



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF RADIO ELECTRONICS

## REALIZACE ZAŘÍZENÍ PRO KOMUNIKACI CAR2X A CAR2CAR

REALIZATION OF THE CAR2X AND CAR2CAR COMMUNICATION DEVICE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

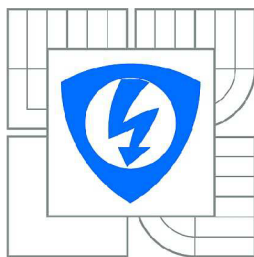
Bc. MILAN ŠTOHANZL

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN PROKOPEC, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav radioelektroniky

# Diplomová práce

magisterský navazující studijní obor  
**Elektronika a sdělovací technika**

**Student:** Bc. Milan Štohanzl

**ID:** 72797

**Ročník:** 2

**Akademický rok:** 2010/2011

## NÁZEV TÉMATU:

**Realizace zařízení pro komunikaci Car2X a Car2Car**

## POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti hardwarové realizace komunikačního zařízení Car2X a Car2Car. Propojení komunikačního zařízení na straně infrastruktury vytvořte pomocí ethernetu, pro bezdrátovou komunikaci mezi zařízeními porovnejte dostupné standardy bezdrátové komunikace a na základě výsledků rozboru vyberte vhodný standard.

Navrhněte hardwarové zapojení komunikačního zařízení Car2X a Car2Car pro obousměrnou komunikaci. Část zařízení ve vozidle bude zobrazovat data na display, část na straně infrastruktury bude data předávat na ethernet. Připravte desky plošných spojů a vývojové diagramy řídicího software.

Realizujte hardwarovou část zařízení, naprogramujte mikrokontrolér a sestavte podrobnou technickou dokumentaci.

## DOPORUČENÁ LITERATURA:

[1] ZigBee Alliance. Dostupné na WWW: <http://www.zigbee.org/en/index.asp>

[2] PRASAD, R., DENEIRE, L. From WPAN's to Personal Networks: Technologies and Applications. Boston: Artech House, 2006. ISBN: 1-58053-826-6

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 20.5.2011

**Vedoucí práce:** Ing. Jan Prokopec, Ph.D.

**prof. Dr. Ing. Zbyněk Raida**  
Předseda oborové rady

## **ABSTRAKT**

Tato práce zkoumá možnosti Car2Car a Car2X komunikace. Obsahuje přehled vlastností systému, typů přenášených zpráv, apod. Zobrazuje architekturu systému a zabývá se i technickými předpoklady a omezeními systému. S ohledem na fakt, že práce vzniká ještě v době, kdy vývoj tohoto systému není ukončen, práce neobsahuje detaily, které by umožňovaly hlubší technickou představu o problematice komunikace mezi pohybujícími se vozidly. Práce se dále zabývá možnostmi realizace zařízení, komunikujícího pomocí podobného standardu. Jako nejvhodnější standard byl zvolen IEEE 802.11a. Mobilního zařízení je vytvořena pomocí vývojového kitu Mini 2440 a komunikace je realizována pomocí WiFi modulu OWS451i, který pracuje jako AT modem. Mini 2440 i strana serveru jsou založeny na operačním systému Linux.

## **KLÍČOVÁ SLOVA**

Car2Car, Car2X, síť, komunikační systém, komunikace mezi vozidly, IEEE 802.11p, IEEE 802.11a, TCP/IP, UDP/IP.

## **ABSTRACT**

This work explores possibilities of Car2Car and Car2X communication. It contains survey of system properties, types of transmission messages, etc. It represents architecture of the system and deals with technical expectations and limitations of system. In the light of the fact that this work is created in the time, when development of this system hasn't finished yet, the work doesn't contain details which would allow deeper technical view about area of vehicular communication. The work also deals with the possibility of realization of device, communicating with a similar standard. Like the most suitable standard was chosen an IEEE 802.11a. Mobile unit has been realized by single board computer Mini 2440 and communication has been realized by WiFi module OWS451i, which works as AT modem. Mini 2440 and infrastructure server are based on Linux operation system.

## **KEYWORDS**

Car2Car, Car2X, network, communication system, vehicular communication, IEEE 802.11p, IEEE 802.11a, TCP/IP, UDP/IP.

ŠTOHANZL, M. *Realizace zařízení pro komunikaci Car2X a Car2Car*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 48 s., 24 s. příloh. Vedoucí diplomové práce Ing. Jan Prokopec, Ph.D.



## PROHLÁŠENÍ

Jako autor diplomové práce na téma „Realizace zařízení pro komunikaci Car2X a Car2Car“ dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Janu Prokopcovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne .....

.....

(podpis autora)

## Obsah:

1	Úvod.....	4
2	Systém Car2Car a Car2X.....	5
2.1	Typy přenášených zpráv .....	5
2.1.1	Bezpečnost provozu na pozemních komunikacích.....	5
2.1.2	Efektivita provozu na pozemních komunikacích.....	5
2.1.3	Infotainment.....	7
2.2	Systémové předpoklady a meze.....	8
2.2.1	Technické předpoklady a omezení .....	8
2.3	Struktura systému .....	11
2.3.1	Přehled systému .....	11
2.3.2	Architektura jednotlivých komponent .....	13
2.3.3	Základní komunikační principy .....	15
2.3.4	Vrstvy architektury a související protokoly .....	15
3	Simulace provozu Car2Car nebo Car2X .....	17
3.1	Rozdíl mezi IEEE 802.11p a 802.11(a, n) .....	17
3.1.1	PHY vrstva IEEE 802.11p.....	17
3.2	IEEE 802.11a.....	18
3.3	Dostupné WiFi moduly.....	19
3.3.1	Popis WiFi modulu cB-OWL221a .....	19
3.3.2	Popis WiFi modulu OWS451i .....	22
3.3.3	Popis sběrnice SPI .....	29
3.3.4	Popis sběrnice UART .....	30
4	TCP/IP nebo UDP/IP z hlediska programování v C/C++ (Unixový OS).....	31
4.1	Protokol IP .....	31
4.2	Protokol TCP .....	31
4.3	Protokol UDP.....	31
4.4	Socket.....	31
4.5	Model komunikace klient – server.....	32
4.5.1	Program TCP server.....	32
4.5.2	Program TCP klient .....	32
5	Přístup k HW prostředkům z hlediska programování v C/C++ (Unixový OS).....	37
5.1	Funkce open:.....	37
5.2	Funkce close: .....	37
5.3	Funkce ioctl: .....	37
5.4	Nastavení parametrů sériové linky: .....	38
5.5	Funkce read a write:.....	38
6	Komunikační program pro Mini 2440 a OWS451 .....	39
6.1	Příprava WiFi modulu: .....	39
6.2	Programy pro stranu vývojového kitu: .....	40
6.2.1	Program pro skenování dostupných sítí:.....	40
6.2.2	Popis programu pro komunikaci na straně vývojového kitu: .....	40
6.3	Programy pro stranu infrastruktury: .....	42
6.3.1	Program TCP_server: .....	42
6.3.2	Program UDP_sender: .....	42
6.3.3	Program UDP_mess:.....	42
7	Závěr .....	43
8	Seznam zkratk.....	45
9	Literatura.....	47

## Seznam obrázků:

<b>Obr. 1.</b> Příklad použití systému. [2].....	4
<b>Obr. 2.</b> Příklad využití systému při navigaci vozidel. [1] .....	6
<b>Obr. 3.</b> Požadované frekvence v Evropě. [2] .....	9
<b>Obr. 4.</b> Architektura systému. [2] .....	11
<b>Obr. 5.</b> Model systému. [2] .....	12
<b>Obr. 6.</b> RSU jako rozšíření komunikační řady. [2] .....	14
<b>Obr. 7.</b> RSU jako bod infrastruktury. [2] .....	14
<b>Obr. 8.</b> RSU jako brána k internetu. [2] .....	14
<b>Obr. 9.</b> Architektura C2C komunikační vrstvy. [2] .....	16
<b>Obr. 10.</b> Blokové schéma modulu cB-OWL221a. [6] .....	20
<b>Obr. 11.</b> Rozmístění konektorů na cB-OWL221a TOP. [6] .....	20
<b>Obr. 12.</b> Softwarové schéma modulu OWL221a. [6] .....	22
<b>Obr. 13.</b> WiFi modul cB-OWS451i (pohled shora). [13] .....	23
<b>Obr. 14.</b> Blokové schéma modulu cB-OWS451i. [13] .....	23
<b>Obr. 15.</b> Blokové schéma pro spojení přes WiFi modul.....	26
<b>Obr. 16.</b> Rozmístění periférií vývojového kitu Mini 2440. [8].....	27
<b>Obr. 17.</b> Konektor CON4. [8] .....	28
<b>Obr. 18.</b> Konektor CON2. [8] .....	29
<b>Obr. 19.</b> SPI s jedním slave zařízením. [9] .....	29
<b>Obr. 20.</b> SPI s více slave zařízením. [9].....	30
<b>Obr. 21.</b> Formát přenosu na sběrnici UART.....	30
<b>Obr. 22.</b> Obsluha klienta. ....	33
<b>Obr. 23.</b> TCP server. ....	34
<b>Obr. 24.</b> TCP klient. ....	35
<b>Obr. 25.</b> Komunikace se serverem.....	36
<b>Obr. 26.</b> Sken dostupných WiFi sítí.....	41
<b>Obr. 27.</b> Výsledné zapojení systému.....	42

## Seznam tabulek:

<b>Tabulka 1:</b> Modulační schémata IEEE 802.11a. ....	17
<b>Tabulka 2:</b> Výkonové třídy IEEE 802.11p. [12] .....	18
<b>Tabulka 3:</b> Porovnání fyzických vrstev IEEE 802.11a a IEEE 802.11p. [12] .....	18
<b>Tabulka 4:</b> Povolný výkon v pásmu b, c, d. [5].....	19
<b>Tabulka 5:</b> Odběr proudu modulu OWL221a v závislosti na režimu činnosti. [6].....	21
<b>Tabulka 6:</b> Informace o radiových prostředcích. [6].....	21
<b>Tabulka 7:</b> Vývody konektoru J8. [13] .....	24
<b>Tabulka 8:</b> Informace o radiových prostředcích. [13].....	24
<b>Tabulka 9:</b> SPI piny.....	29

## Seznam příloh:

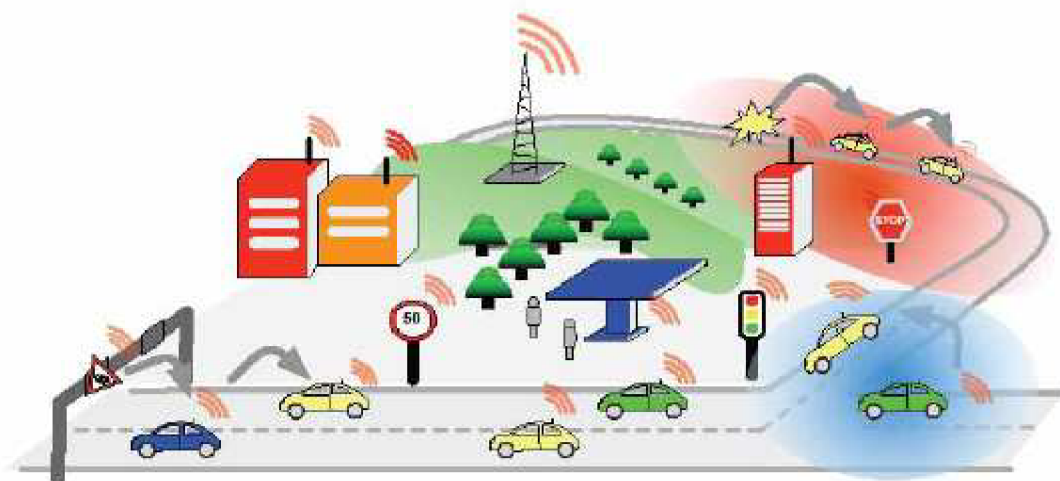
- Příloha 1 – Zkušenosti a poznámky k práci s kity Mini 2440 a 6410.
- Příloha 2 – AT příkazy pro OWS451

# 1 Úvod

CAR 2 CAR Communication Consortium je Evropská nezisková průmyslová organizace podporovaná evropskými výrobci automobilů. Cílem C2C-CC je další zvýšení bezpečnosti silničního provozu a zvýšení účinnosti inteligentních dopravních systémů pomocí systémů podporujících Vehicle-2-Roadside Communications. C2C-CC podporuje vytvoření evropské normy pro budoucí komunikaci mezi vozidly všech značek. C2C-CC pracuje ve spolupráci s evropskými a mezinárodními organizacemi pro normalizaci, zejména ETSI TC ITS.

Komunikační technologie pro spolupráci ITS a C2C komunikace je odvozena od standardu IEEE 802.11, také známý jako WiFi síť. Frekvence přidělená tomuto systému v Evropě je 5,9 GHz, tedy shodně jako u podobného systému v USA. Síť typu ad-hoc se vytváří automaticky, jsou-li dvě nebo více vozidel v dosahu radiového spojení. Předávají se informace o poloze, rychlosti a směru jednotlivých vozidel. Po příjmu informací dojde k jejich zpracování a porovnání s vlastními údaji a v případě potřeby jsou vyslány výstražné informace.

Rozsah jednoho LAN spojení je omezen na několik set metrů, každé vozidlo slouží zároveň jako router a umožňuje posílání zpráv přes multi-hop do dalších vozidel a ITS stanic. Směrovací algoritmus je založen na pozici vozidla a je schopen zvládnout rychlé změny topologie sítě ad-hoc. [1] [3]



**Obr. 1.** Příklad použití systému. [2]

Na **Obr. 1** je vidět:

- vozidla, která využívají systém pro příjem výstražných zpráv a doporučení pro volbu trasy,
- provozovatelé silniční dopravy, kteří dostávají údaje o provozu, díky tomu mohou řídit provoz efektivnějším způsobem,
- terminály internetových služeb, ke kterým je možné se připojit například u čerpacích stanic.

## 2 Systém Car2Car a Car2X

### 2.1 Typy přenášených zpráv

#### 2.1.1 Bezpečnost provozu na pozemních komunikacích

##### Dopředné upozornění na nebezpečí kolize:

Ve všech regionech světa dochází velmi často k nárazům vozidel do zadní části jiného vozidla. Tyto nehody jsou způsobeny rozptýlením řidiče a náhlým brzděním vozidla před ním. Při normální jízdě vozidla vybavená systémem C2C sdílí (anonymně) informace, jako je pozice, rychlost a směr. Aby bylo možné předpovědět kolizi vozidel, sleduje vozidlo chování vlastního řidiče a chování všech vozidel v okolí. V případě, že vozidlo detekuje nebezpečí kolize, upozorní řidiče vizuálním, akustickým nebo hmatovým signálem. To v mnoha případech poskytne řidiči dostatek času, aby stihl hrozící kolizi zabránit. [2]

##### Požadavky:

- vozidla mezi sebou sdílí informace na vzdálenost přibližně 20 až 200 metrů,
- relativně přesné polohy vozidel,
- předávané informace jsou důvěryhodné,
- proniknutí systému na celý trh a jeho využití ve všech vozidlech.

##### Dopředné upozornění před kolizí:

Systém pracuje stejně jako předchozí a v případě, že vyhodnotí kolizi jako nevyhnutelnou, využije předané informace o vozidle (poloha vozidla, velikost a typ vozidla) k efektivnímu nastavení bezpečnostních prvků, jako například nastavení airbagů, bezpečnostních pásů apod. [2]

##### Požadavky:

- vozidla mezi sebou sdílí informace na vzdálenost přibližně 20 až 100 metrů,
- relativně přesné polohy vozidel,
- předávané informace jsou důvěryhodné,
- proniknutí systému na celý trh a jeho využití ve všech vozidlech,
- rychlé spojení mezi vozidly, u kterých se předpokládá nevyhnutelná kolize.

##### Upozornění na nebezpečné místo:

Vozidla mezi sebou sdílí informace, které se týkají nebezpečných míst na vozovce, jako například kluzké vozovky nebo výmoly. Systém ESP může vyhodnotit své nastavení a sdílet informace o stavu vozovky s ostatními vozidly. Vozidla, která obdrží tyto informace je mohou poskytnout řidiči nebo je využít pro nastavení podvozku a elektronických systémů, jako například ESP. [2]

##### Požadavky:

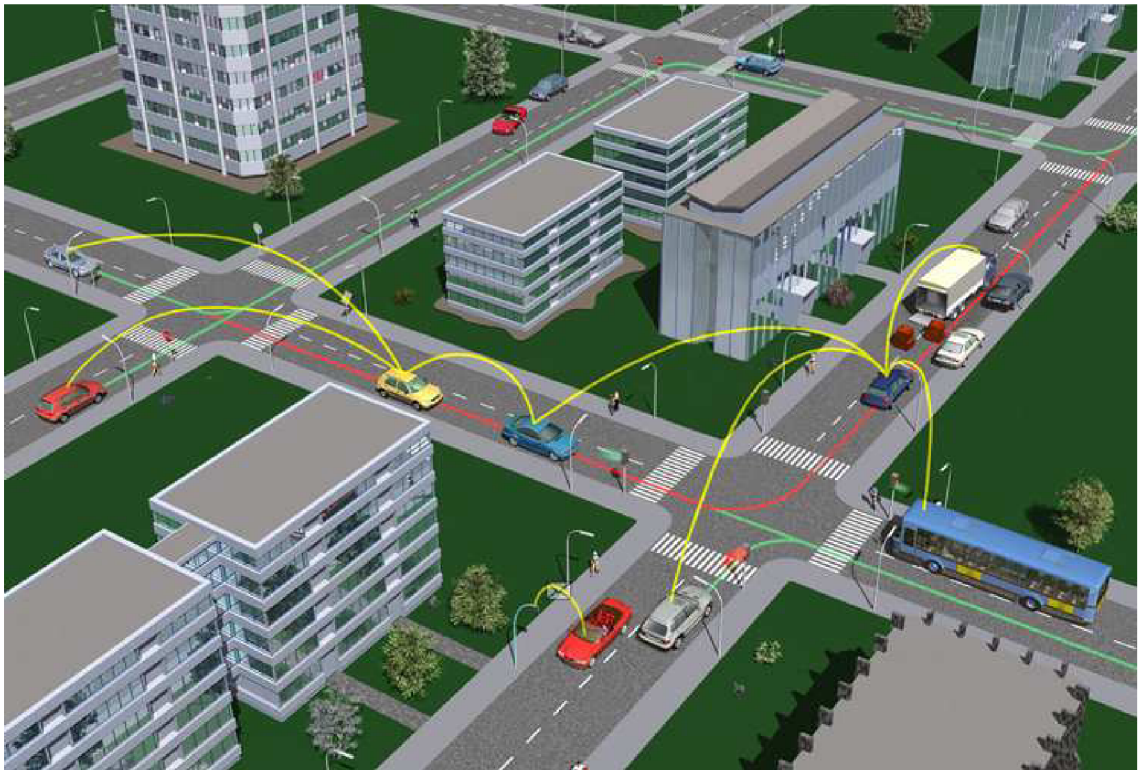
- předávané informace jsou důvěryhodné,
- proniknutí systému na celý trh a jeho využití ve všech vozidlech,
- vozidla sdílí informace s určením oblasti pomocí multi-hop přístupu,
- vozidla jsou schopna sledovat a vyhodnocovat přijaté informace.

#### 2.1.2 Efektivita provozu na pozemních komunikacích

Vozidla vytváří nové informace týkající se provozu nebo sdílí existující informace prostřednictvím C2C. Tím může dojít k větší plynulosti provozu a úsporám nákladů na provoz. [2]

### Navádění a navigace na vyšší úrovni:

Navádění a navigace na vyšší úrovni používá informace shromážděné vlastníkem infrastruktury a poskytuje je řidičům. Informace mohou být také získány od jednotlivých vozidel. [1] [2]



**Obr. 2.** Příklad využití systému při navigaci vozidel. [1]

Hlavním sledovaným jevem je hustota provozu a dopravní zácpy. V případě zácpy, nebo přetížení některého úseku cesty jsou ostatní vozidla naváděna na alternativní trasy.

Požadavky:

- infrastruktura shromažďuje a poskytuje aktuální informace o dopravních zácpách,
- poskytované informace jsou důvěryhodné,
- silniční jednotky jsou schopny přenést informace na projíždějící vozidla.

### Zprávy optimální rychlosti jízdy na zelenou:

Řidičům jsou poskytovány informace o optimální rychlosti, při které projedou následující křižovatkou na zelenou. Vozidla obdrží informace o umístění křižovatkou a o času signálu semaforu (počet sekund do změny ze zelené na červenou). Vozidlo vypočítá optimální rychlost z přijatých informací a z vlastní polohy. Bude-li se vozidlo pohybovat v okolí optimální rychlosti je pravděpodobné, že nebude muset zastavit nebo zpomalit při průjezdu křižovatkou. Tento fakt může zvýšit plynulost dopravy a snížit spotřebu pohonných hmot. [2]

Požadavky:

- řízené křižovatky přenášejí informace o své poloze a časové informace pro každý směr a každý pruh s optimálním načasováním,
- poskytované informace jsou důvěryhodné.

### **C2C pro vozidla integrovaného záchranného systému:**

Vozidla integrovaného záchranného systému využívají C2C systém pro zajištění plynulosti jízdy v provozu. Tato vozidla vyžadují úhybné manévry od ostatních účastníků provozu. Bez námitek z provozu bude provoz buď automaticky upraven, nebo bude řidičům razeno, jak jednat. Hlavním cílem této části systému je zajistit plynulou jízdu vozidlům integrovaného záchranného systému bez vážnějšího dopadu na vlastní provoz. [2]

Požadavky:

- vozidla jsou schopna sdílet informace mezi sebou na vzdálenost odpovídající potřebě úhybného manévru,
- poskytované informace jsou důvěryhodné,
- vozidla se dohodnou na opatřeních s cílem umožnit průjezd vozidlům integrovaného záchranného systému.

### **2.1.3 Infotainment**

Kategorie infotainment slouží k zachycení případů použití systému, které přímo nesouvisí s bezpečností nebo účinností provozu. Většina z těchto případů použití je na základě požadavků majitele vozidla a to za účelem poskytnutí ostatních informací a zábavy. Jiné případy jsou transparentní pro řidiče, ale stále plní funkci jako například zvýšení hospodaření s pohonnými hmotami a diagnostické informace, které jsou pak přístupné autoservisům. [2]

### **Přístup k internetu ve vozidle:**

Přístup k internetu ve vozidle slouží k použití všech běžných druhů služeb založených na IP ve vozidlech. Multi-hop cesta k RSU slouží jako brána k internetu, kde RSU jsou stacionární jednotky rozmístěné podél pozemních komunikací. Tím získává řidič a posádka vozidla výhodu přístupu k informacím dostupným na internetu. [2]

Požadavky:

- vozidla jsou schopna se připojit k silniční jednotce, která poskytuje připojení k internetu,
- vozidla jsou schopna adresovat internetové servery přes silniční jednotky,
- možnost posílání multi-hop zpráv v případě, že vozidlo nemůže přímo komunikovat se silniční jednotkou,
- funkce dynamické údržby cest, která zajišťuje potřebné parametry kvality služeb a resetuje multi-hop trasy v případě potřeby,
- jako alternativa je systémem C2C podporováno připojení pomocí IEEE 802.11a/b/g/n WiFi.

### **Upozornění na místní zajímavosti:**

Upozornění na místní zajímavosti umožňuje místním firmám, turistickým atrakcím a jiným zajímavým místům inzerovat jejich dostupnost do vozidel v okolí. V tomto případě vysílají silniční jednotky informace, jako je například umístění těchto



míst, otevírací doba, cena vstupného a podobné. Velké množství informací je pak adaptivně filtrované podle hustoty provozu a vozidla vybírají vhodné informace, které předávají řidiči. Například při nízké hodnotě ukazatele paliva jsou řidiči prezentovány místní čerpací stanice a ceny pohonných hmot. Výhodou tohoto systému je lepší účinnost reklamy oproti reklamě, která je vysílána pomocí FM rádia stovky kilometrů daleko, nebo surfování po internetu. [2]

Požadavky:

- silniční jednotky jsou schopny vysílat informace do okolních vozidel,
- poskytované informace jsou důvěryhodné.

### **Vzdálená diagnostika:**

Vzdálená diagnostika umožňuje připojení u autoservisů a přenesení diagnostických dat bez fyzického připojení k vozidlu. Podle požadavků zákazníka mohou být přenášeny relevantní diagnostické informace včetně historie technického stavu vozidla, aby umožnily technikům snadno odhalit pravou příčinu problému s technickým stavem vozidla. Systém může také sloužit pro aktualizaci software bez fyzického připojení. Cílem je zkrácení doby, po kterou musí vozidlo setrvat v servisu a snížení nákladů na servis. [2]

Požadavky:

- vozidla jsou schopna vytvořit důvěryhodné a bezpečné spojení s jednotkou autoservisu,
- vozidla jsou schopna identifikovat a diagnostikovat sama sebe a tyto informace předávat jenom oprávněným uživatelům.

## **2.2 Systémové předpoklady a meze**

C2X komunikační systém, je v podstatě distribuovaná a sebe-organizující se síť, která je schopna vyrovnat se s přerušovaným přístupem ke komunikační infrastruktuře. Standardizace komunikačních protokolů, stejně jako u konvenčních komunikačních systémů, zajišťuje interoperabilitu na úrovni sítě. Specifickým aspektem C2X je požadavek na standardizaci i aktivní bezpečnost aplikací. C2X standard musí zahrnovat metody pro detekci a klasifikaci rizik v datech, která si vozidla navzájem vyměňují. [2] Předpoklady a omezení pro úspěšný operační systém mohou být rozděleny do dvou hlavních skupin:

- ekonomické předpoklady a omezení,
- technické předpoklady a omezení.

Tato práce se bude dále zabývat pouze technickými předpoklady a omezeními.

### **2.2.1 Technické předpoklady a omezení**

#### **Anonymita a bezpečnost:**

C2C-CC komunikační systém umožňuje výměnu dat mezi vozidly. Přestože některé aplikace mohou spoléhat na jednoduché typy zpráv z některého výše popsaného scénáře, jiné aplikace a situace obsahují nezbytné údaje, které musejí být předány na jeden nebo více konkrétních uzlů, identifikovatelných pomocí identifikátoru. Na druhé straně musí být zachována dostatečná anonymita vozidla a jeho řidiče alespoň na úrovni srovnatelné s uživateli mobilních telefonů. Jeden z technických přístupů k dosažení anonymity je použití dočasných identifikátorů místo pevných.

Aby systém C2X mohl fungovat v celé Evropě musí vyhovovat právním předpisům na požadavky soukromí v jednotlivých zemích. V některých zemích je soukromí na přání zákazníka povinné nebo je stanoveno zákonem, v jiných zemích zákony vyžadují možnost identifikovat řidiče za každé situace. V důsledku toho musí systém C2X vyhovět různým požadavkům na anonymitu a bezpečnost v celé Evropě.

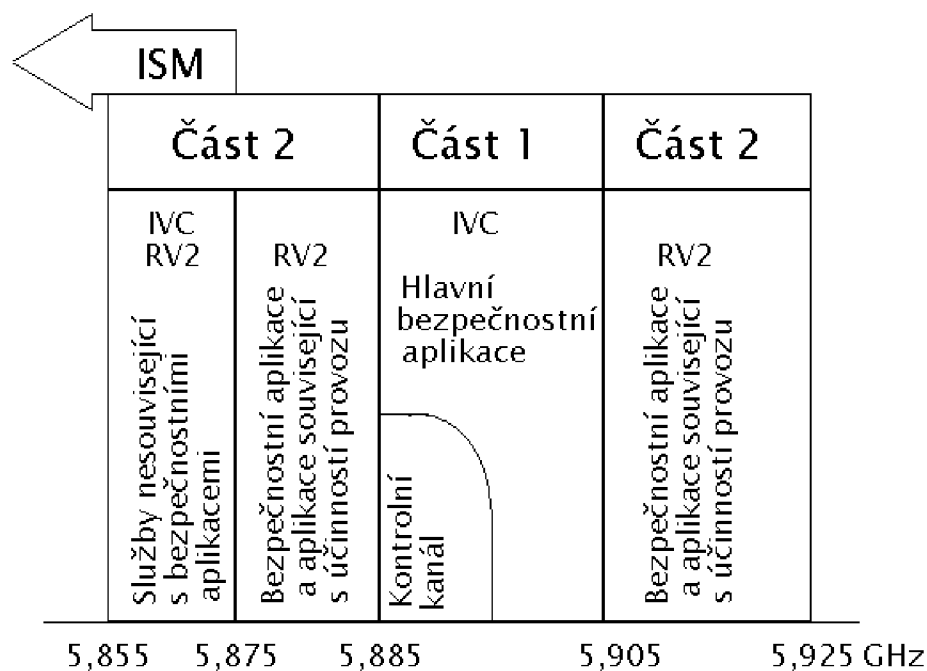
Aby byl systém C2C-CC úspěšný, musí pracovat s vysokou dostupností a spolehlivostí. V případě, že řidič opakovaně obdrží nesprávné údaje, přestane tomuto systému věřit. Nesprávné údaje mohou být způsobeny poruchou systému nebo mohou být vytvořeny uživateli se zlými úmysly. Technicky musí být zajištěno, aby nebylo možné do systému záměrně vyslat nepravdivá data. To se dá splnit přístupem založeným na digitálním podpisu a certifikáty.

Důležitou roli hrají i právní otázky. Například v případě, kdy správná nebo špatná zpráva vyvolá změnu v provozu a ta vede k dopravní nehodě. V současné době není jasné, kdo by v takovém případě nesl odpovědnost. Situace se stane ještě komplikovanější, objeví-li se na trhu více různých aplikací založených na C2C-CC. [2]

### Efektivní ochrana frekvenčního pásma:

Pro aplikace spojené s bezpečností provozu musí být C2X komunikace dostatečně robustní aby byla zajištěna určitá úroveň kvality služeb, například minimální latence a maximální spolehlivost při odesílání, předávání a přijímání zpráv. To nelze zaručit v případě, kdy jiný nezabezpečený systém využívá stejné frekvenční pásmo a zabírá značnou část přidělené šířky frekvenčního pásma. Kritické bezpečnostní zprávy se musejí dostat do všech požadovaných přijímačů v požadovaném čase a to i v případě, že uživatel v jiném automobilu současně stahuje videa z internetu pomocí standardního WLAN přístupového bodu.

Pokud by se systém používal ve veřejně přístupném frekvenčním pásmu, jako je například ISM, nedala by se zaručit dostatečná robustnost a spolehlivost systému. Z tohoto důvodu C2C-CC podporuje efektivní a promyšlené rozdělení chráněného frekvenčního pásma. [2]



Obr. 3. Požadované frekvence v Evropě. [2]

Frekvenční pásmo určené pro kritické bezpečnostní aplikace se nachází v pásmu mezi 5,885 a 5,905 GHz. Alokační proces však stále probíhá.

### **Dynamika systému:**

C2C komunikační systém musí pracovat v situacích s velmi malou hustotou provozu na pozemních komunikacích, ale i v situacích s velkou hustotou provozu, jako jsou dopravní zácpy a hlavní křižovatky. Tyto dvě situace způsobují různé technické problémy. V případě malé hustoty provozu se vozidla často dostávají mimo dosah dalších vozidel a nemohou navázat komunikaci. V hustém provozu může dojít k přetížení systému vlivem omezené šířky frekvenčního pásma.

Dalším aspektem je pronikání systému C2C na trh. Protože vozidla, která budou jako první tímto systémem vybavená, mohou být v provozu velmi dlouho, musí být systém schopný zvládat situace s hustým provozem již od začátku.

Dále je nutné, aby systémy pozdější generace byl kompatibilní i se systémem první generace. [2]

### **Povinné senzory:**

Integrace systému C2C do vozidel není řešena C2C-CC. Odpovědnost za tuto integraci do vozidel mají výrobci vozidel a dodavatelé automobilového průmyslu. Nicméně je určitá skupina podmínek, které musejí být splněny pro integraci systému do vozidla. To zahrnuje dostupnost základní sady senzorů, kde důležitou roli hrají bezpečnostní systémy.

Příkladem mohou být údaje o pozici, které jsou důležité pro komunikaci systému a mnoho dalších aplikací. Pro komunikaci systému postačuje přesnost srovnatelná s dnešní GPS, některé aplikace však mohou vyžadovat přesnost mnohem vyšší. Získání přesnějších údajů o poloze vozidla může být zajištěno systémy ve vozidle nebo mohou být získávány z venkovních serverů. Je nutné poznamenat, že systém GPS není povinný pro C2C-CC systém, lze využít jakýkoliv jiný systém se srovnatelnými vlastnostmi.

C2C-CC bude specifikovat strukturu dat (včetně parametrů, jako je třeba přesnost nebo jejich aktuálnost), které budou muset být v každém vozidle obsahující systém C2C.

Parametry, které jsou pro systém C2C povinné jsou:

- data o pozici,
- rychlost vozidla,
- směr jízdy,
- údaj o zapnutí výstražných světel,
- brzdový výkon (zpomalení vozidla),
- ABS, ESP a ASR senzory,
- senzor deště a údaj o stavu stěračů.

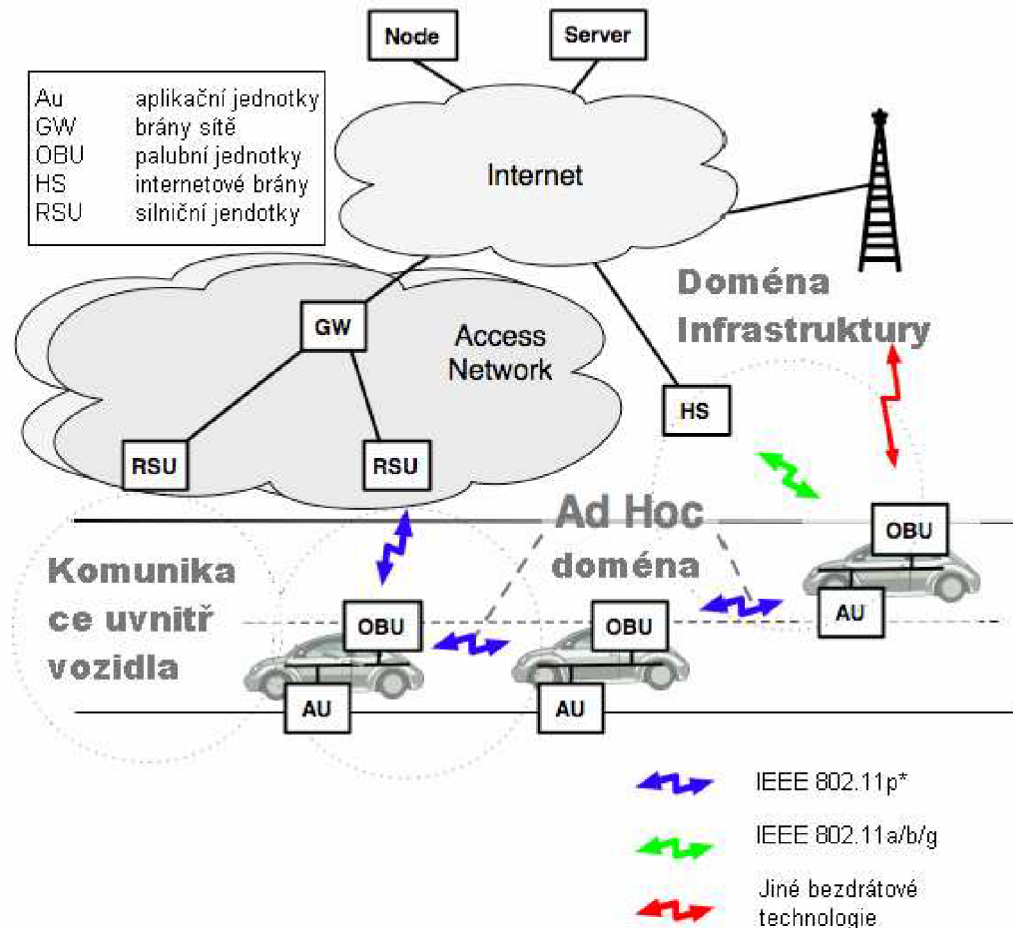
První tři údaje jsou vyžadovány systémem C2C-CC, aby mohl systém fungovat a dále mohou být využity pro celou další řadu aplikací. Ostatní parametry jsou vybrány pro různé aplikace, a to není s konečnou platností rozhodnuto, zda jsou povinné, či nikoliv. Je možné, že s dalšími aplikacemi poroste požadavek na rozšíření o nové senzory.

Senzory ve vozidle pro systém C2C-CC mohou být připojeny přímo k systému, čímž dojde k vytvoření vlastní sběrnice ve vozidle, nebo může být systém C2X připojený přes vhodné rozhraní přímo k elektronické řídicí jednotce vozidla. [2]

## 2.3 Struktura systému

### 2.3.1 Přehled systému

Na Obr. 4 je vidět návrh referenční architektury C2C systému. Jsou zde vidět tři oblasti provozu (domény): komunikace uvnitř vozidla, ad-hoc doména (komunikace mezi vozidly) a páteřní síť (doména infrastruktury).



Obr. 4. Architektura systému. [2]

#### Komunikace uvnitř vozidla:

Komunikace uvnitř vozidla se skládá z palubní jednotky (OBU) a aplikační jednotky (AU). AU je obvykle specializované zařízení, které provádí jednu, nebo sadu aplikací a využívá komunikační schopnosti OBU. AU může být integrovanou součástí vozidla (jedno vozidlo může obsahovat i více AU) a musí být trvale připojené k OBU. Jako AU může být využito i přenosné zařízení, jako například notebook, PDA nebo gamepad, které lze připojit k OBU. OBU a AU může být propojeno pomocí kabelu, nebo může být využito bezdrátové komunikace typu Bluetooth, WUSB nebo UWB. AU a OBU mohou být integrovány i do jedné fyzické jednotky. [2]

#### Ad-hoc domény:

Ad-hoc domény, nebo také VANET síť (komunikace point to point mezi RSU a OBU), se skládají z vozidel vybavených palubními jednotkami OBU a jednotek RSU rozmístěných podél silnice. OBU musí minimálně obsahovat bezdrátové komunikační

zařízení (krátkého dosahu) určené pro bezpečnost silničního provozu. Palubní jednotky OBU tvoří mobilní síť ad-hoc (MANET), které umožňují komunikaci mezi uzly plně distribuovaným způsobem (C2C komunikace). Jinak řečeno, není nutné tyto sítě centrálně koordinovat. Jsou-li palubní jednotky v radiovém dosahu, mohou spolu komunikovat přímo. V opačném případě může být využito směrových protokolů k předání zprávy pomocí multi-hop přes jiné OBU nebo místní RSU. Úkolem RSU je zlepšit bezpečnost provozu na pozemních komunikacích tím, že přijímá a odesílá zprávy v ad-hoc systému a tím zvyšuje radiový dosah. OBU a RSU tvoří statické i mobilní přenosové uzly. RSU navíc mohou být propojeny s infrastrukturou a internetem a tím poskytovat připojení OBU k internetu.

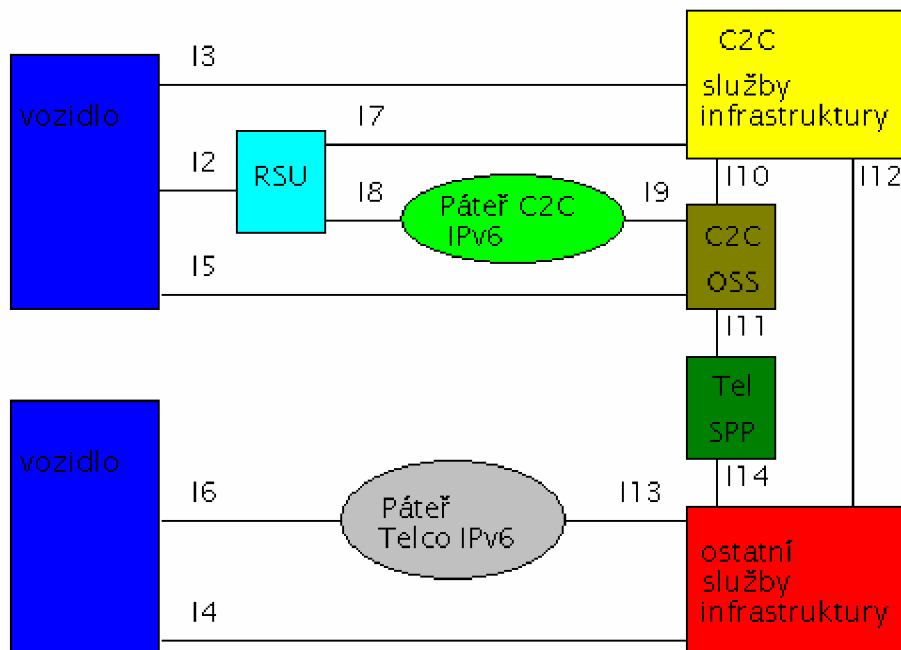
V ad-hoc systému mohou dvě RSU sdílet informace buď přímo (jsou-li v přímém radiovém kontaktu) nebo prostřednictvím OBU pomocí multi-hop přístupu. [2]

### Páteřní síť:

OBU mohou být vybaveny i alternativními bezdrátovými systémy, které nesouvisí s bezpečnostními systémy a slouží například k připojení k internetu prostřednictvím veřejných a komerčních serverů.

Zatímco RSU (i s přístupem na internet) budou rozmisťovány podél pozemních komunikací v souladu s C2C a budou provozovány majitelem či správcem komunikací, alternativní bezdrátové systémy (HS) mohou být vystavěny v méně kontrolovaném prostředí. OBU jednotky mohou být navíc vybaveny přijímači dalších komerčních systémů jako například GSM, GPRS, UMTS, HSDPA, WIMAX, 4G apod. a to vždy pro systémy, nesouvisící s bezpečnostními aplikacemi.

Doména infrastruktury je spojena s infrastrukturou pomocí certifikace PKI. Certifikační autorita (CA) je subjekt, vydávající digitální certifikáty pro OBU a RSU jednotky. Tyto certifikáty jsou používány v komunikaci mezi uzly k zabezpečení a identifikaci uzlu. [2]



Obr. 5. Model systému. [2]

- Ad-hoc komunikace mezi vozidly je zastoupena rozhraním I1, komunikace s jednotkami pozemní komunikace je zastoupena rozhraním I2.
- Přístup ke specializovaným C2C službám je zastoupen pomocí rozhraní I3, zatímco přístup k ostatním komerčním službám je zastoupen pomocí I4. Když je stejná C2C služba přístupná přes RSU je využito rozhraní I7.
- Služby C2C OSS jsou dostupné pomocí rozhraní I5. C2C OSS zahrnuje všechny služby systému souvisejících s provozem systému jako např. ověřování, autorizace, management registrů a databází, apod. Používá-li vozidlo přístup ke komerčním telekomunikačním systémům, přistupuje k nim přes I6 a I13, nebo I8 a I9.
- Systémy C2C OSS a Telco SPP budou muset spolupracovat přes rozhraní I11. Podobně body I10, I12 a I14 slouží pro vzájemnou spolupráci systémů.

Komunikace přes I1 bude s největší pravděpodobností založena na standardu IEEE 802.11p upraveného pro Evropské podmínky (802.11p\*). Toto rozhraní podporuje všechny očekávané komunikační postupy.

Komunikace na I2 až I6 bude s největší pravděpodobností založena na standardu IEEE 802.11a/b/g/n. [2]

### 2.3.2 Architektura jednotlivých komponent

#### AU - Aplikační jednotky:

AU jsou zařízení umístěná uvnitř vozidla, které využívá komunikační schopnosti OBU. AU mohou být vytvořeny zvláště pro bezpečnostní aplikace, pro navigační aplikace a pro přenosové aplikace sloužící například na propojení PDA k internetu. Jednotky AU mohou být zabudovány přímo do vozidla a zajišťovat alespoň minimální požadované aplikace (většinou bezpečnostní), nebo mohou být vyjímatelné z vozidla a zajišťovat další aplikace. Tyto přenosné aplikační jednotky by měly být schopny automatické konfigurace k OBU po připojení ve vozidle. Stejně tak jedna jednotka OBU musí být schopna komunikovat s několika připojenými jednotkami AU. AU a OBU může být fyzicky vytvořeno i jako jediné zařízení.

Protože některé z přijatých zpráv může vyžadovat více AU počítá se s tím, že jedna AU bude plnit funkci správce dat.

C2C-CC předpokládá využití IPv6 protokolu pro připojení AU jednotek, ale z důvodu zpětné kompatibility se počítá i s implementací IPv4. [2]

#### OBU - palubní jednotky:

OBU jednotky slouží ke komunikaci mezi vozidly (C2C) a ke komunikaci mezi vozidlem a infrastrukturou (C2X). Každé vozidlo by mělo obsahovat jednu palubní jednotku OBU, která umožňuje připojení AU jednotek a předává jejich údaje prostřednictvím radiového spojení ostatním dostupným OBU jednotkám v ad-hoc doméně. Toto radiové spojení je založeno na standardu IEEE 802.11p\* a slouží k předávání dat souvisejících s bezpečnostními aplikacemi. Palubní jednotky mohou být vybaveny i jinými radiovými rozhraními, která umožní v ad-hoc doméně předávání dat, která nesouvisejí s bezpečnostními aplikacemi. Zde se počítá se standardem IEEE 802.11 a/b/g/n. [2]

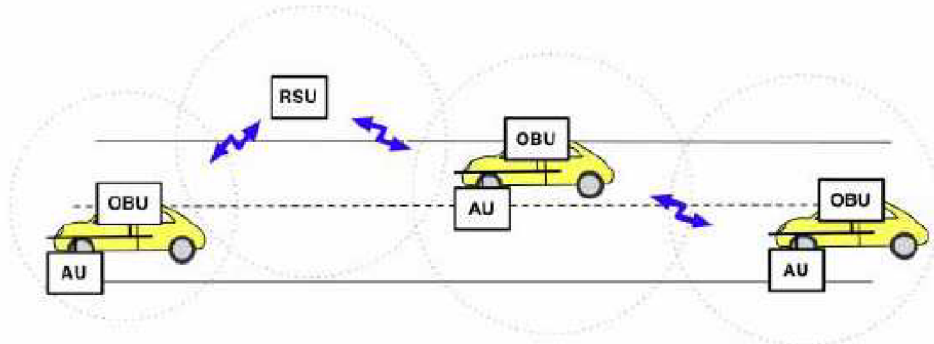
#### RSU – silniční jednotky:

RSU jsou stacionární jednotky rozmístěné podél pozemních komunikací (silnice nižších tříd a dálnice), nebo na místech jako například čerpačí stanice, parkoviště,

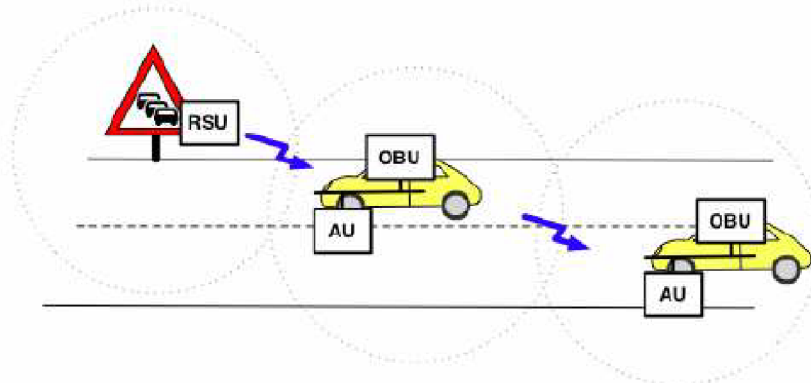
restaurace, apod. RSU jednotky musejí být vybaveny síťovým rozhraním pro komunikaci dle IEEE 802.11p\*. Dále mohou být vybaveny rozhraním pro komunikaci v rámci sítě infrastruktury. [2]

Hlavní funkce RSU:

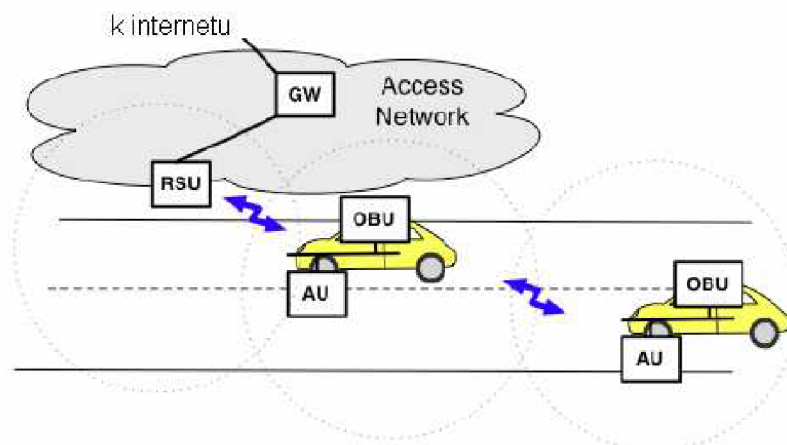
- Rozšíření komunikační řady v ad-hoc doméně prostřednictvím re-distribuce informace mezi OBU. OBU komunikují v rámci dosahu RSU jak je vidět na **Obr. 6**.
- Komunikace C2X, kdy RSU tvoří bod infrastruktury a předává bezpečnostní informace k dostupným OBU jednotkám (viz **Obr. 7**).
- Brána k internetu (viz **Obr. 8**).



**Obr. 6.** RSU jako rozšíření komunikační řady. [2]



**Obr. 7.** RSU jako bod infrastruktury. [2]



**Obr. 8.** RSU jako brána k internetu. [2]

### 2.3.3 Základní komunikační principy

Komunikace s krátkým dosahem C2C je založena na dvou hlavních komunikačních principech:

- Poskytuje prostorové a časové šíření informací mezi vozidly, které slouží hlavně pro bezpečnostní aplikace, ale nejen pro ně.
- C2C komunikační systém umožňuje doručení zprávy pomocí běžné (packet-switched, tzn. s přepínáním paketů) sítě v bezdrátovém prostředí mobilních uzlů a nabízí podobné typy komunikace jako unicast, multicast, anycast, a broadcast přizpůsobené automobilovému prostředí. Unicast je zasílání zpráv individuálnímu příjemci, multicast je zasílání zpráv skupině příjemců, anycast je zasílání zpráv vybraným příjemcům a broadcast je zasílání zpráv všem dostupným příjemcům.

Zatímco konvenční komunikace jsou typicky sender-centric (centrální vysílač), C2C komunikační systém rozlišuje mezi sender-centric a receiver-centric (centrální příjemce) šíření informací.

V receiver-centric šíření zdrojový uzel detekuje nebezpečí svými vlastními senzory a distribuuje informace do svého okolí. Příjemci sloučí informace se svými vlastními informacemi a distribuuje je dále. Prostorové a časové šíření je řízeno uzlem, který je příjemcem a působí jako forwarder. Na vstupu určuje důležitost informací pro své sousedy a rozhoduje o jejich přeposlání dále.

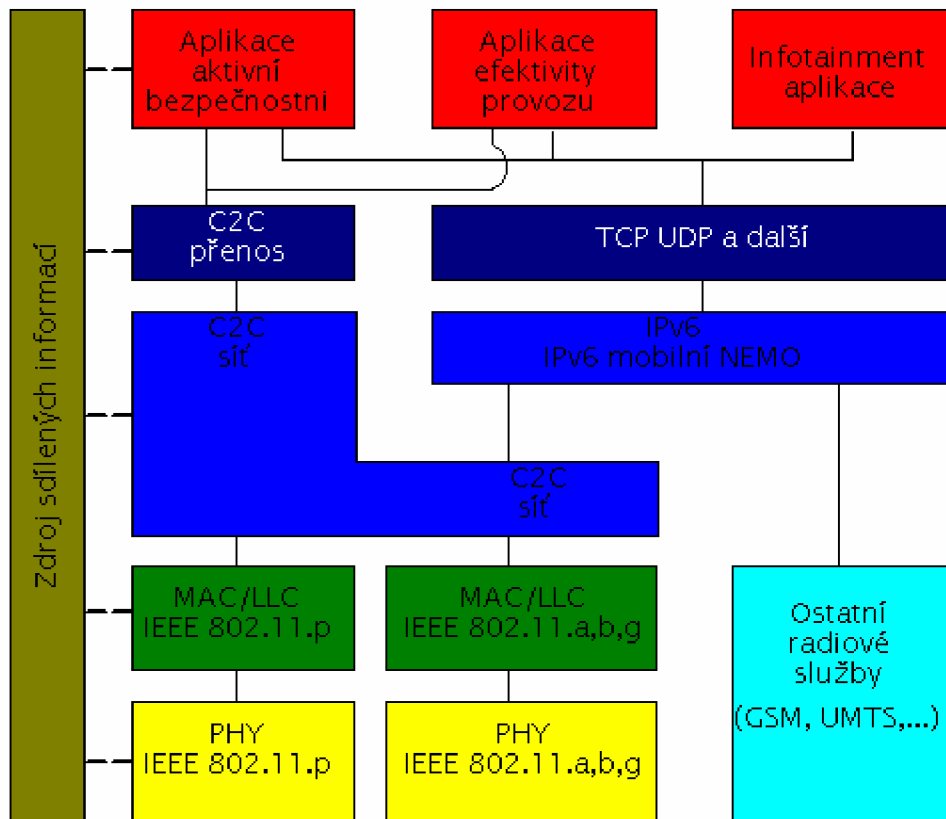
V sender-centric šíření zdrojový uzel definuje zeměpisné oblasti a předává informace do svého okolí. Příjemce rozhoduje o odeslání přijaté zprávy na základě toho, zda se nalézá v dané zeměpisné oblasti nebo ne. [2]

### 2.3.4 Vrstvy architektury a související protokoly

Na **Obr. 9** je zobrazena architektura C2C komunikační vrstvy. Červeně označené bloky (aplikace aktivní bezpečnosti, aplikace efektivity provozu a infotainment aplikace) představují protokoly aplikační vrstvy pro daný typ služeb. Tmavomodře označený blok C2C přenos reprezentuje transportní vrstvu a světle modře označený blok C2C síť představuje síťovou vrstvu.

C2C rozlišuje mezi třemi základními typy bezdrátových technologií: IEEE 802.11p\* bezdrátové technologie, konvenční bezdrátové technologie LAN založené na IEEE 802.11a/b/g/n a další radiové technologie, jako například GSM, UMTS, GPRS apod. Nejvyšší radiovou vrstvou jsou MAC a PHY konkrétní bezdrátové technologie, síťová vrstva umožňuje bezdrátovou multi-hop komunikaci založenou na geografickém adresování a směrování. Také vykonává funkce specifické pro automobilový provoz. [2]





**Obr. 9.** Architektura C2C komunikační vrstvy. [2]

Jak je vidět, aplikace nesouvisející s bezpečnostními informacemi používají tradiční protokoly TCP a UDP přes IPv6 a bezdrátový přístup multi-hop komunikace ke komunikaci s ostatními aplikacemi v dopravních prostředcích nebo se silničními jednotkami a internetovými uzly. Tyto aplikace mohou C2C komunikační systém obejít pomocí IEEE 802.11a/b/g při přímé komunikaci s WiFi hot-spotem. [2]

### 3 Simulace provozu Car2Car nebo Car2X

S ohledem na to, že se na trh doposud nedostaly žádné komunikační moduly pro daná frekvenční pásma (viz **Obr. 3**), bude se zbytek práce zabývat pouze možností komunikace podobného zařízení. Standardu IEEE 802.11p\* je nejbližší standard IEEE 802.11a. Oba dva standardy jsou určeny pro pásmo v okolí 5 GHz a používají OFDM pro přenos dat i záhlaví.

Vlastní komunikace může být realizována pomocí TCP/IP nebo UDP/IP přenosu dat ze zařízení simulujícího vozidlo do zařízení simulujícího infrastrukturu. Na straně infrastruktury může být TCP server očekávající spojení od mobilního zařízení, nebo UDP server očekávající data od mobilního zařízení. Stejně tak zde mohou být klientské aplikace (TCP nebo UDP), které mohou navazovat spojení na server umístěný v mobilním zařízení.

#### 3.1 Rozdíl mezi IEEE 802.11p a 802.11(a, n)

IEEE 802.11p je založen na OFDM modulaci na rádiovém rozhraní stejně jako IEEE 802.11a. IEEE 802.11p obsahuje drobné úpravy MAC a PHY, aby spojení bylo sestaveno rychleji a bylo stabilnější pro pohybující se vozidla.

IEEE 802.11a má definováno osm různých přenosových rychlostí přenosu dat, které jsou uvedeny v **Tabulka 1**.

**Tabulka 1:** Modulační schémata IEEE 802.11a.

Přenosová rychlost [Mbit/s]	Modulace na nosné	Kódový poměr
6	BPSK	1/2
9	BPSK	3/4
12	QPSK	1/2
18	QPSK	3/4
24	16QAM	1/2
36	16QAM	3/4
48	64QAM	1/2
54	64QAM	3/4

System používá 64 nosných, z toho 52 nosných slouží pro přenos dat a 4 slouží jako pilotní nosné. Ostatní nosné neobsahují data. Vzdálenost mezi nosnými je 312,5 kHz. Nominální šířka pásma zabraná signálem je 16,6 MHz. Data jsou vysílána po dobu 4  $\mu$ s, preamble 8  $\mu$ s.

Přenosový burst začíná preambulí obsahující tréninkovou sekvenci, informace o použitém vysílači kvůli frekvenčním korekcím a odhad kanálu. Následující část dat obsahuje informace o modulaci, délce přenášených dat a další doplňkové informace.

##### 3.1.1 PHY vrstva IEEE 802.11p

IEEE 802.11p má definovány tři různé modely fyzické vrstvy s rozdílnou šířkou pásma (20 MHz, 10 MHz, 5 MHz). Různých módů je dosaženo snížením časování (vzorkovací frekvence). IEEE 802.11a používá režim s plným časováním a tedy s šířkou

pásma 20 MHz. IEEE 802.11p používá obvykle režim s polovičním časováním a s šířkou pásma 10 MHz. [12]

Tento režim má vliv na další parametry:

- Šířka pásma: IEEE 802.11p obvykle používá šířku pásma 10 MHz. Díky tomu je signál robustnější proti fadingu.
- Vzdálenost mezi nosnými: IEEE 802.11p používá poloviční vzdálenost mezi nosnými oproti 802.11a.
- Délka symbolů: ta je oproti IEEE 802.11a dvojnásobná. Tím je signál robustnější proti fadingu.

Kromě časování hodin je u IEEE 802.11p změna u ACR (potlačení sousedních kanálů) a SEM (spektrální emisní maska), které jsou definovány přísněji. SEM jsou definovány pro všechny režimy s ohledem na šířku pásma a to vždy 4x, protože jsou definovány čtyři výkonové třídy (A, B, C, D). [12]

**Tabulka 2:** Výkonové třídy IEEE 802.11p. [12]

Výkonová třída	Maximální vysílaný výkon [dBm]
CLASS A	0
CLASS B	10
CLASS C	20
CLASS D	28,8

**Tabulka 3:** Porovnání fyzických vrstev IEEE 802.11a a IEEE 802.11p. [12]

Parametr	IEEE 802.11a	IEEE 802.11p	Rozdíl
Přenosová rychlost:	6, 9, 12, 18, 24, 36, 48, 54	3, 4,5, 6, 9, 12, 18, 24, 27	poloviční
Použité modulace:	BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM	bez změny
Kódový poměr:	1/2, 2/3, 3/4	1/2, 2/3, 3/4	bez změny
Počet subnosných:	52	52	bez změny
Délka trvání symbolu:	4 $\mu$ s	8 $\mu$ s	dvojnásobný
Ochranný interval:	0,8 $\mu$ s	1,6 $\mu$ s	dvojnásobný
Perioda FFT:	3,2 $\mu$ s	6,4 $\mu$ s	dvojnásobný
Délka preambule:	16 $\mu$ s	32 $\mu$ s	dvojnásobný
Vzdálenost mezi subnosnými:	312,5 kHz	156,25 kHz	poloviční

### 3.2 IEEE 802.11a

Na rozdíl od ostatních standardů WiFi (IEEE 802.11b/g) pracuje tento ve frekvenčním pásmu 5,15 až 5,35 GHz a 5,725 až 5,850 GHz. Dostupná šířka frekvenčního pásma (325 MHz). Používá modulaci OFDM, která je oproti starším typům vyspělejší a robustnější. Hlavní výhodou OFDM je v její odolnosti vůči vícecestnému šíření. Toho je dosaženo přidáním tzv. cyklického prefixu. Přenosová rychlost na jeden kanál je 6, 9, 12, 18, 24, 36, 48 nebo 54 Mbit/s.

**Tabulka 4:** Povolený výkon v pásmu b, c, d. [5]

Pásmo	Frekvence [MHz]	Výkon [mW] EIRP	Poznámka
b	5150-5250	200	pouze indoor
c	5250-5350	200	pouze indoor
d	5470-5725	1000	max. 50 mW/MHz

### 3.3 Dostupné WiFi moduly

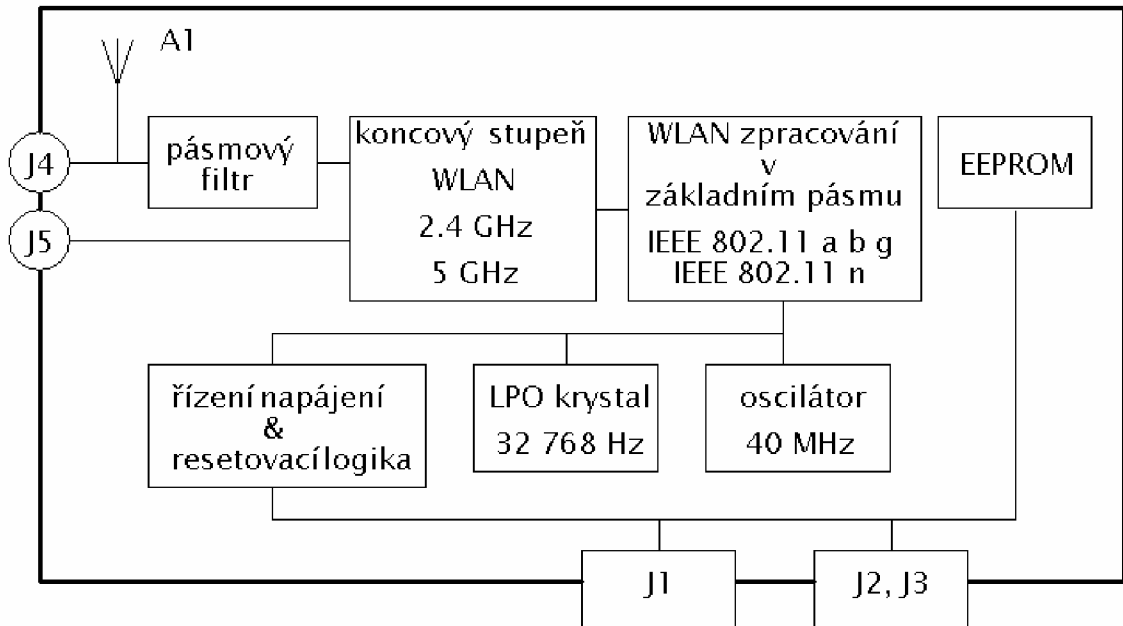
S ohledem na výše uvedené byl pro komunikaci zvolen modul od firmy connectBlue s označením cB-OWL221a, později modul OWS451i.

#### 3.3.1 Popis WiFi modulu cB-OWL221a

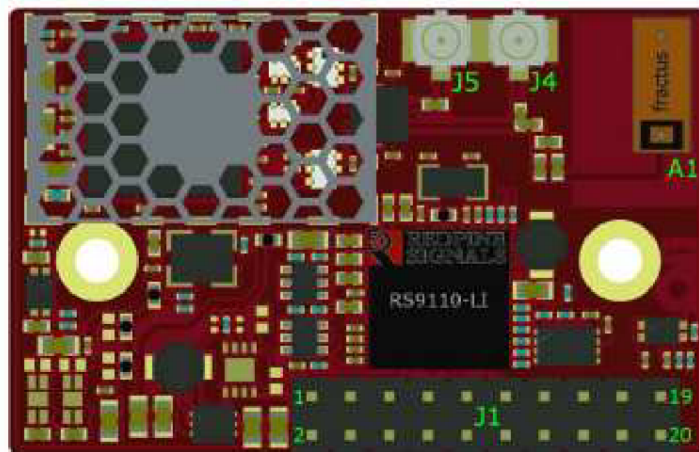
IEEE 802.11a b g n modul označený jako cB-OWL221a je určen k integraci do průmyslových zařízení. Podle výrobce (connBlue) vyniká vysokou energetickou účinností, kompatibilitou, robustností a spolehlivostí. Modul je vytvořen tak, aby jeho integrace do zařízení byla co nejjednodušší. To znamená, že spolu s modulem jsou dodávány ovladače pro vybrané operační systémy (Linux 2.6, Windows XP, Windows CE 6.0 a další). [6]

Klíčové rysy:

- Možnost komunikace v IEEE 802.11a b g nebo IEEE 802.11n.
- Hardwarový akcelerátor WEP, AES a CRC32.
- Podpora WPA a WPA2.
- QoS 802.11e a WMM.
- Rozhraní SPI (existují i jiné varianty s rozhraním SDIO).
- Podpora režimů ad-hoc a komunikace s infrastrukturou.
- Bluetooth co-location s PTA (Packet Traffic Arbitration).
- Certifikace pro provoz v Evropě.
- V souladu s normami EMC.
- Teplotní rozsah od -30 °C do 85 °C (průmyslové použití).
- Komunikace pomocí interní antény (s možností připojení externí antény).
- Podpora diverzifikačního příjmu ze dvou antén.



Obr. 10. Blokové schéma modulu cB-OWL221a. [6]



Obr. 11. Rozmístění konektorů na cB-OWL221a TOP. [6]

Označení pinů:

- J1: 2 x 10 pinová lišta (komunikace a napájení)
- J2 a J3 plošky ze zadní strany slouží pro připojení „board to board“.
- A1: interní anténa
- J4: konektor pro připojení primární externí antény (50 Ω).
- J5: Konektor pro připojení pomocné externí antény (dual diverzity) (50 Ω).

Podpora diverzitního příjmu z interní antény a jedné externí antény, nebo ze dvou externích antén zvyšuje oblast použitelnosti tohoto modulu v různých prostředích, kde může docházet k únikům signálu. S ohledem na to, že tato konstrukce se zabývá pouze simulací provozu C2C nebo C2X ve statických podmínkách (se zařízením nebo AP nebude během komunikace pohybováno) postačí využití interní antény.

**Parametry napájení:**

Napájecí napětí modulu by mělo být minimálně 3,3 V a maximálně 5,5 V. To umožňuje připojit modul k mnoha aplikacím bez nutnosti přizpůsobovat napájecí napětí konkrétní hodnotě.

Hodnota napájecího proudu se mění v závislosti na módu, ve kterém modul právě pracuje. Některé hodnoty jsou uvedeny v následující tabulce. Parametrem pro tyto hodnoty je napájecí napětí s hodnotou 3,6 V. [6]

**Tabulka 5:** Odběr proudu modulu OWL221a v závislosti na režimu činnosti. [6]

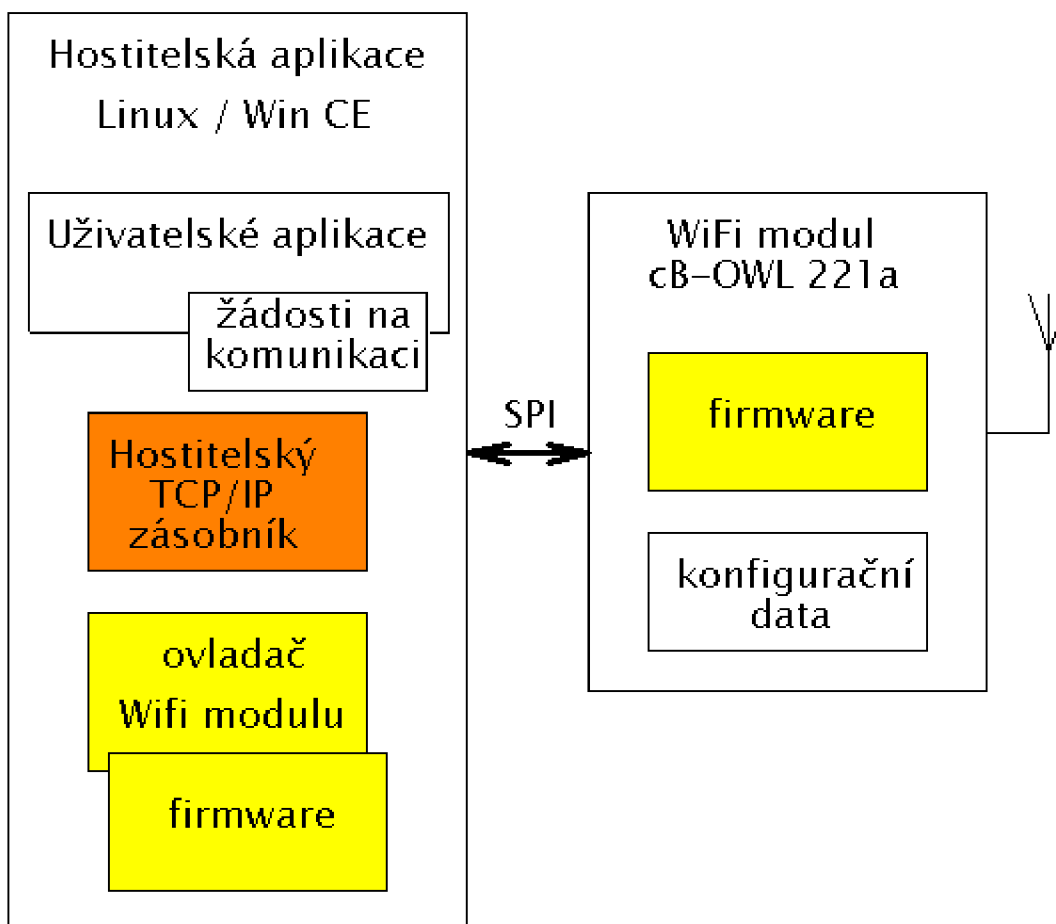
mód	typická hodnota [mA]	maximální hodnota [mA]
nečinný, bez firmware	16	
nečinný, s firmware, úsporný	2	
nečinný, s firmware	120	
nečinný, navázané spojení	125	
špičkový odběr		350

SPI komunikace s hostitelským zařízením:

Modul umožňuje komunikovat s hostitelským zařízením pomocí SPI sběrnice a to buď v pomalém módu (CLK do 25 MHz) a nebo v rychlém módu (CLK do 80 MHz).

**Rádiové prostředky:****Tabulka 6:** Informace o rádiových prostředcích. [6]

Rádio:	Redpine Signals RS9110 + Airoha 8230
Výstupní výkon:	802.11b (CCK): +20dBm
	802.11g (OFDM): +17dBm
	802.11a (OFDM): +15dBm
Maximální vstupní výkon přijímače:	-10 dBm
Pracovní frekvence:	2.412 - 2.462 GHz, channel 1 – 11 (5 MHz / kanál)
	5.180 - 5.240 GHz, U-NII-1, channel 36, 40, 44, 48 (20 MHz / kanál)

**Popis softwaru****Obr. 12.** Softwarové schéma modulu OWL221a. [6]

Firmware dodávaný pro modul je pouze v binární podobě a nelze jej upravovat, dle vlastních požadavků. TCP/IP zásobník pro příslušný WiFi modul je prodáván zvlášť a jeho nákup je nevýhodný při pořízení jediného kusu. Tento fakt není nikde oficiálně uveden a distributoři nebo přímo výrobce jej sdělí pouze na přímý dotaz. Z tohoto důvodu se práce dále zabývá možnostmi modulu OWS451i.

**3.3.2 Popis WiFi modulu OWS451i**

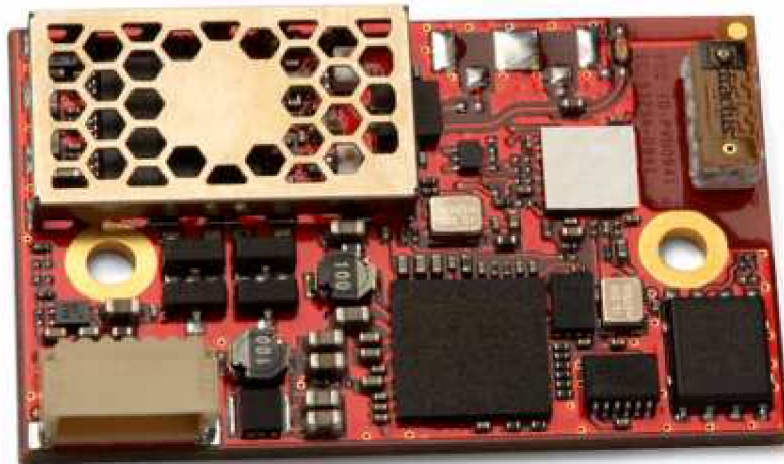
Podobně jako modul cB-OWL221a je i WiFi modul OWS451i určený pro komunikaci v pásmu 2,4 GHz nebo 5 GHz.

Hlavním rozdílem oproti modulu cB-OWL221a je to, že modul nepotřebuje žádný ovladač ani hostitelský počítač. Veškerý software je obsažený přímo v modulu. To umožňuje modul připojit téměř k libovolnému mikrokontroleru, který obsahuje jednotku UART nebo SPI. [7]

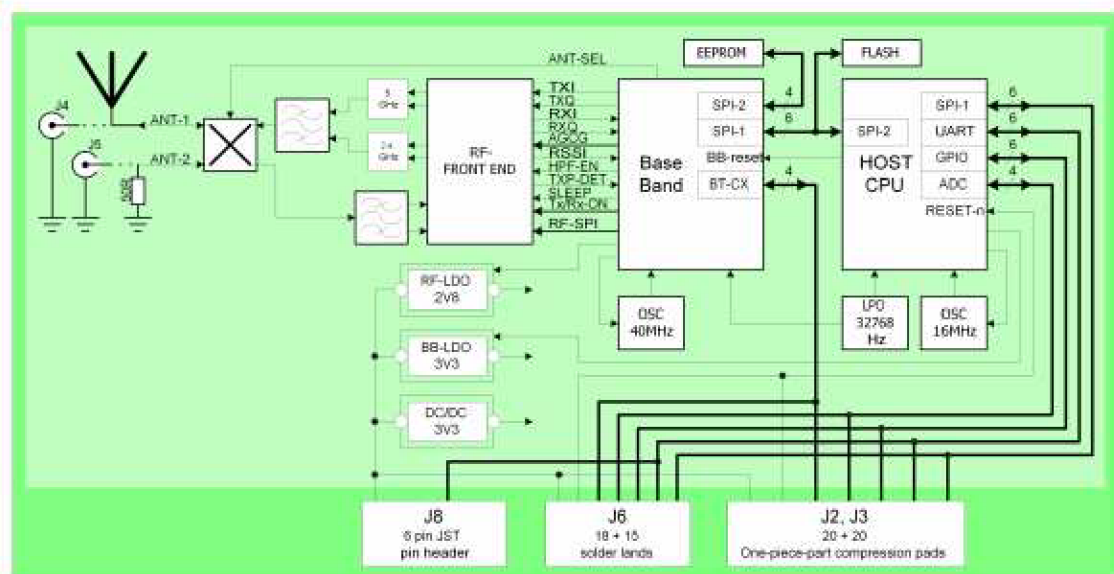
Klíčové rysy:

- Možnost komunikace v IEEE 802.11a b g nebo IEEE 802.11n.
- Hardwarový akcelerátor WEP, AES a CRC32.
- Podpora WPA a WPA2.
- QoS 802.11e a WMM.

- Rozhraní SPI nebo UART (3V logika).
- Podpora režimů ad-hoc a komunikace s infrastrukturou.
- Podpora TCP/UDP protokolu.
- Podpora DHCP server/klient módu.
- Certifikace pro provoz v Evropě.
- V souladu s normami EMC.
- Teplotní rozsah od -30 °C do 85 °C (průmyslové použití).
- Integrovaná anténa.



Obr. 13. WiFi modul cB-OWS451i (pohled shora). [13]



Obr. 14. Blokové schéma modulu cB-OWS451i. [13]

Stejně jako modul cB-OWL221a obsahuje i tento modul několik „konektorů“, přes které může být připojen k řídicímu zařízení. Konektory J4 a J5 jsou osazeny pouze u verze bez interní antény a slouží k připojení externích antén. Konektory J2, J3 a J6 jsou umístěny ze spodní strany a jsou realizovány vodivými ploškami. Konektor J8 je umístěn z horní strany modulu a jedná se o JST konektor označený jako SM06B-SRSS-TB. Jako vhodný portikus k tomuto konektoru je JST-SHR-06V-S-B. [13]



**Tabulka 7:** Vývody konektoru J8. [13]

Číslo pinu:	Označení:	Popis:
1	Vss	GND – zem
2	Vcc	Napájení 3,3 – 5 V
3	UART-TxD	UART vysílač 3V3
4	UART-RxD	UART přijímač 3V3
5	UART-CTS	UART Clar to send
6	UART-RTS	UART Request to send

**Rádiové prostředky:****Tabulka 8:** Informace o rádiových prostředcích. [13]

Rádio:	Redpine Signals RS9110 + Airoha 8230
Výstupní vf výkon:	802.11b (DSSS): +20dBm
	802.11g (OFDM): +17dBm
	802.11a (OFDM): +15dBm
Maximální vstupní výkon přijímače:	-10 dBm
Pracovní frekvence:	2.412 - 2.484 GHz, channel 1 – 14 (5 MHz / kanál)
	5.180 - 5.240 GHz, U-NII-1, channel 36, 40, 44, 48 (20 MHz / kanál)

Jak je vidět z tabulek Tabulka 6 Tabulka 8, moduly používají stejnou rádiovou jednotku, liší se pouze v tom, že modul cB-OWS451i umožňuje komunikovat i na kanálech 12, 13 a 14.

**Další vlastnosti modulu:**

Z Tabulky Tabulka 7 vyplývá, že WiFi modul komunikuje pomocí UARTu v logických úrovních 3,3 V, napájení může být v rozsahu 3,3 až 5 V. To umožňuje připojit tento modul přímo ke zvolenému vývojovému kitu Mini 2440 (viz kap. 0).

WiFi modul komunikuje s mikrokontrolerem (hostitelskou aplikací) pomocí UARTu implicitně nastaveného na rychlost 57600 baud/s, 8bitů, jeden stop bit bez parity. Dále je aktivní hardwarové řízení toku pomocí signálů RTS a CTS. Na straně vývojového kitu Mini 2440 toto řízení toku není podporováno, a proto je nutné modul napřed překonfigurovat a až poté jej připojit k Mini 2440.

**AT režim a režim pro přenos dat:**

WiFi modul může pracovat ve dvou rozdílných režimech. Jedním z nich je režim AT příkazů, druhý režim (datový) slouží pro přenos dat. Po startu je WiFi modul v datovém režimu. Pro přechod do AT režimu je potřeba odeslat sekvenci skládající se ze tří speciálních znaků, implicitně definovaných jako dopředné lomítko ('/'). Tento znak je uložen v S2 registru a je možné jej změnit pomocí AT\$S2 příkazu. Pro přechod z datového režimu do režimu AT příkazů je nutné dodržet i další kritéria specifikující časový odstup od ostatních přenosů dat mezi WiFi modulem a hostitelským zařízením. Před sekvencí musí být prodleva 1 s, stejně tak i po sekvenci musí být prodleva před prvním AT příkazem nejméně 1 s. Tyto hodnoty jsou specifikovány v AT\*AMET příkazu. Pro přechod z AT režimu do datového režimu slouží AT příkaz AT\*ADDM.

Tento příkaz je WiFi modulem potvrzen výpisem <CR><LF>OK<CR><LF>, stejně jako ostatní AT příkazy. [14]

### **Vyrovňovací paměť:**

V případě, kdy je modul odpojený od sítě, nebo dochází k jeho přepojování k přístupovému bodu apod., nemůže odesílat data, která přijímá přes UART (data v datovém režimu). K jejich zachycení slouží FIFO registr, který je schopný zachytit 256 bajtů. V případě, že nedojde ke spojení dříve, než se zaplní celá vyrovňovací paměť, dojde ke ztrátě dat směrem od nejstarších. Tato ztráta není nijak indikována. [14]

### **Roaming:**

#### a) Roaming v ad-hoc sítích:

V ad-hoc sítích není žádný pevný přístupový bod a jsou vhodné k vytvoření malých a improvizovaných spojení. Jeden z účastníků takovéto sítě musí převzít roli přístupového bodu. Pokud modul mající roli přístupového bodu přestane pracovat (výpadek napájení nebo výpadek signálu), musí jiný účastník sítě převzít jeho roli. Vrátil-li se původní účastník do sítě, považuje se stále za přístupový bod a není schopný spolupracovat s účastníkem jenž přebíral jeho povinnosti.

#### b) Roaming v sítích s přístupovým bodem:

V sítích s přístupovým bodem lze snížit čas roamingu na minimum přesným omezením kanálů, na kterých může modul danou síť nalézt. Přesnou specifikaci těchto kanálů, pomocí AT\*AGCL, nebude modul prohledávat ostatní kanály a tím ušetří čas při sestavování spojení. [14]

### **Úsporné režimy:**

V případě, kdy se využívají různé úsporné režimy dochází ke zvyšování doby odezvy na rádiovém rozhraní. Aby bylo možné optimalizovat nároky na spotřebovávanou energii a odezvu je nutné experimentovat s nastavením a najít optimum pro příslušnou aplikaci. Na straně modulu je možné přepínat do úsporného režimu buď procesor nebo rádiové rozhraní. Pro nastavení různých režimů slouží AT ATS4005 registr (AT příkaz). Tato práce se zabývá možnostmi sestavování rychlých spojení a proto ve výsledném programu není využito úsporných režimů. [14]

#### Online režim:

V tomto režimu nedochází k žádné úspoře energie, na druhou stranu je dosahována nejnižší latence.

#### Režim spánku:

Tento režim je nastaven jako výchozí. Rádiové rozhraní se snaží uspořit energii tím, že se odpojuje. Množství uspořené energie závisí na tom, jak často přístupový bod vysílá. V ad-hoc režimu nemůže docházet k úspoře energie.

#### Stop režim:

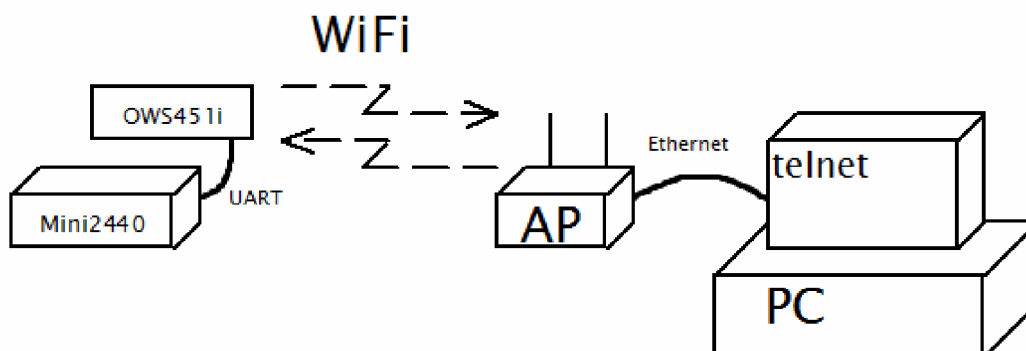
WiFi modul se chová podobně jako v režimu spánku (modul může být spojený s přístupovým bodem), navíc se zde vypíná i procesor a probudit ho může pouze pár přerušení. Jedno přerušení je aktivní ze strany rádiového rozhraní, v případě když dojde k příchodu dat. Druhé přerušení je aktivní ze strany UARTu a to konkrétně na pinu DSR.

**Příklad, jak nastavit WiFi modul pro příchozí a odchozí spojení[14]:**

Tento příklad zobrazuje nejnужnější nastavení WiFi modulu aby došlo k jeho asociaci s přístupovým bodem a mohlo být navázáno spojení. Význam jednotlivých příkazů je popsán v příloze této práce.

1. Nastavení WLAN spojení:
  - 1.1. Připojení se k modulu a vstup do AT režimu.
  - 1.2. Nastavení SSID sítě, do které budeme spojení provádět pomocí AT příkazu:  
AT\*AGSSID="My\_WLAN\_SSID",1
  - 1.3. Nastavení kanálu. Je-li zvolen kanál 0 dochází k automatickému vyhledávání kanálů. V tomto příkladu je použit kanál č. 6. AT\*AGCH=6,1
  - 1.4. Restartování WiFi modulu pomocí příkazu: AT\*AMWS=0,0,0,0,1,0
  - 1.5. Modul je nyní restartován. Aby se asocioval k AP, musejí být nastaveny parametry vyšší vrstvy.
  
2. Nastavení IP adresy:
  - 2.1. Připojení se k modulu a vstup do AT režimu.
  - 2.2. Zakázání přístupu na DHCP server AT příkazem: AT\*ANDHCP=0,1
  - 2.3. Nastavení IP adresy (192.168.0.5), masky podsítě (255.255.255.0) a výchozí brány (192.168.0.1) pomocí AT příkazu:  
AT\*ANIP="192.168.0.5","255.255.255.0","192.168.0.1",1
  - 2.4. Aktivace odposlechu příchozího IP spojení na portu 8787 pomocí příkazu:  
AT\*AMTL=8787,1,1
  - 2.5. Restartování WiFi modulu pomocí příkazu: AT\*AMWS=0,0,0,0,1,0
  - 2.6. Modul je nyní restartován a asociován s AP.

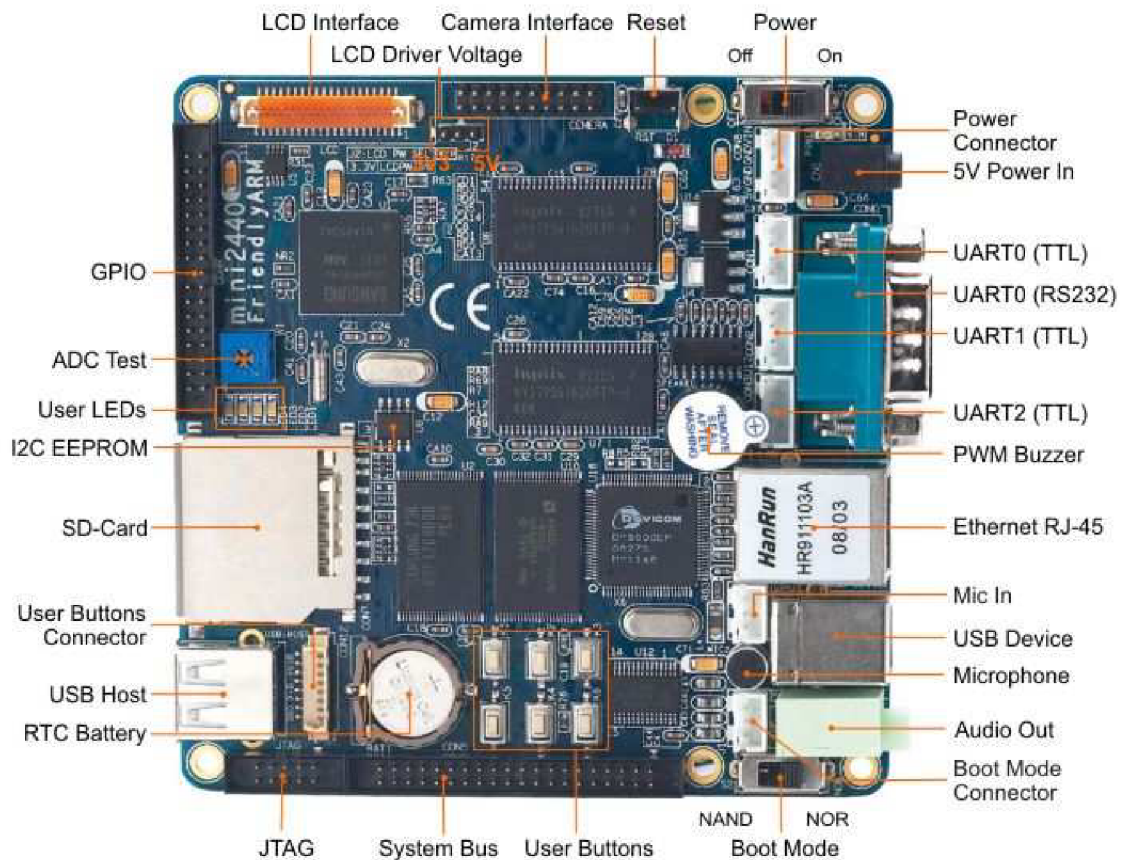
Nyní je modul nakonfigurován a je možné pracovat s přenosem dat (zpráv). Nejjednodušší cestou, jak ověřit správnost nastavení modulu je provést jednoduché spojení. Z PC, které je spojeno pomocí sítě s přístupovým bodem se k modulu připojíme pomocí služby telnet. Nastavení pro telnet je IP adresa WiFi modulu (192.168.0.5) port, na kterém očekává spojení (8787). Zprávy odeslané pomocí služby telnet se vypisují na sériovou linku WiFi modulu v datovém režimu a naopak, data zapsaná na sériovou linku WiFi modulu se přenášejí a zobrazují jako přijatá data přes telnet. [14]



**Obr. 15.** Blokové schéma pro spojení přes WiFi modul.

**Vývojový kit:**

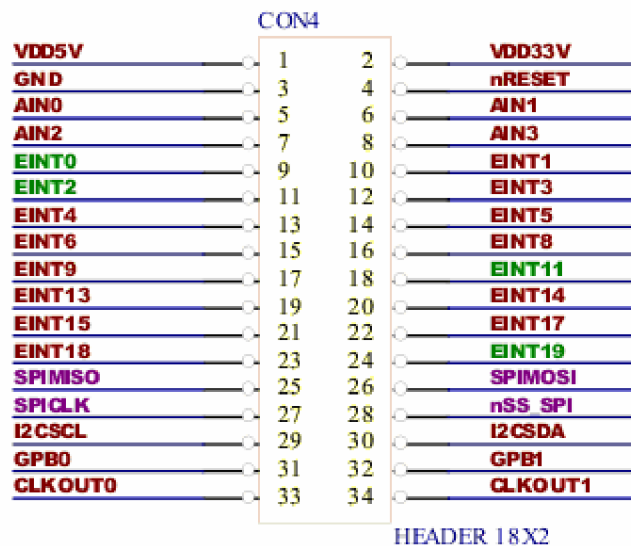
S ohledem na požadavky vybraného WiFi modulu cB-OWL221a a současně OWS451i, je potřeba volit takový řídicí procesor, který umožňuje pracovat s jádrem některého z podporovaných operačních systémů. Takové jsou procesory obsahující jádro ARM v podobě ARM 9 a novější. Návrhem univerzální hardwarové platformy pro systém Linux se zabývá [4]. Jedním, ze závěrů této práce je i fakt, že na dnešním trhu jsou dostupné již hotové vývojové kity a jejich pořizovací cena mnohdy nepřesahuje cenu obsažených součástí pořízených jednotlivě. Z tohoto důvodu byl vybrán vývojový kit firmy FriendlyARM s označením Mini 2440. Výrobce tyto kity označuje jako SBC (Single Board Computer) a přesné označení zvoleného modelu je S3C2440. [8]



**Obr. 16.** Rozmístění periférií vývojového kitu Mini 2440. [8]

Popis vývojového kitu:

- CPU: 400 MHz Samsung S3C2440A ARM920T (ARM 9)
- RAM: 64 MB SDRAM, 32 bit Bus
- Flash: 1GB NAND Flash a 2 MB NOR Flash s BIOSem
- EEPROM: 1024 B připojená na sběrnici I2C
- Sériový port: 2x v TTL úrovních, 1x v RS232 úrovních vyvedený na CAN9
- USB: 1x USB A-Host, 1x USB B-Device
- Ethernet: RJ-45 10/100 MB
- RTC: zdroj reálného času s baterií
- Audio výstup: 3,5 mm stereo jack
- Audio vstup: konektor + elektretový mikrofón
- Reproduktor: PWM bzučák
- Slot na paměťovou kartu: SD
- Kamera: 20ti pinový konektor pro připojení kamery
- LCD rozhraní: SNT displej max. 4096 barev s rozlišením 1024x768  
TFT displej max. 64k barev s rozlišením 1024x768
- Dotykový panel: rozhraní pro 4 drátový odporový dotykový panel
- Uživatelské vstupy: 6x tlačítko, 1x port s možností A/D
- Uživatelské výstupy: 4x LED
- Rozšiřující konektor: 34 pinový (CON4)
- Debug: 10ti pinový JTAG
- Napájení: 5V konektor, vypínač a indikační LED
- Podpora systémů: Windows CE 5 a 6  
Linux 2.6  
Android



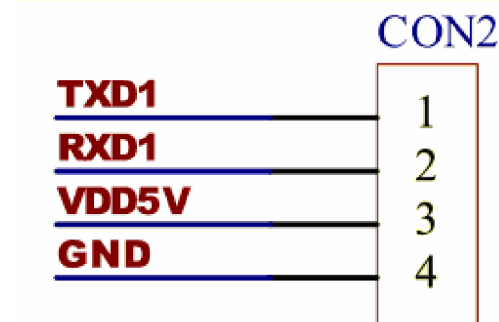
Obr. 17. Konektor CON4. [8]

Pro připojení WiFi modulu cB-OWL221a, je zapotřebí SPI sběrnice, která je vyvedena na výše zmíněný rozšiřující konektor, který je uvedený na Obr. 17.

**Tabulka 9:** SPI piny.

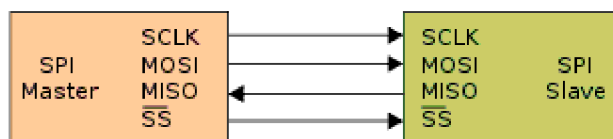
Signál	pin
MISO	25
MOSI	26
SCLK	27
SS	28

Pro připojení WiFi modulu OWS451i stačí sběrnice UART, které jsou na kitu 3, vyvedeny v TTL úrovních (**Obr. 16**)(**Obr. 18**).

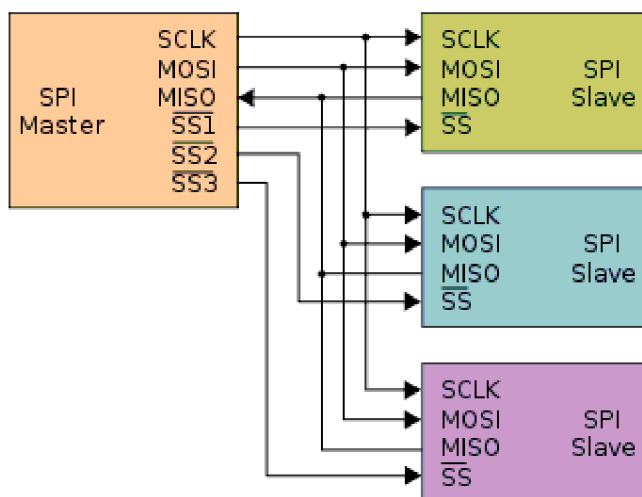
**Obr. 18.** Konektor CON2. [8]

### 3.3.3 Popis sběrnice SPI

SPI (serial peripheral interface) je sériové rozhraní sloužící k připojení periférií k jednomu zařízení. Hierarchie této sběrnice je taková, že jedno zařízení řídí celou komunikaci. Tímto zařízením je mikrokontrolér a v rámci hierarchie je označován jako master. Ostatní zařízení jsou označována jako slave a v rámci sběrnice komunikují pouze v případě, že jsou vybrána pomocí signálu SS (slave select). Tyto signály nastavuje master. Jeho dalším úkolem je generovat hodinový signál na vodič SCLK. Ten je společný pro všechna zařízení na sběrnici. Pro přenos dat jsou určeny zbývající vodiče sběrnice označené jako MOSI (master output slave input) a MISO (master input slave output). Příklad zapojení SPI sběrnice pro jeden master a jeden slave je na **Obr. 19**. Na **Obr. 20** je příklad zapojení SPI sběrnice při použití více zařízení typu slave.

**Obr. 19.** SPI s jedním slave zařízením. [9]





Obr. 20. SPI s více slave zařízeními. [9]

Výhodou této sběrnice je to, že pro oba směry komunikace používá oddělené vodiče a díky tomu může být komunikace velmi rychlá. Nevýhodou této sběrnice je to, že s narůstajícím počtem zařízení typu slave roste požadavek na počet řídicích signálů SS. To ale může být výhodou například s porovnáním oproti sběrnici IIC, kde zařízení typu slave obsahují adresu, která je zařízení pevně dána výrobcem a její variace je řešena na posledních bity adresy. To omezuje počet stejných zařízení připojených k jedné sběrnici IIC na  $2^n$ . U sběrnice SPI můžeme připojit libovolný počet stejných zařízení typu slave a jejich výběr je pouze v režii řídicích signálů SS.

### 3.3.4 Popis sběrnice UART

UART je univerzální asynchronní sběrnice realizovaná pomocí dvou vodičů. Slouží pro komunikaci pouze mezi dvěma zařízeními, kde ani jedno zařízení není nadřazeno druhému. Datová komunikace probíhá duplexně, to znamená, že na jeden vodič jedno zařízení zapisuje a druhé z něj čte a opačně na druhém vodiči. Protože se jedná o asynchronní přenos, je definován klidový stav sběrnice realizovaný jako log 1. Při zahájení zápisu dat na sběrnici je prve generován start bit, potom je přeneseno  $n$  datových bitů, paritní bit a  $m$  stop bitů.  $N$  bývá 7, 8 nebo 9 bitů,  $m$  jeden nebo dva bity. Parita může být sudá, lichá nebo žádná. Datové bity se přenášejí od LSB k MSB. Obě zařízení na sběrnici musí vědět, jaký formát je zvolený, současně musejí mít nastavenou stejnou přenosovou rychlost. Ta se pohybuje od 300 do 115200 baud/s, někdy i výš.



Obr. 21. Formát přenosu na sběrnici UART.

## 4 TCP/IP nebo UDP/IP z hlediska programování v C/C++ (Unixový OS)

### 4.1 Protokol IP

IP (Internet Protocol) je internetový protokol zajišťující komunikaci dvou počítačů předáváním IP paketů. IP paket obsahuje hlavičku a vlastní data. Adresace paketu se provádí pomocí IP adresy, která je součástí hlavičky IP paketu. IP adresa bývá často reprezentována textovým řetězcem, který je nutné pomocí DNS serveru přeložit na IP adresu. [10]

### 4.2 Protokol TCP

TCP (Transmission Control Protocol) umožňuje přenášet data spojovou službou. To znamená, že před vlastním přenosem dat dojde k vytvoření spojení mezi dvěma počítači a až potom jsou tímto spojením přenášena data. Všechna data jsou potvrzována a po ukončení přenosu dochází i k ukončení spojení. TCP paket obsahuje kromě přenášených dat i hlavičku. Součástí TCP hlavičky je i číslo portu. Toto dvou-bajtové číslo umožňuje adresovat příslušná data pouze jediné aplikaci na cílovém počítači. Zatímco IP protokol umožňuje spojení mezi dvěma počítači, TCP protokol se stará o spojení mezi dvěma aplikacemi na těchto počítačích. [10]

### 4.3 Protokol UDP

UDP (User Datagram Protocol) stejně jako TCP protokol slouží k přenosu dat mezi dvěma aplikacemi. Na rozdíl od TCP protokolu používá pro přenos nespojovanou službu, to znamená, že nedochází k navázání spojení. Data přenesená UDP protokolem nejsou potvrzována. Tato služba je sice nespolehlivá, ale s ohledem na menší zátěž sítě se často používá. UDP stejně jako TCP používá číslo portu pro adresaci cílové aplikace. Stejně číslo UDP a TCP portu ale nezpůsobí kolizi. Tato čísla jsou totiž na sobě nezávislá. [10]

### 4.4 Socket

Socket je velmi obecný nástroj pro komunikaci. Mohou ho užívat různé protokoly včetně TCP/IP a UDP/IP. Socketové API je shodné pro všechny operační systémy Unixového typu. Tedy na všech systémech by měly být k dispozici stejné funkce se stejnými parametry. [11]

Typy socketů:

Dva procesy spolu mohou komunikovat pouze tehdy, jsou-li jejich sockety stejného typu a jsou-li ve stejné doméně. V Unixové doméně mohou komunikovat dva procesy, které mají stejný souborový systém. V internetové doméně mohou komunikovat dva procesy, které běží na hostitelských počítačích, které jsou připojeny k internetu. Každý z nich má vlastní formát adresy.

Adresa socketu v Unixové doméně je textový řetězec, který je zaznamenán v systému souborů. Adresa socketu v internetové doméně je tvořena adresou hostitelského počítače. Tato adresa se nazývá IP adresou. Dále každý proces potřebuje znát číslo portu na daném hostiteli. Čísla portů jsou 16ti bitová neznaménková čísla.



Nižší čísla portů jsou v Unixu vyhrazena pro standardní služby. Pro ostatní služby jsou dostupná čísla od 2000 výše.

Existují dva široce používané typy socketů. Stream sockety a datagramové sockety. Stream sockety komunikují nepřetržitým tokem dat, zatímco datagramové sockety musejí číst celé zprávy najednou.

Stream sockety používají TCP protokol. Datagramové sockety používají UDP protokol. [10]

#### **4.5 Model komunikace klient – server**

Komunikace přístupem klient – server je nejčastějším případem, při komunikaci dvou procesů. Jeden proces se chová jako klient a ten se připojuje k druhému procesu, který se chová jako server. Proces chovající se jako klient musí vědět o existenci serveru, musí znát jeho jméno a adresu, zatímco server nemusí znát adresu klienta (dokonce nemusí mít informace o tom, zda proces klient existuje). Po navázání spojení (které iniciuje klient) mohou oba dva procesy zasílat a přijímat informace.

Každý proces si na své straně vytváří vlastní socket, přes který může komunikovat s ostatními procesy. Funkce (systémová volání) pro vytvoření socketu se pro klienta a server mírně liší. [11]

Kroky podílející se na vytvoření socketu na straně klienta:

- Vytvoření socketu pomocí funkce `socket()`.
- Připojení socketu na adresu serveru pomocí funkce `connect()`.
- Odesílání a příjem dat, například pomocí funkcí `read()` a `write()`.

Kroky podílející se na vytvoření socketu na straně serveru:

- Vytvoření socketu pomocí funkce `socket()`.
- Navázání socketu na adresu pomocí funkce `bind()`. Adresa je dána číslem portu na hostitelském počítači.
- Čekání na navázání spojení pomocí funkce `listen()`.
- Přijmutí spojení pomocí funkce `accept()`.
- Odesílání a příjem dat.

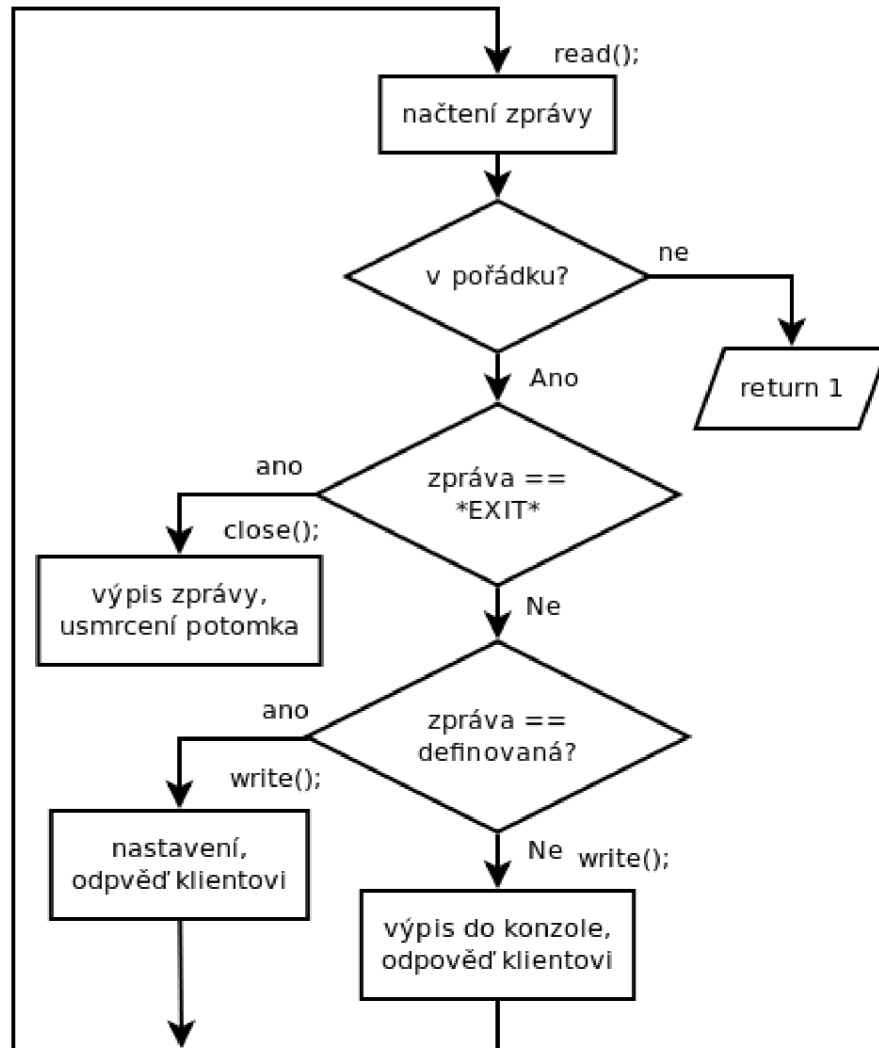
##### **4.5.1 Program TCP server**

Program TCP serveru byl vytvořen na základě výše uvedených informací a používá výše uvedené funkce. Je vytvořen pro Unixové operační systémy a umožňuje příjem zpráv od TCP klientů. Při spouštění programu se zadává jako parametr číslo portu na kterém server očekává klienty. Program TCP server je vytvořen jako konkurentní, to znamená, že může obsluhovat několik klientů najednou. Vývojový diagram TCP serveru je na **Obr. 23**. Detail smyčky obsluhy klienta je na **Obr. 22**.

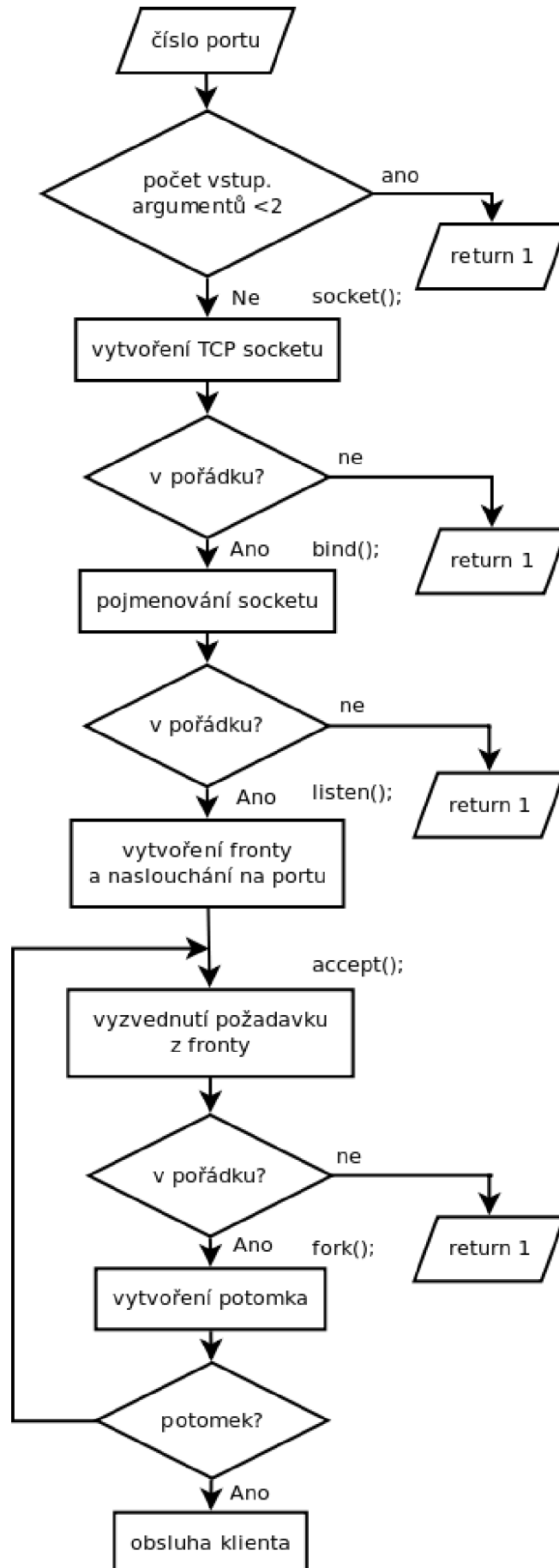
##### **4.5.2 Program TCP klient**

Program TCP klienta byl vytvořen jako klient k příslušnému TCP serveru. Program odesílá uživatelem zadaná data serveru. Při ukončení klienta (volbou 'n' při dotazu na pokračování) odesílá klient zprávu „\*EXIT\*“. Tu server rozpozná a korektně ukončí potomka, který daného klienta obsluhoval. Při spouštění klienta je nutné zadat dva parametry. Prvním parametrem je jméno hostitelského PC, na kterém je program TCP server spuštěný (může být i IP adresa). Druhým parametrem je číslo portu, na

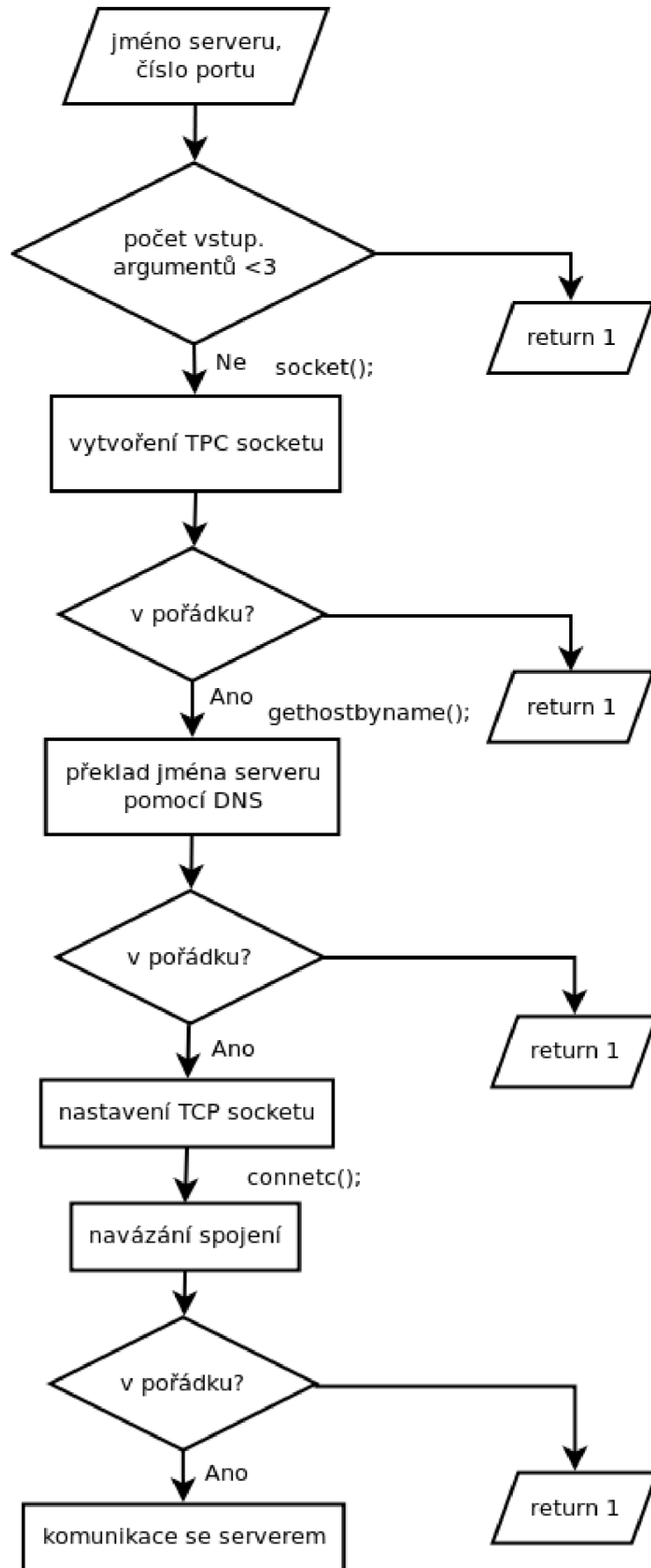
kterém TCP server očekává spojení. Vývojový diagram TCP klienta je na **Obr. 24**. Detail smyčky komunikace se serverem je na **Obr. 25**.



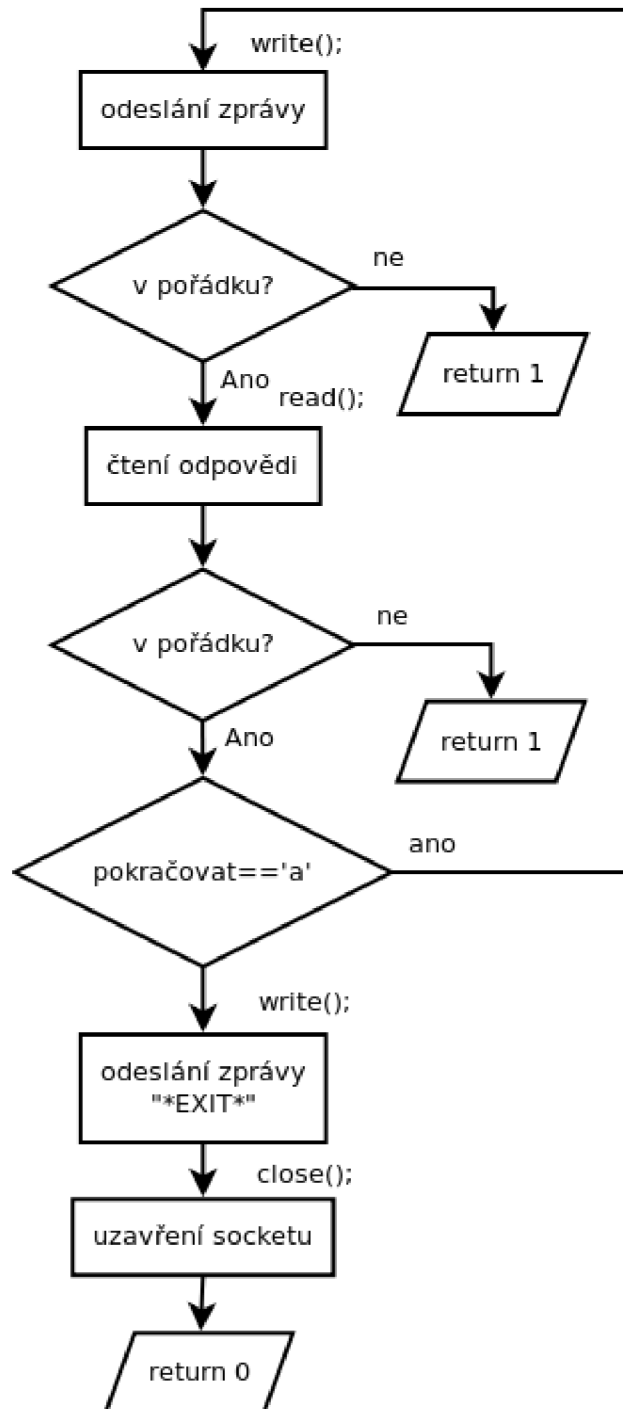
Obr. 22. Obsluha klienta.



Obr. 23. TCP server.



Obr. 24. TCP klient.



Obr. 25. Komunikace se serverem.

## 5 Přístup k HW prostředkům z hlediska programování v C/C++ (Unixový OS)

Tato kapitola ukazuje, jakým způsobem lze přistupovat k HW systému v OS Linux při vývoji aplikace v C/C++. Příklad je přímo zaměřený na práci se sériovou linkou UART. Další příklady, jak využít HW prostředky vývojového kitu FriendlyARM Mini 2440 popřípadě i Mini 6410, jsou uvedeny v příloze této práce.

### 5.1 Funkce open:

Přístup k HW prostředkům je umožněn přes funkci:

```
int open(const char *pathname, int flags);
```

Návratová hodnota je -1 v případě chyby. V ostatních případech je tato hodnota označena jako tzv. deskriptor a přes tento deskriptor lze přistupovat k příslušnému HW. První parametr předaný funkci je cesta, kde se nalézá přístup k HW. Standardně v OS Linux se jedná o složku /dev/. Další parametr (parametry) slouží k nastavení jiných vlastností HW. V rámci této práce je využito například:

```
int pwm_fd;
int fd_serial;
pwm_fd = open("/dev/pwm", 0); // Přístup k PWM bzučáku.
fd_serial = open(MODEMDEVICE, O_RDWR | O_NOCTTY ); // Přístup k sériovému
portu 1.
```

O\_RDWR -povoluje přístup ke čtení i zápisu

O\_NOCTTY -jedná se o tty terminál

Parametr MODEMDEVICE lze definovat makrem:

```
#define MODEMDEVICE "/dev/ttySAC1"
```

### 5.2 Funkce close:

Ukončení práce s HW se provádí pomocí funkce:

```
int close(int fd);
```

Návratová hodnota je -1 v případě chyby. Parametr fd je deskriptor HW, se kterým již nadále nechceme pracovat. V rámci této práce je využito například:

```
close(pwm_fd); // Uzavření PWM bzučáku.
```

### 5.3 Funkce ioctl:

Změnu základních parametrů umožňuje funkce

```
int ioctl(int d, int request, ...);
```

Návratová hodnota je -1 v případě chyby. Parametr d je deskriptor otevřeného HW (viz výše). Další parametry odpovídají nastavovaným parametrům. V rámci této práce je využito například:

```
ioctl(pwm_fd, PWM_IOCTL_STOP); // Vypnutí PWM bzučáku.
ioctl(pwm_fd, PWM_IOCTL_SET_FREQ, freq); // Nastavení frekvence PWM
bzučáku.
```

Parametry lze definovat makrem:

```
#define PWM_IOCTL_SET_FREQ      1
#define PWM_IOCTL_STOP         0
```

#### 5.4 Nastavení parametrů sériové linky:

Pro správu parametrů sériové linky jsou definovány dvě struktury. Struktura **oldtio** slouží pro zálohu parametrů, jaké jsou nastaveny při převzetí sériové linky, struktura **newtio** slouží k nastavení nových parametrů.

Ve funkci `setup_port()`; dochází k záloze parametrů, vynulování struktury pro nové nastavení a následně jsou nastavovány jednotlivé parametry. Parametr **c\_cflag** obsahuje základní nastavení (přenosová rychlost, počet datových a stop bitů, použitá parita, hardwarové řízení toku,...). Parametr **c\_iflag** obsahuje nastavení pro ignoraci bitů s chybou parity a mapování CR znaku. Parametr **c\_lflag** je nastaven na canonický přístup pomocí parametru ICANON. Tento typ přístupu je klíčový pro další práci se sériovou linkou a AT modemem. Tento přístup totiž znamená, že budou čtena data až po příchodu CR znaku. Parametry nastavené ve struktuře se aktivují pomocí funkce **int** `tcsetattr(int fd, int optional_actions, const struct termios *termios_p)`; Návrátová hodnota je -1 v případě chyby. Parametr `fd` je deskriptor sérového portu, parametr `optional_actions` je nastaven pomocí makra na TCSANOW. Posledním parametrem je struktura **newtio**.

#### 5.5 Funkce read a write:

Pro zápis dat na sériovou linku slouží funkce `write()` a pro jejich čtení slouží funkce `read()`.

```
ssize_t read(int fd, void *buf, size_t count);
```

Funkce vrací délku přečtených dat. Prvním parametrem funkce je deskriptor sériové linky, druhým parametrem je prostor, kam mají být přečtená data uložena, a třetím parametrem je délka alokovaného prostoru. Například lze použít:

```
#define      N          256          // Délka bufferu.
char buffer[N];
int res;                               // Pro uložení délky přijatých dat.
.
.
.
res = read(serial_fd,buffer,N);        // Přečtení zprávy ze sériové linky.
buffer[res]='\0';                      // Doplnění o ukončovací znak.
```

```
ssize_t write(int fd, const void *buf, size_t count);
```

Funkce vrací -1 v případě chyby, 0 v případě, kde nedošlo k odeslání žádných dat. V případě, že došlo k odeslání dat, funkce vrací hodnotu délky odeslané zprávy. Prvním parametrem funkce je deskriptor sériové linky, druhý parametr jsou data k odeslání, třetím parametrem je údaj o délce zprávy. Příklad:

```
res = write(serial_fd, "AT*AGSCAN?\r\n",12); // Odeslání AT příkazu pro scan sítě.
```

## 6 Komunikační program pro Mini 2440 a OWS451

Na straně vývojového kitu musí běžet jednoduchý program, který bude představovat TCP server, na který bude možný přístup pomocí programu TCP klient popsaným výše (kapitola 4.54.5.2), nebo program, který se připojí na TCP server popsaný výše (kapitola 4.5.14.5). Použití WiFi modulu OWS451 celou věc zjednodušuje, protože modul sám o sobě obsahuje TCP stack. Po nastavení podle 3.3.2 tvoří WiFi modul TCP server očekávající spojení na příslušném portu. Spojení lze provést pomocí aplikace Telnet nebo programu TCP server. Nastavení WiFi modulu jako klienta pro konkrétní server se provádí následovně (WLAN spojení musí být nastaveno tak, jak je uvedeno v 3.3.2):

1. Nastavení OWS451 jako klient:
  - 1.1. Připojení se k modulu a vstup do AT režimu.
  - 1.2. Nastavení parametrů serveru, ke kterému se budeme připojovat:  
`AT*ADWDRP=0,tcp://192.168.1.10:8008,2,0,„peer1“,1`  
 Nastavení platí pro server s adresou 192.168.1.10 naslouchajícím na portu 8008. „peer1“ je pojmenování pro nastavení.
  - 1.3. Nastavení indexu nastaveného serveru pomocí AT příkazu:  
`AT*ADNPR=1,1`
  - 1.4. Restartování WiFi modulu pomocí příkazu: `AT*AMWS=0,0,0,0,1,0`

Po restartu se modul asociuje s přístupovým bodem a pokusí se připojit k nastavenému serveru. Modul se automaticky spustí v datovém režimu a data přijatá na sériové lince odesílá TCP serveru, data přijatá přes rádiové rozhraní zapisuje na sériovou linku. [14]

Dále je možné na WiFi modulu spustit server pro příjem UDP datagramů. To lze provést nastavením příslušného portu (zde 5000) a povolením příjmu pomocí příkazu: `AT*AMUR=5000,1,1`.

### 6.1 Příprava WiFi modulu:

K dodanému WiFi modulu OWS451i byl nakrimpován portikus konektoru J8, aby nemusely být pájeny kontakty přímo k modulu a ten mohl být později použit i pro jiné aplikace. Jak bylo popsáno výše, konektor obsahuje všechny nezbytně nutné signály.

Dále byl WiFi modul připojen přes převodník USB → UART (3V logic), který obsahoval i signály HW řízení toku a přes tento převodník byl WiFi modul konfigurován z PC pomocí programu „AT Serial Port Adapter Toolbox“, který pro svá zařízení vyvinul výrobce connectBlue. Program starší verze umožnil pohodlný přístup do AT režimu a nastavení sériové komunikace (vypnutí HW řízení toku), dále umožnil nastavení WLAN spojení, nastavení IP adresy a nastavení modulu jako klienta nebo serveru, bez komplikovaného formátování jednotlivých AT příkazů. Po nastavení požadovaných parametrů a restartu WiFi modulu však nedocházelo k jeho asociaci s nastaveným a dostupným přístupovým bodem. Při snaze asociovat WiFi modul s přístupovým bodem bylo vyzkoušeno mnoho přístupů, ať už s pevnou IP nebo s povolením přiřazení IP pomocí DHCP serveru. Všechny pokusy byly neúspěšné. K problémům s WiFi modulem docházelo i tehdy, kdy byl přepnut na rozsah 5 GHz WiFi. Při následném pokusu o sken dostupných přístupových bodů došlo k zaseknutí modulu a ani po jeho restartu (odpojení a připojení k napájení) nebylo možné modul přepnout do AT režimu. Modul musel být restartován s obnovením továrního nastavení.



To bylo provedeno přidržením úrovně GND na pinech 6. a 8. konektoru J6 po připojení napájení po dobu cca 5 vteřin.

Na základě těchto problémů bylo provedeno měření stavových signálů vyvedených na konektor J6 pomocí osciloskopu v režimu zachycení hran. Toto měření prokázalo, že se WiFi modul nechová tak, jak je očekáváno. Na základě těchto zjištění byl kontaktován distributor, který poskytl novější verzi programu pro konfiguraci, která již zohledňuje použitý WiFi modul OWS451i a nový firmware pro WiFi modul (verze 2.6.0 ze dne 24.1.2011). Po aktualizaci firmwaru již nedocházelo k zablokování WiFi modulu při přepnutí rádia na 5 GHz rozsah a došlo k asociaci WiFi modulu s přístupovým bodem. Navázané spojení bylo ovšem velmi nestabilní. Při zjišťování odezvy (pomocí ping) docházelo k velkým prodlevám a velké chybovosti (i při komunikaci na méně než jeden metr). Při pokusu opětovně restartovat WiFi modul s obnovením továrního nastavení došlo k jeho zablokování. To se projevilo tak, že WiFi modul nebyl schopný přejít do AT režimu. Tento fakt vypovídá o tom, že i nejnovější poskytnutý firmware není zcela bez chyb. Po několika dalších přeflashováních firmware na verzi 2.6.0 se podařilo WiFi modul asociovat s přístupovým bodem a navázat relativně stabilní spojení.

## 6.2 Programy pro stranu vývojového kitu:

Po zprovoznění komunikace mezi WiFi modulem a přístupovým bodem byly vytvořeny programy, které slouží k demonstraci možností, jak přistupovat k možné komunikaci. Konkrétní nastavení WiFi modulu je následovné:

- WiFi modul má nastaveno SSID a ostatní parametry WLAN spojení dle použitého přístupového bodu,
- WiFi modul je nastaven s pevnou IP adresou 192.168.1.110 a jako brána je použita adresa přístupového bodu,
- WiFi modul má povolen příjem UDP datagramů na portu 5000,
- WiFi modul navazuje spojení se serverem na adrese 192.168.1.101 a portu 8000.

### 6.2.1 Program pro skenování dostupných sítí:

Pro seznámení se s možnostmi programování komunikace mezi WiFi modulem a vývojovým kitem FriendlyARM byl sestaven jednoduchý program, který po zapnutí přepne WiFi modul do AT režimu a cyklicky skenuje dostupné WiFi sítě. Jejich parametry třídí do struktury a vypisuje na displej. Nalezení alespoň jediné sítě je oznámeno tónem a odlišným tónem je oznámen stav, kdy není dostupná žádná WiFi síť. Zjednodušený vývojový diagram tohoto programu je zobrazen na **Obr. 26**.

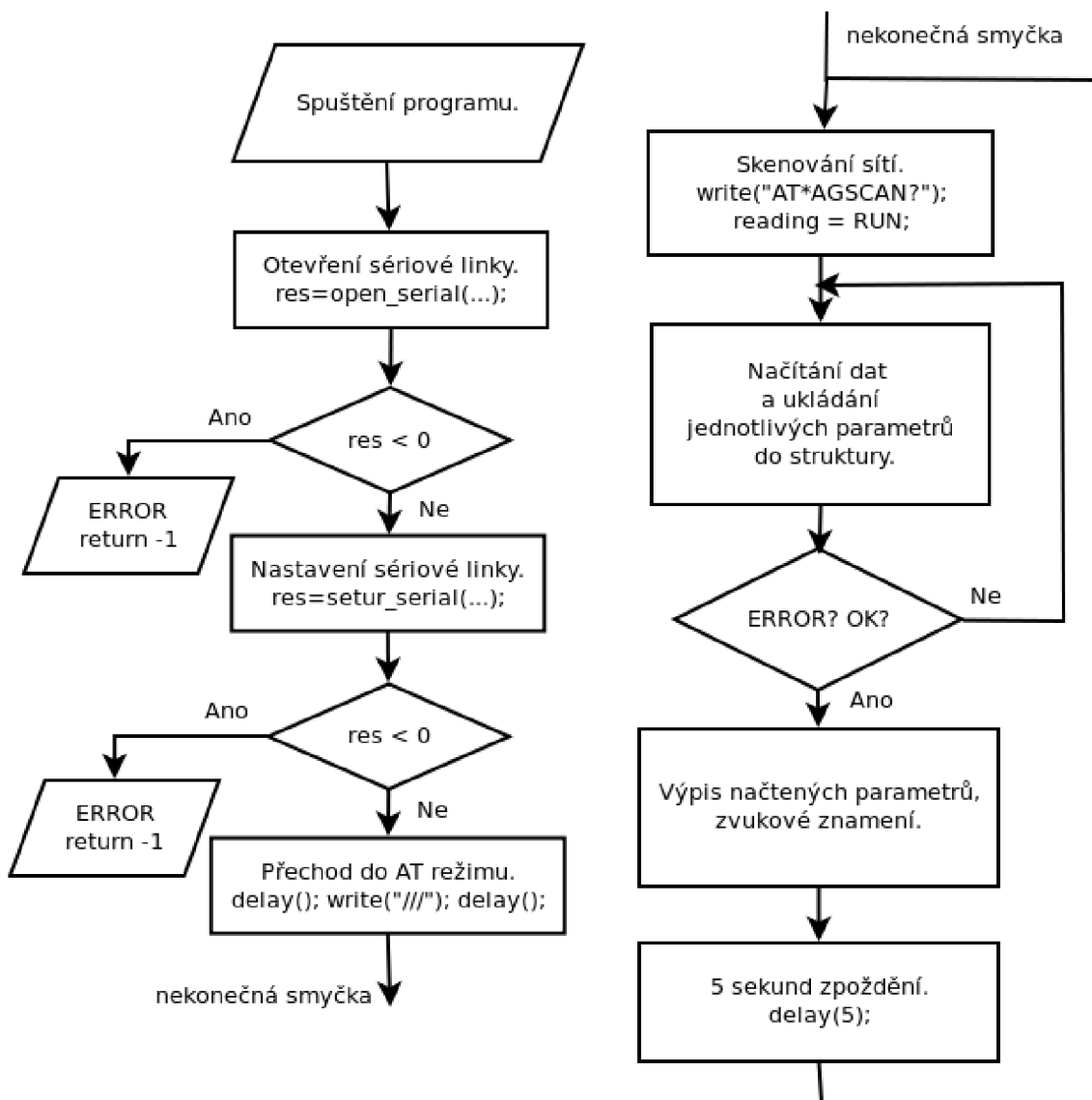
### 6.2.2 Popis programu pro komunikaci na straně vývojového kitu:

Jak bylo uvedeno výše, modul je nastaven pro komunikaci se serverem pomocí TCP a pro příjem UDP datagramů. To vše je obslouženo WiFi modulem a data, která jsou přijata jsou předána na sériovou linku. Zde již nelze rozlišit, jakou technologií a od jakého uživatele data přišla. Jestliže chceme mít toto rozlišeno, musíme již při odesílání dat zajistit jejich podpis.

WiFi modul v datovém režimu nedává žádnou možnost kontroly, zda je spojení s TCP serverem doopravdy navázáno. Přepnutím WiFi modulu do AT režimu dojde k přerušování TCP spojení (lze nastavit i jinak, pak ale hrozí přetečení vyrovnávací paměti a ztráta dat).

Data jsou ze sériové linky čtena pomocí funkce `read()`; a chceme-li odeslat data TCP serveru, musíme je zapsat na sériovou linku pomocí funkce `write()`. Po přečtení jsou data uložena jako řetězec znaků a jsou-li stanovena nějaká pravidla pro řídicí znaky, mohou být v přijatém řetězci tyto znaky vyhledány. Tímto způsobem by mohla být data tříděna a dále zpracovávána podle jejich významu.

Program po spuštění inicializuje a nastaví sériovou linku (stejně jako program pro skenování dostupných sítí). Potom dojde k vytvoření vlákna pro příjem dat funkcí `read()`. Ta je uzavřena spolu s vyhodnocováním dat v nekonečné smyčce. Samostatné vlákno je vytvořeno proto, že funkce `read()` je blokující. Výkon programu se na ní zastavuje v případě, že nejsou žádná data dostupná. V další nekonečné smyčce je uzavřena funkce pro odeslání dat TCP serveru. V této smyčce je vytvořeno zpoždění pomocí funkce `sleep(1)`; (zpoždění o 1 s), aby běh programu nebyl příliš náročný a dal se sledovat na displeji. Program je zkompileován pro vývojový kit Mini 2440 a lze jej spustit příkazem „./c2c“. Program vyžaduje, aby WiFi modul byl připojen k sériové lince číslo 1.



Obr. 26. Sken dostupných WiFi sítí.

### 6.3 Programy pro stranu infrastruktury:

Pro stranu infrastruktury je vytvořen TCP\_server, ke kterému se mobilní zařízení (Mini 2440) bude připojovat, dále program UDP\_sender, který umožňuje zasílat UDP datagramy mobilnímu zařízení a UDP\_mess, který umožňuje odeslat mobilnímu zařízení jakoukoliv uživatelem zadanou zprávu jako UDP datagram.

#### 6.3.1 Program TCP\_server:

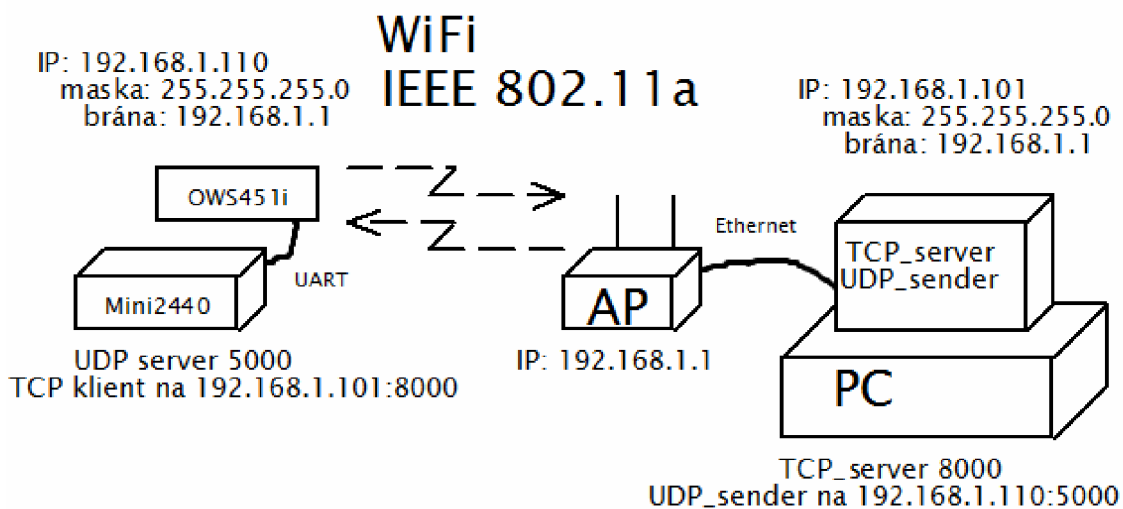
Program TCP\_server je popsán v kap. 4.5.1 a je použit bez velkých změn.

#### 6.3.2 Program UDP\_sender:

Tento program cyklicky odesílá UDP datagramy WiFi modulu. Adresou je pevně nastavená IP adresa WiFi modulu (192.168.1.110) a číslo portu (5000). Program nečeká žádnou odpověď a odesílání je zpožděno pomocí funkce *sleep(1)*; (zpoždění smyčky o 1 s).

#### 6.3.3 Program UDP\_mess:

Tento program slouží k odeslání jedné zprávy, kterou je možné editovat přímo v příkazovém okně. Program se spouští se dvěma parametry. Prvním parametrem je IP adresa příjemce a druhým parametrem je číslo portu. To může vypadat například: „./UDP\_mess 192.168.1.110 5000“.



Obr. 27. Výsledné zapojení systému.

## 7 Závěr

System C2C bude umožňovat celou řadu bezpečnostních služeb i služeb, které se nebudou týkat bezpečnosti provozu na pozemních komunikacích, ale budou mít za úkol zvýšení komfortu cestujících, zajištění zásobování řidiče nebo celé posádky informacemi týkajícími se provozu na pozemních komunikacích, stavu vozidla, místních zajímavostí (reklama) apod.

System C2C je navrhován tak, aby informace týkající se bezpečnosti provozu na pozemních komunikacích byly přenášeny odděleně od ostatních informací. Předpokládá se využití frekvenčního pásma od 5,885 do 5,905 GHz. Tyto informace budou podléhat největšímu zabezpečení, aby byly vždy důvěryhodné a tedy 100 % použitelné. Požadavek na důvěryhodnost dat týkajících se bezpečnosti provozu na pozemních komunikacích je jedním z kritických bodů systému C2C. Dalším důležitým bodem (který je dodnes nevyřešený) je právní otázka zodpovědnosti za vzniklé škody v případě, kdy systém udělá chybu a kvůli té dojde k újmě na majetku nebo lidských životech. Už samotné prokazování faktu, zda za nehodu může chyba systému nebo lidský faktor nebude jednoduché.

Z přenosového hlediska je systém navrhován tak, aby umožňoval komunikaci mezi vozidly, nebo mezi vozidlem a stacionární jednotkou umístěnou u silnice. Tyto jednotky mohou přenášet důležitá data do projíždějících vozidel, nebo mohou sloužit k připojení k internetu vozidel stojících v jejich dosahu.

Realizace zařízení, které by se chovalo podobně jako plánovaná zařízení C2X a C2C, je realizováno pomocí IEEE 802.11a protože je nejpodobnější IEEE 802.11p, které bude systém využívat.

První směr realizace počítal s využitím WiFi modulu OWL221a, který je určen pro systémy s operačním systémem Linux 2.6 apod. Takováto realizace by na straně vývojového kitu umožnila použití stejných zdrojových kódů pro realizaci klient nebo server aplikace jako na straně infrastruktury, zkompileovaných pro příslušnou architekturu (ARM procesor). Využití tohoto modulu je komplikované s ohledem na placený IP stack. Z tohoto důvodu se práce dále zabývala využitím modulu OWS451i. Tento modul je dostupný od ledna 2011 jako engineering sample a hlavní rozdíl oproti OWL221a je v tom, že modul sám o sobě obsahuje kompletní firmware a připojení k hostitelské aplikaci se provádí pomocí sběrnice UART. WiFi modul se chová jako AT modem se specifickou sadou AT příkazů.

Vývojový kit byl volen s ohledem na požadavky WiFi modulu OWL221a. S ohledem na cenovou dostupnost byl vybrán vývojový kit FriendlyARM Mini 2440. Tento vývojový kit umožňuje i připojení náhradního WiFi modulu OWS451i bez jakékoli úpravy.

Problémy, které provázely ožívání WiFi modulu OWS451i jsou popsány v kapitole 6.1. Je zřejmé, že ani současná verze firmware není 100 % odladěna a nějaké problémy mohou přetrvávat. WiFi modul OWS451i může pracovat jako TCP server i klient a jako UDP server. Přijímaná data ze všech rádiových zdrojů předává na sériovou sběrnici bez jejich identifikace. To na jednu stranu zjednodušuje práci programátora, ale pokud je potřeba data odlišit, musejí být opatřena identifikací již od odesílatele. To může navyšovat režii na rádiovém rozhraní. Další problém nastává, je-li současně otevřena komunikace pomocí dvou TCP spojení (WiFi modul jako klient i jako server). V takovém případě nelze WiFi modulu určit, komu má předat data zapisovaná na sériovou sběrnici. Tento případ komunikace by neměl nikdy nastat a právě z tohoto důvodu je WiFi modul pro účely této práce nastaven jako klient přesně danému TCP serveru a UDP server (pouze příjem dat).

Data odesílaná modulem jsou předávána TCP serveru na straně infrastruktury a odpovědi serveru jsou opatřeny identifikací serveru. Modul dále může přijímat UDP datagramy, které by měly být identifikovány jinak (aby nemohly být zaměněny s odpověďmi serveru). UDP datagramy nejsou potvrzovány ani na úrovni aplikační vrstvy. Jejich odesílatel tedy nemůže mít nikdy jistotu, že byla data přijata a akceptována.

TCP server na straně infrastruktury nezjistí, zda jsou jeho data předána příslušné aplikaci (nejsou-li zde vyžadovány odpovědi na úrovni aplikační vrstvy). Nakonfigurovaný WiFi modul se připojí k TCP serveru vždy po připojení napájení (je-li dostupná i rádiová cesta). TCP server nemůže mít v tuto chvíli jistotu, že data, která WiFi modulu odešle (a ta mu TCP protokolem budou potvrzována), zpracovává aplikace, které jsou data určena.

Na základě zkušeností s WiFi modulem OWS451 lze říct, že pro simulaci systému Car2Car a Car2X není tento WiFi modul příliš vhodný. WiFi modul OWS451 nepodporuje IPv6, se kterým systém C2C počítá. WiFi modul sice sám o sobě obstarává veškerou komunikaci na rádiovém rozhraní, na druhou stranu neumožňuje takové možnosti, jako plné síťové rozhraní. To by mohl umožnit původně plánovaný WiFi modul OWL221. Na druhou stranu modul OWS451 je možné připojit i k jednoduchému mikrokontroleru, který je vybaven sběrníci UART. Zde je pak jednoduchost komunikace výhodou.

## 8 Seznam zkratek

4G	4. Generation
ABS	Anti-blocking system
ASR	Antriebs Schlupf Regelung
AP	Access Point
AT	Attention
AU	Application Unit
C2C	Car To Car
C2C-CC	The CAR 2 CAR Communication Consortium
C2X	Car To Infrastructure
CA	Certification Authority
CTS	Clear So Send
DNS	Domain Name Server
ESP	Electronic Stability Programme
ETSI	The European Telecommunications Standards Institute
GPRS	General Packet Radio System
GPS	Global Position System
GSM	Global System for Mobile communications
HS	Hotspot
HSDPA	High Speed Downlink Packet Access
HW	Hard Ware
IEEE	Institute of Electrical and Electronics Engineers
IIC	Inter-Integrated Circuit
IP	Internet Protocol
IPv4	Internet Protocol, version 4
IPv6	Internet Protocol, version 6
LSB	Least Significant Bit
MANET	Mobil Ad-hoc Network
MISO	Master Input, Slave Output
MOSI	Master Output, Slave Input
MSB	Most Significant Bit
OBU	On Board Unit
OS	Operation System
PC	Personal Computer
PKI	Public key infrastructure
RSU	Road-side Unit
RSU	Remote Service Unit
RSSI	Received Signal Strength Indication
RTS	Request To Send
SBC	Single Board Computer
SCLK	Slave Clock
SPI	Serial Peripheral Interface
SS	Slave Select
TC ITS	Technical Committee for Intelligent Transport Systéme
TCP	Transmission Control Protocol
TELNET	Telecommunication Networks
TTL	Transistor – Transistor Logic
UART	Universal Asynchronous Receiver/Transmitter

UDP	User Datagram Protocol
UMTS	Universal Mobile Telephone Standard
VANET	Vehicular Ad-hoc Network
WIMAX	Worldwide Interoperability for Microwave Access

## 9 Literatura

- [1] *CAR 2 CAR Communicaion Consorciium* [online]. 2003 [cit. 2010-01-30]. Mission & Objectives. Dostupný z WWW: <<http://www.car-to-car.org/>>.
- [2] C2C-CC. *CAR 2 CAR Communication Consortium Manifesto : Overview of the C2C-CC System*. 1st edition. [s.l.] : [s.n.], 2007. 94 s. Dostupný z WWW: <<http://www.car-to-car.org/index.php?id=31>>.
- [3] *Klub Honda : Nová technologie* [online]. 2008. 2008 [cit. 2010-01-30]. Aktuality a sport. Dostupné z WWW: <[http://moto.honda.cz/klub-honda/aktuality-a-sport.html/304\\_423-nova-technologie/2](http://moto.honda.cz/klub-honda/aktuality-a-sport.html/304_423-nova-technologie/2)>.
- [4] SKOPAL, Miroslav. *Univerzální hardwarová platforma podporující operační systém Linux*. FEKT VUT v Brně, 2010. 64 s. Diplomová práce. VUT v Brně.
- [5] *Všeobecné oprávnění č. VO-R/12/08.2005-34*. Praha : Český telekomunikační úřad, 9.srpna 2005. 5 s.
- [6] *CB-OWL22X ELECTRICAL AND MECHANICAL DATA SHEET*. Malmö Sweden, 12.2. 2009. 34 s.
- [7] *Product Brief OWS451 (UART interface)*. Malmö Sweden, 2010. 2 s.
- [8] *Manual\_mini2440*. 78 s. Dostupné z WWW: <[http://www.friendlyarm.net/dl.php?file=mini2440\\_manual.pdf](http://www.friendlyarm.net/dl.php?file=mini2440_manual.pdf)>.
- [9] Serial Peripheral Interface. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , last modified on 6. 12. 2010 [cit. 2010-12-18]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](http://cs.wikipedia.org/wiki/Serial_Peripheral_Interface)>.
- [10] DOSTÁL, Radim. *Sockety a C++. Builder* [online]. 2002, [cit. 2010-12-9]. Dostupný z WWW: <[http://www.builder.cz/art/cpp/sokety\\_a\\_cpp.html](http://www.builder.cz/art/cpp/sokety_a_cpp.html)>.
- [11] *LinuxHowtos* [online]. 2006, 25.5. 2007 [cit. 2010-12-18]. Socket Tutorial. Dostupné z WWW: <[http://www.linuxhowtos.org/C\\_C++/socket.htm](http://www.linuxhowtos.org/C_C++/socket.htm)>.
- [12] MÜLLER, Martin. *WLAN 80211p Measurements for V2V DSRC*. 09.2009, [cit. 2010-12-29]. Dostupný z WWW: <[http://www2.rohde-schwarz.com/file\\_12631/1MA152\\_2e.pdf](http://www2.rohde-schwarz.com/file_12631/1MA152_2e.pdf)>.
- [13] *CB-OWS451 ELECTRICAL AND MECHANICAL DATA SHEET*. ver. 3, 25.11. 2010. 41 s.
- [14] *WIRELESS LAN SERIAL PORT ADAPTER AT COMMANDS*. ver. 3, 20.1. 2010. 47 s.



- [15] *Downloader Instructions*. FriendlyARM, [cit. 2010-3-28]. Dostupné z WWW: <[http://www.friendlyarm.net/dl.php?file=linux\\_download.pdf](http://www.friendlyarm.net/dl.php?file=linux_download.pdf)>.
- [16] *Manual\_mini6410*. 358 s. Dostupné z WWW: <[http://www.friendlyarm.net/dl.php?file=mini6410\\_manual\\_20101121.pdf](http://www.friendlyarm.net/dl.php?file=mini6410_manual_20101121.pdf)>.
- [17] *Linuxsoft.cz* [online]. 2011, [cit. 2011-2-18]. C/C++ serial. Dostupné z WWW: <[http://www.linuxsoft.cz/article.php?id\\_article=370](http://www.linuxsoft.cz/article.php?id_article=370)>.
- [18] DOSTÁLEK, Libor; KABELOVÁ, Alena. *Velký průvodce protokoly TCP/IP a systém DNS : 3. aktualizované a rozšířené vydání*. Praha : Computer Press, 2002. 542 s.

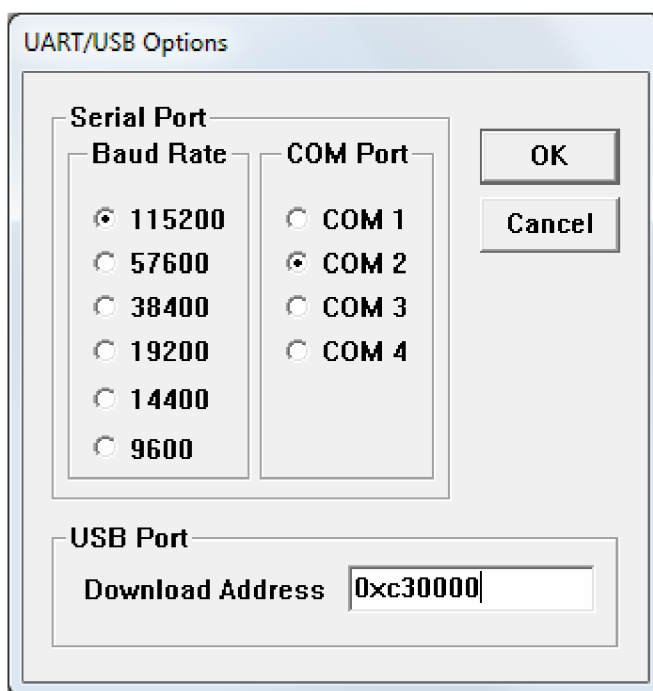
## Příloha 1 – Zkušenosti a poznámky k práci s kity Mini 2440 a 6410.

Tato příloha vznikla na základě praktických zkušeností s používáním vývojových kitů Mini 2440 a Mini 6410. Část této přílohy vychází přímo z anglických a čínských manuálů odpovídající literatuře [8] a [16].

### Zavedení operačního systému do Mini 2440:

Zavedení nového operačního systému do Mini 2440 musí probíhat z osobního počítače s operačním systémem Windows, nejlépe distribuce XP.

1. Do počítače nainstalujeme program **usb-downloader-setup\_20090421** dostupný na DVD distribuovaném s vývojovým kitem, nebo dostupný na webu FriendlyARM.
2. Připojíme Mini 2440 k PC pomocí USB kabelu (konektor USB device) a pomocí sériové linky.
3. Spustíme program DWN.exe, který je také na DVD nebo webu FriendlyARM.
4. Klikneme na Configurations → Options a provedeme nastavení sériové linky. Zde vybereme příslušný sériový port PC, přenosovou rychlost ponecháme 11520. Download Address změníme na 0xc30000. Vše potvrdíme kliknutím na OK.



**Obrázek 1:** Nastavení sériové linky.

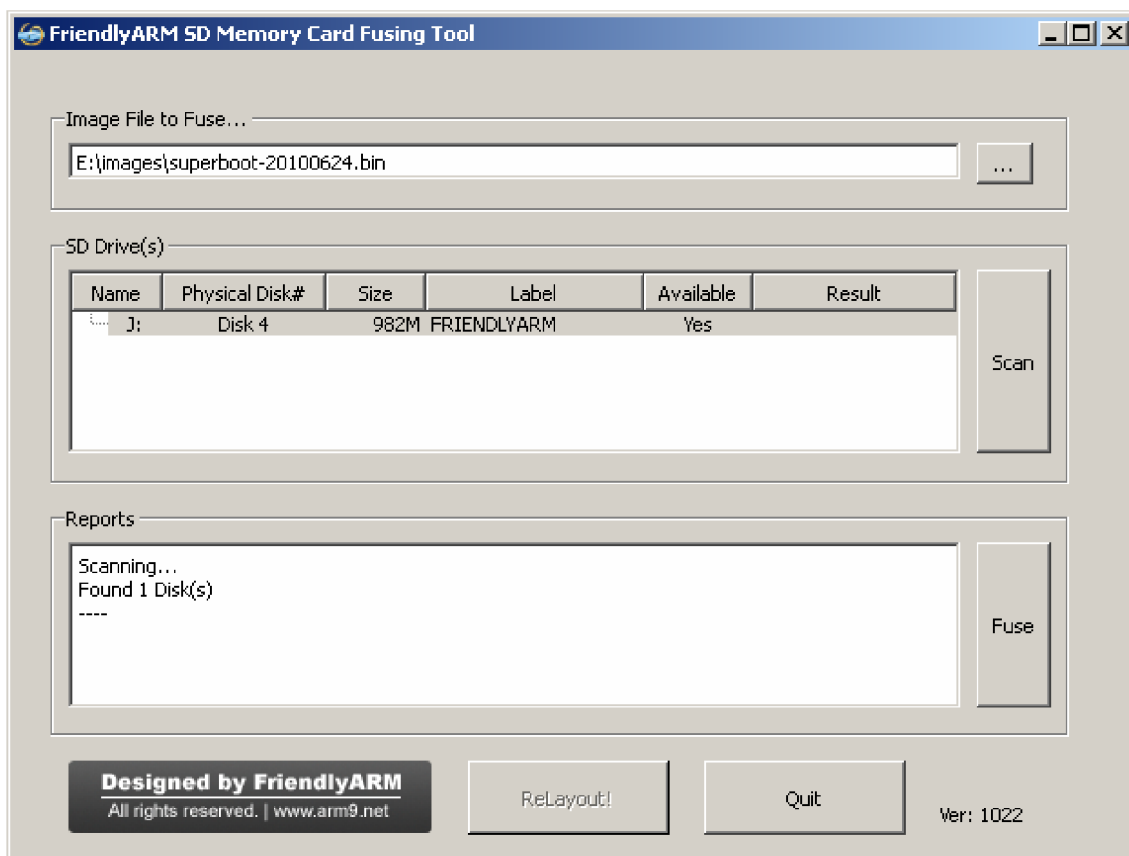
5. V programu DWN otevřeme komunikaci volbou Serial Port → Connect.
6. Nyní lze vývojový kit Mini 2440 spustit. Předtím však musíme přepnout přepínač S2 do polohy NOR (bootování BIOSu). V programu DWN se objeví nabídka BIOS.
7. Volbou „x“ provedeme formát paměti NAND.
8. Stisknutím klávesy „v“ dojde k zavedení bootloaderu. To je indikováno zprávou: „USB host is connected. Waiting a download.“
9. Nyní v programu DWN vybereme USB Port → Transmit/Restore a zvolíme soubor „supervivi\_mini2440“.

10. Stisknutím klávesy „k“ vybereme volbu, která nám umožní zavést kernel. Opět dostaneme zprávu: „USB host is connected. Waiting a download.“
11. Nyní v programu DWN vybereme USB Port → Transmit/Restore a zvolíme soubor image požadovaného jádra.
12. Stisknutím klávesy „y“ vybereme volbu pro nahrání souborového systému.
13. V programu DWN vybereme USB Port → Transmit/Restore a zvolíme soubor image požadovaného systému. Tím může být soubor „root\_qtopia.img“.
14. Nyní je nový systém připravený. Mini 2440 vypneme. Přepneme přepínač S2 do polohy NAND a znovu zapneme.

### **Zavedení operačního systému do Mini 6410:**

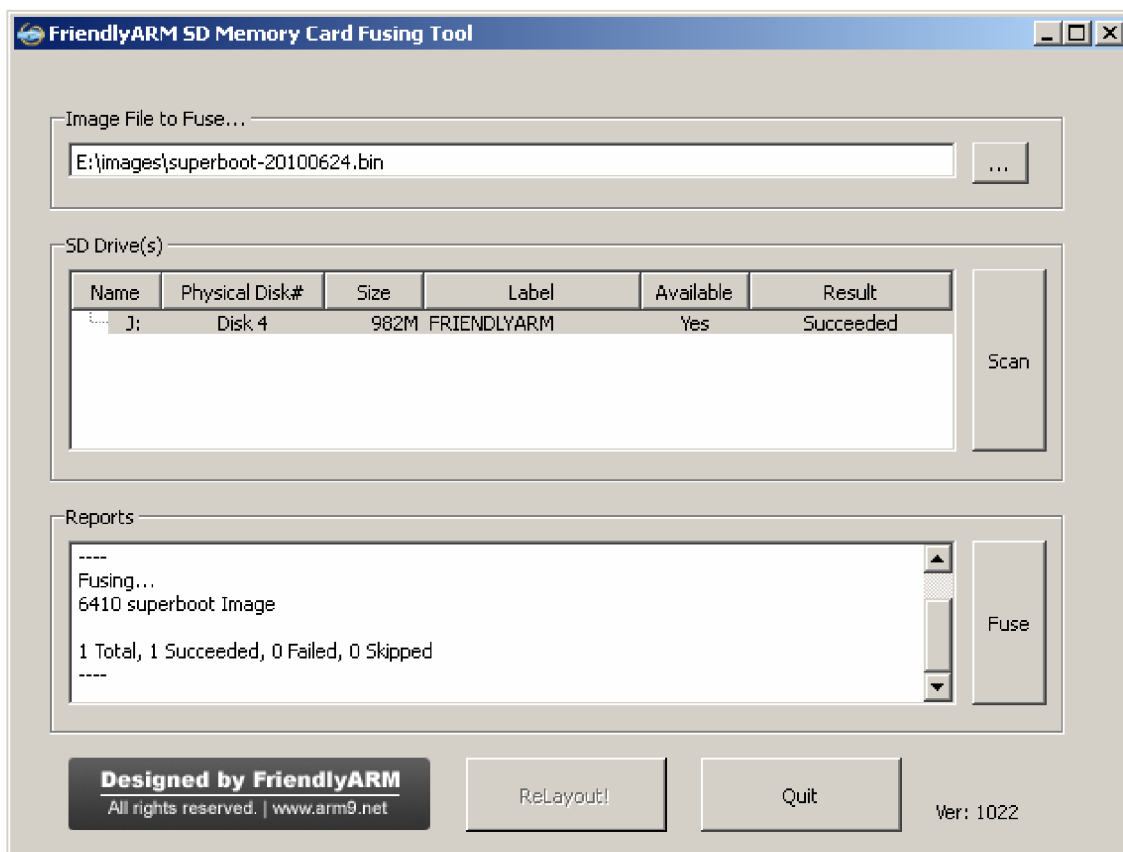
Postup zavedení nového OS je v případě Mini 6410 odlišný. Vlastní OS připravíme na PC a nahrajeme na SD paměťovou kartu. Potom již PC nepotřebujeme.

1. Je potřeba mít k dispozici SD paměťovou kartu (nejlépe 2G a větší). Tu musíme naformátovat na FAT32 a pojmenovat ji „FriendlyARM“.
2. Spustit program SD-Flasher.exe (ve Win Vista a Win7 jako správce), který je na DVD ve složce tools. V položce „Image File to Fuse“ zadat cestu k souboru „superboot-20100624.bin“. Tento soubor je ve složce images na DVD.



**Obrázek 2:** Program pro úpravu SD karty.

3. V programu SD-Flasher kliknout na tlačítko Scan. V položce SD Drivers by se měla objevit připravená SD karta. Je-li položka Available „Yes“, stačí jen kliknout na Fuse. Proběhne-li vše korektně, bude status jako na následujícím obrázku. V případě, kdy je položka Available „No“, je potřeba kliknout na ReLayout! a potvrdit výzvu.



**Obrázek 3:** Program pro úpravu SD karty – výpis úspěšné fúze.

- Po nainstalování superboot není na SD kartě nic vidět. To, zda je superboot na SD kartě nainstalován, se pozná následovně:

K vypnutému kitu připojíme SD kartu, přepínač S2 přepneme do polohy SDBOOT a zapneme kit vypínačem S1. V případě, že je karta prázdná, objeví se na displeji barevné šmouhy a kit se chová jako bez softwaru. V případě, že je superboot nainstalován na SD kartě, dioda LED1 bliká a displej zhasne.

- Nyní stačí na SD kartu nakopírovat složku „images“ včetně jejího obsahu. V této složce se nalézá soubor „FriendlyARM.ini“ a čtveřice složek, z nichž každá obsahuje připravený operační systém. Nebo sem můžeme nakopírovat vlastní systém. Podle toho, jaký systém chceme na kit instalovat musíme upravit soubor „FriendlyARM.ini“. Třetí řádek v tomto souboru specifikuje systém. Upravíme jej takto:

Pro instalaci Linuxu:

OS= Linux

Pro instalaci Androidu:

OS= Android

Pro instalaci Windows CE6:

OS= CE6

Poslední možností je vyzkoušet systém Ubuntu. Ten se přímo na kit neinstaluje, nevešel by se do paměti NAND, stačí jej spustit z SD karty. Je nutné upravit dva řádky ve „FriendlyARM.ini“ následovně:

Action= Run

OS= Ubuntu

6. Po úpravě ini souboru jej stačí uložit a odebrat SD kartu z PC. Potom kartu zasuneme do vypnutého kitu a přepneme vypínač S2 do polohy SDBOOT. Kit zapneme. Po krátké chvilce by měl bzučák pípnout a oznámit zahájení flashování paměti. Průběh flashování je indikován LED diodami. Po ukončení flashování bzučák dvakrát pípne a na LED diodách je tento stav indikován běžícím světlem. Nyní můžeme kit vypnout vypínačem S1, vyjmout SD kartu a přepínač S2 přepnout do polohy NAND.
7. Po zapnutí kitu se na displeji objeví barevné šmouhy a vše může vypadat tak, že došlo k chybě. Nemusí tomu tak být, stačí počkat asi půl minuty a systém začne nabíhat korektně.

### **Vytvoření a spuštění jednoduché aplikace v předdefinovaném systému Linux:**

1. V OS Linux (na PC) nejprve připravíme cross compilátor pro ARM procesory.
  - 1.1. Na přiloženém DVD ve složce Linux nalezneme soubor „arm-gcc-4.3.2.tgz“ nebo „arm-gcc-4.4.1-20100409.tar.gz“. Tento soubor obsahuje kompilátor. Je potřeba ho rozbalit do kořenového adresáře. Jednotlivé soubory by se měly nalézat v: usr/local/arm/4.3.2 nebo usr/local/arm/4.4.1.
  - 1.2. Dále je potřeba nastavit cestu v „bashrc“ souboru. V Terminálu zadáme příkaz: „sudo gedit ~/.bashrc“. Po zadání administrátorského hesla se otevře bashrc soubor pomocí programu gedit. Ze souboru nic nemažeme, pouze na konec přidáme: „PATH=\$PATH:/usr/local/arm/4.3.2/bin“ nebo „PATH=\$PATH:/usr/local/arm/4.4.1/bin“. Tento řádek odpovídá cestě do složky bin souborů, ve které se nalézá soubor „arm-linux-gcc“. Před ukončením geditu soubor uložíme. Po restartu PC můžeme do Terminálu zadat příkaz „echo \$PATH“. Měli bychom obdržet výpis, ve kterém bude i cesta do výše zmíněné složky. Nyní můžeme vyzkoušet, zda je arm-linux-gcc kompilátor připraven. Do Terminálu zadáme příkaz „arm-linux-gcc -v“. Po odeslání příkazu bychom měli dostat rozsáhlý výpis o programu. Tento výpis končí údajem o verzi. Například: gcc version 4.4.1 (Sourcery G++ Lite 2009q3 67).
2. Nyní vytvoříme aplikaci, která na displej vypíše „hello, FriendlyARM!“.
  - 2.1. Vytvoříme textový soubor se jménem „hello.c“. Soubor bude obsahovat:

```
#include <stdio.h>
int main(void) {
    printf("hello, FriendlyARM!\n");
    return 0;
}
```

- 2.2. Soubor zkompilujeme pomocí příkazu: „arm-linux-gcc -o hello hello.c“. Ve složce, kde se nacházel hello.c a ve které jsme provedli kompilaci se procesem kompilace vytvořil soubor hello. Tento soubor by měl být spustitelný přes konzoli na kitu.

3. Spuštění vytvořeného programu.
  - 3.1. Vytvořený soubor hello nakopírujeme na SD kartu a tu vložíme do kitu. Spustíme aplikaci „Terminál“ (Shell, konzole,...).
  - 3.2. Přesuneme se do složky se souborem hello (příkazem: „cd sdcard“) a spustíme pomocí příkazu „./hello“. Na displeji by měl být vypsán text „hello, FriendlyARM!“.

### **Programy využívající HW prostředky kitu Mini 2440 a 6410:**

Podobným způsobem lze vytvářet programy pro Linux (stejně jako pro Win) a kompilovat je pro příslušnou platformu. Mnohdy stačí vzít zdrojový soubor již funkčního Linuxového programu a překompilovat jej v arm-linux-gcc.

1. Chceme-li přistupovat k HW prostředkům, tak stejně jako v Linuxu na PC přistupujeme k HW prostředkům, které jsou dostupné ve složce dev/ v kořenovém adresáři. Konkrétně můžeme používat:

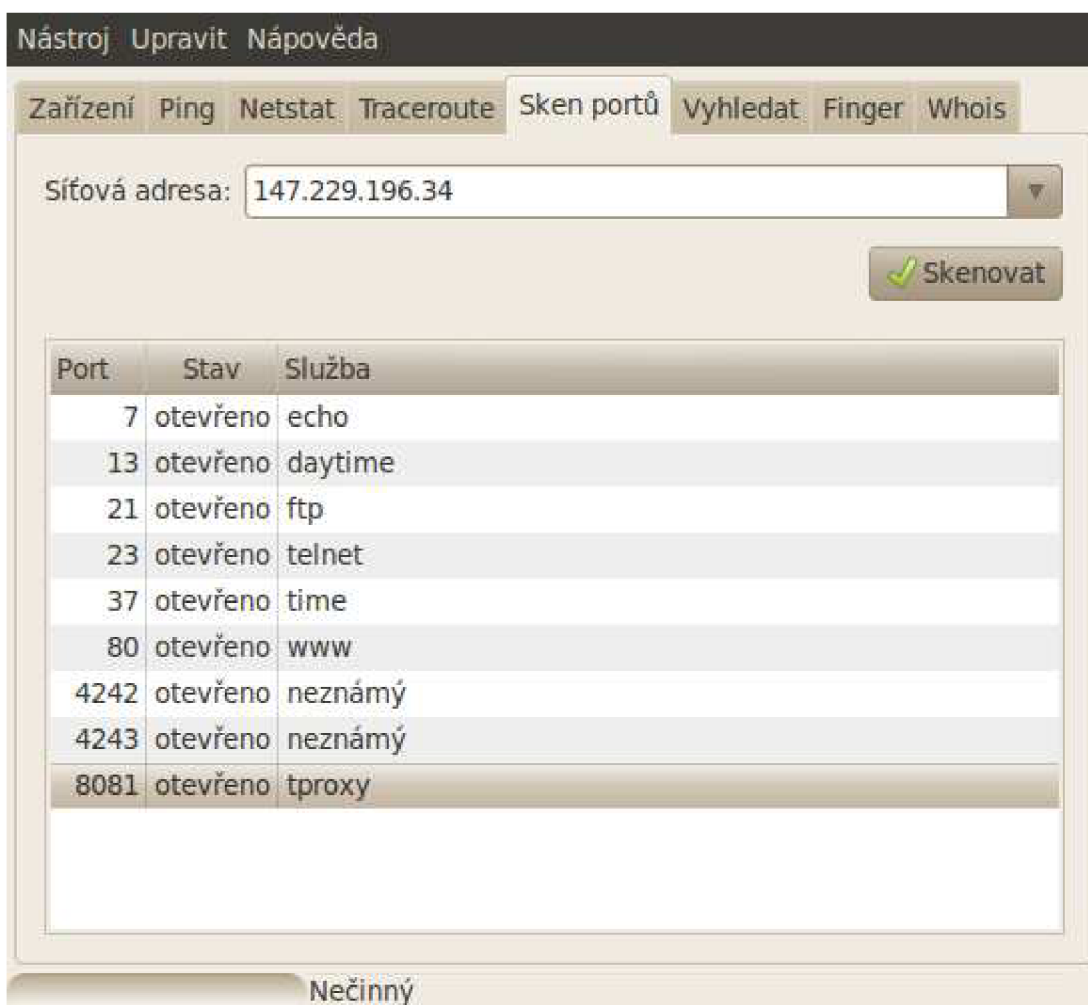
/dev/pwm	-	pwm bzučák na kitu
/dev/leds	-	led diody na kitu
/dev/buttons	-	tlačítka 1-6 na kitu (pouze u 2440)
/dev/buttons	-	tlačítka 1-8 na kitu (pouze u 6410)
/dev/ttySAC0	-	UART0
/dev/ttySAC1	-	UART1
/dev/ttySAC2	-	UART2
/dev/ttySAC3	-	UART3

a další ve složce dev.

2. Na přiloženém CD je kompletní projekt linux\_hw. Tento projekt je určený pro Mini 6410 (obsahuje 8 tlačítek) a pro Mini 2440 by musel být upraven na 6 tlačítek. Projekt je psán v editoru Code::Blocks, ale pro arm musí být zkompilován pomocí: „arm-linux-gcc -o hw\_arm main.c“. Projekt je vytvořen co možná nejpřehledněji, aby jednotlivé funkce byly alespoň částečně univerzální a daly se použít pro jiné programy. Konkrétní verze programu pracuje s tlačítky, s PWM bzučákem a UART0 linkou. Po zapnutí program zašle přes UART0 dvě hlášky (115200 baud/s, 8 bit, bez parity, jeden stop bit) a potom čeká na stisk tlačítka. Jednotlivá tlačítka představují tóny základní C dur stupnice. Na displej se vypisuje příslušný tón, současně se odesílá přes UART0 a hraje na pwm bzučáku. Aplikace se ukončí pomocí CTRL+C.

## Co už na Mini 2440 a 6410 je (Linux)?

Po připojení Mini 2440 nebo 6410 do sítě a skenu portů dostaneme:



**Obrázek 4:** Sken portů u Mini 6410.

1. Připojíme-li se vhodným klientem k portu č. 7, dostaneme zpět všechny zprávy, které odešleme (jednoduché echo).
2. Na portu 13 dostáváme odpověď v podobě času, který je na Mini právě nastavený, včetně data.
3. Na portu 21 běží klasický FTP server. Jméno pro připojení je „plg“ a heslo „plg“. Takto sice můžeme nahrávat nové programy do paměti nebo na SD kartu bez nutnosti přístupu ke kytu a manipulace s HW (odpadá věčné připojování a odpojování SD karty), ale takto nahrané programy po spuštění vypíšou pouze „Segmentation fault“. Je nutné kit vypnout a zapnout nebo programy na SD kartu nahrávat v PC a kartu přepojovat.
4. Na portu 23 běží služba telnet. Jméno uživatele je „root“ heslo není potřeba. Po připojení přes telnet můžeme s Mini 2440 nebo 6410 manipulovat stejně, jako bychom na něm spustili Terminal. Fungují všechny příkazy jako v Terminalu na Mini 2440 i Mini 6410. Můžeme spouštět aplikace, aniž bychom se kytu dotkli. S ohledem na pohodlí je tento přístup velmi komfortní. Na druhou stranu, chceme-li na kytu spustit aplikaci, která má pracovat neustále, musíme ji pustit přes Terminál

- na kitu, protože při ukončení telnetu, alias terminálu, je aplikace také ukončena.
5. Na http portu 80 běží www stránky, které předdefinoval výrobce (zajímavé, ale čínsky).
  6. Porty 4242 a 4243 mají k dispozici neznámé aplikace, u kterých jsem nedostal žádnou odpověď.
  7. Na portu 8081 je vidět aplikace proxy. Jedná se o vlastní aplikace, která byla spuštěna ještě před zahájením scanu. Tato aplikace odpovídala TCP serveru popsaném v kap. 4.5.1.

### ***Sledování systému pomocí sériové linky:***

1. Chceme-li sledovat činnost kitu Mini 2440 nebo 6410 pomocí sériové linky, musíme ji nastavit na rychlost 115200 baud/s, 8 bitů, 1 stop bit a bez parity.
2. Již při startu kitu dostáváme komplexní výpis. Potom jsme vyzváni: „Please press enter to activate this console.“ Po odetřetí můžeme s kitem pracovat na PC tak, jako bychom měli konzoli (Terminal) spuštěnou přímo na kitu. V podstatě je to stejné, jako když se připojíme přes službu Telnet popsanou výše.



## Příloha 2 – AT příkazy pro OWS451

Tato příloha vychází z [14] a popisuje nejdůležitější AT příkazy obsažené ve WiFi modulu OWS451.

### **Standardní AT příkazy:**

#### **AT režim:**

Vstup modulu do AT režimu.

AT<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

#### **Výpis dostupných AT příkazů:**

Vypíše dostupné příkazy.

AT\*<CR>

Kladná odpověď:

<CR><LF><cmd1><CR><LF><cmd2>...<CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

#### **ATZ:**

Příkaz bez jakékoliv funkce, je dostupný s ohledem na zpětnou kompatibilitu.

ATZ<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

#### **AT&F:**

Obnovení továrního nastavení.

AT&F<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

#### **AT&F0:**

Obnovení továrního nastavení (stejně jako předchozí).

AT&F0<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

#### **AT&F1:**

Obnovení statických parametrů.

AT&F1<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### AT echo off:

Vypnutí zpětného zasílání přijatých znaků v AT režimu.

ATE<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### AT echo on/off:

Vypnutí nebo zapnutí zpětného zasílání přijatých znaků v AT režimu.

ATE<echo\_on><CR>

**Tabulka 1:** ATE.

Parametr:	Hodnota:	Popis:
echo_on	integer 0 1	echo on/off echo vypnuto echo zapnuto

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o echu:

ATE?

Kladná odpověď: <CR><LF>echo\_on<CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### ATS2:

Nastavení znaku pro přechod z datového režimu do AT režimu.

ATS2<esc\_znak><CR>

**Tabulka 2:** ATS2.

Parametr:	Hodnota:	Popis:
esc_znak	integer 0...255	esc znak implicitně '/'

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející hodnotu znaku pro přechod do AT režimu:

ATS2?

Kladná odpověď: <CR><LF> esc\_znak <CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### ATS3:

Nastavení znaku pro ukončení AT příkazu.

ATS3<line\_znak><CR>

**Tabulka 3: AT3.**

Parametr:	Hodnota:	Popis:
line_znak	integer 0...127	line znak implicitně 13 (CR)

Kladná odpověď: < line\_znak ><LF>OK< line\_znak ><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející hodnotu znaku pro ukončení AT příkazu:

AT3?

Kladná odpověď: <CR><LF> line\_znak <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### **ATS5:**

Nastavení znaku pro smazání jednoho znaku.

ATS5<back\_znak><CR>

**Tabulka 4: AT5.**

Parametr:	Hodnota:	Popis:
back_znak	integer 0...127	back znak implicitně 8 (BS)

Kladná odpověď: < line\_znak ><LF>OK< line\_znak ><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející hodnotu znaku pro smazání jednoho znaku:

AT5?

Kladná odpověď: <CR><LF> back\_znak <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### **Příkazy linkové vrstvy:**

#### **Autorizační režim:**

Nastavení autorizačního režimu.

AT\*AGAM=<amode>,< uložení ><CR>

**Tabulka 5: AT\*AGAM.**

Parametr:	Hodnota:	Popis:
amode	integer 0 1 2 3 4	režim vypnuto (implicitní hodnota) Shared WPA/WPA2 PSK LEAD PEAP
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>  
 Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o autorizačním režimu:

AT\*AGAM?

Kladná odpověď: <CR><LF>\*AGAM:<amode><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Šifrovací režim:

Nastavení šifrovacího režimu.

AT\*AGEM=<emode>,< uložení ><CR>

**Tabulka 6:** AT\*AGEM.

Parametr:	Hodnota:	Popis:
emode	integer	režim
	0	vypnuto (implicitní hodnota)
	1	WEP64
	2	WEP128
	3	TKIP
uložení	4	AES/CCMP
	integer	uložení nastavení
	0	neukládat
	1	ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o šifrovacím režimu:

AT\*AGEM?

Kladná odpověď: <CR><LF>\*AGEM:<emode><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Ochranný režim:

Nastavení ochranného režimu.

AT\*AGSM=<smode>,< uložení ><CR>

**Tabulka 7:** AT\*AGMS.

Parametr:	Hodnota:	Popis:
smode	integer	režim
	0	bez ochrany (implicitní hodnota)
	1	Shared – WEP64
	2	Shared – WEP128
	3	WPA – PSK – TKIP
	4	WPA2 – PSK – AES/CCMP
	5	LEAP – WPA2
6	LEAP – WEP128	

uložení	integer 0 1	uložení nastavení neukládat ukládat
---------	-------------------	---

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o ochranném režimu:

AT\*AGSM?

Kladná odpověď: <CR><LF>\*AGSM:<smode><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Operační režim:

Nastavení přístupového režimu (ad-hoc nebo infrastructure).

AT\*AGOM=<omode>,< uložení ><CR>

**Tabulka 8:** AT\*AGOM.

Parametr:	Hodnota:	Popis:
omode	integer 1 2	režim infrastructure (implicitní hodnota) ad-hoc
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o operačním režimu:

AT\*AGOM?

Kladná odpověď: <CR><LF>\*AGOM:<omode><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Autorizační/kódovací klíč:

Nastavení autorizačního/kódovacího klíče s indexem 0.

AT\*AGFP=<klíč>,< uložení ><CR>

**Tabulka 9:** AT\*AGFP.

Parametr:	Hodnota:	Popis:
klíč	string	hodnota klíče
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Autorizační/kódovací klíč s indexem:

Nastavení autorizačního/kódovacího klíče s indexem.

AT\*AGFPWI=<index>,<klíč>,< uložení ><CR>

**Tabulka 10:** AG\*AGFPWI.

Parametr:	Hodnota:	Popis:
index	integer 1 2 3 4	hodnota indexu 1 2 3 4
klíč	string	hodnota klíče
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Aktivace autorizačního/kódovacího klíče:

Aktivace autorizačního/kódovacího klíče.

AT\*AGAFP=<index>,< uložení ><CR>

**Tabulka 11:** AT\*AGAFP.

Parametr:	Hodnota:	Popis:
index	integer 1 2 3 4	hodnota indexu 1 (implicitní hodnota) 2 3 4
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuálním klíči:

AT\*AGAFP?

Kladná odpověď: <CR><LF>\*AGAFP:<index><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Uživatelské jméno:

Nastavení uživatelského jména.

AT\*AGUN=<jméno>,< uložení ><CR>

**Tabulka 12: AT\*AGUN.**

Parametr:	Hodnota:	Popis:
jméno	string	uživatelské jméno
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o uživatelském jménu:

AT\*AGUN?

Kladná odpověď: <CR><LF>\*AGUN:<jmeno><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Jméno domény:

Nastavení jména domény.

AT\*AGDN=<jméno>,< uložení ><CR>

**Tabulka 13: AT\*AGDN.**

Parametr:	Hodnota:	Popis:
jméno	string	jméno domény
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuálním uživatelském jméně:

AT\*AGDN?

Kladná odpověď: <CR><LF>\*AGDN:<jméno><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### SSID:

Nastavení SSID.

AT\*AGSSID=<ssid>,< uložení ><CR>

**Tabulka 14: AT\*AGSSID.**

Parametr:	Hodnota:	Popis:
ssid	string	jméno ssid, maximálně 32B

uložení	integer 0 1	uložení nastavení neukládat ukládat
---------	-------------------	---

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuálním SSID:

AT\*AGSSID?

Kladná odpověď: <CR><LF>\*AGSSID:<ssid><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Hodnota RSSI:

Příkaz vracející údaje o aktuálním RSSI:

AT\*AGRSS?

Kladná odpověď: <CR><LF>\*AGRSS:<rsssi><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Nastavení kanálu:

Nastavení kanálu.

AT\*AGCH=<číslo\_kanálu>,< uložení ><CR>

**Tabulka 15:** AT\*AGCH.

Parametr:	Hodnota:	Popis:
číslo_kanálu	integer 0 1...11 1...14 36...38	číslo kanálu automatický (implicitní hodnota)
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuálním čísle kanálu:

AT\*AGCH?

Kladná odpověď: <CR><LF>\*AGCH:<číslo\_kanálu><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Skenování sítě:

Příkaz vracející hodnoty kanálů (a dalších parametrů) sítí v dosahu.

AT\*AGSCAN?

Kladná odpověď: <CR><LF>\*AGSCAN:<bssid>,< omode > , <ssid>,  
<číslo\_kanálu>,<rsssi>,< emode > ,<informace>  
<CR><LF> OK<CR><LF>



Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Tabulka 16:** Význam jednotlivých parametrů AT\*AGSCAN?.

Parametr:	Typ:	Popis:
bssid	mac adresa	mac adresa přístupového bodu
omode	integer 1 2	režim infrastructure ad-hoc
ssie	string	SSID sítě
číslo_kanálu	integer	číslo kanálu, které daná síť používá
rsi	integer	síla signálu
emode	integer 0 1 2 3	režim vypnuto WEP WPA WPA2/RSN
informace	string	informace o použitém WEP

### Nastavení přenosové rychlosti a linkové adaptace:

AT\*AGRTE=<přenosová\_rychlost>,<link\_adaptace>,< uložení ><CR>

**Tabulka 17:** AT\*AGRTE.

Parametr:	Hodnota :	Popis:
přenosová_rychlost	integer 1 2 3 4 5 6 7 8 9 10 11 12  13 14 15 16 17 18 19 20	číslo kanálu 1 Mbit 2 Mbit 5,5 Mbit 6 Mbit 9 Mbit 11 Mbit 12 Mbit 18 Mbit 24 Mbit (implicitní hodnota) 36 Mbit 48 Mbit 54 Mbit (u cB-OWS451) lze nastavit: 6.5 Mbit (MSC0) 13 Mbit (MSC1) 19.5 Mbit (MSC2) 26 Mbit (MSC3) 39 Mbit (MSC4) 52 Mbit (MSC5) 58.5 Mbit (MSC6) 65 Mbit (MSC7)

link_adaptace	integer 0 1	nastavení přizpůsobování přenosové rychlosti vypnuto zapnuto (max. rychlost viz implicit. hodnota)
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuální přenosové rychlosti (nastavené):

AT\*AGRTE?

Kladná odpověď: <CR><LF>\*AGRTE:<přenosová\_rychlost>,<link\_adaptace><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Seznam kanálů:

Zapsání kanálů, které může WiFi modul používat.

AT\*AGCL=<ch1>,<ch2>,...<ch14>,<uložení><CR>

**Tabulka 18:** AT\*AGCL.

číslo_kanálu	integer 0 1...11 1...14 36...38	číslo kanálu automatický (implicitní hodnota)
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o povolených kanálech:

AT\*AGCL?

Kladná odpověď: <CR><LF>\*AGCL:<ch1>,<ch2>,...<ch14><CR><LF>  
OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Režim napájení:

Volba režimu napájení.

AT\*AMPM=<power\_mode>,<uložení><CR>

**Tabulka 19: AT\*AGCL.**

power_mode	integer 1 2 3	číslo kanálu online režim režim spátku (implicitní hodnota) stop režim
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o povolených kanálech:

AT\* AMPM?

Kladná odpověď: <CR><LF>\*AMPM:< power\_mode><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### **Příkazy síťové vrstvy:**

#### **Nastavení IP adresy:**

Nastavení IP adresy, masky podsítě a výchozí brány.

AT\*ANIP=<ip\_adresa>,<maska\_podsítě>,<výchozi\_brana>,<uložení><CR>

**Tabulka 20: AT\*ANIP.**

Parametr:	Hodnota:	Popis:
ip_adresa	ip_adresa	IP adresa, implicitně: 192.168.0.99
maska_podsítě	ip_adresa	maska podsítě, implicitně: 255.255.0.0
výchozí_brána	ip_adresa	IP adresa brány, implicitně: 192.168.0.1
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuální IP, masce podsítě a výchozí bráně:

AT\*ANIP?

Kladná odpověď:

<CR><LF>\*ANIP=<ip\_adresa>,<maska\_podsítě>,<výchozí\_brána><CR><LF>  
OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Aktivace DHCP:**

Povolení přístupu k DHCP serveru.

AT\*ANDHCP=<on>,<uložení><CR>

**Tabulka 21:** AT\*ANDHCP.

Parametr:	Hodnota:	Popis:
on	integer	nastavení
	0	pouze statická IP (implicitní hodnota)
	1	příjem IP adresy od DHCP serveru
uložení	2	modul se chová jako DHCP server
	integer	uložení nastavení
	0	neukládat
	1	ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuálním IP přístupu k DHCP serveru:

AT\*ANDHCP?

Kladná odpověď: <CR><LF>\*ANDHCP=<on><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Jméno hostitele:**

Nastavení jména hostitele.

AT\*ANHN=<hostname>,<uložení><CR>

**Tabulka 22:** AT\*ANHN.

Parametr:	Hodnota:	Popis:
hostname	string	jméno hostitele použité při dynamickém DNS
uložení	integer	uložení nastavení
	0	neukládat
	1	ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející jméno hostitele:

AT\*ANDHCP?

Kladná odpověď: <CR><LF>\*ANHN:<hostname><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Jméno serveru DNS:**

Nastavení informací o DNS serveru.

AT\*ANDNS=<dns1>,<dns2>,<uložení><CR>

**Tabulka 23:** AT\*ANDNS.

Parametr:	Hodnota:	Popis:
dns1 nebo dns2	IP adresa	IP adresa primárního a sekundárního DNS serveru
uložení	integer 0 1	uložení nastavení neukládat ukládat

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející IP adresu pro primární a sekundární DNS server:

AT\*ANDNS?

Kladná odpověď: <CR><LF>\*ANDNS:<dns1>,<dns2><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### ***Příkazy pro nastavení komunikace mezi aplikací a modulem:***

#### **Konfigurace komunikačního rozhraní UART:**

Příkaz pro nastavení UART komunikace:

AT\*AMRS=<baud\_rate>,<data\_bity>,<stop\_bity>,<parita>,<hw\_řízení>,<reserved>,<uložení><CR>

**Tabulka 24:** AT\*AMRS.

Parametr:	Hodnota:	Popis:
baud_rate	integer	nastavení přenos. rychlosti
	1	300
	2	1200
	3	2400
	4	4800
	5	9600
	6	19200
	7	38400
	8	57600 (implicitní hodnota)
	9	115200
	10	230400
	11	460800
	12	921600
	13	1382400
	14	2764800
data_bity	integer	počet datových bitů
	1	8 (implicitní hodnota)
	2	7
	3	6
	4	5
stop_bity	integer	počet stop bitů
	1	1 (implicitní hodnota)
	2	2
parita	integer	nastavení parity

	1 2 3	žádná (implicitní hodnota) lichá sudá
hw_řízení	integer 1 2	nastavení hw. řízení toku cts/rts (implicitní hodnota) žádné
reserved	integer 0	rezervováno pro použití
uložení	integer 0 1	uložení nastavení neukládat uložit

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

Příkaz vracející údaje o aktuálním nastavení UART komunikace:

AT\*AMRS?

Kladná odpověď: <CR><LF>\*AMRS<baud\_rate>,<data\_bity>,<stop\_bity>,<parita><hw\_řízení><CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### ***Příkazy datového režimu:***

#### **Vstup do datového režimu:**

AT\*ADDM<CR>

Kladná odpověď: <CR><LF>OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

#### **Počet serverů pro připojení:**

Zápis počtu vzdálených přístupů.

AT\*ADNPR=< počet\_serverů >,<uložení><CR>

**Tabulka 25:** AT\*ADNPR.

Parametr:	Hodnota:	Popis:
počet_přístupů	integer	Hodnota, jakékoliv číslo mezi 0 a odpovědí z AT*ADNRP?
uložení	integer 0 1	uložení nastavení neukládat ukládat

Příkaz vracející hodnotu vzdálených přístupů:

AT\*ADNPR?

Kladná odpověď: <CR><LF>\*ADNPR:< počet\_serverů><CR><LF>  
OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Čtení nastaveného serveru pro spojení:**

Příkaz pro čtení údajů o serveru, ke kterému se WiFi modul připojuje.

AT\*ADRDRP=<id><CR>

**Tabulka 26:** AT\*ADRDRP.

Parametr:	Hodnota:	Popis:
id	integer	hodnota, jakékoliv číslo mezi 0 a odpovědí z AT*ADM RP - 1

Kladná odpověď: <CR><LF>\* ADRDRP:<adresa>,<spojení>,<update>,<jméno><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Tabulka 27:** Význam dat z odpovědi AT\*ADRDRP.

Parametr:	Typ:	Popis:
adresa	string	adresa ve formátu: <protokol>://IPadresa:port
spojení	integer 0 1 2	doba pro spojení nepoužívá se spojení pro přenos dat neustálé spojení
update	integer	připraveno pro budoucí použití
jméno	string	jméno

**Aktivace TPC portu:**

Povolení TCP spojení a nastavení portu.

AT\*AMTL=<port>,< povolení >,<uložení><CR>

**Tabulka 28:** AT\*AMTL.

Parametr:	Hodnota:	Popis:
port	integer	číslo portu (0...65535)
povolení	integer 0 1	zakázáno (implicitní hodnota) povoleno
uložení	integer 0 1	uložení nastavení neukládat ukládat

Příkaz vracející hodnotu vzdálených přístupů:

AT\*AMTL?

Kladná odpověď: <CR><LF>\*AMTL:<port>,< povolení >,<CR><LF>  
OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

**Aktivace UDP portu:**

Povolení UDP spojení a nastavení portu.

AT\*AMDS=<port>,< povolení >,<uložení><CR>

**Tabulka 29: AT\*AMDS.**

Parametr:	Hodnota:	Popis:
port	integer	číslo portu (0...65535)
povolení	integer 0 1	zakázáno (implicitní hodnota) povoleno
uložení	integer 0 1	uložení nastavení neukládat ukládat

Příkaz vracející hodnotu vzdálených přístupů:

AT\*AMDS?

Kladná odpověď: <CR><LF>\*AMDS:<port>,< povolení >,<CR><LF>

OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>

### Volba rádiového rozhraní:

Výběr 2,4 nebo 5 GHz rádiového rozhraní.

AT\*AMRD=<domain>,<uložení><CR>

**Tabulka 30: AT\*AMRD.**

Parametr:	Hodnota:	Popis:
domain	Integer 1 2 3 4 10 11 12 13 14	číslo portu World (implicitní hodnota) FCC (2,4 GHz) ETSI (2,4 GHz) TELEC (2,4 GHz) U-NII (5 GHz) World + U-NII (2,4 GHz + 5 GHz) FCC + U-NII (2,4 GHz + 5 GHz) ETSI + U-NII (2,4 GHz + 5 GHz) TELEC + U-NII (2,4 GHz + 5 GHz)
uložení	integer 0 1	uložení nastavení neukládat ukládat

Příkaz vracející hodnotu vzdálených přístupů:

AT\*AMRD?

Kladná odpověď: <CR><LF>\*AMRD:<domain><CR><LF> OK<CR><LF>

Chybová odpověď: <CR><LF>ERROR<CR><LF>



**Informační příkazy:****MAC adresa zařízení:**

Přečtení MAC adresy zařízení.

AT\*AILBA?&lt;CR&gt;

Kladná odpověď: &lt;CR&gt;&lt;LF&gt;\* AILBA:&lt;mac\_adresa&gt;&lt;CR&gt;&lt;LF&gt; OK&lt;CR&gt;&lt;LF&gt;

Chybová odpověď: &lt;CR&gt;&lt;LF&gt;ERROR&lt;CR&gt;&lt;LF&gt;

**Informace o zařízení:**

Přečtení informací o modulu.

AT\*AILVI?&lt;CR&gt;

Kladná odpověď: &lt;CR&gt;&lt;LF&gt;\* AILVI:&lt;výrobce&gt;,&lt;verze\_sw&gt;,&lt;wlan\_driver&gt;,&lt;wlan\_fw&gt;,&lt;wlan\_hw&gt;&lt;CR&gt;&lt;LF&gt; OK&lt;CR&gt;&lt;LF&gt;

Chybová odpověď: &lt;CR&gt;&lt;LF&gt;ERROR&lt;CR&gt;&lt;LF&gt;

**Tabulka 31: AT\*AILVI?.**

Parametr:	Typ:	Popis:
výrobce	string	výrobce
verze_sw	string	informace o softwaru sériového rozhraní
wlan_driver	string	verze wlan driveru
wlan_fw	string	verze wlan firmwaru
wlan_hw	string	Informace o výrobci HW

**Další informace o zařízení:**

Přečtení informací o modulu.

AT\*AILTI?&lt;CR&gt;

Kladná odpověď: &lt;CR&gt;&lt;LF&gt;\* AILTI:&lt;hlavní&gt;,&lt;vedlejší&gt;&lt;CR&gt;&lt;LF&gt;

OK&lt;CR&gt;&lt;LF&gt;

Chybová odpověď: &lt;CR&gt;&lt;LF&gt;ERROR&lt;CR&gt;&lt;LF&gt;

**Tabulka 32: AT\*AILTI?.**

Parametr:	Typ:	Popis:
hlavní	integer	typ výrobku
	1	Bluetooth zařízení
	2	WLAN zařízení
vedlejší	integer	802.15.4 zařízení
	0	označení zařízení
	1	cB-OWSPA311g cB-OWS451