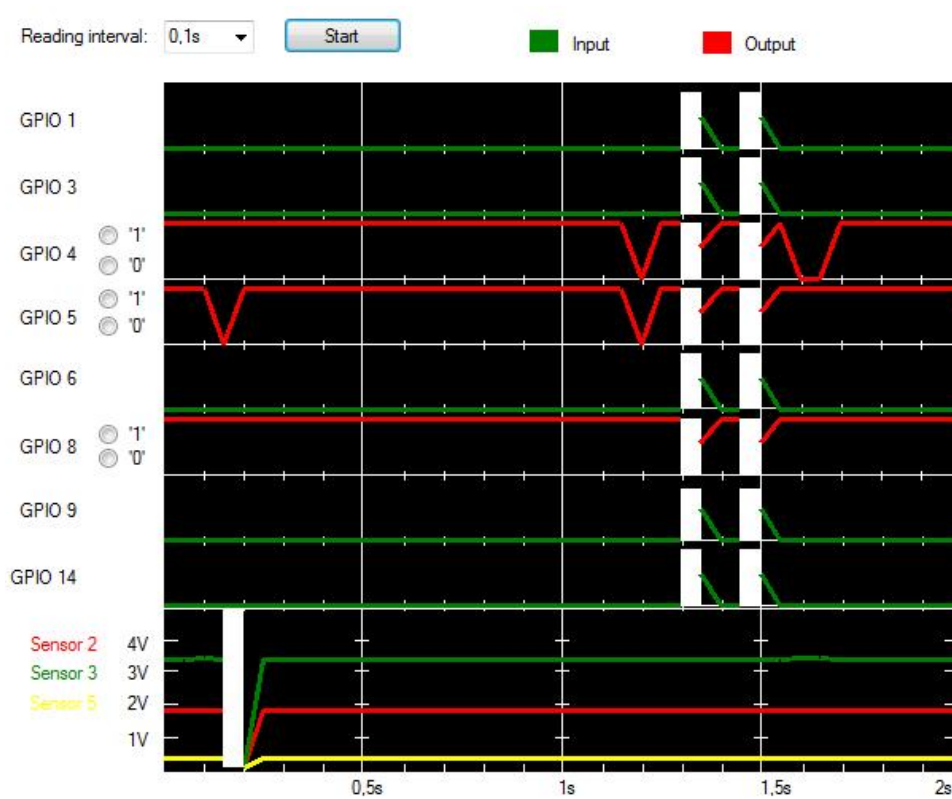
Algoritmus samotného čítania je veľmi podobný predchádzajúcim príkladom, no s tým rozdielom, že je kladený doraz na to, aby modul trávil v príkazovom móde čo najmenej času a zároveň aby komunikácia prebehla čo najspolahlivejšie. Preto je obmedzený počet cyklov v slučkách. Jednou z metód ako zaistiť minimálny čas modulu v príkazovom móde je ak nespracovávame dáta počas tejto doby a tejto činnosti sa venovať až po ukončení komunikácie. V prípade senzorov nastáva isté spomalenie z dôvodu oneskorenia samotného prevodu a aj pre spracovanie vstupných dát sa jedná o náročnejší proces. Aj napriek týmto opatreniam sa stáva že nepríde ku prijatiu dát prípadne neplatných dát. Nakoľko výpadok dát je reálna vlastnosť modulu, je potrebné aj tento poznatok reprezentovať.

Chyby v komunikácii a neprijatí dát je zaistené oddelene pre analógový a digitálny signál. Oba typy majú však podobný systém rozlišovania platných a neplatných dát. Ako digitálne tak analógové hodnoty sú ukladané v poli dát. Každá bunka tohto poľa môže nadobúdať hodnotu ktorá odpovedá trom úrovniam a to prázdna hodnota, užitočná hodnota, hodnota reprezentujúca neprijaté dáta. Pri vykresľovaní prázdna hodnota nie je vykreslená. Užitočné dáta sú vykreslené ako bežné dáta a v prípade neprijatých dát je vykreslený obrazec

naznačujúcu výpadok. Takto prezentované dáta sú zrejme a jednoducho čitateľné pre užívateľa.



Obr. 3.13: Algoritmus komunikácie monitorovania dát



Obr. 3.14: Príklad monitorovania s výpadkom dát

Ako je možné vidieť na obrázku vždy príde k výpadku buď analógovému signálu prípadne digitálnemu. Je to toho dôvodu, že pri prijatí dát sú digitálne a analógové dáta rozdelené do dvoch správ a preto je možné každé z týchto dát vyhodnotiť samostatne a neprichádza ku zbytočnej strate údajov.

4 ZÁVER

V tejto práci som mal navrhnúť čo najuniverzálnejší modul pre bezdrôtový prenos dát. Jednalo sa konkrétne o WiFi rozhranie. Daný modul mal byť schodný pohodlne preniesť a reprezentovať namerané dáta. Bolo potrebné dbať aj na kompatibilitu jednotlivých komponentov a rozličných možných senzorov. V mojej práci som sa snažil zadanie plniť čo najpresnejšie a najefektívnejšie.

V priebehu vykonávania samotnej práce boli jednotlivé požiadavky presnejšie špecifikované a v priebehu semestra sa menili, čím prichádzalo ku zvyšovaniu nárokov na samotný produkt. Jednou z požiadaviek na univerzálnosť bolo použitie samotnej vývojovej dosky EvB 5.1 čo na prvý pohľad malo samotný návrh zjednodušiť a vytvoriť priestor pre tvorbu počítačovej aplikácie. Pravdou je, že vývojová doska priniesla do hardwarového prevedenia viacero neočakávaných komplikácií a samotný návrh skomplikovala. Komplikácie sa vyskytovali najmä v rozličných napäťových úrovniach a prísnej viazanosti na použité súčiastky a ich zapojenie.

Samotná aplikácia poskytuje široké možnosti univerzálnosti samotného návrhu, čo veľmi prísne dodržiava vopred určené požiadavky. Je určená ako pre dátový, tak aj konfiguračný mód samotného modulu. Je schopná obojstranne prenášať požadované dáta, konfigurovať samotný modul a následne aj monitorovať jednotlivé vstupy. Monitorovanie je výhradne v príkazovom móde, kde sa využívajú monitorovacie funkcie samotného modulu. Je to z toho dôvodu, aby prišlo k overeniu rýchlosti prenosu dát v najhoršej možnej variante. Rýchlejší spôsob prenosu dát je pomocou dátového módu modulu, kedy dáta vyslané cez aplikáciu modul reprezentuje ako UART dáta. Táto komunikácia funguje obojsmerne. Vytvorenie a návrh tejto komunikácie je zatiaľ na samotnom užívateľovi.

V čase zadávania práce nebolo presne špecifikované, kde sa v praxi bude samotný modul používať a aké bude mať využitie. V súčasnej dobe je zrejmé, že moja práca poslúži k rýchlej a jednoduchej konfigurácii modulu a následne položí základy pre vývoj nových aplikácií samotného modulu. Modul bude s najväčšou pravdepodobnosťou pracovať v dátovom móde a v ťažko prístupných miestach, kde bude monitorovať potrebné veličiny, prípadne bude vzdialene riadiť iné zariadenia. V súčasnej dobe slúži ako univerzálny testovací obvod pre dielčie bloky budúceho návrhu a iných projektov.

5 ZOZNAM POUŽITÝCH LITERÁRNÝCH ZDROJOV

- [1] JANEČEK, Jan a Martin BÍLÝ. *Lokální síte*. 1. vyd. Praha: ČVUT, prosinec 2003. 1.
- [2] IP Addressing and Subnetting for New Users. *IP Addressing and Subnetting for New Users* [online]. 2005, Sep 26, 2005 [cit. 2013-12-04]. Dostupné z: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a00800a67f5.shtml
- [3] Understanding and Troubleshooting DHCP in Catalyst Switch or Enterprise Networks. *Understanding and Troubleshooting DHCP in Catalyst Switch or Enterprise Networks* [online]. 2008 [cit. 2013-12-04]. DOI: ID: 27470. Dostupné z: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_tech_note09186a00800f0804.shtml
- [4] ROVING NETWORKS, Inc. *RN-171-XV-DS: RN-171-XV 802.11 b/g Wireless LAN Module* [online]. 1.04. Los Gatos, CA: Roving Networks, Inc., 2012, 10/29/2012 [cit. 4.12.2013]. Dostupné z: www.rovingnetworks.com
- [5] AND-TECH.PL. *EvB5.1 user's guide* [online]. 1. vyd. Polsko, 2010 [cit. 2013-12-04]. Dostupné z: <http://and-tech.pl/wp-content/uploads/downloads/2013/05/Manual-EvB5.1-v1-eng.pdf>
- [6] ONPA.CZ. *Kit EvB 4.3* [online]. 1. vyd. Česká republika, 2010 [cit. 2013-12-04]. Dostupné z: <http://shop.onpa.cz/?kit-evb-4.3,27>
- [7] ATMEL CORPORATION. *ATmega32A: Microcontroller with 32KBytes In-System Programmable Flash* [online]. 8155A – 06/08. USA: Atmel Corporation, 2008 [cit. 2013-12-04].
- [8] TEXAS INSTRUMENTS. *6-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR WITH AUTO-DIRECTION SENSING AND ±15-kV ESD PROTECTION* [online]. 1. vyd. Dallas, Texas: Texas Instruments Incorporated, 2008 [cit. 2014-05-04]. Dostupné z: <http://pdf1.alldatasheet.com/datasheet-pdf/view/254905/TI/TXB0106.html>
- [9] ADAM OSBORNE. *Introduction to Microcomputers Volume 1: Basic Concepts*. USA: Osborne-McGraw Hill Berkeley California, 1980. ISBN 0-931988-34-9.
- [10] WIKIPEDIA.ORG. *USART: Synchronní / asynchronní sériové rozhraní USART* [online]. 3. vyd. 2013 [cit. 2013-12-04]. Dostupné z: <http://cs.wikipedia.org/wiki/USART>
- [11] Wired Equivalent Privacy (WEP) on Aironet Access Points and Bridges Configuration Example. *Wired Equivalent Privacy (WEP) on Aironet Access Points and Bridges Configuration Example* [online]. 13.2.2007, č. 1 [cit. 2014-05-22]. DOI: 10953. Dostupné z: <http://www.cisco.com/c/en/us/support/docs/wireless->
- [12] WPA Configuration Overview. *WPA Configuration Overview* [online]. 19.10.2009, č. 1 [cit. 2014-05-22]. DOI: 44721. Dostupné z: <http://www.cisco.com/c/en/us/support/docs/wireless/aironet-1100-series/44721-WPAOverview.html>
- [13] Wi-Fi Protected Access 2 (WPA 2) Configuration Example. *Wi-Fi Protected Access 2 (WPA 2) Configuration Example* [online]. 21.1.2008, č. 1 [cit. 2014-05-22]. DOI: 67134. Dostupné z: <http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/67134-wpa2-config.html>
- [14] Wi-Fi Protected Access (WPA) in a Cisco Unified Wireless Network Configuration Example. *Wi-Fi Protected Access (WPA) in a Cisco Unified Wireless Network Configuration Example* [online]. 26.2.2008, č. 1 [cit. 2014-05-22]. DOI: 100708. Dostupné z: <http://www.cisco.com/c/en/us/support/docs/wireless/4400-series-wireless-lan-controllers/100708-wpa-urn-config.html>
- [15] MICROSOFT. *NetworkStream Class: .NET Framework 4.5* [online]. 2014 [cit. 2014-05-15]. Dostupné z: <http://msdn.microsoft.com/en-us/library/vstudio/system.net.sockets.networkstream>

6 ZOZNAM OBRÁZKOV

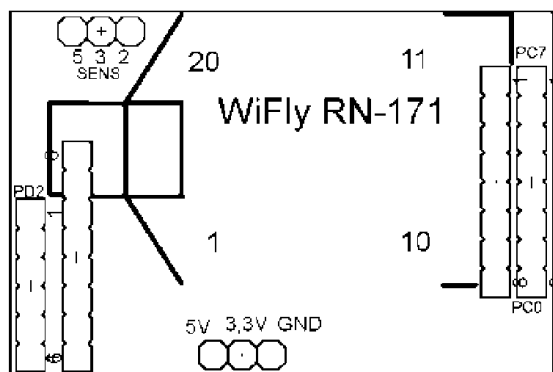
Obr. 2.1: Ukážka topológií sietí.....	7
Obr. 2.2: ISO/OSI model	8
Obr. 2.3: ISO model podľa štandardu IEEE 802	9
Obr. 2.4: Štruktúra rámcov IEEE 802	9
Obr. 2.5: Triedy IP adres	11
Obr. 2.6: Graf závislosti dosahu a rýchlosti pripojenia na použitej technológii	15
Obr. 2.7: Infraštruktúra bezdrôtovej siete.....	16
Obr. 2.8: Rozdiel štruktúry ISO/IEEE 802.11 modelu.....	16
Obr. 2.9: Prístupová metóda IEEE 802.11	17
Obr. 2.10: Rámec formátu MAC	18
Obr. 2.11: Spôsob rozšírenia vysielanej frekvencie	18
Obr. 2.12: Jednotlivé kanáli v pásme 2.4GHz.....	19
Obr. 2.13: Štruktúra rámca PLCP DSSS	19
Obr. 2.14: Dvojstavová (BPSK) a štvorstavová (QPSK) modulácia	20
Obr. 2.15: Štruktúra modulátoru CCK	21
Obr. 2.16: Scrambler $z^{-7} + z^{-4} + 1$	21
Obr. 2.17: Štruktúra modulátoru PBCC	21
Obr. 2.18: Štruktúra rámca IEEE 802.11b.....	22
Obr. 3.1: Vývojová doska EvB 5.1	24
Obr. 3.2: Vzhľad finálnej dosky	27
Obr. 3.3: Schéma vysielania bajtu	29
Obr. 3.4: Schéma prijímania bajtu.....	30
Obr. 3.5: Náhľad na možnosti pripojenia a odpojenia.....	38
Obr. 3.6: Náhľad na záložku TCP-IP console	39
Obr. 3.7: Náklad na záložku I/O config.....	40
Obr. 3.8: Náklad na záložku Monitor	41
Obr. 3.9: Vývojový diagram tlačidla Send	42
Obr. 3.10: Vývojový diagram tlačidla Read config.....	44
Obr. 3.11: Vývojový diagram tlačidla Write config.....	46
Obr. 3.12: Algoritmus zápisu výstupov	48
Obr. 3.13: Algoritmus komunikácie monitorovania dát.....	50
Obr. 3.14: Príklad monitorovania s výpadkom dát.....	51
Obr. 8.1: Servisný popis dosky.....	55
Obr. 8.2: Diery a otvormi v nepájavej maske z vrchnej strany	55
Obr. 8.3: Diery a otvormi v nepájavej maske zo spodnej strany	56
Obr. 8.4: Vodivé cesty a prepoje z vrchnej strany.....	56
Obr. 8.5: Vodivé cesty a prepoje zo spodnej strany	56
Obr. 8.6: Kompletný náhľad na dosku plošného spoja.....	56

7 ZOZNAM PRÍLOH

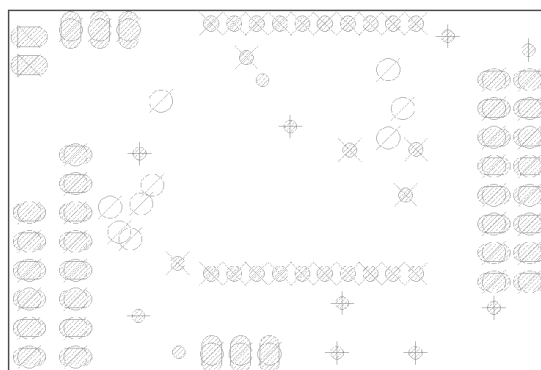
Príloha č. 1: návrh dosky plošných spojov	54
Príloha č. 2: návrh dosky plošných spojov v programe Eagle	Priložené CD
Príloha č. 3: aplikácia pre konfiguráciu a komunikáciu s modulom	Priložené CD
Príloha č. 4: elektronická verzia diplomovej práce	Priložené CD

8 PRÍLOHY

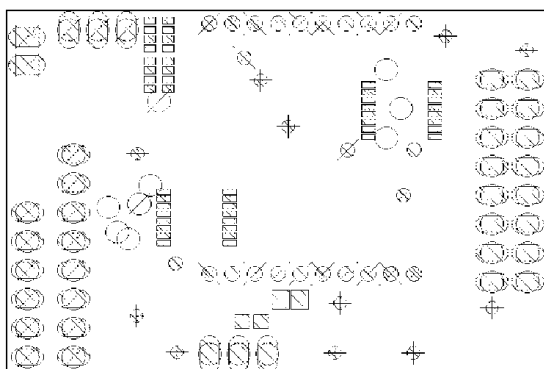
Príloha č. 1: návrh dosky plošných spojov



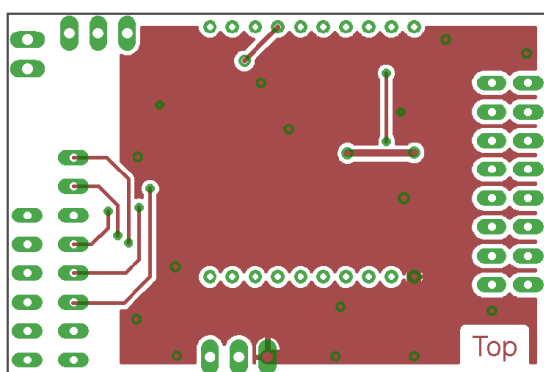
Obr. 8.1: Servisný popis dosky



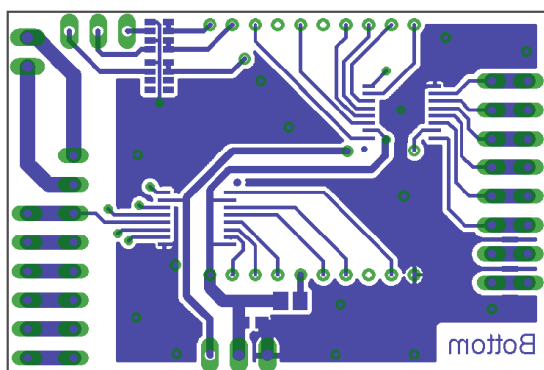
Obr. 8.2: Diery a otvormi v nepájavej maske z vrchnej strany



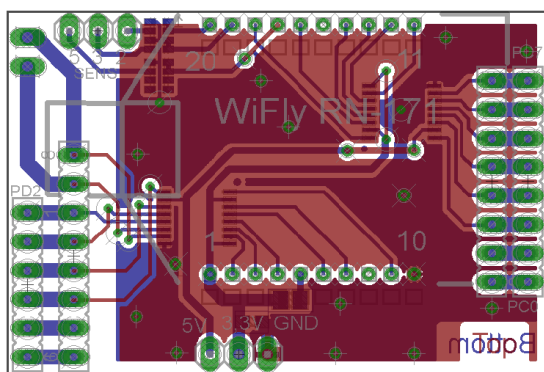
Obr. 8.3: Diery a otvormi v nepájavej maske zo spodnej strany



Obr. 8.4: Vodivé cesty a prepoje z vrchnej strany



Obr. 8.5: Vodivé cesty a prepoje zo spodnej strany



Obr. 8.6: Kompletný náhľad na dosku plošného spoja

Zoznam súčiastok:

Názov:	Popis:	Hodnota:	Puzdro:
C1	kondenzátor	100nF	0602
R1	rezistor	10k	0805
RN1 RN2	pole rezistorov	4x10k	CTS742C083
IC1,IC2	Prevodník tapetových úrovni	TXB0106	TSSOP16
JP1,JP2	vývodový hrebienok typ samec	1X03	rozteč 2,54mm
X1	vývodový hrebienok typ samec	1X02	rozteč 2,54mm
SV1	vývodový hrebienok typ samec	1x6	rozteč 2,54mm
SV2	vývodový hrebienok typ samec	1x8	rozteč 2,54mm
SV4	vývodový hrebienok typ samica	1x8	rozteč 2,54mm
RN-XV:1-10	vývodový hrebienok typ samica	1x10	Rozteč 2mm
RN-XV:11-20	vývodový hrebienok typ samica	1x11	Rozteč 2mm