

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**Mobilné telefóny ako platforma
v projektoch Imagine**

2010

Peter Nosál

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**Mobilné telefóny ako platforma
v projektoch Imagine**

Bakalárska práca

Študijný program: Aplikovaná informatika
Študijný odbor: 9.2.9 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky, FMFI, UK
Školiteľ: Mgr. Pavel Petrovič, PhD.

Bratislava 2010

Peter Nosál

Čestné prehlásenie

Čestne prehlasujem, že túto bakalársku prácu som vypracoval samostatne s použitím uvedených zdrojov.

V Bratislave dňa 11.6.2010

.....
Peter Nosál

Pod'akovanie

Na tomto mieste by som sa rád poďakoval vedúcemu práce Mgr. Pavlovi Petrovičovi, PhD. za návrh zaujímavej témy a cenné rady a pripomienky, a tiež rodine za morálnu podporu počas celého štúdia.

Peter Nosál

Zadanie bakalárskej práce

Študent preskúma problematiku programovania mobilných telefónov v jazyku Java a naimplementuje systém umožňujúci komunikáciu projektov jazyka Imagine Logo s mobilnými telefónmi, na príkladoch študent demonštruje činnosť a použitie systému.

Abstrakt

Táto bakalárska práca sa venuje problematike prepojenia mobilného telefónu s programom Imagine Logo. Úlohou je vytvoriť MIDlet, ktorý bude s Imagine Logo komunikovať cez Bluetooth. MIDlet je aplikácia v jazyku Java určená pre mobilné telefóny. Imagine Logo je vývojové prostredie pre jazyk Logo, a je tiež určené na komunikáciu s Lego Mindstorms NXT 'kockou' cez Bluetooth. Preto bude možné prepojiť mobilný telefón s NXT 'kockou'.

Kľúčové slová: mobilný telefón, Bluetooth, Java, J2ME, MIDlet, Imagine Logo, Lego Mindstorms NXT

Obsah

1. Úvod	1
1.1. Cieľ	1
1.2. Imagine Logo ako nástroj pre výuku programovania	1
1.3. Mobilné telefóny = ceruzka zajtrajška	1
2. Prehľad použitých technológií	3
2.1. Java ME	3
2.1.1. MIDlet	4
2.2. JSR (Java Specification Requests)	7
2.2.1. Bluetooth	7
2.2.1.1. JSR 82	8
2.2.1.1.1. Inicializácia Bluetooth-u	8
2.2.1.1.2. Nájdenie Bluetooth zariadení	9
2.2.1.1.3. Nájdenie Bluetooth služieb	12
2.2.1.1.3.1. UUID	12
2.2.1.2. Bluetooth protokoly	14
2.2.1.2.1. Object Exchange protokol	15
2.2.1.2.1.1. OBEX interface ClientSession	16
2.2.1.3. Bluetooth profily	17
2.2.1.3. Bluetooth master zariadenie	18
3. Návrh	21
3.1. Scenáre použitia	21
3.2. Komponenty systému	23
3.2.1. Chat	24
3.2.2. Fotoaparát	24
3.2.3. Joystick	24
3.2.4. Ovládanie mobilného telefónu z Imagine	24
4. Implementácia	26
4.1. Spojenie MICC a MICS	26
4.2. Komunikácia medzi MICC a MICS	27
4.2.1. Protokol komunikácie z MICC do MICS	27
4.2.2. Protokol komunikácie z MICS do MICC	28
4.3. Komunikácia medzi MICS a Imagine Logo	29
4.4. Podpisovanie MIDlet-u	30
5. Experimentálne overenie, príklady použitia	31
5.1. Inštalácia aplikácie MICC	31
5.2. Spustenie aplikácie MICC a spojenie so serverom	32
5.2.1. Aplikácia MICC – Chat	34
5.2.2. Aplikácia MICC – Fotoaparát	35
5.2.3. Aplikácia MICC – Joystick	36
6. Záver	37
7. Literatúra	38

1. Úvod

1.1. Cieľ

Cieľom tejto bakalárskej práce je vytvoriť aplikáciu určenú pre mobilné telefóny, ktorá umožní spojenie s programom Imagine Logo cez Bluetooth. Mobilný telefón takto bude môcť komunikovať s Imagine, posielat' si navzájom textové správy alebo rôzne druhy príkazov. Práca tiež využíva fotoaparát mobilného telefónu na zachytávanie obrázkov. Aplikácia v mobilnom telefóne bude cez Imagine taktiež môcť ovládať NXT robota, programovateľnú lego stavebnicou.

1.2. Imagine Logo ako nástroj pre výuku programovania

Imagine Logo je prostredie pre programovací jazyk Logo. Logo je funkcionálny programovací jazyk, pri vývoji ovplyvnený jazykom Lisp, ktorému je aj podobný syntaxou. V Logo sa dá precvičovať korytnačia grafika, teda programovanie vektorovej grafiky používaním kurzora (korytnačky) na karteziánskej súradnicovej sústave. Logo taktiež umožňuje prácu so zoznamami, súbormi alebo rekurziu. Pre túto prácu je dôležité, že umožňuje komunikáciu cez sériový port, a taktiež sa dá spojiť cez Bluetooth s NXT kockou.

1.3. Mobilné telefóny = ceruzka zajtrajška

Mobilné telefóny sú v súčasnosti veľmi rozšírené. Prvý mobilný telefón uzrel svetlo sveta v roku 1973. Veľmi sa líšil od súčasných mobilných telefónov, vážil dva kilogramy a dalo sa na ňom iba vyťukať telefónne číslo a uskutočniť hovor. Mobilné telefóny ako ich poznáme dnes sa môžu datovať až od roku 1991 príchodom SIM karty. Dnešné telefóny umožňujú okrem uskutočňovania hovorov a posielania správ, taktiež pripojenie na internet, hranie hier, využívanie multimédií ako počúvanie alebo nahrávanie hudby, fotografovanie alebo nahrávanie videa. Dôležité pre túto prácu je, že umožňujú spojenie cez Bluetooth. Pre vývojárov existuje veľa možností ako vytvárať aplikácie pre

mobilné telefóny. Podporované sú programy napísané v jazykoch C, Java, Python a mnoho ďalších. Aplikácie v tejto práci sú napísané v Java SE a Java ME.

2. Prehľad použitých technológií

2.1. Java ME

Java ME, taktiež označované J2ME, je skratka pre Java Platform, Micro Edition. Používa sa pri tvorbe aplikácií pre mobilné telefóny a tzv. “embedded systems”, čiže systémy určené pre konkrétne zariadenie, napr. digitálne hodinky, MP3 prehrávač, práčku alebo umývačku riadu. V Java ME sa programuje podobne ako v klasickej Java Standard Edition (Java SE), ale má obmedzenú množinu knižníc. Podľa toho, nakoľko obmedzená je táto množina, sa Java ME delí na dve skupiny: Connected Limited Device Configuration (CLDC) a Connected Device Configuration (CDC). CLDC obsahuje len základnú množinu knižníc a vyvíjajú sa v nej aplikácie pracujúce na zariadeniach s nízkou spotrebou energie. CDC obsahuje takmer všetky knižnice ako Java SE, okrem tých, ktoré sú potrebné na prácu s graphical user interface (GUI). Pre mobilné telefóny je určené CLDC. Podľa toho, pre aké zariadenie je CLDC určené, sa ďalej delí na profily. Najbežnejší je Mobile Information Device Profile (MIDP) vo verzii 1.0, určený hlavne pre mobilné telefóny. K CLDC pridáva niekoľko knižníc na prácu s GUI. [1]

Prehľadné rozdelenie:

Java

- **Micro Edition (ME)**
 - o **Connected Limited Device Configuration (CLDC)**
 - **Mobile Information Device Profile (MIDP)**
 - Information Module Profile (IMP)
 - o Connected Device Configuration (CDC)
 - Foundation Profile
 - Personal Basis Profile
 - Personal Profile
- Standard Edition (SE)
- Enterprise Edition (EE)

Tak ako pre Java SE, aj pre Java ME existujú rôzne vývojové prostredia. Najznámejšie sú NetBeans a Eclipse. Táto práca je naprogramovaná v prostredí Eclipse. Na to, aby sa v Eclipse dal vytvoriť a spustiť program napísaný v Java ME, je potrebné mať okrem JRE (Java Runtime Environment) a JDK (Java Development Kit) nainštalované aj ďalšie komponenty.

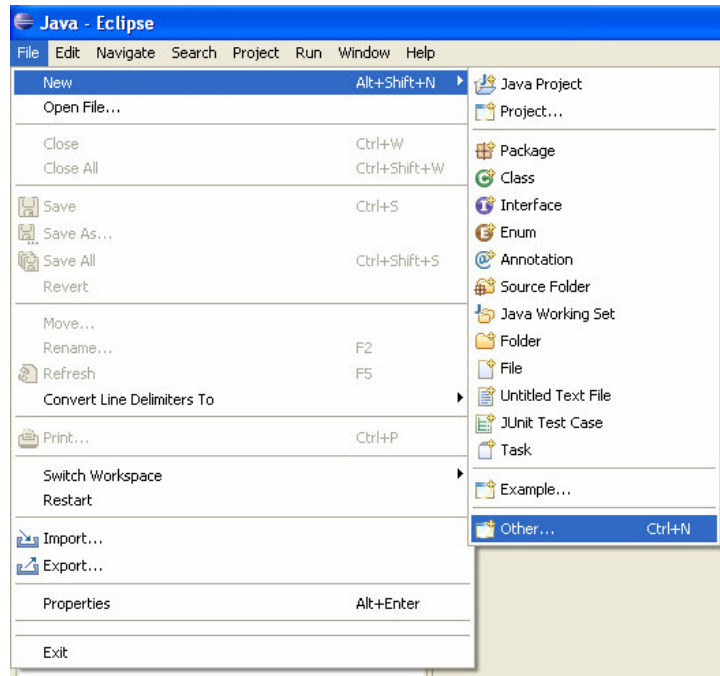
Ako prvé je potrebné do Eclipse nainštalovať J2ME. To sa robí v Eclipse cez horné menu, *Help* → *Install New Softwar*. Tým sa otvorí okno *Available Software*. Do textového pola *Work with* je potrebné zadať adresu <http://www.eclipseme.org/updates>. Zobrazí sa možnosť pridať EclipseME. [2]

Jedna z potrebných inštalácií je Sun Java Wireless Toolkit (WTK). Je to séria pomocných programov, určených na vytváranie Java aplikácií pre mobilné telefóny. Taktiež obsahuje emulátor, cez ktorý je možné spúšťať MIDlet v PC. [3] Ďalej je potrebné nainštalovať S60 SDK. To obsahuje zvyšné potrebné komponenty. [4] WTK aj SDK sa do Eclipse pridávajú cez menu *Window* → *Preferences*. Tým sa otvorí okno *Preferences*, kde je potrebné vybrať *J2ME* → *Device Management* a stlačiť tlačítko *Import*. V novootvorenom okne treba nájsť priečinok s nainštalovaným WTK a SDK a stlačiť tlačítko *Finish*. [2]

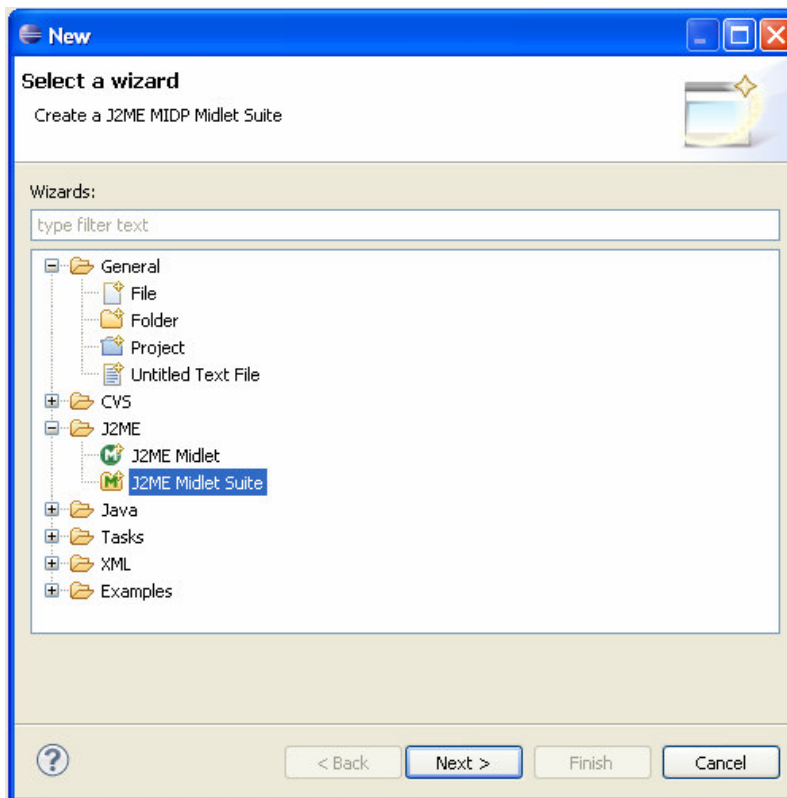
Eclipse tiež nemá knižnice na prácu s Bluetooth pre J2SE. Preto je potrebné použiť iné dostupné riešenia, ako napr. knižnica Bluecove. [5]

2.1.1. MIDlet

MIDlet je aplikácia napísaná v Java ME využívajúca MIDP a preto je určená hlavne pre mobilné telefóny. Je to obyčajná Java trieda, ktorá je podtriedou triedy MIDlet. V Eclipse sa nový MIDlet projekt nevytvára ako obyčajný Java projekt. Ako ukazuje obr. 1, je potrebné ísť cez *File* → *New* → *Other* a z novootvoreného okna na obr. 2 vybrať J2ME Midlet Suite.



Obr. 1 Vytváranie nového MIDlet projektu v Eclipse, krok 1

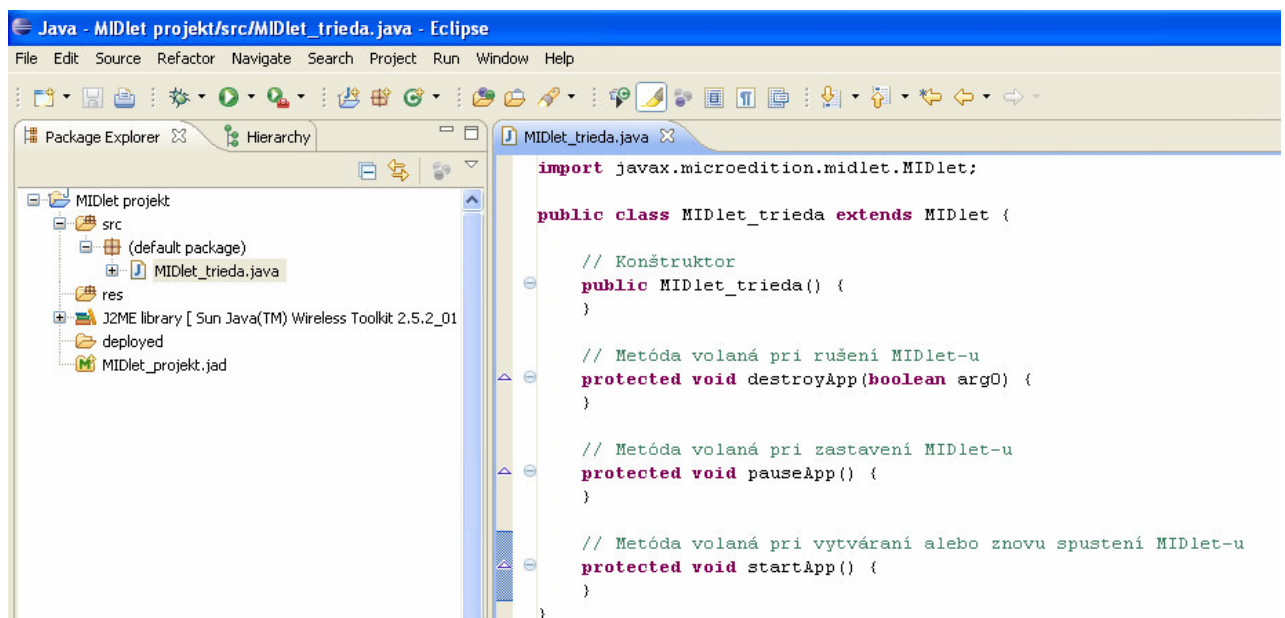


Obr. 2 Vytváranie nového MIDlet projektu v Eclipse, krok 2

Tým sa otvorí nové okno, kde stačí napísať názov projektu. Trieda, ktorá je podtriedou triedy MIDlet sa vytvára podobne, ale v kroku 2 treba vybrať J2ME Midlet. Pokiaľ je v projekte viac tried, treba mať jednu hlavnú, ktorá bude potriedou MIDlet a ostatné budú obyčajné triedy. Tieto zvyšné triedy je potrebné vytvárať klasickým spôsobom, teda nie ako je znázornené na obr. 2.

Existujú tri rôzne stavy, do ktorých môže MIDlet dospieť. Sú to stavy:

- zastavený (paused): MIDlet už bol vytvorený, ale nie je práve aktívny resp. otvorený.
- aktívny (active): MIDlet je vytvorený a aktívny.
- zrušený (destroyed): MIDlet prešiel zo stavu aktívny alebo zastavený do stavu vypnutý.



Obr. 3 Príklad MIDlet triedy

Aby MIDlet mohol byť spustený v mobilnom telefóne, je potrebné ho exportovať do formátu .jar. To sa spraví kliknutím pravého tlačítka myši na daný *MIDlet projekt* → *J2ME* → *Create Package*. Do adresára *deployed* sa uloží daný .jar súbor, ktorý už treba iba skopírovať do mobilného telefónu a spustiť.

2.2. JSR (Java Specification Requests)

Rozširovanie funkcionality Javy sa deje cez proces zvaný Java Community Process (JCP). Na tomto procese sa zúčastňujú expertné skupiny z rôznych spoločností, napr. Cisco Systems, IBM, Nokia Corporation a iné. Tieto skupiny navrhnu Java Specification Requests (JSR), dokument popisujúci špecifikáciu a API pre nové rozšírenie. Pre moju prácu sú dôležité nasledovné JSR:

- JSR 37: Mobile Information Device Profile (MIDP) 1.0: špecifikácia API na programovanie v Java ME
- JSR 82: Bluetooth Java API: špecifikácia API pre podporu Bluetooth v Java ME
- JSR 135: Java Mobile Media API (MMAPI): špecifikácia API pre podporu médií, ako prehrávanie hudby, videa a zachytávanie obrázkov

2.2.1. Bluetooth

Bluetooth je proprietárna technológia a komunikačný protokol, slúžiaci na bezdrôtový prenos dát na krátke vzdialenosti. O jeho formalizovanie a inováciu sa stará Bluetooth Special Interest Group (SIG). V tejto skupine sa musí nachádzať každá spoločnosť, ktorá využíva Bluetooth vo svojich produktoch.

V súčasnosti je Bluetooth veľmi rozšírený. Používa sa najmä v mobilných telefónoch na komunikáciu s inými mobilnými telefónmi, PDA alebo headsetom, ako bezdrôtová komunikácia počítača so vstupnými a výstupnými zariadeniami (myš, klávesnica, tlačiareň a pod.).

Bluetooth pracuje v nelicencovanom pásme ISM (industrial, scientific and medical), ktoré sa používa na priemyselné, vedecké a medicínske účely. Toto pásmo má veľkosť 2.4 GHz (podobne ako Wi-Fi). Jedno zariadenie je master a môže sa naň pripojiť maximálne 7 zariadení. Pre komunikáciu musia byť zariadenia v dosahu, ale nemusia sa fyzicky "vidieť" a teda nemusia byť v tých istých miestnostiach. Rozlišujú sa tri triedy

v závislosti od toho, koľko energie vyžadujú a teda do akej vzdialenosti dokážu komunikovať.

Prehľad Bluetooth tried:

Trieda	mW	dBm	Dosah
Class 1	100 mW	20 dBm	~100 metrov
Class 2	2.5 mW	4 dBm	~10 metrov
Class 3	1 mW	0 dBm	~1 meter

Existujú tri verzie protokolu, líšiace sa hlavne v prenosových rýchlostiach.

Prehľad Bluetooth verzií:

Verzia	Prenosová rýchlosť
Verzia 1.2	1 Mbit/s
Verzia 2.0 + EDR	3 Mbit/s
Verzia 3.0 + HS	24 Mbit/s

Mobilné telefóny používajú verziu 2.0 + EDR.

2.2.1.1. JSR 82

JSR 82 je špecifikácia Bluetooth pre platformu Java. Nachádza sa v balíčku *javax*. Skladá sa z dvoch balíčkov, a to *bluetooth* a *obex*. *javax.bluetooth* obsahuje hlavne triedy umožňujúce nájdenie Bluetooth zariadení, a služieb. *javax.obex* obsahuje zase triedy na komunikáciu cez protokol OBEX.

2.2.1.1.1. Inicializácia Bluetooth-u

Na to, aby bolo možné používať zariadenie Bluetooth, je potrebné ho inicializovať. Tento proces sa skladá z niekoľkých častí, ktoré slúžia na pripravenie zariadenia na potrebnú komunikáciu. Inicializácia nie je súčasťou JSR 82, ale je za ňu

zodpovedný každý dodávateľ Bluetooth zariadenia (či už počítača alebo mobilného telefónu). Používateľ môže k nastaveniam inicializácie pristupovať cez Bluetooth Control Center (BCC), ktoré má väčšinou grafické rozhranie. Možno tu nastaviť, či ostatné zariadenia môžu toto zariadenie vyhľadávať, pripojiť sa naň, nastaviť sériové COM porty atď.

2.2.1.1.2. Nájdenie Bluetooth zariadení

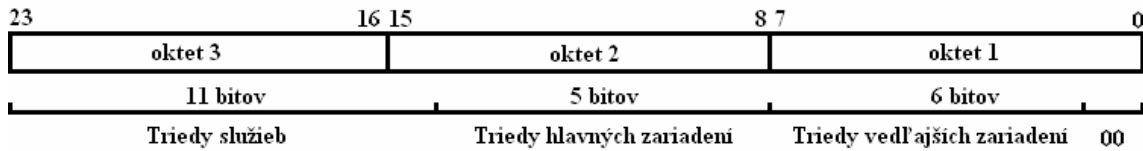
Prvý krok, ktorý je potrebné spraviť pred nadviazaním spojenia so zariadením, je vyhľadanie zariadení v dosahu. O túto činnosť sa stará trieda *javax.bluetooth.DiscoveryAgent* a interface *javax.bluetooth.DiscoveryListener*. Interface *DiscoveryListener* obsahuje štyri metódy, dve slúžiace pri vyhľadávaní zariadení a dve pri vyhľadávaní služieb.

Sú dva spôsoby na vyhľadanie zariadení. Prvý je použiť metódu *DiscoveryAgent.startInquiry()*, ktorá hľadá Bluetooth zariadenia v okolí. Jeden z jej parametrov je číslo, určujúce aké zariadenia sa majú vyhľadávať. Existujú dve možnosti:

- GIAC (General/Unlimited Inquiry Access Code): hľadané budú všetky zariadenia v okolí.
- LIAC (Limited Dedicated Inquiry Access Code): hľadané budú len zariadenia, ktoré sú v móde LIAC. Pretože sa nájde menej zariadení, je tento spôsob rýchlejší. Toto nastavenie sa používa hlavne v prípade, keď je aplikácia spustená na viacerých zariadeniach a spojenie má byť zabezpečené len medzi týmito zariadeniami.

Vždy, keď je nájdené zariadenie, je toto zariadenie vrátené metódou *DiscoveryListener.deviceDiscovered()* ako objekt triedy *RemoteDevice*. Spolu s ním je vrátený aj objekt triedy *DeviceClass*. Trieda *RemoteDevice* má metódy na zistenie mena zariadenia, Bluetooth adresy, či je zapnuté kódovanie atď.

Inštancia triedy *DeviceClass* určuje Class of Device (CoD), ktoré sa skladá z 24 bitov. Ako sú bity usporiadané a ktoré na čo slúžia je vidieť na nasledujúcom obrázku.



Obr. 4 Class of Device - formát bitov

DeviceClass má tri metódy:

- *getMajorDeviceClass()*: vráti, o akú triedu zariadenia ide.
- *getMinorDeviceClass()*: vráti podtriedu zariadenia. Každá trieda zariadenia má množinu iných podtried.
- *getServiceClasses()*: vráti aké služby zariadenie poskytuje.

Každá z týchto metód vracia číslo. Čo jednotlivé čísla znamenajú určuje špecifikácia Bluetooth Assigned Numbers Baseband.

Prehľad tried služieb. Hodnoty sú určené jednotkovým alebo nulovým bitom na 13. – 23. bite. Ak je služba podporovaná, je na danom bite 1, ak nie je, tak 0.

Bit	Triedy služieb
13	Limitovaný objaviteľný mód
14	(rezervované)
15	(rezervované)
16	Lokalizácia (identifikácia umiestnenia)
17	Pripojenie do siete (LAN, ad hoc, ...)
18	Renderovanie (tlačiareň, reproduktor, ...)
19	Zachytávanie dát (scanner, mikrofón, ...)
20	Prenos dát
21	Audio (reproduktor, mikrofón, headset, ...)
22	Telefonovanie (bezdrôtové, modem, headset, ...)
23	Informácie (WEB-server, WAP-server, ...)

Prehľad tried hlavných zariadení (major devices). Hodnoty sú určené jednotkovým alebo nulovým bitom na 8. – 12. bite, napr. $1280 = 2^8 + 2^{10}$.

Hodnota	Triedy hlavných zariadení
256	Počítač (desktop, notebook, PDA, ...)
512	Telefón (mobilný, bezdrôtový, modem, ...)
768	LAN / Network Access point
1024	Audio/Video (headset, reproduktor, stereo, vcr, ...)
1280	Periférne zariadenie (myš, joystick, klávesnica, ...)
1536	Obrazové zariadenie (tlačiareň, scanner, kamera, displej, ...)
7936	Nešpecifikované zariadenie

Podtriedy zariadení závisia od tried hlavných zariadení. Udvávajú presnejší typ zariadenia. Hodnoty sú určené jednotkovým alebo nulovým bitom na 2. – 7. bite, napr. $12 = 2^2 + 2^3$. Pre triedy hlavných zariadení telefón a počítač sú nasledujúce podtriedy:

Hodnota	Podtriedy zariadení pre telefón	Podtriedy zariadení pre počítač
0	Nešpecifikované zariadenie	Nešpecifikované zariadenie
4	Mobilný	Desktop
8	Bezdrôtový	Server
12	Smart phone	Laptop
16	Modem	Ručné PC / PDA
20	ISDN prístup	Palm sized PC / PDA
24	Nepoužíva sa	Nositeľný počítač (napr. v hodinkách)

Po skončení alebo zrušení vyhľadávania sa zavolá metóda *DiscoveryListener.inquiryCompleted()*.

Druhý spôsob vyhľadania zariadení je pomocou metódy *DiscoveryAgent.retrieveDevices()*, ktorá vráti existujúci zoznam zariadení. Keďže nemusí vykonávať vyhľadávanie, je táto metóda rýchlejšia, ale vhodná iba na pripojenie k už známym zariadeniam. Aj táto metóda má jeden z parametrov číslo, ktoré určuje spôsob vyhľadania zariadení. Tieto čísla sú reprezentované konštantami:

- CACHED: vráti zariadenia, ktoré boli nájdené pri poslednom hľadaní.

- PREKNOWN: vráti zariadenia, ktoré sú v Bluetooth Control Center v zozname spárovaných zariadení.

2.2.1.1.3. Nájdenie Bluetooth služieb

Po nájdení aspoň jedného Bluetooth zariadenia sa môže prejsť k vyhľadávaniu služieb na tomto zariadení. Služba je Bluetooth aplikácia, ktorá umožňuje na danom zariadení vykonávať rôzne úkony, poskytovať informácie alebo pracovať so systémovými prvkami. Služba môže byť implementovaná na hardvéri, softvéri alebo na oboch zároveň.

O nájdenie služieb sa stará Service Discovery Protocol (SDP). Všetky informácie o službe sú poskytnuté ako atribúty v **zázname o službe**. Každý atribút obsahuje jednu vlastnosť a je uložený ako dvojica ID atribútu a hodnota. ID atribútu je reprezentované ako 16 bitové číslo, ktoré okrem odlíšenia od ostatných atribútov identifikuje význam atribútu. Pod ID atribútom ServiceClassIDList sa nachádza jeden z dôležitých atribútov, UUID.

2.2.1.1.3.1. UUID

Universally Unique Identifier (UUID), čiže univerzálne unikátny identifikátor. Je to dátový typ, používaný na identifikáciu služieb, protokolov, profilov atď. UUID je 128 bitové číslo a je zaručené, že je vždy a všade unikátne. Existujú rôzne varianty tohto dátového typu, okrem 128 bitových, aj 16 a 32 bitové. Tieto UUID s kratšou dĺžkou v skutočnosti reprezentujú 128 bitovú hodnotu a dajú na ňu použitím vzorca previesť. Sú však vytvorené na zaberanie menej pamäti pri ukladaní a transfere. Existuje rozsah prealokovaných hodnôt 16 bitových UUID použitých pre často používané protokoly, služby a profily.

Prehľad niektorých protokolov spolu s ich UUID:

Skratka protokolu	Protokol	UUID
SDP	Service Discovery Protocol	0x0001
UDP	User Datagram Protocol	0x0002
RFCOMM	Radio Frequency Communication	0x0003
TCP	Telephony Control Protocol	0x0004
OBEX	Object Exchange	0x0008
IP	Internet Protocol	0x0009
FTP	File Transfer Protocol	0x000A
HTTP	Hypertext Transfer Protocol	0x000C
L2CAP	Logical Link Control and Adaptation Protocol	0x0100

Existujú dve rôzne riešenia hľadania služieb. Buď sa zisťuje prítomnosť konkrétnej služby alebo sa vyhľadávajú všetky služby na danom zariadení. V oboch prípadoch sa využívajú UUID. Keď sa zisťuje prítomnosť konkrétnej služby, zisťuje sa prítomnosť UUID danej služby. Pri vyhľadaní všetkých služieb je zoznam služieb vrátený ako zoznam UUID.

Ako je napísané v časti 2.2.1.1.2. nájdenie Bluetooth zariadení, ako na vyhľadávanie zariadení, tak aj na vyhľadávanie služieb slúži trieda *DiscoveryAgent* s metódou *searchServices()*. Jej atribúty obsahujú zariadenie, na ktorom sa majú služby vyhľadávať, zoznam služieb určených UUID, ktoré sa majú na danom zariadení vyhľadávať a zoznam atribútov služieb, ktoré majú byť vrátené. Interface *DiscoveryListener* tiež obsahuje dve metódy slúžiace pri vyhľadávaní služieb. Pri každom nájdení služieb je zavolaná metóda *servicesDiscovered()*. Obsahuje atribút so zoznamom vyžiadaných atribútov služby. Po skončení vyhľadávania sa zavolá metóda *serviceSearchCompleted()*, ktorá určí, či bolo vyhľadávanie úspešne dokončené, predčasne zastavené, či sa vyskytla chyba alebo dané služby neboli nájdené.

2.2.1.2. Bluetooth protokoly

Bluetooth protokoly sa spoločne nachádzajú v Bluetooth protokolovom zásobníku. Tieto zásobníky sa delia na dva druhy implementácie – viacúčelné (general purpose) a embedded (vložené priamo do daného zariadenia).

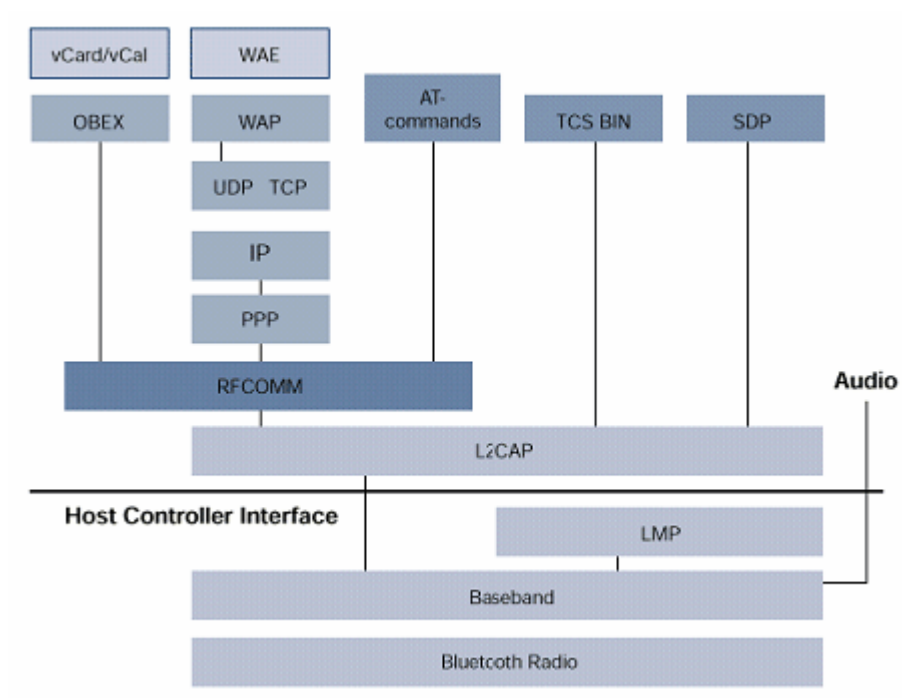
Viacúčelná implementácia sa týka operačných systémov počítača. Keďže operačné systémy v sebe nemajú zabudovanú implementáciu Bluetooth protokolového zásobníku, existuje veľa externých riešení. Prvá implementácia pre Windows bola Widcomm od spoločnosti Widcomm Inc.

Embedded implementácie sa týkajú samotných mobilných telefónov, respektíve iných zariadení. Taktiež existuje veľa riešení, napríklad operačný systém Symbian pre mobilné telefóny obsahuje implementáciu Bluetooth protokolového zásobníku v sebe.

V Bluetooth technológii existuje množstvo protokolov. Hlavné z nich sú vytvorené spoločnosťou Bluetooth SIG. Iné boli prevzaté z rôznych štandardov. Protokoly sa rozdeľujú na dve skupiny – radičové (controller) a hosťovské (host). Radičové sú implementované priamo na zariadení, ktoré obsahuje Bluetooth rádio a mikroprocesor. Hosťovské sú zase implementované v operačnom systéme zariadenia.

Na obr. 5 je znázornený Bluetooth protokolový zásobník. Radičové a hosťovské protokoly sú oddelené hrubou čiarou. Na spodnej časti je Bluetooth rádio, čiže prijímač a vysielač dát. Nad ním, stále vo fyzickej vrstve, je základné hosťovské pásmo. To sa stará o vyhľadanie zariadení v okolí, párovanie s inými zariadeniami, posielanie požiadavky na spojenie a takisto spravuje pakety. Nad základným hosťovským pásmom sa nachádza Link Manager Protocol (LMP). Tento komunikuje s rovnakým protokolom na vzdialenom zariadení, prevádzkuje vstupné a výstupné linky, ich nastavenie, pokiaľ je vyžadovaná, tak aj autentifikáciu zariadenia, atď. Nad LMP sa nachádza Host Controller Interface (HCI), ktoré predstavuje štandardnú komunikáciu medzi radičovým a hosťovským protokolovým zásobníkom. Potom nasledujú hosťovské protokoly.

Najspodnejší z nich, na ktorý sa pripájajú ostatné protokoly, je Logical Link Control and Adaptation Protocol (L2CAP). Stará sa o rozposielanie dát do vyšších protokolov, rozkladá a skladá pakety, atď. Veľkosť paketov môže byť minimálne 45 bytov a maximálne 64 kB. Ďalší dôležitý protokol je Radio frequency communications (RFCOMM). Vytvára virtuálne sériové porty, vďaka čomu nahradzuje spojenie cez kábel. RFCOMM umožňuje až 60 súbežných spojení medzi dvomi Bluetooth zariadeniami, avšak skutočný počet súbežných spojení závisí od samotného Bluetooth zariadenia. Nad RFCOMM sa nachádza najdôležitejší protokol pre túto prácu – OBEX.



Obr. 5 Bluetooth protokolový zásobník

2.2.1.2.1. Object Exchange protokol

Object Exchange (OBEX) je komunikačný protokol, ktorý definuje dátové objekty a umožňuje dvom zariadeniam výmenu týchto objektov cez Bluetooth. Pôvodne bol vytvorený pre Infrared Data Association (IrDA), čiže na prenos dát cez infračervený port, kde sa nazýva IrOBEX. Už tu slúžil na prenos rôznorodých údajov, ako napr. vizitiek, záznamov v kalendári alebo aj aplikácií. V súčasnosti sa o jeho vývoj stará

IrDA a Bluetooth SIG. Okrem Bluetooth a IrDA má uplatnenie aj v RS-232 (štandard, používaný hlavne v sériových portoch), USB a WAP.

OBEX používa model klient-server. Zariadenie disponujúce Bluetooth technológiou, ktoré chce zriadiť OBEX spojenie s iným zariadením, je klient. Zariadenie, ktoré čaká na spojovaciu požiadavku iného zariadenia, je server.

Na prácu s protokol OBEX v platforme JAVA ME existuje knižnica *javax.obex*. Obsahuje 5 rozhraní (interface) a 3 triedy.

2.2.1.2.1.1. OBEX interface ClientSession

javax.obex.ClientSession je rozhranie používané na klientskej strane aplikácie a umožňuje posielanie požiadaviek OBEX operácie. Obex operácie sú CONNECT, SETPATH, PUT, GET a DISCONNECT. Interface obsahuje pre každú z operácií rovnomennú metódu. CONNECT a DISCONNECT sa posielajú pri metódach connect() resp. disconnect(). SETPATH sa posielajú funkciou setPath(), ktorá sa využíva hlavne v File Transfer Profile (FTP), na prehľadávanie súborového systému na serveri. Operácie PUT a GET sú pre túto prácu najdôležitejšie. Metódou put() sa posielajú dáta na server a metódou get() sa prijímajú dáta zo servera.

Príklad PUT operácie:

```
void putObjectViaOBEX(ClientSession conn, HeaderSet head, byte[] obj)
    throws IOException {

    // Zadanie dĺžky dát, ktoré sa idú poslať
    head.setHeader(HeaderSet.LENGTH, new Long(obj.length));

    // Inicializácia PUT požiadavky
    Operation op = conn.put(head);

    // Otvorenie output stream-u
    OutputStream out = op.openOutputStream();

    // Poslanie dát na server
    out.write(obj);
}
```

```

    // Ukončenie operácie
    out.close();
    op.close();
}

```

Príklad GET operácie:

```

byte[] getObjectViaOBEX(ClientSession conn, HeaderSet head) throws
IOException {

    // Poslanie GET požiadavky na server
    Operation op = conn.get(head);

    // Otvorenie input stream-u
    InputStream in = op.openInputStream();

    // Čítanie dát z input stream-u
    ByteArrayOutputStream out = new ByteArrayOutputStream();
    int data = in.read();
    while (data != -1) {
        out.write((byte)data);
        data = in.read();
    }

    // Ukončenie operácie
    in.close();
    op.close();

    byte[] obj = out.toByteArray();
    out.close();

    return obj;
}

```

2.2.1.3. Bluetooth profily

Bluetooth profil je rozhranie určujúce konkrétnu funkcionality. Na to, aby aplikácia v zariadení mohla používať Bluetooth, musí zariadenie podporovať profil, ktorý aplikácia potrebuje použiť. Niektoré profily slúžia iba ako základ pre iné profily, ktoré sú založené na podobných princípoch. Celkový počet profilov evidovaný Bluetooth SIG je 28. Príklady niektorých profilov:

Serial Port Profile (SPP) je Bluetooth profil slúžiaci na vytváranie sériových portov. Na komunikáciu používa RFCOMM protokol. Je základom iných profilov, napr. FAX (profil umožňujúci komunikáciu telefónu, či už mobilného alebo klasického s

káblom, s počítačom s nainštalovaným FAX softvérom), Dial-up Networking Profile (DUN – profil umožňujúci pripojenie na Internet cez Bluetooth), a niekoľko ďalších.

Hands-Free Profile (HFP) je Bluetooth profil, ktorý má uplatnenie v automobiloch. Od roku 2002 ho automobilky začali pridávať do automobilov a od roku 2009 ho majú zabudovaný všetky automobily disponujúce Bluetooth technológiou. Používateľom umožňuje pripojenie na mobilný telefón a uskutočňovať hovor, zatiaľ čo majú samotný telefón napr. vo vrecku.

Bluetooth profily založené na protokole OBEX:

- GOEP (Generic Object Exchange Profile): slúži ako základ pre iné profily.
- OPP (Object Push Profile): slúži na posielanie objektov medzi mobilnými telefónmi, ako napr. obrázkov, virtuálnych vizitiek a iných dát.
- FTP (File Transfer Profile): slúži na prehliadanie, manipuláciu a prenos súborov a priečinkov v súborovom systéme druhého zariadenia.
- SYNCH (Synchronisation Profile): slúži na synchronizáciu dát v Personal Information Manager (PIM). PIM je typ aplikácie, ktorá funguje ako osobný organizér.
- BIP (Basic Imaging Profile): slúži na posielanie obrázkov medzi zariadeniami, pričom umožňuje zmenu veľkosti a typu obrázka. Používa sa napríklad na posielanie slide-ov na video projektor.
- BPP (Basic Printing Profile): slúži na posielanie dát (text, email, virtuálne vizitky) tlačiarňami vybavenej Bluetooth technológiou.

2.2.1.3. Bluetooth master zariadenie

Zariadenie, ktoré čaká, kým sa naň ostatné zariadenia cez Bluetooth pripoja, sa nazýva master alebo server. V tejto kapitole sú opísané kroky potrebné na inicializovanie tohto serveru.

Všetky RFCOMM spojenia sú inicializované metódou *open()* triedy *javax.microedition.io.Connector*. Keďže OBEX je tiež RFCOMM spojenie, rovnako to platí aj pre OBEX. Táto metóda má ako parameter string, ktorý sa nazýva **spojovací string** a má nasledovný formát:

```
{schéma}:{cieľ}{parametre}
```

pričom jednotlivé prvky tohto stringu znamenajú:

- schéma: skratka protokolu, ktorý sa používa, napr. btspp, btgoep.
- cieľ: sieťová adresa začínajúca “//”.
- parametre: nepovinné parametre, vo formáte “parameter=hodnota”.

Prehľad parametrov pre RFCOMM:

Názov	Popis	Možné hodnoty	Určené pre klienta alebo server
master	Či je zariadenie master (server) alebo slave (klient)	true, false	obidva
authenticate	Či zariadenie, ktoré sa chce pripojiť, musí byť overené	true, false	obidva
encrypt	Či má byť spojenie kódované	true, false	obidva
authorize	Či všetky spojenia k tomuto zariadeniu majú byť autorizované	true, false	server
name	Atribút ServiceName v zázname o službe (opísanej v kapitole 2.2.1.1.3. Nájdenie Bluetooth služieb)	validný String	server

Všetky parametre sú nepovinné, pri zadaní iného parametra sa vyhodí výnimka. Bezpečnostné parametre *authenticate*, *encrypt* a *authorize* majú pri nedefinovaní hodnotu *false*, pokiaľ iný parameter nevyžaduje, aby boli *true*. Napr. ak je *encrypt=true* a *authenticate* nie je definované, je *authenticate* nastavené na *true*, pretože kódovanie vyžaduje overenie. Niektoré kombinácie nie sú povolené, napr. “*authenticate=false*” a zároveň “*authorize=true*”.

Pre server časť {cieľ} začína “//localhost”, nasledované dvojbodkou a 128 bitovým UUID.

Príklad spojovacieho stringu pre server:

1. Zariadenie môže byť buď Master alebo Slave:

```
“btspp://localhost:102030405060708090A1B1C1D1E10073; name=Print  
Server; master=false”
```

2. Všetka komunikácia smerujúca na server musí byť overená a autorizovaná:

```
“btspp://localhost:1231242432434A4AA3B056104AC0CD5F;  
authenticate=true; authorize=true; name=Echo”
```

Informácie o technológii Bluetooth sú zo zdrojov [6], [7], [8], [9] a [10].

3. Návrh

Keďže táto práca má za úlohu preskúmať problematiku programovania mobilných telefónov v jazyku Java a najmä prepojiť mobilný telefón s programom Imagine Logo, táto kapitola sa bude zaoberať najmä touto oblasťou. Ďalej budú opísané komponenty, resp. príklady použitia systému.

3.1. Scenáre použitia

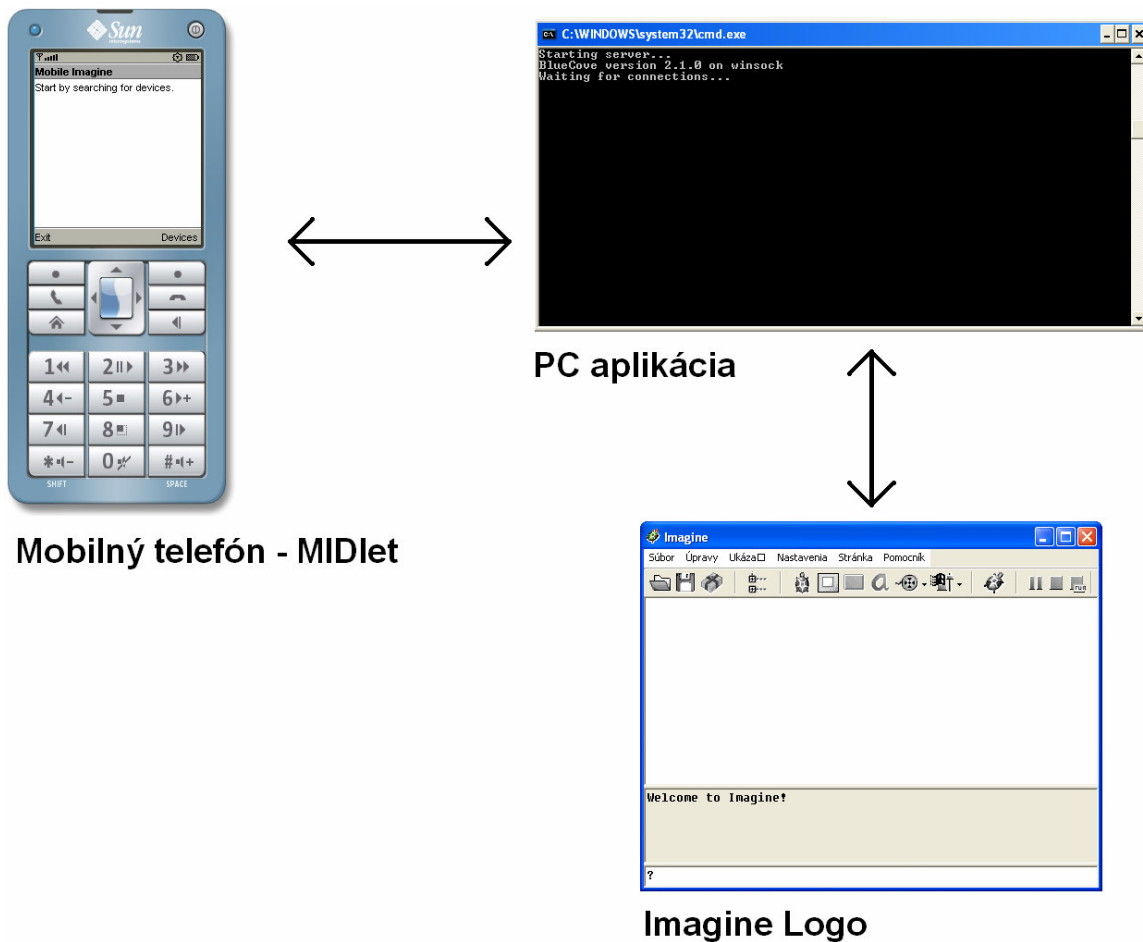
Táto práca vyžaduje vývoj dvoch samostatných aplikácií. Jedna určená pre mobilné telefóny, tiež nazývaná MIDlet, má názov **Mobile Imagine Connector Client (MICC)**. Druhá určená pre PC sa nazýva **Mobile Imagine Connector Server (MICS)**. Zámerom je, aby tieto dve aplikácie komunikovali cez Bluetooth. Ako názvy prezrádzajú, MICS bude fungovať ako server, na ktorý sa pripojí MICC.

Pôvodný zámer práce bol, aby aplikácia v mobilnom telefóne komunikovala priamo s Imagine Logo. To znamená bez nutnosti vytvorenia aplikácie pre PC, ktorá sprostredkúva spojenie medzi mobilným telefónom a Imagine. Na to, aby takáto schéma bola možná, je potrebné, aby mobilný telefón podporoval Serial Port Profile (SPP). Z komunikácie so školiteľom počas tvorby práce vyplynulo, že je veľa telefónov, ktoré SPP nepodporujú. Boli dve možnosti, a to buď pokračovať v tvorbe práce s použitím SPP alebo sa preorientovať na technológiu, ktorá je viac podporovaná, a tak výsledná aplikácia bude fungovať na viacerých mobilných telefónoch. Prednosť dostala druhá možnosť a bol vybraný protokol OBEX, ktorý je viac podporovaný, hlavne v nových operačných systémoch mobilov.

Mobile Imagine Connector Client má za úlohu po spustení umožniť vyhľadanie Bluetooth zariadení v okolí. Používateľ si následne vyberie správne zariadenie, teda PC, na ktorom je spustený Mobile Imagine Connector Server. V tejto fáze musí byť MICS spustené. MICC po vybratí zariadenia skontroluje, či na ňom zapnutá aplikácia MICS. Po úspešnom spojení sa o tom v oboch aplikáciách vypíše správa. Už v tejto fáze môžu

aplikácie komunikovať a využívať komponenty systému opísane v kapitole 3.2. Komponenty systému.

Aplikácia MICS po pripojení s MICC čaká na požiadavku na pripojenie od programu Imagine Logo. S Imagine Logo komunikuje cez sériový port.



Obr. 6 Ilustrácia celkovej komunikácie

3.2. Komponenty systému

V tejto kapitole sú opísané všetky funkcie systému.

Aplikácia v mobilnom telefóne má možnosť zobrazenia piatich rôznych obrazoviek, resp. piatich rôznych módov:

- hlavná obrazovka: vo fáze vyhľadávania zariadení a služieb sa tu vypisuje informácia o začatí vyhľadávania, priebehu a úspešnom, prípadne neúspešnom skončení. V prípade, že MICC stratí spojenie s MICS alebo nastane iná chyba, informácia sa vypíše na tejto obrazovke. Umožňuje prepnúť sa do obrazovky s chatom, fotoaparátom alebo joystickom alebo ukončiť aplikáciu.
- obrazovka s chatom: vypisujú sa tu všetky prijaté a poslané textové správy. Umožňuje sa prepnúť na obrazovku pre napísanie novej správy alebo ísť späť na hlavnú obrazovku.
- obrazovka pre novú správu: používateľ sem napíše textovú správu, ktorú si želá poslať do Imagine. Umožňuje poslať napísanú správu alebo ísť späť na obrazovku s chatom.
- obrazovka s fotoaparátom: zobrazuje aktuálne zachytávaný obraz fotoaparátom. Umožňuje zachytiť obraz alebo sa prepnúť na obrazovku s odfoteným obrázkom.
- obrazovka s odfoteným obrázkom: zobrazuje posledný zachytený obrázok. Pokiaľ ešte nebol vytvorený snímok, je o tom v tejto obrazovke podaná správa. Umožňuje tento obrázok uložiť do mobilného telefónu, poslať ho do PC, prepnúť sa späť do obrazovky s fotoaparátom alebo ísť na hlavnú obrazovku.
- obrazovka s joystickom: slúži na ovládanie NXT robota. Zachytávajú sa stlačenia štyroch klávesov, dopredu (kláves 2), dozadu (kláves 8), doľava (kláves 4) a doprava (kláves 6).

3.2.1. Chat

Chat umožňuje posielanie textových správ z MICC do Imagine a naopak. Keďže všetka komunikácia prechádza cez MICS, aj tu sa vypisuje, čo ktorá strana poslala. Taktiež hneď po úspešnom spojení MICC a MICS sa vyšle uvítacia správa z PC a zobrazí sa v chatovacom okne mobilu.

3.2.2. Fotoaparát

Aplikácia MICC využíva fotoaparát mobilného telefónu. Pokiaľ mobilný telefón fotoaparát neobsahuje, táto funkcionálna nebude podporovaná. Po zvolení módu fotoaparát, sa zmení obrazovka mobilného telefónu a ukáže sa na nej obraz zachytávaný fotoaparátom. Keď používateľ uzná za vhodné, stlačí tlačítko a záber sa odfotí. Po odfotení sa ukáže záber prispôsobený veľkosti obrazovky. V tomto bode má používateľ možnosť obrázok uložiť do pamäte mobilného telefónu, poslať ho aplikácii MICS alebo znova prepnúť do módu fotenia. V prípade poslania obrázku, sa o začatí a úspešnom dokončení prijímania vypíše oznam v MICS.

3.2.3. Joystick

Keď je aplikácia v tomto móde, a s Imagine je spojený NXT robot, je možné ovládať robota z MICC cez joystick. V prípade telefónov s klávesnicou, sú na to určené štyri klávesy: dopredu (kláves 2), dozadu (kláves 8), doľava (kláves 4) a doprava (kláves 6). V prípade dotykového mobilného telefónu bez klávesnice sa zobrazí ovládací panel s rovnakou funkcionálnosťou ako keby tam bola klávesnica. Na tejto obrazovke sa vypisuje, ktorý kláves bol stlačený.

3.2.4. Ovládanie mobilného telefónu z Imagine

Poslaním správneho príkazu z Imagine, je možné v aplikácii MICC zobrazit' hlavnú obrazovku, obrazovku s chatom, fotoaparátom, odfoteným obrázkom alebo

joystickom. Ďalej je možné jedným príkazom zobrazit' obrazovku s fotoaparátom a odfoťiť záber. Iným príkazom je možné poslať späť do PC posledný odfoťený obrázok, ak už nejaký odfoťený bol. Posledný príkaz je na prepnutie na obrazovku s fotoaparátom, odfoťenie záberu a poslanie do PC.

4. Implementácia

V tejto kapitole je opísaný postup implementácie spolu s dôležitými prvkami systému.

4.1. Spojenie MICC a MICS

Ako je napísané v kapitole 3.1. Scenáre použitia, táto práca sa skladá z dvoch aplikácií, a to Mobile Imagine Connector Client (MICC) a Mobile Imagine Connector Server (MICS). Dôležitým prvkom pri spojení dvoch aplikácií cez Bluetooth je spojovací string, opísaný v kapitole 2.2.1.3. Bluetooth master zariadenie. Len na pripomenutie, má formát {schéma}:{cieľ}{parametre}

Keďže je v aplikáciách používaný protokol OBEX, {schéma} je btgoep, čo vyjadruje, že ide o Bluetooth a profil Generic Object Exchange Profile (GOEP).

Časť {cieľ} sa skladá zo sieťovej adresy a UUID. Sieťová adresa je v tomto prípade “//localhost:”. Na získanie unikátneho UUID existuje v Linuxe príkaz *uuid -v1*. Takisto existuje množstvo generátorov UUID online. Pre túto prácu je UUID získané z online generátora UUID [11] a má hodnotu “49C41BB556414702B914EFEE734AF38B”. Čiže {cieľ} vyzerá nasledovne “//localhost:49C41BB556414702B914EFEE734AF38B”.

V poslednej časti {parametre} je pridané len meno “Mobile Imagine Connector” a určenie, že ide o server. Celá časť {parametre} vyzerá nasledovne: “name=Mobile Imagine Connector;master=true”.

Celý spojovací string vyzerá nasledovne:
"btgoep://localhost:49C41BB556414702B914EFEE734AF38B;name=Mobile Imagine Connector;master=true";

4.2. Komunikácia medzi MICC a MICS

Vzhľadom na to, že sa správy rôznych typov posielajú oboma smermi, z mobilného telefónu do PC, a naopak, bolo potrebné vytvoriť protokoly, na základe ktorého sa tieto typy správ odlišia.

4.2.1. Protokol komunikácie z MICC do MICS

Z mobilného telefónu sú do PC posielané tri rôzne typy správ. Pre každý typ je vytvorená konštanta s číselnou hodnotou. Tieto typy správ sú:

- MESSAGE_TYPE_TEXT: má hodnotu 1. Posielanie správy textu v chate.
- MESSAGE_TYPE_BUTTON: má hodnotu 2. Posielanie stlačeného tlačítka pri ovládaní NXT robota.
- MESSAGE_TYPE_IMAGE: má hodnotu 3. Posielanie obrázku.

Počas posielania správ z mobilu do PC sa objavila chyba v implementácii protokolu OBEX. Táto chyba je opísaná aj v online diskusii [12]. Pokiaľ sa na server do PC posielala správa menšia ako 1 kB, čiže 1024 bytov, tak sa na serveri “nezavolá“ metóda na prijatie správy onPut(), ale metóda onDelete(). Riešenie pre tento problém je posielanie vždy aspoň 1 kB. Posielania obrázku sa to netýka, keďže obrázok má vždy veľkosť väčšiu. Problém nastáva pri posielaní textu do chatu alebo pri ovládaní robota, keď sa posielala informácia, ktoré tlačítka bolo stlačené. Pri týchto dvoch správach je formát posielanej správy nasledovný:

```
{typ správy}{dĺžka správy}*{správa}{výplň}
```

pričom jednotlivé prvky tohto stringu znamenajú:

- typ správy: hodnota 2 alebo 3, podľa toho, či sa posielala text do chatu alebo stlačenie tlačítka.
- dĺžka správy: počet znakov posielanej správy ukončená hviezdíčkou.
- správa: samotná posielaná správa.

- výplň: ak dĺžka predošlých častí je menej ako 1024, tak sa pridá 1024 - {dĺžka predošlých častí} núl (znakov '0').

Posielanie obrázku je v nasledovnom formáte:

{typ správy}{obrázok}

pričom jednotlivé prvky tohto stringu znamenajú:

- typ správy: hodnota 1.
- obrázok: obrázok vo formáte JPEG prevedený do poľa bytov. Obrázok sa posiela vo formáte JPEG, pretože Imagine je schopný s týmto formátom pracovať.

4.2.2. Protokol komunikácie z MICS do MICC

Aplikácii v mobilnom telefóne sa dajú z Imagine posielat' príkazy. Tieto príkazy sú jednoznakové, pričom v jednom prípade sú dôležité aj zvyšné znaky v správe. Prehľad príkazov:

Príkaz	Popis
a	Prijať správu do chatu. Samotná správa nasleduje po príkaze.
b	Zmeň obrazovku na hlavnú obrazovku.
c	Zmeň obrazovku na chat.
d	Zmeň obrazovku na fotoaparát.
e	Zmeň obrazovku na obrázky.
f	Zmeň obrazovku na joystick.
g	Zmeň obrazovku na fotoaparát a odfoť záber.
h	Pošli späť do PC obrázok (ak je nejaký odfotený).
j	Zmeň obrazovku na fotoaparát, odfoť záber a pošli ho späť do PC.

4.3. Komunikácia medzi MICS a Imagine Logo

Imagine Logo má rozširujúcu knižnicu pre komunikáciu s NXT kockou, ktoré umožňuje komunikáciu cez sériový port. V Imagine je prednastavený port 51. Aby tento port ostal dostupný na komunikáciu s NXT kockou, na spojenie s aplikáciou MICS je používaný port 5151.

Imagine Logo má na komunikáciu cez sériový port vytvorený vlastný protokol. Tento protokol je opísaný v textovom súbore v priečinku Imagine [13]. Dôležité časti tohto protokolu pre túto prácu:

1. Každý paket začína bytom 07, zvyšok je string vo formáte:

<LEN>!<MESSAGE>

Kde <LEN> je počet znakov v správe <MESSAGE>.

2. Správa <MESSAGE> je v nasledovnom formáte:

<TYPE><ARGC>!<ARG0>!<ARG1>!...!<ARGN>

Kde:

- a) <TYPE>: je jeden znak, ktorý špecifikuje typ správy.
- b) <ARGC>: je počet argumentov, ktoré nasledujú počínajúc 0. Teda ak je jeden argument, tak je toto číslo 0.
- c) <ARGi>: sú podstringy, oddelené znakom '!'.

3. Typy správ a ich argumenty:

Typ správy s argumentami	Popis	príklad
a nickname	Požiadavka na prihlásenie s prezývkou	a0!client
b nickname	Potvrdenie požiadavky na prihlásenie s vrátenou prezývkou	b0!client
w nicklen nick_and_message	Poslanie správy klientovi. nicklen je dĺžka prezývky v bytoch. Nasleduje prezývka spolu so správou.	w1!6!server123

W nick1len nick1_and_nick2len nick2_and_message	Poslanie správy serveru. Argumenty sú na podobnom princípe ako pri posielaní správy klientovi.	W2!6!client6!server 1234
--	--	-----------------------------

4.4. Podpisovanie MIDlet-u

MIDlet je možné podpísať certifikátom. To umožní používateľovi, ktorý si ho inštaluje, vidieť, že daný MIDlet pochádza z dôveryhodného zdroja. Certifikáty vydávajú certifikačné authority. Jediný spôsob pre programátora, ktorý chce získať certifikát, je si ho zakúpiť. Certifikačné authority sú napríklad Thawte, VeriSign alebo Java Verified.

Pokiaľ MIDlet, ktorý sa inštaluje do mobilného telefónu nie je podpísaný certifikátom od certifikačnej authority, zobrazí sa bezpečnostné varovanie (security warning), že aplikácia je nedôveryhodná a môže poškodiť telefón. Toto môže odradiť používateľa od nainštalovania aplikácie, keďže jej nemôže veriť. Počas používania aplikácie sa pri niektorých úkonoch taktiež zobrazí bezpečnostné varovanie a operačný systém mobilného telefónu sa používateľa spýta, či chce pokračovať v danom úkone. Niekedy je možné zadať, aby sa systém pýtal na povolenie vždy pri danom úkone, iba prvýkrát, alebo daný úkon úplne zakázať. Pri zakúpení certifikátu a podpísaní MIDletu sa pri niektoré úkony pridá možnosť povoliť vždy.

Pretože certifikát treba zakúpiť, MIDlet v tejto práci nie je podpísaný.

5. Experimentálne overenie, príklady použitia

V tejto kapitole sú popísané príklady použitia spolu so screen shotmi. Screen shoty sú z aplikácie spustenej na emulátore Wireless Toolkit od spoločnosti Sun Microsystems.

5.1. Inštalácia aplikácie MICC



Obr. 7, 8 Inštalácia aplikácie MICC, krok 1, 2

Ako možno vidieť na obr. 7, pri inštalácii MIDlet-u v kroku 1 sa na mobilnom telefóne zobrazia detaily aplikácie, ktorá sa ide inštalovať. Operačný systém sa pýta

používateľa, či chce aplikáciu nainštalovať. V detaili “Pokročilé” (“Advanced“) je uvedené “neidentifikovaná tretia strana” (“unidentified third party“), pretože MIDlet nie je podpísaný certifikátom. Preto sa aj na ďalšej samostatnej obrazovke uisťuje, či chce používateľ nainštalovať MIDlet, ktorý nie je podpísaný (obr. 8).

5.2. Spustenie aplikácie MICC a spojenie so serverom



Obr. 9 Aplikácia MICC po spojení so serverom

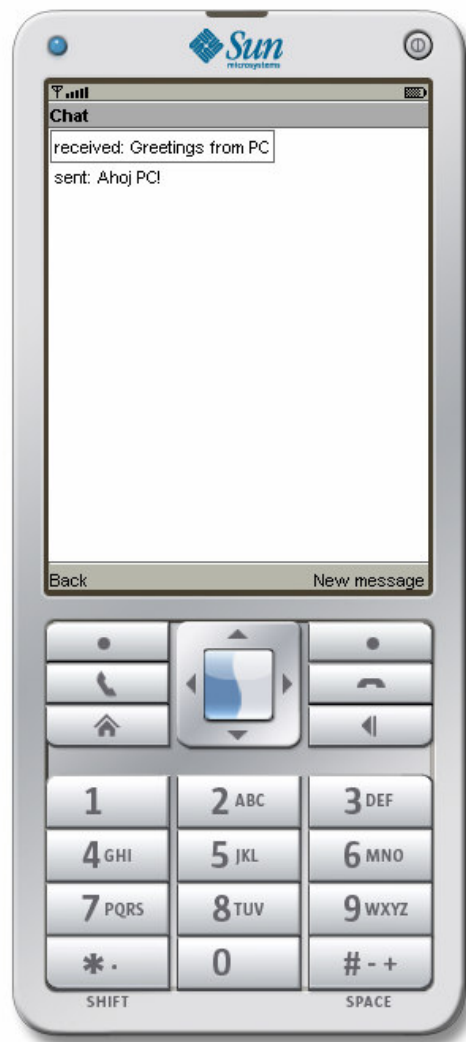
Wireless Toolkit emulátor je schopný emulovať Bluetooth spojenie, ale nevie sa spojiť so samotným Bluetooth zariadením počítača, na ktorom je spustený. Preto

nasledujúce screen shoty sú iba ukážkové a nedajú sa v skutočnosti simulovať na emulátore.

Po spustení aplikácie možno vidieť iba prvý riadok “Začnite vyhľadávaním zariadení.” (“Start by searching for devices.”) a namiesto “Menu” je iba “Zariadenia” (“Devices”). Na obr. 9 je aplikácia už spojená so serverom, používateľ stlačil “Zariadenia” a po vyhľadaní zariadení v okolí si vybral to správne. Možno tu vidieť kontrolné výpisy, že sa začalo a skončilo s vyhľadávaním zariadení, po tom, ako si používateľ vybral zariadenie s Bluetooth menom “PC_1” sa začalo s vyhľadávaním služieb na tomto zariadení. Potrebná služba, čiže aplikácia MICS, sa našla a MICC sa na ňu pripojilo. V “Menu” možno vidieť štyri možnosti, a to:

- “Zariadenia” (“Devices”): znovu vyhľadanie zariadení.
- “Chat”: prepnutie na obrazovku s chatom.
- “Fotoaparát” (“Camera”): prepnutie na obrazovku s fotoaparátom.
- “Joystick”: prenutie na obrazovku s ovládaním NXT robota.

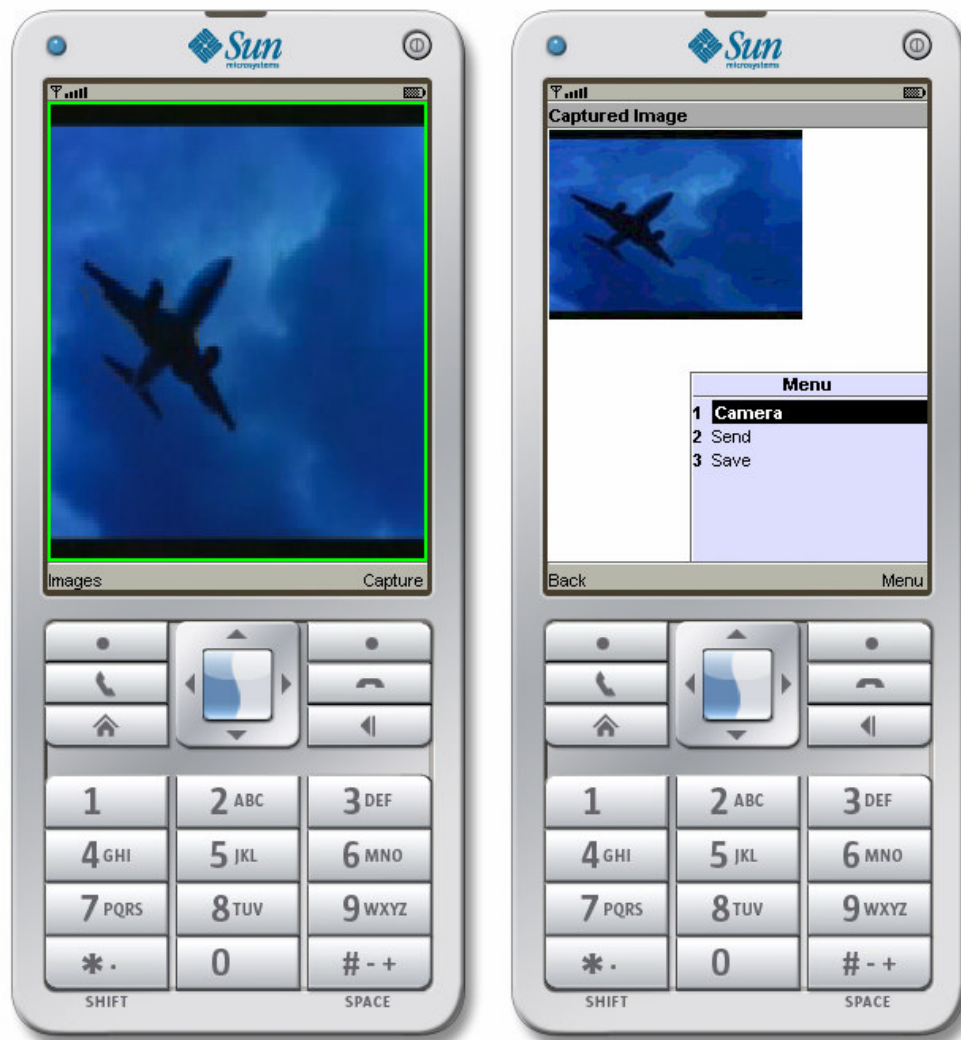
5.2.1. Aplikácia MICC – Chat



Obr. 10 Aplikácia MICC – Chat

Na obr. 10 možno vidieť obrazovku s chatom. Po pripojení k serveru, server pošle uvítaciu správu “Pozdravy z PC” (“Greetings from PC”). Po stlačení “Nová správa” (“New message”), sa zobrazí okno, kde používateľ môže napísať text, ktorý chce poslať na server. Na screen shote možno vidieť, že používateľ odpovedal správou “Ahoj PC!”. Každá správa poslaná zo servera sa vypíše vo formáte “prijaté: “ (“received: ”) a text správy. Každá správa poslaná na server sa vypíše v tvare “poslané: “ (“sent: “).

5.2.2. Aplikácia MICC – Fotoaparát



Obr. 11, 12 Aplikácia MICC – Fotoaparát a odfotený obrázok

Na obr. 11 je zobrazená obrazovka s fotoaparátom. V emulátore sa zobrazuje video, ktorým simuluje obraz vo fotoaparáte. Keď chce používateľ zachytiť obraz, stlačí tlačítko “Odfot” (“Capture”). K poslednému odfotenému obrázku sa môže vrátiť stlačením “Obrázky” (“Images”). Na obr. 12 je zobrazený odfotený obrázok. Používateľ má možnosť znova sa prepnúť na obrazovku s fotoaparátom a odfoťiť nový obrázok. Odfotený obrázok môže poslať aplikácii MICS stlačením “Poslať” (“Send”) alebo ho môže uložiť do pamäte mobilného telefónu stlačením “Uložiť” (“Save”).

5.2.3. Aplikácia MICC – Joystick



Obr. 13 Aplikácia MICC – Joystick

Na obr. 13 je zobrazená obrazovka Joystick. Stláčaním tlačidiel hore, dole, doľava, doprava alebo 2, 8, 4, 6 sa posielajú na server informácie o stlačení týchto tlačidiel a tým sa aj pohýna NXT robot. Na obrazovku mobilného telefónu sa vypíše, ktoré tlačítko bolo stlačené.

6. Záver

Bakalárska práca sa venuje použitiu programovacieho jazyka J2ME, ako aj návrhu a implementácii dvoch aplikácií. Prvá je naprogramovaná v J2ME určená pre mobilné telefóny, čiže MIDlet a nazýva sa Mobile Imagine Connector Client (MICC). Druhá aplikácia je naprogramovaná v J2SE a nazýva sa Mobile Imagine Connector Server (MICS). Úspešne sa podarilo vytvoriť spojenie medzi týmito dvoma aplikáciami cez Bluetooth. MICS sa taktiež vie pripojiť na program Imagine Logo, pričom spolu komunikujú cez sériový port.

Ciele bakalárskej práce boli splnené. Spojenie medzi mobilným telefónom a programom Imagine Logo funguje. Z Imagine sa dajú posielat' príkazy do MICC a taktiež sa dá MICC ovládať z Imagine. Z MICC sa dajú posielat' textové dáta alebo obrázky. Aplikácia bola úspešne testovaná na mobilnom telefóne Nokia 5530 XpressMusic.

Existujú ďalšie možnosti rozširovania využitím možností Imagine Logo alebo mobilného telefónu. Napr. by sa na mobilnom telefóne dali prehrávať zvuky, prípadne nahrávať a posielat' ich späť do PC. Na ovládanie NXT robota by sa dal využiť akcelerometer novších mobilných telefónov. Taktiež by bolo možné vytvoriť skicár, spoločný pre mobilný telefón a Imagine, do ktorého by mohli obe aplikácie naraz kresliť a rovnaký obraz by bol na oboch zariadeniach.

7. Literatúra

1. Kim Topley. *J2ME in a Nutshell*. 2002
2. Nokia Forum. *Mobile Java – Introduction*. 2010.
http://www.forum.nokia.com/document/Mobile_Hands-on_Labs/Module6/Introduction.html
3. Oracle Corporation. *Sun Java Wireless Toolkit for CLDC*. 2010.
<http://java.sun.com/products/sjwtoolkit/download.html>
4. Nokia Forum. *Symbian, S60 platform, and S60 device SDKs*. 2010.
http://www.forum.nokia.com/info/sw.nokia.com/id/ec866fab-4b76-49f6-b5a5-af0631419e9c/S60_All_in_One_SDKs.html
5. BlueCove Team. 2008.
<http://www.bluecove.org/>
6. Bluetooth SIG. 2010.
<http://www.bluetooth.com/English/Pages/default.aspx>
7. Bruce Hopkins, Ranjith Antony. *Bluetooth for Java*. 2003
8. The Forum Nokia Developer Community. 2010.
http://wiki.forum.nokia.com/index.php/Wiki_Home
9. Palo Pacific Technology Pty Ltd. 2010.
<http://www.palowireless.com>
10. Kjell Jørgen Hole. 2009. Bluetooth: Part 7: JSR-82 API for RFCOMM.
<http://www.kjhole.com/index.html>
11. GUID Generator. 2010.
<http://www.guidgenerator.com/>
12. The Forum Nokia Developer Community. 2009. Developer Discussion Board
<http://discussion.forum.nokia.com/forum/showthread.php?t=160700>
13. Imagine Logo Serial Port Protokol.
Textový súbor *net_protocol.txt* obsiahnutý v inštalácii Imagine Logo.