

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA ELEKTROTECHNICKÁ
KATEDRA ŘÍDICÍ TECHNIKY



DIPLOMOVÁ PRÁCE

Dálkové řízení modelu



Praha, 2005

Petr Mužíček

Zadání diplomové práce

Student: Petr Mužíček

Obor: Technická kybernetika

Název tématu: Dálkové řízení modelu

Zásady pro vypracování:

1. Navrhněte uspořádání čidel modelu vozidla pro orientaci v rovině.
2. Upravte model podle Vašeho návrhu. Pro přenos řídicích signálů využijte bezdrátové spojení.
3. Porovnejte možnosti řídicích systému pro řízení tohoto modelu.
4. Vybraným řídicím systémem demonstrujte navržený algoritmus řízení pohybu modelu po zadané dráze.

Seznam odborné literatury: Dodá vedoucí práce

Vedoucí diplomové práce: doc.Ing.Jan Bílek, CSc.

Datum zadání diplomové práce: prosinec 2003

Datum odevzdání diplomové práce: květen 2003

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. , o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

Petr Mužíček

Poděkování

Děkuji především vedoucímu diplomové práce Doc.Ing. Janu Bílkovi CSc. za cenné rady, podněty a připomínky. Dále bych také rád poděkoval za cenné rady Ing. Pavlu Růžičkovi a Ing. Petru Jurčíkovi.

Dále můj vděk patří všem, kteří mě při práci podporovali i jinak než odbornými radami.

Abstrakt

V této diplomové práci je popsán způsob řízení modelu auta počítačem a mobilním telefonem prostřednictvím bezdrátové technologie Bluetooth a řízení pomocí optických čidel. V prvém případě je model auta řízen pokyny zaslanými z počítače respektive z mobilního telefonu přes moduly Bluetooth, které jsou k počítači a k elektronice v autě připojeny. Komunikace s těmito moduly je uskutečněna přes sériové rozhraní. V druhém případě je model řízen mikroprocesorem na základě signálů přijatých z optických čidel na podvozku auta. Cílem práce je ověřit využití bezdrátového propojení pro řízení objektu v reálném čase.

Abstract

In this diploma thesis is being described the way of remote control of a car model by personal computer and by a mobile phone via wireless technology Bluetooth and optical sensors control. In the first case is the car model controlled by the instructions, which are sent from computer to microprocessor situated in the car model. Communication between computer and microprocessor is realised by two Bluetooth modules, which are connected to computer and microprocessor via serial interface. In the second case is the car model controlled by microprocessor based on the signals received from the optical sensors that are on car chassis. The goal of this diploma thesis is to verify wireless connection for the real-time control.

Obsah

1	Úvod	1
2	Dálkové řízení	2
2.1	Možnosti bezdrátové komunikace	2
2.1.1	IEEE 802.11	3
2.1.2	IEEE 802.15	5
2.1.3	Srovnání	7
2.1.4	Shrnutí	10
2.2	Bluetooth	11
2.2.1	Rádio specifikace	11
2.2.2	Baseband specifikace	12
2.2.2.1	Komunikační jednotka Bluetooth	13
2.2.2.2	Přenosový kanál	14
2.2.2.3	Fyzické linky	14
2.2.2.4	Pakety	15
2.2.2.5	Logické kanály	17
2.2.2.6	Adresování zařízení	17
2.2.2.7	Vysílací a přijímací proces	17
2.2.2.8	Časování vysílání a příjmu	18
2.2.2.9	Stavy zařízení a řízení kanálu	20
2.2.2.10	Vytvoření spojení	20
2.2.2.11	Režimy zařízení ve stavu <i>Connection</i>	20
2.2.3	Link Manager Protocol	21
2.2.4	HCI specifikace	21
2.2.4.1	HCI příkazy	23
2.2.4.2	HCI události	27
2.2.4.3	HCI datové pakety	29
2.2.5	HCI RS232 transportní vrstva	30
2.2.6	L2CAP	31
3	Realizace vzdáleného řízení	32
3.1	Modely aut	33
3.1.1	Model auta pro DP	33
3.1.1.1	Potřebné úpravy	33
3.1.1.2	Mikroprocesor	34

3.1.1.3	Bluetooth modul	35
3.1.1.4	Akční člen - servomotor	37
3.1.1.5	Akční člen - ss motor	39
3.1.1.6	A/D převodník	40
3.1.1.7	DC-DC měnič	41
3.1.1.8	Optická čidla	42
3.1.1.9	Výsledná deska plošných spojů	44
3.1.2	Model auta pro BTRC	45
3.2	Řídicí systém	47
3.2.1	PC s Bluetooth modulem	47
3.2.2	Mobilní telefon	47
4	Softwarové řešení vzdáleného řízení	49
4.1	Komunikační protokol	49
4.2	Mikroprocesor PIC	51
4.2.1	Paměťová organizace procesoru	51
4.2.2	Komunikace přes RS232	52
4.2.3	Přerušení	54
4.2.4	A/D převodník	54
4.2.5	Časovač a PWM	56
4.2.6	Struktura programu pro DP	57
4.2.7	Struktura programu pro BTRC	60
4.2.8	Ošetření chybových stavů	61
4.3	Počítač	62
4.3.1	Práce s programem	63
4.3.2	Komunikace s modelem auta	64
4.3.3	Ošetření chybových stavů	66
5	Závěr	67
Literatura		67
A Struktura přiloženého CD ROM		69
B Realizace výsledné desky		70
C Komunikace RS232		72

Seznam obrázků

2.1	Funkční bloky v systému Bluetooth	13
2.2	Point-to-point (a), Piconet (b), Scatternet (c)	13
2.3	TDD schéma a časování	14
2.4	Multi-slot pakety	15
2.5	Formát standardního paketu	16
2.6	Fuknční diagram vysílání	18
2.7	Cyklus TX a RX z pohledu Mastera	19
2.8	Komunikace Mastera s více Slave	19
2.9	Přehled nižších softwarových vrstev	22
2.10	Přehled nižších softwarových vrstev vzhledem k přenosu informace	22
2.11	Stavba paketu HCI příkazu	23
2.12	Stavba paketu HCI události	28
2.13	HCI ACL datový paket	30
2.14	HCI RS232 transportní vrstva	30
2.15	L2CAP v architektuře Bluetooth protokolu	31
3.1	Soustava bezdrátového řízení	32
3.2	Řízení pomocí čidel	32
3.3	Model auta pro DP	33
3.4	Schéma elektroniky	34
3.5	Mikroprocesor PIC 16F874	34
3.6	Bluetooth modul v modelu auta	35
3.7	Schéma 74LVX245	37
3.8	Servomotor	38
3.9	Časové průběhy PWM pro servomotor	38
3.10	Způsob připojení serva k přední nápravě	39
3.11	Schéma zapojení H-můstku pro řízení ss motoru	40
3.12	Diferenční zesilovač a), napěťový sledovač b)	41
3.13	Integrovaný obvod NE5532	41
3.14	Schéma zapojení obvodu dc-dc	42
3.15	Optické čidlo CNY70	42
3.16	Funkce čidla CNY70	43
3.17	Možnosti orientace v rovině	43
3.18	Schéma zapojení optických čidel	44
3.19	Výsledná deska	45

3.20	Model auta pro BTRC	46
3.21	Blokové schéma přídavné elektroniky	46
3.22	Schéma řídicího systému	47
3.23	Mobilní telefon Nokia 6600	48
4.1	Zjednodušené schéma komunikace mezi PC a modelem	50
4.2	Schéma 1 komunikačního cyklu	50
4.3	Zjednodušené schéma komunikace mezi telefonem a modelem	51
4.4	Programátor PIC	51
4.5	Blokové schéma A/D převodníku	55
4.6	Blokové schéma programu pro PIC	57
4.7	Schéma komunikace mezi procesorem a Bluetooth modulem	59
4.8	Struktura programu pro BTRC	60
4.9	Schéma komunikace mezi procesorem a Bluetooth modulem	61
4.10	Hlavní okno aplikace	63
4.11	Informace o připojeném (a) a vzdáleném (b) Bluetooth modulu	64
4.12	Informace o nalezených Slave zařízeních	64
4.13	Okno ovládání modelu auta	65
4.14	Informace o napětí akumulátorů, stavu optických čidel a kvality spojení	65
4.15	Informace o odeslaných a přijatých paketech	66
A.1	Schéma zapojení elektroniky pro DP	69
B.1	Schéma zapojení elektroniky pro DP	70
B.2	Rozmístění součástek	71
B.3	Schéma zapojení přídavné elektroniky pro BTRC	71

Seznam tabulek

2.1	Porovnání rychlostí sítí WPAN a WLAN	3
2.2	Porovnání přenosových rychlostí ZigBee a Bluetooth	7
2.3	Obecné porovnání Wi-Fi, ZigBee a Bluetooth	8
2.4	Porovnání ZigBee a Bluetooth	8
2.5	Výkonové třídy Bluetooth	12
2.6	Identifikátory HCI paketů při komunikaci přes RS232	30
3.1	Popis použitých pinů mikroprocesoru PIC	36
3.2	Popis pinů použitého modulu Bluetooth	36
3.3	Popis pinů převodníku 74LVX245	37
4.1	Výběr banky v registru Status	52
4.2	Nastavení předděličky časovače procesoru	56

Kapitola 1

Úvod

Bezdrátová komunikace je v dnešní době velmi používaný způsob pro ovládání a komunikaci různých zařízení i technologických procesů. V této diplomové práci jsou popsány dva z možných způsobů realizace bezdrátové komunikace.

Prvním způsobem je bezdrátová komunikace osobního počítače s mikroprocesorem. Byl pořízen RC model automobilu, který zvládal jízdu maximální rychlostí vpřed, vzad a zatáčet jednopolohově vlevo a vpravo. Model byl upraven tak, aby zvládal na základě pokynů proporcionální řízení. Přidán byl modul Bluetooth firmy Siemens (SiMo), mikroprocesor PIC firmy Microchip a další elektronika potřebná ke komunikaci modulu s mikroprocesorem a k samotnému řízení modelu. Jako řídicí systém slouží SW v počítači, který je přes rozhraní RS232 připojen k druhému Bluetooth modulu. Takto bylo zrealizováno bezdrátové řízení modelu.

Kromě řízení z počítače je model ještě možné řídit nezávisle na řídicím systému. K tomuto je vybaven optickými čidly připevněnými k přední části podvozku. Model auta se pak pohybuje na základě signálu z těchto čidel. Uživatelský SW v PC umožňuje přepínání mezi těmito způsoby řízení (počítačem nebo čidly). Existence spojení mezi modelem a počítačem je ověřována pravidelným zasíláním paketu obsahujícího informaci o napětí nabíjecích akumulátorů v modelu auta a stavu optických čidel.

Druhým způsobem je komunikace mobilního telefonu s mikroprocesorem. Tento způsob řízení byl realizován v rámci projektu "Bluetooth Remote Control" pro Výzkumné a vývojové centrum v Praze (www.rdc.cz). Pro tento projekt byl pořízen jiný model auta. Důležitá je především SW část, jejíž popis následuje vždy po popisu samotné diplomové práce.

Text je rozdělen do tří kapitol, které pojednávají o možných způsobech bezdrátové komunikace, realizaci zvolené technologie na modelu auta, návrhu a realizaci řídicího systému a poslední kapitola o softwarovém řešení obou projektů.

Kapitola 2

Dálkové řízení

2.1 Možnosti bezdrátové komunikace

Dnešní počítačový svět je doslova přehlcen různými druhy sítí. Klasické drátové jsou z hlediska volnosti problematické, protože je potřeba přístupový konektor. Bezdrátové jsou oproti tomu vítanou změnou. Rádiové či infračervené vlny jsou volně okolo nás.

Bezdrátové sítě nabízejí v principu podobné služby a flexibilitu jako sítě drátové. Z hlediska funkčnosti a výsledku jsou kromě dosahovaných přenosových rychlostí ekvivalentní k sítím drátovým (např. ethernetu).

Rádiové vysílání je náchylné na rušení, a to všemi zařízeními, které mohou na příslušných kmitočtech pracovat. Týká se to zejména bezlicenčního pásma 2.4GHz. Optické bezdrátové sítě či sítě založené na infračerveném záření zase nemohou mít překážky mezi vysílačem a přijímačem. Dosah související s kvalitou přenosu pak omezuje jejich velikost i počet systémů, které se v rámci daného prostoru mohou nacházet, aby nedocházelo ke vzájemnému rušení. Zajištění bezpečnosti bezdrátové komunikace je při rádiovém vysílání jedním z nejobtížnějších problémů, podobně jako roaming a směrování mezi různými sítěmi.

Bezdrátové sítě podle IEEE

IEEE (Institute of Electrical and Electronics Engineers) se specifikací bezdrátových LAN zabývá od roku 1990. Jedná se o následující standardy:

- IEEE 802.11 - bezdrátové lokální sítě WLAN (Wireless Local Area Network)
- IEEE 802.15 - bezdrátové osobní sítě WPAN (Wireless Personal Area Network)
- IEEE 802.16 - širokopásmový bezdrátový přístup (metropolitní sítě)

Bezdrátové lokální sítě WLAN a osobní sítě WPAN se řadí k technologiím nenáročným na napájení, s malým dosahem a malou prostorovou kapacitou. WLAN byly vymýšleny spíše pro podnikové využití a osobní sítě naopak pro domácnosti. Ultra-širokopásmová technologie má střední dosah a vyznačuje se vysokou prostorovou kapacitou. Osobní sítě mají menší dosah i přenosovou kapacitu než lokální.

Topologie WLAN je realizovaná především s návazností na síťovou infrastrukturu, v případě WPAN se primárně jedná o náhodné seskupení komunikujících zařízení (zařízení se může snadno připojit nebo odpojit od sítě). Současně je však možné využít připojení k přístupovému bodu AP (Access Point) jako ve všech typech WLAN.

Frekvenční rozsahy

Bezdrátové sítě WLAN a WPAN pracují ve frekvenčním pásmu 2.4 - 2.4835GHz, tedy zjednodušeně řečeno v pásmu 2.4GHz. Toto pásmo se také často označuje jako ISM (Industrial, Scientific, Medical)¹. V tomto pásmu pracuje mnoho různých bezdrátových zařízení, např. bluetooth produkty, ale i mikrovlnné trouby a v zahraničí i bezdrátové telefony. Kromě tohoto pásmu se vyhrazuje ještě pásmo 5GHz. To používá zatím pouze technologie 802.11a.

Frekvenční rozsah se ovšem liší země od země - v některých státech není povolené plné frekvenční spektrum, protože jeho části jsou již využívány pro jiné účely. V ČR ČTÚ (Český telekomunikační úřad) povoluje plné frekvenční spektrum, jako je tomu v USA nebo ve většině státech Evropy.

Základní porovnání rychlostí obou typů sítí je v tabulce 2.1.

WPAN 2.4GHz		WLAN 2.4GHz	
ZigBee	Bluetooth	802.11b	802.11g
250 kbps	1 Mbps	11 Mbps	54 Mbps

Tabulka 2.1: Porovnání rychlostí sítí WPAN a WLAN

2.1.1 IEEE 802.11

Standard IEEE 802.11 definující bezdrátové LAN (WLAN) začal vznikat počátkem devadesátých let, dokončen byl v roce 1997. V roce 1999 bylo uvedeno rozšíření tohoto standardu v podobě nové varianty nesoucí název IEEE 802.11b, které přineslo některé podstatné změny, především výrazné zvýšení rychlosti z maximálně 2Mbps až na 11Mbps.

Standard od svého vzniku počítal se třemi základními druhy přenosu. Dva využívaly technologie rozprostřeného spektra (Spread Spectrum) na klasických rádiových vlnách, třetí typ byl definován pro infračervený přenos. Ujaly se pouze první dvě technologie - DSSS (Direct Sequence Spread Spectrum) a FHSS (Frequency Hopping Spread Spectrum).

DSSS (Wi-Fi, ZigBee) využívá k přenosu prostřednictvím širokého spektra frekvencí tzv. Barkerovo kódování, nejčastěji v jeho 11bitové podobě a jako modulační metodu diferenční QPSK (Quadrature Phase Shift Keyed). Původní technologie DSSS pracuje na rychlosti 1 nebo 2Mbps (tato rychlosť je společná pro všechna řešení v původním IEEE 802.11). V současné době však DSSS díky zlepšené

¹Bezlicenční frekvenční pásmo určené pro účely výzkumu a medicíny.

přenosové technologii nabízí přenosové rychlosti až 11Mbps. Pro tento přenos však DSSS vyžaduje pásmo o šířce plných 22MHz. Protože šířka celého bezlicenčního pásma ISM je 83.5MHz, mohou být současně používány pouze tři nezávislé DSSS systémy.

FHSS (Bluetooth) využívá modulační metodu GSFK (Gaussian Frequency Shift Keyed). V současných provedeních nabízí přenosovou rychlosť 1.6 nebo 3.2Mbps. Pro přenos používá 78 skokových frekvencí z nichž každá má 79 skoků. Teoreticky může být provozováno současně až 79 FHSS systémů, prakticky se tato hodnota pohybuje kolem 15. Požadovaná šířka pásma je totiž kolem 6MHz. Naprostě bezproblémového přenosu je dosaženo při počtu 12-ti současně provozovaných FHSS systémů.

Obě technologie byly původně navrženy pro bezdrátové LAN (WLAN), a to především v budovách a jiných uzavřených prostorách. Podle Generální licence GL 12/R/2000 nesmí výkon převyšovat 100mW.

Hlavní výhodou DSSS je především vyšší přenosová rychlosť, té je však dosahováno na úkor kvality přenosu. DSSS je navíc podstatně méně efektivní a náchylné k rušení a jiným negativním vlivům. FHSS sice nabízí nižší přenosovou rychlosť, přenosové pásmo je však efektivněji využito a technologie jako taková je podstatně robustnější a následně odolnější. Díky změnám frekvence se dokáže "vyhnout" nepříznivým vlivům, byť se sníží přenosová rychlosť.

Typy standardů 802.11:

- **802.11a** - WLAN v pásmu 5GHz a s rychlosťí až 54Mbps
- **802.11b** - WLAN v pásmu 2.4GHz a s rychlosťí až 11Mbps
- **802.11c** - definice procedur pro síťové mosty (bridge), standard pro přístupové body
- **802.11d** - mezinárodní harmonizace. Se vznikem standardu 802.11 se ukázalo, že je potřeba mezinárodní kooperace a harmonizace. Zejména pásmo 5GHz se používá v mnoha státech různě a bylo třeba tomu standardizaci přizpůsobit tak, aby nevycházela vstřík pouze potřebám USA a Japonska.
- **802.11e** - rozšíření MAC² pro QoS. Zkratka QoS označuje službu Quality of Service zajišťující vyrovnanou kvalitu služby důležitou například pro multimédia.
- **802.11f** - Inter Access Point Protocol (IAPP) - stávající specifikace 802.11 nezahrnují standardizaci komunikace mezi jednotlivými přístupovými body pro zajištění bezproblémového roamingu.
- **802.11g** - zvýšení rychlosťi v pásmu 2.4GHz na 20Mbps se zpětnou kompatibilitou s 802.11b.

²protokol přístupu k médiu (Media Access Protocol)

- **802.11h** - změny v řízení přístupu k spektru 5GHz, které by měly reflektovat připomínky regulátorů evropských zemí tak, aby bylo možno sítě v pásmu 5GHz využívat i mimo budovy.
- **802.11i** - zlepšení bezpečnosti v 802.11 bezdrátových sítích vylepšením autentifikačního a šifrovacího algoritmu.

Zkratka Wi-Fi (Wireless Fidelity) se často zaměňuje s výrazem IEEE 802.11a/b/g. Samotný název Wi-Fi vytvořila WECA (Wireless Ethernet Compatibility Alliance). Jde o označení a logo udělované výrobkům pracujícím podle standardu 802.11a/b/g, které jsou mezi sebou vzájemně propojitelné.

Většina sítí založených na Wi-Fi funguje na buňkovém principu, kdy centrální přístupový bod zprostředkovává připojení všem stanicím v dosahu - analogicky s GSM sítí. Z důvodu zabezpečení proti zneužití disponují současná zařízení podporou 128 bitového šifrování protokolem WEP (Wireless Encryption Protocol), ale obecné principy zabezpečení Wi-Fi nejsou dostatečné pro všechny aplikace. Nahradit WEP by měl WPA (Wi-Fi Protected Access). WPA bude obsahovat několik nových technologií, které se ve WEPu nenacházejí, například zlepšený management klíčů a TKIP (Temporal Key Integrity Protocol). Více v [14].

2.1.2 IEEE 802.15

Bezdrátové sítě s malým dosahem do několika metrů se nejčastěji realizují rádiovou bezdrátovou sítí WPAN (Wireless Personal Area Network), založenou na specifikaci Bluetooth. Vedle této specifikace existují i nové typy malých bezdrátových sítí a všechny vznikají pod hlavičkou IEEE 802.15:

- **802.15.1 WPAN/Bluetooth** (2002) - norma pro WPAN na základě specifikace Bluetooth 1.1, pracující v pásmu 2.4GHz rychlostí 1Mbps
- **802.15.2 Coexistence** - zabývá se koexistencí všech bezdrátových technologií vytvořených v rámci IEEE 802, nejen samotných WPAN, ale i WLAN a WMAN (Wireless Metropolitan Area Network)
- **802.15.3 WPAN High Rate** (2003) - norma pro rychlou bezdrátovou osobní síť (11-55Mbps), přezdívaná WiMedia, pracující v pásmu 2.4GHz, se zabudovanou podporou pro kvalitu služby (QoS) pro náročné domácí multimediální aplikace, cílem je ještě rychlejší verze sítě (802.15.3a) s kapacitou od 110Mbps, která bude založena na technologii **UWB** (UltraWideBand)
- **802.15.4 WPAN Low Rate** (2003) - norma pro pomalejší osobní rádiové sítě do 250kbps s minimálními energetickými potřebami; jako svůj základ používá 802.15.4 také průmyslová specifikace pod označením **ZigBee**

Mezi nejznámější zástupce WPAN patří Bluetooth (IEEE 802.15.1) a ZigBee (IEEE 802.15.4).

Bluetooth

První bezdrátovou osobní sítí byl rádiový systém Bluetooth. Mezi nejvýznamnější prosazovatele Bluetooth patří vedle zakladatelů SIG (Bluetooth Special Interested Group) společnosti 3Com, Agere, Microsoft a Motorola.

Specifikace Bluetooth (první verze byla k dispozici v roce 1999) je charakteristická nízkými nároky na napájení a spoluprací s malými koncovými zařízeními. Rychlosť na fyzické vrstvě dosahuje 1Mbps, přičemž skutečná propustnost dat se pohybuje maximálně kolem 720kbps. Komunikace po Bluetooth nabízí až tři hlasové kanály. Bluetooth pracuje podobně jako WLAN 802.11b v bezlicenčním pásmu 2.4GHz. Na rozdíl od 802.11b ale Bluetooth využívá metody rozprostřeného spektra s přeskakováním kmitočtů FHSS, kdy rádiový signál velmi rychle (1600krát za sekundu) náhodně přeskakuje mezi 79 jedno-MHz kanály.

Veškerou komunikaci v síti Bluetooth řídí hlavní stanice (master) prostřednictvím protokolu výzvy: podřízená stanice (slave) může komunikovat s ostatními výhradně prostřednictvím hlavní stanice.

Bluetooth pro zabezpečení používá stejný protokol jako WLAN, protokol WEP (Wired Equivalent Privacy) s 128bitovým klíčem. Menší dosah sítě ve srovnání se sítěmi 802.11 je s ohledem na bezpečnost velkou výhodou.

UWB a ZigBee

Mezi největší konkurenτy Bluetooth můžeme počítat především standardy IEEE 802.15.3 (UWB) a 802.15.4 (ZigBee).

Oba přitom nabízí poněkud jiné vlastnosti, než jaké má Bluetooth. Jeden z hlavních rozdílů je v přenosové rychlosti. UWB nabízí 100Mbps, což je přibližně sto-násobek přenosové rychlosti Bluetooth. ZigBee je ze všech těchto tří bezdrátových standardů nejpomalejší: 20 - 250kbps. Díky své jednoduchosti je ale také ze všech zde uvedených standardů nejlevnějším produktem.

ZigBee

Standard ZigBee, s přesným označením IEEE 802.15.4, je určen pro tzv. LR-WPAN (Low Rate Wireless Personal Area Network), tedy bezdrátové sítě s krátkým dosahem v řádu desítek metrů a s poměrně malými přenosovými rychlostmi (tab. 2.1) a je tedy přímým konkurentem standardu Bluetooth. Využívá metody rozprostřeného spektra DSSS stejně jako IEEE 802.11.

Byl vyvinut a sestaven institucí IEEE a dále je rozvíjen a podporován ZigBee Aliance (ZBA). ZBA sdružuje přední výrobce elektronických součástek a snaží se dosáhnout návrhu co nejjednodušší jednotky s nízkou spotřebou. ZBA byla založena v září roku 2002 a jde o asociaci společností pracujících společně na vytvoření standardu pro levný obousměrný bezdrátový přenos vyžadující minimum spotřeby elektrické energie. Mezi nejznámější členy ZBA patří společnosti Philips, Motorola a Honeywell. Podrobnější a úplný seznam všech členů aliance je uveden na internetových stránkách www.zigbee.org.

Podle standardu ZigBee se zařízení dělí na zařízení se všemi funkcemi FFD (Full-Functional Device) a zařízení s redukovanými funkcemi RFD (Reduced-Functionality Device). Každá síť pracující se standardem ZigBee musí obsahovat alespoň jedno zařízení RFD nebo FFD a řídící jednotku. Řídící jednotka řídí a koordinuje celou nebo určitou část sítě. Tyto funkce může plnit jen zařízení třídy FFD. Používané topologie sítí pracující se standardem ZigBee mohou být typu hvězda, strom nebo kombinace obou předchozích. V síti může být zapojeno najednou až 254 zařízení k jedné řídící jednotce. Pro srovnání: u technologie Bluetooth může v síti (Piconetu) pracovat najednou pouze osm zařízení.

Zařízení pracující podle standardu ZigBee mohou pracovat nejen v pásmu 2.4GHz (tab. 2.2), ale i v pásmu 915MHz (USA) nebo 868MHz (Evropa). Pro přenos informací v kmitočtovém pásmu 2.4 až 2.4835GHz je pro zařízení pracující podle standardu ZigBee k dispozici 16 přenosových subkanálů s odstupem 5MHz o celkové přenosové kapacitě 250kbps. V pásmu 868.3MHz je k dispozici jeden přenosový subkanál s celkovou přenosovou kapacitou 20kbps a v pásmu 902 až 928MHz je k dispozici přenosových subkanálů deset s odstupem 2MHz a celkovou přenosovou kapacitou 40kbps.

ZigBee IEEE 802.15.4			Bluetooth IEEE 802.15.1
20kbps	40kbps	250kbps	1Mbps
868MHz	915MHz (USA)	2.4MHz	

Tabulka 2.2: Porovnání přenosových rychlostí ZigBee a Bluetooth

Technologie ZigBee se nedá využít pro náročné datové přenosy, ale je vhodná pro osobní bezdrátové sítě WPAN (Wireless Personal Area Network) na větší vzdálenosti než technologie Bluetooth. ZigBee není technicky náročná technologie, má jednoduchou architekturu a vyžaduje minimální příkon elektrické energie - hodí se především pro automatizaci a regulaci v budovách a průmyslových provozech (klimatizace, osvětlení, různé senzory a snímače, zabezpečovací technika a podobně). Více o ZigBee v [13].

2.1.3 Srovnání

Bluetooth vs ZigBee

ZigBee je robustní protokol určený k stavovému přenosu informací nebo k bezpečnému přenosu dat nižší rychlostí. ZigBee přenáší stavy - přenáší informace o tom, že se zapnulo topení, nebo že kamera detekuje pohyb. Bluetooth oproti tomu je náhrada datových kabelů a slouží k propojení "vyšší elektroniky" - tedy k přenášení náročnějších datových objemů.

ZigBee se v mnohém podobá Bluetooth, standard je však optimalizován s ohledem na cenu a hlavně na nízkou spotřebu - pomocí ZigBee by měla komunikovat zařízení napájená několik let jednou baterií. ZigBee proto umožňuje režim hibernace, kdy modul naslouchá, ale spotřeba klesá na pouhý jeden či dva mikroampéry.

Bluetooth potřebuje 100 μ A, Wi-Fi desítky miliampérů.

Vlastnosti	Wi-Fi	Bluetooth	ZigBee
Doba provozu	hodiny	dny	roky
Počet zařízení SLAVE v síti	32	7	255
Dosah	100m	10m	70-300m
Přenosová rychlosť	11Mbps	1Mbps	250kbps

Tabulka 2.3: Obecné porovnání Wi-Fi, ZigBee a Bluetooth

Dalším podstatným rozdílem je lepší podpora časově kritických aplikací - výrazně nižší zpoždění při přenosu krátkých datových paketů, rychlejší "probuzení" a navázání spojení (tab. 2.4).

Akce	ZigBee	Bluetooth
Přidání nového Slave	30 ms	3s
Aktivace Slave z režimu Sleep	15 ms	3s

Tabulka 2.4: Porovnání ZigBee a Bluetooth

Jak je vidět z tabulky 2.2, ZigBee je koncipováno pro nižší přenosové rychlosti, takže se nehodí pro aplikace vyžadující rychlý přenos většího množství dat.

Bluetooth vs Wi-Fi

Bluetooth a Wi-Fi používají stejné frekvenční pásmo 2.4GHz. Koexistence technologií v těsné blízkosti je velice složitá, zařízení se totiž navzájem silně ruší. Bluetooth byl od samého začátku vyvýjen pro komunikaci mezi kapesními zařízeními na vzdálenost jednotek metrů. Této skutečnosti bylo přizpůsobeno vše. Vysílací výkony se pohybují v řádu mW (0,001W).

Technologie Wi-Fi byla primárně navržena jako alternativa lokálních počítačových sítí do míst, kde je složité, popřípadě nežádoucí pokládání kabelových rozvodů. Do vzdálenosti 300 metrů na volném prostranství nabízí Wi-Fi maximální přenosovou rychlosť 11Mbps, přičemž s použitím speciálních antén může být dosah prodloužen přibližně až na dva kilometry.

Bluetooth má nižší požadavky na napájení než Wi-Fi. Zařízení, která na principu Bluetooth pracují, mohou proto být menší, což je pro použití ve spotřební elektroinici velmi výhodné.

Bluetooth je vhodná volba pro jednoduchá zařízení, u nichž rychlosť není rozhodujícím faktorem; je nevhodnější pro PDA, mobily, digitální fotoaparáty a eventuálně pro propojení domácích spotřebičů, které od sebe nejsou příliš vzdáleny. Bluetooth naopak není dobrou náhradou kabelových vedení mezi počítači, popřípadě mezi zařízeními pracujícími s vysokými objemy dat, jako jsou například digitální videokamery. Tam je jednoznačně nevhodnější Wi-Fi.

Bluetooth byl vyvíjen s požadavky na jednoduchost, miniaturizaci a nízkou spotřebu, maximální přenosová kapacita byla méně důležitá než u bezdrátové počítačové sítě. Proto Bluetooth využívá modulaci FHSS a špatně se snáší s Wi-Fi. Řešení pro soužití obou technologií vyvíví někteří dodavatelé čipů pro komunikaci v pásmu 2.4GHz, jako jsou Texas Instruments, Intersil a Silicon Wave, o koexistenci Bluetooth s Wi-Fi se snaží i nově ustavená komise IEEE 802.15.2 Task Force. V nejjednodušším případě by mohl uživatel ručně přepínat mezi Wi-Fi a Bluetooth, uvažuje se i o rychlém přepínání mezi Wi-Fi a Bluetooth nebo o modifikaci omezující přeskakování nosné frekvence Bluetooth pouze na části kmitočtového spektra nevyužité Wi-Fi.

Další možnosti

K bezdrátovému přenosu informace lze použít i další technologie jako např. IrDA nebo magnetickou indukci.

IrDA je standard vytvořený IrDA konsorcium (Infrared Data Association), který definuje jak bezdrátově přenáset digitální data pomocí infračerveného záření. IrDA ve svých specifikacích definuje standardy jak fyzických koncových zařízení, tak protokolů jimiž komunikují IrDA zařízení. IrDA standard vznikl z potřeby mobilně propojit různá zařízení mezi sebou.

IrDA zařízení komunikují pomocí infračervených LED diod s vlnovými délками vyzařovaného světla $875\text{nm} \pm 30\text{nm}$. Na tuto vlnovou délku jsou citlivé i mnohé CCD kamery. Přijímačem jsou PIN fotodiody.

IrDA zařízení dle normy IrDA 1.0 a 1.1 pracují do vzdálenosti 1.0 m při bitové chybovosti BER³ 10-9 a maximální úrovní okolního osvětlení 10klux (denní svit slunce). Tyto hodnoty jsou definovány pro nesouosost vysílače a přijímače 15 stupňů, pro jednotlivé optické prvky se měří výkon do 30 stupňů. Existují směrové vysílače (IR LED) pro větší vzdálenosti, které nedodržují předepsaný úhel 30 stupňů od osy a pro který má vysílač útlum 3dB. Rychlosti jsou pro IrDA v. 1.0 od 2400 do 115200bps, používá se pulsní modulace 3/16 délky původní doby trvaní bitu. Formát dat je stejný jako na sériovém portu, tedy asynchronně vysílané slovo uvozené startbitem.

IrDA v. 1.1 definuje navíc rychlosti 0.576 a 1.152Mbps s pulsním kódovaním 1/4 délky doby trvaní původního bitu (střída 1/4). Při těchto rychlostech je již paket vysílán synchronně uvozen startovní sekvencí.

Integrované IrDA (kombinovaná vysílací infra LED dioda a přijímací PIN fotodioda) mají již zabudované filtry, které zabraňují rušení mimo frekvenční oblast IrDA 2400-115200bps a 0.576-4Mbps.

Magnetická indukce může sloužit pro dvoubodovou i vícebodovou komunikaci mezi zařízeními na velmi krátkou vzdálenost, čímž dokáže nahradit aplikace Bluetooth nebo může sloužit pro domácí sítě či jiné sítě s velmi malým dosahem.

Přenos dat je realizován po elektromagnetických vlnách pole vytvořeného magnetickou indukcí. Elektrický proud procházející magneticko-indukčním systémem

³bit error ratio, poměr chybně přenesených bitů ku správně přeneseným

vytváří magnetické pole a ASIC (Application Specific Integrated Circuit) pak moduluje signál tak, aby v sobě zakódoval přenášená data. Jedničky a nuly binárních dat reprezentují 90-ti stupňové posuny fáze signálu v jednom ze dvou směrů. Anténa pak zesiluje a vysílá signál.

Signál ovšem podléhá rychlému útlumu, takže dosah komunikace je pouhé dva až tři metry. Malý dosah ale znamená také menší náchylnost na odposlech, přičemž bezpečnost lze ještě zvýšit šifrováním a autentizací. Dosah sítě je ovšem kompenzován minimální spotřebou. Spotřeba energie u magneticko-indukčního systému je pětkrát až desetkrát nižší než u Bluetooth.

Magneticko-indukční systémy se nepotýkají s rušením, protože v jejich pásmu nepůsobí mnoho dalších systémů. Pracují totiž v pásmu 13.5MHz určeném pro účely ISM, kde žádná jiná komunikační síť zatím neoperuje. To je samozřejmě velká výzva systémům Wi-Fi nebo Bluetooth, které pracují vedle bezšňůrových telefonů a mikrovlneček v přetíženém volném pásmu 2.4GHz.

Magnetická indukce nemá problém s překážkami (narozdíl od např. infračervené komunikace, která pevnými překážkami neproniká, nebo rádiové komunikace, potýkající se s odrazy od překážek).

Magnetické pole je principiálně jednosměrné, takže je potřeba použít elektronické prvky a zpracování signálu pro nasměrování pole potřebným směrem.

Největším konkurentem magneticko-indukční komunikace je UWB. UWB je totiž podobně jako magnetická indukce velice bezpečné, nenáchylné k rušení a pracuje také na velmi krátkou vzdálenost několika metrů. Předností UWB je rychlosť, která je o mnoho řádů vyšší (100-450Mbps) než u magneticko-indukčních sítí (několik set kbps).

2.1.4 Shrnutí

Cílem této práce bylo bezdrátové řízení modelu automobilu za použití co nejvíce hodnější technologie.

Omezení:

- dosah alespoň 25m
- malá spotřeba (výkon)
- jednoduchost technologie (myšleno pro použití v modelu auta)
- dostupnost technologie

Model automobilu je určen pro výuku v laboratoři na Karlově náměstí. Tím je omezen maximální požadovaný dosah na zhruba 25m. Vzhledem ke členitosti "terénu" (stoly, roboti a další zařízení) je nepoužitelná technologie IrDA i technologie založená na magnetické indukci. Požadavky splňují ZigBee, Wi-Fi a Bluetooth.

Nejmenší spotřebu a nejjednodušší je ZigBee, následuje Bluetooth a kritéria ne splňuje Wi-Fi. Objem přenášených dat není potřeba velký, přenášet se budou pouze příkazy a stavy čidel, popřípadě baterií v modelu auta.

Nejlepším řešením by tedy byla technologie ZigBee, která má větší dosah než Bluetooth, je jednodušší a má i menší spotřebu. Zásadní rozdíl byl však v dostupnosti na trhu. Zatímco technologie Bluetooth byla již rozšířený standard, ZigBee byla teprve v počátcích. Specifikace ZigBee byla standardizována v prosinci roku 2004. Z tohoto důvodu byla pro tuto práci použita právě technologie Bluetooth.

2.2 Bluetooth

Bluetooth definuje globální standard (protokol) určený dvěma a více zařízením na krátké vzdálenosti prostřednictvím rádiového spojení. Používá se mezi mobilními telefony, počítači, PDA zařízeními, atd.

První základní kámen Bluetooth byl položen v květnu roku 1998, kdy vzniklo sdružení The Bluetooth special interest group (SIG). U jeho zrodu stály renomované společnosti Ericsson, IBM, Intel, Nokia a Toshiba. Přibližně za rok SIG připravila standard Bluetooth 1.0.

V roce 2000 se do vývoje tohoto bezdrátového standardu zapojilo Japonsko a Jižní Korea. Dalším důležitým mezníkem ve vývoji Bluetooth se stal březen roku 2001, kdy byla představena vylepšená verze Bluetooth 1.1.

V Bluetooth SIG jsou v této době zastoupené všechny významné společnosti z oblasti informačních technologií a komunikací jako jsou 3Com, Ericsson, IBM, Intel, Microsoft, Motorola, Nokia a dalších téměř 2000 společností.

Specifikace Bluetooth zahrnuje všechny části, které se zúčastňují komunikace od specifikace rádiových signálů až po protokoly vyšších vrstev. Informace uvedené v následujících kapitolách jsou důležité pro pochopení technologie Bluetooth. Více v [10].

2.2.1 Rádio specifikace

Bluetooth (přijímač a vysílač) operuje v bezlicenčním frekvenčním pásmu ISM (Industrial Scientific Medicine) na frekvenci 2.4GHz. Šířka frekvenčního pásma pro Bluetooth zařízení je ve většině států Evropy⁴ 2400 - 2483.5MHz. Šířka rádiového signálu je 1MHz, takže je k dispozici 79 rádiových kanálů. Pro nosné frekvence platí vztah:

$$f = 2402 + k \text{ [MHz]}, \text{ kde } k = 0, \dots, 78$$

Aby byly dodrženy normy mimo pásmo, jsou definována tzv. dolní a horní ochranná pásmá. Dolní ochranné pásmo má šířku 2MHz, horní ochranné pásmo 3.5MHz.

⁴Například Španělsko a Francie má toto pásmo v rozmezí 2446.5 - 2483.5MHz se šírkou kanálu opět 1MHz.

Výkonové třídy

Zařízení dodržující specifikaci Bluetooth jsou z hlediska vyzářeného výkonu rozdělena do tří kategorií (tab. 2.5).

Třída	Výstupní výkon			Dosah
	maximální	nominální	minimální	
1	100mW (20dBm)	nespecifikováno	1mW (0dBm)	100m
2	2.5mW (4dBm)	1mW (0dBm)	0.25mW (-6dBm)	10m
3	1mW (0dBm)	nespecifikováno	nespecifikováno	0.01m

Tabulka 2.5: Výkonové třídy Bluetooth

V České republice je maximální povolený výkon vyzářený zařízením pracujícím v tomto pásmu 100mW, minimální je nepovinný. U zařízení třídy 1 je požadováno řízení výstupního výkonu, které se používá pro omezení vysílačního výkonu nad úrovní 0dBm (1dBm). Pokud toto zařízení nepodporuje, smí vysílačí strana odpovídat pouze třídě 2 nebo 3. Zařízení s implementovaným řízením výkonu (spotřeby) optimalizuje výstupní výkon pomocí příkazů LMP (Link Manager Protocol).

Rušivé vyzařování

Rušivé vyzařování uvnitř i vně pracovního frekvenčního pásma se měří se skákáním nosné frekvence (frequency hopping) vysílače na jediné frekvenci. Znamená to, že kmitočtový syntezátor musí provádět přeskoky mezi vysláním a přijetím časového (time) slotu, ale v okamžiku vysílání nebo příjmu se musí vždy vrátit na původní frekvenci.

2.2.2 Baseband specifikace

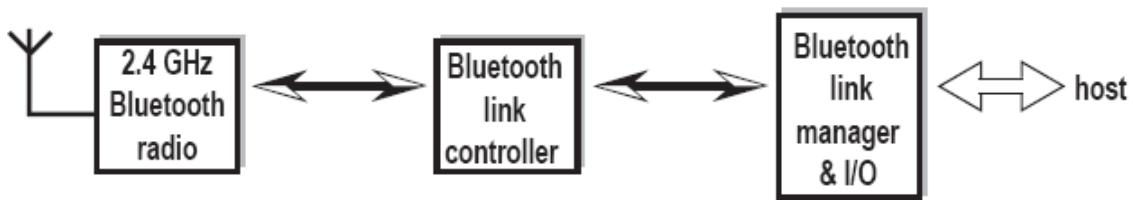
Baseband specifikuje fyzickou vrstvu technologie Bluetooth. Zabezpečuje činnost fyzických linek (spojení), korekci chyb, výběr skokové frekvence a poskytuje i základní funkce kódování. Baseband protokol je implementovaný jako řízení linky (Link Controller - LC), který zabezpečuje funkce jako vytváření spojení nebo řízení spotřeby. Kromě toho řídí také asynchronní a synchronní spojení, přijímá pakety, analyzuje (zkoumá a vyhodnocuje) přijímané pakety a také vyhledává jiná dostupná zařízení Bluetooth.

Jak již bylo řečeno, Bluetooth operuje v bezlicenčním pásmu 2.4GHz. Vzhledem k tomu, že toto pásmo využívá i několik dalších bezdrátových systémů (IEEE 802.11), je nutné zajistit, aby se systémy uvnitř kmitočtového pásma vzájemně nerušily. Technologie Bluetooth využívá nízké vysílační výkony v kombinaci s metodou frekvenčního skákání nosné (Frequency Hopping) v rozprostřeném spektru (FHSS). Nominální počet frekvenčních přeskoků je $1600s^{-1}$, doba obsazenosti rádiového kanálu je tedy velmi krátká - $625\mu s$. Frekvenčním skákáním nosné se zabraňuje interferencím (skládání) a kolísání rádiového signálu.

2.2.2.1 Komunikační jednotka Bluetooth

Komunikační jednotka Bluetooth obsahuje (obr. 2.1):

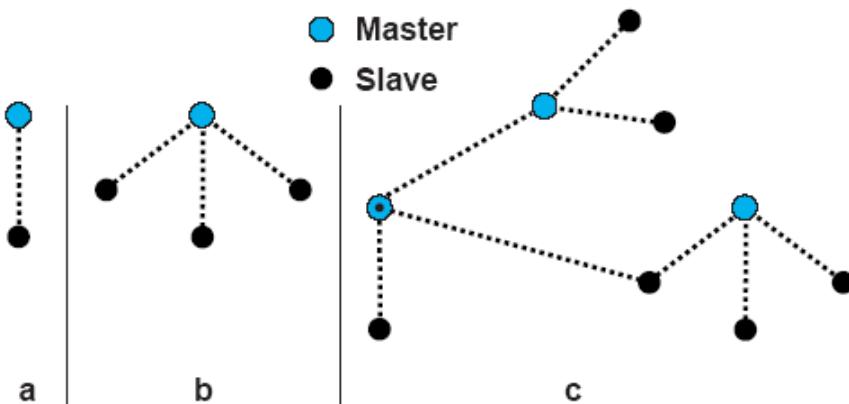
- rádiovou jednotku (Radio Unit - kapitola 2.2.1)
- jednotku řízení spojení (linkový ovladač - Link Control Unit, dále jen **LC**), která ovládá rádiový vysílač/přijímač
- správce linky a I/O⁵ obvodů spoje (Link Manager - dále **LM** a I/O), který zajišťuje komunikace mezi I/O obvody spoje a poskytuje uživateli terminálové rozhranní



Obrázek 2.1: Funkční bloky v systému Bluetooth

Bluetooth systém poskytuje 2 druhy spojení (propojení) :

- **point-to-point** (jeden k druhému)
- **point-to-multipoint** (jeden k více - Piconet, Scatternet)



Obrázek 2.2: Point-to-point (a), Piconet (b), Scatternet (c)

Spojení point-to-point (obr. 2.2a) je spojení dvou Bluetooth jednotek, z nichž jedna plní v rámci spojení roli **Mastera** (hlavní - vytváří spojení) a druhá roli **Slave**. Spojení point-to-multipoint je spojení jednoho Mastera s více zařízeními typu Slave.

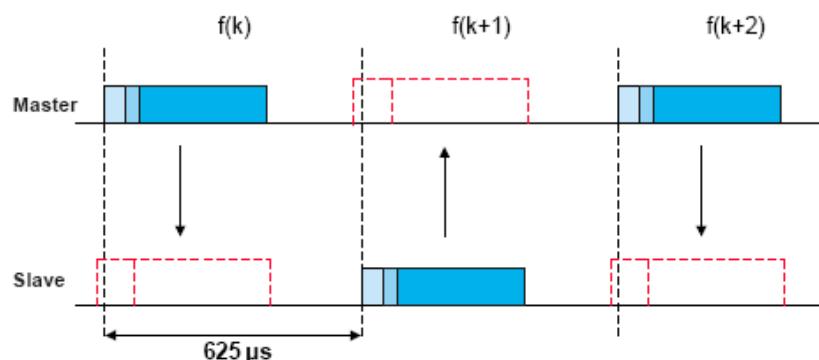
⁵Input/Output - vstup/výstup

Dvě a více Bluetooth jednotek, které sdílí stejný komunikační kanál, tvoří síť, tzv. **Piconet** (obrázek 2.2b). Až sedm Slave může být aktivních v Piconetu. Další Slave mohou být v rámci tohoto Piconetu pouze v režimu Parked (nejsou aktivní, ale stále synchronizovány s Masterem).

Dva a více navzájem provázaných Piconetů tvoří **Scatternet** (obr. 2.2c). Každý Piconet může mít pouze jednoho Mastera, ale jedno ze zařízení Slave může plnit roli Mastera v rámci jiného, přidruženého Piconetu.

2.2.2.2 Přenosový kanál

Kanál je reprezentován pseudonáhodnou sekvencí změn vysílací frekvence (přeskakující rádiová frekvence – frequency hopping) mezi frekvenčními pozicemi 0 až 78. Sekvence změn je jediněčná pro každý Piconet a je odvozena od adresy zařízení typu Master a fáze této sekvence je odvozena od jeho Bluetooth hodin. Přenosový kanál je rozdělen do stejně dlouhých časových rámci (slotů). Nominální délka časového rámce je $625\mu\text{s}$. Existují dva druhy časování kanálů – TDD (Time-Division Duplex) a multi-slot. Časování TDD (obrázek 2.3) odpovídá situaci, v níž se řídicí a řízená jednotka postupně střídají ve vysílání. Master začíná komunikaci v sudých časových slotech a Slave v lichých. Začátek paketu je přiřazen začátku časového slotu.



Obrázek 2.3: TDD schéma a časování

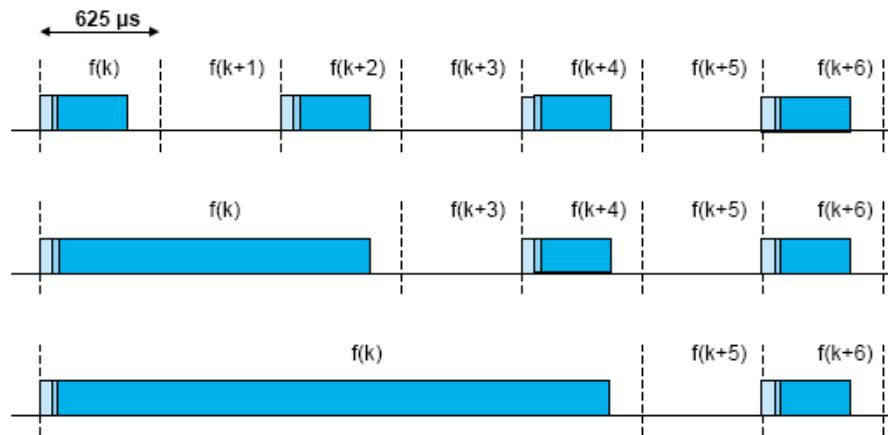
Časování multi-slot (obrázek 2.4) odpovídá využití přeskakové sekvence pro přenos tak, že paket posílaný zařízením typu Master nebo Slave může obsadit více než jeden rámeček (slot), maximálně pět.

Protokol Bluetooth využívá kombinaci kruhu (data jsou zasílány v rámci sítě postupně od jednoho k druhému až k adresátovi) a zasílání paketů (data jsou poslány přímo adresátovi).

2.2.2.3 Fyzické linky

Baseband poskytuje dva druhy linek (spojení), které se liší přenosovými schopnostmi:

- asynchronní (ACL - Asynchronous Connection - Less)



Obrázek 2.4: Multi-slot pakety

- synchronní (SCO - Synchronous Connection - Oriented)

Oba kanály je možné využít k zajištění přenosů dat podle požadavků. Pro zajištění komplikovanějších přenosů, kombinujících oba typy kanálů, je možné v průběhu spojení měnit typ kanálu.

SCO je symetrické point-to-point spojení mezi zařízeními Master a Slave v Piconetu s přenosovou rychlostí 64kbps. Mezi dvěma zařízeními mohou být tři SCO linky najednou. Využívají se pro časově závislé aplikace jako např. přenos zvuku a obrazu. Při SCO spojení Master posílá v pravidelných intervalech paket zařízení Slave, které odpovídá vždy v následujícím časovém slotu. Vytvoření SCO spojení je inicializované (vynucené) zařízením Master pomocí LMP.

ACL využívá časování multi-slot, přičemž je možné dosáhnout přenosové rychlosti 721kbps v jenom směru a 57.6kbps v opačném směru (asymetrický kanál), popř. 433kbps v obou směrech (symetrický kanál). Uvedené přenosové rychlosti platí za předpokladu, že se nevyužívá možnost opravy chyb při přenosu. ACL spojení může být jak point-to-point spojení mezi Master a Slave, tak i point-to-multipoint mezi Masterem a více Slave (max. 8). Mezi Masterem a jedním Slavem může existovat vždy jen jedno ACL spojení (narozdíl od SCO). Pokud není při vysílání paketu Masterem adresován žadný ze zařízení Slave, pak se jedná o zprávu Broadcast (určená všem). ACL spojení je vhodné k přenosu běžných dat a jeho výhodou je větší přenosová rychlosť. Využívá se také pro přenos dat zajišťujících a řídicích komunikaci v rámci buňky.

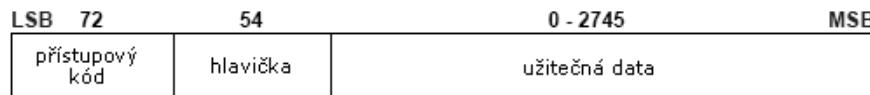
2.2.2.4 Pakety

Pro posílání paketů platí následující:

- nejméně významný bit (LSB - Least Significant Bit) odpovídá bitu b_0
- LSB je první bit, který se posílá

Pokud se například posílají 3 byty $b_2 b_1 b_0 = 110$, pak informace, která se zaslá je $b_0 b_1 b_2 = 011$. LC interpretuje první přijatý bit jako b_0 .

Data jsou rozdělena do paketů. Hlavní formát paketu je na obrázku 2.5



Obrázek 2.5: Formát standardního paketu

Přístupový kód (Access Code) a hlavička (Header) mají pevně danou velikost a to 72 a 54 bitů. Užitečná data (Payload) mohou mít 0 až 2745 bitů. Existují různé typy paketů, které budou popsány níže.

Přístupový kód

Každý paket začíná přístupovým kódem - kód používaný při komunikaci se zařízeními. Pokud za ním následuje hlavička paketu je dlouhý 72 bitů. Pokud ne, pak 68 bitů. Používá se pro synchronizaci a identifikaci. Přístupový kód určuje všechny pakety vyměněné v rámci kanálu Piconetu, tj. všem paketům poslaným v jednom Piconetu předchází stejný přístupový kód. Existují 3 druhy: CAC (Channel Access Code), DAC (Device Access Code) a IAC (Inquiry Access Code). CAC jednoznačně identifikuje vytvořený Piconet a obsahuje informace potřebné k synchronizaci komunikace v rámci Piconetu. DAC a IAC se používají při vyhledávání zařízení, která nejsou součástí jednoho Piconetu.

Hlavička paketu

Obsahuje LC informace a skládá se ze 6-ti částí:

- AM ADDR - 3 byty, adresa uzlu v rámci Piconetu (kapitola 2.2.2.6)
- TYPE - 4 byty, typ přenášeného paketu (Type Code)
- FLOW - 1 bit, příznak řízení toku (Flow Control), GO (Flow=1) , STOP (Flow=0)
- ARQN - 1 bit, potvrzení úspěšného přijetí paketu (Acknowledge Indication)
- SEQN - 1 bit, příznak, zda se jedná o opakováný rámec, nebo zda jde o další paket v pořadí (Sequence Number)
- HEC - 8 bitů, k zabezpečení kontroly integrity hlavičky (Header Error Check)

Celkově tedy obsahuje 18 bitů, které jsou však zakódovány poměrem 1/3 FEC⁶, což ve výsledku dává dříve zmíněných 54 bitů.

⁶Typ kódování vycházející z Hammingova kódování - výsledný kód je dán určitou funkcí.

Typy paketů

Pakety používané v Piconetu souvisí s použitím fyzického kanálu, tedy buď ACL nebo SCO. Pro každý z nich je definováno 12 různých typů paketů. Čtyři řídicí (Control) pakety jsou společné. K rozlišení typu paketu slouží 4 byty TYPE v hlavičce (Header) paketu.

2.2.2.5 Logické kanály

V systému Bluetooth je definováno 5 logických kanálů, které mohou být použité pro přenos různých typů informací:

- řídicí kanál LC (Link Control Control)
- řídicí kanál LM (Link Manager Control)
- uživatelský kanál UA (User Asynchronous Data)
- uživatelský kanál UI (User Isochronous Data)
- uživatelský kanál US (User Synchronous Data)

Řídicí kanály LC a LM jsou používány na úrovni LC a LM. Uživatelské kanály UA, UI a US jsou využity k přenosu asynchronní, isochronní a synchronní uživatelské informace.

2.2.2.6 Adresování zařízení

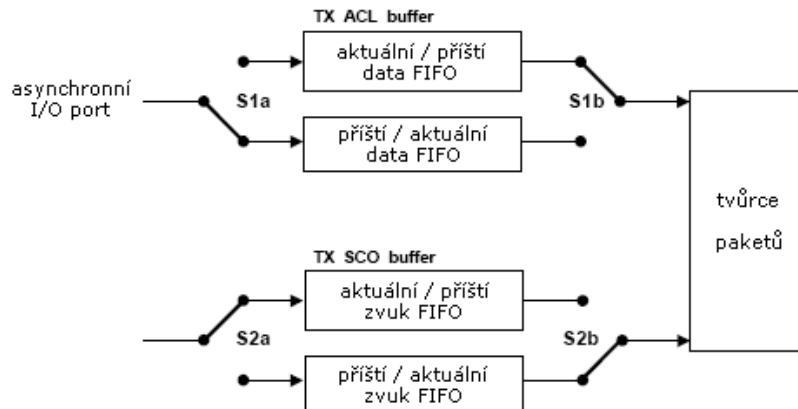
Existují 4 druhy adres, které mohou být přiřazeny zařízení Bluetooth:

- BD ADDR (Bluetooth Device Address) - fyzická jedinečná 48-bitová adresa.
- AM ADDR (Active Member Address) - 3-bitová adresa, která slouží k adresování zařízení Slave v rámci Piconetu. Je platná pouze po dobu existence Piconetu.
- PM ADDR (Parked Member Address) - 8-mi bitová adresa, kterou používá Master k adresování zařízení Slave, která jsou v rámci Piconetu v režimu PARKED. Adresa je platná jen po dobu setrvání Slave v tomto režimu.
- AR ADDR (Access Request Address) - tato adresa je používána zařízeními Slave v režimu PARKED při synchronizaci s Masterem.

2.2.2.7 Vysílací a přijímací proces

Vysílací proces (dále TX) je oddělený pro ACL a SCO spojení. Na obr. 2.6 je ukázán pouze jeden TX ACL buffer⁷ a jeden TX SCO buffer. Slave má buffer TX ACL právě jeden, Master má jeden pro každé zařízení Slave.

⁷Místo v paměti o dané velikosti sloužící k ukládání dat procesu - tzv. vyrovnávací paměť.



Obrázek 2.6: Funkční diagram vysílání

TX buffer se skládá ze dvou FIFO (First In, First Out - data, která byla vložena jako první, jsou jako první načtena) registrů: jeden je zpřístupněn Bluetooth řadiči, který ho čte a z dat tvoří pakety, a do druhého LM nahrává data. Mezi těmito dvěma registry se přepíná pomocí S1a,b, respektive S2a,b.

Přijímací proces (dále RX) pracuje obdobně, ale v opačném směru. Pakety přijaté z LC jsou načteny do jednoho bufferu a z druhého jsou LM převedeny na data.

2.2.2.8 Časování vysílání a příjmu

Bluetooth vysílač/přijímač používá TDD schéma (kapitola 2.2.2.2). V normálním stavu spojení vysílá Master v sudých časových rámcích a Slave v lichých.

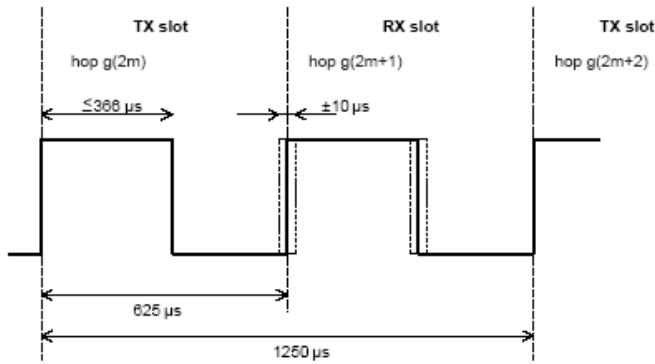
Časová synchronizace Master a Slave

Každá jednotka Bluetooth má své vlastní vnitřní systémové hodiny, které určují časování a skokovou frekvenci přijímače a vysílače.

Piconet je synchronizován systémovými hodinami Mastera. Master dodržuje přesný interval $m.625\mu s$ (m je sudé číslo a vždy větší než 0) mezi následujícími přenosy (vyslání) dat. Zařízení typu Slave přizpůsobí své systémové hodiny pomocí časového offsetu tak, aby byl synchronizován s Masterem. Toto se děje během vytváření spojení. Offset je poupraven vždy s příchozím paketem od Mastera.

Stav spojení

Ve stavu spojení Bluetooth střídavě vysílá a přijímá. Na obrázku 2.7 je ukázán Master při posílání a přijímání paketů, které zabírají jeden časový rámec. Maximální délka užitečných dat (Payload) je $366\mu s$. Každý nový RX i TX přenos je realizován na jiné skokové frekvenci (hop frequency). Vyjímkou jsou pakety, které zabírají více časových rámců (maximálně však 5). V tomto případě daný RX nebo TX přenos pokračuje na stejně skokové frekvenci na jaké začal.



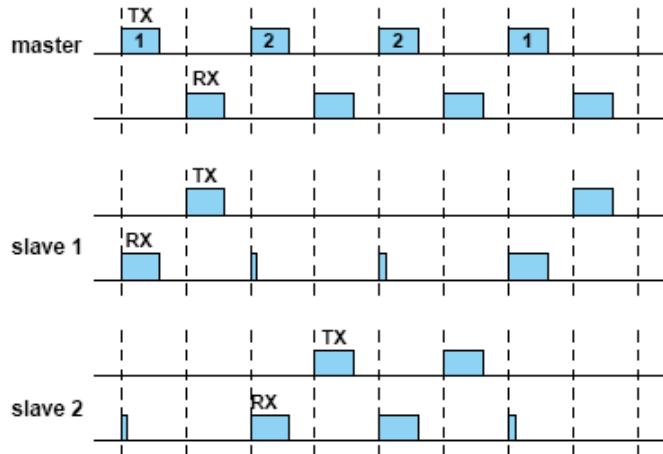
Obrázek 2.7: Cyklus TX a RX z pohledu Mastera

FHS pakety

Jsou zasílány od Mastera ke Slave během nastavování spojení nebo během proměny Mastera za Slave⁸. Slouží k frekvenční a časové synchronizaci.

Komunikace s více Slave

Pokud je k Masterovi připojeno více zařízení Slave, probíhá komunikace podle obrázku 2.8. Master zašle paket určitému zařízení Slave a ten paket přijme, pokud je adresátem (dáno adresou zařízení AM ADDR). Pokud není adresátem, skončí čtení paketu po přijetí AM ADDR.



Obrázek 2.8: Komunikace Mastera s více Slave

V případě broadcast (určená všem Slave) zprávy nemohou Slave odpovídat zasláním paketu zpět Masterovi.

⁸Pokud je povoleno, může dojít během spojení k výměně rolí obou zařízení, tzn. Master se stane Slave a opačně.

2.2.2.9 Stavy zařízení a řízení kanálu

Zařízení Bluetooth se může vyskytovat ve dvou základních stavech: Standby a Connection. Kromě těchto existuje ještě dalších 7 „podstavů“, které slouží k přidávání zařízení Slave do Piconetu, nebo k vytváření nových Piconetů.

Standby je počáteční stav zařízení Bluetooth s nízkou spotřebou energie. Běží pouze vnitřní hodiny a neprobíhá komunikace s okolím. Ve stavu Connection (spojení) je umožněno zařízením Slave a Master komunikovat v rámci jednoho Piconetu.

2.2.2.10 Vytvoření spojení

Vytvoření spojení probíhá dvěma možnými způsoby. Pokud o sobě zařízení nemají žádné informace (BD ADDR, hodinový offset), pak se vykonají po sobě procedury *Inquiry* a *Page*. Pokud zařízení už byly spojené, tzn. mají potřebné informace (BD ADDR a offset), pak je potřeba vykonat jen proceduru *Page*.

Procedura **Inquiry** zabezpečuje nalezení okolních zařízení a výměnu jejich HW adres BD ADDR a hodnot hodinových signálů (Clock offset).

Procedura **Page** slouží k vytvoření spojení (komunikační linky) mezi dvěma zařízeními. Většinou následuje po proceduře Inquiry. Aby mohla být vykonána je potřeba znát BD ADDR. Hodnota hodinového signálu nemusí být známa, pouze urychluje vytvoření spojení. Zařízení, které inicializovalo vytvoření spojení, vykonáním procedury Page se automaticky stává **Masterem**.

2.2.2.11 Režimy zařízení ve stavu *Connection*

Zařízení ve stavu *Connection* může být v jednom ze 4 režimů: Active, Hold, Sniff a Parked.

- **Active** - v tomto režimu se zařízení aktivně účastní komunikace. Master plánuje komunikaci na základě požadavků zařízení Slave. Aktivní zařízení Slave poslouchá komunikaci ve slotu *master-to-slave* - sudé a pokud není adresováno může „spát“ až do dalšího přenosu od Mastera.
- **Hold** - zařízení Slave v tomto režimu udržuje synchronizaci s Piconetem pomocí vnitřních hodin. Do tohoto režimu se Slave dostane na příkaz zařízení Master.
- **Sniff** - Slave se v tomto režimu synchronizuje s Piconetem se sníženou intenzitou (nepřijímá pakety tak často), což mu umožňuje snížit spotřebu energie.
- **Parked** - Slave je stále synchronizován s Piconetem, ale neúčastní se komunikace. Zařízení v tomto režimu má aktivovanou AM ADDR adresu a resynchronizuje se občasným příjmem broadcast zpráv od Mastera.

2.2.3 Link Manager Protocol

Protokol správce linky (LMP - Link Manager protokol) zabezpečuje konfigurování linky (spojení), synchronizaci zařízení při vytváření spojení (ACL i SCO), šifrování spojení, detekci a opravy chyb a další funkce. Signály jsou zpracovány a filtrovány správcem linky LM (Link Manager) v rámci příjmu dat a nejsou zpřístupněny vyšším vrstvám.

LMP zprávy jsou přenášeny v rámci užitečných dat (payload) L2CAP (Logical Link Control and Adaptation). Zprávy zasílané LM mají větší prioritu než uživatelská data.

Mezi činnosti, které LM vykonává patří:

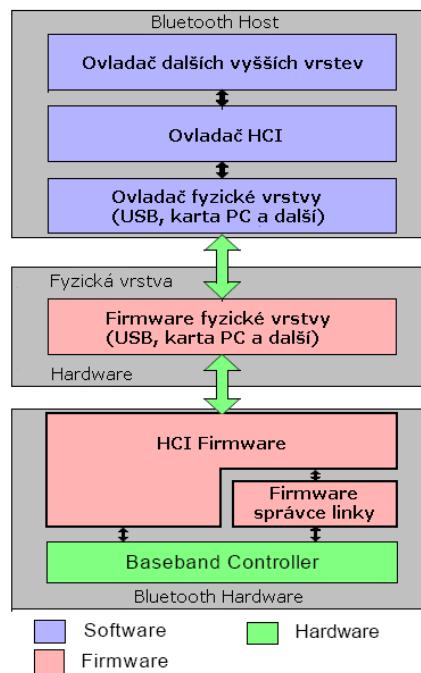
- párování (Pairing) - vytváří se náhodný klíč linky (link key) společný pro obě zařízení
- autentifikace (Authentication) - ověření druhé Bluetooth jednotky
- změna klíče linky (Change Link Key)
- kódování (Encryption)
- žádost o zaslání offsetu hodin (Clock Offset)
- informace o názvu vzdálené jednotky
- uvedení do režimu HOLD, SNIFF nebo PARKED
- kontrola kvality linky

2.2.4 HCI specifikace

HCI (Host Control Interface - rozhraní pro řízení hostem) poskytuje příkazové (programové) rozhraní pro přístup k funkcím, které poskytují Baseband a správce linky LM. Jedná se o rozhraní, které poskytuje jednotný přístup softwarové části k fyzické části zařízení Bluetooth (HW a firmware).

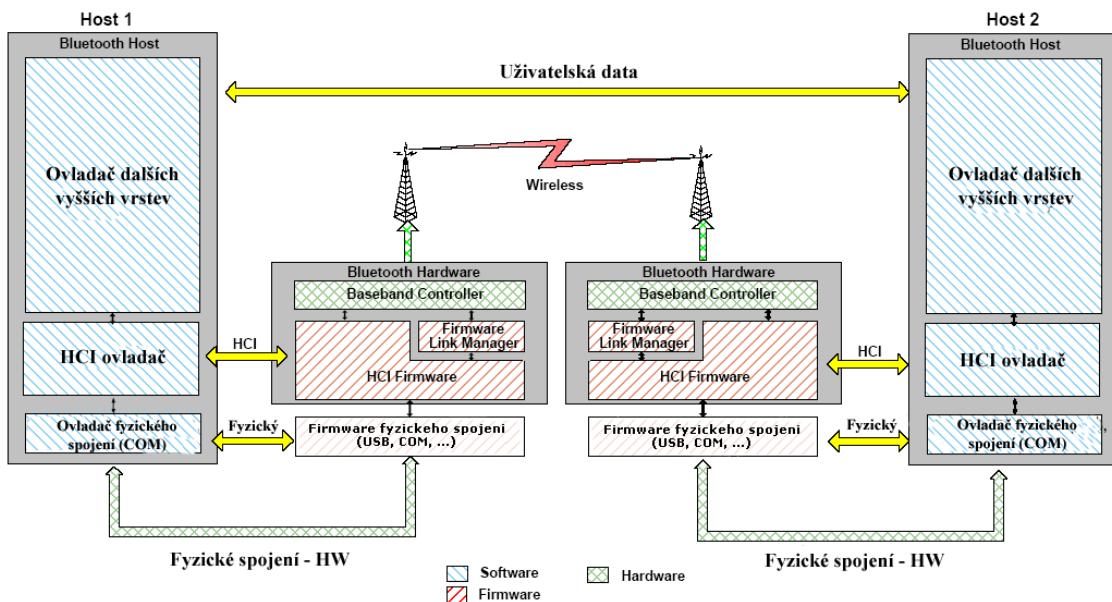
Funkce HCI je rozdělena do třech oddělených částí: Host, Transportní vrstva a Host Controller (obr. 2.9)

- HCI firmware - nachází se v části Host Controller (Bluetooth HW). Implementuje HCI příkazy prostřednictvím příkazů Baseband a LM vrstvy, přístupu k HW stavovým a řídicím registrům a k registrům událostí.
- HCI driver - nachází se v části Host (SW). Zabezpečuje analýzu přijatých událostí, na jejichž základě poskytuje informace vyšším vrstvám. Zprostředkovává komunikaci s vyššími vrstvami.
- komunikace HCI driver a HCI firmware - komunikují přes HC transportní vrstvu. Transportní vrstva definuje několik různých způsobů komunikace: USB, UART a RS232.



Obrázek 2.9: Přehled nižších softwarových vrstev

Obrázek 2.10 ukazuje přenos dat z jednoho zařízení do druhého. Ovladač HCI u Hosta (Hostem může být například počítač nebo mobilní telefon) si vyměňuje data a příkazy s HCI firmwarem, který je součástí HW Bluetooth. Ovladač fyzické vrstvy, neboli vrstva řízení a kontroly transportu u Hosta poskytuje oběma HCI vrstvám možnost výměny informací mezi sebou.



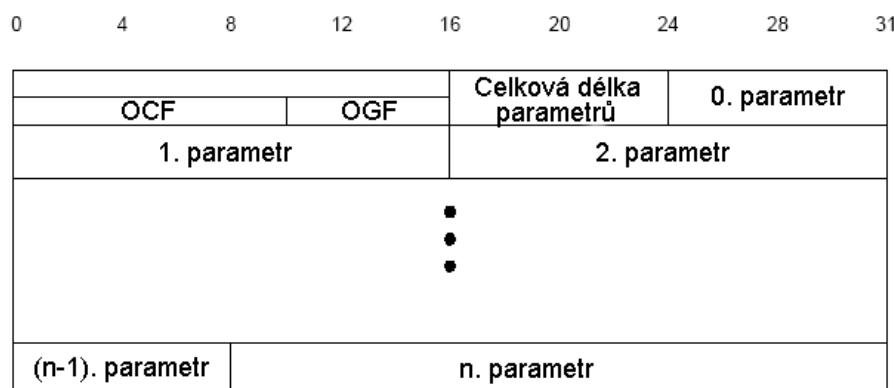
Obrázek 2.10: Přehled nižších softwarových vrstev vzhledem k přenosu informace

2.2.4.1 HCI příkazy

HC rozhraní poskytuje jednotné metody příkazů k přístupu ke schopnostem HW Bluetooth. Příkazy (Commands) HCI poskytují Hostovi schopnost řídit spojení (spojovací vrstvu) se vzdáleným Bluetooth zařízením. Tyto příkazy se většinou týkají správce spojení LM, který si na jejich základě vyměňuje LMP příkazy se vzdáleným Bluetooth zařízením.

Nejmenší jednotka dat, která může být přenesena jedním zařízením do druhého je Baseband paket. Klasický paket je zpráva vyššího protokolu než je baseband. Každý z HCI příkazů potřebuje ke svému splnění různý čas. Splnění příkazu je Hostovi oznámeno událostí (HCI Event). Vše okolo HCI událostí je vysvětleno v kapitole [2.2.4.2](#).

Pakety s HCI příkazy jsou zasílány Hostem do HC (Host Controller) přes transportní vrstvu (RS232). Formát paketu je na obr. [2.11](#). První 2 byty tvoří kód (OpCode), který určuje o jaký příkaz se jedná (do jaké skupiny příkazů patří a jeho číslo v této skupině). Každá skupina je označena jedinečným číslem OGF (OpCode Group Filed), které spolu s označením příkazu ve skupině OCF (OpCode Command Field) tvoří právě OpCode.



Obrázek 2.11: Stavba paketu HCI příkazu

Pro správné provedení příkazu je nutné správně složit OpCode na základě OGF a OCF. Postup složení OpCode HCI příkazu *Reset*:

V případě příkazu *Reset* je OGF 0x03 a OCF 0x0003. OpCode je tvořen 10-ti byty OCF a 6-ti byty OGF. OGF 0x03 lze zapsat bitově 00000011. Protože protokol Bluetooth používá formát přenosu dat tzv. Little Endian (nejdříve se přenáší LSB) je potřeba byty otočit - 11000000. OGF má v OpCode 6 bitů, takže ořízneme dva nejnižší byty. Výsledkem je **110000**. OCF 0x0003 je bitově 0000000000000011. Opět otočíme: 1100000000000000 a ořízneme na výsledných 10 bitů: **1100000000**. Následuje složení bitů v pořadí OCF + OGF: 1100000000 a 110000 = 11000000 00110000. Nakonec je potřeba byty zpětně otočit: 00001100 00000011.

Výsledkem jsou 2 byty: **0x03** a **0x0C**. Pokud tedy posíláme příkaz HCI Reset, první a druhý byte jsou 030C.

Třetí byte (délka parametrů) určuje, kolik následuje bytů s parametry. Samotné parametry jsou dány HCI příkazem, který chceme HC poslat. Parametry jsou dvojího typu. První jsou parametry potřebné k vykonání daného příkazu (Command Parameters), a které se zasílají v paketu příkazu. Druhým typem jsou návratové parametry, tzn. parametry, které budou navíc obsahovat událost (Event).

HCI příkazy jsou rozděleny do 6 skupin:

- příkazy k řízení spojení (Link Control Commands)
- strategické (zásadní) příkazy (Link Policy Commands)
- příkazy HC a baseband (Host Controller and Baseband Commands)
- příkazy k získání informací o parametrech (Informational Parameters)
- příkazy ke zjištění různých stavů (Status Parameters)
- příkazy k testování (Testing Commands)

Příkazy LINK CONTROL

Tyto HCI příkazy umožňují HC řídit spojení se vzdáleným Bluetooth zařízením. Při použití těchto příkazů řídí správce linky LM vytvoření a udržení spojení v síti Piconet, popř. Scatternet. OGF je definováno jako 0x01. Podrobný popis jejich parametrů naleznete v [10].

Příkazy, které patří do této skupiny, a které jsou v práci použity:

Inquiry - OCF 0x0001

HCI příkaz, který způsobí přechod Bluetooth zařízení do režimu Inquiry - vyhledávání dostupných Bluetooth zařízení. Parametry příkazu jsou LAP, doba vyhledávání a počet odpovědí.

Návratové parametry u příkazu Inquiry nejsou. Odpověď je událost *Command Status* ve chvíli, kdy HC spustí režim Inquiry. Po skončení vyhledávání pošle HC Hostovi událost *Inquiry Result* (pokud je v dosahu vzdálené zařízení), která obsahuje informace o tomto zařízení. Nakonec přijde událost *Command Complete*, která Hostovi oznamuje dokončení vyhledávání.

Příklad zaslání příkazu: **01 04 05 33 8B 9E 0A 08** (0104 - OpCode, 05 - délka parametrů, 338B9E - LAP, 0A = 10x1.28 = 12.8s dlouhá doba vyhledávání zařízení, 08 - počet možných odpovědí na Inquiry).

Inquiry Cancel - OCF 0x0002

HCI příkaz, který zastaví probíhající režim Inquiry - ukončí vyhledávání. Lze použít pouze po příkazu *Inquiry*. Příkaz má pouze jeden návratový parametr - Status (1 byte, pokud je 0x00, pak příkaz proběhl úspěšně), který přijde s událostí *Command Complete*.

Příklad zaslání příkazu: **02 04 00** (0204 - OpCode, 00 - délka parametrů).

Create Connection - OCF 0x0005

Příkaz způsobí, že správce spojení LM začne vytvářet spojení s vybraným Bluetooth zařízením. Parametry příkazu jsou např. BD ADDR, typ paketu a hodinový offset.

Návratové parametry nemá. Pokud HC přijme tento příkaz, zašle událost *Command Status* a pokud dojde k vytvoření spojení, pak i událost *Connection Complete*. Stejnou událost pošle i vzdálený HC vzdálenému hostovi.

Příklad zaslání příkazu: **05 04 0D B2 A1 5E 06 00 08 18 CC 00 00 0E 96 00** (0504 - OpCode, 0D - délka parametrů (13), B2A15E060008 - BD ADDR, 18CC - typ paketu, 00 - režim opakování Page Scan R0, 00 - režim Page Scan (hlavní), 0E96 - hodinový offset, 00 - zakázání výměny rolí).

Disconnect - OCF 0x0006

Příkaz k ukončení existujícího spojení. Parametry příkazu jsou Connection Handle a důvod ukončení spojení.

Příkaz Disconnect je bez návratových parametrů.

Příklad zaslání příkazu: **06 04 03 28 00 13** (0604 - OpCode, 03 - délka parametrů, 2800 - Connecion Handle, 13 - vzdálené zařízení ukončilo spojení).

Remote Name Request - OCF 0x0019

Příkaz určený k získání informace o jménu vzdáleného Bluetooth zařízení. Lze poslat pouze, pokud je spojení vytvořeno. HC po přijetí tohoto příkazu zasílá událost *Command Status*.

Příkazy LINK POLICY - OGF 0x02

Poskytuje Hostovi možnost ovlivnit, jak správce spojení LM pracuje s Piconetem. Patří sem příkazy, které uvádí Bluetooth zařízení do režimů HOLD (příkaz *Hold Mode*), SNIFF (příkaz *Sniff Mode*), atd.

Příkazy HC A BASEBAND - OGF 0x03

Poskytuje přístup k řízení různých schopností Bluetooth HW. Používají se ke změně chování lokálního Bluetooth modulu.

Reset - OCF 0x0003

Příkaz způsobí reset HC a správce spojení LM. Bluetooth modul se nastaví podle vlastností ve specifikaci. Návratovým parametrem je Status, který je obsažen v události *Command Complete*, pokud k resetu došlo.

Příklad zaslání příkazu: **03 0C 00** (030C - OpCode, 00 - délka parametrů).

Set Event Filter - OCF 0x0005

Příkaz, kterým se nastavuje: jaká zařízení budou "vyslyšena" při Inquiry, se kterými

zařízeními může být navázáno spojení a povoleno či zakázáno automatické vytvoření spojení s Masterem. Používá se pro nastavení zařízení typu Slave.

Příklad zaslání příkazu: **05 0C 03 02 00 02** (050C - OpCode, 03 - délka parametrů, 02 - typ filtru nastavování spojení, 00 - povolení spojení se všemi zařízeními bez omezení, 02 - automatické spojení).

Change Local Name - OCF 0x0013

Příkaz umožňující změnit název lokálního Bluetooth modulu. Název může být dlouhý 248 bytů. Jediným parametrem příkazu je požadované jméno. Návratovým parametrem je Status (událost *Command Complete*).

Příklad zaslání příkazu: **13 0C 0C 0A 62 74 72 63 2D 73 6C 61 76 65** (130C - OpCode, 0C - délka parametrů, 0A627472632D736C617665 - 'Bluetooth slave').

Read Local Name - OCF 0x0014

Příkaz zjišťující název lokálního Bluetooth modulu. Návratovými parametry jsou Status a jméno (*Command Complete*).

Příklad zaslání příkazu: **14 0C 00** (140C - OpCode, 00 - délka parametrů).

Write Scan Enable - OCF 0x001A

Opět příkaz pro zařízení typu Slave, který určuje, zda bude povolen či zakázán Inquiry Scan, popř. Page Scan. Základní nastavení (po resetu) zakazuje veškeré skeny, tzn., že se Bluetooth modul neodpovídá Masterovi při skenu. Parametrem je povolení či zakázaní daného skenu, návratovým parametrem je Status (*Command Complete*).

Příklad zaslání příkazu: **1A 0C 01 03** (1A0C - OpCode, 01 - délka parametrů, 03 - oba skeny povoleny).

Write Page Scan Activity - OCF 0x001C

Příkaz, který nastavuje vlastnosti Page Scan. Opět pro zařízení typu Slave.

Příklad zaslání příkazu: **1C 0C 04 40 00 20 00** (1C0C - OpCode, 04 - délka parametrů, 4000 - interval Page skenu = $40 \times 0.625\text{ms} = 64\text{ms}$, 2000 - okno Page skenu = $32 \times 0.625\text{ms} = 20\text{ms}$).

Write Inquiry Scan Activity - OCF 0x001E

Podobný jako předchozí, ale pro Inquiry Scan.

Příklad zaslání příkazu: **1E 0C 04 40 00 20 00** (1A0C - OpCode, 04 - délka parametrů, 4000 - interval Inquiry skenu = $40 \times 0.625\text{ms} = 64\text{ms}$, 2000 - okno Inquiry skenu = $32 \times 0.625\text{ms} = 20\text{ms}$).

INFORMAČNÍ PŘÍKAZY - OGF 0x04

Poskytuje informace o Bluetooth zařízení a vlastnostech HC.

Read Buffer Size - OCF 0x0005

Příkaz, který slouží ke zjištění maximálního množství dat ACL a SCO paketů, které

může být zasláno od HC k Hostovi. Návratovými parametry jsou maximální hodnoty a Status (*Command Complete*).

Příklad zaslání příkazu: **05 10 00** (0510 - OpCode, 00 - délka parametrů).

Read BD ADDR - OCF 0x0009

K přečtení 6-ti bytové BD adresy lokálního Bluetooth modulu. Návratové parametry jsou BD ADDR a Status.

Příklad zaslání příkazu: **09 10 00** (0910 - OpCode, 00 - délka parametrů).

Příkazy STATUS - OGF 0x05

K zjištění současného stavu HC a LM.

Get Quality Link - OCF 0x0003

Příkaz, který vrací hodnotu kvality vytvořeného spojení. Parametrem je Connection Handle daného spojení a návratovými Status, Connection Handle a kvalita linky, která může být v rozsahu hodnot 0 - 255.

Příklad zaslání příkazu: **03 14 02 28 00** (0314 - OpCode, 02 - délka parametrů, 2800 - Connection Handle).

2.2.4.2 HCI události

Pakety HCI událostí používá HC k ohlášení Hostovi, že došlo k nějaké události. Host musí být schopen přijmout paket o velikosti až 255 byte. Na každý příkaz, který Host zašle, obrází jednu či více událostí. Jaké události máme očekávat na zaslaný příkaz je uvedeno v kapitole [2.2.4.1](#) u jednotlivých příkazů.

Stavba paketu HCI událostí je na obr. [2.12](#). Každá událost je označena kódem (tedy něco jako OpCode u příkazů). Následuje délka parametrů. Parametry jsou opět dvojího typu. Prvním typem jsou parametry samotné události, druhým typem jsou návratové parametry příkazů.

Specifikace Bluetooth 1.1, ze které vychází, definuje 32 různých událostí označených 0x01 až 0x20. Opět se zmíní jen o těch, se kterými se v této práci pracuje.

Inquiry Complete - 0x01

Událost, která indikuje, že byl ukončen proces vyhledávání *Inquiry*. Jejím parametrem je Status (1 byte, pokud proběhlo v pořádku, pak je Status 0x00).

Příklad přijaté události: **01 01 00** (01 - kód, 01 - délka, 00 - Status)

Inquiry Result - 0x02

Indikuje, že na Inquiry odpovědělo jedno nebo více vzdálených zařízení. Třetí až sedmnáctý byte přijdou kolikrát, kolik zařízení na *Inquiry* odpovědělo. Parametry příkazu jsou např. počet odpovědí, BD ADDR jednotlivých zařízení a hodinový offset.

			0	4	8	12	16	20	24	28	31
Kód události	Délka parametrů		0. parametr události								
1. parametr události			2. parametr události								
•											
•											
•											
(n-1). parametr události			n. parametr události								

Obrázek 2.12: Stavba paketu HCI události

Příklad přijaté události: **02 0F 01 B2 A1 5E 06 00 08 01 00 00 00 00 0E 16** (02 - kód, 0F - délka (15), 01 - počet zařízení, které odpověděli na Inquiry, B2A15E060008 - BD ADDR vzdálené zařízení, 010000 - vlastnosti Page skenu, 000000 - třída zařízení, 0E16 - hodinový offset).

Connection Complete - 0x03

Indikuje oběma hostům (hostovi každé zúčastněné strany), že bylo vytvořeno spojení mezi oběma zařízeními. Parametry příkazu jsou Status, Connection Handle, BD ADDR a typ spojení.

Příklad přijaté události: **03 0B 00 28 00 B2 A1 5E 06 00 08 01 00**(03 - kód, 0B - délka (11), 00 - Status (OK), 2800 - Connection Handle, B2A15E060008 - BD ADDR, 01 - ACL spojení, 00 - kódování zakázáno).

Connection Request - 0x04

Indikuje nově příchozí žádost pokusu o spojení. Žádosti může být vyhověno nebo může být odmítnuta. Parametry jsou BD ADDR, třída zařízení a typ spojení.

Disconnection Complete - 0x05

Indikuje ukončení spojení mezi dvěma zařízeními. Parametry jsou Status (zda bylo rozpojení úspěšné), Connection Handle a důvod.

Příklad přijaté události: **05 04 00 28 00 13**(03 - kód, 04 - délka, 00 - Status (OK), 2800 - Connection Handle, 13 - spojení ukončeno vzdáleným zařízením).

Remote Name Request Complete - 0x07

Indikuje, že žádosti o zjištění jména vzdáleného zařízení bylo vyhověno. Parametry jsou Status, BD ADDR a jméno vzdáleného zařízení.

Příklad přijaté události: **07 FF 00 B2 A1 5E 06 00 08 0A 62 74 72 63 2D 73 6C 61 76 65 00 00 ... 00**(07 - kód, FF - délka (256), 00 - Status (OK), B2A15E060008 - BD ADDR, 0A 62 74 72 63 2D 73 6C 61 76 65 00 00 ... 00 - jméno, kde každý byte určuje ordinální hodnotu znaku).

Command Complete - 0x0E

Je používán HC jako odpověď na spoustu příkazů, kdy vrací Status a další parametry. Indikuje dokončení příkazu. Parametry jsou: počet paketů HCI příkazů (počet paketů, které může Host posílat HC), OpCode příkazu (OpCode příkazu, na který je tato událost odpovědí) a návratové parametry příkazů.

Příklad přijaté události: **0E 04 05 0C 00**(0E - kód, 04 - délka, 05 0C - OpCode příkazu (Set Event Filter), 00 - Status (návratový parametr příkazu Set Event Filter)).

Command Status - 0x0F

Indikuje, že HC příkaz přijal a začíná s jeho realizací. Je nutný pro asynchronní operace. Pokud příkaz nemůže být vykonán, je zaslána tato událost s číslem chyby (Status). Parametry jsou: Status, počet paketů HCI příkazů a OpCode příkazu.

Příklad přijaté události: **0F 04 00 01 05 04**(0F - kód, 04 - délka, 00 - Status (OK), 01 - počet možných paketů, 0504 OpCode příkazu (Create Connection)).

Hardware Error - 0x10

Zasílá HC v případě, že došlo k HW chybě. Parametrem je číslo chyby.

Příklad přijaté události: **10 01 00**(10 - kód, 01 - délka, 00 - specifické číslo).

Number Of Completed Packets - 0x13

Indikuje, kolik HCI datových paketů bylo přijato a odesláno HC od posledního zaslání této události. Na každý zasláný datový paket přijde Hostovi tato událost. Parametry jsou: počet Connection Handle, Connection Handle a počet odeslaných paketů.

Příklad přijaté události: **13 04 01 28 00 01**(13 - kód, 04 - délka, 01 - počet spojení, 2800 - Connection Handle, 01 - 1 přijatý a odeslaný paket).

2.2.4.3 HCI datové pakety

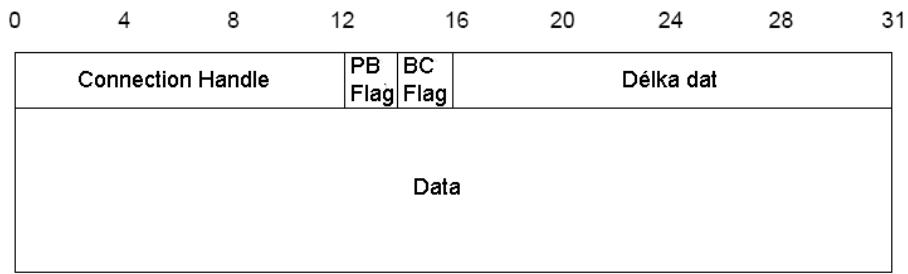
Datové pakety slouží k výměně dat (informací) mezi HC a Hostem. Datové pakety je možné zasílat, až ve chvíli, kdy je vytvořeno spojení se vzdáleným Bluetooth zařízením.

ACL datové pakety

ACL datový paket je na obr. [2.13](#).

První 2 byty tvoří Connection Handle, tedy jedinečné číslo spojení, spolu s dalšími čtyřmi byty, kterými se určuje o jaká data jde a zda jde o první datový paket. Rozdíl mezi ACL a SCO datovým paketem je právě v těchto čtyřech bitech - u SCO paketu jsou trvale nulové. Pak již následují 2 byty určující délku dat a samotná data.

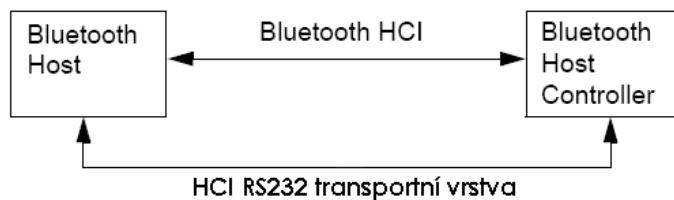
Příklad datového paketu: **28 20 01 44**(2820 - Connection Handle + flag byty (první dat. paket), 01 - délka, 44 - zaslana data).



Obrázek 2.13: HCI ACL datový paket

2.2.5 HCI RS232 transportní vrstva

Tato transportní vrstva slouží k fyzickému přenosu dat mezi Hostem a HC (obr. 2.14).



Obrázek 2.14: HCI RS232 transportní vrstva

Základní pravidla:

- pakety HCI příkazů mohou být posílány pouze Hostem
- pakety HCI událostí mohou být zasílány pouze HC
- datové pakety mohou být zasílány oběma směry

HCI není schopné rozlišit typ paketu (příkaz, událost, ...). Proto před tím, než je paket odeslán, je potřeba přidat na začátek paketu identifikátor (1 byte - tabulka 2.6).

Typ HCI paketu	Identifikátor
HCI příkaz	0x01
HCI ACL data	0x02
HCI SCO data	0x03
HCI událost	0x04

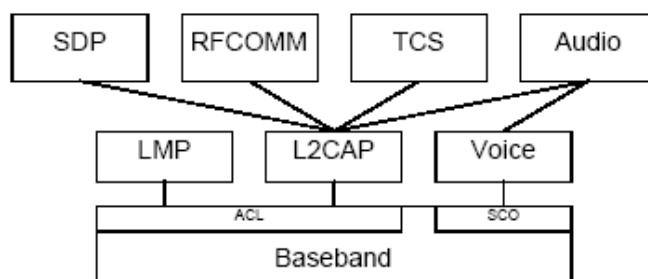
Tabulka 2.6: Identifikátory HCI paketů při komunikaci přes RS232

Například HCI příkaz *Reset* je přes RS232 zasílán: 01 03 0C 00 (01 - HCI příkaz, 030C - OpCode, 00 - délka).

2.2.6 L2CAP

L2CAP je protokol logického řízení a adaptace spojení (Logical Link Control And Adaptation Protocol). Podporuje protokoly vyšších vrstev, segmentování (rozdělení) paketů a jejich opětovné složení.

Tento protokol je nad protokolem Baseband a patří do vrstvy datového spojení (Baseband patří jak do datového, tak i do fyzického spojení). Poskytuje služby na přenos se spojením (Connection-Oriented) a bez spojení (Connection-Less) vyšším vrstvám. Existují dva typy spojení: asynchronní ACL a synchronní SCO. L2CAP podporuje pouze první z nich - ACL (Asynchronous Connection-Less). Kam patří



Obrázek 2.15: L2CAP v architektuře Bluetooth protokolu

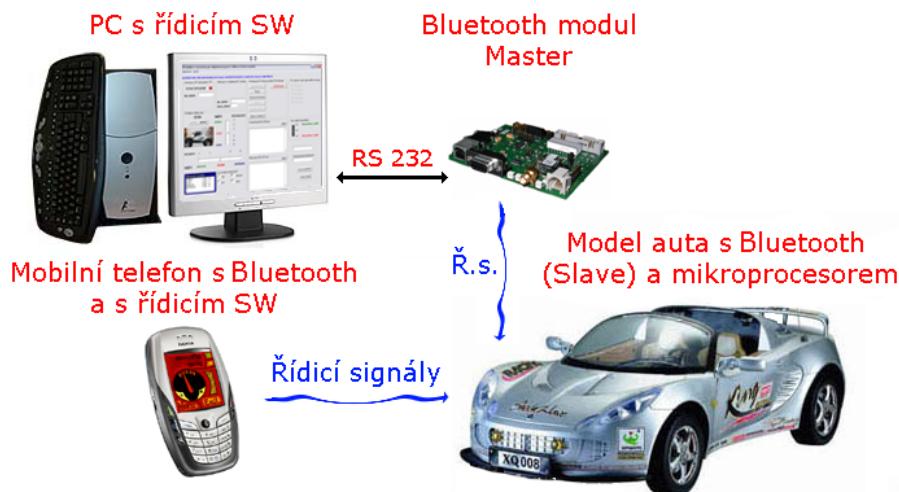
L2CAP v celkové architektuře protokolu Bluetooth je vidět na obrázku 2.15. SDP (Service Discovery Protokol), RFCOMM a TCS jsou vyšší protokoly, které nejsou pro tuto práci důležité a jsou vysvětleny ve specifikaci Bluetooth.

Kapitola 3

Realizace vzdáleného řízení

Realizace vzdáleného řízení se skládá ze 2 částí:

- bezdrátové řízení modelu auta pomocí počítače, popř. mobilního telefonu



Obrázek 3.1: Soustava bezdrátového řízení

- řízení modelu auta pomocí optických čidel



Obrázek 3.2: Řízení pomocí čidel

3.1 Modely aut

Realizace vzdáleného řízení pomocí počítače s modulem Bluetooth a řízení mobilním telefonem byly v rámci dvou současně běžících projektů. Prvním projektem byla diplomová práce "Model Remote Control" (dále DP), druhým byl projekt "Bluetooth Remote Control" (BTRC) pro Výzkumné a vývojové centrum v Praze (www.rdc.cz).

Pro každý z těchto projektů jsem měl k dispozici jiný model automobilu, na kterých byly provedeny odlišné úpravy. Společným cílem obou projektů byla demonstrace bezdrátového řízení zvoleného modelu. Cílem DP byla navíc možnost řízení pomocí optických čidel. Úpravy modelů se lišily v závislosti na splnění zmiňovaných cílů.

3.1.1 Model auta pro DP

Model auta (obr. 3.3) byl zakoupen školou a jeho výběr jsem nemohl ovlivnit. Jednalo se o RC¹ model. Poháněn byl stejnosměrným motorem, a to pouze jednou rychlostí vpřed a vzad. Zatáčení bylo řešeno pomocí cívky s vinutím a dvěma magnety připojenými k přední nápravě. V závislosti na polaritě vinutí cívky byl přitahován první nebo druhý magnet. Na plošné desce byl vysílač pro příjem příkazů z vysílače. Model byl napájen čtyřmi dobíjecími monočlánky, tzn. v nabitém stavu 4x1.2V.



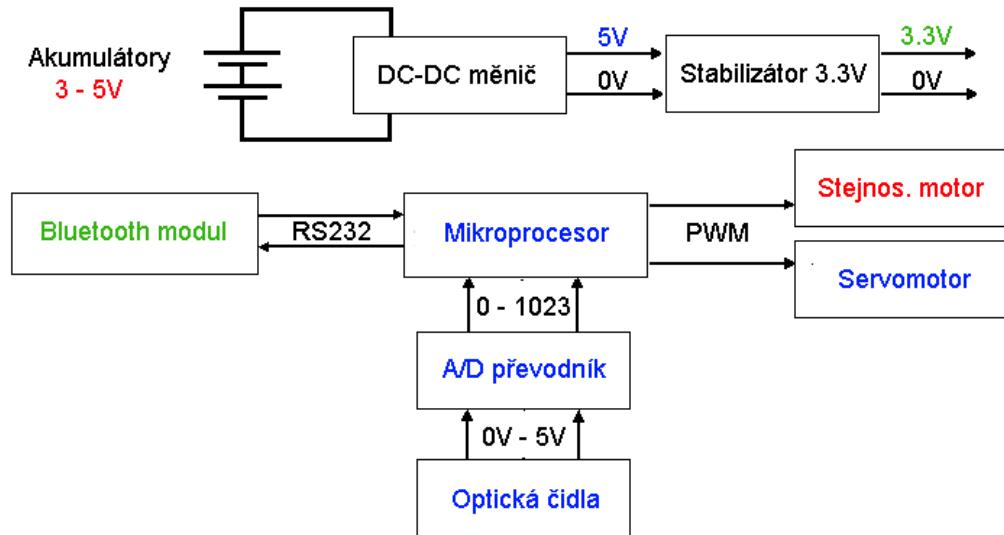
Obrázek 3.3: Model auta pro DP

3.1.1.1 Potřebné úpravy

Aby byl model lépe ovladatelný, bylo třeba přidat k motoru další elektroniku pro pohyb různými rychlostmi a oběma směry. Podobně i zatáčení bylo potřeba změnit tak, aby bylo možné nastavit více úhlů natočení vlevo a vpravo. Jelikož stávající řešení nebylo možné upravit, aby vyhovovalo zadání, bylo nutné ho nově navrhnut. Kromě úpravy akčních členů, bylo potřeba přidat dálkový přijímač a vysílač (Bluetooth modul) včetně mikroprocesoru, který bude komunikovat s Bluetooth modulem

¹Radio Control model - řízen radiovými signály v pásmu 27MHz

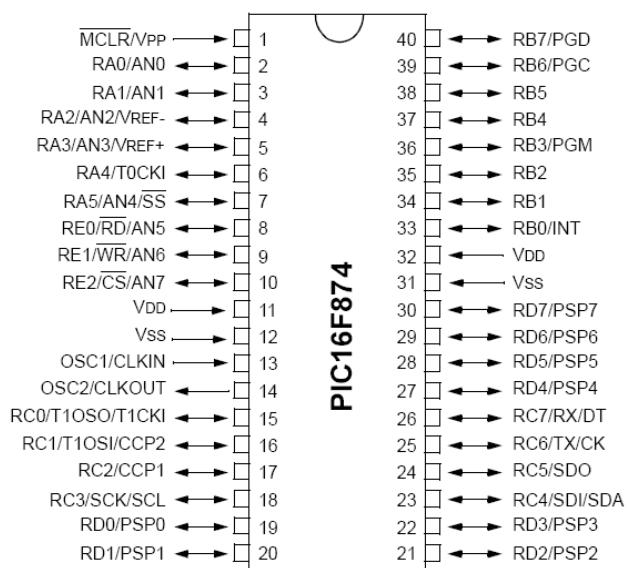
a na základě přijatých dat ovládat motory a posílat v pravidelných časových intervalech informaci o stavu nabití akumulátorů do počítače. Blokové schéma elektroniky v modelu auta je na obr. 3.4. Barevně jsou odlišeny napájecí úrovně.



Obrázek 3.4: Schéma elektroniky

3.1.1.2 Mikroprocesor

Výrobců mikroprocesorů je celá řada: Microchip, Atmel, Elan, Nec, Intel, atd.



Obrázek 3.5: Mikroprocesor PIC 16F874

Z celé řady těchto výrobců jsem zvolil firmu Microchip. Hlavním důvodem bylo, že mikroprocesory této firmy se používají v rámci výuky na katedře a tento model

má sloužit pro výuku. Ze stejného důvodu jsem vybral i mikroprocesor PIC 16F874 (obrázek 3.5), se kterým studenti pracují a zároveň vyhovuje požadavkům této úlohy.

Základní parametry PIC16F874:

- RISC procesor ²
- 35 instrukcí (jednocyklové, instrukce větvení dvoucyklové)
- externí frekvence až DC 20MHz
- 8-mi úrovňový zásobník
- až 14 zdrojů přerušení
- přímé, nepřímé adresování
- napájení od 2.5V do 5.5V, nízká spotřeba (do 0.6mA při 3V a 4MHz)
- proudové zatížení I/O³ pinů (vývodů) $\pm 25\text{mA}$
- 10-bitový A/D⁴ převodník - 8 možných vstupů
- pět I/O portů: A,B,C,D,E
- 2 piny (TX, RX) pro komunikaci přes RS232
- datová paměť 192 bytů

V tabulce 3.1 jsou vysvětleny jednotlivé piny, které jsem v práci využil. Zapojení mikroprocesoru s krystalem je součástí přílohy. Krystal jsem použil externí s maximální možnou frekvencí 20MHz - jeden strojový cyklus mikroprocesoru trvá 200ns ($1/20000000/4[\text{s}]$).

3.1.1.3 Bluetooth modul

Bluetooth modul plní v modelu auta roli rádiového přijímače a vysílače. S mikroprocesorem je propojen sériovým rozhraním (kapitola 3.1.1.2).



Obrázek 3.6: Bluetooth modul v modelu auta

²Procesor s redukovanou (omezenou) instrukční sadou.

³I - input, vstupní; O - output, výstupní

⁴Analogově-číslicový převodník

PIN	Název	Použití
1	VPP	Vstup, reset
2	RA0/AN0	Vstup A/D převodníku, připojen k akumulátorům v modelu
3	RA1/AN1	Vstup A/D převodníku, připojen k 1. optickému čidlu
4	RA2/AN2	Vstup A/D převodníku, připojen k 2. optickému čidlu
5	RA3/AN3	Vstup A/D převodníku
8	VSS	Vstup, napájení 5V
9	OSC1	Vstup hodinového oscilátoru, signál z ext. krystalu 20MHz
11	VDD	Vstup, zem, 0V
12	VSS	Vstup, napájení 5V
24	RC5	Výstup, HW reset modulu Bluetooth
25	TX	Výstup, komunikace s modulem Bluetooth přes RS232
26	RX	Vstup, komunikace s modulem Bluetooth přes RS232
31	VSS	Vstup, napájení 5V
32	VDD	Vstup, zem, 0V
34	RB1	Výstup, řízení stejnosměrného motoru směrem vpřed
35	RB2	Výstup, řízení stejnosměrného motoru směrem vzad
37	RB4	Výstup, řízení servo motoru
38	RB5	Vstup/výstup
39	RB6	Výstup, signalizace LED diody o režimu řízení
40	RB7	Výstup, signalizace LED diody o stavu spojení

Tabulka 3.1: Popis použitých pinů mikroprocesoru PIC

Modul firmy Siemens, rozšířený o anténu a s vyvedenými piny, byl zakoupen od firmy Czechlabs⁵ a je na obrázku 3.6. Umístěn je v modelu auta a v komunikaci plní roli Slave⁶. V tabulce 3.2 jsou uvedeny významy jednotlivých pinů Bluetooth modulu, které jsou v práci využity (číslovány zleva doprava).

PIN	Název	Použití
1	VCC	Napájení (3.3V)
2	CTS	Řízení přenosu (handshake)
3	RTS	Řízení přenosu (handshake)
4	TX	Vysílání - komunikace s mikroprocesorem přes RS232
5	RX	Příjem - komunikace s mikroprocesorem přes RS232
7	GND	Zem, logická 0
17	Reset	Hardwareový reset

Tabulka 3.2: Popis pinů použitého modulu Bluetooth

Bluetooth modul vyžaduje napájení 3.3V, které získáme z 5V zapojením napěťo-

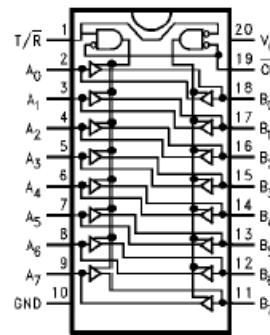
⁵Více informací na www.czechlabs.cz.

⁶Slave - poslouchá pokyny hlavního modulu - Mastera.

vého stabilizátoru LF33CDT ihned na výstup dc-dc měniče.

Mikroprocesor PIC je napájen 5V - logická jednička na výstupním pinu odpovídá tomuto napětí. Komunikace přes RS232 probíha tedy na úrovni 0V a 5V. Narozdíl od modulu Bluetooth, který má úrovně 0V a 3.3V.

Z toho důvodu bylo potřeba upravit 5V úroveň z PICu (pin TX) na úroveň 3.3V, aby mohla být propojena se vstupem Bluetooth modulu (TX). Opačným směrem není úprava úrovně signálu potřeba, protože PIC rozpoznává 3.3V již jako logickou jedničku.



Obrázek 3.7: Schéma 74LVX245

Funkci převodníku 5V na 3.3V řeší integrovaný obvod 74LVX245, jehož schéma je na obrázku 3.7. Popis jednotlivých vývodů je v tabulce 3.3.

PIN	Název	Použití
1	T/R	Směr přenosu
2-9	A0-A9	Vstupní/výstupní piny
10	GND	Zem, 0V
11-18	B0-B9	Výstupní/vstupní piny
19	CE	Povolení přenosu
20	VCC	Napájení 3.3V

Tabulka 3.3: Popis pinů převodníku 74LVX245

V zavislosti na vstupu T/R je určen směr převodu (pro T/R = 1 je směr z A_i do B_i). Vstupní napětí, které musí být v rozsahu 0V až 5.5V je převáděno na napájecí napětí. V našem případě z napětí 5V na napětí 3.3V, kterým musí být obvod napájen.

3.1.1.4 Akční člen - servomotor

Jak již bylo řečeno, cílem úpravy zatáčení bylo získat více poloh natočení přední nápravy. To znamenalo nahradu stávající konstrukce. Náš požadavek splňuje servomotor, který umožňuje nastavit zadanou polohu.

Požadavky na servomotor byly dané modelem auta: malé rozměry, napájení maximálně 5V a takovou sílu, aby zvládl točit přední nápravou i na místě. Model

auta váží cca 1 kilogram. Volba nakonec padla na servo HS-81 firmy HITEC (obrázek 3.8). Řídicí vstup je řízen pomocí pulsní šířkové modulace PWM (Pulse Width Modulation - obr. 3.9).



Obrázek 3.8: Servomotor

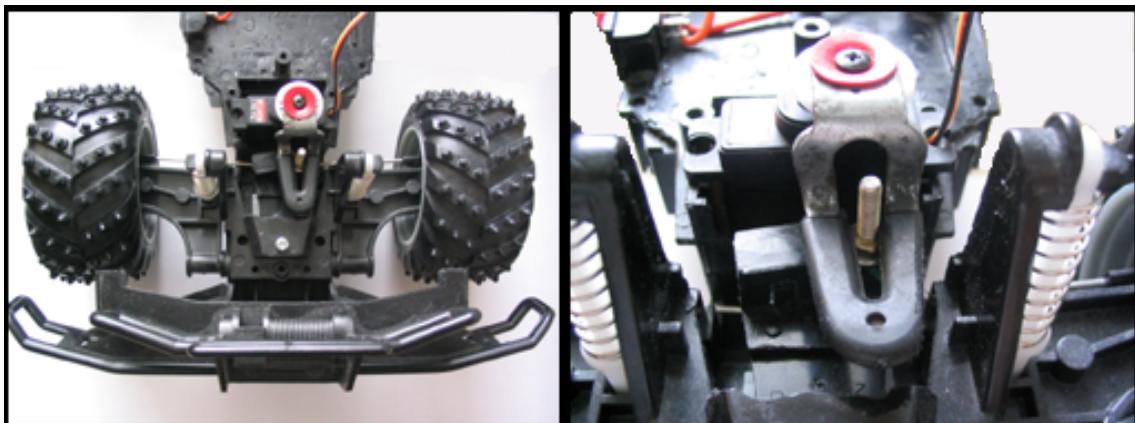
Parametry servomotoru:

- 3 vstupy: napájení 4.8 - 6V, zem a řídicí vstup
- pracuje v teplotním rozsahu -20°C až $+60^{\circ}\text{C}$
- síla 2.6N.m (při 5V)
- řízení 0.9 - 2.1ms, kdy 1.5ms je neutrální poloha a perioda 20ms
- odběr při běhu 220mA
- váha 16.6g



Obrázek 3.9: Časové průběhy PWM pro servomotor

Největším problémem bylo uchycení servomotoru v modelu a jeho spojení s přední nápravou. Možných řešení existuje několik, ale většina nešla realizovat z důvodu špatné konstrukce modelu, která neumožňovala proporcionální zatáčení. Část z umělé hmoty, ve které byl umístěn magnet, byla připevněna přímo k táhlům přední nápravy a vykazovala velikou vůli.



Obrázek 3.10: Způsob připojení serva k přední nápravě

Nejlepším technickým řešením bylo přichytit servomotor přímo k těmto táhům, ale toto bylo vzhledem k parametrům modelu nerealizovatelné. Proto nejlepším řešením bylo použití čepu (šroubu navrtaného místo magnetu) a vidlice, která je připevněna k servomotoru (obr. 3.10).

I toto řešení však neumožňuje více jak 2 polohy vlevo a vpravo.

3.1.1.5 Akční člen - ss motor

Zatímco u zatáčení bylo potřeba elektroniku nahradit, u stejnosměrného motoru bylo možné ponechat použitý motor a jen doplnit elektroniku, která by umožňovala řídit motor oběma směry pomocí PWM.

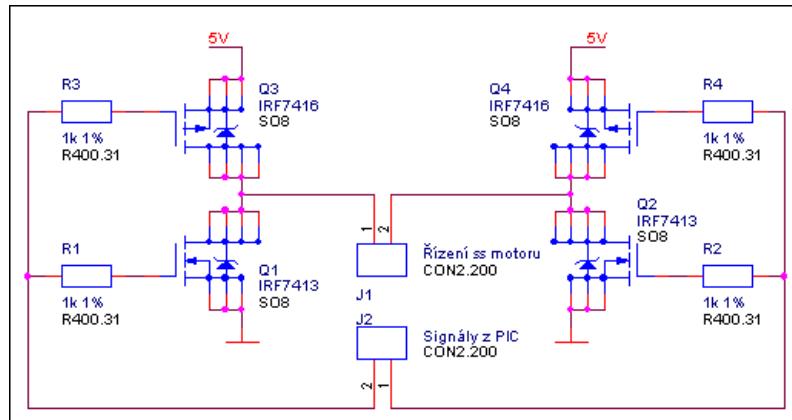
Stejnosměrný motor roztočíme na různé strany změnou polarity vstupního napětí. Tohoto dosáhneme můstkem podle [2]. Zapojení takového obvodu z diskrétních součástek je složitější oproti použití speciálního integrovaného obvodu, např. obvodu L293b.

Zapojení s tímto obvodem bylo součástí návrhu desky plošných spojů, ale ukázalo se jako nedokonalé. Na tomto obvodu se „ztrácí“ napětí (saturační) úměrné procházejícímu proudu a v našem případě (0.5A) to bylo až 2.6V. Ve výsledku byl stejnosměrný motor napájen pouze 2.4V a dodávaný proud nepřekročil 0.4A. Bohužel i další integrované obvody pro řízení PWM byly dimenzovány pro napájení alespoň 12V.

Stejnosměrný motor v modelu auta potřebuje pro okamžitý rozjezd špičkově až 5A, v ustáleném stavu (při běhu) cca 0.5A. Pro možnost řídit motor pomocí PWM, bylo nutné sestavit můstek, který je schopen potřebný proud dodat. Z toho důvodu není možné použít bipolární tranzistory, u kterých se v otevřeném stavu projevuje saturace a nedojde k jejich úplnému otevření, což způsobí omezení proudu do motoru a samozřejmě pokles napětí, které je na motor přivedeno.

Proto je konečným řešením můstek realizovaný z dvojic unipolárních tranzistorů MOSFET ([4]). Konkrétně jsem použil tranzistory IRF 7416 (typ P) a 7413 (typ N) pro každou větev můstku. Jejich důležitou vlastností je „propustnost“ 5A (Drain - Source) již při napětí 3V (Gate - Source). Zapojení můstku je vidět na obr.

3.11. Můstek je napájen přímo z akumulátorů (tedy vstup dc-dc), aby proudově nezatěžoval dc-dc měnič.



Obrázek 3.11: Schéma zapojení H-můstku pro řízení ss motoru

3.1.1.6 A/D převodník

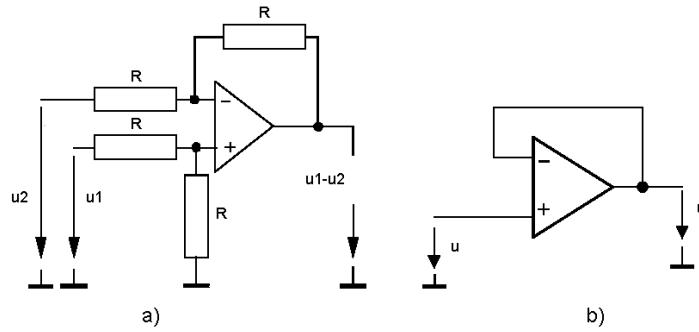
A/D převodník převádí napěťovou úroveň na odpovídající číslicovou hodnotu v závislosti na nastavené referenci a je přímo součástí mikroprocesoru. Referenční hodnoty napětí mohou být dány napájecím napětím (použito v této práci) nebo jinými hodnotami napětí přivedenými na referenční vstupy. V tomto případě je spodní mez dána 0V a horní 5V, takže minimální hodnota 0 odpovídá 0V a maximální hodnota 1024 odpovídá 5V. Číslo 1024 proto, že jsme použili 10-ti bitový převodník.

A/D převodník jsem použil ze dvou důvodů. Prvním je potřeba zobrazení aktuální informace o stavu napětí akumulátorů. Druhým důvodem je potřeba získání číselné informace z optických čidel. Zatímco informace o stavu akumulátorů je mikroprocesorem pouze poslána do PC, informace z čidel jsou mikroprocesorem využívány a pokud je model ve stavu automatického řízení, rozhoduje na jejich základě o řízení motorů.

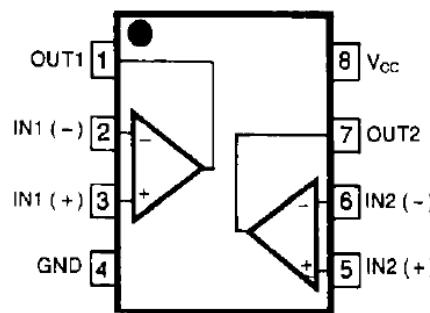
Proto, abychom mohli napěťovou úroveň přivádět na analogový vstup mikroprocesoru, musíme zajistit její dostatečně malý odpor. K tomuto účelu slouží zapojení s operačním zesilovačem [2]. K úpravě signálů z čidel jsem použil zapojení rozdílového zesilovače (obr. 3.12a) se zesílením rovno jedné. Tím na výstupu získám rozdílové napětí a s velmi malým odporem ([2] a [3]).

Napěťová úrověň z akumulátorů je nejdříve zapojena k odporovému děliči (poměr 1:2), aby v případě připojení jiného zdroje energie, nedošlo k připojení více jak 5V na vstup mikroprocesoru (osetření pro napájení do 11V). Výsledek je přiveden na vstup napěťového sledovače (obr. 3.12b), čímž opět zajistíme velmi malý výstupní odpor.

Operační zesilovače (dále OZ) jsou realizovány integrovaným obvodem NE5532 (obr. 3.13), který obsahuje 2 OZ. Pro zapojení signálů z čidel a akumulátorů jsou potřeba dva tyto obvody.



Obrázek 3.12: Diferenční zesilovač a), napěťový sledovač b)



Obrázek 3.13: Integrovaný obvod NE5532

Jednotlivé výstupy ze zesilovačů je možné přivézt přímo na vstupy mikroprocesoru a dále řešit softwarově.

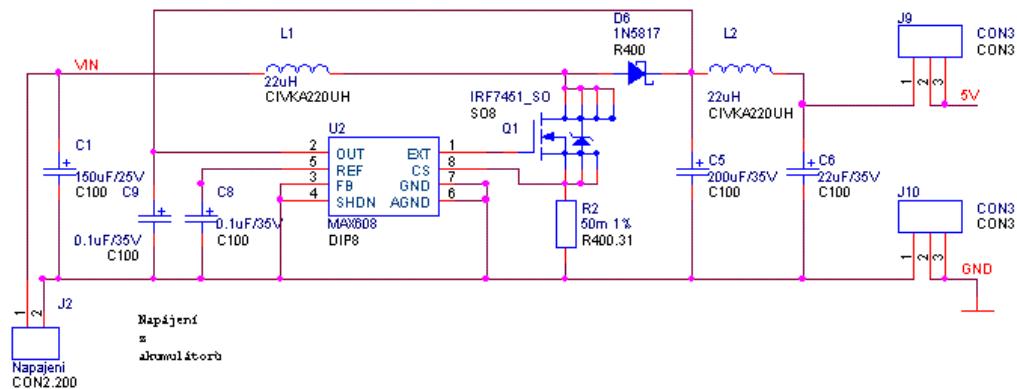
3.1.1.7 DC-DC měnič

Aby mohl navržený elektronický systém fungovat a být napájen akumulátory i při jejich částečném vybití, bylo potřeba stabilizovat 5V pomocí bloku s integrovaným obvodem MAX608. Zapojení tohoto obvodu je zvyšující dc-dc měnič⁷ [2]. Schéma zapojení je na obr. 3.14. Při výměně akumulátoru za typ s vyšším napětím je možnost oddělit tento dc-dc měnič od ostatní elektroniky na desce a přivést vlastní napájení. K tomuto slouží jumpery (přepínače) J9 a J10.

Za tímto dc-dc měničem je připojen LC filtr (L2, C6), který snižuje zvlnění a překmity výstupního napětí. Hodnoty indukčnosti a kapacity byly stanoveny experimentálně a to s výsledkem: zvlnění napětí max. 100mV a maximální překmit 250mV se střední hodnotou 5.07V při zátěži 200mA.

Indukčnosti použité v tomto zapojení musí mít velmi malý vnitřní odpor. Použil jsem indukčnosti s odporem 0.1Ω , které jsou k dostání u firmy Spezial (www.spezial.cz). MAX608 není v České republice běžně k dostání a bylo nutné ho jako vzorek ob-

⁷Měni stejnosměrné napětí opět na stejnosměrné, vyšší nebo stejně hodnoty.



Obrázek 3.14: Schéma zapojení obvodu dc-dc

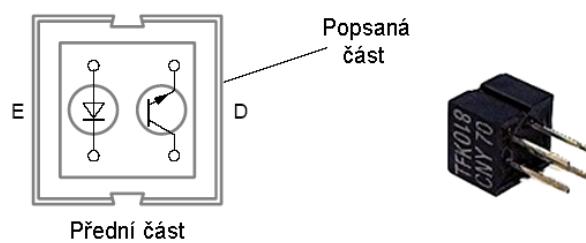
jednat přímo od výrobce (<http://www.maxim-ic.com>). Mosfet pro toto zapojení byl použit s rozhodovací úrovni 2.7V (LL logika) a opět s malým vnitřním odporem.

3.1.1.8 Optická čidla

Základním způsobem jak realizovat sledování barvy jsou optická čidla. Optických čidel existuje celá řada, některá z nich byla již použita v jiné diplomové práci na naší škole (např. barevné míčky). V případě barevných míčků se jednalo o čidla FX-D1P ([12]), která jsou schopna rozpoznat až 5 barev. Tyto čidla se pro tuto úlohu nehodí: vysoká cena a spotřeba.

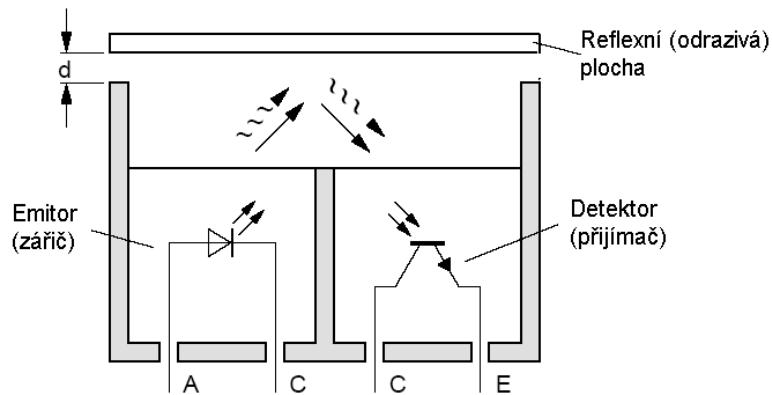
Řešením jsou optická čidla tvořená infračervenou diodou a optickým tranzistorem. Fungují na principu osvětlení báze optotranzistoru odraženým světlem z diody. Tranzistor se "otevře" v závislosti na intenzitě odraženého světla.

Při použití optického čidla CNY70 máme obě součástky pohromadě (diodu i tranzistor) a nemusíme se starat o jejich vzájemnou polohu. Je běžně dostupné a stojí rádově desítky korun. Čidlo i jeho schéma ukazuje obr. 3.15. Funkce tohoto čidla



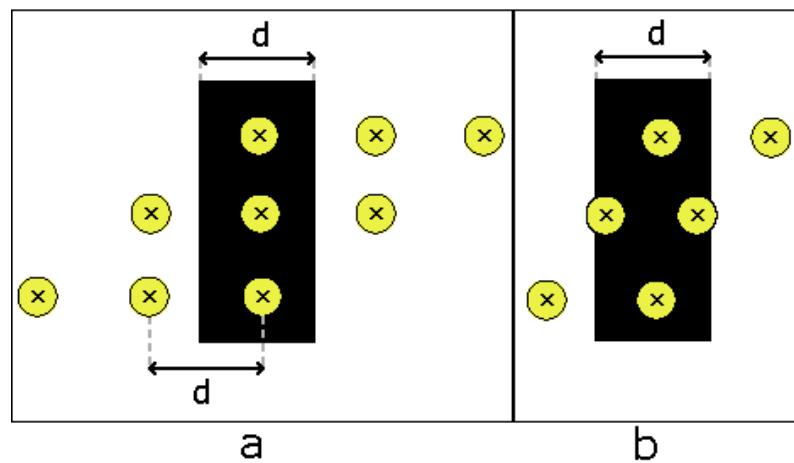
Obrázek 3.15: Optické čidlo CNY70

spočívá v odrazivosti a vzdálenosti d plochy (obr. 3.16). V této práci jsem využil matného černého papíru v kontrastu s lesklým bílým papírem.



Obrázek 3.16: Funkce čidla CNY70

Dráhu jsem zhotovil z matného černého papíru a její okolí z lesklého bílého. Abychom mohli sledovat cestu v rovině, musíme čidla použít více. Minimální počet jsou dvě. Původní návrh počítal s třemi čidly (obr. 3.17a), kde počáteční stav byl: prostřední čidlo na dráze a krajní mimo dráhu. Opuštění vytyčené trasy poznáme ve chvíli, kdy se na čáře objeví jedno z krajních čidel. Šířka čáry musí být rovna vzdálenosti optotranzistorů, čímž zajistíme, že se bude na dráze vždy vyskytovat pouze jedno čidlo. Počet čidel snížíme pokud zapojíme pouze dvě podle obr. 3.17b. Počáteční stav: obě čidla na dráze, vyjetí z dráhy nám signalizuje čidlo, které se objeví mimo dráhu.



Obrázek 3.17: Možnosti orientace v rovině

Deska plošných spojů je navržena pro obě řešení, tedy obsahuje třístupový konektor pro čidla. Výběr počtu závisí jen na zvoleném algoritmu v mikroprocesoru.

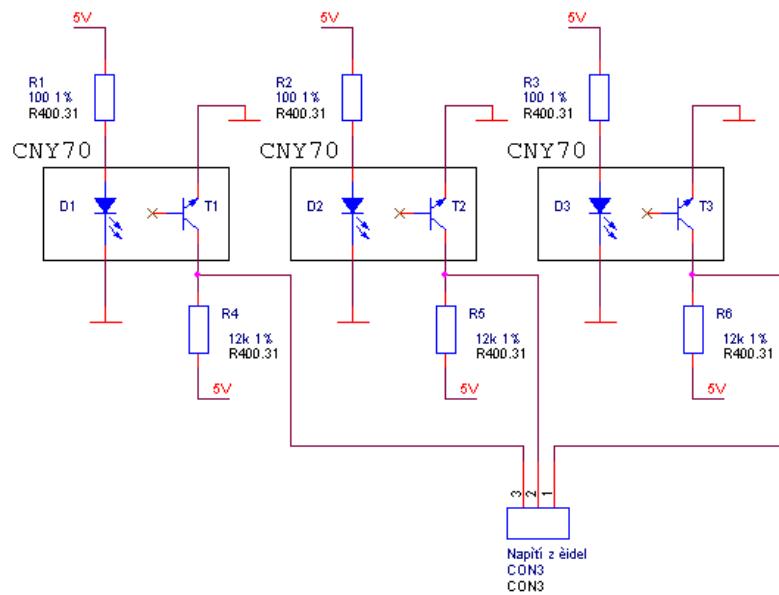
Napětí, které naměříme na kolektoru tranzistoru můžeme zpracovat dvěma způsoby:

1. přivézt na vstup komparátoru, kde je druhý vstup nastaven na zlomové napětí

a výstup komparátoru (logická 1 nebo 0) přivézt na binární vstup mikroprocesoru - **HW řešení**

- přivézt na analogový vstup mikroprocesoru a po A/D převodu programově nastavit zlomovou hodnotu - **SW řešení**

V této úloze byl použit druhý způsob, tedy SW řešení. Schéma zapojení desky s třemi čidly je na obr. 3.18. Zapojení s dvěma čidly je analogické.



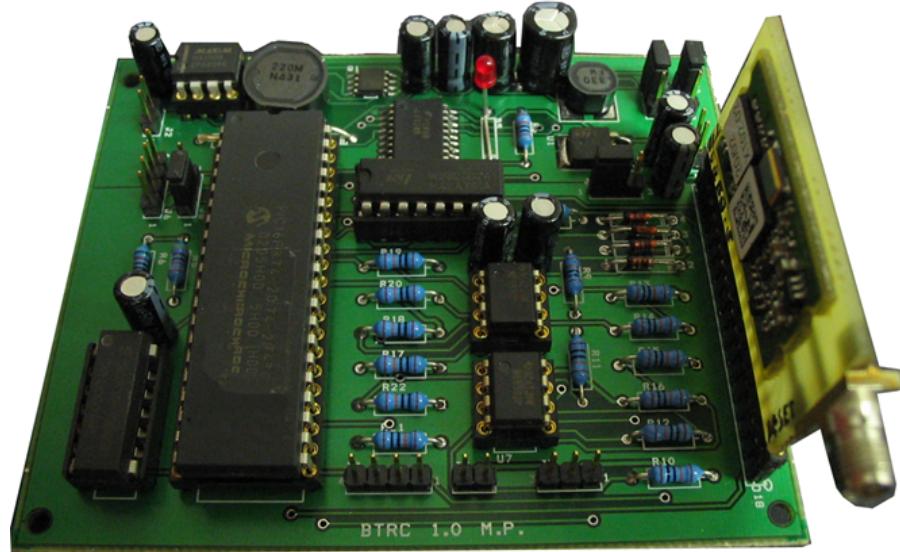
Obrázek 3.18: Schéma zapojení optických čidel

3.1.1.9 Výsledná deska plošných spojů

Návrh zapojení elektroniky a následný návrh desky jsem vypracoval v programovém prostředí OrCad v.9.2. Návrh elektroniky a jednotlivé vrstvy jsou v příloze. Samotná deska je na obr. 3.22.

Na desce je 11 konektorů:

- J1** - konektor pro modul Bluetooth.
- J2** - konektor pro vstup napájení z akumulátorů auta.
- J3** - konektor pro napájení a řízení servomotoru.
- J4** - konektor pro napájení a zároveň řízení stejnosměrného motoru.
- J5** - konektor pro napájení čidel na podvozku auta.
- J6** - konektor pomocných vstupů-výstupů.
- J7** - konektor pro přivedení napětí z čidel a akumulátorů.
- J8** - konektor pro reset mikroprocesoru. Pokud jsou spojeny piny 1 a 2, PIC pracuje normálně, při spojení pinů 2 a 3 dojde k jeho resetu.
- J9 a J10** - konektory pro výběr vstupního napájení desky. Pokud jsou spojeny piny 1 a 2, pak je deska napájena z dc-dc měniče. Pokud jsou rozpojené, očekává se přivedení 5V resp. 0V na pin 2 nebo 3 konektoru J9 resp. J10.



Obrázek 3.19: Výsledná deska

J11 - konektor pro výběr napájení stejnosměrného motoru. Pokud jsou spojeny piny 1 a 2, pak je motor napájen ze vstupu dc-dc měniče (z akumulátorů). Pokud jsou rozpojené, očekává se se přivedení 5V na pin 2 nebo 3.

3.1.2 Model auta pro BTRC

Model auta je opět RC model (1:6) (obr. 3.20). Pohon je řešen stejnosměrným motorem, který je napájen přímé z dobíjecího 12V akumulátoru. Přední náprava je ovládána stejnosměrným motorem, který točí koly doleva a doprava. Krajní poloha je signalizována propojením vodičů v konstrukci přední nápravy.

Realizace modelu

Nenavrhuvala se celá elektronika jako u modelu pro DP, ale pouze tzv. přídavná elektronika, která nahradila integrovaný obvod, který byl součástí původního řešení a který řídil model na základě signálů přijatých z rádiového přijímače.

Vstupy tohoto obvodu byly výstupy rádiového přijímače na desce, výstupy byly signály vpřed, vzad, vlevo a vpravo. Napájen byl 3.3V. Blokové schéma přídavné elektroniky je na obr. 3.21. Schéma zapojení přídavné elektroniky najeznete v příloze.

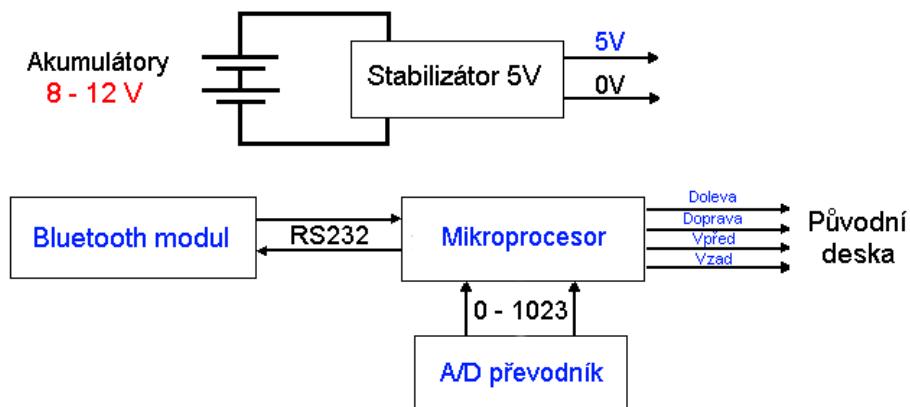
Mikroprocesor

Mikroprocesor jsme použili PIC 16F874 firmy Microchip s externím krystalem 4MHz. Použitím pomalejšího krystalu došlo také ke snížení přenosové rychlosti po RS232 mezi mikroprocesorem a Bluetooth modulem z 57.6kbps na 9.6kbps. Použití

odlišného krystalu o proti DP nebylo nijak úmyslné, pouze byl "k dispozici" a účelům této úlohy vyhovoval.



Obrázek 3.20: Model auta pro BTRC



Obrázek 3.21: Blokové schéma přídavné elektroniky

Úrovně výstupních pinů mikroprocesoru pro řízení modelu jsou přes převodník 74LVX245 převedeny na 0 a 3.3V a poté připojeny na původní výstupy IO. Převodník je napájen z původní desky 3.3V.

Bluetooth modul

Další změnou byl Bluetooth modul. Pro tuto práci jsme použili modul s vyvědeným konektorem pro RS232 (pro PC) a s napájením 5V, opět nabízený firmou Czechlabs. Tento Bluetooth modul (piny RX a TX) musel být k mikroprocesoru (k pinům RX a TX) připojen přes RS232 konektor a přes integrovaný obvod MAX232, který převádí napěťové úrovně sériového kanálu počítače na úroveň 0 a 5V.

Deska s přídavnou elektronikou je napájena 12V přímo z akumulátorů. Přes napěťový stabilizátor 7805 (5V) je napájen Bluetooth modul, mikroprocesor a MAX 3232. Pokud hodnota akumulátoru neklesne pod 7.5V, je zaručena správná funkce

stabilizátoru 7805. V případě, že se hodnota blíží této hranici, mikroprocesor pošle zprávu do mobilního telefonu.

A/D převodník

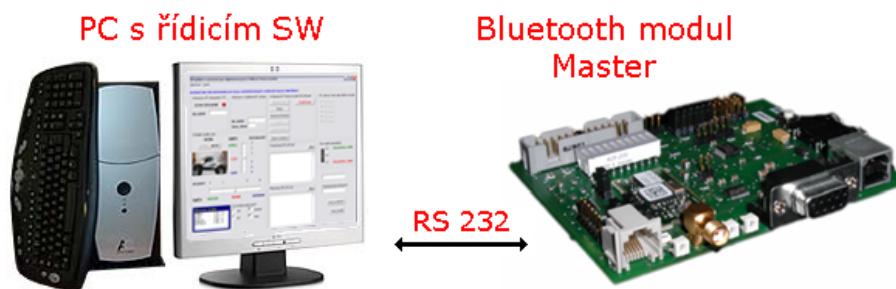
Napětí z akumulátorů je přes odporový dělič (1:3) přivedeno opět přes napěťový sledovač na analogový pin mikroprocesoru. A/D převodníkem je zpracovávána informace o stavu nabití akumulátorů.

3.2 Řídicí systém

V této práci jsou ukázány realizace dvou možných řídicích systému - osobního počítače s připojeným Bluetooth modulem a mobilním telefonem podporujícím technologií Bluetooth.

3.2.1 PC s Bluetooth modulem

Řídicí systém tvořený osobním počítačem s ovládacím softwarem a Bluetooth modulem připojeným přes rozhraní RS232 je na obr. 3.22. Bluetooth modul, který je k počítači připojen, je ve vytvářeném spojení v režimu Master, to znamená tím, kdo spojení vytváří a řídí.



Obrázek 3.22: Schéma řídicího systému

Zatímco v modelu auta je použit samostatný modul Bluetooth, k počítači je připojena deska firmy Siemens, jejíž součástí modul je. Deska je napájena 5V a k počítači připojena pomocí sériového kabelu.

Deska umožňuje i komunikaci přes USB, signalizaci přenosu dat, přenos zvuku, atd., ale v rámci této práce jsem využil pouze její základní vlastnosti.

Pokud bychom měli použít stejný Bluetooth modul jako v modelu auta, bylo by potřeba použít integrovaný obvod MAX3232, který převádí napěťové úrovně Bluetooth (0 a 3.3V) na úrovně sériového rozhraní PC.

3.2.2 Mobilní telefon

Dalším možným způsobem realizace řídicího systému je mobilní telefon (popř. PDA) podporující technologii Bluetooth. Pro tento způsob řízení jsme použili mo-

obilní telefon Nokia 6600 s operačním systémem Symbian (obr. 3.23).



Obrázek 3.23: Mobilní telefon Nokia 6600

Kapitola 4

Softwarové řešení vzdáleného řízení

Softwarové řešení diplomové práce (DP) se skládá ze dvou částečně samostatných částí. První z nich je program pro mikroprocesor PIC, který komunikuje s připojeným Bluetooth modulem (Slave) a podle komunikačního protokolu i se vzdáleným Bluetooth zařízením připojeným k počítači (Master). Zároveň PIC řídí akční členy, vyhodnocuje informaci z čidel a posílá jako odpověď Masterovi (Bluetooth u PC) informaci o stavu napětí akumulátorů. Druhou je program pro počítač, který má za úkol zasláním příkazů do připojeného Bluetooth modelu řídit model auta a zároveň zobrazovat informace o stavu a kvalitě spojení, stavu nabití akumulátorů, atd.

Softwarové řešení práce v rámci projektu BTRC se skládá z programu pro mikroprocesor a programu pro mobilní telefon, který byl napsán pro operační systém Symbian v.6. Tento SW napsal druhý člen týmu Aleš Kučík. Proto se o něm nebudu více zmiňovat - zdrojové soubory naleznete na přiloženém CD-ROMu.

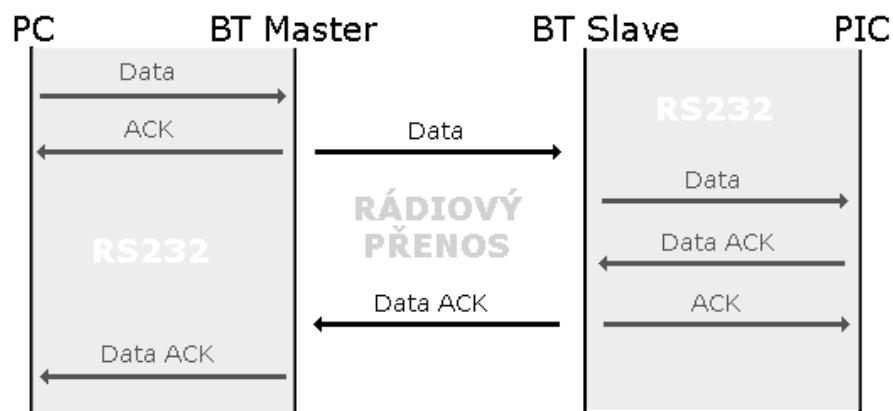
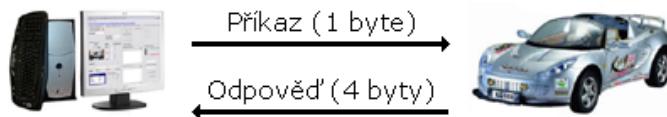
Popis programu je rozdělen do tří částí. V první je vysvětlen komunikační protokol mezi řídicím systémem a modelem auta, který je téměř shodný pro obě práce. V druhé je popsán způsob, jakým se pracuje s mikroprocesorem a struktury programu pro obě práce. Poslední část popisuje řídicí SW pro počítač.

4.1 Komunikační protokol

Projekt DP

Komunikace mezi oběma Bluetooth moduly (u počítače a v modelu auta) probíhá po jejich spojení dle obr. 4.1 a to v pravidelných časových úsecích. Počtem bytů je myšleno počet datových bytů, které jsou přenášeny v rámci daného paketu. V datovém paketu příkazu se posílá byte obsahující informaci o režimu řízení (automatický/ruční) a povely pro řízení periférií. V datovém paketu odpovědi přichází 4 byty nesoucí informaci z A/D převodníku o stavu akumulátoru a napětí z jednotlivých čidel.

Jeden cyklus komunikace je na obr. 4.2. Začíná odesláním datového paketu



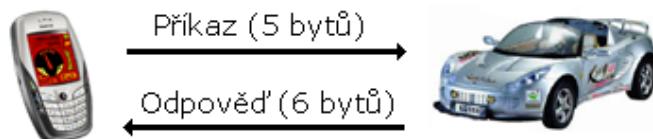
(Data) z PC do Bluetooth modulu (Master) přes RS232, kde je zpracován a odeslán druhému Bluetooth modulu (Slave). Zároveň Master zasílá zpět počítači potvrzení (ACK) o odeslání datového paketu.

Bluetooth Slave po přijetí a zpracování zprávy posílá datový paket mikroprocesoru opět přes rozhraní RS232. Tento datový paket je mikroprocesorem zpracován, a dále je poslán datový paket s odpovědí (Data ACK). Bluetooth Slave opět pošle informaci směrem k Bluetooth Master a zároveň potvrzení o odeslání mikroprocesoru. Cyklus končí ve chvíli, kdy počítač přijme datový paket s odpovědí.

Projekt BTRC

Komunikační protokol projektu BTRC je opět založen na pravidelném zasílání datových paketů mobilním telefonem do modelu. Navýšení počtu datových bytů je způsobeno tím, že SW v mobilním telefonu přistupuje ke službám technologie Bluetooth pomocí vyšší vrstvy L2CAP. Ve skutečnosti se přenáší pouze jeden informační byte (+ 4 byty L2CAP vrstvy) obsahující povely pro periférie. Odpověď jsou 2 byty (+ 4 byty L2CAP vrstvy), z nichž první signalizuje, zda byl zasláný příkaz "srozumitelný" a druhý byte informaci o nabití akumulátoru.

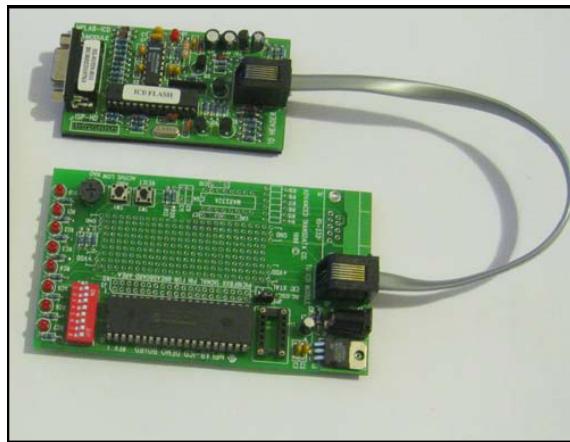
Komunikační cyklus je shodný s projektem DP (obr. 4.2), pouze počítač je nahrazen mobilním telefonem.



Obrázek 4.3: Zjednodušené schéma komunikace mezi telefonem a modelem

4.2 Mikroprocesor PIC

Pro napsání programu pro PIC je určeno vývojové prostředí MPLAB. Programovat lze, bud' v jazyce Assembler, jehož překladač je součástí programu, nebo v jazyce C. Překladač jazyka C pro obvody PIC není součástí programu MPLAB a je potřeba ho zvlášť zakoupit. Na internetu lze některé freewarové překladače jazyka C najít, ale jen pro některé typy procesorů řady PIC. Realizovaný program je napsaný v jazyce Assembler.



Obrázek 4.4: Programátor PIC

K naprogramování procesoru jsem měl k dispozici MPLAB ICD desku (obrázek 4.4). Jedná se vlastně o samotný programátor a vývojovou desku, která umožnuje mimo jiné reset procesoru, vyvolání externího přerušení a signalizaci hodnot výstupů diodami LED a v neposlední řadě komunikaci procesoru přes RS232. Programátor se k počítači připojuje opět přes rozhraní RS232.

4.2.1 Paměťová organizace procesoru

Programová paměť

PIC16F874 má adresový prostor 13 bitů a 13-bitový programový čítač¹ na adresování 8k x 14b (bitů) programové paměti, dohromady tedy 8192 bytů. Programová paměť je rozdělena na stránky (0000h - 07FFh, 0800h - 0FFFh). Vektor reset je

¹Hodnota čítače určující adresový prostor procesoru.

na adresu 0000h a vektor přerušení na adresu 0004h. Znamená to, že pokud dojde k restartování procesoru, popř. k přerušení, „skočí“ program na danou adresu. Pokud dojde k restartování procesoru, provedu opětovnou inicializaci a skok na začátku programu. Pokud dojde k přerušení, volám obsluhu přerušení, kde řeším situaci, která nastala. Po provedení obsluhy přerušení se program vrací na místo, kde se v době vzniku přerušení nacházel.

Datová paměť

Datová paměť je rozdělena do 4 bank, které obsahují, tzv. hlavní účelové registry (General purpose registers) a speciální funkční registry (Special function registers, dále SFR). Výběr banky je dán poměrem bitů SFR registru *Status* (tabulka 4.1).

RP1:RP0	Banka
00	0
01	1
10	2
11	3

Tabulka 4.1: Výběr banky v registru Status

Speciální funkční registry

Speciální funkční registry slouží k nastavení různých vlastností procesoru. Patří mezi ně například povolení různých přerušení, povolení RS232, nastavení přenosové rychlosti, povolení A/D převodníku, povolení a nastavení vnitřního časovače, nastavení a zobrazování různých příznaků, atd. Podrobněji se o těchtoregistrech zmíním při řešení konkrétních problémů v následujících podkapitolách.

4.2.2 Komunikace přes RS232

Přes rozhraní RS232 komunikuje procesor s Bluetooth modulem (Slave). PIC má k dispozici 2 piny, první slouží pro zaslání dat z procesoru - pin RC6 (TX), druhý k přijímání dat do procesoru - pin RC7 (RX).

USART (The Universal Synchronous Asynchronous Receiver Transmitter - univerzální synchronní asynchronní příjem a vysílání) je jedním ze dvou² sériových I/O modulů. USART může být nakonfigurován jako plně-duplexní³ asynchronní (příjemce a vysílač jsou na sobě nezávislí, nejsou synchronizováni, data mohou být poslány kdykoliv) systém, který může komunikovat s perifériemi typu CRT terminál nebo osobní počítač. Druhou možností USART je polo-duplexní⁴ synchronní (příjemce má synchronizovány vnitřní hodiny s vysílačem - data jsou vysílány v pravidelných okamžicích) systém ke komunikaci s A/D nebo D/A integrovanými obvody, sériovými paměti, atd.

²Druhým je MSSP - hlavní synchronní sériový port, používaný pro komunikaci s jinými mikrokontroléry.

³Zařízení může v jednu chvíli jak přijímat, tak i vysílat.

⁴Zařízení buď přijímá nebo vysílá.

V rámci této práce jsem využil režim asynchronní:

- *non-return-zero* (NRZ) formát (1 START bit, 8 datových bitů, 1 STOP bit)
- vyhrazený 8-bitový generátor přenosové rychlosti na čipu (k odvození standardních frekvencí přenosové rychlosti od oscilátoru)
- posílá se a přijímá se nejdříve LSB (nejméně významný bit)
- vysílač a přijímač jsou funkčně nezávislí, ale používají stejný formát dat a stejnou přenosovou rychlosť
- parita (paritní bit - kontrolní) není podporována HW, ale lze implementovat v SW jako 9.bit

Pro práci s USART jsou potřeba registry *PIE1* - povolení periférních přerušení, *PIR1* - příznaky vzniklých přerušení, *RCSTA* - nastavení přijímání a povolení USART, *TXSTA* - nastavení vysílání a přenosové rychlosti a *SPBRG* - registr generátoru přenosové rychlosti.

Nastavení asynchronního vysílače:

1. Nastavení bitu *BRGH* v registru (*SPBRG*) pro vyšší přenosovou rychlosť.
2. Povolení asynchronního sériového portu resetem bitu *SYNC* (*TXSTA*) a nastavením bitu *SPEN* (*RCSTA*).
3. Povolení vysílání nastavením bitu *TXEN*.
4. Nahrání dat (bytu) do *TXREG* registru = start vysílání.
5. Než se odešle další byte, čeká se na uvolnění *TXREG* registru - signalizováno bitem *TXIF* (*PIR1*) v 1.
6. Pro další odesílání se vše opakuje od kroku 4.

Nastavení asynchronního přijímače:

1. Nastavení bitu *BRGH* v registru (*SPBRG*) pro vyšší přenosovou rychlosť.
2. Povolení asynchronního sériového portu resetem bitu *SYNC* (*TXSTA*) a nastavením bitu *SPEN* (*RCSTA*).
3. Povolení přerušení při přijatém bytu nastavením bitu *RCIE* (*PIE1*).
4. Povolení periférních přerušení nastavením bitů *GIE* a *PEIE* registru *INTCON*.
5. Povolení příjmu nastavením bitu *CREN*.
6. Flag (příznakový) bit *RCIF* je v 1, když je přijetí kompletní a generuje se přerušení (pokud je nastaven bit *RCIE*).

7. Přečtení bytu z přijatých dat přečtením *RCREG* registru.
8. Pokud došlo k nějaké chybě, odstraní se resetováním bitu *CREN*.

Přenosová rychlosť se nastavuje registrem **SPBRG** a bitem *BRGH* registru *TXSTA*. Výsledná rychlosť (baud rate) je dána vzorcem:

$$\text{Asynchronní přenosová rychlosť} = \frac{F_{OSC}}{X(Y+1)}$$

kde X je 16 (pri *BRGH* = 1) nebo 64 (*BRGH* = 0), Y je hodnota v registru *SPBRG* a *F_{OSC}* je perioda externého oscilátoru.

4.2.3 Přerušení

Procesor PIC 16F874 nabízí 14 zdrojů přerušení. K nastavení vlastností přerušení a jejich povolení slouží jednotlivé bity registru *INTCON*. Použité bity:

Bit 7 *GIE*: 1 = povolení všech odmaskovaných (povolených) přerušení, 0 = zákaz všech přerušení

Bit 6 *PEIE*: 1 = povolení všech odmaskovaných přerušení od periférií, 0 = zákaz všech přerušení od periférií

Bit 5 *TOIE*: 1 = povolení přerušení od časovače TMR0, 0 = zákaz přerušení od časovače TMR0

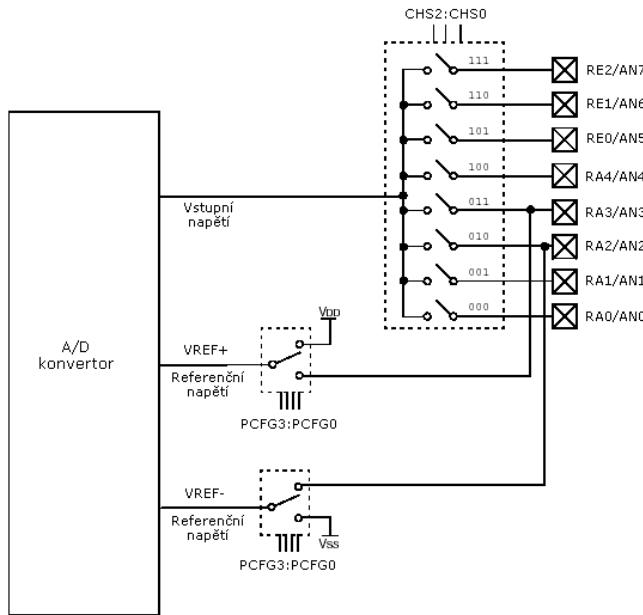
Bit 2 *TOIF*: 1 = příznak, že registr TRM0 přetekl (jeho hodnota překročila 255), 0 = registr nepřetekl

Pokud dojde k přerušení, nastaví se odpovídající příznak do 1. Podle tohoto lze rozpoznat, která událost přerušení vyvolala. Po obslužení přerušovací rutiny (část programu, která je volána při vzniku přerušení) se musí příznakové bity resetovat softwarově, jinak by docházelo k opakovanému přerušení ihned po opuštění této rutiny. Dokud nedojde k obslužení přerušovací rutiny je bit GIE resetován (automaticky mikroprocesorem). Teprve po opuštění rutiny je tento bit opět nastaven (HW) a je povoleno další přerušení.

4.2.4 A/D převodník

A/D převodník je součástí procesoru PIC. K dispozici je 8 analogových pinů: RA0-5, RE0-2 (4.5). K nastavení A/D převodníku slouží registry *ADCON0* a *ADCON1*. K nastavení a spuštění A/D převodu je potřeba:

1. Nastavení A/D modulu:
 - určit hodiny převodu (registrování *ADCON0*)
 - určit jaký pin (kanál) se bude měřit (*ADCON0*)
 - povolit A/D převod celkově (*ADCON0*)
 - nastavit referenci (\pm)
2. Nastavení přerušení:



Obrázek 4.5: Blokové schéma A/D převodníku

- reset bitu *ADIF* (*PIR1*)
 - nastavení bitu *ADIE* (*PIE1*)
 - nastavené bitu *PEIE* (*INTCON*)
 - nastavení bitu *GIE* (*INTCON*)
3. Čekání požadovanou dobu před zahájením A/D převodu (cca $20\mu\text{s}$)
 4. Spuštění převodu nastavením bitu *GO* (*ADCON0*)
 5. Čekání na dokončení převodu (přerušení - ADIF)
 6. Přečtení výsledku z registrů *ADRESH* a *ADRESL*
 7. Pro další A/D převod se opakuje celý cyklus:
 - od 1 - pokud chceme změnit kanál (vstupní pin, na kterém se měří napětí)
 - od 2 - pokud pin zůstává stejný

Hodiny převodu (A/D conversion clock) pracují s danou periodou, která je odvozena od vnitřních hodin procesoru a je jejich násobkem. Perioda vnitřních hodin je $T_{OSC} = \frac{1}{4}$ periody krystalu $= 0.2 \mu\text{s}$. Nejmenší možný čas potřebný k převodu jednoho bitu je $T_{AD} = 1.6 \mu\text{s}$ a jelikož k dispozici jsou násobky 2 T_{OSC} , 8 T_{OSC} a 32 T_{OSC} , použil jsem násobek poslední.

Určení vstupního pinu se provádí třemi bity *CHS2:0* registru *ADCON0*. Nastavení reference je dáno opět třemi bity. Určuje se jimi, zda budou jako analogové vstupy použity všechny analogové piny a zda bude referenční napětí dáno napájecím

(+5V a 0V) nebo zda bude dané hodnotami napětí na pinech RA3 (horní referenční napětí) a RA2 (spodní referenční napětí).

Dokončení A/D převodu je signalizováno přerušením (pokud je povoleno) a resetováním bitu *GO* (vždy). Výsledná hodnota (10-ti bitové číslo) je uložena ve dvou registrech. V prvním z nich je uloženo nejvyšších 8 bitů a ve druhém zbylé 2 bity doplněné nulami.

10-ti bitový A/D převodník má nejmenší díl (rozdíl napětí, při kterém rozpozná změnu) daný podílem rozdílu referenčních napětí a počtem bitů. V tomto případě je to $\frac{5-0}{2^{10}} = 0.00488 V = 4.88 mV$.

4.2.5 Časovač a PWM

Následuje řešení PWM pro motory. Základem PWM je vnitřní časovač procesoru - registr, který při přetečení⁵ vyvolá přerušení. Dobu, za kterou k přerušení od časovače dojde jsem zvolil 0.1 ms. Tento minimální čas je potřeba pro řízení servomotoru. Hodnota časovače je dána registrum *TMR0*, to znamená 8 bitů a maximální decimální hodnota je tedy 255. Frekvence přerušení je dána:

1. hodnotou v registru *TMR0*
2. frekvencí externího krystalu procesoru
3. předděličkou

Frekvence externího krystalového oscilátoru je 20MHz. Tento kmitočet je pro vnitřní časování procesoru vydělen čtyřmi. Dále jde nastavit, tzv. předdělička, díky které můžeme frekvenci ještě snížit. Její hodnota se nastavuje bity *PS2:PS0* registru *OPTION REG*. Možné hodnoty předděličky jsou v tabulce 4.2.

Hodnoty bitů PS2:PS0	Hodnota předděličky
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

Tabulka 4.2: Nastavení předděličky časovače procesoru

Abych dosáhl výsledné hodnoty času 0.1ms je předdělička nastavena na poměr 1:4 a počáteční hodnota v registru *TMR0* na 131, takže se mění jen od 131 do 256, to znamená 125 inkrementací. Tato hodnota se musí při každém přerušení do registru *TMR0* opět ukládat. Výsledkem je frekvence:

⁵Hodnota, která má být uložena je větší než paměť přidělená časovači.

$$\frac{2 \cdot 10^7}{4.4.125} = 1.10^4 \frac{1}{s}$$

a perioda přerušení:

$$\frac{1}{1.10^4} = 0.1 \text{ ms}$$

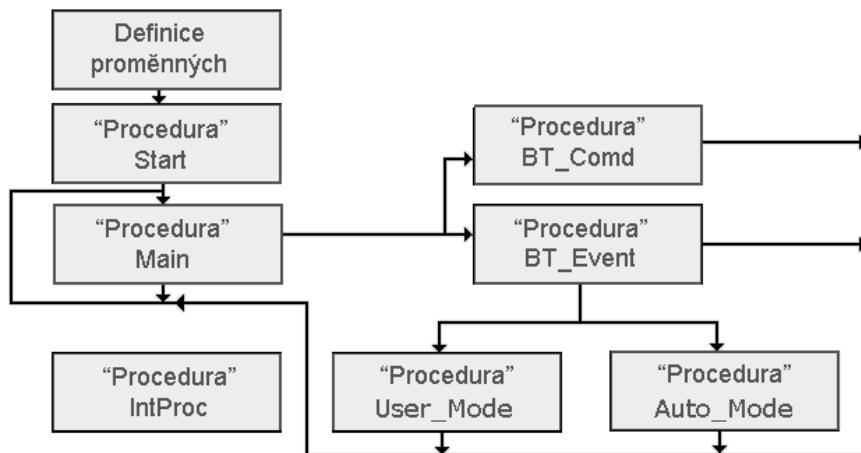
Generování PWM

Při každém přerušení od časovače je zvýšena o jedničku proměnná (registrovým) *Count TMR0* a v každém cyklu hlavního programu je porovnávána s dvěma hodnotami. První je požadovaná délka impulsu (T_1), druhou je požadovaná perioda výsledného PWM signálu (T_2). Pokud je *Count TMR0* mezi 0 a T_1 , pak je výsledkem hodnota 1. Pokud je mezi T_1 a T_2 , pak je hodnota PWM 0. Hodnota T_2 je pevně daná, jak pro stejnosměrný motor, tak i pro servo motor a je $T_2 = 200$. Tím je dáná perioda PWM 20ms (frekvence 50Hz). Hodnota T_1 je, buď získána z přijatého paketu z počítače (v režimu ručního řízení) nebo z části programu, která zpracovává informaci z čidel (v automatickém režimu).

Další možností jak řešit PWM v mikroprocesoru je použití vnitřní PWM jednotky, ale frekvence, kterou bych tím dostal, je příliš veliká.

4.2.6 Struktura programu pro DP

Blokové schéma programu obsahující hlavní části je na obrázku 4.6. Program se skládá z osmi hlavních procedur.



Obrázek 4.6: Blokové schéma programu pro PIC

Program pro mikroprocesor začíná nastavením formátu souboru výpisu překladu zdrojového kódu a vložením souboru *.inc s definicí hardware procesoru. Dále se definují odskoky do částí programu, které řeší restart (*Start*) a přerušení (*IntProc*) procesoru. Po těchto nutných krocích následuje samotný program sestávající se z těchto částí:

Definice proměnných

Definicí proměnných je myšleno přidělení symbolických jmen místům (bytům) v paměti mikroprocesoru. K tomuto je možno využít 96 bytů v bance 0, 80 bytů v bance 1 a shodně 96 bytů v bankách 2 a 3.

Procedura Start

V této části jsou proměnným (registrům) přiřazeny hodnoty, např. pro počáteční natočení servomotoru (neutrální poloha), pro počáteční stav stejnosměrného motoru (zastaven), atd. Dále se nastavují speciální funkční registry, např.:

- **TXSTA, RXSTA** - povolení RS232 (příjem, vysílání), povolení přerušení při příjmu dat
- **SPBRG** - nastavení přenosové rychlosti na 57.6kbps
- **PIE1, INTCON** - povolení přerušení při dokončení A/D převodu
- **ADCON0, ADCON1** - nastavení a povolení A/D převodu
- **TRISA, B, C** - povolení a nastavení analogových vstupů
- **PORTA, B, C** - povolení a nastavení digitálních vstupů a výstupů

Toto částí program začíná - volá se vždy po restartu procesoru a při výskytu chyby v komunikaci (více v části řešení chybových stavů).

Procedura Main

Tato část následuje ihned po skončení procedury *Start*. Je to hlavní smyčka programu, která volá procedury *BT Comd* a *BT Event*, ze kterých jsou volány již všechny zbývající procedury kromě *IntProc*, která je volána pouze při vzniku přerušení.

BT Comd a *BT Event* jsou volány v závislosti na stavu bitu *Transmit* pomocného registru *Proces*. Dále se v této části programu volá procedura *SS Routine*, která "řídí" výstupní piny mikroprocesoru pro řízení stejnosměrného motoru.

Procedury BT Comd a BT Event

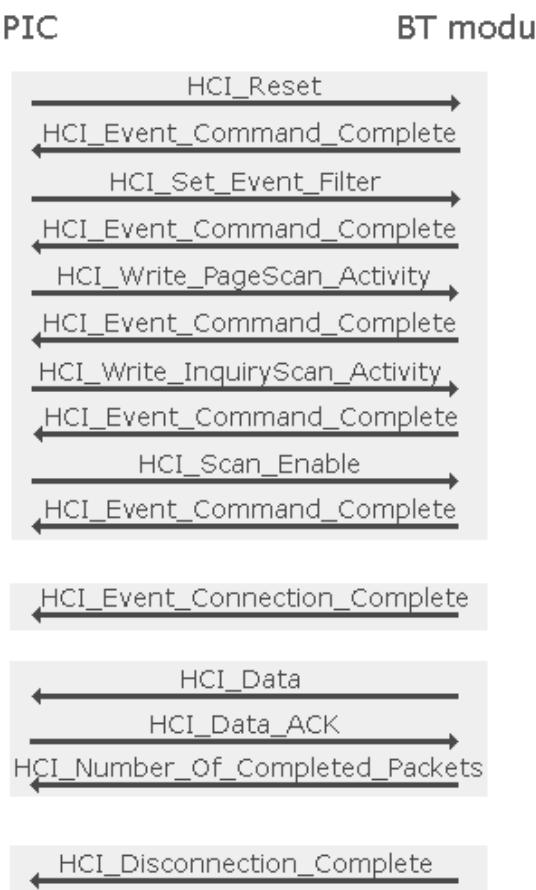
BT Comd slouží ke komunikaci směrem k Bluetooth modulu přes rohnaní RS232 - posílá pakety (HCI příkazy a HCI data). *BT Event* slouží ke komunikaci směrem od Bluetooth modulu - zpracovává přijaté pakety (HCI události a HCI data). Každému HCI příkazu odpovídá specifická HCI událost. Program po odeslání příkazu nebo dat čeká na specifický paket dané délky.

Prvním příkazem je *Reset*, následuje *Set Event Filter*, *Write Page Scan Acitivity*, *Write Inquiry Scan Activity* a *Scan Enable*, které jsou potřeba pro navázání spojení s Bluetooth modulem a jeho nastavení jako Slave. Na každý z těchto příkazů Bluetooth modul odpoví událostí *Command Complete*. Po odeslání těchto 5 příkazů se

program nachází v tzv. pohotovostním stavu, kdy čeká na událost *Connection Complete*, která signalizuje vytvoření spojení se vzdáleným Bluetooth modulem (u PC).

Ve stavu „Spojení“ se periodicky opakuje čekání mikroprocesoru na datový paket, odeslání datového paketu s odpovědí (Data ACK) a nakonec přijetí události *Number Of Completed Packets*, která potvrzuje úspěšné odeslání datového paketu. Toto se opakuje, dokud trvá spojení mezi oběma Bluetooth moduly, tzn. dokud nedojde k chybě spojení nebo Bluetooth Master nezašle událost *Disconnection Complete* znamenající ukončení spojení.

Celkové schéma komunikace je na obr. 4.7. Presný popis paketů vyměněných v rámci komunikace mikroprocesoru s Bluetooth modulem je uveden v příloze.



Obrázek 4.7: Schéma komunikace mezi procesorem a Bluetooth modulem

Pakety se posílají byte po bytu nahráváním do registru *TXREG* a čtou podobým způsobem, tedy opět byte po bytu, tentokrát z registru *RXREG*.

Procedura User Mode

Tato část programu je volána z procedury *BT Event* v případě, že přijatý byte v datovém paketu obsahuje informaci o tom, že se řízení modelu nachází v uživatelském režimu (nastavuje se v aplikaci v PC). Pokud je zavolána, zpracuje se přijatý byte - hodnoty pro PWM servomotoru a stejnosměrného motoru jsou zapsány do příslušných pomocných registrů.

Procedura Auto Mode

Tato procedura se volá v opačném případě než *User mode* a je opět volána z procedury *BT Event*. V tomto případě je přijatý byte zapomenut a hodnoty PWM pro oba motory jsou nastaveny na základě informací z optických čidel.

Samotné řízení motorů je řešeno v části *IntProc*.

Procedura IntProc

Při vzniku přerušení se volá blok *IntProc*, který obsahuje řešení konkrétní příčiny přerušení. Přerušení může být způsobeno časovačem (každých 0.1ms), A/D převodníkem nebo přijatým bytem.

V případě přerušení od časovače se generuje PWM pro oba motory (z důvodu přesnosti, obzvláště pro servomotor) a také se každých 25ms spouší A/D převod.

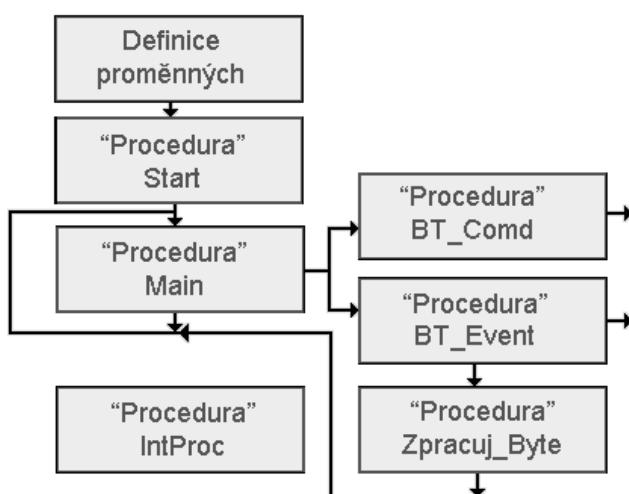
Přerušení vzniklé A/D převodem znamená, že je převod hotov a je potřeba výsledek uložit a zpracovat, než se spustí další převod.

Při přerušení od sériového rozhraní (příjem) se načte přijatý byte do jednoho z 25 připravených registrů. Každý paket je ukládán samostatně pro lepší kontrolu případné chyby. V případě správnosti celého paketu dojde k vynulování všech 25 registrů a další přijatý byte je již ukládán do 1. registru.

Během výskytu programu v části *IntProc* nedochází k jinému přerušení a program, tzv. "čeká" v hlavní smyčce na obsloužení této procedury.

4.2.7 Struktura programu pro BTRC

Struktura programu pro řízení mobilním telefonem (obr. 4.8) je velmi podobná té, kdy je řízena počítačem. Samozřejmě chybí procedura *Auto mode*, protože v této práci je pouze "ruční" režim.



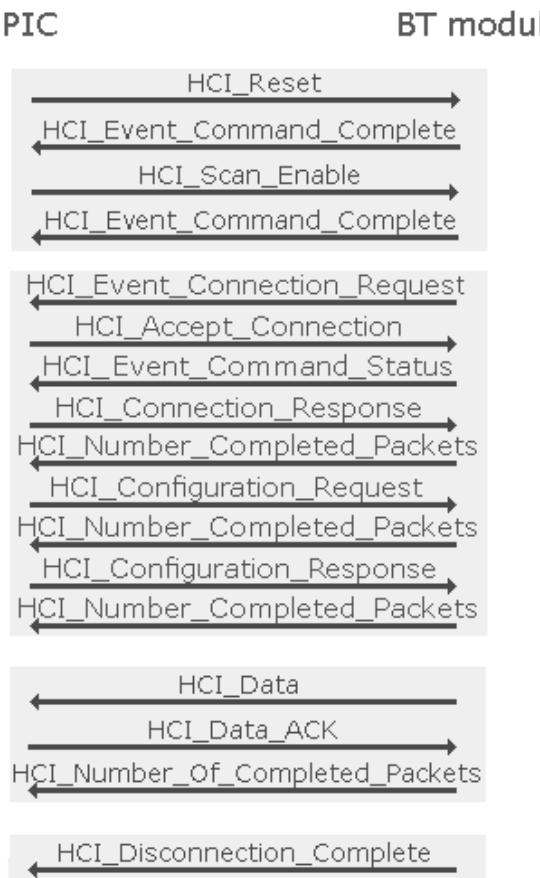
Obrázek 4.8: Struktura programu pro BTRC

Zásadní rozdíl je v komunikaci s Bluetooth modulem, kdy komunikace mikro-

procesor - Bluetooth Slave zůstává na úrovni HCI vrstvy, ale komunikace Bluetooth Master - mobilní telefon probíhá na vrstvě L2CAP a tím je ovlivněna i komunikace HCI.

Zatímco při komunikaci s počítačem jsme Bluetooth Slave nastavovali pěti HCI příkazy, v tomto případě ho nastavujeme dvěma - *HCI Reset* a *HCI Set Event Filter*.

Další výrazná změna je při vytváření spojení. Zatímco u DP přišla pouze událost *Connection Complete*, v tomto případě se musí do vytvoření spojení zapojit i jednotka Bluetooth Slave (obr. 4.9). Výměna dat zůstává stejná, jen složení a délka datového paketu je jiná.



Obrázek 4.9: Schéma komunikace mezi procesorem a Bluetooth modulem

Přesný popis paketů vyměněných v rámci komunikace mikroprocesoru s Bluetooth modulem je uveden opět v příloze.

4.2.8 Ošetření chybových stavů

Ošetřením chybových stavů je myšleno ošetření chyb, které mohou vzniknout mezi procesorem a Bluetooth modulem při komunikaci po RS232. Chyby mohou být rozděleny do 2 skupin:

1. byl přijat správný počet bytů, ale nesprávného obsahu

2. do daného časového okamžiku nebyl přijat správný počet bytů

První možností je myšlen případ, kdy příjde událost, která nese informační byte *Status*, který pokud je různý od *0x00* znamená, že došlo k chybě. Možnost, že by dorazil zcela jiný paket než se očekává je téměř nemožné, Bluetooth jednotka pošle paket pouze jako odpověď na nějaký podnět (HCI příkaz nebo HCI data), který dostane buď od mikroprocesoru nebo od vzdálené Bluetooth jednotky. Protože je komunikační protokol pevně definován, stejně jako příkazy, které zasílá počítač, nemůže dojít k příjetí "nečekaného" paketu.

Druhou možností je, že vypršel definovaný čas na přijetí paketu. Tento čas je dvojího typu:

- mikroprocesor odešle příkaz nebo data a čeká na událost od Bluetooth Slave - tento čas může být velmi krátký
- mikroprocesor čeká na data od Bluetooth Slave (Master) - tento čas je dán periodicitou zasílání datových paketů z počítače

V prvém případě vzniklá chyba znamená chybu při komunikaci s jednotkou Bluetooth Slave - je potřeba zaslat příkaz *HCI Reset*, tedy zavolat znovu proceduru *Start*.

V druhém případě vzniklá chyba znamená chybu v komunikaci se vzdáleným zařízením. Důsledek je stejný - opět se volá procedura *Start*.

4.3 Počítač

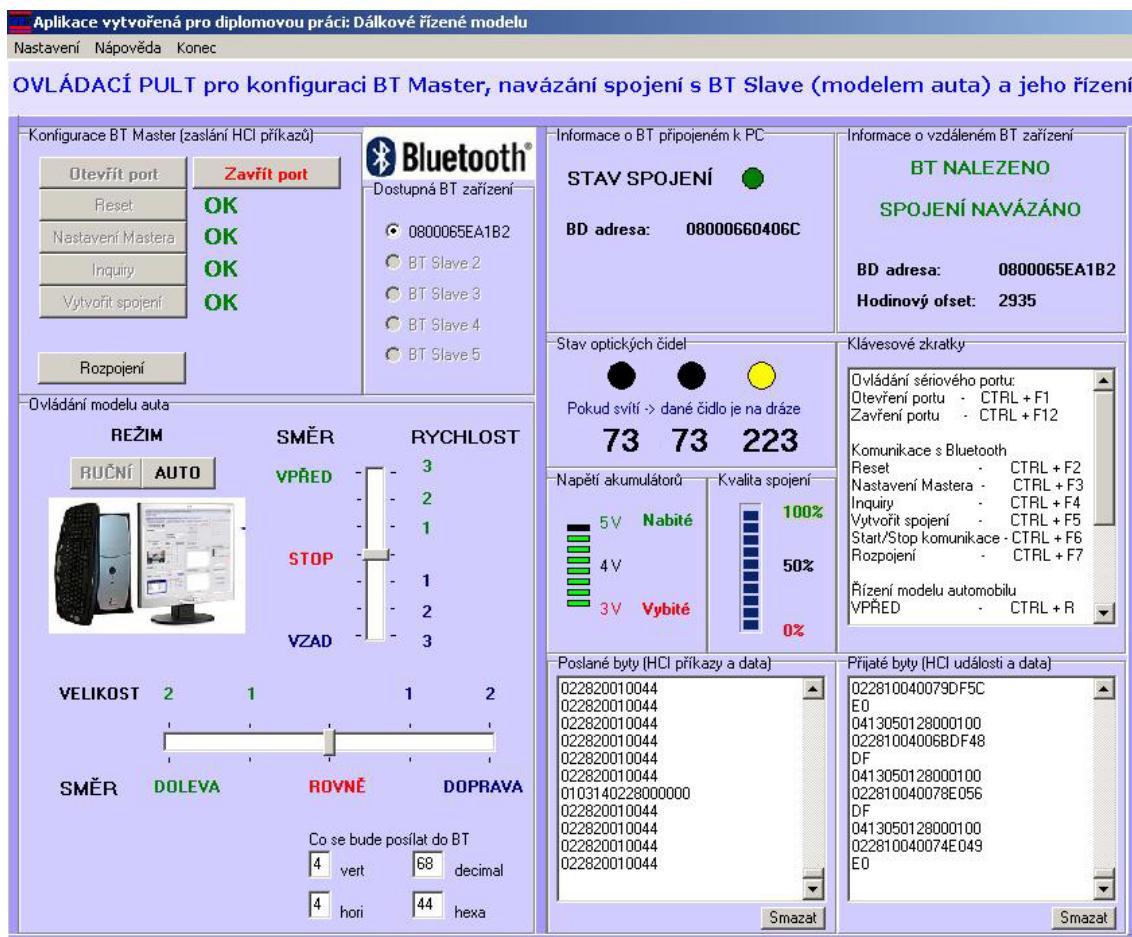
Řídicí software pro počítač se týká projektu DP. Jeho cílem je ovládání modelu auta s možností přepnutí do režimu, kdy mikroprocesor přijímá řídicí pakety z PC, ale rozhoduje se o pohybu na základě informace přijaté z optických čidel. Tyto dva způsoby řízení můžeme rozdělit na ruční a automatický režim řízení.

Aplikace je napsaná v Borland C++ Builderu verze 5 a je jí možné spustit v systému Microsoft Windows 95 a vyšším. Po spuštění souboru aplikace.exe se objeví hlavní okno programu (obr. 4.10).

Aplikace komunikuje s Bluetooth modulem pomocí sériového portu počítače. Způsobů přístupů na sériový port je více, záleží na zvoleném programovacím prostředí a operačním systému. Pro nezávislost na verzi operačního systému Windows jsem použil řešení přístupu na hardwarový blok pomocí aplikačního rozhraní daného operačního systému.

Operační systém je multitaskingový - jednotlivé procesy mohou pracovat paralelně. Ve WIN32 je na sériový port po jeho otevření hleděno jako na soubor, který je otevřen pro čtení i pro zápis. O komunikaci se sériovým portem se tedy stará jeden konkrétní proces (samostatně běžící vlákno aplikace).

Více o způsobech přístupu k portům najdeme v [7], [8] nebo [9].



Obrázek 4.10: Hlavní okno aplikace

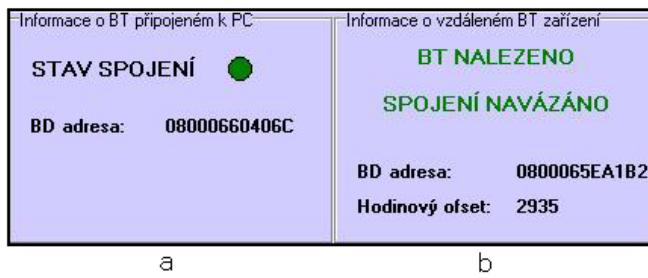
4.3.1 Práce s programem

Před samotným používáním aplikace, tzn. před započetím práce s připojeným Bluetooth modulem, je potřeba zkонтrolovat, zda jsou nastavené hodnoty přenosové rychlosti a sériového portu správné. Automaticky je nastaven port COM1 a 57600 kbps. Tyto parametry lze změnit v menu, stisknutím záložky *Nastavení - Sériová linka - Port nebo Baud rate* a výběrem odpovídající hodnoty. V menu se lze pohybovat i klávesovými zkratkami.

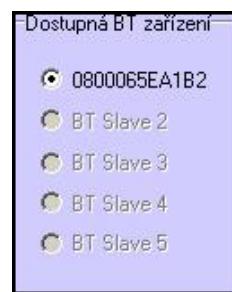
Další nastavitelnou položkou je perioda vysílání, která určuje jak často PC posílá po připojení data mikroprocesoru. Základní nastavení je 100ms.

Samotná práce s Bluetooth modulem začíná otevřením zvoleného sériového portu. K dispozici jsou tlačítka *Reset* a *Nastavení Mastera*. Stiskem tlačítka *Reset* dojde k zaslání HCI příkazu *Reset* do Bluetooth modulu.

Pokud není chyba v komunikaci dojde po kliknutí na tlačítko *Nastavení Mastera* k povolení tlačítka *Inquiry* a zároveň k zobrazení BD adresy připojeného Bluetooth modulu (obr. 4.11a). Stisk tlačítka *Inquiry* způsobí vyhledávaní okolních aktivních zařízení Bluetooth. Všechna dostupná zařízení se objeví v okně *Dostupná Bluetooth zařízení* (obr. 4.12). Program je přizpůsoben 5-ti novým zařízením a vždy ukáže BD



Obrázek 4.11: Informace o připojeném (a) a vzdáleném (b) Bluetooth modulu



Obrázek 4.12: Informace o nalezených Slave zařízeních

adresy nalezených zařízení pro jejich snazší rozpoznání. Spojení s daným Bluetooth zařízením se vytvoří jeho vybráním z *Checkboxu* a stiskem tlačítka *Vytvořit spojení*.

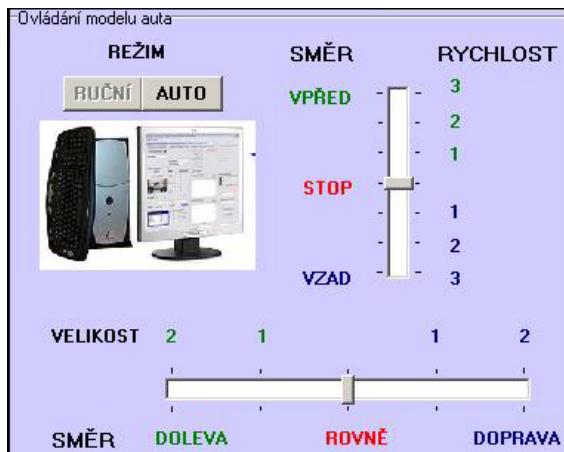
Pokud všechny příkazy spouštené stisky tlačítek proběhnou úspěšně, objeví se zelený nápis *OK*, v opačném případě červený nápis *KO*.

4.3.2 Komunikace s modelem auta

Po vytvoření spojení mezi oběma Bluetooth moduly se v okně *Informace o vzdáleném Bluetooth zařízení* objeví BD adresa vzdáleného Bluetooth a jeho hodinový offset (obr. 4.11b) a program začne zasílat v pravidelných časových intervalech datový paket, obsahující aktuální informaci o stavu řízení, tedy:

- o režimu řízení (ruční / auto)
- o pokynech jízdy vpřed a vzad (3 různé rychlosti oběma směry + stání)
- o pokynech zatáčení (2 směry natočení přední nápravy + rovně)

Samotné ovládání modelu auta a přepínání mezi režimy se nachází v okně *Ovládání modelu auta* (obr. 4.13). Pokud je model v automatickém režimu, je zobrazen obrázek auta s dráhou. V opačném případě počítač, který signalizuje režim ručního řízení.



Obrázek 4.13: Okno ovládání modelu auta

Ovládat lze panel řízení, buď pomocí myši nebo klávesových zkratek. Mikroprocesor v odpovědi na tento datový paket reaguje zasláním datového paketu nesoucího informaci o stavu nabité akumulátoru a stavu trojice čidel (okna *Napětí akumulátorů* a *Stav optických čidel*). Mimo to se program dotazuje připojeného Bluetooth modulu na kvalitu vytvořeného spojení a přijatou informaci ukazuje v okně *Kvalita spojení* (4.14).

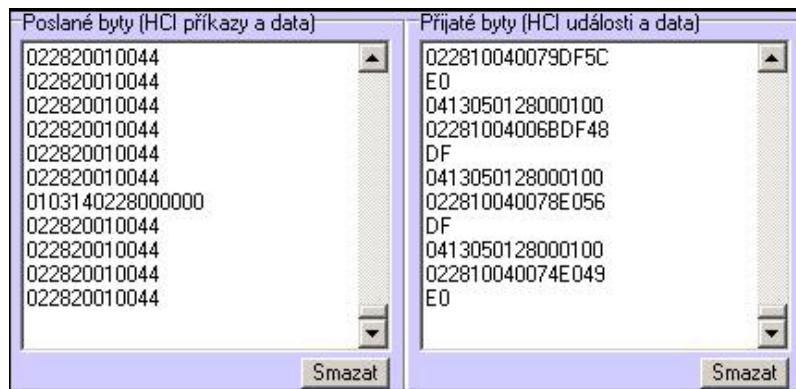


Obrázek 4.14: Informace o napětí akumulátorů, stavu optických čidel a kvalitě spojení

Ukončit vytvořené spojení lze dvěma způsoby:

- stiskem tlačítka *Rozpojení* - dojde k zaslání příkazu Disconnect
- zavřením aplikace - dojde k zaslání HCI příkazu Disconnect, k uzavření portu a zavření aplikace

Veškerá komunikace po sériové lince je průběžně zobrazována v oknech *Poslané byty* a *Přijaté byty* (obr. 4.15).



Obrázek 4.15: Informace o odeslaných a přijatých paketech

4.3.3 Ošetření chybových stavů

Chybové stavy můžeme rozdělit na 2 skupiny:

1. Chyby během komunikace počítače s připojeným Bluetooth modulem během navazování spojení.
2. Chyby během komunikace Bluetooth modulů během vytvořeného spojení.

První chybu poznáme po stisku tlačítka, kdy se objeví červený nápis *KO* - znamená to, že je chyba v komunikaci s Bluetooth modulem a je potřeba zaslat *Reset*. Pokud ani toto nepomůže, je jediným řešením HW Reset, případně vypnutí a opětovné zapnutí Bluetooth modulu. Tato situace může nastat v případě, že Bluetooth modul po RS232 přijme paket, kterému nerozumí, tedy např. pokud nastavíme odlišnou přenosovou rychlosť oproti nastavené v Bluetooth modulu.

Druhým chybovým stavem je přerušení spojení mezi Bluetooth moduly. Rozpoznáme ji ve chvíli, kdy aplikace stále posílá datový paket, ale nepřichází odpověď. Způsobit to může chyba v komunikaci s připojeným Bluetooth modulem, popř. poodjetí modelu auta z dosažitelné vzdálenosti - dojde k přerušení spojení. Program v mikroprocesoru v takovém případě po určité době zinicializuje Bluetooth Slave a "připraví" ho k dalšímu spojení. Aplikace nahlásí chybu spojení a je potřeba znova zopakovat postup navázání spojení s Bluetooth modulem u počítače.

Kapitola 5

Závěr

Tato diplomová práce zahrnuje práci na dvou projektech současně. Hlavním projektem byla samotná diplomová práce na katedře řídicí techniky, vedlejším byl projekt pro Výzkumné a vývojové centrum v Praze.

Rozdíl mezi oběma projekty byl v použití řídicího systému. V prvém případě je použit počítač připojený k Bluetooth modulu, v druhém mobilní telefon Nokia 6600. Hlavním přínosem druhého projektu je demonstrace způsobu navázání spojení mikroprocesor - Bluetooth modul - mobilní telefon a jejich vzájemné komunikace.

Práci na obou projektech lze rozdělit na hardwarovou a softwarovou část. V hardwarové části byla navržena a realizována elektronika v obou modelech automobilu včetně řídicího systému. V softwarové části byl vytvořen program pro mikroprocesor pro oba projekty a byla naprogramována řídicí aplikace pro počítač.

Hardwarová část se skládala z elektrické a mechanické části. V elektrické části jsem se nejdříve zabýval komunikací Bluetooth modulu s počítačem přes RS232 a vzájemnou komunikací mikroprocesoru s Bluetooth modulem. Dále jsem navrhl zapojení elektroniky potřebné pro vzájemnou komunikaci a ovládání periférií mikroprocesorem (motory, čidla, ...).

Do mechanické části patří úprava přední nápravy pro proporcionální řízení, umístění servomotoru, nahraď desky plošných spojů za nově navrženou a přichycení desky s čidly k podvozku automobilu.

Softwarová část se skládá ze dvou programů. První z nich, program pro mikroprocesor, je napsán ve dvou verzích. Prvá pro komunikaci s počítačem, druhá pro komunikaci s mobilním telefonem. Druhým programem je aplikace pro počítač zajíšťující komunikaci s připojeným i vzdáleným Bluetooth modulem a umožňující přenos uživatelských požadavků.

Rychlosť přenosu informace mezi řídicím systémem a modelem automobilu je dostatečná, celková doba jednoho komunikačního cyklu je kratší než 100ms a lze tedy tento způsob řízení považovat za **řízení v reálném čase**.

Literatura

- [1] VACEK, V.: *Učebnice programování PIC*. Vydavatelství BEN - technická literatura, Praha 2002.
- [2] DOC.ING.ONDŘEJ VYSOKÝ, CSc.: *Elektronické systémy II*. Vydavatelství ČVUT, Praha 1999.
- [3] HAASZ, V., SEDLÁČEK, M.: *Elektrická měření*. Vydavatelství ČVUT, Praha 1998.
- [4] FOIT, J., HUDEC, L.: *Součástky moderní elektroniky*. Vydavatelství ČVUT, Praha 1998.
- [5] ZÁHLAVA, V.: *OrCad 10*. Vydavatelství Grada Publishing, a.s., Praha 2004.
- [6] JELEN, J.: *Fyzika II*. Vydavatelství ČVUT, Praha 1998.
- [7] VACEK, VÁCLAV: *Sériová komunikace ve WIN32*. Vydavatelství BEN, Praha 2003.
- [8] MATOUŠEK, DAVID: *C++ Builder vývojové prostředí 3.díl*. Vydavatelství BEN, Praha 2003.
- [9] MARKUS, LINKE: *Visual Basic 5*. Vydavatelství Grada Publishing, Praha 1998.
- [10] THE OFFICAL BLUETOOTH MEMBERSHIP SITE: <http://www.bluetooth.org>
- [11] MICROCHIP TECHNOLOGY INC.: <http://www.microchip.com>
- [12] LABLINK: <http://dce.felk.cvut.cz/lablink/main.php>
- [13] ZIGBEE ALLIANCE: <http://www.zigbee.org>
- [14] WI-FI ALLIANCE: <http://www.wi-fi.org>
- [15] ČESKOSLOVENSKÉ SDRUŽENÍ UŽIVATELŮ TEXU: <http://www.cstug.cz/>

Dodatek A

Struktura přiloženého CD ROM

Přiložený CD ROM v jednotlivých složkách obsahuje:



Obrázek A.1: Schéma zapojení elektroniky pro DP

Diplomová práce – Tato diplomová práce ve formátu PDF.

Manuály – Dokumentace k používanému hardwaru a softwaru ve formátu PDF.

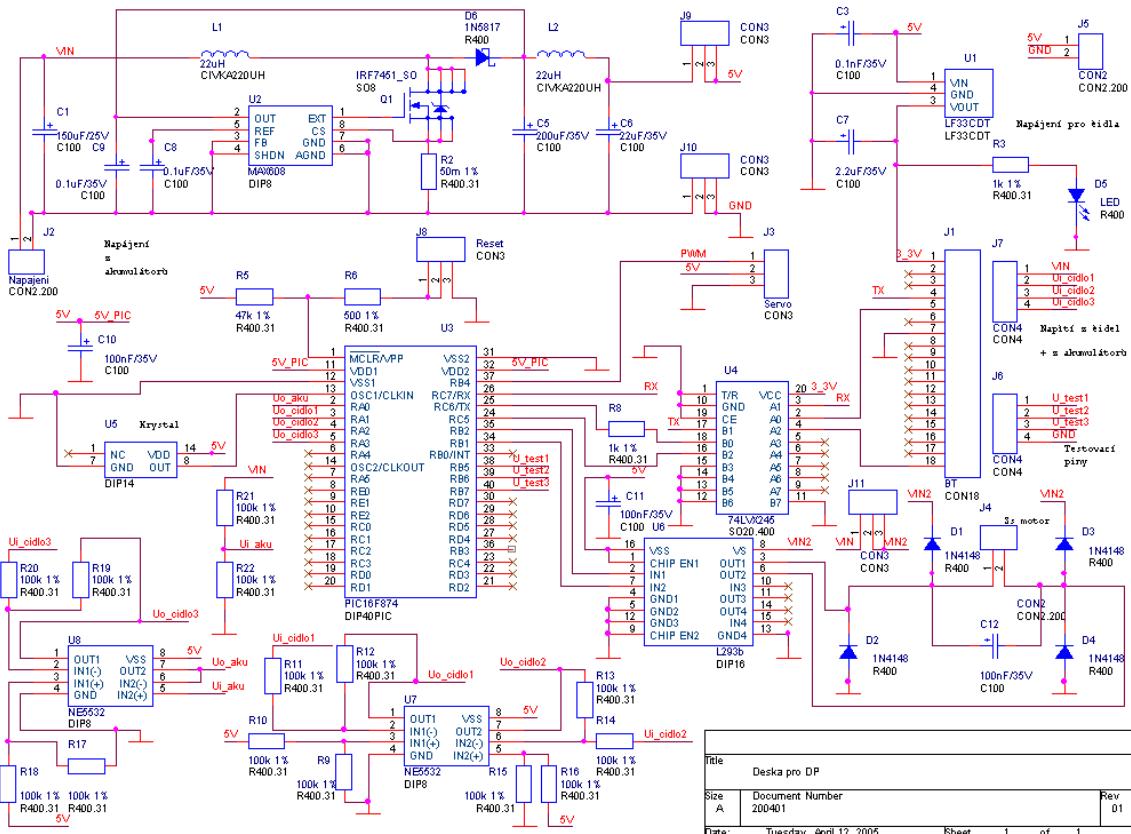
Obrázky, videa – Veškeré použité obrázky ve formátech PNG a JPG.

Projekty – Kompletní projekt pro mikroprocesor v prostředí MPLAB, kompletní projekt aplikace pro počítač a mobilní telefon, deska plošných spojů v programu OrCad.

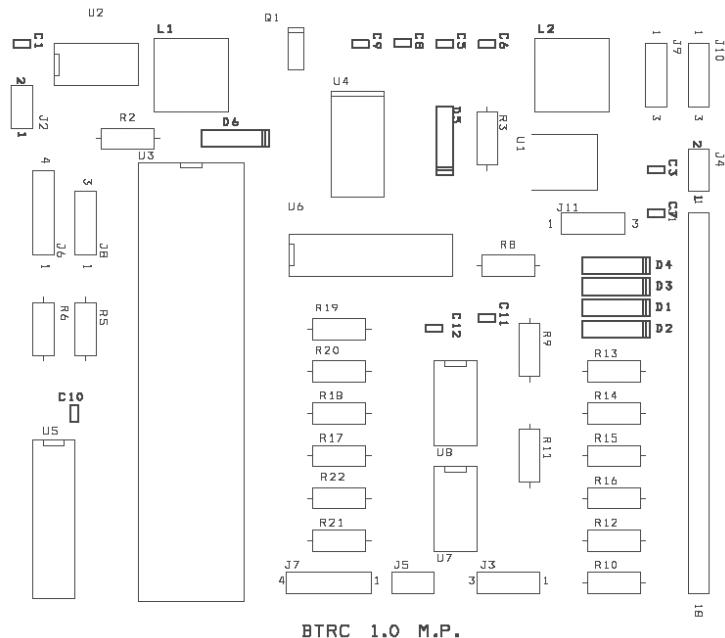
Dodatek B

Realizace výsledné desky

Schéma zapojení elektroniky projektu DP je na obr. B.1.

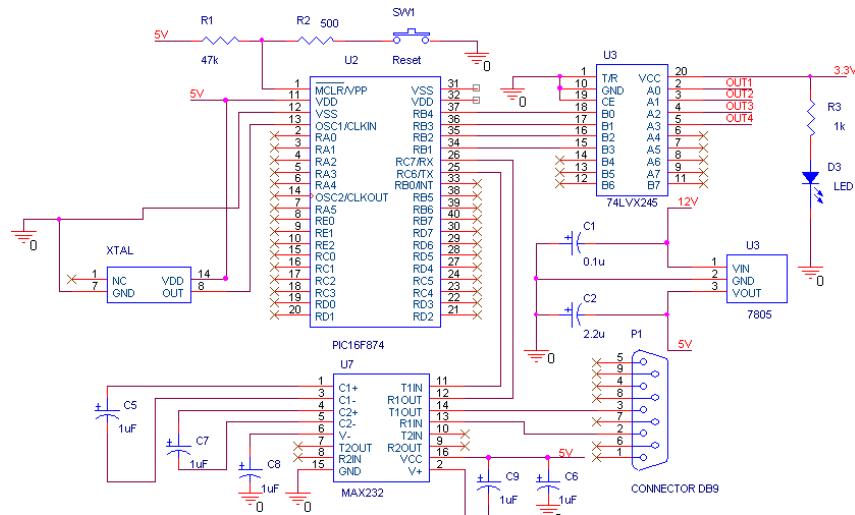


Obrázek B.1: Schéma zapojení elektroniky pro DP



Obrázek B.2: Rozmístění součástek

Schéma zapojení elektroniky projektu BTRC je na obr. B.3.



Obrázek B.3: Schéma zapojení přídavné elektroniky pro BTRC

Dodatek C

Komunikace RS232

Projekt DP

Zachycená komunikace během inicializace a následné komunikace s Bluetooth modulem

Parametrizace Master

01 05 10 00

01 - označení příkazu, 0510 - OpCode: Read buffer size, 00 - délka parametrů: 0 bytů
04 0E 0B 01 05 10 00 C0 00 40 08 00 08 00

04 - označení události, 0E - typ události: Command Complete, 0B - délka parametrů: 11 bytů, 0510 - OpCode: Read Buffer Size, 00 - Status: OK, C000 - délka datového paketu HC ACL: 192 bytů, 40 - délka datového paketu HC SCO, 0800 - celkový počet HC ACL datových paketů: 8, 0800 - celkový počet HC SCO datových paketů: 8

01 09 10 00

01 - označení příkazu, 0910 - OpCode: Read BD ADDR, 00 - délka parametrů: 0 bytů

04 0E 0A 01 09 10 00 6C 40 60 06 00 08

04 - označení události, 0E - typ události: Command Complete, 0A - délka parametrů: 10 bytů, 0109 - OpCode: Read BD ADDR, 00 - Status: OK. 6C4060060008 - BD ADDR

Parametrizace Slave

01 05 0C 03 02 00 02

01 - označení příkazu, 050C - OpCode: HCI Set Event Filter, 03 - délka parametrů: 3 byty, 02 - typ filtru: nastavení spojení, 00 - typ podmínky filtru Inquiry Result: nové zařízení odpovídá na inquiry, 02 - typ filtru Connection Setup: povoleni spojeni

se zařízením se specif. BD ADDR

04 0E 04 01 05 0C 00

04 - označení události, 0E - typ události: Command Complete, 04 - délka parametrů: 4 byty, 01 - počet HCI paketů: 1, 050C - OpCode: HCI Set Event Filter, 00 - Status: OK

01 1C 0C 04 40 00 20 00

01 - označení příkazu, 1C0C - OpCode: Write Page Scan Activity, 04 - délka parametrů: 4 byty, 4000 - interval Page skenu: $64 \times 0.625\text{ms} = 40\text{ms}$, 2000 - interval Page okna: $32 \times 0.625\text{ms} = 20\text{ms}$

04 0E 04 01 1C 0C 00

04 - označení události, 0E - typ události: Command Complete, 04 - délka parametrů: 4 byty, 01 - počet HCI paketů: 1, 1C0C - OpCode, 00 - Status: OK

01 1E 0C 04 40 00 20 00

01 - označení příkazu, 1E0C - OpCode: Write Inquiry Scan Activity, 04 - délka parametrů: 4 byty, 4000 - interval Inquiry skenu: $64 \times 0.625\text{ms} = 40\text{ms}$, 2000 - interval Inquiry okna: $32 \times 0.625\text{ms} = 20\text{ms}$

04 0E 04 01 1E 0C 00

04 - označení události, 0E - typ události: Command Complete, 04 - délka parametrů: 4 byty, 01 - počet HCI paketů: 1, 1E0C - OpCode, 00 - Status: OK

01 1A 0C 01 03

01 - označení příkazu, 1A0C - OpCode: Write Scan Enable, 01 - délka parametrů: 1 byte, 03 - povolen0 skenu: Inquiry a Page skeny povoleny

04 0E 04 01 1A 0C 00

04 - označení události, 0E - typ události: Command Complete, 04 - délka parametrů: 4 byty, 01 - počet HCI paketů: 1, 1A0C - OpCode, 00 - Status: OK

Master vyhledává zařízení pomocí příkazu Inquiry

01 01 04 05 33 8B 9E 0A 08

01 - označení příkazu, 050C - OpCode: HCI Inquiry, 05 - délka parametrů: 5 bytů, 338B9E - LAP, 0A - délka Inquiry: $10 \times 1.28 = 12.8\text{s}$, 08 - počet zařízení, od kterých Master přijme odpověď na Inquiry : 8

04 0F 04 00 01 01 04 04 02 0F 01 B2 A1 5E 06 00 08 01 00 00 00 00

0E 16 04 01 01 00

04 - označení události, 0E - typ události: Command Status Event, 04 - délka parametrů: 4 byty, 00 - Status: OK, 01 - počet HCI paketů: 1, 01 04 - OpCode, 04 - označení události, 02 - typ události: Inquiry Result, 0F - délka parametrů: 15, 01 - počet zařízení, která odpověděla na Inquiry, B2 A1 5E 06 00 08 - BD ADDR: 08 00 06 5E A1 B2, 01 - režim opakování Page skenu: R1, 00 - režim periody Page skenu: P0, 00 - režim Page skenu: hlavní sken, 00 00 00 - třída zařízení, 0E 16 - hodinový offset: 16 0E, 04 - označení události, 01 - typ události: Inquiry Complete, 01 - délka

parametrů, 00 - Status: OK

Master zašle příkaz na vytvoření spojení - reakce na straně Master

01 05 04 0D B2 A1 5E 06 00 08 18 CC 00 00 0E 96 00

01 - označení příkazu, 0504 - OpCode: Create Connection, 0D - délka parametrů: 13 bytů, B2 A1 5E 06 00 08 - BD adresa zařízení: 08 00 06 5E A1 B2, 18 CC - typ paketu, 00 - režim opakování Page skenu: R0, 00 - režim Page skenu: hlavní, 0E 96 - hodinový offset: 96 0E, 00 - povolení záměny rolí

04 0F 04 00 01 05 04 04 03 0B 00 28 00 B2 A1 5E 06 00 08 01 00 04 1B 03 28 00 05

04 - označení události, 0F - typ události: Command Status Event, 04 - délka parametrů: 4 byty, 00 - Status: OK, 01 - počet paketů HCI příkazů, 05 04 - OpCode - Create Connection, 04 - označení události, 03 - typ události - Connection Complete, 0B - délka parametrů: 11, 00 - Status: OK, 28 00 - Connection Handle, B2 A1 5E 06 00 08 - BD adresa, 01 - typ vytvořeného spojení: ACL, 00 - režim kódování: zakázán, 04 - označení události, 1B - typ události: maximální změna rámců, 03 - délka parametrů, 28 00 - Connection Handle, 05 - LMP Max Slots: mohou být 1,3 i 5

Master zašle příkaz na vytvoření spojení - reakce na straně Slave

04 03 0B 00 28 00 B2 A1 5E 06 00 08 01 00 04 1B 03 28 00 05

04 - označení události, 03 - typ události: Connection Complete, 0B - délka parametrů, 00 - Status: OK, 28 00 - Connection handle, B2 A1 5E 06 00 08 - BD adresa: 08 00 06 5E A1 B2, 01 - typ vytvořeného spojení: ACL, 00 - režim kódování: zakázán, 04 - označení události, 1B - typ události: Max Slot Change, 03 - délka parametrů: 3, 28 00 - Connection Handle, 05 - LMP Max Slots: může být 1, 3 i 5 časových slotů

Zaslání dat pomocí ACL spojení

02 28 20 01 00 73

02 - označení datového paketu, 28 20 - Connection Handle, 01 00 - délka dat: 1 byte, 73 - datový byte **04 13 05 01 28 00 01 00**

04 - označení události, 13 - typ události - kompletně odeslané pakety, 05 - délka parametrů, 01 - počet spojení: 1, 28 00 - Connection Handle, 01 00 - počet kompletně odeslaných paketů: 1

Přijetí dat pomocí ACL spojení

02 28 20 01 00 73

02 - označení datového paketu, 28 20 - Connection Handle, 01 00 - délka dat: 1 byte, 73 - datový byte

Slave zašle příkaz rozpojení - reakce na straně Slave

01 06 04 03 28 00 13

01 - označení příkazu, 06 04 - OpCode: HCI Disconnect, 03 - délka parametrů: 3, 28 00 - Connection Handle, 13 - Důvod -*i* Uživatel ukončil spojení

04 0F 04 00 01 06 04 04 05 04 00 28 00 16 04 - označení události, 0F - typ události - Command Status, 04 - délka parametrů, 00 - Status: OK, 01 - počet HCI příkazů, 06 04 - OpCode, 04 - označení události, 05 - typ události - Disconnection Complete, 04 - délka parametrů, 00 - Status: OK, 28 00 - Connection Handle, 16 - Důvod: Spojení ukončeno lokálním hostem

Slave zašle příkaz rozpojení - reakce na straně Master

04 05 04 00 28 00 13

04 - značení události, 05 - typ události - Disconnect Complete, 04 - délka parametrů, 00 - Status: OK, 28 00 - Connection Handle, 13 - důvod: spojení ukončeno na žádost hosta

Projekt BTRC

Komunikace mezi mobilním telefonem (L2CAP) a bluetooth jednotkou

Nastavení Slave

01 03 0C 00

01 - označení příkazu, 03 0C - OpCode: Reset, 00 - délka parametrů

04 0E 04 01 03 0C 00 04 - označení události, 0E - typ události: Command Complete, 04 - délka parametrů: 4, 01 - počet HCI paketů příkazu: 1, 03 0C - OpCode: Reset, 00 - Status - OK

01 1A 0C 01 03

01 - označení příkazu, 1A0C - OpCode: Write Scan Enable, 01 - délka parametrů, 03 - inquiry sken povolen, page sken povolen,

04 0E 04 01 1A 0C 00

04 - označení události, 0E - typ události: Command Complete, 00 - status OK

Vytvoření spojení + zaslání prvého datového paketu - reakce na straně Slave

04 04 0a 8e 9f 39 6d 0e 00 04 02 50 01

04 - označení události, 04 - typ události: Connection Request, 8e 9f 39 6d 0e 00 - BD adresa, 04 02 50 - třída zařízení, 01 - typ spojení: ACL

01 09 04 07 8e 9f 39 6d 0e 00 01

01 - označení příkazu, 0904 - OpCode: Accept Connection, 8e 9f 39 6d 0e 00 - BD adresa, 01 - role: pouze Slave

04 0F 04 00 01 09 04

04 - označení události, typ události: Command Status, 04 - délka: 4, 00 - Status: OK, 01 - počet HCI paketů příkazu: 1, 09 04 - Opcode

04 03 0B 00 28 00 8E 9F 39 6D 0E 00 01 00

04 - označení události, 03 - typ události: Connection Complete, 0B - délka parametrů: 11, 00 - Status: OK, 29 00 - Connection Handle, 8E 9F 39 6D 0E 00 - BD adresa, 01 - typ spojení: ACL, 00 - režim kódování: zakázán

02 28 20 0C 00 08 00 01 00 02 01 04 00 01 20 50 00

02 - datový paket, 2820 Connection Handle, 0C - délka dat: 13, 08 00 - délka paketu L2CAP: 8, 01 00 - CID, 02 - typ L2CAP: Connection Request, 01 - identifikátor L2CAP, 04 00 - délka parametrů, 01 20 - Protocol Service Multiplexor - PSM: nastavuje se v mobil. telefonu, 5F 00 - source CID

04 1B 03 28 00 05

04 - označení události, 1B - typ události: Max Slots Change, 03 délka parametrů: 3, 29 00 - Connection Handle, 05 - LMP: lze používat 1, 3 i 5 časových rámci

02 28 20 10 00 0c 00 01 00 03 01 08 00 FF 01 50 00 00 00 00 00

02 - označení datového paketu, 28 20 - Connection Handle, 10 00 - délka dat: 10, 0C 00 - délka L2CAP paketu: 13, 01 00 - CID, 03 - typ L2CAP: Connection Response, 01 - identifikátor, 08 00 - délka parametrů, FF 01 - cílové CID, 5F 00 - zdrojové, 00 00 - Status: žádné další informace, 00 00 - výsledek - Spojení úspěšné

04 13 05 01 28 00 01 00

04 - označení události, 13 - typ události: Number Of Completed Packets, 05 - délka parametrů: 5, 01 - počet HCI paketů, 29 00 - Connection Handle, 01 00 - počet úspěšně odeslaných paketů

02 28 20 0c 00 08 00 01 00 04 01 04 00 50 00 00 00

02 - označení datového paketu, 28 20 - Connection Handle, 0C 00 - délka dat: 13, 08 00 - délka paketu L2CAP, 01 00 - CID, 04 - typ L2CAP paketu: Configuration Request, 01 - identifikátor, 04 00 - délka parametrů: 4, 50 00 - cílové CID, 00 00 - Príznaky - žádné

04 13 05 01 28 00 01 00

04 - označení události, 13 - typ události: Number Of Completed Packets, 05 - délka

parametrů: 5, 01 - počet HCI paketů, 29 00 - Connection Handle, 01 00 - počet úspěšně odeslaných paketů

02 28 20 0e 00 0a 00 01 00 05 02 06 00 50 00 00 00 00 00

02 - označení datového paketu, 28 20 - Connection Handle, 0E 00 - délka parametrů: 14 bytů, 0A 00 - délka paketu L2CAP: 10 bytů, 01 00 - CID, 05 - typ L2CAP paketu: Configure Response, 02 - identifikátor, 06 00 - délka L2CAP paketu, 5F 00 - zdrojové CID, 00 00 - příznaky: žádné, 00 00 - výsledek: OK

04 13 05 01 28 00 01 00

04 - označení události, 13 - typ události: Number Of Completed Packets, 05 - délka parametrů: 5, 01 - počet HCI paketů, 29 00 - Connection Handle, 01 00 - počet úspěšně odeslaných paketů

02 28 20 05 00 01 00 FF 01 61

02 - označení datového paketu, 28 20 - Connection Handle, 05 00 - délka paketu: 5 bytů, 01 00 - délka paketu L2CAP, FF 01 - CID, 61 - zaslaná data: 61